

ПРОЕКТИРОВАНИЕ ЦИФРОВЫХ УСТРОЙСТВ НА ПРОГРАММИРУЕМЫХ ЛОГИЧЕСКИХ ИНТЕГРАЛЬНЫХ СХЕМАХ

Старшинов В.С.

Научный руководитель: Мальчуков А.Н.
Томский политехнический университет
vss21@tpu.ru

Введение

В настоящее время все большее применение при разработке цифровых устройств применяются программируемые логические интегральные схемы (ПЛИС) [1].

Существенным преимуществом ПЛИС является их универсальность и возможность быстрого программирования под выполнение функций практически любого цифрового устройства. ПЛИС представляет собой полуфабрикат, на основе которого разработчик, обладающий персональным компьютером, имеет возможность проектирования цифрового устройства в рекордно короткие сроки. Обеспечивается это несложными и относительно недорогими аппаратными средствами программирования и специальным программным обеспечением, называемым системой автоматизированного проектирования (САПР) [2].

Структура ПЛИС и принципы программирования

Программируемая логическая интегральная схема – электронный компонент, используемый для создания цифровых интегральных схем. В отличие от обычных цифровых микросхем, логика работы ПЛИС задается посредством программирования с помощью специальных средств: программаторов и программного обеспечения [3].

Обобщенная структура ПЛИС представлена на рис.1.

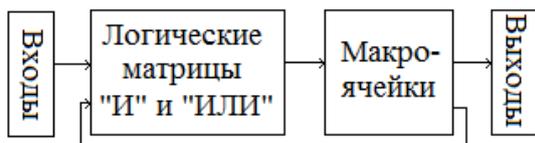


Рис. 1. Обобщенная структура ПЛИС

Программирование на ПЛИС осуществляется с помощью языков описания аппаратуры Verilog и VHDL.

На верхнем уровне эти языки очень схожи – модель аппаратуры описывается в виде взаимодействующих блоков (модулей) и для каждого из них определяется интерфейс и реализация. Интерфейсы модулей описывают входные, выходные и двусторонние порты, благодаря которым модули соединяются друг с другом с целью обмена данными, а также управляющими сигналами. Реализация задает элементы внутреннего состояния и порядок

вычисления значений выходных интерфейсов на основе этого состояния и значений входных портов, а также правила обновления внутреннего состояния [4].

Общий вид маршрута проектирования цифровых устройств

Проектирование устройств на основе ПЛИС выполняется с применением специализированных САПР. Проектирование с помощью таких САПР заключается в последовательном использовании предоставляемых программных средств. В терминологии САПР такой процесс называется маршрутом проектирования. Проектирование традиционно разделяют на этапы: системный, структурно-алгоритмический и функционально-логический, конструкторско-технологический.

На системном этапе весь проект разбивается на части, определяются их назначение и взаимосвязь, принимается решение о способах реализации частей и выбирается способ описания устройства.

Структурно-алгоритмический и функционально-логический этапы проектирования устройств на основе ПЛИС базируются на итерационном вводе и верификации описаний параллельно функционирующих процессов, каждый из которых реализует заданный алгоритм.

Конструкторско-технологический этап связан с выбором способа описания устройства.

Современные САПР поддерживают несколько способов описания устройства:

- с использованием языков описания аппаратных средств (VHDL, Verilog, AHDL и др.) и специализированного текстового редактора;
- схемотехнический способ описания с помощью программы визуального проектирования, позволяющей разработчику помещать на рабочую область функциональные блоки и производить их соединение. По окончании визуального проектирования схема преобразуется в языковое описание;
- графическое представление цифровых автоматов в специализированном редакторе, обеспечивающем преобразование полученного графического представления в языковое описание;
- описание комбинационной логики с помощью таблиц истинности, карт Карно, функций алгебр.

Компилятор должен проанализировать пользовательский проект (схемы и текстовые

описания на Verilog HDL или VHDL) и сгенерировать список всех элементов схемы и связи между ними, который называется netlist. Netlist должен быть оптимизирован – логические функции нужно минимизировать, возможные дублированные регистры нужно удалить. Затем компилятор должен вместить всю логику из netlist в имеющуюся архитектуру ПЛИС. Этот процесс выполняет fitter. Fitter размещает логические элементы и выполняет трассировку связей между ними (процесс place and route) [3].

После выполнения трассировки и верификации результатов автоматически может быть сгенерирован файл с конфигурационной последовательностью, содержащий информацию о коммутации и функциональности всех ресурсов кристалла. На заключительном этапе маршрута проектирования выполняются программирование ПЛИС и последующая внутрисхемная верификация устройства (проверка работоспособности на макетной ПЛИС) [5,6].

Использование синтезируемого процессора Nios II в Altera Quartus

Nios II – это программный 32-разрядный процессор, оптимизированный для реализации в FPGA производства компании Altera. Процессор построен по архитектуре RISC, имеет 32-х разрядные шины данных и адреса, 32 регистра общего назначения и 32 источника внешних прерываний.

В микросхемах Altera SoC FPGA (Cyclone V SoC, Arria V SoC, Arria 10 SoC) Nios II может быть использован как сопроцессор, выполняющий какие-то определенные функции (управление моторами, зарядом аккумуляторов, тачскрином и т.п.) для того, чтобы разгрузить основное вычислительное ядро. Синтезируемый процессор Nios II позволяет получить наилучшее соотношение производительность/ аппаратные затраты при реализации на ПЛИС на Altera.

Создание прошивки ПЛИС для устройства с использованием Nios II делится на две части — описание аппаратной части и описание программной части.

В аппаратной части надо выбрать ПЛИС, с которой будет дальнейшая работа. После этого надо создать top-level файл (схема или HDL, который будет описывать как ядро подключено к внешнему миру. В списке библиотек необходимо выбрать Nios II Processor и выбрать его модификацию: nios II/f (fast), nios II/e (economy) или nios II/s (standart).

После генерирования заданных настроек происходит сохранение файла описания системы, а SOPC Builder автоматически перейдет на вторую вкладку и начнет генерацию описания нашего процессора. Для того, чтобы компиляция прошла успешно, необходимо расставить номера выводов в Pin Planner. Полученный файл прошивки (с

расширением *.sof) нужно загрузить в ПЛИС привычным методом для дальнейшего программирования.

В программной части надо выбрать файл описания системы (с расширением *.sopcinfo) и ядро, для которого будем писать (в системе может быть много процессорных ядер).

После этого необходимо создать новый файл для исходника нашей программы. Файл автоматически добавится в проект, но вот добавить его в сборочный скрипт нужно руками. Ввиду того, что сам процессор уже залит в ПЛИС, пишется программа на языке С.

Полученный файл (с расширением *.elf) загружается в процессор, и программа выполняется [3,7].

Заключение

В работе представлены структура ПЛИС и принципы его программирования, обобщенный маршрут проектирования цифровых устройств с описанием этапов. Кроме этого, было рассмотрено использование синтезируемого процессора Nios II.

Выполненные проекты на ПЛИС обладают многими преимуществами: высокой функциональностью, легкостью модификации, большим быстродействием и большой степенью автоматизации.

Список использованных источников

5. Бутаев М.М., Вашкевич Н.П., Гурин Е.И., Коннов Н.Н. Проектирование цифровых устройств на программируемых логических интегральных схемах: Учебн. пособие. - Пенза: Изд-во Пенз. гос. техн. ун-та, 1996. - 4 с.
6. Правила оформления схем цифровых устройств. [Электронный ресурс]. – URL: <http://mydocx.ru/4-110927.html> (Дата обращения 22.10.2015)
7. Архитектура ПЛИС. [Электронный ресурс]. – URL: <http://marsohod.org/index.php/ourblog/11-blog/265-fpga> (Дата обращения 22.10.2015)
8. Обзор методов описания встраиваемой аппаратуры и построения инструментария кросс-разработки. [Электронный ресурс]. – URL: <http://citforum.ru/programming/embedded/languages/2.shtml> (Дата обращения 22.10.2015)
9. Маршрут разработки цифровой СБИС на основе БМК. [Электронный ресурс]. – URL: <http://sibac.info/17496> (Дата обращения 22.10.2015)
10. Попов А.Ю. Проектирование цифровых устройств с использованием ПЛИС: Уч. пособие. — М.: Изд-во МГТУ им. Н.Э. Баумана, 2009. — 51 с.
11. Altera Nios II QuickStart. [Электронный ресурс]. – URL: <http://we.easyelectronics.ru/plis/altera-nios-ii-quickstart-osvaivaem-principy-postroeniya-sistemy-i-infrastrukturu-sborki.html> (Дата обращения 22.10.2015)