

Цифровой сигнальный процессор с нетрадиционной рекуррентной потоковой архитектурой

Ю.А. Степченков, В.Н. Волчек, В.С. Петрухин, А.А. Прокофьев, Р.А. Зеленев

Учреждение Российской академии наук Институт проблем информатики РАН (ИПИ РАН),
{YStepchenkov, VVolchek, VPetrukhin, AProkofyev, RZelenov}@ipiran.ru

Аннотация — В статье представлены результаты разработки цифрового сигнального процессора с нетрадиционной рекуррентной потоковой архитектурой. Помимо специфики самой архитектуры, рассматриваются особенности организации вычислительного устройства, ориентированного на эффективное исполнение алгоритмов цифровой обработки сигналов. Отмечено, что в качестве аппаратного базиса для разработки используется ПЛИС фирмы Altera семейства Stratix III, а в качестве языка описания аппаратуры – язык VHDL. Показаны особенности VHDL-описания основных функциональных блоков для эффективного синтеза в ПЛИС. Приведены результаты синтеза.

Ключевые слова — Цифровой сигнальный процессор; потоковая архитектура; рекуррентность; многоядерность; параллелизм; VHDL-синтез; ПЛИС.

I. ВВЕДЕНИЕ

В компьютерной индустрии основными методами повышения производительности компьютеров и вычислительных систем на их основе являются развитие их архитектуры и увеличение тактовой частоты элементной базы. В последние годы, мировые производители процессоров, имея возможность использовать более совершенные технологии, зачастую действовали однобоко, делая ставку, в первую очередь, на повышение тактовой частоты, не уделяя должного внимания совершенствованию архитектуры [1].

Если повышение тактовой частоты – быстро исчерпываемый ресурс в силу существующих технологических ограничений, то повышение эффективности архитектуры (организации вычислений) – практически неограниченный ресурс. Одним из наиболее эффективных архитектурных способов повышения производительности вычислительных систем является обеспечение поддержки параллельных вычислений.

Идеи параллелизма реализуются на всех уровнях – от одноядерных и многоядерных процессоров до систем массового параллелизма (СМП) с сотнями тысяч процессоров. При этом полезное использование отдельного процессора в СМП может характеризоваться единицами процентов, что малоэффективно для встраиваемых цифровых сигнальных процессоров (ЦСП) реального времени.

Архитектура используемых вычислительных средств базируется на классической архитектуре фон-Неймана (ФН), которой свойственен ряд недостатков: излишняя сложность, неэффективность распараллеливания, низкий коэффициент полезного использования аппаратных ресурсов и пр.

Эти и другие факторы стимулируют поиски новых путей развития компьютерной индустрии; необходимы инновационные (нетрадиционные) подходы к построению архитектур вычислительных систем [2].

Наиболее перспективным подходом к снижению издержек при организации параллельных вычислений является переход к событийно-ориентированным вычислительным архитектурам. На логическом уровне это ориентация на потоковый (data-flow) принцип организации вычислений [3], а на схемотехническом – на самосинхронную схемотехнику [4].

Созданием подобной нетрадиционной архитектуры нового поколения занимается Институт проблем информатики РАН. Эта архитектура – развитие потокового подхода, но построенного на другом базисе:

- оба потока – поток данных и поток команд – интегрированы в один поток самодостаточных данных; традиционная двухпоточность сводится к однопоточности;

- поток самодостаточных данных хранит рекуррентно свёрнутый (сжатый) алгоритм решения конкретной задачи;

- в основе парадигмы вычислений лежит графодинамическое представление алгоритмов, рекуррентно свёрнутых до момента инициации исполнения и саморазворачивающихся в ходе процесса выполнения задач [5].

В качестве практической апробации разрабатываемой архитектуры предлагается гибридный двухуровневый вариант ее реализации в виде управляющего и операционного уровней. На управляющий уровень возложены минимально необходимые функции управления операционным уровнем и связь с внешним окружением, и он реализован на традиционных ФН-принципах. Многоядерный операционный уровень реализован в виде рекуррентного обработчика сигналов (РОС), предназначенного для эффективного исполнения параллельных алгоритмов в речевой об-

ласти. Разработка подобного ЦСП представляется перспективным и своевременным проектом, соответствующим тенденциям развития области цифровой обработки сигналов (ЦОС) [5]. Многоядерный РОС в текущем исполнении имеет в своем составе четыре процессорных ядра (ПЯ), способных работать параллельно. Подобный уровень параллелизма можно назвать параллелизмом на уровне ПЯ.

