Микросхема интегральная К1879ВГ1Т

Руководство по эксплуатации

ЮФКВ.431268.044РЭ

Инв.№подл.	Подп. и дата	Взам.инв.№	Инв.№дубл.	Подп. и дата
47472	30.09.2024			

Содержание

1 Описание и работа микросхемы интегральной К1879ВГ1Т 19

1.1 Назначение изделия 19

1.2 Технические характеристики МК Optimal+ 19

1.2.1 Основные функциональные характеристики 19

1.2.2 Список выводов 23

1.2.3 Расположение выводов Optimal+ 29

1.2.4 Электрические параметры 30

1.2.5 Временные диаграммы и динамические параметры тактовых сигналов и сигналов сброса 33

1.2.5.1 Временные диаграммы внешнего тактового сигнала HSE в режиме Bypass 33

1.2.5.2 Временные диаграммы внешнего тактового сигнала LSE в режиме Bypass 34

1.2.5.3 Временные диаграммы внешнего сигнала сброса NRST 34

1.2.5.4 Временные диаграммы и динамические параметры сигналов интерфейса Ethernet 35

1.2.5.5 Временные диаграммы и динамические параметры сигналов интерфейса I2C 37

1.2.5.6 Временные диаграммы и динамические параметры сигналов интерфейса USART 39

1.2.5.7 Временные диаграммы и динамические параметры сигналов интерфейса SPI 40

Удостоверен ЮФКВ.431268.044-УД

					ЮФКВ.431268.044РЭ				
Изм	**Лист**	**№ докум.**	**Подп.**	**Дата**					
Разраб.		**Чумаченко**			**Микросхема интегральная К1879ВГ1Т Руководство по эксплуатации**	**Лит.**		**Лист**	**Листов**
Пров.		**Панфилов**						**2**	**1162**
Н.контр.									
Утв.		**Черников**							

Инв.№подл.	**Подп. и дата**	**Взам.инв.№**	**Инв.№дубл.**	**Подп. и дата**	**Перв.примен.**
47472	30.09.2024				**ЮФКВ.431268.044**

1.3 Устройство и работа МК Optimal+ 41
1.3.1 Структурная схема 41
1.3.2 Основные режимы работы 46
1.3.2.1 Функциональный режим 46
1.3.2.2 Тестовый режим 47
1.3.3 Карта памяти и шинный коммутатор 48
1.3.3.1 Блок шинного коммутатора 48
1.3.3.2 Карта памяти 52
1.3.4 Подсистема центрального процессора 58
1.3.4.1 Процессорное ядро 58
1.3.4.2 Кэш-память команд (ICACHE) 59
1.3.4.2.1 Программная модель 60
1.3.4.2.2 Программирование L1 CACHE 64
1.3.4.2.3 Поддержание когерентности между кэшем и памятью 65
1.3.4.3 Контроллер прерываний (NVIC) 66
1.3.4.3.1 Регистры 66
1.3.4.3.2 Уровневые и импульсные прерывания 70
1.3.4.3.3 Карта прерываний 71
1.3.5 Подсистема памяти 76
1.3.5.1 Контроллер встроенной флэш-памяти (FMC) 76
1.3.5.1.1 Характеристики контроллера FMC 76
1.3.5.1.2 Устройство и функционирование блока 77
1.3.5.1.3 Управление питанием контроллера FMC 92
1.3.5.1.4 Описание регистров 100
1.3.5.1.5 Указания к программированию 142
1.3.5.2 Внутреннее ОЗУ 146

1.3.5.2.1 Сборка подсистемы основной статической памяти (SRAM) 146
1.3.5.2.2 Статическая память батарейного домена питания (BKPSRAM) 159
1.3.6 Системный конфигурационный контроллер (SYSCR) 165
1.3.6.1 FCELL_SCHMITT 165
1.3.6.2 SD_SEL 166
1.3.6.3 BOOTM 166
1.3.6.4 CFG_SRAM 166
1.3.6.5 CFG_PSLVERR 167
1.3.6.6 PD_CTRL_CFG 167
1.3.7 Подсистема управления сигналами сброса и синхросигналами (RCU) 167
1.3.7.1 Общее описание 167
1.3.7.2 Внешние выводы блока RCU 168
1.3.7.3 Блок управления сигналами сброса (RCTL) 169
1.3.7.3.1 Сброс по питанию 169
1.3.7.3.2 Системный сброс 169
1.3.7.3.3 Сброс резервного домена Backup 171
1.3.7.4 Блок управления синхросигналами (CCTL) 171
1.3.7.4.1 Структура синхросигналов 171
1.3.7.4.2 Внешний синхросигнал HSE 174
1.3.7.4.3 Внутренние RC-генераторы HSI0 и HSI1 174
1.3.7.4.4 Внутренние блоки PLL 175
1.3.7.4.5 Внешний синхросигнал LSE 176
1.3.7.4.6 Внутренний RC-генератор LSI 177
1.3.7.4.7 Системный синхросигнал SYSCLK 177
1.3.7.4.8 Синхросигнал системного таймера SysTick 178
1.3.7.4.9 Детектор отказов генератора HSE 178

1.3.7.4.10 Детектор отказов генератора LSE 179
1.3.7.4.11 Внешние выходы синхросигналов MCO и LSCO 180
1.3.7.5 Блок контроля частоты синхросигналов (CFAU) 180
1.3.7.5.1 Общее описание 180
1.3.7.5.2 Принцип работы 181
1.3.7.5.3 Порядок настройки 183
1.3.7.6 Прерывания RCU 183
1.3.7.7 Карта регистров RCU 184
1.3.7.7.1 Список регистров RCU 184
1.3.7.7.2 Регистр управления источниками синхросигналов (RCU_CSCR) 185
1.3.7.7.3 Регистр статуса источников синхросигналов (RCU_CSSR) 186
1.3.7.7.4 Регистр калибровки внутренних источников синхросигналов (RCU_ICSCR0) 187
1.3.7.7.5 Регистр калибровки внутренних источников синхросигналов (RCU_ICSCR1) 188
1.3.7.7.6 Регистр конфигурации синхросигналов 0 (RCU_CCFGR0) 188
1.3.7.7.7 Регистр конфигурации синхросигналов 1 (RCU_CCFGR1) 189
1.3.7.7.8 Регистр конфигурации синхросигналов 1 (RCU_CCFGR2) 190
1.3.7.7.9 Регистр настроек PLL0 0 (RCU_PLL0CR0) 190
1.3.7.7.10 Регистр настроек PLL0 1 (RCU_PLL0CR1) 191
1.3.7.7.11 Регистр настроек PLL0 2 (RCU_PLL0CR2) 191
1.3.7.7.12 Регистр настроек PLL0 3 (RCU_PLL0CR3) 192
1.3.7.7.13 Регистр настроек PLL1 0 (RCU_PLL1CR0) 192
1.3.7.7.14 Регистр настроек PLL1 1 (RCU_PLL1CR1) 193
1.3.7.7.15 Регистр настроек PLL1 2 (RCU_PLL1CR2) 193
1.3.7.7.16 Регистр разрешения тактирования периферии 0 (RCU_PCER0) 194
1.3.7.7.17 Регистр разрешения тактирования периферии 1 (RCU_PCER1) 196

1.3.7.7.18 Регистр выбора независимых синхросигналов периферии 0 (RCU_PICSR0) 197
1.3.7.7.19 Регистр выбора независимых синхросигналов периферии 1 (RCU_PICSR1) 199
1.3.7.7.20 Регистр выбора независимых синхросигналов периферии 2 (RCU_PICSR2) 201
1.3.7.7.21 Регистр сброса периферии 0 (RCU_PRR0) 202
1.3.7.7.22 Регистр сброса периферии 1 (RCU_PRR1) 204
1.3.7.7.23 Регистр управления сбросом при падении напряжения питания (RCU_BORCR)..... 205
1.3.7.7.24 Регистр управления блоком контроля частоты (RCU_CFACR) 206
1.3.7.7.25 Регистр статуса блока контроля частоты (RCU_CFASR) 207
1.3.7.7.26 Регистр верхнего порога счетчика контроля частоты (RCU_CFAULR) 207
1.3.7.7.27 Регистр нижнего порога счетчика контроля частоты (RCU_CFALLR) 208
1.3.7.7.28 Регистр маски прерываний (RCU_IRQER) 208
1.3.7.7.29 Регистр статуса прерываний (RCU_IRQSR)..... 209
1.3.7.7.30 Регистр сброса прерываний (RCU_IRQCR)..... 209
1.3.7.7.31 Регистр конфигурации резервного домена (RCU_BDCR) 210
1.3.7.7.32 Регистр статуса сбросов (RCU_RSR) 212
1.3.8 Система управления питанием (PMU) 213
1.3.8.1 Схема и домены питания МК Optimal+..... 213
1.3.8.1.1 Схема питания МК Optimal+..... 213
1.3.8.1.2 Порты питания и земли..... 213
1.3.8.1.3 Встроенные регуляторы напряжения (HLDO, DCDC, BLDO) 214
1.3.8.1.4 Блоки контроля напряжения питания (POR-BOR)..... 215

1.3.8.1.5 Домены питания 216
1.3.8.2 Режимы энергопотребления МК Optimal+ 217
1.3.8.2.1 Перечень режимов энергопотребления 217
1.3.8.2.2 Режим нормальной работы RUN 217
1.3.8.2.3 Режим пониженного энергопотребления DEEPSLEEP0 218
1.3.8.2.4 Режим пониженного энергопотребления DEEPSLEEP1 218
1.3.8.2.5 Режим пониженного энергопотребления SHUTDOWN 219
1.3.8.2.6 Режим батарейного домена VBAT 219
1.3.8.3 Способы переключения режимов энергопотребления 220
1.3.8.3.1 Диаграмма переходов между режимами энергопотребления 220
1.3.8.3.2 Контроллер управления питанием PMU 220
1.3.9 Цифровые интерфейсы 237
1.3.9.1 Контроллер интерфейса GPIO (GPIO) 237
1.3.9.1.1 Основные характеристики контроллера GPIO 237
1.3.9.1.2 Описание работы контроллера GPIO 238
1.3.9.1.3 Описание регистров контроллера GPIO 243
1.3.9.2 Контроллер интерфейса GPIO батарейного домена питания (BKPGPIO) 250
1.3.9.2.1 Описание BKPGPIO 250
1.3.9.2.2 Описание регистров 251
1.3.9.3 Контроллер интерфейса USART (USART) 256
1.3.9.3.1 Особенности USART контроллера 256
1.3.9.3.2 Структурная схема USART 257
1.3.9.3.3 Описание внешних выводов контроллера USART 259
1.3.9.3.4 Принципы функционирования USART 259
1.3.9.3.5 Описание регистров 275
1.3.9.4 Контроллер интерфейса I2C (I2C) 288

1.3.9.4.1 Характеристики контроллера I2C 288
1.3.9.4.2 Общее описание 288
1.3.9.4.3 Описание внешних выводов контроллера I2C 289
1.3.9.4.4 Протокол передачи 290
1.3.9.4.5 Описание регистров 291
1.3.9.4.6 I2C_Slave 293
1.3.9.4.7 I2C_Master 299
1.3.9.4.8 Интеграция в микроконтроллере 307
1.3.9.5 Контроллер интерфейса CAN (CAN) 308
1.3.9.5.1 Основные характеристики CAN контроллера 308
1.3.9.5.2 Структурная схема контроллера CAN 308
1.3.9.5.3 Описание внешних выводов контроллера CAN 309
1.3.9.5.4 Функциональное описание контроллера CAN 309
1.3.9.5.5 Карта регистров 351
1.3.9.5.6 4. Формат кадра CAN FD 416
1.3.9.6 Контроллер интерфейса SPI (SPI) 440
1.3.9.6.1 Характеристики контроллера SPI 440
1.3.9.6.2 Структурная схема контролера SPI 440
1.3.9.6.3 Описание внешних выводов контроллера SPI 442
1.3.9.6.4 Работа контроллера SPI 442
1.3.9.6.5 Карта регистров контролера SPI 458
1.3.9.6.6 Описание регистров контролера SPI 459
1.3.9.7 Сборка контроллеров низкоскоростных интерфейсов (LSIF_PORT) 465
1.3.9.7.1 Подключение контроллеров низкоскоростных интерфесов к IFMUX .. 465
1.3.9.7.2 Описание внешних выводов блка IFMUX 467
1.3.9.8 Контроллер интерфейса LCD (LCD) 467

1.3.9.8.1 Основные характеристики контроллера LCD 467
1.3.9.8.2 Описание LCD контроллера 467
1.3.9.8.3 Описание внешних выводов контроллера LCD 468
1.3.9.8.4 Физические принципы работы LCD дисплея 469
1.3.9.8.5 Драйверы выходов 471
1.3.9.8.6 Принципы генерирования выходного напряжения на COMx и SEGx выходах 471
1.3.9.8.7 Расчет выходного напряжения 473
1.3.9.8.8 Буферы кадров 474
1.3.9.8.9 Dead time 475
1.3.9.8.10 MUX SEG и COM 476
1.3.9.8.11 Перенастройка выходных уровней напряжения (тестовый режим) 476
1.3.9.8.12 Рекомендации по программированию LCD контроллера 477
1.3.9.8.13 Прерывания 477
1.3.9.8.14 Интеграция 478
1.3.9.8.15 Описание регистров 478
1.3.9.9 Контроллер интерфейса USB (USB_FS) 490
1.3.9.9.1 Характеристики контроллера USB_FS 490
1.3.9.9.2 Описание внешних выводов контроллера USB 491
1.3.9.9.3 Устройство и функционирование блока 491
1.3.9.9.4 Описание регистров 494
1.3.9.9.5 Примечания к программному обеспечению 535
1.3.9.9.6 Описание дополнительных управляющих регистров USB_AUX 569
1.3.9.9.7 Интеграция в микроконтроллере 570
1.3.9.10 Сборка высокоскоростных контроллеров внешней памяти (HSI) 571
1.3.9.10.1 Описание внешних выводов HSI 572

1.3.9.10.2 Контроллер интерфейса xSPI (xSPI) ... 573
1.3.9.10.3 Контроллер SD/eMMC (SD/eMMC) ... 659
1.3.9.11 Контроллер интерфейса Ethernet (ETH) ... 743
1.3.9.11.1 Характеристики контроллера ETH ... 743
1.3.9.11.2 Устройство и функционирование блока ... 743
1.3.9.11.3 Описание регистров ... 751
1.3.9.11.4 Интеграция в микроконтроллере ... 758
1.3.10 Контроллер внешних прерываний (EXTINT) ... 759
1.3.10.1 Характеристики контроллера внешних прерываний EXTINT ... 759
1.3.10.2 Устройство и функционирование блока ... 760
1.3.10.2.1 Выдача событий ... 762
1.3.10.2.2 Фильтрация сигнала ... 762
1.3.10.2.3 Обработка обнаруженного события ... 763
1.3.10.2.4 Уровневый режим работы ... 763
1.3.10.2.5 Программный запуск событий ... 764
1.3.10.2.6 Временные характеристики ... 764
1.3.10.3 Описание регистров ... 764
1.3.10.3.1 Общее описание регистров ... 764
1.3.10.3.2 Поддержка TrustZone ... 765
1.3.10.3.3 Карта регистров ... 765
1.3.10.3.4 Описание полей регистров ... 766
1.3.10.4 Указания к программированию ... 775
1.3.10.4.1 Ограничения ... 775
1.3.10.4.2 Работа с фильтром ... 775
1.3.10.4.3 Внешний сброс ... 776
1.3.10.5 Интеграция контроллера EXTINT в МК Optimal+ ... 776

1.3.10.5.1 Основной контроллер EXTINT 776
1.3.10.5.2 EXTINT_ACOMP 777
1.3.10.5.3 EXTINT_AON 777
1.3.11 Подсистема прямого обмена событиями (ELC) 777
1.3.11.1 Общее описание 777
1.3.11.1.1 Устройство и функционирование блока 778
1.3.11.1.2 Программный запуск триггеров 779
1.3.11.1.3 Безопасный режим 780
1.3.11.1.4 Блокировка изменения конфигурации 780
1.3.11.2 Список событий 780
1.3.11.3 Список триггеров 785
1.3.11.4 Описание регистров 787
1.3.11.4.1 Общее описание регистров 787
1.3.11.4.2 Поддержка безопасности регистров 788
1.3.11.4.3 Карта регистров 788
1.3.11.4.4 Описание полей регистров 790
1.3.12 Контроллер прямого доступа в память (GPDMA) 794
1.3.12.1 Характеристики контроллера GPDMA 794
1.3.12.2 Описание функционирования контроллера GPDMA 794
1.3.12.2.1 Структурная схема контроллера GPDMA 794
1.3.12.2.2 Описание функционирования контроллера GPDMA 795
1.3.12.3 Описание регистров контроллера GPDMA 818
1.3.12.3.1 Карта регистров контроллера GPDMA 818
1.3.12.3.2 Описание полей регистров контроллера GPDMA 819
1.3.13 Подсистема таймеров 828
1.3.13.1 Общее описание подсистемы таймеров 828

1.3.13.2 Таймер процессорного ядра (SysTick) 829
1.3.13.2.1 Описание SysTick 829
1.3.13.2.2 Описание регистров 829
1.3.13.3 Блок системных таймеров общего назначения (SYSTIM) 831
1.3.13.3.1 Устройство блока системных таймеров общего назначения 832
1.3.13.3.2 Описание регистров 833
1.3.13.3.3 Прерывания 839
1.3.13.4 Таймеры общего назначения (TIM) 839
1.3.13.4.1 Структурная схема таймеров общего назначения (TIM) 840
1.3.13.4.2 Описание внешних выводов блоков TIM 842
1.3.13.4.3 Описание счетчика 842
1.3.13.4.4 Теневые регистры 844
1.3.13.4.5 События 846
1.3.13.4.6 Прерывания 849
1.3.13.4.7 Канал захвата 849
1.3.13.4.8 Измерение характеристик входного сигнала 850
1.3.13.4.9 Объединение каналов 851
1.3.13.4.10 Канал сравнения (ШИМ) 851
1.3.13.4.11 Break функция 852
1.3.13.4.12 Dead Zone функция 855
1.3.13.4.13 Одиночный режим 856
1.3.13.4.14 Режим энкодера 856
1.3.13.4.15 Интеграция модуля TIM 858
1.3.13.4.16 Описание регистров 858
1.3.13.5 Блок сторожевого таймера (WDT) 872
1.3.13.5.1 Основные характеристики сторожевого таймера (WDT) 872

1.3.13.5.2 Интеграция сторожевого таймера WDT 873
1.3.13.5.3 Описание работы сторожевого таймера 875
1.3.13.5.4 Описание регистров сторожевого таймера WDT 892
1.3.13.6 Блок счётчика реального времени (RTC) 902
1.3.13.6.1 Характеристики RTC 902
1.3.13.6.2 Описание RTC 902
1.3.13.6.3 Описание внешних выводов блока RTC 903
1.3.13.6.4 Программный сброс 904
1.3.13.6.5 Защита от записи. 904
1.3.13.6.6 Делитель тактового сигнала 904
1.3.13.6.7 Инициализация RTC 904
1.3.13.6.8 Будильники 905
1.3.13.6.9 Временные метки 907
1.3.13.6.10 Wakeup событие 907
1.3.13.6.11 Поддержка TrustZone 907
1.3.13.6.12 Корректировка RTC 908
1.3.13.6.13 Синхронизация между доменами 909
1.3.13.6.14 Восстановление тактового сигнала в APB-домене 912
1.3.13.6.15 Прерывания 912
1.3.13.6.16 Интеграция RTC 913
1.3.13.6.17 Описание регистров 913
1.3.14 Подсистема аналоговых блоков 925
1.3.14.1 Введение 925
1.3.14.2 Состав Подсистема аналоговых блоков 926
1.3.14.3 Основные характеристики подсистемы аналоговых блоков 927
1.3.14.4 Основные параметры подсистемы аналоговых блоков 927

1.3.14.5 Структурная схема подсистемы аналоговых блоков 927
1.3.14.5.1 Введение 927
1.3.14.5.2 Основные характеристики аналогового макроблока 929
1.3.14.5.3 Основные параметры аналогового макроблока 929
1.3.14.5.4 Внешние выводы аналогового макроблока 929
1.3.14.5.5 Мультиплексор каналов АЦП 930
1.3.14.6 Блок аналого-цифрового преобразователя (ADC) 933
1.3.14.6.1 Основные параметры аналогоцифрового преобразователя 934
1.3.14.6.2 Структурная схема блока аналогоцифрового преобразователя 935
1.3.14.6.3 Управление питанием аналогоцифрового преобразователя 937
1.3.14.6.4 Организация работы блока аналогоцифрового преобразователя 940
1.3.14.7 Блок цифро-аналоговых преобразователей (DAC) 994
1.3.14.7.1 Основные параметры цифроаналоговых преобразователей 994
1.3.14.7.2 Структурная схема блока цифроаналоговых преобразователей 996
1.3.14.7.3 Управление питанием цифроаналоговых преобразователей 998
1.3.14.7.4 Организация работы блока цифроаналоговых преобразователей 998
1.3.14.8 Источник опорного напряжения (VREF) 1022
1.3.14.8.1 Введение 1022
1.3.14.8.2 Функциональное описание VREFBUF 1023
1.3.14.9 Аналоговый компаратор (ACOMP) 1029
1.3.14.9.1 Введение 1029
1.3.14.9.2 Основные характеристики COMP 1030
1.3.14.9.3 Основные параметры компараторов 1030
1.3.14.9.4 Механизм блокировки конфмгурационных регистров компаратора ... 1031
1.3.14.9.5 Работа блока компараторов в режиме "окна" 1032
1.3.14.9.6 Гистерезис 1032

1.3.14.9.7 Функция подавления коротких импульсов срабатывания компаратора 1032
1.3.14.9.8 Работа цифрового фильтра блока аналоговых компараторов 1033
1.3.14.10 Операционный усилитель (PGA) 1038
1.3.14.10.1 Введение 1038
1.3.14.10.2 Основные характеристики ОУ 1038
1.3.14.10.3 Описание функций ОУ 1038
1.3.14.10.4 Регистры блоков ОУ 1044
1.3.14.10.5 PGAx_CSR 1045
1.3.14.10.6 PGA_SELECT_OPERANDS 1045
1.3.14.10.7 PGA_SELECT_LEVEL 1046
1.3.14.10.8 PGAx_CSR 1046
1.3.14.10.9 PGA_SELECT_OPERANDS 1047
1.3.14.10.10 PGA_SELECT_LEVEL 1047
1.3.14.10.11 TSENS_ENABLE 1048
1.3.14.10.12 VBAT13_ENABLE 1048
1.3.14.11 Термодатчик (TSENS) 1050
1.3.14.11.1 Блок термодатчика (TSENS) 1050
1.3.15 Подсистема безопасности 1051
1.3.15.1 Ключевые функциональные характеристики 1051
1.3.15.2 Архитектура безопасности TrustZone 1052
1.3.15.2.1 Состав и структура архитектуры TrustZone 1052
1.3.15.2.2 Включение и начальное состояние компонент архитектуры TrustZone 1054
1.3.15.3 Блок ускорения криптографических и математических операций (CMU) 1054
1.3.15.3.1 Характеристики блока CMU 1054
1.3.15.3.2 Устройство и функционирование блока CMU 1056

1.3.15.3.3 Алгоритм работы с ключевой информацией 1057
1.3.15.4 Блок расчета контрольных сумм (CRC) 1061
1.3.15.4.1 Основные характеристики блока расчета контрольных сумм (CRC) .. 1061
1.3.15.4.2 Описание работы блока расчета контрольных сумм CRC 1062
1.3.15.4.3 Описание регистров блока расчета контрольных сумм (CRC)..... 1074
1.3.15.5 Блок генерации случайных чисел (TRNG)..... 1083
1.3.15.5.1 Генератор случайных чисел (TRNG)..... 1083
1.3.15.6 Блок детектирования внешнего воздействия (TAMP)..... 1095
1.3.15.6.1 Характеристики TAMP 1095
1.3.15.6.2 Описание TAMP контроллера..... 1095
1.3.15.6.3 Описание внешних выводов контроллера TAMP 1097
1.3.15.6.4 Описание внутренних сигналов..... 1097
1.3.15.6.5 Защита на запись 1098
1.3.15.6.6 Настройка доступа к backup registers..... 1098
1.3.15.6.7 Разделение доменов 1099
1.3.15.6.8 Режимы работы TAMP..... 1099
1.3.15.6.9 Выходные генерируемые события..... 1101
1.3.15.6.10 Поддержка TrustZone 1101
1.3.15.6.11 Прерывания..... 1101
1.3.15.6.12 Описание регистров 1102
1.3.16 Подсистема отладки и трассировки (DBG)..... 1110
1.3.16.1 Контроллер отладочного порта SWJ-DP..... 1112
1.3.16.1.1 Контроллер JTAG порта 1114
1.3.16.1.2 Переключение контроллера SWJ-DP в режим Serial-Wire..... 1118
1.3.16.1.3 Переключение контроллера SWJ-DP в режим JTAG..... 1119
1.3.16.1.4 Регистры контроллера отладочного порта SWJ-DP..... 1120

1.3.16.2 Корневые таблицы подсистемы отладки и трассировки 1125
1.3.16.2.1 Главная корневая таблица подсистемы отладки и трассировки ROM Table 1125
1.3.16.2.2 Корневая таблица компонентов отладки и трассировки ядра Core ROM Table 1126
1.3.16.3 Компоненты Secure Debug Channel 1127
1.3.16.3.1 Программно доступные регистры компонентов SDC 1128
1.3.16.4 Генератор временных меток Timestamp Generator 1133
1.3.16.4.1 Программно доступные регистры генератора TSGen 1133
1.3.16.5 Поты доступа Core AHB-AP и System AHB-AP 1135
1.3.16.6 Компоненты отладки и трассировки ядра 1135
1.3.16.6.1 Компонент CTI 1135
1.3.16.6.2 Компонент BPU 1136
1.3.16.6.3 Компонент DWT 1136
1.3.16.6.4 Компонент ITM 1137
1.3.16.6.5 Компонент SCS 1137
1.3.16.7 Порт сбора трасс TPIU 1137
1.3.16.7.1 Программно доступные регистры порта сбора трасс TPIU 1138
1.3.16.8 Блок управления системными сигналами Debug Sysregs 1142
1.3.16.8.1 Программно доступные регистры блока Debug Sysregs 1143
1.3.16.9 Блок Debug Logic Decoder 1146
1.3.16.9.1 Программно доступные регистры блока Debug Logic Decoder 1146
1.3.17 Режимы загрузки МК Optimal+ 1147
1.4 Маркировка и пломбирование 1149
1.5 Упаковка 1150
2 Использование по назначению 1153
2.1 Эксплуатационные ограничения 1153

2.1.1 Эксплуатационные ограничения по температуре окружающей среды 1153
2.1.2 Эксплуатационные ограничения по электрическим характеристикам 1153
2.2 Подготовка изделия к использованию 1153
2.2.1 Указания о взаимосвязи (соединении) данного изделия с другими изделиями 1153
2.2.2 Указания по включению и опробованию работы изделия 1154
2.3 Использование изделия 1155
2.3.1 Порядок контроля работоспособности изделия 1155
2.3.2 Перечень возможных неисправностей в процессе использования изделия по назначению и рекомендации по действиям при их возникновении 1155
2.3.3 Перечень режимов работы изделия, а также характеристики основных режимов работы 1156
2.3.4 Порядок и правила перевода изделия с одного режима работы на другой с указанием необходимого для этого времени 1156
2.3.5 Порядок приведения изделия в исходное положение 1156
2.3.6 Порядок выключения изделия, содержание и последовательность осмотра изделия после окончания работы 1156
3 Техническое обслуживание изделия 1157
4 Текущий ремонт изделия 1158
5 Хранение изделия 1159
6 Транспортирование изделия 1160
7 Утилизация изделия 1161

1 Описание и работа микросхемы интегральной К1879ВГ1Т

1.1 Назначение изделия

Микросхема интегральная К1879ВГ1Т (далее по тексту - МК Optimal+) предназначена для использования в широком спектре энергоэффективных электронных изделий таких как приборы учета (счётчики), устройства кузовной автомобильной электроники, IoT-устройства, носимые персональные устройства, индустриальные сенсоры и пр.

1.2 Технические характеристики МК Optimal+

1.2.1 Основные функциональные характеристики

Основные технические характеристики МК Optimal+ представлены в таблице 1.1.

Таблица 1.1 – Основные технические характеристики МК Optimal+

Наименование параметра		**Значение**
Процессорная подсистема		
Процессорное ядро	Количество, шт.	1
	Система команд	Armv8-M
	Микроархитектура	Cortex-M33
	Максимальная частота, МГц	100
	FPU	Да
	DSP-расширение	Да
	MPU	Да
Кэш-память команд		8 Кбайт
Встроенный контроллер прерываний (NVIC)		Да
Система аппаратной поддержки отладки	DAP-контроллер	Да (JTAG-DP, SW-DP)
	ETM-контроллер	Да
Подсистема ОЗУ		
Встроенное ОЗУ	Объём, Кбайт	128
	Поддержка механизмов контроля и исправления ошибок	Да
Встроенное Standby-ОЗУ (BKPSRAM)	Объём, Кбайт	1
	Поддержка механизмов контроля и исправления ошибок	Да
Подсистема ППЗУ		
Встроенная флеш-память	Общий объём, Кбайт	1024
	Количество банков	2

Наименование параметра		Значение
	Поддержка наложения операций чтения и записи/удаления (Read-While-Write)	Да
	Поддержка механизмов контроля и исправления ошибок	ECC
	Поддержка механизма защиты от записи (WriteProtection)	Да
	Поддерживаемые механизмы программирования	через DAP-контроллер самопрограммирование
Встроенная память параметров (OPTMEM)		Да
Внешняя флеш-память	Тип интерфейса	OctalSPI/xSPI
	Количество, шт.	2
	Тип устройства	Master
	Отображение на память	Да
	Режимы передачи данных	Single-SPI Dual-SPI Quad-SPI OctalSPI xSPI (JESD216)
Сервисная подсистема		
Контроллер DMA	Общее количество каналов	16
Контроллер прямого обмена событиями (ELC)	Количество	1
Контроллер внешних прерываний (EXTINT)	Количество	1
Подсистема таймеров		
WDT	Количество, шт.	1
	Разрядность	16
Независимый WDT	Количество, шт.	1
	Разрядность	16
Таймер процессорного ядра (SysTick)	Количество, шт.	1
	Количество каналов	1
	Разрядность	24
	Направление счета	по убыванию
	Разрядность	24
Системный таймер SYSTIM	Разрядность	32
	Направление счета	down
	Количество каналов	2
	Количество таймеров	1
	Генерация прерываний	есть
Таймеры общего назначения TIM_X0/1	Количество таймеров, шт.	8
	Количество каналов на таймер	4
	Общее количество каналов	32

Наименование параметра		Значение
	Разрядность канала	16
	Режимы работы канала	Счётчик ШИМ CapCom
	Направление счета	по убыванию по возрастанию
	Объединение каналов	есть
	Генерация прерываний	есть
	Реакция на событие	есть
Счётчик часов реального времени (RTC)	Количество	1
	Режимы счёта	Режим календаря
	Возможность тактирования от внешнего источника синхросигнала	Да
Интерфейсная подсистема		
SPI	Количество	4
	Режим работы	Master/Slave
	Разрядность данных	4-16
	Программируемая полярность и фаза синхросигнала	Да
I2C	Количество, шт.	4
	Режим работы	Master/Slave
	Максимальная скорость передачи, Кбит/с	400
UART	Количество, шт.	до 8
USART	Количество, шт.	до 8
LIN	Количество, шт.	до 8
CAN	Количество, шт.	4
	Поддержка FD	Да
USB	Количество, шт.	1
	Режим работы	host/device
	Скорость	FS/LS(host only)
SD/MMC	Количество, шт.	2
	Поддержка стандартов SD	SD 6.0
	Поддержка стандартов MMC	eMMC 5.1
Ethernet	Количество, шт.	1
	Скорость обмена, Мбит/с	10/100
	Режимы передачи	полудуплекс полный дуплекс
	Тип интерфейса данных	RMII
	Тип управляющего интерфейса	MDIO
GPIO	Количество линий	до 50
	Количество, шт.	1

Наименование параметра		Значение
Контроллер сегментных LCD-дисплеев	Поддерживаемая максимальная конфигурация сегментов LCD-дисплея	40x8/44x4
Аналоговая подсистема		
АЦП	Количество, шт.	1
	Тип	SAR
	Разрядность	12
	Количество каналов	16
	Максимальная скорость	1 Мвыб./с
ЦАП	Количество, шт.	2
	Разрядность	12
	Максимальная скорость	1 Мвыб./с
	Выходной усилитель	Да
Программируемый операционный усилитель	Количество, шт.	2
Аналоговый компаратор	Количество, шт.	2
Термодатчик	Количество, шт.	1
Подсистема обеспечения безопасности		
Блок криптоускорителя	Симметричные алгоритмы	AES-128/192/256, Кузнечик, Магма
	Ассиметричные алгоритмы	RSA Diffie-Hellmann
	Хэш-алгоритмы	SHA-1 SHA-224 SHA-256 MD5 ГОСТ 34.11-2018 SM3
Блок генерации случайных чисел	Количество, шт.	1
	Разрядность чисел	32
Аппаратная поддержка TrustZone		Да
Блок расчёта CRC	Количество, шт.	1
	Разрядность данных	8/16/32
	Программируемые коэффициенты полинома	Да
	Программируемая длина полинома	8/16/32
Блок детектирования попыток взлома (TAMP)	Количество внутренних tamper-портов	8
	Количество внешних tamper-портов (выводы)	1
Подсистема генерации синхросигналов и сигналов сброса		
	Высокоскоростной (HSE), шт.	1

Наименование параметра		Значение
Внешние источники опорных синхросигналов	Тип источника синхросигнала	Резонатор Генератор
	Диапазон частот входного синхросигнала HSE	4-48 МГц
	Низкоскоростной (LSE), шт.	1
	Тип источника синхросигнала	Резонатор Генератор
	Диапазон частот входного синхросигнала LSE, кГц	32,768 кГц
Встроенные источники опорных синхросигналов	Высокоскоростной (HSI), шт.	2
	Частота генерируемого синхросигнала HSI	4 -64 МГц
	Низкоскоростной (LSI), шт.	1
	Частота генерируемого синхросигнала LSI	32 кГц/250 Гц
ФАПЧ (PLL), шт.		2
Блок измерения частотной точности синхросигнала (CFAU), шт.		1
Выводы выходного синхросигнала, шт.		2
Подсистема обеспечения и управления питанием		
Диапазон опорного питания, В		1,8 - 3,6
Блок регулятора напряжения	LDO	2
	DC-DC	1
Блок генерации сброса по питанию (POR), шт.		2
Блок мониторинга просадки уровня напряжения электропитания, шт.		2
Поддержка режимов энергосбережения	Количество режимов	5
	Перечень режимов	RUN (6 подрежимов) DEEPSLEEP0 DEEPSLEEP1 SHUTDOWN VBAT
Батарейный домен, шт		1
Корпус		
Тип корпуса		LQFP
Количество выводов		80
Шаг выводов, мм		0,5
Поддержка механизма защиты от считывания флеш-памяти (Readout-защита, RDP)	Да	

1.2.2 Список выводов

Список и функциональное назначение выводов микросхемы интегральной К1879ВГ1Т представлены в таблице 1.2.

Таблица 1.2 – Список выводов микросхемы интегральной К1879ВГ1Т

Номер вывода	Наименование вывода	Тип (Вх/Вых)	Функциональное назначение				
			Порты с функцией пробуждения[1)]	Порты управления LCD дисплеем[2)]	Порты аналоговых интерфейсов[3)]	Порты цифровых интерфейсов общего назначения[1)]	Порты цифровых интерфейсов специального назначения[1),4)]
1	PB9	Вх/Вых	WKP_25	SEG_20		GPIO_B_9 IFMUX_B_9 EXTINT_43	*AF4*: ETH_TXD0 *AF14*: JTDI[5)]
2	PB10	Вх/Вых	WKP_26	SEG_21		GPIO_B_10 IFMUX_B_10 EXTINT_44	*AF4*: ETH_TXD1 *AF14*: NJTRST[5)]
3	PB11	Вх/Вых	WKP_27	SEG_22		GPIO_B_11 IFMUX_B_11 EXTINT_45	*AF4*: ETH_REF_CLK
4	PB17	Вх/Вых		SEG_23		GPIO_B_17 IFMUX_B_17 EXTINT_51	*AF4*: ETH_RX_ER
5	PB18	Вх/Вых		SEG_24		GPIO_B_18 IFMUX_B_18 EXTINT_52	*AF4*: ETH_CRS_DV
6	PB19	Вх/Вых		SEG_25		GPIO_B_19 IFMUX_B_19 EXTINT_53	*AF5*: MDC
7	PB20	Вх/Вых		SEG_26		GPIO_B_20 IFMUX_B_20 EXTINT_54	*AF5*: MDIO
8	VLX_DCDC	Вх					
9	VDD_DCDC	Вх					
10	VSS_DCDC	Вх					
11	VDD11	Вх					
12	VDD33	Вх					
13	VSS	Вх					
14	PC8	Вх/Вых				GPIO_C_8 IFMUX_C_8 EXTINT_63	*AF6*: USB_DM
15	PC9	Вх/Вых				GPIO_C_9 IFMUX_C_9 EXTINT_64	*AF6*: USB_DP
16	PC10	Вх/Вых		SEG_27	ADC_18	GPIO_C_10 IFMUX_C_10 EXTINT_65	
17	PC11	Вх/Вых		SEG_28	ADC_17	GPIO_C_11 IFMUX_C_11 EXTINT_66	
18	PC12	Вх/Вых		SEG_29	ADC_14 PGA0_IN-	GPIO_C_12 IFMUX_C_12 EXTINT_67	*AF5*: MDIO
19	VDD_USB	Вх		VDD_LCD			VDD_USB
20	VDD33	Вх					

Номер вывода	Наименование вывода	Тип (Вх/Вых)	Функциональное назначение				
			Порты с функцией пробуждения[1)]	Порты управления LCD дисплеем[2)]	Порты аналоговых интерфейсов[3)]	Порты цифровых интерфейсов общего назначения[1)]	Порты цифровых интерфейсов специального назначения[1),4)]
21	PC13	Вх/Вых		SEG_30	ADC_13 PGA0_OUT	GPIO_C_13 IFMUX_C_13 EXTINT_68	*AF5*: MDIO
22	PC14	Вх/Вых		SEG_31	ADC_12 PGA0_IN+	GPIO_C_14 IFMUX_C_14 EXTINT_69	
23	PC15	Вх/Вых		SEG_32	ADC_11 COMP0	GPIO_C_15 IFMUX_C_15 EXTINT_70	
24	PC16	Вх/Вых		SEG_33	ADC_10 COMP1	GPIO_C_16 IFMUX_C_16 EXTINT_71	
25	PGA1_IN	Вх			ADC_9 PGA1_IN+		
26	DAC1	Вх/Вых			ADC_2 DAC_1		
27	AVDD	Вх					
28	AVSS	Вх					
29	DAC0	Вх/Вых			ADC_1 DAC_0		
30	VREFp	Вх					
31	PD0	Вх/Вых		SEG_34	ADC_8 PGA1_IN-	GPIO_D_0 IFMUX_D_0 EXTINT_76	
32	PD1	Вх/Вых		SEG_35	ADC_7 PGA1_OUT	GPIO_D_1 IFMUX_D_1 EXTINT_77	
33	PD2	Вх/Вых		SEG_36	ADC_6	GPIO_D_2 IFMUX_D_2 EXTINT_78	*AF0*: HSI0_WP
34	PD3	Вх/Вых		SEG_37	ADC_5	GPIO_D_3 IFMUX_D_3 EXTINT_79	*AF0*: HSI0_SS3 *AF1*: HSI0_PU_PD_DATA2
35	VDD33	Вх					
36	PD4	Вх/Вых		SEG_38	ADC_4	GPIO_D_4 IFMUX_D_4 EXTINT_80	*AF0*: HSI0_SS2 *AF1*: HSI0_BPWR
37	PD5	Вх/Вых		SEG_39	ADC_3	GPIO_D_5 IFMUX_D_5 EXTINT_81	*AF0*: HSI0_SS1 *AF1*: HSI0_CDn
38	PD13	Вх/Вых		COM_0		GPIO_D_13 IFMUX_D_13 EXTINT_89	*AF0*: HSI0_DATA1
39	PD14	Вх/Вых		COM_1		GPIO_D_14 IFMUX_D_14 EXTINT_90	*AF0*: HSI0_DATA0
40	PD15	Вх/Вых		COM_2		GPIO_D_15 IFMUX_D_15 EXTINT_91	*AF0*: HSI0_CLK *AF14*: BOOT0[8)]

Номер вывода	Наименование вывода	Тип (Вх/Вых)	Функциональное назначение				
			Порты с функцией пробуждения[1)]	Порты управления LCD дисплеем[2)]	Порты аналоговых интерфейсов[3)]	Порты цифровых интерфейсов общего назначения[1)]	Порты цифровых интерфейсов специального назначения[1),4)]
41	VDD33	Вх					
42	PD16	Вх/Вых	WKP_0	COM_3		GPIO_D_16 IFMUX_D_16 EXTINT_92	*AF0*: HSI0_CMD *AF1*: HSI0_SS0
43	PD17	Вх/Вых	WKP_1	COM_4		GPIO_D_17 IFMUX_D_17 EXTINT_93	*AF0*: HSI0_DATA3 *AF7*: MCO[6)]
44	PD18	Вх/Вых	WKP_2	COM_5		GPIO_D_18 IFMUX_D_18 EXTINT_94	*AF0*: HSI0_DATA2
45	PD19	Вх/Вых	WKP_3	COM_6		GPIO_D_19 IFMUX_D_19 EXTINT_95	*AF0*: HSI0_DATA4 *AF1*: HSI0_CMD_DIR *AF2*: HSI0_SS1 *AF3*: HSI0_CDn
46	PD25	Вх/Вых	WKP_4	SEG_0		GPIO_D_25 IFMUX_D_25 EXTINT_101	*AF0*: HSI0_SS1 *AF1*: HSI0_CDn *AF2*: HSI0_RST *AF7*: MCO[6)]
47	PD26	Вх/Вых	WKP_5	COM_7		GPIO_D_26 IFMUX_D_26 EXTINT_102	*AF0*: HSI0_DATA5 *AF1*: HSI0_D0_DIR *AF2*: HSI0_SS2 *AF3*: HSI0_BPWR
48	PD27	Вх/Вых	WKP_6	SEG_1		GPIO_D_27 IFMUX_D_27 EXTINT_103	*AF0*: HSI0_DATA6 *AF1*: HSI0_D123_DIR *AF2*: HSI0_SS3 *AF3*: HSI0_PU_PD_DATA2
49	PD28	Вх/Вых	WKP_7	SEG_2		GPIO_D_28 IFMUX_D_28 EXTINT_104	*AF0*: HSI0_DATA7 *AF1*: HSI0_1v8 *AF2*: HSI0_WP *AF7*: MCO[6)]
50	NRST[7)]	Вх/Вых					
51	OSC_IN	Вх					
52	OSC_OUT	Вых					
53	VDD11	Вх					
54	VDD33	Вх					
55	VSS	Вх					
56	PA9	Вх/Вых	WKP_8	SEG_3		GPIO_A_9 IFMUX_A_9 EXTINT_9	*AF0*: HSI1_DS *AF14*: SWO[5)]
57	PA10	Вх/Вых	WKP_9	SEG_4		GPIO_A_10 IFMUX_A_10 EXTINT_10	*AF0*: HSI1_SS1 *AF1*: HSI1_CDn *AF5*: MDC
58	PA11	Вх/Вых	WKP_10	SEG_5		GPIO_A_11 IFMUX_A_11 EXTINT_11	*AF0*: HSI1_DATA1 *AF5*: MDC
59	PA12	Вх/Вых	WKP_11	SEG_6		GPIO_A_12 IFMUX_A_12 EXTINT_12	*AF0*: HSI1_DATA0 *AF4*: ETH_RXD0
60	VDD33	Вх					

Номер вывода	Наименование вывода	Тип (Вх/Вых)	Функциональное назначение				
			Порты с функцией пробуждения[1]	Порты управления LCD дисплеем[2]	Порты аналоговых интерфейсов[3]	Порты цифровых интерфейсов общего назначения[1]	Порты цифровых интерфейсов специального назначения[1),4]
61	PA13	Вх/Вых	WKP_12	SEG_7		GPIO_A_13 IFMUX_A_13 EXTINT_13	*AF0*: HSI1_CLK *AF4*: ETH_RXD1
62	PA14	Вх/Вых	WKP_13	SEG_8		GPIO_A_14 IFMUX_A_14 EXTINT_14	*AF0*: HSI1_CMD *AF1*: HSI1_SS0 *AF4*: ETH_REF_CLK
63	PA15	Вх/Вых	WKP_14	SEG_9		GPIO_A_15 IFMUX_A_15 EXTINT_15	*AF0*: HSI1_DATA3 *AF4*: ETH_RX_ER
64	PA16	Вх/Вых	WKP_15	SEG_10		GPIO_A_16 IFMUX_A_16 EXTINT_16	*AF0*: HSI1_DATA2 *AF4*: ETH_CRS_DV
65	PA21	Вх/Вых	WKP_16	SEG_11		GPIO_A_21 IFMUX_A_21 EXTINT_21	*AF0*: HSI1_DATA4 *AF1*: HSI1_CMD_DIR *AF2*: HSI1_SS2 *AF3*: HSI1_BPWR
66	PA22	Вх/Вых	WKP_17	SEG_12		GPIO_A_22 IFMUX_A_22 EXTINT_22	*AF0*: HSI1_DATA5 *AF1*: HSI1_D0_DIR *AF2*: HSI1_SS3 *AF3*: HSI1_PU_PD_DATA2
67	PA23	Вх/Вых	WKP_18	SEG_13		GPIO_A_23 IFMUX_A_23 EXTINT_23	*AF0*: HSI1_DATA6 *AF1*: HSI1_D123_DIR *AF2*: HSI1_WP
68	PA24	Вх/Вых	WKP_19	SEG_14		GPIO_A_24 IFMUX_A_24 EXTINT_24	*AF0*: HSI1_DATA7 *AF1*: HSI1_1v8 *AF2*: HSI1_RST
69	VDD33	Вх					
70	BVDD33	Вх					
71	BVDD11	Вх					
72	OSC32_IN	Вх					
73	OSC32_OUT	Вых					
74	PE0[6]	Вх/Вых				GPIO_E_0 EXTINT_29	*AF7*: LSCO[6] *AF8*: TAMP_IN0 *AF9*: TAMP_OUT0 *AF10*: RTC_ALARM_A *AF11*: RTC_CALIB *AF12*: RTC_WAKEUP
75	PB4	Вх/Вых	WKP_20	SEG_15		GPIO_B_4 IFMUX_B_4 EXTINT_38	*AF4*: ETH_RXD0
76	PB5	Вх/Вых	WKP_21	SEG_16		GPIO_B_5 IFMUX_B_5 EXTINT_39	*AF4*: ETH_RXD1
77	PB6	Вх/Вых	WKP_22	SEG_17		GPIO_B_6 IFMUX_B_6 EXTINT_40	*AF14*: JTMS/SWDIO[5]
78	PB7	Вх/Вых	WKP_23	SEG_18		GPIO_B_7 IFMUX_B_7 EXTINT_41	*AF14*: JTCK/SWCLK[5]
79	VDD33	Вх					

Номер вывода	Наименование вывода	Тип (Вх/Вых)	Функциональное назначение				
			Порты с функцией пробуждения[1)]	Порты управления LCD дисплеем[2)]	Порты аналоговых интерфейсов[3)]	Порты цифровых интерфейсов общего назначения[1)]	Порты цифровых интерфейсов специального назначения[1),4)]
80	PB8	Вх/Вых	WKP_24	SEG_19		GPIO_B_8 IFMUX_B_8 EXTINT_42	*AF4*: ETH_TX_EN *AF14*: JTDO/SWO[5)]

[1)] Описание выводов с функциями **WKP_x** приведено в разделе 1.3.10.5.3

Описание выводов с функциями **GPIO_x** и **IFMUX_x** приведено в разделе 1.3.9.1.2.2 и 1.3.9.7.2 соответственно. Переключение выводов на работу с функциями **GPIO_x** и **IFMUX_x** производится в блоке GPIO. Описание приведено в 1.3.9.1.2.3 и в 1.3.9.1.2.5; функция **IFMUX_x** соотвествует *AF15*

Описание выводов с функциями **EXTINT_x** приведено в разделе 1.3.10.5.1

Описание выводов с функциями **HSI0_x** и **HSI1_x** приведено в разделе 1.3.9.10.1

Описание выводов с функциями **ETH_x**, **MDIO** и **MDC** приведено в разделе 1.3.9.11.2.2

Описание выводов с функциями **USB_x** приведено в разделе 1.3.9.9.2

[2)] Описание выводов с функцией управления LCD дисплеем приведено в разделе 1.3.9.8.3, переключение выводов на работу с функцией LCD производится в блоке GPIO, описание приведено в 1.3.9.1.2.3

[3)] Описание выводов с аналоговыми функциями приведено в разделе 1.3.14.5.4, переключение выводов на работу с аналоговыми функциями производится в блоке GPIO, описание приведено в 1.3.9.1.2.3

[4)] **AFi**, где **i** изменяется от 0 до 15, для портов цифровых интерфейсов специального назначения указывает номер альтернативной функции при настройке мультиплексирования вывода в контроллере GPIO

[5)] Описание выводов с функциями отладки приведено в разделах 1.3.16.1 и 1.3.16.7

[6)] Описание выводов с функциями **MCO** и **LSCO** приведено в разделе 1.3.7.4.11

[7)] Выводы **PAx**, **PBx**, **PCx** и **PDx** являются 5 В толерантными, выводы **NRST** и **PE0** - не 5 В толерантные

[8)] Описание вывода с функцией **BOOT0** приведено в разделе 1.3.17

Общая информация по нумерации цифровых альтернативных функций GPIO:

– AF0, AF1, AF2, AF3 - выводы сборки высокоскоростных интерфейсов с внешней памятью HSI;

– AF4 - выводы интерфейса Ethernet (ETH);

– AF5 - выводы интерфейса MDIO;

– AF6 - выводы интерфейса USB;

– AF7 - выводы сигналов тактирования **MCO** и **LSCO** (CLKOUT);

– AF8 - выводы TAMP_IN блока TAMP(используется только для PE0);

– AF9 - выводы TAMP_OUT блока TAMP (используется только для PE0);

– AF10 - выводы RTC_ALARM контроллер часов реального времени (используется только для PE0);

– AF11 - выводы RTC_CALIB контроллер часов реального времени (используется только для PE0);

– AF12 - выводы RTC_WAKEUP контроллер часов реального времени (используется только для PE0);

– AF13 - не используется;

– AF14 - выводы отладочных интерфейсов (debug);

– AF15 - выводы блок мультиплексирования низкоскоростных интерфейсов IFMUX (USART, I2C, SPI, CAN, TAMP).

1.2.3 Расположение выводов Optimal+

Расположение выводов корпуса МК Optimal+ приведено на рисунке 1.1.

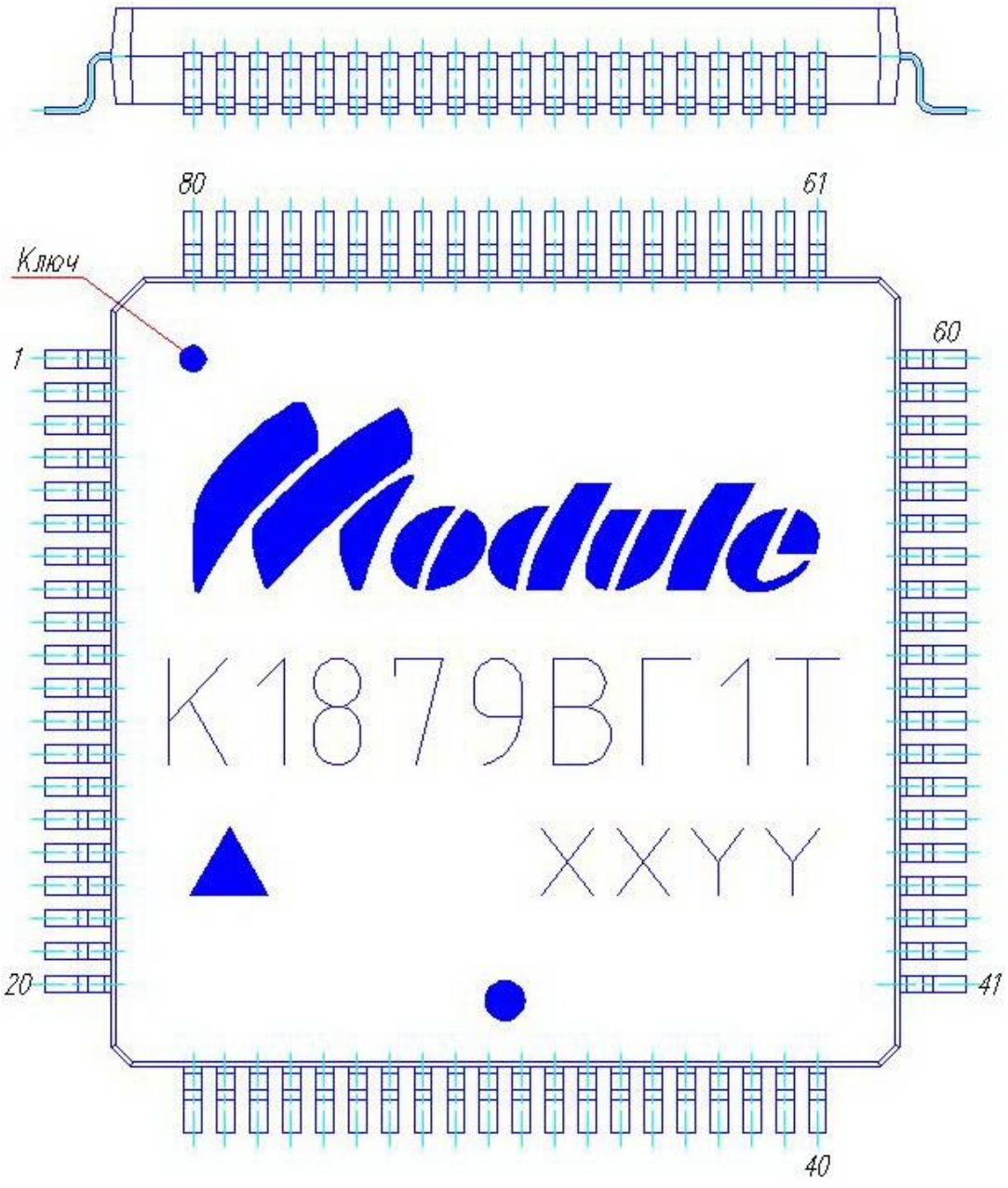


Рисунок 1.1 – Расположение выводов корпуса МК Optimal+

1.2.4 Электрические параметры

В таблице 1.3 приведены предельно допустимые и предельные режимы эксплуатации МК Optimal+.

Таблица 1.3 – Предельно допустимые и предельные режимы эксплуатации МК Optimal+

Наименование параметра режима, единица измерения, режим измерения	Буквенное обозначение	Норма параметра			
		предельно допустимого режима		предельного режима	
		не менее	не более	не менее	не более
Напряжение питания на выводах VDD, В	U_{CC}	1,8	3,6	-0,3	3,9
Напряжение питания DCDC преобразователя на выводе VDD_DCDC, В	U_{CCDC}	1,8	3,6	-0,3	3,9
Напряжение питания аналоговое на выводе AVDD, В	U_{CCA}	1,8	3,6	-0,3	3,9
Напряжение питания батарейное на выводе VBAT, В	U_{CCB}	1,8	3,6	-0,3	3,9
Напряжение питания USB на выводе VDD_USB, В	U_{CCU}	3,0	3,6	-0,3	3,9
Схемы ввода-вывода					
Входное напряжение высокого уровня, В, на входах портов PA, PB, PC, PD: – при $1,8 \le U_{CC} < 2,2$ В – при $2,2 \le U_{CC} < 3,0$ В – при $3,0 \le U_{CC} \le 3,6$ В	U_{IH}	 $0,7 \cdot U_{CC}$ 1,7 2,0	U_{CC}	—	$U_{CC} + 0,3$
Входное напряжение низкого уровня, В, на входах портов PA, PB, PC, PD: – при $1,8 \le U_{CC} < 2,2$ В – при $2,2 \le U_{CC} < 3,0$ В – при $3,0 \le U_{CC} \le 3,6$ В	U_{IL}	 0	 $0,3 \cdot U_{CC}$ 0,7 0,8	 -0,3	 —
Выходной ток высокого уровня, мА, на выводах портов PA, PB, PC, PD	I_{OH}	-8	—	-14	—
Выходной ток низкого уровня, мА, на выводах портов PA, PB, PC, PD	I_{OL}	—	8	—	14
Ёмкость нагрузки, пФ, на выводах портов PA, PB, PC, PD	C_L	—	30	—	—
Генераторы частоты					
Частота следования импульсов тактовых сигналов, МГц	f_C	—	100	—	—
Частота следования импульсов тактовых сигналов HSE на выводе OSC_IN, МГц – в режиме кварцевого осциллятора – в режиме генератора	f_{C_HSE}	 8 8	 48 48	—	—

Наименование параметра режима, единица измерения, режим измерения	Буквенное обозначение	Норма параметра			
		предельно допустимого режима		предельного режима	
		не менее	не более	не менее	не более
Частота следования импульсов тактовых сигналов LSE на выводе OSC32_IN, кГц – в режиме кварцевого осциллятора – в режиме генератора	f_{C_LSE}	 32 32	 33 33	—	—

В таблице 1.4 приведены электрические параметры МК Optimal+ при приемке и поставке.

Таблица 1.4 – Электрические параметры МК Optimal+ при приемке и поставке

Наименование параметра, единица измерения, режим измерения	Буквенное обозначение	Норма параметра		Температура среды, ºС
		не менее	не более	
Выходное напряжение низкого уровня, В, на выводах портов PA, PB, PC, PD: – при U_{CC}=3,6 В: – при I_{OL}= 4 мА – при I_{OL}= 8 мА – при U_{CC}=3,0 В: – при I_{OL}= 3 мА – при I_{OL}= 6 мА – при U_{CC}=2,2 В: – при I_{OL}= 2 мА – при I_{OL}= 4 мА	U_{OL}	—	 0,4 0,4 0,4 0,4 0,4 0,4	 -40 +25 +105
Выходное напряжение высокого уровня, В, на выводах портов PA, PB, PC, PD – при U_{CC}=3,0 В: – при I_{OH}= -4 мА – при I_{OH}= -8 мА – при U_{CC}=2,2 В: – при I_{OH}= -3 мА – при I_{OH}= -6 мА – при U_{CC}=1,8 В: – при I_{OH}= -2 мА – при I_{OH}= -4 мА	U_{OH}	 2,4 2,4 2,0 2,0 1,4 1,4	—	 -40 +25 +105
Входной ток утечки низкого уровня, мкА, на выводах портов PA, PB, PC, PD: – без pullup/pulldown при U_{PAD}=0 В и U_{CC}=3,6	I_{ILL}	-1	+1	-40 +25 +105
Входной ток утечки высокого уровня, мкА, на выводах портов PA, PB, PC, PD – без pullup/pulldown при U_{PAD}=U_{CC}=3,6 В	I_{ILH}	-1	+1	-40 +25 +105

Наименование параметра, единица измерения, режим измерения	Буквенное обозначение	Норма параметра		Температура среды, °С
		не менее	не более	
Ток потребления динамический суммарный по цепям источников питания в режиме RUN.HLDO_xx[1] VDD, BVDD33, AVDD, VDD_USB, мА, при f_C = 100 MHz $U_{CC}=U_{CCDC}=U_{CCA}=U_{CCU}=U_{CCB}$ = 2,2В (для диапазона от 1,8 до 2,2 В); 3,0В (для диапазона от 2,2 до 3,0 В); 3,6В (для диапазона от 3,0 до 3,6 В);	$I_{CC_SUM_RUN}$	 — — —	 202 148 123	 -40 +25 +105
Параметры АЦП				
Разрядность АЦП	E_{NADC}	12		-40 +25 +105
Параметры ЦАП				
Разрядность ЦАП, единиц младшего разряда	E_{NADC}	12		-40 +25 +105
Минимальное выходное напряжение ЦАП, В	U_{ODAC_min}	-	0,08	-40 +25 +105
Максимальное выходное напряжение ЦАП, В	U_{ODAC_max}	U_{REF}– 0,08	-	-40 +25 +105
[1]Режимы энергопотребления описаны в таблице 1.175				

Суммарный ток потребления динамический $I_{CC_SUM_BAT}$ по цепи источника питания в батарейном режиме VBAT при U_{CCB} =3,6В и T=(+25±2)°C не должен превышать 5 мкА. Батарейный режим энергопотребления VBAT описан в Таблице 1.175 при следующих условиях:

– задействованы устройства RTC и TAMP с источником тактирования от осциллятора LSE;

– включена функция детектора отказа LSE (LSECFD);

– блок контроля напряжения BOR33VBAT работает в импульсном режиме со скважностью 1%;

– задействован один буфер GPIO_E в режиме входа, остальные выключены.

1.2.5 Временные диаграммы и динамические параметры тактовых сигналов и сигналов сброса

1.2.5.1 Временные диаграммы внешнего тактового сигнала HSE в режиме Bypass

Осциллятор HSE в режиме Bypass допускает подачу на внешний вывод OSC_IN синхросигнала двух видов:

– аналоговый сигнал синусоидальной формы, в этом случае размах сигнала с учетом постоянной составляющей не должен выходить за диапазон (0, VDD33) вольт;

– цифровой КМОП-сигнал - меандр с уровнем напряжения от 0 до VDD33 вольт.

Временные диаграммы внешнего тактового сигнала HSE приведены на рисунке 1.2, временные параметры приведены в таблице 1.5.

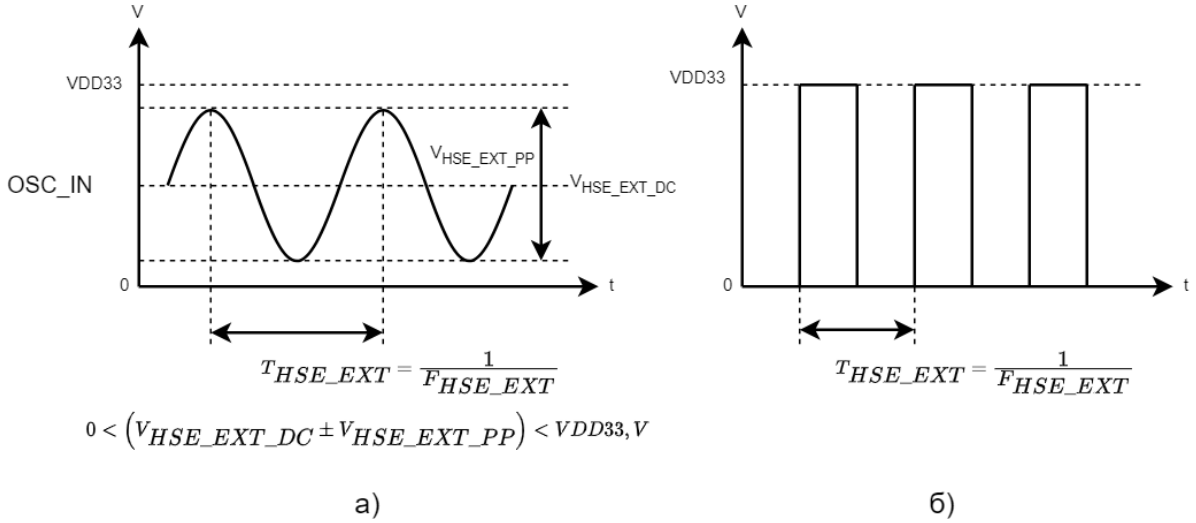


Рисунок 1.2 – Временные диаграммы внешнего синхросигнала HSE в режиме Bypass: а - аналоговый сигнал, б - цифровой КМОП-сигнал

Таблица 1.5 – Временные параметры внешнего синхросигнала HSE

Наименование параметра	Буквенное обозначение	Значение параметра		
		не менее	номинал	не более
Частота синхросигнала HSE	F_{HSE_EXT}	4 МГц	-	48 МГц
Коэффициент заполнения HSE	$DUCY_{HSE_EXT}$	45%	–	55%

1.2.5.2 Временные диаграммы внешнего тактового сигнала LSE в режиме Bypass

Осциллятор LSE в режиме Bypass допускает подачу на внешний вывод OSC32_IN синхросигнала двух видов:

– аналоговый сигнал синусоидальной формы, в этом случае размах сигнала с учетом постоянной составляющей не должен выходить за диапазон (0, BVDD33) вольт.

– цифровой КМОП-сигнал - меандр с уровнем напряжения от 0 до BVDD33 вольт.

Временные диаграммы внешнего тактового сигнала LSE приведены на рисунке 1.3, временные параметры приведены в таблице 1.6.

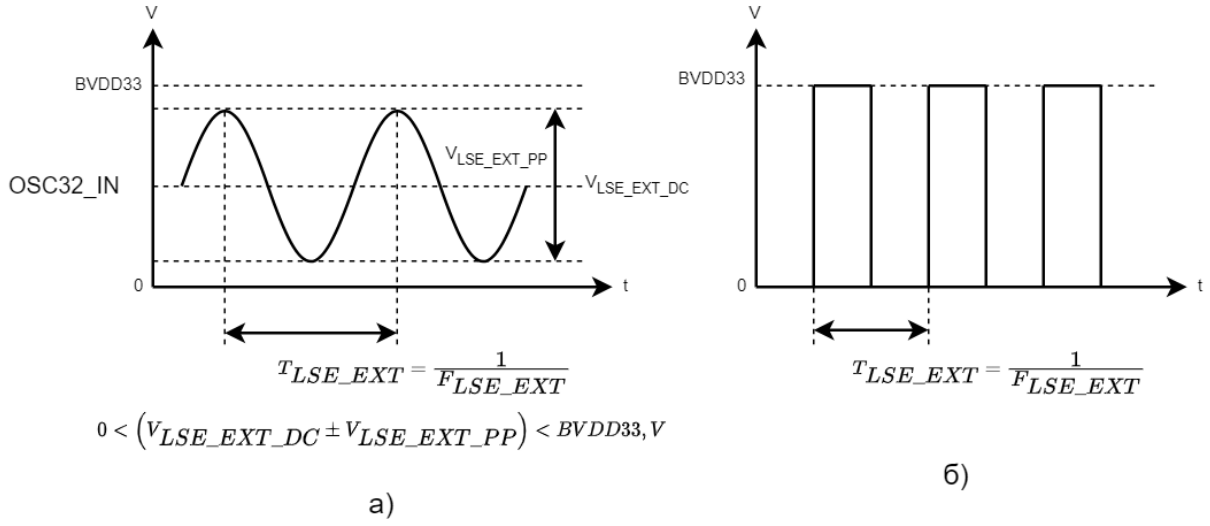


Рисунок 1.3 – Временные диаграммы внешнего синхросигнала LSE в режиме Bypass: а - аналоговый сигнал, б - цифровой КМОП-сигнал

Таблица 1.6 – Временные параметры внешнего синхросигнала LSE

Наименование параметра	Буквенное обозначение	Значение параметра		
		не менее	номинал	не более
Частота синхросигнала LSE	F_{LSE_EXT}	-	32.768 кГц	-
Коэффициент заполнения LSE	$DUCY_{LSE_EXT}$	45%	–	55%

1.2.5.3 Временные диаграммы внешнего сигнала сброса NRST

Временная диаграмма внешнего сброса NRST приведена на рисунке 1.4, временные параметры приведены в таблице 1.7.

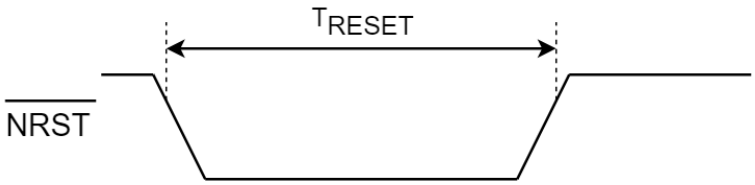


Рисунок 1.4 – Временная диаграмма сигнала сброса NRST

Таблица 1.7 – Временные параметры сигнала сброса NRST

Наименование параметра	Буквенное обозначение	Значение параметра		
		не менее	номинал	не более
Время удержания NRST	T_{RESET}	100 нс	-	-

1.2.5.4 Временные диаграммы и динамические параметры сигналов интерфейса Ethernet

На рисунке 1.5 приведены параметры сигналов интерфейса Ethernet RMII.

Измерения времени должны отсчитываться от точки, в которой сигнал ETH_REF_CLK пересекает опорный уровень напряжения до минимального уровня логической единицы или максимального уровня логического ноля.

Значения параметров приведены в таблице 1.8.

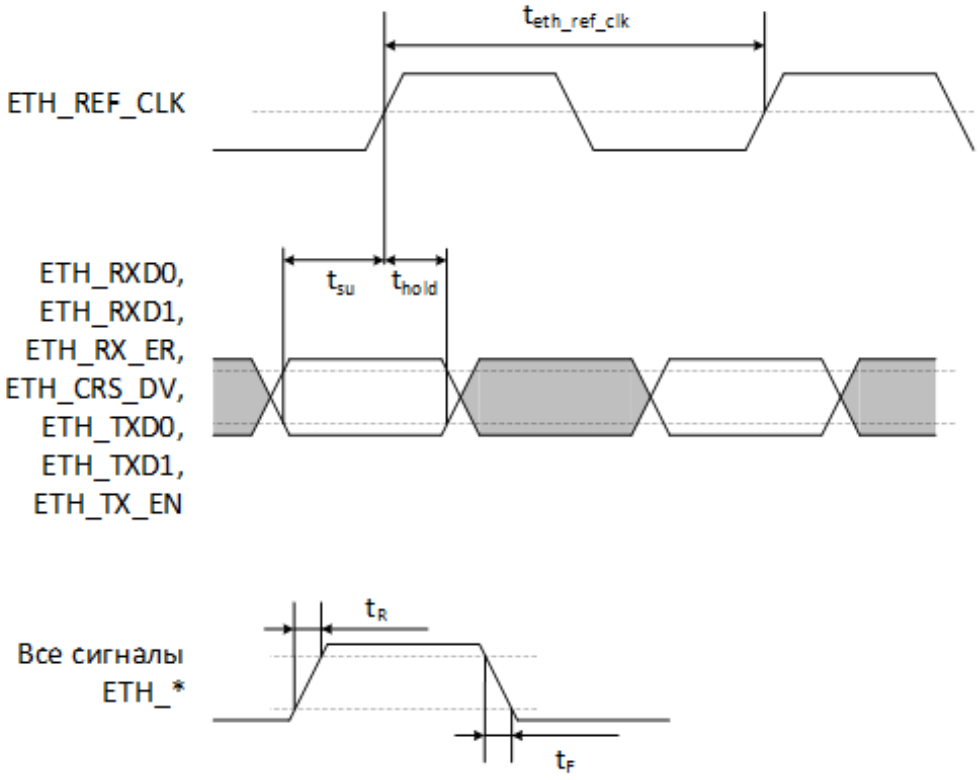


Рисунок 1.5 – Параметры сигналов интерфейса Ethernet RMII

На рисунках 1.6 и 1.7 приведены параметры сигналов интерфейса Ethernet MDIO.

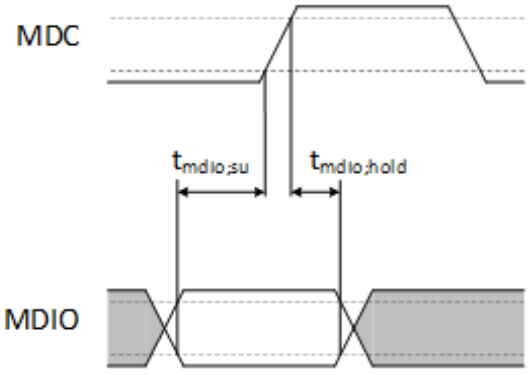


Рисунок 1.6 – Параметры сигналов интерфейса Ethernet MDIO при управлении линией MDIO микроконтроллером

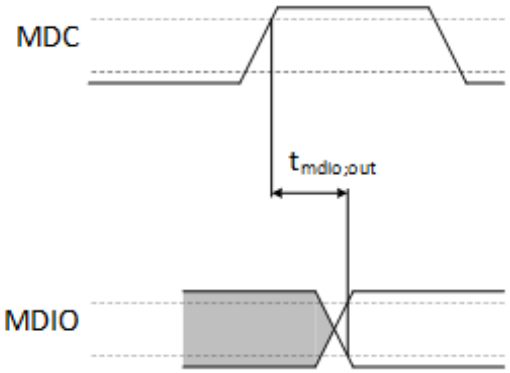


Рисунок 1.7 – Параметры сигналов интерфейса Ethernet MDIO при управлении линией MDIO внешним PHY

Таблица 1.8 – Значения параметров интерфейса Ethernet RMII

Наименование параметра	Буквенное обозначение	Значение параметра		
		не менее	номинал	не более
Частота опорного тактового сигнала ETH_REF_CLK	$t_{eth_ref_clk}$	–	20 нс	–
Скважность опорного тактового сигнала ETH_REF_CLK		35%	–	65%
Задержка от изменения сигналов ETH_RXD0, ETH_RXD1, ETH_RX_ER, ETH_CRS_DV, ETH_TXD0, ETH_TXD1, ETH_TX_EN до фронта ETH_REF_CLK	t_{su}	4 нс	–	–
Задержка от фронта ETH_REF_CLK до изменения сигналов ETH_RXD0, ETH_RXD1, ETH_RX_ER, ETH_CRS_DV, ETH_TXD0, ETH_TXD1, ETH_TX_EN	t_{hold}	2 нс	–	–
Время фронта любого сигнала RMII	t_R	–	–	5 нс
Время спада любого сигнала RMII	t_F	–	–	5 нс
Задержка от переключения сигнала MDIO до фронта сигнала MDC	$t_{mdio;su}$	10 нс	–	–
Задержка от фронта сигнала MDC до переключения сигнала MDIO, управляемого микроконтроллером	$t_{mdio;hold}$	10 нс	–	–
Задержка от фронта сигнала MDC до переключения сигнала MDIO, управляемого внешним PHY	$t_{mdio;out}$	0 нс	–	300 нс

1.2.5.5 Временные диаграммы и динамические параметры сигналов интерфейса I2C

На рисунке 1.8 приведены параметры сигналов интерфейсов I2C.

Значения параметров приведены в таблице 1.9.

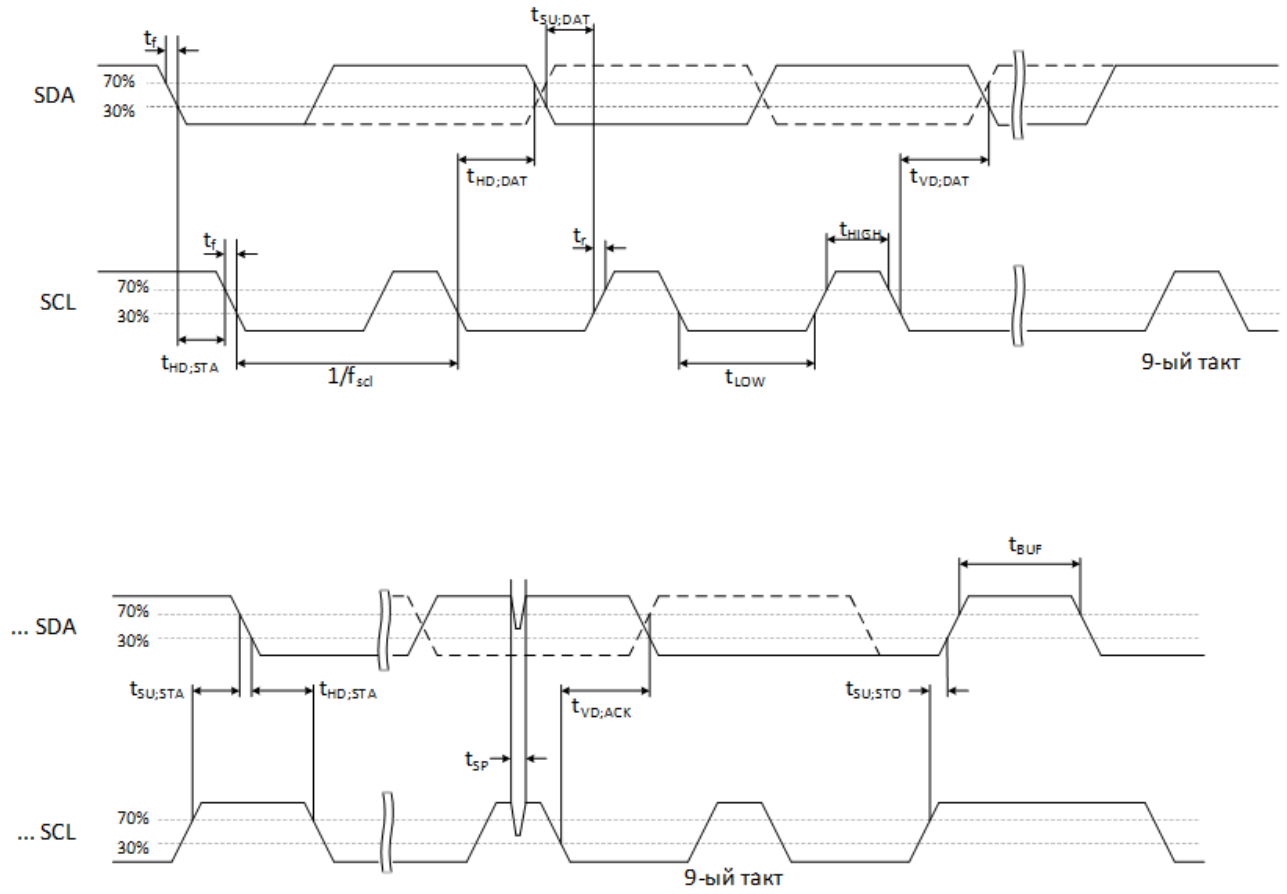


Рисунок 1.8 – Параметры сигналов интерфейса I2C

Таблица 1.9 – Значения параметров интерфейса I2C

Наименование параметра	Буквенное обозначение	Значение параметра			
		Стандартный режим		Скоростной режим	
		не менее	не более	не менее	не более
Частота сигнала синхронизации SCL	f_{scl}	0 кГц	100 кГц	0 кГц	400 кГц
Задержка от спада сигнала SDA до спада сигнала SCL при генерации СТАРТ условия (в том числе повторном)	$t_{HD;STA}$	4,0 мкс	–	0,6 мкс	–
Время низкого уровня сигнала SCL	t_{LOW}	4,7 мкс	–	1,3 мкс	–
Время высокого уровня сигнала SCL	t_{HIGH}	4,0 мкс	–	0,6 мкс	–
Задержка от фронта сигнала SCL до спада сигнала SDA при повторной генерации СТАРТ условия	$t_{SU;STA}$	4,7 мкс	–	0,6 мкс	–
Задержка от спада сигнала SCL до переключения сигнала SDA	$t_{HD;DAT}$	0,0 мкс	–	0,0 мкс	–
Задержка от переключения сигнала SDA до фронта сигнала SCL	$t_{SU;DAT}$	250 нс	–	100 нс	–
Время фронта сигналов SDA и SCL	t_r	–	1000 нс	20 нс	300 нс

Время спада сигналов SDA и SCL	t_f	–	300 нс	20 нс * VDD / 5,5 В	300 нс
Задержка от фронта сигнала SCL до фронта сигнала SDA при генерации СТОП условия	$t_{SU;STO}$	4,0 мкс	–	0,6 мкс	–
Время между условиями СТОП и СТАРТ, когда шина I2C свободна	t_{BUF}	4,7 мкс	–	1,3 мкс	–
Время от спада сигнала SCL до установки сигнала SDA в валидное значение при передаче бита данных	$t_{VD;DAT}$	–	3,45 мкс	–	0,9 мкс
Время от спада сигнала SCL до установки сигнала SDA в валидное значение при передаче бита подтверждения	$t_{VD;ACK}$	–	3,45 мкс	–	0,9 мкс
Ширина импульса помех, который должен быть подавлен входным фильтром контроллера	t_{SP}	–	–	0	50 нс

1.2.5.6 Временные диаграммы и динамические параметры сигналов интерфейса USART

На рисунках 1.9 и 1.10 приведены временные диаграммы сигналов интерфейса USART микросхемы К1879ВГ1Т.

Значения параметров временных диаграмм приведены в таблице 1.10.

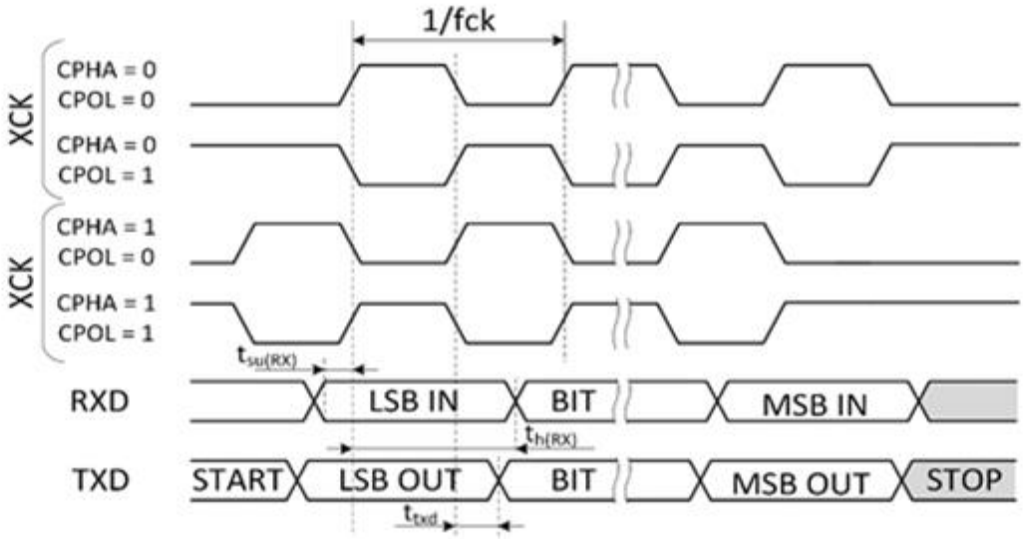


Рисунок 1.9 – Временные диаграммы сигналов интерфейса USART в режиме master

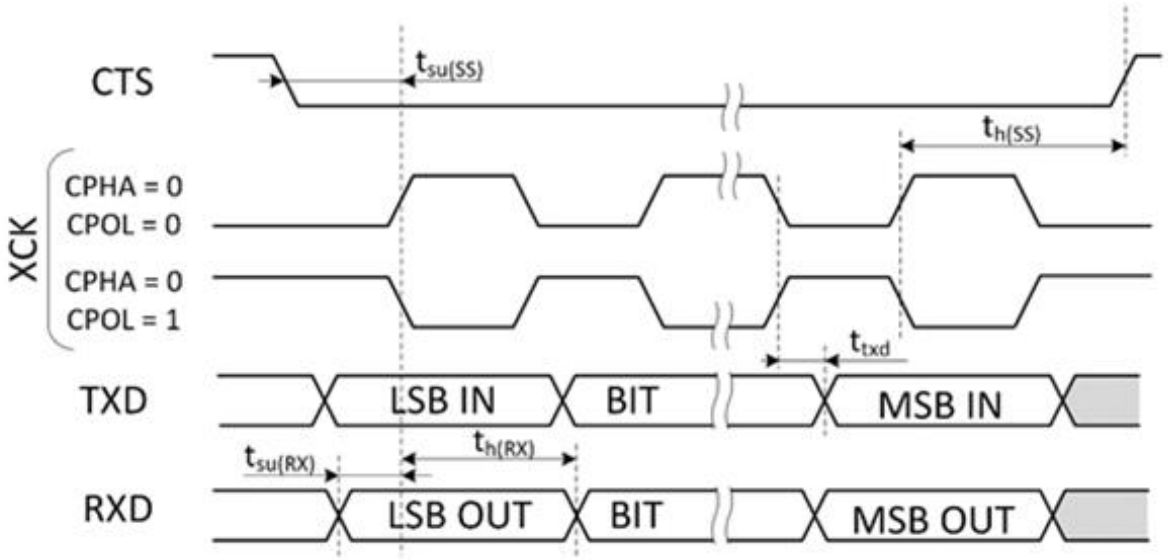


Рисунок 1.10 – Временные диаграммы сигналов интерфейса USART в режиме slave

Таблица 1.10 – Значения параметров временных диаграмм интерфейса USART

Наименование параметра	Буквенное обозначение	Значение параметра		
		не менее	номинал	не более
Выдаваемая или принимаемая частота опорного синхросигнала на XCK	f_{CK}	–	–	10 МГц
Время предустановки входных данных в master режиме	$t_{SU(RX)}$	14 нс	–	–
Время предустановки входных данных в slave режиме	$t_{SU(RX)}$	10 нс	–	–
Время удержания входных данных в master режиме	$t_{H(RX)}$	10 нс	–	–
Время удержания входных данных в slave режиме	$t_{H(RX)}$	10 нс	–	–
Задержка выходных данных относительно XCK в master режиме	t_{TXD}	2 нс	–	–
Задержка выходных данных относительно XCK в slave режиме	t_{TXD}	25 нс	–	–
Время предустановки сигнала SS в slave режиме	$t_{SU(SS)}$	Tker(1) + 2 нс	–	–
Время удержания сигнала SS в slave режиме	$t_{H(SS)}$	10 нс	–	–
1) Tker – период синхросигнала clk_usart USART контроллера после внутреннего делителя.				

1.2.5.7 Временные диаграммы и динамические параметры сигналов интерфейса SPI

На рисунках 1.11 и 1.12 приведены временные диаграммы обмена по интерфейсу SPI, входящему в состав микросхемы К1879ВГ1Т.

Значения параметров временной диаграммы приведены в таблице 1.11.

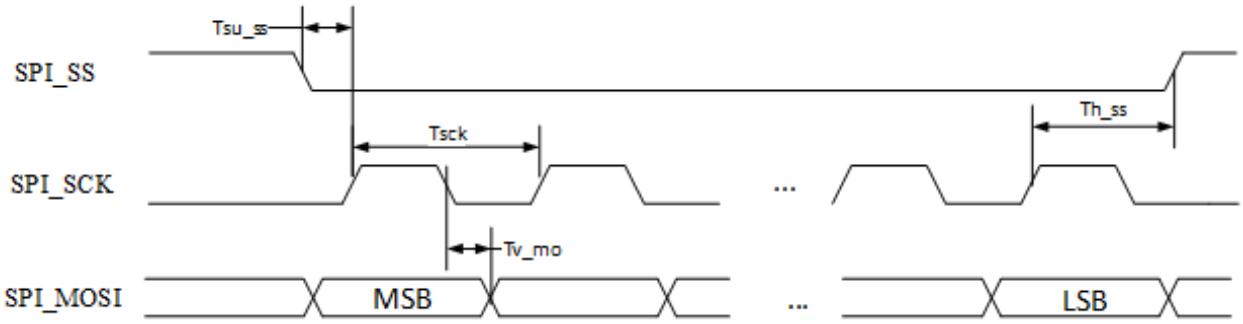


Рисунок 1.11 – Временная диаграмма выдачи данных по интерфейсу SPI

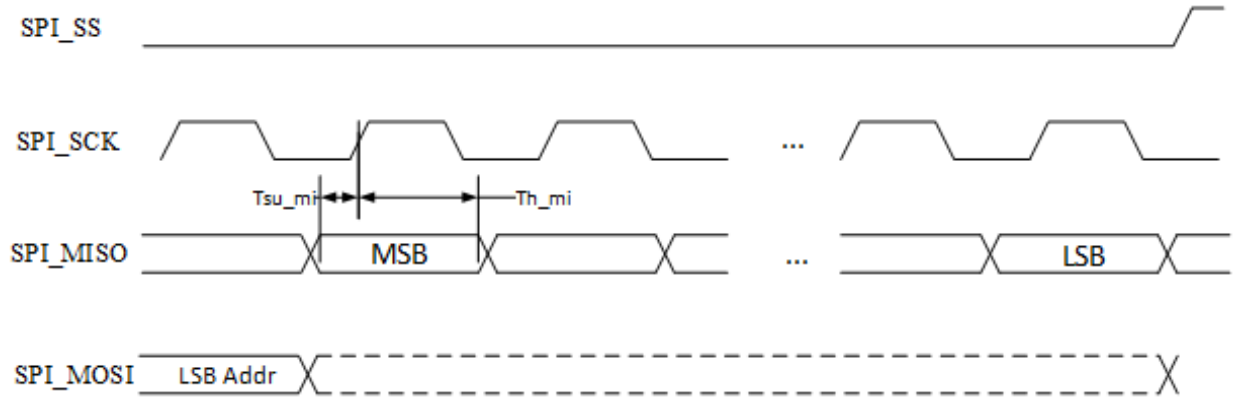


Рисунок 1.12 – Временная диаграмма приема данных по интерфейсу SPI

Таблица 1.11 – Значения параметров временной диаграммы

Наименование параметра	Буквенное обозначение	Значение параметра		
		не менее	номинал	не более
Время предустановки сигнала SPI_SS относительно фронта синхросигнала SPI_SCK	Tsu_ss	75 нс	–	-
Период синхросигнала SPI_SCK	Tsck	100 нс	–	-
Время установки сигнала SPI_MISO относительно синхросигнала SPI_SCK при при выдаче данных	Tv_mo	-	-	25нс
Время удержания сигнала SPI_SS относительно фронта синхросигнала SPI_SCK при завершении выдачи данных	Th_ss	75 нс	–	-
Время предустановки сигнала SPI_MISO относительно синхросигнала SPI_SCK при приеме данных	Tsu_mi	25 нс	–	–
Время удержания сигнала SPI_MISO относительно синхросигнала SPI_SCK при приеме данных	Th_mi	25 нс	–	–

1.3 Устройство и работа МК Optimal+

1.3.1 Структурная схема

Структурная схема МК Optimal+ представлена на рисунке 1.13.

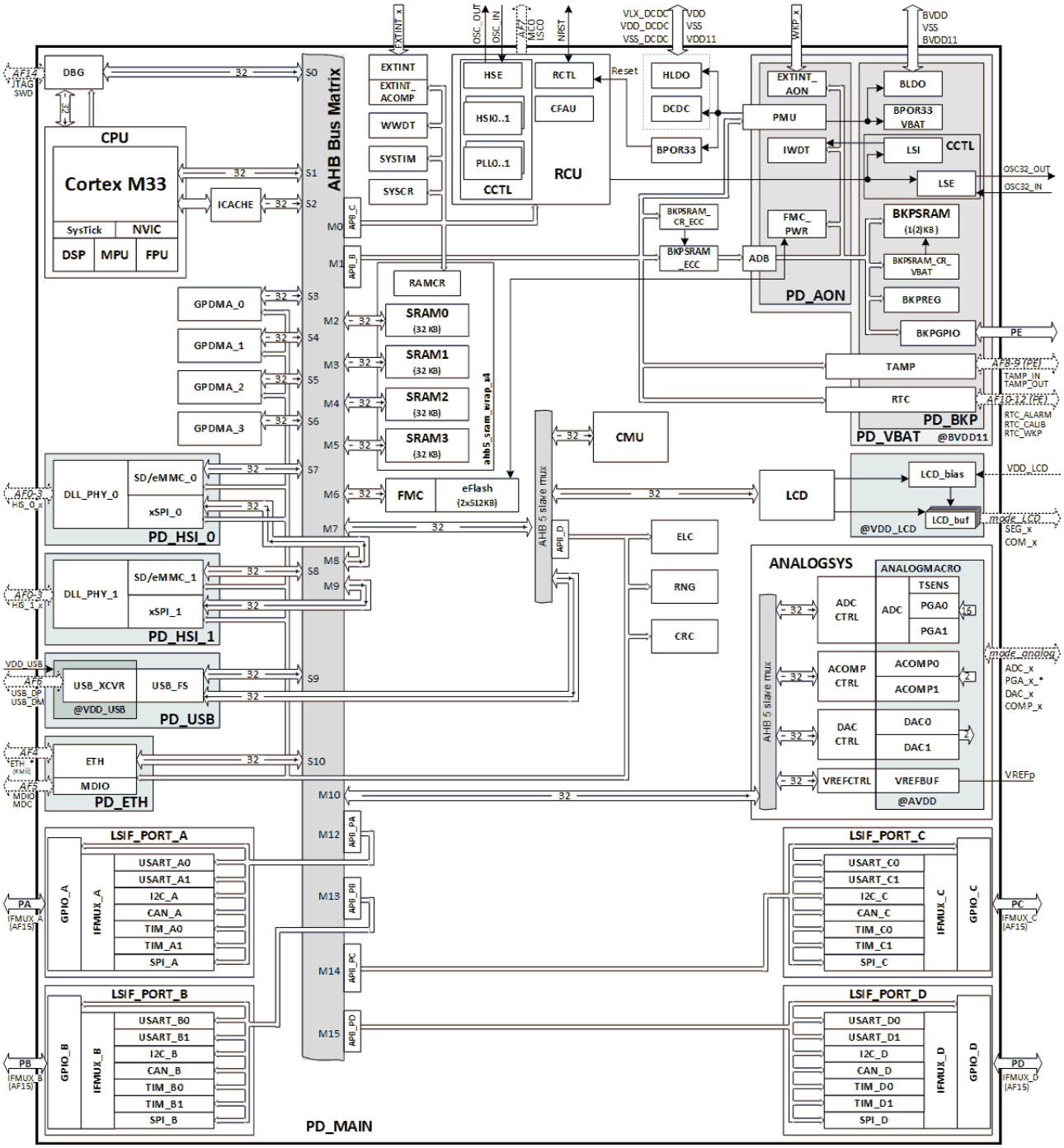


Рисунок 1.13 – Структурная схема МК Optimal+

МК Optimal+ включает следующие блоки:

– CPU - центральное 32-разрядное процессорное ядро ARM Cortex M33, включающее:

a) FPU - блок операций с плавающей точкой одинарной точности;

b) DSP расширение (ARMv8M DSP Extension);

c) Security расширение (ARMv8M Security Extension);

d) NVIC - встроенный контроллер прерываний;

e) MPU - блок защиты памяти;

– DBG - блок аппаратной отладки JTAG/SWD ;

– ICACHE - кэш-память команд процессорного ядра;

– AHB Bus Matrix - коммутатор системной шины AHB:

– AHB 5 Slave Mux - мультиплексор шины AHB для подключенния нескольких slave-интерфейсов;

– APB_B/C/D/PA/PB/PC/PD - блоки преобразования шины AHB в шину APB;

– SRAM0..3 - 4 банка встроенной ОЗУ;

– RAMCR - контроллера управления встроенной ОЗУ:

– FMC - контроллер встроенной eFlash-памяти;

– SYSCR - системный контроллер;

– RCU - подсистема управления сигналами сброса и синхронизации, включающая:

a) CCTL - – блок управления синхросигналами, включащий источники тактовых сигналов:

i) HSE — 4-48 МГц высокочастотный внешний генератор/резонатор;

ii) HSI0, HSI1 — 2 унифицированных 4-64 МГц высокочастотных внутренних RC-генератора;

iii) PLL0, PLL1 — 2 унифицированных блока ФАПЧ для получения высоких значений частот;

iv) LSE — 32,768 кГц низкочастотный внешний генератор/резонатор (в батарейном домене питания);

v) LSI — 32 кГц / 250 Гц низкочастотный внутренний RC-генератор (в батарейном домене питания);

b) RCTL - блок управления сигналами сброса;

c) блок контроля частоты синхросигналов (CFAU);

– набор источников и блоков контроля питания, включающий:

a) DCDC - блок импульсного преобразователя напряжения;

b) HLDO - блок линейного стабилизатора напряжения основного домена питания (PD_MAIN);

c) BPOR33 - схему сброса по питанию основного домена;

d) BLDO - блок линейного стабилизатора напряжения питания батарейного домена (PD_VBAT);

e) BPOR33_VBAT - схему сброса по питанию батарейного домена;

– домен батарейного питания (PD_VBAT), включающий:

a) PD_AON - always-on поддомен, включающий:

i) PMU - блок управления питанием;

ii) IWDT - независимый сторожевой таймер;

iii) EXTINT_AON - контроллер внешних событий для пробуждения из режима SHUTDOWN;

iv) FMC_PWR - контроллер управления режимами работы eFlash-памяти;

b) PD_BKP - поддомен резервного питания, включающий:

i) BKPSRAM - банк встроенной back-up памяти;

ii) BKPSRAM_CR_VBAT - контроллер состояния банка памяти BKPSRAM;

iii) BKPREG - блок back-up регистров;

iv) BKPGPIO - контроллер GPIO для буферов батарейного домена питания;

v) RTC - контроллер часов реального времени;

vi) TAMP - блок детектирования внешнего воздействия;

– дополнительные блоки для вхаимодействия с батарейным доменом питания:

a) ADB - блок, отвечающий за переход шины APB между доменами питания;

b) BKPSRAM_ECC - блок расчёта ECC для данных банка памяти BKPSRAM;

c) BKPSRAM_CR_ECC - контроллер блока BKPSRAM_ECC;

– набор таймеров, включающий:

a) SysTick - 24 разрядный системный таймер в составе CPU (1 шт.);

b) SYSTIM - 2-канальный (по 32 разряда) таймер общего назначения (1 шт.);

c) TIM - 4-канальный (по 16 разрядов) таймер (8 шт. в составе LSIF_PORT_x по 2 шт.);

d) WWDT - оконный сторожевой таймер;

e) IWDT - независимый сторожевой таймер (в PD_AON);

– LSIF_PORT_A/B/C/D - сборки контроллеров низкоскоростных интерфейсов, включающие контроллеры:

a) GPIO - контроллер буферов ввода/вывода, включающий:

i) IFMUX - блок мультиплексирования низкоскоростных интерфейсов;

b) USART - контроллер интерфейсов UART/USART (2 шт. в составе LSIF_PORT_x, всего 8 шт.);

c) I2C - контроллер интерфейса I2C (1 шт. в составе LSIF_PORT_x, всего 4 шт.);

d) CAN - контроллер интерфейсов CAN/CAN-FD (1 шт. в составе LSIF_PORT_x, всего 4 шт.);

e) TIM - 4-канальный таймер (2 шт. в составе LSIF_PORT_x, всего 8 шт.);

f) SPI - контроллер интерфейса SPI (1 шт. в составе LSIF_PORT_x, всего 4 шт.);

– LCD - контроллер LCD дисплея 40x8 или 44x4 (1 шт.);

– PD_ETH - контроллер интерфейсов Ethernet и MDIO в отключаемом домене питания, включающий:

a) ETH - контроллер интерфейса Ethernet 10/100 (с интерфейсом RMII) (1 шт.);

b) MDIO - контроллер интерфейса MDIO в составе контроллера ETH (1 шт.);

– PD_HSI_0/1 сборки высокоскоростных интерфейсов с внешней памятью в отключаемых доменах питания, включающие:

a) xSPI - контроллер интерфейсов xSPI/Octal/Quad/Dual SPI (1 шт. в составе PD_HSI_x, всего 2 шт.);

b) SD/eMMC - контроллер интерфейсов SD6.0/eMMC5.1 (1 шт. в составе PD_HSI_x, всего 2 шт.);

c) DLL_PHY - блок линий задержки для работы контроллеров SD/eMMC и xSPI на высоких скоростях (1 шт. в составе PD_HSI_x, всего 2 шт.);

– PD_USB - контроллер интерфейса USB в отключаемом домене питания, включающий:

a) USB_FS - контроллер интерфейса USB, поддерживающий скорости LS/FS, работающий в режиме host или device (1 шт.);

b) USB_XCVR - USB FS/LS трансивер (1 шт.);

– EXTINT - контроллер внешних событий;

– EXTINT_ACOMP - контроллер прерываний для аналоговых компараторов (ACOMP0, ACOMP1) в адресном пространстве EXTINT;

– ANALOGSYS - подсистема аналоговых блоков, включающая:

a) ANALOGMACRO - сборку аналоговых блоков:

i) ADC - 16-канальный 12-разрядный АЦП последовательного приближения с аналоговым мультиплексором на входе;

ii) TSENS - температурный датчик;

iii) PGA0/1 - операционные усилители (2 шт.);

iv) ACOMP0/1 - аналоговые компараторы (2 шт.);

v) DAC0/1 - цифро-аналоговые преобразователи (2 шт.);

vi) VREFBUF - встроенный источник опорного напряжения (1 шт.);

b) набор контроллеро аналоговых блоков:

i) ADC CTRL - цифровой контроллер ADC, TSENS и PGA;

ii) ACOMP CTRL - цифровой контроллер ACOMP0/1;

iii) DAC CTRL - цифровой контроллер DAC0/1;

iv) VREFBUF CTRL - цифровой контроллер VREFBUF;

– GPDMA_0/1/2/3 - 4 контроллера прямого доступа к памяти общего назначения;

– CMU - блок ускорения криптографических и математических операций;

– ELC - система прямого обмена событиями;

– CRC - блок расчёта CRC для произвольных полиномов;

– RNG - генератор случайных чисел.

1.3.2 Основные режимы работы

МК Optimal+ может работать в двух режимах:

– функциональный режим

– тестовый режим

Переключение микросхемы между режимами выполняется при помощи TAP-контроллера тестового режима (см.п.1.3.2.2).

1.3.2.1 Функциональный режим

Функциональный режим предназначен для выполнения МК Optimal+ функций по вводу, обработке и выводу информации. Выполнение данных функций производится под управлением пользовательского ПО, исполняемого на процессорном ядре Cortex-M33, с задействованием аппартных блоков интерфейсных контроллеров (например, USART, Ethernet и др.) и специализированных блоков ускорения приколадных функций (например,DMA, криптоускорители и др.). В рамках функционального режима предусмотрены несколько режимов энергопотребления, описание которых приведено в п. 1.3.8.2.

1.3.2.2 Тестовый режим

Для отладочных целей в микросхеме предусмотрен режим граничного сканирования Boundary Scan по стандарту 1149.1.

Данный режим предназначен для проведения граничного сканирования микросхемы и проверки целостности цифровых буферов ввода/вывода.

Из-за ограниченного числа выводов TAP-контроллер граничного сканирования (IR 5 бит) соединен в цепочку Daisy-Chain вместе с TAP-контроллером процессорного ядра (IR 4 бита), схема соединения приведена на рисунке 1.14.

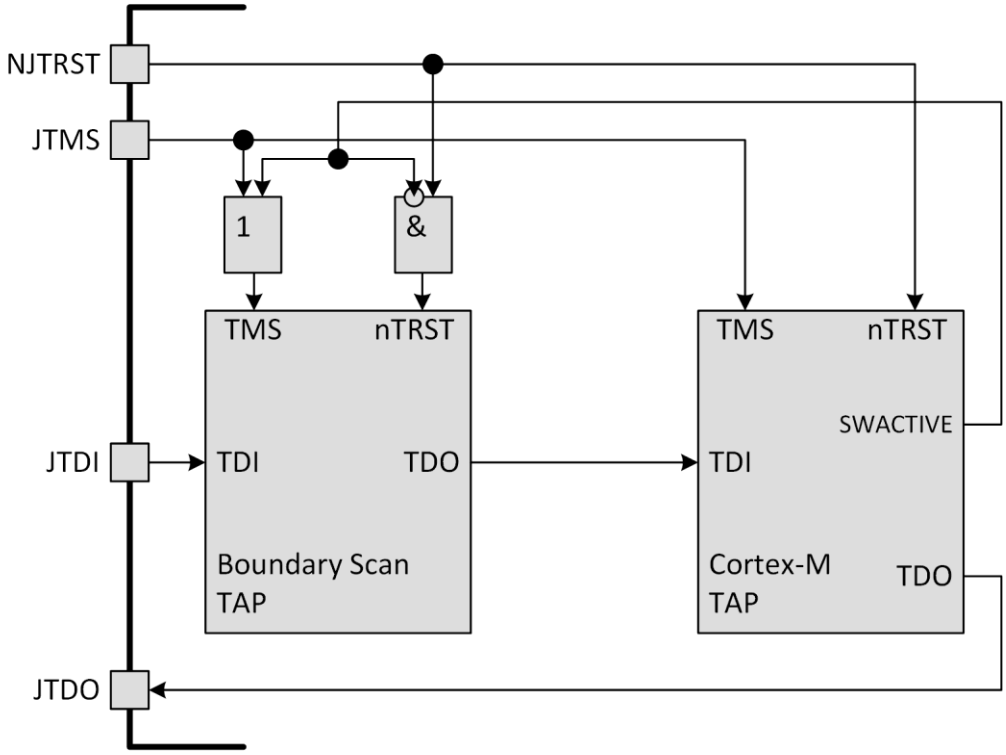


Рисунок 1.14 – Схема объединения TAP-контроллеров в цепочку

Чтобы получить доступ к режиму граничного сканирования, необходимо:

1) Установить активный сигнал сброса NRST = 0;

2) Установить инструкцию BYPASS (1111) в TAP-контроллер Cortex-M;

3) Для каждого IR-сдвига цепочка сканирования содержит 9 бит (=5+4) и в неиспользуемый TAP-контроллер должна задвигаться инструкция BYPASS;

4) При каждом сдвиге данных неиспользуемый TAP, находящийся в режиме BYPASS, добавляет 1 дополнительный бит данных в цепочку сканирования данных.

Чтобы получить доступ к TAP Cortex-M в целях отладки, необходимо установить инструкцию BYPASS (11111) в TAP-контроллер граничного сканирования.

Как только Serial-Wire будет выбран с использованием специальной последовательности Arm JTAG, TAP-контроллер граничного сканирования автоматически отключается (на JTMS принудительно устанавливается высокий уровень, и на NJTRST — низкий).

1.3.3 Карта памяти и шинный коммутатор

1.3.3.1 Блок шинного коммутатора

Архитектура микроконтроллера основана на ядре Arm Cortex-M33. Эта архитектура также включает в себя 32-разрядную многослойную матрицу коммутатора шин AHB5, которая соединяет 11 master-интерфейсов и 15 групп slave-интерфейсов.

– master-интерфейсы:

a) отладочная шина DBG для доступа к памяти и регистрам устройств. Шина DBG недоступна в нормальном режиме работы устройства;

b) шина команд ядра процессора C-Bus, которая подключается к коммутатору через кэш-память команд (ICACHE);

c) шина данных ядра процессора S-Bus;

d) 4 универсальных канала прямого доступа к памяти GPDMA0/GPDMA1;

e) 2 master-интерфейса контроллеров SD/eMMC;

f) master-интерфейс контроллера USB 2.0 Full Speed;

g) master-интерфейс контроллера Ethernet.

– slave-устройства:

a) внутренняя флэш-память FMC (1 Мбайт);

b) внутренняя память SRAM0 (32 Кбайт);

c) внутренняя память SRAM1 (32 Кбайт);

d) внутренняя память SRAM2 (32 Кбайт);

e) внутренняя память SRAM3 (32 Кбайт);

f) группа периферийных устройств с интерфейсом APB подключенных к порту APB_C;

g) группа периферийных устройств с интерфейсом APB подключенных к порту APB_B и резервная оперативная память (BKPSRAM);

h) группа периферийных устройств с интерфейсом AHB5 (контроллеры LCD, CMU и USB) и устройств с интерфейсом APB (подключенные к APB_D);

i) 2 контроллера внешней энергонезависимой памяти xSPI;

j) группа периферийных устройств с интерфейсом AHB5 (контроллеры ADCCTRL и DACCTRL);

k) 4 универсальных коммуникационных контроллера (подключенных к портам APB_PA, APB_PB, APB_PC, APB_PD).

Схема матричного коммутатора приведена на рисунке 1.15.

Детально каждый из портов блока описан далее.

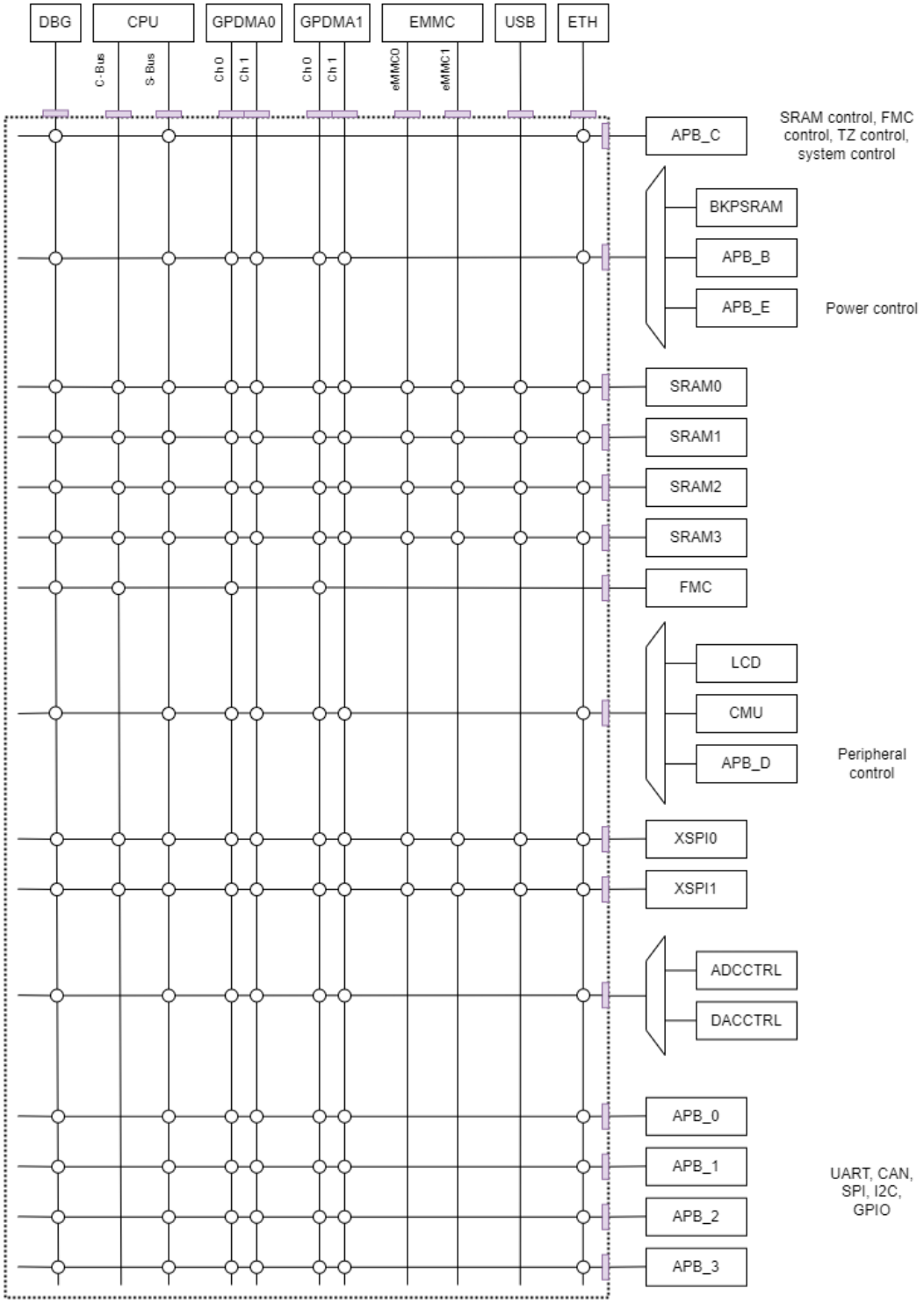


Рисунок 1.15 – Схема шинной коммутации

C-Bus

Это шина команд ядра Cortex-M33. Шина используется для выборки команд и доступа к данным во внутренней памяти, отображенным в области кода. Эта шина предназначена для доступа к внутренней флэш-памяти и банкам внутренней статической памяти (SRAM0,

SRAM1, SRAM2 и SRAM3). Банки SRAM0, SRAM1, SRAM2 и SRAM3 доступны по этой шине с непрерывным отображением. Эта шина также предназначена для чтения кода из внешней памяти через интерфейсы xSPI_0 и xSPI_1.

S-Bus

Это шина данных ядра Cortex-M33. Эта шина используется ядром для доступа к данным, расположенным на периферийном устройстве или в области SRAM. Эта шина предназначена для блоков внутренней статической памяти (SRAM0, SRAM1, SRAM2, SRAM3 и BKPSRAM), а также для периферийных устройств, включая периферийные устройства с интерфейсами APB и AHB5. SRAM0, SRAM1, SRAM2 и SRAM3 доступны на этой шине с непрерывным отображением.

Шины блоков GPDMA_0 и GPDMA_1

Эти шины соединяют контроллеры GPDMA с шинным коммутатором. Эти шины предназначены для доступа к внутренней флэш-памяти, блокам внутренней статической памяти (SRAM0, SRAM1, SRAM2, SRAM3 и BKPSRAM), периферийным устройствам на шинах AHB/APB и внешним запоминающим устройствам через интерфейс xSPI.

Контроллеры eMMC_0 и eMMC_1 с master-интерфейсами

Эти шины соединяют ПДП-интерфейсы eMMC_0 и eMMC_1 с матричным коммутатором. Эти шины используются eMMC_0 и eMMC_1 только для загрузки/сохранения данных из/в память. Эти шины предназначены для доступа к данным: внутренней флэш-памяти, внутренних банков SRAM (SRAM0, SRAM1, SRAM2 и SRAM3) и внешних запоминающих устройств через интерфейс xSPI.

Контроллер USB с master-интерфейсом

Эта шина соединяет ПДП-интерфейс контроллера USB с матричным коммутатором. Шина используются только для загрузки/сохранения данных из/в память. Эта шина предназначена для доступа к данным: внутренней флэш-памяти, внутренних банков SRAM (SRAM0, SRAM1, SRAM2 и SRAM3) и внешних запоминающих устройств через интерфейс xSPI.

Контроллер Ethernet с master-интерфейсом

Эта шина соединяет ПДП-интерфейс контроллера Ethernet с матричным коммутатором. Шина используются только для загрузки/сохранения данных из/в память. Эта шина предназначена для доступа к данным: внутренней флэш-памяти, внутренних банков SRAM (SRAM0, SRAM1, SRAM2 и SRAM3) и внешних запоминающих устройств через интерфейс xSPI.

Матричный коммутатор

Блок шинного матричного коммутатора (interconnect) обеспечивает коммутацию подключенных устройств согласно карте памяти системы. Блок осуществляет арбитраж между master-устройствами с использованием циклического алгоритма (round robin). Коммутатор имеет матричную структуру и каждый master-порт подключается к данному slave-порту без задержки. Задержка доступа (latency) от любого master-устройства до slave-устройств составляет как минимум 1 такт.

Мосты AHB/APB

Мосты AHB/APB обеспечивают полностью синхронные соединения между шинами AHB и APB, позволяя гибко выбирать периферийную частоту.

1.3.3.2 Карта памяти

В таблице 1.12 приведена карта памяти МК Optimal+.

Таблица 1.12 – Карта памяти МК Optimal+

M33 область памяти	**Область памяти**	**Начальный адрес**	**Конечный адрес**	**Размер (байт)**	
System Vendor_SYS	Зарезервировано	0xE010_0000	0xFFFF_FFFF	511 M	512 M
System Private Peripheral Bus	PPB	0xE000_0000	0xE00F_FFFF	1 M	
Ext.Device (S-AHB)	Зарезервировано	0xA000_0000	0xDFFF_FFFF	1 G	
Ext.RAM (S-AHB)	xSPI_1 NSC	0x9000_0000	0x9FFF_FFFF	256 M	1 G
	xSPI_1 NS	0x8000_0000	0x8FFF_FFFF	256 M	
	xSPI_0 NSC	0x7000_0000	0x7FFF_FFFF	256 M	
	xSPI_0 NS	0x6000_0000	0x6FFF_FFFF	256 M	
Peripheral (S-AHB)	Periph NSC	0x5000_0000	0x5FFFFFFF	256M	512M
	Periph NS	0x4000_0000	0x4FFFFFFF	256M	
SRAM (S-AHB)	Зарезервировано	0x3002_0000	0x3FFFFFFF		512M
	SRAM NSC	0x3000_0000	0x3001FFFF	128K	
	Зарезервировано	0x2002_0000	0x2FFFFFFF		
	SRAM NS	0x2000_0000	0x2001FFFF	128K	
Code (C-AHB)	Зарезервировано	0x1011_0000	0x1FFFFFFF		512M
	Code NS	0x1000_0000	0x1010FFFF	1M+64K	

M33 область памяти	Область памяти	Начальный адрес	Конечный адрес	Размер (байт)	
	Зарезервировано	0x0011_0000	0x0FFFFFFF		
	Code NS	0x0000_0000	0x0010FFFF	1M+64K	

По умолчанию запрещена выборка и выполнение команд из областей памяти Peripheral, Ext.Device, System PPB и System Vendor_SYS, эти области памяти имеют выставленный атрибут XN (eXecute-Never). Однако, как и все остальные атрибуты памяти, данный атрибут может быть переназначен с помощью MPU.

В таблице 1.13 приведена более детальная карта памяти области Peripheral.

Таблица 1.13 – Карта памяти Peripheral устройств

Порт	Начальный адрес NS	Конечный адрес NS	Начальный адрес S	Конечный адрес S	Размер	Описание
M0(APB_C)	0x4000_0000	0x4000_0FFF	0x5000_0000	0x5000_0FFF	4K	WWDT
	0x4000_1000	0x4000_1FFF	0x5000_1000	0x5000_1FFF	4K	EXTINT
	0x4000_2000	0x4000_2FFF	0x5000_2000	0x5000_2FFF	4K	RCU
	0x4000_3000	0x4000_3FFF	0x5000_3000	0x5000_3FFF	4K	Зарезервировано
	0x4000_4000	0x4000_4FFF	0x5000_4000	0x5000_4FFF	4K	SYSCR
	0x4000_5000	0x4000_5FFF	0x5000_5000	0x5000_5FFF	4K	Зарезервировано
	0x4000_6000	0x4000_6FFF	0x5000_6000	0x5000_6FFF	4K	SRAM Controller
	0x4000_7000	0x4000_7FFF	0x5000_7000	0x5000_7FFF	4K	Secure Debug Channel
	0x4000_8000	0x4000_8FFF	0x5000_8000	0x5000_8FFF	4K	Зарезервировано
	0x4000_9000	0x4000_9FFF	0x5000_9000	0x5000_9FFF	4K	Зарезервировано
	0x4000_A000	0x4000_AFFF	0x5000_A000	0x5000_AFFF	4K	Зарезервировано
	0x4000_B000	0x4000_BFFF	0x5000_B000	0x5000_BFFF	4K	Зарезервировано
	0x4000_C000	0x4000_CFFF	0x5000_C000	0x5000_CFFF	4K	Зарезервировано
	0x4000_D000	0x400F_FFFF	0x5000_D000	0x500F_FFFF		Зарезервировано
M1(APB_B)	0x4010_0000	0x4010_0FFF	0x5010_0000	0x5010_0FFF	4K	IWDT
	0x4010_1000	0x4010_1FFF	0x5010_1000	0x5010_1FFF	4K	Debug authctrl APB interface
	0x4010_2000	0x4010_2FFF	0x5010_2000	0x5010_2FFF	4K	EXTINT_AON
	0x4010_3000	0x4010_3FFF	0x5010_3000	0x5010_3FFF	4K	FMC Power controller

Порт	Начальный адрес NS	Конечный адрес NS	Начальный адрес S	Конечный адрес S	Размер	Описание
	0x4010_4000	0x4010_4FFF	0x5010_4000	0x5010_4FFF	4K	Зарезервировано
	0x4010_5000	0x4010_5FFF	0x5010_5000	0x5010_5FFF	4K	Зарезервировано
	0x4010_6000	0x4010_6FFF	0x5010_6000	0x5010_6FFF	4K	Зарезервировано
	0x4010_7000	0x4010_7FFF	0x5010_7000	0x5010_7FFF	4K	Зарезервировано
	0x4010_8000	0x4010_8FFF	0x5010_8000	0x5010_8FFF	4K	Backup GPIO
	0x4010_9000	0x4010_9FFF	0x5010_9000	0x5010_9FFF	4K	Backup register
	0x4010_A000	0x4010_AFFF	0x5010_A000	0x5010_AFFF	4K	BKPSRAM
	0x4010_B000	0x4010_BFFF	0x5010_B000	0x5010_BFFF	4K	BKPRSRAM_CR_VBAT
	0x4010_C000	0x4010_CFFF	0x5010_C000	0x5010_CFFF	4K	Зарезервировано
	0x4010_D000	0x4010_DFFF	0x5010_D000	0x5010_DFFF	4K	Зарезервировано
	0x4010_E000	0x4010_EFFF	0x5010_E000	0x5010_EFFF	4K	Зарезервировано
	0x4010_F000	0x4010_FFFF	0x5010_F000	0x5010_FFFF	4K	Зарезервировано
	0x4011_0000	0x4011_0FFF	0x5011_0000	0x5011_0FFF	4K	BKPRSRAM_CR_ECC
	0x4011_1000	0x4011_1FFF	0x5011_1000	0x5011_1FFF	4K	TAMP
	0x4011_2000	0x4011_2FFF	0x5011_2000	0x5011_2FFF	4K	RTC
	0x4011_3000	0x4011_3FFF	0x5011_3000	0x5011_3FFF	4K	PMU
-	0x4011_4000	0x403F_FFFF	0x5011_4000	0x503F_FFFF		Зарезервировано
M7(LCD)	0x4040_0000	0x4040_FFFF	0x5040_0000	0x5040_FFFF	64K	LCD
M7(CMU)	0x4041_0000	0x4041_FFFF	0x5041_0000	0x5041_FFFF	64K	CMU
M7(USB_OTG)	0x4042_0000	0x4042_FFFF	0x5042_0000	0x5042_FFFF	64K	USB_FS
M7(APB_D)	0x4043_0000	0x4043_0FFF	0x5043_0000	0x5043_0FFF	4K	RNG
	0x4043_1000	0x4043_1FFF	0x5043_1000	0x5043_1FFF	4K	ELC
	0x4043_2000	0x4043_2FFF	0x5043_2000	0x5043_2FFF	4K	uDMA_0 control
	0x4043_3000	0x4043_3FFF	0x5043_3000	0x5043_3FFF	4K	uDMA_1 control
	0x4043_4000	0x4043_5FFF	0x5043_4000	0x5043_5FFF	8K	eMMC controller and soft PHY
	0x4043_6000	0x4043_7FFF	0x5043_6000	0x5043_7FFF	8K	eMMC controller and soft PHY
	0x4043_8000	0x4043_BFFF	0x5043_8000	0x5043_BFFF	16K	xSPI controller
	0x4043_C000	0x4043_FFFF	0x5043_C000	0x5043_FFFF	16K	xSPI controller

Порт	Начальный адрес NS	Конечный адрес NS	Начальный адрес S	Конечный адрес S	Размер	Описание
	0x4044_0000	0x4047_FFFF	0x5044_0000	0x5047_FFFF	256K	Ethernet 10/100M controller
	0x4048_0000	0x4048_0FFF	0x5048_0000	0x5048_0FFF	4K	uDMA_2 control
	0x4048_1000	0x4048_1FFF	0x5048_1000	0x5048_1FFF	4K	uDMA_3 control
	0x4048_2000	0x4048_2FFF	0x5048_2000	0x5048_2FFF	4K	FMC
	0x4048_3000	0x4048_3FFF	0x5048_3000	0x5048_3FFF	4K	CRC
-	0x4048_4000	0x4071_FFFF	0x5048_4000	0x5071_FFFF		Зарезервировано
M10(APB_A0)	0x4072_0000	0x4072_0FFF	0x5072_0000	0x5072_0FFF	4K	ACOMPCTRL
-	0x4072_1000	0x4072_FFFF	0x5072_1000	0x5072_FFFF		Зарезервировано
M10(APB_A1)	0x4073_0000	0x4073_FFFF	0x5073_0000	0x5073_FFFF	4K	VREFBUF
-	0x4074_0000	0x409F_FFFF	0x5074_0000	0x509F_FFFF		Зарезервировано
M12(APB_PA)	0x40A0_0000	0x40A0_0FFF	0x50A0_0000	0x50A0_0FFF	4K	Контроллер GPIO
	0x40A0_1000	0x40A0_1FFF	0x50A0_1000	0x50A0_1FFF	4K	Контроллер USART 0
	0x40A0_2000	0x40A0_2FFF	0x50A0_2000	0x50A0_2FFF	4K	Контроллер USART 1
	0x40A0_3000	0x40A0_3FFF	0x50A0_3000	0x50A0_3FFF	4K	Контроллер I2C
	0x40A0_4000	0x40A0_4FFF	0x50A0_4000	0x50A0_4FFF	4K	Контроллер CAN
	0x40A0_5000	0x40A0_5FFF	0x50A0_5000	0x50A0_5FFF	4K	Контроллер TIM 0
	0x40A0_6000	0x40A0_6FFF	0x50A0_6000	0x50A0_6FFF	4K	Контроллер TIM 1
	0x40A0_7000	0x40A0_7FFF	0x50A0_7000	0x50A0_7FFF	4K	Контроллер SPI
_	0x40A0_8000	0x40AF_FFFF	0x50A0_8000	0x50AF_FFFF		Зарезервировано
M13(APB_PB)	0x40B0_0000	0x40B0_0FFF	0x50B0_0000	0x50B0_0FFF	4K	Контроллер GPIO
	0x40B0_1000	0x40B0_1FFF	0x50B0_1000	0x50B0_1FFF	4K	Контроллер USART 0
	0x40B0_2000	0x40B0_2FFF	0x50B0_2000	0x50B0_2FFF	4K	Контроллер USART 1
	0x40B0_3000	0x40B0_3FFF	0x50B0_3000	0x50B0_3FFF	4K	Контроллер I2C
	0x40B0_4000	0x40B0_4FFF	0x50B0_4000	0x50B0_4FFF	4K	Контроллер CAN

Порт	Начальный адрес NS	Конечный адрес NS	Начальный адрес S	Конечный адрес S	Размер	Описание
	0x40B0_5000	0x40B0_5FFF	0x50B0_5000	0x50B0_5FFF	4K	Контроллер TIM 0
	0x40B0_6000	0x40B0_6FFF	0x50B0_6000	0x50B0_6FFF	4K	Контроллер TIM 1
	0x40B0_7000	0x40B0_7FFF	0x50B0_7000	0x50B0_7FFF	4K	Контроллер SPI
_	0x40B0_8000	0x40BF_FFFF	0x50B0_8000	0x50BF_FFFF		Зарезервировано
M14(APB_PC)	0x40C0_0000	0x40C0_0FFF	0x50C0_0000	0x50C0_0FFF	4K	Контроллер GPIO
	0x40C0_1000	0x40C0_1FFF	0x50C0_1000	0x50C0_1FFF	4K	Контроллер USART 0
	0x40C0_2000	0x40C0_2FFF	0x50C0_2000	0x50C0_2FFF	4K	Контроллер USART 1
	0x40C0_3000	0x40C0_3FFF	0x50C0_3000	0x50C0_3FFF	4K	Контроллер I2C
	0x40C0_4000	0x40C0_4FFF	0x50C0_4000	0x50C0_4FFF	4K	Контроллер CAN
	0x40C0_5000	0x40C0_5FFF	0x50C0_5000	0x50C0_5FFF	4K	Контроллер TIM 0
	0x40C0_6000	0x40C0_6FFF	0x50C0_6000	0x50C0_6FFF	4K	Контроллер TIM 1
	0x40C0_7000	0x40C0_7FFF	0x50C0_7000	0x50C0_7FFF	4K	Контроллер SPI
_	0x40C0_8000	0x40CF_FFFF	0x50C0_8000	0x50CF_FFFF		Зарезервировано
M15(APB_PD)	0x40D0_0000	0x40D0_0FFF	0x50D0_0000	0x50D0_0FFF	4K	Контроллер GPIO
	0x40D0_1000	0x40D0_1FFF	0x50D0_1000	0x50D0_1FFF	4K	Контроллер USART 0
	0x40D0_2000	0x40D0_2FFF	0x50D0_2000	0x50D0_2FFF	4K	Контроллер USART 1
	0x40D0_3000	0x40D0_3FFF	0x50D0_3000	0x50D0_3FFF	4K	Контроллер I2C
	0x40D0_4000	0x40D0_4FFF	0x50D0_4000	0x50D0_4FFF	4K	Контроллер CAN
	0x40D0_5000	0x40D0_5FFF	0x50D0_5000	0x50D0_5FFF	4K	Контроллер TIM 0
	0x40D0_6000	0x40D0_6FFF	0x50D0_6000	0x50D0_6FFF	4K	Контроллер TIM 1
	0x40D0_7000	0x40D0_7FFF	0x50D0_7000	0x50D0_7FFF	4K	Контроллер SPI
_	0x40D0_8000	0x4FFF_FFFF	0x50D0_8000	0x5FFF_FFFF		Зарезервировано

В таблице 1.14 приведена карта памяти периферийных устройств Private Peripheral Bus (PPB) процессорного ядра Cortex-M33. Запросы с адресами 0xE000_0000-0xE004_3FFF

обслуживаются периферийными устройствами внутри процессорного ядра (ITM, DWT, BPU и SCS) и не выходят наружу. Запросы по адресам 0xE004_4000-0xE00F_FFFF выходят наружу ядра через шину EPPB (external PPB) стандарта APB, это адресное пространство предназначено для подключения внешних CoreSight Debug компонентов, например, ROM-таблицы.

Таблица 1.14 – Карта памяти периферийных устройств Private Peripheral Bus (PPB) ядра Cortex-M33

Начальный адрес	**Конечный адрес**	**Размер (байт)**	**Устройство**
0xE00F_F000	0xE00F_FFFF	4K	CoreSight ROM table
0xE004_C000	0xE00F_EFFF	716K	Зарезервировано
0xE004_B000	0xE004_BFFF	4K	SDC Ext
0xE004_A000	0xE004_AFFF	4K	TS Gen
0xE004_8000	0xE004_9FFF	8K	AHB-AP
0xE004_6000	0xE004_7FFF	8K	AHB-AP Core
0xE004_4000	0xE004_5FFF	8K	AHB-AP Sys
0xE004_3000	0xE004_3FFF	4K	Зарезервировано
0xE004_2000	0xE004_2FFF	4K	CrossTrigger Interface (CTI)
0xE004_1000	0xE004_1FFF	4K	ETM
0xE004_0000	0xE004_0FFF	4K	Trace Port Interface Unit (TPIU)
0xE002_F000	0xE003_FFFF	68K	Зарезервировано
0xE002_E000	0xE002_EFFF	4K	System Control Space (SCS) (Non-Secure alias)
0xE000_F000	0xE001_FFFF	124K	Зарезервировано
0xE000_E000	0xE000_EFFF	4K	System Control Space (SCS) (Secure)
0xE000_3000	0xE000_DFFF	44K	Зарезервировано
0xE000_2000	0xE000_2FFF	4K	Breakpoint Unit (BPU)
0xE000_1000	0xE000_1FFF	4K	Data Watchpoint and Trace (DWT)
0xE000_0000	0xE000_0FFF	4K	Instrumentation Macrocell (ITM)

В таблице 1.15 приведена карта памяти области System Control Space (SCS) процессорного ядра Cortex-M33.

Таблица 1.15 – Карта памяти области System Control Space (SCS) (Secure) ядра Cortex-M33

Начальный адрес	**Используемый объем (байт)**	**Устройство**
0xE000_EFB0	80	Debug Identification Block
0xE000_EF34	24	Floating-Point Extension (FPU)
0xE000_EF00	4	Software Interrupt Generation (STIR reg)
0xE000_EDF0	272	Debug Control Block
0xE000_EDD0	32	Security Attribution Unit (SAU)
0xE000_ED90	64	Memory Protection Unit (MPU)

Начальный адрес	Используемый объем (байт)	Устройство
0xE000_ECFC	148	System Control Block (SCB)
0xE000_E100	<1280	Nested Vectored Interrupt Controller
0xE000_E010	16	SysTick Timer
0xE000_E004	12	Implementation Control Block (ICTR, ACTLR, CPPWR regs)

1.3.4 Подсистема центрального процессора

1.3.4.1 Процессорное ядро

МК Optimal+ основан на процессорном ядре ARM Cortex-M33. В реализацию данного ядра в микроконтроллере включена поддержка FPU- и DSP- расширений, а также ядро поддерживает расширения безопасности (TrustZone и ARMv8M Security Extension).

В таблице 1.16 приведены выбранные при реализации МК Optimal+ конфигурационные опции процессорного ядра.

Таблица 1.16 – Опции реализации ядра Cortex-M33 в МК Optimal+

Опция	Значение
Поддержка FPU	FPU с одинарной точностью
DSP расширение (ARMv8M DSP Extension) (поддержка дополнительных арифметических команд, команд SIMD обработки и команд работы с битовыми полями)	да
Security расширение (ARMv8M Security Extension)	да
Non-secure protected memory regions	8
Secure protected memory regions	8
Security Attribution Unit (SAU)	8
Number of bits of interrupt priority	4
Debug watchpoints and breakpoints	4 watchpoint and 8 breakpoint comparators
ITM and Data Watchpoint and Trace (DWT) trace functionality	да
Embedded Trace Macrocell (ETM)	да
Micro Trace Buffer (MTB)	да
Cross Trigger Interface (CTI)	да
Wake-up Interrupt Controller (WIC)	да
External coprocessor interface	нет
Arm Custom Instructions (ACIs)	нет

На рисунке 1.16 приведена структурная схема процессорного ядра.

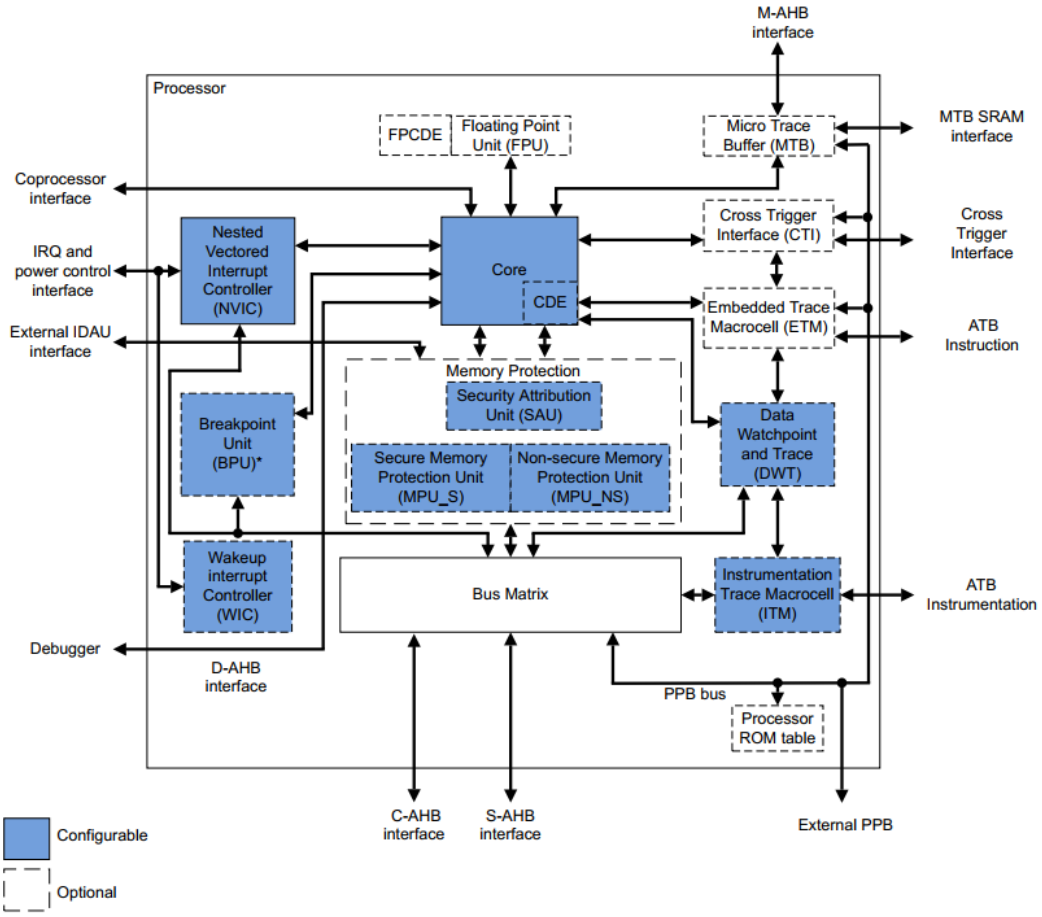


Рисунок 1.16 – Структурная схема процессорного ядра Cortex-M33

1.3.4.2 Кэш-память команд (ICACHE)

В МК Optimal+ имеется L1 кэш инструкций, подключенный к C-AHB интерфейсу процессорного ядра.

Особенности

– 2-канальный множественно-ассоциативный кэш

– 16-битные кэш-линии

– размер - 8 КБ

– поддержка bypass режима

Кэш поддерживает разделение ресурсов на Secure и Non-secure, сохраняя исходные аттрибуты кэшируемых данных. Однако, при изменении настроек SAU и IDAU, данные в кэше не меняют свои аттрибуты. Для поддержания безопасности, перед перенастройкой блоков SAU и IDAU необходимо выключить и инвалидировать кэш инструкций.

Кэшируются только запросы на чтение. Транзакции записи(если они есть) проходят кэш насквозь. При операции записи данных, которые уже присутствуют в кэше, необходимо отключить и инвалидировать кэш.

Таблица 1.17 – Опции реализации L1 кэш инструкций

Опция	**Значение**	**Описание**
Размер	8KB	_
INVMAT	0	Invalidate on Write Match, при кэшируемой записи в линию, присуствующую в ICACHE, инвалидирует эту линию
COFFSIZE	3	Размер кэшируемой области. Адресуемое пространство - 2^(32-COFFSIZE) (Нужно объяснение смысла)
STATS	1	Включает функциональность сбора статистики
XOM	1	Поддержка Execute Only Memory
REDUCE_READS	1	? (Нужно объяснение смысла)

1.3.4.2.1 Программная модель

Таблица 1.18 – Регистры L1 кэш

Сдвиг	**Название**	**Доступ**	**Значение после сброса**	**Полное название**
0x000	ICHWPARAMS	RO	Зависит от конфигурации	Hardware Parameter register
0x004	ICCTRL	RW	0x04	Instruction Cache Control Register
0x008-0x0FC	-	-	-	Зарезервировано.
0x100	ICIRQSTAT	RO	0x0	Interrupt Request Status register
0x104	ICIRQSCLR	WO	0x0	Interrupt Status Clear register
0x108	ICIRQEN	RW	0x0	Interrupt Enable register
0x10C	ICDBGFILLERR	RO	0x0	Debug Fill Error Register
0x200-0x2FC	-	-	-	Зарезервировано.
0x300	ICSH	RO	0x0	Instruction Cache Statistic Hit register
0x304	ICSM	RO	0x0	Instruction Cache Statistic Miss Count register
0x308	ICSUC	RO	0x0	Instruction Cache Statistic Uncached Count register
0x30C-0xFCC	-	-	-	Зарезервировано.
0xFD0	PIDR4	RO	0x04	Product ID Register 4.

Сдвиг	Название	Доступ	Значение после сброса	Полное название
0xFD4	PIDR5	RO	0x0	Product ID Register 5.
0xFD8	PIDR6	RO	0x0	Product ID Register 6.
0xFDC	PIDR7	RO	0x0	Product ID Register 7.
0xFE0	PIDR0	RO	0x57	Product ID Register 0.
0xFE4	PIDR1	RO	0xB8	Product ID Register 1.
0xFE8	PIDR2	RO	0x01B	Product ID Register 2.
0xFEC	PIDR3	RO	0x00	Product ID Register 3.
0xFF0	CIDR0	RO	0x0D	Component ID Register 0.
0xFF4	CIDR1	RO	0xF0	Component ID Register 1.
0xFF8	CIDR2	RO	0x05	Component ID Register 2.
0xFFC	CIDR3	RO	0xB1	Component ID Register 3.

1.3.4.2.1.1 Регистр ICHWPARAMS

Доступен только на чтение, описывает конфигурацию ICACHE

Таблица 1.19 – Регистр ICHWPARAMS

Биты	Название	Доступ	Значение после сброса	Описание
[31:16]	COFFSET	RO	0x0	Cacheable Offset Address. Задает старшие [31:16] биты кэшируемого региона памяти
[15:12]	COFFSIZE	RO	0x3	Размер кэшируемой области. Задает размер кэшируемой области например: 0x0: 4GB 0x1: 2GB 0x2: 1GB 0x3: 512MB
[11:7]	Зарезервировано	RO	-	Зарезервировано.
[6]	INVMAT	RO	Parameterized	Указывает, включена ли опция инвалидации кэш линии при кэш-попадании для транзакйии записи
[5]	DMA	RO	0x0	Наличие DMA Engine: 0 = Кэш инструкций не поддерживает prefetch и locking.

Биты	Название	Доступ	Значение после сброса	Описание
[4]	STATS	RO	0x1	Наличие функциональности сбора статистики
[3:0]	CSIZE	RO	13	Размер кэша: 9: 512 byte 10: 1 KB 11: 2 KB 12: 4 KB 13: 8 KB 14: 16 KB Другие значения зарезервированы.

1.3.4.2.1.2 Регистр ICCTRL

Регистр упраления кэшем

Таблица 1.20 – Регистр ICCTRL

Биты	Название	Доступ	Значение после сброса	Описание
[31:7]	Зарезервировано	RO	0x0	-
[6]	POINV_SMP	RO	0x1	Семплированное значение порта POINV, означает наличие функциональности Power-on-invalidate
[5]	HALLOC	RW	0x0	При записи 1, запрос кода обработчика прерываний рассматривается как любой другой запрос кода При записи 0, при кэш-промахе для запроса кода обработчика прерывания, не происходит добавления этого кода в кэш. Это делает запросы кода обработчиков более предсказыемыми и предотвращает забивания кэша, когда присутствует много обработчиков
[4]	STATC	WO	0x0	Сброс статистики. Запись 1 сбрасывает данные о статистике запросов в кэш
[3]	STATEN	RW	0x0	Включение сбора статистики
[2]	FINV	WO	0x0	Полная инвалидация кэша
[1]	Зарезервировано	RO	0x0	Зарезервировано.
[0]	CACHEEN	RW	0x0	Включение кэша. Запись 1 включает кэш, при записи 0 все запросы проходят насквозь

1.3.4.2.1.3 Регистры ICIRQSTAT, ICIRQSCLR и ICIRQEN

Регистры работы с прерываниями ICACHE

– ICIRQSTAT(RO) - содержит статусы прерываний ICACHE

– ICIRQSCLR(WO) - запись 1 в соответсвующий бит сбрасывает прерывание

– ICIRQEN(RW) - запись 0 маскирует соответсвующее прерывание

Все 3 регистра имеют одинаковую структуру, показанную ниже

Таблица 1.21 – Регистр ICIRQSTAT, ICIRQSCLR, and ICIRQEN

Биты	Название	Значение после сброса	Описание
[31:6]	-	0x0	Зарезервировано.
[5]	SS	0	Прерывание насыщения статистики. Указывает на то, что внутренние счетчики статистикы насытились
[4]	SV	0	Стутус прерывания нарушения Security.
[3]	CFE	0	Прерывание ошибки заполнения кэша. Указывает, что произошла ошибка на шине при попытке заполнить кэш
[2]	CEC	0	Прерывание завершения включения кэша
[1]	CDC	0	Прерывание завершения выключения кэша
[0]	IC	0	Прерывание завершения инвалидации. Означает, что инвалидация кэша завершена

1.3.4.2.1.4 Регистр ICDBGFILLERR

Регистр позволяет программному обеспечению определить последний адрес, который привел к ошибке на шине

Таблица 1.22 – Регистр ICDBGFILLERR

Биты	Название	Доступ	Значение после сброса	Описание
[31:0]	ERRADDR	RO	0x0	Адрес последнего запроса, приведшего к ошибке на шине при попытке заполнения кэша

1.3.4.2.1.5 Регистры ICSHR, ICSM и ICSUC

Регистры статистика кэша

Таблица 1.23 – Регистр ICSHR

Биты	Название	Доступ	Значение после сброса	Описание
[31:0]	CSHR	RO	0x0	Колличество кэш-попаданий

Таблица 1.24 – Регистр ICSM

Биты	Название	Доступ	Значение после сброса	Описание
[31:0]	CSM	RO	0x0	Колличество кэш-промахов

Таблица 1.25 – Регистр ICSUC

Биты	Название	Доступ	Значение после сброса	Описание
[31:0]	CSUC	RO	0x0	Колличество некэшируемых запросов

1.3.4.2.1.6 Регистры CIDRx, PIDRx

ARM-специфичные регистры идентификации

Таблица 1.26 – Регистры CIDRx, PIDRx

Биты	Название	Доступ	Значение после сброса	Описание
PIDR0	Part number	RO	0x57	Product ID Register 0.
PIDR1	Part number	RO	0xB8	Product ID Register 1.
PIDR2	Part revision	RO	0x01B	Product ID Register 2.
PIDR3	Revision number	RO	0x00	Product ID Register 3.
PIDR4	Зарезервировано	RO	0x04	Product ID Register 4.
PIDR5	Зарезервировано	RO	0x00	Product ID Register 5.
PIDR6	Зарезервировано	RO	0x00	Product ID Register 6.
PIDR7	Зарезервировано	RO	0x00	Product ID Register 7.
CIDR0	Preamble	RO	0x0D	Component ID Register 0.
CIDR1	Preamble	RO	0xF0	Component ID Register 1.
CIDR2	Preamble	RO	0x05	Component ID Register 2.
CIDR3	Preamble	RO	0xB1	Component ID Register 3.

1.3.4.2.2 Программирование L1 CACHE

Ниже представлены рекомендации по работе и управлении L1 CACHE

1.3.4.2.2.1 Инициализация

После подачи сброса и питания, кэш начинает работу в выключенном состоянии и приступает к процессу инвалидации. Все запросы в это время проходят его насквозь.

Включение кэша осуществляется записью 1 в бит CACHEEN. Однако, до завершения процесса инвалидации, все запросы все еще проходят кэш насквозь.

По завершению данного процесса, выставляется статус IC_STATUS, а когда соответсвующее прерывание включено, выставляется и прерывание. При необходимости, можно установить момент завершения инвалидации до появления прерывания, считывая значение регистра ICIRQSTAT.

1.3.4.2.2.2 Отключение

Отключение кэша происходит после записи 1 в бит CACHEEN. Все невыполненные запросы сначала будут завершены, затем происходит выключение кэша. Для определения факта выключения кэша можно использовать прерывание CDC либо статус бит CDC_STATUS.

1.3.4.2.2.3 Инвалидация

Инвалидация происходит после записи 1 в бит FINV. После этого происходит полная инвалидация кэша. То, как кэш себя ведет во время инвалидации, описано выше(см "Инициализация")

1.3.4.2.3 Поддержание когерентности между кэшем и памятью

L1 CACHE не имеет возможности поддерживать когерентность между памятью, хранящей код и соответствующей кэш-линией, содержащей этот код.

При модификации памяти, программное обеспечение должно полностью инвалидировать кэш и дождаться соответствующего прерывания. Отсуствие когерентности между Secure кэш-линий и ее измененной версии в памяти является проблемой с точки зрения безопасности. Для поддержания когерентности необходимо выполнять следущие шаги:

– Отключить кэш и дождаться соответствующего прерывания

– Полностью инвалидировать кэш, дождаться соответствующего прерывания

– Изменить необходимый участок в памяти

– Заново включить кэш, дождаться соответствующего прерывания

Аналогичные шаги стоит предпринимать, когда меняется не сам код, но его Security атрибуты.

1.3.4.3 Контроллер прерываний (NVIC)

Процессор содержит в себе контроллер внутренних прерываний NVIC, который поддерживает:

– 1-480 прерываний.

– Программируемый уровень приоритета от 0 до 255. Более высокий уровень соответствует меньшему приоритету, поэтому уровень 0 — это наивысший приоритет прерывания. В реализации с расширением безопасности в небезопасном состоянии приоритет также зависит от значения AIRCR.PRIS;

– Уровневое и импульсное значение сигналов прерываний.

– Упрощенную оследовательную обработку прерываний (Tail-Chaining).

– Внешнее немаскируемое прерывание (NMI).

– An optional Wake-up Interrupt Controller (WIC).

– Поздние поступающие прерывания(Late arriving interrupts).

1.3.4.3.1 Регистры

Регистры контроллера внутренних прерываний NVIC представлены в таблицах 1.27 - 1.36.

Таблица 1.27 – Регистры контроллера NVIC

Address	Name	Type	Required privilege	Reset value	Description
0xE000E004	ICTR	RO	Configurable	0x0000000x	Interrupt Controller Type Register
0xE000E100-0xE000E13C	NVIC_ISER0-NVIC_ISER15	RW	Privileged	0x00000000	Interrupt Set Enable Registers.
0XE000E180-0xE000E1BC	NVIC_ICER0-NVIC_ICER15	RW	Privileged	0x00000000	Interrupt Clear Enable Registers.
0XE000E200-0xE000E23C	NVIC_ISPR0-NVIC_ISPR15	RW	Privileged	0x00000000	Interrupt Set Pending Registers.
0XE000E280-0xE000E2BC	NVIC_ICPR0-NVIC_ICPR15	RW	Privileged	0x00000000	Interrupt Clear Pending Registers
0xE000E300-0xE000E33C	NVIC_IABR0-NVIC_IABR15	RW	Privileged	0x00000000	Interrupt Active Bit Registers.
0xE000E380-0xE000E3BC	NVIC_ITNS0-NVIC_ITNS15	RW	Privileged	0x00000000	Interrupt Target Non-secure Registers on page 4-315.
0xE000E400-0xE000E5DC	NVIC_IPR0-NVIC_IPR119	RW	Privileged	0x00000000	Interrupt Priority Registers.
0xE000EF00	STIR	WO	Configurable	0x00000000	

Поскольку в МК Optimal+ поддерживаются Secure и Non-secure режимы :

– Доступ к регистрам может быть RAZ/WI (read as zero / write ingored) в зависимости от значения NVIC_ITNS.

– Данные регистры не дублируются для Secure/NS состояний.

1.3.4.3.1.1 Регистр Interrupt Controller Type Register

Таблица 1.28 – Регистр NVIC_ICTR

Биты	Название	Описание
[31:4]	_	Зарезервировано
[3:0]	INTLINESNUM	Общее колличество прерываний (группами по 32): 0b0000 - 1..32 0b0001 - 33..64

После снятия сигнала сброса принимает уже верное значение.

1.3.4.3.1.2 Регистр Interrupt Set Enable

Таблица 1.29 – Регистры NVIC_ISER0-NVIC_ISER15

Биты	Название	Описание
[31:0]	SETENA	Включение прерываний 32n + m для SETENA[m] в регистре NVIC_ISERn

Запись:

– 0 Нет эффекта.

– 1 Включает прерывание 32n+m.

Чтение:

– 0 Прерывание 32n+m выключено.

– 1 Прерывание 32n+m включено.

1.3.4.3.1.3 Регистр Interrupt Clear Enable

Таблица 1.30 – Регистры NVIC_ICER0-NVIC_ICER15

Биты	Название	Описание
[31:0]	CLRENA	Выключение прерываний 32n + m для CLRENA[m] в регистре NVIC_ICERn

Запись:

– 0 Нет эффекта.

– 1 Выключает прерывание 32n+m.

Чтение: - 0 Прерывание 32n+m выключено. - 1 Прерывание 32n+m включено.

1.3.4.3.1.4 Регистр Interrupt Set Pending

Таблица 1.31 – Регистры NVIC_ISPR0-NVIC_ISPR15

Биты	Название	Описание
[31:0]	SETPEND	Установка прерываний 32n + m для SETPEND[m] в регистре NVIC_ISPRn

Запись:

– 0 Нет эффекта.

– 1 Устанавливает прерывание 32n+m.

Чтение:

– 0 Прерывание 32n+m не было выставлено.

– 1 Прерывание 32n+m выставлено.

Запись 1 в уже ожидающее прерывание не имеет эффекта, а в выключенное - устанавливает его.

1.3.4.3.1.5 Регистр Interrupt Clear Pending

Таблица 1.32 – Регистры NVIC_ICPR0-NVIC_ICPR15

Биты	Название	Описание
[31:0]	CLRPEND	Снимает прерывания 32n + m для CLRPEND[m] в регистре NVIC_ICPRn

Запись:

– 0 Нет эффекта.

– 1 Снимает прерывание 32n+m.

Чтение:

– 0 Прерывание 32n+m не было выставлено.

– 1 Прерывание 32n+m выставлено.

Запись 1 в уже активное прерывание не имеет эффекта, а в выключенное - устанавливает его.

1.3.4.3.1.6 Регистр Interrupt Active Bit

Таблица 1.33 – Регистры NVIC_IABR0-NVIC_IABR15

Биты	Название	Описание
[31:0]	ACTIVE	Отображает, активны ли прерывания 32n + m для ACTIVE[m] в регистре NVIC_IABRn

1.3.4.3.1.7 Регистр Interrupt Target Non-secure

Таблица 1.34 – Регистры NVIC_ITNS0-NVIC_ITNS15

Биты	Название	Описание
[31:0]	ITNS	Отображает, является ли Non-secure прерывание 32n + m для ITNS[m] в регистре NVIC_ITNSn

1.3.4.3.1.8 Регистр Interrupt Priority

Таблица 1.35 – Регистры NVIC_IPR0-NVIC_IPR119

Биты	Название	Описание
[31:24]	Приоритет, сдвиг 3	Содержит приоритет прерывания номер (4n+3) для регистра NVIC_IPRn
[23:16]	Приоритет, сдвиг 2	Содержит приоритет прерывания номер (4n+2) для регистра NVIC_IPRn
[15:8]	Приоритет, сдвиг 1	Содержит приоритет прерывания номер (4n+1) для регистра NVIC_IPRn
[7:0]	Приоритет, сдвиг 0	Содержит приоритет прерывания номер (4n+0) для регистра NVIC_IPRn

Так же, значение приоритета зависит от значения бита AIRCR.PRIS, который устанавливает Secure прерываниям более высокий приоритет, чем Non-secure. Доступ к регистрам может быть RAZ/WI (read as zero / write ingored) в зависимости от значения NVIC_ITNS. Данные регистры не дублируются для Secure/NS состояний.

1.3.4.3.1.9 Регистр Software Trigger Interrupt

Таблица 1.36 – Регистры NVIC_STIR

Биты	Название	Описание
[31:9]	_	Зарезервировано

Биты	Название	Описание
[8:0]	INTID	Номер выставляемого прерывания. Запись 0x3 приведет к выставлению IRQ3.

Если значение бита USERSETMPEND в регистре CCR установлено в 1, то непривилегированное ПО имеет доступ к данному регистру. Только привилегированное ПО может установить данный бит.

1.3.4.3.2 Уровневые и импульсные прерывания

Процессор поддерживает как уровневые, так и импульсные прерывания. Уровневое прерывание удерживается до тех пор, пока периферийное устройство не сбросит сигнал прерывания. Обычно это происходит из-за того, что процедура обработки прерывания обращается к периферии, вызывая очистку запроса на прерывание. Импульсное прерывание — это сигнал прерывания, который считывается синхронно на нарастающем фронте тактового сигнала процессора. Чтобы NVIC обнаружил прерывание, периферийное устройство должно подать сигнал прерывания хотя бы на один такт, в течение которого NVIC обнаружит импульс и зафиксирует прерывание. Когда процессор входит в процедуру обработки прерывания, он автоматически снимает состояние ожидания с прерывания. В случае уровневого прерывания, если сигнал не сброшен до возвращения процессора из процедуры обработки прерывания, прерывание снова становится ожидающим, и процессору необходимо выполнить свою процедуру обработки прерывания заново. Это означает, что периферийное устройство может удерживать сигнал прерывания установленным до тех пор, пока оно больше не будет нуждаться в обслуживании.

1.3.4.3.2.1 Аппаратно-программное управление прерываниями

Процессор фиксирует все прерывания. Периферийное прерывание становится ожидающим по одной из следующих причин:

– NVIC обнаруживает, что сигнал прерывания активен, а соответствующее прерывание неактивно;

– NVIC обнаруживает нарастающий фронт сигнала прерывания;

– ПО записывает бит соответствующего регистра разрешения установки прерывания.

Ожидающее прерывание остаётся ожидающим до одного из следующих событий:

– процессор входит в процедуру обработки прерывания для прерывания, изменяя состояние прерывания с ожидающего на активное;

– для уровневого прерывания, когда процессор возвращается из процедуры обработки прерывания, NVIC опрашивает сигнал прерывания. Если сигнал установлен, состояние

прерывания изменяется на ожидающее, что может привести к немедленному повторному входу процессора в процедуру обработки прерывания. В противном случае состояние прерывания изменяется на неактивное;

– для импульсного прерывания NVIC продолжает отслеживать сигнал прерывания, и если он пульсирует, состояние прерывания изменяется на ожидающее и активное. В этом случае, когда процессор возвращается из процедуры обработки прерывания, состояние прерывания изменяется на ожидающее, что может привести к немедленному повторному входу процессора в процедуру обработки прерывания. Если сигнал прерывания не пульсировал во время нахождения процессора в процедуре обработки прерывания, при его возврате состояние прерывания изменяется на неактивное;

– программное обеспечение записывает бит соответствующего регистра очистки ожидающих прерываний. Для уровневого прерывания, если сигнал прерывания всё ещё установлен, состояние прерывания не изменяется. В противном случае состояние прерывания изменяется на неактивное. Для импульсного прерывания состояние прерывания изменяется на:

– неактивное, если состояние было ожидающим;

– активное, если состояние было активным и ожидающим.

1.3.4.3.2.2 Советы по работе с NVIC

Убедитесь, что программное обеспечение использует правильно выровненные доступы к регистрам. Процессор не поддерживает невыровненные доступы к регистрам NVIC.

Прерывание может перейти в ожидающее состояние даже если оно отключено. Отключение прерывания только предотвращает обработку процессором этого прерывания.

Перед программированием VTOR для перемещения векторной таблицы убедитесь, что записи векторной таблицы новой векторной таблицы настроены для обработчиков ошибок, NMI и всех разрешённых исключений, таких как прерывания.

1.3.4.3.3 Карта прерываний

Карта прерываний представлена в таблице 1.37.

Таблица 1.37 – Карта прерываний

Exception number	IRQ number	Source	Name	Description	Vector offset
	-	Arm	_	Initial stack pointer	0x4

Exception number	IRQ number	Source	Name	Description	Vector offset
1	-	Arm	Reset	Initial program counter (reset vector)	0x8
2	-	Arm	NMI	Non-Maskable interrupt (NMI) - HSEFAIL, LSEFAIL events in RCU	0xC
3	-	Arm	HardFault	Hard Fault	0x10
4	-	Arm	MemManage	MemManage fault	0x14
5	-	Arm	BusFault	BusFault	0x18
6	-	Arm	UsageFault	UsageFault	0x1C
7	-	Arm	SecureFault	SecureFault	0x20
8	-	Arm	_	Reserved	0x24
9	-	Arm	_	Reserved	0x28
10	-	Arm	_	Reserved	0x2C
11	-	Arm	SVC	Supervisor Call (SVCall)	0x30
12	-	Arm	Debug monitor	DebugMonitor	0x34
13	-	Arm	_	Reserved	0x38
14	-	Arm	PendSV	Pendable request for system service (PendableSrvReq)	0x3C
15	-	Arm	SYSTICK	System Tick Timer (SysTick)	0x40
16		RNG	rng_irq	Random Number Generator interrupt	0x44
17	1	GPDMA_0	gpdma_0_irq	DMA 0 interrupt	0x48
18	2	GPDMA_1	gpdma_1_irq	DMA 1 interrupt	0x4C
19	3	FMC	fmc_irq	Flash Memory Controller interrupt	0x50
20	4	eMMC_0	emmc_0_irq	eMMC_0 controller interrupt	0x54
21	5	eMMC_1	emmc_1_irq	eMMC_1 controller interrupt	0x58
22	6	USB_FS	usb_fs_irq	USB controller interrrupt	0x5C
23	7	xSPI_0	xspi_0_irq	xSPI_0 controller interrupt	0x60
24	8	xSPI_1	xspi_1_irq	xSPI_1 controller interrupt	0x64
25	9	ETH	eth_irq	Ethernet controller interrupt	0x68
26	10	WWDT	wwdt_irq	Window Watch Dog Timer interrupt	0x6C

Exception number	IRQ number	Source	Name	Description	Vector offset
27	11	PMU	pmu_irq	Power Management Unit interrupt	0x70
28	12	RCU	rcu_irq	Reset and Clock control Unit interrupt	0x74
29	13	CFAU	cfau_irq	Clock Frequency Analysis Unit interrupt	0x78
30	14	SYSCR	syscr_irq	System Controller interrupt	0x7C
31	15	SYSTIM	systim_irq_0	System Timer 0 interrupt	0x80
32	16	SYSTIM	systim_irq_1	System Timer 1 interrupt	0x84
33	17	EXTINT	extint_0_irq	External interrupt 0	0x88
34	18		extint_1_irq	External interrupt 1	0x8C
35	19		extint_2_irq	External interrupt 2	0x90
36	20		extint_3_irq	External interrupt 3	0x94
37	21		extint_4_irq	External interrupt 4	0x98
38	22		extint_5_irq	External interrupt 5	0x9C
39	23		extint_6_irq	External interrupt 6	0xA0
40	24		extint_7_irq	External interrupt 7	0xA4
41	25		extint_8_irq	External interrupt 8	0xA8
42	26		extint_9_irq	External interrupt 9	0xAC
43	27		extint_10_irq	External interrupt 10	0xB0
44	28		extint_11_irq	External interrupt 11	0xB4
45	29		extint_12_irq	External interrupt 12	0xB8
46	30		extint_13_irq	External interrupt 13	0xBC
47	31		extint_14_irq	External interrupt 14	0xC0
48	32		extint_15_irq	External interrupt 15	0xC4

Exception number	IRQ number	Source	Name	Description	Vector offset
49	33	RAMCR	ramcr_irq	RAM Controller interrupt	0xC8
50	34	IWDT	iwdt_irq	Independent Watch Dog Timer interrupt	0xCC
51	35	TAMP	tamp_irq	Tamper Controller interrupt	0xD0
52	36	RTC	rtc_irq	Real Time Counter interrupt	0xD4
53	37	LSIF_PORT_A	usart_0_a_irq	LSIF_A USART_0 interrupt	0xD8
54	38		usart_1_a_irq	LSIF_A USART_1 interrupt	0xDC
55	39		i2c_a_irq	LSIF_A I2C interrupt	0xE0
56	40		can_a_irq	LSIF_A CAN interrupt	0xE4
57	41		tim_0_a_irq	LSIF_A TIM_0 interrupt	0xE8
58	42		tim_1_a_irq	LSIF_A TIM_1 interrupt	0xEC
59	43		spi_a_irq	LSIF_A SPI interrupt	0xF0
60	44	LSIF_PORT_B	usart_0_b_irq	LSIF_B USART_0 interrupt	0xF4
61	45		usart_1_b_irq	LSIF_B USART_1 interrupt	0xF8
62	46		i2c_b_irq	LSIF_B I2C interrupt	0xFC
63	47		can_b_irq	LSIF_B CAN interrupt	0x100
64	48		tim_0_b_irq	LSIF_B TIM_0 interrupt	0x104
65	49		tim_1_b_irq	LSIF_B TIM_1 interrupt	0x108
66	50		spi_b_irq	LSIF_B SPI interrupt	0x10C
67	51	LSIF_PORT_C	usart_0_c_irq	LSIF_C USART_0 interrupt	0x110
68	52		usart_1_c_irq	LSIF_C USART_1 interrupt	0x114
69	53		i2c_c_irq	LSIF_C I2C interrupt	0x118
70	54		can_c_irq	LSIF_C CAN interrupt	0x11C

Exception number	IRQ number	Source	Name	Description	Vector offset
71	55		tim_0_c_irq	LSIF_C TIM_0 interrupt	0x120
72	56		tim_1_c_irq	LSIF_C TIM_1 interrupt	0x124
73	57		spi_c_irq	LSIF_C SPI interrupt	0x128
74	58	LSIF_PORT_D	usart_0_d_irq	LSIF_D USART_0 interrupt	0x12C
75	59		usart_1_d_irq	LSIF_D USART_1 interrupt	0x130
76	60		i2c_d_irq	LSIF_D I2C interrupt	0x134
77	61		can_d_irq	LSIF_D CAN interrupt	0x138
78	62		tim_0_d_irq	LSIF_D TIM_0 interrupt	0x13C
79	63		tim_1_d_irq	LSIF_D TIM_1 interrupt	0x140
80	64		spi_d_irq	LSIF_D SPI interrupt	0x144
81	65	ADC	adc_irq_0	Analog-to-Digital Converter interrupt 0	0x148
82	66	DAC	dac_irq_0	Digital-to-Analog Converter interrupt	0x14C
83	67	ACOMP	acomp_irq_0	Maskable interrupt from Acomp 0	0x150
84	68	ACOMP	acomp_irq_1	Maskable interrupt from Acomp 1	0x154
85	69	ACOMP	acomp_irq_2	acomp_irq_0 AND acomp_irq_1	0x158
86	70	CPU	cpu_cti_irq[0]	CPU CTI interrupt 0	0x15C
87	71	CPU	cpu_cti_irq[1]	CPU CTI interrupt 1	0x160
88	72	DBG	dbg_se-cure_dbg_ch_irq	Secure Debug Channel interrupt	0x164
89	73	GPDMA_2	gpdma_2_irq	DMA 2 interrupt	0x168
90	74	GPDMA_3	gpdma_3_irq	DMA 3 interrupt	0x16C
91	75	GPDMA_0	gpdma_0_done	DMA 0 Done	0x170
92	76	GPDMA_1	gpdma_1_done	DMA 1 Done	0x174
93	77	GPDMA_2	gpdma_2_done	DMA 2 Done	0x178
94	78	GPDMA_3	gpdma_3_done	DMA 3 Done	0x17C

Exception number	IRQ number	Source	Name	Description	Vector offset
95	79	BKPSRAM_CR	bkpsram_cr_irq	BKPSRAM Control Interrupt	0x180
96	80	ACOMP	comparator_irq_0	Raw Acomp 0 irq	0x184
97	81	ACOMP	comparator_irq_1	Raw Acomp 1 irq	0x188
98	82	LCD	lcd_irq	LCD irq	0x18C
99	83	CRC	crc_irq	CRC irq	0x190

1.3.5 Подсистема памяти

1.3.5.1 Контроллер встроенной флэш-памяти (FMC)

1.3.5.1.1 Характеристики контроллера FMC

Контроллер FMC в составе МК Optimal+ реализует доступ к встроенному ПЗУ типа Flash-память (embedded Flash, далее eFlash-память).

Контроллер FMC обладает следующими характеристиками:

– два банка памяти с возможностью их перестановки (bank swapping);

– суммарный объем доступной флэш памяти 1Мбайт + 64Кбайт:

– 512 Кбайт основной области памяти в каждом из двух банков;

– 32 Кбайт информационной области памяти в каждом из двух банков;

– размер страницы для любой области памяти составляет 8 Кбайт;

– чтение данных из eFlash-памяти словами по 128 бит;

– запись данных одиночными словами или блоками до 8 слов;

– внутренний буфер на 4 ранее зачитанных слова;

– поддержка чтения из одного банка во время записи/стирания другого;

– поддержка проверки и генерации кодов коррекции ошибок при операциях чтении и записи (ECC):

– обнаружение и исправление одиночной ошибки (SEC);

– обнаружение двойной ошибки (DED);

– поддержка режимов пониженного потребления независимо для каждого банка;

– поддержка постраничных режимов безопасности (ARM TrustZone);

– количество циклов перезаписи:

– не менее 10 000 циклов перезаписи для любой области eFlash-памяти;

– не менее 100 000 циклов перезаписи для младших 256 Кбайт основной области каждого банка;

– защита от записи, чтения и стирания:

– постраничная;

– побанковая;

– автоматическая защита банков памяти при нарушении доступа;

– автоматическое чтение после выхода из состояния сброса Option Memory (OptMem);

– формирование прерываний по завершению длительных операций (запись, стирание):

– формирование прерываний по нарушению безопасности;

– системный интерфейс AHB 5;

– интерфейс для программирования APB 4.

1.3.5.1.2 Устройство и функционирование блока

Структурная схема контроллера FMC представлена на рисунке 1.17.

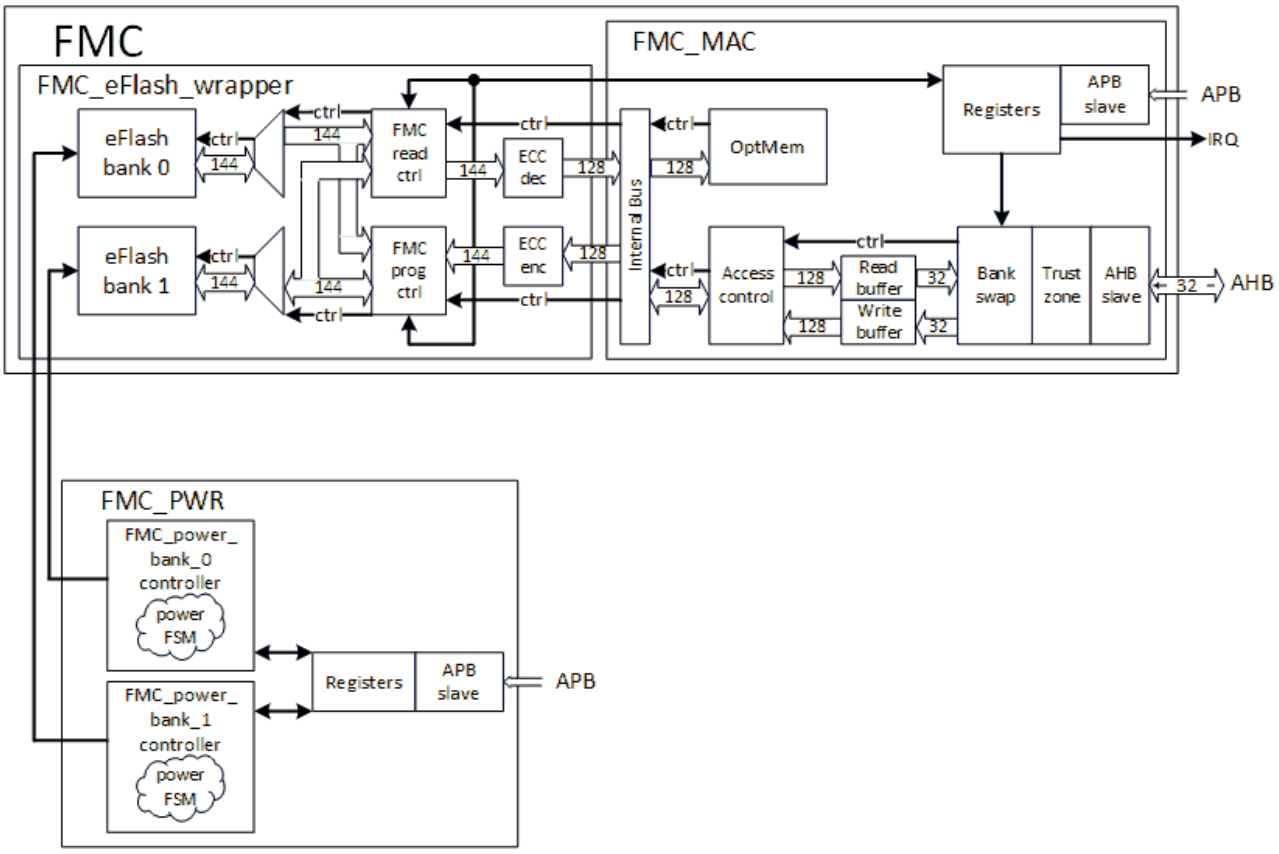


Рисунок 1.17 – Структурная схема FMC

1.3.5.1.2.1 Организация eFlash-памяти

Контроллер FMC включает два банка eFlash-памяти, каждый из которых состоит из двух областей - основной (512Кбайт) и информационной (32 Кбайт). Общий объём eFlash-памяти

контроллера FMC составляет 1Мбайт+64Кбайт. Адреса каждой области (без учета базового смещения) приведены в таблице 1.38.

Области доступны на системной шине по двум базовым адресам - non_secure и secure. Значения базовых смещений приведены в главе 1.3.3.

Таблица 1.38 – Карта памяти контроллера FMC на AHB шине

Диапазон адресов	**Описание области**
0x000000..0x07FFFF	основная память, банк 0, 512 Кбайт (64 страницы)
0x080000..0x0FFFFF	основная память, банк 1, 512 Кбайт (64 страницы)
0x100000..0x107FFF	информационная память, банк 0, 32 Кбайт (4 страницы)
0x108000..0x10FFFF	информационная память, банк 1, 32 Кбайт (4 страницы)

Контроллер FMC позволяет выполнять перестановку банков памяти (bank swapping) - по адресам, соответствующим банку 0, будет доступны данные, хранящиеся в банке 1, и наоборот. Для выполнения перестановки банков необходимо установить 1 в поле BANK_SWAP регистра FMC_CFG. Петестановка банков памяти затрагивает и основнойю и информационную области памяти.

eFlash-память имеет страничную организацию, размер одной страницы одинаков для основной и информационной областей и составляет 8 Кбайт.

Обращение к eFlash-памяти осуществляется словами шириной 144 бита, из которых 128 бит значимых данных и 16 бит ECC (старшие 64 бита данных + 8 бит ECC лдя старших 64 бит данных + младшие 64 бита данных + 8 бит ECC для младших 64 бит данных).

Младшие 256 Кбайт основной памяти каждого банка позволяют 100 000 циклов перезаписи. Остальная память позволяет 10 000 циклов перезаписи.

Каждый банк eFlash-памяти включает в себя две резервные страницы для замены налету повреждённых страниц (Redundancy). Чтение и применение таблицы замен повреждённых страниц выполняется автоматически при выходе контроллера FMC из состояния сброса. Таблица замен формируется в результате тестирования, проводимого на заводе-изготовителе. Для пользователя подмена повреждённой страницы на резервную просиходит прозрачно. Для информации в полях FMC_FLASH_BANK_0_RED_INFO[PAGE_0_REMAP_ADDR], FMC_FLASH_BANK_0_RED_INFO[PAGE_1_REMAP_ADDR], FMC_FLASH_BANK_1_RED_INFO[PAGE_0_REMAP_ADDR], FMC_FLASH_BANK_1_RED_INFO[PAGE_1_REMAP_ADDR] приведены смещения адресов повреждённых страниц.

Контроллер FMC позволяет выполнять следующие сперации с eFlash-памятью:

– чтение слова данных;

– запись одного слова данных;

– запись блока до 8 слов данных;

– стирание страницы eFlash-памяти;

– стирание диапазона страниц eFlash-памяти;

– полное стирание основной области eFlash-памяти;

– автоматическое чтение и применение калибровочных параметров (операция Racell).

Ограничения доступа к eFlash-памяти:

1) после включения контроллера FMC автоматически выполняется операция Recall, любая программная операция будет задержана до завершения операции Recall (операция Recall длится около 400 нс);

2) в один момент времни возможно чтение только из одного банка;

3) запрещено одновременное стирание или запись в разные банки (ограничение должно быть обеспечено пользователем);

4) разрешено чтение из одного банка в процессе записи или стирания другого банка;

5) запрещены одновременные чтение и запись/стирание одного банка (ограничение должно быть обеспечено пользователем).

1.3.5.1.2.2 Временные характеристики

Длительность цикла чтения при напряжении ядра 1,1 В составляет 25 нс, при напряжении ядра 0,9 В - 40 нс.

При изменении частоты HCLK для корректной работы контроллера FMC необходимо программно перенастраивать поля FREQUENCY и LATENCY регистра FMC_CFG.

В поле FMC_CFG[FREQUENCY] задаётся частота HCLK в МГц. Поле FMC_CFG[FREQUENCY] отвечает за временные характеристики всех операций, кроме чтения данных. Некорректное значение FMC_CFG[FREQUENCY] при выходе контроллера FMC из состояния сброса приведёт к некорректному выполнению операции REcall, что в свою очередь приведёт к некорректному выполнению всех остальных операций с eFlash-памятью.

При настройке значени поля FMC_CFG[FREQUENCY] необходимо руководствоваться следующими ограничениям:

– если частота HCLK представляет собой нецелое число, то в поле FMC_CFG[FREQUENCY] необходимо указать значение, округлённое в верх до ближайшего

целого (например, для частоты HCLK 12,5 МГц, нужно установить FMC_CFG[FREQUENCY] = 13);

– если частота HCLK задаётся с большой погрешностью, то в поле FMC_CFG[FREQUENCY] необходимо указать максимально возможное значение (например, для частоты HCLK 100 МГц +/- 10%, нужно установить FMC_CFG[FREQUENCY] = 110);

– при повышении частоты HCLK необходимо сначала установить соответствующее значение FMC_CFG[FREQUENCY], а потом повышать частоту;

– при понижении частоты HCLK необходимо сначала понизить частоту, а потом установить соответствующее значение FMC_CFG[FREQUENCY];

– **обращения к eFlash-памяти при значении FMC_CFG[FREQUENCY], не соответствующем реальной частоте HCLK, могут безвозвратно повредить память**.

Поле FMC_CFG[LATENCY] определяет задержку при выполнении операции чтения. Значение в поле FMC_CFG[LATENCY] указывается в зависимости от частоты HCLK и уровня напряжения питания ядра. Допустимые значения поля FMC_CFG[LATENCY] представлены в таблицах 1.39 и 1.40.

Таблица 1.39 – Значение поля FMC_CFG[LATENCY] в зависимости от частоты работы (при напряжении питания ядра 1,1 В)

частота, МГц	FMC_CTL[LATENCY]						
	0	1	2	3	4	5	6
<= 40	OK	OK	OK	OK	OK	OK	OK
> 40	INVALID	OK	OK	OK	OK	OK	OK
> 80	INVALID	INVALID	OK	OK	OK	OK	OK
> 120	INVALID	INVALID	INVALID	OK	OK	OK	OK
> 160	INVALID	INVALID	INVALID	INVALID	OK	OK	OK

Таблица 1.40 – Значение поля FMC_CFG[LATENCY] в зависимости от частоты работы (при напряжении питания ядра 0,9 В)

частота, МГц	FMC_CTL[LATENCY]						
	0	1	2	3	4	5	6
<= 25	OK	OK	OK	OK	OK	OK	OK
> 25	INVALID	OK	OK	OK	OK	OK	OK
> 50	INVALID	INVALID	OK	OK	OK	OK	OK
> 75	INVALID	INVALID	INVALID	OK	OK	OK	OK
> 100	INVALID	INVALID	INVALID	INVALID	OK	OK	OK
> 125	INVALID	INVALID	INVALID	INVALID	INVALID	OK	OK
> 150	INVALID	INVALID	INVALID	INVALID	INVALID	INVALID	OK

1.3.5.1.2.3 Чтение из eFlash-памяти

Чтение из eFlash-памяти является прозрачной операцией, не требующей программного контроля (за исключением предварительной настройки временных параметров). Это позволяет использовать eFlash-память в качестве загрузочного пространства.

Далее описан механизм чтения контроллера FMC.

Чтение из eFlash-памяти производится 128-битными словами. Для минимизации времени исполнения и потребления питания контроллер FMC буферизует полученные данные и потом выдаёт их на системную шину.

1.3.5.1.2.3.1 Очистка буфера чтения

Буферизация считанных из eFlash-памяти данных может привести к следующим проблемам:

1) ошибки в данных после их изменения в eFlash-памяти записью/стиранием;

2) корректному чтению из защищенного от чтения адреса.

Для решения этих проблем буфер должен быть очищен. Очистка буффера чтения может выполняться:

1) автоматически при любой записи/стирании; функция автоматической очистки может быть программно включена/выключена в поле FMC_CFG[A_CLR_RBUF]. Функция включена по умолчанию и рекомендована быть включенной;

2) однократно при программной установке поля FMC_CONTROL[CLR_RBUF]; рекомендуется использовать после изменений настроек защиты памяти.

1.3.5.1.2.3.2 Предвыборка чтения

Предвыборка чтения - это предварительное чтение из eFlash-памяти в буфер чтения контроллера FMC одного 128-битного слова. При этом используется адрес, следующий за последним адресом, использованным для чтения из контроллера FMC на системной шине.

Предвыборка чтения программно включается/выключается в поле FMC_CFG[PREF_EN].

Включение предвыборки при чтений по последовательным адресам увеличивает производительность, т.к. к моменту чтения по состемной шине данных из следующего 128-битного слова, они уже предзачитаны в буфера чтения контроллера FMC.

Однако, при чтений по случайным адресам включение предвыборки снизит производительность и увеличит потребление питания, за счёт лишних операций чтения из eFlash-памяти.

1.3.5.1.2.4 Запись в eFlash-память

Запись в eFlash-память производится 128-битными словами, по адресам, выровненным по 16 байт. В остальном операция записи в eFlash-память с программной точки зрения выполняется прозрачно, однако занимает длительное время. Для информировании о завершении операции записи контроллер FMC выставляет статус FMC_IRQ_STATUS[WRITE_CMPL] и генерирует прерывание fmc_irq.

Для уменьшения износа eFlash-памяти операция записи на аппаратном уровне выполняется за несколько итераций, точное число которых зависит от технологических особенностей конкретной ячейки памяти, поэтому время записи не детерминировано и может меняться. Минимальное время записи 128-битного слова можно считать равным 200 мкс.

Контроллер FMC буферизует транзакции записи системной шины (ширина 32 бита) в 128-битный буфер (ширина одного слова eFlash-памяти). Предварительная буферизация позволяет не задерживать транзакции на системной шине.

Данные из буфера записи контроллера FMC записываются в eFlash-память при выполнении одного из следующих условий:

– заполнение буфера:

– буфер дополняется транзакциями с такими же битами [31:4] адреса;

– размер и порядок дополняющих транзакций записи может быть любой;

– получение транзакции записи с отличающимися битами [31:4] адреса;

– принудительная запись (подробнее см. 1.3.5.1.5.1.1).

В отсутствии перечисленных выше условий:

– буфер будет ждать дополнения сколь угодно долго, статус FMC_IRQ_STATUS[WRITE_CMPL] не будет установлен;

– чтение данных разрешено из любого банка;

– **чтение данных по сдресу, равному адресу буферизованной записи, вернёт старые данные, хранящиеся в eFlash-памяти**;

– флаг FMC_DBG_STATUS[WBUF_EMPTY] будет равен 0.

Запись в eFlash-память успешно завершена, только если установлен бит FMC_IRQ_STATUS[WRITE_CMPL] и отсутствуют ошибки.

Информация о заполненности буфера записи отображается в поле FMC_DBG_STATUS[WBUF_EMPTY].

Рекомендуемые реализации процедур записи в eFlash-память приведены в пункте 1.3.5.1.5.1.

1.3.5.1.2.4.1 Принудительная запись

Контроллер FMC выполнить принудительную запись в eFlash-память не полного буфера записи. Для выполнения принудительной записи надо установить бит FMC_CONTROL[FWRITE].

С помощью принудительной записи можно реализовать запись блока данных, не кратного 128 битам.

Контроллеро FMC рассчитывает ECC независимо для старших 64 бит и для младших 64 бит 128-битного слова, минимальный блок данных для принудительной записи составляет 64 бита. При необходимости записи в eFlash-памяит блока данных меньше 64 бит, оставшиеся биты будут автоматически дополнены значением 1, и ECC будет рассчитываться именно для дополненного 64-битного блока данных.

Так как запись в eFlash-память выполняется 128-битными словами вне зависимости от числа значимых бит, принудительная запись по 64 бита является мнее эффективной по времени.

Принудительную запись рекомендуется использовать:

– перед изменением настроек контроллера FMC (перестановкой банков, изменением защиты или безопасности памяти);

– при процедуре завершения работы микроконтроллера.

1.3.5.1.2.4.2 Блочная запись

Контроллер FMC поддерживает режим блочной записи. В этом режиме данные пишутся в eFlash-память блоками по несколько (максимум 8) 128-битных слов. Непрерывность и последовательность 128-битных слов в рамках одного блока не требуется, однако, слова одного блока должны попадать в некоторый интервал адресов - иметь одинаковые [31:9] биты.

Режим блочной записи включается и выключается в поле FMC_CFG[WRITE_MODE]. Запрещено менять это поле, если есть неоконченная процедура записи в любой банк.

Блок записывается во eFlash-память:

– автоматически, если блок был заполнен - т.е. подряд поступила запись восьми 128-битных слов с подходящими адресами;

– автоматически, если блок был не пустой и поступила запись 128-битного слова с неподходящим адресом или команда стирания данных из любой области eFlash-памяти любого банка;

– принудительно, программной установкой в 1 поля FMC_CONTROL[BLOCK_FWRITE].

Иначе блочная запись бесконечно ожидает, доступ в другой банк по чтению возможен.

Рекомендуемая реализация процедуры блочной записи приведена в пункте 1.3.5.1.5.2.

Блочная запись в eFlash-память успешно завершена, только если установлен бит FMC_IRQ_STATUS[WRITE_CMPL] и отсутствуют ошибки.

Использование блочной записи приводит к общему ускорению записи.

1.3.5.1.2.5 Стирание памяти

Стирание eFlash-памяти устанавливает биты данных в занчение 1. Стирание необходимо производить перед записью, т.к. запись может переводить значение бита только из 1 в 0, но не наоборот.

В контроллере FMC реализованы два механизма стирания:

1) постраничное стирание (одной страницы или диапазона страниц);

2) общее стирание основной области банка eFlash-памяти.

Рекомендуется использовать постраничное стирание, а не общее, т.к. внезапная просадка напряжения питания при выполнении общего стирания банка eFlash-памяти может привести к безвозвратной потере работоспособности (более подробно см. 1.3.5.1.3).

Одновременное стирание в двух банках памяти запрещено. Присутствует неотключаемая аппаратная защита от такой ситуации:

1) при запросе постраничного/общего стирания при наличии не завершенного постраничного/общего стирания любого банка:

a) запрос будет проигнорирован;

b) будет установлен бит FMC_IRQ_STATUS[ERP_ERR]/FMC_IRQ_STATUS[MERP_ERR];

c) бит FMC_IRQ_STATUS[ERASE_CMPL]/FMC_IRQ_STATUS[M_ERASE_CMPL] по этому событию установлен не будет (он будет установлен по завершении первого запущенного стирания).

Рекомендуемые реализации процедур стирания приведены в пункте 1.3.5.1.5.3.

1.3.5.1.2.5.1 Постраничное стирание

Процедура действует на диапазон страниц eFlash-памяти. Диапазон программно задаётся адресом первой и последней страницы включительно в регистрах FMC_ERASE_ADR_L и FMC_ERASE_ADR_H. За один запуск будет стёрто (8 Кбайт * количество_страниц) данных. Начальный адрес диапазона стирания данных автоматически выравнивается по размеру 8 Кбайт.

По завершении операции постраничного стирания будет сгенерировано прерывание fmc_irq (если не замаскирован) и установлен статус ERASE_CMPL (поле FMC_IRQ_STATUS.[ERASE_CMPL]).

Операция стирания одной страницы занимаеи от

Для уменьшения износа eFlash-памяти операция постраничного стирания на аппаратном уровне выполняется за несколько итераций, точное число которых зависит от технологических особенностей конкретной ячейки памяти, поэтому время стирания одной страницы не детерминировано и может меняться.

Страницы стираются последовательно и полное время стирания будет кратно больше.

Для стирания одной страницы eFlash-памяти необходимо записать одинаковые адреса в регистры FMC_ERASE_ADR_L и FMC_ERASE_ADR_H.

1.3.5.1.2.5.2 Общее стирание банка eFlash-памяти

Операция общего стирания банка eFlash-памяти доступна только для основных областией банков eFlash-памяти. При выполнении операции общего банка eFlash-памяти стирания за одину операцию стирается 512 Кбайт данных. Для каждого банка eFlash-памяти в программных регистрах предусмотрены свои контрольно-управляющие поля.

По завершении операции общего стирания банка eFlash-памяти будет сгенерировано прерывание fmc_irq (если не замаскирован) и установлен статус M_ERASE_CMPL (поле FMC_IRQ_STATUS.[M_ERASE_CMPL]).

Операция общего стирания банка eFlash-памяти длительная - от 20 до 100 мс.

Для информационных областей eFlash-памяти доступна только операция постраничного стирания.

1.3.5.1.2.5.3 Автоматическая защита

Хотя одновременные стирание и запись/чтение из одного банка запрещены, дополнительно предусмотрен программно включаемый механизм автоматической защиты eFlash-памяти при выполнении операций стирания.

Если в поле FMC_CFG[A_ER_PR] установлено значение 0, при незавершенной операции стирания запросы записи/чтения в соответствующий банк eFlash-памяти будут ожидать окончания стирания.

Если в поле FMC_CFG[A_ER_PR] установлено значение 0, при незавершенной операции стирания, соответствующий банк eFlash-памяти будет защищён от записей/чтений. Поведение будет таким же, как при установленной защите WRP и RDP. При завершении операции стирания защита автоматически отключается.

Автоматическая защита никак не влияет на значения программных настроек защиты из пункта 1.3.5.1.2.8.

Использование автоматической защиты предполагается в процессе отладки ПО.

1.3.5.1.2.6 ECC коррекция памяти

Контроллер FMC поддерживает коррекцию ошибок данныхЮ содержащихся в eFlash-памяти с помощью ECC (кода Хэмминга). ECC рассчитывается и проверяется для старших 64 бит и младших 64 бит данных 128 битного слова. ECC контроллера FMC автоматически исправляет ошибку в одном бите (SEC) и детектирует ошибку двух бит (DED).

Проверка правильности и исправление ошибок данных выполняется только при чтении из eFlash-памяти, исправление ошибки в данных, хранящихся в eFlash-памяти не производится.

Для включение/выключение рассчёта и записи ECC в eFlash-память используется поле FMC_CFG[ECC_ENC]. Для включение/выключение проверки ECC, коррекции данных и генерации информации об ошибках используется поле FMC_CFG[ECC_DEC].

В случае ошибки в одном из 64-бит данных (SEC) при чтении по системной шине:

1) исправленные данные будут выданы на системную шину;

2) будет установлен бит FMC_IRQ_STATUS[ECC_SEC] и сгенерировано прерывание fmc_irq, если не замаскировано соответствующим битом регистра FMC_IRQ_MASK;

3) данные, хранящиеся в eFlash-памяти изменены не будут (ошибка в eFlash-памяти сохранится).

В случае ошибки более чем в одном из 64-бит данных (DED) при чтении по системной шине:

1) данные с ошибкой без изменений будут выданы на системную шину;

2) будет установлен бит FMC_IRQ_STATUS[ECC_DED] и вызвано прерывание fmc_irq, если не замаскировано соответствующим битом регистра FMC_IRQ_MASK;

3) будет возвращена ошибка на системной шине (HRESP = 1), если в поле FMC_CFG[ECC_HRESP]установлено значение 1:

a) если транзакция была инициирована ARM ядром, данные с ошибкой не будут использованы и ARM ядром выдаст исключение BusFold (5);

b) реакция других мастеров на системной шине описана в соответствующих разделах;

4) данные, хранящиеся в eFlash-памяти изменены не будут (ошибка в eFlash-памяти сохранится).

В случае ошибки SEC при чтении механизмом OptMem:

1) будет установлен бит FMC_IRQ_STATUS[ECC_SEC] и вызвано прерывание fmc_irq, если не замаскировано соответствующим битом регистра FMC_IRQ_MASK;

2) исправленные данные будут выданы на вектор OptMem и доступны в регистрах FMC_OM_VEC_x.

В случае ошибки DED при чтении механизмом OptMem:

1) будет установлен бит FMC_IRQ_STATUS[ECC_DED] и вызвано прерывание fmc_irq, если не замаскировано соответствующим битом регистра FMC_IRQ_MASK;

2) данные с ошибкой без изменений будут выданы на вектор OptMem и доступны в регистрах FMC_OM_VEC_x.

При любой ошибке в регистре FMC_ECC сохраняется адрес обращения. Чтобы не потерять значение, перезапись адреса блокируется до сброса битов FMC_IRQ_STATUS[ECC_SEC] и FMC_IRQ_STATUS[ECC_DED].

В случае обнаружения ошибки SEC рекомендуется как можно скорее исправить её в eFlash-памяти. Для этого надо:

1) вычитать содержащую SEC ошибку страницу;

2) стереть страницу;

3) перезаписать считанные данные обратно.

1.3.5.1.2.7 Перестановка банков

Контроллер FMC имеет возможность аппаратной перестановки eFlash-банков 0 и 1. Для этого надо инвертировать поле BANK_SWAP регистра FMC_CFG.

Перестановка банков eFlash-памяти оказывает влияние на адреса при записи, чтении и постраничном стирании. На общее стирание банка eFlash-памяти настройка BANK_SWAP влияния не оказывает. Все побанковые настройки, статусы и команды контроллера FMC привязаны к конкретному физическому банку и сохранятся при перестановке.

Перестановка банков eFlash-памяти должна производиться в отсутствии входящих или исполняемых транзакций записи/чтения/стирания. Ограничение должно быть обеспечено пользователем.

Рекомендуемые реализации приведена в главе 1.3.5.1.5.4.

1.3.5.1.2.8 Защита памяти

Контроллер FMC имеет возможность аппаратной защиты eFlash-памяти от чтения, записи, стирания или общего стирания банка.

Память может быть защищена двумя способами:

1) защитой отдельного банка eFlash-памяти; настраивается в полях регистра FMC_PROTECT_CFG_CMN (WRP_0, WRP_1, RDP_0, RDP_1);

2) защитой отдельной страницы банка eFlash-памяти; настраивается:

a) страницы информационной памяти в полях регистра FMC_PROTECT_CFG_CMN (WRP_IP0 .. WRP_IP7, RDP_IP0 .. RDP_IP7) не защищаются от общего стирания, т.к. общее стирание не может быть выполнено для информационной области eFlash-памяти;

b) страницы основной области eFlash-памяти в полях регистров FMC_WRP_CFG_0, FMC_WRP_CFG_1, FMC_WRP_CFG_2, FMC_WRP_CFG_3, FMC_RDP_CFG_0, FMC_RDP_CFG_1, FMC_RDP_CFG_2, FMC_RDP_CFG_3; установленная защита хотя бы одной страницы основной области eFlash-памяти запрещает общее стирание соответствующего банка.

Защита памяти настраивается для конкретного банка. При перестановке банков (при изменении бита FMC_CFG[BANK_SWAP]) настройки защиты сохраняются по отношению к ним, но меняются по отношению к адресному пространству системной шины.

Изменение настроек защиты должно производиться в отсутствии входящих или исполняемых транзакций записи/чтения/стирания. Ограничение должно быть обеспечено пользователем.

При нарушении защиты памяти, вызвавшая её процедура:

– не будет исполнена;

– будет завершена:

– чтение вернёт данные 0xE1E1E1E1 (при обращении к банку 0) или 0xE2E2E2E2 (при обращении к банку 1);

– соответствующий бит регистра FMC_CONTROL (ERASE, MERASE0 или MERASE1) будет сброшен в 0;

– установится соответствующий бит регистра FMC_IRQ_STATUS (WRITE_CMPL или ERASE_CMPL или M_ERASE_CMPL);

– будет установлен один из статусов ошибки (FMC_IRQ_STATUS[RDP_ERR], FMC_IRQ_STATUS[WRP_ERR], FMC_IRQ_STATUS[ERP_ERR], FMC_IRQ_STATUS[MERP_ERR]) и вызвано прерывание fmc_irq, если не замаскировано соответствующим битом регистра FMC_IRQ_MASK.

При установленных в 1 значениях полей FMC_CFG[WRP_BLK] или FMC_CFG[RDP_BLK] при обнаружении ошибки доступа дополнительно будет усилена защита eFlash-памяти. При этом блокируется банк eFlash-памяти, в который была направлена вызвавшая ошибку транзакция - автоматически устанавливается один из битов регистра FMC_PROTECT_CFG_CMN (WRP_0 или WRP_1 или RDP_0 или RDP_1).

1.3.5.1.2.8.1 Блокировка настроек защиты

Блокировка настроек защиты осуществляется установкой в 1 поля FMC_PROTECT_CFG_CMN[PROT_BLK_RST]. При этом будет запрещено программное изменение всех полей, обеспечивающих защиту памяти.

Сброс поля FMC_PROTECT_CFG_CMN[PROT_BLK_RST] в 0 возможен только посредством аппаратного сброса контроллера FMC.

1.3.5.1.2.9 Безопасность

Контроллер FMC поддерживает возможность secure или non-secure режима доступа к eFlash-памяти и своим программным регистрам.

Процедуры записи, чтения считаются безопасными (secure), если у их транзакций на системной шине установлен признак Secure.

Процедуры стирания считаются безопасными (secure), если у запускающих их транзакций (т.е. устанавливающих биты ERASE, MERASE0, MERASE1 в регистре FMC_CONTROL) на системной шине установлен признак Secure.

Доступ к регистру считается безопасным (secure), если у его транзакции на системной шине установлен признак Secure.

Изменение настроек безопасности должно производиться в отсутствии входящих или исполняемых транзакций записи/чтения/стирания. Ограничение должно быть обеспечено пользователем.

1.3.5.1.2.9.1 Запись, чтение и стирание eFlash-памяти

Secure или non-secure режим выполнения операций записи, чтения и стирания настраивается для каждой отдельной страницы:

– информационной области eFlash-памяти - в полях регистра FMC_SECURITY_CFG_CMN (SEC_IP0 .. SEC_IP7);

– основной области eFlash-памяти - в полях регистров FMC_SECURITY_CFG_0, FMC_SECURITY_CFG_1, FMC_SECURITY_CFG_2, FMC_SECURITY_CFG_3; установленный secure режим доступа хотя бы для одной страницы запрещает non-secure общее стирание банка eFlash-памяти, соответствующего адресу страницы.

Secure или non-secure режим настраивается для конкретного диапазона адресов на системной шине. При перестановке банков (при изменении бита FMC_CFG[BANK_SWAP]) настройки secure или non-secure режима меняются по отношению к ним, но сохраняются по отношению к адресному пространству системной шины.

При нарушении безопасности, вызвавшая её процедура:

– не будет исполнена;

– будет завершена:

– чтение вернёт данные 0xE0E0E0E0;

– соответствующий бит регистра FMC_CONTROL (ERASE, MERASE0 или MERASE1) будет сброшен в 0;

– установится соответствующий бит регистра FMC_IRQ_STATUS (WRITE_CMPL или ERASE_CMPL или M_ERASE_CMPL);

– будут установлены общий FMC_SAFETY_IRQ[SEC_ERR] и соответствующий типу ошибки (SEC_ERR, W_ERR, R_ERR, E_ERR, ME0_ERR, ME1_ERR, APB_ERR) статусы и вызвано прерывание fmc_sec_err_irq, если не замаскировано в FMC_SAFETY_MASK[MASK].

1.3.5.1.2.9.2 Доступ к регистрам

Доступ к следующим регистрам возможен только в secure режиме:

– FMC_SECURITY_CFG_CMN;

– FMC_SECURITY_CFG_0;

– FMC_SECURITY_CFG_1;

– FMC_SECURITY_CFG_2;

– FMC_SECURITY_CFG_3;

– FMC_SAFETY_IRQ;

– FMC_SAFETY_MASK.

1.3.5.1.2.10 Option Memory (OptMem)

Часть eFlash-памяти контроллера FMC отводится под Option Memory (OptMem). OptMem аппаратно читается сразу после снятия сигнала сброса контроллера FMC и задаёт начальную конфигурацию МК Optimal+. До тех пор пока не будут прочитаны данные OptMem, не будет снят аппаратный сброс в процессорного ядра МК Optimal+.

Обновление OptMem возможно двумя способами:

1) автоматически после аппаратного сброса (может быть включена/отключена установкой BOOT_OPT_DIS);

2) по программному запросу.

В любом случае механизм OptMem считывает 512 бит данных из начала страницы 3 информационной области банка 0 eFlash-памяти (соответствует системному адресу 0x106000).

Статус FMC_IRQ_STATUS[OM_FINISH] указывает на завершение чтения OptMem. При установке статуса OM_FINISH будет сгенерировано прерывание fmc_irq, если не замаскировано в регистре FMC_IRQ_MASK[OM_FINISH].

По завершении чтения OptMem данные конфигураций можно считать программно из регистров FMC_OM_VEC_0, FMC_OM_VEC_1, FMC_OM_VEC_2, FMC_OM_VEC_3, FMC_OM_VEC_4, FMC_OM_VEC_5, FMC_OM_VEC_6, FMC_OM_VEC_7.

1.3.5.1.2.10.1 Карта соответствия конфигураций

В таблице 1.41 представлено описание конфигураций, содержащихся в векторе OptMem.

Таблица 1.41 – Карта конфигураций в OptMem

Номер 128-битного слова	Расположение данных в слове	Название	Описание
слово_0	31:0	FMC_OM_CMN_CFG_0	1.3.5.1.4.5.46
	63:32	FMC_OM_CMN_CFG_1	1.3.5.1.4.5.47
	95:64	FMC_OM_CPU_NS_STADDR_0	1.3.5.1.4.5.48

Номер 128-бит-ного слова	Расположение данных в слове	Название	Описание
	127:96	FMC_OM_CPU_NS_STADDR_1	1.3.5.1.4.5.49
слово_1	31:0	FMC_OM_CPU_SEC_STADDR0	1.3.5.1.4.5.50
	63:32	FMC_OM_CPU_SEC_STADDR1	1.3.5.1.4.5.51
	95:64	FMC_OM_PROTECT_CFG_CMN	1.3.5.1.4.5.52
	127:96	FMC_OM_SECURITY_CFG_CMN	1.3.5.1.4.5.53
слово_2	31:0	FMC_OM_WRP_CFG_0	1.3.5.1.4.5.54
	63:32	FMC_OM_WRP_CFG_1	1.3.5.1.4.5.55
	95:64	FMC_OM_WRP_CFG_2	1.3.5.1.4.5.56
	127:96	FMC_OM_WRP_CFG_3	1.3.5.1.4.5.57
слово_3	31:0	FMC_OM_SECURITY_CFG_0	1.3.5.1.4.5.58
	63:32	FMC_OM_SECURITY_CFG_1	1.3.5.1.4.5.59
	95:64	FMC_OM_SECURITY_CFG_2	1.3.5.1.4.5.60
	127:96	FMC_OM_SECURITY_CFG_3	1.3.5.1.4.5.61

1.3.5.1.2.11 Прерывание длительных операций

Контроллер FMC имеет возможность прервать выполнение длительных операций по программному запросу.

Могут быть прерваны:

– запись, блочная запись;

– постраничное стирание - в регистре FMC_ERASE_ADR_L будет сохранен адрес, на котором произошло прерывание.

– общее стирание основной области банка eFlash-памяти.

Чтение не будет прервано - будет исполнено обычным образом.

Чтобы прервать выполнение длительной операции, необходимо:

– убедиться, что прерывание выполнение длительной операции не активно (т.е. поле FMC_CONTROL[ABORT] сброшено в 0);

– установить поле FMC_CONTROL[ABORT] в 1;

– дождаться сброса поля FMC_CONTROL[ABORT] в 0.

Про прерывании выполнения длительной операции буфер записи не будет очищен (если в нем содержалась часть записываемого слова, то она не будет потеряна).

1.3.5.1.3 Управление питанием контроллера FMC

С точки зрения управления питанием для каждого банка eFlash-памяти контроллера FMC предусмотрены следующие режимы потребления:

– RUN - выполняется какая-либо операция (чтение, запись, постраничное стирание или общее стирание банка памяти);

– STANDBY - контроллер готов к работе, но не выполняет никаких операций;

– SLEEP - отключены часть внутренних схем макроса eFlash-памяти, выполнение каких-либо операций не доступно;

– POWER_DOWN - снято питание с макроса eFlash-памяти, выполнение каких-либо операций не доступно.

Актуальное состояние банка eFlash-памяти отображается в регистре FMC_CFG - поля B0_POWER_STATE и B1_POWER_STATE.

Если банк eFlash-памяти находится в состоянии POWER_DOWN или SLEEP или программно запрошен переход в одно из этих состояний, автоматически включается защита соответствующего банка eFlash-памяти от записи, чтения и стирания. Поведение будет таким же, как при установленной защите WRP и RDP. При возврате в состояние STANDBY или RUN защита автоматически отключается.

Для сохранения работоспособности макросов eFlash-памяти в составе контроллера FMC необходимо соблюдение определённых предосторожностей:

1) во время выполнения операции записи, постраничного стирания или общего стирания банка eFlash-памяти, напряжение питания должно оставаться стабильным;

2) внезапное пропадание напряжения питания (или просадке ниже 1,47 В для входного напряжения или ниже 0.65 В для напряжения ядра) во время выполнения длительной операции (операции записи, постраничного стирания или общего стирания банка памяти) может привести к безвозвратной потере работоспособности макроса eFlash-памяти;

3) при подаче сигнала сброса контроллера FMC во время выполнения длительной операции необходимо выждать 5 мкс при прерывании операции записи или постраничного стирания и 100 мкс при прерывании общего стирания банка eFlash-памяти, прежде чем выполнять новую операцию.

Для корректного выхода контроллера FMC из состояния сброса и реализации механизмов управления питанием используется отдельный блок, расположенный в независимом домене питания PD_AON - FMC Power контроллер.

FMC Power контроллер обладает собственным набором регистров, интегрированных в систему посредством интерфейса APB (см. 1.3.5.1.4.4).

FMC Power контроллер позволяет независимо отслеживать состояние и управлять работой каждого банка eFlash-памяти контроллера FMC. Каждый макрос eFlash-памяти может находиться в следующих состояниях:

– POWER_DOWN - снято питание ядра с макроса eFlash-памяти (соответствующий банк контроллера FMC находится в состоянии POWER_DOWN), потребление банка памяти контроллера FMC минимально;

– SWITCH_ON - на макрос eFlash-памяти подано питание, но он ещё не готов к работе (соответствующий банк контроллера FMC находится в состоянии POWER_DOWN), длительность включения (T_{wk}) в соответствии со спецификацией макроса eFlash-памяти составляет не менее 5 мкс;

– STANDBY - макрос eFlash памяти готов к работе и не выполняет никаких длительных операций - могут выполняться только операции чтения (соответствующий банк контроллера FMC находится в состоянии STANDBY);

– ACTIVE - макрос eFlash-памяти выполняет длительную операцию - запись, стирание страницы или общее стирание банка памяти (соответствующий банк контроллера FMC находится в состоянии RUN);

– BOR - произошла внезапная просадка напряжения питания ниже уровня заданного в поле RCU_BORCR[BOR33LVL] (по умолчанию 1,72 В) для предотвращения повреждения макроса eFlash-памяти прерываются любые операции и выдерживается время не менее T_{nvh} при прерывании операции чтения или стирания страницы и T_{nvh1} при прерывании операции общего стирания банка памяти (по спецификацией макроса eFlash-памяти T_{nvh} >= 5 мкс и T_{nvh1} >= 100 мкс); на контроллер FMC подаётся сигнал аппаратного сброса;

– RESETTING - на контроллер FMC подаётся или был подан сигнал аппаратного сброса (контроллер FMC находится в состоянии POWER_DOWN);

– SLEEP - макрос eFlash-памяти переведён в режим пониженного энергопотребления (сна), в нём отключены некоторые внутренние схемы (соответствующий банк контроллера FMC находится в состоянии SLEEP);

– WAKE_UP - выход макроса eFlash-памяти из состояния сна (соответствующий банк контроллера FMC всё ещё находится в состоянии SLEEP), в соответствии со спецификацией макроса eFlash-памяти выдерживается задержка T_{wk}.

Машина состояний FMC Power контроллера представлена на рисунке 1.18).

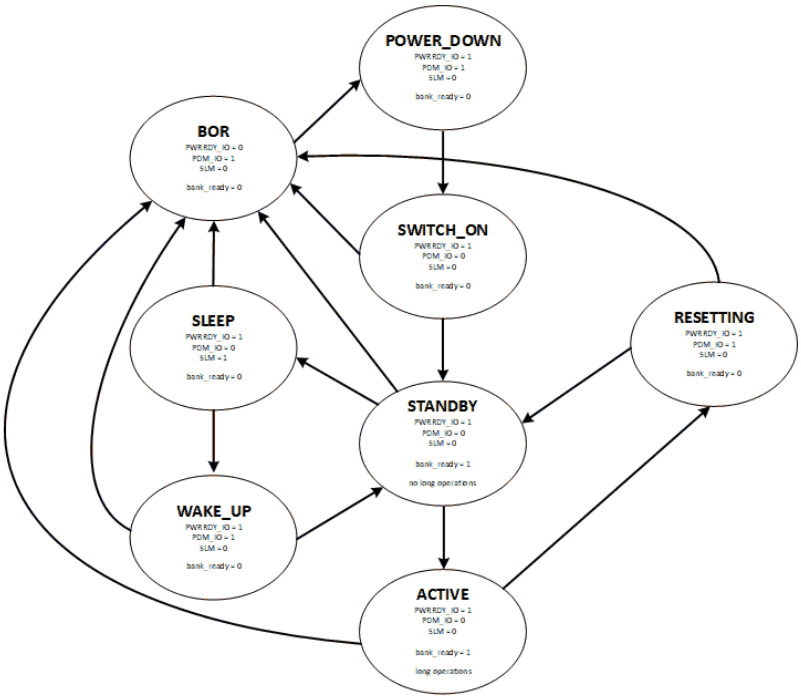


Рисунок 1.18 – Машина состояний FMC Power контроллера

1.3.5.1.3.1 Настройка FMC Power контроллера

Управление питанием контроллера FMC выполняется через регистры FMC Power контроллера.

Все регистры FMC Power контроллера всегда доступны на чтение, но для возможности записи необходимо сначала выполнить операцию разблокировки доступа - записать ключ 0x7375646F в регистр FMC_PWR_KEY. Доступ на запись к регистрам FMC Power контроллера будет сохраняться пока в регистр FMC_PWR_KEY не будет записано любое значение, отличное от ключа.

Если регистры FMC Power контроллера доступны на запись, то из регистра FMC_PWR_KEY считывается 0x1, в противном случае - 0x0.

Для управления состоянием питания банков контроллера FMC используется регистр FMC_PWR_SET_STATE:

– поля BANK_x_SET_POWER_STATE отвечают за состояние отдельных банков контроллера FMC;

– поля LVE_MODE и SET_LVE позволяют регулировать работу контроллера FMC при изменении уровноя напряжения ядра;

– поля BANK_x_SET_PDM носят отладочный характер и позволяют программно управлять сигналом, информирующем о стабильности напряжения питания ядра PDM_IO;

– поле SET_PWRRDY носит отладочный характер и позволяют программно управлять сигналом, информирующем о стабильности напряжения опорного питания PWRRDY_IO;

– значения полей BANK_x_SET_PDM и SET_PWRRDY применяются только при условии установки 0x1 в поле PWR_MODE.

1.3.5.1.3.2 Операции по управлению питанием контроллера FMC

1.3.5.1.3.2.1 Включение контроллера FMC

После включения МК Optimal+ контроллер FMC находится в состоянии POWER_DOWN, FMC Power контроллер также находится в состоянии POWER_DOWN. Выход из состояния POWER_DOWN осуществляется:

– аппаратно при условии установки флага auto_power_up соответствующего банка памяти контроллера FMC;

– программно - подачей команды включения банка памяти контроллера FMC (GOTO_ENABLE) - записью в поле FMC_PWR_SET_STATE[BANK_x_SET_POWER_STATE] значения 0x2.

Программное включение банков памяти контроллера FMC требуется если при включении МК Optimal+ не установлен флаг auto_power_up соответствующего банка памяти контроллера FMC, или если банк памяти контроллера FMC был програмно выключен (переведён в состояние POWER_DOWN).

FMC Power контроллер выдерживает задержку T_{pud} тактов своей частоты работы перед началом выполнения включения - переходом в состояние SWITCH_ON. Спецификация макроса eFlash-памяти требует выдерживать не менее 100 нс после стабилизации питания ядра до включения.

После выхода из состояния POWER_DOWN банк памяти контроллера FMC аппаратно зачитывает тримминги (выполняет операцию RECALL) и таблицу замен повреждённых страниц, после чего переходит в состояние STANDBY.

Статус включения банка памяти контроллра FMC отражается в полях B0_POWER_ON и B1_POWER_ON регистра FMC_IRQ_STATUS.

1.3.5.1.3.2.2 Выключение банка памяти контроллера FMC

Выключения банка памяти контроллера FMC (переход в состояние POWER_DOWN) осуществляется:

– аппарано:

– при снятии питания или аппаратном сбросе и контроллера FMC, и FMC Power контроллера;

– при выходе из состояния BOR;

– программно подачей команды выключения банка памяти контроллера FMC (GOTO_POWER_DOWN) - записью в поле FMC_PWR_SET_STATE[BANK_x_SET_POWER_STATE] значения 0x0;

1.3.5.1.3.2.3 Сброс контроллера FMC

Для выполнения аппаратного сброса контроллера FMC необходимо дождаться завершения всех операций. Сброс контроллера FMC во время выполнения длительной операции (запись, стирание страницы или полное стирание банка памяти памяти) является нештатной ситуацией, которая обрабатывается особым образом.

При условии, что выполнен аппаратный сброс контроллера FMC, но FMC Power контроллер не сбрасывался, после выхода из состояния POWER_DOWN банк контроллер FMC аппаратно зачитывает только таблицу замен повреждённых страниц, после чего переходит в состояние STANDBY. Т.к. питание с макроса eFalsh-памяти не снималось, операция RECALL не выполняется.

Поведение после выполнения аппаратного сброса и контроллера FMC, и FMC Power контроллера аналогично поведению после включения МК Optimal+.

1.3.5.1.3.2.4 Режим сна макроса eFlash-памяти

В режиме сна (состояние SLEEP) за счёт отключения внутренних схем снижается потребление макрос eFlash-памяти, однако переход из режима сна в активный режим работы не может выполняться мгновенно.

Для перехода в режим сна необходимо подать команду включения банка памяти контроллера FMC (GOTO_SLEEP) - записать в поле FMC_PWR_SET_STATE[BANK_x_SET_POWER_STATE] значение 0x1.

Для выхода из состояния сна необходимо подать команду включения банка памяти контроллера FMC (GOTO_ENABLE), после чего FMC Power контроллер перейдёт в состояние WAKE_UP.

В случае выполнения аппаратного сброса контроллера FMC, банк(и) которого находится в состоянии SLEEP они будут аппаратно включены (переведены в состояние STANDBY).

1.3.5.1.3.2.5 Изменение уровня напряжения контроллера FMC

Для изменения уровня напряжения питания ядра МК Optimal+ необходимо дождаться завершения контроллером FMC всех операций (аппаратура FMC Power контроллера аппаратно задерживает сигнал LVE, управляющий включением режима работы eFlash-памяти на пониженном напряжении ядра, до завершения любой операции, однако корректность выполнения операции при этом не гарантируется).

Т.к. не возможность точно определить момент стабилизации напряжения VDD_{core} после отправки команды на его изменение, FMC Power контроллер изменяет сигнал LVE с задержкой T_{vls} мкс.

Для МК Optimal+ предусмотрены следующие режимы выработки сигнала LVE (определяется полем FMC_PWR_SET_STATE[LVE_MODE]):

– 0 - полностью аппаратный - сигнал LVE изменяется через T_{vls} после получения информации об изменении уровня напряжения VDD_{core};

– 1 - полностью программный - состояние сигнала LVE задаётся полем FMC_PWR_SET_STATE[SET_LVE];

– 2 - смешанный - из программного и аппаратного значений выбирается то, которое соответствует пониженному уровню напряжения VDD_{core}.

1.3.5.1.3.2.6 Поведение контроллера FMC в нештатных ситуациях

Нештатными ситуациями для контроллера FMC являются пропадание питания или аппаратный сброс во время выполнения длинной операции (записи, стирания страницы или полного стирания банка).

1.3.5.1.3.2.6.1 Аппаратный сборс контроллера FMC во время выполнения длинной операции

Если сигналом аппаратного сброса была прервана длительная операция, перед возвратом контроллера FMC в рабочее состояние выдерживается время не менее T_{nvh} при прерывании операции чтения или стирания страницы и T_{nvh1} при прерывании операции общего стирания банка памяти, а после этого выдерживается время восстановления T_{rcv}. По спецификацией макроса eFlash-памяти $T_{nvh} \geq 5$ мкс, $T_{nvh1} \geq 100$ мкс, $T_{rcv} \geq 10$ мкс, всё это время контроллер FMC находится в состоянии POWER_DOWN, а FMC Power контроллер находится в состоянии RESETTING.

После выхода из состояния RESETTING для каждого банка памяти контроллера FMC аппаратно выполняется чтение таблицы замен повреждённых страниц, т.к. эта информация была утеряна выполнением аппаратного сброса, но операция RECALL **не** выполняется, т.к. питание с макроса eFlash-памяти не снималось.

1.3.5.1.3.2.6.2 Пропадание питания МК Optimal+ во время выполнения длинной операции контроллером FMC

Если произошла внезапная просадка напряжения питания ниже уровня заданного в поле RCU_BORCR[BOR33LVL] (по умолчанию 1,72 В) для предотвращения повреждения макроса eFlash-памяти прерываются любые операции и подаётся аппаратный сброс на контроллер FMC и FMC Power контроллер.

За счёт общей ёмкости цепей питания МК Optimal+ время T_{nvh} при прерывании операции чтения или стирания страницы при внезапная просадка напряжения питания будет выдерживаться автоматически. Время T_{nvh1}, необходимое при прерывании операции общего стирания банка памяти может быть несоблюдено в следствии слшком быстрой просадки напряжения. Поэтому операцию общего стирания банка памяти пользователь выполняет под свою ответственность. **Производитель МК Optimal+ рекомендует вместо общего стирания банка памяти выполнять постраничное стирание для диапазона равного основной области банка памяти**.

1.3.5.1.3.2.6.3 Информация и состоянии контроллера FMC при выполнении аппаратного сброса

При возникновении нештатной ситуации типа аппаратный сброса контролелра FMC во время выполнения длительной операции в регистре FMC_PWR_ERR_STATE FMC Power контроллера сохраняется информация о типе операции, а также номер и область банка памяти контроллера FMC.

Значение в регистре FMC_PWR_ERR_STATE изменяется только при возникновении следующей нештатной ситуации или при аппаратном сбросе FMC Power контроллера.

При возникновении нештатной ситуации результат выполнения текущей операции следует считать некорректным, и после восстановления работоспособности контроллера FMC, операцию повторить.

При выполнении блочной записи в поле FMC_PWR_ERR_STATE[ADDESS] сохраняется первый адрес блока данных, однако, все данные блока следует считать некорректными.

1.3.5.1.4 Описание регистров

1.3.5.1.4.1 Общее описание регистров

Контроллер флэш памяти FMC имеет набор программно доступных регистров, требующий 4 КБайт адресов на системной шине APB.

Карта регистров приведена в таблице 1.42.

При дальнейшем описании отдельных полей регистров используется следующее обозначение режимов доступа пользователя к полям:

– ЧТ – поле доступно только для чтения. Запись данных в поле игнорируется;

– ЗП – поле доступно только для записи. Чтение возвращает значение 0;

– ЧТ/ЗП – поле доступно как для записи, так и для чтения;

Доступ к регистрам осуществляется посредством системного интерфейса управления (APB slave). Поддерживается только 32-х разрядный доступ. Байтовая или двух-байтовая запись будет проигнорирована и возвращена ошибка шины.

Запись в зарезервированные или ЧТ биты или адреса не имеет действия, никакие ошибки не генерируются. Чтение из зарезервированных или ЗП битов или адресов возвращает 0, никакие ошибки не генерируются.

1.3.5.1.4.2 Поддержка безопасности регистров

Некоторые регистры FMC доступны только в безопасном режиме работы.

Возможности поддержки безопасности каждого конкретного регистра приведены в его описании.

Секьюрный доступ к не секьюрным регистрам разрешен.

Не секьюрный доступ к секьюрным регистрам не будет иметь действия (при записи) или вернет 0 (при чтении), а также установит бит FMC_SAFETY_IRQ.APB_ERR. Автоматически установится бит FMC_SAFETY_IRQ.IRQ.

1.3.5.1.4.3 Карта регистров

Таблица 1.42 – Карта регистров FMC

Адрес	Название	Доступ	Значение после сброса	Описание
0x000	FMC_ID	ЧТ	0x5F434D46	Идентификатор устройства
0x004	FMC_VERSION	ЧТ	0x1	Версия устройства
0x008	FMC_IRQ_STATUS	ЧТ	0x0	Статусы прерываний контроллера FMC
0x00C	FMC_IRQ_MASK	ЧТ/ЗП	0x0	Маска (разрешение) прерываний

Адрес	Название	Доступ	Значение после сброса	Описание
0x010	FMC_IRQ_TEST	ЗП	0x0	Программная установка статуса прерывания
0x014	FMC_IRQ_CLR	ЗП	0x0	Программный сброс статуса прерывания
0x018	FMC_CFG	ЧТ/ЗП		Настройки контроллера FMC
0x01C	FMC_CONTROL	ЧТ/ЗП		Команды контроллера FMC
0x020	FMC_ERASE_ADR_L	ЧТ/ЗП	0x0	Адрес первой страницы процедуры постраничного стирания
0x024	FMC_ERASE_ADR_H	ЧТ/ЗП	0x0	Адрес последней страницы процедуры постраничного стирания
0x028	FMC_DBG_STATUS	ЧТ		Отладочные статусы контроллера FMC
0x02C	FMC_ECC	ЧТ/ЗП	0x0	Адрес обнаруженной ошибки ECC во флэш памяти
0x030	FMC_PROTECT_CFG_CMN	ЧТ/ЗП	0x0	Общие настройки защиты
0x034	FMC_WRP_CFG_0	ЧТ/ЗП	0x0	Настройки постраничной защиты от записи и стирания 0
0x038	FMC_WRP_CFG_1	ЧТ/ЗП	0x0	Настройки постраничной защиты от записи и стирания 1
0x03C	FMC_WRP_CFG_2	ЧТ/ЗП	0x0	Настройки постраничной защиты от записи и стирания 2
0x040	FMC_WRP_CFG_3	ЧТ/ЗП	0x0	Настройки постраничной защиты от записи и стирания 3
0x044	FMC_RDP_CFG_0	ЧТ/ЗП	0x0	Настройки постраничной защиты от чтения 0
0x048	FMC_RDP_CFG_1	ЧТ/ЗП	0x0	Настройки постраничной защиты от чтения 1
0x04C	FMC_RDP_CFG_2	ЧТ/ЗП	0x0	Настройки постраничной защиты от чтения 2
0x050	FMC_RDP_CFG_3	ЧТ/ЗП	0x0	Настройки постраничной защиты от чтения 3
0x054	FMC_SECURITY_CFG_CMN	ЧТ/ЗП	0x0	Общие настройки безопасности
0x058	FMC_SECURITY_CFG_0	ЧТ/ЗП	0x0	Настройки постраничной безопасности 0
0x05C	FMC_SECURITY_CFG_1	ЧТ/ЗП	0x0	Настройки постраничной безопасности 1
0x060	FMC_SECURITY_CFG_2	ЧТ/ЗП	0x0	Настройки постраничной безопасности 2
0x064	FMC_SECURITY_CFG_3	ЧТ/ЗП	0x0	Настройки постраничной безопасности 3
0x068	FMC_SAFETY_IRQ	ЧТ/ЗП	0x0	Прерывание о нарушении безопасности
0x06C	FMC_SAFETY_MASK	ЧТ/ЗП	0x0	Маска (разрешение) прерывания о нарушении безопасности
0x200	FMC_FLASH_SETTINGS_KEY	ЧТ/ЗП	0x0	Ключ, разрешающий запись в регистры настройки макросов eFlash памяти контроллера FMC
0x204	FMC_FLASH_SETTINGS	ЧТ/ЗП	0x1	Общие настройки работы с макросами eFlash памяти
0x208	FMC_FLASH_GLOBAL_TIMINGS	ЧТ/ЗП	0x11020410	Общие настройки временных параметров работы с макросами eFlash памяти
0x210	FMC_FLASH_READ_TIMINGS	ЧТ/ЗП	0x21	Настройки временных параметров чтения данных из макроса eFlash памяти

Адрес	Название	Доступ	Значение после сброса	Описание
0x214	FMC_FLASH_RECALL_TIMINGS	ЧТ/ЗП	0x07060301	Настройки временных параметров чтения калибровочных данных (триммингов) из макроса eFlash памяти
0x220	FMC_SMW_TIMER_OPTINS	ЧТ/ЗП	0x210	Настройки таймеров для смарт-операций
0x224	FMC_SMW_SETTING_OPTIONS_H	ЧТ/ЗП	0x014AC715	Настрокйки напряжений и временных параметров для смарт-операций записи
0x228	FMC_SMW_SETTING_OPTIONS_L	ЧТ/ЗП	0x0F0B4000	Настрокйки напряжений и временных параметров для смарт-операций записи
0x22C	FMC_SMW_SETTING_OPTIONS2	ЧТ/ЗП	0x00A80151	Настрокйки напряжений и временных параметров для дополнительных смарт-операций записи
0x230	FMC_SMW_SETTING_OPTIONS3	ЧТ/ЗП	0x0001BD00	Дополнительные параметры для смарт-операций записи
0x234	FMC_SMW_SMP_WHV_OPTIONS_H	ЧТ/ЗП	0x77777777	Настройка напряжений смарт-операций записи
0x238	FMC_SMW_SMP_WHV_OPTIONS_L	ЧТ/ЗП	0x77777765	Настройка напряжений смарт-операций записи
0x23C	FMC_SMW_SME_WHV_OPTIONS_H	ЧТ/ЗП	0xCCCCCCCC	Настройка напряжений смарт-операций стирания
0x240	FMC_SMW_SME_WHV_OPTIONS_L	ЧТ/ЗП	0xCCBA9876	Настройка напряжений смарт-операций стирания
0x244	FMC_SMW_DEBUG_STAT	ЧТ/ЗП	0x0	Статистика выполнения смарт-операции
0x250	FMC_FLASH_BANK_0_RED_INFO	ЧТ/ЗП	0x0	Таблица замен повреждённых страниц для банка 0
0x254	FMC_FLASH_BANK_1_RED_INFO	ЧТ/ЗП	0x0	Таблица замен повреждённых страниц для банка 1
0x400	FMC_OM_CMN_CFG_0	ЧТ	0x0	Данные 0 вектора конфигураций OptMem
0x404	FMC_OM_CMN_CFG_1	ЧТ	0x0	Данные 1 вектора конфигураций OptMem
0x408	FMC_OM_CPU_NS_STADDR_0	ЧТ	0x00000000	Данные 2 вектора конфигураций OptMem
0x40C	FMC_OM_CPU_NS_STADDR_1	ЧТ	0x00108000	Данные 3 вектора конфигураций OptMem
0x410	FMC_OM_CPU_SEC_STADDR0	ЧТ	0x10000000	Данные 4 вектора конфигураций OptMem
0x414	FMC_OM_CPU_SEC_STADDR1	ЧТ	0x10108000	Данные 5 вектора конфигураций OptMem
0x418	FMC_OM_PROTECT_CFG_CMN	ЧТ	0x0	Данные 6 вектора конфигураций OptMem
0x41C	FMC_OM_SECURITY_CFG_CMN	ЧТ	0x0	Данные 7 вектора конфигураций OptMem
0x420	FMC_OM_WRP_CFG_0	ЧТ	0x0	Данные 8 вектора конфигураций OptMem
0x424	FMC_OM_WRP_CFG_1	ЧТ	0x0	Данные 9 вектора конфигураций OptMem
0x428	FMC_OM_WRP_CFG_2	ЧТ	0x0	Данные 10 вектора конфигураций OptMem
0x42C	FMC_OM_WRP_CFG_3	ЧТ	0x0	Данные 11 вектора конфигураций OptMem

Адрес	Название	Доступ	Значение после сброса	Описание
0x430	FMC_OM_SECURITY_CFG_0	ЧТ	0x0	Данные 12 вектора конфигураций OptMem
0x434	FMC_OM_SECURITY_CFG_1	ЧТ	0x0	Данные 13 вектора конфигураций OptMem
0x438	FMC_OM_SECURITY_CFG_2	ЧТ	0x0	Данные 14 вектора конфигураций OptMem
0x43C	FMC_OM_SECURITY_CFG_3	ЧТ	0x0	Данные 15 вектора конфигураций OptMem

1.3.5.1.4.4 Карта регистров FMC_PWR контроллера

В таблице 1.43 представлена карта регистров FMC_PWR контроллера.

Таблица 1.43 – Карта регистров FMC

Адрес	Название	Доступ	Значение после сброса	Описание
0x000	FMC_PWR_KEY	ЧТ/ЗП	0x0	Ключ, разрешающий запись в регистры FMC_PWR контроллера
0x004	FMC_PWR_SET_STATE	ЧТ/ЗП	0x01300200	Настройка состояния банков eFlash памяти контроллера FMC
0x008	FMC_PWR_CUR_STATE	ЧТ	0x0	Текущее состояние FMC_PWR контроллера
0x00C	FMC_PWR_ERR_STATE	ЧТ	0x0	Состояние контроллера FMC в момент когда произошёл аппартный сброс или пропадание напряжения
0x010	FMC_PWR_SETTINGS	ЧТ/ЗП	0x01110A10	Настройки FMC_PWR контроллера
0x014	FMC_PWR_TIMINGS	ЧТ/ЗП	0x0A64050A	Настройки временных параметров FMC_PWR контроллера

1.3.5.1.4.5 Описание полей регистров

1.3.5.1.4.5.1 FMC_ID (0x000)

Регистр идентификатора контроллера FMC. В таблице 1.44 представлено описание полей регистра FMC_ID.

Таблица 1.44 – Описание полей регистра FMC_ID

Биты	Поле	Доступ	Значение после сброса	Описание
31:0	ID	ЧТ	0x5F636D66	идентификатор контроллера FMC

1.3.5.1.4.5.2 FMC_VERSION (0x004)

Регистр версии контроллера FMC. В таблице 1.45 представлено описание полей регистра FMC_VERSION.

Таблица 1.45 – Описание полей регистра FMC_VERSION

Биты	Поле	До-ступ	Значение после сброса	Описание
31:0	VERSION	ЧТ	0x1	версия устройства FMC

1.3.5.1.4.5.3 FMC_IRQ_STATUS (0x008)

Регистр отображает статусы прерываний контроллера FMC. Доступен только на чтение. Запись в любой из битов не имеет действия. Для сброса статуса надо установить соответствующий бит в регистре FMC_IRQ_CLR. Для установки статуса установите соответствующий бит в регистре FMC_IRQ_TEST (возможность предусмотрена для отладочных целей).

В таблице 1.46 представлено описание полей регистра FMC_IRQ_STATUS.

Таблица 1.46 – Описание полей регистра FMC_IRQ_STATUS

Биты	Поле	До-ступ	Значение после сброса	Описание
0	WRITE_CMPL	ЧТ	0	Устанавливается в 1 автоматически при завершении (успешном или нет) записи в любой банк
1	ERASE_CMPL	ЧТ	0	Устанавливается в 1 автоматически при завершении (успешном или нет) стирания всех страниц из указанного набора
2	M_ERASE_CMPL	ЧТ	0	Устанавливается в 1 автоматически при завершении (успешном или нет) массового стирания основной памяти любого банка
3	RDP_ERR	ЧТ	0	Устанавливается в 1 автоматически при попытке чтения из защищенного RDP банка/страницы
4	WRP_ERR	ЧТ	0	Устанавливается в 1 автоматически при попытке записи в защищенный WRP банк/страницу
5	ERP_ERR	ЧТ	0	Устанавливается в 1 автоматически при попытке стирания: - защищенного WRP банка/страницы; - при не заверешенном любом другом стирании.
6	MERP_ERR	ЧТ	0	Устанавливается в 1 автоматически при попытке массового стирания: - любого из защищенных банков; - банка, содержащего хотя бы одну защищенную WRP страницу основной памяти; - при не заверешенном любом другом стирании.
7	ECC_SEC	ЧТ	0	Устанавливается в 1 автоматически при ошибке ECC SEC
8	ECC_DED	ЧТ	0	Устанавливается в 1 автоматически при ошибке ECC DED
9	B0_POWER_ON	ЧТ	0	Устанавливается в 1 автоматически при включении банка 0
10	B1_POWER_ON	ЧТ	0	Устанавливается в 1 автоматически при включении банка 1

Биты	Поле	До-ступ	Значение после сброса	Описание
11	B0_WAKEUP	ЧТ	0	Устанавливается в 1 автоматически при переходе банка 0 из состояния SLEEP в состояние STANDBY или RUN
12	B1_WAKEUP	ЧТ	0	Устанавливается в 1 автоматически при переходе банка 1 из состояния SLEEP в состояние STANDBY или RUN
13	B0_RECALL	ЧТ	0	Устанавливается в 1 автоматически при завершении загрузки триммингговых значений банка 0
14	B1_RECALL	ЧТ	0	Устанавливается в 1 автоматически при завершении загрузки триммингговых значений банка 1
15	OM_FINISH	ЧТ	0	Устанавливается в 1 автоматически при завершении инициализации конфигураций OptMem

1.3.5.1.4.5.4 FMC_IRQ_MASK (0x00C)

Регистр маски прерываний контроллера FMC. В таблице 1.47 представлено описание полей регистра FMC_IRQ_MASK.

Таблица 1.47 – Описание полей регистра FMC_IRQ_MASK

Биты	Поле	До-ступ	Значение после сброса	Описание
0	WRITE_CMPL	ЧТ/ЗП	0	1 - разрешение генерации прерывания при установке статуса FMC_IRQ_STATUS.WRITE_CMPL. 0 - прерывание запрещено;
1	ERASE_CMPL	ЧТ/ЗП	0	1 - разрешение генерации прерывания при установке статуса FMC_IRQ_STATUS.ERASE_CMPL. 0 - прерывание запрещено;
2	M_ERASE_CMPL	ЧТ/ЗП	0	1 - разрешение генерации прерывания при установке статуса FMC_IRQ_STATUS.M_ERASE_CMPL. 0 - прерывание запрещено;
3	RDP_ERR	ЧТ/ЗП	0	1 - разрешение генерации прерывания при установке статуса FMC_IRQ_STATUS.RDP_ERR. 0 - прерывание запрещено;
4	WRP_ERR	ЧТ/ЗП	0	1 - разрешение генерации прерывания при установке статуса FMC_IRQ_STATUS.WRP_ERR. 0 - прерывание запрещено;
5	ERP_ERR	ЧТ/ЗП	0	1 - разрешение генерации прерывания при установке статуса FMC_IRQ_STATUS.ERP_ERR. 0 - прерывание запрещено;
6	MERP_ERR	ЧТ/ЗП	0	1 - разрешение генерации прерывания при установке статуса FMC_IRQ_STATUS.MERP_ERR. 0 - прерывание запрещено;
7	ECC_SEC	ЧТ/ЗП	0	1 - разрешение генерации прерывания при установке статуса FMC_IRQ_STATUS.ECC_SEC. 0 - прерывание запрещено;
8	ECC_DED	ЧТ/ЗП	0	1 - разрешение генерации прерывания при установке статуса FMC_IRQ_STATUS.ECC_DED. 0 - прерывание запрещено;

Биты	Поле	Доступ	Значение после сброса	Описание
9	B0_POWER_ON	ЧТ/ЗП	0	1 - разрешение генерации прерывания при установке статуса FMC_IRQ_STATUS.B0_POWER_ON. 0 - прерывание запрещено;
10	B1_POWER_ON	ЧТ/ЗП	0	1 - разрешение генерации прерывания при установке статуса FMC_IRQ_STATUS.B1_POWER_ON. 0 - прерывание запрещено;
11	B0_WAKEUP	ЧТ/ЗП	0	1 - разрешение генерации прерывания при установке статуса FMC_IRQ_STATUS.B0_WAKEUP. 0 - прерывание запрещено;
12	B1_WAKEUP	ЧТ/ЗП	0	1 - разрешение генерации прерывания при установке статуса FMC_IRQ_STATUS.B1_WAKEUP. 0 - прерывание запрещено;
13	B0_RECALL	ЧТ/ЗП	0	1 - разрешение генерации прерывания при установке статуса FMC_IRQ_STATUS.B0_RECALL. 0 - прерывание запрещено;
14	B1_RECALL	ЧТ/ЗП	0	1 - разрешение генерации прерывания при установке статуса FMC_IRQ_STATUS.B1_RECALL. 0 - прерывание запрещено;
15	OM_FINISH	ЧТ/ЗП	0	1 - разрешение генерации прерывания при установке статуса FMC_IRQ_STATUS.OM_FINISH. 0 - прерывание запрещено;

1.3.5.1.4.5.5 FMC_IRQ_TEST (0x010)

Регистр для тестирования прерываний контроллера FMC. В таблице 1.48 представлено описание полей регистра FMC_IRQ_TEST.

Таблица 1.48 – Описание полей регистра FMC_IRQ_TEST

Биты	Поле	Доступ	Значение после сброса	Описание
0	WRITE_CMPL	ЗП	0	Запись: 1 - установит в 1 статус FMC_IRQ_STATUS.WRITE_CMPL. 0 - не имеет действия;
1	ERASE_CMPL	ЗП	0	Запись: 1 - установит в 1 статус FMC_IRQ_STATUS.ERASE_CMPL. 0 - не имеет действия;
2	M_ERASE_CMPL	ЗП	0	Запись: 1 - установит в 1 статус FMC_IRQ_STATUS.M_ERASE_CMPL. 0 - не имеет действия;
3	RDP_ERR	ЗП	0	Запись: 1 - установит в 1 статус FMC_IRQ_STATUS.RDP_ERR. 0 - не имеет действия;
4	WRP_ERR	ЗП	0	Запись: 1 - установит в 1 статус FMC_IRQ_STATUS.WRP_ERR. 0 - не имеет действия;
5	ERP_ERR	ЗП	0	Запись: 1 - установит в 1 статус FMC_IRQ_STATUS.ERP_ERR. 0 - не имеет действия;

Биты	Поле	Доступ	Значение после сброса	Описание
6	MERP_ERR	ЗП	0	Запись: 1 - установит в 1 статус FMC_IRQ_STATUS.MERP_ERR. 0 - не имеет действия;
7	ECC_SEC	ЗП	0	Запись: 1 - установит в 1 статус FMC_IRQ_STATUS.ECC_SEC. 0 - не имеет действия;
8	ECC_DED	ЗП	0	Запись: 1 - установит в 1 статус FMC_IRQ_STATUS.ECC_DED. 0 - не имеет действия;
9	B0_POWER_ON	ЗП	0	Запись: 1 - установит в 1 статус FMC_IRQ_STATUS.B0_POWER_ON. 0 - не имеет действия;
10	B1_POWER_ON	ЗП	0	Запись: 1 - установит в 1 статус FMC_IRQ_STATUS.B1_POWER_ON. 0 - не имеет действия;
11	B0_WAKEUP	ЗП	0	Запись: 1 - установит в 1 статус FMC_IRQ_STATUS.B0_WAKEUP. 0 - не имеет действия;
12	B1_WAKEUP	ЗП	0	Запись: 1 - установит в 1 статус FMC_IRQ_STATUS.B1_WAKEUP. 0 - не имеет действия;
13	B0_RECALL	ЗП	0	Запись: 1 - установит в 1 статус FMC_IRQ_STATUS.B0_RECALL. 0 - не имеет действия;
14	B1_RECALL	ЗП	0	Запись: 1 - установит в 1 статус FMC_IRQ_STATUS.B1_RECALL. 0 - не имеет действия;
15	OM_FINISH	ЗП	0	Запись: 1 - установит в 1 статус FMC_IRQ_STATUS.OM_FINISH. 0 - не имеет действия;

1.3.5.1.4.5.6 FMC_IRQ_CLR (0x014)

Регистр очистки статуса прерываний контроллера FMC. В таблице 1.49 представлено описание полей регистра FMC_IRQ_CLR.

Таблица 1.49 – Описание полей регистра FMC_IRQ_CLR

Биты	Поле	Доступ	Значение после сброса	Описание
0	WRITE_CMPL	ЗП	0	Запись: 1 - сбросит в 0 статус FMC_IRQ_STATUS.WRITE_CMPL. 0 - не имеет действия;
1	ERASE_CMPL	ЗП	0	Запись: 1 - сбросит в 0 статус FMC_IRQ_STATUS.ERASE_CMPL. 0 - не имеет действия;
2	M_ERASE_CMPL	ЗП	0	Запись: 1 - сбросит в 0 статус FMC_IRQ_STATUS.M_ERASE_CMPL. 0 - не имеет действия;

Биты	Поле	Доступ	Значение после сброса	Описание
3	RDP_ERR	ЗП	0	Запись: 1 - сбросит в 0 статус FMC_IRQ_STATUS.RDP_ERR. 0 - не имеет действия;
4	WRP_ERR	ЗП	0	Запись: 1 - сбросит в 0 статус FMC_IRQ_STATUS.WRP_ERR. 0 - не имеет действия;
5	ERP_ERR	ЗП	0	Запись: 1 - сбросит в 0 статус FMC_IRQ_STATUS.ERP_ERR. 0 - не имеет действия;
6	MERP_ERR	ЗП	0	Запись: 1 - сбросит в 0 статус FMC_IRQ_STATUS.MERP_ERR. 0 - не имеет действия;
7	ECC_SEC	ЗП	0	Запись: 1 - сбросит в 0 статус FMC_IRQ_STATUS.ECC_SEC. 0 - не имеет действия;
8	ECC_DED	ЗП	0	Запись: 1 - сбросит в 0 статус FMC_IRQ_STATUS.ECC_DED. 0 - не имеет действия;
9	B0_POWER_ON	ЗП	0	Запись: 1 - сбросит в 0 статус FMC_IRQ_STATUS.B0_POWER_ON. 0 - не имеет действия;
10	B1_POWER_ON	ЗП	0	Запись: 1 - сбросит в 0 статус FMC_IRQ_STATUS.B1_POWER_ON. 0 - не имеет действия;
11	B0_WAKEUP	ЗП	0	Запись: 1 - сбросит в 0 статус FMC_IRQ_STATUS.B0_WAKEUP. 0 - не имеет действия;
12	B1_WAKEUP	ЗП	0	Запись: 1 - сбросит в 0 статус FMC_IRQ_STATUS.B1_WAKEUP. 0 - не имеет действия;
13	B0_RECALL	ЗП	0	Запись: 1 - сбросит в 0 статус FMC_IRQ_STATUS.B0_RECALL. 0 - не имеет действия;
14	B1_RECALL	ЗП	0	Запись: 1 - сбросит в 0 статус FMC_IRQ_STATUS.B1_RECALL. 0 - не имеет действия;
15	OM_FINISH	ЗП	0	Запись: 1 - сбросит в 0 статус FMC_IRQ_STATUS.OM_FINISH. 0 - не имеет действия;

1.3.5.1.4.5.7 FMC_CFG (0x018)

Регистр настроек контроллера FMC. В таблице 1.50 представлено описание полей регистра FMC_CFG.

Таблица 1.50 – Описание полей регистра FMC_CFG

Биты	Поле	До-ступ	Значение после сброса	Описание
1:0	WRP_BLK	ЧТ/ЗП	0	Реакция контроллера при обнаружении записи или стирания по защищенному адресу: 0 - немедленное завершение обращения, установка статуса; 1 - дополнительная блокировка всех записей или стираний в этот банк (аппаратная установка FMC_PROTECT_CFG_CMN.WRP_0 или FMC_PROTECT_CFG_CMN.WRP_1)
3:2	RDP_BLK	ЧТ/ЗП	0	Реакция контроллера при обнаружении чтения из защищенного адреса: 0 - немедленное завершение обращения, установка статуса; 1 - дополнительная блокировка всех чтений из этого банка (аппаратная установка FMC_PROTECT_CFG_CMN.WRP_0 или FMC_PROTECT_CFG_CMN.WRP_1)
4	BANK_SWAP	ЧТ/ЗП	0	1 - включение перестановки банков (см. 1.3.5.1.2.7); 0 - исходное расположение банков.
5	PREF_EN	ЧТ/ЗП	0	1 - включение предвыборки чтения (см. 1.3.5.1.2.3.2); 0 - выключено.
6	A_CLR_RBUF	ЧТ/ЗП	1	1 - включение автоматической очистки буфера чтения (см. 1.3.5.1.2.3.1); 0 - выключено.
7	WRITE_MODE	ЧТ/ЗП	0	1 - включение режима блочной записи (см. 1.3.5.1.2.4.2); 0 - выключено.
15:8	FREQUENCY	ЧТ/ЗП	0x10	Частота работы контроллера в МГц, округлённая до единиц в большую сторону (см. 1.3.5.1.2.2). **Минимальное значение 2 МГц**. Значения 0 и 1 - запрещены
19:16	LATENCY	ЧТ/ЗП	0x0	Задержка чтения данных из контроллера: 0 - 1 такт, 1 - 2 такта и т.д.
21:20	B0_POWER_STATE	ЧТ		Режим энергопотребления банка 0 (см. 1.3.5.1.3): Чтение: 0 - текущий режим - POWER_DOWN; 1 - текущий режим - SLEEP; 2 - текущий режим - STANDBY; 3 - текущий режим - RUN.
23:22			0	Зарезервировано
25:24	B1_POWER_STATE	ЧТ		Режим энергопотребления банка 1 (см. 1.3.5.1.3): Чтение: 0 - текущий режим - POWER_DOWN; 1 - текущий режим - SLEEP; 2 - текущий режим - STANDBY; 3 - текущий режим - RUN.
27:26			0	Зарезервировано

Биты	Поле	До-ступ	Значение после сброса	Описание
28	A_ER_PR	ЧТ/ЗП	0	Автоматическая защита от записей и чтений из банка, исполняющего общее или постраничное стирание (см. 1.3.5.1.2.5.3): 1 - включена; 0 - выключена
29	ECC_ENC	ЧТ/ЗП	1	Разрешение записи ECC во флэш память: 1 - включена; 0 - выключена
30	ECC_DEC	ЧТ/ЗП	1	Разрешение проверки ECC при чтении из флэш памяти: 1 - включена; 0 - выключена
31	ECC_HRESP	ЧТ/ЗП	0	Выдача ошибки системной шины при обнаружении двойной ошибки ECC DED: 1 - включена; 0 - выключена

1.3.5.1.4.5.8 FMC_CONTROL (0x01C)

Регистр управления контроллера FMC. В таблице 1.51 представлено описание полей регистра FMC_CONTROL.

Таблица 1.51 – Описание полей регистра FMC_CONTROL

Биты	Поле	До-ступ	Значение после сброса	Описание
0	ERASE	ЧТ/ЗП	0	запись: 1 - запуск процедуры стирания страниц с номерами от FMC_ERASE_ADR_L.PAGE_NUM до FMC_ERASE_ADR_H.PAGE_NUM включительно. Требуется одновременная запись соответствующего ключа в поле KEY; 0 - не имеет действия; чтение: 1 - процедура исполняется; 0 - процедура не исполняется.
1	MERASE0	ЧТ/ЗП	0	запись: 1 - запуск процедуры общего стирания основной памяти банка 0. Требуется одновременная запись соответствующего ключа в поле KEY; 0 - не имеет действия; чтение: 1 - процедура исполняется; 0 - процедура не исполняется.

Биты	Поле	Доступ	Значение после сброса	Описание
2	MERASE1	ЧТ/ЗП	0	запись: 1 - запуск процедуры общего стирания основной памяти банка 1. Требуется одновременная запись соответствующего ключа в поле KEY; 0 - не имеет действия; чтение: 1 - процедура исполняется; 0 - процедура не исполняется.
3	ABORT	ЧТ/ЗП	0	запись: 1 - запуск процедуры прекращения исполняемых процедур (см. 1.3.5.1.2.11). Требуется одновременная запись соответствующего ключа в поле KEY; 0 - не имеет действия; чтение: 1 - процедура исполняется; 0 - процедура не исполняется.
4	FWRITE	ЗП	0	запись: 1 - принудительный запуск процедуры записи во флэш память с текущим содержимым буфера (см. 1.3.5.1.2.4.1). Требуется одновременная запись соответствующего ключа в поле KEY; 0 - не имеет действия;
5	BLOCK_FWRITE	ЗП	0	запись: 1 - принудительный запуск процедуры блочной записи во флэш память (см. 1.3.5.1.2.4.2). Требуется одновременная запись соответствующего ключа в поле KEY; 0 - не имеет действия;
6	CLR_RBUF	ЗП	0	запись: 1 - очистка буфера чтения (см. 1.3.5.1.2.3.1). Требуется одновременная запись соответствующего ключа в поле KEY; 0 - не имеет действия;
7	OM_RUN	ЧТ/ЗП	0	запись: 1 - перезапуск инициализации конфигураций OptMem. Требуется одновременная запись соответствующего ключа в поле KEY; 0 - не имеет действия; чтение: 1 - процедура исполняется; 0 - процедура не исполняется.
31:16	KEY	ЗП	0	запись: 0x3A01 - ключ для команд ERASE, ABORT, FWRITE, BLOCK_FWRITE, CLR_RBUF, OM_RUN; 0x3B02 - ключ для команд MERASE0, MERASE1; другие значения - запись в регистр будет проигнорирована.

1.3.5.1.4.5.9 FMC_ERASE_ADR_L (0x020)

Регистр содержит адрес начальной страницы для процедуры постраничного стирания.

Описание процедуры постраничного стирания приведено в 1.3.5.1.2.5.1.

В таблице 1.52 представлено описание полей регистра FMC_ERASE_ADR_L.

Таблица 1.52 – Описание полей регистра FMC_ERASE_ADR_L

Биты	Поле	Доступ	Значение после сброса	Описание
20:13	PAGE_NUM	ЧТ/ЗП	0	Адрес начальной страницы для процедуры постраничного стирания. Аппаратно инкрементируется, пока не станет равен FMC_ERASE_ADR_H.PAGE_NUM и соответствует адресу текущей стираемой страницы. Соответствует системной карте памяти (т.е. при перестановке банков будет соответствовать другим физическим страницам).

1.3.5.1.4.5.10 FMC_ERASE_ADR_H (0x024)

Регистр содержит адрес последней страницы для процедуры постраничного стирания. Стирание производится включая эту страницу.

В таблице 1.53 представлено описание полей регистра FMC_ERASE_ADR_H.

Таблица 1.53 – Описание полей регистра FMC_ERASE_ADR_H

Биты	Поле	Доступ	Значение после сброса	Описание
20:13	PAGE_NUM	ЧТ/ЗП	0	Адрес последней страницы для процедуры постраничного стирания. Соответствует системной карте памяти (т.е. при перестановке банков будет соответствовать другим физическим страницам). Устанавливаемое сюда значение должно быть больше или равно значению FMC_ERASE_ADR_L.PAGE_NUM.

1.3.5.1.4.5.11 FMC_DBG_STATUS (0x028)

Регистр отображает текущие состояния контроллера FMC. Доступен только на чтение. Запись в любой из битов не имеет действия.

В таблице 1.54 представлено описание полей регистра FMC_DBG_STATUS.

Таблица 1.54 – Описание полей регистра FMC_DBG_STATUS

Биты	Поле	Доступ	Значение после сброса	Описание
0	BANK0_BUSY	ЧТ	0	1 - банк 0 занят
1	BANK1_BUSY	ЧТ	0	1 - банк 1 занят
2	WBUF_EMPTY	ЧТ	1	1 - буфер записи свободен
3	RBUF_EMPTY	ЧТ	1	1 - буфер чтения свободен
4	WRITE_EXEC	ЧТ	0	1 - исполняется запись во флэш память

Биты	Поле	До-ступ	Значение после сброса	Описание
5	FLASH_INIT	ЧТ	0	1 - окончание инициализации флэш памяти. Устанавливается если были установлены биты B0_ENABLE, B1_ENABLE, B0_RECALL, B1_RECALL регистра FMC_IRQ_STATUS. Сбрасывается только аппаратным сбросом FMC.
7:6	AUTO_POWER_ON	ЧТ		Значение входного конфигурационного сигнала auto_power_on_i
8	OM_FIRST_ENABLE	ЧТ		Значение входного конфигурационного сигнала optmem_first_enable
9	B0_PSTATE_REQ_ON	ЧТ		Запрошенный режим энергопотребления банка 0
10	B1_PSTATE_REQ_ON	ЧТ		Запрошенный режим энергопотребления банка 1
12:11	AUTO_START_ON	ЧТ		бит [11]: 1 - автоматическое чтение RECALL при включении банка 0 разрешено; 0 - запрещено; бит [12]: 1 - автоматическое чтение RECALL при включении банка 1 разрешено; 0 - запрещено
13	WRITE_UNCMPL	ЧТ	0	1 - присутствует незавершенная транзакция записи (WRITE_EXEC

1.3.5.1.4.5.12 FMC_ECC (0x02C)

Регистр указывает адрес, при чтении по которому произошла SEC или DED ошибка ECC. Регистр блокируется до тех пор, пока оба поля FMC_IRQ_STATUS.ECC_SEC и FMC_IRQ_STATUS.ECC_DED не будут сброшены в 0.

В таблице 1.55 представлено описание полей регистра FMC_ECC.

Таблица 1.55 – Описание полей регистра FMC_ECC

Биты	Поле	До-ступ	Значение после сброса	Описание
18:0	ADDR	ЧТ	0	Младшая часть адреса
19	BANK_NUM	ЧТ	0	Абсолютный номер банка (без учета перестановки банков)
20	MEM_TYPE	ЧТ	0	Тип памяти: 0 - основная память; 1 - информационная.

1.3.5.1.4.5.13 FMC_PROTECT_CFG_CMN (0x030)

Регистр защиты основных областей и страниц информационных областей eFlash-памяти от записи и стирания.

Для каждого бита регистра:

1 - защита включена;

0 - защита выключена.

В таблице 1.56 представлено описание полей регистра FMC_PROTECT_CFG_CMN.

Таблица 1.56 – Описание полей регистра FMC_PROTECT_CFG_CMN

Биты	Поле	До-ступ	Значение после сброса	Описание
0	WRP_0	ЧТ/ЗП	0	Защита банка 0 от записей и стираний. Может устанавливаться автоматически.
1	WRP_1	ЧТ/ЗП	0	Защита банка 1 от записей и стираний. Может устанавливаться автоматически.
2	RDP_0	ЧТ/ЗП	0	Защита банка 0 от чтений. Может устанавливаться автоматически.
3	RDP_1	ЧТ/ЗП	0	Защита банка 1 от чтений. Может устанавливаться автоматически.
4	WRP_IP0	ЧТ/ЗП	0	Защита от записи и стираний информационной страницы 0 банка 0
5	WRP_IP1	ЧТ/ЗП	0	Защита от записи и стираний информационной страницы 1 банка 0
6	WRP_IP2	ЧТ/ЗП	0	Защита от записи и стираний информационной страницы 2 банка 0
7	WRP_IP3	ЧТ/ЗП	0	Защита от записи и стираний информационной страницы 3 банка 0
8	WRP_IP4	ЧТ/ЗП	0	Защита от записи и стираний информационной страницы 0 банка 1
9	WRP_IP5	ЧТ/ЗП	0	Защита от записи и стираний информационной страницы 1 банка 1
10	WRP_IP6	ЧТ/ЗП	0	Защита от записи и стираний информационной страницы 2 банка 1
11	WRP_IP7	ЧТ/ЗП	0	Защита от записи и стираний информационной страницы 3 банка 1
12	RDP_IP0	ЧТ/ЗП	0	Защита от чтений информационной страницы 0 банка 0
13	RDP_IP1	ЧТ/ЗП	0	Защита от чтений информационной страницы 1 банка 0
14	RDP_IP2	ЧТ/ЗП	0	Защита от чтений информационной страницы 2 банка 0
15	RDP_IP3	ЧТ/ЗП	0	Защита от чтений информационной страницы 3 банка 0
16	RDP_IP4	ЧТ/ЗП	0	Защита от чтений информационной страницы 0 банка 1
17	RDP_IP5	ЧТ/ЗП	0	Защита от чтений информационной страницы 1 банка 1
18	RDP_IP6	ЧТ/ЗП	0	Защита от чтений информационной страницы 2 банка 1
19	RDP_IP7	ЧТ/ЗП	0	Защита от чтений информационной страницы 3 банка 1
30:20			0	Зарезервировано

Биты	Поле	До-ступ	Значение после сброса	Описание
31	PROT_BLK_RST	ЧТ/ЗП	0	Запрет программного изменения настроек защиты от записи и чтения (см. 1.3.5.1.2.8.1). Запись: 1 - запрет включен; 0 - не имеет действия. Чтение: 1 - запрет включен; 0 - запрет выключен.

1.3.5.1.4.5.14 FMC_WRP_CFG_0 (0x034)

Регистр настройки постраничной защиты от записи и стирания страниц 0-31 банка 0.

В таблице 1.57 представлено описание полей регистра FMC_WRP_CFG_0.

Таблица 1.57 – Описание полей регистра FMC_WRP_CFG_0

Биты	Поле	До-ступ	Значение после сброса	Описание
0	WRP_0_P0	ЧТ/ЗП	0	Защита от записи и стирания страницы 0 банка 0 1 - защита включена; 0 - выключена.
…				
31	WRP_0_P31	ЧТ/ЗП	0	Защита от записи и стирания страницы 31 банка 0 1 - защита включена; 0 - выключена.

1.3.5.1.4.5.15 FMC_WRP_CFG_1 (0x038)

Регистр настройки постраничной защиты от записи и стирания страниц 32-63 банка 0.

В таблице 1.58 представлено описание полей регистра FMC_WRP_CFG_1.

Таблица 1.58 – Описание полей регистра FMC_WRP_CFG_1

Биты	Поле	До-ступ	Значение после сброса	Описание
0	WRP_0_P32	ЧТ/ЗП	0	Защита от записи и стирания страницы 32 банка 0 1 - защита включена; 0 - выключена.
…				
31	WRP_0_P63	ЧТ/ЗП	0	Защита от записи и стирания страницы 63 банка 0 1 - защита включена; 0 - выключена.

1.3.5.1.4.5.16 FMC_WRP_CFG_2 (0x03C)

Регистр настройки постраничной защиты от записи и стирания страниц 0-31 банка 1.

В таблице 1.59 представлено описание полей регистра FMC_WRP_CFG_2.

Таблица 1.59 – Описание полей регистра FMC_WRP_CFG_2

Биты	Поле	До-ступ	Значение после сброса	Описание
0	WRP_1_P0	ЧТ/ЗП	0	Защита от записи и стирания страницы 0 банка 1 1 - защита включена; 0 - выключена.
…				
31	WRP_1_P31	ЧТ/ЗП	0	Защита от записи и стирания страницы 31 банка 1 1 - защита включена; 0 - выключена.

1.3.5.1.4.5.17 FMC_WRP_CFG_3 (0x040)

Регистр настройки постраничной защиты от записи и стирания страниц 32-63 банка 1.

В таблице 1.60 представлено описание полей регистра FMC_WRP_CFG_3.

Таблица 1.60 – Описание полей регистра FMC_WRP_CFG_3

Биты	Поле	До-ступ	Значение после сброса	Описание
0	WRP_1_P32	ЧТ/ЗП	0	Защита от записи и стирания страницы 32 банка 1 1 - защита включена; 0 - выключена.
…				
31	WRP_1_P63	ЧТ/ЗП	0	Защита от записи и стирания страницы 63 банка 1 1 - защита включена; 0 - выключена.

1.3.5.1.4.5.18 FMC_RDP_CFG_0 (0x044)

Регистр настройки постраничной защиты от чтения страниц 0-31 банка 0.

В таблице 1.61 представлено описание полей регистра FMC_RDP_CFG_0.

Таблица 1.61 – Описание полей регистра FMC_RDP_CFG_0

Биты	Поле	До-ступ	Значение после сброса	Описание
0	RDP_0_P0	ЧТ/ЗП	0	Защита от чтения страницы 0 банка 0 1 - защита включена; 0 - выключена.
…				
31	RDP_0_P31	ЧТ/ЗП	0	Защита от чтения страницы 31 банка 0 1 - защита включена; 0 - выключена.

1.3.5.1.4.5.19 FMC_RDP_CFG_1 (0x048)

Регистр настройки постраничной защиты от чтения страниц 32-63 банка 0.

В таблице 1.62 представлено описание полей регистра FMC_RDP_CFG_1.

Таблица 1.62 – Описание полей регистра FMC_RDP_CFG_1

Биты	Поле	До-ступ	Значение после сброса	Описание
0	RDP_0_P32	ЧТ/ЗП	0	Защита от чтения страницы 32 банка 0 1 - защита включена; 0 - выключена.
…				
31	RDP_0_P63	ЧТ/ЗП	0	Защита от чтения страницы 63 банка 0 1 - защита включена; 0 - выключена.

1.3.5.1.4.5.20 FMC_RDP_CFG_2 (0x04C)

Регистр настройки постраничной защиты от чтения страниц 0-31 банка 1.

В таблице 1.63 представлено описание полей регистра FMC_RDP_CFG_2.

Таблица 1.63 – Описание полей регистра FMC_RDP_CFG_2

Биты	Поле	До-ступ	Значение после сброса	Описание
0	RDP_1_P0	ЧТ/ЗП	0	Защита от чтения страницы 0 банка 1 1 - защита включена; 0 - выключена.
…				
31	RDP_1_P31	ЧТ/ЗП	0	Защита от чтения страницы 31 банка 1 1 - защита включена; 0 - выключена.

1.3.5.1.4.5.21 FMC_RDP_CFG_3 (0x050)

Регистр настройки постраничной защиты от чтения страниц 32-63 банка 1.

В таблице 1.64 представлено описание полей регистра FMC_RDP_CFG_3.

Таблица 1.64 – Описание полей регистра FMC_RDP_CFG_3

Биты	Поле	До-ступ	Значение после сброса	Описание
0	RDP_1_P32	ЧТ/ЗП	0	Защита от чтения страницы 32 банка 1 1 - защита включена; 0 - выключена.
…				
31	RDP_1_P63	ЧТ/ЗП	0	Защита от чтения страницы 63 банка 1 1 - защита включена; 0 - выключена.

1.3.5.1.4.5.22 FMC_SECURITY_CFG_CMN (0x054)

Регистр настроек безопасности для информационных областей eFlash-памяти.

Для каждого бита регистра:

– 1 - только безопасный доступ к странице;

– 0 - любой доступ к странице.

Адреса в таблице указаны без учета базового смещения, которое имеет разные значения для секьюрного и несекьюрного доступа.

В таблице 1.65 представлено описание полей регистра FMC_SECURITY_CFG_CMN.

Таблица 1.65 – Описание полей регистра FMC_SECURITY_CFG_CMN

Биты	Поле	До-ступ	Значение после сброса	Описание
0	SEC_IP0	ЧТ/ЗП	0	Секьюрность информационной страницы 0 (т.е. адресов 0x100000..0x101FFF)
1	SEC_IP1	ЧТ/ЗП	0	Секьюрность информационной страницы 1 (т.е. адресов 0x102000..0x103FFF)
2	SEC_IP2	ЧТ/ЗП	0	Секьюрность информационной страницы 2 (т.е. адресов 0x104000..0x105FFF)
3	SEC_IP3	ЧТ/ЗП	0	Секьюрность информационной страницы 3 (т.е. адресов 0x106000..0x107FFF)
4	SEC_IP4	ЧТ/ЗП	0	Секьюрность информационной страницы 4 (т.е. адресов 0x108000..0x109FFF)
5	SEC_IP5	ЧТ/ЗП	0	Секьюрность информационной страницы 5 (т.е. адресов 0x10A000..0x10BFFF)
6	SEC_IP6	ЧТ/ЗП	0	Секьюрность информационной страницы 6 (т.е. адресов 0x10C000..0x10DFFF)
7	SEC_IP7	ЧТ/ЗП	0	Секьюрность информационной страницы 7 (т.е. адресов 0x10E000..0x10FFFF)
31:8			0	Зарезервировано

1.3.5.1.4.5.23 FMC_SECURITY_CFG_0 (0x058)

Регистр настроек постраничной безопасности для страниц 0 - 31.

Для каждого бита регистра:

– 1 - только безопасный доступ к странице;

– 0 - любой доступ к странице.

Адреса в таблице указаны без учета базового смещения, которое имеет разные значения для секьюрного и несекьюрного доступа.

В таблице 1.66 представлено описание полей регистра FMC_SECURITY_CFG_0.

Таблица 1.66 – Описание полей регистра FMC_SECURITY_CFG_0

Биты	Поле	До-ступ	Значение после сброса	Описание
0	SEC_P0	ЧТ/ЗП	0	Секьюрность страницы 0 (т.е. адресов 0x000000..0x001FFF)
1	SEC_P1	ЧТ/ЗП	0	Секьюрность страницы 1 (т.е. адресов 0x002000..0x003FFF)
…				
31	SEC_P31	ЧТ/ЗП	0	Секьюрность страницы 31 (т.е. адресов 0x03E000..0x03FFFF)

1.3.5.1.4.5.24 FMC_SECURITY_CFG_1 (0x05C)

Регистр настроек постраничной безопасности для страниц 32 - 63.

Для каждого бита регистра:

– 1 - только безопасный доступ к странице;

– 0 - любой доступ к странице.

Адреса в таблице указаны без учета базового смещения, которое имеет разные значения для секьюрного и несекьюрного доступа.

В таблице 1.67 представлено описание полей регистра FMC_SECURITY_CFG_1.

Таблица 1.67 – Описание полей регистра FMC_SECURITY_CFG_1

Биты	Поле	До-ступ	Значение после сброса	Описание
0	SEC_P32	ЧТ/ЗП	0	Секьюрность страницы 32 (т.е. адресов 0x040000..0x081FFF)
1	SEC_P33	ЧТ/ЗП	0	Секьюрность страницы 33 (т.е. адресов 0x042000..0x083FFF)
…				
31	SEC_P63	ЧТ/ЗП	0	Секьюрность страницы 63 (т.е. адресов 0x07E000..0x07FFFF)

1.3.5.1.4.5.25 FMC_SECURITY_CFG_2 (0x060)

Регистр настроек постраничной безопасности для страниц 64 - 195.

Для каждого бита регистра:

– 1 - только безопасный доступ к странице;

– 0 - любой доступ к странице.

Адреса в таблице указаны без учета базового смещения, которое имеет разные значения для секьюрного и несекьюрного доступа.

В таблице 1.68 представлено описание полей регистра FMC_SECURITY_CFG_2.

Таблица 1.68 – Описание полей регистра FMC_SECURITY_CFG_2

Биты	Поле	До-ступ	Значение после сброса	Описание
0	SEC_P64	ЧТ/ЗП	0	Секьюрность страницы 64 (т.е. адресов 0x080000..0x081FFF)
1	SEC_P65	ЧТ/ЗП	0	Секьюрность страницы 65 (т.е. адресов 0x082000..0x083FFF)
…				
31	SEC_P95	ЧТ/ЗП	0	Секьюрность страницы 95 (т.е. адресов 0x0BE000..0x0BFFFF)

1.3.5.1.4.5.26 FMC_SECURITY_CFG_3 (0x064)

Регистр настроек постраничной безопасности для страниц 96 - 127.

Для каждого бита регистра:

– 1 - только безопасный доступ к странице;

– 0 - любой доступ к странице.

Адреса в таблице указаны без учета базового смещения, которое имеет разные значения для секьюрного и несекьюрного доступа.

В таблице 1.69 представлено описание полей регистра FMC_SECURITY_CFG_3.

Таблица 1.69 – Описание полей регистра FMC_SECURITY_CFG_3

Биты	Поле	До-ступ	Значение после сброса	Описание
0	SEC_P96	ЧТ/ЗП	0	Секьюрность страницы 96 (т.е. адресов 0x0C0000..0x0C1FFF)
1	SEC_P97	ЧТ/ЗП	0	Секьюрность страницы 97 (т.е. адресов 0x0C2000..0x0C3FFF)
…				
31	SEC_P127	ЧТ/ЗП	0	Секьюрность страницы 127 (т.е. адресов 0x0FE000..0x0FFFFF)

1.3.5.1.4.5.27 FMC_SAFETY_IRQ (0x068)

Регистр статуса прерываний по нарушению безопасности.

В таблице 1.70 представлено описание полей регистра FMC_SAFETY_IRQ.

Таблица 1.70 – Описание полей регистра FMC_SAFETY_IRQ

Биты	Поле	До-ступ	Значение после сброса	Описание
0	SEC_ERR	ЧТ/ЗП	0	Суммарный статус ошибки. Устанавливается в 1, если устанавливается хотя бы одно из других полей этого регистра. Чтение: показывает статус ошибки безопасности. Устанавливается в 1 при: - запросе записи/чтения/постраничного или общего стирания с настройками секьюрности, меньшими, чем у адресуемой страницы; - программном доступе к регистру с бОльшими настройками секьюрности, чем у запроса. Запись: 0 - не имеет действия; 1 - сбросит все поля этого регистра;
1	W_ERR	ЧТ	0	Чтение: 1 - обнаружена ошибка безопасности при процедуре записи во флэш память; 0 - не было ошибки; Запись - не имеет действия. Для сброса надо установить FMC_SAFETY_IRQ.IRQ.
2	R_ERR	ЧТ	0	Чтение: 1 - обнаружена ошибка безопасности при процедуре чтения из флэш памяти; 0 - не было ошибки; Запись - не имеет действия. Для сброса надо установить FMC_SAFETY_IRQ.IRQ.
3	E_ERR	ЧТ	0	Чтение: 1 - обнаружена ошибка безопасности при процедуре постраничного стирания флэш памяти; 0 - не было ошибки; Запись - не имеет действия. Для сброса надо установить FMC_SAFETY_IRQ.IRQ.
4	ME0_ERR	ЧТ	0	Чтение: 1 - обнаружена ошибка безопасности при процедуре общего стирания банка 0 флэш памяти; 0 - не было ошибки; Запись - не имеет действия. Для сброса надо установить FMC_SAFETY_IRQ.IRQ.
5	ME1_ERR	ЧТ	0	Чтение: 1 - обнаружена ошибка безопасности при процедуре общего стирания банка 1 флэш памяти; 0 - не было ошибки; Запись - не имеет действия. Для сброса надо установить FMC_SAFETY_IRQ.IRQ.
6	APB_ERR	ЧТ	0	Чтение: 1 - обнаружена ошибка безопасности при доступе к регистрам; 0 - не было ошибки; Запись - не имеет действия. Для сброса надо установить FMC_SAFETY_IRQ.IRQ.

Биты	Поле	До-ступ	Значение после сброса	Описание
31:7			0	Зарезервировано

1.3.5.1.4.5.28 FMC_SAFETY_MASK (0x06C)

Регистр маски прерываний по ошибкам безовасности.

В таблице 1.71 представлено описание полей регистра FMC_SAFETY_MASK.

Таблица 1.71 – Описание полей регистра FMC_SAFETY_MASK

Биты	Поле	До-ступ	Значение после сброса	Описание
0	MASK	ЧТ/ЗП	0	1 - разрешение генерации прерывания при ошибке безопасности. 0 - прерывание запрещено;
31:1			0	Зарезервировано

1.3.5.1.4.5.29 FMC_FLASH_SETTINGS_KEY (0x200)

Запись в регистр FMC_FLASH_SETTINGS_KEY значения 0x7375646F разрешает запись во все остальные регистры контроллера FMC, отвечающие за настройку макросов eFlash памяти, запись любого другого значения - запрещает. Если запись в регистры контроллера FMC, отвечающие за настройку макросов eFlash памяти, разрешена, то из регистра FMC_FLASH_SETTINGS_KEY читается 0x1.

В таблицах 1.72 и 1.73 представлено описание полей регистра FMC_FLASH_SETTINGS_KEY при операциях записи и чтения, соответственно.

Таблица 1.72 – Описание полей регистра FMC_FLASH_SETTINGS_KEY при записи

Биты	Поле	До-ступ	Значение после сброса	Описание
31:0	KEY	ЗП	0	Ключ для записи в регистры контроллера FMC, отвечающие за настройку макросов eFlash памяти

Таблица 1.73 – Описание полей регистра FMC_FLASH_SETTINGS_KEY при чтении

Поле	Название	Программный доступ	Состояние после сброса	Описание
0	WR_PERMISSION	ЧТ	0	Флаг разрашения записи
31:1		ЧТ	0	Зарезервировано

1.3.5.1.4.5.30 FMC_FLASH_SETTINGS (0x204)

Регистр настроек алгоритмов доступа к eFlash памяти контроллера FMC.

В таблице 1.74 представлено описание полей регистра FMC_FLASH_SETTINGS.

Таблица 1.74 – Описание полей регистра FMC_FLASH_SETTINGS

Поле	Название	Программный доступ	Состояние после сброса	Описание
0	USE_SMW	ЧТ/ЗП	0x1	Использовать смарт-операции для записи, стирания страницы и стирания банка eFlash памяти
3:1	-	ЧТ	0	Зарезервировано
4	SMW_PV_READ	ЧТ/ЗП	0	Включение режима дозаписи данных (битовые ячейки, зодержащие 0 не записываются)
7:5	-	ЧТ	0	Зарезервировано
8	SMW_HEM_CTRL	ЧТ/ЗП	0	Включение режима программного выбора алгоритма смарт-операции записи
11:9	-	ЧТ	0	Зарезервировано
12	SMW_HEM	ЧТ/ЗП	0	Выбор алгоритма смарт-операции записи: 0 - алгоритм для битовых ячеек, выдерживающих 10 000 циклов перезаписи; 1 - алгоритм для битовых ячеек, выдерживающих 100 000 циклов перезаписи.
15:13	-	ЧТ	0	Зарезервировано
16	LOCK_TIMINGS	ЧТ/ЗП	0	Блокировка изменения настроек временных параметров (до аппаратного сброса контроллера FMC)
31:17	-	ЧТ	0	Зарезервировано

1.3.5.1.4.5.31 FMC_FLASH_GLOBAL_TIMINGS (0x208)

Регистр настроек временных параметров доступа к eFlash памяти контроллера FMC.

В таблице 1.75 представлено описание полей регистра FMC_FLASH_GLOBAL_TIMINGS.

Таблица 1.75 – Описание полей регистра FMC_FLASH_GLOBAL_TIMINGS

Поле	Название	Программный доступ	Состояние после сброса	Описание
7:0	Tus	ЧТ/ЗП	0x11	Длительность 1 мкс в тактах системной частоты
15:8	T200ns	ЧТ/ЗП	0x04	Длительность 200 нс в тактах системной частоты
23:16	T100ns	ЧТ/ЗП	0x02	Длительность 100 нс в тактах системной частоты
27:24	T20ns	ЧТ/ЗП	0x01	Длительность 20 нс в тактах системной частоты
31:28	T10ns	ЧТ/ЗП	0x01	Длительность 10 нс в тактах системной частоты

1.3.5.1.4.5.32 FMC_FLASH_READ_TIMINGS (0x210)

Регистр настроек временных параметров доступа к eFlash памяти контроллера FMC для операции чтения.

В таблице 1.76 представлено описание полей регистра FMC_FLASH_READ_TIMINGS.

Таблица 1.76 – Описание полей регистра FMC_FLASH_READ_TIMINGS

Поле	Название	Программный доступ	Состояние после сброса	Описание
3:0	Tpws	ЧТ/ЗП	0x1	Длительность активног оуровня сигнала SE в тактах системной частоты
7:4	Tcyc	ЧТ/ЗП	0x2	Длительность цикла чтения в тактах системной частоты
31:8	-	ЧТ	0	Зарезервировано

1.3.5.1.4.5.33 FMC_FLASH_RECALL_TIMINGS (0x214)

Регистр настроек временных параметров доступа к eFlash памяти контроллера FMC при операции чтения и применения подстроечных коэффициентов (Recall).

В таблице 1.77 представлено описание полей регистра FMC_FLASH_RECALL_TIMINGS.

Таблица 1.77 – Описание полей регистра FMC_FLASH_RECALL_TIMINGS

Поле	Название	Программный доступ	Состояние после сброса	Описание
7:0	Trcsr	ЧТ/ЗП	0x01	Задержка установки сигнала SE от начала операции Recall в тактах системной частоты
15:8	Trcsf	ЧТ/ЗП	0x03	Задержка снятия сигнала SE от начала операции Recall в тактах системной частоты
23:16	Trccf	ЧТ/ЗП	0x06	Задержка снятия сигнала RECALL от начала операции Recall в тактах системной частоты
31:24	Trccyc	ЧТ/ЗП	0x07	Длительность операции Recall в тактах системной частоты

1.3.5.1.4.5.34 FMC_SMW_TIMER_OPTINS (0x220)

Регистр настройки таймеров для смарт-операций записи, постраничного стирания и полного стирания основной области памяти одного банка.

В таблице 1.78 представлено описание полей регистра FMC_SMW_TIMER_OPTINS.

Таблица 1.78 – Описание полей регистра FMC_SMW_TIMER_OPTINS

Поле	Название	Программный доступ	Состояние после сброса	Описание
7:0	Tus	ЧТ/ЗП	0x10	Длительность 1 мкс в тактах системной частоты
12:8	Tmacc	ЧТ/ЗП	0x02	Длительность задержка Tmacc (> 90 нс) в тактах системной частотыю На основе этого параметр расчитываются другие задержки: Tmpws = Tmacc + 1; Tmnws = Tmacc + 2; Tmcyc = (Tmacc + 1) * 2.
31:13	-	ЧТ	0	Зарезервировано

1.3.5.1.4.5.35 FMC_SMW_SETTING_OPTIONS_H (0x224)

Регистр настроек смарт-операций записи, постраничного стирания и полного стирания основной области памяти одного банка.

В таблице 1.79 представлено описание полей регистра FMC_SMW_SETTING_OPTIONS_H.

Таблица 1.79 – Описание полей регистра FMC_SMW_SETTING_OPTIONS_H

Поле	Название	Программный доступ	Состояние после сброса	Описание
1:0	TERS_CTRL_2_1	ЧТ/ЗП	0x1	TERS_CTRL[33:32]
3:2	TPGM_CTRL	ЧТ/ЗП	0x1	TPGM_CTRL[35:34]
6:4	TNVS_CTRL	ЧТ/ЗП	0x1	TNVS_CTRL[38:36]
9:7	TNVH_CTRL	ЧТ/ЗП	0x6	TNVH_CTRL[41:39]
12:10	TPGS_CTRL	ЧТ/ЗП	0x1	TPGS_CTRL[44:42]
21:13	MAX_ERASE	ЧТ/ЗП	0x56	MAX_ERASE[53:45]
26:22	MAX_PROG	ЧТ/ЗП	0x5	MAX_PROG [58:54]
31:27	-	ЧТ	0	Зарезервировано

1.3.5.1.4.5.36 FMC_SMW_SETTING_OPTIONS_L (0x228)

Регистр настроек смарт-операций записи, постраничного стирания и полного стирания основной области памяти одного банка.

В таблице 1.80 представлено описание полей регистра FMC_SMW_SETTING_OPTIONS_L.

Таблица 1.80 – Описание полей регистра FMC_SMW_SETTING_OPTIONS_L

Поле	Название	Программный доступ	Состояние после сброса	Описание
13:0	-	ЧТ	0	Зарезервировано
16:14	MV	ЧТ/ЗП	0x5	MV_CTRL
19:17	MV_value	ЧТ/ЗП	0x5	MV_CTRL
20	-	ЧТ	0	Зарезервировано
23:21	MV_CTRL	ЧТ/ЗП	0x0	MV_CTRL
25:24	INITIAL_WIPGM	ЧТ/ЗП	0x3	IPGM_CTRL
27:26	FINAL_WIPGM	ЧТ/ЗП	0x3	IPGM_CTRL
30:28	-	ЧТ	0x0	Зарезервировано
31	TERS_CTRL_0	ЧТ/ЗП	0x0	TERS_CTRL[31]

1.3.5.1.4.5.37 FMC_SMW_SETTING_OPTIONS2 (0x22C)

Регистр настроек смарт-операций записи, постраничного стирания и полного стирания основной области памяти одного банка.

В таблице 1.81 представлено описание полей регистра FMC_SMW_SETTING_OPTIONS2.

Таблица 1.81 – Описание полей регистра FMC_SMW_SETTING_OPTIONS2

Поле	Название	Программный доступ	Состояние после сброса	Описание
2:0	THVS_CTRL	ЧТ/ЗП	0x1	THVS_CTRL
5:3	TRCV_CTRL	ЧТ/ЗП	0x2	TRCV_CTRL
9:6	EXTRA_PULSE	ЧТ/ЗП	0x5	EXTRA_PULSE
17:10	WHV_COUNTER	ЧТ/ЗП	0x?	WHV_COUNTER
20:18	POST_TERS	ЧТ/ЗП	0x2	POST_TERS
22:21	POST_TPGM	ЧТ/ЗП	0x1	POST_TPGM
24:23	VERIFY_OPTION	ЧТ/ЗП	0x1	VERIFY_OPTION
26:25	TPGM_OPTION	ЧТ/ЗП	0x0	TPGM_OPTION
27	MASK0_OPTION	ЧТ/ЗП	0x?	MASK0_OPTION
28	DISABLE_PRER_OPTION	ЧТ/ЗП	0x?	DISABLE_PRER_OPTION
31:29	-	ЧТ	0	Зарезервировано

1.3.5.1.4.5.38 FMC_SMW_SETTING_OPTIONS3 (0x230)

Регистр настроек смарт-операций записи, постраничного стирания и полного стирания основной области памяти одного банка.

В таблице 1.82 представлено описание полей регистра FMC_SMW_SETTING_OPTIONS3.

Таблица 1.82 – Описание полей регистра FMC_SMW_SETTING_OPTIONS3

Поле	Название	Программный доступ	Состояние после сброса	Описание
7:0	HEM_WHM_COUNTER	ЧТ/ЗП	0x0	HEM_WHM_COUNTER
16:8	HEM_MAX_ERASE	ЧТ/ЗП	0x1BD	HEM_MAX_ERASE
31:17	-	ЧТ	0	Зарезервировано

1.3.5.1.4.5.39 FMC_SMW_SMP_WHV_OPTIONS_H (0x234)

В регистре FMC_SMW_SMP_WHV_OPTIONS_H указывается ряд параметров напряжений для 8-15 итераций смарт-операции записи стирания. После 15-й итерации параметры напряжения не изменяются.

В таблице 1.83 представлено описание полей регистра FMC_SMW_SMP_WHV_OPTIONS_H.

Таблица 1.83 – Описание полей регистра FMC_SMW_SMP_WHV_OPTIONS_H

Поле	Название	Программный доступ	Состояние после сброса	Описание
31:0	SMW_SMP_WHV_OPTION	ЧТ	0x77777777	SMW_SMP_WHV_OPTION[63:32]

1.3.5.1.4.5.40 FMC_SMW_SMP_WHV_OPTIONS_L (0x238)

В регистре FMC_SMW_SMP_WHV_OPTIONS_L указывается ряд параметров напряжений для 0-7 итераций смарт-операции записи стирания.

В таблице 1.84 представлено описание полей регистра FMC_SMW_SMP_WHV_OPTIONS_L.

Таблица 1.84 – Описание полей регистра FMC_SMW_SMP_WHV_OPTIONS_L

Поле	Название	Программный доступ	Состояние после сброса	Описание
31:0	SMW_SMP_WHV_OPTION	ЧТ	0x77777765	SMW_SMP_WHV_OPTION[31:0]

1.3.5.1.4.5.41 FMC_SMW_SME_WHV_OPTIONS_H (0x23C)

В регистре FMC_SMW_SME_WHV_OPTIONS_H указывается ряд параметров напряжений для 8-15 итераций смарт-операции постраничного стирания. После 15-й итерации параметры напряжения не изменяются.

В таблице 1.85 представлено описание полей регистра FMC_SMW_SME_WHV_OPTIONS_H.

Таблица 1.85 – Описание полей регистра FMC_SMW_SME_WHV_OPTIONS_H

Поле	Название	Программный доступ	Состояние после сброса	Описание
31:0	SMW_SME_WHV_OPTION	ЧТ	0xCCCCCCCC	SMW_SME_WHV_OPTION[63:32]

1.3.5.1.4.5.42 FMC_SMW_SME_WHV_OPTIONS_L (0x240)

В регистре FMC_SMW_SME_WHV_OPTIONS_L указывается ряд параметров напряжений для 0-7 итераций смарт-операции постраничного стирания.

В таблице 1.86 представлено описание полей регистра FMC_SMW_SME_WHV_OPTIONS_L.

Таблица 1.86 – Описание полей регистра FMC_SMW_SME_WHV_OPTIONS_L

Поле	Название	Программный доступ	Состояние после сброса	Описание
31:0	SMW_SME_WHV_OPTION	ЧТ	0xCCBA9876	SMW_SME_WHV_OPTION[31:0]

1.3.5.1.4.5.43 FMC_SMW_DEBUG_STAT (0x244)

В регистре FMC_SMW_DEBUG_STAT указывается число итераций, выполненных во время смарт-операции записи или постраничного стирания и параметры последней итерации.

В таблице 1.87 представлено описание полей регистра FMC_SMW_DEBUG_STAT.

Таблица 1.87 – Описание полей регистра FMC_SMW_DEBUG_STAT

Поле	Название	Программный доступ	Состояние после сброса	Описание
8:0	SMW_LAST_SET	ЧТ/ЗП	0	
15:9	-	ЧТ	0	Зарезервировано
24:16	SMW_LOOP_NUM	ЧТ/ЗП	0	
31:25	-	ЧТ	0	Зарезервировано

1.3.5.1.4.5.44 FMC_FLASH_BANK_0_RED_INFO (0x250)

Регистр FMC_FLASH_BANK_0_RED_INFO содержит таблицу замен повреждённых страниц для банка 0 контроллера FMC.

В таблице 1.88 представлено описание полей регистра FMC_FLASH_BANK_0_RED_INFO.

Таблица 1.88 – Описание полей регистра FMC_FLASH_BANK_0_RED_INFO

Поле	Название	Программный доступ	Состояние после сброса	Описание
9:0	PAGE_0_REMAP_ADDR	ЧТ/ЗП	0x1	Адрес повреждённой страницы, заменяемой нулевой резервной страницей
14:10	-	ЧТ	0	Зарезервировано
15	PAGE_0_REMAP	ЧТ/ЗП	0x2	Флаг использования нулевой резервной страницей
25:16	PAGE_1_REMAP_ADDR	ЧТ/ЗП	0x1	Адрес повреждённой страницы, заменяемой первой резервной страницей
30:26	-	ЧТ	0	Зарезервировано
31	PAGE_1_REMAP	ЧТ/ЗП	0x2	Флаг использования первой резервной страницей

1.3.5.1.4.5.45 FMC_FLASH_BANK_1_RED_INFO (0x254)

Регистр FMC_FLASH_BANK_1_RED_INFO содержит таблицу замен повреждённых страниц для банка 1 контроллера FMC.

В таблице 1.89 представлено описание полей регистра FMC_FLASH_BANK_1_RED_INFO.

Таблица 1.89 – Описание полей регистра FMC_FLASH_BANK_1_RED_INFO

Поле	Название	Программный доступ	Состояние после сброса	Описание
9:0	PAGE_0_REMAP_ADDR	ЧТ/ЗП	0x1	Адрес повреждённой страницы, заменяемой нулевой резервной страницей
14:10	-	ЧТ	0	Зарезервировано
15	PAGE_0_REMAP	ЧТ/ЗП	0x2	Флаг использования нулевой резервной страницей
25:16	PAGE_1_REMAP_ADDR	ЧТ/ЗП	0x1	Адрес повреждённой страницы, заменяемой первой резервной страницей
30:26	-	ЧТ	0	Зарезервировано
31	PAGE_1_REMAP	ЧТ/ЗП	0x2	Флаг использования первой резервной страницей

1.3.5.1.4.5.46 FMC_OM_CMN_CFG_0 (0x400)

Регистр FMC_OM_CMN_CFG_0 содержит значение части OptMem вектора. При включенном OptMem состояние после сброса будет определяться содежимым флэш памяти, а приведенное ниже значение будет недоступно.

В таблице 1.90 представлено описание полей регистра FMC_OM_CMN_CFG_0.

Таблица 1.90 – Описание полей регистра FMC_OM_CMN_CFG_0

Поле	Название	Программный доступ	Состояние после сброса	Описание
0	TZEN	ЧТ	0	1 - старт микросхемы с поддержкой TrustZone; 0 - старт микросхемы без поддержки TrustZone;
1	BOOT0SRC	ЧТ	0	Выбор источника значения конфигурационного бита BOOT0: 0 - значение BOOT0 определяется значением, захватываемым в момент снятия активного уровня сигнала системного сброса с вывода согласно (см.п.1.2.2); 1 - значение BOOT0 определяется значением поля FMC_OM_CMN_CFG_0[BOOT0VAL].
2	BOOT0VAL	ЧТ	0	Значение конфигурационного бита BOOT0 при условии, что FMC_OM_CMN_CFG_0[BOOT0SRC] = 1
3	FMC_BANK_SWAP	ЧТ	0	1 - включение перестановки банков FMC (т.е. начальное значение поля FMC_CFG.BANK_SWAP); 0 - перестановка выключена
4	SRAM0_RST	ЧТ	0	1 - включение автоматической очистки SRAM0 при снятии сброса; 0 - выключено

Поле	Название	Программный доступ	Состояние после сброса	Описание
5	SRAM1_RST	ЧТ	0	1 - включение автоматической очистки SRAM1 при снятии сброса; 0 - выключено
6	SRAM2_RST	ЧТ	0	1 - включение автоматической очистки SRAM2 при снятии сброса; 0 - выключено
7	SRAM3_RST	ЧТ	0	1 - включение автоматической очистки SRAM3 при снятии сброса; 0 - выключено
8	BKPSRAM_RST	ЧТ	0	1 - включение автоматической очистки BKPSRAM при снятии сброса; 0 - выключено
11:9			0	Зарезервировано
12	WDT0_ON	ЧТ	0	1 - включение WDT0_ON при снятии сброса; 0 - WDT0_ON выключен
13	WDT1_ON	ЧТ	0	1 - включение WDT1_ON при снятии сброса; 0 - WDT1_ON выключен
15:14			0	Зарезервировано
16	ETH_EDCLDISABLE	ЧТ	0	0 - Ethernet EDCL по умолчанию включен; 1 - выключен
20:17	ETH_EDCLADDR	ЧТ	0	значение четырех младших бит EDCL MAC и EDCL IP
25:21	ETH_PHYRSTADDR	ЧТ	0	значение адреса внешнего Ethernet PHY, используемого в MDIO интерфейсе
31:26			0	Зарезервировано

1.3.5.1.4.5.47 FMC_OM_CMN_CFG_1 (0x404)

Регистр FMC_OM_CMN_CFG_1 содержит значение части OptMem вектора. При включенном OptMem состояние после сброса будет определяться содежимым флэш памяти, а приведенное ниже значение будет недоступно.

В таблице 1.91 представлено описание полей регистра FMC_OM_CMN_CFG_1.

Таблица 1.91 – Описание полей регистра FMC_OM_CMN_CFG_1

Поле	Название	Программный доступ	Состояние после сброса	Описание
0	BOR33EN	ЧТ	0	
3:1	BOR33LVL	ЧТ	0	
6:4	BOR33FS	ЧТ	0	
7			0	Зарезервировано
8	BOR33VBATEN	ЧТ	0	
11:9	BOR33VBATLVL	ЧТ	0	

Поле	Название	Программный доступ	Состояние после сброса	Описание
14:12	BOR33VBATFS	ЧТ	0	
15			0	Зарезервировано
23:16	MCU_PROT	ЧТ	0	Настройки защиты микроконтроллера
31:24			0	Зарезервировано

1.3.5.1.4.5.48 FMC_OM_CPU_NS_STADDR_0 (0x408)

Регистр FMC_OM_CPU_NS_STADDR_0 содержит значение части OptMem вектора. При включенном OptMem состояние после сброса будет определяться содежимым флэш памяти, а приведенное ниже значение будет недоступно.

В таблице 1.92 представлено описание полей регистра FMC_OM_CPU_NS_STADDR_0.

Таблица 1.92 – Описание полей регистра FMC_OM_CPU_NS_STADDR_0

Биты	Поле	Доступ	Значение после сброса	Описание
[6:0]	-	ЧТ	0x00	Зарезервировано
[31:7]	NSBOOTADDR0	ЧТ	0x00000000	Стартовый адрес 0 для незащищённого режима загрузки

1.3.5.1.4.5.49 FMC_OM_CPU_NS_STADDR_1 (0x40C)

Регистр FMC_OM_CPU_NS_STADDR_1 содержит значение части OptMem вектора. При включенном OptMem состояние после сброса будет определяться содежимым флэш памяти, а приведенное ниже значение будет недоступно.

В таблице 1.93 представлено описание полей регистра FMC_OM_CPU_NS_STADDR_1.

Таблица 1.93 – Описание полей регистра FMC_OM_CPU_NS_STADDR_1

Биты	Поле	Доступ	Значение после сброса	Описание
[6:0]	-	ЧТ	0x00	Зарезервировано
[31:7]	NSBOOTADDR1	ЧТ	0x00108000	Стартовый адрес 2 для незащищённого режима загрузки

1.3.5.1.4.5.50 FMC_OM_CPU_SEC_STADDR0 (0x410)

Регистр FMC_OM_CPU_SEC_STADDR0 содержит значение части OptMem вектора. При включенном OptMem состояние после сброса будет определяться содежимым флэш памяти, а приведенное ниже значение будет недоступно.

В таблице 1.94 представлено описание полей регистра FMC_OM_CPU_SEC_STADDR0.

Таблица 1.94 – Описание полей регистра FMC_OM_CPU_SEC_STADDR0

Биты	Поле	Доступ	Значение после сброса	Описание
[0]	BOOTLOCK	ЧТ	0x0	Включение режима единой точки старта (см.раздел 1.3.17) для защищённого режима загрузки: 0 - режим единой точки старта отключён; 1 - режим единой точки старта включён;
[6:1]	-	ЧТ	0x00	Зарезервировано
[31:7]	SECBOOTADDR0	ЧТ	0x10000000	Стартовый адрес 0 для защищённого режима загрузки

1.3.5.1.4.5.51 FMC_OM_CPU_SEC_STADDR1 (0x414)

Регистр FMC_OM_CPU_SEC_STADDR1 содержит значение части OptMem вектора. При включенном OptMem состояние после сброса будет определяться содежимым флэш памяти, а приведенное ниже значение будет недоступно.

В таблице 1.95 представлено описание полей регистра FMC_OM_CPU_SEC_STADDR1.

Таблица 1.95 – Описание полей регистра FMC_OM_CPU_SEC_STADDR1

Биты	Поле	Доступ	Значение после сброса	Описание
[6:0]	-	ЧТ	0x00	Зарезервировано
[31:7]	SECBOOTADDR1	ЧТ	0x10108000	Стартовый адрес 1 для защищённого режима загрузки

1.3.5.1.4.5.52 FMC_OM_PROTECT_CFG_CMN (0x418)

Регистр FMC_OM_PROTECT_CFG_CMN содержит значение части OptMem вектора. При включенном OptMem состояние после сброса будет определяться содежимым флэш памяти, а приведенное ниже значение будет недоступно.

Значение этого регистра будет автоматически установлено в регистр FMC_PROTECT_CFG_CMN при снятии сброса микросхемы.

В таблице 1.96 представлено описание полей регистра FMC_OM_PROTECT_CFG_CMN.

Таблица 1.96 – Описание полей регистра FMC_OM_PROTECT_CFG_CMN

Поле	Название	Программный доступ	Состояние после сброса	Описание
0	WRP_0	ЧТ	0	Защита банка 0 от записей и стираний.
1	WRP_1	ЧТ	0	Защита банка 1 от записей и стираний.

Поле	Название	Программный доступ	Состояние после сброса	Описание
3:2			0	Зарезервировано
4	WRP_IP0	ЧТ	0	Защита от записи и стираний информационной страницы 0 банка 0
5	WRP_IP1	ЧТ	0	Защита от записи и стираний информационной страницы 1 банка 0
6	WRP_IP2	ЧТ	0	Защита от записи и стираний информационной страницы 2 банка 0
7	WRP_IP3	ЧТ	0	Защита от записи и стираний информационной страницы 3 банка 0
8	WRP_IP4	ЧТ	0	Защита от записи и стираний информационной страницы 0 банка 1
9	WRP_IP5	ЧТ	0	Защита от записи и стираний информационной страницы 1 банка 1
10	WRP_IP6	ЧТ	0	Защита от записи и стираний информационной страницы 2 банка 1
11	WRP_IP7	ЧТ	0	Защита от записи и стираний информационной страницы 3 банка 1
30:12			0	Зарезервировано
31	PROT_BLK_RST	ЧТ	0	Запрет программного изменения настроек защиты от записи и чтения

1.3.5.1.4.5.53 FMC_OM_SECURITY_CFG_CMN (0x41C)

Регистр FMC_OM_SECURITY_CFG_CMN содержит значение части OptMem вектора. При включенном OptMem состояние после сброса будет определяться содежимым флэш памяти, а приведенное ниже значение будет недоступно.

Значение этого регистра будет автоматически установлено в регистр FMC_SECURITY_CFG_CMN при снятии сброса микросхемы.

В таблице 1.97 представлено описание полей регистра FMC_OM_SECURITY_CFG_CMN.

Таблица 1.97 – Описание полей регистра FMC_OM_SECURITY_CFG_CMN

Биты	Поле	Доступ	Значение после сброса	Описание
0	SEC_IP0	ЧТ	0	Секьюрность информационной страницы 0
1	SEC_IP1	ЧТ	0	Секьюрность информационной страницы 1
2	SEC_IP2	ЧТ	0	Секьюрность информационной страницы 2
3	SEC_IP3	ЧТ	0	Секьюрность информационной страницы 3
4	SEC_IP4	ЧТ	0	Секьюрность информационной страницы 4

Биты	Поле	До-ступ	Значение после сброса	Описание
5	SEC_IP5	ЧТ	0	Секьюрность информационной страницы 5
6	SEC_IP6	ЧТ	0	Секьюрность информационной страницы 6
7	SEC_IP7	ЧТ	0	Секьюрность информационной страницы 7
31:8			0	Зарезервировано

1.3.5.1.4.5.54 FMC_OM_WRP_CFG_0 (0x420)

Регистр FMC_OM_WRP_CFG_0 содержит значение части OptMem вектора. При включенном OptMem состояние после сброса будет определяться содежимым флэш памяти, а приведенное ниже значение будет недоступно.

Значение этого регистра будет автоматически установлено в регистр FMC_WRP_CFG_0 при снятии сброса микросхемы.

В таблице 1.98 представлено описание полей регистра FMC_OM_WRP_CFG_0.

Таблица 1.98 – Описание полей регистра FMC_OM_WRP_CFG_0

Биты	Поле	До-ступ	Значение после сброса	Описание
31:0		ЧТ	0	

1.3.5.1.4.5.55 FMC_OM_WRP_CFG_1 (0x424)

Регистр FMC_OM_WRP_CFG_1 содержит значение части OptMem вектора. При включенном OptMem состояние после сброса будет определяться содежимым флэш памяти, а приведенное ниже значение будет недоступно.

Значение этого регистра будет автоматически установлено в регистр FMC_WRP_CFG_1 при снятии сброса микросхемы.

В таблице 1.99 представлено описание полей регистра FMC_OM_WRP_CFG_1.

Таблица 1.99 – Описание полей регистра FMC_OM_WRP_CFG_1

Биты	Поле	До-ступ	Значение после сброса	Описание
31:0		ЧТ	0	

1.3.5.1.4.5.56 FMC_OM_WRP_CFG_2 (0x428)

Регистр FMC_OM_WRP_CFG_2 содержит значение части OptMem вектора. При включенном OptMem состояние после сброса будет определяться содежимым флэш памяти, а приведенное ниже значение будет недоступно.

Значение этого регистра будет автоматически установлено в регистр FMC_WRP_CFG_2 при снятии сброса микросхемы.

В таблице 1.100 представлено описание полей регистра FMC_OM_WRP_CFG_2.

Таблица 1.100 – Описание полей регистра FMC_OM_WRP_CFG_2

Биты	Поле	До-ступ	Значение после сброса	Описание
31:0		ЧТ	0	

1.3.5.1.4.5.57 FMC_OM_WRP_CFG_3 (0x42C)

Регистр FMC_OM_WRP_CFG_3 содержит значение части OptMem вектора. При включенном OptMem состояние после сброса будет определяться содежимым флэш памяти, а приведенное ниже значение будет недоступно.

Значение этого регистра будет автоматически установлено в регистр FMC_WRP_CFG_3 при снятии сброса микросхемы.

В таблице 1.101 представлено описание полей регистра FMC_OM_WRP_CFG_3.

Таблица 1.101 – Описание полей регистра FMC_OM_WRP_CFG_3

Биты	Поле	До-ступ	Значение после сброса	Описание
31:0		ЧТ	0	

1.3.5.1.4.5.58 FMC_OM_SECURITY_CFG_0 (0x430)

Регистр FMC_OM_SECURITY_CFG_0 содержит значение части OptMem вектора. При включенном OptMem состояние после сброса будет определяться содежимым флэш памяти, а приведенное ниже значение будет недоступно.

Значение этого регистра будет автоматически установлено в регистр FMC_SECURITY_CFG_0 при снятии сброса микросхемы.

В таблице 1.102 представлено описание полей регистра FMC_OM_SECURITY_CFG_0.

Таблица 1.102 – Описание полей регистра FMC_OM_SECURITY_CFG_0

Биты	Поле	До-ступ	Значение после сброса	Описание
31:0		ЧТ	0	

1.3.5.1.4.5.59 FMC_OM_SECURITY_CFG_1 (0x434)

Регистр FMC_OM_SECURITY_CFG_1 содержит значение части OptMem вектора. При включенном OptMem состояние после сброса будет определяться содежимым флэш памяти, а приведенное ниже значение будет недоступно.

Значение этого регистра будет автоматически установлено в регистр FMC_SECURITY_CFG_1 при снятии сброса микросхемы.

В таблице 1.103 представлено описание полей регистра FMC_OM_SECURITY_CFG_1.

Таблица 1.103 – Описание полей регистра FMC_OM_SECURITY_CFG_1

Биты	Поле	До-ступ	Значение после сброса	Описание
31:0		ЧТ	0	

1.3.5.1.4.5.60 FMC_OM_SECURITY_CFG_2 (0x438)

Регистр FMC_OM_SECURITY_CFG_2 содержит значение части OptMem вектора. При включенном OptMem состояние после сброса будет определяться содежимым флэш памяти, а приведенное ниже значение будет недоступно.

Значение этого регистра будет автоматически установлено в регистр FMC_SECURITY_CFG_2 при снятии сброса микросхемы.

В таблице 1.104 представлено описание полей регистра FMC_OM_SECURITY_CFG_2.

Таблица 1.104 – Описание полей регистра FMC_OM_SECURITY_CFG_2

Биты	Поле	До-ступ	Значение после сброса	Описание
31:0	DATA	ЧТ	0	

1.3.5.1.4.5.61 FMC_OM_SECURITY_CFG_3 (0x43C)

Регистр FMC_OM_SECURITY_CFG_3 содержит значение части OptMem вектора. При включенном OptMem состояние после сброса будет определяться содежимым флэш памяти, а приведенное ниже значение будет недоступно.

Значение этого регистра будет автоматически установлено в регистр FMC_SECURITY_CFG_3 при снятии сброса микросхемы.

В таблице 1.105 представлено описание полей регистра FMC_OM_SECURITY_CFG_3.

Таблица 1.105 – Описание полей регистра FMC_OM_SECURITY_CFG_3

Биты	Поле	Доступ	Значение после сброса	Описание
31:0		ЧТ	0	

1.3.5.1.4.5.62 FMC_PWR_KEY (0x000)

Запись в регистр FMC_PWR_KEY значения 0x7375646F разрешает запись во все остальные регистры FMC_PWR контроллера, запись любого другого значения - запрещает. Если запись в регистры FMC_PWR контроллера разрешена, то из регистра FMC_PWR_KEY читается 0x1.

В таблицах 1.106 и 1.107 представлено описание полей регистра FMC_PWR_KEY при операциях записи и чтения, соответственно.

Таблица 1.106 – Описание полей регистра FMC_PWR_KEY при записи

Биты	Поле	Доступ	Значение после сброса	Описание
31:0	KEY	ЗП	0x0	Ключ для записи в регистры FMC_PWR контроллера

Таблица 1.107 – Описание полей регистра FMC_PWR_KEY при чтении

Поле	Название	Программный доступ	Состояние после сброса	Описание
0	WR_PERMISSION	ЧТ	0x0	Флаг разрашения записи
31:1	-	ЧТ	0x0	Зарезервировано

1.3.5.1.4.5.63 FMC_PWR_SET_STATE (0x004)

Регистр настройки состояния банков eFlash памяти контроллера FMC.

В таблице 1.108 представлено описание полей регистра FMC_PWR_SET_STATE.

Таблица 1.108 – Описание полей регистра FMC_PWR_SET_STATE

Поле	Название	Программный доступ	Состояние после сброса	Описание
1:0	BANK_0_SET_POWER_STATE	ЧТ/ЗП	0	Установка желаемого состояния банка eFlash памяти 0: 0 - GOTO_POWER_DOWN; 1 - GOTO_SLEEP; 2 - GOTO_ENABLE; 3 - резерв (аналогично ENABLE).

Поле	Название	Программный доступ	Состояние после сброса	Описание
3:2	-	ЧТ	0	Зарезервировано
5:4	BANK_1_SET_POWER_STATE	ЧТ/ЗП	0	Установка желаемого состояния банка eFlash памяти 1 (допустимые значения аналогичноы BANK_0_SET_POWER_STATE)
7:6	-	ЧТ	0	Зарезервировано
9:8	LVE_MODE	ЧТ/ЗП	2	Способ установки режима пониженного напряжения (управление входом LVE макроса eFlash памяти): 0 - Hard Ware - значение расчитывается на основе настройки внутреннего регулятора напряжения; 1 - Soft Ware - значение устанавливается программно (настройка внутреннего регулятора напряжения игнорируется); 2 - Mixed - значение расчитывается на основе настройки внутреннего регулятора напряжения, но может быть изменено программно; 3 - резерв - работает как Mixed.
11:10	-	ЧТ	0	Зарезервировано
12	SET_LVE	ЧТ/ЗП	0	Программная установка режиме пониженного напряжения
15:13	-	ЧТ	0	Зарезервировано
16	PWR_MODE	ЧТ/ЗП	0	Способ установки сигналов управления питанием макроса eFlash памяти (PWRRDY_IO и PDM_IO): 0 - Hard Ware - используются значения, в соответствии с машиной состояний (см. рисунок 1.18); 1 - Soft Ware - программная установка;
19:17	-	ЧТ	0	Зарезервировано
20	BANK_0_SET_PDM	ЧТ/ЗП	0x3	Программная установка сигнала стабильности напряжения ядра для банка 0 eFlash памяти
21	BANK_1_SET_PDM	ЧТ/ЗП	0x3	Программная установка сигнала стабильности напряжения ядра для банка 1 eFlash памяти
23:22	-	ЧТ	0	Зарезервировано
24	SET_PWRRDY	ЧТ/ЗП	0x1	Программная установка сигнала стабильности входного напряжения
31:25	-	ЧТ	0	Зарезервировано

1.3.5.1.4.5.64 FMC_PWR_CUR_STATE (0x008)

Регистр текущего состояния банков eFlash памяти контроллера FMC.

В таблице 1.109 представлено описание полей регистра FMC_PWR_CUR_STATE.

Таблица 1.109 – Описание полей регистра FMC_PWR_CUR_STATE

Поле	Название	Программный доступ	Состояние после сброса	Описание
2:0	BANK_0_CUR_POWER_STATE	ЧТ	0	Акутальное состояние банка 0 eFlash памяти: 0 - POWER_DOWN - банк 0 выключен; 1 - SWITCH_ON - банк 0 включается; 2 - SLEEP - банк 0 в режиме сна; 3 - WAKE_UP - банк 0 выходит из режима сна; 4 - BOR - пропало питание контроллера FMC; 5 - RESETTING - выполняется аппартный сброс контроллера FMC; 6 - STANDBY - банк 0 включен и находится в режиме ожидания (выполняются только операции чтения); 7 - ACTIVE - банк 0 включен и находится в активном режиме (выполняется операция записи, стирания страницы или стирания всего банка).
3	-	ЧТ	0	Зарезервировано
6:4	BANK_1_CUR_POWER_STATE	ЧТ	0	Акутальное состояние 1 eFlash памяти (значения аналогичноы BANK_0_CUR_POWER_STATE)
7	-	ЧТ	0	Зарезервировано
8	FMC_LVE_STATE	ЧТ	0	Состояние входа LVE макроса eFlash памяти
11:9	-	ЧТ	0	Зарезервировано
12	SYSTEM_LVE_STATE	ЧТ	0	Состояние напряжения ядра (настройка внутреннего регулятора напряжения): 0 - 1,1 В; 1 - 0,9 В.
31:13	-	ЧТ	0	Зарезервировано

1.3.5.1.4.5.65 FMC_PWR_ERR_STATE (0x00C)

Регистр состояния контроллера FMC в момент когда произошёл аппартный сброс или пропадание напряжения.

В таблице 1.110 представлено описание полей регистра FMC_PWR_ERR_STATE.

Таблица 1.110 – Описание полей регистра FMC_PWR_ERR_STATE

Поле	Название	Программный доступ	Состояние после сброса	Описание
15:0	ADDESS	ЧТ	0	Адрес операции
17:16	MEMORY_ARRAY	ЧТ	0	Область банка памяти: 0 - Main; 1 - Information; 2,3 - резерв.

Поле	Название	Программный доступ	Состояние после сброса	Описание
18	-	ЧТ	0	Зарезервировано
19	BANK_NUBER	ЧТ	0	Номер банка памяти
22:20	OP_CODE	ЧТ	0	Код операции: 0 - нет операции; 1 - PROG - операция записи; 2 - PAGE_ERASE - операция стирания страницы памяти; 3 - MASS_ERASE - операция стирания всей области банка памяти; 4-7 - резерв.
23	-	ЧТ	0	Зарезервировано
24	FMC_RESETTING	ЧТ	0	Флаг, указывающий, что произошёл аппаратный сброс контроллера FMC
25	BOR	ЧТ	0	Флагг, указывающий, что произошло пропадание напряжения контроллера FMC
31:26	-	ЧТ	0	Зарезервировано

1.3.5.1.4.5.66 FMC_PWR_SETTINGS (0x010)

Регистр состояния контроллера FMC в момент когда произошёл аппартный сброс или пропадание напряжения.

В таблице 1.111 представлено описание полей регистра FMC_PWR_SETTINGS.

Таблица 1.111 – Описание полей регистра

Поле	Название	Программный доступ	Состояние после сброса	Описание
7:0	Tus	ЧТ/ЗП	0x10	Длительность одной микросекунды в тактах (астота работы в МГц)
15:8	Tpud	ЧТ/ЗП	0x0A	Задержка включения контроллера FMC в тактах (можно имитировать Tpdh)
16	AUTO_RECALL	ЧТ/ЗП	0x1	Автоматически выполнять операцию RECALL после подачи питания на макросы eFlash памяти
19:17	-	ЧТ	0	Зарезервировано
20	REPAIR_EN	ЧТ/ЗП	0x1	Автоматически выполнять чтение таблицы замен страниц после включения контроллера FMC
23:21	-	ЧТ	0	Зарезервировано
24	OPTMEM_EN	ЧТ/ЗП	0x1	Автоматически выполнять чтение Option Memory после включения контроллера FMC
27:25	-	ЧТ	0	Зарезервировано

Поле	Название	Программный доступ	Состояние после сброса	Описание
28	OPTMEM_EN_SW_CTRL	ЧТ	0	Переключение между программным и аппаратным управлением чтением Option Memory: 0 - используется BOOT_OPT_EN (поле OPTMEM_EN игнорируется); 1 - используется поле OPTMEM_EN (состояние BOOT_OPT_EN игнорируется).
31:29	-	ЧТ	0	Зарезервировано

1.3.5.1.4.5.67 FMC_PWR_TIMINGS (0x014)

Регистр настройки временных параметров FMC_PWR контроллера.

В таблице 1.112 представлено описание полей регистра FMC_PWR_TIMINGS.

Таблица 1.112 – Описание полей регистра FMC_PWR_TIMINGS

Поле	Название	Программный доступ	Состояние после сброса	Описание
7:0	Tvls	ЧТ/ЗП	0x0A	Задержка изменения сигнала LVE, отвечающего за включение режима работы на пониженном напряжении ядра
15:8	Twk	ЧТ/ЗП	0x05	Задержка включения банка eFlash памяти при выходе из режима POWER DOWN или SLEEP,эта же задержка применяется после операции записи или стирания страницы (Tnvh)
23:16	Tnvh1	ЧТ/ЗП	0x64	Задержка при прерывании операции стирания банка eFlash памяти
31:24	Trcv	ЧТ/ЗП	0x0A	Задержка после выполнения любой длительной операции

1.3.5.1.5 Указания к программированию

Все описанные операции, а также чтение, следует делать во включенном состоянии контроллера FMC и после завершения чтения OptMem.

1.3.5.1.5.1 Операция записи в eFlash-память

Рекомендуемая последовательность действий при выполнении операции записи одного 128-битного слова по выровненному по границе 16 байт адресу:

1) сбросить статус FMC_IRQ_STATUS[WRITE_CMPL] записью 1 в FMC_IRQ_CLR[WRITE_CMPL] [1] [2];

2) заполнить буфер - записать четыре 32-битных слова по последовательным адресам [3], и адресом первой транзакции выровненной по границе 16 байт [4];

3) дождаться установки статуса FMC_IRQ_STATUS[WRITE_CMPL] (вызова прерывания fmc_irq, если оно разрешено) [5];

4) запись завершена.

Рекомендуемая последовательность действий при выполнении операции записи одного блока данных размером меньше 128 бит по выровненному по размеру адресу:

1) сбросить статус FMC_IRQ_STATUS[WRITE_CMPL] записью 1 в FMC_IRQ_CLR[WRITE_CMPL] [1] [2];

2) записать данное по требуемому адресу;

3) вызвать принудительную запись 1.3.5.1.5.1.1;

4) дождаться установки статуса FMC_IRQ_STATUS[WRITE_CMPL] (вызова прерывания, если оно разрешено) [5];

5) запись завершена.

Примечание 1: при отслеживании завершения операции записи с помощью прерывания, проверить, что оно разрешено (поле FMC_IRQ_MASK[WRITE_CMPL] установлено в 1).

Примечание 2: при отладке ПО первым пунктом операции записи рекомендуется проверять, что буфер записи пуст (FMC_DBG_STATUS[WBUF_EMPTY] == 1). Если это не так, скорее всего в данный момент есть другая незавершенная операция записи. Пересечение записей не ведёт к потере данных, но увеличит время исполнения и потребление. Также будет нарушение порядка прерываний.

Примечание 3: контроллер FMC расчитан на то, что транзакций может быть не 4, данные могут быть любой ширины и адреса могут идти не по порядку - главное, чтобы они соответствовали адресу записываемого 128-битного слова и заполнили буфер.

Примечание 4: после заполнения буфера запись в eFlash-память начнётся автоматически - буфер опустошится (FMC_DBG_STATUS[WBUF_EMPTY] == 1) и банк станет занят (FMC_DBG_STATUS[BANK0_BUSY] == 1 или FMC_DBG_STATUS[BANK1_BUSY] == 1).

Примечание 5: при отладке ПО после окончания операции записи также рекомендуется проверять, что буфер записи пуст (FMC_DBG_STATUS[WBUF_EMPTY] == 1). Если это не так, скорее всего на текущую операцию наложилась другая запись.

