

Разработка и отладка модели микропроцессора архитектуры MIPS и ее реализация на программируемых логических интегральных схемах (ПЛИС)

А.А. Царинжапов, А.В. Кошевенко

Дальневосточный федеральный университет, г. Владивосток

Аннотация: В работе рассматривается разработка и отладка модели конвейерного микропроцессора архитектуры MIPS (Microprocessor without Interlocked Pipeline Stages), написанной на языке описания аппаратуры VHDL (VHSIC (Very high speed integrated circuits) Hardware Description Language). Важным этапом разработки является моделирование временной диаграммы работы микропроцессора при помощи программы ModelSim с выполнением отладочных тестов. Для доказательства адекватности представленной модели производится синтез разработанного ядра микропроцессора MIPS на программируемую логическую интегральную схему (ПЛИС) с последующим выполнением программ, записанных во внешнюю память устройства.

Ключевые слова: микропроцессор, тест, программируемая логическая интегральная схема, цифровое устройство, верификация, модель, язык описания аппаратуры, field-programmable gate array (FPGA), конвейер, MIPS.

Введение

Современное развитие вычислительных технологий требует создания новых цифровых устройств, способных решать поставленные задачи с минимальным временем их решения и с возможностью применения универсального подхода для проектирования таких устройств. Проектирование подобных устройств целесообразно основывать на использовании реконфигурируемой логики и средств автоматизированного проектирования. Ключевой особенностью таких систем является возможность создания любых цифровых устройств используя аппаратную и программную составляющую, а также средства отладки, с помощью которых можно выявить функциональные ошибки синтезированного устройства перед синтезом на кристалл ПЛИС (Программируемая Логическая Интегральная Схема). Микроэлектроника бурно развивается, а приход новой элементной базы в виде FPGA (Field-Programmable Gate Array) открывает все больше возможностей в проектировании микропроцессорных ядер и цифровых устройств. Преимущества использования микросхем программируемой



логики очевидны, они позволяют выполнить проектирование и отладку устройства перед его запуском в серийное производство (т.е. реализацию в ASIC - Application-Specific Integrated Circuit - интегральная схема, специализированная для решения конкретной задачи), а также использоваться в качестве интегрированной элементной базы в современных вычислительных модулях в виде микропроцессорной системы или реконфигурируемой вычислительной структуры, которая применяется в построении высокопроизводительных вычислительных устройств включая супер ЭВМ. Кристаллы ПЛИС обладают высокой производительностью, компактностью, энергоэффективностью за счет чего используются не только при проектировании и создании высокопроизводительных систем, но также в малогабаритных устройствах для промышленного, бытового и военного назначения. Возможность изменения структуры логических ячеек позволяет сконфигурировать их под решения конкретно заданных задач, на базе которых можно строить микропроцессоры с разнообразной архитектурой и микроархитектурой [1]. Проектирование микропроцессоров на основе интегральных схем с программируемой логикой является важным этапом в прототипировании будущих процессоров перед их запуском в серийное производство. Под этим этапом подразумевается отладка будущей модели процессора и его функциональных блоков на кристалле ПЛИС. В случае пренебрежении данным этапом - цена ошибки будет очень велика, так как невозможность внести исправления в готовый кристалл процессора на аппаратном уровне приведет к финансовым потерям или неработающей, небезопасной системе. Примером служат аппаратные уязвимости, обнаруженные в архитектурах и микроархитектурах процессоров ARM, Intel, AMD и других уже после их изготовления и ввода в эксплуатацию. Именно тестирование в процессе проектирования на отладочных платах с программируемой логикой позволяет обнаружить недостатки, которые

невозможно выявить в обычных интегральных схемах, на которых реализуется серийный образец процессора или другое сложное цифровое устройство. FPGA прототипирование используют в той или иной степени как отечественные, так и зарубежные разработчики: МЦСТ, Intel, IBM и другие.

Основные проблемы проектирования и диагностирования микропроцессорных систем, а также их последующей верификации — это использование готовых сложных функциональных блоков (IP-ядра (Intellectual Property), СФ-блоки (сложные функциональные) — готовые блоки для проектирования микросхем (например, для построения систем на кристалле)), разработанных для разнообразных систем, которые невозможно диагностировать по отдельности. После интеграции СФ блока в общую схему спроектированного устройства проводятся системные тесты для проверки, как этот блок работает в составе системы в целом. Все тесты проходят в циклическом регрессионном порядке. При проектировании с нуля функциональных блоков микропроцессора проводятся как отдельные тесты каждого блока, так и совместные в составе всей системы.

