

УДК: 004.2 + 004.4  
DOI 10.53403/9785951505071\_2022\_185

## АРХИТЕКТУРА СПЕЦИАЛИЗИРОВАННЫХ БОЛЬШИХ ИНТЕГРАЛЬНЫХ СХЕМ ДЛЯ ВЫСОКОПРОИЗВОДИТЕЛЬНЫХ ВЫЧИСЛЕНИЙ

*С. Г. Елизаров<sup>1</sup>, В. В. Корнеев<sup>2</sup>*

<sup>1</sup>Дизайн-центр цифровой микроэлектроники, МГУ им. М. В. Ломоносова, Москва  
<sup>2</sup>Научно-исследовательский институт «Квант», Москва

В докладе рассматривается архитектура С-СБИС, ориентированная на технологические нормы 28 нм и менее для решения распараллеливаемых вычислительно сложных задач символьной обработки и дискретной математики. Для «нативного» распараллеливания программ применен подход «легких потоков», когда фрагменты кода единой программы передаются с очень низкими накладными расходами на множество исполнительных ядер, а результаты вычислений сохраняются в общей памяти. Описаны основные аппаратные блоки разработанной С-СБИС, в том числе вычислительные ядра, накристалльная коммуникационная сеть, специализированный на проблемную область процессор с SIMD архитектурой.

*Ключевые слова:* СБИС, архитектура

### **Особенности архитектуры СБИС, обусловленные технологией изготовления**

Архитектура специализированных больших интегральных схем С-СБИС должна разрабатываться с ориентацией на эффективную реализацию алгоритмов области специализации. При этом критериями эффективности реализации выступают производительность, потребляемая мощность, производительность на единицу мощности и т. д. в зависимости от требований проблемной области.

В связи с тем, что в настоящее время коммерчески доступны технологические процессы 28/16/7 нм... в предлагаемой архитектуре С-СБИС будем учитывать технологический сдвиг [1]. Современные технологии микроэлектроники вносят свои ограничения в создаваемые архитектуры С-СБИС, проявившиеся при достижении технологической нормы 90 нм, а также дальнейшие ограничения, начиная с норм 28 нм. Эти ограничения обусловлены: 1) усиливающимся по мере роста степени интеграции энергопотреблением интегральных схем, существенно определяемым суммарной длиной проводников, доставляющих команды и данные к функциональным устройствам; 2) ограничениями площади локальных кластеров на СБИС, состоящих из функциональных устройств и источников команд и операндов для них, использующих синхронное тактовое дерево, а также переход к архитектурам класса GALS (Globally Asynchronous, Locally Synchronous), в которых реализована глобальная асинхронность при передаче между синхронными вычислительными устройствами разных кластеров; 3) проблемой «темного кремния», заключающейся в схемотехнической возможности формирования схемы, которая при ее реализации на кристалле СБИС приведет к превышению предельно допустимой рассеиваемой мощности, что делает необходимым целый комплекс мер по обеспечению снижения потребляемой мощности, как на технологическом, так и на архитектурном уровне.

Поэтому специализация должна реализовываться в рамках ограничений, задаваемых технологией изготовления кристалла. Пусть требуется создать специализированный микропроцессорный кристалл, предельно достижимой производительности при вышеприведенных ограничениях. Из-за ограничений на время распространения тактового сигнала, синхронная схема специализированного процессорного элемента (ПЭ) может занимать площадь и содержать количество транзисторов, не

более наперед определенных для используемых технологических норм и тактовой частоты. Поэтому архитектура будет высоко параллельной класса GALS.

Более того, ограничения по потребляемому току в статике и динамике, а также выделяемому и отводимому теплу, приводят к тому, что в каждом такте может изменяться состояние только части из общего количества транзисторов. Поэтому рациональным решением служит введение в ПЭ блоков памяти, создающих эффект большого процента не переключающихся в каждом такте транзисторов. Оставшиеся после реализации блоков памяти транзисторы используются в специализированных функциональных устройствах. На большом кристалле может разместиться много (тысячи) указанных ПЭ, функционирование которых требует загрузки и выгрузки их блоков памяти для смены программ и данных ПЭ, а также инициализации выполнения в них программ. Такие архитектурные решения использованы в кристаллах разной специализации, например: PEZY-SC2 [2], ориентированном на численные методы решения дифференциальных уравнений, и Graphcore Colossus IPU [3], созданном для решения задач искусственного интеллекта.

