

Высокопроизводительные реконфигурируемые вычислительные системы на основе ПЛИС VIRTEX-7

И.И. Левин¹, А.И. Дордопуло², И.А. Каляев¹, В.А. Гудков¹

НИИ многопроцессорных вычислительных систем имени академика А.В. Каляева Южного федерального университета, г. Таганрог, Россия¹, Южный научный центр Российской академии наук, г. Ростов-на-Дону, Россия²

В статье рассматриваются сравнительные характеристики реконфигурируемых вычислительных систем (РВС) на основе вычислительных модулей 24V7-750 и «Тайгета», содержащих программируемые логические интегральные схемы (ПЛИС) семейства Xilinx Virtex-7. Отличительными характеристиками РВС на основе ПЛИС Xilinx Virtex-7 по сравнению с аналогичными системами на основе ПЛИС Xilinx Virtex-6 являются увеличение производительности в 1,7 раза и улучшение остальных технико-экономических показателей: удельной производительности, энергоэффективности и др. Рассматривается решение прикладной задачи с помощью разработанного комплекса средств разработки прикладного программного обеспечения для РВС.

1. Введение

Реконфигурируемые вычислительные системы (РВС), широко используемые для решения вычислительно трудоемких задач в различных областях науки и техники, по сравнению с многопроцессорными вычислительными системами кластерной архитектуры обладают рядом существенных преимуществ: высокими реальной и удельной производительностями при решении задач, высокой энергоэффективностью и др. В полной мере преимущества от использования реконфигурируемых вычислительных систем (РВС) достигаются при использовании в качестве основного вычислительного элемента аппаратного ресурса программируемых логических интегральных схем (ПЛИС) [1], объединенных в единое вычислительное поле высокоскоростными каналами передачи данных.

Различные отечественные и зарубежные компании создают как отдельные ускорители с одним-двумя кристаллами ПЛИС, так и целые вычислительные комплексы. Такие фирмы как Nallatech [2] и Pico Computing [3] выпускают ряд ускорителей и несущих плат с небольшим числом (до четырех) кристаллов ПЛИС, которые используются в создании серверов и гибридных кластерных систем фирмами HP и IBM. Компании Convey [4] и Maxeler Technologies [5] создают гибридные суперкомпьютеры на основе собственных гетерогенных кластерных узлов, каждый из которых может содержать от 1 до 4 кристаллов ПЛИС и несколько универсальных процессоров. Похожее решение используется и компанией SRC [6], которая выпускает узлы, названные MAP processor, для стойки (MAPstation) форм-фактором 1U, 2U и 4U. MAPstation 1U содержит один MAP processor. MAPstation 2U содержит до трех MAP processor. MAPstation 4U может содержать до 10 различных модулей – MAP processor, модуль с универсальным микропроцессором или модуль памяти.

В отличие от перечисленных фирм в НИИ многопроцессорных вычислительных систем имени академика А.В. Каляева Южного федерального университета (НИИ МВС ЮФУ, г. Таганрог) разрабатываются суперкомпьютеры с возможностью объединения нескольких печатных плат в единый вычислительный контур, где основным вычислительным ресурсом системы являются кристаллы ПЛИС. Методы разработки и создания таких систем успешно развиваются в НИИ МВС ЮФУ. Концепция построения РВС [7] позволила создать целый ряд высокопроизводительных систем различных архитектур и конфигураций, выпускаемых серийно и предназначенных для решения вычислительно трудоемких задач различных предметных областей. Опыт успешной эксплуатации в организациях и ведомствах Российской Федерации различных конфигураций ранее созданных РВС на основе ПЛИС Xilinx семейств Virtex-4, Virtex-5 и Virtex-6 использовался для разработки перспективных реконфигурируемых вычислительных систем на основе ПЛИС Xilinx Virtex-7, описание которых представлено в настоящей статье.

2. РВС на основе ПЛИС Xilinx Virtex-7

2.1 РВС на основе ВМ «Плеяда»

Реконфигурируемая вычислительная система РВС-7 на основе ПЛИС Virtex-7, разработанная по государственному контракту №14.527.12.0004 от 03.10.2011, содержит вычислительное поле из 576 микросхем ПЛИС Virtex-7 XC7V585T-FFG1761 объемом 58 миллионов эквивалентных вентилей каждая, конструктивно объединенных в один вычислительный шкаф высотой 47U с пиковой производительностью 10^{15} операций с фиксированной запятой в секунду.