Соответственно, проектирование микропроцессора для последующего синтеза на FPGA является сложной задачей, требующей знания архитектуры и микроархитектуры будущего устройства, понимания различных языков программирования аппаратного и программного уровней, организации микропроцессорных блоков, умения разработать и провести тестирование с последующей верификацией устройства, умения работать в симулированной среде, знания схемотехники и принципов работы с периферией и средствами автоматизированного проектирования [2].

Актуальность данного исследования вызвана необходимостью реализации современных микропроцессорных ядер на базе ПЛИС с последующей разработкой тестов для их отладки. Для этого требуется разработать эффективные методы тестирования и прототипирования

микропроцессорных ядер с последующей их верификацией для использования в электронных устройствах. В рамках исследования используется предоставленная компанией Imagination Technology FPGA Cyclone V 5CEBA4F23C7N, на которой реализуется проектирование и синтез модели микропроцессора MIPS – архитектуры и ее верификация.

1 Процесс разработки модели микропроцессора

Для демонстрации процесса проектирования и отладки модели микропроцессора используется доработанная структурная схема учебной модели, предложенная в [3], и представленная на рис.1. На базе нее написана модель конвейерного микропроцессора на языке описания аппаратуры VHDL, система команд была переработана и расширена до 42 инструкций.

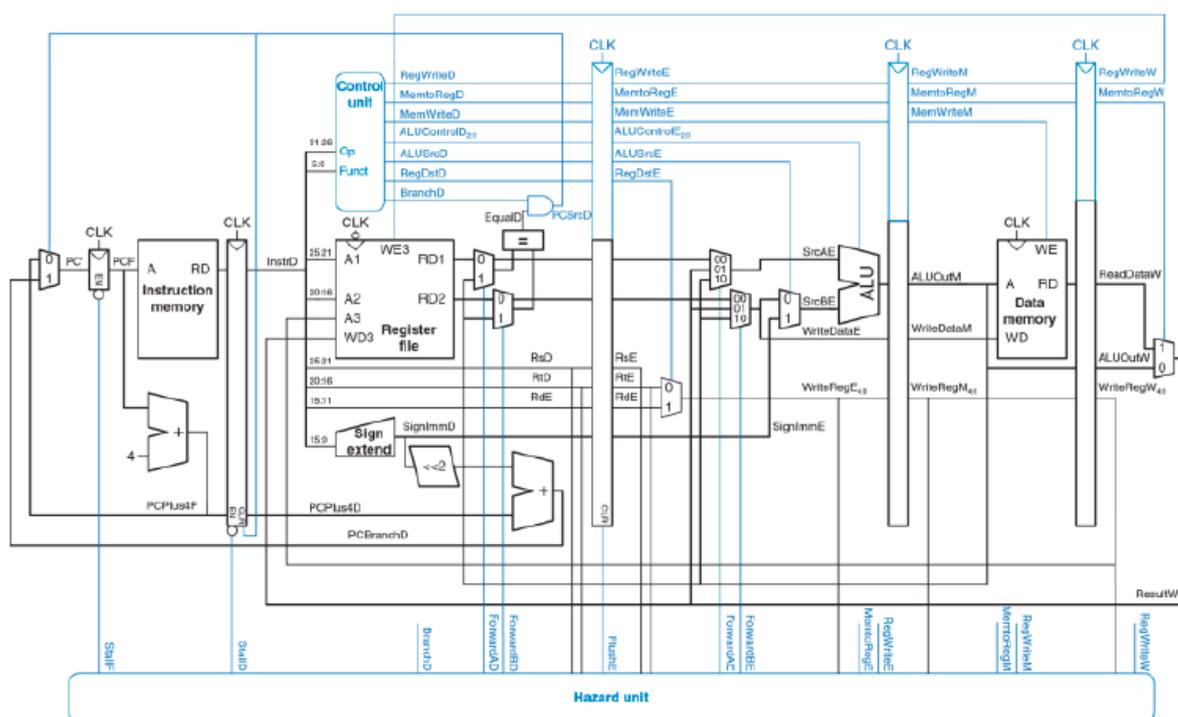


Рис.1 - 5-ступенчатый конвейерный процессор с устранением конфликтов

Перед тем как начать разработку конвейерного микропроцессора MIPS на языке описания аппаратуры VHDL, было принято решение кодовую часть разделить на стадии, а также продемонстрировать наглядно разбиение на стадии при моделировании на уровне регистровых передач - Register Type Level (RTL) дизайна схемы и при проведении тестирования в программе

Modelsim. Эти решения скажутся на интерпретации модели, а также позволят структурировать код. Для проектирования микропроцессора использовался САПР Quartus Prime 15.1 со встроенным симулятором Modelsim-Altera. RTL-дизайн схемы модели конвейерного процессора представлен на рис.2.