Специализация архитектуры кристалла определяет, служит ли он только полем однородных ПЭ, загружаемых и управляемых из внешнего по отношению к кристаллу хост-компьютера, как Graphcore Colossus, либо реализуется иерархическая гетерогенная кластерная архитектура, в которой на каждом уровне иерархии вычислительные устройства имеют локальные блоки памяти, как в PEZY-SC2. В этом архитектурном варианте связь между хост-компьютером и специализированным кристаллом частично перекладывается на управляющие процессоры кластеров, что снижает требования к производительности хост-компьютера и пропускной способности канала связи между ним и специализированным кристаллом или множества объединенных специализированных кристаллов.

Программирование вычислительных систем, состоящих из хост-компьютеров и поля однородных вычислительных модулей (ВМ) или, более того, иерархии гетерогенных кластеров представляет сложную проблему, требующую от программиста знания технических деталей аппаратной реализации.

В отличие от упомянутых выше суть представляемой в докладе архитектуры С-СБИС при использовании иерархии гетерогенных кластеров состоит в реализации всех функций по организации и управлению хост-компьютера в собственно С-СБИС. То есть хост-компьютер может передать программу и исходные данные в процессор управления С-СБИС и получить из него результат выполнения. Программа для процессора управления создается с использованием привычных программистам С, С++, OpenCL, подходов и библиотек. Используется «нативный» подход к распараллеливаю вычислений, при котором от программиста требуется только указать, это может выполняться параллельно.

## Архитектура С-СБИС

### *Логическая структура С-СБИС*

Основные концепции представляемой архитектуры и программирования изложены в [4]. Для получения адекватных оценок эффективности предлагаемой архитектуры проведено проектирование в САПР С-СБИС для размера кристалла  $6 \times 6$  мм<sup>2</sup>. Выбор такого размера обусловлен сравнительной простотой доступа на микроэлектронные производства с технологией 28 нм.

Архитектура разработанного кристалла приведена на рис. 1. Логически С-СБИС состоит из универсальной компоненты и компоненты специализации. Универсальная компонента служит для организации вычислений, доставки данных и выгрузки результатов из специализированной компоненты. Соответственно, специализированная компонента обеспечивает высоко эффективные вычисления. Эта архитектура MALT (Manycore Architecture with Lightweight Threads) [5] служит фундаментом для построения семейств специализированных СБИС. Представленная ниже С-СБИС принадлежит семейству MALT-С, специализация которого решение идеально распараллеливаемых вычислительно сложных задач символьной обработки и дискретной математики.

