

На графиках (рис. 3, б) показано распределение плотности p ненулевых элементов в спектральных составляющих (в элементах декомпозиции) в зависимости от протяженности всплеска вдоль оси x_1 . Плотность ненулевых элементов в спектральных компонентах рассчитывалась как число ненулевых элементов в поле спектральных компонент, отнесенное к полному числу (площади) всех спектральных компонент. Очевидно значительное повышение плотности, соответствующее выделению всплесками радиальных волн определенной частоты.

Заключение

В случае многомерного сигнала предлагаемый способ отличается от вейвлет-преобразования отсутствием заранее выбранной многомерной функции разложения. Базис выбирается из элементов исходного сигнала путем вычислений для каждой спектральной компоненты, совокупность которых образует спектр. Каждую спектральную компоненту n -мерного сигнала можно считать суперпозицией одномерного вейвлет-преобразования векторов, образующих исходный сигнал по каждому из n -измерений, на базе функции типа меандр, с определенным масштабным коэффициентом по протяженности. Амплитуда элементов, соответствующих спектральной компоненте, может принимать дискретные положительные или отрицательные значения.

Предлагаемый способ позволяет получить отображение сигнала в фазовое пространство и, следовательно, получить четкую привязку спектральных особенностей сигнала к пространству и (или) времени. Стоит заметить, что отобразить одномерный сигнал на фазовую плоскость позволяет оконное преобразование Фурье.

Предложенный способ декомпозиции n -мерных сигналов может быть выполнен в целочисленной арифметике (без потери точности), что позволяет считать его достаточно быстрым и сравнительно легко реализуемым с помощью аппаратных средств.

Литература

1. Чобану М. Многомерные многоскоростные системы обработки сигналов. – М.: Техносфера, 2009. – 480 с.
2. Шарк Г.Г. Применение вейвлетов для ЦОС. – М.: Техносфера, 2007. – 192 с.
3. Гришенцев А.Ю., Коробейников А.Г. Разработка модели решения обратной задачи вертикального зондирования ионосферы // Научно-технический вестник СПбГУ ИТМО. – 2011. – № 2 (72). – С. 109–113.
4. Сизиков В.С., Кривых А.В. Применение способа эталонных примеров при решении обратной задачи спектроскопии методом регуляризации // Изв. вузов. Приборостроение. – 2011. – Т. 54. – № 9. – С. 44–51.
5. Заявка на изобретение. Способ построения спектра n -мерных неразделимых цифровых сигналов. Гришенцев А.Ю., Коробейников А.Г. №2011126856, от 29.06.2011.

Гришенцев Алексей Юрьевич

– Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики, кандидат технических наук, доцент, tigerpost@ya.ru

Коробейников Анатолий Григорьевич

– Санкт-Петербургский филиал Института земного магнетизма, ионосферы и распространения радиоволн им. Н.В. Пушкова РАН, доктор технических наук, профессор, зам. директора, Korobeynikov_A_G@mail.ru

УДК 004.274

ОРГАНИЗАЦИЯ И ИНСТРУМЕНТАЛЬНЫЕ СРЕДСТВА РЕКОНФИГУРИРУЕМЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ

А.С. Румянцев

Приводятся основные понятия, концепции и особенности реконфигурируемых вычислительных систем. Определены критерии классификации данных систем, выполнено деление наиболее известных из них по предложенным критериям. Представлены существующие подходы к оценке эффективности реконфигурируемых вычислительных систем. Рассмотрено состояние инструментальных средств проектирования для данных систем.

Ключевые слова: реконфигурируемые вычислительные системы, «зернистость» вычислений, критерии классификации вычислительных систем, оценка эффективности вычислительных систем.

Введение

Стремительный прогресс в полупроводниковой электронике вместе с появлением совершенно новых областей ее применения неизбежно приводит не только к повышению требований со стороны вычислительной техники, но и к большей степени зависимости этих требований от решаемой задачи. По-

вышение уровня интеграции микросхем позволяет поместить сотни миллионов транзисторов на одном кристалле, но это также ужесточает требования к процессу разработки и средствам проектирования, увеличивая полное время разработки и единовременные затраты на создание интегральных схем [1]. Все вышеупомянутые тенденции, вкуче с ограничением по энергопотреблению и стремлением к уменьшению времени выхода продукта на рынок, приводят к необходимости реконфигурируемости в будущих проектах интегральных схем.