Основным структурным компонентом РВС-7, предназначенным для установки в стандартную 19" вычислительную стойку, является вычислительный модуль (ВМ) 24V7-750 (ВМ «Плеяда»), в состав которого входят: четыре платы вычислительного модуля (ПВМ) 6V7-180, представленные на рис. 1; управляющий модуль УМ-7; подсистема питания; подсистема охлаждения и другие подсистемы. Фотография ВМ 24V7-750 представлена на рис. 2.

В состав ПВМ 6V7-180 входят:

- вычислительное поле, состоящее из 6-ти ПЛИС XC7V585T-1FFG1761 семейства Virtex-7 производства фирмы Xilinx. Между собой ПЛИС вычислительного поля соединены последовательно, передача данных осуществляется по 144 дифференциальным линиям LVDS-интерфейса на частоте 800 МГц;

- контроллер ПВМ, выполненный на ПЛИС XC6V130T-1FFG1156C производства Xilinx;

- 12 каналов интерфейса LVDS на частоте 800 МГц по 25 дифференциальных пар каждый (разъёмы типа SS4) для связи с другими вычислительными модулями;

- узлы основной и резервной загрузки ПЛИС по интерфейсам JTAG-1 и JTAG-2;

- подсистема синхронизации (генераторы ECS-2033-250-BN и распределители тактовых импульсов IDT5T9316NLI;

- распределённая память в составе 12-ти микросхем динамической памяти (MT47H128M16HR-25E с организацией 128 М*16 и частотой записи/чтения до 400 МГц). К ПЛИС вычислительного поля, а также к ПЛИС контроллера базового модуля, подключено по две микросхемы памяти DDR2. Объем оперативной памяти на ПВМ - 3 Гбайта;

- 2 канала интерфейса LVDS по 20 дифференциальных пар для связи с персональным компьютером и внешней аппаратурой;

- подсистема загрузки ПЛИС;

- подсистема питания, в состав которой входят DC-DC преобразователи напряжения, вырабатывающие напряжения питания: +1 В – питание ядер ПЛИС; +2,5 В – питание узла тактирования; +1,8 В – питание микросхем памяти DDR2, +3,3 В – буферных каскадов ПЛИС.

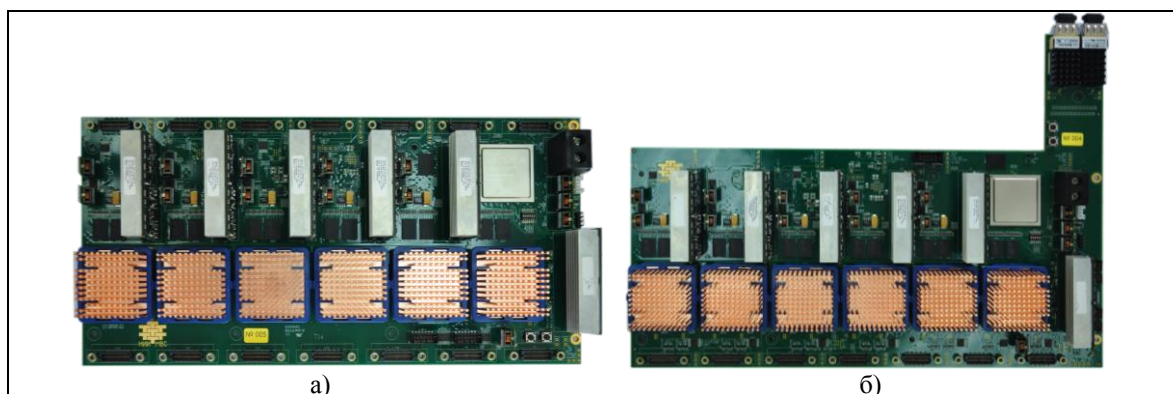


Рис. 1. Фотография платы ПВМ 6V7-180 (а – ПВМ №0-2 ВМ 24V7-750, б - ПВМ №3 ВМ 24V7-750 с оптическим разъемом для связи с другими ВМ 24V7-750)

