УДК 004.274

МИКРОПРОЦЕССОРНОЕ ЯДРО С ФУНКЦИЕЙ ЧАСТИЧНОЙ РЕКОНФИГУРАЦИИ НА FPGA

Босько Н.В.

Белорусский государственный университет информатики и радиоэлектроники[1]
г. Минск, Республика Беларусь

Луцик Ю.А. – канд. технических наук, доцент

Аннотация. В данной работе рассматривается архитектура синтезируемого микропроцессора, использующего функцию частичной реконфигурации FPGA для изменения отдельных частей своей микроархитектуры с целью обеспечения наиболее благоприятных условий для решения конкретной задачи и увеличения производительности.

Ключевые слова. FPGA, частичная реконфигурация, синтезируемые микропроцессорные ядра, реконфигурируемые вычислители.

Введение

FPGA (Field Programmable Gate Array) – реконфигурируемые микросхемы, которые могут быть запрограммированы на реализацию пользовательских логических схем.

Синтезируемые микропроцессорные ядра для FPGA представляют собой исходный код на языке описания аппаратуры (например, VHDL или Verilog), предназначенный для реализации на микросхеме FPGA. Зачастую они используются в системах, где от них требуется решение большого круга разнородных задач, некоторые из которых могут требовать определенных особенностей архитектуры микропроцессорного ядра для наиболее эффективного их выполнения. Однако, в большинстве случаев для решения широкого ряда разнородных задач применяются микропроцессорные ядра общего назначения, не специализированные под решение каких-либо конкретных задач.

Большинство современных микросхем FPGA поддерживают функцию, называемую частичной реконфигурацией. Частичная реконфигурация позволяет изменить отдельную часть реализуемой на FPGA логической схемы во время работы, при этом не затрагивая другие, статические части.

В данной работе описывается разработанная архитектура синтезируемого микропроцессора, способного анализировать предстоящий к исполнению код и, используя функцию частичной реконфигурации, динамически изменять отдельные части своей микроархитектуры для обеспечения наиболее благоприятных условий для решения конкретной задачи и, тем самым, увеличения производительности. Данная разработка названа микропроцессорным ядром с функцией частичной реконфигурации, либо реконфигурируемым микропроцессором.

Частичная реконфигурация FPGA

Способность FPGA к реконфигурации – изменению реализуемой ресурсами кристалла FPGA логической схемы – является их определяющим свойством. При полной реконфигурации происходит сброс и реконфигурация всех имеющихся на кристалле ресурсов. Под функцией частичной реконфигурации понимается возможность изменения отдельной части реализуемой на FPGA логической схемы, в то время как остальная, статическая её часть остается неизменной и продолжает работать. Функция частичной реконфигурации предоставляет новые возможности при разработке на FPGA и имеет широкий потенциал применения, от реализации принципиально новых систем до оптимизации стандартных.

На данный момент самыми известными и распространенными производителями FPGA являются Xilinx (принадлежит AMD) и Altera (принадлежит Intel). Возможность частичной реконфигурации реализована в FPGA обоих производителей. Рассматриваемая далее информация является общей для FPGA Xilinx и Altera.

Текущая конфигурация FPGA определяется данными, содержащимися в энергозависимой памяти, называемой конфигурационной. С помощью отдельных битов задаются реализуемые LUT функции, соединения между логическими блоками на кристалле и т.д. Образующими элементами конфигурационной памяти являются ячейки SRAM. Генерируемые используемой средой разработки битстримы (файлы для конфигурации FPGA) содержат в себе информацию для записи в конфигурационную память, необходимую для реализации на кристалле заданной логической схемы (дизайна). В процессе полной реконфигурации FPGA перед записью в конфигурационную память новой информации происходит её полный сброс, после чего новыми данными задается другой дизайн. Частичная реконфигурация же позволяет осуществить сброс и перезапись только определенной части конфигурационной памяти, задающей текущую конфигурацию лишь отдельного, заранее заданного участка кристалла FPGA. Как и полная реконфигурация, частичная реконфигурация производится с помощью битстримов. Частичные битстримы (предназначенные для частичной реконфигурации), в отличие от полных, содержат только данные для программирования необходимой области кристалла.

При этом реконфигурируемых областей может быть задано несколько, и каждой из них будет соответствовать свой набор частичных битстримов, каждый из которых будет конфигурировать этот участок одной из заданных пользователем логических схем, доступных для реализации на данной реконфигурируемой области. Общая схема принципа работы частичной реконфигурации представлена на рисунке 1.