Цель данной работы состоит в том, чтобы определить критерии классификации и выполнить деление наиболее известных реконфигурируемых вычислительных систем (РВС) в соответствии с этими критериями, а также представить подходы к проектированию систем на их основе. Кроме того, в работе обобщены основные понятия, концепции и особенности РВС.

Основные понятия и концепции РВС

Термин «реконфигурируемая вычислительная система» (РВС) до сих пор не имеет устоявшегося определения, однако среди множества определений РВС можно выделить следующие определения как наиболее точные:

- система, которая обладает возможностью «аппаратного программирования» (изменение способа использования аппаратных блоков посредством определенного набора управляющих линий) [2];
- система, обладающая способностью к изменению «воплощаемой» при реализации модели вычислений [3].

Основные концепции, на которых основаны РВС:

1. разработка проблемно-ориентированных аппаратных блоков с учетом уровня гранулярности вычислительного процесса и обрабатываемых данных для определенного класса задач;
2. использование параллелизма при выполнении вычислений за счет наличия нескольких обрабатывающих блоков и конвейеризации;
3. реконфигурация проблемно-ориентированных аппаратных блоков для адаптации к изменившимся параметрам решаемой задачи (класса задач), для совместного и повторного использования вычислительных ресурсов и т.д.;
4. применение, помимо реконфигурируемых проблемно-ориентированных блоков (РПОБ), процессоров общего назначения и специализированных сверхбольших интегральных схем (СБИС) в случаях, когда прямое использование РПОБ неэффективно в силу различных условий (например, ограничений реального времени и т.д.).

Реализация определенной функциональности вычислительной системы аппаратно дает лучшую энергоэффективность с точки зрения производительности на потребляемую мощность, но данный подход имеет ограниченную гибкость для удовлетворения будущих или изменчивых требований к системе. Архитектуры на основе процессоров, в особенности на основе процессоров общего назначения, обеспечивают лучшую гибкость, но имеют низкую энергоэффективность с точки зрения производительности на потребляемую мощность. РВС должны занять пустующую нишу между специализированными системами и системами общего назначения (рис. 1) [4].

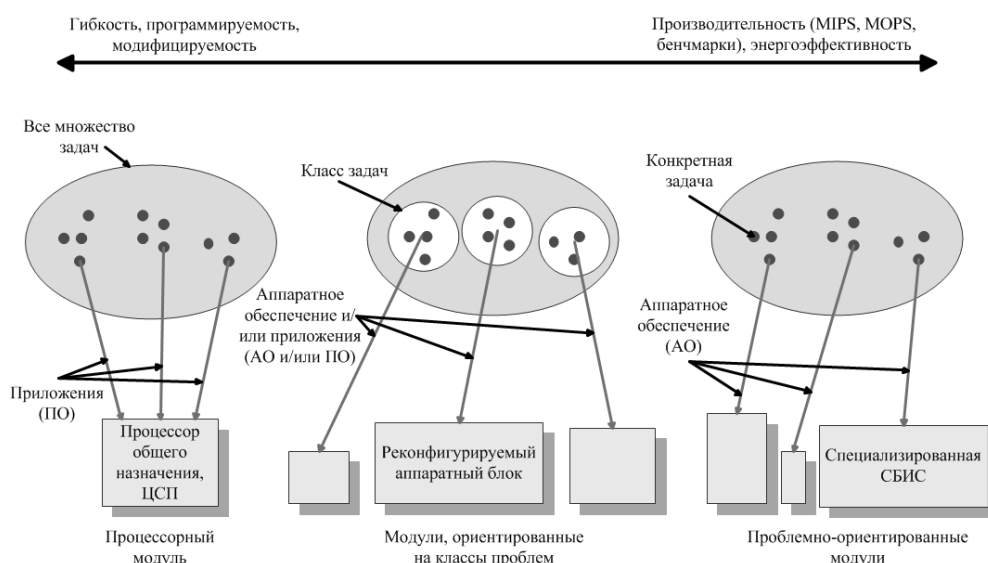


Рис. 1. Спектр задач и возможных программно-аппаратных реализаций:
ПО – программное обеспечение; ЦСП – цифровой сигнальный процессор

