



Нейропроцессор NM6403

**Введение  
в архитектуру**  
Версия 1.1b

---

---

## Содержание

---

---

1 Общие характеристики и функциональные особенности .....	4
2 Структура нейропроцессора .....	6
2.1 Основные узлы и шины нейропроцессора .....	6
2.2 RISC-ядро .....	11
2.3 Векторный сопроцессор .....	17
2.4 Программируемые интерфейсы с внешними шинами .....	21
2.5 Коммуникационные порты ввода/вывода .....	23
3 Организация памяти .....	26
4 Система прерываний .....	27
5 Краткое описание системы команд .....	28
5.1 Форматы команд .....	28
5.2 Методы адресации данных .....	30
5.3 Методы модификации адресных регистров .....	32
5.4 Методы модификации РС при переходах .....	33
5.5 Анализируемое сочетание признаков для команд управления .....	34
5.6 Программно доступные регистры .....	35
5.7 Арифметико-логические операции, задаваемые в скалярной команде .....	37
5.8 Арифметико-логические операции, задаваемые в векторной команде .....	38
6 Инициализация нейропроцессора после системного сброса .....	40
6.1 Инициализация нейропроцессора по глобальной шине .....	40
6.2 Инициализация нейропроцессора по коммуникационному порту 1 .....	40

---

# 1 Общие характеристики и функциональные особенности

---

NM6403 представляет собой высокопроизводительный микропроцессор со статической суперскалярной архитектурой. В его состав входят устройства управления, вычисления адреса и обработки скаляров, а также узел для поддержки операций над векторами с элементами переменной разрядности. Кроме того, имеются два идентичных программируемых интерфейса для работы с внешней памятью различного типа, а также два коммуникационных порта, аппаратно совместимых с портами ЦПС TMS320C4x, для возможности построения многопроцессорных систем.

## **Основные характеристики нейропроцессора**

- тактовая частота - 50 МГц (длительность процессорного такта - 20 нс);
- число эквивалентных вентилях - 115.000;
- технология 0,5 мкм;
- корпус 256BGA;
- напряжение питания от 3.0 В до 3.6В;
- адресное пространство - 16 Гбайт;
- формат скалярных и векторных данных:
  - ◆ 32-разрядные скалярные данные,
  - ◆ вектора с элементами переменной разрядности от 1 до 64, упакованные в 64-разрядные блоки данных;
- аппаратная поддержка операций умножения вектора на матрицу или матрицы на матрицу;
- аппаратная реализация функции насыщения с программируемым порогом насыщения;
- два устройства генерации адресов данных;
- регистры:
  - ◆ восемь 32-разрядных регистров общего назначения,
  - ◆ восемь 32-разрядных адресных регистров,
  - ◆ три блока внутренней памяти по 32\*64 бит ,
  - ◆ специальные регистры управления и состояния;
- команды нейропроцессора 32- и 64-разрядные (одна команда обычно задаёт две операции);
- два 64-разрядных программируемых интерфейса для работы с любым типом внешней памяти. Каждый интерфейс поддерживает обмен с

---

двумя банками памяти разного типа (статическая или динамическая память);

- два скоростных байтовых коммуникационных порта ввода/вывода, аппаратно совместимых с портами TMS320C4х.

### **Применение**

- акселераторы для PC и рабочих станций:
  - ◆ эмуляция нейронных сетей,
  - ◆ сигнальная обработка,
  - ◆ аппаратная поддержка векторно-матричных операций;
- основной блок для построения больших суперпараллельных вычислительных систем и реализации нейросетевых технологий.

### **Производительность**

- скалярные операции:
  - ◆ 50 MIPS,
  - ◆ 200 MOPS для 32-разрядных данных;
- векторные операции:
  - ◆ 1.200.000.000 умножений и сложений в секунду (при перемножении матриц с 8-разрядными элементами);
- внешний интерфейс:
  - ◆ пропускная способность каждого 64-разрядного интерфейса с внешней памятью - до 400 Мбайт/с,
  - ◆ темп обмена по каждому коммуникационному порту ввода/вывода - до 20 Мбайт/с.

### 2.1 Основные узлы и шины нейропроцессора

Нейропроцессор предназначен для обработки 32-разрядных скалярных данных и данных программируемой разрядности, упакованных в 64-разрядные слова, которые в дальнейшем будут называться векторами упакованных данных. Структурная схема нейропроцессора представлена на рис.1. Функциональное назначение внешних выводов нейропроцессора приведено в табл.1.

Основными узлами нейропроцессора являются:

- **RISC CORE** - центральный процессорный узел, предназначенный для выполнения операций сдвига и арифметико-логических операций над 32-разрядными скалярными данными, формирования 32-разрядных адресов команд и данных при обращениях к внешней памяти и выполнения всех основных функций по управлению работой нейропроцессора.
- **VCP** - векторный сопроцессор, предназначенный для выполнения арифметических и логических операций над 64-разрядными векторами данных программируемой разрядности.
- **LMI** и **GMI** - два одинаковых блока программируемого интерфейса с локальной и глобальной 64-разрядными внешними шинами, к каждой из которых может быть подключена внешняя память, содержащая до  $2^{31}$  32-разрядных ячеек. Обмен данными с внешней памятью может осуществляться как 32-разрядными, так и 64-разрядными словами. В последнем случае нейропроцессор одновременно выбирает две соседних ячейки памяти. Адресация внешней памяти осуществляется страничным способом, при котором на одну и ту же внешнюю 15-разрядную адресную шину в режиме разделения времени выдаются как младшие, так и старшие разряды адреса. Причем старшие разряды адреса выдаются только при переходе к выборке новой страницы памяти. Каждый блок программируемого интерфейса обеспечивает эффективную работу нейропроцессора с двумя банками внешней памяти различного объема, различного типа и различного быстродействия без использования дополнительного оборудования. В данных блоках предусмотрена аппаратная поддержка режима разделяемой памяти для различных мультипроцессорных конфигураций внешних шин.
- **CP0** и **CP1** - два идентичных коммуникационных порта, каждый из которых обеспечивает обмен информацией по двунаправленному байтовому линку между нейропроцессором и его абонентом, предназначен для построения высокопроизводительных мультипроцессорных систем на основе нейропроцессоров и полностью совместим с коммуникационным портом сигнального процессора TMS320C4x фирмы Texas Instruments. Каждый коммуникационный порт

---

имеет встроенный контроллер ПДП, позволяющий обмениваться 64-разрядными данными с внешней памятью, подключенной к локальной и (или) глобальной внешним шинам.

Нейропроцессор имеет пять внутренних шин, через которые осуществляется обмен информацией между его основными узлами:

- **LOCAL ADDRESS BUS** и **GLOBAL ADDRESS BUS** служат для пересылки адресов команд, сформированных RISC-ядром, и адресов данных, сформированных RISC-ядром в программном режиме или коммуникационными портами в режиме ПДП, в соответствующие блоки программируемого интерфейса при обращении к внешней памяти.
- **OUTPUT DATA BUS** служит для пересылки данных, подлежащих записи в локальную или глобальную внешнюю память, из RISC-ядра, векторного сопроцессора и коммуникационных портов в блоки программируемого интерфейса;
- **INPUT BUS #1** и **INPUT BUS #2** предназначены для пересылки данных и команд, считанных из локальной или глобальной внешней памяти, из блоков программируемого интерфейса в любой из основных узлов нейропроцессора. Причем в программном режиме работы нейропроцессора пересылка скалярных данных осуществляется только по шине **INPUT BUS #2**, а пересылка векторных данных – только по шине **INPUT BUS #1**. Пересылка данных в режиме ПДП и пересылка команд могут осуществляться по любой из этих внутренних шин.

Межрегистровые пересылки скалярных данных и пересылки констант из регистра команд в программно доступные регистры осуществляются через блоки программируемого интерфейса с использованием внутренних шин **OUTPUT DATA BUS** и **INPUT BUS #2**.

64-разрядные шины **INPUT BUS #1**, **INPUT BUS #2** и **OUTPUT DATA BUS** позволяют за один такт пересылать как 32-разрядные, так и 64-разрядные слова данных. Поэтому с целью уменьшения числа выполняемых операций пересылок типа “регистр-регистр” и “память-регистр” большинство 32-разрядных регистров нейропроцессора могут программно объединяться в регистровые пары при выполнении данных операций. Кроме того, нейропроцессор содержит несколько 64-разрядных управляющих регистров, что так же позволяет говорить о выполнении операций пересылок над 64-разрядными скалярными данными.

Выборка команд из внешней памяти осуществляется 64-разрядными словами, каждое из которых представляет собой одну 64-разрядную команду или две 32-разрядных команды.