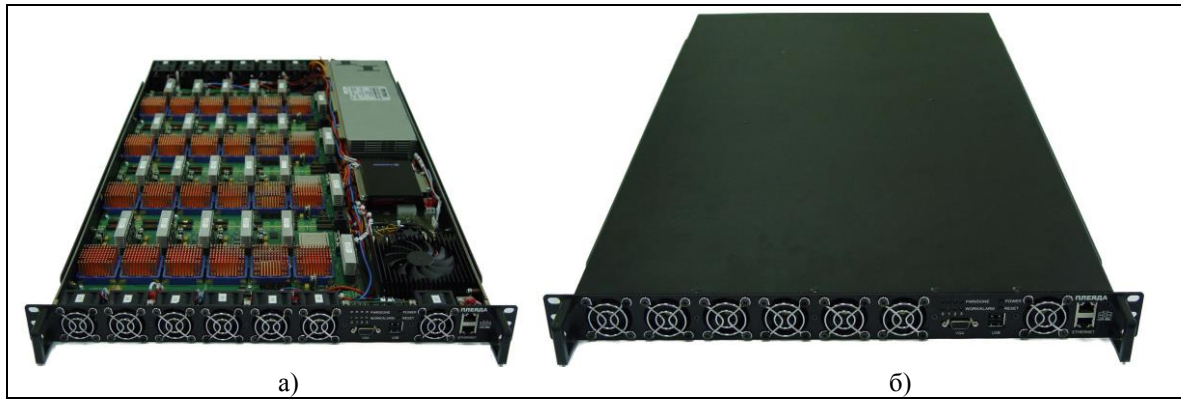


Рис. 2. Фотография вычислительного модуля (ВМ) 24V7-750 (а - со снятой верхней крышкой, б - с установленной верхней крышкой)

Производительность одной ПВМ 6V7-180 составляет 645,9 Гфлопс при обработке 32-разрядных данных с плавающей запятой, а производительность ВМ 24V7-750 составляет 2,58 Тфлопс при обработке 32-разрядных данных с плавающей запятой. Производительность РВС-7 при комплектации от 24 до 36 ВМ 24V7-750 составит от 62 до 93 Тфлопс при обработке 32-разрядных данных с плавающей запятой и 19,4 – 29,4 Тфлопс при обработке 64-разрядных данных с плавающей запятой. Областью применения РВС-7 и вычислительных комплексов на ее основе является решение задач цифровой обработки сигналов и многоканальная цифровая фильтрация.

2.2 РВС на основе ВМ «Тайгета»

На основе ПЛИС Virtex-7 также разработан новый вычислительный модуль «Тайгета» в конструктивном исполнении высотой 2U, предназначенный для установки в стандартную 19" вычислительную стойку. ВМ «Тайгета», представленный на рис. 3а, содержит четыре ПВМ 8V7-200, представленных на рис. 3б, соединенных быстрыми LVDS-каналами, встроенную управляющую ЭВМ, систему питания, систему управления, систему охлаждения и другие подсистемы.

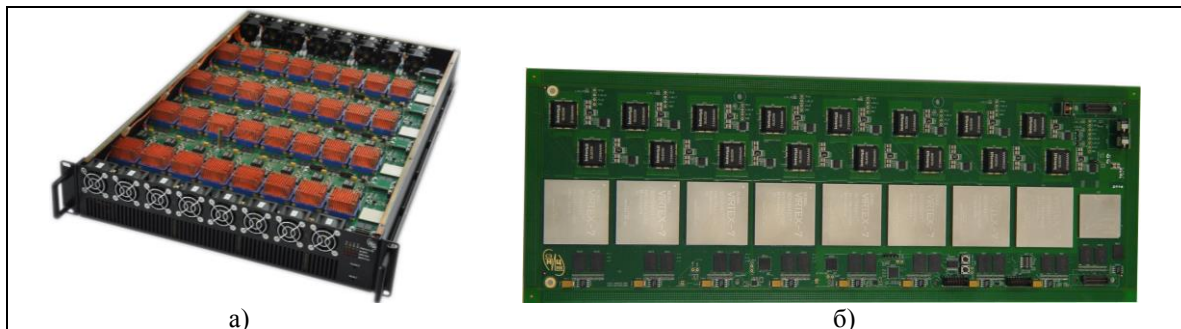


Рис. 3. Фотография ВМ «Тайгета» (а – фотография ВМ «Тайгета» со снятой верхней крышкой, б – фотография ПВМ 8V7-200)

ПВМ 8V7-200, лежащая в основе ВМ «Тайгета», представляет собой 20-слойную печатную плату с двухсторонним монтажом элементов, на которой располагаются 8 ПЛИС типа XC7VX485T-1FFG1761, содержащих 48,5 миллионов эквивалентных вентилей, 16 микросхем распределенной памяти SDRAM типа DDR2 общим объемом 2 Гбайт, интерфейсы LVDS и Ethernet и другие компоненты.