Критерии классификации РВС

На текущий момент не существует устоявшегося множества критериев классификации для РВС. Большинство работ, посвященных РВС и их применению, используют различные наборы критериев, подчас противоречащие друг другу. В работе предложен набор критериев классификации для РВС и выполнено распределение наиболее значимых и перспективных из них по данным критериям (табл. 1).

РВС	Критерии классификации РВС						
	1	2	3	4	5	6	7
<i>Butter</i> [5]	шинный блок	крупно-зернистая (32 бит)	+, <i>shift</i> , ×, <i>float</i>	2D-массив	динамическая частичная	ручное	мульти-медиа
<i>DP_FPGA</i>	функциональный блок	смешанная (1 и 4 бита)	<i>bit level</i> , +, <i>shift</i> , <i>regs</i>	2D-массив	динамическая полная	полуавтоматическое (<i>FPGA</i>)	ускорение типовых вычислительно интенсивных задач
<i>DreAM</i>	сопроцессор	крупнозернистая (8, 16 бит)	+, <i>shift</i> , <i>regs</i>	2D-массив	динамическая частичная	нет данных	алгоритмы беспровод. сетей
<i>Garp</i> [6]	сопроцессор	мелкозернистая (2 бит)	<i>bit level</i> , +, <i>shift</i> , <i>regs</i>	2D-сеть	динамическая полная	автоматическое (<i>SUIF based C-compile</i>)	ускорение обработки циклов
<i>MORA</i> [7]	шинный блок	крупно-зернистая (8, 16 бит)	+, ×, <i>regs</i>	2D-сеть	динамическая частичная	нет данных	мульти-медиа
<i>Pipe-Rench</i>	сопроцессор	крупно-зернистая (128 бит)	+, <i>shift</i> , <i>regs</i>	1D-массив	статическая полная	автоматическое (<i>PipeRench tools</i>)	распараллеливание задач
<i>RaPiD</i>	сопроцессор	крупно-зернистая (16 бит)	+, ×, <i>regs</i>	1D-массив	статическая полная	автоматическое (<i>HLL-RaPiD-C</i>)	распараллеливание задач
<i>REMARK</i>	функциональный блок	крупно-зернистая (16 бит)	+, <i>shift</i> , ×, <i>regs</i>	2D-сеть	статическая полная	автоматическое (<i>modif. GCC</i>)	мульти-медиа
<i>МАНГУ-CT</i> [8]	сопроцессор	мелко-зернистая (1 бит)	<i>bit level</i> , ×, <i>shift</i> , <i>regs</i>	2D-сеть	динамическая полная	автоматическое (<i>HLL-COLAMO</i>)	цифровая обработка сигналов и данных

Таблица 1. Распределение наиболее известных РВС по предложенным критериям классификации

1. Роль и расположение реконфигурируемых узлов в системе: функциональный блок центрального вычислителя или контроллера, сопроцессор, вычислительный блок, подсоединенный к системной шине (шинный блок) и вычислительный блок, внешний по отношению к плате или чипу.
2. Степень гранулярности реконфигурируемых узлов РВС (степень зернистости вычислений и обрабатываемых данных): мелкозернистая (к примеру, минимальный логический блок программируемой логической интегральной схемы (ПЛИС), крупнозернистая (к примеру, процессорный узел в многопроцессорной системе) и смешанная.
3. Типы аппаратных ресурсов доступные в РВС. По данному критерию РВС можно разделить на классы в зависимости от поддерживаемого множества операций – сложения, умножения, деления, сдвига на переменную величину и т.д.
4. Способ организации коммуникационной среды РВС:
 - плоская организация структуры РВС (представляет собой одноуровневую сеть соединений реконфигурируемых узлов, блоков памяти и внешних интерфейсов);
 - иерархическая организация структуры РВС (содержит несколько уровней соединения реконфигурируемых узлов и иных ресурсов).
5. Метод и степень реконфигурации вычислительных узлов в РВС.
 - Методы реконфигурации РВС бывают следующих видов:

- статическая реконфигурация (выполняется до момента использования системы по целевому назначению и остается неизменной на всем протяжении времени работы системы);
 - динамическая реконфигурация (может происходить во время работы системы, причем множество раз).
 - По степени реконфигурации вычислительных узлов PBC делятся на следующие классы:
 - обладающие возможностью только полной реконфигурации;
 - обладающие возможностью частичной реконфигурации.
6. Наличие инструментальных средств, степень автоматизации процесса проектирования и возможность интеграции в существующие цепочки проектирования.
7. Целевая область применения (решаемый класс проблем) – например, ускорение обработки циклов, распараллеливание вычислительно интенсивных задач, мультимедиа и т.д.

Подходы к оценке эффективности PBC

При проектировании вычислительной системы разработчик зачастую сталкивается с необходимостью выбора между несколькими альтернативными способами реализации одного и того же функционала на базе различных платформ. В подобных случаях для обоснованного выбора между имеющимися альтернативами используются методы многокритериальной оптимизации. Чтобы использовать методы многокритериальной оптимизации, все множество рассматриваемых альтернатив должно иметь ортогональный набор метрик, позволяющий сравнивать их между собой с учетом возлагаемых на систему задач. На текущий момент существует достаточно много метрик, используемых при сравнении вычислительных систем, решающих один класс задач: отношение достигаемой пропускной способности (средней скорости формирования результирующих данных) к занимаемой системой площади (в квадратных единицах половины минимальной ширины канала транзистора в используемой технологии) (критерий Дехона) [9], отношение потребляемой мощности к планируемой пропускной способности системы и т.д.

Однако данные метрики не способны полностью отразить все аспекты PBC (например, возможность реализации определенной задачи с использованием разных алгоритмов в зависимости от того, будут ли они реализованы программно или аппаратно). Для решения данной задачи было предложено несколько видов метрик, из которых хотелось бы отметить следующие три метрики, которые в наибольшей степени подходят для использования в качестве метрик сравнения PBC.

1. «Ускорение» – ускорение, получаемое по сравнению с использованием процессора общего назначения или цифрового сигнального процессора:

$$\text{"ускорение"} = \frac{T_{\text{выпЦПУ}}}{T_{\text{выпPBC}} + T_{\text{настройки}} + T_{\text{коммутации}}}.$$

Метрика была предложена как часть методологии оценки PBC в работе [10].

2. Реманентность (R). Метрика, позволяющая оценить динамические характеристики реконфигурируемости PBC:

$$R = \frac{N_a \times F_e}{N_c \times F_c},$$

где N_a – количество вычислительных узлов в PBC; F_e – частота тактирования PBC; N_c – количество вычислительных узлов, которые могут быть реконфигурируемы за один цикл; F_c – максимальная частота реконфигурации.

3. Вычислительная плотность (OD). Метрика, позволяющая оценить масштабируемость PBC:

$$OD = \frac{N_a}{A},$$

где N_a – количество вычислительных узлов в PBC; A – занимаемая PBC площадь. Две последние метрики для оценки PBC вводятся в работе [11].

Состояние средств разработки систем на базе PBC

Проектирование системы на базе PBC намного сложнее, чем проектирование систем на платформе с преобладанием программной или аппаратной компоненты. Системы на базе PBC подразумевают беспрецедентную степень конвергенции между программным и аппаратным обеспечением от стадии проектирования до стадии непосредственного использования системы. Вопросы совместного программно-аппаратного проектирования (Software/Hardware Codesign) приобретают в области PBC решающее значение. Большинство существующих средств разработки систем на базе PBC поддерживает одну из трех цепочек проектирования, показанных на рис. 2. Перечень применяемых подходов на каждой из стадий процесса проектирования PBC приведен в табл. 2.