Рис.2 - RTL-дизайн модели конвейерного микропроцессора MIPS

После создания описания проекта уровня RTL требуется произвести его функциональное тестирование, т.е. удостовериться, что все заявленные функции в проектной спецификации реализованы и работают корректно. Для осуществления тестирования проекта требуется описать тестовое окружение (Test Environment), представляющее собой в самом простейшем случае модуль более высокого уровня абстракции, включающий в себя экземпляр тестируемого проекта и реализующий определенные сценарии тестирования. Основная задача тестового окружения — подавать на входные сигналы тестируемого проекта тестовые комбинации и анализировать выходные сигналы на корректность работы тестируемого проекта [4,5].

2 Отладка модели

2.1 Отладка компонентов модели микропроцессора

При отладке модели микропроцессор рассматривается как устройство, состоящее из нескольких блоков, описывающих его функциональные особенности (передачу, хранение, обработку данных и управление), связанные с архитектурой системы команд. Тестирование происходит по отдельности каждой функции. Тестирование микропроцессора предполагает отладку всех функциональных блоков. В качестве среды отладки модели используется программа ModelSim Altera Started Edition 10.4b. Рассмотрим тестирование отдельных компонентов системы на примере написания TestBench для конкретного компонента. Компонентом для тестирования в качестве примера используется контроллер управления арифметико-логическим устройством (АЛУ), входящий в блок Execute (выполнение). Контроллер управления АЛУ указывает, какую арифметическую операцию выполняет АЛУ (ALUOp_out). Контроллер управления АЛУ содержит восемь входных комбинаций (3 входных управляющих сигнала - ALUOp_in) и пять комбинаций, используемых для выбора операции – func 5:0 [8,9].

Выбираем компонент ALU_CONTROL инициализируем сигналы каждые 20ps. Данные подаются для определения операции АЛУ, включая несуществующие операции (в таком случае выполнение теста завершается, при нахождении неверной операции).

Первая операция ALUOp_in = 010, funct = 100100, что соответствует инструкции R - типа, на выходе ALUOp_out = 0000, действительно данная операция соответствует команде AND. На рис.3 представлена симуляция работы данного компонента при помощи разработанного теста.