Поскольку работа имеет исследовательский характер, чрезвычайно важна возможность гибкого внесения изменений в структуру РОС при минимальных финансовых и временных затратах. С другой стороны, предполагаемый базис реализации должен соответствовать программно-аппаратным требованиям управляющего уровня, позволив сосредоточиться непосредственно на предмете разработки – рекуррентном операционном уровне. В соответствии с этими ключевыми критериями, в качестве элементной базы реализации РОС выбраны программируемые логические интегральные схемы (ПЛИС) и, в частности, семейство Stratix III фирмы Altera [5] (система на кристалле), т.к. они содержат многообразную синтезируемую и аппаратную периферию, в том числе позволяющую реализовать управляющий уровень РОС.

В структуру гибридного РОС входят (см. рис. 1): управляющий процессор (Nios II), распределитель, четыре однотипных ПЯ и интерфейс межпроцессорного обмена. Двухпортовая буферная память – информационный интерфейс двух уровней РОС [5].

В РОС используется капсульный стиль программирования, основные понятия которого соответствуют принятым в телекоммуникационных приложениях: инкапсуляция, собственно капсула и декапсуляция [6]. Суть капсульного программирования заключается в процессе формирования капсулы-шаблона (инкапсуляции), которая заносится в буферную память. По мере прихода в РОС обрабатываемых данных, управляющий уровень производит процесс заполнения шаблона, после завершения которого капсула инициируется и поступает на обработку. Процессу обработки капсулы на операционном уровне предшествует процедура декапсуляции [7].

К настоящему времени, на языке описания аппаратуры VHDL реализованы и синтезированы в ПЛИС Stratix III все блоки операционного уровня: распределитель, память совпадений (ПС), вычислительное устройство (ВУ).

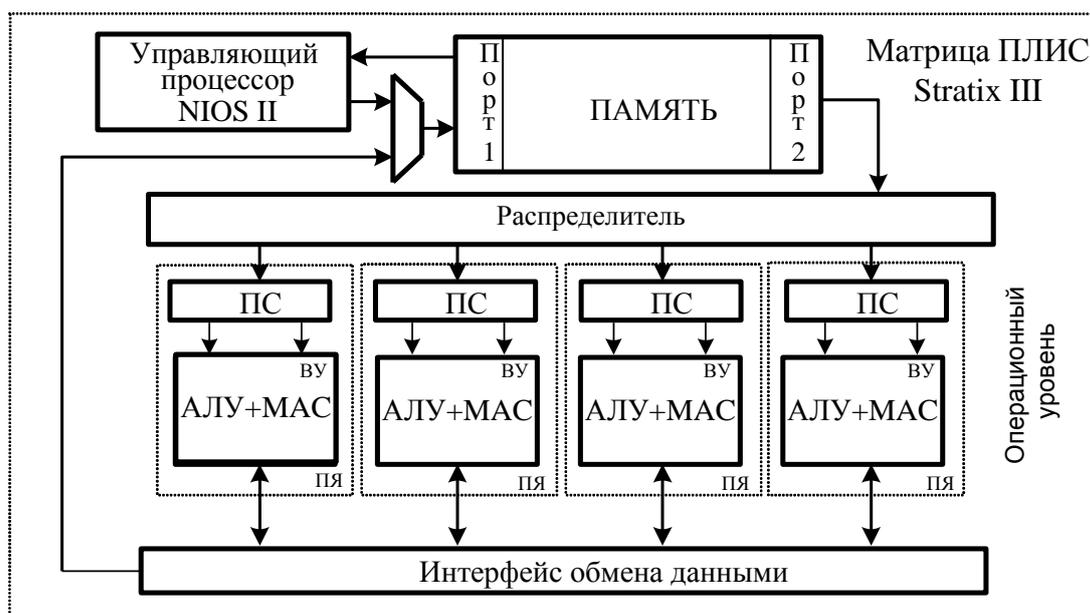


Рис. 1. Структура РОС на основе ПЛИС

II. УПРАВЛЯЮЩИЙ УРОВЕНЬ РОС

Специфика программирования многоядерного потокового РОС скрыта от пользователя традиционным ФН-процессором, который также решает задачи управляющего уровня ЦСП. Фирма Altera и ее партнеры предлагают на выбор несколько синтезируемых процессоров традиционной архитектуры [8]. По совокупности характеристик "производи-