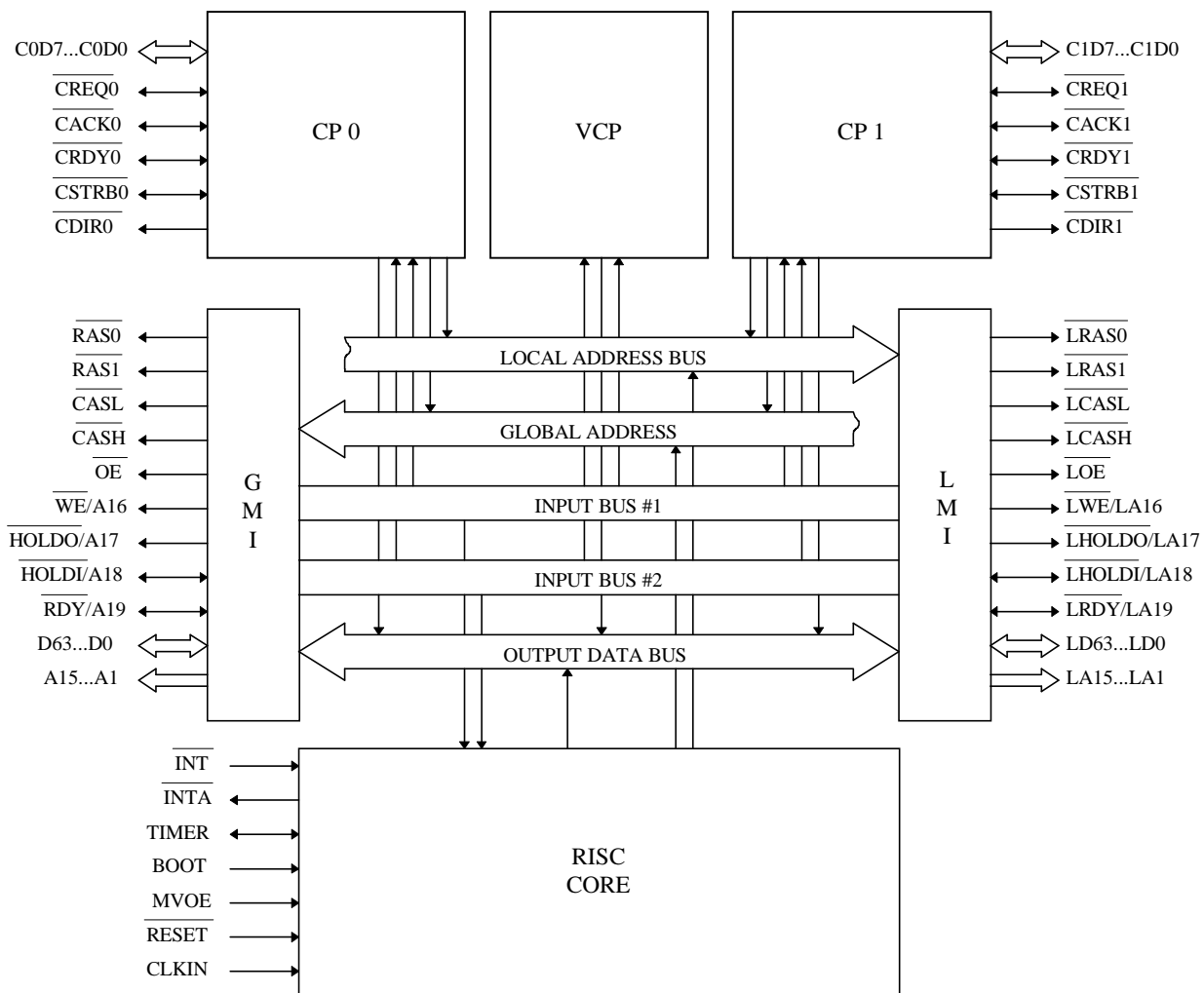


Рис. 2-1. Общая структура нейропроцессора



**Таблица 2-1. Функциональные выводы нейропроцессора**

Сигнал <sup>1)</sup>	Кол-во	Тип <sup>2)</sup>	Функциональное назначение
<b>Интерфейс с глобальной (локальной) шиной <sup>3)</sup></b>			
(L)D63 – (L)D0	64	I/O	64-разрядная шина данных
(L)A15 – (L)A1	15	O(Z)	15-разрядная шина адресов
$\overline{(L)RAS0}$ / $\overline{(L)CS0}$ <sup>4)</sup>	1	O(Z)	строб адреса строки 0-го банка динамической памяти / выборка 0-го банка статической памяти
$\overline{(L)RAS1}$ / $\overline{(L)CS1}$ <sup>4)</sup>	1	O(Z)	строб адреса строки 1-го банка динамической памяти / выборка 1-го банка статической памяти
$\overline{(L)CASL}$ / $\overline{(L)WEL}$ <sup>4)</sup>	1	O(Z)	строб адреса столбца 32 младших разрядов динамической памяти / разрешение записи в 32 младших разряда статической памяти
$\overline{(L)CASH}$ / $\overline{(L)WEN}$ <sup>4)</sup>	1	O(Z)	строб адреса столбца 32 старших разрядов динамической памяти / разрешение записи в 32 старших разряда статической памяти
$\overline{(L)OE}$	1	O(Z)	разрешение выдачи данных из внешней памяти
$\overline{(L)WE}$ / (L)A16 <sup>4)</sup>	1	O(Z)	признак цикла (запись/чтение) для динамической памяти / 16-разряд адреса статической памяти
$\overline{(L)HOLD0}$ / (L)A17 <sup>5)</sup>	1	O(Z)	запрос нейропроцессора на захват шины / 17-й разряд шины адресов
$\overline{(L)HOLD1}$ / (L)A18 <sup>5)</sup>	1	I/O	запрос от внешнего устройства на захват шины / 18-й разряд шины адресов
$\overline{(L)RDY}$ / (L)A19 <sup>5)</sup>	1	I/O	сигнал готовности / 19-й разряд шины адресов
<b>Коммуникационный порт <math>x</math> (<math>x=0,1</math>) <sup>6)</sup></b>			
CxD7 - CxD0	8	I/O	8-разрядная шина данных
$\overline{CREQx}$	1	I/O	запрос шины
$\overline{CACKx}$	1	I/O	подтверждение запроса шины
$\overline{CSTRBx}$	1	I/O	строб данных
$\overline{CRDYx}$	1	I/O	сигнал готовности данных
$\overline{CDIRx}$	1	O	направление передачи данных
<b>Общее управление</b>			
CLK	1	I	тактовый сигнал
RESET	1	I	сброс
$\overline{INT}$	1	I	внешнее прерывание
$\overline{INTA}$	1	O	подтверждение внешнего прерывания
BOOTM <sup>7)</sup>	1	I	режим начальной загрузки (0 – загрузка из глобальной памяти, 1 – загрузка через коммуникационный порт CP1)
TIMER <sup>8)</sup>	1	I/O	выход программируемых таймеров

Примечания: 1) Для выводов со знаком инверсии активным является низкий уровень сигнала.

2) I - вход,

O - выход,

O(Z) - выход с высокоимпедансным состоянием,

I/O – двунаправленный вывод.

3) По функциональному составу внешних выводов интерфейсы с глобальной и локальной шинами идентичны. В именах сигналов интерфейса с локальной шиной добавлен префикс “L”.

- 
- 4) Функциональное назначение выходов  $\overline{(\text{L})RAS0}/\overline{(\text{L})CS0}$ ,  $\overline{(\text{L})RAS1}/\overline{(\text{L})CS1}$ ,  $\overline{(\text{L})CASL}/\overline{(\text{L})WEL}$ ,  $\overline{(\text{L})CASH}/\overline{(\text{L})WEN}$  и  $\overline{(\text{L})WE}/(\text{L})A16$  зависит от типа памяти (динамическая/статическая), к которой обращается нейропроцессор в текущем цикле чтения/записи.
  - 5) Выводы  $\overline{(\text{L})HOLDO}/(\text{L})A17$ ,  $\overline{(\text{L})HOLDI}/(\text{L})A18$  и  $\overline{(\text{L})RDY}/(\text{L})A19$  при многопроцессорной конфигурации внешней шины используются для поддержки асинхронного протокола передачи управления шиной, а при однопроцессорной конфигурации шины – для выдачи разрядов адреса с 17-го по 19-й. В последнем случае вывод  $\overline{(\text{L})RDY}/(\text{L})A19$  может использоваться и для приема внешнего сигнала готовности при обращении к банку памяти, для работы с которым нейропроцессор запрограммирован на асинхронный способ обмена.
  - 6) По функциональному составу внешних выводов коммуникационные порты 0 и 1 идентичны.
  - 7) Вход  $\text{BOOTM}$  должен быть подключен к шине земли или питания в зависимости от используемого режима инициализации нейропроцессора.
  - 8) В процессе начальной загрузки нейропроцессора из глобальной памяти ( $\text{BOOTM}=0$ ) вывод  $\text{TIMER}$  работает в режиме входа, задающего тип 1-го банка памяти на глобальной шине. При реализации данного банка на микросхемах статической памяти к данному выводу должен быть подключен резистор pull-down, а при использовании динамической памяти – резистор pull-up.

## 2.2 RISC-ядро

Блок-схема RISC-ядра нейропроцессора представлена на рис.2. В его состав входят следующие устройства:

- **RALU** - регистровое АЛУ;
- **DAG1** и **DAG2** - первый и второй генераторы адресов данных;
- **PROGRAMM SEQUENCER** - генератор адресов команд;
- **CONTROL UNIT** - блок управления.

Для пересылки информации между блоками, входящими в состав перечисленных выше устройств, в RISC-ядре нейропроцессора имеется ряд внутренних шин:

- шины первого и второго операндов АЛУ,
- шина результата арифметико-логической операции или операции сдвига,
- шины первого и второго операндов АУ1,
- шины первого и второго операндов АУ2,
- входная шина счетчика команд РС.

Все блоки и внутренние шины RISC-ядра нейропроцессора являются 32-разрядными.

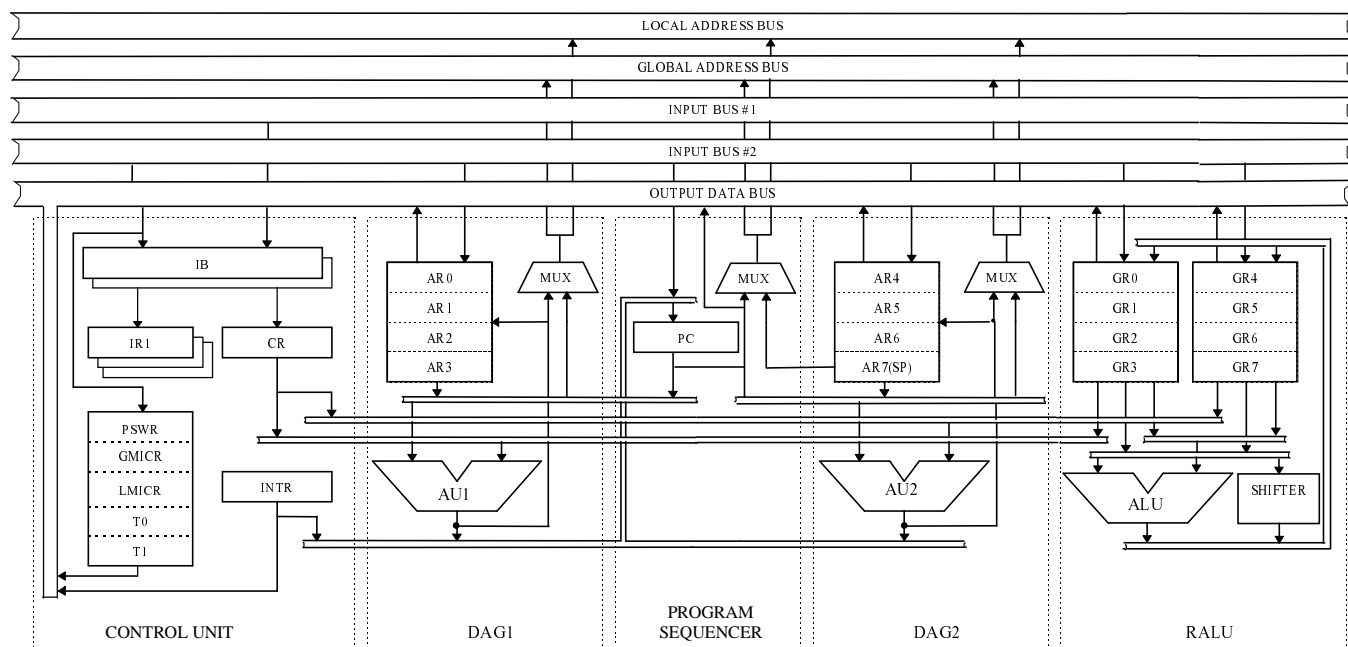


Рис. 2-2. Структурная схема RISC-ядра нейропроцессора

---

## 2.2.1 Регистровое АЛУ

Регистровое АЛУ (RALU) служит для оперативного хранения до восьми 32-разрядных скалярных данных и выполнения над ними операций сдвига, одно- и двухоперандных арифметических и логических операций. При выполнении операций в RALU формируются признаки нулевого результата, отрицательного результата, переноса и арифметического переполнения, которые могут фиксироваться в регистре слова состояния процессора и использоваться при выполнении условных команд управления. Данные, хранящиеся в RALU, могут также использоваться в качестве адресов или смещений адресов при выполнении ряда команд обращения к памяти и команд управления.