Производительность одной ПВМ 8V7-200 составляет 667 ГФлопс при обработке 32-разрядных данных с плавающей запятой, а производительность ВМ «Тайгета» составляет

2,66 ТФлопс при обработке 32-разрядных данных с плавающей запятой. Производительность РВС на основе ВМ «Тайгета» при комплектации от 18 ВМ «Тайгета» составляет 48 Тфлопс при обработке 32-разрядных данных с плавающей запятой и 23 Тфлопс при обработке 64-разрядных данных с плавающей запятой. РВС на основе ВМ «Тайгета» позволяет сократить стоимость поставки вычислительной системы для задач определенного класса (например, для задач символьной обработки), обеспечивая при этом такую же производительность, как и РВС-7 с 24 ВМ 24V7-750. Поэтому ВМ «Тайгета» являются наиболее предпочтительными для построения высокопроизводительных вычислительных комплексов для решения задач символьной обработки данных, поскольку обеспечивают существенное конкурентное преимущество по большинству технико-экономических параметров: удельной производительности, энергоэффективности и др.

3. Программное обеспечение РВС на основе ПЛИС Xilinx Virtex-7

Для вычислительных систем на основе ПЛИС Virtex-7 сохраняется преемственность принципов программирования [7-10]: программирование всех рассмотренных вычислительных модулей осуществляется с помощью единого комплекса системного программного обеспечения, поддерживающего структурно-процедурные методы организации вычислений и определяющего не только организацию параллельных процессов и потоков данных, но и структуру вычислительной системы в поле логических ячеек ПЛИС. Наиболее характерной отличительной особенностью комплекса программного обеспечения РВС на основе ПЛИС Xilinx Virtex-7 является поддержка проблемно-ориентированных софт-архитектур, позволяющих создавать и программировать макрообъекты, представляющие собой совокупность вычислительных устройств, выполняющих определенную группу команд и соединенных между собой коммутационной системой. Это обеспечивает при тех же принципах программирования и использовании языка высокого уровня для программирования РВС возможность простой адаптации программных компонентов средств разработки для РВС при переходе на новые топологии ПВМ без внесения существенных изменений в код программных компонентов комплекса, а также позволяет сократить время решения прикладных задач.

Большинство существующих коммерческих систем проектирования (Xilinx ISE, Altium Designer и другие) обеспечивает в рамках одного проекта работу только с одним кристаллом ПЛИС. При разработке решения на нескольких ПЛИС инженеру-схемотехнику приходится самому распределять элементы вычислительной структуры алгоритма решаемой задачи между различными проектами, которые будут соответствовать определенным кристаллам ПЛИС многокристальной РВС и учитывать топологию связей между кристаллами ПЛИС. Поэтому программирование многокристальных РВС с помощью указанных средств разработки требует участия в программировании системы как специалиста-схемотехника, создающего конфигурацию вычислительной системы с учетом особенностей ее архитектуры и элементной базы, так и прикладного программиста, создающего параллельную программу, описывающую потоки данных в созданной схемотехником виртуальной вычислительной структуре.

Процесс программирования РВС делится на два этапа. На первом этапе создается вычислительная структура для решения прикладной задачи. На втором этапе программируется организация вычислительного процесса в созданной вычислительной структуре с целью получения результата. При этом наиболее труден именно первый этап, поскольку большинство пользователей привыкли программировать организацию вычислительного процесса, опираясь на неизменную аппаратную платформу компьютера.

Возникающие у пользователей проблемы побуждают разработчиков РВС искать пути их преодоления. Одним из таких путей является использование универсального макрообъекта, который располагается в одной ПЛИС, имеет структуру, не изменяющуюся в зависимости от размещения в ПЛИС базового модуля, а также имеет методы представления задачи в структурно-процедурной форме. Реконфигурируемые системы, у которых был реализован данный принцип, получили название РВС с универсальной софт-архитектурой. С точки зрения пользователя, реконфигурируемая система с универсальной софт-архитектурой наиболее близка многопроцессорной ЭВМ на основе универсальных микропроцессоров. Программирование РВС на

уровне макропроцессоров в универсальной софт-архитектуре относится к первому – верхнему уровню программирования РВС [11].