тельность, универсальность, занимаемая площадь кристалла", а также при наличии подробной документации и соответствующего программного обеспечения для синтеза в ПЛИС фирмы Altera, в качестве управляющего процессора выбран синтезируемый процессор Nios II. Он представляет собой процессор общего назначения с настраиваемой конфигурацией, что обеспечивает высокую гибкость сис-

темы. Ядро Nios II – конвейерный RISC-процессор (Reduced Instruction Set Computer) с одноклковым выполнением команд. Это позволяет легко синхронизировать управляющий и операционный уровни РОС.

При синтезе в ПЛИС семейства Stratix III ядро Nios II обеспечивает большую, по сравнению с конкурентами, производительность за счет реализации деления и сдвига на аппаратных DSP-блоках [9]. Это позволяет возложить на управляющий процессор функции:

- масштабирования входных данных для последующей обработки на операционном уровне;
- вычислителя для обработки возможных последовательных частей алгоритма, реализуемого РОС;
- интерфейса между стандартным и специализированным программным обеспечением РОС;
- устройства управления для обработки исключительных ситуаций;
- устройства-компоновщика капсул для их исполнения на операционном уровне РОС.

III. ОПЕРАЦИОННЫЙ УРОВЕНЬ РОС

Разрабатываемая рекуррентная потоковая архитектура обеспечивает минимальное (предельное) количество фаз обработки данных – три, по сравнению с пятью в традиционной фон-неймановской и потоковой парадигмах вычислений [10]. Данное обстоятельство позволяет реализовать на операционном уровне РОС трёхступенчатый вычислительный конвейер со следующими фазами исполнения:

- распределение обрабатываемых и результирующих данных (между ПЯ);
- сравнение полей-тегов самодостаточных операндов, формирование пары операндов для обработки;
- выполнение команды.

Первой ступенью операционного уровня РОС является распределитель, отвечающий за распределение входных данных по ПЯ: прием данных с управляющего уровня и контроль потоков данных внутри операционного уровня. За один такт синхронизации РОС, распределитель должен обеспечить все ПЯ данными (до четырех пар 52-разрядных операндов). За счет введения специальных режимов работы распределителя удалось уменьшить размер капсулы, сэкономив этим буферную память и минимизировав трафик между управляющим и операционным уровнями. Синтез показал, что распределитель выполняет подготовку всех необходимых данных за 15,7 нс (63,5 МГц).

Следующей ступенью конвейера РОС является память совпадения (ПС). При совпадении полей-тегов операндов (хранящегося и поступившего в ПС) в каждом ПЯ выдается пара операндов на обработку в ВУ. По сравнению с реализацией ПС, описанной в

[11], ее функциональность расширена возможностью выбора, на какой из двух входов ВУ подать операнды, в зависимости от режима и типа операнда. ПС фиксирует также исключительные ситуации и управляет режимами работы ВУ. Время срабатывания этой ступени конвейера 15,3 нс (65,35 МГц).

Третью и последнюю ступень вычислительного конвейера выполняет вычислительное устройство РОС. В силу специфики этого функционального блока и его ориентации на ЦОС рассмотрим его более подробно.

IV. ВЫЧИСЛИТЕЛЬНОЕ УСТРОЙСТВО РОС

Вычислительное устройство – один из самых сложных и ресурсозатратных функциональных узлов РОС. Как и ПС, ВУ присутствует в каждом процессорном ядре; от его реализации зависит эффективность работы всего РОС. По своему назначению и составу аппаратных узлов ВУ достаточно стандартно. Подобные по структуре функциональные узлы присутствуют во многих современных ЦСП. Их предназначение – эффективное выполнение алгоритмов ЦОС за счет поддержки ряда требований [12].