RALU включает в себя следующие блоки:

**GR0, ... , GR7** - регистры общего назначения, образующие регистровый файл, который имеет два входных порта, подключенных соответственно к шине *INPUT BUS #2* нейропроцессора и шине результата арифметико-логической операции или операции сдвига, и три выходных порта, подключенных соответственно к шине *OUTPUT DATA BUS* нейропроцессора и шинам первого и второго операндов ALU. Кроме того, регистры GR0, ... , GR3 имеют выходной порт, подключенный к шине второго операнда AU1, что позволяет использовать содержимое данных регистров при формировании адресов команд и данных с помощью DAG1 или при выполнении операций по модификации содержимого адресных регистров, входящих в состав DAG1. Регистры GR4, ... , GR7 имеют выходной порт, подключенный к шине второго операнда AU2, что позволяет использовать содержимое данных регистров при формировании адресов команд и данных с помощью DAG2 или при выполнении операций по модификации содержимого адресных регистров, входящих в состав DAG2.

**ALU** - арифметико-логическое устройство, выполняющее за один такт одну из шестнадцати логических или одиннадцати арифметических операций над содержимым одного или двух любых регистров общего назначения. Причем арифметические операции выполняются над данными, представленными в дополнительном коде.

**SHIFTER** – сдвигатель, выполняющий за один такт циклический, логический или арифметический сдвиг на любое число разрядов вправо или влево содержимого любого регистра общего назначения, выдаваемого на шину первого операнда ALU.

## 2.2.2 Первый генератор адресов данных

Первый генератор адресов данных (DAG1) служит для формирования адресов данных при выполнении команд обращения к памяти и адресов переходов при выполнении команд управления. Кроме того, DAG1 обеспечивает оперативное хранение и модификацию до

---

четырёх 32-разрядных адресов данных, адресов переходов или смещений адресов переходов.

DAG1 включает в себя следующие блоки:

**AR0, ... , AR3** - адресные регистры, образующие регистровый файл, который имеет два входных порта, подключенных соответственно к шине *INPUT BUS #2* нейропроцессора и выходам AU1, и два выходных порта, подключенных соответственно к шине *OUTPUT DATA BUS* нейропроцессора и шине первого операнда AU1.

**AU1** – первое арифметическое устройство для выполнения одно- или двухоперандных арифметических операций при вычислении адреса или модификации содержимого одного из регистров AR0, ..., AR3. В качестве первого операнда может использоваться содержимое одного из регистров AR0, ..., AR3 или PC, а в качестве второго – содержимое одного из регистров GR0 - GR3 или 32-разрядная константа из кода команды.

**MUX** - мультиплексор для выдачи на одну из внутренних адресных шин нейропроцессора информации с выходов AU1 или с шины первого операнда AU1 в зависимости от выполняемого метода адресации. При нулевом значении старшего разряда адреса, сформированного на выходах данного мультиплексора, выдача адреса из DAG1 будет осуществляться на *LOCAL ADDRESS BUS*, а при единичном - на *GLOBAL ADDRESS BUS*.

### 2.2.3 Второй генератор адресов данных

По своей структуре и выполняемым функциям второй генератор адресов данных (DAG2) аналогичен DAG1. Специфика DAG2 заключается в том, что один из его адресных регистров AR7(SP) дополнительно выполняет функции системного указателя стека при обработке прерываний и выполнении команд перехода к подпрограмме и возврата из подпрограммы или прерывания.

DAG2 включает в себя следующие блоки:

**AR4, ... , AR7(SP)** - адресные регистры, образующие регистровый файл, который имеет два входных порта, подключенных соответственно к шине *INPUT BUS #2* нейропроцессора и выходам AU2, и два выходных порта, подключенных соответственно к шине *OUTPUT DATA BUS* нейропроцессора и шине первого операнда AU2.

**AU2** – первое арифметическое устройство для выполнения одно- или двухоперандных арифметических операций при вычислении адреса или модификации содержимого одного из регистров AR4, ..., AR7. В качестве первого операнда может использоваться содержимое одного из регистров AR4, ..., AR7 или PC, а в качестве второго – содержимое одного из регистров GR4 – GR7 или 32-разрядная константа из кода команды.

**MUX** - мультиплексор для выдачи на одну из внутренних адресных шин нейропроцессора информации с выходов AU2 или с шины первого

---

операнда AU2 в зависимости от выполняемого метода адресации. При нулевом значении старшего разряда адреса, сформированного на выходах данного мультиплексора, выдача адреса из DAG1 будет осуществляться на *LOCAL ADDRESS BUS*, а при единичном - на *GLOBAL ADDRESS BUS*.

#### 2.2.4 Генератор адресов команд

Генератор адресов команд (PROGRAMM SEQUENCER) служит для формирования адреса очередной 64-разрядной команды или очередной пары 32-разрядных команд на линейных участках программы, когда вычисление адреса каждой следующей команды осуществляется путем инкремента адреса текущей команды. Кроме того, PROGRAMM SEQUENCER используется для формирования адрес-векторов прерываний и для формирования адреса системного стека при выполнении команд перехода к подпрограмме.

PROGRAMM SEQUENCER включает в себя следующие блоки:

**PC** - счетчик команд, предназначенный для хранения адреса текущей выбираемой из памяти 64-разрядной команды (или пары 32-разрядных команд) и вычисления адреса следующей команды (или пары команд) путем увеличения его содержимого на два. При выполнении команд перехода новое содержимое поступает в PC с выходов AU1 или AU2, при переходах к подпрограммам обработки прерываний – с выхода регистра INTR, а при выполнении команд возврата из подпрограммы или прерывания – с шины *INPUT BUS #2* нейропроцессора. Выходы PC подключены к шине *OUTPUT DATA BUS* нейропроцессора, что делает данный регистр программно доступным по чтению.

**MUX** - мультиплексор для выдачи на одну из внутренних адресных шин нейропроцессора увеличенного на два содержимого счетчика команд PC или содержимого указателя стека AR7(SP). При нулевом значении старшего разряда адреса, сформированного на выходах данного мультиплексора, выдача адреса из PROGRAMM SEQUENCER будет осуществляться на *LOCAL ADDRESS BUS*, а при единичном - на *GLOBAL ADDRESS BUS*.

#### 2.2.5 Блок управления

Блок управления (CONTROL UNIT) выполняет предварительный анализ и дешифрацию команд, выбранных из внешней памяти, формирует сигналы управления всеми узлами нейропроцессора в процессе конвейерного выполнения команд, обрабатывает все запросы на внутренние и внешние прерывания, осуществляет арбитраж поступающих от различных узлов нейропроцессора запросов на использование его внутренних и внешних шин, формирует внешние сигналы  $\overline{INTA}$  и TIMER.

CONTROL UNIT включает в себя следующие блоки:

---

**IB** - буфер команд, обеспечивающий хранение и предварительный анализ до четырех 32-разрядных или до двух 64-разрядных команд, выбранных из внешней памяти.

**IR1, ... , IR6** - регистры команд на различных ступенях конвейера выполнения команд.

**CR** – регистр для хранения константы, содержащейся в текущей выполняемой команде.

**GMICR** - регистр управления интерфейсом с глобальной шиной, содержимое которого определяет конфигурацию глобальной шины, устанавливает адресные границы между первым и вторым банками памяти, подключенными к глобальной шине, и устанавливает для каждого банка памяти размеры адресной страницы, режим работы (синхронный/асинхронный), тип используемых микросхем памяти, и длительности отдельных фаз в циклах обращения к памяти.

**LMICR** - регистр управления интерфейсом с локальной шиной выполняет те же функции, что и GMICR, но в отношении LMI.

**T0, T1** – программируемые таймеры, предназначенные для формирования через программируемые интервалы времени сигналов прерываний и сигналов на выходе TIMER нейропроцессора. Режим работы каждого таймера (однократный или периодический) задается соответствующими разрядами регистра PSWR. Причем, если хотя бы в одном из регистров LMICR или GMICR прописана работа с банком динамической памяти, то T1 будет использоваться в качестве счетчика, определяющего период регенерации динамической памяти.

**PSWR** - регистр слова состояния процессора, предназначенный для хранения признаков результата последней выполненной в RALU операции, масок всех обрабатываемых прерываний и управляющей информации, задающей режимы работы таймеров T0, T1 и внешнего вывода TIMER, режим инициализации обмена по коммуникационным портам CP0 и CP1, останов каналов приема и передачи коммуникационных портов CP0 и CP1, очистку AFIFO и WFIFO, запрет на передачу управления локальной и глобальной шинами внешнему устройству. Данный регистр и счетчик команд PC образуют регистровую пару, содержимое которой запоминается в системном стеке при выполнении команд перехода к подпрограмме и при переходе к подпрограмме обслуживания прерывания.

**INTR** - регистр запросов на прерывания и ПДП, который служит для хранения всех запросов на ПДП, поступивших от коммуникационных портов CP0 и CP1, и всех запросов на внутренние и внешнее прерывания до момента начала их обслуживания. Кроме того, в данном регистре хранится информация о состоянии AFIFO и WFIFO (пустое, полное), о состоянии каждого из коммуникационных портов CP0 и CP1 (прием/передача), о количестве 64-разрядных слов, хранящихся в AFIFO и RAM, и о принадлежности локальной и глобальной шин нейропроцессору в данный момент времени. Регистр INTR, выходы

---

которого подключены к шине *OUTPUT DATA BUS* нейропроцессора, программно доступен только по чтению.

Регистры PSWR, GMICR, LMICR, T0 и T1 программно доступны как по чтению, так и по записи. Их входы подключены к шине *INPUT BUS #2*, а выходы - к шине *OUTPUT DATA BUS* нейропроцессора. Кроме того аппаратные средства нейропроцессора поддерживают побитовую установку и побитовый сброс регистра PSWR в программном режиме.



