УДК 004.383

Гетерогенная многопроцессорная система на кристалле с производительностью 512 Gflops

Эйсымонт Алексей Леонидович, Черников Антон Владимирович,
Косоруков Дмитрий Евгеньевич, Насонов Илья Игоревич,
Комлев Арсений Александрович
eisymont@module.ru

ЗАО НТЦ «Модуль»
125190, г. Москва, а/я 166
Тел. +7 (495) 531-30-80
http://www.module.ru

Аннотация: В статье рассмотрены вопросы реализации энергоэффективной гетерогенной и толерантной к задержкам выполнения операций с памятью системы на кристалле (СнК) с тактовой частотой 1 ГГц, пиковой производительностью 512 Gflops и иерархически организованной внутренней памятью. СнК содержит 16 векторных ядер NMC4 из семейства NeuroMatrix и пять скалярных ядер Cortex-A5 фирмы ARM.

Ключевые слова: архитектура NeuroMatrix, многоядерная гетерогенная СнК, векторная архитектура, VLIW, проблема стены памяти, встроенный вычислитель, обработка сигналов, нейронные сети, физическое проектирование.

Heterogeneous multicore system on chip with 512 Gflops peak performance

Eysymont Alexey, Chernikov Anton,
Kosorukov Dmitry, Nasonov Ilya, Komlev Arseny
eisymont@module.ru

JSC RC «Module»
P.O. Box 166, Moscow, Russia, 125190
http://www.module.ru

Abstract: This article is devoted to questions and methods of implementing energy efficient heterogeneous and tolerant to memory latency system on chip (SoC) operating at 1GHz frequency, with 512 Gflops peak performance and hierarchically organized internal memory. SoC contains sixteen NeuroMatrix NMC4 processor cores and five ARM Cortex-A5.

Keywords: NeuroMatrix architecture, multicore heterogeneous SoC, vector architecture, VLIW, memory wall problem, embedded computer, digital signal processing, neural networks, physical design.

Введение

СнК NM6408 разрабатывалась для встроенных вычислителей информационно-управляющих систем (ИУС). Современные приложения для встроенных вычислителей, такие как первичная обработка сигналов (преобразования Фурье, Адамара, фильтры и т.д.) и многослойные нейронные сети разного типа в настоящее время требуют достижения реальной производительности около десятка Tflops над вещественными числами одинарной точности. До недавнего времени встроенные вычислители достигали высокой производительности и энергоэффективности за счет использования в них специализированных процессоров. Например, использовались процессоры обработки сигналов (DSP) или нейросетевые процессоры разного типа. В настоящее время, для экономии затрат на разработку и эксплуатацию, а также, из-за быстро меняющихся прикладных задач и алгоритмов, стараются вместо жестко специализированных процессоров создавать конкурирующие с ними по производительности и энергоэффективности проблемно-ориентированные процессоры с более широкими возможностями для решения различных задач и использования широкого класса алгоритмов из выбранных предметных областей [1,2].

Современная мировая практика показывает, что получается создавать удачные проблемно-ориентированные процессоры, если учитывать особенности работы с памятью и подходящую организацию вычислений в выбранной предметной области, а также имеет смысл использовать архитектурные и микроархитектурные приёмы, хорошо отработанные при создании элементной базы суперкомпьютеров [2,4]. Например, требования обеспечения высокой производительности и одновременно высокой энергоэффективности во встроенных вычислителях в настоящее время приводят к тому, что в них напрямую используются графические процессоры GPU фирм NVIDIA [3] или AMD, применение которых типично в вычислительных узлах суперкомпьютеров, причем большей частью в суперкомпьютерах высшего диапазона производительности [4]. Другой пример, кроме GPU, в вычислительных узлах суперкомпьютеров применяются и суперскалярные ядра (CPU), что обеспечивает лучшую адаптируемость узлов к разному типу вычислений — такая особенность

архитектуры узлов называется гетерогенностью. Во встроенных вычислителях ситуация с адаптируемостью похожа. CPU применяются для управления вычислениями и для вычислений с низким параллелизмом по данным, но с хорошей пространственно-временной локализацией обращений к памяти. Характерный пример такого совместного использования разнотипных процессорных ядер во встроенном вычислителе — новейшая система на кристалле NVIDIA Xavier, содержащая 8 ядер ARMv8 и 512 CUDA-ядер графического процессора Volta [3,5].

При разработке СнК NM6408, первоочередными были вопросы организации адекватной работы с памятью и общей организации вычислений, соответствующих приложениям для встроенных систем. Исследования профилей работы с памятью для наиболее актуальных приложений, таких как первичная обработка сигналов и нейросети показывают, что обращения к памяти в этих приложениях обладают либо плохой пространственной локализацией при хорошей временной (обработка сигналов), либо наоборот (нейронные сети) [6,7]. Таким образом, обычное применение кэш-памятей с малыми задержками выполнения обращений к ним для таких задач оказывается практически бесполезным, поскольку применение кэш-памятей предполагает одновременно хорошую и пространственную и временну́ю локализацию обращений к памяти. Если же одна или другая локализация плохая, то за счет кэш-памятей обычно не удается в достаточной степени обеспечить данными функциональные устройства, которые способны выполнять вычисления с высокой производительностью. В таких случаях при создании процессоров для суперкомпьютеров обычным решением является использование архитектуры, обеспечивающей высокий темп выполнения обращений к памяти. Такое решение позволяет задействовать большое количество функциональных устройств и получить высокую реальную производительность даже при наличии достаточно больших задержек выполнения каждого их этих обращений (задержка доступа во внешнюю динамическую память, на сегодняшний день, составляет десятки-сотни процессорных тактов).

Способностью обеспечивать мощный поток обращений к памяти обладают архитектуры векторных и мультитредовых процессоров, при этом память, в свою очередь, должна иметь сильное расслоение, чтобы в ней можно было одновременно обрабатывать много обращений. Такие архитектуры называют толерантными по производительности к большим задержкам обращений к памяти. Кроме того, известно, что при наличии в приложении высокого параллелизма по данным такие архитектуры не уступают архитектурам, использующим кэш-памяти, на задачах с одновременно хорошей пространственной и временной локализацией, а такие приложения также встречаются и для встроенных систем.

Для обеспечения наибольшей производительности процессорных ядер NMC4, используемых в СнК NM6408, была выбрана классическая архитектура векторного процессора, с конвейерными функциональными устройствами, векторными регистрами, а также внутренней памятью с большим расслоением. Такая архитектура была впервые применена еще во второй половине 70-х годов в американском суперкомпьютере Cray-1 и в модифицированном виде используется до сих пор, особенно успешно в векторных микропроцессорах SX-ACE японской фирмы NEC [8] и при построении специальных мультитредово-векторных суперкомпьютеров СВ-класса [4].