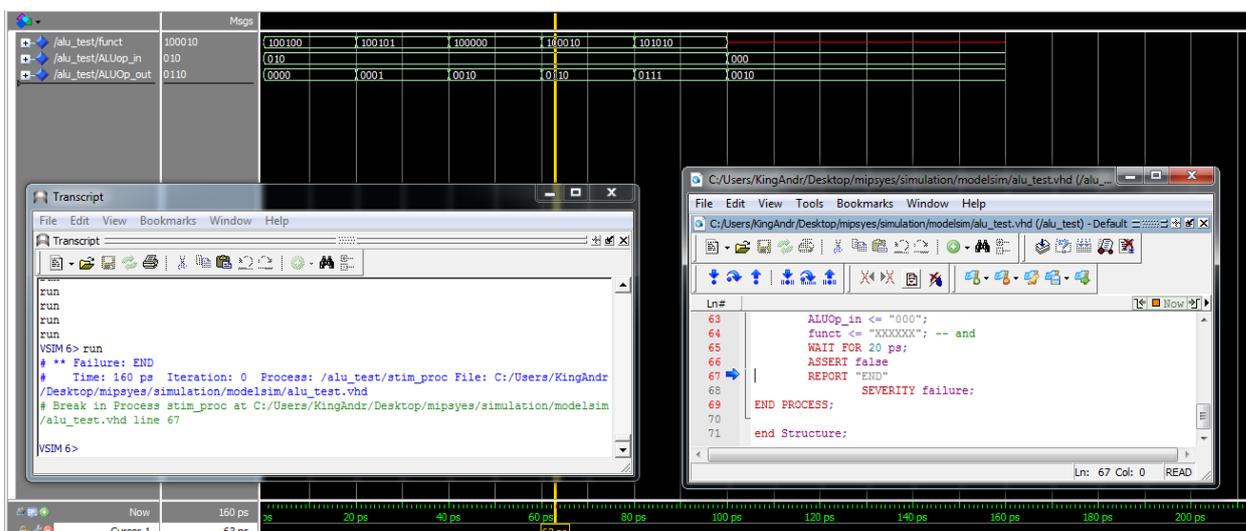


Рис.3 – Выполнение теста для контроллера управления АЛУ в ModelSim

2.1 Комплексная отладка модели микропроцессора

В настоящее время все чаще термин «функциональное тестирование» заменяют на термин «функциональная верификация», подчеркивая тем самым резкое усложнение этого процесса в связи с не прекращающимся ростом сложности проектируемых устройств. Также, часто термин «функциональная верификация» относят к более сложному процессу тестирования, в котором применяются современные техники и фреймворки (программное обеспечение, облегчающее разработку и объединение разных компонентов большого программного проекта) для создания сложных тестовых окружений, а «функциональным тестированием» считают относительно простую проверку проекта на корректность работы [5,10].

Для проверки работоспособности модели прототипа ядра микропроцессора MIPS в программе симуляции ModelSim отладим программу, написанную на языке ассемблера и переведенную в машинный код в формате HEX, которая выполняет вычисление n-го числа Фибоначчи. Переведенные ассемблерные инструкции в шестнадцатеричный формат загружаются в память ROM. Зададим число Фибоначчи $n = 35$. Данное число будет равно 9227465. Переведем десятичное число в шестнадцатеричный

формат и получим 8CCCC9h, что соответствует решению, представленному на временной диаграмме Modelsim, рис.4.

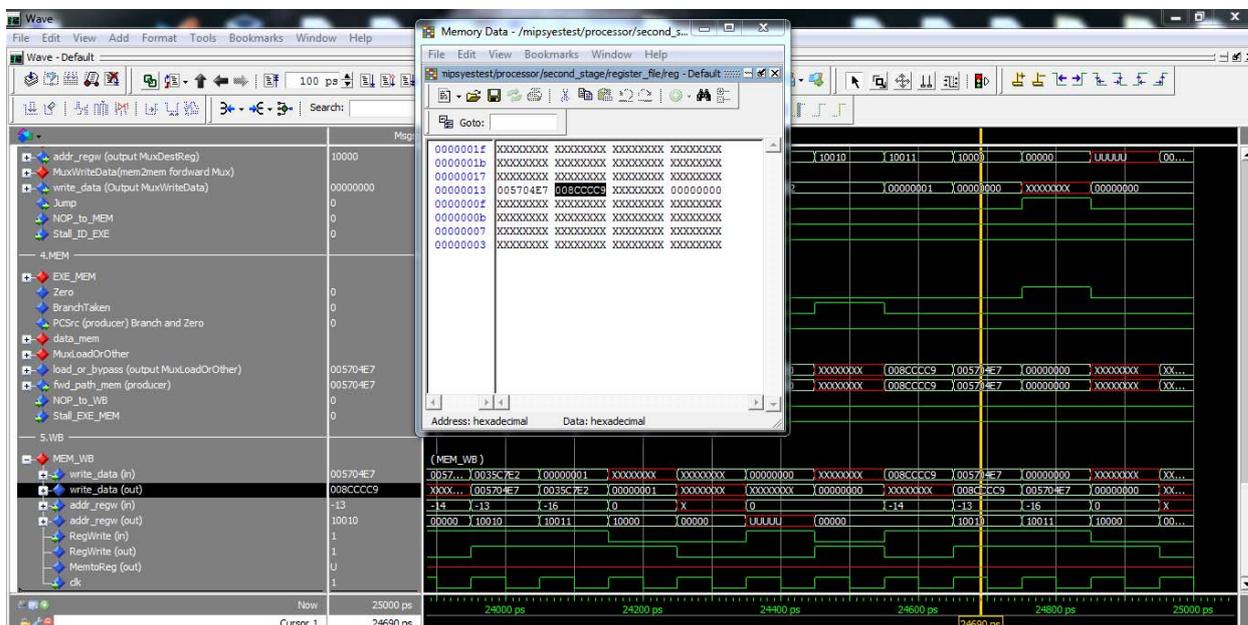


Рис.4 – Выполнение тестовой программы и вывод результата в ModelSim

3 Синтез отлаженной модели MIPS-микропроцессора на FPGA

Для синтеза используется отладочная плата DE0-CV на основе FPGA Cyclone V 5CEBA4F23C7N.