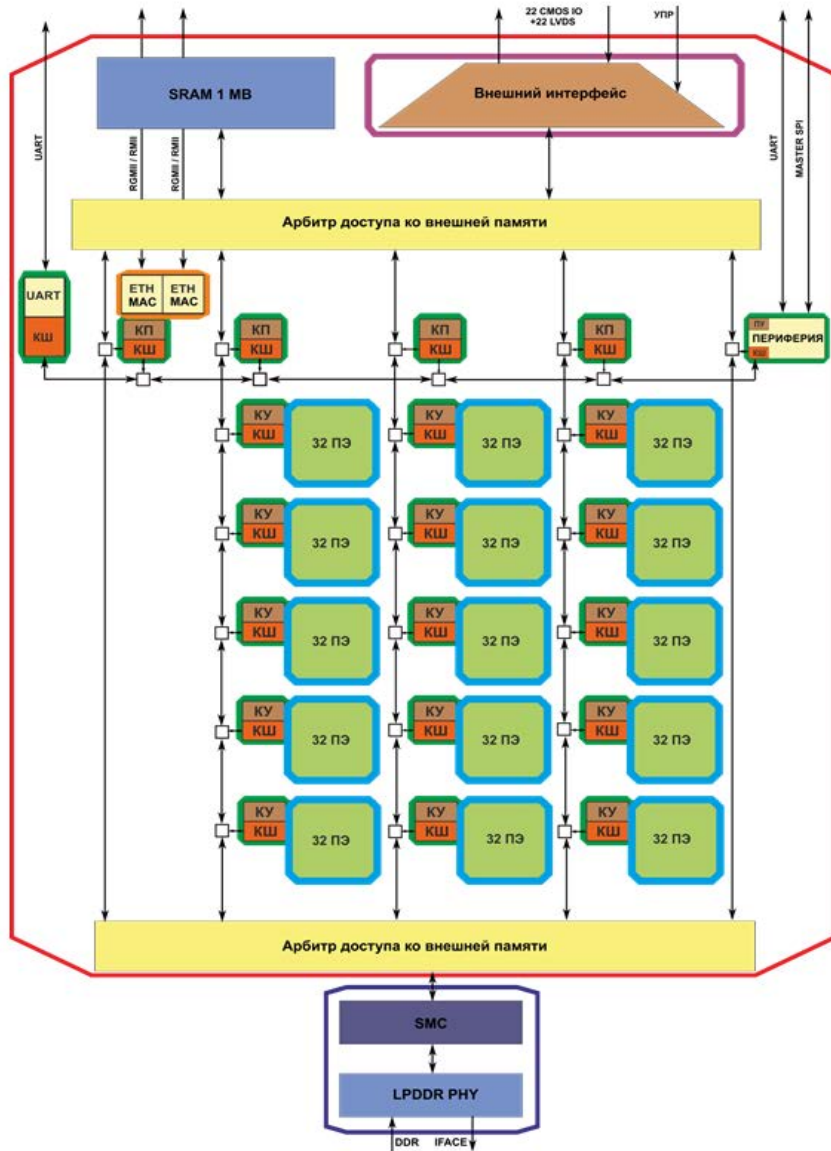


Рис. 1. Архитектура П-СБИС

### *Универсальная компонента С-СБИС*

Универсальная компонента кристалла С-СБИС включает 3 типа процессоров: процессор управления ПУ, коммуникационные процессоры КП и вычислительные процессоры КУ.

Процессор управления включает:

- 32-битное процессорное RISC-ядро общего назначения MB-Lite, по набору инструкций совместимое с архитектурой MicroBlaze;
- ПЗУ инструкций (на шине инструкций MB-Lite);
- декодер адреса (шины данных MB-Lite);
- кеш инструкций;
- блок локальной памяти;
- контроллер КШ пакетной шины;
- контроллер периферийных интерфейсов.

Исполняемый код программы может частично храниться в ПЗУ инструкций (ROM-памяти), либо подгружаться через используемую пакетную шину с промежуточным хранением инструкций

в кеш-памяти (8 Кбайт). Данные для исполняемой программы могут быть загружены по пакетной шине в блок локальной памяти. Благодаря низко латентному доступу к локальной памяти данных достигается эффективная работа алгоритмов, требующих небольших объемов данных. Кеш данных отсутствует, по причине значительных накладных расходов по обеспечению кеш-когерентности.

На шине данных процессорного ядра присутствует декодер адреса, обеспечивающий подключение к шине данных как блоков памяти, так и других устройств с регистровым интерфейсом, отображаемым на эту шину.

Процессор управления функционирует под операционной системой Minix3. Вычислительный процессор КУ имеет тот же состав оборудования как у управляющего процессора, за исключением того, что в нем отсутствует контроллер периферийных интерфейсов, но добавлен интерфейс с кластером ПЭ. К каждому вычислительному процессору подключен один кластер (32 ПЭ).

Коммуникационные процессоры КП не имеют вышеупомянутых контроллера периферийных интерфейсов, а также интерфейса с кластером ПЭ и по набору инструкций совместимы с архитектурой MicroBlaze.

Пакетная шина с малыми задержками, используемая для объединения абонентов внутри С-СБИС построена на базе коммутаторов КОМ канального уровня  $2 \times 2$ . Один вход и выход коммутатора служат для подключения абонента посредством контроллера КШ пакетной шины, а оставшиеся вход и выход используются для соединения с предшествующим и последующим коммутаторами с образованием конвейера. На краю конвейера находится оконечный блок шины или терминатор шины, в случае отсутствия абонента.