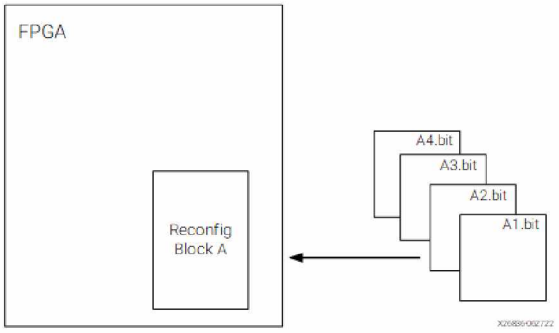

Рисунок 1 – Иллюстрация принципа работы частичной реконфигурации из официальной документации Xilinx [1]

Для произведения частичной реконфигурации FPGA требуется доставить нужный частичный битстрим на специализированный контроллер, имеющей доступ к записи и чтению конфигурационной памяти и отвечающий за контроль над процессом реконфигурации. При этом изначальное место хранения битстрима в большинстве случаев не имеет значения. Он может быть получен по некоторому интерфейсу с ПК, вычитан из имеющейся на плате с FPGA внешней памяти и т.д. Доступ к вышеуказанному контроллеру осуществляется с помощью специализированных портов. И в устройствах Xilinx, и в устройствах Altera есть возможность доставки битстрима с использованием внешних портов (например, JTAG), и с использованием внутренних (например, ICAP в Xilinx), доступных из пользовательской логики. Использование внутренних портов позволяет достичь полностью автономной частичной реконфигурации, не требующей внешнего по отношению к FPGA устройства для доставки битстрима или принятия решения о реконфигурации [1].

Реконфигурируемые вычислители

Описываемая в данной работе архитектура микропроцессора попадает в класс архитектур, называемых реконфигурируемыми вычислителями (Reconfigurable Computing). Под реконфигурируемыми вычислителями обычно понимаются такие вычислительные системы, которые включают в себя элементы реконфигурируемой логики и благодаря этому имеют возможность в какой-то степени изменять свою внутреннюю структуру. Реконфигурируемые вычислители, представляющие собой микропроцессорное ядро, включающее в себя реконфигурируемую логику либо взаимодействующее с ней, называют реконфигурируемыми процессорами. Существующие реконфигурируемые процессоры можно классифицировать на основе ряда признаков. Например, насколько тесно связаны микропроцессорное ядро и реконфигурируемая логика, происходит ли реконфигурация во время работы процессора (динамически реконфигурируемые), либо для реконфигурации процессор необходимо остановить (статически реконфигурируемые) и т.д. Представленную в данной статье разработку можно отнести к динамически реконфигурируемым процессорам с включенной в тракт данных реконфигурируемой логикой. Помимо этого, важным различием между реконфигурируемыми процессорами является источник сигнала о начале реконфигурации, т.е. что является триггером для начала реконфигурации системы. В большинстве существующих реконфигурируемых архитектур используется подход, при котором вышеуказанный триггер, а также другая информация о требуемой реконфигурации, входят в состав исполняемого кода. Другим подходом, используемым в том числе в представленной в данной работе архитектуре, является анализ исполняемого кода во время работы процессора и использование результатов анализа для принятия решений, связанных с реконфигурацией [2, 3, 4].

Обзор аналогов

Для проведения разработки был рассмотрен ряд существующих аналогов. Ниже приведено краткое описание нескольких из них, наиболее схожих с разработанной архитектурой.

Архитектура Warp, предложенная в [5], предназначена для исполнения стандартного бинарного кода своей архитектуры набора команд (ISA – Instruction Set Architecture) и не требует особого компилятора либо предварительной обработки ассемблерного кода для работы. Во время исполнения программы отдельный функциональный модуль (называемый Profiler) анализирует код и выделяет определенные критические участки кода, которые могут быть заменены логической схемой. В архитектуре присутствует аппаратный модуль CAD, синтезирующий логические схемы на основе найденных модулем Profiler участков кода и генерирующий битстримы в процессе работы программы.

После этого реконфигурируемая область программируется сгенерированным модулем, а вместо критического участка кода вставляется команда перехода к функции, обеспечивающей произведение вычислений на аппаратном функциональном модуле. Стоит отметить, что в Warp-процессорах используется специально разработанная структура FPGA, называемая W-FPGA.