Кроме наиболее требовательных по производительности нейросетевых задач и задач обработки сигналов, перед современными встроенными системами также стоят и вычислительные задачи, лучше решаемые на обычных универсальных CPU. По причине наличия таких задач, а также для выполнения задач управления и поддержки операционной системы, в состав СнК NM6408 были включены скалярные RISC ядра с популярной архитектурой ARM, в данном случае Cortex A5, что обеспечивает гетерогенность этой системы.

Толерантность к задержкам обращений к памяти, энергоэффективность и гетерогенность — главные цели, поставленные при разработке СнК NM6408. Эти цели достигаются за счет следующих принятых решений:

- для обеспечения высокого темпа обработки обращений от векторных функциональных устройств внутренняя память имеется при каждом функциональном устройстве и сильно расслоена;
- локализация памяти при функциональных устройствах одновременно способствует обеспечению высокой энергоэффективности, поскольку эти памяти имеют небольшой размер, что резко сокращает затраты энергии на обращение к памяти [1];
- для сокращения количества обращений к памяти в векторных устройствах введено большое количество векторных регистров, которые используются как FIFO-буфера при передаче данных между векторными устройствами минуя банки внутренней памяти;
- для обеспечения подкачки и выгрузки данных между памятями с высоким темпом выполнения обращений (близкими к процессорному ядру) и менее быстрыми памятями более высоких уровней иерархии (удалёнными от процессорного ядра) используется большое количество автономно работающих блоков прямого доступа к памяти (ПДП), часть из которых применяется для межкристальных обменов при построении сетей из СнК;
- простота работы с иерархической памятью обеспечивается её адресуемостью через единое адресное пространство;
- для повышения количества операций, выполняемых за такт, функциональные векторные устройства реализованы в виде динамически реконфигурируемых сложно-функциональных устройств, способных выполнять до восьми операций за такт;
- синхронизация вычислений в скалярных и векторных ядрах процессора поддерживается введёнными командами эксклюзивной работы с памятью и развитой многоярусной системой межпроцессорных прерываний.

Далее рассмотрим, как перечисленные концептуальные архитектурные принципы реализуются в СнК NM6408.

Структурная схема СнК

Разработанная СнК NM6408 содержит 21 процессорный узел (ПУ), имеет пять интерфейсов с внешней памятью типа DDR3, интерфейс с хост-процессором на базе PCIe2.0 и четыре высокоскоростных интерфейса для связи с внешними процессорными системами. В состав СнК входят 16 идентичных ПУ на базе NMC4, которые обеспечивают суммарную производительность системы в 512 Gflops, и пять ПУ на базе Cortex-A5, предназначенных для управления системой. Один из ПУ на базе Cortex-A5 осуществляет общее управление системой, а остальные 20 ПУ разбиты на четыре одинаковых процессорных кластера, содержащие по четыре ПУ на базе NMC4 и один управляющий ПУ на базе Cortex-A5. На рис. 1 приведена упрощенная общая структурная схема СнК NM6408, на которой выделены процессорные узлы, включающие процессорные ядра и локализованные при них памяти, распределенная система коммутаторов, а также внешние интерфейсы.

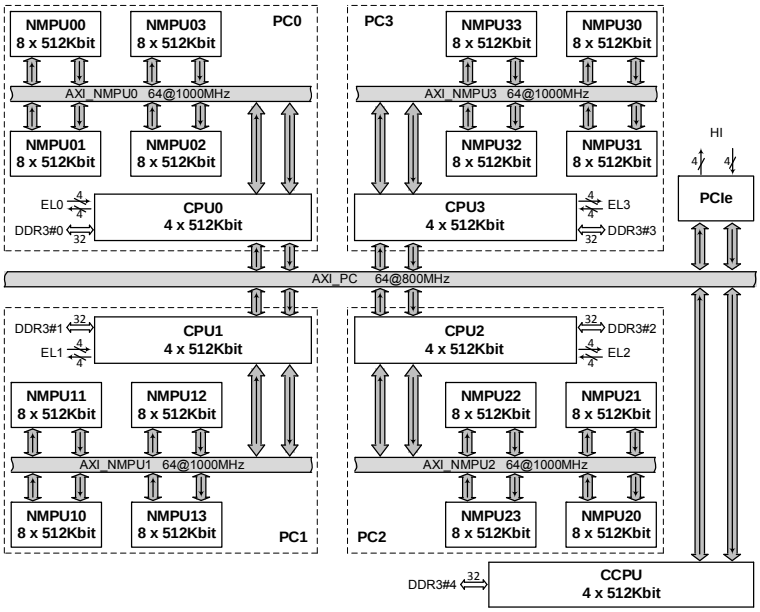

Рис. 1. Упрощенная общая структурная схема СнК NM6408

CCPU (Central Cortex Processing Unit) — центральный управляющий ПУ на базе ядра Cortex-A5 (объемы кэш-памяти данных и команд L1 по 32 КБ каждая, общая кэш-память L2 – 512 КБ). Содержит 4 банка внутренней памяти по 512 Кбит каждый (общий объем 256 КБ), работает на частоте 800 МГц, подключен к главному коммутатору AXI_PC и имеет интерфейс с DDR3 памятью по 32-разрядной шине данных (DDR3#4). Предназначен для общего управления СнК NM6408.

PCi (Processing Cluster) — процессорный кластер (i = 0...3), состоящий из 4-х ПУ на базе векторных ядер NMC4 (NMPUij, j = 0...3) и локального управляющего ПУ на базе скалярного ядра Cortex-A5 (CPUi). Суммарная пиковая производительность одного кластера составляет 128 Gflops (32-х разрядные числа), а общий объем внутренней памяти всех пяти ПУ кластера — 18 Мбит (2.25 МБ). Четыре ПУ NMPUij и ПУ CPUi подключены к общей локальной шине AXI_NMPUi.

CPUi (Cortex Processing Unit) — локальный управляющий ПУ кластера PCi на базе Cortex-A5 (содержит L1 кэш-памяти данных и команд, объёмом по 32КБ каждая), содержит 4 банка внутренней памяти по 512 Кбит (общий объем 256 КБ), работает на частоте 800 МГц, имеет интерфейс с DDR3 памятью по 32-разрядной шине данных (DDR3#i), а также высокоскоростной интерфейс ELi. Предназначен для локального управления четырьмя NMPUij, входящими в кластер и поддержки обмена данными через внешние интерфейсы ELi и DDRk.

ELi (External Link) — внешний последовательный 4-проводной дуплексный интерфейс, работающий на частоте 5 ГГц и предназначенный для обмена данными с другими СнК NM6408 в построенных на их основе многопроцессорных системах. Каждый такой интерфейс обеспечивает пропускную способность в 2 ГБ/с в каждую сторону.