На другом уровне программирования РВС находятся специализированные решения, которые создаются схемотехниками из примитивов ПЛИС – Logic Cells, BlockRAM, DSP и т.п. Схемотехники и алгоритмисты создают в рамках базовой архитектуры РВС специализированную вычислительную структуру, на которой пользователи будут программировать решение прикладной задачи. Программирование на уровне примитивов ПЛИС относится к нижнему уровню программирования РВС.

На сегодняшний день созданы программные методы и средства создания макрообъектов на различных уровнях программирования РВС (рис. 4).

Опыт разработки РВС на базе ПЛИС показывает, что для каждого класса задач можно подобрать определенный набор вычислительных узлов, на основе которых создаются оптимальные вычислительные структуры (макрообъекты), наиболее эффективно решающие задачи данного класса. При этом наибольшая эффективность структур достигается в том случае, если вычислительные элементы располагаются в поле логических блоков всего множества ПЛИС базового модуля или в вычислительном поле РВС. Это позволяет определить макрообъект [7,11] как архитектурно неделимую совокупность функциональных узлов (объектов), объединенных пространственной коммутационной системой и размещенных в одной или нескольких ПЛИС. При этом возможно изменение количества функциональных узлов, параметров узлов (разрядности операндов, числа информационных каналов, системы команд и т.п.), но не их назначения.



Рис. 4. Уровни программирования РВС

Каждый макрообъект может содержать множество разнородных функциональных узлов, объединенных между собой внутренним пространственным коммутатором, который позволяет соединить элементы в функционально законченное вычислительное устройство, а также обеспечивает корректное взаимодействие и транзит информации между различными макрообъектами. Для использования макрообъекта при программировании мультиконвейерных структур в рамках вычислительного поля РВС необходимо на этапе проектирования выполнить следующие условия:

- определить номенклатуру функциональных узлов, которые допустимо использовать при создании макрообъектов для различных предметных областей;
- определить параметры функциональных узлов, которые могут изменяться в процессе конструирования макрообъекта (разрядность, количество входов/выходов, типы внешних интерфейсов, систему команд функциональных узлов и т.п.);
- разработать библиотеки функциональных узлов на схемотехническом уровне, которые фактически являются IP-ядрами, созданными в терминах логических схем ПЛИС или в терминах языка описания логических схем VHDL;

- создать набор типовых интерфейсов, посредством которых можно соединить функциональные узлы в вычислительные структуры согласно информационному графу алгоритма и которые также являются IP-ядрами;

- определить структуры макрообъектов для различных предметных областей.

Таким образом, макрообъект является «заготовкой», которая может доопределяться пользователем в процессе создания конкретного технического решения, а затем тиражироваться в схемотехническом ресурсе ПЛИС базовых модулей в нужном количестве и соединяться с подобными или другими макрообъектами в вычислительные структуры, которые оптимально соответствуют структуре решаемой задачи. С точки зрения программиста РВС макрообъект – это виртуальное проблемно-ориентированное устройство с программируемыми параметрами, «заготовка» при программировании архитектуры РВС, и может быть реализовано в ее вычислительном поле без изменения функций.

Информационные потоки в пределах макрообъекта и вычислительной структуры, построенной на их основе, образуются с помощью контроллеров распределенной памяти, которые организуют скоростные параллельные обращения к сегментированной внутренней и внешней памяти макрообъекта. Динамические коммутаторы обеспечивают переключение информационных потоков между источниками и приемниками, причем изменение адреса источника (приемника) может производиться произвольно с каждым данным. Совокупность статических и динамических коммутаторов обеспечивает возможность создания разнообразных вычислительных структур для решения различных задач из проблемной области, для которой создавался данный макрообъект.

Понятие макрообъекта связано с аппаратной платформой РВС, под которой традиционно подразумеваются: тип, количество ПЛИС и связи между ними, организация памяти, связи между ПЛИС и памятью, устройства ввода/вывода и т.п., физически реализованные в виде микросхем и устройств в составе РВС. Аппаратную платформу, созданную пользователем или разработчиками РВС в поле ПЛИС в дополнение к АП в классическом понимании, будем называть реконфигурируемой аппаратной платформой (РАП). Под термином «архитектура РВС» будем понимать совокупность аппаратной платформы, программных средств и их взаимодействие, которые существенны для пользователя РВС при решении прикладной задачи. Под термином «софт-архитектура РВС» понимается архитектура РВС с РАП, в которой можно без перезагрузки конфигурации ПЛИС вычислительного поля путем программной настройки создавать необходимые вычислительные структуры для решения прикладных задач пользователя.