Наиболее важная операция в ЦОС – суммирование результатов умножения. Эта операция одинаково важна для цифровых фильтров, быстрого преобразования Фурье (БПФ) и множества других алгоритмов ЦОС. ВУ оптимизировано для выполнения повторяющихся математических операций, таких как умножение с накоплением. Для этих целей в ВУ, как и в стандартных ЦСП, присутствует аппаратный блок MAC (Multiplication with Accumulation), позволяющий за один вычислительный такт перемножить два 16-разрядных операнда и накапливать результат умножения в одном из двух внутренних 40-разрядных регистров. Разрядности входных обрабатываемых чисел в 16 битов вполне достаточно для целого ряда применений в области речевой обработки.

Помимо операции умножения с накоплением, блок MAC, входящий в состав РОС, способен выполнять команды арифметического, логического сдвигов и округления результатов. ВУ содержит также 16-разрядное арифметико-логическое устройство (АЛУ), выполняющее основные арифметические и все логические команды.

Все перечисленные аппаратные узлы являются, по сути, стандартом для большинства ЦСП. Учитывая этот факт, разработчики ПЛИС и, в частности, фирма Altera встраивают в свои микросхемы аппаратные блоки, изначально специализированные на задачи обработки сигналов. Кроме аппаратных, существуют также синтезируемые блоки DSP, в том числе и распространяемые бесплатно.

В схемах семейства Stratix III присутствуют аппа-

ратные DSP-блоки, эффективно реализующие основные операции ЦОС: умножение, умножение со сложением, умножение с накоплением, сдвиг [13].

Для того чтобы эту функциональность задействовать, как правило, необходимо использовать синтезируемые DSP-блоки – так называемые IP-ядра. Они представляют собой настраиваемые библиотечные функции, которые можно успешно применять для реализации стандартных архитектур вычислителей в устройствах ЦОС. Однако, из-за того, что при реализации РОС применяются эффективные, но нестандартные аппаратные решения и система команд (речь о них пойдет ниже), использовать существующие на данный момент синтезируемые DSP-блоки и, соответственно, весь потенциал ПЛИС не представляется возможным.

Во многих ЦСП (например, dsPIC30F компании Microchip) в рамках одного процессорного ядра в каждом вычислительном цикле используется только один из блоков; при использовании MAC АЛУ простаивает и наоборот. ВУ одного ПЯ содержит два 40-разрядных регистра-аккумулятора (как и в ряде стандартных ЦСП) и один 16-разрядный регистр – выходной аккумулятор (в силу специфики РОС). В стандартных ЦСП, как правило, в каждом вычислительном цикле в ПЯ используется только один аккумулятор из двух, а при стандартном подходе в РОС – один аккумулятор из трех.

Каждый из перечисленных узлов и регистров ВУ представляет собой независимый аппаратный ресурс, цена простоя которого достаточно высока. Это обстоятельство натолкнуло на мысль о внедрении поддержки их одновременной работы – суперскалярности – в ВУ.

Суперскалярность – это архитектура ВУ, использующая несколько декодеров команд, которые могут нагружать работой множество исполнительных блоков. Планирование динамического выполнения потока команд осуществляется самим вычислительным ядром [14]. Таким образом, наделение ВУ свойством суперскалярности позволяет достичь параллельности на уровне команд в дополнение к обозначенной выше параллельности на уровне ПЯ.

Анализ алгоритмов обработки голосовых сигналов выявил необходимость использования команд сложения/вычитания и умножения в суперскалярном режиме. Потенциально существует возможность выполнять и другие комбинации команд, но практической необходимости этого в рассматриваемой области сейчас не возникает. Рассмотрим суперскалярную организацию ВУ более подробно.

Как уже было сказано, в РОС отсутствует поток команд и данных. Вместо них есть поток самодостаточных данных, которые несут в себе всю необходимую информацию для выполнения (содержат и

команду, и данные). Как следствие, в ВУ всегда анализируются коды операций обоих поступающих с выхода ПС операндов. Если они совпадают, то выполняется текущая операция в обычном последовательном режиме, с полезной работой только одного аппаратного узла ВУ и одного или двух аккумуляторов. Если коды операций не совпадают – это либо ошибка, либо переход в суперскалярный режим (сложение/вычитание и умножение). При этом один операнд для команды сложения/вычитания отправляется в АЛУ, а другой – в блок MAC. Для формирования пары для каждого из поступивших операндов считывается значение из двух 40-разрядных аккумуляторов, присутствующих в блоке MAC, или из памяти коэффициентов (постоянное запоминающее устройство в каждом ВУ).