Архитектура rMIPS [6] реализована на FPGA Xilinx как синтезируемый процессор и использует внутренний порт ICAP для частичной реконфигурации. Для работы с данной архитектурой используется специальная библиотека, содержащая в себе набор арифметических функций (например, вычисления синуса), реализованных программно, а также соответствующие им функциональные модули, с помощью которых те же вычисления могут быть проведены аппаратно (в виде их исходного HDL-кода либо готовых частичных битстримов). В составе rMIPS имеется определенное число «слотов», в которые могут быть помещены вышеуказанные реконфигурируемые функциональные модули, называемые RFU (Reconfigurable Functional Unit). Процессом реконфигурации управляет так называемая система DPR (Dynamic Partial Reconfiguration System). По умолчанию вызовы в коде функций из описанной выше библиотеки всегда указывают на их программную реализацию. Система DPR мониторит исполняемый на данный момент код, и при обнаружении вызова одной из библиотечных функций проверяет, имеется ли сейчас ее аппаратная реализация в одном из «слотов». Если да, система DPR останавливает начавшееся выполнение программной реализации функции и вместо этого передает управление версии функции, использующей аппаратный функциональный модуль. Решение о конфигурации «слотов» тем или иным функциональным модулем, либо о замене одного функционального модуля на другой, принимается с помощью алгоритма, учитывающего частоту использования функций и число тактов, затрачиваемое на выполнение функций в программной и аппаратной реализации.

В концептуальной архитектуре RISPP (Rotating Instruction Set Processing Platform) [7] применен уникальный подход к имплементации аппаратных функциональных модулей, призванных заменить участки кода для ускорения вычислений. Для каждой программы определяется набор специальных инструкций (SI – Special Instruction), каждая из которых должна выполнять некоторую вычислительную операцию. При исполнении такой инструкции процессором необходимые вычисления могут быть произведены либо программно, либо с использованием соответствующего аппаратного функционального модуля. Уникальность архитектуры заключается в том, что каждый из доступных функциональных модулей состоит из набора более простых модулей, называемых атомами. В данной архитектуре именно атомы являются реконфигурируемыми единицами. В тракт данных процессора включена «атомная инфраструктура», представляющая собой набор «слотов», в каждый из которых может быть помещен один атом. Каждый атом предназначен для выполнения одной, относительно простой вычислительной операции. Такой подход позволяет обеспечить совместное использование ресурсов (отдельных атомов) различными функциональными модулями, тем самым обеспечивая эффективное их использование. Функциональные модули также могут иметь несколько различных реализаций (каждая из которых называется молекулой), содержащих разное кол-во атомов и различающихся по производительности. В код программы также добавляются так называемые инструкции предсказания (FI – Forecast Instructions), сообщающие о необходимости в скором времени исполнения SI, однако окончательное решение о времени произведения реконфигурации и выборе добавляемых/заменяемых атомов принимается отдельным функциональным модулем во время работы программы на основе ряда факторов.

Архитектура реконфигурируемого микропроцессора

Структурная схема разработки приведена на рисунке 2. Реконфигурируемый микропроцессор предполагается реализовать на FPGA фирмы Xilinx.

Разработка состоит из собственно реконфигурируемого микропроцессорного ядра, включающего в себя N реконфигурируемых слотов, и блока управления процессом реконфигурации (далее – блок управления), предназначенного для управления процессом реконфигурации и принятия всех связанных с реконфигурацией решений. К реконфигурируемому микропроцессору подключена память программы, в данном случае реализованная на FPGA, и внешняя память, содержащая частичные битстримы.

В каждый из реконфигурируемых слотов в составе ядра может быть помещен один из M доступных для этого слота реконфигурируемых модулей. На рисунке 2 частичные битстримы, содержащиеся во внешней памяти, схематично обозначены как rm<№ слота>-<№ модуля>.bit. Например, реконфигурируемый модуль №1 для слота №2 обозначен как rm2-1.bit. Каждый реконфигурируемый модуль представляет собой относительно простую логическую схему, реализующую некоторую вычислительную операцию. Реконфигурируемые слоты в составе микропроцессорного ядра соединены последовательно, т.е. результат обработки данных одним реконфигурируемым модулем может быть передан на вход другого. Такой подход к реконфигурации схож с подходом, примененным в концептуальной архитектуре RISPP, описанной в [7].