---

## 2.3 Векторный сопроцессор

Векторный сопроцессор (VCP) является основным вычислительным узлом нейропроцессора. VCP ориентирован на обработку данных произвольной разрядности от 1 до 64 разрядов, упакованных в 64-разрядные слова. Структурная схема VCP представлена на рис. 2-3.

Основными узлами VCP являются:

**OU** - операционное устройство;

**RCS** - циклический сдвигатель вправо;

**SU1, SU2** - узлы, аппаратно реализующие функцию насыщения;

**F1CR, F2CR** - регистры управления функцией насыщения;

**SWITCH 3→2** - коммутатор 3 в 2;

**WBUF** и **WOPER** - память весовых коэффициентов и операционная память весовых коэффициентов;

**WFIFO** - FIFO весовых коэффициентов;

**AFIFO** - накопительное FIFO;

**RAM** - векторный регистр;

**VR** - регистр порогов;

**SB1** и **SB2** - регистр границ синапсов и операционный регистр границ синапсов;

**NB1** и **NB2** - регистр границ нейронов и операционный регистр границ нейронов.

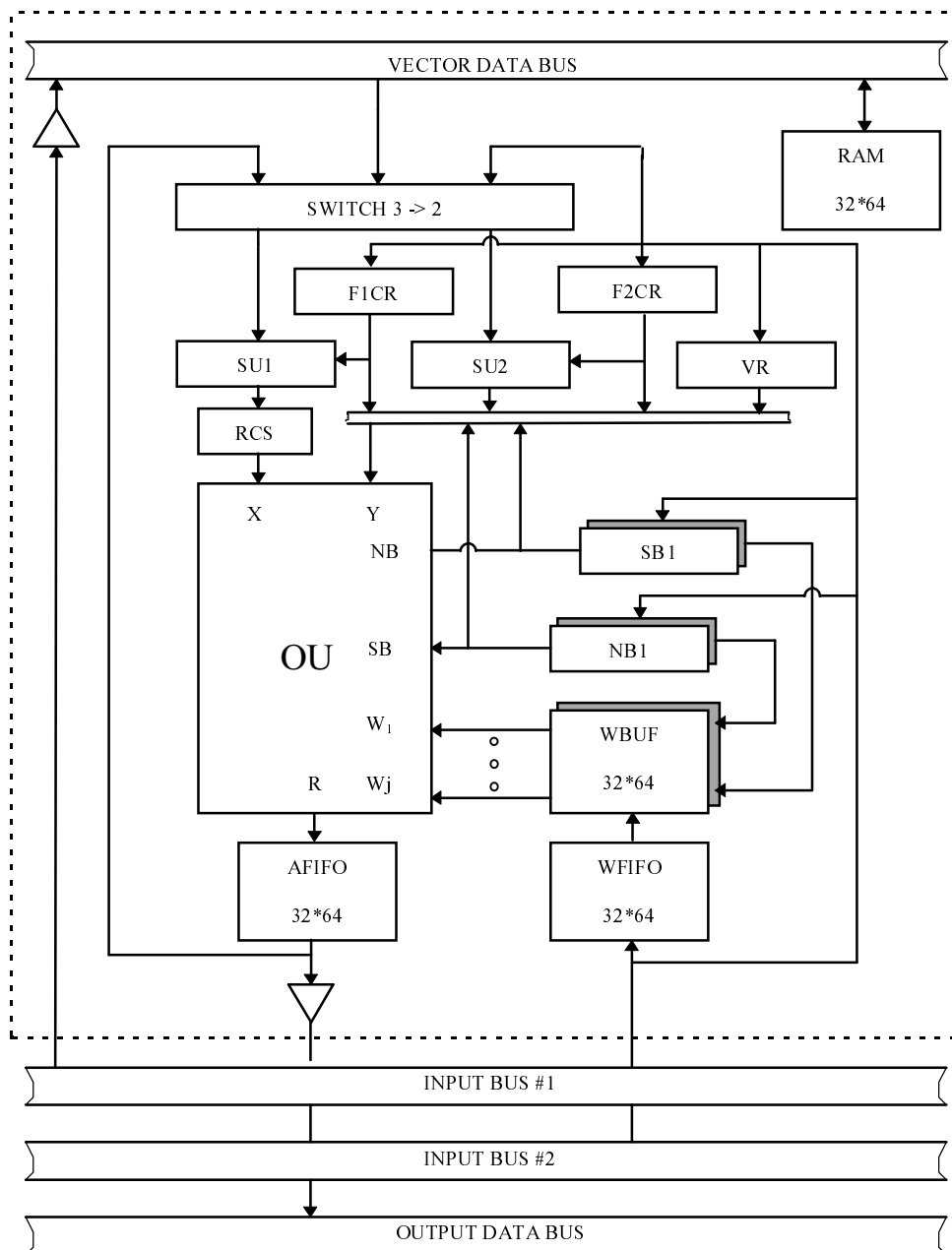


Рис. 2-3. Структурная схема устройства для векторных операций (VCP)

### Операционное устройство **OU**

**OU** служит для выполнения арифметических и логических операций над 64-разрядными словами упакованных данных  $\mathbf{X}=\{X_K \dots X_1\}$  и  $\mathbf{Y}=\{Y_1 \dots Y_1\}$ , поступающими соответственно на входы **X** и **Y** **OU**, и матрицей весов **WOPER**, которая подается на входы  $W_1, \dots, W_J$  в виде  $J$  64-разрядных слов упакованных весовых коэффициентов  $W_1=\{W_{11} \dots W_{11}\}, \dots, W_J=\{W_{11} \dots W_{11}\}$ . Результат каждой операции формируется на выходе **R** в виде 64-разрядного слова упакованных данных  $\mathbf{R}=\{R_1 \dots R_1\}$ . Переменная  $I$  может принимать любое целочисленное значение от 1 до 64 в зависимости от содержимого **NB2** ( $I$  равно количеству единиц в **NB2**), а переменная  $J$  - от 1 до 32 в зависимости от содержимого **SB2** ( $J$  равно количеству единиц в **SB2**). Значения, принимаемые переменной  $K$ ,

---

зависят от типа выполняемой операции:  $K=I$  - для арифметических операций,  $K=J$  - для операции взвешенного суммирования.

Тип выполняемой операции задается кодом команды. Операции выполняются в конвейере с темпом одна операция за такт. Никаких признаков результатов операций **OU** не формирует.

### Циклический сдвигатель вправо **RCS**

В зависимости от кода команды 64-разрядные слова, поступающие на вход **X OU**, проходят через **RCS** без изменений или циклически сдвигаются вправо на один разряд. За один такт выполняется сдвиг одного слова как единого операнда, не зависимо от количества данных в слове.

### Узлы, аппаратно реализующие функцию насыщения **SU1**, **SU2** и их регистры управления **F1CR**, **F2CR**

**SU1** и **SU2** служат для вычисления нелинейных функций активации над 64-разрядными словами упакованных данных. Для каждого нелинейного преобразователя в **VCP** предусмотрен свой программно доступный регистр управления (**F1CR** для **SU1** и **F2CR** для **SU2**), содержимое которого в зависимости от кода выполняемой команды может определять количество и разрядности данных, составляющих обрабатываемое слово, а также максимальное абсолютное значение результата вычисления функции насыщения для каждого составляющего слова.

### Коммутатор 3 в 2 **SWITCH 3→2**

Коммутатор 3 в 2 **SWITCH 3→2** обеспечивает выбор двух источников векторных данных, поступающих на входы исполнительных узлов **VCP**. В зависимости от кода команды через коммутатор 3 в 2 на вход каждого нелинейного преобразователя могут поступать следующие вектора слов упакованных данных:

- ♦ вектора данных с нулевыми значениями всех разрядов;
- ♦ содержимое **RAM** с шины векторных данных **VCP** - **VECTOR DATA BUS**;
- ♦ содержимое **AFIFO** по цепи обратной связи **VCP**;
- ♦ вектора данных из внешней памяти.

### Памяти весовых коэффициентов **WBUF** и **WOPER**

Память весовых коэффициентов состоит из двух матриц ячеек памяти **WBUF** и **WOPER**, каждая из которых имеет емкость 32x64 бита и позволяет хранить матрицу весов **W** в виде  $J$  64-разрядных слов упакованных весовых коэффициентов:  $W_1=\{W_{11} \dots W_{11}\}, \dots, W_J=\{W_{J1} \dots W_{J1}\}$ .

---

**WOPER** служит для хранения матрицы весов, используемой в операциях взвешенного суммирования, выполняемых **OU**. Выходы всех ячеек **WOPER** соединены непосредственно с соответствующими входами **OU**, а их входы - с выходами соответствующих ячеек **WBUF**. Благодаря этому запись во все ячейки **WOPER** осуществляется за один такт по команде **LOAD**. При этом содержимое **WBUF** целиком копируется в **WOPER**. Одновременно с этим содержимое регистров **SB1** и **NB1** копируется в регистры **SB2** и **NB2**.

**WBUF** служит для подкачки из **WFIFO** новой матрицы весов на фоне выполнения **OU** текущих операций взвешенного суммирования с использованием старой матрицы весов, хранящейся в **WOPER**. Загрузка матрицы весов в **WBUF** инициируется одной командой и осуществляется за 32 такта путем последовательной записи  $J$  64-разрядных слов упакованных весовых коэффициентов, выбираемых из **WFIFO**. При этом количество загружаемых слов  $J$  определяется содержимым **SB1** ( $J$  равно количеству единиц в **SB1**). В режиме записи **WBUF** работает по аналогии с памятью магазинного (стекового) типа. При этом первым загружается слово  $W_1 = \{W_{11} \dots W_{11}\}$ , а последним - слово  $W_J = \{W_{J1} \dots W_{J1}\}$ . Во всех этих словах количество весовых коэффициентов и границы между ними совпадают и определяются содержимым регистра **NB1**.

#### FIFO весовых коэффициентов **WFIFO**

Двухпортовое **WFIFO** имеет емкость 32x64 бит и используется в качестве накопительного буфера в процессе подкачки матрицы весов в **WBUF** из внешней памяти. Запись и чтение из **WFIFO** ведется 64-разрядными словами упакованных весовых коэффициентов. Загрузка **WFIFO** осуществляется через внутреннюю шину данных нейропроцессора - **INPUT BUS #2**.

#### Накопительное FIFO **AFIFO**

Двухпортовое **AFIFO** емкостью 32x64 бита используется в **VCP** в качестве аккумулятора и служит для хранения одного вектора 64-разрядных слов упакованных данных, которые являются результатом выполнения последней векторной операционной команды.