Внедрение этих решений позволило повысить эффективность выполнения ряда алгоритмов (см. таблицу 1).

Таблица 1

Эффективность реализации алгоритмов

Алгоритм	Число шагов вычисления (ЧШВ)	Реализация		
		dsPIC30F	РОС	¼ РОС ²⁾
"Бабочка"	ЧШВ "бабочки"	9	4 (4) ¹⁾	4
Rasta-фильтрация	ЧШВ одного параметра	8	2	8
	Общее ЧШВ	400	27	-
Евклидово расстояние	ЧШВ одной координаты	2	1 (4) ¹⁾	1
	Общее ЧШВ	32	13	-
LSP-параметры	ЧШВ одного параметра	3	2 (4) ¹⁾	2
	Общее ЧШВ	62	16	-

¹⁾ в скобках указано, что в РОС одновременно вычисляются 4 "бабочки", 4 координаты или 4 параметра

²⁾ ЧШВ для одноядерного варианта исполнения РОС

Однако возможности одновременного использования нескольких аппаратных узлов РОС, не ограничиваются только параллельным исполнением двух команд за один такт синхронизации. Поскольку статистические данные свидетельствуют, что БПФ – это базовый алгоритм ЦОС, в том числе и в области голосовых технологий, в состав команд РОС была введена специальная многоцикловая команда "Butt" (Butterfly), одновременно задействующая максимальное количество функциональных узлов ВУ.

Команда "Butt" реализует базовую операцию алгоритма БПФ по основанию 2 с прореживанием по времени ("Бабочка"). В современных ЦСП, например, в том же dsPIC30F от Microchip, этот алгоритм выполняется за 9 вычислительных шагов [10]. В РОС специальная команда "Butt" позволяет сократить количество циклов, в рамках одного ПЯ, до четырех, при наличии одинакового количества вычислительных ресурсов с большинством существующих ЦСП (см. таблицу 1). Это стало возможным путём оптимального распределения ресурсов при

вычисления "Бабочки". При реализации этой команды параллельно задействуются все существующие аппаратные узлы ВУ, а именно – 16-разрядное АЛУ, 16-разрядный умножитель и 40-разрядный сумматор.

Ещё одна особенность ВУ заключается во внедрении, так называемого, псевдосуперскалярного режима. Суть этого режима заключается в следующем. Обычная операция умножения с накоплением предполагает сложение в 40-разрядном сумматоре ранее накопленного результата с результатом умножения поступивших двух операндов в текущем цикле. В псевдосуперскалярном режиме в текущей операции обрабатываются не два, а три операнда; третий – содержимое 40-разрядного регистра. В рамках этой операции результат умножения не накапливается, а после однократного суммирования (вычитания) с входным операндом уходит на обработку в соседнее ПЯ. Поскольку в таком режиме блоки АЛУ и МАС не задействуются в параллель, этот режим назван псевдосуперскалярным. Он используется, например, в алгоритме Rasta-фильтрации.

Все эти решения при одинаковом, по сравнению с dsPIC30F от Microchip, количестве аппаратных блоков ВУ позволяют достичь ощутимого эффекта в выполнении ряда алгоритмов ЦОС (см. таблицу 1). В результате синтеза в ПЛИС было выявлено, что рабочая частота ВУ, а следовательно, и всей ступени вычислительного конвейера РОС, равна 88.3 МГц.

Несмотря на специфику разрабатываемой архитектуры, ВУ при незначительной доработке может быть использовано и в традиционных изделиях ЦОС. Этому также способствует реализация всего РОС на языке описания аппаратуры VHDL, что позволяет использовать вычислитель в качестве IP-ядра.

V. ОСОБЕННОСТИ РЕАЛИЗАЦИИ РОС НА VHDL

Язык VHDL – одно из общепринятых средств описания разрабатываемой аппаратуры. Его преимущества – минимальные сроки разработки, возможность мультиплатформенного применения и простота,

из которой следует экономическая эффективность.