NMPUij (NeuroMatrix Processing Unit) — векторный вычислительный ПУ (j = 0…3) на базе NMC4. Каждый такой узел содержит 8 банков внутренней памяти по 512 Кбит каждый (общий объем 512 КБ) и работает на частоте 1 ГГц. Основные вычислительные ресурсы СнК сосредоточены в ПУ данного типа.

DDRk — один из пяти (k = 0…4) интерфейсов с внешней памятью типа DDR3-1600 с 32-разрядной шиной данных, обеспечивающей пропускную способность 6.4 ГБ/с. Максимальный объем внешней памяти, подключаемой к одному такому интерфейсу, составляет 1ГБ.

HI (Host Interface) – интерфейс с хост-процессором, реализованный на стандартном 4-канальном дуплексном PCIe2.0 интерфейсе с пропускной способностью 2 ГБ/с в каждую сторону.

Распределенная система коммутаторов, соединяющая все процессорные узлы, построена в соответствии с AMBA AXI 3.0 протоколом.

Далее рассмотрим компоненты СнК NM6408, начав с векторного вычислительного процессорного узла NMPU, на рис. 2 представлена его структурная схема.

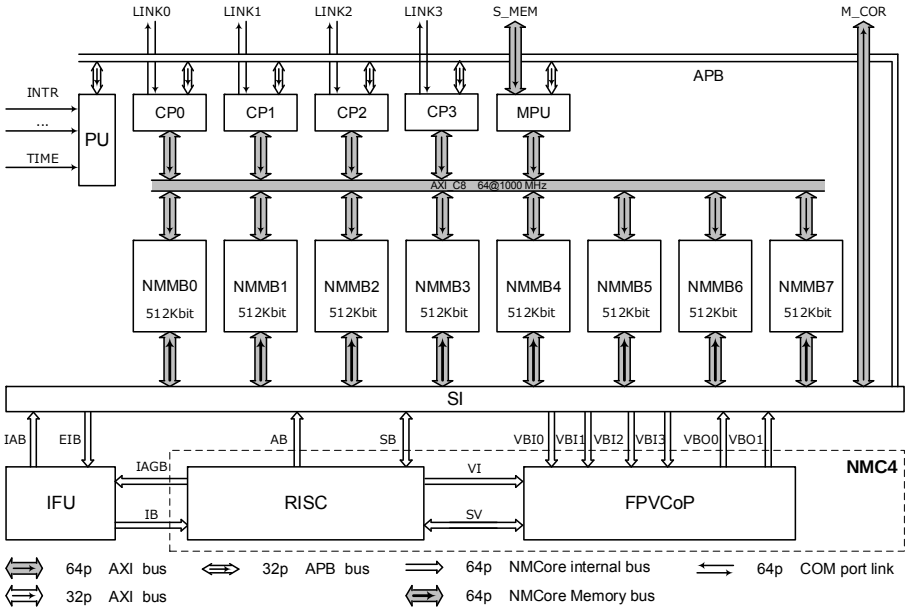

Рис. 2. Структурная схема векторного вычислительного ПУ NMPU

NMC4 — процессорное ядро со скалярным RISC процессором с VLIW-архитектурой и матрично-векторным сопроцессором арифметики с плавающей точкой (FPVCoP) из семейства NeuroMatrix [9,10]. Это ядро связано через системный интегратор SI с банками внутренней памяти NMMB0… NMMB7 посредством следующих 64-разрядных шин:

- шина команд IB, через которую передаются 64-разрядные слова с 32-разрядными скалярными VLIW-командами и векторными командами из блока преднакачки команд IFU;
- шина IAGB, через которую RISC ядро выдает адреса переходов блоку преднакачки IFU;
- шина AB, через которую в системный интегратор SI выдаются адреса скалярных данных и начальные адреса векторов с их длинами и шагом выборки элементов из памяти, которые попадают в адресные генераторы блока SI;
- шина SB, служит для обмена скалярными данными;
- шины входных векторных данных VBI0…VBI3;
- шины выходных векторных данных VBO0 … VBO1.

RISC процессор дешифрирует скалярные и векторные команды, а также отслеживает информационные зависимости скалярных команд и часть зависимостей векторных команд. Далее векторные команды передаются в векторный сопроцессор FPVCoP по шине VI, в котором происходят дальнейшие проверки информационных зависимостей и конвейерное выполнение этих команд. В векторный сопроцессор из RISC процессора также передаются скалярные данные по шине SV.

Сопроцессор FPVCoP содержит одно устройство преобразования форматов векторов и четыре векторных арифметических ядра. Каждое векторное арифметическое ядро имеет по 8 векторных регистров, каждый из которых содержит 32 64-х разрядных слова. Эти векторные регистры используются как FIFO-очереди для передачи данных непосредственно между векторными устройствами без обращений к памяти, а также портами считывания/записи векторов. Устройство преобразования форматов векторов также содержит входную и выходную FIFO-очереди. Архитектура ядра NMC4 позволяет организовать эффективную обработку потоков данных с зацеплением выполнения векторных операций (stream-based processing).

Векторный сопроцессор FPVCoP может выполнять до 32-х операций с числами с плавающей точкой одинарной точности за такт, что достигается за счет реализации арифметических векторных блоков в виде реконфигурируемых сложно-функциональных устройств, способных выполнять не только простые поэлементные операции над векторами, но и операции над комплексными числами, а также умножение с накоплением матрицы 2x2 на 2-х компонентный вектор за один процессорный такт.

Важной характеристикой ПУ NMPU является баланс производительности и пропускной способности памяти, который определяется следующими свойствами:

- пиковой производительностью сопроцессора FPVCoP 32 Gflops для чисел с плавающей точкой (FP32);
- пропускной способностью банков внутренней памяти NMMB со стороны системного интегратора SI в 64 ГБ/с;
- общей пропускной способностью каналов между системным интегратором SI и сопроцессором FPVCoP, которая на считывание составляет 32 ГБ/с, а на запись – 16 ГБ/с.

Таким образом, баланс пиковой производительности сопроцессора FPVCoP к пропускной способности на считывание из системного интегратора SI составляет 32 ГБ/с / 32Gflops = 1 Б/flop. Баланс пиковой

производительности сопроцессора FPVCoP к пропускной способности системного интегратора SI по доступу к внутренним банкам памяти NMMB составляет 64ГБ/с / 32Gflops = 2 Б/flop.

IFU (Instruction Fetch Unit) — блок предвыборки команд, который выстраивает в единую очередь команды, считываемые из внутренней или внешней памяти через системный интегратор SI по шине EIB в соответствии с адресом, выставляемым по шине IAB. Блок IFU содержит кэш-память команд объёмом 8КБ.