#### Векторный регистр **RAM**

**RAM** представляет собой однопортовую память типа FIFO емкостью 32x64 бита, которая подключена к шине векторных данных **VCP - VECTOR DATA BUS**. Основное отличие **RAM** от обычного FIFO заключается в том, что после чтения из **RAM** его содержимое не изменяется.

#### Регистр порогов **VR**

Регистр порогов **VR** служит для хранения 64-разрядного слова упакованных порогов или смещений. Существуют команды, по которым при выполнении операции взвешенного суммирования содержимое **VR** подается на вход **Y OU**.

---

## 2.4 Программируемые интерфейсы с внешними шинами

Нейропроцессор имеет два идентичных интерфейса с локальной и глобальной шинами. Каждый интерфейс имеет следующие особенности:

- суммарное число выводов 88, в том числе 64-разрядная шина данных и шина адреса до 19 разрядов с возможностью выдачи по ней 30-разрядного адреса в мультиплексном режиме;
- работа как с 64-разрядными словами данных, так и 32-разрядными полусловами данных (младший разряд 32-разрядного адреса используется для адресации 32-разрядного полуслова внутри 64-разрядного слова данных);
- возможность выполнения одноктактных циклов “чтение из памяти” и “запись в память”;
- наличие сигналов для работы нескольких процессоров с общей памятью;
- программируемое пользователем разбиение адресного пространства на один или два банка памяти, которые могут различаться типом памяти (SRAM, Flash ROM, DRAM, EDO DRAM), размером страницы памяти и временными параметрами;
- поддержка страничного режима для каждого банка памяти независимо от его типа в другом банке;
- возможность формировать циклы ожидания (задаются как программно, так и с помощью внешнего сигнала готовности);
- аппаратная поддержка режима регенерации для DRAM и EDO DRAM;
- набор управляющих сигналов интерфейса, позволяющего работать с внешней памятью различного типа напрямую без использования внешнего контроллера.

Нейропроцессор поддерживает как однопроцессорный, так и многопроцессорный режим работы по любой из двух внешних шин. Если к общей памяти подключены два процессора, то арбитраж для доступа к ней осуществляется между ними без использования внешнего контроллера.

Возможны три типа конфигурации внешней шины для многопроцессорного режима:

- 1) многопроцессорная конфигурация 1-го типа (банк 0 - “общий”, банк 1 - “общий”);
- 2) многопроцессорная конфигурация 2-го типа (банк 0 - “свой”, банк 1 - “общий”);
- 3) многопроцессорная конфигурация 3-го типа (банк 0 - “свой”, банк 1 - “чужой”).

Пример конфигурации 1-го типа можно видеть на рис. 2-4. Данная конфигурация характеризуется тем, что доступ в память (MEMORY

---

BANK1, MEMORY BANK2) может осуществляться только одним процессором - NP1 или NP2 - в данный момент времени.

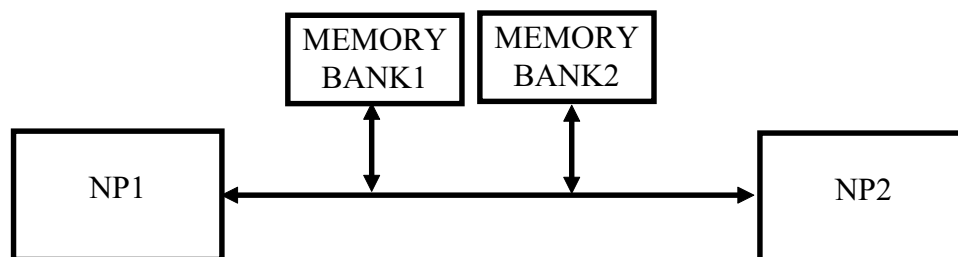


Рис. 2-4. Многопроцессорная конфигурация 1-го типа (банк 0 - "общий", банк 1 - "общий")

Пример конфигурации 2-го типа приведён на рис. 2-5. Она отличается от первой тем, что каждый процессор имеет свой банк памяти, которому другой не имеет доступа: NP1 принадлежит MEMORY BANK1, а NP2 - MEMORY BANK2. Банк MEMORY BANK3 является общим для NP1 и NP2, которые осуществляют к нему доступ поочередно через буфера BUFFER1 и BUFFER2.

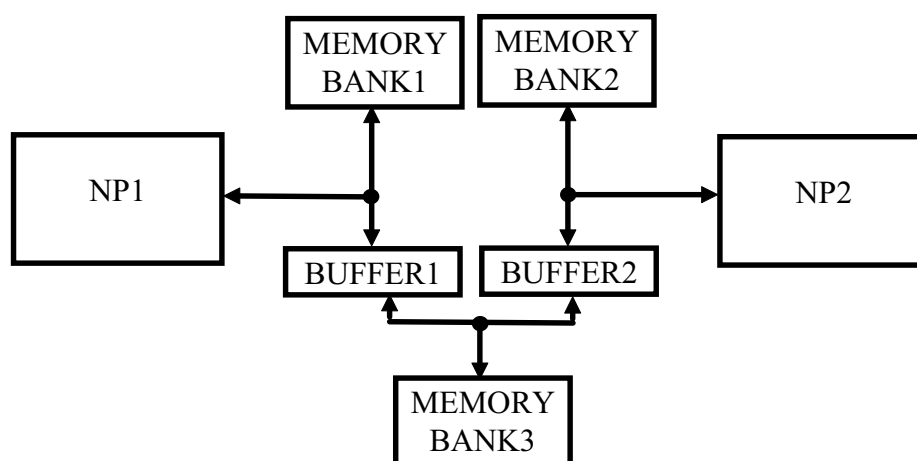


Рис. 2-5. Многопроцессорная конфигурация 2-го типа (банк 0 - "свой", банк 1 - "общий")

Пример конфигурации 3-го типа показан на рис. 2-6. Она характеризуется тем, что каждый процессор имеет свой банк памяти: NP1 принадлежит MEMORY BANK1, а NP2 - MEMORY BANK2, но существует возможность каждому процессору обратиться к чужому банку через буфер BUFFER.

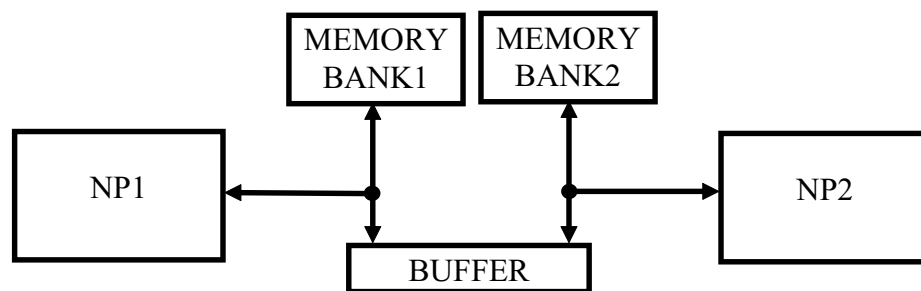


Рис. 2-6. Многопроцессорная конфигурация 3-го типа (банк 0 - "свой", банк 1 - "чужой")

## 2.5 Коммуникационные порты ввода/вывода

Нейропроцессор содержит два идентичных высокоскоростных коммуникационных порта: 0-й и 1-й, каждый из которых обеспечивает двунаправленный интерфейс для связи с внешним устройством. Рис. 2-7 показывает внутреннюю структуру одного коммуникационного порта.

Каждый порт  $CP_x$  ( $x = 0,1$ ) содержит:

**СРІ $x$**  - устройство управления интерфейсом порта, которое осуществляет арбитраж при передаче данных между нейропроцессором и внешним устройством через шину данных коммуникационного порта;

**МУХ** - мультиплексор для формирования адреса при запросе на ПДП от коммуникационного порта;

**ОСС $x$**  - счётчик канала вывода, который определяет число выводимых через коммуникационный порт 64-разрядных слов;

**ОСА $x$**  - регистр адреса канала вывода, который задаёт адрес памяти, откуда будут в режиме ПДП считываться данные при выводе через порт;

**ОСДР** - регистр данных канала вывода;

**ИСС $x$**  - счётчик канала ввода, который определяет число вводимых через коммуникационный порт 64-разрядных слов;

**ИСА $x$**  - регистр адреса канала ввода, который задаёт адрес памяти, куда будут в режиме ПДП записываться данные при вводе через порт;

**ИСДР $x$**  - регистр данных канала ввода.

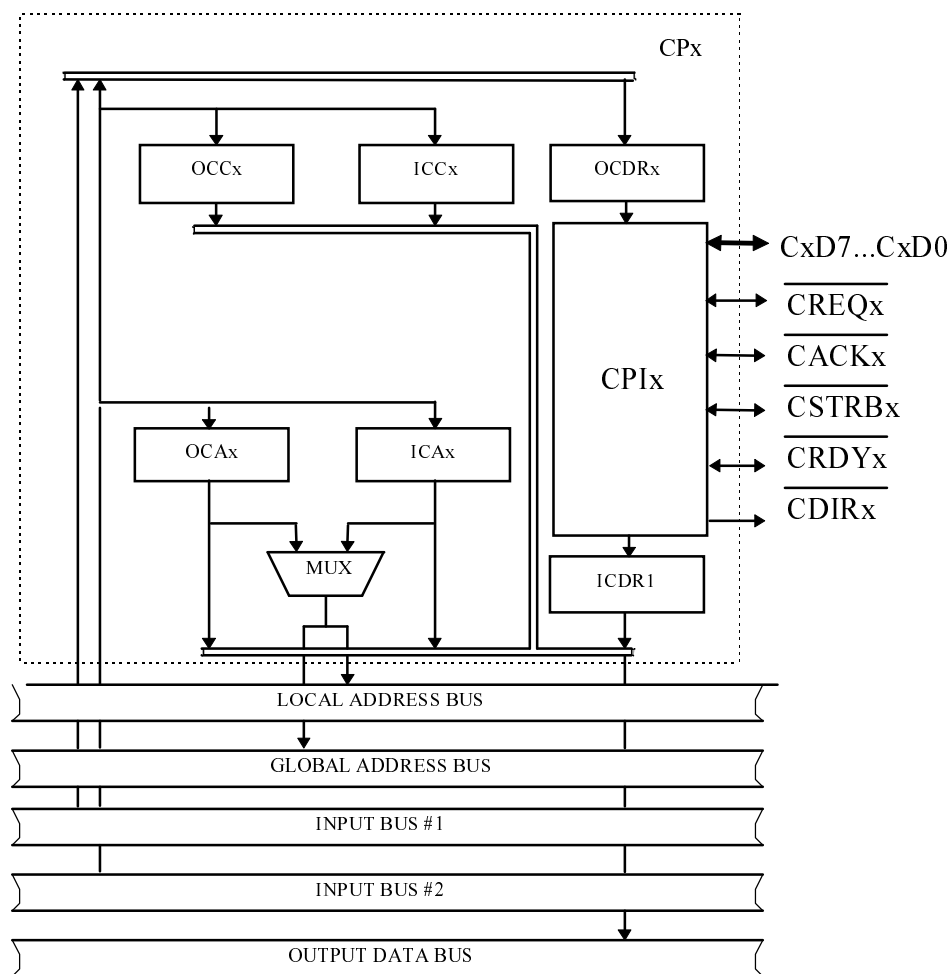


Рис. 2-7. Структурная схема коммуникационного порта  $CP_x$  ( $x = 0, 1$ )