Коммутаторы с использованием каналов (линков) объединяются в однонаправленный конвейер. Абоненты шины посредством контроллеров шины КШ подключаются к соответствующим портам коммутаторов. Абонент может быть подключен к двум разнонаправленным конвейерам, как это показано на рис.1. Такая топология выбрана исходя из требований к коммуникационной подсистеме – возможность одновременных передач данных между несколькими абонентами шины, направленность потоков данных, низкая задержка и малое количество затрачиваемых ресурсов.

Ширина портов коммутатора и линка – 65 бит: 64 бита – данные и бит, который служит для указания, текущее слово шины представляет собой заголовок.

Пакетная шина реализует принцип червячной передачи (wormhole), то есть, в каждый момент времени коммутатор хранит одно слово передаваемого пакета.

Пакетная шина использует статическую маршрутизацию. Заголовок пакета содержит номер абонента, которому предназначен запрос, в каждом узле коммутационной сети номер текущего абонента сравнивается со значением в заголовке и, исходя из этого, осуществляется маршрутизация сообщений.

Максимальная длина пакета на канальном уровне – 256 слов или 2 КБ. Одновременно шина может осуществлять передачу нескольких сообщений.

Количество пакетов, одновременно передаваемой по шине, определяется количеством коммутаторов. В системе из 16 абонентов, соединенных однонаправленным конвейером, может находиться максимум 8 пакетов по 2 слова в каждом или 1 пакет длиной 16 и больше слов.

При передаче данных используется коммутация без буферизации. То есть, длинный пакет разделен на части и, в зависимости от заголовка, часть пакета, на следующем такте, попадает в соответствующий выходной порт. При наличии у обоих входных портов пришедших данных, предпочтение отдается первому порту коммутатора. Для того, чтобы не прерывать работу абонента шины, когда приходят данные, реализована входная буферизация у второго порта.

Передача сообщений по шине на протокольном уровне является тегированной, что позволяет более эффективно использовать приходящие данные со стороны абонента.

На протокольном уровне реализовано несколько типов сообщений, таких как сообщения различной длины, broadcast-сообщения, запись и чтение в удаленную память массивов различной длины (DMA).

На кристалле С-СБИС реализовано 6 пар двунаправленных пакетных шин:

– одна для связи процессора управления с коммуникационными процессорами и периферийными устройствами;

- три для связи коммуникационных процессоров с соответствующими вычислительными процессорами;
- одна для доступа процессора управления к внутри кристалльному блоку памяти (1 МБ) и блоку внешней памяти DDR;
- одна для обмена между сетевыми контроллерами и блоками внутри кристалльной и внешней памяти.

Разделяемая ПУ, КУ, КП память расположена непосредственно на кристалле, а также может подключена через интерфейс DDR или LVDS. Возможны комбинации этих вариантов. Пропускная способность накристалльной памяти значительно выше пропускной способности DDR или LVDS. Но размер накристалльной памяти ограничен и обычно не превышает 1 Мегабайта. В случае использования накристалльной памяти совместно с внешней памятью адресное пространство начинается в накристалльной памяти и продолжается во внешней. Доступ к блокам памяти выполняется через SMC (smart memory controller) «умный» контроллер памяти, выполняющий атомарные операции и синхронизацию доступов к памяти на базе FE-битов [4].

### Специализированная компонента С-СБИС

Для высоко эффективной параллельной обработки цифровой и символьной информации вычислительные процессоры КУ используют специализированную систему с SIMD архитектурой, показанной на рис. 2.