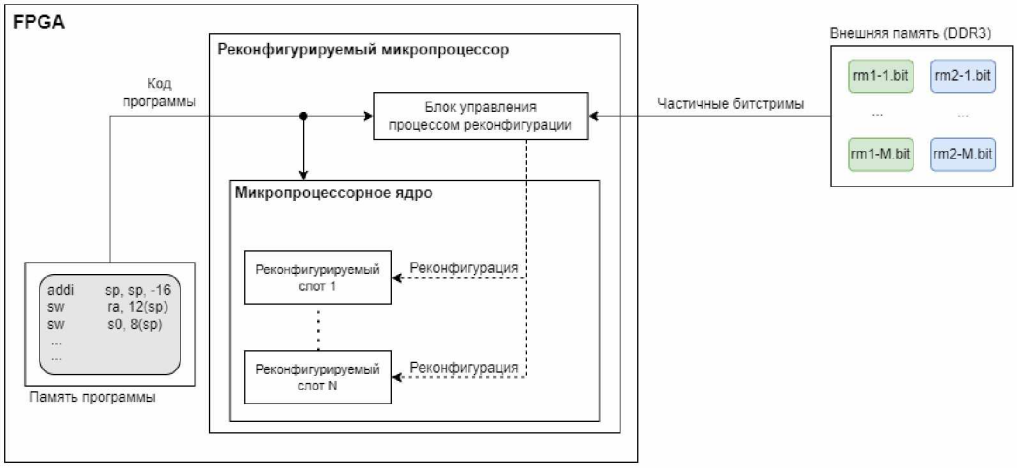

Рисунок 2 – Структурная схема реконфигурируемого микропроцессора

После запуска микропроцессорного ядра начинается исполнение кода программы традиционным способом. Тем временем, блок управления анализирует код программы. При анализе кода блок управления ищет последовательности инструкций, функционал которых может быть заменен логической схемой, составленной из расположенных в определенной последовательности имеющихся реконфигурируемых модулей. При нахождении блоком управления такой последовательности, принимается решение о проведении реконфигурации. В реконфигурируемые слоты помещаются необходимые модули, а в коде программы блок инструкций, функционал которых будет реализован аппаратно, заменяется специальной командой, передающей управление реконфигурируемым модулям. За вычитку необходимых частичных битстримов из памяти и подачу их на конфигурационный порт также отвечает блок управления. После проведения необходимых вычислений, управление передается обратно коду программы.

После запуска микропроцессора блок управления начинает анализ кода с K-й инструкции. Это необходимо для того, чтобы после принятия блоком управления решения о реконфигурации оставалось достаточно времени для ее осуществления до достижения данного участка кода микропроцессорным ядром. Соответственно, число K определяется на основе времени, затрачиваемого на реконфигурацию, и может различаться в зависимости от используемой микросхемы FPGA, размера реконфигурируемых модулей и т.д.

Предполагается реализация микропроцессорного ядра с использованием нескольких независимых групп последовательно соединенных реконфигурируемых слотов. Таким образом, если будут обнаружены несколько пригодных для аппаратной реализации блоков инструкций, размещенных в коде программы недалеко друг от друга, оба блока могут быть заменены аппаратным эквивалентом с использованием двух групп реконфигурируемых слотов. При использовании одной группы, функционал только одного из обнаруженных блоков инструкций можно было бы реализовать аппаратно, так как времени между окончанием исполнения одного блока и началом исполнения следующего было бы недостаточно для проведения реконфигурации.

Заключение

Реконфигурируемые микропроцессоры эффективно совмещают гибкость программного обеспечения и производительность аппаратного, и могут благодаря этому обеспечить более высокую скорость исполнения программ по сравнению со статическими аналогами. В то же время с ними связаны определенные проблемы, в частности, сложность разработки программного обеспечения. Одной из основных особенностей разработанной архитектуры является именно возможность исполнения стандартного ассемблерного кода, что обеспечивает переносимость программ и простоту работы с устройством.

Список использованных источников:

1. AMD. Vivado Design Suite User Guide: Dynamic Function eXchange [Электронный ресурс]. – Режим доступа: https://docs.amd.com/r/en-US/ug909-vivado-partial-reconfiguration/Introduction-to-Dynamic-Function-eXchange. – Дата доступа: 20.03.2024.

2. Ingredients of Adaptability: A Survey of Reconfigurable Processors / A. Chattopadhyay // VLSI Design, 2013.

3. Architecture, challenges and applications of dynamic reconfigurable computing / Y. N. Lu [et al.] // Journal of Semiconductors, 2020.

4. An Introduction to Reconfigurable Computing / K. Compton, S. Hauck // Northwestern University, Dept. of ECE Technical Report, 1999.

5. Warp Processors / G. Stitt // ACM Transactions on Design Automation of Electronic Systems, 2006.

6. Dynamic self-reconfiguration of a MIPS-based soft-core processor architecture / S. Nolting // Journal of Parallel Distributed Computing, 2017.

7. Run-time Adaptation for Reconfigurable Embedded Processors / L. Bauer, J. Henkel – Springer, 2011.

UDC 004.274

PARTIALLY RECONFIGURABLE MICROPROCESSOR CORE ON FPGA

Bosko N.V.

Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus

Lytsik Y.A. – PhD in Technical Sciences

Annotation. This paper describes an architecture of a soft-core microprocessor that uses the FPGA partial reconfiguration function to alter parts of its microarchitecture in order to ensure the most favorable conditions for solving the current task, thus improving performance.

Keywords. FPGA, partial reconfiguration, soft-core microprocessor, reconfigurable computing.