### Двунаправленные линии управления и данных

Каждый коммуникационный порт содержит следующие двунаправленные линии управления и данных:

$\overline{CREQ}_x$  - запрос на выдачу по шине данных коммуникационного порта;

$\overline{CACK}_x$  - подтверждение на захват шины данных коммуникационного порта при получении сигнала  $\overline{CREQ}_x$  от другого нейропроцессора;

$\overline{CSTRB}_x$  - строб порта коммуникации. Передающий нейропроцессор устанавливает этот сигнал для индикации, что он выдал очередные данные на шину данных коммуникационного порта;

$\overline{CRDY}_x$  - сигнал готовности коммуникационного порта. Принимающий нейропроцессор устанавливает этот сигнал для индикации, что он принял данные через шину данных коммуникационного порта;

$C_xD(7-0)$  - шина данных коммуникационного порта. По этой шине пересылаются данные в обе стороны между двумя нейропроцессорами или между нейропроцессором и другим внешним устройством.



---

Кроме того, имеется выход  $\overline{CDIRx}$ , который указывает, в каком режиме работает в данный момент коммуникационный порт - ввода (на выходе высокий уровень) или вывода (на выходе низкий уровень).

Нейропроцессор использует 32-разрядный вычисляемый адрес при обращении во внешнюю память, причём обмен происходит по 32 или 64 разряда. Таким образом, доступное адресное пространство равно 16 Гбайт. Оно делится на две равные части - локальное и глобальное (см. рис. 3-1). Если старший разряд адреса равен нулю, идёт обращение к локальной памяти, если единица - к глобальной. Младший разряд вычисляемого адреса используется при доступе к 32-разрядным данным: если он равен нулю, используется младшая часть памяти (разряды 31 - 0), если единица - старшая (разряды 63 - 32). При обращении за 64-разрядными данными или при выборке команд он игнорируется.

Обмен 32-разрядными данными с внешней памятью производится только скалярными командами, если в качестве источника или приёмника в них указан 32-разрядный регистр. В случае, когда это 64-разрядный регистр или когда обмен задаётся векторной командой, используются соответственно 64-разрядные данные.

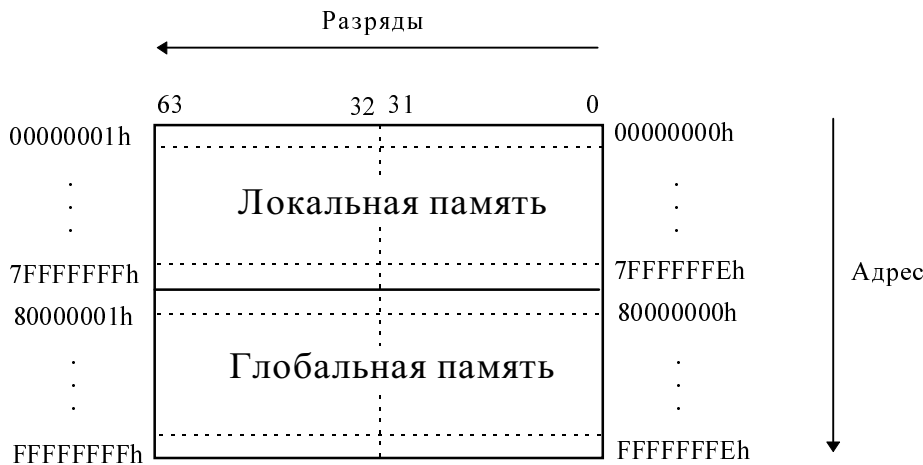


Рис. 3-1. Адресное пространство нейропроцессора

Нейропроцессор поддерживает одно внешнее прерывание и девять внутренних:

- два прерывания от таймеров;
- прерывание по переполнению при выполнении арифметических операций в RISC-ядре;
- прерывание по запрещенной векторной команде;
- четыре прерывания от каналов ввода-вывода по завершению обмена через коммуникационные порты;
- пошаговое прерывание в режиме отладки.

Данные прерывания представлены в таблице 4-1 и расположены в порядке уменьшения приоритета сверху вниз (при установке нескольких запросов на прерывание будет обслуживаться тот запрос, который имеет наибольший приоритет).

Таблица 4-1 Прерывания нейропроцессора

№	Тип прерывания	Адрес-вектор прерывания
1.	Обнуление системного таймера T0	00000000h
2.	Прерывание по переполнению при выполнении арифметических операций в RISC-ядре	00000008h
3.	Прерывание по запрещённой векторной команде	00000010h
4.	Внешнее прерывание	00000018h
5.	Прерывание по завершению ввода по коммуникационному порту 1	00000020h
6.	Прерывание по завершению ввода по коммуникационному порту 0	00000028h
7.	Прерывание по завершению вывода по коммуникационному порту 1	00000030h
8.	Прерывание по завершению вывода по коммуникационному порту 0	00000038h
9.	Обнуление таймера T1	00000040h
10.	Пошаговое прерывание	00000048h

### 5.1 Форматы команд

Нейропроцессор работает с командами фиксированной длины по 32 или 64 разряда (см. рис. 5-1). Можно выделить 4 основных группы команд: векторные команды и команды скалярные - управления, модификации адресных регистров и пересылки с возможностью в каждой из них задать операцию обработки операндов. Все команды выполняются за один такт синхронизации.

#### Команды пересылки типа регистр-память

Рист/пр-к  $\leftrightarrow$  ( $f_{\text{адр}}(AR_i, GR_i)$ );  $AR_i \leftarrow f_M(AR_i, GR_i)$  (R/W = 0 - чтение; R/W = 1 - запись)

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
P	0	1	AM			R/W	ARi			Rист/пр-к							КОП				СК										

32

Рист/пр-к  $\leftrightarrow$  ( $f_{\text{адр}}(AR_i, \text{Смещение})$ );  $AR_i \leftarrow f_M(AR_i, \text{Смещение})$  (R/W = 0 - чтение; R/W = 1 - запись)

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
P	1	1	0	AMC			R/W	ARi			Rист/пр-к							КОП				СК									
АДРЕС (СМЕЩЕНИЕ)																															

63

32

#### Команды пересылки типа регистр-регистр

Rпр-к  $\leftarrow$  Rист

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
P	1	1	1	Rпр-к					Rист							КОП				СК											

Rпр-к  $\leftarrow$  константа

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
P	1	0	0	Rпр-к					0 0 x x x x				КОП				СК														

63

32

Сброс/установка битов регистра PSWR (R/S = 0 - сброс; R/S = 1 - установка)

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
P	1	0	0	R/S	1	1	1	0	1	0 0 x x x x				КОП				СК													

63

32

Рис. 5-1. Форматы команд нейропроцессора

## Команды модификации адресных регистров

$AR_j \leftarrow f_m(AR_i, GR_i)$

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	
P	1	0	1	RM	1		AR <sub>i</sub>	0	1	x	x	AR <sub>j</sub>																				

$AR_j \leftarrow f_m(AR_i, \text{Смещение})$

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	
P	1	0	0	RMC	1		AR <sub>i</sub>	0	1	x	x	AR <sub>j</sub>																				
63																																32
КОНСТАНТА-СМЕЩЕНИЕ																																

Нет операций ввода/вывода и модификации адресных регистров

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	
P	1	0	1	x	x	0	x	x	x	0	1	x	x	x	x																	

Нет операций ввода/вывода и модификации адресных регистров

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	
P	1	0	0	x	x	0	x	x	x	0	1	x	x	x	x																	
63																																32
КОНСТАНТА-СМЕЩЕНИЕ																																

## Команды управления

Переход/переход к подпрограмме ( $J/C = 0$  - переход;  $J/C = 1$  - переход к подпрограмме)

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
P	0	0	0	PM	J/C		AR <sub>i</sub>	1	0	Условие																					

Переход/переход к подпрограмме со смещением ( $J/C = 0$  - переход;  $J/C = 1$  - переход к подпрограмме)

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	
P	1	0	0	PMC	J/C		AR <sub>i</sub>	1	0	Условие																						
63																																32
АДРЕС (смещение)																																

Возврат из подпрограммы/прерывания ( $S/I = 0$  - возврат из подпрограммы;  $S/I = 1$  - возврат из прерывания)

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
P	0	0	0	S/I	0	1	1	1	1	1	1	1	Условие																		

Рис. 5-1. Форматы команд нейтропроцессора (продолжение)

## Векторные команды

AFIFO $\leftrightarrow$ (fадр.(ARi, GRi)); ARi $\leftarrow$ f<sub>m</sub>(ARi, GRi) (R/W = 0 - чтение; R/W = 1 - запись)

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
P	0	0	AM	R/W	ARi	0	1	R_W	W	количество	К О П В К																				

WFIFO $\leftarrow$ (fадр.(ARi, GRi)); ARi $\leftarrow$ f<sub>m</sub>(ARi, GRi) (R/W = 0 - чтение; R/W = 1 - запись)

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
P	0	0	AM	x	ARi	0	0	1	W	количество	К О П В К																				

Нет операций ввода/вывода

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
P	0	0	x	x	x	x	x	x	x	0	0	0	W	x	x	x	x	x	К О П В К												

## Обозначения:

AM - метод адресации fадр.(ARi, GRi) с модификацией адресного регистра ARi $\leftarrow$ f<sub>m</sub>(ARi, GRi) (см. табл. 5-1)

AMC- метод адресации с помощью константы fадр.(ARi, Смещение) с модификацией адресного регистра ARi $\leftarrow$ f<sub>m</sub>(ARi, Смещение)(см. табл. 5-2)

RM - метод модификации адресного регистра ARj $\leftarrow$ f<sub>m</sub>(ARi, GRi) (см. табл. 5-3)

RMC - метод модификации адресного регистра с помощью константы ARj $\leftarrow$ f<sub>m</sub>(ARi, Смещение) (см. табл. 5-4)

PM - метод модификации PC при переходах см. табл. 5-5)

PMC - метод модификации PC при переходах с помощью константы (см. табл. 5-6)

ARi - адресный регистр i (i = 0, ..., 7)

ARj - адресный регистр j (j = 0, ..., 3 если 24-й разряд команды равен 0,  
j = 4, ..., 7 если 24-й разряд команды равен 1)

Условие - анализируемое сочетание признаков для команд управления (см. табл. 5-7)

Рист/пр-к - регистр-источник/приёмник в командах пересылки данных (см. табл. 5-8)

Количество - число повторений данной команды, может быть от 1 до 32 (только для векторных команд)

R\_W - управление записью в RAM (0 - нет записи/1-есть запись)

W - управление загрузкой весов из WFIFO в OU (0 - нет загрузки/1-есть загрузка)

КОП СК - код арифметической операции для скалярной команды (см. раздел 5.7)

КОП ВК - код арифметической операции для векторной команды (см. раздел 5.8)

P - бит параллельной работы (0 - запрещён запуск данной команды на исполнение, пока не закончатся все ранее выбранные команды/ 1 - разрешён запуск)

*Рис. 5-1. Форматы команд нейропроцессора (продолжение)*

## 5.2 Методы адресации данных

Метод адресации задаётся для 32-разрядных скалярных команд пересылки типа регистр-память и векторных команд полем AM, описание которого см. в табл. 5-1. Для 64-разрядных скалярных команд пересылки

типа регистр-память метод адресации определяется полем АМС (см табл. 5-2).