1.3.5.1.5.1.1 Последовательность принудительной записи

1) установить бит FMC_CONTROL[FWRITE] (с соответствующим ключом);

2) дождаться сброса бита FMC_CONTROL[FWRITE].

В рабочем состоянии бит FMC_CONTROL[FWRITE] должен сброситься за 1-2 чтения регистра.

1.3.5.1.5.2 Операция блочной записи в eFlash-память

Рекомендуемая последовательность действий при выполнении операции записи одного блока по выровненному по границе 16 байт адресу:

1) перейти в режим блочной записи, установив бит FMC_CFG[WRITE_MODE] равным 1;

2) убедиться, что нет незавершенной операции записи;

3) сбросить статус FMC_IRQ_STATUS[WRITE_CMPL] записью 1 в FMC_IRQ_CLR[WRITE_CMPL] [1];

4) заполнить блок - записать несколько 128-битных слова:

– каждое 128-битное слово - это четыре 32-битных слова по последовательным адресам [2], и адресом первой транзакции выровненной по границе 16 байт;

– адреса 128-битных слов - не обязательно последовательные, но их старшие части (биты [31:9]) должны быть одинаковые;

5) если количество 128-битных слов больше 8, программно активировать принудительную запись, установив в 1 поле FMC_CONTROL[BLOCK_FWRITE] (с соответствующим ключом);

6) дождаться установки статуса FMC_IRQ_STATUS[WRITE_CMPL] (вызова прерывания fmc_irq, если оно разрешено) [5];

7) запись завершена;

8) выйти из режима блочной записи, установив бит FMC_CFG[WRITE_MODE] равным 0.

Примечание 1: при отслеживании завершения операции записи с помощью прерывания, проверить, что оно разрешено (поле FMC_IRQ_MASK[WRITE_CMPL] установлено в 1).

Примечание 2: контроллер FMC расчитан на то, что транзакций может быть не 4, данные могут быть любой ширины и адреса могут идти не по порядку - главное, чтобы они соответствовали адресу записываемого 128-битного слова и заполнили буфер.

1.3.5.1.5.3 Операции стирания флэш памяти

Рекомендуемая последовательность действий при выполнении операции постраничного стирания:

1) убедиться, что не исполняется никакая операция стирания (FMC_CONTROL[ERASE] == 0 && FMC_CONTROL[MERASE0] == 0 && FMC_CONTROL[MERASE1] == 0) [1] [2];

2) сбросить статус FMC_IRQ_STATUS[ERASE_CMPL] записью 1 в FMC_IRQ_CLR[ERASE_CMPL];

3) записать адреса начальной и конечной страниц в регистры FMC_ERASE_ADR_L и FMC_ERASE_ADR_H [2];

4) установить флаг запуска операции - поле FMC_CONTROL[ERASE] (с соответствующим ключом);

5) дождаться установки статуса FMC_IRQ_STATUS[ERASE_CMPL] (вызова прерывания, если оно разрешено в FMC_IRQ_MASK[ERASE_CMPL]);

6) стирание указанного диапазона страниц завершено.

Рекомендуемая последовательность действий при выполнении операции общего стирания основной области банка памяти:

1) убедиться, что не исполняется никакая операция стирания (FMC_CONTROL[ERASE] == 0 && FMC_CONTROL[MERASE0] == 0 && FMC_CONTROL[MERASE1] == 0) [1];

2) сбросить статус FMC_IRQ_STATUS[M_ERASE_CMPL] записью 1 в FMC_IRQ_CLR[M_ERASE_CMPL];

3) установить флаг запуска операции - записать 1 в FMC_CONTROL[MERASE0] или FMC_CONTROL[MERASE1] (с соответствующим ключом);

4) дождаться установки статуса FMC_IRQ_STATUS[M_ERASE_CMPL] (вызова прерывания, если оно разрешено);

5) стирание банка завершено.

Примечание 1: при отслеживании завершения операции с помощью прерывания, проверить, что оно разрешено (поле FMC_IRQ_MASK[ERASE_CMPL] установлено в 1).

Примечание 2: если предыдущее постраничное стирание не завершено, настраивать или начинать новое запрещено. Смена FMC_ERASE_ADR_L, FMC_ERASE_ADR_H или установка FMC_CONTROL[ERASE] при установленном FMC_CONTROL[ERASE] приведут к неопределенному поведению.

Примечание 3: поля регистров FMC_ERASE_ADR_L и FMC_ERASE_ADR_H имеет такое смещение, что для основной флэш памяти исходный байтовый адрес на системной шине не надо программно модифицировать. Можно записать любой адрес внутри страницы - его младшая не значащая часть будет отброшена.

1.3.5.1.5.4 Перестановка банков

Рекомендуемая последовательность действий при выполнении перестановки банков:

1) обеспечить отсутствие транзакций к контроллеру FMC;

2) убедиться, что буфер записи пуст (FMC_DBG_STATUS[WBUF_EMPTY] == 1) [1];

3) убедиться, что оба банка бездействуют (FMC_DBG_STATUS[BANK0_BUSY] == 0 и FMC_DBG_STATUS[BANK1_BUSY] == 0);

4) очистить буфер чтения - записать 1 в FMC_CONTROL[CLR_RBUF] (с соответствующим ключом);

5) переставить банки, инвертировав значение поля FMC_CFG[BANK_SWAP].

Примечание 1: если нет - сбросить FMC_IRQ_STATUS[WRITE_CMPL], осуществить принудительную запись и дождаться установки FMC_IRQ_STATUS[WRITE_CMPL];

1.3.5.2 Внутреннее ОЗУ

Внутреннее ОЗУ разделено на подсистему основной статической памяти (SRAM) суммарным объемом 160КБайт (128КБайт для хранения данных и 32КБайта для хранения кодов коррекции ошибок) и статическую память батарейного домена питания (BKPSRAM) объемом 2КБайта (используется и для данных, и для кодов коррекции ошибок).

1.3.5.2.1 Сборка подсистемы основной статической памяти (SRAM)

Сборка подсистемы основной статической памяти состоит из 4 банков памяти и управляющего контроллера. Каждый банк памяти имеет свой AHB интерфейс и ему соответствует окно адресов в карте памяти.

Каждый банк памяти поддерживает защиту данных кодами коррекции ошибок ECC. Код коррекции ошибок ECC позволяет исправить одиночную ошибку на 32 битное слово данных и детектировать двойную ошибку. В случае возникновения одиночной или двойной ошибки выставляется маскируемое прерывание.

Внешняя настройка тактировани каждого из банков памяти, задается битами SRAMx_EN (SRAM0_EN, SRAM1_EN, SRAM2_EN, SRAM3_EN) в регистре RCU_PCER1 блока RCU (см. 1.3.7.7.17).

Программный доступ при снятом тактировании будет возвращать ошибку системной шины и вызывать исключение процессорного ядра.

1.3.5.2.1.1 Структурная схема подсистемы памяти SRAM

Структурная схема подсистемы памяти SRAM приведена на рисунке 1.19.

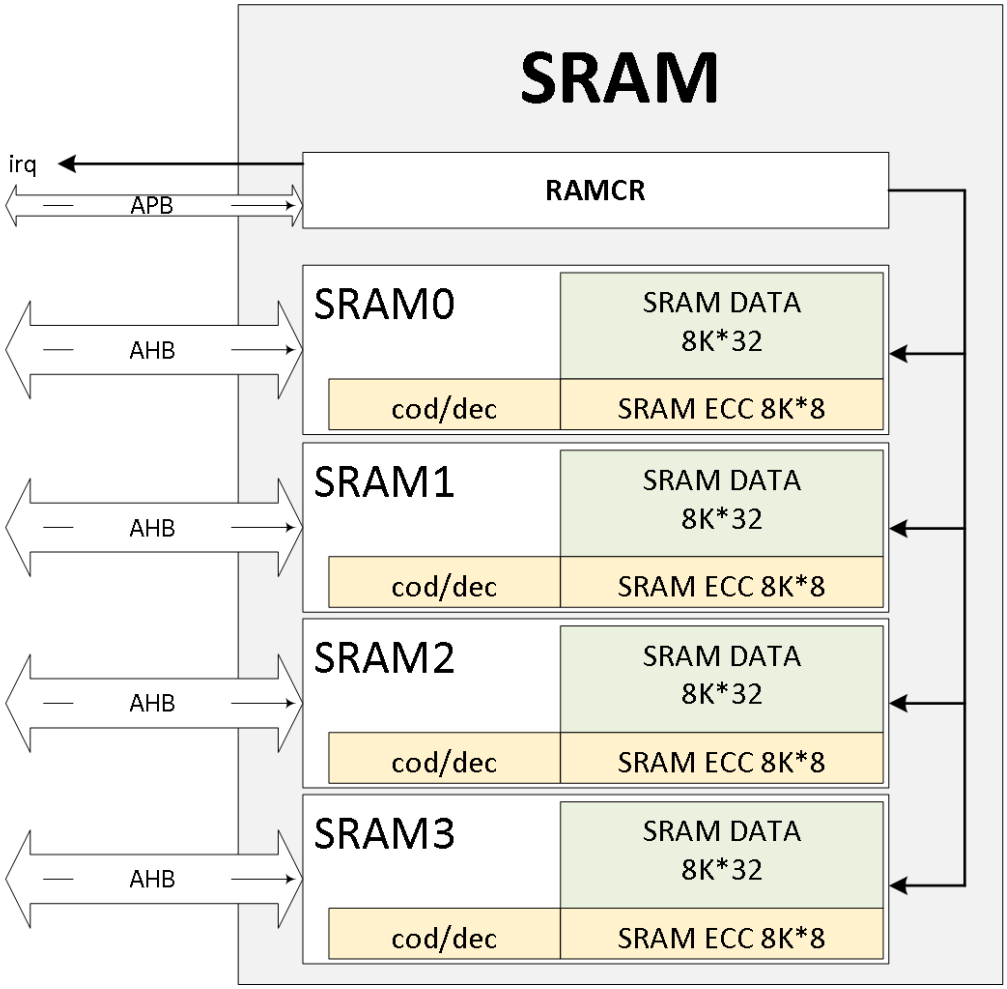


Рисунок 1.19 – Структурная схема подсистемы памяти SRAM

Описание блоков структурной схемы подсистемы памяти SRAM приведено в таблице 1.113.

Таблица 1.113 – Описание блоков структурной схемы подсистемы памяти SRAM

Блок	Описание
SRAM	Сборка подблоков, верхний уровень подсистемы памяти SRAM
RAMCR	Подблок для программирования настроек блоков SRAM0(1,2,3): включения/отключения режимов пониженного энергопотребления блоков памяти, настрек ECC, обработки прерываний в случае возникновения ошибок выявленных ECC
SRAM0(1,2,3)	Подблоки содержащие сборку баков памяти и кодеров/декодеров реализующих коррекцию ошибок ECC
cod/dec	Подблок содержащий кодер/декодер для коррекции ошибок ECC
SRAM DATA	Память для хранения данных
SRAM ECC	Память для хранения кодов коррекции ошибок ECC

1.3.5.2.1.2 Режимы пониженного энергопотребления (LS, DS, SD)

Память имеет 3 режима пониженного энергопотребления:

– LS - Light Sleep - "легки сон" - режим характеризуется быстрым временем выхода на рабочий режим;

– DS - Deep Sleep - "глубокий сон" - временем выхода на рабочий режим дольше чем у LS;

– SD - Shut Down - память выключена (записанные значения не сохраняются).

Если память данных находится в режимах пониженного энергопотребления (LS, DS, SD), а также если включено ECC и память для ECC находится в режимах пониженного энергопотребления (LS, DS, SD), то запись в память игнорируется, чтение возвращает значение "32'h000_e7707". Если в блоке SYSCR в регистре CFG_PSLVERR[cfg_pslverr] == 1, то в случае обращений к памяти при LS, DS, SD будет формироваться ошибка на шине AHB (HRESP = 1).

1.3.5.2.1.3 Стирание памяти

При возникновении сигнала sys_erase_i от блока TAMP происходит стирание всех памятей (и памятей данных и памятей хранящих коды коррекции ошибок). При этом при sys_erase_i == 1 запись в память игнорируется, чтение возвращает значение "32'h000_e7707".

Если в блоке SYSCR в регистре CFG_PSLVERR[cfg_pslverr] == 1, то в случае обращений к памяти при sys_erase_i == 1 будет формироваться ошибка на шине AHB (HRESP = 1).

Если в этот момент какие-то памяти находятся в режимах пониженного энергопотребления (LS, DS, SD), то память переводится в рабочий режим и после этого память стирается (в память данных записываются значение 32'h0000_0000, в память ECC соответствующий значению 32'h0000_0000 код коррекции).

Программное стирание производится через записи единиц в нужные биты регистра AHB5_SRAM_PROGR_ERASE. Программное стирание не распространяется на памяти в режимах пониженного энергопотребления (LS, DS, SD).

1.3.5.2.1.4 Перенастройка памяти при изменении напряжения питания

В зависимости от напряжения питания микросхемы должно быть выставлено необходимое значение:

– или в регистрах AHB5_SRAM_CFG_SRAM* в полях cfg_mar (см. 1.3.5.2.1.5.2.6) и в регистре AHB5_SRAM_SETTINGS в полях sram*_cfg_mar_syscr установлен 0;

– или в регистре AHB5_SRAM_SETTINGS в полях sram*_cfg_mar_syscr установлена 1 и установлено необходимое значение в блоке SYSCR в регитсре CFG_SRAM.

1.3.5.2.1.5 Описание регистров подсистемы памяти SRAM

1.3.5.2.1.5.1 Карта регистров подсистемы памяти SRAM

Задание конфигурационных параметров и отображение текущего состояния подсистемы памяти SRAM ведется через регистры. Доступ к регистрам осуществляется посредством системного интерфейса управления (интерфейс APB 4/5 slave). Каждое обращение к регистрам имеет выравнивание до 32-х разрядов. Карта регистров приведена в таблице 1.114.

При обращении к регистрам СФ-блока по зарезервированным адресам никаких ошибок не возникает. При попытке записи в регистры, предназначенные только для чтения, операция записи завершается успешно, однако, состояние регистров не изменяется.

Таблица 1.114 – Карта регистров подсистемы памяти SRAM

Адрес	Название	Доступ	RST	Описание
0x000	AHB5_SRAM_ID	RO	32'h4d41_5253	Идентификатор устройства
0x004	AHB5_SRAM_VER	RO	32'h0041_0120	Номер версии устройства и конфигурация
0x008-0x00C	-	-	32'he7707_ad7	Резерв
0x010	AHB5_SRAM_IRQ_STATUS[1)]	RO	32'h0	Регистр статуса прерывания
0x014	AHB5_SRAM_IRQ_MASK[1)]	RW	32'h0	Регистр маски прерывания
0x018	AHB5_SRAM_IRQ_CLEAR[1)]	WO(W1C)	32'h0	Регистр сброса статуса прерывания
0x01C	AHB5_SRAM_IRQ_TEST[1)]	WO	32'h0	Регистр установки статуса прерывания
0x020	AHB5_SRAM_SETTINGS[1)]	RW	32'h0	Регистр настройки блока
0x024-0x03C	-	-	32'he7707_ad7	Резерв
0x040	AHB5_SRAM_SEC_ADDR_0	RO	32'h0	Регистр адреса при обращении, по которому (по шине AHB0) зафиксирована одиночная ошибка ECC Регистр обновляется только когда соответствующий бит статуса прерывания равен "0" (AHB5_SRAM_IRQ_STATUS[0]==0)
0x044	AHB5_SRAM_SEC_ADDR_1	RO	32'h0	Регистр адреса при обращении, по которому (по шине AHB1) зафиксирована одиночная ошибка ECC Регистр обновляется только когда соответствующий бит статуса прерывания равен "0" (AHB5_SRAM_IRQ_STATUS[4]==0)
0x048	AHB5_SRAM_SEC_ADDR_2	RO	32'h0	Регистр адреса при обращении, по которому (по шине AHB2) зафиксирована одиночная ошибка ECC Регистр обновляется только когда соответствующий бит статуса прерывания равен "0" (AHB5_SRAM_IRQ_STATUS[8]==0)

Адрес	Название	Доступ	RST	Описание
0x04C	AHB5_SRAM_SEC_ADDR_3	RO	32’h0	Регистр адреса при обращении, по которому (по шине AHB3) зафиксирована одиночная ошибка ECC Регистр обновляется только когда соответствующий бит статуса прерывания равен "0" (AHB5_SRAM_IRQ_STATUS[12]==0)
0x050-0x05C	-	-	32’h0	Резерв
0x060	AHB5_SRAM_DED_ADDR_0	RO	32’h0	Регистр адреса при обращении, по которому (по шине AHB0) зафиксирована двойная ошибка ECC Регистр обновляется только когда соответствующий бит статуса прерывания равен "0" (AHB5_SRAM_IRQ_STATUS[1]==0)
0x064	AHB5_SRAM_DED_ADDR_1	RO	32’h0	Регистр адреса при обращении, по которому (по шине AHB1) зафиксирована двойная ошибка ECC Регистр обновляется только когда соответствующий бит статуса прерывания равен "0" (AHB5_SRAM_IRQ_STATUS[5]==0)
0x068	AHB5_SRAM_DED_ADDR_2	RO	32’h0	Регистр адреса при обращении, по которому (по шине AHB2) зафиксирована двойная ошибка ECC Регистр обновляется только когда соответствующий бит статуса прерывания равен "0" (AHB5_SRAM_IRQ_STATUS[9]==0)
0x06C	AHB5_SRAM_DED_ADDR_3	RO	32’h0	Регистр адреса при обращении, по которому (по шине AHB3) зафиксирована двойная ошибка ECC Регистр обновляется только когда соответствующий бит статуса прерывания равен "0" (AHB5_SRAM_IRQ_STATUS[13]==0)
0x070-0x07C	-	-	32’h0	Резерв
0x080	AHB5_SRAM_CFG_SRAM_DATA_0[1)]	RW	32’h0	Регистр управления характеристиками и состоянием памяти данных SRAM0
0x084	AHB5_SRAM_CFG_SRAM_DATA_1[1)]	RW	32’h0	Регистр управления характеристиками и состоянием памяти данных SRAM1
0x088	AHB5_SRAM_CFG_SRAM_DATA_2[1)]	RW	32’h0	Регистр управления характеристиками и состоянием памяти данных SRAM2
0x08C	AHB5_SRAM_CFG_SRAM_DATA_3[1)]	RW	32’h0	Регистр управления характеристиками и состоянием памяти данных SRAM3
0x090-0x09C	-	-	32’h0	Резерв
0x0A0	AHB5_SRAM_CFG_SRAM_ECC_0[1)]	RW	32’h0	Регистр управления характеристиками и состоянием памяти кодов детектирования коррекции ошибок SRAM0
0x0A4	AHB5_SRAM_CFG_SRAM_ECC_1[1)]	RW	32’h0	Регистр управления характеристиками и состоянием памяти кодов детектирования коррекции ошибок SRAM1
0x0A8	AHB5_SRAM_CFG_SRAM_ECC_2[1)]	RW	32’h0	Регистр управления характеристиками и состоянием памяти кодов детектирования коррекции ошибок SRAM2

Адрес	Название	Доступ	RST	Описание
0x0AC	AHB5_SRAM_CFG_SRAM_ECC_3[1)]	RW	32'h0	Регистр управления характеристиками и состоянием памяти кодов детектирования коррекции ошибок SRAM3
0x0B0-0x0BC	-	-	32'h0	Резерв
0x0C0	AHB5_SRAM_POWER_TIMING[1)]	RW	32'h8_0140	Регистр установки временных задержек при переключении характеристик и состояний памятей
0x0C4	AHB5_SRAM_PROGR_ERASE[1)]	WO(W1C)	32'h0	Регистр запуска прогрмного стирания памятей
0x0C8-0xFFC	-	-	32'he7707_ad7	Резерв

Примечания:

1) Описание регистра приведено ниже;

1.3.5.2.1.5.2 Описание полей регистров подсистемы памяти SRAM

1.3.5.2.1.5.2.1 AHB5_SRAM_IRQ_STATUS (0x010)

Описание полей регистра AHB5_SRAM_IRQ_STATUS приведено в таблице 1.115.

Таблица 1.115 – Поля регистра AHB5_SRAM_IRQ_STATUS

Бит	Поле	RST	Описание
31:16	-	-	Резерв
SRAM3			
15	-	-	Резерв
14	sram3_irq_st_clr_compl	0	Завершено стирание банка памяти SRAM3
13	sram3_irq_st_ded	0	Зафиксирована двойная ошибка ECC при чтении из банка памяти SRAM3
12	sram3_irq_st_sec	0	Зафиксирована одиночная ошибка ECC при чтении из банка памяти SRAM3
SRAM2			
11	-	-	Резерв
10	sram2_irq_st_clr_compl	0	Завершено стирание банка памяти SRAM2
9	sram2_irq_st_ded	0	Зафиксирована двойная ошибка ECC при чтении из банка памяти SRAM2
8	sram2_irq_st_sec	0	Зафиксирована одиночная ошибка ECC при чтении из банка памяти SRAM2
SRAM1			
7	-	-	Резерв
6	sram1_irq_st_clr_compl	0	Завершено стирание банка памяти SRAM1
5	sram1_irq_st_ded	0	Зафиксирована двойная ошибка ECC при чтении из банка памяти SRAM1
4	sram1_irq_st_sec	0	Зафиксирована одиночная ошибка ECC при чтении из банка памяти SRAM1

Бит	Поле	RST	Описание
SRAM0			
3	-	-	Резерв
2	sram0_irq_st_clr_compl	0	Завершено стирание банка памяти SRAM0
1	sram0_irq_st_ded	0	Зафиксирована двойная ошибка ECC при чтении из банка памяти SRAM0
0	sram0_irq_st_sec	0	Зафиксирована одиночная ошибка ECC при чтении из банка памяти SRAM0

1.3.5.2.1.5.2.2 AHB5_SRAM_IRQ_MASK (0x014)

Описание полей регистра AHB5_SRAM_IRQ_MASK приведено в таблице 1.116.

При значении поля "sram**irq_mask**" равном 1, возникновение 1 в соответствующем статусе вызовет возникновение прерывания (пример: при (AHB5_SRAM_IRQ_MASK[0] == 1) & (AHB5_SRAM_IRQ_STATUS[0] == 1) выход блока irq_o = 1).

Таблица 1.116 – Поля регистра AHB5_SRAM_IRQ_MASK

Бит	Поле	RST	Описание
31:16	-	-	Резерв
SRAM3			
15	-	-	Резерв
14	sram3_irq_mask_clr_compl	0	Разрешение прерывания при статусе "Завершено стирание банка памяти SRAM3"
13	sram3_irq_mask_ded	0	Разрешение прерывания при статусе "Зафиксирована двойная ошибка ECC при чтении из банка памяти SRAM3"
12	sram3_irq_mask_sec	0	Разрешение прерывания при статусе "Зафиксирована одиночная ошибка ECC при чтении из банка памяти SRAM3"
SRAM2			
11	-	-	Резерв
10	sram2_irq_mask_clr_compl	0	Разрешение прерывания при статусе "Завершено стирание банка памяти SRAM2"
9	sram2_irq_mask_ded	0	Разрешение прерывания при статусе "Зафиксирована двойная ошибка ECC при чтении из банка памяти SRAM2"
8	sram2_irq_mask_sec	0	Разрешение прерывания при статусе "Зафиксирована одиночная ошибка ECC при чтении из банка памяти SRAM2"
SRAM1			
7	-	-	Резерв
6	sram1_irq_mask_clr_compl	0	Разрешение прерывания при статусе "Завершено стирание банка памяти SRAM1"
5	sram1_irq_mask_ded	0	Разрешение прерывания при статусе "Зафиксирована двойная ошибка ECC при чтении из банка памяти SRAM1"

Бит	Поле	RST	Описание
4	sram1_irq_mask_sec	0	Разрешение прерывания при статусе "Зафиксирована одиночная ошибка ECC при чтении из банка памяти SRAM1"
SRAM0			
3	-	-	Резерв
2	sram0_irq_mask_clr_compl	0	Разрешение прерывания при статусе "Завершено стирание банка памяти SRAM0"
1	sram0_irq_mask_ded	0	Разрешение прерывания при статусе "Зафиксирована двойная ошибка ECC при чтении из банка памяти SRAM0"
0	sram0_irq_mask_sec	0	Разрешение прерывания при статусе "Зафиксирована одиночная ошибка ECC при чтении из банка памяти SRAM0"

1.3.5.2.1.5.2.3 AHB5_SRAM_IRQ_CLEAR (0x018)

Описание полей регистра AHB5_SRAM_IRQ_CLEAR приведено в таблице 1.117.

Запись "1" в "sram**irq_clear**" сбрасывает соответствующий бит регистра AHB5_SRAM_IRQ_STATUS. Запись "0" игнорируется.

Таблица 1.117 – Поля регистра AHB5_SRAM_IRQ_CLEAR

Бит	Поле	RST	Описание
31:16	-	-	Резерв
SRAM3			
15	-	-	Резерв
14	sram3_irq_clear_clr_compl	0	Сброс статуса "Завершено стирание банка памяти SRAM3"
13	sram3_irq_clear_ded	0	Сброс статуса "Зафиксирована двойная ошибка ECC при чтении из банка памяти SRAM3"
12	sram3_irq_clear_sec	0	Сброс статуса "Зафиксирована одиночная ошибка ECC при чтении из банка памяти SRAM3"
SRAM2			
11	-	-	Резерв
10	sram2_irq_clear_clr_compl	0	Сброс статуса "Завершено стирание банка памяти SRAM2"
9	sram2_irq_clear_ded	0	Сброс статуса "Зафиксирована двойная ошибка ECC при чтении из банка памяти SRAM2"
8	sram2_irq_clear_sec	0	Сброс статуса "Зафиксирована одиночная ошибка ECC при чтении из банка памяти SRAM2"
SRAM1			
7	-	-	Резерв
6	sram1_irq_clear_clr_compl	0	Сброс статуса "Завершено стирание банка памяти SRAM1"
5	sram1_irq_clear_ded	0	Сброс статуса "Зафиксирована двойная ошибка ECC при чтении из банка памяти SRAM1"

Бит	Поле	RST	Описание
4	sram1_irq_clear_sec	0	Сброс статуса "Зафиксирована одиночная ошибка ECC при чтении из банка памяти SRAM1"
SRAM0			
3	-	-	Резерв
2	sram0_irq_clear_clr_compl	0	Сброс статуса "Завершено стирание банка памяти SRAM0"
1	sram0_irq_clear_ded	0	Сброс статуса "Зафиксирована двойная ошибка ECC при чтении из банка памяти SRAM0"
0	sram0_irq_clear_sec	0	Сброс статуса "Зафиксирована одиночная ошибка ECC при чтении из банка памяти SRAM0"

1.3.5.2.1.5.2.4 AHB5_SRAM_IRQ_TEST (0x01C)

Регистр используется только в тестовых целях.

Описание полей регистра AHB5_SRAM_IRQ_TEST приведено в таблице 1.118.

Запись "1" в "sram**irq_test***" устанавливает соответствующий бит регистра AHB5_SRAM_IRQ_STATUS. Запись "0" игнорируется.

Таблица 1.118 – Поля регистра AHB5_SRAM_IRQ_TEST

Бит	Поле	RST	Описание
31:16	-	-	Резерв
SRAM3			
15	-	-	Резерв
14	sram3_irq_test_clr_compl	0	Установка статуса "Завершено стирание банка памяти SRAM3"
13	sram3_irq_test_ded	0	Установка статуса "Зафиксирована двойная ошибка ECC при чтении из банка памяти SRAM3"
12	sram3_irq_test_sec	0	Установка статуса "Зафиксирована одиночная ошибка ECC при чтении из банка памяти SRAM3"
SRAM2			
11	-	-	Резерв
10	sram2_irq_test_clr_compl	0	Установка статуса "Завершено стирание банка памяти SRAM2"
9	sram2_irq_test_ded	0	Установка статуса "Зафиксирована двойная ошибка ECC при чтении из банка памяти SRAM2"
8	sram2_irq_test_sec	0	Установка статуса "Зафиксирована одиночная ошибка ECC при чтении из банка памяти SRAM2"
SRAM1			
7	-	-	Резерв
6	sram1_irq_test_clr_compl	0	Установка статуса "Завершено стирание банка памяти SRAM1"
5	sram1_irq_test_ded	0	Установка статуса "Зафиксирована двойная ошибка ECC при чтении из банка памяти SRAM1"

Бит	Поле	RST	Описание
4	sram1_irq_test_sec	0	Установка статуса "Зафиксирована одиночная ошибка ECC при чтении из банка памяти SRAM1"
SRAM0			
3	-	-	Резерв
2	sram0_irq_test_clr_compl	0	Установка статуса "Завершено стирание банка памяти SRAM0"
1	sram0_irq_test_ded	0	Установка статуса "Зафиксирована двойная ошибка ECC при чтении из банка памяти SRAM0"
0	sram0_irq_test_sec	0	Установка статуса "Зафиксирована одиночная ошибка ECC при чтении из банка памяти SRAM0"

1.3.5.2.1.5.2.5 AHB5_SRAM_SETTINGS (0x020)

Описание полей регистра AHB5_SRAM_SETTINGS приведено в таблице 1.119.

Таблица 1.119 – Поля регистра AHB5_SRAM_SETTINGS

Бит	Поле	RST	Описание
31:16	-	-	Резерв
SRAM3			
15	sram3_cfg_mar_syscr	0	Использовать значение поле "cfg_mar" из SYSCR для памяти SRAM3
14	sram3_ded_resp_dis	0	Выключение ошибки на шине AHB (HRESP) при возникновении двойной ошибок для памяти SRAM3. При sram3_ded_resp_dis = 1 ошибка HRESP формироваться не будет
13	sram3_rd_reg_slice_en	0	Включение регистровой станции для чтения из памяти SRAM3
12	sram3_ecc_en	0	Включение кодов коррекции ошибок для памяти SRAM3
SRAM2			
11	sram2_cfg_mar_syscr	0	Использовать значение поле "cfg_mar" из SYSCR для памяти SRAM2
10	sram2_ded_resp_dis	0	Выключение ошибки на шине AHB (HRESP) при возникновении двойной ошибок для памяти SRAM2. При sram2_ded_resp_dis = 1 ошибка HRESP формироваться не будет
9	sram2_rd_reg_slice_en	0	Включение регистровой станции для чтения из памяти SRAM2
8	sram2_ecc_en	0	Включение кодов коррекции ошибок для памяти SRAM2
SRAM1			
7	sram1_cfg_mar_syscr	0	Использовать значение поле "cfg_mar" из SYSCR для памяти SRAM1
6	sram1_ded_resp_dis	0	Выключение ошибки на шине AHB (HRESP) при возникновении двойной ошибок для памяти SRAM1. При sram1_ded_resp_dis = 1 ошибка HRESP формироваться не будет

Бит	Поле	RST	Описание
5	sram1_rd_reg_slice_en	0	Включение регистровой станции для чтения из памяти SRAM1
4	sram1_ecc_en	0	Включение кодов коррекции ошибок для памяти SRAM1
SRAM0			
3	sram0_cfg_mar_syscr	0	Использовать значение поле "cfg_mar" из SYSCR для памяти SRAM0
2	sram0_ded_resp_dis	0	Выключение ошибки на шине AHB (HRESP) при возникновении двойной ошибок для памяти SRAM0. При sram0_ded_resp_dis = 1 ошибка HRESP формироваться не будет
1	sram0_rd_reg_slice_en	0	Включение регистровой станции для чтения из памяти SRAM0
0	sram0_ecc_en	0	Включение кодов коррекции ошибок для памяти SRAM0

1.3.5.2.1.5.2.6 AHB5_SRAM_CFG_SRAM_DATA_0, 1, 2, 3 (0x080-0x08C) и AHB5_SRAM_CFG_SRAM_ECC_0, 1, 2, 3 (0x0A0-0x08AC)

Описание полей регистров AHB5_SRAM_CFG_SRAM_DATA_0, 1, 2, 3 и AHB5_SRAM_CFG_SRAM_ECC_0, 1, 2, 3 приведено в таблице 1.120.

Значение полей cfg_mar после сброса (RST) задано для старта на напряжении 1,1В.

Запись в данный регистр при sys_erase_active == 1 приведет к возникновению ошибки на шине APB (pslverr = 1).

Таблица 1.120 – Поля регистров AHB5_SRAM_CFG_SRAM_DATA(ECC)

Бит	Поле	RST	Описание
31:30	-	-	Резерв
29	sys_block	0	0 - штатная работа. 1 - TAMP выставил системную блокировка доступа к памяти. Запись в память игнорируется. Чтение возвращает значение "0". Если в блоке SYSCR в регистре CFG_PSLVERR[cfg_pslverr] == 1, то в случае обращений к памяти при sys_block == 1 будет формироваться ошибка на шине AHB (HRESP = 1).
28	sys_erase_active	0	0 - штатная работа. 1 - пришло событие от TAMP - производится программное стирание памяти Если в блоке SYSCR в регистре CFG_PSLVERR[cfg_pslverr] == 1, то в случае обращений к памяти при sys_erase_i == 1 будет формироваться ошибка на шине AHB (HRESP = 1). Значение sys_erase_active одинаковое для всех регистров AHB5_SRAM_CFG_SRAM_*

Бит	Поле	RST	Описание
27:25	-	-	Резерв
24	rm_complite	0	Изменение настроек cfg_mar завершено (RO - поле доступно только для чтения), сразу после сброса поле имеет значение 0, затем после числа тактов, прописанного в регистре AHB5_SRAM_POWER_TIMING, значение переходит в 1
23	-	-	Резерв
22	ls_complite	0	Выход из состояния LS завершен (RO - поле доступно только для чтения), сразу после сброса поле имеет значение 0, затем после числа тактов, прописанного в регистре AHB5_SRAM_POWER_TIMING, значение переходит в 1
21	-	-	Резерв
20	sd_ds_complite	0	Выход из состояний SD и DS завершен (RO - поле доступно только для чтения), сразу после сброса поле имеет значение 0, затем после числа тактов, прописанного в регистре AHB5_SRAM_POWER_TIMING, значение переходит в 1
19	cfg_mar_syscr	-	Дублирование поля sram*_cfg_mar_syscr регистра AHB5_SRAM_SETTINGS (RO - поле доступно только для чтения)
18:17	-	-	Резерв
16:3	cfg_mar	14’h1320	Поле задает настройку памяти зависящую от напряжения питания: * для 1,1В cfg_mar должно равняться 14’h1320; * для 1,0В cfg_mar должно равняться 14’h1228; * для 0,9В cfg_mar должно равняться 14’h1070;
2	ls	0	0 - нормальная работа памяти Запись 1 - перевод памяти в состояние "Light sleep" При изменении значения поля с 1 на 0 происходит вывод памяти из состояния "Light sleep" (время выхода задается в регистре AHB5_SRAM_POWER_TIMING)
1	ds	0	0 - нормальная работа памяти Запись 1 - перевод памяти в состояние "Deep sleep" При изменении значения поля с 1 на 0 происходит вывод памяти из состояния "Deep sleep" (время выхода задается в регистре AHB5_SRAM_POWER_TIMING)
0	sd	0	0 - нормальная работа памяти Запись 1 - перевод памяти в состояние "Shut Down" (отключение) При изменении значения поля с 1 на 0 происходит вывод памяти из состояния "Shut Down" (время выхода задается в регистре AHB5_SRAM_POWER_TIMING)

1.3.5.2.1.5.2.7 AHB5_SRAM_POWER_TIMING (0x0C0)

Описание полей регистра AHB5_SRAM_POWER_TIMING приведено в таблице 1.121. Рекомендуется использовать значения заданные по умолчанию.

Таблица 1.121 – Поля регистра AHB5_SRAM_POWER_TIMING

Бит	Поле	RST	Описание
31:16	tf_rm	16’h8	Количество тактов необходимое для изменения настроек cfg_mar
15:12	-	-	Резерв
11:8	tf_ls	4’h1	Количество тактов необходимое для выхода из состояния "Light sleep"
7:0	tf_sd_ds	8’h40	Количество тактов необходимое для выхода из состояний "Shut Down" или "Deep sleep"

1.3.5.2.1.5.2.8 AHB5_SRAM_PROGR_ERASE (0x0C4)

Описание полей регистра AHB5_SRAM_PROGR_ERASE приведено в таблице 1.122.

Таблица 1.122 – Поля регистра AHB5_SRAM_PROGR_ERASE

Бит	Поле	RST	Описание
31:16	-	-	Резерв
SRAM3			
15:13	-	-	Резерв
12	progr_erase_3	1’b0	Запуск стирания 3 банка памяти. Стирание памяти данных происходит только при режиме нормальной работа память - заданном в регистре AHB5_SRAM_CFG_SRAM_DATA_3. Стирание памяти кодов коррекции ошибок происходит только при AHB5_SRAM_SETTINGS[sram3_ecc_en] = 1 и режиме нормальной работа память - заданном в регистре AHB5_SRAM_CFG_SRAM_ECC_3
SRAM2			
11:9	-	-	Резерв
8	progr_erase_2	1’b0	Запуск стирания 2 банка памяти. Стирание памяти данных происходит только при режиме нормальной работа память - заданном в регистре AHB5_SRAM_CFG_SRAM_DATA_2. Стирание памяти кодов коррекции ошибок происходит только при AHB5_SRAM_SETTINGS[sram2_ecc_en] = 1 и режиме нормальной работа память - заданном в регистре AHB5_SRAM_CFG_SRAM_ECC_2
SRAM1			
7:5	-	-	Резерв
4	progr_erase_1	1’b0	Запуск стирания 1 банка памяти. Стирание памяти данных происходит только при режиме нормальной работа память - заданном в регистре AHB5_SRAM_CFG_SRAM_DATA_1. Стирание памяти кодов коррекции ошибок происходит только при AHB5_SRAM_SETTINGS[sram1_ecc_en] = 1 и режиме нормальной работа память - заданном в регистре AHB5_SRAM_CFG_SRAM_ECC_1
SRAM0			

Бит	Поле	RST	Описание
3:1	-	-	Резерв
0	progr_erase_0	1’b0	Запуск стирания 0 банка памяти. Стирание памяти данных происходит только при режиме нормальной работа память - заданном в регистре AHB5_SRAM_CFG_SRAM_DATA_0. Стирание памяти кодов коррекции ошибок происходит только при AHB5_SRAM_SETTINGS[sram0_ecc_en] = 1 и режиме нормальной работа память - заданном в регистре AHB5_SRAM_CFG_SRAM_ECC_0

1.3.5.2.2 Статическая память батарейного домена питания (BKPSRAM)

1.3.5.2.2.1 Описание подсистемы памяти BKPSRAM

Памяти BKPSRAM имеет полный объем 2КБайт (512*32 бит). При этом она может использоваться в двух режимах:

– без кодов коррекции ошибок (ECC - настройка задается полем bkpsram_ecc_en в регистре BKPSRAM_ECC) - вся память используется для хранения данных;

– с ECC - 1КБайт памяти используется для хранения данных и 1КБайт используется для хранения ECC (коррекция одиночной ошибки и обнаружение двойной ошибки на каждый байт данных).

1.3.5.2.2.2 Режимы пониженного энергопотребления (LS, DS, SD)

Память имеет 3 режима пониженного энергопотребления:

– LS - Light Sleep - "легки сон" - режим характеризуется быстрым временем выхода на рабочий режим;

– DS - Deep Sleep - "глубокий сон" - временем выхода на рабочий режим дольше чем у LS;

– SD - Shut Down - память выключена (записанные значения не сохраняются).

Если память данных находится в режимах пониженного энергопотребления (LS, DS, SD), а также если включено ECC и память для ECC находится в режимах пониженного энергопотребления (LS, DS, SD), то запись в память игнорируется, чтение возвращает значение "32’h000_e7707". Если в блоке SYSCR в регистре CFG_PSLVERR[cfg_pslverr] == 1, то в случае обращений к памяти при LS, DS, SD будет формироваться ошибка на шине APB (PSLVERR = 1).

1.3.5.2.2.2.1 Стирание памяти

При возникновении сигнала sys_erase_i от блока TAMP происходит стирание всей памяти (и памяти данных и памяти коды коррекции ошибок). При этом при sys_erase_i == 1 запись в память игнорируется, чтение возвращает значение "32'h000_e7707".

Если в блоке SYSCR в регистре CFG_PSLVERR[cfg_pslverr] == 1, то в случае обращений к памяти при sys_erase_i == 1 будет формироваться ошибка на шине APB (PSLVERR = 1).

Если в этот момент память находится в режиме пониженного потребления (LS, DS, SD), то память переводится в рабочий режим и после этого память стирается (в память записываются значение 32'h0000_0000).

1.3.5.2.2.3 Перенастройка памяти при изменении напряжения питания

В зависимости от напряжения питания микросхемы должно быть выставлено необходимое значение в регистре BKPSRAM_CFG_SRAM в поле cfg_mar (см. 1.3.5.2.2.4.2.1).

1.3.5.2.2.4 Описание регистров BKPSRAM

Задание конфигурационных параметров и отображение текущего состояния BKPSRAM ведется через регистры. Доступ к регистрам осуществляется посредством системного интерфейса управления (интерфейс APB 4/5 slave). Каждое обращение к регистрам имеет выравнивание до 32-х разрядов. Карта регистров приведена в таблице 1.123.

При обращении к регистрам СФ-блока по зарезервированным адресам никаких ошибок не возникает. При попытке записи в регистры, предназначенные только для чтения, операция записи завершается успешно, однако, состояние регистров не изменяется.

Настройка BKPSRAM проводится через 2 набора регистров:

– Настройка режимов энергопотребления проводится в "Backup SRAM Control Registers vbat"(BKPSRAM_CR_VBAT)

– Настройка ECC и обработка прерываний об ошибка ECC проводится в "Backup SRAM Control Registers ECC"(BKPSRAM_CR_ECC)

1.3.5.2.2.4.1 Карта регистров BKPSRAM_CR_VBAT

Таблица 1.123 – Карта регистров BKPSRAM_CR_VBAT

Адрес	Название	Доступ	RST	Описание
0x000-0x07C	-	-	32'h0	Резерв
0x080	BKPSRAM_CFG_SRAM	RW	32'h0	Регистр управления характеристиками и состоянием памяти BKPSRAM

Адрес	Название	Доступ	RST	Описание
0x0B0-0x0BC	-	-	32’h0	Резерв
0x0C0	BKPSRAM_POWER_TIMING	RW	32’h1_0040	Регистр установки временных задержек при переключении характеристик и состояний памятей
0x0C4-0xFFC	-	-	32’he7707_ad7	Резерв

1.3.5.2.2.4.2 Описание полей регистров BKPSRAM_CR_VBAT

1.3.5.2.2.4.2.1 BKPSRAM_CFG_SRAM (0x080)

Описание полей регистра BKPSRAM_CFG_SRAM приведено в таблице 1.124.

Значение полей cfg_mar после сброса (RST) задано для старта на напряжении 1,1В.

Запись в данный регистр при sys_erase_active == 1 приведет к возникновению ошибки на шине APB (pslverr = 1).

Таблица 1.124 – Поля регистра BKPSRAM_CFG_SRAM

Бит	Поле	RST	Описание
31:30	-	-	Резерв
29	sys_block	0	0 - штатная работа. 1 - TAMP выставил системную блокировка доступа к памяти. Запись в память игнорируется. Чтение возвращает значение "0". Если в блоке SYSCR в регистре CFG_PSLVERR[cfg_pslverr] == 1, то в случае обращений к памяти при sys_block == 1 будет формироваться ошибка на шине APB (PSLVERR = 1).
28	sys_erase_active	0	0 - штатная работа. 1 - пришло событие от TAMP - производится программное стирание памяти Если в блоке SYSCR в регистре CFG_PSLVERR[cfg_pslverr] == 1, то в случае обращений к памяти при sys_erase_i == 1 будет формироваться ошибка на шине APB (PSLVERR = 1).
27:21	-	-	Резерв
20	cfg_complite	0	Настройка памяти завершена (RO - поле доступно только для чтения), сразу после сброса поле имеет значение 0, затем после числа тактов, прописанного в регистре BKPSRAM_POWER_TIMING, значение переходит в 1
19:17	-	-	Резерв

Бит	Поле	RST	Описание
16:3	cfg_mar	14’h1320	Поле задает настройку памяти зависящую от напряжения питания: * для 1,1В cfg_mar должно равняться 14’h1320; * для 1,0В cfg_mar должно равняться 14’h1228; * для 0,9В cfg_mar должно равняться 14’h1070;
2	ls	0	0 - нормальная работа памяти Запись 1 - перевод памяти в состояние "Light sleep" При изменении значения поля с 1 на 0 происходит вывод памяти из состояния "Light sleep" (время выхода задается в регистре BKPSRAM_POWER_TIMING)
1	ds	0	0 - нормальная работа памяти Запись 1 - перевод памяти в состояние "Deep sleep" При изменении значения поля с 1 на 0 происходит вывод памяти из состояния "Deep sleep" (время выхода задается в регистре BKPSRAM_POWER_TIMING)
0	sd	0	0 - нормальная работа памяти Запись 1 - перевод памяти в состояние "Shut Down" (отключение) При изменении значения поля с 1 на 0 происходит вывод памяти из состояния "Shut Down" (время выхода задается в регистре BKPSRAM_POWER_TIMING)

1.3.5.2.2.4.2.2 BKPSRAM_POWER_TIMING (0x0C0)

Описание полей регистра BKPSRAM_POWER_TIMING приведено в таблице 1.125. Рекомендуется использовать значение заданное по умолчанию.

Таблица 1.125 – Поля регистра BKPSRAM_POWER_TIMING

Бит	Поле	RST	Описание
31:8	-	-	Резерв
7:0	tf	8’h40	Количество тактов необходимое для настройка памяти (выхода из состояний "Shut Down" или "Deep sleep" или "Light sleep" или изменения cfg_mar)

1.3.5.2.2.4.3 Карта регистров BKPSRAM_CR_ECC

Таблица 1.126 – Карта регистров BKPSRAM_CR_ECC

Адрес	Название	Доступ	RST	Описание
0x000-0x00C	-	-	32’he7707_ad7	Резерв
0x010	BKPSRAM_IRQ_STATUS [1)]	RO	32’h0	Регистр статуса прерывания
0x014	BKPSRAM_IRQ_MASK[1)]	RW	32’h0	Регистр маски прерывания
0x018	BKPSRAM_IRQ_CLEAR[1)]	WO(W1C)	32’h0	Регистр сброса статуса прерывания
0x01C	BKPSRAM_IRQ_TEST[1)]	WO	32’h0	Регистр установки статуса прерывания
0x020	BKPSRAM_ECC[1)]	RW	32’h0	Регистр включения ECC

Адрес	Название	Доступ	RST	Описание
0x024-0x03C	-	-	32’he7707_ad7	Резерв
0x040	BKPSRAM_SEC_ADDR	RO	32’h0	Регистр адреса при обращении, по которому зафиксирована одиночная ошибка ECC Регистр обновляется только когда соответствующий бит статуса прерывания равен "0" (BKPSRAM_IRQ_STATUS[0]==0)
0x044-0x05C	-	-	32’h0	Резерв
0x060	BKPSRAM_DED_ADDR	RO	32’h0	Регистр адреса при обращении, по которому зафиксирована двойная ошибка ECC Регистр обновляется только когда соответствующий бит статуса прерывания равен "0" (BKPSRAM_IRQ_STATUS[1]==0)
0x064-0xFFC	-	-	32’he7707_ad7	Резерв

Примечания:

1) Описание регистра приведено ниже;

1.3.5.2.2.4.4 Описание полей регистров BKPSRAM_CR_ECC

1.3.5.2.2.4.4.1 BKPSRAM_IRQ_STATUS (0x010)

Описание полей регистра BKPSRAM_IRQ_STATUS приведено в таблице 1.127.

Таблица 1.127 – Поля регистра BKPSRAM_IRQ_STATUS

Бит	Поле	RST	Описание
31:2	-	-	Резерв
1	bkpsram_irq_st_ded	0	Зафиксирована двойная ошибка ECC при чтении из банка памяти BKPSRAM
0	bkpsram_irq_st_sec	0	Зафиксирована одиночная ошибка ECC при чтении из банка памяти BKPSRAM

1.3.5.2.2.4.4.2 BKPSRAM_IRQ_MASK (0x014)

Описание полей регистра BKPSRAM_IRQ_MASK приведено в таблице 1.128.

При значении поля bkpsram_irq_mask_*" равном 1, возникновение 1 в соответствующем статусе вызовет возникновение прерывания (пример: при (BKPSRAM_IRQ_MASK[0] == 1) & (BKPSRAM_IRQ_STATUS[0] == 1) выход блока irq_o = 1).

Таблица 1.128 – Поля регистра BKPSRAM_IRQ_MASK

Бит	Поле	RST	Описание
31:2	-	-	Резерв
1	bkpsram_irq_mask_ded	0	Разрешение прерывания при статусе "Зафиксирована двойная ошибка ECC при чтении из банка памяти BKPSRAM"

Бит	Поле	RST	Описание
0	bkpsram_irq_mask_sec	0	Разрешение прерывания при статусе "Зафиксирована одиночная ошибка ECC при чтении из банка памяти BKPSRAM"

1.3.5.2.2.4.4.3 BKPSRAM_IRQ_CLEAR (0x018)

Описание полей регистра BKPSRAM_IRQ_CLEAR приведено в таблице 1.129.

Запись "1" в bkpsram_irq_clear_*" сбрасывает соответствующий бит регистра BKPSRAM_IRQ_STATUS. Запись "0" игнорируется.

Таблица 1.129 – Поля регистра BKPSRAM_IRQ_CLEAR

Бит	Поле	RST	Описание
31:2	-	-	Резерв
1	bkpsram_irq_clear_ded	0	Сброс статуса "Зафиксирована двойная ошибка ECC при чтении из банка памяти BKPSRAM"
0	bkpsram_irq_clear_sec	0	Сброс статуса "Зафиксирована одиночная ошибка ECC при чтении из банка памяти BKPSRAM"

1.3.5.2.2.4.4.4 BKPSRAM_IRQ_TEST (0x01C)

Регистр используется только в тестовых целях.

Описание полей регистра BKPSRAM_IRQ_TEST приведено в таблице 1.130.

Запись "1" в bkpsram_irq_test_*" устанавливает соответствующий бит регистра BKPSRAM_IRQ_STATUS. Запись "0" игнорируется.

Таблица 1.130 – Поля регистра BKPSRAM_IRQ_TEST

Бит	Поле	RST	Описание
31:2	-	-	Резерв
1	bkpsram_irq_test_ded	0	Установка статуса "Зафиксирована двойная ошибка ECC при чтении из банка памяти BKPSRAM"
0	bkpsram_irq_test_sec	0	Установка статуса "Зафиксирована одиночная ошибка ECC при чтении из банка памяти BKPSRAM"

1.3.5.2.2.4.4.5 BKPSRAM_ECC (0x020)

Описание полей регистра BKPSRAM_ECC приведено в таблице 1.131.

Таблица 1.131 – Поля регистра BKPSRAM_ECC

Бит	Поле	RST	Описание
31:3	-	-	Резерв

Бит	Поле	RST	Описание
2	bkpsram_ded_resp_dis	0	Выключение ошибки на шине APB (PSLVERR) при возникновении двойной ошибоки приобращении к памяти BKPSRAM. При bkpsram_ded_resp_dis = 1 ошибка PSLVERR формироваться не будет
1	-	-	Резерв
0	bkpsram_ecc_en	0	Включение кодов коррекции ошибок для памяти BKPSRAM

1.3.6 Системный конфигурационный контроллер (SYSCR)

В МК Optimal+ имеется системный контроллер SYSCR .

Таблица 1.132 – Список регистров системного контроллера

Смещение	Название	Описание
0x0	FCELL_SCHMITT	Включение триггера Шмитта для выводов с функцией WKP_15 - WKP_0
0x4	SD_SEL	Выбор режима работы между eMMC/SD и XSPI
0x8	BOOTM	Регистр выбора режима загрузки микросхемы
0xC	CFG_SRAM	Регистр выбора режима работы памятей
0x10	_	Зарезервировано
0x14	CFG_PSLVERR	Регистр, отвечающий за значение сигнала ошибки интерфейса APB
0x18	PD_CTRL_CFG	Регистр управления настройками контроллера отключаемого домена питания

1.3.6.1 FCELL_SCHMITT

Таблица 1.133 – Описание регистра FCELL_SCHMITT

Биты	Название	Значение после сброса	Доступ	Описание
[31:16]	_	_	_	Резерв
[15:0]	FCELL_SCHMITT	16’b0	R/W	1 - включение триггера Шмитта для выводов с функцией WKP_15 - WKP_0 0 - выключение

1.3.6.2 SD_SEL

Таблица 1.134 – Описание регистра SD_SEL

Биты	Название	Значение после сброса	Доступ	Описание
[31:2]	_	_	_	Резерв
[1]	SD_SEL_1	1’b1	R/W	1 - работает eMMC_1/SD_1 0 - работает xSPI_1
[0]	SD_SEL_0	1’b0	R/W	1 - работает eMMC_0/SD_0 0 - работает xSPI_0

1.3.6.3 BOOTM

Регистр BOOTM содержит биты, отвечающие за режим загрузки микросхемы. Все поля имеют значение после сброса равные значению соответствующих выводом микросхемы в момент снятия активного уровня сброса. Ниже преставлены описание полей регистра и расшифровка режимов загрузки.

Таблица 1.135 – Описание регистра BOOTM

Биты	Название	Значение после сброса	Доступ	Описание
[31:1]	_	_	_	Резерв
[0]	BOOT0	Значение с пада BOOT0	R/W	Бит выбора адреса старта процессорного ядра

1.3.6.4 CFG_SRAM

Регистр содержит конфигурацию для макросов памяти в МК Optimal+.

Таблица 1.136 – Описание регистра CFG_SRAM

Биты	Название	Значение после сброса	Доступ	Описание
[31:17]	_	_	_	Резерв
[16:3]	cfg_mar	14’h1320	R/W	Поле задает настройки памятей зависящие от напряжения питания: * для 1,1В cfg_mar должно равняться 14’h1320; * для 1,0В cfg_mar должно равняться 14’h1228; * для 0,9В cfg_mar должно равняться 14’h1070;
[2:0]	_	3’b000	R/W	Резерв

1.3.6.5 CFG_PSLVERR

Таблица 1.137 – Описание регистра CFG_PSLVERR

Биты	Название	Значение после сброса	Доступ	Описание
[31:1]	_	_	_	Резерв
[0]	CFG_PSLVERR	1'b1	R/W	1 - при ошибке доступа к устройствам по шине APB выставляется соответствующее значение для сигнала ошибки 0 - ошибка игнорируется

1.3.6.6 PD_CTRL_CFG

Таблица 1.138 – Описание полей регистра PD_CTRL_CFG

Биты	Поле	Доступ	Значение после сброса	Описание
31:8	-	ЧТ	0x0	Зарезервировано
7:4	MEM_ON_DLY	ЧТ/ЗП	0x05	Задержка выхода макроса памяти из режима Shutdown в тактах прескейлера
3:0	PRESCALER	ЧТ/ЗП	0x0A	Перскейлер (делитель синхросигнала) для расчёта задержек

1.3.7 Подсистема управления сигналами сброса и синхросигналами (RCU)

1.3.7.1 Общее описание

Блок управления сигналами сброса и синхросигналами (далее - RCU, Reset and Clock control Unit) управляет различными видами сброса и генерирует все тактовые сигналы для шин и периферийных устройств. Упрощенная структурная схема блока RCU показана на рисунке 1.20.

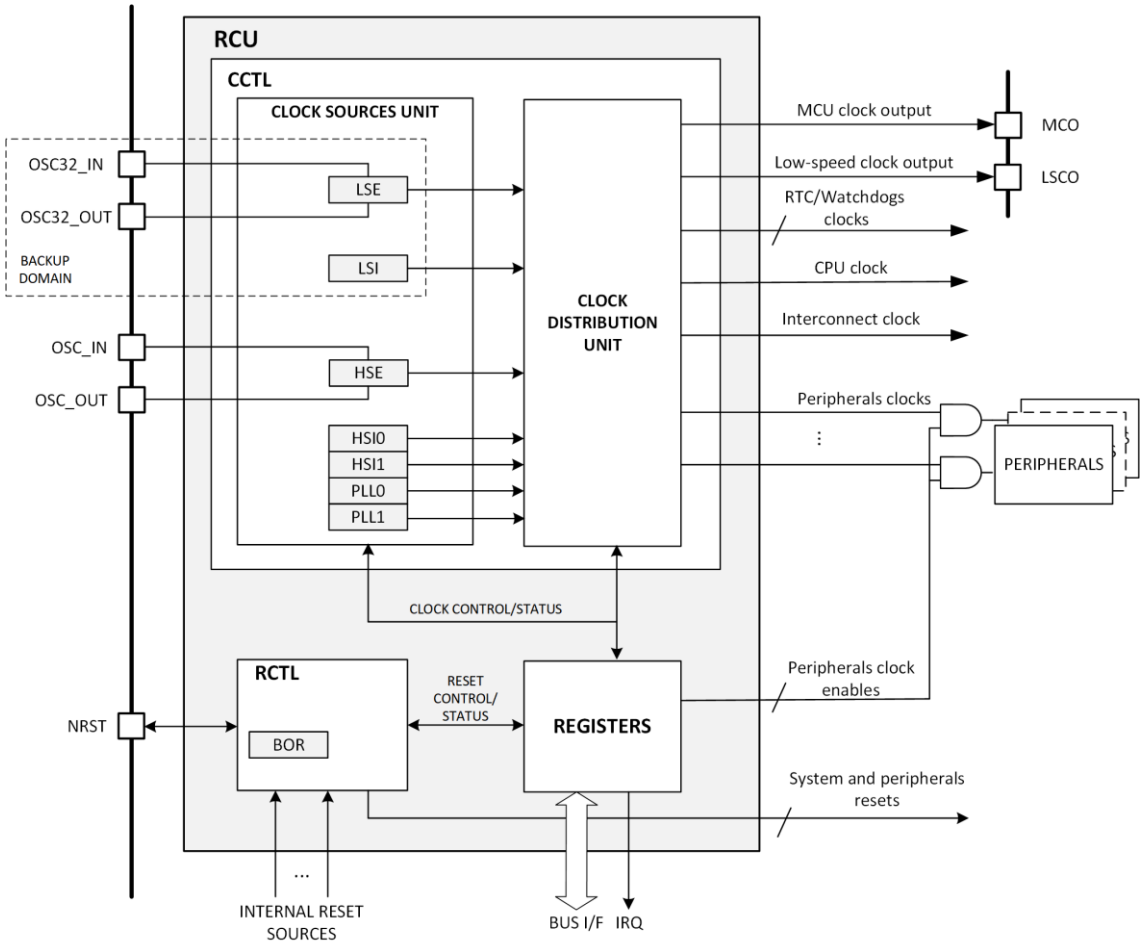


Рисунок 1.20 – Структура блока RCU

Блок RCU состоит из следующих основных частей:

– блок управления синхросигналами (CCTL), который включает в себя блок источников синхросигналов и систему их распределения на основе мультиплексоров и делителей частоты;

– блок управления сигналами сброса (RCTL), отвечающий за формирование системных сигналов сброса на основе различных событий;

– блок программно-доступных регистров, для конфигурации блока и считывания статуса.

1.3.7.2 Внешние выводы блока RCU

В таблице 1.139 приведен список внешних выводов микроконтроллера, которые соединены с блоком RCU.

Таблица 1.139 – Внешние выводы блока RCU

Название вывода	Тип	Описание
NRST	I/O	Сигнал системного сброса
OSC32_IN	I	Вход генератора LSE
OSC32_OUT	O	Выход генератора LSE
OSC_IN	I	Вход генератора HSE
OSC_OUT	O	Выход генератора HSE
MCO	O	Основной выход синхросигнала микроконтроллера
LSCO	O	Выход синхросигнала низкой частоты

1.3.7.3 Блок управления сигналами сброса (RCTL)

Блок управления сигналами сброса Reset Control unit (RCTL) формирует три вида сброса:

– Сброс по питанию;

– Системный сброс;

– Сброс резервного домена Backup.

Сброс по питанию сбрасывает всю систему, за исключением резервного домена. Системный сброс приводит к сбросу ядра процессора и периферийных устройств, за исключением контроллера отладки и резервного домена. Сброс резервного домена приводит к сбросу резервного домена. Эти сбросы могут быть вызваны внешним сигналом, внутренними событиями и генераторами сброса.

1.3.7.3.1 Сброс по питанию

Сброс по питанию формируется при возникновении одного из следующих событий:

– Подача питания VDD (Power-on сброс);

– Снижение питания VDD ниже допустимого предела (Brown-out сброс), если включена соответствующая функция в регистре RCU_BORCR (см. 1.164).

Сброс по питанию устанавливает для всех регистров значения сброса, кроме тех, которые находятся в резервном домене Backup.

1.3.7.3.2 Системный сброс

Системный сброс устанавливает для всех регистров значения сброса, за исключением регистров резервного домена Backup и флагов сброса в регистре RCU_RSR.

Системный сброс формируется при возникновении одного из следующих событий:

– Низкий уровень на выводе NRST;

– Сброс от WWDT;

– Сброс от IWDT;

– Программный сброс SWRST (см. 1.3.7.3.2.1);

– Сброс по питанию VDD33;

– Сброс при выходе из режимов низкого энергопотребления.

Источник сброса может быть идентифицирован путем проверки флагов сброса в регистре RCU_RSR. Все источники сброса, кроме сброса LPMRST, воздействуют на вывод NRST, который имеет встроенный подтягивающий регистр и управляется изнутри МК Optimal+ через транзистор с открытым стоком. При сбросе от внутренних источников гарантируется минимальная длительность импульса сброса в 20 мкс. В случае внешнего сброса длительность импульса определяется внешней схемой. Упрощенная схема системного сброса показана на рисунке 1.21.

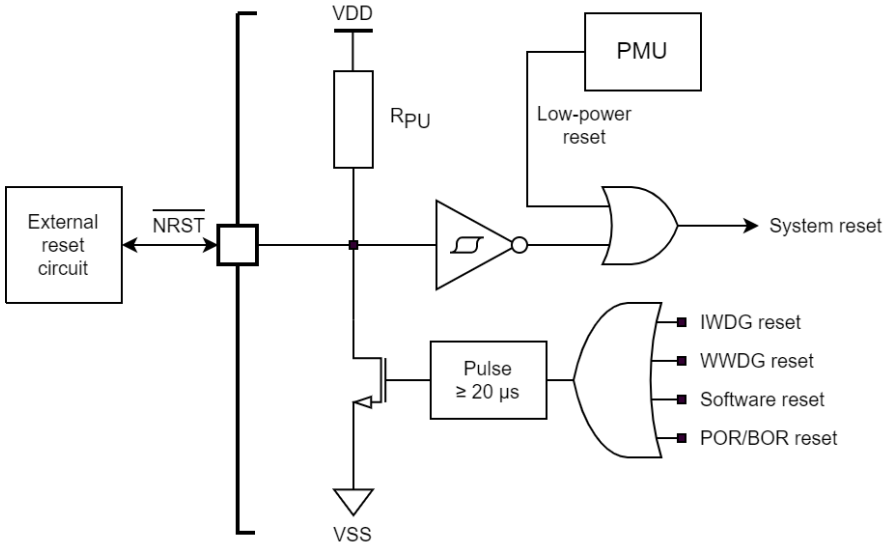


Рисунок 1.21 – Схема системного сброса микроконтроллера

1.3.7.3.2.1 Программный сброс (SWRST)

Программный сброс — это внутренний сброс, генерируемый настройкой бита SYSRESETREQ в регистре AIRCR процессорного ядра. Когда бит SYSRESETREQ установлен в 1, генерируется программный сброс.

1.3.7.3.2.2 Сброс при выходе из режимов низкого энергопотребления (LPMRST)

Сброс LPMRST формируется после выхода микроконтроллера из режима низкого энергопотребления SHUTDOWN, после запуска и стабилизации внутреннего регулятора напряжения HLDO.

1.3.7.3.3 Сброс резервного домена Backup

Сброс резервного домена генерируется при возникновении одного из следующих событий:

– Программный сброс, инициируемый установкой бита BDRST в регистре управления резервным доменом RCU_BDCR;

– Подача питания на вход VBAT (схема BPOR).

1.3.7.4 Блок управления синхросигналами (CCTL)

1.3.7.4.1 Структура синхросигналов

Блок управления синхросигналами Clock Control unit (CCTL) содержит следующие источники тактовых сигналов:

– LSE — 32,768 кГц низкочастотный внешний генератор/резонатор;

– HSE — 4-48 МГц высокочастотный внешний генератор/резонатор;

– LSI — 32 кГц / 250 Гц низкочастотный внутренний RC-генератор;

– HSI0, HSI1 — два унифицированных 4-64 МГц высокочастотных внутренних RC-генератора;

– PLL0, PLL1 — два унифицированных блока ФАПЧ для получения высоких значений частот.

Каждый источник синхросигналов можно включать или выключать независимо, когда он не используется, для оптимизации энергопотребления.

Четыре типа источников могут быть использованы в качестве источника синхросигналов для системной частоты SYSCLK:

– HSI0;

– LSI;

– HSE;

– PLL.

Для настройки частоты AHB и доменов APB можно использовать несколько делителей. Максимальная частота доменов AHB и APB составляет 150 МГц.

Все периферийные тактовые сигналы являются производными от тактовых сигналов периферийной шины, к которой они подключены (HCLK, PCLKx), за исключением ряда периферийных устройств, которые могут получать независимые тактовые сигналы напрямую от

источников. Выбор сигналов синхронизации для этих устройств доступен через регистры управления.

Схема сигналов синхронизации приведена на рисунке 1.22.

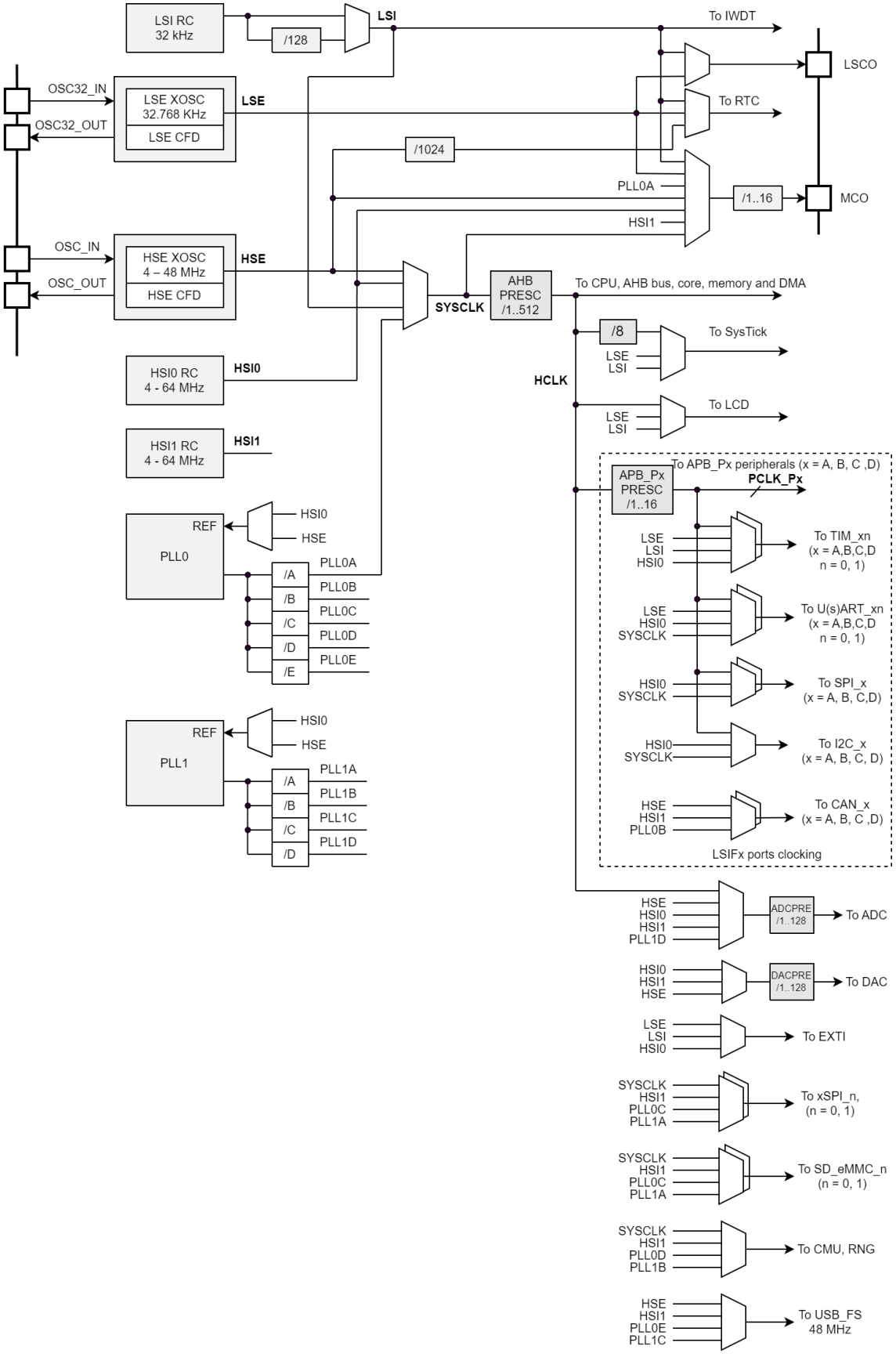


Рисунок 1.22 – Структура сигналов синхронизации

1.3.7.4.2 Внешний синхросигнал HSE

Высокочастотный внешний тактовый сигнал (HSE) может быть получен из двух возможных источников:

– Внешний резонатор;

– Внешний генератор.

Внешний резонатор

Резонатор и нагрузочные конденсаторы должны располагаться как можно ближе к выводам, чтобы свести к минимуму искажения на выходе и время стабилизации при запуске.

Внешний резонатор может быть отключен путем установки соответствующего бита в регистре управления.

Внешний генератор

В данном режиме внешний источник синхросигнала может быть подключен ко входу OSC_IN, в то время как выход OSC_OUT может использоваться в качестве GPIO.

Управление включением и выключением HSE осуществляется битом HSEON в регистре RCU_CSCR. Перед включением HSE необходимо задать режим внешнего резонатора или генератора битом HSEBYP, а также установить поле HSERANGE в соответствии с номинальной частотой подключенного внешнего генератора или резонатора. Статус стабильности генератора отображается битом HSERDY в регистре RCU_CSSR, при HSERDY = 0 выходной синхросигнал HSE маскируется.

1.3.7.4.3 Внутренние RC-генераторы HSI0 и HSI1

Источники HSI0 и HSI1 являются унифицированными RC-генераторами с базовой частотой 4 МГц, с возможностью умножения этой частоты на целый коэффициент от 1 до 16, выбор частоты осуществляется полями HSI0FSEL и HSI1FSEL регистра RCU_ICSCR0. Перестройка частоты HSI0, HSI1 происходит за типовое время 7 мкс, в течение которого выходной синхросигнал этих источников маскируется.

Генераторы HSI0, HSI1 позволяют получить источник синхросигнала без использования внешних компонентов на плате, причем время запуска этих источников меньше, чем у HSE. Однако даже после калибровки частота HSI0, HSI1 будет менее точной, чем у HSE.

Генератор HSI0 является стартовым источником системного синхросигнала после системного сброса, с начальной частотой 16 МГц. Также HSI0 также является источником системного синхросигнала после выхода из режимов низкого потребления

(DEEPSLEEPx/SHUTDOWN), и резервным источником тактового сигнала в случае отказа генератора HSE.

Генератор HSI1 может использоваться для тактирования некоторых периферийных блоков, а также интерфейса USB-FS при настройке на частоту 48 МГц.

Калибровка

Частота генератора может варьироваться от одного чипа к другому из-за различий в технологическом процессе, поэтому каждый чип калибруется на заводе с точностью 1,5% при температуре окружающей среды T_A = 25°C. После сброса заводское значение калибровки загружается в регистр калибровки внутренних источников синхронизации.

Тем не менее в процессе работы частота внутреннего генератора зависит от колебаний температуры, напряжения питания и других факторов. Для компенсации влияния этих факторов, частота HSI может быть скорректирована с помощью установки соответствующего значения TRIM в регистре калибровки внутренних источников синхронизации. Процедура контроля и подстройки частоты генератора HSI с помощью блока CFAU описана в пункте 1.3.7.5.

1.3.7.4.4 Внутренние блоки PLL

Для получения высоких значений тактовых частот микроконтроллер имеет в своем составе два одинаковых блока ФАПЧ - PLL0 и PLL1. В качестве источников опорной частоты для PLL выступают генераторы HSI0 и HSE. Упрощенная структура блока PLL показана на рисунке 1.23

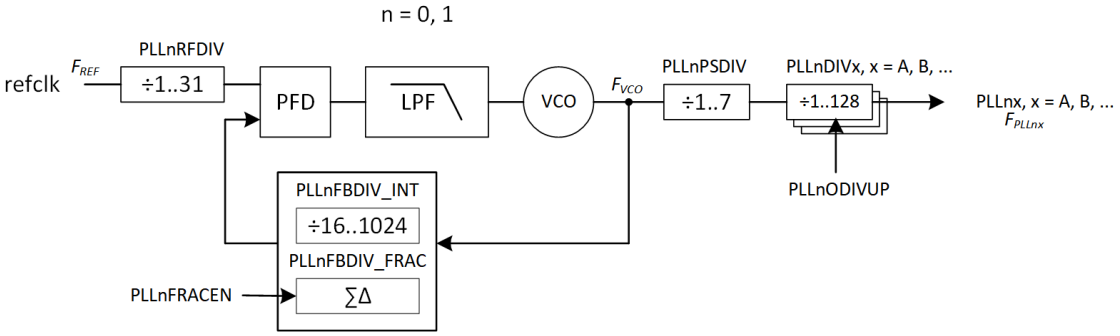


Рисунок 1.23 – Структура PLL

Настройка выходной частоты PLL осуществляется набором делителей:

– PLLnRFDIV - делитель опорной частоты, значение от 1 до 31;

– PLLnPSDIV - предвыходной делитель частоты VCO, значение от 1 до 7;

– PLLnFBDIV - делитель обратной связи, состоящий из целочисленной (PLLnFBDIV_INT) и дробной составляющей (PLLnFBDIV_FRAC):

– дробная составляющая имеет точность $\frac{1}{2^{22}}$ и может быть опционально включена или выключена с помощью битов PLLnFRACEN в регистрах RCU_PLLnCR0;

– целочисленная составляющая может принимать значения от 16 до 1024 при PLLnFRACEN = 0 и значения от 20 до 240 при PLLnFRACEN = 1;

– PLLnDIVx, - выходные делители, где x - индекс выходного синхросигнала A, B, C и т. д. Значения от 1 до 128.

Значения выходных частот вычисляются следующим образом:

$$F_{VCO} = \frac{F_{REF}(PLLnFBDIV_INT + \frac{PLLnFBDIV_FRAC \cdot PLLnFRACEN}{2^{22}})}{PLLnRFDIV},$$

$$F_{PLLnx} = \frac{F_{VCO}}{PLLnPSDIV \cdot (PLLnDIVx+1)}, \text{ n = 0, 1, x = A, B, … .}$$

Конфигурацию параметров PLL необходимо производить, когда PLL находится в выключенном состоянии (PLLnON = 0), за исключением настроек выходных делителей PLLnDIVx, которые можно изменять в процессе работы и активировать с помощью битов PLLnODIVUP в регистрах RCU_PLLnCR0. Перед включением PLL необходимо выбрать опорный синхросигнал с помощью поля PLLnSRC в регистре RCU_PLLnCR, и убедиться, что выбранный источник опорного синхросигнала находится в стабильном состоянии.

1.3.7.4.5 Внешний синхросигнал LSE

Источник LSE, также как и HSE, может быть получен, используя внешний резонатор или внешний генератор, настроенный на частоту 32,768 кГц. Источник LSE является высокоточным источником синхросигнала для блока часов реального времени (RTC) и других периферийных устройств.

Внешний резонатор

Резонатор и нагрузочные конденсаторы должны располагаться как можно ближе к выводам, чтобы свести к минимуму искажения на выходе и время стабилизации при запуске.

Внешний резонатор может быть отключен путем установки соответствующего бита в регистре управления.

Внешний генератор

В данном режиме внешний источник синхросигнала может быть подключен ко входу OSC32_IN, выход OSC32_OUT при этом не используется.

Управление включением и выключением LSE осуществляется битом LSEON в регистре RCU_BDCR. Перед включением LSE необходимо задать режим внешнего резонатора или

генератора битом LSEBYP. Статус стабильности генератора отображается битом LSERDY в регистре RCU_BDCR, при LSERDY = 0 выходной синхросигнал LSE маскируется.

1.3.7.4.6 Внутренний RC-генератор LSI

LSI используется в качестве источника низкоскоростного синхросигнала. Генератор LSI может продолжать работу в режимах пониженного энергопотребления DEEPSLEEPx/SHUTDOWN, для тактирования IWDG и RTC. Для уменьшения энергопотребления частоту данного синхросигнала можно понизить с 32 кГц до 250 Гц, используя делитель на 128.

1.3.7.4.7 Системный синхросигнал SYSCLK

Основным внутренним синхросигналом микроконтроллера является синхросигнал SYSCLK, который тактирует процессорное ядро, коммутационную среду AHB, периферийные шины APBx и т.д. В качестве источника SYSCLK могут использоваться следующие синхросигналы:

– HSI0 (внутренний высокочастотный RC-генератор);

– LSI (внутренний низкочастотный RC-генератор);

– HSE (внешний высокочастотный генератор/резонатор);

– PLL0 (накристальная схема ФАПЧ).

По умолчанию после системного сброса в качестве источника SYSCLK выбран генератор HSI0, настроенный на частоту 16 МГц. Программный выбор источника SYSCLK осуществляется полем SW[1:0] регистра RCU_CCFGR0, статус переключения отображается в поле SWS[1:0] в этом же регистре. Так как переключение частоты между источниками имеет некоторую задержку, то после изменения поля SW[1:0] ПО должно удостовериться в статусе переключения путем чтения поля SWS[1:0].

Система переключения источников SYSCLK выполнена таким образом, чтобы исключить ситуацию зависания системы вследствие пропадания частоты SYSCLK, что означает следующее:

– если в поле SW[1:0] выбран источник, который не находится в рабочем режиме, то переключение будет аппаратно заблокировано до тех пор, пока выбранный источник не выйдет в рабочий режим;

– невозможно выключить источник, который в настоящее время используется как системная частота SYSCLK (непосредственно или в качестве опорного синхросигнала PLL0).

1.3.7.4.8 Синхросигнал системного таймера SysTick

Ядро Cortex-M33 имеет в своем составе системный таймер SysTick, который может тактироваться либо непосредственно от частоты ядра, либо от внешнего опорного синхросигнала. Выбор между внутренней и внешней частотой определяется битом CLKSOURCE регистра ядра SYST_CSR. В качестве внешнего опорного синхросигнала (SYST_CSR.CLKSOURCE = 0) возможен выбор из следующих источников:

– частота HCLK, деленная на 8;

– генератор LSI;

– генератор LSE.

Выбор источника внешнего синхросигнала SysTick осуществляется полем SYSTICKSRC регистра RCU_CCFGR0. В случае, если SysTick тактируется от источников LSI или LSE, их частота не должна превышать четверть частоты HCLK, то есть должно выполняться следующее соотношение:

$$F_{LSI,LSE} \leq \frac{F_{HCLK}}{4} \quad (1)$$

где $F_{LSI,LSE}$ — тактовая частота источников LSE и LSI,

F_{HCLK} — тактовая частота HCLK.

Упрощенная схема тактирования SysTick показана на рисунке 1.24.

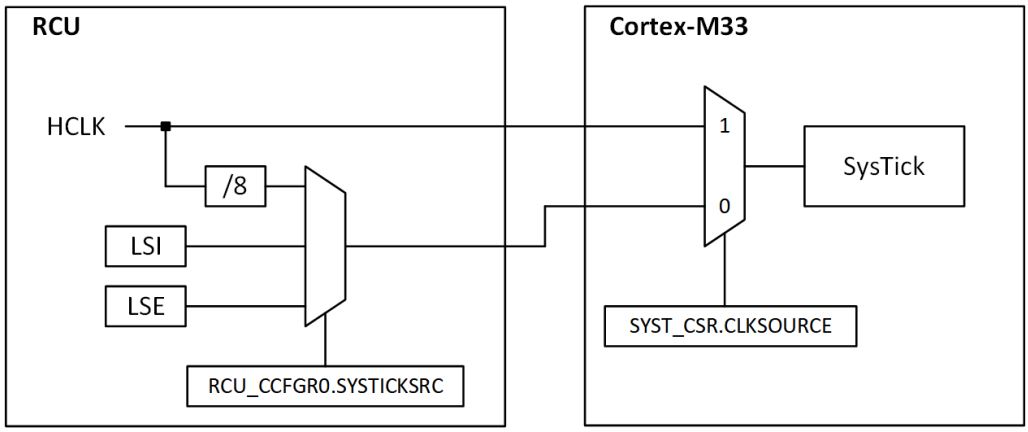


Рисунок 1.24 – Схема тактирования SysTick

1.3.7.4.9 Детектор отказов генератора HSE

Схема детектора отказа HSE (далее HSECFD - HSE clock failure detection) позволяет обнаружить пропадание импульсов от генератора HSE, если он по какой-то причине вышел из строя. Функция HSECFD может быть программно включена битом HSECFDEN регистра RCU_CSCR, при этом детектор отказов активируется только после выхода генератора HSE в

рабочий режим и автоматически отключается при программном выключении HSE. При обнаружении отказа HSE происходит следующее:

– генератор HSE принудительно выключается;

– если HSE на момент отказа использовался как источник системной частоты SYSCLK (напрямую или через PLL), то происходит автоматическое переключение SYSCLK на резервный источник HSI0;

– если HSE на момент отказа являлся опорным синхросигналом PLL и схема PLL использовалась в качестве источника SYSCLK, то PLL принудительно выключается;

– устанавливается бит IRQS_HSEFAIL в регистре RCU_IRQSR и формируется немаскируемое прерывание в процессорное ядро.

Следует отметить, что схема HSECFD обнаруживает только полное отключение синхросигнала HSE и не отслеживает глитчи и отклонение частоты.

1.3.7.4.10 Детектор отказов генератора LSE

Схема детектора отказа LSE (далее LSECFD - LSE clock failure detection) позволяет обнаружить пропадание импульсов от генератора LSE, если он по какой-то причине вышел из строя. Функция LSECFD может быть программно включена битом LSECFDEN регистра RCU_BDCR, при этом детектор отказов активируется только после выхода генератора LSE в рабочий режим и автоматически отключается при программном выключении LSE. При обнаружении отказа LSE формируется событие прерывания, подключенное к входу немаскируемого прерывания процессорного ядра, с возможностью пробуждения из режимов низкого энергопотребления. Статус отказа генератора LSE также отражается в бите LSEFAIL регистра RCU_BDCR.

При обнаружении отказа генератора LSE программное обеспечение обязано выполнить следующие действия:

– сбросить бит LSECFDEN в регистре RCU_BDCR;

– выключить отказавший генератор LSE (бит LSEON = 0 в регистре RCU_BDCR);

– изменить источник тактирования RTC на LSI или HSE либо предпринять другие меры для обеспечения безопасности приложения.

Следует отметить, что схема LSECFD обнаруживает только полное отключение синхросигнала LSE и не отслеживает глитчи и отклонение частоты.

1.3.7.4.11 Внешние выходы синхросигналов MCO и LSCO

Для отладочных целей и тактирования внешних устройств предусмотрена возможность вывода внутренних синхросигналов микроконтроллера на внешние выводы.

Внешний вывод MCO

Основной выход синхросигнала микроконтроллера MCO (Microcontroller Clock Output) позволяет выводить следующие синхросигналы:

– SYSCLK;

– HSI0;

– HSI1;

– HSE;

– PLL0A;

– LSI;

– LSE.

Частота выводимого синхросигнала может быть поделена на целый коэффициент от 1 до 16. Выбор синхросигнала для вывода и коэффициент деления задаются битами MCOSEL[3:0] и MCOPRE[3:0] в регистре конфигурации синхросигналов 0 RCU_CCFGR0.

Внешний вывод LSCO

Выход синхросигнала низкой частоты LSCO (Low Speed Clock Output) позволяет вывести один из низкоскоростных тактовых сигналов наружу:

– LSI;

– LSE.

Этот выход остается доступным во всех режимах низкого энергопотребления. Выбор синхросигнала LSCO и его включение управляются битами LSCOSEL и LSCOEN регистра конфигурации резервного домена RCU_BDCR.

1.3.7.5 Блок контроля частоты синхросигналов (CFAU)

1.3.7.5.1 Общее описание

Блок контроля частоты синхросигналов (далее CFAU - Clock Frequency Analysis Unit) подсчитывает число импульсов тактового сигнала, подлежащего измерению, в течение периода опорного синхросигнала, выбранного в качестве источника эталонной частоты. В конце каждого периода измерения происходит сравнение количества подсчитанных импульсов с заданным диапазоном, и в случае выхода за пределы диапазона формируется событие прерывания.

Характеристики блока CFAU приведены в таблице 1.140. Структурная схема блока CFAU показана на рисунке 1.25.

Таблица 1.140 – Характеристики блока CFAU

Параметр	**Значение**
Измеряемые синхросигналы	- LSI; - HSI0; - HSI1; - SYSCLK.
Опорные синхросигналы	- LSE; - HSE.
Делитель измеряемого синхросигнала	Программируемый, 1 - 32.
Делитель опорного синхросигнала	Программируемый, 1 - 16384.
Формируемые события прерывания	- Завершение измерения; - Ошибка частоты; - Переполнение счетчика.

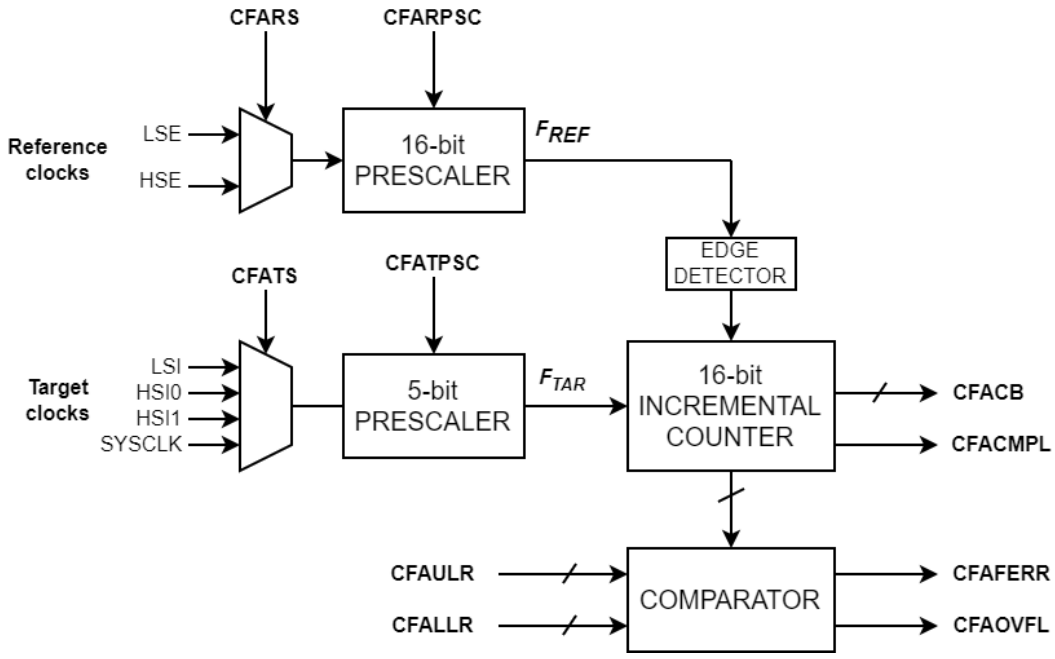


Рисунок 1.25 – Структура блока контроля частоты CFAU

1.3.7.5.2 Принцип работы

Блок CFAU представляет собой инкрементный счетчик, тактируемый от измеряемого синхросигнала. В конце каждого периода измерения по положительному фронту опорного синхросигнала происходит следующее:

– формируется событие прерывания о завершении периода измерения IRQS_CFACMPL;

– текущее значение счетчика копируется в буферный регистр RCU_CFASR[CFACBR], а сам счетчик сбрасывается;

– текущее значение счетчика сравнивается с верхним и нижним пороговыми значениями, задаваемых в регистрах RCU_CFAULR и RCU_CFALLR соответственно, и при не выполнении условия RCU_CFALLR < RCU_CFASR[CFACBR] < RCU_CFAULR формируется событие прерывания ошибки частоты IRQ_CFAFERR.

В случае, если счетчик достиг значения 0xFFFF до конца текущего периода измерения, формируется событие прерывания о переполнении IRQ_CFAOVFL. Пример работы блока CFAU показан на рисунке 1.26:

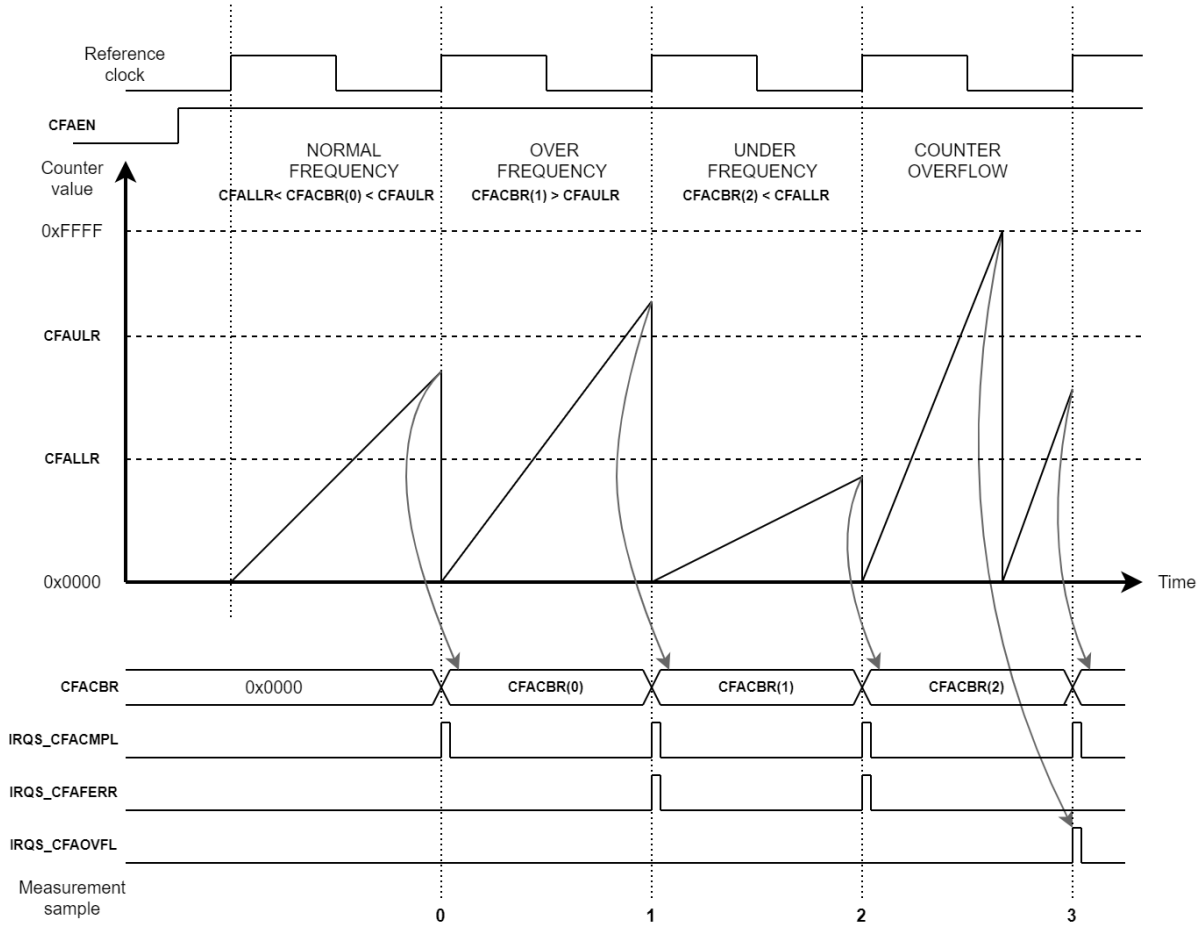


Рисунок 1.26 – Пример работы блока CFAU

1.3.7.5.3 Порядок настройки

Конфигурация CFAU осуществляется регистрами RCU_CFACR, RCU_CFAULR и RCU_CFALLR Конфигурация выполняется в следующем порядке:

1) Убедится, что источники опорного и измеряемого синхросигналов включены и готовы;

2) Убедится, что CFAU находится в выключенном состоянии (RCU_CFACR[CFAEN] = 0);

3) Выбрать опорный и измеряемый синхросигналы с помощью полей RCU_CFACR[CFARS] и RCU_CFACR[CFATS] соответственно;

4) Настроить делители опорного и измеряемого синхросигнала с помощью полей RCU_CFACR[CFARPSC] и RCU_CFACR[CFATPSC] соответственно;

5) Задать допустимый диапазон частоты измеряемого синхросигнала с помощью регистров RCU_CFAULR и RCU_CFALLR;

6) Разрешить работу CFAU, установив бит RCU_CFACR[CFAEN] = 1.

После включения CFAU результаты измерений будут фиксироваться в регистре RCU_CFASR, а также будут формироваться соответствующие события прерывания в регистре RCU_IRQS.

Для корректной работы CFAU должны выполняться следующие условия:

– Верхний порог задаваемого диапазона должен быть большие нижнего порога: CFAUL > CFALL;

– Частота опорного синхросигнала должна быть ниже частоты измеряемого синхросигнала:

$F_{REF} < F_{TAR}$, частоты указаны после делителей.

1.3.7.6 Прерывания RCU

В таблице 1.141 приведен список событий прерывания, формируемых блоком RCU.

Таблица 1.141 – Прерывания блока RCU

Вектор NVIC	Статусный бит в регистре RCU_IRQSR	Описание	Бит разрешения в регистре RCU_IRQER	Способ сброса
RCU (№28)	IRQS_LSIRDY	Переход LSI в рабочий режим	IRQE_LSIRDY	Запись 1 в бит IRQC_LSIRDY
	IRQS_LSERDY	Переход LSE в рабочий режим	IRQE_LSERDY	Запись 1 в бит IRQC_LSERDY
	IRQS_HSI0RDY	Переход HSI0 в рабочий режим	IRQE_HSI0RDY	Запись 1 в бит IRQC_HSI0RDY

Вектор NVIC	Статусный бит в регистре RCU_IRQSR	Описание	Бит разрешения в регистре RCU_IRQER	Способ сброса
	IRQS_HSI1RDY	Переход HSI1 в рабочий режим	IRQE_HSI1RDY	Запись 1 в бит IRQC_HSI1RDY
	IRQS_HSERDY	Переход HSE в рабочий режим	IRQE_HSERDY	Запись 1 в бит IRQC_HSERDY
	IRQS_PLL0RDY	Переход PLL0 в рабочий режим	IRQE_PLL0RDY	Запись 1 в бит IRQC_PLL0RDY
	IRQS_PLL1RDY	Переход PLL1 в рабочий режим	IRQE_PLL1RDY	Запись 1 в бит IRQC_PLL1RDY
NMI (№2)	IRQS_HSEFAIL	Отказ HSE	Всегда разрешено	Запись 1 в бит IRQC_HSEFAIL
	IRQS_LSEFAIL	Отказ LSE	Всегда разрешено	Запись 1 в бит IRQC_LSEFAIL
RCU (№28)	IRQS_CFACMPL	Завершение измерения CFAU	IRQE_CFACMPL	Запись 1 в бит IRQC_CFACMPL
	IRQS_CFAFERR	Ошибка частоты CFAU	IRQE_CFAFERR	Запись 1 в бит IRQC_CFAFERR
	IRQS_CFAOVFL	Переполнение счетчика CFAU	IRQE_CFAOVFL	Запись 1 в бит IRQC_CFAOVFL

1.3.7.7 Карта регистров RCU

1.3.7.7.1 Список регистров RCU

В таблице 1.142 представлен список программно-доступных регистров подсистемы RCU, а в таблицах 1.143 - 1.173 - описание их полей.

Таблица 1.142 – Список программно-доступных регистров RCU

Адрес	Название	Описание
0x000	RCU_CSCR	Регистр управления источниками синхросигналов
0x004	RCU_CSSR	Регистр статуса источников синхросигналов
0x008	RCU_ICSCR0	Регистр калибровки внутренних источников синхросигналов 0
0x00C	RCU_ICSCR1	Регистр калибровки внутренних источников синхросигналов 1
0x010	-	*резерв*
0x014	-	*резерв*
0x018	RCU_CCFGR0	Регистр конфигурации синхросигналов 0
0x01C	RCU_CCFGR1	Регистр конфигурации синхросигналов 1
0x020	RCU_CCFGR2	Регистр конфигурации синхросигналов 2
0x024	-	*резерв*
0x028	RCU_PLL0CR0	Регистр настроек PLL0
0x02C	RCU_PLL0CR1	Регистр настроек PLL0
0x030	RCU_PLL0CR2	Регистр настроек PLL0
0x034	RCU_PLL0CR3	Регистр настроек PLL0
0x038	RCU_PLL1CR0	Регистр настроек PLL1
0x03C	RCU_PLL1CR1	Регистр настроек PLL1
0x040	RCU_PLL1CR2	Регистр настроек PLL1
0x044	-	*резерв*

Адрес	Название	Описание
0x048	-	*резерв*
0x050	RCU_PCER0	Регистр разрешения тактирования периферии 0
0x054	RCU_PCER1	Регистр разрешения тактирования периферии 1
0x058 - 0x05C	-	*резерв*
0x060	RCU_PICSR0	Регистр выбора независимых синхросигналов периферии 0
0x064	RCU_PICSR1	Регистр выбора независимых синхросигналов периферии 1
0x068	RCU_PICSR2	Регистр выбора независимых синхросигналов периферии 2
0x06C	-	*резерв*
0x070	-	*резерв*
0x078	RCU_PRR0	Регистр сброса периферии 0
0x07C	RCU_PRR1	Регистр сброса периферии 1
0x080 - 0x098	-	*резерв*
0x09C	RCU_BORCR	Регистр управления сбросом при падении напряжения питания
0x0A0 - 0x0BC	-	*резерв*
0x0C0	RCU_CFACR	Регистр управления блоком контроля частоты
0x0C4	RCU_CFASR	Регистр статуса блока контроля частоты
0x0C8	RCU_CFAULR	Регистр верхнего порога счетчика контроля частоты
0x0CC	RCU_CFALLR	Регистр нижнего порога счетчика контроля частоты
0x0B0 - 0x0DC	-	*резерв*
0x0E0	RCU_IRQER	Регистр маски прерываний
0x0E4	RCU_IRQSR	Регистр статуса прерываний
0x0E8	RCU_IRQCR	Регистр сброса прерываний
0x0EC	-	*резерв*
0x0F0	RCU_BDCR	Регистр конфигурации резервного домена
0x0F4	RCU_RSR	Регистр статуса сбросов

1.3.7.7.2 Регистр управления источниками синхросигналов (RCU_CSCR)

Смещение адреса: 0x000.

Значение по сбросу: 0x0000_0001.

Таблица 1.143 – Описание полей регистра RCU_CSCR

Биты	Название	Доступ	Значение по сбросу	Описание
[31:28]	—	R0	0x0	*резерв*
[27:26]	HSI1_TEST	R/W	0x0	Выбор тестового режима HSI1 (см. спецификацию на SGC21547_16M_HDX01).
[25:24]	HSI0_TEST	R/W	0x0	Выбор тестового режима HSI0 (см. спецификацию на SGC21547_16M_HDX01).

Биты	Название	Доступ	Значение по сбросу	Описание
[23:18]	—	R0	0x0	*резерв*
[17]	PLL1ON	R/W	0x0	Управление генератором PLL1: 0x0 - PLL1 выключен; 0x1 - PLL1 включен.
[16]	PLL0ON	R/W	0x0	Управление генератором PLL0: 0x0 - PLL0 выключен; 0x1 - PLL0 включен. Данный бит не может быть сброшен, если PLL0 используется в качестве источника системного синхросигнала.
[15:9]	—	R0	0x0	*резерв*
[8]	HSECFDEN	R/W	0x0	Управление детектором отказов генератора HSE: 0x0 - детектор отказов HSE выключен; 0x1 - детектор отказов HSE включен. При HSECFDEN = 1 детектор отказов активируется только при условии HSERDY = 1.
[7:6]	HSERANGE	R/W	0x0	Выбор диапазона частоты HSE: 0x0 - 4..8 МГц; 0x1 - 8..16 МГц; 0x2 - 16..32 МГц; 0x3 - 32..48 МГц. Данное поле допускается изменять только когда HSE находится в выключенном состоянии (HSEON = 0).
[5]	HSEBYP	R/W	0x0	Управление обходом генератора HSE: 0x0 - HSE используется; 0x1 - обход HSE внешним синхросигналом со входа OSC_IN. Данный бит допускается изменять только когда HSE находится в выключенном состоянии (HSEON = 0).
[4]	HSEON	R/W	0x0	Управление генератором HSE: 0x0 - HSE выключен; 0x1 - HSE включен. Данный бит не может быть сброшен, если HSE используются в качестве системного синхросигнала (напрямую или через PLL).
[3:2]	—	R0	0x0	*резерв*
[1]	HSI1ON	R/W	0x0	Управление генератором HSI1: 0x0 - HSI1 выключен; 0x1 - HSI1 включен.
[0]	HSI0ON	R/W	0x0	Управление генератором HSI0: 0x0 - HSI0 выключен; 0x1 - HSI0 включен. Данный бит не может быть сброшен, если HSI0 используются в качестве системного синхросигнала (напрямую или через PLL).

1.3.7.7.3 Регистр статуса источников синхросигналов (RCU_CSSR)

Смещение адреса: 0x004.

Значение по сбросу: 0x0000_0001.

Регистр RCU_CSSR содержит статусную информацию об источниках синхросигналов. При выходе интересующего источника в рабочий режим соответствующий бит регистра аппаратно устанавливается в 1, что свидетельствует о стабильности синхросигнала источника и возможности его использования для тактирования устройств.

Таблица 1.144 – Описание полей регистра RCU_CSSR

Биты	Название	Доступ	Значение по сбросу	Описание
[31:10]	—	R0	0x0	*резерв*
[9]	PLL1RDY	R	0x0	Флаг готовности генератора PLL1: 0x0 - синхросигнал PLL1 нестабилен; 0x1 - синхросигнал PLL1 стабилен.
[8]	PLL0RDY	R	0x0	Флаг готовности генератора PLL0: 0x0 - синхросигнал PLL0 нестабилен; 0x1 - синхросигнал PLL0 стабилен.
[7:5]	—	R0	0x0	*резерв*
[4]	HSERDY	R	0x0	Флаг готовности генератора HSE: 0x0 - синхросигнал HSE нестабилен; 0x1 - синхросигнал HSE стабилен.
[3:2]	—	R0	0x0	*резерв*
[1]	HSI1RDY	R	0x0	Флаг готовности генератора HSI1: 0x0 - синхросигнал HSI1 нестабилен; 0x1 - синхросигнал HSI1 стабилен.
[0]	HSI0RDY	R	0x1	Флаг готовности генератора HSI0: 0x0 - синхросигнал HSI0 нестабилен; 0x1 - синхросигнал HSI0 стабилен.

1.3.7.7.4 Регистр калибровки внутренних источников синхросигналов (RCU_ICSCR0)

Смещение адреса: 0x008.

Значение по сбросу: 0x0713_0713.

Таблица 1.145 – Описание полей регистра RCU_ICSCR0

Биты	Название	Доступ	Значение по сбросу	Описание
[31:29]	HSI1TT	R/W	0x0	Коэффициент термокомпенсации HSI1
[28:20]	HSI1FT	R/W	0x71	Коэффициент подстройки частоты HSI1
[19:16]	HSI1FSEL	R/W	0x3	Выбор частоты генератора HSI1: F_{HSI1} = 4 * (HSI1FSEL + 1) МГц
[15:13]	HSI0TT	R/W	0x0	Коэффициент термокомпенсации HSI0
[12:4]	HSI0FT	R/W	0x71	Коэффициент подстройки частоты HSI0
[3:0]	HSI0FSEL	R/W	0x3	Выбор частоты генератора HSI0: F_{HSI0} = 4 * (HSI0FSEL + 1) МГц

1.3.7.7.5 Регистр калибровки внутренних источников синхросигналов (RCU_ICSCR1)

Смещение адреса: 0x00C.

Значение по сбросу: 0x0000_0000.

Таблица 1.146 – Описание полей регистра RCU_ICSCR1

Биты	Название	Доступ	Значение по сбросу	Описание
[31:4]	-	R0	0x0	*резерв*
[3:2]	HSI1IT	R/W	0x0	Коэффициент подстройки источника опорного тока HSI1. Данное поле предназначено только для отладки и не должно изменятся в процессе работы.
[1:0]	HSI0IT	R/W	0x0	Коэффициент подстройки источника опорного тока HSI0. Данное поле предназначено только для отладки и не должно изменятся в процессе работы.

1.3.7.7.6 Регистр конфигурации синхросигналов 0 (RCU_CCFGR0)

Смещение адреса: 0x018.

Значение по сбросу: 0x0000_0000.

Таблица 1.147 – Описание полей регистра RCU_CCFGR0

Биты	Название	Доступ	Значение по сбросу	Описание
[31:24]	—	R0	0x0	*резерв*
[23:20]	MCOPRE	R/W	0x0	Коэффициент деления частоты MCO: 0b0000 - MCO делится на 1; 0b0001 - MCO делится на 2; … 0b1111 - MCO делится на 16.
[19]	—	R0	0x0	*резерв*
[18:16]	MCOSEL	R/W	0x0	Выбор синхросигнала для внешнего вывода MCO: 0b000: синхросигнал MCO выключен; 0b001: синхросигнал MCO = SYSCLK; 0b010: синхросигнал MCO = HSI0; 0b011: синхросигнал MCO = HSI1; 0b100: синхросигнал MCO = HSE; 0b101: синхросигнал MCO = PLL0A; 0b110: синхросигнал MCO = LSI; 0b111: синхросигнал MCO = LSE.
[15:8]	—	R0	0x0	*резерв*

Биты	Название	Доступ	Значение по сбросу	Описание
[7:6]	SYSTICKSRC	R/W	0x0	Выбор внешнего синхросигнала для системного таймера SysTick: 0b00 - частота HCLK/8; 0b01 - частота LSI; 0b10 - частота LSE; 0b11 - *резерв*.
[5:4]	—	R0	0x0	*резерв*
[3:2]	SWS	R	0x0	Статус переключателя источника системного синхросигнала: 0b00 - источником SYSCLK является HSI0; 0b01 - источником SYSCLK является LSI; 0b10 - источником SYSCLK является HSE; 0b11 - источником SYSCLK является PLL0A.
[1:0]	SW	R/W	0x0	Выбор источника системного синхросигнала: 0b00 - источником SYSCLK выбран HSI0; 0b01 - источником SYSCLK выбран LSI; 0b10 - источником SYSCLK выбран HSE; 0b11 - источником SYSCLK выбран PLL0A. Данное поле аппаратно переопределяется на источник HSI0 в случае отказа генератора HSE, если он на момент отказа являлся источником SYSCLK.

1.3.7.7.7 Регистр конфигурации синхросигналов 1 (RCU_CCFGR1)

Смещение адреса: 0x01C.

Значение по сбросу: 0x0000_0000.

Таблица 1.148 – Описание полей регистра RCU_CCFGR1

Биты	Название	Доступ	Значение по сбросу	Описание
[31]	PD_DIS	R/W	0x0	Отключение синхросигнала PCLK_PD 0x0: синхросигнал PCLK_PD включен; 0x1: синхросигнал PCLK_PD выключен.
[30]	PC_DIS	R/W	0x0	Отключение синхросигнала PCLK_PC 0x0: синхросигнал PCLK_PC включен; 0x1: синхросигнал PCLK_PC выключен.
[29]	PB_DIS	R/W	0x0	Отключение синхросигнала PCLK_PB 0x0: синхросигнал PCLK_PB включен; 0x1: синхросигнал PCLK_PB выключен.
[28]	PA_DIS	R/W	0x0	Отключение синхросигнала PCLK_PA 0x0: синхросигнал PCLK_PA включен; 0x1: синхросигнал PCLK_PA выключен.
[27:24]	PPRE_PD	R/W	0x0	Коэффициент деления частоты PCLK_PD: PCLK_PD = HCLK / (PPRE_PD + 1).
[23:20]	PPRE_PC	R/W	0x0	Коэффициент деления частоты PCLK_PC: PCLK_PC = HCLK / (PPRE_PC + 1).

Биты	Название	Доступ	Значение по сбросу	Описание
[19:16]	PPRE_PB	R/W	0x0	Коэффициент деления частоты PCLK_PB: PCLK_PB = HCLK / (PPRE_PB + 1).
[15:12]	PPRE_PA	R/W	0x0	Коэффициент деления частоты PCLK_PA: PCLK_PA = HCLK / (PPRE_PA + 1).
[11:9]	—	R0	0x0	*резерв*
[8:0]	HPRE	R/W	0x0	Коэффициент деления частоты HCLK: HCLK = SYSCLK / (HPRE + 1).

1.3.7.7.8 Регистр конфигурации синхросигналов 1 (RCU_CCFGR2)

Смещение адреса: 0x020.

Значение по сбросу: 0x0000_0901.

Таблица 1.149 – Описание полей регистра RCU_CCFGR2

Биты	Название	Доступ	Значение по сбросу	Описание
[31:15]	—	R0	0x0	*резерв*
[14:8]	DACPRE	R/W	0x9	Коэффициент деления частоты DAC: CLK_DAC = CLK(DAC_SEL) / (DACPRE + 1).
[7]	—	R0	0x0	*резерв*
[6:0]	ADCPRE	R/W	0x1	Коэффициент деления частоты ADC: CLK_ADC = CLK(ADC_SEL) / (ADCPRE + 1).

1.3.7.7.9 Регистр настроек PLL0 0 (RCU_PLL0CR0)

Смещение адреса: 0x028.

Значение по сбросу: 0x0000_0148.

Таблица 1.150 – Описание полей регистра RCU_PLL0CR0

Биты	Название	Доступ	Значение по сбросу	Описание
[31:13]	—	R0	0x0	*резерв*
[12:8]	PLL0RFDIV	R/W	0x1	Значение делителя опорного синхросигнала PLL0, 1..31
[7]	—	R0	0x0	*резерв*
[6:4]	PLL0PSDIV	R/W	0x4	Значение выходного делителя PLL0, 1..7
[3]	PLL0ODIVUP	R/W1	0x1	Применение настроек делителей PLL0A..PLL0E. После изменения полей PLL0A..PLL0E необходимо записать 1 в данный бит, чтобы изменения вступили в силу. Бит сбрасывается автоматически после изменения режима делителей. Данный бит принудительно устанавливается в 1 когда PLL0 находится в выключенном состоянии (PLL0ON = 0).

Биты	Название	Доступ	Значение по сбросу	Описание
[2]	PLL0FRACEN	R/W	0x0	Включение дробного множителя PLL0: 0x0 - дробный множитель выключен; 0x1 - дробный множитель включен.
[1:0]	PLL0SRC	R/W	0x0	Выбор опорного синхросигнала PLL0: 0x0 - отключение опорного синхросигнала PLL0; 0x1 - HSI0; 0x2 - HSE; 0x3 - *резерв*. Данное поле допускается изменять только когда PLL0 находится в выключенном состоянии (PLL0ON = 0).

1.3.7.7.10 Регистр настроек PLL0 1 (RCU_PLL0CR1)

Смещение адреса: 0x02C.

Значение по сбросу: 0x1000_0000.

Таблица 1.151 – Описание полей регистра RCU_PLL0CR1

Биты	Название	Доступ	Значение по сбросу	Описание
[31:22]	PLL0FBDIV_INT	R/W	0x40	Целочисленная часть делителя обратной связи PLL0
[21:0]	PLL0FBDIV_FRAC	R/W	0x0	Дробная часть делителя обратной связи PLL0

1.3.7.7.11 Регистр настроек PLL0 2 (RCU_PLL0CR2)

Смещение адреса: 0x030.

Значение по сбросу: 0x8080_8080.

Таблица 1.152 – Описание полей регистра RCU_PLL0CR2

Биты	Название	Доступ	Значение по сбросу	Описание
[31:24]	PLL0D	R/W	0x80	Режим делителя PLL0D: 0b00000000 - деление на 1; 0b00000001 - деление на 2; … 0b01111111 - деление на 128; 0b1xxxxxxx - выключение делителя.
[23:16]	PLL0C	R/W	0x80	Режим делителя PLL0C: 0b00000000 - деление на 1; 0b00000001 - деление на 2; … 0b01111111 - деление на 128; 0b1xxxxxxx - выключение делителя.
[15:8]	PLL0B	R/W	0x80	Режим делителя PLL0B: 0b00000000 - деление на 1; 0b00000001 - деление на 2; … 0b01111111 - деление на 128; 0b1xxxxxxx - выключение делителя.

Биты	Название	Доступ	Значение по сбросу	Описание
[7:0]	PLL0A	R/W	0x80	Режим делителя PLL0A: 0b00000000 - деление на 1; 0b00000001 - деление на 2; … 0b01111111 - деление на 128; 0b1xxxxxxx - выключение делителя. Старший бит этого поля принудительно удерживается в 0 когда PLL0 используется как источник SYSCLK.

1.3.7.7.12 Регистр настроек PLL0 3 (RCU_PLL0CR3)

Смещение адреса: 0x034.

Значение по сбросу: 0x0000_0080.

Таблица 1.153 – Описание полей регистра RCU_PLL0CR3

Биты	Название	Доступ	Значение по сбросу	Описание
[31:8]	-	R0	0x0	*резерв*
[7:0]	PLL0E	R/W	0x80	Режим делителя PLL0E: 0b00000000 - деление на 1; 0b00000001 - деление на 2; … 0b01111111 - деление на 128; 0b1xxxxxxx - выключение делителя.

1.3.7.7.13 Регистр настроек PLL1 0 (RCU_PLL1CR0)

Смещение адреса: 0x038.

Значение по сбросу: 0x0000_0148.

Таблица 1.154 – Описание полей регистра RCU_PLL1CR0

Биты	Название	Доступ	Значение по сбросу	Описание
[31:13]	—	R0	0x0	*резерв*
[12:8]	PLL1RFDIV	R/W	0x1	Значение делителя опорного синхросигнала PLL1, 1..31
[7]	—	R0	0x0	*резерв*
[6:4]	PLL1PSDIV	R/W	0x4	Значение выходного делителя PLL1, 1..7
[3]	PLL1ODIVUP	R/W1	0x1	Применение настроек делителей PLL1A..PLL1D. После изменения полей PLL1A..PLL1D необходимо записать 1 в данный бит, чтобы изменения вступили в силу. Бит сбрасывается автоматически после изменения режима делителей. Данный бит принудительно устанавливается в 1 когда PLL1 находится в выключенном состоянии (PLL1ON = 0).

Биты	Название	Доступ	Значение по сбросу	Описание
[2]	PLL1FRACEN	R/W	0x0	Включение дробного множителя PLL1: 0x0 - дробный множитель выключен; 0x1 - дробный множитель включен.
[1:0]	PLL1SRC	R/W	0x0	Выбор опорного синхросигнала PLL1: 0x0 - отключение опорного синхросигнала PLL1; 0x1 - HSI0; 0x2 - HSE; 0x3 - *резерв*. Данное поле допускается изменять только когда PLL1 находится в выключенном состоянии (PLL1ON = 0).

1.3.7.7.14 Регистр настроек PLL1 1 (RCU_PLL1CR1)

Смещение адреса: 0x03C.

Значение по сбросу: 0x1000_0000.

Таблица 1.155 – Описание полей регистра RCU_PLL1CR1

Биты	Название	Доступ	Значение по сбросу	Описание
[31:22]	PLL1FBDIV_INT	R/W	0x40	Целочисленная часть делителя обратной связи PLL1
[21:0]	PLL1FBDIV_FRAC	R/W	0x0	Дробная часть делителя обратной связи PLL1

1.3.7.7.15 Регистр настроек PLL1 2 (RCU_PLL1CR2)

Смещение адреса: 0x040.

Значение по сбросу: 0x8080_8080.

Таблица 1.156 – Описание полей регистра RCU_PLL1CR2

Биты	Название	Доступ	Значение по сбросу	Описание
[31:24]	PLL1D	R/W	0x80	Режим делителя PLL1D: 0b00000000 - деление на 1; 0b00000001 - деление на 2; … 0b01111111 - деление на 128; 0b1xxxxxxx - выключение делителя.
[23:16]	PLL1C	R/W	0x80	Режим делителя PLL1C: 0b00000000 - деление на 1; 0b00000001 - деление на 2; … 0b01111111 - деление на 128; 0b1xxxxxxx - выключение делителя.
[15:8]	PLL1B	R/W	0x80	Режим делителя PLL1B: 0b00000000 - деление на 1; 0b00000001 - деление на 2; … 0b01111111 - деление на 128; 0b1xxxxxxx - выключение делителя.

Биты	Название	Доступ	Значение по сбросу	Описание
[7:0]	PLL1A	R/W	0x80	Режим делителя PLL1A: 0b00000000 - деление на 1; 0b00000001 - деление на 2; … 0b01111111 - деление на 128; 0b1xxxxxxx - выключение делителя.

1.3.7.7.16 Регистр разрешения тактирования периферии 0 (RCU_PCER0)

Смещение адреса: 0x050.

Значение по сбросу: 0x0000_8000.

Таблица 1.157 – Описание полей регистра RCU_PCER0

Биты	Название	Доступ	Значение по сбросу	Описание
[31]	GPIO_D_EN	R/W	0x0	Разрешение тактирования GPIO_D: 0x0 - тактирование GPIO_D выключено; 0x1 - тактирование GPIO_D включено.
[30]	CAN_D_EN	R/W	0x0	Разрешение тактирования CAN_D: 0x0 - тактирование CAN_D выключено; 0x1 - тактирование CAN_D включено.
[29]	I2C_D_EN	R/W	0x0	Разрешение тактирования I2C_D: 0x0 - тактирование I2C_D выключено; 0x1 - тактирование I2C_D включено.
[28]	SPI_D_EN	R/W	0x0	Разрешение тактирования SPI_D: 0x0 - тактирование SPI_D выключено; 0x1 - тактирование SPI_D включено.
[27]	TIM_D1_EN	R/W	0x0	Разрешение тактирования TIM_D1: 0x0 - тактирование TIM_D1 выключено; 0x1 - тактирование TIM_D1 включено.
[26]	TIM_D0_EN	R/W	0x0	Разрешение тактирования TIM_D0: 0x0 - тактирование TIM_D0 выключено; 0x1 - тактирование TIM_D0 включено.
[25]	USART_D1_EN	R/W	0x0	Разрешение тактирования USART_D1: 0x0 - тактирование USART_D1 выключено; 0x1 - тактирование USART_D1 включено.
[24]	USART_D0_EN	R/W	0x0	Разрешение тактирования USART_D0: 0x0 - тактирование USART_D0 выключено; 0x1 - тактирование USART_D0 включено.
[23]	GPIO_C_EN	R/W	0x0	Разрешение тактирования GPIO_C: 0x0 - тактирование GPIO_C выключено; 0x1 - тактирование GPIO_C включено.
[22]	CAN_C_EN	R/W	0x0	Разрешение тактирования CAN_C: 0x0 - тактирование CAN_C выключено; 0x1 - тактирование CAN_C включено.
[21]	I2C_C_EN	R/W	0x0	Разрешение тактирования I2C_C: 0x0 - тактирование I2C_C выключено; 0x1 - тактирование I2C_C включено.

Биты	Название	Доступ	Значение по сбросу	Описание
[20]	SPI_C_EN	R/W	0x0	Разрешение тактирования SPI_C: 0x0 - тактирование SPI_C выключено; 0x1 - тактирование SPI_C включено.
[19]	TIM_C1_EN	R/W	0x0	Разрешение тактирования TIM_C1: 0x0 - тактирование TIM_C1 выключено; 0x1 - тактирование TIM_C1 включено.
[18]	TIM_C0_EN	R/W	0x0	Разрешение тактирования TIM_C0: 0x0 - тактирование TIM_C0 выключено; 0x1 - тактирование TIM_C0 включено.
[17]	USART_C1_EN	R/W	0x0	Разрешение тактирования USART_C1: 0x0 - тактирование USART_C1 выключено; 0x1 - тактирование USART_C1 включено.
[16]	USART_C0_EN	R/W	0x0	Разрешение тактирования USART_C0: 0x0 - тактирование USART_C0 выключено; 0x1 - тактирование USART_C0 включено.
[15]	GPIO_B_EN	R/W	0x1	Разрешение тактирования GPIO_B: 0x0 - тактирование GPIO_B выключено; 0x1 - тактирование GPIO_B включено.
[14]	CAN_B_EN	R/W	0x0	Разрешение тактирования CAN_B: 0x0 - тактирование CAN_B выключено; 0x1 - тактирование CAN_B включено.
[13]	I2C_B_EN	R/W	0x0	Разрешение тактирования I2C_B: 0x0 - тактирование I2C_B выключено; 0x1 - тактирование I2C_B включено.
[12]	SPI_B_EN	R/W	0x0	Разрешение тактирования SPI_B: 0x0 - тактирование SPI_B выключено; 0x1 - тактирование SPI_B включено.
[11]	TIM_B1_EN	R/W	0x0	Разрешение тактирования TIM_B1: 0x0 - тактирование TIM_B1 выключено; 0x1 - тактирование TIM_B1 включено.
[10]	TIM_B0_EN	R/W	0x0	Разрешение тактирования TIM_B0: 0x0 - тактирование TIM_B0 выключено; 0x1 - тактирование TIM_B0 включено.
[9]	USART_B1_EN	R/W	0x0	Разрешение тактирования USART_B1: 0x0 - тактирование USART_B1 выключено; 0x1 - тактирование USART_B1 включено.
[8]	USART_B0_EN	R/W	0x0	Разрешение тактирования USART_B0: 0x0 - тактирование USART_B0 выключено; 0x1 - тактирование USART_B0 включено.
[7]	GPIO_A_EN	R/W	0x0	Разрешение тактирования GPIO_A: 0x0 - тактирование GPIO_A выключено; 0x1 - тактирование GPIO_A включено.
[6]	CAN_A_EN	R/W	0x0	Разрешение тактирования CAN_A: 0x0 - тактирование CAN_A выключено; 0x1 - тактирование CAN_A включено.
[5]	I2C_A_EN	R/W	0x0	Разрешение тактирования I2C_A: 0x0 - тактирование I2C_A выключено; 0x1 - тактирование I2C_A включено.

Биты	Название	Доступ	Значение по сбросу	Описание
[4]	SPI_A_EN	R/W	0x0	Разрешение тактирования SPI_A: 0x0 - тактирование SPI_A выключено; 0x1 - тактирование SPI_A включено.
[3]	TIM_A1_EN	R/W	0x0	Разрешение тактирования TIM_A1: 0x0 - тактирование TIM_A1 выключено; 0x1 - тактирование TIM_A1 включено.
[2]	TIM_A0_EN	R/W	0x0	Разрешение тактирования TIM_A0: 0x0 - тактирование TIM_A0 выключено; 0x1 - тактирование TIM_A0 включено.
[1]	USART_A1_EN	R/W	0x0	Разрешение тактирования USART_A1: 0x0 - тактирование USART_A1 выключено; 0x1 - тактирование USART_A1 включено.
[0]	USART_A0_EN	R/W	0x0	Разрешение тактирования USART_A0: 0x0 - тактирование USART_A0 выключено; 0x1 - тактирование USART_A0 включено.

1.3.7.7.17 Регистр разрешения тактирования периферии 1 (RCU_PCER1)

Смещение адреса: 0x054.

Значение по сбросу: 0x0100_008F.

Таблица 1.158 – Описание полей регистра RCU_PCER1

Биты	Название	Доступ	Значение по сбросу	Описание
[31]	TRACE_EN	R/W	0x0	Разрешение тактирования порта трассировки TPIU: 0x0 - тактирование TPIU выключено; 0x1 - тактирование TPIU включено.
[30:28]	—	R0	0x0	*резерв*
[27]	XSPI_EMMC1_EN	R/W	0x0	Разрешение тактирования XSPI1/EMMC1: 0x0 - тактирование XSPI1/EMMC1 выключено; 0x1 - тактирование XSPI1/EMMC1 включено.
[26]	XSPI_EMMC0_EN	R/W	0x0	Разрешение тактирования XSPI0/EMMC0: 0x0 - тактирование XSPI0/EMMC0 выключено; 0x1 - тактирование XSPI0/EMMC0 включено.
[25]	USB_FS_EN	R/W	0x0	Разрешение тактирования USB_FS: 0x0 - тактирование USB_FS выключено; 0x1 - тактирование USB_FS включено.
[24]	ETH_EN	R/W	0x1	Разрешение тактирования ETH: 0x0 - тактирование ETH выключено; 0x1 - тактирование ETH включено.
[23:18]	—	R0	0x0	*резерв*
[17]	CRC_EN	R/W	0x0	Разрешение тактирования CRC: 0x0 - тактирование CRC выключено; 0x1 - тактирование CRC включено.
[16]	TRNG_EN	R/W	0x0	Разрешение тактирования TRNG: 0x0 - тактирование TRNG выключено; 0x1 - тактирование TRNG включено.

Биты	Название	Доступ	Значение по сбросу	Описание
[15]	RTC_APB_EN	R/W	0x0	Разрешение тактирования RTC (APB-интерфейс): 0x0 - тактирование RTC выключено; 0x1 - тактирование RTC включено.
[14]	WWDT_EN	R/W	0x0	Разрешение тактирования WWDT: 0x0 - тактирование WWDT выключено; 0x1 - тактирование WWDT включено.
[13]	ELC_EN	R/W	0x0	Разрешение тактирования ELC: 0x0 - тактирование ELC выключено; 0x1 - тактирование ELC включено.
[12]	LCD_EN	R/W	0x0	Разрешение тактирования LCD: 0x0 - тактирование LCD выключено; 0x1 - тактирование LCD включено.
[11]	DMA3_EN	R/W	0x0	Разрешение тактирования DMA3: 0x0 - тактирование DMA3 выключено; 0x1 - тактирование DMA3 включено.
[10]	DMA2_EN	R/W	0x0	Разрешение тактирования DMA2: 0x0 - тактирование DMA2 выключено; 0x1 - тактирование DMA2 включено.
[9]	DMA1_EN	R/W	0x0	Разрешение тактирования DMA1: 0x0 - тактирование DMA1 выключено; 0x1 - тактирование DMA1 включено.
[8]	DMA0_EN	R/W	0x0	Разрешение тактирования DMA0: 0x0 - тактирование DMA0 выключено; 0x1 - тактирование DMA0 включено.
[7]	FMC_EN	R/W	0x1	Разрешение тактирования FMC: 0x0 - тактирование FMC выключено; 0x1 - тактирование FMC включено.
[6:4]	—	R0	0x0	*резерв*
[3]	SRAM3_EN	R/W	0x1	Разрешение тактирования SRAM3: 0x0 - тактирование SRAM3 выключено; 0x1 - тактирование SRAM3 включено.
[2]	SRAM2_EN	R/W	0x1	Разрешение тактирования SRAM2: 0x0 - тактирование SRAM2 выключено; 0x1 - тактирование SRAM2 включено.
[1]	SRAM1_EN	R/W	0x1	Разрешение тактирования SRAM1: 0x0 - тактирование SRAM1 выключено; 0x1 - тактирование SRAM1 включено.
[0]	SRAM0_EN	R/W	0x1	Разрешение тактирования SRAM0: 0x0 - тактирование SRAM0 выключено; 0x1 - тактирование SRAM0 включено.

1.3.7.7.18 Регистр выбора независимых синхросигналов периферии 0 (RCU_PICSR0)

Смещение адреса: 0x060.

Значение по сбросу: 0x0000_0000.

Таблица 1.159 – Описание полей регистра RCU_PICSR0

Биты	Название	Доступ	Значение по сбросу	Описание
[31:30]	-	R0	0x0	*резерв*
[29:28]	CAN_B_SEL	R/W	0x0	Выбор источника синхросигнала ядра CAN_B: 0b00 - выбран синхросигнал HSE; 0b01 - выбран синхросигнал HSI1; 0b10 - выбран синхросигнал PLL0B; 0b11 - *резерв*.
[27:26]	I2C_B_SEL	R/W	0x0	Выбор источника синхросигнала ядра I2C_B: 0b00 - выбран синхросигнал PCLK_PB; 0b01 - выбран синхросигнал SYSCLK; 0b10 - выбран синхросигнал HSI0; 0b11 - *резерв*.
[25:24]	SPI_B_SEL	R/W	0x0	Выбор источника синхросигнала ядра SPI_B: 0b00 - выбран синхросигнал PCLK_PB; 0b01 - выбран синхросигнал SYSCLK; 0b10 - выбран синхросигнал HSI0; 0b11 - *резерв*.
[23:22]	TIM_B1_SEL	R/W	0x0	Выбор источника синхросигнала ядра TIM_B1: 0b00 - выбран синхросигнал PCLK_PB; 0b01 - выбран синхросигнал LSI; 0b10 - выбран синхросигнал HSI0; 0b11 - выбран синхросигнал LSE.
[21:20]	TIM_B0_SEL	R/W	0x0	Выбор источника синхросигнала ядра TIM_B0: 0b00 - выбран синхросигнал PCLK_PB; 0b01 - выбран синхросигнал LSI; 0b10 - выбран синхросигнал HSI0; 0b11 - выбран синхросигнал LSE.
[19:18]	USART_B1_SEL	R/W	0x0	Выбор источника синхросигнала ядра USART_B1: 0b00 - выбран синхросигнал PCLK_PB; 0b01 - выбран синхросигнал SYSCLK; 0b10 - выбран синхросигнал HSI0; 0b11 - выбран синхросигнал LSE.
[17:16]	USART_B0_SEL	R/W	0x0	Выбор источника синхросигнала ядра USART_B0: 0b00 - выбран синхросигнал PCLK_PB; 0b01 - выбран синхросигнал SYSCLK; 0b10 - выбран синхросигнал HSI0; 0b11 - выбран синхросигнал LSE.
[15:14]	-	R0	0x0	*резерв*
[13:12]	CAN_A_SEL	R/W	0x0	Выбор источника синхросигнала ядра CAN_A: 0b00 - выбран синхросигнал HSE; 0b01 - выбран синхросигнал HSI1; 0b10 - выбран синхросигнал PLL0B; 0b11 - *резерв*.
[11:10]	I2C_A_SEL	R/W	0x0	Выбор источника синхросигнала ядра I2C_A: 0b00 - выбран синхросигнал PCLK_PA; 0b01 - выбран синхросигнал SYSCLK; 0b10 - выбран синхросигнал HSI0; 0b11 - *резерв*.

Биты	Название	Доступ	Значение по сбросу	Описание
[9:8]	SPI_A_SEL	R/W	0x0	Выбор источника синхросигнала ядра SPI_A: 0b00 - выбран синхросигнал PCLK_PA; 0b01 - выбран синхросигнал SYSCLK; 0b10 - выбран синхросигнал HSI0; 0b11 - *резерв*.
[7:6]	TIM_A1_SEL	R/W	0x0	Выбор источника синхросигнала ядра TIM_A1: 0b00 - выбран синхросигнал PCLK_PA; 0b01 - выбран синхросигнал LSI; 0b10 - выбран синхросигнал HSI0; 0b11 - выбран синхросигнал LSE.
[5:4]	TIM_A0_SEL	R/W	0x0	Выбор источника синхросигнала ядра TIM_A0: 0b00 - выбран синхросигнал PCLK_PA; 0b01 - выбран синхросигнал LSI; 0b10 - выбран синхросигнал HSI0; 0b11 - выбран синхросигнал LSE.
[3:2]	USART_A1_SEL	R/W	0x0	Выбор источника синхросигнала ядра USART_A1: 0b00 - выбран синхросигнал PCLK_PA; 0b01 - выбран синхросигнал SYSCLK; 0b10 - выбран синхросигнал HSI0; 0b11 - выбран синхросигнал LSE.
[1:0]	USART_A0_SEL	R/W	0x0	Выбор источника синхросигнала ядра USART_A0: 0b00 - выбран синхросигнал PCLK_PA; 0b01 - выбран синхросигнал SYSCLK; 0b10 - выбран синхросигнал HSI0; 0b11 - выбран синхросигнал LSE.

1.3.7.7.19 Регистр выбора независимых синхросигналов периферии 1 (RCU_PICSR1)

Смещение адреса: 0x064.

Значение по сбросу: 0x0000_0000.

Таблица 1.160 – Описание полей регистра RCU_PICSR1

Биты	Название	Доступ	Значение по сбросу	Описание
[31:30]	-	R0	0x0	*резерв*
[29:28]	CAN_D_SEL	R/W	0x0	Выбор источника синхросигнала ядра CAN_D: 0b00 - выбран синхросигнал HSE; 0b01 - выбран синхросигнал HSI1; 0b10 - выбран синхросигнал PLL0B; 0b11 - *резерв*.
[27:26]	I2C_D_SEL	R/W	0x0	Выбор источника синхросигнала ядра I2C_D: 0b00 - выбран синхросигнал PCLK_PD; 0b01 - выбран синхросигнал SYSCLK; 0b10 - выбран синхросигнал HSI0; 0b11 - *резерв*.

Биты	Название	Доступ	Значение по сбросу	Описание
[25:24]	SPI_D_SEL	R/W	0x0	Выбор источника синхросигнала ядра SPI_D: 0b00 - выбран синхросигнал PCLK_PD; 0b01 - выбран синхросигнал SYSCLK; 0b10 - выбран синхросигнал HSI0; 0b11 - *резерв*.
[23:22]	TIM_D1_SEL	R/W	0x0	Выбор источника синхросигнала ядра TIM_D1: 0b00 - выбран синхросигнал PCLK_PD; 0b01 - выбран синхросигнал LSI; 0b10 - выбран синхросигнал HSI0; 0b11 - выбран синхросигнал LSE.
[21:20]	TIM_D0_SEL	R/W	0x0	Выбор источника синхросигнала ядра TIM_D0: 0b00 - выбран синхросигнал PCLK_PD; 0b01 - выбран синхросигнал LSI; 0b10 - выбран синхросигнал HSI0; 0b11 - выбран синхросигнал LSE.
[19:18]	USART_D1_SEL	R/W	0x0	Выбор источника синхросигнала ядра USART_D1: 0b00 - выбран синхросигнал PCLK_PD; 0b01 - выбран синхросигнал SYSCLK; 0b10 - выбран синхросигнал HSI0; 0b11 - выбран синхросигнал LSE.
[17:16]	USART_D0_SEL	R/W	0x0	Выбор источника синхросигнала ядра USART_D0: 0b00 - выбран синхросигнал PCLK_PD; 0b01 - выбран синхросигнал SYSCLK; 0b10 - выбран синхросигнал HSI0; 0b11 - выбран синхросигнал LSE.
[15:14]	-	R0	0x0	*резерв*
[13:12]	CAN_C_SEL	R/W	0x0	Выбор источника синхросигнала ядра CAN_C: 0b00 - выбран синхросигнал HSE; 0b01 - выбран синхросигнал HSI1; 0b10 - выбран синхросигнал PLL0B; 0b11 - *резерв*.
[11:10]	I2C_C_SEL	R/W	0x0	Выбор источника синхросигнала ядра I2C_C: 0b00 - выбран синхросигнал PCLK_PC; 0b01 - выбран синхросигнал SYSCLK; 0b10 - выбран синхросигнал HSI0; 0b11 - *резерв*.
[9:8]	SPI_C_SEL	R/W	0x0	Выбор источника синхросигнала ядра SPI_C: 0b00 - выбран синхросигнал PCLK_PC; 0b01 - выбран синхросигнал SYSCLK; 0b10 - выбран синхросигнал HSI0; 0b11 - *резерв*.
[7:6]	TIM_C1_SEL	R/W	0x0	Выбор источника синхросигнала ядра TIM_C1: 0b00 - выбран синхросигнал PCLK_PC; 0b01 - выбран синхросигнал LSI; 0b10 - выбран синхросигнал HSI0; 0b11 - выбран синхросигнал LSE.
[5:4]	TIM_C0_SEL	R/W	0x0	Выбор источника синхросигнала ядра TIM_C0: 0b00 - выбран синхросигнал PCLK_PC; 0b01 - выбран синхросигнал LSI; 0b10 - выбран синхросигнал HSI0; 0b11 - выбран синхросигнал LSE.

Биты	Название	Доступ	Значение по сбросу	Описание
[3:2]	USART_C1_SEL	R/W	0x0	Выбор источника синхросигнала ядра USART_C1: 0b00 - выбран синхросигнал PCLK_PC; 0b01 - выбран синхросигнал SYSCLK; 0b10 - выбран синхросигнал HSI0; 0b11 - выбран синхросигнал LSE.
[1:0]	USART_C0_SEL	R/W	0x0	Выбор источника синхросигнала ядра USART_C0: 0b00 - выбран синхросигнал PCLK_PC; 0b01 - выбран синхросигнал SYSCLK; 0b10 - выбран синхросигнал HSI0; 0b11 - выбран синхросигнал LSE.

1.3.7.7.20 Регистр выбора независимых синхросигналов периферии 2 (RCU_PICSR2)

Смещение адреса: 0x068.

Значение по сбросу: 0x0000_0000.

Таблица 1.161 – Описание полей регистра RCU_PICSR2

Биты	Название	Доступ	Значение по сбросу	Описание
[31:22]	-	R0	0x0	*резерв*
[21:20]	DAC_SEL	R/W	0x0	Выбор источника синхросигнала ядра DAC: 0b00 - выбран синхросигнал HSI0; 0b01 - выбран синхросигнал HSI1; 0b10 - выбран синхросигнал HSE; 0b11 - *резерв*.
[19]	-	R0	0x0	*резерв*
[18:16]	ADC_SEL	R/W	0x0	Выбор источника синхросигнала ядра ADC: 0b000 - выбран синхросигнал HCLK; 0b001 - выбран синхросигнал SYSCLK; 0b010 - выбран синхросигнал PLL1_CKD; 0b011 - выбран синхросигнал HSE; 0b100 - выбран синхросигнал HSI0.
[15:14]	EXTINT_SEL	R/W	0x0	Выбор источника синхросигнала ядра EXTINT: 0b00 - выбран синхросигнал HSI0; 0b01 - выбран синхросигнал LSI; 0b10 - выбран синхросигнал LSE; 0b11 - *резерв*.
[13:12]	EMMC1_SEL	R/W	0x0	Выбор источника синхросигнала ядра EMMC1: 0b00 - выбран синхросигнал SYSCLK; 0b01 - выбран синхросигнал HSI1; 0b10 - выбран синхросигнал PLL0C; 0b11 - выбран синхросигнал PLL1A.
[11:10]	EMMC0_SEL	R/W	0x0	Выбор источника синхросигнала ядра EMMC0: 0b00 - выбран синхросигнал SYSCLK; 0b01 - выбран синхросигнал HSI1; 0b10 - выбран синхросигнал PLL0C; 0b11 - выбран синхросигнал PLL1A.

Биты	Название	Доступ	Значение по сбросу	Описание
[9:8]	XSPI1_SEL	R/W	0x0	Выбор источника синхросигнала ядра XSPI1: 0b00 - выбран синхросигнал SYSCLK; 0b01 - выбран синхросигнал HSI1; 0b10 - выбран синхросигнал PLL0C; 0b11 - выбран синхросигнал PLL1A.
[7:6]	XSPI0_SEL	R/W	0x0	Выбор источника синхросигнала ядра XSPI0: 0b00 - выбран синхросигнал SYSCLK; 0b01 - выбран синхросигнал HSI1; 0b10 - выбран синхросигнал PLL0C; 0b11 - выбран синхросигнал PLL1A.
[5:4]	LCD_SEL	R/W	0x0	Выбор источника синхросигнала ядра LCD: 0b00 - выбран синхросигнал HCLK; 0b01 - выбран синхросигнал LSE; 0b10 - выбран синхросигнал LSI; 0b11 - *резерв*.
[3:2]	USB_FS_SEL	R/W	0x0	Выбор источника синхросигнала USB_FS: 0b00 - выбран синхросигнал HSI1; 0b01 - выбран синхросигнал HSE; 0b10 - выбран синхросигнал PLL0E; 0b11 - выбран синхросигнал PLL1C.
[1:0]	CMU_RNG_SEL	R/W	0x0	Выбор источника синхросигнала ядер CMU и RNG: 0b00 - выбран синхросигнал SYSCLK; 0b01 - выбран синхросигнал HSI1; 0b10 - выбран синхросигнал PLL0D; 0b11 - выбран синхросигнал PLL1B.

1.3.7.7.21 Регистр сброса периферии 0 (RCU_PRR0)

Смещение адреса: 0x078.

Значение по сбросу: 0x0000_0000.

Таблица 1.162 – Описание полей регистра RCU_PRR0

Биты	Название	Доступ	Значение по сбросу	Описание
[31]	GPIO_D_RST	R/W	0x0	Управление сбросом GPIO_D: 0x0 - сброс GPIO_D не активирован; 0x1 - сброс GPIO_D активирован.
[30]	CAN_D_RST	R/W	0x0	Управление сбросом CAN_D: 0x0 - сброс CAN_D не активирован; 0x1 - сброс CAN_D активирован.
[29]	I2C_D_RST	R/W	0x0	Управление сбросом I2C_D: 0x0 - сброс I2C_D не активирован; 0x1 - сброс I2C_D активирован.
[28]	SPI_D_RST	R/W	0x0	Управление сбросом SPI_D: 0x0 - сброс SPI_D не активирован; 0x1 - сброс SPI_D активирован.
[27]	TIM_D1_RST	R/W	0x0	Управление сбросом TIM_D1: 0x0 - сброс TIM_D1 не активирован; 0x1 - сброс TIM_D1 активирован.

Биты	Название	Доступ	Значение по сбросу	Описание
[26]	TIM_D0_RST	R/W	0x0	Управление сбросом TIM_D0: 0x0 - сброс TIM_D0 не активирован; 0x1 - сброс TIM_D0 активирован.
[25]	USART_D1_RST	R/W	0x0	Управление сбросом USART_D1: 0x0 - сброс USART_D1 не активирован; 0x1 - сброс USART_D1 активирован.
[24]	USART_D0_RST	R/W	0x0	Управление сбросом USART_D0: 0x0 - сброс USART_D0 не активирован; 0x1 - сброс USART_D0 активирован.
[23]	GPIO_C_RST	R/W	0x0	Управление сбросом GPIO_C: 0x0 - сброс GPIO_C не активирован; 0x1 - сброс GPIO_C активирован.
[22]	CAN_C_RST	R/W	0x0	Управление сбросом CAN_C: 0x0 - сброс CAN_C не активирован; 0x1 - сброс CAN_C активирован.
[21]	I2C_C_RST	R/W	0x0	Управление сбросом I2C_C: 0x0 - сброс I2C_C не активирован; 0x1 - сброс I2C_C активирован.
[20]	SPI_C_RST	R/W	0x0	Управление сбросом SPI_C: 0x0 - сброс SPI_C не активирован; 0x1 - сброс SPI_C активирован.
[19]	TIM_C1_RST	R/W	0x0	Управление сбросом TIM_C1: 0x0 - сброс TIM_C1 не активирован; 0x1 - сброс TIM_C1 активирован.
[18]	TIM_C0_RST	R/W	0x0	Управление сбросом TIM_C0: 0x0 - сброс TIM_C0 не активирован; 0x1 - сброс TIM_C0 активирован.
[17]	USART_C1_RST	R/W	0x0	Управление сбросом USART_C1: 0x0 - сброс USART_C1 не активирован; 0x1 - сброс USART_C1 активирован.
[16]	USART_C0_RST	R/W	0x0	Управление сбросом USART_C0: 0x0 - сброс USART_C0 не активирован; 0x1 - сброс USART_C0 активирован.
[15]	GPIO_B_RST	R/W	0x0	Управление сбросом GPIO_B: 0x0 - сброс GPIO_B не активирован; 0x1 - сброс GPIO_B активирован.
[14]	CAN_B_RST	R/W	0x0	Управление сбросом CAN_B: 0x0 - сброс CAN_B не активирован; 0x1 - сброс CAN_B активирован.
[13]	I2C_B_RST	R/W	0x0	Управление сбросом I2C_B: 0x0 - сброс I2C_B не активирован; 0x1 - сброс I2C_B активирован.
[12]	SPI_B_RST	R/W	0x0	Управление сбросом SPI_B: 0x0 - сброс SPI_B не активирован; 0x1 - сброс SPI_B активирован.
[11]	TIM_B1_RST	R/W	0x0	Управление сбросом TIM_B1: 0x0 - сброс TIM_B1 не активирован; 0x1 - сброс TIM_B1 активирован.

Биты	Название	Доступ	Значение по сбросу	Описание
[10]	TIM_B0_RST	R/W	0x0	Управление сбросом TIM_B0: 0x0 - сброс TIM_B0 не активирован; 0x1 - сброс TIM_B0 активирован.
[9]	USART_B1_RST	R/W	0x0	Управление сбросом USART_B1: 0x0 - сброс USART_B1 не активирован; 0x1 - сброс USART_B1 активирован.
[8]	USART_B0_RST	R/W	0x0	Управление сбросом USART_B0: 0x0 - сброс USART_B0 не активирован; 0x1 - сброс USART_B0 активирован.
[7]	GPIO_A_RST	R/W	0x0	Управление сбросом GPIO_A: 0x0 - сброс GPIO_A не активирован; 0x1 - сброс GPIO_A активирован.
[6]	CAN_A_RST	R/W	0x0	Управление сбросом CAN_A: 0x0 - сброс CAN_A не активирован; 0x1 - сброс CAN_A активирован.
[5]	I2C_A_RST	R/W	0x0	Управление сбросом I2C_A: 0x0 - сброс I2C_A не активирован; 0x1 - сброс I2C_A активирован.
[4]	SPI_A_RST	R/W	0x0	Управление сбросом SPI_A: 0x0 - сброс SPI_A не активирован; 0x1 - сброс SPI_A активирован.
[3]	TIM_A1_RST	R/W	0x0	Управление сбросом TIM_A1: 0x0 - сброс TIM_A1 не активирован; 0x1 - сброс TIM_A1 активирован.
[2]	TIM_A0_RST	R/W	0x0	Управление сбросом TIM_A0: 0x0 - сброс TIM_A0 не активирован; 0x1 - сброс TIM_A0 активирован.
[1]	USART_A1_RST	R/W	0x0	Управление сбросом USART_A1: 0x0 - сброс USART_A1 не активирован; 0x1 - сброс USART_A1 активирован.
[0]	USART_A0_RST	R/W	0x0	Управление сбросом USART_A0: 0x0 - сброс USART_A0 не активирован; 0x1 - сброс USART_A0 активирован.

1.3.7.7.22 Регистр сброса периферии 1 (RCU_PRR1)

Смещение адреса: 0x07C.

Значение по сбросу: 0x0000_0000.

Таблица 1.163 – Описание полей регистра RCU_PRR1

Биты	Название	Доступ	Значение по сбросу	Описание
[31:30]	—	R0	0x0	*резерв*
[29]	XSPI1_RST	R/W	0x0	Управление сбросом XSPI1_RST: 0x0 - сброс XSPI1_RST не активирован; 0x1 - сброс XSPI1_RST активирован.

Биты	Название	Доступ	Значение по сбросу	Описание
[28]	XSPI0_RST	R/W	0x0	Управление сбросом XSPI0_RST: 0x0 - сброс XSPI0_RST не активирован; 0x1 - сброс XSPI0_RST активирован.
[27]	EMMC1_RST	R/W	0x0	Управление сбросом EMMC1_RST: 0x0 - сброс EMMC1_RST не активирован; 0x1 - сброс EMMC1_RST активирован.
[26]	EMMC0_RST	R/W	0x0	Управление сбросом EMMC0_RST: 0x0 - сброс EMMC0_RST не активирован; 0x1 - сброс EMMC0_RST активирован.
[25]	USB_FS_RST	R/W	0x0	Управление сбросом USB_FS_RST: 0x0 - сброс USB_FS_RST не активирован; 0x1 - сброс USB_FS_RST активирован.
[24]	ETH_RST	R/W	0x0	Управление сбросом ETH: 0x0 - сброс ETH не активирован; 0x1 - сброс ETH активирован.
[23:17]	—	R0	0x0	*резерв*
[16]	TRNG_RST	R/W	0x0	Управление сбросом TRNG: 0x0 - сброс TRNG не активирован; 0x1 - сброс TRNG активирован.
[15:14]	—	R0	0x0	*резерв*
[13]	ELC_RST	R/W	0x0	Управление сбросом ELC: 0x0 - сброс ELC не активирован; 0x1 - сброс ELC активирован.
[12]	LCD_RST	R/W	0x0	Управление сбросом LCD: 0x0 - сброс LCD не активирован; 0x1 - сброс LCD активирован.
[11]	DMA3_RST	R/W	0x0	Управление сбросом DMA3: 0x0 - сброс DMA3 не активирован; 0x1 - сброс DMA3 активирован.
[10]	DMA2_RST	R/W	0x0	Управление сбросом DMA2: 0x0 - сброс DMA2 не активирован; 0x1 - сброс DMA2 активирован.
[9]	DMA1_RST	R/W	0x0	Управление сбросом DMA1: 0x0 - сброс DMA1 не активирован; 0x1 - сброс DMA1 активирован.
[8]	DMA0_RST	R/W	0x0	Управление сбросом DMA0: 0x0 - сброс DMA0 не активирован; 0x1 - сброс DMA0 активирован.
[7:0]	—	R0	0x0	*резерв*

1.3.7.7.23 Регистр управления сбросом при падении напряжения питания (RCU_BORCR)

Смещение адреса: 0x09C.

Значение по сбросу: 0x0000_0000.

Таблица 1.164 – Описание полей регистра RCU_BORCR

Биты	Название	Доступ	Значение по сбросу	Описание
[31:15]	—	R0	0x0	*резерв*
[14:12]	BOR33VBATFS	R/W	0x0	Частота импульсного режима BOR33VBAT: 0x0 - непрерывный режим. 0x1 - частота LSI/6; 0x2 - частота LSI/12; … 0x7 - частота LSI/384.
[11:9]	BOR33VBATLVL	R/W	0x0	Пороговый уровень срабатывания схемы BOR33VBAT: 0b000 - 1.720 В; 0b001 - 2.021 В; 0b010 - 2.150 В; 0b011 - 2.279 В; 0b100 - 2.408 В; 0b101 - 2.580 В; 0b110 - 2.709 В; 0b111 - 2.838 В.
[8]	BOR33VBATEN	R/W	0x0	Управление схемой BOR33VBAT: 0x0 - BOR33VBAT выключен; 0x1 - BOR33VBAT включен.
[7]	—	R0	0x0	*резерв*
[6:4]	BOR33FS	R/W	0x0	Частота импульсного режима BOR33: 0x0 - непрерывный режим. 0x1 - частота LSI/6; 0x2 - частота LSI/12; … 0x7 - частота LSI/384.
[3:1]	BOR33LVL	R/W	0x0	Пороговый уровень срабатывания схемы BOR33: 0b000 - 1.720 В; 0b001 - 2.021 В; 0b010 - 2.150 В; 0b011 - 2.279 В; 0b100 - 2.408 В; 0b101 - 2.580 В; 0b110 - 2.709 В; 0b111 - 2.838 В.
[0]	BOR33EN	R/W	0x0	Управление схемой BOR33: 0x0 - BOR33 выключен; 0x1 - BOR33 включен.

1.3.7.7.24 Регистр управления блоком контроля частоты (RCU_CFACR)

Смещение адреса: 0x0C0.

Значение по сбросу: 0x0000_0000.

Таблица 1.165 – Описание полей регистра RCU_CFACR

Биты	Название	Доступ	Значение по сбросу	Описание
[31:16]	CFARPSC	R/W	0x0	Значение делителя опорного синхросигнала: $F_{REF} = F(CFARS) / (CFARPSC + 1)$.
[15:13]	—	R0	0x0	*резерв*
[12:8]	CFATPSC	R/W	0x0	Значение делителя измеряемого синхросигнала: $F_{TAR} = F(CFATS) / (CFATPSC + 1)$.
[7:6]	—	R0	0x0	*резерв*
[5:4]	CFATS	R/W	0x0	Выбор измеряемого синхросигнала: 0x0 - LSI; 0x1 - HSI0; 0x2 - HSI1; 0x3 - SYSCLK.
[3:2]	—	R0	0x0	*резерв*
[1]	CFARS	R/W	0x0	Выбор опорного синхросигнала контроля частоты: 0x0 - LSE; 0x1 - HSE.
[0]	CFAEN	R/W	0x0	Управление схемой контроля частоты: 0x0 - CFAU выключен; 0x1 - CFAU включен.

1.3.7.7.25 Регистр статуса блока контроля частоты (RCU_CFASR)

Смещение адреса: 0x0C4.

Значение по сбросу: 0x0000_0000.

Таблица 1.166 – Описание полей регистра RCU_CFASR

Биты	Название	Доступ	Значение по сбросу	Описание
[31:19]	—	R0	0x0	*резерв*
[18]	CFAOVFL	R	0x0	Флаг переполнения счетчика. Данный бит обновляется в конце каждого периода измерения.
[17]	CFAFERR	R	0x0	Флаг ошибки частоты. Данный бит обновляется в конце каждого периода измерения.
[16]	CFACMPL	R	0x0	Флаг завершения измерения. Данный бит инвертируется в конце каждого периода измерения.
[15:0]	CFACBR	R	0x0	Буфер счетчика контроля частоты. Данное поле обновляется в конце каждого периода измерения.

1.3.7.7.26 Регистр верхнего порога счетчика контроля частоты (RCU_CFAULR)

Смещение адреса: 0x0C8.

Значение по сбросу: 0x0000_0000.

Таблица 1.167 – Описание полей регистра RCU_CFAULR

Биты	Название	Доступ	Значение по сбросу	Описание
[31:16]	—	R0	0x0	*резерв*
[15 :0]	CFAUL	R/W	0x0	Значение верхнего порога счетчика контроля частоты.

1.3.7.7.27 Регистр нижнего порога счетчика контроля частоты (RCU_CFALLR)

Смещение адреса: 0x0CC.

Значение по сбросу: 0x0000_0000.

Таблица 1.168 – Описание полей регистра RCU_CFALLR

Биты	Название	Доступ	Значение по сбросу	Описание
[31:16]	—	R0	0x0	*резерв*
[15 :0]	CFALL	R/W	0x0	Значение нижнего порога счетчика контроля частоты.

1.3.7.7.28 Регистр маски прерываний (RCU_IRQER)

Смещение адреса: 0x0E0.

Значение по сбросу: 0x0000_0000.

Регистр RCU_IRQER служит для маскирования событий прерываний. При единичном значении битов регистра RCU_IRQER возникновение событий прерываний приводит к установке соответствующих битов регистра RCU_IRQSR и формированию запроса в системный контроллер прерываний. При нулевом значении битов события прерываний маскируются.

Таблица 1.169 – Описание полей регистра RCU_IRQER

Биты	Название	Доступ	Значение по сбросу	Описание
[31:23]	—	R0	0x0	*резерв*
[22]	IRQE_CFAOVFL	R/W	0x0	Переполнение счетчика контроля частоты
[21]	IRQE_CFAFERR	R/W	0x0	Выход измеряемой частоты за заданные пределы
[20]	IRQE_CFACMPL	R/W	0x0	Завершение измерения в блоке контроля частоты CFAU
[19:7]	—	R0	0x0	*резерв*
[6]	IRQE_PLL1RDY	R/W	0x0	Переход генератора PLL1 в рабочий режим
[5]	IRQE_PLL0RDY	R/W	0x0	Переход генератора PLL0 в рабочий режим
[4]	IRQE_HSERDY	R/W	0x0	Переход генератора HSE в рабочий режим
[3]	IRQE_HSI1RDY	R/W	0x0	Переход генератора HSI1 в рабочий режим
[2]	IRQE_HSI0RDY	R/W	0x0	Переход генератора HSI0 в рабочий режим

Биты	Название	Доступ	Значение по сбросу	Описание
[1]	IRQE_LSERDY	R/W	0x0	Переход генератора LSE в рабочий режим
[0]	IRQE_LSIRDY	R/W	0x0	Переход генератора LSI в рабочий режим

1.3.7.7.29 Регистр статуса прерываний (RCU_IRQSR)

Смещение адреса: 0x0E4.

Значение по сбросу: 0x0000_0000.

Регистр RCU_IRQSR содержит информацию, отображающую состояние прерываний блока RCU. Статус прерываний отображается после маскирования, которое настраивается в регистре RCU_IRQER. При возникновении определенного события и отсутствии маскирования соответствующей бит регистра RCU_IRQSR аппаратно устанавливается в единичное значение. Сброс статуса прерываний осуществляется аппаратным сбросом или программно через регистр RCU_IRQCR.

Таблица 1.170 – Описание полей регистра RCU_IRQSR

Биты	Название	Доступ	Значение по сбросу	Описание
[31:23]	—	R0	0x0	*резерв*
[22]	IRQS_CFAOVFL	R	0x0	Переполнение счетчика контроля частоты
[21]	IRQS_CFAFERR	R	0x0	Выход измеряемой частоты за заданные пределы
[20]	IRQS_CFACMPL	R	0x0	Завершение измерения в блоке контроля частоты CFAU
[19:18]	—	R0	0x0	*резерв*
[17]	IRQS_LSEFAIL	R	0x0	Отказ генератора LSE
[16]	IRQS_HSEFAIL	R	0x0	Отказ генератора HSE
[15:7]	—	R0	0x0	*резерв*
[6]	IRQS_PLL1RDY	R	0x0	Переход генератора PLL1 в рабочий режим
[5]	IRQS_PLL0RDY	R	0x0	Переход генератора PLL0 в рабочий режим
[4]	IRQS_HSERDY	R	0x0	Переход генератора HSE в рабочий режим
[3]	IRQS_HSI1RDY	R	0x0	Переход генератора HSI1 в рабочий режим
[2]	IRQS_HSI0RDY	R	0x0	Переход генератора HSI0 в рабочий режим
[1]	IRQS_LSERDY	R	0x0	Переход генератора LSE в рабочий режим
[0]	IRQS_LSIRDY	R	0x0	Переход генератора LSI в рабочий режим

1.3.7.7.30 Регистр сброса прерываний (RCU_IRQCR)

Смещение адреса: 0x0EC.

Значение по сбросу: 0x0000_0000.

Регистр RCU_IRQCR служит для сброса статуса прерываний, который отображается в регистре RCU_IRQSR. Для сброса какого-либо прерывания необходимо в соответствующий бит регистра RCU_IRQCR записать единицу. Запись нулевого значения игнорируется. Чтение регистра RCU_IRQCR всегда возвращает значение 0.

Таблица 1.171 – Описание полей регистра RCU_IRQCR

Биты	Название	Доступ	Значение по сбросу	Описание
[31:23]	—	R0	0x0	*резерв*
[22]	IRQC_CFAOVFL	R/W1C	0x0	Переполнение счетчика контроля частоты
[21]	IRQC_CFAFERR	R/W1C	0x0	Выход измеряемой частоты за заданные пределы
[20]	IRQC_CFACMPL	R/W1C	0x0	Завершение измерения в блоке контроля частоты CFAU
[19:18]	—	R0	0x0	*резерв*
[17]	IRQC_LSEFAIL	R0/W1C	0x0	Отказ генератора LSE
[16]	IRQC_HSEFAIL	R0/W1C	0x0	Отказ генератора HSE
[15:7]	—	R0	0x0	*резерв*
[6]	IRQC_PLL1RDY	R0/W1C	0x0	Переход генератора PLL1 в рабочий режим
[5]	IRQC_PLL0RDY	R0/W1C	0x0	Переход генератора PLL0 в рабочий режим
[4]	IRQC_HSERDY	R0/W1C	0x0	Переход генератора HSE в рабочий режим
[3]	IRQC_HSI1RDY	R0/W1C	0x0	Переход генератора HSI1 в рабочий режим
[2]	IRQC_HSI0RDY	R0/W1C	0x0	Переход генератора HSI0 в рабочий режим
[1]	IRQC_LSERDY	R0/W1C	0x0	Переход генератора LSE в рабочий режим
[0]	IRQC_LSIRDY	R0/W1C	0x0	Переход генератора LSI в рабочий режим

1.3.7.7.31 Регистр конфигурации резервного домена (RCU_BDCR)

Смещение адреса: 0x0F0.

Значение по сбросу: 0x0001_8880.

Данный регистр сбрасывается сбросом резервного домена (см. раздел 1.3.7.3.3), за исключением битов BDRST, LSCOSEL и LSCOEN, которые сбрасываются только сбросом по питанию резервного домена. Любые другие внешние или внутренние источники сброса не влияют на данный регистр.

Таблица 1.172 – Описание полей регистра RCU_BDCR

Биты	Название	Доступ	Значение по сбросу	Описание
[31]	BDRST	R/W	0x0	Программный сброс резервного домена: 0x0 - сброс резервного домена неактивен; 0x1 - сброс резервного домена активен.
[30:28]	—	R0	0x0	*резерв*

Биты	Название	Доступ	Значение по сбросу	Описание
[27:26]	LSI_TEST	R/W	0x0	Выбор тестового режима LSI (см. спецификацию на SGC21547_32K_HDX01).
[25]	LSCOSEL	R/W	0x0	Выбор синхросигнала для внешнего вывода LSCO: 0x0 - выбран синхросигнал LSI; 0x1 - выбран синхросигнал LSE.
[24]	LSCOEN	R/W	0x0	Разрешение выдачи синхросигнала на внешний вывод LSCO: 0x0 - синхросигнал LSCO выключен; 0x1 - синхросигнал LSCO включен.
[23]	RTC_EN	R/W	0x0	Разрешение тактирования RTC: 0x0 - тактирование RTC выключено; 0x1 - тактирование RTC включено;
[22]	—	R0	0x0	*резерв*
[21:20]	RTCSEL	R/W	0x0	Выбор источника синхросигнала ядра RTC: 0b00 - синхросигнал RTC выключен; 0b01 - выбран синхросигнал LSE; 0b10 - выбран синхросигнал LSI; 0b11 - выбран синхросигнал HSE/1024.
[19]	LSIPRE	R/W	0x0	Управление делителем частоты LSI: 0x0 - делитель выключен; 0x1 - частота LSI делится на 128.
[18:10]	LSIFT	R/W	0x62	Коэффициент подстройки частоты LSI.
[9]	LSIRDY	R	0x0	Флаг готовности генератора LSI: 0x0 - синхросигнал LSI нестабилен; 0x1 - синхросигнал LSI стабилен.
[8]	LSION	R/W	0x0	Управление генератором LSI: 0x0 - генератор LSI выключен; 0x1 - генератор LSI включен.
[7]	LSESYSEN	R/W	0x1	Разрешение тактирования системного дерева синхросигнала LSE: 0x0 - LSE тактирует только RTC и LSECFD; 0x1 - LSE тактирует все блоки, предусмотренные спецификацией RCU.
[6]	LSEFAIL	R	0x0	Статус детектора отказов генератора LSE: 0x0 - отказ LSE не был обнаружен; 0x1 - отказ LSE был обнаружен.
[5]	LSECFDEN	R/W	0x0	Управление детектором отказов генератора LSE: 0x0 - детектор отказов LSE выключен; 0x1 - детектор отказов LSE включен.
[4:3]	—	R0	0x0	*резерв*
[2]	LSEBYP	R/W	0x0	Режим обхода генератора LSE: 0x0 - LSE используется; 0x1 - обход LSE внешним синхросигналом со входа OSC32_IN. Данный бит допускается изменять только когда LSE находится в выключенном состоянии (LSEON = 0).
[1]	LSERDY	R	0x0	Флаг готовности генератора LSE: 0x0 - синхросигнал LSE нестабилен; 0x1 - синхросигнал LSE стабилен.

Биты	Название	Доступ	Значение по сбросу	Описание
[0]	LSEON	R/W	0x0	Управление генератором LSE: 0x0 - генератор LSE выключен; 0x1 - генератор LSE включен.

1.3.7.7.32 Регистр статуса сбросов (RCU_RSR)

Смещение адреса: 0x0F4.

Значение по сбросу: 0x0000_0003.

Регистр RCU_RSR содержит статусную информацию о событиях сброса. При возникновении одного из возможных событий сброса соответствующий бит регистра аппаратно устанавливается в единицу. Любая программная запись сбрасывает все биты регистра. Регистрация событий сброса, кроме сброса по питанию BORRST, имеет накопительный эффект - при возникновении нового события сброса не происходит аппаратной очистки битов других событий сброса.

Таблица 1.173 – Описание полей регистра RCU_RSR

Биты	Название	Доступ	Значение по сбросу	Описание
[31:6]	—	R0	0x0	*резерв*
[5]	LPMRST	R/WC	0x0	Статус сброса после выхода из режима низкого потребления SHUTDOWN: 0x0 - сброс LPMRST не был активирован; 0x1 - сброс LPMRST был активирован.
[4]	WWDGRST	R/WC	0x0	Статус сброса от оконного сторожевого таймера WWDG: 0x0 - сброс WWDG не был активирован; 0x1 - сброс WWDG был активирован.
[3]	IWDGRST	R/WC	0x0	Статус сброса от независимого сторожевого таймера IWDG: 0x0 - сброс IWDG не был активирован; 0x1 - сброс IWDG был активирован.
[2]	SWRST	R/WC	0x0	Статус программного сброса: 0x0 - программный сброс не был активирован; 0x1 - программный сброс был активирован.
[1]	PINRST	R/WC	0x1	Статус сброса от внешнего вывода NRST: 0x0 - сброс NRST не был активирован; 0x1 - сброс NRST был активирован.
[0]	PBORRST	R/WC	0x1	Статус сброса по питанию: 0x0 - сброс по питанию не был активирован; 0x1 - сброс по питанию был активирован.

1.3.8 Система управления питанием (PMU)

1.3.8.1 Схема и домены питания МК Optimal+

1.3.8.1.1 Схема питания МК Optimal+

Схема питания МК Optimal+ показана на рисунке 1.27.

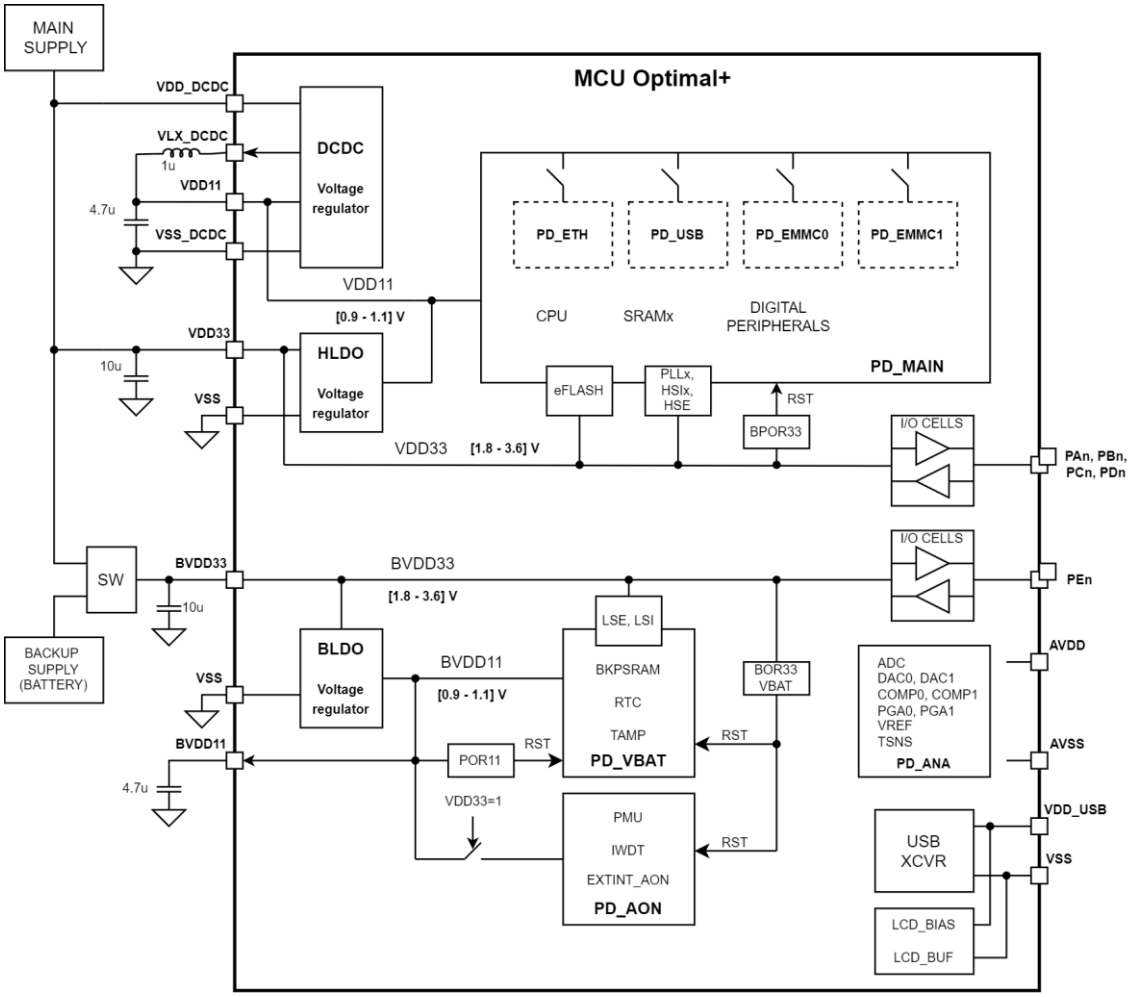


Рисунок 1.27 – Схема питания МК Optimal+

1.3.8.1.2 Порты питания и земли

В таблице 1.174 приведен список портов питания-земли МК Optimal+. Допустимые уровни напряжения для портов питания приведены в разделе 1.2.4.

Таблица 1.174 – Порты питания/земли МК Optimal+

Порт	Описание
VDD33	Основное внешнее питание (I/O-буфера основного домена, регулятора HLDO, осцилляторы - HSE, HSIn, PLLn, встроенная Flash-память)

VDD_DCDC	Внешнее питание регулятора DCDC, подключается к тому же источнику что и VDD33. Если DCDC не используется, подключается к земле
VLX_DCDC	Выход регулятора DCDC, подключается через индуктивность к выводу VDD11. Если DCDC не используется, подключается к земле
VDD11	Питание ядра основного домена, управляется регуляторами HLDO или DCDC
BVDD33	Внешнее питание батарейного домена (I/O-буфера батарейного домена, регулятор BLDO, осцилляторы - LSE, LSI). Если функция батарейного домена не используется, подключается к VDD33
BVDD11	Питание ядра батарейного домена, управляется регулятором BLDO
AVDD	Питание аналоговой подсистемы. Если аналоговая периферия не используется, подключается к VDD33
VDD_USB	Совмещенное питание трансивера USB и буферов LCD
VSS	Основная земля
VSS_DCDC	Земля регулятора DCDC
AVSS	Аналоговая земля

1.3.8.1.3 Встроенные регуляторы напряжения (HLDO, DCDC, BLDO)

Для преобразования внешнего напряжения питания, которое имеет диапазон 1.8 - 3.6 В, до уровня 0.9 - 1.1 В, необходимого для питания цифрового ядра, МК Optimal+ имеет в своем составе несколько встроенных регуляторов напряжения:

Линейный преобразователь HLDO

Линейный регулятор HLDO типа Low-dropout по-умолчанию является источником питания для основного домена PD_MAIN. Регулятор HLDO отличается отсутствием помех, однако имеет невысокий КПД. Выходное напряжение HLDO управляется контроллером PMU и имеет три фиксированных номинальных уровня - 0.9 В, 1.0 В, 1.1. В.

Импульсный преобразователь DCDC

Импульсный понижающий преобразователь DCDC (Buck-converter) может использоваться для питания основного домена PD_MAIN. Преобразователь DCDC отличается высоким КПД по сравнению с HLDO, однако его работа вносит шум по питанию. Выходное напряжение DCDC управляется контроллером PMU и имеет три фиксированных номинальных уровня - 0.9 В, 1.0 В, 1.1. В.

Линейный преобразователь BLDO

Линейный преобразователь BLDO типа Low-dropout, имеющий ограниченную нагрузочную способность и низкий ток покоя Iq, предназначен для питания логики резервного домена PD_VBAT. Регулятор BLDO всегда включен при наличии внешнего питания BVDD33. Выходное напряжение BLDO управляется контроллером PMU и имеет три фиксированных номинальных уровня - 0.9 В, 1.0 В, 1.1. В.

1.3.8.1.4 Блоки контроля напряжения питания (POR-BOR)

Для корректной инициализации микросхемы при подаче и снятии питания МК Optimal+ имеет в своем составе несколько блоков контроля напряжения питания, которые формируют сигналы сброса при подаче питания (POR - Power-on reset) или при отключении питания (BOR - Brown-out reset). Всего имеется три таких блока:

POR11

Блок POR11 служит для инициализации батарейного домена PD_VBAT при подаче внешнего питания BVDD33 и запуске регулятора BLDO. Временная диаграмма работы POR11 представлена на рисунке 1.28.

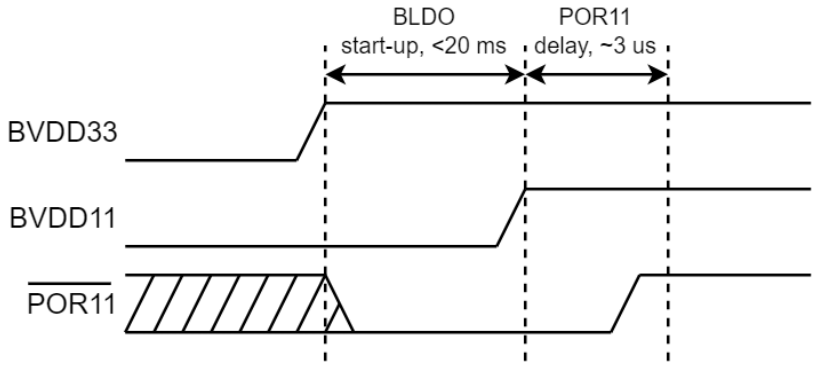


Рисунок 1.28 – Временная диаграмма POR11

BOR33VBAT

Блок BOR33VBAT служит для инициализации батарейного домена PD_VBAT при снижении напряжения питания на выводе BVDD33 ниже заданного уровня. Временная диаграмма работы BOR33VBAT представлена на рисунке 1.29.

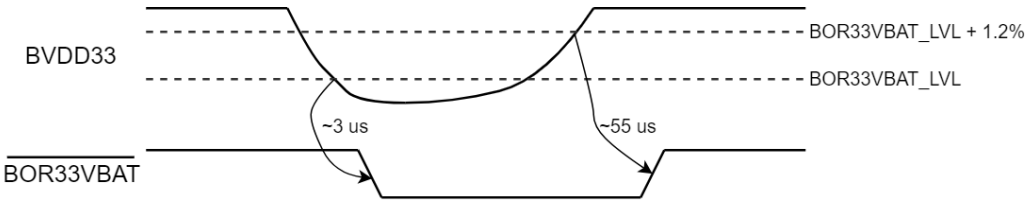


Рисунок 1.29 – Временная диаграмма BOR33VBAT

Функция BOR33VBAT управляется регистром RCU_BORCR (см. таблицу 1.164).

BPOR33

Блок BPOR33 совмещает функции сброса Power-on и Brown-out, и служит для инициализации основного домена PD_MAIN при подаче и снятии питания VDD33. Временная диаграмма работы BPOR33 представлена на рисунке 1.30.

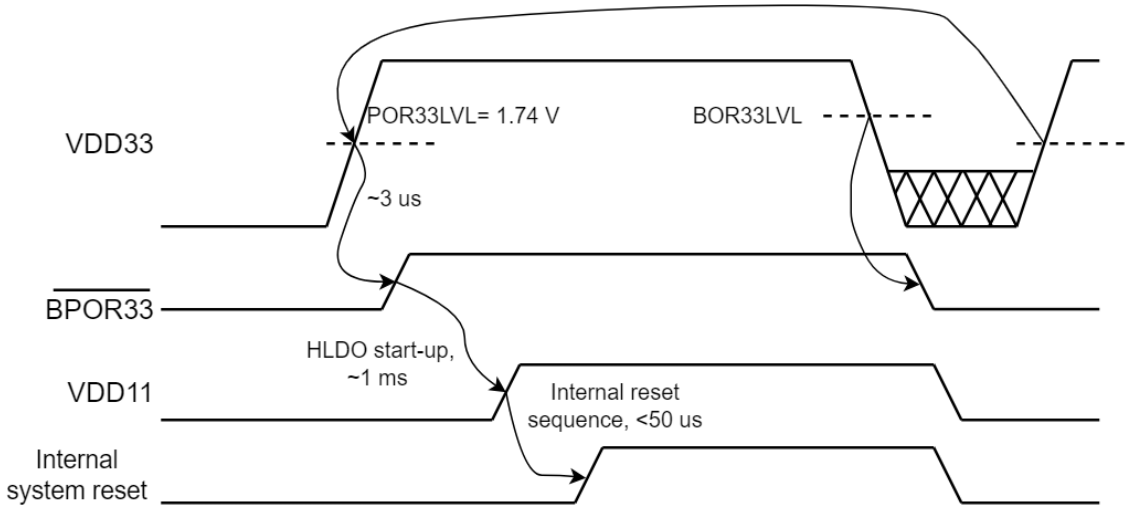


Рисунок 1.30 – Временная диаграмма BPOR33

Функция BOR33 управляется регистром RCU_BORCR (см. таблицу 1.164).

1.3.8.1.5 Домены питания

Доменом питания называется группа схем, имеющая общий источник напряжения питания, например выход регулятора или выключатель питания (power switch). МК Optimal+ имеет следующие домены питания:

– **PD_MAIN** - основной домен питания, содержащий процессорное ядро, встроенное ОЗУ, основную периферию и т.п. Домен PD_MAIN получает питание от регулятора HLDO или DCDC. В целях энергосбережения домен PD_MAIN может быть полностью обесточен путем выключения обоих регуляторов HLDO и DCDC. Домен PD_MAIN содержит в себе несколько суб-доменов, которые можно программно выключать и включать с помощью встроенных выключателей питания (power switch):

– **PD_ETH** - домен, содержащий контроллер интерфейса Ethernet 10/100;

– **PD_USB** - домен, содержащий контроллер интерфейса USB Full-speed;

– **PD_EMMC0** - домен, содержащий контроллер интерфейса EMMC0;

– **PD_EMMC1** - домен, содержащий контроллер интерфейса EMMC1;

– **PD_VBAT** - домен, содержащий часы реального времени RTC, блок TAMP, и резервное ОЗУ BKPSRAM. Домен PD_VBAT получает питание от регулятора BLDO и всегда включен при наличии внешнего питания BVDD33.

– **PD_AON** - always-on домен, содержащий контроллер управления питанием PMU, независимый сторожевой таймер IWDT и логику пробуждения из режимов низкого энергопотребления. Домен PD_AON получает питание от регулятора BLDO через выключатель, который

управляется схемой контроля напряжения BPOR33. Домен PD_AON включен только при наличии внешнего питания VDD33.

– **PD_ANA** - домен, содержащий блоки аналоговой подсистемы. Домен PD_ANA питается от отдельных портов питания и земли AVDD/AVSS.

1.3.8.2 Режимы энергопотребления МК Optimal+

1.3.8.2.1 Перечень режимов энергопотребления

Режимом энергопотребления называется предопределенная комбинация состояний регуляторов напряжения (HLDO, DCDC, BLDO) и доменов питания. В таблице 1.175 приведен перечень режимов энергопотребления МК Optimal+.

Таблица 1.175 – Режимы энергопотребления

Режимы	**RUN.HLDO_xx**	**RUN.DCDC_xx**	**DEEPSLEEP0**	**DEEPSLEEP1**	**SHUTDOWN**	**VBAT**
Условие входа	-	-	WFI + SLEEPDEEP + WIU_ICR_0 = 0x1	WFI + SLEEPDEEP + WIU_ICR_0 = 0x2	WFI + SLEEPDEEP + WIU_ICR_0 = 0x3	VDD33 power-off
Условие выхода	-	-	WKUP/NRST pin edge, RTC event, IWDT event, LSE-FAIL, Debug access	WKUP/NRST pin edge, RTC event, IWDT event, LSE-FAIL, Debug access	WKUP/NRST pin edge, RTC event, IWDT event, LSE-FAIL	VDD33 power-on
Домен	Состояние доменов					
PD_MAIN	ON	ON	CLKOFF	CLKOFF	OFF	OFF
PD_ETH	OPT	OPT	CLKOFF/OFF[1)]	OFF	OFF	OFF
PD_USB	OPT	OPT	CLKOFF/OFF[1)]	OFF	OFF	OFF
PD_EMMC0	OPT	OPT	CLKOFF/OFF[1)]	OFF	OFF	OFF
PD_EMMC1	OPT	OPT	CLKOFF/OFF[1)]	OFF	OFF	OFF
PD_AON	ON	ON	ON	ON	ON	OFF
PD_VBAT	ON	ON	ON	ON	ON	ON
Регулятор	Состояние регуляторов					
HLDO	0.9 - 1.1 В	HIZ	0.9 В	0.9 В	Pull-down	Pull-down
DCDC	HIZ	0.9 - 1.1 В	HIZ	HIZ	Pull-down	Pull-down
BLDO	0.9 - 1.1 В	0.9 - 1.1 В	0.9 В	0.9 В	0.9 В	1.1 В

Примечания:

– ON - питание и синхросигналы домена включены;
– CLKOFF - питание домена включено, синхросигналы выключены;
– OFF - питание домена выключено;
– OPT - состояние домена определяется программными настройками PMU и RCU.

[1)] Домен находится в состоянии OFF если до входа в режим DEEPSLEEP0 он был программно отключен через регистры PMU.

1.3.8.2.2 Режим нормальной работы RUN

Режим RUN является основным режимом работы, при котором выполняется приложение. В этом режиме домен PD_MAIN питается от регулятора HLDO или DCDC, с включенными

системными синхросигналами. В зависимости от типа питающего регулятора и уровня напряжения, режим RUN делится на несколько подрежимов:

– RUN.HLDO_11 - работа от регулятора HLDO с выходным уровнем напряжения 1.1 В. Это режим по-умолчанию после подачи внешнего питания VDD33;

– RUN.HLDO_10 - работа от регулятора HLDO с выходным уровнем напряжения 1.0 В;

– RUN.HLDO_09 - работа от регулятора HLDO с выходным уровнем напряжения 0.9 В;

– RUN.DCDC_11 - работа от регулятора DCDC с выходным уровнем напряжения 1.1 В;

– RUN.DCDC_10 - работа от регулятора DCDC с выходным уровнем напряжения 1.0 В;

– RUN.DCDC_09 - работа от регулятора DCDC с выходным уровнем напряжения 0.9 В.

В зависимости от требований исполняемого приложения к производительности и энергоэффективности, ток потребления в режиме RUN можно регулировать несколькими способами:

– настройка выходного уровня напряжения регуляторов HLDO и DCDC с помощью контроллера PMU;

– индивидуальное отключение питания доменов PD_ETH, PD_USB, PD_EMMC0, PD_EMMC1 с помощью контроллера PMU;

– индивидуальное отключение питания банков встроенного ОЗУ с помощью контроллера AHB5_SRAM;

– настройка частот синхросигналов с помощью контроллера RCU;

– отключение синхросигналов отдельных блоков с помощью контроллера RCU.

1.3.8.2.3 Режим пониженного энергопотребления DEEPSLEEP0

Режим пониженного энергопотребления DEEPSLEEP0 переводит микроконтроллер в следующее состояние:

– все синхросигналы в основном домене PD_MAIN принудительно отключаются;

– питание основного домена PD_MAIN принудительно переводится на регулятор HLDO с выходным уровнем 0.9 В.

В режиме DEEPSLEEP0 сохраняется содержимое всех регистров и банков памяти.

1.3.8.2.4 Режим пониженного энергопотребления DEEPSLEEP1

Режим пониженного энергопотребления DEEPSLEEP1 переводит микроконтроллер в следующее состояние:

– все синхросигналы в основном домене PD_MAIN принудительно отключаются;

– питание основного домена PD_MAIN принудительно переводится на регулятор HLDO с выходным уровнем 0.9 В;

– питание доменов PD_ETH, PD_USB, PD_EMMC0, PD_EMMC1 принудительно отключается.

В режиме DEEPSLEEP1 сохраняется содержимое всех регистров и банков памяти, кроме тех, которые входят в отключаемые домены PD_ETH, PD_USB, PD_EMMC0, PD_EMMC1.

1.3.8.2.5 Режим пониженного энергопотребления SHUTDOWN

Режим пониженного энергопотребления SHUTDOWN переводит микроконтроллер в следующее состояние:

– регуляторы HLDO и DCDC выключаются с переводом выходов в Pull-down;

– домен PD_MAIN полностью обесточен.

В режиме SHUTDOWN теряется содержимое всех регистров и банков памяти, кроме тех, которые входят в домен PD_VBAT.

1.3.8.2.6 Режим батарейного домена VBAT

Режим батарейного домена характеризуется наличием питания на выводе BVDD33 и отсутствием основного питания VDD33. В этом режиме домен PD_VBAT может сохранять функциональность, задействованную до отключения питания VDD33, а именно:

– часы реального времени RTC и блок TAMP могут продолжать работать, используя синхросигнал от осциллятора LSE или LSI;

– Backup-регистры и Backup-SRAM хранят данные, записанные до отключения питания VDD33;

– функция детектора отказа осциллятора LSE (LSECFD) может продолжать работу;

– блок контроля напряжения BOR33VBAT может продолжать работу в непрерывном или импульсном режиме.

Примечание: для того, чтобы домен PD_VBAT корректно продолжил работу после отключения питания VDD33, на момент отключения питания VDD33 должен быть включен блок контроля напряжения питания BOR33.

1.3.8.3 Способы переключения режимов энергопотребления

1.3.8.3.1 Диаграмма переходов между режимами энергопотребления

Диаграмма возможных переходов между режимами энергопотребления показана на рисунке 1.31.

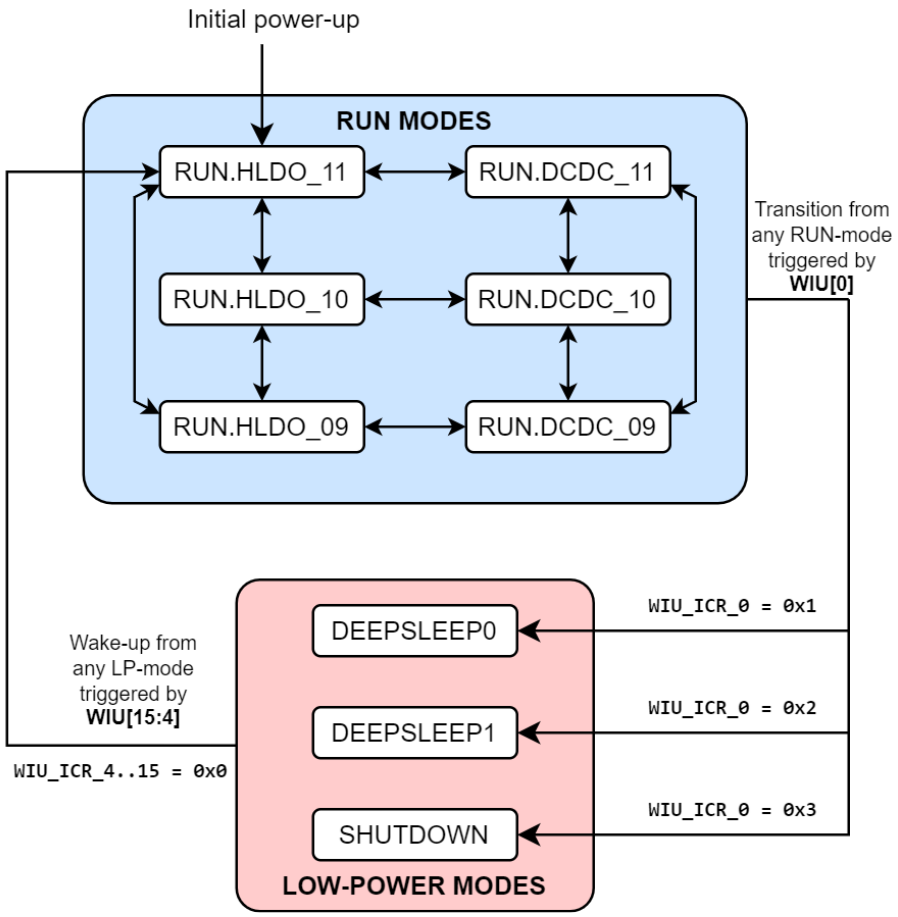


Рисунок 1.31 – Диаграмма переходов между режимами энергопотребления МК Optimal+

1.3.8.3.2 Контроллер управления питанием PMU

1.3.8.3.2.1 Общее описание

Переходы между режимами энергопотребления осуществляются контроллером управления питания PMU. Контроллер PMU позволяет управлять состоянием встроенных регуляторов (HLDO, DCDC, BLDO) и доменов питания. Структурная схема контроллера PMU показана на рисунке 1.32.

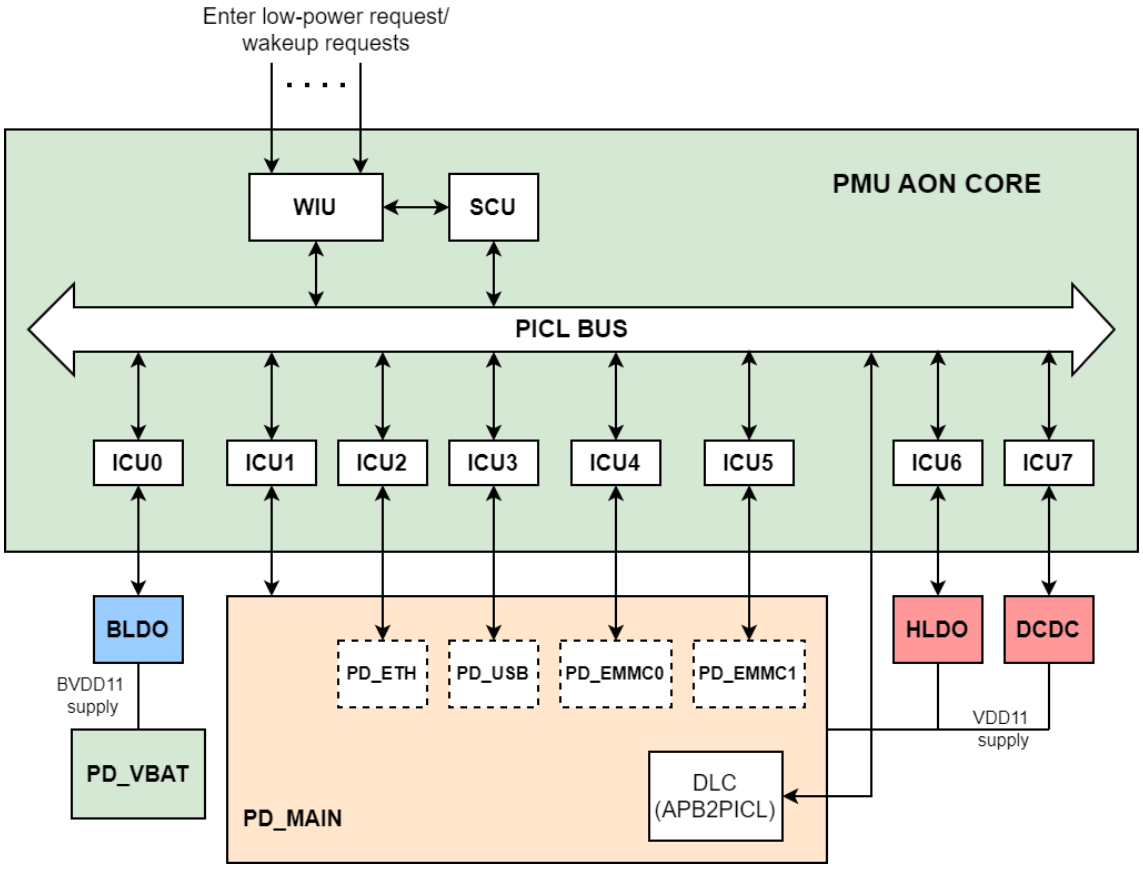


Рисунок 1.32 – Структурная схема контроллера PMU

Контроллер PMU состоит из следующих основных узлов:

– Блоки ICUn (Island Control Unit) - исполнительные устройства, управляющие состоянием регуляторов напряжения или доменов питания. Всего есть 8 блоков ICU:

– ICU0 - управляет состоянием регулятора BLDO;

– ICU1 - управляет состоянием домена PD_MAIN;

– ICU2 - управляет состоянием домена PD_ETH;

– ICU3 - управляет состоянием домена PD_USB;

– ICU4 - управляет состоянием домена PD_EMMC0;

– ICU5 - управляет состоянием домена PD_EMMC1;

– ICU6 - управляет состоянием регулятора HLDO;

– ICU7 - управляет состоянием регулятора DCDC.

– Блок SCU (Shift Control Unit) - блок, выполняющий последовательности для переходов между режимами энергопотребления.

– Блок WIU (Wake-up Interrupt Unit) - контроллер прерываний, который запускает блок SCU по внешним событиям или программному запросу.

– Блок DLC (Domain Link Controller) - интерфейс для программного доступа к узлам PMU со стороны процессора.

Узлы PMU взаимодействуют по внутренней шине PICL (Power Island Control Link).

1.3.8.3.2.2 Описание работы PMU

Каждому режиму энергопотребления, описанному в разделе 1.3.8.2, соответствует определенная комбинация состояний блоков ICU. Состояние каждого блока ICU можно программно изменять в процессе работы через соответствующие регистры ICUn_MT_CR, однако основным механизмом выполнения переходов между режимами энергопотребления является использование контроллера прерываний WIU. Контроллер WIU по аппаратному событию или программному запросу запускает блок SCU, который в автономном режиме выполняет необходимые операции на шине PICL по конфигурации блоков ICU. Выполняемая последовательность для каждой линии WIU задается в соответствующем регистре WIU_ICR_n. В таблице 1.176 приведен список последовательностей и их коды, а в таблице 1.177 приведено подключение линий WIU к аппаратным событиям.

Таблица 1.176 – Последовательности переходов между режимами энергопотребления

Код последовательности	**Описание**
0x00	Переход DEEPSLEEPx, SHUTDOWN → RUN.HLDO_11
0x01	Переход RUN.* → DEEPSLEEP0
0x02	Переход RUN.* → DEEPSLEEP1
0x03	Переход RUN.* → SHUTDOWN
0x04	Переход RUN.DCDC_09 → RUN.DCDC_10
0x05	Переход RUN.DCDC_09 → RUN.DCDC_11
0x06	Переход RUN.DCDC_10 → RUN.DCDC_09
0x07	Переход RUN.DCDC_10 → RUN.DCDC_11
0x08	Переход RUN.DCDC_11 → RUN.DCDC_09
0x09	Переход RUN.DCDC_11 → RUN.DCDC_10
0x0A	Переход RUN.DCDC_09 → RUN.HLDO_09
0x0B	Переход RUN.DCDC_10 → RUN.HLDO_10
0x0C	Переход RUN.DCDC_11 → RUN.HLDO_11
0x0D	Переход RUN.HLDO_09 → RUN.HLDO_10
0x0E	Переход RUN.HLDO_09 → RUN.HLDO_11
0x0F	Переход RUN.HLDO_10 → RUN.HLDO_09
0x10	Переход RUN.HLDO_10 → RUN.HLDO_11
0x11	Переход RUN.HLDO_11 → RUN.HLDO_09
0x12	Переход RUN.HLDO_11 → RUN.HLDO_10

Код последовательности	Описание
0x13	Переход RUN.HLDO_09 → RUN.DCDC_09
0x14	Переход RUN.HLDO_10 → RUN.DCDC_10
0x15	Переход RUN.HLDO_11 → RUN.DCDC_11

Таблица 1.177 – Подключение линий WIU

Линии WIU	Аппаратное подключение	Описание
WIU[15:8]	EXTINT_AON[7:0]	Пробуждение по внешним выводам с функцией WKP_x, смультиплексированным в контроллере EXTINT_AON (см. 1.3.10.5.3)
WIU[7]	DPSLV.CxxxPWRUPREQ	Пробуждение по запросу от отладчика
WIU[6]	LSEFAIL	Пробуждение при отказе осциллятора LSE
WIU[5]	RTC/TAMP	Пробуждение от RTC или TAMP
WIU[4]	IWDT/NRST	Пробуждение от IWDT или по низкому уровню на выводе NRST
WIU[3:1]	-	Данные прерывания устанавливаются только программно
WIU[0]	CPU.SLEEPDEEP	Вход в режимы пониженного энергопотребления DEEPSLEEPx, SHUTDOWN по запросу от CPU

Рекомендованный сценарий использования WIU:

– WIU[15:4] используются для пробуждения из режимов низкого энергопотребления по аппаратным событиями, и всегда должны быть настроены на последовательность DEEPSLEEPx, SHUTDOWN → RUN.HLDO_11 (0x0);

– WIU[3:1] могут использоваться для программных переходов между RUN-режимами путем записи битов WIU_IFR[3:1], регистры WIU_ICR_1..WIU_ICR_3 при этом должны содержать код одной из последовательностей перехода между RUN-режимами;

– WIU[0] используется для следующих сценариев:

– Программные переходы между RUN-режимами путем записи бита WIU_IFR[0], регистр WIU_ICR_0 при этом должен содержать код одной из последовательностей перехода между RUN-режимами;

– Аппаратный переход по сигналу SLEEPDEEP от CPU в режимы пониженного энергопотребления DEEPSLEEPx, SHUTDOWN. Регистр WIU_ICR_0 при этом должен содержать соответствующий код последовательности перехода в DEEPSLEEPx, SHUTDOWN.

1.3.8.3.2.3 Прерывания PMU

В таблице 1.178 приведен список событий прерывания, формируемых блоком PMU.

Таблица 1.178 – Прерывания блока PMU

Вектор NVIC	Статусный бит в регистре DLCPD_IFR	Описание	Бит маски в регистре DLCPD_IMR	Способ сброса
PMU (№27)	ICU_OK_F	Успешная смена режима ICUn	ICU_OK_M	Запись 1 в биты регистра DLCPD_IOK_IFR
	ICU_DELAYED_F	Задержка смены режима ICUn	ICU_DELAYED_M	Запись 1 в биты регистра DLCPD_IDL_IFR
	ICU_DENIED_F	Отказ смены режима ICUn	ICU_DENIED_M	Запись 1 в биты регистра DLCPD_IDN_IFR
	PICL_OK_F	Завершение транзакции PICL по запросу от DLC	PICL_OK_M	Запись 1 в бит PICL_OK_F
	SCU_OK_F	Успешное завершение последовательности SCU	SCU_OK_M	Запись 1 в бит SCU_OK_F
	SCU_FL_F	Завершение последовательности SCU с ошибкой	SCU_FL_M	Запись 1 в бит SCU_FL_F
	CGU_OK_F	Успешное завершение калибровки CGU	CGU_OK_M	Запись 1 в бит CGU_OK_F
	CGU_FL_F	Завершение калибровки CGU с ошибкой	CGU_FL_M	Запись 1 в бит CGU_FL_F

1.3.8.3.2.4 Карта регистров PMU

1.3.8.3.2.4.1 Регистры с непосредственным доступом DLCPD

В таблице 1.179 представлен список программно-доступных регистров с непосредственным доступом, в таблицах 1.180 - 1.188 - описание их полей.

Таблица 1.179 – Список программно-доступных регистров PMU с непосредственным доступом

Адрес	Название	Описание
0x00	DLCPD_MSR	Регистр статуса
0x04	DLCPD_MPACR	Регистр настройки доступа по шине PICL
0x08	DLCPD_MPADR	Регистр данных доступа по шине PICL
0x0C	DLCPD_IMR	Регистр маски прерываний
0x10	DLCPD_IFR	Регистр флагов прерываний
0x14	DLCPD_IOK_IFR	Регистр флагов прерываний ICU_OK
0x18	DLCPD_IDL_IFR	Регистр флагов прерываний ICU_DELAYED
0x1C	DLCPD_IDN_IFR	Регистр флагов прерываний ICU_DENIED
0x20	DLCPD_IUP_IFR	Регистр флагов прерываний ICU_UPDATED

1.3.8.3.2.4.1.1 Регистр статуса (DLCPD_MSR)

Смещение адреса: 0x00.

Значение по сбросу: N/A.

Таблица 1.180 – Описание полей регистра DLCPD_MSR

Биты	Название	Доступ	Значение по сбросу	Описание
[31:5]	—	R0	0x0	*резерв*

Биты	Название	Доступ	Значение по сбросу	Описание
[4]	CGU_BUSY	R	-	Признак активности калибровки CGU: 0x0 - калибровка CGU не выполняется; 0x1 - калибровка CGU в процессе выполнения;
[3:2]	—	R0	0x0	*резерв*
[1]	SCU_BUSY	R	-	Признак активности блока SCU: 0x0 - нет активной последовательности SCU; 0x1 - SCU выполняет последовательность.
[0]	PICL_BUSY	R	-	Признак активности шины PICL: 0x0 - нет активной транзакции на шине PICL; 0x1 - выполняется транзакция на шине PICL.

1.3.8.3.2.4.1.2 Регистр настройки доступа по шине PICL (DLCPD_MPACR)

Смещение адреса: 0x04.

Значение по сбросу: 0x0000_0000.

Таблица 1.181 – Описание полей регистра DLCPD_MPACR

Биты	Название	Доступ	Значение по сбросу	Описание
[31:29]	—	R0	0x0	*резерв*
[28]	PAS	R/W	0x0	Запуск транзакции по шине PICL. Установка этого бита инициирует обращение по шине PICL. Бит сбрасывается автоматически после завершения транзакции.
[27:25]	—	R0	0x0	*резерв*
[24]	PAD	R/W	0x0	Тип доступа по шине PICL: 0x0 - запись; 0x1 - чтение;
[23:16]	—	R0	0x0	*резерв*
[15:0]	PAA	R/W	0x0	Адрес доступа PICL

1.3.8.3.2.4.1.3 Регистр данных доступа по шине PICL (DLCPD_MPADR)

Смещение адреса: 0x08.

Значение по сбросу: 0x0000_0000.

Таблица 1.182 – Описание полей регистра DLCPD_MPADR

Биты	Название	Доступ	Значение по сбросу	Описание
[31:16]	—	R0	0x0	*резерв*
[15:0]	PRWDATA	R/W	0x0	Данные чтения/записи шины PICL: Данное поле автоматически обновляется после завершения транзакции чтения.

1.3.8.3.2.4.1.4 Регистр маски прерываний (DLCPD_IMR)

Смещение адреса: 0x0C.

Значение по сбросу: 0x3000_063F.

Таблица 1.183 – Описание полей регистра DLCPD_IMR

Биты	Название	Доступ	Значение по сбросу	Описание
[31:13]	-	R/W	0x300	*резерв*
[12]	CGU_FL_M	R/W	0x0	Маска прерывания CGU_FL.
[11]	CGU_OK_M	R/W	0x0	Маска прерывания CGU_OK.
[10:9]	—	R/W	0x3	*резерв*
[8]	SCU_FL_M	R/W	0x0	Маска прерывания SCU_FL.
[7]	SCU_OK_M	R/W	0x0	Маска прерывания SCU_OK.
[6]	PICL_OK_M	R/W	0x0	Маска прерывания PICL_OK.
[5:4]	—	R/W	0x3	*резерв*
[3]	ICU_UPDATED_M	R/W	0x1	Маска прерывания ICU_UPDATED.
[2]	ICU_DENIED_M	R/W	0x1	Маска прерывания ICU_DENIED.
[1]	ICU_DELAYED_M	R/W	0x1	Маска прерывания ICU_DELAYED.
[0]	ICU_OK_M	R/W	0x1	Маска прерывания ICU_OK.

1.3.8.3.2.4.1.5 Регистр статуса прерываний (DLCPD_IFR)

Смещение адреса: 0x10.

Значение по сбросу: N/A.

Таблица 1.184 – Описание полей регистра DLCPD_IFR

Биты	Название	Доступ	Значение по сбросу	Описание
[31:13]	-	R	-	*резерв*
[12]	CGU_FL_F	R/W	-	Данный бит устанавливается при завершении калибровки CGU с ошибкой. Бит сбрасывается записью 1.
[11]	CGU_OK_F	R/W	-	Данный бит устанавливается при завершении калибровки CGU без ошибок. Бит сбрасывается записью 1.
[10:9]	—	R/W	-	*резерв*
[8]	SCU_FL_F	R/W	-	Данный бит устанавливается при завершении последовательности SCU с ошибкой. Бит сбрасывается записью 1.
[7]	SCU_OK_F	R/W	-	Данный бит устанавливается при завершении последовательности SCU без ошибок. Бит сбрасывается записью 1.
[6]	PICL_OK_F	R/W	-	Данный бит устанавливается при завершении транзакции PICL. Бит сбрасывается записью 1.
[5:4]	—	R/W	-	*резерв*
[3]	ICU_UPDATED_F	R/W	-	Данный бит устанавливается если установлен хотя бы один бит в регистре DLCPD_IUP_IFR.
[2]	ICU_DENIED_F	R/W	-	Данный бит устанавливается если установлен хотя бы один бит в регистре DLCPD_IDN_IFR.
[1]	ICU_DELAYED_F	R/W	-	Данный бит устанавливается если установлен хотя бы один бит в регистре DLCPD_IDL_IFR.
[0]	ICU_OK_F	R/W	-	Данный бит устанавливается если установлен хотя бы один бит в регистре DLCPD_IOK_IFR.

1.3.8.3.2.4.1.6 Регистр флагов прерываний ICU_OK (DLCPD_IOK_IFR)

Смещение адреса: 0x14.

Значение по сбросу: N/A.

Таблица 1.185 – Описание полей регистра DLCPD_IOK_IFR

Биты	Название	Доступ	Значение по сбросу	Описание
[31:8]	-	R0	-	*резерв*
[7:0]	ICU_OK_FLAGS	R/W	-	Флаги прерывания ICU_OK. Каждый бит данного поля устанавливается в 1 если соответствующий ICU выполнил смену режима по запросу от управляющего интерфейса. Биты можно сбросить записью 1.

1.3.8.3.2.4.1.7 Регистр флагов прерываний ICU_DELAYED (DLCPD_IDL_IFR)

Смещение адреса: 0x18.

Значение по сбросу: N/A.

Таблица 1.186 – Описание полей регистра DLCPD_IDL_IFR

Биты	Название	Доступ	Значение по сбросу	Описание
[31:8]	-	R0	-	*резерв*
[7:0]	ICU_DELAYED_FLAGS	R/W	-	Флаги прерывания ICU_DELAYED. Каждый бит данного поля устанавливается в 1 если соответствующий ICU отложил выполнение смены режима. Биты можно сбросить записью 1.

1.3.8.3.2.4.1.8 Регистр флагов прерываний ICU_DENIED (DLCPD_IDN_IFR)

Смещение адреса: 0x1C.

Значение по сбросу: N/A.

Таблица 1.187 – Описание полей регистра DLCPD_IDN_IFR

Биты	Название	Доступ	Значение по сбросу	Описание
[31:8]	-	R0	-	*резерв*
[7:0]	ICU_DENIED_FLAGS	R/W	-	Флаги прерывания ICU_DENIED. Каждый бит данного поля устанавливается в 1 если соответствующий ICU отказал в выполнении смены режима. Биты можно сбросить записью 1.

1.3.8.3.2.4.1.9 Регистр флагов прерываний ICU_UPDATED (DLCPD_IUP_IFR)

Смещение адреса: 0x20.

Значение по сбросу: N/A.

Таблица 1.188 – Описание полей регистра DLCPD_IUP_IFR

Биты	Название	Доступ	Значение по сбросу	Описание
[31:8]	-	R0	-	*резерв*

Биты	Название	Доступ	Значение по сбросу	Описание
[7:0]	ICU_UPDATED_FLAGS	R/W	-	Флаги прерывания ICU_UPDATED. Каждый бит данного поля устанавливается в 1 если соответствующий ICU сменил режим или уровень. Биты можно сбросить записью 1.

1.3.8.3.2.4.2 Регистры с косвенным доступом

1.3.8.3.2.4.2.1 Адресация регистров устройств на шине PICL

Косвенный доступ используется для регистров устройств на шине PICL - WIU, SCU, CGU, блоки ICU и DMU. Для доступа к этим регистрам используется 16-битный адрес, разбитый на два поля:

1) BLOCK_ADDR(DLCPD_MPACR.PAA[15:8]): 8-битное смещение базового адреса устройства на шине PICL;

2) REG_ADDR(DLCPD_MPACR.PAA[7:0]): 8-битное смещение регистра относительно базового адреса устройства;

В таблице 1.189 представлены базовые адреса устройств на шине PICL.

Таблица 1.189 – Адреса устройств на шине PICL

Устройство	Адрес (BLOCK_ADDR)
WIU	0x01
ICU0 (управление регулятором BLDO)	0x10
ICU1 (управление доменом PD_MAIN)	0x11
ICU2 (управление доменом PD_ETH)	0x12
ICU3 (управление доменом PD_USB)	0x13
ICU4 (управление доменом PD_EMMC0)	0x14
ICU5 (управление доменом PD_EMMC1)	0x15
ICU6 (управление регулятором HLDO)	0x16
ICU7 (управление регулятором DCDC)	0x17

1.3.8.3.2.4.2.2 Регистры WIU

В таблице 1.190 представлен список программно-доступных регистров WIU.

Таблица 1.190 – Список программно-доступных регистров WIU

Адрес	Название	Описание	Значение по сбросу
0x00	WIU_ISPMR	Регистр маски прерываний	0x0000
0x02	WIU_IFR	Регистр флагов прерываний	0x0000
0x04	WIU_ICR_0	Регистр конфигурации прерывания WIU[0]	0x0002
0x06	WIU_ICR_1	Регистр конфигурации прерывания WIU[1]	0x0000
0x08	WIU_ICR_2	Регистр конфигурации прерывания WIU[2]	0x0000
0x0A	WIU_ICR_3	Регистр конфигурации прерывания WIU[3]	0x0000

Адрес	Название	Описание	Значение по сбросу
0x0C	WIU_ICR_4	Регистр конфигурации прерывания WIU[4]	0x0000
0x0E	WIU_ICR_5	Регистр конфигурации прерывания WIU[5]	0x0000
0x10	WIU_ICR_6	Регистр конфигурации прерывания WIU[6]	0x0000
0x12	WIU_ICR_7	Регистр конфигурации прерывания WIU[7]	0x0000
0x14	WIU_ICR_8	Регистр конфигурации прерывания WIU[8]	0x0000
0x16	WIU_ICR_9	Регистр конфигурации прерывания WIU[9]	0x0000
0x18	WIU_ICR_10	Регистр конфигурации прерывания WIU[10]	0x0000
0x1A	WIU_ICR_11	Регистр конфигурации прерывания WIU[11]	0x0000
0x1C	WIU_ICR_12	Регистр конфигурации прерывания WIU[12]	0x0000
0x1E	WIU_ICR_13	Регистр конфигурации прерывания WIU[13]	0x0000
0x20	WIU_ICR_14	Регистр конфигурации прерывания WIU[14]	0x0000
0x22	WIU_ICR_15	Регистр конфигурации прерывания WIU[15]	0x0000
0x24	WIU_IER	Регистр признаков завершенных прерываний	0x0000

Регистр маски прерываний (WIU_ISPMR)

Смещение адреса: 0x00.

Значение по сбросу: 0x0000.

Таблица 1.191 – Описание полей регистра WIU_ISPMR

Биты	Название	Доступ	Значение по сбросу	Описание
[15:0]	WIU_IRQ_M	R/W	0x0	Биты маскирования соответствующих прерываний WIU, маскирование осуществляется при значении 1.

Регистр флагов прерываний WIU (WIU_IFR)

Смещение адреса: 0x02.

Значение по сбросу: 0x0000.

Таблица 1.192 – Описание полей регистра WIU_IFR

Биты	Название	Доступ	Значение по сбросу	Описание
[15:0]	WIU_IRQ_F	R/W	0x0	Флаги прерываний WIU. Каждый бит устанавливается в 1 программной записью или при обнаружении положительного фронта на соответствующей линии WIU_IRQ. Биты устанавливаются в 0 после завершения соответствующей последовательности SCU либо если соответствующий бит маски в регистре WIU_ISPMR равен 0.

Регистр конфигурации прерывания WIU[0] (WIU_ICR_0)

Смещение адреса: 0x04.

Значение по сбросу: 0x0002.

Таблица 1.193 – Описание полей регистра WIU_ICR_0

Биты	Название	Доступ	Значение по сбросу	Описание
[15:5]	-	R0	0x0	*резерв*
[4:0]	SEQ_SEL_IRQ	R/W	0x2	Идентификатор последовательности, запускаемой по прерыванию WIU[0]: 0x1 - переход в режим DEEPSLEEP0; 0x2 - переход в режим DEEPSLEEP1; 0x3 - переход в режим SHUTDOWN.

Регистр конфигурации прерываний WIU[3:1] (WIU_ICR_1 .. WIU_ICR_3)

Смещение адреса: 0x04 + 2 * n, где n = [1..3] - номер линии WIU.

Значение по сбросу: 0x0000.

Описание полей регистров WIU_ICR_1..WIU_ICR_3

Биты	Название	Доступ	Значение по сбросу	Описание
[15:5]	-	R0	0x0	*резерв*
[4:0]	SEQ_SEL_IRQ	R/W	0x0	Идентификатор последовательности, запускаемой по прерыванию WIU[n]

Регистры конфигурации прерываний WIU[15:4] (WIU_ICR_4 .. WIU_ICR_15)

Смещение адреса: 0x04 + 2 * n, где n = [4..15] - номер линии WIU.

Значение по сбросу: 0x0000.

Описание полей регистров WIU_ICR_4..WIU_ICR_15

Биты	Название	Доступ	Значение по сбросу	Описание
[15:5]	-	R0	0x0	*резерв*
[4:0]	SEQ_SEL_IRQ	R/W	0x0	Идентификатор последовательности, запускаемой по прерыванию WIU[n]: 0x0 - переход в режим RUN.HLDO_11 из режимов низкого энергопотребления DEEPSLEEPx, SHUTDOWN

1.3.8.3.2.4.2.3 Регистры ICU0 (управление регулятором BLDO)

В таблице 1.194 представлен список программно-доступных регистров ICU0.

Таблица 1.194 – Список программно-доступных регистров ICU0

Адрес	Название	Описание
0x00	ICU0_MT_SR	Регистр статуса режима ICU0
0x02	ICU0_MT_CR	Регистр управления режимом ICU0

Регистр статуса режима ICU0 (ICU0_MT_SR)

Смещение адреса: 0x00.

Значение по сбросу: N/A.

Таблица 1.195 – Описание полей регистра ICU0_MT_SR

Биты	Название	Доступ	Значение по сбросу	Описание
[15:5]	-	R	-	*резерв*
[4]	ICUMTS	R	-	Статусный бит режима ICU0: 0x0 - текущий режим ICU0 отражается в поле CICUM; 0x1 - текущий режим ICU0 не определен в таблице режимов.
[3:0]	CICUM	R	-	Поле возвращает текущий режим ICU0 когда бит ICUMTS равен нулю.

Регистр управления режимом ICU0 (ICU0_MT_CR)

Смещение адреса: 0x02.

Значение по сбросу: 0x0002.

Таблица 1.196 – Описание полей регистра ICU0_MT_CR

Биты	Название	Доступ	Значение по сбросу	Описание
[15:4]	-	R0	0x0	*резерв*
[3:0]	ICUMTR	R/W	0x2	Задание режима ICU0 (регулятор BLDO): 0x0 - выходное напряжение BLDO равно 0.9 В; 0x1 - выходное напряжение BLDO равно 1.0 В; 0x2 - выходное напряжение BLDO равно 1.1 В. Остальные значения зарезервированы.

1.3.8.3.2.4.2.4 Регистры ICU1 (управление доменом PD_MAIN)

В таблице 1.197 представлен список программно-доступных регистров ICU1.

Таблица 1.197 – Список программно-доступных регистров ICU1

Адрес	Название	Описание
0x00	ICU1_MT_SR	Регистр статуса режима ICU1
0x02	ICU1_MT_CR	Регистр управления режимом ICU1

Регистр статуса режима ICU1 (ICU1_MT_SR)

Смещение адреса: 0x00.

Значение по сбросу: N/A.

Таблица 1.198 – Описание полей регистра ICU1_MT_SR

Биты	Название	Доступ	Значение по сбросу	Описание
[15:5]	-	R	-	*резерв*
[4]	ICUMTS	R	-	Статусный бит режима ICU1: 0x0 - текущий режим ICU1 отражается в поле CICUM; 0x1 - текущий режим ICU1 не определен в таблице режимов.
[3:0]	CICUM	R	-	Поле возвращает текущий режим ICU1 когда бит ICUMTS равен нулю.

Регистр управления режимом ICU1 (ICU1_MT_CR)

Смещение адреса: 0x02.

Значение по сбросу: 0x0000.

Таблица 1.199 – Описание полей регистра ICU1_MT_CR

Биты	Название	Доступ	Значение по сбросу	Описание
[15:4]	-	R0	0x0	*резерв*
[3:0]	ICUMTR	R/W	0x0	Задание режима ICU1 (домен PD_MAIN): 0x0 - синхросигналы в домене PD_MAIN включены; 0x1 - синхросигналы в домене PD_MAIN остановлены. Остальные значения зарезервированы.

1.3.8.3.2.4.2.5 Регистры ICU2 (управление доменом PD_ETH)

В таблице 1.200 представлен список программно-доступных регистров ICU2.

Таблица 1.200 – Список программно-доступных регистров ICU2

Адрес	Название	Описание
0x00	ICU2_MT_SR	Регистр статуса режима ICU2
0x02	ICU2_MT_CR	Регистр управления режимом ICU2

Регистр статуса режима ICU2 (ICU2_MT_SR)

Смещение адреса: 0x00.

Значение по сбросу: N/A.

Таблица 1.201 – Описание полей регистра ICU2_MT_SR

Биты	Название	Доступ	Значение по сбросу	Описание
[15:5]	-	R	-	*резерв*
[4]	ICUMTS	R	-	Статусный бит режима ICU2: 0x0 - текущий режим ICU2 отражается в поле CICUM; 0x1 - текущий режим ICU2 не определен в таблице режимов.
[3:0]	CICUM	R	-	Поле возвращает текущий режим ICU2 когда бит ICUMTS равен нулю.

Регистр управления режимом ICU2 (ICU2_MT_CR)

Смещение адреса: 0x02.

Значение по сбросу: 0x0000.

Таблица 1.202 – Описание полей регистра ICU2_MT_CR

Биты	Название	Доступ	Значение по сбросу	Описание
[15:4]	-	R0	0x0	*резерв*
[3:0]	ICUMTR	R/W	0x0	Задание режима ICU2 (домен PD_ETH): 0x0 - питание домена PD_ETH включено; 0x1 - питание домена PD_ETH выключено; Остальные значения зарезервированы.

1.3.8.3.2.4.2.6 Регистры ICU3 (управление доменом PD_USB)

В таблице 1.203 представлен список программно-доступных регистров ICU3.

Таблица 1.203 – Список программно-доступных регистров ICU3

Адрес	Название	Описание
0x00	ICU3_MT_SR	Регистр статуса режима ICU3
0x02	ICU3_MT_CR	Регистр управления режимом ICU3

Регистр статуса режима ICU3 (ICU3_MT_SR)

Смещение адреса: 0x00.

Значение по сбросу: N/A.

Таблица 1.204 – Описание полей регистра ICU3_MT_SR

Биты	Название	Доступ	Значение по сбросу	Описание
[15:5]	-	R	-	*резерв*
[4]	ICUMTS	R	-	Статусный бит режима ICU3: 0x0 - текущий режим ICU3 отражается в поле CICUM; 0x1 - текущий режим ICU3 не определен в таблице режимов.
[3:0]	CICUM	R	-	Поле возвращает текущий режим ICU3 когда бит ICUMTS равен нулю.

Регистр управления режимом ICU3 (ICU3_MT_CR)

Смещение адреса: 0x02.

Значение по сбросу: 0x0000.

Таблица 1.205 – Описание полей регистра ICU3_MT_CR

Биты	Название	Доступ	Значение по сбросу	Описание
[15:4]	-	R0	0x0	*резерв*
[3:0]	ICUMTR	R/W	0x0	Задание режима ICU3 (домен PD_USB): 0x0 - питание домена PD_USB включено; 0x1 - питание домена PD_USB выключено; Остальные значения зарезервированы.

1.3.8.3.2.4.2.7 Регистры ICU4 (управление доменом PD_EMMC0)

В таблице 1.206 представлен список программно-доступных регистров ICU4.

Таблица 1.206 – Список программно-доступных регистров ICU4

Адрес	Название	Описание
0x00	ICU4_MT_SR	Регистр статуса режима ICU4
0x02	ICU4_MT_CR	Регистр управления режимом ICU4

Регистр статуса режима ICU4 (ICU4_MT_SR)

Смещение адреса: 0x00.

Значение по сбросу: N/A.

Таблица 1.207 – Описание полей регистра ICU4_MT_SR

Биты	Название	Доступ	Значение по сбросу	Описание
[15:5]	-	R	-	*резерв*
[4]	ICUMTS	R	-	Статусный бит режима ICU4: 0x0 - текущий режим ICU4 отражается в поле CICUM; 0x1 - текущий режим ICU4 не определен в таблице режимов.
[3:0]	CICUM	R	-	Поле возвращает текущий режим ICU4 когда бит ICUMTS равен нулю.

Регистр управления режимом ICU4 (ICU4_MT_CR)

Смещение адреса: 0x02.

Значение по сбросу: 0x0000.

Таблица 1.208 – Описание полей регистра ICU4_MT_CR

Биты	Название	Доступ	Значение по сбросу	Описание
[15:4]	-	R0	0x0	*резерв*
[3:0]	ICUMTR	R/W	0x0	Задание режима ICU4 (домен PD_EMMC0): 0x0 - питание домена PD_EMMC0 включено; 0x1 - питание домена PD_EMMC0 выключено; Остальные значения зарезервированы.

1.3.8.3.2.4.2.8 Регистры ICU5 (управление доменом PD_EMMC1)

В таблице 1.209 представлен список программно-доступных регистров ICU5.

Таблица 1.209 – Список программно-доступных регистров ICU5

Адрес	Название	Описание
0x00	ICU5_MT_SR	Регистр статуса режима ICU5
0x02	ICU5_MT_CR	Регистр управления режимом ICU5

Регистр статуса режима ICU5 (ICU5_MT_SR)

Смещение адреса: 0x00.

Значение по сбросу: N/A.

Таблица 1.210 – Описание полей регистра ICU5_MT_SR

Биты	Название	Доступ	Значение по сбросу	Описание
[15:5]	-	R	-	*резерв*
[4]	ICUMTS	R	-	Статусный бит режима ICU5: 0x0 - текущий режим ICU5 отражается в поле CICUM; 0x1 - текущий режим ICU5 не определен в таблице режимов.

Биты	Название	Доступ	Значение по сбросу	Описание
[3:0]	CICUM	R	-	Поле возвращает текущий режим ICU5 когда бит ICUMTS равен нулю.

Регистр управления режимом ICU5 (ICU5_MT_CR)

Смещение адреса: 0x02.

Значение по сбросу: 0x0000.

Таблица 1.211 – Описание полей регистра ICU5_MT_CR

Биты	Название	Доступ	Значение по сбросу	Описание
[15:4]	-	R0	0x0	*резерв*
[3:0]	ICUMTR	R/W	0x0	Задание режима ICU5 (домен PD_EMMC1): 0x0 - питание домена PD_EMMC1 включено; 0x1 - питание домена PD_EMMC1 выключено; Остальные значения зарезервированы.

1.3.8.3.2.4.2.9 Регистры ICU6 (управление регулятором HLDO)

В таблице 1.212 представлен список программно-доступных регистров ICU6.

Таблица 1.212 – Список программно-доступных регистров ICU6

Адрес	Название	Описание
0x00	ICU6_MT_SR	Регистр статуса режима ICU6
0x02	ICU6_MT_CR	Регистр управления режимом ICU6

Регистр статуса режима ICU6 (ICU6_MT_SR)

Смещение адреса: 0x00.

Значение по сбросу: N/A.

Таблица 1.213 – Описание полей регистра ICU6_MT_SR

Биты	Название	Доступ	Значение по сбросу	Описание
[15:5]	-	R	-	*резерв*
[4]	ICUMTS	R	-	Статусный бит режима ICU6: 0x0 - текущий режим ICU6 отражается в поле CICUM; 0x1 - текущий режим ICU6 не определен в таблице режимов.
[3:0]	CICUM	R	-	Поле возвращает текущий режим ICU6 когда бит ICUMTS равен нулю.

Регистр управления режимом ICU6 (ICU6_MT_CR)

Смещение адреса: 0x02.

Значение по сбросу: 0x0002.

Таблица 1.214 – Описание полей регистра ICU6_MT_CR

Биты	Название	Доступ	Значение по сбросу	Описание
[15:4]	-	R0	0x0	*резерв*
[3:0]	ICUMTR	R/W	0x2	Задание режима ICU6 (регулятор HLDO): 0x0 - выходное напряжение HLDO равно 0.9 В; 0x1 - выходное напряжение HLDO равно 1.0 В; 0x2 - выходное напряжение HLDO равно 1.1 В; 0x3 - регулятор HLDO выключен, выход в высокоимпедансном состоянии (Hi-Z); 0x4 - регулятор HLDO выключен, выход подтянут к земле (Pull-down). Остальные значения зарезервированы.

1.3.8.3.2.4.2.10 Регистры ICU7 (управление регулятором DCDC)

В таблице 1.215 представлен список программно-доступных регистров ICU7.

Таблица 1.215 – Список программно-доступных регистров ICU7

Адрес	Название	Описание
0x00	ICU7_MT_SR	Регистр статуса режима ICU7
0x02	ICU7_MT_CR	Регистр управления режимом ICU7

Регистр статуса режима ICU7 (ICU7_MT_SR)

Смещение адреса: 0x00.

Значение по сбросу: N/A.

Таблица 1.216 – Описание полей регистра ICU7_MT_SR

Биты	Название	Доступ	Значение по сбросу	Описание
[15:5]	-	R	-	*резерв*
[4]	ICUMTS	R	-	Статусный бит режима ICU7: 0x0 - текущий режим ICU7 отражается в поле CICUM; 0x1 - текущий режим ICU7 не определен в таблице режимов.
[3:0]	CICUM	R	-	Поле возвращает текущий режим ICU7 когда бит ICUMTS равен нулю.

Регистр управления режимом ICU7 (ICU7_MT_CR)

Смещение адреса: 0x02.

Значение по сбросу: 0x0000.

Таблица 1.217 – Описание полей регистра ICU7_MT_CR

Биты	Название	Доступ	Значение по сбросу	Описание
[15:4]	-	R0	0x0	*резерв*

Биты	Название	Доступ	Значение по сбросу	Описание
[3:0]	ICUMTR	R/W	0x0	Задание режима ICU7 (регулятор DCDC): 0x0 - регулятор DCDC выключен, выход в высокоимпедансном состоянии (Hi-Z); 0x1 - выходное напряжение DCDC равно 0.9 В; 0x2 - выходное напряжение DCDC равно 1.0 В; 0x3 - выходное напряжение DCDC равно 1.1 В; 0x4 - регулятор DCDC выключен, выход подтянут к земле (Pull-down). Остальные значения зарезервированы.

1.3.9 Цифровые интерфейсы

1.3.9.1 Контроллер интерфейса GPIO (GPIO)

1.3.9.1.1 Основные характеристики контроллера GPIO

Контроллер GPIO обладает следующими характеристиками:

– число линий у одного контроллера до 32;

– независимая настройка каждого буфера ввода/вывода:

– направление (вход или выход);

– подтяжка к 0, к 1 или без подтяжки (floating);

– включение/выключение триггера Шмитта в режиме входа;

– в режиме выхода доступны следующие настройки:

– выход push-pull или open-drain (выход с открытым стоком);

– управление выходным током (output drive strengths);

– управление скорость изменения сигнала (slew-rate)

– режимы работы:

– GPIO:

– задание значений всех выходов GPIO;

– независимое управление каждой выходной линией:

– установка 1;

– установка 0;

– возможность настройки задержка установки значения на выходе в половину такта или в такт опорной частоты (или без задержки);

– максимальная скорость переключения выходов - 2 такта опорной частоты;

– асинхронная развязка на входе;

– аналоговый;

– LCD;

– до 15 альтернативных функций и блок IFMUX, который позволяет гибко мультиплексировать линии подключенных к нему контроллеров;

– поддержка TrustZone:

– защищённый доступ к регистрам;

– защищённый доступ к буферам ввода/вывода;

– блокировка настройки отдельных буферов GPIO, которая снимается только аппаратным сбросом;

– интерфейс программирования APB slave.

Если буфер находится в защищённом режиме работы, то доступ к нему возможен только в защищённом режиме. В случае работы контроллера GPIO это означает, что доступ к соответствующим регистрам возможен только в защищённом режиме. В случае работы альтернативной функции доступ к буферам предоставляется только устройствам, находящимся в защищённом режиме работы.

1.3.9.1.2 Описание работы контроллера GPIO

1.3.9.1.2.1 Структурная схема контроллера GPIO

Структурная схема контроллера GPIO приведена на рисунке 1.33.

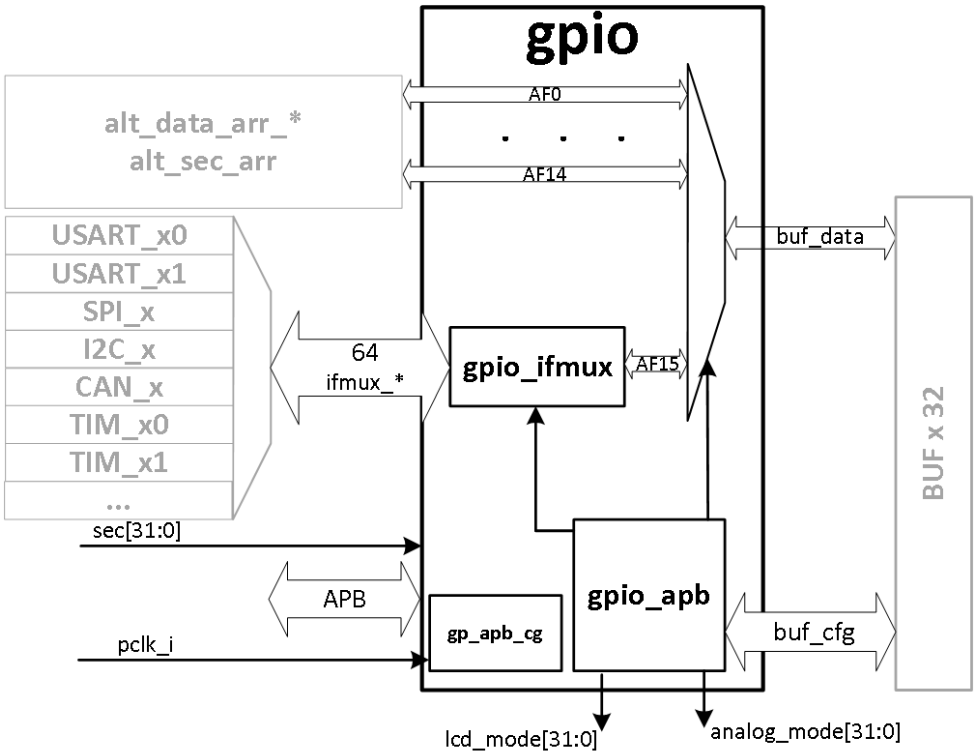


Рисунок 1.33 – Структурная схема контроллера GPIO

Описание блоков структурной схемы контроллера GPIO приведено в таблице 1.218.

Таблица 1.218 – Описание блоков структурной схемы контроллера GPIO

Блок	Описание
gpio	Сборка поблоков, верхний уровень контроллера GPIO
gpio_ifmux	Подблок для мультиплексирования сигналов от низкоскоростных интерфейсов на любой из буферов ввода-вывода
gpio_apb	Подблок для программирования GPIO
gp_apb_cg	Подблок для управления снятием тактирования с GPIO

У контроллера GPIO есть возможность снятия тактирования (режим низкого энергопотребления).

Внешняя настройка тактировани контроллера GPIO, задается битами GPIO_x_EN (GPIO_A_EN, GPIO_B_EN, GPIO_C_EN, GPIO_D_EN) в регистре RCU_PCER0 блока RCU (см. 1.3.7.7.16).

Программный доступ в любой регистр при снятом тактировании будет возвращать ошибку системной шины и вызывать исключение процессорного ядра.

1.3.9.1.2.2 Описание внешних выводов контроллера GPIO

На базе одного контроллера GPIO строится сборка контроллеров низкоскоростных интерфейсов (см. 1.3.9.7).

Каждая линия контроллера GPIO управляет одним буфером ввода/вывода. Для именования внешних выводов контроллеров GPIO в составе МК Optimal+ принята следующая схема: GPIO_<_имя_порта_>_<_номер_линии_> (например, 9-я линия контроллера GPIO порта *B* будет называться GPIO_B_9).

Каждый порт в составе МК Optimal+ использует не все линии контроллера GPIO, точный список используемых выводов для каждого порта см. в разделе 1.2.2.

1.3.9.1.2.3 Настройка режима работы

Настройка режима работы производится в регистрах GPIO_i_CFG для каждого буфера **независимо** в поле [mode].

Контроллер поддерживает следующие режимы работы: * Включение альтернативной функций (alt_mode: GPIO_i_CFG[mode] = 2'b00) * Включение альтернативной аналоговой функции (analog_mode: GPIO_i_CFG[mode] = 2'b01) * Режим работы GPIO (gpio_mode: GPIO_i_CFG[mode] = 2'b10) * Режим работы LCD (lcd_mode: GPIO_i_CFG[mode] = 2'b11)

1.3.9.1.2.4 Настройка буферов ввода/вывода

Настройка буферов ввода/вывода производится в регистрах GPIO_i_CFG для каждого буфера **независимо**.

Настройка производится только для буферов, которые есть в таблице 1.2.2.

Если буфер работает в режиме GPIO, то направление (вход или выход) задается программно в регистрах GPIO_i_CFG (или GPIO_SET_OE_0_15, GPIO_SET_OE_16_31). Если к буферу подключена аналоговая функция или LCD, то цифровой буфер рекомендуется отключается (input_enable = 0 и output_enable = 0). Если буфер работает в режиме альтернативной цифровой функции направлением работы буфера управляет блок соотвествующие это альтернативной цифровой функции.

Схема управление работой выхода в режиме open-drain приведена на рисунке 1.34.

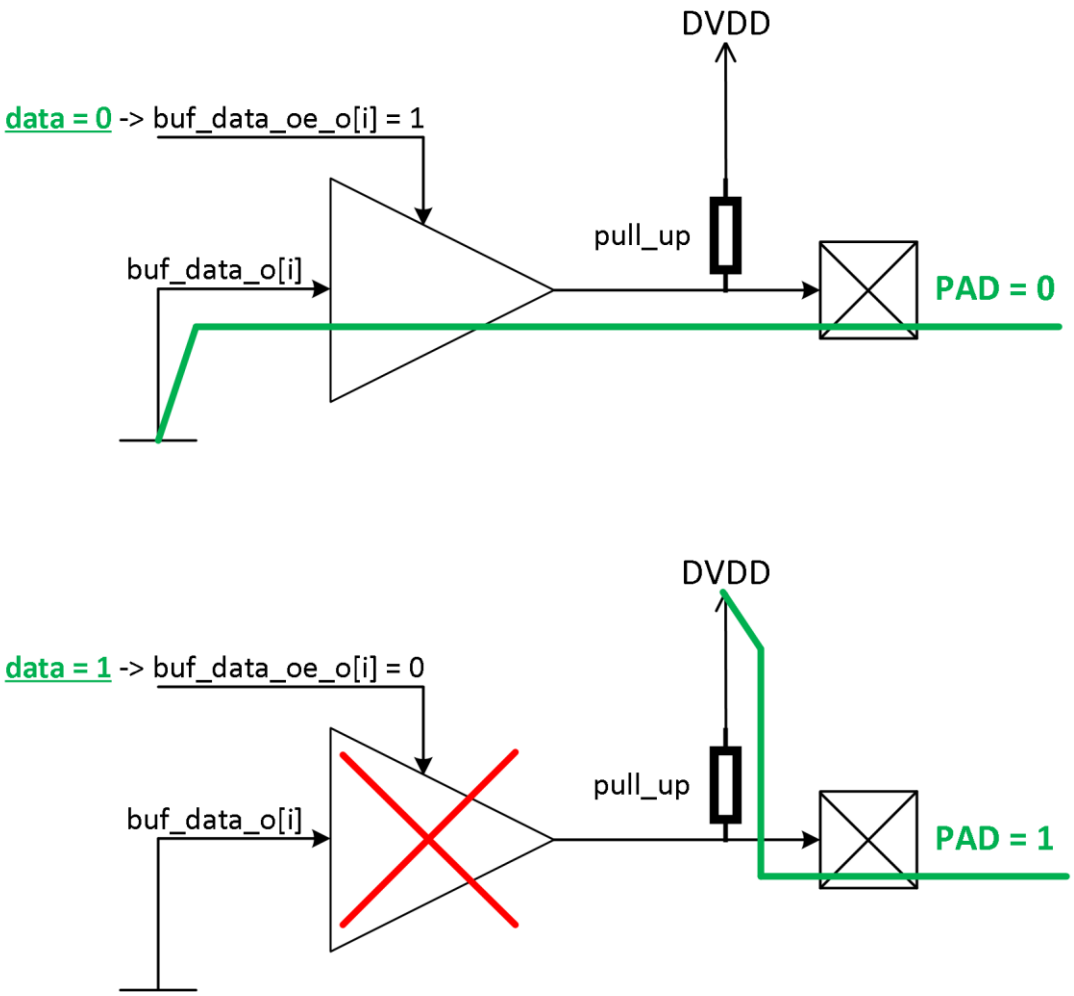


Рисунок 1.34 – Работа выхода в режиме open-drain

1.3.9.1.2.5 Альтернативные функции

Контроллер GPIO поддерживает до 15 альтернативных функций. В поле alt_func в регистре GPIO_i_CFG выбирается какая из альтернативных функций подключена к буферу при mode = 2'b00 (alt_mode). Описание мультиплексирование альтернативных функций приведено в 1.2.2.

При alt_func = 15 выбирается линия подключенная к IFMUX с номером заданным в поле ifmux_sel (см. 1.3.9.7). IFMUX позволяет гибко мультиплексировать линии, подключенных к нему контроллеров, и подключать любую линию (от 0 до 63) к любому из буферов ввода/вывода (от 0 до 31).

1.3.9.1.2.6 Задание значений выходов при работе режиме GPIO;

Контроллер поддерживает три способа задание значений выходов GPIO:

– задание всех значений выходов GPIO через регистр GPIO_OUTPUT;

– независимое управление каждой выходной линией через регистры GPIO_SET_CLR_0_15 и GPIO_SET_CLR_16_31:

– установка 1 (GPIO_SET_CLR_0_15[15:0] и GPIO_SET_CLR_16_31[15:0]);

– установка 0 (GPIO_SET_CLR_0_15[31:16] и GPIO_SET_CLR_16_31[31:16]);

– управление отдельной выходной линией через регистры GPIO_i_CFG: установка и сброс через поля регистров GPIO_i_CFG[bits_set] и GPIO_i_CFG[bits_clr] (для возможности установить нужное значение одновременно с настройкой других параметров буфера).

Управление установкой 1 и 0 специально разбито на 2 группы 0-15 и 16-31, так чтобы в рамках 32х битных регистров этот механизм позволял сделать одновременное переключение сигналов в рамках группы и из 0 в 1 и из 1 в 0.

В регистре GPIO_OUTPUT всегда хранится (оттуда можно прочитать) актуальное значение на выходах блока, даже когда управление производится через регистры GPIO_SET_CLR_0_15, GPIO_SET_CLR_16_31, GPIO_i_CFG.

1.3.9.1.2.7 Асинхронная развязка на входе блока GPIO

Асинхронная развязка на входе блока GPIO используется только при работе в mode = 2’b10 (gpio_mode), для этого используется ячейка синхронизации из 2х триггеров.

1.3.9.1.2.8 Поддержка TrustZone:

В контроллере GPIO к регистрам GPIO_WP_ACCESS и GPIO_BUF_SECURE поддерживается только защищённый доступ.

Доступ к регистрам GPIO_CFG_WRITE_PROTECT, GPIO_LOCK_WRITE_PROTECT зависит от настроек, заданных в регистре GPIO_WP_ACCESS.

Доступ к регистрам GPIO_INPUT, GPIO_OUTPUT, GPIO_SET_CLR_0_15, GPIO_SET_CLR_16_31, GPIO_i_CFG, GPIO_LOCK_CFG или к части их бит зависит от настроек, заданных в регистре GPIO_BUF_SECURE.

Доступ к линиям всех альтернативных функций (alt_data_*, ifmux_data_*, analog_mode_o) зависит от совпадения значения доступа, заданного в линии и в регистре GPIO_BUF_SECURE.

1.3.9.1.2.9 Блокировка настройки

В контроллере GPIO кроме TrustZone поддерживается еще блокировки настройки:

– блокировка от случайного изменения настройки буферов GPIO (регистров GPIO_i_CFG), которая задается в регистре GPIO_CFG_WRITE_PROTECT;

– блокировка от изменения настройки буферов GPIO до аппаратного сброса, которая задается в регистре GPIO_LOCK_CFG. При этом биты регистра GPIO_LOCK_CFG защищены от случайного изменения, эта защита задается в регистре GPIO_LOCK_WRITE_PROTECT.

1.3.9.1.3 Описание регистров контроллера GPIO

1.3.9.1.3.1 Карта регистров контроллера GPIO

Задание конфигурационных параметров и отображение текущего состояния контроллера GPIO ведется через регистры. Доступ к регистрам осуществляется посредством системного интерфейса управления (интерфейс APB 4/5 slave). Каждое обращение к регистрам имеет выравнивание до 32-х разрядов. Карта регистров приведена в таблице 1.219.

Часть регистров контроллера GPIO доступны только в защищённом режиме работы.

В случае перевод буфера ввода/вывода в защищённый режим работы, регистры, соответствующие этому буферу, требуют доступа в защищённом режиме.

При обращении к регистрам СФ-блока по зарезервированным адресам никаких ошибок не возникает. При попытке записи в регистры, предназначенные только для чтения, операция записи завершается успешно, однако, состояние регистров не изменяется.

Таблица 1.219 – Карта регистров контроллера GPIO

Адрес	Название	Доступ	RST	Описание
0x000	GPIO_ID	RO	32’h4f49_5047	Идентификатор устройства
0x004	GPIO_VER	RO	32’h0A00_0120[7)]	Номер версии устройства и конфигурация
0x008-0x00C	-	-	32’he7707_ad7	Резерв
0x010	GPIO_INPUT	RO[1)]	32’h0[8)]	Состояние входов в режиме GPIO (GPIO_0 = [0] бит, GPIO_31 = [31] бит)
0x014	GPIO_OUTPUT	RW[1)]	32’h0	Задание выходов в режиме GPIO (GPIO_0 = [0] бит, GPIO_31 = [31] бит)
0x018[2)]	GPIO_SET_CLR_0_15	WO[1)]	32’h0	Задание 1(биты [15:0]) или 0(биты [31:16]) на выходах в режиме GPIO для линий 0-15
0x01C[2)]	GPIO_SET_CLR_16_31	WO[1)]	32’h0	Задание 1(биты [15:0]) или 0(биты [31:16]) на выходах в режиме GPIO для линий 16-31
0x020-0x09C[2)]	GPIO_i_CFG	RW[1,3)]	RST_GPIO_i_CFG[8,9)]	Настройка буферов ввода/вывода, переключения линий на работу с альтернативными функциями и мультиплексирование линий: i = 0 (адрес 0x020) … i = 31(адрес 0x09c)
0x0A0[2)]	GPIO_CFG_WRITE_PROTECT	RW[6)]	32’h1	Защита от записи в регистры GPIO_i_CFG, GPIO_SET_OE_0_15, GPIO_SET_OE_16_31
0x0A4[2)]	GPIO_LOCK_WRITE_PROTECT	RW[6)]	32’h1	Защита от записи в регистр GPIO_LOCK_CFG

Адрес	Название	Доступ	RST	Описание
0x0A8	GPIO_LOCK_CFG	RW[1,4)]	32'h0	Запрет перенастройки буферов ввода/вывода (действует до аппаратного сброса): [0] бит - GPIO_0_CFG, GPIO_SET_OE_0_15[0], GPIO_SET_OE_0_15[16] … [31] бит - GPIO_31_CFG, GPIO_SET_OE_16_31[15], GPIO_SET_OE_16_31[31]
0x0AC	-	-	32'he7707_ad7	Резерв
0x0B0[5)]	GPIO_BUF_SECURE	RW	32'h0	Изменение уровня доступа к линиям GPIO и к настройке буферов им соответствующей: 0 - доступ и в защищенном и в незащищенном режиме 1 - доступ только в защищенном режиме (тогда при незащищенном обращении запись будет игнорироваться, а чтение будет возвращать 0) [0] бит - определяет доступ к GPIO_INPUT бит [0], GPIO_OUTPUT бит [0], GPIO_SET_CLR_0_15 биты [0] и [16], GPIO_0_CFG все биты, GPIO_SET_OE_0_15 биты [0] и [16], GPIO_LOCK_CFG бит [0] … [31] бит - определяет доступ к GPIO_INPUT бит [31], GPIO_OUTPUT бит [31], GPIO_SET_CLR_16_31 биты [15] и [31], GPIO_31_CFG все биты, GPIO_SET_OE_16_31 биты [15] и [31], GPIO_LOCK_CFG бит [31]
0x0B4[2,5)]	GPIO_WP_ACCESS	RW	32'h0	Изменение уровня доступа регистрам GPIO_CFG_WRITE_PROTECT и GPIO_LOCK_WRITE_PROTECT
0x0B8-0x0BC	-	-	32'he7707_ad7	Резерв
0x0C0[2)]	GPIO_SET_OE_0_15	WO[1,3)]	32'h0	Дополнительная (в добавок к GPIO_i_CFG) возможность настройки направления работы буферов в режиме GPIO для линий 0-15, позволяет одновременно (одной записью) настраивать несколько линий GPIO
0x0C4[2)]	GPIO_SET_OE_16_31	WO[1,3)]	32'h0	Дополнительная (в добавок к GPIO_i_CFG) возможность настройки направления работы буферов в режиме GPIO для линий 16-31, позволяет одновременно (одной записью) настраивать несколько линий GPIO
0x0C8-0xFFC	-	-	32'he7707_ad7	Резерв

Примечания:

1) Доступ к регистру или к части его бит зависит от настроек заданных в регистре GPIO_BUF_SECURE;

2) Описание регистра приведено ниже;
3) Доступ к регистру или к части его бит зависит от настроек заданных в регистре GPIO_CFG_WRITE_PROTECT & GPIO_LOCK_CFG;
4) Доступ к регистру или к части его бит зависит от настроек заданных в регистре GPIO_LOCK_WRITE_PROTECT;
5) Регистр доступен только в защищённом режиме;
6) Доступ к регистру зависит от настроек заданных в регистре GPIO_WP_ACCESS;
7) Значение зависит от параметров заданных при конфигурации блока;
8) Значение регистра может измениться сразу после снятия RST, поэтому прочитанное значение может отличаться (при использовании параметра USE_CFG_PORT = 1, сразу после снятия RST будет защелкнуто значение с массива cfg_port_i);
9) Значение части бит регистра зависит от параметров RST_GPIO_0_CFG, RST_GPIO_1_CFG ... RST_GPIO_31_CFG заданных при интеграции блока;

1.3.9.1.3.2 Описание полей регистров контроллера GPIO

1.3.9.1.3.2.1 GPIO_SET_CLR_0_15 (0x018) и GPIO_SET_CLR_16_31 (0x01C)

Описание полей регистров GPIO_SET_CLR_0_15 и GPIO_SET_CLR_16_31 приведено в таблице 1.220.

Таблица 1.220 – Поля регистра GPIO_SET_CLR

Бит	Поле	RST	Описание
31:16	bits_clr[1,2)]	16'h0000	Запись "1" в i-ый бит поля bits_clr (бит регистра 16+i, i от 0-15) устанавливает "0" на: соответствующий i-ый выход блока для регистра GPIO_SET_CLR_0_15 соответствующий i+16 выход блока для регистра GPIO_SET_CLR_16_31
15:0	bits_set[1,2)]	16'h0000	Запись "1" в i-ый бит поля bits_set (i от 0 до 15) устанавливает "1" на: соответствующий i-ый выход блока для регистра GPIO_SET_CLR_0_15 соответствующий i+16 выход блока для регистра GPIO_SET_CLR_16_31

Примечания:

1) Доступность поля или части его бит зависит от настроек заданных в регистре GPIO_BUF_SECURE
2) Приоритет у установки "0" (при записи 1 в [0] и [16] биты на нулевом выходе будет "0").
   Запись "0" в любые биты не изменяет состояние.
   Чтение всегда возвращает все "0".

1.3.9.1.3.2.2 GPIO_i_CFG (0x020-0x09C)

Настройка производится только для буферов, которые есть в таблице 1.2.2.

Описание полей регистра GPIO_i_CFG приведено в таблице 1.221.

Если в регистре GPIO_BUF_SECURE установлен только защищенный доступ до данного регистра, а приходит обращение в незащищенном режиме, то запись будет игнорироваться, а чтение будет возвращать 0.

Таблица 1.221 – Поля регистра GPIO_i_CFG

Бит	Поле	RST	Описание
31	gpio_input(RO)	-	Запись игнорируется. Чтение возвращает значение на входе GPIO_i (где "i" соответствует "i" в "GPIO_i_CFG")
30	bits_clr(WO)[1)]	1’b0	Запись "1" в этот бит устанавливает в "0" на соответствующий i-ый выход блока (где "i" соответствует "i" в "GPIO_i_CFG"). Запись "0" игнорируется. Чтение возвращает "0"
29	bits_set(WO)[1)] / gpio_output(RO)	RST_GPIO_i_CFG[2)]	Запись "1" в этот бит устанавливает в "1" на соответствующий i-ый выход блока (где "i" соответствует "i" в "GPIO_i_CFG"). Запись "0" игнорируется. Чтение возвращает значение на выходе GPIO_i
28:27	mode	RST_GPIO_i_CFG[2)]	Режим работы: 2’b00 - включение альтернативной функций (alt_mode) 2’b01 - включение альтернативной аналоговой функции (analog_mode) 2’b10 - режим работы GPIO (gpio_mode) 2’b11 - режим работы LCD (lcd_mode)
26	-	-	Резерв
25:20	ifmux_sel	RST_GPIO_i_CFG[2)]	Выбор настройки мультиплексирования в блоке IFMUX (при mode = 2’b00 (alt_mode) и alt_func = 15). К "i" буферу(buf_data_*) подключаются сигналы ifmux_data_*[ifmux_sel], где [i] соответствует "i" в "GPIO_i_CFG")
19:16	alt_func	RST_GPIO_i_CFG[2)]	Выбор альтернативной цифровой функции при mode = 2’b00 (alt_mode): alt_func = 0-14 (выбирается линия альтернативной функций: к "i" буферу(buf_data_*) подключаются сигналы "alt_data_*[i][alt_func]", где [i] соответствует "i" в "GPIO_i_CFG") alt_func = 15 (выбирается линия подключенная к IFMUX заданная в поле ifmux_sel)
15:14	-	-	Резерв
13	pull_enable	RST_GPIO_i_CFG[2)]	Включение pull_sel (При pull_enable = 0 буфер ввода/вывода без нагрузки - Hi-Z) При mode = 2’b01 (analog_mode) или mode = 2’b11 (lcd_mode) данное поле равно 0
12	pull_sel	RST_GPIO_i_CFG[2)]	1 - Pull-up; 0 - Pull-down (только при pull_enable = 1)
11:10	-	-	Резерв

Бит	Поле	RST	Описание
9	schmitt	RST_GPIO_i_CFG[2)]	Включение триггера Шмитта при работе в режиме входа
8	input_enable	RST_GPIO_i_CFG[2)]	При mode != 2'b00 (НЕ alt_mode) или alt_func = 15 (режиме работы IFMUX) (buf_data_ie_o = input_enable): 1 - буфер работает как вход; 0 - входной буфер НЕ работает. При mode = 2'b00 (alt_mode) и alt_func != 15 работой входного буфера управляют сигналы alt_data_ie_arr_i. При mode = 2'b00 (alt_mode) и alt_func = 15 для входов и двунаправленных сигналов должно быть установлено значение input_enable = 1 При mode = 2'b01 (analog_mode) или mode = 2'b11 (lcd_mode) должно быть установлено значение input_enable = 0
7	invert	RST_GPIO_i_CFG[2)]	Инвертирование выходного/входного значения буфера При mode = 2'b01 (analog_mode) или mode = 2'b11 (lcd_mode) данное поле равно 0
6	slew_rate	RST_GPIO_i_CFG[2)]	Скорость изменения сигнала в режиме выхода
5:4	driver_strength	RST_GPIO_i_CFG[2)]	Выходной ток: 2'b00 - резерв; 2'b01 - номинальные 4 мА; 2'b10 - номинальные 8 мА; 2'b11 - резерв
3:2	delay_data_out	RST_GPIO_i_CFG[2)]	Задание задержки установки значения на выходном буфере (buf_data_o) относительно записи по APB при режиме работы mode = 2'b10 (gpio_mode) : 2'b00 - нет задержки; 2'b01 - задержка половина такта опорной частоты pclk_i (значение устанавливается по заднему фронту(срезу) опорной частоты pclk_i) ; 2'b10 - задержка на 1 такт опорной частоты pclk_i; 2'b11 - резерв. Данное поле рекомендуется использовать для уменьшения пикового тока потребления групп буферов при одновременном переключении большого количества буферов при режиме работы mode = 2'b10 (gpio_mode) (При этом важно учитывать настройку выходного тока этих групп буферов - driver_strength). Так же данное поле позволяет более гибко настраивать временные диаграммы с использованием меньшего числа записей по интерфейсу APB (например, при имитации на GPIO интерфейсов, имеющих строб валидности данных, delay_data_out позволяет задерживать строб, чтобы на шинах данных успели закончится переходные процессы до установки строба)

Бит	Поле	RST	Описание
1	open_drain	RST_GPIO_i_CFG[2)]	Режим работы выходного буфера: open_drain = 1 (при этом должен быть или внешний резистор подтягивающий выход к питанию или установлены pull_enable = 1 и pull_sel = 1); или push-pull (open_drain = 0). В режиме open_drain = 1 (при mode = 2'b00 (alt_mode) или mode = 2'b10 (gpio_mode)) выходной сигнал "buf_data_o" переводится в 0, а значение, которое должно было быть на "buf_data_o", подается инвертированным на управление буфером "buf_data_oe_o". При mode = 2'b01 (analog_mode) или mode = 2'b11 (lcd_mode) выходной буфер НЕ работает (buf_data_oe_o = 0). Данное поле равно 0
0	output_enable	RST_GPIO_i_CFG[2)]	При mode = 2'b10 (gpio_mode): 1 - буфер работает как выход; 0 - выходной буфер НЕ работает (входной буфер включается отдельно в input_enable). При mode = 2'b01 (analog_mode) или mode = 2'b11 (lcd_mode) выходной буфер НЕ работает (buf_data_oe_o = 0). Данное поле равно 0. При mode = 2'b00 (alt_mode) работой выходного буфера управляют сигналы alt_data_oe_arr_i или ifmux_data_oe_i

Примечания:

1) Приоритет у установки "0" (при записи 1 в [29] и [30] биты на выходе будет "0").
2) Значение полей зависит от параметров RST_GPIO_0_CFG, RST_GPIO_1_CFG ... RST_GPIO_31_CFG заданных при интеграции блока

1.3.9.1.3.2.3 GPIO_CFG_WRITE_PROTECT (0x0A0)

Описание полей регистра GPIO_CFG_WRITE_PROTECT приведено в таблице 1.222.

Таблица 1.222 – Поля регистра GPIO_CFG_WRITE_PROTECT

Бит	Поле	RST	Описание
31:8	gpio_cfg_wp_en	0	Запись в это поле значения 0x545257 позволяет изменить поле gpio_cfg_wp, иначе запись в gpio_cfg_wp игнорируется. Чтение этого поля возвращает 0
7:1	-	-	Резерв
0	gpio_cfg_wp	1	1 - запись в регистры GPIO_0_CFG - GPIO_31_CFG запрещена 0 - запись в регистры GPIO_0_CFG - GPIO_31_CFG разрешена Для изменения этого поля надо одновременно записать код в поле gpio_cfg_wp_en Рекомендуется устанавливать gpio_cfg_wp = 0 только на момент изменения регистров GPIO_0_CFG - GPIO_31_CFG и затем устанавливать gpio_cfg_wp = 1

1.3.9.1.3.2.4 GPIO_LOCK_WRITE_PROTECT (0x0A4)

Описание полей регистра GPIO_LOCK_WRITE_PROTECT приведено в таблице 1.223.

Таблица 1.223 – Поля регистра GPIO_LOCK_WRITE_PROTECT

Бит	Поле	RST	Описание
31:8	gpio_lock_wp_en	0	Запись в это поле значения 0x545257 позволяет изменить поле gpio_lock_wp, иначе запись в gpio_lock_wp игнорируется. Чтение этого поля возвращает 0
7:1	-	-	Резерв
0	gpio_lock_wp	1	1 - запись в регистр GPIO_LOCK_CFG запрещена 0 - запись в регистр GPIO_LOCK_CFG разрешена Для изменения этого поля надо одновременно записать код в поле gpio_lock_wp_en Рекомендуется устанавливать gpio_lock_wp = 0 только на момент изменения регистра GPIO_LOCK_CFG и затем устанавливать gpio_lock_wp = 1

1.3.9.1.3.2.5 GPIO_WP_ACCESS (0x0B4)

Описание полей регистра GPIO_WP_ACCESS приведено в таблице 1.224.

Таблица 1.224 – Поля регистра GPIO_WP_ACCESS

Бит	Поле	RST	Описание
31:6	-	-	Резерв
5:4	gpio_lock_wp_access		Уровень доступа к регистру GPIO_LOCK_WRITE_PROTECT 2’b00 - доступ и в защищенном и в незащищенном режиме 2’b01 - запись только в защищенном режиме (при незащищенном обращении запись будет игнорироваться), чтение и в защищенном и в незащищенном режиме 2’b1x - доступ только в защищенном режиме (тогда при незащищенном обращении запись будет игнорироваться, а чтение будет возвращать 0)
3:2	-	-	Резерв
1:0	gpio_cfg_wp_access	0	Уровень доступа к регистру GPIO_CFG_WRITE_PROTECT 2’b00 - доступ и в защищенном и в незащищенном режиме 2’b01 - запись только в защищенном режиме (при незащищенном обращении запись будет игнорироваться), чтение и в защищенном и в незащищенном режиме 2’b1x - доступ только в защищенном режиме (тогда при незащищенном обращении запись будет игнорироваться, а чтение будет возвращать 0)

1.3.9.1.3.2.6 GPIO_SET_OE_0_15 (0x0C0) и GPIO_SET_OE_16_31 (0x0C4)

Описание полей регистров GPIO_SET_OE_0_15 и GPIO_SET_OE_16_31 приведено в таблице 1.225.

Таблица 1.225 – Поля регистра GPIO_SET_OE

Бит	Поле	RST	Описание
31:16	set_oe	16'h0000	Запись "1" в i-ый бит поля set_oe (бит регистра 16+i, i от 0-15) устанавливает GPIO_i_CFG[output_enable] = 1 и GPIO_i_CFG[input_enable] = 0 (где GPIO_i_CFG от 0-15 для регистра GPIO_SET_OE_0_15 и от 16-31 для регистра GPIO_SET_OE_16_31). Запись "0" в i-ый бит поля set_oe (бит регистра 16+i, i от 0-15) устанавливает GPIO_i_CFG[output_enable] = 0 и GPIO_i_CFG[input_enable] = 1 (где GPIO_i_CFG от 0-15 для регистра GPIO_SET_OE_0_15 и от 16-31 для регистра GPIO_SET_OE_16_31)
15: 0	mask_set_oe	16'h0000	Запись "1" в i-ый бит поля mask_set_oe (i от 0 до 15) включает перенастройку направления работы: соответствующего i-ого буфера для регистра GPIO_SET_OE_0_15 соответствующего i+16 буфера для регистра GPIO_SET_OE_16_31

1.3.9.2 Контроллер интерфейса GPIO батарейного домена питания (BKPGPIO)

1.3.9.2.1 Описание BKPGPIO

1.3.9.2.1.1 Настройка буферов ввода/вывода и режима работы

Настройка буферов ввода/вывода и режима работы производится в регистрах BKPGPIO_CFG0 для каждого буфера **независимо**.

Если буфер работает в режиме GPIO, то направление (вход или выход) задается программно в регистрах BKPGPIO_CFG0. Если буфер работает в режиме альтернативной цифровой функции направлением работы буфера управляют сигналы func_data_oe_i.

1.3.9.2.1.2 Альтернативные функции

Контроллер BKPGPIO поддерживает 5 альтернативных функций. В поле alt_func в регистре BKPGPIO_CFG0 выбирается какая из альтернативных функций подключена к буферу. При включенной альтернативной функции, буфер управляется сигналами func_data_oe_i, func_data_ie_i, func_data_i. В таб. 1.226 представлен набор альтернативных функции для каждого из буферов.

Таблица 1.226 – Альтернативный функции BKPGPIO

Название функции	Направление	Описание
LSCO	Выход	Синхросигнал низкой частоты (см. 1.3.7.2)
TAMP_INx	Вход	Блок TAMP по изменению входного сигнала генерирует запрос на стирание данных и/или блокировку доступа. Это же событие может использоваться и RTC для генерации wakeup и/или проставления временных меток. Возможно использование нескольких входов. Например, по одному мы только блокируем запись, а по второму уже все стираем
TAMP_OUT	Выход	Установился хотя бы один из флагов, то есть произошло изменения на входах TAMP (внутренние и внешние события)
RTC_ALARM_A	Выход	Блок RTC может выдавать на выход 1 после срабатывания будильника A. Сброс происходит программно
RTC_CALIB	Выход	Калибровочный сигнал RTC с частотой 1Гц, 256Гц или 512Гц
RTC_WAKEUP	Выход	Блок RTC генерирует сигнал wakeup по одному из следующий событий: RTC_ALARM_A, RTC_ALARM_B, TAMP_OUT. Выбор события настраивается в RTC

1.3.9.2.2 Описание регистров

1.3.9.2.2.1 Общее описание регистров

Контроллер имеет набор программно доступных регистров, требующий 4 КБайт адресов на системной шине APB.

Поддерживается только 32-х разрядный доступ. Байтовая или двух-байтовая запись будет проигнорирована и возвращена ошибка шины.

Карта регистров приведена в таблице 1.227.

Адреса, который не указаны в таблице 1.227 считаются резервными. Чтение из них возвращает 0xEEFFADDC, а запись игнорируется. BKPGPIO контроллер не различает secure и non-secure транзакции.

1.3.9.2.2.2 Карта регистров

Таблица 1.227 – Карта регистров LCD

Адрес	Название	Доступ	Значение после сброса	Описание
0x000	BKPGPIO_ID	ЧТ	0x47504B42	BKPGPIO ID
0x004	BKPGPIO_VERSION	ЧТ	0x1	Версия BKPGPIO
0x020	BKPGPIO_CFG0	ЧТ/ЗП	0x0	Конфигурационный регистр буфера буфера PE0
0x0A0	BKPGPIO_CFG_WP	ЧТ/ЗП	0x1	Защита от записи в регистры BKPGPIO_CFG0
0x0A4	BKPGPIO_LOCK_WP	ЧТ/ЗП	0x1	Защита от записи в регистр BKPGPIO_LOCK_CFG
0x0A8	BKPGPIO_LOCK_CFG	ЧТ/ЗП	0x0	Запрет перенастройки буферов ввода/вывода (BKPGPIO_CFG0)

Адрес	Название	Доступ	Значение после сброса	Описание
0x0B4	BKPGPIO_WP_ACCESS	ЧТ/ЗП	0x0	Запрет перенастройки буферов ввода/вывода (BKPGPIO_CFG0)

1.3.9.2.2.3 Описание полей регистров

1.3.9.2.2.3.1 BKPGPIO_ID (0x000)

Регистр идентификатора BKPGPIO контроллера.

В таблице 1.228 представлено описание полей регистра.

Таблица 1.228 – Описание полей регистра BKPGPIO_ID

Биты	Поле	Доступ	Значение после сброса	Описание
31:0	BKPGPIO_ID	ЧТ	0x47504B42	Идентификатор BKPGPIO контроллера

1.3.9.2.2.3.2 BKPGPIO_VERSION (0x004)

Регистр идентификатора BKPGPIO контроллера.

В таблице 1.229 представлено описание полей регистра.

Таблица 1.229 – Описание полей регистра BKPGPIO_VERSION

Биты	Поле	Доступ	Значение после сброса	Описание
31:0	BKPGPIO_VERSION	ЧТ	0x1	Версия BKPGPIO контроллера

1.3.9.2.2.3.3 BKPGPIO_CFG0 (0x020)

Регистр настройки буферов ввода/вывода BKPGPIO контроллера.

В таблице 1.230 представлено описание полей регистра.

Таблица 1.230 – Описание полей регистра BKPGPIO_CFG0

Биты	Поле	Доступ	Значение после сброса	Описание
31	GPIO_INPUT	ЧТ/ЗП	0	Запись игнорируется. Чтение возвращает значение на входе буфера
30	bits_clr	ЧТ/ЗП	0	Запись "1" в этот бит устанавливает в "0" выход буфера в режиме GPIO. Запись "0" игнорируется. Чтение возвращает "0". **приоритет у установки "0"**
29	bits_set	ЧТ/ЗП	0	Запись "1" в этот бит устанавливает в "1" выход буфера режиме GPIO. Запись "0" игнорируется. Чтение возвращает значение на выходе **приоритет у установки "0"**

Биты	Поле	Доступ	Значение после сброса	Описание
28	mode	ЧТ/ЗП	0	Режим работы: 0 - включение альтернативной функций (alt_mode) 1 - режим работы GPIO (gpio_mode)
27:20	-	ЧТ/ЗП	0	Зарезервировано
19:16	alt_func	ЧТ/ЗП	0	Выбор альтернативной функции (alt_mode): 4'h0 - 4'h6 - Зарезервировано; 4'h7 - Альтернативная функция 7 (LSCO); 4'h8 - Альтернативная функция 8 (TAMP_IN); 4'h9 - Альтернативная функция 9 (TAMP_OUT); 4'ha - Альтернативная функция 10 (RTC_ALARM); 4'hb - Альтернативная функция 11 (RTC_CALIB); 4'hc - Альтернативная функция 12 (RTC_WAKEUP); 4'hd - 4'hf - Зарезервировано.
15:14	-	ЧТ/ЗП	0	Зарезервировано
13	pull_enable	ЧТ/ЗП	0	Включение pull_sel (При pull_enable = 0 буфер ввода/вывода без нагрузки - Hi-Z)
12	pull_sel	ЧТ/ЗП	0	1 - Pull-up; 0 - Pull-down (только при pull_enable = 1)
11:10	-	ЧТ/ЗП	0	Зарезервировано
9	schmitt	ЧТ/ЗП	0	Включение триггера Шмитта при работе в режиме входа
8	input_enable	ЧТ/ЗП	0	При mode = 1: 1 - буфер работает как вход; 0 - входной буфер НЕ работает. При mode = 0 (alt_mode) работой входного буфера управляют сигналы func_data_ie_i.
7	invert	ЧТ/ЗП	0	Инвертирование выходного/входного значения буфера 0 - входной сигнал не инвертируется 1 - входной сигнал инвертируется
6	slew_rate	ЧТ/ЗП	0	Скорость изменения сигнала в режиме выхода
5:4	driver_strength	ЧТ/ЗП	0	Выходной ток: 2'b00 - резерв; 2'b01 - номинальные 4 мА; 2'b10 - номинальные 8 мА; 2'b11 - резерв
3:2	-	ЧТ/ЗП	0	Зарезервировано
1	open_drain	ЧТ/ЗП	0	Режим работы выходного буфера: 1 - open_drain (при этом должен быть или внешний резистор подтягивающий выход к питанию или установлены pull_enable = 1 и pull_sel = 1); 0 - push-pull. В режиме open_drain = 1 выходной сигнал "buf_data_o" переводится в 0, а значение, которое должно было быть на "buf_data_o", подается инвертированным на управление буфером "buf_data_oe_o".

Биты	Поле	Доступ	Значение после сброса	Описание
0	output_enable	ЧТ/ЗП	0	При mode = 1 (gpio_mode): 1 - буфер работает как выход; 0 - выходной буфер НЕ работает (входной буфер включается отдельно в input_enable). При mode = 0 (alt_mode) работой выходного буфера управляют сигналы func_data_oe_i.

1.3.9.2.2.3.4 BKPGPIO_CFG_WP (0x0A0)

Регистр защиты перенастройки BKPGPIO контроллера.

В таблице 1.231 представлено описание полей регистра.

Таблица 1.231 – Описание полей регистра BKPGPIO_CFG_WP

Биты	Поле	Доступ	Значение после сброса	Описание
0	GPIO_CFG_WP	ЧТ/ЗП	1	Для изменения этого поля надо одновременно записать код в поле GPIO_CFG_WP_EN. Рекомендуется устанавливать GPIO_CFG_WP = 0 только на момент изменения регистров BKPGPIO_CFG0 и затем устанавливать GPIO_CFG_WP = 1 1 - запись в регистры BKPGPIO_CFG0 запрещена 0 - запись в регистры BKPGPIO_CFG0 разрешена
7:1	-	ЧТ	0	Зарезервировано
31:8	GPIO_CFG_WP_EN	ЗП	0	Запись в это поле значения 0x545257 позволяет изменить поле GPIO_CFG_WP, иначе запись в GPIO_CFG_WP игнорируется. Чтение этого поля возвращает 0.

1.3.9.2.2.3.5 BKPGPIO_LOCK_WP (0x0A4)

Регистр защиты блокировки перенастройки BKPGPIO контроллера.

В таблице 1.232 представлено описание полей регистра.

Таблица 1.232 – Описание полей регистра BKPGPIO_LOCK_WP

Биты	Поле	Доступ	Значение после сброса	Описание
0	GPIO_LOCK_WP	ЧТ/ЗП	1	Для изменения этого поля надо одновременно записать код в поле GPIO_LOCK_WP_EN Рекомендуется устанавливать GPIO_LOCK_WP = 0 только на момент изменения регистра GPIO_LOCK_CFG и затем устанавливать GPIO_LOCK_WP = 1 1 - запись в регистр GPIO_LOCK_CFG запрещена 0 - запись в регистр GPIO_LOCK_CFG разрешена
7:1	-	ЧТ	0	Зарезервировано

Биты	Поле	Доступ	Значение после сброса	Описание
31:8	GPIO_LOCK_WP_EN	ЗП	0	Запись в это поле значения 0x545257 позволяет изменить поле GPIO_LOCK_WP, иначе запись в GPIO_LOCK_WP игнорируется. Чтение этого поля возвращает 0.

1.3.9.2.2.3.6 BKPGPIO_LOCK_CFG (0x0A8)

Регистр блокировки перенастройки BKPGPIO контроллера.

В таблице 1.233 представлено описание полей регистра.

Таблица 1.233 – Описание полей регистра BKPGPIO_LOCK_CFG

Биты	Поле	Доступ	Значение после сброса	Описание
0	LOCK0	ЧТ/ЗП	0	Блокировка доступа к регистру BKPGPIO_CFG0 0 - Изменение регистра BKPGPIO_CFG0 разрешено. 1 - Изменение регистра BKPGPIO_CFG0 заблокировано (запись игнорируется). Сброс бита LOCK0 в 0 происходит только после аппаратного сброса.
31:1	-	ЧТ	0	Зарезервировано

1.3.9.2.2.3.7 BKPGPIO_WP_ACCESS (0x0B4)

Регистр настройки доступа до регистров блокировок BKPGPIO контроллера.

В таблице 1.234 представлено описание полей регистра.

Таблица 1.234 – Описание полей регистра BKPGPIO_WP_ACCESS

Биты	Поле	Доступ	Значение после сброса	Описание
1:0	GPIO_CFG_WP_ACCESS	ЧТ/ЗП	0	Уровень доступа к регистру GPIO_CFG_WRITE_PROTECT 2’b00 - доступ и в защищенном и в незащищенном режиме 2’b01 - запись только в защищенном режиме (при незащищенном обращении запись будет игнорироваться), чтение и в защищенном и в незащищенном режиме. 2’b1x - доступ только в защищенном режиме (тогда при незащищенном обращении запись будет игнорироваться, а чтение будет возвращать 0)
3:2	-	ЧТ	0	Зарезервировано

Биты	Поле	До-ступ	Значение по-сле сброса	Описание
5:4	GPIO_LOCK_WP_ACCESS	ЧТ/ЗП	0	Уровень доступа к регистру GPIO_LOCK_WRITE_PROTECT 2'b00 - доступ и в защищенном и в незащищенном режиме 2'b01 - запись только в защищенном режиме (при незащищенном обращении запись будет игнорироваться), чтение и в защищенном и в незащищенном режиме. 2'b1x - доступ только в защищенном режиме (тогда при незащищенном обращении запись будет игнорироваться, а чтение будет возвращать 0)
31:6	-	ЧТ	0	Зарезервировано

1.3.9.3 Контроллер интерфейса USART (USART)

1.3.9.3.1 Особенности USART контроллера

Контроллер USART обладает следующими характеристиками:

– Поддержка синхронного (USART) и асинхронного (UART) режимов работы

– Поддерживаемые протоколы работы в режиме UART:

  – LIN;

  – RS-232;

  – RS-422;

  – RS-485;

– параметры обмена:

  – число бит в посылке:

    – 5 бит;

    – 6 бит;

    – 7 бит;

    – 8 бит;

    – 9 бит;

  – число стоповых бит:

    – 1 бит;

    – 2 бита;

  – контроль чётности;

– размер RX буфера (слов): 32;

– размер TX буфера (слов): 32;

– дополнительные возможности:

– программируемый делитель опорной частоты (целая и дробная часть);

– 4 побитовых фильтра с масками;

– настраиваемая схема сэмплирования символов:

– 16 сэмплов на символ;

– 8 сэмплов на символ;

– 1 сэмпл на символ (для высокоскоростного обмена);

– поддержка сигналов управления потоком (RTS/CTS);

– автоопределение скорости передачи (baudrate);

– генерация и приём Break-последовательностей;

– генерация маскируемых прерываний;

– дополнительные режимы работы:

– режим сна;

– loopback;

– интерфейс программирования и передачи данных: APB4 slave.

1.3.9.3.2 Структурная схема USART

Структурная схема USART приведена на рисунке 1.35. Описание блоков из который состоит USART контроллер приведено в таблице 1.235.

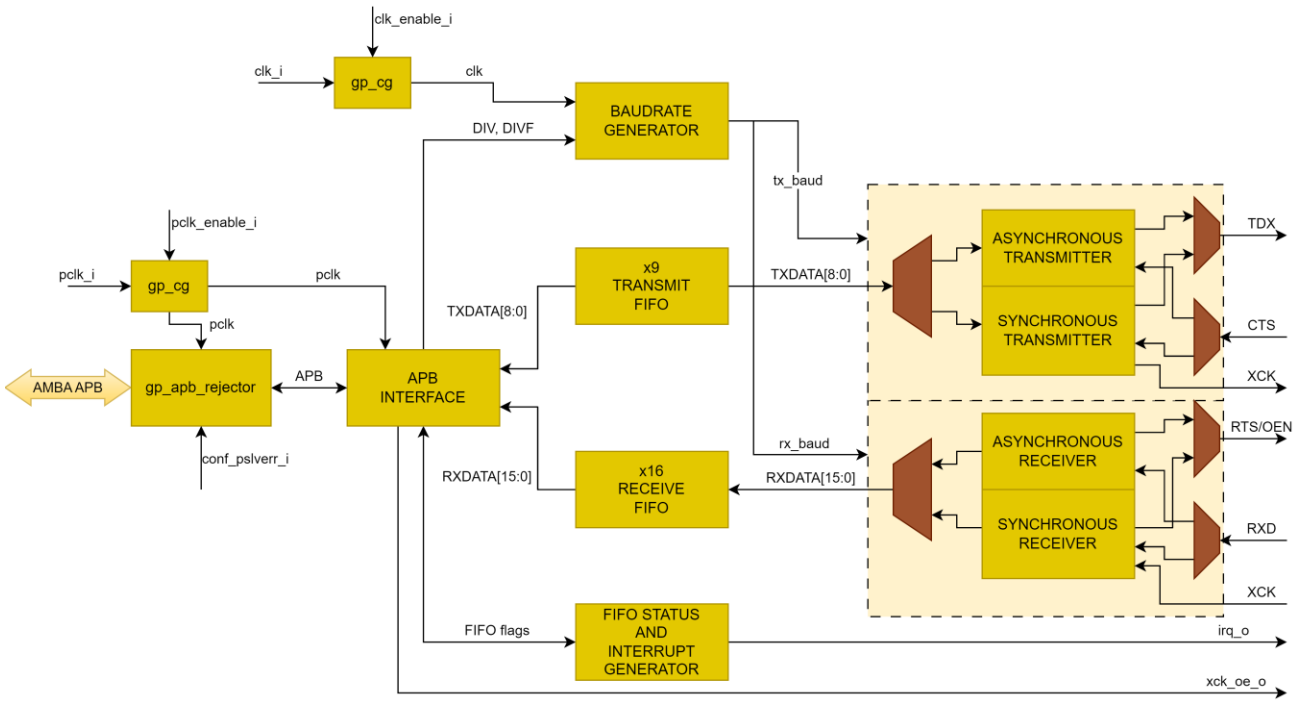


Рисунок 1.35 – Структурная схема USART

Таблица 1.235 – Описание назначения составных блоков USART

Название блока	**Назначение и основные функции**
APB INTERFACE AND REGISTER BLOCK	Блок служит для чтения и записи программного сброса, чтения регистров состояний, чтения и записи управляющих регистров, чтения FIFO приёмника и записи FIFO передатчика.
BAUD RATE GENERATOR	Блок содержит счётчик, генерирующий внутреннюю тактовую частоту – rx_baud и tx_baud, для контроля приёма и передачи USART.
TRANSMIT FIFO	Блок FIFO передатчика, представляющий собой буфер памяти FIFO шириной 9 бит и настраиваемой глубиной (параметр TX_BUF_DEPTH). Данные записываются через интерфейс APB и хранятся в FIFO передатчика, пока не будут считаны логикой синхронного или асинхронного передатчика.
RECEIVE FIFO	Блок FIFO приёмника, представляющий собой буфер памяти FIFO шириной 16 бит и настраиваемой глубиной (параметр RX_BUF_DEPTH). Получаемые данные и соответствующие биты состояния помещаются в FIFO приёмника до тех пор, пока не будут считаны через APB интерфейс.
ASYNCHRONOUS TRANSMITTER	Блок реализует параллельно-последовательное преобразование над данными, считанными из FIFO передатчика, при асинхронном режиме работы USART.
ASYNCHRONOUS RECEIVER	Блок реализует последовательно-параллельное преобразование над полученным потоком битов после обнаружения подтвержденного стартового бита, при асинхронном режиме работы USART.
SYNCHRONOUS TRANSMITTER	Блок реализует параллельно-последовательное преобразование над данными, считанными из FIFO передатчика, а также служит для генерации сигнала синхронизации, при синхронном режиме работы USART.
SYNCHRONOUS RECEIVER	Блок реализует последовательно-параллельное преобразование над полученным потоком битов при обнаружении сигнала синхронизации, при синхронном режиме работы USART.
FIFO STATUS AND INTERRUPT GENERATION	Блок служит для хранения состояний FIFO приёмника и FIFO передатчика, а также служит для генерации прерываний USART.
gp_apb_rejector	Блок служит для корректного завершения APB транзакций, когда с APB части USART контроллера снят тактовый сигнал. Данный блок может завершать транзакции с ошибкой (pslverr_o в единице) или без нее. Это задается с помощью входа conf_pslverr_i. Если на этом входе единица, то транзакции будут завершаться с ошибкой, когда на USART не подан тактовый сигнал. При этом транзакции записи игнорируются, а чтение возвращает 0x0BAD_ACCE5.
gp_cg	Блок служит для блокирования подачи тактового сигнала. Так как USART имеет два тактовый входа: pclk_i и clk_i, то и блоков gp_cg тоже используется два. Соответственно входов для блокировки тактовый сигналов также два: pclk_enable_i и clk_enable_i. Если на входе блокировки тактового сигнала единицы, то это тактовый сигнал будет снят. Так же данный блок включает в себя синхронизаторы сброса, они дают задержку на снятие сброс в два такта той частоты, на которой они работают, а сброс происходит асинхронно.
SAMPLING	Блок входит в состав ASYNCHRONOUS RECEIVER и отвечает за: – выборку входных данных; – определения скорости передачи данных; – фазовую подстройку; – генерирование сигнала "бит принят" (rx_trigger) для остальных блоков.

1.3.9.3.3 Описание внешних выводов контроллера USART

Список внешних выводов контроллера USART и их описание приведены в таблице 1.236

Таблица 1.236 – Список внешних выводов контроллера USART

Наименование вывода	Тип (Вх/Вых)	Функциональное назначение
RXD	Вх	Линия приема данных
TXD	Вых	Линия передачи данных
RTS/OEN	Вых	Информирование об готовности принимать данные (request to send)
CTS	Вх	Разрешение отправки данных (clear to send).
XCK	Вых	Опорный тактовый сигнал в синхронном режим. Используется только когда включен режим master

1.3.9.3.4 Принципы функционирования USART

1.3.9.3.4.1 Формат передачи данных по USART в асинхронном режиме (UART)

Протокол передачи данных по USART при асинхронном режиме работы представлен на рис. 1.36. Данные передаются в виде посылок, включающих в себя стартовый бит, 5, 6, 7, 8 или 9 (только в режиме работы с APB) бит данных, бит чётности (опционально), 1 или 2 стоповых бита. Длительность всех битов одинакова.

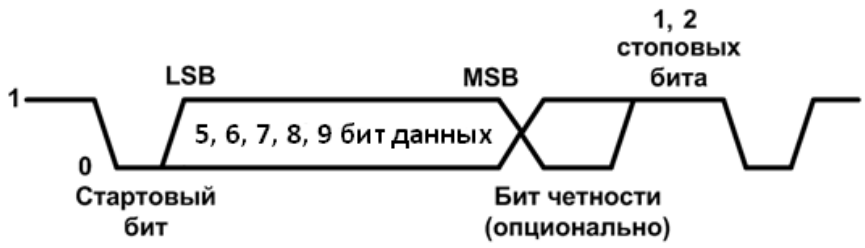


Рисунок 1.36 – Протокол передачи данных по USART в асинхронном режиме

Асинхронный режим включен, когда USART_SYNC[SME] = 0.

1.3.9.3.4.2 Формат передачи данных по USART в синхронном режиме

Протокол передачи данных по USART при синхронном режиме работы представлен на рис. 1.37. Данные передаются в виде посылок, включающих в себя стартовый бит, данный (5, 6, 7, 8 или 9 бит), бит чётности (опционально) и стоповый бит. Длительность всех битов одинакова. Во время стартового и стопового битов сигнал синхронизации XCK не передается.

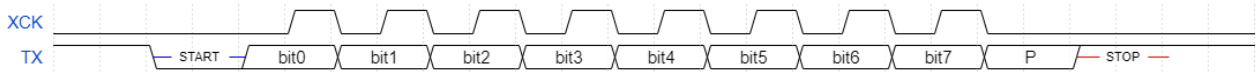


Рисунок 1.37 – Протокол приёма и передачи данных по USART в синхронном режиме

Синхронный режим включен, когда USART_SYNC[SME] = 1.

1.3.9.3.4.3 Интерфейс сброса

USART сбрасывается глобальными сигналами сброса. Также для сброса USART предусмотрен программный сброс. Он применяется путем записи 1 в нулевой бит регистра USART_SWRST. Программный сброс заканчивается, только после того как бит регистра USART_SWRST будет сброшен аппаратно.

1.3.9.3.4.4 Управление USART

Управляющие данные записываются через интерфейс AMBA APB в следующие регистры:

1) USART_CTRL – определяет следующие параметры: включение USART, режим работы через APB, задействован ли режим петли, выбор режима функционирования USART, задействованы ли сигналы RTS/CTS, выбор полярности сигналов RTS/CTS, выбор чётности, количество стоповых бит, длина слова и режим выгрузки данных из FIFO приёмника.

2) USART_BDIV – определяет значение делителя скорости передачи данных.

3) USART_FIFOWM – определяет уровни прерываний при заполнении FIFO приёмника и опустошении FIFO передатчика.

4) USART_TIMEOUT – определяет тайм-аут, при превышении которого при приёме данных возникает соответствующее прерывание.

5) USART_SYNC – определяет следующие параметры: задействован ли синхронный режим работы USART, полярность и фазу синхросигнала.

1.3.9.3.4.5 Генератор частоты работы USART

В данном контроллере реализовано два делителя входного тактового сигнала: DIV и DIVF. Первый является целочисленным, а второй дробным. Они делят входною частоту в соответствии со следующей формулой:

$$F_{div} = \frac{f_{clk}}{(div + \frac{divf}{256})}$$

Этот деленный опорный сигнал F_{dif} задает скорость передачи. В зависимости от выбранной схемы сэмплирования применятся одна из следующих формул:

$Baudrate = \frac{F_{div}}{16}$, при 16 битной схеме сэмплирования.

$Baudrate = \frac{F_{div}}{8}$, при 8 битной схеме сэмплирования.

$Baudrate = F_{div}$, при однобитной схеме сэмплирования.

Для примера рассмотрим опорную частоту USART равную f = 100 МГц. При данной частоте программируемый делитель опорной частоты в составе USART, позволяет задавать стандартные скорости передачи данных (baud rate) до 12.5 Мбод/с, а также нестандартные скорости передачи данных до 30 Мбод/с в пределах погрешности в 3 %.

При включенной однобитной схемы сэмплирования div не должно быть меньше 3.

Значение целого делителя DIV задается в битовом поле USART_BDIV[DIV], а дробного DIVF в USART_BDIV[DIVF]. Их можно менять только, когда USART неактивен, т.е. бит UEN регистра USART_CTRL равен 0.

В таблице 1.237 указаны примеры скоростей USART при дробном делителе равном нулю, оверсамплинге 16 и опорной частоте 100 МГц.

Так как для скорости 921600 погрешность получается больше 3%, то необходимо использовать дробный делитель. Если установить divf в 200 то, тогда ошибка составит 0.006%.

Таблица 1.237 – Список поддерживаемых стандартных скоростей передачи данных по USART (таблица для 100 МГц)

Скорость передачи, бод	**Погрешность, %**	**Целый делитель (дробный делитель равен нулю)**	
110	0.00032	56818	0xDDF2
134,5	0.00087	46468	0xB584
150	0.0008	41667	0xA2C3
300	0.0016	20833	0x5161
600	0.0032	10417	0x28B1
600	0.0032	10417	0x28B1
1200	0.0064	5208	0x1458
1800	0.0064	3472	0xD90
2000	0	3125	0xC35
2400	0.0064	2604	0xA2C
3600	0.0064	1736	0x6C8
4800	0.0064	1302	0x516
7200	0.0064	868	0x364
9600	0.0064	651	0x28B

Скорость передачи, бод	Погрешность, %	Целый делитель (дробный делитель равен нулю)	
19200	0.1472	326	0x146
38400	0.1472	163	0xA3
56000	0.352	112	0x70
115200	0.4672	54	0x36
921600	3.2192	7	0x7
6250000	0	1	0x1

В синхронном режиме работы сэмплирование не поддерживается, однако для расчета скорости передачи данных необходимо использовать формулу, как при 16-ти битном сэмплировании.

1.3.9.3.4.6 Передача данных

Передаваемые данные записываются в FIFO передатчика через интерфейс APB. Затем посылки поочередно отправляются в синхронный или асинхронный (в зависимости от значения бита SME регистра USART_SYNC) передатчик, где дополняются параметрами, указанными в регистре USART_CTRL (стартовый бит, опционально бит чётности, 1 или 2 стоповых бита для асинхронного режима). Затем данные передаются по линии TXD (по линии TXD передаётся только то количество бит данных, которое указано в регистре USART_CTRL).

Биты регистра USART_CTRL можно изменять только когда USART контроллер выключен или одновременно с включением.

Формат записи данных в FIFO передатчика представлен на рис. 1.38:

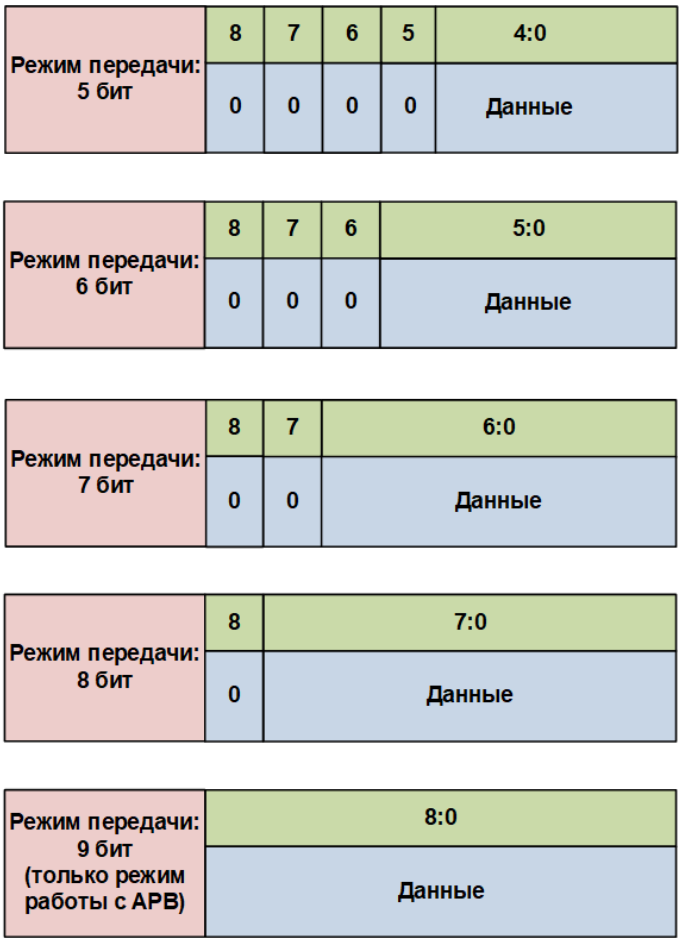


Рисунок 1.38 – Формат данных для передачи

Данные для передачи необходимо записывать в регистр USART_TXDATA. Младшие 9 бит регистра USART_TXDATA записываются в соответствии с приведённым выше форматом, а в остальные старшие биты должно записываться нулевое значение.

Для отслеживания занятости передающего FIFO используются USART_FIFO_STATE и USART_FIFOWM регистры. USART_FIFO_STATE отображает текущее количество слов в TRANSMIT FIFO. В USART_FIFOWM задается уровень опустошения FIFO на передачу, после которого будет выставлен статусный бит или сгенерировано прерывание.

Значение бита чётности зависит от значения битов PEN, EPS и SPS регистра USART_CTRL согласно таблице 1.238.

Таблица 1.238 – Значение бита чётности в зависимости от значений битов регистра USART_CTRL

PEN	EPS	SPS	Бит чётности
0	x	x	Не передается

PEN	EPS	SPS	Бит чётности
1	1	0	Проверка на нечётность
1	0	0	Проверка на чётность
1	0	1	0 (space)
1	1	1	1 (mark)

1.3.9.3.4.7 Приём данных в асинхронном режиме (UART)

Если в регистре USART_CTRL установлен бит UEN и включен асинхронный режим работы (USART_SYNC[SME] = 0), то когда приёмник находится в режиме простоя (сигнал RXD продолжительное время равен 1), и на входе данных обнаруживается 0 (получен стартовый бит), запускается счётчик приёмника и начинается приём данных.

USART поддерживает 3 схемы сэмплирования данных:

– 16-ти битное;

– 8-ми битное;

– однобитное.

На рис. 1.39 приведены примеры сэмплирования в первых двух режимах. По средним трем точкам с помощью схемы мажоритирования определяется значение принимаемого бита и наличие помехи (см. таблицу 1.239). Флаг помехи выставляется, когда в трех точках мажоритирования обнаруживаются разные значения. Стартовый бит начинается при обнаружении среза входного сигнала. Затем, как и для всех остальных битов запускается схема сэмплирования и нахождения помехи. Однако схема мажоритирования работает при каждом сэмпле, а не только в середине. Если эта схема выдаст единицу во время стартового бита, то после этого прекращается прием и контроллер переходит в режим ожидания нового стартового бита. Если же схема всегда выдает нуль или нуль с наличием помехи, то это будет считаться корректным стартовым битом и прием остальных данных продолжится.

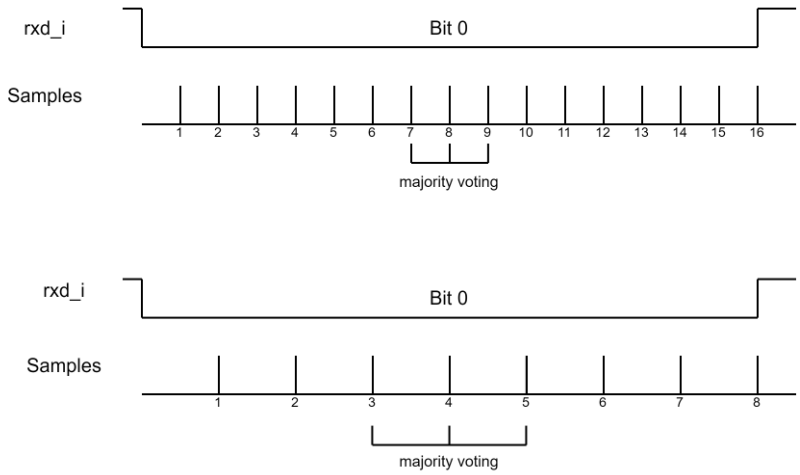


Рисунок 1.39 – Сэмплирование в 16-ти и 8-ми битных режимах

Таблица 1.239 – Схема мажоритирования

Значения в точках сэмплирования (7, 8, 9 или 3, 4, 5)	**Принимаемый бит**	**Флаг помехи**
000	0	0
001	0	1
010	0	1
011	1	1
100	0	1
101	1	1
110	1	1
111	1	0

На рис. 1.40 приведен пример сэмплирования в однобитном режиме при USART_BDIV[DIV] равном 5. Для данного режима не используется схема мажоритирования, а значение принимаемого бита определяется по одной средней точке. В данном режиме никогда не выставляется флаг помехи. Начало стартового бита определяется по переходу входного сигнала из единицы в нуль. В течение всего стартового бита проверяется, что входной сигнал всегда равен нулю. Если же это не так, то прием стартового бита прекращается и контроллер переходит в режим ожидания нового стартового бита.



Рисунок 1.40 – Сэмплирование в однобитном режиме.

Бит чётности проверяется, если данная опция задана в регистре USART_CTRL. В завершение, действительный стоповый бит подтверждается, если сигнал RXD находится в состоянии 1, в противном случае имеет место быть ошибка стопового бита. Когда получено полное слово, данные помещаются в FIFO приёмника. Затем данные передаются по шине AMBA APB:

Режим выгрузки данных – data and flags – режим, где каждый символ в дальнейшем выгружается со своим статусным регистром. Фактически, отводятся 4 бита для указания индикации обрыва линии, ошибки чётности, ошибки стопового бита и ошибки переполнения FIFO приёмника (в этом случае возможна потеря данных). Выгружаемая структура представлена на рис. 1.41

15	14	13	12	11	10	9	8:0
Ошибка переполнения FIFO приемника	Индикация обрыва линии	Ошибка четности	Ошибка стопового бита	Принят Break символ	Ошибка помехи	Резерв	Принимаемые данне

Рисунок 1.41 – Структура выгружаемых данных при режиме выгрузки data and flags.

Режим выгрузки данных – data only – режим, в котором символы выгружаются без флагов об ошибках (при этом в регистре USART_STATUS ошибки фиксироваться будут). Выгружаемая структура данных представлена на рис. 1.42.

15:9	8:0
Резерв	Принимаемые данне

Рисунок 1.42 – Структура выгружаемых данных при режиме выгрузки data only.

Данные, полученные по USART и сохраненные в приемном FIFO, необходимо считывать из регистров USART_RXDATA. Младшие два байта регистра USART_RXDATA устанавливаются в соответствии с приведённым выше форматом, а в оставшихся старших байтах устанавливается нулевое значение.

Для отслеживания занятости приемного FIFO используются USART_FIFO_STATE и USART_FIFOWM регистры. USART_FIFO_STATE отображает текущее количество слов в RECEIVE FIFO. В USART_FIFOWM задается уровень наполнения FIFO на прием, после которого будет выставлен статусный бит или сгенерировано прерывание.

Для поддержки работы с другими блоками используются два выходных сигнала tx_watermark_o и rx_watermark_o. Они информируют блоки об опустошении или превышении watermark в одном из FIFO. Выходной сигнал tx_watermark_o находится в "1" пока в TX FIFO находится меньше или равное кол-во данных, чем указано в TX watermark. Выходной сигнал

rx_watermark_o находится в "1" пока в RX FIFO находится больше данных, чем указано в watermark.

С помощью регистра USART_TIMEOUT можно установить время таймаута на прием. В асинхронном режиме работы задаётся в битовых интервалах. Если в заданный промежуток асинхронный приёмник не получит стартовый бит, то сработает прерывание по таймауту приёмника и будет выставлена соответствующая ошибка в бите USART_STATUS[RTRIS]. При нулевом значении регистра USART_TIMEOUT функция таймаута выключена. Также таймаут выключен, когда выключен USART контроллер или в режиме RS-232, когда RTS сигнал переходит в активное состояние. Значение в данном регистре можно изменять только, когда контроллер выключен.

1.3.9.3.4.8 Автоматическое определение baudrate и корректировка фазы

Автоматическое определение baudrate происходит на основе длины стартового бита. Для этого необходимо, что бы после него шел бит "1".

Автоматическое определение baudrate включается с помощью бита USART_BDIV[AUTO_BR_DETECT]. Пока этот бит установлен происходит определение baudrate. Во время определения baudrate происходит определение только целой часть делителя.

Baudrate определяется в течение стартового бита. После прихода среза на вход RXD начинается инкрементирование USART_BDIV[DIV] или каждый, или каждые 16, или каждые 8 тактов пока не будет обнаружен фронт на RXD, то есть начало бита "1". Количество тактов определяется режимом сэмплирования.

Корректировка фазы заключается в сбросе счетчика в BAUDRATE GENERATOR в USART_BDIV[DIV] значение. Это происходит каждый раз при начале стартового бита.

Корректировка фазы представлена на рисунке 1.43.

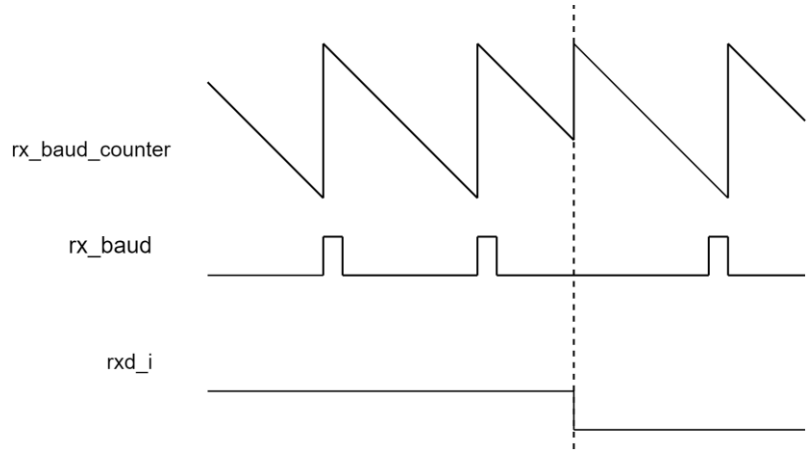


Рисунок 1.43 – Корректировка фазы

1.3.9.3.4.9 Синхронный режим

Синхронный режим включается с помощью бита USART_SYNC[SME].

В синхронном режиме данные передаются или принимаются в соответствии с синхросигналом XCK. Всего возможно четыре комбинации соотношения фазы и уровня импульсов XCK по отношению к сигналам данных. Эти комбинации можно задать с помощью бита управления фазой синхронизации (CPHA) и бита разрешения инвертированного ввода-вывода (CPOL) в регистре USART_SYNC.

Синхронный режим имитирует принцип работы Motorola SPI интерфейса. В этом интерфейсе есть одно ведущее (master) устройство и одно или несколько ведомых (slave). Master генерирует опорный синхросигнал для ведомых и из-за этого только он может начать передачу данных. Данный интерфейс предлагает использование еще одного дополнительного сигнала SS (Slave Select). С его помощью master сигнализирует о начале передачи символа, а slave начинает прием и передачу, если TXFIFO не пустое. Между последовательной передачей символов сигнал SS переходит в неактивное состояние на один период XCK.

Изменения любых битов регистра USART_SYNC во время приёма или передачи приведёт к нарушению работы как приёмника, так и передатчика. Изменения в данный регистр следует записывать только тогда, когда USART выключен(бит UEN равен 0).

Максимально возможная скорость передачи в синхронном режиме составляет $Baudrate = \frac{f}{16}$.

1.3.9.3.4.9.1 Режим master

В данном режиме USART контроллер выступает ведущим на шине USART, для этого бит XDIR должен быть равен единице и включен стандарт RS-485 (см. таблицу 1.240). Для передачи данных он генерирует тактовый сигнал на выходе XCK. Одновременно с передачей всегда происходит и прием (см. рис. 1.44). Это означает, что для приема необходимо передать какой-либо символ.

Одновременный прием и передача реализованы путем внутреннего дублирования сигнала XCK в блок SYNCHRONOUS RECEIVER.

Частота выходного синхросигнала определяется по той же формуле, что и для асинхронного 16-ти битного сэмплирования.

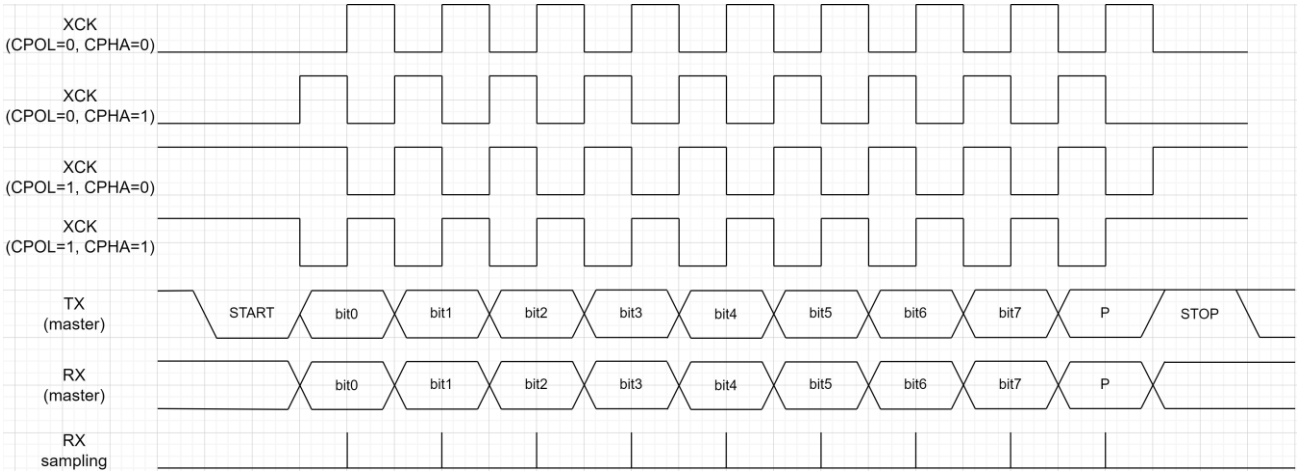


Рисунок 1.44 – Установка и выборка данных в синхронном master режиме USART.

Выход RTS используется для генерации сигнала SS. Он переходит в активное состояние одновременно с выставлением на линию TX стартового (на самом деле на один такт раньше CLK) и возвращается в неактивное состояние после окончания передачи стопового бита. CTS не используется(всегда находится в активном состоянии, SYNCHRONOUS RECEIVER постоянно принимает входящие символы).

1.3.9.3.4.9.2 Режим slave

В данном режиме USART контроллер выступает ведомым устройством на шине USART, бит XDIR равен нулю включен стандарт RS-232 (см. таблицу 1.240). Это означает, что XCK является входом через который поступают импульсы синхронизации. При приеме символа может также происходить и передача. Передача происходит только, если передающее FIFO не пустое.

В режиме RS-232 сигнал на выходе RTS генерирует блок SYNCHRONOUS TRANSMITTER. Он выставляется на время передачи всего символа. CTS вход используется, в качестве slave select. Когда он переходит в активное состояние, то это означает, что master начинает передачу. Если CPHA = 0, то USART контроллер выставляет первый бит на шину сразу как только прочитан символ из передающего FIFO и CTS находится в нуле, однако это происходит только, если не было фронта(CPOL=0) или среза(CPOL=1) на XCK.

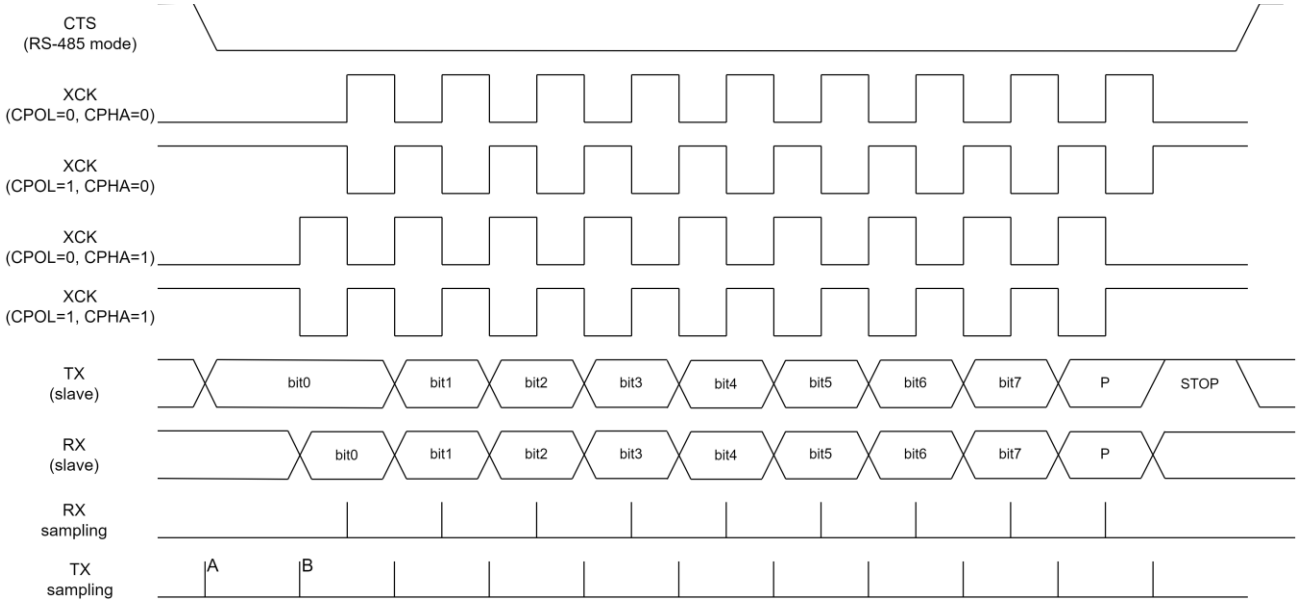


Рисунок 1.45 – Установка и выборка данных в синхронном slave режиме USART.

На рис. 1.45 в момент времени A TX sampling происходит только когда CPHA = 0 и CTS = 0 и передающего FIFO уже считан символ, но тогда не происходит TX sampling в момент времени B. Если же CPHA = 1, то все наоборот, TX sampling происходит только в момент времени B.

Если вход CTS не используется, то его можно занулить с помощью бита SS в регистре USART_SYNC. Однако эта возможность должна быть включена только, когда CPHA = 1.

При передаче возникает задержка между XCK и выдаваемыми данными на TX. Она может составлять до 5 тактов опорной частоты USART контроллера.

В синхронном режиме функция таймаута не поддерживается, поэтому в регистре USART_TIMEOUT может находится любое значение.

1.3.9.3.4.10 Фильтрация принимаемых символов

Для фильтрации принимаемых символов используется четыре побитовых фильтра с масками. Фильтры применяются последовательно. То есть если был принят символ

подходящий к фильтру **A** с его маской, то для следующего символа будет применяться фильтр **B**. Если же символ не подходит по фильтру **A**, то он не будет записан в приемное FIFO. После того как пришел символ, который подходит под фильтр **D**, фильтрация переходит в режим "прием всего", в котором принимаются все символы. Для того, что бы включить фильтрацию, необходимо обновить любой из фильтров. В LIN режиме это происходит автоматически после прихода break символа или так же при обновлении регистров фильтров.

1.3.9.3.4.11 LIN режим

Данный режим служит для поддержки протокола LIN2.1. В нем каждый кадр состоит из следующих частей:

– BREAK символ. Выход удерживается в нуле в течение 13 битовых интервалов;

– SYNC. Байт со значением 0x55;

– ID. Байт состоящий из шести битного id и двух защитных бит;

– DATA. Получаемые или отправляемые данные;

– CHEKSUM.

Для поддержки LIN протокола USART контроллер может принимать и отправлять BREAK символы, а остальная часть протокола реализуется программно.

Для включение LIN режима необходимо:

– установить бит USART_CTRL[LIN]

– включить USART в асинхронном режиме

– включить режим data and flags

– выбрать длину символа 8 бит (либо только 8 бит, либо 7 бит и бит четности).

Например, это можно сделать следующей настройкой:

– USART_SYNC[SME] = 0. Асинхронный режим

– USART_CTRL[LIN] = 1. Включение режима LIN

– USART_CTRL[DUM] = 0. Режим data and flags

– USART_CTRL[WLEN] = 3b011. Длина символа 8 бит

– USART_CTRL[STP] = 2b00. Один стоповый бит

– USART_CTRL[PEN] = 0. Передача бита чётности запрещена

– USART_CTRL[UEN] = 1. Включение контроллера USART

Для отправки break символа необходимо установить девятый бит в отправляемом символе, записав в регистр USART_TXDATA 0x100 значение. Запрос на отправку break символа запишется в общую очередь в FIFO.

После того как был принят корректный break символ, то в приемное FIFO запишется нулевой символ, а поле флагов установится флаг BC.

Корректным break символом является от 11 до 13 нулевых бит и один разделительный (единица). Если приходит 9, а за ними единичный бит, то в приемном FIFO сохранится 0x0000. Если приходит 10 нулевых бит, а за ним единичный, то в приемном FIFO сохранится 0x1000, то есть выставится флаг ошибки стопового бита. Если приходит больше 13 нулевых бит, то тогда в приемном FIFO сохранится 0x5000. То есть ошибка стопового бита и обрыв линии.

С помощью регистра USART_BREAK_LEN можно настроить длину отравляемого break символа, а так же максимально допустимую длину break при приеме (см. таблицу 1.252).

Все отправленные данные также принимаются в приемное FIFO.

1.3.9.3.4.12 Биты ошибок

При приёме данных пять битов ошибок помещаются в область [15:12] и [10] FIFO приёмника:

– Ошибка помехи [10]

Данная ошибка возникает, когда схемой мажоритирования во время приема бита было найдено несоответствие между тремя сэмплами.

– Ошибка стопового бита [12]

Данная ошибка возникает, если при приёме данных не был подтвержден хотя бы один стоповый бит (сигнал RXD не находился в состоянии 1). Возникновение данной ошибки возможно только при работе в асинхронном режиме.

– Ошибка чётности [13]

Данная ошибка возникает, если чётность полученных данных не совпадает с чётностью, заданной в битах EPS и SPS регистра USART_CTRL.

– Ошибка обрыва линии [14]

Данная ошибка возникает, когда обнаруживается условие обрыва, что означает, что на вход подается 0 продолжительностью более времени передачи полного слова (стартовый бит, данные, бит чётности, стоповые биты). Возникновение данной ошибки возможно только при работе в асинхронном режиме.

– Ошибка переполнения [15]

В 16 бите FIFO приёмника помещается ошибка, указывающая на ошибку переполнения. Бит переполнения не относится к какому-либо символу в FIFO приёмника, а устанавливается, когда FIFO заполнен и следующий символ полностью пришел в сдвиговый регистр. Данные в

сдвиговом регистре перезаписываются, но не записываются в FIFO. Как только в FIFO приёмника появляется свободная область и принимается другой символ, значение бита переполнения копируется в FIFO приёмника вместе с полученным символом. Затем состояние переполнения сбрасывается. Структура элемента очереди FIFO приёмника представлена на рис. 1.41.

1.3.9.3.4.13 Режим петли (loopback)

Режим петли включается при установке бита LBE регистра USART_CTRL в 1. В данном режиме данные передаваемые по линии TXD принимаются линией RXD, выходной сигнал RTS замыкается на входной сигнал CTS, а выходной сигнал синхронизации передатчика замыкается на входной сигнал синхронизации приёмника. Данный режим позволяет провести петлевое тестирование данных USART.

1.3.9.3.4.14 Управление потоком данных посредством сигналов RTS/CTS

Стандарт функционирования USART может быть выбран путем установки битов RTSen и CTSen в соответствующие значения в регистре USART_CTRL. Все стандарты функционирования работают равнозначно в синхронном и в асинхронном режимах работы. Выбор полярности сигнала RTS выбирается с помощью бита POL регистра USART_CTRL. Выбор поддерживаемого стандарта описан в таблице 1.240:

Таблица 1.240 – Выбор режима функционирования USART

Стандарт	**CTSen**	**RTSen**
RS-422	0	0
RS-232	1	1
RS-485	0	1

1.3.9.3.4.14.1 Стандарт RS-232

Для поддержки стандарта RS-232 помимо линий RX и TX также используются линии RTS и CTS. С их помощью можно организовать аппаратное управление потоком данных. Данный вид управления может использоваться некоторыми медленными устройствами или устройством с простой схемной реализацией.

Передатчик перед отправкой посылки проверяет вход CTS. Если сигнал CTS находится в состоянии 0, то передача проходит, в противном случае – нет (active low). Если данный сигнал переходит в 1 во время пересылки, то текущая передача будет завершена до конца, а следующая передача, начнется только, после того как CTS перейдет обратно в 0.

Приёмник устанавливает 0 на выходе RTS, если он готов принимать данные (в приемном FIFO есть свободное место), и устанавливает 1, если от передатчика требуется остановить передачу (active low). В этом случае прерывание по тайм-ауту срабатывать не будет, а при возвращении RTS обратно в 0, отсчёт тайм-аута начнется заново.

1.3.9.3.4.14.2 Стандарт RS-485

Для поддержки стандарта RS-485 сигнал RTS используется для управления выдачей данных на линию с возможностью программирования полярности RTS (указывается в поле POL регистра USART_CTRL, по умолчанию – active low). Сигнал CTS для поддержки RS-485 не используется.

1.3.9.3.4.14.3 Стандарт RS-422

Для поддержки стандарта RS-422 сигналы RTS и CTS не используются.

1.3.9.3.4.15 Прерывания

В USART генерируется уровневое прерывание USART_IRQ, представляющее собой объединение по ИЛИ событий прерываний, перечисленных в регистре USART_STATUS. Если на возникшее событие прерывания регистра USART_STATUS установлена маска в регистре USART_MASK (бит маски соответствующего прерывания равен 0), то прерывание USART_IRQ по данному событию генерироваться не будет. После сброса все прерывания замаскированы.

Сброс событий прерываний USART производится побитово путем записи единиц в соответствующие биты регистра USART_CLEAR.

Тестовое прерывание TEST устанавливается после записи единицы в регистр USART_TEST, а сбрасывается так же, как и все остальные через USART_CLEAR.

1.3.9.3.4.16 Интеграция USART

Контроллер разделен на два тактовых домена: apb-домен и core-домен. Это сделано для того, что бы при изменении системной частоты core-домен, выполняющий передачу данных на шине, не надо было перенастраивать.

У контроллера на входе стоит специальный блок gp_apb_rejector. Его назначение заключается в корректном завершении apb-транзакций, пока с USART контроллера снято тактирование. В зависимости от входа conf_pslverr_i транзакции могут завершаться с ошибкой или без.

Если conf_pslverr_i = 1, то генерируется ошибка. При этом все транзакции записи игнорируются, а чтение возвращает 0x0BAD_ACCE5.

При обращении к USART по шине APB младшие два бита (нулевой и первый) адреса не анализируются. APB транзакция на запись с pstrb не равным 4'b1111 возвращается с ошибкой (pslverr = 1) и никакие данные не записываются. Запись по зарезервированным адресам игнорируется, а чтение возвращает 0xEEFFADDC. При этом транзакции завершаются без ошибок. Так как младшие два бита не анализируются, то обращения по не выровненным адресам автоматически выравниваются.

1.3.9.3.5 Описание регистров

1.3.9.3.5.1 Общее описание регистров

Контроллер имеет набор программно доступных регистров, требующий 4 КБайт адресов на системной шине APB.

Поддерживается только 32-х разрядный доступ. Байтовая или двух-байтовая запись будет проигнорирована и возвращена ошибка шины.

Карта регистров приведена в таблице 1.241.

Адреса, который не указаны в таблице 1.241 считаются резервными. Чтение из них возвращает 0xEEFFADDC, а запись игнорируется. USART контроллер не различает secure и non-secure транзакции.

1.3.9.3.5.2 Карта регистров

Таблица 1.241 – Карта регистров USART

Адрес	Название	Доступ	Значение после сброса	Описание
0x000	USART_ID	ЧТ	0x54525355	Идентификатор устройства
0x004	USART_VERSION	ЧТ	0x0100	Номер версии устройства и конфигурация
0x008	USART_SWRST	ЧТ/ЗП	0x0	Программный сброс
0x00C	USART_BDIV	ЧТ/ЗП	0x400	Значение делителя скорости передачи.
0x010	USART_CTRL	ЧТ/ЗП	0x0	Управляющие биты и информация о режимах приёма и передачи данных
0x014	USART_STATUS	ЧТ	0x0	Возвращение текущего необработанного значения состояния перед маскированием соответствующего прерывания
0x018	USART_MASK	ЧТ/ЗП	0x0	Устанавливает маску прерываний
0x01C	USART_CLEAR	ЗП	0x0	Побитовый сброс прерываний
0x020	USART_TEST	ЗП	0x0	Установка прерывания TEST

Адрес	Название	Доступ	Значение после сброса	Описание
0x024	USART_FIFOWM	ЧТ/ЗП	см. описание регистра	Информация о выбранной границе уровня заполнения FIFO приёмника или передатчика
0x028	USART_FIFO_STATE	ЧТ	0x0	Информация о состоянии FIFO приёмника, FIFO передатчика и разрешения на отправку данных
0x02C	USART_SYNC	ЧТ/ЗП	0x0	Включение и настройка синхронного режима работы USART
0x030	USART_TIMEOUT	ЧТ/ЗП	0x0	Установка тайм-аута, при превышении которого при приёме данных возникает соответствующее прерывание
0x038	USART_BREAK_LEN	ЧТ/ЗП	0x2	Длина отправляемого и максимальная длина принимаемого break символа
0x03C	-	ЧТ	-	Зарезервировано
0x040 - 0x040+4	USART_TXDATA	ЧТ/ЗП	0x0	Отправляемые данные
0x050 - 0x050+4	USART_RXDATA	ЧТ	0x0	Получаемые данные
0x060	USART_FILTER_A	ЧТ/ЗП	0x0	Фильтр A
0x064	USART_FILTER_B	ЧТ/ЗП	0x0	Фильтр B
0x068	USART_FILTER_C	ЧТ/ЗП	0x0	Фильтр C
0x06C	USART_FILTER_D	ЧТ/ЗП	0x0	Фильтр D
0x070	USART_FILTER_MASK_A	ЧТ/ЗП	0x0	Маска фильтра A
0x074	USART_FILTER_MASK_B	ЧТ/ЗП	0x0	Маска фильтра B
0x078	USART_FILTER_MASK_C	ЧТ/ЗП	0x0	Маска фильтра C
0x07C	USART_FILTER_MASK_D	ЧТ/ЗП	0x0	Маска фильтра D

1.3.9.3.5.3 Описание полей регистров

1.3.9.3.5.3.1 USART_ID (0x000)

Регистр идентификатора USART контроллера

В таблице 1.242 представлено описание полей регистра.

Таблица 1.242 – Описание полей регистра USART_ID

Биты	Поле	Доступ	Значение после сброса	Описание
31:0	ID	ЧТ	0x54525355	Идентификатор контроллера LCD

1.3.9.3.5.3.2 USART_VERSION (0x004)

Регистр версии USART контроллера

В таблице 1.243 представлено описание полей регистра.

Таблица 1.243 – Описание полей регистра USART_VERSION

Биты	Поле	Доступ	Значение после сброса	Описание
7:0	SUBVERSION	ЧТ	0	Номер подверсии устройства
15:8	VERSION	ЧТ	1	Номер версии устройства
31:16	-	ЧТ	0	Зарезервировано

1.3.9.3.5.3.3 USART_SWRST (0x008)

Регистр программного сброса USART контроллера

В таблице 1.244 представлено описание полей регистра.

Таблица 1.244 – Описание полей регистра USART_SWRST

Биты	Поле	Доступ	Значение после сброса	Описание
0	SWRST	ЧТ/ЗП	0	0 - программный сброс завершен. 1 - установка программного сброса. Сброс данного бита происходит автоматически по завершению программного сброса.
31:1	-	ЧТ	0	Зарезервировано

1.3.9.3.5.3.4 USART_TXDATA (0x040)

Регистр данных отправки USART контроллера

В таблице 1.245 представлено описание полей регистра.

Таблица 1.245 – Описание полей регистра USART_TXDATA

Биты	Поле	Доступ	Значение после сброса	Описание
8:0	TXDATA	ЧТ/ЗП	0	Отправляемые данные Если включен режим LIN, то если восьмой бит (девятый по счету) установлен в единицу, USART отправит break символ.
31:9	-	ЧТ	0	Зарезервировано

1.3.9.3.5.3.5 USART_RXDATA (0x050)

Регистр принятых данных USART контроллера

В таблице 1.246 представлено описание полей регистра.

Таблица 1.246 – Описание полей регистра USART_RXDATA

Биты	Поле	Доступ	Значение после сброса	Описание
8:0	RXDATA	ЧТ	0	Принимаемые данные
9	-	ЧТ	0	Зарезервировано

Биты	Поле	Доступ	Значение после сброса	Описание
10	NE	ЧТ	0	Во время приема была обнаружена помеха
11	BC	ЧТ	0	Принят break символ
12	FE	ЧТ	0	Ошибка стопового бита. Устанавливается в 1, если в полученном символе отсутствует корректный стоповый бит (корректный стоповый бит – 1). Установка данного бита возможна только в асинхронном режиме работы.
13	PE	ЧТ	0	Ошибка чётности. Устанавливается в 1, когда чётность полученных данных не совпадает с чётностью, указанной в битах EPS и SPS регистра USART_CTRL.
14	BE	ЧТ	0	Индикация обрыва линии. Устанавливается в 1, когда обнаруживается условие обрыва, что означает, что на вход подается 0 продолжительностью более времени передачи полного слова (стартовый бит, данные, бит чётности, стоповые биты). Установка данного бита возможна только в асинхронном режиме работы.
15	OE	ЧТ	0	Ошибка переполнения. Устанавливается в 1, если данные получены, когда FIFO приёмника уже заполнено. Как только в FIFO появляется свободное пространство, данный бит сбрасывается в 0.
31:16	-	ЧТ	0	Зарезервировано

1.3.9.3.5.3.6 USART_FIFO_STATE (0x028)

Регистр статуса FIFO USART контроллера

В таблице 1.247 представлено описание полей регистра.

Таблица 1.247 – Описание полей регистра USART_FIFO_STATE

Биты	Поле	Доступ	Значение после сброса	Описание
10:0	RXFS	ЧТ	0	Количество слов в FIFO приёмника.
15:11	-	ЧТ	0	Зарезервировано
26:16	TXFS	ЧТ	0	Количество слов в FIFO передатчика.
31:27	-	ЧТ	0	Зарезервировано

1.3.9.3.5.3.7 USART_BDIV (0x00C)

Регистр настройки делителей USART контроллера

В таблице 1.248 представлено описание полей регистра.

Таблица 1.248 – Описание полей регистра USART_BDIV

Биты	Поле	Доступ	Значение после сброса	Описание
7:0	DIVF	ЧТ/ЗП	0	Значение дробного делителя скорости приема/передачи. Данное поле можно изменять только, когда контроллер выключен.
23:8	DIV	ЧТ/ЗП	0x4	Значение делителя скорости приема/передачи. Данное поле можно изменять только, когда контроллер выключен.
27:24	-	ЧТ	0	Зарезервировано
28	AUTO_BR_DETECT	ЧТ/ЗП	0	Автоматическое определение скорости приема/передачи. После установки бита в единицу и если USART включен, то начинается определение скорости приема/передачи. После окончания определения данный бит сбрасывается. Данное поле можно изменять только, когда контроллер выключен.
31:29	-	ЧТ	0	Зарезервировано

1.3.9.3.5.3.8 USART_CTRL (0x010)

Регистр настройки USART контроллера

В таблице 1.249 представлено описание полей регистра.

Таблица 1.249 – Описание полей регистра USART_CTRL

Биты	Поле	Доступ	Значение после сброса	Описание
0	UEN	ЧТ/ЗП	0	Доступность USART: 0 - USART выключен 1 - USART включен
1	LBE	ЧТ/ЗП	0	Включение режима петли. Если данный бит равен 1, сигнал TXD передает данные на сигнал RXD, сигнал RTS замыкается на сигнал CTS, а выход синхросигнала синхронного передатчика XCK замыкается на вход синхросигнала синхронного приёмника.
2	-	ЧТ	0	Зарезервировано
3	RTSen	ЧТ/ЗП	0	Устанавливается в 1, если включен режим RTS
4	CTSen	ЧТ/ЗП	0	Устанавливается в 1, если включен режим CTS
5	POL	ЧТ/ЗП	0	Выбор полярности сигнала RTS 0: активный режим равен 0 (active low) 1: активный режим равен 1 (active high)
6	PEN	ЧТ/ЗП	0	Разрешение передачи бита чётности: 0: передача бита чётности запрещена 1: передача бита чётности разрешена
7	EPS	ЧТ/ЗП	0	Выбор метода контроля чётности: 0 - контроль на чётность 1 - контроль на нечётность

Биты	Поле	Доступ	Значение после сброса	Описание
8	SPS	ЧТ/ЗП	0	Выбор способа формирования бита чётности: 0 - бит чётности формируется на основе передаваемых данных 1 - бит чётности принимает фиксированное значение: EPS = 1, тогда бит чётности передается и проверяется как 1 EPS = 0, тогда бит чётности передается и проверяется как 0
9	-	ЧТ	0	Зарезервировано
12:10	WLEN	ЧТ/ЗП	0	Длина слова: b101 : b111 - Зарезервировано b100 - 9 бит b011 - 8 бит b010 - 7 бит b001 - 6 бит b000 - 5 бит
13	DUM	ЧТ/ЗП	0	Определяет режим выгрузки данных. 0: режим data and flags 1: режим data only
15:14	STP	ЧТ/ЗП	0	Выбор количества стоповых бит (только для режима data and flags): 00 - 1 бит 01 - 2 бита 10 - Зарезервировано 11 - Зарезервировано
16	LIN	ЧТ/ЗП	0	Включение режима LIN. Если USART работает в асинхронном режиме, то включение LIN позволяет отправлять и принимать break символы. 0 - LIN режим выключен. 1 - LIN режим включен.
17	TXBRX	ЧТ/ЗП	0	Transmitting block receiving. Блокировка приема символов, во время их отправки. Когда USART подключен к общей шине, как в LIN интерфейсе, то USART контроллер будет принимать свои же отправленные символы. Это можно запретить с помощью этого бита. 0: USART контроллер принимает все входящие символы. 1: USART контроллер не принимает символы, которые сам же и отправляет. Данная функция работает только в асинхронном режиме.
19:18	SM	ЧТ/ЗП	0	Sampling mode Выбор режима сэмплирования. 00 - сэмплирование по 16 точкам. 01 - сэмплирование по по 8 точкам. 10 - сэмплирование по середине бита. 11 - Зарезервировано.
31:20	-	ЧТ	0	Зарезервировано

1.3.9.3.5.3.9 USART_SYNC (0x02C)

Регистр настройки синхронного режима USART контроллера

В таблице 1.250 представлено описание полей регистра.

Таблица 1.250 – Описание полей регистра USART_SYNC

Биты	Поле	Доступ	Значение после сброса	Описание
0	SME	ЧТ/ЗП	0	Включение синхронного режима работы USART: 0: синхронный режим выключен 1: синхронный режим включен
1	XDIR	ЧТ/ЗП	0	Выбор входа/выхода для синхросигнала XCK. 0 - USART контроллер выступает ведомым (slave) устройством на шине. XCK является входом. 1 - USART контроллер выступает ведущим (master) устройством на шине. XCK является выходом.
2	CPOL	ЧТ/ЗП	0	Бит разрешения инвертированного ввода-вывода.
3	CPHA	ЧТ/ЗП	0	Бит выбора фазы сигнала синхронизации.
4	SS	ЧТ/ЗП	0	Бит программной установки CTS в нуль. 0 - CTS работает как вход. 1 - CTS всегда равен нулю.
31:5	-	ЧТ	0	Зарезервировано

1.3.9.3.5.3.10 USART_FIFOWM (0x024)

Регистр настройки заполненности FIFO при которой будут генерироваться прерывания.

В таблице 1.251 представлено описание полей регистра.

Таблица 1.251 – Описание полей регистра USART_FIFOWM

Биты	Поле	Доступ	Значение после сброса	Описание
10:0	RXFW	ЧТ/ЗП	0	Receive fifo watermark. Когда количество слов в приемном буфере становится больше, указанного в данном поле, то выставляется флаг "заполнения до заданного уровня" RXRIS. После сброса значение этого поля равно половине размера приемного буфера.
15:11	-	ЧТ	0	Зарезервировано
26:16	TXFW	ЧТ/ЗП	0	Transmit fifo watermark. Когда количество слов в передающем буфере становится меньше или равно, указанного в данном поле, то выставляется флаг "опустошения до заданного уровня" TXRIS. После сброса значение этого поля равно половине размера приемного буфера.
31:27	-	ЧТ	0	Зарезервировано

1.3.9.3.5.3.11 USART_BREAK_LEN (0x038)

Регистр настройки заполненности FIFO при которой будут генерироваться прерывания.

В таблице 1.252 представлено описание полей регистра.

Таблица 1.252 – Описание полей регистра USART_BREAK_LEN

Биты	Поле	До-ступ	Значение по-сле сброса	Описание
2:0	TX_BREAK_LEN	ЧТ/ЗП	0x2	Длина отправляемого break символа. 000 - 11 бит. 001 - 12 бит. 010 - 13 бит. 011 - 14 бит. 100 - 15 бит. 101 - 16 бит. 110 - 17 бит. 111 - 18 бит.
15:3	-	ЧТ	0	Зарезервировано
18:16	RX_BREAK_LEN	ЧТ/ЗП	0	Максимальная длина принимаемого break символа. 000 - 13 бит. 001 - 14 бит. 010 - 15 бит. 011 - 16 бит. 100 - 17 бит. 101 - 18 бит. 110 - 19 бит. 111 - до бесконечности.
31:19	-	ЧТ	0	Зарезервировано

1.3.9.3.5.3.12 USART_STATUS (0x014)

Регистр статуса USART контроллера.

В таблице 1.253 представлено описание полей регистра.

Таблица 1.253 – Описание полей регистра USART_STATUS

Биты	Поле	До-ступ	Значение по-сле сброса	Описание
0	TEST	ЧТ	0	Установлено тестовое прерывание.
1	FERIS	ЧТ	0	Ошибка стопового бита.
2	PERIS	ЧТ	0	Ошибка бита чётности.
3	BERIS	ЧТ	0	Ошибка обрыва линии.
4	RERIS	ЧТ	0	Переполнение FIFO приёмника.
5	TERIS	ЧТ	0	Переполнение FIFO передатчика.
6	TXRIS	ЧТ	0	FIFO передатчика опустошилось до заданного уровня.
7	RXRIS	ЧТ	0	FIFO приёмника наполнилось до заданного уровня.

Биты	Поле	Доступ	Значение после сброса	Описание
8	RTRIS	ЧТ	0	Прерывание по тайм-ауту при получении данных. Возникает, когда во время транзакции не поступают никакие новые данные в течение запрограммированного периода.
9	TX_SENDING	ЧТ	0	Устанавливается после того, как контроллер начинает передачу. Сбрасывается программно через USART_CLEAR регистр.
10	RXBC	ЧТ	0	Был принят break символ.
31:11	-	ЧТ	0	Зарезервировано

1.3.9.3.5.3.13 USART_MASK (0x018)

Регистр маски прерываний USART контроллера.

В таблице 1.254 представлено описание полей регистра.

Таблица 1.254 – Описание полей регистра USART_MASK

Биты	Поле	Доступ	Значение после сброса	Описание
0	TESTM	ЧТ/ЗП	0	Маска TEST прерывания 0 - запрещено прерывание. 1 - разрешено прерывание.
1	FEIM	ЧТ/ЗП	0	Маска прерывания ошибки стопового бита 0 - запрещено прерывание. 1 - разрешено прерывание.
2	PEIM	ЧТ/ЗП	0	Маска прерывания ошибки бита чётности 0 - запрещено прерывание. 1 - разрешено прерывание.
3	BEIM	ЧТ/ЗП	0	Маска прерывания ошибки обрыва линии 0 - запрещено прерывание. 1 - разрешено прерывание.
4	REIM	ЧТ/ЗП	0	Маска прерывания ошибки переполнения FIFO приёмника 0 - запрещено прерывание. 1 - разрешено прерывание.
5	TEIM	ЧТ/ЗП	0	Маска прерывания ошибки переполнения FIFO передатчика 0 - запрещено прерывание. 1 - разрешено прерывание.
6	TXIM	ЧТ/ЗП	0	Маска прерывания по заполнению FIFO передатчика до заданного уровня 0 - запрещено прерывание. 1 - разрешено прерывание.
7	RXIM	ЧТ/ЗП	0	Маска прерывания по заполнению FIFO приёмника до заданного уровня 0 - запрещено прерывание. 1 - разрешено прерывание.

Биты	Поле	Доступ	Значение после сброса	Описание
8	RTIM	ЧТ/ЗП	0	Маска прерывания ошибки тайм-аута при получении данных. 0 - запрещено прерывание. 1 - разрешено прерывание.
9	TX_SENDING	ЧТ/ЗП	0	Маска TX_SENDING прерывания 0 - запрещено прерывание. 1 - разрешено прерывание.
10	RXBCIM	ЧТ/ЗП	0	Маска прерывания приема break символа 0 - запрещено прерывание. 1 - разрешено прерывание.
31:11	-	ЧТ	0	Зарезервировано

1.3.9.3.5.3.14 USART_CLEAR (0x01C)

Регистр сброса прерываний USART контроллера.

В таблице 1.255 представлено описание полей регистра.

Таблица 1.255 – Описание полей регистра USART_CLEAR

Биты	Поле	Доступ	Значение после сброса	Описание
0	TEST	ЗП	0	Запись единицы сбрасывает TEST прерывание.
1	FERIC	ЗП	0	Запись единицы сбрасывает прерывание ошибки стопового бита.
2	PERIC	ЗП	0	Запись единицы сбрасывает прерывание ошибки бита чётности.
3	BERIC	ЗП	0	Запись единицы сбрасывает прерывание ошибки обрыва линии.
4	RERIC	ЗП	0	Запись единицы сбрасывает прерывание переполнения FIFO приёмника.
5	TERIC	ЗП	0	Запись единицы сбрасывает прерывание переполнения FIFO передатчика.
6	TXRIC	ЗП	0	Запись единицы сбрасывает прерывание TXRIS.
7	RXRIC	ЗП	0	Запись единицы сбрасывает прерывание RXRIS.
8	RTRIC	ЗП	0	Запись единицы сбрасывает прерывание тайм-аута.
9	TX_SENDING	ЗП	0	Запись единицы сбрасывает прерывание TX_SENDING.
10	RXBC	ЗП	0	Запись единицы сбрасывает прерывание принятия break символа.
31:11	-	ЧТ	0	Зарезервировано

1.3.9.3.5.3.15 USART_TEST (0x020)

Регистр установки прерывания test.

В таблице 1.256 представлено описание полей регистра.

Таблица 1.256 – Описание полей регистра USART_TEST

Биты	Поле	Доступ	Значение после сброса	Описание
0	TEST	ЗП	0	Запись единицы устанавливает прерывание TEST.
31:1	-	ЧТ	0	Зарезервировано

1.3.9.3.5.3.16 USART_TIMEOUT (0x030)

Регистр установки тайм-аута USART контроллера.

В таблице 1.257 представлено описание полей регистра.

Таблица 1.257 – Описание полей регистра USART_TIMEOUT

Биты	Поле	Доступ	Значение после сброса	Описание
31:0	RTM	ЧТ/ЗП	0	Значение таймаута для синхронного и асинхронного приёмников. В асинхронном режиме работы задаётся в битовых интервалах. Если в заданный промежуток асинхронный приёмник не получит стартовый бит, то сработает прерывание по таймауту приёмника и будет выставлена соответствующая ошибка в бите USART_STATUS[RTRIS]. В синхронном режиме данный регистр игнорируется. При нулевом значении данного регистра таймаут будет игнорироваться. Отключение MUSART сбрасывает внутренний счётчик этого регистра.

1.3.9.3.5.3.17 USART_FILTER_A (0x060)

Регистр настройки фильтра A USART контроллера.

В таблице 1.258 представлено описание полей регистра.

Таблица 1.258 – Описание полей регистра USART_FILTER_A

Биты	Поле	Доступ	Значение после сброса	Описание
8:0	FILTER_A	ЧТ/ЗП	0	Фильтр A. Побитовый фильтр принимаемых символов.
31:9	-	ЧТ	0	Зарезервировано

1.3.9.3.5.3.18 USART_FILTER_B (0x064)

Регистр настройки фильтра B USART контроллера.

В таблице 1.259 представлено описание полей регистра.

Таблица 1.259 – Описание полей регистра USART_FILTER_B

Биты	Поле	До-ступ	Значение по-сле сброса	Описание
8:0	FILTER_B	ЧТ/ЗП	0	Фильтр B. Побитовый фильтр принимаемых символов.
31:9	-	ЧТ	0	Зарезервировано

1.3.9.3.5.3.19 USART_FILTER_C (0x068)

Регистр настройки фильтра C USART контроллера.

В таблице 1.260 представлено описание полей регистра.

Таблица 1.260 – Описание полей регистра USART_FILTER_C

Биты	Поле	До-ступ	Значение по-сле сброса	Описание
8:0	FILTER_C	ЧТ/ЗП	0	Фильтр C. Побитовый фильтр принимаемых символов.
31:9	-	ЧТ	0	Зарезервировано

1.3.9.3.5.3.20 USART_FILTER_D (0x06C)

Регистр настройки фильтра D USART контроллера.

В таблице 1.261 представлено описание полей регистра.

Таблица 1.261 – Описание полей регистра USART_FILTER_D

Биты	Поле	До-ступ	Значение по-сле сброса	Описание
8:0	FILTER_D	ЧТ/ЗП	0	Фильтр D. Побитовый фильтр принимаемых символов.
31:9	-	ЧТ	0	Зарезервировано

1.3.9.3.5.3.21 USART_FILTER_MASK_A (0x070)

Регистр настройки маски фильтра A USART контроллера.

В таблице 1.262 представлено описание полей регистра.

Таблица 1.262 – Описание полей регистра USART_FILTER_MASK_A

Биты	Поле	Доступ	Значение после сброса	Описание
8:0	FILTER_MASK_A	ЧТ/ЗП	0	Маска фильтра A. Маска является побитовой. Если бит маски установлен в единицу, то в принимаемых символах в этом бите будет ожидаться значение, указанное в фильтре. Если в результате логической функции: &(~(FILTER_MASK_A & (FILTER_A ^ rx_data)) будет единица, то принимаемый символ будет записан в буфер.
31:9	-	ЧТ	0	Зарезервировано

1.3.9.3.5.3.22 USART_FILTER_MASK_B (0x074)

Регистр настройки маски фильтра B USART контроллера.

В таблице 1.263 представлено описание полей регистра.

Таблица 1.263 – Описание полей регистра USART_FILTER_MASK_B

Биты	Поле	Доступ	Значение после сброса	Описание
8:0	FILTER_MASK_B	ЧТ/ЗП	0	Маска фильтра B. Маска является побитовой. Если бит маски установлен в единицу, то в принимаемых символах в этом бите будет ожидаться значение, указанное в фильтре. Если в результате логической функции: &(~(FILTER_MASK_B & (FILTER_B ^ rx_data)) будет единица, то принимаемый символ будет записан в буфер.
31:9	-	ЧТ	0	Зарезервировано

1.3.9.3.5.3.23 USART_FILTER_MASK_C (0x078)

Регистр настройки маски фильтра C USART контроллера.

В таблице 1.264 представлено описание полей регистра.

Таблица 1.264 – Описание полей регистра USART_FILTER_MASK_C

Биты	Поле	Доступ	Значение после сброса	Описание
8:0	FILTER_MASK_C	ЧТ/ЗП	0	Маска фильтра C. Маска является побитовой. Если бит маски установлен в единицу, то в принимаемых символах в этом бите будет ожидаться значение, указанное в фильтре. Если в результате логической функции: &(~(FILTER_MASK_C & (FILTER_C ^ rx_data)) будет единица, то принимаемый символ будет записан в буфер.
31:9	-	ЧТ	0	Зарезервировано

1.3.9.3.5.3.24 USART_FILTER_MASK_D (0x07C)

Регистр настройки маски фильтра D USART контроллера.

В таблице 1.265 представлено описание полей регистра.

Таблица 1.265 – Описание полей регистра USART_FILTER_MASK_D

Биты	Поле	Доступ	Значение после сброса	Описание
8:0	FILTER_MASK_D	ЧТ/ЗП	0	Маска фильтра D. Маска является побитовой. Если бит маски установлен в единицу, то в принимаемых символах в этом бите будет ожидаться значение, указанное в фильтре. Если в результате логической функции: &(~(FILTER_MASK_D & (FILTER_D ^ rx_data)) будет единица, то принимаемый символ будет записан в буфер.
31:9	-	ЧТ	0	Зарезервировано

1.3.9.4 Контроллер интерфейса I2C (I2C)

1.3.9.4.1 Характеристики контроллера I2C

Контроллер I2C предназначен для подключения микроконтроллера к внешней шине I2C. Обладает следующими характеристиками:

– поддерживаемые режимы работы:

a) master;

b) slave;

– параметры адресации:

a) 7-битная адресация;

b) 10-битная адресация;

– скорость работы:

a) 100 Кбит/с;

b) 400 Кбит/с;

– генерация маскируемых прерываний;

– работа от двух тактовых сигналов (см. 1.3.9.4.8).

1.3.9.4.2 Общее описание

Контроллер I2C представляет собой сборку двух контроллеров:

– I2C_Slave (см. 1.3.9.4.6);

– I2C_Master (см. 1.3.9.4.7);

– дополнительную управляющую схему, осуществляющую программный выбор используемого контроллера и программный сброс обоих контроллеров;

– схему отключения тактирования (CG и Rej).

Структурная схема I2C представлена на рисунке 1.46.

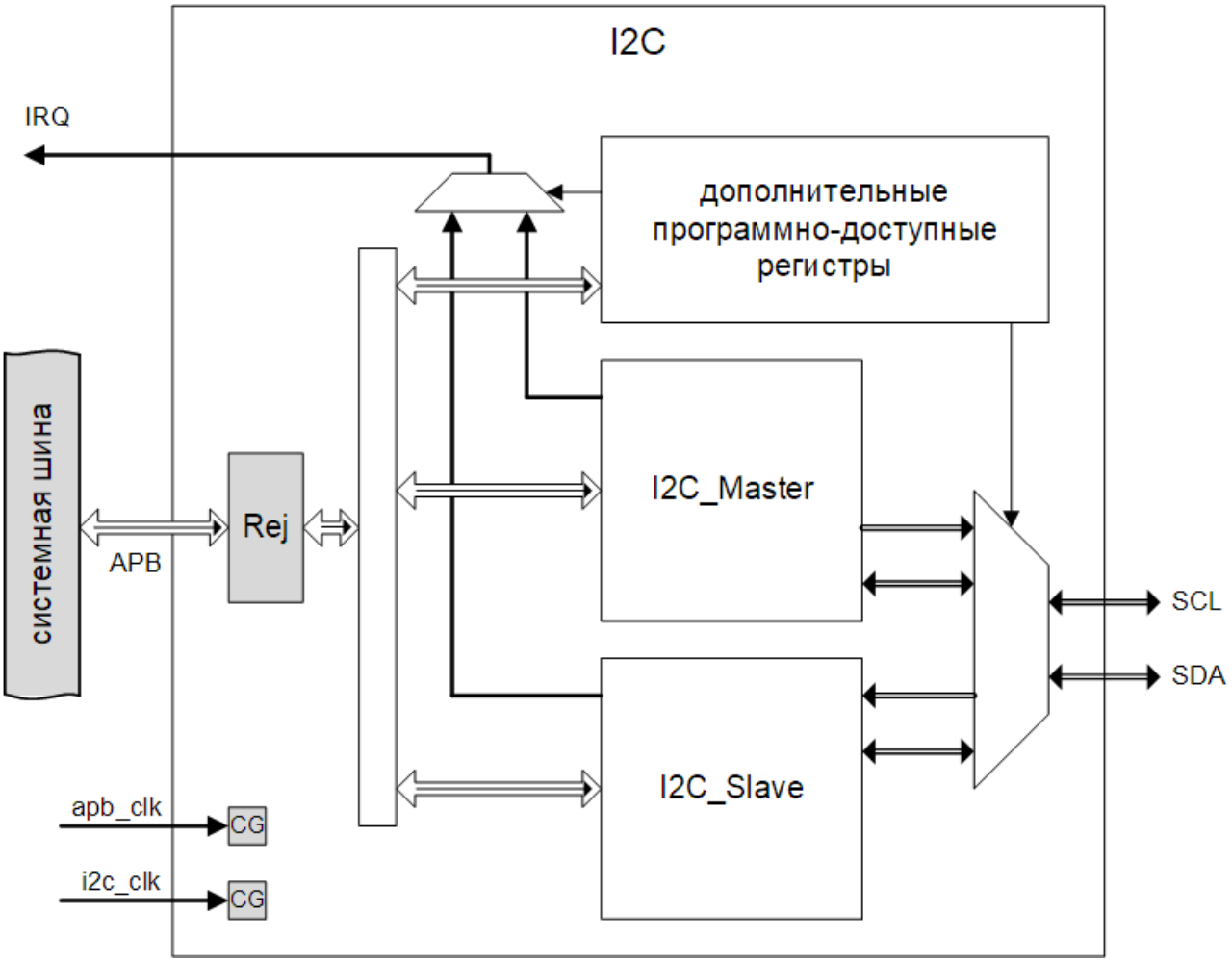


Рисунок 1.46 – Структурная схема I2C

В один момент времени может работать только один из контроллеров. Выбор осуществляется программным образом в бите I2C_MODE[SELECT] - выбранный контроллер подключается к внешним сигналам I2C, а также к сигналу прерывания.

Со стороны системной шины APB доступ есть всегда до обоих контроллеров - к каждому по своему смещению адреса (см. 1.267)

1.3.9.4.3 Описание внешних выводов контроллера I2C

Список внешних выводов контроллера I2C и их описание приведены в теблице 1.266

Таблица 1.266 – Список внешних выводов контроллера I2C

Наименование вывода	Тип (Вх/Вых)	Функциональное назначение
SDA	Вх/Вых	Двунаправленная линия для последовательного приёма/передачи данных[1].
SCL	Вх/Вых	Линия для приёма/передачи синхросигнала[1]. В режиме I2C_Master - Вых. В режиме I2C_Slave - Вх.
[1)] контроллер GPIO необходимо настроить на работу буфера в режиме открытого коллектора.		

1.3.9.4.4 Протокол передачи

I2C-шина – простая 2-х-проводная последовательная шина с многими мастерами с обнаружением коллизий и арбитражем. Шина состоит из последовательных линий данных SDA и синхросигналов SCL.

Передача по I2C-шине начинается с условия СТАРТ. Условие СТАРТ определяется как переход из высокого в низкий уровень SDA линии, в то время как на SCL – высокий уровень. Эти условия всегда генерируются мастером. Шина считается занятой после СТАРТ условия и свободна после некоторого интервала времени после СТОП условия. Требуемый интервал после СТОП до следующего СТАРТ определяется в I2C спецификации и зависит от скорости работы.

На рисунке 1.47 показана временная диаграмма транзакции по шине I2C. I2C-master сначала генерирует условие СТАРТ и затем передает 7-битный адрес I2C-slave устройства, к которому обращается. Бит $R/\overline{W}$, следующий за адресом определяет направление передачи данных. В этом случае - при операции записи - этот бит равен 0. После того как I2C-master передал адрес и бит $R/\overline{W}$, он освобождает линию SDA. Приемник подтягивает SDA линию в низкий уровень, чтобы подтвердить передачу. Если приемник не подтверждает передачу, I2C-master может генерировать СТОП условие, чтобы прекратить передачу или начать новую передачу сгенерировав повторное условие СТАРТ.

После того как первый байт был подтвержден, I2C-master передает байт данных. Если бит $R/\overline{W}$ был установлен в 1, I2C-master будет действовать как приемник во время фазы передачи. После того, как байт данных был передан, приемник подтверждает байт и I2C-master генерирует условие СТОП, чтобы завершить передачу.

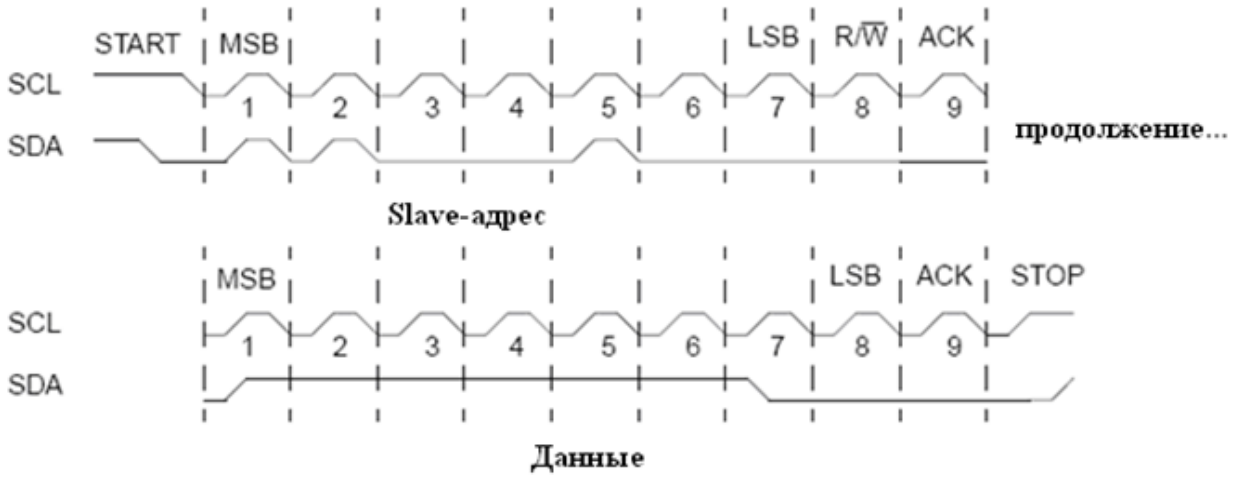


Рисунок 1.47 – Транзакция по шине I2C

I2C-slave может поддерживать 10-битную адресацию. В этом случае I2C-master сначала передает пять зарезервированных битов, после которых следуют два первых бита 10-битного адреса и бит $R/\overline{W}$, сброшенный в 0. Следующий передаваемый байт содержит оставшиеся 8 бит 10-битного адреса. Если нужна операция записи, то потом I2C-master передает данные I2C-slave. Если нужна операция чтения, I2C-master снова генерирует условие СТАРТ и повторяет передачу первой части 10-битного адреса, но при этом бит $R/\overline{W}$ установлен в 1.

Может возникнуть ситуация, когда скорость передачи данных слишком высока для I2C-slave или ему требуется время на обработку данных. В этом случае, чтобы увеличить тактовый интервал, I2C-slave может удерживать SCL в низком уровне после того, как I2C-master установит его туда.

1.3.9.4.5 Описание регистров

1.3.9.4.5.1 Общее описание регистров

Контроллер I2C имеет набор программно доступных регистров, требующий 4 КБайт адресов на системной шине APB.

Карта регистров приведена в таблице 1.267. Она состоит из:

– группы регистров контроллера I2C_Slave. Карта и подробное описание полей приведено в 1.3.9.4.6.2;

– группы регистров контроллера I2C_Master. Карта и подробное описание полей приведено в 1.3.9.4.7.2;

– двух дополнительных управляющих регистров.

При дальнейшем описании отдельных полей регистров используется следующее обозначение режимов доступа пользователя к полям.

– ЧТ – поле доступно только для чтения. Запись данных в поле игнорируется;

– ЗП – поле доступно только для записи. Чтение возвращает значение 0;

– ЧТ/ЗП – поле доступно как для записи, так и для чтения;

Доступ к регистрам осуществляется посредством системного интерфейса управления (APB slave). Поддерживается только 32-х разрядный доступ. Байтовая или двух-байтовая запись запрещена.

Запись в зарезервированные или ЧТ биты или адреса не имеет действия, никакие ошибки не генерируются. Чтение из зарезервированных или ЗП битов или адресов возвращает 0, никакие ошибки не генерируются.

1.3.9.4.5.2 Карта регистров

Таблица 1.267 – Карта регистров I2C

Адрес	Название	Доступ	Значение после сброса	Описание
0x000				По этому смещению расположены регистры контроллера I2C_Slave
0x100				По этому смещению расположены регистры контроллера I2C_Master
0x200	I2C_MODE	ЧТ/ЗП	0x0	Выбор используемого контроллера
0x204	I2C_SW_RST	ЗП	0x0	Программный сброс

1.3.9.4.5.3 Описание полей регистров

1.3.9.4.5.3.1 Регистр I2C_MODE

Смещение адреса 0x200.

В таблице 1.268 представлено описание полей регистра I2C_MODE.

Таблица 1.268 – Описание полей регистра I2C_MODE

Биты	Поле	Доступ	Значение после сброса	Описание
0	SELECT	ЧТ/ЗП	0x0	выбор используемого контроллера: 0 - I2C_Slave 1 - I2C_Master
31:1			0	Зарезервировано

1.3.9.4.5.3.2 Регистр I2C_SW_RST

Смещение адреса 0x204.

В таблице 1.269 представлено описание полей регистра I2C_SW_RST.

Таблица 1.269 – Описание полей регистра I2C_SW_RST

Биты	Поле	Доступ	Значение после сброса	Описание
0	SW_RST	ЗП	0x0	программный сброс обоих контроллеров. Запись 1 запускает процедуру сброса. Запись 0 не имеет действия.
31:1			0	Зарезервировано

1.3.9.4.6 I2C_Slave

1.3.9.4.6.1 Устройство и функционирование блока

Контроллер I2C_Slave совместим со стандартом Philips и поддерживает 7 и 10-битную адресацию. Поддерживаются стандартный режим 100 Кбит/с (Standard-mode) и скоростной режим 400 Кбит/с (Fast-mode). Структурная схема представлена на рисунке 1.48.

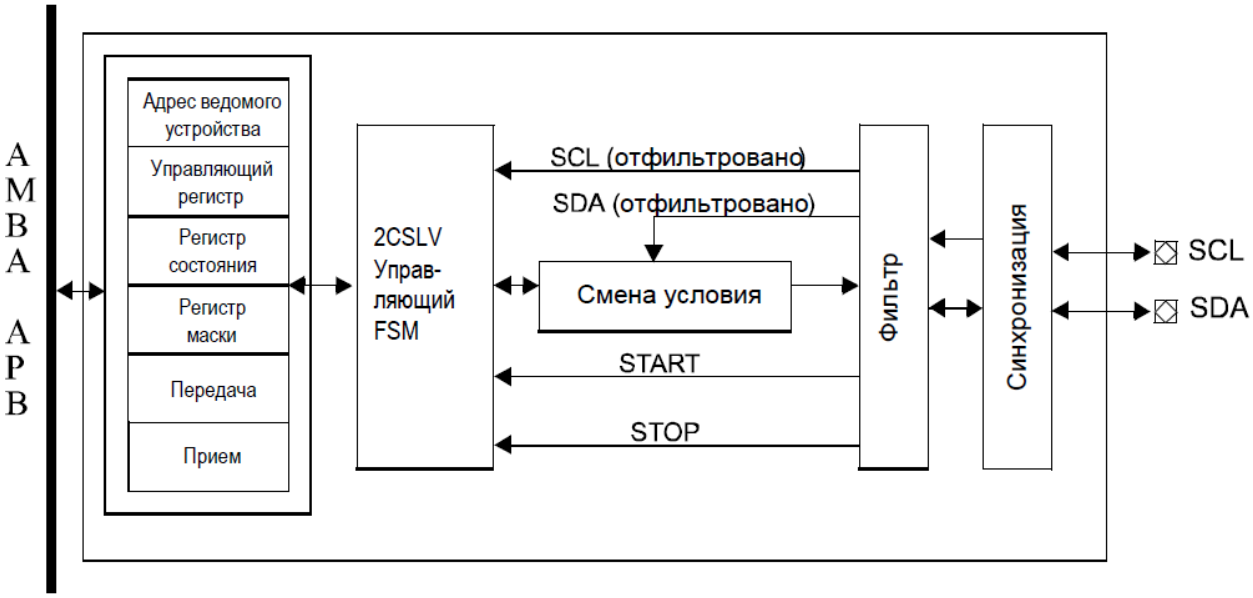


Рисунок 1.48 – Структурная схема I2C_Slave

1.3.9.4.6.1.1 Адресация I2C_Slave

Контроллер I2C_Slave имеет возможность программной настройки своего адреса в поле SLVADDR[SLVADDR]. Может использоваться 7- или 10-битная адресация, в зависимости от значения поля SLVADDR[TBA].

1.3.9.4.6.1.2 Тактирование и скорость работы

I2C_Slave делает выборку входящих тактовых сигналов I^2C SCL и не вводит в систему дополнительные диапазоны изменения тактовой частоты. Входящие сигналы SCL и SDA сначала проходят двухэтапную синхронизацию, после чего проходят фильтр нижних частот, который состоит из четырех регистров.

Условия СТАРТ и СТОП обнаруживаются, если при SCL == 1, SDA сначала находится в одном значении два такта i2c_clk подряд, а потом переключается и сохраняет новый уровень два такта i2c_clk подряд.

Синхронизаторы и фильтры ограничивают минимальную частоту в системе. Чтобы ядро приняло значение SCL как новое тактовое значение, сигнал SCL должен быть стабильным на протяжении хотя бы четырех циклов системного тактового генератора. Отклик ядра на переход будет дан с задержкой, поскольку обе линии проходят двухэтапную синхронизацию, а затем фильтрацию. Поэтому для обнаружения перехода на SCL ядру необходимо более восьми циклов системного тактового генератора.

Наличие синхронизаторов и фильтров ограничивают минимально разрешенную системную частоту i2c_clk:

– 2 МГц - для работы I2C_Slave в стандартном режиме 100 кГц;

– 6 МГц - для работы I2C_Slave в скоростном режиме 400 кГц.

1.3.9.4.6.1.3 Программная модель

I2C_Slave поддерживает четыре основных конфигурации, настраиваемые посредством битов RMOD и TMOD регистра CTRL. Эти конфигурации задают алгоритм работы I2C_Slave в случае приема или передачи байта.

I2C_Slave всегда выдает отрицательное завершение NAK, если регистр приема RX полон и получен еще один байт. Если регистр приема свободен, поле CTRL[RMOD] определяет поведение контроллера:

– 0 - I2C_Slave будет опрашивать шину I2C и ждать следующего действия I2C-master;

– 1 - I2C_Slave переведет и будет удерживать SCL в низком уровне, пока регистр приема не будет прочтен, а бит регистра состояния STAT[REC] не будет очищен. I2C_Slave не будет принимать новый байт, пока предыдущий не будет прочтён из регистра RX.

Когда I2C_Slave получает запрос на чтение, оно проверяет бит CTRL[TV], показывающий наличие валидных данных в регистре TX. Если CTRL[TV] установлен, I2C_Slave подтвердит адрес и приступит к передаче данных, хранящихся в регистре передачи TX. После

завершения передачи байта, I2C_Slave установит в CTRL[TV] значение бита CTRL[TAV]. Эта возможность позволяет отправлять один и тот же байт на все запросы чтения без программного вмешательства. Значение бита CTRL[TMOD] определяет, как действует I2C_Slave после того, как был передан байт и I2C-master подтвердил его. Если I2C-master вернёт NAK на переданный байт, передача будет завершена, и I2C_Slave перейдет в состояние простоя.

– Если TMOD сброшен в 0, когда I2C-master подтверждает прием байта, I2C_Slave продолжит опрашивать шину I2C в ожидании следующего действия I2C-master. Если I2C-master начнет выполнять последовательные операции чтения, I2C_Slave будет отвечать на все последующие запросы байтом в регистре передачи TX.

– Если TMOD установлен в 1, I2C_Slave переведет и будет удерживать SCL в низком уровне после того, как I2C-master подтвердит переданный байт. SCL будет удерживаться до тех пор, пока бит CTRL[TV] не будет программно установлен в 1. Обратите внимание, что если бит CTRL[TAV] установлен в 1, бит CTRL[TV] устанавливается сразу, и I2C_Slave будет работать также, как при CTRL[TMOD] == 0.

При установленных в 1 CTRL[TMOD] или CTRL[RMOD], шина I2C будет заблокирована I2C_Slave до тех пор, пока ПО не подтвердит переданный или полученный байт. Это может снизить производительность работы шины I2C и сказаться на передаче единичных байтов, поскольку I2C-master не сможет генерировать условия СТОП или СТАРТ, когда SCL удерживается в низком уровне.

I2C_Slave генерирует три типа событий в регистре STAT:

1) когда I2C_Slave возвращает NAK на полученный байт или адрес при чтении, устанавливается бит STAT[NAK];

2) в случае успешного приема байта I2C_Slave устанавливает бит STAT[REC];

3) в случае успешной передачи байта I2C_Slave устанавливает бит STAT[TRA].

Установка соответствующих битов в регистре маски позволяет использовать эти три бита в качестве источников прерываний. Конкретный номер и идентификатор прерывания для контроллера приведен в главе 1.3.9.4.8.

1.3.9.4.6.2 Описание регистров

I2C_Slave программируется через регистры, отображаемые в адресном пространстве системной шины. Кроме базового адреса контроллера I2C, эти регистры дополнительно смещены на значение, указанное в таблице 1.267.

Карта регистров I2C_Slave представлена в таблице 1.270.

Таблица 1.270 – Карта регистров I2C_Slave

Адрес	Название	До-ступ	Значение по-сле сброса	Описание
0x00	SLVADDR	ЧТ/ЗП		Регистр адреса устройства
0x04	CTRL			Управляющий регистр
0x08	STAT			Регистр статуса
0x0C	MASK			Регистр маски
0x10	RX			Регистр данных приема
0x14	TX			Регистр данных передачи

1.3.9.4.6.2.1 Описание полей регистров

1.3.9.4.6.2.1.1 Регистр SLVADDR

Смещение адреса 0x00

В таблице 1.271 представлено описание полей регистра.

Таблица 1.271 – Описание полей регистра SLVADDR

Биты	Поле	Доступ	Значение после сброса	Описание
31	TBA	ЧТ/ЗП	1	1 - I2C_Slave интерпретирует значение поля SLVADDR как 10-битный адрес; 0 - в качестве адреса используются младшие 7 бит поля SLVADDR.
30:10			0	Зарезервировано
9:0	SLVADDR	ЧТ/ЗП	0	Адрес этого устройства I2C_Slave на шине I2C.

1.3.9.4.6.2.1.2 Регистр CTRL

Смещение адреса 0x04

В таблице 1.272 представлено описание полей регистра.

Таблица 1.272 – Описание полей регистра CTRL

Биты	Поле	Доступ	Значение после сброса	Описание
31:5			0	Зарезервировано

Биты	Поле	Доступ	Значение после сброса	Описание
4	RMOD	ЧТ/ЗП		Поведение при приёме данных: 0 - I2C_Slave принимает один байт и выдает NAK всем остальным передачам, пока ПО не прочитает регистр приема для подтверждения полученного байта; 1 - I2C_Slave принимает один байт и удерживает SCL на низком уровне, пока ПО не прочитает регистр приема для подтверждения полученного байта.
3	TMOD	ЧТ/ЗП		Поведение при передаче данных: 0 - I2C_Slave передает один и тот же байт, если I2C-master запросит более одного байта при передаче. I2C_Slave выдает NAK на все запросы чтения, пока не будет установлен бит TV; 1 - I2C_Slave передает один байт и удерживает SCL на низком уровне, пока ПО не установит бит TV для подтверждения передачи байта.
2	TV	ЧТ/ЗП		ПО устанавливает этот бит в 1, чтобы показать, что данные в регистре передачи валидны. I2C_Slave автоматически сбрасывает этот бит после передачи байта. Когда этот бит сброшен в 0, в зависимости от занчения TMOD I2C_Slave будет либо выдавать NAK, либо задерживать входящие запросы чтения.
1	TAV	ЧТ/ЗП		1 - I2C_Slave не сбрасывает бит TV автоматически после передачи байта.
0	EN	ЧТ/ЗП		1 - I2C_Slave включен и реагирует на входящие запросы по шине I2C с адресацией, соответствующей SLVADDR[SLVADDR]; 0 - I2C_Slave выключен и оставляет данные SCL и SDA в состоянии Hi-Z.

1.3.9.4.6.2.1.3 Регистр STAT

Смещение адреса 0x08

В таблице 1.273 представлено описание полей регистра.

Таблица 1.273 – Описание полей регистра STAT

Биты	Поле	Доступ	Значение после сброса	Описание
31:3			0	Зарезервировано
2	REC	ЧТ	0	Устанавливается в 1 автоматически, после того как I2C_Slave принял байт. Сбрасывается в 0 автоматически после программного чтения регистра RX.

Биты	Поле	Доступ	Значение после сброса	Описание
1	TRA	ЧТ/ЗП	0	Устанавливается в 1 автоматически, после того как I2C_Slave передал байт. Запись: 1 - сбросит значение поля в 0; 0 - не имеет действия;
0	NAK	ЧТ/ЗП	0	Устанавливается в 1 автоматически, когда I2C_Slave выдал NAK на запрос чтения или записи. Не устанавливается в 1 при выдаче NAK в результате получения транзакции с несоответствующим адресом. Запись: 1 - сбросит значение поля в 0; 0 - не имеет действия;

1.3.9.4.6.2.1.4 Регистр MASK

Смещение адреса 0x0C

В таблице 1.274 представлено описание полей регистра.

Таблица 1.274 – Описание полей регистра MASK

Биты	Поле	Доступ	Значение после сброса	Описание
31:3			0	Зарезервировано
2	RECE	ЧТ/ЗП	0	1 - разрешение генерации прерывания по установке бита STAT[REC]; 0 - запрещено;
1	TRAE	ЧТ/ЗП	0	1 - разрешение генерации прерывания по установке бита STAT[TRA]; 0 - запрещено;
0	NAKE	ЧТ/ЗП	0	1 - разрешение генерации прерывания по установке бита STAT[NAK]; 0 - запрещено;

1.3.9.4.6.2.1.5 Регистр RX

Смещение адреса 0x10

В таблице 1.275 представлено описание полей регистра.

Таблица 1.275 – Описание полей регистра RX

Биты	Поле	Доступ	Значение после сброса	Описание
31:8			0	Зарезервировано
7:0	RECBYTE	ЧТ	0	Последний принятый по шине I2C байт данных. Данные валидны только если был установлен бит STAT[REC].

1.3.9.4.6.2.1.6 Регистр TX

Смещение адреса 0x14

В таблице 1.276 представлено описание полей регистра.

Таблица 1.276 – Описание полей регистра TX

Биты	Поле	Доступ	Значение после сброса	Описание
31:8			0	Зарезервировано
7:0	TRABYTE	ЧТ/ЗП		Байт, который будет послан при следующем полученном по шине I2C запросе чтения.

1.3.9.4.7 I2C_Master

1.3.9.4.7.1 Устройство и функционирование блока

Контроллер I2C_Master совместим со стандартом Philips и поддерживает 7 и 10-битную адресацию. Поддерживаются стандартный режим 100 Кбит/с (Standard-mode) и скоростной режим 400 Кбит/с (Fast-mode). Структурная схема представлена на рисунке 1.49.

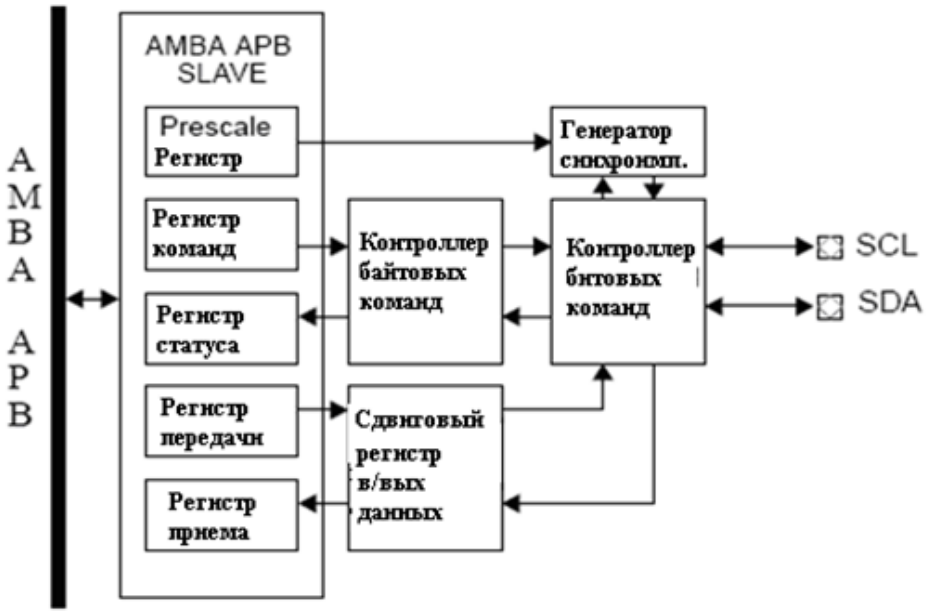


Рисунок 1.49 – Структурная схема I2C_Master

1.3.9.4.7.1.1 Тактирование и скорость работы

I2C_Master использует регистр PRESCALE, чтобы задать частоту линии SCL и частоту 5*SCL, которая используется внутри контроллера. Значение PRESCALE регистра требуется задать в соответствии с формулой:

$$Prescale = \frac{i2c_clk}{5 \cdot SCL_{frequency}} - 1$$

Где SCLfrequency равна 100 кГц для работы в стандартном режиме и 400 кГц для работы в скоростном режиме.

Например, для того, чтобы использовать I2C_Master в стандартном режиме с частотой i2c_clk == 60 МГц, требуется записать следующее значение в поле PRESCALE[Clock_prescale]:

$$Prescale = \frac{60\,Mhz}{5 \cdot 100kHz} - 1 = 119 = 0x77$$

Регистр PRESCALE разрешено изменять только когда контроллер I2C_Master выключен. Минимальное рекомендуемое значение PRESCALE[Clock_prescale] - 3. Это связано с проблемами синхронизации - при более низких значениях будут нарушены временные ограничения I2C.

Таким образом при работе в стандартном режиме минимальное значение частоты i2c_clk ограничено 2 МГц.

При работе в скоростном режиме минимально допустимая частота i2c_clk составляет 20 МГц. При меньших значениях не будет выдержено минимально требуемое стандартом значение setup в 100 нс. Если всё же требуется, чтобы I2C_Master работал от более низкой i2c_clk, следует обеспечить стабильность получаемых от подключенных устройств данных по шине SDA в течении одного такта i2c_clk до фронта SCL.

1.3.9.4.7.1.2 Программная модель

Инициализация I2C_Master состоит из:

– записи нужного значения в регистр PRESCALE;

– разрешения прерываний в поле CTRL[IEN];

– включения контроллера установкой в 1 поля CTRL[EN].

Чтобы передать байт в I2C-slave устройство, I2C_Master должен сгенерировать условие СТАРТ, послать адрес требуемого I2C-slave устройства с битом $R/\overline{W}$, установленным в 0. После того, как I2C-slave подтвердит адрес, I2C_Master передаст данные, подождёт подтверждения и генерирует СТОП условие. Ниже приведена последовательность записи в подключенное I2C-slave устройство:

1) сдвинуть влево на один бит адрес I2C-slave устройства и записать результат в поле TX.[TDATA]. Поле TX.[RW] сбросить в 0;

2) сгенерировать условие СТАРТ и послать содержимое регистра передачи TX установкой в 1 битов STA и WR в командном регистре CMD;

3) подождать прерывание или пока бит TIP в статусном регистре STAT не сбросится в 0;

4) считать бит RXACK в статусном регистре. Если RXACK == 0, I2C-slave устройство подтвердило передачу, можно переходить к шагу 5. Если RXACK == 1, передача не подтверждена, перейти к шагу 1;

5) записать передаваемые данные в регистр передачи TX;

6) послать данные к I2C-slave устройству и сгенерировать СТОП условие установкой битов STO и WR битов в командном регистре;

7) подождать прерывание или пока бит TIP в статусном регистре STAT не сбросится в 0;

8) проверить, что I2C-slave подтвердил прием данных, прочитав бит RXACK в регистре статуса. RXACK должен быть сброшен в 0.

Чтобы считать байт из адресуемой памяти подключенного по I2C шине I2C-slave устройства, большая часть приведенной выше последовательности повторяется. Данные, записываемые в этом случае – это адрес ячейки памяти в I2C-slave устройстве. После записи этого адреса I2C_Master генерирует повторное условие СТАРТ и считывает данные из I2C-slave устройства. Ниже приведена последовательность чтения из I2C-slave устройства:

1) сдвинуть влево на один бит адрес I2C-slave устройства и записать результат в поле TX.[TDATA]. Поле TX.[RW] сбросить в 0;

2) сгенерировать условие СТАРТ и послать содержимое регистра передачи TX установкой в 1 битов STA и WR в командном регистре CMD;

3) подождать прерывание или пока бит TIP в статусном регистре STAT не сбросится в 0;

4) считать бит RXACK в статусном регистре. Если RXACK == 0, I2C-slave устройство подтвердило передачу, можно переходить к шагу 5. Если RXACK == 1, передача не подтверждена, перейти к шагу 1;

5) записать в регистр передачи TX адрес ячейки памяти, которая должна быть считана из I2C-slave;

6) установить бит WR в командном регистре;

7) подождать прерывание или пока бит TIP в статусном регистре STAT не сбросится в 0;

8) считать бит RxACK в статусном регистре. При успешном завершении транзакции RxACK должен быть равен 0;

9) повторно записать сдвинутый адрес I2C-slave устройства в поле TX.[TDATA]. Поле TX.[RW] установить в 1 для чтения данных;

10) установить STA и WR биты в командном регистре, чтобы сгенерировать повторное условие СТАРТ;

11) подождать прерывание или пока бит TIP в статусном регистре STAT не сбросится в 0;

12) считать бит RxACK в статусном регистре. При успешном завершении транзакции RxACK должен быть равен 0;

13) подготовиться получить данные от I2C-slave устройства - установить в 1 биты CMD[RD], CMD[ACK] и CMD[STO]. Установка ACK бит в 1 означает конец передачи;

14) подождать прерывание или пока бит TIP в статусном регистре STAT не сбросится в 0;

15) принятые данные теперь могут быть читаны из регистра приема RX.

Для того, чтобы выполнить последовательность чтений, I2C_Master может повторять итерации 13-15 не устанавливая ACK и STO биты в шаге 13. Чтобы закончить последовательность, биты ACK и STO должны быть установлены. Последовательное чтение может не поддерживаться конкретным I2C-slave устройством - необходимо проверить это в его документации.

Следующая последовательность показывает, как записать один байт в адресуемую память I2C-slave устройства. Сначала адресуется I2C-slave и передается адрес ячейки памяти. После того, как I2C-slave подтвердил адресацию ячейки памяти, данные которые должны быть записаны передаются без генерации нового условия СТАРТ:

1) сдвинуть влево на один бит адрес I2C-slave устройства и записать результат в поле TX.[TDATA]. Поле TX.[RW] сбросить в 0;

2) сгенерировать условие СТАРТ и послать содержимое регистра передачи TX установкой в 1 битов STA и WR в командном регистре CMD;

3) подождать прерывание или пока бит TIP в статусном регистре STAT не сбросится в 0;

4) считать бит RXACK в статусном регистре. Если RXACK == 0, I2C-slave устройство подтвердило передачу, можно переходить к шагу 5. Если RXACK == 1, передача не подтверждена, перейти к шагу 1;

5) записать в регистр передачи TX адрес ячейки памяти, которая должна быть записана в I2C-slave;

6) установить бит WR в командном регистре;

7) подождать прерывание или пока бит TIP в статусном регистре STAT не сбросится в 0;

8) считать бит RxACK в статусном регистре. При успешном завершении транзакции RxACK должен быть равен 0;

9) записать в регистр передачи TX передаваемый байт данных;

10) установить WR и STO в командном регистре, чтобы послать байт данных и сгенерировать условие СТОП;

11) подождать прерывание или пока бит TIP в статусном регистре STAT не сбросится в 0;

12) считать бит RxACK в статусном регистре. При успешном завершении транзакции RxACK должен быть равен 0.

Представленные выше примеры последовательностей в общем случае могут быть применены к I2C-slave устройствам. Однако, работа некоторых устройств может отклоняться от представленного выше протокола. Пожалуйста, с вопросами обращайтесь за справкой к документации конкретного используемого I2C-slave устройства.

Так же стоит отметить, что программный драйвер I2C_Master должен отслеживать бит потери арбитража STAT.[AL].

1.3.9.4.7.2 Описание регистров

1.3.9.4.7.2.1 Карта регистров

I2C_Master программируется через регистры, отображаемые в адресном пространстве системной шины. Кроме базового адреса контроллера I2C, эти регистры дополнительно смещены на значение, указанное в таблице 1.267.

Карта регистров I2C_Master представлена в таблице 1.277.

Таблица 1.277 – Карта регистров I2C_Master

Адрес	Название	До-ступ	Значение по-сле сброса	Описание
0x00	PRESCALE	ЧТ/ЗП	0xFFFF	Регистр делителя тактового сигнала
0x04	CTRL	ЧТ/ЗП	0x0	Управляющий регистр
0x08	TX	ЗП	0x0	Регистр данных передачи
0x08	RX	ЧТ	0x0	Регистр данных приема
0x0C	CMD	ЗП	0x0	Регистр команд
0x0C	STAT	ЧТ	0x0	Регистр статуса
0x10	FILT	ЧТ/ЗП	0x7F	Регистр динамического фильтра

1.3.9.4.7.2.2 Описание полей регистров

1.3.9.4.7.2.2.1 Регистр PRESCALE

Смещение адреса 0x00

В таблице 1.278 представлено описание полей регистра.

Таблица 1.278 – Описание полей регистра PRESCALE

Биты	Поле	Доступ	Значение после сброса	Описание
31:16			0	Зарезервировано
15:0	Clock_prescale	ЧТ/ЗП	0xFFFF	Значение используется, чтобы предварительно поделить частоту SCL линии. Не изменяйте значение этого регистра, пока поле EN управляющего регистра не сброшено в 0. Минимальное рекомендуемое значение этого поля 0x0003. Меньшие значения могут вызвать у I2C_Master нарушение временных требований I2C из-за синхронизационных ограничений.

1.3.9.4.7.2.2.2 Регистр CTRL

Смещение адреса 0x04

В таблице 1.279 представлено описание полей регистра.

Таблица 1.279 – Описание полей регистра CTRL

Биты	Поле	Доступ	Значение после сброса	Описание
31:8			0	Зарезервировано
7	EN	ЧТ/ЗП	0	1 - контроллер I2C_Master включен; 0 - выключен.

Биты	Поле	Доступ	Значение после сброса	Описание
6	IEN	ЧТ/ЗП	0	1 - разрешена генерация прерываний по завершении транзакции; 0 - запрещено.
5:0			0	Зарезервировано

1.3.9.4.7.2.2.3 Регистр TX

Смещение адреса 0x08

В таблице 1.280 представлено описание полей регистра.

Таблица 1.280 – Описание полей регистра TX

Биты	Поле	Доступ	Значение после сброса	Описание
31:8			0	Зарезервировано
7:1	TDATA	ЗП	0	Старшие значащие биты следующего передаваемого по шине I2C байта.
0	RW	ЗП	0	При передаче данных это младший значащий бит данных. При передаче адреса I2C-slave устройства это RW бит: 1 - чтение из I2C-slave; 0 - запись в I2C-slave.

1.3.9.4.7.2.2.4 Регистр RX

Смещение адреса 0x08

В таблице 1.281 представлено описание полей регистра.

Таблица 1.281 – Описание полей регистра RX

Биты	Поле	Доступ	Значение после сброса	Описание
31:8			0	Зарезервировано
7:0	RDATA	ЧТ	0	Последний принятый по шине I2C байт.

1.3.9.4.7.2.2.5 Регистр CMD

Смещение адреса 0x0C

В таблице 1.282 представлено описание полей регистра.

Таблица 1.282 – Описание полей регистра CMD

Биты	Поле	Доступ	Значение после сброса	Описание
31:8			0	Зарезервировано
7	STA	ЗП	0	Запись: 1 - запуск генерации условия СТАРТ на I2C-шине. Этот бит также используется, чтобы сгенерировать повторяющиеся условия СТАРТ; 0 - не имеет действия.
6	STO	ЗП	0	Запись: 1 - запуск генерации условия СТОП на I2C-шине; 0 - не имеет действия.
5	RD	ЗП	0	Запись: 1 - запуск фазы чтения с I2C шины; 0 - не имеет действия.
4	WR	ЗП	0	Запись: 1 - запуск фазы записи по I2C шине; 0 - не имеет действия.
3	ACK	ЗП	0	Запись: 0 - послать завершение ACK по шине I2C; 1 - послать завершение NACK по шине I2C. Поле используется, когда I2C_Master работает в режиме приемника.
2:1			0	Зарезервировано
0	IACK	ЗП	0	Подтверждение прерывания. Запись: 1 - обнуляет флаг прерывания STAT[IF]; 0 - не имеет действия.

1.3.9.4.7.2.2.6 Регистр STAT

Смещение адреса 0x0C

В таблице 1.283 представлено описание полей регистра.

Таблица 1.283 – Описание полей регистра STAT

Биты	Поле	Доступ	Значение после сброса	Описание
31:8			0	Зарезервировано
7	RxACK	ЧТ	0	Полученные подтверждения от ведомого устройства. 0 - I2C-slave устройство подтвердило передачу; 1 - подтверждение не получено.
6	BUSY	ЧТ	0	Шина I2C занята. Этот бит устанавливается аппаратно в: 1 - при обнаружении СТАРТ условия; 0 - при обнаружении СТОП условия.
5	AL	ЧТ	0	Устанавливается аппаратно в 1, когда I2C_Master теряет арбитраж шины I2C. Это происходит, если обнаружено СТОП условие, которое не было запрошено, либо когда I2C_Master устанвливает SDA в высокий уровень, но при этом SDA остается на низком уровне.
4:2			0	Зарезервировано

Биты	Поле	Доступ	Значение после сброса	Описание
1	TIP	ЧТ	0	Идет передача. Устанавливается аппаратно в 1 при передаче данных и в 0, когда передача завершена. Этот бит также устанавливается, когда I2C_Master генерирует СТОП условие.
0	IF	ЧТ	0	Флаг прерывания. Устанавливается аппаратно в 1, когда передача байта завершена или когда потерян арбитраж шины I2C. Если установлен бит CTRL[IEN], будет вызвано прерывание. Конкретный номер и идентификатор прерывания приведен в главе 1.3.9.4.8.

1.3.9.4.7.2.2.7 Регистр FILT

Смещение адреса 0x10

В таблице 1.284 представлено описание полей регистра.

Таблица 1.284 – Описание полей регистра FILT

Биты	Поле	Доступ	Значение после сброса	Описание
31:7			0	Зарезервировано
6:0	FILT	ЧТ/ЗП	0x7F	Это поле устанавливает значение перезагрузки для счетчика динамического фильтра. I2C_Master игнорирует все импульсы на шине I2C, которые будут короче FILT*Ti2c_clk и некоторые импульсы короче 2*FILT*Ti2c_clk–1.

1.3.9.4.8 Интеграция в микроконтроллере

Контроллер I2C использует два внутренних тактовых сигнала микроконтроллера:

1) i2c_clk - от этой частоты работает ядро контроллера и протокол I2C;

2) apb_clk - на этой частоте работает программный интерфейс.

I2C допускает независимые значения этих двух частот, что позволяет не перенастраивать его при смене частоты системной шины apb_clk.

При любом выключенном в RCU тактовом сигнале программный доступ в любой регистр будет возвращать ошибку системной шины и вызывать исключение процессорного ядра.

Контроллер I2C встречается в микроконтроллере в четырёх экземплярах в составе каждого LSIF_PORT:

1) I2C_A;

2) I2C_B;

3) I2C_C;

4) I2C_D.

Каждый контроллер I2C имеет свой базовый адрес на системной шине.

Каждый контроллер I2C генерирует своё прерывание - i2c_a_irq, i2c_b_irq, i2c_c_irq, i2c_d_irq. I2C генерируют уровневые прерывания.

1.3.9.4.8.1 Подключение сигналов SDA и SCL

Сигналы SDA и SCL каждого контроллера мультиплексированы с другими сигналами микроконтроллера. Для работы I2C требуется предварительная программная настройка мультиплексоров.

Буферы микроконтроллера могут подтягивать сигналы SDA и SCL шины I2C к питанию. Это позволяет не использовать подтягивающие резисторы на печатной плате. Требуется предварительная программная настройка.

1.3.9.5 Контроллер интерфейса CAN (CAN)

1.3.9.5.1 Основные характеристики CAN контроллера

CAN контроллер обладает следующими характеристиками:

– поддерживаемые стандарты:

a) 2.0 B;

b) CAN FD;

c) поддержка ISO и non-ISOCAN контроллер кадров;

– размер RX буфера (кадров): 8;

– размер TX буфера (кадров): 8;

– дополнительные режимы работы:

a) loopback (TX замыкается на RX);

b) self test (бит подтверждение игнорируется);

c) bus monitoring (принимает кадры, но не подтверждает их);

– логгирование ошибок;

– аппаратный расчёт CRC;

– генерация маскируемых прерываний;

– интерфейс программирования и передачи данных: APB slave.

1.3.9.5.2 Структурная схема контроллера CAN

На рисунке 1.50 представлена структурная схема контроллера CAN.

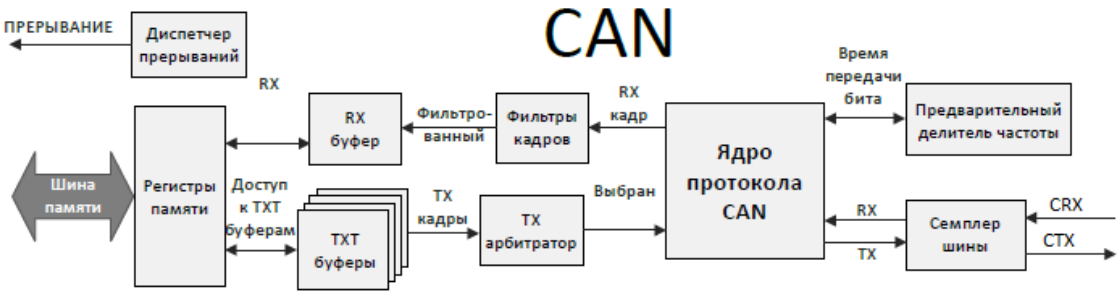


Рисунок 1.50 – Структурная схема контроллера CAN

1.3.9.5.3 Описание внешних выводов контроллера CAN

Список внешних выводов контроллера CAN и их описание приведены в таблице 1.285

Таблица 1.285 – Список внешних выводов контроллера CAN

Наименование вывода	Тип (Вх/Вых)	Функциональное назначение
CRX	Вх	Линия приема данных
CTX	Вых	Линия передачи данных

1.3.9.5.4 Функциональное описание контроллера CAN

1.3.9.5.4.1 Тактирование

CAN контроллер работает по выделенному опорному тактовому сигналу clk_can, который генерируется в RCU (см. раздел 1.3.7.4). При этом обращение к регистрам происходит по системному тактовому сигналу SYSCLK.

1.3.9.5.4.2 Сброс

После включения CAN контроллер может быть сброшен либо аппаратным, либо программным способом. Программный сброс выполняется с помощью команды MODE[RST] = 1. Если CAN контроллер был сброшен аппаратным способом, он становится недоступным в течение двух тактовых интервалов опорного тактового генератора. Например, если частота опорного тактового генератора CAN контроллера составляет 100 МГц, ПО ожидает 20 нс после команды аппаратного сброса. В случае программного сброса ожидание не требуется. Как аппаратный, так и программный сброс действуют аналогичным образом. После любого сброса CAN контроллера переходит в следующее состояние:

– CAN контроллер отключается и перестает получать данные по шине CAN (состояние отключения от шины).

– Все регистры памяти в CAN контроллере содержат значение сброса.

– Компоненты памяти в CAN контроллере (буфер TXT и буфер RX) не сбрасываются.

1.3.9.5.4.3 Организация памяти

Карта распределения памяти CAN контроллера организована в прямом порядке байтов (например, регистр EWL находится на адресе 0x2C, регистр ERP на адресе 0x2D, а регистр FAULT_STATE — на адресе 0x2E). CAN контроллер имеет 32-разрядную память, однако доступ ко всем функциям CAN контроллера можно получить посредством 8/16-разрядного доступа (с соответствующей конфигурацией, см. настройки MODE[RXBAM] — автоматический режим буфера RX).

1.3.9.5.4.4 Временной счетчик

У CAN контроллера реализован временной счетчик для передачи по времени или формирования временных меток для полученных кадров. Его непосредственное значение можно считать из регистров TIMESTAMP_H и TIMESTAMP_L. Ширина временного счетчика равна 64 бит. Количество активных битов во временном счетчике доступно в регистре TS_INFO. Счетчик имеет постоянное значение равное 0.

1.3.9.5.4.5 Режимы работы

После сброса CAN контроллер выключен и не участвует в обмене данными по шине CAN (нет передачи, получения, отслеживания). Прежде чем включать CAN контроллер, его необходимо настроить согласно инструкциям из 1.3.9.5.4.8. По завершении настройки его можно включить установкой бита SETTINGS[ENA] = 1. Когда SETTINGS[ENA] = 1, CAN контроллер начинает интеграцию шины и присоединяется к обмену данными по шине CAN после получения 11 последовательных рецессивных бит. Когда CAN контроллер присоединяется к обмену данными по шине CAN, он находится в состоянии error-active (во время интеграции он был bus-off). В этот момент CAN контроллер начинает обмен данными по шине CAN. Момент, когда CAN контроллер присоединяется к обмену данными по шине CAN, можно определить прерыванием FCS и последующей чтением регистра FAULT_STATE (см. 2.12). Базовые режимы работы CAN контроллера представлены на рисунке 1.51.

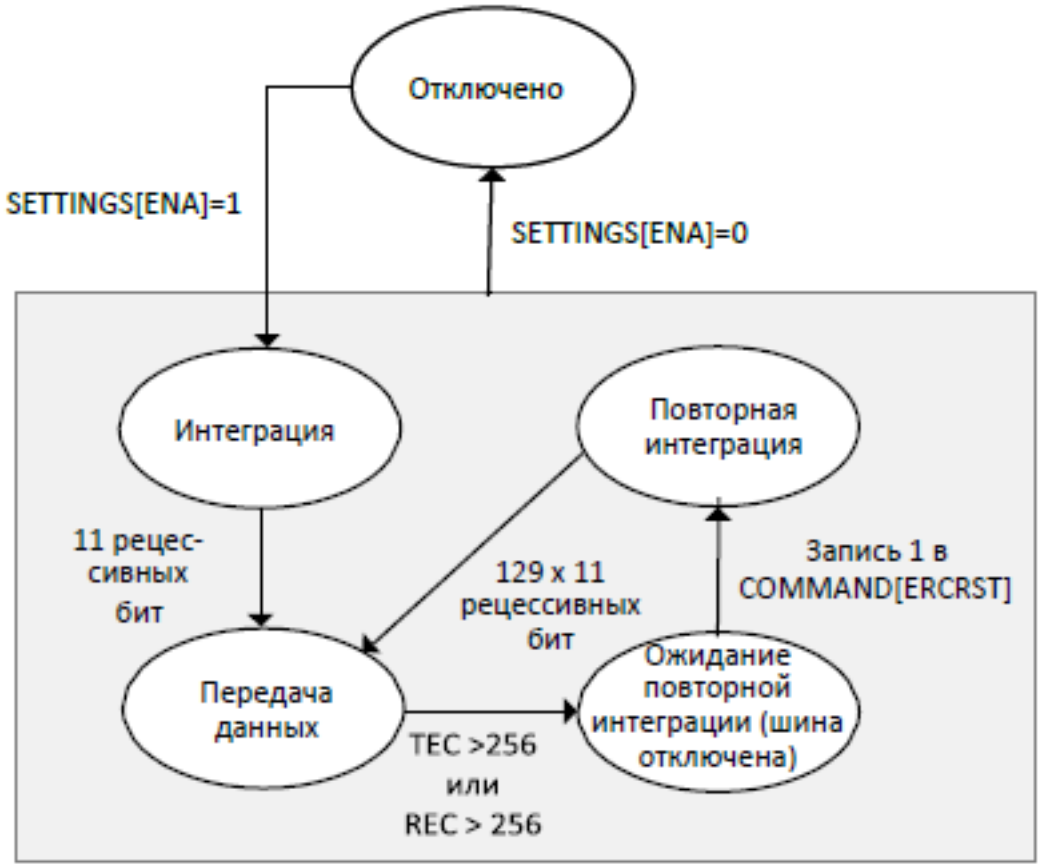


Рисунок 1.51 – Режимы работы

Когда CAN контроллер находится в error-active состоянии, он принимает участие в обмене данными по шине CAN. Если CAN контроллер не действует по ошибкам и отключается от шины, он перестает обмениваться данными по шине CAN и ждет запуска повторной интеграции, пока не получает команду сброса счетчика ошибок (команда COMMAND[ERCRST] = 1). После этой команды CAN контроллер начинает повторную интеграцию. Повторная интеграция длится до тех пор, пока CAN контроллер не обнаружит 129 серий из 11 последовательных рецессивных бит. После 129 таких серий CAN контроллер снова начинает действовать по ошибкам.

CAN контроллер можно отключить в любое время — для этого необходимо записать 0 в регистр SETTINGS[ENA]. В этом случае:

– CAN контроллер немедленно прекращает обмен данными по шине CAN и передает только рецессивные биты.

– Счетчики TEC/REC сбрасываются в 0, CAN контроллер отключается от шины.

– Все TXT-буферы переходят в состояние "Empty" (см. рисунок 1.57), содержимое TXT-буфера остается действительным (память не сбрасывается).

– RX-буфер сбрасывается в исходное состояние (см. пп 1.3.9.5.4.10.5).

Если CAN контроллер отключается посредством записи SETTINGS[ENA] = 0, он не должен передавать никакие кадры, т. к. это может привести к передаче ошибочного кадра другими узлами на шине CAN. Поэтому программному драйверу необходимо следит за тем, чтобы никакие TXT-буферы в CAN контроллер не были в состояниях "Ready", "TX in progress" или "Abort in progress" (см. пп 1.3.9.5.4.9).

Обратите внимание, что COMMAND[ERCRST] является может быть отложенной. Это значит, что если эта команда будет дана, когда CAN контроллер не в состоянии bus-off, CAN контроллер запомнит ее и автоматически начнет повторную интеграцию, когда войдет в состояние bus-off. Это объясняется тем, что команда может быть выдана заранее (во время обмена данными), и CAN контроллер выполнит повторную интеграцию сразу после отключения от шины (без дополнительной программной задержки, вызванной взаимодействием с программным драйвером).

1.3.9.5.4.6 Последовательность инициализации

Последовательность инициализации CAN контроллер включает следующие шаги:

1) Сброс (аппаратный или программный сброс)

2) Конфигурация CAN контроллер:

a) Настройка прерываний

b) Настройка скорости передачи битов согласно инструкциям на этой странице

c) Настройка других функций (фильтры, специальные режимы и т. д.)

3) Включение CAN контроллер посредством написания SETTINGS[ENA] = 1.

4) Опрос регистра FAULT_STATE или ожидание прерывание (INT_STAT[FCSI]). Интеграция завершается, когда FAULT_STATE[ERA]=1 (CAN контроллер находится в состоянии error active).

5) Инициализация завершена, программный драйвер может отправлять и получать кадры.

1.3.9.5.4.7 Последовательность деинициализации

Последовательность деинициализации CAN контроллер включает следующие шаги:

1) Необходимо убедиться, что никакой TXT-буфер не находится в состояниях "Ready", "TX in progress" или "Abort in progress". Для этого необходимо выполнить команду **Set abort** (см. пп 1.3.9.5.4.9) для буферов TXT и не записывать следующие кадры на передачу в буферы TXT.

2) Установить SETTINGS[ENA]=0.

1.3.9.5.4.8 Конфигурация шины CAN

1.3.9.5.4.8.1 Скорость передачи данных

Скорость передачи данных в шине CAN определяется опорным тактовым генератором (см. пп 1.3.9.5.4.1). Основная единица времени в шине CAN — это квант времени. Для получения кванта времени из опорного тактового сигнала (clk_can) его частота должна быть разделена предварительным делителем скорости передачи данных. CAN контроллер обладает отдельным предварительным делителем частоты для номинальной скорости передачи (регистр BTR[BRP]) и скорости передачи данных (регистр BTR_FD[BRP_FD]). Для настройки скорость передачи в CAN контроллер необходимо указать продолжительность Prop_Seg, Phase_Seg1 и Phase_Seg2 (как показано на рисунке 1.52). Они указываются в полях BTR (номинальная скорость передачи) и BTR_FD (скорость передачи данных).

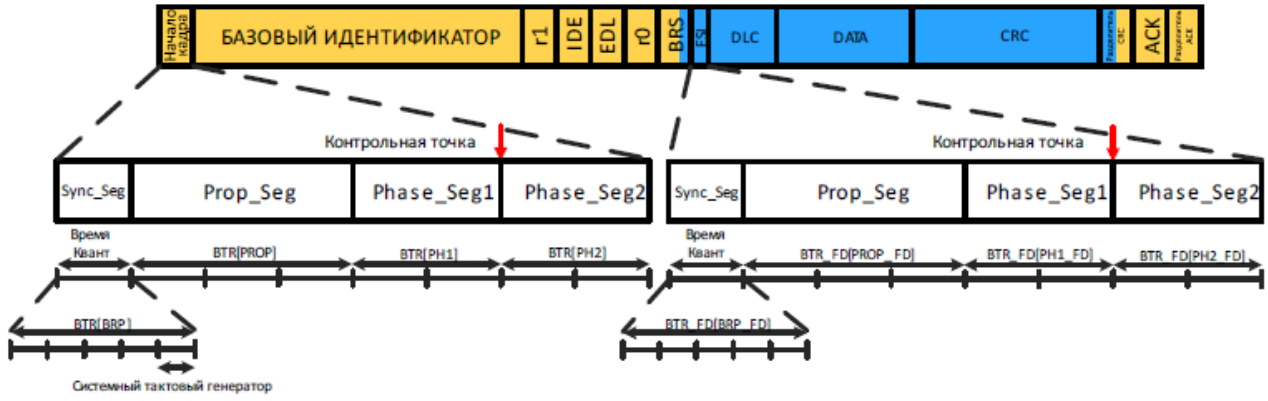


Рисунок 1.52 – Время передачи бита

Пример 500 Кбит / 2 Мбит

В автомобильной отрасли стандартная конфигурация шины CAN составляет 500 Кбит/с номинальной скорости передачи и 2 Мбит/с скорости передачи данных. На примере ниже показана конфигурация с частотой опорного тактового генератора 100 МГц с sample point 80% бита:

```
#define CAN_BASE 0x12000000
```

```
#define BTR_ADDR CAN_BASE+0x24

#define BTR_FD_ADDR CAN_BASE+0x28

uint32 btr;

btr = (4 << 19); // Кванта времени: 4

btr |= 29; // Распр.: 29

btr |= (10 << 7); // Фаза 1: 10

btr |= (10 << 13); // Фаза 2: 10

btr |= (3 << 27); // SJW: 3

can_write_32(BTR_ADDR, btr); // (29+10+10+1)*4=200*10 нс=2 мкс=500 Кбит

uint32 btr_fd;

btr_fd = (1 << 19); // Кванта времени: 1

btr_fd |= 29; // Распр.: 29

btr_fd |= (10 << 7); // Фаза 1: 10

btr_fd |= (10 << 13); // Фаза 2: 10

btr_fd |= (3 << 27); // SJW: 3

can_write_32(BTR_FD_ADDR, btr_fd); // (29+10+10+1)*1=50*10 нс=0,5мкс=2 Мбит
```

1.3.9.5.4.8.2 Задержка передатчика

Задержка передатчика — это задержка распространения сигналов, передаваемых CAN контроллер через выход CAN_TX на вход CAN_RX, как показано на рисунке 1.53. Эта задержка включает распространение сигнала на приемопередатчик физического уровня, задержку самого приемопередатчика и задержку от приемопередатчика на вход CAN_RX ядра CAN контроллер. CAN контроллер измеряет собственную задержку передатчика при передаче кадраCAN контроллер (вне зависимости от того, меняется ли скорость передачи в кадре) смене FDF (EDL) на r0 бит, как показано на рисунке 1.54. Задержку передатчика можно прочитать после измерения в регистре TRV_DELAY. Задержка передатчика измеряется периодами опорного тактового генератора.

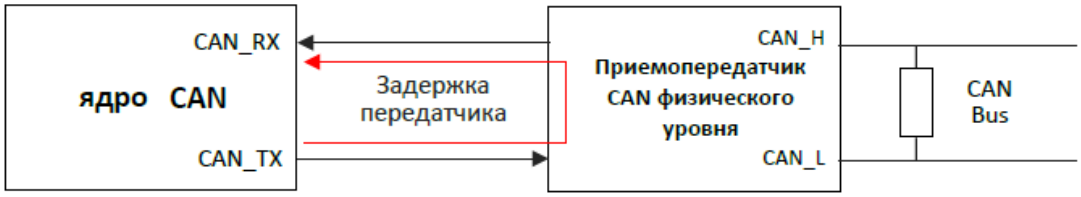


Рисунок 1.53 – Задержка передатчика

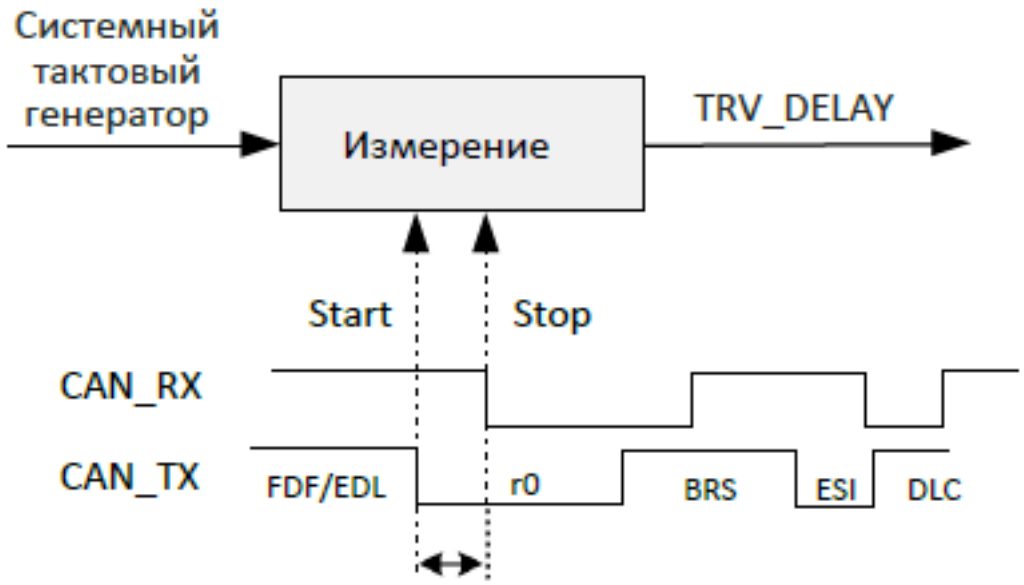


Рисунок 1.54 – Измерение задержки передатчика

Примечание Измеренная задержка передатчика включает задержку на входе CAN контроллер (что составляет 2 тактовых интервала опорного тактового генератора). Это означает, что измеренная задержка передатчика всегда будет на 2 выше фактической задержки от CAN_TX на CAN_RX (например, если распространение сигнала от CAN_TX в CAN_RX занимает 110 нс (11 тактовых интервалов при частоте 100 МГц), измеренная задержка передатчика будет составлять 13).

Примечание Измерение задержки передатчика ограничено 127 тактовыми интервалами. Если задержка между CAN_TX и CAN_RX дольше, измеряется только 127 интервалов. Когда частота опорного тактового генератора составляет 100 МГц, значит максимальная измеряемая задержка передатчика составляет 1,27 мкс, что больше, чем требуется большинство приемопередатчиков CAN.

1.3.9.5.4.8.3 Secondary sample point ((вторичная точка отсчета)

CAN контроллер использует secondary sample point для выявления ошибок битов. Ее расположение настраивается как задержка от начала битового времени (Sync_Seg) в кратных величинах опорного тактового генератора (не квантах времени!). Расположение secondary sample point может быть фиксированным (только SSP_CFG[SSP_OFFSET]), производным от задержки передатчика (SSP_CFG[SSP_OFFSET] + TRV_DELAY) или отключенным (без SSP), как показано на рисунке 1.55. Когда вторичная sample point (точка отсчета) отключена, для передачи CAN контроллер использует регулярную контрольную точку, настроенную регистром BRP_FD (см рисунок 1.56).

Примечание Смещение вторичной контрольной точки (SSP_CFG[SSP_OFFSET]) настраивается в диапазоне от 0 до 255. Внутренний диапазон расположения вторичной контрольной точки тоже составляет от 0 до 255. Если по какой-либо причине расположение вторичной контрольной точки превышает 255 тактовых циклов от начала бита (например, из-за большой измеренной задержки передатчика), она приравнивается к 255.

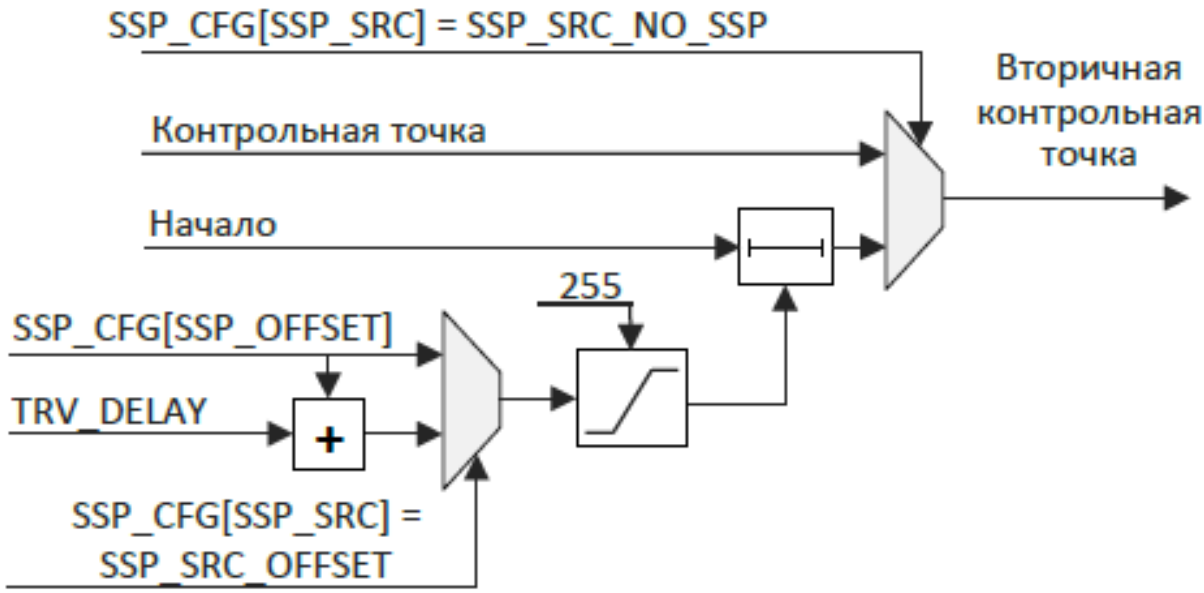


Рисунок 1.55 – Вторичная sample point (точка отсчета)

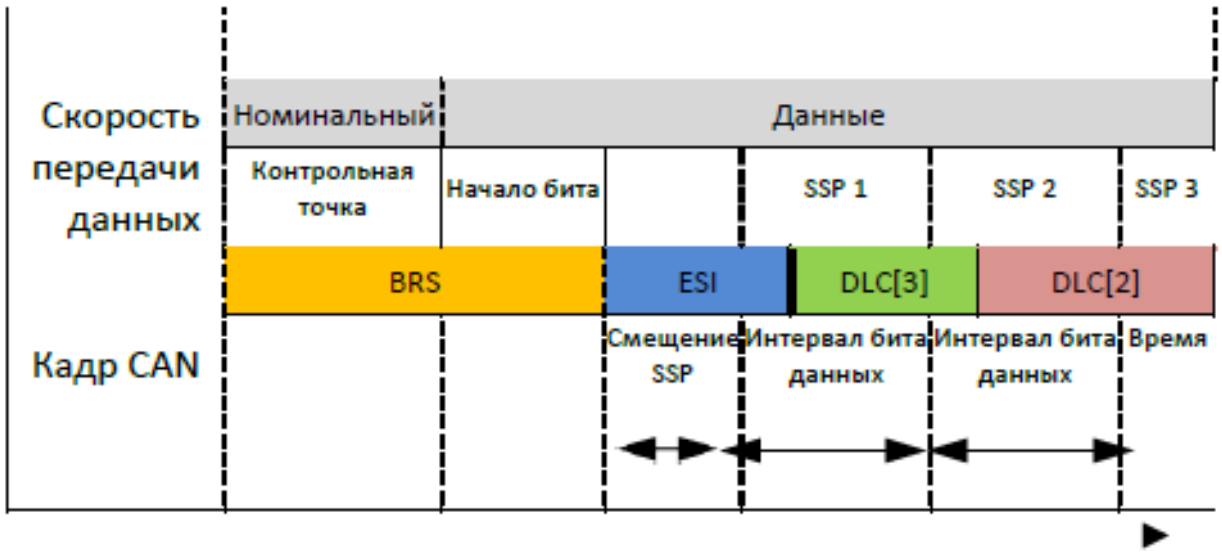


Рисунок 1.56 – Вторичная sample point (точка отсчета) 2

Примечание Поскольку задержка входа CAN контроллер составляет 2 интервала опорного тактового генератора (минимальная кванта времени), расположение вторичной контрольной точки должно настраиваться как минимум на 2 для компенсации собственной задержки входа (если SSP_CFG[SSP_OFFSET] < 3 и SSP_CFG[SSP_SRC] = SSP_SRC_OFFSET, невозможно передавать кадрыCAN контроллер без обнаружения ошибки бита в кадре, переданном CAN контроллер).

Примечание При использовании вторичной контрольной точки CAN контроллер может обработать не более 4 битов между CAN_TX и CAN_RX. Например, если частота опорного тактового генератора = 100 МГц и скорость передачи данных = 5 Мбит/с, то длительность бита данных = 20 интервалов опорного тактового генератора. Таким образом, наиболее позднее расположение вторичной контрольной точки составляет 20 * 4 = 80 интервалов опорного тактового генератора. Это ограничение распространяется на конечное расположение вторичной контрольной точки (с включенным SSP_CFG[SSP_OFFSET]/TRV_DELAY). Пользователям не следует настраивать расположение вторичной контрольной точки позднее, чем 4 длительности бита данных.

1.3.9.5.4.8.4 Поддержка CAN FD

CAN контроллер поддерживает ISO и non-ISO версии протокола CAN контроллер. Если выбран протокол ISO, CAN контроллер соответствует стандарту ISO11898-1 2015. Если выбрана другая версия, CAN контроллер соответствует спецификации CAN FD 1.0. Выбрать одну

из этих версий можно с помощью регистра SETTINGS[NISOFD]. SETTINGS[NISOFD] может быть изменен, только когда CAN контроллер отключен (SETTINGS[ENA] = 0).

1.3.9.5.4.8.5 Поддержка CAN FD

CAN контроллер поддерживает обнаружение исключения протокола. Исключение протокола включается посредством MODE[PEX] = 1. MODE[PEX] может быть изменен, только когда CAN контроллер отключен (SETTINGS[ENA]=0). Алгоритм исключения протокола отличается для различных режимов CAN (см. пп 1.3.9.5.4.1). Если MODE[PEX] = 1 и CAN контроллер обнаруживает исключение протокола, контроллер переходит в состояние интеграции шины и ждет отслеживания 11 последовательных рецессивных битов на сигнале CAN_RX. Счетчики REC/TEC не меняются при исключении протокола или при состоянии локализации сбоя CAN контроллер. Когда происходит исключение протокола, устанавливается флажок STATUS[PEXS]. STATUS[PEXS] может быть очищен с помощью команды COMMAND[CPEXS] = 1. Если MODE[PEX] = 0 и выполнены условия для исключения протокола, CAN контроллер передает кадр ошибки.

1.3.9.5.4.8.6 Тип внедрения

ISO11898-1 2015 определяет три типа внедрения протокола CAN (см таблицу 1.286): Классический CAN, устойчивый CAN FD и включенный CAN FD. CAN контроллер поддерживает все эти три типа внедрения. Соответствие каждому типу реализации можно изменить с помощью битов MODE[FDE] и SETTINGS[PEX]. Оба эти бита могут быть изменены, только когда CAN контроллер отключен (SETTINGS[ENA] = 0).

Таблица 1.286 – Тип внедрения CAN

Тип внедрения	MODE[FDE]	SETTING[PEX]	Алгоритм работы
Classic CAN	0	0	Когда CAN контроллер обнаруживает рецессивный бит FDF (бит после IDE в базовом кадре, бит после RTR/r1 в расширенном кадре), он отвечает кадром ошибки.
CAN FD tolerant	0	1	Когда CAN контроллер обнаруживает рецессивный бит FDF, он обнаруживает исключение протокола и входит в состояние интеграции шины.
CAN FD enabled	1	0	CAN контроллер может получать / передавать кадры CAN FD. Когда CAN контроллер обнаруживает рецессивное значение в расположении резервного бита (один бит после бита FDF), он отвечает кадром ошибки.

Тип внедрения	MODE[FDE]	SETTING[PEX]	Алгоритм работы
CAN FD enabled – с исключением протокола		1	CAN контроллер может получать / передавать кадры CAN FD. Когда CAN контроллер обнаруживает рецессивное значение в расположении резервного бита (один бит после бита FDF), он обнаруживает исключение протокола и входит в состояние интеграции шины. Эта конфигурация устойчива к будущим расширениям протокола CAN FD (например, CAN XL).

Примечание Когда CAN контроллер настраивается как классический CAN / устойчивый CAN FD (MODE[FDE] = 0), а пользователь пытается отправить кадр CAN FD (FRAME_FORMAT_W[FDF_BIT] = 1 в TXT-буфере), тип кадра в TXT-буфере игнорируется и отправляется кадр CAN 2.0.

Примечание Когда CAN контроллер настраивается как классический CAN / устойчивый CAN FD, регистр SETTINGS[NISOFD] оказывается неэффективным.

Примечание Согласно пункту 10.9.10 стандарта ISO11898-1 2015, внедрение включенного CAN FD не может быть настроено на режим, где будет реализовано внедрение устойчивого CAN FD. Поэтому пользователь может выбрать этот вариант только в целях оценки / отладки!

Примечание Согласно исходной спецификации CAN 2.0, получатели принимают биты R0 и R1 любых значений, однако стандарт ISO119898-1 2015 гласит (Таблица A.1), чтобы в таком случае кадры ошибки должны отправляться за счет внедрения классического CAN. Такое расхождение в спецификациях CAN допускается для соответствия ISO11898-1 2015.

1.3.9.5.4.8.7 Минимальное время бита / максимальная скорость передачи

Интервал опорного тактового генератора равен минимальной кванте времени, поэтому влияет на минимально возможную скорость передачи в CAN. CAN контроллер имеет следующие ограничения:

– Phase_Seg2 >= 2 минимальных кванта времени. Это действительно для номинальной скорости передачи и скорости передачи битов.

– Sync_Seg + Prop_Seg + Phase_Seg1 > 2 минимальных кванта времени. Это действительно для номинальной скорости передачи и скорости передачи битов.

Соблюдение этих условий поможет достичь длительности бит в 5 квант времени. Обратите внимание, что возможна номинальная скорость передачи, однако рекомендуется как минимум 8 квант времени за битовый интервал (см. 1.1). Для скорости передачи битов можно использовать 5 квант времени за битовый интервал.

Например, когда номинальная скорость передачи составляет 250 Кбит/с, скорость передачи данных битов составляет 1 Мбит/с, минимально возможная частота опорного тактового генератора — 5 МГц. Обратите внимание, что это абсолютный максимум, который дает маленький запас в расположении контрольной точки. Поэтому для такой скорости передачи рекомендуется использовать частоту тактового генератора как минимум 10 МГц.

1.3.9.5.4.9 Передача кадров CAN

Кадры CAN передаются CAN контроллер с помощью TXT буферов. CAN контроллер содержит 2-8 буферов TXT (кол-во буферов TXT выбирается во время синтеза). Количество имеющихся TXT-буферов можно прочитать в регистре TXTB_INFO. Если имеется N буферов, то это буферы с индексами 1 – "N". Каждый TXT-буфер может быть в одном из состояний, описанных на рисунке 1.57. Состояние каждого TXT-буфера можно прочитать в регистре TX_STATUS. ПО может контролировать состояние TXT-буфера посредством регистра TX_COMMAND и выдавать три типа команд:

Set ready запрашивает перевод TXT-буфера в состояние "Ready".

Set abort запрашивает перевод TXT-буфера в состояние "Aborted" или "Abort in progress".

Set empty запрашивает перевод TXT-буфера в состояние "Empty".

Каждый TXT-буфер хранит один кадр CAN. Кадр CAN FD на 64 байта может войти один TXT-буфер. TXT-буфер предназначен только для записи (кадр CAN не может быть прочитан) и доступен только в состояниях "Empty", "TX OK", "TX failed" или "Aborted". Кадр CAN хранится в TXT-буферах в областях памяти TXT Buffer 1 - TXT Buffer 8, описанных в разделе 3.

После того как программный драйвер сохраняет кадр CAN в TXT-буфере, он выдает команду **Set ready** в TXT-буфер, чтобы запросить передачу хранящегося кадра CAN. TXT-буфер переходит в состояние "Ready", после чего CAN контроллер может передать кадр из этого TXT-буфера. Когда CAN контроллер начинает передачу из этого TXT-буфера, он переходит в состояние "TX in progress". CAN контроллер начинает передачу из TXT-буфера, находящегося в состоянии "Ready", если он обнаруживает доминантный бит во время третьего бита intermission (в этом случае бит начала кадра пропускается), либо как только шина оказывается в режиме idle. Обратите внимание, что в режиме Time triggered будет другой алгоритм работы (см. пп 1.3.9.5.4.9.2).

Когда CAN контроллер находится в error-passive и передавал предыдущий кадр, то он приостанавливает последующую передачу на 8 битовых интервалов. Когда CAN контроллер

успешно передает кадр CAN (доступ к общей шине не потерян, нет кадров ошибки), TXT-буфер переходит в состояние "TX OK". Если появляется кадр ошибки или потерян арбитраж шины, TXT-буфер переходит в состояние "Ready", предпринимается повторная попытка передачи в ближайший intermission или во время idle.

Примечание Когда CAN контроллер работает в режиме отслеживания шины (MODE[BMM] = 1) или режиме ограниченной работы (MODE[ROM] = 1), он всегда оказывается в состоянии "TX failed" при выдаче команды **Set ready**, без попытки передать кадр.

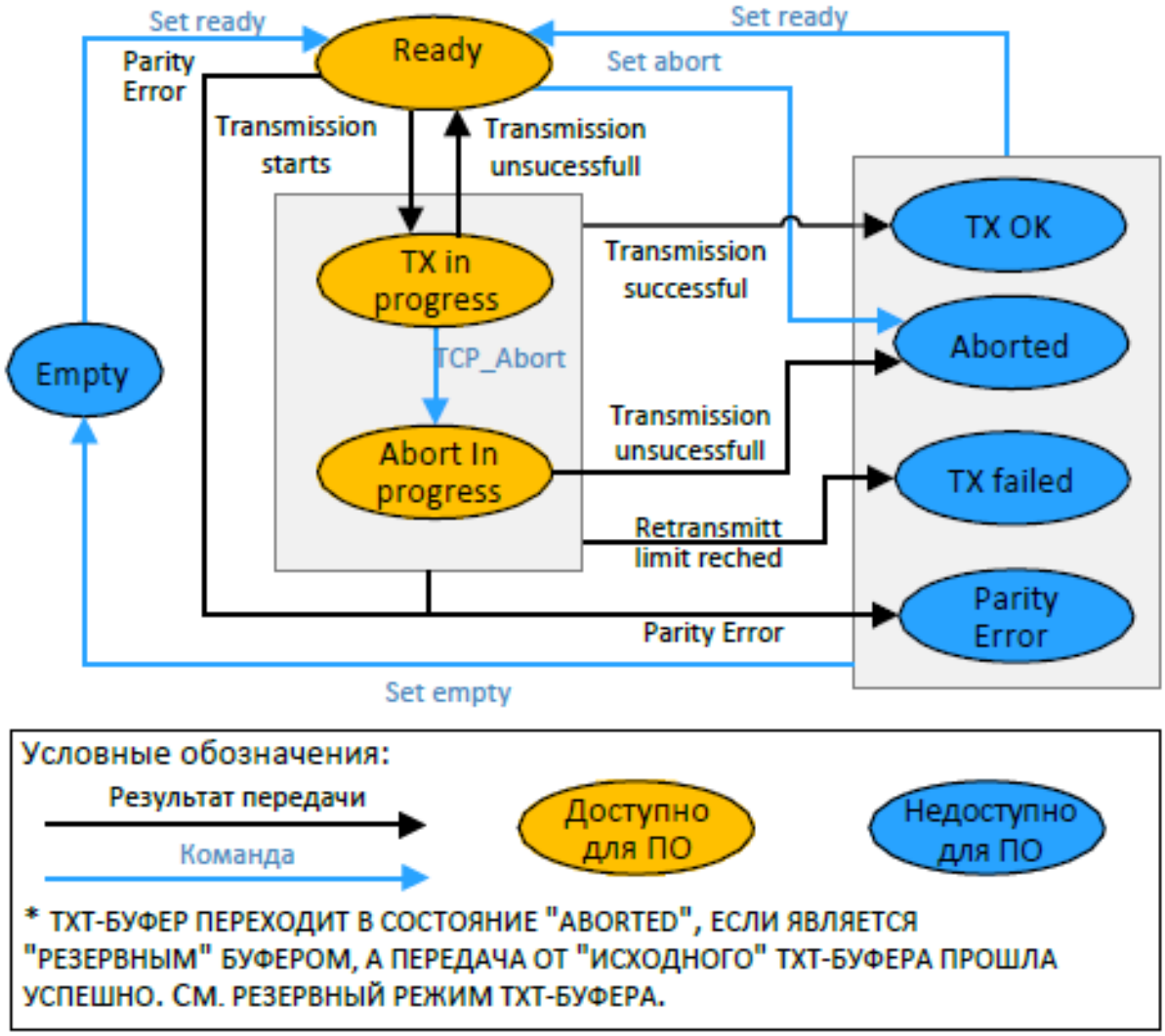


Рисунок 1.57 – Состояния TXT-буферов

1.3.9.5.4.9.1 Выбор TXT-буфера

Если несколько TXT-буферов в состоянии "Ready", CAN контроллер выбирает TXT-буфер с самым высоким приоритетом и передает кадр CAN с помощью этого TXT-буфера.

Приоритет TXT-буферов настраивается в регистре TX_PRIORITY. Если у двух TXT-буферов равный приоритет, выбор отдается в пользу TXT-буфера с меньшим индексом. Поток передачи данных показан на рисунке 1.58.

Примечание Чем выше значение TX_PRIORITY[TX*P], тем выше приоритет TXT-буфера * (например. если TX_PRIORITY[TX1P] = 2, а TX_PRIORITY[TX2P]=5, то TXT-буфер 2 имеет приоритет 5, а TXT-буфер 1 — приоритет 2. Когда оба TXT-буфера находятся в состоянии готовности, CAN контроллер отдает предпочтение TXT-буферу 2, а не TXT-буферу 1).

Примечание Приоритет "резервных" TXT-буферов при MODE[TXBBM] = 1 не определяется соответствующим параметром TX_PRIORITY[TX*P], он настраивается битом, соответствующим "исходному" TXT-буферу. См. пп 1.3.9.5.4.13.3.

1.3.9.5.4.9.2 Режим передачи по времени (Time triggered)

CAN контроллер поддерживает режим Time triggered. Этот режим включается при MODE[TTTM] = 1. В этом режиме CAN контроллер пытается передать кадр из TXT-буфера с высочайшим приоритетом, только когда значение временного счетчика (см. пп 1.3.9.5.4.4) достигает метки времени, хранящейся в TIMESTAMP_L_W и TIMESTAMP_U_W этого TXT-буфера. Предполагается, что временной счетчик является беззнаковым и инкрементирующимся. Когда временной счетчик достигает значения, хранящегося в TIMESTAMP_L_W, TIMESTAMP_U_W, кадр из TXT-буфера допускается к передаче (при условии, что он хранится в TXT-буфере с высочайшим приоритетом и в состоянии "Ready"), как показано на рисунке 1.59. Обратите внимание, что это не значит, что CAN контроллер передаст кадр сразу, он все равно будет ждать, когда шина перейдет в режим ожидания. Если TXT-буфер находится в состоянии "Ready", а временной счетчик пока не достиг момента передачи, CAN контроллер ждет, когда это условие будет удовлетворено. Если в течение этого времени другой узел в шине CAN начнет передавать кадр, CAN контроллер станет получателем этого кадра.

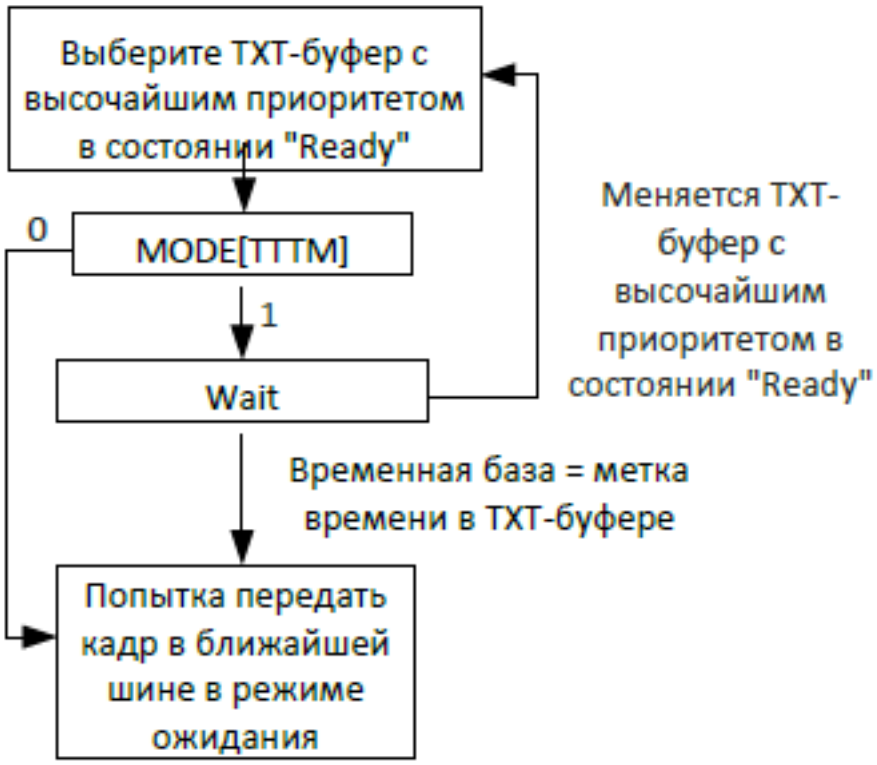


Рисунок 1.58 – Выбор TXT-буфера

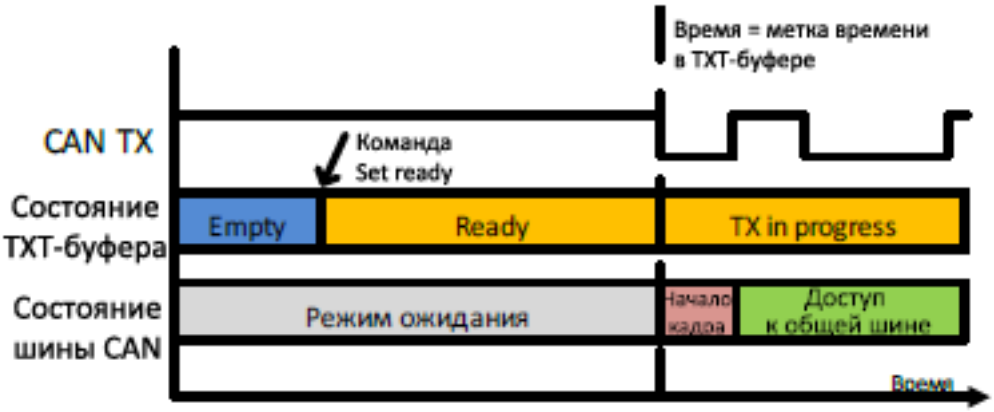


Рисунок 1.59 – Time triggered

Если кадр CAN требуется передать как можно скорее (без передачи по времени), программный драйвер должен сохранить 0x00000000 в TIMESTAMP_L_W, TIMESTAMP_U_W. Обратите внимание, что time triggered всегда учитывается только для TXT-буфера с высочайшим приоритетом в состоянии "Ready". Перед передачей по времени всегда в первую очередь учитывается приоритет TXT-буфера. Приоритет TXT-буфера и time triggered работают следующим образом:

– Если TXT-буфер A имеет более высокий приоритет, чем TXT-буфер B, CAN контроллер выберет кадр из TXT-буфера A, даже если его время передачи больше (передача начнется позже), чем у TXT-буфера B.

– Если приоритет TXT-буфера изменится (и изменится TXT-буфер с высочайшим приоритетом в состоянии "Ready"), то CAN контроллер выберет кадр из нового TXT-буфера с высочайшим приоритетом в состоянии "Ready". Это соответствует действительности, пока кадр из ранее выбранного TXT-буфера ожидает, когда временной счетчик достигнет времени передачи. После начала передачи кадра приоритет TXT-буфера больше не учитывается (нет перестановки кадров).

1.3.9.5.4.9.3 Режим передачи по времени (Time triggered)

Тип кадра CAN, передаваемого CAN контроллер, определяется содержимым FRAME_FORMAT_W в TXT-буфере, а настройки CAN контроллер соответствуют рисунку 1.60.

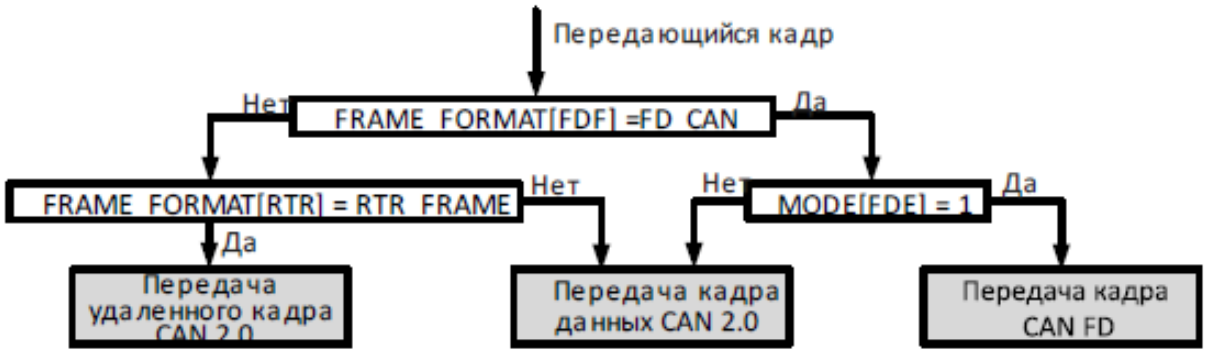


Рисунок 1.60 – Тип передаваемого кадра

Примечание Когда FRAME_FORMAT_W[FDF] = FD_CAN и MODE[FDE] = 0, CAN контроллер передает кадр CAN 2.0. Если в таком случае TXT-буфер содержит кадр CAN FD, полезная информация в котором составляет более 8 байтов, 8 байтов после 8-го бита не будут передаваться.

Примечание Когда FRAME_FORMAT_W[RTR] = RTR_FRAME и FRAME_FORMAT_W[FDF] = FD_CAN, флаг RTR игнорируется, CAN контроллер передает кадр данных CAN FD (в протоколе CAN FD больше нет удаленных кадров).

1.3.9.5.4.9.4 Ограничение повторной отправки

CAN контроллер может ограничить количество повторных попыток отправки одного TXT-буфера. Ограничение повторной передачи включается при значении SETTINGS[RTRLE]

= 1. Количество повторных передач настраивается в параметре SETTINGS[RTRTH]. Первая попытка передачи кадра CAN не считается повторной передачей. Возможные варианты конфигурации представлены в таблице 1.287.

Таблица 1.287 – Конфигурация ограничения повторной передачи

SETTINGS [RTRTH]	SETTINGS [RTRLE]	Алгоритм
-	0	Передача кадров осуществляется без каких-либо ограничений (до успешной передачи или до отключения блока от шины).
0	1	Повторная передача не осуществляется (так называемый покадровый режим).
1 - 15	1	Передача кадра осуществляется SETTINGS[RTRTH] + 1 раз (изначальная передача + SETTINGS[RTRTH] повторных попыток).

Если количество повторный передач превысило ограничение для одного TXT-буфера (ошибка кадра или потерян доступ к общей шине), то этот TXT-буфер переходит в состояние "TX failed". Если TXT-буфер, использовавшийся для передачи, был изменен между двумя передачами (например, был выбран из-за более высокого приоритета), внутренний счетчик повторных передач сбрасывается, и у нового кадра (из нового TXT-буфера) снова появляется SETTINGS[RTRTH]+1 попыток передачи. Если CAN контроллер возвращается к передачи от исходного TXT-буфера, он не помнит предыдущее количество попыток передачи и снова пытается передать кадра CAN SETTINGS[RTRTH]+1 раз. Если TXT-буфер, использующийся для передачи на данный момент, переходит в состояние "Aborted", внутренний счетчик повторных передач также сбрасывается. Если такой TXT-буфер снова переходит в состояние "Ready", CAN контроллер пытается передать его SETTINGS[RTRTH]+1 раз. Текущее количество попыток передачи одного кадра содержится во внутреннем счетчике, который можно прочитать в регистре RETR_CTR.

1.3.9.5.4.9.5 Прерывание

Если программный драйвер ранее запросил передачу кадра CAN посредством команды **Set ready**, он может запросить прервать передачу с помощью команды **Set abort**. Если TXT-буфер получит команду **Set abort**, находясь в состоянии "Ready" (передача еще не началась), он сразу перейдет в состоянии "Aborted". Если TXT-буфер находится в состоянии "TX in progress" (передача началась), он перейдет в состоянии "Abort in progress". В этом случае буфер перейдет в состоянии "Aborted" при следующем кадре ошибки или потере доступа к общей шине. Обратите внимание, что когда TXT-буфер находится в состоянии "Abort in progress", в случае успешной текущей передачи он может перейти в состояние TX OK, а если будет достигнуто ограничение повторной передачи — в состояние "TX failed state".

1.3.9.5.4.9.6 Алгоритм TXT-буфера при отключении от шины

Когда CAN контроллер отключается от шины из-за TEC > 255, TXT-буферы могут изменить свое состояние следующим образом:

1) Все TXT-буферы, находящиеся в режиме "Ready", "TX in Progress" или "Abort in Progress", немедленно переходят в состояние "TX failed". Этот параметр включается посредством настройки SETTINGS[TBFBO] = 1 и является для TXT-буферов конфигурацией по умолчанию.

2) Когда CAN контроллер отключается от шины, TXT-буфер, использующийся для передачи, реагирует так же, как если бы был передан любой другой кадр ошибки. Этот параметр включается посредством настройки SETTINGS[TBFBO] = 0. Если этот буфер не получит команду "Set abort" или не будет достигнуто ограничение повторной передачи, буфер перейдет в состояние "Ready". Когда CAN контроллер завершит реинтеграцию (см. пп 1.3.9.5.4.5), передача из этого TXT-буфера начнется так же, как от обычного TXT-буфера, выбранного по приоритету. Этот параметр допускает отключение от шины и реинтеграцию без необходимости программного взаимодействия с TXT-буферами.

1.3.9.5.4.9.7 Пример кода

```
#define CAN_BASE 0x12000000

#define TX_COMMAND_ADDR (CAN_BASE + 0x74)

#define TXT_BUFFER_1_BASE (CAN_BASE + 0x100)

/* Вставить кадр CAN в TXT-буфер 1 */

uint32_t frame_format_word = 0;

frame_format_word |= 4; // DLC = 4

frame_format_word |= (1 << 7); // Кадр CAN FD

frame_format_word |= (1 << 9); // Изменить скорость передачи данных

can_write_32(TXT_BUFFER_1_BASE, frame_format_word); // Сохранить формат кадра

uint32_t id_word = (55 << 18); // Идентификатор: 55 can_write_32(TXT_BUFFER_1_BASE + 0x4
, id_word); // Сохранить идентификатор can_write_32(TXT_BUFFER_1_BASE + 0x8, 1000); // O
тправить когда временной

can_write_32(TXT_BUFFER_1_BASE + 0xC, 0); // счетчик равен 1000 can_write_32(TXT_BUFFER_
1_BASE + 0x10, 0xAABBCCDD); // Данные: 0xAA 0xBB 0xCC 0xDD

/* Выдать команду Set ready */
```

```
uint32_t command = 0;

command |= 0x2; // Команда Set Ready

command |= (1 << 8); // Выбрать TXT-буфер 1

can_write_32(TX_COMMAND_ADDR, command); // Выдать команду
```

Обратите внимание Когда CAN контроллер включается посредством настройки SETTINGS[ENA] = 1, во время интеграции с шиной CAN он остается отключенным от шины. Если в это время выдать команду **Set ready** TXT-буферу, последний сразу перейдет в состояние "Aborted", если SETTINGS[TBFBO] = 1. Прежде чем выдавать команду **Set Ready** какому-либо TXT-буферу, ПО должно дождаться, когда узел перейдет в error-active (либо путем опроса FAULT_STATE, либо посредством прерывания FCS).

Примечание TXT-буферы не инициализируются и не сбрасываются. Поэтому, прежде чем выдавать команду **Set ready**, ПО должно вставить действительный кадр CAN в соответствующий TXT-буфер для передачи.

Примечание CAN контроллер передает кадры Overload только по факту перегрузки. CAN контроллер не имеет внутренних условий, при которых кадр Overload будет передать без обнаружения перегрузки.

1.3.9.5.4.10 Получение кадра CAN

CAN контроллер содержит один RX-буфер, где хранятся полученные кадры CAN. Размер RX-буфера измеряется в 32-битных слов, их количество может быть прочитано из регистра RX_MEM_INFO. RX-буфер организован как FIFO. После успешного получения на шине CAN (кадров ошибки не было) кадр CAN сохраняется в RX-буфере. ПО считывает кадр CAN из RX-буфера путем последовательного чтения регистра RX_DATA. Одно чтение регистра RX_DATA считывает одно слово из буфера RX. RX-буфер работает в одном из двух режимов:

– Автоматический режим: при чтении регистра RX_DATA указатель чтения RX-буфера FIFO передвигается автоматически. Этот режим следует использовать только при чтении RX_DATA посредством 32-битного доступа. Команда COMMAND[RXRPMV] = 1 не имеет результата в этом режиме.

– Ручной режим: при чтении регистра RX_DATA указатель чтения RX-буфера FIFO НЕ передвигается. Для передвижения указателя чтения ПО должно выдать команду COMMAND[RXRPMV] = 1. Этот режим можно использовать, когда RX-буфер считывается

посредством 8/16/32-битного доступа, поскольку он поддерживает чтение одного слова из памяти RX-буфера посредством 4x8-битного или 2x16-битного доступа.

Режим RX-буфера настраивается с помощью бита MODE[RXBAM]. Формат кадра CAN в RX-буфере описан в разделе 3 и показан на рисунке 1.61. Кадр CAN в RX-буфере составляет от 4 до 20 слов памяти. Его размер определяется следующим образом:

Размер RX-кадра в словах = 4 + ячейка (длина поля данных / 4)

1.3.9.5.4.10.1 Счет кадров

RX-буфер содержит счетчик кадров CAN. Этот счетчик считывает данные с регистра RX_STATUS[RXFRCE]. Счетчик увеличивается, когда кадр сохраняется в RX-буфере, и уменьшается, когда в RX-буфере считывается последнее слово кадра CAN.

1.3.9.5.4.10.2 Память RX-буфера

RX-буфер имеет следующую информацию о состоянии:

– Количество свободных слов памяти, которые могут быть считаны с RX_MEM_INFO[RX_MEM_FREE].

– Расположение указателя записи, которое может быть считано с RX_POINTERS[RX_WPP].

– Расположение указателя чтения, которое может быть считано с RX_POINTERS[RX_RPP].

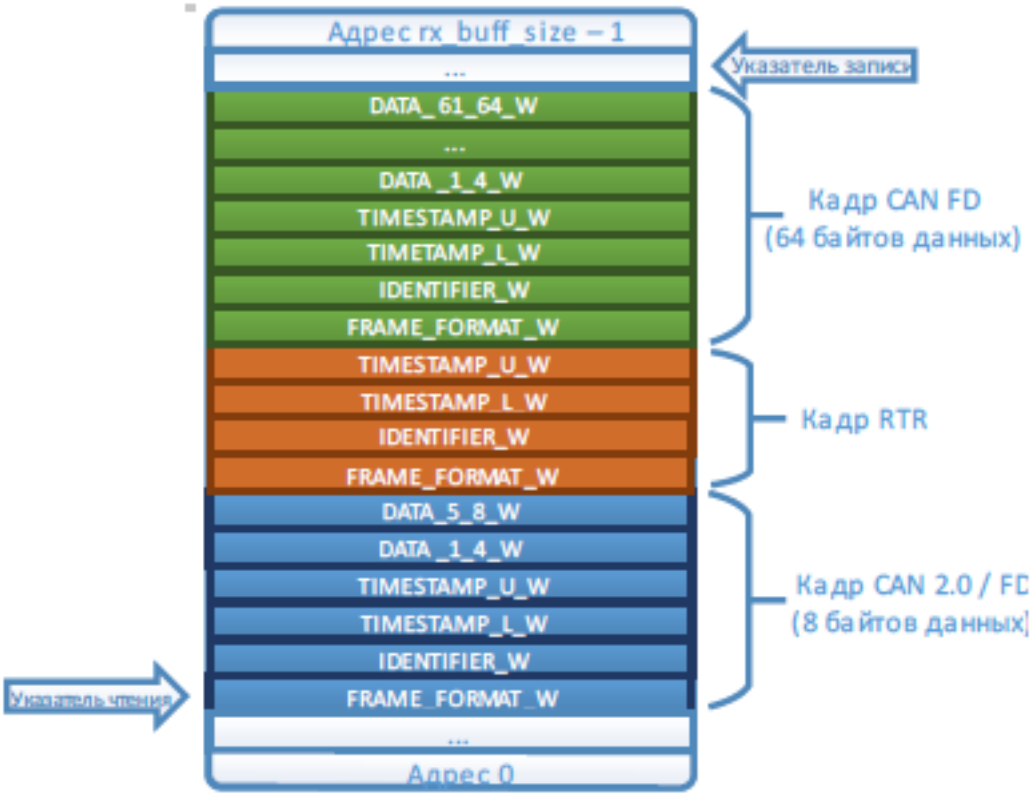


Рисунок 1.61 – RX-буфер

1.3.9.5.4.10.3 Состояние RX-буфера

RX-буфер, который не содержит кадров CAN, является пустым. В таком состоянии RX_STATUS[RXE]=1. RX-буфер, все слова памяти которого заняты кадрами CAN, является полным. В таком состоянии RX_STATUS[RXF]=1. Обратите внимание, что если, к примеру, у RX-буфера 2 свободных слова памяти, он не является полным, однако даже самый маленький кадр CAN не войдет в буфер.

1.3.9.5.4.10.4 2.10.4 Переполнение

Если во время получения кадра CAN в RX-буфере оказалось недостаточно свободного места для хранения полученного кадра CAN, это приводит к переполнению, и кадр отбрасывается (RX-буфер FIFO переполнен). В этом случае ставится флажок Overrun. Флажок Overrun остается вплоть до сброса и может быть считан с STATUS[DOR]. Флажок Overrun сбрасывается командой COMMAND[CDO]=1.

1.3.9.5.4.10.5 Очистка

Для сброса RX-буфера выполните команду COMMAND[RRB]=1. После сброса RX-буфера его содержимое сохраняется (память не стирается), однако указатели записи и чтения устанавливаются на 0, как и счетчик кадров. RX-буфер будет выглядеть так, будто не был получен ни один кадр. Если эта команда будет выдана во время приема кадра CAN, полученный кадр также будет отброшен.

1.3.9.5.4.10.6 Защита от непоследовательности

Чтение кадра CAN с RX-буфера включает несколько чтений регистра RX_DATA. Каждое чтение сдвигает указатель чтения в RX-буфере (операция чтения с побочным эффектом). В случае ошибки (например, ошибки шины, ошибки ECC) во время чтения с регистра RX_DATA, считанные данные могут быть потеряны (RX-буфер сдвигает указатель чтения, и программный драйвер не может прочитать слово повторно). Из-за этого программный драйвер и RX-буфер могут стать непоследовательными. Программный драйвер может восстановиться от этого состояния с помощью RX_STATUS[RXMOF]. Если следующее чтение с регистра RX_DATA должно вернуть слово, отличающееся от FRAME_FORMAT_W (указатель чтения RX-буфера указывает на середину кадра), то RX_STATUS[RXMOF] = 1. Если во время чтения кадра программный драйвер переходит в непоследовательное состояние, он несколько раз считывает с RX_DATA до тех пор, как не будет достигнуто значение RX_STATUS[RX_MOF] = 0. Когда обнаруживается это состояние, RX_DATA указывает на слово FRAME_FORMAT_W в новом кадре, либо RX-буфер признается пустым (если во время чтения единственного кадра в RX-буфере произошла ошибка).

1.3.9.5.4.10.7 Формирование меток времени

Когда CAN контроллер получает кадр CAN, он сохраняет его метку времени (опорное значение временного счетчика) в словах TIMES- TAMP_L_W, TIMESTAMP_U_W RX-буфера. Временная метка полученного кадра берется в sample point стартового бита либо в 6-м бите конца кадра (момент получения кадра CAN считается действительным согласно ISO11898-1 2015). Расположение, где берется временная метка, настраивается посредством RX_SETTINGS[RTSOP].

1.3.9.5.4.10.8 Фильтрация кадров

Полученные кадры CAN фильтруются аппаратными фильтрами. Есть три битовых фильтра (A,B,C) и один диапазонный фильтр. Полученный кадр CAN сохраняется в RX-

буфере, если проходит хотя бы один фильтр (логическое ИЛИ между результатами фильтра). Фильтры кадра применяются на идентификаторе полученного кадра, только если включен режим фильтрации "Acceptance" (Прием) (MODE[AFM] = 1). MODE[AFM] может быть изменен, только когда CAN контроллер отключен (SETTINGS[ENA] = 0). Когда режим фильтрации "Acceptance" отключен, фильтрация не выполняется, и каждый полученный кадр CAN сохраняется в RX-буфере.

Каждый фильтр может быть выборочно настроен на принятие только определенных типов кадров CAN (CAN 2.0 / CAN FD) и типов идентификаторов (например, кадр только с базовым идентификатором, кадр с базовым + расширенным идентификатором). Такая конфигурация доступа в регистре FILTER_CONTROL. Чтобы отключить любой фильтр, все биты в регистре FILTER_CONTROL, принадлежащие этому фильтру, следует настроить на значение 0. Работа фильтров кадров описана на рисунке 1.62.

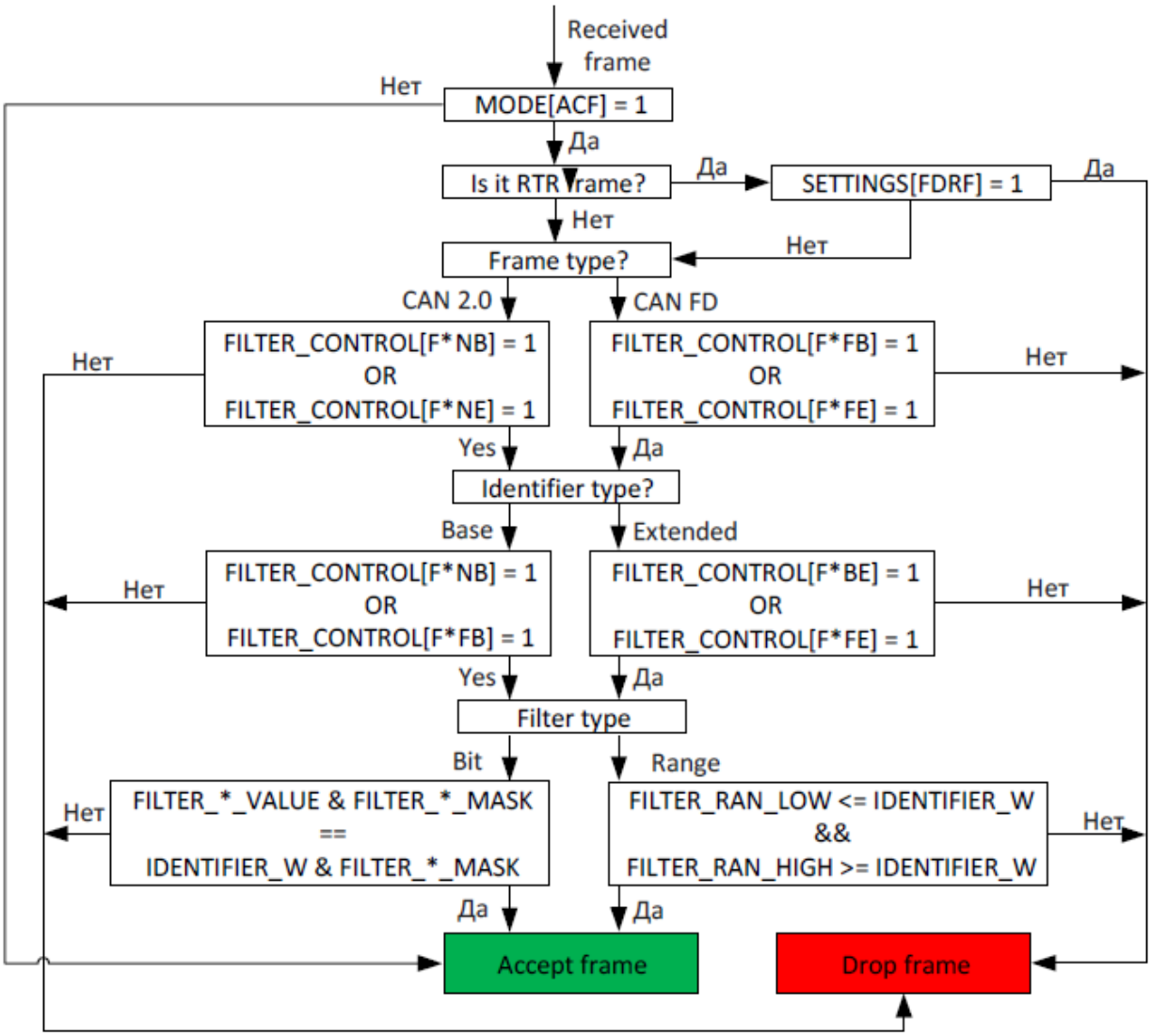


Рисунок 1.62 – Работа фильтров кадров

– означает A/B/C/R на основе типа фильтра

Фильтр битов

Фильтр битов проверяет, равен ли идентификатор полученного кадра CAN предустановленному идентификатору в регистре FILTER_X_VALUE (X=A,B,C на основе экземпляра фильтра). Сравниваются только биты, предоставленные маской фильтра в регистре FILTER_X_MASK.

Примечание Если фильтр битов используется для фильтрации кадров с базовыми идентификаторами, настройте FILTER_X_MASK[17:0] = 0b00000000000000000.

Фильтр диапазона

Фильтр диапазона определяет, входит ли идентификатор полученного кадра в определенный десятичный диапазон. Нижний предел этого десятичного диапазона задан параметром FILTER_RAN_LOW, верхний предел — параметром FILTER_RAN_HIGH.

Примечание Когда фильтр диапазона используется для фильтрации кадров с базовыми идентификаторами, настройте FILTER_RAN_LOW[17:0] = 0b000000000000000000 и FILTER_RAN_HIGH[17:0] = 0b111111111111111111.

1.3.9.5.4.10.9 Пример кода 1: Прием кадра в автоматическом режиме (32-битный доступ)

```
#define CAN_BASE 0x12000000

#define RX_DATA_ADDR (CAN_BASE + 0x6C)

#define RX_STATUS_ADDR (CAN_BASE + 0x68)

/* Выполнение опроса RX-буфера, пока в нем не появится кадр */ uint32_t rx_status;

do \{

rx_status = can_read_32(RX_STATUS_ADDR);

} while ((rx_status & 0x1) == 0)

/* Чтение кадра из RX-буфера */ uint8_t data[64];

uint32_t tmp;

uint32_t ffw = can_read_32(RX_DATA_ADDR);

uint32_t id = can_read_32(RX_DATA_ADDR);

uint32_t ts_l = can_read_32(RX_DATA_ADDR);

uint32_t ts_h = can_read_32(RX_DATA_ADDR);

uint32_t rwcnt = (ffw >> 11) & 0x1F;

for(int i = 0; i < rwcnt; i++)\{

tmp = can_read_32(RX_DATA_ADDR);

data[i*4] = tmp & 0xFF;

data[i*4+1] = (tmp >> 8) & 0xFF;

data[i*4+2] = (tmp >> 16) & 0xFF;
```

```
data[i*4+3] = (tmp >> 24) & 0xFF;

}
```

1.3.9.5.4.10.10 Пример кода 2: Прием кадра в ручном режиме (8-битный доступ)

```
#define CAN_BASE 0x12000000

#define RX_DATA_ADDR (CAN_BASE + 0x6C)

#define RX_STATUS_ADDR (CAN_BASE + 0x68)

#define COMMAND_ADDR (CAN_BASE + 0xC)

#define MOVE_RX_BUF_READ_PTR() can_write_8(COMMAND_ADDR, 1 << 2)

/* Выполнение опроса RX-буфера, пока в нем не появится кадр */ uint8_t rx_status;

do \{

rx_status = can_read_8(RX_STATUS_ADDR);

} while ((rx_status & 0x1) == 0)

/* Чтение слова формата кадра и перемещение указателя RX */

uint8_t data[64];

uint16_t ffw = (uint16_t)can_read_8(RX_DATA_ADDR);

ffw |= (((uint16_t)can_read_8(RX_DATA_ADDR + 0x1)) << 8);

MOVE_RX_BUF_READ_PTR();

/* Чтение идентификатора CAN и перемещение указателя RX к первому слову данных */

uint32_t id = (uint32_t)can_read_8(RX_DATA_ADDR);

id |= ((uint32_t)can_read_8(RX_DATA_ADDR + 0x1)) << 8;

id |= ((uint32_t)can_read_8(RX_DATA_ADDR + 0x2)) << 16;

id |= ((uint32_t)can_read_8(RX_DATA_ADDR + 0x3)) << 24;

for (int i = 0; i < 3; i++)

MOVE_RX_BUF_READ_PTR();

/* Чтение байтов данных */

uint16_t rwcnt = (ffw >> 11) & 0x1F;

for(int i = 0; i < rwcnt; i++)\{
```

```
data[i*4] = can_read_8(RX_DATA_ADDR);

data[i*4+1] = can_read_8(RX_DATA_ADDR + 0x1);

data[i*4+2] = can_read_8(RX_DATA_ADDR + 0x2);

data[i*4+3] = can_read_8(RX_DATA_ADDR + 0x3);

MOVE_RX_BUF_READ_PTR();

}
```

1.3.9.5.4.10.11 Пример кода 3: Конфигурация фильтра битов

```
#define CAN_BASE 0x12000000

#define FILTER_CONTROL_ADDR (CAN_BASE + 0x5C)

#define FILTER_A_VAL_ADDR (CAN_BASE + 0x40)

#define FILTER_A_MASK_ADDR (CAN_BASE + 0x3C)

uint32_t filter_mask = 0xF << 18; // Сравнение 4 младших битов базового идентификатора

uint32_t filter_val = 0x2 << 18; // Должен быть равен 0x2 (0010)

/* Конфигурация фильтра A */

can_write_32(FILTER_A_VAL_ADDR, filter_val);

can_write_32(FILTER_A_MASK_ADDR, filter_mask);

/* Включение приема кадров CAN 2.0 и CAN FD только с базовыми идентификаторами */

uint32_t filter_control = 0x5; // FANB, FAFB

can_write_32(FILTER_CONTROL_ADDR, filter_control);
```

1.3.9.5.4.11 Состояния ошибок (FAULT CONFINEMENT)

Состояние ошибок CAN контроллер считывается с регистра FAULT_STATE. Схема переходов состояние ошибок показана на рисунке 1.63. Счетчики локализации сбоя считываются с регистров REC и TEC. Эти счетчики соответствуют счетчикам ошибок передачи и приема, согласно ISO11898-1. CAN контроллер дополнительно предлагает счетчики, различающие ошибки, обнаруженные при номинальной скорости и скорости передачи данных. Счетчик ошибок номинальной скорости считывается с регистра ERR_NORM и увеличивается на 1 при каждой ошибке, обнаруженной при номинальной скорости. Счетчик ошибок скорости передачи

данных считывается с регистра ERR_FD и увеличивается на 1 при каждой ошибке, обнаруженной при скорости передачи данных.

Все четыре счетчика (REC, TEC, ERR_NORM, ERR_FD) управляются ПО. Поскольку эта функция напрямую влияет на соответствие ядра CAN контроллер стандарту ISO11898-1, она может работать только при MODE[TSTM] = 1 (в тестовом режиме). Все счетчики можно настроить с помощью ПО посредством регистра CTR_PRES. Пороговые значения для лимита предупреждений об ошибках и переход на режим без учета ошибок приведены в регистрах EWL и ERP. По умолчанию EWL и ERP соответствуют ISO11898-1. В тестовом режиме (MODE[TSTM] = 1) регистры EWL ERP поддерживают запись.

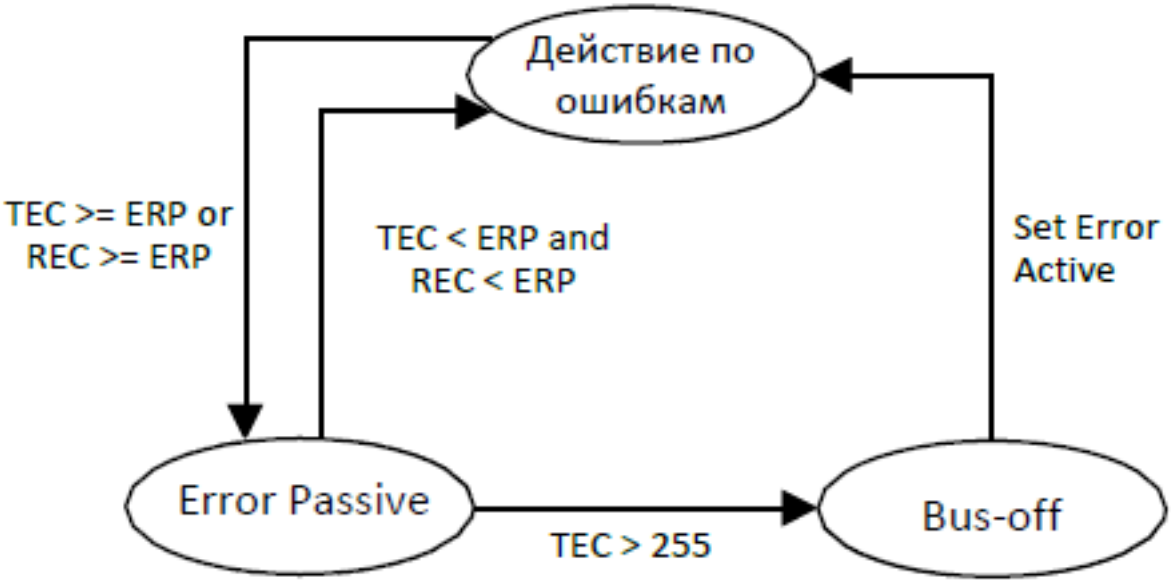


Рисунок 1.63 – Локализация сбоя

1.3.9.5.4.12 Прерывания

CAN контроллер создает прерывания из различных источников. Каждый источник прерываний имеет три параметра:

– Interrupt mask — задается посредством INT_MASK_SET, сбрасывается INT_MASK_CLR.

– Interrupt enable — задается посредством Set by INT_ENA_SET, сбрасывается INT_ENA_CLR.

– Interrupt status — задается аппаратным обеспечением при появлении события, сбрасывается посредством записи в INT_STAT.

Взаимосвязь между параметрами прерываний показана на рисунке 1.64. Состояние прерывания задается, когда в CAN контроллер выполняется определенное условие. Чтобы настроить состояние прерывания, его соответствующий бит маски прерывания должен быть равен 0 (прерывание без маски). Прерывание создается, если состояние прерывания настроено, и соответствующее прерывание включено. Состояние прерывания можно прочитать в CAN контроллер посредством регистра INT_STAT. Обратите внимание, что если аппаратная часть задает состояние прерывания одновременно с операцией сброса посредством записи в регистр INT_STAT, прерывание остается заданным (этот параметр имеет приоритет).

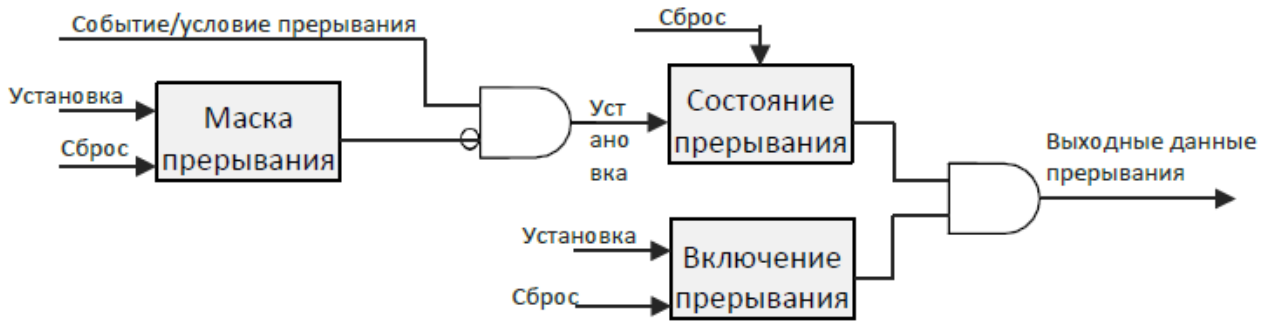


Рисунок 1.64 – Прерывания

1.3.9.5.4.12.1 Передача и прием кадров

Когда CAN контроллер успешно передает кадр CAN (не было кадров ошибки вплоть до конца кадра), создается прерывание TX (INT_STAT[TXI]). Когда CAN контроллер успешно получает кадр CAN (не было кадров ошибки вплоть до одного бита перед концом кадра), создается прерывание RX (INT_STAT[RXI]).

1.3.9.5.4.12.2 Локализация сбоя

Когда счетчик ошибок передачи (TEC) или счетчик ошибок приема (REC) достигают значения в регистре EWL, создается прерывание лимита ошибок (INT_STAT[EWLI]). Когда меняется состояние локализации сбоя CAN контроллер, создается прерывание состояния локализации сбоя(INT_STAT[FCSI]). Этот бит задается при любом изменении состояния локализации сбоя (даже при отключенной шине при действии по ошибкам).

1.3.9.5.4.12.3 TXT-буферы и RX-буфер

В случае переполнения RX-буфера создается прерывание переполнения данных (INT_STAT[DOI]). Когда RX-буфер заполнен, выдается прерывание полного RX-буфера (INT_STAT[RXFI]). Если RX-буфер по-прежнему полон после очистки INT_STAT[RXFI],

прерывание генерируется снова. Если в RX-буфере хранится хотя бы один кадр CAN, генерируется прерывание, указывающее, что RX-буфер не пустой (INT_STAT[RBNEI]). Когда TXT-буфер переходит из состояний "Ready", "TX in progress" или "Abort in progress" в состояния "TX OK", "Aborted" или "TX failed", генерируется прерывание аппаратного изменения TXT-буфера (INT_STAT[TXBHCI]).

1.3.9.5.4.12.4 Кадр ошибки и перегрузки

Когда CAN контроллер начинает передачу кадра ошибки, создается прерывание ошибки шины (INT_STAT[BEI]). Когда начинается передача кадра перегрузки, создается прерывание кадра перегрузки (INT_STAT[OFI]).

1.3.9.5.4.12.5 Другое

Когда CAN контроллер переключает скорости передачи данных на шине CAN, создается прерывание переключения скорости передачи данных (INT_STAT[BSI]). Когда CAN контроллер теряет доступ к общей шине, создается прерывание потери доступа к общей шине (INT_STAT[ALI]).

1.3.9.5.4.13 Отказоустойчивость

CAN контроллер реализует следующие механизмы отказоустойчивости:

– Защита по четности в RX-буфера

– Защита по четности в TXT-буфера

– Резервный режим TXT-буфера (MODE[TXBBM]).

Для работы этих механизмов должны быть выполнены следующие условия:

*CAN контроллер должен поддерживать защиту по четности (STATUS[SPRT]=1). Если STATUS[SPRT]=0, CAN контроллер не содержит защиту по четности (не было синтеза с этим механизмом), и этот раздел не актуален. * Программный драйвер задает SETTINGS[PCHKE] = 1. Этот бит включает обнаружение ошибки четности. SW изменит SET- TINGS[PCHKE] только при SETTINGS[ENA] = 0.

1.3.9.5.4.13.1 Защита по четности в RX-буфера

Когда CAN контроллер получает кадр CAN и сохраняет его в RX-буфера, он добавляет один бит четности в каждое слово RX-буфера. Когда программный драйвер считывает кадр из RX-буфера, для проверки на ошибку четности в кадре он может прочитать кадр STATUS[RXPRE]. CAN контроллер задает STATUS[RXPRE] при каждом чтении из регистра

RX_DATA, если бит четности в слове, считываемом из RX-буфера, не соответствует рассчитанному биту четности.

Поскольку ложный единичный отказ (SEU) в RX-буфера может изменить слово FRAME_FORMAT_W[DLC] полученного кадра в RX-буфера, это может ограничить длину кадра RX с точки зрения программного драйвера, поэтому необходимо ввести RX-буфер в состояние непоследовательности, где программный драйвер прочитал только часть полученного кадра. В такой ситуации все дальнейшие кадры, считываемые с RX-буфера, будут искажены. Во избежание этого при чтении RX-кадров из RX-буфера можно следовать процдеуре ниже с упором на проверку четности:

```
#define CAN_BASE 0x12000000

#define STATUS_ADDR (CAN_BASE + 0x8)

#define COMMAND_ADDR (CAN_BASE + 0xC)

#define RX_STATUS_ADDR (CAN_BASE + 0x68)

#define RX_DATA_ADDR (CAN_BASE + 0x6C)

/* Считать кадр в  RX-буфера и проверить ошибку четности.*/

uint8_t data[64];

uint32_t tmp;

uint32_t ffw = can_read_32(RX_DATA_ADDR);

uint32_t id = can_read_32(RX_DATA_ADDR);

uint32_t ts_l = can_read_32(RX_DATA_ADDR);

uint32_t ts_h = can_read_32(RX_DATA_ADDR);

/* Если ошибка четности в FRAME_FORMAT_W, RWCNT может быть ненадежным. */

if (can_read_32(STATUS_ADDR) >> 10) & 0x1)

goto rx_buffer_flush;

uint32_t rwcnt = (ffw >> 11) & 0x1F;

for(int i = 0; i < rwcnt; i++)\{

tmp = can_read_32(RX_DATA_ADDR);

data[i*4] = tmp & 0xFF;
```

```
data[i*4+1] = (tmp >> 8) & 0xFF;

data[i*4+2] = (tmp >> 16) & 0xFF;

data[i*4+3] = (tmp >> 24) & 0xFF;

if (can_read_32(STATUS_ADDR) >> 10) & 0x1)

goto parity_err_handler;

}

return RX_FRAME_READ_OK;

/* Считать искаженный RX-кадр до начала нового кадра. */

parity_err_handler:

int i=0;

while (i < 16) \{

if (((can_read_32(RX_STATUS_ADDR) >> 2) & 0x1) == 0)\{

can_write_32(COMMAND_ADDR, 0x200);

return RX_FRAME_DROPPED;

}

i++;

can_read_32(RX_DATA_ADDR);

}

/* На этом этапе возникает риск, что RX-буфер не находится в состоянии последовательност
и. */ rx_buffer_flush:

can_write_32(COMMAND_ADDR, 0x202);

return RX_BUFFER_RESET;
```

Примечание В примере ниже предполагается, что RX-буфер считывается в автоматическом режиме (MODE[RXBAM]=1). Однако если одно слово из RX-буфера будет прочитано посредством, к примеру 4 x 8-битного доступа в MODE[RXBAM]=0, CAN контроллер задает STATUS[RXPE] при каждом чтении RX_DATA, содержащем ошибку четности.

Примечание Когда ПО выдает COMMAND[CRXPE]=1, бит STATUS[RXPE] очищается.

Примечание Когда SETTINGS[PCHKE] = 0, CAN контроллер игнорирует ошибку четности, обнаруженную в RX-буфере (STATUS[RXPE] не задан, COMMAND[CRXPE] не приносит результат).

Примечание При записи в RX-буфера через регистры тестирования (см. пп 1.3.9.5.4.16.5) не обновляется бит четности соответствующего слова RX-буфера. См. пп 1.3.9.5.4.13.4.

1.3.9.5.4.13.2 Защита по четности в TXT-буфера

Когда ПО сохраняет кадр CAN в TXT-буфере, CAN контроллер добавляет бит четности к каждому слову в TXT-буфера. Когда CAN контроллер пытается передать кадр из TXT-буфера и обнаруживает ошибку четности в TXT-буфера, выполняются следующие действия:

1) Если CAN контроллер обнаруживает ошибку четности в FRAME_FORMAT_W, IDENTIFIER_W, TIMESTAMP_U_W или TIMES- TAMP_L_W, он не пытается передать кадр CAN.

2) Если CAN контроллер не обнаруживает ошибку четности в словах TXT-буфера, упомянутых в предыдущем пункте, он пытается передать кадр CAN.

3) Если во время передачи кадра CAN CAN контроллер обнаруживает ошибку четности в любом из слов DATA_1_4_W - DATA_61_64_W, из которых передает полезные данные кадра CAN, он начинает передавать кадр ошибки.

Если CAN контроллер обнаруживает ошибку четности в TXT-буфера, как описано на шагах 1 или 3, этот TXT-буфер переходит в состояние "Parity Error", как показано на рисунке 1.57, и задается бит STATUS[TXPE].

Примечание Если CAN контроллер обнаружит ошибку четности в TXT-буфере, ПО запишет целый кадр CAN в TXT-буфер, прежде чем попытается использовать его в дальнейших передачах.

Примечание Когда ПО выдает COMMAND[CTXPE]=1, бит STATUS[TXPE] очищается.

Примечание Когда SETTINGS[PCHKE]=0, CAN контроллер игнорирует ошибку четности, обнаруженную в TXT-буферах (STATUS[TXPE] не задан, COMMAND[CTXPE] не приносит результат, TXT-буферы не переходят в состояние ошибки четности).

Примечание При записи в TXT-буфера через регистры тестирования (см. пп 1.3.9.5.4.16.5) не обновляется бит четности соответствующего слова TXT-буфера. См. пп 1.3.9.5.4.13.4.

Примечание CAN контроллер не обнаруживает ошибок четности в FRAME_TEST_W. Цель FRAME_TEST_W — преднамеренно исказить передаваемый кадр (например, для тестирования сценариев ошибки в шине CAN). Эта функция вряд ли принесет пользу в приложениях, требующих защиту по четности (высоконадежные приложения, разработчики которых хотят добиться отказоустойчивости).

1.3.9.5.4.13.3 Резервный режим TXT-буфера

Когда MODE[TXBBM]=1, CAN контроллер работает в резервном режиме TXT-буфера. В резервном режиме TXT-буферы со смежными индексами формируют пары (например, если TXTB_INFO[TXT_BUFFER_COUNT]=8 (CAN контроллер содержит 8 TXT-буферов), возможно 4 пары TXT-буферов: 1-2, 3-4, 5-6, 7-8) как показано на рисунке 1.65.



Рисунок 1.65 – Пары TXT-буферов

Работа CAN контроллер в резервном режиме TXT-буфера обеспечивает дополнительную отказоустойчивость, т. к. в случае ошибки четности в "оригинальном" TXT-буфере в паре TXT-буфер с более высоким индексом служит как "резервный". Принцип работы CAN контроллер в резервном режиме TXT-буфера показан на рисунке 1.66 и описывается в этом разделе.

Когда MODE[TXBBM]=1, и CAN контроллер обнаруживает ошибку четности в "оригинальном" TXT-буфере, этот TXT-буфер переходит в состояние "Parity Error", а CAN контроллер пытается передать кадр из своего "резервного" TXT-буфера (например, если CAN контроллер обнаруживает ошибку четности в TXT-буфере 3, он пытается передать кадр из TXT-буфера 4). Если CAN контроллер удается успешно передать кадр из "оригинального" TXT-буфера, его "резервный" буфер переходит в состояние "Aborted" (CAN контроллер не передает кадр в "резервном" TXT-буфере).

Когда CAN контроллер передает кадр из "резервного" TXT-буфера из-за ошибки четности в "оригинальном" TXT-буфере и также обнаруживает ошибку четности в "резервном" TXT-буфере, CAN контроллер задает бит STATUS[TXDPE] (двойная ошибка четности).

Когда CAN контроллер работает в резервном режиме TXT-буфера, программный контроль над TXT-буферами имеет следующие отличия от сценария MODE[TXBBM]=0:

– Оба TXT-буфера в паре имеют равные приоритеты, они получают их от TX_PRIORITY[TX*P] "оригинального" TXT-буфера (например, приоритет TXT-буферов 1 и 2 предоставляется посредством TX_PRIORITY[TX1P], TX_PRIORITY[TX2P] не влияет на приоритет). *CAN контроллер автоматически применяет команды от ПО к каждому "оригинальному" TXT-буферу, а также к соответствующему "резервному" TXT-буферу (например, если ПО дает команду TXT-буферу 1 (TX_COMMAND[TXB1] = 1), CAN контроллер автоматически применяет ее к TXT-буферу 2).

Предполагается, что при попытке отправить кадр CAN ПО сохраняет равное количество кадров CAN в обоих TXT-буферах из пары. В таком случае резервный режим TXT-буфера дает следующий результат: Если в оригинальном TXT-буфере происходит ошибка четности, аналогичный кадр передается из "резервного" TXT-буфера.

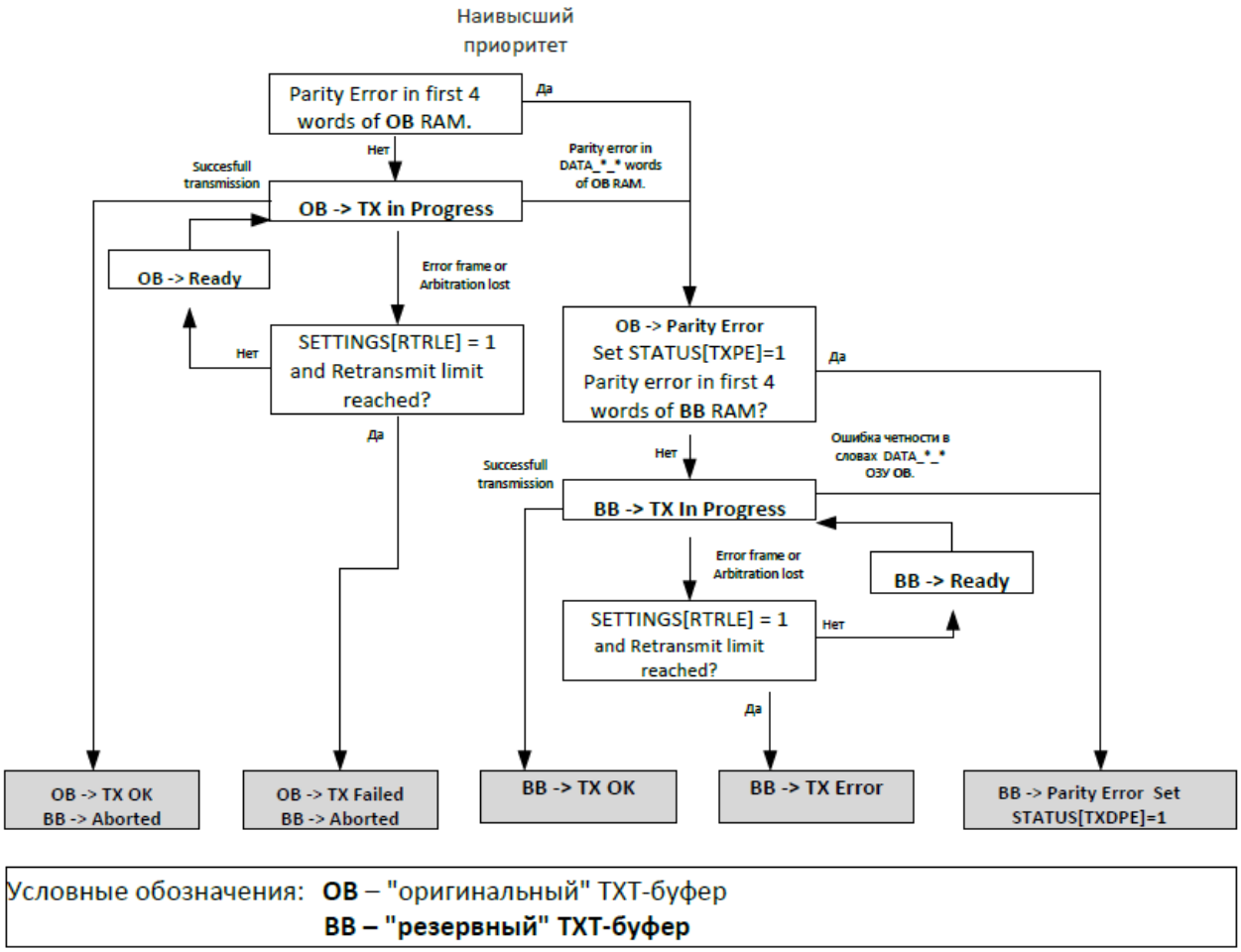


Рисунок 1.66 – Работа в резервном режиме TXT-буфера

Примечание Хранение одинаковых кадров в TXT-буферах обеспечивается отдельными записями в каждый буфер. CAN контроллер не сохраняет один и тот же кадр в оба TXT-буфера автоматически во избежание возможного SEU (единичный отказ) в момент сохранения кадра в TXT-буфер. В случае такого SEU кадр может сохраниться в оба TXT-буфера с ошибкой четности.

Примечание ПО не требуется сохранять равные кадры в оба TXT-буфера из пары. Если ошибка четности случится в "оригинальном" TXT-буфере, ПО может просто сохранить любой подлежащий передаче кадр в "резервный" TXT-буфер.

Примечание ПО должно настроить MODE[TXBBM] = 1 вместе с SETTINGS[PCHKE] = 1. Если MODE[TXBBM] = 1 одновременно с SETTINGS[PCHKE] = 0, CAN контроллер игнорирует ошибки четности в "оригинальных" TXT-буферах и никогда не передает кадр из "резервных" TXT-буферов.

Примечание Если CAN контроллер обнаруживает ошибку четности в "оригинальном" TXT-буфере во время передачи кадра CAN, а другой TXT-буфер с более высоким приоритетом, чем у выбранной пары, переходит в состояние "Ready" (из-за того, что ПО дает команду "Set Ready"), во время следующей передачи CAN контроллер попытается передать кадр из TXT-буфера с более высоким приоритетом (игнорируя "резервный" TXT-буфер).

Примечание Резервный режим TXT-буфера поддерживается, только когда CAN контроллер содержит четное количество TXT-буферов. Если CAN контроллер содержит нечетное количество TXT-буферов, один из них останется без "резервного" буфера. В этом случае ПО не будет использовать свободный "оригинальный" TXT-буфер при MODE[TXBBM] = 1. Если этот TXT-буфер доступен при MODE[TXBBM], алгоритм работы CAN контроллер не определен.

1.3.9.5.4.13.4 Тестирование защиты по четности

Если CAN контроллер включает область память для регистров тестирования (STATUS[STRGS] = 1) (см. раздел 3), запись в TXT-буфер / RX-буфера через эту область памяти не обновляет значение бита четности, хранящееся в каждом слове памяти TXT-буфера / RX-буфера. Благодаря этому в TXT-буфере / RX-буфера становится возможной внутрикристальная проверка обнаружения четности. Проверить возможность обнаружения четности на RX-буфера можно следующим образом:

1) ПО считывает память RX-буфера через область память регистров тестирования (подробнее об этой процедуре).

2) ПО меняет бит в слове памяти кадра CAN, прочитанного на предыдущем шаге, и сохраняет этот измененный кадр в RX-буфера посредством области памяти регистров тестирования.

3) ПО считывает кадр из RX-буфера через регистр RX_DATA, а затем считывает STATUS[RXPE]. Если STATUS[RXPE] = 1, то алгоритм обнаружения ошибка четности в RX-буфера будет работать правильно.

Проверить возможность обнаружения четности на TXT-буфера можно следующим образом:

1) ПО вставляет кадр CAN в TXT-буфер.

2) ПО считывает такой кадр посредством области памяти регистров тестирования, меняет бит в случайном слове и сохраняет это слово в области памяти регистров тестирования.

3) ПО отправляет команду **Set ready** (посредством регистра TX_COMMAND) в TXT-буфер, где хранился кадр CAN на двух предыдущих шагах.

4) CAN контроллер пытается отправить кадр из этого TXT-буфера (с предположением, что ни один TXT-буфер не находится в состоянии "Ready"). При чтении слова памяти, содержащего инвертирование разрядов, CAN контроллер отправляет кадр ошибки и настраивает STA- TUS[TXPE]=1.

5) ПО считывает STATUS[TXPE]. Если STATUS[TXPE]=1, алгоритм обнаружения четности в TXT-буфера работает правильно.

Примечание Когда ПО инвертирует случайный бит в TXT-буфера, это должно произойти в словах памяти, которые считывает CAN контроллер при попытке отправить кадр. Например, если ПО инвертирует бит в DATA_61_64_W, но вставленный кадр CAN содержит только 8 байтов данных (FRAME_FORMAT_W[DLC]=1000), CAN контроллер не попытается прочитать слово DATA_61_64_W из TXT-буфера (будут прочитаны только DATA_1_4_W и DATA_5_8_W), поэтому бит STATUS[TXPE] не будет выставлен.

Примечание При получении доступа к RX-буфера / TXT-буфера посредством области памяти регистров тестирования TSTCTRL[TMENA] (бит включения доступа для тестирования) должен быть настроен только при выполнении доступа, а не во время работы контроллера. Как правило, такой доступ включает следующее:

1) Настройка TSTCTRL[TMENA]=1.

2) Чтение /запись RX-буфера / TXT-буфера посредством регистров TST_DEST, TST_WDATA, TSTCTRL, TST_RDATA.

3) Настройка TSTCTRL[TMENA]=0.

1.3.9.5.4.14 Особые режимы

1.3.9.5.4.14.1 Режим обратной связи

В режиме обратной связи любой кадр CAN, переданный CAN контроллер из любого из TXT-буферов, сохраняется в RX-буфере, если передача прошла успешно (и кадр проходит фильтры кадра). Обратите внимание, что хотя CAN контроллер сохраняет собственные отправленные кадры в RX-буфере, он все равно действует как передатчик, а значит не подтверждает собственный кадр! Для успешной передачи кадр должен быть подтвержден другим узлом на шине CAN. Режим обратной связи включен когда SETTINGS[ILBP]=1. SETTINGS[ILBP] может быть изменен, только когда CAN контроллер отключен (SETTINGS[ENA] = 0).

1.3.9.5.4.14.2 Режим самотестирования

В режиме самотестирования CAN контроллер признает переданный кадр действительным, если не получает доминантный бит во время периода ACK. Этот режим можно использовать вместе с режимом обратной связи для проверки работы CAN контроллер, когда это единственный узел на шине. Режим самотестирования включается при MODE[STM]=1. MODE[STM] может быть изменен, только когда CAN контроллер отключен (SETTINGS[ENA] = 0).

1.3.9.5.4.14.3 Режим запрета подтверждения

Когда включен режим запрета подтверждения, CAN контроллер, действующий как получатель кадра CAN, не передает доминантный бит во время периода ACK, даже если полученный CRC совпадает с рассчитанным CRC. Режим запрета подтверждения включается при MODE[ACF] = 1. MODE[ACM] может быть изменен, только когда CAN контроллер отключен (SETTINGS[ENA] = 0).

1.3.9.5.4.14.4 Режим самоподтверждения

Когда включен режим самоподтверждения, CAN контроллер отправляет доминантный бит ACK, даже если передает кадр CAN. Режим самоподтверждения включается при MODE[SAM] = 1. MODE[SAM] может быть изменен только при SETTINGS[ENA] = 0.

1.3.9.5.4.14.5 Режим отслеживания шины

В режиме отслеживания шины CAN контроллер не передает, а только получает кадры CAN. Если в TXT-буфер вносится кадр CAN и дается команда **Set ready**, кадр не передается, а TXT-буфер немедленно переходит в состояние "TX failed". В режиме отслеживания шины CAN контроллер не передает доминирующие биты в шине. Если требуется передать в шину доминантный бит (например, ACK или кадр ошибки), то он перенаправляется, чтобы CAN контроллер получил этот бит, а остальные узлы в шине CAN не видели этот доминантный бит. Режим отслеживания шины включается при MODE[BMM]=1. MODE[BMM] может быть изменен, только когда CAN контроллер отключен (SETTINGS[ENA] = 0).

1.3.9.5.4.14.6 Режим ограниченной работы

В режиме ограниченной работы CAN контроллер может получать кадры в шине CAN, но не передает никакие кадры. Если в TXT-буфер записывается кадр CAN и дается команда **Set ready**, кадр не передается, а TXT-буфер немедленно переходит в состояние "TX failed". В

режиме ограниченной работы CAN контроллер передает ACK действительным кадрам, но не отправляет ни кадры ошибки, ни кадры перегрузки. Если будет обнаружено состояние ошибки или перегрузки, CAN контроллер входит в состояние интеграции шины и ждет 11 последовательных рецессивных битов. В режиме ограниченной работы счетчики REC и TEC не меняются, поэтому CAN контроллер всегда остается в состоянии действия по ошибкам. Режим ограниченной работы включается при MODE[ROM] = 1. MODE[ROM] может быть изменен, только когда CAN контроллер отключен (SETTINGS[ENA] = 0).

1.3.9.5.4.14.7 Режим тестирования

У CAN контроллер есть режим тестирования, который включается при MODE[TSTM] = 1. В режиме тестирования CAN контроллер выполняет следующие функции:

– Регистр ERP доступен для записи, что позволяет настраивать пороговое значение для перехода от состояния действия по ошибкам к состоянию бездействия.

– Регистр EWL доступен для записи, что позволяет настраивать пороговое значение для создания прерывания лимита для предупреждений об ошибке (INT[EWLI]).

– Регистр CTR_PRES доступен для записи, поэтому программный драйвер может модифицировать все счетчики ошибок.

Примечание Режим тестирования следует использовать только для целей отладки / разработки (например, тестирования алгоритма высших слоев в состоянии бездействия при ошибках). Его не следует использовать во время штатной работы CAN контроллер.

1.3.9.5.4.15 Искажение переданных кадров CAN

CAN контроллер предоставляет следующие способы для искажения/изменения переданного кадра CAN:

– Инвертировать бит поля CRC.

– Инвертировать бит поля счетчика заполнителя или поля четности заполнителя.

– Заменить DLC на произвольное значение.

Все функции по искажению переданных кадров CAN настраиваются для каждого переданного кадра в слове памяти FRAME_TEST_W в TXT-буфере, подробнее см. в подразделах ниже. Эти функции доступны только в режиме тестирования (MODE[TSTM]=1). Если MODE[TSTM]=0,CAN контроллер игнорирует эту конфигурацию и передает неискаженные файлы (как при штатной работе). Искажение касается только передаваемых кадров, еслиCAN

контроллер является получателем, он не искажает этот кадр (он не искажает кадры, переданные другими узлами CAN в сети).

Примечание Искажение бита или замена поля бита на альтернативное значение происходит до наполнения бита, поэтому инвертирование бита может изменить длину кадра из-за дополнительного/удаленного бита наполнения.

Примечание Чтобы несколько раз повторить передачу кадра с искаженным битом, используйте алгоритм "Ограничение повторной передачи", см. таблицу 1.287.

Примечание Слово FRAME_TEST_W кадра CAN есть только в TXT-буферах, его нет в RX-буфере (самый длинный кадр CAN в RX-буфере включает 20 слов, а не 21).

1.3.9.5.4.15.1 Инвертирование бита поля CRC

Когда FRAME_TEST_W[FCRC] = 1,CAN контроллер передает инвертированный бит в расположении бита поля CRC, указанном FRAME_TEST_W[TPR

Например:

– FRAME_TEST_W[TPRM] = 0x0 → Бит в расположении 0 в поле CRC (первый бит поля CRC) передается с противоположным значением.

– FRAME_TEST_W[TPRM] = 0xE → Бит в расположении 14 в поле CRC (15-й бит поля CRC) передается с противоположным значением.

Примечание Если FRAME_TEST_W[FIND] больше длины поля CRC, бит не инвертируется.

1.3.9.5.4.15.2 Инвертирование битового заполнения

Когда FRAME_TEST_W[FSTC] = 1,CAN контроллер передает инвертированный бит заполнения в расположении, указанном FRAME_TEST_W[TPRM]. Например:

– FRAME_FORMAT_W[TPRM] = 0x0 → Первый бит заполнения передается с противоположным значением.

– FRAME_FORMAT_W[TPRM] = 0x2 → Третий бит заполнения передается с противоположным значением.

– FRAME_FORMAT_W[TPRM] = 0x3 → Бит четности наполнения передается с противоположным значением.

1.3.9.5.4.15.3 Замена DLC на произвольное значение

Когда FRAME_TEST_W[SDLC] = 1, в поле кода длины данных кадра CAN вместо FRAME_TEST_W[DLC]

передаются биты FRAME_TEST_W[CPRM][3:0]. Количество переданных байтов данных производится из поля FRAME_TEST_W[DLC].

1.3.9.5.4.16 Прочие характеристики

1.3.9.5.4.16.1 Код ошибки

CAN контроллер содержит регистр кода ошибки. В этом регистре хранится тип и расположение последней ошибки в шине CAN, которая привела к передаче кадра ошибки. Кода ошибки обновляется в той контрольной точке бита, когда была обнаружена ошибка. Код ошибки считывается посредством регистра ERR_CAPT. По стандартуCAN контроллер различные типы ошибок не являются взаимоисключающими (например, ошибка бита и ошибка наполнения могут произойти одновременно, когда значение бита наполнения искажено на противоположное значение). В таком случае захват кода ошибки захватывает только один тип ошибки с наивысшим приоритетом. Приоритеты типов ошибки определяются следующим образом (ошибка формы имеет наивысший приоритет):

Приоритет	**1**	**2**	**3**	**4**	**5**
Тип ошибки	Ошибка формы	Ошибка бита	Ошибка CRC	Ошибка ACK	Ошибка наполнения

Примечание Ошибка наполнения, которая случилась во время наполнения фиксированного бита кадраCAN контроллер, отражается в регистре захвата кода ошибки как ошибка формы.

Примечание В описанном выше порядке приоритета ошибки есть исключение. Если во время выборки поля доступа к общей шине и рецессивного значения отправляется доминантный бит, захватывается ошибка наполнения, а не ошибка бита.

1.3.9.5.4.16.2 Потери арбитража к общей шине

CAN контроллер содержит регистр потери арбитража к общей шине (ALC). Этот регистр хранит расположение бита в кадре CAN, гдеCAN контроллер в последний раз потерял доступ к общей шине.

1.3.9.5.4.16.3 Счетчики трафика

CAN контроллер может измерять кадры CAN, передаваемые/полученные на шине CAN. При каждом успешно переданном кадре CAN регистр TX_COUNTER увеличивается на 1. При каждом успешно полученном кадре CAN регистр RX_COUNTER увеличивается на 1. Регистр TX_COUNTER может быть очищен с помощью команды COMMAND[TXFCRST]=1. Регистр

RX_COUNTER может быть очищен с помощью команды COMMAND[RXFCRST]=1. КогдаCAN контроллер находится в режиме обратной связи и переданный кадр сохраняется в RX-буфере, регистр RX_COUNTER тоже увеличивается. Счетчики трафика не являются обязательными дляCAN контроллер. Наличие счетчиков трафика можно определить по биту STATUS[STCNT].

1.3.9.5.4.16.4 Регистр отладки

CAN контроллер включает регистр отладки (DEBUG_REGISTER), который напрямую отражает часть/поле кадра CAN, который передается/принимается на данный момент.

1.3.9.5.4.16.5 Тестируемость памяти

CAN контроллер поддерживает производственное тестирование своих внутренних компонентов памяти (TXT-буфера и RX-буфера) через область памяти регистров тестирования. Подробнее о тестировании памяти.

1.3.9.5.5 Карта регистров

CAN контроллер — это 32-битный компонент, поддерживающий 8, 16 или 32-битный доступ. Неупорядоченный доступ не поддерживается. Доступ к биту или половине слова поддерживается. Память упорядочена в обратном порядке байтов. Запись в область памяти только для чтения не дает результат. Чтение из области памяти только для записи может выдавать неопределенные значения. Карта памяти CAN контроллера состоит из следующих областей памяти:

Область памяти	**Смещение адреса**
Управляющие регистры	0x000
TXT-буфер 1	0x100
TXT-буфер 2	0x200
TXT-буфер 3	0x300
TXT-буфер 4	0x400
TXT-буфер 5	0x500
TXT-буфер 6	0x600
TXT-буфер 7	0x700
TXT-буфер 8	0x800
Регистры тестирования	0x900

1.3.9.5.5.1 Управляющие регистры

Область памяти управляющих регистров.

Биты [31:24]	Биты [23:16]	Биты [15:8]	Биты [7:0]	Смещение адреса
VERSION		DEVICE_ID		0x0
SETTINGS		MODE		0x4
STATUS				0x8
COMMAND				0xC
Зарезервировано		INT_STAT		0x10
Зарезервировано		INT_ENA_SET		0x14
Зарезервировано		INT_ENA_CLR		0x18
Зарезервировано		INT_MASK_SET		0x1C
Зарезервировано		INT_MASK_CLR		0x20
BTR				0x24
BTR_FD				0x28
FAULT_STATE		ERP	EWL	0x2C
TEC		REC		0x30
ERR_FD		ERR_NORM		0x34
CTR_PRES				0x38
FILTER_A_MASK				0x3C
FILTER_A_VAL				0x40
FILTER_B_MASK				0x44
FILTER_B_VAL				0x48
FILTER_C_MASK				0x4C
FILTER_C_VAL				0x50
FILTER_RAN_LOW				0x54
FILTER_RAN_HIGH				0x58
FILTER_STATUS		FILTER_CONTROL		0x5C
RX_MEM_INFO				0x60
RX_POINTERS				0x64
Зарезервировано	RX_SETTINGS	RX_STATUS		0x68
RX_DATA				0x6C
TX_STATUS				0x70
TXTB_INFO		TX_COMMAND		0x74
TX_PRIORITY				0x78
TS_INFO	ALC	RETR_CTR	ERR_CAPT	0x7C
SSP_CFG		TRV_DELAY		0x80
RX_FR_CTR				0x84
TX_FR_CTR				0x88
DEBUG_REGISTER				0x8C
YOLO_REG				0x90
TIMESTAMP_LOW				0x94
TIMESTAMP_HIGH				0x98
Зарезервировано				…

1.3.9.5.5.1.1 DEVICE_ID

Тип: только чтение **Смещение:**

0x0 **Размер:** 2 байта

Идентификатор CAN контроллера. Позволяет проверить, можно ли получить доступ к CAN контроллеру по его базовому адресу.

Битовый индекс	15	14	13	12	11	10	9	8
Имя поля	DEVICE_ID[15:8]							
Значение после сброса	1	1	0	0	1	0	1	0

Битовый индекс	7	6	5	4	3	2	1	0
Имя поля	DEVICE_ID[7:0]							
Значение после сброса	1	1	1	1	1	1	0	1

Идентификатор устройства **DEVICE_ID**

0b1100101011111101 - CAN_ID - Идентификатор CAN контроллера.

1.3.9.5.5.1.2 VERSION

Тип: только чтение **Смещение:**

0x2 **Размер:** 2 байта

Регистр версий. Выдает версию CAN контроллера.

Битовый индекс	15	14	13	12	11	10	9	8
Имя поля	VER_MAJOR							
Значение после сброса	X	X	X	X	X	X	X	X

Битовый индекс	7	6	5	4	3	2	1	0
Имя поля	VER_MINOR							
Значение после сброса	X	X	X	X	X	X	X	X

VER_MINOR Младшая часть версии CAN контроллера. Например, для версии 2.1 это поле имеет значение 0x01.

VER_MAJOR Старшая часть версии CAN контроллера. Например, для версии 2.1 это поле имеет значение 0x02.

1.3.9.5.5.1.3 MODE

Тип: чтение-запись **Смещение:** 0x4 **Размер:** 2 байта

Битовый индекс	15	14	13	12	11	10	9	8
Имя поля	Зарезервировано				SAM	TXBBM	RXBAM	TSTM
Значение после сброса	-	-	-	-	0	0	1	0

Битовый индекс	7	6	5	4	3	2	1	0
Имя поля	ACF	ПЗУ	TTTM	FDE	AFM	STM, ПАР	BMM	RST
Значение после сброса	0	0	0	1	0	0	0	0

RST Программный сброс. Запись логики 1 сбрасывает CAN контроллер. После того как записана логика 1, логику 0 писать не требуется, этот бит очищается автоматически.

BMM Режим отслеживания шины. В этом режиме CAN контроллер получает только кадры и отправляет только рецессивные биты в шине CAN. Когда отправляется доминирующий бит, он перенаправляется без изменения значения шины. Когда включается этот режим, CAN контроллер не будет передавать никакие кадры из TXT-буферов.

0b0 - BMM_DISABLED - Режим отслеживания шины отключен.

0b1 - BMM_ENABLED - Режим отслеживания шины включен.

STM Режим самотестирования. В этом режиме переданный кадр считается действительным, если не было получено доминирующее подтверждение.

0b0 - STM_DISABLED - Режим самотестирования отключен.

0b1 - STM_ENABLED - Режим самотестирования включен.

AFM Режим фильтров приема. Если этот режим включен, в RX-буфере сохраняются только RX-кадры, которые прошли фильтры кадров. Если режим отключен, каждый полученный кадр сохраняется в RX-буфере. Этот бит имеет значение, только если доступен хотя бы один фильтр. В противном случае этот бит резервируется.

0b0 - AFM_DISABLED - Режим фильтра приема отключен

0b1 - AFM_ENABLED - Режим фильтра приема включен

FDE Включить гибкую скорость передачи данных. Когда гибкая скорость передача включена, CAN контроллер распознает кадры CAN контроллере (бит FDF = 1).

0b0 - FDE_DISABLE - Поддержка гибкой скорости передачи данных отключена.

0b1 - FDE_ENABLE - Поддержка гибкой скорости передачи данных включена.

TTTM Режим передачи по времени

0b0 - TTTM_DISABLED -

0b1 - TTTM_ENABLED -

ROM Режим ограниченной работы.

0b0 - ROM_DISABLED - Режим ограниченной работы отключен.

0b1 - ROM_ENABLED - Режим ограниченной работы включен.

ACF Режим запрета подтверждения Когда этот режим включен, подтверждение не отправляется, даже если полученный CRC совпадает с рассчитанным.

0b0 - ACF_DISABLED - Режим запрета подтверждения отключен.

0b1 - ACF_ENABLED - Режим запрета подтверждения включен.

TSTM Режим тестирования. В режиме тестирование у нескольких регистров есть особые функции. Подробнее см. в описании к режиму тестирования.

RXBAM Автоматический режим RX-буфера.

0b0 - RXBAM_DISABLED - Автоматический режим RX-буфера отключен.

0b1 - RXBAM_ENABLED - Автоматический режим RX-буфера включен.

TXBMM Резервный режим TXT-буфера

0b0 - TXBBM_DISABLED - Резервный режим TXT-буфера отключен.

0b1 - TXBBM_ENABLED - Резервный режим TXT-буфера включен.

SAM Режим самоподтверждения.

0b0 - SAM_DISABLE - Не отправлять доминирующий бит ACK, когда CAN контроллер отправляет бит подтверждения.

0b1 - SAM_ENABLE - Не отправлять доминирующий бит ACK, когда CAN контроллер передает кадр CAN.

1.3.9.5.5.1.4 SETTINGS

Тип:	чтение-запись	**Смещение:**	
0x6	**Размер:**	2 байта	

Битовый индекс	15	14	13	12	11	10	9	8
Имя поля	Зарезервировано				PCHKE	FDRF	TBFBO	PEX
Значение после сброса	-	-	-	-	X	0	1	0

Битовый индекс	7	6	5	4	3	2	1	0
Имя поля	NISOFD	ENA	ILBP	RTRTH				RTRLE
Значение после сброса	0	0	0	0	0	0	0	0

RTRLE Включить ограничение повторной передачи. Если этот режим включен, CAN контроллер предпринимает повторную попытку отправки каждого кадр не более RTR_TH раз.

0b0 - RTRLE_DISABLED - Ограничение повторной передачи отключено.

0b1 - RTRLE_ENABLED - Ограничение повторной передачи включено.

RTRTH Пороговое значение ограничения повторной передачи. Максимальное количество попыток повторной передачи, когда включен режим SETTINGS[RTRLE].

ILBP Режим внутренней обратной внутренней связи. Когда этот режим включен, CAN контроллер получает любой кадр, который передает.

0b0 - INT_LOOP_DISABLED - Обратная внутренняя связь отключена.

0b1 - INT_LOOP_ENABLED - Обратная внутренняя связь включена.

ENA Главный бит включения CAN контроллера. Когда этот режим включен, CAN контроллер обменивается данными по шине CAN. Когда режим отключен, он отключен от шины и не принимает участия в обмене данными по шине CAN.

0b0 - CAN_DISABLED - контроллер CAN отключено.

0b1 - CAN_ENABLED - контроллер CAN включено.

NISOFD Без ISO FD. Когда этот бит задан, CAN контроллер соответствует спецификации NON-ISO CAN контроллере (нет поля счетчика наполнения). Этот бит может быть изменен только при SETTINGS[ENA]=0.

0b0 - ISO_FD - Контроллер CAN соответствует спецификации ISO CAN контроллере.

0b1 - NON_ISO_FD - Контроллер CAN соответствует спецификации NON ISO CAN контроллере.

PEX Обработка исключения протокола. Когда этот бит задан, CAN контроллер начинает интеграцию при обнаружении исключения протокола. Этот би может быть изменен только при SETTINGS[ENA] = ’0’.

0b0 - PROTOCOL_EXCEPTION_DISABLED - Обработка исключения протокола отключена.

0b1 - PROTOCOL_EXCEPTION_ENABLED - Обработка исключения протокола включена.

TBFBO Все TXT-буферы должны переходить в состояние "TX failed", когда CAN контроллер отключается от шины.

0b0 - TXTBUF_FAILED_BUS_OFF_DISABLED - TXT-буферы не переходят в состояние "TX failed", когда CAN контроллер отключается от шины.

0b1 - TXTBUF_FAILED_BUS_OFF_ENABLED - TXT-буферы переходят в состояние "TX failed", когда CAN контроллер отключается от шины.

FDRF Фильтры кадров отбрасывают удаленные кадры.

0b0 - DROP_RF_DISABLED - Фильтры кадров принимают кадры RTR.

0b1 - DROP_RF_ENABLED - Фильтры кадров отбрасывают кадры RTR.

PCHKE Включить проверку четности в TXT-буферах и RX-буфере.

1.3.9.5.5.1.5 STATUS

Тип: только чтение **Смещение:** 0x8 **Размер:** 4 байта

Битовый индекс	31	30	29	28	27	26	25	24
Имя поля	Зарезервировано							
Значение после сброса	-	-	-	-	-	-	-	-

Битовый индекс	23	22	21	20	19	18	17	16
Имя поля	Зарезервировано					SPRT	STRGS	STCNT
Значение после сброса	-	-	-	-	-	X	X	X

Битовый индекс	15	14	13	12	11	10	9	8
Имя поля	Зарезервировано				TXDPE	TXPE	RXPE	PEXS
Значение после сброса	-	-	-	-	0	0	0	0

Битовый индекс	7	6	5	4	3	2	1	0
Имя поля	IDLE	EWL	TXS	RXS	EFT	TXNF	DOR	RXNE
Значение после сброса	1	0	0	0	0	1	0	0

RXNE RX-буфер не пуст. Этот бит равен 1, когда хотя бы один кадр хранится в RX-буфере.

DOR Флажок переполнения данных. Этот бит задается, когда кадр отбрасывается, поскольку в RX-буфере недостаточно места. Этот бит может быть очищен с помощью команды COMMAND[RRB].

TXNF Статус TXT-буферов. Этот бит задается, когда хотя бы один TXT-буфер в состоянии "Empty".

EFT В настоящее время передается кадр ошибки.

RXS CAN контроллер является получателем кадра CAN.

TXS CAN контроллер является передатчиком кадра CAN.

EWL Счетчик ошибок TX (TEC) или счетчик ошибок RX (REC) равны или выше лимита предупреждений об ошибках (EWL).

IDLE Шина простаивает (не идет передача/прием кадров) либо CAN контроллер отключен от шины.

PEXS Статус исключения протокола (флажок). Задается при исключении протокола. Очищается посредством команды COMMAND[CPEXS]=1.

RXPE Задается, когда во время чтения кадра CAN от RX-буфера посредством регистра RX_DATA обнаруживается ошибка четности.

TXPE флажок ошибки четности TXT-буферов. Задается, когда во время передачи от TXT-буфера обнаруживается ошибка четности.

TXDPE Двойная ошибка четности TXT-буфера. Задается в резервном режиме TXT-буфера, когда обнаруживается ошибка четности в "резервном" TXT-буфере.

STCNT Поддержка счетчиков трафика. Когда этот бит составляет 1, счетчики трафика включены.

STRGS поддержка регистров тестирования для тестируемости памяти. Когда этот бит составляет 1, регистры тестирования включены.

SPRT Поддержка защиты по четности в каждом слове TXT-буфера и RX-буфера.

1.3.9.5.5.1.6 COMMAND

Тип:	только запись	**Смещение:**
0xC	**Размер:**	4 байта

Позволяет выдавать команды CAN контроллеру. Запись логики 1 в каждый бит дает команду CAN контроллеру. После того как записана логика 1, логику 0 писать не требуется.

Битовый индекс	31	30	29	28	27	26	25	24
Имя поля	Зарезервировано							
Значение после сброса	-	-	-	-	-	-	-	-

Битовый индекс	23	22	21	20	19	18	17	16
Имя поля	Зарезервировано							
Значение после сброса	-	-	-	-	-	-	-	-

Битовый индекс	15	14	13	12	11	10	9	8
Имя поля	Зарезервировано					CTXDPE	CTXPE	CRXPE
Значение после сброса	-	-	-	-	-	0	0	0

Битовый индекс	7	6	5	4	3	2	1	0
Имя поля	CPEXS	TXFCRST	RXFCRST	ERCRST	CDO	RRB	RXRPMV	Зарезервировано
Значение после сброса	0	0	0	0	0	0	X	-

RXRPMV Перемещение указателя чтения RX-буфера.

RRB Сброс RX-буфера. Эта команда очищает RX-буфер и сбрасывает его указатели памяти.

CDO Флажок переполнения открытых данных в RX-буфере.

ERCRST Сброс счетчиков ошибок. Когда блок отключен от шины, выполнение этой команды запрашивает стирание счетчиков TEC и REC после 128 последовательных случаев 11 рецессивных битов. По завершении счетчики TEC и REC стираются, состояние локализации сбоя переходит в действие по ошибкам. Когда блок не отключен от шины либо когда блок отключен от шины, поскольку отключен сам (SETTINGS[ENA]= ’0’), эта команда не дает результат.

RXFCRST Очистить счетчик трафика RX-шины (регистр RX_COUNTER).

TXFCRST Очистить счетчик трафика TX-шины (регистр TX_COUNTER).

CPEXS Очистить статус исключения протокола (STATUS[PEXS]).

CRXPE Очистить флажок STATUS[RXPE].

CTXPE Очистить флажок STATUS[TXPE].

CTXDPE Очистить флажок STATUS[TXDPE].

1.3.9.5.5.1.7 INT_STAT

Type:	однократн. чтение-запись	**Смещение:**
0x10	**Размер:**	2 байта

Регистр состояния прерывания. Чтение этого регистра выдает логику 1 для каждого прерывания. Запись логики 1 в любой бит очищает статус прерывания. Запись логики 0 не дает результата.

Битовый индекс	15	14	13	12	11	10	9	8
Имя поля	Зарезервировано				TXBHCI	RBNEI	BSI	RXFI
Значение после сброса	-	-	-	-	0	0	0	0

Битовый индекс	7	6	5	4	3	2	1	0
Имя поля	OFI	BEI	ALI	FCSI	DOI	EWLI	TXI	RXI
Значение после сброса	0	0	0	0	0	0	0	0

RXI Прерывание полученного кадра.

TXI Прерывание переданного кадра.

EWLI Прерывание лимита предупреждений об ошибке. Это прерывание создается, когда TEC и REC меньше EWL и один из них становится равным или выше EWL, либо когда TEC и REC становятся меньше EWL. Когда прерывание очищается и REC или TEC по-прежнему равны или выше EWL, прерывание не создается заново.

DOI Прерывание переполнения данных. Перед очищением этого прерывания необходимо очистить STATUS[DOR] во избежание повторного прерывания.

FCSI Прерывание изменения состояния локализации сбоя. Прерывание устанавливается, когда узел переходит в состояние бездействия при ошибках (из действия по ошибкам), отключения от шины (от бездействия при ошибках) или действия по ошибкам (из отключения от шины после повторной интеграции или от состояния бездействия при ошибках).

ALI Прерывание потери доступа к общей шине.

BEI Прерывание ошибки шины.

OFI Прерывание кадра перегрузки.

RXFI Прерывание заполненного RX-буфера.

BSI Прерывание изменения скорости битов.

RBNEI Прерывание не пустого RX-буфера. Если очистить это прерывание или не прочитать содержимое RX-буфера посредством RX_DATA, это прерывание будет активировано повторно.

TXBHCI Прерывание аппаратной команды TXT-буфера. Это прерывание создается, когда TXT-буфер получает аппаратную команду от ядра CAN, которая меняет состояние TXT-буфера в "TX OK", "Error" или "Aborted".

1.3.9.5.5.1.8 INT_ENA_SET

Тип:	чтение-запись (одн.)	**Смещение:**
0x14	**Размер:**	2 байта

Установление включения прерывания. Запись логики 1 в бит включает соответствующее прерывание. Запись логики 0 не дает результата. Чтение этого регистра выдает логику 1 для каждого включенного прерывания. Если прерывание захватывается в INT_STAT, включенное прерывание приводит к тому, что CAN контроллер активирует прерывание. Прерывания работают по уровням. Оно остается активным, пока статус прерывания не будет очищен или прерывание не будет отключено.

Битовый индекс	15	14	13	12	11	10	9	8
Имя поля	Зарезервировано				INT_ENA_SET[11:8]			
Значение после сброса	-	-	-	-	0	0	0	0

Битовый индекс	7	6	5	4	3	2	1	0
Имя поля	INT_ENA_SET[7:0]							
Значение после сброса	0	0	0	0	0	0	0	0

INT_ENA_SET Значение бита эквивалентно регистру INT_STAT.

1.3.9.5.5.1.9 INT_ENA_CLR

Тип: только запись **Смещение:** 0x18 **Размер:** 2 байта

Регистр очищения включения прерывания. Запись логики 1 включает соответствующее прерывание. Запись логики 0 не дает результата. Чтение этого регистра не дает результата. Отключенное прерывание не приводит к тому, что CAN контроллер активирует прерывание, даже если оно становлено в регистре статуса прерывания.

Битовый индекс	15	14	13	12	11	10	9	8
Имя поля	Зарезервировано				INT_ENA_CLR[11:8]			
Значение после сброса	-	-	-	-	0	0	0	0

Битовый индекс	7	6	5	4	3	2	1	0
Имя поля	INT_ENA_CLR[7:0]							
Значение после сброса	0	0	0	0	0	0	0	0

INT_ENA_CLR Значение бита эквивалентно регистру INT_STAT.

1.3.9.5.5.1.10 INT_MASK_SET

Тип:	чтение-запись (одн.)	**Смещение:**
0x1C	**Размер:**	2 байта

Установление маски прерывания. Запись логики 1 маскирует соответствующее прерывание. Запись логики 0 не дает результата. Чтение этого регистра выдает логику 1 для каждого маскированного прерывания. В случае маскирования определенного прерывания оно не будет захвачено в регистре INT_STAT, когда будут выполнены внутренние условия для этого прерывания (например, RX-буфер не пуст для RXNEI).

Битовый индекс	15	14	13	12	11	10	9	8
Имя поля	Зарезервировано				INT_MASK_SET[11:8]			
Значение после сброса	-	-	-	-	0	0	0	0

Битовый индекс	7	6	5	4	3	2	1	0
Имя поля	INT_MASK_SET[7:0]							
Значение после сброса	0	0	0	0	0	0	0	0

INT_MASK Значение бита эквивалентно регистру INT_STAT.

1.3.9.5.5.1.11 INT_MASK_CLR

Тип:	только запись	**Смещение:**
0x20	**Размер:**	2 байта

Регистр очищения маски прерывания. Запись логики 1 снимает маску с соответствующего прерывания. Запись логики 0 не дает результата. Чтение этого регистра не дает результата. В случае снятия маски с определенного прерывания оно будет захвачено в регистре INT_STAT, когда будут выполнены внутренние условия для этого прерывания (например, RX-буфер не пуст для RXNEI).

Битовый индекс	15	14	13	12	11	10	9	8
Имя поля	Зарезервировано				INT_MASK_CLR[11:8]			
Значение после сброса	-	-	-	-	0	0	0	0

Битовый индекс	7	6	5	4	3	2	1	0
Имя поля	INT_MASK_CLR[7:0]							
Значение после сброса	0	0	0	0	0	0	0	0

INT_MASK_CLR Значение бита эквивалентно регистру INT_STAT.

1.3.9.5.5.1.12 BTR

Тип: чтение-запись **Смещение:** 0x24 **Размер:** 4 байта

Примечание: Регистр может быть записан, только когда SETTINGS[ENA] = 0, в противном случае запись не принесет результат.

Регистр побитовой синхронизации для номинальной скорости передачи бита.

Битовый индекс	31	30	29	28	27	26	25	24
Имя поля	SJW					BRP[7:5]		
Значение после сброса	0	0	0	1	0	0	0	0

Битовый индекс	23	22	21	20	19	18	17	16
Имя поля	BRP[4:0]					PH2[5:3]		
Значение после сброса	0	1	0	1	0	0	0	0

Битовый индекс	15	14	13	12	11	10	9	8
Имя поля	PH2[2:0]			PH1[5:1]				
Значение после сброса	1	0	1	0	0	0	0	1

Битовый индекс	7	6	5	4	3	2	1	0
Имя поля	PH1[0]	PROP						
Значение после сброса	1	0	0	0	0	1	0	1

PROP Сегмент распространения

PH1 Сегмент фазы 1

PH2 Сегмент фазы 2

BRP Предварительный делитель скорости передачи данных

SJW Ширина скачка синхронизации

1.3.9.5.5.1.13 BTR_FD

Тип:	чтение-запись	**Смещение:**
0x28	**Размер:**	4 байта

Примечание: Регистр может быть записан, только когда SETTINGS[ENA] = 0, в противном случае запись не принесет результат.

Регистр побитовой синхронизации для скорости передачи бита.

Битовый индекс	31	30	29	28	27	26	25	24
Имя поля	SJW_FD					BRP_FD[7:5]		
Значение после сброса	0	0	0	1	0	0	0	0

Битовый индекс	23	22	21	20	19	18	17	16
Имя поля	BRP_FD[4:0]					Зарезервировано	PH2_FD[4:3]	

Значение после сброса	0	0	1	0	0	-	0	0

Битовый индекс	15	14	13	12	11	10	9	8
Имя поля	PH2_FD[2:0]			Зарезерви-ровано	PH1_FD[4:1]			
Значение после сброса	0	1	1	-	0	0	0	1

Битовый индекс	7	6	5	4	3	2	1	0
Имя поля	PH1_FD[0]	Зарезерви-ровано	PROP_FD					
Значение после сброса	1	-	0	0	0	0	1	1

PROP_FD Сегмент распространения

PH1_FD Сегмент фазы 1

PH2_FD Сегмент фазы 2

BRP_FD Предварительный делитель скорости передачи данных

SJW_FD Ширина скачка синхронизации

1.3.9.5.5.1.14 EWL

Тип:	чтение-запись	**Смещение:**
0x2C	**Размер:**	1 байт

Примечание: Регистр может быть записан, только когда MODE[TSTM] = 1, в противном случае запись не принесет результат. Регистр лимита предупреждений об ошибке. Этот регистр может быть изменен только при SETTINGS[ENA]=0.

Битовый индекс	7	6	5	4	3	2	1	0
Имя поля	EW_LIMIT							
Значение после сброса	0	1	1	0	0	0	0	0

EW_LIMIT Лимит предупреждений об ошибке. Если достигнут лимит предупреждений об ошибке, может быть создано прерывание. Лимит предупреждений об ошибке указывает на поврежденную шину. R_FD.

1.3.9.5.5.1.15 ERP

Тип:	чтение-запись	**Смещение:**
0x2D	**Размер:**	1 байт

Примечание: Регистр может быть записан, только когда MODE[TSTM] = 1, в противном случае запись не принесет результат. Регистр лимита бездействия при ошибках. Этот регистр может быть изменен только при SETTINGS[ENA]=0.

Битовый индекс	7	6	5	4	3	2	1	0
Имя поля	ERP_LIMIT							
Значение после сброса	1	0	0	0	0	0	0	0

ERP_LIMIT Лимит бездействия при ошибках. Когда один из счетчиков ошибок (REC/TEC) превышает это значение, состояние локализации сбоя меняется на бездействие при ошибках.

1.3.9.5.5.1.16 FAULT_STATE

Тип:	только чтение	**Смещение:**
0x2E	**Размер:**	2 байта

Состояние локализации сбоя CAN контроллера.

Битовый индекс	15	14	13	12	11	10	9	8
Имя поля	Зарезервировано							
Значение после сброса	-	-	-	-	-	-	-	-

Битовый индекс	7	6	5	4	3	2	1	0
Имя поля	Зарезервировано					BOF	ERP	ERA
Значение после сброса	-	-	-	-	-	0	0	1

ERA Действие по ошибкам

ERP Бездействие при ошибках

BOF Отключение от шины

1.3.9.5.5.1.17 REC

Тип: только чтение **Смещение:** 0x30 **Размер:** 2 байта

Битовый индекс	15	14	13	12	11	10	9	8
	Имя поля	Зарезервировано						
REC_VAL[8]	Значение после сброса	-	-	-	-	-	-	-

Битовый индекс	7	6	5	4	3	2	1	0
Имя поля	REC_VAL[7:0]							

Значение после сброса	0	0	0	0	0	0	0	0

REC_VAL Счетчик ошибок RX (REC).

1.3.9.5.5.1.18 TEC

Тип: только чтение **Смещение:**
0x32 **Размер:** 2 байта

Битовый индекс	15	14	13	12	11	10	9	8
	Имя поля	Зарезервировано						
TEC_VAL[8]	Значение после сброса	-	-	-	-	-	-	-

Битовый индекс	7	6	5	4	3	2	1	0
Имя поля	TEC_VAL[7:0]							
Значение после сброса	0	0	0	0	0	0	0	0

TEC_VAL Счетчик ошибок TX (TEC).

1.3.9.5.5.1.19 ERR_NORM

Тип: только чтение **Смещение:**
0x34 **Размер:** 2 байта

Битовый индекс	15	14	13	12	11	10	9	8
Имя поля	ERR_NORM_VAL[15:8]							
Значение после сброса	0	0	0	0	0	0	0	0

Битовый индекс	7	6	5	4	3	2	1	0
Имя поля	ERR_NORM_VAL[7:0]							
Значение после сброса	0	0	0	0	0	0	0	0

ERR_NORM_VAL Количество ошибок, произошедших при номинальной скорости битов.

1.3.9.5.5.1.20 ERR_FD

Тип: только чтение **Смещение:** 0x36 **Размер:** 2 байта

Битовый индекс	15	14	13	12	11	10	9	8
Имя поля	ERR_FD_VAL[15:8]							
Значение после сброса	0	0	0	0	0	0	0	0

Битовый индекс	7	6	5	4	3	2	1	0
Имя поля	ERR_FD_VAL[7:0]							
Значение после сброса	0	0	0	0	0	0	0	0

ERR_FD_VAL Количество ошибок, произошедших при скорости передачи битов.

1.3.9.5.5.1.21 CTR_PRES

Тип: только запись **Смещение:** 0x38 **Размер:** 4 байта

Примечание: Регистр может быть записан, только когда MODE[TSTM] = 1, в противном случае запись не принесет результат. Регистр предустановки счетчика. Этот регистр позволяет модифицировать счетчики ошибок.

Битовый индекс	31	30	29	28	27	26	25	24
Имя поля	Зарезервировано							

Значение после сброса	-	-	-	-	-	-	-	-

Битовый индекс	23	22	21	20	19	18	17	16
Имя поля	Зарезервировано							
Значение после сброса	-	-	-	-	-	-	-	-

Битовый индекс	15	14	13	12	11	10	9	8
Имя поля	Зарезервировано			EFD	ENORM	PRX	PTX	CTPV[8]
Значение после сброса	-	-	-	0	0	0	0	0

Битовый индекс	7	6	5	4	3	2	1	0
Имя поля	CTPV[7:0]							
Значение после сброса	0	0	0	0	0	0	0	0

CTPV Устанавливаемое значение счетчика.

PTX Предустановленное значение от CTPV для счетчика ошибок TX (TEC).

PRX Предустановленное значение от CTPV для счетчика ошибок RX (REC).

ENORM Стереть счетчик ошибок при номинальной скорости (ERR_NORM).

EFD Стереть счетчик ошибок при скорости передачи данных (ERR_FD).

1.3.9.5.5.1.22 FILTER_A_MASK

Тип: чтение-запись **Смещение:** 0x3C **Размер:** 4 байта

Примечание: Регистр существует, только при значении sup_filt_A = true. В противном случае этот адрес резервируется.

Битовый индекс	31	30	29	28	27	26	25	24
Имя поля	Зарезервировано			BIT_MASK_A_VAL[28:24]				
Значение после сброса	-	-	-	0	0	0	0	0

Битовый индекс	23	22	21	20	19	18	17	16
Имя поля	BIT_MASK_A_VAL[23:16]							
Значение после сброса	0	0	0	0	0	0	0	0

Битовый индекс	15	14	13	12	11	10	9	8
Имя поля	BIT_MASK_A_VAL[15:8]							
Значение после сброса	0	0	0	0	0	0	0	0

Битовый индекс	7	6	5	4	3	2	1	0
Имя поля	BIT_MASK_A_VAL[7:0]							
Значение после сброса	0	0	0	0	0	0	0	0

BIT_MASK_A_VAL Маска фильтра A. Формат идентификатора совпадает с IDENTIFIER_W TXT-буфера или RX-буфера. Если фильтра A нет, записи в этот регистр не дают результаты, и чтение выдает одни нули.

1.3.9.5.5.1.23 FILTER_A_VAL

Тип: чтение-запись **Смещение:** 0x40 **Размер:** 4 байта

Примечание: Регистр существует, только при значении sup_filt_A = true. В противном случае этот адрес резервируется.

Битовый индекс	31	30	29	28	27	26	25	24
Имя поля	Зарезервировано			BIT_VAL_A_VAL[28:24]				
Значение после сброса	-	-	-	0	0	0	0	0

Битовый индекс	23	22	21	20	19	18	17	16
Имя поля	BIT_VAL_A_VAL[23:16]							
Значение после сброса	0	0	0	0	0	0	0	0

Битовый индекс	15	14	13	12	11	10	9	8
Имя поля	BIT_VAL_A_VAL[15:8]							
Значение после сброса	0	0	0	0	0	0	0	0

Битовый индекс	7	6	5	4	3	2	1	0
Имя поля	BIT_VAL_A_VAL[7:0]							
Значение после сброса	0	0	0	0	0	0	0	0

BIT_VAL_A_VAL Значение фильтра A. Формат идентификатора совпадает с IDENTIFIER_W TXT-буфера или RX-буфера.

Если фильтра A нет, записи в этот регистр не дают результаты, и чтение выдает одни нули.

1.3.9.5.5.1.24 FILTER_B_MASK

Тип:	чтение-запись	**Смещение:**
0x44	**Размер:**	4 байта

Примечание: Регистр существует только при значении sup_filt_B = true. В противном случае этот адрес резервируется.

Битовый индекс	31	30	29	28	27	26	25	24
Имя поля	Зарезервировано			BIT_MASK_B_VAL[28:24]				
Значение после сброса	-	-	-	0	0	0	0	0

Битовый индекс	23	22	21	20	19	18	17	16
Имя поля	BIT_MASK_B_VAL[23:16]							
Значение после сброса	0	0	0	0	0	0	0	0

Битовый индекс	15	14	13	12	11	10	9	8
Имя поля	BIT_MASK_B_VAL[15:8]							
Значение после сброса	0	0	0	0	0	0	0	0

Битовый индекс	7	6	5	4	3	2	1	0
Имя поля	BIT_MASK_B_VAL[7:0]							
Значение после сброса	0	0	0	0	0	0	0	0

BIT_MASK_B_VAL Маска фильтра B. Формат идентификатора совпадает с IDENTIFIER_W TXT-буфера или RX-буфера. Если фильтра A нет, записи в этот регистр не дают результаты, и чтение выдает одни нули.

1.3.9.5.5.1.25 FILTER_B_VAL

Тип:	чтение-запись	**Смещение:**
0x48	**Размер:**	4 байта

Примечание: Регистр существует только при значении sup_filt_B = true. В противном случае этот адрес резервируется.

Битовый индекс	31	30	29	28	27	26	25	24
Имя поля	Зарезервировано			BIT_VAL_B_VAL[28:24]				
Значение после сброса	-	-	-	0	0	0	0	0

Битовый индекс	23	22	21	20	19	18	17	16
Имя поля	BIT_VAL_B_VAL[23:16]							
Значение после сброса	0	0	0	0	0	0	0	0

Битовый индекс	15	14	13	12	11	10	9	8
Имя поля	BIT_VAL_B_VAL[15:8]							
Значение после сброса	0	0	0	0	0	0	0	0

Битовый индекс	7	6	5	4	3	2	1	0
Имя поля	BIT_VAL_B_VAL[7:0]							

Значение после сброса	0	0	0	0	0	0	0	0

BIT_VAL_B_VAL Значение фильтра B. Формат идентификатора совпадает с IDENTIFIER_W TXT-буфера или RX-буфера.

Если фильтра A нет, записи в этот регистр не дают результаты, и чтение выдает одни нули.

1.3.9.5.5.1.26 FILTER_C_MASK

Тип:	чтение-запись	**Смещение:**
0x4C	**Размер:**	4 байта

Примечание: Регистр существует только при значении sup_filt_C = true. В противном случае этот адрес резервируется.

Битовый индекс	31	30	29	28	27	26	25	24
Имя поля	Зарезервировано			BIT_MASK_C_VAL[28:24]				
Значение после сброса	-	-	-	0	0	0	0	0

Битовый индекс	23	22	21	20	19	18	17	16
Имя поля	BIT_MASK_C_VAL[23:16]							
Значение после сброса	0	0	0	0	0	0	0	0

Битовый индекс	15	14	13	12	11	10	9	8
Имя поля	BIT_MASK_C_VAL[15:8]							

Значение после сброса	0	0	0	0	0	0	0	0

Битовый индекс	7	6	5	4	3	2	1	0
Имя поля	BIT_MASK_C_VAL[7:0]							
Значение после сброса	0	0	0	0	0	0	0	0

BIT_MASK_C_VAL Маска фильтра C. Формат идентификатора совпадает с IDENTIFIER_W TXT-буфера или RX-буфера. Если фильтра A нет, записи в этот регистр не дают результаты, и чтение выдает одни нули.

1.3.9.5.5.1.27 FILTER_C_VAL

Тип:	чтение-запись	**Смещение:**
0x50	**Размер:**	4 байта

Примечание: Регистр существует только при значении sup_filt_C = true. В противном случае этот адрес резервируется.

Битовый индекс	31	30	29	28	27	26	25	24
Имя поля	Зарезервировано			BIT_VAL_C_VAL[28:24]				
Значение после сброса	-	-	-	0	0	0	0	0

Битовый индекс	23	22	21	20	19	18	17	16
Имя поля	BIT_VAL_C_VAL[23:16]							
Значение после сброса	0	0	0	0	0	0	0	0

Битовый индекс	15	14	13	12	11	10	9	8
Имя поля	BIT_VAL_C_VAL[15:8]							
Значение после сброса	0	0	0	0	0	0	0	0

Битовый индекс	7	6	5	4	3	2	1	0
Имя поля	BIT_VAL_C_VAL[7:0]							
Значение после сброса	0	0	0	0	0	0	0	0

BIT_VAL_C_VAL Значение фильтра C. Формат идентификатора совпадает с IDENTIFIER_W TXT-буфера или RX-буфера.

Если фильтра A нет, записи в этот регистр не дают результаты, и чтение выдает одни нули.

1.3.9.5.5.1.28 FILTER_RAN_LOW

Тип:	чтение-запись	**Смещение:**
0x54	**Размер:**	4 байта

Примечание: Регистр существует только при значении sup_range = true. В противном случае этот адрес резервируется.

Битовый индекс	31	30	29	28	27	26	25	24
Имя поля	Зарезервировано			BIT_RAN_LOW_VAL[28:24]				
Значение после сброса	-	-	-	0	0	0	0	0

Битовый индекс	23	22	21	20	19	18	17	16
Имя поля	BIT_RAN_LOW_VAL[23:16]							
Значение после сброса	0	0	0	0	0	0	0	0

Битовый индекс	15	14	13	12	11	10	9	8
Имя поля	BIT_RAN_LOW_VAL[15:8]							
Значение после сброса	0	0	0	0	0	0	0	0

Битовый индекс	7	6	5	4	3	2	1	0
Имя поля	BIT_RAN_LOW_VAL[7:0]							
Значение после сброса	0	0	0	0	0	0	0	0

BIT_RAN_LOW_VAL Нижний порог диапазона фильтра. Формат идентификатора совпадает с IDENTIFIER_W TXT-буфера или RX-буфера. Если фильтр диапазона не поддерживается, записи в этот регистр не дают результаты, и чтение выдает одни нули.

1.3.9.5.5.1.29 FILTER_RAN_HIGH

Тип:	чтение-запись	**Смещение:**
0x58	**Размер:**	4 байта

Примечание: Регистр существует только при значении sup_range = true. В противном случае этот адрес резервируется.

Битовый индекс	31	30	29	28	27	26	25	24
Имя поля	Зарезервировано			BIT_RAN_HIGH_VAL[28:24]				
Значение после сброса	-	-	-	0	0	0	0	0

Битовый индекс	23	22	21	20	19	18	17	16
Имя поля	BIT_RAN_HIGH_VAL[23:16]							
Значение после сброса	0	0	0	0	0	0	0	0

Битовый индекс	15	14	13	12	11	10	9	8
Имя поля	BIT_RAN_HIGH_VAL[15:8]							
Значение после сброса	0	0	0	0	0	0	0	0

Битовый индекс	7	6	5	4	3	2	1	0
Имя поля	BIT_RAN_HIGH_VAL[7:0]							
Значение после сброса	0	0	0	0	0	0	0	0

BIT_RAN_HIGH_VAL Верхний порог фильтра диапазона. Формат идентификатора совпадает с IDENTIFIER_W TXT-буфера или RX-буфера. Если фильтр диапазона не поддерживается, записи в этот регистр не дают результаты, и чтение выдает одни нули.

1.3.9.5.5.1.30 FILTER_CONTROL

Тип:	чтение-запись	**Смещение:**
0x5C	**Размер:**	2 байта

Регистр управления фильтром. Настраивает фильтры кадров на прием только выбранных типов кадров. Каждый бит активен в логике 1.

Битовый индекс	15	14	13	12	11	10	9	8
Имя поля	FRFE	FRFB	FRNE	FRNB	FCFE	FCFB	FCNE	FCNB
Значение после сброса	0	0	0	0	0	0	0	0

Битовый индекс	7	6	5	4	3	2	1	0
Имя поля	FBFE	FBFB	FBNE	FBNB	FAFE	FAFB	FANE	FANB
Значение после сброса	0	0	0	0	1	1	1	1

FANB Базовый кадр CAN принимается фильтром A.

FANE Расширенный кадр CAN принимается фильтром A.

FAFB Базовый кадр CAN контроллере принимается фильтром A.

FAFE Расширенный кадр CAN контроллере принимается фильтром A.

FBNB Базовый кадр CAN принимается фильтром B.

FBNE Расширенный кадр CAN принимается фильтром B.

FBFB Базовый кадр CAN контроллере принимается фильтром B.

FBFE Расширенный кадр CAN контроллере принимается фильтром B.

FCNB Базовый кадр CAN принимается фильтром C.

FCNE Расширенный кадр CAN принимается фильтром C.

FCFB Базовый кадр CAN контроллере принимается фильтром C.

FCFE Расширенный кадр CAN контроллере принимается фильтром C.

FRNB Базовый кадр CAN принимается фильтром диапазона.

FRNE Расширенный кадр CAN принимается фильтром диапазона.

FRFB Базовый кадр CAN контроллере принимается фильтром диапазона.

FRFE Расширенный кадр CAN контроллере принимается фильтром диапазона.

1.3.9.5.5.1.31 FILTER_STATUS

Тип:	только чтение	**Смещение:**
0x5E	**Размер:**	2 байта

Статус фильтра указывает, доступны ли фильтры кадров в CAN контроллере.

Битовый индекс	15	14	13	12	11	10	9	8
Имя поля	Зарезервировано							
Значение после сброса	-	-	-	-	-	-	-	-

Битовый индекс	7	6	5	4	3	2	1	0
Имя поля	Зарезервировано				SFR	SFC	SFB	SFA
Значение после сброса	-	-	-	-	X	X	X	X

SFA Логика 1, когда доступен фильтр A. В противном случае логика 0.

SFB Логика 1, когда доступен фильтр B. В противном случае логика 0.

SFC Логика 1, когда доступен фильтр C. В противном случае логика 0.

SFR Логика 1, когда доступен фильтр диапазона. В противном случае логика 0.

1.3.9.5.5.1.32 RX_MEM_INFO

Тип: только чтение **Смещение:** 0x60 **Размер:** 4 байта

Битовый индекс	31	30	29	28	27	26	25	24
Имя поля	Зарезервировано			RX_MEM_FREE[12:8]				
Значение после сброса	-	-	-	X	X	X	X	X

Битовый индекс	23	22	21	20	19	18	17	16
Имя поля	RX_MEM_FREE[7:0]							
Значение после сброса	X	X	X	X	X	X	X	X

Битовый индекс	15	14	13	12	11	10	9	8
Имя поля	Зарезервировано			RX_BUFF_SIZE[12:8]				
Значение после сброса	-	-	-	X	X	X	X	X

Битовый индекс	7	6	5	4	3	2	1	0
Имя поля	RX_BUFF_SIZE[7:0]							
Значение после сброса	X	X	X	X	X	X	X	X

RX_BUFF_SIZE Размер RX-буфера в 32-битных словах.

RX_MEM_FREE Количество свободных 32-битных слов в RX-буфере.

1.3.9.5.5.1.33 RX_POINTERS

Тип: только чтение **Смещение:** 0x64 **Размер:** 4 байта

Битовый индекс	31	30	29	28	27	26	25	24
Имя поля	Зарезервировано				RX_RPP[11:8]			
Значение после сброса	-	-	-	-	0	0	0	0

Битовый индекс	23	22	21	20	19	18	17	16
Имя поля	RX_RPP[7:0]							
Значение после сброса	0	0	0	0	0	0	0	0

Битовый индекс	15	14	13	12	11	10	9	8
Имя поля	Зарезервировано				RX_WPP[11:8]			
Значение после сброса	-	-	-	-	0	0	0	0

Битовый индекс	7	6	5	4	3	2	1	0
Имя поля	RX_WPP[7:0]							
Значение после сброса	0	0	0	0	0	0	0	0

RX_WPP Расположение указателя записи в RX-буфере. Указатель записи обновляется при сохранении полученного кадра.

RX_RPP Расположение указателя чтения в RX-буфере. Указатель чтения обновляется при чтении полученного кадра.

1.3.9.5.5.1.34 RX_STATUS

Тип: только чтение **Смещение:** 0x68 **Размер:** 2 байта

Битовый индекс	15	14	13	12	11	10	9	8
Имя поля	Зарезерви-ровано	RXFRC[10:4]						
Значение после сброса	-	0	0	0	0	0	0	0

Битовый индекс	7	6	5	4	3	2	1	0
Имя поля	RXFRC[3:0]				Зарезерви-ровано	RXMOF	RXF	RXE
Значение после сброса	0	0	0	0	-	0	0	1

RXE RX-буфер пуст. В нем не хранится ни один кадр CAN.

RXF RX-буфер полон, все слова памяти RX-буфера заполнены.

RXMOF Середина кадра RX-буфера. Если RXMOF = 1, следующее чтение из регистра RX_DATA выдает любое кроме первого слова (FRAME_FORMAT_W) кадра CAN.

RXFRC Счетчик кадров RX-буфера. Кол-во кадров CAN, хранящихся в RX-буфере.

1.3.9.5.5.1.35 RX_SETTINGS

Тип:	чтение-запись	**Смещение:**
0x6A	**Размер:**	1 байт

Настройки RX-буфера FIFO.

Битовый индекс	7	6	5	4	3	2	1	0
Имя поля	Зарезервировано							RTSOP
Значение после сброса	-	-	-	-	-	-	-	0

RTSOP Параметр метки времени получения. Этот регистр может быть изменен только при SETTINGS[ENA]=0.

0b0 - RTS_END - Метка времени полученного кадра в RX FIFO берется в последнем бите поля EOF.

0b1 - RTS_BEG - Метка времени полученного кадра в RX FIFO берется в поле SOF.

1.3.9.5.5.1.36 RX_DATA

Тип: только чтение **Смещение:** 0x6C **Размер:** 4 байта

Битовый индекс	31	30	29	28	27	26	25	24
Имя поля	RX_DATA[31:24]							
Значение после сброса	0	0	0	0	0	0	0	0

Битовый индекс	23	22	21	20	19	18	17	16
Имя поля	RX_DATA[23:16]							
Значение после сброса	0	0	0	0	0	0	0	0

Битовый индекс	15	14	13	12	11	10	9	8
Имя поля	RX_DATA[15:8]							
Значение после сброса	0	0	0	0	0	0	0	0

Битовый индекс	7	6	5	4	3	2	1	0
Имя поля	RX_DATA[7:0]							
Значение после сброса	0	0	0	0	0	0	0	0

RX_DATA Данные RX-буера в расположении указателя чтения в FIFO. При чтении этого регистра указатель чтения автоматически увеличивается, если MODES[RXBAM]=1 и RX-буфер не пуст. Если MODE[RXBAM]=1, этот регистр должен быть прочитан посредством

32-битного доступа. При чтении из этого регистра STATUS[RXPE] устанавливается, если в считываемом слове RX-буфера обнаружена ошибка четности.

1.3.9.5.5.1.37 TX_STATUS

Тип: только чтение **Смещение:** 0x70 **Размер:** 4 байта

Битовый индекс	31	30	29	28	27	26	25	24
Имя поля	TX8S				TX7S			
Значение после сброса	1	0	0	0	1	0	0	0

Битовый индекс	23	22	21	20	19	18	17	16
Имя поля	TX6S				TX5S			
Значение после сброса	1	0	0	0	1	0	0	0

Битовый индекс	15	14	13	12	11	10	9	8
Имя поля	TX4S				TX3S			
Значение после сброса	1	0	0	0	1	0	0	0

Битовый индекс	7	6	5	4	3	2	1	0
Имя поля	TX2S				TX1S			
Значение после сброса	1	0	0	0	1	0	0	0

TX1S Статус TXT-буфера 1. 0b0000 - TXT_NOT_EXIST - контроллер не содержит TXT-буфер (касается только TXT-буферов 3-8, когда CAN контроллере синтезировался с менее чем 8 TXT-буферами). 0b0001 - TXT_RDY - TXT-буфер в состоянии "Ready", он ждет, когда CAN контроллере начнет передачу.
0b0010 - TXT_TRAN - TXT-буфер находится в состоянии "TX in progress". CAN контроллере передает кадр.
0b0011 - TXT_ABTP - TXT-буфер в состоянии "Abort in progress". 0b0100 - TXT_TOK - TXT-буфер в состоянии "TX OK".
0b0110 - TXT_ERR - TXT-буфер в состоянии "Failed". 0b0111 - TXT_ABT - TXT-буфер в состоянии "Aborted". 0b1000 - TXT_ETY - TXT-буфер в состоянии "Empty".
0b1001 - TXT_PER - TXT-буфер в состоянии "Parity Error". CAN контроллере выявил ошибку четности в этом буфере.

TX2S Статус TXT-буфера 2. Значение поля бита аналогично TX1S.

TX3S Статус TXT-буфера 3. Значение поля бита аналогично TX1S.

TX4S Статус TXT-буфера 4. Значение поля бита аналогично TX1S.

TX5S Статус TXT-буера 5. Значение поля бита аналогично TX1S.

TX6S Статус TXT-буфера 6. Значение поля бита аналогично TX1S.

TX7S Статус TXT-буфера 7. Значение поля бита аналогично TX1S.

TX8S Статус TXT-буфера 8. Значение поля бита аналогично TX1S.

1.3.9.5.5.1.38 TX_COMMAND

Тип:	только запись	**Смещение:**
0x74	**Размер:**	2 байта

Регистр команд для TXT-буферов. Команда активируется путем записи логика 1 в бит TXC(E|R|A). TXT-буфер, который получает эту команду, выбирается путем установления бита TXB[1-8] на логику 1. Команду и индекс можно установить посредством единичного доступа. Либо индекс можно настроить заранее. Биты TXC(E|R|A) стираются автоматически после выполнения команды. Значения команд см. в описании TXT-буфера. Если TXCE и TXCR применены одновременно, применяется только команда TXCE. Если несколько команд применяются

одновременно, к буферу будут применены только команды, несущие результат для текущего состояния TXT-буфера.

Битовый индекс	15	14	13	12	11	10	9	8
Имя поля	TXB8	TXB7	TXB6	TXB5	TXB4	TXB3	TXB2	TXB1
Значение после сброса	0	0	0	0	0	0	0	0

Битовый индекс	7	6	5	4	3	2	1	0
Имя поля	Зарезервировано					TXCA	TXCR	TXCE
Значение после сброса	-	-	-	-	-	0	0	0

TXCE Выдает команду "set empty".

TXCR Выдает команду "set ready".

TXCA Выдает команду "set abort".

TXB1 Команда выдана TXT-буферу 1.

TXB2 Команда выдана TXT-буферу 2.

TXB3 Команда выдана TXT-буферу 3. Если количество TXT-буферов меньше 3, это поле резервируется и не имеет функций.

TXB4 Команда выдана TXT-буферу 4. Если количество TXT-буферов меньше 4, это поле резервируется и не имеет функций.

TXB5 Команда выдана TXT-буферу 5. Если количество TXT-буферов меньше 5, это поле резервируется и не имеет функций.

TXB6 Команда выдана TXT-буферу 6. Если количество TXT-буферов меньше 6, это поле резервируется и не имеет функций.

TXB7 Команда выдана TXT-буферу 7. Если количество TXT-буферов меньше 7, это поле резервируется и не имеет функций.

TXB8 Команда выдана TXT-буферу 8. Если количество TXT-буферов меньше 8, это поле резервируется и не имеет функций.

1.3.9.5.5.1.39 TXTB_INFO

Тип:	только чтение	**Смещение:**
0x76	**Размер:**	2 байта

Регистр с информацией о поддерживаемых функциях TXT-буферов.

Битовый индекс	15	14	13	12	11	10	9	8
Имя поля	Зарезервировано							
Значение после сброса	-	-	-	-	-	-	-	-

Битовый индекс	7	6	5	4	3	2	1	0
Имя поля	Зарезервировано				TXT_BUFFER_COUNT			
Значение после сброса	-	-	-	-	X	X	X	X

TXT_BUFFER_COUNT Количество TXT-буферов в CAN контроллере. Самый меньший буфер — это всегда 1. Наивысший буфер представлен индексом, равным количеству существующих буферов.

1.3.9.5.5.1.40 TX_PRIORITY

Тип:	чтение-запись	**Смещение:**
0x78	**Размер:**	4 байта

Приоритет TXT-буферов. Для передачи выбирается TXT-буфер в состоянии "Ready" с самым высоким приоритетом.

Битовый индекс	31	30	29	28	27	26	25	24
Имя поля	Зарезерви-ровано	TXT8P			Зарезерви-ровано	TXT7P		
Значение после сброса	-	0	0	0	-	0	0	0

Битовый индекс	23	22	21	20	19	18	17	16
Имя поля	Зарезерви-ровано	TXT6P			Зарезерви-ровано	TXT5P		
Значение после сброса	-	0	0	0	-	0	0	0

Битовый индекс	15	14	13	12	11	10	9	8
Имя поля	Зарезерви-ровано	TXT4P			Зарезерви-ровано	TXT3P		
Значение после сброса	-	0	0	0	-	0	0	0

Битовый индекс	7	6	5	4	3	2	1	0
Имя поля	Зарезерви-ровано	TXT2P			Зарезерви-ровано	TXT1P		
Значение после сброса	-	0	0	0	-	0	0	1

TXT1P Приоритет TXT-буфера 1.

TXT2P Приоритет TXT-буфера 2.

TXT3P Приоритет TXT-буфера 3. Если количество TXT-буферов меньше 3, это поле резервируется и не имеет функций.

TXT4P Приоритет TXT-буфера 4. Если количество TXT-буферов меньше 4, это поле резервируется и не имеет функций.

TXT5P Приоритет TXT-буфера 5. Если количество TXT-буферов меньше 5, это поле резервируется и не имеет функций.

TXT6P Приоритет TXT-буфера 6. Если количество TXT-буферов меньше 6, это поле резервируется и не имеет функций.

TXT7P Приоритет TXT-буфера 7. Если количество TXT-буферов меньше 7, это поле резервируется и не имеет функций.

TXT8P Приоритет TXT-буфера 8. Если количество TXT-буферов меньше 8, это поле резервируется и не имеет функций.

1.3.9.5.5.1.41 ERR_CAPT

Тип:	только чтение	**Смещение:**
0x7C	**Размер:**	1 байт

Регистр захвата кода ошибки. Определяет расположение в кадре CAN, где была обнаружена последняя ошибка.

Битовый индекс	7	6	5	4	3	2	1	0
Имя поля	ERR_TYPE			ERR_POS				
Значение после сброса	0	0	0	1	1	1	1	1

ERR_POS Расположение последней ошибки.

0b00000 - ERC_POS_SOF - Ошибка в начале кадра

0b00001 - ERC_POS_ARB - Ошибка в поле доступа к общей шине

0b00010 - ERC_POS_CTRL - Ошибка в поле управления

0b00011 - ERC_POS_DATA - Ошибка в поле данных

0b00100 - ERC_POS_CRC - Ошибка в поле CRC

0b00101 - ERC_POS_ACK - Ошибка в разделителе CRC, поле ACK или разделителе ACK

0b00110 - ERC_POS_EOF - Ошибка в поле конца кадра

0b00111 - ERC_POS_ERR - Ошибка во время кадра ошибки

0b01000 - ERC_POS_OVRL - Ошибка в кадре перегрузки

0b11111 - ERC_POS_OTHER - Другое расположение ошибки

ERR_TYPE Тип последней ошибки.

0b000 - ERC_BIT_ERR - Ошибка бита

0b001 - ERC_CRC_ERR - Ошибка CRC

0b010 - ERC_FRM_ERR - Ошибка формы

0b011 - ERC_ACK_ERR - Ошибка подтверждения 0b100 - ERC_STUF_ERR - Ошибка наполнения

0b101 - ERC_PRT_ERR - Ошибка четности в TXT-буфера слов DATA_1_4_W … DATA_61_64_W.

1.3.9.5.5.1.42 RETR_CTR

Тип:	только чтение	**Смещение:**
0x7D	**Размер:**	1 байт

Текущее значение счетчика повторной передачи.

Битовый индекс	7	6	5	4	3	2	1	0
Имя поля	Зарезервировано				RETR_CTR_VAL			
Значение после сброса	-	-	-	-	0	0	0	0

RETR_CTR_VAL Текущее значение счетчика повторной передачи.

1.3.9.5.5.1.43 ALC

Тип:	только чтение	**Смещение:**
0x7E	**Размер:**	1 байт

Регистр захвата потери доступа к общей шине. Определяет расположение последней потери доступа к общей шине в кадре CAN.

Битовый индекс	7	6	5	4	3	2	1	0
Имя поля	ALC_ID_FIELD			ALC_BIT				
Значение после сброса	0	0	0	0	0	0	0	0

ALC_BIT Расположение бита захвата потери доступа к общей шине. Если ALC_ID_FIELD = ALC_BASE_ID, то индекс бита БАЗОВОГО идентификатора, в котором был потерян доступ к общей шине, предоставляется следующим образом: 11 - ALC_VAL. Если ALC_ID_FIELD = ALC_EXTENSION, то индекс бита РАСШИРЕННОГО идентификатора, где был потерян доступ к общей шине, предоставляется следующим образом: 18 - ALC_VAL. Для других значений ALC_ID_FIELD это значение не определено.

ALC_ID_FIELD Часть идентификатора CAN, в котором был потерян доступ к общей шине.

0b000 - ALC_RSVD - Блок не потерял доступ к общей шине с последнего сброса.

0b001 - ALC_BASE_ID - Доступ к общей шине был потерян во время этого базового идентификатора.

0b010 - ALC_SRR_RTR - Доступ к общей шине был потерян во время первого бита после базового идентификатора (SRR расширенного кадра, бит RTR базового кадра CAN 2.0)

0b011 - ALC_IDE - Доступ к общей шине был потерян во время кадра IDE.

0b100 - ALC_EXTENSION - Доступ к общей шине был потерян во время расширения идентификатора. 0b101 - ALC_RTR - Доступ к общей шине был потерян во время бита RTR после расширения идентификатора!

1.3.9.5.5.1.44 TS_INFO

Тип: только чтение **Смещение:** 0x7F **Размер:** 1 байт

Информация об интеграции метки времени

Битовый индекс	7	6	5	4	3	2	1	0
Имя поля	Зарезервировано		TS_BITS					
Значение после сброса	-	-	X	X	X	X	X	X

TS_BITS Количество активных битов временного сетчика CAN контроллере за вычетом 1 (0x3F = 64-битная временной счетчик).

1.3.9.5.5.1.45 TRV_DELAY

Тип:	только чтение	**Смещение:**
0x80	**Размер:**	2 байта

Регистр задержки передатчика. При передаче кадра CAN контроллере измеряется задержка передатчика. После того как измерение будет выполнено (после бита FDF), его можно будет прочитать в этом регистре. Значение в этом регистре действительно с первой передачи кадра CAN контроллере. После каждого следующего измерения значение обновляется.

Битовый индекс	15	14	13	12	11	10	9	8
Имя поля	Зарезервировано							
Значение после сброса	-	-	-	-	-	-	-	-

Битовый индекс	7	6	5	4	3	2	1	0
Имя поля	Зарезерви-ровано	TRV_DELAY_VALUE						
Значение после сброса	-	0	0	0	0	0	0	0

TRV_DELAY_VALUE Измеренная задержка передатчика в нескольких из минимальных квант времени.

1.3.9.5.5.1.46 SSP_CFG

Тип:	чтение-запись	**Смещение:**
0x82	**Размер:**	2 байта

Примечание: Регистр может быть записан, только когда SETTINGS[ENA] = 0, в противном случае запись не принесет результат.

Регистр конфигурации вторичной контрольной точки. Используется передатчиком в скорости передачи данных для расчета вторичной контрольной точки.

Битовый индекс	15	14	13	12	11	10	9	8
Имя поля	Зарезервировано							SSP_SRC
Значение после сброса	-	-	-	-	-	-	0	0

Битовый индекс	7	6	5	4	3	2	1	0
Имя поля	SSP_OFFSET							
Значение после сброса	0	0	0	0	1	0	1	0

SSP_OFFSET Смещение вторичной контрольной точки. Значение дается как несколько минимальных квант времени.

SSP_SRC Источник вторичной контрольной точки.

0b00 - SSP_SRC_MEAS_N_OFFSET - Расположение SSP = TRV_DELAY (измеренная задержка передатчика) + SSP_OFFSET.

0b01 - SSP_SRC_NO_SSP - SSP не используется. Во время скорости передачи данных передатчик использует регулярную контрольную точку.

0b10 - SSP_SRC_OFFSET - Расположение SSP = SSP_OFFSET. Измеренное значение задержки передатчика игнорируется.

1.3.9.5.5.1.47 RX_FR_CTR

Тип:	только чтение	**Смещение:**
0x84	**Размер:**	4 байта

Примечание: Регистр существует только при значении sup_traffic_ctrs = true. В противном случае этот адрес резервируется.

Битовый индекс	31	30	29	28	27	26	25	24
Имя поля	RX_FR_CTR_VAL[31:24]							
Значение после сброса	0	0	0	0	0	0	0	0

Битовый индекс	23	22	21	20	19	18	17	16
Имя поля	RX_FR_CTR_VAL[23:16]							
Значение после сброса	0	0	0	0	0	0	0	0

Битовый индекс	15	14	13	12	11	10	9	8
Имя поля	RX_FR_CTR_VAL[15:8]							
Значение после сброса	0	0	0	0	0	0	0	0

Битовый индекс	7	6	5	4	3	2	1	0
Имя поля	RX_FR_CTR_VAL[7:0]							

Значение после сброса	0	0	0	0	0	0	0	0

RX_FR_CTR_VAL Количество кадров, полученных CAN контроллере.

1.3.9.5.5.1.48 TX_FR_CTR

Тип: только чтение **Смещение:** 0x88 **Размер:** 4 байта

Примечание: Регистр существует только при значении sup_traffic_ctrs = true. В противном случае этот адрес резервируется.

Битовый индекс	31	30	29	28	27	26	25	24
Имя поля	TX_FR_CTR_VAL[31:24]							
Значение после сброса	0	0	0	0	0	0	0	0

Битовый индекс	23	22	21	20	19	18	17	16
Имя поля	TX_FR_CTR_VAL[23:16]							
Значение после сброса	0	0	0	0	0	0	0	0

Битовый индекс	15	14	13	12	11	10	9	8
Имя поля	TX_FR_CTR_VAL[15:8]							
Значение после сброса	0	0	0	0	0	0	0	0

Битовый индекс	7	6	5	4	3	2	1	0
Имя поля	TX_FR_CTR_VAL[7:0]							

Значение после сброса	0	0	0	0	0	0	0	0

TX_FR_CTR_VAL Количество кадров, переданных CAN контроллере.

1.3.9.5.5.1.49 DEBUG_REGISTER

Тип:	только чтение	**Смещение:**
0x8C	**Размер:**	4 байта

Регистр для состояния чтения контроллера. Этот регистр предназначен только в целях отладки!

Битовый индекс	31	30	29	28	27	26	25	24
Имя поля	Зарезервировано							
Значение после сброса	-	-	-	-	-	-	-	-

Битовый индекс	23	22	21	20	19	18	17	16
Имя поля	Зарезервировано					PC_SOF	PC_OVR	PC_SUSP
Значение после сброса	-	-	-	-	-	0	0	0

Битовый индекс	15	14	13	12	11	10	9	8
Имя поля	PC_INT	PC_EOF	PC_ACKD	PC_ACK	PC_CRCD	PC_CRC	PC_STC	PC_DAT
Значение после сброса	0	0	0	0	0	0	0	0

Битовый индекс	7	6	5	4	3	2	1	0
Имя поля	PC_CON	PC_ARB	DESTUFF_COUNT			STUFF_COUNT		

Значение после сброса	0	0	0	0	0	0	0	0

STUFF_COUNT Фактический счетчик наполнения по модулю 8 согласно определению в протоколе ISO FD. Счетчик наполнения стирается в начале кадра CAN и увеличивается на один с каждым битом наполнения до поля счетчика наполнения в кадре ISO FD. Затем он остается фиксированным до начала следующего кадра. В наполнении не по стандарту ISO FD или в нормальном наполнении CAN биты считаются до конца кадра. Обратите внимание, что это поле НЕ получает серую кодировку по стандарту ISO FD. Счетчик наполнения рассчитывается до тех пор, пока контроллер получает и передает данные на шине. Во время приема это значение остается фиксированным!

DESTUFF_COUNT Фактический счетчик исключения нулевых бит по модулю 8 согласно определению в протоколе ISO FD. Счетчик исключения нулевых бит стирается в начале кадра и увеличивается на один с каждым исключением нулевого бита до поля счетчика наполнения в кадре ISO FD. Затем он остается фиксированным до начала следующего кадра. В исключении нулевых битов не по стандарту ISO FD или в исключении нулевых битов CAN биты считаются до конца кадра. Обратите внимание, что это поле НЕ получает серую кодировку по стандарту ISO FD. Счетчик исключения нулевых битов считается в обоих случаях. Приемопередатчик, а также приемник.

PC_ARB Машина состояния контроля протокола в поле доступа к общей шине.

PC_CON Машина состояния контроля протокола в поле управления.

PC_DAT Машина состояния контроля протокола в поле данных.

PC_STC Машина состояния контроля протокола в поле счетчика наполнения.

PC_CRC Машина состояния контроля протокола в поле CRC.

PC_CRCD Машина состояния контроля протокола в поле разделителя CRC.

PC_ACKCRC Машина состояния контроля протокола в поле ACK.

PC_ACKD Машина состояния контроля протокола в поле разделителя ACK.

PC_EOF Машина состояния контроля протокола в поле конца файла.

PC_INTARB Машина состояния контроля протокола в поле паузы.

PC_SUSP Машина состояния контроля протокола в поле приостановки передачи.

PC_OVR Машина состояния контроля протокола в поле перегрузки.

PC_SOF Машина состояния контроля протокола в поле начала кадра.

1.3.9.5.5.1.50 3.1.50 YOLO_REG

Тип: только чтение **Смещение:**
0x90 **Размер:** 4 байта

Регистр для веселья :)

Битовый индекс	31	30	29	28	27	26	25	24
Имя поля	YOLO_VAL[31:24]							
Значение после сброса	1	1	0	1	1	1	1	0

Битовый индекс	23	22	21	20	19	18	17	16
Имя поля	YOLO_VAL[23:16]							
Значение после сброса	1	0	1	0	1	1	0	1

Битовый индекс	15	14	13	12	11	10	9	8
Имя поля	YOLO_VAL[15:8]							
Значение после сброса	1	0	1	1	1	1	1	0

Битовый индекс	7	6	5	4	3	2	1	0
Имя поля	YOLO_VAL[7:0]							
Значение после сброса	1	1	1	0	1	1	1	1

YOLO_VAL Что еще может быть в этом регистре?

1.3.9.5.5.1.51 TIMESTAMP_LOW

Тип:	только чтение	**Смещение:**
0x94	**Размер:**	4 байта

Регистр с текущим значением временного счетчика CAN контроллере. В регистрах TIMESTAMP_LOW/HIGH не реализуется теневое копирование, и пользователю необходимо позаботиться о правильном чтении обоих регистров, поскольку в период между чтением TIMESTAMP_LOW и TIMESTAMP_HIGH может быть превышено TIMESTAMP_LOW.

Битовый индекс	31	30	29	28	27	26	25	24
Имя поля	TIMESTAMP_LOW[31:24]							
Значение после сброса	X	X	X	X	X	X	X	X

Битовый индекс	23	22	21	20	19	18	17	16
Имя поля	TIMESTAMP_LOW[23:16]							
Значение после сброса	X	X	X	X	X	X	X	X

Битовый индекс	15	14	13	12	11	10	9	8
Имя поля	TIMESTAMP_LOW[15:8]							
Значение после сброса	X	X	X	X	X	X	X	X

Битовый индекс	7	6	5	4	3	2	1	0
Имя поля	TIMESTAMP_LOW[7:0]							
Значение после сброса	X	X	X	X	X	X	X	X

TIMESTAMP_LOW Биты 31:0 временного счетчика.

1.3.9.5.5.1.52 TIMESTAMP_HIGH

Тип:	только чтение	**Смещение:**
0x98	**Размер:**	4 байта

Регистр с текущим значением временного счетчика CAN контроллере. В регистрах TIMESTAMP_LOW/HIGH не реализуется теневое копирование, и пользователю необходимо позаботиться о правильном чтении обоих регистров, поскольку в период между чтением TIMESTAMP_LOW и TIMESTAMP_HIGH может быть превышено TIMESTAMP_LOW.

Битовый индекс	31	30	29	28	27	26	25	24
Имя поля	TIMESTAMP_HIGH[31:24]							
Значение после сброса	X	X	X	X	X	X	X	X

Битовый индекс	23	22	21	20	19	18	17	16
Имя поля	TIMESTAMP_HIGH[23:16]							
Значение после сброса	X	X	X	X	X	X	X	X

Битовый индекс	15	14	13	12	11	10	9	8
Имя поля	TIMESTAMP_HIGH[15:8]							
Значение после сброса	X	X	X	X	X	X	X	X

Битовый индекс	7	6	5	4	3	2	1	0
Имя поля	TIMESTAMP_HIGH[7:0]							
Значение после сброса	X	X	X	X	X	X	X	X

TIMESTAMP_HIGH Биты 63:32 временного счетчика.

1.3.9.5.5.2 TXT-буфер 1

Доступ к этой области памяти сопоставлен с TXT-буфером 1. Кадр CAN контроллере для передачи может быть добавлен в этот буфер. Макет кадра соответствует макету, описанному в главе "Формат кадра CAN контроллере". Первый адрес в этой области (TXTB1_DATA_1) соответствует FRAME_FORMAT_W, второй адрес (TXTB1_DATA_2) соответствует IDENTIFIER_W и т. д. Последний адрес (TXTB1_DATA_20) соответствует DATA_61_64_W. Адреса между ними выстраиваются линейно. Эта область памяти предназначена только для записи, а доступ для чтения выдает все нули. Эта область предназначена только для записи.

Биты [31:24]	Биты [23:16]	Биты [15:8]	Биты [7:0]	Смещение адреса
TXTB1_DATA_1				0x100
TXTB1_DATA_2				0x104
TXTB1_DATA_3				0x108
TXTB1_DATA_4				0x10C
TXTB1_DATA_5				0x110
TXTB1_DATA_6				0x114
TXTB1_DATA_7				0x118
TXTB1_DATA_8				0x11C
TXTB1_DATA_9				0x120
TXTB1_DATA_10				0x124
TXTB1_DATA_11				0x128
TXTB1_DATA_12				0x12C
TXTB1_DATA_13				0x130
TXTB1_DATA_14				0x134
TXTB1_DATA_15				0x138
TXTB1_DATA_16				0x13C
TXTB1_DATA_17				0x140
TXTB1_DATA_18				0x144
TXTB1_DATA_19				0x148

Биты [31:24]	Биты [23:16]	Биты [15:8]	Биты [7:0]	Смещение адреса
TXTB1_DATA_20				0x14C
TXTB1_DATA_21				0x150
Зарезервировано				…

1.3.9.5.5.3 TXT-буфер 2

Доступ к этой области памяти сопоставлен с TXT-буфером 2. Кадр CAN контроллере для передачи может быть добавлен в этот буфер. Макет кадра соответствует макету, описанному в главе "Формат кадра CAN контроллере". Первый адрес в этой области (TXTB2_DATA_1) соответствует FRAME_FORMAT_W, второй адрес (TXTB2_DATA_2) соответствует IDENTIFIER_W и т. д. Последний адрес (TXTB2_DATA_20) соответствует DATA_61_64_W. Адреса между ними выстраиваются линейно. Эта область памяти предназначена только для записи, а доступ для чтения выдает все нули. Эта область предназначена только для записи.

Биты [31:24]	Биты [23:16]	Биты [15:8]	Биты [7:0]	Смещение адреса
TXTB2_DATA_1				0x200
TXTB2_DATA_2				0x204
TXTB2_DATA_3				0x208
TXTB2_DATA_4				0x20C
TXTB2_DATA_5				0x210
TXTB2_DATA_6				0x214
TXTB2_DATA_7				0x218
TXTB2_DATA_8				0x21C
TXTB2_DATA_9				0x220
TXTB2_DATA_10				0x224
TXTB2_DATA_11				0x228
TXTB2_DATA_12				0x22C
TXTB2_DATA_13				0x230
TXTB2_DATA_14				0x234
TXTB2_DATA_15				0x238
TXTB2_DATA_16				0x23C
TXTB2_DATA_17				0x240
TXTB2_DATA_18				0x244
TXTB2_DATA_19				0x248
TXTB2_DATA_20				0x24C
TXTB2_DATA_21				0x250
Зарезервировано				…

1.3.9.5.5.4 TXT-буфер 3

Доступ к этой области памяти сопоставлен с TXT-буфером 1. Кадр CAN контроллере для передачи может быть добавлен в этот буфер. Макет кадра соответствует макету, описанному в главе "Формат кадра CAN контроллере". Первый адрес в этой области (TXTB3_DATA_1) соответствует FRAME_FORMAT_W, второй адрес (TXTB3_DATA_2) соответствует IDENTIFIER_W и т. д. Последний адрес (TXTB3_DATA_20) соответствует DATA_61_64_W. Адреса между ними выстраиваются линейно. Эта область памяти предназначена только для записи, а доступ для чтения выдает все нули. Эта область предназначена только для записи.

Биты [31:24]	Биты [23:16]	Биты [15:8]	Биты [7:0]	Смещение адреса
TXTB3_DATA_1				0x300
TXTB3_DATA_2				0x304
TXTB3_DATA_3				0x308
TXTB3_DATA_4				0x30C
TXTB3_DATA_5				0x310
TXTB3_DATA_6				0x314
TXTB3_DATA_7				0x318
TXTB3_DATA_8				0x31C
TXTB3_DATA_9				0x320
TXTB3_DATA_10				0x324
TXTB3_DATA_11				0x328
TXTB3_DATA_12				0x32C
TXTB3_DATA_13				0x330
TXTB3_DATA_14				0x334
TXTB3_DATA_15				0x338
TXTB3_DATA_16				0x33C
TXTB3_DATA_17				0x340
TXTB3_DATA_18				0x344
TXTB3_DATA_19				0x348
TXTB3_DATA_20				0x34C
TXTB3_DATA_21				0x350
Зарезервировано				…

1.3.9.5.5.5 TXT-буфер 4

Доступ к этой области памяти сопоставлен с TXT-буфером 4. Кадр CAN контроллере для передачи может быть добавлен в этот буфер. Макет кадра соответствует макету,

описанному в главе "Формат кадра CAN контроллере". Первый адрес в этой области (TXTB4_DATA_1) соответствует FRAME_FORMAT_W, второй адрес (TXTB4_DATA_2) соответствует IDENTIFIER_W и т. д. Последний адрес (TXTB4_DATA_20) соответствует DATA_61_64_W. Адреса между ними выстраиваются линейно. Эта область памяти предназначена только для записи, а доступ для чтения выдает все нули. Эта область предназначена только для записи.

Биты [31:24]	Биты [23:16]	Биты [15:8]	Биты [7:0]	Смещение адреса
TXTB4_DATA_1				0x400
TXTB4_DATA_2				0x404
TXTB4_DATA_3				0x408
TXTB4_DATA_4				0x40C
TXTB4_DATA_5				0x410
TXTB4_DATA_6				0x414
TXTB4_DATA_7				0x418
TXTB4_DATA_8				0x41C
TXTB4_DATA_9				0x420
TXTB4_DATA_10				0x424
TXTB4_DATA_11				0x428
TXTB4_DATA_12				0x42C
TXTB4_DATA_13				0x430
TXTB4_DATA_14				0x434
TXTB4_DATA_15				0x438
TXTB4_DATA_16				0x43C
TXTB4_DATA_17				0x440
TXTB4_DATA_18				0x444
TXTB4_DATA_19				0x448
TXTB4_DATA_20				0x44C
TXTB4_DATA_21				0x450
Зарезервировано				…

1.3.9.5.5.6 TXT-буфер 5

Доступ к этой области памяти сопоставлен с TXT-буфером 4. Кадр CAN контроллере для передачи может быть добавлен в этот буфер. Макет кадра соответствует макету, описанному в главе "Формат кадра CAN контроллере". Первый адрес в этой области (TXTB5_DATA_1) соответствует FRAME_FORMAT_W, второй адрес (TXTB5_DATA_2) соответствует IDENTIFIER_W и т. д. Последний адрес (TXTB5_DATA_20) соответствует

DATA_61_64_W. Адреса между ними выстраиваются линейно. Эта область памяти предназначена только для записи, а доступ для чтения выдает все нули. Эта область предназначена только для записи.

Биты [31:24]	Биты [23:16]	Биты [15:8]	Биты [7:0]	Смещение адреса
TXTB5_DATA_1				0x500
TXTB5_DATA_2				0x504
TXTB5_DATA_3				0x508
TXTB5_DATA_4				0x50C
TXTB5_DATA_5				0x510
TXTB5_DATA_6				0x514
TXTB5_DATA_7				0x518
TXTB5_DATA_8				0x51C
TXTB5_DATA_9				0x520
TXTB5_DATA_10				0x524
TXTB5_DATA_11				0x528
TXTB5_DATA_12				0x52C
TXTB5_DATA_13				0x530
TXTB5_DATA_14				0x534
TXTB5_DATA_15				0x538
TXTB5_DATA_16				0x53C
TXTB5_DATA_17				0x540
TXTB5_DATA_18				0x544
TXTB5_DATA_19				0x548
TXTB5_DATA_20				0x54C
TXTB5_DATA_21				0x550
Зарезервировано				…

1.3.9.5.5.7 TXT-буфер 6

Доступ к этой области памяти сопоставлен с TXT-буфером 4. Кадр CAN контроллере для передачи может быть добавлен в этот буфер. Макет кадра соответствует макету, описанному в главе "Формат кадра CAN контроллере". Первый адрес в этой области (TXTB6_DATA_1) соответствует FRAME_FORMAT_W, второй адрес (TXTB6_DATA_2) соответствует IDENTIFIER_W и т. д. Последний адрес (TXTB6_DATA_20) соответствует DATA_61_64_W. Адреса между ними выстраиваются линейно. Эта область памяти предназначена только для записи, а доступ для чтения выдает все нули. Эта область предназначена только для записи.

Биты [31:24]	Биты [23:16]	Биты [15:8]	Биты [7:0]	Смещение адреса
TXTB6_DATA_1				0x600
TXTB6_DATA_2				0x604
TXTB6_DATA_3				0x608
TXTB6_DATA_4				0x60C
TXTB6_DATA_5				0x610
TXTB6_DATA_6				0x614
TXTB6_DATA_7				0x618
TXTB6_DATA_8				0x61C
TXTB6_DATA_9				0x620
TXTB6_DATA_10				0x624
TXTB6_DATA_11				0x628
TXTB6_DATA_12				0x62C
TXTB6_DATA_13				0x630
TXTB6_DATA_14				0x634
TXTB6_DATA_15				0x638
TXTB6_DATA_16				0x63C
TXTB6_DATA_17				0x640
TXTB6_DATA_18				0x644
TXTB6_DATA_19				0x648
TXTB6_DATA_20				0x64C
TXTB6_DATA_21				0x650
Зарезервировано				…

1.3.9.5.5.8 TXT-буфер 7

Доступ к этой области памяти сопоставлен с TXT-буфером 4. Кадр CAN контроллере для передачи может быть добавлен в этот буфер. Макет кадра соответствует макету, описанному в главе "Формат кадра CAN контроллере". Первый адрес в этой области (TXTB7_DATA_1) соответствует FRAME_FORMAT_W, второй адрес (TXTB7_DATA_2) соответствует IDENTIFIER_W и т. д. Последний адрес (TXTB7_DATA_20) соответствует DATA_61_64_W. Адреса между ними выстраиваются линейно. Эта область памяти предназначена только для записи, а доступ для чтения выдает все нули. Эта область предназначена только для записи.

Биты [31:24]	Биты [23:16]	Биты [15:8]	Биты [7:0]	Смещение адреса
TXTB7_DATA_1				0x700
TXTB7_DATA_2				0x704
TXTB7_DATA_3				0x708
TXTB7_DATA_4				0x70C
TXTB7_DATA_5				0x710
TXTB7_DATA_6				0x714
TXTB7_DATA_7				0x718
TXTB7_DATA_8				0x71C
TXTB7_DATA_9				0x720
TXTB7_DATA_10				0x724
TXTB7_DATA_11				0x728
TXTB7_DATA_12				0x72C
TXTB7_DATA_13				0x730
TXTB7_DATA_14				0x734
TXTB7_DATA_15				0x738
TXTB7_DATA_16				0x73C
TXTB7_DATA_17				0x740
TXTB7_DATA_18				0x744
TXTB7_DATA_19				0x748
TXTB7_DATA_20				0x74C
TXTB7_DATA_21				0x750
Зарезервировано				…

1.3.9.5.5.9 TXT-буфер 8

Доступ к этой области памяти сопоставлен с TXT-буфером 4. Кадр CAN контроллере для передачи может быть добавлен в этот буфер. Макет кадра соответствует макету, описанному в главе "Формат кадра CAN контроллере". Первый адрес в этой области (TXTB8_DATA_1) соответствует FRAME_FORMAT_W, второй адрес (TXTB8_DATA_2) соответствует IDENTIFIER_W и т. д. Последний адрес (TXTB8_DATA_20) соответствует DATA_61_64_W. Адреса между ними выстраиваются линейно. Эта область памяти предназначена только для записи, а доступ для чтения выдает все нули. Эта область предназначена только для записи.

Биты [31:24]	Биты [23:16]	Биты [15:8]	Биты [7:0]	Смещение адреса
TXTB8_DATA_1				0x800
TXTB8_DATA_2				0x804
TXTB8_DATA_3				0x808

Биты [31:24]	Биты [23:16]	Биты [15:8]	Биты [7:0]	Смещение адреса
TXTB8_DATA_4				0x80C
TXTB8_DATA_5				0x810
TXTB8_DATA_6				0x814
TXTB8_DATA_7				0x818
TXTB8_DATA_8				0x81C
TXTB8_DATA_9				0x820
TXTB8_DATA_10				0x824
TXTB8_DATA_11				0x828
TXTB8_DATA_12				0x82C
TXTB8_DATA_13				0x830
TXTB8_DATA_14				0x834
TXTB8_DATA_15				0x838
TXTB8_DATA_16				0x83C
TXTB8_DATA_17				0x840
TXTB8_DATA_18				0x844
TXTB8_DATA_19				0x848
TXTB8_DATA_20				0x84C
TXTB8_DATA_21				0x850
Зарезервировано				…

1.3.9.5.5.10 Регистры тестирования

Область памяти регистров тестирования. Содержит регистры с функциями производственного тестирования.

Биты [31:24]	Биты [23:16]	Биты [15:8]	Биты [7:0]	Смещение адреса
TST_CONTROL				0x900
TST_DEST				0x904
TST_WDATA				0x908
TST_RDATA				0x90C
Зарезервировано				…

1.3.9.5.5.10.1 TST_CONTROL

Тип: чтение-запись **Смещение:** 0x900 **Размер:** 4 байта

Примечание: Регистр может быть записан, только когда MODE[TSTM] = 1, в противном случае запись не принесет результат.

Регистр контроля тестирования. Содержит конфигурацию функций тестирования.

Битовый индекс	31	30	29	28	27	26	25	24
Имя поля	Зарезервировано							
Значение после сброса	-	-	-	-	-	-	-	-

Битовый индекс	23	22	21	20	19	18	17	16
Имя поля	Зарезервировано							
Значение после сброса	-	-	-	-	-	-	-	-

Битовый индекс	15	14	13	12	11	10	9	8
Имя поля	Зарезервировано							
Значение после сброса	-	-	-	-	-	-	-	-

Битовый индекс	7	6	5	4	3	2	1	0
Имя поля	Зарезервировано						TWRSTB	TMAENA
Значение после сброса	-	-	-	-	-	-	X	X

TMAENA Включить тестовый доступ к памяти CAN контроллере.

TWRSTB Запись 1 предоставляет доступ для записи к памяти/адресу, предоставленному регистром TST_DEST. Запись 0 не требуется, этот бит очищается автоматически.

1.3.9.5.5.10.2 TST_DEST

Тип: чтение-запись **Смещение:** 0x904 **Размер:** 4 байта

Примечание: Регистр может быть записан, только когда MODE[TSTM] = 1, в противном случае запись не принесет результат.

Битовый индекс	31	30	29	28	27	26	25	24
Имя поля	Зарезервировано							
Значение после сброса	-	-	-	-	-	-	-	-

Битовый индекс	23	22	21	20	19	18	17	16
Имя поля	Зарезервировано				TST_MTGT			
Значение после сброса	-	-	-	-	X	X	X	X

Битовый индекс	15	14	13	12	11	10	9	8
Имя поля	TST_ADDR[15:8]							
Значение после сброса	X	X	X	X	X	X	X	X
Битовый индекс	7	6	5	4	3	2	1	0
Имя поля	TST_ADDR[7:0]							
Значение после сброса	X	X	X	X	X	X	X	X

TST_ADDR Адрес для доступа к тестовой памяти внутри тестируемой памяти.

TST_MTGT Целевая память, к которой предоставляется доступ.

0b0000 - TMTGT_NONE - Целевая память для тестового доступа не выбрана.

0b0001 - TMTGT_RXBUF - Для тестового доступа выбрана память RX-буфера.

0b0010 - TMTGT_TXTBUF1 - Для тестового доступа выбрана память TXT-буфера 1.

0b0011 - TMTGT_TXTBUF2 - Для тестового доступа выбрана память TXT-буфера 2.

0b0100 - TMTGT_TXTBUF3 - Для тестового доступа выбрана память TXT-буфера 3.

0b0101 - TMTGT_TXTBUF4 - Для тестового доступа выбрана память TXT-буфера 4.

0b0110 - TMTGT_TXTBUF5 - Для тестового доступа выбрана память TXT-буфера 5.

0b0111 - TMTGT_TXTBUF6 - Для тестового доступа выбрана память TXT-буфера 6.

0b1000 - TMTGT_TXTBUF7 - Для тестового доступа выбрана память TXT-буфера 7.

0b1001 - TMTGT_TXTBUF8 - Для тестового доступа выбрана память TXT-буфера 8.

1.3.9.5.5.10.3 TST_WDATA

Тип:	чтение-запись	**Смещение:**
0x908	**Размер:**	4 байта

Примечание: Регистр может быть записан, только когда MODE[TSTM] = 1, в противном случае запись не принесет результат.

Битовый индекс	31	30	29	28	27	26	25	24
Имя поля	TST_WDATA[31:24]							
Значение после сброса	X	X	X	X	X	X	X	X

Битовый индекс	23	22	21	20	19	18	17	16
Имя поля	TST_WDATA[23:16]							
Значение после сброса	X	X	X	X	X	X	X	X

Битовый индекс	15	14	13	12	11	10	9	8
Имя поля	TST_WDATA[15:8]							

Значение после сброса	X	X	X	X	X	X	X	X

Битовый индекс	7	6	5	4	3	2	1	0
Имя поля	TST_WDATA[7:0]							
Значение после сброса	X	X	X	X	X	X	X	X

TST_WDATA Запись данных для тестового доступа.

1.3.9.5.5.10.4 TST_RDATA

Тип: только чтение **Смещение:** 0x90C **Размер:** 4 байта

Битовый индекс	31	30	29	28	27	26	25	24
Имя поля	TST_RDATA[31:24]							
Значение после сброса	X	X	X	X	X	X	X	X

Битовый индекс	23	22	21	20	19	18	17	16
Имя поля	TST_RDATA[23:16]							
Значение после сброса	X	X	X	X	X	X	X	X

Битовый индекс	15	14	13	12	11	10	9	8
Имя поля	TST_RDATA[15:8]							
Значение после сброса	X	X	X	X	X	X	X	X

Битовый индекс	7	6	5	4	3	2	1	0
Имя поля	TST_RDATA[7:0]							
Значение после сброса	X	X	X	X	X	X	X	X

TST_RDATA Чтение данных для тестового доступа.

1.3.9.5.6 4. Формат кадра CAN FD

Описание формата кадра CAN, хранящегося в TXT-буферах и RX-буфере.

1.3.9.5.6.1 Формат кадра CAN FD

Биты [31:24]	Биты [23:16]	Биты [15:8]	Биты [7:0]	Смещение адреса
FRAME_FORMAT_W				0x0
IDENTIFIER_W				0x4
TIMESTAMP_L_W				0x8
TIMESTAMP_U_W				0xC
DATA_1_4_W				0x10
DATA_5_8_W				0x14
DATA_9_12_W				0x18
DATA_13_16_W				0x1C
DATA_17_20_W				0x20
DATA_21_24_W				0x24
DATA_25_28_W				0x28
DATA_29_32_W				0x2C
DATA_33_36_W				0x30
DATA_37_40_W				0x34
DATA_41_44_W				0x38
DATA_45_48_W				0x3C
DATA_49_52_W				0x40
DATA_53_56_W				0x44
DATA_57_60_W				0x48
DATA_61_64_W				0x4C
FRAME_TEST_W				0x50

1.3.9.5.6.1.1 FRAME_FORMAT_W

Тип:

Смещение: 0x0

Размер: 4 байта

Слово формата кадра с метаданными кадра CAN.

Битовый индекс	31	30	29	28	27	26	25	24
Имя поля	Зарезервировано							
Значение после сброса	-	-	-	-	-	-	-	-

Битовый индекс	23	22	21	20	19	18	17	16
Имя поля	Зарезервировано							
Значение после сброса	-	-	-	-	-	-	-	-

Битовый индекс	15	14	13	12	11	10	9	8
Имя поля	RWCNT					ESI_RSV	BRS	Зарезерви-ровано
Значение после сброса	X	X	X	X	X	X	X	-

Битовый индекс	7	6	5	4	3	2	1	0
Имя поля	FDF	IDE	RTR	Зарезерви-ровано	DLC			
Значение после сброса	X	X	X	-	X	X	X	X

DLC Код длины данных.

RTR Логика 1 указывает на удаленный кадр. Имеет значение только для кадров CAN. CAN FD не включает кадры RTR. 0b0 - NO_RTR_FRAME - Кадр CAN не является кадром RTR. 0b1 - RTR_FRAME - Кадр CAN является кадром RTR.

IDE Тип расширенного идентификатора. Логика 1 указывает на кадр CAN с базовым идентификатором и расширением идентификатора. Логика 0 указывает на кадр CAN только с

базовым идентификатором.
0b0 - BASE - У кадра базовый идентификатор (11 битов)
0b1 - EXTENDED - У кадра расширенный идентификатор (11 + 18 битов)

FDF Гибкий формат скорости передачи. Различает между кадрами CAN 2.0 и CAN FD.
0b0 - NORMAL_CAN - Кадр является кадром CAN.
0b1 - FD_CAN - Кадр является кадром CAN FD.

BRS Смена скорости передачи данных. В случае с кадрами CAN FD указывает, меняется ли скорости передачи данных в зависимости от кадра CAN FD. Этот бит не имеет значения для кадров CAN.
0b0 - BR_NO_SHIFT - Скорость передачи не должна меняться, если кадр является кадром CAN FD.
0b1 - BR_SHIFT - Скорость передачи должна меняться, если кадр является кадром CAN FD.

ESI_RSV Бит индикатора состояния ошибки для полученных кадров CAN FD. Бит не имеет значения ни для кадров CAN, ни для переданных кадров CAN FD (в TXT-буфере).
0b0 - ESI_ERR_ACTIVE - Переданный или полученный кадр CAN FD действует при ошибке.
0b1 - ESI_ERR_PASIVE - Переданный или полученный кадр CAN FD бездействует при ошибке.

RWCNT Размер кадра CAN в RX-буфере без FRAME_FORMAT WORD. (Например, кадр RTR RWCNT=3, 64 байта Кадр FD RWCNT=19). В TXT-буфере это поле не имеет значения.

1.3.9.5.6.1.2 IDENTIFIER_W

Тип:
Смещение: 0x4
Размер: 4 байта

Идентификатор CAN.

Битовый индекс	**31**	**30**	**29**	**28**	**27**	**26**	**25**	**24**
Имя поля	Зарезервировано			IDENTIFIER_BASE[10:6]				
Значение после сброса	-	-	-	X	X	X	X	X

Битовый индекс	23	22	21	20	19	18	17	16
Имя поля	IDENTIFIER_BASE[5:0]						IDENTIFIER_EXT[17:16]	
Значение после сброса	X	X	X	X	X	X	X	X

Битовый индекс	15	14	13	12	11	10	9	8
Имя поля	IDENTIFIER_EXT[15:8]							
Значение после сброса	X	X	X	X	X	X	X	X

Битовый индекс	7	6	5	4	3	2	1	0
Имя поля	IDENTIFIER_EXT[7:0]							
Значение после сброса	X	X	X	X	X	X	X	X

IDENTIFIER_EXT Расширенный идентификатор кадра CAN. Имеет значение, только если FRAME_FORMAT_W имеет расширенный идентификатор.

IDENTIFIER_BASE Базовый идентификатор кадра CAN.

1.3.9.5.6.1.3 TIMESTAMP_L_W

Тип:

Смещение: 0x8

Размер: 4 байта

Битовый индекс	31	30	29	28	27	26	25	24
Имя поля	TIME_STAMP_L_W[31:24]							

Битовый индекс	31	30	29	28	27	26	25	24
Значение после сброса	X	X	X	X	X	X	X	X

Битовый индекс	23	22	21	20	19	18	17	16
Имя поля	TIME_STAMP_L_W[23:16]							
Значение после сброса	X	X	X	X	X	X	X	X

Битовый индекс	15	14	13	12	11	10	9	8
Имя поля	TIME_STAMP_L_W[15:8]							
Значение после сброса	X	X	X	X	X	X	X	X

Битовый индекс	7	6	5	4	3	2	1	0
Имя поля	TIME_STAMP_L_W[7:0]							
Значение после сброса	X	X	X	X	X	X	X	X

TIME_STAMP_L_W Младшие 32 бита метки времени, когда кадр должен быть передан или когда был получен.

1.3.9.5.6.1.4 TIMESTAMP_U_W

Тип:

Смещение: 0xC

Размер: 4 байта

Битовый индекс	31	30	29	28	27	26	25	24
Имя поля	TIMESTAMP_U_W[31:24]							
Значение после сброса	X	X	X	X	X	X	X	X

Битовый индекс	23	22	21	20	19	18	17	16
Имя поля	TIMESTAMP_U_W[23:16]							
Значение после сброса	X	X	X	X	X	X	X	X

Битовый индекс	15	14	13	12	11	10	9	8
Имя поля	TIMESTAMP_U_W[15:8]							
Значение после сброса	X	X	X	X	X	X	X	X

Битовый индекс	7	6	5	4	3	2	1	0
Имя поля	TIMESTAMP_U_W[7:0]							
Значение после сброса	X	X	X	X	X	X	X	X

TIMESTAMP_U_W Старшие 32 бита метки времени, когда кадр должен быть передан или когда был получен.

1.3.9.5.6.1.5 DATA_1_4_W

Тип:

Смещение: 0x10

Размер: 4 байта

Битовый индекс	31	30	29	28	27	26	25	24
Имя поля	DATA_4							

Битовый индекс	31	30	29	28	27	26	25	24
Значение после сброса	X	X	X	X	X	X	X	X

Битовый индекс	23	22	21	20	19	18	17	16
Имя поля	DATA_3							
Значение после сброса	X	X	X	X	X	X	X	X

Битовый индекс	15	14	13	12	11	10	9	8
Имя поля	DATA_2							
Значение после сброса	X	X	X	X	X	X	X	X

Битовый индекс	7	6	5	4	3	2	1	0
Имя поля	DATA_1							
Значение после сброса	X	X	X	X	X	X	X	X

DATA_1 Байт данных 1 кадра CAN.

DATA_2 Байт данных 2 кадра CAN.

DATA_3 Байт данных 3 кадра CAN.

DATA_4 Байт данных 4 кадра CAN.

1.3.9.5.6.1.6 DATA_5_8_W

Тип:

Смещение: 0x14

Размер: 4 байта

Битовый индекс	**31**	**30**	**29**	**28**	**27**	**26**	**25**	**24**
Имя поля	DATA_8							
Значение после сброса	X	X	X	X	X	X	X	X

Битовый индекс	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Имя поля	DATA_7							
Значение после сброса	X	X	X	X	X	X	X	X

Битовый индекс	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Имя поля	DATA_6							
Значение после сброса	X	X	X	X	X	X	X	X

Битовый индекс	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Имя поля	DATA_5							
Значение после сброса	X	X	X	X	X	X	X	X

DATA_5 Байт данных 5 кадра CAN.

DATA_6 Байт данных 6 кадра CAN.

DATA_7 Байт данных 7 кадра CAN.

DATA_8 Байт данных 8 кадра CAN.

1.3.9.5.6.1.7 DATA_9_12_W

Тип:

Смещение: 0x18

Размер: 4 байта

Битовый индекс	**31**	**30**	**29**	**28**	**27**	**26**	**25**	**24**
Имя поля	DATA_12							
Значение после сброса	X	X	X	X	X	X	X	X

Битовый индекс	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Имя поля	DATA_11							
Значение после сброса	X	X	X	X	X	X	X	X

Битовый индекс	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Имя поля	DATA_10							
Значение после сброса	X	X	X	X	X	X	X	X

Битовый индекс	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Имя поля	DATA_9							
Значение после сброса	X	X	X	X	X	X	X	X

DATA_9 Байт данных 9 кадра CAN.

DATA_10 Байт данных 10 кадра CAN.

DATA_11 Байт данных 11 кадра CAN.

DATA_12 Байт данных 12 кадра CAN.

1.3.9.5.6.1.8 DATA_13_16_W

Тип:

Смещение: 0x1C

Размер: 4 байта

Битовый индекс	**31**	**30**	**29**	**28**	**27**	**26**	**25**	**24**
Имя поля	DATA_16							
Значение после сброса	X	X	X	X	X	X	X	X

Битовый индекс	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Имя поля	DATA_15							
Значение после сброса	X	X	X	X	X	X	X	X

Битовый индекс	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Имя поля	DATA_14							
Значение после сброса	X	X	X	X	X	X	X	X

Битовый индекс	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Имя поля	DATA_13							
Значение после сброса	X	X	X	X	X	X	X	X

DATA_13 Байт данных 13 кадра CAN.

DATA_14 Байт данных 14 кадра CAN.

DATA_15 Байт данных 15 кадра CAN.

DATA_16 Байт данных 16 кадра CAN.

1.3.9.5.6.1.9 DATA_17_20_W

Тип:

Смещение: 0x20

Размер: 4 байта

Битовый индекс	**31**	**30**	**29**	**28**	**27**	**26**	**25**	**24**
Имя поля	DATA_20							
Значение после сброса	X	X	X	X	X	X	X	X

Битовый индекс	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Имя поля	DATA_19							
Значение после сброса	X	X	X	X	X	X	X	X

Битовый индекс	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Имя поля	DATA_18							
Значение после сброса	X	X	X	X	X	X	X	X

Битовый индекс	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Имя поля	DATA_17							
Значение после сброса	X	X	X	X	X	X	X	X

DATA_17 Байт данных 17 кадра CAN.

DATA_18 Байт данных 18 кадра CAN.

DATA_19 Байт данных 19 кадра CAN.

DATA_20 Байт данных 20 кадра CAN.

1.3.9.5.6.1.10 DATA_21_24_W

Тип:

Смещение: 0x24

Размер: 4 байта

Битовый индекс	**31**	**30**	**29**	**28**	**27**	**26**	**25**	**24**
Имя поля	DATA_24							
Значение после сброса	X	X	X	X	X	X	X	X

Битовый индекс	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Имя поля	DATA_23							
Значение после сброса	X	X	X	X	X	X	X	X

Битовый индекс	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Имя поля	DATA_22							
Значение после сброса	X	X	X	X	X	X	X	X

Битовый индекс	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Имя поля	DATA_21							

Битовый индекс	7	6	5	4	3	2	1	0
Значение после сброса	X	X	X	X	X	X	X	X

DATA_21 Байт данных 21 кадра CAN.

DATA_22 Байт данных 22 кадра CAN.

DATA_23 Байт данных 23 кадра CAN.

DATA_24 Байт данных 24 кадра CAN.

1.3.9.5.6.1.11 DATA_25_28_W

Тип:

Смещение: 0x28

Размер: 4 байта

Битовый индекс	31	30	29	28	27	26	25	24
Имя поля	DATA_28							
Значение после сброса	X	X	X	X	X	X	X	X

Битовый индекс	23	22	21	20	19	18	17	16
Имя поля	DATA_27							
Значение после сброса	X	X	X	X	X	X	X	X

Битовый индекс	15	14	13	12	11	10	9	8
Имя поля	DATA_26							
Значение после сброса	X	X	X	X	X	X	X	X

Битовый индекс	7	6	5	4	3	2	1	0
Имя поля	DATA_25							
Значение после сброса	X	X	X	X	X	X	X	X

DATA_25 Байт данных 25 кадра CAN.

DATA_26 Байт данных 26 кадра CAN.

DATA_27 Байт данных 27 кадра CAN.

DATA_28 Байт данных 28 кадра CAN.

1.3.9.5.6.1.12 DATA_29_32_W

Тип:

Смещение: 0x2C

Размер: 4 байта

Битовый индекс	31	30	29	28	27	26	25	24
Имя поля	DATA_32							
Значение после сброса	X	X	X	X	X	X	X	X

Битовый индекс	23	22	21	20	19	18	17	16
Имя поля	DATA_31							
Значение после сброса	X	X	X	X	X	X	X	X

Битовый индекс	15	14	13	12	11	10	9	8
Имя поля	DATA_30							
Значение после сброса	X	X	X	X	X	X	X	X

Битовый индекс	7	6	5	4	3	2	1	0
Имя поля	DATA_29							
Значение после сброса	X	X	X	X	X	X	X	X

DATA_29 Байт данных 29 кадра CAN.

DATA_30 Байт данных 30 кадра CAN.

DATA_31 Байт данных 31 кадра CAN.

DATA_32 Байт данных 32 кадра CAN.

1.3.9.5.6.1.13 DATA_33_36_W

Тип:

Смещение: 0x30

Размер: 4 байта

Битовый индекс	31	30	29	28	27	26	25	24
Имя поля	DATA_36							
Значение после сброса	X	X	X	X	X	X	X	X

Битовый индекс	23	22	21	20	19	18	17	16
Имя поля	DATA_35							
Значение после сброса	X	X	X	X	X	X	X	X

Битовый индекс	15	14	13	12	11	10	9	8
Имя поля	DATA_34							
Значение после сброса	X	X	X	X	X	X	X	X

Битовый индекс	7	6	5	4	3	2	1	0
Имя поля	DATA_33							
Значение после сброса	X	X	X	X	X	X	X	X

DATA_33 Байт данных 33 кадра CAN.

DATA_34 Байт данных 34 кадра CAN.

DATA_35 Байт данных 35 кадра CAN.

DATA_36 Байт данных 36 кадра CAN.

1.3.9.5.6.1.14 DATA_37_40_W

Тип:

Смещение: 0x34

Размер: 4 байта

Битовый индекс	31	30	29	28	27	26	25	24
Имя поля	DATA_40							
Значение после сброса	X	X	X	X	X	X	X	X

Битовый индекс	23	22	21	20	19	18	17	16
Имя поля	DATA_39							
Значение после сброса	X	X	X	X	X	X	X	X

Битовый индекс	15	14	13	12	11	10	9	8
Имя поля	DATA_38							

Битовый индекс	15	14	13	12	11	10	9	8
Значение после сброса	X	X	X	X	X	X	X	X

Битовый индекс	7	6	5	4	3	2	1	0
Имя поля	DATA_37							
Значение после сброса	X	X	X	X	X	X	X	X

DATA_37 Байт данных 37 кадра CAN.

DATA_38 Байт данных 38 кадра CAN.

DATA_39 Байт данных 39 кадра CAN.

DATA_40 Байт данных 40 кадра CAN.

1.3.9.5.6.1.15 DATA_41_44_W

Тип:

Смещение: 0x38

Размер: 4 байта

Битовый индекс	31	30	29	28	27	26	25	24
Имя поля	DATA_44							
Значение после сброса	X	X	X	X	X	X	X	X

Битовый индекс	23	22	21	20	19	18	17	16
Имя поля	DATA_43							
Значение после сброса	X	X	X	X	X	X	X	X

Битовый индекс	15	14	13	12	11	10	9	8
Имя поля	DATA_42							
Значение после сброса	X	X	X	X	X	X	X	X

Битовый индекс	7	6	5	4	3	2	1	0
Имя поля	DATA_41							
Значение после сброса	X	X	X	X	X	X	X	X

DATA_41 Байт данных 41 кадра CAN.

DATA_42 Байт данных 42 кадра CAN.

DATA_43 Байт данных 43 кадра CAN.

DATA_44 Байт данных 44 кадра CAN.

1.3.9.5.6.1.16 DATA_45_48_W

Тип:

Смещение: 0x3C

Размер: 4 байта

Битовый индекс	31	30	29	28	27	26	25	24
Имя поля	DATA_48							
Значение после сброса	X	X	X	X	X	X	X	X

Битовый индекс	23	22	21	20	19	18	17	16
Имя поля	DATA_47							

Битовый индекс	23	22	21	20	19	18	17	16
Значение после сброса	X	X	X	X	X	X	X	X

Битовый индекс	15	14	13	12	11	10	9	8
Имя поля	DATA_46							
Значение после сброса	X	X	X	X	X	X	X	X

Битовый индекс	7	6	5	4	3	2	1	0
Имя поля	DATA_45							
Значение после сброса	X	X	X	X	X	X	X	X

DATA_45 Байт данных 45 кадра CAN.

DATA_46 Байт данных 46 кадра CAN.

DATA_47 Байт данных 47 кадра CAN.

DATA_48 Байт данных 48 кадра CAN.

1.3.9.5.6.1.17 DATA_49_52_W

Тип:

Смещение: 0x40

Размер: 4 байта

Битовый индекс	31	30	29	28	27	26	25	24
Имя поля	DATA_52							
Значение после сброса	X	X	X	X	X	X	X	X

Битовый индекс	23	22	21	20	19	18	17	16
Имя поля	DATA_51							
Значение после сброса	X	X	X	X	X	X	X	X

Битовый индекс	15	14	13	12	11	10	9	8
Имя поля	DATA_50							
Значение после сброса	X	X	X	X	X	X	X	X

Битовый индекс	7	6	5	4	3	2	1	0
Имя поля	DATA_49							
Значение после сброса	X	X	X	X	X	X	X	X

DATA_49 Байт данных 49 кадра CAN.

DATA_50 Байт данных 50 кадра CAN.

DATA_51 Байт данных 51 кадра CAN.

DATA_52 Байт данных 52 кадра CAN.

1.3.9.5.6.1.18 DATA_53_56_W

Тип:

Смещение: 0x44

Размер: 4 байта

Битовый индекс	31	30	29	28	27	26	25	24
Имя поля	DATA_54							

Битовый индекс	31	30	29	28	27	26	25	24
Значение после сброса	X	X	X	X	X	X	X	X

Битовый индекс	23	22	21	20	19	18	17	16
Имя поля	DATA_55							
Значение после сброса	X	X	X	X	X	X	X	X

Битовый индекс	15	14	13	12	11	10	9	8
Имя поля	DATA_56							
Значение после сброса	X	X	X	X	X	X	X	X

Битовый индекс	7	6	5	4	3	2	1	0
Имя поля	DATA_53							
Значение после сброса	X	X	X	X	X	X	X	X

DATA_53 Байт данных 53 кадра CAN.

DATA_56 Байт данных 56 кадра CAN.

DATA_55 Байт данных 55 кадра CAN.

DATA_54 Байт данных 54 кадра CAN.

1.3.9.5.6.1.19 DATA_57_60_W

Тип:

Смещение: 0x48 **Размер:** 4 байта

Битовый индекс	31	30	29	28	27	26	25	24
Имя поля	DATA_60							

Битовый индекс	31	30	29	28	27	26	25	24
Значение после сброса	X	X	X	X	X	X	X	X

Битовый индекс	23	22	21	20	19	18	17	16
Имя поля	DATA_59							
Значение после сброса	X	X	X	X	X	X	X	X

Битовый индекс	15	14	13	12	11	10	9	8
Имя поля	DATA_58							
Значение после сброса	X	X	X	X	X	X	X	X

Битовый индекс	7	6	5	4	3	2	1	0
Имя поля	DATA_57							
Значение после сброса	X	X	X	X	X	X	X	X

DATA_57 Байт данных 57 кадра CAN.

DATA_58 Байт данных 58 кадра CAN.

DATA_59 Байт данных 59 кадра CAN.

DATA_60 Байт данных 60 кадра CAN.

1.3.9.5.6.1.20 DATA_61_64_W

Тип:

Смещение: 0x4C

Размер: 4 байта

Битовый индекс	31	30	29	28	27	26	25	24
Имя поля	DATA_64							
Значение после сброса	X	X	X	X	X	X	X	X

Битовый индекс	23	22	21	20	19	18	17	16
Имя поля	DATA_63							
Значение после сброса	X	X	X	X	X	X	X	X

Битовый индекс	15	14	13	12	11	10	9	8
Имя поля	DATA_62							
Значение после сброса	X	X	X	X	X	X	X	X

Битовый индекс	7	6	5	4	3	2	1	0
Имя поля	DATA_61							
Значение после сброса	X	X	X	X	X	X	X	X

DATA_61 Байт данных 61 кадра CAN.

DATA_62 Байт данных 62 кадра CAN.

DATA_63 Байт данных 63 кадра CAN.

DATA_64 Байт данных 64 кадра CAN.

1.3.9.5.6.1.21 FRAME_TEST_W

Тип:

Смещение: 0x50

Размер: 4 байта

Битовый индекс	**31**	**30**	**29**	**28**	**27**	**26**	**25**	**24**
Имя поля	Зарезервировано							
Значение после сброса	-	-	-	-	-	-	-	-

Битовый индекс	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Имя поля	Зарезервировано							
Значение после сброса	-	-	-	-	-	-	-	-

Битовый индекс	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Имя поля	Зарезервировано			TPRM				
Значение после сброса	-	-	-	X	X	X	X	X

Битовый индекс	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Имя поля	Зарезервировано					SDLC	FCRC	FSTC
Значение после сброса	-	-	-	-	-	X	X	X

FSTC Инвертировать бит поля счетчика наполнения, когда этот кадр передается. Это поле действует только в переданных кадрах.

FCRC Инвертировать бит поля CRC, когда этот кадр передается. Это поле действует только в переданных кадрах.

SDLC Поменять DLC в переданном кадре.

TPRM Тестовый параметр

1.3.9.6 Контроллер интерфейса SPI (SPI)

1.3.9.6.1 Характеристики контроллера SPI

Контроллер SPI обладает следующими характеристиками:

– поддерживаемые режимы работы:

a) master;

b) slave;

– параметры обмена:

a) длина посылки: 8 - 16 бит;

b) настройка полярности и фазы;

c) изменение эндианности;

– поддерживаемые протоколы:

a) формат Motorolla (SPI);

b) формат Texas Instruments (SSI);

c) формат National Semiconductor (Microwave);

– размер RX буфера (байт): 32;

– размер TX буфера (байт): 32;

– прескейлер;

– генерация маскируемых прерываний;

– интерфейс программирования и передачи данных APB slave.

1.3.9.6.2 Структурная схема контролера SPI

Структурная схема контроллера SPI представлена на рисунке 1.67. Описание блоков структурной схемы контроллера SPI представлено в таблице 1.288.

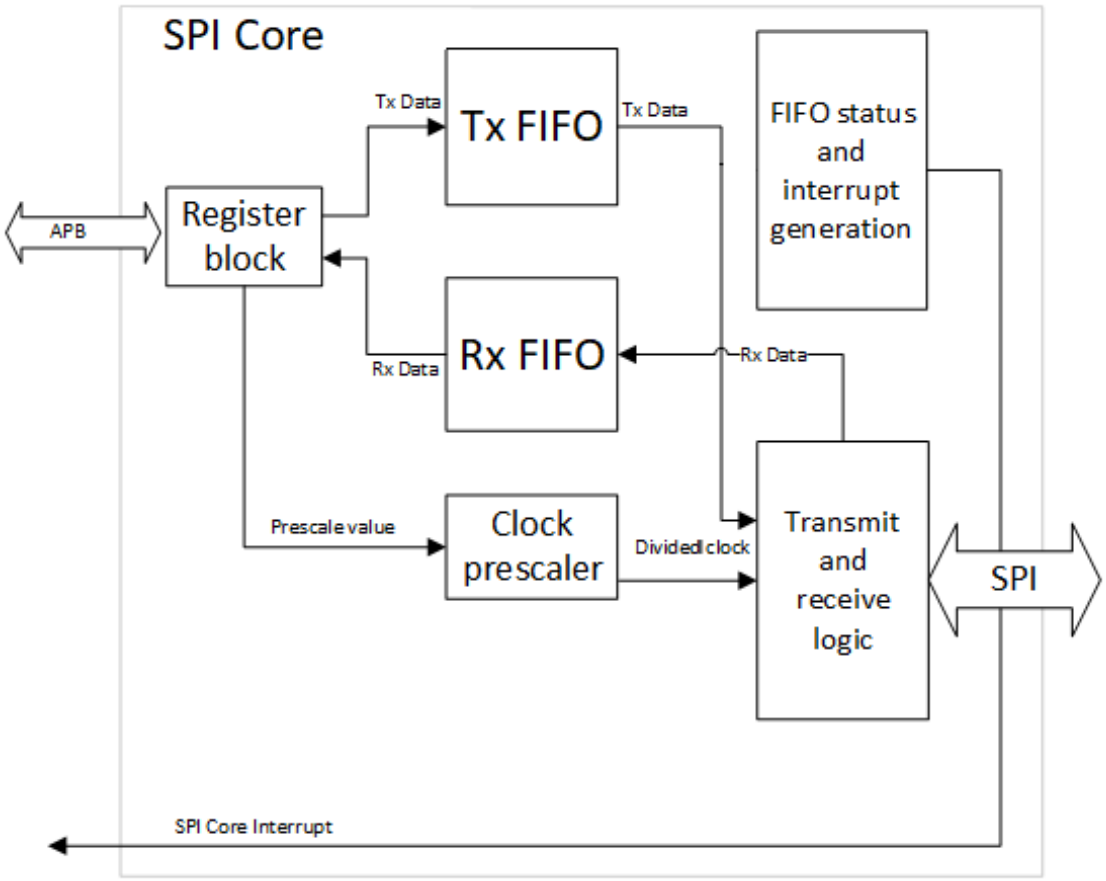


Рисунок 1.67 – Структурная схема контроллера SPI

Таблица 1.288 – Описание блоков на структурной схеме контроллера SPI

Название блока	**Назначение и основные функции**
SPI Core	Контроллер SPI
SPI Core. Register block	Блок регистров в составе контроллера SPI
SPI Core. TxFIFO	Передающий буфер FIFO
SPI Core. RxFIFO	Принимающий буфер FIFO
SPI Core. Clock prescaler	Генератор синхросигнала SPI
SPI Core. Transmit and receive logic	Блок, реализующий интерфейс SPI
SPI Core. FIFO status and interrupt generation	Блок состояния буферов FIFO и генерации запросов прерывания.

У контроллера SPI есть возможность снятия тактирования (режим низкого энергопотребления).

Внешняя настройка тактировани контроллеров SPI, задается битами SPI_x_EN (SPI_A_EN, SPI_B_EN, SPI_C_EN, SPI_D_EN) в регистре RCU_PCER0 блока RCU (см. 1.3.7.7.16).

Программный доступ при снятом тактировании будет возвращать ошибку системной шины и вызывать исключение процессорного ядра.

1.3.9.6.3 Описание внешних выводов контроллера SPI

Список внешних выводов контроллера SPI и их описание приведены в таблице 1.289.

Таблица 1.289 – Список внешних выводов контроллера SPI

Наименование вывода	Тип (Вх/Вых)	Функциональное назначение
RX	Вх	Линия для приема данных: В режиме SPI_Master - SPI_MISO В режиме SPI_Slave - SPI_MOSI
TX	Вых	Линия для передачи данных: В режиме SPI_Master - SPI_MOSI В режиме SPI_Slave - SPI_MISO
SCK	Вх/Вых	Линия для приёма/передачи синхросигнала: В режиме SPI_Master - Вых. В режиме SPI_Slave - Вх.
SS	Вх/Вых	Линия для выбора Slave устройства: В режиме SPI_Master - Вых. В режиме SPI_Slave - Вх.

1.3.9.6.4 Работа контроллера SPI

Контроллер SPI представляет собой интерфейс синхронного последовательного обмена данными с периферийными устройствами, способный функционировать в качестве ведущего устройства и поддерживающий протоколы передачи данных SPI фирмы Motorola, Microwire фирмы National Semiconductor, а также SSI фирмы Texas Instruments.

Контроллер SPI осуществляет преобразование данных, полученных от периферийного устройства, из последовательной в параллельную форму. Через APB считываются и записываются данные, а также управляющаю информация и информация о состоянии блока. Приемные и передаваемые данные буферизуются с помощью буферов FIFO, обеспечивающих хранение до 8 слов шириной 16 бит, независимо для режимов приема и передачи.

Контроллер SPI содержит программируемые делитель и предделитель частоты, формирующие тактовый сигнал обмена данными.

Режим обмена данными контроллера SPI, формат сообщения и количество бит данных задаются программно с помощью регистров управления SPI_CR0 и SPI_CR1.

Контроллер SPI формирует четыре маскируемых запроса прерывания:

– SPI_TXINTR – запрос на обслуживание буфера передачи;

– SPI_RXINTR – запрос на обслуживание буфера приема;

– SPI_RORINTR – переполнение буфера приема;

– SPI_RTINTR – истечение периода ожидания при наличии данных в буфере приема.

Данные запросы формируют общий сигнал прерывания, возникающий в случае активности одного из вышеуказанных независимых немаскированных прерываний.

В зависимости от выбранного режима работы контроллера SPI сигнал SS используется:

– или для синхронизации сообщений (интерфейс SSI, активное состояние – высокий уровень),

– или для выбора ведомого режима (интерфейсы SPI и Microwire, активное состояние – низкий уровень).

1.3.9.6.4.1 Структурная схема контроллера SPI

Структурная схема контроллера SPI представлена на рисунке 1.67. Ниже приведено описание блоков, входящих в контроллер SPI.

Блок регистров (Register block)

Блок регистров (Register block) осуществляет хранение записанных данных, или данных, подлежащих считыванию, через интерфейс AMBA APB.

Предделитель частоты (Clock prescaler)

Контроллер SPI формирует тактовый сигнал обмена данными SCK с помощью внутреннего предделителя частоты (Clock prescaler), состоящего из двух последовательно соединенных счетчиков без цепи сброса.

Для деления частоты сигнала SPI_CLK (SPI_CLK_A(B,C,D)) приходящего от RCU (см. 1.3.7) можно задать коэффициент предварительного деления частоты в диапазоне от 2 до 254 с шагом 2 путем записи значения в регистр SPI_CPSR. Так как младший значащий разряд регистра SPI_CPSR не используется, исключается возможность деления частоты на нечетный коэффициент, что, в свою очередь, гарантирует формирование тактового сигнала симметричной формы (с одинаковой длительностью полупериодов высокого и низкого уровня).

Полученное значение снова делится на коэффициент, задаваемый программно в диапазоне от 1 до 256 путем записи соответствующего значения в регистр управления SPI_CR0, для получения конечного тактового сигнала обмена данными SCK для ведущего режима.

FIFO буфер передачи (Tx FIFO)

FIFO буфер передачи (Tx FIFO) имеет ширину 16 бит, глубину 8 слов, схему организации доступа типа «первый вошел, первый вышел» (FIFO – First-In, First-Out). Данные,

записанные через шину AMBA APB, сохраняются в буфере до тех пор, пока не будут считаны блоком передачи данных.

Параллельно поступающие данные записываются в буфер передачи перед преобразованием в последовательную форму и передачей подключенному ведомому устройству через выходной контакт TX.

FIFO буфер приема (Rx FIFO)

FIFO буфер приема (Rx FIFO) имеет ширину 16 бит, глубину 8 слов, схему организации доступа типа «первый вошел, первый вышел». Данные, принятые от периферийного устройства через последовательный интерфейс, сохраняются блоком приема данных в нем до тех пор, пока не будут считаны через шину AMBA APB.

Блок приема и передачи данных (Transmit and receive logic)

Контроллер SPI формирует тактовый сигнал обмена данными для подключенных ведомых устройств, формируемый путем деления частоты сигнала SPI_CLK, как уже было описано ранее.

Блок передачи последовательно считывает значения из буфера передачи и производит их преобразование из параллельной в последовательную форму. Далее поток последовательных данных и сигнал SS, тактированные сигналом SCK, передаются по линии TX к подключенным ведомым устройствам. Блок приема выполняет преобразование данных, поступающих синхронно с линии RX, из последовательной в параллельную форму, после чего загружает их в буфер приема, откуда они могут быть считаны через интерфейс шины APB.

Блок формирования прерываний (FIFO status and interrupt generation)

Контроллер SPI генерирует комбинированное прерывание как функция «ИЛИ» независимых запросов на прерывание. Активным уровнем сигнала запроса прерывания является высокий.

1.3.9.6.4.2 Настройка контроллера SPI

После сброса работа контроллера SPI блокируется до выполнения процедуры задания конфигурации.

Для этого необходимо запрограммировать регистры управления SPI_CR0 и SPI_CR1 для конфигурирования периферийного устройства, работающего по одному из следующих протоколов передачи данных:

– SPI фирмы Motorola;

– SSI фирмы Texas Instruments;

– Microwave фирмы National Semiconductor.

– Для установки требуемой скорости передачи данных необходимо запрограммировать регистр предварительного делителя частоты SPI_CPSR.

1.3.9.6.4.3 Разрешение работы контроллера SPI

Необходимо либо инициализировать буфер передачи путем записи в него до восьми 16-разрядных слов при неработающем контроллере SPI, либо разрешить прерывание запросом на обслуживание буфера FIFO передатчика. После включения SPI блок приема и передачи данных начинает обмен данными по линиям передачи TX и приема RX.

1.3.9.6.4.4 Соотношения между тактовыми сигналами

Частота входного тактового сигнала контроллера SPI должна обеспечивать поддержку требуемого диапазона скоростей обмена данными. Отношение минимальной частоты сигнала контроллера SPI к максимальной частоте сигнала SCK составляет два.

Для обеспечения максимальной скорости обмена 1,8432 Мбит/с частота сигнала контроллера SPI должна составлять не менее 3,6864 МГц. При частоте тактового сигнала контроллера SPI, равной 3,6864 МГц, в регистр SPI_CPSR должно быть записано значение 2, а поле SCR[7:0] регистра SPI_CR0 должно быть установлено в 0.

Минимальная допустимая частота тактового сигнала контроллера SPI определяется следующим неравенством:

$FSPI_CLK(min) \geq 2\,x\,FSCK(max)$

Максимальная допустимая частота тактового сигнала контроллера SPI определяется следующим неравенством:

$FSPI_CLK(max) \geq 254\,x\,256\,x\,FSCK(min)$

1.3.9.6.4.5 Программирование регистра управления SPI_CR0

Более подробная информация о присваивании значений битам данного регистра приведена в пп. 1.3.9.6.6.1 (описание полей регистра SPI_CR0).

Регистр SPI_CR0 предназначен для:

– задания тактовой частоты, определяющей скорость информационного обмена;

– выбора одного из трех протоколов обмена данными;

– выбора размера слова данных.

Значение тактовой частоты, определяющей скорость информационного обмена (Serial Clock Rate (SCR) – тактовая частота, определяющая скорость информационного обмена),

наряду со значением коэффициента деления тактового сигнала CPSDVSR регистра SPI_CPSR используется для определения скорости передачи и приема данных в зависимости от частоты тактового сигнала контроллера SPI.

Формат сообщения задается путем установки значения поля FRF, а размер слова данных – путем установки значения поля DSS.

Для протокола SPI фирмы Motorola, кроме того, задается полярность и фаза сигнала (биты SPH и SPO).

1.3.9.6.4.6 Программирование регистра управления SPI_CR1

Более подробная информация о присваивании значений битам данного регистра приведена в пп.1.3.9.6.6.2 (описание полей регистра SPI_CR1).

Регистр SPI_CR1 предназначен для:

– выбора режима работы master или slave;

– настройки функции тестирования с обратной связью;

– включение контроллера SPI.

Для включение контроллера SPI необходимо записать в поле SSE значение «1».

1.3.9.6.4.7 Формирование тактового сигнала обмена данными

Тактовый сигнал обмена данными формируется путем деления частоты тактового сигнала контроллера SPI SPI_CLK. На первом этапе формирования частота данного сигнала делится на четный коэффициент CPSDVSR, находящийся в диапазоне от 2 до 254 и доступный для программирования через регистр SPI_CPSR. Сформированный сигнал далее поступает на делитель частоты с коэффициентом (1 + SCR) в диапазоне от 1 до 256, где значение SCR доступно для программирования через SPI_CR0.

Частота выходного тактового сигнала обмена данными SCK определяется следующим равенством

$$F_{SCK} = \frac{F_{SPI_CLK}}{CPSDSVR\ X\ (1 + SCR)}$$

Например, в случае если частота сигнал SPI_CLK составляет 3,6864 МГц, а значение CPSDVSR = 2, частота сигнала SCK находится в интервале от 7,2 кГц до 1,8432 МГц.

1.3.9.6.4.8 Формат сообщений

Каждое сообщение содержит, в зависимости от запрограммированного значения, от 4 до 16 бит данных. Передача данных начинается со старшего значащего разряда. Можно выбрать одну из следующих структур сообщения:

– SSI фирмы Texas Instruments;

– SPI фирмы Motorola;

– Microwire фирмы National Semiconductor.

Во всех режимах построения сообщения тактовый сигнал SCK находится в неактивном состоянии, когда контроллер SPI находится в неактивном состоянии, и переход в другой режим с запрограммированной частотой производится только при активной передаче или приеме данных. Перевод сигнала SCK в неактивное состояние используется как признак режима ожидания приемника, то есть наличия в буфере приема необработанных данных по истечении заданного интервала времени.

В режимах SPI и Microwire выходной сигнал SS имеет активный низкий уровень и поддерживается в низком уровне в течение всего периода передачи сообщения.

В режиме SSI фирмы Texas Instruments перед началом каждого сообщения на выходе SS формируется импульс длительностью один такт. В данном режиме блок приема и передачи данных контроллера SPI, равно как ведомое «внекристальное» устройство, передает данные на линию по переднему фронту сигнала SCK, а считывает данные с другого устройства из линии по заднему фронту этого сигнала.

В отличие от полнодуплексных режимов передачи данных SSI и SPI, режим Microwire фирмы National Semiconductor использует специальный способ обмена данными между ведущим и ведомым устройствами, функционирующий в режиме полудуплекса. В данном режиме на «внекристальное» ведомое устройство перед началом передачи сообщения посылается специальная восьмибитная управляющая последовательность. В течение всего времени передачи данной последовательности приемник не обрабатывает каких-либо входных данных. После того, как сигнал передан и декодирован ведомым устройством, оно выдерживает паузу в один такт после передачи последнего бита восьмибитной управляющей последовательности, после чего передает в адрес ведущего устройства запрошенные данные. Длина сообщения от ведомого устройства может составлять от 4 до 16 бит. Таким образом, общая длина сообщения составляет от 13 до 25 бит.

Формат последовательного синхронного обмена данными SSI фирмы Texas Instruments

На рисунке 1.68 продемонстрирован формат последовательного синхронного обмена данными протокола SSI фирмы Texas Instruments для одиночного обмена.

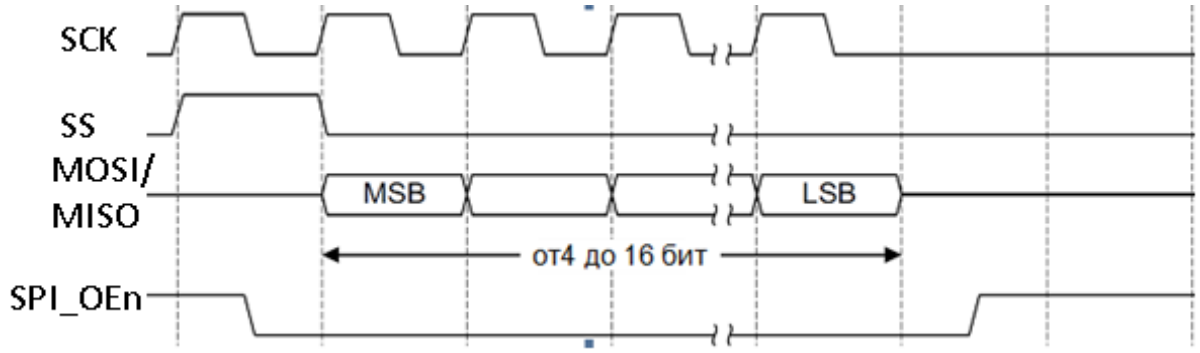


Рисунок 1.68 – Формат синхронного обмена протокола SSI фирмы Texas Instruments (одиночный обмен)

В данном режиме при неактивном контроллере SPI сигналы SCK и SS переводятся в низкий логический уровень, а линия передачи данных MOSI поддерживается в третьем состоянии. После появления хотя бы одного элемента в буфере передачи сигнал SS переводится в высокий логический уровень на время, соответствующее одному периоду сигнала SCK. Значение из буфера при этом переносится в сдвиговый регистр блока передачи. По следующему переднему фронту сигнала SCK старший значащий разряд сообщения (4 – 16 бит данных) выдается на выход линии MOSI. Аналогично, старший значащий разряд сообщения сдвигается на выход линии MISO ведомым устройством последовательной передачи данных.

SPI устройства последовательно загружают биты данных в сдвиговый регистр по заднему фронту сигнала SCK. Принятые данные переносятся из сдвигового регистра в буфер приема после загрузки в него младшего значащего бита данных по очередному переднему фронту тактового сигнала APB интерфейса (PCLK).

Временная диаграмма последовательного синхронного обмена по протоколу SSI фирмы Texas Instruments при непрерывной передаче данных представлена на рисунке 1.69.

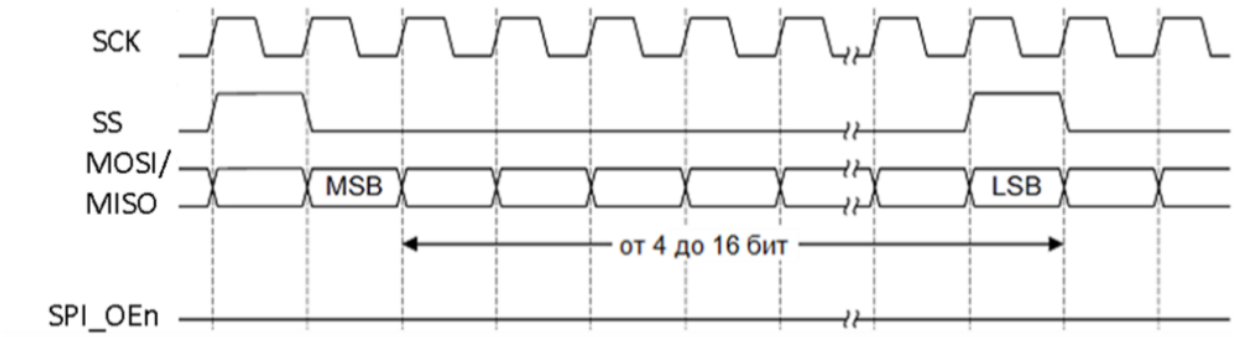


Рисунок 1.69 – Формат последовательного синхронного обмена по протоколу SSI фирмы Texas Instruments (непрерывный обмен)

Формат обмена данными SPI фирмы Motorola.

Интерфейс SPI фирмы Motorola использует четыре сигнальных линии, при этом сигнал SS выполняет функцию выбора ведомого устройства. Основной функцией протокола SPI фирмы Motorola является возможность выбора состояния и фазы сигнала SCK в режиме ожидания (при неактивном блоке приема и передачи данных) путем задания значений битов SPO и SPH регистра управления SPI_CR0.

Выбор полярности тактового сигнала – бит SPO.

Если бит SPO равен 0, то в режиме ожидания линия SCK переводится в низкий логический уровень. В противном случае при отсутствии обмена данными линия SCK переводится в высокий логический уровень.

Выбор фазы тактового сигнала – бит SPH.

Значение бита SPH определяет фронт тактового сигнала, по которому осуществляется выборка данных и изменение состояния на выходной линии. Оно оказывает значительное влияние на первый бит, передаваемый по разрешению или запрету тактового перехода до первого обнаружения фронта тактового сигнала.

В случае если бит SPH установлен в низкий уровень, регистрация данных приемником осуществляется после первого обнаружения фронта тактового сигнала.

Если бит SPH установлен в высокий уровень, регистрация данных приемником осуществляется после второго обнаружения фронта тактового сигнала.

Формат обмена данными SPI фирмы Motorola, SPO=0, SPH=0.

На рисунках 1.70 и 1.71 показаны временные диаграммы одиночного и непрерывного обмена в режиме SPI с SPO=0, SPH=0.

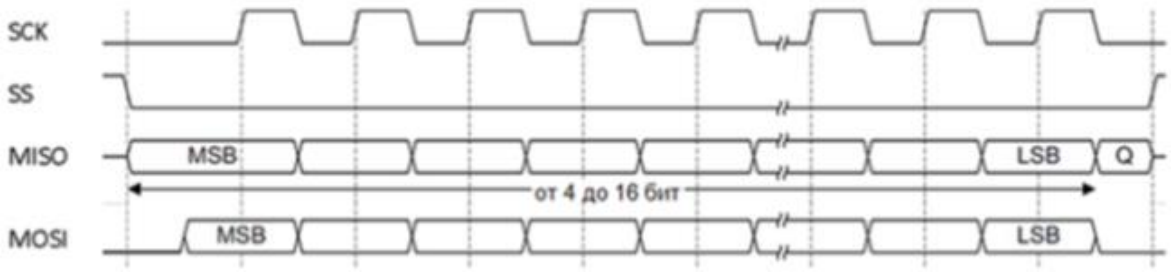


Рисунок 1.70 – Формат обмена по протоколу SPI фирмы Motorola, SPO=0, SPH=0 (одиночный обмен)

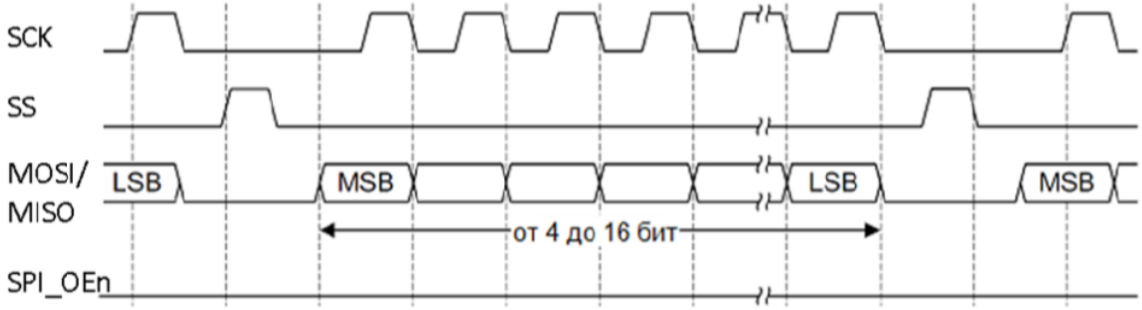


Рисунок 1.71 – Формат обмена по протоколу SPI фирмы Motorola, SPO=0, SPH=0 (непрерывный обмен)

В данном режиме во время ожидания:

– сигнал SCK переводится в низкий логический уровень;

– сигнал SS переводится в высокий логический уровень;

– линия передачи данных MOSI произвольно переводится в низкий логический уровень.

Если работа контроллера SPI разрешена и в буфере передачи содержатся допустимые данные, начало передачи инициализируется переводом сигнала SS в низкий логический уровень. Данное событие запускает передачу данных на линию MOSI.

По истечении полутакта сигнала SCK на линию MOSI передаются допустимые данные. К этому моменту должны быть сформированы данные на линиях обмена как контроллера SPI, так и ответного устройства. По истечении следующего полутакта сигнал SCK переводится в высокий логический уровень.

Далее данные регистрируются по переднему фронту и выдаются на линию по заднему фронту сигнала SCK.

В случае передачи одного слова данных после приема его последнего бита линия SS переводится в высокий логический уровень по истечении одного периода тактового сигнала SCK.

Тем не менее, в режиме непрерывной передачи данных на линии SS должны формироваться импульсы высокого логического уровня между передачами каждого слова данных. Это связано с тем, что в режиме SPH=0 линия выбора ведомого устройства в низком уровне блокирует запись в сдвиговый регистр. По завершении приема последнего бита блока данных линия SS переводится в состояние, соответствующее режиму ожидания, по истечении одного такта сигнала SCK.

Формат обмена данными SPI фирмы Motorola, SPO=0, SPH=1

На рисунке 1.72 продемонстрирована временная диаграмма одиночного и непрерывного обмена данными в режиме SPI с SPO=0, SPH=1.

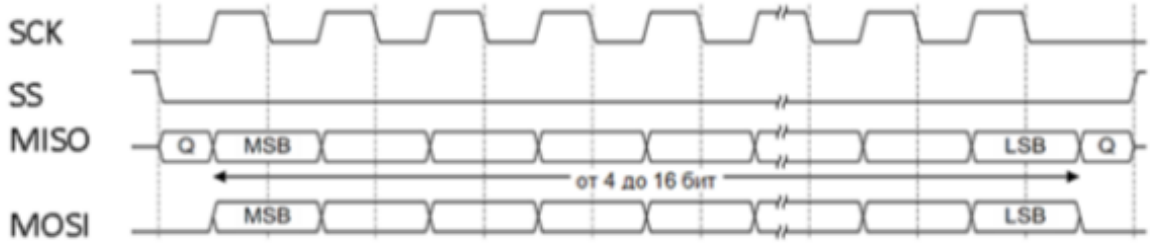


Рисунок 1.72 – Формат обмена по протоколу SPI фирмы Motorola, SPO=0,SPH=1 +(одиночный и непрерывный обмен)

В данном режиме во время ожидания:

– сигнал SCK переводится в низкий логический уровень;

– сигнал SS переводится в высокий логический уровень;

– линия передачи данных MOSI произвольно переводится в низкий логический уровень.

Если работа контроллера SPI разрешена и в буфере передачи содержатся допустимые данные, начало передачи инициализируется сигналом SS от контроллера SPI, который переводится в низкий логический уровень. По истечении полутакта сигнала SCK на соответствующих линиях обмена как ведущего, так и ведомого устройств разрешена передача данных. В это же время включается линия SCK, на которой формируется передний фронт сигнала.

Далее данные регистрируются по заднему фронту и выдаются на линию по переднему фронту сигнала SCK.

В случае передачи одного слова данных после приема его последнего бита линия SS переводится в высокий логический уровень по истечении одного периода тактового сигнала SCK.

В режиме непрерывной передачи данных линия SS находится в низком логическом уровне между последовательными сообщениями и переводится в высокий уровень по окончании выдачи последнего бита блока данных.

Формат обмена данными SPI фирмы Motorola, SPO=1, SPH=0

Временные диаграммы одиночного и непрерывного обмена в режиме SPI с SPO=1, SPH=0 представлены на рисунках 1.73 и 1.74.

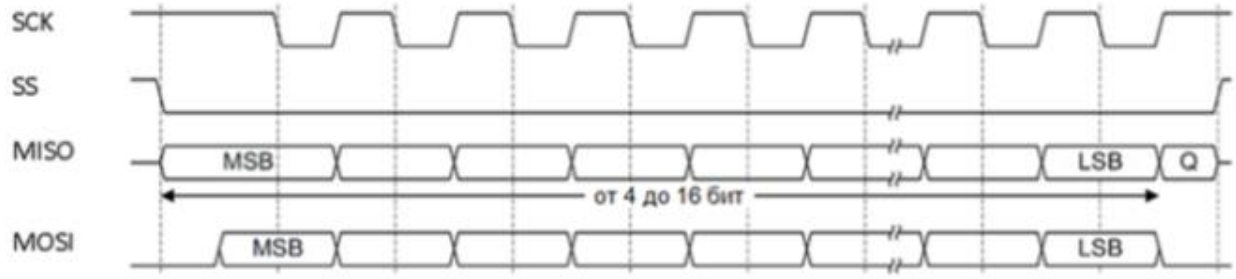


Рисунок 1.73 – Формат обмена по протоколу SPI фирмы Motorola, SPO=1, SPH=0 (одиночный обмен)

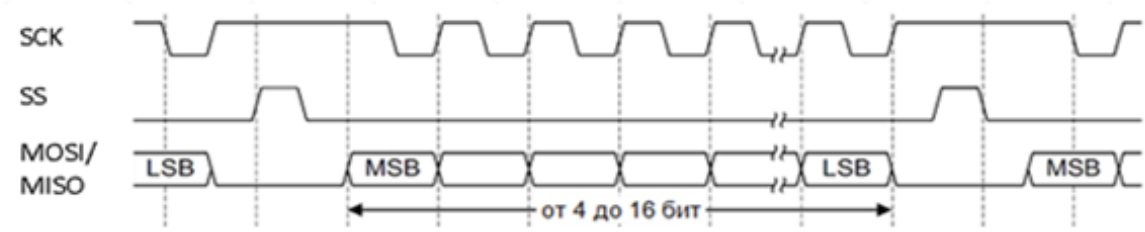


Рисунок 1.74 – Формат обмена по протоколу SPI фирмы Motorola, SPO=1 SPH=0 (непрерывный обмен)

В данном режиме во время ожидания:

– сигнал SCK переводится в высокий логический уровень;

– сигнал SS переводится в высокий логический уровень;

– линия передачи данных MOSI произвольно переводится в низкий логический уровень.

Если работа контроллера SPI разрешена и в буфере передачи содержатся корректные данные, сигнал SS переводится в низкий логический уровень, что указывает на начало обмена

данными и разрешает передачу данных от ведомого устройства на входную линию MISO ведущего.

По истечении полутакта сигнала на линию MOSI передаются допустимые данные от ведущего устройства. К этому моменту должны быть сформированы данные на линиях обмена как ведущего, так и ведомого устройства. По истечении следующего полутакта сигнал SCK переводится в низкий логический уровень. Это значит, что данные регистрируются по заднему фронту и выдаются на линию по переднему фронту сигнала SCK.

В случае передачи одного слова данных после приема его последнего бита линия SS переводится в высокий логический уровень по истечении одного периода тактового сигнала SCK.

В режиме непрерывной передачи данных линия SS находится в низком логическом уровне между последовательными информационными словами и переводится в высокий уровень по окончании приема последнего бита блока данных, как и в режиме передачи одного слова.

Формат обмена данными SPI фирмы Motorola, SPO=1, SPH=0

Временнные диаграммы одиночного и непрерывного обмена в режиме SPI с SPO=1, SPH=0 представлены на рисунках 1.75 и 1.76.

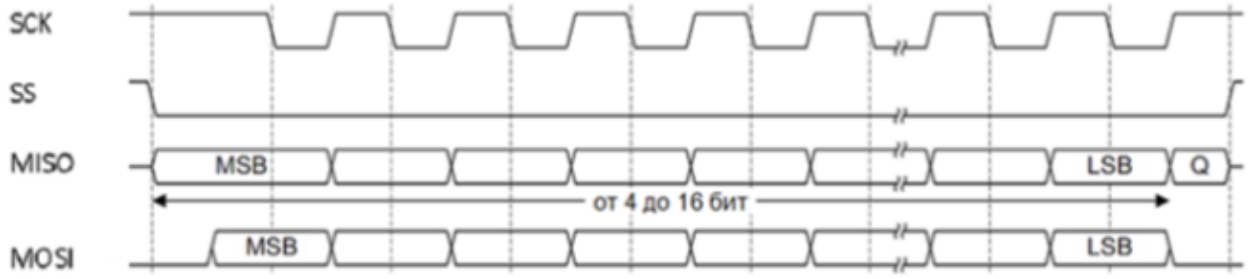


Рисунок 1.75 – Формат обмена по протоколу SPI фирмы Motorola, SPO=1, SPH=0 (одиночный обмен)

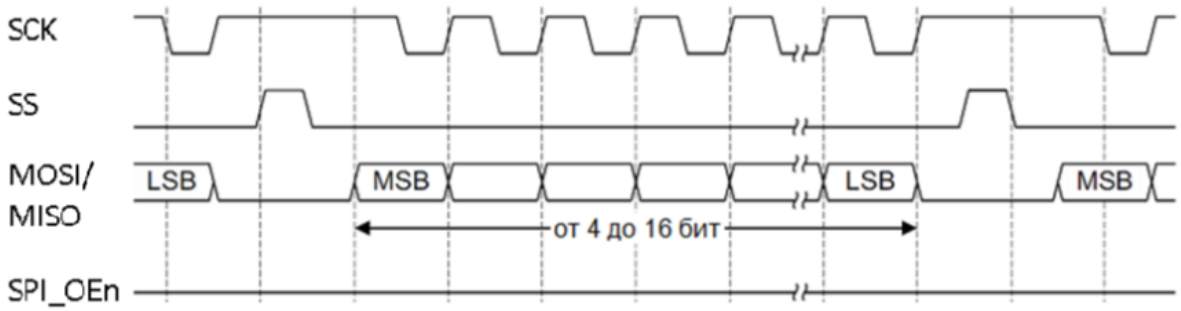


Рисунок 1.76 – Формат обмена по протоколу SPI фирмы Motorola, SPO=1 SPH=0 (непрерывный обмен)

В данном режиме во время ожидания:

– сигнал SCK переводится в высокий логический уровень;

– сигнал SS переводится в высокий логический уровень;

– линия передачи данных MOSI произвольно переводится в низкий логический уровень.

Если работа контроллера SPI разрешена и в буфере передачи содержатся корректные данные, сигнал SS переводится в низкий логический уровень, что указывает на начало обмена данными и разрешает передачу данных от ведомого устройства на входную линию MISO ведущего.

По истечении полутакта сигнала на линию MOSI передаются допустимые данные от ведущего устройства. К этому моменту должны быть сформированы данные на линиях обмена как ведущего, так и ведомого устройства. По истечении следующего полутакта сигнал SCK переводится в низкий логический уровень. Это значит, что данные регистрируются по заднему фронту и выдаются на линию по переднему фронту сигнала SCK.

В случае передачи одного слова данных после приема его последнего бита линия SS переводится в высокий логический уровень по истечении одного периода тактового сигнала SCK.

Тем не менее, в режиме непрерывной передачи данных на линии SS должны формироваться импульсы высокого логического уровня между передачами каждого слова данных. Это связано с тем, что в режиме SPH=0 линия выбора ведомого устройства в низком уровне блокирует запись в сдвиговый регистр. По завершении приема последнего бита блока данных линия SS переводится в состояние, соответствующее режиму ожидания, по истечении одного такта сигнала SCK.

Формат обмена данными SPI фирмы Motorola, SPO=1, SPH=1

На рисунке 1.77 продемонстрирована временная диаграмма одиночного и непрерывного обмена данными в режиме SPI с SPO=1, SPH=1.

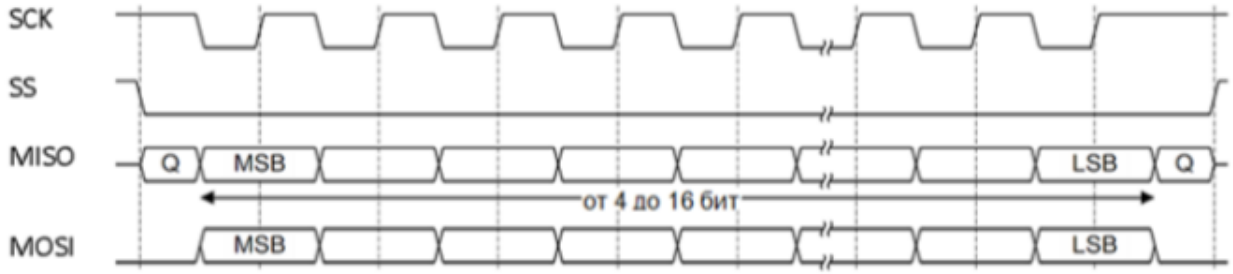


Рисунок 1.77 – Формат обмена по протоколу SPI фирмы Motorola, SPO=1,SPH=1 (одиночный и непрерывный обмен)

В данном режиме во время ожидания блок приема и передачи данных:

– сигнал SCK переводится в высокий логический уровень;

– сигнал SS переводится в высокий логический уровень;

– линия передачи данных MOSI произвольно переводится в низкий логический уровень.

Если работа контроллера SPI разрешена и в буфере передачи содержатся допустимые данные, начало передачи инициализируется сигналом SS от ведущего устройства, который переводится в низкий логический уровень. По истечении полутакта сигнала SCK на соответствующих линиях обмена как ведущего, так и ведомого устройств разрешена передача данных. В это же время включается линия SCK, на которой формируется задний фронт сигнала. Далее данные регистрируются по переднему фронту и выдаются на линию по заднему фронту сигнала SCK.

В случае передачи одного слова данных после приема его последнего бита линия SS переводится в высокий логический уровень по истечении одного периода тактового сигнала SCK.

В режиме непрерывной передачи данных линия SS постоянно находится в низком логическом уровне до окончания приема последнего бита блока данных, после чего возвращается в состояние незанятости, как описано выше.

В режиме непрерывной неравномерной передачи данных линия SS находится в низком логическом уровне между последовательными информационными словами и переводится в высокий уровень по окончании приема последнего бита блока данных, как и в режиме передачи одного слова.

Формат обмена данными Microwire фирмы National Semiconductor.

На рисунке 1.78 представлена диаграмма одиночного обмена в режиме Microwire, предлагаемом фирмой National Semiconductor.

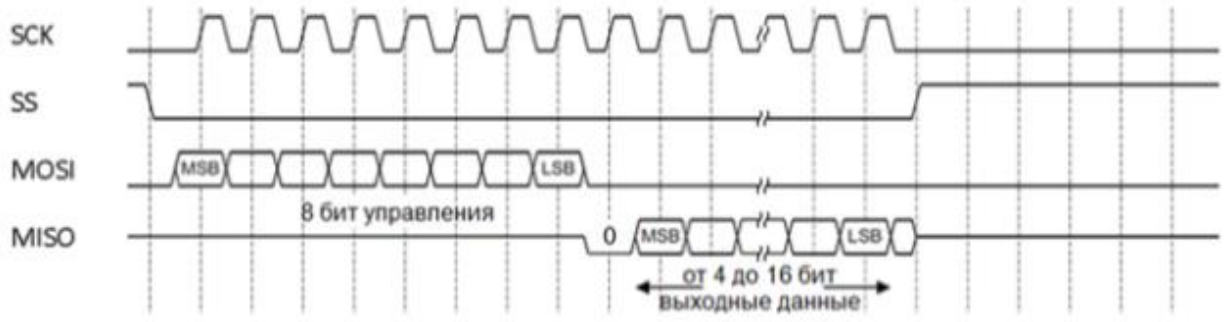


Рисунок 1.78 – Формат обмена по протоколу Microwire фирмы National Semiconductor (одиночный обмен)

Протокол передачи данных Microwire во многом схож с протоколом SPI, за исключением того, что обмен данными между ведущим и ведомыми устройствами в нем осуществляется в полудуплексном режиме вместо полнодуплексного. Каждый информационный обмен начинается с передачи ведущим ведомому устройству специального восьмибитного управляющего слова. В течение всего времени передачи данной последовательности ведущий не обрабатывает каких-либо входных данных. После того, как сигнал передан и декодирован ведомым устройством, оно выдерживает паузу в один тактовый интервал после передачи последнего бита восьмибитной управляющей последовательности, после чего передает в адрес ведущего устройства запрошенные данные. Длина сообщения от ведомого устройства может составлять от 4 до 16 бит. Таким образом, общая длина сообщения составляет от 13 до 25 бит.

В данном режиме во время ожидания:

– сигнал SCK переводится в низкий логический уровень;

– сигнал SS переводится в высокий логический уровень;

– линия передачи данных MOSI произвольно переводится в низкий логический уровень.

Переход в режим информационного обмена происходит после записи управляющего байта в буфер передачи. По заднему фронту сигнала SS данные из буфера передачи переносятся в сдвиговый регистр блока передачи, откуда, начиная со старшего значащего разряда, восьмибитное слово данных сдвигается без сохранения выдвигаемых разрядов в линию MOSI. Линия SS остается в низком логическом уровне в течение всей передачи сообщения. Линия MISO в ходе данной передачи находится третьем состоянии.

Ведомое устройство осуществляет прием каждого управляющего бита данных в сдвиговый регистр по переднему фронту каждого сигнала SCK.

По окончании приема последнего бита ведомым устройством управляющий байт декодируется в течение одного тактового интервала, после чего ведомое устройство передает запрошенные данные в адрес ведущего. Биты данных передаются на линию MISO по заднему фронту сигнала SCK. Ведущий, в свою очередь, регистрирует каждый бит по переднему фронту тактового сигнала SCK. В случае одиночного информационного обмена по окончании приема последнего бита слова данных сдвиговым регистром сигнал SS переводится в высокий уровень на время, соответствующее одному тактовому интервалу, что служит командой для переноса принятого слова данных из сдвигового регистра в буфер приема.

В режиме непрерывной передачи данных передача начинается и завершается таким же образом, что и при одиночной передаче. Тем не менее, линия SS постоянно находится в низком уровне, и передача данных осуществляется непрерывно. Управляющий байт следующего сообщения следует непосредственно после младшего значащего бита полученных данных из текущего сообщения. Каждое из полученных значений передается из сдвигового регистра на задний фронт сигнала SCK после поступления младшего значащего бита сообщения.

На рисунке 1.79 представлена временная диаграмма непрерывного обмена в режиме Microwire, предлагаемом фирмой National Semiconductor.

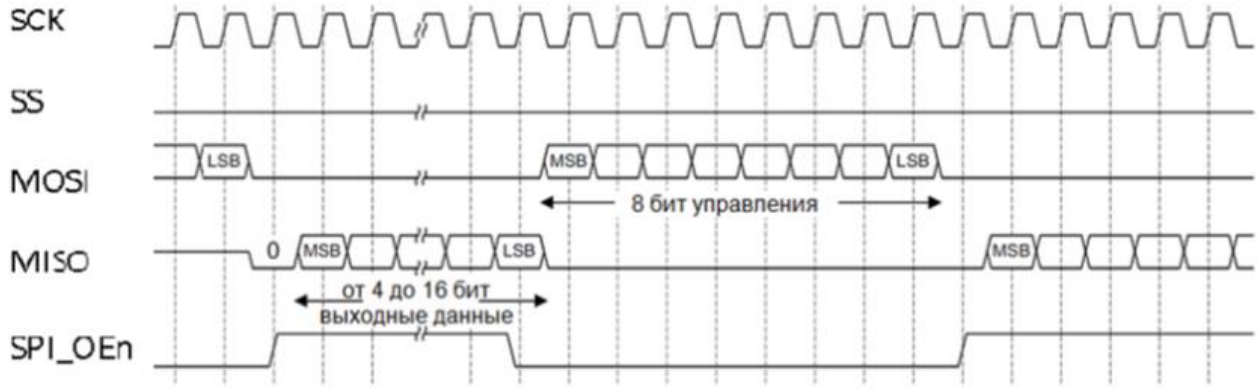


Рисунок 1.79 – Формат обмена по протоколу Microwire фирмы National Semiconductor (непрерывный обмен)

1.3.9.6.4.9 Прерывания

Контроллер SPI генерирует четыре запроса на прерывание:

– SPI_RXINTR запрос на обслуживание буфера приема;

– SPI_TXINTR запрос на обслуживание буфера передачи;

– SPI_RORINTR переполнение буфера приема;

– SPI_RTINTR превышение времени ожидания.

Каждый из четырех сигналов запроса на прерывание может быть маскирован путем установки соответствующего бита в регистре маски SPI_IMSC. Установка соответствующего бита маски в высокий уровень разрешает прерывание.

Признаки возникновения каждого из условий прерывания можно считать из регистров SPI_RIS и SPI_MIS.

Прерывание по заполнению буфера приема SPI_RXINTR формируется в случае, если буфер приемника содержит четыре или более слов данных.

Прерывание по заполнению буфера передачи SPI_TXINTR формируется в случае, если буфер передатчика содержит четыре или менее слов данных. Состояние прерывания SPI_TXINTR не зависит от поля SSE регистра SPI_CR1, что позволяет организовать взаимодействие программного обеспечения с передатчиком одним из двух способов.

– запись данных в буфер передачи до включения контроллера SPI и разрешения прерываний;

– предварительное включение контроллера SPI и прерываний и заполнение буфера передатчика FIFO в ходе проведения процедуры обслуживания прерываний.

Прерывание по переполнению буфера приема SPI_RORINTR формируется в случае, если буфер уже заполнен и осуществлена попытка записать в него дополнительные данные. При этом данные перезаписываются в регистре сдвига приемника, но в буфер приема не заносятся.

Прерывание по времени ожидания буфера приема SPI_RTINTR возникает в случае, если буфер приема не пуст и контроллер SPI находится в неактивном состоянии в течение 32 периодов. Данный механизм обеспечивает осведомленность пользователя о наличии в буфере приема необработанных данных. Прерывание по времени ожидания снимается либо после считывания данных из буфера приема до его опустошения, либо после приема новых данных по входной линии RX. Кроме того, данное прерывание может быть снято путем внесения значения в бит RTIC регистра SPI_ICR.

1.3.9.6.5 Карта регистров контролера SPI

Карта регистров контроллера SPI представлена в таблице 1.290. Доступ к резервным или неиспользуемым адресам не допускается. Попытка доступа к данным адресам может привести к непредсказуемому поведению системы.

Если не указано особо в сопровождающем тексте:

– не допускается модификация неопределенных разрядов регистра;

– при чтении зарезервированные разряды регистра необходимо игнорировать.

Таблица 1.290 – Карта регистров контроллера SPI

Адрес	Название	Доступ	Значение после сброса	Описание
0x00	SPI_CR0	RW	0x0	Регистр управления 0
0x04	SPI_CR1	RW	0x0	Регистр управления 1
0x08	SPI_DR	RW	0xX	Регистр данных
0x0C	SPI_SR	RO	0x3	Регистр состояния
0x10	SPI_CPSR	RW	0x0	Регистр предварительного делителя частоты
0x14	SPI_IMSC	RW	0x0	Регистр установки и сброса маски прерывания
0x18	SPI_RIS	RO	0x8	Регистр состояния прерываний без учета маскирования
0x1C	SPI_MIS	RO	0x0	Регистр состояния маскированных прерываний
0x20	SPI_ICR	WO	0x0	Регистр сброса прерываний
0x24 - 0xFDF	-	-	-	Резерв

1.3.9.6.6 Описание регистров контролера SPI

1.3.9.6.6.1 SPI_CR0 (0x00)

Регистр управления SPI_CR0 предназначенн для управления контроллером SPI. Поля регистра SPI_CR0 представлены в таблице 1.291.

Таблица 1.291 – Поля регистра SPI_CR0

Поле	Биты	Описание поля
DSS	3:0	Выбор размера слова данных: 0000 Резерв, неопределенный режим функционирования 0001 Резерв, неопределенный режим функционирования 0010 Резерв, неопределенный режим функционирования 0011 4 бита 0100 5 бит 0101 6 бит 0110 7 бит 0111 8 бит 1000 9 бит 1001 10 бит 1010 11 бит 1011 12 бит 1100 13 бит 1101 14 бит 1110 15 бит 1111 16 бит
FRF	5:4	Формат сообщения: 00 Формат обмена данными SPI фирмы Motorola 01 Формат обмена данными SSI фирмы Texas Instruments 10 Формат обмена данными Microwire фирмы National Semiconductor 11 Резерв, неопределенный режим функционирования
SPO	6	Полярность сигнала SCK (используется только в режиме обмена SPI фирмы Motorola)
SPH	7	Фаза сигнала SCK (используется только в режиме обмена SPI фирмы Motorola)
SCR	15:8	Скорость последовательного обмена. Значение поля SCR используется при задании скорости передачи и приема. Скорость передачи данных удовлетворяет соотношению: $F_{SCK} = \frac{F_{SPI_CLK}}{CPSDSVR*(1+SCR)}$, где CPSDVSR – четное число в диапазоне от 2 до 254, запрограммированное через регистр SPI_CPSR, и SCR – число от 0 до 255
-	31:16	Резерв

1.3.9.6.6.2 SPI_CR1 (0x04)

Регистр управления SPI_CR1 предназначен для управления контроллером SPI. Описание полей регистра SPI_CR1 представлено в таблице 1.292.

Таблица 1.292 – Поля регистра SPI_CR1

Поле	Биты	Описание поля
LBM	0	Отладочный режим петли (Loop back): 0 Нормальный режим работы 1 Выход регистра сдвига передатчика соединен с входом сдвигового регистра приемника
SSE	1	0 контроллера SPI выключен 1 контроллера SPI включен
MS	2	Выбор режима работы: 0 контроллера SPI ведущий (master); 1 контроллера SPI ведомый (slave)
SOD	3	Настройка отключения TX в режиме ведомого. Этот бит имеет значение только в режиме ведомого, MS=1. Настройка может использоваться в системах с несколькими ведомыми (multiple-slave) В системах с несколькими подчиненными устройствами ведущий может передавать сообщение всем подчиненным устройствам в системе, гарантируя при этом, что только одно подчиненное устройство передает данные на свою линию последовательного вывода. 0 контроллера SPI может управлять выходом TX (MISO) в режиме ведомого; 1 контроллера SPI неуправляет выходом TX (MISO) в режиме ведомого
-	31:4	Резерв

1.3.9.6.6.3 SPI_DR (0x08)

При чтении регистра SPI_DR обеспечивается доступ к данным буфера FIFO приемника.

При осуществлении записи в регистр SPI_DR производится запись данных в буфер передачи. Значения данных вычитываются из буфера передачи блоком передатчика по одному. Они загружаются в сдвиговый регистр передатчика, откуда последовательно выдаются на линию TX с заданной скоростью информационного обмена.

В случае если выбран размер информационного слова менее 16 бит, перед записью в буфер необходимо обеспечить выравнивание данных по границе поля DATA. Блок передачи игнорирует неиспользуемые биты. Принятые информационные слова размером менее 16 бит автоматически выравниваются по правой границе в буфере приемника. Описание полей регистра SPI_DR представлено в таблице 1.293.

Таблица 1.293 – Поля регистра SPI_DR

Поле	Биты	Описание поля
DATA	15:0	Доступ к данным: При чтении - буфер приема При записи - буфер передачи В случае если контроллер SPI запрограммирован на размер информационного слова менее 16 бит, перед записью необходимо обеспечить выравнивание данных по правой границе. Блок передачи игнорирует неиспользуемые биты. Принятые информационные слова автоматически выравниваются по правой границе в блоке приемника
-	31:16	Резерв

1.3.9.6.6.4 SPI_SR (0x0C)

Регистр состояния SPI_SR доступен только для чтения и содержит информацию о состоянии буферов приема и передачи и занятости контроллера SPI. Описание полей регистра SPI_SR представлено в таблице 1.294.

Таблица 1.294 – Поля регистра SPI_SR

Поле	Биты	Описание поля
TFE	0	Буфер передачи пуст, только для чтения: 0 Буфер передачи не пуст 1 Буфер передачи пуст
TNF	1	Буфер передачи не заполнен, только для чтения: 0 Буфер передачи заполнен 1 Буфер передачи не заполнен
RNE	2	Буфер приема не пуст, только для чтения: 0 Буфер приема пуст 1 Буфер приема не пуст
RFF	3	Буфер приема заполнен, только для чтения: 0 Буфер приема не заполнен 1 Буфер приема заполнен
BSY	4	Флаг занятости контроллера SPI, только для чтения: 0 Модуль контроллер SPI неактивен 1 Модуль контроллер SPI в настоящее время передает и/или принимает данные, либо в буфере передачи находятся данные
-	31:5	Резерв

1.3.9.6.6.5 SPI_CPSR (0x10)

Регистр SPI_CPSR является регистром предделителя тактовой частоты и используется для задания коэффициента деления, на который должен делиться входной сигнал SPI_CLK перед дальнейшим использованием.

Программируемое в данном регистре значение должно быть целым числом в диапазоне от 2 до 254. Младший значащий разряд регистра принудительно устанавливается в ноль. Если записать в регистр SPI_CPSR нечетное число, его последующее чтение даст результатом это число, но с установленным в ноль младшим битом. Описание полей регистра SPI_CPSR представлено в таблице 1.295.

Таблица 1.295 – Поля регистра SPI_CPSR

Поле	Биты	Описание поля
CPSDVSR	7:0	Коэффициент деления тактовой частоты. Записываемое значение должно быть целым числом в диапазоне от 2 до 254, в зависимости от частоты сигнала SPI_CLK. Младший значащий разряд регистра принудительно устанавливается в ноль
-	31:8	Резерв

1.3.9.6.6.6 SPI_IMSC (0x14)

Регистр SPI_IMSC является регистром установки и сброса маски прерывания. Данный регистр доступен для чтения и записи.

При чтении выдается текущее значение маски. При этом запись «1» в разряд разрешает соответствующее прерывание. Запись «0» производит сброс соответствующей маски.

После сброса все биты регистра маски устанавливаются в нулевое состояние. Поля регистра SPI_IMSC представлены в таблице 1.296.

Таблица 1.296 – Поля регистра SPI_IMSC

Поле	Биты	Описание поля
RORIM	0	Маска прерывания по переполнению буфера приемника: 0 – Маскирование при переполнении буфера FIFO приемника. 1 – При переполнении буфера приема маскирование не производится
RTIM	1	Маска прерывания по времени ожидания буфера FIFO приемника: 0 – Буфер приема не пуст, и не было попыток его чтения в течение времени ожидания с маскированием 1 – Буфер приема не пуст, и не было попыток его чтения в течение времени ожидания без маскирования
RXIM	2	Маска прерывания по заполнению буфера FIFO приемника: 0 – При заполнении на 50 % и менее буфера приема производится маскирование. 1 – При заполнении на 50 % и менее буфера приема маскирование не производится
TXIM	3	Маска прерывания по заполнению буфера FIFO передатчика: 0 – При заполнении на 50 % и менее буфера передачи производится маскирование. 1 – При заполнении на 50 % и менее буфера передачи маскирование отсутствует.

Поле	Биты	Описание поля
-	31:4	Резерв

1.3.9.6.6.7 SPI_RIS (0x18)

Регистр SPI_RIS содержит текущее состояние прерываний без учета маскирования. Данный регистр доступен только для чтения.

При чтении данного регистра выдается текущее исходное значение состояния соответствующего прерывания до события маскирования. Данные, записываемые в регистр, игнорируются. Поля регистра SPI_RIS представлены в таблице 1.297.

Таблица 1.297 – Поля регистра SPI_RIS

Поле	Биты	Описание поля
RORRIS	0	Состояние до маскирования прерывания SPI_RORINTR
RTRIS	1	Состояние до маскирования прерывания SPI_RTINTR
RXRIS	2	Состояние до маскирования прерывания SPI_RXINTR
TXRIS	3	Состояние до маскирования прерывания SPI_TXINTR
-	31:4	Резерв

1.3.9.6.6.8 SPI_MIS (0x1C)

Регистр SPI_MIS содержит текущее состояние прерываний с учетом маскирования. Данный регистр доступен только для чтения. При чтении данного регистра выдается текущее значение маскированного состояния соответствующего прерывания. Данные, записываемые в регистр, игнорируются. Поля регистра SPI_MIS представлены в таблице 1.298.

Таблица 1.298 – Поля регистра SPI_MIS

Поле	Биты	Описание поля
RORMIS	0	Состояние после маскирования прерывания SPI_RORINTR
RTMIS	1	Состояние после маскирования прерывания SPI_RTINTR
RXMIS	2	Состояние после маскирования прерывания SPI_RXINTR
TXMIS	3	Состояние после маскирования прерывания SPI_TXINTR
-	31:4	Резерв

1.3.9.6.6.9 SPI_ICR (0x20)

Регистр SPI_ICR является регистром сброса прерываний и доступен только для записи. Сброс признака прерывания по заданному событию производится путем записи «1» в соответствующий бит. Запись в любой из разрядов регистра «0» игнорируется. Поля регистра SPI_ICR представлены в таблице 1.298.

Таблица 1.299 – Поля регистра SPI_ICR

Поле	Биты	Описание поля
RORIC	0	Сброс прерывания SPI_RORINTR
RTIC	1	Сброс прерывания SPI_RTINTR
-	31:2	Резерв

1.3.9.7 Сборка контроллеров низкоскоростных интерфейсов (LSIF_PORT)

МК Optimal+ обладает обширным набором низкоскоростных интерфейсов:

– 8 контроллеров USART

– 4 контроллера SPI;

– 4 контроллера I2C;

– 4 контроллера CAN;

– 8 блоков TIM.

Перечисленные выше контроллеры сгруппированы в сборки контроллеров низкоскоростных интерфейсов (LSIF_PORT), построенные на базе контроллера GPIO. Каждая сборка LSIF_PORT включает:

– 1 контроллер GPIO (со встроенным блоком IFMUX);

– 2 контроллера USART;

– 1 контроллер SPI;

– 1 контроллер I2C;

– 1 контроллер CAN;

– 2 блока TIM.

Контроллеры низкоскоростных интерфейсов, входящие в состав сборки LSIF_PORT, подключаются к блоку IFMUX, входящему в состав контроллера GPIO.

Блок IFMUX реализует мультиплексирование внешних выводов контроллеров низкоскоростных интерфейсов на линии контроллера GPIO в качестве альтернативных функций по принципу «любой на любой», т.е. любой внешний вывод любого из перечисленных выше контроллеров низкоскоростных интерфейсов может быть подключен на любой вывод GPIO как альтернативная функция через блок IFMUX.

1.3.9.7.1 Подключение контроллеров низкоскоростных интерфесов к IFMUX

Подключение внешних выводов контроллеров низкоскоростных интерфесов к внутренним линиям блока IFMUX фиксировано (приведено в таблице 1.300). Выбор какой именно внешний вывод какого контроллера низкосокоростного интерфейса будет выведен на линию

контроллера GPIO задаётся в регистре GPIO_i_CFG - в поле ifmux_sel необходимо указать номер внутренней линии блока IFMUX.

Таблица 1.300 – Подключение контроллеров к IFMUX

№ внутренней лини IFMUX	Интерфейс	Название сигнала	Направление	Использование
0	USART0	RXD	Вх	Всегда
1		TXD	Вых	Всегда
2		RTS/OEN	Вых	Только в некоторых протоколах
3		CTS	Вх	Только в некоторых протоколах
4		XCK	Вх/Вых	Только в USART (в UART не используется)
5	USART1	RXD	Вх	Всегда
6		TXD	Вых	Всегда
7		RTS/OEN	Вых	Только в некоторых протоколах
8		CTS	Вх	Только в некоторых протоколах
9		XCK	Вх/Вых	Только в USART (в UART не используется)
10	I2C	SDA	Вх/Вых	Всегда
11		SCL	Вх/Вых	Всегда
12	CAN	CRX	Вх	Всегда
13		CTX	Вых	Всегда
14	TIM0	CH0	Вх/Вых	Опционально
15		CH1	Вх/Вых	Опционально
16		CH2	Вх/Вых	Опционально
17		CH3	Вх/Вых	Опционально
18		ENT	Вх	Опционально
19		BRK	Вх	Опционально
20	TIM1	CH0	Вх/Вых	Опционально
21		CH1	Вх/Вых	Опционально
22		CH2	Вх/Вых	Опционально
23		CH3	Вх/Вых	Опционально
24		ENT	Вх	Опционально
25		BRK	Вх	Опционально
26	SPI	RX	Вх	Всегда
27		TX	Вых	Всегда
28		SCK	Вх/Вых	Всегда
29		SS	Вх/Вых	Всегда

1.3.9.7.2 Описание внешних выводов блка IFMUX

Каждая линия блка IFMUX соответствует линии контроллера GPIO, поэтому для именования внешних выводов блка IFMUX в составе МК Optimal+ принята схема, аналогичная именованию внешних выводов контроллера GPIO: IFMUX_<_имя_порта_>_<_номер_линии_>.

Так как порты в составе МК Optimal+ используют не все линии контроллера GPIO, точный список используемых внешних выводов для каждого блка IFMUX см. в разделе 1.2.2.

Под функцией вывода IFMUX_x в разделе 1.2.2 понимается любой из внешних выводов контроллеров низкоскоростных интерфейсов. Более подробное описание внешних выводов контроллеров низкоскоростных интерфейсов представлено в соответствующих пунктах:

– интерфейс USART - 1.3.9.3.3;

– интерфейс SPI - 1.3.9.6.3;

– интерфейс I2C - 1.3.9.4.3;

– интерфейс CAN - 1.3.9.5.3;

– интерфейс TIM - 1.3.13.4.2.

1.3.9.8 Контроллер интерфейса LCD (LCD)

1.3.9.8.1 Основные характеристики контроллера LCD

LCD контроллер обладает следующими характеристиками:

– до 44 линий SEG;

– до 8 линий COM;

– шаги смещения напряжения (BIAS): 1, 1/2, 1/3, 1/4;

– теневой буфер для нового кадра;

– размер одного буфера 44x8 бит;

– использование внешнего тактового сигнала;

– интерфейс программирования и передачи данных: AHB slave.

1.3.9.8.2 Описание LCD контроллера

На рис. 1.80 представлена структурная схема контроллера. Основным его назначением является управление символьным LCD дисплеем.

Далее в документации для сигналов и регистров, у которых есть номера в названии, вместо любой доступной цифры может использоваться латинская буква "x" противоположенного регистра. Например, перечисление выходов SEG0 - SEG43 может быть заменено на SEGx.

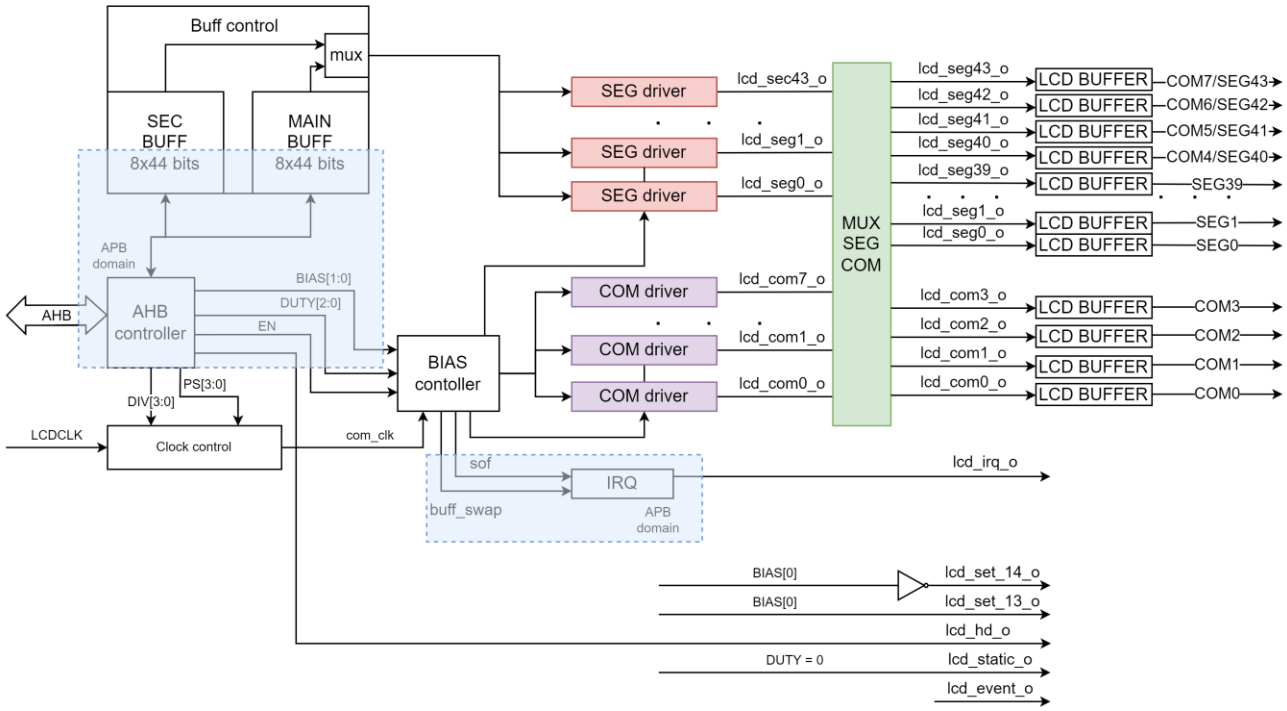


Рисунок 1.80 – Структура LCD контроллера

В таб. 1.301 представлено описание блоков LCD контроллера.

Таблица 1.301 – Описание назначения составных блоков LCD контроллера

Название блока	**Назначение и основные функции**
AHB controller	Блок служит для чтения регистров состояний, чтения и записи управляющих регистров, записи в MAIN и SEC BUFF.
Buff control	Блок служит для смены между MAIN и SEC BUFF буферами кадров и их выдачи на дисплей.
Clock control	В соответствии с установленными делителями, генерирует сигнал развертки.
BIAS controller	Контролирует начало и конец кадров и полукадров, а так же задержку между переключением активного буфера. Генерирует выходной сигнал на COMx выходы.
SEG (COM) driver	Выдает на выходе соответствующий уровень напряжения в закодированном виде.
IRQ	Конкатенирует все события от LCD контроллера и выдает их через выход lcd_irq_o

1.3.9.8.3 Описание внешних выводов контроллера LCD

Список внешних выводов контроллера LCD и их описание приведены в таблице 1.302

Таблица 1.302 – Список внешних выводов контроллера LCD

Наименование вывода	Тип (Вх/Вых)	Функциональное назначение
VDD_LCD	Вх	Опорное напряжение для lcd_bias.
COM_x[1]	Вых	Задает выходное напряжение на COMx см. 1.3.9.8.5.
SEG_x[2]	Вых	Задает выходное напряжение на SEGx см. 1.3.9.8.5.
COM_x/SEG_x[3]	Вых	Задает выходное напряжение на мультиплексированных выходах см. 1.3.9.8.5.
[1)] x — это число от 0 до 3; [2)] x — это число от 0 до 39; [3)] x — это число от 4 до 7 для COM и от 40 до 43.		

1.3.9.8.4 Физические принципы работы LCD дисплея

Изображение в LCD дисплее формируется из сегментов (пикселей). Каждый из которых может находится в двух состояниях: активном или неактивном. В зависимости от типа дисплея, активный сегмент может светится (позитивный тип) или наоборот гаснуть (негативный тип). Сегмент управляется разностью напряжения между SEGx и COMx. Для его активации требуется, чтобы среднеквадратичная разность напряжений была больше порогового значения. Пороговое значение определяется производителем дисплея.

Важной особенностью таких дисплеев является то, что под действием постоянного напряжения материал анода начинает "перетекать" на катод (электромиграция). Для предотвращения этого эффекта необходимо, чтобы постоянная составляющая напряжения между COM и SEG была равна нулю. Это достигается за счет того, что разность напряжений между катодом и анодом поочерёдно устанавливают то положительной, то отрицательной. Для этого вывод одного кадра разбит на два полукадра: positive и negative.

Изображение формируется «построчно». Каждой строке соответствует свой выход COM, а каждому элементу строки — выход SEG. В течение одного полукадра поочерёдно активируются все выходы COM. За счёт инертности жидкого кристалла изображение не мигает. Время, в течение которого активен только один COM называется фазой. Переключение между активными выходами COMx формирует развёртку изображения. Изменить порядок активизации COMx выходов нельзя. Кадр всегда начинается с активного COM0 и идет последовательно.

На рис. 1.81 представлена диаграмма кадра, включающего 4 выхода COM.

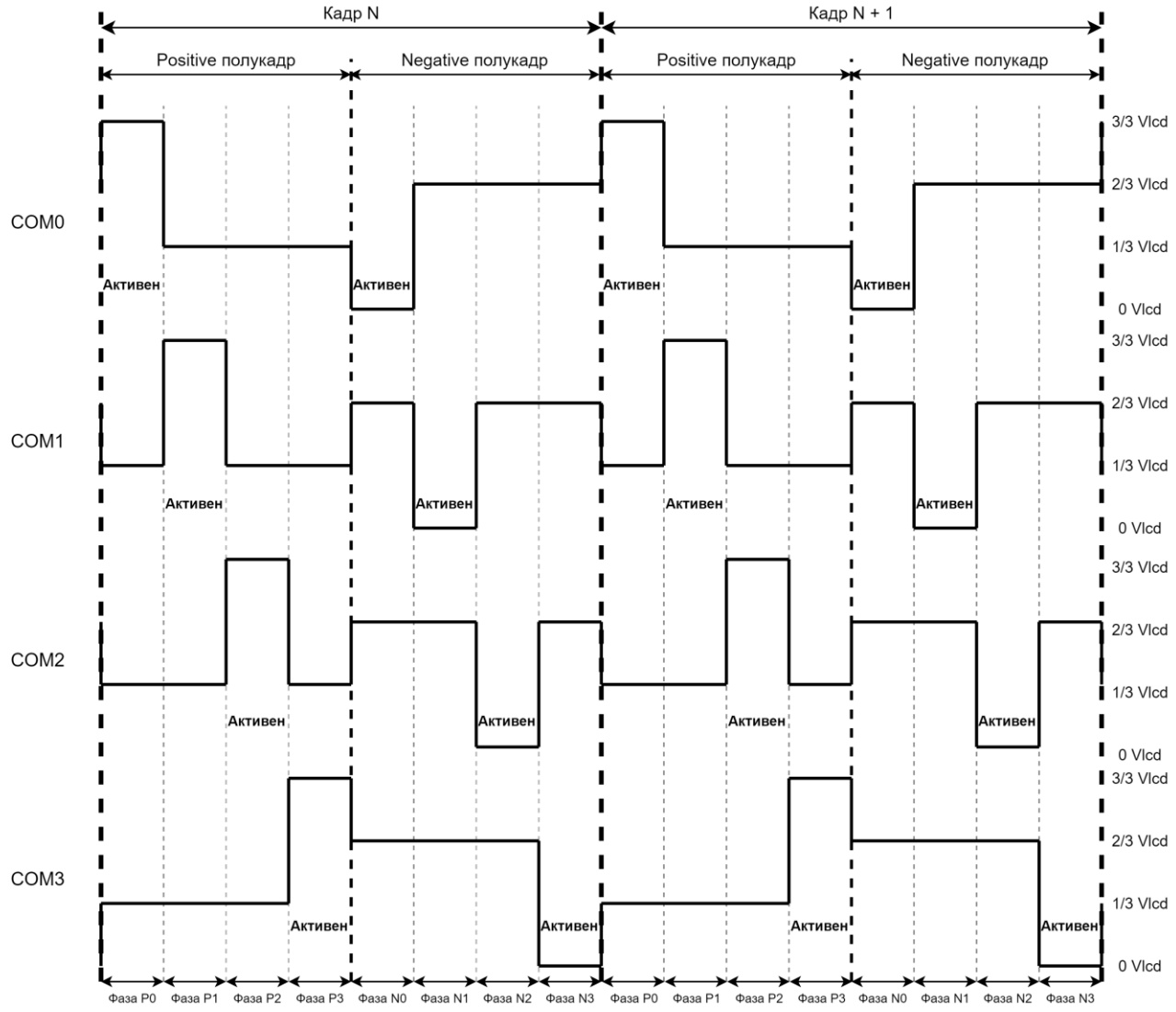


Рисунок 1.81 – Пример сигналов развертки для BIAS 1/3 и DUTY 4

Частота развертки (время одной фазы) задается с помощью делителя тактового сигнала Clock control. Он делит частоту входного тактового сигнала lcd_clk по следующей формуле:

$f_{com_clk} = \frac{f_{lcd_clk}}{2^{PS}(16+DIV)}$.

PS и DIV это коэффициенты делителя, указанные в регистре LCD_CTRL.

Частота обновления кадров на дисплее определяется следующей формулой:

$f_{frame_clk} = \frac{f_{com_clk}}{duty \times 2}$, где duty - количество используемых COM выходов.

В качестве тактового сигнала может выступать или LSE, или LSI.

1.3.9.8.5 Драйверы выходов

Для создания дробных уровней напряжения используются lcd_bias блоки, которые представляют из себя резистивные делители (см. рис. 1.82). Далее эти уровни напряжения поступают в lcd_buffer, где LCD контроллер с помощью выходов lcd_segX_o или lcd_comX_o выбирает необходимое выходное напряжение на SEGx или COMx.

В резистивном делители есть возможность задавать два набора сопротивлений: низкое (R_L) и высокое (R_H). Это определяется с помощью бит LCD_CTRL[HD]. Когда он установлен, то ключи замкнуты, что означает низкое сопротивление. Данную настройку не рекомендуется изменять во время выдачи очередного кадра, потому как это может привести к тому, что среднее напряжение между SEGx и COMx станет отличным от нуля.

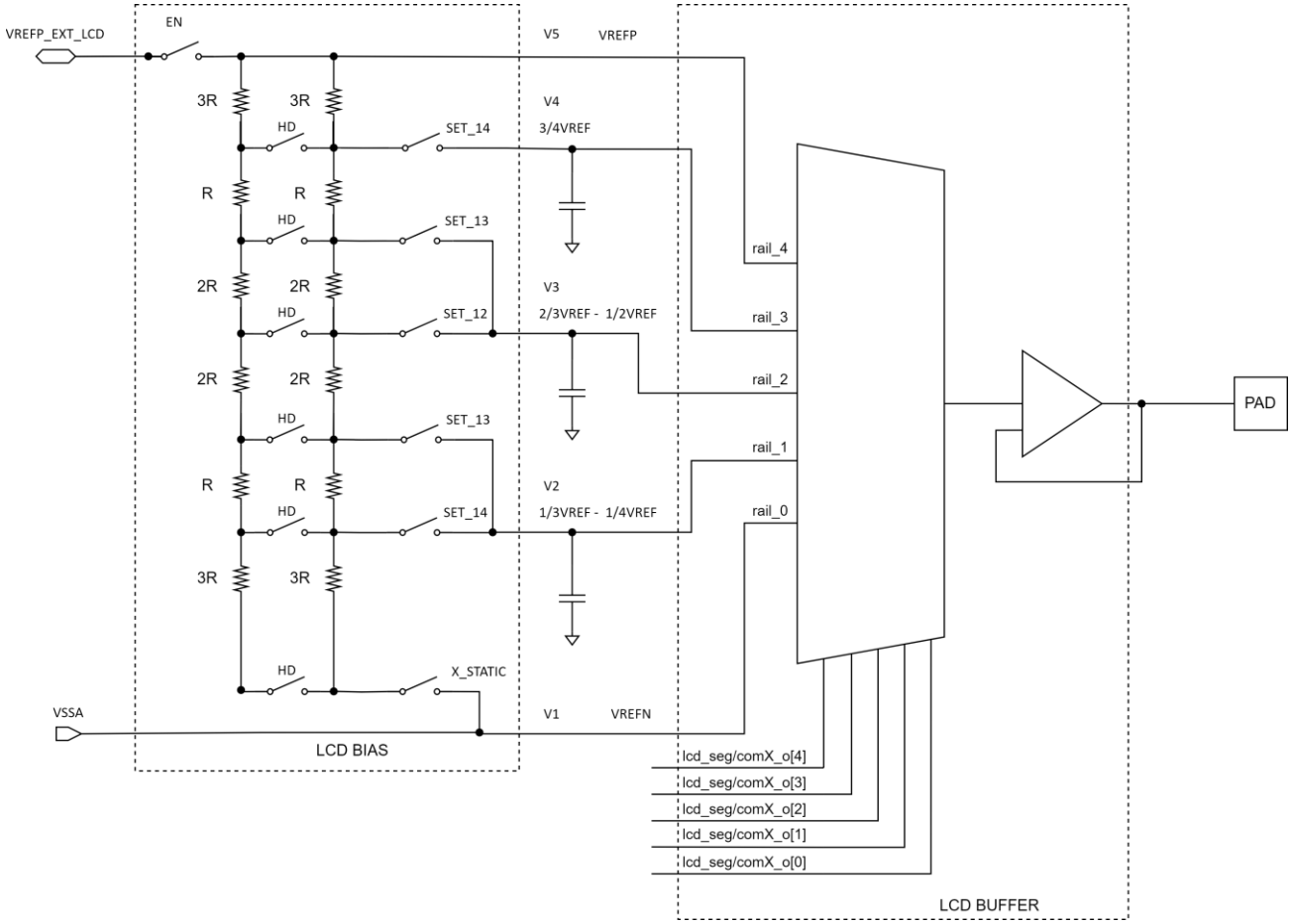


Рисунок 1.82 – Драйвер SEGx и COMx выходов

1.3.9.8.6 Принципы генерирования выходного напряжения на COMx и SEGx выходах

В течение positive полукадра COM driver задает на выходе COMx напряжение V_{lcd}, если сейчас активен текущий COMx выход. Если же он не активен, то напряжение будет зависеть

от BIAS (см. таблицу 1.303). В течение negative полукадра на выходе COMx будет напряжение V_{ss}, если текущий COMx выход сейчас активен. Если же он не активен, то напряжение будет зависеть от BIAS (см. таблицу 1.303).

Если какой-то из COMx выходов не используется, то на нем будет установлено V_{ss} напряжение.

Когда используется только COM0, то этот выход всегда активен и BIAS равен 1, а битовое поле LCD_CTRL[BIAS] ни на что не влияет.

Таблица 1.303 – Напряжение на COMx выходе

Полукадр	**Активность**	**BIAS**	**Напряжение**
positive	+	x	V_{lcd}
positive	-	1/2	1/2 V_{lcd}
positive	-	1/3	1/3 V_{lcd}
positive	-	1/4	1/4 V_{lcd}
negative	+	x	V_{ss}
negative	-	1/2	1/2 V_{lcd}
negative	-	1/3	2/3 V_{lcd}
negative	-	1/4	3/4 V_{lcd}

Выходное напряжение определяется на основе значений в рабочем буфере сегментов SEGx для текущего активного COMx (см. таблицу 1.304).

Уровни напряжения, указанные в таб. 1.304 генерируются всегда, если параметр CFG_OUT_LEVELS = 0, а если он равен единице, то для генерирования соответствующих уровней необходимо записать в регистр LCD_OUT_LEVELS определенное значение (см. пп 1.3.9.8.11).

Таблица 1.304 – Напряжение на SEGx выходе

Полукадр	**Активность SEGx**	**BIAS**	**Напряжение**
positive	+	x	V_{ss}
positive	-	1	V_{lcd}
positive	-	1/2	V_{lcd}
positive	-	1/3	2/3 V_{lcd}
positive	-	1/4	1/2 V_{lcd}
negative	+	x	V_{lcd}
negative	-	1	V_{ss}
negative	-	1/2	V_{ss}
negative	-	1/3	1/3 V_{lcd}
negative	-	1/4	1/2 V_{lcd}

На рис. 1.83 приведен пример выходного напряжения между COM0 и SEG0, когда данный сегмент активен. Так же из данного рисунка видно, что сегмент подключенный между COM2 и SEG0 тоже активен.

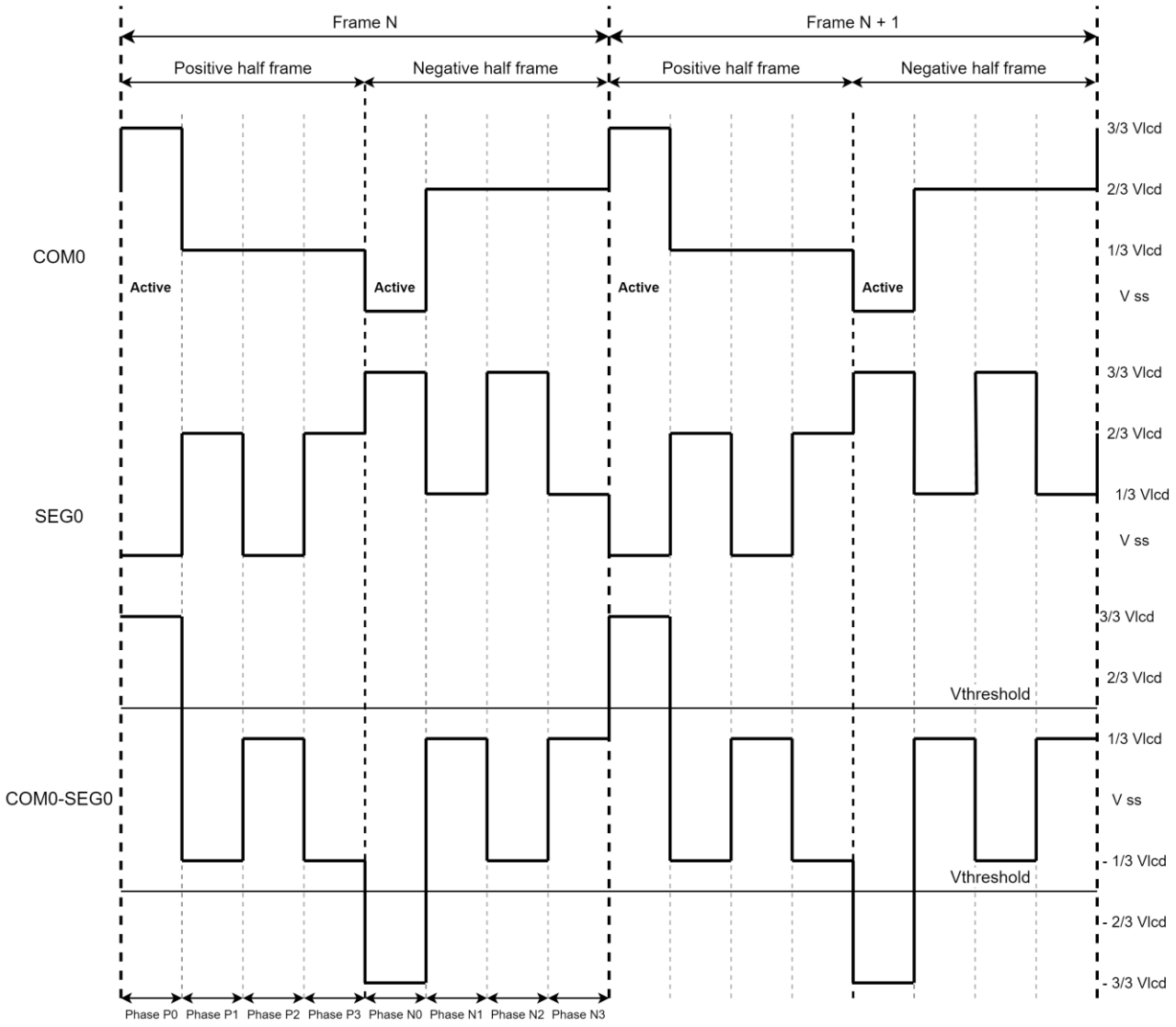


Рисунок 1.83 – Выходное напряжение между COM0 и SEG0

Когда используется только COM0 (LCD_CTRL[DUTY] = 000), то BIAS равно 1 независимо от поля LCD_CTRL[BIAS].

1.3.9.8.7 Расчет выходного напряжения

У каждого дисплея есть пороговое напряжение включения $V_{th(on)}$ и выключения $V_{th(off)}$. Для того, чтобы дисплей корректно работал необходимо выполнение следующих условий:

$V_{on(RMS)} \geq V_{th(on)}$, где $V_{on(RMS)}$ - среднеквадратичное напряжение на сегменте, когда он включен.

$V_{off(RMS)} \leq V_{th(off)}$, где $V_{off(RMS)}$ - среднеквадратичное напряжение на сегменте, когда он выключен.

Для расчета среднеквадратичного напряжения используются следующие формулы:

$$V_{on(RMS)} = Vlcd\sqrt{\frac{b^2(d-1)+1}{d}},$$

$$V_{off(RMS)} = Vlcd\sqrt{\frac{b^2(d-1)+(1-2b)^2}{d}},$$

где b - BIAS (1, 1/2, 1/3, 1/4);
d - DUTY (1, 2, 3, 4, 5, 6, 7, 8).

1.3.9.8.8 Буферы кадров

Всего LCD контроллер может управлять дисплеями, содержащими до $8 \times 40 = 320$ сегментов (пикселей). Для хранения кадра используются два буфера MAIN BUFF и SEC BUFF, каждый из которых содержит по одному кадру. В процессе работы изображение на дисплей циклически выводится из одного буфера. Этот буфер называется рабочим. Другой в это время будет теневым. Существует два режима переключения рабочего буфера:

– MANUAL. Переключение рабочего буфера между MAIN и SEC происходит программно. Для этого необходимо установить бит LCD_BUFF_STATUS[SWAP_BUFF_REQ]. Переключение произойдет после окончания текущего кадра и бит LCD_BUFF_STATUS[SWAP_BUFF_REQ] сбросится.

– BLINK. Переключение рабочего буфера между MAIN BUFF и SEC BUFF происходит автоматически после заданного количества кадров, указанного в регистре LCD_BLINK.

Смена рабочего буфера всегда происходит в конце кадра, т.е. во время последней фазы negative полукадра. Поэтому после сброса первый кадр всегда выводится из MAIN буфера, даже если установлен запрос на смену буфера.

Режим буферов выбирается до включения контроллера и во время работы не должен изменяться. Запись через регистры LCD_SEC_COMx_A и LCD_SEC_COMx_B или LCD_MAIN_COMx_A и LCD_MAIN_COMx_B в рабочий буфер **игнорируется**.

После выключения контроллера рабочим буфером остается, тот который был рабочим в момент выключения. Однако если в момент выключения был установлен флаг запроса на смену рабочего буфера или если выключение совпало со сменой буфера в BLINK режиме, то

рабочий буфер изменится в момент выключения. Поэтому для более надежной работы рекомендуется проверять какой из буферов является рабочим и про изводить запись в другой.

Программный и аппаратный сброс устанавливают MAIN буфер рабочим всегда.

На рис. 1.84 изображен MAIN буфер кадров. Значения сегментов (активен/неактивен) задаются в этом буфере (наборе регистров). Пустые поля являются резервными. SEC буфер имеет такую же структуру, только у него базовый адрес начинается с 0x200.

Адрес	Название регистра	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x100	LCD_MAIN_COM0_A	SEG31	SEG30	SEG29	SEG28	SEG27	SEG26	SEG25	SEG24	SEG23	SEG22	SEG21	SEG20	SEG19	SEG18	SEG17	SEG16	SEG15	SEG14	SEG13	SEG12	SEG11	SEG10	SEG9	SEG8	SEG7	SEG6	SEG5	SEG4	SEG3	SEG2	SEG1	SEG0
0x104	LCD_MAIN_COM0_B																					SEG43	SEG42	SEG41	SEG40	SEG39	SEG38	SEG37	SEG36	SEG35	SEG34	SEG33	SEG32
0x108	LCD_MAIN_COM1_A	SEG31	SEG30	SEG29	SEG28	SEG27	SEG26	SEG25	SEG24	SEG23	SEG22	SEG21	SEG20	SEG19	SEG18	SEG17	SEG16	SEG15	SEG14	SEG13	SEG12	SEG11	SEG10	SEG9	SEG8	SEG7	SEG6	SEG5	SEG4	SEG3	SEG2	SEG1	SEG0
0x10C	LCD_MAIN_COM1_B																					SEG43	SEG42	SEG41	SEG40	SEG39	SEG38	SEG37	SEG36	SEG35	SEG34	SEG33	SEG32
0x110	LCD_MAIN_COM2_A	SEG31	SEG30	SEG29	SEG28	SEG27	SEG26	SEG25	SEG24	SEG23	SEG22	SEG21	SEG20	SEG19	SEG18	SEG17	SEG16	SEG15	SEG14	SEG13	SEG12	SEG11	SEG10	SEG9	SEG8	SEG7	SEG6	SEG5	SEG4	SEG3	SEG2	SEG1	SEG0
0x114	LCD_MAIN_COM2_B																					SEG43	SEG42	SEG41	SEG40	SEG39	SEG38	SEG37	SEG36	SEG35	SEG34	SEG33	SEG32
0x118	LCD_MAIN_COM3_A	SEG31	SEG30	SEG29	SEG28	SEG27	SEG26	SEG25	SEG24	SEG23	SEG22	SEG21	SEG20	SEG19	SEG18	SEG17	SEG16	SEG15	SEG14	SEG13	SEG12	SEG11	SEG10	SEG9	SEG8	SEG7	SEG6	SEG5	SEG4	SEG3	SEG2	SEG1	SEG0
0x11C	LCD_MAIN_COM3_B																					SEG43	SEG42	SEG41	SEG40	SEG39	SEG38	SEG37	SEG36	SEG35	SEG34	SEG33	SEG32
0x120	LCD_MAIN_COM4_A	SEG31	SEG30	SEG29	SEG28	SEG27	SEG26	SEG25	SEG24	SEG23	SEG22	SEG21	SEG20	SEG19	SEG18	SEG17	SEG16	SEG15	SEG14	SEG13	SEG12	SEG11	SEG10	SEG9	SEG8	SEG7	SEG6	SEG5	SEG4	SEG3	SEG2	SEG1	SEG0
0x124	LCD_MAIN_COM4_B																					SEG43	SEG42	SEG41	SEG40	SEG39	SEG38	SEG37	SEG36	SEG35	SEG34	SEG33	SEG32
0x128	LCD_MAIN_COM5_A	SEG31	SEG30	SEG29	SEG28	SEG27	SEG26	SEG25	SEG24	SEG23	SEG22	SEG21	SEG20	SEG19	SEG18	SEG17	SEG16	SEG15	SEG14	SEG13	SEG12	SEG11	SEG10	SEG9	SEG8	SEG7	SEG6	SEG5	SEG4	SEG3	SEG2	SEG1	SEG0
0x12C	LCD_MAIN_COM5_B																					SEG43	SEG42	SEG41	SEG40	SEG39	SEG38	SEG37	SEG36	SEG35	SEG34	SEG33	SEG32
0x130	LCD_MAIN_COM6_A	SEG31	SEG30	SEG29	SEG28	SEG27	SEG26	SEG25	SEG24	SEG23	SEG22	SEG21	SEG20	SEG19	SEG18	SEG17	SEG16	SEG15	SEG14	SEG13	SEG12	SEG11	SEG10	SEG9	SEG8	SEG7	SEG6	SEG5	SEG4	SEG3	SEG2	SEG1	SEG0
0x134	LCD_MAIN_COM6_B																					SEG43	SEG42	SEG41	SEG40	SEG39	SEG38	SEG37	SEG36	SEG35	SEG34	SEG33	SEG32
0x138	LCD_MAIN_COM7_A	SEG31	SEG30	SEG29	SEG28	SEG27	SEG26	SEG25	SEG24	SEG23	SEG22	SEG21	SEG20	SEG19	SEG18	SEG17	SEG16	SEG15	SEG14	SEG13	SEG12	SEG11	SEG10	SEG9	SEG8	SEG7	SEG6	SEG5	SEG4	SEG3	SEG2	SEG1	SEG0
0x13C	LCD_MAIN_COM7_B																					SEG43	SEG42	SEG41	SEG40	SEG39	SEG38	SEG37	SEG36	SEG35	SEG34	SEG33	SEG32

Рисунок 1.84 – MAIN буфер

1.3.9.8.9 Dead time

Данная функция позволяет изменить контрастность изображения не изменяя сами кадры. Это происходит за счет вставки между всеми кадрами интервала времени (dead time), в течение которого на выходах COMx и SEGx устанавливается V_{ss}. Этот интервал задается с помощью поля LCD_CTRL[DEAD_TIME], а измеряется он в периодах развертки (фазах).

1.3.9.8.10 MUX SEG и COM

Выходы COM4 - COM7 и SEG40 - SEG43 замультиплексированы (см. рис. 1.80). Когда в поле LCD_CTRL[DUTY] установлено значение меньше чем 0x4, то все выходы SEG40-SEG43 доступны. Если же используется больше чем 4 COM выхода (DUTY >= 0x4), то дополнительные выходы COM4-COM7 будут автоматически мультиплексироваться с SEG40 - SEG43. Например, если LCD_CTRL[DUTY] = 5, то выходы SEG43/COM4 и SEG42/COM5 будут работать как COM4 и COM5, а выходы SEG41/COM6 и SEG40/COM7, как SEG41 и SEG40.

1.3.9.8.11 Перенастройка выходных уровней напряжения (тестовый режим)

LCD контроллер имеет возможность перенастройки активный выходных уровней напряжения на COMx и SEGx выходах. Для этого нужно выставить желаемые уровни в регистре LCD_OUT_LEVELS (см. таблицу 1.319) и установить поле CFG_OUT_LEVELS = 1 (включения прямого управления). Это рекомендуется выполнять до включения LCD контроллера.

lcd_segX_o и lcd_comX_o сигналы выбирают одну из линий напряжения (rail_x) в соответствии с таб. 1.305. Каждая из этих линий подключена к определенному напряжению, которое зависит от выбранного режима BIAS, то есть нулевого бита поля LCD_CTRL[BIAS].

Таблица 1.305 – Кодирование выходного напряжения.

Значения LCD_OUT_LEVELS	**Линия напряжения**	**Значение COMx/SEGx**		**Значение на выходах lcd_segX_o и lcd_comX_o**
		bias[0] = 0	bias[0] = 1	
0x0	rail_0	V_{ss}	V_{ss}	0x01
0x1	rail_1	1/4 V_{lcd}	1/3 V_{lcd}	0x02
0x2	rail_2	1/2 V_{lcd}	2/3 V_{lcd}	0x04
0x4	rail_3	3/4 V_{lcd}	3/4 V_{lcd}	0x08
0x8	rail_4	V_{lcd}	V_{lcd}	0x10

В 1.306 приведены значения, которые необходимо записать в регистр LCD_OUT_LEVELS для того, что бы LCD контроллер генерировал такие же паттерны, которые описаны в таблицах 1.304 и 1.303.

Таблица 1.306 – Значение для LCD_OUT_LEVELS регистра.

BIAS	**Значение регистра LCD_OUT_LEVELS**
1 (STATIC)	0x08080880
1/2	0x22080880
1/3	0x21081280
1/4	0x41082280

1.3.9.8.12 Рекомендации по программированию LCD контроллера

Процессы включения и выключения LCD контроллера занимает некоторое время (зависит от системной частоты и lcd_clk). Они состоят из следующих этапов:

– Запуск включения (выключения). Записать в нулевой бит регистра LCD_EN единицу (нуль).

– Ожидание включения (выключения). Считывать статусный бит регистра LCD_CTRL[EN_STATUS] до тех пор пока там не будет единица (нуль). Это означает, что контроллер включился (выключился). Так же этот процесс можно отследить с помощью прерываний.

При выключении LCD_CTRL[EN_STATUS] будет находится в единице до окончания текущего кадра. Это необходимо для того, чтобы на дисплее постоянная составляющая напряжения оставалась равной нулю.

Программный сброс происходит в произвольный момент кадра. Из-за это его рекомендуется производить только, когда контроллер выключен, что бы не допустить некорректной подачи напряжения на дисплей.

Когда LCD контроллер включен запрещено писать в следующие регистры: LCD_CTRL, LCD_BLINK, LCD_OUT_LEVELS. Поэтому до включения необходимо:

– установить режим смены буферов, а для режима BLINK еще и задать количество кадров между сменой буферов.

– записать в память буфера MAIN валидные значения кадра.

– установить необходимое значение BIAS.

– установить маску прерываний.

1.3.9.8.13 Прерывания

LCD контроллер поддерживает уровневые прерывания. Они отслеживаются с помощью регистра LCD_IRQ_STATUS. Ниже приведен их список:

– TEST. Данное прерывание устанавливается программно после записи единицы в регистр LCD_IRQ_TEST. Сброс происходит после записи единицы в бит LCD_IRQ_CLEAR[CLEAR_TEST].

– SOF (Start of frame). Данное прерывание устанавливается, когда начинается новый кадр и удерживается, пока не будет сброшено. Сброс происходит после записи единицы в LCD_IRQ_CLEAR[CLEAR_SOF]. Данное прерывание не выставляется в момент включения и выключения.

– BUFF_SWAP. Данное прерывание устанавливается, когда происходит смена рабочего буфера. Это может произойти как в режиме MANUAL, так и в BLINK. Сброс происходит после записи единицы в LCD_IRQ_CLEAR[CLEAR_BUFF_SWAP]. В момент выключения данное прерывание устанавливается только, если установлен запрос на смену буфера или пришло время смены в BLINK режиме.

– TURN_ON. Данное прерывание устанавливается, когда происходит включение LCD контроллера. Сброс происходит после записи единицы в LCD_IRQ_CLEAR[CLEAR_TURN_ON].

– TURN_OFF. Данное прерывание устанавливается, когда происходит выключение LCD контроллера. Сброс происходит после записи единицы в LCD_IRQ_CLEAR[CLEAR_TURN_OFF].

Все прерывания объединяются по "или" и по "и" объединяются с регистром маски LCD_IRQ_MASK, а затем выводятся на выход lcd_irq_o.

1.3.9.8.14 Интеграция

Контроллер разделен на два тактовых домена: ahb-домен и core-домен. Это сделано для того, что бы при изменении системной частоты core-домен, выполняющий передачу данных на шине, не надо было перенастраивать.

У контроллера на входе стоит специальный блок который корректно завершает ahb-транзакций, пока с LCD контроллера снято тактирование. При этом все транзакции записи игнорируются, а чтение возвращает 0x0.

При доступе по не выровненным адресам, младшие биты адреса не анализируются и транзакция завершается без ошибок (hresp всегда равен нулю).

Запись по зарезервированным адресам игнорируется, а чтение возвращает 0xEEFFADDC. При этом транзакции завершаются без ошибок. Байтовый доступ поддерживается.

1.3.9.8.15 Описание регистров

1.3.9.8.15.1 Общее описание регистров

Контроллер имеет набор программно доступных регистров, требующий 4 КБайт адресов на системной шине AHB.

Карта регистров приведена в таблице 1.307.

Адреса, который не указаны в таблице 1.307 считаются резервными. Чтение из них возвращает 0xEEFFADDC, а запись игнорируется. LCD контроллер не различает secure и non-secure транзакции.

1.3.9.8.15.2 Карта регистров

Таблица 1.307 – Карта регистров LCD

Адрес	Название	Доступ	Значение после сброса	Описание
0x000	LCD_ID	ЧТ	0x3044434C	ID LCD контроллера
0x004	LCD_VERSION	ЧТ	0x13371448	Версия LCD контроллера
0x008	LCD_SW_RST	ЧТ/ЗП	0x0	Программный сброс
0x00C	LCD_EN	ЧТ/ЗП	0x0	Включение LCD контроллера
0x010	LCD_CTRL	ЧТ/ЗП	0x0	Настройка контроллера
0x014	LCD_BLINK	ЧТ/ЗП	0x0	Период смены рабочего буфера
0x018	LCD_BUFF_STATUS	ЧТ/ЗП	0x0	Настройка буфера
0x01C	LCD_IRQ_STATUS	ЧТ	0x0	Статус прерываний
0x020	LCD_IRQ_MASK	ЧТ/ЗП	0x0	Маска прерываний
0x024	LCD_IRQ_CLEAR	ЗП	0x0	Сброс прерываний
0x028	LCD_IRQ_TEST	ЗП	0x0	Установка прерывания TEST
0x034	LCD_OUT_LEVELS	ЧТ/ЗП	0x08080880	Настройка активных уровней на SEGx и COMx
0x100	LCD_MAIN_COM0_A	ЗП	0x0	Значение сегментов с 31 по 0 в MAIN BUFF для COM0
0x104	LCD_MAIN_COM0_B	ЗП	0x0	Значение сегментов с 43 по 32 в MAIN BUFF для COM0
0x108	LCD_MAIN_COM1_A	ЗП	0x0	Значение сегментов с 31 по 0 в MAIN BUFF для COM1
0x10C	LCD_MAIN_COM1_B	ЗП	0x0	Значение сегментов с 43 по 32 в MAIN BUFF для COM1
0x110	LCD_MAIN_COM2_A	ЗП	0x0	Значение сегментов с 31 по 0 в MAIN BUFF для COM2
0x114	LCD_MAIN_COM2_B	ЗП	0x0	Значение сегментов с 43 по 32 в MAIN BUFF для COM2
0x118	LCD_MAIN_COM3_A	ЗП	0x0	Значение сегментов с 31 по 0 в MAIN BUFF для COM3
0x11C	LCD_MAIN_COM3_B	ЗП	0x0	Значение сегментов с 43 по 32 в MAIN BUFF для COM3
0x120	LCD_MAIN_COM4_A	ЗП	0x0	Значение сегментов с 31 по 0 в MAIN BUFF для COM4
0x124	LCD_MAIN_COM4_B	ЗП	0x0	Значение сегментов с 43 по 32 в MAIN BUFF для COM4
0x128	LCD_MAIN_COM5_A	ЗП	0x0	Значение сегментов с 31 по 0 в MAIN BUFF для COM5

Адрес	Название	Доступ	Значение после сброса	Описание
0x12C	LCD_MAIN_COM5_B	ЗП	0x0	Значение сегментов с 43 по 32 в MAIN BUFF для COM5
0x130	LCD_MAIN_COM6_A	ЗП	0x0	Значение сегментов с 31 по 0 в MAIN BUFF для COM6
0x134	LCD_MAIN_COM6_B	ЗП	0x0	Значение сегментов с 43 по 32 в MAIN BUFF для COM6
0x138	LCD_MAIN_COM7_A	ЗП	0x0	Значение сегментов с 31 по 0 в MAIN BUFF для COM7
0x13C	LCD_MAIN_COM7_B	ЗП	0x0	Значение сегментов с 43 по 32 в MAIN BUFF для COM7
0x200	LCD_SEC_COM0_A	ЗП	0x0	Значение сегментов с 31 по 0 в SEC BUFF для COM0
0x204	LCD_SEC_COM0_B	ЗП	0x0	Значение сегментов с 43 по 32 в SEC BUFF для COM0
0x208	LCD_SEC_COM1_A	ЗП	0x0	Значение сегментов с 31 по 0 в SEC BUFF для COM1
0x20C	LCD_SEC_COM1_B	ЗП	0x0	Значение сегментов с 43 по 32 в SEC BUFF для COM1
0x210	LCD_SEC_COM2_A	ЗП	0x0	Значение сегментов с 31 по 0 в SEC BUFF для COM2
0x214	LCD_SEC_COM2_B	ЗП	0x0	Значение сегментов с 43 по 32 в SEC BUFF для COM2
0x218	LCD_SEC_COM3_A	ЗП	0x0	Значение сегментов с 31 по 0 в SEC BUFF для COM3
0x21C	LCD_SEC_COM3_B	ЗП	0x0	Значение сегментов с 43 по 32 в SEC BUFF для COM3
0x220	LCD_SEC_COM4_A	ЗП	0x0	Значение сегментов с 31 по 0 в SEC BUFF для COM4
0x224	LCD_SEC_COM4_B	ЗП	0x0	Значение сегментов с 43 по 32 в SEC BUFF для COM4
0x228	LCD_SEC_COM5_A	ЗП	0x0	Значение сегментов с 31 по 0 в SEC BUFF для COM5
0x22C	LCD_SEC_COM5_B	ЗП	0x0	Значение сегментов с 43 по 32 в SEC BUFF для COM5
0x230	LCD_SEC_COM6_A	ЗП	0x0	Значение сегментов с 31 по 0 в SEC BUFF для COM6
0x234	LCD_SEC_COM6_B	ЗП	0x0	Значение сегментов с 43 по 32 в SEC BUFF для COM6
0x238	LCD_SEC_COM7_A	ЗП	0x0	Значение сегментов с 31 по 0 в SEC BUFF для COM7
0x23C	LCD_SEC_COM7_B	ЗП	0x0	Значение сегментов с 43 по 32 в SEC BUFF для COM7

1.3.9.8.15.3 Описание полей регистров

1.3.9.8.15.3.1 LCD_ID (0x000)

Регистр идентификатора LCD контроллера

В таблице 1.308 представлено описание полей регистра.

Таблица 1.308 – Описание полей регистра LCD_ID

Биты	Поле	Доступ	Значение после сброса	Описание
31:0	ID	ЧТ	0x3044434C	Идентификатор контроллера LCD

1.3.9.8.15.3.2 LCD_VERSION (0x004)

Регистр версии LCD контроллера

В таблице 1.309 представлено описание полей регистра.

Таблица 1.309 – Описание полей регистра LCD_VERSION

Биты	Поле	Доступ	Значение после сброса	Описание
31:0	LCD_VERION	ЧТ	0x13371448	Версия контролера LCD

1.3.9.8.15.3.3 LCD_SW_RST (0x008)

Регистр программного сброса LCD контроллера

В таблице 1.310 представлено описание полей регистра.

Таблица 1.310 – Описание полей регистра LCD_SW_RST

Биты	Поле	Доступ	Значение после сброса	Описание
0	SW_RST	ЧТ/ЗП	0	Производит программный сброс. Для этого необходимо установить данный бит в единицу. Когда он сбросится в нуль — это означает, что программный сброс завершен. Сбрасывается только аппаратно.
31:1	-	ЧТ	0	Зарезервировано

1.3.9.8.15.3.4 LCD_EN (0x00C)

Регистр включения LCD контроллера

В таблице 1.311 представлено описание полей регистра.

Таблица 1.311 – Описание полей регистра LCD_EN

Биты	Поле	Доступ	Значение после сброса	Описание
0	EN	ЧТ/ЗП	0	Устанавливает запрос на включение или выключение контроллера. Запрос считается исполненным, когда значение этого бита будет совпадать с LCD_CTRL[EN_STATUS]. Если в процессе включения (выключения) сбросить (установить) этот бит, то будет отрабатываться уже новый запрос. При этом старый запрос может как успеть отработать, так и не успеть. 0 - Запрос на выключение контроллера или он уже выключен. 1 - Запрос на включение контроллера или он уже включен.
31:1	-	ЧТ	0	Зарезервировано

1.3.9.8.15.3.5 LCD_CTRL (0x010)

Регистр настройки LCD контроллера

В таблице 1.312 представлено описание полей регистра.

Таблица 1.312 – Описание полей регистра LCD_CTRL

Биты	Поле	Доступ	Значение после сброса	Описание
0	EN_STATUS	ЧТ	0	Статусный бит включения LCD контроллера. 0 - LCD контроллер выключен. 1 - LCD контроллер включен.
3:1	-	ЧТ	0	Зарезервировано
5:4	BIAS	ЧТ/ЗП	0	Установка количества уровней выходного напряжения. Когда DUTY = 0, то BIAS считается равным единице и данное битовое поле перестает влиять на работу. 00 - Два уровня (1/2) 01 - Три уровня (1/3) 10 - Четыре уровня (1/4) 11 - Зарезервировано
7:6	-	ЧТ	0	Зарезервировано

Биты	Поле	До-ступ	Значение по-сле сброса	Описание
10:8	DUTY	ЧТ/ЗП	0	Установка количества используемых COMx выходов. 000 - Используется только COM0 001 - Используются COM0 и COM1 010 - Используются COM0, COM1, COM2 011 - Используются COM0, COM1, COM2, COM3 100 - Используются COM0, COM1, COM2, COM3, COM4 101 - Используются COM0, COM1, COM2, COM3, COM4, COM5 110 - Используются COM0, COM1, COM2, COM3, COM4, COM5, COM6 111 - Используются COM0, COM1, COM2, COM3, COM4, COM5, COM6, COM7
11	-	ЧТ	0	Зарезервировано
12	BUFF_MODE	ЧТ/ЗП	0	Режим работы буферов 0 - MANUAL 1 - BLINK
15:13	-	ЧТ	0	Зарезервировано
19:16	DIV	ЧТ/ЗП	0	Делитель частоты 0000 - 0 0001 - 1 0010 - 2 и т.д.
23:20	PS	ЧТ/ЗП	0	Делитель частоты 0000 - 0 0001 - 1 0010 - 2 и т.д.
26:24	DEAD_TIME	ЧТ/ЗП	0	Данное битовое поле задает длину временного интервала (dead time) между всеми кадрами. В течение него выходное напряжение на lcd_segX_o и lcd_comX_o становился равным Vss. Это позволяет уменьшить контрастность, не меняя сами кадры. 000 - dead time выключен. 001 - dead time равен одному периоду развертки. 010 - dead time равен двум периодам развертки. 011 - dead time равен трем периодам развертки. и т.д.
27	-	ЧТ	0	Зарезервировано
28	HD	ЧТ/ЗП	0	Изменяет выходное сопротивление делителя напряжения. 0 - высокое сопротивление. 1 - низкое сопротивление.
31:29	-	ЧТ	0	Зарезервировано

1.3.9.8.15.3.6 LCD_BLINK (0x014)

Регистр настройки задержки между сменой кадров LCD контроллера

В таблице 1.313 представлено описание полей регистра.

Таблица 1.313 – Описание полей регистра LCD_BLINK

Биты	Поле	Доступ	Значение после сброса	Описание
31:10	-	ЧТ	0	Зарезервировано
9:0	BLINK_PRD	ЧТ/ЗП	0	Задает количество кадров прежде, чем поменяется рабочий буфер в режиме BLINK. 0 - Каждый кадр меняется рабочий буфер 1 - После каждого второго кадра меняется рабочий буфер 2 - После каждого третьего кадра меняется рабочий буфер и т.д.

1.3.9.8.15.3.7 LCD_BUFF_STATUS (0x018)

Регистр статуса буферов LCD контроллера

В таблице 1.314 представлено описание полей регистра.

Таблица 1.314 – Описание полей регистра LCD_BUFF_STATUS

Биты	Поле	Доступ	Значение после сброса	Описание
0	SWAP_BUFF_REQ	ЧТ/ЗП	0	Устанавливает запрос на изменение рабочего буфера. Когда включен режим буферов BLINK, то запись в данный бит не игнорируется и запись происходит. При этом данный бит не влияет на работу и не сбрасывается аппаратно. Если включен режим MANUAL, то запись в данный бит меняет рабочий буфер на следующем кадре. 0 - нет запроса на изменение рабочего буфера. 1 - установлен запрос на изменение рабочего буфера.
3:1	-	ЧТ	0	Зарезервировано
4	ACT_BUFF	ЧТ	0	Отображает какой из буферов является активным в данный момент. 0 - рабочий буфер MAIN. 1 - рабочий буфер SEC.
31:5	-	ЧТ	0	Зарезервировано

1.3.9.8.15.3.8 LCD_IRQ_STATUS (0x01C)

Регистр статуса прерываний LCD контроллера

В таблице 1.315 представлено описание полей регистра.

Таблица 1.315 – Описание полей регистра LCD_IRQ_STATUS

Биты	Поле	Доступ	Значение после сброса	Описание
0	TEST	ЧТ	0	Статусный бит прерывания TEST. Когда данный бит находится в единице, то данное прерывание установлено.
3:1	-	ЧТ	0	Зарезервировано
4	SOF	ЧТ	0	Данный бит устанавливается когда начинается новый кадр и удерживается, пока не будет сброшен через регистр LCD_IRQ_CLEAR.
7:5	-	ЧТ	0	Зарезервировано
8	BUFF_SWAP	ЧТ	0	Данный бит устанавливается после того, как произошла смена рабочего буфера и удерживается, пока не будет сброшен через регистр LCD_IRQ_CLEAR.
11:9	-	ЧТ	0	Зарезервировано
12	TURN_ON	ЧТ	0	Данный бит устанавливается после того, как происходит включение LCD контроллера удерживается, пока не будет сброшен через регистр LCD_IRQ_CLEAR.
15:13	-	ЧТ	0	Зарезервировано
16	TURN_OFF	ЧТ	0	Данный бит устанавливается после того, как происходит выключение LCD контроллера удерживается, пока не будет сброшен через регистр LCD_IRQ_CLEAR.
31:17	-	ЧТ	0	Зарезервировано

1.3.9.8.15.3.9 LCD_IRQ_MASK (0x020)

Регистр маски прерываний LCD контроллера

В таблице 1.316 представлено описание полей регистра.

Таблица 1.316 – Описание полей регистра LCD_IRQ_MASK

Биты	Поле	Доступ	Значение после сброса	Описание
0	MASK_TEST	ЧТ/ЗП	0	Маскирование прерывания TEST. 0 - прерывание запрещено. 1 - прерывание разрешено.
3:1	-	ЧТ	0	Зарезервировано
4	MASK_SOF	ЧТ/ЗП	0	Маскирование прерывания SOF. 0 - прерывание запрещено. 1 - прерывание разрешено.
7:5	-	ЧТ	0	Зарезервировано
8	MASK_BUFF_SWAP	ЧТ/ЗП	0	Маскирование прерывания BUFF_SWAP. 0 - прерывание запрещено. 1 - прерывание разрешено.

Биты	Поле	Доступ	Значение после сброса	Описание
11:9	-	ЧТ	0	Зарезервировано
12	MASK_TURN_ON	ЧТ/ЗП	0	Маскирование прерывания TURN_ON. 0 - прерывание запрещено. 1 - прерывание разрешено.
15:13	-	ЧТ	0	Зарезервировано
16	MASK_TURN_OFF	ЧТ/ЗП	0	Маскирование прерывания TURN_OFF. 0 - прерывание запрещено. 1 - прерывание разрешено.
31:17	-	ЧТ	0	Зарезервировано

1.3.9.8.15.3.10LCD_IRQ_CLEAR (0x024)

Регистр сброса прерываний LCD контроллера

В таблице 1.317 представлено описание полей регистра.

Таблица 1.317 – Описание полей регистра LCD_IRQ_CLEAR

Биты	Поле	Доступ	Значение после сброса	Описание
0	CLEAR_TEST	ЗП	0	Запись единицы в данный бит сбрасывает прерывание TEST.
3:1	-	ЧТ	0	Зарезервировано
4	CLEAR_SOF	ЗП	0	Запись единицы в данный бит сбрасывает прерывание SOF.
7:5	-	ЧТ	0	Зарезервировано
8	CLEAR_BUFF_SWAP	ЗП	0	Запись единицы в данный бит сбрасывает прерывание BUFF_SWAP.
11:9	-	ЧТ	0	Зарезервировано
12	CLEAR_TURN_ON	ЗП	0	Запись единицы в данный бит сбрасывает прерывание TURN_ON.
15:13	-	ЧТ	0	Зарезервировано
16	CLEAR_TURN_OFF	ЗП	0	Запись единицы в данный бит сбрасывает прерывание TURN_OFF.
31:17	-	ЧТ	0	Зарезервировано

1.3.9.8.15.3.11LCD_IRQ_TEST (0x028)

Регистр установки тестового прерывания LCD контроллера

В таблице 1.318 представлено описание полей регистра.

Таблица 1.318 – Описание полей регистра LCD_IRQ_TEST

Биты	Поле	Доступ	Значение после сброса	Описание
0	TEST	ЗП	0	Запись единицы в данный бит устанавливает прерывание TEST.
31:1	-	ЧТ	0	Зарезервировано

1.3.9.8.15.3.12LCD_OUT_LEVELS (0x034)

Регистр сброса прерываний LCD контроллера

В таблице 1.319 представлено описание полей регистра.

Таблица 1.319 – Описание полей регистра LCD_OUT_LEVELS

Биты	Поле	Доступ	Значение после сброса	Описание
3:0	SEG_ACT_POS	ЧТ/ЗП	0	Устанавливается напряжение на выходах SEGx во время позитивного полукадра, когда данный сегмент включен. 0000 - rail_0. 0001 - rail_1. 0010 - rail_2. 0100 - rail_3. 1000 - rail_4. остальные - Зарезервировано.
7:4	SEG_ACT_NEG	ЧТ/ЗП	0x8	Устанавливается напряжение на выходах SEGx во время негативного полукадра, когда данный сегмент включен. 0000 - rail_0. 0001 - rail_1. 0010 - rail_2. 0100 - rail_3. 1000 - rail_4. остальные - Зарезервировано.
11:8	SEG_INACT_POS	ЧТ/ЗП	0x8	Устанавливается напряжение на выходах SEGx во время позитивного полукадра, когда данный сегмент выключен. 0000 - rail_0. 0001 - rail_1. 0010 - rail_2. 0100 - rail_3. 1000 - rail_4. остальные - Зарезервировано.

Биты	Поле	До-ступ	Значение по-сле сброса	Описание
15:12	SEG_INACT_NEG	ЧТ/ЗП	0	Устанавливается напряжение на выходах SEGx во время негативного полукадра, когда данный сегмент выключен. 0000 - rail_0. 0001 - rail_1. 0010 - rail_2. 0100 - rail_3. 1000 - rail_4. остальные - Зарезервировано.
19:16	COM_ACT_POS	ЧТ/ЗП	0x8	Устанавливается напряжение на выходе COMx во время позитивного полукадра, когда данный COMx активен. 0000 - rail_0. 0001 - rail_1. 0010 - rail_2. 0100 - rail_3. 1000 - rail_4. остальные - Зарезервировано.
23:20	COM_ACT_NEG	ЧТ/ЗП	0	Устанавливается напряжение на выходе COMx во время негативного полукадра, когда данный COMx активен. 0000 - rail_0. 0001 - rail_1. 0010 - rail_2. 0100 - rail_3. 1000 - rail_4. остальные - Зарезервировано.
27:24	COM_INACT_POS	ЧТ/ЗП	0x8	Устанавливается напряжение на выходе COMx во время позитивного полукадра, когда данный COMx неактивен. 0000 - rail_0. 0001 - rail_1. 0010 - rail_2. 0100 - rail_3. 1000 - rail_4. остальные - Зарезервировано.
31:28	COM_INACT_NEG	ЧТ/ЗП	0	Устанавливается напряжение на выходе COMx во время негативного полукадра, когда данный COMx неактивен. 0000 - rail_0. 0001 - rail_1. 0010 - rail_2. 0100 - rail_3. 1000 - rail_4. остальные - Зарезервировано.

1.3.9.8.15.3.13LCD_MAIN_COM0_A (0x100)

Регистр MAIN буфера для COM0 LCD контроллера

В таблице 1.320 представлено описание полей регистра.

Таблица 1.320 – Описание полей регистра LCD_MAIN_COM0_A

Биты	Поле	Доступ	Значение после сброса	Описание
0	SEG0_COM0	ЗП	0	Активность сегмента между SEG0 и COM0 0 - не активен 1 - активен
1	SEG1_COM0	ЗП	0	Активность сегмента между SEG1 и COM0 0 - не активен 1 - активен
. . .				
31	SEG31_COM0	ЗП	0	Активность сегмента между SEG31 и COM0 0 - не активен 1 - активен

1.3.9.8.15.3.14LCD_MAIN_COM0_B (0x104)

Регистр MAIN буфера для COM0 LCD контроллера

В таблице 1.321 представлено описание полей регистра.

Таблица 1.321 – Описание полей регистра LCD_MAIN_COM0_B

Биты	Поле	Доступ	Значение после сброса	Описание
0	SEG32_COM0	ЗП	0	Активность сегмента между SEG32 и COM0 0 - не активен 1 - активен
1	SEG33_COM0	ЗП	0	Активность сегмента между SEG33 и COM0 0 - не активен 1 - активен
. . .				
10	SEG43_COM0	ЗП	0	Активность сегмента между SEG43 и COM0 0 - не активен 1 - активен
31:11	-	ЧТ	0	Зарезервировано

Описание регистров LCD_MAIN_COMx_A, LCD_MAIN_COMx_B и LCD_SEC_COMx_A, LCD_SEC_COMx_B аналогично описанию регистров LCD_MAIN_COM0_A и LCD_MAIN_COM0_B.

1.3.9.9 Контроллер интерфейса USB (USB_FS)

1.3.9.9.1 Характеристики контроллера USB_FS

Контроллер USB_FS обладает следующими характеристиками:

– полная совместимость со спецификацией USB 2.0;

– возможность работы с устройствами и хостами на полной и низкой скорости (FS - 12 Мбит/с, LS - 1,5 Мбит/с);

– 32 канала;

– поддержка всех типов передачи данных (Control, Interrupt, Bulk и Isochronous);

– программно настраиваемое направление каналов, тип передачи и размер FIFO;

– поддержка аппаратного планирования и улучшенной работы с большими буферами;

– автоматическая повторная передача поврежденных пакетов данных;

– режимы низкого потребления:

– возможность снятия тактирования;

– возможность снятия питания;

– работа от двух тактовых сигналов (см. 1.3.9.9.7).

Благодаря описанным выше возможностям, USB_FS является высокоэффективным с точки зрения ПО:

– все критические функции, связанные с USB, реализованы аппаратно. Поэтому для нормальной работы программное обеспечение должно управлять только процессом перечисления (enumeration);

– как только канал установлен, единственной задачей ПО является обеспечение буферов данных (весь протокол USB управляется аппаратными средствами, включая переключение данных, повторные передачи, опрос периодических каналов). Это сводит количество и частоту программных прерываний к минимуму;

– задержка прерываний зависит не от времени, требуемого уровнем пакетов USB, а от размера буферов данных и пропускной способности каналов;

– при смене системной частоты контроллер не требует перенастройки.

1.3.9.9.2 Описание внешних выводов контроллера USB

Список внешних выводов контроллера USB и их описание приведены в таблице 1.322

Таблица 1.322 – Список внешних выводов контроллера USB

Наименование вывода	Тип (Вх/Вых)	Функциональное назначение
VDD_USB	Вх	Аналоговое питание приёмо-передатчика USB_FS (3,3 В)
USB_DM	Вх/Вых	Отрицательная линия двунаправленного дифференциального сигнала шины USB
USB_DP	Вх/Вых	Положительная линия двунаправленного дифференциального сигнала шины USB

1.3.9.9.3 Устройство и функционирование блока

1.3.9.9.3.1 Интерфейс DMA Master

1.3.9.9.3.1.1 Введение

USB_FS является мастером на системной шине и использует DMA механизм для:

– записи данных, полученных по USB шине;

– чтения данных для их передачи по шине USB.

USB_FS запрашивает все операции чтения и записи DMA автономно. USB_FS поддерживает один канал DMA, но может обращаться к любым доступным адресам на системной шине.

USB_FS DMA имеет внутренний буфер памяти, используемый для хранения передаваемых и принимаемых данных и управляющей информации.

1.3.9.9.3.1.2 Управление внутренним буфером в режиме хоста

USB имеет байтовую адресацию. Тем не менее, все операции чтения и записи DMA должны быть выровнены по границе 4 байт.

Получая данные от шины USB, контроллер USB_FS ожидает, пока уровень заполнения буфера не превысит количество байт, заданное программно с помощью поля *DMA Receive Burst Length Field*. Все полученные по шине USB данные будут разделены на одну или несколько транзакций записи DMA.

Если USB_FS работает в режиме хоста и собирается отправить данные через USB, он ждет, пока первые байты данных не окажутся в буфере. Программное обеспечение может указать, сколько данных должно быть предварительно вычитано для того, чтобы USB_FS начал передачу по шине USB, см. «*Уровень предварительной выборки при передаче DMA TX*».

USB_FS считывает данные по системной шине, разбивая их на одну или несколько транзакций чтения DMA. Длина транзакции может быть задана программно с помощью поля *DMA Transmit Burst Length Field*.

1.3.9.9.3.1.3 Управление внутренним буфером в режиме устройства

Наличие внутреннего буфера освобождает системную шину от жестких требований к задержкам и пропускной способности протокола USB. Для каждого канала может быть буферизован ровно 1 USB-пакет.

Для операций передачи (каналы IN) DMA должен предварительно выбрать данные из основной памяти во внутренний буфер, прежде чем начнется транзакция USB. Если DMA (еще) не выполнил предварительную выборку данных, USB_FS отвечает на токен IN завершением NAK.

Для операций приема (каналы OUT) DMA сохраняет данные, полученные от шины USB во внутренний буфер. DMA начнёт перемещать данные в системную память по завершении транзакции USB. Если USB_FS получает токен OUT, а внутренний буфер занят, контроллер ответит на токен OUT завершением NAK.

DMA управляет одним отдельным подбуфером для каждого канала. Состояние заполнения подбуфера может быть определено программно путем чтения поля *DMA Buffer Valid Field*. Управление распределением памяти буфера DMA является задачей программного обеспечения. На рисунке 1.85 показано расположение подбуферов для каждого канала во внутреннем буфере.

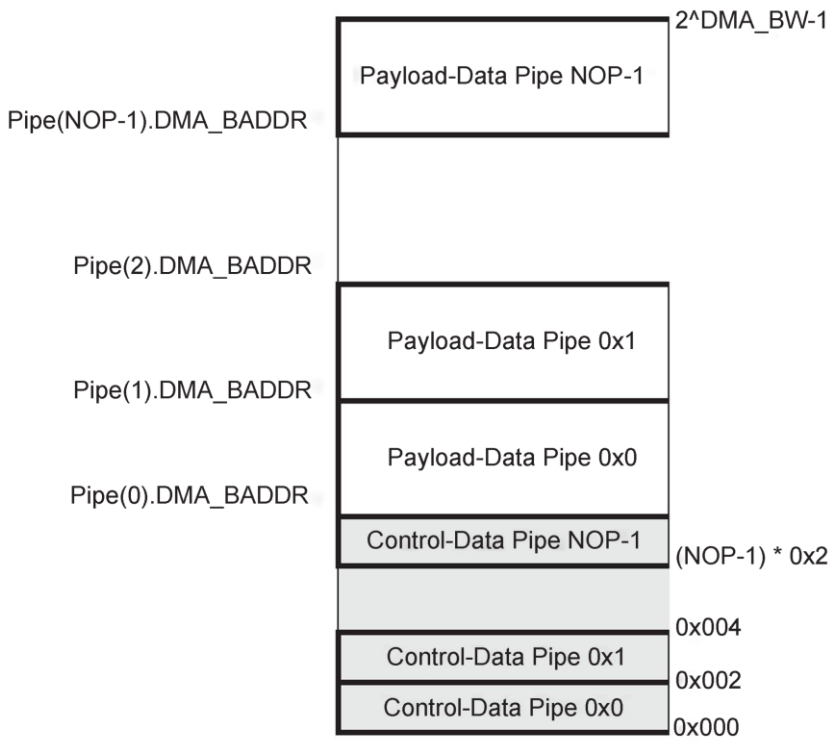


Рисунок 1.85 – Карта памяти внутреннего буфера DMA

DMA должен хранить во внутреннем буфере 8 байт (или 2 двойных слова) внутренней управляющей информации для каждого канала. Эта область начинается с адреса 0x000. Таким образом, первый возможный адрес для данных полезной нагрузки $NOP * 2$.

Важно, чтобы начальные адреса подбуферов были точно рассчитаны программным обеспечением и записаны в поле адреса *DMA Buffer Address Field* для каждого канала. Количество двойных слов, зарезервированных для каждого канала, рассчитывается с помощью поля *Maximum Packet Size Field*. Максимальный размер пакета должен быть округлен в большую сторону до следующей границы двойного слова:

$$NUM - DWORDS = int((MPS + 3)/4) + 1$$

Начальный адрес для определенного канала i может быть вычислен по начальному адресу и размеру подбуфера предыдущего канала:

$$START\text{–}ADDRESS_i = NUM\text{–}DWORDS_{i-1} + START\text{–}ADDRESS_{i-1}$$

DMA может глобально управляться битом *DMA Manager Flush Bit*. Когда этот бит программно устанавливается в 1, DMA прекращает свою работу с каналами передачи. Каналы приема обрабатываются по-прежнему. Когда этот бит переходит из 1 в 0, DMA возвращается к исходному состоянию и начинает свою работу.

1.3.9.9.4 Описание регистров

1.3.9.9.4.1 Общее описание регистров

Контроллер внешних прерываний USB_FS имеет набор программно доступных регистров, доступных на шине AHB по базовому адресу USB_OTG = 0x40420000.

Карта регистров приведена в таблице 1.323.

В этом разделе приведено описание всех регистров контроллера USB_FS.

Все регистры имеют ширину 32 бита. Биты, помеченные как зарезервированные, всегда равны нулю.

Существуют различные типы доступа:

– rw : чтение/запись (read/write): бит(ы) считывается и может быть записан;

– r: только чтение (read-only): бит(ы) доступен(ы) для чтения и не доступен(ы) для записи. cow: очистка после записи (clear-on-write): бит(ы) очищается(ются) после записи «1».

1.3.9.9.4.2 Карта регистров

Таблица 1.323 – Карта регистров USB_FS

Адрес	Название	Доступ	Значение после сброса	Описание
0x000	ID	ЧТ	0x00000073	Идентификатор контроллера
0x004	USB_CTRL	ЧТ/ЗП	0x0	Регистр управления контроллером
0x008	FRM_TIMER	ЧТ/ЗП	0x0	Регистр таймера фрейма
0x00C	MAIN_EV	ЧТ/ЗП	0x0	Статус основных событий
0x010	MAIN_EM	ЧТ/ЗП	0x0	Маска основных событий
0x014	PIPE_EV	ЧТ/ЗП	0x0	Статус событий каналов
0x018	PIPE_EM	ЧТ/ЗП	0x0	Маска событий каналов
0x028	PIPE_SEL	ЧТ/ЗП	0x0	Регистр выбора канала
0x02C	PORT_STAT	ЧТ/ЗП	0x00000008	Регистр состояния порта
0x030	PORT_CTRL	ЧТ/ЗП	0x0	Регистр управления портом
0x034	PSC_EV	ЧТ/ЗП	0x0	Статус изменения состояния порта
0x038	PSC_EM	ЧТ/ЗП	0x0	Маска изменения состояния порта
0x03C	PIPE_DMA_CFG	ЧТ/ЗП	0x0	Регистр конфигурации DMA канала
0x040	PIPE_CTRL	ЧТ/ЗП	0x0	Регистр управления каналом
0x044	PIPE_CFG	ЧТ/ЗП	0x0	Регистр конфигурации канала
0x048	PIPE_ADDR	ЧТ/ЗП	0x0	Регистр адреса канала
0x04C	PIPE_STAT	ЧТ/ЗП	0x00000300	Регистр состояния канала
0x050	PIPE_DATA_PTR	ЧТ/ЗП	0x0	Указатель на данные канала
0x054	PIPE_DATA_TBYTES	ЧТ/ЗП	0x0	Параметры транзакции канала
0x058	PIPE_ADATA_PTR	ЧТ/ЗП	0x0	Альтернативный указатель на данные канала

Адрес	Название	До-ступ	Значение по-сле сброса	Описание
0x05C	PIPE_ADATA_TBYTES	ЧТ/ЗП	0x0	Альтернативные параметры транзакции канала
0x060	DEBUG_CTRL	ЧТ/ЗП	0x0	Регистр управления отладкой
0x064	DEBUG_PID	ЧТ/ЗП	0x0	Отладочные PID
0x068	DEBUG_STAT	ЧТ	0x0	Статус отладки
0x06C	TEST	ЧТ/ЗП	0x5AF8CAFE	Тестовый регистр
0x080	MAIN_CFG	ЧТ	0x00000820	Регистр основных конфигураций
0x084	MODE_CFG	ЧТ	0x0	Регистр конфигурации режима
0x088	DMA_CONFIG	ЧТ	0x000D2020	Регистр конфигурации DMA
0x0A0	DMA_RX_CFG	ЧТ/ЗП	0x00000001	Регистр конфигурации DMA RX
0x0A4	DMA_TX_CFG	ЧТ/ЗП	0x00140001	Регистр конфигурации DMA TX
0x0A8	DMA_DEV_CFG	ЧТ/ЗП	0x0	Адрес данных SETUP

1.3.9.9.4.3 Описание полей регистров

1.3.9.9.4.3.1 Общая группа регистров

1.3.9.9.4.3.1.1 Регистр ID

Идентификатор: ID

Адрес 0x000

В таблице 1.324 представлено описание полей регистра.

Таблица 1.324 – Регистр ID

Имя поля	Поле	Состояние после сброса	Программный до-ступ	Описание
–	31: 13	–	–	Зарезервировано
CORE_ID	12: 8	0x7	r	ID ядра
REV_ID	7: 0	0x07	r	ID ревизии

ID ядра . Это битовое поле показывает уникальный ID ядра.

ID ревизии . Эти биты показывают номер ревизии ядра.

1.3.9.9.4.3.1.2 Регистр управления USB

Этот регистр действителен, только если ядро работает в режиме хоста.

Идентификатор: USB_CTRL

Адрес 0x004

В таблице 1.325 представлено описание полей регистра.

Таблица 1.325 – Регистр управления USB

Имя поля	Поле	Состояние после сброса	Программный доступ	Описание
–	31: 6	–	–	Зарезервировано
XCVRP	5	0x0	rw	Патч для трансивера
PSE	4	0x0	rw	Периодическое расписание разрешить
ASE	3	0x0	rw	Асинхронное расписание разрешить
HRS	2	0x0	rw	Пуск/стоп хоста
XSUSP	1	0x0	rw	Приостановка XCVR
CSUSP	0	0x0	rw	Приостановка ядра

Патч для трансивера . При установке значения 1 каждый SOP будет начинаться с перехода J в K вместо IDLE (ожидание) в K. Это может привести к улучшению качества сигнала при использовании приемопередатчиков с большой задержкой разрешения выхода. Должен быть установлен в микроконтроллере.

Периодическое расписание разрешить . При установке значения 1 и установленном бите *Host Run/Stop* ядро планирует периодические транзакции (прерывания и изохронные). Если программа установит этот бит в ноль, все ожидающие периодические транзакции текущего фрейма будут завершены.

Асинхронное расписание разрешить . Если установлено значение 1 и установлен бит *Host Run/Stop*, ядро планирует асинхронные транзакции (массовые и управляющие). Если программа установит этот бит в ноль, все ожидающие асинхронные транзакции текущего фрейма будут завершены.

Пуск/стоп хоста . Если установлен в 1, ядро запускает фреймворк и планирование транзакций. Если программа установит этот бит в ноль, все ожидающие транзакции текущего фрейма будут завершены, а в конце фрейма будет установлен бит события *Host Controller Halted Event Bit* , и больше SOF посылаться не будут.

Бит будет очищен аппаратно, если была обнаружена серьезная ошибка.

Приостановка XCVR . Этот бит управляет выходным сигналом xcvr_suspend_n. Если этот бит установлен в 1, низкоактивный выход xcvr_suspendm_n будет установлен на 0, чтобы включить режим низкого энергопотребления подключенного трансивера. Любое изменение в сигнале ser_rx_dm или ser_rx_dp приведет к деактивации выходного сигнала xcvr_suspend_n. (Возврат к 1)

Приостановка ядра . Этот бит управляет выходным сигналом core suspend_n. Если этот бит установлен в 1, низкоактивный выход core suspend_n будет установлен на 0, чтобы включить режим низкого энергопотребления этого ядра. Этот сигнал должен быть подключен к внешнему тактовому контроллеру. Любое изменение сигналов ser_rx_dm или ser_rx_dp приведет к деактивации выходного сигнала core suspend n. (Возврат к 1)

1.3.9.9.4.3.1.3 Регистр таймера фрейма

Идентификатор: FRM_TIMER

Адрес 0x008

В таблице 1.326 представлено описание полей регистра.

Таблица 1.326 – Регистр таймера фрейма

Имя поля	Поле	Состояние после сброса	Программный доступ	Описание
–	31: 13	–	–	Зарезервировано
AGSOF	12	0x0	r	Искусственно сгенерированный SOF
FTLOCK	11	0x0	r	Таймер фрейма заблокирован
FRM_NR	10: 0	0x000	rw	Номер текущего фрейма

Таблица 9.3. Регистр таймера фрейма

Искусственно сгенерированный SOF . Это поле действительно, только если ядро работает в периферийном режиме.

Этот бит устанавливается, если бит *Frame Event Bit* регистра *Main Event Register* был сгенерирован искусственно в случае потери токена SOF.

Таймер фрейма заблокирован . Это поле действительно, только если ядро работает в периферийном режиме.

Оно указывает на то, что внутренний таймер фрейма заблокирован. Оно устанавливается, если получено более двух последовательных токенов SOF, и очищается, если пропущено не менее двух последовательных токенов SOF.

Номер текущего фрейма . В этом поле отображается номер текущего фрейма.

Если ядро работает в режиме хоста, таймер фрейма – это свободно работающий счетчик, который будет увеличиваться каждую 1 мс. Этот регистр не может быть записан, если контроллер хоста не находится в состоянии Halted.

Если ядро работает в периферийном режиме, таймер фреймов будет синхронизироваться с поступающими токенами SOF. В этом поле отображается номер фрейма текущего полученного токена SOF.

1.3.9.9.4.3.2 Группа регистров событий

1.3.9.9.4.3.2.1 Регистр основных событий

Идентификатор: MAIN_EV

Адрес 0x00C

В таблице 1.327 представлено описание полей регистра.

Таблица 1.327 – Регистр основных событий

Имя поля	Поле	Состояние после сброса	Программный доступ	Описание
–	31: 6	–	–	Зарезервировано
BWERR_EV	5	0x0	cow	Событие ошибки пропускной способности
HCHA_EV	4	0x0	cow	Событие остановки контроллера хоста
GPIPE_EV	3	0x0	r	Событие передачи глобального канала
GPORT_EV	2	0x0	r	Событие изменения состояния глобального порта
FRM32_EV	1	0x0	cow	Событие фрейма 32
FRM_EV	0	0x0	cow	Событие фрейма

Событие ошибки пропускной способности . Это поле действительно, только если ядро работает в режиме хоста.

Этот бит будет установлен аппаратно, если для завершения всех периодических передач во фрейме не хватает пропускной способности. Это приводит к прерыванию, если установлен бит *Bandwidth Error Event Mask Bit* регистра *Main Event Mask Register*.

Событие остановки контроллера хоста . Это поле действительно, только если ядро работает в режиме хоста.

Бит устанавливается, если контроллер хоста остановился в результате установки бита *Host Run/Stop Bit* в состояние остановки и все ожидающие транзакции фрейма завершены. Это приводит к прерыванию, если бы установлен соответствующий бит *HC Halted Mask Bit* регистра *Main Event Mask Register*.

Событие передачи глобального канала . Бит устанавливается, если установлен один из битов *Pipe Event Register*. Это приводит к прерыванию, если установлен бит *Global Pipe Event Mask Bit* регистра *Main Event Mask Register*.

Событие изменения состояния глобального порта . Бит устанавливается, если установлен один из битов *Port Status Change Event Register*. Это приводит к прерыванию, если установлен бит *Global Port Status Change Event Mask Bit* регистра *Main Event Mask Register*.

Событие фрейма 32 . Бит устанавливается после каждого 32-го токена SOF. Это приводит к прерыванию, если был установлен соответствующий бит *Frame 32 Event Mask Bit* регистра *Main Event Mask Register*. Может использоваться программными таймерами.

Событие фрейма . Этот бит устанавливается каждую 1 мс. Это приводит к прерыванию, если установлен бит *Frame Event Mask Bit* регистра *Main Event Mask Register*.

1.3.9.9.4.3.2.2 Регистр маски основных событий

Идентификатор: MAIN_EM

Адрес 0x010

В таблице 1.328 представлено описание полей регистра.

Таблица 1.328 – Регистр маски основных событий

Имя поля	Поле	Состояние после сброса	Программный доступ	Описание
–	31: 6	–	–	Зарезервировано
BWERR_EM	5	0x0	rw	Маска события ошибки пропускной способности
HCHA_EM	4	0x0	rw	Маска остановки HC
GPIPE_EM	3	0x0	rw	Маска событий глобального канала
GPORT_EM	2	0x0	rw	Маска события изменения состояния глобального порта
FRM32_EM	1	0x0	rw	Маска события фрейма 32
FRM_EM	0	0x0	rw	Маска события фрейма

Маска события ошибки пропускной способности . Установка этого бита приведет к прерыванию, если активен бит *Bandwidth Error Event Bit*.

Маска остановки HC . Установка этого бита приведет к прерыванию, если активен бит *Host Controller Halted Event Bit*.

Маска событий глобального канала . Установка этого бита приведет к прерыванию, если активен бит *Global Pipe Transfer Event Bit*.

Маска события изменения состояния глобального порта . Установка этого бита приведет к прерыванию, если активен бит *Global Port Status Change Event Bit*.

Маска события фрейма 32 . Установка этого бита приведет к прерыванию, если активен бит *Frame 32 Event Bit*.

Маска события фрейма . Установка этого бита приведет к прерыванию, если активен бит *Frame Event Bit*.

1.3.9.9.4.3.2.3 Регистр событий каналов

NOP = число каналов = 32.

Идентификатор: PIPE_EV

Адрес 0x014

В таблице 1.329 представлено описание полей регистра.

Таблица 1.329 – Регистр событий каналов

Имя поля	Поле	Состояние после сброса	Программный доступ	Описание
–	31: NOP	–	–	Зарезервировано
PI_EV	NOP-1: 0	0x0000	cow	Событие канала

Событие канала . В этом битовом поле существует 1 бит для каждого канала, для которого сконфигурировано это ядро. Номер канала соответствует индексу бита в этом битовом поле.

Если ядро работает в режиме хоста, это событие запускается ядром, если выполняется одно из следующих условий: Канал становится неактивным, а бит *Active Status Bit* из *Pipe Status Register* был очищен аппаратными средствами из-за нормального завершения или события ошибки. Буфер данных был завершен: бит достоверности данных буфера *Data Buffer Valid Bit* или бит достоверности альтернативного буфера данных *Alternative Data Buffer Valid Bit* был очищен аппаратными средствами.

Если ядро работает в периферийном режиме, это событие запускается ядром, если буфер данных был завершен: бит достоверности данных буфера *Data Buffer Valid Bit* или бит достоверности альтернативного буфера данных *Alternative Data Buffer Valid Bit* был очищен аппаратными средствами.

1.3.9.9.4.3.2.4 Регистр маски событий каналов

Идентификатор: PIPE_EM

Адрес 0x018

В таблице 1.330 представлено описание полей регистра.

Таблица 1.330 – Регистр маски событий каналов

Имя поля	Поле	Состояние после сброса	Программный доступ	Описание
–	31: NOP	–	–	Зарезервировано
PI_EM	NOP-1: 0	0x0000	rw	Маска события канала

Маска события канала . Установка одного из этих битов разрешает генерацию прерывания по событию от канала для соответствующего канала.

1.3.9.9.4.3.3 Группа регистров страницы

1.3.9.9.4.3.3.1 Регистр выбора канала

Физическое регистровое пространство для регистров конфигурации и состояния канала распределяется постранично регистром *Pipe Select Register*. Таким образом, все регистры группы *Pipe Register Group* существуют для каждого канала.

Идентификатор: PIPE_SEL

Адрес 0x028

В таблице 1.331 представлено описание полей регистра.

Таблица 1.331 – Регистр выбора канала

Имя поля	Поле	Состояние после сброса	Программный доступ	Описание
–	31: 5	–	–	Зарезервировано
PI_SEL	4: 0	0x00	rw	Выбор канала

Выбор канала . В зависимости от количества поддерживаемых каналов (NOP), на которые настроено ядро, один из каналов может быть выбран с помощью этого регистра для доступа к соответствующим регистрам состояния и конфигурации.

Ширина этого битового поля зависит от параметра NOP и равна log2(NOP).

Это битовое поле кодируется следующим образом:

– b00000: Канал 0;

– … ;

– b11111: Канал 31.

1.3.9.9.4.3.4 Группа регистров порта

1.3.9.9.4.3.4.1 Регистр состояния порта

Идентификатор: PORT_STAT

Адрес 0x02C

В таблице 1.332 представлено описание полей регистра.

Таблица 1.332 – Регистр состояния порта

Имя поля	Поле	Состояние после сброса	Программный доступ	Описание
–	31: 10	–	–	Зарезервировано
LINESTATE	9: 8	0x0	r	Состояние линии USB
OCURC	7	0x0	r	Состояние превышения тока
DLS	6	0x0	r	Низкая скорость устройства
PCS	5	0x0	r	Состояние подключения порта
CONN_ID	4	0x0	r	Значение идентификатора разъема USB
VB_SESS_END	3	0x1	r	Конец сеанса VB
–	2	–	–	Зарезервировано
V_SESS_VLD	1	0x0	r	Vsession действителен
VBUS_VLD	0	0x0	r	Vbus действителен

Состояние линии USB . Этот бит показывает текущее состояние линии: LINESTATE[1:0] = [ser_rx_dp, ser_rx_dm].

Состояние превышения тока . Этот бит будет установлен на единицу, если ядро выявило превышение тока, в противном случае он равен нулю. Состояние перегрузки по току определяется так: сигнал vb_vld снят, а vb_on установлен.

Низкая скорость устройства . Это поле действительно, только если ядро работает в режиме хоста.

Этот бит указывает, является ли подключенное устройство полно- или низкоскоростным. Этот бит действителен, только если бит *Port Connect Status Bit* указывает на наличие подключенного устройства:

– b0: Подключено полноскоростное устройство;

– b1: Подключено низкоскоростное устройство.

Состояние подключения порта . Это поле действительно, только если ядро работает в режиме хоста.

Этот бит указывает, подключено ли устройство:

– b0: Устройство не подключено

[label = ”PCS_DISCONN”];

– b1: Устройство подключено к корневому порту

[label = ”PCS_CONN”].

Значение идентификатора разъема USB . Этот бит показывает значение идентификатора разъема USB. Если установлена единица, это указывает на то, что ядро работает как B-устройство с начальным периферийным режимом. Если установить его в ноль, ядро будет работать как A-устройство с начальным режимом хоста. Ядро может менять роли хоста и периферии с помощью протокола Host Negotiation Protocol (HNP). Таким образом, реальный режим работы ядра зависит от этого бита в сочетании с выбранным завершением (см. *Termination Select Bit*). При изменении этого бита аппаратура установит бит *Connect/Disconnect Event Bit*:

– b0: A-устройство (начальный хост)

[label = ”CONN_ID_A”];

– b1: B-устройство (начальное периферийное)

[label = ”CONN_ID_B”].

Конец сеанса VB . Аппаратное обеспечение установит этот бит в единицу, если установлен вход vb_sess_end, указывая, что V_{BUS} находится выше $V_{B\text{-}SESS\text{-}END}$.

Vsession действителен . Аппаратные средства установят этот бит в единицу, если установлен вход va_sess_vld, указывая, что V_{BUS} выше $V_{A\text{-}SESS\text{-}VLD}$ или $V_{B\text{-}SESS\text{-}VLD}$.

Vbus действителен . Аппаратные средства установят этот бит в единицу, если установлен вход vb_vld, что указывает на напряжение V_{BUS} выше 4,4 В.

Для OTG-устройств этот бит устанавливается в единицу, когда V_{BUS} превышает $V_{A\text{-}VBUS\text{-}VLD}$. Когда V_{BUS} опускается ниже этого порога, возникает состояние перегрузки по току.

1.3.9.9.4.3.4.2 Регистр управления портом

Идентификатор: PORT_CTRL

Адрес 0x030

В таблице 1.333 представлено описание полей регистра.

Таблица 1.333 – Регистр управления портом

Имя поля	Поле	Состояние после сброса	Программный доступ	Описание
–	31: 26	–	–	Зарезервировано
COID OV	25	0x0	rw	Перезапись идентификатора соединения
VBUS_VLD_OV	24	0x0	rw	Перезапись Vbus Valid
P_LEN	23: 16	0x00	rw	Длина импульса
–	15: 13	–	–	Зарезервировано
ID_PU	12	0x0	rw	Разрешить ID-Pullup
–	11	–	–	Зарезервировано
VBS_ON	10	0x0	rw	Управление запросом сеанса VBUS
DCHRG	9	0x0	rw	Разрешить схему разряда
TERM_ENA	8	0x0	rw	Разрешить завершение
TERM_SEL	7	0x0	rw	Выбор завершения
VB_ON	6	0x0	rw	Управление VBUS
PSUSP	5	0x0	rw	Приостановка порта
PENA	4	0x0	rw	Разрешить порт
FPRESU	3	0x0	rw	Принудительное возобновление
URESET	2	0x0	rw	Сброс USB
PTESTC	1: 0	0x0	rw	Контроль тестирования порта

Перезапись идентификатора соединения . Программа может использовать этот бит, чтобы принудить бит *USB Connector ID Value Bit* установить 1. Если программа установит бит на 1, внутреннее значение USB Connector ID Value будет принудительно равно 1, и ядро USB будет работать как устройство. Если бит установлен на 0, то используется значение бита *USB Connector ID Value Bit*.

Перезапись Vbus Valid . Программа может использовать этот бит, чтобы принудить бит *Vbus Valid Bit* установить 1. Если программа установит бит на 1, внутренний порт Vbus Valid будет принудительно приравнен 1. Если бит установлен в 0, используется значение бита *Vbus Valid Bit* .

Длина импульса . Он определяет длительность импульса в миллисекундах, если ядро намерено запросить сеанс с помощью импульсного сигнала линии данных или импульсного сигнала V_{BUS}. Установка этого битового поля в ноль позволяет программно управлять длиной

импульса. Если установить ненулевое значение, это поле будет обновляться аппаратно каждую миллисекунду:

– 0x00: Программно контролируемая длина импульса. Программа должна вручную сбросить бит *VBUS Session Request Control Bit* или *Termination Enable Bit*;

– 0x01: Длительность импульса 1 мс. Аппаратное обеспечение сбросит бит *VB US Session Request Control Bit* и бит *Termination Enable Bit* автоматически через 1 мс;

– 0xFF: Длина импульса 255 мс. Аппаратное обеспечение автоматически сбросит бит *VBUS Session Request Control Bit* и бит *Termination Enable Bit* через 255 мс.

Если в этом битовом поле произойдет обратный отсчет от единицы до нуля, будет установлен бит *Pulse End Event Bit* регистра *Main Event Register*.

Разрешить ID-Pullup . С помощью этого бита программа может управлять выходным сигналом ID-pullup (id_pullup).

*Я*дро обновит состояние бита *USB Connector ID Value Bit* только в том случае, если этот бит станет неактивным.

Управление запросом сеанса VBUS . С помощью этого бита программное обеспечение может управлять схемой питания (chrg_vbus).

Если он установлен на 1, будет установлен вывод vbs_on.

Разрешить схему разряда . С помощью этого бита программное обеспечение может управлять схемой разряда для ускорения разряда конденсаторов питания шины хоста.

Установка этого бита на 1 приведет к установке вывода dischrg_vbus.

Разрешить завершение . С помощью этого бита программа может разрешить завершение линии данных в порту OTG.

Если он установлен в 0, то выходы rpu_ena и rpd_ena привязаны к нулю.

Программное обеспечение может установить этот бит после обнаружения следующих условий после сброса питания:

– значение ID указывает на то, что ядро должно работать как B-устройство и изначально находится в периферийном режиме, и было обнаружено действительное питание шины;

– значение ID указывает на то, что ядро должно работать как A-устройство.

Программа может установить этот бит для переключения роли хоста/периферии в соответствии с протоколом Host Negotiation Protocol.

Подтягивающий резистор также может быть включен, если программа намерена запросить сеанс, используя импульсы линии данных (в соответствии с протоколом запроса сеанса).

Для этого необходимо, чтобы биты *VBUS Control Bit* и *VBUS Session Request Control Bit* указывали на выключенное состояние, а также приводило к установке выхода dlp_active.

Выбор завершения . С помощью этого бита программа может выбрать завершение линии данных в порту:

– b0: Завершение хоста. Если установлен бит *Termination Enable Bit*, это приведет к установке rpd_ena = 1 и rpu_ena = 0

[label = ”TERM _SEL -HOST”];

– b1: Завершение устройства. Если установлен бит *Termination Enable Bit*, это приведет к установке rpu_ena = 1 и rpd_ena = 0

[label = ”TERM _SEL _DEV”].

Программное обеспечение может установить этот бит после обнаружения следующих условий после сброса питания:

– значение ID указывает на то, что ядро должно работать как B-устройство и изначально находится в периферийном режиме;

– было обнаружено действительное питание шины.

Программа может установить этот бит для переключения роли хоста/периферии в соответствии с протоколом Host Negotiation Protocol.

Управление VBUS . С помощью этого бита программное обеспечение может управлять схемой питания (vb_on).

Если он установлен на 1, будет установлен вывод vb_on.

Программа **должна** установить этот бит, чтобы иметь возможность обнаружить подключенное устройство!

Приостановка порта . Это поле действительно, только если ядро работает в режиме хоста.

Бит может быть установлен программно, если драйвер устройства намерен перевести подключенное устройство в приостановленное состояние. Установка бита блокирует распространение данных по корневому порту. Если бит установлен, ядро чувствительно к обнаружению возобновления работы.

Бит также должен быть установлен, чтобы подготовить ядро к смене роли хоста/периферийного устройства. (Host Negotiation Protocol [Протокол переговоров с хостом]).

Установка бита *Force Resume Bit* автоматически зачищает этот бит.

Разрешить порт . Это поле действительно, только если ядро работает в режиме хоста.

Бит будет установлен автоматически аппаратными средствами после завершения программного сброса USB. Ядро начнет отправлять токены SOF, если подключенное устройство работает на полной скорости, или стробы поддержания активности на низкой скорости, если подключенное устройство работает на низкой скорости.

Принудительное возобновление . Если ядро работает в режиме хоста:

Бит может быть установлен программно, если подключенное устройство должно завершить режим приостановки. Бит будет установлен аппаратно, если обнаружено состояние удаленного пробуждения (Remote Wakeup) (установлен бит *Resume Event Bit*). В этом случае программное обеспечение должно завершить последовательность возобновления, установив этот бит на ноль. Бит будет оставаться установленным, пока последовательность возобновления не будет завершена аппаратным способом с помощью низкоскоростного EOP (на это указывает бит *Resume Complete Event Bit*).

Если ядро работает в периферийном режиме, бит может быть установлен программно, если он намерен разбудить подключенный к нему хост/хаб. ПО (Remote Wakeup) отвечает за время возобновления удаленного пробуждения и должно установить этот бит на время минимум на 1 мс, но не более 15 мс (см. п. 7.1.7.7 [1] спецификации USB).

Сброс USB . Это поле действительно, только если ядро работает в режиме хоста.

Когда программа записывает единицу в этот бит, ядро запускает последовательность сброса USB (состояние SE0). Запись нуля в этот бит завершает последовательность сброса. Может возникнуть задержка перед изменением состояния бита на ноль, поскольку аппаратное обеспечение очистит бит, если завершит последовательность сброса.

Программное обеспечение должно удерживать бит в состоянии единицы достаточно долго, как определено в спецификации USB 2.0, п. 7.3.2. (10…20 мс):

– b0: Не выполняется по сбросу USB

[label = ”URESET_INACTIVE”];

– b1: Ядро управляет сбросом USB

[label = ”URESET_ACTIVE”].

Контроль тестирования порта . Это битовое поле позволяет переводить порт в определенные тестовые режимы. Когда поле равно нулю, порт не работает в тестовом режиме. Кодировка тестового режима следующая:

– b00: Тестовый режим отключен

[label = "PTESTC_DIS"];

– b01: Тест J_State

[label = ”PTESTC_JST”];

– b10: Тест K_State

[label = ”PTESTC_KST”];

– b11: Тест SE0

[label = ”PTESTC_SE0”].

1.3.9.9.4.3.4.3 Регистр статуса изменения состояния порта

Идентификатор: PSC_EV

Адрес 0x034

В таблице 1.334 представлено описание полей регистра.

Таблица 1.334 – Регистр статуса изменения состояния порта

Имя поля	Поле	Состояние после сброса	Программный доступ	Описание
–	31: 12	–	–	Зарезервировано
VBSESS_END_EV	11	0x0	cow	Событие завершения сеанса VB
VSESS_VLD_EV	10	0x0	cow	Событие Vsession Valid
IDC_EV	9	0x0	cow	Событие изменения идентификатора
P_END_EV	8	0x0	cow	Событие окончания импульса
PWRSC_EV	7	0x0	cow	Событие изменения состояния питания
CDC_EV	6	0x0	cow	Событие подключения/отключения
URES_EV	5	0x0	cow	Событие сброса USB
SUSP_EV	4	0x0	cow	Событие приостановки
RSUC_EV	3	0x0	cow	Событие завершения возобновления
RSU_EV	2	0x0	cow	Событие возобновления
BERR_EV	1	0x0	cow	Событие ошибки помехи
OCU_EV	0	0x0	cow	Событие превышения тока

Событие завершения сеанса VB . Бит устанавливается, если изменился *VB Session End Bit*.

Событие Vsession Valid . Бит устанавливается, если изменился бит *Vsession Valid Bit*.

Событие изменения идентификатора. Если установлена единица, этот бит указывает, что бит *USB Connector ID Value Bit* был изменен.

Событие окончания импульса . Это поле действительно, только если ядро работает в периферийном режиме.

Бит устанавливается, если бит *Pulse Length Field* изменяется с единицы на ноль.

Событие изменения состояния питания . Бит устанавливается, если изменился *Vbus Valid Bit*.

Событие подключения/отключения . Это поле действительно, только если ядро работает в режиме хоста.

Если установлена единица, этот бит указывает, что бит *Port Connect Status Bit* изменился.

Событие сброса USB . Если ядро работает в режиме хоста, после того как программное обеспечение выполнит сброс USB, очистив бит *USB Reset Bit*, этот бит события будет установлен после завершения аппаратного сброса. (Ядро продолжает передавать сигнал сброса после отмены сигнала URESET до следующего начала фрейма).

Если ядро работает в периферийном режиме, этот бит устанавливается, если обнаружен сброс USB (линии данных превышают 2,5 мкс SE0).

Событие приостановки . Этот бит действителен, только если ядро работает в периферийном режиме.

Этот бит указывает на отсутствие активности на шине в течение более 3 мс. Программное обеспечение может использовать это событие для активации режима пониженного энергопотребления.

Если режим низкого энергопотребления приводит к тому, что тактовый генератор ядра USB удаляется, ограничивающее оборудование отвечает за включение контроллера тактового генератора, если на линиях данных USB подается сигнал о состоянии, отличном от состояния IDLE (ожидание). Для этого требуется дополнительная, асинхронная схема.

Если HNP не был включен ранее, программное обеспечение должно установить бит *Port Suspend Bit*, чтобы он был чувствителен к сигналу возобновления, и перейти в состояние низкого энергопотребления. Если HNP был разрешен, программное обеспечение теперь может переключиться в режим хоста в соответствии с правилами HNP.

Событие завершения возобновления . Если ядро работает в режиме хоста, этот бит будет установлен после того, как ядро завершит последовательность возобновления с помощью низкоскоростного EOP.

Если ядро работает в периферийном режиме, этот бит устанавливается, если последовательность возобновления была завершена хостом с помощью низкоскоростного EOP.

Событие возобновления . Если ядро работает в режиме хоста, этот бит устанавливается, если драйвер хоста перевел порт в приостановленное состояние и было обнаружено условие возобновления, вызванное подключенным устройством (удаленное пробуждение).

Если ядро работает в периферийном режиме, этот бит устанавливается, если было обнаружено условие возобновления. Согласно спецификации USB, это определяется как K-State длительностью более 2,5 мкс. Однако спецификация USB требует определять как условие возобновления также нормальный трафик. Ядро способно обнаружить как возобновление переход J-K.

Событие ошибки помехи . Это поле действительно, только если ядро работает в режиме хоста.

Этот бит становится равным единице, только если корневой порт отключен из-за соответствующих условий, существующих в точке EOF2.

Событие превышения тока . Этот бит будет установлен на единицу, если ядро обнаружило состояние перегрузки по току (см. бит *Over Current Condition Bit*).

1.3.9.9.4.3.4.4 Регистр масок изменения состояния порта

Идентификатор: PSC_EM

Адрес 0x038

В таблице 1.335 представлено описание полей регистра.

Таблица 1.335 – Регистр масок изменения состояния порта

Имя поля	Поле	Состояние после сброса	Программный доступ	Описание
–	31: 12	–	–	Зарезервировано
VBSESS_END_EM	11	0x0	rw	Маска события сеанса VB Session Valid
VSESS_VLD_EM	10	0x0	rw	Маска события Vsession Valid
IDC_EM	9	0x0	rw	Маска события ID
P_END_EM	8	0x0	rw	Маска события окончания импульса

Имя поля	Поле	Состояние после сброса	Программный доступ	Описание
PWRSC_EM	7	0x0	rw	Маска события изменения состояния питания
CDC_EM	6	0x0	rw	Маска события подключения/отключения
URES_EM	5	0x0	rw	Маска события сброса USB
SUSP_EM	4	0x0	rw	Маска события приостановки
RSUC_EM	3	0x0	rw	Маска события завершения возобновления
RSU_EM	2	0x0	rw	Маска события возобновления
BERR_EM	1	0x0	rw	Маска события ошибки помех
OCU_EM	0	0x0	rw	Маска события превышения тока

Маска события сеанса VB Session Valid . Установка этого бита приведет к прерыванию, если активен бит *VB Session End Event Bit*.

Маска события Vsession Valid . Установка этого бита приведет к прерыванию, если активен бит *Vsession Valid Event Bit*.

Маска события ID . Установка этого бита приведет к прерыванию, если активен бит *ID Change Event Bit*.

Маска события окончания импульса . Установка этого бита приведет к прерыванию, если активен бит *Pulse End Event Bit*.

Маска события изменения состояния питания . Установка этого бита приведет к прерыванию, если активен бит *Power Status Change Event Bit*.

Маска события подключения/отключения . Установка этого бита приведет к прерыванию, если активен бит *Connect/Disconnect Event Bit*.

Маска события сброса USB . Установка этого бита приведет к прерыванию, если активен бит *USB Reset Event Bit*.

Маска события приостановки . Установка этого бита приведет к прерыванию, если активен бит *Suspend Event Bit*.

Маска события завершения возобновления . Установка этого бита приведет к прерыванию, если активен бит *Resume Complete Event Bit*.

Маска события возобновления . Установка этого бита приведет к прерыванию, если активен бит *Resume Event Bit*.

Маска события ошибки помех . Установка этого бита приведет к прерыванию, если активен бит *Babble Error Event Bit*.

Маска события превышения тока . Установка этого бита приведет к прерыванию, если активен бит *Over Current Event Bit*.

1.3.9.9.4.3.5 Группа регистров канала

В этом разделе определены структуры данных, которые будут использоваться для обмена всей информацией о контроле/состоянии канала между аппаратным и программным обеспечением.

Если ядро работает в периферийном режиме, канал[0] зарезервирован для обработки управляющей передачи. Никакой другой канал не может быть использован для управляющей передачи. Поскольку канал управления определен как двунаправленный, ядро чувствительно к токенам для обоих направлений. Транзакция SETUP всегда будет принята с квитированием ACK, как того требует спецификация USB. Если буфер данных не доступен, то по умолчанию на токены IN или OUT будет отвечать NAK.

Все регистры каналов будут установлены в значения сброса после того, как ядро обнаружит сброс USB. Регистры каналов для pipe[0] имеют другие значения сброса.

Существуют некоторые дополнительные параметры DMA_BAW, TBW и AWM.

AWM (максимальная разрядность адреса ‖ Address Width Max) определяется из DMA_AW:

Если DW=16, AWM должна быть установлена на DMA_AW-1. Для DW=32 ее необходимо установить на DMA_AW-2.

TBW (суммарная ширина байтов ‖ Total Bytes Width) равен параметру DMA_AW, но имеет верхний предел 28.

DMA_BAW (ширина адреса буфера DMA ‖ DMA Buff Address Width) определяется из DMA_BW: DMA_BAW = DMA_BW–2.

1.3.9.9.4.3.5.1 Регистр конфигурации DMA канала

Правила доступа:

– этот регистр можно изменять, даже если канал активирован (бит *Activate Pipe Bit* установлен на 1). И этот регистр не должен изменяться, если буфер данных действителен (бит *Data Buffer Valid Bit* или *Alternative Data Buffer Valid Bit* установлен на 1);

– когда ядро сконфигурировано для использования двухпортовой RAM (DMA_CFG=1) следующее поле (поля) доступны только для чтения.

Идентификатор: PIPE_DMA_CFG

Адрес 0x03C

В таблице 1.336 представлено описание полей регистра.

Таблица 1.336 – Регистр конфигурации DMA канала

Имя поля	Поле	Состояние после сброса	Программный доступ	Описание
–	31: DMA_BAW	–	–	Зарезервировано
DMA_BADDR	DMA_BAW-1 : 0	0x0000	rw	Адрес буфера DMA

Адрес буфера DMA . Это поле действительно, только если ядро работает в периферийном режиме.

Это поле задает начальный адрес области RAM внутри локального буфера DMA.

1.3.9.9.4.3.5.2 Регистр управления каналом

Идентификатор: PIPE_CTRL

Адрес 0x040

В таблице 1.337 представлено описание полей регистра.

Таблица 1.337 – Регистр управления каналом

Имя поля	Поле	Состояние после сброса	Программный доступ	Описание
–	31: 3	–	–	Зарезервировано
ACT	2	0x0	rw	Активировать канал
TPID	1: 0	0x0	rw	PID токена

Активировать канал . Если ядро работает в режиме хоста, установка этого бита инициирует одну или несколько транзакций с заданной конфигурацией канала. Бит автоматически очищается, если передача завершена или возникла ошибка. Передача данных инициируется, только если программное обеспечение предоставляет действительное буферное пространство.

Если ядро работает в периферийном режиме, установка этого бита разрешает ответ на конечную точку с заданными параметрами канала. Если действительное буферное пространство не предоставлено (бит *Data Buffer Valid Bit* установлен в 0), конечная точка ответит

квитированием NAK. Бит будет автоматически установлен для канала[0] после обнаружения сброса USB. Для всех остальных каналов он будет автоматически очищен аппаратно после обнаружения сброса USB.

PID токена . Если ядро работает в режиме хоста, это поле указывает PID (и направление), используемый для отправляемого токена.

Если ядро работает в периферийном режиме, это поле указывает направление потока данных для этой конечной точки. Для направления приема битовое поле должно быть установлено в TPID_IN. Для направления передачи битовое поле должно быть установлено в TPID_OUT. Битовое поле будет автоматически установлено в TPID_SETUP для канала[0] после получения токена SETUP:

– b00: Передача OUT

[label = ”TPID_OUT”];

– b01: Передача IN

[label = ”TPID_IN”];

– b10: Передача SETUP

[label = ”TPID_SETUP”];

– b11: Зарезервировано.

Правила доступа - это поле может быть изменено, даже если канал активирован (бит *Activate Pipe Bit* установлен на 1). И это поле не должно изменяться, если буфер данных действителен (бит *Data Buffer Valid Bit* установлен на 1).

1.3.9.9.4.3.5.3 Регистр конфигурации канала

Правила доступа:

– если ядро работает в режиме хоста, содержимое этого регистра не должно изменяться, если бит *Activate Pipe Bit* или бит *Active Status Bit* равен 1;

– если ядро работает в периферийном режиме, содержимое этого регистра (за исключением бита *Stall Pipe Bit*) не должно изменяться, если бит *Activate Pipe Bit* равен 1.

Идентификатор: PIPE_CFG

Адрес 0x044

В таблице 1.338 представлено описание полей регистра.

Таблица 1.338 – Регистр конфигурации канала

Имя поля	Поле	Состояние после сброса	Программный доступ	Описание
–	31	–	–	Зарезервировано
IOT	30	0x0	rw	Прерывание по транзакции
HIDBE	29	0x0	rw	Остановка при ошибке буфера данных ISO
SKIPISO	28	0x0	rw	Пропуск токена ISO
PI	27: 24	0x0	rw	Интервал опроса
POFF	23: 16	0x00	rw	Смещение опроса
STALL	15	0x0	rw	Канал остановки
ACID	14	0x0	rw	Принять поврежденные данные ISO
EPS	13	0x0	rw	Скорость конечной точки
STRM	12	0x0	rw	Режим потоковой передачи
ET	11: 10	0x0	rw	Тип конечной точки
MPS	9: 0	0x000	rw	Максимальный размер пакета

Прерывание по транзакции . Это поле действительно, только если ядро работает в режиме хоста.

Если этот бит установлен, ядро будет подавать сигнал прерывания после каждой выполненной транзакции на этом канале, независимо от статуса транзакции. Это означает, что прерывание будет генерироваться и для NAK-транзакций или после возникновения ошибки (тайм-аут, ошибка CRC). Установка этого бита полезна, только если планирование транзакций должно осуществляться программно.

Остановка при ошибке буфера данных ISO . Это поле действительно, только если ядро работает в режиме хоста.

Если бит *Skip ISO Token Bit* не установлен и ядро посылает токен IN для изохронных каналов IN, а буфер не доступен, ядро остановит этот канал, когда активируется бит HIDBE. Если HIDBE деактивировано, ядро не изменит состояние канала.

Пропуск токена ISO . Это поле действительно, только если ядро работает в режиме хоста.

Когда этот бит установлен, HC не будет посылать токен IN/OUT для изохронных каналов, когда нет буферного пространства для данных. Если токен IN отправлен (бит *Skip ISO Token Bit* равен 0), а буферное пространство отсутствует, HC не будет записывать данные в

RAM. Полученные данные будут потеряны. Для каналов OUT HC отправит последовательность OUT-DATA0 в отсутствие буферного пространства.

Интервал опроса . Это поле действительно, только если ядро работает в режиме хоста.

Это значение определяет интервал опроса для этого канала как 2^{PI}, если конечная точка сконфигурирована как конечная точка прерывания или изохронная конечная точка. Интервалы опроса для конечных точек прерывания с полной/низкой скоростью кодируются следующим образом:

– b0000: 1 фрейм;

– b0001: 2 фрейма;

– b0010: 4 фрейма;

– : : :

– b1000: 256 фреймов.

PI должен быть ≤8. Установка PI в значение больше 8, приведет к неопределенным результатам.

Смещение опроса . Это поле действительно, только если ядро работает в режиме хоста и для конечных точек прерывания.

Определяет смещение фрейма интервала опроса, на который запланирован канал. Канал опрашивается, если PI_OFF[PI-1:0] совпадает с текущим фрагментом номера фрейма FRM_NR[PI-1:0]. $(8 \geq PI \geq 1)$

Канал остановки . Этот бит действителен, только если ядро работает в периферийном режиме.

Установка этого бита приведет к ответу STALL на конечную точку с заданными параметрами канала. Бит будет автоматически очищен для канала[0] после обнаружения сброса USB или получения токена SETUP.

Принять поврежденные данные ISO . Когда этот бит установлен, ядро принимает и поврежденные данные, передаваемые по каналам изохронного приема. Пакеты с ошибкой CRC принимаются полностью. Пакеты с ошибкой подстановки битов (bitstuff) принимаются до тех пор, пока не произойдет ошибка bitstuff.

Скорость конечной точки . Это поле действительно, только если ядро работает в режиме хоста.

Указывает скорость работы функции с этой конечной точкой:

– b0: Полная скорость

[label = ”EPS_FULL”];

– b1: Низкая скорость

[label = ”EPS_LOW”].

Если устройство, подключенное к корневому порту, работает на полной скорости, а бит установлен на низкую скорость, автоматически будет сгенерирован токен PRE.

Режим потоковой передачи . Этот бит действителен, только для конечных точек/каналов приема. Этот бит следует использовать для конечных точек/каналов периодического приема с высокой пропускной способностью.

Если этот бит активирован, то для этого канала используется потоковый режим:

– данные, полученные несколькими транзакциями шины, могут быть сохранены в одном буфере;

– выравнивание байтов между буферами выполняется таким образом, чтобы можно было сцеплять разные буферы, не сдвигая все данные на 1 или более байт.

Если этот бит отключен для этого канала:

– данные одной транзакции для этого канала хранятся в отдельном буфере;

– выравнивание байтов между буферами не производится.

Тип конечной точки . Указывает тип передачи для транзакций этого канала/конечной точки:

– b00: Управляющая передача

[label = ”ET_CTRL”];

– b01: Изохронная передача

[label = ”ET_ISO”];

– b10: Передача массива

[label = ”ET_BULK”];

– b11: Передача по прерыванию

[label = ”ET_INT”].

Максимальный размер пакета . Это поле определяет максимальное количество байт в пакете данных, которое может быть отправлено на конечную точку этого канала или получено с нее.

Если ядро работает в режиме хоста и если будет получено больше байт, а ядро сконфигурировано для работы в режиме хоста, будет сгенерирована ошибка помехи.

Если ядро работает в периферийном режиме, битовое поле будет игнорироваться, если полученная транзакция была транзакцией SETUP (транзакции SETUP должны приниматься всегда).

1.3.9.9.4.3.5.4 Регистр адреса канала

Правила доступа: идентичен регистру *Pipe Configuration Register* (Регистр конфигурации канала).

Идентификатор: PIPE_ADDR

Адрес 0x048

В таблице 1.339 представлено описание полей регистра.

Таблица 1.339 – Регистр адреса канала

Имя поля	Поле	Состояние после сброса	Программный доступ	Описание
–	31: 11	–	–	Зарезервировано
EPADDR	10: 4	0x00	rw	Адрес конечной точки
EPNR	3: 0	0x0	rw	Номер конечной точки

Адрес конечной точки . Это поле определяет адрес USB-устройства этого канала.

Оно должно быть задано как для режима хоста, так и для периферийного режима. При работе в периферийном режиме это поле определяет адрес устройства. Программа должна установить это поле только для канала[0] после успешного завершения запроса SetAddress (установка адреса). Значение этого поля для канала[0] используется для всех остальных каналов. Оно будет автоматически сброшено на 0x00, если обнаружен сброс USB.

Номер конечной точки . Указывает номер конечной точки этого канала/конечной точки.

Это поле должно быть установлено как для режима хоста, так и для режима периферии. Если ядро работает в периферийном режиме, битовое поле будет автоматически установлено на 0x0 для канала[0].

1.3.9.9.4.3.5.5 Регистр состояния канала

Этот регистр содержит полную информацию о состоянии выбранного канала и может быть изменен аппаратно.

Правила доступа - этот регистр можно изменять, даже если канал активирован (бит *Activate Pipe Bit* установлен на 1). И этот регистр не должен изменяться, если буфер данных действителен (бит *Data Buffer Valid Bit* установлен на 1).

Идентификатор: PIPE_STAT

Адрес 0x04C

В таблице 1.340 представлено описание полей регистра.

Таблица 1.340 – Регистр состояния канала

Имя поля	**Поле**	**Состояние после сброса**	**Программный доступ**	**Описание**
–	31: 16	–	–	Зарезервировано
TDIR_DIFF	15	0x0	r	Разница в направлении передачи EP0
ERS	14: 12	0x0	r	Причина ошибки
DMA_ERROR	11	0x0	cow	Ошибка DMA
DMA_BVLD	10	0x0	r	Буфер DMA действителен
CERR	9: 8	0x3	rw	Счетчик ошибок
DBERR	7	0x0	r	Ошибка буфера данных
ACTS	6	0x0	r	Активное состояние
HALT	5	0x0	r	Канал остановлен
BBL	4	0x0	r	Обнаружена помеха
DBSEL	3	0x0	rw	Выбранный буфер данных
DT	2	0x0	rw	Переключение данных
DBOFF	1: 0	0x0	rw	Смещение байта данных

Разница в направлении передачи EP0 . Этот бит действителен, только если ядро работает в режиме устройства, и только для нулевой конечной точки управления.

Этот бит устанавливается аппаратно, если сконфигурированное направление конечной точки (TPID) не совпадает с полученным направлением конечной точки (IN/OUT). В этом случае ядро посылает квитирование NAK и устанавливает DBV в 0. Это индикатор того, что этап передачи данных завершен и хост запускает этап состояния.

Бит не устанавливается, если получен токен SETUP, DBV равен 0 или канал остановлен.

Причина ошибки . Это поле показывает причину события остановки канала:

– b000: Ошибок не обнаружено .

[label = ”ERS_NO_ERROR”];

– b001: Получен токен остановки .

[label = ”ERS_STALL”];

– b010: Обнаружена помеха устройства.

[label = ”ERS_BABBLE”];

– b011: Обнаружен тайм-аут .

[label = ”ERS_TIMEOUT”];

– b100: Обнаружена ошибка BadXfer (неправильный PID) .

[label = ”ERS_BADXFER”];

– b101: Обнаружена ошибка подстановки битов .

[label = ”ERS_BS_ERROR”];

– b110: Обнаружена ошибка CRC .

[label = ”ERS_CRC_ERROR”];

Бит будет обновляться аппаратно, если происходит событие канала.

Ошибка DMA . Этот бит действителен, только когда ядро сконфигурировано на использование порта DMA для передачи данных (DMA – CFG=0).

Этот бит устанавливается аппаратно, когда во время передачи данных по этому каналу на интерфейсе DMA сообщается о состоянии ошибки. В этом случае ядро очищает биты *Data Buffer Valid Bit* и *Alternative Data Buffer Valid Bit*. Таким образом, никакая дальнейшая передача данных, относящихся к этому каналу, происходить не будет, пока программное обеспечение не обработает это событие ошибки.

Если подключенная шина DMA имеет тип AHB, то этот бит устанавливается в результате реакции на ошибку.

Буфер DMA действителен . Этот бит действителен только в периферийном режиме и ядро сконфигурировано на использование порта DMA для передачи данных (DMA_CFG=0).

Этот бит устанавливается аппаратно, когда буфер DMA этого канала содержит действительные данные. Этот бит будет установлен на 1, если:

– для данных приема - данные были получены через USB и будут записаны через интерфейс DMA как можно скорее;

– для данных передачи - данные были предварительно получены диспетчером DMA и будут отправлены после следующего токена IN, адресованного этому каналу.

Счетчик ошибок . Это поле действительно, только если ядро работает в режиме хоста.

Показывает состояние 2-битного обратного счетчика, который отслеживает количество обнаруженных последовательных ошибок. Ядро уменьшает счетчик, если транзакция завершилась неудачно. Если счетчик досчитает от единицы до нуля, канал будет остановлен и установит бит *Pipe Halted Bit*. Когда бит *Activate Pipe Bit* регистра *Pipe Control Register* переходит из состояния 0x0 в состояние 0x1, ядро сбрасывает это битовое поле.

Ошибка буфера данных . Этот бит действителен, только если ядро работает в режиме хоста.

Этот бит устанавливается аппаратно для изохронных каналов IN, когда ядро отправило токен IN, хотя буфер не доступен. В этом случае полученные данные будут потеряны.

Активное состояние . Это поле действительно, только если ядро работает в режиме хоста.

Этот бит показывает текущее состояние канала. Он будет установлен после того, как бит *Activate Pipe Bit* будет установлен на 1. Он будет очищен, если бит *Activate Pipe Bit* был установлен на 0 или канал был остановлен из-за нескольких событий передачи (остановка канала, помеха [babble], короткий пакет для каналов управления/массивов и т.д.).

Канал остановлен . Это поле действительно, только если ядро работает в режиме хоста.

Устанавливается на 1 ядром во время обновления состояния, если ядро работает в режиме хоста, чтобы указать, что на устройстве/конечной точке произошла серьезная ошибка. Это может быть вызвано ошибками помех, обратным отсчетом счетчика ошибок до нуля или приемом квитирования STALL от устройства. Каждый раз, когда в результате транзакции этот бит устанавливается на единицу, бит *Activate Pipe Bit* также устанавливается на 0.

Когда бит *Activate Pipe Bit* регистра *Pipe Control Register* переходит из состояния 0x0 в состояние 0x1, ядро сбрасывается на этот бит.

Обнаружена помеха . Это поле действительно, только если ядро работает в режиме хоста.

Устанавливается в 1 контроллером хоста при обновлении состояния, если во время транзакции обнаружена помеха (babble). В дополнение к установке этого бита аппаратное обеспечение также устанавливает бит *Pipe Halted Bit* на 1 (только для конечных точек IN). Когда бит *Activate Pipe Bit* регистра *Pipe Control Register* переходит из состояния 0x0 в состояние 0x1, ядро сбрасывается на этот бит.

Выбранный буфер данных . Этот бит действителен, только если канал сконфигурирован для использования альтернативного буфера данных. В противном случае бит застрянет на нуле.

Бит указывает, какой буфер в текущий момент обрабатывается аппаратно:

– b0: (Нормально) Обрабатывается буфер данных .

[label = ”DBSEL_ NORMAL”];

– b1: Обрабатывается альтернативный буфер данных .

[label = ”DBSEL_ALTERNATE”].

Бит будет обновляться аппаратно, если контроллер данных переключается между буферами данных. Бит также может быть изменен программно.

Переключение данных . Этот бит показывает текущее переключение данных. Он действителен для всех массовых и управляющих передач и передач по прерыванию. Этот бит будет изменен аппаратно в соответствии с текущим состоянием переключения данных. Бит также может быть изменен программно:

– b0: Значение переключения данных DATA0 .

[label = ”DT_DATA0”];

– b1: Значение переключения данных DATA1 .

[label = ”DT_DATA1”].

Если ядро работает в периферийном режиме, когда получен токен SETUP, бит автоматически устанавливается в DT_DATA1 для канала[0].

Смещение байта данных . Для передаваемых данных в этом поле указывается текущий байт данных, который должен быть отправлен в слове данных. Для принимаемых данных это поле указывает на первый байт в слове данных, в котором должен быть сохранен первый принятый байт.

Если параметр ядра DW (разрядность данных ‖ Data Width) установлен на 16, действителен только младший бит этого битового поля. Это битовое поле может быть изменено программно:

– для каналов передачи: это битовое поле отслеживает текущее смещение байта для этого канала, оно устанавливается аппаратно на 0 после того, как буфер становится недействительным;

– для каналов потокового приема: это битовое поле отслеживает текущее смещение байта для этого канала. Если канал работает в потоковом режиме (см. *Streaming Mode Bit*), это

байтовое смещение также используется, чтобы исключить программные операции сдвига байтов. Программное обеспечение должно только инициализировать смещение байта в начале передачи;

– для непотоковых каналов приема: это битовое поле отслеживает текущее смещение байта для этого канала, оно устанавливается аппаратно на 0 после того, как буфер становится недействительным.

Если ядро работает в периферийном режиме, битовое поле будет автоматически установлено на 0x0 для канала[0], если будет получена транзакция SETUP.

1.3.9.9.4.3.5.6 Регистр указателя данных канала

Содержимое этого регистра может быть изменено аппаратно.

Правила доступа - содержимое этого регистра не должно изменяться, если разрешен доступ к регистру *Pipe Configuration Register* и активен бит *Data Buffer Valid Bit*.

Идентификатор: PIPE_DATA_PTR

Адрес 0x050

В таблице 1.341 представлено описание полей регистра.

Таблица 1.341 – Регистр указателя данных канала

Имя поля	Поле	Состояние после сброса	Программный доступ	Описание
–	31: AWM	–	–	Зарезервировано
DPTR	AWM-1 : 0	0x00	rw	Указатель данных

Указатель данных . Для передаваемых данных это поле указывает на первый байт данных к отправке. Для принимаемых данных это поле указывает на адрес, в котором должен быть сохранен первый принятый байт. Ширина этого поля зависит от ширины адреса используемого RAM. Важно отметить, что это битовое поле представляет собой вещественный DATAWORD POINTER (Указатель на слово данных). Это означает, что полный адрес вычисляется путем добавления к этому битовому полю действительных битов DBOFF.

Если ядро работает в периферийном режиме, битовое поле будет автоматически установлено на содержимое адресного поля *DMA Setup Address Field* для канала[0], если будет получена транзакция SETUP.

1.3.9.9.4.3.5.7 Регистр общего числа байтов данных канала

Содержимое этого регистра может быть изменено аппаратно.

Правила доступа - содержимое этого регистра не должно изменяться, если разрешен доступ к регистру *Pipe Configuration Register* и активен бит *Data Buffer Valid Bit*.

Идентификатор: PIPE_DATA_TBYTES

Адрес 0x054

В таблице 1.342 представлено описание полей регистра.

Таблица 1.342 – Регистр общего числа байтов данных канала

Имя поля	Поле	Состояние после сброса	Программный доступ	Описание
DBV	31	0x0	rw	Буфер данных действителен
–	30: TBW	–	–	Зарезервировано
TBYTES	TBW-1 : 0	0x00	rw	Всего байтов для передачи

Буфер данных действителен . Программа должна установить этот бит, чтобы указать, что данные, описанные регистрами чтобы указать, что данные, описанные регистрами *Pipe Data Pointer Register* и *Pipe Data Total Bytes Register*, действительны, соответственно буфер действителен для входящих данных. Этот бит будет очищен аппаратно после завершения работы с буфером (например, все данные были переданы или приняты).

Всего байтов для передачи . Это битовое поле содержит общее количество передаваемых или принимаемых байтов.

Это поле будет уменьшаться в соответствии с количеством принятых/переданных байтов при каждой транзакции.

Программа должна убедиться, что это поле записывается, только если буфер данных недействителен. Запись в этот регистр, если данные подтверждены, приведет к неопределенным результатам.

Для каналов приема программа должна всегда устанавливать значение этого поля не менее чем на (например, кратное) *Maximum Packet Size Field*. Передача данных на прием будет начата, только если TBYTES ≥ MPS. Если передача на прием завершена, а оставшееся значение TBYTES меньше *Maximum Packet Size Field*, произойдет событие канала, и больше передачи инициироваться не будут. Обработка нулевых пакетов зависит от режима передачи.

Если ядро работает в периферийном режиме, если полученная транзакция была транзакцией SETUP, битовое поле будет проигнорировано аппаратурным обеспечением (транзакции SETUP должны приниматься всегда).

1.3.9.9.4.3.5.8 Регистр альтернативного указателя данных канала

Содержимое этого регистра может быть изменено аппаратно.

Этот регистр существует, только если для этого канала включена поддержка альтернативных буферов. В противном случае это битовое поле застрянет на нуле.

Правила доступа -содержимое этого регистра не должно изменяться, если разрешен доступ к регистру *Pipe Configuration Register* и активен бит *Alternative Data Buffer Valid Bit*.

Идентификатор: PIPE_ADATA_PTR

Адрес 0x058

В таблице 1.343 представлено описание полей регистра.

Таблица 1.343 – Регистр альтернативного указателя данных канала

Имя поля	Поле	Состояние после сброса	Программный доступ	Описание
–	31: AWM	–	–	Зарезервировано
ADPTR	AWM-1 : 0	0x00	rw	Альтернативный указатель данных

Альтернативный указатель данных . Этот регистр существует, только если для этого канала включена поддержка альтернативных буферов.

В противном случае это битовое поле застрянет на нуле.

Для передаваемых данных это поле указывает на первый байт альтернативного буфера данных к отправке. Для принимаемых данных это поле указывает на адрес, в котором должен быть сохранен первый принятый байт. Ширина этого поля зависит от ширины адреса используемого RAM.

1.3.9.9.4.3.5.9 Альтернативный регистр общего числа байтов данных канала

Содержимое этого регистра может быть изменено аппаратно.

Этот регистр существует, только если для этого канала включена поддержка альтернативных буферов. В противном случае это битовое поле застрянет на нуле.

Правила доступа - содержимое этого регистра не должно изменяться, если разрешен доступ к регистру *Pipe Configuration Register* и активен бит *Alternative Data Buffer Valid Bit*.

Идентификатор: PIPE_ADATA_TBYTES

Адрес 0x05C

В таблице 1.344 представлено описание полей регистра.

Таблица 1.344 – Альтернативный регистр общего числа байтов данных канала

Имя поля	Поле	Состояние после сброса	Программный доступ	Описание
ADBV	31	0x0	rw	Альтернативный буфер данных действителен .
–	30: TBW	–	–	Зарезервировано
ATBYTES	TBW-1 : 0	0x00	rw	Альтернативное число байтов к передаче

Альтернативный буфер данных действителен .. Этот бит действителен, только если канал сконфигурирован для использования альтернативного буфера данных. В противном случае бит застрянет на нуле.

Программа должна установить этот бит, чтобы указать, что данные, описанные регистрами *Pipe Alternative Data Pointer Register* и *Pipe Alternative Data Total Bytes Register*, действительны, соответственно буфер действителен для входящих данных. Этот бит будет очищен аппаратно:

– после завершения работы буфера приема это условие проверяется во время обновления состояния. Буфер будет завершен, если поле *Total Bytes To Transfer Field* будет меньше поля *Maximum Packet Size Field* и бит *Selected Data Buffer Bit* будет равен 1;

– после завершения работы буфера передачи это условие проверяется во время обновления состояния. Буфер будет завершен, если поле *Total Bytes To Transfer Field* равно 0x0 и бит *Selected Data Buffer Bit* будет равен 1;

– состояние канала (*Active Status Bit*) изменилось с активного на неактивное, и ядро работает в режиме хоста.

Альтернативное число байтов к передаче . Содержит общее количество байтов альтернативного буфера данных, которые будут переданы или получены.

Это поле будет уменьшаться в соответствии с количеством принятых/переданных байтов при каждой транзакции.

Программа должна убедиться, что это поле записывается, только если альтернативный буфер данных недействителен. Запись в этот регистр, если данные подтверждены, приведет к неопределенным результатам.

1.3.9.9.4.3.6 Группа отладочных регистров

Эти регистры предоставляют различные возможности отладки. Они не используются для нормальной работы. Режим отладки включается, если один из битов *Debug Control Register* установлен на единицу.

Режим отладки действителен только для канала 0. Базовая конфигурация этого канала будет использоваться для транзакции и изменяться в зависимости от конфигурации отладки.

Ядро может быть сконфигурировано для поддержки отладки или расширенных возможностей отладки. Поэтому некоторые регистры или биты могут быть недействительны (застрять в нуле).

1.3.9.9.4.3.6.1 Регистр управления отладкой

Режим отладки включается, если один из управляющих битов установлен на единицу.

Идентификатор: DEBUG_CTRL

Адрес 0x060

В таблице 1.345 представлено описание полей регистра.

Таблица 1.345 – Регистр управления отладкой

Имя поля	Поле	Состояние после сброса	Программный доступ	Описание
–	31: 11	–	–	Зарезервировано
FTXSOFE	10	0x0	rw	Принудительная передача ошибки SOF
UDTPID	9	0x0	rw	Использовать PID отладочного токена
UDHSPID	8	0x0	rw	Использовать PID отладочного квитирования
UDDPID	7	0x0	rw	Использовать PID отладочных данных
FRXCRC16G	6	0x0	rw	Принудительно получить хороший CRC16
FRXCRC5G	5	0x0	rw	Принудительно получить хороший CRC5
FRXCRCE	4	0x0	rw	Принудительно получить ошибку CRC
FTXCRC16E	3	0x0	rw	Принудительно передать ошибку CRC16
FTXCRC5E	2	0x0	rw	Принудительно передать ошибку CRC5

Имя поля	Поле	Состояние после сброса	Программный доступ	Описание
DBSTX	1	0x0	rw	Запретить передачу подстановки битов
DBSERRDET	0	0x0	rw	Запретить обнаружение ошибок подстановки битов

Принудительная передача ошибки SOF . Если этот бит установлен, то ядро будет генерировать неправильный CRC5 только для каждого SOF (дополнение к единицам).

Бит будет существовать, только если была реализована поддержка отладочных возможностей. В противном случае бит застрянет на нуле.

Использовать PID отладочного токена . Если бит установлен, ядро будет использовать поле *Debug Token PID Field* для следующей транзакции.

Бит будет существовать, только если была реализована поддержка расширенных возможностей отладки. В противном случае бит застрянет на нуле.

Использовать PID отладочного квитирования . Если бит установлен, то для следующей транзакции ядро будет использовать поле *Debug Handshake PID Field*.

Бит будет существовать, только если была реализована поддержка расширенных возможностей отладки. В противном случае бит застрянет на нуле.

Использовать PID отладочных данных . Если бит установлен, то для следующей транзакции ядро будет использовать поле *Debug Data PID Field* .

Бит будет существовать, только если была реализована поддержка расширенных возможностей отладки. В противном случае бит застрянет на нуле.

Принудительно получить хороший CRC16 . Если этот бит установлен, ядро будет рассматривать каждый CRC16 полученных данных как хороший CRC. Бит будет существовать, только если была реализована поддержка отладочных возможностей. В противном случае бит застрянет на нуле.

Принудительно получить хороший CRC5 . Если этот бит установлен, ядро будет рассматривать каждый полученный CRC5 входящего токена или SOF как хороший CRC.

Бит будет существовать, только если была реализована поддержка отладочных возможностей. В противном случае бит застрянет на нуле.

Принудительно получить ошибку CRC . Если этот бит установлен, ядро будет рассматривать каждый полученный CRC как неправильный CRC.

Бит будет существовать, только если была реализована поддержка отладочных возможностей. В противном случае бит застрянет на нуле.

Принудительно передать ошибку CRC16 . Если этот бит установлен, то ядро будет генерировать CRC16 (дополнение единицы) неправильных данных.

Бит будет существовать, только если была реализована поддержка отладочных возможностей. В противном случае бит застрянет на нуле.

Принудительно передать ошибку CRC5 . Если этот бит установлен, ядро будет генерировать неправильный CRC5 для токена, исключающего пакет SOF (дополнение к единицам).

Бит будет существовать, только если была реализована поддержка отладочных возможностей. В противном случае бит застрянет на нуле.

Запретить передачу подстановки битов . Если этот бит установлен, ядро не будет выполнять подстановку битов (bitstuffing) в выходном потоке.

Бит будет существовать, только если была реализована поддержка отладочных возможностей. В противном случае бит застрянет на нуле.

Запретить обнаружение ошибок подстановки битов . Если этот бит установлен, ядро не будет сообщать об ошибка подстановки битов (bitstuff errors). Поскольку каждая ошибка подстановки битов приводит к ошибке CRC, код завершения будет показывать только ошибки CRC, а не ошибки подстановки битов (bitstuff errors).

Бит будет существовать, только если была реализована поддержка отладочных возможностей. В противном случае бит застрянет на нуле.

1.3.9.9.4.3.6.2 Регистр PID отладки

Этот регистр будет существовать, только если была реализована поддержка расширенных возможностей отладки. В противном случае бит застрянет на нуле.

Идентификатор: DEBUG_PID

Адрес 0x064

В таблице 1.346 представлено описание полей регистра.

Таблица 1.346 – Регистр PID отладки

Имя поля	Поле	Состояние после сброса	Программный доступ	Описание
–	31: 24	–	–	Зарезервировано
DHSPID	23: 16	0x00	rw	PID квитирования отладки
DTPID	15: 8	0x00	rw	PID отладочного токена
DDPID	7: 0	0x00	rw	PID отладочных данных

PID квитирования отладки . Это битовое поле будет отправлено как PID квитирования, если устройство вернуло данные.

PID отладочного токена . Это битовое поле будет отправлено как PID токена, если включен режим отладки, вместо PID токена IN/OUT/SETUP.

PID отладочных данных . Это битовое поле будет отправлено как PID данных, если включен режим отладки, вместо PID DATA0, DATA1, DATA2 или MDATA.

1.3.9.9.4.3.6.3 Регистр статуса отладки

Этот регистр будет существовать, только если была реализована поддержка расширенных возможностей отладки. В противном случае бит застрянет на нуле.

Идентификатор: DEBUG_STAT

Адрес 0x068

В таблице 1.347 представлено описание полей регистра.

Таблица 1.347 – Регистр статуса отладки

Имя поля	Поле	Состояние после сброса	Программный доступ	Описание
–	31: 8	–	–	Зарезервировано
DRXPID	7: 0	0x00	r	Отладка PID приема

Отладка PID приема . Это битовое поле содержит полученный PID последней транзакции, независимо от статуса транзакции.

1.3.9.9.4.3.6.4 Тестовый регистр

Этот регистр существует только для тестов системной интеграции.

Идентификатор: TEST

Адрес 0x06C

В таблице 1.348 представлено описание полей регистра.

Таблица 1.348 – Тестовый регистр

Имя поля	Поле	Состояние после сброса	Программный доступ	Описание
RW32	31: 0	0x5AF8CAFE	rw	Чтение/запись 32

Чтение/запись 32 . Это тестовое поле можно читать и записывать в любом порядке и с любым содержимым. При каждом обращении к записи сохраняется дополнение 1 к записанному значению.

1.3.9.9.4.3.7 Группа регистров конфигурации ядра

Следующие регистры показывают параметры, с которыми сконфигурировано ядро. Более подробная информация о параметрах и сигналах конфигурации ядра приведена в главе 3. Все битовые поля отражают конфигурацию ядра, изменение конфигурации путём записи в эти регистры невозможно.

1.3.9.9.4.3.7.1 Основной регистр конфигурации

Идентификатор: MAIN_CFG

Адрес 0x080

В таблице 1.349 представлено описание полей регистра.

Таблица 1.349 – Основной регистр конфигурации

Имя поля	Поле	Состояние после сброса	Программный доступ	Описание
–	31: 20	–	–	Зарезервировано
DMA_CCFG	19	DMA_CFG	r	Канал DMA
XDBG_CFG	18	0x0	r	Конфигурация расширенной отладки
DBG_CFG	17	0x0	r	Конфигурация отладки
–	16: 12	–	–	Зарезервировано
DW_CFG	11: 6	DW	r	Конфигурация разрядности данных
NOP_CFG	5: 0	NOP	r	Конфигурация количества каналов

Конфигурация канала DMA . Этот бит показывает, настроено ли это ядро на использование канала DMA. См. конфигурационный параметр DMA_CFG:

– b0: *Нативный режим DMA* . Ядро использует канал DMA.

[label = ”DMA_ON”];

– b1: *Режим двухпортовой RAM* . Ядро использует интерфейс локального буфера оперативной памяти порта B для обмена данными с двухпортовым модулем RAM. Ядро имеет исключительный доступ к RAM.

[label = ”DMA_EXRAM”].

Конфигурация расширенной отладки . Этот бит показывает, настроено ли это ядро на расширенную поддержку отладки.

Конфигурация отладки . Этот бит показывает, настроено ли это ядро на поддержку отладки.

Конфигурация разрядности данных . Это битовое поле показывает разрядность данных (DW, data width) ядра.

Конфигурация количества каналов . Это поле показывает количество каналов (NOP), с которыми сконфигурировано ядро.

1.3.9.9.4.3.7.2 Регистр конфигурации режима

Идентификатор: MODE_CFG

Адрес 0x084

В таблице 1.350 представлено описание полей регистра.

Таблица 1.350 – Регистр конфигурации режима

Имя поля	Поле	Состояние после сброса	Программный доступ	Описание
–	31: NOP	–	–	Зарезервировано
ABUFF_CFG	NOP-1: 0	0x0000	r	Конфигурация альтернативного буфера

Конфигурация альтернативного буфера . Это битовое поле показывает, какой канал сконфигурирован для использования альтернативного буфера.

1.3.9.9.4.3.7.3 Регистр конфигурации DMA

Идентификатор: DMA_CONFIG

Адрес 0x088

В таблице 1.351 представлено описание полей регистра.

Таблица 1.351 – Регистр конфигурации DMA

Имя поля	Поле	Состояние после сброса	Программный доступ	Описание
–	31: 24	–	–	Зарезервировано
DMA_BCFG	23: 16	DMA_BW	r	Конфигурация ширины буфера DMA
DMA_ACFG	15: 8	DMA_AW	r	Конфигурация ширины адреса DMA
DMA_WCFG	7: 0	DMA_DW	r	Конфигурация разрядности данных DMA

Конфигурация ширины буфера DMA . Это поле показывает ширину локального буфера DMA.

Конфигурация ширины адреса DMA . В этом поле отображается ширина адреса канала DMA.

Конфигурация разрядности данных DMA . В этом поле отображается разрядность данных канала DMA.

1.3.9.9.4.3.8 Группа регистров управления DMA

Эти регистры управляют поведением интерфейса DMA. Программа может изменять содержимое регистров только в отсутствие USB-трафика. Когда ядро сконфигурировано на использование двухпортовой RAM (DMA_CFG=1), следующие поля доступны только для чтения.

1.3.9.9.4.3.8.1 Регистр конфигурации DMA RX

Идентификатор: DMA_RX_CFG

Адрес 0x0A0

В таблице 1.352 представлено описание полей регистра.

Таблица 1.352 – Регистр конфигурации DMA RX

Имя поля	Поле	Состояние после сброса	Программный доступ	Описание
–	31: 3	–	–	Зарезервировано
DMA_RX_LEN	2: 0	0x1	rw	Длина пакета приема DMA

Длина пакета приема DMA . Это поле управляет размером операций записи DMA во время или после операций приёма USB. Минимальный размер пакета составляет 16. Это поле определяет количество байт, передаваемых при каждой операции DMA. Таким образом, количество тактов для каждого пакета зависит от параметра DMA_DW. Например, для DMA_DW=32 размер пакета 16 приведет к 4-битовым пакетам. Поле кодируется следующим образом:

– 0x0: резерв;

– 0x1: размер пакета = 16;

– 0x2: размер пакета = 32;

– 0x3: размер пакета = 64;

– : : :;

– 0x6: размер пакета = 512;

– 0x7: резерв.

1.3.9.9.4.3.8.2 Регистр конфигурации DMA TX

Идентификатор: DMA_TX_CFG

Адрес 0x0A4

В таблице 1.353 представлено описание полей регистра.

Таблица 1.353 – Регистр конфигурации DMA TX

Имя поля	Поле	Состояние после сброса	Программный доступ	Описание
–	31: 27	–	–	Зарезервировано
DMA_TX_LVL	26: 16	0x14	rw	Уровень предварительной выборки DMA
–	15: 4	–	–	Зарезервировано
DMA_FLUSH	3	0x0	rw	Выравнивание полей диспетчера DMA
DMA_TX_LEN	2: 0	0x1	rw	Длина пакета передачи DMA

Уровень предварительной выборки DMA . Это поле действительно, только если ядро работает в режиме хоста.

Это поле контролирует количество байтов, которые должны быть предварительно выбраны перед началом транзакции USB. Планировщик начинает операцию передачи по USB, когда указанное количество операций чтения DMA успешно завершено или все данные к передаче по USB извлечены. Заданное количество байт должно быть меньше 0x400 – DMA_DW/8 и больше 0x14, указанного в поле *DMA Transmit Burst Length Field*. Примечание. При использовании медленных DMA-интерфейсов это поле должно быть установлено на максимальный размер пакета. В противном случае возможно недостаточное наполнение буфера, и ядро инвертирует CRC, делая данные недействительными.

Выравнивание полей диспетчера DMA . Этот бит должен использоваться только в периферийном режиме.

Когда этот бит установлен на 1, диспетчер DMA больше не будет осуществлять предварительную выборку передаваемых данных. Принятые данные обрабатываются дополнительно.

Задний фронт этого бита сбрасывает все внутренние состояния диспетчера DMA.

Длина пакета передачи DMA . Это поле управляет размером операций чтения DMA перед или во время операций передачи USB. Кодировка такая же, как и для поля *DMA Receive Burst Length Field*.

1.3.9.9.4.3.8.3 Регистр конфигурации устройства DMA

Идентификатор: DMA_DEV_CFG

Адрес 0x0A8

В таблице 1.354 представлено описание полей регистра.

Таблица 1.354 – Регистр конфигурации устройства DMA

Имя поля	Поле	Состояние после сброса	Программный доступ	Описание
–	31: DMA_AW	–	–	Зарезервировано
DMA_SADDR	DMA_AW-1 : 0	0x0000	rw	Адрес настройки DMA

Адрес настройки DMA . Если ядро работает в режиме устройства, это поле определяет местоположение буфера данных для входящих данных SETUP. Соответствующий размер буфера фиксирован и составляет 8 байт. Программа должна обеспечить инициализацию этого поля после завершения передачи сигнала сброса USB. Соответствующие временны́е требования см. в спецификации USB 2.0, п. 7.1.7.5.

1.3.9.9.5 Примечания к программному обеспечению

1.3.9.9.5.1 Управление транзакциями USB

В этом разделе описывается управление транзакциями USB в ядре USB_FS. Как правило, этим занимается модуль диспетчера транзакций. (*TM*)

USB_FS может быть сконфигурирован для поддержки до 16 конечных точек/каналов. Модуль регистров-файлов дает отдельный набор регистров для каждой конечной точки, как показано на рисунке 1.86 Доступ к этим регистровым ресурсам осуществляется «постранично». Это означает, что программа выбирает соответствующую страницу регистра канала путем записи в поле *Pipe Select Field*. Каждая страница канала содержит полный набор битов/полей со всей необходимой управляющей и статусной информацией (включая указатели и длину буфера данных) для обработки транзакций USB.

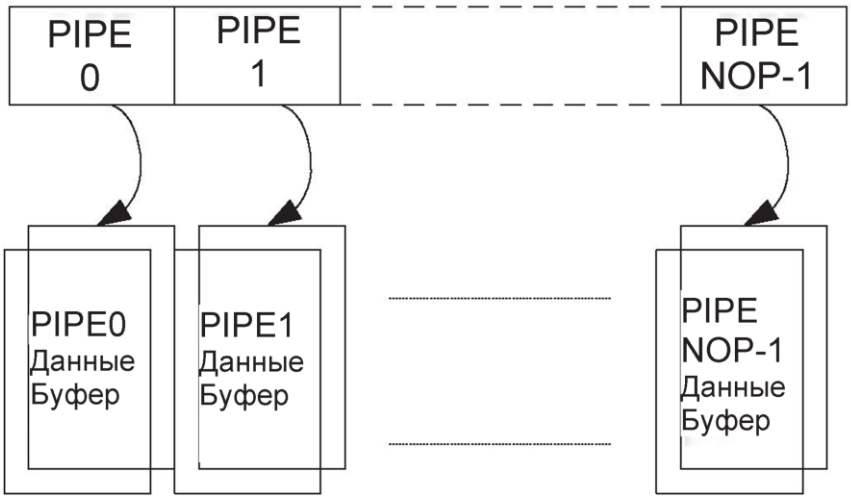


Рисунок 1.86 – Структура страницы канала

Все операции по планированию выполняются аппаратно. После того как программа сконфигурировала соответствующие регистры канала, ядро USB_FS планирует транзакции USB, оценивает ответы устройства и выполняет обновление состояния. Программа должна управлять потоком данных и реагировать на серьезные ошибки.

1.3.9.9.5.1.1 Режим хоста

Если ядро работает в режиме хоста, Диспетчер транзакций (Transaction Manager, TM) управляет генерацией токенов SOF или низкоскоростных активных соединений, а также планированием различных типов передачи данных. TM проверяет доступную пропускную способность фрейма. Транзакции запускаются только при наличии достаточной пропускной способности. Периодические передачи планируются, если активен бит *Periodic Schedule Bit* . Асинхронные передачи планируются только при активном бите *Asynchronous Schedule Bit*.

TM инициирует различные типы передачи в порядке, показанном на рисунке 1.87. Для изохронных передач TM сканирует все *Pipe Registers Pages*, начиная с индекса 0. Когда он достигает индекса канала (NOP-1), он переключается на тип передачи по прерыванию и снова выполняет сканирование. Наконец, обрабатываются массовые и управляющие конечные точки.

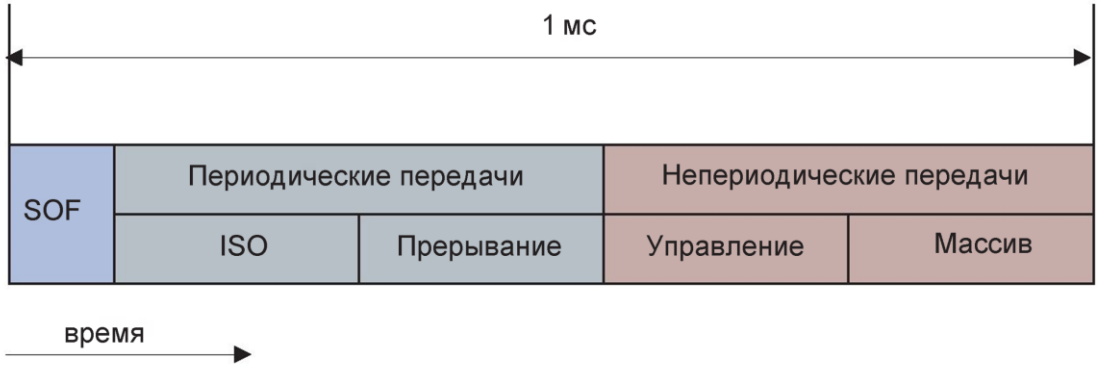


Рисунок 1.87 – Планирование передач

Периодические передачи (прерывания и изохронные) обрабатываются автоматически. Изохронные и управляющие передачи инициируются один раз на фрейм и канал. Передачи по прерыванию планируются в соответствии с заданным интервалом опроса и индексом расписания.

Вся оставшаяся полоса пропускания используется для массовых транзакций. Массовые каналы могут быть запланированы более одного раза на фрейм.

TM запускает USB-транзакции и выполняет обновление состояния канала в соответствии с правилами, определенными на рисунках 1.88, 1.89, 1.90, 1.91, 1.92.

Транзакция будет выполнена, если *PreCondition* (предусловие) истинно.

Обозначения блок-схемы транзакций. Для следующих блок-схем транзакций используются следующие обозначения:

– **Регистр.** Все поля регистров ядра обозначаются в верхнем регистре, например FRAME_NR. Имена полей группы *Pipe Register Group* обозначаются префиксом PIPE.*, например, PIPE.ACT означает ACT-бит регистра управления каналами (Pipe Control Register), входящего в группу регистров каналов (Pipe Register Group);

– **Внутренние переменные блок-схемы.** Все внутренние переменные, используемые в диаграмме, набраны курсивным шрифтом в смешанном регистре, например *PreCondition*;

– **Действия.** Действия, которые используются в схеме более одного раза, обозначаются как вызов функции, в смешанном регистре, курсивным шрифтом. Используются следующие виды действий:

– ***deactivatePipe()*** Бит *Pipe Active Bit* очищается аппаратно;

– ***haltPipe()*** Бит *Pipe Active Bit* очищается, а бит *Pipe Halted Bit* устанавливается аппаратно, чтобы сообщить программному обеспечению о серьезной ошибке;

– ***nextBuffer()*** Текущий буфер данных не может принять больше данных. Бит *Pipe (Alternative) Data Buffer Valid Bit* очищается аппаратно. Бит *Pipe Data Buffer Select Bit* переключается, если для этого канала включена поддержка альтернативных буферов;

– ***discardData()*** Все данные, полученные при последней передаче по USB, полностью игнорируются, а соответствующий указатель буфера и счетчик длины возвращаются;

– ***revertData()*** Передача данных при последней передаче USB не удалась. Таким образом, а соответствующий указатель буфера и счетчик длины должны вернуться.

Атрибуты состояния транзакции. Атрибуты статуса транзакций обозначаются как Rx.<атрибут> или Tx.<атрибут>, например, Rx.Len означает длину транзакции приема.

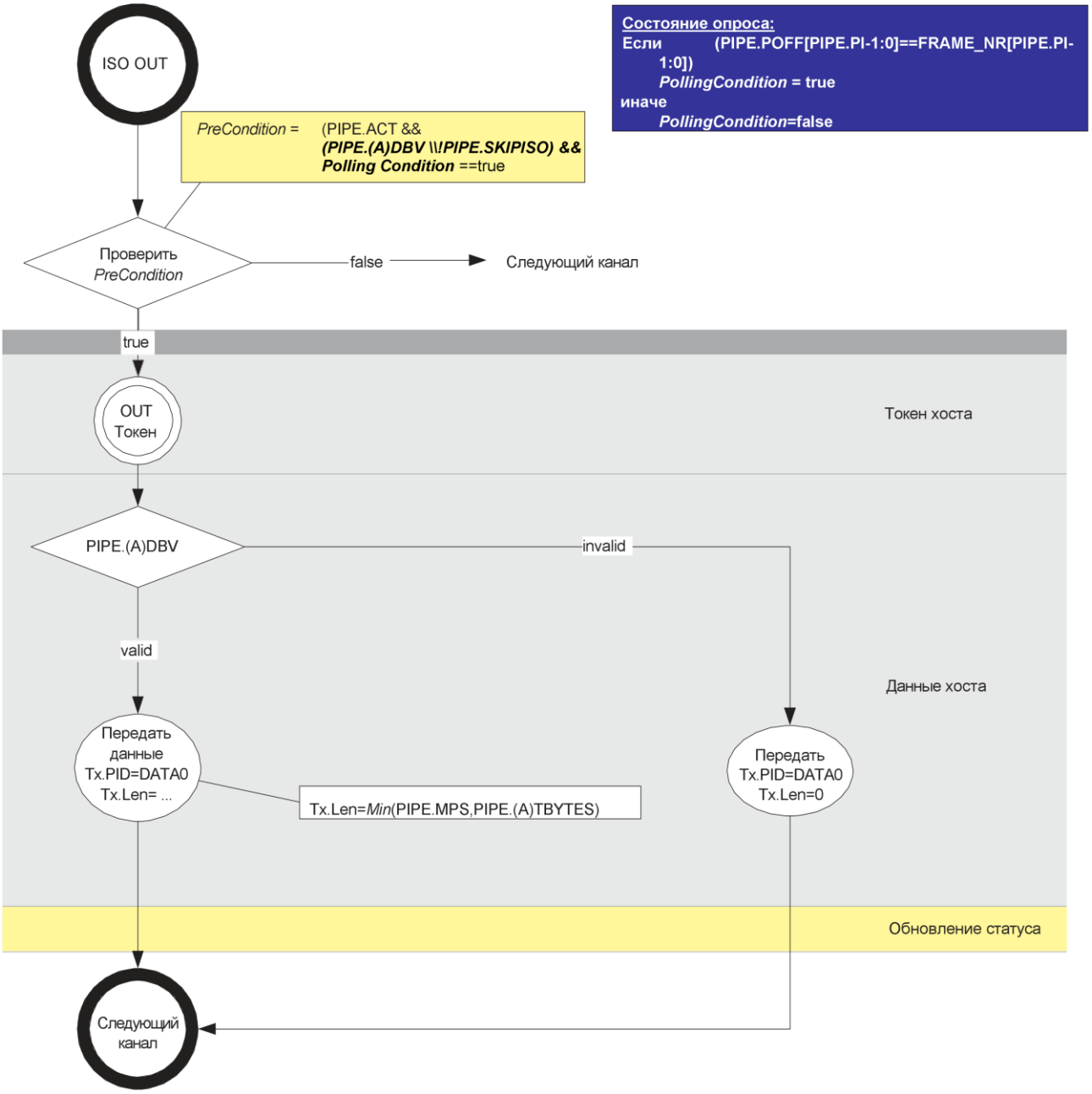


Рисунок 1.88 – Изохронный OUT хоста

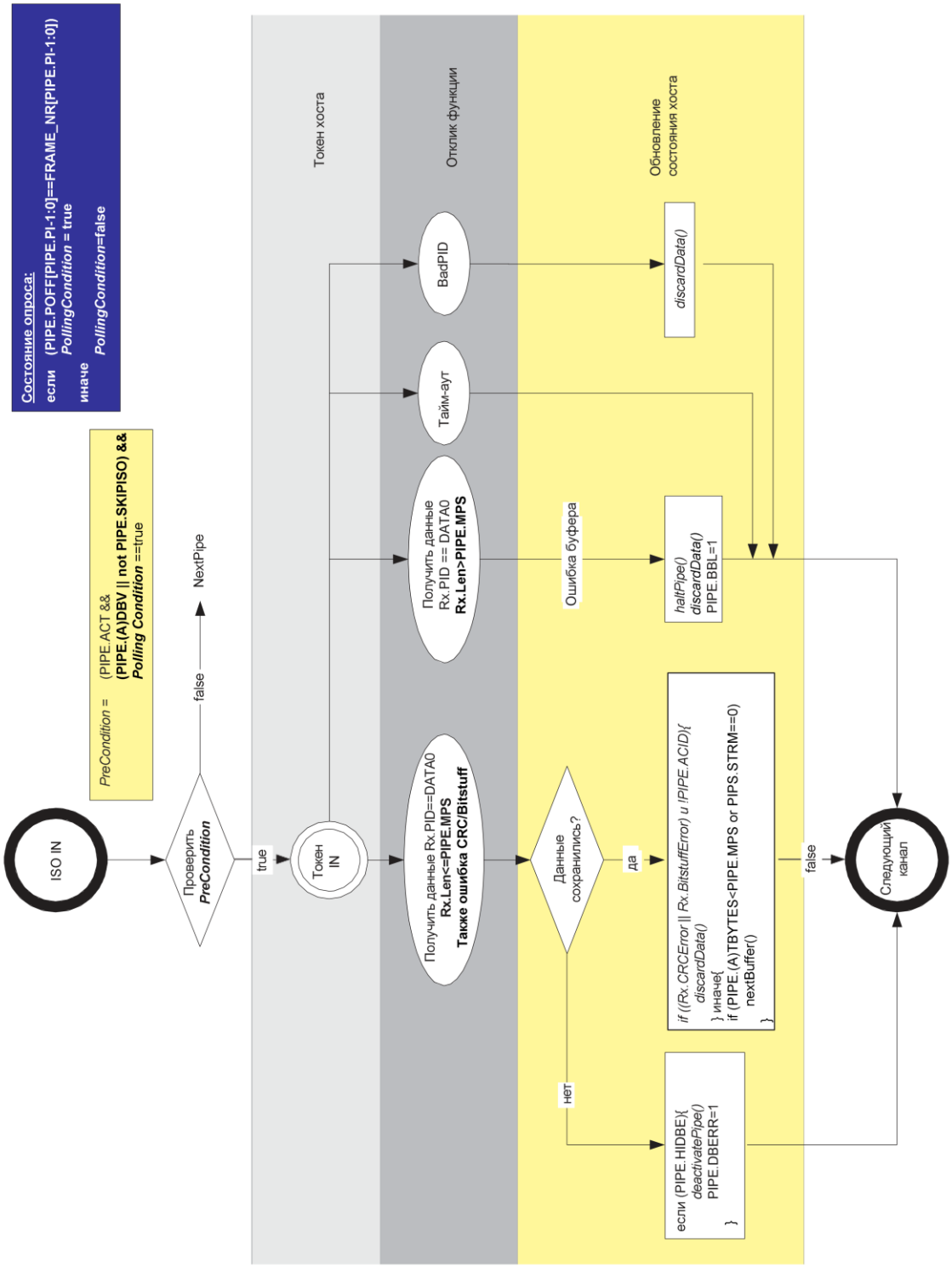


Рисунок 1.89 – Изохронный IN хоста

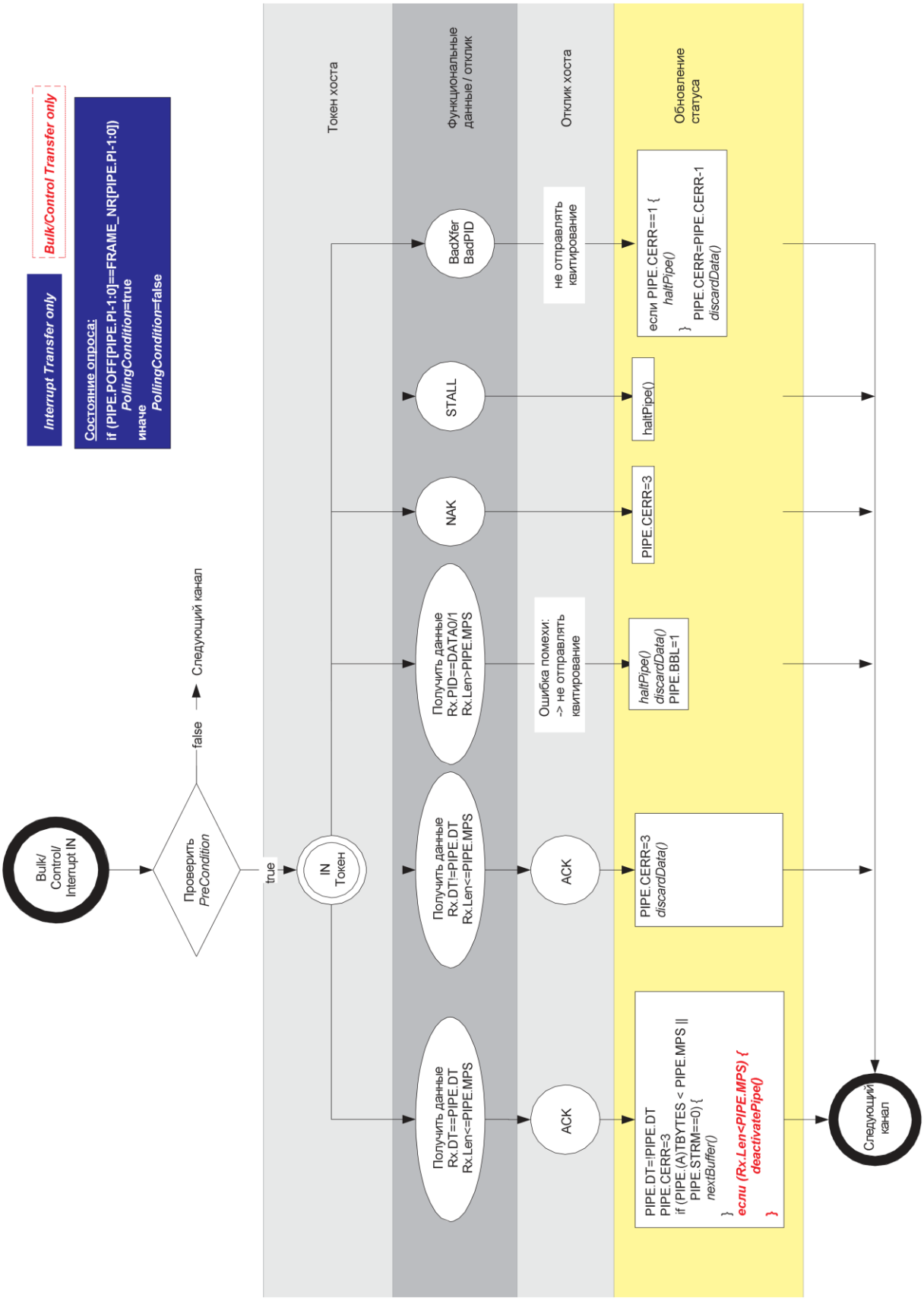


Рисунок 1.90 – Host Bulk/Control/Interrupt IN

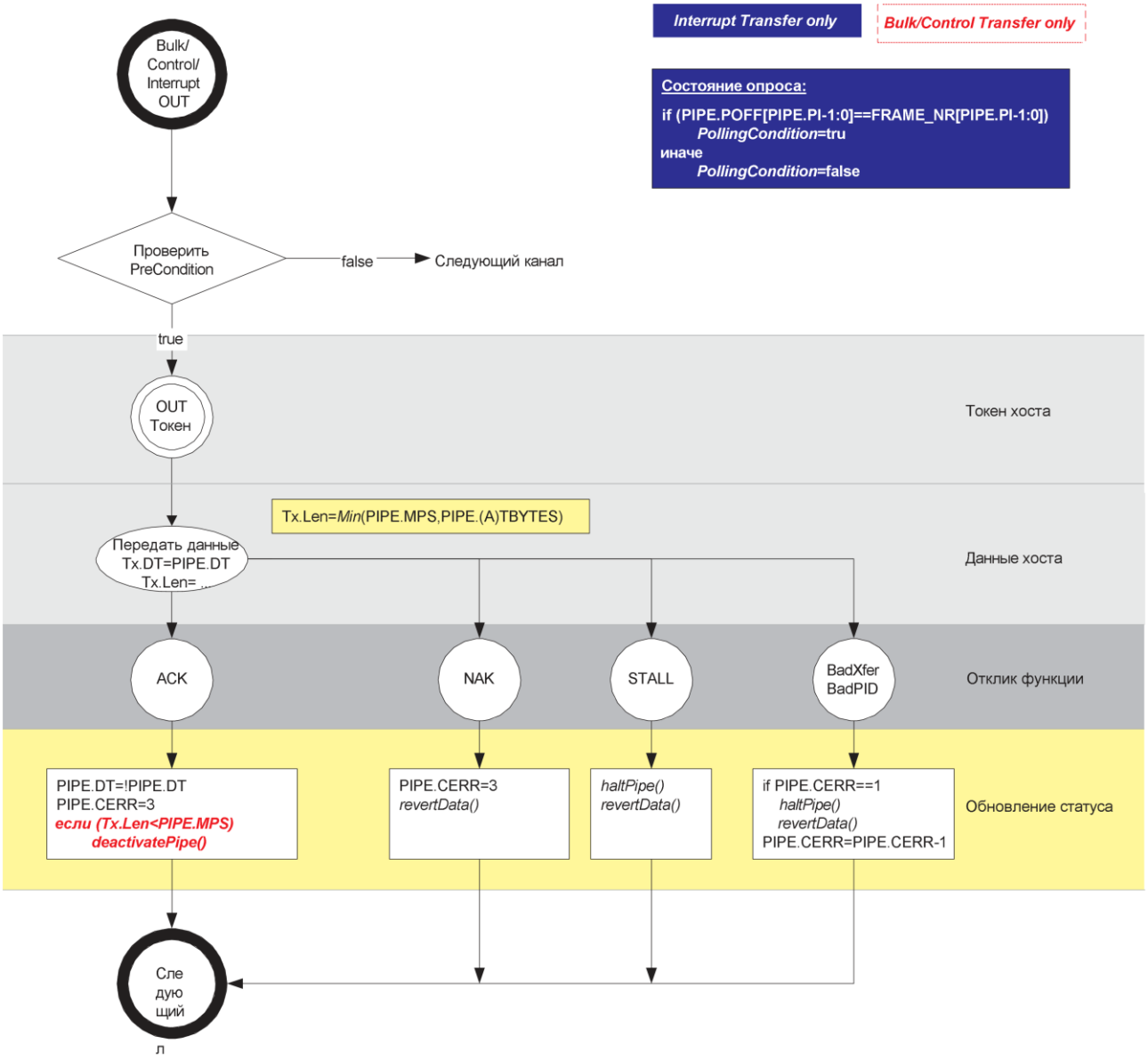


Рисунок 1.91 – Bulk/Control/Interrupt OUT хоста

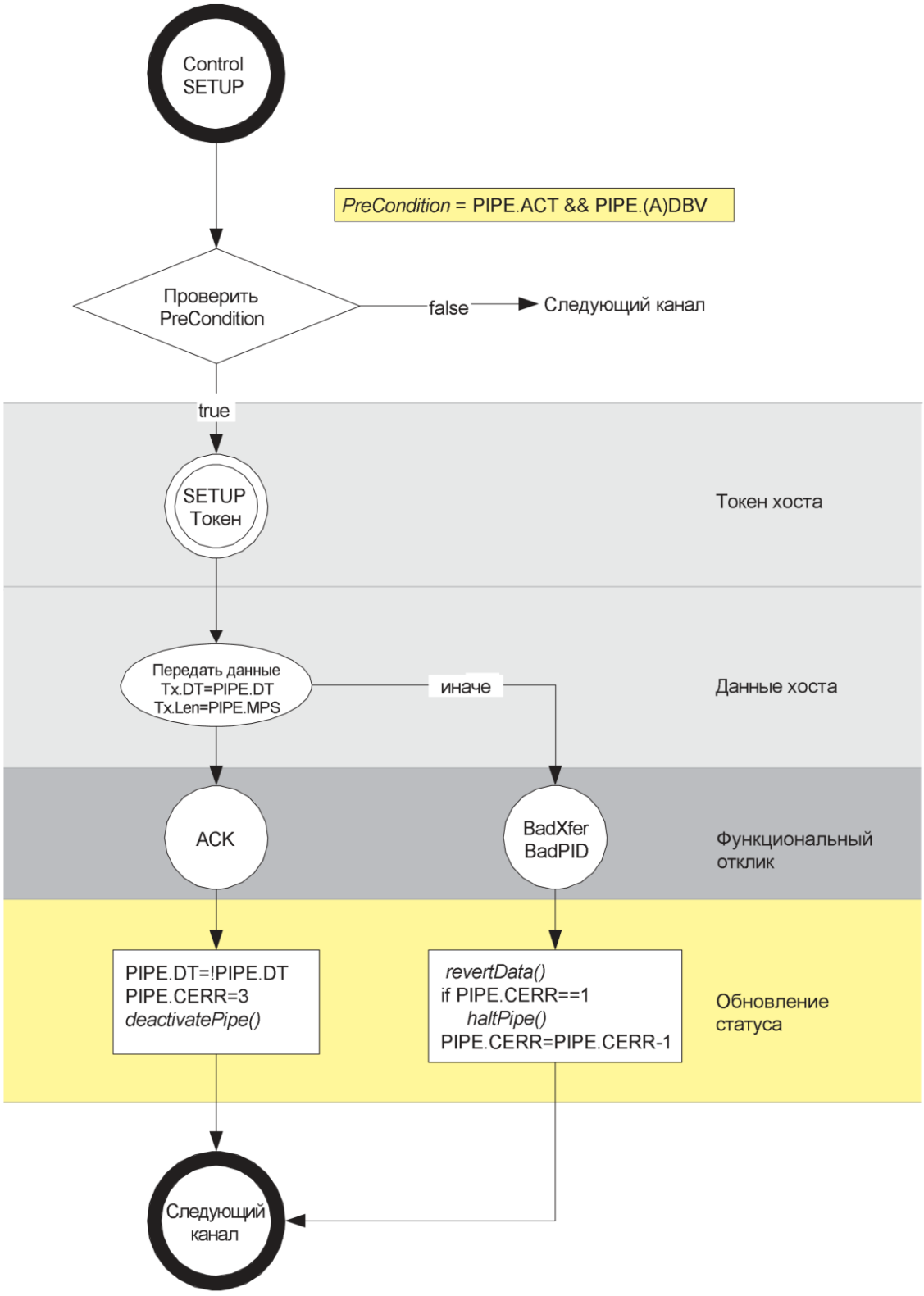


Рисунок 1.92 – SETUP управления хостом

1.3.9.9.5.1.2 Режим устройства

Если ядро работает в режиме устройства, USB_FS проверяет статус транзакции после завершения транзакции. Обновление статуса в соответствии с правилами, определенными на рисунках 1.93, 1.94, 1.95, 1.96, будет выполнено после завершения передачи данных по USB.

USB_FS использует поле адреса пакета токена USB, чтобы определить, предназначен ли следующий пакет данных для *этого* устройства. Если это условие выполняется, поле конечной точки и PID в пакете токена USB используется аппаратным обеспечением для выбора соответствующей страницы канала. Таким образом, информация из выбранной страницы канала используется для следующей передачи данных по USB.

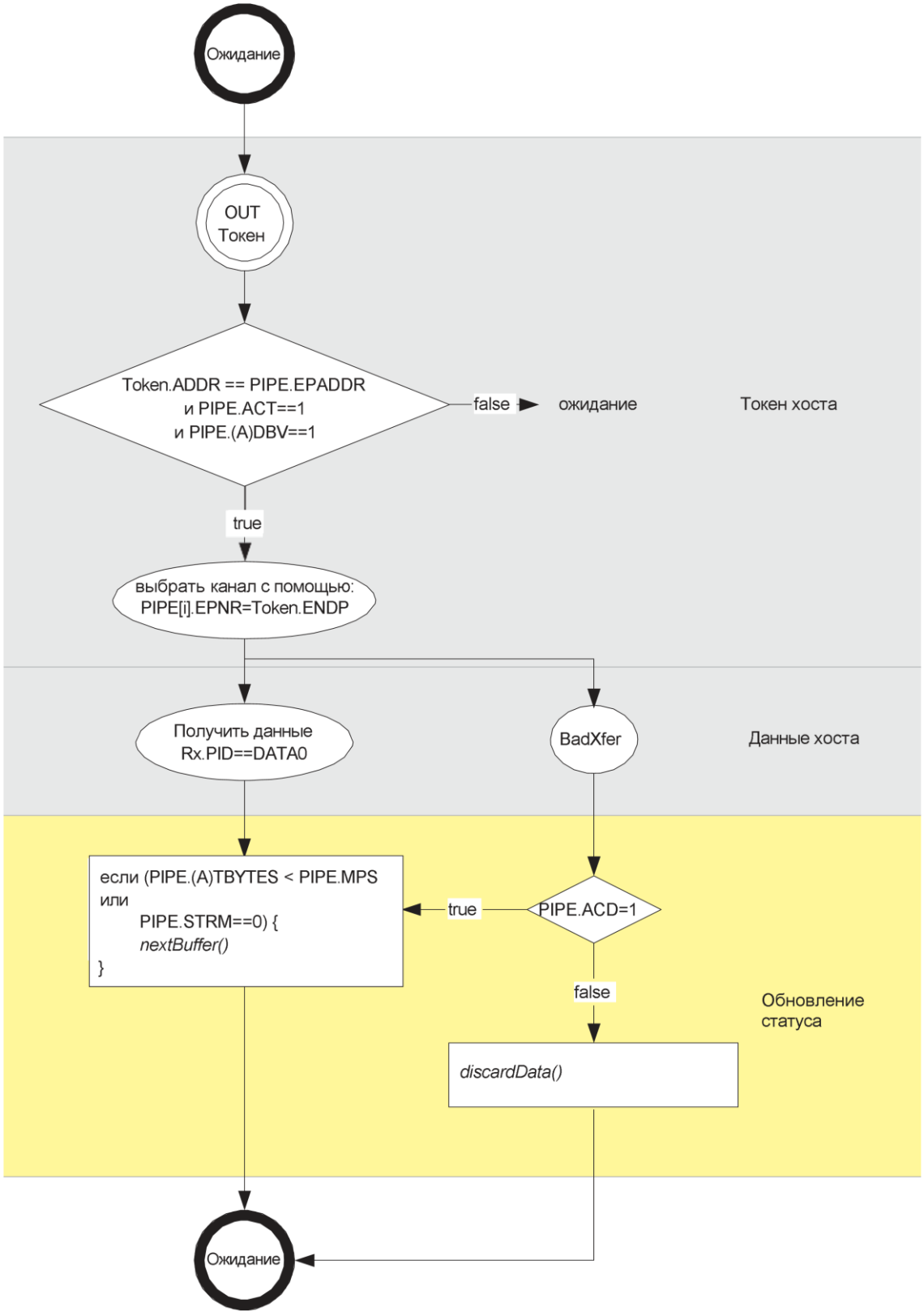


Рисунок 1.93 – Изохронный OUT устройства

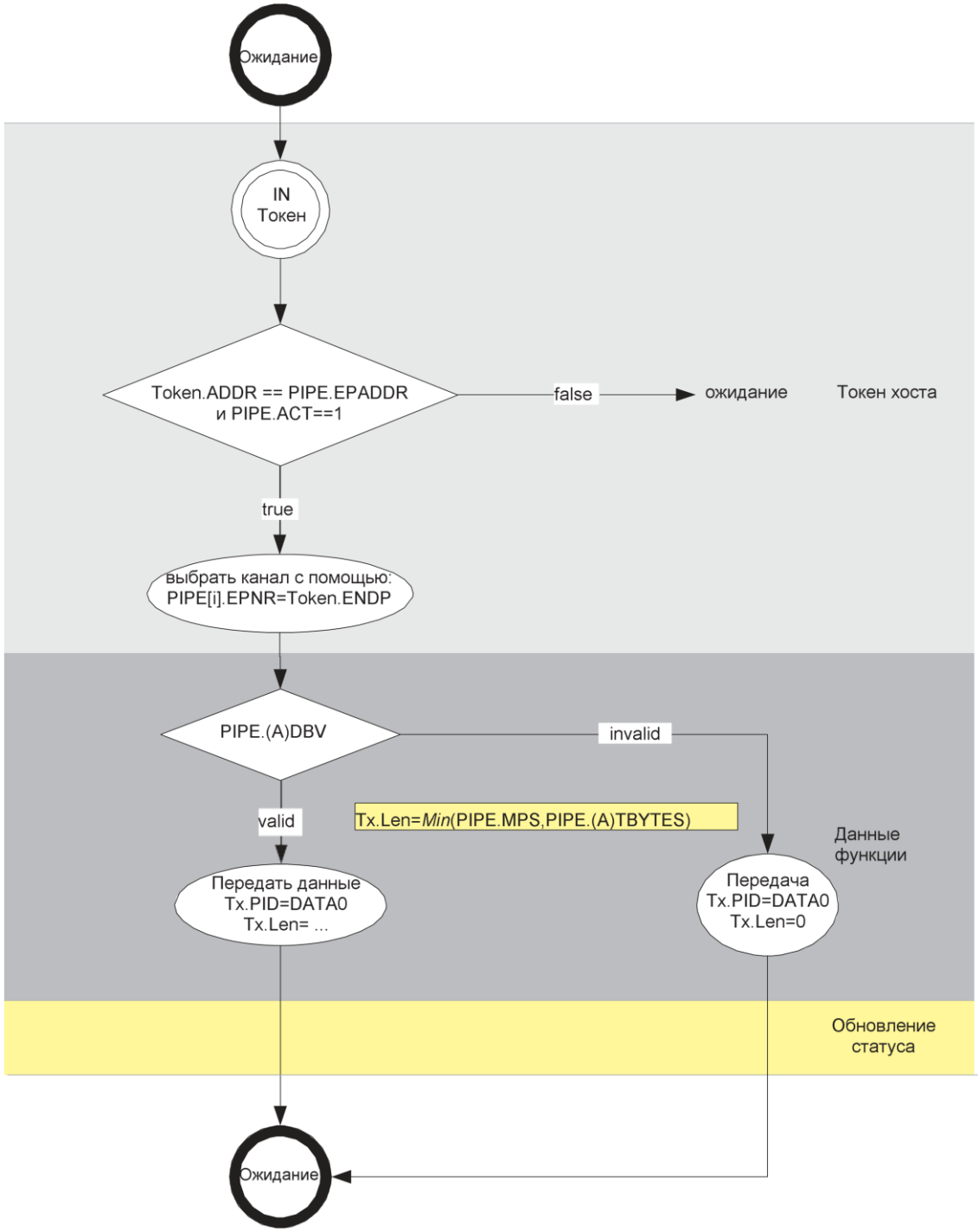


Рисунок 1.94 – Изохронный IN устройства

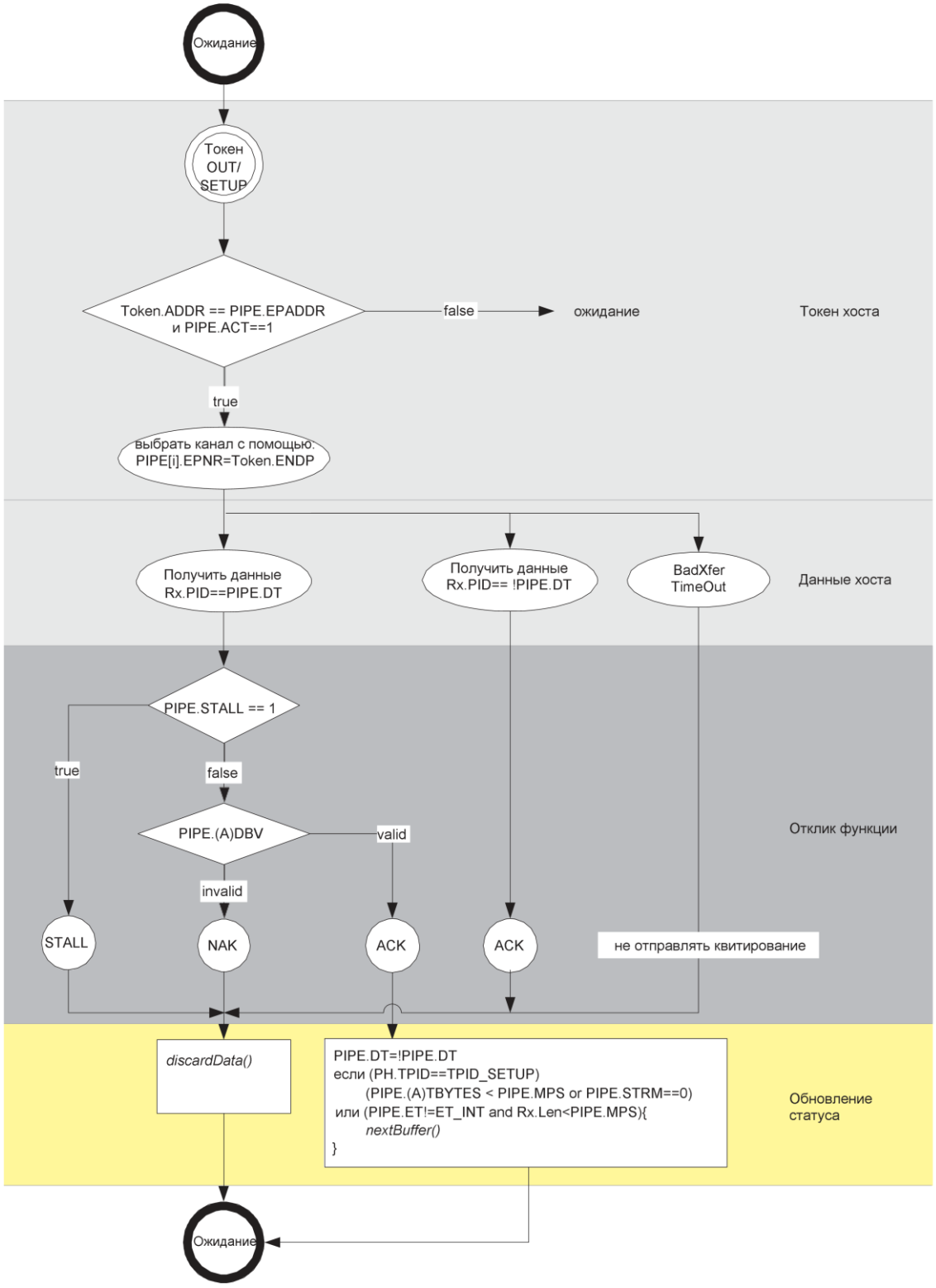


Рисунок 1.95 – Bulk/Control/Interrupt OUT/SETUP устройства

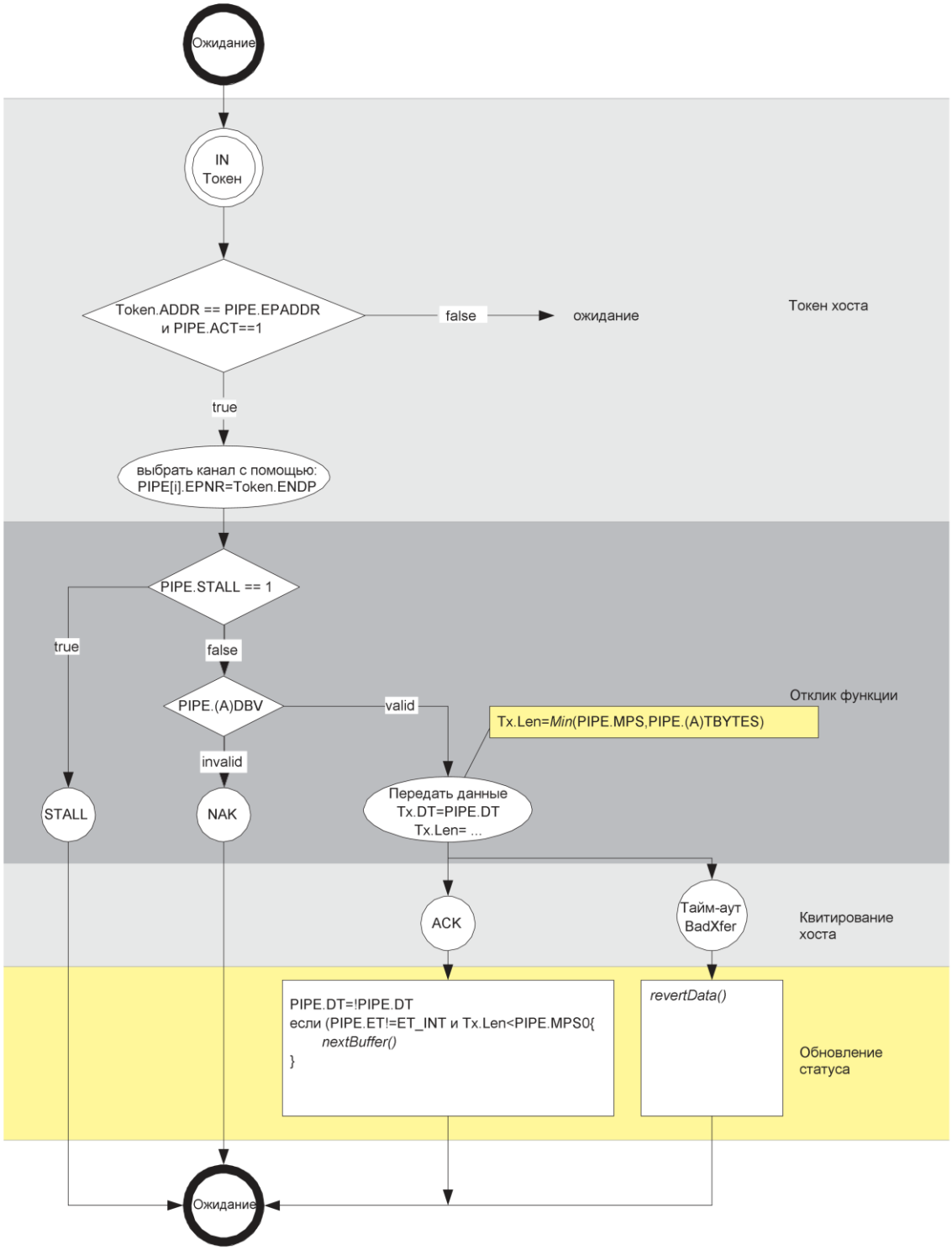


Рисунок 1.96 – Bulk/Control/Interrupt IN устройства

1.3.9.9.5.2 Иерархия прерываний

На рисунке 1.97 показана иерархия прерываний, используемая в USB_FS. Все возможные источники прерываний могут быть включены с помощью соответствующих регистров

маски. Установка бита маски приводит к прерыванию, если соответствующий бит события установлен в единицу.

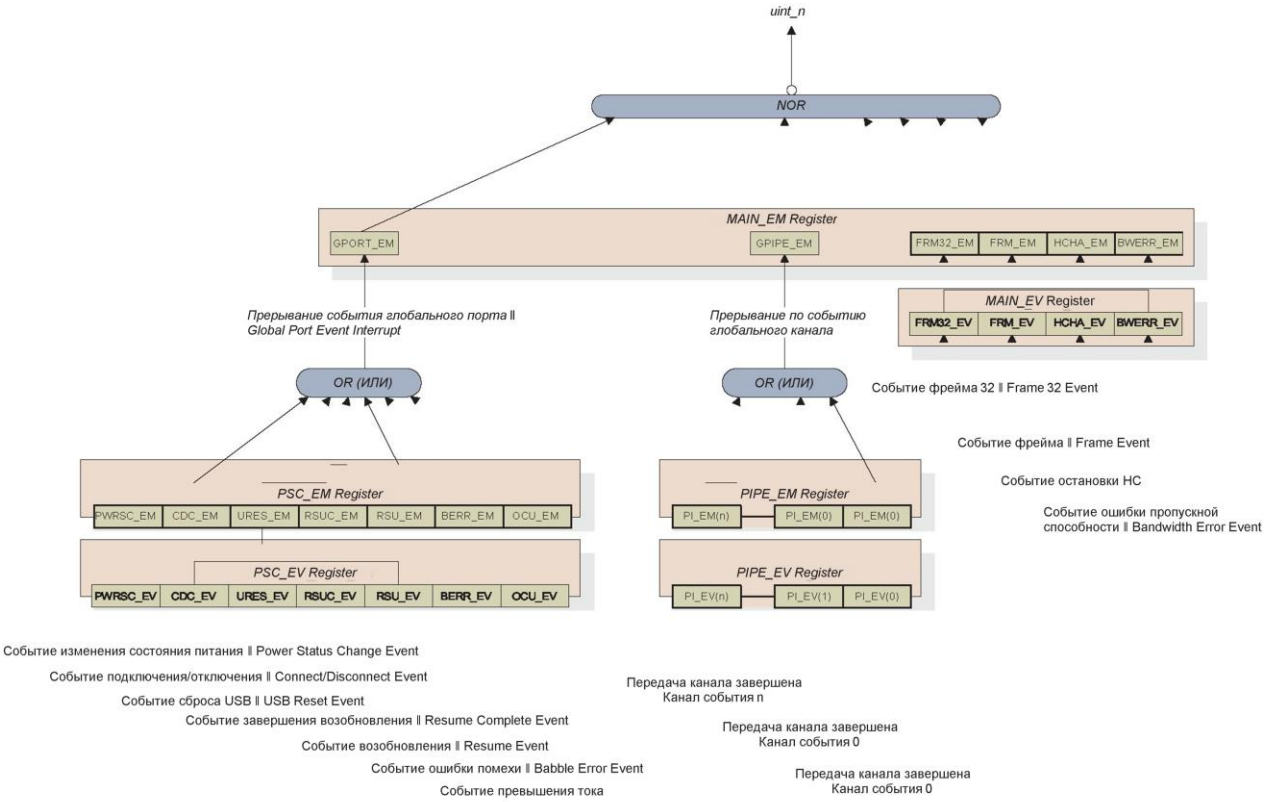


Рисунок 1.97 – Иерархия прерываний

1.3.9.9.5.3 Обработка порта

Вся информация об управлении и состоянии порта хранится в группе регистров *Port Register Group*.

Программное обеспечение информируется через *Port Status Change Event Register* (Регистр события изменения состояния порта), если произошло изменение состояния порта.

Порт поддерживает следующие функции:

– обнаружение помех (babble);

– обнаружение подключения/отключения;

– обнаружение перегрузки по току;

– определение скорости устройства;

– генерация сброса порта;

– перевод порта в приостановленное состояние;

– обнаружение/генерация возобновления работы порта;

– переключение порта в несколько тестовых режимов.

1.3.9.9.5.3.1 возможности тестирования порта

Спецификация USB определяет некоторые тестовые режимы для высокоскоростных хост-контроллеров/хабов и функций. (см. спецификацию USB, п. 7.1.20)

USB_FS поддерживает следующие тестовые режимы. Эти тестовые режимы получены из высокоскоростных тестов и адаптированы для работы с полно-/низкоскоростным ядром:

– **Test_SE0_NAK.** В режиме хоста трансивер порта управляет SE0. В режиме устройства USB_FS_ отвечает на любой пакет токена IN с квитированием NAK в течение нормального времени отклика;

– **Test_J.** Приемопередатчик порта переходит в состояние J;

– **Test_K.** Приемопередатчик порта переходит в состояние K. Test_Force_Enable Не применяется для полно-/низкоскоростных устройств. Не поддерживается USB_FS;

– **Test_Packet.** Порт будет повторять передачу тестового пакета. Тестовый пакет определяется спецификацией USB. Необходимо, чтобы программное обеспечение обеспечило соответствующие данные и конфигурацию канала, как описано ниже.

Вход в тестовые режимы Test_J, Test_K и Test_SE0_NAK осуществляется путем установки битового поля *PTESTC* регистра *Port Control Register* в соответствующее значение.

Тестовый пакет должен быть сформирован с помощью программного обеспечения, как описано в следующем разделе.

Ниже приведены требования к программному обеспечению для генерации тестовых пакетов.

Программа должна подготовить тестовый пакет как обычный пакет данных. Поля *Port Enable, Periodic Schedule Enable* и *Host Run/Stop* должны быть установлены на единицу. Для отправки тестового пакета используйте канал 0 USB_FS, а программа должна инициализировать значения конфигурации,приведенные в таблице 1.355.

Таблица 1.355 – Конфигурация канала тестового пакета

Регистр	Бит/поле	Значение	Описание
PIPE_CFG	*STRM*	0	Потоковый режим не поддерживается.
	ET	ET_ISO	Тип канала ISO.
	MPS	0x35	Длина пакета.
	ACID	0	Это поле игнорируется в режиме передачи.

Регистр	Бит/поле	Значение	Описание
	EPS	x	Выберите скорость конечной точки: x = 0 для полной скорости (full speed), x = 1 для низкой скорости (low speed).
	PI	0	Опрос каждого фрейма.
	POFF	0	Без смещения.
	ACID	0	Это поле игнорируется в режиме передачи.
PIPE_STAT	*DBOFF*	0	Смещение буфера данных канала равно нулю.
	DBSEL	0	Выбран первый буфер.
	DT	0	Ожидается PID данных DATA0.
PIPE_CTRL	*TPID*	TPID_OUT	Выбрать режим передачи.
	ACT	1	Канал разрешен.

Установите в поле *Total Bytes to Transfer* значение 0x35 (53 байта) и данные тестового пакета, которые должны быть применены:

– 0x00000000;

– 0x00000000;

– 0xAAAAAA00;

– 0xAAAAAAAA;

– 0xEEEEEEAA;

– 0xEEEEEEEE;

– 0xFFFFFEEE;

– 0xFFFFFFFF;

– 0xFFFFFFFF;

– 0xDFBF7FFF;

– 0xFDFBF7EF;

– 0xDFBF7EFC;

– 0xFDFBF7EF;

– 0x7E.

Ядро начнет отправку тестового пакета после того, как поле *Data Buffer Valid* будет установлено на единицу. Тестовый пакет будет предварен обычным полно-/низкоскоростным шаблоном SYNC, а завершен обычным полно-/низкоскоростным EOP в соответствии со скоростью подключенного устройства. Чтобы отправить тестовый пакет в цикле, снова установите поле *Data Buffer Valid* на единицу, если текущая передача завершена.

1.3.9.9.5.4 Управление буфером

Поток данных от ядра или к ядру USB_FS осуществляется с помощью буферов данных, а не просто FIFO. Основное преимущество диспетчера буферов заключается в том, что каждый буфер может содержать больше данных полезной нагрузки, чем один пакет данных USB. Задача ядра USB_FS_core – разделить большой буфер на отдельные USB-пакеты.

Для операций приема данных ядро USB_FS_core сохраняет данные нескольких USB-пакетов в одном буфере.

Каждый буфер данных характеризуется начальным адресом и размером. Длина определяется полем *Total Bytes Field* или *Alternative Total Bytes Field*. Начальный адрес определяется полем указателя данных *Data Pointer Field* или полем альтернативного указателя данных *Alternative Data Pointer Field*.

Когда программа предоставляет аппаратному обеспечению буфер, она устанавливает бит *Data Buffer Valid Bit* или *Alternate Data Buffer Valid Bit*. После завершения, приостановки или остановки буфера данных аппаратное обеспечение очистит бит достоверности, и будет сгенерировано событие канала. Нормальной ситуацией является завершение буфера, что означает, что все данные переданы и регистр общего числа байтов (Total Bytes Register) установлен на ноль. Буфер приема может быть приостановлен, поскольку не активирован режим потоковой передачи, но Total Bytes Register еще не равен нулю. Буфер остановлен из-за ошибки, приводящей к остановке канала. «Альтернативные» версии этих битов будут существовать, только если аппаратура настроена на использование альтернативного буфера.

Программа может работать с одним единственным буфером. Этого может быть достаточно для асинхронных каналов. Для периодических конечных точек с высокой пропускной способностью необходимо, чтобы ядро USB_FS_core могло генерировать непрерывный поток данных. Использование управления одним буфером может вызвать проблемы, поскольку аппаратное обеспечение не может выполнять USB-трафик для этой конечной точки, когда буфер данных истекает, а программное обеспечение должно предоставить новый буфер в течение очень короткого времени.

Чтобы избежать этой проблемы, ядро USB_FS может быть настроено на поддержку «альтернативных буферов». Это позволяет оборудованию обрабатывать 1-й буфер данных, пока программа заполняет 2-й буфер, и наоборот. После инициализации канала аппаратное обеспечение всегда начинает работу с неальтернативного буфера данных. Бит *Data Buffer Select Bit* используется для хранения информации о том, какой буфер данных находится в процессе

обработки. Этот бит переключается аппаратными средствами после истечения срока действия буфера данных и включения альтернативного буфера для этого канала.

Для написания эффективных программных драйверов необходимо учитывать следующие замечания:

– каналы передачи Bulk/Control/Interrupt/Isochronous:

– когда размер буфера кратен *Maximum Packet Size*, аппаратное обеспечение генерирует USB-пакеты длиной MPS;

– если размер буфера не кратен *Maximum Packet Size*, аппаратное обеспечение генерирует первые USB-пакеты длиной MPS, а последний USB-пакет – как короткий пакет, (что имеет особое значение в контексте USB для конечных точек управления и массовой передачи данных);

– когда размер буфера равен 0, аппаратное обеспечение генерирует пакет с нулевыми данными;

– каналы массового/контрольного приема:

– для получения большого количества USB-пакетов в одном буфере следует активировать потоковый режим;

– размер буфера должен быть кратен *Maximum Packet Size*;

– состояние короткого пакета характеризуется значением общего количества байт, которое больше 0 и не кратно *Maximum Packet Size* после истечения срока действия буфера. Короткие пакеты всегда генерируют события канала, отключая его;

– прерывистые/изохронные каналы приема:

– для каналов с высокой пропускной способностью следует активировать режим потоковой передачи;

– размер буфера должен быть кратен удвоенному *Maximum Packet Size*.

1.3.9.9.5.5 Обработка буфера данных для потоковых каналов

USB_FS поддерживает автоматическую обработку больших буферов данных. Все данные USB передаются по каналам передачи данных, которые в спецификации USB называются каналами [в оригинале – pipes].

Для каналов с высокой пропускной способностью рекомендуется конфигурировать ядро на наличие альтернативных буферов. Эта функция может быть активирована для всех каналов. Это позволяет полностью отделить задержку прерывания микроконтроллера или DSP от любых временны́х требований USB.

Все данные, которые хранятся в буфере или должны быть извлечены из буфера, будут обрабатываться с порядком байт Little Endian.

Как правило, буфер данных, которые должны быть переданы или получены, кратен допустимому максимальному размеру пакета. USB_FS позволяет программе предоставлять большие буферы с размером, превышающим Maximum Packet Size (максимальный размер пакета). Буферы передачи будут автоматически разбиваться на части максимального размера пакета (Maximum Packet Size), а буферы приема будут автоматически сцепляться.

1.3.9.9.5.5.1 Потоковая передача данных для каналов приема

Полученные данные можно сцеплять и автоматически выравнивать по байтам с помощью функции Streaming Mode (режим потоковой передачи).

При этом необходимо учитывать некоторые программные проблемы, связанные с обработкой буферов потоковых данных:

– для всех каналов потокового приема: $bufferSize \geq 2 \times MaxPacketSize$;

– для каналов приема с высокой пропускной способностью и режимом передачи iso/bulk рекомендуется использовать потоковый режим.

Описание проблемы. Следующая ситуация возникает, если для каналов приема отключен потоковый режим и ядро получает пакеты с полезной нагрузкой данных, которая не помещается в кратное количество слов данных (слова данных содержат либо 2 байта, если разрядность данных = 16; либо 4 байта, если разрядность данных = 32).

Поскольку модель потока данных USB ориентирована на байты, может случиться так, что ядро получит много данных и сохранит их в нескольких буферах. Особенно при изохронном типе передачи, который позволяет передавать пакеты разного размера, вероятность появления пустых байтовых полей очень высока, как показано на рисунке 1.98. Как правило, заполнение буфера начинается с адреса байта 0. В приведенном примере на один буфер приходится всего 25 или 27 байт.

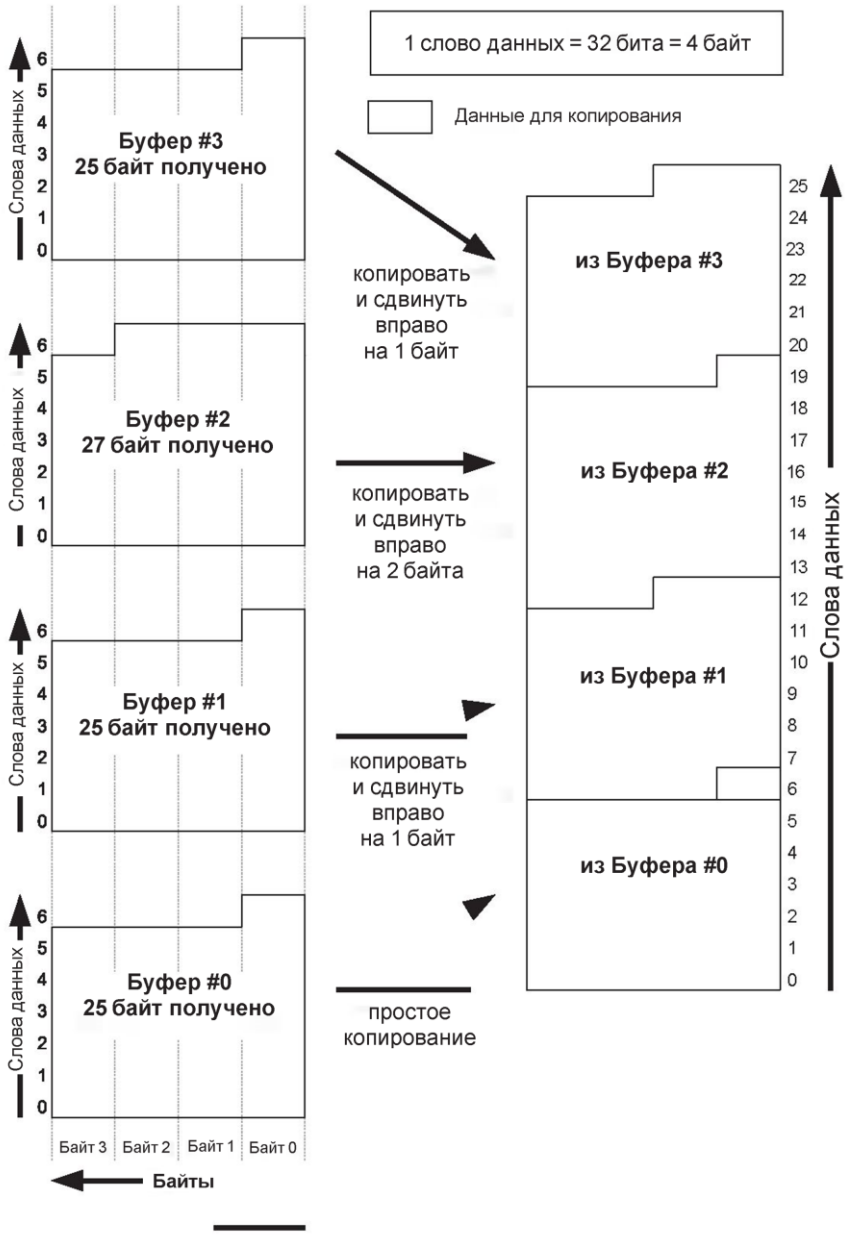


Рисунок 1.98 – Сцепление буферов данных без потоковой передачи

Теперь программа должна сцеплять эти буферы, поскольку они могут принадлежать одному и тому же изображению, видео или чему-то еще. Поскольку в конце этих буферов могут находиться слова данных с пустыми полями байтов, необходимо выполнить сдвиг байтов. Это должно быть выполнено программой с помощью специальной процедуры сдвига байтов (что может занять много времени) или с помощью канала DMA с возможностью сдвига байтов.

Как показано на рисунке 1.98, передача завершается после записи буфера #3 в память. Теперь программа должна сцеплять все буферы. Это делается следующим образом:

– скопировать буфер #0, слова данных 0-6;

– скопировать буфер #1, слова данных 0-6, со сдвигом всех слов данных на 1 байт;

– скопировать буфер #2, слова данных 0-6, со сдвигом всех слов данных на 2 байта;

– скопировать буфер #3, слова данных 0-6, со сдвигом всех слов данных на 1 байт.

Важно отметить, что 3 из 4 операций копирования должны выполняться **со сдвигом байтов**.

Решение. Описанная выше проблема возникает из-за того, что выравнивание байтов должно выполняться *после* заполнения нескольких буферов данными. Простая суммирующая логика в ядре решает эту проблему. Выравнивание байтов выполняется *во время* заполнения буферов данными, как показано на рисунке 1.99 , если активирован потоковый режим.

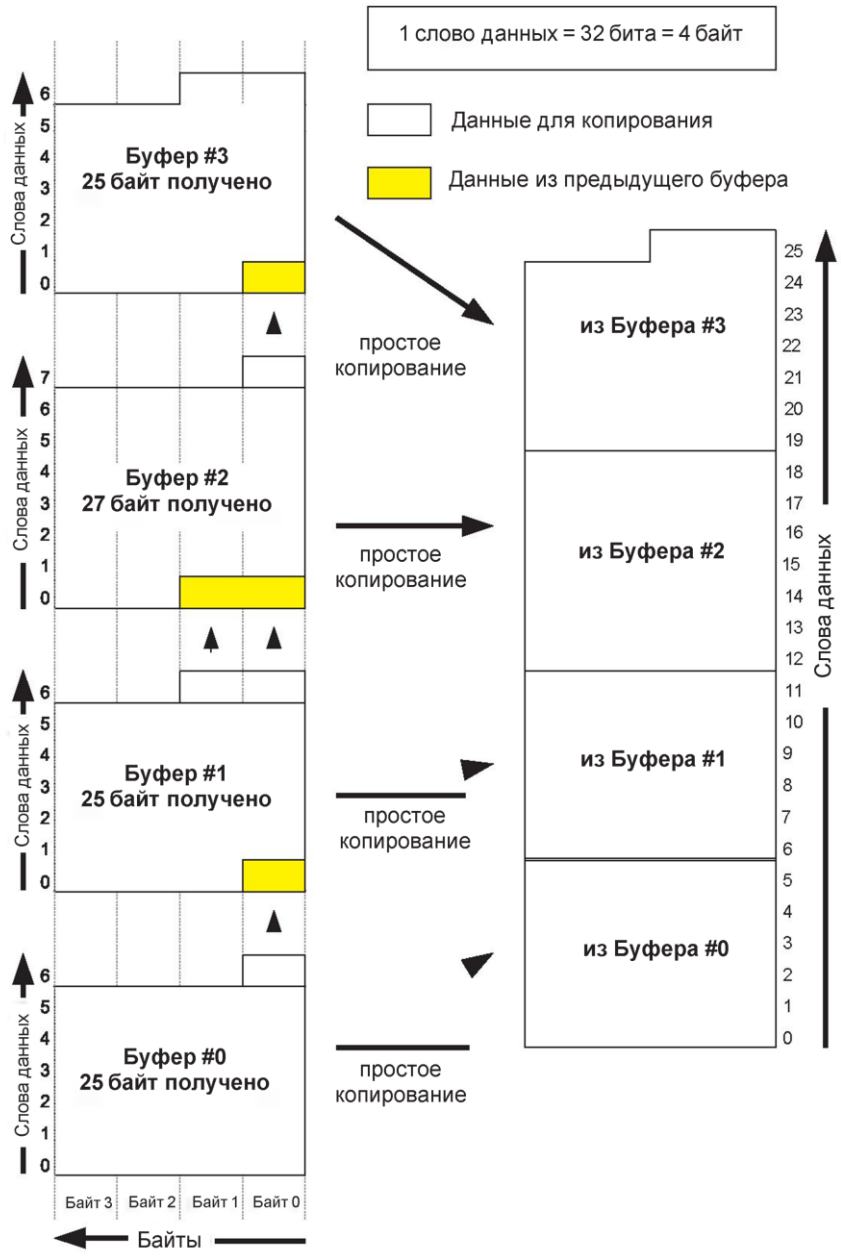


Рисунок 1.99 – Сцепление буферов данных с потоковой передачей

После получения буфера #0 и записи всех данных в память ядро сохраняет текущее значение смещения байта (*DBOFF* = 1) внутри ядра. Если в память записывается первый пакет буфера #1, ядро знает смещение последнего байта предыдущего буфера и записывает байты #0 слова данных #6 буфера #0 снова в память, в начало буфера #1.

После получения буфера #1 и записи всех данных в память ядро сохраняет текущее значение смещения байта (*DBOFF* = 2) внутри ядра. Если в память записывается первый пакет

буфера #2, ядро знает смещение последнего байта предыдущего буфера и записывает байты #0/#1 слова данных #6 буфера #1 снова в память, в начало буфера #2.

После получения буфера #2 и записи всех данных в память ядро сохраняет текущее значение смещения байта (*DBOFF* = 1) внутри ядра,

и снова записывает байт #0 слова данных #6 буфера #2 в память, если получен первый пакет из буфера #2.

После того как буфер #3 записан в память, вся передача завершена. Теперь программа должна скопировать весь буфер. Это делается следующим образом:

– скопировать буфер #0, слова данных 0-5;

– скопировать буфер #1, слова данных 0-5;

– скопировать буфер #2, слова данных 0-6;

– скопировать буфер #3, слова данных 0-6.

Все операции копирования могут выполняться **без сдвига байтов**.

1.3.9.9.5.6 Права доступа к регистрам каналов для режима хоста

В этом разделе описывается, как программа получает доступ к странице *Pipe Register Page*. Поскольку существуют регистры, которые должны быть модифицированы аппаратно, необходимо определить некоторые правила доступа, чтобы избежать конфликтов между программными и аппаратными модификациями регистра.

Программа может получить доступ к регистрам канала, если она выберет определенную страницу, установив поле *Pipe Select Field*. Каждой странице регистра канала присвоено состояние, которое определяется содержимым битов *PIPE_CTRL.ACT_Bit* и *PIPE_STAT.ACTS_Bit* (далее – *ACT* и *ACTS*), как показано на рисунке 1.100.

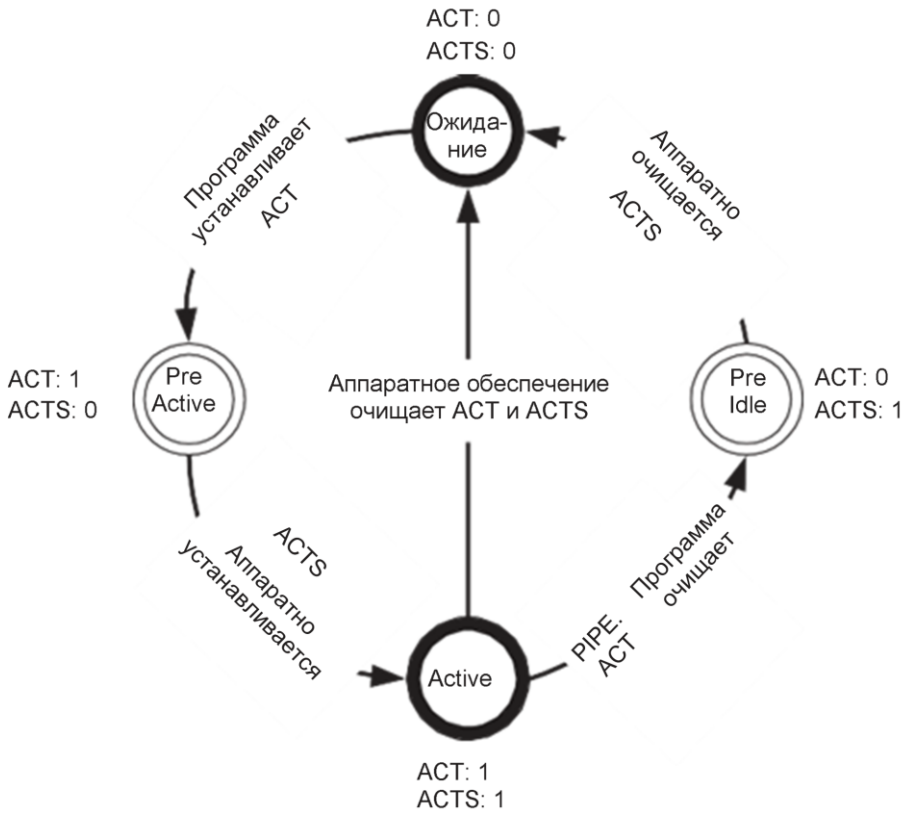


Рисунок 1.100 – Состояния канала

Ожидание ‖ Idle. Это начальное состояние. Оно определяется ACT = 0 and ACTS = 0. Аппаратное обеспечение полностью игнорирует этот канал. Программа может изменять все регистры, относящиеся к этой странице регистров, пока бит ACT остается равным 0. Программа может сконфигурировать канал для требуемой передачи данных. Когда конфигурация канала завершена, бит ACT можно установить, чтобы инициировать USB-трафик для этого канала.

Предактивное ‖ PreActive. Это состояние активируется, когда программа устанавливает бит ACT. Состояние определяется ACT = 1 и ACTS = 0. Аппаратное обеспечение установит бит ACTS на 1 при следующей обработке этой страницы регистров.

Активное ‖ Active. Это состояние активируется, когда аппаратное обеспечение установило бит ACTS. Для этого канала генерируется USB-трафик. Аппаратные средства могут изменять выделенные биты. Программа должна получить доступ только к регистрам буфера данных. Когда состояние аппаратного обеспечения меняется на Idle (Ожидание), оно зачищает биты ACT и ACTS и генерирует событие канала, которое может вызвать прерывание.

Предожидательное ‖ Pre Idle. Это состояние активируется, когда программа зачищает бит ACT. USB-трафик по-прежнему может выполняться для этого канала, а аппаратное

обеспечение может изменять выделенные биты. Аппаратные средства очистят бит ACTS в худшем случае через 1 мс, когда произойдет инкремент следующего фрейма.

Они обработают этот канал и перейдут в состояние Idle. Будет сгенерировано событие канала, которое может привести к прерыванию.

В таблице 1.356 показано, как программа и аппаратное обеспечение могут изменять выделенные ресурсы регистров канала:

Таблица 1.356 – Правила доступа к регистрам каналов

Состояние	Idle		PreActive		Active		Pre Idle	
	SW	HW	SW	HW	SW	HW	SW	HW
PIPE_ADDR	A	F	F	F	F	F	F	F
PIPE_CFG	A	F	F	F	F	F	F	F
PIPE_CTRL	A	F	F	F	A*	F	F	F
PIPE_STAT	A	F	D	D	D	D	D	D
PIPE_DPTR	A	F	D	D	D	D	D	D
PIPE_TBYTES	A	F	D	D	D	D	D	D
PIPE_ADPTR	A	F	D	D	D	D	D	D
PIPE_ATBYTES	A	F	D	D	D	D	D	D

Значение символов:

– **A** - доступ на запись разрешен (**A*** программе разрешено сбрасывать только бит ACT);

– **F** - доступ на запись запрещен;

– **D** - программа может вносить изменения, если (*Alternative*) *Data Buffer Valid* не установлен. Аппаратные средства могут вносить изменения, если (*Alternative*) *Data Buffer Valid* установлен.

1.3.9.9.5.7 Обработка каналов управления периферийными устройствами

В этом разделе описывается, как программа должна использовать регистры канала управления при работе в качестве USB-устройства.

Протокол USB определяет канал управления по умолчанию для конфигурации устройства и управления им. Этот канал имеет некоторые фиксированные атрибуты:

– тип передачи Control Transfer (управляющая передача);

– Max Packet Size ≤ 64 байтам;

– фаза данных, следующая за токеном SETUP, **всегда** равна 8 байтам и **всегда** определяет всегда запрос USB;

– канал должен быть активен сразу после получения сброса USB. Согласно спецификации USB, этап SETUP **должен** завершаться квитированием ACK;

– номер конечной точки – 0.

– канал двунаправленный.

Эти особенности требуют особого обращения с каналом управления, если ядро работает в периферийном режиме. USB_FS использует регистры управления и состояния канала как в режиме хоста, так и в режиме периферии. Если ядро работает в периферийном режиме, он должен быть чувствителен к USB-токенам обоих направлений для канала управления. (токены SETUP, OUT и IN).

Настройки канала управления по умолчанию. После получения сброса USB канал[0] резервируется как канал управления. В таблице 1.357 приведены настройки по умолчанию для этого канала.

Таблица 1.357 – Настройки канала управления по умолчанию после сброса USB

Регистр	***Бит/поле***	***Значение***	***Описание***
PIPE_CTRL	*ACT*	1	Канал разрешен. Ядро будет отвечать на токены на канале управления.
	TPID	TPID_SETUP	Сначала ожидается токен SETUP.
PIPE_CFG	*STRM*	–	Сброс USB не влияет на этот бит.
	EPS	–	Сброс USB не влияет на этот бит.
	ET	ET_CTRL	Канал является управляющим.
	STALL	0	Канал не остановлен.
	MPS	–	Сброс USB не влияет на это битовое поле. Тем не менее, токены SETUP всегда будут приниматься с квитированием ACK.
PIPE_ADDR	*EPADDR*	0x00	Значение сброса адреса устройства.
	EPNR	0x0	Конечной точкой управления всегда является EP0.
PIPE_STAT	все только чтение	–	Те же значения сброса, что и после сброса при включении питания.

Регистр	*Бит/поле*	*Значение*	*Описание*
	DBOFF	0	Смещение буфера данных канала равно нулю.
	DBSEL	0	Выбран первый буфер.
	DT	0	Сначала ожидается токен SETUP с PID данных DATA0.
PIPE_DATA_TBYTES	*TBYTES*	0	Данных для передачи нет. Токены SETUP будут приняты при квитировании ACK в любом случае.
PIPE_DATA_PTR	*DPTR*	0	Это значение зарезервировано как указатель по умолчанию для транзакций IN/OUT канала управления. Оно недействительно для транзакций SETUP. Запросы USB, следующие за токеном SETUP, будут записаны по адресу 0x0000.
	DBV	0	Действительного буфера данных нет. Это приведет к ответу NAK на токены IN и OUT. Тем не менее, токены SETUP будут приняты с ACK.

Последовательность передачи данных канала управления. Последовательность направления потока данных в канале управления следует правилам, определенным в спецификации USB. Предусмотрены следующие случаи:

– этап установки с этапом передачи данных IN (хост получает данные) и этап состояния;

– этап установки с этапом передачи данных OUT (хост отправляет данные) и этапом состояния;

– этап установки без этапа данных. Запрос завершается только на этапе состояния.

Эти случаи показаны на рисунках 1.101, 1.102, 1.103

Кроме того, следует учитывать, что этап передачи данных IN может быть завершен без передачи всех данных. Хост может отменить этап передачи данных, отправив OUT-пакет с нулевыми данными до того, как будут получены все данные.

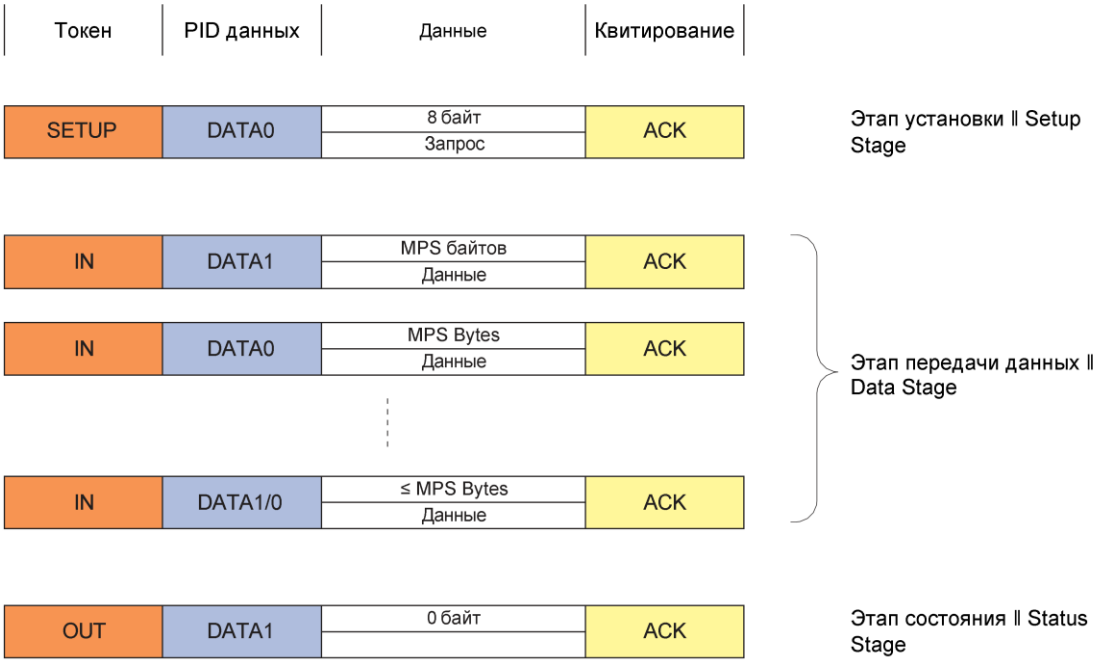


Рисунок 1.101 – Запрос настройки с этапом передачи данных IN ‖ Setup Request with IN data stage

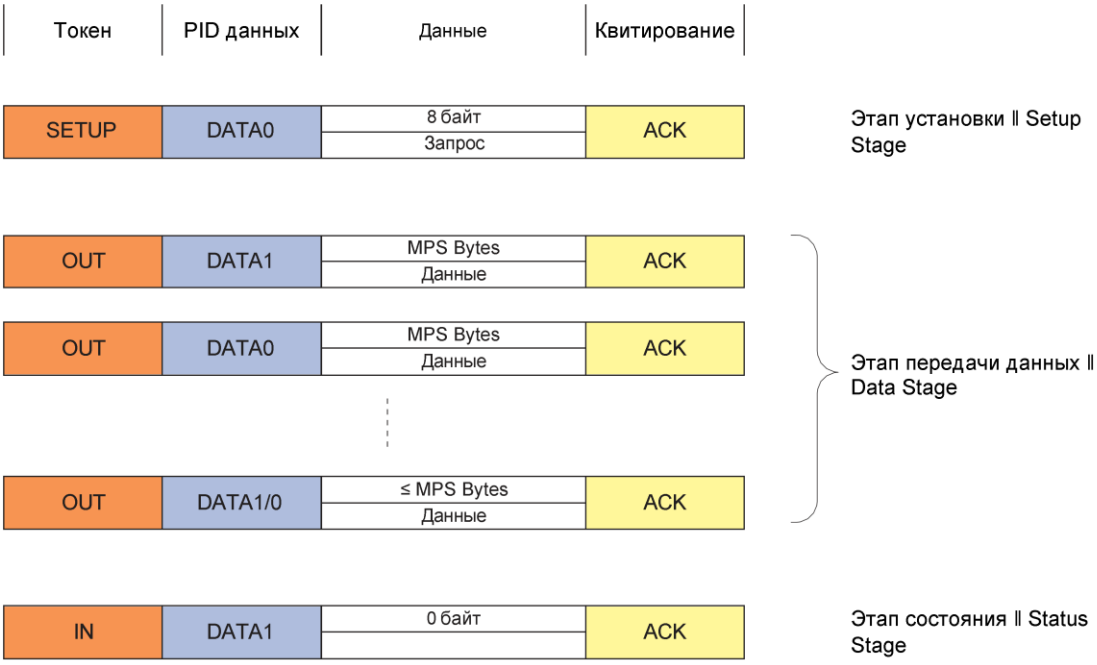


Рисунок 1.102 – Запрос настройки с этапом OUT ‖ Setup Request with OUT stage



Рисунок 1.103 – Запрос настройки без этапа данных ‖ Setup Request without data stage

1.3.9.9.5.7.1 Поток данных программы для запросов установки

На рисунках 1.104, 1.105, 1.106 показан поток данных и необходимые действия программы для всех типов запросов SETUP.

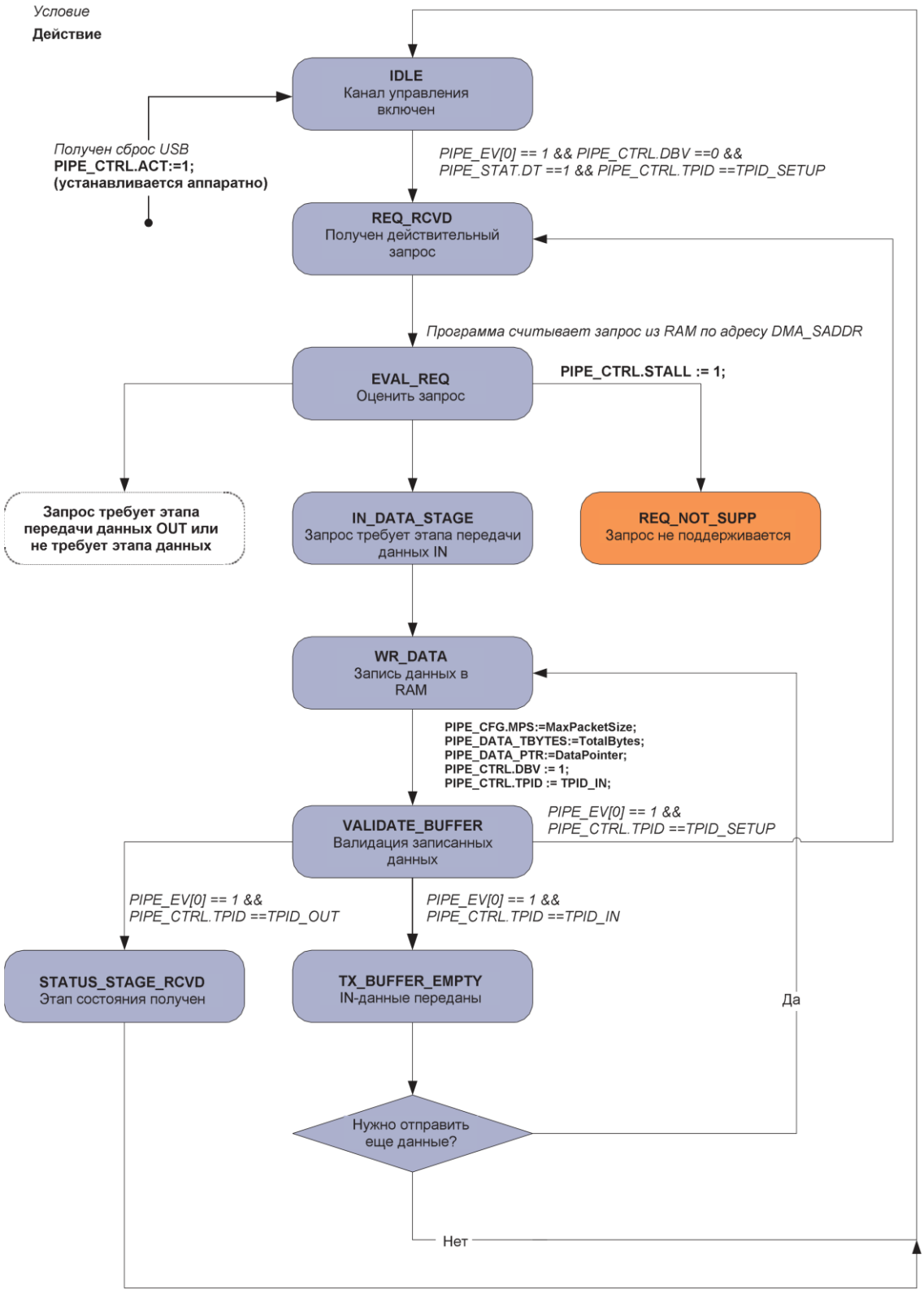


Рисунок 1.104 – Поток данных для запроса установки с этапом передачи данных IN

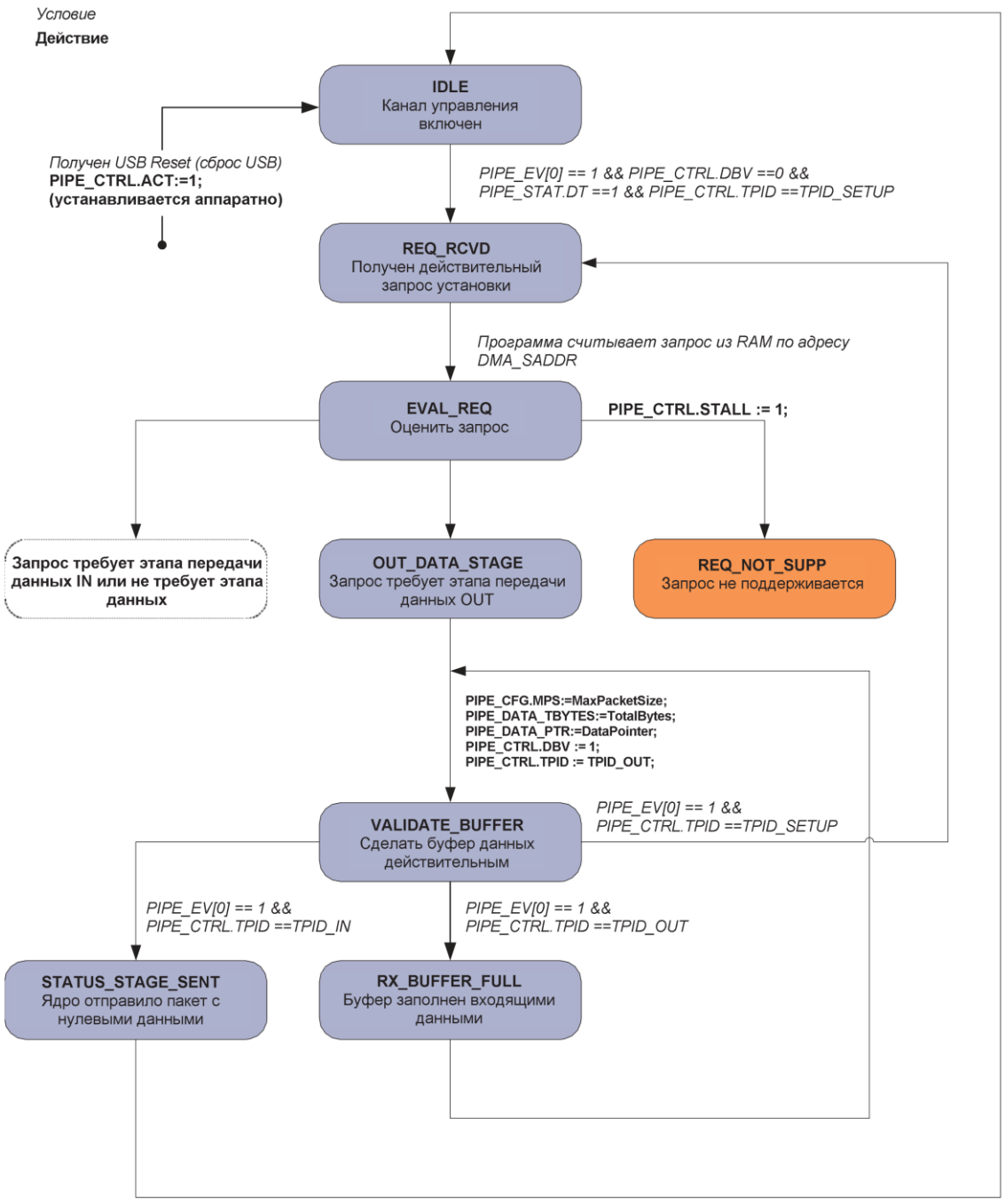


Рисунок 1.105 – Поток данных для запроса установки с этапом передачи данных OUT

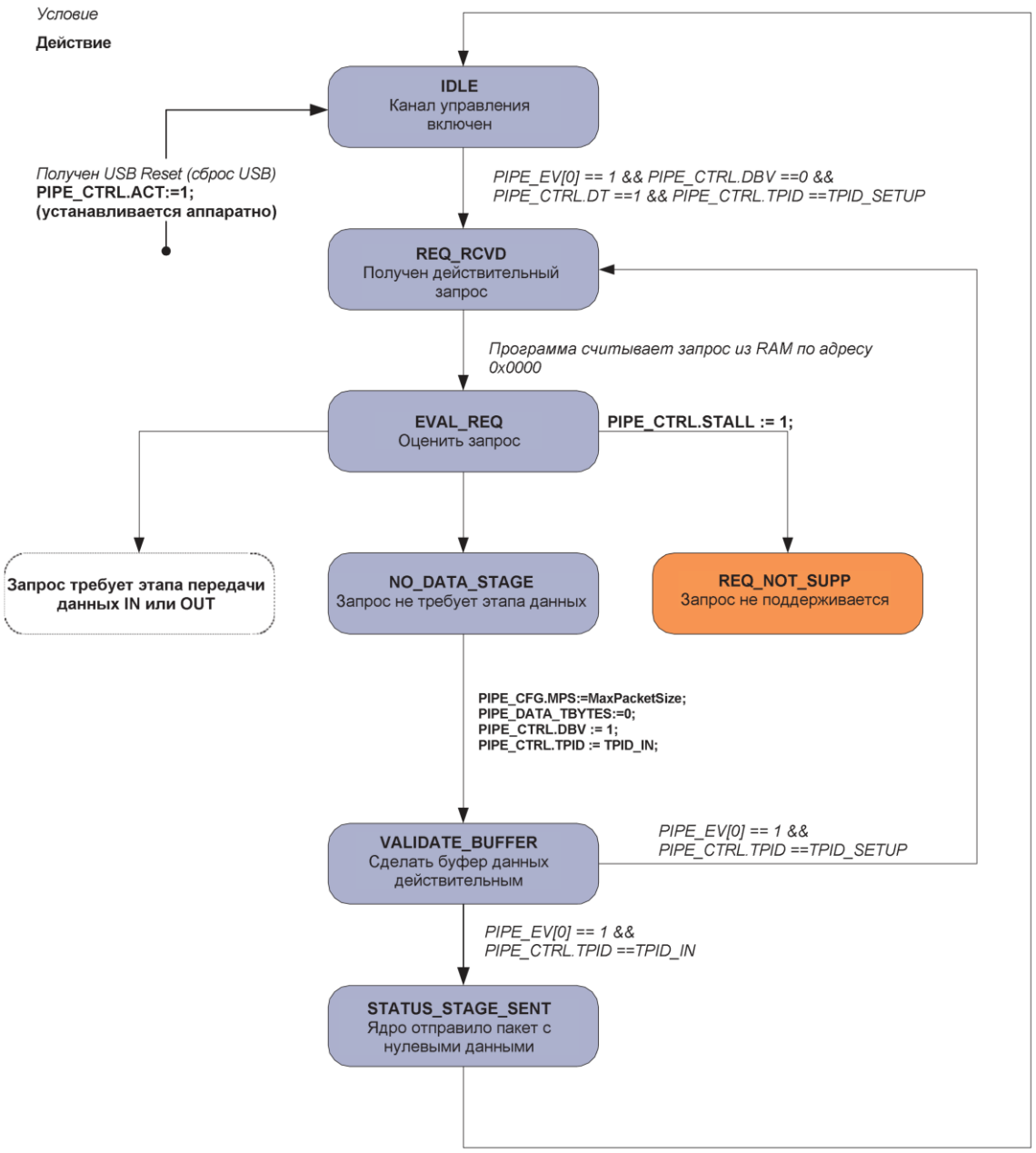


Рисунок 1.106 – Поток данных для запроса установки без этапа данных

Описание состояния потока данных

IDLE. В это состояние ядро переходит после получения действительного сброса USB. Канал конечной точки управления автоматически включается аппаратно, и ядро чувствительно к входящим токенам.

REQ_RCVD. Каждый запрос USB начинается с токена SETUP, за которым следует 8 байт с закодированным запросом. О входящем запросе свидетельствует появление события канала для канала[0] и поле TPID регистра управления каналами (Pipe Control Register), указывающее на получение токена SETUP. Аппаратные средства автоматически установят

переключатель данных в положение DATA1 и аннулируют буфер данных. Входящие токены IN и OUT будут получать NAK, пока программа не подтвердит буфер данных.

EVAL_REQ. Программа должна оценить запрос и определить, требуется ли для него этап данных IN или OUT или этап данных не требуется.

IN_DATA_STAGE. Программа определила, что запрос требует этапа передачи данных IN.

OUT_DATA_STAGE. Программа определила, что запрос требует этапа данных OUT.

NO_DATA_STAGE. Программа определила, что запрос не требует этапа данных.

WR_DATA. Заполните буферную область RAM данными для отправки.

REQ_NOT_SUPP. Если программа обнаружила неподдерживаемый запрос, она должна установить бит STALL в регистре управления каналом (Pipe Control Register). Это приведет к квитированию STALL в ответ на следующий токен (IN или OUT).

VALIDATE_BUFFER. Программа должна установить все специфические для буфера значения, такие как максимальный размер пакета (Maximum Packet Size) для канала, общий объем передаваемых данных и указатель данных в буфере. Проверка буфера завершается установкой бита DBV и битового поля TPID регистра управления каналом (Pipe Control Register).

TX_BUFFER_EMPTY. В ответ на токены IN аппаратное обеспечение будет отправлять буфер данных частями максимального размера пакета. После того как все данные будут отправлены, произойдет событие канала, и поле TPID регистра управления каналом (Pipe Control Register) покажет последний полученный токен как токен IN. Если объем данных, заданный TBYTES, кратен MPS, ядро автоматически отправит нулевой пакет данных в качестве последнего пакета.

RX_BUFFER_FULL. Аппаратное обеспечение заполнит буфер данных полученными данными. Если в буфере больше нет места, произойдет событие канала, и поле TPID регистра управления каналом (Pipe Control Register) покажет последний принятый токен как токен OUT.

STATUS_STAGE_RCVD. Программа может определить завершение передачи SETUP, проверив поле TPID регистра управления каналом, которое должно указывать на OUT PID.

STATUS_STAGE_SENT. Если поле TPID регистра управления каналом установлено в IN PID, а буфер данных действителен, ядро отправит нулевой пакет данных в ответ на токен IN в качестве этапа состояния.

1.3.9.9.6 Описание дополнительных управляющих регистров USB_AUX

1.3.9.9.6.1 Общее описание регистров

Набор дополнительных управляющих регистров USB_AUX.

Карта регистров приведена в таблице 1.358.

При дальнейшем описании отдельных полей регистров используется следующее обозначение режимов доступа пользователя к полям:

– ЧТ – поле доступно только для чтения. Запись данных в поле игнорируется;

– ЗП – поле доступно только для записи. Чтение возвращает значение 0;

– ЧТ/ЗП – поле доступно как для записи, так и для чтения;

Доступ к регистрам осуществляется посредством системного интерфейса управления (AHB5 slave).

– поддерживается только 32-х разрядный доступ. Байтовая или двух-байтовая запись будет некорректно исполнена;

– не поддерживаются burst транзакции.

Запись в зарезервированные или ЧТ биты или адреса не имеет действия, никакие ошибки не генерируются. Чтение из зарезервированных или ЗП битов или адресов возвращает 0, никакие ошибки не генерируются.

1.3.9.9.6.2 Карта регистров

Таблица 1.358 – Карта регистров USB_AUX

Адрес	Название	Доступ	Значение после сброса	Описание
0x800	USB_AUX_CTRL_0	ЧТ/ЗП	0x0	Управление сигналами USB контроллера
0x804	USB_AUX_CTRL_1	ЧТ/ЗП	0xFFFFFFFF	Управление сигналами USB контроллера

1.3.9.9.6.3 Описание полей регистров

1.3.9.9.6.3.1 Регистр USB_AUX_CTRL_0

Смещение адреса 0x800

В таблице 1.359 представлено описание полей регистра.

Таблица 1.359 – Описание полей регистра USB_AUX_CTRL_0

Поле	Название	Программный доступ	Состояние после сброса	Описание
0	DBG_SUPPORT	ЧТ/ЗП	0	управление сигналом контроллера dbg_support
1	DMA_NO_SYNC	ЧТ/ЗП	0	управление сигналом контроллера dma_no_sync
2	XTD_DBG_SUPPORT	ЧТ/ЗП	0	управление сигналом контроллера xtd_dbg_support
3	ID_DIG_PIN	ЧТ	0	значение ID_DIG, установленное на внешнем пине (режим работы контроллера) 0 - USB Host; 1 - USB Device.
4	ID_DIG_SW	ЧТ/ЗП	0	программно установленное значение ID_DIG (режим работы контроллера): 0 - USB Host; 1 - USB Device.
5	ID_DIG_MODE	ЧТ/ЗП	0	выбор источника ID_DIG: 0 - поле ID_DIG_SW; 1 - поле ID_DIG_PIN (т.е. внешний пин).
6	OVER_CURRENT	ЧТ/ЗП	0	управление сигналом контроллера over_current
7	V_SESS_VLD	ЧТ/ЗП	0	управление сигналом контроллера v_sess_vld
8	VB_SESS_END	ЧТ/ЗП	0	управление сигналом контроллера vb_sess_end
9	VBUS_VLD	ЧТ/ЗП	0	управление сигналом контроллера vbus_vld
10	VB_ON	ЧТ	0	текущее значение сигнала vb_on
11	CHRG_VBUS	ЧТ	0	текущее значение сигнала chrg_vbus
12	DISCHRG_VBUS	ЧТ	0	текущее значение сигнала dischrg_vbus
13	DLP_ACTIVE	ЧТ	0	текущее значение сигнала dlp_active
14	CORE_SUSPEND_N	ЧТ	0	текущее значение сигнала core_suspend_n
15	XCVR_RCV	ЧТ	0	текущее значение сигнала xcvr_rcv
16	RPU_OPT	ЧТ/ЗП	0	управление сигналом Phy RPU_OPT

1.3.9.9.6.3.2 Регистр USB_AUX_CTRL_1

Смещение адреса 0x804

В таблице 1.360 представлено описание полей регистра.

Таблица 1.360 – Описание полей регистра USB_AUX_CTRL_1

Поле	Название	Программный доступ	Состояние после сброса	Описание
31:0	ALT_BUFFER_SUPPORT	ЧТ/ЗП	0xFFFFFFFF	управление сигналом контроллера alt_buffer_support

1.3.9.9.7 Интеграция в микроконтроллере

Контроллер USB_FS использует два внутренних тактовых сигнала микроконтроллера:

1) usb_fs_clk - от этой частоты работает ядро контроллера и протокол USB;

2) ahb_clk - на этой частоте работают интерфейсы с системой.

USB_FS допускает независимые значения этих двух частот.

При любом выключенном в RCU тактовом сигнале программный доступ в любой регистр будет возвращать ошибку системной шины и вызывать исключение процессорного ядра.

При выключенном в PMU питании контроллера USB_FS программный доступ в любой регистр будет возвращать ошибку системной шины и вызывать исключение процессорного ядра.

Контроллер USB_FS генерирует уровневое прерывание usb_fs_irq.

1.3.9.10 Сборка высокоскоростных контроллеров внешней памяти (HSI)

Сборка высокоскоростных контроллеров внешней памяти (HSI) состоит из контроллера xSPI, контроллера SD/eMMC (SD) и общего Combo Soft PHY.

Структурная схема HSI приведена на рисунке 1.107.

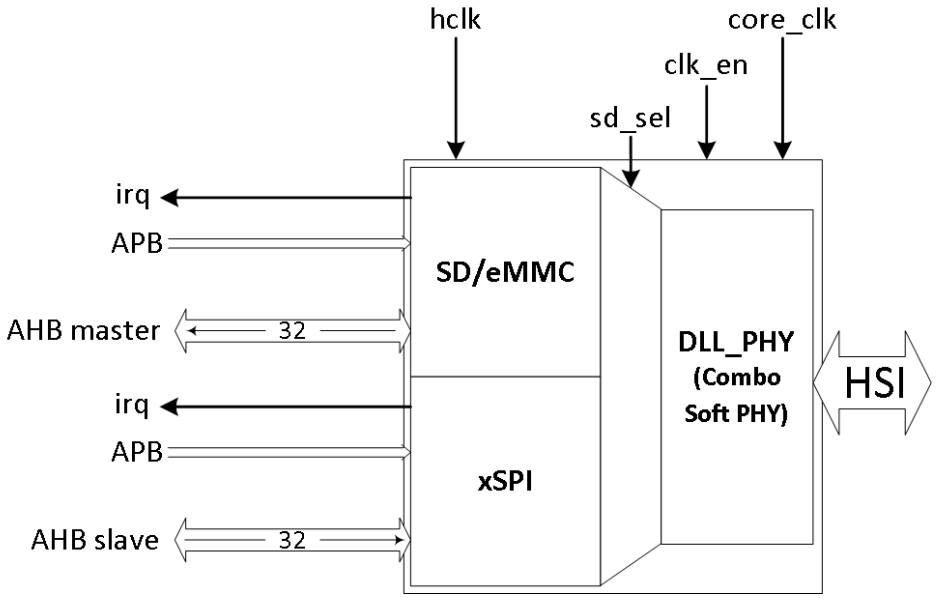


Рисунок 1.107 – Структурная схема HSI

В микросхеме есть 2 одинаковые сборки HSI0 и HSI1.

В один момент времени может работать только один контроллер в сборе HSIx, или SD, или xSPI. Это выбирется битом sd_sel_x (sd_sel_0, sd_sel_1) в регистре SD_SEL блока SYSCR (см. 1.3.6.2).

У HSI есть режимы низкого энергопотребления: . возможность снятия тактирования; . возможность снятия питания.

Внешняя настройка тактировани, задается битом XSPIx_EMMC0_EN (XSPI0_EMMC0_EN, XSPI1_EMMC0_EN) в регистре RCU_PCER1 блока RCU (см. 1.3.7.7.17).

Управление питанием сборок HSI0 и HSI1 задается в PMU (см. 1.3.8).

Программный доступ при снятом тактировании или снятом питании будет возвращать ошибку системной шины и вызывать исключение процессорного ядра.

1.3.9.10.1 Описание внешних выводов HSI

Список внешних выводов HSI и их описание приведены в таблице 1.361.

Таблица 1.361 – Список внешних выводов контроллера SPI

Наименование вывода	Тип (Вх/Вых)	eMMC	SD	xSPI	Функциональное назначение
HSIx_CLK[1)]	Вых	+[2)]	+	+	Линия для передачи синхросигнала
HSIx_CMD	Вх/вых	+	+		Линия для приёма/передачи команд
HSIx_DATA0, HSIx_DATA1, HSIx_DATA2, HSIx_DATA3	Вых	+	+	+	Линии для приёма/передачи данных
HSIx_DATA4, HSIx_DATA5, HSIx_DATA6, HSIx_DATA7	Вых	+		+	Линии для приёма/передачи данных
HSIx_DS	Вх	+		+	Линии для приёма строба значимости данных (Data Strobe)
HSIx_SS0, HSIx_SS1, HSIx_SS2, HSIx_SS3	Вых			+	Линии для выбора Slave устройства (Slave Select)
HSIx_WP	Вых	+	+	+	Линии для сигнала защиты данных от перезаписи (Write Protect)
HSIx_RST	Вых	+		+	Линии для сигнала сброса внешней микросхемы
HSIx_BPWR	Вых	+	+		Линии для сигнала управления питанием внешнего устройства памяти (Bus PoWeR)
HSIx_CDn	Вх	+	+		Линии для сигнала обнаружения внешнего устройства (Card Detect), активным является низкий уровень сигнала
HSIx_PU_PD_DATA2	Вых	+	+		Линии для управления внешней подтяжкой на линии DATA2
HSIx_1v8	Вых		+		Линии для управления напряжением питания на внешней памяти
HSIx_CMD_DIR	Вых		+		Линии для управления направлением внешнего буфера на линии CMD
HSIx_D0_DIR	Вых		+		Линии для управления направлением внешнего буфера на линии DATA0

Наименование вывода	Тип (Вх/Вых)	eMMC	SD	xSPI	Функциональное назначение
HSIx_D123_DIR	Вых		+		Линии для управления направлением внешних буферов на линиях DATA1, DATA2, DATA3

Примечания:

1) "x" - "0" для HSI0 и "1" для HSI1
2) "+" - вывод может использоваться в соответствующем режиме работы

1.3.9.10.2 Контроллер интерфейса xSPI (xSPI)

1.3.9.10.2.1 Обзор архитектуры

Контроллер xSPI предлагает следующие функции:

– Поддержка памяти с помощью интерфейса Legacy SPI

– Поддержка памяти с использование интерфейса HyperBus (HyperFlash или HyperRAM)

– Поддержка памяти, совместимой с JEDEC JESD216 профиля 1 или 2

– Поддержка устройство энергозависимой памяти HyperRAM, pSRAM или Octa-RAM

– Поддержка устройств SPI-NAND

– Интерфейс ведомого устройство AHB и APB

– Прямая адресация пати через AHB-slave интерфейс

– Поддержка режима XIP

– Функции обеспечения целостности данных (CRC или ECC) для поддерживающей их флеш-памяти

– Альтернативный программно-формируемый генератор инструкций (STIG) низкого уровня

– Поддержка маскирования байтов RWDS/DQSM для устройств энергозависимого ОЗУ

– Совместимость xSPI с Combo Soft PHY

– Обнаружение устройств с помощью стандарта Serial Flash Discoverable Parameter (SFDP), определенного в JESD216

На рисунке 1.108 показана структурная схема контроллера xSPI.

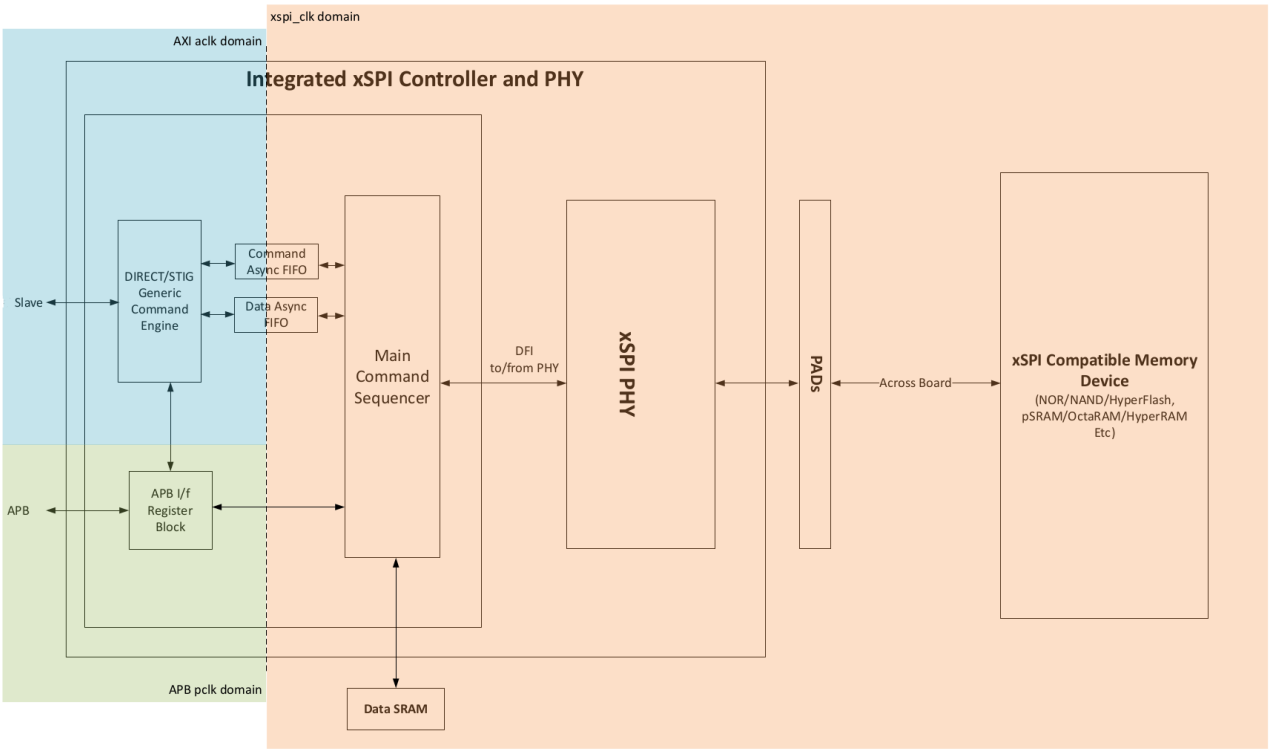


Рисунок 1.108 – Структурная схема контроллера xSPI с PHY

Интерфейс APB используется для доступа к регистрам конфигурации и статуса.

AHB-slave используется в режиме DIRECT, в режиме STIG (по необходимости).

Используется 2 FIFO для асинхронного перехода между тактовыми частотами AHB и xSPI. "FIFO данных" требуется во всех режимах работы. "FIFO команды" требуется для режима STIG.

В контроллере используется SRAM памяти для буферизации данных.

"Main command sequencer" отвечает за построение структуры команд и соблюдение требуемого времени, а также взаимодействует с PHY и регулирует поток данных между физическим уровнем и системным интерфейсом.

PHY отвечает за обработку низкоуровневой синхронизации сигналов данных, адреса и управления.

1.3.9.10.2.1.1 Основные режимы использования

Контроллер xSPI может работать в одном из двух режимов работы DIRECT или STIG. Они могут выполнять стандартные операции чтения/записи в память устройства.

1.3.9.10.2.1.1.1 Режим DIRECT

В режиме DIRECT операции чтения и записи приходят в этот контроллер через интерфейс AHB-slave и напрямую переводятся в операции чтения и записи на запоминающее устройство. Иногда этот процесс называют eXecute-In-Place. Он позволяет выполнить код напрямую или просто получить доступ к подключенному запоминающему устройству. Обратите внимание, что иногда процесс eXecute-In-Place можно спутать с устройством, работающим в так называемом режиме “XIP” (иногда его называют непрерывным режимом). XIP — это функция, которая есть в некоторых устройствах для улучшения качества чтения, особенно операций чтения с произвольным доступом, происходящих при выполнении кода напрямую из запоминающего устройства. Контроллер действительно поддерживает устройства, работающие в режиме XIP (во всех рабочих режимах, не только DIRECT), однако хосту необходимо выполнить ряд шагов, чтобы войти в режим XIP и выйти из него.

Эти шаги описаны в главе 1.3.9.10.2.2.1 настоящего документа.

Подробнее об использовании режима DIRECT и об ограничениях см. пп 1.3.9.10.2.2.3 "Работа в режиме DIRECT".

1.3.9.10.2.1.1.2 Режим STIG

STIG (программно-формируемый генератор инструкций) — это режим, который настраивается и запускается посредством программного интерфейса. Это гибкий и доступный для конфигурации режим. Он не только выполняет стандартные операции чтения и записи, но и позволяет получать доступ к регистрам запоминающего устройства и/или запускать в запоминающем устройстве функции, например стирание.

Режим STIG использует пути данных, аналогичные режиму DIRECT. Однако команды настраиваются и запускаются посредством регистров APB. Данные могут передаваться либо из AHB-slave интерфейса, либо через регистры по APB.

Подробнее об использовании режима STIG и об ограничениях см. пп 1.3.9.10.2.2.5 "Работа в режиме STIG".

1.3.9.10.2.1.2 Поддерживаемые типы запоминающих устройств

Контроллер xSPI разработан в соответствии со спецификацией xSPI JESD216, поэтому поддерживает любое устройство, совместимое с этим стандартом. В этом стандарте эти устройства относятся к типу профиля 1 или 2. Согласно этому стандарту, тбольшинство устройств с последовательной флеш-памятью NOR совместимы с типом профиля 1. Устройства,

поддерживающие HyperBus (большинство устройств HyperFlash, HyperRAM, pSRAM) обычно совместимы с типом профиля 2.

Контроллер поддерживает следующие типы устройств:

1) SPI (single/dual/quad/octal)

a) Такие устройства аналогичны профилю 1 в спецификации JEDEC JESD216 и могли быть произведены до выпуска этой спецификации, либо производитель просто решил не указывать совместимость с xSPI в документации.

2) HyperFlash

a) Такие устройства аналогичны профилю 2 в спецификации JEDEC JESD216 и могли быть произведены до выпуска этой спецификации, либо производитель просто решил не указывать совместимость с xSPI в документации.

3) Любые устройства, совместимые с "xSPI профиля 1", где профиль 1 описан в спецификации xSPI JEDEC JESD216.

4) Любые устройства, совместимые с "xSPI профиля 2", где профиль 2 описан в спецификации xSPI JEDEC JESD216.

5) Последовательные устройства SRAM (например, OctaRam, HyperRAM, pSRAM). Обмен данными с этими энергозависимыми ОЗУ осуществляется так же, как с устройствами профиля 1 или 2, однако для сохранения состояния им требуется время от времени самообновляться. Кроме того, им не требуются некоторые элементы последовательностей, используемых для чтения или программирования традиционных устройств флеш-памяти.

6) Последовательные устройства NAND (иногда их называют SPI-NAND или NAND-SPI).

1.3.9.10.2.2 Функциональные возможности

1.3.9.10.2.2.1 XIP

Режим работы, который заставляет контроллер xSPI выполнять код напрямую из подключенного запоминающего устройства - **eXecute-In-Place**.

Термин XIP может использоваться в другом контексте. Многие устройства флеш-памяти могут работать в режиме, который в их документации описан как "XIP" (иногда его также называют непрерывным режимом). Режим XIP избавляет контроллер от необходимости отправлять код операции чтения для любой операции чтения, что оптимизирует процесс чтения. Если обычно код операции чтения передается в течение 8 периодов тактового сигнала

устройства (SDR, последовательная передача кода операции), то задержка операции чтения сокращается на 8 тактов. Это может заметно улучшить производительность при выполнении кода прямо из устройства. С другой стороны, если войти в режим XIP, устройство сможет выполнять только операции READ. Если требуется другая операция, необходимо выйти из режима XIP. Контроллер поддерживает режим XIP, однако контроллер xSPI должен выполнить ряд шагов, чтобы войти в режим XIP и выйти из него. Перечень этих шагов может отличаться в зависимости от выбранного рабочего режима.

1.3.9.10.2.2.2 Настройка контроллера

Чтобы использовать контроллер в режиме DIRECT, STIG, необходимо настроить несколько регистров конфигурации. Они начинаются со смещения 0x380.

1) Выбрать, какому профилю соответствует подключенное устройство. Термин "профиль" описан в спецификации JEDEC JESD216. Обычно, устройства последовательной флеш-памяти на базе NOR относятся к профилю 1, а устройства HyperFlash / HyperRAM / PSRAM относятся к профилю 2.

Примечание.

– Контроллер относит HyperFlash к подклассу профиля 2 – HF, а HyperRAM/ PSRAM к подклассу профиля 2 – HR.

– Профиль 1 — это режим по умолчанию, который выбирает контроллер.

2) Если вам потребуется вывести регистры из их состояния по умолчанию, настройте поля global_seq_cfg (0x390).

3) Для устройств SPI NAND необходимо обновить регистр global_seq_cfg_1 (0x394).

4) Настройте, как контроллер должен отправлять последовательность READ или PROGRAM на запоминающее устройство зависит от подключенного запоминающего устройства, поэтому изучите документацию этого устройства.

На некоторых устройствах перед операцией PROGRAM может понадобиться выполнить подпоследовательность включения записи (например, настроить в устройстве регистр включения записи (WEL)). По умолчанию предполагается, что это действительно необходимо запоминающему устройству, но это можно отключить, записью в регистр we_seq_cfg_0 (0x440).

В некоторых случаях после выполнения операции требует провести опрос устройства на корректность и готовность. Поскольку разные типы устройств по-разному интерпретируют и отвечают на запросы статуса и существуют разные типы запросов статуса, контроллер требует настроить несколько программируемых регистров в соответствии с требованиями

устройства. Для режима DIRECT контроллер может дополнительно проверить статус ошибки или занятости устройства после операции PROGRAM или ERASE (или дополнительно для SOFT_RESET/READ на устройствах SPI NAND). Для устройств SPI NAND контроллер также может проверить состояние ECC после команды READ PAGE. Все эти параметры настраиваются в регистрах stat_seq_cfg_*, начиная со смещения 0x450.

Подробнее об использовании различных рабочих режимов см. в посвященных им разделах. В описаниях регистров вы найдете инструкции о том, как программировать эти поля. Если понадобится изменить любой из параметров, необходимо повторно инициализировать регистры stat_seq_cfg_*.

В таблице 1.362 приведены все регистры seq_cfg, которые необходимо инициализировать при работе в режиме DIRECT.

Таблица 1.362 – Информация о регистрах seq_cfg

	Устройства профиля 1	**Устройства профиля 2 – HyperFlash**	**Устройства профиля 2 – HyperRam**	**Устройства SPI NAND**
Операции PROGRAM	prog_seq_cfg_0 (0x420) prog_seq_cfg_1 (0x424)	prog_seq_cfg_2 (0x420)	prog_seq_cfg_2 (0x420)	prog_seq_cfg_0 (0x420) prog_seq_cfg_1 (0x424)
Операции READ	read_seq_cfg_0 (0x430) read_seq_cfg_1 (0x434)	read_seq_cfg_2 (0x438)	read_seq_cfg_2 (0x438)	read_seq_cfg_0 (0x430) read_seq_cfg_1 (0x434)
Операция включения записи	we_seq_cfg_0 (0x440)	Н/Д	Н/Д	we_seq_cfg_0 (0x440)
Проверка состояния сбоя устройства Проверка состояния занятости устройства	stat_seq_cfg_0 (0x450) stat_seq_cfg_1 (0x454) stat_seq_cfg_2 (0x458) stat_seq_cfg_3 (0x45c) stat_seq_cfg_5 (0x464) stat_seq_cfg_7 (0x46c) stat_seq_cfg_8 (0x470) stat_seq_cfg_9 (0x474)	stat_seq_cfg_4 (0x460) stat_seq_cfg_5 (0x464) stat_seq_cfg_7 (0x46c) stat_seq_cfg_8 (0x470) stat_seq_cfg_9 (0x474)	Н/Д	stat_seq_cfg_0 (0x450) stat_seq_cfg_1 (0x454) stat_seq_cfg_2 (0x458) stat_seq_cfg_3 (0x45c) stat_seq_cfg_5 (0x464) stat_seq_cfg_7 (0x46c) stat_seq_cfg_8 (0x470) stat_seq_cfg_9 (0x474)
Проверка состояния ECC	Н/Д	Н/Д	Н/Д	stat_seq_cfg_10 (0x478)

1.3.9.10.2.2.3 Работа в режиме DIRECT

В режиме DIRECT операции чтения и записи от стороннего хоста вносятся в этот контроллер через AHB-salve и напрямую переводятся в операции чтения и записи на запоминающем устройстве.

Режим DIRECT включается по умолчанию и контролируется посредством поля work_mode регистра ctrl_config (0x230). Этот регистр можно изменить, только когда контроллер в состоянии IDLE. Для этого необходимо опросить **ctrl_busy** регистра ctrl_status (0x100). Когда включен режим DIRECT, регистры cmd_reg* (которые требуются только для других рабочих режимов) игнорируются, их следует избегать. Транзакции чтения и программирования/записи в запоминающее устройство будут выполняться только как обращения от системы по AHB-salve.

Примечание. В режиме DIRECT размер страницы для транзакции записи или чтения должен быть настроен шире, чем выбранный размер слова операции AHB (HSIZE). Кроме того, границы страницы xSPI должны быть выровнены с размером слова операции AHB (HSIZE).

Также необходимо програмировать регистры режима DIRECT - 0x398, 0x39c и 0x3a0.

1.3.9.10.2.2.3.1 Ограничения записи AHB в режиме DIRECT

По умолчанию контроллер не поддерживает следующее:

– Побайтовые записи (т. е. запись AHB со значением HSIZE равным "0") в устройство с 16-битной адресацией или при работе в режиме DDR octal.

– Отправка записи на нечетный адрес с 16-битной адресацией или при работе в режиме DDR octal.

– Использование стробов записи AHB для маскирования байтов на шине записи.

Для поддержки вышеуказанного алгоритма можно настроить rwds_byte_mask_cap (бит 4 регистра direct_access_cfg (0x398)), если подключенное устройство поддерживает маскирование байтов через сигнал DQS. Как правило, с помощью DQS запоминающее устройство передает строб с данными чтения для чтения выборок данных. Однако некоторые устройства используют этот сигнал в качестве маски байта для операций записи, что позволяет избежать описанных выше ограничений.

1.3.9.10.2.2.3.2 Активация и деактивация режима XIP в режиме DIRECT

Как описано в пп 1.3.9.10.2.2.1 "XIP", некоторые запоминающие устройства поддерживают "режим XIP" (непрерывный режим). Поля mode_bit_xip и mode_bit_xip_dis контролируют способ и время активации и деактивации режима XIP на устройстве при использовании режима DIRECT. По умолчанию режим XIP не активирован.

Прежде чем работать с запоминающим устройством в режиме XIP, сначала обратитесь к технической документации памяти и ознакомьтесь с тем, как следует активировать и

деактивировать режим XIP. Контроллер предлагает параметры для активации и деактивации режима, которые отличаются достаточной гибкостью. Чтобы включить XIP на запоминающем устройстве, необходимо настроить определенный бит в регистре конфигурации/состояния памяти (этот шаг является обязательным не для всех устройств флеш-памяти, однако этого можно быстро добиться в режиме STIG), а затем отправить команду чтения с верным значением в байте режима, расположенном сразу после фазы адреса в операции чтения (см таблицу 1.363).

Для этого войдите в режим XIP из режима DIRECT — установите бит **mode_bit_xip_en** в регистре direct_access_cfg (0x398) и выполните операцию чтения AHB. После того как операция чтения будет выполнена, а последовательность READ будет отправлена на устройство, контроллер отправит значение, зашифрованное в поле **xip_en_mb_val** из регистра **direct_access_cfg**, сразу после фазы адреса.

После завершения операции чтения бит **xip_en** в регистре xip_mode_cfg (0x388) будет установлен в 1. Это будет означать, что контроллер и запоминающее устройство настроены для работы в режиме XIP.

Обратите внимание, что первая операция чтения для включения режима XIP в контроллере и запоминающем устройстве не должна содержать больше 64Б данных во избежание переносов между страницами.

Чтобы покинуть режим XIP как на контроллере, так и на запоминающем устройстве, необходимо отправить новую операцию READ на устройство, (но при этом добавить "бит отключения режима XIP) после фазы адреса. Для отключения режима XIP необходимо установить бит **mode_bit_xip_dis** в регистре direct_access_cfg (0x398) и отправить новую операцию чтения по AHB. Когда будет выполнена операция чтения, а последовательность READ будет отправлена на устройство, после фазы адреса в операции контроллер отправит значение, зашифрованное в поле **xip_en_dis_val** регистра **direct_access_cfg** (0x398). По завершении бит xip_en в регистре xip_mode_cfg (0x388) устанавливается в 0. Это означает, что контроллер и запоминающее устройство вышли из режима XIP и снова готовы к работе в нормальном режиме.

Обратите внимание, что первая операция для отключения режима XIP в контроллере и запоминающем устройстве не должна содержать больше 64Б данных во избежание переносов между страницами.

Таблица 1.363 – Нормальная/XIP-операция чтения в режиме DIRECT

xip_en	mode_bit_xip_en	mode_bit_xip_dis	read_seq_p1_mb_en	Описание
0	0	0	0	Режим нормального чтения без байта режима (количество пустых тактов: **read_seq_p1_dummy_cnt**).
0	0	1	0	Режим нормального чтения с дополнительным байтом отключения режима после фазы адреса (количество пустых тактов: **read_seq_p1_mb_dummy_cnt**).
0	0	0	1	Режим нормального чтения с дополнительным байтом отключения режима после фазы адреса (количество пустых тактов: **read_seq_p1_mb_dummy_cnt**).
0	1	-	-	Включает режим XIP в контроллере и запоминающем устройства при следующей операции чтения — байт включения режима передается после фаз кода операции и адреса. После этого бит xip_en устанавливается в 1 (количество пустых тактов: **read_seq_p1_mb_dummy_cnt**).
1	0	0	-	Контроллер выдает ошибку **dir_cmd_err**. Когда контроллер настроен на работу в режиме XIP, должен быть установлен или mode_bit_xip_en, или mode_bit_xip_dis.
1	0	1	-	Отключение режима XIP в контроллере и запоминающем устройства при следующей операции чтения — байт выключения режима передается после фазы адреса. После этого бит xip_en устанавливается в 0 (количество пустых тактов: **read_seq_p1_mb_dummy_cnt**).
1	1	-	-	Контроллер работает в режиме XIP — байт включения режима передается после фазы адреса (количество пустых тактов: **read_seq_p1_mb_dummy_cnt**).
0	0	1	1	Режим нормального чтения с дополнительным байтом отключения режима после фазы адреса (количество пустых тактов: **read_seq_p1_mb_dummy_cnt**).

1.3.9.10.2.2.4 Последовательности команд в режиме DIRECT

В режиме DIRECT хост может выполнить операции PROGRAM или READ и поручить контроллеру выстроить их в последовательность. При работе в этих режимах предполагается, что вместе с ними будет использоваться режим STIG, который позволит выполнять команды или целые последовательности, не доступные в основном режиме передачи данных.

1.3.9.10.2.2.4.1 Последовательность PROGRAM (в режиме DIRECT)

Последовательность PROGRAM позволяет выполнять операции записи — перенос данных из хоста в устройство xSPI.

Во время выполнения последовательности PROGRAM в режиме DIRECT всегда используется AHB-salve, количество данных для передачи совпадает с количеством байтов в операции записи AHB, а адрес основан на адресе AHB.

Последовательность PROGRAM поддерживается для устройств профиля 1, 2 – HyperFlash, устройств профиля 2 – HyperRAM и SPI NAND.

На схеме 1.109 вы можете увидеть, как последовательность PROGRAM отображается в интерфейсе xSPI для устройств профиля 1.

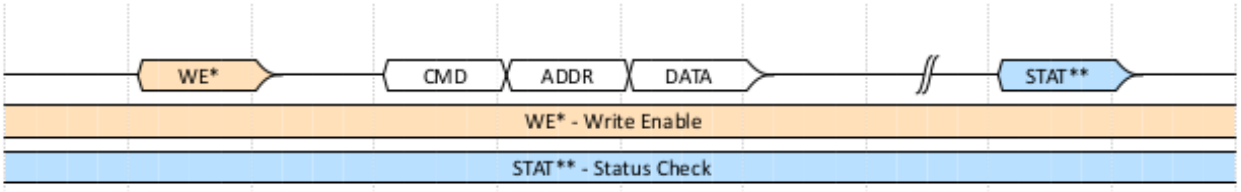


Рисунок 1.109 – Последовательность PROGRAM (профиль 1)

Последовательность PROGRAM начинается с выполнения операции WE (включение записи), если она включена посредством регистра we_seq_cfg_0 (0x440). Затем выполняет фазы команды, адреса и данных. Каждую фазу можно настроить с помощью регистров конфигурации **prog_seq_cfg_0 (0x0420)** и **prog_seq_cfg_1 (0x0424)**. Когда все байты данных будут отправлены, контроллер выполнит подпоследовательность STAT, чтобы убедиться, что данные сохранены в устройстве и оно готово для другой команды. Подпоследовательность STAT можно настроить с помощью регистров stat_seq_cfg_* (смещения адреса начиная с 0x450).

На схеме 1.110 вы можете увидеть, как последовательность PROGRAM отображается в интерфейсе xSPI для устройств профиля 2.

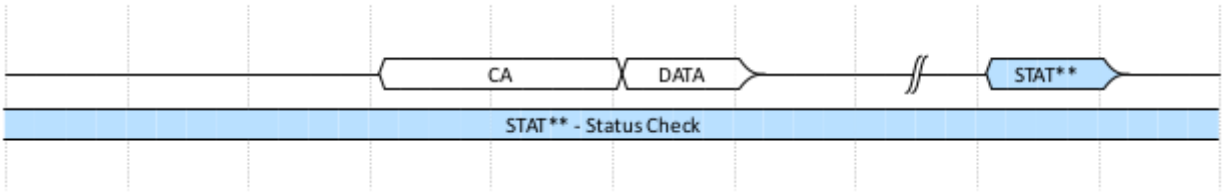


Рисунок 1.110 – Последовательность PROGRAM (профиль 2 — HF, профиль 2 — HR)

Последовательность программирования начинается с отправки команды/адреса (CA) и фазы данных. Эти фазы можно настроить с помощью регистра конфигурации **prog_seq_cfg_2 (0x0428)**. Фаза CA — это 48-битный вектор, структура которого представлена в таблице 1.364.

Примечание. В таблице 1.364 показано 13 зарезервированных битов, начинающихся с позиции 0x3. Хотя это стандартно, контроллер xSPI позволяет пользователю при необходимости изменить количество зарезервированных битов и начальную позицию с помощью регистра hf_offset_reg (0x1020). Также для команд профиля 2 — HF и профиля 2 — HR используется только 8D-8D-8D.

Таблица 1.364 – Примеры команд программирования

Название	**Биты**	**Значение**
Направление (R/W)	47	Установлено значение 0
Адресное пространство (R/M)	46	Согласно **prog_seq_p2_target**
Тип Bursts	45	Согласно **prog_seq_p2_burst_type**
Расширение команды	44:40	Если **prog_seq_p2_mask_cmd_mod**=0: все 0. В противном случае — все 1.
Адрес строки и верхнего столбца	39:16	Биты адреса ADDR[26:3].
Зарезервировано	15:3	Зарезервированы — все 0.
Адрес нижнего столбца	2:0	Биты адреса ADDR[2:0].

Когда все байты данных будут отправлены, контроллер выполнит подпоследовательность STAT, чтобы убедиться, что данные сохранены в устройстве. Подпоследовательность STAT можно настроить с помощью регистров stat_seq_cfg_* (смещения адреса начиная с 0x450).

На схеме 1.111 вы можете увидеть, как последовательность PROGRAM отображается в интерфейсе xSPI для устройств SPI NAND.

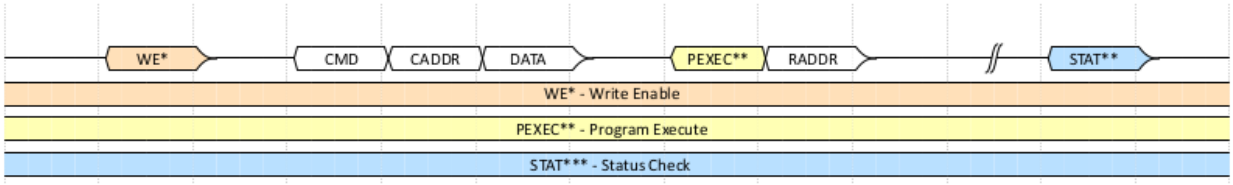


Рисунок 1.111 – Последовательность PROGRAM (SPI NAND)

Эта последовательность начинается с операции WE (включение записи). Далее она отправляет команду загрузки программы с фазой адреса столбца и данных. Эту фазу можно настроить с помощью регистров конфигурации **prog_seq_cfg_0 (0x0420)** и **prog_seq_cfg_1 (0x0424)**. Отправляется команда PEXEC (выполнение программы — 10h), после которой

следуют адрес строки (страницы) (на этой фазе не требуется конфигурация — в 1-битном режиме SDR используется 24-битный адрес строки). Когда все данные отправлены, контроллер выполняет подпоследовательность STAT, чтобы убедиться, что данные сохранены на устройстве (хост должен настроить последовательность STAT так, чтобы она соответствовала команде GetFeature). Подпоследовательность STAT можно настроить с помощью регистров stat_seq_cfg_* (смещения адреса начиная с 0x450).

1.3.9.10.2.2.4.2 Последовательность READ (в режиме DIRECT)

Последовательность READ поддерживается для устройств профиля 1, 2 – HyperFlash, устройств профиля 2 – HyperRAM и SPI NAND. На схеме 1.112 вы можете увидеть, как последовательность READ отображается в интерфейсе xSPI для устройств профиля 1.

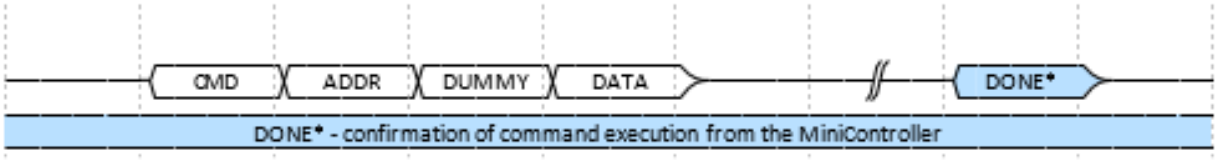


Рисунок 1.112 – Последовательность READ — тип 0 (профиль 1)

Сначала контроллер отправляет фазу команды и адреса. После этого следуют дополнительные пустые такты и данные для чтения. Каждую фазу можно настроить с помощью регистров конфигурации **read_seq_cfg_0 (0x0430)** и **read_seq_cfg_1 (0x0434)**.

На схеме 1.113 вы можете увидеть, как последовательность READ отображается в интерфейсе xSPI для устройств профиля 2 (HyperFlash или HyperRAM).

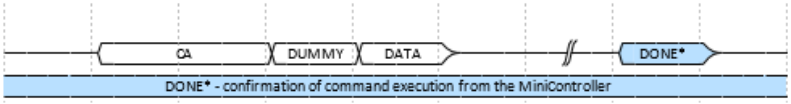


Рисунок 1.113 – Последовательность READ (профиль 2 — HF, профиль 2 — HR)

Последовательность чтения начинается с отправки команды/адреса (CA). После этого следуют дополнительные пустые такты и обязательная фаза данных. Фаза CA — это 48-битный вектор, структура которого представлена в таблице 1.365. Эту фазу можно настроить с помощью регистра конфигурации **read_seq_cfg_2 (0x0438)**. Для команд профиля 2 — HF и профиля 2 — HR используется только 8D-8D-8D.

Таблица 1.365 – Значение фазы команды/адреса для команды чтения (профиль 2 — HF, профиль 2 — HR)

Название	**Биты**	**Значение**
Направление (R/W)	47	Установлено значение 1
Адресное пространство (R/M)	46	Согласно **read_seq_p2_target**
Тип Bursts	45	Согласно **read_seq_p2_burst_type**
Расширение команды	44:40	Если **read_seq_p2_mask_cmd_mod**=0: все 0. В противном случае — все 1.
Адрес строки и верхнего столбца	39:16	Биты адреса ADDR[26:3].
Зарезервировано	15:3	Зарезервированы — все 0.
Адрес нижнего столбца	2:0	Биты адреса ADDR[2:0].

На схеме 1.114 вы можете увидеть, как последовательность READ отображается в интерфейсе xSPI для устройств SPI NAND.

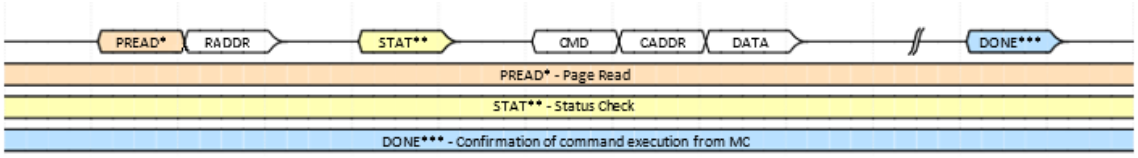


Рисунок 1.114 – Последовательность READ — тип 3 (SPI NAND)

Сначала контроллер отправляет команду чтения страницы (PREAD) с фазой адреса строки (страницы). Далее выполняется подпоследовательность STAT с целью проверить, загружено ли содержимое страницы в регистр кэша(хост должен настроить последовательность STAT в соответствии с командой Get Feature). Подпоследовательность STAT можно настроить с помощью регистров stat_seq_cfg_* (смещения адреса начиная с 0x450). В следующей фазе передается настраиваемая команда, после которой следуют дополнительные пустые такты и адрес столбца. Таким образом можно отправить команду чтения из кэш-памяти.

Для **read_seq_p1_cache_random_read_en** = 1 контроллер выполняет последовательность, представленную на схеме 1.115.

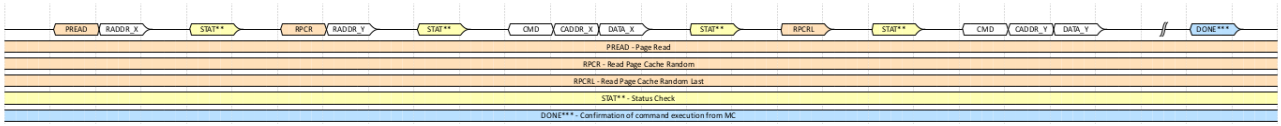


Рисунок 1.115 – Последовательность READ — тип 3 (SPI NAND) с Read Page Cache Random

В этом последовательности выполняются следующие шаги:

1) Команда чтения страницы (13h) выполняется с адресом первой страницы (страница X)

2) Команда Get Feature собирает информацию о состоянии, проверяется бит OIP

3) Команда Read Page Cache Random (30h) выполняется с адресом следующей страницы (страница Y)

4) Команда Get Feature собирает информацию о состоянии, проверяется бит OIP

5) Команда чтения из кэша позволяет получить данные со страницы X

6) Команда Get Feature собирает информацию о состоянии, проверяется бит CRBSY

7) Шаги с 3 по 6 необходимо повторить, чтобы прочитать все ожидаемые страницы до последней

8) Выполняется команда 8 Read Page Cache Random Last (3Fh)

9) Команда Get Feature собирает информацию о состоянии, проверяется бит OIP

10) Команда чтения из кэша позволяет получить данные последней страницы

Фазы последовательностей команд чтения из кэша страницы, которые связаны с передачей данных. Адрес столбца может быть настроен с помощью регистров конфигурации **read_seq_cfg_0 (0x0430)** и **read_seq_cfg_1 (0x0434)**. Фазы с адресом строки не требуют конфигурации — передается 24-битный адрес с помощью одного бита данных и режима SDR.

1.3.9.10.2.2.4.3 Последовательность INTERNAL WRITE ENABLE (WE)

Контроллер выполняет последовательность INTERNAL WRITE ENABLE (см. предыдущие временные диаграммы), чтобы включить бит WEL для операций PROGRAM. Эта последовательность передается только для устройств профиля 1 и SPI NAND (т. е. в поле **seq_type** в регистре **global_seq_cfg (0x0390)** устанавливается значение от 0 до 3).

На схеме 1.116 вы можете увидеть, как последовательность WE отображается в интерфейсе xSPI для устройств профиля 1.

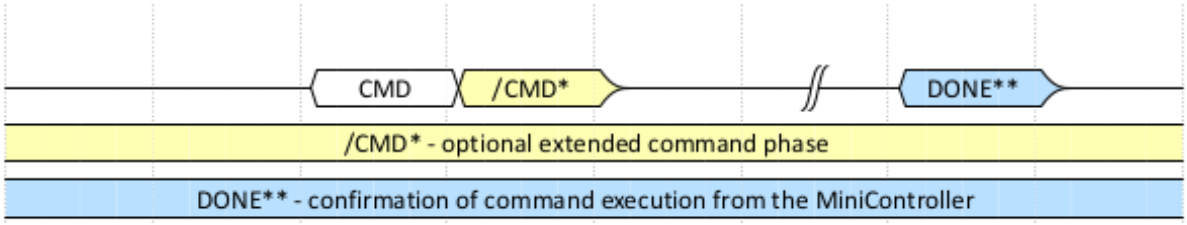


Рисунок 1.116 – Последовательность INTERNAL WRITE ENABLE (WE)

Контроллер отправляет только мнемоническую схему команды и дополнительное расширение команды. Последовательность можно настроить с помощью регистра **we_seq_cfg_0 (0x0440)**.

1.3.9.10.2.2.4.4 Последовательность INTERNAL STATUS (STAT)

С помощью последовательности INTERNAL STATUS контроллер проверяет, успешно ли завершены операции PROGRAM. Эта последовательность использует группу параметров конфигурации, хранящуюся в регистре **stat_seq_cfg_5**. Эти параметры описывают, как контроллер интерпретирует полученное значение состояния:

– **stat_seq_dev_rdy_idx** — позиция бита, которая предоставляет информацию о готовности/занятости после выполнения команды PROGRAM.

– **stat_seq_dev_rdy_val** — противоположность состояния готовности/занятости после выполнения команды PROGRAM. Если состояние, считанное с устройства, отвечает значению этого бита, контроллер предполагает, что устройство готово.

– **stat_seq_dev_rdy_size** — количество битов состояния, которое считывает контроллер (8 или 16 битов), чтобы проверить состояние готовности устройства после команды PROGRAM.

– **stat_seq_prog_fail_idx** — расположение бита, который предоставляет информацию о состоянии сбоя после команды PROGRAM.

– **stat_seq_prog_fail_val** — противоположность информации о состоянии сбоя после команды PROGRAM. Если состояние, считанное с устройства, отвечает значению этого бита, контроллер предполагает, что операция прервана.

– **stat_seq_prog_fail_size** — количество битов состояния, которые считываются (8 или 16 битов) с целью проверить состояние сбоя устройства после команды PROGRAM.

Примечание. Если для проверки состояния готовности/занятости и состояния сбоя используется одна и та же команда, контроллер выполняет одну команду состояния в интерфейсе устройства. В противном случае отправляются две отдельные команды состояния — сначала для проверки состояния готовности/занятости, а затем — для проверки состояния сбоя.

Также обратите внимание, что если во время проверки состояния возникнет ошибка DQS, контроллер предполагает, что устройство выдало состояние готовности. В этом случае сообщение об ошибке устройства не отправляется.

На схеме 1.117 вы можете увидеть, как последовательность STAT отображается в интерфейсе xSPI для устройств профиля 1 или SPI NAND.

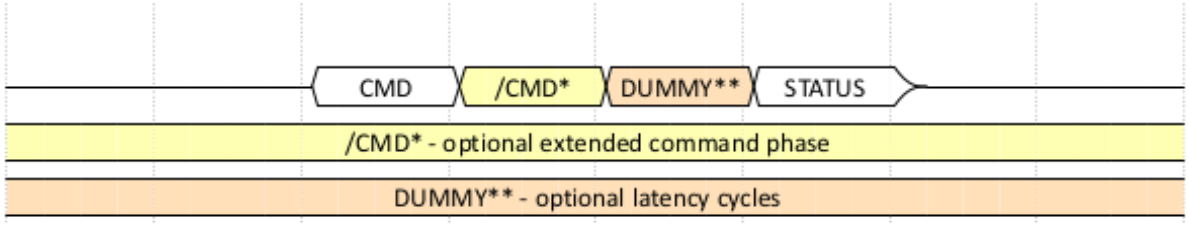


Рисунок 1.117 – Последовательность ISEQ_STAT — тип 0/3 (устройства профиля 1 и SPI NAND)

В этой последовательности контроллер отправляет только код операции команды. Считываются 8 или 16 битов состояния. С ее помощью можно прочитать регистр состояния или регистр "флагов" с устройства xSPI. Эту последовательность можно настроить с помощью регистров **stat_seq_cfg_0 (0x0450)**, **stat_seq_cfg_1 (0x0454)**, **stat_seq_cfg_2 (0x0458)** и **stat_seq_cfg_3 (0x045c)**.

На схеме 1.118 вы можете увидеть, как последовательность STAT отображается в интерфейсе xSPI для устройств профиля 2 — HyperFlash.

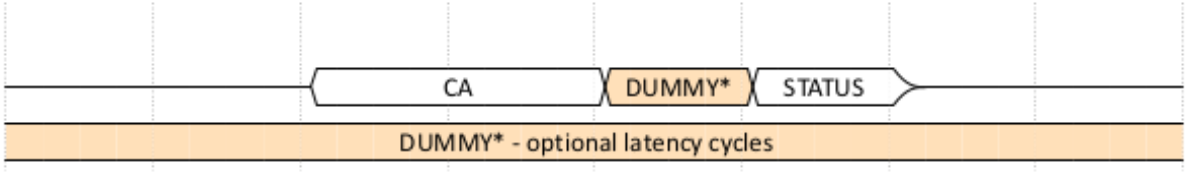


Рисунок 1.118 – Последовательность ISEQ_STAT — тип 1 (устройства профиля 2 — HF)

Последовательность начинается с отправки команды/адреса (CA). После этого следуют дополнительные пустые такты и обязательная фаза данных. Фаза CA — это 48-битный вектор, структура которого представлена в таблице 1.366. Эту фазу можно настроить с помощью регистра конфигурации **stat_seq_cfg_4 (0x0460)**. Для команд профиля 2 — HF используется только 8D-8D-8D.

Таблица 1.366 – Значение фазы команды/адреса для команды чтения (профиль 2 — HF)

Название	**Биты**	**Значение**
Направление (R/W)	47	Установлено значение 1
Адресное пространство (R/M)	46	Установлено значение 0
Тип Bursts	45	Установлено значение 1
Расширение команды	44:40	Все биты устанавливаются на значение, заданное посредством **stat_seq_p2_mask_cmd_mod**.

Название	Биты	Значение
Адрес строки и верхнего столбца	39:16	Биты [26:3] адреса 0x555 (0x0000AA).
Зарезервировано	15:3	Зарезервированы — все 0.
Адрес нижнего столбца	2:0	Биты [2:0] адреса 0x555 (0x5).

1.3.9.10.2.2.5 Работа в режиме STIG

1.3.9.10.2.2.5.1 Обзор режима STIG

STIG — или программно-формируемый генератор инструкций — это режим, позволяющий ПО определяет какие низкоуровневые команды и последовательности отправлять запоминающему устройству. По сравнению с режимом DIRECT он предлагает больше гибкости, поскольку ПО может контролировать низкоуровневые характеристики каждой операции. Однако этот способ работает в основном за счет ПО. Предполагается, что режим STIG может нормально работать вместе с DIRECT.

Чтобы включить режим STIG, установите значение 1 в поле **work_mode** в регистре **ctrl_config (0x0230)**. Это можно сделать, только когда контроллер находится в состоянии простоя. Для настройки работы в режиме STIG требуется пять регистров команд: cmd_reg0 (0x0000), cmd_reg1 (0x0004), cmd_reg2 (0x0008), cmd_reg3 (0x000c) и cmd_reg4 (0x0010). Доступ для записи к регистру cmd_reg0 запускает выполнение команды, записывать в него данные следует в последнюю очередь. После запуска хост должен дождаться завершения работы STIG, а для этого необходимо опросить cmd_status (0x0044).

1.3.9.10.2.2.5.2 Регистры команд

ПО хоста настраивает работу STIG с помощью регистров команд, начинающихся с 0x0000 (с cmd_reg0 до cmd_reg4). Запись данных в регистр cmd_reg0 просто запускает работу в режиме STIG и не выполняет никаких других задач. Таким образом, данные в cmd_reg0 следует записывать в последнюю очередь. Если объединить регистры команд с cmd_reg1 по cmd_reg4, они вмещают 128-битную инструкцию для STIG, как показано в таблице 1.367.

Таблица 1.367 – Объединение cmd_reg для инструкции STIG

Биты	Название	Описание
31:0	cmd_reg0	Запись данных в этот регистр запускает работу в режиме STIG и не выполняет никаких других задач. Чтение игнорируется.
31:0	cmd_reg1	Хранит биты [31:0] инструкции STIG
31:0	cmd_reg2	Хранит биты [63:32] инструкции STIG
31:0	cmd_reg3	Хранит биты [95:64] инструкции STIG
31:0	cmd_reg4	Хранит биты [127:96] инструкции STIG

Для отправки команды или последовательности в запоминающее устройство используются одна или несколько инструкций STIG. Можно настроить запуск любой операции, которую предлагает поддерживаемое устройство, что делает инструкции STIG доступными для программирования. В поле "Тип инструкции" каждой инструкции описан тип операции, которую должен отправить хост. Список доступных "Типов инструкции" приведен в разделах ниже. В зависимости от необходимой операции хосту может потребоваться выдать более одной инструкции STIG. Например, чтобы выполнить операцию READ для Legacy SPI-устройства флэш-памяти NOR, хосту необходимо отправить инструкцию STIG, чтобы описать команду (код операции, адрес, биты режима, пустые такты и т. д.), а затем отправить вторую инструкцию STIG, чтобы описать последовательность данных, связанную с операцией чтения. Отправка нескольких связанных инструкций STIG - склеивание инструкций. Буферизация для склеивания инструкции осуществляется в FIFO-инструкции в контроллере. Глубина составляет 16, поэтому можно склеить до 16 инструкций STIG. Попытка склеить более 16 инструкций STIG приведет к ошибке. У контроллера есть два способа решить, следует ли склеивать инструкцию STIG с текущей инструкцией. Первый — через поле "Тип инструкции". Например, некоторым операциям всегда требуется более одной инструкции, последняя из которых является инструкцией данных (например, операция READ). Второй способ — поле INSTR_LINK. Только некоторые операции "Типа инструкции" могут настроить поле INSTR_LINK так, как описано в следующем разделе.

Когда хост выполняет инструкцию STIG, контроллер выполняет его буферизацию в локальном FIFO. Это означает, что хост может выполнить несколько инструкций STIG, а контроллер последовательно обработает их в том порядке, в котором они были получены. Контроллер позаботится о требованиях к синхронизации xSPI (они отличаются в зависимости от типа подключаемой памяти). Однако именно хост должен избегать выполнения недопустимой команды или команды, которую не сможет интерпретировать запоминающее устройство. Аналогичным образом, при выдаче инструкций STIG хост должен проследить за тем, чтобы команды, требующие простоя устройства, отправлялись, только когда запоминающее устройство не занято выполнением предыдущей команды. STIG не сможет проследить за этим автоматически.

STIG поддерживает запись и чтение данных в запоминающее устройство. Для этого используется встроенный FIFO данных. Если поступила операция чтения, а FIFO данных заполнен, контроллер автоматически прервет передачу данных для чтения в память и перезапустит

ее, как только FIFO станет доступным (пересчитав верный новый начальный адрес исходя из объема собранных данных. Некоторые инструкции STIG имеют ограничения по максимальному количеству данных для чтения.

Примечание. xSPI может работать с устройствами HyperRAM, принадлежащими группе устройств профиля 2.0. Наиболее оптимальные последовательности для выполнения операций записи/чтения HyperRAM — CMD_CODE_SEQ в склейке с DATA_SEQ.

Контроллер поддерживает четыре варианта инструкций STIG (см. таблицу 1.368).

Таблица 1.368 – Варианты инструкций STIG

Вариант инструкции STIG	**Описание**
Вариант 1: Legacy SPI, xSPI профиля 1, SPI NAND	Этот вариант следует использовать при попытке отправить команду Legacy SPI или устройству профиля 1. Допустимые команды, которые можно построить с помощью этого типа инструкции, описаны в разделе 1.3.9.10.2.2.5.3.
Вариант 2: Legacy HyperFlash, xSPI профиля 2	Этот вариант следует использовать при отправке команды Legacy HyperFlash или устройству профиля 2. Допустимые команды, которые можно построить с помощью этого типа инструкции, описаны в разделе 1.3.9.10.2.2.5.4.
Вариант 3: Общий	Этот вариант следует использовать для формирования последовательности команд, которая не является известной инструкцией протокола.
Склеенная последовательность данных	Этот вариант нельзя использовать отдельно, его необходимо совместить с другими вариантами. Обычно он определяет, как будут передаваться данные с учетом информации из основных вариантов команд выше. Это означает, что хост не должен выполнять эту инструкцию первой

Вариант 1: команды STIG для Legacy SPI, xSPI профиля 1 или устройств SPI-NAND

В таблице 1.369 представлена структура команды STIG.

Таблица 1.369 – Вариант 1: структура команды STIG

Название	**Биты**	**Описание**
Зарезервировано	127:126	-
Включение tCMS/tCEM (TCMS_EN)	125	Установление этого бита включает функцию, которая автоматически заставляет контроллер приостановить работу запоминающего устройства, чтобы выполнить периодическое обновление. Эта функция требуется некоторым устройствам энергозависимой памяти (например, HyperRAM). Ее называют синхронизацией tCMS. После установки этого бита контроллер приостанавливает работу согласно регистру dev_active_max (0x1018).

Название	Биты	Описание
Завершение инструкции (INSTR_LINK)	124	Этот бит следует использовать, только когда в поле instruction_type инструкции задано значение =0, 32, 36 или 96. Подробнее о настройке instruction_type см. в разделе 1.3.9.10.2.2.5.3. В поле, назначенном этому биту, указано, будет ли завершена инструкция (по завершении инструкции должен быть деактивирован CS#) либо инструкция будет склеена со следующей инструкцией, если она готова к выполнению из FIFO команд. 0: Текущая последовательность завершается и не будет склеена с другой инструкцией этого же типа. 1: Текущая последовательность будет склеена со следующей инструкцией STIG.
Зарезервировано	123:119	-
Вариант CRC (CRC_VARIANT)	118	При расчете CRC команды установите 0, чтобы рассчитать CRC только для байт адреса, либо 1, чтобы рассчитать CRC для всех байты в последовательности.
Включение CRC (CRC_EN)	117	Включает динамический расчет CRC и выводит это значение в интерфейсе устройства флеш-памяти xSPI в режимах передачи с настройками ADDR_EDGE_MODE и ADDR_NO_OF_IOS. Если настроен вариант DDR (8D), рассчитанное количество байт CRC сохраняется в течение всего периода тактового сигнала флеш-памяти.
Включение XIP (XIP_EN)	116	Если установить этот бит, контроллер заблокирует отправку кода операции, что необходимо, когда устройство работает в режиме XIP (иногда его называют непрерывным режимом). 0 — фаза передачи кода операции отправлена. 1 — фаза передачи кода операции не отправлена (XIP включен).
Смещение адреса (ADDR_SHIFT)	115	Если настройки ADDR_EDGE_MODE и ADDR_NO_OF_IOS определяют вариант DDR 8D, за один период тактового сигнала устройства флеш-памяти необходимо отправить 2 байта. Поэтому хост должен установить четное количество байтов. Однако существует ряд исключений, например команда Adesto Octal DTR SFDP, которая требует выравнивание адресов флеш-памяти {A2,A1,A0,не имеет значения}, а не стандартное выравнивание {A3,A2,A1,A0}. Бит ADDR_SHIFT смещает цепочки байтов адреса на один байт влево и заполняет самый младший. Значимый байт с нулями, обеспечивающий структуру адреса DDR 8D, который требует Adesto.
Зарезервировано	114:111	-
Код банка (BANK_NUM)	110:108	Это поле сообщает контроллеру, на каком запоминающем устройстве должна быть выполнена последовательность. Значение соотвествует CS.
Режим границы кода операции (OPCODE_EDGE_MODE)	107	Этот бит выбирает для запоминающего устройства операцию SDR или DDR 0 – SDR 1 – DDR
Зарезервировано	106	-
Режим ввода/вывода кода операции (OPCODE_NO_OF_IOS)	105:104	Определяет количество активных IO для фазы кода операции инструкции. 0 – Один 1 – Два 2 – Четыре 3 – Восемь
Зарезервировано	103:101	-

Название	Биты	Описание
Включение расширения кода операции (OPCODE_EXT_EN)	100	Включает код операции с двумя байтами для инструкции. 0 — на фазе кода операции этой инструкции будет передан только CMD [87:80]. 1 — на фазе кода операции этой инструкции будет передан CMD [87:80], после которого следует CMD_EXT[79:72]. Если настройки OPCODE_EDGE_MODE и OPCODE_NO_OF_IOS определяют вариант DDR 8D, за один период тактового сигнала устройства флеш-памяти необходимо отправить 2 байта. Таким образом хост также должен установить OPCODE_EXT_EN. Если этого не сделать, возникнет ошибка.
Режим границы адреса (ADDR_EDGE_MODE)	99	0 – SDR 1 – DDR
Зарезервировано	98	-
Режим ввода/вывода адреса (ADDR_NO_OF_IOS)	97:96	Определяет количество активных IO для фазы адреса инструкции: 0 – Один 1 – Два 2 – Четыре 3 – Восемь
Зарезервировано	95	-
Количество байтов адреса (ADDR_NO_OF_BYTES)	94:92	Определяет количество байтов адреса, которые будут отправлены в инструкции. В одной последовательности может быть отправлено до 6 байтов адреса. Здесь запрещено указывать значение 7 и более. Значение 0 подразумевает, что байты адреса не будут отправлены (т. е. фаза адреса отключена). Если настройки ADDR_EDGE_MODE и ADDR_NO_OF_IOS определяют вариант DDR 8D, за один период тактового сигнала устройства флеш-памяти необходимо отправить 2 байта. Поэтому хост должен установить четное количество байтов адреса. Если хост установит ADDR_NO_OF_BYTES = 1, 3 или 5, контроллер выдаст ошибку.
Зарезервировано	91:90	-

Название	Биты	Описание
Количество байтов данных (DATA_NO_OF_BYTES) или Количество байтов режима (MODE_NO_OF_BYTES)	89:88	В случаях, когда количество байтов данных (или режима) меньше или равно 2, это поле позволяет включить данные в инструкцию напрямую, не склеивая отдельные инструкции последовательности данных. Если требуется передать более двух байтов данных, инструкцию можно склеить с последовательностью данных. Также обратите внимание, что если передать до 2 байтов данных (или режима) напрямую из этой инструкции, формат, в котором они будут доставлены на устройство, зависит от настроек ADDR_EDGE_MODE и ADDR_NO_OF_IOS. Если хосту понадобится передать эти данные в другом формате, в это поле необходимо установить значение 0 и передать данные с помощью склеенной инструкции последовательности данных или общей инструкции, где формат байтов данных может быть настроен отдельно. 0 – передается ноль байт данных/режима (фаза данных отключена) 1 – передается один байт 2 – передается два байта 3 – Н/Д Если настройки ADDR_EDGE_MODE и ADDR_NO_OF_IOS определяют вариант DDR 8D, за один период тактового сигнала устройства флеш-памяти необходимо отправить 2 байта. Поэтому хост должен установить четное количество байтов. Если этого не сделать, возникнет ошибка.
CMD	87:80	Мнемоника кода операции
CMD_EXT	79:72	Мнемоника расширения кода операции
ADDR5	71:64	Байт 5 фазы адреса
ADDR4	63:56	Байт 4 фазы адреса
ADDR3	55:48	Байт 3 фазы адреса
ADDR2	47:40	Байт 2 фазы адреса
ADDR1	39:32	Байт 1 фазы адреса
ADDR0	31:24	Байт 0 фазы адреса
DATA1/MODE1	23:16	Байт 1 фазы данных/режима
DATA0/MODE0	15:8	Байт 0 фазы данных/режима
Зарезервировано	7	-
Тип инструкции	6:0	Это поле определяет тип (индекс) инструкции. Описание каждого значения и соответствующую инструкцию можно найти в пп 1.3.9.10.2.2.5.3

Вариант 2: команды STIG для Legacy HyperFlash или xSPI профиля 2

В таблице 1.370 представлена структура команды STIG.

Таблица 1.370 – Вариант 2: структура команды STIG

Название	Биты	Описание
Зарезервировано	127:126	-

Название	Биты	Описание
Включение tCMS/tCEM (TCMS_EN)	125	Установление этого бита включает функцию, которая автоматически заставляет контроллер приостановить работу запоминающего устройства, чтобы выполнить периодическое обновление. Эта функция требуется некоторым устройствам энергозависимой памяти (например, HyperRAM). Ее называют синхронизацией tCMS. После установки этого бита контроллер приостанавливает работу согласно регистру dev_active_max (0x1018).
Завершение инструкции (INSTR_LINK)	124	Этот бит следует использовать, только когда в поле instruction_type инструкции задано значение =0, 32, 36 или 96. Подробнее о настройке instruction_type см. в разделе 1.3.9.10.2.2.5.4. В поле, назначенном этому биту, указано, будет ли завершена инструкция (по завершении инструкции должен быть деактивирован CS#) либо инструкция будет склеена со следующей инструкцией, если она готова к выполнению из FIFO команд. 0: Текущая последовательность завершается и не будет склеена с другой инструкцией этого же типа. 1: Текущая последовательность будет склеена со следующей инструкцией STIG.
Зарезервировано	123:121	-
Маска включения модификатора для расширения команды (EXT_CMD_MOD_EN)	120	Определяет вариант расширения команды xSPI профиля 2.0. 0 – маска модификатора команды отключена. Биты 44:40 из модификатора команд xSPI профиля 2.0 будут заполнены ADDR_BIT_44:ADDR_BIT_40 (биты 84:80). 1 – маска модификатора команды включена. Все биты 44:40 модификатора команд xSPI профиля 2.0 устанавливаются на 1 (маски единиц будут распределены на эти биты).
Зарезервировано	119:118	-
Включение CRC (CRC_EN)	117	Включает динамический расчет CRC и выводит это значение в интерфейсе устройства флеш-памяти xSPI.
Зарезервировано	116:111	-
Код банка (BANK_NUM)	110:108	Это поле сообщает контроллеру, на каком запоминающем устройстве должна быть выполнена последовательность. Значение соотвествует CS.
Зарезервировано	107:90	-
Количество байтов данных (DATA_NO_OF_BYTES) или Количество байтов режима (MODE_NO_OF_BYTES)	89:88	В случаях, когда количество байтов данных (или режима) меньше или равно 2, это поле позволяет включить данные в инструкцию напрямую, не склеивая отдельные инструкции последовательности данных. Если требуется передать более двух байтов данных, инструкцию можно склеить с последовательностью данных. 0 – передается ноль байтов данных/режима (фаза данных отключена), 1 – Н/Д – ошибка конфигурации 2 – передается два байта 3 – Н/Д – ошибка конфигурации
R/W	87	Соответствует 47-му биту команды/адреса (CA) для устройств Legacy HyperFlash и xSPI профиля 2.0 0 – операция записи 1 – операция чтения

Название	Биты	Описание
TARGET	86	Соответствует 46-му биту команды/адреса (CA) для устройств Legacy HyperFlash и xSPI профиля 2.0 (называется "адресным пространством" (AS)). 0 – пространство памяти 1 – пространство регистра В Legacy HyperFlash эта функция не используется, поэтому этот бит следует установить в 0. В будущем устройства xSPI смогут поддерживать функционал "TARGET", поэтому контроллер позволит пользователю указать здесь значение.
BURST_TYPE	85	Соответствует 45-му биту команды/адреса (CA) для устройств Legacy HyperFlash и xSPI профиля 2.0 Контроллер поддерживает только INCR Bursts, поэтому для этого бита необходимо установить значение 1
ADDR_44_40	84:80	Соответствует битам 44:40 адресного пространства флеш-памяти. Контроллер автоматически преобразует адресное пространство флеш-памяти от хоста в распределение битов команды/адреса (CA) на основе зарезервированной ширины адреса, указанной в регистре hf_offset (0x1020). Старшие биты адреса отбрасываются после того, как учтена позиция зарезервированных битов адреса. Количество отброшенных битов совпадает с количеством битов в позиции.
Зарезервировано	79:64	-
ADDR_39_0	63:24	Соответствует битам 39:0 адресного пространства флеш-памяти. Контроллер автоматически преобразует адресное пространство флеш-памяти от хоста в распределение битов команды/адреса (CA) на основе зарезервированной ширины адреса, указанной в регистре hf_offset (0x1020). Старшие биты адреса отбрасываются после того, как учтена позиция зарезервированных битов адреса. Количество отброшенных битов совпадает с количеством битов в позиции.
DATA1/MODE1	23:16	Байт 1 фазы данных/режима.
DATA0/MODE0	15:8	Байт 0 фазы данных/режима.
Зарезервировано	7	-
Тип инструкции	6:0	Это поле определяет тип (индекс) инструкции. Описание каждого значения и соответствующую инструкцию можно найти в пп 1.3.9.10.2.2.5.4

Вариант 3 команды STIG

В таблице 1.371 представлена структура команды STIG.

Таблица 1.371 – Вариант 3: структура команды STIG

Название	Биты	Описание
Зарезервировано	127:125	-

Название	Биты	Описание
Завершение инструкции (INSTR_LINK)	124	Этот бит следует использовать, только когда в поле instruction_type инструкции задано значение =0, 32, 36 или 96. Подробнее о настройке instruction_type см. в 1.3.9.10.2.2.5.5 В поле, назначенном этому биту, указано, будет ли завершена инструкция (по завершении инструкции должен быть деактивирован CS#) либо инструкция будет склеена со следующей инструкцией, если она готова к выполнению из FIFO команд. 0: Текущая последовательность завершается и не будет склеена с другой инструкцией этого же типа. 1: Текущая последовательность будет склеена со следующей инструкцией STIG.
Зарезервировано	123:118	-
Включение CRC (CRC_EN)	117	Включает динамический расчет CRC и выводит это значение в интерфейсе xSPI в режимах передачи с настройками ADDR_EDGE_MODE и ADDR_NO_OF_IOS. Если настроен вариант DDR 8D, рассчитанное количество байт CRC сохраняется в течение всего периода тактового сигнала флеш-памяти.
Зарезервировано	116:111	-
Код банка (BANK_NUM)	110:108	Это поле сообщает контроллеру, на каком запоминающем устройстве должна быть выполнена последовательность. Значение соотвествует CS.
Режим границы (EDGE_MODE)	107	0 – SDR 1 – DDR
Зарезервировано	106	-
Режим ввода/вывода (NO_OF_IOS)	105:104	Определяет количество активных IO для инструкции: 0 – Один 1 – Два 2 – Четыре 3 – Восемь
Зарезервировано	103:92	-
Количество байтов (NO_OF_BYTES)	91:88	Определяет количество байтов, которые будут отправлены в инструкции. Необходимо отправить хотя бы один байт, но не более 10 байтов в одной последовательности. Если настройки EDGE_MODE и NO_OF_IOS определяют вариант DDR 8D, за один период тактового сигнала устройства флеш-памяти необходимо отправить 2 байта. Поэтому хост должен установить четное количество байтов (допустимые значения NO_OF_BYTES = 0, 2, 4, 6, 8 или 10). Если хост установит NO_OF_BYTES = 1, 3, 5, 7 или 9, контроллер не примет запрос на передачу и выдаст ошибку.
BYTE9	87:80	10-й байт на отправку, если NO_OF_BYTES = 10
BYTE8	79:72	9-й байт на отправку, если NO_OF_BYTES >=9
BYTE7	71:64	8-й байт на отправку, если NO_OF_BYTES >=8
BYTE6	63:56	7-й байт на отправку, если NO_OF_BYTES >=7
BYTE5	55:48	6-й байт на отправку, если NO_OF_BYTES >=6
BYTE4	47:40	5-й байт на отправку, если NO_OF_BYTES >=5
BYTE3	39:32	4-й байт на отправку, если NO_OF_BYTES >=4
BYTE2	31:24	3-й байт на отправку, если NO_OF_BYTES >=3
BYTE1	23:16	2-й байт на отправку, если NO_OF_BYTES >=2

Название	Биты	Описание
BYTE0	15:8	1-й байт на отправку
Зарезервировано	7	-
Тип инструкции	6:0	Это поле определяет тип (индекс) инструкции. Описание каждого значения и соответствующую инструкцию можно найти в пп 1.3.9.10.2.2.5.5

Инструкция для склеенной последовательности данных STIG

В таблице 1.372 описана структура основного варианта инструкции данных, которую должен выполнить хост в качестве последней склеенной инструкции последовательности инструкций. Этот вариант нельзя использовать отдельно, его необходимо совместить с другими вариантами. Обычно он определяет, как будут передаваться данные с учетом информации из трех основных вариантов команд выше. Это означает, что хост не должен выполнять эту инструкцию в первую очередь — этот вариант всегда должен быть последним.

Таблица 1.372 – Инструкция для склеенной последовательности данных STIG

Название	Биты	Описание
Зарезервировано	127:123	-
Размер блока данных CRC (CRC_CHUNK_SIZE)	122:120	Представляет размер блока данных CRC, настроенного в запоминающем устройстве. Это количество байтов, после которого в шине размещается байт CRC. 1 Н/Д 2 Размер блока данных 8 Б 3 Размер блока данных 16 Б 4 Размер блока данных 32 Б 5 Размер блока данных 64 Б 6 Размер блока данных 128 Б 7 Размер блока данных 256 Б 8 Размер блока данных 512 Б
Включение инверсии состояния чтения CRC (CRC_INV)	119	Этот бит контролирует, запросит ли контроллер еще часть информации о состоянии, чтобы проверить целостность данных. Предполагается, что будет инвертирован избыточный блок данных (актуально для устройств Macronix MX25 Octa). Если избыточные данные из флеш-памяти не будут инвертированы, будет создана ошибка CRC. Этот бит должен быть установлен, только если CRC_EN = 1, DATA_EDGE_MODE = 1 (DDR), DATA_NO_OF_IOS = 3 (OCTAL) и DATA_NO_OF_BYTES = 2, ведь этот набор параметров соответствует реализации устройств Macronix MX25 Octa. Если CRC_INV = 1, настройка CRC_CHUNK_SIZE игнорируется, поскольку эти два параметра последовательности исключают друг друга.
Включить выход CRC (CRC_OE)	118	Этот бит указывает, будет ли CDC выведен на обоих фронтах тактового сигнала с инверсией CRC на 2-м фронте тактового сигнала. 0 – инвертированный CRC не выводится 1 – инвертированный CRC выводится

Название	Биты	Описание
Включение CRC (CRC_EN)	117	Включает расчет CRC и выводит это значение в интерфейсе устройства флеш-памяти xSPI в режимах передачи, указанных настройками ADDR_EDGE_MODE и ADDR_NO_OF_IOS. Если настроен вариант DDR (8D), рассчитанное количество байт CRC сохраняется в течение всего периода тактового сигнала флеш-памяти.
Включение невыровненного блока данных CRC (CRC_UAL_CHUNK_EN)	116	Если установить этот бит, контроллер будет учитывать адрес команды, чтобы определить, куда будут возвращены байты CRC из запоминающего устройства. Этот бит можно установить в 1, только если эта последовательность (DATA_SEQ) склеена с последовательностью READ_PROFILE_1. В противном случае необходимо установить в 0. Например, если оригинальный адрес FLASH — 0x4, а CRC_CHUNK_SIZE = 16 Б, первый CRC от устройства будет получен через 12 байтов (CRC_CHUNK_SIZE – (адрес устройства)). Поддерживаются даже четные адреса (когда CRC_UAL_CHUNK_EN = 1), как того требуют устройства Macronix MX25 Octa.
Проверка невыровненного блока данных CRC (CRC_UAL_CHUNK_CHK)	115	Если установить этот бит, контроллер проверит корректность невыровненного блока данных CRC с устройства флеш-памяти. Можно установить в 1, только если CRC_UAL_CHUNK_EN = 1. В противном случае необходимо установить в 0. Поскольку устройства Macronix MX25 Octa не гарантируют корректность расчета CRC для невыровненного блока данных, контроллер позволяет контролировать эту проверку. Если настроенный параметр DATA_NO_OF_BYTES позволит вместить много блоков данных CRC, корректность CRC будет проверяться по всем фрагментам CRC, выровненным с блоком данных, вне зависимости от значения CRC_UAL_CHUNK_CHK.
Зарезервировано	114:113	-.
Смена данных	112	Когда этот бит установлен, меняется порядок байтов, передаваемых в одном периоде тактового сигнала DDR 8D. Если DATA_EDGE_MODE и DATA_NO_OF_IOS не отражают режим DDR 8D, этот бит следует установить в 0.
Зарезервировано	111	-.
Код банка (BANK_NUM)	110:108	Это поле сообщает контроллеру, на каком запоминающем устройстве должна быть выполнена последовательность. Значение соотвествует CS.
Режим границы данных (DATA_EDGE_MODE)	107	0 – SDR 1 – DDR
Зарезервировано	106	-
Режим ввода/вывода данных (DATA_NO_OF_IOS)	105:104	Определяет количество активных IO для всех байтов данных инструкции. Байты данных, связанные с инструкцией, берутся из внутреннего FIFO данных (для операций записи) и размещаются во внутреннем FIFO данных (для операций чтений). 0 – Один 1 – Два 2 – Четыре 3 – Восемь
Зарезервировано	103:101	-
Направление передачи (DIR)	100	Определяет направление потока данных 0 – чтение из запоминающего устройства 1 – запись на запоминающее устройство

Название	Биты	Описание
Зарезервировано	99:93	-.
Размер данных в расположении адреса флеш-памяти (DATA_PER_ADDR)	92	Указывает, сколько байтов данных назначено одному адресу флеш-памяти. Этот параметр позволяет контроллеру вручную рассчитывать последовательные адреса всех байтов данных в последовательности. Шаг увеличения адреса всегда равен 1. 0 – адрес запоминающего устройства состоит из одного байта 1 – адрес запоминающего устройства состоит из двух байтов
Зарезервировано	91:90	-
Количество пустых тактов (задержка) (NO_OF_DUMMY)	89:84	Определяет количество пустых тактов перед фазой данных. Обратите внимание, что определение тактов задержки для Legacy HyperFlash и xSPI профиля 2.0 пересекается с фазой передачи адреса на один период тактового сигнала, поэтому количество требуемых тактов задержки будет на один меньше запрограммированного значения. Если задержка внешней платы приведет к дополнительной задержке длительностью еще в несколько периодов тактового сигнала (для команд чтения), модуль PHY выполнит компенсацию.
Включение чтения через границы в HyperFlash (HF_READ_BOUND_EN)	83	С помощью этого параметра контроллер рассчитывает операцию чтения, пересекающую границы страницы, где данные недействительны (на основе количества пустых тактов (NO_OF_DUMMY) и начального адреса инструкции). Во время периода задержки устройство не выдает DQS.
Зарезервировано	82:80	-.
Количество байтов данных (DATA_NO_OF_BYTES)	79:48	Определяет количество байтов данных, которые будут отправлены в инструкции. Все байты данных берутся из FIFO данных контроллера (для операций записи) и размещаются в FIFO данных (для операций чтения). Если настройки DATA_EDGE_MODE и DATA_NO_OF_IOS определяют вариант DDR 8D, за один период тактового сигнала устройства флеш-памяти необходимо отправить 2 байта. Поэтому хост должен установить четное количество байтов данных. Если этого не сделать, контроллер выдаст ошибку.
Зарезервировано	47:25	-.
Источник данных состояния (STATUS_SOURCE)	24	Вступает в силу, только если DIR = 0 (операция чтения) и доступно не более 2 байтов для чтения. Если для чтения доступно более 2 байтов, этот бит имеет нулевое значение. Этот бит определяет, как хост в конечном итоге получает данные. Если для чтения доступно не более 2 байтов, хост может передать эти 2 байта в регистре состояния STIG (0x44). В противном случае необходимо собрать данные для чтения с помощью AHB-salve, как описано в разделе "Как использовать режим STIG". 0 – Данные для чтения отправляются на AHB-salve. 1 – Данные для чтения отправляются в регистр состояния STIG (0x44).
Зарезервировано	23:7	-
Тип инструкции	6:0	Для инструкции склеенных данных это значение всегда зафиксировано на 127.

1.3.9.10.2.2.5.3 Стандартные операции устройства, выполняемые с помощью варианта 1

В таблице 1.373 представлены стандартные операции, которые может выполнять хост на устройстве с помощью 1-го варианта инструкции, а также значение, которое хост должен

ввести в поле типа инструкции. Также в таблице указано, поддерживает ли инструкция склеивание. Обратите внимание, что если инструкция данных склеена с одной из этих инструкций, в поле типа для инструкции данных следует указать 128.

Таблица 1.373 – Операции варианта 1

Алгоритм	**Тип инструкции**	**Склеивание инструкции**
READ	1	Да – склеивание с инструкцией данных
READ IN XIP	1	Да – склеивание с инструкцией данных
WRITE	0	Да, если длина данных > 2 байта
SET WEL	0	Нет
CLEAR WEL	0	Нет
ERASE SECTOR	0	Нет
CHIP ERASE	0	Нет
PROGRAM SUSPEND	0	Нет
PROGRAM RESUME	0	Нет
READ DEVICE REGISTER	1	Нет
WRITE DEVICE REGISTER	0	Нет
SETUP READ BURST	0	Нет
CLEAR FLAG REGISTER	0	Нет
ENTER DEEP POWERDOWN	0	Нет
EXIT DEEP POWERDOWN	0	Нет
RESET ENABLE	0	Нет
SOFT RESET	2	Нет
READ JEDEC ID	1	Нет
READ SFDP	1	Да – склеивание с инструкцией данных
ENTER DEFAULT PROTOCOL MODE	0	Нет
ENTER SDR OCTAL	0	Нет
GLOBAL PROTECTION	0	Нет
GLOBAL UNPROTECTION	0	Нет
PROTECT SECTOR	0	Нет
UNPROTECT SECTOR	0	Нет
ENTER 4-BYTE ADDRESS MODE	0	Нет
EXIT 4-BYTE ADDRESS MODE	0	Нет
SPI_NAND PAGE READ	1	Да – склеивание с инструкцией данных
SPI_NAND READ FROM CACHE	0	Да – склеивание с инструкцией данных
SPI_NAND READ PAGE CACHE RAN- DOM	0	Нет

Алгоритм	Тип инструкции	Склеивание инструкции
SPI_NAND READ PAGE CACHE LAST	0	Нет
SPI_NAND PROGRAM LOAD	0	Да, если длина данных > 2 байта
SPI_NAND PROGRAM LOAD RANDOM	0	Да, если длина данных > 2 байта
SPI_NAND PROGRAM EXECUTE	0	Да, если длина данных > 2 байта
SPI_NAND SET FEATURE	0	Да, если длина данных > 2 байта
SPI_NAND GET FEATURE	1	Да – склеивание с инструкцией данных
SPI_NAND READ ID	32	Да – склеивание с инструкцией данных или варианта

1.3.9.10.2.2.5.4 Стандартные операции устройства, выполняемые с помощью варианта 2

В таблице 1.374 представлены стандартные операции, которые может выполнять хост на устройстве с помощью 2-го варианта, а также значение, которое хост должен ввести в поле типа инструкции. В таблице приведены поля инструкций, где для выполнения операции необходимо задать фиксированное значение, а также указано, поддерживает ли инструкция склеивание. Обратите внимание, что если инструкция данных склеена с одной из этих инструкций, в поле типа для инструкции данных следует указать 128.

Таблица 1.374 – Операции варианта 2

Алгоритм	Тип инструкции	Склеивание инструкции
WRITE/READ для устройств HyperRAM	32	Да – склеивание с инструкцией данных или варианта
WREN 1	33	Нет
WREN 2	34	Нет
Profile 2 READ	35	Да – склеивание с инструкцией данных
Profile 2 PROGRAM	36	Да, если длина данных > 2 байта Можно склеивать с данными или другим вариантом
ERASE SECTOR	37	Нет
CHIP ERASE	38	Нет
ERASE SUSPEND	39	Нет
ERASE RESUME	40	Нет
PROGRAM SUSPEND	41	Нет
PROGRAM RESUME	42	Нет
READ STATUS	43	Да – склеивание с инструкцией данных
READ VOLATILE MEMORY REGISTER	44	Да – склеивание с инструкцией данных

Алгоритм	Тип инструкции	Склеивание инструкции
READ NON_VOLATILE MEMORY REGISTER	45	Да – склеивание с инструкцией данных
READ CUSTOM MEMORY REGISTER	47	Да – склеивание с инструкцией данных
STATUS REGISTER CLEAR	48	Нет
WRITE VOLATILE MEMORY REGISTER	49	Нет
WRITE NON_VOLATILE MEMORY REGISTER	50	Нет
WRITE CUSTOM MEMORY REGISTER ENABLE	51	Да – дополняется другой инструкцией варианта 2 с типом = 32
ENTER DEEP POWER DOWN	52	Нет
RESET/ASO EXIT	53	Нет
ENTER DEFAULT PROTOCOL MODE	54	Нет
READ SFDP	55	Да – склеивание с инструкцией данных

1.3.9.10.2.2.5.5 Стандартные операции устройства, выполняемые с помощью варианта 3

Этот вариант следует использовать для формирования общей или пользовательской последовательности команд, которая не является известной инструкцией протокола. Для этого в поле типа инструкции укажите 96. Эта инструкция может быть склеена с другой инструкцией варианта 3 или с инструкцией последовательности данных.

1.3.9.10.2.2.5.6 Переключение между режимами работы

Рабочий режим STIG предлагает интерфейс для выполнения низкоуровневых операций в интерфейсе устройства флеш-памяти xSPI. Его следует использовать вместе с режимом DIRECT, поэтому вам может понадобиться часто входить в режим STIG и покидать его. Задача по включению и выключения режима STIG, когда целевое устройство флеш-памяти xSPI находится в режиме простоя, лежит на ПО хоста.

Перед подключением в режим STIG хост должен дождаться завершения всех операций, адресованных на то же устройство флеш-памяти xSPI. Прежде чем покидать режим STIG, хост должен убедиться, что устройство находится в состоянии простоя. Если после выполнения команды устройство xSPI перешло в состояние занятости, может быть недостаточно дождаться завершения последней команды режима STIG. В таких случаях необходимо опросить регистр состояния устройства — для этого хост должен выдать инструкции STIG, чтобы выполнить команды чтения состояния, и делать это до тех пор, пока запоминающее устройство не перейдет в состояние простоя. Когда устройство окажется в состоянии простоя, хост сможет покинуть режим STIG.

1.3.9.10.2.2.5.7 Как использовать режим STIG

Чтобы выполнить операцию с помощью STIG, хост может выполнить следующие шаги:

1) Дождаться, когда контроллер окажется в состоянии IDLE. Чтобы убедиться в этом, необходимо опросить **ctrl_busy** регистра ctrl_status (**0x0100**.

2) Чтобы включить режим STIG, установите значение 2'b01 в поле **work_mode** в регистре **ctrl_config (0x0230)**.

3) Установите регистры команд с cmd_reg1 по cmd_reg4.

4) Если завершение запрошенной операции STIG должно вызвать внешнее прерывание и/или **sdma_trigg** для операций через AHB-salve, включите прерывание **stig_done**. Чтобы включить эти прерывания, установите биты **stig_done_en**, **sdma_trigg_en** и **intr_en** в регистре **intr_enable (0x0114)**.

5) Выполните запись в регистре **cmd_reg0 (0x0000)**, чтобы запустить выполнение инструкции STIG. Бит **gcmd_eng_mc_busy** в регистре **ctrl_status (0x0100)** устанавливается в 1.

6) С учетом значения в поле **тип инструкции** хост должен выбрать одно из следующего:

a) для склеивания с последовательностью данных — перейдите на шаг 7.

b) для других инструкций — если инструкция не содержала условие склеивания, то перейдите к шагу 9, в противном случае — к шагу 3.

7) Если инструкция, содержащая команду последовательности данных, указывала направление чтения и количество данных, которые следует прочитать, состояла из одного или двух битов, а поле **STATUS_SOURCE** в инструкции было установлено в 1, то перейдите к шагу 9. Во всех остальных случаях, когда для передачи данных использовался AHB-salve, перейдите к шагу 8.

8) На этом шаге описано, как хост должен передавать данные через AHB-salve, чтобы выполнить операцию STIG. На этом шаге бит **gcmd_eng_busy** в регистре **ctrl_status (0x0100)** устанавливается в 1.

a) Опросите бит **sdma_trig** в регистре состояния прерывания (0x110) либо дождитесь прерывания, если прерывание было sdma_trigg размаскировано. Когда будет готово, переходите в шагу б.

b) Прочитайте регистры sdma_size(0x240) и sdma_trd_info(0x244), чтобы узнать количество байтов для передачи, идентификатор цепочки и направление передачи.

c) Очистите флаг sdma_trigg в регистре intr_status (0x0110).

d) Теперь сторонний хост может передавать данные. Для операций программирования сторонний хост должен выполнить один или несколько запросов на запись AHB, общий размер которых совпадает с передачей данных на шаге б. Для операций чтения сторонний хост должен выполнить один или несколько запросов на чтение AHB, общий размер которых совпадает с передачей данных на шаге б. Адрес, выданный сторонним хостом, должен совпадать с адресом, указанным в регистрах sdma_addr* в позициях 0x24c и 0x250.

e) Если хост проигнорирует какое-либо из описанных выше требований и начнет передачу данных, когда ведомый DMA не готов, то в регистре intr_status (0x0110) будет установлен флаг sdma_err. Когда в регистре intr_enable (0x0114) устанавливается флаг sdma_err_en, также запускается прерывание. Когда устанавливается бит sdma_err_rsp в dma_settings (0x023c), выдается сообщение ERROR, после устранения которого выдается ответ OK.

f) Если хост отправит AHB-slave неподдерживаемую операцию, ведомый DMA проигнорирует этот доступ, а в регистре intr_status (0x0110) будет установлен флаг sdma_err. Если во время передачи через ведомый DMA в системной шине обнаруживается отклик об ошибке AHB, в последнем состоянии операции устанавливаются биты Fail и bus_error. Состояние последней операции можно проверить в регистре cmd_status (0x0044).

g) Выполнение операций в неверном направлении (т. е. при отправке команды операции чтения XSPI хост выполняет операцию записи в ведомый порт DMA) запрещено и приводит к установлению флага **sdma_err** в регистре **intr_status (0x0110)**. Когда в регистре **intr_enable (0x0114)** устанавливается флаг **sdma_err_en**, запускается прерывание. Когда устанавливается бит **sdma_err_rsp** в **dma_settings (0x023c)**, выдается сообщение ERROR, после устранения которого выдается ответ OK.

9) Дождитесь завершения операции STIG. Чтобы убедиться в этом, опросите бит **COMPLETE** в регистре **cmd_status (0x0044)** или посмотрите внешнее прерывание от **stig_done** (если правильно включено хостом). На этом шаге бит **gcmd_eng_busy** в регистре **ctrl_status (0x0100)** устанавливается в 1.

10) Прочитайте регистр **cmd_status (0x0044)**, чтобы узнать состояние STIG. На этом шаге оба флага занятости команд устанавливаются в 0.

11) Если требуется выполнить другую операцию STIG, перейдите к шагу 3. В противном случае, если необходимо изменить режим работы, сначала опросите бит **ctrl_busy** в регистре **ctrl_status (0x0100)**, чтобы дождаться, когда контроллер перейдет в состояние простоя.

1.3.9.10.2.2.5.8 Проверка состояния завершения STIG

Чтобы проверить состояние последней операции, хосту необходимо прочитать регистр cmd_status (0x0044).

Когда будет запущена новая операция, бит **COMPLETE** будет очищен, после чего все остальные биты в этом регистры теряют свое значение. По завершении операции устанавливается бит **COMPLETE**, а остальные биты в регистре состояния (см. таблицу 1.375) приобретают значение.

Таблица 1.375 – Регистр состояния

Биты	Название	Описание
31:16	DATA_FROM_DEV	В этом поле хранится один или два байта, считанных с запоминающего устройства. Поля инструкции DATA_SEQ необходимо заполнить следующим образом: DATA_NO_OF_BYTES = 1 — один байт или 2 — два байта, STATUS_SOURCE = 1. В других случаях данные с устройства флеш-памяти доступны в интерфейсе данных ведомого устройства.
15	COMPLETE	Установление этого бита означает, что контроллер обновил информацию о состоянии, а операция завершена. Этот бит устанавливается, даже когда операция завершилась неудачей.
14	FAIL	Установление этого бита означает, что операция завершена неудачно.
13:4	Зарезервировано	Зарезервировано
3	DQS_ERROR	Во время операции чтения данных было обнаружено некорректное количество DQS. Это информация от PHY о том, что либо не появился строб DQS во время чтения (например, устройство не подключено к контроллеру), либо сигнал rd_del_sel имеет некорректное значение (поле в phy_gate_lpbk_ctrl_reg), считанные данные с устройства флеш-памяти искажены, а указатели FIFO распределены неверно. dll_rst_n или rst_n очищают флаги dqs_underrun и dqs_overflow в PHY и очищают все указатели чтения данных PHY. PHY требует выполнения одного из этих условий, прежде чем возобновить работу после утверждения DQS_ERROR.
2	CRC_ERROR	Этот бит сообщает об ошибке CRC.
1	BUS_ERROR	Установление этого бита означает, что контроллер получил ошибку в AHB.
0	CMD_ERROR	Этот бит означает, что обнаружена недействительная последовательность команд.

1.3.9.10.2.2.5.9 Ограничения STIG

Режим STIG не поддерживает функцию HOLD# (альтернативная функция контакта DQ3 в некоторых Legacy SPI). Поскольку режим STIG полностью контролируется программным обеспечением, функция HOLD# теряет свою ценность, так как передачей данных в режиме STIG может управлять ПО более высокого уровня.

Контроллер xSPI не поддерживает команды передачи данных из флеш-памяти WRAP (все данные на устройство/с устройства считаются линейными).

При использовании инструкции STIG общего типа может быть прочитано не более 64 байт. Таким же образом, не более 64 байт может быть прочитано для всех инструкций на "Legacy HyperFlash и xSPI профиля 2.0", кроме READ_PROFILE_2 и READ_SFDP_P2.

Если используется режим XIP, операция чтения, использующаяся для входа в режим XIP или выхода из него, не должна пытаться передать более 64 байт.

1.3.9.10.2.2.5.10 Выстраивание последовательности операций

Инструкции STIG хранятся в основной FIFO данных. С учетом типа инструкции склеенные инструкции отправляются и проходят буферизацию на втором уровне FIFO внутренней кэш-памяти. Кэш-память FIFO вмещает до 16 склеенных последовательностей (если запрашивается больше склеенных последовательностей, создается ошибка некорректной последовательности). Затем эти инструкции обрабатываются вместе, когда поле INSTR_LINK = 0 для последней инструкции в цепочке.

Контроллер переводит инструкции и разбивает их на правильную последовательность, которую требует запоминающее устройство на основе типа команды. На схеме 1.119 показана общая структура xSPI-фрейма.

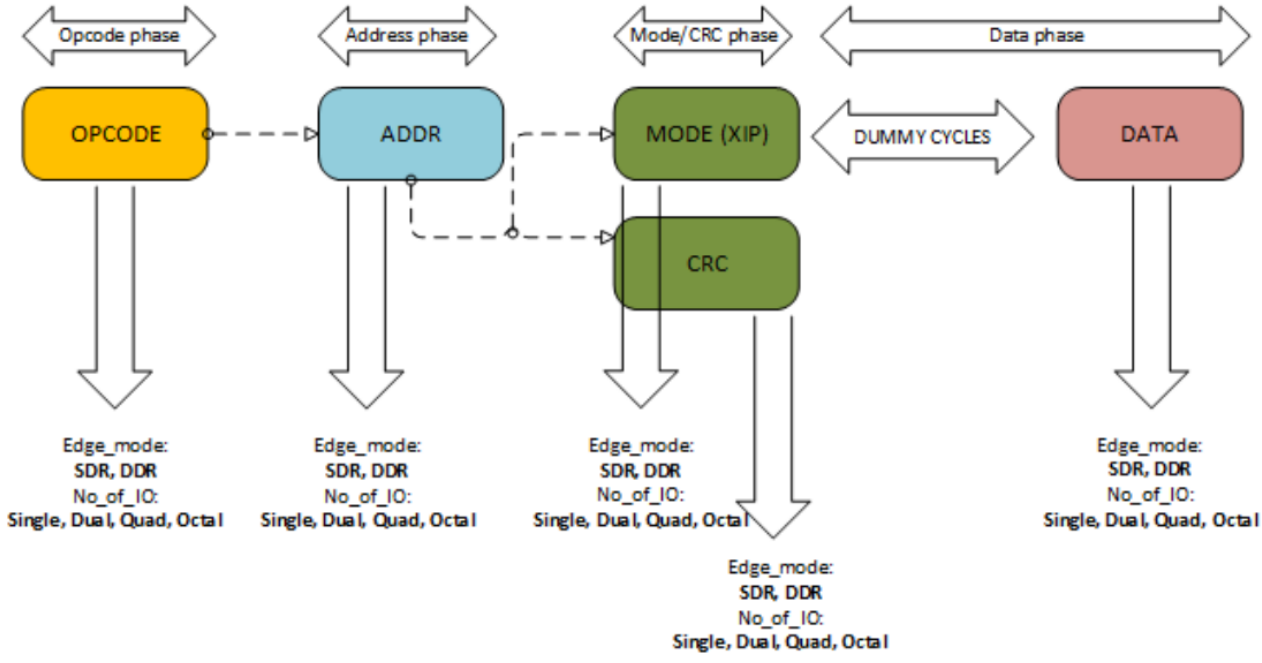


Рисунок 1.119 – Общая структура xSPI последовательности

Одноцветные блоки соответствуют фазам передачи xSPI. В зависимости от структуры команды (см. JESD216 или спецификации конкретного устройства xSPI) некоторые фазы могут быть отброшены, однако порядок всегда сохраняется. Edge_Mode и No_of_IO также могут различаться между фазами. Согласно JESD216, допускается только пошаговый рост количества используемых IO или количества фронтов тактового сигнала. Это означает, к примеру, что текущие устройства поддерживают вариант передачи (opcode=1S,addr=8D,data=8D), но не поддерживают вариант(opcode=1S,addr=8D,data=4S). JESD216 не допускает более одной передачи, где количество активных линий ввода/вывода отличается друг от друга. Также обратите внимание, что если любая из фаз настроена как 8D, а связанное значение NO_OF_BYTES соответствует нечетному значению, то контроллер выдаст ошибку и не примет инструкцию. Вариант 8D всегда требует четное количество байт. Благодаря гибкой структуре последовательности команд контроллер может настроить Edge_Mode и No_of_IO, что может стать надежным решением для новых вариантов в будущих версиях JESD216.

1.3.9.10.2.2.6 Обработка ошибок/событий в режимах DIRECT и STIG

Обработка ошибок/событий в режимах DIRECT и STIG

В зависимости от выбранного режима работы контроллер сообщает об ошибках немного по-разному:

– В режиме STIG контроллер сообщает о состоянии посредством регистра cmd_status (0x44).

– В режиме DIRECT контроллер сообщает о состоянии посредством битов dir_* в регистра intr_status (0x110).

В таблице 1.376 описаны ошибки, которые могут происходить.

Таблица 1.376 – Ошибки

Название	**Описание**
Ошибка DQS	Этот тип ошибки возникает, когда подключенный PHY обнаруживает ситуацию dqs_underrun или dqs_overflow. Ошибка DQS может привести к искажению потока данных, передаваемого из флеш-памяти xSPI. Хост должен утвердить dll_rst_n или rst_n, чтобы очистить флаги dqs_underrun и dqs_overflow в PHY и очистить все указатели чтения данных PHY.

Название	Описание
Ошибка системной шины	Выходы состояния AHB HRESP никогда не устанавливаются при работе в режиме DIRECT. В режиме STIG, если передача идет по AHB-slave, сообщение об ошибке возникнет в следующих случаях: 1 Хост пытается получить доступ к AHB-slave раньше, чем получит разрешение (см. соответствующие шаги в разделе о STIG). 2 Хост выдает Bursts неподдерживаемого типа. Сейчас контроллер поддерживает инкрементные Bursts (без WRAP)
Ошибка CMD	Режим STIG: Может возникать, если не следовать последовательности STIG согласно настоящему документу. Чтобы исправить ошибку, внимательно перепроверьте последовательность и повторите попытку. Режим DIRECT: В этом режиме запускается прерывание **dir_cmd_err**, когда хост выдает операцию AHB, которую контроллер не может выполнить по одной из следующих причин: – Расширение команды не включено, при режиме OCTAL DDR. – Количество байт адреса не является четным, при режиме OCTAL DDR. – Значение адреса не является четным, при режиме OCTAL DDR. – Количество байт данных не является четным, при режиме OCTAL DDR, кроме случаев, когда включено маскирование байт RWDS посредством прямого доступа к регистру cfg (доступно только для режима DIRECT). – Количество байт для последовательности проверки состояния не является четным, при режиме OCTAL DDR. – Устройство и контроллер настроены для работы в режиме XIP, однако со стороны хоста был осуществлен доступ для записи. – Устройство и контроллер настроены для работы в режиме XIP, однако отправка байта режима (включить/выключить) отключена. – Устанавливается параметр **seq_data_per_addr**, адрес или размер передачи имеют не четное значение, кроме случаев, когда включено маскирование байт RWDS посредством прямого доступа к регистру cfg (доступно только для режима DIRECT). – Размер слова (HSIZE) превышает размер выбранной страницы в устройстве профиля 1 или 2 — HF.
Ошибка CRC	Когда включена проверка CRC, эта ошибка возникает, когда контроллер обнаруживает ошибку CRC.
Ошибка устройства	Ошибка устройства возникает в случае сбоя подключенного запоминающего устройства при проверке состояния последней операции стирания/программирования/чтения. Она возможна только в режиме DIRECT. Состояние операции чтения можно проверить только на устройствах SPI NAND. Если в таком случае будет обнаружена ошибка, это значит, что устройство столкнулось с неисправимой ошибкой ECC.

Название	Описание
Ошибка ECC	Исправимая ошибка ECC возникает, когда во время операции чтения подключенное запоминающее устройство выдает ошибку, которую можно исправить, а в регистре **stat_seq_cfg_10 (0x0478)** установлено поле **stat_seq_ecc_fail_en**.

1.3.9.10.2.2.7 Использование программного PHY с контроллером

PHY отвечает за обработку низкоуровневой синхронизации сигналов данных, адреса и управления между устройством и контроллером.

PHY обладает собственными программируемыми регистрами. Доступ к ним можно получить через стандартное адресное пространство контроллера через APB, начиная со смещения адреса 0x2000.

Примечание. PHY разработан для нескольких протоколов, поэтом некоторые регистры могут не использоваться с xSPI.

1.3.9.10.2.2.7.1 Инициализация PHY

Прежде чем хост сможет обмениваться данными с запоминающим устройством, необходимо инициализировать и заблокировать PHY. Программный PHY содержит три синтезируемых линии задержки (DLL), которые состоят из 256 элементов задержки. Первый DLL - ведущий DLL. Его главная цель — измерить, сколько элементов задержки требуется для соответствия периоду тактового сигнала xspi_clk. Другие два DLL являются ведомыми DLL, один — для выходного пути передачи, а другой — для входного пути приема. С помощью измерений ведущего DDL ведомый DLL передачи может повлиять на то, какая доля периода тактового сигнала будет добавлена к передаче данных в качестве задержки (для улучшения глазковой диаграммы данных в запоминающем устройстве). Аналогичным образом, ведомый DLL приема позволяет сместить синхронизацию выборки для чтения в виде доли от тактового периода, чтобы найти оптимальную точку выборки для входящих данных. Однако перед этим необходимо настроить ведущий DLL, что делается сразу после сброса при подаче питания (PoR). Это называется блокировкой PHY. Блокировка PHY является внутренней функцией PHY, которая помогает заблокировать ведущий DLL. Пока PHY не заблокирован, он не может обмениваться данными с запоминающим устройством. Если вы решите отключить ведущий DLL (также называется режим насыщения), ведомый DLL нельзя будет запрограммировать с учетом доли от тактового периода, поэтому его придется настраивать вручную. Этот режим подходит для работы на низкой скорости передачи данных, когда у ведущего DLL недостаточно элементов задержки для измерения полного периода синхронизации. Для работы на низкой скорости не

требуется точная настройка ведомого DLL (особенно при выполнении SDR), поэтому допускается настройка вручную.

Примечание. Блокировку PHY не следует путать с обучением PHY. Обучение PHY — это программная процедура, которая может быть реализована для поиска идеальной точки стробиования и, следовательно, для обеспечения максимально возможной частоты xspi_clk. Контроллер сам не реализует обучение PHY.

Чтобы заблокировать ведущий DLL и подготовить PHY к передаче данных, для некоторых регистров PHY, начиная с 0x2000, необходимо запрограммировать подходящие значения на основе частоты xspi_clk и системных параметров, например, отклонений в плате, задержкек контактных площадок и т. п. Если не запрограммировать подходящие значения, PHY не будет заблокирован, поэтому не сможет правильно запуститься. Следовательно, это важный шаг.

Перед попыткой блокировки необходимо настроить 6 регистров PHY. Ниже собраны главные аспекты.

Регистр dll_phy_ctrl (0x1034)

Обычно регистр используется, чтобы контролировать обновление ведомых DLL на уровне PHY для компенсации сдвига, вызванного динамическими факторами, например температурой или OCV. Также в нем содержатся настройки, зависящие от подключенного запоминающего устройства и режима работы.

sdr_edge_active в бите 21 — это бит конфигурации, указывающий, какой фронт контроллера будет передавать данные на запись. Он требуется только для работы SDR. По умолчанию контроллер распространяет данные на запись по нарастающему фронту тактового сигнала. Его следует оставить в состоянии по умолчанию "0" кроме случаев, когда запоминающее устройство (в режиме SDR) не может сделать выборку данных.

dqs_last_data_drop в бите 20 следует установить на "1", только если все условия ниже верны.

1) Подключенное запоминающее устройство передает данные на чтение по негативному фронту тактового сигнала.

2) Запоминающее устройство поддерживает DQS, а PHY настроен для выборки данные на чтение с помощью сигнала DQS.

В таком случае DQS, отправленный запоминающим устройством, будет содержать на один фронт больше, чем в обычных ситуациях, и PHY должен это компенсировать.

Регистр phy_dq_timing (0x2000)

Регистр контролирует синхронную работу шины данных (DQ) во время передачи от PHY на запоминающее устройство. Можно внести коррективы, чтобы контролировать, когда подключенные DQ будут переключать направление (т.е. добавить задержку в зависимости от того, когда выходной сигнал переключается с входа на выход и с выхода на вход), а также есть возможность определить дополнительную задержку на самой шине данных в единицах ½ такта.

Примечание. Для data_select_oe_end (по умолчанию равен 0x2 и управляет переключением параметра output enable с output на input) всегда необходимо устанавливать значение не менее 0x1. Этот регистр не влияет на процедуру блокировки.

Регистр phy_dqs_timing (0x2004)

Регистр определяет метод выборки данных для чтения, который будет использовать PHY.

Примечание. Чтобы наладить успешное чтение любых данных с устройства, этот регистр необходимо обновлять с учетом метода выборки данных для чтения.

Регистр phy_gate_lpbk_ctrl (0x2008)

Этот регистр контролирует выборку данных для чтения, а также включает в PHY петлевой режим тестирования BIST.

Перед блокировкой необходимо настроить несколько полей. Если поле не упомянуто ниже, его можно оставить в состоянии по умолчанию.

– gate_cfg

– Когда PHY не использует DQS с устройства для выборки данных для чтения, PHY анализирует информацию из контроллера, чтобы примерно определить, в каком периоде тактового сигнала данные для чтения будут возвращаться от устройства, а значит когда следует делать выборки на шине DQ. Если полная задержка всей операции чтения (на основе задержки WRITE CLK PAD + трассировка печатной платы к устройству + трассировка печатной платы от устройства + задержка DQ PAD) превышает один период тактового сигнала, то PHY должен задержать такт, в котором будет отбирать данные. Поэтому, когда DQS от устройства не используется для выборки данных для чтения, в этом поле необходимо установить значение по формуле (полная задержка в нс / период тактового сигнала в нс) с округлением в меньшую сторону.

– rd_del_sel

– Рекомендуется использовать значение, рассчитанное по той же формуле, что и gate_cfg выше, однако с округлением в большую сторону, а затем добавить 2. rd_del_sel = (полная задержка в нс / период тактового сигнала в нс), с округлением в большую сторону + 2.

Регистр phy_dll_master_ctrl (0x200c)

Этот регистр контролирует ведущий DLL, поэтому его очень важно настроить перед блокировкой PHY.

– param_dll_bypass_mode

– По умолчанию — в 1. Когда параметр установлен, ведущий DLL отключается, после чего PHY всегда сразу блокируется. Так активируется режим насыщения. Если пользователь хочет использовать PHY в режиме насыщения, следует установить этот параметр.

– param_phase_detect_sel

– По умолчанию 0x0, однако рекомендуется установить на 0x2.

– param_dll_start_point

– Ведущему DLL может понадобиться время, чтобы измерить период тактового сигнала, а затем выдать сигнал до достижении блокировки. Начинать необходимо с фиксированного количества элементов задержки, а затем добавлять или убирать их, пока не будет найдена оптимальная задержка, подходящая для периода тактового сигнала. Если установить это поле, время блокировки может сократиться. В этом поле необходимо устанавливать значение не выше 7/8 от тактового периода с учетом пессимистичного прогноза задержки. Например, если частота составляет 200 МГц (время периода 5 нс) с пессимистичной задержкой 80 ps, в этом поле можно задать = 5 * (7/8) / 0.080 = 54 элемента. Если время блокировки не имеет значения, можно использовать небольшое значение 0x04, чтобы DLL не блокировался на гармонике. Обратите внимание, что с таким небольшим значением изначальное время блокировки будет дольше. Если значение будет меньше 0x04, DLL может быть не заблокирован.

Регистр phy_dll_slave_ctrl (0x2010)

Этот регистр контролирует ведомые DLL передачи и приема. Когда PHY работает в режиме насыщения, количество элементов задержки можно вручную запрограммировать для обоих DLL. В противном случае каждый DLL будет отложен на 1/256-ю от периода тактового сигнала. Этот регистр не нужно обновлять для изначальной блокировки PHY.

На стороне передачи линию DQ можно отложить с помощью ведомого DLL передачи. На стороне получения интервал выборки откладывается ведомым DLL приема. Хотя может показаться просто выбрать изначальные значения для этих полей, на начальную точку глазковой диаграммы данных могут повлиять время настройки, отклонения между DQS и данными, задержки платы и другие факторы.

Проведение полной последовательности обучения — это единственный способ рассчитать наиболее оптимальную задержку, чтобы найти центр глазковой диаграммы данных. Инструкции ниже помогут выбрать исходные значения.

– Пусть quarter_cycle_period = четверть периода тактового сигнала в ps. Например, если период тактового сигнала составлял 10 нс, установите для этой переменной значение 2500.

– Рассчитайте предполагаемую задержку по каждому реализованному элементу задержки PHY в ps. Элементы задержки в PHY состоят из двух элементов NAND2, выбранных в утвержденных ячейках. Например, если задержка по каждому NAND2 составляет 12 ps, то элемент задержки — 24 ps.

– Назовем эту переменную delay_of_each_element.

– Определите максимальное отклонение в нс между стробом DQS и данным DQ согласно документации на запоминающее устройство (tdqsq). max_dqs_dq_skew = tdqsq.

– Чтобы получить отклонение в виде процента от периода, разделите максимальное отклонение на период тактового сигнала.

– max_dqs_dq_skew_perclk = max_dqs_dq_skew / период тактового сигнала

– Определите время в нс, которое требуется устройству, чтобы обработать данные для чтения после фронта тактового сигнала. Это время также указывается в документации запоминающего устройства. Обратите внимание, что для SDR, а также для каждого фронта для действительных данных в DDR будет указано разное время. Выберите наиболее пессимистичное значение (большее).

– Назовем эту переменную "rx_setup"

– Если rx_setup больше ½ периода тактового сигнала, определите, насколько он превышает ½ периода тактового сигнала в нс.

– rx_setup_gt_half = rx_setup – (clock_period / 2)

– Если rx_setup+4 больше периода тактового сигнала, определите, насколько он превышает период тактового сигнала в нс.

– rx_setup_gt_full = rx_setup + 4 – clock_period

– Определите время настройки в ps, которое потребуется целевому запоминающему устройству, прежде чем отбирать данные на запись. Часто в документации это называется tDVCH.

– Назовем эту переменную "tx_setup"

– Чтобы узнать количество элементов задержки, необходимых для настройки, разделите tx_setup на задержку каждого элемента.

– Назовем эту переменную "tx_setup_de"

– Чтобы узнать количество элементов задержки, необходимых для ¼ периода тактового сигнала, разделите quarter_cycle_period на задержку каждого элемента.

– Назовем эту переменную "quarter_cycle_de"

– Теперь разделите на 5, чтобы получить количество элементов, задержки, необходимых для 1/20-й периода тактового сигнала. Это значение будет выбрано в качестве запаса отклонения при передаче.

– Назовем эту переменную "tx_skew_margin_de"

– Таблица 1.377 поможет вам установить значение в ведомом DLL приема

Таблица 1.377 – Параметры ведомого DLL приема

Параметры	**Исходное значение для DLL приема**
Режим насыщения = 1 Использование DQS от устройства	roundup((max_dqs_dq_skew*1000)/ delay_of_each_element) + 1
Режим насыщения = 1 Использование внутренней петли DQS rx_setup ⇐ половины периода тактового сигнала	1
Режим насыщения = 1 Использование внутренней петли DQS rx_setup > половины периода тактового сигнала rx_setup + 4 ⇐ период тактового сигнала	Достаточно длительное время относительно периода тактового сигнала. Необходимо соответствующим образом отложить изначальную точку выборки. roundup((rx_setup_gt_half*1000)/delay_of_each_element) + 1.
Режим насыщения = 1 Использование внутренней петли DQS rx_setup +4 > периода тактового сигнала	Очень длительное время относительно периода тактового сигнала. Для режимов SDR рекомендуем установить sdr_edge_active, чтобы фронт захвата был отрицательным. roundup((rx_setup_gt_full*1000)/delay_of_each_element) + 1
Режим насыщения = 0 Использование DQS от устройства max_dqs_dq_skew_perclk > 1	255
Режим насыщения = 0 Использование DQS от устройства max_dqs_dq_skew_perclk ⇐ 1	max_dqs_dq_skew_perclk * 256 + 0.01*256

Параметры	Исходное значение для DLL приема
Режим насыщения = 0 Использование внутренней петли DQS rx_setup ⇐ половины периода тактового сигнала	0
Режим насыщения = 0 Использование внутренней петли DQS rx_setup > половины периода тактового сигнала rx_setup + 4 ⇐ период тактового сигнала	roundup(256*(rx_setup_gt_half/clk_period)) + 25; Добавление 25 дополнительных элементов для обеспечения минимального отклонения.
Режим насыщения = 0 Использование внутренней петли DQS rx_setup +4 > периода тактового сигнала	roundup(256*(rx_setup_gt_full/clk_period)) + 25; Добавление 25 дополнительных элементов для обеспечения минимального отклонения.

– Таблица 1.378 поможет вам установить изначальные значения в ведомом DLL передачи

Таблица 1.378 – Параметры ведомого DLL передачи

Параметры	Исходное значение для хранения в DLL передачи
Режим насыщения = 1 setup флеш-памяти занимает больше четверти периода тактового сигнала — запас отклонения (tx_setup_de > (quarter_cycle_de – tx_skew_margin_de))	0x5
Режим насыщения = 1 setup флеш-памяти занимает не более четверти периода тактового сигнала — запас отклонения (tx_setup_de ⇐ (quarter_cycle_de – tx_skew_margin_de))	Установлено 25% т. е. округление в большую сторону (quarter_cycle_de)
Режим насыщения = 0 setup флеш-памяти занимает больше четверти периода тактового сигнала (tx_setup_de > quarter_cycle_de)	0x15
Режим насыщения = 0 setup флеш-памяти занимает не более четверти периода тактового сигнала (tx_setup_de ⇐ quarter_cycle_de)	0x33

– Обратите внимание, что максимальное значение, которое может быть запрограммировано для любого ведомого DLL, составляет 255. Если какие-либо расчеты выше нарушают это условие, то установите значение 255.

1.3.9.10.2.2.7.2 Повторная синхронизация ведомых DLL

Пока работает PHY, ведомые линии задержки время от времени могут требовать повторной синхронизации в результате изменений температуры или OCV и с учетом того, как это изменение меняют задержку элементов задержки. Частота, при которой должна выполняться повторная синхронизация, зависит от того, как сильно динамические факторы способны повлиять на задержку посредством DLL и достаточно ли будет этого изменения, чтобы точка выборки вышла за пределы действительного окна данных. Нет необходимости выполнять повторную синхронизацию, если dll_lock_value не изменилось или изменилось немного — ПО

должно отслеживать dll_lock_value в регистре phy_dll_obs_reg_0 (0x201) и запускать повторную синхронизацию, если dll_lock_value изменилось сильно (например, на 10%) после последней синхронизации. Либо пользователь может решить выполнять повторную синхронизацию периодически, достаточно часто, чтобы учитывать эти изменения в задержке, что не повлияет на общую производительность.

Пока идет повторная синхронизация, невозможно получить доступ к запоминающему устройству. Чтобы настроить автоматические регулярные повторные синхронизации DLL, установите регистр dll_phy_update_cnt (0x1030) на любое значение кроме 0x0. Запрограммированное значение определит интервал времени между каждой повторной синхронизацией в периодах xspi_clk. Обратите внимание, что пока идет повторная синхронизация, интерфейс памяти должен простаивать. Если повторная синхронизация должна быть выполнена, пока контроллер готов обмениваться данными с запоминающим устройством, повторная синхронизация будет автоматически отложена до завершения текущей операции. Аналогичным образом, если повторная синхронизация уже идет, контроллер будет отклонять любую штатную операцию, активируемую хостом, пока повторная синхронизация не будет завершена. Поле resync_idle_cnt регистра dll_phy_ctrl (0x1034) определяет время ожидания после завершения повторной синхронизации до возобновления обмена данными.

1.3.9.10.2.2.8 Целостность данных в запоминающем устройстве

Некоторые запоминающие устройства, например Macronix MX25, оснащены продвинутыми функциями проверки целостности данных. Ниже описана совместимость контроллера с этими функциями.

1.3.9.10.2.2.8.1 ECC (Проверка и исправление ошибок)

Некоторые запоминающие устройства предлагают встроенный алгоритм ECC. Во время работы программы страницы устройство автоматически проведет ECC. Во время операции чтения устройство автоматически проверит и при необходимости исправит ошибки. Эти устройства могут выводит сигнал о состоянии ECC (ECS#), чтобы пользователи следили за ходом функции ECC.

1.3.9.10.2.2.8.2 Проверка четности CRC

Некоторые запоминающие устройства предлагают встроенные возможности проверки CRC. Для операций записи контроллер должен отправить устройству код CRC на проверку. Для операций чтения код CRC передается устройством и должен быть проверен контроллером.

Контроллер может провести CRC по адресу, который рассчитывается операцией исключающего ИЛИ по всем байтам адреса. Код передается сразу после байтов адреса. Также контроллер может проводить CRC на больших фрагментах данных для записи (включают от 8 до 512 байтов). Код CRC вставляется в исходящий поток данных сразу после этого фрагмента данных. Порядок настройки контроллера на использование CRC зависит от активного режима работы. CRC по байтам данных рассчитывается операцией исключающего ИЛИ по всем байтам данных во фрагменте.

1.3.9.10.2.2.8.3 Использование CRC в режиме DIRECT

Чтобы включить CRC, установите бит 8 регистра global_seq_cfg (0x390). Вы можете сделать так, чтобы код CRC всегда следовал после байтов адреса или вставлялся в поток данных после фрагмента данных (вне зависимости от того, что передает поток — байты адреса или байты данных). Это можно контролировать посредством бита 9 того же регистра. Для некоторых устройств требуется сначала отправить код CRC со значением CRC, а затем повторить с инвертированным значением. Этот параметр включается посредством бита 10 того же регистра. Размер фрагмента данных, указывающий, сколько байтов будет защищать CRC, контролируется битами 14:12. Если первый код CRC защищает адрес и часть байтов данных, бит 16 необходимо установить в 1. Если требуется проверить первый код CRC на данные для чтения, также следует установить бит 17.

1.3.9.10.2.2.8.4 Использование CRC в режиме STIG

Параметры, которые определяются в регистре global_seq_cfg для режима DIRECT, описаны в таблице 1.372.

CRC включается посредством поля CRC_EN (бит 117). Для некоторых устройств требуется сначала отправить код CRC со значением CRC, а затем повторить с инвертированным значением. Этот параметр включается посредством поля CRC_INV (бит 118). Размер фрагмента данных, указывающий, сколько байтов будет защищать CRC, контролируется с помощью поля "CRC_Chunk_SIZE" (биты 122:120). Если первый код CRC защищает адрес и часть байтов данных, поле CRC_UAL_CHUNK_EN (бит 116) необходимо установить в 1. Если требуется проверить первый код CRC на данные для чтения, также следует установить поле CRC_UAL_CHUNK_CHK (бит 115).

1.3.9.10.2.2.9 Низкоуровневые функции устройства

При работе в любом из основных режимов (DIRECT или STIG) вам доступны некоторые низкоуровневые функции.

1.3.9.10.2.2.9.1 Защита записи

Некоторые устройства, работающие в одном или в двух режимах, поддерживают сигнал защита записи посредством контакта DQ2. Контроллер поддерживает эту функцию посредством регистра wp_settings (0x1000). Подробнее см. в описании к регистру.

1.3.9.10.2.2.9.2 Сброс устройства

У некоторых устройств есть специальный контакт #RESET, а другие поддерживают аппаратный сброс через контакт DQ3 при работе в единичном или сдвоенном режиме. Контроллер позволяет передать команду RESET с помощью регистра reset_pin_settings (0x1004).

1.3.9.10.2.2.9.3 Настройки SPI CPOL/CPHA

При работе в SDR обмен данными с запоминающим устройством можно осуществлять в режиме SPI MODE 0 (CPOL=0, CPHA=0) или SPI MODE 3 (CPOL=1, CPHA=1). Расположены в регистре (0x1008).

1.3.9.10.2.2.9.4 Настройки задержки сброса JEDEC

Когда запоминающее устройство получает инструкцию сброса JEDEC, в соответствии со стандартом JEDEC контроллер должен соблюдать параметры синхронизации tCSL и tCSH. Если понадобится изменить настройки со значений по умолчанию, с помощью регистра jedec_rst_timing_reg (0x100c) пользователь может запрограммировать конкретные значения.

1.3.9.10.2.2.9.5 Настройки задержки устройства

Регистр dev_delay_reg (0x1010) позволяет установить минимальную задержку между операциями в интерфейсе запоминающего устройства.

1.3.9.10.2.2.9.6 Настройки задержки восстановления после сброса

После аппаратного сброса пользователь может установить минимальную задержку перед тем, как сигнал обращения к микросхеме перейдет в активное состояние. Это можно сделать с помощью rst_recovery_reg (0x1014).

1.3.9.10.2.2.9.7 Приостановление работы контроллера

Некоторым устройствам энергозависимой памяти периодически необходимо обновляться (например, HyperRAM). Это называется синхронизацией tCMS. С помощью регистра dev_active_max (0x1018) и бита seq_tcmd_en из регистра global_seq_reg (0x390) контроллер может установить максимальное время, в течение которого сигнал обращения к микросхеме будет оставаться в активном состоянии, чтобы выполнить периодическое обновление. Контроллер будет автоматически разбивать большие операции, это будет оставаться невидимым для интерфейса хоста.

1.3.9.10.2.2.9.8 Настройка адресации Legacy HyperFlash или устройства профиля 2

По умолчанию адрес хоста необходимо перестроить в структуру "команда-адрес", необходимую для многих Legacy HyperFlash/устройств профиля 2. Обычно поле CA включает зарезервированный раздел, поэтому некоторые биты адреса хоста необходимо сместить. Регистр hf_offset_reg (0x1020) позволяет пользователю настроить сопоставление зарезервированных битов.

1.3.9.10.2.2.10 Подключение нескольких устройств к контроллеру

С помощью CS к контроллеру xSPI можно подключить до 4 запоминающих устройств. Допускается подключение нескольких типов устройств, однако не все из них можно использовать одновременно. Например, если CS 0 подключено к quad SPI NOR Flash, а CS 1 подключено к устройству HyperBus, то при переключении хоста между этими устройствами понадобится перенастраивать регистры xSPI.

1.3.9.10.2.2.11 Работа с устройством SPI NAND

Для устройств SPI NAND адрес запоминающего устройства содержит адреса СТРОКИ и СТОЛБЦА. На рисунке 1.120 представлена структура этого поля. Границу между адресами СТРОК и СТОЛБЦОВ можно установить с помощью поля **seq_page_ca_size в регистре *global_seq_cfg_1 (0x0394)**. Если для выбранных последовательностей требуется выравнивание определенного количества битов по 8 (например, если последовательности требуется 16 битов, содержащих 4 пустых бита, после которых следует 12 битов адреса столбца), то дополнительные (пустые) биты будут установлены в 0.

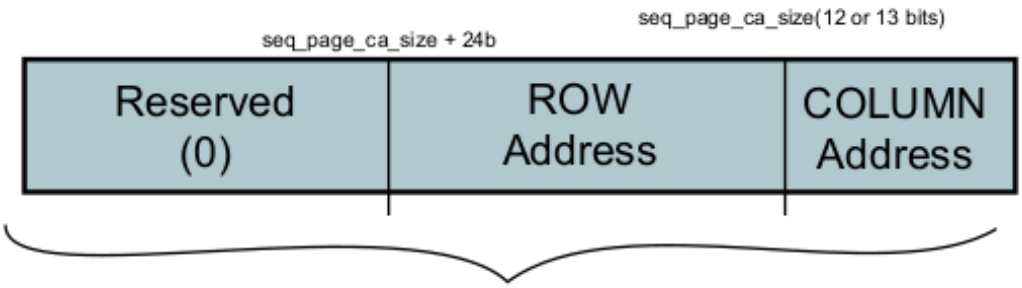


Рисунок 1.120 – Схема адреса памяти для устройств SPI NAND

Устройствам SPI NAND требуется определенная структура страницы. Программное обеспечение должно запрограммировать контроллер таким образом, чтобы он следовал этой структуре.

1.3.9.10.2.3 Как программировать IP

Регистры конфигурации контроллера можно запрограммировать с помощью интерфейсе APB, только когда контроллер находится в состоянии простоя. В противном случае доступ к этим регистрам игнорируется. Чтобы проверить состояние контроллера, прочитайте поле ctrl_busy регистра ctrl_status (0x0100).

1.3.9.10.2.3.1 Конфигурация прерываний

xSPI генерирует одно прерывание, синхронное с тактовой частоты APB. Информация о нем передается через регистр **intr_status (0x0110)**. Каждый отдельный источник прерывания можно маскировать или размаскировать с помощью регистра **intr_enable (0x0114)**. Биты состояния прерывания реализуются как запись для очистки. Чтобы включить прерывание, следуйте инструкциям ниже:

1 Чтобы включить прерывания от контроллера xSPI, необходимо установить бит **intr_en** в регистре **intr_enable (0x0114)**.

2 Отдельные прерывания можно включить с помощью битов регистра **intr_enable (0x0114)**.

Прерывание от контроллера xSPI остается активным, пока все флаги прерывания не будут очищены посредством ПО.

Биты состояния прерывания не очищаются автоматически после включения текущего прерывания. Чтобы предыдущее событие не приводило к активации прерывания в момент включения последнего, рекомендуется сначала очистить бит состояния прерывания, соответствующий событию прерывания.

1.3.9.10.2.3.2 Функциональные ограничения

При работе с Micron Octal, если пользователь хочет применить команды DDR OCTAL FAST READ или DDR OCTAL I/O FAST READ с включенным DQS, то вместо расширенного режима SPI лучше настроить память в режиме Octal SPI.

При работе с HyperRAM не поддерживаются операции записи с дополнительной начальной задержкой. Поскольку CSDA_MIN полностью контролирует синхронизацию восстановления после чтения или записи, пользователю следует установить CSDA_MIN > tRWR.

При работе с Adesto ATXP для чтения состояния из памяти вместо прерывания активного состояния рекомендуется выполнить команду чтения регистра состояния. Контроллер требует, чтобы хост определил абсолютное количество байтов к чтению, что нельзя предугадать для этой команды, поскольку не зафиксирован интервал синхронизации в состоянии занятости. Во время активной фазы чтения данных контроллер всегда создает такт синхронизации, поэтому фаза чтения данных без синхронизации в рамках прерывания активного состояния не поддерживается.

1.3.9.10.2.4 Описания регистров

Регистр команд 0

Запись данных в этот регистр инициализирует новую операцию в xSPI Flash Controller в режиме STIG.

Смещение: 0x0000

Описание полей регистра приведено в таблице 1.379.

Таблица 1.379 – Описание полей регистра

Биты	Название	Описание	Доступ	Сброс
31:0	cmd0	Поле регистра команд 0.	Чтение/ запись	0x0000 0000

Регистр команд 1

Этот регистр используется только в режиме STIG.

Смещение: 0x0004

Описание полей регистра приведено в таблице 1.380.

Таблица 1.380 – Описание полей регистра команд 1

Биты	Название	Описание	Доступ	Сброс
31:0	cmd1	Поле регистра команд 1.	Чтение/ запись	0x0000 0000

Регистр команд 2

Этот регистр используется только в режиме STIG.

Смещение: 0x0008

Описание полей регистра приведено в таблице 1.381.

Таблица 1.381 – Описание полей регистра команд 2

Биты	Название	Описание	Доступ	Сброс
31:0	cmd2	Поле регистра команд 2.	Чтение/ запись	0x0000 0000

Регистр команд 3

Этот регистр используется только в режиме STIG.

Смещение: 0x000C

Описание полей регистра приведено в таблице 1.382.

Таблица 1.382 – Описание полей регистра команд 3

Биты	Название	Описание	Доступ	Сброс
31:0	cmd3	Поле регистра команд 3.	Чтение/ запись	0x0000 0000

Регистр команд 4

Этот регистр используется только в режиме STIG.

Смещение: 0x0010

Описание полей регистра приведено в таблице 1.383.

Таблица 1.383 – Описание полей регистра команд 4

Биты	Название	Описание	Доступ	Сброс
31:0	cmd4	Поле регистра команд 4.	Чтение/ запись	0x0000 0000

Регистр команд 5

Этот регистр используется только в режиме STIG.

Смещение: 0x0014

Описание полей регистра приведено в таблице 1.384.

Таблица 1.384 – Описание полей регистра команд 5

Биты	Название	Описание	Доступ	Сброс
31:0	cmd5	Поле регистра команд 5.	Чтение/ запись	0x0000 0000

Регистр статуса

Этот регистр статуса сообщает статус команды для выбранный цепочки в режиме STIG после завершения операции в устройстве флеш-памяти xSPI.

Смещение: 0x0044

Описание полей регистра приведено в таблице 1.385.

Таблица 1.385 – Описание полей регистра статуса

Биты	Название	Описание	Доступ	Сброс
31:0	cmd_status	Поле регистра статуса команды. При работе в режиме STIG этот регистр сообщает статус завершения STIG.	Только чтение	0x0000 0000

Общий регистр состояния контроллера

Смещение: 0x0100

Описание полей регистра приведено в таблице 1.386.

Таблица 1.386 – Описание полей регистра состояния контроллера

Биты	Название	Описание	Доступ	Сброс
31:17	Зарезервировано		Только чтение	0x0000
16	init_comp	Этот бит устанавливается, когда контроллер xSPI завершил сброс и инициализацию.	Только чтение	0
15:10	Зарезервировано		Только чтение	0x00
9:8	init_fail	Состояние процесса инициализации: 2'b00: устройство xSPI обнаружено, 2'b01: Сбой инициализации, 2'b10: Обнаружено устаревшее устройство SPI, 2'b11: Н/Д.	Только чтение	0x0
7	ctrl_busy	Этот бит указывает, находится ли контроллер в занятом состоянии. 1: Контроллер занят. 0: Контроллер простаивает. Обратите внимание, что этот бит направляется в интерфейс контроллера через контакт ctrl_busy.	Только чтение	0
6	discovery_busy	Когда этот бит установлен, идет обнаружение устройств. Если обнаружение устройств заблокировано, этот бит также устанавливается во время инициализации PHY.	Только чтение	0
5	Зарезервировано		Только чтение	0
4	gcmd_eng_mc_bus y	Этот бит предназначен только для режима STIG. Этот бит указывает, когда контроллер ожидает следующей/последней инструкции в цепочке склеенных инструкций или выполняет запрошенную последовательность в интерфейсе xSPI.	Только чтение	0

Биты	Название	Описание	Доступ	Сброс
3	gcmd_eng_busy	При работе в режиме DIRECT этот бит устанавливается на высоком уровне, если генератор DIRECT CMD занят. Обратите внимание, что пока этот бит на высоком уровне, AHB-slave не будет принимать новые запросы. При работе в режиме STIG этот бит устанавливается, когда занят движок STIG.	Только чтение	0
2:0	Зарезервировано		Только чтение	0

Регистр состояния прерывания

Смещение: 0x0110

Описание полей регистра приведено в таблице 1.387.

Таблица 1.387 – Описание полей регистра состояния прерывания

Биты	Название	Описание	Доступ	Сброс
31:29	Зарезервировано		Только чтение	0x0
28	dir_dev_err	Устанавливается при сбое работы программы в режиме DIRECT. Это значит, что при попытке проверить состояние устройство выдало бит сбоя программы. Для основного xSPI этот бит также устанавливается, когда в режиме DIRECT происходит неисправимая ошибка ECC.	Чтение/ запись W1toClr	0
27	dir_ecc_corr_err	Для основного xSPI этот бит устанавливается, когда в режиме DIRECT происходит ошибка ECC, которую можно исправить.	Чтение/ запись W1toClr	0
26	dir_cmd_err	Обычно этот бит устанавливается, когда хост пытался сделать действие, которое не поддерживается или не корректно. Например, хост мог запустить операцию записи, когда устройство работает в режиме XIP. Полный список условий ошибок описан в разделе "Обработка ошибок/событий" в руководстве пользователя.	Чтение/ запись W1toClr	0
25	dir_dqs_err	Этот бит устанавливается, когда во время чтения или проверки состояния в режиме DIRECT обнаруживается некорректное количество импульсов DQS. Это состояние передается из программного уровня PHY и может означать, что PHY настроен неверно, из-за чего ожидает стробы DQS, но не получает их, либо значение rd_del_sel некорректно, что привело к искажению данных и отклонению указателя в PHY. Чтобы решить эту проблему, необходимо сбросить PHY, чтобы очистить флажки dqs_underrun и dqs_overflow и сбросить указатели чтения данных.	Чтение/ запись W1toClr	0

Биты	Название	Описание	Доступ	Сброс
24	dir_crc_err	Этот бит устанавливается, когда включена проверка CRC, и команда проверки операции чтения или состояния в режиме DIRECT выявила ошибку CRC.	Чтение/ запись W1toClr	0
23	stig_done	Этот бит устанавливается после завершения последней инструкции в склеенной цепочке.	Чтение/ запись W1toClr	0
22	sdma_err	Этот бит устанавливается, когда обнаружен несанкционированный доступ к ведомому интерфейсу DMA.	Чтение/ запись W1toClr	0
21	sdma_trigg	Этот бит устанавливается, когда выполнено условие запуска для ведомого DMA. Он предназначен только для режима STIG, когда необходимо получить доступ в AHB-slave.	Чтение/ запись W1toClr	0
20	cmd_ignored	Рабочий режим STIG: Контроллер выявил, что команда была отправлена, когда движок STIG был уже занят, поэтому проигнорировал ее.	Чтение/ запись W1toClr	0
19:17	Зарезервировано		Только чтение	0
16	ctrl_idle	Это общий статус, указывающий, что контроллер xSPI вернулся в состояние простоя.	Чтение/ запись W1toClr	0
15	gp_open_drain_3	Просто отражает, что во входном контакте xspi_dfi_gp_open_drain[3] был обнаружен переход с высокого на низкий или с низкого на высокий уровень.	Чтение/ запись W1toClr	0
14	gp_open_drain_2	Просто отражает, что во входном контакте xspi_dfi_gp_open_drain[2] был обнаружен переход с высокого на низкий или с низкого на высокий уровень.	Чтение/ запись W1toClr	0
13	gp_open_drain_1	Просто отражает, что во входном контакте xspi_dfi_gp_open_drain[1] был обнаружен переход с высокого на низкий или с низкого на высокий уровень.	Чтение/ запись W1toClr	0
12	gp_open_drain_0	Просто отражает, что во входном контакте xspi_dfi_gp_open_drain[0] был обнаружен переход с высокого на низкий или с низкого на высокий уровень.	Чтение/ запись W1toClr	0
11:0	Зарезервировано		Только чтение	0x000

Регистр включения прерывания

Если установлен выбранный бит этого регистра, то фронт нарастания соответствующего бита в intr_status вызовет прерывание.

Смещение: 0x0114

Описание полей регистра приведено в таблице 1.388.

Таблица 1.388 – Описание полей регистра включения прерывания

Биты	Название	Описание	Доступ	Сброс
31	intr_en	Флажок глобального включения прерываний.	Чтение/ запись	0
30:29	Зарезервировано		Только чтение	0x0
28	dir_dev_err_en	Включает прерывание, когда в режиме DIRECT случается неисправимая ошибка ECC или сбой программы.	Чтение/ запись	0
27	dir_ecc_corr_err_e n	Включает прерывание, когда в режиме DIRECT случается неисправимая ошибка ECC.	Чтение/ запись	0
26	dir_cmd_err_en	Включает прерывание, когда в режиме DIRECT случается недействительная последовательность команд.	Чтение/ запись	0
25	dir_dqs_err_en	Включает прерывание, когда в режиме DIRECT контроллер выдает ошибку DQS после команды проверки состояния или чтения.	Чтение/ запись	0
24	dir_crc_err_en	Включает прерывание, когда в режиме DIRECT контроллер выдает ошибку CRC после команды проверки состояния или чтения.	Чтение/ запись	0
23	stig_done_en	Включает прерывание по завершении инструкции в склеенной цепочке.	Чтение/ запись	0
22	sdma_err_en	Включает прерывание, когда обнаружен несанкционированный доступ к ведомому интерфейсу DMA.	Чтение/ запись	0
21	sdma_trigg_en	Включает прерывание, когда выполнено условие запуска для ведомого DMA.	Чтение/ запись	0
20	cmd_ignored_en	Включает прерывание для обнаружения игнорируемой команды.	Чтение/ запись	0
19:17	Зарезервировано		Только чтение	0
16	ctrl_idle_en	Включает прерывание для обнаружения контроллера, который вернулся в состояние простоя.	Чтение/ запись	0
15	gp_open_drain_3_en	Включает прерывание для обнаружения перехода с высокого на низкий или с низкого на высокий уровень во входном контакте xspi_dfi_gp_open_drain[3].	Чтение/ запись	0
14	gp_open_drain_2_en	Включает прерывание для обнаружения перехода с высокого на низкий или с низкого на высокий уровень во входном контакте xspi_dfi_gp_open_drain[2].	Чтение/ запись	0
13	gp_open_drain_1_en	Включает прерывание для обнаружения перехода с высокого на низкий или с низкого на высокий уровень во входном контакте xspi_dfi_gp_open_drain[1].	Чтение/ запись	0

Биты	Название	Описание	Доступ	Сброс
12	gp_open_drain_0_en	Включает прерывание для обнаружения перехода с высокого на низкий или с низкого на высокий уровень во входном контакте xspi_dfi_gp_open_drain[0].	Чтение/ запись	0
11:0	Зарезервировано		Только чтение	0x000

Регистр прерывания по обнаружению ошибки цепочки

Если установлен выбранный бит этого регистра, фронт нарастания соответствующего бита в trd_error_intr_status приведет к установлению внешней линии прерывания.

Смещение: 0x0134

Описание полей регистра приведено в таблице 1.389.

Таблица 1.389 – Описание полей регистра включения прерывания

Биты	Название	Описание	Доступ	Сброс
31:8	Зарезервировано		Только чтение	0x00 0000
7:0	trd_error_intr_en	Включает прерывание для обнаружения ошибки цепочки.	Чтение/ запись	0x00

Значение счетчика ожидания при длительном опросе

Значение счетчика ожидания при длительном опросе.Значение счетчика ожидания при длительном опросе.

Смещение: 0x0208

Описание полей регистра приведено в таблице 1.390.

Таблица 1.390 – Описание полей регистра значения счетчика ожидания при длительном опросе

Биты	Название	Описание	Доступ	Сброс
31:16	Зарезервировано	Только чтение	0x0000	15:0

Значение счетчика цикла отслеживания статуса

Смещение: 0x020C

Описание полей регистра приведено в таблице 1.391.

Таблица 1.391 – Описание полей регистра значения счетчика цикла отслеживания статуса

Биты	Название	Описание	Доступ	Сброс
31:16	Зарезервировано		Только чтение	0x0000

Биты	Название	Описание	Доступ	Сброс
15:0	short_polling	Минимальное количество системных циклов после задержки длительного опроса перед тем, как контроллер начнет опрашивать статус, если во время первой попытки опроса статуса контроллер был занят. Значение длительного опроса должно быть гораздо больше значения краткого опроса.	Чтение/ запись	0x01F4

Регистр управления устройствами

Смещение: 0x0230

Описание полей регистра приведено в таблице 1.392.

Таблица 1.392 – Описание полей регистра управления устройствами

Биты	Название	Описание	Доступ	Сброс
31:7	Зарезервировано		Только чтение	0x000 0000
6:5	work_mode	Поле для выбора рабочего режима контроллеров. Допустимые значения: 00 - режим DIRECT; 01 - режим STIG;	Чтение/ запись	0x0
4	Зарезервировано		Только чтение	1
3:0	Зарезервировано		Только чтение	0x0

Смещение: 0x023C Зарезервировано

Размер блока данных, переданного в ведомый модуль DMA

Смещение: 0x0240

Описание полей регистра приведено в таблице 1.393.

Таблица 1.393 – Описание полей регистра размера блока данных, переданного в ведомый модуль DMA

Биты	Название	Описание	Доступ	Сброс
31:0	sdma_size	Размер блока данных в байтах, переданного в ведомый модуль DMA. Размер данных округляется до размера слова шины данных.	Только чтение	0x0000 0000

Информация для текущей операции ведомого DMA, связанной с цепочкой выполнения

Смещение: 0x0244

Описание полей регистра приведено в таблице 1.394.

Таблица 1.394 – Описание полей регистра информации для текущей операции ведомого DMA, связанной с цепочкой выполнения

Биты	Название	Описание	Доступ	Сброс
31:9	Зарезервировано		Только чтение	0x00 0000
8	sdma_dirb	Направление передачи, связанное с текущей передачей ведомого DMA (0 - чтение; 1 - запись).	Только чтение	0
7:3	Зарезервировано		Только чтение	0x00
2:0	sdma_trd	Номер цепочки, связанной с блоком передаваемых данных для ведомого модуля DMA.	Только чтение	0x0

Регистр, предназначенный для настройки контроллера из режиме XIP в режиме DIRECT

Смещение: 0x0388

Описание полей регистра приведено в таблице 1.395.

Таблица 1.395 – Описание полей регистра, предназначенного для настройки контроллера из режиме XIP в режиме DIREC

Биты	Название	Описание	Доступ	Сброс
31:24	Зарезервировано		Только чтение	0x00
23:16	xip_dis_mb_val	Значение битов режима, необходимых для отключения режима XIP.	Чтение/ запись	0xFF
15:8	xip_en_mb_val	Значение битов режима, необходимых для включения режима XIP.	Чтение/ запись	0x00
7:0	xip_en	Включение режима XIP для выбранного хранилища памяти. Если режим XIP включен, действительны только последовательности чтения. Вызов любой другой последовательности команд будет игнорироваться, CMD_ERROR/DSC_ERROR/dir_cmd_err будет на переднем фронте.	Чтение/ запись	0x00

Регистр для настройки стандартных значений для последовательностей в режиме DIRECT

Смещение: 0x0390

Описание полей регистра приведено в таблице 1.396.

Таблица 1.396 – Описание полей регистра, предназначенного для настройки контроллера из режиме XIP в режиме DIREC

Биты	Название	Описание	Доступ	Сброс
31:25	Зарезервировано		Только чтение	0x00
24:23	seq_type	Тип последовательности (стандартно для всех последовательностей): 0 – ПРОФИЛЬ 1, 1 – ПРОФИЛЬ 2 – HF (HyperFlash), 2 – ПРОФИЛЬ 2 – HR (HyperRAM), 3 – SPI NAND.	Чтение/ запись	0x0
22	Зарезервировано		Только чтение	0
21	seq_data_per_addr	Выбирает организацию данных памяти xSPI. Обратите внимание, что указатель/адрес xSPI, указанный при отправке команды DIRECT, всегда выравнивается по байту (т. е. если установлено это поле, адрес xSPI должен быть четным). 1’b0 – 1 Б на один адрес памяти (адрес xSPI не переводится), 1’b1 – 2 Б на один адрес памяти (перевод адреса xSPI из байтового адреса в словный адрес будет выполнен автоматически).	Чтение/ запись	0
20	seq_data_swap	Включает обратный порядок байтов. Этот бит может быть установлен, только когда фаза данных отражает режим Octal DDR. В других режимах этот бит должен быть установлен на низком уровне. Допустимые значения: 1’b0 – отключить, 1’b1 – включить.	Чтение/ запись	0
19	Зарезервировано		Только чтение	0
18	seq_tcms_en	Установление этого бита включает синхронизацию tCMS в ПРОФИЛЕ 1. Подробнее см. в регистре dev_active_max в позиции 0x1018. Допустимые значения: 1’b0 – отключить, 1’b1 – включить.	Чтение/ запись	0

Биты	Название	Описание	Доступ	Сброс
17	seq_crc_ual_chunk_chk	Установление этого бита включает CRC-проверку невыровненного фрагмента данных из устройства флеш-памяти. Его можно установить на высоком уровне, только если seq_crc_ual_chunk_en = 1. В противном случае необходимо установить на низком уровне. Допустимые значения: 1’b0 – отключить, 1’b1 – включить.	Чтение/ запись	0
16	seq_crc_ual_chunk_en	Если установить этот бит, адрес команды будет учитываться для определения количества байтов во фрагменте данных для проверки CRC, который должно предоставить устройство флеш-памяти. Допустимые значения: 1’b0 – отключить, 1’b1 – включить.	Чтение/ запись	0
15	Зарезервировано		Только чтение	0
14:12	seq_crc_chunk_size	Это поле указывает количество байтов, после которого будет проводится CRC. Допустимые значения: 3’b000 – не актуально; 3’b001 – 8 Б; 3’b010 – 16 Б; 3’b011 – 32 Б; 3’b100 – 64 Б; 3’b101 – 128 Б; 3’b110 – 256 Б; 3’b111 – 512 Б.	Чтение/ запись	0x2
11	Зарезервировано		Только чтение	0
10	seq_crc_oe	Это поле определяет, ожидает ли контроллер, что устройство xSPI включить CRC-проверку данных на обоих фронтах тактового импульса SPI в виде CRC→CRC последовательность. Допустимые значения: 1’b0 – отключить, 1’b1 – включить.	Чтение/ запись	0
9	seq_crc_variant	Выбор варианта CRC. Допустимые значения: 1’b0 – CRC рассчитывается для всех байтов фазы адреса и добавляется в шину после фазы адреса. 1’b1 – CRC рассчитывается для всех байтов в последовательности и добавляется в шину после всех байтов в последовательности.	Чтение/ запись	0

Биты	Название	Описание	Доступ	Сброс
8	seq_crc_en	Если установить этот бит, будет включен динамический расчет CRC на основе всех предыдущих байтов в текущей последовательности, а значение будет добавлено в интерфейс устройства флеш-памяти xSPI. Не требуется для устаревших устройств Hyper Flash и xSPI профиля 2.0, но может принести пользу для внешнего устройства отслеживания флеш-памяти для обеспечения целостности данных. Допустимые значения: 1’b0 – отключить, 1’b1 – включить.	Чтение/ запись	0
7:4	seq_page_size_pgm	Определяет размер страницы устройства, используемого для операций программирования. Количество байтов на странице = 2^page_size. Допустимые значения: 4’b0000 – 1 Б; 4’b0001 – 2 Б; …, 4’b1000 – 256 Б; 4’b1001 – 512 Б; 4’b1010 – 1024 Б; 4’b1011 – 2048 Б; 4’b1100 – 4096 Б; 4’b1101 – не актуально, …, 4’b1111 – не актуально. Это поле не используется в режиме DIRECT для устройств профиля 2 – HR или когда выбрано устройство SPI NAND.	Чтение/ запись	0x8
3:0	seq_page_size_rd	Определяет размер страницы устройства, используемого для операций чтения. Количество байтов на странице = 2^page_size. Допустимые значения: 4’b0000 – 1 Б; 4’b0001 – 2 Б; …, 4’b1000 – 256 Б; 4’b1001 – 512 Б; 4’b1010 – 1024 Б; 4’b1011 – 2048 Б; 4’b1100 – 4096 Б; 4’b1101 – не актуально, …, 4’b1111 – не ограничено (контроллер отправит все данные в одной команде xSPI).	Чтение/ запись	0xF

Регистр для настройки стандартных значений для последовательностей в режиме DIRECT

Смещение: 0x0394

Описание полей регистра приведено в таблице 1.397.

Таблица 1.397 – Описание полей регистра для настройки стандартных значений для последовательностей в режиме DIRECT

Биты	Название	Описание	Доступ	Сброс
31:30	Зарезервировано		Только чтение	0x0
29:28	seq_plane_cnt	Количество плат в устройстве SPI NAND (закодировано как 2^N): 0 – одна плата; 1 – две платы; 2 – четыре платы; 3 – зарезервировано.	Чтение/ запись	0x0
27	Зарезервировано		Только чтение	0
26:24	seq_page_per_block	Количество страниц на каждый блок для устройства SPI NAND (закодировано как 2^N): 0 – 1 страница на блок; 1 – 2 страницы на блок; … 6 – 64 страницы на блок; 7 – 128 страниц на блок.	Чтение/ запись	0x0
23:17	Зарезервировано		Только чтение	0x00
16	seq_page_ca_size	Ширина адреса столбца для устройств SPI NAND. Значение этого поля помогает рассчитать адрес следующей страницы, если размер данных, указанный в последовательности, превышает вместительность текущей страницы. 1’d0 – 12-битная ширина адреса; 1’d1 – 13-битная ширина адреса.	Чтение/ запись	0
15:9	Зарезервировано		Только чтение	0x00
8:0	seq_page_size_ext	Определяет расширенную область размера страницы (размер резервной области) для устройств SPI NAND. Количество байт данных, передаваемых на/с каждую(ой) страницу(ы), увеличивается на значение из этого поля. Это поле не используется в режиме DIRECT.	Чтение/ запись	0x000

Регистр, где хранится определенная конфигурация, необходимая для работы в режиме DIRECT

Смещение: 0x0398

Описание полей регистра приведено в таблице 1.398.

Таблица 1.398 – Описание полей регистра, где хранится определенная конфигурация, необходимая для работы в режиме DIRECT

Биты	Название	Описание	Доступ	Сброс
31:29	Зарезервировано		Только чтение	0x0
28:16	dac_addr_mask	Эта маска используется для маскирования битов [44:32] системного адреса для передачи на Чтение/ запись для устройств профиля 2.	Чтение/ запись	0x0000
15:13	Зарезервировано		Только чтение	0x0
12	rmp_addr_en	Включает повторное сопоставление адреса в ведомом интерфейсе данных. Когда установлено на 1, входящий адрес AHB-slave будет адаптирован и отправлен на устройство флеш-памяти в виде (адрес – N), где N — это значение, хранящееся в регистре повторного сопоставления адреса (direct_access_rmp в позиции 0x1c).	Чтение/ запись	0
11:10	Зарезервировано		Только чтение	0x0
9	mode_bit_xip_dis	Этот бит используется для выхода из режима XIP в устройстве. Если установлен на 1, в следующей операции чтения контроллер отправит биты режима, указанные в xip_dis_mb_val. Это приведет к отключению режима XIP как на устройстве, так и на контроллере.	Чтение/ запись	0
8	mode_bit_xip_en	Этот бит используется для входа в режим XIP в устройстве. Если установлен на 1, в следующей операции чтения контроллер отправит биты режима, указанные в xip_en_mb_val. Это приведет к переключению устройства и контроллера в режим XIP.	Чтение/ запись	0
7:5	Зарезервировано		Только чтение	0x0

Биты	Название	Описание	Доступ	Сброс
4	rwds_cap_en	Когда этот параметр включен, контроллер поддерживает маскирование байтов в подключенном устройстве, автоматически переводя входящие стробы записи AHB-slave. Если подключенное устройство не поддерживает маскирование байтов RWDS, этот бит следует настроить на низкий уровень. Обратите внимание, что на низком уровне контроллер предполагает, что все байты на канале записи AHB имеют эквивалентный строб записи AHB, заданный на "1". Также не забывайте, что когда этот бит на низком уровне, пользователь не может отправлять операции побайтовой записи в восьмеричное устройство DDR или устройство с 16-битной адресацией.	Чтение/ запись	0
3	Зарезервировано		Только чтение	0 _
2:0	dac_bank_num	Это поле выбирает хранилище (подключенное запоминающее устройство), которое станет целью контроллера при работе в режиме DIRECT.	Чтение/ запись	0x0

Регистр адаптации входящего адреса [31:0] интерфейса AHB

Когда параметр rmp_addr_en установлен на 1, входящий адрес AHB-slave будет адаптирован и отправлен на устройство флеш-памяти в виде (адрес – N), где N[31:0] — это значение, хранящееся в этом регистре.

Смещение: 0x039C

Описание полей регистра приведено в таблице 1.399.

Таблица 1.399 – Описание полей регистра адаптации входящего адреса (31:0) интерфейса AHB

Биты	Название	Описание	Доступ	Сброс
31:0	rmp_addr_val	Повторное сопоставление входящего адреса AHB-slave с другим адресом, используемым устройством флеш-памяти. Значение этого регистра должно быть сопоставлено с 8 байтами.	Чтение/ запись	0x0000 0000

Регистр адаптации входящего адреса [63:32] интерфейса AHB

Когда параметр rmp_addr_en установлен на 1, входящий адрес AHB-slave будет адаптирован и отправлен на устройство флеш-памяти в виде (адрес – N), где N[63:32] — это значение, хранящееся в этом регистре.

Смещение: 0x03A0

Описание полей регистра приведено в таблице 1.400.

Таблица 1.400 – Описание полей регистра адаптации входящего адреса (63:32) интерфейса AHB

Биты	Название	Описание	Доступ	Сброс
31:0	rmp_addr_val_1	Повторное сопоставление входящего адреса ведомого интерфейса данных с другим адресом, используемым устройством флеш-памяти.	Чтение/ запись	0x0000 0000

Регистр настройки последовательности PROGRAM для устройств профиля 1 и SPI NAND в режиме DIRECT

Смещение: 0x0420

Описание полей регистра приведено в таблице 1.401.

Таблица 1.401 – Описание полей регистра настройки последовательности PROGRAM для устройств профиля 1 и SPI NAND в режиме DIRECT

Биты	Название	Описание	Доступ	Сброс
31:30	Зарезервировано		Только чтение	0x0
29:24	prog_seq_p1_dummy_cnt	Количество пустых циклов в устройствах профиля 1. Если 0, пустые циклы отключены.	Чтение/ запись	0x00
23	prog_seq_p1_data_ edge	Выбор между режимами SDR/DDR для фазы данных.	Чтение/ запись	0
22	Зарезервировано		Только чтение	0
21:20	prog_seq_p1_data_ios	Количество линий данных, используемых для отправки фазы данных. 2’b00 – используется одна линия данных (т. е. последовательная передача). 2’b01 – используются две линии данных. 2’b10 – используются четыре линии данных. 2’b11 – используется восемь линий данных.	Чтение/ запись	0x0
19	prog_seq_p1_addr_edge	Выбор между режимами SDR/DDR для фазы адреса.	Чтение/ запись	0
18	Зарезервировано		Только чтение	0
17:16	prog_seq_p1_addr_ios	Количество линий данных, используемых для отправки фазы адреса. 2’b00 – используется одна линия данных (т. е. последовательная передача). 2’b01 – используются две линии данных. 2’b10 – используются четыре линии данных. 2’b11 – используется восемь линий данных.	Чтение/ запись	0x0
15	Зарезервированоb		Только чтение	0

Биты	Название	Описание	Доступ	Сброс
14:12	prog_seq_p1_addr_cnt	Количество байтов адреса.	Чтение/ запись	0x3
11	prog_seq_p1_cmd_edge	Выбор между режимами SDR/DDR для фазы команды.	Чтение/ запись	0
10	Зарезервировано		Только чтение	0
9:8	prog_seq_p1_cmd_ios	Количество линий данных, используемых для отправки фазы команды. 2'b00 – используется одна линия данных (т. е. последовательная передача). 2'b01 – используются две линии данных. 2'b10 – используются четыре линии данных. 2'b11 – используется восемь линий данных.	Чтение/ запись	0x0
7:0	prog_seq_p1_cmd_ val	Мнемоническое значение команды.	Чтение/ запись	0x02

Регистр настройки последовательности PROGRAM для устройств профиля 1 и SPI NAND в режиме DIRECT

Смещение: 0x0424

Описание полей регистра приведено в таблице 1.402.

Таблица 1.402 – Описание полей регистра настройки последовательности PROGRAM для устройств профиля 1 и SPI NAND в режиме DIRECT

Биты	Название	Описание	Доступ	Сброс
31:16	Зарезервировано		Только чтение	0x0000
15:8	prog_seq_p1_cmd_ext_val	Значение расширения команды.	Чтение/ запись	0xFD
7:1	Зарезервировано		Только чтение	0x00
0	prog_seq_p1_cmd_ext_en	Включение расширения команды.	Чтение/ запись	0

Регистр настройки последовательности PROGRAM для устройств профиля 2 в режиме DIRECT

Смещение: 0x0428

Описание полей регистра приведено в таблице 1.403.

Таблица 1.403 – Описание полей регистра настройки последовательности PROGRAM для устройств профиля 2 в режиме DIRECT

Биты	Название	Описание	Доступ	Сброс
31:14	Зарезервировано		Только чтение	0x0 0000

Биты	Название	Описание	Доступ	Сброс
13:8	prog_seq_p2_latency_cnt	Количество циклов задержки для устройств профиля 2 — только HR. Если установить этот бит на 0, циклы задержки будут отключены. Это значение должно быть установлено на "N-1", где "N" — это количество тактовых циклов задержки, ожидаемое запоминающим устройством.	Чтение/ запись	0x00
7:3	Зарезервировано		Только чтение	0x00
2	prog_seq_p2_mask_cmd_mod	Определяет вариант расширения команды для устройства профиля 2. Это влияет на биты [44:40] команды/адреса. Если этот бит установлен на 1, биты CA[44:40] будут состоять только из единиц. Если в режиме DIRECT этот бит установлен на 0, то биты [44:40] команды/адреса будут установлены на (логическая операция И между sAWADDR[45:41] и dac_addr_mask[12:8]).	Чтение/ запись	0
1	prog_seq_p2_burst_type	Тип импульса — соответствует 45-му назначению битов команды/адреса (CA). Хотя пользователь может установить здесь обернутый импульс, в настоящее время сам контроллер не поддерживает обертывание импульсов, поэтому этот параметр всегда должен быть установлен на "1". Значения: Допустимые значения: 1’b0 – обернутый импульс; 1’b1 – линейный импульс.	Чтение/ запись	1
0	prog_seq_p2_target	Целевое пространство — соответствует 46-му назначению битов команды/адреса (CA). Допустимые значения: 1’b0 – пространство памяти, 1’b1 – пространство регистра.	Чтение/ запись	0

Регистр настройки последовательности READ для устройств профиля 1 и SPI NAND в режиме DIRECT

Смещение: 0x0430

Описание полей регистра приведено в таблице 1.404.

Таблица 1.404 – Описание полей регистра настройки последовательности READ для устройств профиля 1 и SPI NAND в режиме DIRECT

Биты	Название	Описание	Доступ	Сброс
31:30	Зарезервировано		Только чтение	0x0

Биты	Название	Описание	Доступ	Сброс
29:24	read_seq_p1_dummy_cnt	Количество пустых циклов. Если 0, пустые циклы отключены. Это поле используется, когда отправка битов режима отключена. В противном случае следует использовать read_seq_p1_mb_dummy_cnt.	Чтение/ запись	0x00
23	read_seq_p1_data_edge	Выбор между режимами SDR/DDR для фазы данных.	Чтение/ запись	0
22	Зарезервировано		Только чтение	0
21:20	read_seq_p1_data_ios	Количество линий данных, используемых для отправки фазы данных. 2’b00 – используется одна линия данных (т. е. последовательная передача). 2’b01 – используются две линии данных. 2’b10 – используются четыре линии данных. 2’b11 – используется восемь линий данных.	Чтение/ запись	0x0
19	read_seq_p1_addr_edge	Выбор между режимами SDR/DDR для фазы адреса.	Чтение/ запись	0
18	Зарезервировано		Только чтение	0
17:16	read_seq_p1_addr_ios	Количество линий данных, используемых для отправки фазы адреса. 2’b00 – используется одна линия данных (т. е. последовательная передача). 2’b01 – используются две линии данных. 2’b10 – используются четыре линии данных. 2’b11 – используется восемь линий данных.	Чтение/ запись	0x0
15	Зарезервировано		Только чтение	0
14:12	read_seq_p1_addr_cnt	Количество байтов адреса.	Чтение/ запись	0x3
11	read_seq_p1_cmd_edge	Выбор между режимами SDR/DDR для фазы команды.	Чтение/ запись	0
10	Зарезервировано		Только чтение	0
9:8	read_seq_p1_cmd_ios	Количество линий данных, используемых для отправки фазы команды. 2’b00 – используется одна линия данных (т. е. последовательная передача). 2’b01 – используются две линии данных. 2’b10 – используются четыре линии данных. 2’b11 – используется восемь линий данных.	Чтение/ запись	0x0
7:0	read_seq_p1_cmd_val	Мнемоническое значение команды.	Чтение/ запись	0x03

Регистр настройки последовательности READ для устройств профиля 1 и SPI NAND в режиме DIRECT

Смещение: 0x0434

Описание полей регистра приведено в таблице 1.405.

Таблица 1.405 – Описание полей регистра настройки последовательности READ для устройств профиля 1 и SPI NAND в режиме DIRECT

Биты	Название	Описание	Доступ	Сброс
31	read_seq_p1_mb_en	Установите на 1’b1, чтобы после байтов адреса были отправлены биты режима, заданные в поле xip_dis_mb_val.	Чтение/ запись	0
30	Зарезервировано		Только чтение	0
29:24	read_seq_p1_mb_dummy_cnt	Количество пустых циклов. Если 0, пустые циклы отключены. Это поле используется, когда отправка битов режима включена.	Чтение/ запись	0x00
23:16	Зарезервировано		Только чтение	0x00
15:8	read_seq_p1_cmd_ext_val	Значение расширения команды.	Чтение/ запись	0xFC
7:5	Зарезервировано		Только чтение	0x0
4	read_seq_p1_cache_random_read_en	Этот бит заставляет алгоритм последовательности чтения использовать команды случайного чтения кэша страницы/последнего случайного чтения кэша страницы. Это поле не используется в режиме DIRECT.	Чтение/ запись	0
3:1	Зарезервировано		Только чтение	0x0
0	read_seq_p1_cmd_ext_en	Включение расширения команды.	Чтение/ запись	0

Регистр настройки последовательности READ для устройств профиля 2 в режиме DIRECT

Смещение: 0x0438

Описание полей регистра приведено в таблице 1.406.

Таблица 1.406 – Описание полей регистра настройки последовательности READ для устройств профиля 2 в режиме DIRECT

Биты	Название	Описание	Доступ	Сброс
31:14	Зарезервировано		Только чтение	0x0 0000

Биты	Название	Описание	Доступ	Сброс
13:8	read_seq_p2_latency_cnt	Количество циклов задержки. Если установить этот бит на 0, тактовые циклы задержки будут отключены. Это значение должно быть установлено на "N-1", где "N" — это количество тактовых циклов задержки, ожидаемое запоминающим устройством.	Чтение/ запись	0x0F
7:4	Зарезервировано		Только чтение	0x0
3	read_seq_p2_hf_bound_en	С помощью этого поля контроллер рассчитывает операцию чтения, пересекающую границу страницы. Оно действительно, только если выбрано устройство профиля 2 – HF. Допустимые значения: 1’b0 – отключить, 1’b1 – включить.	Чтение/ запись	1
2	read_seq_p2_mask_cmd_mod	Определяет вариант расширения команды для устройства профиля 2. Этот бит влияет на биты [44:40] команды/адреса. Если этот бит установлен на 1, биты [44:40] команды/адреса будут установлены на 1. В режиме DIRECT этот бит установлен на 0, то биты [44:40] команды/адреса будут установлены на (логическая операция И между sARADDR[45:41] и dac_addr_mask[12:8]).	Чтение/ запись	0
1	read_seq_p2_burst_type	Тип импульса — соответствует 45-му назначению битов команды/адреса (CA). Хотя пользователь может установить здесь обернутый импульс, в настоящее время сам контроллер не поддерживает обертывание импульсов, поэтому этот параметр всегда должен быть установлен на "1". Значения: 1’b0 – обернутый импульс; 1’b1 – линейный импульс.	Чтение/ запись	1
0	read_seq_p2_target	Целевое пространство — соответствует 46-му назначению битов команды/адреса (CA). Допустимые значения: 1’b0 – пространство памяти, 1’b1 – пространство регистра.	Чтение/ запись	0

Регистр настройки последовательности фиксации включения записи (WEL) для устройств профиля 1 и SPI NAND в режиме DIRECT

Смещение: 0x0440

Описание полей регистра приведено в таблице 1.407.

Таблица 1.407 – Описание полей регистра настройки последовательности фиксации включения записи (WEL) для устройств профиля 1 и SPI NAND в режиме DIRECT

Биты	Название	Описание	Доступ	Сброс
31:25	Зарезервировано		Только чтение	0x00
24	we_seq_p1_en	Включает отправку команды WEL.	Чтение/ запись	1
23:16	we_seq_p1_cmd_ext_val	Значение расширения команды.	Чтение/ запись	0xF9
15	we_seq_p1_cmd_ext_en	Включение расширения команды.	Чтение/ запись	0
14:12	Зарезервировано		Только чтение	0x0
11	we_seq_p1_cmd_edge	Выбор между режимами SDR/DDR для фазы команды.	Чтение/ запись	0
10	Зарезервировано		Только чтение	0
9:8	we_seq_p1_cmd_ios	Количество линий данных, используемых для отправки фазы команды. 2’b00 – используется одна линия данных (т. е. последовательная передача). 2’b01 – используются две линии данных. 2’b10 – используются четыре линии данных. 2’b11 – используется восемь линий данных.	Чтение/ запись	0x0
7:0	we_seq_p1_cmd_val	Мнемоническое значение команды.	Чтение/ запись	0x06

Регистр настройки последовательности проверки статуса для устройств профиля 1 и SPI NAND в режиме DIRECT

Смещение: 0x0450

Описание полей регистра приведено в таблице 1.408.

Таблица 1.408 – Описание полей регистра настройки последовательности проверки статуса для устройств профиля 1 и SPI NAND в режиме DIRECT

Биты	Название	Описание	Доступ	Сброс
31:23	Зарезервировано		Только чтение	0x000
22	stat_seq_p1_data_edge	Выбор между режимами SDR/DDR для фазы данных.	Чтение/ запись	0
21:20	stat_seq_p1_data_ios	Количество линий данных, используемых для отправки фазы данных. 2’b00 – используется одна линия данных (т. е. последовательная передача). 2’b01 – используются две линии данных. 2’b10 – используются четыре линии данных. 2’b11 – используется восемь линий данных.	Чтение/ запись	0x0

Биты	Название	Описание	Доступ	Сброс
19:13	Зарезервировано		Только чтение	0x00
12	stat_seq_p1_addr_edge	Выбор между режимами SDR/DDR для фазы адреса.	Чтение/ запись	0
11:10	stat_seq_p1_addr_ios	Количество линий данных, используемых для отправки фазы адреса. 2'b00 – используется одна линия данных (т. е. последовательная передача). 2'b01 – используются две линии данных. 2'b10 – используются четыре линии данных. 2'b11 – используется восемь линий данных.	Чтение/ запись	0x0
9:8	stat_seq_p1_addr_cnt	Количество байтов адреса для всех последовательностей статуса. Поле кодируется следующим образом: 2'b00 – один байт адреса; 2'b01 – два байта адреса; 2'b10 – три байта адреса; 2'b11 – четыре байта адреса.	Чтение/ запись	0x0
7:6	Зарезервировано		Только чтение	0x0
5	stat_seq_p1_cmd_ext_en	Включение расширения команды.	Чтение/ запись	0
4	stat_seq_p1_cmd_edge	Выбор между режимами SDR/DDR для фазы команды.	Чтение/ запись	0
3:2	Зарезервировано		Только чтение	0x0
1:0	stat_seq_p1_cmd_ios	Количество линий данных, используемых для отправки команды. 2'b00 – используется одна линия данных (т. е. последовательная передача). 2'b01 – используются две линии данных. 2'b10 – используются четыре линии данных. 2'b11 – используется восемь линий данных.	Чтение/ запись	0x0

Регистр настройки последовательности проверки статуса для устройств профиля 1 и SPI NAND в режиме DIRECT

Смещение: 0x0454

Описание полей регистра приведено в таблице 1.409.

Таблица 1.409 – Описание полей регистра настройки последовательности проверки статуса для устройств профиля 1 и SPI NAND в режиме DIRECT

Биты	Название	Описание	Доступ	Сброс
31	Зарезервировано		Только чтение	0

Биты	Название	Описание	Доступ	Сброс
30	stat_seq_p1_ers_fail_addr_en	Включает фазу адреса для проверки статуса сбоя после операции ERASE. Это поле не используется в режиме DIRECT.	Чтение/ запись	0
29:24	stat_seq_p1_ers_fail_dummy_cnt	Количество пустых тактовых циклов, используемых для проверки статуса сбоя после операции ERASE. Это поле не используется в режиме DIRECT.	Чтение/ запись	0x00
23	Зарезервировано		Только чтение	0
22	stat_seq_p1_prog_fail_addr_en	Включает фазу адреса для проверки статуса сбоя после операции PROGRAM.	Чтение/ запись	0
21:16	stat_seq_p1_prog_fail_dummy_cnt	Количество пустых тактовых циклов, используемых для проверки статуса сбоя после операции PROGRAM.	Чтение/ запись	0x00
15:7	Зарезервировано		Только чтение	0x000
6	stat_seq_p1_dev_rdy_addr_en	Включает фазу адреса для проверки статуса готовности/занятости после операции PROGRAM/ERASE и SOFT RESET/READ (только для SPI NAND).	Чтение/ запись	0
5:0	stat_seq_p1_dev_rdy_dummy_cnt	Этот регистр определяет количество пустых тактовых циклов, необходимых для проверки статуса готовности/занятости после операции PROGRAM/ERASE или SOFT RESET/READ (только для SPI NAND).	Чтение/ запись	0x00

Регистр настройки последовательности проверки статуса для устройств профиля 1 и SPI NAND в режиме DIRECT

Смещение: 0x0458

Описание полей регистра приведено в таблице 1.410.

Таблица 1.410 – Описание полей регистра настройки последовательности проверки статуса для устройств профиля 1 и SPI NAND в режиме DIRECT

Биты	Название	Описание	Доступ	Сброс
31:24	stat_seq_p1_prog_fail_cmd_val	Мнемоническое значение для команды, используемой для проверки статуса сбоя после операции PROGRAM.	Чтение/ запись	0x05
23:16	Зарезервировано		Только чтение	0x05
15:8	stat_seq_p1_ers_fail_cmd_val	Мнемоническое значение для команды, используемой для проверки статуса сбоя после операции ERASE. Это поле не используется в режиме DIRECT.	Чтение/ запись	0x05

Регистр настройки последовательности проверки статуса для устройств профиля 1 и SPI NAND в режиме DIRECT

Смещение: 0x045C

Описание полей регистра приведено в таблице 1.411.

Таблица 1.411 – Описание полей регистра настройки последовательности проверки статуса для устройств профиля 1 и SPI NAND в режиме DIRECT

Биты	Название	Описание	Доступ	Сброс
31:24	stat_seq_p1_prog_fail_cmd_ext_val	Значение расширения команды, используемой для проверки статуса сбоя после операции PROGRAM.	Чтение/ запись	0xFA
23:16	Зарезервировано		Только чтение	0xFA
15:8	stat_seq_p1_ers_fail_cmd_ext_val	Значение расширения команды, используемой для проверки статуса сбоя после операции ERASE. Это поле не используется в режиме DIRECT.	Чтение/ запись	0xFA
7:0	stat_seq_p1_dev_rdy_cmd_ext_val	Значение расширения команды, используемой для проверки статуса готовности/занятости после операции PROGRAM/ERASE и SOFT RESET/READ (только для SPI NAND).	Чтение/ запись	0xFA

Регистр настройки последовательности проверки статуса для устройств профиля 2 – HF в режиме DIRECT

Смещение: 0x0460

Описание полей регистра приведено в таблице 1.412.

Таблица 1.412 – Описание полей регистра настройки последовательности проверки статуса для устройств профиля 2 – HF в режиме DIRECT

Биты	Название	Описание	Доступ	Сброс
31:14	Зарезервировано		Только чтение	0x0 0000
13:8	stat_seq_p2_latency_cnt	Количество циклов задержки между чтением CA и STATUS. Это значение должно быть установлено на "N-1", где "N" — это количество тактовых циклов задержки, ожидаемое запоминающим устройством.	Чтение/ запись	0x0F
7:3	Зарезервировано		Только чтение	0x00

Биты	Название	Описание	Доступ	Сброс
2	stat_seq_p2_mask_cmd_mod	Определяет вариант расширения команды для устройства профиля 2. Значение этого бита влияет на биты [44:40] команды/адреса. Если этот бит установлен на 1, биты [44:40] команды/адреса будут установлены на 1. В противном случае биты [44:40] команды/адреса будут установлены на 0.	Чтение/ запись	0
1:0	Зарезервировано		Только чтение	0x0

Регистр настройки последовательности проверки статуса для устройств профиля 1, SPI NAND и профиля 2 – HF в режиме DIRECT

Смещение: 0x0464

Описание полей регистра приведено в таблице 1.413.

Таблица 1.413 – Описание полей регистра настройки последовательности проверки статуса для устройств профиля 1, SPI NAND и профиля 2 – HF в режиме DIRECT

Биты	Название	Описание	Доступ	Сброс
31	Зарезервировано		Только чтение	0
30	stat_seq_prog_fail_en	Включает проверку статуса сбоя после операции PROGRAM.	Чтение/ запись	0
29	stat_seq_prog_fail_size	Размер слова статуса (0-1 Б, 1-2 Б).	Чтение/ запись	0
28	stat_seq_prog_fail_val	Значение, которое будет сравниваться с выбранным битом статуса для обнаружения, что после операции PROGRAM устройство находится в состоянии сбоя.	Чтение/ запись	0
27:24	stat_seq_prog_fail_idx	Это поле определяет, какой бит слова статуса содержит информацию о сбое после команды PROGRAM.	Чтение/ запись	0x0
23:15	Зарезервировано		Только чтение	0x000
14	stat_seq_ers_fail_en	Включает проверку статуса сбоя после операции ERASE. Это поле не используется в режиме DIRECT.	Чтение/ запись	0
13	stat_seq_ers_fail_size	Размер слова статуса (0-1 Б, 1-2 Б). Это поле не используется в режиме DIRECT.	Чтение/ запись	0
12	stat_seq_ers_fail_val	Значение, которое будет сравниваться с выбранным битом статуса для обнаружения, что после операции ERASE устройство находится в состоянии готовности. Это поле не используется в режиме DIRECT.	Чтение/ запись	0

Биты	Название	Описание	Доступ	Сброс
11:8	stat_seq_ers_fail_idx	Это поле определяет, какой бит слова статуса содержит информацию о сбое после команды ERASE. Это поле не используется в режиме DIRECT.	Чтение/ запись	0x0
7	Зарезервировано		Только чтение	0
6	stat_seq_dev_rdy_en	Включает проверку статуса готовности/занятости после операции PROGRAM/ERASE.	Чтение/ запись	1
5	stat_seq_dev_rdy_size	Размер слова статуса (0-1 Б, 1-2 Б).	Чтение/ запись	0
4	stat_seq_dev_rdy_val	Значение, которое будет сравниваться с выбранным битом статуса для обнаружения, что после операции PROGRAM/ERASE или SOFT RESET/READ (только для SPI NAND) устройство находится в состоянии готовности.	Чтение/ запись	0
3:0	stat_seq_dev_rdy_idx	Это поле определяет, какой бит слова статуса содержит информацию о готовности/занятости, которая будет опрашиваться после операции PROGRAM/ERASE или SOFT RESET/READ (только для SPI NAND).	Чтение/ запись	0x0

Регистр настройки последовательности проверки статуса для устройств профиля 1, SPI NAND и профиля 2 – HF в режиме DIRECT

Смещение: 0x046C

Описание полей регистра приведено в таблице 1.414.

Таблица 1.414 – Описание полей регистра настройки последовательности проверки статуса для устройств профиля 1, SPI NAND и профиля 2 – HF в режиме DIRECT

Биты	Название	Описание	Доступ	Сброс
31:0	stat_seq_dev_rdy_addr	Значение адреса, используемое для проверки статуса готовности/занятости после операции PROGRAM/ERASE и SOFT RESET/READ (только для SPI NAND).	Чтение/ запись	0x0000 0000

Регистр настройки последовательности проверки статуса для устройств профиля 1, SPI NAND и профиля 2 – HF в режиме DIRECT

Смещение: 0x0470

Описание полей регистра приведено в таблице 1.415.

Таблица 1.415 – Описание полей регистра настройки последовательности проверки статуса для устройств профиля 1, SPI NAND и профиля 2 – HF в режиме DIRECT

Биты	Название	Описание	Доступ	Сброс
31:0	stat_seq_prog_fail_addr	Значение адреса, используемое для проверки статуса сбоя после операции PROGRAM.	Чтение/ запись	0x0000 0000

Регистр настройки последовательности проверки статуса для устройств SPI NAND в режиме DIRECT

Смещение: 0x0478

Описание полей регистра приведено в таблице 1.416.

Таблица 1.416 – Описание полей регистра настройки последовательности проверки статуса для устройств SPI NAND в режиме DIRECT

Биты	Название	Описание	Доступ	Сброс
31	stat_seq_ecc_fail_en	Включает проверку статуса ECC после операции READ PAGE.	Чтение/ запись	0
30:28	Зарезервировано		Только чтение	0x0
27	stat_seq_crdy_val	Значение, которое будет сравниваться с выбранным битом статуса для обнаружения, что после операции случайного чтения кэша страницы (бит CRBSY) устройство находится в состоянии готовности. Это поле не используется в режиме DIRECT.	Чтение/ запись	0
26:24	stat_seq_crdy_idx	Это поле определяет, какой бит статуса содержит информацию бита занятости при чтении кэша (CRBSY) для операции случайного чтения кэша страницы SPI NAND. Это поле не используется в режиме DIRECT.	Чтение/ запись	0x0
23:16	stat_seq_ecc_corr_val	Значение, которое будет сравниваться со словом статуса, замаскированным полем stat_seq_ecc_fail_mask, чтобы обнаружить, что во время операции чтения страницы SPI NAND устройство выдало исправимую ошибку ECC. С помощью этого значения можно обнаружить одну группу исправимых ошибок от устройства xSPI.	Чтение/ запись	0x00
15:8	stat_seq_ecc_fail_val	Значение, которое будет сравниваться со словом статуса, замаскированным полем stat_seq_ecc_fail_mask, чтобы обнаружить, что во время операции чтения страницы SPI NAND устройство выдало неисправимую ошибку ECC.	Чтение/ запись	0x00

Биты	Название	Описание	Доступ	Сброс
7:0	stat_seq_ecc_fail_mask	Маска, использующаяся для выбора, в каких битах слова статуса хранится статус ECC.	Чтение/ запись	0x00

Регистр сдержит идентификационный номер выпуска

Смещение: 0x0F00

Описание полей регистра приведено в таблице 1.417.

Таблица 1.417 – Описание полей регистра сдержит идентификационный номер выпуска

Биты	Название	Описание	Доступ	Сброс
31:16	xspi_ctrl_magic_number	"Магическое число" контроллера. Это уникальный номер, принадлежащий контроллеру xSPI.	Только чтение	0x6522
15:8	xspi_ctrl_fix	Фиксированный номер (номер младшей версии).	Только чтение	0x02
7:0	xspi_ctrl_rev	Номер версии контроллера.	Только чтение	0x06

Регистр показывает доступные аппаратные возможности контроллера

Смещение: 0x0F04

Описание полей регистра приведено в таблице 1.418.

Таблица 1.418 – Описание полей регистра сдержит идентификационный номер выпуска

Биты	Название	Описание	Доступ	Сброс
31:26	Зарезервировано		Только чтение	0x00
25:24	Зарезервировано		Только чтение	0x1
23:22	sfr_intf	Интерфейс SFR типа 1-APB, другие значения зарезервированы.	Только чтение	0x1
21:20	Зарезервировано		Только чтение	0x3
19:18	Зарезервировано		Только чтение	0x0
17	Зарезервировано		Только чтение	0
16	Зарезервировано		Только чтение	1
15:13	Зарезервировано		Только чтение	0x0
12	asf_available	Функции ASF в наличии.	Только чтение	0
11:4	Зарезервировано		Только чтение	0x00

Биты	Название	Описание	Доступ	Сброс
3:0	n_threads	Количество цепочек, доступных в контроллере. Используется следующее кодирование: 0 – одна цепочка 1 – две цепочки 2 – четыре цепочки 3 – восемь цепочек 4-15 – зарезервировано.	Только чтение	0x3

Регистр защиты от записи

Смещение: 0x1000

Описание полей регистра приведено в таблице 1.419.

Таблица 1.419 – Описание полей регистра защиты от записи

Биты	Название	Описание	Доступ	Сброс
31:2	Зарезервировано		Только чтение	0x0000 0000
1	wp_enable	Включает передачу устройству сигнала защиты от записи (путем переключения направления контактной площадки DQ2).	Чтение/ запись	0
0	wp	Сигнал защиты от записи для всех устройств. Значение этого регистра напрямую связано с выходным сигналом DQ2. Значение может быть изменено, только когда интерфейс устройства флеш-памяти xSPI находится в состоянии простоя. Значение сигнала защиты от записи перезаписывается, если DQ2 представлен контактом действительной операции. Контроллер не проверяет синхронизацию установления/удержания защиты от записи — это должен делать хост. Контроллер не назначает значение защиты от записи во время фазы чтения данных в любой активной передаче. Защита от записи в DQ2 функции поддерживается только устройствами флеш-памяти и только в случае, если контроллер настроен в единичном или сдвоенном режимах SPI.	Чтение/ запись	1

Регистр программно контролируемого аппаратного сброса

Смещение: 0x1004

Описание полей регистра приведено в таблице 1.420.

Таблица 1.420 – Описание полей регистра программно контролируемого аппаратного сброса

Биты	Название	Описание	Доступ	Сброс
31:16	Зарезервировано		Только чтение	0x0000

Биты	Название	Описание	Доступ	Сброс
15	sw_ctrled_hw_rst_bank7	Активирует сигнал программно контролируемого аппаратного сброса в хранилище 7 (т. е. устройство подключается к биту выбора микросхемы 7). Актуально только для устройств, поддерживающих контакт #RESET: 0 – В CS[7] отключена активация программно контролируемого аппаратного сброса. 1– В CS[7] включена активация программно контролируемого аппаратного сброса.	Чтение/ запись	0
14	sw_ctrled_hw_rst_bank6	Активирует сигнал программно контролируемого аппаратного сброса в хранилище 6 (т. е. устройство подключается к биту выбора микросхемы 6). Актуально только для устройств, поддерживающих контакт #RESET: 0 – В CS[6] отключена активация программно контролируемого аппаратного сброса. 1 – В CS[6] включена активация программно контролируемого аппаратного сброса.	Чтение/ запись	0
13	sw_ctrled_hw_rst_bank5	Активирует сигнал программно контролируемого аппаратного сброса в хранилище 5 (т. е. устройство подключается к биту выбора микросхемы 5). Актуально только для устройств, поддерживающих контакт #RESET: 0 – В CS[5] отключена активация программно контролируемого аппаратного сброса. 1 – В CS[5] включена активация программно контролируемого аппаратного сброса.	Чтение/ запись	0
12	sw_ctrled_hw_rst_bank4	Активирует сигнал программно контролируемого аппаратного сброса в хранилище 4 (т. е. устройство подключается к биту выбора микросхемы 4). Актуально только для устройств, поддерживающих контакт #RESET: 0 – В CS[4] отключена активация программно контролируемого аппаратного сброса. 1 – В CS[4] включена активация программно контролируемого аппаратного сброса.	Чтение/ запись	0

Биты	Название	Описание	Доступ	Сброс
11	sw_ctrled_hw_rst_bank3	Активирует сигнал программно контролируемого аппаратного сброса в хранилище 3 (т. е. устройство подключается к биту выбора микросхемы 3). Актуально только для устройств, поддерживающих контакт #RESET: 0 – В CS[3] отключена активация программно контролируемого аппаратного сброса. 1 – В CS[3] включена активация программно контролируемого аппаратного сброса.	Чтение/ запись	0
10	sw_ctrled_hw_rst_bank2	Активирует сигнал программно контролируемого аппаратного сброса в хранилище 2 (т. е. устройство подключается к биту выбора микросхемы 2). Актуально только для устройств, поддерживающих контакт #RESET: 0 – В CS[2] отключена активация программно контролируемого аппаратного сброса. 1 – В CS[2] включена активация программно контролируемого аппаратного сброса.	Чтение/ запись	0
9	sw_ctrled_hw_rst_bank1	Активирует сигнал программно контролируемого аппаратного сброса в хранилище 1 (т. е. устройство подключается к биту выбора микросхемы 1). Актуально только для устройств, поддерживающих контакт #RESET: 0 – В CS[1] отключена активация программно контролируемого аппаратного сброса. 1 – В CS[1] включена активация программно контролируемого аппаратного сброса.	Чтение/ запись	0
8	sw_ctrled_hw_rst_bank0	Активирует сигнал программно контролируемого аппаратного сброса в хранилище 0 (т. е. устройство подключается к биту выбора микросхемы 0). Актуально только для устройств, поддерживающих контакт #RESET: 0 – В CS[0] отключена активация программно контролируемого аппаратного сброса. 1 – В CS[0] включена активация программно контролируемого аппаратного сброса.	Чтение/ запись	0
7:5	Зарезервировано		Только чтение	0x0
4	sw_ctrled_hw_rst_option	Определяет параметры аппаратного СБРОСА следующим образом: 0 – функционал аппаратного сброса устройства включается посредством специализированного контакта. 1 – функционал аппаратного сброса включается посредством контакта DQ3.	Чтение/ запись	0

Биты	Название	Описание	Доступ	Сброс
3:2	Зарезервировано		Только чтение	0x0
1	rst_dq3_enable	Включает передачу команды RESET в порт DQ3 (путем переключения направления контактной площадки DQ3).	Чтение/ запись	0
0	sw_ctrled_hw_rst	Сигнал программно контролируемого аппаратного сброса. Значение этого поля напрямую связано с DQ3 или выходным сигналом RESET (в зависимости от sw_ctrled_hw_rst_option). Значение может быть изменено, только когда интерфейс устройства флеш-памяти xSPI находится в состоянии простоя. Если DQ3 — это контакт действительной операции, значение сигнала программно контролируемого аппаратного сброса перезаписывается. Контроллер не проверяет синхронизацию установления/удержания аппаратного сброса устройства — это должен делать хост. Хост также отвечает за активацию подходящего метода СБРОСА — для этого он выбирает соответствующий номер хранилища и метод СБРОСА (согласно настоящему регистру). Контроллер не назначает значение СБРОСА во время фазы чтения данных в любой активной передаче (поскольку направление передачи переключается на этом этапе).	Чтение/ запись	1

Регистр настройки CPOL/CPHA для устаревшего режима SPI

Этот регистр определяет настройки CPOL/CPHA для устаревшего режима SPI (по умолчанию — 0).

Смещение: 0x1008

Описание полей регистра приведено в таблице 1.421.

Таблица 1.421 – Описание полей регистра программно контролируемого аппаратного сброса

Биты	Название	Описание	Доступ	Сброс
31:1	Зарезервировано		Только чтение	0x0000 0000

Биты	Название	Описание	Доступ	Сброс
0	spi_clock_mode	Определяет тактовый режим SPI. Для передач DDR этот бит всегда должен быть на низком уровне для соответствия синхронизации устройств флеш-памяти DDR. Для передач SDR допустимы следующие значения: 0 – Режим SPI 0 (когда шина SPI в состоянии простоя, тактовая синхронизация на низком уровне); 1 – Режим SPI 3 (когда шина SPI в состоянии простоя, тактовая синхронизация на высоком уровне);	Чтение/ запись	0

Регистр задержки выбора устройства, необходимые для инструкции сброса JEDEC

Смещение: 0x100C

Описание полей регистра приведено в таблице 1.422.

Таблица 1.422 – Описание полей регистра программно контролируемого аппаратного сброса

Биты	Название	Описание	Доступ	Сброс
31:16	Зарезервировано		Только чтение	0x0000
15:8	tCSL_delay	Определяет, сколько циклов xspi_clk входит в синхронизацию tCSL инструкции сброса JEDEC.	Чтение/ запись	0x80
7:0	tCSH_delay	Определяет, сколько циклов xspi_clk входит в синхронизацию tCSH инструкции сброса JEDEC.	Чтение/ запись	0x80

Регистр относительной задержки выбора устройства в отношении интерфейса устройства флеш-памяти xSPI

Смещение: 0x1010

Описание полей регистра приведено в таблице 1.423.

Таблица 1.423 – Описание полей регистра программно контролируемого аппаратного сброса

Биты	Название	Описание	Доступ	Сброс
31:24	csda_min_delay	CSDA_MIN — минимальное время для отмены подтверждения выбора микросхемы.	Чтение/ запись	0x01
23:16	Зарезервировано		Только чтение	0x00
15:8	cseot_delay	CSEOT – завершение передачи при выборе микросхемы. Позволяет улучшить последний активный фронт тактового сигнала с учетом времени отмены подтверждения выбора микросхемы.	Чтение/ запись	0x01

Биты	Название	Описание	Доступ	Сброс
7:0	cssot_delay	CSSOT – начало передачи при выборе микросхемы. Позволяет улучшить время отмены подтверждения выбора микросхемы до первого активного фронта тактового сигнала	Чтение/ запись	0x00

Регистр относительной задержки восстановления после сброса в отношении интерфейса устройства флеш-памяти xSPI

Смещение: 0x1014

Описание полей регистра приведено в таблице 1.424.

Таблица 1.424 – Описание полей регистра относительной задержки восстановления после сброса в отношении интерфейса устройства флеш-памяти xSPI

Биты	Название	Описание	Доступ	Сброс
31:0	rst_recovery	Это поле определяет дополнительную задержку для отмены подтверждения выбора микросхемы с учетом времени восстановления после сброса устройства.	Чтение/ запись	0x0000 000A

Регистр максимальных количеств циклов xspi_clk, в течение которых выбор микросхемы будет оставаться активным в запоминающем интерфейсе

Этот регистр позволяет ввести максимальное количество циклов xspi_clk, в течение которых выбор микросхемы будет оставаться активным (низк.) в запоминающем интерфейсе. Иногда его называют синхронизацией tCMS или tCEM. Как правило, он актуален только для ОЗУ, которым требуется периодическое обновление. Обычно обновление выполняется внутри устройства, но чтобы этой операции нашлось место среди привычных запросов контроллера, контроллеру может понадобиться ограничить время доступа. Пользователь может запрограммировать этот регистр, чтобы заставить контроллер следовать требованию.

Смещение: 0x1018

Описание полей регистра приведено в таблице 1.425.

Таблица 1.425 – Описание полей регистра относительной задержки восстановления после сброса в отношении интерфейса устройства флеш-памяти xSPI

Биты	Название	Описание	Доступ	Сброс
31:0	dev_active_max	Значение в этом поле действительно только в случае, если установлен бит 18 регистра global_seq_cfg в позиции 0x390. При работе в режиме STIG это значение также действительно только в случае, если установлен бит TCMS_EN инструкции STIG. Этот бит следует включать только при работе с ОЗУ, которым требуется ограничение времени для выбора микросхемы малой разрядности (в спецификации устройства обычно называется tCMS или tCEM.	Чтение/ запись	0x0000 0080

Регистр для устаревших устройств Hyper Flash или xSPI профиля 2.0

Этот регистр предназначен только для устаревших устройств Hyper Flash или xSPI профиля 2.0 и может использоваться для настройки расположения "зарезервированной" области поля команды/адреса (CA).

Смещение: 0x1020

Описание полей регистра приведено в таблице 1.426.

Таблица 1.426 – Описание полей регистра относительной задержки восстановления после сброса в отношении интерфейса устройства флеш-памяти xSPI

Биты	Название	Описание	Доступ	Сброс
31:14	Зарезервировано		Только чтение	0x0 0000
13:8	hf_offset_size	Размер позиции зарезервированной области в формате команды.	Чтение/ запись	0x0D
7:6	Зарезервировано		Только чтение	0x0
5:0	hf_offset_index	Начальный индекс зарезервированной области в формате команды.	Чтение/ запись	0x03

Регистр конфигураций повторной синхронизации ведомого DLL устройства PHY

Смещение: 0x1030

Описание полей регистра приведено в таблице 1.427.

Таблица 1.427 – Описание полей регистра конфигураций повторной синхронизации ведомого DLL устройства PHY

Биты	Название	Описание	Доступ	Сброс
31:0	resync_cnt	Это поле определяет интервал времени (в циклах xspi_clk) для отправки обновления (утверждение dfi_ctrlupd_req на высоком уровне) в устройство PHY для повторной синхронизации значений ведомого DLL со значениями ведущего DLL, а также для повторной синхронизации указателей FIFO чтения и записи на пути чтения. Если значение в этом поле равно нулю, контроллер не будет передавать запросы на обновление DLL в PHY. Хост может напрямую контролировать сигнал dfi_ctrlupd_req с помощью поля dfi_ctrlupd_req в регистре dll_phy_ctrl. ПРИМЕЧАНИЕ: Пока эта функция включена, доступ к регистрам PHY невозможен.	Чтение/ запись	0x0000 0000

Регистр конфигураций повторной синхронизации ведомого DLL устройства PHY

Когда PHY используется с контролером xSPI, функция обнаружения устройств автоматически обновляет этот регистр во время инициализации.

Смещение: 0x1034

Описание полей регистра приведено в таблице 1.428.

Таблица 1.428 – Описание полей регистра конфигураций повторной синхронизации ведомого DLL устройства PHY

Биты	Название	Описание	Доступ	Сброс
31:26	Зарезервировано		Только чтение	0x00
25	dfi_ctrlupd_req	Сигнал для повторной синхронизации DLL и указателей FIFO чтения и записи. Чтобы отправить запрос на обновление в PHY, сначала хост должен установить это поле на высоком уровне, а затем дождаться, когда этот бит будет установлен на низком уровне. Этот сигнал не следует использовать, когда включено автоматическая повторная синхронизация, т. е. "dll_phy_update_cnt" не равен нулю.	Чтение/ запись	0
24	dll_rst_n	Сигнал для сброса DLL устройства PHY и начало поиска блокировки.	Чтение/ запись	1
23:22	Зарезервировано		Только чтение	0x0

Биты	Название	Описание	Доступ	Сброс
21	sdr_edge_active	PHY отбирает данные на обоих фронтах тактового генератора отбора. В режиме SDR требуется только один образец. Если этот бит на низком уровне, контроллер передает данные от положительного фронта тактового генератора отбора PHY. Если этот бит на высоком уровне, контроллер передает данные от отрицательного фронта тактового генератора отбора PHY. В режиме DDR этот бит необходимо установить на низком уровне.	Чтение/ запись	0
20	dqs_last_data_drop_en	Этот бит следует установить, когда используемое устройство флеш-памяти отправляет данные по отрицательному фронту тактового генератора флеш-памяти и возвращает их с DQS, а PHY настроен для отбора данных в режиме DQS. В этом случае количество фронтов DQS равно количество запрошенных данных + 1. Если этот бит установлен, в конце передачи контроллер запрашивает избыточные данные, очищая FIFO устройства PHY.	Чтение/ запись	0
19:18	Зарезервировано		Только чтение	0x0
17	extended_wr_mode	Это поле регистра PHY не актуально для xSPI Flash Controller.	Чтение/ запись	1
16	extended_rd_mode	Это поле регистра PHY не актуально для xSPI Flash Controller.	Чтение/ запись	1
15:12	Зарезервировано		Только чтение	0x0
11:8	resync_high_wait_cnt	Это поле определяет количество циклов xspi_clk, в течение которых необходимо утвердить запрос на обновление DLL (dfi_ctrlupd_req) для повторной синхронизации DLL и указателей чтения и записи FIFO.	Чтение/ запись	0x7
7:0	resync_idle_cnt	Это поле определяет время ожидания (в циклах xspi_clk) между отменой подтверждения запроса на обновление DLL (dfi_ctrlupd_req) и возобновлением трафика в PHY.	Чтение/ запись	0x07

1.3.9.10.3 Контроллер SD/eMMC (SD/eMMC)

Контроллер SD обладает следующими характеристиками:

1) поддерживаемые стандарты:

a) SD 6.0 (до UHS-I);

b) SDIO 4.1;

c) MMC/eMMC 5,1;

2) стандартный набор регистров совместимый со стандартной спецификацией контроллера SD версии 6.00

3) настраиваемая ширина шины;

1) обмен данными до 50 Мбайт/с;

2) SoftPhy;

3) генерация маскируемых прерываний;

4) интерфейс программирования - APB slave;

5) интерфейс передачи данных - AHB master;

6) DMA:

a) контроллер простого DMA (SDMA) - DMA с одной операцией;

b) контроллер расширенного DMA (ADMA) - DMA в режиме Scatter-Gather;

c) настраиваемая длина пакета для передач DMA.

1.3.9.10.3.1 Описание архитектуры

1.3.9.10.3.1.1 Структурная схема

На рисунке 1.121 показана структурная схема контроллера SD/eMMC.

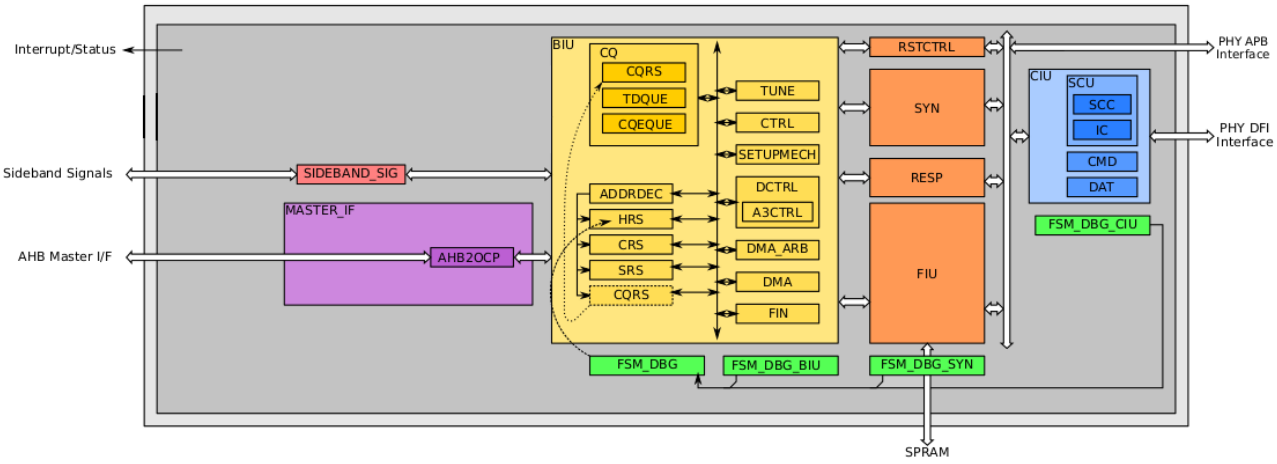


Рисунок 1.121 – Структурная схема контроллера SD/eMMC

Компоненты контроллера:

1) **BIU** - интерфейсный блок шины, состоящий из следующих компонентов:

a) **HRS** - набор регистров контроллера, реализация дополнительных регистров не определена спецификацией контроллера SD (выпущена «SD Association») и зависит от исполнения контроллера SD;

b) **SRS** - набор регистров слотов, реализация регистров определена стандартной спецификацией контроллера SD;

c) **CRS** - набор общих регистров, реализация регистров общей области определена стандартной спецификацией контроллера SD;

d) **CQRS** - регистр организации очереди команд, реализация регистров указаны в пп 1.3.9.10.3.5.58. Регистры CQE Стандарта eMMC 5.1. Регистры расположены в модуле CQ, но доступны как и любые другие регистры;

e) **DMA** - блок, отвечающий за передачу данных между внешней памятью и буферами контроллера. Работает в одном из двух режимов - SDMA или ADMA2. SDMA - простой DMA с одной операцией. ADMA - DMA в режиме Scatter-Gather, который использует таблицу дескрипторов;

f) **ADDRDEC** - декодер адреса для транзакций;

g) **CTRL** - модуль управления выполнением команд;

h) **DCTRL** - модуль управления передачей данных включает в себя движки SDMA, ADMA2 и ADMA3;

i) **SETUPMECH** - блок автоматической настройки на базе DMA;

j) **FIN** - входной мультиплексор/арбитр FIFO, упрощает доступом к внутреннему FIFO. Два компонента имеют доступ к алгоритму FIFO - регистр буфера (SRS08) и DMA.

k) **DMA_ARB** - арбитр DMA, разделяет доступ к **DMA** между **DCTRL** и **CQ**;

l) **TUNE** - модуль управления настройкой UHS-I;

m) **CQ Engine** - модуль управления, который обрабатывает команды, запрошенные через регистры CQ:

i) **TDQUE** - регистр очереди задач, список запрошенных задач, ожидающих передачи устройству;

ii) **CQEQUE** - очередь выполнения, список задач, отправляемых устройству, которые не выполняются;

2) **RSTCTRL** - контроллер сброса, модуль выполняет программный сброс во всех областях тактовых сигналов;

3) **SYN** - модуль синхронизации между блоками BIU (область тактовых сигналов clk) и CIU (область тактовых сигналов sdmclk);

4) **SREG** - общие регистры с пакетами команд / пакетами ответов / неподдерживаемыми ответами;

5) **FIU** - интерфейсный блок FIFO, реализующий механизм буферизации;

6) **CIU** - интерфейсный модуль карты:

a) **CMD** - модуль управления линией CMD, включает в себя логику управления передачей команд;

b) **DAT** - модуль управления линией DAT, включает в себя логику управления передачей команд;

c) **IC** - контроллер прерывания, включает в себя логику обнаружения прерывания карты SDIO (SD режим);

d) **SCC** - контроллер тактирования, включает в себя делитель частоты для интерфейса SD (SD режим);

e) **SCU** - блок управления слотами, содержит экземпляры **IC** и **SCC**;

7) **MASTER_IF** - функция-обертка в транзакцию между внутренним интерфейсом, управляемым DMA, и внешним интерфейсом, используемым для интеграции;

8) **Модули отладки**:

a) **FSM_DBG_BIU** - сбор состояний FSM из модуля **BIU** и субкомопонентов;

b) **FSM_DBG_CIU** - сбор состояний FSM из модуля **CIU** и субкомопонентов;

c) **FSM_DBG** - блок, который декодирует запрос от HRS и предоставляет текущий статус FSM от любого из блоков **FSM_DBG**

1.3.9.10.3.2 Карта адресов регистров

1.3.9.10.3.2.1 Карта памяти

В таблице 1.429 указаны регистры контроллера.

Таблица 1.429 – Регистры контроллера SD/eMMC

Начало	Конец	Название	Описание
000h	0EFh	Набор главных регистров (HRS)	Регистры управления и состояния. Эти регистры являются дополнительными и не указаны в Стандартной спецификации контроллера SD («SD Card Association»).
2F0h	2FFh	Набор общих регистров (CRS)	Регистры состояний, определенные Стандартной спецификацией контроллера SD («SD Card Association»).

Начало	Конец	Название	Описание
200h	2EFh	Набор регистров слотов (SRS)	Набор регистров слотов, определенных Стандартной спецификацией контроллера SD для слота SD/eMMC.
400h	4FFh	Набор регистров организации очереди команд (CQRS)	Регистры управления и состояний, определенные Стандартной спецификацией eMMC (JEDEC), касающейся организации очереди команд.

Примечание: Незадействованные поля/регистры не описаны. Чтение - 00000000h. Запись в эти регистры игнорируется.

1.3.9.10.3.3 Описание функционирования

1.3.9.10.3.3.1 Доступ к регистрам Combo PHY

Программное обеспечение имеет доступ к регистрам Combo PHY через HRS04 (адрес регистров PHY) и HRS05 (порт данных регистров PHY). Регистры PHY можно читать или записывать следующим образом. Следующий этап можно начинать после завершения предыдущего этапа.

Доступ для чтения

1) Программа устанавливает адрес регистра PHY HRS04.

2) Программа читает порт данных регистра PHY HRS05. Значение в HRS05 - это значение из регистра PHY.

Доступ для записи

1) Программа устанавливает адрес регистра PHY HRS04.

2) Программа устанавливает порт данных регистра PHY HRS05. Значение в регистре HRS05 обновляет регистр PHY во время доступа для записи.

1.3.9.10.3.3.2 Спецификация DMA

Контроллер SD/eMMC поддерживает два режима DMA:

– SDMA - реализован и указан в Стандартной спецификации контроллера SD версии 1.00

– ADMA2 - реализован и указан в Стандартной спецификации контроллера SD версии 2.00

Примечание: ADMA1, реализованный и указанный в Стандартной спецификации контроллера SD версии 2.00, стал устаревшим в стандарте и не поддерживается контроллером SD.

Режим DMA для текущей передачи выбирается через регистр SRS10.DMASEL и может быть разным для каждой следующей передачи данных. Драйвер контроллера может изменить режим DMA, либо если не установлены биты состояния включения передачи данных записи (SRS09.WTA) и включения передачи данных чтения (SRS09.RTA).

DMA поддерживает режимы 64-битовой и 32-битовой адресации.

В таблице 1.430 указано как выбрать DMA и режим адресации, задав поля регистра SRS10.DMASEL, SRS15.HV4E и SRS15.A64B. Программный драйвер должен избегать настроек с «-» в поле «Режим DMA».

Таблица 1.430 – Режим DMA

SRS10.DMASEL	SRS15.HV4E	SRS15.A64B	Режим DMA
0	0	0	SDMA 32 бит
		1	-
	1	0	SDMA 32 бит
		1	SDMA 64 бит
1	0	0	-
		1	-
	1	0	-
		1	-
2	0	0	ADMA2 32 бит
		1	-
	1	0	ADMA2 32 бит
		1	ADMA2 64 бит
3	0	0	-
		1	ADMA2 64 бит
	1	0	-
		1	-

Передача DMA в каждом режиме может быть остановлена путем задания остановки в бите запроса промежутка между блоками (SRS10.SBGR). Передачи DMA можно повторно запустить, установив бит запроса продолжения (SRS10.CR).

В случае возникновения ошибки драйвер контроллера может отметить передачу DMA в каждом режиме, установив программный сброс для линии DAT (SRS11.SRCMD) и передав команду отмены (в случае выполнения передачи нескольких блоков).

1.3.9.10.3.3.2.1 SDMA

Простой режим DMA (с одной операцией) использует регистры контроллера SD для описания передачи данных. Регистр адреса SDMA (SRS00.SAAR или SRS22.DMASA1 /

SRS23.DMASA2) определяет базовый адрес блока данных. Длина передачи данных определяется числом блоков (SRS01.BCCT) и размером передаваемых блоков (SRS01.TBS). Значение адреса SDMA не ограничено - блок данных может начинаться с любого адреса (адрес не обязательно должен быть выровнен по какой-либо границе).

SDMA ожидает на каждой границе, указанной в регистре границы буфера SDMA (SRS01.SDMABB или U2SDMABB).

После достижения границы буфера контроллер SD останавливает текущую передачу и генерирует прерывание. Программа должна обновить регистр адреса SDMA, чтобы продолжить передачу.

Когда SDMA останавливается на границе буфера, регистр адреса SDMA указывает адрес для следующих передаваемых данных. SDMA перезапускает передачу после записи самого старшего байта в регистре адреса SDMA.

SDMA не использует регистр состояния ошибки ADMA (SRS21). Схема работы SDMA показана на рисунке 1.122.

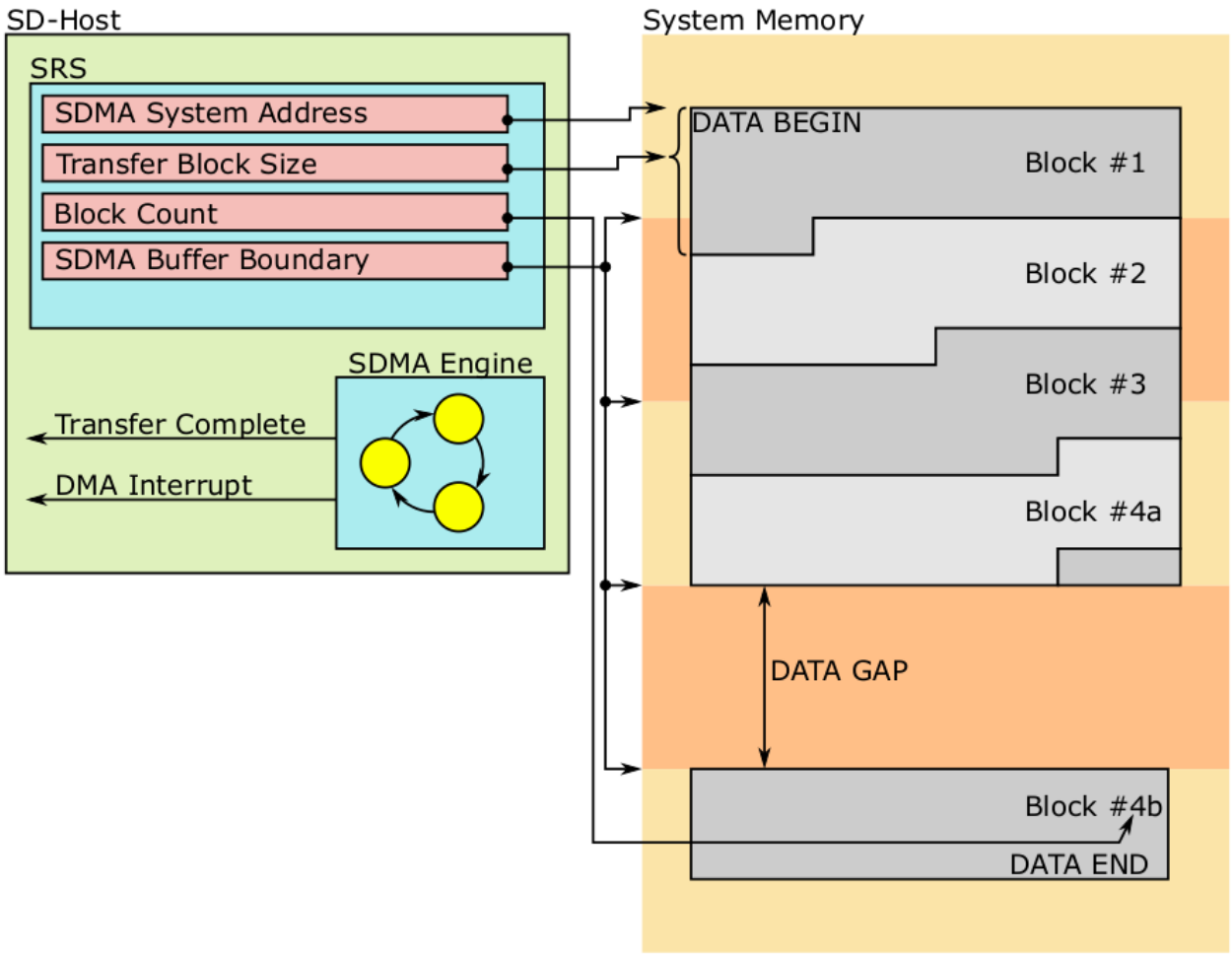


Рисунок 1.122 – Схема работы SDMA

1.3.9.10.3.3.2.2 ADMA2

Описание

Режим расширенного DMA версии 2 (ADMA2) использует список дескрипторов для описания передач данных. Регистры контроллера SD только определяют базовый адрес списка дескрипторов. В дескрипторах указываются базовые адреса и размеры страниц данных. Режим ADMA2 определен в Спецификации контроллера SD версии 2.00 (а также более новых версиях спецификации).

Контроллер SD поддерживает ADMA2 в режиме 64-битовой или 32-битовой адресации.

Страницы данных

В режиме ADMA2 контроллер SD передает данные постранично. Страница - блок данных, определенный одним дескриптором ADMA2. Каждый дескриптор ADMA2 может

определить только одну страницу данных. Начальный адрес страницы данных должен быть выровнен по 4-байтовой границе (2 младших бита устанавливаются на 0) в режиме 32-битовой адресации и по 8-байтовой границе (3 младших бита устанавливаются на 0) в режиме 64-битовой адресации. Размер каждой странице данных является произвольным и не зависит от размера предыдущей или следующей страницы данных. Он также может отличаться от размера блока данных SD-карты (SRS01.TBS).

Таблица дескрипторов ADMA2

Передачи ADMA2 настраиваются в списке дескрипторов. Базовый адрес списка устанавливается в регистре адреса ADMA (SRS22.DMASA1, SRS23.DMASA2), независимо от того, выполняется передача данных чтения или записи. Список дескрипторов ADMA2 состоит из ряда 64-битовых / 96-битовых / 128-битовых дескрипторов с различными функциями (см рисунок 1.123). Каждый дескриптор может:

– Выполнять передачу страницы данных указанного размера

– Выполнять переход на произвольный адрес следующего дескриптора

Кроме того, каждый дескриптор имеет следующие флаги:

– VAL - обозначение действительного дескриптора ADMA;

– END - обозначение последнего дескриптора в списке;

– INT - обозначение запроса прерывания после завершения дескриптора.

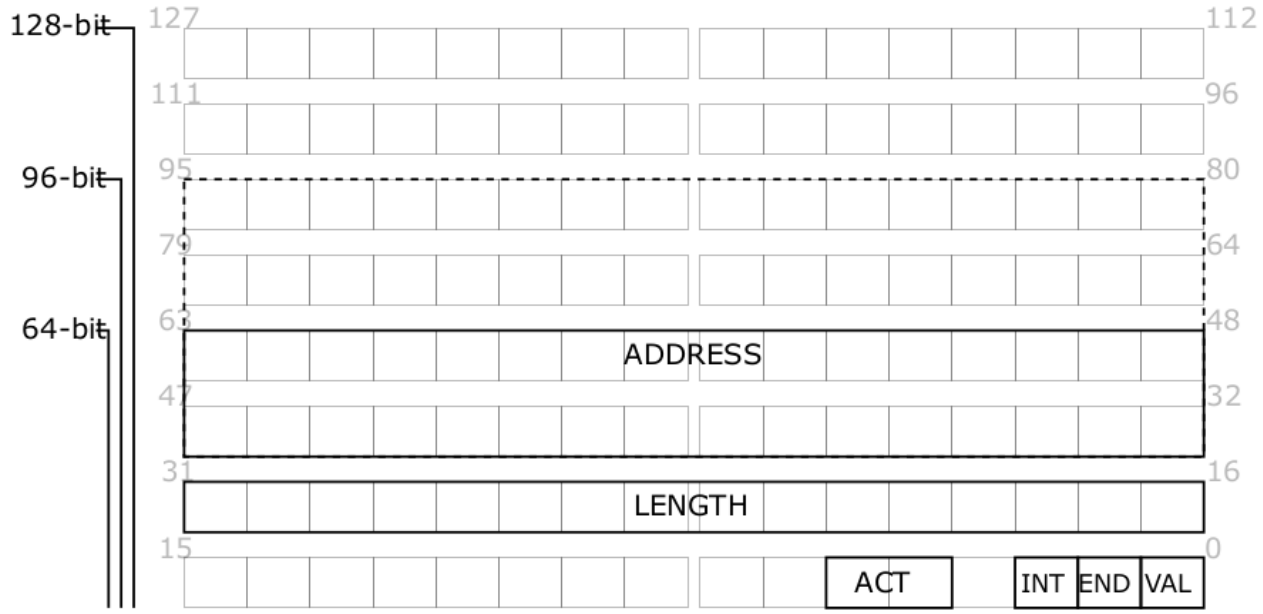


Рисунок 1.123 – Схема дескрипторов ADMA2

Описание полей дескрипторов ADMA показано в таблице 1.431.

Таблица 1.431 – Поля дескрипторов ADMA

Бит	Обозначение	Описание
95…32 / 63…32	**ADDRESS**	Поле содержит адрес страницы данных или адрес следующего дескриптора, в зависимости от типа дескрипторов. Если тип дескриптора - TRAN, после содержит адрес страницы. Если тип дескриптора - LINK, поле содержит адрес следующего дескриптора в списке
31..16	**LENGTH**	Поле содержит длину страницы данных в байтах. Если поле содержит 0, длина страницы равна 64 кбайтам
5..4	**ACT**	Поле определяет тип дескриптора: * 2’b00 (NOP) - нет операции, переход к следующему дескриптору в списке; * 2’b01 (зарезервировано) - аналогично NOP; * 2’b10 (TRAN) - передача данных из указанной страницы и переход к следующему дескриптору в списке; * 2’b11 (LINK) - переход к следующему дескриптору в списке, указанному в поле ADDRESS этого дескриптора
2	**INT**	Если этот бит установлен, генерируется прерывание (SRS12.DMAINT), когда ADMA2 завершает обработку дескриптора.
1	**END**	Если этот бит установлен, он сигнализирует об остановке передачи и генерирует прерывание после завершения передачи.
0	**VAL**	Когда этот бит установлен, он указывает действительный дескриптор в списке. При сборе этого бита генерируется прерывание по ошибке ADMA, и ADMA2 останавливает обработку списка дескриптора. Этот бит предотвращает выход ADMA2 из под контроля из-за неверных дескрипторов

На рисунке 1.124 представлена схема работы ADMA2:

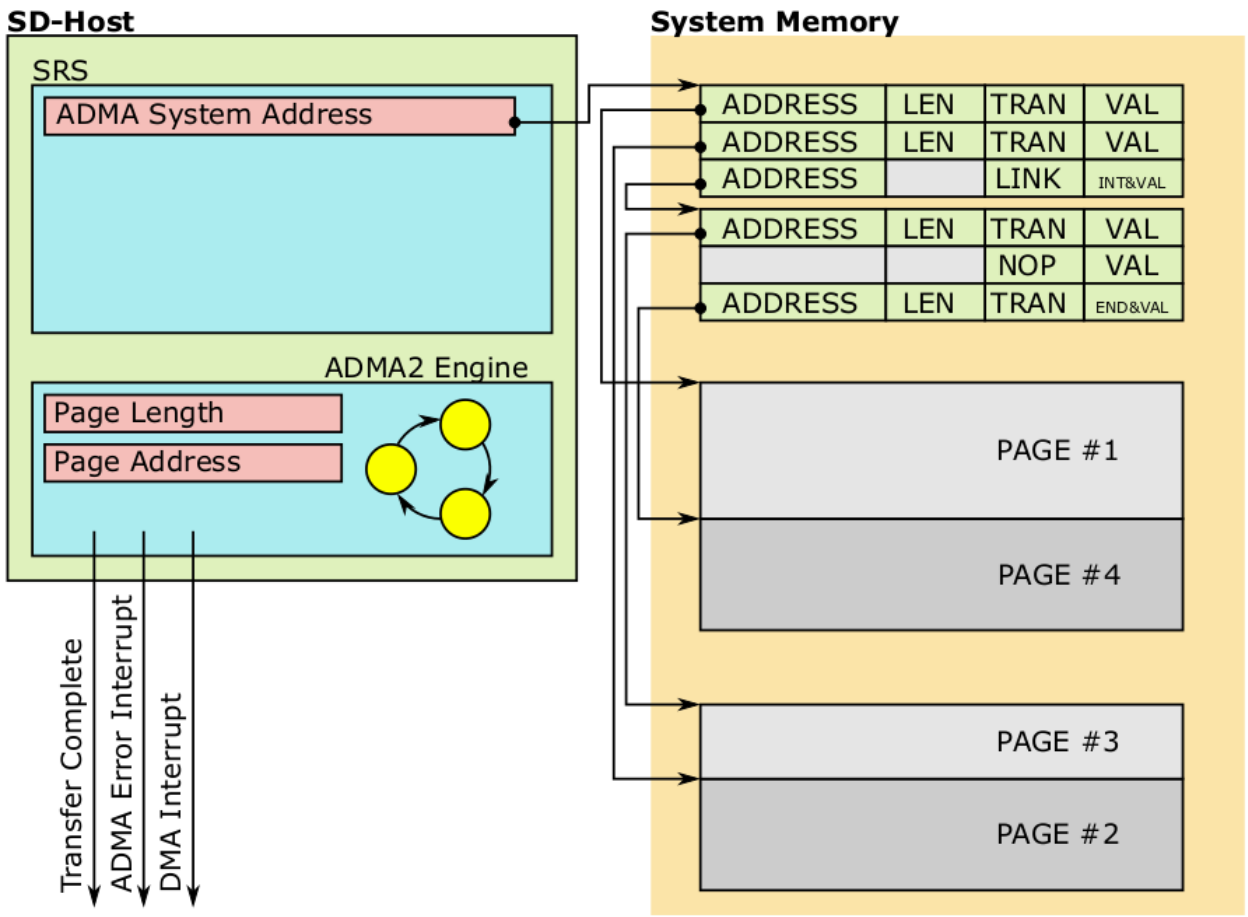


Рисунок 1.124 – Схема работы ADMA2

1.3.9.10.3.3.2.3 Ошибки DMA

Контроллер предусматривает следующие механизмы защиты:

– валидация дескрипторов - DMA в режиме Scatter-Gather проверяет, установлен ли бит **VAL** в дескрипторах. Этот механизм используется для дескрипторов всех типов, включая ADMA2 (SD) и организации очереди команд (eMMC). Если **VAL** установлен на 0, DMA останавливает передачу и сообщает об ошибке DMA.

– Обнаружение ошибки транзакции AHB - этот механизм позволяет DMA останавливать ожидающие передачи и сообщать об ошибке через регистры состояний. DMA не передает новые запросы, завершает все уже запрошенные/находящиеся в очереди транзакции и останавливает передачу, когда все транзакции завершены, без нарушения протокола шины.

Кроме того, в случае транзакции чтения данные чтения изменяются на 0x00, что защищает DMA в режиме Scatter-Gather от потенциально произвольных данных в качестве

дескриптора. При ошибке транзакции шины контроллер сообщает об ошибке AHB (HRS03[3:0]) и может сообщить об ошибке DMA (SRS12[25]).

1.3.9.10.3.3.3 Синхронизация SD/eMMC

1.3.9.10.3.3.3.1 Описание

Синхронизация SD и eMMC требует определенных параметров контроллера и регистров Combo PHY. Драйвер программы / пользовательского приложения ответственен за обновление определенных регистров по рассчитанным параметрам.

Расчет параметров применим для всех режимов скорости. Режимы более высокой скорости требуют дополнительной подготовки, указанной в 1.3.9.10.3.3.3.2.

1.3.9.10.3.3.3.2 Настройка

Спецификации стандарта контроллера SD/физического уровня определяют режим скорости UHS-I SDR104, Спецификация стандарта eMMC 5.1 определяет режимы скорости HS200 и HS400. Эти режимы не определяют положение окна действительных данных относительно тактовых сигналов, а задают ширину действительного окна и обеспечивают его отправку от устройства в течение 0-2 тактовых циклов.

Режимы можно использовать после регулировки точки выборки относительно центра плавающего окна действительных данных. Процедура одинаковая для обоих стандартов. Она предусматривает 40 итераций, в течение которых хост отправляет команду шаблона настройки чтения. В каждой итерации настраивается точка выборки данных (увеличивается). Цель - полное покрытие как минимум одного тактового периода SDCLK, изменив фазу тактового сигнала выборки с 0 на как минимум 360 градусов на следующем этапе.

Каждый этап начинается с настройки точки выборки, контроллер передает команду и получает ответ и шаблон данных чтения.

После завершения этапа контроллер получает информацию об успешном выполнении итерации (смог ли контроллер получить действительный ответ и шаблон данных, или хост обнаружил ошибку протокола).

С учетом статистики успешных итераций после настройки программное/аппаратное обеспечение может задать точку выборки, близкую к середине окна действительных данных.

Последовательность настройки немного отличается для SD (1.3.9.10.3.3.3.3) и eMMC (1.3.9.10.3.3.3.4).

На рисунке 1.125 показаны моменты, когда необходимо обновление параметров контроллера/физического уровня во время переключения между режимами скорости.

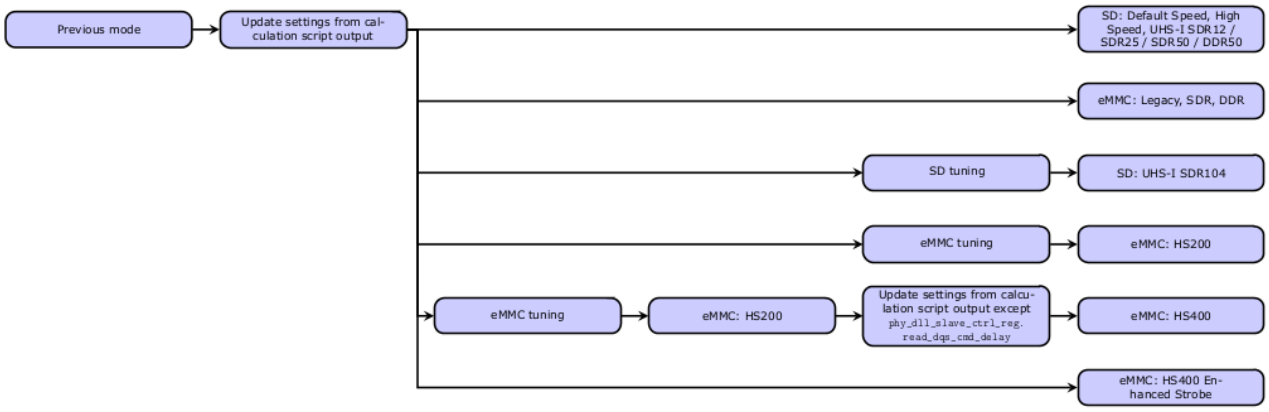


Рисунок 1.125 – Обновление параметров

1.3.9.10.3.3.3.3 Последовательность настройки для SD

Последовательность показана на рисунке 1.126. Программное обеспечение передает 40 CMD19, аппаратное обеспечение регулирует точку выборки данных (параметры DLL физического уровня) и «разбирает» статистику успешных/неуспешных операций. На основании этой статистики хост либо переносит точку выборки в середину наиболее широкого набора итераций, либо сообщает, что эта операция невозможна, так как нет успешных итераций.

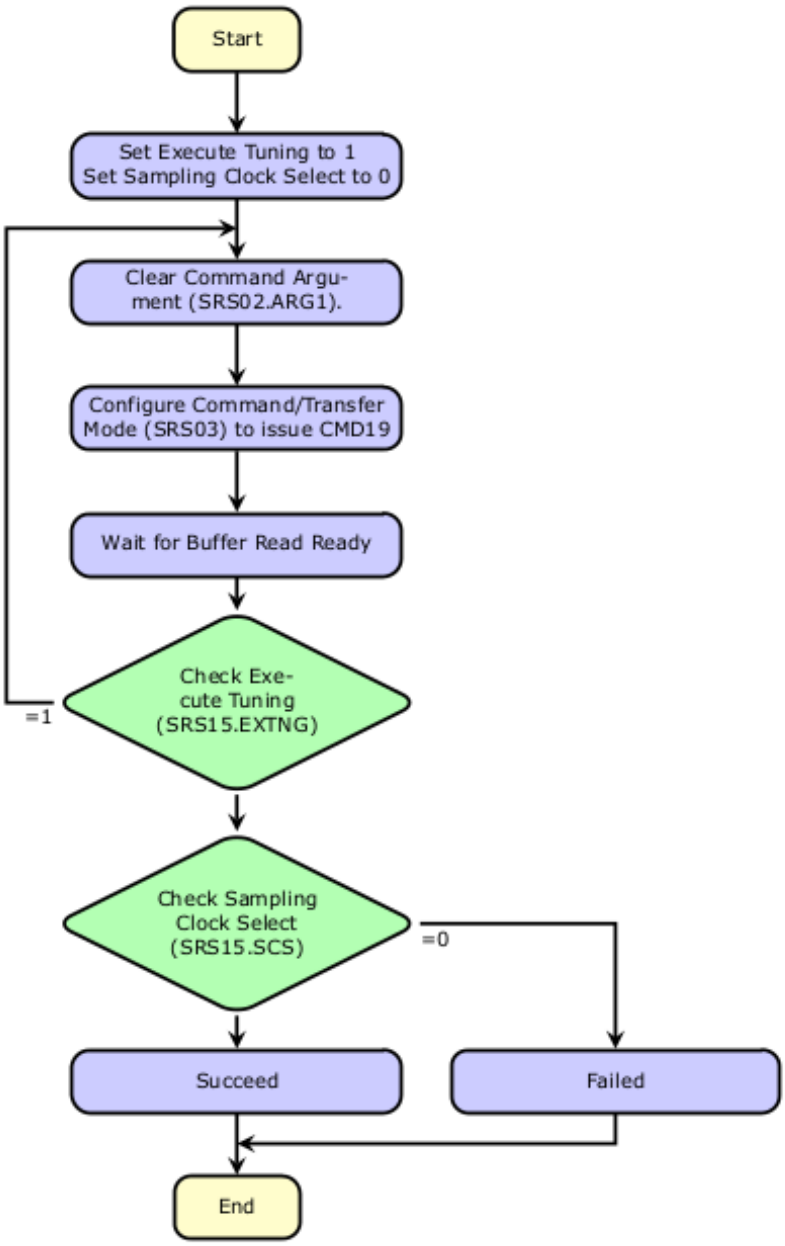


Рисунок 1.126 – Последовательность настройки для SD

1.3.9.10.3.3.3.4 Последовательность настройки для eMMC

Последовательность показана на рисунке 1.127. Она состоит из 40 команд передачи шаблона настройки данных (CMD21). При каждой команде программное обеспечение проверяет состояния команды и данных, затем сравнивает полученный блок данных с шаблоном блока настройки.

Контроллер может сообщить об ошибке протокола в любой итерации. В этом случае программное обеспечение запускает базовую функцию устранения ошибок, которая

заключается в настройке программного сброса для CMD (SRS11.SRCMD) и сброса программного обеспечения для DAT (SRS11.SRDAT) на 1. Затем программное обеспечение дожидается завершения сброса.

После разборки результатов итераций программа ищет наиболее широкий диапазон успешных итераций. Программа проверяет результаты итераций после обертки через 39 в 0. Например, если были успешными итерации 0-4 и 38-39, программа определяет центр в положении 1.

Параметры для центра окна данных обновляются в настройках DLL (phy_dll_slave_ctrl_reg). Если итерация не была успешна, программа переключает хост и устройство в другой режим с более низкой скоростью.

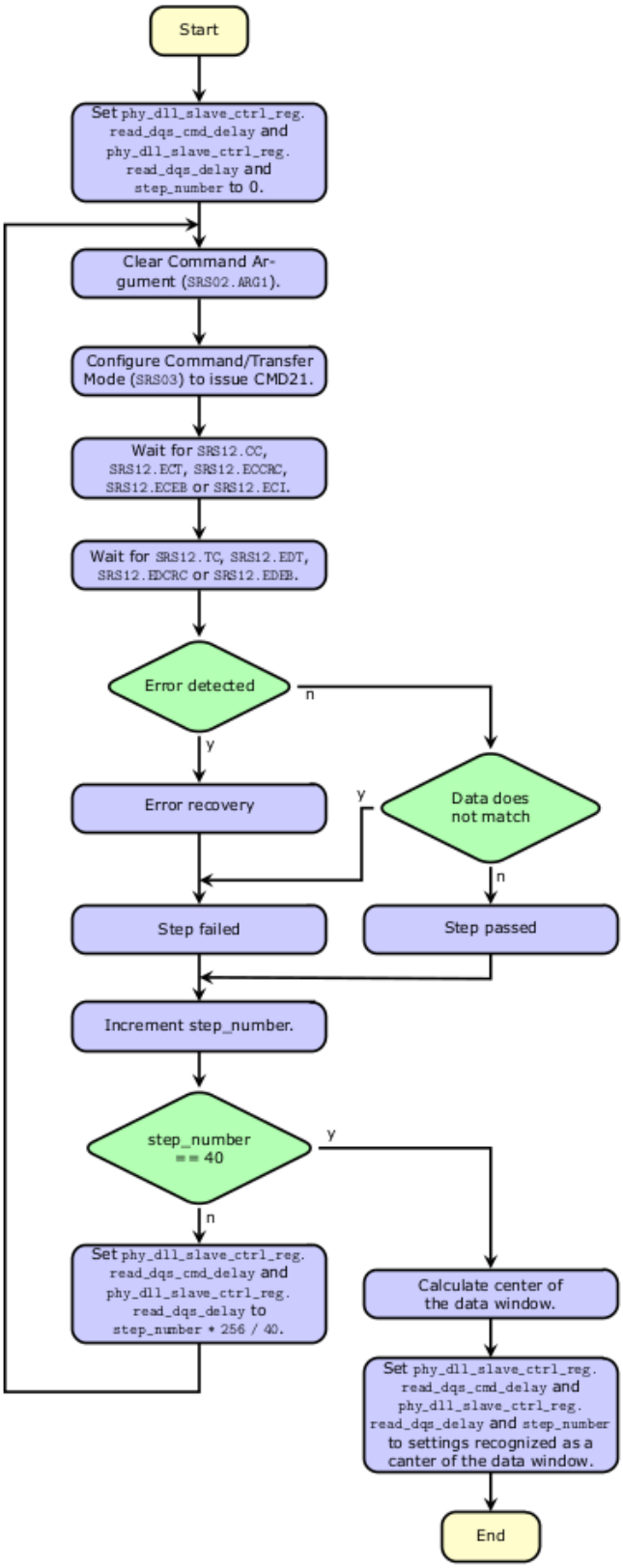


Рисунок 1.127 – Последовательность настройки для eMMC

1.3.9.10.3.3.4 Обработка ошибок

1.3.9.10.3.3.4.1 Устранение ошибок обобщенных операций

На рисунке 1.128 приведен пример процедуры устранения ошибок обобщенных операций для обработки ошибок, обнаруженных хост-контроллером SD/eMMC. После обнаружения ошибки и сообщения о ней в любом регистре состояния прерывания программа идентифицирует, возможно ли устранить ошибку или нет. Если ошибку можно устранить, программа выполняет процедуру с программным сбросом для логики CMD и/или DAT. Основной целью этой процедуры являются сброс логики, на которую повлияла ошибка, и подготовка к следующей транзакции (например, для возобновления прерванной операции чтения/записи).

Эта процедура скорее всего не поможет, если ошибку нельзя устранить. Для неустранимой ошибки обычно необходим полный программный/аппаратный сброс и цикл включения устройства.

Ту же процедуру можно использовать для устройств SD, SDIO и eMMC.

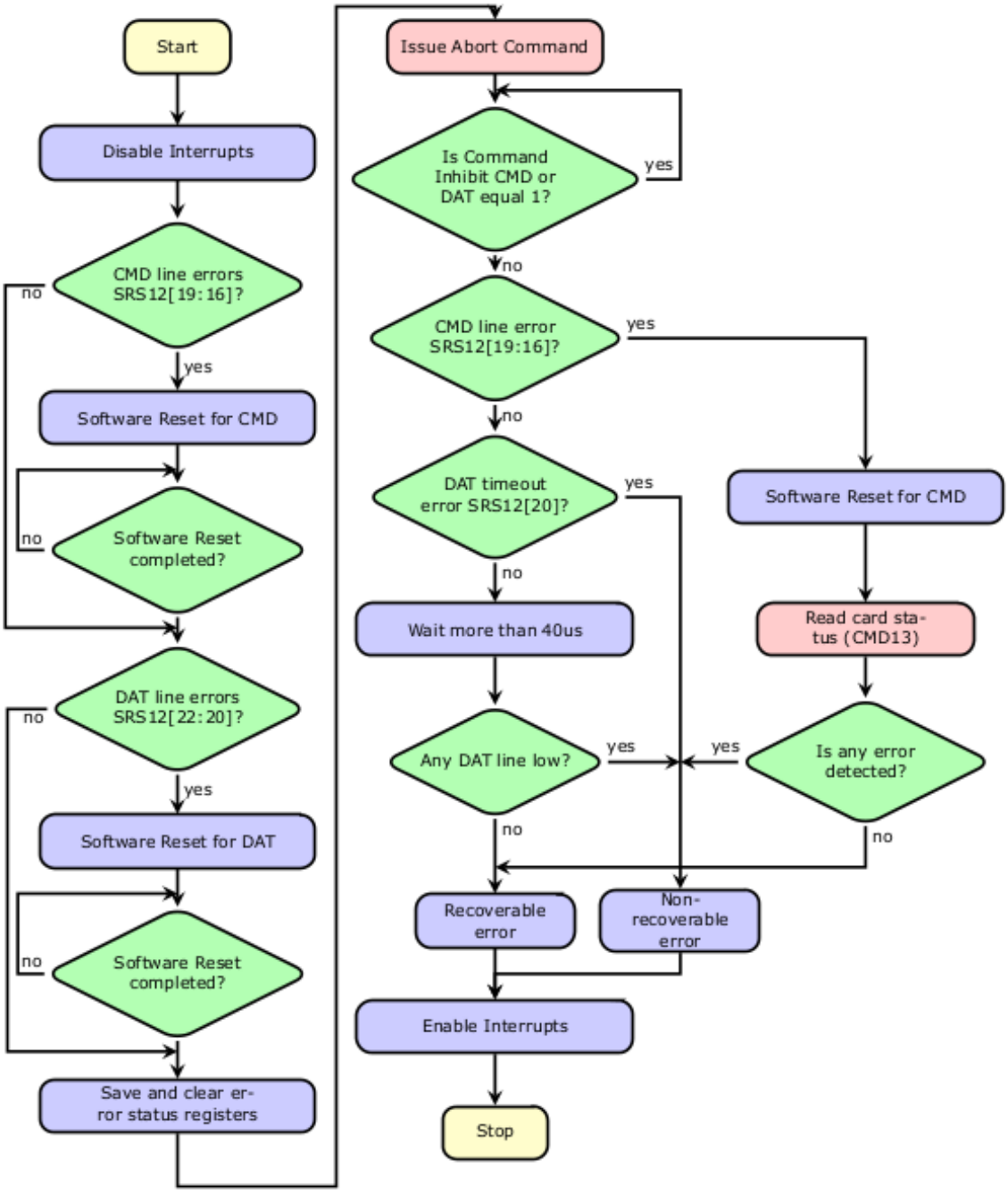


Рисунок 1.128 – Процедура устранения ошибок обобщенных операций

1.3.9.10.3.3.4.2 Устранение ошибки организации очереди команд

На рисунке 1.129 приведен пример процедуры устранения ошибки организации очереди команд, необходимой для перевода хост-контроллера и устройства и состояния ошибки и подготовки к следующим операциям.

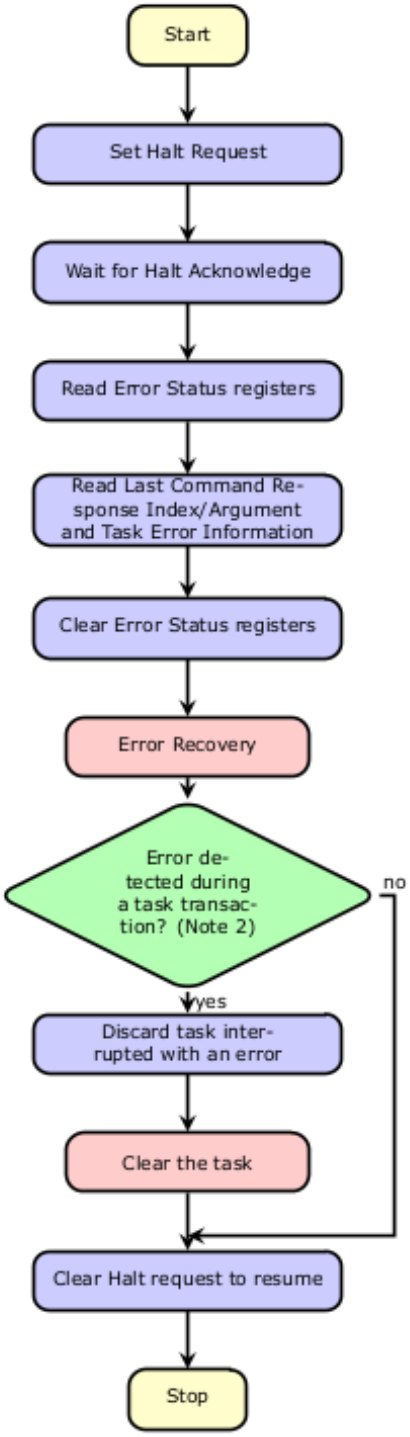


Рисунок 1.129 – Процедура устранения ошибки организации очереди команд

Примечание: Хост-контроллер SD всегда отвечает на запрос остановки, подтверждая остановку. Программа дожидается подтверждения перед запуском процедуры устранения ошибок обобщенных операций. Сигнал подтверждения приходит после истечения неопределенного времени (времени, необходимого для завершения ожидающей операции). Подтверждение

устанавливается, когда команда и тракт передачи данных сообщают свои состояния (была ли операция выполнена успешна, или она была прервана из-за ошибки).

Примечание 2: отмена / сброс задачи необходим в случае ее прерывания из-за ошибки.

Функция организации очереди команд предусматривает механизмы защиты, встроенные в хост-контроллер SD. Она также предусматривает автоматическое обнаружение ошибок для ответа устройства R1.

Ниже представлен полный список ошибок, которые могут быть обнаружены функцией организации очереди команд:

– Ошибка ответа R1

– Ошибки команд (индекс/CRC/конечный бит/тайм-аут)

– Ошибки передачи данных (CRC/конечный бит/тайм-аут)

– Ошибка DMA

– Ошибка интерфейса ведущего устройства

После обнаружения любой из указанных ошибок внутренняя логика может остановить работу, сообщить о типе ошибки и дождаться запуска процедуры устранения ошибок организации очереди команд.

Функция организации очереди команд может отправить CMD13 SQS (отправка состояние очереди) устройству во время транзакции данных. Если эта функция включена (т. е. CQRS16.CQSSCBC > 0), и выполняется транзакция данных, контроллер передает CMD13 SQS параллельно с транзакцией данных и сообщает о возможной ошибке CMD13 SQS в регистры состояний ошибок команд (SRS15[4:1]).

Хост-контроллер SD/eMMC может сообщить об ошибке текущего предела, но эта ошибка не прерывает очередь команд и транзакцию. Эта ошибка зависит от приложения и должна быть классифицирована как неустранимая ошибка. Реакция программного обеспечения должна быть соответствующей.

Хост-контроллер SD/eMMC не использует механизм проверки ошибок ответов, если включена организация очереди команд.

Программа может либо опросить регистры состояний, либо полагаться на прерывания, сгенерированные хост-контроллером. Если ошибка может привести к прерыванию системы, программа включает требуемые регистры состояний прерываний перед передачей задачи. Генерирование прерывания настраивается с помощью следующих регистров:

– Включить состояние ошибки (SRS13[31:16])

– Включить сигнал ошибки (SRS14[31:16])

– Включить состояние обнаружения ошибки ответа (CQRS05[2])

– Включить сигнал обнаружения ошибки ответа (CQRS06[2])

Таблица 1.432 – Варианты ошибок

Работа	**Событие (любое событие)**	**Последствия**
Чтение дескриптора задачи перед CMD44 (организация последовательности задач)	Ошибка ведущего устройства	Блокировка и передача идентификатора задачи в режиме ошибки ответа
Чтение дескриптора задачи перед CMD46 / CMD47 (выполнение задачи)	Ошибка ведущего устройства	Блокировка и передача идентификатора задачи в режиме ошибки передачи данных
Чтение дескриптора задачи перед DCMD (выполнение задачи)	Ошибка ведущего устройства	Блокировка и передача идентификатора задачи в режиме ошибки ответа
CMD44	Ошибка команды	Блокировка и передача идентификатора задачи в режиме ошибки ответа
CMD45	Ошибка команды	Блокировка и передача идентификатора задачи в режиме ошибки ответа
CMD13, пока контроллер остановлен	Ошибка команды	Блокировка, идентификатор задачи не передается
CMD13 во время транзакции CQ	Ошибка команды	Блокировка, идентификатор задачи не передается
CMD46 / CMD47	Ошибка команды Ошибка данных Ошибка DMA	Блокировка и передача идентификатора в режиме ошибки передачи данных
DCMD	Ошибка команды	Блокировка и передача идентификатора в режиме ошибки ответа
DCMD	Ошибка тайм-аута данных	Блокировка и передача идентификатора в режиме ошибки ответа

Ошибка передачи данных в информации об ошибке задачи:

– установка на 1, если ошибка возникает во время CMD46 или CMD47

– сброс на 0, если ошибка возникает во время CMD44, CMD45 или DCMD

Ошибка режима ответа в информации об ошибке задачи:

– установка на 1, если ошибка возникает во время CMD44, CMD45 или DCMD

– сброс на 0, если ошибка возникает во время CMD46 или CMD47

Информация об ошибке задачи не меняется, когда CQ отключается или блокируется.

1.3.9.10.3.3.4.3 Устранение ошибок неполного завершения/переполнения Combo PHY

Флаги неполного завершения или переполнения Combo PHY указывают нарушение PHY FIFO. После сообщения об ошибке программа должна выполнить действия:

– Задание программного сброса для CMD и сброса программы для DAT.

– Установка PHY_SW_RESET (dll_rst_n) на 0 (сброс активации)

– Установка PHY_SW_RESET (dll_rst_n) на 1 (сброс деактивации)

1.3.9.10.3.3.5 Последовательность команд

1.3.9.10.3.3.5.1 Описание

Организация очереди команд (CQ) - функция, доступная в стандарте eMMC с версии 5.1. Она позволяет подать несколько (до 32) команд, установив биты сигнального регистра задач. Если во время передачи не появляются ошибки, хост может читать конфигурацию команд в списке дескрипторов задач с помощью внутреннего DMA и рабочих операций.

Движок CQ запускает все операции последовательно для передачи задач. Если хост готов к выполнению задач, но следующая в очереди задача не готова, устройство принимает первую готовую задачу для минимизации времени простоя.

При выполнении задач учитывается приоритет задач, заданный во время записи сигнального регистра задач, и готовности устройства к выполнению задачи.

Функция организации очереди команд совместима со стандартом eMMC. Подробную информацию см. в Стандарте eMMC5.1, подраздел «Приложение B. Интерфейс хост-контроллера для организации очереди команд».

1.3.9.10.3.3.5.2 Дополнительные режимы работы

– DCMD выполняется, только когда не выполняются другие задачи, даже если стандарт позволяет отправить DCMD во время передачи данных, когда установлено поле синхронизации CMD.

1.3.9.10.3.3.5.3 Порядок выполнения задач

Движок организации очереди команд обрабатывает и выполняет задачи в порядке их подачи. Порядок задач определяется двумя правилами:

– Если в сигнальном регистре задач задано несколько старших битов, наибольший приоритет имеет самый младший бит.

– В случае нескольких записей в сигнальном регистре задач задача, переданная ранее, имеет больший приоритет.

1.3.9.10.3.3.6 Механизм дескриптора автоматического конфигурирования

1.3.9.10.3.3.6.1 Описание

Дескриптор автоматического конфигурирования выполняет последовательность предварительной инициализации автоматически без влияния программного обеспечения. Для этого механизма нужны два списка дескрипторов в памяти системы. Первый список содержит дескрипторы связи для каждого режима скорости. Второй список содержит параметры для предварительной инициализации, заданной дескриптором связи.

Регистры HRS42.DESCMECH_EN и HRS40 (с регистром HRS41, если используется 64-битовый адрес) должны быть установлены на включение дескриптора автоматического конфигурирования. В случае обнаружения изменения режима скорости механизм блокирует возможность записи программой в регистры ядра и начинает чтение дескрипторов с движком ADMA. Устанавливается высокий сигнал desc_mech_active, что означает выполнение автоматического конфигурирования. Регистры хоста и физического уровня автоматически обновляются после задания новых значений.

В случае возникновения ошибки во время автоматического конфигурирования (устанавливается высокий сигнал desc_mech_error) механизм останавливает выполнение операции, и в поле регистра HRS43.ERROR_VAL будет установлен тип ошибки. Механизм повторного включения (последовательность включения-выключения) в регистре HRS40 необходим для устранения ошибок, очистки регистра HRS43 и повторной активации дескриптора автоматического конфигурирования.

Списки дескрипторов в памяти системы указаны на рисунке 1.130

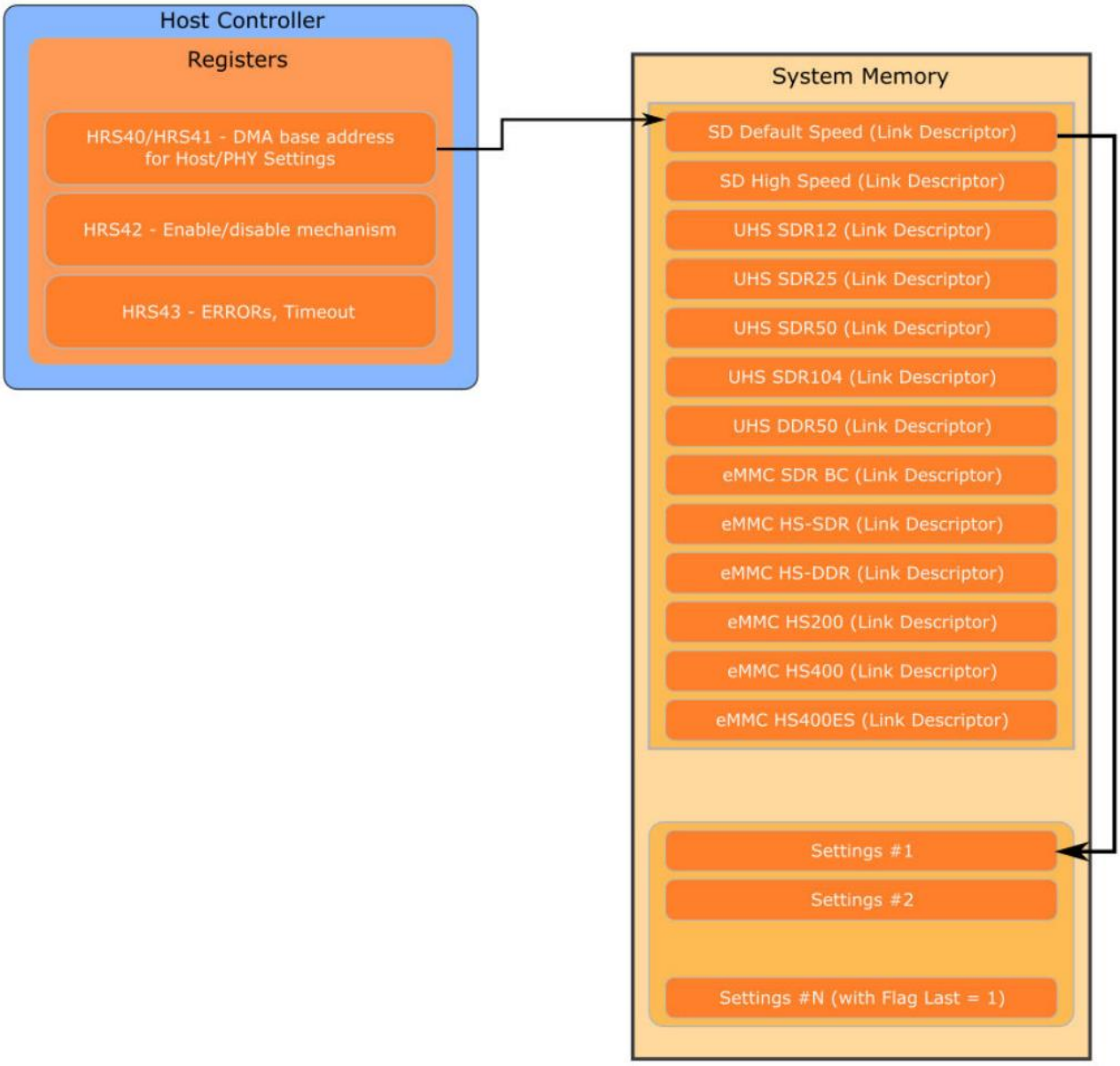


Рисунок 1.130 – Списки дескрипторов в памяти системы

1.3.9.10.3.3.6.2 Содержимое дескрипторов

Длина дескрипторов, составляемых для механизма дескриптора автоматического конфигурирования, всегда составляет 128 битов. Другая длина запрещена. Дескрипторы связи содержат поля, описанные в пп Таблица дескрипторов ADMA2. В поле ACT задается LINK, в поле LENGTH задается «0», в поле адреса ADDRESS указывается адрес списка дескрипторов с параметрами.

Дескриптор с параметрами содержит поля, указанные на рисунке 1.131.

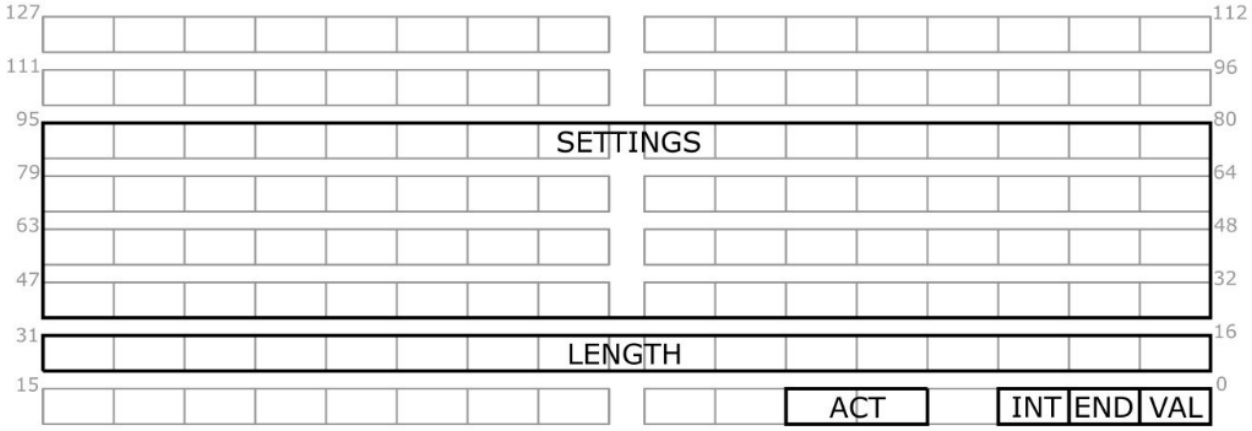


Figure 24. Descriptor with settings Layout

Рисунок 1.131 – Схема дескриптора с параметрами

Поля VAL, END и INT имеют назначение, указанное в пп Таблица дескрипторов ADMA2. Дескриптор не указывает какие-либо данные в памяти, поэтому в поле LENGTH задается «0». Адрес для следующего дескриптора с параметрами автоматически увеличивается.

В поле ACT в дескрипторах с параметрами можно задать любое значение. Поле «SETTINGS» (ПАРАМЕТРЫ) дескриптора содержит:

– тип команды (CMD)

– адрес регистра (REG_ADDRESS)

– новое/расчетное значение регистра (EX_VALUE/NEW_VALUE)

– маску/диапазон битов (MASK)

Описание полей «SETTINGS» (ПАРАМЕТРЫ) дескриптора приведено в таблице 1.433.

Таблица 1.433 – Поле «SETTINGS» дескриптора

Имя поля	**Позиция бита**	**Ширина**	**Описание**
CMD	[63:62]	2	Типы команд: 00b - WRITE 01b - READ_BLOCK 10b - READ_NONBLOCK 11b - WAIT
MASK	[61:52]	10	Диапазон битов расчетного значения чтения MASK [4:0] - нижний предел диапазона битов MASK [9:5] - верхний предел диапазона битов Одно и то же значение для нижнего и верхнего пределов диапазона битов выбирает один из 32 битов в расчетном значении чтения.
REG_ADDRESS	[47:32]	16	Адрес регистра.

Имя поля	Позиция бита	Ширина	Описание
EX_VALUE/NEW_VALUE	[31:0]	32	Для команды записи это новое значение, записываемое в заданный регистр. Для команд чтения это значение является расчетным.

1.3.9.10.3.3.6.3 Список дескрипторов с примером последовательности предварительной инициализации

Пример списка дескрипторов с набором требуемых регистров представлен в таблице 1.434.

Таблица 1.434 – Пример списка дескрипторов с набором требуемых регистров

№	Название регистра	Адрес	Поле	Тип CMD	Примечание
1	HRS09	0x0024	PHY_SW_RESET	WRITE	
2	HRS09	0x0024	RDDATA_EN, RDCMD_EN, EXTENDED_WR_MODE, EXTENDED_RD_MODE	WRITE	Устанавливается PHY_SW_RESET с предыдущим значением. Этот дескриптор нельзя объединить с предыдущим дескриптором.
3a	HRS04	0x0010	0x2004	WRITE	Адрес регистра DQSTM
3b	HRS05	0x0014	USE_EXT_LPBK_DQS, USE_LPBK_DQS, USE_PHONY_DQS, USE_PHONY_DQS_CMD	WRITE	Требуемые поля регистра DQSTM
4a	HRS04	0x0010	0x2000	WRITE	Адрес регистра DQTM
4b	HRS05	0x0014	IO_MASK_ALWAYS_ON, IO_MASK_END, IO_MASK_START, DATA_SELECT_OE_END	WRITE	Требуемые поля регистра DQTM
5a	HRS04	0x0010	0x2008	WRITE	Адрес регистра GLPBK
5b	HRS05	0x0014	SYNC_METHOD, SW_HALF_CYCLE_SHIFT, RD_DEL_SEL, GATE_CFG_ALWAYS_ON	WRITE	Требуемые поля регистра GLPBK
6a	HRS04	0x0010	0x200C	WRITE	Адрес регистра DLLMC
6b	HRS05	0x0014	DLL_BYPASS_MODE	WRITE	Требуемое поле регистра DLLMC
7a	HRS04	0x0010	0x2010	WRITE	Адрес регистра DLLSC
7b	HRS05	0x0014	CLK_WR_DELAY	WRITE	Требуемые поля регистра DLLSC
8	HRS09	0x0024	PHY_SW_RESET	WRITE	
9	HRS09	0x0024	PHY_INIT_COMPLETE	READ_BLOCK	

1.3.9.10.3.3.7 Сигналы низкого/высокого напряжения

До Стандарта физического уровня SD версии 6.0 интерфейс SD работает с сигналами 3,3 В (в обязательном режиме без UHS) и 1,8 В (в дополнительном режиме UHS-I). Хост-контроллер, поддерживающий сигналы 3,3 В, определяется Стандартом физического уровня SD версии

6.0 и более поздней версии как хост-контроллер SD с сигналами высокого напряжения (ВН).

Стандарт физического уровня SD версии 6.0 предусматривает вариант, в котором не поддерживаются сигналы 3,3 В, и требуются только сигналы 1,8 В. Хост-контроллер, поддерживающий только сигналы 1,8 В, определяется стандартом как хост-контроллер SD с сигналами низкого напряжения (СНН).

Хост-контроллер имеет регистр хоста с сигналами низкого напряжения (SRS17.LVSH), который информирует о сигналах LVS (1) или HVS (0). Значение регистра по умолчанию - 1. При интеграции это поле может потребоваться обновить с учетом требований.

1.3.9.10.3.3.7.1 Сигналы низкого напряжения

Хост-контроллер с сигналами низкого напряжения поддерживает платы LVS и не поддерживает карты HVS. Все платы на базе стандарта физического уровня SD версии 6.0 и все платы на базе физического уровня SD версии 6.0 и выше, но без маркировки LV (низкое напряжение) или A2 (класс производительности 2) не поддерживаются.

Если отсутствие совместимости с предыдущими версиями не является проблемой, SDA (организация по стандартизации) выпускает LVS для упрощения требований к входным и выходным контактным столбикам.

Для LVS требуется последовательность инициализации для включения SD-карты в режиме LVS. Последовательность LVS представлена на рисунке 1.132.

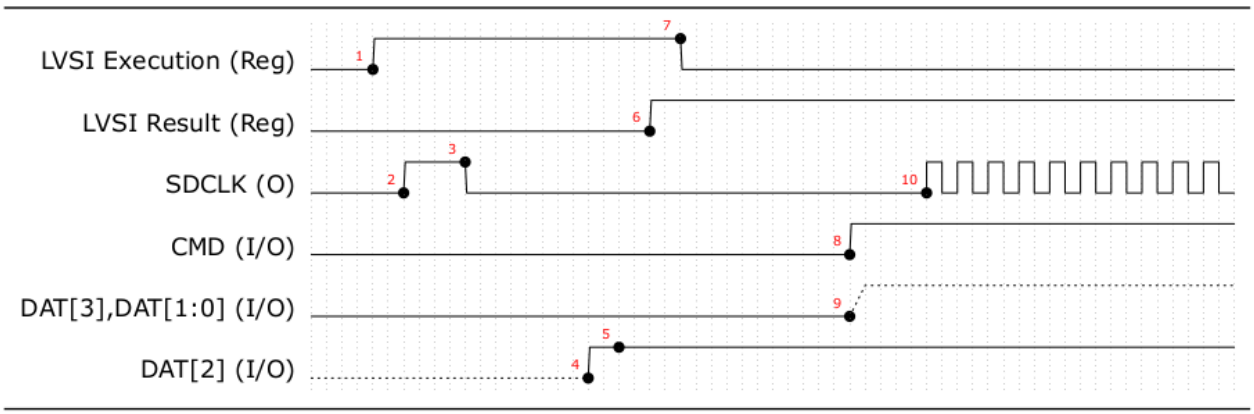


Рисунок 1.132 – Последовательность LVS

Последовательность LVS состоит из следующих этапов:

1) Программа подтверждает, что регистр хоста с сигналами низкого напряжения (SRS17.LVSH) высокий

2) Программное обеспечение устанавливает HRS09.LVSI_CNT и HRS09.LVSI_TCKSEL, рассчитывающие импульс SDCLK с минимум 15 мсек

3) Программное обеспечение включает внутренний тактовый генератор (SRS11.ICE) и ожидает стабилизации внутреннего тактового генератора (SRS11.ICS).

4) Программа включает выполнение LVSI (SRS15.LVSIEXEC) (1)

5) Хост устанавливает низкие CMD, DAT[3], DAT[1:0] и понижает DAT[2]

6) Хост-контроллер генерирует один тактовый импульс шириной, указанной в HRS09 (2-3)

7) Программное обеспечение ожидает в течение минимум 5010 мкм (3-5)

8) Карта устанавливает высокий DAT[2] через 5 мсек после окончания спадающего фронта тактового сигнала (3-4)

9) Программное обеспечение проверяет регистр результатов LVSI (SRS09.LVSIRSLT) (6)

10) Если результат LVSI = 1 (успешно)

a) Хост повышает линии CMD, DAT (8,9)

b) Программа включает SDCLK (10)

11) Если результат LVSI = 0 (не выполнено)

a) Программа отключает питание карты

1.3.9.10.3.4 Программный сброс

Хост-контроллер SD/eMMC можно сбросить с помощью программного обеспечения несколькими способами в зависимости от выбранного режима. Программный сброс можно запустить в любое время, установив один из следующих битов:

– HRS00.SWR (программный сброс) сбрасывает всю внутреннюю логику. Этот сброс общий для хост-контроллера SD и работает аналогично аппаратному сбросу. Подробную информацию см. в описании к регистру HRS00.SWR.

– SRS11.SRFA (программный сброс для всех) сбрасывает всю внутреннюю логику заданного слота. Подробную информацию см. в описании к регистру SRS11.SRFA.

– SRS11.SRCMD (программный сброс для CMD) сбрасывает часть, ответственную за генерирование команд и проверку ответов для заданного слота. Подробную информацию см. в описании к регистру SRS11.SRCMD.

– SRS11.SRDAT (программный сброс для DAT) сбрасывает часть, ответственную за передачи данных для указанного слота. Подробную информацию см. в описании к регистру SRS11.SRDAT.

1.3.9.10.3.5 Регистры

Регистры представлены в таблицах 1.435 - 1.513.

1.3.9.10.3.5.1 HRS00 (0x0000)

HRS00 - общий информационный регистр.

Таблица 1.435 – Поля регистра HRS00

Бит	Тип	Описание	Сброс
23-16	R	**SAV - доступный слот** Поле указывает, что контроллер поддерживает один слот.	8’h1
0	W1S	**SWR - программный сброс** Если установлен на 1, выполняется сброс всего ядра. После завершения сброса бит SWR автоматически очищается. Для выполнения запрошенного сброса потребуется некоторое время, поэтому программное обеспечение должно всегда запрашивать состояние бита SWR и выполнять другие операции, только если бит SWR сброшен на 0. Нет разницы между программным сбросом SWR и SRS11.SRFA. Оба варианта сброса имеют одинаковые триггеры.	1’h0

1.3.9.10.3.5.2 HRS01 (0x0004)

HRS01 - регистр настройки устранения дребезга.

Таблица 1.436 – Поля регистра HRS01

Бит	Тип	Описание	Сброс
23-0	Чтение/запись	**DP - период устранения дребезга** Определяет число тактовых циклов системы (clk), используемых логикой устранения дребезга, которая обнаруживает вставку и извлечение карты. Период устранения дребезга равен DP * tclk, где tclk - период тактового цикла clk. Если уровень сигнала pad_mem_ctrl_0 не меняется в течение заданного периода устранения дребезга, логика ядра декодирует состояние карты как стабильное и активирует событие card_inserted или card_removed. Обычно значение DP должно составлять примерно 20 мсек. Этот регистр сбрасывается на DEBOUNCE_PERIOD.	24'h32

1.3.9.10.3.5.3 HRS02 (0x0008)

HRS02 - регистр настройки шины.

Таблица 1.437 – Поля регистра HRS02

Бит	Тип	Описание	Сброс
17-16	Чтение/запись	**OTN - число ожидающих передач** Указывает число ожидающих передач по интерфейсу DMA (ведущего устройства). Число ожидающих передач составляет (OTN + 1), где OTN может быть задан в диапазоне от 0 до 3. Этот регистр устанавливается на 3 после сброса (т. е. 4 ожидающие передачи).	2'h3
3-0	Чтение/запись	**PBL - настраиваемая длина пакета** Это поле определяет максимальное число порций в пакете DMA. Значение можно изменить при отсутствии активной передачи. Этот регистр устанавливается на 0 после сброса. – 0001b - 1 порция в пакете – 0010b - 2 порции в пакете – 0011b - 4 порции в пакете – 0100b - 8 битов в пакете – другое - 16 порций в пакете	4'h0

1.3.9.10.3.5.4 HRS04 (0x0010)

Таблица 1.438 – Поля регистра HRS04

Бит	Тип	Описание	Сброс
15-0	Чтение/запись	**PHYREGADDR - адрес регистра PHY** Это поле определяет адрес регистра PHY для получения доступа чтения/записи к регистру PHY через HRS05.	16'h0

1.3.9.10.3.5.5 HRS05 (0x0014)

Таблица 1.439 – Поля регистра HRS05

Бит	Тип	Описание	Сброс
31-0	Чтение/запись	**PHYREGDATA - регистр PHY порта данных** Доступ к этому регистру генерирует транзакцию чтения или записи в регистр Combo PHY. Если это поле предназначено для чтения, транзакция чтения передается в PHY через интерфейс APB. Значение, полученное при транзакции, передается обратно в качестве результата доступа. Если это поле установлено на запись, PHY передается транзакция записи через интерфейс APB вместе с записанными данными. Интерфейс APB контроллера - Combo PHY не поддерживает невыровненные передачи, поэтому доступ к этому полю ограничен только 32-битовыми транзакциями. Невыровненный (16 бит или 8 бит) доступ к этому полю может привести к непредвиденным результатам.	32'h0

1.3.9.10.3.5.6 HRS06 (0x0018)

HRS06 - регистры управления eMMC.

Таблица 1.440 – Поля регистра HRS06

Бит	Тип	Описание	Сброс
2-0	Чтение/запись	**EMM - выборе режима eMMC** Это поле задает режим eMMC. Режим должен задаваться с учетом параметров устройства eMMC. Если используется SD-карта, это поле должно быть 000b. – 000b - SD-карта используется – 010b - SDR – 011b - DDR – 100b - HS200 – 101b - HS400 – 110b - усиленный стробирующий импульс HS400 – другое - устаревшее	3'h0

1.3.9.10.3.5.7 HRS07 (0x001c)

HRS07 - регистр информации о задержке ввода-вывода.

Таблица 1.441 – Поля регистра HRS07

Бит	Тип	Описание	Сброс
20-16	Чтение/запись	**RW_COMPENSATE - значение компенсации ожидания чтения** С учетом задержек между PAD и dfi_rddata, dfi_wrdata и PAD, а также задержек, связанными с требованиями к синхронизации ожидания чтения, сигнал dat[2] должен быть установлен на 0 раньше, чем контроллер прочтет конечный бит данных чтения. Проектировщик должен обновить этот регистр, указав задержку числа трактов данных в тактовых циклах sdmclk. Если значение превышает 10, и значение поля выбора частоты SDCLK (последовательного включения SRS11.SDCFSH, SRS11.SDCFSL) равно 0, то параметр io_mask_start в регистре физического уровня phy_dq_timing_reg должен быть установлен на значение, равное (RW_COMPENSATE-10)*2.	5'h0

Бит	Тип	Описание	Сброс
4-0	Чтение/запись	**IDELAY_VAL - задержка входного сигнала для ввода-вывода.** Проектировщик должен изменить в этом регистре задержку ввода-вывода на соответствующее значение входного сигнала. Задержка составляет половину периода sdmclk. Если sdmclk работает на частоте 200 МГц, то 1 равно 2,5 нсек. Это значение будет использоваться для компенсации задержки линии DAT, когда контроллер читает прерывание карты.	5'h0

1.3.9.10.3.5.8 HRS08 (0x0020)

HRS08 - регистр управления обновлением и состояния DLL физического уровня.

Таблица 1.442 – Поля регистра HRS0B

Бит	Тип	Описание	Сброс
1	R	**PHY_DLL_UPDACK** Этот регистр содержит значение, считанное с порта sdphy_dfi_ctrlupd_ack.	1'h0
0	Чтение/запись	**PHY_DLL_UPDREQ** Этот регистр управляет портом sdphy_dfi_ctrlupd_req.	1'h0

1.3.9.10.3.5.9 HRS09 (0x0024)

HRS09 - регистр управления и состояния физического уровня.

Таблица 1.443 – Поля регистра HRS09

Бит	Тип	Описание	Сброс
31-28	Чтение/запись	**LVSI_CNT** Это поле определяет период импульса SDCLK во время идентификации LVS. Период изменяется в диапазоне от LVSI_CNT * 2л(LVSI_TCKSEL+1) * t_CLK до (LVSI_CNT + 1) * 2л(LVSI_TCKSEL+1) * t_CLK. Рекомендуется использовать меньшее значение LVSI_TCKSEL и большее значение LVSI_CNT для уменьшения изменения периода импульса SDCLK. Изменение - +1/-1 LVSI_TCLKSEL.	4'hf
27-22	Чтение/запись	**LVSI_TCKSEL** Это поле определяет величину LVSI_CNT. – 0 - 2*t_CLK (два тактовых цикла CLK) – 1 - 4*t_CLK (четыре тактовых цикла CLK) – 2 - 8*t_CLK (восемь тактовых циклов CLK) – 3 - 16*t_CLK (шестнадцать тактовых циклов) – N - (2л(N+1))*t_CLK где N должен быть равен от 0 до 47. Значения выше 47 являются резервными.	6'h7
16	Чтение/запись	**RDDATA_EN** Если 1, dfi_rddata_en принудительно устанавливается на 1, иначе логика контроллер управляет сигналом.	1'h0
15	Чтение/запись	**RDCMD_EN** Если 1, dfi_rdcmd_en принудительно устанавливается на 1, иначе логика контроллер управляет сигналом.	1'h0

Бит	Тип	Описание	Сброс
3	Чтение/запись	**EXTENDED_WR_MODE** Управляет портом sdphy_param_extended_wr_mode. При непрограммном сбросе этот регистр очищается.	1'h0
2	Чтение/запись	**EXTENDED_RD_MODE** Управляет портом sdphy_param_extended_rd_mode. При непрограммном сбросе этот регистр очищается.	1'h0
1	R	**PHY_INIT_COMPLETE** Это поле содержит значение, считываемое с порта sdphy_dfi_init_complete.	1'h0
0	Чтение/запись	**PHY_SW_RESET** Это поле управляет sdphy_dll_rst_n.	1'h1

1.3.9.10.3.5.10HRS10 (0x0028)

HRS10 - регулировка точки запуска SDCLK контроллера.

Таблица 1.444 – Поля регистра HRS10

Бит	Тип	Описание	Сброс
22	Чтение/запись	**RDDATA_SWAP** Если 1, байты dfi_rddata [7:0] и [15:8] меняются местами.	1'h0
19-16	Чтение/запись	**HCSDCLKADJ** Это поле позволяет регулировать механизм управления потоком, который отключает SDCLK. Если значение равно 0, источник тактирования (dfi_webar/dfi_webar_high) будет отключен сразу после конечного бита блока данных. Увеличение этого значение приведет к отключению тактового сигнала раньше на этапе периода SDCLK.	4'h0

*

*

1.3.9.10.3.5.11HRS11 (0x002c)

HRS11 - управление eMMC.

Таблица 1.445 – Поля регистра HRS11

Бит	Тип	Описание	Сброс
0	Чтение/запись	**EMMC_RST** Это поле управляет dfi_rstbar, которое используется для сброса eMMC.	1'h0

1.3.9.10.3.5.12HRS12 (0x0030)

HRS12 - состояние прерывания хоста.

Таблица 1.446 – Поля регистра HRS12

Бит	Тип	Описание	Сброс
3	W1C	**Переполнение PHY DAT - состояние, получаемое от Combo PHY, указывающее состояние DAT FIFO.**	1'h0

Бит	Тип	Описание	Сброс
2	W1C	**Неполное завершение PHY DAT - состояние, получаемое от Combo PHY, указывающее состояние DAT FIFO.**	1'h0
1	W1C	**Переполнение PHY CMD - состояние, получаемое от Combo PHY, указывающее состояние CMD FIFO.**	1'h0
0	W1C	**Неполное завершение PHY CMD - состояние, получаемое от Combo PHY, указывающее состояние CMD FIFO.**	1'h0

1.3.9.10.3.5.13HRS13 (0x0034)

HRS13 - включение состояния хоста.

Таблица 1.447 – Поля регистра HRS13

Бит	Тип	Описание	Сброс
3	Чтение/запись	**Маска для состояния PHYDATOF** – 1 - включено – 0 - маскировано	1'h0
2	Чтение/запись	**Маска для состояния PHYDATUR** – 1 - включено – 0 - маскировано	1'h0
1	Чтение/запись	**Маска для состояния PHYCMDOF** – 1 - включено – 0 - маскировано	1'h0
0	Чтение/запись	**Маска для состояния PHYCMDUR** – 1 - включено – 0 - маскировано	1'h0

1.3.9.10.3.5.14HRS14 (0x0038)

HRS14 - включение сигнала хоста.

Таблица 1.448 – Поля регистра HRS14

Бит	Тип	Описание	Сброс
3	Чтение/запись	**Включение прерывания для состояния PHYDATOF** – 1 - включено – 0 - маскировано	1'h0
2	Чтение/запись	**Включение прерывания для состояния PHYDATUR** – 1 - включено – 0 - маскировано	1'h0
1	Чтение/запись	**Включение прерывания для состояния PHYCMDOF** – 1 - включено – 0 - маскировано	1'h0
0	Чтение/запись	**Включение прерывания для состояния PHYCMDOF** – 1 - включено – 0 - маскировано	1'h0

1.3.9.10.3.5.15HRS16 (0x0040)

HRS16 - задержка выхода CMD/DAT.

Таблица 1.449 – Поля регистра HRS16

Бит	Тип	Описание	Сброс
31-28	Чтение/запись	**WRDATA1_SDCLK_DLY** Значение в этом поле определяет задержку сигнала dfi_wrdata[15:8]. Задержка является значением, равным – t_SDCLK/2.	4'h0
27-24	Чтение/запись	**WRDATA0_SDCLK_DLY** Значение в этом поле определяет задержку сигнала dfi_wrdata[7:0]. Задержка является значением, равным – t_SDCLK/2.	4'h0
23-20	Чтение/запись	**WRCMD1_SDCLK_DLY** Значение в этом поле определяет задержку сигнала dfi_wrcmd. Задержка является значением, равным * t_SDCLK/2.	4'h0
19-16	Чтение/запись	**WRCMD0_SDCLK_DLY** Значение в этом поле определяет задержку сигнала dfi_wrcmd. Задержка является значением, равным * t_SDCLK/2.	4'h0
15-12	Чтение/запись	**WRDATA1_DLY** Значение в этом поле определяет задержку сигнала dfi_wrdata[15:8]. Задержка является значением, равным – t_SDMCLK.	4'h0
11-8	Чтение/запись	**WRDATA0_DLY** Значение в этом поле определяет задержку сигнала dfi_wrdata[7:0]. Задержка является значением, равным – t_SDMCLK.	4'h0
7-4	Чтение/запись	**WRCMD1_DLY** Значение в этом поле определяет задержку сигнала dfi_wrcmd[1]. Задержка является значением, равным * t_SDMCLK.	4'h0
3-0	Чтение/запись	**WRCMD0_DLY** Значение в этом поле определяет задержку сигнала dfi_wrcmd[0]. Задержка является значением, равным * t_SDMCLK.	4'h0

1.3.9.10.3.5.16HRS29 (0x0074)

HRS29 - системный код SD.

«Номер изделия» - идентификационный номер, относящийся к правой части (LSB).

Таблица 1.450 – Поля регистра HRS29

Бит	Тип	Описание	Сброс
31-0	R	**Номер изделия**	32'hxxxx

1.3.9.10.3.5.17HRS30 (0x0078)

HRS30 - регистр производительности хоста.

Этот регистр указывает наличие настраиваемых опций в конфигурации контроллера SD/eMMC. Этот регистр дает информацию о функциях, которые не поддерживают стандартные регистры производительности (SRS16-SRS18).

Таблица 1.451 – Поля регистра HRS30

Бит	Тип	Описание	Сброс
1	R	**Поддержка усиленного стробирующего импульса скорости 400** В этом поле указывается поддержка усиленного стробирующего импульса HS400 (1) или (0).	1'h1
0	R	**Поддержка организации очереди команд** В этом поле указывается поддержка организации очереди команд (1) или (0).	1'h1

1.3.9.10.3.5.18 HRS31 (0x007c)

HRS31 - версия контроллера.

Этот регистр содержит номер версии контроллера.

Таблица 1.452 – Поля регистра HRS31

Бит	Тип	Описание	Сброс
27-16	R	**Версия контроллера** Номер версии контроллера	12'h602
7-0	R	**Номер исправленной версии** Номер исправления версии контроллера.	8'h2

1.3.9.10.3.5.19 HRS32 (0x0080)

HRS32 - регистр контроля FSM.

Таблица 1.453 – Поля регистра HRS32

Бит	Тип	Описание	Сброс
31	Чтение/запись	**LOAD - запрос обновление монитора FSM** При установке этого бита на 1 запускается внутренний контроль FSM для загрузки выбранного FSM. После завершения этот бит автоматически сбрасывается аппаратным обеспечением, и можно прочесть состояние FSM.	1'h0
30-16	W	**ADDR - адрес FSM** Это поле выбирает считываемое состояние FSM. Все доступные состояния машины указаны в разделе «Отладка» в руководстве пользователя.	15'h0
15-0	R	**DATA - состояние FSM** Этот регистр содержит состояние чтения FSM. Перед чтением пользователь должен выбрать адрес FSM (ADDR), установить бит LOAD и дождаться его аппаратного сброса.	16'h0

1.3.9.10.3.5.20 HRS33 (0x0084)

HRS33 - регистр состояния настройки 0.

Таблица 1.454 – Поля регистра HRS33

Бит	Тип	Описание	Сброс
31-0	R	**STAT0 - состояние настройки 0** После вызова процедуры настройки UHS-I каждый бит этого регистра представляет собой состояние одного этапа настройки. Это поле соответствует этапам настройки 31-0. – 0 - этап не выполнен – 1 - этап пройден	32’h0

1.3.9.10.3.5.21 HRS34 (0x0088)

HRS34 - регистр состояния настройки 1.

Таблица 1.455 – Поля регистра HRS34

Бит	Тип	Описание	Сброс
7-0	R	**STAT1 - состояние настройки 1** Это поле - продолжение поля STAT0. Это значение представляет собой состояние этапов настройки 39-32.	8’h0

1.3.9.10.3.5.22 HRS36 (0x0090)

HRS36 - регистр состояния загрузки.

Таблица 1.456 – Поля регистра HRS36

Бит	Тип	Описание	Сброс
6	R	**Ошибка загрузки - ошибка механизма дескриптора**	1’h0
5	R	**Ошибка загрузки - ошибка конечного бита**	1’h0
4	R	**Ошибка загрузки - ошибка CRC данных**	1’h0
3	R	**Ошибка загрузки - ошибка тайм-аута данных**	1’h0
2	R	**Ошибка загрузки - ошибка из-за недействительного подтверждения**	1’h0
1	R	**Ошибка загрузки - ошибка из-за тайм-аута подтверждения**	1’h0
0	R	**Активация загрузки** Указывает, что загрузка выполняется, и операция не может прерываться записью в любые регистры.	1’h0

1.3.9.10.3.5.23 HRS40 (0x00a0)

HRS40 - базовый адрес 0 для механизма дескриптора автоматического конфигурирования.

Таблица 1.457 – Поля регистра HRS40

Бит	Тип	Описание	Сброс
31-0	Чтение/запись	**BASE_ADDR0 - младший базовый адрес списка дескрипторов** Младшие 32 бита базового адреса для механизма дескриптора автоматического конфигурирования.	32’h0

1.3.9.10.3.5.24HRS41 (0x00a4)

HRS41 - базовый адрес 1 механизма дескриптора автоматического конфигурирования.

Таблица 1.458 – Поля регистра HRS41

Бит	Тип	Описание	Сброс
31-0	Чтение/запись	**BASE_ADDR1 - старший базовый адрес списка дескрипторов** Старшие 32 бита базового адреса для механизма дескриптора автоматического конфигурирования, если поддерживается 64-битовый адрес.	32’h0

1.3.9.10.3.5.25HRS42 (0x00a8)

HRS42 - включение/выключение механизма дескриптора автоматического конфигурирования.

Таблица 1.459 – Поля регистра HRS42

Бит	Тип	Описание	Сброс
4-1	Чтение/запись	**DESCMECH_TM - значение тайм-аута механизма дескриптора** Это значение определяет интервал, по которому определяется тайм-аут механизма дескриптора. Интервал рассчитывается следующим образом: – 1111b - clk*2л(29+2); – 1110b - clk*2л(28+2); – 1101b - clk*2л(27+2); – …; – 0001b - clk*2л(15+2); – 0000b - clk*2л(14+2). Где clk - тактовый период системы.	4’h0
0	Чтение/запись	**DESCMECH_EN - включение/выключение механизма дескриптора автоматического конфигурирования для обновления последовательности предварительной инициализации контроллера/физического уровня.**	1’h0

1.3.9.10.3.5.26HRS43 (0x00ac)

HRS43 - состояние ошибки механизма дескриптора автоматического конфигурирования.

Таблица 1.460 – Поля регистра HRS43

Бит	Тип	Описание	Сброс
3-1	R	**ERROR_VAL - тип ошибки механизма дескриптора автоматического конфигурирования** Типы ошибок: – 000b - первый дескриптор не LINK; – 001b - недействительный дескриптор (поле VAL = 0); – 010b - ошибка DMA; – 011b - тайм-аут (не выполнен вызов дескриптора или обновление регистра); – 100b - неверная маска для команды чтения (значение mask[9:5] меньше mask[4:0]); – 101b - дескриптор с командой, отличной от команды чтения READ_BLOCK, после дескриптора с командой READ_NONBLOCK.	3'h0
0	R	**Возникла ошибка во время работы механизма дескриптора автоматического конфигурирования.**	1'h0

1.3.9.10.3.5.27SRS00 (0x0200)

SRS00 - адрес системы SDMA / аргумент 2 / число 32-битовых блоков.

Таблица 1.461 – Поля регистра SRS00

Бит	Тип	Описание	Сброс
31-0	Чтение/запись	**SAAR - адрес системы SDMA / аргумент 2 / число 32-битовых блоков** Это поле используется в качестве: – регистра-счетчика 32-битовых блоков; – адреса памяти системы SDMA; – аргумента Auto CMD23. **Счетчик 32-битовых блоков** Это поле позволяет определить число блоков данных для следующей передачи. Оно используется, когда контроллер установлен в режим совместимости с версией 4 (SRS15.HV4E=1), включен счетчик блоков (SRS03.BCE=1), и счетчик 16-битовых блоков для текущей передачи сброшен (SRS01.BCCT=0). Значение этого поля уменьшается после каждой передачи блоков. Если это поле установлено на 0, передача блоков данных не выполняется. Программа не должна читать это поле во время передачи данных, так как может быть возвращено непредвиденное значение. Запись в этом поле во время передачи данных игнорируется. Когда контроллер установлен в режим совместимости с версией 4 (SRS15.HV4E=0), или счетчик 16-битовых блоков для текущей передачи не равен 0, счетчик блоков определяет (SRS01.BCCT). – 00000000h - нет передачи блоков – Передача блока(ов) 00000001h..FFFFFFFFh - 1..4294967295. **Адрес системы** Это регистр используется в качестве базового адреса при выборе движка SDMA (SRS03.DMAE=1 и SRS10.DMASEL=0) и SRS15.HV4E=0. При остановке SDMA на границе буфера SDMA программа обновляет адрес системы, при записи в SAAR[31:24] передача SDMA возобновляется. Программный драйвер устанавливает адрес для ячейки следующих данных в памяти системы. **Аргумент Auto CMD23** Auto CMD23 может использовать в не-DMA или ADMA2 режиме (когда SRS15.HV4E=0) или в не-DMA, SDMA, ADMA2 режимах (когда SRS15.HV4E=1).	32’h0

1.3.9.10.3.5.28 SRS01 (0x0204)

SRS01 - размер блока / число блоков.

Таблица 1.462 – Поля регистра SRS01

Бит	Тип	Описание	Сброс
31-16	Чтение/запись	**BCCT - число блоков для текущей передачи** В этом поле можно задать число блоков данных для следующей передачи. Этот регистр используется, если SRS03.BCE установлен на 1, SRS15.HV4E установлен на 0 или SRS15.H4VE установлен на 1, и это поле не равно 0, в противном случае регистр будет игнорироваться. Если SRS15.HV4E==1, и это поле == 0, будет выбран регистр числа 32-битовых блоков (SRS00.SAAR). Значение уменьшается после каждой передачи блоков. Если это поле установлено на 0, передача блоков данных не выполняется. Во время передачи данных операция чтения может вернуть недействительное значение, а операции записи будут игнорироваться. – 0000h - нет передачи блоков – Передача блока(ов) 0001h..FFFFh - 1..65535 block(s).	16'h0
14-12	Чтение/запись	**SDMABB - граница буфера SDMA** В этом поле можно задать границу адреса системы для движка SDMA. Передача SDMA останавливается до пересечения границы адреса и генерирует прерывание DMA (SRS12.DMAINT). После прерывания DMA, если SRS15.HV4E установлен на 0, программа должна записать новый адрес системы SDMA (SRS00.SAAR / SRS22.DMASA1) для возобновления транзакции SDMA. – 0 - граница адреса 4 кбита – 1 - граница адреса 8 кбитов – 2 - граница адреса 16 кбитов – 3 - граница адреса 32 кбита – 4 - граница адреса 64 кбита – 5 - граница адреса 128 кбитов – 6 - граница адреса 256 кбитов – 7 - граница адреса 512 кбитов	3'h0

Бит	Тип	Описание	Сброс
11-0	Чтение/запись	**TBS - размер передаваемого блока** В этом поле определяется размер блоков для передач блоков данных. Во время передачи данных операции чтения могут вернуть недействительное значение, а операции записи будут игнорироваться. Программа не установит значение, превышающее размер физически реализованного внутреннего буфера FIFO. Размер буфера равен 2лFIFODEPTH, где FIFODEPTH - настраиваемый параметр ядра. SD/MMC (память) использует блоки размером до 512 байтов. SDIO может использовать блоки размером до 2048 байтов. – 000h - не используется – 001h - 1 бит данных – 002h - 2 бита данных – 003h - 3 бита данных – … – 1FFh - 511 битов данных – 200h - 512 битов данных – … – 800h - 2048 битов данных – другое - не используется. Рекомендуется использовать размер блоков данных (512 байтов), заданный в программе, в случае передачи нескольких блоков данных (SRS03.MSBS==1). Использование блоков меньшего размера может привести к неожиданной ошибке ответа при активации управления потоком (т. е. при отключенном SDCLK) во время передачи ответа.	12’h0

1.3.9.10.3.5.29 SRS02 (0x0208)

SRS02 - аргумент 1.

Таблица 1.463 – Поля регистра SRS02

Бит	Тип	Описание	Сброс
31-0	Чтение/запись	**ARG1 - аргумент команды 1** Это поле содержит 32-битовый аргумент для команды, передаваемый файлом записи SRS03.CIDX.	32’h0

1.3.9.10.3.5.30 SRS03 (0x020c)

SRS03 - команда/режим передачи.

Таблица 1.464 – Поля регистра SRS03

Бит	Тип	Описание	Сброс
29-24	Чтение/запись	**CIDX - индекс команды** Это поле содержит номер (индекс) передаваемой команды. Индекс может быть равен 00-63, что означает, что поддерживаются все команды (CMD00-CMD63 и ACMD00-ACMD63), определенные в соответствующих спецификациях. Запись в это поле активирует передачу фактической команды. Это поле должно быть записано, только когда бит запрета команды установлен на 0 в регистре текущего состояния (SRS09.CICMD). Список доступных команд см. в соответствующей спецификации карты/устройства.	6'h0
23-22	Чтение/запись	**CT - тип команды** Это поле определяет тип команды. – Нормальная команда (0) - используется по умолчанию, когда другие типы команд не используются – Команда приостановки (01b) - не используется – Команда возобновления (10b) - не используется – Команда отмены (11b) - используется, когда программа останавливает текущую передачу данных (чтения или записи). Передача данных чтения заканчивается посредством остановки передачи во внутренний буфер. Передача данных записи завершается при переключении линии DAT в состояние High-Z. Затем после передачи команды отмены программа инициирует программный сброс. Функции приостановки и возобновления не поддерживаются контроллером SD версии 4.00 и более поздних версий, команды приостановки и возобновления не используются.	2'h0
21	Чтение/запись	**DPS - выбор текущих данных** Установка на 1 для команд передачи данных (т. е. данных чтения или записи с помощью линии DAT). Установка на 0 для всех других команд, включая: – Команды, использующие только линию CMD – Команды с занятыми сигналами на линии DAT[0] (SRS03.RTS=11b)	1'h0
20	Чтение/запись	**CICE - включение проверки индекса команды** При установке на 1 контролер проверяет, равно ли поле с индексом команды в ответе значению SRS03.CIDX. Если 0, проверка выполнена, и поле индекса команды в ответе игнорируется. Рекомендуемые настройки зависят от типа ответа. Подробную информацию см. в таблице ниже: – SRS03.RTS=00: 0 - нет ответа – SRS03.RTS=01: 0 - R2 – SRS03.RTS=10: 0 - R3, R4 – SRS03.RTS=10: 1 - R1, R5, R6, R7 – SRS03.RTS=11: 1 - R1b, R5b	1'h0

Бит	Тип	Описание	Сброс
19	Чтение/запись	**CRCCE - включение проверки CRC команды** При установке на 1 контроллер проверяет действительность поля CRC в ответе. Если 0, проверка CRC отключена, и поле CRC в ответе игнорируется. Проверка CRC должна быть отключена для ответов, которые не содержат фактическое значение CRC (некоторые ответы содержат все 1s вместо поля CRC), пи включена для ответов других типов. Рекомендуемые настройки зависят от типа ответа. Подробную информацию см. в таблице ниже: SRS03.RTS=00: 0 - нет ответа SRS03.RTS=01: 1 - R2 SRS03.RTS=10: 0 - R3, R4 SRS03.RTS=10: 1 - R1, R5, R6, R7 SRS03.RTS=11: 1 - R1b, R5b	1'h0
18	Чтение/запись	**SCF - флаг подкоманды** Этот бит добавлен из версии Version 4.10 для разделения основной команды и подкоманды. При передаче основной команды этот бит устанавливается на 0, при передаче подкоманды этот бит устанавливается на 1. Значение этого бита проверяется состоянием подкоманды в регистре текущего состояния (SRS09.SCMDS). Драйвер контроллера определяет основную команду или подкоманду. Контроллер не обращается к этому биту для передачи команды.	1'h0
17-16	Чтение/запись	**RTS - выбор типа ответа** определяет расчетную длину ответа. 00b - нет ответа 01b - 136-битовый ответ 10b - 48-битовый ответ 11b - 48-битовый ответ с BUSY Каждая команда предусматривает один из типов ответов, указанных выше. Информацию о проверке типа ответа, соответствующего переданной команде, см. спецификации карты/устройства.	2'h0
8	Чтение/запись	**RID - отключение прерывания ответа** При установке на 1 прерывание выполнения команды (SRS12.CC) будет отключено. Контроллер будет игнорировать SRS13.CC_SE и функционировать, как если бы SRS13.CC_SE был установлен на 0. При установке на 0 SRS12.CC будет включен или отключен в зависимости от бита SRS13.CC_SE.	1'h0
7	Чтение/запись	**RECE - включение проверки ошибки ответа** При установке на 1 контроллер будет проверять ответы R1/R5. В случае обнаружения ошибки в ответе бит SRS12.ERSP будет установлен на 1. Программа отправит этот бит, только если ожидается ответ R1/R5. Программа установит биты SRS03.RID и RECE на 1, если контроллер проверяет ответы R1/R5 на ошибки. Оба бита будут сброшены на 0, когда программный драйвер проверит ответы R1/R5 на ошибки. В случае ошибки ответа бит SRS12.ERSP (в регистре состояния прерывания) будет установлен на 1.	1'h0

Бит	Тип	Описание	Сброс
6	Чтение/запись	**RECT - тип ответа R1/R5** Выберите тип ответа R1 или R5 для проверки содержимого ответа. Будут оцениваться перечисленные ниже биты ошибки. RECT = 0, тип ответа - R1 (память SD): – бит 31 OUT_OF_RANGE – бит 30 ADDRESS_ERROR – бит 29 BLOCK_LEN_ERROR – бит 26 WP_VIOLATION – бит 25 CARD_IS_LOCKED – бит 23 COM_CRC_ERROR – бит 21 CARD_ECC_FAILED – бит 20 CC_ERROR – бит 19 ERRORRECT RECT = 1, тип ответа - R5 (SDIO): – бит 7 COM_CRC_ERROR – бит 3 ERROR – бит 1 FUNCTION_NUMBER – бит 0 OUT_OF_RANGE Это поле игнорируется, если SRS03.RECE=0.	1’h0
5	Чтение/запись	**MSBS - выбор нескольких блоков/одного блока** В этом поле можно задать передачу одного или нескольких блоков. 0 - один блок 1 - несколько блоков Это поле предусматривает аппаратную защиту с помощью бита запрета команды DAT в регистре текущего состояния (SRS09.CIDAT). Если CIDAT=1, все записи в этом поле игнорируются.	1’h0
4	Чтение/запись	**DTDS - выбор направления передачи данных** Выбор направления передачи данных для команд при DPS=1. – 0 - запись – 1 - чтение Для команд с SRS03.DPS=0 это поле игнорируется. Это поле предусматривает аппаратную защиту с помощью бита запрета команды DAT в регистре текущего состояния (SRS09.CIDAT). Если CIDAT=1, все записи в этом поле игнорируются.	1’h0

Бит	Тип	Описание	Сброс
3-2	Чтение/запись	**ACE - включение Auto CMD** Это поле позволяет отправить одну дополнительную команду карте/устройству при подаче команды. – 00b - нет автоматической передачи команды – 01b - Auto CMD12 – 10b - Auto CMD23 – 11b - выбор автоматической передачи команды CMD Если Auto CMD отключена (00b), контроллер не будет передавать дополнительную команду. Этот параметр используется, если автоматическая передача команды не нужна. Если установлена Auto CMD12 (01b), контроллер передает CMD12 (Отмена) автоматически после завершения последней передачи одного или нескольких блоков. Если включена Auto CMD23 (10b), контроллер передает CMD23 (задать число блоков) автоматически перед подачей команды передачи данных. Аргумент этой команды можно задать в SRS00. Если включена функция автоматического выбора Auto CMD (11b), контроллер передает CMD12 или CMD23 в зависимости от результата идентификации карты. Если SRS15.CMD23E == 1, контроллер передает CMD23, если установлен Auto CMD23. Если SRS15.CMD23E == 0, контроллер передает CMD12, если установлен Auto CMD12. При возникновении ошибки поданная команда не передается. Это поле предусматривает аппаратную защиту с помощью бита запрета команды DAT в регистре текущего состояния (SRS09.CIDAT). Если SRS09.CIDAT=1, все записи в этом поле игнорируются.	2'h0
1	Чтение/запись	**BCE - включение счетчика блоков** При установке на 1 для следующей передачи данных включается активный регистр-счетчик блоков. Активный регистр - либо SRS01.BCCT, либо SRS00.SAAR. При передаче каждого блока значение счетчика автоматически уменьшается. Передача нескольких блоков завершается, когда значение счетчика блоков станет 0. Если этот параметр включен, нельзя будет выполнить ограниченную передачу. При установке на 0 подсчет блоков отключается, и SRS01.BCCT сохраняет это значение. Передача будет ограниченной в не-DMA и SDMA режимах. В случае ADMA режима передача может быть ограниченной или неограниченной. Ограниченная передача заканчивается при чтении дескриптора с состоянием END (длина передачи определяется таблицей дескрипторов). В случае неограниченной передачи программа прямо задает тип команды ABORT для остановки передачи. Это поле предусматривает аппаратную защиту с помощью бита запрета команды DAT в регистре текущего состояния (SRS09.CIDAT). Если CIDAT=1, все записи в этом поле игнорируются.	1'h0

Бит	Тип	Описание	Сброс
0	Чтение/запись	**DMAE - включение DMA** При установке на 1 включается функционал DMA. DMA может быть включен, только если он поддерживается, как указано в информации о поддержке DMA в регистре SRS16.DMAS. Если DMA не поддерживается (из-за конфигурации контроллера), этот бит игнорируется. Это поле предусматривает аппаратную защиту с помощью бита запрета команды DAT в регистре текущего состояния (SRS09.CIDAT). Если SRS09.CIDAT=1, все записи в этом поле игнорируются. Примечание: в режиме ADMA2 предусмотрена только ограниченная передача, т. е. этот бит будет установлен на 1.	1’h0

1.3.9.10.3.5.31SRS04 (0x0210)

Регистры SRS04 - SRS07 хранят ответ, возвращаемый картой.

Схема фактического ответа устройства и содержимое регистров SRS04 - SRS07 зависят от типа ответа. Тип ответа определяется полем RTS (тип ответа) для всех заданных пользователем команд.

Отдельные случаи - ответ Auto-CMD12 (R1b в спецификации SD-памяти) и ответ Auto-CMD23 (R1 в спецификации SD-памяти). Ответы Auto-CMD12 и Auto-CMD23 обрабатываются ядром автоматически и переходят в регистр SRS07, независимо от значения RTS.

Взаимосвязь SRS04-SRS07 с полем полученного ответа:

– Auto-CMD12 resp: поле ответа R[39:8] - RESP3[31:0];

– Auto-CMD23 resp: поле ответа R[39:8] - RESP3[31:0];

– Нет ответа: RTS=00b;

– 136-бит: RTS=01b, поле ответа R[127:8] - RESP3[23:0], RESP2[31:0], RESP1[31:0], RESP0[31:0];

– 48-bitбит: RTS=10b, поле ответа R[39:8] - RESP0[31:0];

– 48-бит с BUSY: RTS=11b, поле ответа R[39:8] - RESP0[31:0].

Примечание по реализации: значения регистров неопределенные после сброса и будут действительны после получения ответа.

Таблица 1.465 – Поля регистра SRS04

Бит	Тип	Описание	Сброс
31-0	R	**RESP0 - регистр ответа № 0**	32’h0

1.3.9.10.3.5.32SRS05 (0x0214)

Описано в SRS04.

Таблица 1.466 – Поля регистра SRS05

Бит	Тип	Описание	Сброс
31-0	R	**RESP1 - регистр ответа № 1**	32’h0

1.3.9.10.3.5.33SRS06 (0x0218)

Описано в SRS04.

Таблица 1.467 – Поля регистра SRS06

Бит	Тип	Описание	Сброс
31-0	R	**RESP2 - регистр ответа № 2**	32’h0

1.3.9.10.3.5.34SRS07 (0x021c)

Описано в SRS04.

Таблица 1.468 – Поля регистра SRS07

Бит	Тип	Описание	Сброс
31-0	R	**RESP3 - регистр ответа № 3**	32’h0

1.3.9.10.3.5.351SRS08 (0x0220)

SRS08 - буфер данных.

Таблица 1.469 – Поля регистра SRS08

Бит	Тип	Описание	Сброс
31-0	Чтение/запись	**BDP - порт данных буфера** Поле предназначено для доступа к внутреннему буферу (блоку данных) в режиме передачи не-DMA. 8-битовый, 16-битовый или 32-битовый доступ к SRS08 возможен со следующими ограничениями: – при прямом порядке байтов возможен только последовательный непрерывный доступ. Например, если программа получает доступ к BDP[7:0], то следующая передача получит доступ к BDP[15:8]. Пропуск байтов запрещен; – каждый новый блок начинается с младшего байта BDP (BDP[7:0]); – если размер блока не является кратным 32 битам, и программа получает доступ к BDP с помощью 32-битовых слов, то лишние байты последнего слова игнорируются. Это позволяет программному драйверу использовать только передачи 32-битовых данных, независимо от размера блока; – доступ к регистру с предупреждением - указатели FIFO могут быть повреждены, если буфер не готов, или если число доступов превышает размер передаваемого блока (SRS01.TBS). Ниже указаны все передачи (различные варианты включения байтов), разрешенные на SRS08: Ширина передачи = 32 бита: – be[3:0] = 'b1111 → BDP[31:0]. Ширина передачи = 16 битов: – be[3:0] = 'b0011 → BDP[15:0]; – be[3:0] = 'b1100 → BDP[31:16]. Ширина передачи = 8 битов: – be[3:0] = 'b0001 → BDP[7:0]; – be[3:0] = 'b0010 → BDP[15:8]; – be[3:0] = 'b0100 → BDP[23:16]; – be[3:0] = 'b1000 → BDP[31:24].	32'h0

1.3.9.10.3.5.36 SRS09 (0x0224)

SRS09 - регистр текущего состояния.

Таблица 1.470 – Поля регистра SRS09

Бит	Тип	Описание	Сброс
28	R	**SCMDS - состояние подкоманды** Регистр SRS03 и регистры ответов (SRS04-SRS07) используются для основных команд и подкоманд. Это состояние используется для различения состояния ошибки, основной команды или подкоманды, указанных в регистра состояния прерывания ошибки. Сразу перед чтением этого регистра SRS03.SCF копируется в это состояние. Это состояние вступает силу только после генерирования прерывания по ошибке ответа (SRS12.ERSP). Прерывание по ошибке данных также генерируется командой, не переданной по ошибке (SRS09.CNIBE), или прерыванием по ошибке Auto CMD (SRS12.EAC) с командной, не переданной по ошибке, Auto CMD12 (SRS15.CNIACE). SRS09.SCMDS указывает непереданную команду (основную команду или подкоманду).	1'h0

Бит	Тип	Описание	Сброс
27	R	**CNIBE - команда, не переданная по ошибке** Параметр состояния CNIBE указывает, что команда не может быть подана по ошибке, кроме ошибки Auto CMD12 (эквивалентное состояние ошибки Auto CMD12 определяется как SRS15.CNIACE.) Это состояние устанавливается на 1, когда контроллер не может подать команду после настройки регистра команды.	1'h0
26	R	**LVSIRSLT - результат идентификации LVS** Результат идентификации сигналов низкого напряжения. Этот бит содержит действительную информацию, только если бит выполнения идентификации LVS был изменен с 1 на 0. Это поле сбрасывается при наступлении любого из следующих условий: (a) питание шины SD для VDD1 установлено 0 (b) при вставке карты отображается извлечение карты (c) HRS00.SWR (программный сброс)	1'h0
24	R	**CMDSL - уровень сигнала линии CMD** Значение равно фактическому уровню сигнала на линии CMD интерфейса SD (pad_mem_cmd). Используется для отладки.	1'h0
23-20	R	**DATSL1 - уровень сигнала линии DAT[3:0]** Значение равно фактическому уровню сигнала на линии DAT интерфейса SD/eMMC: – SRS09.23 - pad_mem_data[3] – SRS09.22 - pad_mem_data[2] – SRS09.21 - pad_mem_data[1] – SRS09.20 - pad_mem_data[0]	4'h0
19	R	**WPSL - уровень контакта переключения защиты от записи** Значение равно фактическому уровню сигнала контакта с защитой от записи интерфейса SD/eMMC (*pad_mem_wpbar*). – 1 - запись включена – 0 - запись отключена	1'h0
18	R	**CDSL - уровень контакта обнаружения карты** Значение равно уровню инвертированного сигнала на контакте обнаружения карты интерфейса SD/eMMC (*pad_mem_ctrl_O*). – 1 - карта вставлена – 0 - нет карты в слоте Устранение дребезга не реализовано на CDSL, поэтому при нормальной работе рекомендуется использовать бит «Карта вставлена» (CI). Бит CDSL используется только для отладки.	1'h0
17	R	**CSS - стабильное состояние карты** Указывает стабильный уровень контакта обнаружения карты (CDSL). – 1 - значение CDSL стабильно – 0 - CDSL не стабильно (во время вставки/извлечения карты или сброса) Поле используется для отладки.	1'h0
16	R	**CI - карта вставлена** Указывает, что карта вставлена в слот. – 0 - нет карты в слоте – 1 - карта вставлена В отличие от SRS09.CDSL, значение бита SRS09.CI гарантированно стабильное (т. е. на этом биту обеспечивается устранение дребезга). Во время нормальной работы контроллера рекомендуется использовать этот бит.	1'h0

Бит	Тип	Описание	Сброс
11	R	**BRE - включить чтение буфера** Это поле отображает состояние буфера данных (SRS08.BDP) для передачи данных чтения в не-DMA режиме. – 1 - из буфера данных можно считать действительные данные – 0 - нет действительных данных в буфере данных После чтения всего блока данных этот бит изменяется на 0.	1'h0
10	R	**BWE - включить запись буфера** Этот бит отображает состояние буфера данных (SRS08.BDP) для передачи данных записи в не-DMA режиме. – 1 - данные можно записать в буфер данных – 0 - данные нельзя записать После чтения всего блока данных этот бит изменяется на 0. Этот бит будет сброшен в случае SBGR при передаче данных записей в не-DMA режиме (даже если внутренний буфер готов). Буфер нельзя записывать после SBGR. В случае установки BWR только одно действие из S/W предназначено для очистки состояния прерывания.	1'h0
9	R	**RTA - передача данных чтения активна** Указывает состояние передачи данных чтения. – 0 - передача данных чтения не выполняется – 1 - передача данных чтения выполняется Бит устанавливается на 1 после отправки команды чтения или перезапуска передачи данных чтения с помощью запроса продолжения (SRS10.CREQ). Бит устанавливается на 0 аппаратным обеспечением после последнего блока передачи данных чтения или остановки передачи данных чтения с помощью запроса интервала блока (SRS10.SBGR). В обоих случаях все данные должны быть считаны системой из внутреннего буфера данных после установки этого бита на 0. Другими словами SRS09.RTA=0 означает, что все данные уже переданы системе, и внутренний буфер данных пуст.	1'h0
8	R	**WTA - передача данных записи активна** Указывает состояние передачи данных записи. – 0 - передача данных записи не выполняется – 1 - передача данных записи выполняется Бит устанавливается на 1 после отправки команды записи или перезапуска передачи данных записи с помощью запроса продолжения (SRS10.CREQ). Бит устанавливается на 0 аппаратным обеспечением после последнего блока передачи данных записи или остановки передачи данных записи с помощью запроса остановки в промежутке между блоками (SRS10.SBGR). В обоих случаях все данные должны быть переданы на карту из внутреннего буфера данных после установки этого бита на 0. Другими словами, WTA=0 означает, все данные уже переданы на карту, и уже был получен ответ CRC для последнего блока данных.	1'h0
7-4	R	**DATSL2 - уровень сигнала линии DAT[7:4]** Значение равно фактическому уровню сигнала на линии DAT интерфейса SD/eMMC: – SRS9.7 - pad_mem_data[7] – SRS9.6 - pad_mem_data[6] – SRS9.5 - pad_mem_data[5] – SRS9.4 - pad_mem_data[4]	4'h0

Бит	Тип	Описание	Сброс
2	R	**DLA - активна линия DAT** указывает использование линий DAT интерфейса SD в текущий момент. – 1 - линии DAT активны (используются) – 0 - линии DAT неактивны (не используются) Этот бит устанавливается на 1, когда активны биты передачи данных чтения или записи (SRS09.RTA=1 или SRS09.WTA=1), или если карта указывает состояние «занято» на линиях DAT. Карта может стать «занятой» сразу после операции записи или команды запроса ответа с состоянием «занято». Спадающий фронт этого бита (изменение с 1 на 0) напрямую активирует прерывание завершения передачи (SRS12.TC).	1'h0
1	R	**CIDAT - запрет команды DAT** Указывает, что контроллер может подать команду для линии DAT. Команды, использующие линию DAT, включают в себя команды данных записи или чтения с ответом «занято». – 1 - команда, использующая линию DAT, не может быть передана – 0 - команда, использующая линию DAT, может быть передана Если CIDAT=1, то SRS03[15:0] обеспечен защитой от записи. Программа может записать SRS03[15:0], только если CIDAT=0.	1'h0
0	R	**CICMD - запрет команды CMD** Указывает, что контроллер может подать команду. – 0 - команда может быть передана – 1 - команда не может быть передана Если бит установлен на 0, он означает, что линия CMD не используется, и контроллер может подать команду SD с помощью линии CMD. Этот бит устанавливается сразу после записи CI, указывает начало передачи команды. Этот бит сбрасывается при получении ответа по команде. Даже если запрет команды DAT установлен на 1, команды, использующие линию CMD, могут быть переданы, если запрет команды CMD установлен на 0. Изменение с 1 на 0 напрямую активирует прерывание выполнения команды (SRS12.CC).	1'h0

1.3.9.10.3.5.37SRS10 (0x0228)

SRS10 - управление контроллером 1 (общее / питание / промежуток между блоками / активация).

Таблица 1.471 – Поля регистра SRS10

Бит	Тип	Описание	Сброс
26	Чтение/запись	**WORM - включение события активации при извлечении SD-карты** При установке на 1 активируется событие включения через извлечение карты в регистре SRS12.CR.	1'h0
25	Чтение/запись	**WOIS - включение события активации при вставке карты** При установке на 1 включает событие активации через вставку карты в регистре SRS12.CIN.	1'h0
24	Чтение/запись	**WOIQ - включение события активации при прерывании карты** При установке на 1 включает событие активации через прерывание карты в регистре SRS12.CINT.	1'h0

Бит	Тип	Описание	Сброс
19	Чтение/запись	**IBG - прерывание в промежутке между блоками** При установке на 1 активирует обнаружение прерывания в промежутке между блоками в случае передачи нескольких блоков. Этот бит действителен только в режиме SD4. Если SD-карта не должна сообщать о прерывании во время передачи нескольких блоков, этот бит должен быть установлен на 0.	1'h0
18	Чтение/запись	**RWC - управление ожиданием чтения** При установке на 1 активируется управление ожиданием чтения. Функция ожидания чтения является дополнительной для карт SDIO. Если карта не поддерживает ожидание чтения, этот бит никогда не должен быть установлен на 1, в противном случае возможен конфликт с линией DAT.	1'h0
17	W1S	**CR - запрос продолжения** При установке на 1 выполняется перезапуск передачи, ранее остановленной с помощью бита остановки в промежутке между блоками. Программа устанавливает бит SRS10.SBGR (остановка в промежутке между блоками) на 0 перед заданием запроса продолжения (CR). Если SRS10.SBGR=1, все операции записи в запрос продолжения игнорируются. Сброс бита SRS10.SBGR можно выполнить перед или вместе с записью CR. Бит запроса продолжения сбрасывается автоматически контроллером, когда SRS09.DLA (линия DAT активна) меняется с 0 на 1, указывая фактический перезапуск передачи.	1'h0
16	Чтение/запись	**SBGR - запрос остановки в промежутке между блоками** При установке на 1 выполняется организация порядка остановки транзакций чтения и записи в следующем промежутке между блоками при передаче в не-DMA режиме, режимах SDMA и ADMA. Программа поддерживает SBGR=1 до завершения текущей передачи (обычно ожидая бит завершения передачи). После активации события завершения передачи программа устанавливает SBGR на 0. В случае передачи данных чтения контроллер останавливается после получения с карты следующего блока данных. Используется механизм ожидания чтения, если он включен с помощью SRS10.RWC, или останавливается блок с карты (*pad_mem_webar_t*), если функция ожидания чтения отключена. В случае передачи данных записи контроллер останавливается после записи последнего блока в буфер данных. Контроллер передает все данные, уже записанные во внутреннем буфере данных, перед остановкой передачи. В случае остановки передачи данных записи в не-DMA режиме программное обеспечение устанавливает этот бит только в промежутке между блоками (блок (режим SD)).	1'h0
11-9	Чтение/запись	**BVS - выбор напряжения шины SD** Это поле используется для настройки уровня напряжения VDD1. Состояние этого поля напрямую управляет портом pad_bv. – 000b-100b - зарезервировано – 101b - 1,8 В (стандарт) для встроен. – 110b - 3,0V (стандарт) – 111b - 3,3 В (стандарт) – другое - зарезервировано	3'h0

Бит	Тип	Описание	Сброс
8	Чтение/запись	**BP - питание шины SD для VDD1** При установке на 1, напряжение VDD1 подается на карту/устройство. Состояние этого бита напрямую управляет *pad_mem_ctrl_1*. При установке бита на 0 контроллер прекращает управление линиями SDCLK, CMD/DAT. Если устройство подключено к контроллеру, линии понижаются перед отключением VDD1. Контроллер установит этот бит автоматически на 0 при извлечении карты из слота (т. е. при переключении с высокого уровня контакта pad_mem_ctrl_0 на низкий уровень). Таким образом, обеспечивается поддержка «горячего извлечения».	1'h0
7	Чтение/запись	**CDSS - выбор сигнала обнаружения карты** Механизм обнаружения карты основывается либо на порте pad_mem_ctrl_0, либо на значении регистра. 0 - контакт pad_mem_ctrl_0 (нормальный режим) 1 - бит CDTL(SRS10.6) (тестовый режим)	1'h0
6	Чтение/запись	**CDTL - уровень тестирования обнаружения карты** Обозначает состояние вставки карты, когда SRS10.CDSS=1. Бит предусмотрен для тестирования. 0 - нет карты 1 - карта вставлена	1'h0
5	Чтение/запись	**EDTW - ширина расширенной передачи данных** Этот бит включает/отключает режим ширины 8-битовой шины DAT. 0 - ширина шины выбирается с помощью SRS10.DTW 1 - включен 8-битовый режим	1'h0
4-3	Чтение/запись	**DMASEL - выбор DMA** В этом поле можно выбрать режим DMA. Работа поля зависит от бита совместимости контроллера (SRS15.HV4E). Режим совместимости контроллера версии 3.00 (SRS15.HV4E=0) 00b - режим SDMA 01b - зарезервировано 10b - ADMA2 (32-битовый адрес) 11b - ADMA2 (64-битовый адрес) Режим совместимости контроллера версии 4.00 (SRS15.HV4E=1) 00b - режим SDMA 01b - не используется 10b - режим ADMA2 (ADMA3 не поддерживается или отключен) 11b - выбор ADMA2 или ADMA3 Ширина шины адреса ADMA2 настраивается с помощью бита 64-битовой адресации в регистре 2 контроллера, когда SRS15.HV4E=1.	2'h0
2	Чтение/запись	**HSE - включение высокой скорости** Выбор рабочего режима со скоростью по умолчанию (HSE=0) или высокой скоростью (HSE=1). Максимальная тактовая частота SD равна 0-25 МГц в режиме со скоростью по умолчанию и 0-50 МГц в режиме с высокой скоростью.	1'h0
1	Чтение/запись	**DTW - ширина передачи данных** Бит используется для установки ширины шины DAT на 1 или 4. 0 - 1-битовый режим 1 - 4-битовый режим Этот бит игнорируется, если SRS10.EDTW устанавливается на 1 (выбирается 8-битовый режим).	1'h0

Бит	Тип	Описание	Сброс
0	Чтение/запись	**LEDC - управление LED** Состояние этого бита напрямую управляет портом внешних светодиодных индикаторов контроллера. LEDC=1 - включение светодиодных индикаторов, whileEDC=0 - выключение светодиодных индикаторов. Программное обеспечение включает светодиодные индикаторы для предупреждения пользователя о том, что не следует извлекать карту во время выполнения передачи.	1'h0

1.3.9.10.3.5.38SRS11 (0x022c)

SRS11 - управление контроллера 2 (тактирование, тайм-аут, сброс).

Таблица 1.472 – Поля регистра SRS11

Бит	Тип	Описание	Сброс
26	W1S	**SRDAT - программный сброс для линии DAT** При установке на 1 выполняется сброс логики тракта данных, включая буферы данных и логику DMA. Сбрасываются следующие регистры и биты: Регистр SRS08: – Буфер Регистр SRS09: – Включение чтения буфера – Включение записи буфера – Передача данных чтения активна – Передача данных записи активна – Активна линия DAT – Регистр запрета команд DATSRS10: – Запрос продолжения – Регистр SRS12 запроса остановки в промежутке между блоками: – Чтение буфера готово – Запись буфера готова – Прерывание DMA – Событие в промежутке между блоками – Передача завершена После сброса бит SRS11.SRDAT автоматически очищается. Для выполнения сброса потребуется некоторое время, поэтому программное обеспечение дожидается SRS11.SRDAT=0 и продолжает выполнять другие операции, только если SRS11.SRDAT=0.	1'h0
25	W1S	**SRCMD - программный сброс для линии CMD** При установке на 1 выполняется сброс логики, касающейся генерирования команд и проверки ответов. Сбрасываются следующие регистры и биты: – Регистр SRS09: Запрет команды CMD – Регистр SRS12: Завершение команды После сброса бит SRS11.SRCMD автоматически очищается. Для выполнения сброса потребуется некоторое время, поэтому программное обеспечение дожидается SRCMD=0 и продолжает выполнять другие операции, только если SRS11.SRCMD=0.	1'h0

Бит	Тип	Описание	Сброс
24	W1S	**SRFA - программный сброс для всех** При установке на 1 весь слот сбрасывается. После сброса бит SRFA автоматически очищается. Для выполнения сброса потребуется некоторое время, поэтому программное обеспечение дожидается SRFA=0 и продолжает выполнять другие операции, только если SRFA=0. Кроме ого, после SRFA программа должна выполнить сброс и повторную инициализацию карты, вставленную в слот. Питание SD-карты может быть включено через 1 мсек после очистки этого бита, чтобы обеспечить надлежащий сброс SD-карты.	1'h0
19-16	Чтение/запись	**DTCV - значение счетчика тайм-аута данных** Это значение указывает интервал, по которому определяется тайм-аут линии DAT. Интервал рассчитывается следующим образом: – 1111b - зарезервировано – 1110b - t_sdmclk*2л(27+2) – 1101b - t_sdmclk*2л(26+2) – … – 0001b - t_sdmclk*2л(14+2) – 0000b - t_sdmclk*2л(13+2) Где t_sdmclk - тактовый период sdmclk. См. информацию о факторах, из-за которых генерируются тайм-аут данных, в описании регистра ошибки по тайм-ауту данных (SRS12.EDT).	4'h0
15-8	Чтение/запись	**SDCFSL - выбор частоты SDCLK (нижняя часть)** Этот регистр и SRS11.SDCFSH используются для расчета тактовой частоты SDCLK. Частота SDCLK рассчитывается по следующим формулам: – sdclk = sdmclk, где (N=0) – sdclk = sdmclk/2N, где (N>0) Переменная N - сумма SRS11.SDCFSH и SRS11.SDCFSL. Значения регистров SDCFSL и SDCFSH можно изменить, только если SRS11.SDCE (включение тактирования SD)=0.	8'h0
7-6	Чтение/запись	**SDCFSH - выбор частоты SDCLK (верхняя часть)** Этот регистр является расширением SDCFSL.	2'h0
2	Чтение/запись	**SDCE - включение тактирования SD** При установке на 1 тактирование SDCLK включено. При сбросе на 0 тактирование SDCLK останавливается. Контроллер сбрасывает SDCE автоматически при извлечении карты из слота (т. е. при переключении с высокого уровня контакта *pad_mem_ctrl_0* на низкий уровень). Тактирование SDCLK должно быть остановлено программой при изменении делителя частоты (т. е. бит SDCE будет сброшен перед записью в SRS11.SDCFSL, SRS11.SDCFSH).	1'h0
1	R	**ICS - стабилизация внутреннего тактового генератора** При чтении 1 указывает, что тактовый генератор на контакте sdmclk контроллера стабилен после установки ICE на 1. При чтении 0 указывает, что тактовый генератор еще не стабилизировался (например, внешний PLL генерирует сигнал о том, что тактовый генератор еще не заблокирован). Значение ICS равно фактическому уровню сигнала на контакте ICS контроллера. Пользователь подключает ICS к внешнему PLL в случае необходимости. В противном случае ICS должен быть подключен напрямую к выходу ICE контроллера.	1'h0

Бит	Тип	Описание	Сброс
0	Чтение/запись	**ICS - включение внутреннего тактового генератора** Это поле предназначено для управления (включения/выключения) внешним тактовым генератором (например, PLL). Биты ICE каждого слота соединены логическим «ИЛИ» и управляют контактом ICE. Это означает, что контакт ICE равен 0, только когда ICE в 0 для каждого слота контроллера. Контакт ICE составляет 1, если хотя бы один из битов ICE установлен на 1. При установке на 0 тактовый генератор на контакте sdmclk может быть оставлен внешне. Если sdmclk остановлен, контроллер переключается в режим низкой мощности. Регистры хоста продолжают работать (операции чтения и записи действительны), даже если тактовый генератор на sdmclk остановлен. Установки бита ICE не влияет на обнаружение карты. Это означает, что обнаружение карты срабатывает, даже если остановлен тактовый генератора на sdmclk.	1'h0

1.3.9.10.3.5.39SRS12 (0x0230)

SRS12 - состояние ошибки/нормального прерывания.

Таблица 1.473 – Поля регистра SRS12

Бит	Тип	Описание	Сброс
27	W1C	**ERSP - ошибка ответа** Генерируется при обнаружении ошибки в ответе R1 или R5. Ошибки проверяются, только если RECE установлен на 1.	1'h0
25	W1C	**EADMA - ошибка ADMA** Генерируется при возникновении ошибки во время передачи данных чтения или записи ADMA. Для устранения причины ошибки состояние движка ADMA при возникновении ошибки сохраняется в регистр состояний ошибок ADMA, адрес дескриптора, обрабатываемого при возникновении ошибки, сохраняется в регистре адресов системы ADMA.	1'h0
24	W1C	**EAC - ошибка Auto CMD (только режим SD)** Генерируется при возникновении ошибки во время передачи команды Auto CMD12/Auto CMD23. Означает одно из следующих условий: – один из битов в регистре SRS15 изменился с 0 на 1, – Auto CMD12 не выполняется из-за ошибки предыдущей команды.	1'h0
23	W1C	**ECL - ошибка текущего предела** Это поле содержит ошибку, сообщенную на входном контакте *pad_cle* контроллера. Генерирование ошибки происходит вне этого контроллера IP. Примечание: если внешний источник питания устройства SD/eMMC не обнаружил и не сообщил об этой ошибке, установите ошибку текущего предела (вход *sdphy_dfi_cle* ядра контроллера) на 0.	1'h0
22	W1C	**EDEB - ошибка конечного бита данных (только режим SD)** При установке на 1 указывает обнаружение 0 в позиции конечного бита передачи данных чтения, использующей линию DAT, или позиции конечного бита состояния CRC записи.	1'h0

Бит	Тип	Описание	Сброс
21	W1C	**EDCRC - ошибка CRC данных (только режим SD)** При установке на 1 указывает обнаружение ошибки CRC при передаче данных чтения, при которой используется линия DAT, или обнаружении состояния CRC данных записи со значением, отличным от 010. Этот бит устанавливается на 1 сразу при обнаружении конфликта на линии CMD. О конфликте сообщается путем установки этого бита и SRS12.EDT на 1.	1'h0
20	W1C	**EDT - ошибка тайм-аута данных (только режим SD)** При установке на 1 указывает на обнаружение одного из следующих условий тайм-аута: 1) Тайм-аут состояния «занято» для ответа с состоянием «занято». 2) Тайм-аут состояния «занято» после состояния CRC записи. 3) Тайм-аут состояния CRC записи. 4) Тайм-аут данных чтения. Этот бит устанавливается на 1 сразу при обнаружении конфликта на линии CMD.	1'h0
19	W1C	**ECI - ошибка указателя команды (только режим SD)** При установке на 1 указывает, что в ответе команды возникла ошибка указателя.	1'h0
18	W1C	**ECEB - ошибка конечного бита команды (только режим SD)** При установке на 1 указывает обнаружение конечного бита ответа команды, равного 0.	1'h0
17	W1C	**ECCRC - ошибка CRC команды (только режим SD)** При установке на 1 указывает на возникновение ошибки CRC команды.	1'h0
16	W1C	**ECT - ошибка тайм-аута команды** При установке на 1 указывает отсутствие ответа в течение 64 циклов SDCLK от конечного бита команды.	1'h0
15	R	**EINT - прерывание ошибки** Этот бит устанавливается, если задается любой из битов в диапазоне SRS12[31:16]. Программа может проверить на наличие ошибки, считав этот первый один бит.	1'h0
14	R	**CQINT - прерывание организации очереди команд** Прерывание активируется при установке хотя бы одного из битов в регистре CQIS. Прерывание сбрасывается только при сбросе источника прерывания в регистре CQIS.	1'h0
13	R	**FXE - событие FX** Если SRS03.RECE установлен на 1, и SRS03.RECT установлен на 0, прерывание означает, что 14-й бит ответа, сохраненный как 6-й бит регистра SRS04, установлен на 1. Если SRS03.RECE установлен на 1, это прерывание может сбросить только следующий ответ типа R1, содержащий 14-й бит состояния карты, равный 0. Если SRS03.RECE установлен на 0, прерывание означает, что 14-й бит ответа, сохраненный как 6-й бит регистра SRS04, установлен на 1, кроме следующих случаев: 1) Если аргумент бита 15 CMD13 равен 1, то ответ не изменит значения этого прерывания. 2) Если переданная команда не имеет ответа, значение этого прерывания не изменится.	1'h0

Бит	Тип	Описание	Сброс
8	R	**CINT - прерывание карты** Указывает прерывание карты. CINT не проверяется тактовым генератором карты, поэтому прерывание можно обнаружить даже при остановке тактового генератора SD (SRS11.SDCE=0). CINT не сбрасывается путем записи 1. Вместо этого программа сбрасывает источник прерывания внутри карты. После обнаружения прерывания карты программа останавливает выполнение обнаружения прерываний, сбрасывая SRS13.CINT_SE на 0. Затем программа сбрасывает источник прерывания внутри карты с помощью соответствующих команд. Подробную информацию см. в спецификации карты SDIO. После сброса источника прерывания карта передает сигнал прерывания контроллеру. После завершения прерывания включается функция обнаружения прерывания путем установки SRS13.CINT_SE обратно на 1.	1'h0
7	W1C	**CR - извлечение карты** Генерируется, когда бит SRS09.CI меняется с 1 на 0, указывая событие извлечения карты. 1 указывает на извлечение карты из слота. 0 указывает, что состояние карты стабильно (вставлена или извлечена), или что выполняется устранение дребезга.	1'h0
6	W1C	**CIN - вставка карты** Генерируется, когда бит SRS09.CI меняется с 0 на 1, указывая событие вставки карты. 1 указывает на вставку карты в слот. 0 указывает, что состояние карты стабильно (вставлена или извлечена), или что выполняется устранение дребезга.	1'h0
5	W1C	**BRR - чтение буфера готово** Генерируется, когда BRE меняется с 0 на 1, указывая, что программа может читать буфер данных. Это поле работает по-другому при последовательности настройки SD, т. е. когда бит выбора тактового генератора выборки (SRS15.SCS) равен 1. Он устанавливается на 1 после завершения настройки, независимо от результата этапа. В соответствии со Стандартом, во время настройки SD о прерываниях не сообщается, кроме уведомления о готовности буфера к чтению.	1'h0
4	W1C	**BWR - запись буфера готова** Генерируется, когда BWE меняется с 0 на 1, указывая, что программа может записать в буфер данных.	1'h0
3	W1C	**DMAINT - прерывание DMA** В режиме SDMA прерывание DMA генерируется, когда контроллер обнаруживает границу буфера SDMA хоста. В режиме ADMA прерывание DMA генерируется, когда флаг INT устанавливается в дескрипторе ADMA, обрабатываемом в настоящий момент.	1'h0
2	W1C	**BGE - событие в промежутке между блоками** Генерируется при остановки транзакции чтения/записи в промежутке между блоками в результате установки SRS10.SBGR на 1.	1'h0

Бит	Тип	Описание	Сброс
1	W1C	**TC - передача завершена** Режим SD: Генерируется, когда передача, использующая линию DAT, завершена. Передачи, использующие линию DAT, включают в себя передачи и команды чтения/записи с ответом с состоянием «занято». В случае передачи данных чтения TC указывает, что все данные были переданы с карты в систему (т. е. FIFO хоста пустой после чтения последнего блока данных). В случае передачи данных записи TC указывает, что все данные были переданы из хоста на карту (т. е. FIFO хоста пустой после записи последнего блока данных), и что карта приняла данные (после последнего блока был сгенерирован сигнал «занято»). В случае команды с ответом «занято» TC указывает, что сигнал «занято» был передан после ответа.	1'h0
0	W1C	**CC - завершение команды** генерируется при получении конечного бита ответа на команду Auto-CMD12. Команда Auto-CMD12 не генерирует CC.	1'h0

1.3.9.10.3.5.40SRS13 (0x0234)

SRS13 - включение состояния ошибки/нормального состояния.

Таблица 1.474 – Поля регистра SRS13

Бит	Тип	Описание	Сброс
27	Чтение/запись	**ERSP_SE - включение состояния ошибки ответа (только режим SD)** 1 - включено 0 - маскировано	1'h0
25	Чтение/запись	**EADMA_SE - включение состояния ошибки ADMA (только режим SD)** 1 - включено 0 - маскировано	1'h0
24	Чтение/запись	**EAC_SE - включение состояния ошибки Auto CMD (только режим SD)** 1 - включено 0 - маскировано	1'h0
23	Чтение/запись	**ECL_SE - включение состояния ошибки текущего предела (только режим SD)** 1 - включено 0 - маскировано	1'h0
22	Чтение/запись	**EDEB_SE - включение состояния ошибки конечного бита данных (только режим SD)** 1 - включено 0 - маскировано	1'h0
21	Чтение/запись	**EDCRC_SE - включение состояния ошибки CRC данных (только режим SD)** 1 - включено 0 - маскировано	1'h0
20	Чтение/запись	**EDT_SE - включение состояния ошибки тайм-аута данных (только режим SD)** 1 - включено 0 - маскировано	1'h0

Бит	Тип	Описание	Сброс
19	Чтение/запись	**ECI_SE - включение состояния ошибки индекса команды (только режим SD)** 1 - включено 0 - маскировано	1'h0
18	Чтение/запись	**ECEB_SE - включение состояния ошибки конечного бита команды (только режим SD)** 1 - включено 0 - маскировано	1'h0
17	Чтение/запись	**ECCRC_SE - включение состояния ошибки CRC команды (только режим SD)** 1 - включено 0 - маскировано	1'h0
16	Чтение/запись	**ECT_SE - включение состояния ошибки тайм-аута команды (только режим SD)** 1 - включено 0 - маскировано	1'h0
14	Чтение/запись	**CQINT_SE - включение состояния очереди команд** 1 - включено 0 - маскировано	1'h0
13	Чтение/запись	**FXE_SE - включение состояния события FX** 1 - включено 0 - маскировано	1'h0
8	Чтение/запись	**CINT_SE - включение состояния прерывания карты** 1 - включено 0 - маскировано	1'h0
7	Чтение/запись	**CR_SE - включение состояния извлечения карты** 1 - включено 0 - маскировано	1'h0
6	Чтение/запись	**CIN_SE - включение состояния вставки карты** 1 - включено 0 - маскировано	1'h0
5	Чтение/запись	**BRR_SE - включение состояния готовности буфера для чтения** 1 - включено 0 - маскировано	1'h0
4	Чтение/запись	**BWR_SE - включение состояния готовности буфера к записи** 1 - включено 0 - маскировано	1'h0
3	Чтение/запись	**DMAINT_SE - включение состояния прерывания DMA** 1 - включено 0 - маскировано	1'h0
2	Чтение/запись	**BGE_SE - включение состояния события в промежутке между блоками** 1 - включено 0 - маскировано	1'h0
1	Чтение/запись	**TC_SE - включение состояния завершения передачи** 1 - включено 0 - маскировано	1'h0

Бит	Тип	Описание	Сброс
0	Чтение/запись	**CC_SE - включение состояния завершения команды** 1 - включено 0 - маскировано	1'h0

1.3.9.10.3.5.41SRS14 (0x0238)

SRS14 - включение сигнала ошибки/нормального сигнала.

Таблица 1.475 – Поля регистра SRS14

Бит	Тип	Описание	Сброс
27	Чтение/запись	**ERSP_IE - включение прерывания ошибки ответа** 1 - включено 0 - маскировано	1'h0
25	Чтение/запись	**EADMA_IE - включение прерывания ошибки ADMA (только режим SD)** 1 - включено 0 - маскировано	1'h0
24	Чтение/запись	**EAC_IE - включение прерывания ошибки Auto CMD (только режим SD)** 1 - включено 0 - маскировано	1'h0
23	Чтение/запись	**ECL_IE - включение прерывания ошибки текущего предела (только режим SD)** 1 - включено 0 - маскировано	1'h0
22	Чтение/запись	**EDEB_IE - включение прерывания ошибки конечного бита данных (только режим SD)** 1 - включено 0 - маскировано	1'h0
21	Чтение/запись	**EDCRC_IE - включение прерывания ошибки CRC данных (только режим SD)** 1 - включено 0 - маскировано	1'h0
20	Чтение/запись	**EDT_IE - включение прерывания ошибки тайм-аута данных (только режим SD)** 1 - включено 0 - маскировано	1'h0
19	Чтение/запись	**ECI_IE - включение прерывания ошибки индекса команды (только режим SD)** 1 - включено 0 - маскировано	1'h0
18	Чтение/запись	**ECEB_IE - включение прерывания ошибки конечного бита команды (только режим SD)** 1 - включено 0 - маскировано	1'h0
17	Чтение/запись	**ECCRC_IE - включение прерывания ошибки CRC команды (только режим SD)** 1 - включено 0 - маскировано	1'h0

Бит	Тип	Описание	Сброс
16	Чтение/запись	**ECT_IE - включение прерывания ошибки тайм-аута команды (только режим SD)** 1 - включено 0 - маскировано	1'h0
14	Чтение/запись	**CQINT_IE - включение прерывания очереди команд** 1 - включено 0 - маскировано	1'h0
13	Чтение/запись	**FXE_IE - включение прерывания события FX** 1 - включено 0 - маскировано	1'h0
8	Чтение/запись	**CINT_IE - прерывание карты - включение прерывания** 1 - включено 0 - маскировано	1'h0
7	Чтение/запись	**CR_IE - включение прерывания извлечения карты** 1 - включено 0 - маскировано	1'h0
6	Чтение/запись	**CIN_IE - включение прерывания вставки карты** 1 - включено 0 - маскировано	1'h0
5	Чтение/запись	**BRR_IE - включение прерывания готовности буфера к чтению** 1 - включено 0 - маскировано	1'h0
4	Чтение/запись	**BWR_IE - включение прерывания готовности буфера к записи** 1 - включено 0 - маскировано	1'h0
3	Чтение/запись	**DMAINT_IE - включение прерывания DMA** 1 - включено 0 - маскировано	1'h0
2	Чтение/запись	**BGE_IE - включение прерывания события в промежутке между блоками** 1 - включено 0 - маскировано	1'h0
1	Чтение/запись	**TC_IE - включение прерывания завершения передачи** 1 - включено 0 - маскировано	1'h0
0	Чтение/запись	**CC_IE - включение прерывания завершения команды** 1 - включено 0 - маскировано	1'h0

1.3.9.10.3.5.42SRS15 (0x023c)

SRS15 - управление контроллером № 2 / состояние ошибки Auto CMD.

Таблица 1.476 – Поля регистра SRS15

Бит	Тип	Описание	Сброс
31	Чтение/запись	**PVE - включить предустановленное значение** При установке этого бита на 1 срабатывает автоматическое обновление регистров SRS11.SDCFSL, SRS11.SDCFSH, SRS11.CGS, SRS15.DSS контроллером. Значения для обновления устройство получает из SRS24 - SRS27, значения зависят от SRS15.UMS.	1'h0
29	Чтение/запись	**A64B - 64-битовая адресация** Указывает режим адресации для завершения DMA. Это поле игнорируется, если SRS15.HV4E=0. 0 - 32-битовая адресация 1 - 64-битовая адресация	1'h0
28	Чтение/запись	**HV4E - включение контроллера версии 4.00** Выбор режима совместимости с прошлыми версиями (контроллер SD версии 3.00) или режим контроллера SD версии 4.00. – версия 3.00 – версия 4.00 Программа может выбрать регистр адреса системы SRS00 (если этот бит установлен на 0) или SRS23 / SRS22 (если этот бит установлен на 1) для движка SDMA.	1'h0
27	Чтение/запись	**CMD23E - включить CMD23** Во время идентификации карты драйвер контроллера устанавливает этот бит на 1, если карта поддерживает CMD23 (SCR[33]==1).	1'h0
26	Чтение/запись	**ADMA2LM - режим длины ADMA2** Этот бит выбирает либо 16-битовую, либо 26-битовую длину ADMA2. 0 - 16-битовая длина данных 1 - 26-битовая длина данных	1'h0
24	Чтение/запись	**LVSIEXEC - выполнение идентификации LVS** При установке этого поля на 1 генерируется один импульсный сигнал на выходе SDCLK. Этот бит не меняется, пока DAT[2] остается низким. После обнаружения высокого DAT[2] значение этого поля автоматически становится 0, подтверждая завершение последовательности сигналов низкого напряжения.	1'h0
23	W0C	**SCS - выбор тактового генератора для проверки** Контроллер обновляет этот бит после завершения настройки. Если этот бит установлен на 1, процедура настройки успешно завершена. В противном случае процедура не выполняется, и логика настройки тактового генератора отключается. Этот бит действителен только после завершения процедуры. Запись 1 игнорируется. Запись 0 сбросит и отключит блок настройки.	1'h0

Бит	Тип	Описание	Сброс
22	Чтение/запись	**EXTNG - выполнение настройки** Этот регистр управляет процедурой настройки. Процедура запускается, когда бит устанавливается на 1. Процедуру можно отменить при очистке бита. Бит составляет 1 во время выполнения процедуры и 0 после завершения процедуры. SCS = 0, EXTNG = 0 - сброс и отключение логики настройки тактового генератора SCS = 0, EXTNG = 1 - сброс и перезапуск настройки SCS = 1, EXTNG = 0 - остановка настройки SCS = 1, EXTNG = 1 - запуск повторной настройки (без сброса логики настройки тактового генератора)	1'h0
21-20	Чтение/запись	**DSS - выбор мощности драйвера** Этот бит управляет электрическими параметрами драйвера ввода-вывода через выход *sdphy_dfi_drvss* ядра контроллера SD/eMMC). Можно реализовать до 4 конфигураций драйвера ввода-вывода: – 00 - драйвер типа B (по умолчанию) – 01 - драйвер типа A – 10 - драйвер типа C – 11 - драйвер типа D Этот бит игнорируется при сбросе V18SE.	2'h0
19	Чтение/запись	**V1SSE - включение сигналов 1,8 В** Этот бит управляет уровнем напряжения входных и выходных сигналов. Если бит установлен на 0 или 1, блок ввода-вывода использует сигналы 3,3 В или 1,8 В соответственно. Драйвер SW установит этот бит на 1 в режиме UHS-I. В зависимости от выбранного режима интерфейса SD программ установит это поле следующим образом: – 0 - скорость по умолчанию, высокая скорость – 1 - режим UHS-I	1'h0
18-16	Чтение/запись	**UMS - выбор режима UHS** Используется для выбора одного из режимов UHS-I. 000b - SDR12 001b - SDR25 010b - SDR50 011b - SDR104 100b - DDR50 101b - зарезервировано 110b - зарезервировано 111b - зарезервировано Выбранный режим UHS-I (когда значение находится в диапазоне 000b-100b) будет игнорироваться, если V18SE установлен на 0.	3'h0
7	R	**CNIACE - команда не передана по ошибке Auto CMD12** 1 указывает, что команда не была выполнена контроллером из-за ошибки предыдущей Auto CMD12. Если контроллер обнаруживает ошибку во время Auto-CMD12, все дальнейшие попытки генерирования команд блокируются. Для устранения ошибки потребуется программный сброс. Бит устанавливается на 0 при обнаружении ошибки Auto CMD23 (устанавливается любой из битов SRS15[4:1]).	1'h0

Бит	Тип	Описание	Сброс
5	R	**ACRE - ошибка ответа Auto CMD** 1 означает обнаружение ошибки в ответе на автоматическую команду.	1'h0
4	R	**ACIE - ошибка индекса Auto CMD** 1 означает ошибку индекса команды в ответе Auto CMD.	1'h0
3	R	**ACEBE - ошибка конечного бита Auto CMD** 1 указывает, что конечный бит ответа Auto-CMD равен 0.	1'h0
2	R	**ACCE - ошибка CRC Auto CMD** 1 означает обнаружение ошибки CRC в ответе Auto CMD или конфликта на линиях CMD: ACCE(SRS15.2) = 0, ACTE(SRS15.1) = 0 - нет ошибки ACCE(SRS15.2) = 0, ACTE(SRS15.1) = 1 - обнаружена ошибка тайм-аута Auto CMD ACCE(SRS15.2) = 1, ACTE(SRS15.1) = 0 - обнаружена ошибка CRC Auto CMD ACCE(SRS15.2) = 1, ACTE(SRS15.1) = 1 - обнаружен конфликт на линии CMD	1'h0
1	R	**ACTE - ошибка тайм-аута Auto CMD** 1 указывает отсутствие ответа в течение 64 тактовых циклов SDCLK от конечного бита Auto CMD или конфликт на линиях CMD (описание поля SRS15.ACCE см. в таблице). Если этот бит установлен на 1, другие биты состояния ошибки (SRS15[4:2]) игнорируются.	1'h0
0	R	**ACNE - Auto CMD12 не выполнена** 1 означает, что контроллер не может подать Auto CMD12 из-за какой-то ошибки. Если этот бит установлен на 1, другие биты состояния ошибки (SRS15[4:1]) игнорируются. Бит устанавливается на 0 при обнаружении ошибки Auto CMD23 (устанавливается любой из битов SRS15[4:1]).	1'h0

1.3.9.10.3.5.43SRS16 (0x0240)

SRS16 - характеристики №1.

Таблица 1.477 – Поля регистра SRS16

Бит	Тип	Описание	Сброс
31-30	R	**SLT - тип слота** Эти биты указывают тип слота. – 00 - съемный слот для карты – 01 - встроенный слот для одного устройства – 10 - слот общей шины – 11 - зарезервировано	2'h0
29	R	**AIS - поддержка асинхронного прерывания** 0 - не поддерживается	1'h0
28	R	**Аб4SV3 - поддержка б4-битовой адресации системы** 0 - 64-битовая адресация для V3 не поддерживается 1 - 64-битовая адресация для V3 поддерживается	1'h1
27	R	**Аб4SV4 - поддержка б4-битовой адресации системы версией 4** 0 - 64-битовая адресация для V4 не поддерживается 1 - 64-битовая адресация для V4 поддерживается	1'h1

Бит	Тип	Описание	Сброс
26	R	**VS1S - поддержка напряжения 1,8 В** 0 - не поддерживается 1 - поддерживается	1'h1
25	R	**VS30 - поддержка напряжения 3,0 В** 0 - не поддерживается 1 - поддерживается	1'h1
24	R	**VS33 - поддержка напряжения 3,3 В** 0 - не поддерживается 1 - поддерживается	1'h1
23	R	**SRS - поддержка приостановки / возобновления** 0 - не поддерживается 1 - поддерживается Контроллер не поддерживает механизм приостановки / возобновления.	1'h0
22	R	**DMAS - поддержка SDMA** 0 - не поддерживается 1 - поддерживается Этот бит определяет поддержку SDMA.	1'h1
21	R	**HSS - поддержка высокой скорости** 0 - не поддерживается 1 - поддерживается	1'h1
20	R	**ADMA1S - поддержка ADMA1** 0 - не поддерживается 1 - поддерживается	1'h0
19	R	**ADMA2S - поддержка ADMA2** 0 - не поддерживается 1 - поддерживается	1'h1
18	R	**SEDS - поддержка 8-битового встроенного устройства** 0 - не поддерживается 1 - поддерживается Если этот бит установлен на 0, регистр SRS10.EDTW не выполняется.	1'h0
17-16	R	**SRS16.MBL - макс. длина блока** Это значение указывает максимальный размер блока, передаваемого контроллером. Можно задать один из трех размеров, указанных ниже: – 00b - 512 байтов – 01b - 1024 байта – 10b - 2048 байтов – 11b - зарезервировано Размер физического буфера FIFO определяется отдельным параметром FIFODEPTH, размер физического буфера равен 2лFIFODEPTH * 8 байтов. Следовательно, максимальный размер блока, определяемый MBL, всегда будет меньше или равен размеру физического буфера.	2'h2

Бит	Тип	Описание	Сброс
15-8	R	**BCSDCLK - базовая тактовая частота для тактового генератора SD** Поле определяет базовую тактовую частоту для тактового генератора SD в единицах 1 МГц. Базовый тактовый сигнал - тактовый сигнал, подаваемый на контакт sdmclk контроллера. Максимальная поддерживаемая тактовая частота - от 10 МГц до 255 МГц. Если BCSDCLK = 0, система должно получить информацию о тактовых сигналах другим методом (т. е. не указанным в спецификации).	8'hc8
7	R	**TCU - единица тактовой частоты тайм-аута** Поле задает единицу частоты для SRS16.TCF. 0 - кГц 1 - МГц	1'h1
5-0	R	**TCF - тактовая частота тайм-аута** Определяет базовую тактовую частоту для обнаружения ошибки тайм-аута данных. Бит SRS16.TCU определяет используемую единицу. 111111b - 63 кГц (SRS16.TCU=0) или 63 МГц (SRS16.TCU=1) 111110b - 63 кГц (SRS16.TCU=0) или 63 МГц (SRS16.TCU=1) … 000001b - 1 кГц (SRS16.TCU=0) или 1 МГц (SRS16.TCU=1) 000000b - система должно получить информацию о тактовых сигналах другим методом (т. е. не указанным в спецификации).	6'h32

1.3.9.10.3.5.44SRS17 (0x0244)

SRS17 - характеристики № 2.

Таблица 1.478 – Поля регистра SRS17

Бит	Тип	Описание	Сброс
31	R	**LVSH - контроллер с сигналами низкого напряжения** – 1 - контроллер с LVS – 0 - контроллер с не LVS	1'h1
28	R	**VDD2S - поддержка VDD2** – 1 - VDD2 поддерживается – 0 - VDD2 не поддерживается	1'h0
27	R	**ADMA3SUP - поддержка ADMA3** – 1 - ADMA3 поддерживается – 0 - ADMA3 не поддерживается	1'h1
23-16	R	**CLKMPR - умножитель тактовых сигналов** Это поле должно быть установлено на 0 (постоянно), так как умножитель тактовых сигналов не поддерживается.	8'h0

Бит	Тип	Описание	Сброс
15-14	R	**RTNGM - режимы повторной настройки** В зависимости от метода возврата между процедурами повторной настройки на длину данных распространяются некоторые ограничения. Ядро может работать с поддержкой одного из трех методов: – 0 - режим 1: программный драйвер использует таймер для расчета возврата операции повторной настройки. Длина данных между операциями ограничена 4 Мб. – 1 - режим 2: драйвер использует либо запрос повторной настройки (для этой цели используется внешний входной контакт uhsi_retune_req), либо таймер для определения времени выполнения следующей операции повторной настройки. Длина данных между операциями ограничена 4 Мб. – 2 - режим 3: этот режим аналогичен режиму 2 с одним исключением. Ядро может выполнять автоматическую настройку во время передачи данных, поэтому нет ограничения по длине данных. Режим 3 в настоящий момент не поддерживается. Драйвер может настроить любой таймер с помощью RTNGCNT. Это поле должно быть установлено на 0 или 1, так как режим 3 не поддерживается.	2'h0
13	R	**UTSM50 - настройка использования SDR50** – 1 - настройка необходима в режиме SDR50 – 0 - настройка не нужна в режиме SDR50	1'h0
11-8	R	**RTNGCNT - счетчик-таймер для повторной настройки** Эти биты содержат исходное значение для таймера, используемого для периодического запуска повторной настройки. – 0h - таймер повторной настройки отключен – 1h - 1 секунда – … – n - 2л(n-1) секунд – … – Bh - 1024 секунды – Eh-Ch - зарезервировано – Fh - получение этой информации другим способом	4'h0
6	R	**DRVD - поддержка драйвера линии 1,8 В типа D** – 1 - драйвер типа D поддерживается – 0 - драйвер типа D не поддерживается	1'h1
5	R	**DRVC - поддержка драйвера линии 1,8В типа C** – 1 - драйвер типа C поддерживается – 0 - драйвер типа C не поддерживается	1'h1
4	R	**DRVA - поддержка драйвера линии 1,8В типа A** – 1 - драйвер типа A поддерживается – 0 - драйвер типа A не поддерживается	1'h1
3	R	**UHSII - поддержка UHS-II / UHS-III** 0 - не поддерживается UHS-II	1'h0
2	R	**DDR50 - поддержка DDR50** – 1 - режим DDR50 поддерживается – 0 - режим DDR50 не поддерживается	1'h1

Бит	Тип	Описание	Сброс
1	R	**SDR104 - поддержка SDR104** – 1 - режим SDR104 поддерживается – 0 - режим SDR104 не поддерживается	1'h1
0	R	**SDR50 - поддержка SDR50** – 1 - режим SDR50 поддерживается – 0 - режим SDR50 не поддерживается	1'h1

1.3.9.10.3.5.45 SRS18 (0x0248)

SRS18 - характеристики № 3.

Таблица 1.479 – Поля регистра SRS18

Бит	Тип	Описание	Сброс
23-16	R	**MC1S - максимальный ток для 1,8В** – 0 - система должна получить значение другим методом – 1 - 4 мА – 2 - 8 мА – 3 - 12 мА – … – 255 - 1020 мА	8'h20
15-8	R	**MC30 - максимальный ток для 3,0В** – 0 - система должна получить значение другим методом – 1 - 4 мА – 2 - 8 мА – 3 - 12 мА – … – 255 - 1020 мА	8'h20
7-0	R	**MC33 - максимальный ток для 3,3 В** – 0 - система должна получить значение другим методом – 1 - 4 мА – 2 - 8 мА – 3 - 12 мА – … – 255 - 1020 мА	8'h20

1.3.9.10.3.5.46 SRS19 (0x024c)

SRS19 - характеристики № 4.

Таблица 1.480 – Поля регистра SRS19

Бит	Тип	Описание	Сброс
7-0	R	**MC1SV2 - максимальный ток для 1,8В VDD2** – 0 - система должна получить значение другим методом – 1 - 4 мА – 2 - 8 мА – 3 - 12 мА – … – 255 - 1020 мА	8’h20

1.3.9.10.3.5.47 SRS20 (0x0250)

SRS20 - принудительное событие.

Каждое поле этого регистра относится к определенному состоянию ошибки. Запись 1 в поле устанавливает состояние ошибки. Эта функция предусмотрена для отладки ПО.

Таблица 1.481 – Поля регистра SRS20

Бит	Тип	Описание	Сброс
27	W	**ERESP_FE - принудительное событие ошибки ответа**	1’h0
26	W	**ETUNE_FE - принудительное событие ошибки настройки**	1’h0
25	W	**EADMA_FE - принудительное событие ошибки ADMA**	1’h0
24	W	**EAC_FE - принудительное событие ошибки Auto CMD**	1’h0
23	W	**ECL_FE - принудительное событие ошибки текущего предела**	1’h0
22	W	**EDEB_FE - принудительное событие ошибки конечного бита данных**	1’h0
21	W	**EDCRC_FE - принудительное событие ошибки CRC данных**	1’h0
20	W	**EDT_FE - принудительное событие ошибки тайм-аута данных**	1’h0
19	W	**ECI_FE - принудительное событие ошибки индекса команды**	1’h0
18	W	**ECEB_FE - принудительное событие ошибки конечного бита команды**	1’h0
17	W	**ECCRC_FE - принудительное событие ошибки CRC команды**	1’h0
16	W	**ECT_FE - принудительное событие ошибки тайм-аута команды**	1’h0
7	W	**CNIACE_FE - принудительное событие отсутствия подачи команды по ошибке Auto CMD12**	1’h0
4	W	**ACIE_FE - принудительное событие ошибки индекса Auto CMD**	1’h0
3	W	**ACEBE_FE - принудительное событие ошибки конечного бита Auto CMD**	1’h0
2	W	**ACCE_FE - принудительное событие ошибки CRC Auto CMD**	1’h0
1	W	**ACTE_FE - принудительное событие ошибки тайм-аута Auto CMD**	1’h0
0	W	**ACNE_FE - принудительное событие невыполнения Auto CMD12**	1’h0

1.3.9.10.3.5.48 SRS21 (0x0254)

SRS21 - состояние ошибки ADMA.

Таблица 1.482 – Поля регистра SRS21

Бит	Тип	Описание	Сброс
2	R	**EADMAL - ошибка несоответствия дины ADMA** Этот бит устанавливается, когда: – общая длина данных, указанная в дескрипторах ADMA, отличается от значений, указанных в полях числа блоков и длины блока (если установлен бит счетчика блоков). – общая длина данных не может быть разделена на целые блоки указанной длины (если установлен бит включения счетчика блоков).	1'h0
1-0	R	**EADMAS - состояние ошибки ADMA** Значение в этом поле означает состояние ADMA. Возможные значения: 00b - ST_STOP (остановка ADMA) 01b - ST_FDS (дескриптор выборки) 10b - не используется 11b - ST_TRF (передача данных)	2'h0

1.3.9.10.3.5.49 SRS22 (0x0258)

Адрес системы ADMA/SDMA SRS22 1.

Таблица 1.483 – Поля регистра SRS22

Бит	Тип	Описание	Сброс
31-0	Чтение/запись	**DMASA1 - адрес системы ADMA** Это поле содержит физический адрес текущего обрабатываемого дескриптора ADMA или адрес системы SDMA. Драйвер хоста устанавливает этот регистр с базовым адресом таблицы дескрипторов перед началом передач ADMA. Драйвер хоста не должен записывать этот регистр во время передачи данных. При обработке движком ADMA списка дескрипторов значение ADMASA всегда увеличивается для указания следующего обрабатываемого дескриптора. В случае возникновения ошибки ADMA регистр сохраняет адрес дескриптора в зависимости от значения регистра состояния ошибки ADMA (SRS21.EADMAS) в соответствии с таблицей ниже: 00b - указывает следующий ошибочный дескриптор 01b - указывает ошибочный дескриптор 10b - не используется 11b - указывает следующий ошибочный дескриптор Движок ADMA хоста игнорирует 2 или 3 младших бита в этом регистре в случае 32-битовой или 64-битовой адресации соответственно. Если SRS15.HV4E установлен на 1, и выбран движок SDMA, для задания адреса памяти системы используется это поле вместо SRS00. Этот регистр увеличивается и указывает следующую доступную ячейку памяти.	32'h0

1.3.9.10.3.5.50 SRS23 (0x025c)

Адрес системы ADMA/SDMA SRS23 2.

Таблица 1.484 – Поля регистра SRS23

Бит	Тип	Описание	Сброс
31-0	Чтение/запись	**DMASA2 - адрес ADMA системы № 2** В режиме ADMA если включена 64-битовая адресация (SRS15.A64B=1), это поле сохраняет биты 63-32 физического адреса таблицы дескрипторов ADMA. В режиме SDMA, если хост совместим с версией 4.x, и включена 64-битовая адресация (SRS15.HV4E=1 и SRS15.A64B=1), это поле сохраняет биты 63-32 адреса системы.	32’h0

1.3.9.10.3.5.51SRS24 (0x0260)

SRS24 - предустановленное значение (скорость по умолчанию).

SRS24[31:16] - скорость по умолчанию, если:

– SRS15.V18SE=0;

– SRS10.HSE=0.

Регистр HWINIT. Этот регистр инициализируется аппаратным обеспечением после сброса, и считываемое значение будет соответствовать IP-конфигурации.

Таблица 1.485 – Поля регистра SRS24

Бит	Тип	Описание	Сброс
31-30	R	**DSSPV## - выбор мощности драйвера - предустановленное значение** Это поле может использоваться программным обеспечением для обновления SRS15.DSS.	2’h0
25-16	R	**SDCFSPV## - выбор тактовой частоты SDCLK - предустановленное значение** Это поле может использоваться программным обеспечением для обновления SRS11.SDCFSH и SRS11.SDCFSL.	10’h4

1.3.9.10.3.5.52SRS25 (0x0264)

SRS25 - предустановленное значение (высокая скорость и SDR12)

SRS25[15:0] - высокая скорость, если:

SRS15.V18SE=0

SRS.HSE=1

SRS25[31:16] - SDR12, если:

SRS15.V18SE=1

SRS15.UMS=000b

Регистр HWINIT. Необходимо отметить, что этот регистр инициализируется аппаратным обеспечением после сброса, и считываемое значение будет соответствовать IP-конфигурации.

Таблица 1.486 – Поля регистра SRS25

Бит	Тип	Описание	Сброс
31-30	R	**DSSPV## - выбор мощности драйвера - предустановленное значение** Это поле может использоваться программным обеспечением для обновления SRS15.DSS.	2’h0
25-16	R	**SDCFSPV## - выбор тактовой частоты SDCLK - предустановленное значение** Это поле может использоваться программным обеспечением для обновления SRS11.SDCFSH и SRS11.SDCFSL.	10’h4
15-14	R	**DSSPV## - выбор мощности драйвера - предустановленное значение** Это поле может использоваться программным обеспечением для обновления SRS15.DSS.	2’h0
9-0	R	**SDCFSPV## - выбор тактовой частоты SDCLK - предустановленное значение** Это поле может использоваться программным обеспечением для обновления SRS11.SDCFSH и SRS11.SDCFSL.	10’h2

1.3.9.10.3.5.53SRS26 (0x0268)

SRS26 - предустановленное значение (SDR25 и SDR50)

SRS26[15:0] - SDR25, если:

– SRS15.V18SE=1;

– SRS15.UMS=001b.

SRS26[31:16] - SDR50, если:

– SRS15.V18SE=1;

– SRS15.UMS=010b.

Регистр HWINIT. Необходимо отметить, что этот регистр инициализируется аппаратным обеспечением после сброса, и считываемое значение будет соответствовать IP-конфигурации.

Таблица 1.487 – Поля регистра SRS26

Бит	Тип	Описание	Сброс
31-30	R	**DSSPV## - выбор мощности драйвера - предустановленное значение** Это поле может использоваться программным обеспечением для обновления SRS15.DSS.	2’h0
25-16	R	**SDCFSPV## - выбор тактовой частоты SDCLK - предустановленное значение** Это поле может использоваться программным обеспечением для обновления SRS11.SDCFSH и SRS11.SDCFSL.	10’h1
15-14	R	**DSSPV## - выбор мощности драйвера - предустановленное значение** Это поле может использоваться программным обеспечением для обновления SRS15.DSS.	2’h0

Бит	Тип	Описание	Сброс
10	R	**CGSPV## - выбор тактового генератора - предустановленное значение** Это поле может использоваться программным обеспечением для обновления SRS11.DSS.	1’h0
9-0	R	**SDCFSPV## - выбор тактовой частоты SDCLK - предустановленное значение** Это поле может использоваться программным обеспечением для обновления SRS11.SDCFSH и SRS11.SDCFSL.	10’h2

1.3.9.10.3.5.54SRS27 (0x026c)

SRS27 - предустановленное значение (SDR104 и DDR50)

SRS27[15:0] - SDR104, если:

– SRS15.V18SE=1;

– SRS15.UMS=011b.

SRS27[31:16] - DDR50, если:

– SRS15.V18SE=1;

– SRS15.UMS=100b.

Регистр HWINIT. Необходимо отметить, что этот регистр инициализируется аппаратным обеспечением после сброса, и считываемое значение будет соответствовать IP-конфигурации.

Таблица 1.488 – Поля регистра SRS27

Бит	Тип	Описание	Сброс
31-30	R	**DSSPV## - выбор мощности драйвера - предустановленное значение** Это поле может использоваться программным обеспечением для обновления SRS15.DSS.	2’h0
25-16	R	**SDCFSPV## - выбор тактовой частоты SDCLK - предустановленное значение** Это поле может использоваться программным обеспечением для обновления SRS11.SDCFSH и SRS11.SDCFSL.	10’h2
15-14	R	**DSSPV## - выбор мощности драйвера - предустановленное значение** Это поле может использоваться программным обеспечением для обновления SRS15.DSS.	2’h0
9-0	R	**SDCFSPV## - выбор тактовой частоты SDCLK - предустановленное значение** Это поле может использоваться программным обеспечением для обновления SRS11.SDCFSH и SRS11.SDCFSL.	10’h0

1.3.9.10.3.5.55SRS30 (0x0278)

SRS30 идентификационный адрес ADMA3 1.

Таблица 1.489 – Поля регистра SRS30

Бит	Тип	Описание	Сброс
31-0	Чтение/запись	**Адрес встроенного дескриптора ADMA3 № 1** Это поле содержит физический адрес текущего обрабатываемого встроенного дескриптора ADMA3. Драйвер хоста устанавливает этот регистр с базовым адресом идентификатора таблицы дескрипторов перед началом передач ADMA3. Драйвер хоста не должен записывать этот регистр во время передачи данных. При обработке движком ADMA3 списка дескрипторов значение идентификатора ADMA3 всегда увеличивается для указания следующего обрабатываемого идентификатора. Движок ADMA3 хоста игнорирует 2 или 3 младших бита в этом регистре в случае 32-битовой или 64-битовой адресации соответственно. Если ADMA3 использует 32-битовую адресацию, запись в этом регистре запускает ADMA3.	32’h0

1.3.9.10.3.5.56 SRS31 (0x027c)

SRS30 - идентификационный адрес ADMA3 2.

Таблица 1.490 – Поля регистра SRS31

Бит	Тип	Описание	Сброс
31-0	Чтение/запись	**Адрес встроенного дескриптора ADMA3 № 2** Если включена 64-битовая адресация (SRS15.A64B=1), это поле сохраняет биты 63-32 физического адреса таблицы встроенных дескрипторов ADMA3. Если ADMA3 использует 64-битовую адресацию, запись в этом регистре запускает ADMA3.	32’h0

1.3.9.10.3.5.57 CRS63 (0x02fc)

CRS63 - версия контроллера/состояние прерывания слота.

Таблица 1.491 – Поля регистра CRS63

Бит	Тип	Описание	Сброс
23-16	R	**SVN - номер версии спецификации.** В этом поле указана версия спецификации контроллера. – 0h - версия 1.00 – 1h - версия 2.00 – 2h - версия 3.00 – 3h - версия 4.00 – 4h - версия 4.10 – 5h - версия 4.20 – 6h-FFh - зарезервировано	8’h5
0	R	**ISES - сигнал прерывания для каждого слота.** Это поле указывает активацию сигнала прерывания или срабатывания.	1’h0

1.3.9.10.3.5.58 CQRS00 (0x0400)

CQRS00 - версия функции организации очереди команд.

Таблица 1.492 – Поля регистра CQRS00

Бит	Тип	Описание	Сброс
11-8	R	**CQVN1 - номер версии eMMC** Номер версии поддерживаемого стандарта eMMC - 5.	4’h5
7-4	R	**CQVN2 - дополнительный номер версии eMMC** Дополнительный номер версии поддерживаемого стандарта eMMC - 1.	4’h1
3-0	R	**CQVN3 - суффикс версии eMMC** Суффикс версии поддерживаемого стандарта eMMC - 0.	4’h0

1.3.9.10.3.5.59 CQRS01 (0x0404)

CQRS01 - характеристики организации очереди команд.

Таблица 1.493 – Поля регистра CQRS01

Бит	Тип	Описание	Сброс
15-12	R	**ITCFMUL - умножитель тактовой частоты внутреннего таймера (ITCFMUL)** Определяет умножитель внутренней тактовой частоты таймера объединения и периода опроса SQS. 0 - 0,001 МГц 1 - 0,01 МГц 2 - 0,1 МГц 3 - 1 МГц 4 - 10 МГц ITCFMUL и ITCFVAL определяют тактовую частоту.	4’h0
9-0	R	**ITCFVAL - тактовая частота внутреннего таймера (ITCFVAL)** Значение определяет внутреннюю тактовую частоту таймера объединения и периода опроса SQS. Частота равна ITCFMUL * ITCFVAL.	10’h0

1.3.9.10.3.5.60 CQRS02 (0x0408)

CQRS02 - конфигурация организации очереди команд.

Таблица 1.494 – Поля регистра CQRS02

Бит	Тип	Описание	Сброс
12	Чтение/запись	**CQDCE - включение прямой команды (DCMD)** Обработка дескриптора задачи для слота 31 в качестве дескриптора задачи передачи данных (0) или дескриптора задачи прямой команды (1).	1’h0
8	Чтение/запись	**CQTDS - дескриптор задачи** Дескриптор задачи расчетного значения 128 битов (1) или 64 битов (0). Этот параметр может измениться только после отключения организации очереди команд (CQE = 0).	1’h0

Бит	Тип	Описание	Сброс
0	Чтение/запись	**CQE - включение организации очереди команд** Включает (1) или отключает (0) организацию очереди команд. Этот бит может быть активирован только после завершения всех предыдущих транзакций. Этот бит может быть сброшен после завершения и сброса всех задач. При установке этого бита на 1 SRS15.HV4E автоматически устанавливается на 1.	1'h0

1.3.9.10.3.5.61 CQRS03 (0x040c)

CQRS03 - управление организацией очереди команд.

Таблица 1.495 – Поля регистра CQRS03

Бит	Тип	Описание	Сброс
8	W1S	**CQCAT - сброс всех задач** Сброс (1) всех активных задач в контроллере. Программное обеспечение должно опросить этот регистр до завершения операции (бит автоматически очищается). Программа может установить этот бит, только если заблокирован движок CQ. Программа должна сбросить все запрошенные задачи в устройстве eMMC. Запись (0) не оказывает влияния.	1'h0
0	Чтение/запись	**CQHLT - блокировка CQ** Движок может быть заблокирован путем установки этого бита на 1. Любая выполняющаяся операция будет завершена, а ожидающая операция - остановлена. После завершения или остановки всех задач этот бит будет установлен на 1. Контроллер не запустит автоматически новую операцию, но программа может использовать регистры SRS для подачи любой команды в обход CQE. Движок CQ запускает операцию после блокировки путем записи 0 в этот регистр. Запись 0 игнорируется, если движок CQ не заблокирован.	1'h0

1.3.9.10.3.5.62 CQRS04 (0x0410)

CQRS04 - состояние прерывания организации очереди команд.

Этот регистр имеет несколько битов состояний, касающихся определенных прерываний. В случае возникновения прерывания при включенном состоянии прерывания организации очереди команд бит состояния будет установлен на 1. Биты могут быть сброшены программным обеспечением.

Запись 0 сбрасывает бит. Запись 1 будет проигнорирована.

Таблица 1.496 – Поля регистра CQRS04

Бит	Тип	Описание	Сброс
3	W1C	**CQTCL - задача сброшена (TCL)** После завершения операции сброса задач или отдельной задачи CQE установит эти на 1.	1'h0
2	W1C	**CQREDI - прерывание обнаружения ошибки ответа (RED)** В случае обнаружения ошибки в ответе, полученном от eMMC CQE устанавливает этот бит на 1. ПО может выбрать биты для анализа путем выбора CQRMEM.	1'h0

Бит	Тип	Описание	Сброс
1	W1C	**CQTCC - прерывание завершения задачи (TCC)** CQE устанавливает этот бит, когда или задача INT=1 завершена, или регистр объединения прерываний сообщает о прерывании.	1'h0
0	W1C	**CQHAC - прерывание выполнения блокировки (HAC)** CQE устанавливает этот бит, если значение CQHLT меняется с 0 на 1.	1'h0

1.3.9.10.3.5.63 CQRS05 (0x0414)

CQRS05 - включение состояния прерывания организации очереди команд.

Биты включения состояния включают источники прерываний. Состояние включается, если бит установлен на 1 (ПО записывает 1 в это поле).

Таблица 1.497 – Поля регистра CQRS05

Бит	Тип	Описание	Сброс
3	Чтение/запись	**CQTCLST - включение состояния сброса задачи (TCL)** Включение регистра CQTCL.	1'h0
2	Чтение/запись	**CQREDST - включение состояния обнаружения ошибки ответа (RED)** Включение регистра CQREDI.	1'h0
1	Чтение/запись	**CQTCCST - включение состояния завершения задачи (TCC)** Включение регистра CQHAC.	1'h0
0	Чтение/запись	**CQHACST - включение состояния выполнения блокировки (HAC)** Включение регистра CQTCLST.	1'h0

1.3.9.10.3.5.64 CQRS06 (0x0418)

CQRS06 - включение сигнала прерывания организации очереди команд.

Этот регистр позволяет включить или отключить уведомление о прерывании отдельно для каждого бита состояния прерывания организации очереди команд. Если бит состояния прерывания установлен на 1, и установлено соответствующее поле в этом регистре (ПО записывает 1 в это поле), состояние прерывания передается на порт прерывания.

Таблица 1.498 – Поля регистра CQRS06

Бит	Тип	Описание	Сброс
3	Чтение/запись	**CQTCLSI - включение сигнала сброса задачи (TCL)** Включение сигнала прерывания от регистра CQTCL.	1'h0
2	Чтение/запись	**CQREDSI - включение сигнала обнаружения ошибки ответа (TCC)** Включение сигнала прерывания от регистра CQREDI.	1'h0
1	Чтение/запись	**CQTCCSI - включение сигнала завершения задачи (TCC)** Включение сигнала прерывания от регистра CQTCC.	1'h0
0	Чтение/запись	**CQHACSI - включение сигнала выполнения блокировки (HAC)** Включение сигнала прерывания от регистра CQHLT.	1'h0

1.3.9.10.3.5.65 CQRS07 (0x041c)

CQRS07 - объединение прерываний.

Этот регистр позволяет группировать передачи CQ и сообщать об одном прерывании всей группы запрошенных задач.

Таблица 1.499 – Поля регистра CQRS07

Бит	Тип	Описание	Сброс
31	Чтение/запись	**CQICED - включение/выключение объединения прерываний** Включает механизм объединения для создания групповых прерываний.	1'h0
20	R	**CQICSB - бит состояния объединения прерываний (ICSB)** CQE устанавливает этот бит на 1 после завершения задачи с INT=0.	1'h0
16	W	**CQICCTR - сброс счетчика и таймера (ICCTR)** ПО сбрасывает таймер объединения и счетчик.	1'h0
15	W	**CQICCTHWEN - включение записи порога счетчика объединения прерываний (ICCTH-WEN)** Эта запись включает CQICCTH. Если этот бит установлен на 1, поле обновляется.	1'h0
12-8	Чтение/запись	**CQICCTH - порог счетчика объединения прерываний (ICCTH)** CQE увеличивает значение внутреннего счетчика после завершения задачи с INT=0. Если внутренний счетчик достигает этого значения, счетчик объединения генерирует прерывание. ПО может выбрать пороговое значение от 1 до 31. ПО может отключить внутренний счетчик и генерирование прерываний, установив это поле на 0.	5'h0
7	W	**CQICTOVALEN - включение записи порогового значения тайм-аута объединения прерываний (ICTOVAL- WEN)** Эта запись включает CQICTOVAL. Если этот бит установлен на 1, поле обновляется.	1'h0
6-0	Чтение/запись	**CQICTOVAL - значение тайм-аута объединения прерываний (ICTOVAL)** CQE генерирует прерывание при достижении внутренним счетчиком периода, указанного в этом поле. Счетчик запускается после завершения первой передачи с INT=0. Значение счетчика увеличивается каждый раз при истечения периода внутреннего тактового сигнала * 1024. ПО может отключить этот таймер, установив это поле на 0.	7'h0

1.3.9.10.3.5.66 CQRS08 (0x0420)

CQRS08 - базовый адрес списка дескрипторов задачи организации очереди команд.

Таблица 1.500 – Поля регистра CQRS08

Бит	Тип	Описание	Сброс
31-0	Чтение/запись	**CQTDLBA - базовый адрес списка дескрипторов (младший)** Базовый адрес (32 младших бита) списка дескрипторов задач. ПО записывает значения по границе 1 кб (10 меньших битов должны быть установлена на 0). Аппаратное обеспечение игнорирует 10 младших битов. ПО обновляет этот регистр, только если CQE отключен.	32'h0

1.3.9.10.3.5.67 CQRS09 (0x0424)

CQRS09 - базовый адрес списка дескрипторов задачи организации очереди команд (старшие 32 бита).

Таблица 1.501 – Поля регистра CQRS09

Бит	Тип	Описание	Сброс
31-0	Чтение/запись	**CQTDLBA - базовый адрес списка дескрипторов (старший)** Базовый адрес (32 старших бита) списка дескрипторов задач. Этот регистр не используется при 32-битовой адресации (ПО не меняет это значение). ПО обновляет этот регистр, только если CQE отключен.	32'h0

1.3.9.10.3.5.68 CQRS10 (0x0428)

CQRS10 - сигнальный регистр задач организации очереди команд.

CQ имеет 32 задач для отдельных битов для запуска операции по требуемой задачи. ПО записывает 1 в любую ячейку от 0 до 31 за запуска задачи от 0 до 31. ПО может запросить несколько задач в одной записи. Движок CQ обрабатывает задачи в порядке запроса:

– в случае запроса нескольких задач при одной записи регистра задача с меньшим номером имеет наибольший приоритет по сравнению с задачей с большим номером;

– задач(и), запрошенная(ые) при предыдущей записи регистра, имеет(ют) более высокий приоритет по сравнению с задачей(ами), запрошенной(ыми) при более поздней записи регистра.

Порядок задач поддерживается на всех этапах транзакции. Если определенная задача не готова к выполнению движок CQ выбирает следующую задачу с большим номером.

Движку CQ необходимо несколько тактовых циклов для постановки запрошенных в одной записи сигнального регистра задач в очередь. Интерфейс ведомого устройства завершает передачу данных записи после организации очереди всех задач.

Если ПО устанавливает бит на 0 в этом регистре, соответствующая задача не будет выполнена, это значение будет проигнорировано.

Сигнальный бит задачи будет 1 до завершения задачи, сброса задачи битом сброса всех задач или битом сброса задачи с этим номером или отключения движка CQ (CQE=0).

Таблица 1.502 – Поля регистра CQRS10

Бит	Тип	Описание	Сброс
31:0	W1S	**CQTD31 - сигнальные биты задачи организации очереди команд №№31-00**	32'h0

1.3.9.10.3.5.69 CQRS11 (0x042c)

CQRS11 - уведомление о выполнении задачи.

32 бита соответствуют 32 задачам. При выполнении задачи N бит N устанавливается на 1. Установленный на 1 бит может быть сброшен путем записи 1 в этот бит.

Таблица 1.503 – Поля регистра CQRS11

Бит	Тип	Описание	Сброс
31:0	W1C	**CQTCN31 - уведомление о выполнении задачи №№ 31-00**	32'h0

1.3.9.10.3.5.70 CQRS12 (0x0430)

CQRS12 - состояние очереди устройства.

Таблица 1.504 – Поля регистра CQRS12

Бит	Тип	Описание	Сброс
31-0	R	**CQDQS - состояние очереди устройства** Этот регистр указывает состояние устройства eMMC. Задача N готова к выполнению при установке бита N в этом регистре на 1. Этот регистр обновляется каждый раз при получении ответа для SEND_QUEUE_STATUS (CMD13).	32'h0

1.3.9.10.3.5.71 CQRS13 (0x0434)

CQRS13 - ожидающие задачи устройства.

Этот регистр содержит информацию о задачах, передаваемых eMMC (CMD44 и CMD45), но не выполняемых. Задача N передается, но не выполняется, если бит N установлен на 1. Бит N сбрасывается после завершения задачи N.

Таблица 1.505 – Поля регистра CQRS13

Бит	Тип	Описание	Сброс
31:0	R	**CQDPT31 - ожидающие задачи устройства №№31-00**	32'h0

1.3.9.10.3.5.72 CQRS14 (0x0438)

CQRS14 - сброс задачи.

ПО устанавливает бит N на 1 в этом регистре для сброса задачи N. Бит остается равен 1 до завершения операции сброса. После завершения операции CQE сбрасывает этот бит на 0. ПО проверяет, заблокирован ли движок CQ, перед сбросом задач. ПО может сбросить только одну задачу. При установке любого бита в этом регистре ПО не нужно запрашивать сброс новой задачи. Эта операция сбрасывает только задачи в контроллере. ПО должно сбросить задачи в устройстве.

Запись 0 в регистре игнорируется.

Таблица 1.506 – Поля регистра CQRS14

Бит	Тип	Описание	Сброс
31:0	W1S	**CQTC31 - сброс задач организации очереди команд №№31-00**	32’h0

1.3.9.10.3.5.73 CQRS16 (0x0440)

CQRS16 - конфигурация состояния отправки 1.

Таблица 1.507 – Поля регистра CQRS16

Бит	Тип	Описание	Сброс
19-16	Чтение/запись	**CQSSCBC - счетчик блоков команд состояния отправки (CBC)** ПО может определить, отправляет ли CQE команду SEND_QUEUE_STATUS (CMD13) во время передачи данных и когда. Если этот регистр установлен на 0, CQE не отправляет CMD13 во время передачи данных. Значение 1, 2 или N означает, что CQE передает CMD13 во время последней передачи, до последней передачи или до (N-1) до последнего блока соответственно. Разрешенный диапазон регистра - от 0 до 15.	4’h0
15-0	Чтение/запись	**CQSSCIT - таймер простоя команды состояния (CIT)** Если CQE не работает, контроллер может опросить устройство, передав SEND_QUEUE_STATUS (CMD13) с интервалом, указанным в этом регистре. Разрешенный диапазон регистра - от 1 до 65535. Интервал рассчитывается как внутренний тактовый период CQSSICT *.	16’h0

1.3.9.10.3.5.74 CQRS17 (0x0444)

CQRS17 - конфигурация состояния отправки 2.

Таблица 1.508 – Поля регистра CQRS17

Бит	Тип	Описание	Сброс
15-0	Чтение/запись	**CQSQSR - состояние очереди отправки RCA** ПО записывает 16-битовое значение RCA, которое отправляется в качестве аргумента в команде SEND_QUEUE_STATUS (CMD13).	16’h0

1.3.9.10.3.5.75 CQRS18 (0x0448)

CQRS18 - ответ команды для задачи прямой команды.

Таблица 1.509 – Поля регистра CQRS18

Бит	Тип	Описание	Сброс
31-0	R	**CQDCLR - последний ответ на прямую команду** CQE сохраняет ответ на последнюю команду DCMD.	32’h0

1.3.9.10.3.5.76 CQRS20 (0x0450)

CQRS20 - маска ошибки режима ответа.

Таблица 1.510 – Поля регистра CQRS20

Бит	Тип	Описание	Сброс
31-0	Чтение/запись	**CQRMEM - маска ошибки режима ответа** CQE может автоматически обнаруживать ошибки в ответе. ПО определяет проверяемые биты в ответе. Все биты, установленные на 1 (записанные ПО), анализируются. CQE сообщает о прерывании по обнаружению ошибки ответа (CQREDI), если бит N CQRMEM установлен на 1, и бит N в ответе установлен на 1. Ответ для SEND_QUEUE_STATUS (CMD13), автоматически отправляемый движком CQE, игнорируется.	32’h0

1.3.9.10.3.5.77 CQRS21 (0x0454)

CQRS21 - информация об ошибке задачи.

Таблица 1.511 – Поля регистра CQRS21

Бит	Тип	Описание	Сброс
31	R	**CQDTEFV - действительные поля ошибки передачи данных** Контроллер устанавливает этот бит на 1 при обнаружении ошибки во время передачи данных. Контроллер сбрасывает этот бит на 0 при обнаружении ошибки, когда передача данных не выполняется.	1’h0
28-24	R	**CQDTETID - идентификатор ошибки задачи передачи данных** Контроллер меняет в этом поле идентификатор ошибки активной передачи данных, в течение которой возникла ошибка.	5’h0
21-16	R	**CQDTECI - индекс ошибочной команды передачи данных** Контроллер меняет в этом поле индекс команды передачи данных, в течение которой возникла ошибка.	6’h0
15	R	**CQRMEFV - действительные поля ошибки режима ответа** Контроллер устанавливает этот бит на 1, если обнаружена ошибка во время активной транзакции команды. Контроллер устанавливает этот бит на 0, если обнаружена ошибка, и транзакция команды не выполняется.	1’h0
12-8	R	**CQRMETID - идентификатор задачи ошибки режима ответа** Контроллер меняет в этом поле идентификатор активной команды, при которой возникла ошибка.	5’h0
5-0	R	**CQRMECI - индекс команды ошибки режима ответа** Контроллер меняет в этом поле индекс команды, при которой возникла ошибка.	6’h0

1.3.9.10.3.5.78 CQRS22 (0x0458)

CQRS22 - индекс ответа команды.

Таблица 1.512 – Поля регистра CQRS22

Бит	Тип	Описание	Сброс
5-0	R	**CQLCRI - индекс ответа на последнюю команду** Контроллер меняет в этом поле индекс команды при получении ответа.	6’h0

1.3.9.10.3.5.79 CQRS23 (0x045c)

CQRS23 - аргумент ответа команды.

Таблица 1.513 – Поля регистра CQRS23

Бит	Тип	Описание	Сброс
31-0	R	**CQLCRA - аргумент ответа последней команды** Контроллер меняет в этом поле аргумент команды при получении ответа.	32’h0

1.3.9.11 Контроллер интерфейса Ethernet (ETH)

1.3.9.11.1 Характеристики контроллера ETH

Контроллер интерфейса Ethernet (ETH) обладает следующими характеристиками:

1) поддерживаемые скорости передачи данных:
   a) 10 Мбит/с;
   b) 100 Мбит/с;
2) поддерживаемые режимы передачи данных:
   a) полудуплекс;
   b) полный дуплекс;
3) поддержка протоколов:
   a) TCP;
   b) IPv4;
   c) UDP;
4) встроенный контроллер DMA:
   a) один канал - для передачи данных;
   b) один канал - для приёма данных;
5) генерация маскируемых прерываний;
6) интерфейс для взаимодействия в внешним блоком PHY: RMII;
7) контроллер MDIO для настройки внешнего PHY;
8) режимы низкого потребления:
   a) возможность снятия тактирования;
   b) возможность снятия питания;
9) работа от двух тактовых сигналов (см. 1.3.9.11.4).

1.3.9.11.2 Устройство и функционирование блока

1.3.9.11.2.1 Структура контроллера ETH

Контроллер ETH состоит из следующих основных функциональных частей:

– каналов DMA;

– FIFO (буферы внутренней памяти);

– Ethernet приемника и передатчика RMII;

– AHB мастер порта;

– программных регистров;

– интерфейса MDIO;

– схему отключения тактирования и питания (CG и Rej).

Структурная схема контроллера ETH приведена на рисунке 1.133.

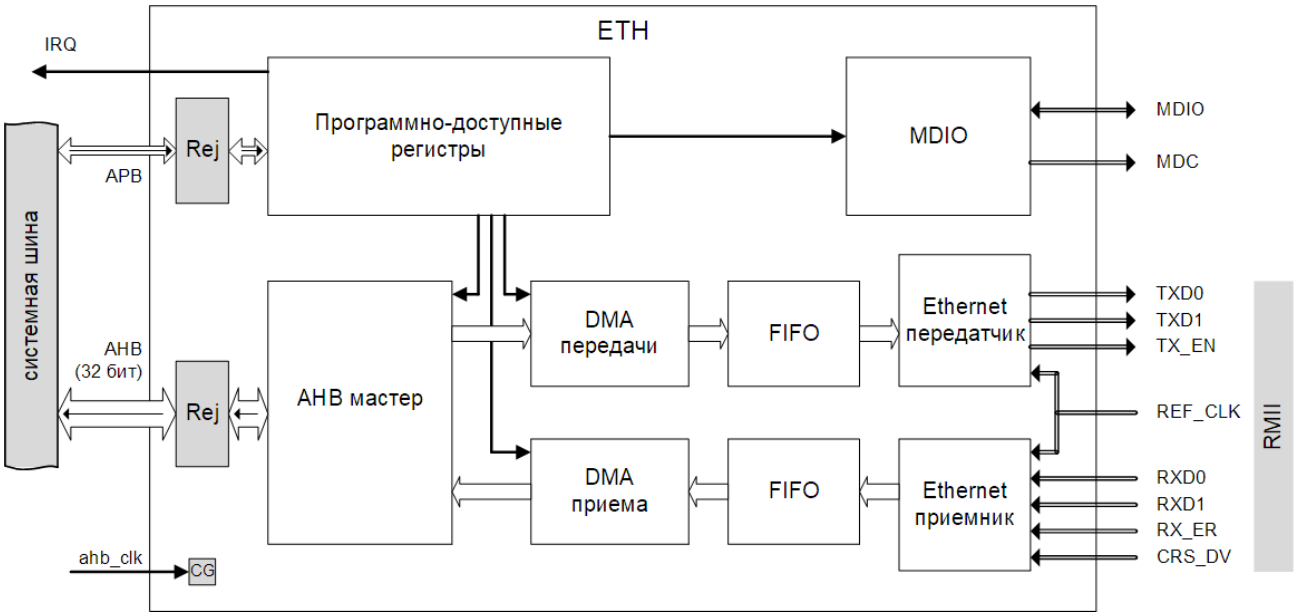


Рисунок 1.133 – Структурная схема контроллера ETH

Основной функционал реализуют каналы DMA, которые используются для передачи данных между системной шиной и сетью Ethernet. Один канал DMA используется передатчиком и один - приемником. DMA каналы управляются через программно доступные регистры.

MDIO интерфейс используется для доступа к конфигурационным и статусным регистрам в одином или нескольких внешних Ethernet PHY, подключенных к ETH. MDIO также управляется через программно доступные регистры.

Интерфейс RMII (Reduced Media Independent Interface) используется для подключения ETH к внешнему PHY.

1.3.9.11.2.2 Описание внешних выводов контроллера ETH

Список внешних выводов контроллар ETH и их описание приведены в теблице 1.514

Таблица 1.514 – Список внешних выводов контроллера ETH

Наименование вывода	Тип (Вх/Вых)	Функциональное назначение
ETH_REF_CLK	Вх	RMII интерфейс - опорный тактовый сигнал 50 МГц
ETH_RXD0	Вх	RMII интерфейс - входящие данные, бит 0
ETH_RXD1	Вх	RMII интерфейс - входящие данные, бит 1
ETH_RX_ER	Вх	RMII интерфейс - ошибка приёма
ETH_CRS_DV	Вх	RMII интерфейс - валидность входящих данных
ETH_TXD0	Вых	RMII интерфейс - исходящие данные, бит 0
ETH_TXD1	Вых	RMII интерфейс - исходящие данные, бит 1
ETH_TX_EN	Вых	RMII интерфейс - разрешение передачи
MDC	Вых	Линия для передачи синхросигнала. Значение частоты равно ahb_clk/64
MDIO	Вх/Вых	Двунаправленная линия для последовательного приёма/передачи данных

1.3.9.11.2.3 Поддержка протоколов

Контроллер ETH соответствует стандартам IEEE 802.3-2002 и 802.3Q-2003.

Не поддерживается дополнительный подуровень управления (optional control sublayer). Это означает, что пакеты с типом 0x8808 (единственные в настоящее время определенные управляющие пакеты) отбрасываются.

1.3.9.11.2.4 Передача данных

Передающее DMA используется для отправки пакетов в сеть Ethernet. Для определения передаваемых данных используются дескрипторы, размещенные в памяти микроконтроллера. Каждый дескриптор состоит из двух 32 разрядных слов.

1.3.9.11.2.4.1 Установка дескриптора

Формат дескриптора приведен в таблицах 1.515 и 1.516. Количество передаваемых байт устанавливается в поле LENGTH, а поле ADDRESS содержит указатель на передаваемые данные в памяти микроконтроллера. Адрес должен быть выровнен с точностью до 4 байтного слова. По окончании передачи пакета будет сгенерировано прерывание, если бит IE установлен (также требуется, чтобы был установлен бит CTRL[TI]).

Таблица 1.515 – Поля дескриптора передатчика ETH, слово 0 - txdesc_w0 (смещение адреса 0x0)

Биты	Название	Описание поля
31:16		Зарезервировано
15	AL	Поле будет аппаратно установлено в 1, если пакет не был передан и было достигнуто максимальное количество попыток его передачи.

Биты	Название	Описание поля
14	UE	Поле будет аппаратно установлено в 1, если пакет был некорректно передан из-за ошибки нехватки данных в FIFO.
13	IE	1 - разрешение генерации прерывания по отправке пакета этого дескриптора; 0 - запрещено. Также для генерации прерывания в управляющем регистре CTRL должен быть установлен в 1 бит TI. Поле будет установлено независимо от того, успешно был передан пакет или он был завершен с ошибкой.
12	WR	1 - указатель на дескриптор возвращается в 0 после того, как текущий дескриптор был использован. 0 - указатель будет увеличен на 8. Указатель на дескриптор автоматически возвращается в 0 при достажении границы в 1 Кбайт.
11	EN	1 - разрешение исполнения дескриптора. 0 - исполнение запрещено. Это поле следует всегда устанавливать последним из всех полей дескриптора.
10:0	LENGTH	Количество байт, которое должно быть передано.

Таблица 1.516 – Поля дескриптора передатчика, ETH слово 1 - txdesc_w1 (смещение адреса 0x4)

Биты	Название	Описание поля
31:2	ADDRESS	Указатель на область памяти, откуда передаваемые данные будут загружены.
1:0		Зарезервировано

1.3.9.11.2.4.2 Начало передачи

Установить поле разрешения исполнения txdesc_w0[EN] в дескрипторе не достаточно, чтобы начать передачу. Сначала в регистре TXBASE контроллера ETH должен быть установлен указатель на область памяти, содержащей дескрипторы. Адрес должен быть выровнен по границе 1 Кбайт. Разряды с 31 по 10 содержат базовый адрес области дескрипторов, в то время как разряды с 9 по 3 формируют указатель на конкретный дескриптор. Первый дескриптор должен быть размещен по базовому адресу и, когда он будет использован контроллером ETH, поле указателя будет увеличено на 8 и станет указывать на следующий дескриптор. Указатель будет автоматически возвращен обратно в 0 при достижении границы 1 Кбайт (т.е. после использования дескриптора со смещением адреса 0x3F8). Поле txdesc_w0[WR] может быть установлено в любом дескрипторе, чтобы вернуть указатель к 0 предварительно - после исполнения этого дескриптора.

Поле указателя также сделано доступным по записи для максимальной гибкости, но нужно соблюдать осторожность при записи указателя дескриптора в регистр. Регистр TXBASE нельзя программно менять, если передача активна.

Последний шаг для запуска передачи - это установка поля разрешения передачи TE в регистре CTRL. Это сообщает контроллеру ETH, что в таблице есть разрешенные дескрипторы. После программной установки этот бит не сбрасывается, пока есть незавершенные дескрипторы (в том числе, если соответствующие им передачи уже исполняются). Дескрипторы всегда должны быть разрешены до разрешения передачи (т.е. установки поля CTRL[TE]).

1.3.9.11.2.4.3 Обработка дескриптора после передачи

Когда передача пакета закончена, её статус записывается в первое слово соответствующего дескриптора. Поле txdesc_w0[UE] устанавливается, если FIFO стало пустым до того, как пакет был полностью передан. Поле txdesc_w0[AL] устанавливается, если произошло больше коллизий передачи, чем разрешено. Максимальное количество попыток передачи равно 16. Пакет был успешно передан, если оба этих разряда равны 0. Другие разряды первого слова дескриптора устанавливаются в 0 после передачи, в то время как второе слово остается неизменным.

Бит txdesc_w0[EN] может быть использован как индикатор, когда дескриптор можно использовать снова (после того, как он был сброшен контроллером ETH). В ETH 3 разряда, которые содержат статус передачи. Ошибка передатчика STAT[TE] устанавливается каждый раз, когда передача заканчивается с ошибкой (когда по крайней мере один из 2 разрядов статуса в передаваемом дескрипторе был установлен). Прерывание передатчика STAT[TI] устанавливается каждый раз, когда передача закончилась успешно.

Бит ошибки STAT[TA] устанавливается, когда системная шина возвращает ошибку во время чтения дескриптора или во время чтения пакета данных. Любые активные передачи будут прекращены и передатчик будет выключен. Передатчик может быть включен снова установкой поля CTRL[TE].

1.3.9.11.2.4.4 Установка данных передачи

Данные для передачи должны быть размещены, начиная с адреса, указанного в поле адреса дескриптора. Контроллер ETH не добавляет к данным поля Ethernet адреса и типа, поэтому они заранее должны быть расположены среди данных пакета пользователем. Контроллер ETH автоматически добавляет 4 байта Ethernet CRC в конце каждого пакета. Данные каждого дескриптора будут посылаться как единственный Ethernet пакет. Если поле размера в дескрипторе больше, чем 1514 байт, пакет не будет передан.

1.3.9.11.2.5 Прием данных

Принимающее DMA используется для получения данных из сети Ethernet. Для определения принимаемых данных используются дескрипторы, размещенные в памяти микроконтроллера. Каждый дескриптор состоит из двух 32 разрядных слов.

1.3.9.11.2.5.1 Установка дескрипторов

Формат дескриптора приведен в таблицах 1.517 и 1.518. Количество принимаемых байт устанавливается в поле LENGTH, а поле ADDRESS содержит указатель, по которому принимаемые данные будут сохранены в памяти микроконтроллера. Адрес должен быть выровнен с точностью до 4 байтного слова. ETH никогда не будет записывать в память больше, чем 1514 байт. Если бит IE установлен, прерывание будет сгенерировано, когда пакет записан в память (также требуется установка бита RI в регистре CTRL).

Таблица 1.517 – Поля дескриптора передатчика ETH, слово 0 - rxdesc_w0 (смещение адреса 0x0)

Биты	Название	Описание поля
31:27		Зарезервировано
26	MC	Устанавливается в 1 аппаратно, если получен пакет с адресом групповой (multicast) рассылки (но не широковещательным - broadcast).
25:19		Зарезервировано
18	LE	Устанавливается в 1 аппаратно, если поле длины/типа пакета не совпадает с действительным числом принятых байтов.
17	OE	Устанавливается в 1 аппаратно, если кадр (frame) был некорректно принят из-за переполнения FIFO.
16	CE	Устанавливается в 1 аппаратно, если ошибка CRC была обнаружена в этом кадре (frame).
15	FT	Устанавливается в 1 аппаратно, если кадр больший, чем максимальный размер был принят. Лишняя часть кадра будет отброшена.
14	AE	Устанавливается в 1 аппаратно, если нечетное число полубайтов было принято.
13	IE	1 - разрешение генерации прерывания по приему пакета по этому дескриптору; 0 - запрещено. Также для генерации прерывания в управляющем регистре CTRL должен быть установлен в 1 бит RI. Поле будет установлено независимо от того, успешно был принят пакет или он был завершен с ошибкой.
12	WR	1 - указатель на дескриптор возвращается в 0 после того, как текущий дескриптор был использован. 0 - указатель будет увеличен на 8. Указатель на дескриптор автоматически возвращается в 0 при достажении границы в 1 Кбайт.

Биты	Название	Описание поля
11	EN	1 - разрешение исполнения дескриптора. 0 - исполнение запрещено. Это поле следует всегда устанавливать последним из всех полей дескриптора.
10:0	LENGTH	Количество байт, которое должно быть принято.

Таблица 1.518 – Поля дескриптора приемника ETH, слово 1 - rxdesc_w1 (смещение адреса 0x4)

Биты	Название	Описание поля
31:2	ADDRESS	Указатель в памяти, куда будут загружены принятые данные.
1:0		Зарезервировано.

1.3.9.11.2.5.2 Начало приема

Установить поле rxdesc_w0[EN] в дескрипторе не достаточно, чтобы начать прием. Сначала в регистре RXBASE контроллера ETH должен быть установлен указатель на область памяти, содержащей дескрипторы. Адрес должен быть выровнен по границе 1 Кбайт. Разряды с 31 по 10 содержат базовый адрес области дескрипторов, в то время как разряды с 9 по 3 формируют указатель на конкретный дескриптор. Первый дескриптор должен быть размещен по базовому адресу и, когда он будет использован контроллером ETH, поле указателя будет увеличено на 8 и станет указывать на следующий дескриптор. Указатель будет автоматически возвращен обратно в 0 при достижении границы 1 Кбайт (т.е. после использования дескриптора со смещением адреса 0x3F8). Поле rxdesc_w0[WR] может быть установлено в любом дескрипторе, чтобы вернуть указатель к 0 предварительно - после исполнения этого дескриптора.

Поле указателя также сделано доступным по записи для максимальной гибкости, но нужно соблюдать осторожность при записи указателя дескриптора в регистр. Регистр RXBASE нельзя программно менять, если прием активен.

Последний шаг для запуска приема - это установка поля RE в регистре CTRL. После этого контроллер ETH считает первый дескриптор и будет ожидать входящий пакет.

1.3.9.11.2.5.3 Обработка дескриптора после приема

При окончании приема контроллер ETH сбрасывает бит rxdesc_w0[EN] в исполненом дескрипторе. Другие управляющие разряды (WR, IE) также сбрасываются. Количество принятых байтов устанавливается в поле LENGTH. Сохраняемые в системной памяти части Ethernet-кадра - это адрес назначения, адрес источника, тип и поля данных. Разряды 18-14 в первом слове дескриптора – статус разряды, указывающие различные ошибки приема. Все пять разрядов – нули после приема без ошибок. Разряды статуса описаны в таблице 1.517.

При приеме пакета с неподходящим MAC адресом, будет установлен бит STAT[IA].

Пакеты размера больше максимального размера вызывают установку rxdesc_w0[FT] разряда. Не гарантируется, что поле длина содержит правильное значение принятых байтов. Счетчик останавливается, когда слово, содержащее последний байт вплоть до предельно максимального размера, было записано в память.

Поле ADDRESS в дескрипторе никогда не изменяется контроллером ETH.

1.3.9.11.2.5.4 Ошибки системной шины во время приема

Если ошибка системной шины происходит во время чтения дескриптора или сохранения данных, будет установлен бит ошибки STAT[RA] и приемник будет выключен. Данные текущего приема будут отброшены. Приемник может быть включен снова программной установкой бита RE в регистре CTRL.

1.3.9.11.2.6 Принимаемые MAC адреса

В конфигурации по умолчанию контроллер принимает пакеты либо с конкретным адресом (unicast), установленным в регистрах MAC-адреса, либо с широковещательным адресом.

Также можно включить поддержку групповой (multicast) рассылки, и в этом случае для фильтрации полученных пакетов используется хэш-функция. Регистры HMSB и HLSB определяют, какие адреса могут приниматься. Каждый из 64 битов регистров сопоставляется с группой адресов с помощью хэш-функции. Функция сопоставления:

1) над адресом полученного пакета расчитывается 32-битный Ethernet CRC;

2) младшие 6 бит результата используются как номер бита в регистрах {HMSB, HLSB};

3) если бит установлен в единицу, пакет будет получен. Если ноль - проигнорирован.

Бит в дескрипторе приема также устанавливается, если на него был получен пакет групповой (multicast) рассылки.

1.3.9.11.2.7 MDIO интерфейс

MDIO интерфейс обеспечивает доступ к конфигурационным и статусным регистрам внешнего Ethernet PHY через двухпроводной интерфейс. Контроллер ETH имеет встроенный механизм, полностью поддерживающий MDIO.

MDIO интерфейс может быть использован для доступа к от 1 до 32 отдельних PHY, содержащих от 1 до 32 16-разрядных регистров каждый.

Для запуска операции чтения надо установить адрес PHY в поле MDIO[PHYADDR], адрес регистра в поле MDIO[REGADDR] и 1 в поле MDIO[RD]. Сразу после этого автоматически установится в 1 бит занятости интерфейса MDIO[BU]. По окончании операции бит MDIO[BU]

автоматически сбросится в 0. Если операция завершена успешно, MDIO[LF] бит будет равен 0, и поле MDIO[DATA] будет содержать считанные данные. Если операция завершена с ошибкой, MDIO[LF] бит будет равен 1, поле MDIO[DATA] будет не валидно.

Для запуска операции записи надо установить адрес PHY в поле MDIO[PHYADDR], адрес регистра в поле MDIO[REGADDR], записываемые данные в поле MDIO[DATA] и 1 в поле MDIO[WR]. Сразу после этого автоматически установится в 1 бит занятости интерфейса MDIO[BU]. По окончании операции бит MDIO[BU] автоматически сбросится в 0. Если операция завершена успешно, бит MDIO[LF] будет равен 0. Если операция завершена с ошибкой, бит MDIO[LF] будет равен 1.

1.3.9.11.2.8 Отладочный доступ к буферу

Контроллер ETH поддерживает режим отладки, обеспечивая доступ к принятым и передаваемым пакетам в своих внутренних буферах памяти.

Разрешение отладочного доступа к внутренним буферам включается программно в бите CTRL[RD]. Перед этим приемник и передатчик должны быть выключены, иначе данные в буферах могут быть повреждены.

1.3.9.11.3 Описание регистров

Контроллер ETH управляется через программно доступные регистры, расположенные на системной шине.

Также, в отладочных целях, имеется программный доступ к внутренним буферам (FIFO) памяти контроллера.

Карта регистров приведена в таблице 1.519.

1.3.9.11.3.1 Карта регистров

Таблица 1.519 – Карта регистров ETH

Адрес	Название	Доступ	Значение после сброса	Описание
0x000	CTRL			Управляющий регистр ETH
0x004	STAT			Регистр статуса/прерываний ETH
0x008	MACMSB			Регистр MAC-адрес MSB
0x00C	MACLSB			Регистр MAC-адрес LSB
0x010	MDIO			Регистр управления/статуса MDIO
0x014	TXBASE			Регистр базового адреса таблицы дескрипторов передатчика

Адрес	Название	Доступ	Значение после сброса	Описание
0x018	RXBASE			Регистр базового адреса таблицы дескрипторов приемника
0x020	HMSB			Hash table msb
0x024	HLSB			Hash table lsb
0x10000 - 0x107FC				Отладочный доступ к буферу передатчика
0x20000 - 0x207FC				Отладочный доступ к буферу приемника

1.3.9.11.3.2 Описание полей регистров

1.3.9.11.3.2.1 Регистр CTRL

Смещение адреса 0x000

В таблице 1.520 представлено описание полей регистра.

Таблица 1.520 – Описание полей регистра CTRL

Поле	Название	Программный доступ	Состояние после сброса	Описание
31	EA	ЧТ		Установлен в 1, если контроллер ETH поддерживает протокол EDCL
30:28	BS	ЧТ		Показывает количество памяти, используемое в EDCL буферах. 0=1кВ, 1=2кВ, …6=64кВ.
27				Зарезервировано
26	MA	ЧТ		Установлен в 1, если контроллер ETH поддерживает прерывания MDIO
25	MC	ЧТ		Установлен в 1, если контроллер ETH поддерживает прием пакетов групповой (multicast) рассылки
24:15				Зарезервировано
14	ED	ЧТ/ЗП		1 - EDCL выключен. Поле доступно только если контроллер поддерживает протокол EDCL; 0 - EDCL включен. Значение по сбросу берется с входного сигнала контроллера ethi.edcldisable
13	RD	ЧТ/ЗП		1 - включение режима отладки внутрненних буферов памяти; 0 - выключено

Поле	Название	Программный доступ	Состояние после сброса	Описание
12	DD	ЧТ/ЗП	0	1 - выключение автомата определения параметров PHY (speed, duplex) в режиме EDCL. 0 - автомат включен; Если операция определения параметров не завершается успешно, MDIO интерфейс будет заблокирован в режиме занятости. Если программному обеспечению нужен доступ к MDIO, можно установить этот бит в 1 и, когда поле MDIO.BU станет равным 0, интерфейс будет доступен. Автомат не может быть программно включен повторно.
11	ME	ЧТ/ЗП	0	1 - разрешение приема пакетов групповой (multicast) рассылки; 0 - запрещено
10	PI	ЧТ/ЗП	0	1 - разрешение генерации прерывания при изменениях статусов обнаруженных внешних PHY; 0 - запрещено
9:8				Зарезервировано
7	SP	ЧТ/ЗП	1	Устанавливает текущий режим скорости: 0 - 10 Mbit; 1 - 100 Mbit. Значение по умолчанию автоматически считывается из PHY после сброса.
6	RS	ЧТ/ЗП	0	Запись: 1 - запуск процедуры программного сброса контроллера ETH. По окончании процедуры поле будет аппаратно сброшено в 0; 0 - не имеет действия. Чтение: 1 - процедура программного сброса исполняется; 0 - не исполняется. После записи 1 в это поле, ПО не должно делать ничего другого, кроме опроса этого поля и ожидания его сброса в 0.
5	PM	ЧТ/ЗП	0	1 - ETH работает в смешанном режиме, который означает, что он будет получать все пакеты независимо от адреса назначения; 0 - обычный режим работы.
4	FD	ЧТ/ЗП	0	1 - ETH работает в дуплексном режиме; 0 - ETH работает в полудуплексе.
3	RI	ЧТ/ЗП	0	1 - разрешение прерываний приемника. Прерывание будет сгенерировано каждый раз, когда пакет принят и этот бит установлен в 1. Прерывание генерируется независимо от того, был ли приём пакета завершен успешно или с ошибкой; 0 - запрещено.

Поле	Название	Программный доступ	Состояние после сброса	Описание
2	TI	ЧТ/ЗП	0	1 - разрешение прерываний передатчика. Прерывание будет сгенерировано каждый раз, когда пакет передан и этот бит установлен в 1. Прерывание генерируется независимо от того, была ли передача пакета завершена успешно или с ошибкой; 0 - запрещено.
1	RE	ЧТ/ЗП	0	Каждый раз при разрешении новых дескрипторов приема следует устанавливать это поле в 1. Пока это поле равно 1, ETH будет считывать новые дескрипторы приема, и как только он встретит неразрешенный дескриптор, он остановится до тер пор, пока RE не будет установлен снова; 0 - выборка дескрипторов приема выключена.
0	TE	ЧТ/ЗП	0	Каждый раз при разрешении новых дескрипторов передачи следует устанавливать это поле в 1. Пока это поле равно 1, ETH будет считывать новые дескрипторы передачи, и как только он встретит неразрешенный дескриптор, он остановится до тер пор, пока TE не будет установлен снова; 0 - выборка дескрипторов передачи выключена.

1.3.9.11.3.2.2 Регистр STAT

Смещение адреса 0x004

В таблице 1.521 представлено описание полей регистра.

Таблица 1.521 – Описание полей регистра STAT

Поле	Название	Программный доступ	Состояние после сброса	Описание
31:28				Зарезервировано
27:24	NRD	ЧТ	0	Показывает размер таблицы дескрипторов DMA: 0b0000 = 128; 0b0001 = 256; 0b0010 = 512; 0b0010 = 1024; 0b0100 = 2048; 0b0101 = 4096; 0b0110 = 8192; 0b0111 = 16384; 0b1000 = 32768; 0b1001 = 65536.
23:9				Зарезервировано
8	PS	ЧТ/ЗП	0	Устанавливается в 1 автоматически при обнаружении изменения статуса внешнего PHY. Запись 1 сбросит значение поля в 0.

Поле	Название	Программный доступ	Состояние после сброса	Описание
7	IA	ЧТ/ЗП	0	Устанавливается в 1 автоматически при получении пакет с адресом, не соответствующим MAC адресу. Запись 1 сбросит значение поля в 0.
6	TS	ЧТ/ЗП	0	Устанавливается в 1 автоматически при приеме пакета меньше минимального размера. Запись 1 сбросит значение поля в 0.
5	TA	ЧТ/ЗП		Устанавливается в 1 автоматически при получении ошибки системной шины во время работы DMA передатчика. Запись 1 сбросит значение поля в 0. Поле не сбрасывается ни программным, ни аппаратным сбросом и должно быть сброшено записью 1 перед началом работы.
4	RA	ЧТ/ЗП		Устанавливается в 1 автоматически при получении ошибки системной шины во время работы DMA приемника. Запись 1 сбросит значение поля в 0. Поле не сбрасывается ни программным, ни аппаратным сбросом и должно быть сброшено записью 1 перед началом работы.
3	TI	ЧТ/ЗП		Устанавливается в 1 автоматически при успешной передаче пакета. Запись 1 сбросит значение поля в 0. Поле не сбрасывается ни программным, ни аппаратным сбросом и должно быть сброшено записью 1 перед началом работы.
2	RI	ЧТ/ЗП		Устанавливается в 1 автоматически при успешном приеме пакета. Запись 1 сбросит значение поля в 0. Поле не сбрасывается ни программным, ни аппаратным сбросом и должно быть сброшено записью 1 перед началом работы.
1	TE	ЧТ/ЗП		Устанавливается в 1 автоматически при ошибке в процессе передачи пакета. Запись 1 сбросит значение поля в 0. Поле не сбрасывается ни программным, ни аппаратным сбросом и должно быть сброшено записью 1 перед началом работы.
0	RE	ЧТ/ЗП		Устанавливается в 1 автоматически при ошибке в процессе приема пакета. Запись 1 сбросит значение поля в 0. Поле не сбрасывается ни программным, ни аппаратным сбросом и должно быть сброшено записью 1 перед началом работы.

1.3.9.11.3.2.3 Регистр MACMSB

Смещение адреса 0x008

В таблице 1.522 представлено описание полей регистра.

Таблица 1.522 – Описание полей регистра MACMSB

Поле	Название	Программный доступ	Состояние после сброса	Описание
31:16				Зарезервировано
15:0	MACMSB	ЧТ/ЗП		Два старших значащих байта MAC – адреса. Поле не сбрасывается ни программным, ни аппаратным сбросом и должно быть установлено программно перед началом работы.

1.3.9.11.3.2.4 Регистр MACLSB

Смещение адреса 0x00C

В таблице 1.523 представлено описание полей регистра.

Таблица 1.523 – Описание полей регистра MACLSB

Поле	Название	Программный доступ	Состояние после сброса	Описание
31:0	MACLSB	ЧТ/ЗП		Четыре младших значащих байта MAC – адреса. Поле не сбрасывается ни программным, ни аппаратным сбросом и должно быть установлено программно перед началом работы.

1.3.9.11.3.2.5 Регистр MDIO

Смещение адреса 0x010

В таблице 1.524 представлено описание полей регистра.

Таблица 1.524 – Описание полей регистра MDIO

Поле	Название	Программный доступ	Состояние после сброса	Описание
31:16	DATA	ЧТ/ЗП	0	Содержит данные, которые: - считываются при чтении по MDIO; - передаются при записи по MDIO.
15:11	PHYADDR	ЧТ/ЗП	0	Это поле содержит адрес PHY, к которому будут вестись обращение при чтении или записи по MDIO.
10:6	REGADDR	ЧТ/ЗП	0	Это поле содержит адрес регистра, к которому будут вестись обращение при чтении или записи по MDIO.
5:4				Зарезервировано

Поле	Название	Программный доступ	Состояние после сброса	Описание
3	BU	ЧТ	0	1 - выполняется операция чтения/записи по MDIO. 0 - операция закончена, MDIO в состоянии бездействия.
2	LF	ЧТ	1	Устанавливается в 1, если поле BUSY == 0 и связь по MDIO не была установлена.
1	RD	ЧТ/ЗП	0	Запись: 1 - запуск операции чтения по MDIO. По завершении данные будут доступны в поле DATA. 0 - не имеет действия.
0	WR	ЧТ/ЗП	0	Запись: 1 - запуск операции записи по MDIO. Данные берутся из поля DATA. 0 - не имеет действия.

1.3.9.11.3.2.6 Регистр TXBASE

Смещение адреса 0x014

В таблице 1.525 представлено описание полей регистра.

Таблица 1.525 – Описание полей регистра TXBASE

Поле	Название	Программный доступ	Состояние после сброса	Описание
31:10	BASEADDR	ЧТ/ЗП		Базовый адрес таблицы дескрипторов передачи. Поле не сбрасывается ни программным, ни аппаратным сбросом и должно быть установлено программно перед началом работы.
9:3	DESCPNT	ЧТ/ЗП	0	Указатель на конкретные дескрипторы передачи. Значение автоматически увеличивается контроллером ETH.
2:0				Зарезервировано

1.3.9.11.3.2.7 Регистр RXBASE

Смещение адреса 0x018

В таблице 1.526 представлено описание полей регистра.

Таблица 1.526 – Описание полей регистра RXBASE

Поле	Название	Программный доступ	Состояние после сброса	Описание
31:10	BASEADDR	ЧТ/ЗП		Базовый адрес таблицы дескрипторов приема. Поле не сбрасывается ни программным, ни аппаратным сбросом и должно быть установлено программно перед началом работы.
9:3	DESCPNT	ЧТ/ЗП	0	Указатель на конкретные дескрипторы приема. Значение автоматически увеличивается контроллером ETH.
2:0				Зарезервировано

1.3.9.11.3.2.8 Регистр HMSB

Смещение адреса 0x020

В таблице 1.527 представлено описание полей регистра.

Таблица 1.527 – Описание полей регистра HMSB

Поле	Название	Программный доступ	Состояние после сброса	Описание
31:0	HMSB	ЧТ/ЗП		Старшая часть хэша для фильтрации адресов при приеме пакетов групповой (multicast) рассылки. Поле не сбрасывается ни программным, ни аппаратным сбросом и должно быть установлено программно перед началом работы.

1.3.9.11.3.2.9 Регистр HLSB

Смещение адреса 0x024

В таблице 1.528 представлено описание полей регистра.

Таблица 1.528 – Описание полей регистра HLSB

Поле	Название	Программный доступ	Состояние после сброса	Описание
31:0	HLSB	ЧТ/ЗП		Младшая часть хэша для фильтрации адресов при приеме пакетов групповой (multicast) рассылки. Поле не сбрасывается ни программным, ни аппаратным сбросом и должно быть установлено программно перед началом работы.

1.3.9.11.4 Интеграция в микроконтроллере

Контроллер ETH использует два тактовых сигнала:

1) eth_ref_clk - от этой внешней частоты работает ядро контроллера и протокол Ethernet;

2) ahb_clk - на этой частоте работают интерфейсы с системной шиной.

ETH допускает независимые значения этих двух частот, что позволяет не перенастраивать его при смене частоты системной шины ahb_clk.

При установке слишком низкой частоты ahb_clk контроллер ETH будет терять входящие пакеты. Минимальная частота:

– 2,5 МГц - для режима 10 Мбит/с;

– 18 МГц - для режима 100 Мбит/с.

При выключенном в RCU тактовом сигнале ahb_clk программный доступ в любой регистр будет возвращать ошибку системной шины и вызывать исключение процессорного ядра.

При выключенном в PMU питании контроллера ETH программный доступ в любой регистр будет возвращать ошибку системной шины и вызывать исключение процессорного ядра.

ETH поддерживает только 32-х разрядный доступ к регистрам. Байтовая или двух-байтовая запись будет проигнорирована, ошибка шины возвращена не будет.

Запись в зарезервированные или ЧТ биты или адреса не имеет действия, никакие ошибки не генерируются. Чтение из зарезервированных или ЗП битов или адресов возвращает 0, никакие ошибки не генерируются.

Контроллер ETH генерирует уровневое прерывание - eth_irq.

1.3.9.11.4.1 Подключение сигналов RMII и MDIO

Сигналы интерфейсов RMII и MDIO мультиплексированы с другими сигналами микроконтроллера. Для работы ETH требуется предварительная программная настройка мультиплексоров, которая производится в блоке GPIO.

Буферы микроконтроллера могут подтягивать сигнал MDIO. Это позволяет не использовать подтягивающие резисторы на печатной плате. Требуется предварительная программная настройка.

1.3.10 Контроллер внешних прерываний (EXTINT)

1.3.10.1 Характеристики контроллера внешних прерываний EXTINT

Контроллер внешних прерываний EXTINT предназначен для отслеживания состояний внешних сигналов микроконтроллера, их обработки и генерации на их основе прерываний разного типа.

EXTINT обладает следующими характеристиками:

– обработка сигналов от всех портов ввода/вывода (PA, PB, PC, PD):

a) мультиплексирование для снижения общего количества (105 → 16);

b) обнаружение прерываний:

i) по фронту;

ii) по спаду;

c) передача уровневого прерывания:

i) настраиваемая полярность;

d) фильтрация входного сигнала;

e) маскирование, хранение, программный сброс и установка прерываний;

– генерация событий двух видов:

a) уровневое;

b) переключаемое;

– интерфейс программирования APB slave;

– поддержка TrustZone:

a) secured доступ;

b) privileged доступ;

– программный сброс EXTINT.

1.3.10.2 Устройство и функционирование блока

EXTINT получает сигналы от всех внешних ножек микроконтроллера (PA, PB, PC, PD) и коммутирует максимум 16 из них к каналам обработки. Каждый канал имеет независимую настройку условия выработки прерывания и глитч-фильтр. Любой внешний вывод МК Optimal+ может быть назначен любому каналу контроллера EXTINT. Номер сигнала, к которому коммутируется канал задается в EXTINT_CFG_x[CH_MAP]. Внутри каналов происходит обработка сигналов, распознавание событий и передача их в NVIC или ELC.

Для разрешения обработки сигнала настройки секьюрности ножки (задается в контроллере GPIO) и канала в регистре EXTINT_SECURE должны совпадать.

При несовпадении при включении канала будет установлен бит EXTINT_TRUSTZONE_IRQ.IRQ.

Каждый канал может быть включен и выключен программно с помощью EXTINT_CFG_x[CH_EN]. По умолчанию все каналы выключены.

Каждый канал может управлять внешним подключением поступающего в него сигнала через поле EXTINT_CFG_x[SIG_CTRL].

EXTINT синхронизует сигнал на частоту extint_clk. Это значит, что импульсы длительностью менее 1/extint_clk не будут гарантированно обнаружены.

EXTINT разделен на два тактовых домена:

– extint_clk - на этой частоте происходит выборка входящего сигнала. При использовании фильтра эту частоту можно дополнительно уменьшить внутри EXTINT (см. 1.3.10.2.2).

– hclk - на этой частоте работает программный интерфейс.

EXTINT допускает независимые значения этих двух частот, что позволяет не перенастраивать его при смене частоты системной шины hclk.

Структурная схема EXTINT представлена на рисунке 1.134.

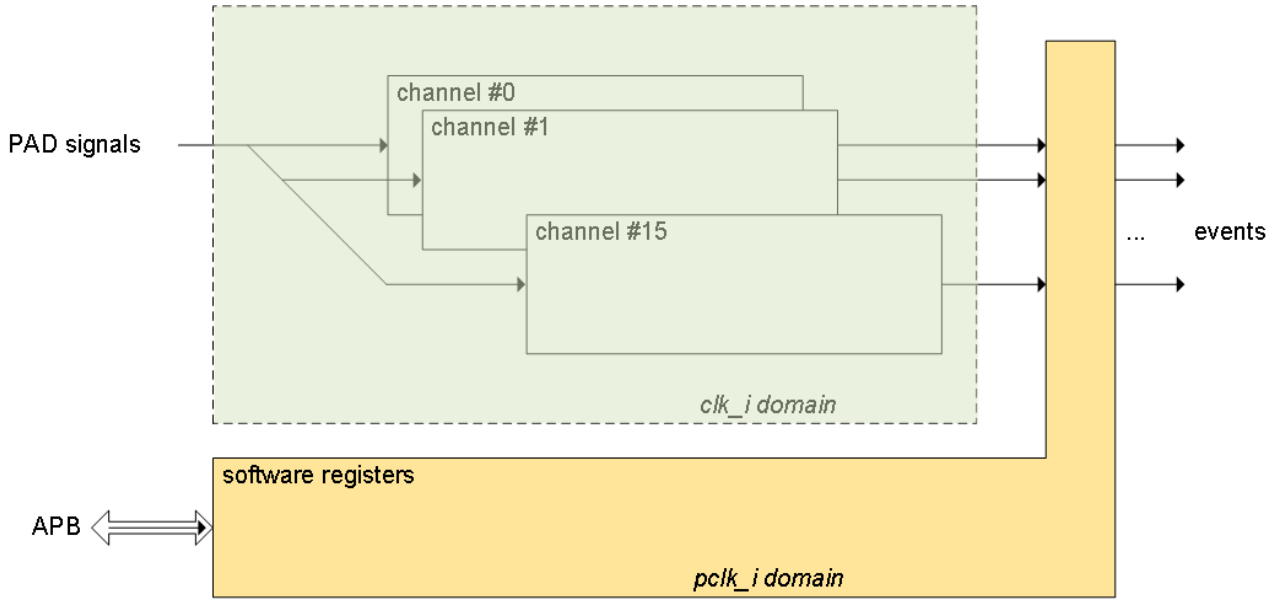


Рисунок 1.134 – Структурная схема EXTINT

Схема каналов обработки сигналов устройства EXTINT представлена на рисунке 1.135.

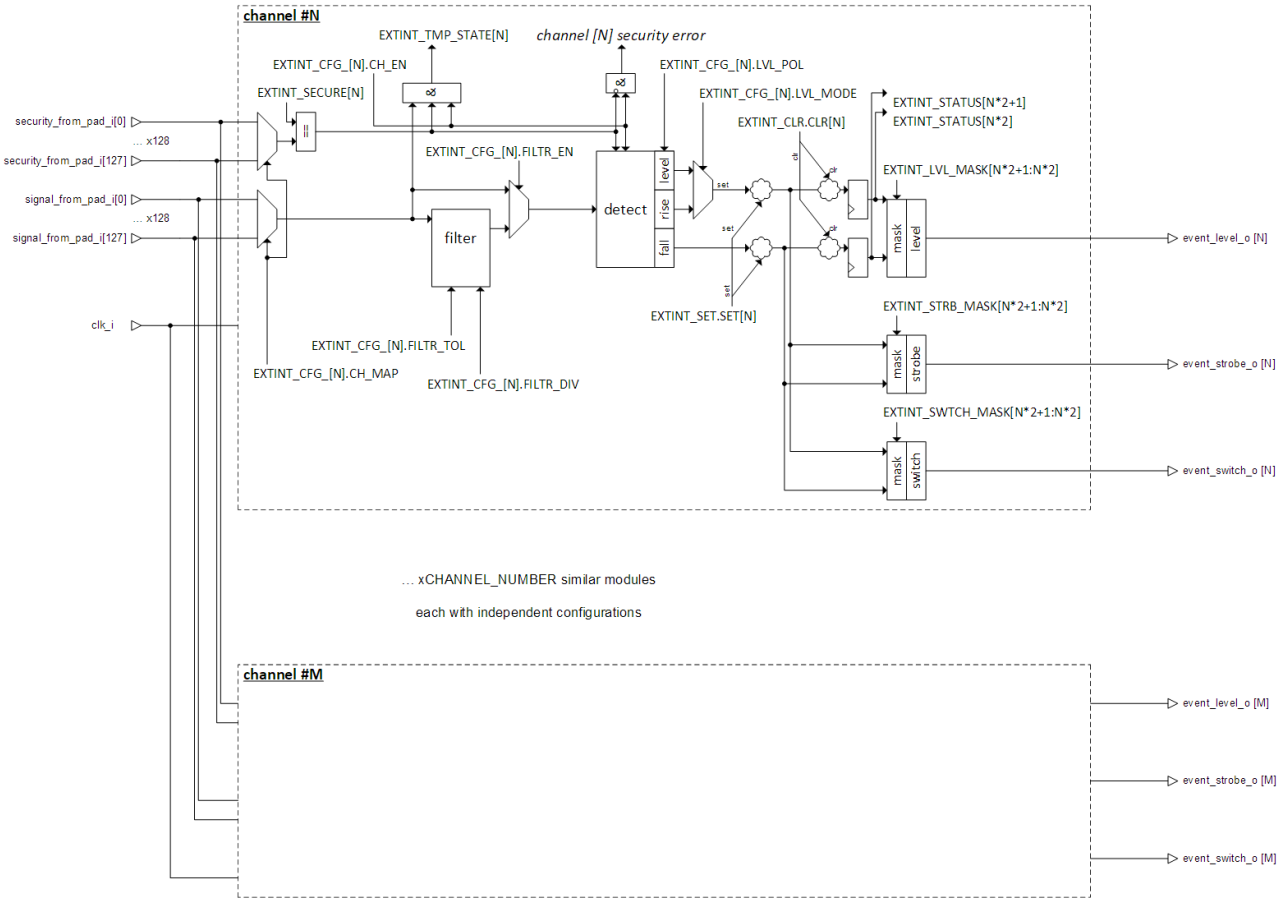


Рисунок 1.135 – Обработка входящего сигнала в EXTINT

1.3.10.2.1 Выдача событий

Каждый канал EXTINT имеет два типа сигналов для выдачи выходных событий:

– уровневый - при выдаче события устанавливается в активное состояние и удерживается в нём до сброса соответствующего статуса. При активном маскировании сигнал сбрасывается в 0. Такие сигналы подключены к NVIC;

– переключаемый - при выдаче события инвертирует своё значение. При активном маскировании сигнал удерживается последнем своём состоянии и не переключается при обнаружении события. Такие сигналы подключены к ELC.

Подключение сигналов к конкретным СФ-блокам приведено в главе 1.3.10.5.

1.3.10.2.2 Фильтрация сигнала

EXTINT имеет программно включаемую с помощью EXTINT_CFG_x[FILTR_EN] возможность фильтрации входящего сигнала (блок filter на рисунке 1.135).

Фильтр представляет собой счётчик, который начинает инкрементироваться при изменении входного сигнала. Если сигнал возвращается в прежнее значение, счётчик

декрементируется. Если значение счётчика превышает программно установленный верхний порог EXTINT_CFG_x[FILTR_TOL], происходит генерация события.

Фильтр имеет программно настраиваемую частоту захвата сигнала. Она кратно меньше частоты extint_clk и настраивается в EXTINT_CFG_x[FILTR_DIV].

Включение канала с фильтром следует делать в соответствии с 1.3.10.4.2.

1.3.10.2.3 Обработка обнаруженного события

EXTINT сохраняет в регистре статуса EXTINT_STATUS отдельно обнаружение фронта/уровня и спада сигнала независимо для каждого канала (блок detect на рисунке 1.135).

Статус может быть сброшен записью 1 в соответствующий бит регистра EXTINT_CLR.

В отладочных целях статус может быть установлен независимо от внешнего сигнала записью 1 в соответствующий бит регистра EXTINT_TEST.

Для статуса фронта/уровня и для статуса спада есть отдельная маска для каждого канала и для каждого типа выходного сигнала. Они задаются в регистрах EXTINT_LVL_MASK, EXTINT_STRB_MASK, EXTINT_SWTCH_MASK.

1.3.10.2.4 Уровневый режим работы

Каждый канал EXTINT может быть переведен в уровневый режим работы установкой в 1 поля EXTINT_CFG_x[LVL_MODE].

В этом режиме вместо фронта сигнала детектируется его уровень. Если он равен активному значению, установленному в поле EXTINT_CFG_x[LVL_POL]:

– событие считается обнаруженным;

– устанавливается младший бит соответствующего поля EXTINT_STATUS[STATUS_CHx];

– устанавливается уровневое выходное событие (если не замаскировано в соответствующем бите регистра EXTINT_LVL_MASK).

В уровневом режиме сигнал также может проходить через фильтр.

Старшие биты полей EXTINT_STATUS[STATUS_CHx] не меняют своё поведение при переходе в уровневый режим и по-прежнему регистрируют только спад сигнала. Их использование в этом режиме не рекомендуется.

Переключаемые выходные события не меняют своё поведение при переходе в уровневый режим и по-прежнему реагируют только на фронт или спад. Их использование в этом режиме не рекомендуется.

1.3.10.2.5 Программный запуск событий

Генерация каждого события может быть запущена программно. Для этого надо установить соответствующий бит в регистре EXTINT_TEST.

Установка бита генерирует событие однократно. Для повторной генерации требуется повторная установка.

Если одновременно с программной генерацией будет обнаружено такое же внешнее событие, наружу будет выдано только одно событие.

Одновременно могут быть запущены события для любого набора каналов.

1.3.10.2.6 Временные характеристики

Задержки обработки сигнала (от изменения значения соответствующей внешней ножки микросхемы, при условии постоянства настроек канала):

1) FILTR_EN = 0

– уровневое событие и переключаемое событие: 3 Textint_clk + 3 Thclk

2) FILTR_EN = 1

Фильтр дает дополнительную задержку генерации выходного события. Значение задержки зависит от настроек фильтра. Задержка может меняться. Максимальные значения приведены ниже:

– уровневое событие и переключаемое событие: (FILTR_TOL+1)*(4^FILTR_DIV)+1 Textint_clk + 3 Thclk

Одновременная генерация событий от разных каналов при одновременном их появлении на входных ножках микросхемы не гарантирована.

Одновременная генерация разных типов выходных событий (уровневое, импульсное, переключаемой) одного канала не гарантированы. Разбежка может составлять 1 такт Thclk.

1.3.10.3 Описание регистров

1.3.10.3.1 Общее описание регистров

Контроллер внешних прерываний EXTINT имеет набор программно доступных регистров, требующий 4 Кбайт адресов на системной шине APB.

Карта регистров приведена в таблице 1.529.

При дальнейшем описании отдельных полей регистров используется следующее обозначение режимов доступа пользователя к полям:

– ЧТ – поле доступно только для чтения. Запись данных в поле игнорируется;

– ЗП – поле доступно только для записи. Чтение возвращает значение 0;

– ЧТ/ЗП – поле доступно как для записи, так и для чтения;

– ЧТ/ЗП1О – поле доступно для чтения. Запись 1 сбросит значение, запись 0 не имеет действия;

Доступ к регистрам осуществляется посредством системного интерфейса управления (APB slave). Поддерживается только 32-х разрядный доступ. Байтовая или двух-байтовая запись будет проигнорирована и возвращена ошибка шины, что вызовет переход процессорного ядра в исключение.

Запись в зарезервированные или ЧТ биты или адреса не имеет действия, никакие ошибки не генерируются. Чтение из зарезервированных или ЗП битов или адресов возвращает 0, никакие ошибки не генерируются.

1.3.10.3.2 Поддержка TrustZone

Некоторые регистры EXTINT доступны только в защищённом режиме работы.

Некоторые регистры EXTINT имеют наборы полей, доступ к которым зависит от их настроек и может быть разным в пределах одного регистра.

Возможности поддержки TrustZone каждого конкретного регистра приведены в его описании.

Secure доступ к non-secure регистрам разрешен. Privileged доступ к non-privileged регистрам разрешен.

Non-secure доступ к secure регистрам не будет иметь действия (при записи) или вернет 0 (при чтении). А также установит EXTINT_TRUSTZONE_IRQ[IRQ]. Non-privileged доступ к priviledeg регистрам не будет иметь действия (при записи) или вернет 0 (при чтении). А также установит EXTINT_TRUSTZONE_IRQ[IRQ].

1.3.10.3.3 Карта регистров

Таблица 1.529 – Карта регистров EXTINT

Адрес	Название	Доступ	Значение после сброса	Описание
0x000	EXTINT_ID	ЧТ	0x49545845	Идентификатор устройства
0x004	EXTINT_VERSION	ЧТ	0x1	Версия устройства
0x008	EXTINT_SW_RST	ЧТ/ЗП	0x0	Программный сброс
0x00C	EXTINT_TMP_STATE	ЧТ	0x0	Текущее состояние входного сигнала
0x010	EXTINT_STATUS	ЧТ	0x0	Статус событий
0x014	EXTINT_TEST	ЗП	0x0	Программная установка событий

Адрес	Название	Доступ	Значение после сброса	Описание
0x018	EXTINT_CLR	ЗП	0x0	Программный сброс событий
0x01C	EXTINT_LVL_MASK	ЧТ/ЗП	0x0	Маска (разрешение) уровневого выхода событий
0x024	EXTINT_SWTCH_MASK	ЧТ/ЗП	0x0	Маска (разрешение) переключаемого выхода событий
0x028	EXTINT_CMN_CFG	ЧТ/ЗП	0x1	Конфигурация EXTINT
0x030	EXTINT_SECURE	ЧТ/ЗП	0x0	Настройки секьюрности
0x034	EXTINT_PRIVILEGE	ЧТ/ЗП	0x0	Настройки привилегированности
0x038	EXTINT_TRUSTZONE_IRQ	ЧТ/ЗП	0x0	Прерывание о нарушении секьюрности
0x03C	EXTINT_TRUSTZONE_MASK	ЧТ/ЗП	0x0	Маска (разрешение) прерывания о нарушении секьюрности
0x040	EXTINT_CFG_0	ЧТ/ЗП	0x23C	Настройки канала 0
0x044	EXTINT_CFG_1	ЧТ/ЗП	0x23C	Настройки канала 1
0x048	EXTINT_CFG_2	ЧТ/ЗП	0x23C	Настройки канала 2
0x04C	EXTINT_CFG_3	ЧТ/ЗП	0x23C	Настройки канала 3
0x050	EXTINT_CFG_4	ЧТ/ЗП	0x23C	Настройки канала 4
0x054	EXTINT_CFG_5	ЧТ/ЗП	0x23C	Настройки канала 5
0x058	EXTINT_CFG_6	ЧТ/ЗП	0x23C	Настройки канала 6
0x05C	EXTINT_CFG_7	ЧТ/ЗП	0x23C	Настройки канала 7
0x060	EXTINT_CFG_8	ЧТ/ЗП	0x23C	Настройки канала 8
0x064	EXTINT_CFG_9	ЧТ/ЗП	0x23C	Настройки канала 9
0x068	EXTINT_CFG_10	ЧТ/ЗП	0x23C	Настройки канала 10
0x06C	EXTINT_CFG_11	ЧТ/ЗП	0x23C	Настройки канала 11
0x070	EXTINT_CFG_12	ЧТ/ЗП	0x23C	Настройки канала 12
0x074	EXTINT_CFG_13	ЧТ/ЗП	0x23C	Настройки канала 13
0x078	EXTINT_CFG_14	ЧТ/ЗП	0x23C	Настройки канала 14
0x07C	EXTINT_CFG_15	ЧТ/ЗП	0x23C	Настройки канала 15

1.3.10.3.4 Описание полей регистров

1.3.10.3.4.1 Регистр EXTINT_ID

Смещение адреса 0x000

В таблице 1.530 представлено описание полей регистра.

Таблица 1.530 – Описание полей регистра EXTINT_ID

Биты	Поле	Доступ	Значение после сброса	Описание
31:0	ID	ЧТ	0x49545845	идентификатор устройства EXTINT

TrustZone: любой доступ

1.3.10.3.4.2 Регистр EXTINT_VERSION

Смещение адреса 0x004

В таблице 1.531 представлено описание полей регистра.

Таблица 1.531 – Описание полей регистра EXTINT_VERSION

Биты	Поле	Доступ	Значение после сброса	Описание
31:0	VERSION	ЧТ	0x1	версия устройства EXTINT

TrustZone: любой доступ

1.3.10.3.4.3 Регистр EXTINT_SW_RST

Смещение адреса 0x008

Процедура программного сброса сбросит все программные регистры, внутренние автоматы и выходные сигналы событий.

Сигналы уровневых выходных событий будут установлены в 0.

Сигналы переключаемых выходных событий сохранят свои значения.

Если сигнал уровневого выходного события находится в 1, то целевое устройство может некорректно обработать его переключение в 0 при программном сбросе. Поэтому перед осуществлением программного сброса EXTINT рекомендуется:

– замаскировать все события всех каналов;

– обработать все ожидающие события.

Программный сброс занимает 6 Thclk + 2 Textint_clk. Поскольку Textint_clk может быть существенно больше Thclk, рекомендуется дождаться сброса бита RESET, прежде чем считать сброс завершенным.

В таблице 1.532 представлено описание полей регистра.

Таблица 1.532 – Описание полей регистра EXTINT_SW_RST

Биты	Поле	Доступ	Значение после сброса	Описание
0	RESET	ЧТ/ЗП	0x0	запись: 1 - запуск процедуры программного сброса; 0 - не имеет действия; чтение: 1 - процедура программного сброса не завершена; 0 - процедура программного сброса завершена.
31:1			0	Зарезервировано

TrustZone: если EXTINT_SECURE[SECURE] != 0, то только секьюрный доступ; если EXTINT_PRIVILEGE[PRIVILEGE] != 0, то только привилегированный доступ; иначе любой доступ.

1.3.10.3.4.4 Регистр EXTINT_TMP_STATE

Смещение адреса 0x00C

В таблице 1.533 представлено описание полей регистра.

Таблица 1.533 – Описание полей регистра EXTINT_TMP_STATE

Биты	Поле	Доступ	Значение после сброса	Описание
15:0	TMP_STATE	ЧТ	0x00000000	текущее состояние сигналов всех каналов; Состояние конкретного сигнала всегда равно 0, если соответствующий канал выключен (CH_EN == 0) или не соответствует настройкам TrustZone; Состояние конкретного сигнала не учитывает фильтрацию, даже если она включена;
31:16			0	Зарезервировано

TrustZone: побитовый. Значение каждого бита (соответствует одному каналу) будет актуальным, если настройки секьюрности и привилегированности этого канала ниже или совпадают с параметрами транзакции. Иначе чтение вернет 0.

1.3.10.3.4.5 Регистр EXTINT_STATUS

Смещение адреса 0x010

Регистр отображает статусы каналов. Поля устанавливаются при обнаружении фронта или спада сигнала. Доступен только на чтение. Запись в любой из битов не имеет действия. Для сброса статуса установите соответствующий бит в регистре EXTINT_CLR. Для установки статуса установите соответствующий бит в регистре EXTINT_TEST (возможность предусмотрена для отладочных целей).

В таблице 1.534 представлено описание полей регистра.

Таблица 1.534 – Описание полей регистра EXTINT_STATUS

Биты	Поле	Доступ	Значение после сброса	Описание
1:0	STATUS_CH0	ЧТ	0	бит [0] - обнаружен фронт или активный уровень сигнала в канале 0 (в зависимости от EXTINT_CFG_0[LVL_MODE]); бит [1] - обнаружен спад сигнала в канале 0.

Биты	Поле	Доступ	Значение после сброса	Описание
3:2	STATUS_CH1	ЧТ	0	бит [0] - обнаружен фронт или активный уровень сигнала в канале 1 (в зависимости от EXTINT_CFG_1[LVL_MODE]); бит [1] - обнаружен спад сигнала в канале 1.
…		ЧТ	0	аналогично для остальных каналов
31:30	STATUS_CH15	ЧТ	0	бит [0] - обнаружен фронт или активный уровень сигнала в канале 15 (в зависимости от EXTINT_CFG_15[LVL_MODE]); бит [1] - обнаружен спад сигнала в канале 15.

TrustZone: по 2 бита. Значение каждых двух бит (соответствуют одному каналу) будет актуальным, если настройки секьюрности и привилегированности этого канала ниже или совпадают с параметрами транзакции. Иначе чтение вернет 0.

1.3.10.3.4.6 Регистр EXTINT_TEST

Смещение адреса 0x014

Запись 0 в любой бит регистра не имеет действия.

В таблице 1.535 представлено описание полей регистра.

Таблица 1.535 – Описание полей регистра EXTINT_TEST

Биты	Поле	Доступ	Значение после сброса	Описание
1:0	SET_CH0	ЗП	0	бит [0] - запись 1 - установит в 1 бит EXTINT_STATUS[STATUS_CH0[0]]; бит [1] - запись 1 - установит в 1 бит EXTINT_STATUS[STATUS_CH0[1]].
3:2	SET_CH1	ЗП	0	бит [0] - запись 1 - установит в 1 бит EXTINT_STATUS[STATUS_CH1[0]]; бит [1] - запись 1 - установит в 1 бит EXTINT_STATUS[STATUS_CH1[1]].
…		ЗП	0	аналогично для остальных каналов
31:30	SET_CH15	ЗП	0	бит [0] - запись 1 - установит в 1 бит EXTINT_STATUS[STATUS_CH15[0]]; бит [1] - запись 1 - установит в 1 бит EXTINT_STATUS[STATUS_CH15[1]].

TrustZone: по 2 бита. Каждые два бита (соответствуют одному каналу) доступны, если настройки секьюрности и привилегированности их канала ниже или совпадают с параметрами транзакции.

1.3.10.3.4.7 Регистр EXTINT_CLR

Смещение адреса 0x018

Запись 0 в любой бит регистра не имеет действия.

В таблице 1.536 представлено описание полей регистра.

Таблица 1.536 – Описание полей регистра EXTINT_CLR

Биты	Поле	Доступ	Значение после сброса	Описание
1:0	CLR_CH0	ЗП	0	бит [0] - запись 1 - сбросит в 0 бит EXTINT_STATUS[STATUS_CH0[0]]; бит [1] - запись 1 - сбросит в 0 бит EXTINT_STATUS[STATUS_CH0[1]].
3:2	CLR_CH1	ЗП	0	бит [0] - запись 1 - сбросит в 0 бит EXTINT_STATUS[STATUS_CH1[0]]; бит [1] - запись 1 - сбросит в 0 бит EXTINT_STATUS[STATUS_CH1[1]].
…		ЗП	0	аналогично для остальных каналов
31:30	CLR_CH15	ЗП	0	бит [0] - запись 1 - сбросит в 0 бит EXTINT_STATUS[STATUS_CH15[0]]; бит [1] - запись 1 - сбросит в 0 бит EXTINT_STATUS[STATUS_CH15[1]].

TrustZone: по 2 бита. Каждые два бита (соответствуют одному каналу) доступны, если настройки секьюрности и привилегированности их канала ниже или совпадают с параметрами транзакции.

1.3.10.3.4.8 Регистр EXTINT_LVL_MASK

Смещение адреса 0x01C

В таблице 1.537 представлено описание полей регистра.

Таблица 1.537 – Описание полей регистра EXTINT_LVL_MASK

Биты	Поле	Доступ	Значение после сброса	Описание
1:0	MASK_CH0	ЧТ/ЗП	0	бит [0]: 1 - разрешение генерации уровневого события канала 0 при обнаружении фронта или активного уровня; 0 - запрещено; бит [1]: 1 - разрешение генерации уровневого события канала 0 при обнаружении спада; 0 - запрещено.
3:2	MASK_CH1	ЧТ/ЗП	0	бит [0]: 1 - разрешение генерации уровневого события канала 1 при обнаружении фронта или активного уровня; 0 - запрещено; бит [1]: 1 - разрешение генерации уровневого события канала 1 при обнаружении спада; 0 - запрещено.
…		ЧТ/ЗП	0	аналогично для остальных каналов

Биты	Поле	Доступ	Значение после сброса	Описание
31:30	MASK_CH15	ЧТ/ЗП	0	бит [0]: 1 - разрешение генерации уровневого события канала 15 при обнаружении фронта или активного уровня; 0 - запрещено; бит [1]: 1 - разрешение генерации уровневого события канала 15 при обнаружении спада; 0 - запрещено.

TrustZone: по 2 бита. Каждые два бита (соответствуют одному каналу) доступны, если настройки секьюрности и привилегированности их канала ниже или совпадают с параметрами транзакции.

1.3.10.3.4.9 Регистр EXTINT_SWTCH_MASK

Смещение адреса 0x024

В таблице 1.538 представлено описание полей регистра.

Таблица 1.538 – Описание полей регистра EXTINT_SWTCH_MASK

Биты	Поле	Доступ	Значение после сброса	Описание
1:0	MASK_CH0	ЧТ/ЗП	0	бит [0]: 1 - разрешение генерации переключаемого события канала 0 при обнаружении фронта или активного уровня; 0 - запрещено; бит [1]: 1 - разрешение генерации переключаемого события канала 0 при обнаружении спада; 0 - запрещено.
3:2	MASK_CH1	ЧТ/ЗП	0	бит [0]: 1 - разрешение генерации переключаемого события канала 1 при обнаружении фронта или активного уровня; 0 - запрещено; бит [1]: 1 - разрешение генерации переключаемого события канала 1 при обнаружении спада; 0 - запрещено.
…		ЧТ/ЗП	0	аналогично для остальных каналов
31:30	MASK_CH15	ЧТ/ЗП	0	бит [0]: 1 - разрешение генерации переключаемого события канала 15 при обнаружении фронта или активного уровня; 0 - запрещено; бит [1]: 1 - разрешение генерации переключаемого события канала 15 при обнаружении спада; 0 - запрещено.

TrustZone: по 2 бита. Каждые два бита (соответствуют одному каналу) доступны, если настройки секьюрности и привилегированности их канала ниже или совпадают с параметрами транзакции.

1.3.10.3.4.10 Регистр EXTINT_CMN_CFG

Смещение адреса 0x028

В таблице 1.539 представлено описание полей регистра.

Таблица 1.539 – Описание полей регистра EXTINT_CMN_CFG

Биты	Поле	Доступ	Значение после сброса	Описание
0	DIS_IP_MASK	ЧТ/ЗП	1	1 - внешнее аппаратное выключение EXTINT разрешено; 0 - запрещено.
31:1			0	зарезервировано

TrustZone: любой доступ.

1.3.10.3.4.11 Регистр EXTINT_SECURE

Смещение адреса 0x030

В таблице 1.540 представлено описание полей регистра.

Таблица 1.540 – Описание полей регистра EXTINT_SECURE

Биты	Поле	Доступ	Значение после сброса	Описание
15:0	SECURE	ЧТ/ЗП	0	Настройка секьюрности для каждого канала. 1 в бите [N] - ограничивает программный доступ к регистрам и полям, связанным с работой соответствующего канала N только секьюрными запросами; позволяет каналу N работать с секьюрным внешним сигналом. 0 в бите [N] - снимает ограничения доступа к регистрам; запрещает каналу N работать с секьюрным внешним сигналом.
31:16			0	зарезервировано

TrustZone: только секьюрный доступ на запись и чтение.

1.3.10.3.4.12 Регистр EXTINT_PRIVILEGE

Смещение адреса 0x034

В таблице 1.541 представлено описание полей регистра.

Таблица 1.541 – Описание полей регистра EXTINT_PRIVILEGE

Биты	Поле	Доступ	Значение после сброса	Описание
15:0	PRIVILEDGE	ЧТ/ЗП	0	Настройка привилегированности для каждого канала 1 в бите [N] - ограничивает программный доступ к регистрам и полям, связанным с работой соответствующего канала N только привилегированными запросами. 0 в бите [N] - снимает ограничения доступа к регистрам.
31:16			0	зарезервировано

TrustZone: только привилегированный доступ на запись и чтение.

1.3.10.3.4.13 Регистр EXTINT_TRUSTZONE_IRQ

Смещение адреса 0x038

В таблице 1.542 представлено описание полей регистра.

Таблица 1.542 – Описание полей регистра EXTINT_TRUSTZONE_IRQ

Биты	Поле	Доступ	Значение после сброса	Описание
0	IRQ	ЧТ/ЗП1О	0	Чтение: показывает статус ошибки TrustZone. Устанавливается при: - несоответствии настройки SECURE любого из каналов и настройки SECURE подключенного к нему внешнего сигнала. Устанавливается только при включенном канале. - программном доступе к регистру с бОльшими настройками секьюрности или привилегированности, чем у запроса Запись: 0 - не имеет действия 1 - сбросит этот статус
31:1			0	Зарезервировано

TrustZone: только секьюрный и привилегированный доступ на запись и чтение.

1.3.10.3.4.14 Регистр EXTINT_TRUSTZONE_MASK

Смещение адреса 0x03C

В таблице 1.543 представлено описание полей регистра.

Таблица 1.543 – Описание полей регистра EXTINT_TRUSTZONE_MASK

Биты	Поле	Доступ	Значение после сброса	Описание
0	MASK	ЧТ/ЗП	0	1 - разрешение генерации прерывания при ошибке TrustZone. 0 - прерывание запрещено;
31:1			0	Зарезервировано

TrustZone: только секьюрный и привилегированный доступ на запись и чтение.

1.3.10.3.4.15 Регистр EXTINT_CFG_0

Смещение адреса 0x040

Регистр настройки канала 0.

В таблице 1.544 представлено описание полей регистра.

Таблица 1.544 – Описание полей регистра EXTINT_CFG_0

Биты	Поле	Доступ	Значение после сброса	Описание
0	CH_EN	ЧТ/ЗП	0	включение канала. Если канал выключен, события не генерируются, соответствующий бит поля TMP_STATE всегда равен 0. 0 - выключен; 1 - включен.
1	FILTR_EN	ЧТ/ЗП	0	включение фильтра 0 - выключен; 1 - включен.
5:2	FILTR_TOL	ЧТ/ЗП	0xF	граница срабатывания фильтра 0 - зарезервировано 1..15 - разрешенные значения
7:6	FILTR_DIV	ЧТ/ЗП	0	частота выборки фильтра 0 - x1; 1 - x4; 2 - x16; 3 - x64.
8	LVL_MODE	ЧТ/ЗП	0	уровневый режим работы: 0 - выключен; 1 - включен. Описание приведено в 1.3.10.2.4.
9	LVL_POL	ЧТ/ЗП	1	активный уровень входного сигнала при включенном уровневом режиме работы: 0 - низкий; 1 - высокий.
10	SIG_CTRL	ЧТ/ЗП	0	управление настройками входящих сигналов (см. 1.3.10.5) 0 - выключен; 1 - включен.

Биты	Поле	Доступ	Значение после сброса	Описание
15:11			0	Зарезервировано
22:16	CH_MAP	ЧТ/ЗП	0	номер входного сигнала, подключаемого к каналу. Разрешенные значения 0..127. Описание входных сигналов приведено в 1.3.10.5.
31:23			0	Зарезервировано

TrustZone: доступен, если настройки секьюрности и привилегированности соответствующего канала ниже или совпадают с параметрами транзакции.

1.3.10.3.4.16 Регистры EXTINT_CFG_1 .. EXTINT_CFG_15

Смещения адресов 0x044 .. 0x07C

Регистры настройки каналов 1 .. 15.

Поля всех регистров аналогичны полям регистра EXTINT_CFG_0.

1.3.10.4 Указания к программированию

1.3.10.4.1 Ограничения

Настройки и включения/выключения разных каналов можно проводить независимо друг от друга.

Настройки и включения/выключения в пределах одного канала имеют следующие ограничения:

– изменения CH_MAP, FILTR_TOL, FILTR_DIV при установленном FILTR_EN или CH_EN запрещены;

– одновременное изменение (в одной команде) CH_MAP, FILTR_TOL, FILTR_DIV и установка/сброс CH_EN и FILTR_EN разрешены.

1.3.10.4.2 Работа с фильтром

Для повышения надёжности включения фильтра требуется следующая последовательность:

– установка требуемых CH_MAP, FILTR_TOL, FILTR_DIV и FILTR_EN=1;

– ожидание extint_clk*FILTR_TOL*FILTR_DIV;

– установка CH_EN=1.

1.3.10.4.3 Внешний сброс

1) Если сигнал уровневого или переключаемого выходного события находится в 1, то целевое устройство может некорректно обработать их переключение в 0 при внешнем сбросе. Поэтому перед осуществлением внешнего сброса EXTINT рекомендуется:

– замаскировать все события всех каналов;

– обработать все ожидающие события;

– убедиться, что целевые устройства, подключенные к сигналам переключаемых событий, не будут реагировать на входное событие (замаскировать входящие события внутри этих устройств или выключить устройства).

1.3.10.5 Интеграция контроллера EXTINT в МК Optimal+

Контроллер EXTINT используется в МК Optimal+ в трех экземплярах:

– EXTINT_MAIN (далее основной контроллер EXTINT) - контроллер EXTINT, обслуживающий все внешние сигнальные выводы МК Optimal+;

– EXTINT_ACOMP - контроллер EXTINT, обслуживающий аналоговый компаратор (ACOMP);

– EXTINT_AON - контроллер EXTINT, отвечающий за пробуждение от внешних сигналов в режиме SHUTDOWN.

Ниже приведены особенности интеграции каждого экземпляра контроллера EXTINT.

1.3.10.5.1 Основной контроллер EXTINT

Основной контроллер EXTINT реализует 16 каналов для передачи внутрь МК Optimal+ информации о внешних событиях.

Основной контроллер EXTINT подключается параллельно с контроллером GPIO к буферам ввода/вывода ,т.е. каждому внешнему сигнальному выводу соответствует входной сигнал основного контроллера EXTINT. Для именования внешних выводов основного контроллеров EXTINT в составе МК Optimal+ принята следующая схема: EXTINT_<_номер входного сигнала_> (например, 52-й входной сигнал основного контроллера EXTINT будет называться EXTINT_52).

Для настройки канала основного контроллера EXTINT необходимо в поле CH_MAP регистра EXTINT_CFG_i указать номер входного сигнала. Соответствие номера входного сигнала основного контроллера EXTINT и внешнего вывода МК Optimal+ указано в 1.2.2.

Параллельное подключение основного контроллера EXTINT и контроллера GPIO дополнительно позволяет:

– анализировать внешние выводы МК Optimal+ при работе альтернативных функций контроллера GPIO;

– оставаться активным даже при условии снятия сигнала синхронизации с контроллера GPIO для уменьшения энергопотребления.

Как следствие перечисленных возможностей основной контроллер EXTINT может использоваться как контроллер пробуждения (Wake Up) по различным интерфейсам в любых режимах энергопотребления, не предполагающих отклюление HLDO/DCDC. Например, пробуждение контроллера USART выполняется при переключении из «1» в «0» входа RX, что может отслеживаться основноым контроллером EXTINT даже при выключенном контроллере USART.

1.3.10.5.2 EXTINT_ACOMP

Контроллер EXTINT_ACOMP используется для генерации запросов прерывания аналоговыми компараторами (ACOMP). Контроллер EXTINT_ACOMP реализует 2 канала и 2 входных сигнала, подключенные к ACOMP0 и ACOMP1, соответственно.

1.3.10.5.3 EXTINT_AON

Контроллер EXTINT_AON расположен в AON и реализует 8 каналов для пробуждения МК Optimal+ от внешних сигналов в режиме энергопотребления SHUTDOWN.

Входными сигналами для контроллера являются все выводы, помеченные в 1.2.2 как WKP_x.

1.3.11 Подсистема прямого обмена событиями (ELC)

1.3.11.1 Общее описание

Для ускорения взаимодействия между блоками в составе МК Optimal+ используется подсистема прямого обмена событиями. Центральным узлом данной подсистемы является контроллер управления событиями ELC. Задачей ELC является коммутация блоков, генерирующих события (events), и блоков, реагирующих на эти события (triggers).

Под *событием* понимается входящий в ELC сигнал, информирующий о значимом изменении состояния некоторого другого СФ-блока, на который возможна какая-либо реакция, например, получение данных интерфейсным контроллером.

Под *триггером* понимается исходящий из ELC сигнал, инициирующий выполнение какой-либо операции блоком, специально предназначенным для этого, например, запуск работы канала GPDMA.

Блок ELC обладает следующими характеристиками:

– число событий: 138 (уточняется на этапе разработки КД) + событие №0;

– число триггеров: 69 (уточняется на этапе разработки КД);

– независимая настройка триггеров;

– каждый триггер может быть запущен программно;

– каждый триггер может работать в защищённом режиме;

– возможность блокировки переконфигурации блока;

– интерфейс программирования: APB slave.

1.3.11.1.1 Устройство и функционирование блока

Со стороны событий ELC выполняет только роль коммутации, не внося задержку и не изменяя форму сигнала.

Каждому событию соответствует номер (см. 1.3.11.2). Контроллеры GPIO не подключены к ELC, функцию генерации событий при изменении входов микроконтроллера МК Optimal+ выполняет контроллер EXTINT (среди прочего, подключенный к ELC).

Каждому триггеру соответствует программно доступный регистр, в котором указывается номер события, которое должно коммутроваться к этому триггеру. Каждый триггер соединён с соответствующим портом блока, реагирующего на события (см. 1.3.11.3). Событие №0 используется для выключения триггера ELC, т.е. если в поле ELC_TRIG_CFG[MAP] триггера содержится значение 0, то триггер не передает входящие события (тем не менее может быть сгенерирован программно).

Триггер может ошибочно сработать при изменении поля ELC_TRIG_CFG.MAP или сбросе блока ELC. Поэтому необходимо предварительно замаскировать его в целевом блоке.

Схема генерации триггеров приведена на рисунке 1.136.

Параметры:
N = 206 – количество входных событий
M = 69 – количество выходных триггеров

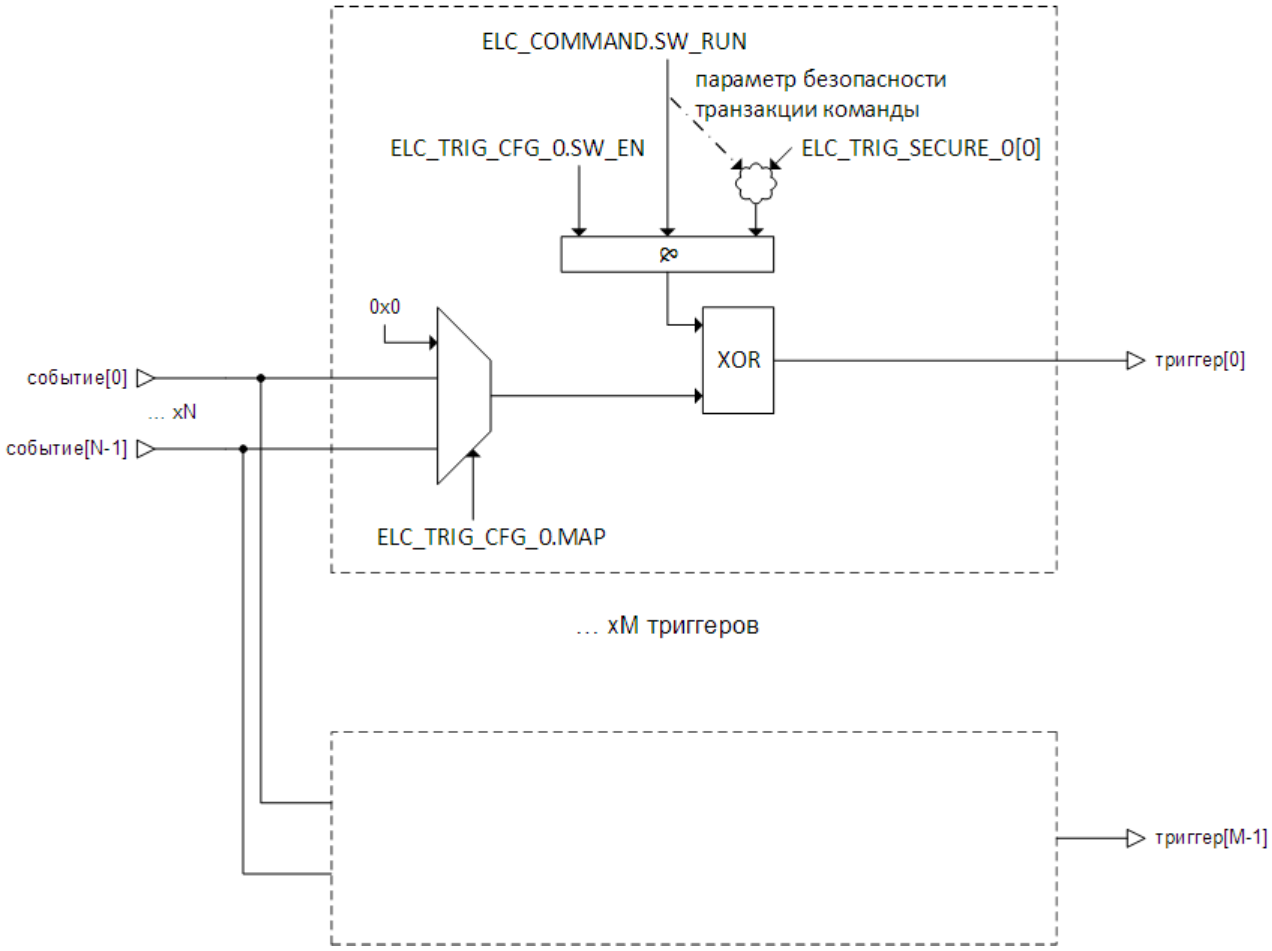


Рисунок 1.136 – Схема генерации триггеров в ELC

1.3.11.1.2 Программный запуск триггеров

Каждый триггер может быть запущен программно. Для этого надо:

– установить разрешение в бите SW_EN в соответствующем регистре ELC_TRIG_CFG;

– запустить программную генерацию триггеров, установив бит SW_RUN в регистре ELC_COMMAND.

Если в процессе программной генерации триггера произошло его входное событие, то они оба могут быть потеряны. Поэтому при программной генерации пользователь должен обеспечить отсутствие соответствующего входного события.

Одна команда установки бита SW_RUN генерирует триггер однократно. Для повторной генерации надо повторно установить бит SW_RUN.

Одновременно могут быть установлены биты SW_EN для любого количества триггеров, что приведет к одновременной их генерации при установке SW_RUN.

1.3.11.1.3 Безопасный режим

Для каждого триггера можно включить безопасный режим работы, установив соответствующий бит в регистре ELC_TRIG_SECURE_x.

В безопасном режиме:

– настройка триггера может выполняться только секьюрными транзакциями;

– программный запуск может выполняться только секьюрными транзакциями.

1.3.11.1.4 Блокировка изменения конфигурации

В ELC предусмотрена возможность программной блокировки записи в конфигурационные регистров с целью их защиты.

Включается блокировка записью константы 0x10CC в поле ELC_LOCK[LOCK]. При этом регистры ELC_TRIG_SECURE и ELC_TRIG_CFG станут недоступны на запись. Программная генерация триггеров будет по-прежнему возможна.

Снятие блокировки возможно только сбросом.

1.3.11.2 Список событий

Список событий и блоков, являющихся источниками данных событий, приведён в таблице 1.545

Таблица 1.545 – Список событий блока ELC

Номер события	Событие	Источник	Описание
0	nop	ELC	Ничего не выполнять
1	systim_event0	SYSTIM	Событие от 0 канала системного таймера
2	systim_event1	SYSTIM	Событие от 1 канала системного таймера
3	iwdt_event	IWDT	Событие IWDT
4	wwdt_event	WWDT	Событие WWDT
5	rtc_event	RTC	(НЕ ПОДКЛЮЧЕНО) Событие RTC
6	cfau_int	CFAU	(НЕ ПОДКЛЮЧЕНО) Прерывание от CFAU
7	pwd_int	PVD	(НЕ ПОДКЛЮЧЕНО) Прерывание от PVD
8	extint_0	EXTINT	Событие на линии 0 контроллера EXTINT
9	extint_1	EXTINT	Событие на линии 1 контроллера EXTINT
10	extint_2	EXTINT	Событие на линии 2 контроллера EXTINT
11	extint_3	EXTINT	Событие на линии 3 контроллера EXTINT

Номер события	Событие	Источник	Описание
12	extint_4	EXTINT	Событие на линии 4 контроллера EXTINT
13	extint_5	EXTINT	Событие на линии 5 контроллера EXTINT
14	extint_6	EXTINT	Событие на линии 6 контроллера EXTINT
15	extint_7	EXTINT	Событие на линии 7 контроллера EXTINT
16	extint_8	EXTINT	Событие на линии 8 контроллера EXTINT
17	extint_9	EXTINT	Событие на линии 9 контроллера EXTINT
18	extint_10	EXTINT	Событие на линии 10 контроллера EXTINT
19	extint_11	EXTINT	Событие на линии 11 контроллера EXTINT
20	extint_12	EXTINT	Событие на линии 12 контроллера EXTINT
21	extint_13	EXTINT	Событие на линии 13 контроллера EXTINT
22	extint_14	EXTINT	Событие на линии 14 контроллера EXTINT
23	extint_15	EXTINT	Событие на линии 15 контроллера EXTINT
24	gpdma_0_event0	GPDMA_0	(НЕ ПОДКЛЮЧЕНО) Событие от 0 канала блока GPDMA_0
25	gpdma_0_event1	GPDMA_0	(НЕ ПОДКЛЮЧЕНО) Событие от 1 канала блока GPDMA_0
26	gpdma_0_event2	GPDMA_0	(НЕ ПОДКЛЮЧЕНО) Событие от 2 канала блока GPDMA_0
27	gpdma_0_event3	GPDMA_0	(НЕ ПОДКЛЮЧЕНО) Событие от 3 канала блока GPDMA_0
28	gpdma_0_event4	GPDMA_0	(НЕ ПОДКЛЮЧЕНО) Событие от 4 канала блока GPDMA_0
29	gpdma_0_event5	GPDMA_0	(НЕ ПОДКЛЮЧЕНО) Событие от 5 канала блока GPDMA_0
30	gpdma_0_event6	GPDMA_0	(НЕ ПОДКЛЮЧЕНО) Событие от 6 канала блока GPDMA_0
31	gpdma_0_event7	GPDMA_0	(НЕ ПОДКЛЮЧЕНО) Событие от 7 канала блока GPDMA_0
32	gpdma_1_event0	GPDMA_1	(НЕ ПОДКЛЮЧЕНО) Событие от 0 канала блока GPDMA_1
33	gpdma_1_event1	GPDMA_1	(НЕ ПОДКЛЮЧЕНО) Событие от 1 канала блока GPDMA_1
34	gpdma_1_event2	GPDMA_1	(НЕ ПОДКЛЮЧЕНО) Событие от 2 канала блока GPDMA_1
35	gpdma_1_event3	GPDMA_1	(НЕ ПОДКЛЮЧЕНО) Событие от 3 канала блока GPDMA_1
36	gpdma_1_event4	GPDMA_1	(НЕ ПОДКЛЮЧЕНО) Событие от 4 канала блока GPDMA_1
37	gpdma_1_event5	GPDMA_1	(НЕ ПОДКЛЮЧЕНО) Событие от 5 канала блока GPDMA_1
38	gpdma_1_event6	GPDMA_1	(НЕ ПОДКЛЮЧЕНО) Событие от 6 канала блока GPDMA_1
39	gpdma_1_event7	GPDMA_1	(НЕ ПОДКЛЮЧЕНО) Событие от 7 канала блока GPDMA_1

Номер события	Событие	Источник	Описание
40	acomp_0	ACOMP	(НЕ ПОДКЛЮЧЕНО) Событие 0 от аналогового компаратора ACOMP
41	acomp_1	ACOMP	(НЕ ПОДКЛЮЧЕНО) Событие 1 от аналогового компаратора ACOMP
42	adc_0	ADC	(НЕ ПОДКЛЮЧЕНО) Событие 0 от ADC
43	cmu_event	CMU	(НЕ ПОДКЛЮЧЕНО) Событие от CMU
44	tim_a0_ch_out0	TIM_A0	Событие канала 0 блока TIM_A0
45	tim_a0_ch_out1	TIM_A0	Событие канала 1 блока TIM_A0
46	tim_a0_ch_out2	TIM_A0	Событие канала 2 блока TIM_A0
47	tim_a0_ch_out3	TIM_A0	Событие канала 3 блока TIM_A0
48	tim_a0_trigger	TIM_A0	Особое событие блока TIM_A0
49	tim_a1_ch_out0	TIM_A1	Событие канала 0 блока TIM_A1
50	tim_a1_ch_out1	TIM_A1	Событие канала 1 блока TIM_A1
51	tim_a1_ch_out2	TIM_A1	Событие канала 2 блока TIM_A1
52	tim_a1_ch_out3	TIM_A1	Событие канала 3 блока TIM_A1
53	tim_a1_trigger	TIM_A1	Особое событие блока TIM_A1
54	tim_b0_ch_out0	TIM_B0	Событие канала 0 блока TIM_B0
55	tim_b0_ch_out1	TIM_B0	Событие канала 1 блока TIM_B0
56	tim_b0_ch_out2	TIM_B0	Событие канала 2 блока TIM_B0
57	tim_b0_ch_out3	TIM_B0	Событие канала 3 блока TIM_B0
58	tim_b0_trigger	TIM_B0	Особое событие блока TIM_B0
59	tim_b1_ch_out0	TIM_B1	Событие канала 0 блока TIM_B1
60	tim_b1_ch_out1	TIM_B1	Событие канала 1 блока TIM_B1
61	tim_b1_ch_out2	TIM_B1	Событие канала 2 блока TIM_B1
62	tim_b1_ch_out3	TIM_B1	Событие канала 3 блока TIM_B1
63	tim_b1_trigger	TIM_B1	Особое событие блока TIM_B1
64	tim_c0_ch_out0	TIM_C0	Событие канала 0 блока TIM_C0
65	tim_c0_ch_out1	TIM_C0	Событие канала 1 блока TIM_C0
66	tim_c0_ch_out2	TIM_C0	Событие канала 2 блока TIM_C0
67	tim_c0_ch_out3	TIM_C0	Событие канала 3 блока TIM_C0
68	tim_c0_trigger	TIM_C0	Особое событие блока TIM_C0
69	tim_c1_ch_out0	TIM_C1	Событие канала 0 блока TIM_C1
70	tim_c1_ch_out1	TIM_C1	Событие канала 1 блока TIM_C1
71	tim_c1_ch_out2	TIM_C1	Событие канала 2 блока TIM_C1
72	tim_c1_ch_out3	TIM_C1	Событие канала 3 блока TIM_C1
73	tim_c1_trigger	TIM_C1	Особое событие блока TIM_C1
74	tim_d0_ch_out0	TIM_D0	Событие канала 0 блока TIM_D0
75	tim_d0_ch_out1	TIM_D0	Событие канала 1 блока TIM_D0
76	tim_d0_ch_out2	TIM_D0	Событие канала 2 блока TIM_D0
77	tim_d0_ch_out3	TIM_D0	Событие канала 3 блока TIM_D0

Номер события	Событие	Источник	Описание
78	tim_d0_trigger	TIM_D0	Особое событие блока TIM_D0
79	tim_d1_ch_out0	TIM_D1	Событие канала 0 блока TIM_D1
80	tim_d1_ch_out1	TIM_D1	Событие канала 1 блока TIM_D1
81	tim_d1_ch_out2	TIM_D1	Событие канала 2 блока TIM_D1
82	tim_d1_ch_out3	TIM_D1	Событие канала 3 блока TIM_D1
83	tim_d1_trigger	TIM_D1	Особое событие блока TIM_D1
84	usart_a0_rx_wm	USART_A0	(НЕ ПОДКЛЮЧЕНО) RX буфер USART_A0 наполнен до заданного уровня
85	usart_a0_tx_wm	USART_A0	(НЕ ПОДКЛЮЧЕНО) TX буфер USART_A0 опустошён до заданного уровня
86	usart_a1_rx_wm	USART_A1	(НЕ ПОДКЛЮЧЕНО) RX буфер USART_A1 наполнен до заданного уровня
87	usart_a1_tx_wm	USART_A1	(НЕ ПОДКЛЮЧЕНО) TX буфер USART_A1 опустошён до заданного уровня
88	usart_b0_rx_wm	USART_B0	(НЕ ПОДКЛЮЧЕНО) RX буфер USART_B0 наполнен до заданного уровня
89	usart_b0_tx_wm	USART_B0	(НЕ ПОДКЛЮЧЕНО) TX буфер USART_B0 опустошён до заданного уровня
90	usart_b1_rx_wm	USART_B1	(НЕ ПОДКЛЮЧЕНО) RX буфер USART_B1 наполнен до заданного уровня
91	usart_b1_tx_wm	USART_B1	(НЕ ПОДКЛЮЧЕНО) TX буфер USART_B1 опустошён до заданного уровня
92	usart_c0_rx_wm	USART_C0	(НЕ ПОДКЛЮЧЕНО) RX буфер USART_C0 наполнен до заданного уровня
93	usart_c0_tx_wm	USART_C0	(НЕ ПОДКЛЮЧЕНО) TX буфер USART_C0 опустошён до заданного уровня
94	usart_c1_rx_wm	USART_C1	(НЕ ПОДКЛЮЧЕНО) RX буфер USART_C1 наполнен до заданного уровня
95	usart_c1_tx_wm	USART_C1	(НЕ ПОДКЛЮЧЕНО) TX буфер USART_C1 опустошён до заданного уровня
96	usart_d0_rx_wm	USART_D0	(НЕ ПОДКЛЮЧЕНО) RX буфер USART_D0 наполнен до заданного уровня
97	usart_d0_tx_wm	USART_D0	(НЕ ПОДКЛЮЧЕНО) TX буфер USART_D0 опустошён до заданного уровня
98	usart_d1_rx_wm	USART_D1	(НЕ ПОДКЛЮЧЕНО) RX буфер USART_D1 наполнен до заданного уровня
99	usart_d1_tx_wm	USART_D1	(НЕ ПОДКЛЮЧЕНО) TX буфер USART_D1 опустошён до заданного уровня
100	spi_a0_tx_event	SPI_A0	(НЕ ПОДКЛЮЧЕНО) SPI_a0 событие передачи
101	spi_a0_rx_event	SPI_A0	(НЕ ПОДКЛЮЧЕНО) SPI_a0 событие приема
102	spi_b0_tx_event	SPI_B0	(НЕ ПОДКЛЮЧЕНО) SPI_b0 событие передачи
103	spi_b0_rx_event	SPI_B0	(НЕ ПОДКЛЮЧЕНО) SPI_b0 событие приема
104	spi_c0_tx_event	SPI_C0	(НЕ ПОДКЛЮЧЕНО) SPI_c0 событие передачи
105	spi_c0_rx_event	SPI_C0	(НЕ ПОДКЛЮЧЕНО) SPI_c0 событие приема

Номер события	Событие	Источник	Описание
106	spi_d0_tx_event	SPI_D0	(НЕ ПОДКЛЮЧЕНО) SPI_d0 событие передачи
107	spi_d0_rx_event	SPI_D0	(НЕ ПОДКЛЮЧЕНО) SPI_d0 событие приема
108	i2c_a0_tr_cmpl	I2C_A0	(НЕ ПОДКЛЮЧЕНО) I2C_A0 транзакция завершена
109	i2c_b0_tr_cmpl	I2C_B0	(НЕ ПОДКЛЮЧЕНО) I2C_B0 транзакция завершена
110	i2c_c0_tr_cmpl	I2C_C0	(НЕ ПОДКЛЮЧЕНО) I2C_C0 транзакция завершена
111	i2c_d0_tr_cmpl	I2C_D0	(НЕ ПОДКЛЮЧЕНО) I2C_D0 транзакция завершена
112	can_a0_rx_wm	CAN_A0	(НЕ ПОДКЛЮЧЕНО) RX буфер CAN_A0 наполнен до заданного уровня
113	can_a0_rx_full	CAN_A0	(НЕ ПОДКЛЮЧЕНО) RX буфер CAN_A0 полон
114	can_a0_rx_noempty	CAN_A0	(НЕ ПОДКЛЮЧЕНО) RX буфер CAN_A0 не пуст
115	can_a0_tx_wm	CAN_A0	(НЕ ПОДКЛЮЧЕНО) TX буфер CAN_A0 опустошён до заданного уровня
116	can_a0_tx_empty	CAN_A0	(НЕ ПОДКЛЮЧЕНО) TX буфер CAN_A0 пуст
117	can_b0_rx_wm	CAN_B0	(НЕ ПОДКЛЮЧЕНО) RX буфер CAN_B0 наполнен до заданного уровня
118	can_b0_rx_full	CAN_B0	(НЕ ПОДКЛЮЧЕНО) RX буфер CAN_B0 полон
119	can_b0_rx_noempty	CAN_B0	(НЕ ПОДКЛЮЧЕНО) RX буфер CAN_B0 не пуст
120	can_b0_tx_wm	CAN_B0	(НЕ ПОДКЛЮЧЕНО) TX буфер CAN_B0 опустошён до заданного уровня
121	can_b0_tx_empty	CAN_B0	(НЕ ПОДКЛЮЧЕНО) TX буфер CAN_B0 пуст
122	can_c0_rx_wm	CAN_C0	(НЕ ПОДКЛЮЧЕНО) RX буфер CAN_C0 наполнен до заданного уровня
123	can_c0_rx_full	CAN_C0	(НЕ ПОДКЛЮЧЕНО) RX буфер CAN_C0 полон
124	can_c0_rx_noempty	CAN_C0	(НЕ ПОДКЛЮЧЕНО) RX буфер CAN_C0 не пуст
125	can_c0_tx_wm	CAN_C0	(НЕ ПОДКЛЮЧЕНО) TX буфер CAN_C0 опустошён до заданного уровня
126	can_c0_tx_empty	CAN_C0	(НЕ ПОДКЛЮЧЕНО) TX буфер CAN_C0 пуст
127	can_d0_rx_wm	CAN_D0	(НЕ ПОДКЛЮЧЕНО) RX буфер CAN_D0 наполнен до заданного уровня
128	can_d0_rx_full	CAN_D0	(НЕ ПОДКЛЮЧЕНО) RX буфер CAN_D0 полон
129	can_d0_rx_noempty	CAN_D0	(НЕ ПОДКЛЮЧЕНО) RX буфер CAN_D0 не пуст
130	can_d0_tx_wm	CAN_D0	(НЕ ПОДКЛЮЧЕНО) TX буфер CAN_D0 опустошён до заданного уровня
131	can_d0_tx_empty	CAN_D0	(НЕ ПОДКЛЮЧЕНО) TX буфер CAN_D0 пуст

Номер события	Событие	Источник	Описание
132	lcd_new	LCD	Буфер контроллера LCD готов для приёма нового кадра
133	xspi_0_int	xSPI_0	(НЕ ПОДКЛЮЧЕНО) Прерывание от xSPI_0
134	xspi_1_int	xSPI_1	(НЕ ПОДКЛЮЧЕНО) Прерывание от xSPI_1
135	emmc_0_int	eMMC_0	Прерывание от eMMC_0
136	emmc_1_int	eMMC_1	Прерывание от eMMC_1
137	usb_otg_int	USB_OTG	Прерывание от USB_OTG
138	eth_int	ETH	Прерывание от ETH

1.3.11.3 Список триггеров

Список триггеров и блоков, реагирующих на данные триггеры, приведён в таблице 1.546

Таблица 1.546 – Список триггеров блока ELC

Номер триггера	Триггер	Получатель	Описание
0	gpdma_0_trigger0	GPDMA_0	(НЕ ПОДКЛЮЧЕНО) Запуск 0 канала блока GPDMA_0
1	gpdma_0_trigger1	GPDMA_0	(НЕ ПОДКЛЮЧЕНО) Запуск 1 канала блока GPDMA_0
2	gpdma_0_trigger2	GPDMA_0	(НЕ ПОДКЛЮЧЕНО) Запуск 2 канала блока GPDMA_0
3	gpdma_0_trigger3	GPDMA_0	(НЕ ПОДКЛЮЧЕНО) Запуск 3 канала блока GPDMA_0
4	gpdma_0_trigger4	GPDMA_0	(НЕ ПОДКЛЮЧЕНО) Запуск 4 канала блока GPDMA_0
5	gpdma_0_trigger5	GPDMA_0	(НЕ ПОДКЛЮЧЕНО) Запуск 5 канала блока GPDMA_0
6	gpdma_0_trigger6	GPDMA_0	(НЕ ПОДКЛЮЧЕНО) Запуск 6 канала блока GPDMA_0
7	gpdma_0_trigger7	GPDMA_0	(НЕ ПОДКЛЮЧЕНО) Запуск 7 канала блока GPDMA_0
8	gpdma_1_trigger0	GPDMA_1	(НЕ ПОДКЛЮЧЕНО) Запуск 0 канала блока GPDMA_1
9	gpdma_1_trigger1	GPDMA_1	(НЕ ПОДКЛЮЧЕНО) Запуск 1 канала блока GPDMA_1
10	gpdma_1_trigger2	GPDMA_1	(НЕ ПОДКЛЮЧЕНО) Запуск 2 канала блока GPDMA_1
11	gpdma_1_trigger3	GPDMA_1	(НЕ ПОДКЛЮЧЕНО) Запуск 3 канала блока GPDMA_1
12	gpdma_1_trigger4	GPDMA_1	(НЕ ПОДКЛЮЧЕНО) Запуск 4 канала блока GPDMA_1
13	gpdma_1_trigger5	GPDMA_1	(НЕ ПОДКЛЮЧЕНО) Запуск 5 канала блока GPDMA_1

Номер триггера	Триггер	Получатель	Описание
14	gpdma_1_trigger6	GPDMA_1	(НЕ ПОДКЛЮЧЕНО) Запуск 6 канала блока GPDMA_1
15	gpdma_1_trigger7	GPDMA_1	(НЕ ПОДКЛЮЧЕНО) Запуск 7 канала блока GPDMA_1
16	tim_0_ch_in0	TIM_0	Запуск канала 0 блока TIM_0
17	tim_0_ch_in1	TIM_0	Запуск канала 1 блока TIM_0
18	tim_0_ch_in2	TIM_0	Запуск канала 2 блока TIM_0
19	tim_0_ch_in3	TIM_0	Запуск канала 3 блока TIM_0
20	tim_0_brek	TIM_0	Команда break блоку TIM_0
21	tim_0_ext_event	TIM_0	Внешнее событие блока TIM_0
22	tim_1_ch_in0	TIM_1	Запуск канала 0 блока TIM_1
23	tim_1_ch_in1	TIM_1	Запуск канала 1 блока TIM_1
24	tim_1_ch_in2	TIM_1	Запуск канала 2 блока TIM_1
25	tim_1_ch_in3	TIM_1	Запуск канала 3 блока TIM_1
26	tim_1_brek	TIM_1	Команда break блоку TIM_1
27	tim_1_ext_event	TIM_1	Внешнее событие блока TIM_1
28	tim_2_ch_in0	TIM_2	Запуск канала 0 блока TIM_2
29	tim_2_ch_in1	TIM_2	Запуск канала 1 блока TIM_2
30	tim_2_ch_in2	TIM_2	Запуск канала 2 блока TIM_2
31	tim_2_ch_in3	TIM_2	Запуск канала 3 блока TIM_2
32	tim_2_brek	TIM_2	Команда break блоку TIM_2
33	tim_2_ext_event	TIM_2	Внешнее событие блока TIM_2
34	tim_3_ch_in0	TIM_3	Запуск канала 0 блока TIM_3
35	tim_3_ch_in1	TIM_3	Запуск канала 1 блока TIM_3
36	tim_3_ch_in2	TIM_3	Запуск канала 2 блока TIM_3
37	tim_3_ch_in3	TIM_3	Запуск канала 3 блока TIM_3
38	tim_3_brek	TIM_3	Команда break блоку TIM_3
39	tim_3_ext_event	TIM_3	Внешнее событие блока TIM_3
40	tim_4_ch_in0	TIM_4	Запуск канала 0 блока TIM_4
41	tim_4_ch_in1	TIM_4	Запуск канала 1 блока TIM_4
42	tim_4_ch_in2	TIM_4	Запуск канала 2 блока TIM_4
43	tim_4_ch_in3	TIM_4	Запуск канала 3 блока TIM_4
44	tim_4_brek	TIM_4	Команда break блоку TIM_4
45	tim_4_ext_event	TIM_4	Внешнее событие блока TIM_4
46	tim_5_ch_in0	TIM_5	Запуск канала 0 блока TIM_5
47	tim_5_ch_in1	TIM_5	Запуск канала 1 блока TIM_5
48	tim_5_ch_in2	TIM_5	Запуск канала 2 блока TIM_5
49	tim_5_ch_in3	TIM_5	Запуск канала 3 блока TIM_5
50	tim_5_brek	TIM_5	Команда break блоку TIM_5
51	tim_5_ext_event	TIM_5	Внешнее событие блока TIM_5

Номер триггера	Триггер	Получатель	Описание
52	tim_6_ch_in0	TIM_6	Запуск канала 0 блока TIM_6
53	tim_6_ch_in1	TIM_6	Запуск канала 1 блока TIM_6
54	tim_6_ch_in2	TIM_6	Запуск канала 2 блока TIM_6
55	tim_6_ch_in3	TIM_6	Запуск канала 3 блока TIM_6
56	tim_6_brek	TIM_6	Команда break блоку TIM_6
57	tim_6_ext_event	TIM_6	Внешнее событие блока TIM_6
58	tim_7_ch_in0	TIM_7	Запуск канала 0 блока TIM_7
59	tim_7_ch_in1	TIM_7	Запуск канала 1 блока TIM_7
60	tim_7_ch_in2	TIM_7	Запуск канала 2 блока TIM_7
61	tim_7_ch_in3	TIM_7	Запуск канала 3 блока TIM_7
62	tim_7_brek	TIM_7	Команда break блоку TIM_7
63	tim_7_ext_event	TIM_7	Внешнее событие блока TIM_7
64	acomp_trigger	ACOMP	(НЕ ПОДКЛЮЧЕНО) Запуск аналогового компоратора ACOMP
65	adc_trigger0	ADC	Запуск 0 канала ADC
66	adc_trigger1	ADC	Запуск 1 канала ADC
67	adc_trigger2	ADC	Запуск 3 канала ADC
68	dac_trigger0	DAC	Запуск 0 канала DAC
69	dac_trigger1	DAC	Запуск 1 канала DAC

1.3.11.4 Описание регистров

1.3.11.4.1 Общее описание регистров

Контроллер управления событиями ELC имеет набор программно доступных регистров, требующий 4 КБайт адресов на системной шине APB.

Карта регистров приведена в таблице 1.547.

При дальнейшем описании отдельных полей регистров используется следующее обозначение режимов доступа пользователя к полям:

– ЧТ – поле доступно только для чтения. Запись данных в поле игнорируется;

– ЗП – поле доступно только для записи. Чтение возвращает значение 0;

– ЧТ/ЗП – поле доступно как для записи, так и для чтения;

– ЧТ/ЗП1О – поле доступно для чтения. Запись 1 сбросит значение, запись 0 не имеет действия.

Доступ к регистрам осуществляется посредством системного интерфейса управления (APB slave). Поддерживается только 32-х разрядный доступ. Байтовая или двух-байтовая запись будет проигнорирована и возвращена ошибка шины, что вызовет переход процессорного ядра в исключение.

Запись в зарезервированные или ЧТ биты или адреса не имеет действия, никакие ошибки не генерируются. Чтение из зарезервированных или ЗП битов или адресов возвращает 0, никакие ошибки не генерируются.

1.3.11.4.2 Поддержка безопасности регистров

Некоторые регистры ELC доступны только в безопасном режиме работы.

Некоторые регистры ELC имеют наборы полей, доступ к которым зависит от их настроек и может быть разным в пределах одного регистра.

Возможности поддержки безопасности каждого конкретного регистра приведены в его описании (TrustZone).

Secure доступ к non-secure регистрам разрешен. Privileged доступ к non-privileged регистрам разрешен.

Non-secure доступ к secure регистрам вызовет прерывание и не будет иметь действия (при записи) или вернет 0 (при чтении). Non-privileged доступ к priviledeg регистрам вызовет прерывание и не будет иметь действия (при записи) или вернет 0 (при чтении).

1.3.11.4.3 Карта регистров

Таблица 1.547 – Карта регистров ELC

Адрес	Название	Доступ	Значение после сброса	Описание
0x000	ELC_ID	ЧТ	0x5f434c45	Идентификатор устройства
0x004	ELC_VERSION	ЧТ	0x00000001	Номер версии устройства и конфигурация
0x008	ELC_COMMAND	ЗП	0x0	Программные команды
0x00C	ELC_LOCK	ЧТ/ЗП	0x0	Запрет перенастройки буферов ввода/вывода (действует до аппаратного сброса)
0x010	ELC_TRIG_SECURE_0	ЧТ/ЗП	0x0	Безопасный режим для триггеров 0-31
0x014	ELC_TRIG_SECURE_1	ЧТ/ЗП	0x0	Безопасный режим для триггеров 32-63
0x018	ELC_TRIG_SECURE_2	ЧТ/ЗП	0x0	Безопасный режим для триггеров 64-68
0x01C		-	-	Резерв
0x020	ELC_TRIG_CFG_0	ЧТ/ЗП	0x0	Настройка триггера 0
0x024	ELC_TRIG_CFG_1	ЧТ/ЗП	0x0	Настройка триггера 1
0x028	ELC_TRIG_CFG_2	ЧТ/ЗП	0x0	Настройка триггера 2
0x02C	ELC_TRIG_CFG_3	ЧТ/ЗП	0x0	Настройка триггера 3
0x030	ELC_TRIG_CFG_4	ЧТ/ЗП	0x0	Настройка триггера 4
0x034	ELC_TRIG_CFG_5	ЧТ/ЗП	0x0	Настройка триггера 5
0x038	ELC_TRIG_CFG_6	ЧТ/ЗП	0x0	Настройка триггера 6
0x03C	ELC_TRIG_CFG_7	ЧТ/ЗП	0x0	Настройка триггера 7

Адрес	Название	До-ступ	Значение по-сле сброса	Описание
0x040	ELC_TRIG_CFG_8	ЧТ/ЗП	0x0	Настройка триггера 8
0x044	ELC_TRIG_CFG_9	ЧТ/ЗП	0x0	Настройка триггера 9
0x048	ELC_TRIG_CFG_10	ЧТ/ЗП	0x0	Настройка триггера 10
0x04C	ELC_TRIG_CFG_11	ЧТ/ЗП	0x0	Настройка триггера 11
0x050	ELC_TRIG_CFG_12	ЧТ/ЗП	0x0	Настройка триггера 12
0x054	ELC_TRIG_CFG_13	ЧТ/ЗП	0x0	Настройка триггера 13
0x058	ELC_TRIG_CFG_14	ЧТ/ЗП	0x0	Настройка триггера 14
0x05C	ELC_TRIG_CFG_15	ЧТ/ЗП	0x0	Настройка триггера 15
0x060	ELC_TRIG_CFG_16	ЧТ/ЗП	0x0	Настройка триггера 16
0x064	ELC_TRIG_CFG_17	ЧТ/ЗП	0x0	Настройка триггера 17
0x068	ELC_TRIG_CFG_18	ЧТ/ЗП	0x0	Настройка триггера 18
0x06C	ELC_TRIG_CFG_19	ЧТ/ЗП	0x0	Настройка триггера 19
0x070	ELC_TRIG_CFG_20	ЧТ/ЗП	0x0	Настройка триггера 20
0x074	ELC_TRIG_CFG_21	ЧТ/ЗП	0x0	Настройка триггера 21
0x078	ELC_TRIG_CFG_22	ЧТ/ЗП	0x0	Настройка триггера 22
0x07C	ELC_TRIG_CFG_23	ЧТ/ЗП	0x0	Настройка триггера 23
0x080	ELC_TRIG_CFG_24	ЧТ/ЗП	0x0	Настройка триггера 24
0x084	ELC_TRIG_CFG_25	ЧТ/ЗП	0x0	Настройка триггера 25
0x088	ELC_TRIG_CFG_26	ЧТ/ЗП	0x0	Настройка триггера 26
0x08C	ELC_TRIG_CFG_27	ЧТ/ЗП	0x0	Настройка триггера 27
0x090	ELC_TRIG_CFG_28	ЧТ/ЗП	0x0	Настройка триггера 28
0x094	ELC_TRIG_CFG_29	ЧТ/ЗП	0x0	Настройка триггера 29
0x098	ELC_TRIG_CFG_30	ЧТ/ЗП	0x0	Настройка триггера 30
0x09C	ELC_TRIG_CFG_31	ЧТ/ЗП	0x0	Настройка триггера 31
0x0A0	ELC_TRIG_CFG_32	ЧТ/ЗП	0x0	Настройка триггера 32
0x0A4	ELC_TRIG_CFG_33	ЧТ/ЗП	0x0	Настройка триггера 33
0x0AC	ELC_TRIG_CFG_34	ЧТ/ЗП	0x0	Настройка триггера 34
0x0A8	ELC_TRIG_CFG_35	ЧТ/ЗП	0x0	Настройка триггера 35
0x0B0	ELC_TRIG_CFG_36	ЧТ/ЗП	0x0	Настройка триггера 36
0x0B4	ELC_TRIG_CFG_37	ЧТ/ЗП	0x0	Настройка триггера 37
0x0B8	ELC_TRIG_CFG_38	ЧТ/ЗП	0x0	Настройка триггера 38
0x0BC	ELC_TRIG_CFG_39	ЧТ/ЗП	0x0	Настройка триггера 39
0x0C0	ELC_TRIG_CFG_40	ЧТ/ЗП	0x0	Настройка триггера 40
0x0C4	ELC_TRIG_CFG_41	ЧТ/ЗП	0x0	Настройка триггера 41
0x0C8	ELC_TRIG_CFG_42	ЧТ/ЗП	0x0	Настройка триггера 42
0x0CC	ELC_TRIG_CFG_43	ЧТ/ЗП	0x0	Настройка триггера 43
0x0D0	ELC_TRIG_CFG_44	ЧТ/ЗП	0x0	Настройка триггера 44
0x0D4	ELC_TRIG_CFG_45	ЧТ/ЗП	0x0	Настройка триггера 45
0x0D8	ELC_TRIG_CFG_46	ЧТ/ЗП	0x0	Настройка триггера 46
0x0DC	ELC_TRIG_CFG_47	ЧТ/ЗП	0x0	Настройка триггера 47

Адрес	Название	Доступ	Значение после сброса	Описание
0x0E0	ELC_TRIG_CFG_48	ЧТ/ЗП	0x0	Настройка триггера 48
0x0E4	ELC_TRIG_CFG_49	ЧТ/ЗП	0x0	Настройка триггера 49
0x0E8	ELC_TRIG_CFG_50	ЧТ/ЗП	0x0	Настройка триггера 50
0x0EC	ELC_TRIG_CFG_51	ЧТ/ЗП	0x0	Настройка триггера 51
0x0F0	ELC_TRIG_CFG_52	ЧТ/ЗП	0x0	Настройка триггера 52
0x0F4	ELC_TRIG_CFG_53	ЧТ/ЗП	0x0	Настройка триггера 53
0x0F8	ELC_TRIG_CFG_54	ЧТ/ЗП	0x0	Настройка триггера 54
0x0FC	ELC_TRIG_CFG_55	ЧТ/ЗП	0x0	Настройка триггера 55
0x100	ELC_TRIG_CFG_56	ЧТ/ЗП	0x0	Настройка триггера 56
0x104	ELC_TRIG_CFG_57	ЧТ/ЗП	0x0	Настройка триггера 57
0x108	ELC_TRIG_CFG_58	ЧТ/ЗП	0x0	Настройка триггера 58
0x10C	ELC_TRIG_CFG_59	ЧТ/ЗП	0x0	Настройка триггера 59
0x110	ELC_TRIG_CFG_60	ЧТ/ЗП	0x0	Настройка триггера 60
0x114	ELC_TRIG_CFG_61	ЧТ/ЗП	0x0	Настройка триггера 61
0x118	ELC_TRIG_CFG_62	ЧТ/ЗП	0x0	Настройка триггера 62
0x11C	ELC_TRIG_CFG_63	ЧТ/ЗП	0x0	Настройка триггера 63
0x120	ELC_TRIG_CFG_64	ЧТ/ЗП	0x0	Настройка триггера 64
0x124	ELC_TRIG_CFG_65	ЧТ/ЗП	0x0	Настройка триггера 65
0x128	ELC_TRIG_CFG_66	ЧТ/ЗП	0x0	Настройка триггера 66
0x12C	ELC_TRIG_CFG_67	ЧТ/ЗП	0x0	Настройка триггера 67
0x130	ELC_TRIG_CFG_68	ЧТ/ЗП	0x0	Настройка триггера 68

1.3.11.4.4 Описание полей регистров

1.3.11.4.4.1 Регистр ELC_ID

Смещение адреса 0x000

В таблице 1.548 представлено описание полей регистра.

Таблица 1.548 – Описание полей регистра ELC_ID

Поле	Название	Программный доступ	Состояние после сброса	Описание
31:0	ID	ЧТ	0x5f434c45	идентификатор устройства ELC

TrustZone: любой доступ

1.3.11.4.4.2 Регистр ELC_VERSION

Смещение адреса 0x004

В таблице 1.549 представлено описание полей регистра.

Таблица 1.549 – Описание полей регистра ELC_VERSION

Поле	Название	Программный доступ	Состояние после сброса	Описание
31:0	VERSION	ЧТ	0x1	версия устройства ELC

TrustZone: любой доступ

1.3.11.4.4.3 Регистр ELC_COMMAND

Смещение адреса 0x008

В таблице 1.550 представлено описание полей регистра.

Таблица 1.550 – Описание полей регистра ELC_COMMAND

Поле	Название	Программный доступ	Состояние после сброса	Описание
0	SW_RUN	ЗП	0	Запись: 1 - однократный программный запуск всех триггеров, у которых установлен бит SW_EN 0 - не имеет действия
31:1			0	Зарезервировано

TrustZone: при не секьюрной установке SW_RUN запускаются только не секьюрные триггеры. При секьюрной установке SW_RUN запускаются все триггеры.

1.3.11.4.4.4 Регистр ELC_LOCK

Смещение адреса 0x00C

В таблице 1.551 представлено описание полей регистра.

Таблица 1.551 – Описание полей регистра ELC_LOCK

Поле	Название	Программный доступ	Состояние после сброса	Описание
15:0	LOCK	ЧТ/ЗП	0	Запись: 0x10CC - включение блокировки другие значения - не имеет действия Чтение: 0 - запись не заблокирована; 1 - запись заблокирована.
31:16			0	Зарезервировано

TrustZone: если все биты ELC_TRIG_SECURE равны 0, любой доступ. Если хотя бы один равен 1 - только секьюрный доступ.

1.3.11.4.4.5 Регистр ELC_TRIG_SECURE_0

Смещение адреса 0x010

В таблице 1.552 представлено описание полей регистра.

Таблица 1.552 – Описание полей регистра ELC_TRIG_SECURE_0

Поле	Название	Программный доступ	Состояние после сброса	Описание
31:0		ЧТ/ЗП	0	Каждый бит (с 0 по 31) - настройка безопасности триггера (с 0 по 31) 1 в бите N - ограничивает программный доступ к регистрам, связанным с работой соответствующего триггера N только секьюрными запросами; 0 в бите N - снимает ограничения.

TrustZone: только секьюрный доступ на запись и чтение.

1.3.11.4.4.6 Регистр ELC_TRIG_SECURE_1

Смещение адреса 0x014

В таблице 1.553 представлено описание полей регистра.

Таблица 1.553 – Описание полей регистра ELC_TRIG_SECURE_1

Поле	Название	Программный доступ	Состояние после сброса	Описание
31:0		ЧТ/ЗП	0	Каждый бит (с 0 по 31) - настройка безопасности триггера (с 32 по 63) 1 в бите N - ограничивает программный доступ к регистрам, связанным с работой соответствующего триггера N только секьюрными запросами; 0 в бите N - снимает ограничения.

TrustZone: только секьюрный доступ на запись и чтение.

1.3.11.4.4.7 Регистр ELC_TRIG_SECURE_2

Смещение адреса 0x018

В таблице 1.554 представлено описание полей регистра.

Таблица 1.554 – Описание полей регистра ELC_TRIG_SECURE_2

Поле	Название	Программный доступ	Состояние после сброса	Описание
4:0		ЧТ/ЗП	0	Каждый бит (с 0 по 4) - настройка безопасности триггера (с 64 по 68) 1 в бите N - ограничивает программный доступ к регистрам, связанным с работой соответствующего триггера N только секьюрными запросами; 0 в бите N - снимает ограничения.
31:5			0	Зарезервировано

TrustZone: только секьюрный доступ на запись и чтение.

1.3.11.4.4.8 Регистр ELC_TRIG_CFG_0

Смещение адреса 0x020

В таблице 1.555 представлено описание полей регистра.

Таблица 1.555 – Описание полей регистра ELC_TRIG_CFG_0

Поле	Название	Программный доступ	Состояние после сброса	Описание
7:0	MAP	ЧТ/ЗП	0	Номер входного события, подключенного к триггеру 0. Допустимые значение приведены в таблице в таблице 1.545. Установка других значений запрещена. Целевой блок должен игнорировать входящий триггер во время изменения этого поля.
15:8			0	Зарезервировано
16	SW_EN	ЧТ/ЗП	0	Разрешение программного запуска триггера 0 0 - запрещено; 1 - разрешено.
31:17			0	Зарезервировано

TrustZone: доступен, если соответствующая настройка секьюрности ниже или совпадает с параметрами программного запроса.

1.3.11.4.4.9 Регистры ELC_TRIG_CFG_1 .. ELC_TRIG_CFG_68

Смещения адресов 0x020 .. 0x130

Регистры настройки триггеров 1 .. 68.

Поля всех регистров аналогичны полям регистра ELC_TRIG_CFG_0.

1.3.12 Контроллер прямого доступа в память (GPDMA)

1.3.12.1 Характеристики контроллера GPDMA

Контроллеры GPDMA в составе МК Optimal+ выполняют операции прямого доступа к памяти (ПДП), разгружая тем самым вычислительные ресурсы центрального процессора.

Контроллер GPDMA обладает следующими характеристиками:

– настраиваемое число каналов (до 32);

– настраиваемый уровень приоритета для каждого канала;

– поддерживаются различные типы передачи:

  – передача «память-память»;

  – передача «память-устройство;

  – передача "устройство-память";

– поддержка передачи данных различной разрядности;

– каждому каналу доступны первичная и альтернативная структура управляющих данных канала;

– все управляющие данные канала хранятся в памяти системы в формате с прямым порядком байтов;

– поддерживаются только одиночные транзакции на шине AHB;

– разрядность данных для передачи равна разрядности исходных данных;

– можно задать число передач за один цикл от 1 до 1024;

– увеличение адреса передачи может быть больше разрядности данных;

– генерация сигнала ошибки AHB master интерфейса;

– интерфейс программирования: APB slave;

– интерфейс передачи данных: AHB master.

1.3.12.2 Описание функционирования контроллера GPDMA

1.3.12.2.1 Структурная схема контроллера GPDMA

Структурная схема контроллера GPDMA показана на рисунке 1.137.

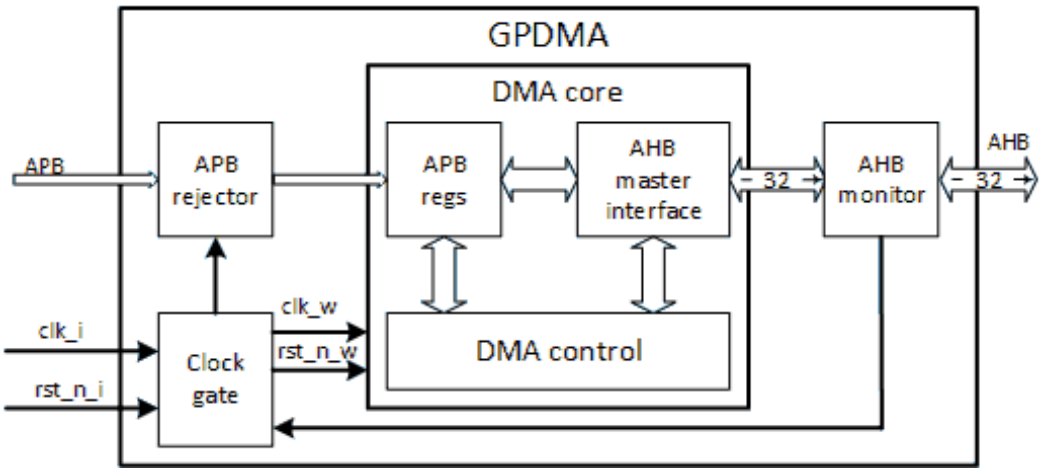


Рисунок 1.137 – Структурная схема контроллера GPDMA

Функциональные блоки, входящие в состав контроллера GPDMA содержит представлены в таблице 1.556:

Таблица 1.556 – Описание функциональных блоков контроллера GPDMA

Блок	**Описание**
DMA core	Верхний уровень блока, выполняющего операции прямого доступа к памяти
APB regs	Блок, реализующий доступ к регистрам управления через APB slave интерфейс
AHB master interface	Блок, реализующий доступ системной шине для передачи данных через AHB master интерфейс
DMA control	Блок управления операциями прямого доступа к памяти
Clock gate	Блок управления тактовым сигналом (подача и снятие)
APB rejector	Блок ответа на APB транзакции, в случае снятия тактового сигнала
AHB monitor	Блок мониторинга незавершённых транзакция на шине AHB

1.3.12.2.2 Описание функционирования контроллера GPDMA

1.3.12.2.2.1 Интерфейс AHB master

1.3.12.2.2.1.1 Типы AHB транзакций

Контроллер GPDMA поддерживает только одиночные транзакции на шине AHB. Отсутствие пакетных транзакций незначительно влияет на производительность системы, так как пакетные транзакции наиболее эффективны в одноуровневых системах AHB, где устройству нужно передавать запрос шине или получать доступ к памяти вне ИС. Контроллер GPDMAC предназначен для многоуровневых систем, включающих в себя память внутри ИС.

1.3.12.2.2.1.2 Разрядность передаваемых данных

Контроллер GPDMA поддерживает передачу данных с разрядностью 8, 16 или 32 битов.

Контроллер поддерживает передачу 32-битовых данных при получении доступа к структуре управляющих данных канала. Следует задать разрядность передаваемых исходных данных, аналогичную разрядности передаваемых адресату данных.

1.3.12.2.2.1.3 Управление защитой

Контроллер GPDMA предусматривает возможность настройки сигналов управления HPROT[3:1] шины AHB. Настройка HPROT позволяет настраивать следующие типы транзакций:

– кэшируемая;

– буферизируемая;

– привилегированная.

В таблице 1.557 указаны коды сигналов HPROT.

Таблица 1.557 – Сигналы защиты

HPROT[3] кэшируемая	HPROT[2] буферизируемая	HPROT[1] привилегированная	HPROT[0] данные/код операции	Описание
-	-	-	1[1)]	Доступ к данным
-	-	0	-	Доступ пользователя
-	-	1	-	Привилегированный доступ
	0	-	-	Небуферизируемая передача
	1	-	-	Буферизируемая передача
0	-	-	-	Некэшируемая передача
1	-	-	-	Кэшируемая передача
[1)] Контроллер GPDMA задает высокий сигнал HPROT[0] для индикации доступа к данным.				

Для каждого цикла ПДП можно настроить исходную передачу и конечную передачу для использования различных параметров управления защитой. Порядок настройки управления защитой доступа см. пункт 1.3.12.2.2.3.1.

Также можно настроить отдельные параметры управления защитой при доступе к структуре управляющих данных канала. Порядок настройки управления защитой доступа см. «Настройка DMA».

1.3.12.2.2.1.4 Инкрементация адреса

Контроллер позволяет настроить значение инкремента адреса, которое будет использоваться при чтении исходных данных или записи конечных данных. Инкремент зависит от размера передаваемого пакета данных. В таблице 1.558 указаны возможные комбинации.

Таблица 1.558 – Значения инкремента адреса

Разрядность пакетных данных (биты)	Значение инкремента адреса
8	байт, полуслово или слово
16	полуслово или слово
32	слово

Минимальное значение инкремента адреса всегда должно быть равно размеру пакета данных. Максимальное значение инкремента адреса, допускаемое контроллером, - одно слово.

Если адрес источника или назначения должен оставаться неизменным, при доступе в порядке FIFO можно настроить использование контроллером постоянного адреса (см. 1.3.12.2.2.3.1).

1.3.12.2.2.2 Блок управления прямым доступом к памяти (DMA control)

1.3.12.2.2.2.1 Арбитраж прямого доступа к памяти

Можно настроить время выполнения контроллером GPDMA арбитража во время ПДП. Таким образом, можно уменьшить задержку обработки канала с более высоким приоритетом.

Контроллер задает четыре бита (R_power, далее *R*), которые настраивают число передач на шине AHB перед повторным арбитражем. Интервал арбитража определяется как 2^R, например, если R=4, то интервал арбитража составляет 2^4, т. е. контроллер выполняет арбитраж через каждые 16 передач ПДП.

В таблице 1.559 указаны интервалы арбитража.

Таблица 1.559 – Интервал арбитража при передаче на шине AHB

R_power	Арбитраж после *x* передач ПДП
b0000	x=1
b0001	x=2
b0010	x=4
b0011	x=8
b0100	x=16
b0101	x=32
b0110	x=64
b0111	x=128

R_power	Арбитраж после *x* передач ПДП
b1000	x=256
b1001	x=512
b1010-b1111	x=1024

Следует проявлять внимательность, чтобы не назначить каналу с низким приоритетом большое значение R_power, в противном случае контроллер GPDMA не будет обрабатывать запросы высокого приоритета до повторного арбитража.

Если $N > 2^R$ и не является целым кратным 2^R, то контроллер GPDMA выполняет последовательности 2^R передач, пока не будет выполнено условие $N < 2^R$. Контроллер GPDMA выполняет оставшиеся N передачи в конце цикла ПДП.

Значение битов R_power сохраняется в структуре управляющих данных канала. Подробную информацию о расположении битов R_power в структуре данных см. пункт 1.3.12.2.2.3.1.

1.3.12.2.2.2.2 Приоритет каналов

Когда контроллер GPDMA выполняет арбитраж, он определяет следующий обрабатываемый канал с использованием следующих данных:

– номер канала;

– уровень приоритета (по умолчанию или высокий), назначаемый каналу.

Можно настроить для каждого канала либо уровень приоритета по умолчанию, либо высокий уровень приоритета, настроив регистр chnl_priority_set см. 1.3.12.3.2.15.

Канал с номером 0 имеет наибольший приоритет. Чем больше номер канала, тем меньше уровень приоритета. В таблице 1.560 указаны уровни приоритета каналов ПДП в порядке убывания.

Таблица 1.560 – Приоритет канала ПДП

Номер канала	Уровень приоритета	Приоритет канала в порядке убывания
0	Высокий	Канал ПДП с наибольшим приоритетом
1	Высокий	-
2	Высокий	-
-	Высокий	-
-	Высокий	-
-	Высокий	-
30	Высокий	-
31	Высокий	-
0	По умолчанию	-

Номер канала	Уровень приоритета	Приоритет канала в порядке убывания
1	По умолчанию	-
2	По умолчанию	-
-	По умолчанию	-
-	По умолчанию	-
-	По умолчанию	-
30	По умолчанию	-
31	По умолчанию	Канал ПДП с наименьшим приоритетом

После завершения передачи ПДП контроллер опрашивает все доступные каналы ПДП. На рисунке 1.138 показан процесс определения следующей передачи ПДП.

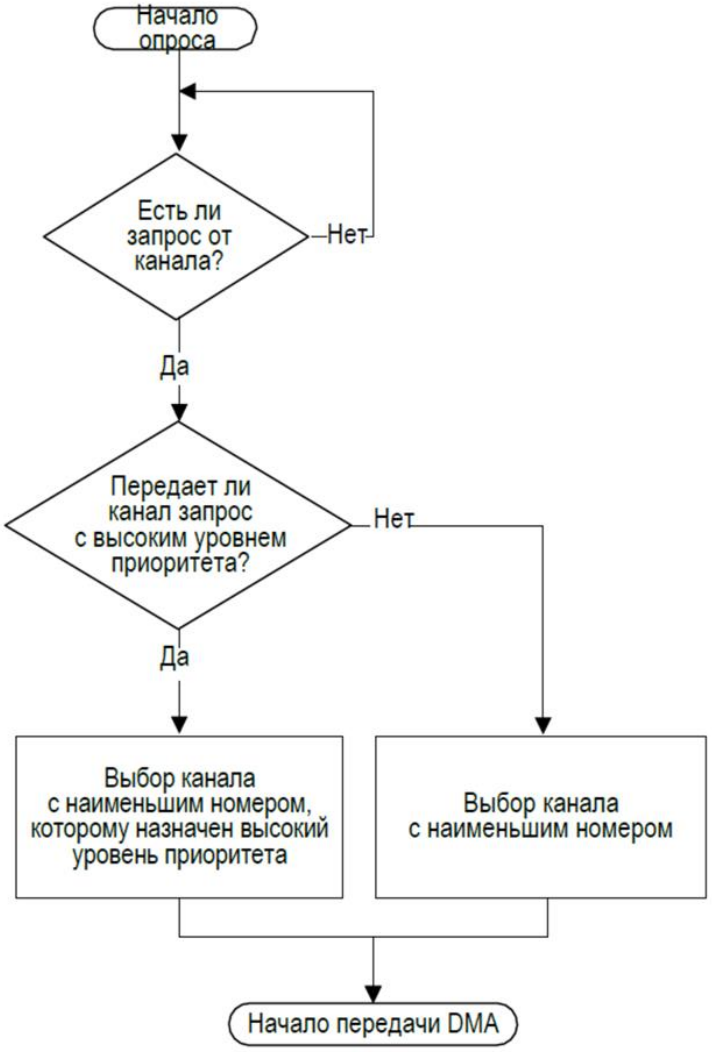


Рисунок 1.138 – Блок-схема опроса

1.3.12.2.2.2.3 Типы цикла ПДП

Биты cycle_ctrl управляют выполнением контроллером GPDMA цикла ПДП. Можно задать биты cycle_ctrl в соответствии с таблицей 1.561.

Таблица 1.561 – Типы цикла ПДП

cycle_ctrl	Описание
b000	Структура управляющих данных канала недействительная
b001	Базовая передача ПДП
b010	Автоматический запрос
b011	Режим «пинг-понг»
b100	Работа с памятью в режиме *scatter-gather* с использованием первичной структуры данных
b101	Работа с памятью в режиме *scatter-gather* с использованием альтернативной структуры данных
b110	Зарезервировано
b111	Зарезервировано

Биты cycle_ctrl находятся поле, определяющем конфигурацию канала (channel_cfg).

Для всех типов циклов контроллер GPDMA выполняет арбитраж после 2^R передач ПДП. При задании канала с низким приоритетом с большим значением 2^R, то все другие каналы не могут выполнять передачу ПДП, пока передача ПДП по каналу с низким приоритетом не будет выполнена. По этой причине следует проявлять внимательность при задании R_power, чтобы избежать значительного увеличения задержки для каналов с высоким приоритетом.

Контроллер GPDMA реализует следующие типы циклов ПДП:

– недействительный;

– базовый;

– автоматический запрос;

– «пинг-понг»;

– работа с памятью в режиме *scatter-gather*.

1.3.12.2.2.2.3.1 Недействительный цикл

После завершения контроллером GPDMA цикла ПДП он задает «недействительный» цикл во избежание повтора одного и того же цикла ПДП.

1.3.12.2.2.2.3.2 Базовый

В этом режиме можно настроить контроллер GPDMA на использование либо первичной, либо альтернативной структуры данных. После активации канала и получения контроллером запроса выполняется цикл ПДП:

1) контроллер GPDMA выполняет 2^R передач. Если число передач остается нулевым, цикл продолжается с этапа 3;

2) контроллер GPDMA выполняет арбитраж:

a) если канал с высоким приоритетом запрашивает обработку, то контроллер GPDMA обрабатывает этот канал;

b) если периферийное устройство или программа передает запрос контроллеру GPDMA, то цикл продолжается с этапа 1;

3) контроллер GPDMA устанавливает сигнал **dma_done[C]** в 1 на один такт **HCLK**, что информирует центральный процессор о завершении цикла ПДП.

1.3.12.2.2.2.3.3 Автоматический запрос

При работе контроллера GPDMA в этом режиме ему необходимо только получить один запрос для активации, чтобы завершить полный цикл ПДП. Выполняется передача большого количества данных без значительного увеличения задержки обработки запросов с более высоким приоритетом или необходимости в нескольких запросах от центрального процессора или периферийного устройства.

Можно настроить контроллер GPDMA на использование либо первичной, либо альтернативной структуры данных. После активации канала и получения контроллером GPDMA запроса для этого канала выполняется цикл ПДП:

1) контроллер GPDMA выполняет 2^R передач для канала C. Если число передач остается нулевым, цикл продолжается с этапа 3;

2) контроллер GPDMA выполняет арбитраж. Если канал C имеет наивысший приоритет, цикл ПДП продолжается с этапа 1;

3) контроллер GPDMA устанавливает сигнал **dma_done[C]** в 1 на один такт **HCLK**, что информирует центральный процессор о завершении цикла ПДП.

1.3.12.2.2.2.3.4 Режим «пинг-понг»

В режиме «пинг-понг» контроллер GPDMA выполняет цикл ПДП с использованием одной из структур данных, затем выполняет цикл ПДП с использованием другой структуры данных. Контроллер GPDMA продолжает выполнять циклы с переключением с первичной на альтернативную структуру и обратно до тех пор, пока не считает «недействительную» структуру данных, или пока центральный процессор не отключит канал.

На рисунке 1.139 показан пример операции ПДП в режиме «пинг-понг».

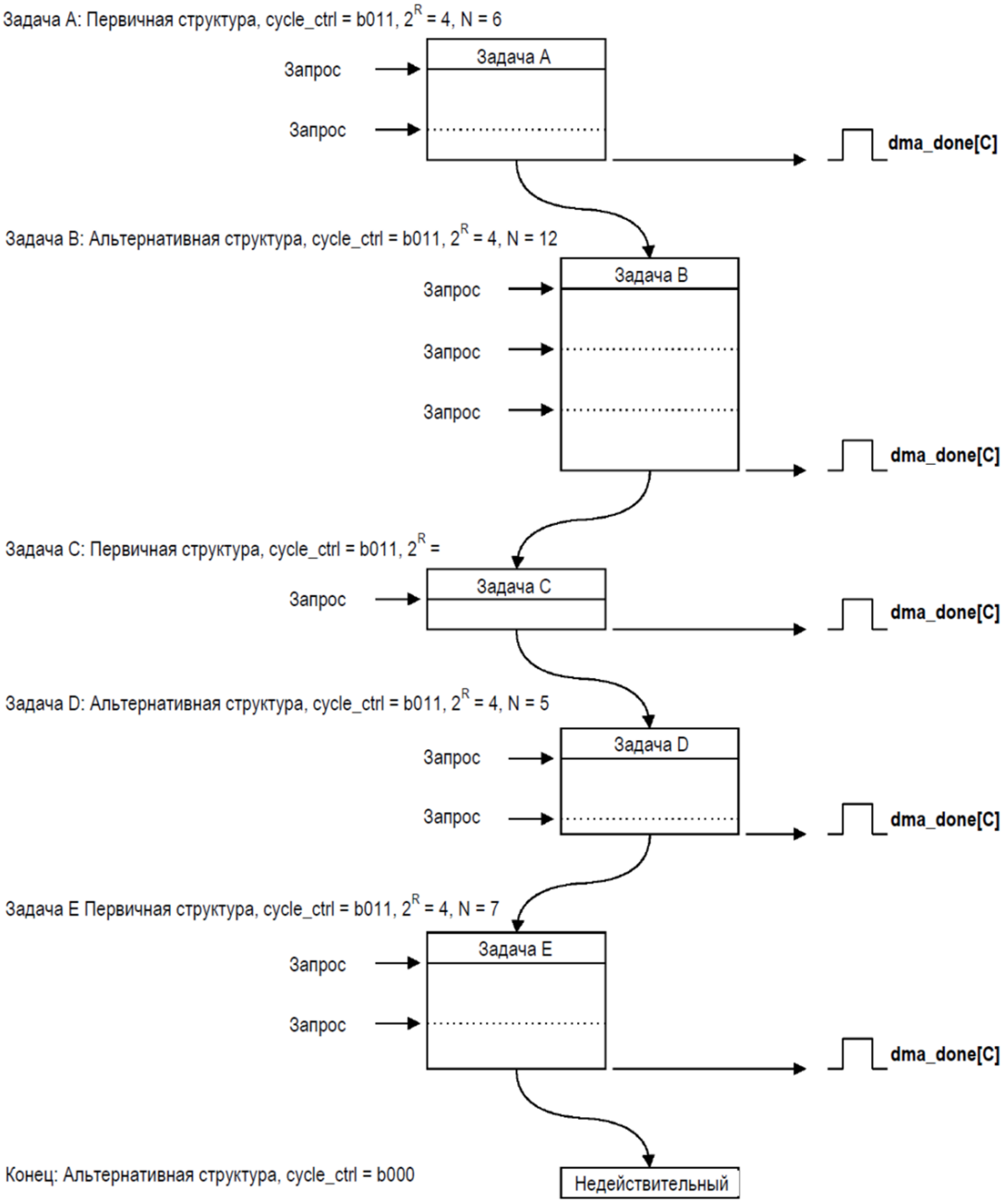


Рисунок 1.139 – Пример цикла в режиме «пинг-понг»

На рисунке 1.139 Задача A:

1) центральный процессор задает первичную структуру данных для задачи A;

2) центральный процессор задает альтернативную структуру данных для задачи B. После этого контроллер GPDMA сразу же переключается на задачу B после завершения задачи A при условии, что канал с большим приоритетом не требует обработки;

3) контроллер GPDMA получает запрос и выполняет четыре передачи ПДП;

4) контроллер GPDMA выполняет арбитраж. После получения контроллером GPDMA запроса для этого канала цикл продолжается, если канал имеет наивысший приоритет;

5) контроллер GPDMA выполняет оставшиеся две ПДП передачи;

6) контроллер GPDMA устанавливает сигнал **dma_done[C]** в 1 на один такт **HCLK** и переходит к арбитражу.

После завершения задачи A центральный процессор может задать первичную структуру данных для задачи C. После этого контроллер GPDMA сразу же переключается на задачу C после завершения задачи B при условии, что канал с большим приоритетом не требует обработки.

После получения контроллером GPDMA нового запроса для канала, имеющего наивысший приоритет, начинается выполнение задачи B:

1) контроллер GPDMA выполняет четыре передачи ПДП;

2) контроллер GPDMA выполняет арбитраж. После получения контроллером GPDMA запроса для этого канала цикл продолжается, если канал имеет наивысший приоритет;

3) контроллер GPDMA выполняет четыре передачи ПДП;

4) контроллер GPDMA выполняет арбитраж. После получения контроллером GPDMA запроса для этого канала цикл продолжается, если канал имеет наивысший приоритет;

5) контроллер GPDMA выполняет оставшиеся четыре ПДП передачи;

6) контроллер GPDMA устанавливает сигнал **dma_done[C]** в 1 на один такт **HCLK** и переходит к арбитражу.

После завершения задачи B центральный процессор может задать альтернативную структуру данных для задачи D.

После получения контроллером GPDMA нового запроса для канала, имеющего наивысший приоритет, начинается выполнение задачи C:

1) контроллер GPDMA выполняет две передачи ПДП;

2) контроллер GPDMA устанавливает сигнал **dma_done[C]** в 1 на один такт **HCLK** и переходит к арбитражу.

После завершения задачи C центральный процессор может задать первичную структуру данных для задачи E.

После получения контроллером GPDMA нового запроса для канала, имеющего наивысший приоритет, начинается выполнение задачи D:

1) контроллер GPDMA выполняет четыре передачи ПДП;

2) контроллер GPDMA выполняет арбитраж. После получения контроллером GPDMA запроса для этого канала цикл продолжается, если канал имеет наивысший приоритет;

3) контроллер GPDMA выполняет оставшуюся передачу ПДП;

4) контроллер GPDMA устанавливает сигнал **dma_done[C]** в 1 на один такт **HCLK** и переходит к арбитражу.

После получения контроллером GPDMA нового запроса для канала, имеющего наивысший приоритет, начинается выполнение задачи E:

1) контроллер GPDMA выполняет четыре передачи ПДП;

2) контроллер GPDMA выполняет арбитраж. После получения контроллером GPDMA запроса для этого канала цикл продолжается, если канал имеет наивысший приоритет;

3) контроллер GPDMA выполняет оставшиеся три ПДП передачи;

4) контроллер GPDMA устанавливает сигнал **dma_done[C]** в 1 на один такт **HCLK** и переходит к арбитражу.

После получения контроллером GPDMA нового запроса для канала, имеющего наивысший приоритет, начинается выполнение новой задачи. Тем не менее, так как центральный процессор не установил альтернативную структуру данных, и после выполнения задачи D контроллер GPDMA задает биты cycle_ctrl на b000, то выполняется операция ПДП в режиме «пинг-понг».

Также можно остановить цикл ПДП в режиме «пинг-понг», как показано на рисунке 2-12, страница 2-24, если установить задачу E в качестве базового цикла ПДП, задав cycle_ctrl на 3’b001.

1.3.12.2.2.2.3.5 Работа с памятью в режиме *scatter-gather*

При работе с памятью в режиме *scatter-gather* контроллер GPDMA получает первый запрос и выполняет четыре передачи ПДП с использованием первичной структуры данных. После завершения передачи контроллер GPDMA начинает цикл ПДП с использованием альтернативной структуры данных. После завершения этого цикла контроллер GPDMA выполняет четыре передачи ПДП с использованием первичной структуры данных. Контроллер GPDMA продолжает переключать первичную и альтернативную структуры данных, пока:

– центральный процессор задает альтернативную структуру данных для базового цикла;

– он читает недействительную структуру данных.

После завершения контроллером GPDMA N первичных передач он делает первичную структуру данных недействительной, устанавливая cycle_ctrl на b000.

Контроллер GPDMA только передает **dma_done[]**, когда операция в режиме *scatter-gather* будет завершена базовым циклом.

В режиме *scatter-gather* контроллер GPDMA использует первичную структуру данных для настройки альтернативной структуры данных. В таблице 1.562 перечислены поля channel_cfg для первичной структуры данных, в которых следует задать постоянные значения, и которые могут быть определены пользователем.

Таблица 1.562 – Channel_cfg для первичной структуры данных при работе с памятью в режиме _scatter-gather

Бит	**Поле**	**Значение**	**Описание**
Поля с постоянными значениями			
[31:30]	dst_inc	b10	В качестве значения увеличения адреса контроллер GPDMA использует слово
[29:28]	dst_size	b10	Контроллер GPDMA выполняет передачи слов
[27:26]	src_inc	b10	В качестве значения увеличения адреса контроллер GPDMA использует слово
[25:24]	src_size	b10	Контроллер GPDMA выполняет передачи слов
[17:14]	R_power	b0010	Контроллер GPDMA выполняет четыре передачи ПДП
[3]	next_useburst	0	В случае цикла работы с памятью в режиме *scatter-gather* этот бит должен быть нулевым
[2:0]	cycle_ctrl	b100	Настройка контроллера GPDMA для выполнения цикла ПДП с памятью в режиме *scatter-gather*
Определяемые пользователем значения			
[23:21]	dst_prot_ctrl	-	Настройка состояния **HPROT,** когда контроллер GPDMA записывает конечные данные
[20:18]	src_prot_ctrl	-	Настройка состояния **HPROT,** когда контроллер GPDMA читает исходные данные
[13:4]	n_minus_1	N[1)]	Настройка контроллера GPDMA на выполнение *N* передач ПДП, где *N* - кратное четырем

Бит	Поле	Значение	Описание
[1)] Так как поле R_power установлено в значение 4, следует задать N равное кратному 4. Значение, заданное N/4, - число раз настройки альтернативной структуры данных.			

Подробную информацию см. пункт 1.3.12.2.2.3.1.

На рисунке 1.140 приведен пример работы с памятью в режиме *scatter-gather*.

Инициализация:

1) настройка первичной структуры для включения копирования операций A, B, C и D: cycle_ctrl = b100, 2^R = 4, N = 16;

2) запись первичных исходных данных в память с использованием структуры, показанной в таблице 1.563.

Таблица 1.563 – Запись первичных исходных данных в память

	src_data_end_ptr	dst_data_end_ptr	channel_cfg	Не используется
Данные для задачи A	0x0A000000	0x0AE00000	cycle_ctrl = b101, 2^R = 4, N = 3	0xXXXXXXXX
Данные для задачи B	0x0B000000	0x0BE00000	cycle_ctrl = b101, 2^R = 2, N = 8	0xXXXXXXXX
Данные для задачи C	0x0C000000	0x0CE00000	cycle_ctrl = b101, 2^R = 8, N = 5	0xXXXXXXXX
Данные для задачи D	0x0D000000	0x0DE00000	cycle_ctrl = b001, 2^R = 4, N = 4	0xXXXXXXXX

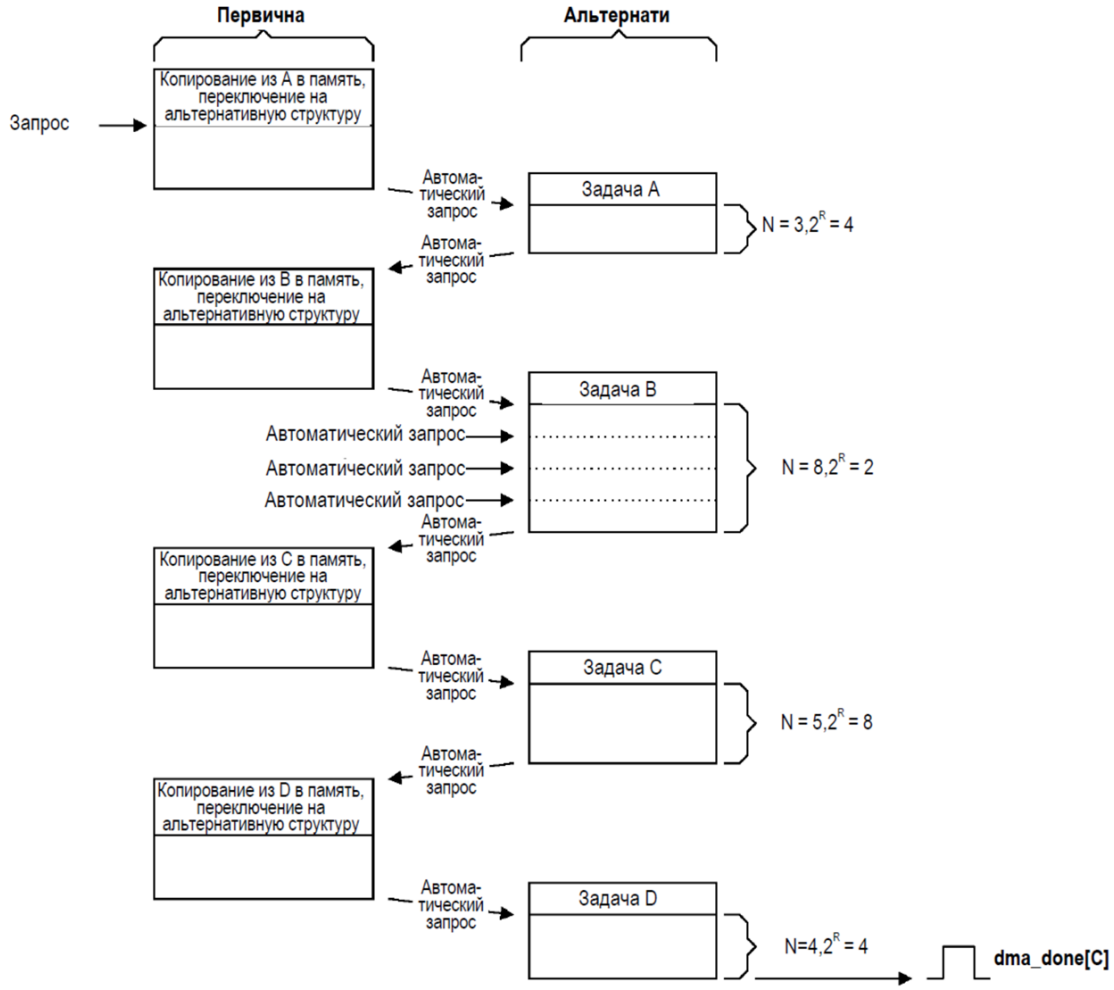


Рисунок 1.140 – Пример работы с памятью в режиме _scatter-gather

На рисунке 1.140 инициализация:

1) центральный процессор настраивает первичную структуру данных для работы с памятью в режиме *scatter-gather*, устанавливая cycle_ctrl на b100. Так как структура данных одного канала состоит из четырех слов, следует установить 2^R на 4. В этом примере выполняются четыре задачи, поэтому N устанавливается на 16;

2) центральный процессор записывает структуру данных для задач A, B, C и D в области памяти, указанные первичным src_data_end_ptr;

3) центральный процессор активирует канал.

Работа с памятью в режиме *scatter-gather* начинается, когда контроллер GPDMA получает запрос от центрального процессора. Операция выполняется следующим образом.

Первичная структура, копия A:

1) после получения запроса контроллер GPDMA выполняет четыре передачи ПДП. Эти передачи записывают альтернативную структуру данных для задачи A;

2) контроллер GPDMA генерирует автоматический запрос для канала и выполняет арбитраж;

3) контроллер GPDMA выполняет задачу A. После завершения задачи он генерирует автоматический запрос для канала и выполняет арбитраж.

Первичная структура, копия B:

1) контроллер GPDMA выполняет четыре передачи ПДП. Эти передачи записывают альтернативную структуру данных для задачи B;

2) контроллер GPDMA генерирует автоматический запрос для канала и выполняет арбитраж;

3) контроллер GPDMA выполняет задачу B. После завершения задачи он генерирует автоматический запрос для канала и выполняет арбитраж;

Первичная структура, копия C:

1) Контроллер GPDMA выполняет четыре передачи ПДП. Эти передачи записывают альтернативную структуру данных для задачи C;

2) контроллер GPDMA генерирует автоматический запрос для канала и выполняет арбитраж;

3) контроллер GPDMA выполняет задачу C. После завершения задачи он генерирует автоматический запрос для канала и выполняет арбитраж.

Первичная структура, копия D:

1) контроллер GPDMA выполняет четыре передачи ПДП. Эти передачи записывают альтернативную структуру данных для задачи D;

2) контроллер GPDMA устанавливает биты cycle_ctrl первичной структуры данных на b000, чтобы указать, что эта структура данных недействительна;

3) контроллер GPDMA генерирует автоматический запрос для канала и выполняет арбитраж;

4) контроллер GPDMA выполняет задачу D в базовом цикле;

5) контроллер GPDMA устанавливает сигнал **dma_done[C]** в 1 на один такт **HCLK** и переходит к арбитражу.

1.3.12.2.2.3 Структура управляющих данных канала

Требования к памяти для структуры управляющих данных канала:

– содержать непрерывную область, к которой есть доступ у контроллера GPDMA и центраьлного процессора

– иметь базовый адрес, являющийся целым кратным общего размера структуры управляющих данных канала.

На рисунке 1.141 показана память, которая нужна контроллеру GPDMA для структуры управляющих данных каналов при использовании всех 32 каналов и дополнительной альтернативной структуры данных.

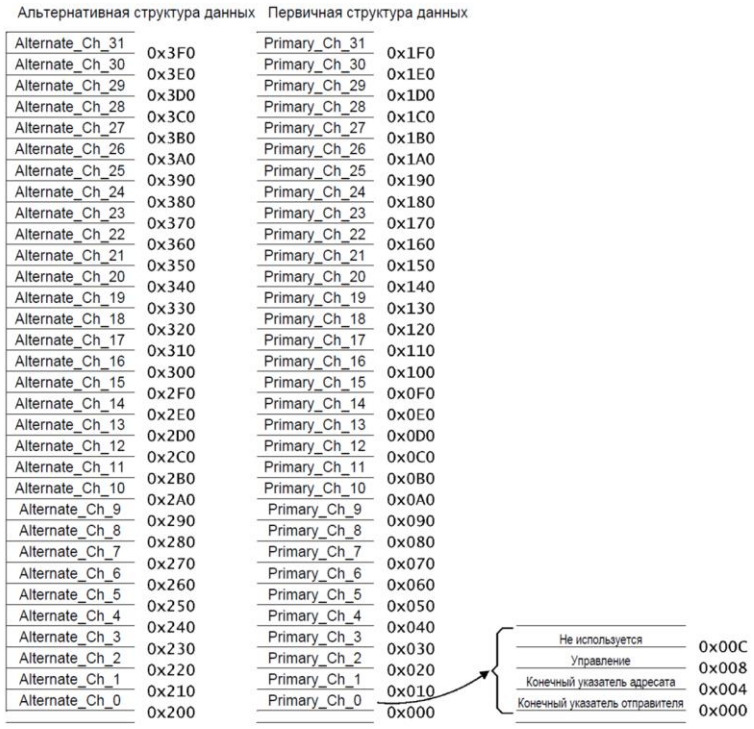


Рисунок 1.141 – Карта памяти на 32 канала, включая альтернативную структуру данных

Пример структуры на рисунке 1.141, занимающей 1 Кб памяти системы. В этом примере контроллер GPDMA использует 10 младших битов адреса для получения доступа ко всем элементам структуры, следовательно, базовый адрес должен быть 0xXXXXX000, 0xXXXXX400, 0xXXXXX800 или 0xXXXXXC00.

Можно настроить базовый адрес для первичной структуры данных, прописав соответствующее значение в регистре ctrl_base_ptr (см. 1.3.12.3.2.3).

Требуемый объем памяти системы зависит от:

– числа каналов ПДП, настраиваемых в контроллере GPDMA;

– настройки канала ПДП на использование альтернативной структуры данных (см. 1.3.12.3.2.13).

В таблице 1.564 указаны биты адреса, которые контроллер GPDMA использует для получения доступа к элементам структуры управляющих данных канала, в зависимости от числа каналов контроллера GPDMA.

Таблица 1.564 – Настройка битов адреса для структуры управляющих данных канала

Число используемых каналов ПДП	**Биты адреса**						
	[9]	**[8]**	**[7]**	**[6]**	**[5]**	**[4]**	**[3:0]**
1						A	0X0, 0x4 или 0x8
2					A	C[0]	
	3-4				A	C[1]	
C[0]		5-8			A	C[2]	
C[1]	C[0]		9-16		A	C[3]	
C[2]	C[1]	C[0]		17-32	A	C[4]	

Где:

– **A** - выбор одной из структуры управляющих данных канала:

– **A = 0** - выбор первичной структуры данных;

– **A = 1** - выбор альтернативной структуры данных;

– **C[*x*:0]** - выбор канала ПДП;

– **Address[3:0]** - выбор одного из элементов управления:

– 0x0 Выбор конечного указателя исходных данных;

– 0x4 Выбор конечного указателя конечных данных;

– 0x8 Выбор конфигурации управляющих данных;

– 0xC Контроллер GPDMA не имеет доступа к этой ячейке адреса. При необходимости можно настроить центральный процессор на использование этой области в качестве памяти системы.

Не нужно рассчитывать базовый адрес альтернативной структуры данных, так как регистр alt_ctrl_base_ptr содержит эту информацию (см. 1.3.12.3.2.4).

На рисунке 1.142 показан пример, где контроллер GPDMA использует три канала ПДП и альтернативную структуру данных.



Рисунок 1.142 – Карта памяти на три канала ПДП, включая альтернативную структуру данных

Пример структуры на рисунке 1.142 занимает 128 байт памяти системы. В этом примере контроллер GPDMA использует шесть младших битов адреса для получения доступа ко всем элементам структуры, следовательно, базовый адрес должен быть 0xXXXXXX00 или 0xXXXXXX80.

В таблице 1.565 указаны допустимые значения базового адреса, которые можно назначить для первичной структуры данных в зависимости от числа каналов контроллера GPDMA.

Таблица 1.565 – Допустимые базовые адреса

Число каналов ПДП	**Допустимые базовые адреса[1)] для первичной структуры данных**
1	0xXXXXXX00, 0xXXXXXX20, 0xXXXXXX40, 0xXXXXXX60, 0xXXXXXX80, 0xXXXXXXA0, 0xXXXXXXC0, 0xXXXXXXE0
2	0xXXXXXX00, 0xXXXXXX40, 0xXXXXXX80, 0xXXXXXXC0
3-4	0xXXXXXX00, 0xXXXXXX80
5-8	0xXXXXX000, 0xXXXXX100, 0xXXXXX200, 0xXXXXX300, 0xXXXXX400, 0xXXXXX500, 0xXXXXX600, 0xXXXXX700, 0xXXXXX800, 0xXXXXX900, 0xXXXXXA00, 0xXXXXXB00, 0xXXXXXC00, 0xXXXXXD00, 0xXXXXXE00, 0xXXXXXF00
9-16	0xXXXXX000, 0xXXXXX200, 0xXXXXX400, 0xXXXXX600, 0xXXXXX800, 0xXXXXXA00, 0xXXXXXC00, 0xXXXXXE00

Число каналов ПДП	Допустимые базовые адреса[1)] для первичной структуры данных
17-32	0xXXXXX000, 0xXXXXX400, 0xXXXXX800, 0xXXXXXC00
[1)] X является шестнадцатеричным значением.	

Структуры управляющих данных канала включает следующие поля:

– указатель на конечный адрес области данных для передачи (src_data_end_ptr);

– указатель на конечный адрес области данных для приёма (dst_data_end_ptr);

– конфигурация управляющих данных (channel_cfg).

1.3.12.2.2.3.1 Конфигурация управляющих данных

Для каждой передачи ПДП поле channel_cfg содержит управляющие данные для контроллера GPDMA. На рисунке 1.143 указаны назначения битов для этой области памяти.

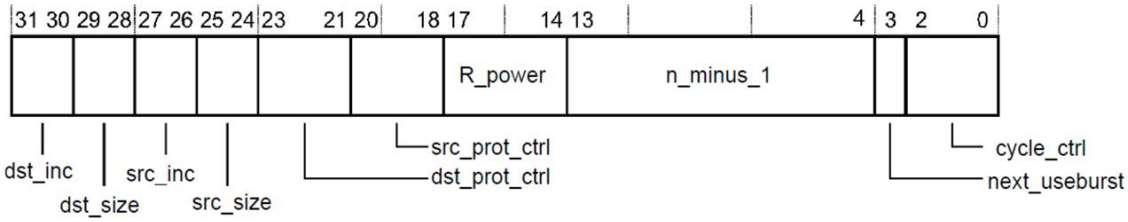


Рисунок 1.143 – Назначения битов channel_cfg

В таблице 1.566 указаны биты channel_cfg.

Таблица 1.566 – Назначения битов channel_cfg

Бит	Название	Описание
[31:30]	dst_inc	Увеличение адреса назначения Увеличение адреса зависит от разрядности исходных данных: **Разрядность исходных данных = байт** b00 = байт. b01 = полуслово. b10 = слово. b11 = нет увеличения. Значение адреса, указанное в ячейке памяти dst_data_end_ptr, не изменится. **Разрядность исходных данных = полуслово** b00 = зарезервировано. b01 = полуслово. b10 = слово. b11 = нет увеличения. Значение адреса, указанное в ячейке памяти dst_data_end_ptr, не изменится. **Разрядность исходных данных = слово** b00 = зарезервировано. b01 = зарезервировано. b10 = слово. b11 = нет увеличения. Значение адреса, указанное в ячейке памяти dst_data_end_ptr, не изменится.
[29:28]	dst_size	Разрядность конечных данных. dst_size должен содержать то же значение, что и src_size.
[27:26]	src_inc	Устанавливает биты для управления увеличением адреса источника. Увеличение адреса зависит от разрядности исходных данных: **Разрядность исходных данных = байт** b00 = байт. b01 = полуслово. b10 = слово. b11 = нет увеличения. Значение адреса, указанное в ячейке памяти src_data_end_ptr, не изменится. **Разрядность исходных данных = полуслово** b00 = зарезервировано. b01 = полуслово. b10 = слово. b11 = нет увеличения. Значение адреса, указанное в ячейке памяти src_data_end_ptr, не изменится. **Разрядность исходных данных = слово** b00 = зарезервировано. b01 = зарезервировано. b10 = слово. b11 = нет увеличения. Значение адреса, указанное в ячейке памяти src_data_end_ptr, не изменится.
[25:24]	src_size	Задание битов для соответствия разрядности исходных данных: b00 = байт b01 = полуслово b10 = слово b11 = зарезервировано.

Бит	Название	Описание
[23:21]	dst_prot_ctrl	Задание битов для управления состоянием **HPROT[3:1],** когда контроллер GPDMA записывает конечные данные. **Бит [23]** Управляет состоянием **HPROT[3]**: 0 - некэшируемый доступ. 1 - кэшируемый доступ. **Бит [22]** Управляет состоянием **HPROT[2]**: 0 - небуферизованный доступ. 1 - буферизованный доступ. **Бит [21]** Управляет состоянием **HPROT[1]**: 0 - непривилегированный доступ. 1 - привилегированный доступ.
[20:18]	src_prot_ctrl	Задание битов для управления состоянием **HPROT[3:1],** когда контроллер GPDMA читает исходные данные. **Бит [20]** Управляет состоянием **HPROT[3]**: 0 - некэшируемый доступ. 1 - кэшируемый доступ. **Бит [19]** Управляет состоянием **HPROT[2]**: 0 - небуферизованный доступ. 1 - буферизованный доступ. **Бит [18]** Управляет состоянием **HPROT[1]**: 0 - непривилегированный доступ. 1 - привилегированный доступ.
[17:14]	R_power	Задание этих битов для настройки числа передач ПДП до проведения контроллером GPDMA повторного арбитража. Возможный интервал арбитража: **b0000** Арбитраж через каждую передачу ПДП. **b0001** Арбитраж через каждые 2 передачи ПДП. **b0010** Арбитраж через каждые 4 передачи ПДП. **b0011** Арбитраж через каждые 8 передач ПДП. **b0100** Арбитраж через каждые 16 передач ПДП. **b0101** Арбитраж через каждые 32 передачи ПДП. **b0110** Арбитраж через каждые 64 передачи ПДП. **b0111** Арбитраж через каждые 128 передач ПДП. **b1000** Арбитраж через каждые 256 передач ПДП. **b1001** Арбитраж через каждые 512 передач ПДП. **b1010-b1111** Арбитраж через каждые 1024 передачи ПДП. Это означает, что арбитраж не выполняется во время передачи ПДП, так как максимальный размер передачи составляет 1024.

Бит	Назва-ние	Описание
[13:4]	n_minus_1	Перед началом цикла ПДП эти биты задают общее число передач ПДП в цикле ПДП. Следует задать эти биты с учетом размера цикла ПДП. 10-битовое значение означает число передач ПДП минус единица. Возможные значения: b000000000 = 1 передача ПДП b000000001 = 2 передачи ПДП b000000010 = 3 передачи ПДП b000000011 = 4 передачи ПДП b000000100 = 5 передач ПДП . . . b111111111 = 1024 передачи ПДП. Контроллер GPDMA обновляет это поле сразу при переключении на арбитраж. Это позволяет контроллеру GPDMA хранить число ожидающих передач ПДП, необходимое для выполнения цикла ПДП.
[3]	next_useburst	Управляет, если бит chnl_useburst_set [C] установлен на 1, когда контроллер GPDMA выполняет работу с периферией в режиме *scatter-gather* и завершает цикл ПДП с использованием альтернативной структуры данных. Сразу перед завершением цикла ПДП с использованием альтернативной структуры данных контроллер GPDMA устанавливает значение бита chnl_useburst_set [C] на 0, если число оставшихся передач меньше 2^R. Бит next_useburst устанавливается на управление, если контроллер GPDMA выполняет дополнительное изменение бита chnl_useburst_set [C]. В случае цикла ПДП работы с периферией в режиме *scatter-gather* после завершения цикла ПДП с альтернативной структурой данных либо: 0 = контроллер GPDMA не изменяет значение бита chnl_useburst_set [C]. Если бит chnl_useburst_set [C] установлен на 0, то для всех оставшихся циклов ПДП работы с периферией в режиме *scatter-gather* контролер отвечает на запросы на **dma_req[]** и **dma_sreq[]** при выполнении цикла ПДП с использованием альтернативной структуры данных. 1 = контроллер GPDMA устанавливает бит chnl_useburst_set [C] на 1. Следовательно, то для всех оставшихся циклов ПДП работы с периферией в режиме *scatter-gather* контролер отвечает на запросы на **dma_req[]** только при выполнении цикла ПДП с использованием альтернативной структуры данных.

Бит	Название	Описание
[2:0]	cycle_ctrl	Режим работы цикла ПДП. Режимы: **b000** Остановка. Означает, что структура данных является недействительной. **b001** Базовый. Контроллер GPDMA должен получить новый запрос перед переходом к арбитражу, чтобы завершить цикл ПДП. **b010** Автоматический запрос. Контроллер GPDMA автоматически вставляет запрос для соответствующего канала во время арбитража. Это означает, что исходного запроса достаточно для завершения цикла ПДП. **b011** «Пинг-понг». Контроллер GPDMA выполняет цикл ПДП с использованием одной из структур данных. После завершения цикла ПДП контроллер GPDMA выполняет цикл ПДП с использованием другой структуры данных. После завершения цикла ПДП и при условии, что центральный процессор изменил исходную структуру данных, контроллер GPDMA выполняет цикл ПДП с исходной структурой данных. Контроллер GPDMA продолжает выполнять циклы ПДП, пока он не прочтет недействительную структуру данных, или пока центральный процессор не изменит биты cycle_ctrl на b001 или b010. (см. 1.3.12.2.2.2.3.4). **b100** Работа с памятью в режиме *scatter-gather*. (см. 1.3.12.2.2.2.3.5). Если контроллер GPDMA работает с памятью в режиме *scatter-gather*, следует использовать это значение в первичной структуре данных. **b101** Работа с памятью в режиме *scatter-gather*. (см. 1.3.12.2.2.2.3.5). Если контроллер GPDMA работает с памятью в режиме *scatter-gather*, следует использовать это значение в альтернативной структуре данных. **b110** - резерв. **b111** - резерв.

В начале цикла ПДП или 2^R передаче ПДП контроллер GPDMA вызывает channel_cfg из памяти системы. После выполнения 2^R или N передач он сохраняет обновленный channel_cfg в памяти системы.

Контроллер GPDMA не поддерживает значение dst_size, отличное от значения src_size. Если контроллер GPDMA обнаруживает несоответствие этих значений, он использует значение src_size в качестве исходного и конечного, а затем при обновлении поля n_minus_1 он также задает поле dst_size, аналогичное полю src_size.

После завершения контроллером GPDMA N передач он устанавливает поле cycle_ctrl на b000, чтобы указать, что данные channel_cfg являются недействительными. Таким образом, обеспечивается защита от повторения одной и той же передачи ПДП.

1.3.12.2.2.3.2 Расчет адреса

Для расчета источника адреса для передачи ПДП контроллер GPDMA выполняет сдвиг влево значение n_minus_1 на величину, указанную в src_inc, а затем вычитает полученное значение из src_data_end_ptr. Аналогично, для расчета адреса назначения для передачи ПДП контроллер GPDMA выполняет сдвиг влево значение n_minus_1 на величину, указанную в dst_inc, а затем вычитает полученное значение из dst_data_end_ptr.

Адрес источника и адрес назначения можно рассчитать по указанным ниже формулам в зависимости от значений src_inc и dst_inc:

– src_inc=b00 и dst_inc=b00:

– адрес источника = src_data_end_ptr - n_minus_1;

– адрес назначения = dst_data_end_ptr - n_minus_1;

– src_inc=b01 и dst_inc=b01:

– адрес источника = src_data_end_ptr - (n_minus_1 << 1);

– адрес назначения = dst_data_end_ptr - (n_minus_1 << 1);

– src_inc=b10 и dst_inc=b10:

– адрес источника = src_data_end_ptr - (n_minus_1 << 2);

– адрес назначения = dst_data_end_ptr - (n_minus_1 << 2);

– src_inc=b11 и dst_inc=b11:

– адрес источника = src_data_end_ptr;

– адрес назначения = dst_data_end_ptr.

В таблице 1.567 указаны адреса назначения для цикла ПДП из шести слов.

Таблица 1.567 – Цикл ПДП из шести слов с приращением в слово

Исходные значения channel_cfg перед циклом ПДП				
src_size=b10, dst_inc=b10, n_minus_1=b101, cycle_ctrl= 1				
Передачи ПДП	**Конечный указатель**	**Счет**	**Разница[1)]**	**Адрес**
	0x2AC	5	0x14	0x298
	0x2AC	4	0x10	0x29C
	0x2AC	3	0xC	0x2A0
	0x2AC	2	0x8	0x2A4
	0x2AC	1	0x4	0x2A8
	0x2AC	0	0x0	0x2AC
Конечные значения channel_cfg после цикла ПДП				
src_size=b10, dst_inc=b10, n_minus_1=0, cycle_ctrl=0				
[1)] Это значение является результатом подсчета со сдвигом влево на значение dst_inc.				

В таблице 1.568 указаны адреса назначения для 12-байтовой передачи ПДП с приращением в полуслово.

Таблица 1.568 – Цикл ПДП из 12 байтов с приращением в полуслово

Исходные значения channel_cfg перед циклом ПДП
src_size=b00, dst_inc=b01, n_minus_1=b1011, cycle_ctrl=1, R_power=b11

Исходные значения channel_cfg перед циклом ПДП				
Передачи ПДП	**Конечный указатель**	**Счет**	**Разница[1)]**	**Адрес**
	0x5E7	11	0x16	0x5D1
	0x5E7	10	0x14	0x5D3
	0x5E7	9	0x12	0x5D5
	0x5E7	8	0x10	0x5D7
	0x5E7	7	0xE	0x5D9
	0x5E7	6	0xC	0x5DB
	0x5E7	5	0xA	0x5DD
	0x5E7	4	0x8	0x5DF
Значения channel_cfg после 2^R передач ПДП				
src_size=b00, dst_inc=b01, n_minus_1=b011, cycle_ctrl=1, R_power=b11				
Передачи ПДП	**Конечный указатель**	**Счет**	**Разница**	**Адрес**
	0x5E7	3	0x6	0x5E1
	0x5E7	2	0x4	0x5E3
	0x5E7	1	0x2	0x5E5
	0x5E7	0	0x0	0x5E7
Конечные значения channel_cfg после цикла ПДП				
src_size=b00, dst_inc=b01, n_minus_1=0, cycle_ctrl=0[2)], R_power=b11				

[1)] Это значение является результатом подсчета со сдвигом влево на значение dst_inc.

[2)] После завершения контроллером GPDMA цикла ПДП он аннулирует ячейку памяти channel_cfg путем очистки поля cycle_ctrl.

1.3.12.3 Описание регистров контроллера GPDMA

1.3.12.3.1 Карта регистров контроллера GPDMA

В таблице 1.569 указаны регистры контроллера GPDMA в порядке смещения базового адреса.

Таблица 1.569 – Описание регистров

Название	Смещение базового адреса	Доступ	Значение после сброса	Описание
dma_status	0x000	ЧТ	0x-0nn0000[1)]	«Статус DMA»
dma_cfg	0x004	ЗП	-	«Конфигурация DMA»
ctrl_base_ptr	0x008	ЧТ/ЗП	0x00000000	«Указатель базового адреса управляющих данных канала»
alt_ctrl_base_ptr	0x00C	ЧТ	0x000000nn[2)]	«Указатель базового адреса альтернативной структуры управляющих данных канала»,
dma_waitonreq_status	0x010	ЧТ	0x00000000	«Статус ожидания каналом запроса»
chnl_sw_request	0x014	ЗП	-	«Программный запрос канала»

Название	Смещение базового адреса	Доступ	Значение после сброса	Описание
chnl_useburst_set	0x018	ЧТ/ЗП	0x00000000	«Установка useburst канала»
chnl_useburst_clr	0x01C	ЗП	-	«Сброс useburst канала»
chnl_req_mask_set	0x020	ЧТ/ЗП	0x00000000	«Настройка маски запроса канала»
chnl_req_mask_clr	0x024	ЗП	-	«Сброс маски запроса канала»
chnl_enable_set	0x028	ЧТ/ЗП	0x00000000	«Включение канала»
chnl_enable_clr	0x02C	ЗП	-	«Сброс включения канала»
chnl_pri_alt_set	0x030	ЧТ/ЗП	0x00000000	«Настройка первичной и альтернативной структуры канала»
chnl_pri_alt_clr	0x034	ЗП	-	«Сброс первичной и альтернативной структуры канала»
chnl_priority_set	0x038	ЧТ/ЗП	0x00000000	«Настройка приоритета каналов»
chnl_priority_clr	0x03C	ЗП	-	«Сброс приоритета каналов»,
-	0x040-0x48	-	-	Зарезервировано
err_clr	0x04C	ЧТ/ЗП	0x00000000	«Сброс ошибки шины»
-	0x050-0xFCC	-	-	Зарезервировано

[1)] Значение сброса зависит от числа каналов ПДП, настроенных для контроллера GPDMA, и если контроллер GPDMA предусматривает логику проверки интеграции.

[2)] Значение сброса зависит от числа каналов ПДП, настроенных для контроллера GPDMA.

1.3.12.3.2 Описание полей регистров контроллера GPDMA

1.3.12.3.2.1 dma_status (0x000)

Регистр только для чтения dma_status возвращает состояние контроллера GPDMA. Чтение этого регистра невозможно, когда контроллер GPDMA находится в состоянии сброса. На рисунке 1.144 указаны поля регистра.

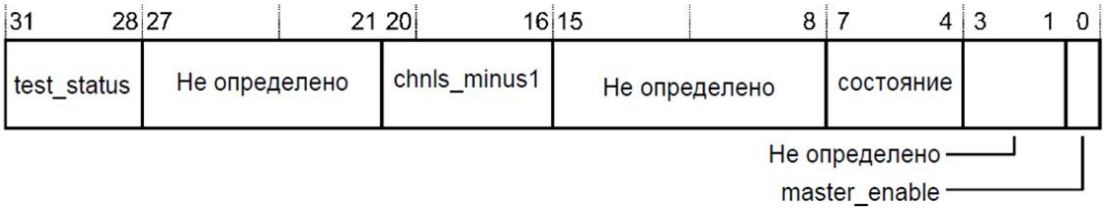


Рисунок 1.144 – Поля регистра dma_status

В таблице 1.570 указаны поля регистра dma_status.

Таблица 1.570 – Поля регистра dma_status

Бит	Название	Описание
[31:28]	test_status	Для уменьшения числа переключателей можно исключить логику проверки интеграции. Чтение: 0x0 = контроллер GPDMA не включает в себя логику проверки интеграции; 0x1 = контроллер GPDMA включает в себя логику проверки интеграции; 0x2-0xF = не определено
[27:21]	-	Не определено
[20:16]	chnls_minus1	Число доступных каналов ПДП минус единица. Например: b00000 = контроллер GPDMA использует 1 канал ПДП; b00001 = контроллер GPDMA использует 2 канала ПДП; b00010 = контроллер GPDMA использует 3 канала ПДП; . . . b11111 = контроллер GPDMA использует 32 канала ПДП
[15:8]	-	Не определено.
[7:4]	состояние	Текущее состояние управления устройства. Возможно следующее состояние: b0000 = режим ожидания; b0001 = чтение данных контроллера GPDMA канала; b0010 = чтение конечного указателя исходных данных; b0011 = чтение конечного указателя конечных данных; b0100 = чтение исходных данных; b0101 = запись конечных данных; b0110 = ожидание запроса ПДП для сброса; b0111 = запись данных контроллера GPDMA канала; b1000 = блокировка; b1001 = выполнено; b1010 = переключение на работу с периферией в режиме *scatter-gather*; b1011-b1111 = не определено
[3:1]	-	Не определено
[0]	master_enable	Статус активации контроллера GPDMA: 0 = контроллер GPDMA включен; 1 = контроллер GPDMA отключен

1.3.12.3.2.2 dma_cfg (0x004)

Регистр dma_cfg только для записи управляет конфигурацией контроллера GPDMA. В таблице 1.571 указаны поля регистра dma_cfg.

Таблица 1.571 – Поля регистра dma_cfg

Бит	Название	Описание
[31:8]	-	Не определено. Запись нуля.

Бит	Название	Описание
[7:5]	chnl_prot_ctrl	Устанавливает защиту AHB-Lite путем управления уровнями сигналов **HPROT[3:1]**: - **бит [7]** Управляет **HPROT[3]** для индикации кэшируемого доступа; - **бит [6]** Управляет **HPROT[2]** для индикации буферизованного доступа; - **бит [5]** Управляет **HPROT[1]** для индикации привилегированного доступа. Если бит [*n*] = 1, соответствующий **HPROT** будет ВЫСОКИМ. Если бит [*n*] = 0, соответствующий **HPROT** будет НИЗКИМ.
[4:1]	-	Не определено. Запись нуля.
[0]	master_enable	Включение контроллера GPDMA: 0 = отключение контроллера GPDMA 1 = включение контроллера GPDMA.

1.3.12.3.2.3 ctrl_base_ptr (0x008)

Регистр ctrl_base_ptr - указатель базового адреса управляющих данных канала. Следует настроить этот регистр, чтобы указатель базового адреса указывал ячейку памяти системы.

Контроллер GPDMA не имеет внутренней памяти для хранения структуры управляющих данных каналов.

Чтение этого регистра невозможно, когда контроллер GPDMA находится в состоянии сброса.

В таблице 1.572 указаны поля регистра ctrl_base_ptr.

Таблица 1.572 – Поля регистра ctrl_base_ptr

Бит	Название	Описание
[31:5]	ctrl_base_ptr	Указатель базового адреса первичной структуры данных. Информацию о структуре данных см. 1.3.12.2.2.3.
[4:0]	-	Не определено. Запись нуля.

1.3.12.3.2.4 alt_ctrl_base_ptr (0x00C)

Регистр alt_ctrl_base_ptr возвращает базовый адрес альтернативной структуры данных. Чтение этого регистра невозможно, когда контроллер GPDMA находится в состоянии сброса.

Благодаря этому регистру нет необходимости в прикладной программе для расчета базового адреса альтернативной структуры данных. В таблице 1.573 указаны поля регистра alt_ctrl_base_ptr.

Таблица 1.573 – Поля регистра alt_ctrl_base_ptr

Бит	Название	Описание
[31:0]	alt_ctrl_base_ptr	Базовый адрес альтернативной структуры данных

1.3.12.3.2.5 dma_waitonreq_status (0x010)

Регистр статуса ожидания каналом запроса возвращает состояние **dma_waitonreq[ ]**. Чтение этого регистра невозможно, когда контроллер GPDMA находится в состоянии сброса.

В таблице 1.574 указаны поля регистра dma_waitonreq_status.

Таблица 1.574 – Поля регистра dma_waitonreq_status

Бит	Название	Описание
[31:0]	dma_waitonreq_status	Статус ожидания каналом запроса. Чтение: - **Бит [*C*] = 0 dma_waitonreq[C]** НИЗКИЙ; - **Бит [*C*] = 1 dma_waitonreq[C]** ВЫСОКИЙ

1.3.12.3.2.6 chnl_sw_request (0x014)

Регистр программного запроса канала позволяет сгенерировать программный запрос ПДП.

В таблице 1.575 указаны поля регистра chnl_sw_request.

Таблица 1.575 – Поля регистра chnl_sw_request

Бит	Название	Описание
[31:0]	chnl_sw_request	Устанавливает соответствующий бит для генерирования программного запроса ПДП в соответствующем канале ПДП. Запись: - **Бит [*C*] = 0** Не создает запрос ПДП для канала C; - **Бит [*C*] = 1** Создает запрос ПДП для канала C. Запись бита, когда канал ПДП не активирован, не создает запрос ПДП для этого канала.

1.3.12.3.2.7 chnl_useburst_set (0x018)

Регистр установки useburst канала отключает для входа одного запроса **dma_sreq[]** возможность генерирования запросов, поэтому только запрос **dma_req[]** генерирует запросы. Чтение регистра возвращает состояние useburst.

В таблице 1.576 указаны поля регистра chnl_useburst_set.

Таблица 1.576 – Поля регистра chnl_useburst_set

Бит	Название	Описание
[31:0]	chnl_useburst_set	Возвращает состояние useburst или отключает для **dma_sreq[C]** возможность генерирования запросов ПДП. Чтение: - **бит [*C*] = 0** Канал ПДП С отвечает на запросы, получаемые на **dma_req[C]** или **dma_sreq[C]**. Контроллер GPDMA выполняет 2^R или одну передачу по шине; - **бит [*C*] = 1** Канал ПДП С не отвечает на запросы, получаемые на **dma_sreq[C]**. Контроллер GPDMA отвечает только на запросы **dma_req[C]** и выполняет 2^R передачи. Запись: - **бит [*C*] = 0** Нет действия. Использование регистра chnl_useburst_clr для установки бита [C] на 0; - **бит [*C*] = 1** Отключает для **dma_sreq[C]** возможность генерирования запросов ПДП. Контроллер GPDMA выполняет 2^R передач. При записи бита, когда канал ПДП отключен, какие-либо действия не выполняются

После завершения предпоследней 2^R передачи, если число оставшихся передач N меньше 2^R, то контроллер GPDMA сбрасывает бит chnl_useburst_set на 0. Это позволяет завершить оставшиеся передачи с помощью **dma_req[ ]** или **dma_sreq[ ]**.

При настройке channel_cfg со значением N меньше 2^R не следует устанавливать соответствующий бит chnl_useburst_set, если периферийное устройство не активировало **dma_req[ ]**.

При работе с периферией в режиме *scatter-gather*, если в channel_cfg установлен бит next_useburst, то контроллер GPDMA устанавливает бит chnl_useburst_set [C] на 1, когда выполняет цикл ПДП с использованием альтернативной структуры данных.

1.3.12.3.2.8 chnl_useburst_clr (0x01C)

Регистр сброса useburst канала включает для **dma_sreq[]** возможность генерирования запросов.

В таблице 1.577 указаны поля регистра chnl_useburst_clr.

Таблица 1.577 – Поля регистра chnl_useburst_clr

Бит	Название	Описание
[31:0]	chnl_useburst_clr	Устанавливает соответствующий бит для **dma_sreq[],** чтобы активировать генерирование запросов. Запись: - **бит [*C*] = 0** Нет действия. Регистр chnl_useburst_set используется для отключения генерирования запросов **dma_sreq[]**; - **бит [*C*] = 1** Включение генерирования запросов ПДП **dma_sreq[C]**. При записи бита, когда канал ПДП отключен, какие-либо действия не выполняются

1.3.12.3.2.9 chnl_req_mask_set (0x020)

Регистр настройки маски запроса канала.

В таблице 1.578 указаны поля регистра chnl_req_mask_set.

Таблица 1.578 – Поля регистра chnl_req_mask_set

Бит	Название	Описание
[31:0]	chnl_req_mask_set	Возвращает состояние маски запроса **dma_req[]** и **dma_sreq[]** или отключает у соответствующего канала возможность генерирования запросов ПДП. Чтение: - **бит [*C*] = 0** Внешние запросы для канала С включены; - **бит [*C*] = 1** Внешние запросы для канала С отключены. Запись: - **бит [*C*] = 0** Нет действия. Регистр chnl_req_mask_clr используется для включения запросов ПДП; - **бит [*C*] = 1** Отключает для **dma_req** и **dma_sreq[C]** возможность генерирования запросов ПДП. При записи бита, когда канал ПДП отключен, какие-либо действия не выполняются

1.3.12.3.2.10 chnl_req_mask_clr (0x024)

Регистр сброса маски запроса канала.

В таблице 1.579 указаны поля регистра chnl_req_mask_clr.

Таблица 1.579 – Поля регистра chnl_req_mask_clr

Бит	Название	Описание
[31:0]	chnl_req_mask_clr	Установка соответствующего бита для включения запросов ПДП для канала по **dma_req[]** и **dma_sreq[]**. Запись: - **бит [*C*] = 0** Нет действия. Регистр chnl_req_mask_set используется для отключения генерирования запросов для **dma_req[]** и **dma_sreq[]**; - **бит [*C*] = 1** Включение генерирования запросов ПДП для **dma_req[C]** или **dma_sreq[C]**. При записи бита, когда канал ПДП отключен, какие-либо действия не выполняются

1.3.12.3.2.11 chnl_enable_set (0x028)

Регистр chnl_enable_set включает канал ПДП. При чтении этот регистр возвращает состояние включения каналов.

В таблице 1.580 указаны поля регистра chnl_enable_set.

Таблица 1.580 – Поля регистра chnl_enable_set

Бит	Название	Описание
[31:0]	chnl_enable_set	Возвращает состояние включения каналов или включает соответствующие каналы. Чтение: - **бит [*C*] = 0** Канал С отключен; - **бит [*C*] = 1** Канал С включен. Запись: - **бит [*C*] = 0** Нет действия. Использование регистра chnl_enable_clr для отключения канала; - **бит [*C*] = 1** Включение канала С. При записи бита, когда канал ПДП отключен, какие-либо действия не выполняются

1.3.12.3.2.12 chnl_enable_clr (0x02C)

Регистр chnl_enable_clr отключает канал ПДП.

В таблице 1.581 указаны поля регистра chnl_enable_clr.

Таблица 1.581 – Поля регистра chnl_enable_clr

Бит	Название	Описание
[31:0]	chnl_enable_clr	Установка соответствующего бита для отключения соответствующего канала ПДП. Запись: - **бит [*C*] = 0** Нет действия. Использование регистра chnl_enable_set для включения каналов ПДП; - **бит [*C*] = 1** Отключение канала С. При записи бита, когда канал ПДП отключен, какие-либо действия не выполняются. Контроллер GPDMA отключает канал, устанавливая соответствующий бит, когда: - он завершает цикл ПДП; - он читает ячейку памяти channel_cfg, которая имеет cycle_ctrl = b000; - на шине AHB-Lite возникает ОШИБКА.

1.3.12.3.2.13 chnl_pri_alt_set (0x030)

Регистр настройки первичной и альтернативной структуры канала позволяет настроить канал ПДП на использование альтернативной структуры данных. При чтении регистр возвращает состояние структуры данных. используемой для соответствующего канала ПДП.

В таблице 1.582 указаны поля регистра chnl_pri_alt_set.

Таблица 1.582 – Поля регистра chnl_pri_alt_set

Бит	Название	Описание
[31:0]	chnl_pri_alt_set	Возвращает состояние структуры управляющих данных канала или выбирает альтернативную структуру данных для соответствующего канала ПДП. Чтение: – **бит [*C*] = 0** Канал ПДП С использует первичную структуру данных; – **бит [*C*] = 1** Канал ПДП С использует альтернативную структуру данных. Запись: – **бит [*C*] = 0** Нет действия. Использование регистра chnl_pri_alt_clr для установки бита [C] на 0; – **бит [*C*] = 1** Выбор альтернативной структуры данных для канала С. При записи бита, когда канал ПДП отключен, какие-либо действия не выполняются. Контроллер GPDMA переключает значение бита chnl_pri_alt_set [C] после завершения: – цикла ПДП из четырех передач с использованием первичной структуры данных при работе с памятью в режиме *scatter-gather* – всех передач с использованием первичной структуры данных в ходе цикла ПДП в режиме «пинг-понг»; – всех передач с использованием альтернативной структуры данных в ходе цикла ПДП следующих циклов: – «пинг-понг»; – работа с памятью в режиме *scatter-gather*.

1.3.12.3.2.14 chnl_pri_alt_clr (0x034)

Регистр сброса первичной и альтернативной структуры канала позволяет настроить канал ПДП на использование первичной структуры данных.

В таблице 1.583 указаны поля регистра chnl_pri_alt_clr.

Таблица 1.583 – Поля регистра chnl_pri_alt_clr

Бит	Название	Описание
[31:0]	chnl_pri_alt_clr	Установка соответствующего бита для выбора первичной структуры данных для соответствующего канала ПДП. Запись: – **бит [*C*] = 0** Нет действия. Использование регистра chnl_pri_alt_set для выбора альтернативной структуры данных; – **бит [*C*] = 1** Выбор первичной структуры данных для канала С. При записи бита, когда канал ПДП отключен, какие-либо действия не выполняются. Контроллер GPDMA переключает значение бита chnl_pri_alt_clr [C] после завершения: – цикла ПДП из четырех передач с использованием первичной структуры данных при работе с памятью в режиме *scatter-gather*; – всех передач с использованием первичной структуры данных в ходе цикла ПДП в режиме «пинг-понг»; – всех передач с использованием альтернативной структуры данных в ходе цикла ПДП следующих циклов: – «пинг-понг»; – работа с памятью в режиме *scatter-gather*.

1.3.12.3.2.15 chnl_priority_set (0x038)

Регистр настройки приоритета каналов позволяет настроить канал ПДП на использование высокого уровня приоритета. При чтении этот регистр возвращает состояние маски приоритета канала.

В таблице 1.584 указаны поля регистра chnl_priority_set.

Таблица 1.584 – Поля регистра chnl_priority_set

Бит	Название	Описание
[31:0]	chnl_priority_set	Возвращает состояние маски приоритета канала или устанавливает высокий приоритет канала. Чтение: – **бит [*C*] = 0** Канал С ПДП использует уровень приоритета по умолчанию; – **бит [*C*] = 1** Канал С ПДП использует высокий уровень приоритета. Запись: – **бит [*C*] = 0** Нет действия. Использование регистра chnl_priority_clr для назначения каналу С уровня приоритета по умолчанию; – **бит [*C*] = 1** Канал С имеет высокий уровень приоритета. При записи бита, когда канал ПДП отключен, какие-либо действия не выполняются.

1.3.12.3.2.16 chnl_priority_clr (0x03C)

Регистр сброса приоритета каналов позволяет настроить канал ПДП на использование уровня приоритета по умолчанию.

В таблице 1.585 указаны поля регистра chnl_priority_clr.

Таблица 1.585 – Поля регистра chnl_priority_clr

Бит	Название	Описание
[31:0]	chnl_priority_clr	Установка соответствующего бита для выбора уровня приоритета по умолчанию для определенного канала ПДП. Запись: - **бит [*C*] = 0** Нет действия. Использование регистра chnl_priority_set для назначения каналу С высокого уровня приоритета; - **бит [*C*] = 1** Канал С имеет уровень приоритета по умолчанию. При записи бита, когда канал ПДП отключен, какие-либо действия не выполняются.

1.3.12.3.2.17 err_clr (0x04C)

Регистр сброса ошибки шины возвращает статус **dma_err** и позволяет сбросить **dma_err**.

В таблице 1.586 указаны поля регистра err_clr.

Таблица 1.586 – Поля регистра err_clr

Бит	Название	Описание
[31:1]	-	Не определено. Запись нуля.

Бит	Название	Описание
[0]	err_clr	Возвращает состояние **dma_err** или устанавливает НИЗКИЙ сигнал. Чтение: – 0 = **dma_err** НИЗКИЙ; – 1 = **dma_err** ВЫСОКИЙ. Запись: – 0 = No нет действия, состояние **dma_err** не изменяется; – 1 = устанавливает НИЗКИЙ **dma_err**. Для тестирования задайте ВЫСОКИЙ **dma_err** с помощью регистра err_set. При деактивации **dma_err** в момент возникновения ОШИБКИ на шине AHB-Lite, состояние ОШИБКИ имеет приоритет, и **dma_err** остается отключенным.

1.3.13 Подсистема таймеров

1.3.13.1 Общее описание подсистемы таймеров

Подсистема таймеров МК Optimal+ включает три типа таймеров:

– SysTick - таймер процессорного ядра;

– SYSTIM - блок системных таймеров общего назначения;

– TIM_ - таймеры общего назначения с поддержкой ШИМ и CapCom.

В таблице 1.587 представлено сравнение основных характеристик таймеров МК Optimal+.

Таблица 1.587 – Характеристики таймеров в МК Optimal+

	SysTick	**SYSTIM**	**TIM_A0/1 - TIM_D0/1**
Разрядность таймера	24	32	16
Число каналов ШИМ или CAPCOM	0	0	4
Возможность счета таймера up/down	down only	down only	up/down
Изменение таймера по событию	-	-	+
Объединение каналов	-	-	+
Теневые регистры	-	+	+
Сброс таймера по событию	-	-	+
Прерывания по достижении периода	-	-	+
Прерывания по достижении нуля	-	+	+
Прерывания по совпадению	-	-	+
Одиночный режим	-	+	+
Схема энкодера	-	-	+
Dead Time	-	-	+
Генерация ШИМ	-	-	+
Приостановка генерации ШИМ	-	-	+
CapCom	-	-	+

1.3.13.2 Таймер процессорного ядра (SysTick)

1.3.13.2.1 Описание SysTick

Процессорное ядро имеет два экземпляра таймера:

– SysTick, Non-secure;

– SysTick, Secure.

SysTick - 24-разрядный таймер процессорного ядра. Он осуществляет отсчет вниз от заданного в регистре предела (SYST_RVR).

Значение SysTick изменяется каждый такт частоты работы процессорного ядра пока SysTick включен (SYST_CSR[ENABLE]). При достижении нуля устанавливается флаг SYST_CSR[COUNTFLAG], значение таймера перезагружается из SYST_RVR и таймер продолжает с него декрементироваться. После чтения флага SYST_CSR[COUNTFLAG] он сбрасывается.

Чтение из SYST_CVR возвращает текущее значение счетчика.

Запись в SYST_CVR очищает SYST_CVR и сбрасывает SYST_CSR[COUNTFLAG]. После этого на следующий так в SYST_CVR загружается значение из SYST_RVR.

Запись нулевого значения в SYST_RVR выключает SysTick при следующем достижении нуля. Нулевое значения будет сохраняться в счетчике, даже при SYST_CSR[ENABLE].

Установка SYST_CSR[TICKINT] в 1 разрешает генерацию исключений, когда счетчиком декрементируется до нуля.

Рекомендуется перед включением SysTick устанавливать необходимое значение периода в SYST_RVR.

Когда процессор остановлен для отладки все SysTick не изменяются.

1.3.13.2.2 Описание регистров

1.3.13.2.2.1 Карта регистров

Таблица 1.588 – Карта регистров SysTick

Адрес	Название	Доступ	Значение после сброса	Описание
0xE000E01C	SYST_CALIB	ЧТ/ЗП	0x80000000	Калибровочное значение
0xE000E010	SYST_CSR	ЧТ/ЗП	0x4	Настройка и статус SysTick
0xE000E018	SYST_CVR	ЧТ/ЗП	0x0	Текущее значение таймера

Адрес	Название	Доступ	Значение после сброса	Описание
0xE000E014	SYST_RVR	ЧТ/ЗП	0x0	Период таймера

1.3.13.2.2.2 Описание полей регистров

1.3.13.2.2.2.1 SYST_CALIB (0xE000E01C)

Регистр калибровочного значение SysTick

В таблице 1.589 представлено описание полей регистра.

Таблица 1.589 – Описание полей регистра SYST_CALIB

Биты	Поле	Доступ	Значение после сброса	Описание
23:0	TENMS	ЧТ	0	Десятки миллисекунд. Зарезервировано
29:24	-	ЧТ	0	Зарезервировано
30	SKEW	ЧТ	0	Зарезервировано
31	NOREF	ЧТ	1	Определяет реализован внешний опорный тактовый сигнал или нет: 0 - Внешний тактовый сигнал реализован 1 - Внешнего тактового сигнала не реализован. Если это бит установлен, то и SYST_CSR[CLKSOURCE] так же равен 1.

1.3.13.2.2.2.2 SYST_CSR (0xE000E010)

Регистр включения SysTick

В таблице 1.590 представлено описание полей регистра.

Таблица 1.590 – Описание полей регистра SYST_CSR

Биты	Поле	Доступ	Значение после сброса	Описание
0	ENABLE	ЧТ/ЗП	0	Включение SysTick: 0 - Счетчик выключен 1 - Счетчик включен
1	TICKINT	ЧТ/ЗП	0	Маскирование прерывания 0 - При достижении нуля счетчиком не будет сгенерировано исключение 1 - При достижении нуля счетчиком будет сгенерировано исключение
2	CLKSOURCE	ЧТ	1	Выбирает источник тактового сигнала 0 - Используется внешний тактовый сигнал 1 - Используется тактовый сигнал как и процессора
15:3	-	ЧТ	0	Зарезервировано

Биты	Поле	Доступ	Значение после сброса	Описание
16	COUNTFLAG	ЧТ/ЗП	0	Флаг достижения нуля. Флаг устанавливается когда счетчик перешел из 1 в 0. 0 - Счетчик не досчитал до нуля 1 - Счетчик досчитал до нуля. Данный флаг сбрасывается при чтении или во время записи в SYST_CVR.
31:17	-	ЧТ	0	Зарезервировано

1.3.13.2.2.2.3 SYST_CVR (0xE000E018)

Регистр текущего значения счетчика SysTick

В таблице 1.591 представлено описание полей регистра.

Таблица 1.591 – Описание полей регистра SYST_CVR

Биты	Поле	Доступ	Значение после сброса	Описание
23:0	CURRENT	ЧТ/ЗП	0	Текущее значение счетчика SysTick. Запись в данное поле любого значения сбрасывает SYST_CSR[COUTFLAG].
31:24	-	ЧТ	0	Зарезервировано

1.3.13.2.2.2.4 SYST_RVR (0xE000E014)

Регистр периода (начального значения) счетчика SysTick

В таблице 1.592 представлено описание полей регистра.

Таблица 1.592 – Описание полей регистра SYST_RVR

Биты	Поле	Доступ	Значение после сброса	Описание
23:0	RELOAD	ЧТ/ЗП	0	Задает период (начальное значение) счетчика.
31:24	-	ЧТ	0	Зарезервировано

1.3.13.3 Блок системных таймеров общего назначения (SYSTIM)

Блок системных таймеров общего назначения (SYSTIM) обладает следующими характеристиками:

– два независимых 32-разрядных таймера;

– для каждого таймера можно настроить режим работы:

– свободный счет;

– периодический;

– одиночный;

– каждый таймер имеет собственный выход прерывания;

– интерфейс программирования: APB slave.

1.3.13.3.1 Устройство блока системных таймеров общего назначения

Блок не имеет сигналов, соединенных с внешними выводами микросхемы. На рисунке 1.145 приведена структурная схема блока SYSTIM.

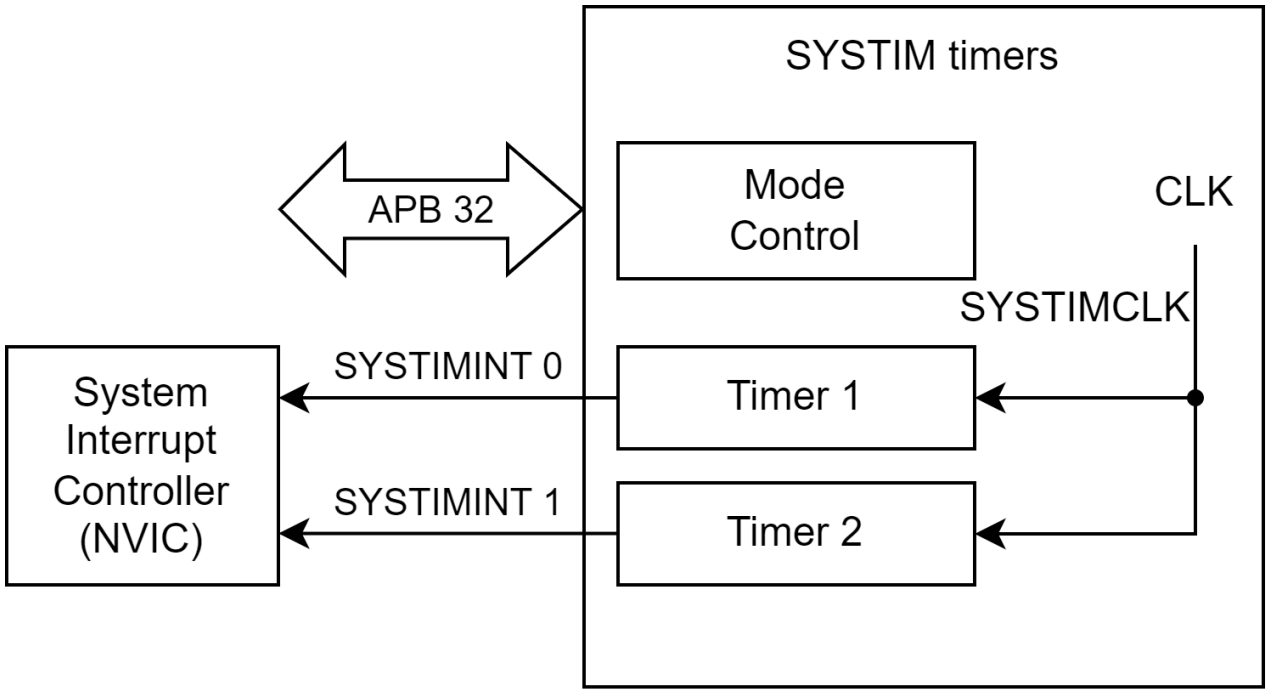


Рисунок 1.145 – Структурная схема блока системных таймеров

SYSTIM состоит из двух идентичных блоков Timer 1 и Timer 2, каждый из которых может функционировать в 16- или 32-битном режиме. Внутри каждого блока находится счетчик.

Оба таймера могут быть программно настроены на работу в одном из следующих режимов:

– свободный счет (free-running) – счетчик таймера постоянно декрементируется, счет автоматически начинается с максимального значения после достижения нуля;

– периодичный (periodic) – аналогично предыдущему, только после достижения нуля счет начинается со значения, предварительно загруженного в регистр TimerXLoad;

– одиночный (one-shot) – счетчик начинает декрементироваться со значения, загруженного в регистр TimerXLoad, после достижения нуля счет останавливается.

Ниже приведена последовательность работы с таймерами:

1) Необходимо провести начальные настройки используемого таймера, т. е. прописать поля регистра TimerXControl.

2) Загрузить начальное значение счетчика в регистр TimerXLoad.

3) Разрешить счет путем установки бита TimerEn регистра TimerXControl. Если после установки этого бита записать новое значение в регистр TimerXLoad, то счетчик продолжит декрементироваться с нового значения.

4) Другой способ записать новое значение счетчика заключается в записи фонового регистра TimerXBGLoad. Эта запись не будет иметь мгновенного эффекта, вместо этого записанное значение перепишется в регистр TimerXLoad автоматически по достижении счетчиком нулевого значения.

5) Текущее значение счетчика может быть считано из регистра TimerXValue в любой момент времени.

6) Каждый раз, когда счетчик достигает нулевого значения, генерируется прерывание. Для сброса сгенерированного прерывания необходимо произвести запись в регистр TimerXIntClr. Генерация прерывания может быть замаскирована путем установки соответствующих битов регистра управления TimerXControl.

1.3.13.3.2 Описание регистров

1.3.13.3.2.1 Общее описание регистров

Спецификация регистров блока системных таймеров SYSTIM представлена в таблице 1.593.

Состояние счетчика, разрядность и делитель частоты - это установки, которые не следует менять, пока Timer X работает. Во время записи новой конфигурации Timer X должен находиться в неактивном состоянии.

В дальнейшем при описании регистров использование X в названии означает, что он относится к Timer 1 либо к блоку Timer 2.

Регистры TimerPCellID0-3 – четыре 8-битных регистра, которые охватывают адресное пространство 0xFF0-0xFFC. Регистры, предназначенные только для чтения, могут быть концептуально представлены как 32-битный регистр. Регистр предназначен для идентификации стандартных периферийных блоков в составе системы.

1.3.13.3.2.2 Карта регистров

Таблица 1.593 – Карта регистров SYSTIM

Адрес	Название	Доступ	Значение после сброса	Описание
0x000	Timer1Load	ЧТ/ЗП	0x00000000	Регистр загрузки. TimerXLoad

Адрес	Название	Доступ	Значение после сброса	Описание
0x004	Timer1Value	ЧТ	0xFFFFFFFF	Регистр текущего значения. TimerXValue
0x008	Timer1Control	ЧТ/ЗП	0x20	Регистр управления. TimerXControl
0x00C	Timer1IntClr	ЗП	-	Регистр снятия прерываний. TimerXIntClr
0x010	Timer1RIS	ЧТ	0x0	Регистр состояния прерывания до наложения маски. TimerXRIS
0x014	Timer1MIS	ЧТ	0x0	Регистр состояния прерывания после наложения маски. TimerXMIS
0x018	Timer1BGLoad	ЧТ/ЗП	0x00000000	Фоновый регистр загрузки. TimerXBGLoad
0x020	Timer2Load	ЧТ/ЗП	0x00000000	Регистр загрузки. TimerXLoad
0x024	Timer2Value	ЧТ	0xFFFFFFFF	Регистр текущего значения. TimerXValue
0x028	Timer2Control	ЧТ/ЗП	0x20	Регистр управления. TimerXControl
0x02C	Timer2IntClr	ЗП	-	Регистр снятия прерываний. TimerXIntClr
0x030	Timer2RIS	ЧТ	0x0	Регистр состояния прерывания до наложения маски. TimerXRIS
0x034	Timer2MIS	ЧТ	0x0	Регистр состояния прерывания после наложения маски. TimerXMIS
0x038	Timer2BGLoad	ЧТ/ЗП	0x00000000	Фоновый регистр загрузки. TimerXBGLoad
0xF00	TimerITCR	ЧТ/ЗП	0x0	Регистр управления тестом интеграции. TimerITCR
0xF04	TimerITOP	ЗП	0x0	Регистр управления тестом интеграции. TimerITOP
0xFE0	TimerPeriphID0	ЧТ	0x04	Идентификатор периферийного устройства. TimerPeriphID0 биты [7:0]
0xFE4	TimerPeriphID1	ЧТ	0x18	Идентификатор периферийного устройства. TimerPeriphID1 биты [15:8]
0xFE8	TimerPeriphID2	ЧТ	0x04	Идентификатор периферийного устройства. TimerPeriphID2 биты [23:16]
0xFEC	TimerPeriphID3	ЧТ	0x00	Идентификатор периферийного устройства. TimerPeriphID3 биты [31:24]
0xFF0	TimerPCellID0	ЧТ	0x0D	Идентификатор. TimerPCellID0 биты [7:0]
0xFF4	TimerPCellID1	ЧТ	0xF0	Идентификатор. TimerPCellID1 биты [15:8]
0xFF8	TimerPCellID2	ЧТ	0x05	Идентификатор. TimerPCellID2 биты [23:16]
0xFFC	TimerPCellID3	ЧТ	0xB1	Идентификатор. TimerPCellID3 биты [31:24]

1.3.13.3.2.3 Описание полей регистров

1.3.13.3.2.3.1 TimerXLoad (0x000 или 0x020)

Регистр загрузки SYSTIM

В таблице 1.594 представлено описание полей регистра.

Таблица 1.594 – Описание полей регистра TimerXLoad

Биты	Поле	Доступ	Значение после сброса	Описание
31:0	TimerXLoad	ЧТ/ЗП	0	После записи в данный регистр счетчик должен начать декрементироваться с этого значения. Если периодический режим активен и текущее значение счетчика достигает нуля, то происходит перезагрузка счетчика. После записи в этот регистр текущее значение счетчика незамедлительно меняется на значение, записанное по активному фронту синхросигнала блока. Отметим, что минимальным значением для TimerXLoad является 1. Если значение TimerXLoad установлено в 0, то прерывания вырабатываются незамедлительно. Значение, считанное из TimerXLoad, - это значение, которое будет прописано в счетчике, как только он станет равным нулю в периодическом режиме.

1.3.13.3.2.3.2 TimerXValue (0x004 или 0x024)

Регистр текущего значения SYSTIM

В таблице 1.595 представлено описание полей регистра.

Таблица 1.595 – Описание полей регистра TimerXValue

Биты	Поле	Доступ	Значение после сброса	Описание
31:0	TimerXValue	ЧТ	0	Содержит текущее значение декрементирующего счетчика. После процесса загрузки, когда прописывается новое значение в TimerXLoad, в регистре TimerXValue незамедлительно отображается новое загруженное значение. Отметим, что старшие 16 бит 32-битного регистра TimerXValue не должны автоматически устанавливаться в нуль, когда используется 16-битный режим. Если TimerXValue находится в 16-битном режиме, тогда старшие 16 битов регистра TimerXValue должны иметь ненулевые значения, если перед этим TimerXValue был в 32-битном режиме, и произошла запись в регистр TimerXLoad, прежде чем перейти в 16-битный режим. То есть в 16 битном режиме старшие 16 бит сохраняют свое предыдущее значение.

1.3.13.3.2.3.3 TimerXControl (0x008 или 0x028)

Регистр управления SYSTIM

В таблице 1.596 представлено описание полей регистра.

Таблица 1.596 – Описание полей регистра TimerXControl

Биты	Поле	До-ступ	Значение по-сле сброса	Описание
0	OneShot	ЧТ/ЗП	0	Выбирает однократный или многократный режим счетчика: 0 - многократный режим (по умолчанию) 1 - однократный режим
1	TimerSize	ЧТ/ЗП	0	Выбирает 16- или 32-битный счетчик - используется: 0 - 16-битный счетчик (по умолчанию) 1 - 32-битный счетчик
3:2	TimerPre	ЧТ/ЗП	0	Биты масштабирования частоты счета (делитель частоты): 00 - синхросигнал не делится (по умолчанию) 01 - синхросигнал делится на 16 10 - синхросигнал делится на 256 11 - Зарезервировано
4	-	ЧТ	0	Зарезервировано
5	IntEnable	ЧТ/ЗП	0	Бит разрешения прерываний: 0 - Прерывание от Timer X неактивно 1 - Прерывание от Timer X активно (по умолчанию)
6	TimerMode	ЧТ/ЗП	0	Бит режима: 0 - Timer X находится в режиме свободного доступа (по умолчанию). 1 - Timer X находится в периодическом режиме
7	TimerEn	ЧТ/ЗП	0	Бит разрешения: 0 - Timer X неактивен (по умолчанию) 1 - Timer X активен
31:8	-	ЧТ	0	Зарезервировано

1.3.13.3.2.3.4 TimerXIntClr (0x00C или 0x02C)

Регистр сброса прерывания SYSTIM

В таблице 1.597 представлено описание полей регистра.

Таблица 1.597 – Описание полей регистра TimerXIntClr

Биты	Поле	До-ступ	Значение по-сле сброса	Описание
31:0	TimerXIntClr	ЗП	0	Запись любого значения в данный регистр снимает запрос на прерывание Timer X.

1.3.13.3.2.3.5 TimerXRIS (0x010 или 0x030)

Регистр состояния прерывания до наложения маски SYSTIM

В таблице 1.598 представлено описание полей регистра.

Таблица 1.598 – Описание полей регистра TimerXRIS

Биты	Поле	Доступ	Значение после сброса	Описание
31:1	-	ЧТ	0	Зарезервировано
0	TimerXRIS	ЧТ	0	Состояние запроса на прерывание до наложения маски

1.3.13.3.2.3.6 TimerXMIS (0x014 или 0x034)

Регистр состояния прерывания после наложения маски SYSTIM

В таблице 1.599 представлено описание полей регистра.

Таблица 1.599 – Описание полей регистра TimerXMIS

Биты	Поле	Доступ	Значение после сброса	Описание
31:1	-	ЧТ	0	Зарезервировано
0	TimerXMIS	ЧТ	0	Состояние запроса на прерывание после наложения маски

1.3.13.3.2.3.7 TimerXBGLoad (0x018 или 0x038)

Регистр теневой загрузки периода SYSTIM

В таблице 1.600 представлено описание полей регистра.

Таблица 1.600 – Описание полей регистра TimerXBGLoad

Биты	Поле	Доступ	Значение после сброса	Описание
31:0	TimerXBGLoad	ЧТ	0	Регистр содержит значение с которого счетчик начинает декрементироваться. Данное значение используется для перезагрузки счетчика в периодическом режиме, когда текущее значение счета достигает нуля. Регистр TimerXBGLoad предоставляет альтернативный метод доступа к регистру TimerXLoad. Отличием является то, что запись в TimerXBGLoad не означает, что счетчик устанавливает новое значение сразу. Чтение из данного регистра вернет то же самое значение, что и из TimerXLoad.

1.3.13.3.2.3.8 TimerPeriphID0 (0xFE0)

Регистр идентификатора SYSTIM

В таблице 1.601 представлено описание полей регистра.

Таблица 1.601 – Описание полей регистра TimerPeriphID0

Биты	Поле	До-ступ	Значение по-сле сброса	Описание
7:0	PartNumber	ЧТ	0x04	Содержит шифр компонента периферийного блока. Для SYSTIM - 0x04
31:8	-	ЧТ	0	Зарезервировано

1.3.13.3.2.3.9 TimerPeriphID1 (0xFE4)

Регистр идентификатора SYSTIM

В таблице 1.602 представлено описание полей регистра.

Таблица 1.602 – Описание полей регистра TimerPeriphID1

Биты	Поле	До-ступ	Значение по-сле сброса	Описание
7:0	DesignerID	ЧТ	0x18	Идентификационный номер проектировщика блока.
31:8	-	ЧТ	0	Зарезервировано

1.3.13.3.2.3.10 TimerPeriphID2 (0xFE8)

Регистр идентификатора SYSTIM

В таблице 1.603 представлено описание полей регистра.

Таблица 1.603 – Описание полей регистра TimerPeriphID2

Биты	Поле	До-ступ	Значение по-сле сброса	Описание
7:0	Revision	ЧТ	0x4	Номер ревизии периферийного блока. Номер ревизии начинается с нуля
31:8	-	ЧТ	0	Зарезервировано

1.3.13.3.2.3.11 TimerPeriphID3 (0xFEC)

Регистр идентификатора SYSTIM

В таблице 1.604 представлено описание полей регистра.

Таблица 1.604 – Описание полей регистра TimerPeriphID3

Биты	Поле	До-ступ	Значение по-сле сброса	Описание
7:0	Configuration	ЧТ	0	Вариант конфигурации. Для SYSTIM 0x0
31:8	-	ЧТ	0	Зарезервировано

1.3.13.3.2.3.12 TimerITCR (0xF00)

Регистр включения «теста интеграции» SYSTIM

В таблице 1.605 представлено описание полей регистра.

Таблица 1.605 – Описание полей регистра TimerITCR

Биты	Поле	Доступ	Значение после сброса	Описание
0	ITEN	ЧТ/ЗП	0	Разрешение режима «теста интеграции». Когда этот бит установлен в 1, SYSTIM переходит в режим теста интеграции, иначе находится в нормальном режиме. «Теста интеграции» - проверки правильности интеграции периферийного блока в систему Когда используется этот режим, маскированные сигналы прерывания напрямую контролируются регистром TimerITOP.
31:1	-	ЧТ	0	Зарезервировано

1.3.13.3.2.3.13 TimerITOP (0xF04)

Регистр значение прерываний в режиме «теста интеграции» SYSTIM

В таблице 1.606 представлено описание полей регистра.

Таблица 1.606 – Описание полей регистра TimerITOP

Биты	Поле	Доступ	Значение после сброса	Описание
0	TIMERINT2	ЗП	0	Значение с выхода TIMERINT2, в режиме теста интеграции
1	TIMERINT1	ЗП	0	Значение с выхода TIMERINT1, в режиме теста интеграции
31:1	-	ЧТ	0	Зарезервировано

1.3.13.3.3 Прерывания

Блок имеет два прерывания, каждое из которых подключено к системному контроллеру прерываний.

1.3.13.4 Таймеры общего назначения (TIM)

МК Optimal+ включает 8 таймеров общего назначения (TIM_A0/1 - TIM_D0/1). Каждый из таймеров общего назначения обладает следующими характеристиками:

– 16-битный счетчик вверх(up), вниз(down), вверх/вниз(up/down);

– 16-битный делитель (прескейлер);

– 4 канала для генерации ШИМ, захвата или сравнения (CapCom);

– внешний сигнал синхронизации для счётчиков;

– режимы работы:

– режим одиночного таймера;

– режим повторения таймера;

– поддержка "мертвой зоны" выходного сигнала (dead zone);

– синхронизация таймера от внешних событий (input trigger);

– схема фильтрации входного сигнал при захвате (input filtering);

– остановка PWM по внешнему сигналу (break);

– генерация внешних событий и прерываний;

– при обновлении таймера;

– при захвате значения на одном из каналов;

– схема энкодера;

– объединение таймеров;

– объединение каналов таймера;

– интерфейс программирования: APB slave.

1.3.13.4.1 Структурная схема таймеров общего назначения (TIM)

Далее по тексту будет для краткости в качестве обозначения таймеров общего назначения будут использоваться следующие синонимичные термины: TIM, таймер.

На рис. 1.146 представлена структурная схема таймеров общего назначения, на которой приняты следующие обозначения:

– заглавными буквами указаны входы и выходы, которые соединяются с внешними портами ввода/вывода;

– маленькими буквами указаны входы и выходы, которые соединяются с другими блоками внутри кристалла.

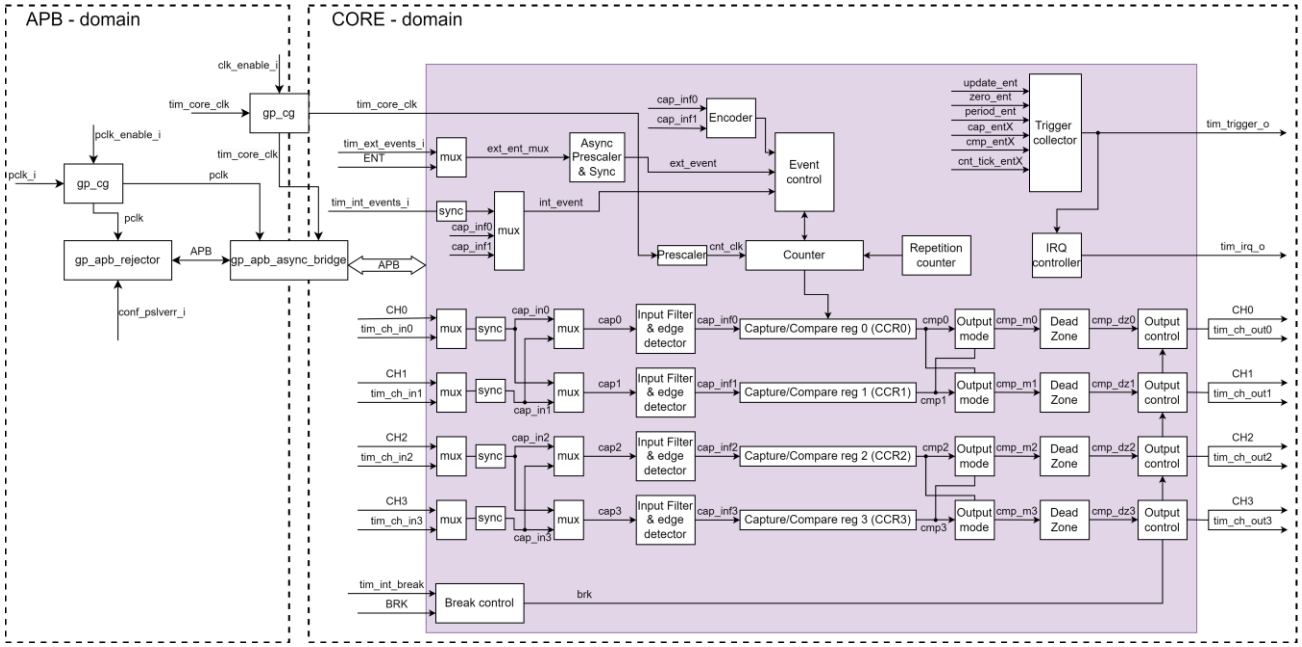


Рисунок 1.146 – Структурная схема таймеров TIM

В таб. 1.607 приведено описание составных блоков таймеров общего назначения.

Таблица 1.607 – Описание назначения составных блоков TIM

Название блока	**Назначение и основные функции**
Counter	Основной 16 битный счетчик. На его основе выдается ШИМ.
Event control	Детектирует все события и перенаправляет их в счетчик для изменения его состояния.
Repetition counter	Счетчик повторений в режиме repeat.
Trigger collector	Данный блок собирает все события TIM и выдает одно из них на ELC
Break control	Детектирует break события и останавливает выдачу ШИМ.
gp_apb_rejector	Блок служит для корректного завершения APB транзакций, когда с APB части USART контроллера снят тактовый сигнал. Данный блок может завершать транзакции с ошибкой (pslverr_o в единице) или без нее. Это задается с помощью входа conf_pslverr_i. Если на этом входе единица, то транзакции будут завершаться с ошибкой, когда на USART не подан тактовый сигнал. При этом транзакции записи игнорируются, а чтение возвращает 0x0BAD_ACCE5.
gp_cg	Блок служит для блокирования подачи тактового сигнала. Так как USART имеет два тактовый входа: pclk_i и clk_i, то и блоков gp_cg тоже используется два. Соответственно входов для блокировки тактовый сигналов также два: pclk_enable_i и clk_enable_i. Если на входе блокировки тактового сигнала единицы, то это тактовый сигнал будет снят. Так же данный блок включает в себя синхронизаторы сброса, они дают задержку на снятие сброс в два такта той частоты, на которой они работают, а сброс происходит асинхронно.
gp_apb_async_bridge	Блок служит для создания асинхронной развязки интерфейса APB. Это позволяет разделить TIM на два домена: apb-домен и core-домен. Первый работает от системном тактового сигнала, а второй от tim_clk

1.3.13.4.2 Описание внешних выводов блоков TIM

Список внешних выводов блоков TIM и их описание приведены в таблице 1.608

Таблица 1.608 – Список внешних выводов блоков TIM

Наименование вывода	Тип (Вх/Вых)	Функциональное назначение
CH0	Вх/Вых	Линия захвата или выдачи ШИМ сигнала
CH1	Вх/Вых	Линия захвата или выдачи ШИМ сигнала
CH2	Вх/Вых	Линия захвата или выдачи ШИМ сигнала
CH3	Вх/Вых	Линия захвата или выдачи ШИМ сигнала
ENT	Вх	Внешний тактовый сигнал или событие
BRK	Вх	Сигнал прекращения выдачи ШИМ

1.3.13.4.3 Описание счетчика

Основой таймера является счетчик, который служит для отсчета определенного временного интервала, периода. Когда TIM служит для создания ШИМ, то счетчик задает опорную частоту генерируемого сигнала. Период устанавливается в регистре TIM_CNT_PRD.

Счетчик может быть включен несколькими способами:

– программно. Установка бита TIM_CNT_CTRL[EN] или программная загрузка фазы TIM_CNT_PHASE;

– аппаратно. После возникновения одного или нескольких событий Start, Load_phase или EN_reset. Для более подробного описания см. пп 1.3.13.4.5.

Счетчик поддерживает три режима счета: вверх(up), вниз(down) и вверх/вниз(updown). Выбор режима осуществляется с помощью битов TIM_CNT_CTRL[CNT_MODE]. В первых двух режимах реальный период счетчика будет равен значению, указанному в регистре TIM_CNT_PRD плюс один, а в updown — удвоенному значению TIM_CNT_PRD.

Сам счетчик может работать в трех режимах:

– бесконечный периодический (infinity). После включения работает бесконечно, пока его не выключат;

– повторяющийся (repeat). После включения счетчик отсчитает указанное в регистре TIM_CNT_RPT количество периодов и выключается (бит TIM_CNT_CTRL[EN] установится в нуль);

– одиночный (single). После включения отсчитает один период и выключится.

Если в регистре TIM_CNT_RPT находится ненулевое значение и не включен одиночный режим, то счетчик работает в периодическом режиме, а как только он дойдет до нуля, то

счетчик выключится. Однако, если во время декрементирования будет записано новое ненулевое значение в теневой регистр, то при достижении нуля, это значение будет скопировано в рабочий регистр, теневая копия обнулится, и счетчик продолжит считать.

Еще одной возможностью счетчика является загрузка в него определенного значения TIM_CNT_PHASE. Эту операцию можно вызвать или программно, путем записи в регистр TIM_CNT_PHASE или аппаратно, после возникновения события Load_phase. Во время загрузки происходит автоматическое включение таймера, если он не был включен. При этом значение фазы всегда загружается в счетчик вне зависимости от того включен он или выключен. Если программная и аппаратная загрузки происходят одновременно, то в TIM_CNT загрузится новое значение из TIM_CNT_PHASE.

Во время Load_phase события можно изменить направления счета счетчика только, когда счетчик находится в режиме updown. При этом сам режим счета TIM_CNT_CTRL[CNT_MODE] не меняется (см. рис. 1.147). Направление счета выбирается с помощью бита TIM_CNT_CTRL[PHASE_DIR].

TIM_CNT_PHASE должно быть меньше или равно TIM_CNT_PRD.

Если для какого-то из входов событий включено действие EN_reset, то счетчик работает в бесконечном режиме.

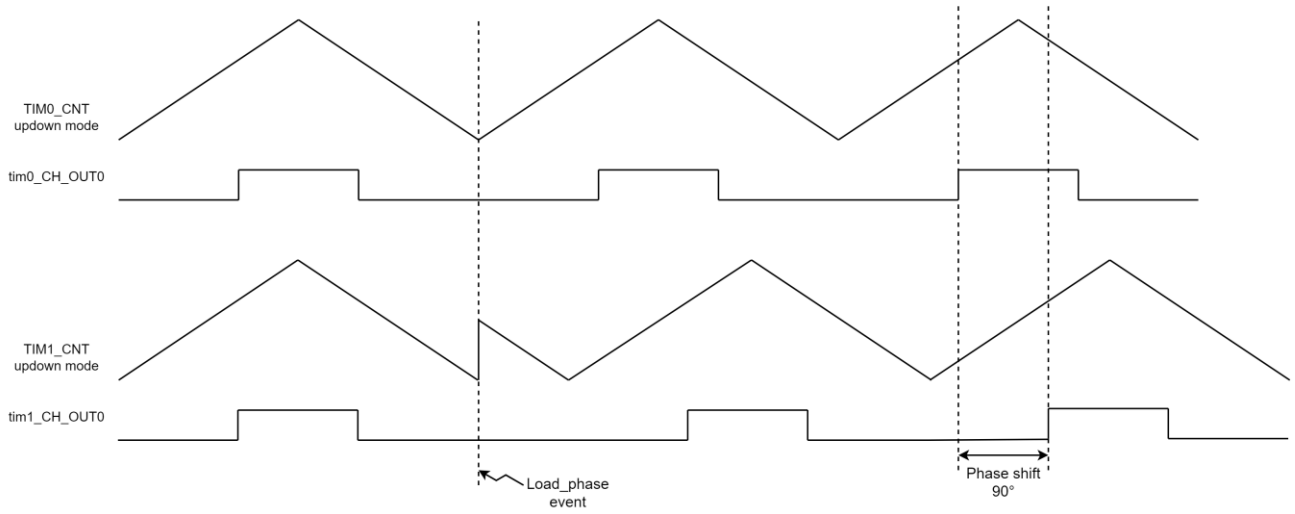


Рисунок 1.147 – Задание фазового сдвига

Пока счетчик выключен, его значение равно нулю в режимах **up** и **updown**, а в режиме **down** — периоду. Поэтому если одновременно включить счетчик и изменить режим с up(updown) на down или с down на up(updown), то произойдет задержка на один тик счетчика.

Это происходит из-за того, что необходимо в счетчик установить начальное значение нового режима.

Счетчик может тактироваться от нескольких источников. Они генерируют сигнал триггера для него (cnt_trig). Ниже приведен список, в котором источники тактирования указаны в порядке уменьшения приоритета:

– Режим энкодера. Счетчик будет изменяться по сигналам CH_IN0 и CH_IN1.

– По внешним или внутренним событиям.

– Если не включен ни один из вышеописанных режимов, то счетчик будет изменяться по тактовому сигналу tim_clk с настроенным делителем.

Текущее значение счетчика может быть прочитано с помощью регистра TIM_CNT.

1.3.13.4.4 Теневые регистры

В таймерах TIM некоторые регистры имеют теневые копии. В их список входят:

– TIM_CNT_PRD;

– TIM_CNT_PRESCALER;

– TIM_CCRx;

– TIM_CNT_RPT.

Основным назначением теневых регистров является обновление рабочих регистров "на ходу", то есть во время работы блока в строго определенный момент времени. Это происходит по событию UPDATE_EVENT (см. пп 1.3.13.4.5).

Теневые регистры включены когда, включен таймер и бит TIM_CNT_CTRL[SHDWn] = 0. Если они включены, то это означает, что по событию UPDATE_EVENT будет происходить обновление рабочих регистров из теневых.

Чтение из регистров, у которых есть теневые копии, всегда возвращает значение из рабочих регистров. То есть читается то значение, которое сейчас используется в работе блока. Запись в эти регистры всегда обновляет теневые копии, даже когда они выключены.

Для регистра TIM_CNT_RPT (и его теневой регистр) предусмотрен вспомогательный регистр CNT_RPT_STATUS, который отображает наличие значения в теневом регистре, которое еще не было загружено в рабочий. CNT_RPT_STATUS устанавливается, когда происходит запись в TIM_CNT_RPT не нулевого значения и включены теневые регистры. CNT_RPT_STATUS сбрасывается автоматически при обновлении или выключении теневых регистров. Копирование из теневого регистра в рабочий происходит по событию UPDATE_EVENT, при этом CNT_RPT_STATUS должен быть установлен.

Теневая копия TIM_CNT_RPT всегда обновляется при записи, когда теневые регистры включены. Когда теневые регистры выключены, то теневая копия TIM_CNT_RPT всегда равна нулю.

На рис. 1.148 приведен пример аппаратного обновления регистра TIM_CCRx. Но так же это можно сделать и программно. Для этого необходимо установить бит TIM_CNT_CTRL[SUR], после чего произойдет обновление регистров, и данный бит сбросится автоматически. Программное обновление теневых регистров произойдет только в том случае, если на следующий такт после записи в регистре TIM_CNT_CTRL поля будут имеют следующие значения: EN = 1, SHDWn = 0, SUR = 1.

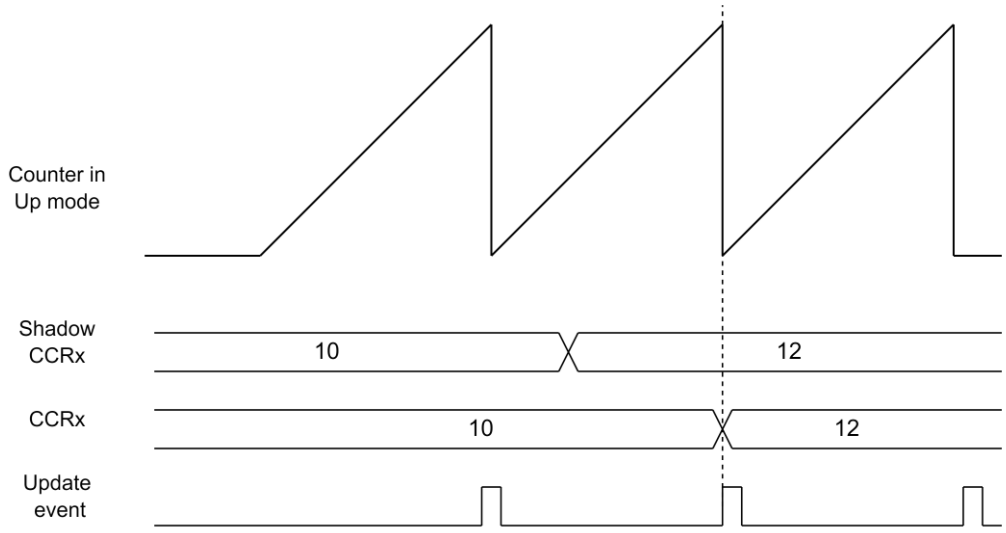


Рисунок 1.148 – Обновление регистров из теневых копий.

Теневые регистры выключены, когда счетчик выключен или бит TIM_CNT_CTRL[SHDWn] установлен.

Регистры TIM_CCRx имеют более сложную логику записи и считывания см. таблицу 1.609. При включении, выключении или смене направления канала значение регистров TIM_CCRx и их теневых копий не меняется.

Таблица 1.609 – Доступность регистров TIM_CCRx

Направление канала	**Теневые регистры**	**Состояние канала**	**Из какого регистра считываются данные**	**В какой регистр пишутся данные**
вход	безразлично	выключен	основной	основной
вход	безразлично	включен	основной	ни в какой
выход	безразлично	выключен	основной	основной и теневой

Направление канала	Теневые регистры	Состояние канала	Из какого регистра считываются данные	В какой регистр пишутся данные
выход	выключены	включен	основной	основной и теневой
выход	включены	включен	основной	теневой

Событие UPDATE_EVENT распространяется на регистры TIM_CCRx только, когда канал включен (TIM_CH_CTRLx[CH_EN] = 1) и настроен на выход (TIM_CH_CTRLx[CH_DIR] = 1). Если канал включен на выход и происходит одновременная запись в регистр TIM_CCRx и событие UPDATE_EVENT, то в рабочий и теневой регистры запишется значение с шины APB.

1.3.13.4.5 События

События — это момент во времени, когда на входах таймера происходит изменение входного сигнала или внутри TIM достигаются определенные условия. Поэтому события можно разделить по типу источника на:

– Внешние :

– ext_ent. Их источником являются изменения состояния буферов ввода/вывода.

– int_ent. Его источником является ELC.

– Внутренние. Их источником является сам таймер. Например, достижение счетчиком нуля.

Сигналы от внешних устройств являются асинхронными и могут иметь частоту большую, чем sys_clk. Чтобы впоследствии сигнал можно было синхронизировать, необходимо понизить его частоту, чтобы она удовлетворяла следующему условию: $\nu_{ext} < 2 \times \nu_{sys_clk}$. Для этого соответствующим образом настраивается асинхронный делитель с помощью битов TIM_EVENT_CTRL[EE_PSC]. После настройки асинхронного делителя сигналы ext_ent и int_ent поступают в Event control блок, где из них выделяется активный сигнал, и генерируются внешние события, которые приводят к выполнению определенного действия (см. таб. 1.611).

Асинхронный делитель можно использовать только при выбранном действии Trigger, а активатором этого действия должен быть либо фронт, либо срез сигнала на одном из выбранных входов: tim_ENT или tim_ext_events_i.

Для внешних событий ext_ent и int_ent можно отдельно настроить свое действие. Однако при этом необходимо следить за тем, чтобы действия не противоречили друг другу. Например, если ext_ent событие управляет включением таймера, то int_ent, настроенное на действие Start, не будет работать.

Схема блока управления событиями представлена на рис. 1.149.

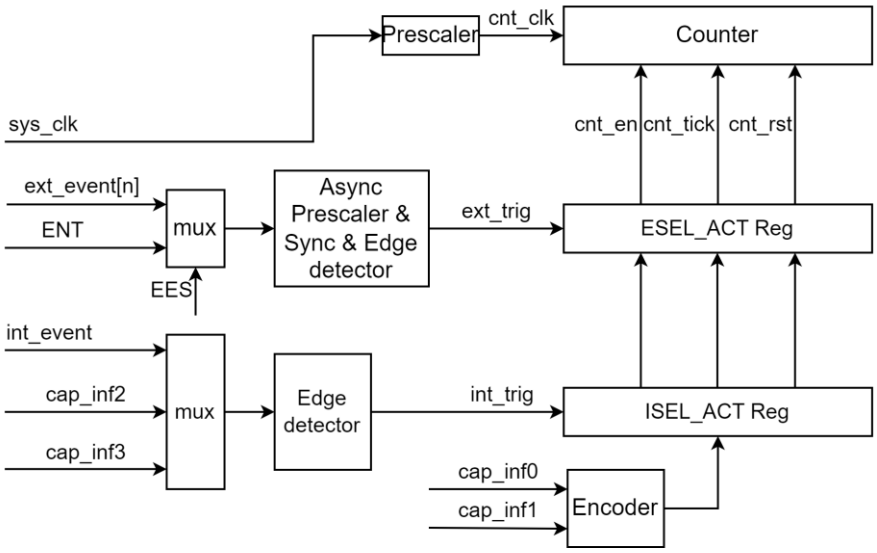


Рисунок 1.149 – Схема блока Event control

Таблица 1.610 – Входящие события блока TIM

Наименование сигнала	**Описание**
tim_ENT	Внешний вход события, который выводится на буфер ввода/вывода.
tim_ext_events_i	Выход с компаратора comp0

Таблица 1.611 – Виды действий блока TIM

Название	**Описание**
None	Ничего не происходит
Start	Когда происходит событие Start счетчик включается. Если данное событие происходит, когда счетчик уже включен, ничего не происходит.
EN_reset	Пока входы внешних событий находятся в активном состоянии, счетчик работает. Когда они переходят в неактивное состояние, то счетчик выключается. Активное состояние выбирается с помощью EXT_EVENT_DIR и INT_EVENT_DIR битовых полей регистра TIM_EVENT_CTRL. После выбора другого действия, то есть переключение с EN_reset на любое другое, счетчик автоматически выключается. Это происходит для обоих внешних событий. Данное действие может работать только по уровню внешних событий.
Trigger	Счетчик изменяется по каждому событию, если он включен.
Load_phase	Запись значения из TIM_CNT_PHASE в TIM_CNT и включение счетчика, если он был выключен.

События, которые включают или выключают счетчик имеют следующие приоритеты:

– Если включен режим EN_reset для любого из внешних событий, то только он управляет включением/выключением таймера. Это так же означает, что single и repeat режимы не работают, статусный бит TIM_IRQ_STATUS[INTERVAL_END_EVENT] не будет выставляться и счетчик оставшихся повторений не дектрементируется;

– Если нигде не выбран режим EN_reset, то все события (программные и аппаратные), которые выключают таймер, имеют приоритет над включением.

Блок TIM генерирует события для взаимодействия с другими блоками:

– UPDATE_EVENT (Update registers event). Во время этого события обновляются все теневые регистры. Оно происходит, при любом из следующих условий:

– происходит событие ZERO_EVENT. При этом repeat режим не должен быть активен;

– происходит событие PRD_EVENT. При этом repeat режим не должен быть активен;

– программное обновление с помощью SUR бита;

– если счетчик работает в повторяющемся режиме, то обновление происходит, когда заканчиваются все повторения.

– ZERO_EVENT (Zeroing counter event). Событие происходит, когда счетчик достигает нуля и при этом включен CNT_CTRL[EN]=1, а направление счета - вниз. Достигает нуля - это означает, что на предыдущем триггере счетчика (cnt_trig) значение счетчика не было равно нулю.

– PRD_EVENT (Period reach event). Событие происходит, когда счетчик достигает значения периода (CNT_PRD) и при этом включен CNT_CTRL[EN]=1, а направление счета - вверх. Достигает периода - это означает, что на предыдущем триггере счетчика (cnt_trig) значение счетчика не было равно периоду.

– CAPx_EVENT (Capture event). Событие происходит, когда после захвата импульса с выбранного канала текущее значение счётчика таймера копируется в регистр TIM_CCRx.

– CMPx_EVENT (Compare event). Событие происходит, когда значение счетчика совпадает с TIM_CCRx любого канала, который включен и работает как выход.

– CNT_TICK_EVENT. Событие происходит, когда значение счетчика изменяется.

– BREAK_EVENT. Событие происходит, когда блок Break распознает на одном из своих входов активный сигнал или происходит программный вызов break события.

– INTERVAL_END_EVENT. Событие происходит после того, как заканчивается период в одиночном режиме или после окончания всех повторений (переход TIM_CNT_RPT из 1 в 0 и при этом в теневой копии больше нет нового значения) в repeat режиме.

На рис. 1.150 и 1.151 изображены примеры временных диаграмм для событий ZERO_EVENT и PRD_EVENT.

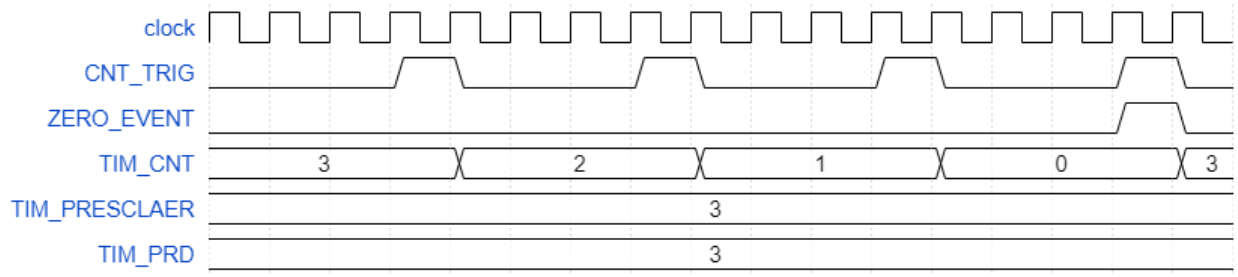


Рисунок 1.150 – Генерирование события zero_event.

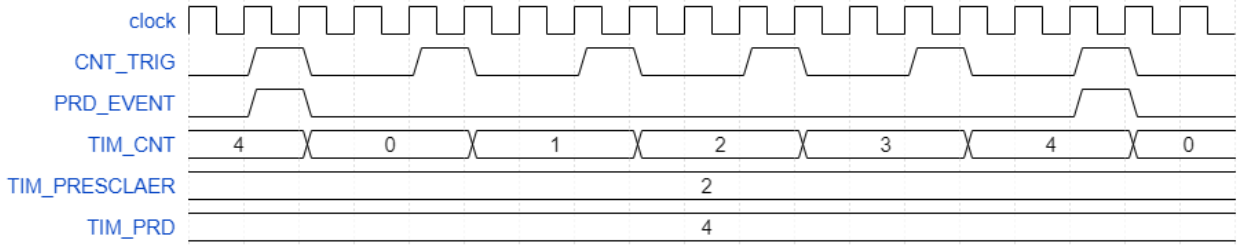


Рисунок 1.151 – Генерирование события prd_event.

1.3.13.4.6 Прерывания

Прерывания отслеживаются, маскируются и сбрасываются с помощью регистров TIM_IRQ_STATUS, TIM_IRQ_MASK, TIM_IRQ_TEST и TIM_IRQ_CLEAR. После того, как происходит одно из событий, которое проецируется в TIM_IRQ_STATUS, соответствующий бит в статусном регистре устанавливается до тех пор, пока не будет сброшено прерывание. Прерывание сбрасываются побитно после записи единицы в соответствующий бит регистра TIM_IRQ_CLEAR. Прерывания являются уровневыми.

Все прерывания можно замаскировать. Для этого используется регистр TIM_IRQ_MASK. Если в каком-то бите установлена единица, то это прерывание разрешено. То, что прерывание замаскировано, не означает, что в статусном регистре не будут устанавливаться соответствующие биты.

Флаги FULL и EMPTY отображают актуальность состояния буфера (см. рис. 1.3.13.4.9), поэтому могут быть сброшены записью в регистр TIM_IRQ_CLEAR только после наполнения или опустошения буфера.

После сброса все прерывания замаскированы.

1.3.13.4.7 Канал захвата

Принцип описания работы канала захвата основан на том пути, который проходит входной сигнал. Сначала выбирается входной сигнал и синхронизируется. Далее он фильтруется, и в нем детектируется фронт, срез или любое изменение. Как только сигнал будет удовлетворять

заданным требованиям, текущее значение счетчика заносится в регистр TIM_CCRx данного канала. Это событие называется захватом (cap_entX). Параметры входного сигнала настраиваются с помощью регистров TIM_CH_CTRLx.

Синхронизация и детектирование входного сигнала вносит задержку до 4 тактов частоты tim_clk.

Фильтрация входного сигнала происходит на частоте tim_clk. Для того, что бы входной сигнал прошел фильтрацию необходимо, что бы он после изменения (фронт или срез) не менялся в течении CH_FLT_I + 1 тактов.

Например, выбран режим детектирования по фронту импульса и длительность фильтрации выставлена на 5 периодов (TIM_CH_CTRLx[CH_FLT_I] = 4). Тогда если входной сигнал перейдет из нуля в единицу и будет установлен в единице не менее чем на шести фронтах tim_clk, идущих подряд, то произойдет событие захвата (см. рис. 1.152).

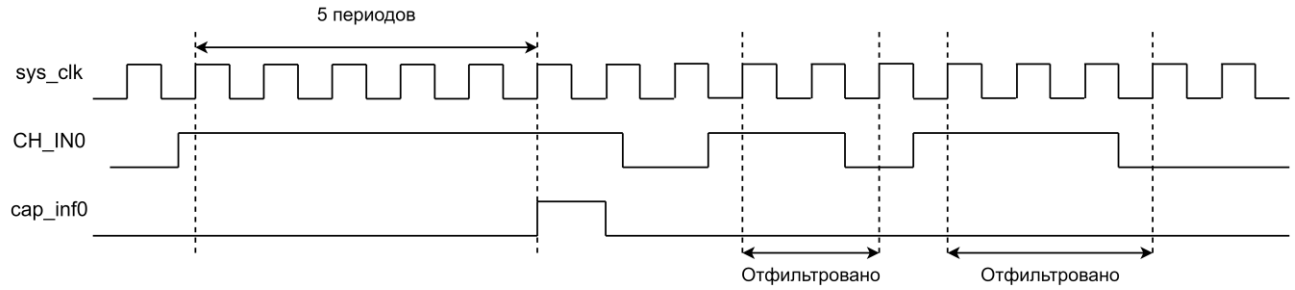


Рисунок 1.152 – Фильтрация входного сигнала по фронту

Важной особенностью регистров TIM_CH_CTRLx является то, что они могут использоваться как для настройки входного канала, так и для выходного. В связи с этим при изменении направления канала нужно следить затем, чтобы все необходимые биты были перезаписаны нужными значениями.

1.3.13.4.8 Измерение характеристик входного сигнала

Благодаря тому, что входной сигнал можно дополнительно направить на другой канал, то это дает возможность одновременно измерять и период сигнала, и его скважность. Например, это можно сделать следующим образом:

– на нулевой канал заведен измеряемый сигнал;

– на этом же канале необходимо выбрать детектирование фронта;

– на втором канале выбирается в качестве источника первый канал, и устанавливается определение изменения сигнала.

В результате таких настроек получается, что по фронту значение счетчика сохраняется в регистрах TIM_CCR1 и TIM_CCR2. Затем срез входного сигнала снова обновит TIM_CCR2. В этот момент можно уже рассчитать время, в течение которого входной сигнал находился в "1". На следующем фронте TIM_CCR1 и TIM_CCR2 снова обновятся, и по разнице между TIM_CCR1 текущим и предыдущим можно узнать период.

1.3.13.4.9 Объединение каналов

Данная функция позволяет использовать все TIM_CCRx регистры как единый буфер FIFO для захвата до четырех значений таймера. Захват происходит только по событиям от нулевого канала. События со входов остальных каналов игнорируются.

Данный режим включен когда биты TIM_CNT_CTRL[BUF_MODE] и TIM_CNT_CTRL[EN] установлены. При этом нулевой канал должен быть включен на вход, чтобы регистрировалось изменение входного сигнала.

В режиме объединения каналов все каналы, кроме нулевого, в не зависимости от настроек регистров TIM_CH_CTRL1, TIM_CH_CTRL2, TIM_CH_CTRL3 находятся в выключенном состоянии. Чтение из буфера происходит через регистр TIM_CCR0, а статусные флаги full и empty в TIM_IRQ_STATUS позволяют отслеживать заполненность буфера.

В режиме буфера флаг TIM_IRQ_STATUS[OVERCAP0_EVENT] используется для определения того, что в уже заполненный буфер записывается (захватывается) новое значение. При этом самое старое значение будет перезаписано.

Одновременное чтение из буфера и запись в него отрабатывают корректно, когда буфер непустой. Если в этот момент буфер будет пуст, то тогда произойдет считывание некорректных данных, но при этом в буфер запишется корректное значение.

1.3.13.4.10 Канал сравнения (ШИМ)

Для генерации ШИМ или других сигналов необходимо включит канал (TIM_CH_CTRLx[CH_EN] = 1) и настроить на выход, установив бит TIM_CH_CTRLx[CH_DIR] в единицу. Генерация сигнала происходит на основе сравнения счетчика с регистром канала TIM_CCRx.

Режимы работы канала сравнения описаны в таб. 1.612. В ней для используется замена обозначения регистров TIM_CNT и TIM_CCRx на CNT и CCRx. Под CCRx_nearby_channel подразумевается регистр TIM_CCRx соседнего канала. Второй и третий каналы являются соседними, так же как и первый и нулевой.

Таблица 1.612 – Режимы канала сравнения

Значения битового поля TIM_CH_CTRLx[CMP_MD]	**Описание**
0000	выход cmpX установлен все время в нуле;
0001	выход cmpX установлен все время в единице;
0010	Импульсный режим 1: выход cmpX установлен в нуле, а когда $CCRx = CNT$, то на выходе единица; [cmpX = (CCRx == CNT)]
0011	Импульсный режим 2: выход cmpX установлен в единицу, а когда $CCRx = CNT$, то на выходе нуль; [cmpX = !(CCRx == CNT)]
0110	PWM1 режим. Если счетчик считает вверх, то выход cmpX установлен в нуле пока $CCRx > CNT$, а иначе — в единице. Если счетчик считает вниз, то выход cmpX установлен в единице пока $CCRx < CNT$, а иначе — в нуле; [inc: cmpX = (CNT >= CCRx), dec: cmpX = (CNT > CCRx)]
0111	PWM2 режим. Если счетчик считает вверх, то выход cmpX установлен в единице пока $CCRx > CNT$, а иначе — в нуле. Если счетчик считает вниз, то выход cmpX установлен в нуле пока $CCRx < CNT$, а иначе — в единице; [inc: cmpX = !(CNT >= CCRx), dec: cmpX = !(CNT > CCRx)]
1110	APWM1 режим. Выход cmpX работает в соответствии с режимом PWM1. При этом пока счетчик считает вверх, то CNT сравнивается с CCRx этого же канала, а когда начинается счет вниз, CNT сравнивается с CCRx соседнего канала. Для этого соседний канал должен быть включен на выход. Если этого не сделать, то во время декрементирования на cmpX будет установлен нуль; [inc: cmpX = (CNT >= CCRx), dec: cmpX = (CNT > CCRx_nearby_channel)]
1111	APWM2 режим. Выход cmpX работает в соответствии с режимом PWM2. При этом пока счетчик считает вверх, то CNT сравнивается с CCRx этого же канала, а когда начинается счет вниз, CNT сравнивается с CCRx соседнего канала Для этого соседний канал должен быть включен на выход. Если этого не сделать, то во время декрементирования на cmpX будет установлена единица. [inc: cmpX = !(CNT >= CCRx), dec: cmpX = !(CNT > CCRx_nearby_channel)]

Канал сравнения может не выводить сигнал наружу, а использоваться только для генерации событий или отсчета интервалов времени.

1.3.13.4.11 Break функция

Данная функция позволяет перевести все выходные каналы в детерминированное break-состояние аппаратно при помощи входов tim_BREAK и tim_int_break. В регистре TIM_BREAK_CTRL настраивается полярность активного сигнала и его фильтрация.

Break функция может работать в двух режимах: остановки и приостановки. В режиме остановки после break события на выходах CH_OUTx устанавливается заданное значение. Даже после того, как на входах tim_BREAK и tim_int_break исчезнет активный сигнал, выходы

CH_OUTx продолжат удерживать заданное значение. Для того, чтобы перевести выходы в обычный режим, необходимо сбросить бит TIM_BREAK_CTRL[BRK_SET].

Во втором режиме активный сигнал на входах tim_BREAK и tim_int_break играет роль сигнала разрешения, то есть пока сигнал активный, выходы CH_OUTx находятся в break-состояние. Как только сигнал исчезает, выходы переходят в нормальный режим (см.рис. 1.154).

Для программного перевода выходных каналов в break-состояние необходимо записать 1 в поле TIM_BREAK_CTRL[BRK_SOFTWARE_SET]. При записи 0 в поле TIM_BREAK_CTRL[BRK_SOFTWARE_SET] каналы переходят в обычный режим.

При этом, пока выходы находятся в break-состоянии, счетчик продолжает считать.

Задержка на распространение аппаратного break события составляет 4 такта частоты работы таймера. (время от изменения на входе tim_int_break или tim_BREAK до выставления бита TIM_BREAK_CTRL[BRK_SET]). Программная установка или сброс break с помощью TIM_BREAK_CTRL[BRK_SOFTWARE_SET] имеет задержку в три такта (время от окончания APB транзакции до установки TIM_BREAK_CTRL[BRK_SET] в нуль). Сброс break путем записи в TIM_BREAK_CTRL[BRK_SET] нуля происходит с задержкой в два такта (время от окончания APB транзакции до сброса в нуль бита TIM_BREAK_CTRL[BRK_SET]).

Если при включении или выключении break (бит TIM_BREAK_CTRL[BRK_EN]) активный вход будет уже находится в таком состоянии, при котором должен был бы генерироваться break, то установка или сброс TIM_BREAK_CTRL[BRK_SET] произойдет с задержкой в два такта.

Переход в break (программным и аппаратным способами) имеет приоритет над сбросом. Поэтому, например, если одновременно произошел программный сброс break (запись в поле TIM_BREAK_CTRL[BRK_SET] нуля) и входы tim_int_break и TIM_BREAK поменялись таким образом, что установился BRK_SET в единицу, то break событие установится.

Структура break control представлена на рисунке 1.153.

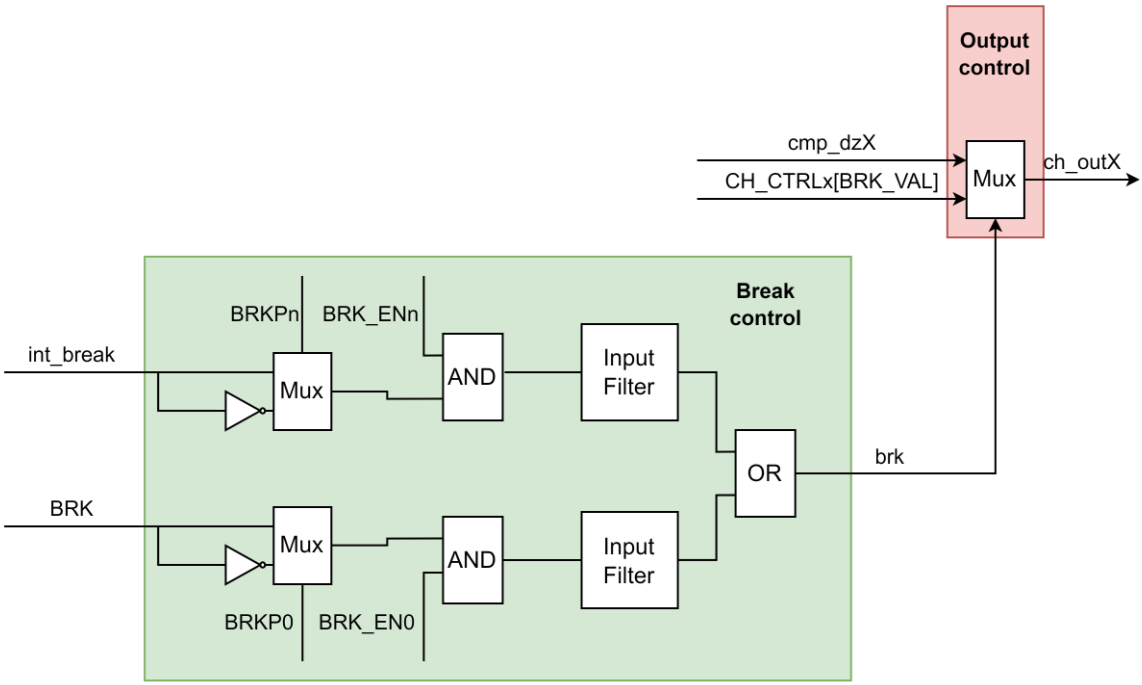


Рисунок 1.153 – Структура break control

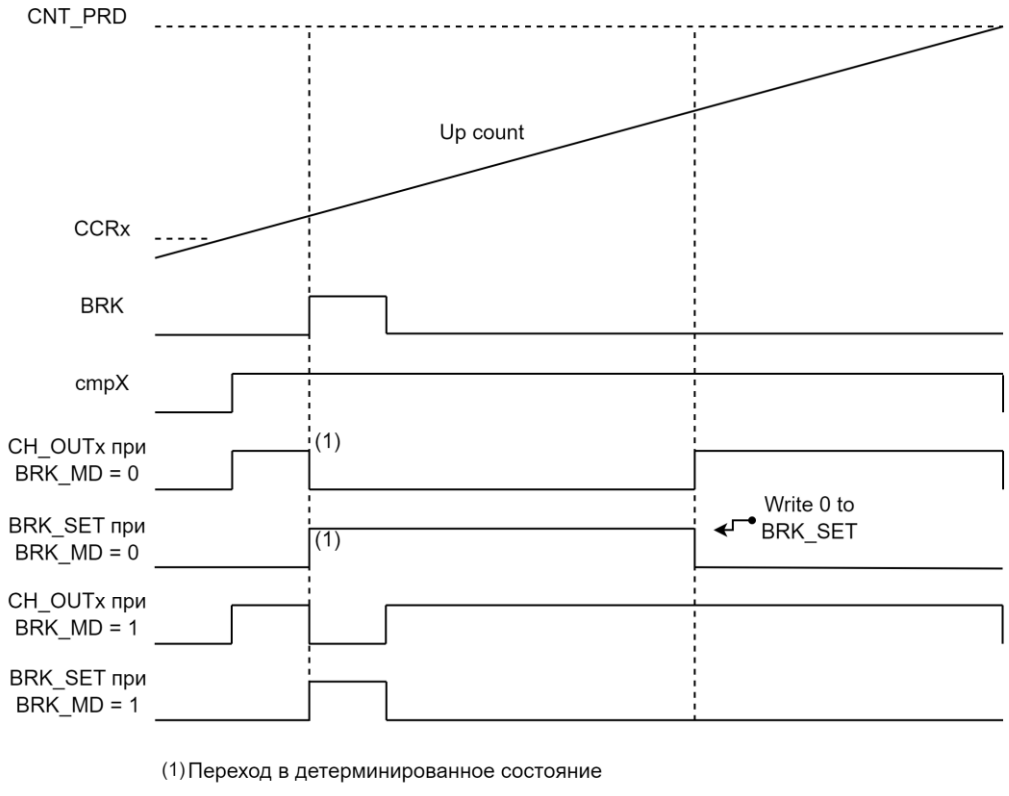


Рисунок 1.154 – Режимы работы break control

1.3.13.4.12 Dead Zone функция

Данная функция позволяет установить задержку на выходном сигнале каналов. Dead Zone можно установить либо по фронту, либо по срезу сигнала. Величина задержки устанавливается для каждого канала отдельно.

На рис. 1.155 приведен пример работы Dead Zone функции, в которой величина задержки равна двум, и два варианта работы по фронту и по срезу.

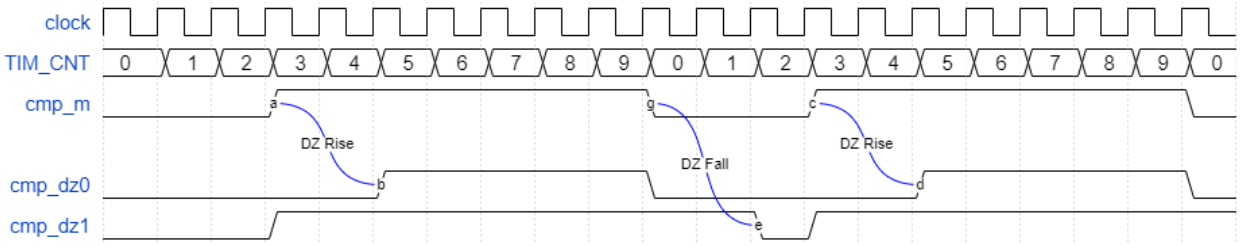


Рисунок 1.155 – Пример работы Dead Zone

Если во время задержки произойдет новое изменение сигнала cmp_m (фронт или срез), то оно будет проигнорировано. На рис. 1.156 приведен пример в котором задержка равна 2-ум тактовым сигналам и выбран режим Dead Zone по фронту cmp_mX (TIM_CH_CTRLx[CH_DZ_MD_O] = 00). На этом рисунке в точке "b" заканчивается Dead Zone от фронта сигнала cmp_m и на выходе продолжается генерироваться сам сигнал cmp_m, а он уже в этот момент равен нулю, поэтому в данном случае на выходе всегда будет нуль.

Для корректной работы с Dead Zone следует придерживаться следующих правил:

– Если Dead zone работает по фронту и величина задержки (TIM_CH_CTRLx[CH_DZ_VAL_O]) больше или равна времени в течение, которого сигнал cmp_mX находится в единице, то на выходе всегда будет нуль;

– Если Dead zone работает по срезу и величина задержки (TIM_CH_CTRLx[CH_DZ_VAL_O]) больше или равна времени в течение, которого сигнал cmp_mX находится в нуле, то на выходе всегда будет единица.

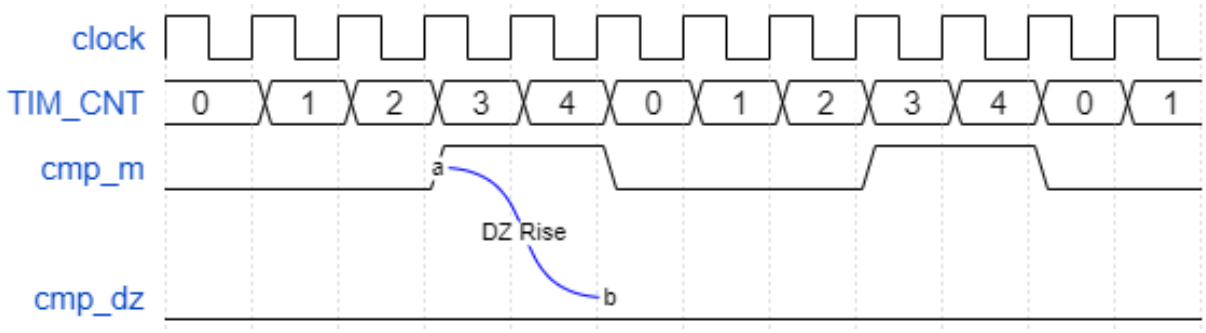


Рисунок 1.156 – Пример некорректной работы Dead Zone

1.3.13.4.13 Одиночный режим

В данном режиме счетчик отсчитывает один период и выключается. Ниже приведен пример настройки таймера в одиночном режиме, который запускается по внешнему событию.

– установить период таймера с помощью TIM_CNT_PRD.

– установить одиночный режим и выбираем режим счета. SM и CNT_MODE биты регистра CNT_CTRL отвечают за эти настройки.

– настроить каналы и выходной источник триггера.

– настроить включение таймера по внешнему событию. Для этого необходимо выбрать источник внешнего события с помощью битов TIM_EVENT_CTRL[ESEL_ACT] и в качестве действия установить "Start".

После обнаружения фронта на выбранном вход, таймер включится, просчитает один период и выключится (см.рис 1.157).

Если одиночный режим включается, когда таймер включен, счетчик досчитает до конца текущего периода и выключится.

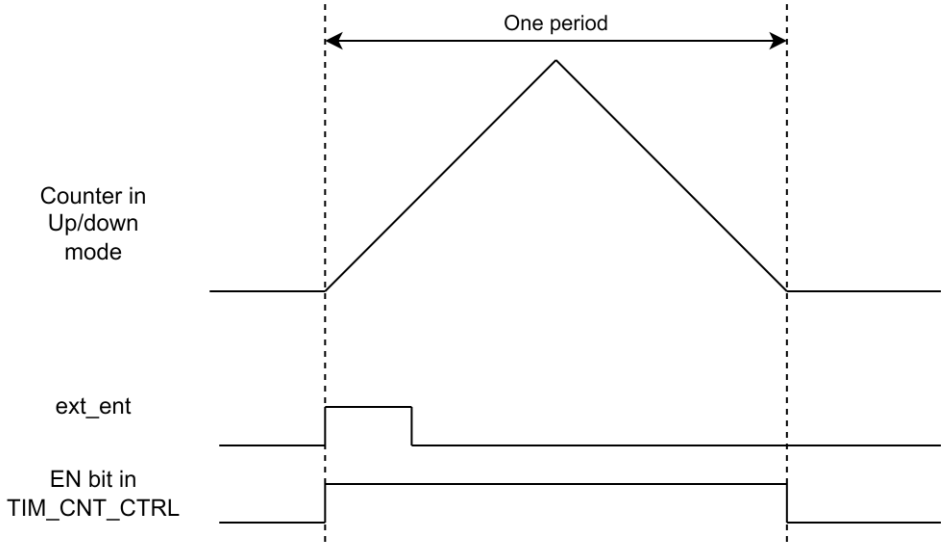


Рисунок 1.157 – Одиночный режим таймера TIM

Одиночный режим имеет приоритет над остальными. То есть если он включен, то содержимое TIM_CNT_RPT игнорируется.

1.3.13.4.14 Режим энкодера

Этот режим может быть использован для определения угла поворота и направления вращения. В данном режиме значение счетчика изменяется при изменении входных сигналов на нулевом и первом каналах (см. таб. 1.613). Up - это увеличение счетчика на единицу, Down -

уменьшение счетчика на единицу. Для включения режима энкодера используется поле TIM_CNT_CTRL[ENCODER_MODE]. При этом нулевой и первый каналы должны быть настроены на детектирование входного сигнала по обоим фронтам, т.е. TIM_CH_CTRL0[CH_ED_I] = 2'b10 и TIM_CH_CTRL1[CH_ED_I] = 2'b10, а режим счета updown (TIM_CNT_CTRL[CNT_MODE] == 2'b10).

Таблица 1.613 – Режимы энкодера блока TIM

Название режима	**ENCODER_MODE**	**Уровень сигнала на соседнем канале**	**CH_IN0**		**CH_IN1**	
			rising	falling	rising	falling
EncoderX2	01	Low	Up	Down	Nothing	Nothing
		High	Down	Up	Nothing	Nothing
EncoderX4	10	Low	Up	Down	Down	Up
		High	Down	Up	Up	Down
Encoder_direct	11	Low	Up	Up	Nothing	Nothing
		High	Down	Down	Nothing	Nothing

В режиме энкодера выбор входного сигнала между сигналами ch_inX и CH_INx, инвертирование входного сигнала и фильтрация также работают.

Если происходит ситуация одновременного инкрментирования и декрементирования, то счетчик не изменяется.

В режиме Encoder_direct сигнал на первом канале задает направление счета. Если вход CH_IN[1] находится в нуле, то счетчик инкрементируется, а если в единице, то декрементируется.

На рис. 1.158 приведен пример работы счетчика в режиме EncoderX4. На этой временной диаграмме не отображены задержки на синхронизацию, фильтрацию и определение фронтов и срезов входного сигнала.

В данном режиме бит TIM_CNT_CTRL[CNT_DIR] отображает направление последней операции. Событие UPDATE_EVENT, декрементирование счетчика повторений или выключение в одиночном режиме также происходит только при достижении нуля счетчиком (ZERO_EVENT). При достижении счётчиком заданного периода (при счёте вверх), счётчик сбрасывается в нуль. При достижении счётчиком нуля (при счёте вниз), в счётчик устанавливается значение периода.

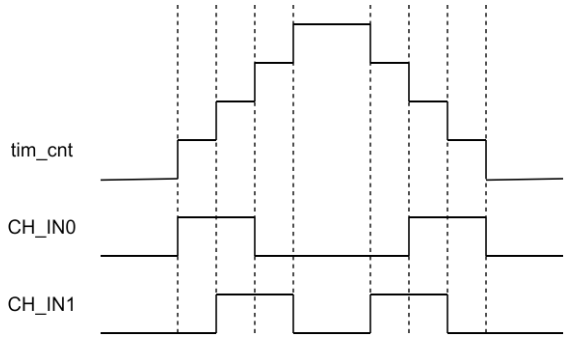


Рисунок 1.158 – Счетчик в режиме EncoderX4

1.3.13.4.15 Интеграция модуля TIM

Контроллер разделен на два тактовых домена: apb-домен и core-домен. Это сделано для того, что бы при изменении системной частоты core-домен, выполняющий передачу данных на шине, не надо было перенастраивать.

У контроллера на входе стоит специальный блок gp_apb_rejector. Его назначение заключается в корректном завершении apb-транзакций, пока с TIM контроллера снято тактирование. В зависимости от входа conf_pslverr_i транзакции могут завершаться с ошибкой или без. Если conf_pslverr_i = 1, то генерируется ошибка. При этом все транзакции записи игнорируются, а чтение возвращает 0x0BAD_ACCE5.

При обращении к TIM по шине APB младшие два бита (нулевой и первый) адреса не анализируются. APB транзакция на запись с pstrb не равным 4’b1111 возвращается с ошибкой (pslverr = 1) и никакие данные не записываются. Запись по зарезервированным адресам игнорируется, а чтение возвращает 0xEEFFADDC. При этом транзакции завершаются без ошибок. Так как младшие два бита не анализируются, то обращения по не выровненным адресам автоматически выравниваются.

1.3.13.4.16 Описание регистров

1.3.13.4.16.1 Общее описание регистров

Контроллер имеет набор программно доступных регистров, требующий 4 КБайт адресов на системной шине APB.

Поддерживается только 32-х разрядный доступ. Байтовая или двух-байтовая запись будет проигнорирована и возвращена ошибка шины.

Карта регистров приведена в таблице 1.614.

Адреса, который не указаны в таблице 1.614 считаются Зарезервированоными. Чтение из них возвращает 0xEEFFADDC, а запись игнорируется. TIM контроллер не различает secure и non-secure транзакции.

1.3.13.4.16.2 Карта регистров

Таблица 1.614 – Карта регистров TIM

Адрес	Название	Доступ	Значение после сброса	Описание
0x000	TIM_ID	ЧТ	0x3X4D4954	ID таймера TIM
0x004	TIM_VERSION	ЧТ	0	Версия таймера
0x008	TIM_SW_RST	ЗП	0	Программный сброс
0x00C	TIM_CNT_CTRL	ЧТ/ЗП	0	Включает и настраивает счетчик таймера
0x010	TIM_CNT_PRD	ЧТ/ЗП	0xFF	Задает период счетчика
0x014	TIM_CNT_PRESCALER	ЧТ/ЗП	0	Задает делитель системной частоты
0x018	TIM_CNT_RPT	ЧТ/ЗП	0	Устанавливает количество повторений периодов счетчика
0x01C	TIM_CNT	ЧТ	0	Текущее значение счетчика
0x020	TIM_CNT_PHASE	ЧТ/ЗП	0	Начальное значение для счетчика
0x024	TIM_BREAK_CTRL	ЧТ/ЗП	0	Включает и настраивает break функцию
0x028	TIM_CH_CTRL0	ЧТ/ЗП	0	Настраивает работу нулевого канала
0x02C	TIM_CH_CTRL1	ЧТ/ЗП	0	Настраивает работу первого канала
0x030	TIM_CH_CTRL2	ЧТ/ЗП	0	Настраивает работу второго канала
0x034	TIM_CH_CTRL3	ЧТ/ЗП	0	Настраивает работу третьего канала
0x038	TIM_CCR0	ЧТ/ЗП	0	Capture/Compare Reg 0
0x03C	TIM_CCR1	ЧТ/ЗП	0	Capture/Compare Reg 1
0x040	TIM_CCR2	ЧТ/ЗП	0	Capture/Compare Reg 2
0x044	TIM_CCR3	ЧТ/ЗП	0	Capture/Compare Reg 3
0x048	TIM_EVENT_CTRL	ЧТ/ЗП	0	Выбирает входное событие и настраивает работу Event control блока
0x04C	TIM_IRQ_STATUS	ЧТ	0	Статус прерываний
0x050	TIM_IRQ_MASK	ЧТ/ЗП	0	Маска прерываний
0x054	TIM_IRQ_CLEAR	ЗП	0	Сброс прерываний
0x058	TIM_IRQ_TEST	ЗП	0	Установка тестового прерывания

1.3.13.4.16.3 Описание полей регистров

1.3.13.4.16.3.1TIM_ID (0x000)

Регистр идентификатора TIM контроллера

В таблице 1.615 представлено описание полей регистра.

Таблица 1.615 – Описание полей регистра TIM_ID

Биты	Поле	Доступ	Значение после сброса	Описание
31:0	TIM_ID	ЧТ	0x3X4D4954	В данном регистре хранится ID устройства. Оно представлено в виде ASCII символов "TIMx", где x это номер TIM модуля 0 или 1 "X" в значении регистра после сброса так же определяется номером TIM'а.

1.3.13.4.16.3.2TIM_VERSION (0x004)

Регистр версии TIM контроллера

В таблице 1.616 представлено описание полей регистра.

Таблица 1.616 – Описание полей регистра TIM_VERSION

Биты	Поле	Доступ	Значение после сброса	Описание
31:0	TIM_VERSION	ЧТ	0	Хранит версию TIM.

1.3.13.4.16.3.3TIM_SW_RST (0x008)

Регистр версии TIM контроллера

В таблице 1.617 представлено описание полей регистра.

Таблица 1.617 – Описание полей регистра TIM_SW_RST

Биты	Поле	Доступ	Значение после сброса	Описание
31:0	TIM_SW_RST	ЧТ	0	Запись любого значения в данный регистр производит программный сброс.

1.3.13.4.16.3.4TIM_CNT_CTRL (0x00C)

Регистр настройки и включения TIM контроллера

В таблице 1.618 представлено описание полей регистра.

Таблица 1.618 – Описание полей регистра TIM_CNT_CTRL

Биты	Поле	Доступ	Значение после сброса	Описание
0	EN	ЧТ/ЗП	0	Включает счетчик. Включит или выключить счетчик можно как программно, путем установки этого бита в единицу, так и аппаратно по приходу события. 0 - счетчик выключен. 1 - счетчик включен.

Биты	Поле	Доступ	Значение после сброса	Описание
1	SHDWn	ЧТ/ЗП	0	Отключает теневые копии регистров. 0 - теневые регистры включены. Регистры автоматически обновляются, когда происходит событие UPDATE_EVENT (см. 1.3.13.4.5). 1 - теневые регистры выключены. Все регистры обновляются сразу после окончания записи по APB.
3:2	CNT_MODE	ЧТ/ЗП	0	Режим счетчика. 00 - счетчик инкрементируется (up). 01 - счетчик декрементируется (down). 10 - счетчик инкрементируется, а затем декрементируется (up/down). 11 - Зарезервировано.
4	SUR	ЧТ/ЗП	0	Программное обновление теневых регистров. После записи единицы в данное поле происходит обновление теневых регистров, если теневые регистры и таймер включены. После этого сбрасывается в нуль. Если установить этот бит, когда теневые регистры или таймер выключены, то бит установится на один такт, но при этом обновление теневых регистров не произойдет. 0 - После записи нуля ничего не происходит. 1 - Запись единицы обновляет теневые регистры.
5	SM	ЧТ/ЗП	0	Single mode. Включение одиночного режима. 0 - одиночный режим выключен. 1 - одиночный режим включен.
7:6	ENCODER_MODE	ЧТ/ЗП	0	Включение режима Encoder 00 - режима Encoder выключен. 01 - включен режим EncoderX2. 10 - включен режим EncoderX4. 11 - включен режим Encoder_direct.
8	PHASE_DIR	ЧТ/ЗП	0	Определяет в какую сторону будет считать счетчик, после загрузки в него PHASE значения при событии Load_phase. Изменение направления счета работает только в режиме счета updown. В режимах up и down при событии Load_phase изменение направления счета не происходит. 0 - счетчик инкрементируется. 1 - счетчик декрементируется.
9	CNT_DIR	ЧТ/ЗП	0	Текущее направление счета счетчика. Запись в данный бит игнорируется. 0 - счетчик инкрементируется. 1 - счетчик декрементируется.
10	BUF_MODE	ЧТ/ЗП	0	Режим буфера. 0 - режим буфера выключен. 1 - режим буфера включен.
31:11	-	ЧТ	0	Зарезервировано

1.3.13.4.16.3.5TIM_CNT_PRD (0x010)

Регистр периода TIM контроллера

В таблице 1.619 представлено описание полей регистра.

Таблица 1.619 – Описание полей регистра TIM_CNT_PRD

Биты	Поле	Доступ	Значение после сброса	Описание
15:0	TIM_CNT_PRD	ЧТ/ЗП	0xFF	Устанавливает максимальное значение счетчика, до которого он будет инкрементироваться или начинать с него декрементироваться. Период измеряется в тактах tim_clk с учетом делителя или во количестве импульсов от внешнего источника (ext_ent_mux).
31:11	-	ЧТ	0	Зарезервировано

1.3.13.4.16.3.6 TIM_CNT_PRESCALER (0x014)

Регистр делителя опорного тактового сигнала TIM контроллера

В таблице 1.620 представлено описание полей регистра.

Таблица 1.620 – Описание полей регистра TIM_CNT_PRESCALER

Биты	Поле	Доступ	Значение после сброса	Описание
15:0	TIM_CNT_PRESCALER	ЧТ/ЗП	0	Задет величину делителя системной частоты для счетчика. 0 - делитель выключен. 1 - системная частота делится на 2. 2 - системная частота делится на 3. 3 - системная частота делится на 4. и т.д.
31:16	-	ЧТ	0	Зарезервировано

1.3.13.4.16.3.7 TIM_CNT_RPT (0x018)

Регистр повторения периодов TIM контроллера

В таблице 1.621 представлено описание полей регистра.

Таблица 1.621 – Описание полей регистра TIM_CNT_RPT

Биты	Поле	Доступ	Значение после сброса	Описание
15:0	TIM_CNT_RPT	ЧТ/ЗП	0	Устанавливает количество повторений для счетчика в режиме repeat. После окончания очередного периода счетчика, данный регистр декрементируется. То есть он отражает текущее количество оставшихся повторений.
31:16	-	ЧТ	0	Зарезервировано

1.3.13.4.16.3.8 TIM_CNT_PHASE (0x020)

Регистр загрузки фазы TIM контроллера

В таблице 1.622 представлено описание полей регистра.

Таблица 1.622 – Описание полей регистра TIM_CNT_PHASE

Биты	Поле	Доступ	Значение после сброса	Описание
15:0	TIM_CNT_PHASE	ЧТ/ЗП	0	Величина фазы. Это то значение, которое будет записано в счетчик при программной загрузке или при действии "Load_phase". Оно должно быть меньше или равно TIM_CNT_PRD. После записи в этот регистр загрузка фаза происходит сразу.
31:16	-	ЧТ	0	Зарезервировано

1.3.13.4.16.3.9TIM_CNT (0x01C)

Регистр текущего значения счетчика TIM контроллера

В таблице 1.623 представлено описание полей регистра.

Таблица 1.623 – Описание полей регистра TIM_CNT

Биты	Поле	Доступ	Значение после сброса	Описание
15:0	TIM_CNT	ЧТ	0	Текущее значение счетчика.
31:16	-	ЧТ	0	Зарезервировано

1.3.13.4.16.3.10 TIM_BREAK_CTRL (0x024)

Регистр настройки break функции TIM контроллера

В таблице 1.624 представлено описание полей регистра.

Таблица 1.624 – Описание полей регистра TIM_BREAK_CTRL

Биты	Поле	Доступ	Значение после сброса	Описание
0	BRK_EN	ЧТ/ЗП	0	0 - Break функция выключена. 1 - Break функция включена.
1	BRK_MD	ЧТ/ЗП	0	Выбор режима break функции. 0 - все выходы переходят в детерминированное значение, указанное в поле регистров TIM_CH_CTRLx[CH_BRK_ACT_O], пока не будет сброшен бит BRK_SET. 1 - все выходы также переходят в детерминированные значения, но только пока на входах tim_BREAK или tim_int_break установлен активный уровень.
2	BRK_SET	ЧТ/ЗП	0	0 - выходы каналов находятся в обычном режиме, управляются TIM_CCRx регистрами. 1 - выходы каналов находятся в break режиме. На выходах установлено значение из TIM_CH_CTRLx[CH_BRK_ACT_O].
3	BRK_SOFTWARE_SET	ЧТ/ЗП	0	Установка break режима программно. 0 - выходы не находятся в break режиме. 1 - выходы находятся в break режиме.

Биты	Поле	Доступ	Значение после сброса	Описание
4	BRKP0	ЧТ/ЗП	0	Выбирается активный уровень входящего сигнала. 0 - нуль на входе является активным. 1 - единица на входе является активной.
5	BRK_EN0	ЧТ/ЗП	0	0 - вход tim_BREAK выключен. 1 - вход tim_BREAK включен.
6	BRKP1	ЧТ/ЗП	0	См. описание бита BRKP0.
7	BRK_EN1	ЧТ/ЗП	0	0 - вход tim_int_break выключен. 1 - вход tim_int_break включен
31:8	-	ЧТ	0	Зарезервировано

1.3.13.4.16.3.11 TIM_CH_CTRLx (0x028 или 0x02C или 0x030 или 0x034)

Регистр настройки входного канала TIM контроллера

В таблице 1.625 представлено описание полей регистра.

Таблица 1.625 – Описание полей регистра TIM_CH_CTRLx

Биты	Поле	Доступ	Значение после сброса	Описание
0	CH_EN	ЧТ/ЗП	0	Включение канала. Когда канал выключен, его вход и выход находятся в высокоимпедансном состоянии. 0 - канал выключен. 1 - канал включен.
1	CH_DIR	ЧТ/ЗП	0	Выбор направления канала (захват или сравнение). 0 - канал находится в режиме входа, захвата. 1 - канал находится в режиме выхода, сравнения.
3:2	-	ЧТ	0	Зарезервировано
6:4	CH_SEL_I	ЧТ/ЗП	0	Выбор входного сигнала канала cap_inX. 000 - tim_CH_INx внешний вход. 001 - tim_ch_inX. 010 - Зарезервировано. 011 - Зарезервировано. 100 - Зарезервировано. 101 - Зарезервировано. 111 - Зарезервировано.
7	CH_MUX_I	ЧТ/ЗП	0	Выбор сигнала capX между прямым tim_ch_inX и cap_in с соседнего канала. 0 - на входе выбирается tim_ch_inX[n]. 1 - на входе выбирается cap_in с соседнего канала.
9:8	CH_ED_I	ЧТ/ЗП	0	Детектирование входного сигнала. 00 - захват происходит по фронту. 01 - захват происходит по срезу. 10 - захват происходит по изменению сигнала. 11 - Зарезервировано.
10	CH_INV_I	ЧТ/ЗП	0	Инвертирование входного сигнала канала capN (см. рис. 1.146). 0 - инвертирование выключено. 1 - инвертирование включено.
11	-	ЧТ	0	Зарезервировано

Биты	Поле	Доступ	Значение после сброса	Описание
15:12	CH_FLT_I	ЧТ/ЗП	0	Настройка длительности фильтрации входных сигналов. Единицей фильтрации является период системной частоты Tsys_clk. Данное поле указывает какое количество периодов должен удерживаться входной сигнал в активном состоянии, чтобы вызвать захват счетчика в регистр захвата данного канала. 0000 - импульс длиной в 1 Tsys_clk. 0001 - импульс длиной в 2 Tsys_clk. 0010 - импульс длиной в 3 Tsys_clk. и т.д.
31:16	-	ЧТ	0	Зарезервировано

1.3.13.4.16.3.12 TIM_CH_CTRLx (0x028 или 0x02C или 0x030 или 0x034)

Регистр настройки выходного канала TIM контроллера

В таблице 1.626 представлено описание полей регистра.

Таблица 1.626 – Описание полей регистра TIM_CH_CTRLx

Биты	Поле	Доступ	Значение после сброса	Описание
0	CH_EN	ЧТ/ЗП	0	Включение канала. Когда канал выключен, его вход и выход находятся в высокоимпедансном состоянии. 0 - канал выключен. 1 - канал включен.
1	CH_DIR	ЧТ/ЗП	0	Выбор направления канала (захват или сравнение). 0 - канал находится в режиме входа, захвата. 1 - канал находится в режиме выхода, сравнения.
3:2	-	ЧТ	0	Зарезервировано

Биты	Поле	Доступ	Значение после сброса	Описание
7:4	CH_CMP_MD_O	ЧТ/ЗП	0	Выбор условия по которому будет генерироваться сигнал cmp_mX 0000 - выход cmpX установлен все время в нуле; 0001 - выход cmpX установлен все время в единице; 0010 - Импульсный режим 1: выход cmpX установлен в нуле, а когда $CCRx = CNT$, то на выходе единица; [cmpX = (CCRx == CNT)] 0011 - Импульсный режим 2: выход cmpX установлен в единицу, а когда $CCRx = CNT$, то на выходе нуль; [cmpX = !(CCRx == CNT)] 0110 - PWM1 режим. Если счетчик считает вверх, то выход cmpX установлен в нуле пока $CCRx > CNT$, а иначе — в единице. Если счетчик считает вниз, то выход cmpX установлен в единице пока $CCRx < CNT$, а иначе — в нуле; [inc: cmpX = (CNT >= CCRx), dec: cmpX = (CNT > CCRx)] 0111 - PWM2 режим. Если счетчик считает вверх, то выход cmpX установлен в единице пока $CCRx > CNT$, а иначе — в нуле. Если счетчик считает вниз, то выход cmpX установлен в нуле пока $CCRx < CNT$, а иначе — в единице; [inc: cmpX = !(CNT >= CCRx), dec: cmpX = !(CNT > CCRx)] 1110 - APWM1 режим. Выход cmpX работает в соответствии с режимом PWM1. При этом пока счетчик считает вверх, то CNT сравнивается с CCRx этого же канала, а когда начинается счет вниз, CNT сравнивается с CCRx соседнего канала. Для этого соседний канал должен быть включен на выход. Если этого не сделать, то во время декрементирования на cmpX будет установлен нуль; [inc: cmpX = (CNT >= CCRx), dec: cmpX = (CNT > CCRx_nearby_channel)] 1111 - APWM2 режим. Выход cmpX работает в соответствии с режимом PWM2. При этом пока счетчик считает вверх, то CNT сравнивается с CCRx этого же канала, а когда начинается счет вниз, CNT сравнивается с CCRx соседнего канала. Для этого соседний канал должен быть включен на выход. Если этого не сделать, то во время декрементирования на cmpX будет установлена единица.+ [inc: cmpX = !(CNT >= CCRx), dec: cmpX = !(CNT > CCRx_nearby_channel)]
15:8	CH_DZ_VAL_O	ЧТ/ЗП	0	Величина задержки для Dead Zone. Она измеряется в периодах тактового сигнала sys_clk.

Биты	Поле	Доступ	Значение после сброса	Описание
17:16	CH_DZ_MD_O	ЧТ/ЗП	0	Настройка режима Dead Zone. 00 - DZ режим работает по фронту. 01 - DZ режим работает по спаду. 10 - Зарезервировано. 11 - Зарезервировано.
19:18	-	ЧТ	0	Зарезервировано
21:20	CH_BRK_ACT_O	ЧТ/ЗП	0	При поступлении сигнала brk выход данного канала переходит в одно из следующих состояний: 00 - высокоимпедансное состояние. 01 - активный низкий уровень. 11 - активный высокий уровень. 10 - Зарезервировано.
31:22	-	ЧТ	0	Зарезервировано

1.3.13.4.16.3.13 TIM_CCRx (0x038 или 0x03C или 0x040 или 0x044)

Регистр захвата и сравнения TIM контроллера

В таблице 1.627 представлено описание полей регистра.

Таблица 1.627 – Описание полей регистра TIM_CCRx

Биты	Поле	Доступ	Значение после сброса	Описание
15:0	TIM_CCRx	ЧТ/ЗП	0	Если канал настроен в режиме захвата, то данный регистр содержит значение счетчика, когда произошло это событие. Это значение будет сохраняться пока не произойдет новых захват. Если канал настроен в режиме сравнения, то в данный регистр можно писать и считывать записанное значение. Для подробного описания доступности регистра см. таблицу 1.609.
31:16	-	ЧТ	0	Зарезервировано

1.3.13.4.16.3.14 TIM_EVENT_CTRL (0x048)

Регистр настройки событий TIM контроллера

В таблице 1.628 представлено описание полей регистра.

Таблица 1.628 – Описание полей регистра TIM_EVENT_CTRL

Биты	Поле	Доступ	Значение после сброса	Описание
1:0	INT_EVENT_SEL	ЧТ/ЗП	0	Internal event select. Задает источник внутренних событий. 00 - tim_int_events. 01 - cap2_inf. 10 - cap3_inf. 11 - Зарезервировано.