После компиляции проекта в окне Flow Summary можно посмотреть использованные ресурсы кристалла FPGA Cyclone V 5CEBA4F23C7N. В программе Altera Quartus 15.1 для визуализации расположения и использования элементарных блоков в общей архитектуре целевой ПЛИС применяется утилита Chip Planner. На карте ресурсов рис.5 показаны ресурсы ПЛИС: LE (логический элемент), объединенные в LAB, аппаратные множители (DSP block), ячейки памяти, буферы ввода вывода, генераторы PLL. Цветом показана степень использования: светлым – неиспользуемые блоки, темным – максимально загруженные. Если увеличить масштаб, то доступна детализация на уровне LE. [6,7]

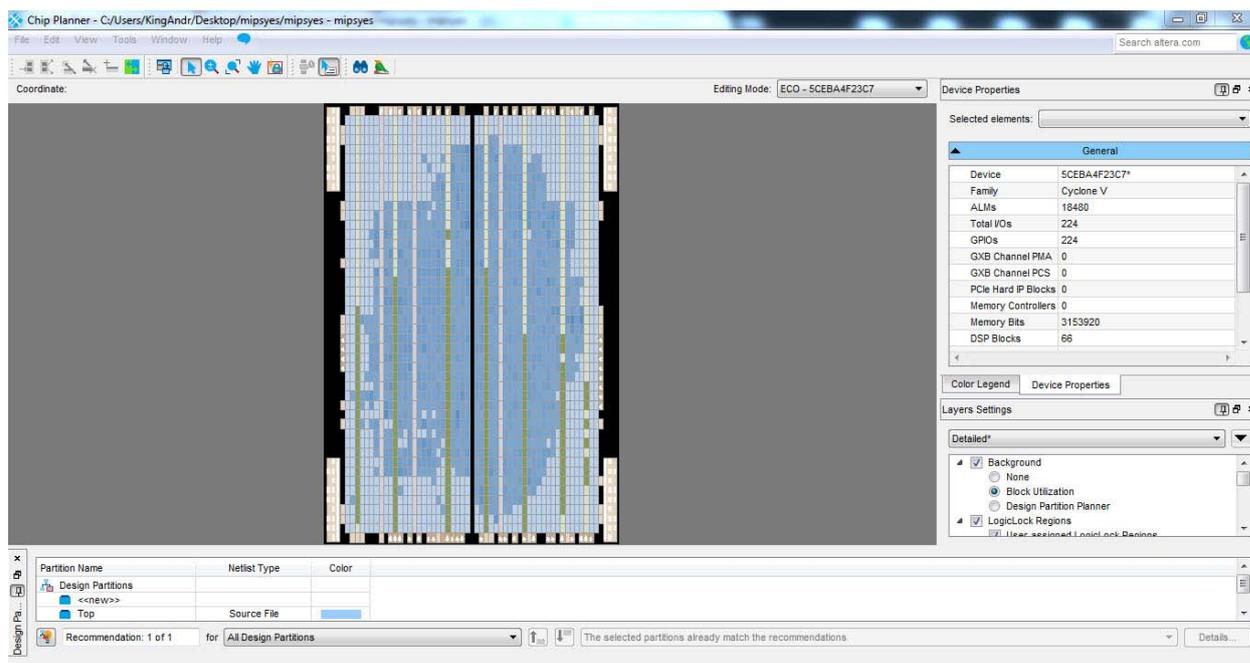


Рис. 5 – Карта ресурсов FPGA Cyclone V 5CEBA4F23C7N

Испытательный стенд написан для тестирования MIPS-процессора и демонстрации моделирования выполнения программы перед программированием FPGA. После отладки в программе для симуляции ModelSim синтезируем (запрограммируем в конфигурационную память) выходной результат после компиляции mipsyes.sof на FPGA Cyclone V 5CEBA4F23C7N в программе Altera Quartus 15.1, на рис.6 показан результат выполнения программы вычисление n-го числа Фибоначчи на отладочной плате DE0-CV.