SI (System Integrator) — системный интегратор, через него осуществляется доступ к банкам внутренней памяти NMMB0 … NMMB7, к банкам памяти других ПУ через AXI порт M_COR, а также к регистрам блоков, расположенных на периферийной шине APB. Системный интегратор содержит адресный генератор для выборки команд RISC процессора, 8 адресных генераторов для доступа к элементам векторов из любой памяти СнК NM6408, коммутатор адресных шин, коммутатор векторных данных, логические блоки контроля и управления ресурсами, необходимыми для выполнения векторных команд загрузки и выгрузки данных.

NMMB0-NMMB7 (NM Memory Bank) — восемь банков внутренней памяти объёмом по 512 Кбит каждый. Каждый банк имеет два порта доступа, один - со стороны системного интегратора SI, другой – со стороны коммуникационных портов CP0 … CP3 и блока защиты памяти MPU. Физически банк памяти реализуется двумя одновходовыми полубанками SRAM-памяти, на которые соответственно отображаются слова банка с четными и нечетными адресами, что позволяет банку обслуживать два запроса за такт в случае несовпадения младших разрядов адресов запросов. Если младшие разряды совпадают, то обслуживается только один запрос в соответствии с установленным приоритетом. Локализация банков памяти вблизи векторного сопроцессора и их расслоение по банкам и полубанкам позволяет достичь высокого темпа выполнения операций с памятью за счет их распараллеливания и тем самым обеспечить высокую реальную производительность процессора (толерантность к задержкам по памяти).

MPU (Memory Protect Unit) – блок защиты банков памяти NMMB, обслуживает внешние запросы на доступ во внутреннюю память ПУ по AXI порту S_MEM. Если запрос попадает в разрешённую область памяти, он обслуживается обычным образом, если нет, то он блокируется с формированием соответствующего прерывания, а генерирующему запрос устройству выдается признак ошибки.

CP0 - CP3 (Communication Port) — коммуникационные порты, которые обеспечивают обмен данными по двунаправленным 64-разрядным каналам связи LINK0…LINK3 и доступ в режиме ПДП к внутренним банкам памяти NMMB0…NMMB7.

PU (Peripheral Unit) — блок периферийных устройств, содержащий контроллер прерываний и таймеры.

Перейдем теперь к рассмотрению основного структурного компонента СнК NM6408 — процессорного кластера PC, структурная схема которого приведена на рис. 3, а ниже перечислены его основные блоки.

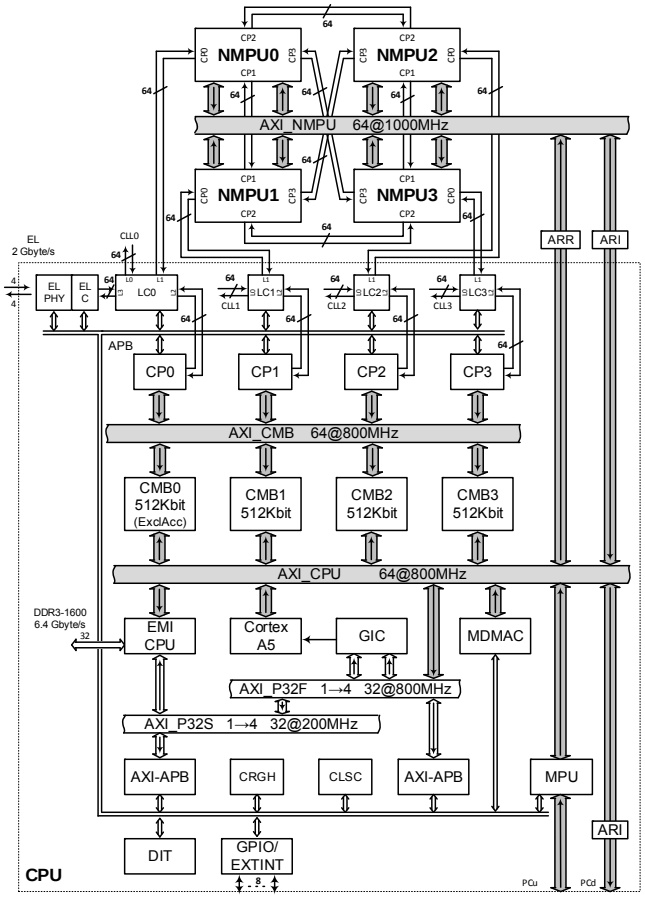

Рис. 3. Структурная схема процессорного кластера PC

NMPU0…NMPU3 — четыре процессорных узла, которые соединены непосредственно «каждый с каждым» тремя 64-разрядными полнодуплексными коммуникационными портами CP1, CP2 и CP3. Обмен данными через эти порты выполняется с использованием механизма ПДП. Эти ПУ дополнительно соединены через коммуникационные порты CP0 с локальным для кластера управляющим ПУ CPU через коммутаторы коммуникационных портов LC0...LC3. Дополнительно ПУ NMPU кластера соединены между собой через коммутатор AXI_NMPU, который также обеспечивает произвольный доступ к памятям всех других ПУ системы.

CPU — управляющий ПУ кластера, содержащий скалярный процессор ARM Cortex-A5, внутреннюю память CMB0…CMB3, а также интерфейсы для интеграции кластера в вычислительную систему. Для связи с другими кластерами и центральным управляющим ПУ CCPU используются высокоскоростные коммуникационные порты CLL0…CLL3, которые обеспечиваются коммутаторами коммуникационных портов LC0...LC3. Кроме этого, коммутатор LC0 обеспечивает работу по внешнему коммуникационному порту EL. Для связи внутри кластера используется AXI коммутатор AXI_CPU.

Cortex-A5 — управляющий скалярный процессор ARM Cortex-A5 с кэш-памятью L1 команд и данных (32КБ+32КБ), который имеет доступ ко всем внутренним банкам NMMB узлов NMPU, банкам памяти CMB, а также управляет всеми периферийными устройствами CPU через шину APB. С целью для защиты внутренних ресурсов NMPU этот управляющий процессор не имеет доступа к регистрам периферийных устройств внутри блоков NMPU.

GIC – контроллер прерываний, который поддерживает до 64-х прерываний с программно настраиваемыми адрес-векторами и приоритетами.

CMB0…CMB3 — четыре банка статической памяти объёмом по 512 Кбит каждый. Банк CMB0 дополнительно поддерживает эксклюзивный тип доступа в память.

CP0…CP3 — четыре коммуникационных порта, соединенных с узлами NMPU через коммутаторы коммуникационных портов LC0...LC3. Для доступа коммуникационных портов к банкам памяти CMB0…CMB3 служит коммутатор AXI_CMB, обеспечивающий доступ каждого порта к любому банку памяти.

EL — внешний 4-проводной последовательный дуплексный интерфейс, работающий на частоте 5 ГГц и предназначенный для обмена данными с другими СнК NM6408. Этот интерфейс позволяет соединять на печатной плате несколько СнК образуя, таким образом, высокопроизводительные вычислительные устройства. Контроллер интерфейса EL соединяется через порт L3 коммутатора коммуникационных портов LC0, который обеспечивает его соединение либо с коммуникационным портом CP0 узла NMPU0, либо с портом CP0 кластера PC, либо с внешним устройством через канал связи CLL0.

MDMAC — высокопроизводительный контроллер ПДП для пересылок «память-память». Основной задачей этого контроллера является обмен данными между внешней памятью DDR3 и внутренними банками кластера CMB0…CMB3. Контроллер также имеет доступ к внутренним банкам памяти NMMB всех NMPU.

MPU — блок защиты памяти. Управляющий узел CPU может запрещать доступ внешних устройств к своим внутренним банкам CMB и банкам NMMB внутри своих процессорных узлов NMPU. Если запрос попадает в разрешённую область памяти, он обслуживается обычным образом, если нет, то он блокируется с формированием соответствующего прерывания, а генерирующему запрос устройству выдается признак ошибки.

GPIO/EXTINT — универсальный блок интерфейсов общего назначения. Он управляет работой восьми внешних выводов, каждый из которых может программно настраиваться на функционирование в качестве выводов общего назначения (GPIO), внешних входов прерывания (EXTINT), а пара выводов GPIO может реализовать интерфейс «запрос-подтверждение».

CLSC — системный контроллер кластера. Он содержит идентификатор своего кластера, а также служит для генерации запросов на прерывание к другим кластерам.

CRGH — генератор тактовых сигналов и сигналов сброса. Он предназначен для управления генерацией синхросигналов и сигналов сброса для NMPU0…NMPU3**.**

Представленная ранее структурная схема СнК NM6408 (см. рис. 1), как отмечалось, была упрощена для выделения основных особенностей этого устройства. Главные упрощения — отсутствовали соединения ПУ посредством множества быстрых каналов межпроцессорного обмена из соединенных друг с другом коммуникационных портов работающих по принципу ПДП, а также отсутствовали блоки, реализующие внутренние и внешние интерфейсы. Уточненная структура без этих упрощений приведена на рис. 4.

Одним из основных принципов построения микросхемы был принцип иерархии, поэтому структурная схема верхнего уровня СнК напоминает структурную схему процессорного кластера PC. На рисунке показано, что процессорные кластеры PC0...PC3 соединены непосредственно «каждый с каждым» тремя 64-разрядными полнодуплексными коммуникационными портами CLL1, CLL2 и CLL3. Обмен данными через эти порты выполняется с использованием механизмов ПДП с пропускной способностью 6.4 ГБ/с в каждую сторону.

Каждый из процессорных кластеров PC0...PC3 также соединен с центральным управляющим ПУ CCPU своим коммуникационным портом CLL0. Со стороны CCPU эти порты CLL0 подключены к коммуникационным портам CP0…CP3 соответственно. Это важно, т.к. быстрые каналы межпроцессорного обмена образуются двумя коммуникационными портами, соединенными либо непосредственно, либо через один или несколько коммутаторов коммуникационных портов LC. Каждый такой порт подключен одним концом к памяти, а другим — к соответствующему ему в быстром канале коммуникационному порту.

Дополнительно кластеры PC0...PC3 соединены между собой через коммутатор AXI_PC, обеспечивающий им доступ во внутренние банки памяти друг друга, поскольку вся внутренняя память СнК находится в едином

адресном пространстве. Через этот же коммутатор кластеры имеют доступ к статической памяти управляющего ПУ CCPU — банкам CCMB0…CCMB3. Банк памяти CCMB0 имеет важную особенность — он поддерживает выполнение эксклюзивных операций, при помощи которых в данном блоке могут быть реализованы семафорные переменные для синхронизации вычислительных процессов (отметим, что такой же особенностью обладает банк CMB0 внутри PC).

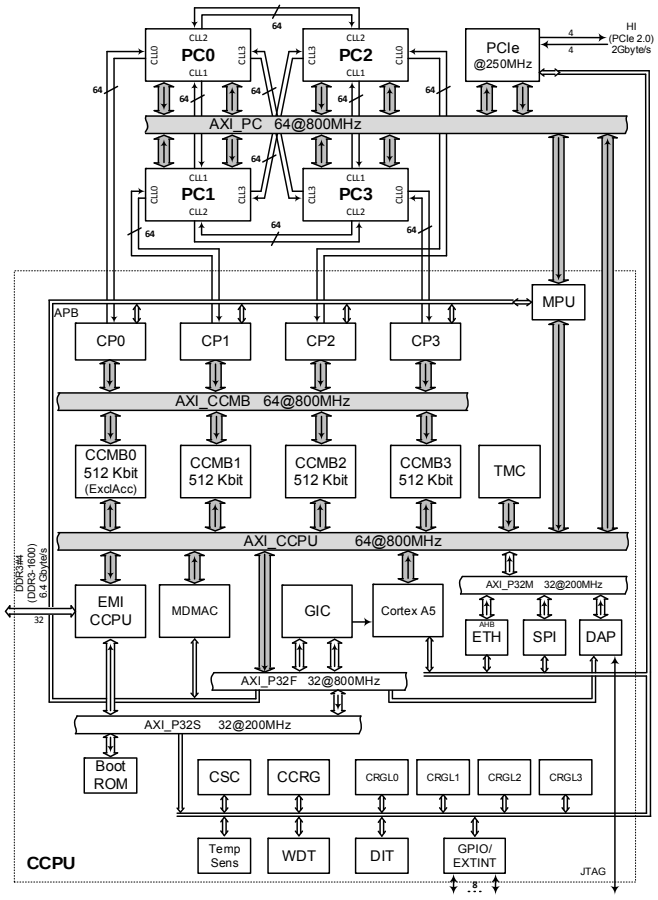

Рис. 4. Структурная схема СнК NM6408

Центральный управляющий ПУ CCPU по своему составу похож на ПУ CPU, использованный внутри PC, но имеет следующие особенности:

- управляющий Cortex A5 имеет L2 кэш-память команд и данных объемом 512КБ в дополнение к двум L1 кэш-памятям команд и данных;
- дополнительные внешние низкоскоростные интерфейсы Ethernet 10/100 и SPI;
- блоки интервальных таймеров DIT и сторожевого таймера WDG;
- загрузочное ПЗУ — BootROM;
- блок контроллера измерения температуры кристалла с температурным датчиком (TempSens);
- CCRG — генератор тактовых сигналов и сигналов сбросов, используемых в CCPU;
- CRGL0…CRGL3 — четыре генератора тактовых сигналов и сигналов сброса для кластеров PC0…PC3;
- блок DAP согласования с внешним отладочным JTAG интерфейсом (реализована стандартная аппаратная схема, в соответствии с технологией ARM CoreSight, при которой кроме стандартной пошаговой отладки, поддерживается сбор трасс выполнения программ со всех пяти ядер Cortex A5, трассы собираются в реальном времени в буфере TMC, после чего их можно выгрузить через порт JTAG и анализировать на персональном компьютере, используя стандартные программно-аппаратные решения фирмы ARM);