Биты	Поле	До-ступ	Значение по-сле сброса	Описание
3:2	-	ЧТ	0	Зарезервировано
6:4	ISEL_ACT	ЧТ/ЗП	0	Internal select action. Выбор действия, которое будет происходить с таймером при приходе события от другого таймера. 000 - None. 001 - Start. 010 - Trigger 011 - EN_reset. 100 - Load_phase.
7	-	ЧТ	0	Зарезервировано
9:8	INT_EVENT_DIR	ЧТ/ЗП	0	Internal event detection 00 - фронт является активатором события. 01 - срез является активатором события. 10 - высокий уровень является активатором события. 11 - низкий уровень является активатором события.
11:10	-	ЧТ	0	Зарезервировано
14:12	EXT_EVENT_SEL	ЧТ/ЗП	0	External event select. Выбирается какое из внешних устройств будет служить источником событий (см. таблицу 1.610). 000 - вход tim_ENT. 001 - вход tim_ext_events_i. и т.д. Зарезервировано
15	-	ЧТ	0	Зарезервировано
18:16	ESEL_ACT	ЧТ/ЗП	0	External select action. Выбор действия, которое будет происходить с таймером при приходе внешнего события. 000 - None. 001 - Start. 010 - Trigger. 011 - EN_reset. 100 - Load_phase.
19	-	ЧТ	0	Зарезервировано
21:20	EXT_EVENT_DIR	ЧТ/ЗП	0	External Event detector 00 - фронт является активатором события. 01 - срез является активатором события. 10 - высокий уровень является активатором события. 11 - низкий уровень является активатором события.
23:22	-	ЧТ	0	Зарезервировано

Биты	Поле	Доступ	Значение после сброса	Описание
26:24	EE_PSC	ЧТ/ЗП	0	External event prescaler. Внешние события должны иметь частоту меньше, чем 1/2 sys_clk. Для этого и используется данный делитель. 000 - делитель выключен. 001 - частота сигнала внешних событий ext_ent_mux делится на 2. 010 - частота сигнала внешних событий ext_ent_mux делится на 4. 011 - частота сигнала внешних событий ext_ent_mux делится на 8. 100 - частота сигнала внешних событий ext_ent_mux делится на 16. 101 - частота сигнала внешних событий ext_ent_mux делится на 32.
27	-	ЧТ	0	Зарезервировано
31:28	TRIG_SEL	ЧТ/ЗП	0	Trigger_out select. Выбор события которое будет генерировать выходной сигнал триггера trigger_out. 0000 - генерация выходного триггера выключена. 0001 - zero event. 0010 - period event. 0011 - update event. 0100 - enable status. Данное событие всегда активно пока таймер включен. Это позволяет включать несколько таймеров одновременно. 0101 - Зарезервировано. 0110 - Зарезервировано. 0111 - Зарезервировано. 1000 - compare event на канале 0. 1001 - compare event на канале 1. 1010 - compare event на канале 2. 1011 - compare event на канале 3. 1100 - capture event на канале 0. 1101 - capture event на канале 1. 1110 - capture event на канале 2. 1111 - capture event на канале 3.

1.3.13.4.16.3.15 TIM_IRQ_STATUS (0x04C)

Регистр статуса прерываний TIM контроллера

В таблице 1.629 представлено описание полей регистра.

Таблица 1.629 – Описание полей регистра TIM_IRQ_STATUS

Биты	Поле	Доступ	Значение после сброса	Описание
0	TEST	ЧТ	0	Тестовое прерывание. Устанавливается только программно через регистр TIM_IRQ_TEST

Биты	Поле	Доступ	Значение после сброса	Описание
1	CAP0_EVENT	ЧТ	0	Capture event on channel 0. Статусный бит устанавливается, когда на входном канале происходит захват, то есть после фильтрации и обнаружения события (фронт, срез и др.).
2	CAP1_EVENT	ЧТ	0	Capture event on channel 1. Аналогично биту CAP0_EVENT.
3	CAP2_EVENT	ЧТ	0	Capture event on channel 2. Аналогично биту CAP0_EVENT.
4	CAP3_EVENT	ЧТ	0	Capture event on channel 3. Аналогично биту CAP0_EVENT.
5	CMP0_EVENT	ЧТ	0	Compare event on channel 0. Статусный бит устанавливается, когда на выходном канале значение сравнения TIM_CCR0 равняется текущему значения счетчика TIM_CNT.
6	CMP1_EVENT	ЧТ	0	Compare event on channel 1. Аналогично биту CMP0_EVENT.
7	CMP2_EVENT	ЧТ	0	Compare event on channel 2. Аналогично биту CMP0_EVENT.
8	CMP3_EVENT	ЧТ	0	Compare event on channel 3. Аналогично биту CMP0_EVENT.
9	ZERO_EVENT	ЧТ	0	Статусный бит устанавливается, когда счетчик достигает нуля и при этом включен CNT_CTRL[EN]=1, а направление счета - вниз.
10	PRD_EVENT	ЧТ	0	Статусный бит устанавливается, когда счетчик достигает значения периода (CNT_PRD) и при этом включен CNT_CTRL[EN]=1, а направление счета - вверх.
11	UPDATE_EVENT	ЧТ	0	Данный бит выставляется после того, как произошло обновление теневых регистров.
12	CNT_TICK_EVENT	ЧТ	0	Когда значение счетчика изменяется, то данный бит устанавливается.
13	BREAK_EVENT	ЧТ	0	После того как происходить событие BREAK_EVENT, устанавливается данный бит (см. таблицу 1.3.13.4.5).
14	INTERVAL_END_EVENT	ЧТ	0	Данный бит устанавливается после того, как заканчивается период в одиночном режиме или оканчиваются все повторений в repeat режиме.
15	INT_EVENT	ЧТ	0	Данный бит устанавливается после того, как приходит событие от другого таймера.
16	EXT_EVENT	ЧТ	0	Данный бит устанавливается после того, как приходит внешнее событие.
17	OVERCAP0_EVENT	ЧТ	0	Если произошло два события захвата на нулевом канале, а между ними не было операции чтения из регистра TIM_CCR0, то данный бит установится в единицу. В режиме буфера данный бит устанавливается, когда буфер заполнен и приходит новое событие захвата. Остальные OVERCAPx_EVENT имеют некорректное значение.
18	OVERCAP1_EVENT	ЧТ	0	См. описание бита OVERCAP0_EVENT.

Биты	Поле	Доступ	Значение после сброса	Описание
19	OVERCAP2_EVENT	ЧТ	0	См. описание бита OVERCAP0_EVENT.
20	OVERCAP3_EVENT	ЧТ	0	См. описание бита OVERCAP0_EVENT.
21	EMPTY	ЧТ	0	Статусный бит пустоты буфера в режиме объединения каналов. 0 - буфер пуст. 1 - в буфере есть захваченное(ые) значения счетчика.
22	FULL	ЧТ	0	Статусный бит полноты буфера в режиме объединения каналов. 0 - в буфере еще есть место. 1 - буфер полон.
31:23	-	ЧТ	0	Зарезервировано

1.3.13.4.16.3.16 TIM_IRQ_MASK (0x050)

Регистр маски прерываний TIM контроллера

В таблице 1.630 представлено описание полей регистра.

Таблица 1.630 – Описание полей регистра TIM_IRQ_MASK

Биты	Поле	Доступ	Значение после сброса	Описание
31:0	TIM_IRQ_MASK	ЧТ/ЗП	0	Данный регистр имеет битовые поля аналогичные регистру TIM_IRQ_STATUS. Маскируются не сами биты регистра TIM_IRQ_STATUS, а выходной сигнал tim_irq_o: tim_irq_o = | (TIM_IRQ_STATUS & TIM_IRQ_MASK) Если в соответствующем бите регистра TIM_IRQ_MASK установлена единица, то данное прерывание включено. Если нуль, то выключено. После сброса все прерывания запрещены.

1.3.13.4.16.3.17 TIM_IRQ_CLEAR (0x054)

Регистр сброса прерываний TIM контроллера

В таблице 1.631 представлено описание полей регистра.

Таблица 1.631 – Описание полей регистра TIM_IRQ_CLEAR

Биты	Поле	Доступ	Значение после сброса	Описание
31:0	TIM_IRQ_CLEAR	ЧТ	0	Данный регистр имеет битовые поля аналогичные регистру TIM_IRQ_STATUS. Запись единицы в соответствующие биты сбрасывает соответствующие прерывания.

1.3.13.4.16.3.18 TIM_IRQ_TEST (0x058)

Регистр установки тестового прерываний TIM контроллера

В таблице 1.632 представлено описание полей регистра.

Таблица 1.632 – Описание полей регистра TIM_IRQ_TEST

Биты	Поле	Доступ	Значение после сброса	Описание
0	TIM_IRQ_TEST	ЗП	0	Запись единицы в данный бит устанавливает тестовое прерывание TIM_IRQ_STATUS[TEST], а сброс происходит через TIM_IRQ_CLEAR.
31:1	-	ЧТ	0	Зарезервировано

1.3.13.5 Блок сторожевого таймера (WDT)

1.3.13.5.1 Основные характеристики сторожевого таймера (WDT)

Сторожевой таймер WDT представляет собой 16-битный убывающий счетчик, работающий на тактовой частоте независимого источника тактирования. Он предназначен для перезагрузки системы в случае, если от нее нет ответа. Ответом системы является периодическая перезагрузка счетчика, которая вновь устанавливает начальное значение отсчета. Перезагрузка происходит путем записи в определенный регистр сторожевого таймера WDT.

Сторожевой таймер WDT обладает следующими особенностями:

– выработка запроса на прерывание или запроса на сброс системы при достижении счетчиком значения 0x0 (в зависимости от настроек);

– два режима перезагрузки счетчика:

– режим с временным окном, в пределах которого разрешена перезагрузка счетчика таймера;

– режим без временного окна (перезагрузка счетчика таймера разрешена всегда);

– работа таймера:

– разрешение работы таймера устанавливается при аппаратном или программном сбросе;

– возможность включения/отключения таймера при установленном разрешении его работы;

– возможность приостановки счетчика таймера внешним сигналом или записью в регистр;

– установка настроек таймера:

– возможность изменения настроек во время работы таймера;

– при приостановке счетчика и последующем возобновлении новые настройки не применяются;

– применение новых настроек возможно при включении таймера или при перезагрузке счетчика таймера;

– настройки таймера:

– доступны предделители 4, 8, 16, 32, 64, 128, 512, 1024;

– установка начального значения отсчета счетчика;

– возможность выработки раннего запроса на прерывание и установка значения счетчика, по которому это происходит;

– настройка таймера для режимов работы с временным окном и без него;

– установка значений верхней и нижней границ временного окна;

– настройка, позволяющая выбрать выработку запроса на прерывание или запроса на сброс системы при достижении счетчиком значения 0x0;

– генерируемые таймером выходные сигналы:

– запрос на сброса системы при достижении счетчиком значения 0x0;

– запросы на прерывание:

– запросы на прерывание при достижении счетчиком значения 0x0;

– ранний запрос на прерывание при достижении установленного в настройках значения счетчика;

– запрос на прерывание при попытке перезагрузки счетчика, когда значение счетчика больше значения верхней границей временного окна;

– запрос на прерывание при попытке перезагрузки счетчика, когда значение счетчика меньше значения нижней границы временного окна;

– события:

– запросы на прерывание при достижении счетчиком значения 0x0;

– ранний запрос на прерывание при достижении установленного в настройках значения счетчика;

– возможность настройки маски для обработки генерируемых прерываний;

– возможность запрета изменения настроек таймера (настраивается через регистр или при сбросе);

– интерфейс программирования APB 5 slave.

1.3.13.5.2 Интеграция сторожевого таймера WDT

Описание входов и выходов сторожевого таймера WDT приведено в 1.633.

Таблица 1.633 – Описание входов и выходов сторожевого таймера WDT

Имя порта	Размер	Направление	Описание
Системные сигналы			
pclk_i	1	вх	Тактовый сигнал для интерфейса APB
clk_wdt_i	1	вх	Тактовый сигнал для счетчика WDT
rst_n_i	1	вх	Инверсный аппаратный сброс
apb_enable_i	1	вх	Разрешение на работу тактового сигнала для интерфейса APB: "1" - разрешено; "0" - запрещено
wdt_enable_i	1	вх	Разрешение на работу тактового сигнала для счетчика WDT: "1" - разрешено; "0" - запрещено
conf_pslverr_i	1	вх	Значение сигнала APB pslverr при отключенном разрешении на работу тактового сигнала для интерфейса APB: "1" - pslverr установлен; "0" - pslverr отсутствует
Интерфейс APB 5 slave			
paddr_i	32	вх	Шина адреса
pprot_i	3	вх	Шина сигналов для защищенного доступа
psel_i	1	вх	Выбор ведомого устройства
penable_i	1	вх	Сигнал корректности данных
pwrite_i	1	вх	Выбор операции записи/чтения
pwdata_i	32	вх	Шина данных для записи
pstrb_i	4	вх	Стробы для операции побайтовой записи
pready_o	1	вых	Сигнал готовности к транзакции
prdata_o	32	вых	Шина считанных данных
pslverr_o	1	вых	Ошибка ведомого устройства
Интерфейс WDT			
wdt_enable_ext_i	1	вх	Включение сторожевого таймера сразу после аппаратного сброса (сигнал от внешнего блока): "1" - разрешено; "0" - запрещено
wdt_lock_ext_i	1	вх	Изменение настроек сторожевого таймера (сигнал от внешнего блока): "1" - запрещено; "0" - разрешено
wdt_suspend_ext_i	1	вх	Приостановка счетчика таймера (сигнал от внешнего блока): "1" - приостановка счетчика; "0" - нормальная работа счетчика
wdt_event_strobe_o	1	вых	Событие достижения значения для выработки раннего запроса на прерывание (биты регистра WDT_SETTINGS[31:16]) или значения для выработки запроса на прерывание (регистр WDT_VALUE = 0x0) Строб с уровнем "1" - произошло одно из указанных событий

wdt_event_o	1	вых	Событие достижения значения для выработки раннего запроса на прерывание (биты регистра WDT_SETTINGS[31:16]) или значения для выработки запроса на прерывание (регистр WDT_VALUE = 0x0) Изменение уровня с "0" на "1" либо с "1" на "0" - произошло одно из указанных событий
wdt_irq_o	1	вых	Сигнал запроса на прерывание: "1" - появление запроса на прерывание; "0" - отсутствие запроса на прерывание
wdt_reset_o	1	вых	Сигнал запроса на сброс системы "1" - появление запроса на сброс системы; "0" - отсутствие запроса на сброс системы

1.3.13.5.3 Описание работы сторожевого таймера

Сторожевой таймер WDT представляет собой 16-битный убывающий счетчик, работающий на тактовой частоте независимого источника тактирования. Он предназначен для перезагрузки системы в случае, если от нее нет ответа. Ответом системы является периодический перезапуск счетчика, который вновь устанавливает начальное значение отсчета. Перезапуск счетчика происходит путем записи в определенный регистр сторожевого таймера WDT.

Если таймер запущен, то он работает непрерывно до момента, пока счетчик не переполнится. В этом случае, в зависимости от настроек, вырабатывается запрос на прерывание или запрос сброс, который перезагружает всю систему.

Сторожевой таймер обладает настройками режима временного окна, начального значения отсчета и выбора предделителя.

В режиме работы с временным окном возможно установить верхнюю и нижнюю временную границу, в пределах которых разрешен перезапуск счетчика. Это позволяет лучше контролировать ход выполнения программы за счет установления временных рамок, в пределах которых должен быть отклик системы. Также присутствует возможность работать в режиме работы без временного окна, что позволяет перезапустить счетчик в любой момент.

Установка начального значения отсчета и выбор одного из доступных предделителей позволяет гибко настраивать период, за который сторожевой таймер должен успеть перезапуститься.

Присутствует возможность приостановки счетчика таймера, что позволяет в особых ситуациях не ожидать ответа от системы. Дальнейший запуск сторожевого таймера продолжает отсчет с момента, где счетчик был остановлен.

Возможно настроить маску выработки запросов на прерывание сторожевого таймера, которые будут обрабатываться системой.

1.3.13.5.3.1 Структурная схема сторожевого таймера

Особенностью сторожевого таймера является использование двух источников тактового сигнала, за счет чего он разделен на два функциональных блока.

Первый блок - блок управления, который обрабатывает сигналы интерфейса APB 5 Slave, хранит настройки сторожевого таймера и управляет его работой.

Второй блок - блок счета, который применяет загруженные ранее настройки и уменьшает счетчик таймера. За счет того, что блок счета работает на частоте независимого источника тактирования, сторожевой таймер работает и в спящем режиме системы.

Вследствие перехода из одного частотного домена в другой в сторожевом таймере присутствует прохождение внутренней синхронизации сигналов, которая занимает 3 такта наименьшей из используемых частот тактового сигнала. Эту задержку необходимо учитывать при работе со сторожевым таймером.

Во время прохождения внутренней синхронизации сигналов устанавливается защита от записи в некоторые регистры. Защита от записи убирается после прохождения внутренней синхронизации сигналов.

Чтение из регистров доступно в любой момент времени.

Структурная схема сторожевого таймера приведена на рисунке 1.159.

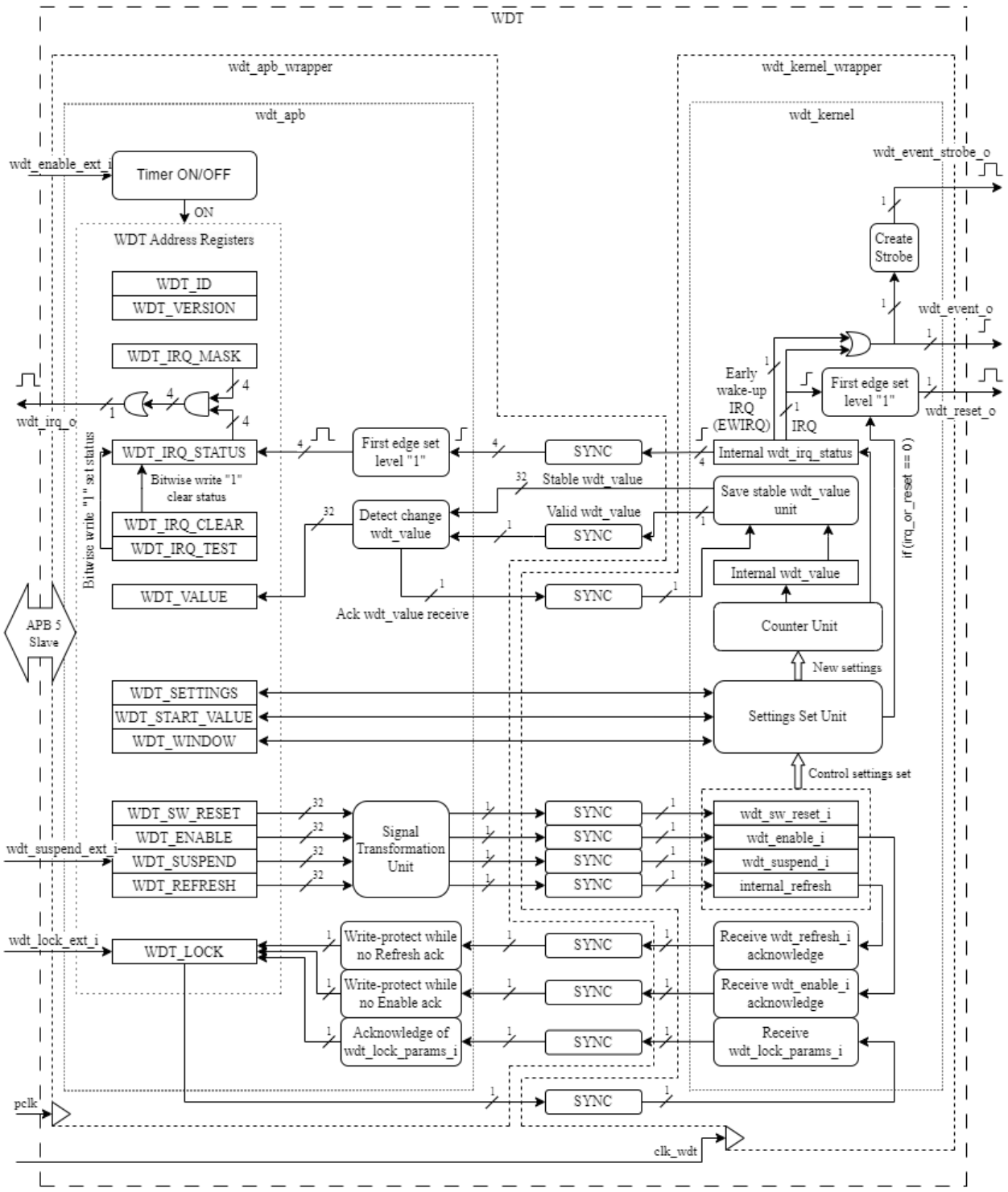


Рисунок 1.159 – Структурная схема сторожевого таймера

1.3.13.5.3.2 Разрешение на работу сторожевого таймера

Разрешение на работу сторожевого таймера устанавливается в момент включения системы открывает доступ к его регистрам. В случае, когда не установлено разрешение на работу сторожевого таймера, доступ к его регистрам закрыт.

Доступ к регистрам настроек сторожевого таймера может быть ограничен при включении системы, что соответствует "1" в бите регистра WDT_LOCK[0]. В этом случае для пользователя доступны для записи только регистры WDT_ENABLE, WDT_SUSPEND, WDT_REFRESH. Для чтения остаются доступны все регистры.

1.3.13.5.3.3 Особенности сигналов APB Slave при работе со сторожевым таймером

Интерфейсом взаимодействия со сторожевого таймера является APB 5 Slave.

Во время аппаратного сброса:

– запись в адресные регистры игнорируется;

– значение prdata_o установлено в 0x0;

– сигнал pready_o установлен в "1";

– сигнал pslverr_o установлен в "1".

Во время программного сброса:

– запись в адресные регистры игнорируется;

– значение prdata_o:

– при чтении из регистра WDT_SW_RESET prdata_o == 0x1;

– при чтении из других регистров prdata_o == 0x0;

– сигнал pready_o установлен в "1";

– сигнал pslverr_o установлен в "0".

Во время обычной работы сторожевого таймера:

– запись в адресные регистры возможна;

– значение prdata_o:

– при чтении из адресного регистра принимает значение, которое хранится по выбранному адресу;

– при чтении по ошибочному адресу возвращает 0xE4E4E4E4;

– сигнал pready_o установлен в "1";

– сигнал pslverr_o:

– устанавливается в "1" в случае, если происходит запись, отличная от 4-х байтного слова (pstrb_i != 0xF);

– сохраняет значение "0" в остальных случаях.

1.3.13.5.3.4 Особенности сигналов интерфейса WDT при работе со сторожевым таймером

Входными сигналами интерфейса WDT являются 3 сигнала: wdt_enable_ext_i, wdt_lock_ext_i и wdt_suspend_ext_i. Входные сигналы устанавливаются внешними блоками.

Сигнал wdt_enable_ext_i служит для запуска сторожевого таймера сразу после аппаратного сброса. Значение сигнала важно в момент сразу после аппаратного сброса.

При значении wdt_enable_ext_i == 1 сторожевой таймер сразу запустится после аппаратного сброса. При этом установится бит WDT_LOCK[1] == 0x1 на время прохождения внутренней синхронизации сигналов на время [2 * (3 такта pclk_i + 3 такта clk_wdt_i)]. Регистры WDT_SETTINGS, WDT_START_VALUE, WDT_WINDOW, WDT_REFRESH заблокированы для записи на время прохождения внутренней синхронизации сигналов. После прохождения внутренней синхронизации сигналов, бит WDT_LOCK[1] == 0x0 и запись в регистры становится снова возможной.

Значение wdt_enable_ext_i == 0 не запускает сторожевой таймер сразу после аппаратного сброса.

Вне зависимости от сигнала wdt_enable_ext_i, запуск и остановка сторожевого таймера возможно через регистр WDT_ENABLE.

Изменение сигнала wdt_enable_ext_i во время работы сторожевого таймера не влияет на его работу.

Сигнал wdt_lock_ext_i служит для запрета на изменение настроек таймера. Значение сигнала важно в момент сразу после аппаратного сброса.

При значении wdt_lock_ext_i == 1 устанавливается запрет на изменение регистров WDT_SETTINGS, WDT_START_VALUE, WDT_WINDOW. В этом случае устанавливается бит WDT_LOCK[0] = 0x1. Для пользователя остаются доступны для записи только регистры WDT_ENABLE, WDT_SUSPEND, WDT_REFRESH.

Значение wdt_lock_ext_i == 0 не запрещает изменение каких-либо регистров.

Вне зависимости от сигнала wdt_lock_ext_i, все регистры доступны для чтения.

Изменение сигнала wdt_lock_ext_i во время работы сторожевого таймера не влияет на его работу.

Сигнал wdt_suspend_ext_i служит для приостановки счетчика сторожевого таймера. Значение сигнала важно в любой момент работы сторожевого таймера.

При значении wdt_suspend_ext_i == 1 счетчик сторожевого таймера будет сохранять свое значение - таймер остановит отсчет. При wdt_suspend_ext_i == 1 любая запись в регистр WDT_SUSPEND возможна, но будет игнорироваться. В остальном, поведение сторожевого таймера эквивалентно взаимодействию с регистром WDT_SUSPEND.

В случае программного сброса при установленном сигнале wdt_suspend_ext_i == 1 регистры сторожевого таймера установятся в значение по сбросу, при этом при дальнейший запуске таймера будут применены новые параметры, но таймер не начнет отсчет.

Значение wdt_suspend_ext_i == 0 выводит счетчика сторожевого таймера из режима приостановки.

Задержка на приостановку таймера при помощи сигнала wdt_suspend_ext_i составляет [2 * (3 такта pclk_i + 3 такта clk_wdt_i)].

Выходными сигналами интерфейса WDT являются 3 сигнала: wdt_event_o, wdt_irq_o, и wdt_reset_o.

Сигнал wdt_event_o указывает на события достижения значения для выработки раннего запроса на прерывание (биты регистра WDT_SETTINGS[31:16]) или значения для выработки запроса на прерывание (регистр WDT_VALUE = 0x0). При любом из указанных событий изменяется уровень сигнала wdt_event_o либо с "0" на "1", либо с "1" на "0". При программном сбросе сигнал wdt_event_o сохраняет свое значение. При аппаратном сбросе сигнал wdt_event_o == 0.

Сигнал wdt_event_strobe_o указывает на те же события, что и сигнал wdt_event_o. При любом из указанных событий появляется строб с уровнем "1" длительностью 1 такт. При нескольких событиях, следующих друг за другом на 2-х соседних тактах, длина строба wdt_event_strobe_o будет также составлять 2 такта. При аппаратном или программном сбросе сигнал wdt_event_strobe_o == 0.

Сигнал wdt_irq_o указывает запрос на прерывание, что соответствует wdt_irq_o == 1. При wdt_irq_o == 0 запрос на прерывание отсутствует.

Сигнал wdt_reset_o указывает запрос на сброс системы, что соответствует wdt_reset_o == 1. После установки wdt_reset_o == 1 запускается необратимый процесс аппаратного сброса системы. В этот момент взаимодействие со сторожевым таймером невозможно. После аппаратного сброса сигнал wdt_reset_o == 0. При wdt_reset_o == 0 запрос на сброс системы отсутствует.

1.3.13.5.3.5 Запуск и остановка сторожевого таймера (WDT_ENABLE)

Запуск и остановка сторожевого таймера осуществляется регистром WDT_ENABLE.

Запись значения 0xEAAE в регистр WDT_ENABLE запускает таймер, запуская убывание счетчика. Запись 0xB4 останавливает таймер, сохраняя установленные настройки. Запись любого другого значения не оказывает эффекта.

При остановке сторожевого таймера сохраняется текущее значение счетчика, пока бит WDT_LOCK[1] == 0x1. После этого, когда бит WDT_LOCK[1] == 0x0, значение счетчика устанавливается в начальное значение, заданное текущими настройками. Это обуславливается прохождением внутренней синхронизацией сигналов.

Длительность прохождения внутренней синхронизации сигналов [2 * (3 такта pclk_i + 3 такта clk_wdt_i)].

После записи значения 0xEAAE или 0xB4 в регистр WDT_ENABLE регистры WDT_SETTINGS, WDT_START_VALUE, WDT_WINDOW, WDT_REFRESH заблокированы для записи на время прохождения внутренней синхронизации сигналов.

После записи значения 0xEAAE сторожевой таймер запускается с параметрами, ранее установленными в регистры WDT_SETTINGS, WDT_START_VALUE, WDT_WINDOW. Возможны несколько последовательных записей значения 0xEAAE, при этом на запуск таймера влияет только первая запись, остальные игнорируются.

После записи значения 0xB4 сторожевой таймер останавливается и сохраняет параметры, ранее установленные в регистры WDT_SETTINGS, WDT_START_VALUE, WDT_WINDOW. Значение регистра WDT_VALUE соответствует значению регистра WDT_START_VALUE. Возможны несколько последовательных записей значения 0xB4, при этом на остановку таймера влияет только первая запись, остальные игнорируются.

1.3.13.5.3.6 Приостановка счетчика сторожевого таймера (WDT_SUSPEND)

Приостановка счетчика сторожевого таймера осуществляется путем записи значения 0xDFDF в регистр WDT_SUSPEND. Запись 0xD2 выводит счетчик сторожевого таймера из режима приостановки. Запись любого другого значения не оказывает эффекта.

Приостановка счетчика сторожевого таймера означает сохранение счетчиком текущего значения WDT_VALUE и статуса сторожевого таймера WDT_IRQ_STATUS.

При приостановке счетчика сторожевого таймера возможна загрузка новых параметров, но их установка невозможна. Также невозможен перезапуск счетчика таймера регистром WDT_REFRESH.

После записи значения 0xDFDF в регистр WDT_SUSPEND происходит приостановка счетчика таймера. При приостановке счетчика сторожевого таймера доступна запись в регистры, при этом новые значения регистров WDT_SETTINGS, WDT_START_VALUE, WDT_WINDOW не будут применены. При приостановки счетчика сторожевого таймера перезапуск счетчика регистром WDT_REFRESH невозможна.

В режиме приостановки регистр WDT_ENABLE доступен для записи. Поведение таймера будет соответствовать обычной работе с регистром WDT_ENABLE за исключением того, что при записи значения 0xB4 в регистр WDT_ENABLE (остановка счетчика), а затем записи в регистр WDT_ENABLE значения 0xEAAE (запуск счетчика), счетчик не начнет отсчет до выхода из режима приостановки (запись значения 0xD2 в регистр WDT_SUSPEND).

После записи значения 0xDFDF в регистр WDT_SUSPEND приостановка счетчика произойдет после прохождения внутренней синхронизации сигналов.

Длительность прохождения внутренней синхронизации сигналов равна: [2 * (3 такта pclk_i + 3 такта clk_wdt_i)].

Во время прохождения внутренней синхронизации сигналов сторожевого таймера все регистры сторожевого таймера доступны для записи или чтения.

1.3.13.5.3.7 Программный сброс сторожевого таймера (WDT_SW_RESET)

Программный сброс сторожевого таймера осуществляется путем записи в регистр WDT_SW_RESET значения 0xADDA. Запись любого другого значения не оказывает влияния.

После записи значения 0xADDA в регистр WDT_SW_RESET начинается программный сброс сторожевого таймера, который длится [2 * (3 такта pclk_i + 3 такта clk_wdt_i)]. Длительность программного сброса обуславливается прохождением внутренней синхронизации сигналов сторожевого таймера.

Во время программного сброса запись во все регистры доступна, но будет игнорироваться.

До завершения программного сброса чтение регистра WDT_SW_RESET возвращает значение 0x1, чтение из других регистров возвращает 0x0. После завершения программного сброса чтение регистра WDT_SW_RESET возвращает значение 0x0.

Регистр WDT_SW_RESET можно использовать для остановки таймера. Значения всех регистров сторожевого таймера, кроме WDT_SW_RESET, устанавливаются в значение по сбросу.

Во время приостановки счетчика таймера при помощи регистра WDT_SUSPEND использование регистра WDT_SW_RESET также устанавливает значения всех регистров сторожевого таймера, кроме WDT_SW_RESET, в значение по сбросу.

Регистр WDT_SW_RESET, после выполнения программного сброса, устанавливается в значение по сбросу.

Сигналы wdt_enable_ext_i, wdt_lock_ext_i и wdt_suspend_ext_i сохранят свое значение после программного сброса.

1.3.13.5.3.8 Установка предделителя, настроек работы и значения для раннего запроса на прерывание (WDT_SETTINGS)

В качестве настроек через регистр WDT_SETTINGS доступны выработка раннего запроса на прерывание и установка его значения, установка предделителя, выработка запроса на прерывания или сброс системы при переполнении счетчика.

С осторожностью стоит устанавливать значение выработки раннего запроса на прерывание, слишком близкое к начальному значению счетчика. В этом случае сторожевой таймер может выработать ранний запрос на прерывание неожиданно рано для пользователя даже при нормальной работе системы.

Биты регистра WDT_SETTINGS[3:0] служат для установки предделителя. Предделитель необходим увеличения периода времени, за который происходит изменение WDT_VALUE на один отсчет и считается по формуле:

– one_value_period [time] = prescaler * (1/clk_wdt_i) [time]

Время работы сторожевого таймера до полной обстановки рассчитывается по формуле:

– watchdog_period [time] = wdt_start_value_reg * one_value_period [time]

Биты регистра WDT_SETTINGS[31:16] служат для установки разрешения на выработку раннего запроса на прерывание. Если WDT_SETTINGS[31:16] > 0x0, то при достижении значения WDT_SETTINGS[31:16] будет выработан ранний запрос на прерывание. При WDT_SETTINGS[31:16] == 0x0 ранний запрос на прерывание не будет выработан.

Бит регистра WDT_SETTINGS[4] служит для выбора выработки запроса на прерывания или выработка запроса на сброс системы при переполнении счетчика. В один момент времени может быть выбрана только одна настройка.

Загрузка нового значения регистра WDT_SETTINGS происходит в любой момент времени. Применение нового значения регистра WDT_SETTINGS происходит либо при запуске таймера (см. регистр WDT_ENABLE) либо при перезапуске счетчика (см. регистр

WDT_REFRESH). Приостановка счетчика не позволяет применить нового значения регистра WDT_SETTINGS (см. регистр WDT_SUSPEND).

1.3.13.5.3.9 Запрет на изменение настроек сторожевого таймера (WDT_LOCK)

Регистр WDT_LOCK служит для запрета на изменение настроек сторожевого таймера.

Пользователю доступна только запись в биты регистра WDT_LOCK[15:0]. Запись в остальные биты регистра недоступна.

Бит WDT_LOCK[0] указывает запрет на изменение настроек сторожевого таймера. Он устанавливается в момент включения системы (сразу после аппаратного сброса) в зависимости от сигнала wdt_lock_ext_i или после записи WDT_LOCK[15:0] == 0xDB26. В этом случае для пользователя остаются доступны для записи только регистры WDT_ENABLE, WDT_SUSPEND, WDT_REFRESH.

Бит WDT_LOCK[1] автоматически устанавливается при записи одного из значимых значений в регистр WDT_ENABLE. Пока установлен этот бит, регистры WDT_SETTINGS, WDT_START_VALUE, WDT_WINDOW, WDT_REFRESH заблокированы для записи. В этот момент происходит прохождение внутренней синхронизации сигналов. После прохождения внутренней синхронизации сигналов бит WDT_LOCK[1] == 0 и появляется возможность записи в заблокированные ранее регистры.

Бит WDT_LOCK[2] автоматически устанавливается в момент перезапуска счетчика (после корректной записи в регистр WDT_REFRESH). Пока установлен этот бит, регистры WDT_SETTINGS, WDT_START_VALUE, WDT_WINDOW, WDT_REFRESH заблокированы для записи, а повторная запись в WDT_REFRESH невозможна. В этот момент проходит внутренняя синхронизация сигналов. После прохождения внутренней синхронизации сигналов бит WDT_LOCK[2] == 0 и появляется возможность записи в заблокированные ранее регистры.

1.3.13.5.3.10 Перезапуск счетчика сторожевого таймера (WDT_REFRESH)

Перезапуск счетчика сторожевого таймера осуществляется путем записи значения 0xAAAA в регистр WDT_REFRESH.

Периодический перезапуск счетчика должна выполняться в ходе нормальной работы системы. Таким образом, отсутствие перезапуска счетчика будет сигнализировать о сбое в работе.

Перезапуск счетчика невозможен при приостановке счетчика сторожевого таймера (см. регистр WDT_SUSPEND) или при выключенном счетчике (см. регистр WDT_ENABLE).

В режиме работы с временным окном перезапуска счетчика должен произойти в пределах этого окна, заданных раннее. Значения, указанные в регистре WDT_WINDOW также входят во временное окно, где можно перезапустить счетчик. В этом случае, счетчик обновит свое значение на начальное и продолжит убывать. Если попытка перезапуска произошла вне границ временного окна, то счетчик продолжит убывать далее, пока не достигнет значения WDT_VALUE = 0x0. При этом установится один из флагов, показывающий, где произошла попытка перезапуска счетчика - перед значением верхней или после значения нижней границы временного окна.

После попытки перезапуска при значении счетчика большем значения верхней границы временного окна или меньшем значения нижней границы временного окна дальнейшая возможность перезапустить счетчик отсутствует.

Для того, чтобы гарантировано успешно перезапустить счетчик, необходимо записать значение 0xAAAA до момента, пока WDT_VALUE > 0x3.

В режиме работы без временного окна его границы необходимо установить максимальными. При перезапуске счетчика он обновит свое значение на начальное и продолжит убывать.

После записи значения 0xAAAA в регистр WDT_REFRESH регистры WDT_SETTINGS, WDT_START_VALUE, WDT_WINDOW, WDT_REFRESH заблокированы для записи на время прохождения внутренней синхронизации сигналов.

Длительность прохождения внутренней синхронизации сигналов равна: [2 * (3 такта pclk_i + 3 такта clk_wdt_i)].

Во время прохождения внутренней синхронизации сигналов бит WDT_LOCK[1] установлен в 1. После завершения прохождения внутренней синхронизации сигналов бит WDT_LOCK[1] устанавливается в 0, и (при выполнении условий для перезапуска счетчика) регистр WDT_VALUE устанавливается в значение регистра WDT_START_VALUE

1.3.13.5.3.11 Установка начального значения счетчика (WDT_START_VALUE)

Установка начального значения счетчика осуществляется записью в биты регистра WDT_START_VALUE[15:0] и может принимать любое значение.

С осторожностью стоит устанавливать слишком малое начального значения счетчика, т.к. система, даже при нормальной работе, может не успеть перезапустить счетчик и будет выработан запрос на прерывание или запрос на сброс (в зависимости от на настроек WDT_SETTINGS[4]).

Загрузка нового значения регистра WDT_START_VALUE происходит в любой момент времени. Применение нового значения регистра WDT_START_VALUE происходит либо при запуске таймера (см. регистр WDT_ENABLE) либо при перезапуске счетчика (см. регистр WDT_REFRESH). Приостановка счетчика не позволяет применить нового значения регистра WDT_START_VALUE (см. регистр WDT_SUSPEND).

1.3.13.5.3.12 Установка временных границ окна (WDT_WINDOW)

Установка верхней границы временного окна осуществляется записью в биты регистра WDT_WINDOW[31:16], а нижней - битами регистра WDT_WINDOW[15:0].

Установленные границы входят в размер временного окна, во время которого можно перезапускать счетчик таймера через регистр WDT_REFRESH.

С осторожностью стоит устанавливать слишком узкие границы временного окна, т.к. даже при нормальной работе системы, в момент попытки перезапуска счетчика сторожевой таймер может выработан один из запросов на прерывание неожиданно для пользователя.

Загрузка нового значения регистра WDT_WINDOW происходит в любой момент времени. Применение нового значения регистра WDT_WINDOW происходит либо при запуске таймера (см. регистр WDT_ENABLE) либо при перезапуске счетчика (см. регистр WDT_REFRESH). Приостановка счетчика сторожевого таймера не позволяет применить нового значения регистра WDT_WINDOW (см. регистр WDT_SUSPEND).

1.3.13.5.3.13 Чтение текущего значения счетчика сторожевого таймера (WDT_VALUE)

Регистр WDT_VALUE предназначен для чтения текущего значения счетчика и не доступен для записи.

Считанное из регистра значение счетчика может отличаться от истинного на 1-2 отсчета. Это обуславливается прохождением внутренней синхронизации сигналов сторожевого таймера.

1.3.13.5.3.14 Чтение статусного регистра сторожевого таймера (WDT_IRQ_STATUS)

Статусный регистр WDT_IRQ_STATUS указывает на события которые произошли в сторожевом таймере. Отображение в статусном регистре зависит от настройки регистра маски WDT_IRQ_MASK.

Регистр WDT_IRQ_STATUS доступен только для чтения. Очистка регистра доступна при помощи регистра WDT_IRQ_CLEAR.

1.3.13.5.3.15 Маска событий сторожевого таймера (WDT_IRQ_MASK)

Регистр маски событий WDT_IRQ_MASK позволяет выбирать, какие события сторожевого таймера, отображенные в регистре WDT_IRQ_STATUS, будут вырабатывать запросы на прерывание.

1.3.13.5.3.16 Очистка статусного регистра сторожевого таймера (WDT_IRQ_CLEAR)

Регистр очистки статуса сторожевого таймера WDT_IRQ_CLEAR позволяет побитно очищать регистр WDT_IRQ_STATUS.

Чтение из регистра WDT_IRQ_CLEAR возвращает 0x0.

1.3.13.5.3.17 Ручная установка статусного регистра сторожевого таймера (WDT_IRQ_TEST)

Регистр ручной установки статусного регистра сторожевого таймера WDT_IRQ_TEST позволяет побитно устанавливать "1" в регистр WDT_IRQ_STATUS.

Чтение из регистра WDT_IRQ_TEST возвращает последнее записанное в него значение.

1.3.13.5.3.18 Выходные сигналы сторожевого таймера

Выходные сигналы сторожевого таймера образуются путем применения регистра маски WDT_IRQ_MASK на статусный регистр WDT_IRQ_STATUS.

В качестве выходных используется четыре сигнала:

– wdt_event_strobe_o - запросы на прерывание при достижении значения WDT_VALUE == 0x0 или значения WDT_SETTING[1)] (строб с уровнем "1");

– wdt_event_o - запросы на прерывание при достижении значения WDT_VALUE == 0x0 или значения WDT_SETTING[1)] (изменение уровня с "0" на "1" или с "1" на "0");

– wdt_irq_o - общий запрос на прерывание при вырабатывании одного из запросов (WDT_IRQ_STATUS & WDT_IRQ_MASK);

– wdt_reset_o - выработан сигнал сброса при счетчика таймера WDT_VALUE == 0x0.

Примечание:

1) Значение WDT_SETTING[31:16] учитывается при настройке WDT_SETTING[6] == 0x1.

1.3.13.5.3.19 Простой режим работы со сторожевым таймером

Самым простым режимом работы со сторожевым таймером является его запуск и дальнейшее периодическое обновление счетчика в ходе выполнения программы. Сторожевой таймер будет работать с предустановленными параметрами.

Последовательность действий, если таймер выключен:

```
1) Записать в регистр WDT_ENABLE значение 0xEAAE.
2) Периодически обновлять счетчик путем записи в регистр WDT_REFRESH значения 0xAAAA.
```

Последовательность действий, когда таймер уже запущен:

```
1) Периодически обновлять счетчик путем записи в регистр WDT_REFRESH значения 0xAAAA.
```

При достижении счетчиком значения WDT_VALUE == 0x0 будет сгенерирован сигнал сброса системы.

Предустановленные параметры сторожевого таймера приведены в таблице 1.634.

Таблица 1.634 – Предустановленные параметры сторожевого таймера (значимые биты)

Биты	Значение	Функционал
WDT_SETTINGS = 0x7FFF0070		
[3:0]	0x0	Предделитель 4
[4]	0x0	Генерация сигнала сброса при переполнении счетчика (вместо запроса на прерывание)
[6]	0x0	Запрещена выработка раннего запроса на прерывание
[31:16]	0x0	Значение для раннего запроса на прерывание
WDT_START_VALUE = 0x0000FFFF		
[15:0]	0xFFFF	Начальное значение счетчика
WDT_WINDOW = 0xFFFF0000		
[15:0]	0x0	Нижняя граница окна
[31:16]	0xFFFF	Верхняя граница окна

Настройка маски прерывания сторожевого таймера для простого режима работы приведена в таблице 1.635.

Таблица 1.635 – Настройка маски запросов на прерывание сторожевого таймера (значимые биты)

Биты	Значение	Функционал
WDT_IRQ_MASK = 0x00000000		
[0]	0x0	Запретить выработку запроса на прерывание при достижением счетчиком WDT_VALUE == 0x0
[1]	0x0	Запретить выработку раннего запроса на прерывание при достижении установленного значения

Биты	Значение	Функционал
[2]	0x0	Запретить выработку запроса на прерывание при попытке перезапуска счетчика, когда значение счетчика больше значения верхней границы временного окна
[3]	0x0	Запретить выработку запроса на прерывание при попытке перезапуска счетчика, когда значение счетчика меньше значения нижней границы временного окна

1.3.13.5.3.20 Режим работы со сторожевым таймером с временным окном

Одним из режимов работы со сторожевым таймером является режим с использованием временного окна. Для этого в задаются его верхняя и нижняя границы, в пределах которого возможен перезапуск счетчика.

При нормальном перезапуске счетчика в пределах временного окна сторожевой таймер продолжит работу, установив счетчик в начальное значение, заданное настройками. Далее счетчик снова начнет убывать.

При попытке перезапуска счетчика вне границ окна счетчик продолжит убывать, но будет сгенерирован сигнал события, а также установится флаг в статусе, указывающий, в какой момент времени была сделана попытка перезапуска - до верхней или после нижней границы временного окна.

Последовательность действий, если таймер выключен:

1) Установить новые значение границ временного окна через регистр WDT_WINDOW.
2) Установить новые настройки через регистр WDT_SETTINGS.
3) установить новое начальное значение счетчика WDT_START_VALUE.
4) Настроить маску при помощи регистра WDT_IRQ_MASK.
5) Записать в регистр WDT_ENABLE значение 0xEAAE для применения новых границ окна и других новых параметров.
6) Периодически обновлять счетчик путем записи в регистр WDT_REFRESH значения 0xAAAA в пределах границ установленного временного окна.

Альтернативная последовательность действий, когда таймер уже запущен:

1) Установить новые значение границ временного окна через регистр WDT_WINDOW.
2) Установить новые настройки через регистр WDT_SETTINGS.
3) установить новое начальное значение счетчика WDT_START_VALUE.
4) Настроить маску при помощи регистра WDT_IRQ_MASK.
5) Записать в регистр WDT_REFRESH значение 0xAAAA для применения новых границ окна и других новых параметров.
6) Периодически обновлять счетчик путем записи в регистр WDT_REFRESH значения 0xAAAA в пределах границ установленного временного окна.

Настройка маски выработки запросов на прерывание сторожевого таймера для режима работы приведена в таблице 1.636.

Таблица 1.636 – Настройка маски выработки запросов на прерывание сторожевого таймера (значимые биты)

Биты	Значение	Функционал
WDT_IRQ_MASK = 0x00000000F		
[0]	0x1	Разрешить выработку запроса на прерывание при достижением счетчиком WDT_VALUE == 0x0
[1]	0x1	Разрешить выработку раннего запроса на прерывание при достижении установленного значения
[2]	0x1	Разрешить выработку запроса на прерывание при попытке перезапуска счетчика, когда значение счетчика больше значения верхней границы временного окна
[3]	0x1	Разрешить выработку запроса на прерывание при попытке перезапуска счетчика, когда значение счетчика меньше значения нижней границы временного окна

1.3.13.5.3.21 Режим работы со сторожевым таймером без временного окна

Одним из режимов работы со сторожевым таймером является режим без использования временного окна.

При нормальном перезапуске счетчика сторожевой таймер продолжит работу. Счетчик снова начнет убывать.

Последовательность действий, если таймер выключен:

```
1) Если необходимо, установить новые настройки через регистр WDT_SETTINGS и установить н
овое начальное значение счетчика WDT_START_VALUE.
2) Записать в регистр WDT_WINDOW значение 0xFFFF0000.
3) Настроить маску при помощи регистра WDT_IRQ_MASK.
4) Записать в регистр WDT_ENABLE значение 0xEAAE для применения новых параметров.
5) Периодически обновлять счетчик путем записи в регистр WDT_REFRESH значения 0xAAAA в п
ределах границ установленного временного окна.
```

Альтернативная последовательность действий, если таймер уже запущен:

```
1) Если необходимо, установить новые настройки через регистр WDT_SETTINGS и установить н
овое начальное значение счетчика WDT_START_VALUE.
2) Записать в регистр WDT_WINDOW значение 0xFFFF0000.
3) Настроить маску при помощи регистра WDT_IRQ_MASK.
4) Записать в регистр WDT_REFRESH значение 0xAAAA для применения новых параметров в пред
елах границ предыдущего установленного временного окна.
5) Периодически обновлять счетчик путем записи в регистр WDT_REFRESH значения 0xAAAA.
```

Настройка маски выработки запросов на прерывание сторожевого таймера для простого режима работы приведена в таблице 1.637.

Таблица 1.637 – Настройка маски выработки запросов на прерывание сторожевого таймера (значимые биты)

Биты	Значение	Функционал
WDT_IRQ_MASK = 0x00000002		
[0]	0x0	Запретить выработку запроса на прерывание при достижением счетчиком WDT_VALUE == 0x0
[1]	0x1	Разрешить выработку раннего запроса на прерывание при достижении установленного значения
[2]	0x0	Запретить выработку запроса на прерывание при попытке перезапуска счетчика, когда значение счетчика больше значения верхней границы временного окна
[3]	0x0	Запретить выработку запроса на прерывание при попытке перезапуска счетчика, когда значение счетчика меньше значения нижней границы временного окна

1.3.13.5.3.22 Обновление настроек сторожевого таймера

Новые настройки сторожевого таймера применяются в нескольких случаях:

1) После записи в регистр WDT_ENABLE значения 0xEAAE (запуск сторожевого таймера).
2) После записи в регистр WDT_REFRESH значения 0xAAAA (во время работы сторожевого таймера).

Обновление настроек никаким из указанных способов невозможно, пока счетчик сторожевого таймера приостановлен при помощи регистра WDT_SUSPEND или сигнала wdt_suspend_ext_i.

При необходимости изменить настройки сторожевого таймера во время его работы, запись в регистр WDT_REFRESH должна происходить в границах ранее выставленного временного окна. В противном случае будет выработан один из запросов на прерывание работы таймера, которые отображаются в регистре WDT_IRQ_STATUS.

1.3.13.5.3.23 Полная остановка работы сторожевого таймера

Существует несколько способов полной остановки работы сторожевого таймера:

1) При помощи регистра WDT_ENABLE.
2) При помощи программного сброса через регистр WDT_SW_RESET.

Если биты WDT_LOCK[2] == 0x0 и WDT_LOCK[1] == 0x0, что означает успешное прохождение внутренней синхронизации сигналов, то запись значение 0xB4 в регистр WDT_ENABLE останавливает сторожевой таймер, сохраняя ранее установленные регистры WDT_SETTINGS, WDT_START_VALUE, WDT_WINDOW. После остановки сторожевого таймера запись в регистр WDT_REFRESH невозможна до момента включения сторожевого таймера.

Программный спрос при помощи записи в регистр WDT_SW_RESET значения 0xADDA устанавливает все регистры таймера в состояние по сбросу, таким образом останавливая сторожевой таймер. Взаимодействие с регистром WDT_SW_RESET возможно в любой момент времени. При этом, если был установлен сигнал wdt_suspend_ext_i == 1, то этот сигнал сохранит свое значение. Необходимо будет сбросить сигнал wdt_suspend_ext_i перед следующей записью в регистр WDT_ENABLE для запуска счетчика.

1.3.13.5.4 Описание регистров сторожевого таймера WDT

1.3.13.5.4.1 Основные особенности регистров

Установка параметров сторожевого таймера WDT и отображение его текущего состояния ведется через регистры. Доступ к регистрам осуществляется посредством системного интерфейса управления APB 5 slave. Поддерживается только 32-х разрядный доступ. Байтовая, двух-байтовая или трех-байтовая запись будет проигнорирована и возвращена ошибка шины, что вызовет переход процессорного ядра в исключение.

Карта регистров приведена в 1.638.

Все регистры, кроме регистра текущего значения счетчика WDT_VALUE и регистра статуса WDT_IRQ_STATUS обновляются на частоте интерфейса APB. Регистры WDT_VALUE и WDT_IRQ_STATUS относятся непосредственно к счетчику таймера и обновляются на другой частоте. Считанное из регистров WDT_VALUE и WDT_IRQ_STATUS значения отличаются от истинных не менее, чем на один такт.

Чтение/запись в регистры WDT доступно в любой момент времени, если таймеру при включении системы была разрешена работа. В противном случае, WDT не будет реагировать на попытки чтения/записи.

Запись:

Попытка записи проходит успешно и без ошибок, если происходит по адресам регистров WDT, которые указаны в таблице 1.638. При этом, после записи специальных значений в один из двух регистров WDT_ENABLE или WDT_REFRESH устанавливается временный запрет на запись в некоторые регистры. Статус временного запрета на запись отображается в регистре WDT_LOCK. Временный запрет на запись реализован для защиты текущих параметров таймера и является особенностью установления внутренней синхронизации сигналов. Попытка записи в любой регистр в этот момент завершится успешно, но состояние регистров не

изменится. При попытке записи по зарезервированным адресам WDT возвращает ошибку 0xE4E4E4E4.

Чтение:

Попытка чтения проходит успешно и без ошибок, если происходит по адресам регистров WDT, которые указаны в таблице 1.638. Чтение разрешено в любой момент времени. При попытке чтения по зарезервированным адресам WDT возвращает ошибку 0xE4E4E4E4.

1.3.13.5.4.2 Карта регистров сторожевого таймера WDT

При дальнейшем описании отдельных полей регистров используется следующее обозначение режимов доступа пользователя к полям:

– RO – поле доступно только для чтения. Запись данных в поле игнорируется;

– WO – поле доступно только для записи. Чтение возвращает значение 0;

– RW – поле доступно как для записи, так и для чтения;

– W1R0 - поле доступно для чтения. Запись 1 сбросит значение, запись 0 не имеет действия.

Таблица 1.638 – Карта регистров сторожевого таймера WDT

Адрес	Название	Доступ	Значение после сброса	Описание
0x000	WDT_ID	RO	0x20544457[1)]	Наименование СФ-блока
0x004	WDT_VERSION	RO	0x302E3176[2)]	Используемая версия СФ-блока
0x008	WDT_ENABLE[3)]	RW	0x0	Запуск и остановка таймера
0x00C	WDT_SUSPEND	RW	0x0	Остановка счетчика (спящий режим)
0x010	WDT_SW_RESET[3)]	RW	0x0	Программный сброс таймера
0x014	WDT_SETTINGS	RW	0x0	Установка предделителя и значения раннего запроса на прерывание
0x018	WDT_LOCK[4)]	RW	0x0	Запрет на изменение всех параметров таймера
0x01C	WDT_REFRESH[3)]	RW	0x0	Перезапуск счетчика
0x020	WDT_START_VALUE	RW	0x0000FFFF	Начальное значение отсчета счетчика
0x024	WDT_WINDOW	RW	0xFFFF0000	Границы временного окна таймера
0x028	WDT_VALUE[5)]	RO	0x0000FFFF	Значение счетчика
0x02C	WDT_IRQ_STATUS	RO	0x0	Статус таймера
0x030	WDT_IRQ_MASK	RW	0x0	Побитовая маска для регистра статуса таймера
0x034	WDT_IRQ_CLEAR	W1R0	0x0	Побитовая очистка регистра статуса таймера
0x038	WDT_IRQ_TEST	RW	0x0	Побитовая ручная установка "1" в регистр статуса таймера

Примечания:

1) Наименование "WDT" в ASCII символах, записанное справа налево.
2) Зависит от параметров, заданных при конфигурации СФ-блока.

3) Запись специальных значений в эти регистры запрещает запись в любой регистр WDT на 3 такта независимого источника тактирования. WDT выдает ошибку 0xE4E4E4E4.
4) Может устанавливаться при включении системы в зависимости от предустановленных настроек.
5) Считанное из регистра значение счетчика таймера может отличаться от истинного на 1-2 отсчета.
Это обуславливается внутренней синхронизацией сигналов сторожевого таймера.

1.3.13.5.4.3 Описание полей регистров WDT

1.3.13.5.4.3.1 WDT_ID (0x000) - Наименование СФ-блока

Регистр доступен только для чтения.

Описание полей регистра WDT_ID приведено в таблице 1.639.

Таблица 1.639 – Описание полей регистра WDT_ID

Бит	Поле	Доступ	Значение после сброса	Описание
31:0	id	RO	0x20544457	Наименование СФ-блока ("WDT", записанное ASCII символами справа налево)

1.3.13.5.4.3.2 WDT_VERSION (0x004) - Версия СФ-Блока

Регистр доступен только для чтения.

Описание полей регистра WDT_VERSION приведено в таблице 1.640.

Таблица 1.640 – Описание полей регистра WDT_VERSION

Бит	Поле	Доступ	Значение после сброса	Описание
31:0	version	RO	0x302E3176	Используемая версия СФ-блока (записанная в ASCII символах справа налево)

1.3.13.5.4.3.3 WDT_ENABLE (0x008) - Запуск и остановка сторожевого таймера

Регистр доступен для записи/чтения.

Описание полей регистра WDT_ENABLE для записи приведено в таблице 1.641.

Таблица 1.641 – Описание полей регистра WDT_ENABLE во время записи

Бит	Поле	Доступ	Значение после сброса	Описание
15:0	enable[1)]	WO	0x0	Запись 0xEAAE запускает таймер. Запись 0xB4 останавливает таймер. Запись любого другого значения не оказывает эффекта
31:16	-	RO	0x0	Резерв

Примечание:

1) В случае, если установлен WDT_SUSPEND, таймер не применяет новые загруженные параметры.

Описание полей регистра WDT_ENABLE для чтения приведено в таблице 1.642.

Таблица 1.642 – Описание полей регистра WDT_ENABLE во время чтения

Бит	Поле	Доступ	Значение после сброса	Описание
0	enable_status	RO	0x0	"1" - таймер запущен "0" - таймер остановлен
31:1	-	RO	0x0	Резерв

1.3.13.5.4.3.4 WDT_SUSPEND (0x00C) - Приостановка работы сторожевого таймера

Регистр доступен для записи/чтения.

Описание полей регистра WDT_SUSPEND во время записи приведено в таблице 1.643.

Таблица 1.643 – Описание полей регистра WDT_SUSPEND во время записи

Бит	Поле	Доступ	Значение после сброса	Описание
15:0	suspend[1)]	WO	0x0	Запись 0xDFDF приостанавливает счетчик. Запись 0xD2 возобновляет счетчик. Запись любого другого значения не оказывает эффекта
31:16	-	RO	0x0	Резерв

Примечание:

1) Во время приостановки счетчика возможна запись новых параметров в регистры WDT_SETTINGS, WDT_START_VALUE, WDT_WINDOW. Они будут сохранены, но не применены.
2) Перезапуск счетчика таймера регистром WDT_REFRESH невозможна.

Описание полей регистра WDT_SUSPEND во время чтения приведено в таблице 1.644.

Таблица 1.644 – Описание полей регистра WDT_SUSPEND во время чтения

Бит	Поле	Доступ	Значение после сброса	Описание
0	suspend_status	RO	0x0	"1" - счетчик приостановлен "0" - счетчик не приостановлен
31:1	-	RO	0x0	Резерв

1.3.13.5.4.3.5 WDT_SW_RESET (0x010) - Программный сброс сторожевого таймера

Регистр доступен для записи/чтения.

Описание полей регистра WDT_SW_RESET во время записи приведено в таблице 1.645.

Таблица 1.645 – Описание полей регистра WDT_SW_RESET во время записи

Бит	Поле	Доступ	Значение после сброса	Описание
15:0	sw_reset	WO	0x0	Запись 0xADDA инициирует программный сброс таймера. После выполнения программного сброса устанавливается 0x0. Запись других значений не оказывает эффекта
31:16	-	RO	0x0	Резерв

Описание полей регистра WDT_SW_RESET во время записи приведено в таблице 1.646.

Таблица 1.646 – Описание полей регистра WDT_SW_RESET во время чтения

Бит	Поле	Доступ	Значение после сброса	Описание
0	sw_reset	RO	0x0	"1" - происходит программный сброс сторожевого таймера[1)] "0" - в данный момент программный сброс сторожевого таймера не происходит
31:1	-	RO	0x0	Резерв

Примечание:

1) Во время программного сброса чтение из любых других регистров будет возвращать значение 0x0.

1.3.13.5.4.3.6 WDT_SETTINGS (0x014) - Установка предделителя, значения для раннего запроса на прерывание, оконного режима, типа генерируемого сигнала при переполнении счетчика

Регистр доступен для записи/чтения.

Описание полей регистра WDT_SETTINGS приведено в таблице 1.647.

Таблица 1.647 – Описание полей регистра WDT_SETTINGS

Бит	Поле	Доступ	Значение после сброса	Описание
3:0	prescaler	RW	0x0	0x0 - предделитель 4 0x1 - предделитель 8 0x2 - предделитель 16 0x3 - предделитель 32 0x4 - предделитель 64 0x5 - предделитель 128 0x6 - предделитель 256 0x7 - предделитель 512 другое - предделитель 1024
4	irq_or_reset	RW	0x0	При достижении счетчиком значения 0x0: "1" - разрешить выработку запроса на прерывание и запретить выработку запроса на сброс системы "0" - запретить выработку запроса на прерывание и разрешить выработку запроса на сброс системы

Бит	Поле	Доступ	Значение после сброса	Описание
15:5	-	RO	0x0	Резерв
31:16	ewirq_value	RW	0x0	Значение для раннего запроса на прерывание. Если WDT_SETTINGS[31:16] > 0x0, на следующий отсчет счетчика после достижения этого значения вырабатывается запрос на прерывание WDT_IRQ_STATUS[1]

1.3.13.5.4.3.7 WDT_LOCK (0x018) - Запрет на изменение настроек сторожевого таймера

Регистр доступен для записи/чтения.

Описание полей регистра WDT_LOCK во время записи приведено в таблице 1.648.

Таблица 1.648 – Описание полей регистра WDT_LOCK во время записи

Бит	Поле	Доступ	Значение после сброса	Описание
15:0	lock_params_value	WO	0x0	Запись 0xDB26 запрещает изменение настроек сторожевого таймера и его программный сброс до момента аппаратного сброса. Запись других значений не оказывает эффекта.
31:16	-	RO	0x0	Резерв

Описание полей регистра WDT_LOCK во время чтения приведено в таблице 1.649.

Таблица 1.649 – Описание полей регистра WDT_LOCK во время чтения

Бит	Поле	Доступ	Значение после сброса	Описание
0	lock_parameters[3)]	RO	0x0	"1" - запрещено изменение параметров таймера и программный сброс (доступны только регистры WDT_ENABLE, WDT_SUSPEND, WDT_REFRESH). "0" - разрешено изменение параметров таймера Установка бита происходит при помощи внешних настроек в момент включения системы или при записи в регистр WDT_LOCK[15:0] значения 0xDB26. Бит сбрасывается только при аппаратном сбросе.
1	enable_on_event[2)]	RO	0x0	"1" - установлена внутренняя защита от записи в регистры WDT_SETTINGS, WDT_START_VALUE, WDT_WINDOW и WDT_REFRESH сторожевого таймера "0" - защита от записи убрана Бит автоматически устанавливается при ожидании внутренней синхронизации сторожевого таймера.

Бит	Поле	Доступ	Значение после сброса	Описание
2	refresh_event[1)]	RO	0x0	"1" - установлена внутренняя защита от записи в регистры WDT_SETTINGS, WDT_START_VALUE, WDT_WINDOW и WDT_REFRESH сторожевого таймера "0" - защита от записи убрана Бит автоматически устанавливается при ожидании внутренней синхронизации сторожевого таймера.
31:3	-	RO	0x0	Резерв

Примечание:

1) Внутренняя защита от записи устанавливается при записи значимого значения в регистр WDT_REFRESH.
Это - особенность внутренней синхронизации сигналов сторожевого таймера. После перезапуска значения счетчика бит устанавливается в "0".
2) Внутренняя защита от записи устанавливается при записи значимого значения в регистр WDT_ENABLE.
Это - особенность внутренней синхронизации сигналов сторожевого таймера. После установления синхронизации бит устанавливается в "0".
3) Убрать бит возможно только при аппаратном сбросе системы.

1.3.13.5.4.3.8 WDT_REFRESH (0x01C) - Перезапуск счетчика

Регистр доступен для записи/чтения.

Описание полей регистра WDT_REFRESH для записи приведено в таблице 1.650.

Таблица 1.650 – Описание полей регистра WDT_REFRESH во время записи

Бит	Поле	Доступ	Значение после сброса	Описание
15:0	refresh	WO	0x0	Запись 0xAAAA перезапускает счетчик таймера. Запись других значений не оказывает эффекта.
31:16	-	RO	0x0	Резерв

Описание полей регистра WDT_ENABLE для чтения приведено в таблице 1.651.

Таблица 1.651 – Описание полей регистра WDT_REFRESH во время чтения

Бит	Поле	Доступ	Значение после сброса	Описание
0	refresh_status	RO	0x0	"1" - Установлена внутренняя защита от записи в регистры WDT_SETTINGS, WDT_START_VALUE, WDT_WINDOW и WDT_REFRESH сторожевого таймера "0" - защита от записи убрана
31:1	-	RO	0x0	Резерв

1.3.13.5.4.3.9 WDT_START_VALUE (0x020) - Установка начального значения счетчика

Регистр доступен для записи/чтения.

Описание полей регистра WDT_START_VALUE приведено в таблице 1.652.

Таблица 1.652 – Описание полей регистра WDT_START_VALUE

Бит	Поле	Доступ	Значение после сброса	Описание
15:0	start_value	RW	0xFFFF	Начальное значение отсчета таймера
31:16	-	RO	0x0	Резерв

1.3.13.5.4.3.10 WDT_WINDOW (0x024) - Установка границ окна

Регистр доступен для записи/чтения.

Описание полей регистра WDT_WINDOW приведено в таблице 1.653.

Таблица 1.653 – Описание полей регистра WDT_WINDOW

Бит	Поле	Доступ	Значение после сброса	Описание
15:0	window_min	RW	0x0	Нижняя граница временного окна
31:16	window_max	RW	0xFFFF	Верхняя граница временного окна

1.3.13.5.4.3.11 WDT_VALUE (0x028) - Текущее значение счетчика

Регистр доступен только для чтения.

Описание полей регистра WDT_VALUE приведено в таблице 1.654.

Таблица 1.654 – Описание полей регистра WDT_VALUE

Бит	Поле	Доступ	Значение после сброса	Описание
15:0	value	RO	0xFFFF	Текущее значение счетчика таймера[1)]
31:16	-	RO	0x0	Резерв

Примечание:

1) Считанное из регистра значение счетчика таймера может отличаться от истинного на 1-2 отсчета.
Это обуславливается внутренней синхронизацией сигналов сторожевого таймера.

1.3.13.5.4.3.12 WDT_IRQ_STATUS (0x02C) - Статус сторожевого таймера

Регистр доступен только для чтения.

Описание полей регистра WDT_IRQ_STATUS приведено в таблице 1.655.

Таблица 1.655 – Описание полей регистра WDT_IRQ_STATUS

Бит	Поле	Доступ	Значение после сброса	Описание
0	irq[1),4),5)]	RO	0x0	"1" - достигнуто значение для выработки запроса на прерывание WDT_VALUE = 0x0 "0" - значение не достигнуто

Бит	Поле	Доступ	Значение после сброса	Описание
1	ewirq[1),3),4),5)]	RO	0x0	"1" - достигнуто значение для выработки раннего запроса на прерывание WDT_SETTINGS[31:16] "0" - значение не достигнуто
2	bcr[1),2),4),5)]	RO	0x0	"1" - попытка перезапуска счетчика, когда значение счетчика больше значения верхней границы временного окна "0" - отсутствие такой попытки
3	acr[1),2),4),5)]	RO	0x0	"1" - попытка перезапуска счетчика, когда значение счетчика меньше значения нижней границы временного окна "0" - отсутствие такой попытки
31:4	-	RO	0x0	Резерв

Примечания:

1) Сбросить флаги возможно при выполнении аппаратного сброса системы, программного сброса сторожевого таймера или побитовой записью в регистр WDT_IRQ_CLEAR.
2) Установка флагов зависит от границ установленного временного окна в регистре WDT_WINDOW.
3) Флаг устанавливается, если WDT_SETTINGS[31:16] > 0x0.
4) Установить флаги вручную можно при помощи регистра WDT_IRQ_TEST.
5) Установка флагов в регистре WDT_IRQ_STATUS не зависит от регистра WDT_IRQ_MASK.

1.3.13.5.4.3.13WDT_IRQ_MASK (0x030) - Маска обрабатываемых событий сторожевого таймера

Регистр доступен для записи/чтения.

Описание полей регистра WDT_IRQ_MASK приведено в таблице 1.656.

Таблица 1.656 – Описание полей регистра WDT_IRQ_MASK

Бит	Поле	Доступ	Значение после сброса	Описание
0	irq_mask	RW	0x0	"1" - разрешить выработку запроса на прерывание[1),2)] при достижении значения WDT_VALUE = 0x0 "0" - запретить выработку запроса на прерывание
1	ewirq_mask	RW	0x0	"1" - разрешить выработку запроса на прерывание[1)] при достижении значения WDT_SETTINGS[31:16] "0" - запретить выработку запроса на прерывание
2	bcr_mask	RW	0x0	"1" - разрешить выработку запроса на прерывание[1)] при попытке перезапуска счетчика, когда значение счетчика больше значения верхней границы временного окна "0" - запретить выработку запроса на прерывание
3	acr_mask	RW	0x0	"1" - разрешить выработку запроса на прерывание[1)] при попытке перезапуска счетчика, когда значение счетчика меньше значения нижней границы временного окна "0" - запретить выработку запроса на прерывание
31:4	-	RO	0x0	Резерв

Примечание:

1) Выходной сигнал запроса на прерывание общий, при выработке одного из этих запросов ус танавливается в "1".
2) При WDT_SETTINGS[4] == 0x1 разрешена выработка запроса на прерывание. При WDT_SETTING S[4] == 0x0 вместо выработки запроса на прерывание
будет выработан запросна сброс системы.

1.3.13.5.4.3.14 WDT_IRQ_CLEAR (0x034) - Очистка регистра статуса

Регистр доступен только для записи.

Чтение регистра возвращает 0x0.

Описание полей регистра WDT_IRQ_CLEAR для записи приведено в таблице 1.657.

Таблица 1.657 – Описание полей регистра WDT_IRQ_CLEAR

Бит	Поле	Доступ	Значение после сброса	Описание
0	irq_clear	W1R0	0x0	"1" - сбросить флаг регистра статуса WDT_IRQ_STATUS[0] "0" - запись не оказывает эффекта
1	ewirq_clear	W1R0	0x0	"1" - сбросить флаг регистра статуса WDT_IRQ_STATUS[1] "0" - запись не оказывает эффекта
2	bcr_clear	W1R0	0x0	"1" - сбросить флаг регистра статуса WDT_IRQ_STATUS[2] "0" - запись не оказывает эффекта
3	acr_clear	W1R0	0x0	"1" - сбросить флаг регистра статуса WDT_IRQ_STATUS[3] "0" - запись не оказывает эффекта
31:4	-	RO	0x0	Резерв

1.3.13.5.4.3.15 WDT_IRQ_TEST (0x038) - Ручное управление регистром статуса

Регистр доступен для записи/чтения.

Описание полей регистра WDT_IRQ_TEST приведено в таблице 1.658.

Таблица 1.658 – Описание полей регистра WDT_IRQ_TEST

Бит	Поле	Доступ	Значение после сброса	Описание
0	irq_test[1)]	RW	0x0	"1" - запись устанавливает флаг регистра статуса WDT_IRQ_STATUS[0] "0" - запись не оказывает эффекта
1	ewirq_test[1)]	RW	0x0	"1" - запись устанавливает флаг регистра статуса WDT_IRQ_STATUS[1] "0" - запись не оказывает эффекта
2	bcr_test[1)]	RW	0x0	"1" - запись устанавливает флаг регистра статуса WDT_IRQ_STATUS[2] "0" - запись не оказывает эффекта

Бит	Поле	До-ступ	Значение по-сле сброса	Описание
3	acr_test[1)]	RW	0x0	"1" - запись устанавливает флаг регистра статуса WDT_IRQ_STATUS[3] "0" - запись не оказывает эффекта
31:4	-	RO	0x0	Резерв

Примечание:

1) Регистр WDT_IRQ_TEST не сбрасывает установленные флаги WDT_IRQ_STATUS. Для этого используется регистр WDT_IRQ_CLEAR.

1.3.13.6 Блок счётчика реального времени (RTC)

1.3.13.6.1 Характеристики RTC

– APB интерфейс.

– Автоматическое переключение на батарейное питание.

– Хранение времени (секунды, минуты, часы, число, месяц, год) в формате BCD (binary-coded decimal).

– Автоматическое определение количества дней в месяце (28, 29, 30, 31).

– Будильник по дате и времени.

– Уровневые прерывания.

– Генерация wakeup события.

– Поддержка технологии TrustZone.

– Защита от записи.

– Дробный делитель.

– Вывод калибровочного сигнала с частотой 1Гц.

– Сохранение временных меток (timestamp) по переключению на батарейное питание или по событию от TAMP (см. раздел 1.3.15.6).

– Low-power mode. Два частотных домена, которые могут отключаться (см. раздел 1.3.8.2).

1.3.13.6.2 Описание RTC

RTC (Real Time Clock) — часы реального времени. Они необходимы для того, чтобы хранить текущее время. Из рис. 1.160 видно, что данный блок разделен на два отдельных домена: APB-домен и RTC-домен. Это сделано для того, чтобы была возможность отключить APB-домен, но при этом RTC-домен продолжал работать.

Отключенный домен — это домен с которого снят тактовый сигнал и опционально снято питание.

Основным источником тактового сигнала (вход rtc_clk_i) для RTC-домена служит LSE (см. рис. 1.22). Он имеет частоту 32768 Гц. При этом на APB-домен должен подаваться тактовый сигнал у которого частота больше чем LSE в два или более раза.

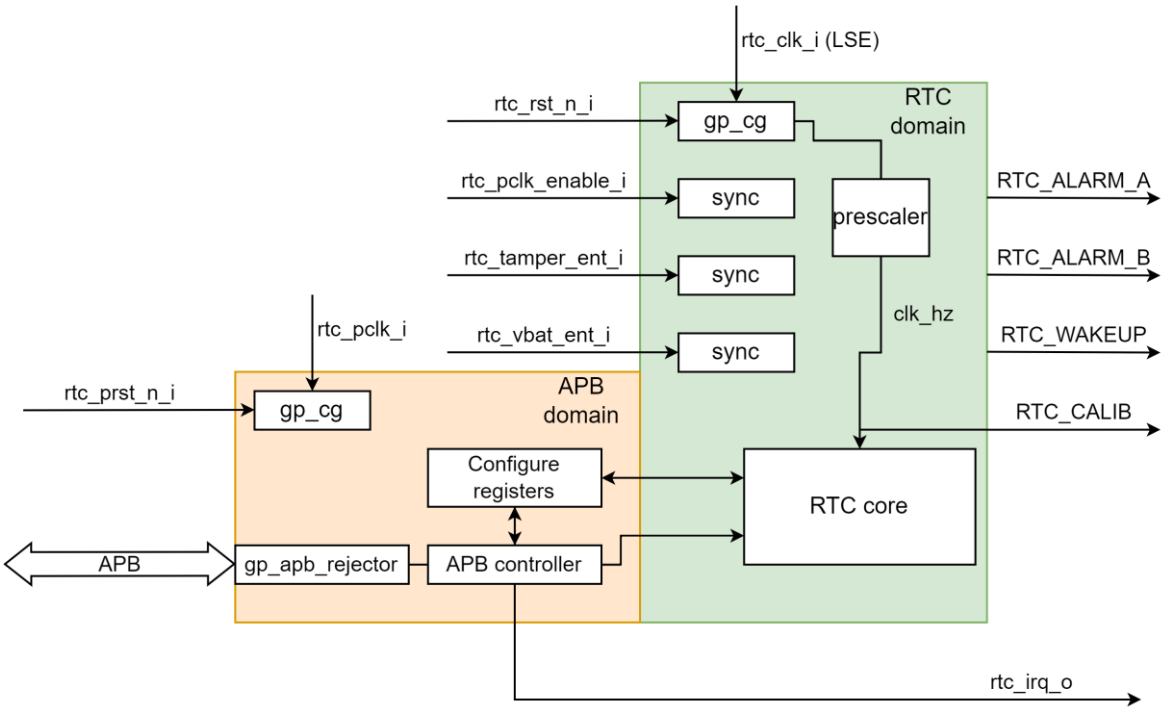


Рисунок 1.160 – Структурная схема RTC

Таблица 1.659 – Описание блоков RTC

Название блока	**Описание**
gp_cg	Clock gate. Отключает тактовый сигнал.
gp_apb_rejector	Служит для корректного завершения APB-транзакций, когда с APB-домена снят тактовый сигнал.
Configure register	Хранит значения теневых регистров в APB-домене.
APB controller	Производит запись и чтение из регистров по интерфейсу APB.
sync	Блок синхронизации сигнала.
prescaler	Делитель частоты внешнего сигнала rtc_clk_i.
RTC core	Основной блок RTC. Он хранит время, инкрементирует его, а также генерирует сигналы будильников (rtc_alarm_X_o).

1.3.13.6.3 Описание внешних выводов блока RTC

Список внешних выводов контроллера RTC и их описание приведены в таблице 1.660

Таблица 1.660 – Список внешних выводов контроллера RTC

Наименование вывода	Тип (Вх/Вых)	Функциональное назначение
RTC_ALARM_A	Вых	Сигнал будильника A
RTC_ALARM_B	Вых	Сигнал будильника B
RTC_CALIB	Вых	Сигнал калибровки
RTC_WAKEUP	Вых	Сигнал пробуждения. Объединенные по "или" сигналы будильников и TAMP_OUT0

1.3.13.6.4 Программный сброс

Для начала операции программного сброса необходимо в регистр RTC_SW_RST записать 0x1. Этот процесс может занять некоторое время. По окончании сброса значение в регистре RTC_SW_RST будет равно 0x0.

Программный сброс сбрасывает все регистры в те же значения, что и аппаратный.

Запись в регистр RTC_SW_RST возможна только с помощью secure APB-транзакции. Запись любого значения в данный бит, пока он установлен в единице, игнорируется.

1.3.13.6.5 Защита от записи.

После аппаратного сброса APB-домена запись во все регистры запрещена (игнорируется). Для разблокирования записи необходимо записать в регистр RTC_WP константу 0x5FAC. После этого в регистре RTC_WP будет находится 0x1 (защита на запись снята). Обратное включение защиты от записи происходит только после аппаратного сброса.

1.3.13.6.6 Делитель тактового сигнала

Основная логика часов реального времени (счетчики) работает от тактового сигнала с частотой 1 Гц (clk_hz). Для его генерации служит внутренний дробный делитель. Он делит внешний тактовый сигнал rtc_clk_i по следующей формуле:

$$F_{clk_hz} = \frac{F_{rtc_clk_i}}{div + \frac{divf}{2^{16}}},$$ где

div - значение поля RTC_PSR[DIV_INT];

divf - значение поля RTC_PSR[DIV_FRAC].

1.3.13.6.7 Инициализация RTC

Инициализация RTC происходит следующим образом:

– проверить, что RTC выключен (бит RTC_EN[EN] равен нулю);

– установить начальное время с помощью регистров RTC_TIME и RTC_DATE;

– настроить делитель RTC_PSR в соответствии с частотой rtc_clk_i;

– установить запрос на включение RTC (записать в RTC_EN[EN] единицу);

– дождаться включения RTC, бит RTC_EN[EN] должен стать равным единице. Задержка на включение и выключение составляет до трех тактов rtc_clk_i с момента окончания транзакции по APB.

Во время того, как RTC включен, запись в регистры RTC_TIME, RTC_DATE, RTC_CTRL запрещена.

Во время того, как RTC выключен значение внутренних счетчиков делителя prescaler останавливается, а после включения счет продолжается с того же места, если не был перезаписан регистр RTC_PSR. Если регистр RTC_PSR был перезаписан, пока RTC находился в выключенном состоянии, то prescaler начнет считать с начала.

1.3.13.6.8 Будильники

В RTC есть возможность устанавливать два независимых будильника A и B, каждый из которых может генерировать событие alarm_a и alarm_b соответственно. Все, что ниже описано для будильника A, будет по аналогии работать и для будильника B.

Будильник срабатывает, когда значение в регистрах RTC_ALARM_TIME_A и RTC_ALARM_DATE_A полностью или частично совпадают с текущим временем и датой. В таб. 1.661 указаны режимы совпадения будильника. В ней "1" обозначает, что необходимо совпадение между полями будильника и текущими временем и датой, а "-" обозначает, что данные поля не имеют значения (по 2 поля отвечают за секунды, минуты и т.д.).

Таблица 1.661 – Режимы будильников

Значение поля регистра RTC_ALARM_CTRL_A[ALARM_MODE_A]	**Секунды**	**Минуты**	**Часы**	**День**	**Месяц**	**День недели**	**Пояснение**
000	1	1	1	1	1	-	Точное время и дата
001	1	1	1	-	-	1	Определенный день недели
010	1	1	1	1	-	-	Каждый месяц в один и тот же день
011	1	1	1	-	-	-	Каждый день в одно и тоже время

Значение поля регистра RTC_ALARM_CTRL_A[ALARM_MODE_A]	Секунды	Минуты	Часы	День	Месяц	День недели	Пояснение
100	1	1	-	-	-	-	Каждый час
101	1	-	-	-	-	-	Каждую минуту
110	-	-	-	-	-	-	Каждую секунду

Будильник A начинает работать только, когда RTC включен и бит RTC_ALARM_CTRL_A[ALARM_EN_A] установлен.

Настройка будильника A осуществляется следующим образом:

– проверить, что будильник выключен (бит RTC_ALARM_CTRL_A[ALARM_EN_A] равен нулю);

– установить время и дату будильника (RTC_ALARM_DATE_A и RTC_ALARM_TIME_A);

– настроить режим будильника (RTC_ALARM_CTRL_A[ALARM_MODE_A]);

– включить будильник, записав в бит RTC_ALARM_CTRL_A[ALARM_EN_A] единицу. Задержка на включение составляет до трех тактов rtc_clk_i после окончания транзакции. Процесс включения закончится, когда бит RTC_ALARM_CTRL_A[ALARM_EN_A] установится в 1.

При перенастройке будильника необходимо, чтобы он был выключен. Процесс выключения также занимает до трех тактов rtc_clk_i, а по окончанию бит RTC_ALARM_CTRL_A[ALARM_EN_A] установится в нуль.

Для подробного объяснения задержек на включение/выключение и синхронизации между APB-доменом и RTC-доменом см. пп 1.3.13.6.13.

Событие alarm_a можно отслеживать с помощью бита RTC_IRQ_STATUS[ALARM_A]. Этот статусный бит устанавливается в момент возникновения событий или когда происходит включение APB-домена, при установленном событии. По событию alarm_a выход rtc_alarm_a_o устанавливается в единицу пока не будет сброшен программно.

Сброс прерываний и событий осуществляется происходит побитово с помощью регистра RTC_IRQ_CLEAR. Сброс прерываний происходит сразу после записи в RTC_IRQ_CLEAR регистр, а сброс событий с задержкой в два такта rtc_clk_i (см. рис. 1.161).

1.3.13.6.9 Временные метки

Временные метки (событие timestamp) — это событие по которому происходит сохранение текущего времени и даты в регистры RTC_TS_DATE и RTC_TS_TIME. Данное событие вызывается другими событиями: rtc_vbat_ent (спад входа rtc_vbat_ent_i) и/или rtc_tamper_ent (фронт входа rtc_tamper_ent_i). Первое — это переход в режим VBAT, а второе — это сигнал от TAMP. С помощью полей VBAT_EN и TAMP_EN регистра RTC_CTRL настраивается, по каким из событий будет сохраняться временная метка. Временные метки проставляются только когда RTC_CTRL[EN] равен 1.

В процессе работы может возникнуть ситуация, когда в процессе включения RTC домена вход rtc_tamper_ent_i уже находится в единице. В такой ситуации произойдет установка события rtc_tamp_ent. Событие rtc_vbat_ent также установится, если в момент включения RTC вход rtc_vbat_ent_i находится в нуле.

Отследить валидность временных меток можно с помощью бита RTC_IRQ_STATUS[TS]. Он устанавливается, когда значение временной метки копируется из RTC-домена в APB, а сбрасывается после записи единицы в RTC_IRQ_CLEAR[TS]. После этого старая метка считается прочитанной, и аппаратуре разрешается перезаписывать новую. Это означает, что временная метка не обновляется, пока не будет сброшен RTC_IRQ_STATUS[TS] бит.

1.3.13.6.10 Wakeup событие

Данное событие позволяет перевести микроконтроллер из одного из режимов низкого энергопотребления в RUN режим (см. таб. 1.175). Wakeup может произойти по следующим событиям: rtc_tamper_ent, alarm_a и alarm_b. Выбор источника wakeup события осуществляется с помощью битов RTC_CTRL[WAKEUP_SEL].

После того, как происходит wakeup событие, бит RTC_IRQ_STATUS[WAKEUP] устанавливается. Данный бит сбросится после записи в RTC_IRQ_CLEAR[WAKEUP] единицы.

1.3.13.6.11 Поддержка TrustZone

В технологии TrustZone определяются два вида (типа) программного доступа к регистрам: secure и non-secure. Тип доступа определяется по сигналу rtc_pprot_i. В данном блоке только для регистров и отдельных полей, касающихся будильников, имеет значение тип программного доступа. В таб. 1.662 приведен список полей регистров и какой тип доступа как завершается.

Таблица 1.662 – Тип доступа регистров

Название регистров	Поле	Тип доступа - результат доступа
RTC_WP	WP	- Secure write - успешная запись - Secure read - успешное чтение - Non-secure write - запись игнорируется - Non-secure read - успешное чтение
RTC_SW_RST	SW_RST	- Secure write - успешная запись - Secure read - успешное чтение - Non-secure write - запись игнорируется - Non-secure read - успешное чтение
RTC_CTRL	SECURE_ALARM_x	- Secure write - успешная запись - Secure read - успешное чтение - Non-secure write - запись игнорируется - Non-secure read - успешное чтение
RTC_ALARM_CTRL_x, RTC_ALARM_TIME_x, RTC_ALARM_DATE_x	—	Если RTC_CTRL[SECURE_ALARM_x] равен 0: - Secure write - успешная запись - Secure read - успешное чтение - Non-secure write - запись игнорируется - Non-secure read - читаются нули Если RTC_CTRL[SECURE_ALARM_x] установлен в 1: - Secure write - успешная запись - Secure read - успешное чтение - Non-secure write - успешная запись - Non-secure read - успешное чтение
RTC_IRQ_MASK	ALARM_x	Если RTC_CTRL[SECURE_ALARM_x] равен 0: - Secure write - успешная запись - Secure read - успешное чтение - Non-secure write - запись игнорируется - Non-secure read - успешное чтение Если RTC_CTRL[SECURE_ALARM_x] установлен в 1: - Secure write - успешная запись - Secure read - успешное чтение - Non-secure write - успешная запись - Non-secure read - успешное чтение
RTC_IRQ_CLEAR	ALARM_x	Если RTC_CTRL[SECURE_ALARM_x] равен 0: - Secure write - успешная запись - Non-secure write - запись игнорируется Если RTC_CTRL[SECURE_ALARM_x] установлен в 1: - Secure write - успешная запись - Non-secure write - успешная запись

1.3.13.6.12 Корректировка RTC

Точность часов реального времени зависит от точности источника тактового сигнала. Изначально предполагается, что частота опорного сигнала LSE составляет 32768 Гц. Однако, кварцевый резонатор имеет некую погрешность при изготовлении, которая приводит к изменению частоты. Данное отклонение носит статический характер, то есть не изменяется со

временем. Так же существует и другой тип отклонения — динамический. Это когда частота кварцевого резонатора изменяется в процессе работы. Основной причиной такого изменения служит колебание температуры.

Для определения отклонения частоты кварцевого резонатора служит выходной сигнал rtc_calib_o. Он представляет из себя деленный LSE сигнал. Частота выходного сигнала задается в битовом поле RTC_CTRL[CALIB_MODE] и может составлять: 512, 256 и 1 Гц.

Корректировка RTC осуществляется путем изменения делителей (RTC_PSR). По отклонению частоты rtc_calib_o от ожидаемой можно рассчитать новые значения делителей. Действительное значение опорной частоты LSE определяется по формуле: $f_{lse} = \frac{f_{calib} \cdot 32768}{f_{calib_r}}$. Далее уже для нового рассчитанного значения опорной частоты определяются коэффициенты делителя по формуле описанной в пп 1.3.13.6.6.

Например, измеренная частота сигнала rtc_calib_o составила 512.0376 Гц. Тогда действительная частота LSE составляет $f_{lse} = \frac{512.0376 \cdot 32768}{512} = 32770.4064$. Для такой опорной частоты необходимо установить делители $div = 32770$ и $divf = ceil(0.4064 \cdot 2^{16}) = 26634$.

Для динамической (температурной) компенсации частоты RTC используется изменение делителей без остановки RTC. После записи в регистр RTC_PSR запускается процесс копирования новых значений в CORE домен. Статус выполнения этой операции отслеживается в регистре RTC_PSR_UPDATE. Пока идет процесс копирования, то RTC_PSR_UPDATE равен единице, а RTC продолжает работать.

Флаг RTC_PSR_UPDATE всегда устанавливается, когда происходит запись в регистр RTC_PSR. Сброс происходит аппаратно, после завершения обновления значений делителей в core-домене.

1.3.13.6.13 Синхронизация между доменами

На рис. 1.161 представлена схема синхронизации между APB-доменом и RTC-доменом в блоке RTC. На ней, в виде прямоугольников, изображены регистры, которые используют синхронизацию между доменами, а для уменьшения изображения в названии регистров опущен префикс "RTC".

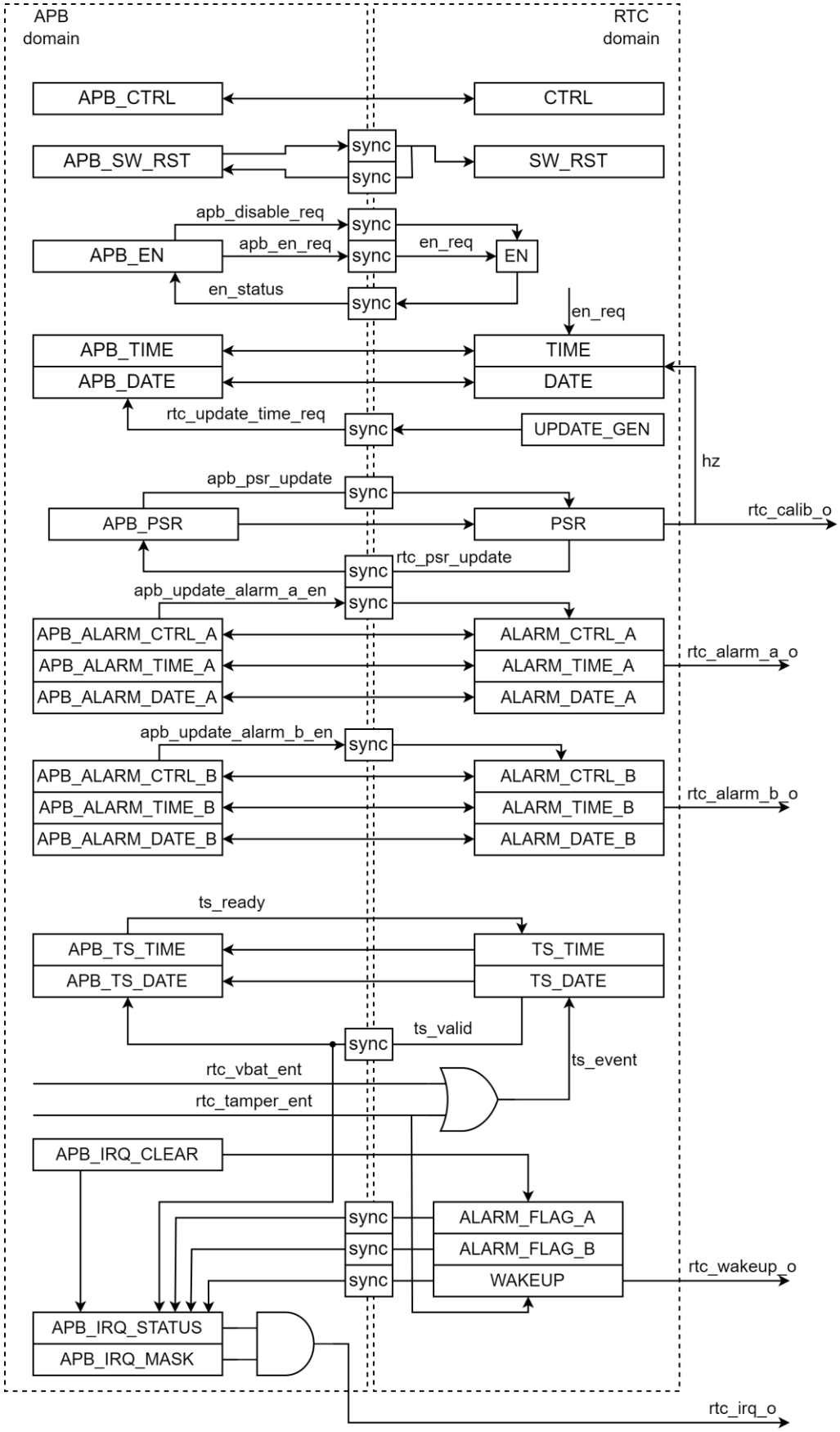
Так как большую часть времени работы данного блока предполагается, что APB-домен отключен, то все основные регистры находятся в RTC-домене. Доступ к ним осуществляется через теневые копии, которые находятся в APB-домене и имеют соответствующий префикс "APB". Из-за этого некоторые регистры имеют следующие особенности в работе:

– Временные регистры (RTC_TIME, RTC_DATE). Запись в эти регистры возможна только, когда RTC находится в выключенном состоянии. Когда RTC включается, то значения из теневых копий APB-домена копируются в рабочие в RTC-домен. В процессе работы значение текущего времени изменяется, поэтому, когда это происходит, рабочие регистры из RTC-домена копируются в теневые в APB-домен по сигналу rtc_update_time_req от генератора обновления времени UPDATE_GEN.

– Регистры будильников (RTC_ALARM_CTRL_x, RTC_ALARM_TIME_x, RTC_ALARM_DATE_x). Во время записи в регистры RTC_ALARM_DATE_x и RTC_ALARM_TIME_x записываемое значение помещается в теневую копию в APB-домене. При включении будильника, установки бита RTC_ALARM_CTRL_x[ALARM_EN_x], происходит копирование из теневых копий в рабочие. Так как в процессе работы может возникнуть ситуация, когда APB-домен будет выключен, а RTC-домен продолжит работать, то из-за этого произойдет расхождение между теневыми и рабочими регистрами. Для того чтобы этого не допустить, после включения APB-домена (входной сигнал rtc_apb_pw_i установился в единицу) данные из рабочих регистров RTC_ALARM_x_x копируются в теневые.

– Регистры временных меток (RTC_TS_x). Данные регистры доступны только на чтение. Копирование из рабочих в теневые происходит с помощью механизма handshake. Со стороны APB-домена выходит сигнал ts_ready. Он установлен, когда в теневых регистрах нет данных, которые не считывали по APB. Со стороны рабочих регистров выходит сигнал ts_valid. Он установлен, когда в рабочих регистрах есть нескопированные данные. Поэтому, если в рабочих регистрах RTC_TS_x хранится временная метка, которая еще не была прочитана, то новая не будет перезаписана (утеряна).

– Регистр включения (RTC_EN). Данный регистр включает/выключает RTC. Эта операция занимает до 3 тактов rtc_clk_i. После завершения операции бит RTC_EN[EN] устанавливается в то значение, которое было в него записано. В период включения или выключения запись в RTC_EN игнорируется. При включении APB-домена (вход rtc_pclk_enable_i устанавливается в единицу) происходит копирование из рабочего регистра (EN) в теневой (APB_EN), тем самым актуализируется информация о том, что RTC включен или выключен.



					ЮФКВ.431268.044РЭ	Лист 911
Изм.	Лист	№ докум.	Подп.	Дата		

Инв.№подл.	Подп. и дата	Взам.инв.№	Инв.№дубл.	Подп. и дата
47472	30.09.2024			

Рисунок 1.161 – Схема синхронизации

1.3.13.6.14 Восстановление тактового сигнала в APB-домене

В RTC блоке предусмотрена возможность отключения APB-домена. При этом RTC-домен может оставаться включенным. В таком случае после включения APB-домена необходимо, чтобы в нем обновились теневые регистры в соответствии с рабочими из RTC-домена. Этот процесс изображен на рис. 1.162. В момент времени 1, до RTC-домена доходит пересинхронизованный сигнал pclk_enable_i, который указывает, что APB-домен включился (apb_domain_pw). Одновременно с этим выставляется сигнал rtc_upload_config, который идет в APB-домен через синхронизатор. В момент времени 2, этот сигнал доходит и происходит процесс копирования. Копирования подвержены следующие регистры: RTC_CTRL, RTC_PST, RTC_TIME, RTC_DATE, RTC_ALARM_CTRL_x, RTC_ALARM_TIME_x, RTC_ALARM_DATE_x.

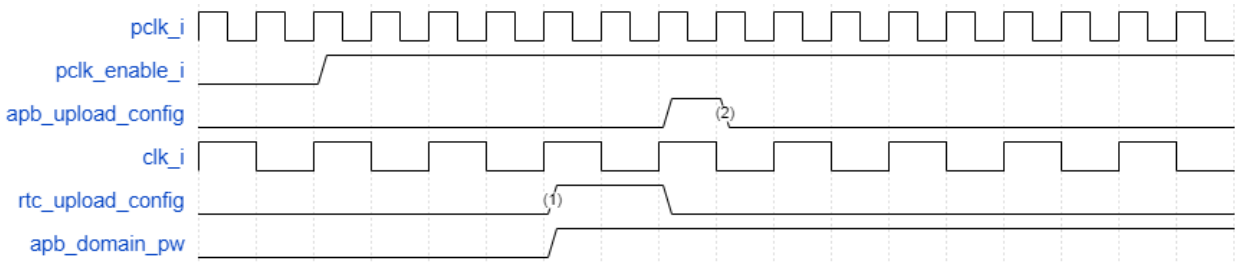


Рисунок 1.162 – Включение APB-домена

1.3.13.6.15 Прерывания

В RTC прерывания реализованы в APB-домене, поэтому они работают только, ABP-домен включен. Ниже приведен список прерываний:

– ALARM_A. Устанавливается при возникновении события alarm_a с задержкой в один такт rtc_clk_i. Сбрасывается программно после записи 1 в бит RTC_IRQ_CLEAR[ALARM_A];

– ALARM_B. Устанавливается при возникновении события alarm_b с задержкой в один такт rtc_clk_i. Сбрасывается программно после записи 1 в бит RTC_IRQ_CLEAR[ALARM_B];

– TS. Устанавливается при возникновении события timestamp. Сбрасывается программно после записи 1 в бит RTC_IRQ_CLEAR[TS];

– WAKEUP. Устанавливается при возникновении события wakeup. Сбрасывается программно после записи 1 в бит RTC_IRQ_CLEAR[WAKEUP];

– TEST. Устанавливается после записи в регистр RTC_IRQ_TEST единицы. Сбрасывается программно после запись 1 в бит RTC_IRQ_CLEAR[TEST];

Прерывания ALARM_A и ALARM_B сбрасываются только, если доступ к регистру RTC_IRQ_CLEAR осуществлялся с необходимым secure доступом (см. пп 1.3.13.6.11).

Выходной сигнал прерывания rtc_irq_o установлен пока установлено хотя бы одно незамаскированное прерывание в регистре RTC_IRQ_STATUS.

1.3.13.6.16 Интеграция RTC

Контроллер разделен на два тактовых домена: apb-домен и core-домен. Это сделано для того, что бы при изменении системной частоты core-домен, выполняющий передачу данных на шине, не надо было перенастраивать.

У контроллера на входе стоит специальный блок gp_apb_rejector. Его назначение заключается в корректном завершении apb-транзакций, пока с RTC контроллера снято тактирование. В зависимости от входа conf_pslverr_i транзакции могут завершаться с ошибкой или без. Если conf_pslverr_i = 1, то генерируется ошибка. При этом все транзакции записи игнорируются, а чтение возвращает 0x0BAD_ACCE5.

При обращении к RTC по шине APB младшие два бита (нулевой и первый) адреса не анализируются. APB транзакция на запись с pstrb не равным 4’b1111 возвращается с ошибкой (pslverr = 1) и никакие данные не записываются. Запись по зарезервированным адресам игнорируется, а чтение возвращает 0xEEFFADDC. При этом транзакции завершаются без ошибок. Так как младшие два бита не анализируются, то обращения по не выровненным адресам автоматически выравниваются.

1.3.13.6.17 Описание регистров

1.3.13.6.17.1 Общее описание регистров

Контроллер имеет набор программно доступных регистров, требующий 4 КБайт адресов на системной шине AHB.

Карта регистров приведена в таблице 1.663.

Адреса, который не указаны в таблице 1.663 считаются резервными. Чтение из них возвращает 0xEEFFADDC, а запись игнорируется. rtc контроллер не различает secure и non-secure транзакции.

В регистры RTC_CTRL, RTC_TIME, RTC_DATE запрещена запись, когда RTC включен или находится в процессе включения. Такая запись может привести к некорректной работе.

В регистры будильников RTC_ALARM_TIME_x, RTC_ALARM_DATE_x и все поля RTC_ALARM_CTRL_x, кроме ALARM_EN_x, запрещена запись пока данный будильник включен.

Запись в регистры RTC_TIME и RTC_DATE некорректного времени или даты приведет к некорректной работе.

Данный блок поддерживает технологию TrustZone, поэтому тип доступа у регистров разный (см. раздел 1.3.13.6.11).

1.3.13.6.17.2 Карта регистров

Таблица 1.663 – Карта регистров RTC

Адрес	Название	Доступ	Значение после сброса	Описание
0x000	RTC_ID	ЧТ	0x00435452	RTC ID
0x004	RTC_VERSION	ЧТ	0x0	Версия RTC
0x008	RTC_SW_RST	ЧТ/ЗП	0x0	Программный сброс
0x00C	RTC_IRQ_STATUS	ЧТ	0x0	Статус прерываний
0x010	RTC_IRQ_MASK	ЧТ/ЗП	0x0	Маска прерываний
0x014	RTC_IRQ_CLEAR	ЗП	0x0	Сброс прерываний
0x018	RTC_IRQ_TEST	ЗП	0x0	Установка TEST прерывания
0x01C	RTC_WP	ЧТ/ЗП	0x0	Защита записи
0x020	RTC_EN	ЧТ/ЗП	0x0	Включение и выключение RTC
0x024	RTC_PSR_UPDATE	ЧТ	0x0	Статус обновления RTC_PSR регистра
0x100	RTC_CTRL	ЧТ/ЗП	0x0	Настройка RTC
0x104	RTC_PSR	ЧТ/ЗП	0x8000	Делитель
0x108	RTC_TIME	ЧТ/ЗП	0x0	Текущее время
0x10C	RTC_DATE	ЧТ/ЗП	0x101	Текущая дата
0x110	RTC_ALARM_CTRL_A	ЧТ/ЗП	0x0	Настройка будильника A
0x114	RTC_ALARM_TIME_A	ЧТ/ЗП	0x0	Время будильника A
0x118	RTC_ALARM_DATE_A	ЧТ/ЗП	0x0	Дата будильника A
0x11C	RTC_ALARM_CTRL_B	ЧТ/ЗП	0x0	Настройка будильника B
0x120	RTC_ALARM_TIME_B	ЧТ/ЗП	0x0	Время будильника B
0x124	RTC_ALARM_DATE_B	ЧТ/ЗП	0x0	Дата будильника B
0x128	RTC_TS_TIME	ЧТ	0x0	Время временной метки
0x12C	RTC_TS_DATE	ЧТ	0x0	Дата временной метки

1.3.13.6.17.3 Описание полей регистров

1.3.13.6.17.3.1RTC_ID (0x000)

Регистр идентификатора RTC контроллера

В таблице 1.664 представлено описание полей регистра.

Таблица 1.664 – Описание полей регистра RTC_ID

Биты	Поле	Доступ	Значение после сброса	Описание
31:0	ID	ЧТ	0x00435452	Идентификатор контроллера RTC

1.3.13.6.17.3.2RTC_VERSION (0x004)

Регистр версии RTC контроллера

В таблице 1.665 представлено описание полей регистра.

Таблица 1.665 – Описание полей регистра RTC_VERSION

Биты	Поле	Доступ	Значение после сброса	Описание
31:0	ID	ЧТ	0	Версия контроллера RTC

1.3.13.6.17.3.3RTC_SW_RST (0x008)

Регистр программного сброса RTC контроллера

В таблице 1.666 представлено описание полей регистра.

Таблица 1.666 – Описание полей регистра RTC_SW_RST

Биты	Поле	Доступ	Значение после сброса	Описание
0	SW_RST	ЧТ/ЗП	0	После установки в единицу начинается процесс программного сброса. Он заканчивается, когда этот бит сбросится в нуль аппаратно. Запись любого значения в данный бит, пока он установлен в единице, игнорируется. Запись происходит только secure-транзакцией.
31:1	-	ЧТ	0	Зарезервировано

1.3.13.6.17.3.4RTC_IRQ_STATUS (0x00C)

Регистр статуса прерываний RTC контроллера

В таблице 1.667 представлено описание полей регистра.

Таблица 1.667 – Описание полей регистра RTC_IRQ_STATUS

Биты	Поле	Доступ	Значение после сброса	Описание
0	TEST	ЧТ	0	Статус прерывания TEST. 0 - прерывание TEST сброшено. 1 - прерывание TEST установлено.
3:1	-	ЧТ	0	Зарезервировано

Биты	Поле	Доступ	Значение после сброса	Описание
4	ALARM_A	ЧТ	0	Статус события alarm_a. 0 - будильник A не сработал. 1 - будильник A сработал.
7:5	-	ЧТ	0	Зарезервировано
8	ALARM_B	ЧТ	0	Статус события alarm_b. 0 - будильник B не сработал. 1 - будильник B сработал.
11:9	-	ЧТ	0	Зарезервировано
12	TS	ЧТ	0	Статус события timestamp. 0 - события timestamp не было. 1 - событие timestamp произошло.
15:13	-	ЧТ	0	Зарезервировано
16	WAKEUP	ЧТ	0	Статус события wakeup. 0 - события wakeup не было. 1 - событие wakeup произошло.
31:17	-	ЧТ	0	Зарезервировано

1.3.13.6.17.3.5RTC_IRQ_MASK (0x010)

Регистр маска прерываний RTC контроллера

В таблице 1.668 представлено описание полей регистра.

Таблица 1.668 – Описание полей регистра RTC_IRQ_MASK

Биты	Поле	Доступ	Значение после сброса	Описание
0	TEST	ЧТ/ЗП	0	Маска прерывания TEST 0 - прерывание TEST запрещено. 1 - прерывания TEST разрешено.
3:1	-	ЧТ	0	Зарезервировано
4	ALARM_A	ЧТ/ЗП	0	Маска прерывания ALARM_A 0 - прерывание ALARM_A запрещено. 1 - прерывания ALARM_A разрешено.
7:5	-	ЧТ	0	Зарезервировано
8	ALARM_B	ЧТ/ЗП	0	Маска прерывания ALARM_B 0 - прерывание ALARM_B запрещено. 1 - прерывания ALARM_B разрешено.
11:9	-	ЧТ	0	Зарезервировано
12	TS	ЧТ/ЗП	0	Маска прерывания TS. 0 - прерывание TS запрещено. 1 - прерывания TS разрешено.
15:13	-	ЧТ	0	Зарезервировано
16	WAKEUP	ЧТ/ЗП	0	Маска прерывания WAKEUP 0 - прерывание WAKEUP запрещено. 1 - прерывания WAKEUP разрешено.
31:17	-	ЧТ	0	Зарезервировано

1.3.13.6.17.3.6RTC_IRQ_CLEAR (0x014)

Регистр сброса прерываний RTC контроллера

В таблице 1.669 представлено описание полей регистра.

Таблица 1.669 – Описание полей регистра RTC_IRQ_CLEAR

Биты	Поле	Доступ	Значение после сброса	Описание
0	TEST	ЗП	0	Запись единицы в данное поле сбрасывает прерывание TEST.
3:1	-	ЧТ	0	Зарезервировано
4	ALARM_A	ЗП	0	Запись единицы в данное поле сбрасывает прерывание ALARM_A и сигнал rtc_alarm_a_o.
7:5	-	ЧТ	0	Зарезервировано
8	ALARM_B	ЗП	0	Запись единицы в данное поле сбрасывает прерывание ALARM_B и сигнал rtc_alarm_b_o.
11:9	-	ЧТ	0	Зарезервировано
12	TS	ЗП	0	Запись единицы в данное поле сбрасывает прерывание TS.
15:13	-	ЧТ	0	Зарезервировано
16	WAKEUP	ЗП	0	Запись единицы в данное поле сбрасывает прерывание WAKEUP и сигнал rtc_wakeup_o.
31:17	-	ЧТ	0	Зарезервировано

1.3.13.6.17.3.7RTC_IRQ_TEST (0x018)

Регистр установки тестового прерывания RTC контроллера

В таблице 1.670 представлено описание полей регистра.

Таблица 1.670 – Описание полей регистра RTC_IRQ_TEST

Биты	Поле	Доступ	Значение после сброса	Описание
0	TEST	ЗП	0	Запись единицы в данный бит устанавливает прерывание TEST.
31:1	-	ЧТ	0	Зарезервировано

1.3.13.6.17.3.8RTC_EN (0x020)

Регистр включения RTC контроллера

В таблице 1.671 представлено описание полей регистра.

Таблица 1.671 – Описание полей регистра RTC_EN

Биты	Поле	До-ступ	Значение по-сле сброса	Описание
0	EN	ЧТ/ЗП	0	Запрос на включение/выключение RTC. Если данный бит равен нулю и в него была записана единица, то это вызовет запрос на включение. Когда RTC включится, то данный бит станет равным единице. С выключением все наоборот. В процессе включения/выключения запись в данный бит игнорируется. 0 - RTC выключен или находится в процессе включения. 1 - RTC включен или находится в процессе выключения.
31:1	-	ЧТ	0	Зарезервировано

1.3.13.6.17.3.9RTC_WP (0x01C)

Регистр блокировки доступа RTC контроллера

В таблице 1.672 представлено описание полей регистра.

Таблица 1.672 – Описание полей регистра RTC_WP

Биты	Поле	До-ступ	Значение по-сле сброса	Описание
15:0	WP	ЧТ/ЗП	0	Блокировка записи в регистры RTC. После аппаратного сброса запись во все регистры игнорируется, а чтение проходи успешно. Для разблокировки записи необходимо в это поле записать 0x5FAC. После этого запись будет разрешена и из данного поля будет считываться 1. Только аппаратный сброс APB-домена может сбросить данный бит, то есть установить защиту от записи. Когда запись заблокирована, то из данного поля считывается 0. 0 - запись в регистры RTC игнорируется. 1 - запись в регистры RTC не игнорируется.
31:16	-	ЧТ	0	Зарезервировано

1.3.13.6.17.3.10 RTC_PSR_UPDATE (0x024)

Регистр запроса на обновление делителей RTC контроллера

В таблице 1.673 представлено описание полей регистра.

Таблица 1.673 – Описание полей регистра RTC_PSR_UPDATE

Биты	Поле	До-ступ	Значение по-сле сброса	Описание
0	PSR_UPDATE	ЧТ/ЗП	0	Данный бит отображает статус обновление регистра RTC_PSR. 0 - Обновление регистра RTC_PSR завершено. 1 - Идет обновление регистра RTC_PSR.
31:1	-	ЧТ	0	Зарезервировано

1.3.13.6.17.3.11 RTC_CTRL (0x100)

Регистр настройки RTC контроллера

В таблице 1.674 представлено описание полей регистра.

Таблица 1.674 – Описание полей регистра RTC_CTRL

Биты	Поле	До-ступ	Значение по-сле сброса	Описание
0	TAMP_EN	ЧТ/ЗП	0	Сохранять текущее время и дату в RTC_TS_TIME и RTC_TS_DATE регистрах при событии rtc_tamper_ent. 0 - не сохранять. 1 - сохранять.
1	VBAT_EN	ЧТ/ЗП	0	Сохранять текущее время и дату в RTC_TS_TIME и RTC_TS_DATE регистрах при событии rtc_vbat_ent. 0 - не сохранять. 1 - сохранять.
2	SECURE_ALARM_A	ЧТ/ЗП	0	Устанавливает доступ в регистры будильника A: RTC_ALARM_CTRL_A, RTC_ALARM_TIME_A, RTC_ALARM_DATE_A, RTC_IRQ_MASK[ALARM_A], RTC_IRQ_CLEAR[ALARM_A]. Данное поле может быть изменено только secure транзакцией. 0 - запись в регистры будильника A возможна только secure транзакцией. 1 - запись в регистры будильника A возможна secure и non-secure транзакциями.
3	SECURE_ALARM_B	ЧТ/ЗП	0	Устанавливает доступ в регистры будильника B: RTC_ALARM_CTRL_B, RTC_ALARM_TIME_B, RTC_ALARM_DATE_B, RTC_IRQ_MASK[ALARM_B], RTC_IRQ_CLEAR[ALARM_B]. Данное поле может быть изменено только secure транзакцией. 0 - запись в регистры будильника B возможна только secure транзакцией. 1 - запись в регистры будильника B возможна secure и non-secure транзакциями.

Биты	Поле	Доступ	Значение после сброса	Описание
6:4	WAKEUP_SEL	ЧТ/ЗП	0	Устанавливает, какое из событий будет служить источником для wakeup: 000 - событие wakeup не происходит 001 - происходит по событию alarm_a 010 - происходит по событию alarm_b 011 - происходит по событию alarm_a или alarm_b 100 - происходит по событию rtc_tamper_ent 101 - происходит по событию rtc_tamper_ent или будильнику A 110 - происходит по событию rtc_tamper_ent или будильнику B 111 - происходит или по событию rtc_tamper_ent, или будильнику A, или по будильнику B
8:7	CALIB_MODE	ЧТ/ЗП	0	Выбор выходного калибровочного сигнала rtc_calib_o. 00 - калибровочный сигнал выключен. 01 - на выход rtc_calib_o подается сигнал с частотой 512 Гц. 10 - на выход rtc_calib_o подается сигнал с частотой 256 Гц. 11 - на выход rtc_calib_o подается сигнал с частотой 1 Гц.
31:9	-	ЧТ	0	Зарезервировано

1.3.13.6.17.3.12 RTC_PSR (0x104)

Регистр настройки делителя RTC контроллера

В таблице 1.675 представлено описание полей регистра.

Таблица 1.675 – Описание полей регистра RTC_PSR

Биты	Поле	Доступ	Значение после сброса	Описание
15:0	DIV_INT	ЧТ/ЗП	0x8000	Целая часть делителя тактового сигнала rtc_clk_i
31:16	DIV_FRAC	ЧТ/ЗП	0	Дробная часть делителя тактового сигнала rtc_clk_i

1.3.13.6.17.3.13 RTC_TIME (0x108)

Регистр настройки времени RTC контроллера

В таблице 1.676 представлено описание полей регистра.

Таблица 1.676 – Описание полей регистра RTC_TIME

Биты	Поле	Доступ	Значение после сброса	Описание
3:0	SU	ЧТ/ЗП	0	Единицы секунд в BCD формате
6:4	ST	ЧТ/ЗП	0	Десятки секунд в BCD формате

Биты	Поле	До-ступ	Значение по-сле сброса	Описание
7	-	ЧТ	0	Зарезервировано
11:8	MNU	ЧТ/ЗП	0	Единицы минут в BCD формате
14:12	MNT	ЧТ/ЗП	0	Десятки минут в BCD формате
15	-	ЧТ	0	Зарезервировано
19:16	HU	ЧТ/ЗП	0	Единицы часов в BCD формате
21:20	HT	ЧТ/ЗП	0	Десятки часов в BCD формате
31:22	-	ЧТ	0	Зарезервировано

1.3.13.6.17.3.14 RTC_DATE (0x10C)

Регистр настройки даты RTC контроллера

В таблице 1.677 представлено описание полей регистра.

Таблица 1.677 – Описание полей регистра RTC_DATE

Биты	Поле	До-ступ	Значение по-сле сброса	Описание
3:0	DU	ЧТ/ЗП	1	Единицы числа в BCD формате
5:4	DT	ЧТ/ЗП	0	Десятки числа в BCD формате
7:6	-	ЧТ	0	Зарезервировано
11:8	MU	ЧТ/ЗП	1	Единицы месяца в BCD формате
12	MT	ЧТ/ЗП	0	Десятки месяца в BCD формате
15:12	-	ЧТ	0	Зарезервировано
19:16	YU	ЧТ/ЗП	0	Единицы года в BCD формате
23:20	YT	ЧТ/ЗП	0	Десятки года в BCD формате
26:24	WEEK	ЧТ/ЗП	0	День недели: 000 - понедельник 001 - вторник 010 - среда 011 - четверг 100 - пятница 101 - суббота 110 - воскресенье 111 - резерв
31:7	-	ЧТ	0	Зарезервировано

1.3.13.6.17.3.15 RTC_ALARM_CTRL_A (0x110)

Регистр настройки будильника A RTC контроллера

В таблице 1.678 представлено описание полей регистра.

Таблица 1.678 – Описание полей регистра RTC_ALARM_CTRL_A

Биты	Поле	Доступ	Значение после сброса	Описание
0	ALARM_EN_A	ЧТ/ЗП	0	Разрешение генерации события alarm_a. Принцип работы данного бита аналогичен биту RTC_EN[EN]. После записи в данный бит необходимо дожидаться, когда он измениться на записанное значение. Если это не выполняется, то работоспособность не гарантируется. 0 - запрещено генерировать событие alarm_a 1 - разрешено генерировать событие alarm_a
3:1	-	ЧТ	0	Зарезервировано
6:4	ALARM_MODE_A	ЧТ/ЗП	0	Режим будильника A (см. таб. 1.661). 000 - Точное время и дата 001 - Определенный день недели 010 - Каждый месяц в один и тот же день 011 - Каждый день в одно и тоже время 100 - Каждый час 101 - Каждую минуту 110 - Каждую секунду 111 - резерв
31:7	-	ЧТ	0	Зарезервировано

1.3.13.6.17.3.16 RTC_ALARM_TIME_A (0x114)

Регистр времени будильника A RTC контроллера

В таблице 1.679 представлено описание полей регистра.

Таблица 1.679 – Описание полей регистра RTC_ALARM_TIME_A

Биты	Поле	Доступ	Значение после сброса	Описание
3:0	SU	ЧТ/ЗП	0	Единицы секунд в BCD формате
6:4	ST	ЧТ/ЗП	0	Десятки секунд в BCD формате
7	-	ЧТ	0	Зарезервировано
11:8	MNU	ЧТ/ЗП	0	Единицы минут в BCD формате
14:12	MNT	ЧТ/ЗП	0	Десятки минут в BCD формате
19:16	HU	ЧТ/ЗП	0	Единицы часов в BCD формате
21:20	HT	ЧТ/ЗП	0	Десятки часов в BCD формате
31:22	-	ЧТ	0	Зарезервировано

1.3.13.6.17.3.17 RTC_ALARM_DATE_A (0x118)

Регистр даты будильника A RTC контроллера

В таблице 1.680 представлено описание полей регистра.

Таблица 1.680 – Описание полей регистра RTC_ALARM_DATE_A

Биты	Поле	До-ступ	Значение по-сле сброса	Описание
3:0	DU	ЧТ/ЗП	0	Единицы числа в BCD формате
5:4	DT	ЧТ/ЗП	0	Десятки числа в BCD формате
11:8	MU	ЧТ/ЗП	0	Единицы месяца в BCD формате
12	MT	ЧТ/ЗП	0	Десятки месяца в BCD формате
23:13	-	ЧТ	0	Зарезервировано
26:24	WEEK	ЧТ/ЗП	0	День недели: 000 - понедельник 001 - вторник 010 - среда 011 - четверг 100 - пятница 101 - суббота 110 - воскресенье 111 - резерв
31:27	-	ЧТ	0	Зарезервировано

1.3.13.6.17.3.18 RTC_ALARM_CTRL_B (0x11C)

Регистр настройки будильника B RTC контроллера

В таблице 1.681 представлено описание полей регистра.

Таблица 1.681 – Описание полей регистра RTC_ALARM_CTRL_B

Биты	Поле	До-ступ	Значение по-сле сброса	Описание
0	ALARM_EN_B	ЧТ/ЗП	0	Разрешение генерации события alarm_b. Принцип работы данного бита аналогичен биту RTC_EN[EN]. После записи в данный бит необходимо дожидаться, когда он измениться на записанное значение. Если это не выполняется, то работоспособность не гарантируется. 0 - запрещено генерировать событие alarm_b 1 - разрешено генерировать событие alarm_b
3:1	-	ЧТ	0	Зарезервировано
6:4	ALARM_MODE_B	ЧТ/ЗП	0	Режим будильника B (см. таб. 1.661). 000 - Точное время и дата 001 - Определенный день недели 010 - Каждый месяц в один и тот же день 011 - Каждый день в одно и тоже время 100 - Каждый час 101 - Каждую минуту 110 - Каждую секунду 111 - резерв
31:7	-	ЧТ	0	Зарезервировано

1.3.13.6.17.3.19 RTC_ALARM_TIME_B (0x120)

Регистр времени будильника B RTC контроллера

В таблице 1.682 представлено описание полей регистра.

Таблица 1.682 – Описание полей регистра RTC_ALARM_TIME_B

Биты	Поле	До-ступ	Значение по-сле сброса	Описание
3:0	SU	ЧТ/ЗП	0	Единицы секунд в BCD формате
6:4	ST	ЧТ/ЗП	0	Десятки секунд в BCD формате
7	-	ЧТ	0	Зарезервировано
11:8	MNU	ЧТ/ЗП	0	Единицы минут в BCD формате
14:12	MNT	ЧТ/ЗП	0	Десятки минут в BCD формате
19:16	HU	ЧТ/ЗП	0	Единицы часов в BCD формате
21:20	HT	ЧТ/ЗП	0	Десятки часов в BCD формате
31:22	-	ЧТ	0	Зарезервировано

1.3.13.6.17.3.20 RTC_ALARM_DATE_B (0x124)

Регистр даты будильника B RTC контроллера

В таблице 1.683 представлено описание полей регистра.

Таблица 1.683 – Описание полей регистра RTC_ALARM_DATE_B

Биты	Поле	До-ступ	Значение по-сле сброса	Описание
3:0	DU	ЧТ/ЗП	0	Единицы числа в BCD формате
5:4	DT	ЧТ/ЗП	0	Десятки числа в BCD формате
11:8	MU	ЧТ/ЗП	0	Единицы месяца в BCD формате
12	MT	ЧТ/ЗП	0	Десятки месяца в BCD формате
23:13	-	ЧТ	0	Зарезервировано
26:24	WEEK	ЧТ/ЗП	0	День недели: 000 - понедельник 001 - вторник 010 - среда 011 - четверг 100 - пятница 101 - суббота 110 - воскресенье 111 - резерв
31:27	-	ЧТ	0	Зарезервировано

1.3.13.6.17.3.21 RTC_TS_TIME (0x128)

Регистр времени временной метки RTC контроллера

В таблице 1.684 представлено описание полей регистра.

Таблица 1.684 – Описание полей регистра RTC_TS_TIME

Биты	Поле	До-ступ	Значение по-сле сброса	Описание
3:0	SU	ЧТ	0	Единицы секунд временной метки в BCD формате
6:4	ST	ЧТ	0	Десятки секунд временной метки в BCD формате
7	-	ЧТ	0	Зарезервировано
11:8	MNU	ЧТ	0	Единицы минут временной метки в BCD формате
14:12	MNT	ЧТ	0	Десятки минут временной метки в BCD формате
15	-	ЧТ	0	Зарезервировано
19:16	HU	ЧТ	0	Единицы часов временной метки в BCD формате
21:20	HT	ЧТ	0	Десятки часов временной метки в BCD формате
31:22	-	ЧТ	0	Зарезервировано

1.3.13.6.17.3.22 RTC_TS_DATE (0x12C)

Регистр даты временной метки RTC контроллера

В таблице 1.685 представлено описание полей регистра.

Таблица 1.685 – Описание полей регистра RTC_TS_DATE

Биты	Поле	До-ступ	Значение по-сле сброса	Описание
3:0	DU	ЧТ	0	Единицы числа временной метки в BCD формате
5:4	DT	ЧТ	0	Десятки числа временной метки в BCD формате
7:6	-	ЧТ	0	Зарезервировано
11:8	MU	ЧТ	0	Единицы месяца временной метки в BCD формате
12	MT	ЧТ	0	Десятки месяца временной метки в BCD формате
19:16	YU	ЧТ	0	Единицы года временной метки в BCD формате
23:20	YT	ЧТ	0	Десятки года временной метки в BCD формате
26:24	WEEK	ЧТ	0	День недели: 000 - понедельник 001 - вторник 010 - среда 011 - четверг 100 - пятница 101 - суббота 110 - воскресенье 111 - резерв
31:27	-	ЧТ	0	Зарезервировано

1.3.14 Подсистема аналоговых блоков

1.3.14.1 Введение

Подсистема аналоговых блоков представляет собой совокупность аналоговых блоков различного назначения, объединенных общим доменом аналогового питания и имеющих

общую системную шину управления по спецификации AHB5. Упрощенная схема подсистемы аналоговых блоков представлена на рисунке 1.163.

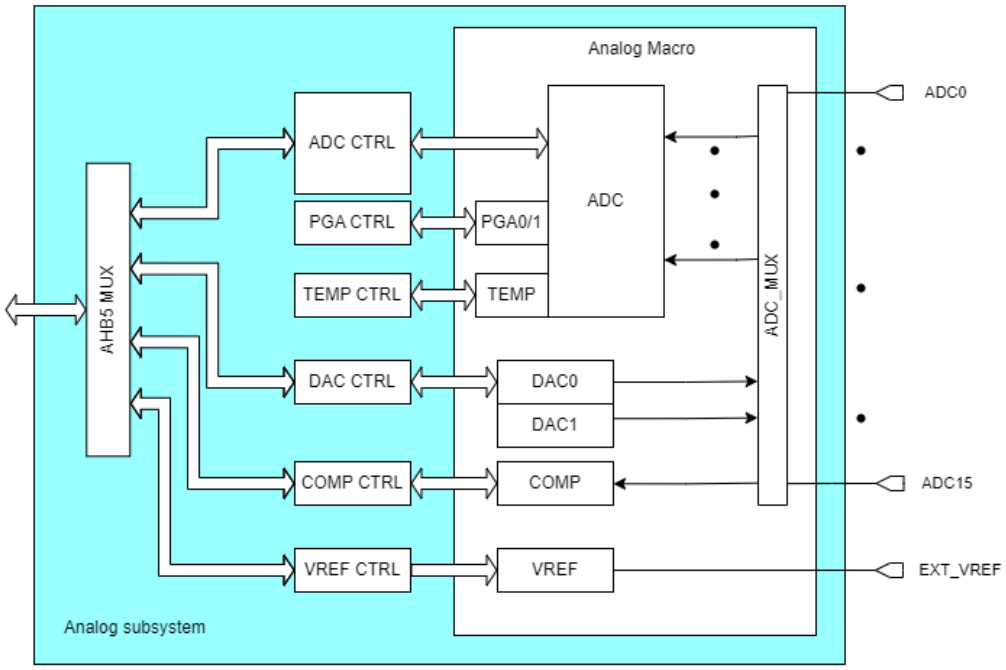


Рисунок 1.163 – Упрощеннная схема подсистемы аналоговых блоков

1.3.14.2 Состав Подсистема аналоговых блоков

Подсистема аналоговых блоков состоит из:

1) Аналогового макроблока Analogmacro, представляющего из себя комплексную сборку аналоговых блоков (см. раздел),

2) Цифровых блоков управления аналоговой подсистемой, в том числе:

a) Мультиплексора шины AHB5

b) Блока управления АЦП ADCCTRL (см. раздел 1.3.14.6.2), совмещенного с блоками управления ОУ PGACTRL и термодатчика TSENSCTRL

c) Блока управления ЦАП DACCTRL (см. раздел 1.3.14.7.2)

d) Блока управления аналоговым компаратором ACOMPCTRL (см. раздел 1.3.14.9.1)

e) Блока управления ИОН VREFCTRL (см. раздел 1.3.14.8.1)

1.3.14.3 Основные характеристики подсистемы аналоговых блоков

1) Каждый канал АЦП имеет переключаемые положительные и отрицательные входы, используемые для выбора источника сигнала:

1.3.14.4 Основные параметры подсистемы аналоговых блоков

В таблице 1.686 приведены основные параметры подсистемы аналоговых блоков

Таблица 1.686 – Основные параметры компараторов

Параметр	Мин.	Тип.	Макс.	Единицы
Напряжение аналогово питания	1,62		3,6	В
Напряжение цифрового питания	0,81		1.21	В
Ток потребления				мкА
Ток потребления в спящем режиме				мкА

1.3.14.5 Структурная схема подсистемы аналоговых блоков

1.3.14.5.1 Введение

Все аналоговые блоки подсистемы аналоговых блоков объединены в единый аналоговый макроблок, характеризующийся отдельным доменом питания и компактным расположением аналоговых компонентов. Блок представляет собой совокупность аналоговых компанентов. Упрощенная схема аналогового макроблока представлена на рисунке 1.164.

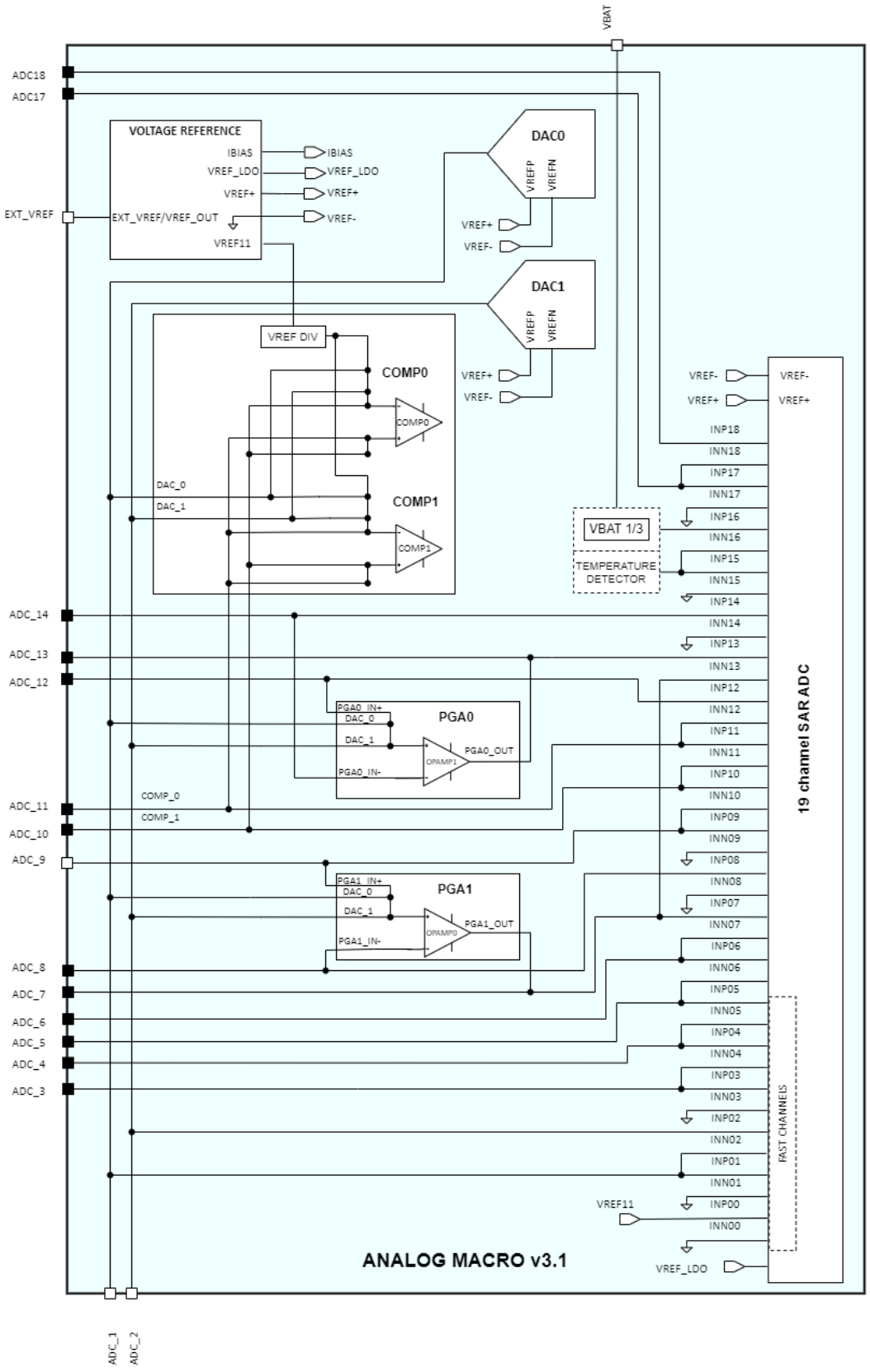


Рисунок 1.164 – Упрощеннная схема аналогового макроблока

Макроблок состоит из . Блока аналого-цифрового преобразователя АЦП . Двух блоков цифро-аналоговых преобразователей ЦАП . Блока аналоговых компараторов . Двух блоков операционных усилителей с программируемым коэффициентом усиления . Блока термодатчика . Блока источника опорного напряжения

Все блоки аналоговой подсистемы описаны в разделах далее.

1.3.14.5.2 Основные характеристики аналогового макроблока

1) Каждый компаратор имеет переключаемые положительные и отрицательные входы, используемые для выбора источника сигнала:

а) Мультиплексные контакты ввода-вывода

1.3.14.5.3 Основные параметры аналогового макроблока

В таблице 1.687 приведены основные параметры аналогового макроблока.

Таблица 1.687 – Основные параметры аналогового макроблока

Параметр	Мин.	Тип.	Макс.	Единицы
Напряжение аналогово питания	1,62		3,6	В
Напряжение цифрового питания	0,81		1.21	В
Ток потребления			ХЗ	мкА
Ток потребления в спящем режиме			ХЗ	мкА

1.3.14.5.4 Внешние выводы аналогового макроблока

В таблице 1.688 приведены внешние выводы аналогового макроблока.

Таблица 1.688 – внешние выводы аналогового макроблока

Название	Функции	Расшифровка	Мультиплексирование с GPIO
DAC0	ADC_1/DAC_0	Вход 1 канала АЦП/Выход 0 канала ЦАП	Нет
DAC1	ADC_2/DAC_1	Вход 2 канала АЦП/Выход 1 канала ЦАП	Нет
PD5	ADC_3	Вход 3 канала АЦП	Да
PD4	ADC_4	Вход 4 канала АЦП	Да
PD3	ADC_5	Вход 5 канала АЦП	Да
PD2	ADC_6	Вход 6 канала АЦП	Да
PD1	ADC_7/PGA1_OUT	Вход 7 канала АЦП/Выход ОУ1	Да
PD0	ADC_8/PGA1_IN-	Вход 8 канала АЦП/Отрицательный вход ОУ1	Да
PGA1_IN	ADC_9/PGA1_IN+	Вход 9 канала АЦП/Положительный вход ОУ1	Нет
PC16	ADC_10/COMP1	Вход 10 канала АЦП/Вход 1 аналогового компаратора	Да

PC15	ADC_11/COMP0	Вход 11 канала АЦП/Вход 0 аналогового компаратора	Да
PC14	ADC_12/PGA1_IN+	Вход 12 канала АЦП/Положительный вход ОУ0	Да
PC13	ADC_13/PGA1_OUT	Вход 13 канала АЦП/Выход ОУ0	Да
PC12	ADC_14/PGA1_IN-	Вход 14 канала АЦП/Отрицательный вход ОУ0	Да
PC11	ADC_17	Вход 17 канала АЦП	Да
PC10	ADC_18	Вход 18 канала АЦП	Да
EXT_VREF	VREFP	Вход внешнего опорного напряжения/выход внетреннего опорного напряжения	Да

1.3.14.5.5 Мультиплексор каналов АЦП

На рисунке 1.165 приведена схема мультиплексирования каналов АЦП. Мультиплексирование осуществляется для каждого канала отдельно, в одиночном, либо, дифференциальном режиме. Управление мультиплексированием осуществляется в блоке управления АЦП. Более подробно с регистрами управления АЦП можно ознакомиться в разделе.

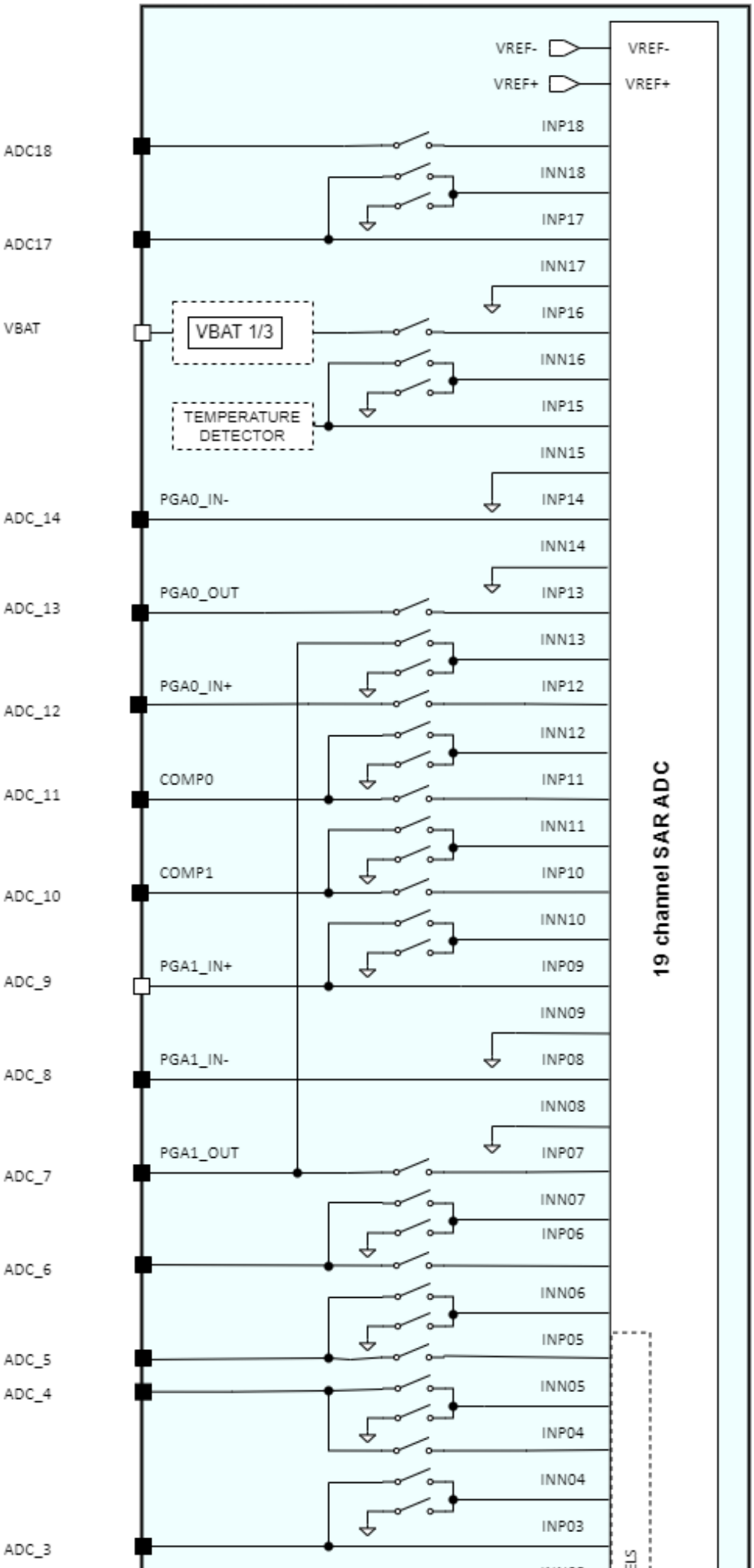

VREF-
VREF+
VREF-
VREF+
ADC18
INP18
INN18
ADC17
INP17
INN17
VBAT
VBAT 1/3
INP16
INN16
TEMPERATURE DETECTOR
INP15
INN15
PGA0_IN-
ADC_14
INP14
INN14
PGA0_OUT
ADC_13
INP13
INN13
PGA0_IN+
ADC_12
INP12
INN12
COMP0
ADC_11
INP11
INN11
COMP1
ADC_10
INP10
INN10
PGA1_IN+
ADC_9
INP09
INN09
PGA1_IN-
ADC_8
INP08
INN08
PGA1_OUT
ADC_7
INP07
INN07
ADC_6
INP06
INN06
ADC_5
INP05
ADC_4
INN05
INP04
INN04
ADC_3
INP03
19 channel SAR ADC

Рисунок 1.165 – Упрощеннная схема формирования выходных сигналов компараторов

Схема мультиплексирования сигналов на входах каналов АЦП представлена на рисунке 1.165.

В таблице 1.689 приведены доступные каналы АЦП в одиночном режиме.

Таблица 1.689 – одиночные каналы АЦП аналогового макроблока

Номер	Источник	Описание
0	VREF11	Внутренне опорное напряжение 1.1В. Значение для калибровки шкалы.
1	ADC_1/DAC_0	Канал АЦП с внешнего входа либо с выхода DAC0
2	ADC_2/DAC_1	Канал АЦП с внешнего входа либо с выхода DAC1
3	ADC_3	Канал АЦП с внешнего входа
4	ADC_4	Канал АЦП с внешнего входа
5	ADC_5	Канал АЦП с внешнего входа
6	ADC_6	Канал АЦП с внешнего входа
7	ADC_7/PGA1	Канал АЦП с внешнего входа либо с выхода PGA1
8	ADC_8	Канал АЦП с внешнего входа
9	ADC_9	Канал АЦП с внешнего входа
10	ADC_10	Канал АЦП с внешнего входа
11	ADC_11	Канал АЦП с внешнего входа
12	ADC_12	Канал АЦП с внешнего входа
13	ADC_13/PGA0	Канал АЦП с внешнего входа либо с выхода PGA0
14	ADC_14	Канал АЦП с внешнего входа
15	TEMP	Выход датчика температуры
16	VBAT13	Значение напряжения батарейного домена VBAT
17	ADC_17	Канал АЦП с внешнего входа
18	ADC_18	Канал АЦП с внешнего входа

В таблице 1.690 приведены доступные каналы АЦП в одиночном режиме.

Таблица 1.690 – дифференциальные каналы АЦП аналогового макроблока

Номер	Источник INP	Источник INN	Описание
0	-	-	Канал недоступен в дифференциальном режиме
1	-	-	Канал недоступен в дифференциальном режиме
2	ADC_2/DAC_1	ADC_1/DAC_0	Внешние входы ADC_2/ADC_1, либо выходы ЦАП DAC_1/DAC_0
3	-	-	Канал недоступен в дифференциальном режиме
4	ADC_4	ADC_3	Внешние входы ADC_4/ADC_3

5	ADC_5	ADC_4	Внешние входы ADC_5/ADC_4
6	ADC_6	ADC_5	Внешние входы ADC_6/ADC_5
7	ADC_7	ADC_6	Внешние входы ADC_7/ADC_6
8	-	-	Канал недоступен в дифференциальном режиме
9	-	-	Канал недоступен в дифференциальном режиме
10	ADC_10	ADC_9	Внешние входы ADC_10/ADC_9
11	ADC_11	ADC_10	Внешние входы ADC_11/ADC_10
12	ADC_12	ADC_11	Внешние входы ADC_12/ADC_11
13	ADC_13/PGA0	ADC_7/PGA1	Внешние входы ADC_13/ADC_7
14	-	-	Канал недоступен в дифференциальном режиме
15	-	-	Канал недоступен в дифференциальном режиме
16	-	-	Канал недоступен в дифференциальном режиме
17	-	-	Канал недоступен в дифференциальном режиме
18	ADC_18	ADC_17	Внешние входы ADC_18/ADC_17

1.3.14.6 Блок аналого-цифрового преобразователя (ADC)

Блок аналогоцифрового преобразования (ADC) предназначен для оцмфровки как внутренних аналоговых сигналов контроллера, таких как напряжение питания или температура кристалла, так и аналоговых сигналов, подаваемых на внешние выводы контроллера. Блок реализован на основе аналогоцифрового преобразователя (АЦП) с аналоговым мультиплексором на входе. Входной аналоговый мультиплексор позволяет подать на вход АЦП до девятнадцати различных аналоговых сигналов. Результат оцифровки фиксируется в регистрах контроллера АЦП и может быть считан процессорным ядров или блоком ПДП.

Чтение данныхЮ управление питанием, а также конфигурация вспомогательных узлов контроллера аналогоцифрового преобразования осуществляется по шине AMBA AHB спецификации 5.0 со следующими особенностями:

– циклы чтения и записи выполняются в блоке за один такт

– аппаратура интерфейса AMBA_AHB блока не поддерживает пакетную передачу данных

Аналогоцифровой преобразователь (АЦП) тактируются отдельным тактовым сигналом (ADC_CLK) асинхронным по отношению к тактовому сигналу шины AMBA AHB.

1.3.14.6.1 Основные параметры аналогоцифрового преобразователя

В таблице 1.691 приведены основные параметры аналогоцифрового преобразователя.

Таблица 1.691 – Основные параметры аналогоцифрового преобразователя

Параметр	Обозначение	Мин.	Тип.	Макс.	Единицы
Напряжение аналогово питания	avddhv	1,62	3,3	3,6	В
Напряжение цифрового питания	dvdd	0,81	0,9	1.21	В
Напряжение смещения	vrefp	1,00			В
Диапазон входных напряжений в униполярном режиме		agndref		vrefp	В
Диапазон входных напряжений в дифференциальном режиме			2*(vrefp-agndref)		B_{ppdif}
Напряжение на входах АЦП в дифференциальном режиме относительно земли V_{cmin} = (vrefp-agndref)/2		V_{cmin}-0,18	V_{cmin}	V_{cmin}+0,18	В
Ток потребления при частоте оцифровки 5,71 MSPS			1061		мкА
Ток потребления при частоте оцифровки 1 MSPS			195		мкА
Ток потребления при частоте оцифровки 10 kSPS			11		мкА
Ток потребления в режиме Deep Power down			0,047		мкА
Ток потребления в режиме Power down			0,27		мкА
Ток потребления в режиме Standby			9,12		мкА
Разрешение		6		12	бит

В таблице 1.692 приведены временные параметры аналогоцифрового преобразователя.

Таблица 1.692 – Временные параметры аналогоцифрового преобразователя

Параметр	Обозначение	Мин.	Тип.	Макс.	Единицы
Тактовая частота АЦП	F_{clk}	0,14		80	МГц
Скважность тактового сигнала АЦП		45	50	55	%
Время оцифровки сигнала: разрешение - 12 бит			14		тактов
Время оцифровки сигнала: разрешение - 10 бит			12		тактов
Время оцифровки сигнала: разрешение - 8 бит			10		тактов
Время оцифровки сигнала: разрешение - 6 бит			8		тактов
Время выхода источника опорного напряжения в рабочий режим	t_{up_ldo}			20	мкс

Время простоя АЦП без регенерации	t_{rearm}			100	мкс

В таблице 1.693 приведены основные характеристики аналогоцифрового преобразователя. Характеристики АЦП приведены при следующих условиях: напряжение питания avddhv = 3,3В, dvdd = 0,9В, температура T = 25°, напряжение смещения vref = avddhv, частота входного сигнала F_{in} = 50 кГц.

Таблица 1.693 – Основные характеристики аналогоцифрового преобразователя

Параметр	Обозначение	Мин.	Тип.	Макс.	Единицы
Дифференциальная нелинейность	DNL	-1,0		+1,0	LSB
Интегральная нелинейность	INL	-2,0		+2,0	LSB
SINAD	SINAD		65		dBFS
Ошибка смещения (калибровка проведена)	Offset error		±2		LSB
Ошибка смещения (калибровка не проведена)	Offset error		±64		LSB
Общее максимальное отклонение (калибровка проведена)	Total unadjusted trror		±5		LSB

1.3.14.6.2 Структурная схема блока аналогоцифрового преобразователя

Структурная схема блока аналогоцифрового преобразователя представлена на рисунке 1.166.

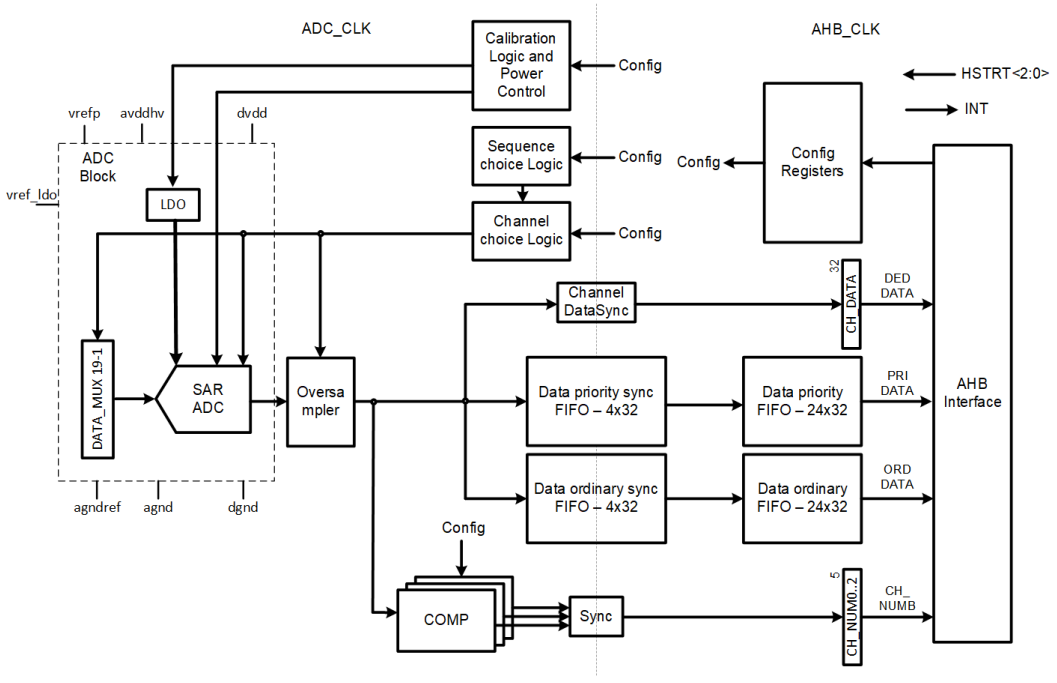


Рисунок 1.166 – Структурная схема блока аналогоцифрового преобразователя

Блок аналогоцифрового преобразователя содержит следующие узлы и блоки:

– аналогоцифровой преобразователь (АЦП) последовательного приближения с аналоговым мультиплексором 19→1 на входе и источником опорного напряжения (LDO).

– блока управления пмтанием АЦП, управляющем включением источника опорного напряжения АЦП, самого АЦП и подачей сигнала сброса на АЦП. Кроме этого блок содержит конечный автомат, выполняющий процедуру калибровки АЦП.

– блок выбора выполняемой последовательности оцифровки. На основе заданных параметров блок выбирает ту последовательность измерений, которыя будет выполняться в текущий момент.

– блок выбора канала. Данный блок выбирает из последовательности канал, вход которого будет оцифровываться, задает на АЦП конфигурацию выбранного канала и управляет процессом оцифровки

– блок передискретизации (Oversampler). В зависимости от конфигурации данный блок суммирует отсчеты измеряемого канала и преобразует данные в заданный формат.

– трех идентичных компараторов, позволяющих сравнивать данные с выхода блока передискретизации с заранее задаваемыми порогами.

Работа всех вышеперечисленных блоков тактируется тактовым сигналом работы аналоговцифрового преобразователя (ADC_CLK).

– блока регистров конфигурации аппаратуры

– регистра данных выделенного канала. Регисто хранит данные оцифровки выделенного приоритетного канала.

– буферов данных оцифровки из 24 регистров каждый - по одному на каждую последовательность. Такой буфер позволяет сохранять данные оцифровки от всех каналов последовательности.

– трех регистров номера канала (по одному на каждый компаратор). Данные регистры хранят номер канала, при сравнении с оцифрованными данными которого компаратор сработал.

– интерфейса с шиной AMBA AHB спецификации 5.0.

Работа всех вышеперечисленных блоков тактируется тактовым сигналом шины AMBA AHB (AHB_CLK).

– схем синхронизации Sync, необходимых при переходе с частоты системной шины (ADC_CLK) на частоту работы преобразователей (AHB_CLK). Схемы синхронизации не предназначены для промежуточного хранения данных.

1.3.14.6.3 Управление питанием аналогоцифрового преобразователя

Аналогоцифровой преобразователь может находиться в разных режимах потребления.

Таблица 1.694 – Режимы потребления аналогоцифровогопреобразователя (АЦП) в зависимости от состояния конфигурационных входов, управляющих питанием

Режим потребления	PWR_DPD	LDO_CTRL	PWR_ENADC
Deep Power-down	1	0	0
Power-down	0	0	0
Standby	0	1	0
Normal operation	0	1	1

Режим Deep Power-down позволяет пользователю снизить не только ток потребления, но и токи утечки до абсолютного минимума. В данном режиме вся внутренняя аппаратура АЦП отключена и питание цифровой части АЦП (dvdd) можно снять. Выход из данного режима возможен только при полностью установившихся напряжения питания и тактовои сигнале АЦП (ADC_CLK). По системному сбросу устанавливается данный режим потребления.

Режим Power-down позволяет снизить потребление АЦП, не выключая его полностью. В данном режиме выключены само АЦП и источник опорного напряжения (LDO), однако ряд внутренних цифровых схем остаются запитанными.

Режим Standby позволяет выключить только собственно АЦП, оставив источник опорного напряжения (LDO) включенным. Это снижает до минимума ток потребления АЦП и позволяет быстро вернуться в нориальный режим измерения без ожидания времени включения LDO.

Режим Normal operation позволяет полностью использовать все возможности АЦП. Ток потребления в этом режиме зависит только от тактовой частотв работы АЦП.

1.3.14.6.3.1 Источник опорного напряжения (LDO)

Источник опорного напряжения (LDO) вырабатывает стабилизированное напряжение номиналом 1,10 В. для использования при аналогоцифровом преобразовании. Источником напряжения, на основе которого формируется опорное напряжение, может быть либо напряжение питания цифровой части АЦП (dvdd), либо напряжение поданное на внешний вывод блока vref_ldo. Выбор источника напряжения для LDO осуществляется полем LDO_SELREF конфигурационного регистра ADC_CFG.

Напряжение питания цифровой части АЦП (dvdd) варьируется в пределах 0,81 - 1,21 В. Однако регулятор напряжения, на основе которого реализован источник опорного напряжения

(LDO) не может обеспечить выходное напряжение 1,10 В во всем диапазоне входных напряжений. Поэтому диапазон входных напряжений LDO разбит на две части 0,81 - 0,99 В и 0,99 - 1,21 В. Выбор соответствующего диапазона входных напряжений LDO осуществляется полем LDO_SELRNG конфигурационного регистра ADC_CFG. Соответственно если источником входного сигнала является внешний вывод vref_ldo, то напряжение на нем должно соответствовать выбранному диапазону.

При несовпадении номинала входного напряжения LDO выбранному диапазону входных напряжений корректная работа АЦП не гарантируется и такое несоответствие может приводить к выгоранию LDO.

Перед изменениием номинала питания цифровой части АЦП или перед измененииеы номинала напряжения на входе vref_ldo должна быть проведена следующая процедура:

– выключить АЦП, установив в логический ноль регистр PWR_ENADC.

– выключить источник опорного напряжения (LDO), установив в логический ноль регистр LDO_CTRL.

– изменить конфигурационные параметры питания LDO в регистре ADC_CFG.

– изменить номиналы напряжения питания цмфровой части АЦП (dvdd) или номиналы напряжения на входе vref_ldo блока.

– провести процедуру вклбчения АЦП в соответствии с разделом 1.3.14.6.3.3.

После включении LDO (установка в единицу регистра LDO_CTRL) необходимо определенное время (не менее 20 мкс) для установки опорного напряжения АЦП. В течении этого времени никакие операции с АЦП невозможны. Отсчет данного времени осуществляется выделенным счетчиком. Значение времени задается в поле LDO_TUP конфигурационного регистра ADC_CFG в тактах тактового сигнала АЦП (ADC_CLK).

При включении LDO после отсчета времени установки LDO данным счетчиком в регисре PWR_STS устанавливается поле LDO_STS. Установка данного поля говорит о том, что источник опорного напряжения полностью готов к работе.

1.3.14.6.3.2 Калибровка аналогоцифрового преобразователя (АЦП)

В блоке аналогоцифрового преобразователя реализован конечный автомат, позволяющий определить калибровочные коэффициенты как для униполярного, так и для дифференциального режимов измерения. В процессе калибровки вход аналогоцифрового преобразователя (АЦП) отключается от входного мультиплексора и на него подается эталонное напряжение, после чего производится ряд измерений с усреднением результата.

Конечный автомат запускается установкой в единицу регистра CAL_STRT. В процессе калибровки данный регистр сбрасывается аппаратно. Признаком окончания процедуры калибровки служит установка поля CAL_STS регистра PWR_STS. Калибровочные коэффициенты записываются в программно доступный регистр CAL_DATA и могут быть считаны программно. Коэффициенты, сохраненные в регистре CAL_DATA впоследствии загружаются в АЦП при выполнении каждой процедуры оцифровки данных.

Имеется возможность, не запуская автомат калибровки, загрузить калибровочные коэффициенты из памяти контроллера. В этом случае поле CAL_STS регистра PWR_STS установится в единицу после операции записи.

Калибровку рекомендуется проводить при изменении номиналов напряжения питания АЦП и сопутствующих узлов (LDO), а также периодически при длительноц работе контроллера. Если точность аналогоцифрового преобразования не критична, то в регистр коэффициентов калибровки необходимо записать код 2040h.

Запуск конечного автомата калибровки и любое изменение калибровочных коэффициентов возможно только если АЦП не находится в процессе оцифровки данных.

1.3.14.6.3.3 Последовательность включения аналогоцифрового преобразователя (АЦП)

После систеиного сброса контроллер переводит аналогоцифровой преобразователь в состояние Deep Power Mode. Типовая процедура приведения АЦП в работоспособное состояние приведена ниже.

1) необходимо убедиться, что на АЦП подано и установилось выбранное напряжение питания цифровой (dvdd) и аналоговой (avddhv) части аналогоцифрового преобразователя. а также подоно опорное напряжение (vrefp), определяющее диапазон входных аналоговых сигналов. Если выбрана конфигурация использования источника опорного напряжения (LDO) с внешним источником опорного напряжения необходимо, чтобы это напряжение было подано на вывод (vref_ldo) блока.

2) необходимо убедиться, что тактовый сигнал АЦП (ADC_CLK) подан и установился.

3) прописать в регистре ADC_CFG параметры конфигурации источника опорного напряжения (LDO), время выхода источника на рабочий режим и максимальное время простоя АЦП, без выполнения процедуры регенерации. Требуемые времена расчитываются исходя из параметров АЦП, приведенных в разделе 1.691 и периода тактового сигнала АЦП (ADC_CLK). Время задается в тактах сигнала ADC_CLK.

4) перевести АЦП в режим Power Down, сбросив регистр PWR_DPD.

5) включить источник опорного напряжения, установив в единицу регистр LDO_CTRL.

6) дождаться установки в единицу разряда LDO_STS регистра PWR_STS, свидетельствующего, что LDO включен и готов к работе.

7) включить АЦП, установив в единицу регистр PWR_ENADC.

8) подато сигнал сброса АЦП, записав единицу в регистр PWR_RSTADC. После подачи сигнала сброса АЦП регистр PWR_RSTADC сбрасывается аппаратно.

9) дождаться установки в единицу разряда ADC_STS регистра PWR_STS, свидетельствующего, что АЦП включен и готов к работе.

10) провести калиьровку АЦП, установив в единицу регистр CAL_STRT, или прописать ранее сохраненные калибровочные кожффициенты в регистр CAL_DATA. В процессе проведения калибровки регистр CAL_STRT сбросится аппаратно.

11) дождаться установки в единицу разряда CAL_STS регистра PWR_STS, свидетельствующего, что калибровочные коэффициенты готовы и находятся в регистре CAL_DATA.

После этого аналогоцифровой преобразователь готов к проведению оцифровки входных аналоговых сигналов.

1.3.14.6.4 Организация работы блока аналогоцифрового преобразователя

1.3.14.6.4.1 Канал оцифровки данных

Вход аналогоцифрового прелбразователя подключен к выходу аналогового мультиплексора 19-1, что позволяет в режиме разделения времени оцифровывать до 19 аналоговых сигналов. Тракт оцифровки одного аналогового сигнала назван каналом оцифровки. Номер канала оцифровки соответствует номеру входа аналогового мультиплексора. Каналы 0 - 5 являются "быстрыми". Каналы 6 - 18 являются "медленными". Так как аналоговый мультиплексор занимает значительное место на кристалле, невозможно полностью выровнять пути данных от каждого входа аналогового мультиплексора до входа собственно АЦП. Плэтому отличие "медленных" каналов от "быстрых" в том, что им требуется юольшее время задержки между переключением аналогового мультиплексора и началом собственно аналогоцифроаого преобразования, что снижает максимальную частоту отсчетов (sampling rate) "медленных" каналов. В остальном эти каналы идентичны.

Поскольку параметры оцифровки разных аналоговых сигналов могут быть разными, эти параметры настраиваются для каждого канала индивидуально в конфигурационных регистрах

кналов CH_CFGi (i=0,…18). Возможности настройки всех каналов одинаковы, за исключением нулевого канала. Нулевой канал нельзя настроить на оцифровку входного дифференциального сигнала.

В конфигурационных регистрах канала имеется возможность задавать следующие параметры:

– поле CH_DEL задает задержку (в тактах тактового сигнала цифроаналогового преобразователя ADC_CLK) между переключением аналогового мультиплексора на входе АЦП и первым стартом процедуры оцифровки канала.

– поля CH_OVSR и CH_OVSS задают параметры передискретизации канала. Поле CH_OVSR задает количество последовательно выполняемых процедур оцифровки канала с последующим суммированием результатов оцифровки. Поле CH_OVSS задает размер сдвига вправо результата суммирования.

– поля CH_RES и CH_MD задают параметры оцифровки. Поле CH_RES задает разрешение входных данных. При разрешении менее 12 бит результат, получаемый с аналогоцифрового преобразователя расположен в старших разрядах получаемого слова, а младшие разряды (в соответствии с разрешением) обнулены. Поле CH_MD задает тип аналогового сигнала - униполярный или дифференциальный.

– поле CH_DTYP определяет тип получаемых данных - беззнаковые или в дополнительном коде.

Все параметры используемых каналов должны быть прописаны до того как запущена процедура оцифровки данных.

1.3.14.6.4.2 Последовательности опроса каналов оцифровки

Каналы оцифровки опрашиваются в определенной последовательности. Пользователь имеет возможность задать две такие последовательности - ординарную и приоритетную. По принципам работы последовательности одинаковы. Единственное исключение - приоритет доступа к аналогоцифровому преобразователю. Если при выполнении ординарной последовательности будет запущена приоритетная, то приоритетная последовательность прервет выполнение ординарной последовательности, будет полностью выполнена, а затем продолжится выполнение ординарной последовательности. Кроме этого имеется возможность запуска олроса заранее выделенного канала, которая имеет наивысший приоритет по доступу к аналогоцифровому преобразователю, т.е. запуск опроса выделенного канала прерывает выполнение и орлинарной и приоритетной последовательностей.

Количество каналов в последовательности может варьироваться от одного до двадцати и задается в регистрах размера последовательности SEC_ORD_NUM для ординарной последовательности и SEC_PRI_NUM для приоритетной последовательности. Порядок опроса каналов оцифровки в последовательности определяется в регистрах SEC_ORD_CFGi(0-3) для ординарной последовательности и SEC_PRI_CFGi(0-3) для приоритетной последовательности. Первым в последовательности будет запущен канал, номер которого отражен в поле ORD_NUM1 (PRI_NUM1), затем канал номер которого отражен в поле ORD_NUM2 (PRI_NUM2), и так далее до конца последовательности. Номер выделенного канала задается в регистре CH_DED_CFG.

Каждая последовательность и опрос выделенного канала могут быть запущены как программно - путем установки в единицу полей STT_ORD, STT_PRI или STT_DED, так и аппаратно - подачей сигнала запуска на внешние выводы блока (HSTRT<2:0>). Сигнал запуска представляет любое изменение состояния сигнвала на этих внешних выводах блока. Аппаратный запуск последовательностей разрешается установкой в единицу полей SEC_ORD_ACT, SEC_PRI_ACT или DED_CH_ACT регистра SEC_ST_CFG. При программном старте последовательностей биты старта сбрасываются аппаратно.

Результатом выполнения последовательности или опроса выделенного канала является набор данных по количеству каналов в последовательности. Данные каналов при выполнении последовательностей сохраняются в выходном буфере ординарной и приоритетной последовательностей. Размер буфера - 24 слова данных. Выходной буфер доступен по чтению регистров SEC_ORD_D и SEC_PRI_D соответственно. Данные опроса выделенного канала сохраняются в регистре данных канала SEC_DED_D.

Однако если данные какого-нибудь канала необходимы только для сравнения на компараторах данного блока и не требуются для дальнейшей программной обработки можно блокировать запись данных канала в выходной Буфер последовательности путем установки в единицу соответствующего разряда в регистре SEC_ORD_CHWE для ординарной и SEC_PRI_CHW для приоритетной последовательностей.

По результатам выполнения последовательностей формируются следующие прерывания:

– по наличию данных в выходном буфере последовательностей.

– по окончанию выполнения последовательностей или опроса выделенного канала.

– по переполнению выходного буфера последовательностей.

При переполнении выходного буфера последовательностей возможны два варианта сохранения данных. В первом случае данные, переполняющие выходной буфер последовательности, игнорируются, во втором случае данные, переполняющме буфер, вытесняют хранящиеся в буфере, начиная с самого старого. Выбор типа поведения буфера при переполнении определяется состоянием полей SEC_ORD_OVER и SEC_PRI_OVER регистра SEC_ST_CFG.

Состояние последовательностей и процедуры опроса выделенного канала отражается в регистре статуса последовательностей SEC_STS.

1.3.14.6.4.3 Блок передискретизации (Oversampling)

Блок передискретизации предназначен для аппаратного выполнения простейших операций над оцифрованными данными. таких как суммирование и сдвиг. Эти операции позволяют выполнять функции усреднения результата, повысить соотношение сигнал/шум, осуществить предварительную фильтрацию и т.п. Параметры передискретизации задаются для каждого канала оцифровки индивидуально в регистрах конфигурации канала. При оцифровке канала производится заданное в конфигурационных регистрах количество измерений. Блок передискретизации суммирует эти измерения и по окончании опроса канала сдвигает результат суммирования вправо на заданное число разрядов.

1.3.14.6.4.4 Компараторы оцифрованных данных

Компаратор позволяет сравнивать получаемые в процессе оцифровки данные с предварительно заданными порогами. В блоке реализованы три идентисных блока компараторов. Данные для сравнения на компаратор подаются с блока передискретизации при окончании работы с данныи каналом оцифровки. оазрешение работы компаратора задается установкой в единицу соответствующих полей регистоа COMP_EN. В регистрах конфигурации компараторов COMP_CFGi (i=0…2) задается тип данных с которыми работает компаратор и условия при которых осуществляется сравнение. Под условиями сравнения понимается номера каналов, данные которых будут выданы на компаратор, и типы последовательностей, при выполнении которых осуществляется опрос данных каналов. Верхний и нижний пороги сравнения задаюися в регистрах COMP_TRE_Hi и COMP_TRE_Li соответственно.

При срабатывании компаратора формируется прерывание по срабатыванию компаратора и в регистрах CMP_CH_NUMi запоминактся номер канала при сравнении данных которого протзошло данное срабатывание. Регистр номера сработавшего компаратора всегда хранит номер последнего срабатывания.

Кроме этого сигнал срабатывания компаратора может приводить к изменению состояния соответствующего внешнего вывода блока. Данное изменение (фронт или срез) синхронизировано с частотой работы шмны данных блока (AHB_CLK). Разрешение на такое изменение задается установкой соответствующих разрядов в регистре COMP_EN блока.

1.3.14.6.4.5 Прерывания блока аналогоцифрового преобразователя

В блоке формируются и сохраняются в регистре ADC_IRR следующие запросы на прерывания:

– прерываниЯ по окончанию выполнения ординарной и приоритетной последовательностей и окончанию опроса выделенного канала ORD_END, PRI_END и DED_END соответственно. Запрос на данное прерывание формируется при записи последних данных в буфер данных соответствующей последовательности. Данnse запросы на прерывание сбрасывается программно.

– прерывание по наличию данных в буфере данных ордмнарной или приоритетной последовательности ORD_FIFO_NE и PRI_FIFO_NE соответственно. Данные запросы устанавливается при записи первых данных последовательности в буфер данных и аппаратно сбрасывается при чтении последних данных из буфера.

– прерывание по потере данных при выполнении ординарной и приоритетной последовательностей и процедуры опроса выделенного канала ORD_FIFO_OVF, PRI_FIFO_OVF, и DED_OVF. Данные запросы на прерывание сбрасывается программно.

– прерывание по сраьаиыванию компараторов ICMPi (i=0…2). Данные запросы на прерывание сбрасывается программно.

Все прерывания могут быть замаскированы установкой соответствуюшей маски в регистре ADC_IMR.

Выходной запрос на прерывание от блока представляет собой свертку всех запросов регистра ADC_IRR по "или" с учетом масок.

1.3.14.6.4.6 Программно доступные регистры блок ADC

В таблице 1.695 представлен список программно доступных регистров блок ADC.

Все программно доступные регистры блока аналогоцифровых преобразователей расположены начиная с базового адреса устройства – ADC_Base.

Таблица 1.695 – Список программно доступных регистров блок ADC

Название	Адрес	Описание
ADC_IRR	0x000	Регистр запросов на прерывание
ADC_IMR	0x004	Регистр масок запросов на прерывание
ADC_IMR_set	0x008	Регистр установки масок запросов на прерывание
ADC_IMR_rst	0x00C	Регистр сброса масок запросов на прерывание
SEC_ORD_D	0x010	Регистр данных ординарной последовательности
SEC_PRI_D	0x014	Регистр данных приоритетной последовательности
SEC_DED_D	0x018	Регистр данных выделенного канала
RESERVED	0x01C	Зарезервировано
CMP_CH_NUM0	0x020	Регистр номера канала компаратора 0
CMP_CH_NUM1	0x024	Регистр номера канала компаратора 1
CMP_CH_NUM2	0x028	Регистр номера канала компаратора 2
RESERVED	(0x02C — 0x03C)	Зарезервировано
CH_CFG0	0x040	Регистр конфигурации канала 0
CH_CFG1	0x044	Регистр конфигурации канала 1
CH_CFG2	0x048	Регистр конфигурации канала 2
CH_CFG3	0x04C	Регистр конфигурации канала 3
CH_CFG4	0x050	Регистр конфигурации канала 4
CH_CFG5	0x054	Регистр конфигурации канала 5
CH_CFG6	0x058	Регистр конфигурации канала 6
CH_CFG7	0x05C	Регистр конфигурации канала 7
CH_CFG8	0x060	Регистр конфигурации канала 8
CH_CFG9	0x064	Регистр конфигурации канала 9
CH_CFG10	0x068	Регистр конфигурации канала 10
CH_CFG11	0x06C	Регистр конфигурации канала 11
CH_CFG12	0x070	Регистр конфигурации канала 12
CH_CFG13	0x074	Регистр конфигурации канала 13
CH_CFG14	0x078	Регистр конфигурации канала 14
CH_CFG15	0x07C	Регистр конфигурации канала 15
CH_CFG16	0x080	Регистр конфигурации канала 16
CH_CFG17	0x084	Регистр конфигурации канала 17
CH_CFG18	0x088	Регистр конфигурации канала 18
RESERVED	0x08C	Зарезервировано
SEC_START	0x090	Регистр программного старта и останова аналогоцифрового преобразования
SEC_ST_CFG	0x094	Регистр конфигурации запуска аналогоцифрового преобразования
RESERVED	(0x098 — 0x09C)	Зарезервировано
SEC_ORD_NUM	0x0A0	Регистр размера ординарной последовательности
SEC_ORD_CFG0	0x0A4	Регистр 0 конфигурации ординарной последовательности

Название	Адрес	Описание
SEC_ORD_CFG1	0x0A8	Регистр 1 конфигурации ординарной последовательности
SEC_ORD_CFG2	0x0AC	Регистр 2 конфигурации ординарной последовательности
SEC_ORD_CFG3	0x0B0	Регистр 3 конфигурации ординарной последовательности
SEC_ORD_CHWE	0x0B4	Регистр разрешения записи данных канала в выходной буфер
RESERVED	(0x0B8 — 0x0BC)	Зарезервировано
SEC_PRI_NUM	0x0C0	Регистр размера приоритетной последовательности
SEC_PRI_CFG0	0x0C4	Регистр 0 конфигурации приоритетной последовательности
SEC_PRI_CFG1	0x0C8	Регистр 1 конфигурации приоритетной последовательности
SEC_PRI_CFG2	0x0CC	Регистр 2 конфигурации приоритетной последовательности
SEC_PRI_CFG3	0x0D0	Регистр 3 конфигурации приоритетной последовательности
SEC_PRI_CHWE	0x0D4	Регистр разрешения записи данных канала в выходной буфер
RESERVED	(0x0D8 — 0x0DC)	Зарезервировано
CH_DED_CFG	0x0E0	Регистр конфигурации выделенного канала
SEC_STS	0x0E4	Регистр статуса аналогоцифрового преобразования
RESERVED	(0x0E8 — 0x0EC)	Зарезервировано
COMP_EN	0x0F0	Регистр разрешения работы компараторов
COMP_EN_set	0x0F4	Регистр установки разрешения работы компараторов
COMP_EN_rst	0x0F8	Регистр сброса разрешения работы компараторов
RESERVED	0x0FC	Зарезервировано
COMP_CFG0	0x100	Регистр конфигурации компаратора 0
COMP_TRE_H0	0x104	Регистр верхнего порога сравнения компаратора 0
COMP_TRE_L0	0x108	Регистр нижнего порога сравнения компаратора 0
RESERVED	0x10C	Зарезервировано
COMP_CFG1	0x110	Регистр конфигурации компаратора 1
COMP_TRE_H1	0x114	Регистр верхнего порога сравнения компаратора 1
COMP_TRE_L1	0x118	Регистр нижнего порога сравнения компаратора 1
RESERVED	0x11C	Зарезервировано
COMP_CFG2	0x120	Регистр конфигурации компаратора 2
COMP_TRE_H2	0x124	Регистр верхнего порога сравнения компаратора 2
COMP_TRE_L2	0x128	Регистр нижнего порога сравнения компаратора 2
RESERVED	(0x12C — 0x17C)	Зарезервировано
PWR_DPD	0x180	Регистр управления режимом минимального потребления (Deep_Power_down)
LDO_CTRL	0x184	Регистр управления включением источника опорного напряжения (LDO)
PWR_ENADC	0x188	Регистр включения рабочего режима АЦП
PWR_RSTADC	0x18C	Регистр системного сброса АЦП
CAL_STRT	0x190	Регистр управления стартом конечного автомата калибровки АЦП

Название	Адрес	Описание
CAL_DATA	0x194	Регистр калибровочных коэффициентов
PWR_STS	0x198	Регистр состояния питания и калибровки АЦП
ADC_CFG	0x19C	Регистр конфигурации АЦП

1.3.14.6.4.6.1 Регистр запросов на прерывание ADC_IRR

Регистр хранит запросы на прерывания от блока аналлогоцифровых преобразователей. Запись логической единицы обнуляет соответствующий бит регистра

В таблице 1.696 представлено описание полей регистра ADC_IRR.

Таблица 1.696 – Описание полей регистра ADC_IRR

Поле	Название	Программный доступ	Состояние после сброса	Описание
31:15	—	ЧТ0	—	Зарезервировано
14	ICMP2	ЧТ/ЗП	0x0	Срабатывание компаратора 2;
13	ICMP1	ЧТ/ЗП	0x0	Срабатывание компаратора 1;
12	ICMP0	ЧТ/ЗП	0x0	Срабатывание компаратора 0;
11:10	Reserved2	-	0x0	Зарезервировано;
9	DED_OVF	ЧТ/ЗП	0x0	Регистр данных выделенного канала был переполнен;
8	DED_END	ЧТ/ЗП	0x0	Окончание процедуры опроса выделенного канала;
7	Reserved1	-	0x0	Зарезервировано;
6	PRI_FIFO_OVF	ЧТ/ЗП	0x0	Буфер данных приоритетной последовательности был переполнен;
5	PRI_FIFO_NE	ЧТ/ЗП	0x0	Буфер данных приоритетной последовательности не пуст;
4	PRI_END	ЧТ/ЗП	0x0	Окончание приоритетной последовательности;
3	Reserved0	-	0x0	Зарезервировано;
2	ORD_FIFO_OVF	ЧТ/ЗП	0x0	Буфер данных ординарной последовательности был переполнен;
1	ORD_FIFO_NE	ЧТ/ЗП	0x0	Буфер данных ординарной последовательности не пуст;
0	ORD_END	ЧТ/ЗП	0x0	Окончание ординарной последовательности;

1.3.14.6.4.6.2 Регистр масок запросов на прерывание ADC_IMR

Регистр хранит маски запросов на прерывания от блока аналогоцифровых преобразователей;

В таблице 1.697 представлено описание полей регистра ADC_IMR.

Таблица 1.697 – Описание полей регистра ADC_IMR

Поле	Название	Программный доступ	Состояние после сброса	Описание
31:15	—	ЧТ0	—	Зарезервировано
14	M_ICMP2	ЧТ/ЗП	0x0	Маска запроса на прерывание по срабатыванию компаратора 2: 0 - маска установлена; 1 - маска сброшена;
13	M_ICMP1	ЧТ/ЗП	0x0	Маска запроса на прерывание по срабатыванию компаратора 1: 0 - маска установлена; 1 - маска сброшена;
12	M_ICMP0	ЧТ/ЗП	0x0	Маска запроса на прерывание по срабатыванию компаратора 0: 0 - маска установлена; 1 - маска сброшена;
11:10	Reserved2	-	0x0	Зарезервировано;
9	M_DED_OVF	ЧТ/ЗП	0x0	Маска запроса на прерывание по перезаписи буфера регистра данных выделенного канала: 0 - маска установлена; 1 - маска сброшена;
8	M_DED_END	ЧТ/ЗП	0x0	Маска запроса на прерывание по окончанию процедуры опроса выделенного канала: 0 - маска установлена; 1 - маска сброшена;
7	Reserved1	-	0x0	Зарезервировано;
6	M_PRI_FIFO_OVF	ЧТ/ЗП	0x0	Маска запроса на прерывание по переполнению буфера приоритетной последовательности: 0 - маска установлена; 1 - маска сброшена;
5	M_PRI_FIFO_NE	ЧТ/ЗП	0x0	Маска запроса на прерывание по наличию данных в буфере приоритетной последовательности: 0 - маска установлена; 1 - маска сброшена;
4	M_PRI_END	ЧТ/ЗП	0x0	Маска запроса на прерывание по окончанию приоритетной последовательности: 0 - маска установлена; 1 - маска сброшена;
3	Reserved0	-	0x0	Зарезервировано;
2	M_ORD_FIFO_OVF	ЧТ/ЗП	0x0	Маска запроса на прерывание по переполнению буфера ординарной последовательности: 0 - маска установлена; 1 - маска сброшена;
1	M_ORD_FIFO_NE	ЧТ/ЗП	0x0	Маска запроса на прерывание по наличию данных в буфере ординарной последовательности: 0 - маска установлена; 1 - маска сброшена;

Поле	Название	Программный доступ	Состояние после сброса	Описание
0	M_ORD_END	ЧТ/ЗП	0x0	Маска запроса на прерывание по окончанию ординарной последовательности: 0 - маска установлена; 1 - маска сброшена;

1.3.14.6.4.6.3 Регистр установки масок запросов на прерывание ADC_IMR_set

Запись логической единицы в разряды регистра устанавливает соответсвующий бит регистра ADC_IMR в единицу;

В таблице 1.698 представлено описание полей регистра ADC_IMR_set.

Таблица 1.698 – Описание полей регистра ADC_IMR_set

Поле	Название	Программный доступ	Состояние после сброса	Описание
31:15	—	ЧТ0	—	Зарезервировано
14	M_ICMP2_set	ЗП	0x0	Установка в 1 маски запроса на прерывание по срабатыванию компаратора 2;
13	M_ICMP1_set	ЗП	0x0	Установка в 1 маски запроса на прерывание по срабатыванию компаратора 1;
12	M_ICMP0_set	ЗП	0x0	Установка в 1 маски запроса на прерывание по срабатыванию компаратора 0;
11:10	Reserved2	-	0x0	Зарезервировано;
9	M_DED_OVF_set	ЗП	0x0	Установка в 1 маски запроса на прерывание по перезаписи буфера регистра данных выделенного канала;
8	M_DED_END_set	ЗП	0x0	Установка в 1 маски запроса на прерывание по окончанию процедуры опроса выделенного канала;
7	Reserved1	-	0x0	Зарезервировано;
6	M_PRI_FIFO_OVF_set	ЗП	0x0	Установка в 1 маски запроса на прерывание по переполнению буфера приоритетной последовательности;
5	M_PRI_FIFO_NE_set	ЗП	0x0	Установка в 1 маски запроса на прерывание по наличию данных в буфере приоритетной последовательности;
4	M_PRI_END_set	ЗП	0x0	Установка в 1 маски запроса на прерывание по окончанию приоритетной последовательности;
3	Reserved0	-	0x0	Зарезервировано;
2	M_ORD_FIFO_OVF_set	ЗП	0x0	Установка в 1 маски запроса на прерывание по переполнению буфера ординарной последовательности;

Поле	Название	Программный доступ	Состояние после сброса	Описание
1	M_ORD_FIFO_NE_set	ЗП	0x0	Установка в 1 маски запроса на прерывание по наличию данных в буфере ординарной последовательности;
0	M_ORD_END_set	ЗП	0x0	Установка в 1 маски запроса на прерывание по окончанию ординарной последовательности;

1.3.14.6.4.6.4 Регистр сброса масок запросов на прерывание ADC_IMR_rst

Запись логической единицы в разряды регистра сбрасывает соответсвующий бит регистра ADC_IMR в ноль;

В таблице 1.699 представлено описание полей регистра ADC_IMR_rst.

Таблица 1.699 – Описание полей регистра ADC_IMR_rst

Поле	Название	Программный доступ	Состояние после сброса	Описание
31:15	—	ЧТ0	—	Зарезервировано
14	M_ICMP2_rst	ЗП	0x0	Сброс в 0 маски запроса на прерывание по срабатыванию компаратора 2;
13	M_ICMP1_rst	ЗП	0x0	Сброс в 0 маски запроса на прерывание по срабатыванию компаратора 1;
12	M_ICMP0_rst	ЗП	0x0	Сброс в 0 маски запроса на прерывание по срабатыванию компаратора 0;
11:10	Reserved2	-	0x0	зарезервировано;
9	M_DED_OVF_rst	ЗП	0x0	Сброс в 0 маски запроса на прерывание по перезаписи буфера регистра данных выделенного канала;
8	M_DED_END_rst	ЗП	0x0	Сброс в 0 маски запроса на прерывание по окончанию процедуры опроса выделенного канала;
7	Reserved1	-	0x0	Зарезервировано;
6	M_PRI_FIFO_OVF_rst	ЗП	0x0	Сброс в 0 маски запроса на прерывание по переполнению буфера приоритетной последовательности;
5	M_PRI_FIFO_NE_rst	ЗП	0x0	Сброс в 0 маски запроса на прерывание по наличию данных в буфере приоритетной последовательности;
4	M_PRI_END_rst	ЗП	0x0	Сброс в 0 маски запроса на прерывание по окончанию приоритетной последовательности;
3	Reserved0	-	0x0	Зарезервировано;
2	M_ORD_FIFO_OVF_rst	ЗП	0x0	Сброс в 0 маски запроса на прерывание по переполнению буфера ординарной последовательности;
1	M_ORD_FIFO_NE_rst	ЗП	0x0	Сброс в 0 маски запроса на прерывание по наличию данных в буфере ординарной последовательности;
0	M_ORD_END_rst	ЗП	0x0	Сброс в 0 маски запроса на прерывание по окончанию ординарной последовательности;

1.3.14.6.4.6.5 Регистр данных ординарной последовательности SEC_ORD_D

Регистр - вершина буфера, хранящего данные ординарной последовательности;

В таблице 1.700 представлено описание полей регистра SEC_ORD_D.

Таблица 1.700 – Описание полей регистра SEC_ORD_D

Поле	Название	Программный доступ	Состояние после сброса	Описание
31:0	SEC_ORD_D	ЧТ	0x0	Данные ординарной последовательности ;

1.3.14.6.4.6.6 Регистр данных приоритетной последовательности SEC_PRI_D

Регистр - вершина буфера, хранящего данные приоритетной последовательности;

В таблице 1.701 представлено описание полей регистра SEC_PRI_D.

Таблица 1.701 – Описание полей регистра SEC_PRI_D

Поле	Название	Программный доступ	Состояние после сброса	Описание
31:0	SEC_PRI_D	ЧТ	0x0	Данные приоритетной последовательности ;

1.3.14.6.4.6.7 Регистр данных выделенного канала SEC_DED_D

Регистр, хранящий данные выделенного канала;

В таблице 1.702 представлено описание полей регистра SEC_DED_D.

Таблица 1.702 – Описание полей регистра SEC_DED_D

Поле	Название	Программный доступ	Состояние после сброса	Описание
31:0	SEC_DED_D	ЧТ	0x0	Данные выделенного канала;

1.3.14.6.4.6.8 Регистр номера канала компаратора 0 CMP_CH_NUM0

Регистр, сохраняет номер канала, данные оцифровки которого привели к срабатыванию компаратора 0;

В таблице 1.703 представлено описание полей регистра CMP_CH_NUM0.

Таблица 1.703 – Описание полей регистра CMP_CH_NUM0

Поле	Название	Программный доступ	Состояние после сброса	Описание
31:5	—	ЧТ0	—	Зарезервировано
4:0	CMP_CH_NUM0	ЧТ	0x0	Номер канала при срабатывании компаратора 0;

1.3.14.6.4.6.9 Регистр номера канала компаратора 1 CMP_CH_NUM1

Регистр, сохраняет номер канала, данные оцифровки которого привели к срабатыванию компаратора 1;

В таблице 1.704 представлено описание полей регистра CMP_CH_NUM1.

Таблица 1.704 – Описание полей регистра CMP_CH_NUM1

Поле	Название	Программный доступ	Состояние после сброса	Описание
31:5	—	ЧТ0	—	Зарезервировано
4:0	CMP_CH_NUM1	ЧТ	0x0	Номер канала при срабатывании компаратора 1;

1.3.14.6.4.6.10 Регистр номера канала компаратора 2 CMP_CH_NUM2

Регистр, сохраняет номер канала, данные оцифровки которого привели к срабатыванию компаратора 2;

В таблице 1.705 представлено описание полей регистра CMP_CH_NUM2.

Таблица 1.705 – Описание полей регистра CMP_CH_NUM2

Поле	Название	Программный доступ	Состояние после сброса	Описание
31:5	—	ЧТ0	—	Зарезервировано
4:0	CMP_CH_NUM2	ЧТ	0x0	Номер канала при срабатывании компаратора 2;

1.3.14.6.4.6.11 Регистр конфигурации канала 0 CH_CFG0

В регистре задается конфигурация работы канала 0 аналогоцифрового преобразователя;

В таблице 1.706 представлено описание полей регистра CH_CFG0.

Таблица 1.706 – Описание полей регистра CH_CFG0

Поле	Название	Программный доступ	Состояние после сброса	Описание
31:20	—	ЧТ0	—	Зарезервировано
19	CH_DTYP	ЧТ/ЗП	0x0	Тип данных канала; 0 - беззнаковые; 1 - знаковые (доп. код);
18	CH_MD	ЧТ/ЗП	0x0	Тип сигнала на входе АЦП; 0 - униполярный сигнал на входе АЦП; 1 - дифференциальный сигнал на входе АЦП;
17:16	CH_RES	ЧТ/ЗП	0x0	Разрядность аналогоцифрового преобразования в канале; 00 - 6 бит; 01 - 8 бит; 10 - 10 бит; 11 - 12 бит;
15:12	CH_OVSS	ЧТ/ЗП	0x0	Величина сдвига вправо после суммирования; 0000 - нет сдвига; 0001 - сдвиг на 1 разряд; 0010 - сдвиг на 2 разряда; … 1111 - сдвиг на 15 разрядов;
11:8	CH_OVSR	ЧТ/ЗП	0x0	Количество суммируемых отсчетов при передискретизации; 0000 - нет передискретизации (1 отсчет); 0001 - 2 отсчета; 0010 - 4 отсчета; … 1010 - 1024 отсчета; 1011 - 1111 - зарезервировано;
7:0	CH_DEL	ЧТ/ЗП	0x0	Время задержки между переключением входного аналогового мультиплексора и началом оцифровки данных канала; 00h - 1 такт задержки; 01h - 2 такта задержки; … feh - 255 тактов задержки; ffh - зарезервировано;;

1.3.14.6.4.6.12 Регистр конфигурации канала 1 CH_CFG1

В регистре задается конфигурация работы канала 1 аналогоцифрового преобразователя;

В таблице 1.707 представлено описание полей регистра CH_CFG1.

Таблица 1.707 – Описание полей регистра CH_CFG1

Поле	Название	Программный доступ	Состояние после сброса	Описание
31:20	—	ЧТ0	—	Зарезервировано
19	CH_DTYP	ЧТ/ЗП	0x0	Тип данных канала; 0 - беззнаковые; 1 - знаковые (доп. код);
18	CH_MD	ЧТ/ЗП	0x0	Тип сигнала на входе АЦП; 0 - униполярный сигнал на входе АЦП; 1 - дифференциальный сигнал на входе АЦП;
17:16	CH_RES	ЧТ/ЗП	0x0	Разрядность аналогоцифрового преобразования в канале; 00 - 6 бит; 01 - 8 бит; 10 - 10 бит; 11 - 12 бит;
15:12	CH_OVSS	ЧТ/ЗП	0x0	Величина сдвига вправо после суммирования; 0000 - нет сдвига; 0001 - сдвиг на 1 разряд; 0010 - сдвиг на 2 разряда; … 1111 - сдвиг на 15 разрядов;
11:8	CH_OVSR	ЧТ/ЗП	0x0	Количество суммируемых отсчетов при передискретизации; 0000 - нет передискретизации (1 отсчет); 0001 - 2 отсчета; 0010 - 4 отсчета; … 1010 - 1024 отсчета; 1011 - 1111 - зарезервировано;
7:0	CH_DEL	ЧТ/ЗП	0x0	Время задержки между переключением входного аналогового мультиплексора и началом оцифровки данных канала; 00h - 1 такт задержки; 01h - 2 такта задержки; … feh - 255 тактов задержки; ffh - зарезервировано;;

1.3.14.6.4.6.13Регистр конфигурации канала 2 CH_CFG2

В регистре задается конфигурация работы канала 2 аналогоцифрового преобразователя;

В таблице 1.708 представлено описание полей регистра CH_CFG2.

Таблица 1.708 – Описание полей регистра CH_CFG2

Поле	Название	Программный доступ	Состояние после сброса	Описание
31:20	—	ЧТ0	—	Зарезервировано
19	CH_DTYP	ЧТ/ЗП	0x0	Тип данных канала; 0 - беззнаковые; 1 - знаковые (доп. код);
18	CH_MD	ЧТ/ЗП	0x0	Тип сигнала на входе АЦП; 0 - униполярный сигнал на входе АЦП; 1 - дифференциальный сигнал на входе АЦП;
17:16	CH_RES	ЧТ/ЗП	0x0	Разрядность аналогоцифрового преобразования в канале; 00 - 6 бит; 01 - 8 бит; 10 - 10 бит; 11 - 12 бит;
15:12	CH_OVSS	ЧТ/ЗП	0x0	Величина сдвига вправо после суммирования; 0000 - нет сдвига; 0001 - сдвиг на 1 разряд; 0010 - сдвиг на 2 разряда; … 1111 - сдвиг на 15 разрядов;
11:8	CH_OVSR	ЧТ/ЗП	0x0	Количество суммируемых отсчетов при передискретизации; 0000 - нет передискретизации (1 отсчет); 0001 - 2 отсчета; 0010 - 4 отсчета; … 1010 - 1024 отсчета; 1011 - 1111 - зарезервировано;
7:0	CH_DEL	ЧТ/ЗП	0x0	Время задержки между переключением входного аналогового мультиплексора и началом оцифровки данных канала; 00h - 1 такт задержки; 01h - 2 такта задержки; … feh - 255 тактов задержки; ffh - зарезервировано;;

1.3.14.6.4.6.14Регистр конфигурации канала 3 CH_CFG3

В регистре задается конфигурация работы канала 3 аналогоцифрового преобразователя;

В таблице 1.709 представлено описание полей регистра CH_CFG3.

Таблица 1.709 – Описание полей регистра CH_CFG3

Поле	Название	Программный доступ	Состояние после сброса	Описание
31:20	—	ЧТ0	—	Зарезервировано
19	CH_DTYP	ЧТ/ЗП	0x0	Тип данных канала; 0 - беззнаковые; 1 - знаковые (доп. код);
18	CH_MD	ЧТ/ЗП	0x0	Тип сигнала на входе АЦП; 0 - униполярный сигнал на входе АЦП; 1 - дифференциальный сигнал на входе АЦП;
17:16	CH_RES	ЧТ/ЗП	0x0	Разрядность аналогоцифрового преобразования в канале; 00 - 6 бит; 01 - 8 бит; 10 - 10 бит; 11 - 12 бит;
15:12	CH_OVSS	ЧТ/ЗП	0x0	Величина сдвига вправо после суммирования; 0000 - нет сдвига; 0001 - сдвиг на 1 разряд; 0010 - сдвиг на 2 разряда; … 1111 - сдвиг на 15 разрядов;
11:8	CH_OVSR	ЧТ/ЗП	0x0	Количество суммируемых отсчетов при передискретизации; 0000 - нет передискретизации (1 отсчет); 0001 - 2 отсчета; 0010 - 4 отсчета; … 1010 - 1024 отсчета; 1011 - 1111 - зарезервировано;
7:0	CH_DEL	ЧТ/ЗП	0x0	Время задержки между переключением входного аналогового мультиплексора и началом оцифровки данных канала; 00h - 1 такт задержки; 01h - 2 такта задержки; … feh - 255 тактов задержки; ffh - зарезервировано;;

1.3.14.6.4.6.15 Регистр конфигурации канала 4 CH_CFG4

В регистре задается конфигурация работы канала 4 аналогоцифрового преобразователя;

В таблице 1.710 представлено описание полей регистра CH_CFG4.

Таблица 1.710 – Описание полей регистра CH_CFG4

Поле	Название	Программный доступ	Состояние после сброса	Описание
31:20	—	ЧТ0	—	Зарезервировано
19	CH_DTYP	ЧТ/ЗП	0x0	Тип данных канала; 0 - беззнаковые; 1 - знаковые (доп. код);
18	CH_MD	ЧТ/ЗП	0x0	Тип сигнала на входе АЦП; 0 - униполярный сигнал на входе АЦП; 1 - дифференциальный сигнал на входе АЦП;
17:16	CH_RES	ЧТ/ЗП	0x0	Разрядность аналогоцифрового преобразования в канале; 00 - 6 бит; 01 - 8 бит; 10 - 10 бит; 11 - 12 бит;
15:12	CH_OVSS	ЧТ/ЗП	0x0	Величина сдвига вправо после суммирования; 0000 - нет сдвига; 0001 - сдвиг на 1 разряд; 0010 - сдвиг на 2 разряда; … 1111 - сдвиг на 15 разрядов;
11:8	CH_OVSR	ЧТ/ЗП	0x0	Количество суммируемых отсчетов при передискретизации; 0000 - нет передискретизации (1 отсчет); 0001 - 2 отсчета; 0010 - 4 отсчета; … 1010 - 1024 отсчета; 1011 - 1111 - зарезервировано;
7:0	CH_DEL	ЧТ/ЗП	0x0	Время задержки между переключением входного аналогового мультиплексора и началом оцифровки данных канала; 00h - 1 такт задержки; 01h - 2 такта задержки; … feh - 255 тактов задержки; ffh - зарезервировано;;

1.3.14.6.4.6.16 Регистр конфигурации канала 5 CH_CFG5

В регистре задается конфигурация работы канала 5 аналогоцифрового преобразователя;

В таблице 1.711 представлено описание полей регистра CH_CFG5.

Таблица 1.711 – Описание полей регистра CH_CFG5

Поле	Название	Программный доступ	Состояние после сброса	Описание
31:20	—	ЧТ0	—	Зарезервировано
19	CH_DTYP	ЧТ/ЗП	0x0	Тип данных канала; 0 - беззнаковые; 1 - знаковые (доп. код);
18	CH_MD	ЧТ/ЗП	0x0	Тип сигнала на входе АЦП; 0 - униполярный сигнал на входе АЦП; 1 - дифференциальный сигнал на входе АЦП;
17:16	CH_RES	ЧТ/ЗП	0x0	Разрядность аналогоцифрового преобразования в канале; 00 - 6 бит; 01 - 8 бит; 10 - 10 бит; 11 - 12 бит;
15:12	CH_OVSS	ЧТ/ЗП	0x0	Величина сдвига вправо после суммирования; 0000 - нет сдвига; 0001 - сдвиг на 1 разряд; 0010 - сдвиг на 2 разряда; … 1111 - сдвиг на 15 разрядов;
11:8	CH_OVSR	ЧТ/ЗП	0x0	Количество суммируемых отсчетов при передискретизации; 0000 - нет передискретизации (1 отсчет); 0001 - 2 отсчета; 0010 - 4 отсчета; … 1010 - 1024 отсчета; 1011 - 1111 - зарезервировано;
7:0	CH_DEL	ЧТ/ЗП	0x0	Время задержки между переключением входного аналогового мультиплексора и началом оцифровки данных канала; 00h - 1 такт задержки; 01h - 2 такта задержки; … feh - 255 тактов задержки; ffh - зарезервировано;;

1.3.14.6.4.6.17 Регистр конфигурации канала 6 CH_CFG6

В регистре задается конфигурация работы канала 6 аналогоцифрового преобразователя;

В таблице 1.712 представлено описание полей регистра CH_CFG6.

Таблица 1.712 – Описание полей регистра CH_CFG6

Поле	Название	Программный доступ	Состояние после сброса	Описание
31:20	—	ЧТ0	—	Зарезервировано
19	CH_DTYP	ЧТ/ЗП	0x0	Тип данных канала; 0 - беззнаковые; 1 - знаковые (доп. код);
18	CH_MD	ЧТ/ЗП	0x0	Тип сигнала на входе АЦП; 0 - униполярный сигнал на входе АЦП; 1 - дифференциальный сигнал на входе АЦП;
17:16	CH_RES	ЧТ/ЗП	0x0	Разрядность аналогоцифрового преобразования в канале; 00 - 6 бит; 01 - 8 бит; 10 - 10 бит; 11 - 12 бит;
15:12	CH_OVSS	ЧТ/ЗП	0x0	Величина сдвига вправо после суммирования; 0000 - нет сдвига; 0001 - сдвиг на 1 разряд; 0010 - сдвиг на 2 разряда; … 1111 - сдвиг на 15 разрядов;
11:8	CH_OVSR	ЧТ/ЗП	0x0	Количество суммируемых отсчетов при передискретизации; 0000 - нет передискретизации (1 отсчет); 0001 - 2 отсчета; 0010 - 4 отсчета; … 1010 - 1024 отсчета; 1011 - 1111 - зарезервировано;
7:0	CH_DEL	ЧТ/ЗП	0x0	Время задержки между переключением входного аналогового мультиплексора и началом оцифровки данных канала; 00h - 1 такт задержки; 01h - 2 такта задержки; … feh - 255 тактов задержки; ffh - зарезервировано;;

1.3.14.6.4.6.18 Регистр конфигурации канала 7 CH_CFG7

В регистре задается конфигурация работы канала 7 аналогоцифрового преобразователя;

В таблице 1.713 представлено описание полей регистра CH_CFG7.

Таблица 1.713 – Описание полей регистра CH_CFG7

Поле	Название	Программный доступ	Состояние после сброса	Описание
31:20	—	ЧТ0	—	Зарезервировано
19	CH_DTYP	ЧТ/ЗП	0x0	Тип данных канала; 0 - беззнаковые; 1 - знаковые (доп. код);
18	CH_MD	ЧТ/ЗП	0x0	Тип сигнала на входе АЦП; 0 - униполярный сигнал на входе АЦП; 1 - дифференциальный сигнал на входе АЦП;
17:16	CH_RES	ЧТ/ЗП	0x0	Разрядность аналогоцифрового преобразования в канале; 00 - 6 бит; 01 - 8 бит; 10 - 10 бит; 11 - 12 бит;
15:12	CH_OVSS	ЧТ/ЗП	0x0	Величина сдвига вправо после суммирования; 0000 - нет сдвига; 0001 - сдвиг на 1 разряд; 0010 - сдвиг на 2 разряда; … 1111 - сдвиг на 15 разрядов;
11:8	CH_OVSR	ЧТ/ЗП	0x0	Количество суммируемых отсчетов при передискретизации; 0000 - нет передискретизации (1 отсчет); 0001 - 2 отсчета; 0010 - 4 отсчета; … 1010 - 1024 отсчета; 1011 - 1111 - зарезервировано;
7:0	CH_DEL	ЧТ/ЗП	0x0	Время задержки между переключением входного аналогового мультиплексора и началом оцифровки данных канала; 00h - 1 такт задержки; 01h - 2 такта задержки; … feh - 255 тактов задержки; ffh - зарезервировано;;

1.3.14.6.4.6.19 Регистр конфигурации канала 8 CH_CFG8

В регистре задается конфигурация работы канала 8 аналогоцифрового преобразователя;

В таблице 1.714 представлено описание полей регистра CH_CFG8.

Таблица 1.714 – Описание полей регистра CH_CFG8

Поле	Название	Программный доступ	Состояние после сброса	Описание
31:20	—	ЧТ0	—	Зарезервировано
19	CH_DTYP	ЧТ/ЗП	0x0	Тип данных канала; 0 - беззнаковые; 1 - знаковые (доп. код);
18	CH_MD	ЧТ/ЗП	0x0	Тип сигнала на входе АЦП; 0 - униполярный сигнал на входе АЦП; 1 - дифференциальный сигнал на входе АЦП;
17:16	CH_RES	ЧТ/ЗП	0x0	Разрядность аналогоцифрового преобразования в канале; 00 - 6 бит; 01 - 8 бит; 10 - 10 бит; 11 - 12 бит;
15:12	CH_OVSS	ЧТ/ЗП	0x0	Величина сдвига вправо после суммирования; 0000 - нет сдвига; 0001 - сдвиг на 1 разряд; 0010 - сдвиг на 2 разряда; … 1111 - сдвиг на 15 разрядов;
11:8	CH_OVSR	ЧТ/ЗП	0x0	Количество суммируемых отсчетов при передискретизации; 0000 - нет передискретизации (1 отсчет); 0001 - 2 отсчета; 0010 - 4 отсчета; … 1010 - 1024 отсчета; 1011 - 1111 - зарезервировано;
7:0	CH_DEL	ЧТ/ЗП	0x0	Время задержки между переключением входного аналогового мультиплексора и началом оцифровки данных канала; 00h - 1 такт задержки; 01h - 2 такта задержки; … feh - 255 тактов задержки; ffh - зарезервировано;;

1.3.14.6.4.6.20 Регистр конфигурации канала 9 CH_CFG9

В регистре задается конфигурация работы канала 9 аналогоцифрового преобразователя;

В таблице 1.715 представлено описание полей регистра CH_CFG9.

Таблица 1.715 – Описание полей регистра CH_CFG9

Поле	Название	Программный доступ	Состояние после сброса	Описание
31:20	—	ЧТ0	—	Зарезервировано
19	CH_DTYP	ЧТ/ЗП	0x0	Тип данных канала; 0 - беззнаковые; 1 - знаковые (доп. код);
18	CH_MD	ЧТ/ЗП	0x0	Тип сигнала на входе АЦП; 0 - униполярный сигнал на входе АЦП; 1 - дифференциальный сигнал на входе АЦП;
17:16	CH_RES	ЧТ/ЗП	0x0	Разрядность аналогоцифрового преобразования в канале; 00 - 6 бит; 01 - 8 бит; 10 - 10 бит; 11 - 12 бит;
15:12	CH_OVSS	ЧТ/ЗП	0x0	Величина сдвига вправо после суммирования; 0000 - нет сдвига; 0001 - сдвиг на 1 разряд; 0010 - сдвиг на 2 разряда; … 1111 - сдвиг на 15 разрядов;
11:8	CH_OVSR	ЧТ/ЗП	0x0	Количество суммируемых отсчетов при передискретизации; 0000 - нет передискретизации (1 отсчет); 0001 - 2 отсчета; 0010 - 4 отсчета; … 1010 - 1024 отсчета; 1011 - 1111 - зарезервировано;
7:0	CH_DEL	ЧТ/ЗП	0x0	Время задержки между переключением входного аналогового мультиплексора и началом оцифровки данных канала; 00h - 1 такт задержки; 01h - 2 такта задержки; … feh - 255 тактов задержки; ffh - зарезервировано;;

1.3.14.6.4.6.21Регистр конфигурации канала 10 CH_CFG10

В регистре задается конфигурация работы канала 10 аналогоцифрового преобразователя;

В таблице 1.716 представлено описание полей регистра CH_CFG10.

Таблица 1.716 – Описание полей регистра CH_CFG10

Поле	Название	Программный доступ	Состояние после сброса	Описание
31:20	—	ЧТ0	—	Зарезервировано
19	CH_DTYP	ЧТ/ЗП	0x0	Тип данных канала; 0 - беззнаковые; 1 - знаковые (доп. код);
18	CH_MD	ЧТ/ЗП	0x0	Тип сигнала на входе АЦП; 0 - униполярный сигнал на входе АЦП; 1 - дифференциальный сигнал на входе АЦП;
17:16	CH_RES	ЧТ/ЗП	0x0	Разрядность аналогоцифрового преобразования в канале; 00 - 6 бит; 01 - 8 бит; 10 - 10 бит; 11 - 12 бит;
15:12	CH_OVSS	ЧТ/ЗП	0x0	Величина сдвига вправо после суммирования; 0000 - нет сдвига; 0001 - сдвиг на 1 разряд; 0010 - сдвиг на 2 разряда; … 1111 - сдвиг на 15 разрядов;
11:8	CH_OVSR	ЧТ/ЗП	0x0	Количество суммируемых отсчетов при передискретизации; 0000 - нет передискретизации (1 отсчет); 0001 - 2 отсчета; 0010 - 4 отсчета; … 1010 - 1024 отсчета; 1011 - 1111 - зарезервировано;
7:0	CH_DEL	ЧТ/ЗП	0x0	Время задержки между переключением входного аналогового мультиплексора и началом оцифровки данных канала; 00h - 1 такт задержки; 01h - 2 такта задержки; … feh - 255 тактов задержки; ffh - зарезервировано;;

1.3.14.6.4.6.22 Регистр конфигурации канала 11 CH_CFG11

В регистре задается конфигурация работы канала 11 аналогоцифрового преобразователя;

В таблице 1.717 представлено описание полей регистра CH_CFG11.

Таблица 1.717 – Описание полей регистра CH_CFG11

Поле	Название	Программный доступ	Состояние после сброса	Описание
31:20	—	ЧТ0	—	Зарезервировано
19	CH_DTYP	ЧТ/ЗП	0x0	Тип данных канала; 0 - беззнаковые; 1 - знаковые (доп. код);
18	CH_MD	ЧТ/ЗП	0x0	Тип сигнала на входе АЦП; 0 - униполярный сигнал на входе АЦП; 1 - дифференциальный сигнал на входе АЦП;
17:16	CH_RES	ЧТ/ЗП	0x0	Разрядность аналогоцифрового преобразования в канале; 00 - 6 бит; 01 - 8 бит; 10 - 10 бит; 11 - 12 бит;
15:12	CH_OVSS	ЧТ/ЗП	0x0	Величина сдвига вправо после суммирования; 0000 - нет сдвига; 0001 - сдвиг на 1 разряд; 0010 - сдвиг на 2 разряда; … 1111 - сдвиг на 15 разрядов;
11:8	CH_OVSR	ЧТ/ЗП	0x0	Количество суммируемых отсчетов при передискретизации; 0000 - нет передискретизации (1 отсчет); 0001 - 2 отсчета; 0010 - 4 отсчета; … 1010 - 1024 отсчета; 1011 - 1111 - зарезервировано;
7:0	CH_DEL	ЧТ/ЗП	0x0	Время задержки между переключением входного аналогового мультиплексора и началом оцифровки данных канала; 00h - 1 такт задержки; 01h - 2 такта задержки; … feh - 255 тактов задержки; ffh - зарезервировано;;

1.3.14.6.4.6.23Регистр конфигурации канала 12 CH_CFG12

В регистре задается конфигурация работы канала 12 аналогоцифрового преобразователя;

В таблице 1.718 представлено описание полей регистра CH_CFG12.

Таблица 1.718 – Описание полей регистра CH_CFG12

Поле	Название	Программный доступ	Состояние после сброса	Описание
31:20	—	ЧТ0	—	Зарезервировано
19	CH_DTYP	ЧТ/ЗП	0x0	Тип данных канала; 0 - беззнаковые; 1 - знаковые (доп. код);
18	CH_MD	ЧТ/ЗП	0x0	Тип сигнала на входе АЦП; 0 - униполярный сигнал на входе АЦП; 1 - дифференциальный сигнал на входе АЦП;
17:16	CH_RES	ЧТ/ЗП	0x0	Разрядность аналогоцифрового преобразования в канале; 00 - 6 бит; 01 - 8 бит; 10 - 10 бит; 11 - 12 бит;
15:12	CH_OVSS	ЧТ/ЗП	0x0	Величина сдвига вправо после суммирования; 0000 - нет сдвига; 0001 - сдвиг на 1 разряд; 0010 - сдвиг на 2 разряда; … 1111 - сдвиг на 15 разрядов;
11:8	CH_OVSR	ЧТ/ЗП	0x0	Количество суммируемых отсчетов при передискретизации; 0000 - нет передискретизации (1 отсчет); 0001 - 2 отсчета; 0010 - 4 отсчета; … 1010 - 1024 отсчета; 1011 - 1111 - зарезервировано;
7:0	CH_DEL	ЧТ/ЗП	0x0	Время задержки между переключением входного аналогового мультиплексора и началом оцифровки данных канала; 00h - 1 такт задержки; 01h - 2 такта задержки; … feh - 255 тактов задержки; ffh - зарезервировано;;

1.3.14.6.4.6.24Регистр конфигурации канала 13 CH_CFG13

В регистре задается конфигурация работы канала 13 аналогоцифрового преобразователя;

В таблице 1.719 представлено описание полей регистра CH_CFG13.

Таблица 1.719 – Описание полей регистра CH_CFG13

Поле	Название	Программный доступ	Состояние после сброса	Описание
31:20	—	ЧТ0	—	Зарезервировано
19	CH_DTYP	ЧТ/ЗП	0x0	Тип данных канала; 0 - беззнаковые; 1 - знаковые (доп. код);
18	CH_MD	ЧТ/ЗП	0x0	Тип сигнала на входе АЦП; 0 - униполярный сигнал на входе АЦП; 1 - дифференциальный сигнал на входе АЦП;
17:16	CH_RES	ЧТ/ЗП	0x0	Разрядность аналогоцифрового преобразования в канале; 00 - 6 бит; 01 - 8 бит; 10 - 10 бит; 11 - 12 бит;
15:12	CH_OVSS	ЧТ/ЗП	0x0	Величина сдвига вправо после суммирования; 0000 - нет сдвига; 0001 - сдвиг на 1 разряд; 0010 - сдвиг на 2 разряда; … 1111 - сдвиг на 15 разрядов;
11:8	CH_OVSR	ЧТ/ЗП	0x0	Количество суммируемых отсчетов при передискретизации; 0000 - нет передискретизации (1 отсчет); 0001 - 2 отсчета; 0010 - 4 отсчета; … 1010 - 1024 отсчета; 1011 - 1111 - зарезервировано;
7:0	CH_DEL	ЧТ/ЗП	0x0	Время задержки между переключением входного аналогового мультиплексора и началом оцифровки данных канала; 00h - 1 такт задержки; 01h - 2 такта задержки; … feh - 255 тактов задержки; ffh - зарезервировано;;

1.3.14.6.4.6.25 Регистр конфигурации канала 14 CH_CFG14

В регистре задается конфигурация работы канала 14 аналогоцифрового преобразователя;

В таблице 1.720 представлено описание полей регистра CH_CFG14.

Таблица 1.720 – Описание полей регистра CH_CFG14

Поле	Название	Программный доступ	Состояние после сброса	Описание
31:20	—	ЧТ0	—	Зарезервировано
19	CH_DTYP	ЧТ/ЗП	0x0	Тип данных канала; 0 - беззнаковые; 1 - знаковые (доп. код);
18	CH_MD	ЧТ/ЗП	0x0	Тип сигнала на входе АЦП; 0 - униполярный сигнал на входе АЦП; 1 - дифференциальный сигнал на входе АЦП;
17:16	CH_RES	ЧТ/ЗП	0x0	Разрядность аналогоцифрового преобразования в канале; 00 - 6 бит; 01 - 8 бит; 10 - 10 бит; 11 - 12 бит;
15:12	CH_OVSS	ЧТ/ЗП	0x0	Величина сдвига вправо после суммирования; 0000 - нет сдвига; 0001 - сдвиг на 1 разряд; 0010 - сдвиг на 2 разряда; … 1111 - сдвиг на 15 разрядов;
11:8	CH_OVSR	ЧТ/ЗП	0x0	Количество суммируемых отсчетов при передискретизации; 0000 - нет передискретизации (1 отсчет); 0001 - 2 отсчета; 0010 - 4 отсчета; … 1010 - 1024 отсчета; 1011 - 1111 - зарезервировано;
7:0	CH_DEL	ЧТ/ЗП	0x0	Время задержки между переключением входного аналогового мультиплексора и началом оцифровки данных канала; 00h - 1 такт задержки; 01h - 2 такта задержки; … feh - 255 тактов задержки; ffh - зарезервировано;;

1.3.14.6.4.6.26 Регистр конфигурации канала 15 CH_CFG15

В регистре задается конфигурация работы канала 15 аналогоцифрового преобразователя;

В таблице 1.721 представлено описание полей регистра CH_CFG15.

Таблица 1.721 – Описание полей регистра CH_CFG15

Поле	Название	Программный доступ	Состояние после сброса	Описание
31:20	—	ЧТ0	—	Зарезервировано
19	CH_DTYP	ЧТ/ЗП	0x0	Тип данных канала; 0 - беззнаковые; 1 - знаковые (доп. код);
18	CH_MD	ЧТ/ЗП	0x0	Тип сигнала на входе АЦП; 0 - униполярный сигнал на входе АЦП; 1 - дифференциальный сигнал на входе АЦП;
17:16	CH_RES	ЧТ/ЗП	0x0	Разрядность аналогоцифрового преобразования в канале; 00 - 6 бит; 01 - 8 бит; 10 - 10 бит; 11 - 12 бит;
15:12	CH_OVSS	ЧТ/ЗП	0x0	Величина сдвига вправо после суммирования; 0000 - нет сдвига; 0001 - сдвиг на 1 разряд; 0010 - сдвиг на 2 разряда; … 1111 - сдвиг на 15 разрядов;
11:8	CH_OVSR	ЧТ/ЗП	0x0	Количество суммируемых отсчетов при передискретизации; 0000 - нет передискретизации (1 отсчет); 0001 - 2 отсчета; 0010 - 4 отсчета; … 1010 - 1024 отсчета; 1011 - 1111 - зарезервировано;
7:0	CH_DEL	ЧТ/ЗП	0x0	Время задержки между переключением входного аналогового мультиплексора и началом оцифровки данных канала; 00h - 1 такт задержки; 01h - 2 такта задержки; … feh - 255 тактов задержки; ffh - зарезервировано;;

1.3.14.6.4.6.27 Регистр конфигурации канала 16 CH_CFG16

В регистре задается конфигурация работы канала 16 аналогоцифрового преобразователя;

В таблице 1.722 представлено описание полей регистра CH_CFG16.

Таблица 1.722 – Описание полей регистра CH_CFG16

Поле	Название	Программный доступ	Состояние после сброса	Описание
31:20	—	ЧТ0	—	Зарезервировано
19	CH_DTYP	ЧТ/ЗП	0x0	Тип данных канала; 0 - беззнаковые; 1 - знаковые (доп. код);
18	CH_MD	ЧТ/ЗП	0x0	Тип сигнала на входе АЦП; 0 - униполярный сигнал на входе АЦП; 1 - дифференциальный сигнал на входе АЦП;
17:16	CH_RES	ЧТ/ЗП	0x0	Разрядность аналогоцифрового преобразования в канале; 00 - 6 бит; 01 - 8 бит; 10 - 10 бит; 11 - 12 бит;
15:12	CH_OVSS	ЧТ/ЗП	0x0	Величина сдвига вправо после суммирования; 0000 - нет сдвига; 0001 - сдвиг на 1 разряд; 0010 - сдвиг на 2 разряда; … 1111 - сдвиг на 15 разрядов;
11:8	CH_OVSR	ЧТ/ЗП	0x0	Количество суммируемых отсчетов при передискретизации; 0000 - нет передискретизации (1 отсчет); 0001 - 2 отсчета; 0010 - 4 отсчета; … 1010 - 1024 отсчета; 1011 - 1111 - зарезервировано;
7:0	CH_DEL	ЧТ/ЗП	0x0	Время задержки между переключением входного аналогового мультиплексора и началом оцифровки данных канала; 00h - 1 такт задержки; 01h - 2 такта задержки; … feh - 255 тактов задержки; ffh - зарезервировано;;

1.3.14.6.4.6.28 Регистр конфигурации канала 17 CH_CFG17

В регистре задается конфигурация работы канала 17 аналогоцифрового преобразователя;

В таблице 1.723 представлено описание полей регистра CH_CFG17.

Таблица 1.723 – Описание полей регистра CH_CFG17

Поле	Название	Программный доступ	Состояние после сброса	Описание
31:20	—	ЧТ0	—	Зарезервировано
19	CH_DTYP	ЧТ/ЗП	0x0	Тип данных канала; 0 - беззнаковые; 1 - знаковые (доп. код);
18	CH_MD	ЧТ/ЗП	0x0	Тип сигнала на входе АЦП; 0 - униполярный сигнал на входе АЦП; 1 - дифференциальный сигнал на входе АЦП;
17:16	CH_RES	ЧТ/ЗП	0x0	Разрядность аналогоцифрового преобразования в канале; 00 - 6 бит; 01 - 8 бит; 10 - 10 бит; 11 - 12 бит;
15:12	CH_OVSS	ЧТ/ЗП	0x0	Величина сдвига вправо после суммирования; 0000 - нет сдвига; 0001 - сдвиг на 1 разряд; 0010 - сдвиг на 2 разряда; … 1111 - сдвиг на 15 разрядов;
11:8	CH_OVSR	ЧТ/ЗП	0x0	Количество суммируемых отсчетов при передискретизации; 0000 - нет передискретизации (1 отсчет); 0001 - 2 отсчета; 0010 - 4 отсчета; … 1010 - 1024 отсчета; 1011 - 1111 - зарезервировано;
7:0	CH_DEL	ЧТ/ЗП	0x0	Время задержки между переключением входного аналогового мультиплексора и началом оцифровки данных канала; 00h - 1 такт задержки; 01h - 2 такта задержки; … feh - 255 тактов задержки; ffh - зарезервировано;;

1.3.14.6.4.6.29 Регистр конфигурации канала 18 CH_CFG18

В регистре задается конфигурация работы канала 18 аналогоцифрового преобразователя;

В таблице 1.724 представлено описание полей регистра CH_CFG18.

Таблица 1.724 – Описание полей регистра CH_CFG18

Поле	Название	Программный доступ	Состояние после сброса	Описание
31:20	—	ЧТ0	—	Зарезервировано
19	CH_DTYP	ЧТ/ЗП	0x0	Тип данных канала; 0 - беззнаковые; 1 - знаковые (доп. код);
18	CH_MD	ЧТ/ЗП	0x0	Тип сигнала на входе АЦП; 0 - униполярный сигнал на входе АЦП; 1 - дифференциальный сигнал на входе АЦП;
17:16	CH_RES	ЧТ/ЗП	0x0	Разрядность аналогоцифрового преобразования в канале; 00 - 6 бит; 01 - 8 бит; 10 - 10 бит; 11 - 12 бит;
15:12	CH_OVSS	ЧТ/ЗП	0x0	Величина сдвига вправо после суммирования; 0000 - нет сдвига; 0001 - сдвиг на 1 разряд; 0010 - сдвиг на 2 разряда; … 1111 - сдвиг на 15 разрядов;
11:8	CH_OVSR	ЧТ/ЗП	0x0	Количество суммируемых отсчетов при передискретизации; 0000 - нет передискретизации (1 отсчет); 0001 - 2 отсчета; 0010 - 4 отсчета; … 1010 - 1024 отсчета; 1011 - 1111 - зарезервировано;
7:0	CH_DEL	ЧТ/ЗП	0x0	Время задержки между переключением входного аналогового мультиплексора и началом оцифровки данных канала; 00h - 1 такт задержки; 01h - 2 такта задержки; … feh - 255 тактов задержки; ffh - зарезервировано;;

1.3.14.6.4.6.30 Регистр программного старта и останова аналогоцифрового преобразования SEC_START

Запись логической единицы в соответствующие разряды данного регистр запускает или останавливает процедуру цифроаналогового преобразования. После старта или останова разряды данного регистра сбрасывается аппаратно;

В таблице 1.725 представлено описание полей регистра SEC_START.

Таблица 1.725 – Описание полей регистра SEC_START

Поле	Название	Программный доступ	Состояние после сброса	Описание
31:6	—	ЧТ0	—	Зарезервировано
5	STP_DED	ЧТ/ЗП	0x0	Программный останов опроса выделенного канала;
4	STP_PRI	ЧТ/ЗП	0x0	Программный останов приоритетной последовательности;
3	STP_ORD	ЧТ/ЗП	0x0	Программный останов ординарной последовательности;
2	STT_DED	ЧТ/ЗП	0x0	Программный старт опроса выделенного канала;
1	STT_PRI	ЧТ/ЗП	0x0	Программный старт приоритетной последовательности;
0	STT_ORD	ЧТ/ЗП	0x0	Программный старт ординарной последовательности;

1.3.14.6.4.6.31 Регистр конфигурации запуска аналогоцифрового преобразования SEC_ST_CFG

В данном регистре задаются параметры запуска аналогоцифрового преобразования и поведение аппаратуры при переполнении выходного буфера последовательностей;

В таблице 1.726 представлено описание полей регистра SEC_ST_CFG.

Таблица 1.726 – Описание полей регистра SEC_ST_CFG

Поле	Название	Программный доступ	Состояние после сброса	Описание
31:5	—	ЧТ0	—	Зарезервировано
4	SEC_PRI_OVER	ЧТ/ЗП	0x0	Поведение выходного буфера данных приоритетной последовательности при переполнении; 0 - игнорирование следующих за переполнением данных; 1 - сдвиг буфера на один отсчет (т.е. удаление самых старых данных);
3	SEC_ORD_OVER	ЧТ/ЗП	0x0	Поведение выходного буфера данных ординарной последовательности при переполнении; 0 - игнорирование следующих за переполнением данных; 1 - сдвиг буфера на один отсчет (т.е. удаление самых старых данных);
2	SEC_DED_ACT	ЧТ/ЗП	0x0	Разрешение аппаратного запуска опроса выделенного канала; 0 - аппаратный запуск запрещен; 1 - аппаратный запуск разрешен;

Поле	Название	Программный доступ	Состояние после сброса	Описание
1	SEC_PRI_ACT	ЧТ/ЗП	0x0	Разрешение аппаратного запуска приоритетной последовательности; 0 - аппаратный запуск запрещен; 1 - аппаратный запуск разрешен;
0	SEC_ORD_ACT	ЧТ/ЗП	0x0	Разрешение аппаратного запуска ординарной последовательности; 0 - аппаратный запуск запрещен; 1 - аппаратный запуск разрешен;

1.3.14.6.4.6.32 Регистр размера ординарной последовательности SEC_ORD_NUM

В регистре задается количество каналов в ординарной последовательности;

В таблице 1.727 представлено описание полей регистра SEC_ORD_NUM.

Таблица 1.727 – Описание полей регистра SEC_ORD_NUM

Поле	Название	Программный доступ	Состояние после сброса	Описание
31:5	—	ЧТ0	—	Зарезервировано
4:0	ORD_NUM	ЧТ/ЗП	0x0	Размер ординарной последовательности; 00000 - 1 канал в последовательности; 00001 - 2 канала в последовательности; … 10011 - 20 каналов в последовательности; 10100 - 11111 - зарезервировано;;

1.3.14.6.4.6.33 Регистр 0 конфигурации ординарной последовательности SEC_ORD_CFG0

В регистре задаются порядковые номера каналов в ординарной последовательности;

В таблице 1.728 представлено описание полей регистра SEC_ORD_CFG0.

Таблица 1.728 – Описание полей регистра SEC_ORD_CFG0

Поле	Название	Программный доступ	Состояние после сброса	Описание
31:30	—	ЧТ0	—	Зарезервировано
29:25	ORD_NUM6	ЧТ/ЗП	0x0	Номер шестого канала в последовательности;
24:20	ORD_NUM5	ЧТ/ЗП	0x0	Номер пятого канала в последовательности;
19:15	ORD_NUM4	ЧТ/ЗП	0x0	Номер четвертого канала в последовательности;
14:10	ORD_NUM3	ЧТ/ЗП	0x0	Номер третьего канала в последовательности;
9:5	ORD_NUM2	ЧТ/ЗП	0x0	Номер второго канала в последовательности;

Поле	Название	Программный доступ	Состояние после сброса	Описание
4:0	ORD_NUM1	ЧТ/ЗП	0x0	Номер первого канала в последовательности;

1.3.14.6.4.6.34 Регистр 1 конфигурации ординарной последовательности SEC_ORD_CFG1

В регистре задаются порядковые номера каналов в ординарной последовательности;

В таблице 1.729 представлено описание полей регистра SEC_ORD_CFG1.

Таблица 1.729 – Описание полей регистра SEC_ORD_CFG1

Поле	Название	Программный доступ	Состояние после сброса	Описание
31:30	—	ЧТ0	—	Зарезервировано
29:25	ORD_NUM12	ЧТ/ЗП	0x0	Номер двенадцатого канала в последовательности;
24:20	ORD_NUM11	ЧТ/ЗП	0x0	Номер одиннадцатого канала в последовательности;
19:15	ORD_NUM10	ЧТ/ЗП	0x0	Номер десятого канала в последовательности;
14:10	ORD_NUM9	ЧТ/ЗП	0x0	Номер девятого канала в последовательности;
9:5	ORD_NUM8	ЧТ/ЗП	0x0	Номер восьмого канала в последовательности;
4:0	ORD_NUM7	ЧТ/ЗП	0x0	Номер седьмого канала в последовательности;

1.3.14.6.4.6.35 Регистр 2 конфигурации ординарной последовательности SEC_ORD_CFG2

В регистре задаются порядковые номера каналов в ординарной последовательности;

В таблице 1.730 представлено описание полей регистра SEC_ORD_CFG2.

Таблица 1.730 – Описание полей регистра SEC_ORD_CFG2

Поле	Название	Программный доступ	Состояние после сброса	Описание
31:30	—	ЧТ0	—	Зарезервировано
29:25	ORD_NUM18	ЧТ/ЗП	0x0	Номер восемнадцатого канала в последовательности;
24:20	ORD_NUM17	ЧТ/ЗП	0x0	Номер семнадцатого канала в последовательности;
19:15	ORD_NUM16	ЧТ/ЗП	0x0	Номер шестнадцатого канала в последовательности;
14:10	ORD_NUM15	ЧТ/ЗП	0x0	Номер пятнадцатого канала в последовательности;

Поле	Название	Программный доступ	Состояние после сброса	Описание
9:5	ORD_NUM14	ЧТ/ЗП	0x0	Номер четырнадцатого канала в последовательности;
4:0	ORD_NUM13	ЧТ/ЗП	0x0	Номер тринадцатого канала в последовательности;

1.3.14.6.4.6.36 Регистр 3 конфигурации ординарной последовательности SEC_ORD_CFG3

В регистре задаются порядковые номера каналов в ординарной последовательности;

В таблице 1.731 представлено описание полей регистра SEC_ORD_CFG3.

Таблица 1.731 – Описание полей регистра SEC_ORD_CFG3

Поле	Название	Программный доступ	Состояние после сброса	Описание
31:10	—	ЧТ0	—	Зарезервировано
9:5	ORD_NUM20	ЧТ/ЗП	0x0	Номер двадцатого канала в последовательности;
4:0	ORD_NUM19	ЧТ/ЗП	0x0	Номер девятнадцатого канала в последовательности;

1.3.14.6.4.6.37 Регистр разрешения записи данных канала в выходной буфер SEC_ORD_CHWE

В регистре для каждого канала задается разрешение записи данных канала в буфер ординарной последовательности;

В таблице 1.732 представлено описание полей регистра SEC_ORD_CHWE.

Таблица 1.732 – Описание полей регистра SEC_ORD_CHWE

Поле	Название	Программный доступ	Состояние после сброса	Описание
31:19	—	ЧТ0	—	Зарезервировано
18:0	ORD_CHWE	ЧТ/ЗП	0x0	Установка/сброс разряда регистра запрещает/разрешает запись данных канала, номер которого соответствует номеру разряда регистра; 0 - данные канала записываются в буфер; 1 - данные канала не записываются в буфер;

1.3.14.6.4.6.38 Регистр размера приоритетной последовательности SEC_PRI_NUM

В регистре задается количество каналов в приоритетной последовательности;

В таблице 1.733 представлено описание полей регистра SEC_PRI_NUM.

Таблица 1.733 – Описание полей регистра SEC_PRI_NUM

Поле	Название	Программный доступ	Состояние после сброса	Описание
31:5	—	ЧТ0	—	Зарезервировано
4:0	PRI_NUM	ЧТ/ЗП	0x0	Размер приоритетной последовательности; 00000 - 1 канал в последовательности; 00001 - 2 канала в последовательности; … 10011 - 20 каналов в последовательности; 10100 - 11111 - зарезервировано;;

1.3.14.6.4.6.39 Регистр 0 конфигурации приоритетной последовательности SEC_PRI_CFG0

В регистре задаются порядковые номера каналов в приоритетной последовательности;

В таблице 1.734 представлено описание полей регистра SEC_PRI_CFG0.

Таблица 1.734 – Описание полей регистра SEC_PRI_CFG0

Поле	Название	Программный доступ	Состояние после сброса	Описание
31:30	—	ЧТ0	—	Зарезервировано
29:25	PRI_NUM6	ЧТ/ЗП	0x0	Номер шестого канала в последовательности;
24:20	PRI_NUM5	ЧТ/ЗП	0x0	Номер пятого канала в последовательности;
19:15	PRI_NUM4	ЧТ/ЗП	0x0	Номер четвертого канала в последовательности;
14:10	PRI_NUM3	ЧТ/ЗП	0x0	Номер третьего канала в последовательности;
9:5	PRI_NUM2	ЧТ/ЗП	0x0	Номер второго канала в последовательности;
4:0	PRI_NUM1	ЧТ/ЗП	0x0	Номер первого канала в последовательности;

1.3.14.6.4.6.40 Регистр 1 конфигурации приоритетной последовательности SEC_PRI_CFG1

В регистре задаются порядковые номера каналов в приоритетной последовательности;

В таблице 1.735 представлено описание полей регистра SEC_PRI_CFG1.

Таблица 1.735 – Описание полей регистра SEC_PRI_CFG1

Поле	Название	Программный доступ	Состояние после сброса	Описание
31:30	—	ЧТ0	—	Зарезервировано
29:25	PRI_NUM12	ЧТ/ЗП	0x0	Номер двенадцатого канала в последовательности;
24:20	PRI_NUM11	ЧТ/ЗП	0x0	Номер одиннадцатого канала в последовательности;
19:15	PRI_NUM10	ЧТ/ЗП	0x0	Номер десятого канала в последовательности;
14:10	PRI_NUM9	ЧТ/ЗП	0x0	Номер девятого канала в последовательности;
9:5	PRI_NUM8	ЧТ/ЗП	0x0	Номер восьмого канала в последовательности;
4:0	PRI_NUM7	ЧТ/ЗП	0x0	Номер седьмого канала в последовательности;

1.3.14.6.4.6.41 Регистр 2 конфигурации приоритетной последовательности SEC_PRI_CFG2

В регистре задаются порядковые номера каналов в приоритетной последовательности;

В таблице 1.736 представлено описание полей регистра SEC_PRI_CFG2.

Таблица 1.736 – Описание полей регистра SEC_PRI_CFG2

Поле	Название	Программный доступ	Состояние после сброса	Описание
31:30	—	ЧТ0	—	Зарезервировано
29:25	PRI_NUM18	ЧТ/ЗП	0x0	Номер восемнадцатого канала в последовательности;
24:20	PRI_NUM17	ЧТ/ЗП	0x0	Номер семнадцатого канала в последовательности;
19:15	PRI_NUM16	ЧТ/ЗП	0x0	Номер шестнадцатого канала в последовательности;
14:10	PRI_NUM15	ЧТ/ЗП	0x0	Номер пятнадцатого канала в последовательности;
9:5	PRI_NUM14	ЧТ/ЗП	0x0	Номер четырнадцатого канала в последовательности;
4:0	PRI_NUM13	ЧТ/ЗП	0x0	Номер тринадцатого канала в последовательности;

1.3.14.6.4.6.42 Регистр 3 конфигурации приоритетной последовательности SEC_PRI_CFG3

В регистре задаются порядковые номера каналов в приоритетной последовательности;

В таблице 1.737 представлено описание полей регистра SEC_PRI_CFG3.

Таблица 1.737 – Описание полей регистра SEC_PRI_CFG3

Поле	Название	Программный доступ	Состояние после сброса	Описание
31:10	—	ЧТ0	—	Зарезервировано
9:5	PRI_NUM20	ЧТ/ЗП	0x0	Номер двадцатого канала в последовательности;
4:0	PRI_NUM19	ЧТ/ЗП	0x0	Номер девятнадцатого канала в последовательности;

1.3.14.6.4.6.43 Регистр разрешения записи данных канала в выходной буфер SEC_PRI_CHWE

В регистре для каждого канала задается разрешение записи данных канала в буфер приоритетной последовательности;

В таблице 1.738 представлено описание полей регистра SEC_PRI_CHWE.

Таблица 1.738 – Описание полей регистра SEC_PRI_CHWE

Поле	Название	Программный доступ	Состояние после сброса	Описание
31:19	—	ЧТ0	—	Зарезервировано
18:0	PRI_CHWE	ЧТ/ЗП	0x0	Установка/сброс разряда регистра запрещает/разрещает запись данных канала, номер которого соответствует номеру разряда регистра; 0 - данные канала записываются в буфер; 1 - данные канала не записываются в буфер;

1.3.14.6.4.6.44 Регистр конфигурации выделенного канала CH_DED_CFG

В регистре задается номер выделенного канала;

В таблице 1.739 представлено описание полей регистра CH_DED_CFG.

Таблица 1.739 – Описание полей регистра CH_DED_CFG

Поле	Название	Программный доступ	Состояние после сброса	Описание
31:5	—	ЧТ0	—	Зарезервировано
4:0	DED_NUM	ЧТ/ЗП	0x0	Номер выделенного канала;

1.3.14.6.4.6.45 Регистр статуса аналогоцифрового преобразования SEC_STS

В регистре отражено состяние аналогоцифрового преобразования;

В таблице 1.740 представлено описание полей регистра SEC_STS.

Таблица 1.740 – Описание полей регистра SEC_STS

Поле	Название	Программный доступ	Состояние после сброса	Описание
31:3	—	ЧТ0	—	Зарезервировано
2	DED_STS	ЧТ	0x0	Статус опроса выделенного канала: 0 – опрос не запущен; 1 – опрос выполняется;
1	PRI_STS	ЧТ	0x0	Статус приоритетной последовательности: 0 - последовательность не запущена; 1 - последовательность выполняется ;
0	ORD_STS	ЧТ	0x0	Статус ординарной последовательности: 0 - последовательность не запущена; 1 - последовательность выполняется ;

1.3.14.6.4.6.46 Регистр разрешения работы компараторов COMP_EN

Запись логической 1единицы в рвзряды регистра разрешает работу компараторов

В таблице 1.741 представлено описание полей регистра COMP_EN.

Таблица 1.741 – Описание полей регистра COMP_EN

Поле	Название	Программный доступ	Состояние после сброса	Описание
31:6	—	ЧТ0	—	Зарезервировано
5	CMP2_OEN	ЧТ/ЗП	0x0	Разрешение выдачи сигнала с компаратора 2 на внешний вывод блока: 0 - выдача запрещена; 1 - выдача разрешена;
4	CMP1_OEN	ЧТ/ЗП	0x0	Разрешение выдачи сигнала с компаратора 1 на внешний вывод блока: 0 - выдача запрещена; 1 - выдача разрешена;
3	CMP0_OEN	ЧТ/ЗП	0x0	Разрешение выдачи сигнала с компаратора 0 на внешний вывод блока: 0 - выдача запрещена; 1 - выдача разрешена;
2	CMP2_EN	ЧТ/ЗП	0x0	Разрешение работы компаратора 2: 0 - компаратор не работает; 1 - компаратор работает;
1	CMP1_EN	ЧТ/ЗП	0x0	Разрешение работы компаратора 1: 0 - компаратор не работает; 1 - компаратор работает;

Поле	Название	Программный доступ	Состояние после сброса	Описание
0	CMP0_EN	ЧТ/ЗП	0x0	Разрешение работы компаратора 0: 0 - компаратор не работает; 1 - компаратор работает;

1.3.14.6.4.6.47 Регистр установки разрешения работы компараторов COMP_EN_set

Запись логической единицы в разряды регистра устанавливает соответсвующий бит регистра COMP_EN в единицу;

В таблице 1.742 представлено описание полей регистра COMP_EN_set.

Таблица 1.742 – Описание полей регистра COMP_EN_set

Поле	Название	Программный доступ	Состояние после сброса	Описание
31:6	—	ЧТ0	—	Зарезервировано
5	CMP2_OEN_set	ЗП	0x0	Установка в 1 разрешения выдачи сигнала с компаратора 2 на внешний вывод блока;
4	CMP1_OEN_set	ЗП	0x0	Установка в 1 разрешения выдачи сигнала с компаратора 1 на внешний вывод блока;
3	CMP0_OEN_set	ЗП	0x0	Установка в 1 разрешения выдачи сигнала с компаратора 0 на внешний вывод блока;
2	CMP2_EN_set	ЗП	0x0	Установка в 1 разрешения работы компаратора 2;
1	CMP1_EN_set	ЗП	0x0	Установка в 1 разрешения работы компаратора 1;
0	CMP0_EN_set	ЗП	0x0	Установка в 1 разрешения работы компаратора 0;

1.3.14.6.4.6.48 Регистр сброса разрешения работы компараторов COMP_EN_rst

Запись логической единицы в разряды регистра сбрасывает соответсвующий бит регистра COMP_EN в ноль;

В таблице 1.743 представлено описание полей регистра COMP_EN_rst.

Таблица 1.743 – Описание полей регистра COMP_EN_rst

Поле	Название	Программный доступ	Состояние после сброса	Описание
31:6	—	ЧТ0	—	Зарезервировано
5	CMP2_OEN_rst	ЗП	0x0	Сброс в 0 разрешения выдачи сигнала с компаратора 2 на внешний вывод блока;
4	CMP1_OEN_rst	ЗП	0x0	Сброс в 0 разрешения выдачи сигнала с компаратора 1 на внешний вывод блока;

Поле	Название	Программный доступ	Состояние после сброса	Описание
3	CMP0_OEN_rst	ЗП	0x0	Сброс в 0 разрешения выдачи сигнала с компаратора 0 на внешний вывод блока;
2	CMP2_EN_rst	ЗП	0x0	Сброс в 0 разрешения работы компаратора 2;
1	CMP1_EN_rst	ЗП	0x0	Сброс в 0 разрешения работы компаратора 1;
0	CMP0_EN_rst	ЗП	0x0	Сброс в 0 разрешения работы компаратора 0;

1.3.14.6.4.6.49 Регистр конфигурации компаратора 0 COMP_CFG0

В регистре задается конфигурация компаратора 0;

В таблице 1.744 представлено описание полей регистра COMP_CFG0.

Таблица 1.744 – Описание полей регистра COMP_CFG0

Поле	Название	Программный доступ	Состояние после сброса	Описание
31:23	—	ЧТ0	—	Зарезервировано
22	CMP_DT0	ЧТ/ЗП	0x0	Тип данных, с которым работает компаратор; 0 - прямой код; 1 - дополнительный код;
21	CMP_DED0	ЧТ/ЗП	0x0	Сравнение данных каналов при при опросе выделенного канала; 0 - сравнивание не производится; 1 - сравнивание производится;
20	CMP_PRI0	ЧТ/ЗП	0x0	Сравнение данных каналов при выполнении приоритетной последовательности; 0 - сравнивание не производится; 1 - сравнивание производится;
19	CMP_ORD0	ЧТ/ЗП	0x0	Сравнение данных каналов при выполнении ординарной последовательности; 0 - сравнивание не производится; 1 - сравнивание производится;
18:0	CMP_CH_NUM0	ЧТ/ЗП	0x0	Установка разряда регистра определяет номер канала с которым работает компаратор; 0 - компаратор не работает с данными канала; 1 - компаратор работает с данными канала;

1.3.14.6.4.6.50 Регистр верхнего порога сравнения компаратора 0 COMP_TRE_H0

В регистре задается верхний порог сравнения компаратора 0;

В таблице 1.745 представлено описание полей регистра COMP_TRE_H0.

Таблица 1.745 – Описание полей регистра COMP_TRE_H0

Поле	Название	Программный доступ	Состояние после сброса	Описание
31:24	—	ЧТ0	—	Зарезервировано
23:0	TRE_H0	ЧТ/ЗП	0x0	Верхний порог сравнения компаратора;

1.3.14.6.4.6.51 Регистр нижнего порога сравнения компаратора 0 COMP_TRE_L0

В регистре задается нижний порог сравнения компаратора 0;

В таблице 1.746 представлено описание полей регистра COMP_TRE_L0.

Таблица 1.746 – Описание полей регистра COMP_TRE_L0

Поле	Название	Программный доступ	Состояние после сброса	Описание
31:24	—	ЧТ0	—	Зарезервировано
23:0	TRE_L0	ЧТ/ЗП	0x0	Нижний порог сравнения компаратора;

1.3.14.6.4.6.52 Регистр конфигурации компаратора 1 COMP_CFG1

В регистре задается конфигурация компаратора 1;

В таблице 1.747 представлено описание полей регистра COMP_CFG1.

Таблица 1.747 – Описание полей регистра COMP_CFG1

Поле	Название	Программный доступ	Состояние после сброса	Описание
31:23	—	ЧТ0	—	Зарезервировано
22	CMP_DT1	ЧТ/ЗП	0x0	Тип данных, с которым работает компаратор; 0 - прямой код; 1 - дополнительный код;
21	CMP_DED1	ЧТ/ЗП	0x0	Сравнение данных каналов при при опросе выделенного канала; 0 - сравнивание не производится; 1 - сравнивание производится;
20	CMP_PRI1	ЧТ/ЗП	0x0	Сравнение данных каналов при выполнении приоритетной последовательности; 0 - сравнивание не производится; 1 - сравнивание производится;
19	CMP_ORD1	ЧТ/ЗП	0x0	Сравнение данных каналов при выполнении ординарной последовательности; 0 - сравнивание не производится; 1 - сравнивание производится;

Поле	Название	Программный доступ	Состояние после сброса	Описание
18:0	CMP_CH_NUM1	ЧТ/ЗП	0x0	Установка разряда регистра определяет номер канала с которым работает компаратор; 0 - компаратор не работает с данными канала; 1 - компаратор работает с данными канала;

1.3.14.6.4.6.53 Регистр верхнего порога сравнения компаратора 1 COMP_TRE_H1

В регистре задается верхний порог сравнения компаратора 1;

В таблице 1.748 представлено описание полей регистра COMP_TRE_H1.

Таблица 1.748 – Описание полей регистра COMP_TRE_H1

Поле	Название	Программный доступ	Состояние после сброса	Описание
31:24	—	ЧТ0	—	Зарезервировано
23:0	TRE_H1	ЧТ/ЗП	0x0	Верхний порог сравнения компаратора;

1.3.14.6.4.6.54 Регистр нижнего порога сравнения компаратора 1 COMP_TRE_L1

В регистре задается нижний порог сравнения компаратора 1;

В таблице 1.749 представлено описание полей регистра COMP_TRE_L1.

Таблица 1.749 – Описание полей регистра COMP_TRE_L1

Поле	Название	Программный доступ	Состояние после сброса	Описание
31:24	—	ЧТ0	—	Зарезервировано
23:0	TRE_L1	ЧТ/ЗП	0x0	Нижний порог сравнения компаратора;

1.3.14.6.4.6.55 Регистр конфигурации компаратора 2 COMP_CFG2

В регистре задается конфигурация компаратора 2;

В таблице 1.750 представлено описание полей регистра COMP_CFG2.

Таблица 1.750 – Описание полей регистра COMP_CFG2

Поле	Название	Программный доступ	Состояние после сброса	Описание
31:23	—	ЧТ0	—	Зарезервировано

Поле	Название	Программный доступ	Состояние после сброса	Описание
22	CMP_DT2	ЧТ/ЗП	0x0	Тип данных, с которым работает компаратор; 0 - прямой код; 1 - дополнительный код;
21	CMP_DED2	ЧТ/ЗП	0x0	Сравнение данных каналов при при опросе выделенного канала; 0 - сравнивание не производится; 1 - сравнивание производится;
20	CMP_PRI2	ЧТ/ЗП	0x0	Сравнение данных каналов при выполнении приоритетной последовательности; 0 - сравнивание не производится; 1 - сравнивание производится;
19	CMP_ORD2	ЧТ/ЗП	0x0	Сравнение данных каналов при выполнении ординарной последовательности; 0 - сравнивание не производится; 1 - сравнивание производится;
18:0	CMP_CH_NUM2	ЧТ/ЗП	0x0	Установка разряда регистра определяет номер канала с которым работает компаратор; 0 - компаратор не работает с данными канала; 1 - компаратор работает с данными канала;

1.3.14.6.4.6.56 Регистр верхнего порога сравнения компаратора 2 COMP_TRE_H2

В регистре задается верхний порог сравнения компаратора 2;

В таблице 1.751 представлено описание полей регистра COMP_TRE_H2.

Таблица 1.751 – Описание полей регистра COMP_TRE_H2

Поле	Название	Программный доступ	Состояние после сброса	Описание
31:24	—	ЧТ0	—	Зарезервировано
23:0	TRE_H2	ЧТ/ЗП	0x0	Верхний порог сравнения компаратора;

1.3.14.6.4.6.57 Регистр нижнего порога сравнения компаратора 2 COMP_TRE_L2

В регистре задается нижний порог сравнения компаратора 2;

В таблице 1.752 представлено описание полей регистра COMP_TRE_L2.

Таблица 1.752 – Описание полей регистра COMP_TRE_L2

Поле	Название	Программный доступ	Состояние после сброса	Описание
31:24	—	ЧТ0	—	Зарезервировано
23:0	TRE_L2	ЧТ/ЗП	0x0	Нижний порог сравнения компаратора;

1.3.14.6.4.6.58 Регистр управления режимом минимального потребления (Deep_Power_down) PWR_DPD

Регистре позволяет включать и выключать режим минимального потребления (Deep_Power_down);

В таблице 1.753 представлено описание полей регистра PWR_DPD.

Таблица 1.753 – Описание полей регистра PWR_DPD

Поле	Название	Программный доступ	Состояние после сброса	Описание
31:1	—	ЧТ0	—	Зарезервировано
0	PWR_DPD	ЧТ/ЗП	0x1	Управление режимом минимального потребления (Deep_Power_down); 0 - режим не активирован; 1 - режим активирован;

1.3.14.6.4.6.59 Регистр управления включением источника опорного напряжения (LDO) LDO_CTRL

Регистр позволяет включать и выключать источник опорного напряжения (LDO) ;

В таблице 1.754 представлено описание полей регистра LDO_CTRL.

Таблица 1.754 – Описание полей регистра LDO_CTRL

Поле	Название	Программный доступ	Состояние после сброса	Описание
31:2	—	ЧТ0	—	Зарезервировано
1	LDO_EN	ЧТ/ЗП	0x0	Управление включением источника опорного напряжения (LDO) ; 0 – источник опорного напряжения не активирован; 1 – источник опорного напряжения активизируется;
0	Reserved	-	0x0	зарезервировано;

1.3.14.6.4.6.60 Регистр включения рабочего режима АЦП PWR_ENADC

Регистр управляет включением рабочего режима АЦП ;

В таблице 1.755 представлено описание полей регистра PWR_ENADC.

Таблица 1.755 – Описание полей регистра PWR_ENADC

Поле	Название	Программный доступ	Состояние после сброса	Описание
31:3	—	ЧТ0	—	Зарезервировано
2	ADC_EN	ЧТ/ЗП	0x0	Управление включением рабочего режима АЦП; 0 – рабочий режим выключен; 1 – рабочий режим включен;
1:0	Reserved	-	0x0	зарезервировано;

1.3.14.6.4.6.61 Регистр системного сброса АЦП PWR_RSTADC

Запись логической единицы в поле регистра приводит к активизации сигнала сброса АЦП. Поле сбрасывается аппаратно после выполнения процедуры сброса.;

В таблице 1.756 представлено описание полей регистра PWR_RSTADC.

Таблица 1.756 – Описание полей регистра PWR_RSTADC

Поле	Название	Программный доступ	Состояние после сброса	Описание
31:4	—	ЧТ0	—	Зарезервировано
3	ADC_RST	ЧТ/ЗП	0x0	Управление подачей сигнала сброса АЦП; 0 – сигнала сброса АЦП не активен или началась процедура снятия сигнала сброса АЦП; 1 – сигнала сброса АЦП активен;
2:0	Reserved	-	0x0	зарезервировано;

1.3.14.6.4.6.62 Регистр управления стартом конечного автомата калибровки АЦП CAL_STRT

Запись логической единицы в поле регистра вызывает старт конечного автомата калибровки АЦП. Сбрасывается аппаратно в процессе выполнения процедуры калибровки;

В таблице 1.757 представлено описание полей регистра CAL_STRT.

Таблица 1.757 – Описание полей регистра CAL_STRT

Поле	Название	Программный доступ	Состояние после сброса	Описание
31:1	—	ЧТ0	—	Зарезервировано
0	CAL_STT	ЧТ/ЗП	0x0	Управление стартом конечного автомата калибровки АЦП;

1.3.14.6.4.6.63 Регистр калибровочных коэффициентов CAL_DATA

Регистр сохраняет последние значения калибровочных коэффициентов, полученные в результате выполнения процедуры калибровки АЦП или программно записываемые пользователем.;

В таблице 1.758 представлено описание полей регистра CAL_DATA.

Таблица 1.758 – Описание полей регистра CAL_DATA

Поле	Название	Программный доступ	Состояние после сброса	Описание
31:14	—	ЧТ0	—	Зарезервировано
13:7	CAL_DD	ЧТ/ЗП	0x0	Значение калибровочного коэффициента при работе АЦП в дифференциальном режиме;
6:0	CAL13_DU	ЧТ/ЗП	0x0	Значение калибровочного коэффициента при работе АЦП в униполярном режиме;

1.3.14.6.4.6.64 Регистр состояния питания и калибровки АЦП PWR_STS

В регистре отслеживается состояние питания и калибровки АЦП;

В таблице 1.759 представлено описание полей регистра PWR_STS.

Таблица 1.759 – Описание полей регистра PWR_STS

Поле	Название	Программный доступ	Состояние после сброса	Описание
31:3	—	ЧТ0	—	Зарезервировано
2	CAL_STS	ЧТ	0x0	Статус выполнения процедуры калибровки; 0 - процедура калибровки АЦП выполняется ; 1 - процедура калибровки АЦП закончена;
1	ADC_STS	ЧТ	0x0	Состояние АЦП; 0 - АЦП не готов к работе; 1 - АЦП полностью готов к работе;
0	LDO_STS	ЧТ	0x0	Состояние источника опорного напряжения (LDO); 0 - источник опорного напряжения не готов к работе; 1 - источник опорного напряжения готов к работе;

1.3.14.6.4.6.65 Регистр конфигурации АЦП ADC_CFG

В регистре задаются параметры включения и работы АЦП;

В таблице 1.760 представлено описание полей регистра ADC_CFG.

Таблица 1.760 – Описание полей регистра ADC_CFG

Поле	Название	Программный доступ	Состояние после сброса	Описание
31:27	—	ЧТ0	—	Зарезервировано
26	LDO_SELRNG	ЧТ/ЗП	0x0	Выбор выбор диапазона напряжений источника входного напряжения LDO; 0 - диапазон - 0,81 – 0,99В; 1 - диапазон - 0,99 – 1,21В;
25	LDO_SELREF	ЧТ/ЗП	0x0	Выбор источника входного напряжения LDO; 0 - источником является цифровое питание АЦП; 1 - источником является внешний источник опорного напряжения;
24:13	LDO_TUP	ЧТ/ЗП	0x0	Время выхода источника опорного напряжения (LDO) на рабочий режим; fffh - один такт тактового сигнала АЦП; ffeh - два такта тактового сигнала АЦП; … 001h - 2_{12}-1 такта тактового сигнала АЦП;
12:0	ADC_REARM	ЧТ/ЗП	0x0	значение времени максимального простоя АЦП без выполнения процесса оцифровки в тактах АЦП; 1fffh - зарезервировано; 1ffeh - два такта тактового сигнала АЦП; … 0001h - 2_{13}-1 такта тактового сигнала АЦП;

В таблицах 1.761, 1.762, 1.763 приведены основные характеристики аналогоцифрового преобразователя.

Таблица 1.761 – Характеристики АЦП

Наименование параметра, единица измерения	Буквенное обозначение	Режим измерения	Мин	Типовое	Макс
Напряжение питания AVDD,В	V_{DDA}	-	1.8	-	3.6
Напряжение смещения положительное, В	V_{REF+}	$V_{DDA} \geq 2$ В	2	-	V_{DDA}
		$V_{DDA} < 2$ В	V_{DDA}		
Напряжение смещения отрицательное, В	V_{REF-}	-	V_{SSA}		
Частота синхросигналов, МГц	fADC		0.14	-	80
Частота выборки FAST каналов, Мв/с	fs	Разрядность, 12 бит	-	-	5.33
		Разрядность, 10 бит	-	-	6.15
		Разрядность,8 бит	-	-	7.27

Наименование параметра, единица измерения	Буквенное обозначение	Режим измерения	Мин	Типовое	Макс
		Разрядность,6 бит	-	-	8.88
Частота выборки SLOW каналов, Мв/с	fs	Разрядность,12 бит	-	-	4.21
		Разрядность,10 бит	-	-	4.71
		Разрядность,8 бит	-	-	5.33
		Разрядность,6 бит	-	-	6.15
Внешняя частота запуска преобразования, МГц	f_{TRIG}	f_{ADC} = 80 МГц Разрядность,12 бит	-	-	5.33
			-	-	15 тактов f_{ADC}
Диапазон входного сигнала, В	V_{AIN}	-	0	-	V_{REF+}
Внешнее входное сопротивление, кОм	R_{AIN}	-	-	-	50
Внутренняя емкость выборки и хранения, пФ	C_{ADC}	-	-	5	-
Время включения, циклов преобразования	t_{STAB}	-	1		
Время калибровки, мкс	t_{CAL}	f_{ADC} = 80 МГц	1.45		
			116 тактов f_{ADC}		
Задержка преобразования	t_{LATR}		1.5	2	2.5 тактов f_{ADC}
Время выборки, мкс	ts	f_{ADC} = 80 МГц	0.03125	-	8.00625
			2.5	-	640.5 тактов f_{ADC}
Время выхода на режим регулятора напряжения, мкс	$t_{ADCVREG_STUP}$	-	-	-	20
Полное время преобразования, включая время выборки, мкс	t_{CONV}	f_{ADC} = 80 МГц Разрядность 12 бит	0.1875	-	8.1625
			t_S + 12.5 циклов для последовательного приближения = 15 до 653 тактов f_{ADC}		
Ток потребления АЦП от источника питания VDDA, мкА	I_{DDA}	fs = 5 Мвыб/с	-	730	830
		fs = 1 Мвыб/с	-	160	220
		fs = 10 квыб/с	-	16	50
Ток потребления АЦП от источника питания VREF+ в однополярном режиме, мкА	I_{DDV_S}	fs = 5 Мвыб/с	-	130	160
		fs = 1 Мвыб/с	-	30	40
		fs = 10 квыб/с	-	0.6	2
Ток потребления АЦП от	I_{DDV_D}	fs = 5 Мвыб/с	-	260	310
		fs = 1 Мвыб/с	-	60	70

Наименование параметра, единица измерения	Буквенное обозначение	Режим измерения	Мин	Типовое	Макс
источника питания VREF+ в дифференциальном режиме, мкА		fs = 10 квыб/с	-	1.3	3

Таблица 1.762 – Предельные характеристики АЦП

Разрядность, бит	Время выборки в тактах @80 МГц	Время выборки, нс @80 МГц	Максимальное значение R_{AIN}, Ом	
			Каналы быстрые	Каналы медленные
12				
	6.5	81.25	330	100
	12.5	156.25	680	470
	24.5	306.25	1500	1200
	47.5	593.75	2200	1800
	92.5	1156.25	4700	3900
	247.5	3093.75	12000	10000
	640.5	8006.75	39000	33000
10				
	6.5	81.25	390	180
	12.5	156.25	820	560
	24.5	306.25	1500	1200
	47.5	593.75	2200	1800
	92.5	1156.25	5600	4700
	247.5	3093.75	12000	10000
	640.5	8006.75	47000	39000
8				
	6.5	81.25	470	270
	12.5	156.25	1000	680
	24.5	306.25	1800	1500
	47.5	593.75	2700	2200
	92.5	1156.25	6800	5600
	247.5	3093.75	15000	12000
	640.5	8006.75	50000	50000
6				
	6.5	81.25	560	330
	12.5	156.25	1200	1000
	24.5	306.25	2700	2200
	47.5	593.75	3900	3300
	92.5	1156.25	8200	6800
	247.5	3093.75	18000	15000
	640.5	8006.75	50000	50000

1) Быстрые каналы: PD3, PD4, PD5, DAC0, DAC1.

2) Медленные каналы: все остальные входы АЦП, кроме быстрых

Таблица 1.763 – Точностные характеристики АЦП

Наименование параметра, единица измерения	**Буквенное обозначение**	**Режим измерения**			**Мин**	**Типовое**	**Макс**
Общая нескорректированная ошибка, EMP	ET	Частота синхросигнала ADC ≤ 80 МГц, Частота выборки ≤ 5.33 Мв/с, 1.65 V ≤ VDDA = VREF+ ≤ 3.6 В	Однополярный сигнал	Быстрый канал (максимальная скорость)	-	5.5	7.5
				Медленный канал (максимальная скорость)	-	4.5	6.5
			Дифференциальный сингал	Быстрый канал (максимальная скорость)	-	4.5	7.5
				Медленный канал (максимальная скорость)	-	4.5	5.5
Ошибка смещения, EMP	EO		Однополярный сигнал	Быстрый канал (максимальная скорость)	-	2	5
				Медленный канал (максимальная скорость)	-	2.5	5
			Дифференциальный сингал	Быстрый канал (максимальная скорость)	-	2	3.5
				Медленный канал (максимальная скорость)	-	2.5	3
Ошибка усиления, EMP	EG		Однополярный сигнал	Быстрый канал (максимальная скорость)	-	4.5	7
				Медленный канал (максимальная скорость)	-	3.5	6
				Fast channel (max speed)	-	3.5	4

Наименование параметра, единица измерения	Буквенное обозначение	Режим измерения			Мин	Типовое	Макс
			Дифференциальный сингал	Медленный канал (максимальная скорость)	-	3.5	5
Ошибка дифференциальной нелинейности, EMP	ED		Однополярный сигнал	Быстрый канал (максимальная скорость)	-	1.2	1.5
				Медленный канал (максимальная скорость)	-	1.2	1.5
			Дифференциальный сингал	Быстрый канал (максимальная скорость)	-	1	1.2
				Медленный канал (максимальная скорость)	-	1	1.2
Ошибка интегральной нелинейности, EMP	EL		Однополярный сигнал	Быстрый канал (максимальная скорость)	-	3	3.5
				Медленный канал (максимальная скорость)	-	2.5	3.5
			Дифференциальный сингал	Быстрый канал (максимальная скорость)	-	2	2.5
				Медленный канал (максимальная скорость)	-	2	2.5
Эффективное количество разрядов, бит	ENOB		Однополярный сигнал	Быстрый канал (максимальная скорость)	10	10.4	-
				Медленный канал (максимальная скорость)	10	10.4	-
			Дифференциальный сингал	Быстрый канал (максимальная скорость)	10.6	10.7	-

Наименование параметра, единица измерения	Буквенное обозначение	Режим измерения			Мин	Типовое	Макс
				Медленный канал (максимальная скорость)	10.6	10.7	-
Отношение сигнал/шум и коэффициент искажения, дБ	SINAD		Однополярный сигнал	Быстрый канал (максимальная скорость)	62	64	-
				Медленный канал (максимальная скорость)	62	64	-
			Дифференциальный сингал	Быстрый канал (максимальная скорость)	65	66	-
				Медленный канал (максимальная скорость)	65	66	-
Отношение сигнал/шум, дБ	SNR		Однополярный сигнал	Быстрый канал (максимальная скорость)	63	65	-
				Медленный канал (максимальная скорость)	63	65	-
			Дифференциальный сингал	Быстрый канал (максимальная скорость)	66	67	-
				Медленный канал (максимальная скорость)	66	67	-
Коэффициент нелинейных искажений, дБ	THD		Однополярный сигнал	Быстрый канал (максимальная скорость)	-	-69	-67
				Медленный канал (максимальная скорость)	-	-71	-67
			Дифференциальный сингал	Быстрый канал (максимальная скорость)	-	-72	-71

Наименование параметра, единица измерения	Буквенное обозначение	Режим измерения			Мин	Типовое	Макс
				Медленный канал (максимальная скорость)	-	-72	-71

1.3.14.7 Блок цифро-аналоговых преобразователей (DAC)

Блок цифроаналоговых преобразователей (DAC) формирует на основе передаваемой контроллером информации аналоговые сигналы для выдачи на внешние выводы микросхемы. Кроме этого сформированное аналоговое напряжение может использоваться в качестве опорного при аналогоцифровом преобразовании блоком ADC и в качестве референсного при работе блока аналоговых компараторов (ACMP). Блок содержит два цифроаналоговых преобразователя, которые могут работать как независимо друг от друга, так и синхронно. Управление работой блока осуществляется по шине AMBA AHB спецификации 5.0 со следующими особенностями:

– циклы чтения и записи выполняются в блоке за один такт

– аппаратура интерфейса AMBA_AHB блока не поддерживает пакетную передачу данных

Цифроаналоговые преобразователи тактируются отдельным тактовым сигналом (DAC_CLK) асинхронным по отношению к тактовому сигналу шины AMBA AHB. Каждый цифроаналоговый преобразователь настраиваеися индивидуально. Синхронная работа подразумевает, что данные для преобразования в аналоговую форму загружаются в оба цифроаналоговых преобразователя по одному фронту тактового сигнала DAC_CLK.

1.3.14.7.1 Основные параметры цифроаналоговых преобразователей

В таблице 1.764 приведены основные параметры цифроаналогового преобразователя.

Таблица 1.764 – Основные параметры цифроаналогового преобразователя

Параметр	Обозначение	Мин.	Тип.	Макс.	Единицы
Напряжение аналогово питания	AVDD_DAC	1,8	3,3	3,6	В
Напряжение цифрового питания	VDD_DAC	0,81	0,9	1.21	В
Опорное напряжение	VREFP_DAC	1,8	3,3	3,6	В
Выходное напряжение ЦАП	VOUT	0		VREF_DAC	В
Выходной ток	IOUT			2,0	мА

Ток потребления по цепям AVDD_DAC+VREFP_DAC при BUF_EN = 1	IAVDD		1300		мкА
Ток потребления по цепям AVDD_DAC+VREFP_DAC при BUF_EN = 0	IAVDD		550		мкА
Ток потребления по цепи VDD_DAC	IVDD		2		мкА
Сопротивление нагрузки	RO	5			кОм
Емкость нагрузки	CO			50	пФ
Эквивалентное сопротивление на землю в выключенном состоянии DAC_EN = 0	RG	1			МОм
Разрешение ЦАП			12		бит

В таблице 1.765 приведены временные параметры цифроаналогового преобразователя. Временные параметры приведены при следующих условиях: напряжение питания AVDD_DAC = 3,3В, VDD = 0,9В, температура T = 25º, сопротивление нагрузки RO = 5 кОм, емкость нагрузки CO = 50 пФ.

Таблица 1.765 – Временные параметры цифроаналогового преобразователя

Параметр	Обозначение	Мин.	Тип.	Макс.	Единицы
Время установки сигнала на выходе после включения ЦАП	t_{su}			1	мкс
Время выключения ЦАП	t_{dis}			0,4	мкс
Период тактового сигнала ЦАП	T_{clk}	0,05	1		мкс
Задержка ЦАП	t_d		200		нс

В таблице 1.766 приведены основные характеристики цифроаналогового преобразователя.

Таблица 1.766 – Основные характеристики цифроаналогового преобразователя

Параметр	Обозначение	Мин.	Тип.	Макс.	Единицы
Дифференциальная нелинейность	DNL	-1,2		+1,2	LSB
Интегральная нелинейность	INL	-1,6		+1,6	LSB
ENOB	ENOB		10		бит
Смещение характеристики ЦАп	Offset	-10		+10	LSB
Коэффициент усиления	Gain		1		
Соотношение сигнал/шум	SNR		62		dB
Общее гармоническое искажение	THD		-65		dB

Формирование сигнала на выходе ЦАП в зависимости от подаваемого кода представлено на на рисунке 1.167.

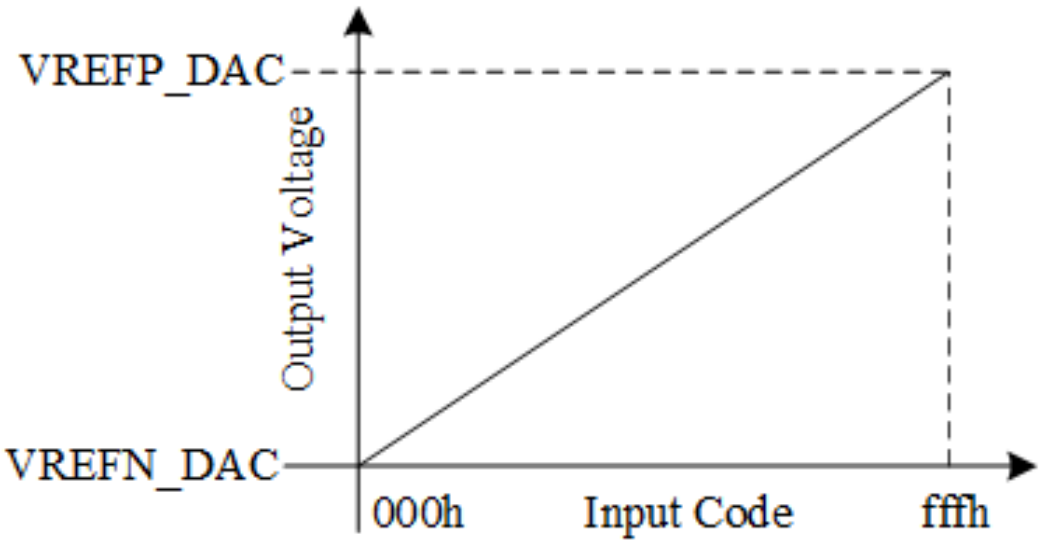


Рисунок 1.167 – Формирование сигнала на выходе ЦАП в зависимости от подаваемого кода

1.3.14.7.2 Структурная схема блока цифроаналоговых преобразователей

Структурная схема блока цифроаналоговых преобразователей представлена на рисунке 1.168.

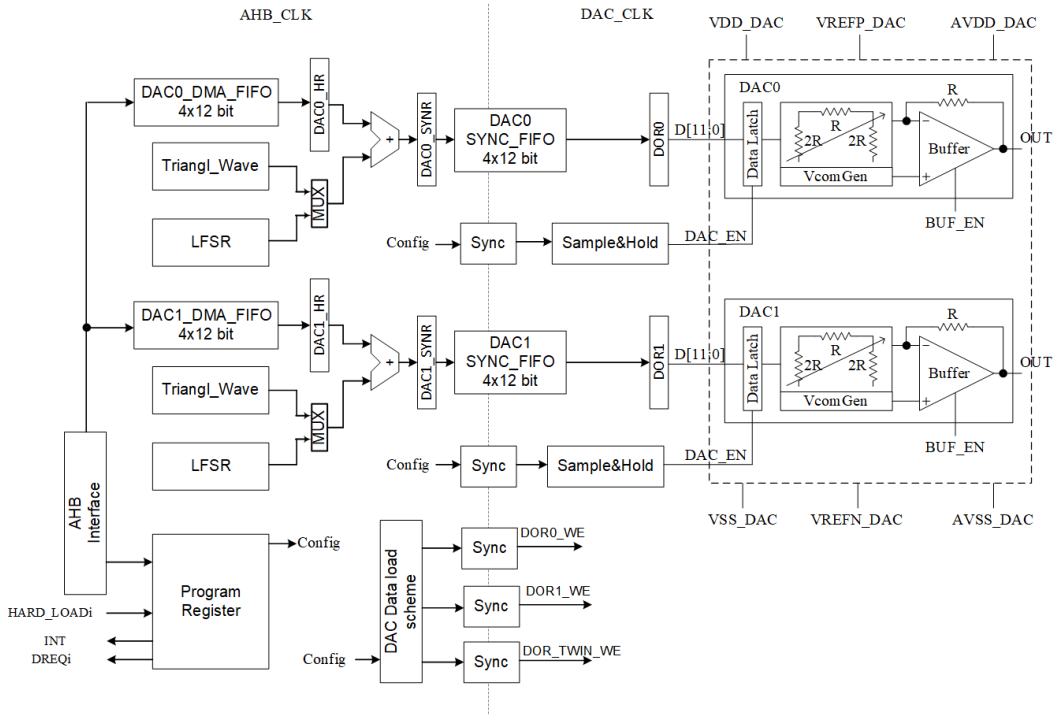


Рисунок 1.168 – Структурная схема блока цифроаналоговых преобразователей

Блок цифроаналоговых преобразователей содержит следующие узлы и блоки:

– два цифроаналоговых преобразователя DAC0 и DAC1, каждый из которых включает в себя управляемую защелку, линейку сопротивлений R2R и отключаемый выходной буфер.

– входные регистры данных DOR0 и DOR1. При включенных цифроаналоговых преобразователях именно запись в регистры DORi начинает собственно цифроаналоговое преобразование.

– два блока Sample&Hold, реализирующих режим работы Sample&Hold для каждого цифроаналогового преобразователя

Работа всех вышеперечисленных блоков тактируется тактовым сигналом работы цифроаналоговых преобразователей (DAC_CLK).

– синхронизирующие буфера данных DACi SYNC FIFO размером в четыре слова, предназначенные для синхронизации передачи данных с частоты шины AHB (AHB_CLK) на частоту работы ЦАП (DAC_CLK)

– программно доступные по чтению регистры DAC0_SYNCR и DAC1_SYNCR, в которых хранится полностью сформированные данные, которые выданы на вход цифроаналоговых преобразователей

– программно доступные по чтению регистры DAC0_HR и DAC1_HR, хранящие данные для цифроаналоговых преобразователей непосредственно считанные из памяти контроллера без накладки шумового или треугольного воздействия

– два блока LFSR, формирующих псевдослучайный сигнал для наложения на исходные данные

– два блока Triangle_Wave, формирующих сигнал треугольной формы для наложения на исходные данные

– два буфера данных DAC0_DMA_FIFO и DAC1_DMA_FIFO, размерностью в четыре слова данных каждый, позволяющие заранее подгрузить из памяти подаваемые на вход цифроаналоговых преобразователей данные

– блок конфигурационных регистров Program Register, задающих параметры и управляющих работой блока

– схемы формирования сигнала разрешения загрузки данных в цифроаналоговые преобразователи DAC Data load

Работа всех вышеперечисленных блоков тактируется тактовым сигналом шины AMBA AHB (AHB_CLK).

– схем синхронизации Sync, необходимых при переходе с частоты системной шины (AHB_CLK) на частоту работы преобразователей (DAC_CLK)

1.3.14.7.3 Управление питанием цифроаналоговых преобразователей

Перед включением цифроаналоговых преобразователей необходимо убедиться, что на них подано питание и установилось напряжение питания аналоговой (AVDD_DAC) и цифровой (VDD_DAC) частей преобразователей. Кроме этого на вход опорного напряжения (VREF_DAC) должен быть подан соответствующий уровень напряжения.

Работой цифроаналоговыми преобразователями управляют два входа DAC_EN и BUF_EN. Управляющий вход DAC_EN разрещает работу непосредственно преобразователя. При нулевом значении этого входа преобразователь не работает, выход блока находится в высокоимпедансном состоянии, потребление преобразователя минимально. Единичное состояние на этом входе разрешает нормальную работу преобразователя.

Вход BUF_EN управляет работой выходного буфера преобразователя. При нулевом значении этого входа буфер выключен и на выход блока подается нарряжение с линейки R2R преобразователя. При единичном значении этого входа буфер работает как повторитель выхода линейки R2R. Включение буфера приводит к увеличению потребления по цепям аналогового питания.

Временная диаграмма включения и выключения преобразователя представлена на рисунке 1.169.

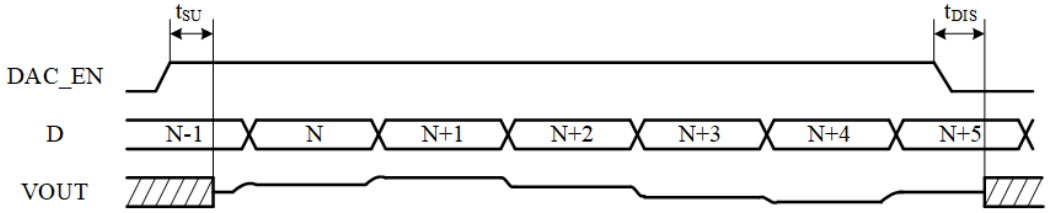


Рисунок 1.169 – Временная диаграмма включения и выключения цифроаналогового преобразователя

1.3.14.7.4 Организация работы блока цифроаналоговых преобразователей

1.3.14.7.4.1 Тракт данных цифроаналоговых преобразователей

Тракты данных обоих цифроаналоговых преобразователей идентичны, поэтому в дальнейшем будет описан тракт и запуск процедуры цифроаналогового преобразования одного блока DAC.

Запуск процедуры оцифровки данных может быть задан как программно, так и аппаратно, в зависимости от поля DACi_CTRL регистра DAC_CFG. Программный запуск осуществляется записью логической единицы в регистр DACi_STR. После выполнения загрузки данных непосредственно в DACi данный регистр аппаратно сбрасывается, позволяя выполнить следующую операцию загрузки данных в DAC. Аппаратный запуск осуществляется подачей изменением состояния входа HARD_LOADi блока на инверсное. Время, между последовательными изменениями состояния входа HARD_LOADi не должно быть менее двух тактов тактового сигнала шины AMBA AHB (AHB_CLK). При работе двух цифроаналоговых преобразователей в синхронном режиме (поле DBL_EN регистра DAC_DBL установлено в единицу) запуск процедуры цифроаналогового преобразования обоих DAC осуществляется аппаратно или программно одномоментно по запуску одного из DAC. Выбор запуск процедуры цифроаналогового преобразования какого DAC приводит к запуску процедуры в синхронном режиме осуществляется полем DBL_CTRL регистра DAC_DBL.

Перед запуском процедуры цифроаналогового преобразования данные, которые необходимо преобразовать в аналоговую форму, должны быть по шине AMBA AHB загружены в регистр следующих данных DACi_HR. В зависимости от поля DACi_FIFO_EN регистра конфигурации DAC_CFG данные могут загружаться либо непосредственно в регистр DACi_HR, либо через буфер данных типа FIFO. Если буфер не используется, то данные в регистре DACi_HR можно изменить, не подавая сигнала запуска процедуры цифроаналогового преобразования. В случае использования буфера данные из буфера записываются в регистр DACi_HR только по сигналу запуска цифроаналогового преобразования.

Загрузка данных в регистры DACi_HR или в буфера данных типа FIFO осуществляется записью данных по шине AMBA AHB в один мз следующих программно доступных регистров DACi_D8, DACi_D12, DACDBL_D8 или DACDBL_D12. При записи данных в регистры DACi_D12 данные разрядов [11:0] шины данных непосредственно записываются в тракт данных соответствующего DAC. При записи данных в регистры DACi_D8 данные разрядов [7:0] шины данных записываются в старшие разряды [11:4] тракта данных соответствующего DAC. При этом младшие разряды тракта [3:0] обнуляются. Это позволяет пользователю хранить данные в памяти контроллера более компактно.

Запись в регистры DACDBL_D8 и DACDBL_D12 аналогична записи в регистры DACi_D8 и DACi_D12, но данные с соответствующих разрядов шины одновременно записываются как в тракт одного, так и в тракт второго DAC. Необходимо отметить, что запись

данных в тракт DAC осуществляется при записи в данные регистры одинаково, независимо от того в совместном или независимом режиме работают оба DAC.

Формат данных, записываемых в тракт каждого DAC может быть как знаковым, так и беззнаковым и задается полями DACi_DTYP регистра DAC_CFG для каждого DAC индивидуально.

После аппаратного или программного запуска цифроаналогового преобразования данные, хранимые в регистре DACi_HR переписываются в регистр DACi_SYNR и одновременно формируется управляющий сигнал на схему синхронизации об изменении данных, загружаемых в DAC. При перезаписи данных из регистра DACi_HR в регистр DACi_SYNR на них могут быть наложены псевдослучайные искажения, формируемые блоком LFSR, или треугольные искаженгия, формируемые блоком Triangle_Wave. Необходимо отметить, что искажения накладываются уже на 12 битные данные, незаисимо от того, какие данные записывались в регистр DACi_HR. Данные из регистра DACi_HR переписываются в регистр DACi_SYNR по сигналу запуска независимо от того загружались ли в регистр DACi_HR из буфера или с шины AMBA AHB новые данные после предыдущего сигнала запуска цифроаналогового преобразования. Если данные в регистре DACi_HR не изменялись, то формируется запрос на прерывание по повторной загрузке в DAC одних и тех же данных (DAC0_UDF).

Схема синхронизации формирует сигнал записи данных в регистр DORi и по положительному фронту тактового сигнала DAC_CLK в него будут записаны данные регистра DACi_SYNR. По следующему отрицательному фронту тактового сигнала DAC_CLK данные защелкнутся во входных защелках DAC и начнется непосредственно цифроаналоговое преобразование.

Временная диаграмма старта цифроаналогового преобразования при аппаратном запуске представлена на рисунке 1.170. Временная диаграмма показывает загрузку данных в DAC, при условии, что тракт сконфигурирован на работу с буфером данных и данные в буфер были предварительно загружены.

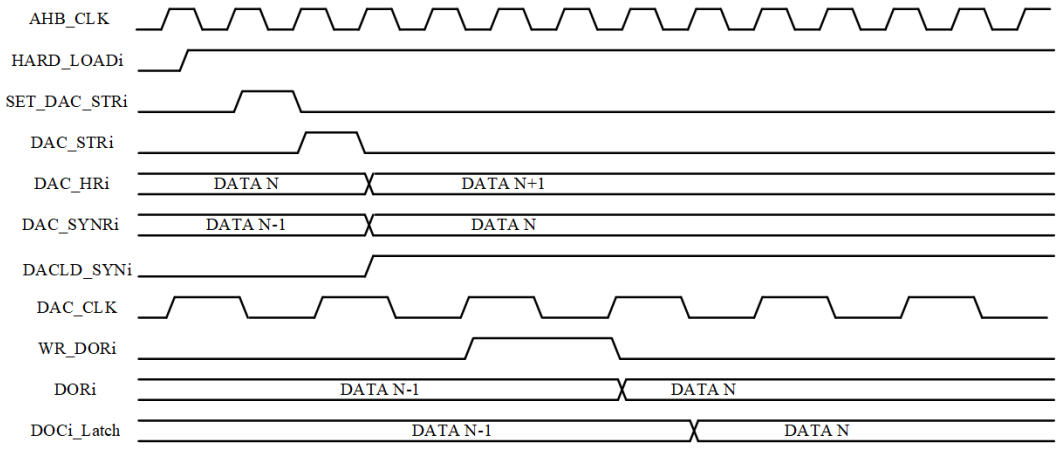


Рисунок 1.170 – Временная диаграмма старта цмфроаналогового преобразования при аппаратном запуске

При программном запуске цифроаналогового преобразования диаграмма аналогична, только регистр DAC_STRi устанавливается программно.

1.3.14.7.4.2 Блок формирования псевдослучайных последовательности LFSR

Блок формирования псевдослучайных последовательности реализован на основе 12-разрядного сдвигового регистра с обратными связями. Структурная схема формирования представлена на рисунке 1.171. В тракте данных у каждого из DAC есть свой генератор псевдослучайных последовательностей, конфигурируемый индивидуально.

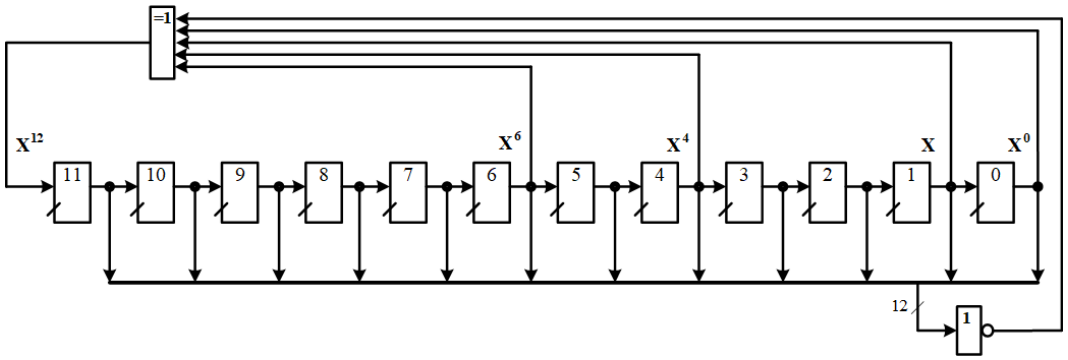


Рисунок 1.171 – Структурная схема формирования псевдослучайной последовательности

Включение формирования псевдослучайной последовательности задается установкой поля DACi_WM регистра DAC_CFG в состояние 01. В выключенном состоянии сдвиговый регистр LFSR устанавливается в начальное состояние AAAh. После включения при каждом запуске цифроаналогового преобразования значение регистра LFSR складывается с содержимым регистра DACi_HR и записывается в регистр DACi_SYNR. Одновременно состояние регистра LFSR изменяется в соответствии с приведенной структурной схемой. Поле DACi_WBW

регистра конфигурации DAC_CFG позволяет ограничить размах генерируемого псевдослучайного числа, обнуляя, в соответствии с установкой, старшие разряды регистра LFSR, направляемые на суммирование с содержимым регистра DACi_HR.

1.3.14.7.4.3 Блок формирования треугольного сигнала Triangle_Wave

Блок Triangle_Wave формирует сигнал треугольной формы, накладываемый на считываемые из памяти данные, предназначенные для преобразования в аналоговую форму.В тракте данных у каждого из DAC есть свой генератор Triangle_Wave, конфигурируемый индивидуально. Включение формирования треугольного сигнала задается установкой поля DACi_WM регистра DAC_CFG в состояние 10. Генератор реализован на основе регистра и инкрементора. В выключенном состоянии регистр данных блока сбрасывается в состояние 000h. После включения при каждом запуске цифроаналогового преобразования значение регистра данных блока складывается с содержимым регистра DACi_HR и записывается в регистр DACi_SYNR. Одновременно происходит инкремент состояния регистра данных блока. После достижения регистром данных блока максимально возможного значения при каждом следующем запуске цифроаналогового преобразования регистр данных будет декрементироваться до достижения значения 000h и т.д. Размах пилообразного сигнала задается полем DACi_WBW регистра конфигурации DAC_CFG. Например состояние поля 0010b означает, что выходные данные блока, направляемые на суммирование с регистром DACi_HR, могут изменяться в диапазоне от 000h до 007h.

1.3.14.7.4.4 Режим LWOD (Load WithOut Data)

Аппаратные блоки LFSR и Triangle_Wave позволяют реализовать режим LWOD (Load WithOut Data). Включение данного режима осуществляется установкой бита DACi_LWOD в регистре DAC_CFG при обязательной активации работы отдого из аппаратных блоков: или LFSR или Triangle_Wave для каждого ЦАП соответственно. Кроме этого необходимо записать в регистр DACi_HR начальные данные. Эти начальные данные будут поданы на вход DAC первыми. В дальнейшем при каждой загрузке данных в DAC происходит сложение этих данных с данными, формируемыми аппаратными блоками LFSR или Triangle_Wave в соответствии с их настройкой. Повторной загрузки данных в регистр DACi_HR при этом не требуется. Формирование аппарайного запроса на данные и запроса на прерывание по загрузке данных при работе в данном режиме не производится, за исключением запроса на начальные данные.

Выход из данного режима работы производится сбросом бита DACi_LWOD в регистре DAC_CFG или выключением блокав LFSR или Triangle_Wave.

1.3.14.7.4.5 Формирование прерываний блока и аппаратных запросов на подкачку данных

Запросы на прерывания фиксируются в регистре DAC_IRR независимо от того установлены или сброшены маски данных прерываний.

Формирование прерываний и аппаратных запросов на подкачку данных производится для каждого DAC индивидуально и не зависит от того работаюи ли блоки DAC независимо или совместно. для каждого блока DAC формируются следующие запросы на прерывание: прерывание по потере данных DACi_OVF, прерывание по повторному использованию данных DACi_UDF и прерывание по запросу данных DACi_REQ.

Прерывание по потере данных DACi_OVF формируется в следующих случаях:

– при работе тракта данных с отключенным буфером данных - в случае между прошлой и текущей записью данных в регистр DACi_HR не был сформирован сигнал запуска цифроаналогового преобразования.

– при работе тракта данных с использованием буфера данных - в случае попытки записи в полный буфер данных.

Данное прерывание сбрасывается программно.

Прерывание по повторному использованию данных DACi_UDF формируется в случае между двумя последовательными запусками цифроаналогового преобразования не было записи (программной или из буфера данных, если он используется) в регистр DACi_HR.

Данное прерывание сбрасывается программно.

Прерывание по запросу данных DACi_REQ формируется в следующих случаях:

– при работе тракта данных с отключенным буфером данных - после сигнала запуска цифроаналогового преобразования. После запуска цифроаналогового преобразования данные из регистра DACi_HR переписываются в регистр DACi_SYNR и регистр DACi_HR готов принять новые данные. При записи новых данных в регистр DACi_HR прерывание в этом случае сбрасывается аппаратно.

– при работе тракта данных с использованием буфера данных - при пустом буфере данных, т.е. предполагается, что при получении данного прерывания пользователь запишет во входное FIFO не менее четырех данных. Запрос аппаратно сбрасывается при записи первых

данных во входной буфер. Повторная установка этого запроса на прерывание производится после того как входной буфер опустеет после записи в него не менее четырех слов данных.

Аппаратный запрос на подгрузку данных, формируемый на выводах DREQi блока, позволяет работать с блоком DAC устройствам типа контроллера ПДП. Запрос на выходах DREQi формируется фронтом. Это означает, что изменение состояния выхода является тем индикатором, который говорит о необходимости подгрузки данных. данный аппаратный запрос формируется аналогично прерыванию по запросу данных DACi_REQ.

Все прерывания, а также формирование аппаратного запроса на подгрузку данных могут быть замаскированы установкой соответствуюшей маски в регистре DAC_IMR.

В режиме LWOD прерывания по запрсу данных и аппаратные запросы на подгрузку данных, за исключением запроса на начальные данные, не формируются.

Выходной запрос на прерывание от блока представляет собой свертку всех запросов регистра DAC_IRR по "или" с учетом масок.

1.3.14.7.4.6 Режим удержания данных Sample&Hold

Режим Sample&Hold применяется для кратного снижения потребления блока при условии, что блок DAC должен относительно длительное время сохранять на выходе постоянный уровень. Для сохранения постоянного уровня обычно схемотехнически применяют конденсатор большой емкости, периодически подзаряжаемый. Примерная схема включения блока в режиме Sample&Hold представлена на рисунке 1.172.

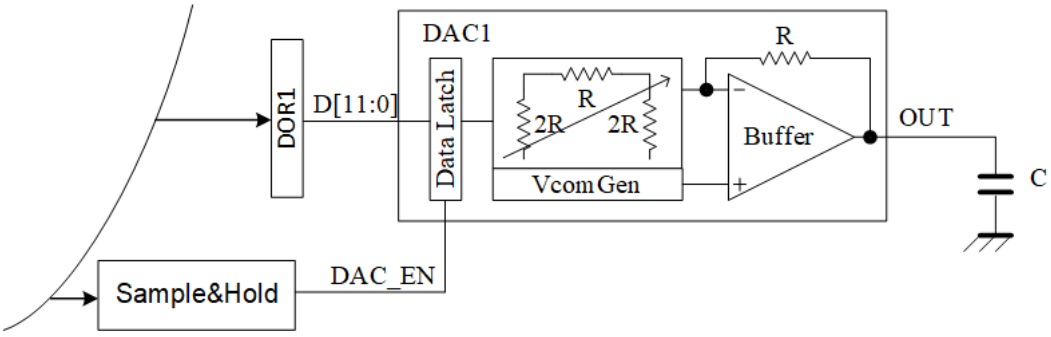


Рисунок 1.172 – Примерная схема включения блока в режиме Sample&Hold

При таком включении для экономии электроэнергии имеется возможность снимать сигнал DAC_EN и отключать тем самым потребление собственно блока DAC. При выключенном блоке DAC из-за токов утечки конденсатор C медленно разряжается. Схема Sample&Hold контроллера DAC периодически подает активный уровень на вход DAC_EN, позволяя блоку DAC подзарядить конденсатор C. Необходимое время установки напряжения на выходе DAC, время

удержания и время регенерации задаются пользователем в регистрах DACiSH_CFG для каждого цифроаналогового преобразователя индивидуально. Временная диаграмма работы в режиме Sample&Hold представлена на рисунке 1.173.

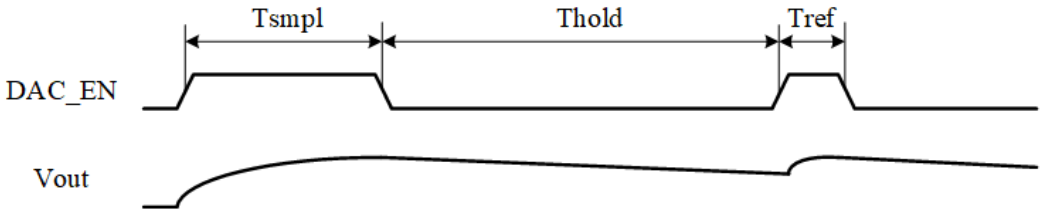


Рисунок 1.173 – Временная диаграмма работы в режиме Sample&Hold

После сигнала запуска цифроаналогового преобразования, при работе блока в данном режиме, DAC в течении времени T_{Sample} выдает на выход напряжение, соответствующее загруженным данным. После этого сигнал DAC_EN снимается на время T_{Hold}, а затем устанавливается на время T_{Ref}. Последовательность включения м выключения DAC повторяется до тех пор, пока не придет новый сигнал запуска цифроаналогового преобразования. После этого начнется новый цикл T_{Sample} - T_{Hold} - T_{Ref}.

1.3.14.7.4.7 Программно доступные регистры блок DAC

В таблице 1.767 представлен список программно доступных регистров блок DAC.

Все программно доступные регистры блока цифроаналоговых преобразователей расположены начиная с базового адреса устройства – DAC_Base.

Таблица 1.767 – Список программно доступных регистров блок DAC

Название	Адрес	Описание
DAC_IRR	0x000	Регистр запросов на прерывание
DAC_IMR	0x004	Регистр масок запросов на прерывание
DAC_IMR_set	0x008	Регистр установки масок запросов на прерывание
DAC_IMR_rst	0x00C	Регистр сброса масок запросов на прерывание
DAC0_STR	0x010	Регистр программного запуска DAC0
DAC1_STR	0x014	Регистр программного запуска DAC1
RESERVED	(0x018 — 0x01C)	Зарезервировано
DAC0_D8	0x020	Регистр записи восьмибитных данных в DAC0
DAC1_D8	0x024	Регистр записи восьмибитных данных в DAC1
DACDBL_D8	0x028	Регистр записи восьмибитных данных при синхронной работе двух цифроаналоговых преобразователей
RESERVED	0x02C	Зарезервировано
DAC0_D12	0x030	Регистр записи двенадцатибитных данных в DAC0

Название	Адрес	Описание
DAC1_D12	0x034	Регистр записи двенадцатибитных данных в DAC1
DACDBL_D12	0x038	Регистр записи двенадцатибитных данных при синхронной работе двух цифроаналоговых преобразователей
RESERVED	0x03C	Зарезервировано
DAC0_HR	0x040	Регистр следующих данных DAC0
DAC0_SYNR	0x044	Регистр сформированных данных DAC0
DAC1_HR	0x048	Регистр следующих данных DAC1
DAC1_SYNR	0x04C	Регистр сформированных данных DAC1
RESERVED	(0x050 — 0x05C)	Зарезервировано
DAC_CFG	0x060	Регистр конфигурации блока цифроаналоговых прелбразователей
DAC0SH_CFG	0x064	Регистр конфигурации режима Sample&Hold DAC0
DAC1SH_CFG	0x068	Регистр конфигурации режима Sample&Hold DAC1
DAC_DBL	0x06C	Регистр управления синхронной работой DAC0 и DAC1
DAC_EN	0x070	Регистр управления и состояния блока DAC

1.3.14.7.4.7.1 Регистр запросов на прерывание DAC_IRR

Регистр зхранит запросы на прерывания от блока цифроаналоговых преобразователей. Запись логической единицы обнуляет соответствующий бит регистра

В таблице 1.768 представлено описание полей регистра DAC_IRR.

Таблица 1.768 – Описание полей регистра DAC_IRR

Поле	Название	Программный доступ	Состояние после сброса	Описание
31:7	—	ЧТ0	—	Зарезервировано
6	DAC1_REQ	ЧТ/ЗП	0x0	Запрос на загрузку данных в DAC1. Запрос аппаратно сбрасывается при записи данных в входной регистр блока;
5	DAC1_UDF	ЧТ/ЗП	0x0	Прерывание по повторной загрузке в DAC1 одних и тех же данных;
4	DAC1_OVF	ЧТ/ЗП	0x0	Прерывание по потере данных от DAC1;
3	Reserved	-	0x0	зарезервировано;
2	DAC0_REQ	ЧТ/ЗП	0x0	Запрос на загрузку данных в DAC0. Запрос аппаратно сбрасывается при записи данных в входной регистр блока;
1	DAC0_UDF	ЧТ/ЗП	0x0	Прерывание по повторной загрузке в DAC0 одних и тех же данных;
0	DAC0_OVF	ЧТ/ЗП	0x0	Прерывание по потере данных от DAC0;

1.3.14.7.4.7.2 Регистр масок запросов на прерывание DAC_IMR

Регистр зхранит маски запросов на прерывания от блока цифроаналоговых преобразователей;

В таблице 1.769 представлено описание полей регистра DAC_IMR.

Таблица 1.769 – Описание полей регистра DAC_IMR

Поле	Название	Программный доступ	Состояние после сброса	Описание
31:8	—	ЧТ0	—	Зарезервировано
7	DAC1_MHREQ	ЧТ/ЗП	0x0	Маска аппаратного выхода запроса на загрузку данных к DAC1: 0 - маска установлена; 1 - маска сброшена;
6	DAC1_MREQ	ЧТ/ЗП	0x0	Маска запроса на прерывание по загрузке данных в DAC1: 0 - маска установлена; 1 - маска сброшена;
5	DAC1_MUDF	ЧТ/ЗП	0x0	Маска запроса на прерывание по повторной загрузке в DAC1 одних и тех же данных: 0 - маска установлена; 1 - маска сброшена;
4	DAC1_MOVF	ЧТ/ЗП	0x0	Маска запроса на прерывание по потере данных от DAC1: 0 - маска установлена; 1 - маска сброшена;
3	DAC0_MHREQ	ЧТ/ЗП	0x0	Маска аппаратного выхода запроса на загрузку данных к DAC0: 0 - маска установлена; 1 - маска сброшена;
2	DAC0_MREQ	ЧТ/ЗП	0x0	Маска запроса на прерывание по загрузке данных в DAC0: 0 - маска установлена; 1 - маска сброшена;
1	DAC0_MUDF	ЧТ/ЗП	0x0	Маска запроса на прерывание по повторной загрузке в DAC0 одних и тех же данных: 0 - маска установлена; 1 - маска сброшена;
0	DAC0_MOVF	ЧТ/ЗП	0x0	Маска запроса на прерывание по потере данных от DAC0: 0 - маска установлена; 1 - маска сброшена;

1.3.14.7.4.7.3 Регистр установки масок запросов на прерывание DAC_IMR_set

Запись логической единицы в разряды регистра устанавливает соответсвующий бит регистра DAC_IMR в единицу;

В таблице 1.770 представлено описание полей регистра DAC_IMR_set.

Таблица 1.770 – Описание полей регистра DAC_IMR_set

Поле	Название	Программный доступ	Состояние после сброса	Описание
31:8	—	ЧТ0	—	Зарезервировано
7	DAC1_MHREQ_set	ЗП	0x0	Установка в 1 маски аппаратного выхода запроса на загрузку данных к DAC1;
6	DAC1_MREQ_set	ЗП	0x0	Установка в 1 маски запроса на прерывание по загрузке данных в DAC1;
5	DAC1_MUDF_set	ЗП	0x0	Установка в 1 маски запроса на прерывание по повторной загрузке в DAC1 одних и тех же данных;
4	DAC1_MOVF_set	ЗП	0x0	Установка в 1 маски запроса на прерывание по потере данных от DAC1;
3	DAC0_MHREQ_set	ЗП	0x0	Установка в 1 маски аппаратного выхода запроса на загрузку данных к DAC0;
2	DAC0_MREQ_set	ЗП	0x0	Установка в 1 маски запроса на прерывание по загрузке данных в DAC0;
1	DAC0_MUDF_set	ЗП	0x0	Установка в 1 маски запроса на прерывание по повторной загрузке в DAC0 одних и тех же данных;
0	DAC0_MOVF_set	ЗП	0x0	Установка в 1 маски запроса на прерывание по потере данных от DAC0;

1.3.14.7.4.7.4 Регистр сброса масок запросов на прерывание DAC_IMR_rst

Запись логической единицы в разряды регистра сбрасывает соответсвующий бит регистра DAC_IMR в ноль;

В таблице 1.771 представлено описание полей регистра DAC_IMR_rst.

Таблица 1.771 – Описание полей регистра DAC_IMR_rst

Поле	Название	Программный доступ	Состояние после сброса	Описание
31:8	—	ЧТ0	—	Зарезервировано
7	DAC1_MHREQ_rst	ЗП	0x0	Сброс в 0 маски аппаратного выхода запроса на загрузку данных к DAC1;
6	DAC1_MREQ_rst	ЗП	0x0	Сброс в 0 маски запроса на прерывание по загрузке данных в DAC1;
5	DAC1_MUDF_rst	ЗП	0x0	Сброс в 0 маски запроса на прерывание по повторной загрузке в DAC1 одних и тех же данных;
4	DAC1_MOVF_rst	ЗП	0x0	Сброс в 0 маски запроса на прерывание по потере данных от DAC1;
3	DAC0_MHREQ_rst	ЗП	0x0	Сброс в 0 маски аппаратного выхода запроса на загрузку данных к DAC0;
2	DAC0_MREQ_rst	ЗП	0x0	Сброс в 0 маски запроса на прерывание по загрузке данных в DAC0;

Поле	Название	Программный доступ	Состояние после сброса	Описание
1	DAC0_MUDF_rst	ЗП	0x0	Сброс в 0 маски запроса на прерывание по повторной загрузке в DAC0 одних и тех же данных;
0	DAC0_MOVF_rst	ЗП	0x0	Сброс в 0 маски запроса на прерывание по потере данных от DAC0;

1.3.14.7.4.7.5 Регистр программного запуска DAC0 DAC0_STR

Запись логической единицы в данный регистр запускает процедуру цифроаналогового преобразования DAC0. После старта данный регистр сбрасывается аппаратно;

В таблице 1.772 представлено описание полей регистра DAC0_STR.

Таблица 1.772 – Описание полей регистра DAC0_STR

Поле	Название	Программный доступ	Состояние после сброса	Описание
31:1	—	ЧТ0	—	Зарезервировано
0	DAC0_PSTR	ЧТ/ЗП	0x0	запуск цифроаналогового преобразования DAC0;

1.3.14.7.4.7.6 Регистр программного запуска DAC1 DAC1_STR

Запись логической единицы в данный регистр запускает процедуру цифроаналогового преобразования DAC1. После старта данный регистр сбрасывается аппаратно;

В таблице 1.773 представлено описание полей регистра DAC1_STR.

Таблица 1.773 – Описание полей регистра DAC1_STR

Поле	Название	Программный доступ	Состояние после сброса	Описание
31:1	—	ЧТ0	—	Зарезервировано
0	DAC1_PSTR	ЧТ/ЗП	0x0	запуск цифроаналогового преобразования DAC1;

1.3.14.7.4.7.7 Регистр записи восьмибитных данных в DAC0 DAC0_D8

Восьмибитные данные, подаваемые на вход цифроаналогового преобразователя DAC0;

В таблице 1.774 представлено описание полей регистра DAC0_D8.

Таблица 1.774 – Описание полей регистра DAC0_D8

Поле	Название	Программный доступ	Состояние после сброса	Описание
31:8	—	ЧТ0	—	Зарезервировано
7:0	DAC0_D8D	ЗП	0x0	Данные данных разрядов отображаются на разряды 11:4 входных данных DAC0;

1.3.14.7.4.7.8 Регистр записи восьмибитных данных в DAC1 DAC1_D8

Восьмибитные данные, подаваемые на вход цифроаналогового преобразователя DAC1;

В таблице 1.775 представлено описание полей регистра DAC1_D8.

Таблица 1.775 – Описание полей регистра DAC1_D8

Поле	Название	Программный доступ	Состояние после сброса	Описание
31:8	—	ЧТ0	—	Зарезервировано
7:0	DAC1_D8D	ЗП	0x0	Данные данных разрядов отображаются на разряды 11:4 входных данных DAC1;

1.3.14.7.4.7.9 Регистр записи восьмибитных данных при синхронной работе двух цифроаналоговых преобразователей DACDBL_D8

Восьмибитные данные, подаваемые на вход цифроаналоговых преобразователей при работе в синхронном режиме;

В таблице 1.776 представлено описание полей регистра DACDBL_D8.

Таблица 1.776 – Описание полей регистра DACDBL_D8

Поле	Название	Программный доступ	Состояние после сброса	Описание
31:16	—	ЧТ0	—	Зарезервировано
15:8	DAC1DBL_D8D	ЗП	0x0	Данные данных разрядов отображаются на разряды 11:4 входных данных DAC1;
7:0	DAC0DBL_D8D	ЗП	0x0	Данные данных разрядов отображаются на разряды 11:4 входных данных DAC0;

1.3.14.7.4.7.10 Регистр записи двенадцатибитных данных в DAC0 DAC0_D12

Двенадцатибитные данные, подаваемые на вход цифроаналогового преобразователя DAC0 ;

В таблице 1.777 представлено описание полей регистра DAC0_D12.

Таблица 1.777 – Описание полей регистра DAC0_D12

Поле	Название	Программный доступ	Состояние после сброса	Описание
31:12	—	ЧТ0	—	Зарезервировано
11:0	DAC0_D12D	ЗП	0x0	Данные данных разрядов отображаются на разряды 11:0 входных данных DAC0;

1.3.14.7.4.7.11 Регистр записи двенадцатибитных данных в DAC1 DAC1_D12

Двенадцатибитные данные, подаваемые на вход цифроаналогового преобразователя DAC1

В таблице 1.778 представлено описание полей регистра DAC1_D12.

Таблица 1.778 – Описание полей регистра DAC1_D12

Поле	Название	Программный доступ	Состояние после сброса	Описание
31:12	—	ЧТ0	—	Зарезервировано
11:0	DAC1_D12D	ЗП	0x0	Данные данных разрядов отображаются на разряды 11:0 входных данных DAC1;

1.3.14.7.4.7.12 Регистр записи двенадцатибитных данных при синхронной работе двух цифроаналоговых преобразователей DACDBL_D12

Двенадцатибитные данные, подаваемые на вход цифроаналоговых преобразователей при работе в синхронном режиме;

В таблице 1.779 представлено описание полей регистра DACDBL_D12.

Таблица 1.779 – Описание полей регистра DACDBL_D12

Поле	Название	Программный доступ	Состояние после сброса	Описание
31:28	—	ЧТ0	—	Зарезервировано
27:16	DAC1DBL_D12D	ЗП	0x0	Данные данных разрядов отображаются на разряды 11:0 входных данных DAC1;
15:12	Reserved	-	0x0	зарезервировано;
11:0	DAC0DBL_D12D	ЗП	0x0	Данные данных разрядов отображаются на разряды 11:0 входных данных DAC0;

1.3.14.7.4.7.13 Регистр следующих данных DAC0 DAC0_HR

Регистр хранит данные, которые будут загружены в DAC0 по следующему событию (без учета wave блоков);

В таблице 1.780 представлено описание полей регистра DAC0_HR.

Таблица 1.780 – Описание полей регистра DAC0_HR

Поле	Название	Программный доступ	Состояние после сброса	Описание
31:12	—	ЧТ0	—	Зарезервировано
11:0	DAC0_HRD	ЧТ	0x0	Данные, которые будут загружены в DAC0;

1.3.14.7.4.7.14 Регистр сформированных данных DAC0 DAC0_SYNR

Регистр хранит сформированные данные, загруженаемые в синхронизирующее FIFO DAC0 (с учетом wave блоков);

В таблице 1.781 представлено описание полей регистра DAC0_SYNR.

Таблица 1.781 – Описание полей регистра DAC0_SYNR

Поле	Название	Программный доступ	Состояние после сброса	Описание
31:12	—	ЧТ0	—	Зарезервировано
11:0	DAC0_SYNRD	ЧТ	0x0	Данные, рагруженые в DAC0;

1.3.14.7.4.7.15 Регистр следующих данных DAC1 DAC1_HR

Регистр хранит данные, которые будут загружены в DAC1 по следующему событию (без учета wave блоков);

В таблице 1.782 представлено описание полей регистра DAC1_HR.

Таблица 1.782 – Описание полей регистра DAC1_HR

Поле	Название	Программный доступ	Состояние после сброса	Описание
31:12	—	ЧТ0	—	Зарезервировано
11:0	DAC1_HRD	ЧТ	0x0	Данные, которые будут загружены в DAC1;

1.3.14.7.4.7.16 Регистр сформированных данных DAC1 DAC1_SYNR

Регистр хранит сформированные данные, загруженые в синхронизирующее FIFO DAC1 (с учетом wave блоков);

В таблице 1.783 представлено описание полей регистра DAC1_SYNR.

Таблица 1.783 – Описание полей регистра DAC1_SYNR

Поле	Название	Программный доступ	Состояние после сброса	Описание
31:12	—	ЧТ0	—	Зарезервировано
11:0	DAC1_SYNRD	ЧТ	0x0	Данные, рагруженые в DAC1;

1.3.14.7.4.7.17 Регистр конфигурации блока цифроаналоговых прелбразователей DAC_CFG

В регистре задается конфигурация работы цифроаналоговых преобразователей при работе в активном режиме;

В таблице 1.784 представлено описание полей регистра DAC_CFG.

Таблица 1.784 – Описание полей регистра DAC_CFG

Поле	Название	Программный доступ	Состояние после сброса	Описание
31:27	—	ЧТ0	—	Зарезервировано
26	DAC1_LWOD	ЧТ/ЗП	0x0	Разрешение режима LWOD (Load WithOut Data) для DAC1: 0 - режим LWOD не активирован; 1 - режим LWOD активирован;
25	DAC1_CTRL	ЧТ/ЗП	0x0	Разрешение аппаратного управление формированием аналогового сигнала DAC1: 0 - аппаратное управление формированием аналогового сигнала запрещено; 1 - аппаратное управление формированием аналогового сигнала разрешено;
24	DAC1_FIFO_EN	ЧТ/ЗП	0x0	Использование входного FIFO данных для DAC1: 0 - входное FIFO не используется; 1 - входное FIFO используется;

Поле	Название	Программный доступ	Состояние после сброса	Описание
23:20	DAC1_WBW	ЧТ/ЗП	0x0	Разиах выходного сигнала блоков LFSR и Triangle_Wave DAC1: 0000 - 1 разряд; 0001 - 2 разряда; 0010 - 3 разряда; … 1010 - 11 разрядов; 1011 - 12 разрядов; 1100 - 1111 - зарезервировано;
19:18	DAC1_WM	ЧТ/ЗП	0x0	Режим работы дополнительных (wave) блоков DAC1: 00 - дополнительные блоки отключены; 01 - включен блок LFSR; 10 - включен блок Triangle_Wave; 11 - зарезервировано;
17	DAC1_DTYP	ЧТ/ЗП	0x0	Тип данных, передаваемых на вход DAC1: 0 - беззнаковые; 1 - знаковые (доп. код);
16	BUF1_EN	ЧТ/ЗП	0x0	Разрешение работы выходного буфера DAC1: 0 - выходной буфер отключен; 1 - выходной буфер включен;
15:11	Reserved	-	0x0	
10	DAC0_LWOD	ЧТ/ЗП	0x0	Разрешение режима LWOD (Load WithOut Data) для DAC0: 0 - режим LWOD не активирован; 1 - режим LWOD активирован;
9	DAC0_CTRL	ЧТ/ЗП	0x0	Разрешение аппаратного управление формированием аналогового сигнала DAC0: 0 - аппаратное управление формированием аналогового сигнала запрещено; 1 - аппаратное управление формированием аналогового сигнала разрешено;
8	DAC0_FIFO_EN	ЧТ/ЗП	0x0	Использование входного FIFO данных для DAC0: 0 - входное FIFO не используется; 1 - входное FIFO используется;
7:4	DAC0_WBW	ЧТ/ЗП	0x0	Разиах выходного сигнала блоков LFSR и Triangle_Wave DAC0: 0000 - 1 разряд; 0001 - 2 разряда; 0010 - 3 разряда; … 1010 - 11 разрядов; 1011 - 12 разрядов; 1100 - 1111 - зарезервировано;
3:2	DAC0_WM	ЧТ/ЗП	0x0	Режим работы дополнительных (wave) блоков DAC0: 00 - дополнительные блоки отключены; 01 - включен блок LFSR; 10 - включен блок Triangle_Wave; 11 - зарезервировано;

Поле	Название	Программный доступ	Состояние после сброса	Описание
1	DAC0_DTYP	ЧТ/ЗП	0x0	Тип данных, передаваемых на вход DAC0: 0 - беззнаковые; 1 - знаковые (доп. код);
0	BUF0_EN	ЧТ/ЗП	0x0	Разрешение работы выходного буфера DAC0: 0 - выходной буфер отключен; 1 - выходной буфер включен;

1.3.14.7.4.7.18 Регистр конфигурации режима Sample&Hold DAC0 DAC0SH_CFG

В регистре задается конфигурация работы DAC0 режиме SampleHOLD

В таблице 1.785 представлено описание полей регистра DAC0SH_CFG.

Таблица 1.785 – Описание полей регистра DAC0SH_CFG

Поле	Название	Программный доступ	Состояние после сброса	Описание
31	—	ЧТ0	—	Зарезервировано
30	DAC0SH_EN	ЧТ/ЗП	0x0	Разрешение работы в режиме Sample&Hold DAC0: 0 - режим работы Sample&Hold не активирован; 1 - режим работы Sample&Hold активирован;
29:28	Reserved	-	0x0	
27:20	DAC0_TREF	ЧТ/ЗП	0x0	Время регенерации данных на выходе DAC0 в режиме Sample&Hold: 00h - 1 такт DAC_CLK; 01h - 2 такта DAC_CLK; 02h - 3 такта DAC_CLK; … FFh – 2^8–1 тактов DAC_CLK;
19:10	DAC0_THLD	ЧТ/ЗП	0x0	Время удержания данных на выходе DAC0 в режиме Sample&Hold: 000h - 1 такт DAC_CLK; 001h - 2 такта DAC_CLK; 002h - 3 такта DAC_CLK; … 3FFh – 2^{10}–1 тактов DAC_CLK;
9:0	DAC0_TSMPL	ЧТ/ЗП	0x0	Время установки данных на выходе DAC0 в режиме Sample&Hold: 000h - 1 такт DAC_CLK; 001h - 2 такта DAC_CLK; 002h - 3 такта DAC_CLK; … 3FFh – 2^{10}–1 тактов DAC_CLK;

1.3.14.7.4.7.19 Регистр конфигурации режима Sample&Hold DAC1 DAC1SH_CFG

В регистре задается конфигурация работы DAC1 режиме SampleHOLD

В таблице 1.786 представлено описание полей регистра DAC1SH_CFG.