*Таблица 5-1. Методы адресации для 32-разрядных скалярных команд пересылки типа регистр-память и векторных команд*

АМ	$f_{\text{адр.}}(AR_i, GR_i)$	$f_{\text{м}}(AR_i, GR_i)$
0 0 0	$GR_i$	$AR_i$
0 0 1	$AR_i+GR_i$	$AR_i+GR_i$
0 1 0	$GR_i$	$GR_i$
0 1 1	Резерв	Резерв
1 0 0	$AR_i$	$AR_i$
1 0 1	$AR_i$	$AR_i+GR_i$
1 1 0	$AR_i-a$	$AR_i-a$
1 1 1	$AR_i$	$AR_i+a$

*Таблица 5-2. Методы адресации для 64-разрядных скалярных команд пересылки типа регистр-память*

АМС	$f_{\text{адр.}}(AR_i, \text{Смещение})$	$f_{\text{м}}(AR_i, \text{Смещение})$
0 0	Адрес	$AR_i$
0 1	$AR_i+ \text{Смещение}$	$AR_i+ \text{Смещение}$
1 0	Адрес (Смещение)	Адрес (Смещение)
1 1	Резерв	Резерв

### 5.3 Методы модификации адресных регистров

Метод модификации адресных регистров задаётся для соответствующих 32-разрядных скалярных команд полем RM, описание которого см. в табл. 5-3. Для 64-разрядных скалярных команд метод модификации определяется полем RMC (см табл. 5-4).

Таблица 5-3. Методы модификации адресного регистра для соответствующих 32-разрядных скалярных команд

RM	$AR_j \leftarrow f_m(AR_i, GR_i)$
0 0	$AR_i$
0 1	$AR_i + GR_i$
1 0	$GR_i$
1 1	Резерв

Таблица 5-4. Методы модификации адресного регистра для соответствующих 64-разрядных скалярных команд

RMC	$f_m(AR_i, \text{Смещение})$
0 0	$AR_i$
0 1	$AR_i + \text{Смещение}$
1 0	Константа- смещение e
1 1	Резерв



#### 5.4 Методы модификации PC при переходах

Метод модификации PC задаётся для 32-разрядных скалярных команд перехода полем PM, описание которого см. в табл. 5-5. Для 64-разрядных скалярных команд перехода метод модификации PC определяется полем PMC (см табл. 5-6).

Таблица 5-5. Методы модификации PC для 32-разрядных скалярных команд перехода

PM	Модификация PC
0 0	PC:=ARi
0 1	PC:=ARi+GRi
1 0	PC:=GRi
1 1	PC:=PC+GRi

Таблица 5-6. Методы модификации PC для 64-разрядных скалярных команд перехода

PMC	Модификация PC
0 0	PC:=ARi
0 1	PC:=ARi+ Смещение
1 0	PC:=Адрес (Смещение)
1 1	PC:=PC+ Смещение

## 5.5 Анализируемое сочетание признаков для команд управления

Скалярные команды управления в зависимости от поля “Условие” могут быть безусловными или выполняться в зависимости от сочетания признаков слова состояния процессора: N - признак знака, Z - признак нуля, V - признак переполнения, C - признак переноса (см. табл. 5-7).

Таблица 5-7. Анализируемое сочетание признаков для скалярных команд управления

Условие	Анализируемое сочетание признаков
0 0 0 0	Переход, если C=0
0 0 0 1	Переход, если V=0
0 0 1 0	Переход, если N+Z=0
0 0 1 1	Переход, если N=0
0 1 0 0	Переход, если $(V \oplus N)+Z=0$
0 1 0 1	Переход, если $V \oplus N=0$
0 1 1 0	Переход, если Z=0
0 1 1 1	Переход безусловный
1 0 0 0	Переход, если C=1
1 0 0 1	Переход, если V=1
1 0 1 0	Переход, если N+Z=1
1 0 1 1	Переход, если N=1
1 1 0 0	Переход, если $(V \oplus N)+Z=1$
1 1 0 1	Переход, если $V \oplus N=1$
1 1 1 0	Переход, если Z=1
1 1 1 1	Перехода нет

## 5.6 Программно доступные регистры

Для скалярных команд пересылок поле Рист/пр-к определяет регистры, участвующие в операциях пересылки данных (см. табл. 5-8), причём в зависимости от кода в этом поле обращение может идти как к 32-разрядным и 64-разрядным регистрам, так и к 64-разрядным регистровым парам.

Таблица 5-8. Поле  $R_{ист/пр-к}$  для скалярных команд пересылок

$R_{ист/пр-к}$	Регистр-источник	Регистр-приемник	Разрядность
0 0 0 0 0 0	AR0	AR0	32
0 0 0 0 0 1	AR1	AR1	32
0 0 0 0 1 0	AR2	AR2	32
0 0 0 0 1 1	AR3	AR3	32
0 0 0 1 0 0	AR4	AR4	32
0 0 0 1 0 1	AR5	AR5	32
0 0 0 1 1 0	AR6	AR6	32
0 0 0 1 1 1	AR7(SP)	AR7(SP)	32
0 0 1 0 0 0	OCA0	OCA0	32
0 0 1 0 0 1	ICA0	ICA0	32
0 0 1 0 1 0	OCA1	OCA1	32
0 0 1 0 1 1	ICA1	ICA1	32
0 0 1 1 0 0	T0	T0	32
0 0 1 1 0 1	LMICR	LMICR	32
0 0 1 1 1 0	GMICR	GMICR	32
0 0 1 1 1 1	PC	PC	32
0 1 0 0 0 0	GR0	GR0	32
0 1 0 0 0 1	GR1	GR1	32
0 1 0 0 1 0	GR2	GR2	32
0 1 0 0 1 1	GR3	GR3	32
0 1 0 1 0 0	GR4	GR4	32
0 1 0 1 0 1	GR5	GR5	32
0 1 0 1 1 0	GR6	GR6	32
0 1 0 1 1 1	GR7	GR7	32

Таблица 5-8. Поле  $R_{ист/пр-к}$  для скалярных команд пересылок (продолжение)

$R_{ист/пр-к}$	Регистр-источник	Регистр-приемник	Разрядность
0 1 1 0 0 0	OCC0	OCC0	32
0 1 1 0 0 1	ICC0	ICC0	32
0 1 1 0 1 0	OCC1	OCC1	32
0 1 1 0 1 1	ICC1	ICC1	32
0 1 1 1 0 0	T1	T1	32
0 1 1 1 0 1	Резерв	PSWRreset	32
0 1 1 1 1 0	INTR	Резерв	32
0 1 1 1 1 1	PSWR	PSWR	32
1 0 0 0 0 0	GR0, AR0	GR0, AR0	64(32+32)
1 0 0 0 0 1	GR1,AR1	GR1,AR1	64(32+32)
1 0 0 0 1 0	GR2,AR2	GR2,AR2	64(32+32)
1 0 0 0 1 1	GR3,AR3	GR3,AR3	64(32+32)
1 0 0 1 0 0	GR4,AR4	GR4,AR4	64(32+32)
1 0 0 1 0 1	GR5,AR5	GR5,AR5	64(32+32)
1 0 0 1 1 0	GR6,AR6	GR6,AR6	64(32+32)
1 0 0 1 1 1	GR7,AR7	GR7,AR7	64(32+32)
1 0 1 0 0 0	OCC0, OCA0	OCC0, OCA0	64(32+32)
1 0 1 0 0 1	ICC0, ICA0	ICC0, ICA0	64(32+32)
1 0 1 0 1 0	OCC1, OCA1	OCC1, OCA1	64(32+32)
1 0 1 0 1 1	ICC1, ICA1	ICC1, ICA1	64(32+32)
1 0 1 1 0 0	T1, T0	T1, T0	64(32+32)
1 0 1 1 0 1	DIR0	DOR0	64
1 0 1 1 1 0	DIR1	DOR1	64
1 0 1 1 1 1	PSWR,PC	PSWR,PC	64(32+32)
1 1 0 0 0 0	Резерв	NBL	32
1 1 0 0 0 1	Резерв	SBL	32
1 1 0 0 1 0	Резерв	F1CRL	32
1 1 0 0 1 1	Резерв	F2CRL	32
1 1 0 1 0 0	Резерв	NBH	32
1 1 0 1 0 1	Резерв	SBH	32
1 1 0 1 1 0	Резерв	F1CRH	32
1 1 0 1 1 1	Резерв	F2CRH	32
1 1 1 0 0 0	Резерв	NB	64
1 1 1 0 0 1	Резерв	SB	64
1 1 1 0 1 0	Резерв	F1CR	64
1 1 1 0 1 1	Резерв	F2CR	64
1 1 1 1 0 0	Резерв	VR	64
1 1 1 1 0 1	Резерв	PSWRset	32
1 1 1 1 1 0	Резерв	VRL	32
1 1 1 1 1 1	Резерв	VRH	32

## 5.7 Арифметико-логические операции, задаваемые в скалярной команде

В скалярной команде можно задать арифметическую, логическую операцию или операцию сдвига над регистрами GR0 - GR7 с помощью поля КОП СК (см. рис. 5-1). Форматы данного поля приведены на рис. 5-2.