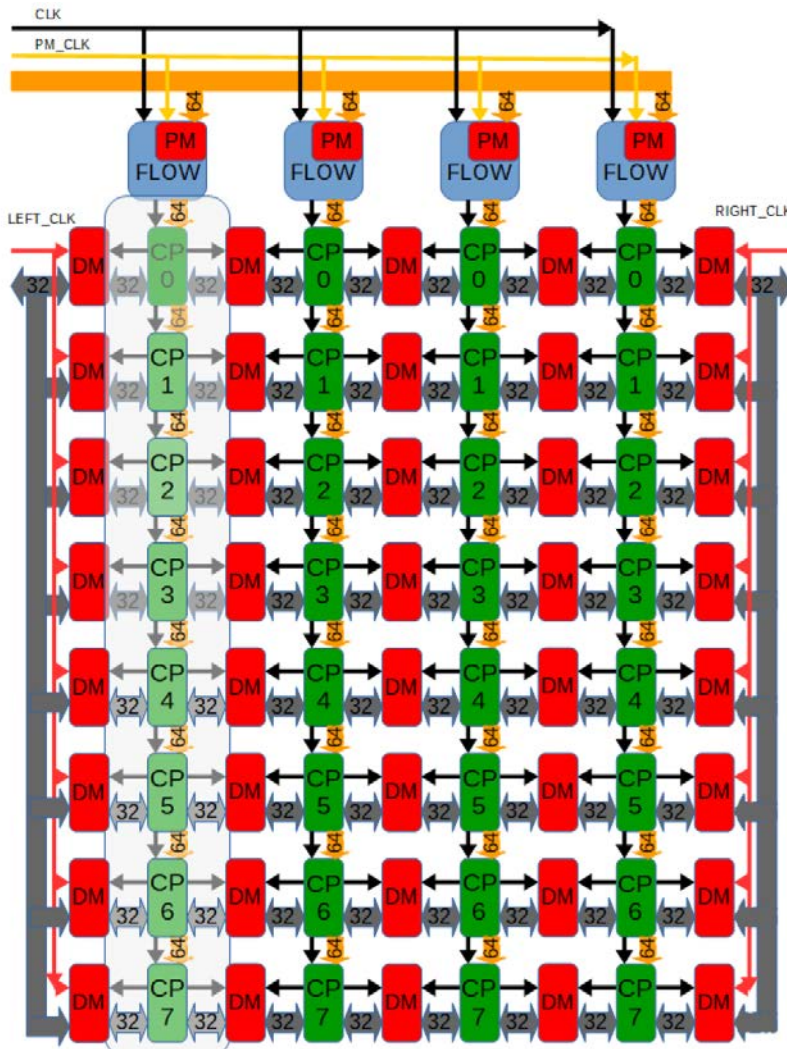


Рис. 2. Архитектура специализированной SIMD системы

Специализированная система состоит из четырех независимых устройств выдачи потока команд, к каждому из которых подключен конвейер из 8-ми процессорных элементов ПЭ, образующих в совокупности 8-ми процессорную систему с SIMD архитектурой. ПЭ выполняет операции над 32 разрядными словами.

Каждый такт команда поступает на выполнение в очередной ПЭ конвейера. Таким образом каждый ПЭ выполняет со сдвигом на такт одну и ту же команду. Ветвление возможно пропуском выполнения поступившей команды в зависимости от результата выполнения предыдущей команды. Каждый ПЭ имеет доступ к двум блокам локальной памяти SDRAM слева и справа от него, объемом каждый 1280 32-разрядных слов.

Вычислительный процессор КУ загружает программу в устройства выдачи потока команд, разные или одну и ту же, а также записывает или читает данные из крайних столбцов блоков памяти. Это обуславливает программное обращение к остальным блокам памяти через регистры соответствующих ПЭ. При этом у ПЭ нет прямого доступа ни к локальной памяти КУ, ни к общей разделяемой памяти.

Все устройства выдачи потока команд и ПЭ функционируют синхронно по-тактно. Проблемы возможных конфликтов по доступу двух ПЭ к разделяемому блоку памяти решаются на уровне программирования.

Структура ПЭ приведена на рис. 3.

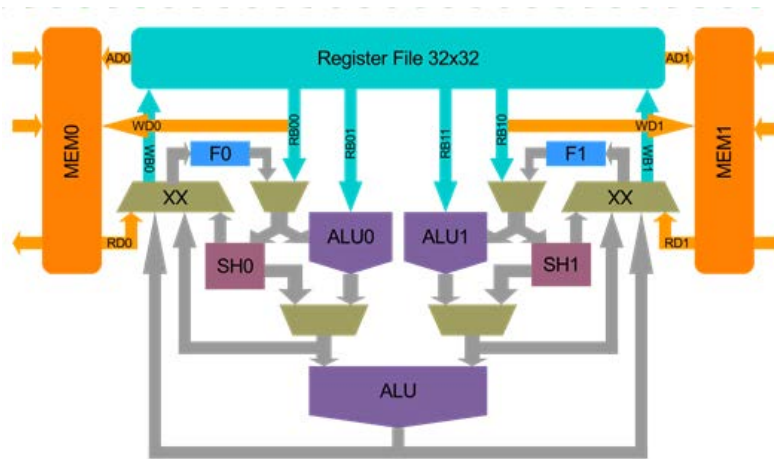


Рис. 3. Структура ПЭ

Процессорные элементы построены по архитектуре VLIW и могут одновременно выполнять 5 действий (3 АЛУ и 2 сдвигателя) с 4 операндами. Ветвление возможно только посредством временной остановки определенных процессорных элементов в зависимости от результата выполнения предыдущей команды. Таким образом, максимальная производительность ПЭ при тактовой частоте 1 ГГц достигает  $5 \times 32 \times 10^9 = 1,6 \times 10^{11}$  операций в секунду.

Максимальная производительность С-СБИС определяется суммой производительностей ПЭ и, например, для приведенного на рис.1 варианта кристалла площадью  $6 \times 6 \text{ мм}^2$  с 15 ПЭ, составляет  $2,4 \times 10^{12}$  операций в секунду. Энергопотребление при тактовой частоте 1 ГГц по результатам моделирования в САПР составляет не более 10 Вт.

### Возможности применения

С использованием СБИС семейства MALT-C возможно построение вычислительных систем для различных проблемных областей. Ориентация на область использования закладывается, например, на уровне конструктивного элемента платы, как показано на рис. 4.

Пример построения стойки вычислительной системы приведен на рис. 5.

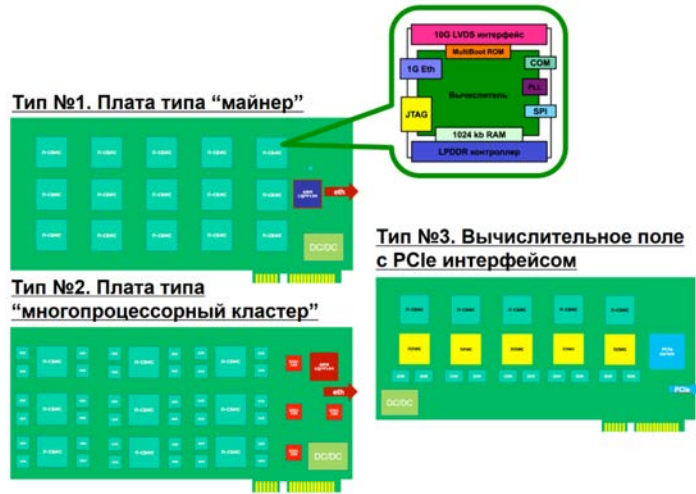


Рис. 4. Платы для возможных проблемных областей применения

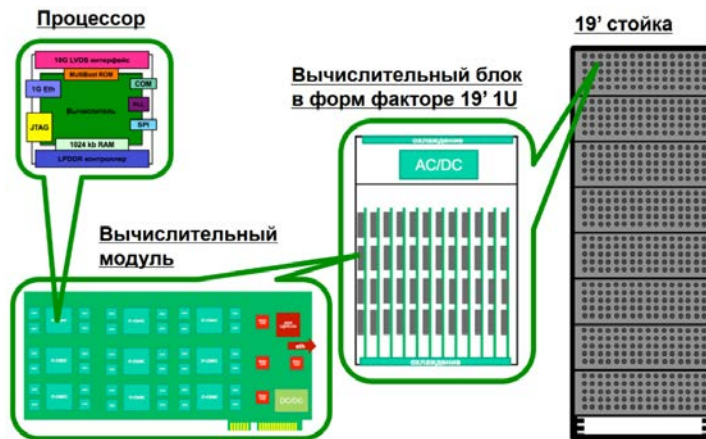


Рис. 5. Пример конструктивного исполнения стойки проблемно-ориентированной вычислительной системы

### Средства программирования

Программист, в зависимости от своих предпочтений и требований к оптимизации прикладного кода, может выбрать один из следующих подходов:

– C++ для MALT. Для программирования скалярных ядер поддерживается стандарт C++17. Работа с легкими потоками `thread` осуществляется через конструкции C++ `std::thread`, `std::mutex` и другие. MALT выступает в роли классического многоядерного процессора, поэтому портировать существующее ПО и библиотеки легко.