В микросхеме СнК NM6408 реализована возможность управления потребляемой мощностью посредством отключения синхросигналов с отдельных процессорных кластеров PC или же только с процессорных узлов NMPU внутри кластеров PC. Основной сложностью здесь была необходимость аппаратно отслеживать отложенные запросы (незаконченные транзакции на шине AXI, когда, например, адрес выдан, а данные ее не получены) в системе коммутаторов, т.к. отключение синхросигнала с блока в котором присутствует отложенный запрос может привести к зависанию всей системы. Для решения данной задачи было выполнено каскадное соединение всех девяти CRG-генераторов, они реализуют между собой интерфейс «запрос-подтверждение» для управления генерацией синхросигналов, а на границах отключаемых блоков находятся специальные блоки изоляторов (ARI) и отражателей запросов (ARR), которые позволяют гарантировать корректную работу блока и всей системы в случае остановки/подачи синхросигнала на эти блоки.

Механизмы межпроцессорного взаимодействия

Для мультипроцессорных систем особенно важными являются средства взаимодействия параллельных процессов и их синхронизации, поскольку возникающие накладные расходы влияют на допустимую гранулированность параллелизма, а, следовательно, и на развиваемую реальную производительность. Далее выделены такие средства, аппаратно реализованные в СнК NM6408:

- имеется возможность доступа любого процессора и любого контроллера ПДП ко всем словам памяти через глобальное адресное пространство, обеспечиваемого системой коммутаторов, соединяющих все процессоры и банки памяти СнК и построенной в соответствии с AMBA AXI протоколом;
- банки памяти CMB0 процессорных кластеров и банк CCMB0 центрального ПУ CCPU поддерживают команды эксклюзивного доступа, которые могут использоваться для реализации семафорных переменных для синхронизации параллельных процессов при работе с критическими разделяемыми ресурсами, прежде всего, участками памяти;
- взаимодействие параллельных процессов облегчено возможностью асинхронной передачи пакетов данных через каналы межпроцессорного обмена (коммуникационные порты, соединенные парами, позволяют вести обмен данными в режиме ПДП между любыми банками внутренней памяти различных ПУ);
- для синхронизации процессов, протекающих в различных ПУ, предусмотрена многоярусная система межпроцессорных прерываний.

Особенности маршрута физического проектирования и результаты

На основании результатов исследования предыдущей микросхемы NM6407, которая содержала всего один процессорный узел NMPU, были оценены мощность потребления и степень интеграции. Как и ожидалось, оценка показала, что СнК NM6408 будет иметь высокую степень интеграции и значительную потребляемую мощность. В связи с этим в качестве технологического базиса была выбрана технология с проектными нормами 28нм и конструкция кристалла FlipChip.

Известно, что задачей физического проектирования является разработка топологии, которая включает размещение логических блоков микросхемы на кристалле при условии выполнения определенных электрических, технологических и конструктивных ограничений, а также требований по быстродействию (рабочей частоте).

Изначально следовало выбрать тип используемого маршрута физического проектирования (последовательность шагов): без использования иерархии (flat, плоский) или иерархический. Известно, что маршрут без использования иерархии проще и предпочтительней в применении, если степень интеграции микросхемы невысока, поскольку при таком маршруте отсутствуют дополнительные трудозатраты на разбиение проекта на части.

Если разрабатываемая микросхема имеет высокую степень интеграции, то такой маршрут приводит к очень долгому (несколько суток) выполнению отдельных шагов маршрута физического проектирования в САПРах и является неприменимым на практике. Эта проблема решается дроблением проекта на части и изменением типа используемого маршрута на иерархический (bottom-up, «снизу-вверх»), в котором блоки микросхемы разрабатываются раздельно. Кроме ускорения работы САПРов за счет уменьшения размеров блоков, при таком подходе возможна параллельная разработка топологии СБИС несколькими инженерами, что также ускоряет процесс проектирования.

К недостаткам такого иерархического подхода можно отнести возникающие накладные расходы на дробление проекта (например, написание sdc-файлов временных ограничений для блоков), а также опасность возникновения возможных проблем, вызванных несогласованностью решений, принятых при разработке топологии отдельных блоков. К таким проблемам можно отнести:

- неправильное распределение бюджетов задержек сигнальных линий между блоком и его логическим окружением на верхнем уровне (сборкой верхнего уровня) в файлах описания временных ограничений (sdc) для блока, вследствие чего будет невозможно достичь планируемого быстродействия СнК;
- несогласованность топологии иерархического блока и топологии сборки верхнего уровня, что может привести к избыточности межсоединений, причём это особенно критично, если иерархический блок используется несколько раз в проекте.

В силу описанных причин, при разработке СнК NM6408 использовался комбинированный иерархический маршрут физического проектирования (top-down, «сверху-вниз»), применяя который возможно избежать упомянутых выше проблем, а также удостовериться в физической реализуемости проекта в соответствии с параметрами ТЗ на ранних этапах разработки.

Основное отличие комбинированного иерархического маршрута в том, что необходимо иметь полное описание микросхемы, включая все иерархические блоки, на уровне связей элементов технологической библиотеки (далее — нетлист) до начала физического проектирования, тогда как в обычном иерархическом маршруте для начала их физического проектирования ограничиваются лишь наличием нетлистов отдельных блоков. Это требование задерживает начало этапа физического проектирования, но зато позволяет избежать множества проблем в процессе проектирования.

Далее более подробно описываются шаги маршрута, использованного при разработке СнК NM6408.

На первом шаге было произведено логическое дробление проекта на иерархические блоки, размер которых обеспечивал бы разумное время выполнения каждого шага маршрута проектирования для каждого блока (до 1-2 суток на используемых нами серверах — это приблизительно 1.5 млн стандартных ячеек на блок). Структурно СнК NM6408 содержит четыре одинаковых блока процессорных кластеров (PC), которые удовлетворяют требованию по максимальной степени интеграции, описанному выше, поэтому именно этот блок был выбран в качестве иерархического.

Логический синтез иерархических блоков PC и оставшейся части микросхемы (CCPU, PCIe и коммутатор верхнего уровня) был произведен раздельно. Соответственно, таким образом, были получены два нетлиста, объединение которых позволяет получить полный нетлист проекта. Еще раз отметим, что наличие полного нетлиста проекта — это необходимое условие для использования комбинированного иерархического маршрута «сверху-вниз».