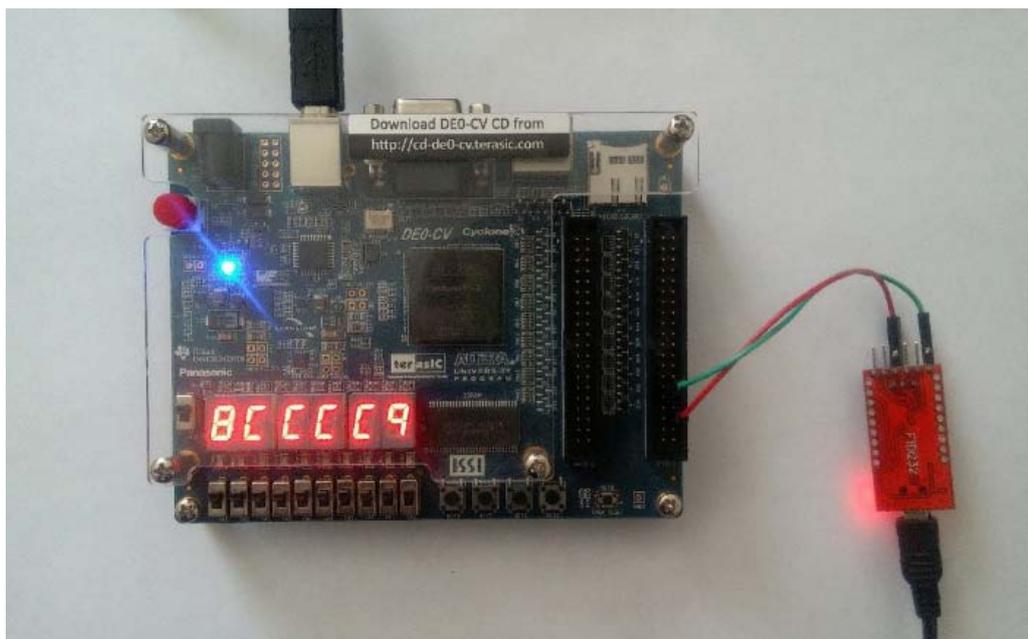


Рис.6 – Результат выполнения программы на отладочной плате DE0-CV

Заключение

Использованный в работе подход отладки функциональных блоков микропроцессора упрощает тестирование модели и делает данную методику применимой, а также доступной для большинства реконфигурируемых цифровых устройств путем подтверждения адекватности полученных результатов на временной диаграмме ModelSim. За счет использования средств отладки возможно предотвратить ряд ошибок, которые возникают в процессе проектирования цифровых устройств, а также подтвердить или опровергнуть адекватность работы представленной симулированной модели, которая в дальнейшем программируется на кристалл ПЛИС. В качестве развития конвейерной микроархитектуры, которая в данной работе является предметом исследования, возможна реализация дополнительных команд, теневого регистра, добавление сопроцессора, удлинение конвейера и ряд других дополнений, которые позитивно скажутся на развитии разработанного ядра архитектуры MIPS для последующего использования в цифровой аппаратуре. FPGA за счет реконфигурируемой структуры логических ячеек позволяет создавать и синтезировать на кристалл

собственные архитектурные разработки, а также описания архитектуры микропроцессоров, которые необходимо реализовать без применения дорогостоящих универсальных технологий. Прототипирование микропроцессоров на ПЛИС является хорошим способом для отладки, выявления недостатков и преимуществ разработанных технологий с возможностью внесения изменений за счет реконфигурируемой логики.

Литература

1. С.В. Юрлин, И.Н. Бычков. Прототипирование на основе ПЛИС для верификации многоядерных микропроцессоров. МЭС-2014. Россия, Москва, октябрь 2014. ИППМ РАН. URL: mes-conference.ru/data/year2014/pdf/D070.pdf.
2. Ф.К. Будылин, И.А. Полищук, М.В. Слесарев, С.В. Юрлин. Опыт прототипирования микропроцессоров компании ЗАО «МЦСТ». Вопросы радиоэлектроники серия ЭВТ, 2012, выпуск 3. URL: mcst.ru/doc/120412/budilin.doc.
3. Харрис Д.М, Харрис С.Л. Цифровая схемотехника и архитектура компьютера: учеб. пособие - 2-е изд., перераб, и доп. - USA: Morgan Kaufman, 2013. - 1625 с.: ил.
4. Алексей Рабоволук. Обзор маршрута проектирования ПЛИС FPGA Advantage компании Mentor Graphics. Часть 4. Моделирование в среде FPGA Advantage // Компоненты и технологии, 2006, №4, URL: kit-e.ru/assets/files/pdf/2006_04_86.pdf.
5. Литвинов Е.И, Шагурин И.И. Лабораторный практикум. Проектирование цифровых устройств с использованием языка Verilog HDL на базе FPGA фирмы Xilinx. Москва, 2012. - 173 с.
6. Vaibbhav Taraate. PLD Based Design with VHDL RTL Design, Synthesis and Implementation - Springer Nature Singapore Pte Ltd, 2017. – 423 p.