Аппаратные платформы для 1 и 4 уровней программирования (см. рис.4) не могут быть изменены пользователем РВС. В этом ряду особое место занимает уровень 2, соответствующий аппаратной платформе уровня 3, программного уровня, поскольку в его создании могут принимать участие пользователи РВС.

Первый уровень программирования РВС базируется на универсальной архитектуре, аппаратная платформа которой создается разработчиками РВС и которая недоступна для изменений со стороны пользователя. Программирование прикладных задач для этого уровня выполняется на языке высокого уровня COLAMO [1-4]. Данный уровень программирования допускает динамическую перестройку вычислительных структур в вычислительном поле РВС и, следовательно, решение многокадровых задач при организации структурно-процедурных вычислений [1]. Универсальность архитектуры дает возможность решать любые задачи, но платой за универсальность является снижение эффективности использования оборудования РВС, а следовательно, и невысокая реальная производительность системы.

При программировании на уровне универсальной софт-архитектуры компоненты параллельной программы (поточковый, структурный и процедурный) [7] транслируются в программу на языке ассемблера Argus [1], состоящую из виртуальных макроопераций. После этого синтезатор WaterConstructor [8] отображает информационный граф параллельной программы на архитектуру РВС. После этого транслятор формирует файлы настройки РВС, которые загружаются в аппаратную платформу, настраивая элементарные процессоры и коммутационную среду в соответствии с информационным графом задачи. На рис. 5 представлена схема преобразования компонент параллельной программы на уровне универсальной софт-архитектуры.

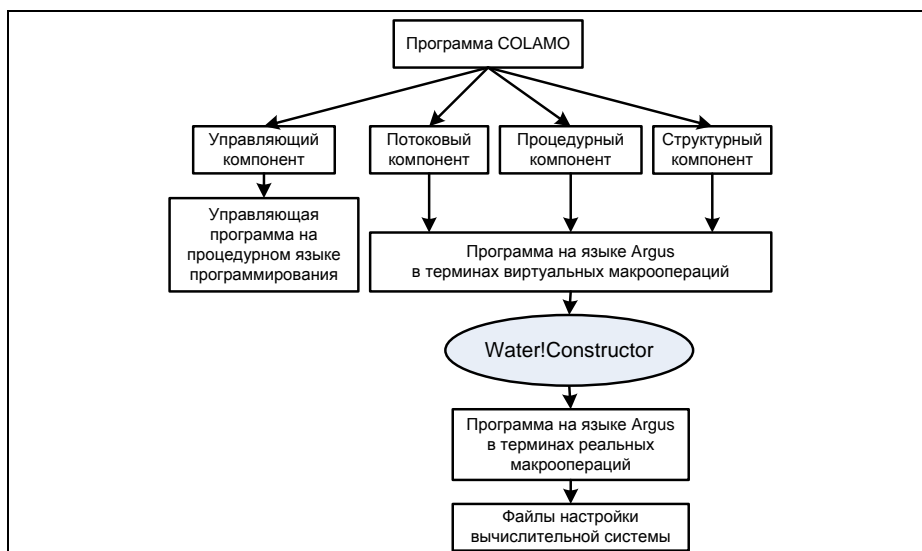


Рис. 5. Преобразование параллельной программы на уровне универсальной софт-архитектуры

В результате в универсальной софт-архитектуре PBC формируется специальная вычислительная структура для решения конкретной задачи. Для изменения вычислительной структуры не нужно перезагружать конфигурацию всего множества ПЛИС, а достаточно программно изменить связи между элементарными процессорами и выполняемые в них команды.

Достоинством программирования на уровне универсальной софт-архитектуры является возможность решения любой задачи. При этом возможна динамическая перенастройка в процессе выполнения задачи за время, не превышающее 100 тактов. Однако универсальность архитектуры приводит к тому, что часть вычислительного ресурса (20-40%) будет простаивать, производительность всей системы снизится, в общем случае составляя 10-50% от пиковой. Кроме того, универсальная софт-архитектура недоступна программисту PBC для модификации, и создание универсальной архитектуры с другой коммутационной средой и другими элементами требует разработки нового описания данной архитектуры PBC и нового синтезатора WaterConstructor.

