

Практика использования встроенного АЦП в ПЛИС семейства MAX 10

Часть 2. Работа с АЦП в ПЛИС семейства MAX 10

Павел Редькин (г. Ульяновск)

Статья содержит информацию по практическому применению аппаратного модуля АЦП, входящего в состав ПЛИС семейства MAX 10 производства Intel (Altera). Во второй части статьи рассмотрены вопросы создания проекта ПЛИС с модулем АЦП в среде Quartus II, настройки конфигурации АЦП в проекте, загрузки конфигурации в ПЛИС на примере конкретной отладочной платы с ПЛИС семейства MAX 10.

Аппаратная платформа ПЛИС с АЦП

В качестве аппаратной платформы ПЛИС с АЦП в данной статье применена отладочная плата Altera MAX 10 FPGA, имеющая «на борту» ПЛИС семейства MAX 10 10M08SAE144C8G с 8000 логическими элементами и одним встроенным аппаратным модулем АЦП. Принципиальную схему и описание платы Altera MAX 10 FPGA можно загрузить со страницы производителя [1].

Помимо ПЛИС на плате Altera MAX 10 FPGA имеется разъём mini-USB для подачи внешнего напряжения питания, стабилизатор напряжения +3,3 В, кварцевый генератор с частотой 50 МГц, микросхема источника опорного напряжения (ИОН), разъём JTAG для загрузки и конфигурирования ПЛИС, схема сброса ПЛИС, пользовательские светодиоды, джамперы и прочие вспомогательные элементы. Аналоговые универсальные

входы АЦП в ПЛИС ADC1IN1-ADC1IN8 на плате Altera MAX 10 FPGA снабжены фильтрующими RC-цепочками, а также входными буферными элементами – операционными усилителями (ОУ) LM2902, включёнными в режиме повторителей напряжения. Цифровые линии GPIO, а также аналоговые универсальные входы ПЛИС выведены на плате на линейки соответствующих разъёмов.

На вход подачи внешнего опорного напряжения ADC_VREF поступает напряжение +2,5 В с выхода микросхемы ИОН MCP1525. Выделенный вход АЦП в ПЛИС ANAIN1 на отладочной плате не задействован (соединён с общим проводом через резистор нулевого сопротивления). Для использования входа ANAIN1 в проекте ПЛИС в плату необходимо внести небольшие доработки, о чём будет подробнее сказано далее.

Для исследования модуля АЦП в ПЛИС автором был собран макет, состоящий из платы Altera MAX 10 FPGA, подключённых к ней жидкокристаллического индикатора 12864ZW и линейки из шести кнопок управления. Индикатор в проекте ПЛИС работает в символьном режиме и позволяет отображать четыре строки по 16 символов в каждой. На свободном макетном поле платы Altera MAX 10 FPGA смонтирован электронный ключ на полевом транзисторе КП501А, нагруженный на звуковой излучатель электромагнитного типа. Данный узел используется в проекте ПЛИС для озвучивания нажатий на кнопки. Общий вид собранного макета представлен на рисунке 9.

Принципиальная схема собранного на базе платы Altera MAX 10 FPGA макета показана на рисунке 10. Для упрощения восприятия часть узлов и элементов, установленных производителем на

плате Altera MAX 10 FPGA, на рисунке не показана. Также на схеме не показаны цепи питания ПЛИС, фильтрующие конденсаторы на этих цепях и некоторые служебные сигналы ПЛИС, не имеющие принципиального значения для рассматриваемой темы. Элементы аналоговой части (ОУ повторителей напряжения, внешние входные аналоговые цепи универсальных входов ПЛИС) на схеме показаны только в части канала 7 АЦП для универсального входа ADC1IN7 (вывод 13 в ПЛИС). Позиционные обозначения элементов и наименования цепей отладочной платы на рисунке 10 соответствуют оригинальной документации производителя [1]. Позиционные обозначения дополнительных элементов макета, не входящих в состав отладочной платы (ЖКИ, кнопок, узла генерации звука), имеют свою собственную нумерацию.

Внешнее питание – постоянное напряжение +5 В – плата получает через разъём J1, имеющий форм-фактор mini-USB. Этот разъём предназначен только для подачи питания, поскольку его контакты цепей данных интерфейса USB (D+, D–) ни с чем на плате не соединены. Поступающее через разъём J1 напряжение +5 В понижается с помощью стабилизатора U1 до значения +3,3 В. Далее из напряжения +3,3 В вырабатываются: напряжение питания аналоговой части платы ПЛИС и аналоговой части платы VCCA_3.3V, напряжение питания ядра ПЛИС VCC_CORE, напряжения питания системы ввода-вывода ПЛИС и цифровой части платы VCC_IO.

Загрузка конфигурации в конфигурационную память и в саму ПЛИС осуществляется через разъём JTAG J10 с элементами его «обвязки».

Цепь аналогового общего провода на плате подводится к общему проводу аналоговых элементов платы: ОУ повторителей напряжения и элементов их «обвязки», ИОН, элементов внешних аналоговых цепей универсальных входов ПЛИС, аналоговому общему проводу ПЛИС, разъёму аналоговых сигналов J3. Цепь цифрового общего провода на плате подводится к общему проводу цифро-

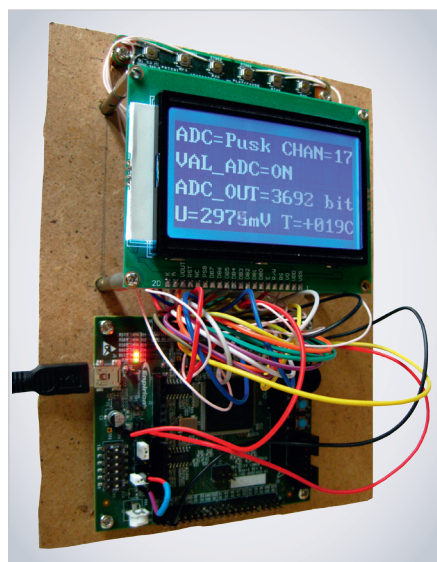


Рис. 9. Общий вид собранного макета с платой Altera MAX 10 FPGA



ЗАО «НАУЧНО-ПРОИЗВОДСТВЕННАЯ ФИРМА «ДОЛОМАНТ»

**ОТВЕТСТВЕННАЯ ЭЛЕКТРОНИКА
ДЛЯ ЖЕСТКИХ УСЛОВИЙ ЭКСПЛУАТАЦИИ**

2021

100% РОССИЙСКАЯ КОМПАНИЯ



ЗАКАЗНЫЕ РАЗРАБОТКИ

Разработка электронного оборудования по ТЗ заказчика в кратчайшие сроки

- Модификация КД существующего изделия
- Разработка спецвычислителя на базе СОМ-модуля
- Конфигурирование модульного корпусированного изделия
- Сборка магистрально-модульной системы по спецификации заказчика
- Разработка изделия с нуля



КОНТРАКТНОЕ ПРОИЗВОДСТВО

Контрактная сборка электроники уровней: модуль / узел / блок / шкаф / комплекс

- ОКР, технологические консультации и согласования
- Макеты, установочные партии, постановка в серию
- