

УДК 62.50

**РАЗВИТИЕ КОМПЕТЕНЦИИ ПО ДИСЦИПЛИНЕ
«МИКРОПРОЦЕССОРНАЯ ТЕХНИКА» НА БАЗЕ
ПЛИС ФИРМЫ ALTERA СЕРИИ CYCLONE III
DEVELOPMENT OF COMPETENCE FOR THE
DISCIPLINE «MICROPROCESSOR TECHNICS»
BASED ON FPGA ALTERA COMPANY CYCLONE
III SERIES.**

Бородин Михаил Юрьевич, кандидат технических наук, доцент кафедры «Электропривод и автоматизация промышленных установок», Уральский федеральный университет имени первого Президента России Б.Н. Ельцина, Россия, 620002, г. Екатеринбург, ул. Мира, 19. E-mail: bmu@k66.ru, Тел.: +7(922)223-66-54

Кондаков Константин Андреевич, студент кафедры «Электропривод и автоматизация промышленных установок», Уральский федеральный университет имени первого Президента России Б.Н. Ельцина, Россия, 620002, г. Екатеринбург, ул. Мира, 19. E-mail: chelovek103@mail.ru. Тел.: +7(982)640-54-35

Грязнов Артем Андреевич, студент кафедры «Электропривод и автоматизация промышленных установок», Уральский федеральный университет имени первого Президента России Б.Н. Ельцина, Россия, 620002, г. Екатеринбург, ул. Мира, 19. E-mail: Argryaznov@ya.ru. Тел.: +7(912)656-75-00

Mikhail Yu. Borodin, Candidate of Technical Sciences, Associate Professor, Department «Electrodrive and automation of industrial plants», Ural Federal University named after the first President of Russia B.N.Yeltsin, 620002, Mira street, 19, Ekaterinburg, Russia. E-mail: bmu@k66.ru. Ph.: +7(922)223-66-54

Konstantin A. Kondakov, Student, Department « Electrodrive and automation of industrial plants», Ural Federal University named after the first President of Russia B.N.Yeltsin,620002, Mira str., 19, Ekaterinburg, Russia. E-mail: chelovek103@mail.ru. Ph.: +7(982)640-54-35

Artem A. Gryaznov, Student, Department « Electrodrive and automation of industrial plants», Ural Federal University named after the first President of Russia B.N.Yeltsin,620002, Mira str., 19, Ekaterinburg, Russia. E-mail: Argryaznov@ya.ru. Ph.: +7(912)656-75-00

Аннотация: Рассматриваются сравнительные характеристики ПЛИС с другими логическими устройствами и целесообразность развития направления и добавления в курс «Микропроцессорная техника» материала для изучения программируемых логических интегральных схем.

Abstract: Consider the comparative characteristics of the FPGA logic with other devices, and the feasibility of the development direction and adding to course "Microprocessor technics" of the material for the study of FPGA.

Ключевые слова: ПЛИС; Альтера; Циклон; VHDL; Верилог; Квартус 2; Симулинк.

Key words: FPGA, Altera; Cyclone; VHDL, Verilog, Quartus II, Simulink.

ВВЕДЕНИЕ

ПЛИС – программируемая логическая интегральная схема, представляет собой цифровую интегральную микросхему, состоящую из программируемых логических блоков и соединений между этими блоками. Первые ПЛИС содержали сравнительно небольшое количество эквивалентных логических вентилях, к тому же производительность схем была невысокой. Поэтому раньше серьезные устройства создавались на базе заказных интегральных схем. Но проектирование и изготовление заказных интегральных схем было довольно трудоемким и

дорогостоящим, к тому же модифицирование схемы практически невозможно. Сейчас ПЛИС стали намного дешевле, быстрее, функциональнее. Причем в любой момент можно внести изменение в проект. Особого внимания заслуживает и тот факт, что современные ПЛИС содержат миллионы эквивалентов логических вентилях, встроенные процессоры и современные высокоскоростные интерфейсы.

Программирование ПЛИС — это создание электрических связей между блоками, с целью синтеза устройств таких как: регистры, счетчики,

таймеры, сумматоры, дешифраторы, шифраторы, триггеры, цифровые компараторы, мультиплексоры и т.д.

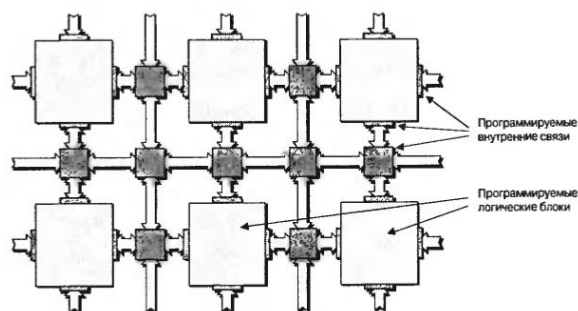


Рис. 1. Архитектура ПЛИС

Создавая связи между синтезируемыми устройствами, мы можем получить довольно сложный проект, причем ПЛИС имеет возможность параллельно выполнять несколько алгоритмов одновременно, в отличие от микроконтроллеров.

Основное и самое главное отличие ПЛИС от микроконтроллера является то, что на микроконтроллере его функции реализуются программно с использованием уже имеющейся внутренней схемы, а в ПЛИС это реализуется аппаратно, то есть созданием соответствующей схемы соединений между внутренними элементами (См. рис. 1). Говорить о том, что предпочтительнее, микроконтроллер или ПЛИС, не совсем корректно, т.к. они созданы для решения различных задач, хотя области их применения могут и пересекаться. Преимуществом ПЛИС над микроконтроллером является то, что последний из-за программной реализации алгоритма медленнее ПЛИС. ПЛИС могут работать на частотах около гигагерца.

ПЛИС ФИРМЫ ALTERA СЕРИИ CYCLONE III

Семейство устройств Cyclone III имеет уникальную комбинацию высокой производительности, высокой функциональности, низкого энергопотребления и низкой цены. Количество логических элементов в данном семействе варьируется от пяти тысяч до двухсот тысяч, а количество ножек от 182 до 535. Встроенная память, предназначенная для загрузки стартовой программы при подаче питания на ПЛИС, составляет от 0,5 до 8 Мб. Дальнейшая отладка и программирование производится с помощью программатора в режимах JTAG (Joint Test Action Group), Active Serial Programming или Passive Serial. Последние два режима предназначены для записи в встроенную память. JTAG необходим для отладки проекта.

Также в этом семействе устройств есть поддержка внутреннего осциллографа. Принцип работы основывается на считывании с отдельной области

памяти состояний входных и выходных сигналов, а также промежуточных, которые служат связью между элементами проекта. Сначала проект записывается через программатор на ПЛИС, и она в свою очередь начинает обрабатывать алгоритм и результаты записывать в отдельную область памяти. Одновременно с этим программатор считывает память и отправляет информацию на персональный компьютер, которая обрабатывается в среде Quartus II и отображается на экране.

Почти все устройства серии Cyclone III имеют 4 кольца фазовой автоподстройки частоты (ФАПЧ) для настройки частоты на которой будет работать алгоритм, заданной в проекте.

Необходимо добавить, что в устройствах этого семейства имеется процессор Nios II, а также они обеспечивают поддержку высокоскоростных интерфейсов внешней памяти таких как DDR, DDR2, SDR SDRAM и QDR II SRAM.

ЯЗЫКИ ПРОГРАММИРОВАНИЯ

В программном обеспечении Quartus II имеется два основных языка: VHDL и Verilog HDL. Verilog поддерживает весьма прогрессивную концепцию, называемую интерфейсом языка программирования или PLI Verilog (Programming Language Interface). В общем виде эта концепция известна как интерфейс прикладного программирования или API (Application Programming Interface). API представляет собой библиотеку программных функций, которая позволяет внешнему программному обеспечению передавать данные в приложение и получать из него выходные результаты. Другими словами, функции API Verilog позволили пользователям расширить функциональность этого языка и поставляемого с ним пакета логического моделирования. VHASIC (Very High Speed Integrated Circuit) или сокращенно VHDL не использует библиотеки, а также отличается большей мощностью на функциональном (булевы выражения и RTL) и поведенческом (алгоритмическом) уровнях абстракции, а также поддерживает некоторые конструкции системного уровня.

Возможности языков в общем похожи, но их применение все же немного имеет разные области. Более подробно об применениях языков и сравнительная характеристика отображена на рис.2.

Изначально, как правило, для описания проекта использовался только один язык. Но с развитием проектов на ПЛИС и использование уже готовых, заранее описанных кем-либо другим блоков приводило к тому, что в проекте появлялась необходимость использовать оба языка. В настоящее время такая возможность есть.

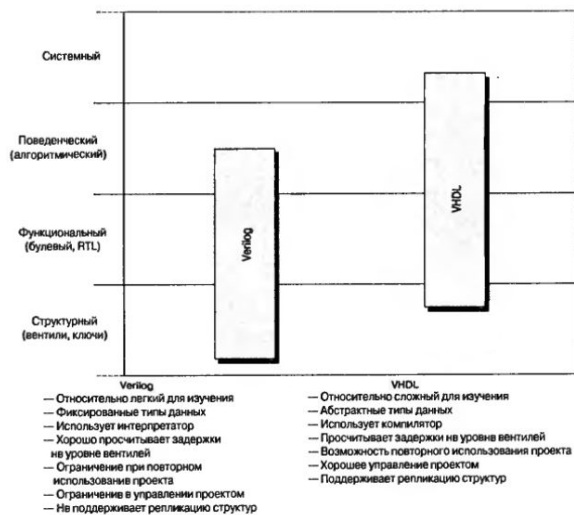


Рис.2. Области применения и сравнительная характеристика языков программирования ПЛИС

ПЛИС стали поддерживать многоязычные программные продукты, в число которых входили модули логического моделирования, синтеза и другие средства, которые могли работать с устройствами, состоящими из блоков, описанных с помощью и VHDL, и Verilog.

Команды, используемые в этих языках, имеют сходство с языком C++. Следовательно, перейти к изучению этих языков программирования не создаст больших трудностей.

Несмотря на то, что развитие одного из двух языка предвещает конец другого, оба языка программирования используются и в настоящее время.

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ QUARTUS II

Quartus II обеспечивает разработку полного проекта, которая легко адаптируется под необходимые требования для проекта. Имеется полный набор инструментов для отладки и конечной записи проекта на ПЛИС.

Оболочка программы имеет довольно функциональный вид. Имеется строка команд, окно с иерархией проекта, с помощью которого легко переходить от одного функционального блока к другому, окно синтеза (показывает на каком этапе происходит синтез проекта и длительность каждого этапа), информационное окно с отображением сообщений и ошибок, поле редактора.

Создание нового проекта начинается с основного описание директории создания проекта, выбора программируемой ПЛИС, а также выбора средств синтеза, симуляции и временного анализа.

Далее, описываются функциональные блоки на языках Verilog или VHDL. Структура функционального блока одинакова для обоих

языков и состоит она из названия блока, описания входов и выходов, текста программы и присвоения промежуточных вычисляемых значений выходным путям блока.

Одним из достоинств среды Quartus II является поддержка графического представления проекта.

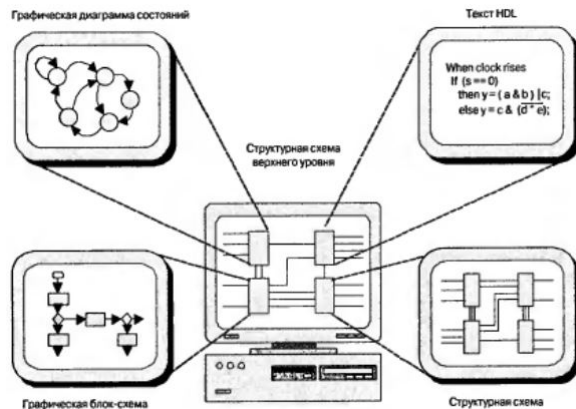


Рис.3. Графическое представление проекта в Quartus II

Функциональные блоки соединяются между собой связями, образуя взаимосвязь между отдельными функциональными блоками и пинами ввода/вывода, образуя готовый проект. Это достаточно удобно, т.к. намного легче найти ошибку, применяя графическое изображение проекта, нежели иметь большое количество строк, в которых легко потеряться. Двойной щелчок мыши по функциональному блоку открывает его HDL описание.

Также в достоинства программы можно добавить уже имеющиеся функциональные блоки, описанные компанией Altera. Также есть возможность формирования этих блоков под наши требования проекта. К примеру, мы с легкостью можем добавить в наш проект счетчик, настроить его разрядность, метод работы (реверсивный или нереверсивный и подобное), до какого значения должен считать счетчик и т.д. Создание таких блоков и их отладка производится с помощью инструмента MegaWizard Plug-In Manager. Также с помощью этого инструмента мы можем быстро добавить в наш проект блоки PLL (ФАПЧ), блоки логических и арифметических операций, фильтры, генераторы сигналов, блоки обработки видео и изображения, интерфейсы (ASI, Ethernet, PCI, RapidIO, SDI и другие), стековую память и некоторые другие.

После создания блок схемы, описания назначений ножек и настройки ПЛИС (настройки неиспользуемых пинов, напряжения и токи на ножках и другие настройки) производим компиляцию проекта. Цикл компиляции показан на рис. 4.

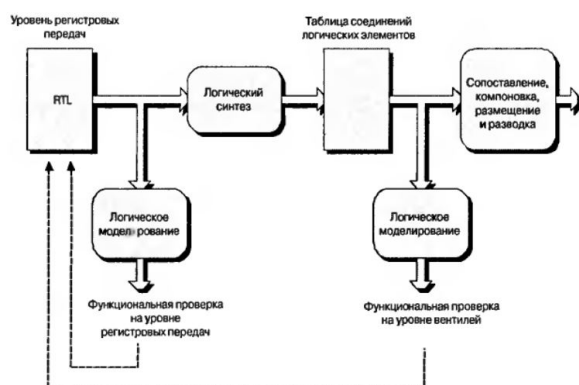


Рис.4. Цикл HDL-проектирования ПЛИС

Цикл обеспечивает синтез проекта с несколькими проверками его работоспособности и правильной компиляции. Средства логического синтеза автоматически конвертируют RTL-описание устройств в набор регистров и булевых выражений, попутно реализуя различные процедуры минимизации и оптимизации, в том числе оптимизацию по временным задержкам и по площади используемого места на кристалле. После этого средства синтеза генерируют таблицы соединений выводов, которые должны соответствовать начальным временным ограничениям. По окончании синтеза Quartus II показывает критические ошибки, синтаксические ошибки и информационные.

Логическую работу проекта можно оценить с помощью инструмента ModelSim. Выбрав нужные нам порты ввода/вывода, задаем входные воздействия и можем наблюдать выходные сигналы. Причем есть возможность осуществить такую проверку с учетом внутренних задержек между программируемыми логическими блоками и портами ввода/вывода.

Также ранее упоминалось, что имеется программный осциллограф в Quartus II. Использовать его мы можем с помощью инструмента SignalTab II Logic Analyzer. Подобно ModelSim мы задаем нужные нам порты ввода/вывода, только входные воздействия задаются не программно, а с помощью физических коммутаций, выполненных с помощью ключей, подключенных к пинам на ПЛИС.

ВЫВОДЫ

Перспективы развития ПЛИС в лабораторном практикуме кафедры довольно значительные. Это связано с большим быстродействием и довольно быстрым развитием технологий устройств. С каждым годом появляются кристаллы более функциональные, и только по количеству логических элементов современные ПЛИС достигают порядка миллиона на настоящее время и это не является пределом.

Также необходимо добавить, что существует возможность программирования ПЛИС с помощью MATLAB, а точнее с помощью Simulink. Компания MathWorks выпустила пакет HDL Coder, с помощью которого и осуществляется связь между Simulink и ПЛИС. Simulink это довольно функциональное программное обеспечение, с помощью которого можно создавать довольно сложные проекты.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Максфилд К. Проектирование на ПЛИС. Архитектура, средства и методы. Москва, Издательский дом «Додэка-XXI». 2007. 407 с.
2. Altera – Quartus II Handbook Version 13.1. Volume 1: Design and Synthesis 2013, 1681 с.
3. Altera – Cyclone III Device Handbook. Volume 1 2012. 348 с.