Для максимальной эффективности синтеза, разработчик, при описании устройства, должен ориентироваться на низкий уровень – структурное описание. Это не всегда возможно, поскольку требует значительно больше времени, чем описание устройства на поведенческом уровне [11], подразумевающим большую абстракцию от конкретного базиса. В языке VHDL одно и то же (функционально) устройство может быть на поведенческом уровне описано разными способами, и это приведет к различным результатам синтеза.

При проектировании сложных устройств с нетрадиционной архитектурой этот вопрос становится наиболее актуальным. Поэтому в ходе разработки РОС, была поставлена задача получения набора рекомендаций и соглашений по стилю кодирования, которые помогли бы разработчику составлять VHDL-код на стадии поведенческого описания так, чтобы результаты синтеза были как можно более оптимальными. В качестве критериев оценки эффективности синтеза выступают аппаратные затраты и временные характеристики схемы.

Полученные рекомендации и правила кодирования на языке VHDL использовались при описании основных функциональных блоков (ФБ) РОС. Например, вместо оператора *if-else*, где возможно, использовался оператор *case*, так как конструкция *case*, в отличие от *if-else*, задействует минимально возможное количество логических элементов ПЛИС. В процессах используется асинхронный сброс всех внутренних регистров, что позволяет в ряде случаев избежать появления защёлки в схеме регистров.

Полный список полученных рекомендаций и используемых конструкций на примере основных ФБ РОС опубликован в [11] и может быть полезен для разработчиков, описывающих свои устройства на языке VHDL с использованием стандартных средств синтеза САПР Altera Quartus II. Результаты синтеза основных ФБ РОС представлены в таблице 2.

Таблица 2

Результаты синтеза функциональных блоков РОС

Параметры	Распределитель	Память совпадений	ВУ
Кол-во задействованных ALUT (из 38000)	640 (2%)	250 (<1%)	1145 (3%)
Кол-во используемых регистров (из 38000)	1347 (4%)	467 (1%)	161 (<1%)
Кол-во блоков памяти М9К (из 400)	0 (0)	3 (<1%)	0 (0)
Общее количество битов памяти (из 5455872)	0 (0)	448 (<1%)	0 (0)
Кол-во задействованных DSP-блоков (из 384)	0 (0)	0 (0)	2 (<1%)
Глобальные тактовые сигналы (из 16)	2 (13%)	2 (13%)	2 (13%)
Среднее использование связей (total/H/V)	2% / 1% / 2%	0% / 0% / 0%	1% / 1% / 1%
Максимальное использование связей (total/H/V)	16% / 16% / 16%	6% / 6% / 6%	7% / 7% / 7%

VI. ЗАКЛЮЧЕНИЕ

Рекуррентный обработчик сигналов, разрабатываемый Институтом проблем информатики РАН, представляет собой гибридный двухуровневый вариант реализации фон-неймановского/рекуррентно-однопоточного РОС на базе нетрадиционных подходов и архитектурных решений, призванных обеспечить эффективное исполнение параллельных алгоритмов в речевой области.

Предложенная архитектура совместима со всеми традиционными архитектурами: специфика программирования многоядерного потокового РОС скрыта от пользователя управляющим уровнем, реализованным на базе традиционного фон-неймановского процессора для встроенных применений Nios II.

Представлены результаты разработки и моделирования операционного уровня архитектуры РОС в виде четырехядерного рекуррентного потокового процессора в однокристалльном исполнении. Это продиктовано областью решаемых задач – задач обработки речи. Основные функциональные блоки – распределитель, память совпадений и вычислительное устройство – реализуют независимые ступени конвейера. Блоки распределителя и памяти совпадений свойственны нетрадиционным архитектурам; вычислительное устройство присутствует во всех современных ЦСП.

Для сокращения временных и финансовых затрат, в рамках исследовательской задачи в качестве аппаратной базы для реализации РОС выбраны ПЛИС фирмы Altera, семейство Stratix III. В качестве управляющего процессора используется синтезируемый процессор Nios II, который является связующим звеном между нетрадиционным многоядерным потоковым процессором и внешним окружением.