7. Bogdan Belean. Application-Specific Hardware Architecture Design with VHDL. - Springer International Publishing, 2018. – 191 p.
8. Амосов В.В. Схемотехника и средства проектирования цифровых устройств. – СПб.: БХВ-Петербург, 2007. - 560 с.: ил.
9. П.Н. Бибило. Синтез логических схем с использованием языка VHDL- М.: СОЛОН - Р, 2009. - 384 с.: ил
10. David A. Patterson, John L. Hennessy. Computer organization and design, 5th Edition. The hardware/software interface - The Morgan Kaufmann Series in Computer Architecture and Design, 2014- 1024 p.
11. Д.В. Быков, А.Д. Неретин. Прогнозирование производительности при реализации алгоритмов генерации случайных последовательностей больших размерностей на реконфигурируемых архитектурах с сопроцессорами // Инженерный вестник Дона, 2014, №2, URL: ivdon.ru/ru/magazine/archive/n2y2014/2414.
12. С.В. Волобуев, А.П. Евдокимов, В.Г. Рябцев. Применение аппаратных и программных средств для изменения режимов работы цифровых устройств, реализованных на ПЛИС // Инженерный вестник Дона, 2017, №4, URL: ivdon.ru/ru/magazine/archive/n4y2017/4478.

References

1. Yurlin S.V, Bychkov I.N. MES, Russia, Moscow, october 2014. URL: mes-conference.ru/data/year2014/pdf/D070.pdf.
 2. F.K. Budylin, I.A. Polishchuk, M.V. Slesarev, S.V. Yurlin. Voprosy radioelektroniki EVT, 2012, №3. URL: mcst.ru/doc/120412/budilin.doc.
 3. Kharris D.M., Kharris S.L. Tsifrovaya skhemotekhnika i arkhitektura komp'yutera [Digital Design and computer architecture]. ucheb. posobie 2-e izd., pererab, i dop. USA Morgan Kaufman, 2013. 1625 p.
 4. Aleksey Rabovolyuk. Komponenty i tekhnologii, 2006, №4. URL: kit-e.ru/assets/files/pdf/2006_04_86.pdf.
-

5. Litvinov E.I., Shagurin I.I. Laboratornyy praktikum. Proektirovanie tsifrovyykh ustroystv s ispol'zovaniem yazyka Verilog HDL na baze FPGA firmy Xilinx [Design of digital devices with use of the Verilog HDL language on the basis of FPGA of company Xilinx]. Moskva, 2012. 173p.
6. Vaibbhav Taraate. PLD Based Design with VHDL RTL Design, Synthesis and Implementation. Springer Nature Singapore Pte Ltd, 2017. 423 p.
7. Bogdan Belean. Application-Specific Hardware Architecture Design with VHDL. Springer International Publishing, 2018. 191 p.
8. Amosov.V.V. Skhemotekhnika i sredstva proektirovaniya tsifrovyykh ustroystv [Circuitry and design tools of digital devices]. SPb. BKhV Peterburg, 2007. 560 p.
9. P.N. Bibilo. Sintez logicheskikh skhem s ispol'zovaniem yazyka VHDL [Synthesis of logical schemes with use of the VHDL language]. M.SOLON R, 2009. 384 p.
10. David A. Patterson, John L. Hennessy. Computer organization and design, 5th Edition. The hardware/software interface. The Morgan Kaufmann Series in Computer Architecture and Design, 2014. 1024 p.
11. D.V. Bykov, A.D. Neretin. Inzhenernyj vestnik Dona (Rus), 2014, №2. URL: ivdon.ru/ru/magazine/archive/n2y2014/2414.
12. S.V. Volobuev, A.P. Evdokimov, V.G. Ryabtsev. Inzhenernyj vestnik Dona (Rus), 2017, №4. URL: ivdon.ru/ru/magazine/archive/n4y2017/4478.