Таблица 1.786 – Описание полей регистра DAC1SH_CFG

Поле	Название	Программный доступ	Состояние после сброса	Описание
31	—	ЧТ0	—	Зарезервировано
30	DAC1SH_EN	ЧТ/ЗП	0x0	Разрешение работы в режиме Sample&Hold DAC1: 0 - режим работы Sample&Hold не активирован; 1 - режим работы Sample&Hold активирован;
29:28	Reserved	-	0x0	
27:20	DAC1_TREF	ЧТ/ЗП	0x0	Время регенерации данных на выходе DAC1 в режиме Sample&Hold: 00h - 1 такт DAC_CLK; 01h - 2 такта DAC_CLK; 02h - 3 такта DAC_CLK; … FFh – 2^8–1 тактов DAC_CLK;
19:10	DAC1_THLD	ЧТ/ЗП	0x0	Время удержания данных на выходе DAC1 в режиме Sample&Hold: 000h - 1 такт DAC_CLK; 001h - 2 такта DAC_CLK; 002h - 3 такта DAC_CLK; … 3FFh – 2^{10}–1 тактов DAC_CLK;
9:0	DAC1_TSMPL	ЧТ/ЗП	0x0	Время установки данных на выходе DAC1 в режиме Sample&Hold: 000h - 1 такт DAC_CLK; 001h - 2 такта DAC_CLK; 002h - 3 такта DAC_CLK; … 3FFh – 2^{10}–1 тактов DAC_CLK;

1.3.14.7.4.7.20 Регистр управления синхронной работой DAC0 и DAC1 DAC_DBL

Регистр управляет синхронной раблтой DAC0 и DAC1;

В таблице 1.787 представлено описание полей регистра DAC_DBL.

Таблица 1.787 – Описание полей регистра DAC_DBL

Поле	Название	Программный доступ	Состояние после сброса	Описание
31:2	—	ЧТ0	—	Зарезервировано
1	DBL_CTRL	ЧТ/ЗП	0x0	Управление синхронной работой DAC0 и DAC1: 0 - управление осуществляет DAC0; 1 - управление осуществляет DAC1;

Поле	Название	Программный доступ	Состояние после сброса	Описание
0	DBL_EN	ЧТ/ЗП	0x0	Разрешение синхронной раблты DAC0 и DAC1 : 0 - DAC0 и DAC1 работают независимо; 1 - DAC0 и DAC1работают синхронно;

1.3.14.7.4.7.21 Регистр управления и состояния блока DAC DAC_EN

Регистр разрешает работу цифроаналоговых преобразователей;

В таблице 1.788 представлено описание полей регистра DAC_EN.

Таблица 1.788 – Описание полей регистра DAC_EN

Поле	Название	Программный доступ	Состояние после сброса	Описание
31:4	—	ЧТ0	—	Зарезервировано
3	DAC1_STS	ЧТ	0x0	Состояние DAC1 : 0 - DAC1 выключен; 1 - DAC1 включен;
2	DAC0_STS	ЧТ	0x0	Состояние DAC0 : 0 - DAC0 выключен; 1 - DAC0 включен;
1	DAC1_EN	ЧТ/ЗП	0x0	Разрешение раблты DAC1 : 0 - работа DAC1 запрещена; 1 - работа DAC1 разрешена;
0	DAC0_EN	ЧТ/ЗП	0x0	Разрешение раблты DAC0 : 0 - работа DAC0 запрещена; 1 - работа DAC0 разрешена;

В таблицах 1.789 и 1.790 приведены справочные характеристики цифроаналогового преобразователя.

Таблица 1.789 – Характеристики ЦАП

Наименование параметра, единица измерения	Буквенное обозначение	Режим измерения	Мин	Типовое	Макс
Напряжение питания ЦАП ON	V_{DDA}		1.80	-	3.6
Напряжение смещения положительное, В	V_{REF+}		1.80	-	V_{DDA}
Напряжение смещения отрицательное, В	V_{REF-}	-	V_{SSA}		

Наименование параметра, единица измерения	Буквенное обозначение	Режим измерения		Мин	Типовое	Макс
Резистивная нагрузка, кОм	R_L	выходной буфер ЦАП включен	соединен с V_{SSA}	5	-	-
			соединен с V_{DDA}	25	-	-
Выходной импеданс, кОм	R_O	выходной буфер DAC OFF		9.6	11.7	13.8
Выходной импеданс выборки и хранения, выходной буфер ЦАП ON, кОм	R_{BON}	V_{DD} = 2.7 В		-	-	2
		V_{DD} = 2.0 В		-	-	3.5
Выходной импеданс выборки и хранения, выходной буфер ЦАП OFF, кОм	R_{BOFF}	V_{DD} = 2.7 В		-	-	16.5
		V_{DD} = 2.0 В		-	-	18.0
Емкость нагрузки, пФ	C_L	выходной буфер ЦАП включен		-	-	50
Емкость нагрузки, мкФ	C_{SH}	Режим выборки и хранения		-	0.1	1
Напряжение на выходе DAC1_OUTx, В	V_{DAC_OUT}	выходной буфер ЦАП включен		0.2	-	V_{REF+} – 0.2
		выходной буфер ЦАП выключен		0	-	V_{REF+}
Время установления (полный диапазон: для 12-разрядного кода переключение от минимального до максимального кода на входе DAC1_OUTx достигает конечного значения ±0.5LSB, ±1 LSB, ±2 LSB, ±4 LSB, ±8 LSB), мкс	$t_{SETTLING}$	Нормальный режим выходной буфер ЦАП включен $C_L \leq 50$ пФ, $R_L \geq 5$ кОм	±0.5 EMP	-	1.7	3
			±1 EMP	-	1.6	2.9
			±2 EMP	-	1.55	2.85
			±4 EMP	-	1.48	2.8
			±8 EMP	-	1.4	2.75
		Нормальный режим, выходной буфер ЦАП выключен, ±1EMP, C_L = 10 пФ		-	2	2.5
Время пробуждения из состояния выключено (установка бита ENx в регистре управления ЦАП) до окончательного состояния ±1 EMP, мкс	t_{WAKEUP}	Нормальный режим, выходной буфер ЦАП включен $C_L \leq 50$ пФ, $R_L \geq 5$ кОм		-	4.2	7.5
		Нормальный режим, выходной буфер ЦАП выключен, $C_L \leq 10$ пФ		-	2	5

Наименование параметра, единица измерения	Буквенное обозначение	Режим измерения		Мин	Типовое	Макс
Подавление изменения питания VDDA, дБ	PSRR	Нормальный режим, выходной буфер ЦАП включен $C_L \leq 50$ пФ, $R_L = 5$ кОм, постоянное напряжение		-	-80	-28
Минимальное время между двумя последовательными записями в регистр DAC_DORx для гарантирования корректного DAC1_OUTx для малых изменений входного кода (1 EMP), мкс	$T_{W_to_W}$	$C_L \leq 50$ пФ, $R_L \geq 5$ кОм $C_L \leq 10$ пФ		1 1.4	-	-
Время выборки в режиме выборки и хранения (код переключается от минимального до максимального кода на входе, когда DAC1_OUTx достигает окончательного значения ±1EMP), мс	t_{SAMP}	DAC1_OUTx Выводы соединены	Выходной буфер ЦАП включен, $C_{SH} = 100$ нФ	-	0.7	3.5
			Выходной буфер ЦАП выключен, $C_{SH} = 100$ нФ	-	10.5	18
Время выборки в режиме выборки и хранения (код переключается от минимального до максимального кода на входе, когда DAC1_OUTx достигает окончательного значения ±1 EMP), мкс		DAC1_OUTx Выводы не соединены (только внутреннее соединение)	Выходной буфер ЦАП выключен	-	2	3.5
Выходной ток утечки,нА	I_{leak}	Режим выборки и хранения, DAC1_OUTx выводы соединены		-	-	-
Внутренняя емкость выборки и хранения, пФ	CIint	-		5.2	7	8.8
Время обрезки среднего кода смещения, мкс	t_{TRIM}	Выходной буфер ЦАП ON		50	-	-
Смещение среднего кода для одного шага обрезки, мкВ	Voffset	$V_{REF+} = 3.6$ V		-	1500	-
		$V_{REF+} = 1.8$ V		-	750	-

Наименование параметра, единица измерения	Буквенное обозначение	Режим измерения		Мин	Типовое	Макс
Ток потребления ЦАП от источника питания V_{DDA}, мкА	I_{DDA}	Выходной буфер ЦАП включен	Без нагрузки, средний код (0x800)	-	315	500
			Без нагрузки, худший код (0xF1C)	-	450	670
		Выходной буфер ЦАП выключен	Без нагрузки, средний код (0x800)	-	-	0.2
		Режим выборки и хранения, C_{SH} = 100 пФ		-	315 x Ton/(Ton+ Toff)	670 x Ton/(Ton+ Toff)
Ток потребления ЦАП от источника питания V_{REF+}, мкА	I_{DDV}	Выходной буфер ЦАП включен	Без нагрузки, средний код (0x800)	-	185	240
			Без нагрузки, наихудший код (0xF1C)	-	340	400
		Выходной буфер ЦАП выключен	Без нагрузки, средний код (0x800)	-	155	205
		Режим выборки и хранения, выходной буфер ЦАП включен, C_{SH} = 100 нФ, худший случай		-	185 x Ton/(Ton+ Toff)	400 x Ton/(Ton+ Toff)
		Режим выборки и хранения, выходной буфер ЦАП выключен, C_{SH} = 100 нФ, худший случай		-	155 x Ton/(Ton+ Toff)	205 x Ton/(Ton+ Toff)

Таблица 1.790 – Характеристики ЦАП

Наименование параметра, единица измерения	Буквенное обозначение	Режим измерения		Мин	Типовое	Макс
Дифференциальная нелинейносить, EMP[2]	DNL	выходной буфер ЦАП включен		-	-	±2
		выходной буфер ЦАП выключен		-	-	±2
Монотонность, EMP	-	10 бит		гарантировано		
Интегральная нелинейносить, EMP[3]	INL	выходной буфер ЦАП включен $C_L \leq 50$ пФ, $R_L \geq 5$ кОм		-	-	±4
		выходной буфер ЦАП выключен $C_L \leq 50$ пФ, без R_L		-	-	±4
Ошибка смещения при коде 0x800, EMP[3]	Offset	выходной буфер ЦАП включен $C_L \leq 50$ пФ, $R_L \geq 5$ кОм	V_{REF+} = 3.6 В	-	-	±12
			V_{REF+} = 1.8 В	-	-	±25
		выходной буфер ЦАП выключен $C_L \leq 50$ пФ, без R_L		-	-	±8

Наименование параметра, единица измерения	Буквенное обозначение	Режим измерения		Мин	Типовое	Макс
Ошибка смещения при коде 0x001, EMP[4]	Offset1	выходной буфер ЦАП выключен $C_L \leq 50$ пФ, без R_L		-	-	±5
Ошибка смещения при коде 0x800 после калибровки, EMP	OffsetCal	выходной буфер ЦАП включен $C_L \leq 50$ пФ, $R_L \geq 5$ кОм	$V_{REF+} = 3.6$ В	-	-	±5
			$V_{REF+} = 1.8$ В	-	-	±7
Ошибка усиления, %[5]	Gain	выходной буфер ЦАП включен $C_L \leq 50$ пФ, $R_L \geq 5$ кОм		-	-	±0.5
		выходной буфер ЦАП выключен ≤ 50 пФ, без R_L		-	-	±0.5
Общая нескорректированная ошибка, EMP	TUE	выходной буфер ЦАП включен $C_L \leq 50$ пФ, $R_L \geq 5$ кОм		-	-	±30
		выходной буфер ЦАП выключен $C_L \leq 50$ пФ, no R_L		-	-	±12
Общая нескорректированная ошибка после калибровки, EMP	TUECal	выходной буфер ЦАП включен $C_L \leq 50$ пФ, $R_L \geq 5$ кОм		-	-	±23
Отношение сигнал/шум, дБ	SNR	выходной буфер ЦАП включен $C_L \leq 50$ пФ, RL ≥ 5 кОм 1 кГц BW, fc= 500 кГц		-	71.2	-
		выходной буфер ЦАП выключен $C_L \leq 50$ пФ, без RL, 1 кГц BW, fc=500 кГц		-	71.6	-
Коэффициент нелинейных искажений, дБ	THD	выходной буфер ЦАП включен $C_L \leq 50$ пФ, $R_L \geq 5$ кОм, 1 кГц		-	-78	-
		выходной буфер ЦАП выключен $C_L \leq 50$ пФ, без R_L, 1 кГц		-	-79	-
Отношение сигнал / шум и коэффициент искажения, дБ		SINAD	выходной буфер ЦАП включен $C_L \leq 50$ пФ, $R_L \geq 5$ кОм, 1 кГц		-	70.4
		-	выходной буфер ЦАП выключен $C_L \leq 50$ пФ, без R_L, 1 кГц		-	71

1) Разница между двумя последовательными кодами - 1 EMP.

2) Разница между измеренным значением в коде i и значением в коде i на идеальной линии от кода 0 до 4095.

3) Разница между измеренным значением в коде (0x001) его идеального значения.

4) Разница между идеальным наклоном передаточной характеристики и измеренным наклоном в диапазоне кодов от 0x000 до 0xFFF когда выходной буфер OFF, и от кода, дающего 0.2 В и (V_{REF+} – 0.2) В, когда выходной буфер ON.

1.3.14.8 Источник опорного напряжения (VREF)

1.3.14.8.1 Введение

Устройство буфера внутреннего опорного напряжения предназначено для формирования набора опорных напряжений, необходимых для работы блоков СБИС.

1.3.14.8.1.1 Структурная схема буфера внутреннего опорного напряжения

Структурная схема буфера внутреннего опорного напряжения представлена на рисунке 1.174.

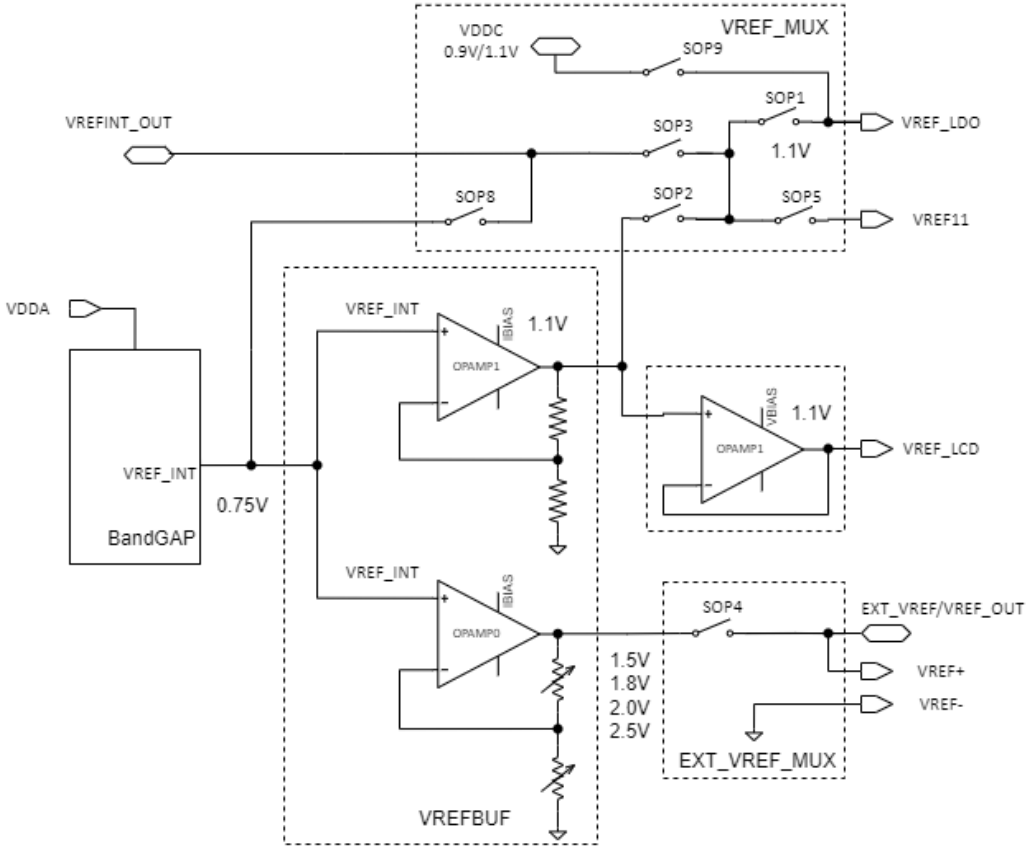


Рисунок 1.174 – Структурная схема буфера внутреннего опорного напряжения

Блок буфера внутреннего опорного напряжения содержит следующие узлы и блоки:

– Блок источника опорного напряжения и опорных токов ИОН (BandGAP)

– Блок буферов опорного напряжения VREFBUF

– Блок мультиплексирования внутренних опорных напряжений VREF_MUX

– Блок мультиплексирования внешнего опорного напряжения EXT_VREF_MUX

– Блок буфера опорного напряжения для буферов ввода-вывода интерфейса ЖК дисплея VREF_LCD

1.3.14.8.2 Функциональное описание VREFBUF

Буфер внутреннего опорного напряжения поддерживает следующие функции.

– Формирование опорного напряжения VREF+, определяющего верхнюю границу шкалы АЦП и ЦАП из сигнала с внешнего входа микросхемы EXT_VREF (Режим внешнего опорного напряжения).

– Формирование опорного напряжения VREF+, определяющего верхнюю границу шкалы АЦП и ЦАП из сигнала ИОН путем усиления до необходимого уровня (Режим внутреннего опорного напряжения).

– Формирование опорного напряжения для питания внутреннего LCD АЦП, VREF_LDO

– Формирование опорного напряжения для питания схем аналогового компаратора и термодатчика VREF11

– Формирование опорного напряжения для питания блоков буферов интерфейса ЖК дисплея VREF_LCD

Внутреннее опорное напряжение можно настроить в четырех различных режимах в зависимости от битов в регистре управления VREF_BUF.

1.3.14.8.2.1 Основные параметры буфера внутреннего опорного напряжения

В таблице 1.791 приведены основные параметры буфера внутреннего опорного напряжения.

Таблица 1.791 – Основные параметры буфера внутреннего опорного напряжения

Параметр	Мин.	Тип.	Макс.	Единицы
Напряжение аналогового питания	1,8		3,6	В
Напряжение цифрового питания	0,81		1.21	В
Ток потребления (на частоте переключения 1МГц.)			100	мкА
Ток потребления в спящем режиме			0,5	мкА
Напряжение встроенного ИОН		0,75		В
Напряжение VREF+ при питании от внешнего источника	0		3,6	В
Напряжение VREF+ при питании от встроенного ИОН	1,5		2,5	В
Напряжение VREF11		1.1		В

Напряжение VREF_LDO		1.1		В
Напряжение VREF_LCD		1.1		В

После первоначального включения питания VREFBUF оказывается в выключенном состоянии. Ключи внутренней коммутации разомкнуты. Для использования VREFBUF его необходимо сконфигурировать.

1.3.14.8.2.2 Регистровая карта блока буфера внутреннего опорного напряжения

Смещение

0x0

Размер

0x1c

1.3.14.8.2.2.1 Регистры

Смещение	Идентификатор	Название	
0x00	r_VREF_ENABLE	COMPx_CSR	#r_VREF_ENABLE
0x04	r_VREF_SELECT_OPERANDS	VREF_SELECT_OPERANDS	#r_VREF_SELECT_OPERANDS
0x08	r_VREF_SELECT_LEVEL	VREF_SELECT_LEVEL	#r_VREF_SELECT_LEVEL
0x0C	r_VREF_BANDGAP_CONTROL	VREF_BANDGAP_CONTROL	#r_VREF_BANDGAP_CONTROL
0x10	r_VREF_TRIMMING	VREF_TRIMMING_CONTROL	#r_VREF_TRIMMING
0x14	r_VREF_LCD_ENABLE	VREF_LCD_ENABLE	#r_VREF_LCD_ENABLE
0x18	r_VREF_AVDD_LDO_ENABLE	VREF_AVDD_LDO_ENABLE	#r_VREF_AVDD_LDO_ENABLE

1.3.14.8.2.2.2 VREF_CSR

Смещение

0x0

Размер

0x4

1.3.14.8.2.2.2.1 Описание

COMPx Control and Status register

1.3.14.8.2.2.2 Состав

Bits	Identifier	Access	Reset	Decoded	Name	
[31:1]	-	-	-		-	
[0]	EN	rw	0x0	ENABLE DISABLE	EN	H #EN

1.3.14.8.2.2.3 VREF_AVDD_LDO_ENABLE

Смещение

0x18

Размер

0x4

1.3.14.8.2.2.3.1 Описание

AVDD power source to bypass ADC LDO

1.3.14.8.2.2.3.2 Состав

Bits	Identifier	Access	Reset	Decoded	Name	
[31:1]	-	-	-		-	
[0]	EN	rw	0x0	ENABLE DISABLE	EN	H #EN

1.3.14.8.2.2.4 VREF_BANDGAP_CONTROL

Смещение

0xc

Размер

0x4

1.3.14.8.2.2.4.1 Описание

VREF Bandgap Control

1.3.14.8.2.2.4.2 Состав

Bits	Identifier	Access	Reset	Decoded	Name	
[31:18]	-	-	-		-	
[17]	BGP_OK_LV	r	0x0		BGP_OK_LV	H #BGP_OK_LV
[16]	BGP_OK_HV	r	0x0		BGP_OK_HV	H #BGP_OK_HV

Bits	Identifier	Access	Reset	Decoded	Name	
[15:7]	-	-	-		-	
[6:1]	BGREG	rw	0x0	bgreg5 bgreg4 bgreg3 bgreg2 bgreg1 bgreg0	BGREG	H #BGREG
[0]	EN	rw	0x0	ENABLE DISABLE	EN	H #EN

1.3.14.8.2.2.5 VREF_LCD_ENABLE

Смещение

0x14

Размер

0x4

1.3.14.8.2.2.5.1 Описание

Voltage reference for LCD buffer

1.3.14.8.2.2.5.2 Состав

Bits	Identifier	Access	Reset	Decoded	Name	
[31:1]	-	-	-		-	
[0]	EN	rw	0x0	ENABLE DISABLE	EN	H #EN

1.3.14.8.2.2.6 VREF_SELECT_LEVEL

Смещение

0x8

Размер

0x4

1.3.14.8.2.2.6.1 Описание

VREF Select Level

1.3.14.8.2.2.6.2 Состав

Bits	Identifier	Access	Reset	Decoded	Name	
[31:4]	-	-	-		-	
[3:0]	LVL	rw	0x0	level3 level2 level1 level0	LVL	H #LVL

1.3.14.8.2.2.7 VREF_SELECT_OPERANDS

Смещение

0x4

Размер

0x4

1.3.14.8.2.2.7.1 Описание

VREF Select Operands

1.3.14.8.2.2.7.2 Состав

Bits	Identifier	Access	Reset	Decoded	Name	
[31:12]	-	-	-		-	
[11:0]	SOP	rw	0x0	sop11 sop10 sop09 sop08 sop07 sop06 sop05 sop04 sop03 sop02 sop01 sop00	SOP	H #SOP

1.3.14.8.2.2.8 VREF_TRIMMING_CONTROL

Смещение

0x10

Размер

0x4

1.3.14.8.2.2.8.1 Описание

VREF Trimming Control

1.3.14.8.2.2.8.2 Состав

Bits	Identifier	Access	Reset	Decoded	Name	
[31:0]	TRIM	rw	0x0		TRIM	H #TRIM

В таблице 1.792 приведены справочные характеристики буфера внутреннего опорного напряжения.

Таблица 1.792 – Характеристики источника опорного напряжения

Наименование параметра, единица измерения	Буквенное обозначение	Режим измерения		Мин	Типовое	Макс
Напряжение питания аналоговое, В	V_{DDA}	Нормальный режим	V_{RS} = 0	2.4	-	3.6
		Ограниченный режим	V_{RS} = 0	1.8	-	2.4
Выходное опорное напряжение, В	V_{REFBUF_OUT}	Нормальный режим	V_{RS} = 0	1.5	-	2.5
		Ограниченный режим	V_{RS} = 0	V_{DDA}-150 мВ	-	V_{DDA}
Шаг калибровки, %	T_{RIM}	-	-	-	±0,05	±0,1
Емкость нагрузки, мкФ	C_L	-	-	0.5	1	1.5
Эквивалентное последовательное сопротивление нагрузки Cload, Ом	esr	-	-	-	-	2
Статический ток нагрузки, мА	I_{load}	-	-	-	-	4
Линейное управление, млн−1 /В	I_{line_reg}	2,8 В ≤ V_{DDA} ≤ 3,6 В	I_{load} = 500 мкА	-	200	1000
			I_{load} = 4 мА	-	100	500
Управление нагрузкой, млн−1 /В	I_{load_reg}	500 мкА ≤ I_{load} ≤4 мА	Нормальный режим	-	50	500
Температурный коэффициент, млн$^{-1}$ / оС	T_{Coeff}	-40 °C < TJ < +125 °C		-	-	Tcoeff_vrefint + 50
		0 °C < TJ < +50 °C		-	-	Tcoeff_vrefint + 50
Коэффициент подавление помех по питанию, дБ	PSRR	Постоянное напряжение		40	60	-
		100 кГц		25	40	-
Время старта, мкс	t_{START}	CL = 0,5 мкФ		-	300	350
		CL = 1,1 мкФ		-	500	650
		CL = 1,5 мкФ		-	650	800
Ток потребления V_{REFBUF} от источника питания V_{DDA} , мкА	I_{DDA}	I_{load} = 0 мкА		-	16	25
		I_{load} = 500 мкА		-	18	30
		I_{load} = 4 мА		-	35	50

1.3.14.9 Аналоговый компаратор (ACOMP)

1.3.14.9.1 Введение

Блок аналоговых компараторов напряжения предназначен для сравнивания аналоговых сигналов, подаваемых с внешних выводов микроконтроллера, с аналогичными сигналами или с напряжением, генерируемым непосредственно самим микроконироллером. Блок представляет собой два аналоговых компаратора с аналоговыми мультиплексорами на входах и схему конфигурирования входов и выходов блока. Упрощенная схема формирования выходных сигналов компараторов представлена на рисунке 1.175.

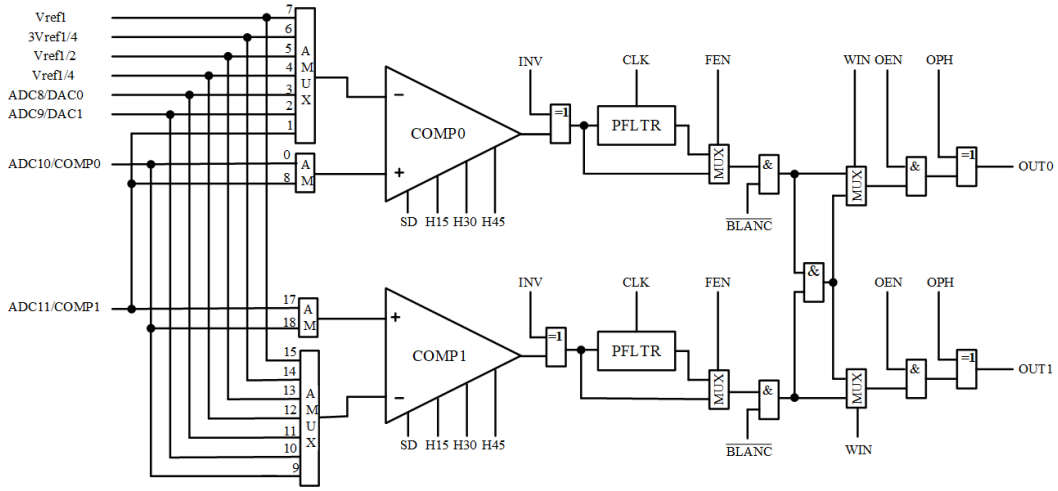


Рисунок 1.175 – Упрощеннная схема формирования выходных сигналов компараторов

Устройство предусматривает два маломощных компаратора COMP0 и COMP1

Компараторы используются для различных функций, в том числе:

1) Активация из режима пониженного энергопотребления при срабатывании аналогового сигнала,

2) Контур поциклового регулирования тока в сочетании с выходным сигналом PWM от таймера.

Конфигурация блока осуществляется по шине AMBA_AHB спецификации 2.0 со следующими особенностями:

– циклы чтения и запмсм выполняются в блоке за один такт;

– аппаратура интерфейса AMBA_AHB блока не поддерживает пакетную передачу данных.

Каждый аналоговый компаратор настраивается индивидуально и может работать как независимо друг от друга, так и совместно в режиме "окна". В последнем случае выходы компараторов объединяются по "и" и выдаются на выбранный в процессе настройки вывод блока.

Выходы блока аналоговых компараторов могут работать как асинхронно, т.е. не синхронизированно ни с какой внутренней частотой микроконтроллера, так и синхронно. В последнем случае тактовым сигналом, синхронизирующим выходы блока, является тактовый сигнал шины AMBA_AHB. Если выходы блока конфигурируются как асинхронные, то после программной настройки блока тактовый сигнал шины AMBA_AHB может быть снят.

Каждый аналоговый компаратор имеет встроенную схему формирования гистерезиса с управляемым порогом. На выходе каждого компаратора возможно использование цифрового фильтра, подавляющего случайные помехи. Необходимо отметить, что синхронный выход юлока осуществляется с выходного регистра фильтра и соответственно смещен на время фильтрации, относительно срабатывания самого компаратора.

1.3.14.9.2 Основные характеристики COMP

1) Каждый компаратор имеет переключаемые положительные и отрицательные входы, используемые для выбора источника сигнала:

a) Мультиплексные контакты ввода-вывода

b) Канал 1 и канал 2 цифро-аналогового преобразователя

c) Внутреннее опорное напряжениеп и три дольные единицы (1/4, 1/2, 3/4), обеспечиваемые делителем (буферным делителем напряжения).

2) Настраиваемый гистерезис

3) Выход компаратопа может быть как асинхронным, так и синхронизитованным с частотой работы управляющей шины

4) Выходные сигналы компаратора можно перенаправить на внешние выводы контроллера, на входы таймера или входы прерывания процессорного ядра контроллера (через контроллер EXTI)

5) Выход компаратора имеет возможность подавления коротких импульсов срабатывания внешним по отношению к контроллеру сигналом

6) Возможность работы пары компараторов в режиме "окна" (двухпороговый режим)

1.3.14.9.3 Основные параметры компараторов

В таблице 1.793 приведены основные параметры аналоговых компараторов.

Таблица 1.793 – Характеристики компаратора

Наименование параметра, единица измерения	Буквенное обозначение	Режим измерения		Мин	Типовое	Макс
Напряжение питания аналоговое, В	V_{DDA}	-		1.8	-	3.6
Диапазон входного напряжения компаратора, В	VIN	-			-	V_{DDA}
Время старта компаратора до достижения заданной задержки распространения, мкс	t_{START}	$V_{DDA} \geq 2,7$ В		-	-	15
		$V_{DDA} < 2,7$ В		-	-	25
Задержка распространения для шага 200 мВ с 100 мВ превышением сигнала, мкс	t_D	$V_{DDA} \geq 2,7$ В		-	0.55	0.9
		$V_{DDA} < 2,7$ В		-	0.65	1
Ошибка смещения компаратора, мВ	V_{offset}	Полный диапазон синфазного сигнала	-	-	±5	±20
Гистерезис компаратора, мВ	V_{hys}	Нет гистерезиса		-		-
		Малый гистерезис		-	8	-
		Средний гистерезис		-	15	-
		Большой гистерезис		-	27	-
Ток потребления от источника питания V_{DDA} мкА	I~DDA``	Средний режим	статика	-	5	7
			При 50 кГц ±100 мВ превышения прямоугольного сигнала	-	6	-
Входной ток смещения компаратора, нА	I_{bias}	-		-	-	-

1.3.14.9.4 Механизм блокировки конфмгурационных регистров компаратора

В случае использования компараторов для выполнения операций с особыми требованиями к безопасности необходимо исключить несанкционированный доступ к регистрам конфигурации компаратора. Это неоюходимо например для обеспечения защиты от перегрузки по току или защиты от перегрева и т.п.

Для этой цели регистры конфигурации каждого компаратора можно защитить от записи установкой бита ACMPi_CFGDIS регистра конфигурации компаратора ACMPi_CFG.

После завершения настройки установка бита ACMPi_CFGDIS запрещает любую запись в данный регистр и регистр становится доступным только для чтения.

Защиту от записи можно деактивировать только сбросив микроконтроллер.

1.3.14.9.5 Работа блока компараторов в режиме "окна"

Целью работы блока компараторов в данном режиме заключается в отслеживании нахождения аналогового напряжения в пределах заданного диапазона, определенного нижним и верхним пределами. При этом один из компараторов настраивается на отслеживание верхнего порога, а другой на отслеживание нижнего порога.

Для работы в данном режиме необходимо с помощью полей ACMPi_N и ACMPi_P выбрать требуемые пороги и источник входного сигнала и полем ACMPi_INV соответствующую фазу выходных сигналов компараторов. Поле ACMPi_WIN задает выход блока, на который выдается результирующий сигнал компараторов.

1.3.14.9.6 Гистерезис

Компаратор предусматривает программируемый гистерезис во избежание несанкционированных переходов выходного сигнала в случае сигналов с помехами. Работа компаратора с заданным гистерезисом представлена на рисунке 1.176.

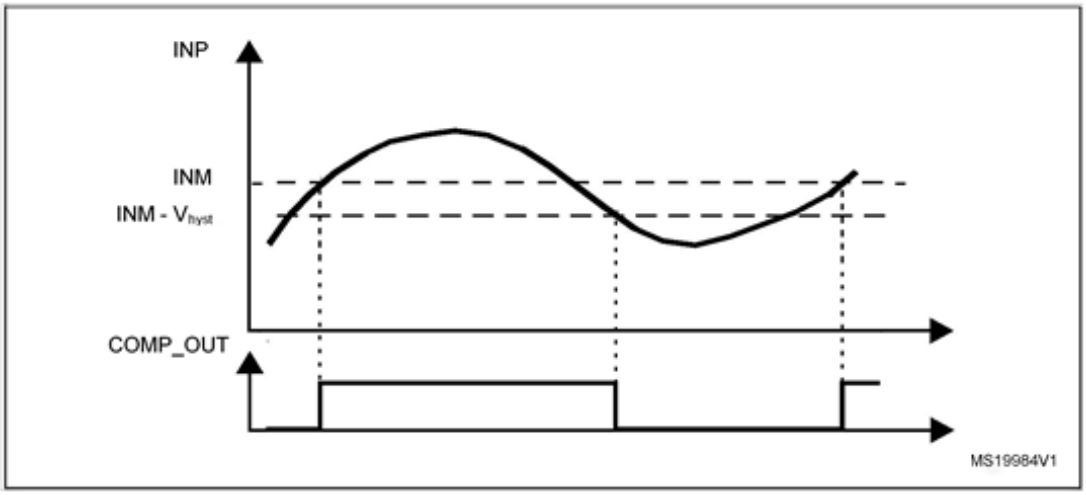


Рисунок 1.176 – Гистерезис компаратора

Размах гистерезиса задается полем ACMPi_HIS регистра конфигурации ACMPi_CFG соответствующего компаратора и выбирается из ряда: 0 - 15 - 30 - 45 мВ.

1.3.14.9.7 Функция подавления коротких импульсов срабатывания компаратора

Функция подавления коротких импульсов срабатывания компаратора используется для предотвращения срабатывания компаратора при резком изменении входного сигнала (переходной процесс изменения входного сигнала), который может привести к короткому выбросу превышающему порог срабатывания компаратора.

Функция подавления реализуется с помощью внешнего вывода BLANCi блока, отдельного для каждого компаратора. Сигнал на входе BLANCi объединяется с выходным сигналом посредством операции «И», блокируя тем самым выход блока.

Пример работы функция подавления приведен на рисунке 1.177.

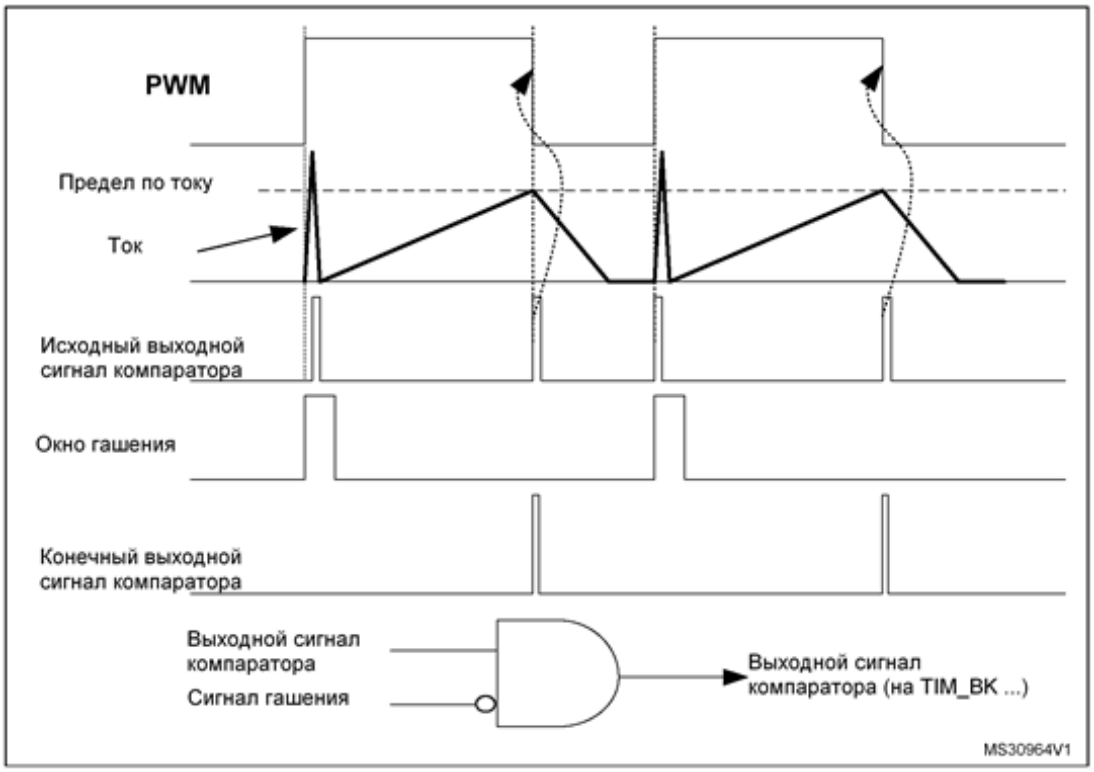


Рисунок 1.177 – Функция подавления коротких импульсов срабатывания компаратора

1.3.14.9.8 Работа цифрового фильтра блока аналоговых компараторов

Цифровой фильтр на выходе компаратора представляет собой цепочку из трех триггеров, соединеных последовательно и формирующих управляющие сигналы для выходного триггера фильтра. Работа фильтра в случае, когда он повторяет выход аналогового компаратора проиллюстрирована на рисунке 1.178.

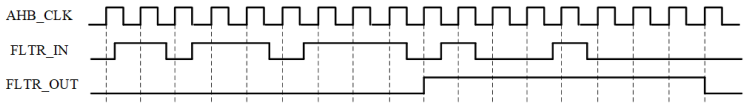


Рисунок 1.178 – Работа фильтра в случае, когда он повторяет выход аналогового компаратора

Работа фильтра, в случае, когда выходной регистр фильтра сбрасывается программно проиллюстрирована на рисунке 1.179.

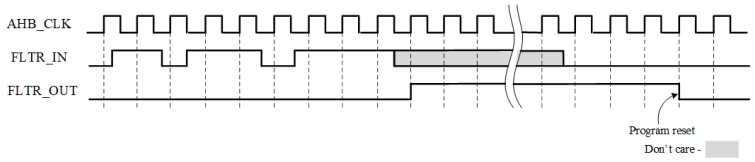


Рисунок 1.179 – Работа фильтра, в случае, когда выходной регистр фильтра сбрасывается программно

1.3.14.9.8.1 Программно доступные регистры канала записи

В таблице 1.794 представлен список программно доступных регистров канала записи.

Все программно доступные регистры блока аналоговых компараторов расположены начиная с базового адреса устройства – ACOMP_Base.

Таблица 1.794 – Список программно доступных регистров канала записи

Название	Адрес	Описание
ACMP0_CFG	0x000	Регистр конфигурации компаратора 0
ACMP1_CFG	0x004	Регистр конфигурации компаратора 1
ACMP_STS	0x008	Регистр состояния компараторов
ACMP_PRST	0x00C	Регистр программного сброса выходных регистров компаратора

1.3.14.9.8.1.1 Регистр конфигурации компаратора 0 ACMP0_CFG

В регистре задается параметры работы компаратора 0

В таблице 1.795 представлено описание полей регистра ACMP0_CFG.

Таблица 1.795 – Описание полей регистра ACMP0_CFG

Поле	Название	Программный доступ	Состояние после сброса	Описание
31:14	—	ЧТ0	—	Зарезервировано
13	ACMP0_OPH	ЧТ/ЗП	0x0	Активная фаза сигнала на выходе OUT0 блока: 0 - активный уровень сигнала на выходе высокий; 1 - активный уровень сигнала на выходе низкий;
12	ACMP0_OEN	ЧТ/ЗП	0x0	Разрешение вывода сигнала компаратора на внешний вывод OUT0 блока: 0 - вывод сигнала запрещен; 1 - вывод сигнала разрешен;
11	ACMP0_WIN	ЧТ/ЗП	0x0	Источник сигнала на выходе Out0 блока: 0 - на выход блока выводится только сигнал компаратора; 1 - на выход блока выводится результирующий сигнал при работе обоих компараторов в режиме “окна”;

Поле	Название	Программный доступ	Состояние после сброса	Описание
10	ACMP0_PRSTEN	ЧТ/ЗП	0x0	Разрешение программного сброса триггера на выходе фильтра компаратора 0: 0 - триггер повторяет состояние на выходе компаратора (с учетом времени фильтрации); 1 - триггер на выходе устанавливается компаратором и сбрасывается программно;
9	ACMP0_FENV	ЧТ/ЗП	0x0	Разрешение работы фильтра на выходе компаратора 0: 0 - фильтр на выходе не используется; 1 - фильтр на выходе используется;
8	ACMP0_INV	ЧТ/ЗП	0x0	Инверсия сигнала на выходе компаратора 0: 0 - сигнал на выходе компаратора не инвертируется; 1 - сигнал на выходе компаратора инвертируется;
7:6	ACMP0_HIS	ЧТ/ЗП	0x0	Размах гистерезиса компаратора компаратора 0: 0b00 - гистерезис отсутствует; 0b01 - размах гистерезиса 15 мВ.; 0b10 - размах гистерезиса 30 мВ.; 0b11 - размах гистерезиса 45 мВ.;
5:4	ACMP0_P	ЧТ/ЗП	0x0	Управление мультиплексором на положительном входе компаратора 0: 0b00 - положительный вход отключен; 0b01 - на вход подается сигнал ADC10/COMP0; 0b10 - на вход подается сигнал ADC11/COMP1; 0b11 - зарезервировано;
3:1	ACMP0_N	ЧТ/ЗП	0x0	Управление мультиплексором на отрицательном входе компаратора 0: 0b000 – отрицательный вход отключен; 0b001 – на вход подается сигнал ADC11/COMP1; 0b010 – на вход подается сигнал ADC9/DAC1; 0b011 – на вход подается сигнал ADC8/DAC0; 0b100 – на вход подается напряжение Vref/4; 0b101 – на вход подается напряжение Vref/2 ; 0b110 – на вход подается напряжение 3*Vref/4; 0b111 – на вход подается напряжение Vref;
0	ACMP0_EN	ЧТ/ЗП	0x0	Разрешение работы компаратора 0: 0 - работа компаратора запрещена; 1 - работа компаратора разрешена;

1.3.14.9.8.1.2 Регистр конфигурации компаратора 1 ACMP1_CFG

В регистре задается параметры работы компаратора 1

В таблице 1.796 представлено описание полей регистра ACMP1_CFG.

Таблица 1.796 – Описание полей регистра ACMP1_CFG

Поле	Название	Программный доступ	Состояние после сброса	Описание
31:14	—	ЧТ0	—	Зарезервировано
13	ACMP1_OPH	ЧТ/ЗП	0x0	Активная фаза сигнала на выходе OUT1 блока: 0 - активный уровень сигнала на выходе высокий; 1 - активный уровень сигнала на выходе низкий;
12	ACMP1_OEN	ЧТ/ЗП	0x0	Разрешение вывода сигнала компаратора на внешний вывод OUT1 блока: 0 - вывод сигнала запрещен; 1 - вывод сигнала разрешен;
11	ACMP1_WIN	ЧТ/ЗП	0x0	Источник сигнала на выходе Out1 блока: 0 - на выход блока выводится только сигнал компаратора; 1 - на выход блока выводится результирующий сигнал при работе обоих компараторов в режиме “окна”;
10	ACMP1_PRSTEN	ЧТ/ЗП	0x0	Разрешение программного сброса триггера на выходе фильтра компаратора 1: 0 - триггер повторяет состояние на выходе компаратора (с учетом времени фильтрации); 1 - триггер на выходе устанавливается компаратором и сбрасывается программно;
9	ACMP1_FENV	ЧТ/ЗП	0x0	Разрешение работы фильтра на выходе компаратора 1: 0 - фильтр на выходе не используется; 1 - фильтр на выходе используется;
8	ACMP1_INV	ЧТ/ЗП	0x0	Инверсия сигнала на выходе компаратора 1: 0 - сигнал на выходе компаратора не инвертируется; 1 - сигнал на выходе компаратора инвертируется;
7:6	ACMP1_HIS	ЧТ/ЗП	0x0	Размах гистерезиса компаратора компаратора 1: 0b00 - гистерезис отсутствует; 0b01 - размах гистерезиса 15 мВ.; 0b10 - размах гистерезиса 30 мВ.; 0b11 - размах гистерезиса 45 мВ.;
5:4	ACMP1_P	ЧТ/ЗП	0x0	Управление мультиплексором на положительном входе компаратора 1: 0b00 - положительный вход отключен; 0b01 - на вход подается сигнал ADC11/COMP1; 0b10 - на вход подается сигнал ADC10/COMP0; 0b11 - зарезервировано;

Поле	Название	Программный доступ	Состояние после сброса	Описание
3:1	ACMP1_N	ЧТ/ЗП	0x0	Управление мультиплексором на отрицательном входе компаратора 1: 0b000 – отрицательный вход отключен; 0b001 – на вход подается сигнал ADC10/COMP0; 0b010 – на вход подается сигнал ADC9/DAC1; 0b011 – на вход подается сигнал ADC8/DAC0; 0b100 – на вход подается напряжение Vref/4; 0b101 – на вход подается напряжение Vref/2 ; 0b110 – на вход подается напряжение 3*Vref/4; 0b111 – на вход подается напряжение Vref;
0	ACMP1_EN	ЧТ/ЗП	0x0	Разрешение работы компаратора 1: 0 - работа компаратора запрещена; 1 - работа компаратора разрешена;

1.3.14.9.8.1.3 Регистр состояния компараторов ACMP_STS

В регистре CMP_STS отражается сотояние выходов компараторов (с учетом задержки фильтрации)

В таблице 1.797 представлено описание полей регистра ACMP_STS.

Таблица 1.797 – Описание полей регистра ACMP_STS

Поле	Название	Программный доступ	Состояние после сброса	Описание
31:2	—	ЧТ0	—	Зарезервировано
1	ACMP1_STS	ЧТ	0x0	Состояние непосредственного выхода компаратора 1
0	ACMP0_STS	ЧТ	0x0	Состояние непосредственного выхода компаратора 0

1.3.14.9.8.1.4 Регистр программного сброса выходных регистров компаратора ACMP_PRST

Запись логической единицы в разряды данного регистра приводит к сбросу выходных регистров компаратора. Сброс возможен при установке полей ACMP1_PRSTEN и ACMP0_PRSTEN регистров конфигурации

В таблице 1.798 представлено описание полей регистра ACMP_PRST.

Таблица 1.798 – Описание полей регистра ACMP_PRST

Поле	Название	Программный доступ	Состояние после сброса	Описание
31:2	—	ЧТ0	—	Зарезервировано
1	ACMP1_PRST	ЗП	0x0	Программный сброс выходного регистра компаратора 1
0	ACMP0_PRST	ЗП	0x0	Программный сброс выходного регистра компаратора 0

1.3.14.10 Операционный усилитель (PGA)

1.3.14.10.1 Введение

Микросхема содержит два операционных усилителя с двумя входами и одним выходом каждый. Три входа-выхода можно подключить к внешним контактам, что обеспечивает внешнее соединение любого типа. Операционный усилитель можно настроить внутри в качестве буфера повторителя или усилителя с неинвертирующим усилением от 2 до 16.

Положительный и отрицательный входы можно подключить к внутреннему цифро-аналоговому преобразователю.

Выход также можно подключить к внутреннему аналого-цифровому преобразователю.

1.3.14.10.2 Основные характеристики ОУ

– Низкий входной ток смещения

– Низкое входное напряжение смещения

– Короткое время активации

– Ширина полосы усиления 1,6 МГц

1.3.14.10.3 Описание функций ОУ

ОУ предусматривает несколько режимов.

Каждый ОУ можно отдельно активировать, при отключении выход становится высокоимпедансным.

В рабочем режиме входы и выход ОУ подключаются, как описано в 1.3.14.10.3.3.

1.3.14.10.3.1 Включение и выключение ОУ

Бит EN включает и отключает операционный усилитель. Параметры регистров ОУ должны быть установлены в требуемое положение перед включением бита EN во избежание ложных эффектов на выходе.

Если выход операционного усилителя не нужен, операционный усилитель можно отключить для экономии энергии. Все ранее заданные параметры //// (включая параметры калибровки) //// сохраняются, пока операционный усилитель отключен.

1.3.14.10.3.2 Исходная конфигурация

Конфигурация по умолчанию операционного усилителя - рабочий режим, в котором три входа-выхода подключены к внешним контактам. В режиме по умолчанию операционный усилитель использует заводские настройки. Лист данных с заводскими настройками см. в разделе электрических характеристик. Стандартные условия - температура 30 °C и напряжение 3 В. Значения можно отрегулировать.

После задания бита EN в регистре PGAx_CSR операционный усилитель включится. Два входных контакта и выходной контакт подключаются в соответствии с пп 1.3.14.10.3.3 «Прохождение сигналов», и можно изменить параметры соединения по умолчанию.

Примечание: входные и выходные контакты можно настроить в аналоговом режиме (состоянии по умолчанию) в соответствующем регистре управления GPIO.

1.3.14.10.3.3 Прохождение сигналов

Прохождение сигналов для контактов операционного усилителя определяется регистром ОУ_CSR.

Соединения двух операционных усилителей (ОУ1 и ОУ2) указаны в таблице 1.799.

Таблица 1.799 – Возможные соединения операционных усилителей

Сигнал	Контакт	Внутренний	Примечание
ОУ1_VINM	PA1 или отдельный контакт(1)	ОУ1_OUT или усилитель с программируемым усилением	Управление битами OPAMODE и VM_SEL.
ОУ1_VINP	PA0	DAC1_OUT1	Управление битом VP_SEL.
ОУ1_VOUT	PA3	ADC1_IN8 ADC2_IN8	Контакт подключается, когда включается операционный усилитель. Вход аналого-цифрового преобразователя управляется аналого-цифровым преобразователем.
ОУ2_VINM	PA7 или отдельный контакт(1)	ОУ2_OUT или усилитель с программируемым усилением	Управление битами OPAMODE и VM_SEL.
ОУ2_VINP	PA6	DAC1_OUT2	управляется битом VP_SEL

Сигнал	Контакт	Внутренний	Примечание
ОУ2_VOUT	PB0	ADC1_IN15 ADC2_IN15	Контакт подключается, когда включается операционный усилитель. Вход аналого-цифрового преобразователя управляется аналого-цифровым преобразователем.

1.3.14.10.3.4 Режимы операционного усилителя

Входы и выходы операционных усилителей доступны на клеммах. Усилители можно использовать в различных конфигурациях:

– Автономный режим (режим внешней настройки усиления),

– Режим буфера повторителя,

– Режим усилителя с программируемым усилением.

Примечание: выходной контакт усилителя напрямую подключается к контактной площадке для вывода сигналов с целью минимизации импеданса выходного сигнала. Его нельзя использовать в качестве универсального входа-выхода, даже если включен режим усилителя с программируемым усилением. Это выход подключается только к каналу аналого-цифрового преобразователя.

Примечание: импеданс сигнала следует поддерживать ниже заданного уровня во избежание тока утечки на входе и формирования значительных артефактов (из-за падения напряжения в активном сопротивлении в источнике). Дополнительную информацию см. в разделе электрических характеристик в паспорте.

Автономный режим (режим внешней настройки усиления)

Прядок использования операционного усилителя в автономном режиме описан ниже (см рисунок 1.180).

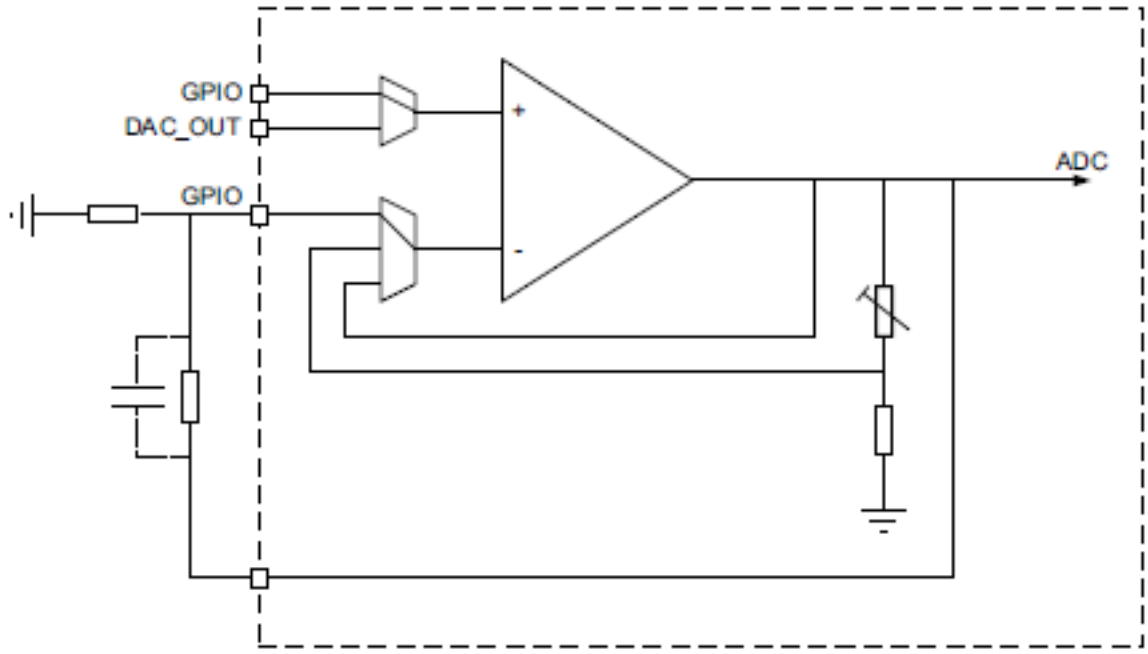


Рисунок 1.180 – Автономный режим: режим внешней настройки усиления

Режим следящего звена

Порядок использования операционного усилителя в качестве буфера повторителя описан ниже (см рисунок 1.181).

Примечание: контакт, соответствующий ОУ_VINM, может использоваться для других целей.

Примечание: сигнал на выходе операционного усилителя также является входным сигналом АЦП. Как следствие, операционный усилитель, настроенный в качестве следящего звена, может использоваться для адаптации импеданса входных сигналов перед их передачей на вход АЦП при условии, что частота входного сигнала совместима с шириной полосы усиления операционного усилителя.

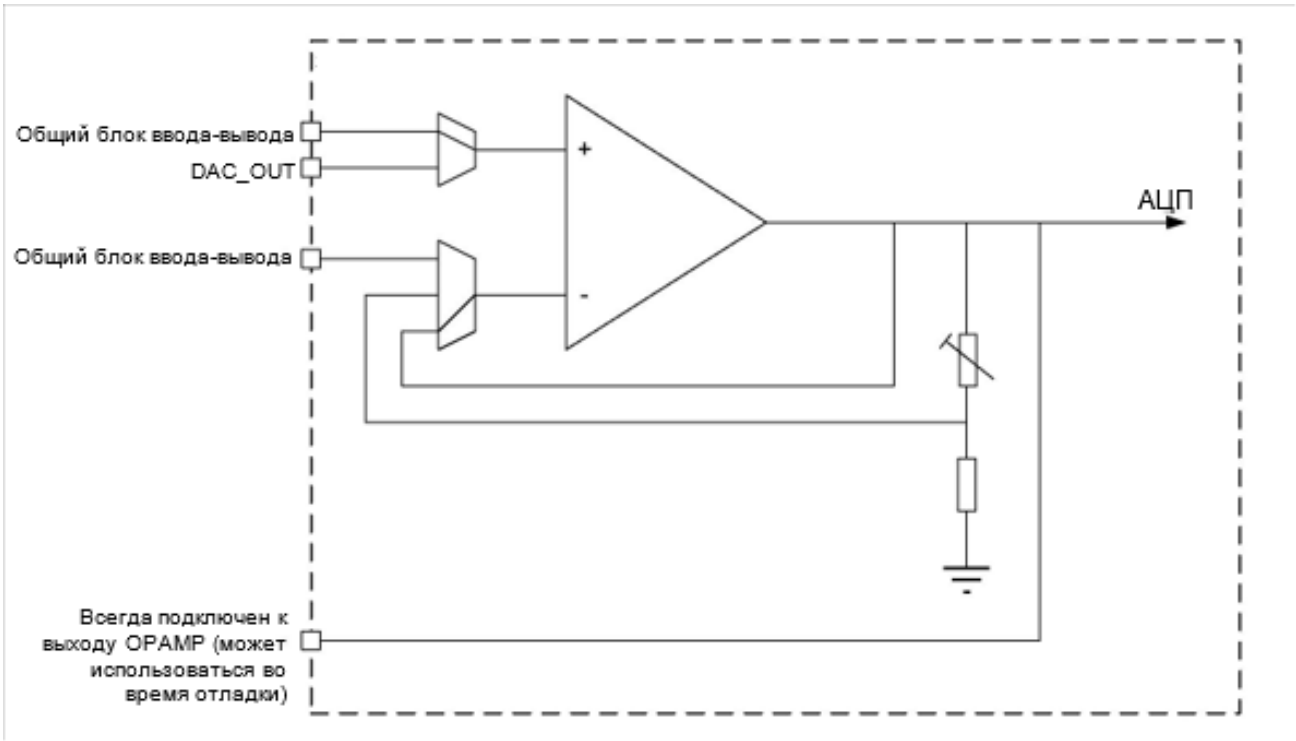


Рисунок 1.181 – Автономный режим: режим внешней настройки усиления

Режим усилителя с программируемым усилением

Порядок использования операционного усилителя для усиления амплитуды входного сигнала описан ниже (см рисунок 1.182).

– Установите биты PGA_GAIN на «Усиление внутреннего усилителя 2, 4, 8 или 16»,

После задания бита EN сигнал на контакте PGAx_IN+ усиливается с выбранным коэффициентом усиления и виден на контакте PGAx_OUT.

Примечание: во избежание насыщения входное напряжение должно оставаться меньше значения, являющегося частным напряжения VDDA и выбранного усиления.

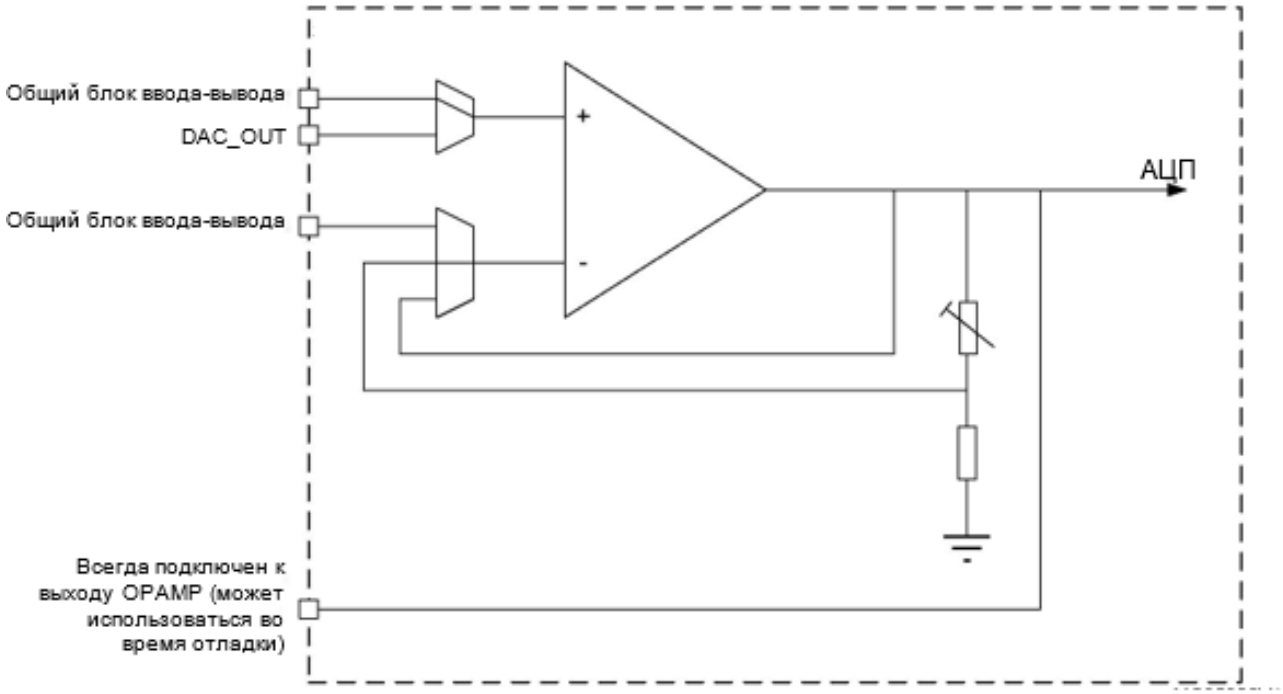


Рисунок 1.182 – Режим PGA, внутреннее усиление (x2/x4/x8/x16), инвертирующий вход не используется

Режим усилителя с программируемым входом и внешней фильтрацией

Порядок использования ОУ для усиления амплитуды входного сигнала с внешней фильтрацией описан ниже (см рисунок 1.183).

– Установите биты PGA_GAIN на «Усиление внутреннего усилителя 2, 4, 8 или 16»,

Любое внешнее соединение на PGAx_IN+ можно использовать параллельно с внутренним усилителем с программируемым усилением, например, конденсатор можно подключить между PGAx_OUT и PGAx_IN- для фильтрации (параметры резисторов, используемых в сети резисторов усилителя с программируемым усилением см. в ТУ).

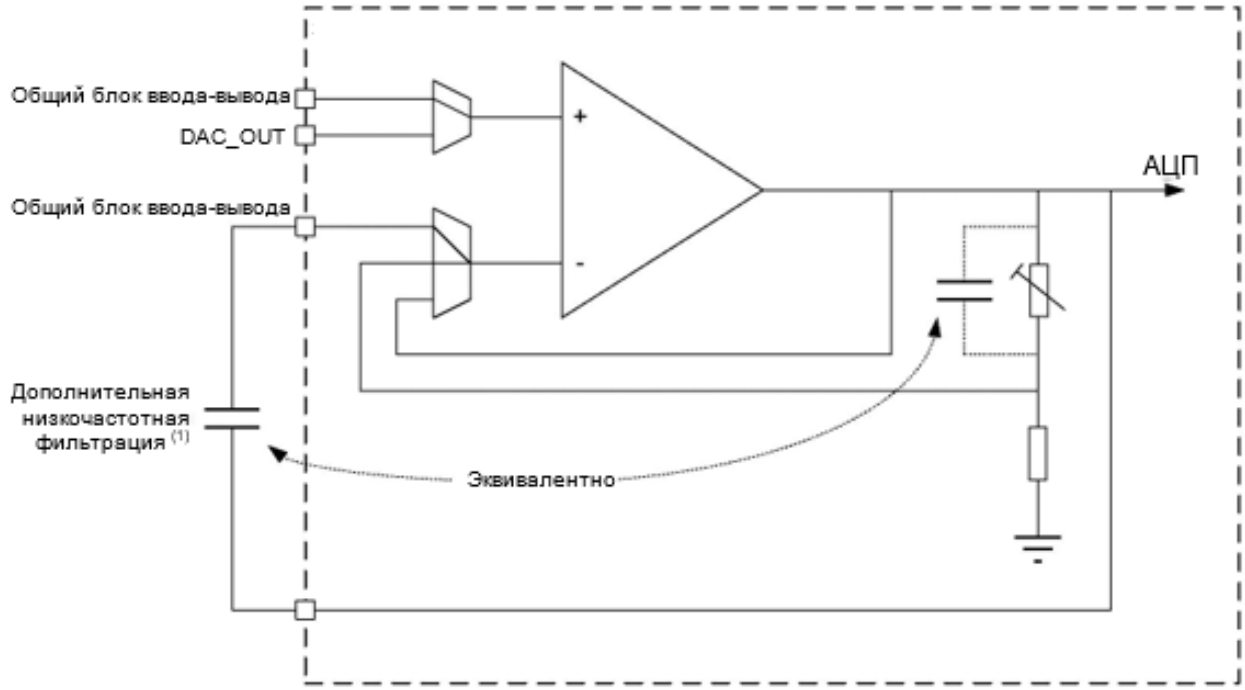


Рисунок 1.183 – Режим PGA, внутреннее усиление (x2/x4/x8/x16), инвертирующий вход используется для фильтрации

1) Усиление зависит от частоты отсечки.

1.3.14.10.3.5 Режимы низкой мощности операционного усилителя

Режимы низкой мощности операционного усилителя приведеы в таблице 1.800.

Таблица 1.800 – Влияние режимов низкой мощности на операционные усилители

Режим	Описание
Спящий режим	Нет действия
Остановка 0 / остановка 1	Нет действия, содержание регистров ОУ сохраняется
Остановка 2	Содержание регистров ОУ сохраняется. Операционный усилитель следует отключить до перехода в режим Остановка 2

1.3.14.10.4 Регистры блоков ОУ

Смещение

0x0

Размер

0x54

1.3.14.10.4.1 Регистровая карта

Смещение	Идентификатор	Название	
0x00	r0_PGA_ENABLE	PGAx_CSR	#r0_PGA_ENABLE
0x04	r0_PGA_SELECT_OPERANDS	PGA_SELECT_OPERANDS	#r0_PGA_SELECT_OPERANDS
0x08	r0_PGA_SELECT_LEVEL	PGA_SELECT_LEVEL	#r0_PGA_SELECT_LEVEL
0x0C	-	-	
0x10	r1_PGA_ENABLE	PGAx_CSR	#r1_PGA_ENABLE
0x14	r1_PGA_SELECT_OPERANDS	PGA_SELECT_OPERANDS	#r1_PGA_SELECT_OPERANDS
0x18	r1_PGA_SELECT_LEVEL	PGA_SELECT_LEVEL	#r1_PGA_SELECT_LEVEL
0x1C	-	-	
0x28	-	-	
0x30	r_TSENS_ENABLE	TSENS_ENABLE	#r_TSENS_ENABLE
0x34	-	-	
0x40	r_VBAT13_ENABLE	VBAT13_ENABLE	#r_VBAT13_ENABLE
0x44	-	-	

1.3.14.10.5 PGAx_CSR

Смещение

0x0

Размер

0x4

1.3.14.10.5.1 Описание

PGAx Control and Status register

1.3.14.10.5.2 Состав

Bits	Identifier	Access	Reset	Decoded	Name	
[31:1]	-	-	-		-	
[0]	EN	rw	0x0	ENABLE DISABLE	EN	H #EN

1.3.14.10.6 PGA_SELECT_OPERANDS

Смещение

0x14

Размер

0x4

1.3.14.10.6.1 Описание

PGA Select Operands

1.3.14.10.6.2 Состав

Bits	Identifier	Access	Reset	Decoded	Name	
[31:8]	-	-	-		-	
[7:4]	SON	rw	0x0	sop03 sop02 sop01 sop00	SON	H #SON
[3:0]	SOP	rw	0x0	sop03 sop02 sop01 sop00	SOP	H #SOP

1.3.14.10.7 PGA_SELECT_LEVEL

Смещение

0x8

Размер

0x4

1.3.14.10.7.1 Состав

PGA Select Level

1.3.14.10.7.2 Contents

Bits	Identifier	Access	Reset	Decoded	Name	
[31:4]	-	-	-		-	
[3:0]	LVL	rw	0x0	level3 level2 level1 level0	LVL	H #LVL

1.3.14.10.8 PGAx_CSR

Смещение

0x10

Размер

0x4

1.3.14.10.8.1 Описание

PGAx Control and Status register

1.3.14.10.8.2 Состав

Bits	Identifier	Access	Reset	Decoded	Name	
[31:1]	-	-	-		-	
[0]	EN	rw	0x0	ENABLE DISABLE	EN	H #EN

1.3.14.10.9 PGA_SELECT_OPERANDS

Смещение

0x4

Размер

0x4

1.3.14.10.9.1 Описание

PGA Select Operands

1.3.14.10.9.2 Состав

Bits	Identifier	Access	Reset	Decoded	Name	
[31:8]	-	-	-		-	
[7:4]	SON	rw	0x0	sop03 sop02 sop01 sop00	SON	H #SON
[3:0]	SOP	rw	0x0	sop03 sop02 sop01 sop00	SOP	H #SOP

1.3.14.10.10 PGA_SELECT_LEVEL

Смещение

0x18

Размер

0x4

1.3.14.10.10.1 Описание

PGA Select Level

1.3.14.10.10.2 Состав

Bits	Identifier	Access	Reset	Decoded	Name	
[31:4]	-	-	-		-	
[3:0]	LVL	rw	0x0	level3 level2 level1 level0	LVL	H #LVL

1.3.14.10.11 TSENS_ENABLE

Смещение

0x30

Размер

0x4

1.3.14.10.11.1 Описание

Thermal Sensor Control register

1.3.14.10.11.2 Состав

Bits	Identifier	Access	Reset	Decoded	Name	
[31:1]	-	-	-		-	
[0]	EN	rw	0x0	ENABLE DISABLE	EN	H #EN

1.3.14.10.12 VBAT13_ENABLE

Смещение

0x40

Размер

0x4

1.3.14.10.12.1 Описание

VBAT measure circuit divider enable

1.3.14.10.12.2 Состав

Bits	Identifier	Access	Reset	Decoded	Name	
[31:1]	-	-	-		-	
[0]	EN	rw	0x0	ENABLE DISABLE	EN	H #EN

В таблице 1.801 приведены справочные характеристики операционного усилителя.

Таблица 1.801 – Характеристики операционного усилителя

Наименование параметра, единица измерения	Буквенное обозначение	Режим измерения	Мин	Типовое	Макс
Напряжение питания аналоговое, В	V_{DDA}	-	1.8	-	3.6
Диапазон входного синфазного сигнала, В	CMIR	-		-	V_{DDA}-0,6

Наименование параметра, единица измерения	Буквенное обозначение	Режим измерения		Мин	Типовое	Макс
Входное напряжение смещения, мВ	VI_{OFFSET}	25 °С, без нагрузки на выходе.		-	-	±1,5
		Все напряжения и температуры		-	-	±3
Дрейф входного напряжения смещения, мкВ/ оС	ΔVI_{OFFSET}	Нормальный режим		-	±5	-
Ток нагрузки, мкА	I_{LOAD}	Нормальный режим	$V_{DDA} \geq 2$ В	-	-	500
Ток нагрузки в режиме PGA, мкА	I_{LOAD_PGA}	Нормальный режим	$V_{DDA} \geq 2$ В	-	-	450
Резистивная нагрузка (подключена к V_{SSA} либо к V~DDA``), кОм	R_{LOAD}	Нормальный режим	$V_{DDA} < 2$ В	4	-	-
Резистивная нагрузка в режиме PGA (подключена к V_{SSA} либо к V_{DDA}), кОм	R_{LOAD_PGA}	Нормальный режим	V~DDA`` < 2 В	4.5	-	-
Емкость нагрузки, пФ	C_{LOAD}	-		-	-	50
Коэффициент подавления синфазного сигнала, дБ	CMRR	Нормальный режим		-	-85	-
Коэффициент подавления помех по питанию, дБ	PSRR	Нормальный режим	$C_{LOAD} \leq 50$ пФ, R~LOAD`` ≥ 4 кОм пост ток	70	85	-
Полоса пропускания усилителя Gain Bandwidth Product, кГц	GBW	Нормальный режим	$V_{DDA} \geq 2{,}4$В (OPA_RANGE = 1)	550	1600	2200
		Нормальный режим	$V_{DDA} < 2{,}4$В (OPA_RANGE = 0)	250	700	950
Скорость нарастания (с 10 до 90% выходного напряжения), В/мс	SR	Нормальный режим	$V_{DDA} \geq 2{,}4$В	-	700	-
		Нормальный режим	$V_{DDA} < 2{,}4$ В	-	300	-
Усиление без замкнутой обратной связи Open loop gain, дБ	AO	Нормальный режим		55	110	-
Высокое напряжение насыщения, мВ	V_{OHSA}	Нормальный режим	I_{load} = max or R_{load} = min Input при V_{DDA}.	V_{DDA} - 100	-	-
Низкое напряжение насыщения, мВ	V_{OLSA}	Нормальный режим	I_{load} = max or R_{load} = min Input при 0.	-	-	100
Запас по фазе Phase margin	φm	Нормальный режим		-	74	-
Запас по усилению Gain margin, дБ	GM	Нормальный режим		-	13	-

Наименование параметра, единица измерения	Буквенное обозначение	Режим измерения		Мин	Типовое	Макс
Время включения из состояния OFF, мкс	t_{WAKEUP}	Нормальный режим	$C_{LOAD} \leq 50$ пФ, $R_{LOAD} \geq 4$ кОм	-	5	10
Входной ток смещения, нА	I_{bias}	Выделенный аналоговый PGA вход	TJ ≤ 75 °C	-	-	1
			TJ ≤ 85 °C	-	-	3
			TJ ≤ 105 °C	-	-	8
			TJ ≤ 125 °C	-	-	15
		Вход общего назначения		-	-	-
Величина усиления без инвертирования	PGA gain	-		-	2	-
				-	4	-
				-	8	-
				-	16	-
R2/R1 сопротивления внутренних резисторов в режиме PGA, кОм/кОм	Rnetwork	PGA Gain = 2		-	80/80	-
		PGA Gain = 4		-	120/40	-
		PGA Gain = 8		-	140/20	-
		PGA Gain = 16		-	150/10	-
PGA ошибка усиления, %	PGA gain error	-		-1	-	1
PGA полоса пропускания для различных величин усиления без инвертирования, МГц	PGA BW	Gain = 2	-	-	GBW/ 2	-
		Gain = 4	-	-	GBW/ 4	-
		Gain = 8	-	-	GBW/ 8	-
		Gain = 16	-	-	GBW/ 16	-
Плотность напряжения шума, нВ/√Гц	en	Нормальный режим	при 1 кГц, нагрузка 4 кОм	-	500	-
		Нормальный режим	при 1 кГц, нагрузка 4 кОм	-	180	-
Ток потребления от источника питания VDDA, мкА	IDDA	Нормальный режим	Без нагрузки, режим покоя	-	120	450

1.3.14.11 Термодатчик (TSENS)

1.3.14.11.1 Блок термодатчика (TSENS)

Датчик температуры можно использовать для измерения температуры перехода (Tj) устройства. Датчик температуры подключен к каналу АЦП, который используются для преобразования выходного напряжения датчика в цифровое значение. Когда датчик не используется, его можно перевести в режим отключения питания. Он поддерживает диапазон температур от –40 до 125 °C.

Выходное напряжение датчика температуры изменяется линейно с температурой. Смещение варьируется от чипа к чипу из-за различий в техпроцессе.

Некалиброванный внутренний датчик температуры больше подходит для приложений, которые обнаруживают колебания температуры, а не абсолютные температуры. Для повышения точности измерения датчика температуры значения калибровки должны сохраняться в памяти системы для каждого устройства во время производства.

Пользовательское приложение затем может считывать калибровочные данные датчика температуры и использовать для повышения точности датчика температуры.

Когда датчик не используется, его можно перевести в режим отключения питания.

Характеристики датчика температуры приведены в таблице 1.802

Таблица 1.802 – Характеристики датчика температуры

Наименование параметра, единица измерения	**Буквенное обозначение**	**Мин**	**Типовое**	**Макс**
Линейность напряжения V_{TS} от температуры, °C	T^L	-	±1	±2
Average slope, мВ/°C	Avg_Slope	2.3	2.5	2.7
Напряжение при 30°C (±5 °C), В	V_{30}	0.742	0.76	0.785
Время старта буфера в непрерывном режиме датчика, мкс	t_{START}	-	8	15
Время старта при переходе в непрерывный режим, мкс	t_{START}	-	70	120
Время выборки АЦП при считывании температуры, мкс	t_{S_temp}	5	-	-
Ток потребления от источника питания V_{DD}, если выбран АЦП, мкА	I_{DD}	-	50	150

1.3.15 Подсистема безопасности

1.3.15.1 Ключевые функциональные характеристики

Система обеспечения безопасности МК Optimal+ обладаетс следующими ключевыми характеристиками:

– изоляция аппаратных ресурсов МК Optimal+ при выполнении защищённого (Secure) и незащищённого (non-secure) ПО при помощи архитектуры TrustZone (см.раздел 1.3.15.2);

– защита областей встроенной флеш-памяти от случайных или несанкционированных преднамеренных операций чтения, записи и стирания(см.раздел 1.3.5.1.2.8);

– возможность задать уникальную точку старта для обеспечения защищённой загрузки (см.раздел 1.3.17);

– генерация истинно-случайных чисел для использования в криптографических операциях (см.раздел 1.3.15.5.1);

– аппаратное ускорение криптографических алгоритмов для шифрования/дешифрования конфиденциальных ПО и данных (см.раздел 1.3.15.3);

– автоматическое стирание конфиденциальных данных при обнаружении попыток несанкционированного воздействия на МК Optimal+ или устройства на его основе (см.раздел 1.3.15.6).

1.3.15.2 Архитектура безопасности TrustZone

Ключевым компонентом системы безопасности МК Optimal+ является архитектура TrustZone, основным назначение которой является защита программных и аппаратных ресурсов, используемых для выполнения защищённого ПО, от несанкционированного доступа со стороны незащищённого ПО.

1.3.15.2.1 Состав и структура архитектуры TrustZone

В состав архитектуры TrustZone МК Optimal+ входят следующие компоненты:

– процессорное ядро CPU с поддержкой расширения системы команд Armv8-M Main Extension и аппаратными блоками IDAU/SAU, обеспечивающими разделение доступной для CPU памяти на защищённые (secure) и незащищённые (non-secure) области;

– шинный коммутатор с поддержкой TrustZone;

– периферийные блоки и блоки памяти с встроенной поддержкой TrustZone (далее по тексту - TZ-защищённые периферийные блоки и блоки памяти), подключаемые напрямую к шинному коммутатору;

– периферийные блоки и блоки памяти без встроенной поддержки TrustZone (далее по тексту TZ-защищаемые периферийный блоки и блоки памяти), подключаемые к шинному коммутатору в сопровождении обеспечивающего их защиту комлекса блоков GTZC, состоящего из:

– контроллеров управления доступом к ресурсам микросхемы со стороны периферийных master-устройств (MSC);

– контроллеров защиты периферийных slave-устройств (PPC_AHB, PPC_APB);

– контроллеров защиты блоков памяти (MPC);

– контроллера безопасности TrustZone (TZSC), управляющий перечисленными выше блоками из состава GTZC;

– контроллера фиксации нарушений защиты TrustZone (TZIC), аккумулирующий события нарушений защиты TrustZone, генерируемые блокам MSC,PPC-AHB, PPC-APB и MPC, и формирующий сигнал прерывания для CPU

Общая структура архитектуры TrustZone МК Optimal+ представлена на рисунке 1.184.

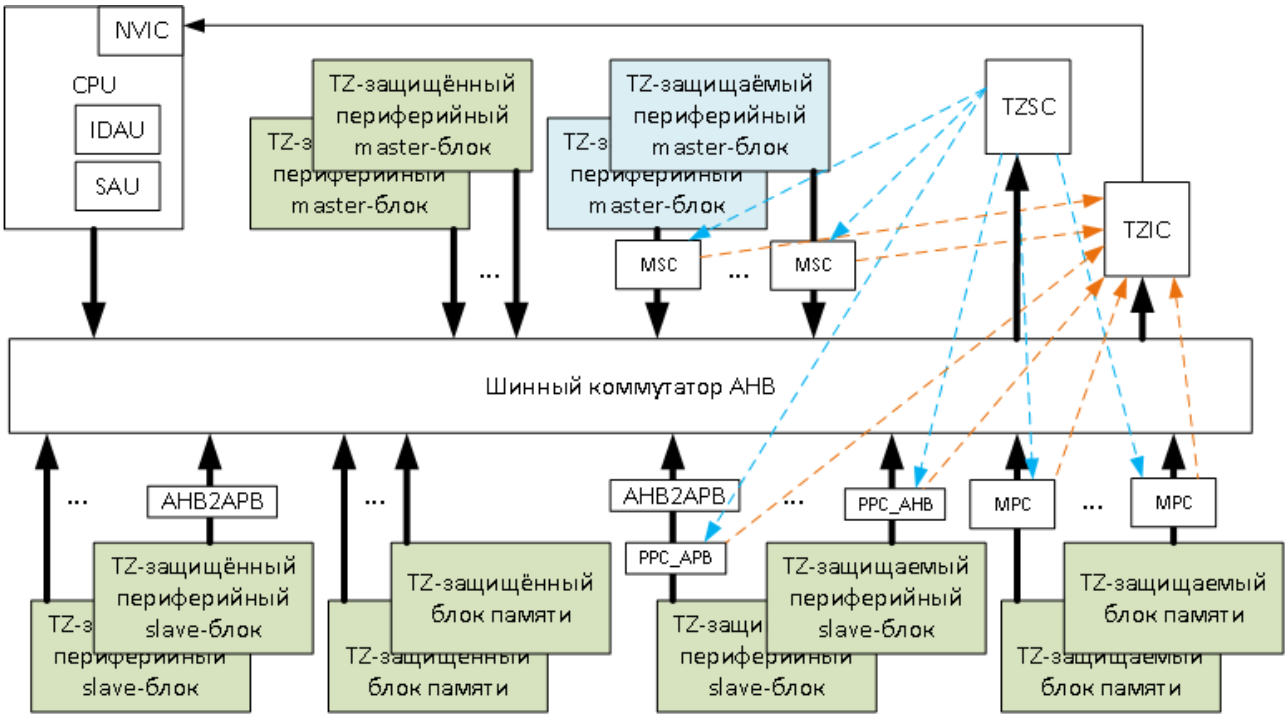


Рисунок 1.184 – Общая структура архитектуры TrustZone МК Optimal+

К ключевым TZ-защищённым периферийным блокам и блокам памяти отсносятся следующие блоки МК Optimal+:

– встроенная флеш-память (FMC);

– встроенное ОЗУ (SRAM0 - SRAM3, RAMCR, BCKSRAM);

– блок управления сигналами сброса и синхросигналами (RCU);

– блок ускорения криптографических и математических операций (CMU);

– системный конфигурационный контроллер (SYSCR);

– комлекс блоков GTZC (TZSC, TZIC);

– подсистема прямого обмена событиями (ELC);

– блок детектирования внешнего воздействия (TAMP);

– контроллеры интерфейсов GPIO (GPIO_A, GPIO_B, GPIO_C, GPIO_D, BKPGPIO);

– блок счётчика реального времени (RTC).

Остальные блоки МК Optimal+ относятся к TZ-защищаемым периферийным блокам и блокам памяти.

1.3.15.2.2 Включение и начальное состояние компонент архитектуры TrustZone

Включение компонетов архитектуры TrustZone происходит после снятия сброса с МК Optimal+ при условии, что поле TZEN, расположенное в энергонезависимой памяти параметров OptMem, было до этого момента установлено в активное состояние "1" (см.разделы 1.3.17 и 1.3.5.1.2.10).

Начальное состояние компонентов, входящих в состав архитектуры TrustZone МК Optimal+, после сброса является следующим:

– начальное разделение памяти на защищённые и незащищённые области жестко задаётся блоком IDAU и соответвует карте памяти, приведённой в таблице 1.12. В дальнейшем, загрузочное ПО может провести изменение разделения памяти при помощи программно доступного блока SAU, который донастройку до 8 областей памяти;

– процессорное ядро CPU старует в защищённом (secure) режиме работы, задаваемый в OPTMEM стартовый адрес обращения CPU (см.разделы 1.3.17 и 1.3.5.1.2.10) должен располагаться в защищённой области памяти;

– весь объём встроенной флеш-память на уровне контроллера FMC настроен как незащищённый, но в дальнейшем, благодаря механизму защиты страниц(см.раздел 1.3.5.1.2.8), может быть гибко поделён загрузочным ПО на защищённые и незащищённые области;

– весь объём внешней памяти, подключаемой через контроллеры интерфейса xSPI (см.раздел 1.3.9.10.2), настроен как незащищённый, но в дальнейшем, через программирование блоков TZSC и MPC (см.раздел 1.3.15.2), может быть гибко поделён загрузочным ПО на защищённые и незащищённые области;

– все периферийные блоки, кроме интерфейсов GPIO (GPIO_A, GPIO_B, GPIO_C, GPIO_D, BKPGPIO), настроены как незащищённые;

– все интерфейсы GPIO (GPIO_A, GPIO_B, GPIO_C, GPIO_D, BKPGPIO) настроены как защищённые.

1.3.15.3 Блок ускорения криптографических и математических операций (CMU)

1.3.15.3.1 Характеристики блока CMU

Блок ускорения криптографических и математических операций (CMU) позволяет ускорить выполнение операций шифрования и дешифрования данных, а также выполнять

логические и арифметические операции над массивами данных. Дополнительно блок CMU обеспечивает работу с обернутыми ключами, т.е. ключами шифрования, предварительно зашифрованными с использованием аппаратного ключа (HUK).

Блок CMU поддерживает выполнение следующих операций:

– выполнение операций шифрования и дешифрования по алгоритмам:

a) AES-128/192/256;

b) "Магма" (ГОСТ 34.12-2018 / RFC 8891);

c) "Кузнечик"(ГОСТ 34.12-2018 / RFC 7801);

– режимы шифрования для симметричных алгоритмов:

a) ECB

b) CBC

c) CTR

d) OFB

– вычисление хэш-функций по алгоритмам:

a) "Стрибог" (ГОСТ 34.11-2018);

b) MD-5;

c) SHA-1:

d) SHA-224;

e) SHA-256;

f) SM3;

– вычисление HMAC

– алгоритм RSA с ключом длиной 1024, 2048, 3072 или 4096 бит;

– генерация 256-битного ключа по протоколу Диффи-Хеллмана;

– перестановка битов и байтов по заданной матрице перестановок;

– операции XOR, OR, AND и инверсия массивов данных.

Структурная схема блока CMU представлена на рисунке 1.185.

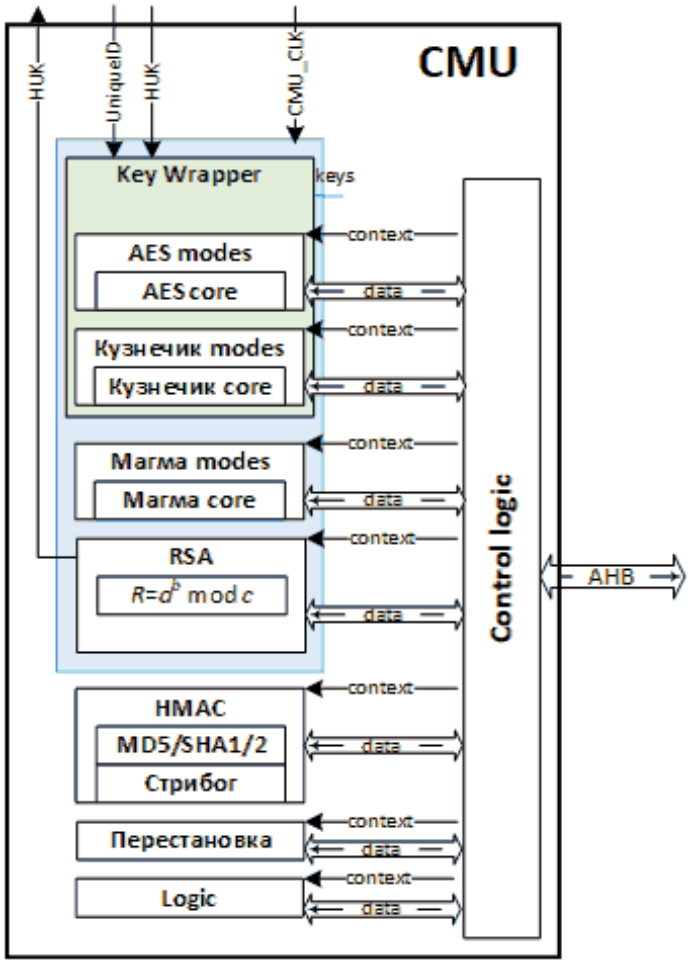


Рисунок 1.185 – Структурная схема блока CMU

1.3.15.3.2 Устройство и функционирование блока CMU

Как для настройки подблоков, реализующих различные алгоритмы в составе CMU, так и для обмена данными используется интерфейс AMBA AHB slave. Каждый из симметричных алгоритмов шифрования в составе CMU поддерживают по 4 набора настроек (контекста), которые могут использоваться в режиме разделения времени.

Для повышения стойкости к атакам по боковым каналам для подблоков шифрования в составе блока CMU предусмотрена возможность работы от внутреннего источника синхросигнала.

Блок CMU аппаратно реализует протокол Диффи-Хеллмана, позволяющий сгенерировать общий для МК Optimal+ и доверенного хоста ключ шифрования, используя открытый канал передачи данных. В дальнейшем этот ключ может использоваться в качестве HUK. Для записи HUK в eFlash-память предусмотрена отдельная шина, доступ к которой невозможен со стороны какого-либо другого устройства, кроме CMU и FMC.

Для выполнения операций асимметричного шифрования предусмотрен встроенный блок памяти размером 4 Кбайт .

Блок CMU выполняет операции обёртывания ключей с помощью HUK для дальнейшего хранения в памяти без ограничений прав доступа. И разворачивания обёрнутых ключей для дальнейшего использования. Для обёртывания ключа применяется либо алгоритм AES-256, либо "Кузнечик". Открытый ключ, который необходимо обернуть, записывается в регистры, доступные только для записи, обёрнутый ключ доступен для чтения.

Все регистры, содержащие ключи и вектора начальной инициализации (в том числе и блок памяти для ключей асимметричного шифрования), доступны только на запись.

1.3.15.3.3 Алгоритм работы с ключевой информацией

1.3.15.3.3.1 Типы ключевой информации

МК Optimal+ предполагает работу со следующими типами ключевой информации (см. таблицу 1.803)

Таблица 1.803 – Типы ключевой информации МК Optimal+

Ключ	Описание	Длина	Доступ	Место хранения
HUK	Hardware Unuque Key Аппаратный ключ	256 бит	CMU RW	Factory Load OTP/ User OTP
WFK	Wrapped Functional Key Обёрнутый ключ для защищённого хранения	зависит от алгоритма	RW	В любом месте в памяти
OK	Open Key Ключ который, хранится в открытом виде	зависит от алгоритма	RW	В любом месте в памяти

За исключением аппаратного ключа (HUK) ключи для шифрования хранятся в любой области памяти либо в открытом виде (OK), если утечка этих ключей не критична с точки зрения информационной безопасности, либо в обёрнутом (зашифрованном) виде (WFK). Хранение ключей в обёрнутом виде позволяет не ограничивать число ключей и не накладывать ограничения на место их хранения, а также упрощает организацию ПЗУ. Для обёртывания ключей используется аппаратный ключ (HUK) уникальный для каждого чипа.

1.3.15.3.3.1.1 Аппаратный ключ (HUK)

HUK является наиболее доверенным ключом МК Optimal+. Данный ключ записывается на фабрике при производстве чипа или же пользователем при производстве готового устройства. Утечка HUK позволит вскрыть все обёрнутые ключи, и как следствие расшифровать все зашифрованные данные.

HUK хранится в специальной области ПЗУ, доступ к которому возможен только блоком CMU по специальной шине. Для записи HUK в CMU предусмотрены специальные регистры, доступ к которым возможен только на запись. Записанный HUK не может быть переписан без полного стирания всей Flash памяти чипа. Программное чтение HUK не доступно, HUK при старте системы переписывается в специальные регистры CMU, которые не отображаются в карту памяти. Таким образом реализуется аппаратная защита от утечки HUK.

Требования к контроллеру памяти для хранения HUK:

– постоянная память, т.к. ключ должен сохраняться после аппаратного сброса;

– отсутствует доступ со стороны коммутационной среды СнК;

– запись доступна только по выделенной шине через CMU;

– чтение доступно только

1.3.15.3.3.1.1.1 Инжекция HUK

HUK может быть записан в чип следующими способами:

1) на фабрике при производстве чипа (одновременно с Unique ID) (рисунок 1.186.1);

2) открытым образом, при этом вопрос безопасности HUK остаётся на том, кто его записывает (рисунок 1.186.2);

3) используя протокол Диффи-Хеллмана (рисунок 1.186.3).

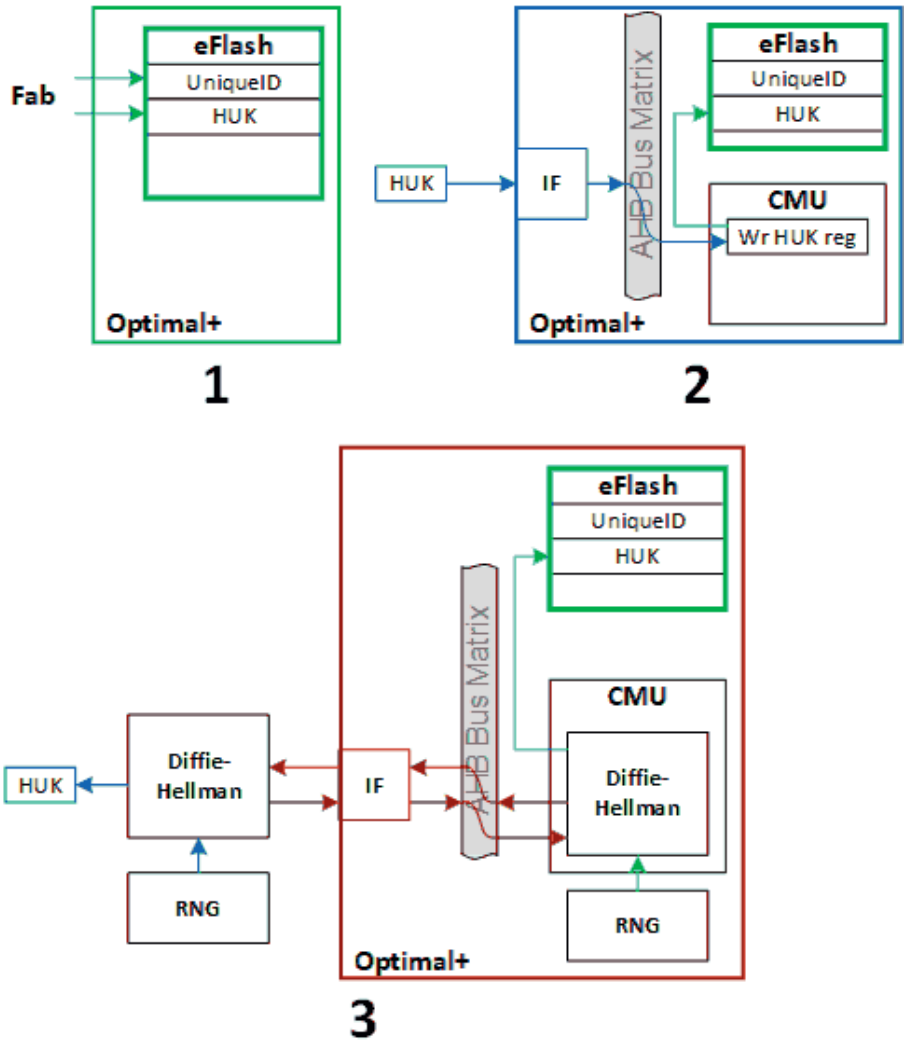


Рисунок 1.186 – Инжекция HUK

Запись HUK открытым образом производится по открытым каналам.

1.3.15.3.3.1.2 Обёрнутые ключи (WFK)

Обёрнутые ключи (WFK) - основные ключи для работы всех алгоритмов шифрования МК Optimal+. За счёт обёртывания WFK может безопасно храниться в любой области памяти. Обернуть можно ключ для любого алгоритма шифрования.

Для обёртывания ключа применяется либо алгоритм AES-256, либо "Кузнечик" (зависит от юрисдикции использования МК Optimal+) в режиме с зацеплением блоков.

Блок CMU может выполнять как дешифрование WFK, так и шифрование (обёртывание) открытого ключа, сделав из него WFK.

1.3.15.3.3.1.2.1 Инжекция WFK

WFK может быть записан в чип следующими способами:

1) обёрнутым на стороне разработчика устройства (рисунок 1.187.1);

2) открытым образом с обёртывание средствами CMU (рисунок 1.187.2).

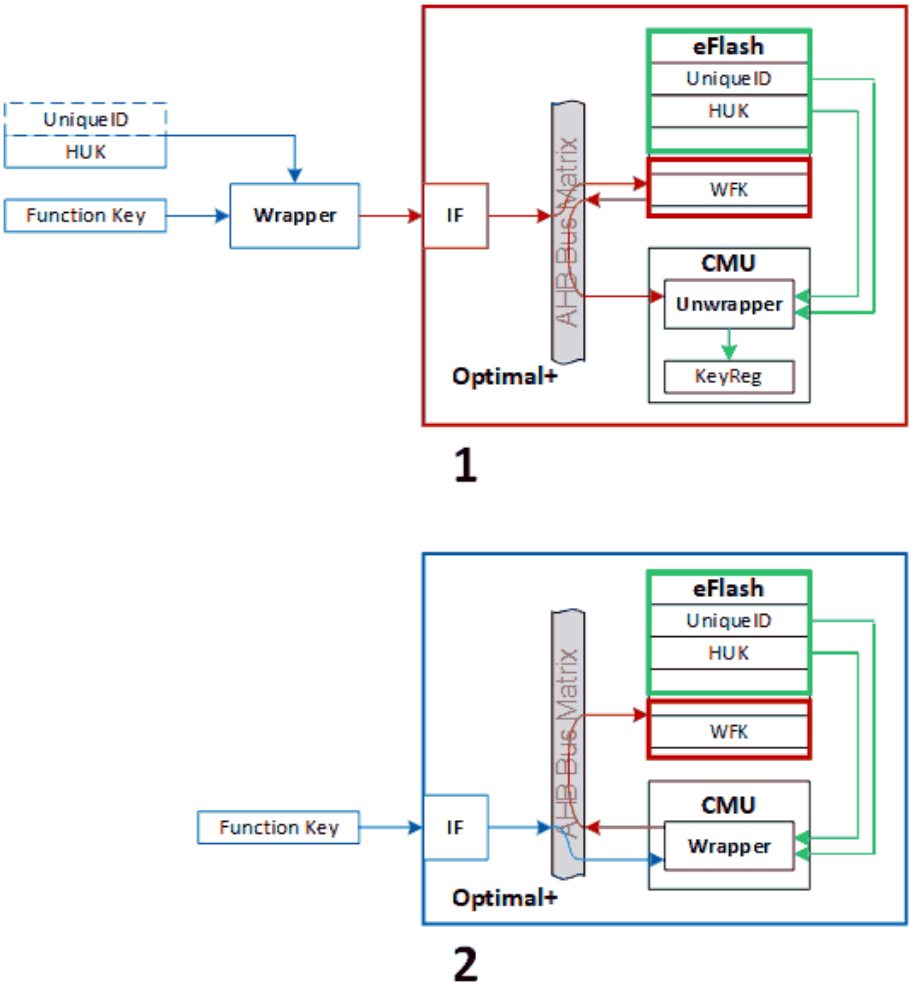


Рисунок 1.187 – Инжекция WFK

Запись открытого ключа в CMU для обёртывания производится по открытым каналам. Запись обёрнутого ключа (как через внешний интерфейс, так и обёрнутого блоком CMU) выполняется программно по открытым каналам.

1.3.15.3.3.1.2.2 Выполнение операций шифрования с использованием WFK

Для выполнения операций шифрования с использованием WFK, его необходимо записать в регистры блока CMU, после чего WFK будет расшифрован и записан во внутренние регистры подблока, выполняющего соответствующий алгоритм.

Расшифрованный WFK программно не доступен, с ним можно только выполнять операции шифрования, но его нельзя прочитать. При событии от блока TAMP расшифрованный WFK стирается.

1.3.15.4 Блок расчета контрольных сумм (CRC)

1.3.15.4.1 Основные характеристики блока расчета контрольных сумм (CRC)

Блок расчета контрольных сумм (CRC) предназначен для вычисления контрольной суммы (CRC) некоторого пакета данных.

Блок расчета контрольных сумм (CRC) обладает следующими характеристиками:

– загрузка данных для расчета CRC:

  – последовательная загрузка данных для расчета CRC;

  – один основной и шестнадцать дополнительных регистров для загрузки данных;

  – возможность загрузки 1-, 2-, 3-, 4- байтных данных для расчета CRC;

  – быстрый расчет CRC в зависимости от размера входных данных:

    – 1 такт для 1 байта;

    – 2 такта для 2 байт;

    – 3 такта для 3 байт;

    – 4 такта для 4 байт;

– настройка полинома:

  – произвольный полином, задаваемый через регистр;

  – выбор степени полинома: CRC-1..CRC-32 (одну из 32 степеней):

– дополнительные настройки для расчета CRC:

  – опция реверса битов входных данных в каждом байте в отдельности;

  – опция реверса всех битов рассчитанного значения CRC;

  – установка начального значения для расчета CRC;

  – установка значения для финальной операции XOR с рассчитанным CRC;

– рассчитанный CRC сохраняется в одном регистре;

– сброс рассчитанного ранее значения CRC производится записью в отдельный регистр;

– дополнительные особенности:

  – поддержка расчета контрольной суммы для NMEA сообщений через CRC-8;

– интерфейс программирования APB 5 slave.

Важно:

– Расчет CRC не является криптостойким алгоритмом и не служит для шифрования данных.

1.3.15.4.2 Описание работы блока расчета контрольных сумм CRC

Блок расчета контрольных сумм CRC предназначен для вычисления контрольной суммы (CRC) пакета данных.

При передаче по проводным линиям связи или радиоканалу возможно искажение передаваемых пакетов данных. Использование CRC позволяется обнаруживать эти ошибки на приемной стороне.

Для расчета CRC используется полином некоторой степени. В общем случае, увеличение степени полинома приводит к повышению помехоустойчивости при передаче данных.

При передаче данных расчет CRC должен проводится два раза: на стороне передатчика и на стороне приемника. Расчет проводится с одинаковым полиномом и одинаковыми настройками расчета (реверса входных и выходных данных, стартовым начальным значение, значением для финальной операции XOR).

Первый расчет CRC происходит на стороне передатчика. Рассчитанное значение CRC прикрепляется к исходным данным и полученный таким образом пакет отправляется приемнику.

Второй расчет CRC происходит на стороне приемника для принятого пакета, за исключением части, где записан CRC. В результате CRC, рассчитанный на приемной стороне, должен совпасть с CRC, который был прикреплен в принятом пакете данных. В этом случае, принятый пакет не был искажен. Если рассчитанные CRC не совпадают, то принятый пакет не является достоверными и был подвергнут негативному воздействию при передаче.

Важно:

– Расчет CRC не является алгоритмом шифрования и служит лишь для проверки целостности пакета данных на приемной стороне. Для криптографического шифрования данных необходимо использовать другие методы и алгоритмы.

1.3.15.4.2.1 Структурная схема блока расчета контрольных сумм CRC

Блок расчета контрольных сумм CRC содержит в себе два функциональных блока.

Первый блок - блок управления (APB-unit-CRC), который обрабатывает сигналы интерфейса APB 5 Slave, хранит настройки блока расчета контрольных сумм CRC и управляет его работой.

Второй блок - блок расчета контрольных сумм CRC (Kernel-CRC), который выдает рассчитывает и выдает контрольную сумму.

Оба блока работают на одной частоте интерфейса АРВ.

Структурная схема блока расчета контрольных сумм CRC приведена на рисунке 1.188.

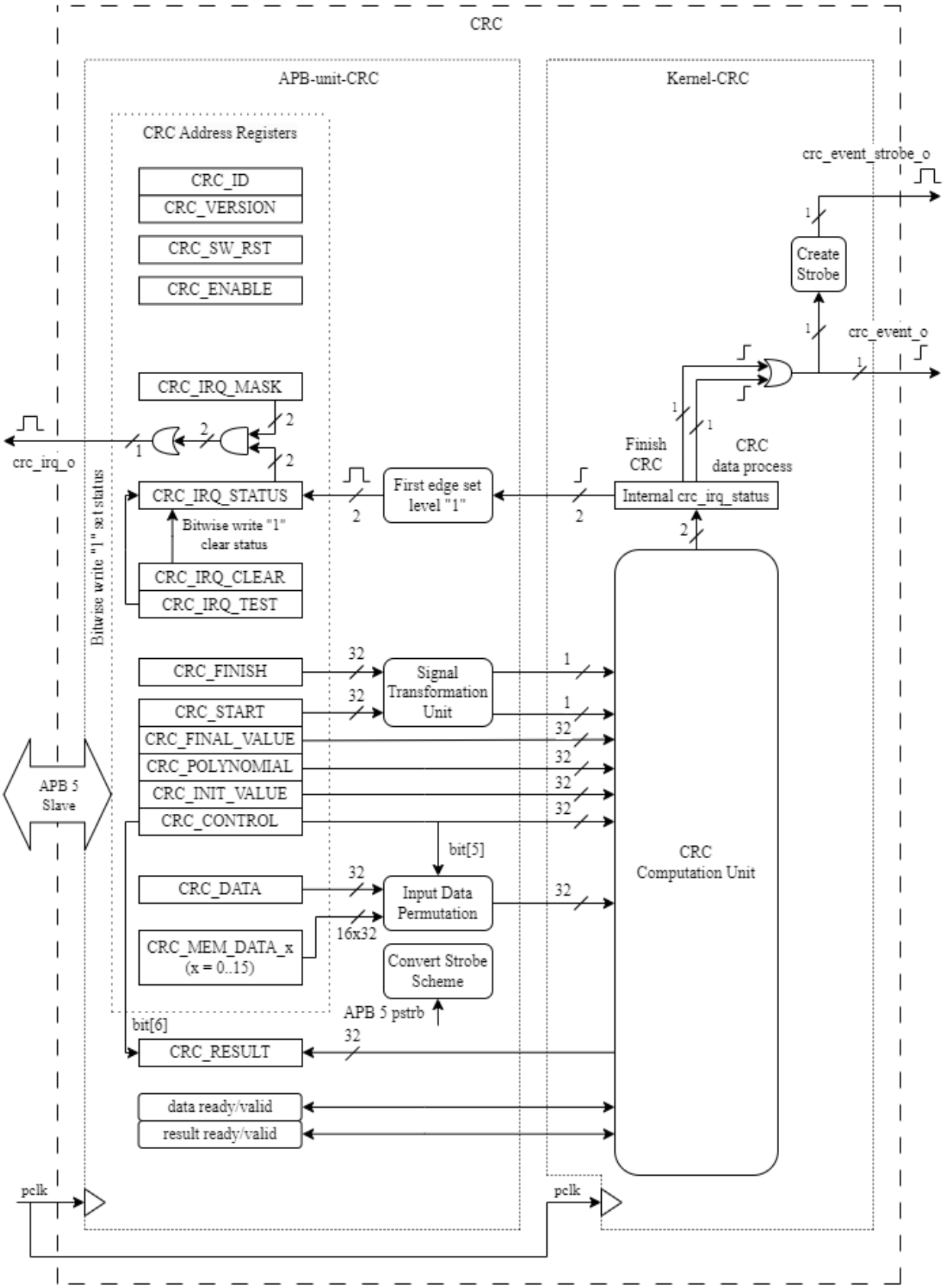


Рисунок 1.188 – Структурная схема блока расчета контрольных сумм CRC

1.3.15.4.2.2 Программный сброс блока расчета контрольных сумм CRC (CRC_SW_RST)

Программный сброс блока расчета контрольных сумм осуществляется путем записи в регистр CRC_SW_RESET значения 0xADDA. Запись любого другого значения не оказывает влияния.

После записи значения 0xADDA в регистр CRC_SW_RESET начинается программный сброс блока расчета контрольных сумм, который длится 1 такт pclk_i.

Во время программного сброса запись во все регистры доступна, но будет игнорироваться.

До завершения программного сброса чтение регистра CRC_SW_RESET возвращает значение 0x1, чтение из других регистров возвращает 0x0. После завершения программного сброса чтение регистра CRC_SW_RESET возвращает значение 0x0.

Регистр CRC_SW_RESET можно использовать для остановки блока расчета контрольных сумм. Значения всех регистров блока расчета контрольных сумм, кроме CRC_SW_RESET, устанавливаются в значение по сбросу.

Регистр CRC_SW_RESET, после выполнения программного сброса, устанавливается в значение по сбросу.

1.3.15.4.2.3 Запуск и остановка блока расчета контрольных сумм (CRC_ENABLE)

Запуск и остановка блока расчета контрольных сумм осуществляется регистром CRC_ENABLE.

Запись значения 0xEAAE в регистр CRC_ENABLE запускает блок расчета контрольных сумм, позволяя инициировать новый расчет CRC. Запись 0xB4 останавливает блок расчета контрольных сумм, сохраняя установленные настройки. Запись любого другого значения не оказывает эффекта.

После записи значения 0xEAAE блок расчета контрольных сумм запускается с параметрами, ранее установленными в регистры CRC_CONTROL, CRC_POLYNOMIAL, CRC_INIT_VALUE_VALUE, CRC_FINAL_VALUE. Возможны несколько последовательных записей значения 0xEAAE, при этом на запуск блок расчета контрольных сумм влияет только первая запись, остальные игнорируются.

После записи значения 0xB4 блок расчета контрольных сумм останавливается и сохраняет параметры, ранее установленные в регистры CRC_CONTROL, CRC_POLYNOMIAL,

CRC_INIT_VALUE_VALUE, CRC_FINAL_VALUE. Значение регистра CRC_RESULT устанавливается в 0x0. Возможны несколько последовательных записей значения 0xB4, при этом на остановку блока расчета контрольных сумм влияет только первая запись, остальные игнорируются.

1.3.15.4.2.4 Входные данные (CRC_DATA)

Блок расчета контрольных сумм CRC поддерживает работу с 1-, 2-, 3-, 4- байтными данными.

Количество байтов входных данных, которое будет использоваться для расчета CRC, определяется стробами (внутренний регистр pstrb_i). Младшие биты этого регистра соответствует младшим используемым байтам CRC_DATA.

Пропуски битов в регистре pstrb_i запрещены, т.е. если, например, pstrb_i = 0x9, то порция данных в регистре CRC_DATA не будет обрабатываться блоком расчета контрольных сумм CRC.

Допустимые значения pstrb_i:

– 0x1;

– 0x3;

– 0x7;

– 0xF;

– 0xE;

– 0xC;

– 0x8.

Таким образом, данные в регистр необходимо записывать в определенные биты:

– 1-байтное слово, pstrb_i == 0x1: CRC_DATA[7:0];

– 2-байтное слово, pstrb_i == 0x3: CRC_DATA[15:0];

– 3-байтное слово, pstrb_i == 0x7: CRC_DATA[23:0];

– 4-байтное слово, pstrb_i == 0xF: CRC_DATA[31:0];

– 3-байтное слово, pstrb_i == 0xE: CRC_DATA[31:23];

– 2-байтное слово, pstrb_i == 0xC: CRC_DATA[31:15];

– 1-байтное слово, pstrb_i == 0x8: CRC_DATA[31:7].

Работа с регистрами CRC_MEM_DATA_x (x == 0..15) соответсвует работе с регистром CRC_DATA.

Время расчета CRC составляет:

– 1 такт для 1 байта;

– 2 такта для 2 байт;

– 3 такта для 3 байт;

– 4 такта для 4 байт.

1.3.15.4.2.5 Настройки для расчета CRC (CRC_CONTROL)

Настройки для расчета CRC определяются регистром CRC_CONTROL.

Поддерживаемые настройки:

– Выбор степени полинома;

– Опция реверса битов входных данных в каждом байте в отдельности;

– Опция реверса всех битов рассчитанного значения CRC.

Выбор степени полинома указывает, где предполагается "явная единица". Например, при использовании CRC-8 "явная единица" предполагается в 8-ом бите. Нумерация битов начинается нуля.

На выбор доступны степени полинома CRC-x, где x находится в диапазоне 1..32:

– CRC-1: CRC_CONTROL[4:0] = 0x01;

– CRC-2: CRC_CONTROL[4:0] = 0x02;

– CRC-3: CRC_CONTROL[4:0] = 0x03;

– …

– CRC-15: CRC_CONTROL[4:0] = 0x0F;

– CRC-16: CRC_CONTROL[4:0] = 0x10;

– CRC-17: CRC_CONTROL[4:0] = 0x11;

– …

– CRC-30: CRC_CONTROL[4:0] = 0x1E;

– CRC-31: CRC_CONTROL[4:0] = 0x1F;

– CRC-32: CRC_CONTROL[4:0] = 0x00.

Реверс всех битов рассчитанного значения CRC (регистр CRC_RESULT) определяется настройкой CRC_CONTROL[6]. Для значения 0x11223344 результат применения настройки будет 0x22CC4488.

Реверс битов входных данных (регистры CRC_DATA, CRC_MEM_DATA_0 - CRC_MEM_DATA_15) в каждом байте в отдельности определяется настройкой CRC_CONTROL[5]. Для значения 0x1A2B3C4D результат применения настройки будет 0x58D43CB2.

1.3.15.4.2.6 Генерирующий полином (CRC_POLYNOMIAL)

Генерирующий полином определяется регистром CRC_POLYNOMIAL и задается без "явной единицы", которая означает степень полинома. Функцию "явной единицы" выполняет настройка CRC_CONTROL[4:0], где выбирается степень полинома.

В зависимости от степени полинома, для расчета CRC используются младшие биты регистра CRC_POLYNOMIAL:

– CRC-x: CRC_POLYNOMIAL[x-1:0], где x находится в диапазоне 1..32.

Неиспользуемые при расчете CRC биты регистра CRC_POLYNOMIAL должны иметь значение 0x0.

1.3.15.4.2.7 Стартовое значение для расчета CRC (CRC_INIT_VALUE)

Инициализирующее значение для начала расчета CRC задается регистром CRC_INIT_VALUE и позволяет реализовывать различные схемы расчета CRC.

В зависимости от степени генерирующего полинома, определяемой битами регистра CRC_CONTROL[4:0], для инициализирующего значение для начала расчета CRC используются младшие биты регистра CRC_INIT_VALUE:

– CRC-x: CRC_INIT_VALUE[x-1:0], где x находится в диапазоне 1..32.

Неиспользуемые при расчете CRC биты регистра CRC_INIT_VALUE должны иметь значение 0x0.

Значение регистра CRC_INIT_VALUE для расчета новой контрольной суммы применяется после записи значения 0x5555 в биты регистра CRC_START[15:0] и не изменяется до этой повторной записи.

Далее в качестве стартового значения при расчете CRC будет использован предыдущий рассчитанный CRC, который не подвержен реверсу, за который отвечает настройка CRC_CONTROL[6]. Это значение сохранено во внутреннем регистре и не доступно для чтения. Таким образом организован расчет CRC при последовательной записи нескольких значений в регистр CRC_DATA или регистры CRC_MEM_DATA_x (x == 0..15).

1.3.15.4.2.8 Значение для финальной операции XOR (CRC_FINAL_VALUE)

Значение для финальной операции XOR определяется регистром CRC_FINAL_VALUE и позволяет реализовывать различные схемы расчета CRC.

В зависимости от степени генерирующего полинома, определяемой битами регистра CRC_CONTROL[4:0], для инициализирующего значение для начала расчета CRC используются младшие биты регистра CRC_FINAL_VALUE:

– CRC-x: CRC_FINAL_VALUE[x-1:0], где x находится в диапазоне 1..32.

Неиспользуемые при расчете CRC биты регистра CRC_FINAL_VALUE должны иметь значение 0x0.

Значение регистра CRC_FINISH_VALUE для расчета новой контрольной суммы применяется после записи значения 0x5555 в биты регистра CRC_START[15:0] и не изменяется до этой повторной записи.

В момент записи значения 0xF1F1 в биты регистра CRC_FINISH[15:0] значение в регистре CRC_FINAL_VALUE используется для финальной операции XOR с рассчитанным значением CRC. Полученное значение сохраняется в регистр CRC_RESULT.

1.3.15.4.2.9 Инициализация нового расчета CRC (CRC_START)

Инициализация нового расчета CRC осуществляется записью значение 0x5555 в биты регистра CRC_START[15:0].

Значения, записанные в регистры CRC_CONTROL, CRC_POLYNOMIAL, CRC_INIT_VALUE_VALUE, CRC_FINAL_VALUE применяются для нового расчета CRC и не изменяются до повторной записи 0x5555 в биты регистра CRC_START[15:0]. Значение регистра CRC_RESULT устанавливается в 0x0.

1.3.15.4.2.10 Запуск финальной операции XOR с рассчитанным результатом CRC (CRC_FINISH)

Запуск финальной операции XOR с рассчитанным результатом CRC осуществляется записью значение 0xF1F1 в биты регистра CRC_FINISH[15:0].

Эта операция может быть выполнена только один раз для одного расчета CRC.

Повторный запуск финальной операции XOR с рассчитанным результатом CRC возможен после:

– После записи значения 0x5555 в биты регистра CRC_START[15:0], если блок расчета контрольных сумм включен (CRC_ENABLE = 0x1);

– После записи значения 0xB4, а затем значения 0xEAAE в биты регистра CRC_ENABLE[15:0] (последовательное отключение и запуск блока расчета контрольных сумм);

– После аппаратного сброса;

– После программного сброса.

1.3.15.4.2.11 Результат расчета CRC (CRC_RESULT)

Результат расчета CRC хранится в регистре CRC_RESULT.

Для чтения достоверного значения CRC необходимо использовать младшие биты регистра:

В зависимости от степени генерирующего полинома, определяемой битами регистра CRC_CONTROL[4:0], для чтения результата расчета CRC используются младшие биты регистра CRC_RESULT:

– CRC-x: CRC_RESULT[x-1:0], где x находится в диапазоне 1..32.

Неиспользуемые при расчете CRC биты регистра CRC_RESULT должны иметь значение 0x0.

После записи значения 0x5555 в биты регистра CRC_START[15:0] или значения 0xB4 в биты регистра CRC_ENABLE[15:0] значение регистра CRC_RESULT == 0x0.

1.3.15.4.2.12 Чтение статусного регистра блока расчета контрольных сумм (CRC_IRQ_STATUS)

Статусный регистр CRC_IRQ_STATUS указывает на события которые произошли в блоке расчета контрольных сумм. Отображение в статусном регистре зависит от настройки регистра маски CRC_IRQ_MASK.

Регистр CRC_IRQ_STATUS доступен только для чтения. Очистка регистра доступна при помощи регистра CRC_IRQ_CLEAR.

1.3.15.4.2.13 Маска событий блока расчета контрольных сумм (CRC_IRQ_MASK)

Регистр маски событий CRC_IRQ_MASK позволяет выбирать, какие события блока расчета контрольных сумм, отображенные в регистре CRC_IRQ_STATUS, будут вырабатывать запросы на прерывание.

1.3.15.4.2.14 Очистка статусного регистра блока расчета контрольных сумм (CRC_IRQ_CLEAR)

Регистр очистки статуса блока расчета контрольных сумм CRC_IRQ_CLEAR позволяет побитно очищать регистр CRC_IRQ_STATUS.

Чтение из регистра CRC_IRQ_CLEAR возвращает 0x0.

1.3.15.4.2.15 Ручная установка статусного регистра блока расчета контрольных сумм (CRC_IRQ_TEST)

Регистр ручной установки статусного регистра блока расчета контрольных сумм CRC_IRQ_TEST позволяет побитно устанавливать "1" в регистр CRC_IRQ_STATUS.

Чтение из регистра CRC_IRQ_TEST возвращает последнее записанное в него значение.

1.3.15.4.2.16 Выходные сигналы блока расчета контрольных сумм

Выходные сигналы блока расчета контрольных сумм образуются путем применения регистра маски CRC_IRQ_MASK на статусный регистр CRC_IRQ_STATUS.

В качестве выходных используется три сигнала:

– crc_event_strobe_o - запросы на прерывание при окончании расчета загруженных входных данных или завершении расчета CRC после финальной операции XOR (строб с уровнем "1");

– crc_event_o - запросы на прерывание при окончании расчета загруженных входных данных или завершении расчета CRC после финальной операции XOR (изменение уровня с "0" на "1" или с "1" на "0");

– crc_irq_o - общий запрос на прерывание при вырабатывании одного из запросов (CRC_IRQ_STATUS & CRC_IRQ_MASK).

1.3.15.4.2.17 Запись значений в регистры CRC_CONTROL, CRC_POLYNOMIAL, CRC_INIT_VALUE_VALUE, CRC_FINAL_VALUE

При корректной записи в регистры CRC_CONTROL, CRC_POLYNOMIAL, CRC_INIT_VALUE_VALUE, CRC_FINAL_VALUE записанные значения сначала сохраняются в теневые регистры, которые недоступны пользователю для чтения.

Загрузка значений из теневых регистров в доступные для чтения регистры CRC_CONTROL, CRC_POLYNOMIAL, CRC_INIT_VALUE_VALUE, CRC_FINAL_VALUE происходит при записи значения 0x5555 в биты регистра CRC_START[15:0], если блок расчета контрольных сумм включен (CRC_ENABLE = 0x1).

Значения теневых регистров и доступных для чтения регистров CRC_CONTROL, CRC_POLYNOMIAL, CRC_INIT_VALUE_VALUE, CRC_FINAL_VALUE устанавливаются в 0x0 в случаях:

– После записи значения 0xEAAE в биты регистра CRC_ENABLE[15:0], если блок расчета контрольных сумм включен (CRC_ENABLE = 0x0)

– После записи значения 0xB4, а затем значения 0xEAAE в биты регистра CRC_ENABLE[15:0] (последовательное отключение и запуск блока расчета контрольных сумм);

– После аппаратного сброса;

– После программного сброса.

1.3.15.4.2.18 Особенности инициализация нового расчета контрольной суммы CRC

При инициализации нового расчета контрольной суммы CRC применяются параметры, записанные в теневые регистры CRC_CONTROL, CRC_POLYNOMIAL, CRC_INIT_VALUE_VALUE, CRC_FINAL_VALUE. Значение регистра CRC_RESULT устанавливается в 0x0.

Инициализация нового расчета контрольной суммы CRC происходит в следующих случаях:

– После записи значения 0x5555 в биты регистра CRC_START[15:0], если блок расчета контрольных сумм включен (CRC_ENABLE = 0x1);

– После записи значения 0xEAAE в биты регистра CRC_ENABLE[15:0], если блок расчета контрольных сумм включен (CRC_ENABLE = 0x0)

– После записи значения 0xB4, а затем значения 0xEAAE в биты регистра CRC_ENABLE[15:0] (последовательное отключение и запуск блока расчета контрольных сумм);

– После аппаратного сброса;

– После программного сброса.

1.3.15.4.2.19 Список распространенных алгоритмов расчета CRC

Список некоторых распространенных алгоритмов расчета CRC и их контрольные значения приведены в таблице 1.804.

В качестве контрольных входных данных используется строка "123456789" в ASCII символах.

Представление контрольной строки в формате HEX: 0x31 0x32 0x33 0x34 0x35 0x36 0x37 0x38 0x39

Таблица 1.804 – Список некоторых распространенных алгоритмов расчета CRC и их контрольные значения

Алгоритм	Полином	Стартовое значение	Реверс входа (CRC_DATA)	Реверс выхода (CRC_RESULT)	Финальное значение XOR	Контрольное значение CRC
CRC-5-USB	0x05	0x1F	Да	Да	0x1F	0x19
CRC-7-MMC	0x09	0x0	Нет	Нет	0x00	0x75
CRC-8-CCITT	0x07	0x0	Нет	Нет	0x00	0xF4
CRC-12-UMTS	0x80F	0x0	Нет	Да	0x000	0xDAF
CRC-15-CAN	0x4599	0x0	Нет	Нет	0x0000	0x059E
CRC-16-CCITT	0x1021	0x0	Нет	Нет	0x0000	0x31C3
CRC-24-Flexray-A	0x5D6DCB	0xFEDCBA	Нет	Нет	0xFFFFFF	0x7979BD
CRC-32-IEEE 802.3	0x04C11DB7	0xFFFFFFFF	Да	Да	0xFFFFFFFF	0xCBF43926

1.3.15.4.2.20 Расчет контрольной суммы для протокола NMEA

NMEA - это общий стандарт представления навигационных данных в текстовом формате (ASCII). Одним из наиболее распространенных случаев является передача навигационным приемником ГНСС-данных в текстовом формате в микроконтроллер через интерфейс UART.

Каждое текстовое сообщение NMEA состоит из информационной части и контрольной суммы (CRC) и в общем имеет вид:

– $[информационная часть]*[контрольная сумма]

Пример NMEA сообщения:

– $GPGGA,004241.47,5532.8492,N,03729.0987,E,1,04,2.0,-0015,M,,,,*31

Для указанной строки контрольная сумма (CRC) информационной части равняется 0x31.

Контрольная сумма является результатом операции XOR с байтами всех ASCII-символов строки, расположенных между '$' и '*', не включая эти символы. Контрольная сумма записывается как число в формате HEX.

Для расчета контрольной суммы NMEA необходимо выставить следующие значения регистров блока расчета контрольных сумм CRC, которые указаны в таблице 1.805.

Таблица 1.805 – Значения регистров блока расчета контрольных сумм CRC для вычисления контрольной суммы NMEA

Регистр	Значение	Комментарий
CRC_POLYNOMIAL	0x1	Используется полином 0x01
CRC_INIT_VALUE	0x0	Стартовое значение равно нулю
CRC_FINAL_VALUE	0x0	Значение для финальной операции XOR равно нулю
CRC_CONTROL	0x8	Установлены настройки: - Используется полином CRC-8; - Стандартный порядок битов рассчитанного значения CRC (нет реверса); - Стандартный порядок битов входных данных (нет реверса).

Работа с входными данными осуществляется также, как описано для регистра CRC_DATA.

Результат контрольной суммы NMEA будет записан в регистр CRC_RESULT[7:0].

Расчет новой контрольной суммы для протокола NMEA инициализируется также, как и при обычном расчете CRC.

1.3.15.4.3 Описание регистров блока расчета контрольных сумм (CRC)

1.3.15.4.3.1 Основные особенности регистров

Установка параметров блока расчета контрольных сумм CRC и отображение его текущего состояния ведется через регистры. Доступ к регистрам осуществляется посредством системного интерфейса управления APB 5 slave. Поддерживается только 32-х разрядный доступ. За исключение регистров CRC_DATA, CRC_MEM_DATA_x [x = 0..15], байтовая, двух-байтовая или трех-байтовая запись будет проигнорирована и возвращена ошибка шины, что вызовет переход процессорного ядра в исключение.

Карта регистров приведена в 1.806.

Все регистры блока CRC обновляются на одной тактовой частоте.

Запись:

Попытка записи проходит успешно и без ошибок, если происходит по адресам регистров CRC, которые указаны в таблице 1.806. При попытке записи по зарезервированным адресам CRC возвращает ошибку 0xE4E4E4E4.

Чтение:

Попытка чтения проходит успешно и без ошибок, если происходит по адресам регистров CRC, которые указаны в таблице 1.806. Чтение разрешено в любой момент времени. При попытке чтения по зарезервированным адресам CRC возвращает ошибку 0xE4E4E4E4.

1.3.15.4.3.2 Карта регистров блока расчета контрольных сумм (CRC)

При дальнейшем описании отдельных полей регистров используется следующее обозначение режимов доступа пользователя к полям:

– RO – поле доступно только для чтения. Запись данных в поле игнорируется;

– WO – поле доступно только для записи. Чтение возвращает значение 0;

– RW – поле доступно как для записи, так и для чтения;

– W1R0 - поле доступно для чтения. Запись 1 сбросит значение, запись 0 не имеет действия.

Таблица 1.806 – Карта регистров блока расчета контрольных сумм (CRC)

Адрес	Название	Доступ	Значение после сброса	Описание
0x000	CRC_ID	RO	0x0	Наименование СФ-блока
0x004	CRC_VERSION	RO	0x0	Используемая версия СФ-блока
0x008	CRC_SW_RST	RW	0x0	Программный сброс СФ-блока
0x00C	CRC_ENABLE	RW	0x0	Включение и отключение блока расчета контрольных сумм
0x010	CRC_DATA	RW	0x0	Основной регистр загрузки входных данных для расчета CRC
0x014	CRC_CONTROL	RW	0x0	Выбор степени полинома, настройка реверса входных данных и выходного результата
0x018	CRC_POLYNOMIAL	RW	0x0	Генерирующий полином для расчета CRC
0x01C	CRC_INIT_VALUE	RW	0x0	Инициализирующее значение для начала расчета CRC
0x020	CRC_FINAL_VALUE	RW	0x0	Значение для финальной операции XOR с рассчитанным результатом CRC
0x024	CRC_START	RW	0x0	Инициализация нового расчета CRC
0x028	CRC_FINISH	RW	0x0	Запуск финальной операции XOR с рассчитанным результатом CRC
0x02C	CRC_RESULT	RO	0x0	Результат расчета CRC
0x030	CRC_IRQ_STATUS	RO	0x0	Статус блока расчета контрольных сумм
0x034	CRC_IRQ_MASK	RW	0x0	Побитовая маска для регистра статуса
0x038	CRC_IRQ_CLEAR	W1R0	0x0	Побитовая очистка регистра статуса
0x03C	CRC_IRQ_TEST	RW	0x0	Побитовая ручная установка "1" в регистр статуса
0x040	CRC_MEM_DATA_0	RW	0x0	Дополнительный регистр загрузки входных данных для расчета CRC (№0)

Адрес	Название	Доступ	Значение после сброса	Описание
0x044	CRC_MEM_DATA_1	RW	0x0	Дополнительный регистр загрузки входных данных для расчета CRC (№1)
0x048	CRC_MEM_DATA_2	RW	0x0	Дополнительный регистр загрузки входных данных для расчета CRC (№2)
0x04C	CRC_MEM_DATA_3	RW	0x0	Дополнительный регистр загрузки входных данных для расчета CRC (№3)
0x050	CRC_MEM_DATA_4	RW	0x0	Дополнительный регистр загрузки входных данных для расчета CRC (№4)
0x054	CRC_MEM_DATA_5	RW	0x0	Дополнительный регистр загрузки входных данных для расчета CRC (№5)
0x058	CRC_MEM_DATA_6	RW	0x0	Дополнительный регистр загрузки входных данных для расчета CRC (№6)
0x05C	CRC_MEM_DATA_7	RW	0x0	Дополнительный регистр загрузки входных данных для расчета CRC (№7)
0x060	CRC_MEM_DATA_8	RW	0x0	Дополнительный регистр загрузки входных данных для расчета CRC (№8)
0x064	CRC_MEM_DATA_9	RW	0x0	Дополнительный регистр загрузки входных данных для расчета CRC (№9)
0x068	CRC_MEM_DATA_10	RW	0x0	Дополнительный регистр загрузки входных данных для расчета CRC (№10)
0x06C	CRC_MEM_DATA_11	RW	0x0	Дополнительный регистр загрузки входных данных для расчета CRC (№11)
0x070	CRC_MEM_DATA_12	RW	0x0	Дополнительный регистр загрузки входных данных для расчета CRC (№12)
0x074	CRC_MEM_DATA_13	RW	0x0	Дополнительный регистр загрузки входных данных для расчета CRC (№13)
0x078	CRC_MEM_DATA_14	RW	0x0	Дополнительный регистр загрузки входных данных для расчета CRC (№14)
0x07C	CRC_MEM_DATA_15	RW	0x0	Дополнительный регистр загрузки входных данных для расчета CRC (№15)

Примечания:

1) Наименование "CRC " в ASCII символах, записанное справа налево.
2) ASCII символы, записанные справа налево. Зависят от параметров, заданных при конфигурации СФ-блока.

1.3.15.4.3.3 Описание полей регистров CRC

1.3.15.4.3.3.1 CRC_ID (0x000) - Наименование СФ-блока

Регистр доступен только для чтения.

Описание полей регистра CRC_ID приведено в таблице 1.807.

Таблица 1.807 – Описание полей регистра CRC_ID

Бит	Поле	Доступ	Значение после сброса	Описание
31:0	id	RO	0x20435243	Наименование СФ-блока ("CRC ", записанное ASCII символами справа налево)

1.3.15.4.3.3.2 CRC_VERSION (0x004) - Версия СФ-Блока

Регистр доступен только для чтения.

Описание полей регистра CRC_VERSION приведено в таблице 1.808.

Таблица 1.808 – Описание полей регистра CRC_VERSION

Бит	Поле	Доступ	Значение после сброса	Описание
31:0	version	RO	0x302E3176	Используемая версия СФ-блока (записана в ASCII символах справа налево)

1.3.15.4.3.3.3 CRC_SW_RST (0x008) - Программный сброс СФ-Блока

Регистр доступен для записи/чтения.

Описание полей регистра CRC_SW_RST во время записи приведено в таблице 1.809.

Таблица 1.809 – Описание полей регистра CRC_SW_RST во время записи

Бит	Поле	Доступ	Значение после сброса	Описание
15:0	sw_rst	WO	0x0	Запись 0xADDA инициирует программный сброс блока расчета контрольных сумм. После выполнения программного сброса устанавливается 0x0. Запись других значений не оказывает эффекта
31:16	-	RO	0x0	Резерв

Описание полей регистра CRC_SW_RST во время записи приведено в таблице 1.810.

Таблица 1.810 – Описание полей регистра CRC_SW_RST во время чтения

Бит	Поле	Доступ	Значение после сброса	Описание
0	sw_rst_status	RO	0x0	"1" - происходит программный сброс блока расчета контрольных сумм[1)] "0" - в данный момент программный сброс блока расчета контрольных сумм не происходит
31:1	-	RO	0x0	Резерв

Примечание:

1) Во время программного сброса чтение из любых других регистров будет возвращать значение 0x0.

1.3.15.4.3.3.4 CRC_ENABLE (0x00C) - Включение и отключение блока расчета контрольных сумм

Регистр доступен для записи/чтения.

Описание полей регистра CRC_ENABLE для записи приведено в таблице 1.811.

Таблица 1.811 – Описание полей регистра CRC_ENABLE во время записи

Бит	Поле	Доступ	Значение после сброса	Описание
15:0	enable	WO	0x0	Запись 0xEAAE запускает блок расчета контрольных сумм. Запись 0xB4 останавливает блок расчета контрольных сумм. Запись любого другого значения не оказывает эффекта
31:16	-	RO	0x0	Резерв

Описание полей регистра CRC_ENABLE для чтения приведено в таблице 1.812.

Таблица 1.812 – Описание полей регистра CRC_ENABLE во время чтения

Бит	Поле	Доступ	Значение после сброса	Описание
0	enable_status	RO	0x0	"1" - блока расчета контрольных сумм запущен "0" - блока расчета контрольных сумм остановлен
31:1	-	RO	0x0	Резерв

1.3.15.4.3.3.5 CRC_DATA (0x010) - Основной регистр загрузки входных данных для расчета CRC

Регистр доступен для записи/чтения.

Описание полей регистра CRC_DATA приведено в таблице 1.813.

Таблица 1.813 – Описание полей регистра CRC_DATA

Бит	Поле	Доступ	Значение после сброса	Описание
31:0	input_data[1)]	RW	0x0	Входные данные для расчета CRC

Примечания:

1) При использовании 1-, 2-, 3-, 4- байтных слов байты должны быть записаны последовательно без пропусков, начиная со старшего или со младшего байта.

1.3.15.4.3.3.6 CRC_CONTROL (0x014) - Выбор степени полинома, настройка реверса входных данных и выходного результата

Регистр доступен для записи/чтения.

Описание полей регистра CRC_CONTROL приведено в таблице 1.814.

Таблица 1.814 – Описание полей регистра CRC_CONTROL

Бит	Поле	Доступ	Значение после сброса	Описание
4:0	polysize	RW	0x0	Выбор степени генерирующего полинома: 0x01 - CRC-1; 0x02 - CRC-2; 0x03 - CRC-3; … 0x1E - CRC-30; 0x1F - CRC-31; 0x00 - CRC-32.
5	reverse_input	RW	0x0	"1" - реверс битов для каждого байта входных данных в отдельности. "0" - стандартный порядок битов входных данных.
6	reverse_output	RW	0x0	"1" - реверс всех битов рассчитанного значения CRC. "0" - стандартный порядок битов рассчитанного значения CRC.
31:7	-	RO	0x0	Резерв

1.3.15.4.3.3.7 CRC_POLYNOMIAL (0x018) - Генерирующий полином для расчета CRC

Регистр доступен для записи/чтения.

Описание полей регистра CRC_POLYNOMIAL приведено в таблице 1.815.

Таблица 1.815 – Описание полей регистра CRC_POLYNOMIAL

Бит	Поле	Доступ	Значение после сброса	Описание
31:0	polynomial[1)]	RW	0x0	Генерирующий полином для расчета CRC

Примечания:

1) В зависимости от степени полинома, определяемой битам CRC_CONTROL[4:0], значение должно быть записано только в младшие биты.

1.3.15.4.3.3.8 CRC_INIT_VALUE (0x01C) - Инициализирующее значение для начала расчета CRC

Регистр доступен для записи/чтения.

Описание полей регистра CRC_INIT_VALUE приведено в таблице 1.816.

Таблица 1.816 – Описание полей регистра CRC_INIT_VALUE

Бит	Поле	Доступ	Значение после сброса	Описание
31:0	init_value[1)]	RW	0x0	Инициализирующее значение для начала расчета CRC

Примечания:

1) В зависимости от степени полинома, определяемой битам CRC_CONTROL[4:0], значение должно быть записано только в младшие биты.

1.3.15.4.3.3.9 CRC_FINAL_VALUE (0x020) - Значение для финальной операции XOR с рассчитанным результатом CRC

Регистр доступен для записи/чтения.

Описание полей регистра CRC_FINAL_VALUE приведено в таблице 1.817.

Таблица 1.817 – Описание полей регистра CRC_FINAL_VALUE

Бит	Поле	Доступ	Значение после сброса	Описание
31:0	final_value[1)]	RW	0x0	Значение для финальной операции XOR с рассчитанным результатом CRC

Примечания:

1) В зависимости от степени полинома, определяемой битам CRC_CONTROL[4:0], значение должно быть записано только в младшие биты.

1.3.15.4.3.3.10 CRC_START (0x024) - Инициализация нового расчета CRC

Регистр доступен для записи/чтения.

Описание полей регистра CRC_START приведено в таблице 1.818.

Таблица 1.818 – Описание полей регистра CRC_START

Бит	Поле	Доступ	Значение после сброса	Описание
15:0	start[1)]	WO	0x0	Запись 0x5555 инициализирует новый расчет CRC. Запись любого другого значения не оказывает эффекта
31:16	-	RO	0x0	Резерв

Примечания:

1) После записи значения 0x5555 значения регистров CRC_INIT_VALUE, CRC_FINAL_VALUE, CRC_CONTROL, CRC_POLYNOMIAL
сохраняют последнее записанное в них значения, которые используются для дальнейшего расчета CRC.

1.3.15.4.3.3.11 CRC_FINISH (0x028) - Запуск финальной операции XOR

Регистр доступен для записи/чтения.

Описание полей регистра CRC_FINISH приведено в таблице 1.819.

Таблица 1.819 – Описание полей регистра CRC_FINISH

Бит	Поле	Доступ	Значение после сброса	Описание
15:0	finish[1)]	WO	0x0	Запись 0xF1F1 запускает финальную операцию XOR с рассчитанным результатом CRC в регистре CRC_RESULT. Запись любого другого значения не оказывает эффекта
31:16	-	RO	0x0	Резерв

Примечания:

1) При расчете CRC учитывается только первая запись значения 0xF1F1. Повторная запись не оказывает эффекта.
Новая запись возможна после записи значения 0x5555 в регистр CRC_START, а также после апаратного и программного сброса.

1.3.15.4.3.3.12 CRC_RESULT (0x02C) - Результат расчета CRC

Регистр доступен только для чтения.

Описание полей регистра CRC_RESULT приведено в таблице 1.820.

Таблица 1.820 – Описание полей регистра CRC_RESULT

Бит	Поле	Доступ	Значение после сброса	Описание
31:0	result[1)]	RO	0x0	Рассчитанное значение CRC

Примечания:

1) Результат хранится в младших битах.

1.3.15.4.3.3.13 CRC_IRQ_STATUS (0x030) - Статус блока расчета контрольных сумм

Регистр доступен только для чтения.

Описание полей регистра CRC_IRQ_STATUS приведено в таблице 1.821.

Таблица 1.821 – Описание полей регистра CRC_IRQ_STATUS

Бит	Поле	Доступ	Значение после сброса	Описание
0	final_crc[1),2)]	RO	0x0	"1" - рассчитано итоговое значение CRC после выполнения финальной операции XOR "0" - итоговое значение CRC не рассчитанно
1	step_crc[1),3)]	RO	0x0	"1" - рассчитано значение CRC после новой загрузки порции 1, 2, 3 или 4 байтных данных "0" - значение CRC после новой загрузки порции 1, 2, 3 или 4 байтных данных не рассчитано
31:2	-	RO	0x0	Резерв

Примечания:

1) Сбросить флаги возможно при выполнении аппаратного сброса системы, программного сброса блока расчета контрольных сумм,
побитовой записью в регистр CRC_IRQ_CLEAR или запись значения 0xB4 в биты регистра CRC_ENABLE[15:0].
Установить флаги вручную можно при помощи регистра CRC_IRQ_TEST.
Установка флагов в регистре CRC_IRQ_STATUS не зависит от регистра CRC_IRQ_MASK.
2) Итоговое значение CRC рассчитывается при записи значения 0xF1F1 в регистр CRC_FINISH
3) Флаг устанавливается после каждой обработки 1, 2, 3 или 4 байтных данных

1.3.15.4.3.3.14CRC_IRQ_MASK (0x034) - Побитовая маска для регистра статуса

Регистр доступен для записи/чтения.

Описание полей регистра CRC_IRQ_MASK приведено в таблице 1.822.

Таблица 1.822 – Описание полей регистра CRC_IRQ_MASK

Бит	Поле	Доступ	Значение после сброса	Описание
0	final_crc_mask	RW	0x0	"1" - разрешить выработку запроса на прерывание[1)] при расчете итогового значения CRC после выполнения финальной операции XOR "0" - запретить выработку запроса на прерывание
1	step_crc_mask	RW	0x0	"1" - разрешить выработку запроса на прерывание[1)] после новой загрузки порции 1, 2, 3 или 4 байтных данных "0" - запретить выработку запроса на прерывание
31:2	-	RO	0x0	Резерв

Примечание:

1) Выходной сигнал запроса на прерывание общий, при выработке одного из этих запросов устанавливается в "1".

1.3.15.4.3.3.15CRC_IRQ_CLEAR (0x038) - Побитовая очистка регистра статуса

Регистр доступен только для записи.

Чтение регистра возвращает 0x0.

Описание полей регистра CRC_IRQ_CLEAR для записи приведено в таблице 1.823.

Таблица 1.823 – Описание полей регистра CRC_IRQ_CLEAR

Бит	Поле	Доступ	Значение после сброса	Описание
0	final_crc_clear	W1R0	0x0	"1" - сбросить флаг регистра статуса CRC_IRQ_STATUS[0] "0" - запись не оказывает эффекта
1	step_crc_clear	W1R0	0x0	"1" - сбросить флаг регистра статуса CRC_IRQ_STATUS[1] "0" - запись не оказывает эффекта
31:2	-	RO	0x0	Резерв

1.3.15.4.3.3.16 CRC_IRQ_TEST (0x03C) - Побитовая ручная установка "1" в регистр статуса

Регистр доступен для записи/чтения.

Описание полей регистра CRC_IRQ_TEST приведено в таблице 1.824.

Таблица 1.824 – Описание полей регистра CRC_IRQ_TEST

Бит	Поле	Доступ	Значение после сброса	Описание
0	final_crc_test[1)]	RW	0x0	"1" - запись устанавливает флаг регистра статуса CRC_IRQ_STATUS[0] "0" - запись не оказывает эффекта
1	step_crc_test[1)]	RW	0x0	"1" - запись устанавливает флаг регистра статуса CRC_IRQ_STATUS[1] "0" - запись не оказывает эффекта
31:2	-	RO	0x0	Резерв

Примечание:

1) Регистр CRC_IRQ_TEST не сбрасывает установленные флаги CRC_IRQ_STATUS. Для этого используется регистр CRC_IRQ_CLEAR.

1.3.15.4.3.3.17 CRC_MEM_DATA_0 - CRC_MEM_DATA_15 (0x040 - 0x07C) - Дополнительный регистр загрузки входных данных для расчета CRC

Регистры доступны для записи/чтения.

Описание полей регистров CRC_MEM_DATA_0 - CRC_MEM_DATA_15 соответствует описанию регистра CRC_DATA и приведено в таблице 1.813.

1.3.15.5 Блок генерации случайных чисел (TRNG)

1.3.15.5.1 Генератор случайных чисел (TRNG)

1.3.15.5.1.1 Генератор случайных чисел обладает следующими характеристиками:

– интерфейс программирования и передачи данных: APB4 slave.

1.3.15.5.1.2 Структурная схема TRNG

На рис. 1.189 изображена структурная схема генератора случайных чисел.

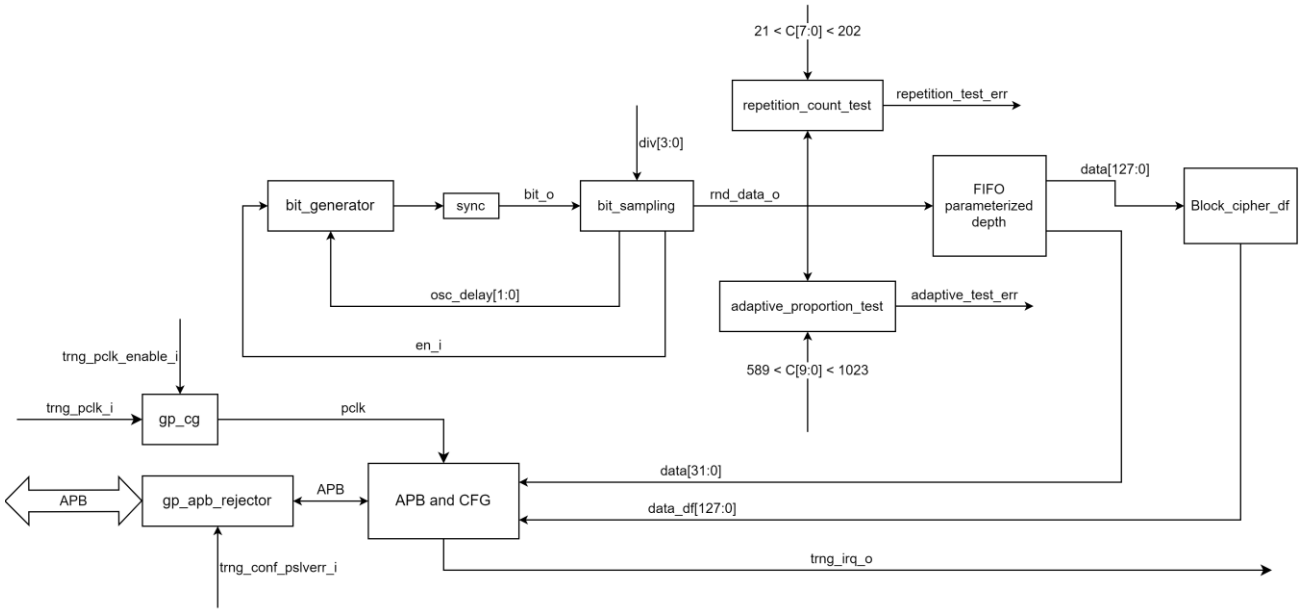


Рисунок 1.189 – Структурная схема TRNG

Таблица 1.825 – Описание входов и выходов.

gp_cg	Блок служит для блокирования подачи тактового сигнала. Для блокировки тактовый сигналов используется вход trng_pclk_enable_i. Если на входе блокировки тактового сигнала единицы, то это тактовый сигнал будет снят. Так же данный блок включает в себя синхронизаторы сброса, они дают задержку на снятие сброс в два такта той частоты, на которой они работают, а сброс происходит асинхронно.
gp_apb_rejector	Блок служит для корректного завершения APB транзакций, когда с APB части USART контроллера снят тактовый сигнал. Данный блок может завершать транзакции с ошибкой (pslverr_o в единице) или без нее. Это задается с помощью входа conf_pslverr_i. Если на этом входе единица, то транзакции будут завершаться с ошибкой, когда на USART не подан тактовый сигнал. При этом транзакции записи игнорируются, а чтение возвращает 0x0BAD_ACCE5.
APB and CFG	Блок служит для включение, настройки и считывания сгенерированных случайных данных из TRNG. Это происходит на частоте trng_pclk_i по интерфейсу APB4.
bit_generator	Блок служит источником энтропии, то есть случайных бит. Подробное описание его работы приведено в пп 1.3.15.5.1.3.
bit_sampling	Данный блок производит выборку случайных бит из bit_generator с заданной частотой сэмплирования.
repetition_count_test	Данный блок производит постоянное тестирование выходных данных из bit_sampling на предмет случайности. Описание данного теста приведено в пп 1.3.15.5.1.4.
adaptive_proportion_test	Данный блок производит постоянное тестирование выходных данных из bit_sampling на предмет случайности в последовательности из 1024 битов данных. Описание данного теста приведено в пп 1.3.15.5.1.4.
FIFO	Данный блок служит для хранения случайных данных, а с помощью системного интерфейса APB происходит их чтение.
Block_cipher_df	Данный блок реализует выходную функцию block_cipher_df из стандарта NIST.SP800-90A. Она необходима для увеличение энтропии случайных чисел.

1.3.15.5.1.3 Генератор случайных бит (bit_generator)

Генерирование случайных битов происходит с помощью схемы представленной на рис. 1.190. В ее основе лежат два кольцевых генератора data_ring_osc и clock_ring_osc, которые управляют входом данных и клока триггера. Они генерируют меандр определенной частоты. В данном случае важно, чтобы частота сигнала data_ring_osc_w была в не сколько раз больше, чем clock_ring_osc_w. Тогда на фронте тактового сигнала вход данных будет произвольным. Если не выдержаться необходимые setup и hold задержки триггера, то он перейдет в метастабильное состояние, выход из которого является случайной величиной.

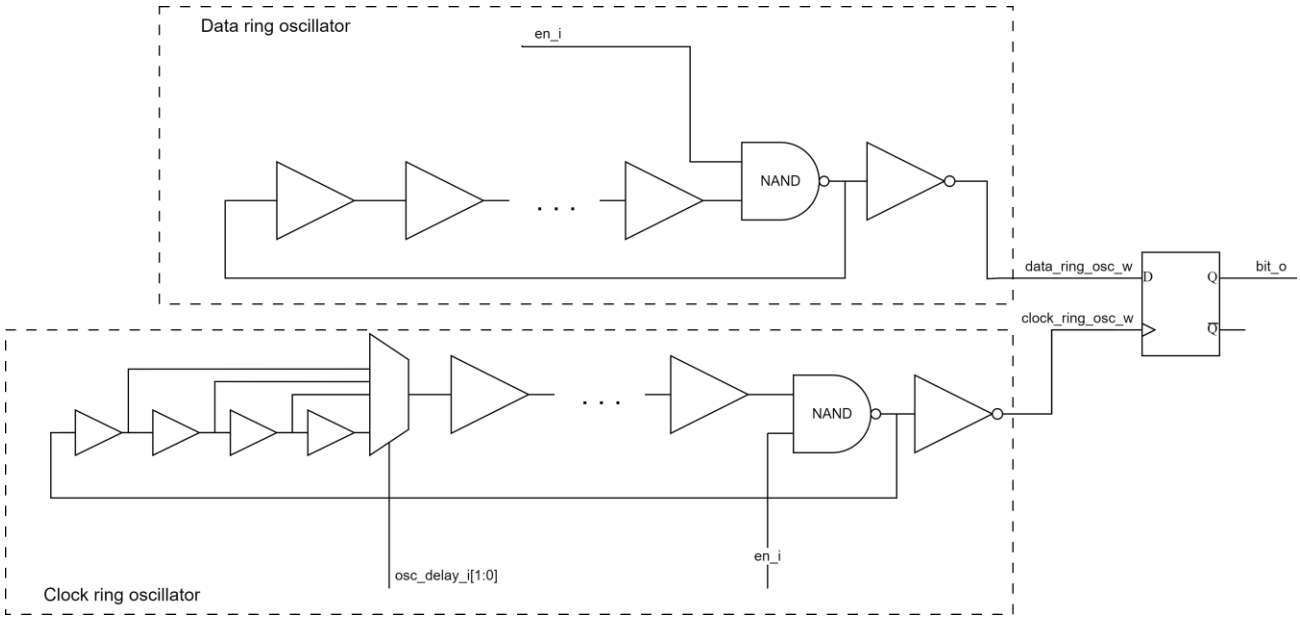


Рисунок 1.190 – Генератор случайных битов

Кольцевой генератор данных выдает сигнал с частотой 390? МГц, а clock_ring_osc с частотой 52? МГц.

Для генерации случайных числе необходимо, чтобы кольцевые генераторы были включены. Это осуществляется с помощью регистра TRNG_CTRL[RING_OSC_EN].

Для кольцевого генератора, который подается на тактовый вход триггера, может быть увеличена задержка путем добавление дополнительных буферов в линию задержки. Это осуществляется с помощью битового поля TRNG_CTRL[OSC_DELAY]. Также есть режим динамического изменения задержки. Этот режим включается путем установки бита TRNG_CTRL[OSC_DELAY_MODE]. При этом поле TRNG_CTRL[OSC_DELAY] перестает быть валидным и не используется.

Частота сэмплирования случайных бит настраивается в поле TRNG_CTRL[DIV]. Для корректного считывания случайных бит необходимо, чтобы частота считывания не превышала частоту генерации (clock_ring_osc_w).

1.3.15.5.1.4 Автоматическое тестирование

Генератор случайных чисел имеет два внутренних блока, которые постоянно проверяют выходные случайные биты в соответствии с NIST.SP800-90B (continuous tests).

Repetition count test отслеживает, чтобы не было нескольких одинаковых чисел подряд. Это максимально допустимое значение rep_cutoff задается в поле TRNG_CTRL[REP_CUTOFF].

Adaptive proportion test отслеживает, чтобы в выборке из 1024 битов количество единиц или нулей не превышало максимально допустимого значения adap_cutoff, которое задается в поле TRNG_CTRL[ADAP_CUTOFF].

После обнаружения ошибок, в регистре TRNG_IRQ_STATUS выставляются соответствующие биты.

Данные тесты можно отключить программно с помощью битов TRNG_TEST_EN[REP_TEST_EN] и TRNG_TEST_EN[ADAP_TEST_EN]. После сброса тесты включены.

1.3.15.5.1.5 Режимы работы

Генератор случайных чисел может работать в двух режимах: FIFO и Block_cipher_df. В первом случае случайные биты записываются в FIFO, а дальше напрямую считываются по системному интерфейсу через регистр TRNG_DATA. Во втором случае в FIFO накапливается 128 случайных бит, затем они проходят через блок Block_cipher_df, который выдает также 128 бит, но с увеличенной энтропией.

Для генерации случайных чисел в любом из режимов должен быть предварительно включены кольцевые генераторы (TRNG_CTRL[RING_OSC_EN]).

Режим работы выбирается с помощью TRNG_CTRL[MODE]. При переходе из одного режима в другой FIFO очищается (сбрасываются счетчики чтения и записи).

1.3.15.5.1.5.1 Режим FIFO

В данном режиме FIFO работает в 32 битном режиме. Это означает, что выходная разрядность данных равна 32. При этом флаг empty сброшен, только когда в FIFO находится хотя бы одно 32-х битное слово. Глубина FIFO составляет FIFO_DEPTH/32 32-х битных слова.

Для генерации случайных бит, а точнее для считывания их блоком bit_sampling и записи в FIFO, необходимо установить запрос на заполнение FIFO. Это осуществляется с помощью регистра TRNG_GEN_REQ. Есть три возможных запроса на заполнение FIFO:

– ONE. Запрос на запись в FIFO нового 32-х битного случайного числа. После того, как будет записано новое случайное число в FIFO, регистр TRNG_GEN_REQ аппаратно сбросится в состояние IDLE. Если FIFO уже полное, то запрос сразу же сбросится, а FIFO ничего не запишется.

– FULL. Запрос заполнение полного FIFO. После установления данного запроса, FIFO начинает заполнятся до максимума, а затем регистр TRNG_GEN_REQ аппаратно сбросится в состояние IDLE. Если FIFO уже полное, то запрос сразу же сбросится, а FIFO ничего не запишется.

– INF. Запрос на постоянное заполнение FIFO. FIFO постоянно заполняется до максимума. Сброс этого запроса происходит программно путем записи в регистр TRNG_GEN_REQ 0x0.

Заполненность FIFO отслеживается с помощь флагов full и empty в регистре TRNG_STATUS. В этом же регистре хранится текущий уровень заполненности.

Чтение из FIFO происходит с помощью регистра TRNG_DATA.

1.3.15.5.1.5.2 Режим Block_cipher_dt

В данном режиме FIFO выступает роли накопителя данных для дальнейшей обработки в блоке Block_cipher_df. Так как этот блок работает со 128 битными данными, то и FIFO начинает работать в 128 битном режиме. Этот режим поддерживает только один запрос ONE. После его установки в регистре TRNG_GEN_REQ, в FIFO начинают записываться случайные биты, пока не будет набрано 128 бит. Затем это число обрабатывается согласно выходной функции Block_Cipher_df из стандарта NIST.SP800-90A. В ее основе лежит алгоритм шифрования AES-128. Он применяется последовательно три раза (см. рис. 1.191). По завершению применения функции регистр запроса TRNG_GEN_REQ сбросится в NONE и выходное значение может быть прочитано путем четырех последовательных чтений из регистра TRNG_DATA.

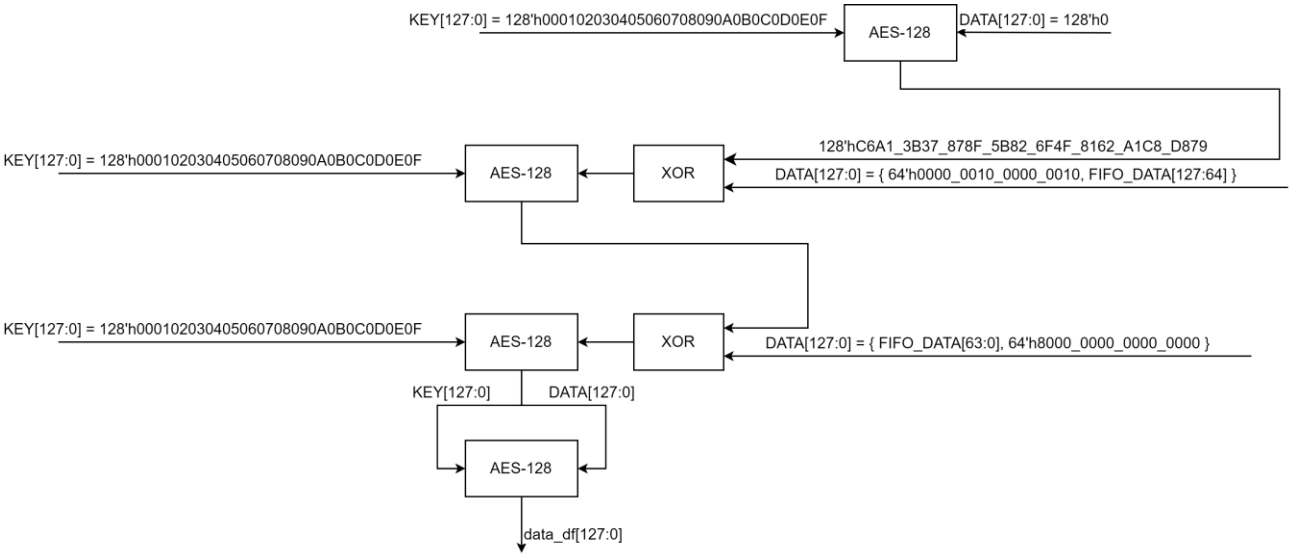


Рисунок 1.191 – Функция Block_Cipher_df

Остальные два запроса FULL и INF являются резервными.

В этом режиме флаг empty сбрасывается на один такт, только когда в буфере поместится 128-битное число. После этого Block_cipher_df сразу считает его и флаг снова устанавливается. Флаг full никогда не устанавливается в этом режиме.

1.3.15.5.1.6 Прерывания

В TRNG генерируется уровневое прерывание, представляющее собой объединение по ИЛИ событий прерываний, перечисленных в регистре TRNG_IRQ_STATUS. Если на возникшее событие прерывания регистра TRNG_IRQ_STATUS установлена маска в регистре TRNG_IRQ_MASK (бит маски соответствующего прерывания равен 0), то прерывание по данному событию генерироваться не будет. После сброса все прерывания замаскированы.

Сброс событий прерываний TRNG производится побитово путем записи единиц в соответствующие биты регистра TRNG_IRQ_CLEAR.

Тестовое прерывание TEST устанавливается после записи единицы в регистр TRNG_IRQ_TEST, а сбрасывается так же, как и все остальные через TRNG_IRQ_CLEAR.

Описание прерываний приведено в таблице 1.3.15.5.1.7.3.8.

1.3.15.5.1.7 Описание регистров

1.3.15.5.1.7.1 Общий список регистров

Контроллер имеет набор программно доступных регистров, требующий 4 КБайт адресов на системной шине APB.

Карта регистров приведена в таблице 1.826.

Адреса, который не указаны в таблице 1.826 считаются резервными. Чтение из них возвращает 0xEEFFADDC, а запись игнорируется. TRGN контроллер не различает secure и non-secure транзакции.

1.3.15.5.1.7.2 Карта регистров

Таблица 1.826 – Карта регистров TRNG

Адрес	Название	Доступ	Значение после сброса	Описание
0x000	TRNG_ID	ЧТ	0x474E5254	Идентификатор устройства
0x004	TRNG_VERSION	ЧТ	0x0	Номер версии устройства и конфигурация
0x008	TRNG_SW_RST	ЧТ/ЗП	0x0	Программный сброс
0x00C	TRNG_CTRL	ЧТ/ЗП	0x000BB088	Настройка
0x010	TRNG_TEST_EN	ЧТ/ЗП	0x0	Включение автоматических тестов
0x014	TRNG_GEN	ЧТ/ЗП	0x0	Запрос на генерирование случайного числа
0x018	TRNG_STATUS	ЧТ	0x2	Статус работы генератора случайных чисел
0x01C	TRNG_IRQ_STATUS	ЧТ	0x0	Статус прерываний
0x020	TRNG_IRQ_MASK	ЧТ/ЗП	0x0	Маска прерываний
0x024	TRNG_IRQ_CLEAR	ЗП	0x0	Побитовый сброс прерываний
0x028	TRNG_IRQ_TEST	ЗП	0x0	Установка тестового прерывания
0x030	TRNG_DATA	ЧТ	0x0	Сгенерированные случайные данные

1.3.15.5.1.7.3 Описание полей регистров

1.3.15.5.1.7.3.1 TRNG_ID (0x000)

Регистр идентификатора TRNG контроллера

В таблице 1.827 представлено описание полей регистра.

Таблица 1.827 – Описание полей регистра TRNG_ID

Биты	Поле	Доступ	Значение после сброса	Описание
31:0	ID	ЧТ	0x474E5254	Идентификатор контроллера TRNG

1.3.15.5.1.7.3.2 TRNG_VERSION (0x004)

Регистр версии TRNG контроллера

В таблице 1.828 представлено описание полей регистра.

Таблица 1.828 – Описание полей регистра TRNG_VERSION

Биты	Поле	Доступ	Значение после сброса	Описание
31:0	VERION	ЧТ	0x0	Версия контроллера TRNG

1.3.15.5.1.7.3.3 TRNG_SW_RST (0x008)

Регистр программного сброса TRNG контроллера

В таблице 1.829 представлено описание полей регистра.

Таблица 1.829 – Описание полей регистра TRNG_SW_RST

Биты	Поле	Доступ	Значение после сброса	Описание
0	SWRST	ЧТ/ЗП	0	Программный сброс. 0 - программный сброс завершен. 1 - установка программного сброса. Сброс данного бита происходит автоматически по завершению программного сброса.
31:1	-	ЧТ	0	Зарезервировано

1.3.15.5.1.7.3.4 TRNG_CTRL (0x00C)

Регистр настройки TRNG контроллера

В таблице 1.830 представлено описание полей регистра.

Таблица 1.830 – Описание полей регистра TRNG_CTRL

Биты	Поле	Доступ	Значение после сброса	Описание
0	RING_OSC_EN	ЧТ/ЗП	0	Включение кольцевых генераторов. 0 - кольцевые генераторы выключены. 1 - кольцевые генераторы включены.
1	MODE	ЧТ/ЗП	0	Режим работы генератора случайных чисел. 0 - чтение случайных данных производится из FIFO. 1 - включена дополнительная обработка (block_cipher_dt), генерируются 128-ми разрядные случайные числа.
9:2	REP_CUTOFF	ЧТ/ЗП	34	Cutoff значение для теста repetition_count. Допустимые значения: от 22 по 201.
19:10	ADAP_CUTOFF	ЧТ/ЗП	748	Cutoff значение для теста adaptive_proportion. Допустимые значения: от 589 по 1022.

Биты	Поле	Доступ	Значение после сброса	Описание
23:20	DIV	ЧТ/ЗП	0	Делитель частоты выборки случайных битов из генераторов битов. 0x0 - выборка происходит каждый такт системной частоты trng_pclk_i. 0x1 - выборка происходит каждые два такта системной частоты trng_pclk_i. 0x2 - выборка происходит каждые три такта системной частоты trng_pclk_i. 0x3 - выборка происходит каждые четыре такта системной частоты trng_pclk_i. и т.д.
25:24	OSC_DELAY	ЧТ/ЗП	0	Дополнительная задержка в кольцевом генераторе, который выступает в качестве тактового сигнала. 00 - нет дополнительной задержки. 01 - один дополнительный буфер задержки. 10 - два дополнительных буфера задержки. 11 - три дополнительных буфера задержки.
26	OSC_DELAY_MODE	ЧТ/ЗП	0	Выбор режима добавления дополнительно задержки кольцевой генератор clock_ring_osc. 0 - дополнительная задержка определятся программно с помощью поля OCS_DELAY. 1 - аппаратное динамическое изменение задержки.
31:27	-	ЧТ	0	Зарезервировано

1.3.15.5.1.7.3.5 TRNG_TEST_EN (0x010)

Регистр выключения тестов TRNG контроллера

В таблице 1.831 представлено описание полей регистра.

Таблица 1.831 – Описание полей регистра TRNG_TEST_EN

Биты	Поле	Доступ	Значение после сброса	Описание
0	REP_TEST_EN	ЧТ/ЗП	0	Выключение автоматического теста repetition_count. 0 - автоматическое тестирование включено. 1 - автоматическое тестирование выключено.
1	ADAP_TEST_EN	ЧТ/ЗП	0	Выключение автоматического теста adaption_proportion. 0 - автоматическое тестирование включено. 1 - автоматическое тестирование выключено.
31:2	-	ЧТ	0	Зарезервировано

1.3.15.5.1.7.3.6 TRNG_GEN (0x014)

Регистр запрос на генерацию случайного числа TRNG контроллера

В таблице 1.832 представлено описание полей регистра.

Таблица 1.832 – Описание полей регистра TRNG_GEN

Биты	Поле	Доступ	Значение после сброса	Описание
1:0	GEN_TYPE	ЧТ/ЗП	0	Установка запроса на генерирование случайных чисел. После окончания выполнения запросов ONE или MAX, данное поле автоматически сбрасывается в NONE. Программная запись NONE запроса, сбрасывает предыдущий не дожидаясь его окончания. 00 - NONE. Нет запроса на генерацию случайного числа. 01 - ONE. Установлен запрос на генерацию одного случайного числа. В зависимости от бита TRNG_CTRL[MODE] это будет или 32-х, или 128-ми битное число. 10 - MAX. Заполнение FIFO до максимума и остановка. В режиме Block_cipher_df данное состояние является резервным. 11 - INF. постоянное заполнение FIFO случайными числами. В режиме Block_cipher_df данное состояние является резервным
31:2	-	ЧТ	0	Зарезервировано

1.3.15.5.1.7.3.7 TRNG_STATUS (0x018)

Регистр статуса TRNG контроллера

В таблице 1.833 представлено описание полей регистра.

Таблица 1.833 – Описание полей регистра TRNG_STATUS

Биты	Поле	Доступ	Значение после сброса	Описание
0	FULL	ЧТ	0	Отображает статус заполненности FIFO. 0 - FIFO не полное. 1 - FIFO полное.
1	EMPTY	ЧТ	1	Отображает статус пустоты FIFO.
3:2	-	ЧТ	0	Зарезервировано
9:4	OCCUPANCY	ЧТ	0	Отображает сколько 32-х или 128-х битных слов находится в FIFO. Разрядность данных зависит от режима работы TRNG_CTLR[MODE]. 0 - FIFO не пустое. 1 - FIFO пустое.
31:10	-	ЧТ	0	Зарезервировано

1.3.15.5.1.7.3.8 TRNG_IRQ_STATUS (0x01C)

Регистр статуса прерываний TRNG контроллера

В таблице 1.834 представлено описание полей регистра.

Таблица 1.834 – Описание полей регистра TRNG_IRQ_STATUS

Биты	Поле	Доступ	Значение после сброса	Описание
0	TEST	ЧТ	0	Программная установка прерывания. 0 - программное прерывание не установлено. 1 - программное прерывание установлено.
1	FULL	ЧТ	0	Отображает статус заполненности FIFO. 0 - FIFO не полное. 1 - FIFO полное.
2	NOT_EMPTY	ЧТ	0	Отображает статус не пустоты FIFO. 0 - FIFO пустое. 1 - FIFO не пустое.
3	ADAP_ERROR	ЧТ	0	Генерируется когда обнаруживается ошибка в adaptive_proportion_test. 0 - ошибка не обнаружена. 1 - ошибка обнаружена.
4	REP_ERROR	ЧТ	0	Генерируется когда обнаруживается ошибка в repetition_count_test. 0 - ошибка не обнаружена. 1 - ошибка обнаружена.
5	GEN_REQ_FINISH	ЧТ	0	Устанавливается, когда завершается выполнение запроса генерации случайного числа. 0 - запрос отсутствует или еще не выполнен. 1 - запрос выполнен.
31:6	-	ЧТ	0	Зарезервировано

1.3.15.5.1.7.3.9 TRNG_IRQ_MASK (0x020)

Регистр маска прерываний TRNG контроллера

В таблице 1.835 представлено описание полей регистра.

Таблица 1.835 – Описание полей регистра TRNG_IRQ_MASK

Биты	Поле	Доступ	Значение после сброса	Описание
0	TEST	ЧТ/ЗП	0	Маска прерывания TEST. 0 - прерывание TEST запрещено. 1 - прерывания TEST разрешено.
1	FULL	ЧТ/ЗП	0	Маска прерывания FULL. 0 - прерывание FULL запрещено. 1 - прерывания FULL разрешено.
2	NOT_EMPTY	ЧТ/ЗП	0	Маска прерывания NOT_EMPTY. 0 - прерывание NOT_EMPTY запрещено. 1 - прерывания NOT_EMPTY разрешено.
3	ADAP_ERROR	ЧТ/ЗП	0	Маска прерывания ADAP_ERROR. 0 - прерывание ADAP_ERROR запрещено. 1 - прерывания ADAP_ERROR разрешено.
4	REP_ERROR	ЧТ/ЗП	0	Маска прерывания REP_ERROR. 0 - прерывание REP_ERROR запрещено. 1 - прерывания REP_ERROR разрешено.

Биты	Поле	Доступ	Значение после сброса	Описание
5	GEN_REQ_FINISH	ЧТ/ЗП	0	Маска прерывания GEN_REQ_FINISH. 0 - прерывание GEN_REQ_FINISH запрещено. 1 - прерывания GEN_REQ_FINISH разрешено.
31:6	-	ЧТ	0	Зарезервировано

1.3.15.5.1.7.3.10 TRNG_IRQ_CLEAR (0x024)

Регистр сброса прерываний TRNG контроллера

В таблице 1.836 представлено описание полей регистра.

Таблица 1.836 – Описание полей регистра TRNG_IRQ_CLEAR

Биты	Поле	Доступ	Значение после сброса	Описание
0	TEST	ЗП	0	Запись единицы в данное поле сбрасывает прерывание TEST.
1	FULL	ЗП	0	Запись единицы в данное поле сбрасывает прерывание FULL.
2	NOT_EMPTY	ЗП	0	Запись единицы в данное поле сбрасывает прерывание NOT_EMPTY.
3	ADAP_ERROR	ЗП	0	Запись единицы в данное поле сбрасывает прерывание ADAP_ERROR.
4	REP_ERROR	ЗП	0	Запись единицы в данное поле сбрасывает прерывание REP_ERROR.
5	GEN_REQ_FINISH	ЗП	0	Запись единицы в данное поле сбрасывает прерывание GEN_REQ_FINISH.
31:6	-	ЧТ	0	Зарезервировано

1.3.15.5.1.7.3.11 TRNG_IRQ_TEST (0x028)

Регистр установки тестового прерывания TRNG контроллера

В таблице 1.837 представлено описание полей регистра.

Таблица 1.837 – Описание полей регистра TRNG_IRQ_TEST

Биты	Поле	Доступ	Значение после сброса	Описание
0	TEST	ЗП	0	Запись единицы в данный бит устанавливает прерывание TEST.
31:1	-	ЧТ	0	Зарезервировано

1.3.15.5.1.7.3.12 TRNG_DATA (0x030)

Регистр чтения сгенерированных случайных данных TRNG контроллера

В таблице 1.838 представлено описание полей регистра.

Таблица 1.838 – Описание полей регистра TRNG_DATA

Биты	Поле	Доступ	Значение после сброса	Описание
31:0	DATA	ЧТ	0	Сгенерированные случайные данные

1.3.15.6 Блок детектирования внешнего воздействия (TAMP)

1.3.15.6.1 Характеристики TAMP

– Стирание данных в backup registers, BSRAM, SRAM2 и блоках, связанных с криптографией, при попытке вскрытия корпуса (tamp_event).

– X внешних входов.

– Y внутренних входов.

– Z выходов для генерации сигналов стирания и блокировки доступа.

– Два режима работы для каждого из входов :

  – Hardware. Стирание происходит аппаратно.

  – Software. Стирание происходит программно.

– Генерирование события timestamp для RTC.

– Настройка доступа к backup reg.

1.3.15.6.2 Описание TAMP контроллера

Данный блок необходим для стирания конфиденциальной информации или блокировки доступа к ней при обнаружении попытки вскрытия корпуса или другого воздействия на микросхему. Данный контроллер может работать в VBAT режиме.

Для того, чтобы это обнаруживать внешнее воздействие на микросхему используются 5 внешних и 5 внутренних входов. Внешние входы выведены через PAD на одну из ножек микроконтроллера, а внутренние соединены с блокам, которые находятся внутри микроконтроллера. (см рис. 1.192).

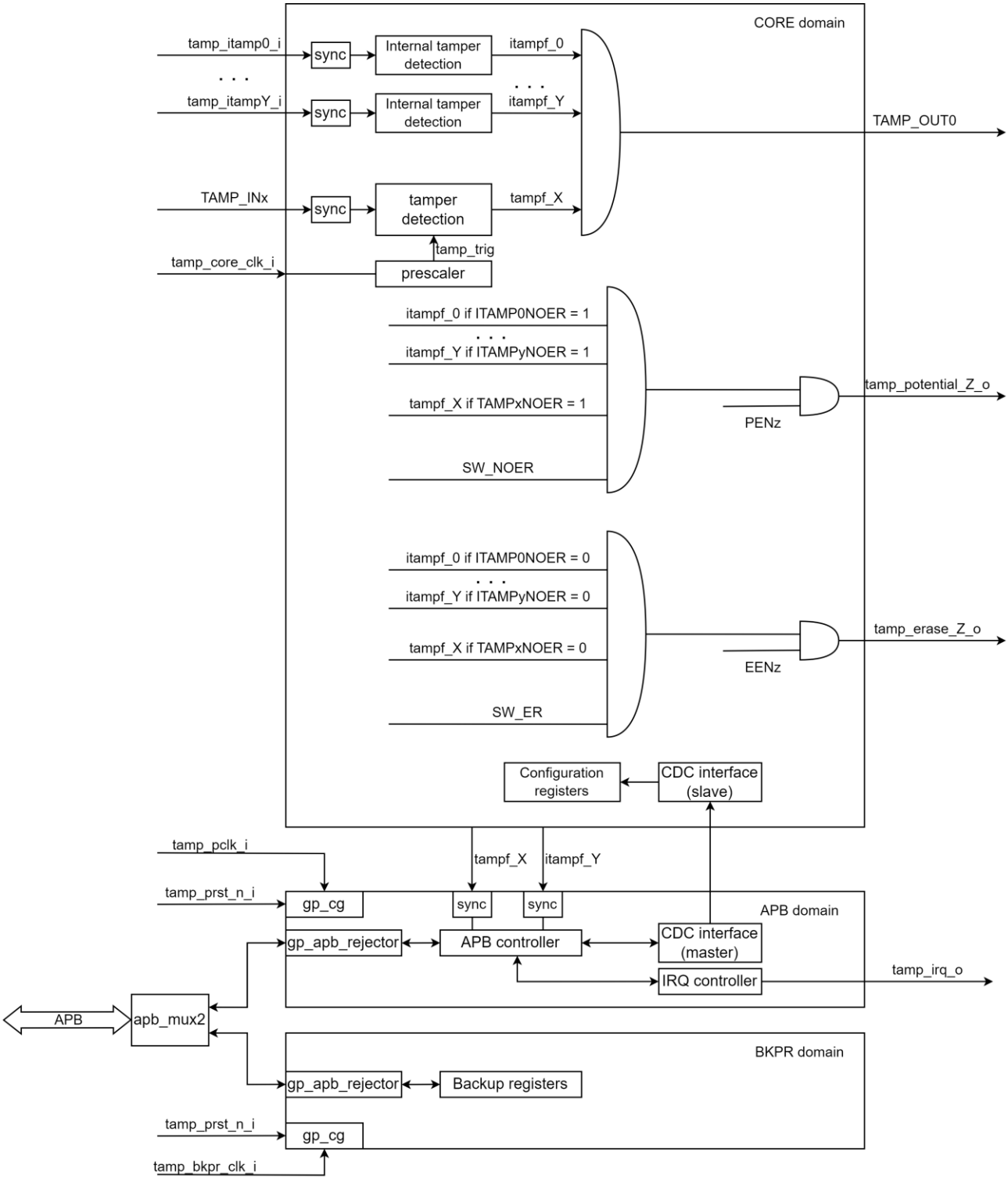


Рисунок 1.192 – Структурная схема TAMP

В таб. 1.839 представлено описание блоков TAMP контроллера.

Таблица 1.839 – Описание назначения составных блоков LCD контроллера

Название блока	Назначение и основные функции
APB controller	Блок служит для чтения регистров состояний, чтения и записи управляющих регистров.
gp_apb_rejector	Блок служит для корректного завершения APB транзакций, когда с APB части USART контроллера снят тактовый сигнал. Данный блок может завершать транзакции с ошибкой (pslverr_o в единице) или без нее. Это задается с помощью входа conf_pslverr_i. Если на этом входе единица, то транзакции будут завершаться с ошибкой, когда на USART не подан тактовый сигнал. При этом транзакции записи игнорируются, а чтение возвращает 0x0BAD_ACCE5.
gp_cg	Блок служит для блокирования подачи тактового сигнала. Так как USART имеет два тактовый входа: pclk_i и clk_i, то и блоков gp_cg тоже используется два. Соответственно входов для блокировки тактовый сигналов также два: pclk_enable_i и clk_enable_i. Если на входе блокировки тактового сигнала единицы, то это тактовый сигнал будет снят. Так же данный блок включает в себя синхронизаторы сброса, они дают задержку на снятие сброс в два такта той частоты, на которой они работают, а сброс происходит асинхронно.

1.3.15.6.3 Описание внешних выводов контроллера TAMP

Список внешних выводов контроллера TAMP и их описание приведены в таблице 1.840

Таблица 1.840 – Список внешних выводов контроллера TAMP

Наименование вывода	Тип (Вх/Вых)	Функциональное назначение
TAMP_IN0	Вх	Сигнал внешнего события
TAMP_OUT0	Вых	Сигнал срабатывания TAMP

1.3.15.6.4 Описание внутренних сигналов

В таблице 1.841 приведен список блоков, которые могут сигнализировать об воздействии, а также блоки содержащий информацию, которая может быть стерта или заблокирована.

Таблица 1.841 – Описание входов и выходов.

Сигналы очистки	Описание
Выходы	
tamp_potential_0	Сигнал блокирует доступ к backup registers
tamp_potential_1	Сигнал блокирует доступ к BKPSRAM
tamp_potential_2	Сигнал блокирует доступ к SRAM
tamp_erase_0	Сигнал вызывает очистку в backup registers
tamp_erase_1	Сигнал вызывает очистку в BKPSRAM
tamp_erase_2	Сигнал вызывает очистку в SRAM
Входы	

Сигналы очистки	Описание
tamp_itamp0	Сигнал от BPOR33VBAT
tamp_itamp1	Сигнал от LSE monitor
tamp_itamp2	Сигнал от COMP0
tamp_itamp3	Сигнал от COMP1
tamp_itamp4	Сигнала от ADC, компаратор 0
tamp_itamp5	Сигнала от ADC, компаратор 1
tamp_itamp6	Сигнала от ADC, компаратор 2
tamp_itamp7	Сигнал от IWDG

TAMP разделен на два тактовых домена APB и CORE. Первый тактируется от tamp_pclk_i, а второй от того же тактового сигнала что и RTC (см. пп 1.3.13.6). При этом тактовые домены расположены в разных доменах питания. CORE-домен находится в PD_VBAT, что позволяет продолжать работу в режиме VBAT.

Для доступа к регистрам используется APB4 интерфейс.

1.3.15.6.5 Защита на запись

После аппаратного сброса только для регистров TAMP включена защита на запись (TAMP_BKPxR и TAMP_BKPRCFG не входят в их число). Для того, что бы ее отключить необходимо в регистр TAMP_WP записать 0x7331, а для включения 0x7330.

1.3.15.6.6 Настройка доступа к backup registers

Backup registers это 32 32-х битных регистра. Они разделены по типу доступа на три области:

– Первая. Чтение и запись только для secure транзакций. Диапазон адресов: с TAMP_BKP0R по TAMP_BKxR, где x настраиваемый параметр в поле TAMP_BKPRCFG[BKPWRS].

– Вторая. Чтение non-secure и secure, а запись только secure транзакциями. Диапазон адресов: с TAMP_BKP(x+1)R по TAMP_BKPyR, где y настраиваемый параметр в поле TAMP_BKPRCFG[BKPWS]. Если поле TAMP_BKPRCFG[BKPWS] меньше или равно чем TAMP_BKPRCFG[BKPWRS], то вторая область отсутствует.

– Третья. Чтение и запись non-secure транзакциями. Диапазон адресов: с TAMP_BKPzR по TAMP_BKP31, где z = y + 1. Данная область присутствует, если остальными не занято все имеющиеся 32 регистра.

Запись в регистр TAMP_BKPRCFG возможна только secure транзакциями и с привилегированным доступом.

Если в регистре TAMP_BKPRCFG установить бит BKPR_LOCK, то тогда запись из первых восьми регистров backup registers будет игнорироваться, а чтение возвращать нули. При этом бит BKPR_LOCK может сбросится только аппаратно.

1.3.15.6.7 Разделение доменов

Из рис. 1.192 видно, что блок TAMP разделен на два тактовых домена TAMP и CORE. Первый домен используется для программной настройки второго. Это происходит с помощью CDC interface. Основной принцип работы его заключается в том, что после установки запроса на обновление конфигурационных регистров, путем записи в TAMP_UPDATE_CFG единицы, происходит копирование их из TAMP-домена в CORE-домен. После окончания этого процесса бит TAMP_UPDATE_CFG[UPDATE_CFG] сбросится аппаратно.

К конфигурационным регистрам относятся: TAMP_EN, TAMP_ERASE_CTRL, TAMP_OUT_EN, TAMP_TRIG_CTRL, TAMP_SWER, TAMP_FILTER.

Также через CDC interface происходит сброс флагов TAMPxF и ITAMPxF. Для этого необходимо в соответствующие биты регистра TAMP_STATUS записать единицу. Это вызовет установку запроса на сброс флагов. После окончания запроса биты флагов в TAMP_STATUS сбросятся.

Запрос на обновление конфигурации (update_req) и запрос на очистку флагов (clr_flag_req) занимают более двух тактов CORE-домена. Если они оба установлены будут установлены, то в таком случае они выполняются последовательно.

1.3.15.6.8 Режимы работы TAMP

Блок TAMP может определять факт внешнего воздействия непосредственно с помощью внешних входов (tamp_IN_i), так и по внутренним (tamp_itamp_i). Внешние входы позволяют непосредственно узнать о физическом вскрытии. Например, в корпусе на фотодиод стал попадать свет или оборвалась проволока. Внутренние входы — это сигналы от других блоков расположенных в микроконтроллер. Например, от LSE, ADC, BOR. После обнаружения события на одном из входов, устанавливается соответствующий флаг обнаружения вскрытия (TAMPxF или ITAMPxF). Установку любого из этих флагов можно обозначить как событие tamp_event.

Эти флаги устанавливаются аппаратно, а сбрасываются только программно. Однако, для флага TAMP0F есть возможность включить другой принцип работы с помощью бита TAMP0MSK. Если этот бит будет установлен, то флаг будет автоматически сбрасываться, после того как на входе TAMP_IN0 сигнал перейдет в неактивное состояние. При таком режиме

работы, если входной сигнал не будет находится в активном состоянии достаточное кол-во времени, то может не произойти стирания данных.

Еще одной особенностью внешних входов является наличие настраиваемой частоты сэмлирования входного значения с помощью TAMP_FILTER[FILTER_DIV], а так же выбор активного уровня сигнала и кол-во последовательных выборок этого активного уровня (TAMP_FILTER[FILTER]), необходимых для выставления флага TAMPxF. Если в процессе работы происходит обновление (запись) конфигурационных регистров, то внутренние счетчики фильтра сбрасываются и он начинает считать с начала (см. пп 1.3.15.6.7).

После обнаружения события могу происходить следующие аппаратные действия:

– стирание данных в backup registers, BKPSRAM (см. пп 1.3.5.2.2), SRAM (см. пп 1.3.5.2.1).

– блокировка чтения backup registers, BKPSRAM, SRAM.

– генерирование события для RTC.

В регистре TAMP_IRQ_STATUS отображаются статусы прерываний. Биты в нем выставляются только в момент установки флагов TAMPxF и ITAMPxF. Это означает, что если, например, сбросить бит TAMP_IRQ_STATUS[TAMPxIRQ], когда установлен соответствующий флаг TAMPxF, то этот бит не установится заново.

Для отслеживания флагов TAMPxF и ITAMPxF используется регистр TAMP_STATUS. Их сброс происходит через этот же регистр. Для этого необходимо в соответствующие биты записать единицу. Так как эти флаги находятся в CORE-домене, то их сброс происходит более двух тактов CORE-домена.

Сигналы стирания tamp_erase_z являются уровневыми с активной единицей. Они представляют из себя объединенные по "или" программный флаг ERASE регистра TAMP_SWER и все аппаратные флаги TAMPxF, ITAMPxF, для которых ITAMPxNOER и TAMPxNOER установлены в нуль. При этом каждый из выходов tamp_erase_z может быть заблокирован (подтянут к нулю) с помощью TAMP_OUT_EN[EENz]. Между установкой флагов и выдачей активного уровня на выходы tamp_erase_z есть задержка в один такт, так как выходные сигналы защелкиваются на регистр.

Сигналы блокировки доступа tamp_potential_z являются уровневыми с активной единицей. Они представляют из себя объединенные по "или" программный флаг BLOCK регистра TAMP_SWER и все аппаратные флаги TAMPxF, ITAMPxF, для которых ITAMPxNOER и TAMPxNOER установлены в единицу. При этом каждый из выходов tamp_erase_z может быть

заблокирован (подтянут к нулю) с помощью TAMP_OUT_EN[PENz]. Между установкой флагов и сигналами tamp_potential_z есть задержка в один такт, так как выходные сигналы защелкиваются на регистр.

1.3.15.6.9 Выходные генерируемые события

Для генерирования временных меток в RTC блоке используется сигнал tamp_event_o. Данный сигнал является объединением по "или" всех флагов TAMPxF и ITAMPxF.

Для корректного захвата этого сигнала необходимо, чтобы он находился в единице в течение как минимум трех тактов rtc_clk. Поэтому перед программным сбросом флагов, нужно убедится, что RTC сохранил временную метку.

1.3.15.6.10 Поддержка TrustZone

В таблице 1.842 указаны типы доступа к каждому регистру.

Таблица 1.842 – Поддержка TrustZone

Название области памяти	Привилегированный доступ (privilege)	Защищенный доступ (secure)
TAMP_BKPxR	Настраиваемый. После сброса разрешен любой.	Настраиваемый. После сброса разрешен любой.
TAMP_WP	Только privilege. Чтение доступно всегда.	Только secure. Чтение доступно всегда.
TAMP_SECCFG	Только privilege. Чтение доступно всегда.	Только secure. Чтение доступно всегда.
TAMP_BKPRCFG	Только privilege. Чтение доступно всегда.	Только secure. Чтение доступно всегда.
остальные регистры	Настраиваемый в TAMP_SECCFG.	Настраиваемый в TAMP_SECCFG.

Транзакция у которой не хватает прав доступа завершается по принципу RZWI.

1.3.15.6.11 Прерывания

TAMP блок имеет один выход прерываний tamp_irq_o. Прерывания работают по уровню и являются соединением отдельных битов регистра TAMP_IRQ_STATUS по "или".

После сброса все прерывания запрещены. Для того, что бы включит одно из них, необходимо в соответствующем бите регистра TAMP_IRQ_MASK установить единицу.

Прерывания сбрасываются побитово с помощью регистра TAMP_IRQ_CLEAR, путем записи единицы в соответствующие биты.

1.3.15.6.11.1 Интеграция TAMP

Контроллер разделен на два тактовых домена: apb-домен и core-домен. Это сделано для того, что бы при изменении системной частоты core-домен, выполняющий передачу данных на шине, не надо было перенастраивать.

У контроллера на входе стоит специальный блок gp_apb_rejector. Его назначение заключается в корректном завершении apb-транзакций, пока с TAMP контроллера снято тактирование. В зависимости от входа conf_pslverr_i транзакции могут завершаться с ошибкой или без. Если conf_pslverr_i = 1, то генерируется ошибка. При этом все транзакции записи игнорируются, а чтение возвращает 0x0BAD_ACCE5.

При обращении к TAMP по шине APB младшие два бита (нулевой и первый) адреса не анализируются. APB транзакция на запись с pstrb не равным 4'b1111 возвращается с ошибкой (pslverr = 1) и никакие данные не записываются. Запись по зарезервированным адресам игнорируется, а чтение возвращает 0xEEFFADDC. При этом транзакции завершаются без ошибок. Так как младшие два бита не анализируются, то обращения по не выровненным адресам автоматически выравниваются.

1.3.15.6.12 Описание регистров

1.3.15.6.12.1 Общее описание регистров

Контроллер имеет набор программно доступных регистров, требующий 4 КБайт адресов на системной шине APB.

Поддерживается только 32-х разрядный доступ. Байтовая или двух-байтовая запись будет проигнорирована и возвращена ошибка шины.

Карта регистров приведена в таблице 1.843.

Адреса, который не указаны в таблице 1.843 считаются резервными. Чтение из них возвращает 0xEEFFADDC, а запись игнорируется. TAMP контроллер поддерживает secure и non-secure транзакции (см. пп 1.3.15.6.10).

1.3.15.6.12.2 Карта регистров

Таблица 1.843 – Карта регистров TAMP

Адрес	Название	Доступ	Значение после сброса	Описание
0x000	TAMP_ID	ЧТ	0x504D4154	TAMP ID
0x004	TAMP_VERSION	ЧТ	0x0	Версия TAMP

Адрес	Название	До-ступ	Значение по-сле сброса	Описание
0x008	TAMP_WP	ЧТ/ЗП	0x0	Сброс защиты на запись
0x00C	TAMP_EN	ЧТ/ЗП	0x0	Включение tamp входов
0x010	TAMP_ERASE_CTRL	ЧТ/ЗП	0x0	Настройка стирания
0x014	TAMP_OUT_EN	ЧТ/ЗП	0x0	Включение выходов стирания и блокировки
0x018	TAMP_TRIG_CTRL	ЧТ/ЗП	0x0	Настройка триггеров в tamper detection
0x01C	TAMP_SWER	ЧТ/ЗП	0x0	Программный вызов стирания
0x020	TAMP_SECCFG	ЧТ/ЗП	0x0	Настройка secure и privilege для TAMP
0x024	TAMP_FILTER	ЧТ/ЗП	0x0	Настройка фильтра для внешнего входа
0x028	TAMP_IRQ_STATUS	ЧТ/ЗП	0x0	Статус прерываний
0x02C	TAMP_IRQ_MASK	ЧТ/ЗП	0x0	Маска прерываний
0x030	TAMP_IRQ_CLEAR	ЗП	0x0	Сброс прерываний
0x034	TAMP_IRQ_TEST	ЗП	0x0	Установка TEST прерывания
0x038	TAMP_STATUS	ЧТ/ЗП	0x0	Статус флагов TAMP
0x03C	TAMP_UPDATE_CFG	ЧТ/ЗП	0x0	Установка запроса на обновление конфигурации
Backup registers				
0x000	TAMP_BKPRCFG	ЧТ/ЗП	0x0	Настройка secure и privilege для BKPxR
0x100	TAMP_BKP0R	ЧТ/ЗП	0x0	Backup registers 0
0x104	TAMP_BKP1R	ЧТ/ЗП	0x0	Backup registers 1
. . .				
0x17C	TAMP_BKP31R	ЧТ/ЗП	0x0	Backup registers 31

1.3.15.6.12.3 Описание полей регистров

1.3.15.6.12.3.1TAMP_ID (0x000)

Регистр идентификатора TAMP контроллера

В таблице 1.844 представлено описание полей регистра.

Таблица 1.844 – Описание полей регистра TAMP_ID

Биты	Поле	До-ступ	Значение по-сле сброса	Описание
31:0	TAMP_ID	ЧТ	0x504D4154	Идентификатор контроллера TAMP

1.3.15.6.12.3.2TAMP_VERSION (0x004)

Регистр идентификатора TAMP контроллера

В таблице 1.845 представлено описание полей регистра.

Таблица 1.845 – Описание полей регистра TAMP_VERSION

Биты	Поле	Доступ	Значение после сброса	Описание
31:0	TAMP_VERSION	ЧТ	0x0	Версия контроллера TAMP

1.3.15.6.12.3.3TAMP_WP (0x008)

Регистр блокировки доступа TAMP контроллера

В таблице 1.846 представлено описание полей регистра.

Таблица 1.846 – Описание полей регистра TAMP_WP

Биты	Поле	Доступ	Значение после сброса	Описание
15:0	WP	ЧТ/ЗП	0	Блокировка записи в регистры TAMP. После аппаратного сброса запись во все регистры игнорируется, а чтение проходи успешно. Для разблокировки записи необходимо в это поле записать 0x7331, а для блокировки 0x7330. Если запись разрешена , то из данного поля будет считываться 1. Когда запись заблокирована, то из данного поля считывается 0. 0 - запись в регистры TAMP игнорируется. 1 - запись в регистры TAMP не игнорируется.
31:16	-	ЧТ	0	Зарезервировано

1.3.15.6.12.3.4TAMP_EN (0x00C)

Регистр включения входов TAMP контроллера

В таблице 1.847 представлено описание полей регистра.

Таблица 1.847 – Описание полей регистра TAMP_EN

Биты	Поле	Доступ	Значение после сброса	Описание
x	TAMPxEN	ЧТ/ЗП	0	Включение детектирования на входе TAMP_INx 0 - детектирование выключено. 1 - детектирование включено.
y + 16	ITAMPyEN	ЧТ/ЗП	0	Включение детектирования на входе tamp_itampY 0 - детектирование выключено. 1 - детектирование включено.

1.3.15.6.12.3.5TAMP_ERASE_CTRL (0x010)

Регистр настройки стирания TAMP контроллера

В таблице 1.848 представлено описание полей регистра.

Таблица 1.848 – Описание полей регистра TAMP_ERASE_CTRL

Биты	Поле	Доступ	Значение после сброса	Описание
x	TAMPxNOER	ЧТ/ЗП	0	Режим стирания данных для TAMP_INx 0 - происходит стирание данных. 1 - не происходит стирания данных, а только блокировка доступа.
y + 16	ITAMPyNOER	ЧТ/ЗП	0	Режим стирания данных для tamp_itamp_Y 0 - происходит стирание данных. 1 - не происходит стирание данных, а только блокировка доступа.

1.3.15.6.12.3.6TAMP_OUT_EN (0x014)

Регистр включения выходов стирания и блокировки TAMP контроллера

В таблице 1.849 представлено описание полей регистра.

Таблица 1.849 – Описание полей регистра TAMP_OUT_EN

Биты	Поле	Доступ	Значение после сброса	Описание
z	PENz	ЧТ/ЗП	0	Разрешает использование выхода tamp_potential_z для блокировки записи и чтения блока, подключенному к нему. 0 - блокировка блока на выходе tamp_potential_z запрещена. 1 - блокировка блока на выходе tamp_potential_z разрешена.
z + 16	EENz	ЧТ/ЗП	0	Разрешает использование выхода tamp_erase_z для стирания данных в блоке, подключенному к нему. 0 - стирание данных запрещено. 1 - стирание данных разрешено.

1.3.15.6.12.3.7TAMP_TRIG_CTRL (0x018)

Регистр настройка активных уровней TAMP контроллера

В таблице 1.850 представлено описание полей регистра.

Таблица 1.850 – Описание полей регистра TAMP_TRIG_CTRL

Биты	Поле	Доступ	Значение после сброса	Описание
x	TAMPxTRG	ЧТ/ЗП	0	Активный уровень для TAMP_INx. 0 - высокий уровень активирует TAMPxF. Если FILTER = 00, то фронт является активатором. 1 - низкий уровень активирует TAMPxF. Если FILTER = 00, то срез является активатором.

Биты	Поле	Доступ	Значение после сброса	Описание
16	TAMP0MSK	ЧТ/ЗП	0	Маска для TAMP_IN0. 0 - tampf_0 и TAMPF0 устанавливаются после обнаружения события TAMP_IN0 и сбрасываются только программно. 1 - tampf_0 и TAMPF0 устанавливаются и держатся пока вход TAMP_IN0 находится в активном состоянии.
17	TAMP1MSK	ЧТ/ЗП	0	Маска для TAMP_IN1. 0 - tampf_1 и TAMPF1 устанавливаются после обнаружения события TAMP_IN1 и сбрасываются только программно. 1 - tampf_1 и TAMPF1 устанавливаются и держатся пока вход TAMP_IN1 находится в активном состоянии.
31:16	-	ЧТ	0	Зарезервировано

1.3.15.6.12.3.8TAMP_SWER (0x01C)

Регистр программного запроса на стирание и блокировку

В таблице 1.851 представлено описание полей регистра.

Таблица 1.851 – Описание полей регистра TAMP_SWER

Биты	Поле	Доступ	Значение после сброса	Описание
0	ERASE	ЧТ/ЗП	0	Запись единицы в этот бит приводит к стиранию данных в backup registers и другой конфиденциальной информации (все выходы tamp_erase_z устанавливаются). 0 - не идет процесс стирания. 1 - идет процесс стирания или уже завершен.
1	BLOCK	ЧТ/ЗП	0	Программная блокировка доступа к backup registers и другой конфиденциальной информации (все выходы tamp_potential_z устанавливаются). 0 - доступ разрешен, если ни один из флагов не установлен (TAMPxF, ITAMPxF). 1 - доступ запрещен.
31:2	-	ЧТ	0	Зарезервировано

1.3.15.6.12.3.9TAMP_SECCFG (0x020)

Регистр настройки secure и privilege доступа

В таблице 1.852 представлено описание полей регистра.

Таблица 1.852 – Описание полей регистра TAMP_SECCFG

Биты	Поле	До-ступ	Значение по-сле сброса	Описание
0	TAMP_PRIV	ЧТ/ЗП	0	Устанавливает privilege доступ к регистрам. 0 - запись в конфигурационные регистры возможна privilege и non-privilege транзакциями 1 - запись в конфигурационные регистры возможна только privilege транзакциями
3:1	-	ЧТ	0	Зарезервировано
4	TAMP_SECURE	ЧТ/ЗП	0	Устанавливает secure доступ к регистрам. 0 - запись в конфигурационные регистры возможна secure и non-secure транзакциями 1 - запись в конфигурационные регистры возможна только secure транзакциями
31:5	-	ЧТ	0	Зарезервировано

1.3.15.6.12.3.10 TAMP_FILTER (0x024)

Регистр входного фильтра TAMP контроллера

В таблице 1.853 представлено описание полей регистра.

Таблица 1.853 – Описание полей регистра TAMP_FILTER

Биты	Поле	До-ступ	Значение по-сле сброса	Описание
2:0	FILTER_DIV	ЧТ/ЗП	0	Частота сэмплирования сигнала на входах TAMP_INx. 000 - tamp_clk / 32768 001 - tamp_clk / 16384 010 - tamp_clk / 8192 011 - tamp_clk / 4096 100 - tamp_clk / 2048 101 - tamp_clk / 1024 110 - tamp_clk / 512 111 - tamp_clk / 256
4:3	FILTER	ЧТ/ЗП	0	Это поле определяет сколько последовательных выборок (сэмплов) активного значение на входах TAMP_INx необходимо для того, чтобы активировать TAMPxF (tampf_X). 00 - TAMPxF активируется после 1 последовательных выборок активного уровня 01 - TAMPxF активируется после 2 последовательных выборок активного уровня 10 - TAMPxF активируется после 4 последовательных выборок активного уровня 11 - TAMPxF активируется после 8 последовательных выборок активного уровня
31:5	-	ЧТ	0	Зарезервировано

1.3.15.6.12.3.11 TAMP_IRQ_STATUS (0x028)

Регистр статуса прерываний TAMP контроллера

В таблице 1.854 представлено описание полей регистра.

Таблица 1.854 – Описание полей регистра TAMP_IRQ_STATUS

Биты	Поле	Доступ	Значение после сброса	Описание
0	TEST_IRQ	ЧТ	0	Статусный бит прерывания TEST_IRQ. Когда данный бит находится в единице, то данное прерывание установлено.
x + 1	TAMPxIRQ	ЧТ	0	Устанавливается по фронту статусного флага TAMPxF.
y + 16	ITAMPyIRQ	ЧТ	0	Устанавливается по фронту статусного флага ITAMPyF.

1.3.15.6.12.3.12 TAMP_IRQ_MASK (0x02C)

Регистр маски прерываний TAMP контроллера

В таблице 1.855 представлено описание полей регистра.

Таблица 1.855 – Описание полей регистра TAMP_IRQ_MASK

Биты	Поле	Доступ	Значение после сброса	Описание
0	TEST_M	ЧТ/ЗП	0	Маска прерываний для TEST_IRQ прерывания 0 - прерывание запрещено 1 - прерывание разрешено
x + 1	TAMPxM	ЧТ/ЗП	0	Маска прерываний для TAMPxIRQ 0 - прерывание запрещено 1 - прерывание разрешено
y + 16	ITAMPyM	ЧТ/ЗП	0	Маска прерываний для ITAMPyF 0 - прерывание запрещено 1 - прерывание разрешено

1.3.15.6.12.3.13 TAMP_IRQ_CLEAR (0x030)

Регистр сброса прерываний TAMP контроллера

В таблице 1.856 представлено описание полей регистра.

Таблица 1.856 – Описание полей регистра TAMP_IRQ_CLEAR

Биты	Поле	Доступ	Значение после сброса	Описание
y + 16	ITAMPyCLR	ЗП	0	Запись единицы в данный бит сбрасывает прерывание ITAMPyIRQ
x + 1	TAMPxCLR	ЗП	0	Запись единицы в данный бит сбрасывает флаг TAMPxIRQ

Биты	Поле	Доступ	Значение после сброса	Описание
0	TEST_CLR	ЗП	0	Запись единицы в данный бит сбрасывает прерывание TEST_IRQ.

1.3.15.6.12.3.14 TAMP_STATUS (0x038)

Регистр статуса TAMP контроллера

В таблице 1.857 представлено описание полей регистра.

Таблица 1.857 – Описание полей регистра TAMP_STATUS

Биты	Поле	Доступ	Значение после сброса	Описание
y + 16	ITAMPyF	ЧТ/ЗП	0	Устанавливается после того как детектируется событие на входе tamp_itampY. Запись единицы запускает процесс сброса флага в core-домене
x + 1	TAMPxF	ЧТ/ЗП	0	Устанавливается после того как детектируется событие на входе TAMP_INx. Запись единицы запускает процесс сброса флага в core-домене
0	-	ЧТ	0	Зарезервировано

1.3.15.6.12.3.15 TAMP_UPDATE_CFG (0x03C)

Регистр запроса на обновление конфигурации TAMP контроллера

В таблице 1.858 представлено описание полей регистра.

Таблица 1.858 – Описание полей регистра TAMP_UPDATE_CFG

Биты	Поле	Доступ	Значение после сброса	Описание
0	UPDATE_CFG	ЧТ/ЗП	0	Запись в данный бит единицы запускает процесс копирования конфигурационных регистров из APB домена в CORE. После окончания копирования бит аппаратно сбросится в нуль.
31:1	-	ЧТ	0	Зарезервировано

1.3.15.6.12.3.16 TAMP_BKPRCFG (0x000)

Регистр настройки secure и private доступа к backup registers

В таблице 1.859 представлено описание полей регистра.

Таблица 1.859 – Описание полей регистра TAMP_BKPRCFG

Биты	Поле	До-ступ	Значение по-сле сброса	Описание
5:0	BKPR_WRS	ЧТ/ЗП	0	Задает область backup registers в которую разрешена запись и чтение только secure транзакциями (первая область). Допустимые значения от 0 до 32. Если BKPR_WRS = 0, то это означает, что первой области backup registers нет. Если BKPR_WRS = 32, то это означает, что во все backup registers возможен только secure доступ.
14:6	-	ЧТ	0	Зарезервировано
15	BKPR_WRP	ЧТ/ЗП	0	Устанавливает privilege доступ к первой области backup registers. 0 - запись и чтение в первую область backup registers возможна privilege и non-privilege транзакциями 1 - запись и чтение в первую область backup registers возможна только privilege транзакциями
21:16	BKPR_WS	ЧТ/ЗП	0	Задает область backup registers в которую разрешена запись только secure транзакциями (вторая область). Допустимые значения от 0 до 32 и при этом BKPWS должно быть больше, чем BKPR_WRS. Если BKPR_WS = 0, то это означает, что второй области backup registers нет. Если BKPR_WS = 32, то это означает, что третей области backup registers нет.
29:22	-	ЧТ	0	Зарезервировано
30	BKPR_WP	ЧТ/ЗП	0	Устанавливает privilege доступ ко второй области backup registers. 0 - запись и чтение во вторую область backup registers возможна privilege и non-privilege транзакциями 1 - запись и чтение во вторую область backup registers возможна только privilege транзакциями
31	BKPR_LOCK	ЧТ/ЗП	0	Блокирует запись в регистры с TAMP_BKP0R по TAMP_BKP7R. Данный бит сбрасывается только после аппаратного сброса батарейного домена или после стирания данных в backup registers. 0 - доступ в регистры с TAMP_BKP0R по TAMP_BKP7R разрешен. 1 - запись в регистры с TAMP_BKP0R по TAMP_BKP7R игнорируется.

1.3.16 Подсистема отладки и трассировки (DBG)

МК Optimal+ включает подсистему отладки и трассировки, реализованную на базе открытой архитектуры CoreSight. Подсистема отладки и трассировки обеспечивает пользователю следующие возможности:

– ввод ядра в режим отладки различными способами:

a) с помощью аппаратных точек останова (hardware breakpoint);

b) с помощью программных точек останова (software breakpoint);

c) с помощью аппаратных точек наблюдения (hardware watchpoint);

d) по событию, полученному через интерфейс обмена триггерными событиями (Cross Trigger Interface);

e) по прерыванию (Vector catch);

f) по запросу от внешнего отладчика;

g) по исполнении очередной инструкции в режиме пошаговой отладки;

– работу с компонентами отладки и трассировки во время системного сброса SRST (debug-under-reset);

– задержку загрузки ядра;

– доступ к внутренним регистрам ядра (core registers) в режиме отладки;

– доступ к памяти и периферийным устройствам по системной шине посредствам ядра в режиме отладки;

– трассировку исполняемой программы, с помощью дополнительного отладочного кода в программе (printf-style debugging);

– трассировку исполняемой программы без дополнительного отладочного кода в программе.

Поддержка функции трассировки выполняется через 4-битный порт сбора трасс (Trace Port).

Подсистема отладки и трассировки приведена на рисунке 1.193.

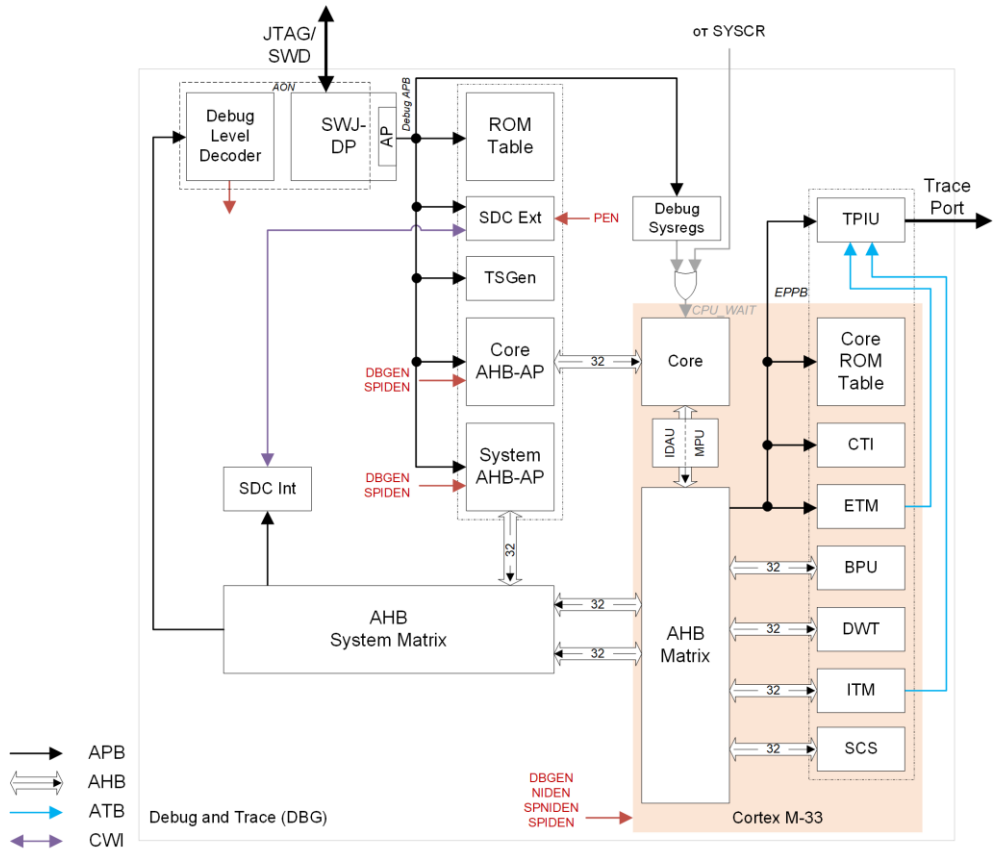


Рисунок 1.193 – Подсистема отладки и трассировки МК Optimal+

1.3.16.1 Контроллер отладочного порта SWJ-DP

Управление подсистемой отладки и трассировки выполняется через стандартный JTAG/Serial-Wire порт SWJ-DP DPv3 с использованием широкодоступных аппаратных отладчиков. По Power-on сбросу контроллер отладочного порта SWJ-DP настроен на работу по интерфейсу JTAG (режим JTAG). При этом контроллер SWJ-DP соединён в цепочку с TAP-контроллером граничного сканирования Boundary Scan (см. рисунок 1.14). TAP-контроллер граничного сканирования должен находиться в состоянии BYPASS, когда внешнее отладочное устройство работает с контроллером SWJ-DP в режиме JTAG. В случае переключения отладочного порта на работу по интерфейсу Serial-Wire (режим Serial-Wire), TAP-контроллер граничного сканирования будет находится в состоянии сброса. Разделы 1.3.16.1.2 и 1.3.16.1.2 содержат информацию о смене режимов контроллера SWJ-DP. Выводы, использующиеся в режимах JTAG и Serial-Wire приведены в таблице 1.860.

Таблица 1.860 – Список выводов контроллера отладочного порта SWJ-DP

Наименование вывода	Функциональное назначение
JTCK/SWCLK	Вход тактового сигнала контроллера JTAG порта/ Вход тактового сигнала контроллера Serial-Wire порта
JTDI	Вход последовательных данных контроллера JTAG порта
JTDO/SWO	Выход последовательных данных контроллера JTAG порта/ Выход последовательных данных порта сбора трасс TPIU[1)]
JTMS/SWDIO	Вход выбора режима тестирования. Обеспечивает переключение между разными режимами тестирования контроллера JTAG порта/ Двунаправленный сигнал последовательных данных контроллера Serial-Wire порта
NJTRST	Вход сигнала сброса контроллера JTAG порта (активный уровень — "0")
[1)] Прочие выводы порта сбора трасс приведены в разделе 1.3.16.7	

Контроллер SWJ-DP осуществляет доступ к компонентам отладки и трассировки при помощи встроенного порта доступа AP, являющегося ведущим устройством на шине Debug APB.

В таблице 1.861 приведены компоненты подсистемы отладки и трассировки, которые доступны по шине Debug APB контроллеру SWJ-DP.

Таблица 1.861 – Компоненты подсистемы отладки и трассировки, доступные по шине Debug APB

Диапазон смещений	Название компонента	Описание
0x0000-0x1FFF	ROM Table	Главная корневая таблица конфигурации подсистемы отладки и трассировки
0x2000-0x3FFF	Secure Debug Channel Ext (SDC Ext)	"Внешний" компонент безопасного канала связи между внешним отладочным устройством и ядром
0x4000-0x5FFF	CPUWAIT Control	Блок управления сигналом задержки загрузки ядра
0x6000-0x7FFF	Core AHB-AP	Порт доступа к шине AHB ядра
0x8000-0x9FFF	System AHB-AP	Порт доступа к системной шине AHB
0xA000-0xBFFF	Timestamp Generator (TSGen)	Генератор временных меток

В таблице 1.862 приведены компоненты подсистемы отладки и трассировки, которые доступны порту доступа к шине AHB ядра.

Таблица 1.862 – Компоненты подсистемы отладки и трассировки, доступные порту Core AHB-AP

Диапазон смещений	Название компонента	Описание
0xE00F_F000-0xE00F_FFFF	Core ROM Table	Корневая таблица компонентов отладки и трассировки ядра
0xE000_E000-0xE000_EFFF	System Control Space (SCS)	Компонент, позволяющий считывать и изменять состояние ядра, управлять им в режиме отладки
0xE000_1000-0xE000_1FFF	Data Watchpoint and Trace unit (DWT)	Компонент, позволяющий задавать аппаратные точки наблюдения и содержащий ряд счётчиков, собирающих метрики об исполняемой ядром программе
0xE000_2000-0xE000_2FFF	Breakpoint unit (BPU)	Компонент, позволяющее задавать аппаратные точки останова
0xE000_0000-0xE000_0FFF	Instrumentation Trace Macrocell (ITM)	Компонент, генерирующий трассировочную информацию в реальном времени с помощью дополнительного отладочного кода (printf style debugging), внедрённого в программу
0xE004_0000-0xE004_0FFF	Trace Port Interface Unit (TPIU)	Порт сбора трасс
0xE004_1000-0xE004_1FFF	Embedded Trace Macrocell (ETM)	Компонент, генерирующий трассировочную информацию в реальном времени на основании инструкций, исполняемых ядром
0xE004_2000-0xE004_2FFF	Cross Trigger Interface (CTI)	Компонент, управляющий распространением сигналов через интерфейс обмена триггерными событиями

Внешнее отладочное устройство с помощью протоколов JTAG и Serial-Wire может взаимодействовать со следующими регистрами:

– регистр DPACC позволяет осуществить доступ к другим регистрам контроллера JTAG-DP, приведённым в таблице 1.865.

– регистр APACC позволяет сформировать запрос на чтение/запись данных портом доступа AP.

1.3.16.1.1 Контроллер JTAG порта

Работа контроллера SWJ-DP по интерфейсу JTAG обеспечивается с помощью машины состояний DBGTAPSM. Она позволяет внешнему отладочному устройству взаимодействовать с регистрами DPACC, APACC, а также ABORT, IDCODE и BYPASS. Последние два не могут быть доступны в режиме Serial-Wire. Схема взаимодействия с портом доступа AP приведена на рисунке 1.194.

– Регистр IDCODE позволяет внешнему отладочному устройству определить идентификатор контроллера JTAG порта;

– регистр ABORT позволяет отменить запрос на чтение/запись данных через порт доступа AP, а также сбросить флаги о зарегистрированных ошибочных ситуациях. Более подробно данный регистр описан в разделе 1.3.16.1.4.

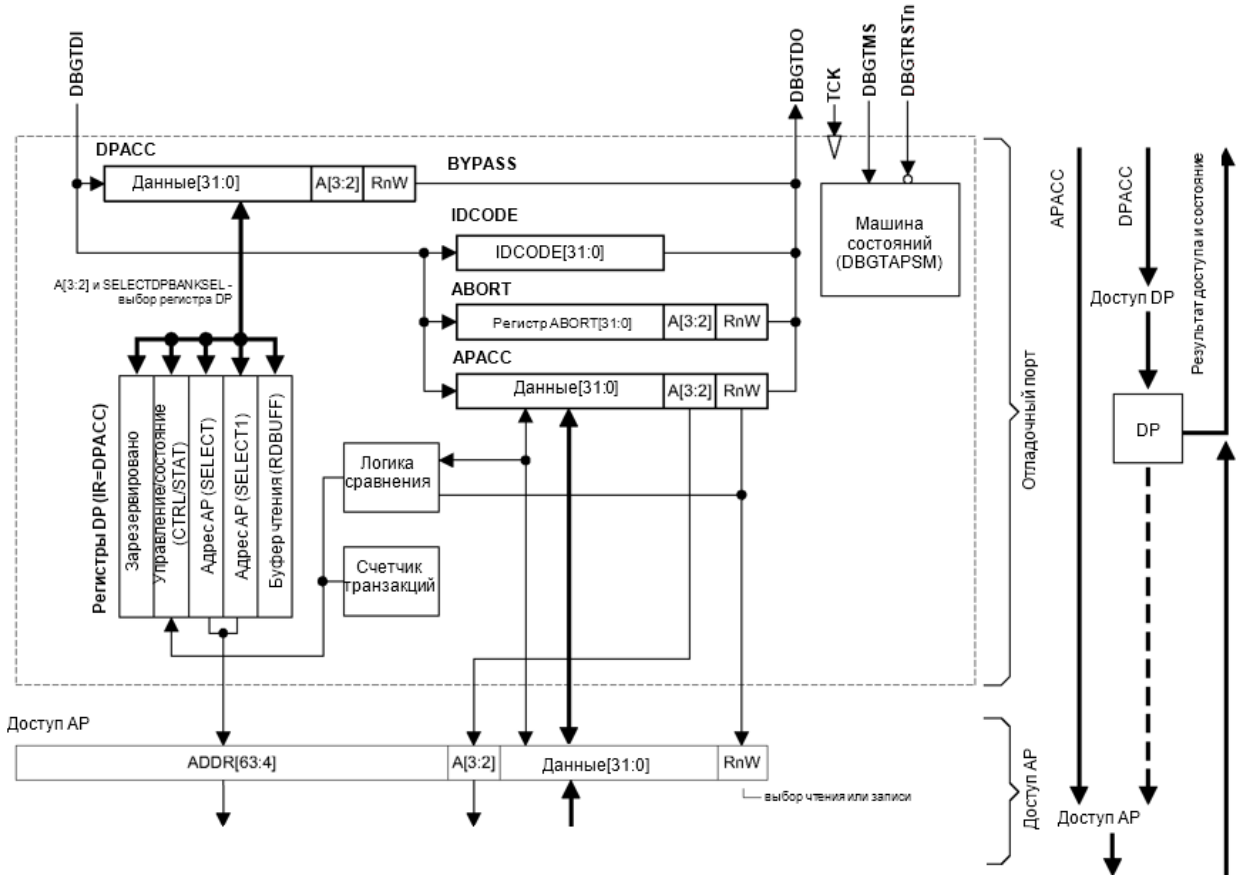


Рисунок 1.194 – Схема контроллера JTAG порта

Машина состояний DBGTAPSM работает в соответствии со стандартом IEEE-1149.1-1990. Конечный автомат её состояний приведён на рисунке 1.195. В процессе работы осуществляется захват значения регистра инструкций IR[3:0] и захват или выдача значения регистра данных DR в зависимости от значения регистра IR. В таблице 1.863 приведены коды инструкций и соответствующие им регистры данных.

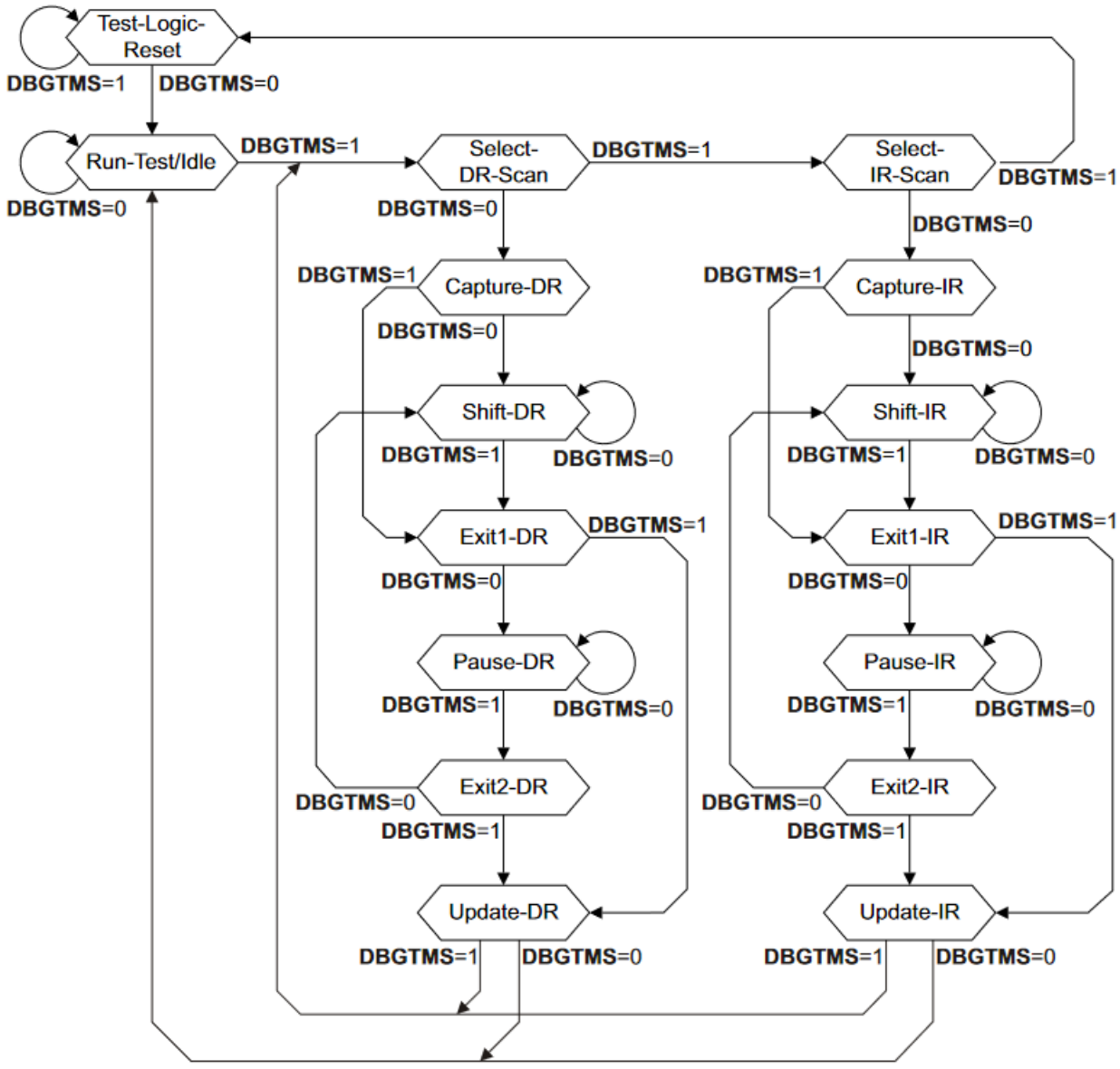


Рисунок 1.195 – Машина состояний DBGTAPSM

Таблица 1.863 – Инструкции DBGTAPSM

Значение регистра IR	Регистр DR	Длина регистра DR	Описание
0000-0111	BYPASS	1	Контроллер JTAG порта в режиме BYPASS

Значение регистра IR	Регистр DR	Длина регистра DR	Описание
1000	ABORT	35	34:8 — зарезервировано; 7 — ORUNERRCLR: сброс поля STICKYORUN регистра CTRL/STAT; 6 — WDERRCLR: сброс поля WDATAERR регистра CTRL/STAT; 5 — STKERRCLR: сброс поля STICKYERR регистра CTRL/STAT; 4 — зарезервировано; 3 — APABORT: запись "1" приводит к отмене запроса на чтение/запись данных через порт доступа AP 2:0 — должны быть "000"
1001	BYPASS	1	Контроллер JTAG порта в режиме BYPASS
1010	DPACC	35	При получении данных от TDI: 34:3 — DATA[31:0]: данные для записи 2:1 — A[3:2]: адрес регистра 0 — RnW: "1" — запрос на чтение; "0" — запрос на запись. При подаче данных на TDO: 34:3 — DATA[31:0]: считанные данные 2:0 — ACK[2:0]: код результата транзакции (отклик): - "001": WAIT - "010": FAULT - "100": OK - прочие: зарезервировано
1011	APACC	35	При получении данных от TDI: 34:3 — DATA[31:0]: данные для записи 2:1 — A[3:2]: разряды [3:2] адреса для выставления портом доступа AP 0 — RnW: "1" — запрос на чтение; "0" — запрос на запись. При подаче данных на TDO: 34:3 — DATA[31:0]: считанные данные 2:0 — ACK[2:0]: код результата транзакции (отклик): - "001": WAIT - "010": FAULT - "100": OK - прочие: зарезервировано
1100,1101	BYPASS	1	Контроллер JTAG порта в режиме BYPASS
1110	IDCODE	32	0x4C01_3477: Идентификатор контроллера JTAG порта
1111	BYPASS	1	Контроллер JTAG порта в режиме BYPASS

Для связи с контроллером SWJ-DP по протоколу Serial-Wire используются используются следующие выводы:

– SWCLK: тактовый сигнал от внешнего отладочного устройства;

– SWDIO: двунаправленный вывод для обмена последовательными данными.

Данные передаются синхронно с тактовым сигналом, начиная с младшего разряда (LSB).

Используется SWD protocol version 2.

Передача/приём данных по протоколу Serial-Wire состоит из трёх фаз:

1) Запрос из 8 бит на запись/чтение от внешнего отладочного устройства;

2) Отклик ACK[2:0] от контроллера SWJ-DP из 3х бит;

3) Передача/приём 33х битов данных (WDATA/RDATA и бит чётности) от внешнего отладочного устройства или от контроллера SWJ-DP соответственно в зависимости от типа запроса.

При этом каждый раз при смене направления следования данных через вывод SWDIO между фазами по умолчанию выдерживается один такт частоты SWCLK.

Таблица 1.864 – Биты запроса на передачу/приём данных по протоколу Serial-Wire

Биты запроса	**Название**	**Описание**
0	Start	старт-бит, "1"
1	APnDP	- 0: доступ к регистрам контроллера SWJ-DP; - 1: доступ к регистрам порта доступа AP.
2	RnW	- 0: запрос на запись; - 1: запрос на чтение.
4:3	A[2:3]	смещение регистра контроллера SWJ-DP (см. таблицу 1.865) или разряды [3:2] адреса для выставления портом доступа AP
5	Parity	бит чётности
6	Stop	стоп-бит, "0"
7	Park	ведущее отладочное устройство отпускает линию

Значения кодов откликов:

– 000: FAULT;

– 010: WAIT;

– 100: OK.

Передача/приём данных считается успешно завершённой, если значение отклика OK.

Отклик WAIT возникает, если контроллера SWJ-DP не успел обработать предыдущий запрос.

В случае, если не включено обнаружение ситуации overrun detection (см. раздел 1.3.16.1.4.5) и полученный отклик принял значение WAIT или FAULT третья фаза передачи/приёма данных отменяется.

1.3.16.1.2 Переключение контроллера SWJ-DP в режим Serial-Wire

Для того, чтобы контроллер отладочного порта SWJ-DP переключился в режим Serial-Wire необходимо следующее:

1) Удержать вывод SWDIOTMS в "1" на протяжении 50 тактов SWCLKTCK. Это приведёт к сбросу используемого в текущий момент интерфейса;

2) Отправить последовательность из следующих 16 битов на вывод JTMS/SWDIO: 0xE79E — передавать, начиная с младшего разряда (LSB);

3) Удержать вывод SWDIOTMS в "1" на протяжении 50 тактов SWCLKTCK. Это приведёт к сбросу интерфейса Serial-Wire и требуется на случай, если до начала переключения контроллер отладочного порта SWJ-DP уже работал в режиме Serial-Wire.

Временная диаграмма переключения приведена на рисунке 1.196.

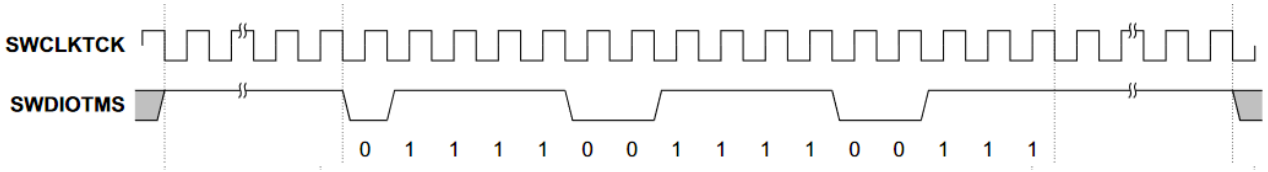


Рисунок 1.196 – Временная диаграмма переключения с JTAG на Serial-Wire

1.3.16.1.3 Переключение контроллера SWJ-DP в режим JTAG

Для того, чтобы контроллер отладочного порта SWJ-DP переключился в режим JTAG необходимо следующее:

1) Удержать вывод SWDIOTMS в "1" на протяжении 50 тактов SWCLKTCK. Это приведёт к сбросу используемого в текущий момент интерфейса;

2) Отправить последовательность из следующих 16 битов на вывод JTMS/SWDIO: 0xE73C — передавать, начиная с младшего разряда (LSB);

3) Удержать вывод SWDIOTMS в "1" на протяжении по крайней мере пяти тактов SWCLKTCK. Это приведёт к сбросу интерфейса JTAG (переходу в состояние Test-Logic-Reset) и требуется на случай, если до начала переключения контроллер отладочного порта SWJ-DP уже работал в режиме JTAG.

Временная диаграмма переключения приведена на рисунке 1.197.

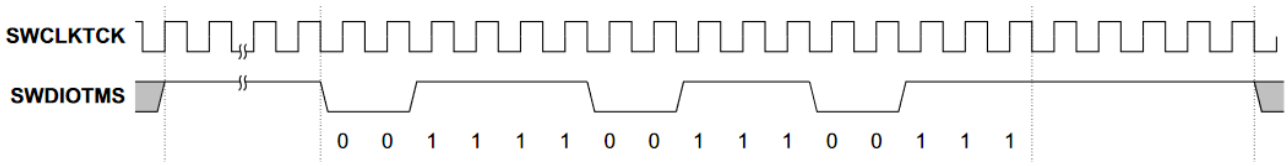


Рисунок 1.197 – Временная диаграмма переключения с Serial-Wire на JTAG

1.3.16.1.4 Регистры контроллера отладочного порта SWJ-DP

В таблице 1.865 приведён список регистров контроллера отладочного порта SWJ-DP. Данные регистры недоступны ядру, доступ к ним возможен только с помощью JTAG и Serial-Wire.

Старшие разряды смещения задаются полем A[2:3]. Два младших разряда смещения — 0x0. Для доступа к некоторым из регистров необходимо сначала задать поле DPBANKSEL регистра SELECT, указывающее адрес банка регистров контроллера SWJ-DP.

Таблица 1.865 – Регистры контроллера отладочного порта SWJ-DP

Смещение	Значение поля DPBANKSEL регистра SELECT	Название	Тип доступа
0x0	x	ABORT	ЗП (доступ только в режиме Serial-Wire. Доступ в режиме JTAG описан в разделе 1.3.16.1.4.1)
	0x0	DPIDR	ЧТ (доступ только в режиме JTAG)
	0x1	DPIDR1	ЧТ (доступ только в режиме JTAG)
	0x2	BASEPTR0	ЧТ (доступ только в режиме JTAG)
	0x3	BASEPTR1	ЧТ (доступ только в режиме JTAG)
0x4	0x0	CTRL/STAT	ЧТ/ЗП
	0x1	DLCR	ЧТ/ЗП (доступ только в режиме Serial-Wire)
	0x2	TARGETID	ЧТ
0x8	x	SELECT	ЗП
0xC	x	RDBUFF	ЧТ (доступ только в режиме Serial-Wire)

1.3.16.1.4.1 Регистр ABORT

Регистр ABORT позволяет отменить запрос на чтение/запись данных через порт доступа AP, а также сбросить флаги о зарегистрированных ошибочных ситуациях.

В режиме JTAG доступ к регистру осуществляется с помощью инструкции ABORT, задаваемой регистром IR машины состояний DBGTAPSM.

В режиме Serial-Wire доступ к регистру осуществляется по смещению 0x0, которое задаётся полем A[2:3] запроса на передачу/приём данных.

В таблице 1.866 приведено описание полей регистра ABORT.

Таблица 1.866 – Разряды регистра ABORT

Разряды	Значение по сбросу	Название	Тип доступа	Описание
31:5	-	-	ЧТ0	Зарезервировано

Разряды	Значение по сбросу	Название	Тип доступа	Описание
4	0	ORUNERRCLR	ЧТ0	Запись "1" в данный разряд приводит к сбросу бита ошибки переполнения STICKYORUN регистра CTRLSTAT
3	0	WDERRCLR	ЧТ0	Запись "1" в данный разряд приводит к сбросу бита ошибки записи данных WDATAERR регистра CTRLSTAT
2	0	STKERRCLR	ЧТ0	Запись "1" в данный разряд приводит к сбросу бита ошибки STICKYERR регистра CTRLSTAT
1	-	-	ЧТ0	Зарезервировано
0	0	ABORTTRANS	ЧТ0	Запись "1" в данный разряд приводит к отмене текущего запроса порта доступа AP

1.3.16.1.4.2 Регистр DPIDR

Регистр DPIDR доступен только в режиме JTAG. Он содержит идентификатор контроллера JTAG порта. В таблице 1.867 приведено описание полей регистра DPIDR.

Таблица 1.867 – Разряды регистра DPIDR

Разряды	Значение по сбросу	Название	Тип доступа	Описание
31:0	0x4C01_3477	DPIDR	ЧТ	Идентификатор контроллера JTAG порта

1.3.16.1.4.3 Регистр DPIDR1

Регистр DPIDR1 содержит дополнительную информацию о контроллере отладочного порта SWJ-DP и является дополнением к регистру DPIDR. В таблице 1.868 приведено описание полей регистра DPIDR1.

Таблица 1.868 – Разряды регистра DPIDR1

Разряды	Значение по сбросу	Название	Тип доступа	Описание
31:8	-	-	ЧТ0	Зарезервировано
7	1	ERRMODE	ЧТ	Данное поле содержит информацию о том, что в данном контроллере SWJ-DP реализован следующий механизм: ошибки, возникшие при запросах от порта доступа AP могут приводить к отклику FAIL по интерфейсам JTAG и Serial-Wire, при этом происходит сброс бита ошибки STICKYERR в регистре CTRLSTAT
6:0	0x20	ASIZE	ЧТ	Данное поле содержит информацию о ширине адреса, выставляемого портом доступа AP

1.3.16.1.4.4 Регистры BASEPTR0 и BASEPTR1

Данные регистры задают базовый адрес главной корневой таблицы (ROM Table) конфигурации подсистемы отладки и трассировки в адресном пространстве внешнего отладочного устройства. Адрес выровнен на границу 4 КБайта. В таблицах 1.869 и 1.870 приведены описания полей регистров.

Таблица 1.869 – Разряды регистра BASEPTR0

Разряды	Значение по сбросу	Название	Тип доступа	Описание
31:12	0x0000	PTR	ЧТ	Разряды [31:12] базового адреса главной таблицы ROM Table
11:1	-	-	ЧТ0	Зарезервировано
0	1	VALID	ЧТ	Данное поле содержит информацию о том, что значение в поле PTR валидное

Таблица 1.870 – Разряды регистра BASEPTR1

Разряды	Значение по сбросу	Название	Тип доступа	Описание
31:0	0x0000_0000	PTR	ЧТ	Разряды [63:32] базового адреса главной таблицы ROM Table

1.3.16.1.4.5 Регистр CTRL/STAT

Регистр CTRL/STAT предназначен для управления контроллером отладочного порта SWJ-DP и мониторинга его состояния.

Таблица 1.871 – Разряды регистра CTRL/STAT

Разряды	Значение по сбросу	Название	Тип доступа	Описание
31:30	-	-	ЧТ0	Зарезервировано
29	0	CDBGPWRUPACK	ЧТ	0 — тактирование компонентов подсистемы отладки и трассировки выключено 1 — питание и тактирование всех компонентов подсистемы отладки и трассировки за исключением порта сбора трасс TPIU включено. Управление тактированием порта TPIU управляется отдельно — полем TRACE_EN регистра RCU_PCER1 подсистемы RCU

Разряды	Значение по сбросу	Название	Тип доступа	Описание
28	-	CDBGPWRUPREQ	ЧТ/ЗП	0 — выставлен запрос на выключение тактирования всех компонентов подсистемы отладки и трассировки. Питание при этом не снимается 1 — выставлен запрос на включение питания и тактирования всех компонентов подсистемы отладки и трассировки за исключением порта TPIU. Его тактирование управляется отдельно — полем TRACE_EN регистра RCU_PCER1 подсистемы RCU
27	0	CDBGRSTACK	ЧТ	0 — сброс компонентов подсистемы отладки и трассировки, находящихся вне ядра, ещё не завершён или не был запрошен 1 — сброс компонентов подсистемы отладки и трассировки, находящихся вне ядра завершён и снят
26	-	CDBGRSTREQ	ЧТ/ЗП	0 — отсутствует запрос на сброс компонентов подсистемы отладки и трассировки, находящихся вне ядра. Запись "0" в целях снятия запроса на сброс разрешена только после установки бита CDBGRSTACK 1 — выставлен запрос на сброс компонентов подсистемы отладки и трассировки, находящихся вне ядра
25	-	-	ЧТ0	Зарезервировано
24	0	ERRMODE	ЧТ/ЗП	0 — ошибки, возникшие при запросах от порта доступа AP приводят к установке бита ошибки STICKYERR в регистре CTRLSTAT 1 — ошибки, возникшие при запросах от порта доступа AP могут приводить к отклику FAIL по интерфейсам JTAG и Serial-Wire, при этом происходит сброс бита ошибки STICKYERR в регистре CTRLSTAT
23:8	-	-	ЧТ0	Зарезервировано
7	0	WDATAERR	ЧТ в режиме Serial-Wire ЧТ0 в режиме JTAG	Данный разряд устанавливается в "1" при возникновении следующих ошибок: — ошибки чётности (parity) или по нарушению целостности (framing) запроса на запись данных по интерфейсу Serial-Wire; — ошибки в запросе на запись, принятом контроллером SWJ-DP, но отклонённом и не выставленном портом доступа AP Данный разряд может быть сброшен записью "1" в поле WDERRCLR
6	0	READOK	ЧТ в режиме Serial-Wire ЧТ0 в режиме JTAG	Данный разряд индицирует успешность последнего запроса на чтение от порта доступа AP 0 — отклик от последнего запроса на чтение — OK 1 — отклик от последнего запроса на чтение — WAIT или FAULT

Разряды	Значение по сбросу	Название	Тип доступа	Описание
5	0	STICKYERR	ЧТ/ЗП1 в режиме JTAG ЧТ в режиме Serial-Wire	Данный разряд устанавливается в "1" по ошибке в результате запроса от порта доступа AP. Сброс данного разряда осуществляется записью "1" в данный разряд (только в режиме JTAG), а также записью "1" в разряд STKERRCLR регистра ABORT
4:2	-	-	ЧТ0	Зарезервировано
1	0	STICKYORUN	ЧТ/ЗП1 в режиме JTAG ЧТ в режиме Serial-Wire	Данный разряд устанавливается в "1" в случае, если очередной запрос к контроллеру SWJ-DP по интерфейсу JTAG или Serial-Wire поступил до окончания обработки предыдущего (overrun detected). Сброс данного разряда осуществляется записью "1" в данный разряд (только в режиме JTAG), а также записью "1" в разряд STKERRCLR регистра ABORT
0	-	ORUNDETECT	ЧТ/ЗП	0 — Обнаружение overrun выключено 1 — Обнаружение overrun включено

1.3.16.1.4.6 Регистр SELECT

Данный регистр задаёт адрес, выставляемый портом доступа AP, а также адрес банка регистров контроллера SWJ-DP (см. таблицу 1.864), в который будет осуществляться дальнейший доступ. В таблице 1.872 приведёно описание полей регистра SELECT.

Таблица 1.872 – Разряды регистра SELECT

Разряды	Значение по сбросу	Название	Тип доступа	Описание
31:4	0x000_0000	ADDR	ЗП	Разряды [31:4] адреса, по которому порт доступа AP должен осуществить запрос на чтение/запись. Разряды [3:2] задаются в регистра APACC
3:0	0x0	DPBANKSEL	ЗП	Адрес банка регистров контроллера SWJ-DP, в который будет осуществляться дальнейший доступ

1.3.16.1.4.7 Регистр DLCR

Данный регистр доступен только в режиме Serial-Wire и задает количество тактов для смены направления следования данных через вывод SWDIO. В таблице 1.873 приведёно описание полей регистра DLCR.

Таблица 1.873 – Разряды регистра DLCR

Разряды	Значение по сбросу	Название	Тип доступа	Описание
31:10	-	-	ЧТ0	Зарезервировано
9:8	0x0	TURNROUND	ЧТ/ЗП	Разряды [31:4] адреса, по которому порт доступа AP должен осуществить запрос на чтение/запись. Разряды [3:2] задаются в регистра APACC
7	-	-	ЧТ0	Зарезервировано
6	1	-	ЧТ1	Зарезервировано
[5:0]	-	-	ЧТ0	Зарезервировано

1.3.16.1.4.8 Регистр RDBUFF

Данный регистр доступен только в режиме Serial-Wire и содержит значение данных, прочитанных портом доступа AP, предыдущим запросом на чтение. В таблице 1.874 приведёно описание полей регистра RDBUFF.

Таблица 1.874 – Разряды регистра DLCR

Разряды	Значение по сбросу	Название	Тип доступа	Описание
31:0	0x0000_0000	RDATA	ЧТ	Данное поле содержит значение данных, прочитанных портом доступа AP, предыдущим запросом на чтение. Чтение из этого поля не приводит к выставлению портом доступа AP запросов

1.3.16.1.4.9 Регистр TARGETID

Регистр TARGETID содержит идентификатор контроллера отладочного порта SWJ-DP. В таблице 1.875 приведено описание полей регистра TARGETID.

Таблица 1.875 – Разряды регистра DLCR

Разряды	Значение по сбросу	Название	Тип доступа	Описание
31:0	0x7D80477	TARGETID	ЧТ	Идентификатор контроллера отладочного порта SWJ-DP

1.3.16.2 Корневые таблицы подсистемы отладки и трассировки

1.3.16.2.1 Главная корневая таблица подсистемы отладки и трассировки ROM Table

ROM Table — это корневая таблица конфигурации системы отладки и трассировки. Данная таблица представляет собой ПЗУ, в ячейках которого хранится информация о базовых

адресах функциональных компонентов системы отладки и трассировки, доступных данному порту доступа AP, а также базовые информация о базовых адресах таблиц конфигурации веток системы. В таблице 1.876 приведено содержимое таблицы ROM Table, доступное по соответствующим смещениям.

Таблица 1.876 – Содержимое таблицы ROM Table

Смещение	Считываемое значение	Описание
0x0000	0x0000_2003[1)]	Значение смещения базового адреса компонента Secure Debug Channel Ext относительно базового адреса ROM Table
0x0004	0x0000_4003[1)]	Значение смещения базового адреса компонента Core AHB-AP относительно базового адреса ROM Table
0x0008	0x0000_6003[1)]	Значение смещения базового адреса компонента System AHB-AP относительно базового адреса ROM Table
0x000C	0x0000_8003[1)]	Значение смещения базового адреса компонента Timestamp Generator относительно базового адреса ROM Table
0x0010	0x0000_0000	Конец таблицы базовых адресов
ARM-специфичные регистры идентификации		
0x0FD0	0x0000_0004	PIDR4
0x0FE0	0x0000_00D8	PIDR0
0x0FE4	0x0000_00B7	PIDR1
0x0FE8	0x0000_000B	PIDR2
0x0FEC	0x0000_0000	PIDR3
0x0FF0	0x0000_000D	CIDR0
0x0FF4	0x0000_0010	CIDR1
0x0FF8	0x0000_0005	CIDR2
0x0FFC	0x0000_00B1	CIDR3

[1)] Значение "0x3" в младших разрядах означает, что данный компонент представлен в системе. Разряды [31:12] базового адреса компонента соответствуют сумме разрядов [31:12] базового адреса самой таблицы ROM Table (0x0_0000) и разрядов [31:12] соответствующего считанного значения из таблицы. Разряды [11:0] базового адреса любого компонента = 0x000

Компонент CPUWAIT не отражён в таблице ROM Table, так как не имеет ARM-специфичных регистров идентификации.

1.3.16.2.2 Корневая таблица компонентов отладки и трассировки ядра Core ROM Table

В ячейках данной корневой таблицы хранится информация о базовых адресах функциональных компонентов системы отладки и трассировки ядра. Они доступны порту доступа Core AHB-AP. В таблице 1.877 приведено содержимое таблицы Core ROM Table, доступное по соответствующим смещениям.

Таблица 1.877 – Содержимое таблицы Core ROM Table

Смещение	Считываемое значение	Описание
0x000	0xFFF0_F003[1)]	Значение смещения базового адреса компонента SCS относительно базового адреса Core ROM Table
0x004	0xFFF0_2003[1)]	Значение смещения базового адреса компонента DWT относительно базового адреса Core ROM Table
0x008	0xFFF0_3003[1)]	Значение смещения базового адреса компонента BPU относительно базового адреса Core ROM Table
0x00C	0xFFF0_1003[1)]	Значение смещения базового адреса компонента ITM относительно базового адреса Core ROM Table
0x010	0xFFF4_1003[1)]	Значение смещения базового адреса порта TPIU относительно базового адреса Core ROM Table
0x014	0xFFF4_2003[1)]	Значение смещения базового адреса компонента ETM относительно базового адреса Core ROM Table
0x018	0xFFF4_3003[1)]	Значение смещения базового адреса компонента CTI относительно базового адреса Core ROM Table
0x01C	0xFFF4_4002	Компонент не представлен
0x020	0x0000_0000	Конец таблицы базовых адресов
ARM-специфичные регистры идентификации		
0xFD0	0x0000_0004	PIDR4
0xFE0	0x0000_0021	PIDR0
0xFE4	0x0000_00BD	PIDR1
0xFE8	0x0000_000B	PIDR2
0xFEC	0x0000_0000	PIDR3
0xFF0	0x0000_000D	CIDR0
0xFF4	0x0000_0090	CIDR1
0xFF8	0x0000_0005	CIDR2
0xFFC	0x0000_00B1	CIDR3
0xFBC	0x4770_2A04	DDEVARCH

[1)] Значение "0x3" в младших разрядах означает, что данный компонент представлен в системе. Разряды [31:12] базового адреса компонента соответствуют сумме разрядов [31:12] базового адреса самой таблицы Core ROM Table (0x0_0000) и разрядов [31:12] соответствующего считанного значения из таблицы. Разряды [11:0] базового адреса любого компонента = 0x000

1.3.16.3 Компоненты Secure Debug Channel

Secure Debug Channel (SDC) — это набор компонентов, обеспечивающий безопасный канал связи между внешним отладочным устройством, подключённым к JTAG/Serial-Wire порту и ядром микроконтроллера. Этот канал связи, приведённый на рисунке 1.198, предоставляет возможность, авторизированным с помощью сертификата пользователям, управлять доступом к компонентам микроконтроллера. Сертификат валидируется с помощью программы, исполняющейся ядром.

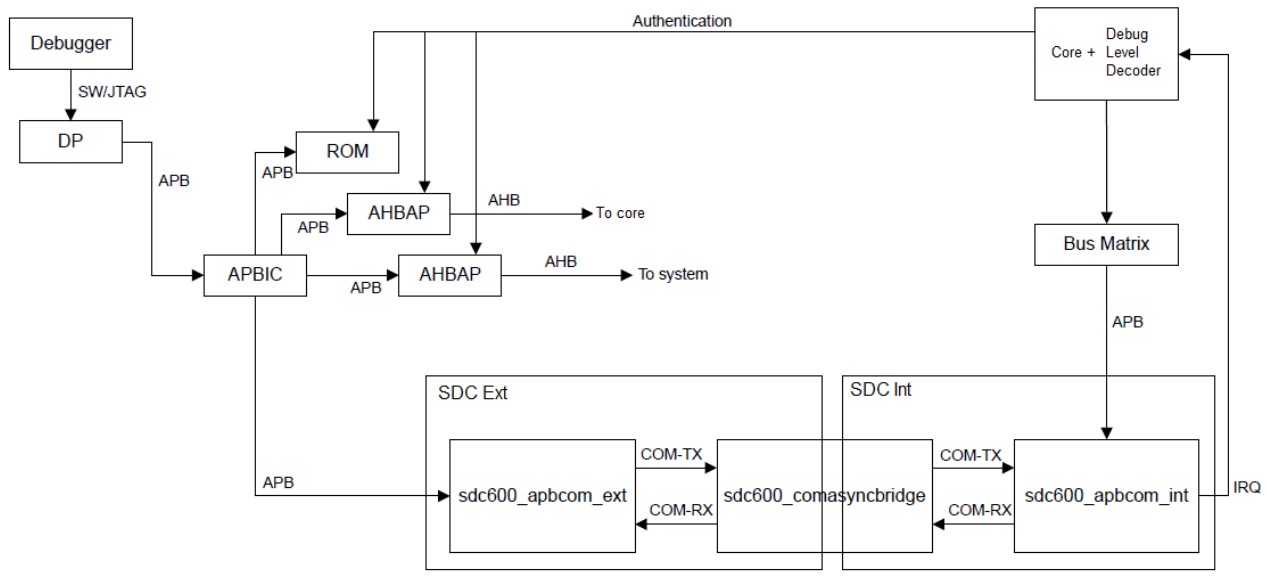


Рисунок 1.198 – Взаимодействие компонентов Secure Debug Channel

Работа канала обеспечивается двумя компонентами:

– SDC Ext;

– SDC Int.

Данные между ними передаются передаются в соответствии со спецификацией ARM Advanced Communications Channel (IHI 0076A).

Последовательность работы следующая:

1) Внешнее отладочное устройство с помощью контроллера SWJ-DP обращается к компоненту SDC Ext;

2) Компонент SDC Ext обращается к компоненту SDC Int с помощью байтового приёмопередающего интерфейса COM Wire Interface (CWI);

3) Компонент SDC Int вырабатывает прерывание IRQ;

4) Ядро начинает обработку прерывания, в котором принимает от SDC Int сертификат и валидирует его;

5) В зависимости от полученного сертификата ядро, с помощью обращений к блоку Debug Level Decoder, задаёт доступ к компонентам системы отладки и трассировки.

1.3.16.3.1 Программно доступные регистры компонентов SDC

В таблице 1.878 приведён список программно доступных регистров компонентов SDC Ext и SDC Int. Доступ к регистрам компонента SDC Ext осуществляется по шине Debug APB,

недоступной ядру. Доступ к регистрам компонента SDC Int осуществляется по шине APB, доступной ядру.

Таблица 1.878 – Программно доступные регистры компонентов SDC Ext и SDC Int

Смещение	Название	Тип доступа	Описание
0x0000	VIDR	ЧТ	Идентификатор компонента
0x0008	FIDTXR	ЧТ	Регистр передающей части
0x000C	FIDRXR	ЧТ	Регистр приёмной части
0x0010	ICSR	ЧТ/ЗП	Регистр прерываний
0x0020	DR	ЧТ/ЗП	Регистр данных
0x002C	SR[1)]	ЧТ/ЗП	Регистр статуса
0x0030	DBR	ЧТ/ЗП	Регистр данных
0x003C	SR[1)]	ЧТ/ЗП	Регистр статуса
[1)] Регистр SR доступен по двум смещениям			

1.3.16.3.1.1 Регистр VIDR

Данный регистр содержит информацию о версии протокола APBCOM Protocol и версии программно доступных регистров, использующихся компонентом SDC. В таблице 1.879 приведёно описание полей регистра VIDR.

Таблица 1.879 – Разряды регистра VIDR

Разряды	Значение по сбросу	Название	Тип доступа	Описание
31:8	-	-	ЧТ0	Зарезервировано
7:4	0x0	PROTVERSION	ЧТ	Значение "0x0" в данном поле означает, что используется версия 0 протокола APBCOM Protocol
3:0	0x0	PMVERSION	ЧТ	Значение "0x0" в данном поле означает, что используется версия 0 программно доступных регистров

1.3.16.3.1.2 Регистр FIDTXR

Данный регистр содержит информацию о передающей части компонентов SDC. В таблице 1.880 приведёно описание полей регистра FIDTXR.

Таблица 1.880 – Разряды регистра FIDTXR

Разряды	Значение по сбросу	Название	Тип доступа	Описание
31:20	-	-	ЧТ0	Зарезервировано

Разряды	Значение по сбросу	Название	Тип доступа	Описание
19:16	0x0	TXFD	ЧТ0	Значение "0x0" в данном поле означает, что глубина буфера (TxEngine FIFO) передачи данных через интерфейс CWI — 1 байт
15:11	-	-	ЧТ0	Зарезервировано
10	1	TXSZ32	ЧТ1	Поддерживаются 32-хразрядные записи по шине APB
9	0	TXSZ16	ЧТ0	Поддерживаются 16-разрядные записи по шине APB
8	0	TXSZ8	ЧТ0	Поддерживаются 16-разрядные записи по шине APB
7:4	0x1	TXW	ЧТ	Значение "0x0" в данном поле означает, что ширина передаваемых по интерфейсу CWI данных — 1 байт. При этом одна 32-хразрядная запись по шине APB вызывает одну передачу данных по интерфейсу CWI. При записи в регистр DR разряды [31:8] должны иметь значение 0xAFAFAF
3:2	-	-	ЧТ0	Зарезервировано
1	0 для компонента SDC Ext; 1 для компонента SDC Int	TXINT	ЧТ	0 — данный компонент может вырабатывать сигнал прерывания по передаче данных через интерфейс CWI 1 — данный компонент не может вырабатывать сигнал прерывания по передаче данных через интерфейс CWI
0	1	TXI	ЧТ1	Значение "1" в данном поле означает, данный компонент может передавать данные по интерфейсу CWI

1.3.16.3.1.3 Регистр FIDRXR

Данный регистр содержит информацию о принимающей части компонентов SDC. В таблице 1.881 приведёно описание полей регистра FIDRXR.

Таблица 1.881 – Разряды регистра FIDRXR

Разряды	Значение по сбросу	Название	Тип доступа	Описание
31:20	-	-	ЧТ0	Зарезервировано
19:16	0x0	RXFD	ЧТ0	Значение "0x0" в данном поле означает, что глубина буфера (RxEngine FIFO) приёма данных через интерфейс CWI — 1 байт
15:11	-	-	ЧТ0	Зарезервировано
10	1	RXSZ32	ЧТ1	Значение "1" в данном поле означает, что поддерживаются 32-хразрядные чтения по шине APB
9	0	RXSZ16	ЧТ0	Значение "0" в данном поле означает, что не поддерживаются 16-разрядные чтения по шине APB

Разряды	Значение по сбросу	Название	Тип доступа	Описание
8	0	RXSZ8	ЧТ0	Значение "0" в данном поле означает, что не поддерживаются 16-разрядные чтения по шине APB
7:4	0x1	RXW	ЧТ	Значение "0" в данном поле означает, что ширина принимаемых по интерфейсу CWI данных — 1 байт
3:2	-	-	ЧТ0	Зарезервировано
1	0 для компонента SDC Ext; 1 для компонента SDC Int	RXINT	ЧТ	0 — данный компонент может вырабатывать сигнал прерывания по приёму данных через интерфейс CWI 1 — данный компонент не может вырабатывать сигнал прерывания по приёму данных через интерфейс CWI
0	1	RXI	ЧТ1	Значение "1" в данном поле означает, данный компонент может принимать данные по интерфейсу CWI

1.3.16.3.1.4 Регистр ICSR

Данный регистр предназначен дял управления прерыванием компонента SDC. В таблице 1.882 приведёно описание полей регистра ICSR.

Таблица 1.882 – Разряды регистра ICSR

Разряды	Значение по сбросу	Название	Тип доступа	Описание
31	0x0	RXFIS	ЧТ0 для компонента SDC Ext; ЧТ/ЗП для компонента SDC Int	При чтении данное поле индицирует произошедшее прерывание по приёму данных через интерфейс CWI; При записи "1" данное поле сбрасывается в "0" при условии, что причина возникновения прерывания (заполнение буфера приёма) устранена
30:20	-	-	ЧТ0	Зарезервировано
19:16	0x0 для компонента SDC Ext; 0x1 для компонента SDC Int	RXFIL	ЧТ0 для компонента SDC Ext; ЧТ/ЗП для компонента SDC Int	Данный компонент вырабатывает прерывание по заполнении буфера приёма байтами в количестве, указанном в данном поле
15	0x0	TXFIS	ЧТ0 для компонента SDC Ext; ЧТ/ЗП для компонента SDC Int	При чтении данное поле индицирует произошедшее прерывание по передаче данных через интерфейс CWI; При записи "1" данное поле сбрасывается в "0" при условии, что причина возникновения прерывания (заполнение буфера передачи) устранена
14:4	-	-	ЧТ0	Зарезервировано
3:0	0x0 для компонента SDC Ext; 0x1 для компонента SDC Int	TXFIL	ЧТ0 для компонента SDC Ext; ЧТ/ЗП для компонента SDC Int	Данный компонент вырабатывает прерывание, по заполнении буфера передачи байтами в количестве, указанном в данном поле

1.3.16.3.1.5 Регистры DR и DBR

Данные регистры предназначены для отправки и приёма данных по интерфейсу CWI. Регистр DR функционально отличается от регистра DBR следующим:

Очередная запись в регистр DR в случае, если буфер принимающей стороны заполнен, приведёт к установке в "1" поля TXOE регистра SR с потерей данных, полученных этой записью. в то время как очередная запись в регистр DBR при тех же условиях будет ожидать опустошения буфера принимающей стороны.

В таблице 1.883 приведёно описание полей регистров DR и DBR.

Таблица 1.883 – Разряды регистров DR и DBR

Разряды	Значение по сбросу	Название	Тип доступа	Описание
31:0	0xAFAF_AFAF	DATA	ЧТ/ЗП	При записи в данное поле младший байт передаётся по интерфейсу CWI. Разряды [31:8] должны иметь значение 0xAFAFAF. В противном случае произойдёт следующее: для DR: ошибка переполнения — установка в "1" поля TXOE регистра SR; для DBR: ожидание опустошения буфера принимающей стороны; При чтении из данного поля возвращается значение байта, полученного по интерфейсу CWI. Разряды [31:8] будут иметь значение 0xAFAFAF

1.3.16.3.1.6 Регистр SR

Данный регистр индицирует текущее состояние компонента SDC. Доступен по двум смещениям. В таблице 1.884 приведёно описание полей регистра SR.

Таблица 1.884 – Разряды регистра SR

Разряды	Значение по сбросу	Название	Тип доступа	Описание
31	1	CFG_PEN	ЧТ1 для компонента SDC Int; ЧТ для компонента SDC Ext	0 — компонент недоступен; 1 — компонент доступен. Доступ определяется блоком Debug Level Decoder
30:24	-	-	ЧТ0	Зарезервировано
23:16	0x00	RXF	ЧТ	0x00 — нет данных, полученных по интерфейсу CWI; 0x01 — по интерфейсу CWI получено по крайней мере одно данное

Разряды	Значение по сбросу	Название	Тип доступа	Описание
15	0	TRINPROG		0 — нет незавершённых передач данных между компонентами SDC; 1 — очередная передача данных между компонентами SDC отменена сигналом DP_ABORT, но предыдущие ещё не завершены
14	0	TXLE	ЧТ/ЗП	Ошибка при передаче данных: При чтении: 0 — нет ошибки при передаче данных; 1 — произошла ошибки при передаче данных; При записи "1" данное поле сбрасывается в "0"
13	0	TXOE	ЧТ/ЗП	Переполнение буфера передачи данных: При чтении: 0 — нет переполнение буфера передачи данных; 1 — произошла попытка передачи данных (запись в регистр DR) при полном буфере передачи данных; При записи "1" данное поле сбрасывается в "0"
12	-	-	ЧТ1	Зарезервировано
11:8	-	-	ЧТ0	Зарезервировано
7:0	0x00	TXS	ЧТ	0x00 — буфер передачи данных заполнен; 0x01 — буфер передачи данных имеет по крайней мере один свободный байт

1.3.16.4 Генератор временных меток Timestamp Generator

Генератор временных меток Timestamp Generator (TSGen) — это компонент, генерирующий 64-хразрядные временные метки для согласования трассировочной информации. Счётчик временных меток тактируется отладочным тактовым сигналом Debug CLK.

1.3.16.4.1 Программно доступные регистры генератора TSGen

В таблице 1.885 приведён список программно доступных регистров генератора. Доступ данным регистрам осуществляется по шине Debug APB, недоступной ядру.

Таблица 1.885 – Программно доступные регистры генератора временных меток Timestamp Generator

Смешение	Название	Тип доступа	Описание
0x000	CNTCR	ЧТ/ЗП	Регистр управления счётчиком генератора
0x0008	CNTCVL	ЧТ/ЗП	Регистр младшей части счётчика
0x000C	CNTCVU	ЧТ/ЗП	Регистр старшей части счётчика
ARM-специфичные регистры идентификации			
0x0FD0	PIDR4	ЧТ	Значение: 0x0000_0000

Смещение	Название	Тип доступа	Описание
0x0FD4	PIDR5	ЧТ	Значение: 0x0000_0000
0x0FD8	PIDR6	ЧТ	Значение: 0x0000_0000
0x0FDC	PIDR7	ЧТ	Значение: 0x0000_0000
0x0FE0	PIDR0	ЧТ	Значение: 0x0000_0093
0x0FE4	PIDR1	ЧТ	Значение: 0x0000_00B1
0x0FE8	PIDR2	ЧТ	Значение: 0x0000_000B
0x0FEC	PIDR3	ЧТ	Значение: 0x0000_0000
0x0FF0	CIDR0	ЧТ	Значение: 0x0000_000D
0x0FF4	CIDR1	ЧТ	Значение: 0x0000_00F0
0x0FF8	CIDR2	ЧТ	Значение: 0x0000_0005
0x0FFC	CIDR3	ЧТ	Значение: 0x0000_00B1

1.3.16.4.1.1 Регистр CNTCR

Данный регистр позволяет управлять счётом генератора. В таблице 1.886 приведёно описание полей регистра CNTCR.

Таблица 1.886 – Разряды регистра CNTCR

Раз-ряды	Значение по сбросу	Название	Тип доступа	Описание
31:1	-	-	ЧТ0	Зарезервировано
0	0	EN	ЧТ/ЗП	0 — инкрементирование счётчика временных меток выключено; 1 — инкрементирование счётчика временных меток включено

1.3.16.4.1.2 Регистр CNTCVL

Данный регистр позволяет считывать и задавать значение младших 32-х разрядов счётчика временных меток. В таблице 1.887 приведено описание полей регистра CNTCVL.

Таблица 1.887 – Разряды регистра CNTCVL

Раз-ряды	Значение по сбросу	Название	Тип доступа	Описание
31:0	0x0000_0000	CNTCVL_L_32	ЧТ/ЗП	При чтении возвращается текущее значение младших 32-х разрядов счётчика временных меток; Значение, записанное в данное поле окажется в младших 32-х разрядах счётчика временных меток только после записи в регистр CNTCVU

1.3.16.4.1.3 Регистр CNTCVU

Данный регистр позволяет считывать и задавать значение старших 32-х разрядов счётчика временных меток. В таблице 1.888 приведено описание полей регистра CNTCVU.

Таблица 1.888 – Разряды регистра CNTCU

Разряды	Значение по сбросу	Название	Тип доступа	Описание
31:0	0x0000_0000	CNTCVU_U_32	ЧТ/ЗП	При чтении возвращается текущее значение старших 32-х разрядов счётчика временных меток; Запись в данное поле обновит старшие 32 разряда счётчика временных меток записанным значением, а также младшие 32 разряда счётчика временных меток значением, записанным до этого в регистр CNTCVL

1.3.16.5 Поты доступа Core AHB-AP и System AHB-AP

Порты доступа AHB-AP (AHP Access Port) — это компоненты, имеющие ведомый порт APB, по которому управляется ведущий AHB порт. Описание портов доступа AHB-AP и их программно доступных регистров приведено в документе Arm CoreSight SoC-600 Technical Reference Manual.

1.3.16.6 Компоненты отладки и трассировки ядра

В данном разделе приведена общая информация о компонентах отладки и трассировки ядра.

Описание компонентов CTI, BPU, DWT, ITM и SCS и их программно доступных регистров приведено в документе Arm Cortex-M33 Processor Technical Reference Manual.

Описание компонента ETM и его программно доступных регистров приведено в документе Arm CoreSight ETM-M33 Technical Reference Manual.

1.3.16.6.1 Компонент CTI

Подсистема отладки и трассировки оснащена системой обмена триггерными событиями (Embedded Cross Trigger). Обмен событиями осуществляется с помощью компонента CTI — Cross Trigger Interface. Данный компонент включает в себя интерфейс Trigger Interface, позволяющий обмениваться триггерными событиями с подключёнными с нему устройствами подсистемы отладки и трассировки. Данный интерфейс имеет входы, с помощью которых принимается информация о произошедших в подключённых устройствах событиях и выходы, по

которым подключённым устройствам передаётся информация о том, что произошло то или иное событие, которое связано с данным выходом. Связь устанавливается программно.

В таблице 1.889 приведены триггерные события, обмен которыми реализован в системе отладки и трассировки МК Optimal+.

Таблица 1.889 – Триггерные события подсистемы отладки и трассировки

Триггерное событие	**Направление события**
Ядро введено в режим отладки	Ядро → CTI
Совпадение по точке наблюдения 0	Ядро → CTI
Совпадение по точке наблюдения 1	Ядро → CTI
Совпадение по точке наблюдения 2	Ядро → CTI
Совпадение по точке наблюдения 3 или событие 0 из ETM	Ядро или ETM → CTI
Событие 1 из ETM	ETM → CTI
Запрос на ввод ядра в режим отладки	CTI → Ядро
Запрос на вывод ядра из режима отладки	CTI → Ядро
Запрос от CTI на прерывание 0	CTI → System
Запрос от CTI на прерывание 1	CTI → System
Событие 0 для ETM	CTI → ETM
Событие 1 для ETM	CTI → ETM
Событие 2 для ETM	CTI → ETM
Событие 3 для ETM	CTI → ETM

Более подробная информация приведена в документе ARM Cortex-M33 Processor Technical Reference Manual.

1.3.16.6.2 Компонент BPU

Breakpoint unit (BPU) — это устройство, позволяющее задавать аппаратные точки останова. Имеется возможность задавать до восьми аппаратных точек останова.

1.3.16.6.3 Компонент DWT

Компонент Data Watchpoint and Trace unit (DWT) позволяет задавать до четырёх аппаратных точек наблюдения и содержит ряд счётчиков, позволяющих собирать метрики об исполняемой ядром программе. Компонент DWT в связке с компонентом ITM позволяет генерировать трассировочную информацию о совпадении по точке наблюдения, о достижении каким-либо счётчиком предельного значения и о значении программного счётчика ядра (PC). Компонент DWT имеет интерфейс обмена триггерными событиями.

1.3.16.6.4 Компонент ITM

Компонент Instrumentation Trace Macrocell (ITM) генерирует трассировочную информацию в реальном времени с помощью дополнительного отладочного кода (printf style debugging), внедрённого в программу, исполняемую ядром. Трассировочная информация, генерируемая ITM, выдаётся по шине ATB.

1.3.16.6.5 Компонент SCS

Компонент System Control Space (SCS) позволяет внешнему отладочному устройству считывать и изменять состояние ядра, управлять им в режиме отладки.

1.3.16.7 Порт сбора трасс TPIU

Порт сбора трасс TPIU — это устройство, собирающее трассы из двух источников (ITM и ETM) по шинам ATB в одну объединённую трассу, которая подаётся на порт Trace Port. На рисунке 1.199 приведена схема порта TPIU. Выводы порта TPIU приведены в таблице 1.890.

Таблица 1.890 – Список выводов порта сбора трасс TPIU

Наименование вывода	Функциональное назначение
SWO	Выход последовательных данных порта сбора трасс в режиме Serial-Wire Output[1)]
[1)] Сигнал SWO может быть выведен на вывод JTDO/SWO (раздел 1.3.16.1) при условии SWOENABLE == 1	

Управление тактированием порта сбора трасс TPIU осуществляется полем TRACE_EN регистра RCU_PCER1 подсистемы управления сигналами сброса и синхросигналами RCU.

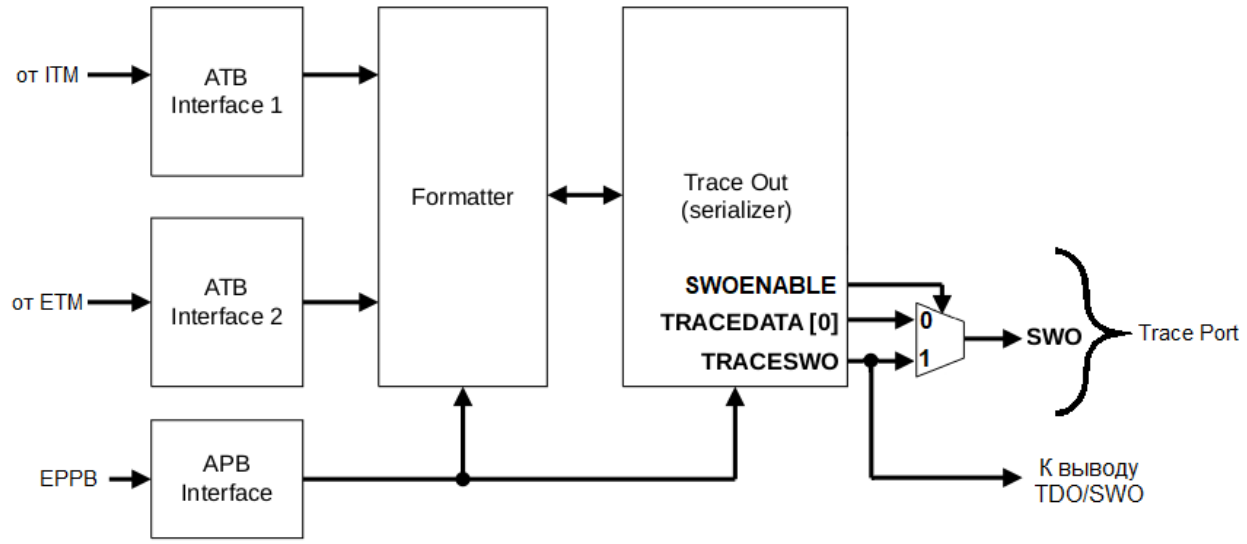


Рисунок 1.199 – Порт сбора трасс TPIU

Блок Formatter преобразует два потока трассировочной информации в один в соответствии с протоколом, описанным в Arm CoreSight Architecture Specification v2.0. Данное преобразование предполагает возможность определения источника трассировочной информации в преобразованной трассе за счёт пакетов синхронизации, генерация которых источниками трасс вызывается компонентом DWT или в соответствии со значением в регистре TPIU_PSCR.

Блок Trace Out преобразует трассировочную информацию в последовательность, которая подаётся на порт Trace Port.

Трассировочная информация может выдаваться только в режиме Serial-Wire Output (SWO).

– Регистр TPIU_DEVID содержит информацию о поддерживаемых форматах выдачи трассы;

– Регистр TPIU_SPPR определяет формат и режим выдачи трассы.

В режиме SWO трасса может собираться либо из двух источников (ITM и ETM), либо из одного — только из ITM. Это определяется режимом блока Formatter.

1.3.16.7.1 Программно доступные регистры порта сбора трасс TPIU

В таблице 1.891 приведён список программно доступных регистров порта сбора трасс TPIU.

Таблица 1.891 – Программно доступные регистры порта сбора трасс TPIU

Смешение	Название	Тип доступа	Описание
0x000	TPIU_SSPSR	ЧТ	Регистр идентификации параллельного порта Trace Port
0x004	TPIU_CSPSR	ЧТ/ЗП	Регистр конфигурации параллельного порта Trace Port
0x010	TPIU_ACPR	ЧТ/ЗП	Регистр управления делителем частоты выдачи данных в режиме Serial-Wire
0x0F0	TPIU_SPPR	ЧТ/ЗП	Регистр формата и режима выдачи трассы
0x300	TPIU_FFSR	ЧТ	Регистр статуса блока Formatter
0x304	TPIU_FFCR	ЧТ/ЗП	Регистр управления блоком Formatter
0x308	TPIU_PSCR	ЧТ/ЗП	Регистр управления пакетами синхронизации
0xFC8	DEVID	ЧТ	Регистр идентификации порта сбора трасс TPIU
0xFCC	DEVTYPE	ЧТ	Значение: 0x0000_0011
ARM-специфичные регистры идентификации			
0xFD0	PIDR4	ЧТ	Значение: 0x0000_0004
0xFD4	PIDR5	ЧТ	Значение: 0x0000_0000
0xFD8	PIDR6	ЧТ	Значение: 0x0000_0000
0xFDC	PIDR7	ЧТ	Значение: 0x0000_0000
0xFE0	PIDR0	ЧТ	Значение: 0x0000_0021

Смещение	Название	Тип доступа	Описание
0xFE4	PIDR1	ЧТ	Значение: 0x0000_00BD
0xFE8	PIDR2	ЧТ	Значение: 0x0000_000B
0xFEC	PIDR3	ЧТ	Значение: 0x0000_0000
0xFF0	CIDR0	ЧТ	Значение: 0x0000_000D
0xFF4	CIDR1	ЧТ	Значение: 0x0000_0090
0xFF8	CIDR2	ЧТ	Значение: 0x0000_0005
0xFFC	CIDR3	ЧТ	Значение: 0x0000_00B1

1.3.16.7.1.1 Регистр TPIU_SSPSR

Данный регистр содержит информацию о возможных конфигурациях порта Trace Port. В таблице 1.892 приведено описание полей регистра TPIU_SSPSR.

Таблица 1.892 – Разряды регистра TPIU_SSPSR

Раз-ряды	Значение по сбросу	Название	Тип доступа	Описание
31:0	0x0000_000B	SSPSR	ЧТ	"1" в N-ом разряде данного поля означает, что поддерживается выдача трассы N+1 выводами порта Trace Port, т.е. поддерживается выдача трассы 1, 2 и 4 выводами

1.3.16.7.1.2 Регистр TPIU_CSPSR

Данный регистр задаёт выбранную конфигурацию порта Trace Port. В таблице 1.893 приведено описание полей регистра TPIU_CSPSR.

Таблица 1.893 – Разряды регистра TPIU_CSPSR

Раз-ряды	Значение по сбросу	Название	Тип доступа	Описание
31:0	0x0000_0001	CSPSR	ЧТ/ЗП	"1" в N-ом разряде данного поля означает, что выбрана конфигурация порта Trace Port, при которой трасса выдаётся N+1 выводами порта. В "1" может быть установлен только один разряд поля из числа разрядов установленных в "1" в поле SSPSR регистра TPIU_SSPSR. Изменение данного поля запрещено во время работы порта Trace Port

1.3.16.7.1.3 Регистр TPIU_ACPR

Данный регистр задаёт частоту выдачи данных в режиме Serial-Wire. В таблице 1.894 приведено описание полей регистра TPIU_ACPR.

Таблица 1.894 – Разряды регистра TPIU_ACPR

Разряды	Значение по сбросу	Название	Тип доступа	Описание
31:13	-		ЧТ0	Зарезервировано
12:0	0x000	PRESCALER	ЧТ/ЗП	Частота F, с которой будет выдаваться трассировочная информация в режиме Serial-Wire: F = (частота сигнала Debug CLK)/(PRESCALER + 1)

1.3.16.7.1.4 Регистр TPIU_SPPR

Данный регистр определяет формат и режим выдачи трассы. В таблице 1.895 приведено описание полей регистра TPIU_SPPR.

Таблица 1.895 – Разряды регистра TPIU_SPPR

Разряды	Значение по сбросу	Название	Тип доступа	Описание
31:2	-	-	ЧТ0	Зарезервировано
2:0	0x1	TXMODE	ЧТ/ЗП	"0x0" — выдача трассы в параллельном режиме; "0x1" — выдача трассы в в асинхронном режиме Serial-Wire с использованием манчестерского кодирования; "0x2" — выдача трассы в в асинхронном режиме Serial-Wire в UART формате.

1.3.16.7.1.5 Регистр TPIU_FFSR

Данный регистр отражает статус блока Formatter. В таблице 1.896 приведено описание полей регистра TPIU_FFSR.

Таблица 1.896 – Разряды регистра TPIU_FFSR

Разряды	Значение по сбросу	Название	Тип доступа	Описание
31:4	-	-	ЧТ0	Зарезервировано
3	1	FtNonSTop	ЧТ1	1 — блок Formatter нельзя остановить
2:1	-	-	ЧТ0	Зарезервировано
0	0	FlInProg	ЧТ	0 — все трассировочные данные, в том числе оставшиеся у источников трасс отправлены через Trace Port; 1 — Flush in progress — трассировочные данные, оставшиеся у источников трасс запрошены (flush) — в ожидании отправки через Trace Port

1.3.16.7.1.6 Регистр TPIU_FFCR

Данный регистр предназначен для управления блоком Formatter. В таблице 1.897 приведено описание полей регистра TPIU_FFCR.

Таблица 1.897 – Разряды регистра TPIU_FFCR

Разряды	Значение по сбросу	Название	Тип доступа	Описание
31:9	-	-	ЧТ0	Зарезервировано
8	1	-	ЧТ1	Зарезервировано
7	-	-	ЧТ0	Зарезервировано
6	0	FOnMan	ЧТ/ЗП1	При записи: 1 — инициировать отправку оставшиеся у источников трасс трассировочных данных через Trace Port; При чтении: 0 — оставшиеся трассировочные данные, запрошенные у источников трасс записью "1" в данное поле, получены портом TPIU; 1 — оставшиеся трассировочные данные, запрошенные у источников трасс записью "1" в данное поле, не получены портом TPIU;
5:1	-	-	ЧТ0	Зарезервировано
1	0	EnFCount	ЧТ/ЗП	0 — Continuous mode — не включать триггерные пакеты (Trigger packet) в трассу 1 — включать триггерные пакеты (Trigger packet) в трассу
0	-	-	ЧТ0	Зарезервировано

1.3.16.7.1.7 TPIU_PSCR

Данный регистр задаёт счётчик периодической синхронизации, управляющий запросами пакетов синхронизации от источников трасс. Данный счётчик декрементируется всякий раз по выдаче очередного байта трассировочной информации портом Trace Port. По достижении счётчиком нуля, порт TPIU запрашивает у источников трасс пакет синхронизации, при этом счётчик принимает исходное значение.

В таблице 1.898 приведено описание полей регистра TPIU_PSCR.

Таблица 1.898 – Разряды регистра TPIU_PSCR

Разряды	Значение по сбросу	Название	Тип доступа	Описание
31:5	-	-	ЧТ0	Зарезервировано

Разряды	Значение по сбросу	Название	Тип доступа	Описание
4:0	0x00	PSCount	ЧТ/ЗП	При чтении возвращается записанное значение; При записи: "0x00" — "0x07": TPIU не будет запрашивать пакетов синхронизации; "0x08" — пакет синхронизации будет запрашиваться по выдаче портом Trace Port каждых 256 байтов; "0x09" — каждых 512 байтов; "0x0A" — каждых 1024 байта; "0x0B" — каждых 2048 байтов; "0x0C" — каждых 10^{12} байтов; "0x0D" — каждых 10^{13} байтов; "0x0E" — каждых 10^{14} байтов; "0x0F" — каждых 10^{15} байтов; "0x10" — каждых 10^{16} байтов; "0x11" — "0x1F": TPIU не будет запрашивать пакетов синхронизации

1.3.16.7.1.8 Регистр DEVID

Данный регистр содержит основную информацию о порте сбора трасс TPIU. В таблице 1.899 приведено описание полей регистра DEVID.

Таблица 1.899 – Разряды регистра DEVID

Разряды	Значение по сбросу	Название	Тип доступа	Описание
31:12	-	-	ЧТ0	Зарезервировано
11	1	NRZVALID	ЧТ1	1 — поддерживается выдача трассы в асинхронном режиме Serial-Wire в UART формате
10	1	MANCVALID	ЧТ1	1 — поддерживается выдача трассы в асинхронном режиме Serial-Wire с использованием манчестерского кодирования
9	1	PTINVALID	ЧТ1	1 — поддерживается параллельная выдача трассы через порт Trace Port
8:6	0x2	FIFOSZ	ЧТ	Данное поле отражает глубину выходного FIFO трасс: "0x2" — 4 байта
5:0	0x01	Trace inputs number	ЧТ	Данное поле отражает количество источников трасс: "0x1" — 2 источника

1.3.16.8 Блок управления системными сигналами Debug Sysregs

Блок Debug Sysregs управляет сигналом задержки загрузки ядра — сигналом CPUWAIT , а также набором сигналов для FMC.

Если сигнал CPUWAIT активен до завершения сброса ядра и продолжает быть активным после, ядро не приступит к загрузке. По сбросу сигнала CPUWAIT ядро приступит к

загрузке. Сигнал CPUWAIT может быть активирован программно; сброшен программно или по Power-ON сбросу.

1.3.16.8.1 Программно доступные регистры блока Debug Sysregs

В таблице 1.900 приведён список программно доступных регистров блока Debug Sysregs. Доступ данным регистрам осуществляется по шине Debug APB, недоступной ядру. Все программно доступные регистры блока Debug Sysregs принимают значение по сбросу по Power-ON сбросу.

Таблица 1.900 – Программно доступные регистры блока Debug Sysregs

Смешение	Название	Тип доступа	Описание
0x000	CPUWAIT_STAT	ЧТ	Регистр статуса сигнала CPUWAIT
0x004	CPUWAIT_SET	ЗП	Регистр активации сигнала CPUWAIT
0x008	CPUWAIT_RST	ЗП	Регистр сброса сигнала CPUWAIT
0x010	FMC_BIST_MUX_SEL	ЧТ/ЗП	Регистр управления источником системных сигналов для BIST-тестирования блок FMC
0x014	FMC_BIST	ЧТ/ЗП	Регистр управления системными сигналами для BIST-тестирования блок FMC
0x018	EFLASH_TM	ЧТ/ЗП	Регистр управления тестовым режимом FMC
0x020	FMC_AUTO_POWER_UP	ЧТ/ЗП	Регистр управления автоматической подачей питания на FMC
0x024	OPTMEM_FIRST_EN	ЧТ/ЗП	Регистр управления сигналом optmem_first_enable

1.3.16.8.1.1 Регистр CPUWAIT_STAT

Данный регистр отражает состояние сигнала CPUWAIT. В таблице 1.901 приведено описание полей регистра CPUWAIT_STAT.

Таблица 1.901 – Разряды регистра CPUWAIT_STAT

Раз-ряды	Значение по сбросу	Название	Тип доступа	Описание
31:1	-	-	ЧТ0	Зарезервировано
0	0	STATUS	ЧТ	0 — сигнал CPUWAIT сброшен; 1 — сигнал CPUWAIT активен

1.3.16.8.1.2 Регистр CPUWAIT_SET

Данный регистр предназначен для активации сигнала CPUWAIT. В таблице 1.902 приведено описание полей регистра CPUWAIT_SET.

Таблица 1.902 – Разряды регистра CPUWAIT_SET

Разряды	Значение по сбросу	Название	Тип доступа	Описание
31:1	-	-	ЧТ0	Зарезервировано
0	-	SET	ЗП1	По записи 1 в данное поле происходит активация сигнала CPUWAIT

1.3.16.8.1.3 Регистр CPUWAIT_RST

Данный регистр предназначен для сброса сигнала CPUWAIT. В таблице 1.903 приведено описание полей регистра CPUWAIT_RST.

Таблица 1.903 – Разряды регистра CPUWAIT_RST

Разряды	Значение по сбросу	Название	Тип доступа	Описание
31:1	-	-	ЧТ0	Зарезервировано
0	-	RST	ЗП1	По записи 1 в данное поле происходит сброс сигнала CPUWAIT

1.3.16.8.1.4 Регистр FMC_BIST_MUX_SEL

Данный регистр предназначен управления мультиплексором, который выбирает источник системных сигналов для BIST-тестирования блока FMC. Системные сигналы BIST-тестирования блока FMC приведены в описании регистра FMC_BIST. В таблице 1.904 приведено описание полей регистра FMC_BIST_MUX_SEL.

Таблица 1.904 – Разряды регистра FMC_BIST_MUX_SEL

Разряды	Значение по сбросу	Название	Тип доступа	Описание
31:1	-	-	ЧТ0	Зарезервировано
0	0	MUX_SEL	ЧТ/ЗП	0 — источником системных сигналов для BIST-тестирования блока FMC являются внешние выводы; 1 — источником системных сигналов для BIST-тестирования блока FMC является блок Debug Sysregs

1.3.16.8.1.5 Регистр FMC_BIST

Данный регистр предназначен для управления системными сигналами для BIST-тестирования блока FMC. В таблице 1.905 приведено описание полей регистра FMC_BIST.

Таблица 1.905 – Разряды регистра FMC_BIST

Разряды	Значение по сбросу	Название	Тип доступа	Описание
31:1	-	-	ЧТ0	Зарезервировано
7	-	-	ЧТ0	Зарезервировано
8	0	BIST_EFLASH_EN	ЧТ/ЗП	Управление питанием макросов eFlash-памяти
7:5	-	-	ЧТ0	Зарезервировано
4	0	BIST_BANK_SEL	ЧТ/ЗП	0 — выбран банк 0 FMC; 1 — выбран банк 1 FMC
3:1	-	-	ЧТ0	Зарезервировано
0	0	BIST_EN	ЧТ/ЗП	0 — выключен BIST режим FMC; 1 — включён BIST режим FMC

1.3.16.8.1.6 Регистр EFLASH_TM

Данный регистр предназначен для управления тестовым режимом блока FMC. В таблице 1.906 приведено описание полей регистра EFLASH_TM.

Таблица 1.906 – Разряды регистра EFLASH_TM

Разряды	Значение по сбросу	Название	Тип доступа	Описание
31:1	-	-	ЧТ0	Зарезервировано
0	0	EFLASH_TM	ЧТ/ЗП	0 — выключен тестовый режим FMC; 1 — включён тестовый режим FMC

1.3.16.8.1.7 Регистр FMC_AUTO_POWER_UP

Данный регистр предназначен управления автоматической подачей питания на банки FMC. В таблице 1.907 приведено описание полей регистра FMC_AUTO_POWER_UP.

Таблица 1.907 – Разряды регистра FMC_AUTO_POWER_UP

Разряды	Значение по сбросу	Название	Тип доступа	Описание
31:2	-	-	ЧТ0	Зарезервировано
1	1	AUTO_POWER_UP1	ЧТ/ЗП	0 — запрещена автоматическая подача питания на банк 1 FMC по снятии сброса; 1 — разрешена автоматическая подача питания на банк 1 FMC по снятии сброса
0	1	AUTO_POWER_UP0	ЧТ/ЗП	0 — запрещена автоматическая подача питания на банк 0 FMC по снятии сброса; 1 — разрешена автоматическая подача питания на банк 0 FMC по снятии сброса

1.3.16.8.1.8 Регистр OPTMEM_FIRST_EN

Данный регистр предназначен для управления тестовым режимом блока FMC. В таблице 1.908 приведено описание полей регистра OPTMEM_FIRST_EN.

Таблица 1.908 – Разряды регистра OPTMEM_FIRST_EN

Разряды	Значение по сбросу	Название	Тип доступа	Описание
31:1	-	-	ЧТ0	Зарезервировано
0	1	OPTMEM_FIRST_EN	ЧТ/ЗП	0 — неактивен сигнал optmem_first_enable; 1 — активен сигнал optmem_first_enable

1.3.16.9 Блок Debug Logic Decoder

Блок Debug Logic Decoder состоит из нескольких регистров, доступ к которым осуществляется по системной шине. Данные регистры управляют сигналами аутентификации ядра, а также сигналами аутентификации, которые разрешают, ограничивают или запрещают доступ портов AHB-AP. Сброс регистров происходит по Power-ON сбросу.

1.3.16.9.1 Программно доступные регистры блока Debug Logic Decoder

В таблице 1.909 приведён список программно доступных регистров блока Debug Logic Decoder.

Таблица 1.909 – Программно доступные регистры блока Debug Logic Decoder

Смещение	Название	Тип доступа	Описание
0x000	DBG_AUTH	ЧТ/ЗП	Регистр управления сигналами аутентификации компонентов подсистемы отладки и трассировки
0x008	CORE_AUTH	ЧТ/ЗП	Регистр управления сигналами аутентификации ядра

1.3.16.9.1.1 Регистр DBG_AUTH

Данный регистр предназначен управления сигналами аутентификации компонентов подсистемы отладки и трассировки. В таблице 1.910 приведено описание полей регистра DBG_AUTH.

Таблица 1.910 – Разряды регистра DBG_AUTH

Разряды	Значение по сбросу	Название	Тип доступа	Описание
31:1	-	-	ЧТ0	Зарезервировано
6	1	AP_CORE_EN	ЧТ/ЗП	0 — запрещён доступ порта Core AHB-AP к шине AHB ядра; 1 — разрешён доступ порта Core AHB-AP к шине AHB ядра

Раз-ряды	Значение по сбросу	Название	Тип доступа	Описание
5	1	AP_CORE_SECURE_EN	ЧТ/ЗП	0 — запрещён доступ порта System AHB-AP к системной шине AHB; 1 — разрешён доступ порта System AHB-AP к системной шине AHB
4	1	AP_SYS_EN	ЧТ/ЗП	0 — запрещён Secure-доступ порта Core AHB-AP к шине AHB ядра; 1 — разрешён Secure-доступ порта Core AHB-AP к шине AHB ядра
3	1	AP_SYS_EN	ЧТ/ЗП	0 — запрещён Secure-доступ порта System AHB-AP к системной шине AHB; 1 — разрешён Secure-доступ порта System AHB-AP к системной шине AHB
2	1	CFG_PEN	ЧТ/ЗП	0 — компонент SDC Ext доступен; 1 — компонент SDC Ext недоступен
1:0	-	-	ЧТ0	Зарезервировано

1.3.16.9.1.2 Регистр CORE_AUTH

Данный регистр предназначен управления сигналами аутентификации ядра В таблице 1.911 приведено описание полей регистра DBG_AUTH.

Таблица 1.911 – Разряды регистра CORE_AUTH

Раз-ряды	Значение по сбросу	Название	Тип доступа	Описание
31:4	-	-	ЧТ0	Зарезервировано
3	1	CPU_SPNIDEN	ЧТ/ЗП	0 — Secure Non-invasive debug prohibited; 1 — Secure Non-invasive debug allowed
2	1	CPU_SPIDEN	ЧТ/ЗП	0 — Secure halting and self-hosted debug prohibited; 1 — Secure halting and self-hosted debug allowed
1	1	CPU_NIDEN	ЧТ/ЗП	0 — Non-invasive debug prohibited; 1 — Non-invasive debug allowed
0	1	CPU_DBGEN	ЧТ/ЗП	0 — Invasive debug prohibited; 1 — Invasive debug allowed

1.3.17 Режимы загрузки МК Optimal+

МК Optimal+ поддерживает два основных режима загрузки:

– незащищённый, предназначенный для использования в конечных изделиях, в которых не требуется обеспечение безопасности от действий потенциальных злоумышленников по несанкционированному доступу к загружаемому во флеш-память пользовательскому ПО и аппаратным ресурсам МК Optimal+;

– защищённый, предназначенный для использования в конечных изделиях, в которых не требуется обеспечение безопасности от действий потенциальных злоумышленников по несанкционированному доступу к загружаемому во флеш-память пользовательскому ПО и аппаратным ресурсам МК Optimal+;

Основными отличиями защищённого режима от незащищённого являются:

– при старте микросхемы включены компоненты TrustZone подсистемы безопасности;

– поддержка обеспечения единой точки старта за счёт фиксации адреса первого обращения процессорного ядра при помощи значения бита BOOTLOCK, расположенного в энергонезависимой памяти параметров (далее по тексту - OptMem, см.раздел 1.3.5.1.2.10).

Выбор одного из двух этих режимов выполняется при помощи значения поля TZEN, расположенного в энергонезависимой памяти параметров OptMem.

Каждый режим, через прораммирование определённых полей в энергонезависимой памяти параметров OptMem, настроить по два стартовых адреса для каждого режима: * поля NSBOOTADDR0 и NSBOOTADDR1 для незащищённого режима; * поля SECBOOTADDR0 и SECBOOTADDR1 для защищённого режима.

Значение полей NSBOOTADDR0, NSBOOTADDR1, SECBOOTADDR0 и SECBOOTADDR1 полностью определяется пользователем при программировании памяти параметров OptMem, но рекомендуемое использование данных полей следующее:

– поля NSBOOTADDR0 и SECBOOTADDR0 рекомендуется использовать для определение стартового адреса пользовательского ПО, реализующего функциональность конечного изделия;

– поля NSBOOTADDR0 и SECBOOTADDR0 рекомендуется использовать для определение стартового адреса системного загрузчика, позволяющего проводить обновление пользовательскго ПО через штатные интерфейсные контроллеры МК Optimal+.

Выбор между двумя стартовыми адресами для каждого режима выполняется на основе значения конфигурационного бита BOOT0. Значение данного бита, в зависимости от значения поля BOOT0SRC из энергонезависимой памяти параметров OptMem, определяется либо выводом микросхемы (см пп 1.2.2), либо полем BOOT0VAL из энергонезависимой памяти параметров OptMem.

Сводная информация по режмима загрузки представлена в таблице 1.912.

Таблица 1.912 – Режимы загрузки МК Optimal+

Режим загрузки	**OptMem [TZEN]**	**OptMem [BOOTLOCK] 2)**	**BOOT0 1)**	**Адрес старта процессорного ядра**	**Значение по умолчанию**
Незащищённый		-		Определяется полем OptMem[NSBOOTADDR0]	FMC, bank 0 (MAIN): 0x00000000
		-	1	Определяется полем OptMem[NSBOOTADDR1]	FMC, bank 1,Info_1 (bootloader): 0x00108000
Защищённый	1			Определяется полем OptMem[SECBOOTADDR0]	FMC, bank 0 (MAIN): 0x10000000
			1	Определяется полем OptMem[SECBOOTADDR1]	FMC, bank 1,Info_1 (bootloader): 0x10108000
		1	-	Определяется полем OptMem[SECBOOTADDR0]	FMC, bank 0 (MAIN): 0x10000000
Примечания:					
1) Источник значения конфигурационного бита BOOT0 определяется значением бита OptMem[NSBOOT0SRC]: - если OptMem[NSBOOT0SRC] = 0, то значение BOOT0 определяется значением, захватываемым в момент снятия активного уровня сигнала системного сброса с вывода согласно (см.п.1.2.2); - если OptMem[NSBOOT0SRC] = 1, то значение BOOT0 определяется битом OptMem[BOOT0VAL] .					
2) Установка OptMem[BOOTLOCK] блокирует перезапись как самого этого бита, так и поля OptMem[SECBOOTADDR0]					

1.4 Маркировка и пломбирование

Маркировка на корпусе микросхемы интегральной К1879ВГ1Т представлена на рисунке 1.200

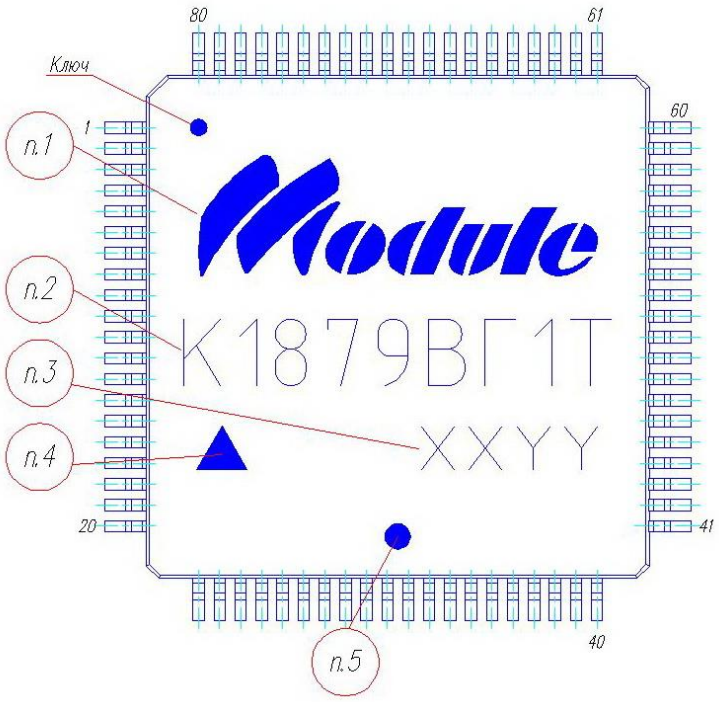


Рисунок 1.200 – Маркировка на корпусе микросхемы интегральной К1879ВГ1Т

В процессе изготовления на корпусе микросхемы интегральной К1879ВГ1Т наносится следующая маркировка:

1) товарный знак АО НТЦ «Модуль»;

2) условное обозначение;

3) дата изготовления, где первые две цифры – год (последние две цифры года), последующие две цифры - календарная неделя;

4) знак чувствительности к статическому электричеству;

5) клеймо ОТК в виде точки. Цвет краски – белый.

Пломбирование микросхем интегральных не осуществляется.

1.5 Упаковка

Поставка микросхем должна производиться в упаковке, предохраняющей микросхемы от механических повреждений при транспортировании и погрузочно-разгрузочных работах, а также для защиты от внешних климатических факторов при транспортировании и хранении.

Для микросхем разработана штатная групповая упаковка.

Микросхемы устанавливаются в ложементы размером 12х12мм паллет TRAY, выполненных из антистатического материала укладываются в антистатический пакет с силикагелем с герметизацией методом откачки воздуха в процессе заварки швов. Пакет с упакованными

микросхемами укладывается в коробку N683. Для осушки воздуха внутри коробки помещен мешок с осушителем – силикагелем. Коробки с двух сторон пломбируются.

В транспортную тару вкладывается упаковочный документ с необходимой информацией по упаковываемым микросхемам. Разработанная упаковка допускает переупаковывание после частичного изъятия микросхем из упаковки. Переупаковывание микросхем с паллетами и силикагелем производится с использованием пакета антистатического Advantek Drylock 2300P-1618 (40,6, 4x4572 см) или аналогичного по свойствам с герметизацией методом откачки воздуха в процессе заварки швов пакета.

Упаковка микросхем представлена на рисунке 1.201.

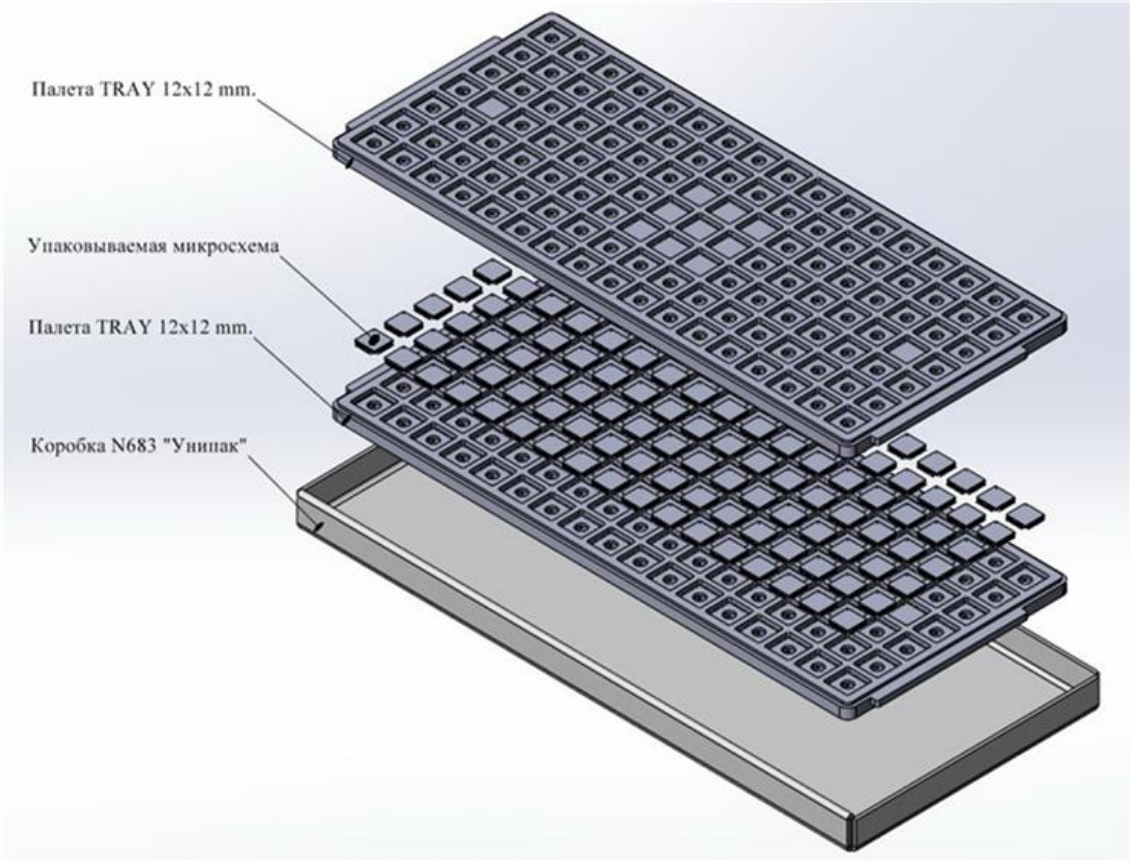


Рисунок 1.201 – Упаковка микросхем (крышка коробки и антистатический пакет не показаны)

В соответствии с п. 2.6 ГОСТ 23216-78 на тару нанесены манипуляционные знаки. Вся маркировка выполнена с помощью шильдиков, изготовленных по специальной технологии, что обеспечивает её сохранность при хранении и транспортировании микросхем. Упаковка и маркировка микросхем соответствует требованиям ГОСТ 23088 и ГОСТ 18725.

Микросхема в штатной упаковке допускает транспортирование всеми видами транспортных средств, при условии выполнения требований, указанных на манипуляционных

знаках, нанесенных на транспортную тару. Пример нанесения указанных знаков приведен на рисунке 1.202.

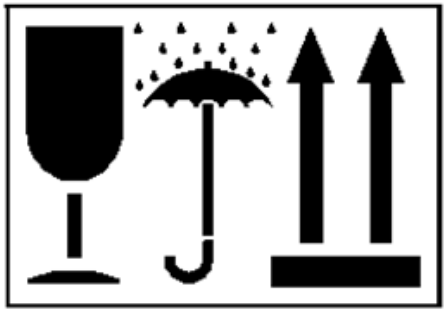


Рисунок 1.202 – Пример нанесения манипуляционных знаков на упаковку

2 Использование по назначению

2.1 Эксплуатационные ограничения

2.1.1 Эксплуатационные ограничения по температуре окружающей среды

МК Optimal+ предназначен для эксплуатации при следующих значениях температуры окружающей среды:

– повышенная температура окружаюшей среды, рабочая – плюс 105 ºС;

– пониженная температура окружаюшей среды, рабочая – минус 40 ºС.

2.1.2 Эксплуатационные ограничения по электрическим характеристикам

Уровень напряжения электропитания, уровни напряжений входных сигналов МК Optimal+ должны быть в пределах, указанных в пп. 1.2.4.

2.2 Подготовка изделия к использованию

2.2.1 Указания о взаимосвязи (соединении) данного изделия с другими изделиями

МК Optimal+ предназначен для функционирования в составе широкого спектра электронных изделий таких как приборы учета (счётчики), устойства кузовной автомобильной электроники, IoT-устройства, носимые персональные устройства, индустриальные сенсоры и пр. В составе данных электронных изделий МК Optimal+ соеиняется с широким спектром пассивной и активной ЭКБ, в том числе ЭКБ функционального назначения (интерфейсные микросхемы физического уровня, другие микроконтроллеры и т.п.). Типовая схема включения МК Optimal+ представлена на рисунке 2.1. Функциональные выводы МК Optimal+, использование которых не требуется для применения в конкретном электронном изделии, допускается оставлять неподключёнными.

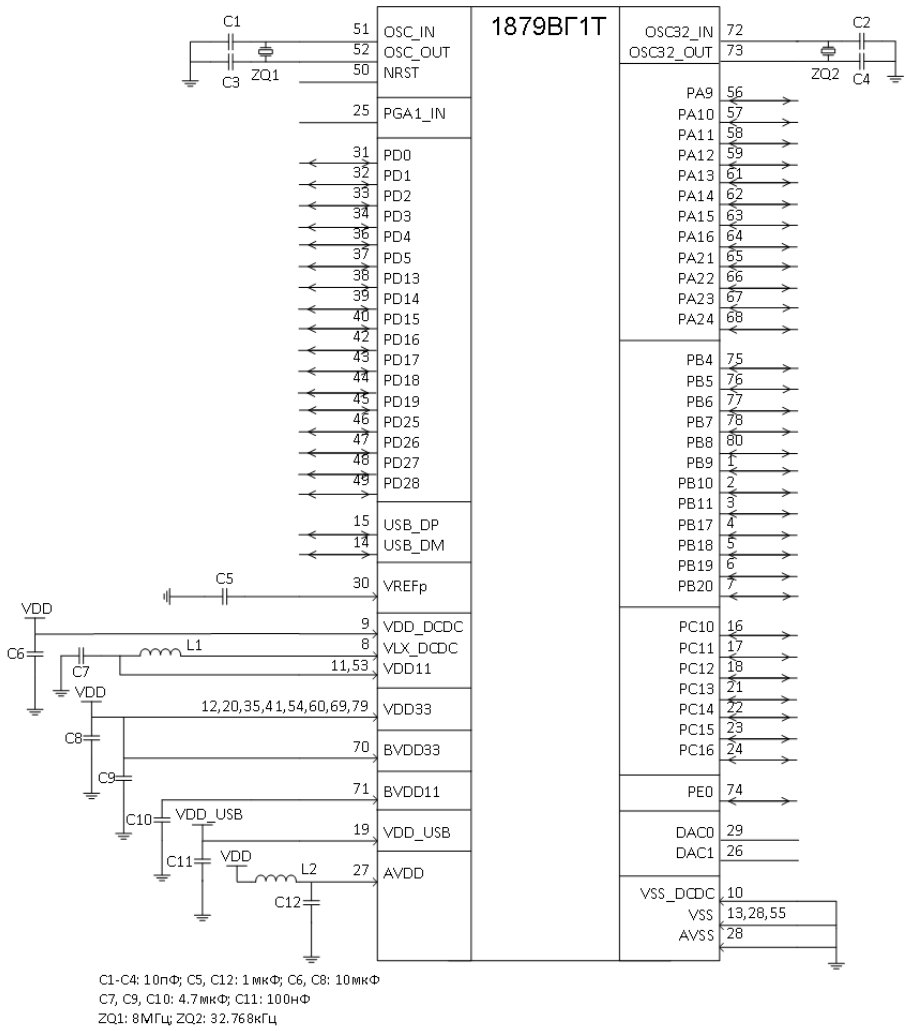


Рисунок 2.1 – Типовая схема включенния МК Optimal+

2.2.2 Указания по включению и опробованию работы изделия

Включение МК Optimal+ производится путём:

1) подачи напряжения питания на вывод BVDD33, с последующей одновременной подачей напряжения питания на выводы VDD, VDD_DCDC, AVDD, VDD_USB. Также допускается одновременная подача напряжений питания на выводы BVDD33, VDD, VDD_DCDC, AVDD и VDD_USB;

2) подачи комбинации уровней логического "0" или "1" на конфигурационные выводов, определяющие режим работы (см. пп. 1.3.2) и режим загрузки микросхемы (см. пп. 1.3.17)

Последовательность подачи остальных сигналов не регламентируется.

Опробование работы изделия проводится согласно пп. 2.3.1.

2.3 Использование изделия

2.3.1 Порядок контроля работоспособности изделия

При первом запуске изделие считается работоспособным, если, после выполнения действий, перечисленных в пп. 2.2.2, пользователь смог выполнить следующие дейсвтия:

1) подключиться к МК Optimal+ через отладочный интерфейс (см.пп.1.3.16);

2) выполнить запись пользовательского ПО во флеш-память МК Optimal+;

3) выполнить конфигурирование МК Optimal+ через запись полей памяти параметров OptMem (см. пп.1.3.5.1.2.10);

4) выполнить перезагрузку МК Optimal+ путем кратковременного отключения питания или подачи активного уровня на вход NRTS_PON;

5) убедится в корректном исполнении записанного во флеш-память пользовательского ПО (критерии корректности полностью определяются самим пользователем).

2.3.2 Перечень возможных неисправностей в процессе использования изделия по назначению и рекомендации по действиям при их возникновении

В процессе функционирования МК Optimal+ могут возникать неисправности двух типов:

– устранимые;

– неустранимые.

К устранимым относятся неисправности, которые могут быть исправлены непосредственно в процессе работы МК Optimal+ или полной перезагрузкой МК Optimal+ путем кратковременного отключения питания или подачи активного уровня на вход NRTS_PON. К таким неисправностям можно отнести одиночные и двойные ошибки флеш-памяти или ОЗУ, детекируемые и исправляемые при помощи аппаратных схем детектирования результате некорректной работы ПО, выполняющегося на центральном процессоре МК Optimal+ или взаимодействующего с ним через внешние интерфейсы или как результат воздействия внешних факторов, не приводящих к перманентному выходу из строя узлов микросхемы.

К неустранимым относятся неисправности, заключающиеся в перманентном выходе из строя микросхемы вцелом или её отдельных узлов. К таким несиправностям можно отнести выход из строя значительного количества ячеек флеш-памяти из-за исчерпания ресурса из записи, выход из строя отдельных узлов в силу превышения предельных значений электрических

параметров (например, встроенных источников вторичного питания, блоков аналоговой подсистмы и др.) В этом случае восстановить полноценную работоспособность МК Optimal+ не представляется возможным и микросхема подлежит замене.

2.3.3 Перечень режимов работы изделия, а также характеристики основных режимов работы

Перечень основных режимов работы МК Optimal+ приведён пп.1.3.2 Функциональный режим работы предусматривает функционирование МК Optimal+ в нескольких режимов энергопотребления, перечень которых приведен в пп. 1.3.8.2.

2.3.4 Порядок и правила перевода изделия с одного режима работы на другой с указанием необходимого для этого времени

Порядок и правила переключения основных режимов работы МК Optimal+ приведён пп.1.3.2 Порядок и правила переключения режимов энергопотребления при работе в функциональном режиме работы МК Optimal+ приведен в пп. 1.3.8.2.

2.3.5 Порядок приведения изделия в исходное положение

МК Optimal+ приводится в исходное положение путём выполнения процедуры полного стирания содержимого флеш-памяти и памяти параметров (OptMem).

2.3.6 Порядок выключения изделия, содержание и последовательность осмотра изделия после окончания работы

Выключение МК Optimal+ выполняется путем отключения электропитания, подаваемом на обозначенные в пп. 2.2.2 выводы.

3 Техническое обслуживание изделия

Изделие является необслуживаемым.

4 Текущий ремонт изделия

Изделие неремонтопригодно. В случае неработоспособности изделия в составе устройства, ремонт устройства может осуществляться путем замены изделия, если подобная операция допускается документацией на устройство.

5 Хранение изделия

Хранение МК Optimal+ должно осуществляться по ГОСТ 18725.

6 Транспортирование изделия

Транспортирование МК Optimal+ должно осуществляться в соответсвии с ГОСТ 18725 с учётом следующих дополнений и уточнений:

– транспортирование микросхем должно осуществляться в упакованном виде всеми видами транспорта, кроме морского;

– перевозку воздушным траспортом следует осуществлять в герметичных отсеках;

– транспортирование микросхем следует осуществлять в крытых транспортных средствах;

– размещение и крепление транспортной тары с упакованными микросхемами в транспортных средствах должно обеспечивать их устойчивое положение и не допускать пе-ремещения во время транспортировки.

7 Утилизация изделия

Микросхемы после снятия с эксплуатации подлежат утилизации в порядке и методами, устанавливаемыми в контракте на поставку.

Лист регистрации изменений									
Изм.	Номера листов (страниц)				Всего листов (страниц) в докум.	№ документа	Входящий № сопроводи-тельного до-кум. и дата	Подп.	Дата
	измене-нных	замене-нных	новых	аннулиро-ванных					