4. Программирование софт-архитектур для предметных областей

Для устранения основных проблем программирования PBC существуют дополнительные промежуточные уровни программирования, которые сочетают все основные достоинства двух существующих уровней программирования. Софт-архитектура представляет собой динамически-перестраиваемые архитектуры PBC, в которых можно без перезагрузки конфигурационных файлов ПЛИС, а только при помощи программной настройки, формировать вычислительные структуры для решения задач. Программирование на втором уровне заключается в создании программистом софт-архитектур. Третий уровень – программирование софт-архитектур - заключается в решении задач с применением динамически-перестраиваемых софт-архитектур, описанных на втором уровне.

Для создания динамически-перестраиваемой софт-архитектуры необходимо выполнить:

- разработку описания макрообъектной софт-архитектуры на языке SADL;
- трансляцию описания софт-архитектуры в промежуточное представление при помощи синтезатора конфигураций параллельно-конвейерных вычислительных структур FireConstructor;
- размещение элементов софт-архитектуры на аппаратной платформе при помощи синтезатора масштабируемых параллельно-конвейерных процедур SteamConstructor.

Создание софт-архитектуры начинается с разработки описания на языке SADL (Soft-Architecture Development Language). Транслятор языка SADL преобразует текст программы в промежуточное представление, используемое синтезатором FireConstructor для размещения на аппаратной платформе PBC.

Результатом размещения софт-архитектуры на аппаратную платформу являются модифицированный файл промежуточного представления и конфигурационные файлы для ПЛИС, участвующих в размещении софт-архитектуры на аппаратной платформе PBC. После того как софт-архитектура была размещена на аппаратной платформе PBC, она может быть использована для решения различных прикладных задач заданной проблемной области.

При разработке прикладной программы с использованием софт-архитектуры необходимо выполнить:

- разработку параллельной программы на языке высокого уровня COLAMO;
- трансляцию параллельной программы и генерацию структурного, процедурного и потокового компонентов;
- отображение структурного компонента параллельной программы на софт-архитектуру при помощи синтезатора конфигураций параллельно-конвейерных вычислительных структур SteamConstructor;
- трансляцию процедурного компонента параллельной программы на уровень команд устройств софт-архитектуры;
- формирование загрузочного out-файла, содержащего команды элементов софт-архитектуры;
- загрузку конфигурационных файлов ПЛИС, полученных в результате размещения элементов софт-архитектуры на аппаратной платформе реконфигурируемой системы;
- загрузку out-файла;
- загрузку в софт-архитектуру исходных данных решаемой задачи;
- запуск программы на исполнение и выгрузку результатов.

Взаимодействие средств разработки прикладных программ при разработке софт-архитектуры показано на рис. 6.