– OpenCL для MALT. Реализация стандарта OpenCL для работы со скалярными и векторными ядрами MALT. Реализация OpenCL поставляется вместе с библиотекой проблемно-ориентированных алгоритмов, оптимизированных под MALT. Теперь перейти на MALT для пользователей ускорителей AMD, NVIDIA, ARM стало проще.

– C для MALT + низкоуровневая библиотека. Реализует возможности архитектуры, позволяет в ручном режиме оптимизировать и распараллеливать программы целевых классов.

Исполнение программы инициируется процессором управления ПУ, которым служит «обычный» многопоточный процессор, функционирующий под управлением POSIX-совместимой ОС Minix3. Для «нативного» распараллеливания программы на множество легких потоков, фрагмент кода единой программы ПУ отправляется им по пакетной шине с очень низкими накладными рас-

ходами на выбранный коммуникационный процессор КП. В свою очередь, КП определяет один или подмножество исполнительных вычислительных процессоров КУ, из числа подключенных к одной с ним пакетной шине. Вычислительные процессоры при этом получают указатель на фрагмент кода, передаваемого им для исполнения. Результаты сохраняются в общей памяти. Такой подход реализован, например, в библиотеке «легких тредов» Qthreads [6], созданной в Sandia National Laboratories. Основная идея – обеспечить механизм порождения параллельно исполняемых потоков на уровне программы процессора, а не на уровне операционной системы. В Qthread порождение потока описывается вызовом вида:

```
void start_thread(int (*)(int), int, ...);
```

Вызов может быть произведен из любого вычислительного процессора. Такой вызов совместно с атомарными, не блокируемыми операциями инкремента и некоторыми другими стандартными операциями создает возможность порождать (spawn) и объединять (join) группы параллельно выполняемых потоков, создавая последовательно-параллельную программу.

### Литература

1. Корнеев В. В., Тарасов И. Е. Особенности архитектуры массово-параллельных проблемно-ориентированных СБИС // Программная инженерия. 2019. Т. 10, № 4. С. 160–166.
2. Torii S., Ishikawa H. ZettaScaler: Liquid immersion cooling Manycore based Supercomputer // ISC 2017. Frankfurt am Main, Germany, June 18–22, 2017.
3. Jia Zh., Tillman B., Maggioni M., Scarpazza D. Dissecting the Graphcore IPU Architecture via Microbenchmarking. Technical Report. December 7, 2019. arXiv:1912.03413v1 [cs.DC] 7 Dec 2019.
4. Елизаров С. Г., Лукьянченко Г. А., Корнеев В. В. Технология параллельного программирования эксафлопсных компьютеров // Программная инженерия. 2015. № 7. С. 3–10.
5. Архитектура MALT [Электронный ресурс]: сайт Malt System. – Режим доступа: <https://maltssystem.ru>.
6. Wheeler K., Murphy R., Thain D. Qthreads: An API for programming with millions of lightweight threads // In Proceedings of the 22nd IEEE International Parallel & Distributed Processing Symposium, April 2008, pp. 1–8.

## THE ARCHITECTURE OF THE APPLICATION SPECIFIC INTEGRATED CIRCUITS FOR HIGH-PERFORMANCE COMPUTING

*S. G. Elizarov<sup>1</sup>, V. V. Korneev<sup>2</sup>*

<sup>1</sup> Design Center for Digital Microelectronics, MSU, Moscow

<sup>2</sup> Research Center “Kvant”, Moscow

The report discusses the S-VLSI circuit architecture focused on technological standards of 28 nm or less for solving computationally hard parallelizable problems of symbolic processing and discrete mathematics. For native parallel programming we use the ‘lightweight threads’ approach dividing the program into multiple threads performing relatively short code fragments from a single program with very low overheads. The results are saved in the shared memory. Main units of the S-VLSI circuit including universal cores computing kernels, on-chip communication network, specialized processor with SIMD-architecture are described.

*Key words:* VLSI circuit, architecture.