Следующие шаги физического проектирования выполнялись с базой данных САПРа, содержащей полный нетлист проекта:

- Разработан примерный план топологии (floorplan) в соответствии со степенью интеграции блоков верхнего уровня. На этом шаге также был разработан примерный план внутренней топологии иерархического блока PC, в том числе выделены области размещения подблоков NMPU, CMB, ARM, DDR3 и EL и определены точные границы PC, чтобы обеспечить оптимальное соединение 4-х блоков процессорных кластеров на верхнем уровне и избежать возможных проблем с быстродействием полученной схемы в дальнейшем.
- Были размещены буфера ввода-вывода (IO buffer) и создана система распределения их питания.
- Создан массив контактных площадок кристалла (bump array) и назначено соответствие между функциональными сигналами микросхемы и конкретными контактным площадками. На этом шаге для получения максимально однородного массива контактных площадок всей микросхемы важно учитывать, какой шаг и тип контактных площадок имеют СФ блоки используемых физических интерфейсов(PHY). Также важно учитывать симметрию расположения контактных площадок, относительно центра кристалла, чтобы разработанную топологию иерархического блока PC можно было использовать без изменений для всех четырех экземпляров процессорных кластеров в СнК.
- Разработана система распределения питания всей микросхемы (сетка питания). На этом шаге, как и на предыдущем, важно учитывать симметрию расположения шин распределения питания относительно центра кристалла.
- Были назначены области предпочтительного размещения сигнальных выводов иерархического блока PC вдоль его границ с верхним уровнем. Это было важно сделать для минимизации длины соединений на верхнем уровне и, в том числе, для минимизации длины соединений между четырьмя экземплярами блока PC.

Следующим шагом было выполнено разделение созданной на предыдущих шагах предварительной топологии на части (partitioning), физическое проектирование которых может выполняться несколькими инженерами параллельно вплоть до финальной стадии маршрута (получения GDSII-файла). Таким образом, общая база данных САПРа разделяется на две:

- Базу данных САПРа, содержащую данные об иерархическом блоке процессорного кластера PC (нетлист и примерный топологический план, включающий сетку питания, размещение сигнальных выводов и точные границы блока).
- Базу данных САПРа, содержащую данные о верхнем уровне проекта. Информация об иерархическом блоке PC включена в эту базу в виде набора упрощенных моделей, описывающих размеры блока и временные диаграммы на его границе.

В базе данных верхнего уровня проекта использовалась Interface Level Model (ILM) временна́я модель. Использование ILM модели вместо обычно используемого в таких случаях набора ETM-моделей в формате .lib (Extracted Timing Model), гарантирует более аккуратный анализ на границах иерархического блока. Кроме того, ILM-модель содержит информацию о шумах и перекрестных наводках (signal integrity) и обеспечивает легкую передачу результатов между разработчиками, занимающимися проектированием иерархического блока и разработчиками верхнего уровня микросхемы.

Наш предыдущий опыт разработки микросхем показывает, что финальный (signoff) статический анализ (STA) с использованием каких-либо временны́х моделей может быть неточным из-за краевых эффектов или ошибок в моделях — поэтому проводился плоский (без использования каких-либо моделей) финальный статический временной анализ микросхемы (flat STA). Такой подход резко снижает риски неточного/неправильного расчета, в случае же нахождения проблем достаточно выполнить одну или несколько итераций ручной или автоматической модификации топологии для исправления найденных нарушений (Engineering Change Order — ECO).

Финальный файл топологии, отправляемый на фабрику (GDSII), получен простой склейкой GDSII файлов, содержащих топологию верхнего уровня СнК и топологию иерархического блока PC. Беспроблемная склейка GDSII файлов возможна вследствие того, что первые шаги маршрута выполнялись для полного проекта и с учетом последующего дробления топологии на части.

Физическая верификация, состоящая из стандартных проверок норм проектирования (DRC), проверок электрических нарушений (ERC) и проверки соответствия между логическим описанием и электрической схемой (LVS), выполнялась для «склеенного» GDSII файла.

Как уже отмечалось, основное достоинство описанного маршрута в возможности ранней оценки реализуемости проекта, в возможности параллельной физической реализации иерархических блоков и при этом низких рисков возникновения различных проблем на стадии объединения проекта. Топология СнК NM6408 была разработана с использованием описанного выше маршрута. Слева на рис. 5 приведена разработанная топология базового иерархического блока — процессорного кластера РС, а справа приведена полная топология СнК.

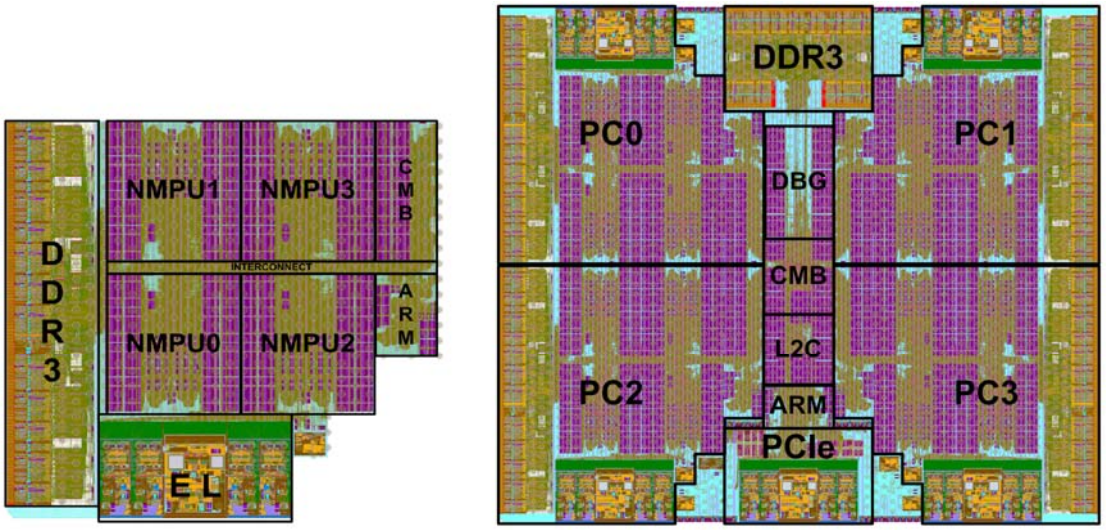

Рис. 5. Топология иерархического блока РС (слева) и полная топология СнК NM6408 (справа)

Основные характеристики полученной микросхемы:

- 16 векторных процессорных ядер с отечественной архитектурой NeuroMatrix, работающих на частоте 1ГГц и обеспечивающих суммарную пиковую производительность 512 Gflops (FP32) и 128 Gflops (FP64);
- 5 скалярных процессорных ядер ARM Cortex A5, работающих на частоте 800МГц, из которых четыре ядра имеют кэш-память команд и данных по 32КБ, а пятое ядро дополнительно общую L2 кэш-память 512КБ;
- внутренняя иерархическая память с сильным расслоением, общим объемом 74Мбит (9,25 МБ);
- пять 32-х разрядных интерфейсов с внешней памятью типа DDR3 пропускной способностью 6.4 ГБ/с каждый;
- четыре дуплексных высокоскоростных коммуникационных порта с пропускной способностью 2 ГБ/с в каждом направлении каждый, позволяющие строить вычислительные системы на базе нескольких СнК;
- PCIe2.0 x4 хост-интерфейс;
- периферийные порты Ethernet 10/100, SPI, GPIO;
- отладка программ выполняется стандартными программно-аппаратными средствами, совместимыми с технологией ARM CoreSight через JTAG порт;
- площадь кристалла ~83 мм^2, 1.05 млрд транзисторов по технологии 28нм;
- корпус— BGA1444, шаг выводов 1мм, 40x40 мм FlipChip;
- максимальная потребляемая мощность – не более 35Вт;
- удельная производительность (энергоэффективность) – 14.6 Gflops/Вт;
- удельная мощность потребления – 0.42 Вт/мм^2.

Новизна полученных результатов

Авторы считают, что в данной работе новыми являются следующие положения и результаты:

- реализованы архитектурные принципы (векторные операции и расслоение памяти) обеспечения толерантности к задержкам обращений к памяти, что позволяет обеспечивать высокую реальную производительность в условиях характерных для задач с плохой пространственной или временной локализацией обращений к памяти;
- реализована гетерогенная архитектура за счет одновременного применения скалярных и векторных ядер, что адекватно отражает характерное разнообразные задач, стоящих перед СнК;
- память, с которой работают ядра, имеет иерархическую организацию, ее элементы локализованы в процессорных узлах, что позволяет добиться высокой энергоэффективности;
- простота работы с иерархической памятью обеспечивается её адресуемостью через единое адресное пространство;
- реализована развитая система широких каналов межпроцессорного обмена типа «точка-точка», работающих в режиме ПДП для эффективной передачи информации между различными уровнями иерархических банков памяти с разными скоростными характеристиками;

- реализована поддержка различных по функциональным и скоростным характеристикам механизмов межпроцессорного взаимодействия для эффективной работы многопроцессорной системы;
- в векторных ядрах NMC4 реализован принцип потоковой обработки с передачей данных непосредственно через векторные регистры, минуя обращения к памяти;
- получен опыт проектирования большой СнК по технологии FlipChip с проектными нормами 28нм.

Заключение и перспективы развития

Микросхема NM6408, разработанная специалистами ЗАО НТЦ «Модуль» – большая система на кристалле, реализующая передовые архитектурные концепции обеспечения толерантности к задержкам обращений к памяти и гетерогенной организации с использованием разнотипных ядер. Первые инженерные образцы получены и проходят испытания.

Ближайший этап – освоение описанных архитектурных особенностей в программном обеспечении и пользователями. Концепции проведения этих работ составлены и детализированы, находятся в стадии реализации. Программирование будет осуществляться на языке C/C++ и ассемблере. Для управления параллельными процессами и их взаимодействием разрабатывается собственная библиотека функций на основе библиотек MPI (Message Passing Interface) и SHMEM (Cray Research «shared memory» library), но сильно упрощенная и адаптированная под имеющиеся архитектурные особенности СнК NM6408 (наличие коммуникационных портов, ярко выраженной иерархии системы памяти, многоярусной системы прерываний, наличия высокопроизводительных MDMAC контроллеров ПДП). Особое внимание уделено средствам отладки, трассировки и анализа процессов выполнения параллельных программ. Разрабатываются имитационные модели разного уровня детализации для тонкой машинно-зависимой оптимизации программ.

Уже востребована разработка новых микросхем типа СнК NM6408, но с повышенной в разы пиковой производительностью и улучшенной архитектурой. Кроме простого увеличения количества процессорных узлов в СБИС такого типа новых поколений, в качестве рассматриваемых вариантов улучшения архитектуры можно назвать: усиление скалярного RISC процессора в ядрах NMC и введение новых типов операций в векторном сопроцессоре.

Литература

[1] Dally W., Balford J. *et al*. An Energy-Efficient Processor Architecture // IEEE Computer Architecture Letters. — Vol. 7, No.1. — Jan. 2008. — P. 29—31.

[2] Nowatzki T., Wright G. *et al.* Pushing the Limits of Accelerator Efficiency While Retaining Programmability // IEEE High performance computer architecture conference. — 2016. — 13 pp.

[3] Durant L., Harris M. *et al*. Inside Volta: The World's Most Advanced Data Center GPU — 10 may 2017. URL: https://devblogs.nvidia.com/parallelforall/inside-volta (дата обращения: 01.09.2017).

[4] Эйсымонт Л.К. Гибридная стратегия развития элементной базы // Открытые системы. СУБД. — 2017. — № 2. — С. 8—11. URL: https://www.osp.ru/os/2017/02/13052216 (дата обращения: 01.09.2017).

[5] Mujtaba H. NVIDIA Announces Xavier Tegra SOC – Features Volta GPU With 7 Billion Transistors, 512 CUDA Cores and 8 ARM64 Custom Cores — Sep 28, 2016. URL: http://wccftech.com/nvidia-xavier-soc-tegra-volta-gpu-announced (дата обращения: 01.09.2017).

[6] Weinberg J. Quantifying locality in the memory access patterns of HPC Applications — University Of California, San Diego. — 2005. — 50 pp.

[7] Murphy R.C., Kogge P.M. On the Memory Access Patterns of Supercomputer Applications: Benchmark Selection and Its Implications // IEEE Transactions on Computers — Vol. 56, No.7. — July 2007. — 9 pp.

[8] Egawa R. *et al*. Early evaluation of the SX-ACE Processor — SC14 — November, 2014. — 2 pp.

[9] Черников В.М., Виксне П.Е., Шелухин А.М., Шевченко П.А., Панфилов А.П., Косоруков Д.Е., Черников А.В. Семейство процессоров обработки сигналов с векторно-матричной архитектурой NeuroMatrix // Электронные компоненты. — 2006. — № 6. — С. 79—84.

[10] Черников В.М., Виксне П.Е., Шелухин А.М., Панфилов А.П. Отечественные высокопроизводительные процессоры цифровой обработки сигналов векторно-матричной архитектуры, перспективы развития // Материалы конференции «Перспективы развития высокопроизводительных архитектур. История, современность и будущее отечественного компьютеростроения».— М.: ИТМиВТ им С.А.Лебедева РАН. — 2008. — Вып. №1. — С. 52—59.