Рис. 2. Существующие цепочки проектирования систем на основе РВС [2]. Серые прямоугольники обозначают ручной, а белые – автоматический шаг(и) процесса проектирования

Стадия	Применяемые подходы
Формализация решаемой задачи	<ul style="list-style-type: none"> графическое описание (SPEAR) специализированный низкоуровневый язык язык высокого уровня (C, Java)
Разделение на программную и аппаратную компоненты	<ul style="list-style-type: none"> ручное (pragma, специальные вызовы) автоматическое (например, в граф потока данных и управления (CDFG))
Анализ и отображение полученного описания на вычислительные узлы РВС	<ul style="list-style-type: none"> прямое отображение библиотека типовых функций и операций интеллектуальное отображение
Непосредственное размещение полученного отображения на РВС	<ul style="list-style-type: none"> алгоритм модельной «закалки» жадные алгоритмы дерево соответствия
Организация маршрутизации между вычислительными узлами РВС	<ul style="list-style-type: none"> статическая маршрутизация варианты алгоритма поиска пути жадные алгоритмы маршрутизации

Таблица 2. Подходы к проектированию систем на базе РВС на разных стадиях

Заключение

Область реконфигурируемых систем является многообещающей и сравнительно молодой исследовательской сферой, поэтому огромное множество особенностей реконфигурируемых систем, их процесс проектирования и инструментальные средства являются открытой темой для исследований и экспериментов.

Предложенные в работе критерии классификации РВС могут быть полезными при выборе архитектуры РВС в зависимости от проектных ограничений и целевой области использования системы. Представленные подходы к оценке эффективности РВС помогут выбрать подходящий вариант РВС исходя из имеющихся альтернатив.

Литература

1. International Technology Roadmap for Semiconductors [Электронный ресурс]. – Режим доступа: <http://public.itrs.net>, свободный. Яз. англ. (дата обращения 07.04.2011).
2. Compton K., Hauck S. Reconfigurable computing: A survey of systems and software // ACM Computer Survey. – 2002. – V. 34. – № 2. – P. 171–210.
3. Платунов А.Е. Теоретические и методологические основы высокоуровневого проектирования встраиваемых вычислительных систем. Докторская диссертация. – СПб: СПбГУ ИТМО, 2010 [Электронный ресурс].

- ресурс]. – Режим доступа: <http://www.dslib.net/avtomat-proektirovanie/platunov.html>, платный. Яз. рус. (дата обращения 20.03.2011).
4. Rabaey J.M. Reconfigurable Computing: The Solution to Low Power Programmable DSP // IEEE ICASSP Conference. – 1997. – V. 1. – P. 275–278.
 5. Garzia F. CREMA: A coarse-grain reconfigurable array with mapping adaptiveness // IEEE Field Programmable Logic and Applications Conference. – 2009. – V. 1. – P. 708–712.
 6. Callahan T.J., Hauser J.R., Wawrzynek J. The Garp architecture and C compiler // IEEE Computer Society. – 2000. – V. 33. – № 4. – P. 62–69.
 7. Chalamalasetti S.R., Purohit S. et al. MORA – An Architecture and Programming Model for a Resource Efficient Coarse Grained Reconfigurable Processor // IEEE Adaptive Hardware and Systems Conference. – 2009. – P. 389–396.
 8. Дордопуло А.И., Каляев И.А., Левин И.И., Семерников Е.А. Семейство многопроцессорных вычислительных систем с динамически перестраиваемой архитектурой // Сб. материалов Международной научно-технической конференции «Многопроцессорные вычислительные и управляющие системы». – Таганрог: Изд-во ТТИ ЮФУ, 2007. – Т. 1. – С. 11–17.
 9. DeHon A. Comparing computing machines // Proceedings of SPIE 3526. – 1998. – P. 124–133.
 10. Bouallegue A., Abdallah F. System Level Methodology Evaluation of Reconfigurable Computing Architectures // 2nd ICTTA Conference. – 2006. – P. 2567–2572.
 11. Benoit P. Metrics for reconfigurable architectures characterization: remanence and scalability // PDP Symposium. – 2003. – P. 8–16.

Румянцев Александр Сергеевич

– Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики, аспирант, dark-streamray@gmail.com