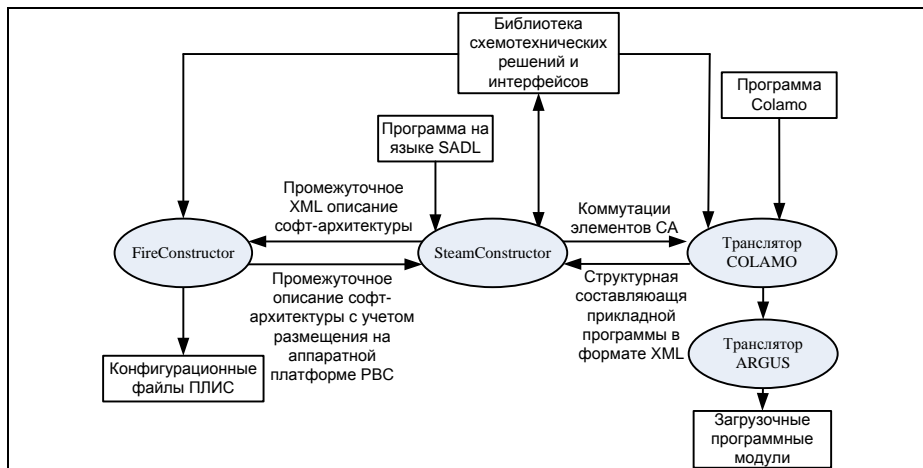


Рис. 6. Взаимодействие средств разработки прикладных программ

Благодаря разработанным программным средствам разработка и модификация софт-архитектур выполняется быстро и не требует привлечения высококвалифицированного специалиста-схемотехника. При этом время, затрачиваемое на создание или модификацию софт-архитектуры, значительно сокращается, а получаемые многокристальные архитектурные решения сравнимы по эффективности с решениями, выполненными специалистами-схемотехниками вручную. В соответствии с основными принципами языка COLAMO параллельные прикладные программы могут быть легко модифицированы для адаптации к доступному вычислительному ресурсу. Автоматизация отображения графов на ресурс PBC позволяет разработчикам прикладных задач мыслить не несколькими ПЛИС, а одной виртуальной ПЛИС с большим логическим объемом. Комплекс разработанных программных средств позволяет программисту PBC разрабатывать и выполнять отладку прикладных параллельных программ для PBC, не вникая в особенности архитектуры PBC, а также самостоятельно создавать и модифицировать различные софт-архитектуры, ориентируясь на предметную область, которой принадлежит решаемая задача.

5. Заключение

Для разработанных ВМ на основе ПЛИС Xilinx Virtex-7 было проведено исследование реальной производительности созданных аппаратных средств. В качестве задачи для исследования производительности ВМ 24V7-750 и ВМ «Тайгета» использовалась задача многоканальной цифровой фильтрации для мониторинга систем цифровой связи третьего и четвертого поколений (КИХ-фильтр высокого порядка), оперирующая данными с фиксированной запятой различной разрядности (16, 24, 32, 34 разряда). Поэтому полученные значения реальной производительности представлены в оп/с, а не в операциях с плавающей запятой: реальная производительность ВМ 24V7-750 составляет 25 Топ/с, а реальная производительность ВМ «Тайгета» - 27 Топ/с. Это позволяет достигать реальной производительности $0,9 \cdot 10^{15}$ оп/с в одной вычислительной стойке для РВС-7 и $0,486 \cdot 10^{15}$ оп/с для РВС на основе ВМ «Тайгета» при решении прикладных задач цифровой обработки сигналов.

Таким образом, конструктивные решения, положенные в основу перспективных вычислительных модулей на основе ПЛИС Xilinx Virtex-7, позволяют сосредоточить в пределах одной вычислительной стойки высотой 47U мощный вычислительный ресурс и обеспечивают удельную производительность РВС на основе ПЛИС Xilinx Virtex-7 на уровне лучших мировых показателей для суперЭВМ с кластерной архитектурой. Это позволяет рассматривать РВС на основе ПЛИС Xilinx Virtex-7 как основу для создания высокопроизводительных вычислительных комплексов нового поколения, обеспечивающих высокую эффективность вычислений и близкий к линейному рост производительности при наращивании вычислительного ресурса.

Литература

1. Каляев А.В., Левин И.И. Модульно-наращиваемые многопроцессорные системы со структурно-процедурной организацией вычислений. М.: Янус-К, 2003. 380 с.
2. <http://www.nallatech.com/>
3. <http://picocomputing.com/>
4. <http://www.conveycomputer.com>
5. <http://www.maxeler.com/>
6. <http://www.srccomp.com/>
7. Каляев И.А., Левин И.И., Семерников Е.А., Шмойлов В.И. Реконфигурируемые мультимедийные вычислительные структуры /Изд. 2-е, перераб. и доп. / Под общ. ред. И.А. Каляева. Ростов-на-Дону: Изд-во ЮНЦ РАН, 2009. 344 с.
8. Каляев И.А., Левин И.И. Семейство реконфигурируемых вычислительных системы с высокой реальной производительностью // Труды международной научной конференции «Параллельные вычислительные технологии» (ПАВТ'2009). Нижний Новгород: электронное издание НГУ имени Н.И. Лобачевского, 2009. С.186-196.
9. Дордопуло А.И., Каляев И.А., Левин И.И., Семерников Е.А. Высокопроизводительные реконфигурируемые вычислительные системы нового поколения // Труды Международной суперкомпьютерной конференции с элементами научной школы для молодежи «Научный сервис в сети Интернет: эксафлопсное будущее». М.: Изд-во МГУ, 2011. С. 42-49.
10. Каляев И.А., Левин И.И., Семерников Е.А., Дордопуло А.И. Реконфигурируемые вычислительные системы на основе ПЛИС семейства Virtex-6 // Сборник трудов Международной научной конференции «Параллельные вычислительные технологии 2011» (ПАВТ 2011). Челябинск-М.: Издательский центр ЮУрГУ [Электронный ресурс], 2011. С. 203–210.
11. I.A. Kalyaev, I.I. Levin, E.A. Semernikov, V.I. Shmoilov. (2012). Reconfigurable multipipeline computing structures. Nova Science Publishers, New York.