Предложенная реализация вычислительного устройства обладает рядом инноваций и предполагает максимально эффективное использование аппаратных узлов, имеющих в большинстве современных ЦСП. Моделирование ряда алгоритмов в области ЦОС показало целесообразность внедрения свойств суперскалярности и так называемой псевдосуперскалярности в вычислительном устройстве РОС.

Все перечисленные блоки реализованы на языке описания аппаратуры VHDL. В ходе разработки получен, принят и опубликован ряд соглашений по стилю кодирования, позволивших сократить аппаратные затраты и улучшить временные характеристики при синтезе в ПЛИС Altera Stratix III с использованием САПР Altera Quartus II.

Одной из особенностей разрабатываемой архитектуры является ее ориентация в перспективе на самосинхронную схемотехнику: самосинхронизация на логическом уровне (по готовности исходных данных)

хорошо сочетается с самосинхронизацией на аппаратном уровне (по готовности результатов). Поэтому в настоящее время, несмотря на то, что используется синхронный схемотехнический базис, взаимодействие между функциональными блоками и ступенями вычислительного конвейера осуществляется асинхронно.

ЛИТЕРАТУРА

- [1] Солохина Т.В., Александров Ю.Н., Глушков А.В., Беляев А.А., Петричкович Я.Я. Отечественные трехъядерные сигнальные микроконтроллеры с производительностью 1,5 GFLOPS // Электронные компоненты. – 2006. - № 6. - С. 73–78.
- [2] Черняк Л. Архитектура фон Неймана, реконфигурируемые компьютерные системы и антимашина // Открытые Системы. – 2008. - № 6. - С. 14–21.
- [3] Бурцев В.С. Параллелизм вычислительных процессов и развитие архитектуры суперЭВМ: Сборник статей / Составители Торчигин В.П., Никольская Ю.Н., Никитин Ю.В. – М.: ТОРУС ПРЕСС, 2006. – 416 с.
- [4] Мизин И.А., Филин А.В. Самосинхронизация – естественная основа параллельных компьютеров // Системы и средства информатики. – М.: ИПИ РАН, 1999. Вып. 9. - С. 225–241.
- [5] Степченков Ю.А., Петрухин В.С. Особенности гибридного варианта реализации на ПЛИС рекуррентного обработчика сигналов // Системы и средства информатики. Дополнительный выпуск. – М.: ИПИ РАН, 2008. - С. 118–129.
- [6] Н.А. Вязовик. Программирование на Java. Введение в сетевые протоколы. URL: <http://www.intuit.ru/department/pl/javapl/16/> (дата обращения: 14.05.2010).
- [7] Исследование программируемости архитектурно-алгоритмических и схемотехнических проблем проектирования рекуррентных компьютеров (заключительный отчет). Шифр "ПАРСЕК". № г.р. 01.20.0412412. – М.: ИПИ РАН, 2006. – 201 с.
- [8] Altera, Embedded Processors. URL: <http://www.altera.com/products/ip/processors/ipm-index.jsp> (дата обращения: 18.12.2009).
- [9] Встраиваемый процессор Nios II от фирмы Altera. URL: http://kazus.ru/lenta/view/0_5131_0.html (дата обращения: 09.01.2010).
- [10] Степченков Ю.А., Петрухин В.С. Перспективы развития цифровых сигнальных процессоров и возможная реализация рекуррентного обработчика сигналов // Системы и средства информатики. Специальный выпуск «Методы и средства разработки информационно-вычислительных систем и сетей». – М.: ИПИ РАН, 2004. - С. 92–140.
- [11] Петрухин В.С., Степченков Ю.А., Волчек В.Н., Прокофьев А.А., Зеленов Р.А. Особенности реализации на ПЛИС основных блоков рекуррентного обработчика сигналов // Системы и средства информатики. Дополнительный выпуск. – М.: ИПИ РАН, 2008. - С. 130–148.
- [12] Уолт Кестер. Глава 7. Аппаратура цифровых сигнальных процессоров. URL: <http://www.analog.com.ru/Public/7.pdf> (дата обращения: 15.01.2010).
- [13] Altera, Stratix III Device Handbook. URL: <http://www.altera.com/literature/lit-stx3.jsp> (дата обращения: 08.10.2009).
- [14] Таненбаум Э. Архитектура компьютера. - СПб.: Питер, 2007. - 844 с.