Форматы поля КОП СК, задающего операцию сдвига  
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

0	0	ТС	Величина сдвига	GR <sub>ист</sub>	GR <sub>пр-к</sub>
---	---	----	-----------------	-------------------	--------------------

ТС	Тип сдвига
0 0	Циклический сдвиг
0 1	Логический сдвиг
1 0	Арифметический сдвиг
1 1	Логический сдвиг через "С"

Формат поля КОП СК, за дающего логическую операцию

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
W	0	КЛОП	GR <sub>ист2</sub>	GR <sub>ист1</sub>	GR <sub>пр-к</sub>										

КЛОП	Код одной из 16 возможных логических операций: GR <sub>пр-к</sub> <- флог(GR <sub>ист2</sub> , GR <sub>ист1</sub> )
------	---

Формат поля КОП СК, за дающего арифметическую операцию

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
W	1	КАОП	GR <sub>ист2</sub>	GR <sub>ист1</sub>	GR <sub>пр-к</sub>										

КАОП	Код арифметической операции
0 0 0 0	GR <sub>пр-к</sub> <- GR <sub>ист2</sub> -GR <sub>ист1</sub>
0 0 0 1	GR <sub>пр-к</sub> <- GR <sub>ист2</sub> -GR <sub>ист1</sub> -1+"С"
0 0 1 0	GR <sub>пр-к</sub> <- GR <sub>ист2</sub> +1
0 0 1 1	GR <sub>пр-к</sub> <- GR <sub>ист2</sub> +"С"
0 1 0 0	GR <sub>пр-к</sub> <- GR <sub>ист2</sub> -1
0 1 0 1	GR <sub>пр-к</sub> <- GR <sub>ист2</sub> -1+"С"
0 1 1 0	GR <sub>пр-к</sub> <- GR <sub>ист2</sub> +GR <sub>ист1</sub>
0 1 1 1	GR <sub>пр-к</sub> <- GR <sub>ист2</sub> +GR <sub>ист1</sub> +"С"
1 0 0 0	Первый шаг умножения
1 0 0 1	Шаг умножения
1 0 1 X	Резерв
1 1 0 0	GR <sub>пр-к</sub> <- -GR <sub>ист2</sub>
1 1 X 1	Резерв
1 1 1 X	Резерв

W	Управление записью в GR <sub>пр-к</sub> и в регистр PSWR признаков
0 1	Есть запись в GR <sub>пр-к</sub> , нет записи признаков
1 0	Есть запись признаков, нет записи в GR <sub>пр-к</sub>
1 1	Есть запись в GR <sub>пр-к</sub> и есть запись признаков

Рис. 5-2. Форматы поля КОП СК, задающего арифметико-логическую операцию в скалярной команде

## 5.8 Арифметико-логические операции, задаваемые в векторной команде

В векторной команде можно задать арифметическую или логическую операцию над данными, упакованными в 64-разрядные слова, с помощью поля КОП ВК (см. рис. 5-1). Форматы данного поля приведены на рис. 5-3.

Формат поля КОП ВК, задающего логическую операцию

12	11	10	9	8	7	6	5	4	3	2	1	0
1	0	КЛОП			FPX	FPY	X	Y	L			

КЛОП	Код одной из 16 возможных логических операций: AFIFO <- fлог(X,Y)
------	---

Формат поля команд КОП ВК, задающего арифметическую операцию

12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	КАОП			FAX	FAY	X	Y	L			

КАОП				Код арифметической операции
X	0	0	X	AFIFO <- X- Y
X	0	1	X	AFIFO <- X+1
X	1	0	X	AFIFO <- X-1
X	1	1	X	AFIFO <- X+Y

Формат поля команд КОП ВК, задающего операцию маскирования или взвешенного суммирования

12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	M	0	SH	FPX	FPY	X	Y	L			

Операция маскирования  $X * M + Y * \bar{M}$

0	0	M	VR	SH	FAX	FAY	X	Y	L			
---	---	---	----	----	-----	-----	---	---	---	--	--	--

Операция взвешенного суммирования типа  $W * X + Y$

SH	Управление циклическим сдвигом операнда X на 1 разряд вправо
0	Нет сдвига
1	Есть сдвиг

VR	Управление выборкой регистра VR в качестве операнда Y
0	VR не является операндом Y
1	VR является операндом Y

M	Выбор операнда - маски	
0	0	Маскирование отсутствует
0	1	Маска выбирается из RAM
1	0	Маска выбирается из AFIF0
1	1	Маска выбирается из внешней памяти

Рис. 5-3. Форматы поля КОП ВК, задающего арифметико-логическую операцию в векторной команде

---

Формат поля команд КОП ВК, задающего запись в AFIFO  
содержимого регистров F2CR, F1CR, NB2, SB2, VR

12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	00	1	X	X	X	00	00	00	L		

Формат поля команд КОП ВК, не задающего никакой  
арифметико-логической операции

12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	00	0	X	X	X	00	00	00	L		

Обозначение:

X и Y - операнды:

00 - операнд равен нулю;

01 - операнд выбирается из RAM;

10 - операнд выбирается из AFIFO;

11 - операнд выбирается из внешней памяти.

FPX и FPY - управление пороговой функцией для операндов X и Y:

0 - функция не используется;

1 - функция используется.

FAX и FAY - управление функцией насыщения для операндов X и Y:

0 - функция не используется;

1 - функция используется.

L(LOAD) - управление перезаписью весов из теневой матрицы (WBUF) в рабочую (WOPER):

0 - нет перезаписи;

1 - есть перезапись.

*Рис. 5-3. Форматы поля КОП ВК, задающего арифметико-логическую операцию в векторной команде (продолжение)*

## Инициализация нейропроцессора после системного сброса

Инициализация нейропроцессора после системного сброса может проводиться двумя способами в зависимости от уровня на входе *BOOTM* :

- ♦ если на входе *BOOTM* низкий уровень, то инициализация проводится по глобальной шине;
- ♦ если на входе *BOOTM* высокий уровень, то инициализация проводится по коммуникационному порту 1.

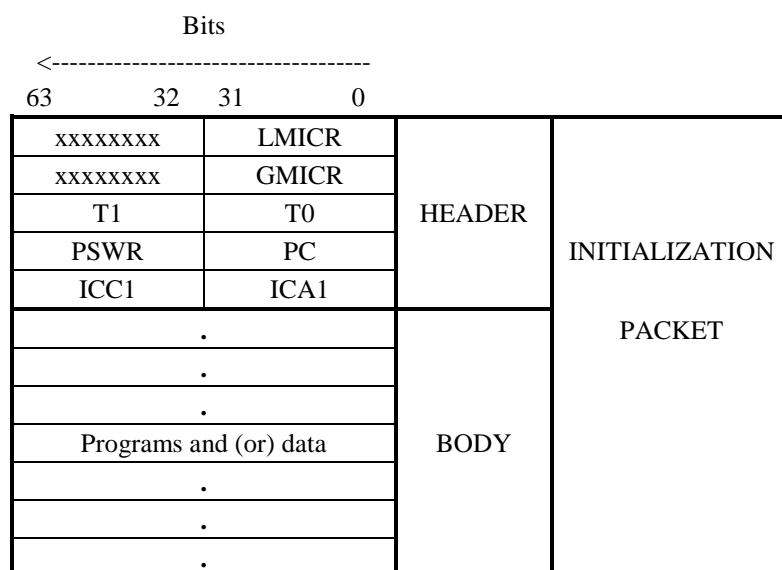
### 6.1 Инициализация нейропроцессора по глобальной шине

Инициализация по глобальной шине проводится после того, как сигнал  $\overline{RESET}$  становится пассивным, на входе *BOOTM* низкий уровень, и пришёл отрицательный импульс на вход внешнего прерывания  $\overline{INT}$ . Данное прерывание не фиксируется, но по нему формируется запрос на выборку первой команды по глобальной шине по адресу C0000000h и начинается обычная работа нейропроцессора.

### 6.2 Инициализация нейропроцессора по коммуникационному порту 1

Если на входе *BOOTM* высокий уровень, то инициализация проводится по коммуникационному порту 1, который после действия сигнала  $\overline{RESET}$  начинает работать на приём. Он получает пакет 64-разрядных двойных слов, первые пять из этих слов являются заголовком пакета, а остальные - телом пакета. Содержимое пакета можно видеть на рис. 6-1.





*Рис. 6-1. Содержимое пакета инициализации нейропроцессора по коммуникационному порту 1*

### Заголовок пакета

1-е двойное слово: старшие 32 разряда не имеют значение, младшие 32 разряда определяют информацию, которая будет записана в регистр управления интерфейсом локальной шины LMICR.

2-е двойное слово: старшие 32 разряда не имеют значение, младшие 32 разряда определяют информацию, которая будет записана в регистр управления интерфейсом глобальной шины GMICR.

3-е двойное слово: старшие 32 разряда будут записаны в таймер T1, а младшие 32 - в таймер T0.

4-е двойное слово: старшие 32 разряда будут записаны в регистр слова состояния процессора (PSWR), а младшие 32 - в счётчик команд PC.

5-е двойное слово: старшие 32 разряда будут записаны в счётчик канала ввода ICC1, а младшие 32 - в регистр адреса канала ввода ICA1.

### Тело пакета

Тело пакета может содержать системные программы, программы пользователя и обработки прерываний, массивы данных и т.д.

После того, как заголовок пакета принят, нейропроцессор имеет полную информацию о своём внешнем окружении и о том, сколько информации необходимо ещё принять и куда её поместить. Далее принимаемые двойные слова (тело пакета) в режиме ПДП записываются в память. Как только пройдёт последнее ПДП, формируется запрос на прерывание по завершению ввода по порту 1. Оно не фиксируется, но по нему происходит обращение за первой командой по адресу, записанному в PC, и начинается обычная работа нейропроцессора.





**АКЦИОНЕРНОЕ ОБЩЕСТВО  
НАУЧНО-ТЕХНИЧЕСКИЙ ЦЕНТР**

**Научно-технический центр Модуль  
АЯ 166, Москва, 125190, Россия  
Тел: +7 (095) 152-9335  
Факс: +7 (095) 152-4661  
E-Mail: [postmast@module.vympel.msk.ru](mailto:postmast@module.vympel.msk.ru)  
WWW: <http://www.module.vympel.msk.ru>**

Напечатано в России

Дата издания: 19 июня 1998

©НТЦ Модуль, 1998

Все права сохранены.

Никакая часть информации, приведенная в данном документе, не может быть адаптирована или воспроизведена, кроме как согласно письменному разрешению владельцев авторских прав.

НТЦ Модуль оставляет за собой право производить изменения как в описании, так и в самом продукте без дополнительных уведомлений. НТЦ Модуль не несет ответственности за любой ущерб, причиненный использованием информации в данном описании, ошибками или недосказанностью в описании, а также путем неправильного использования продукта.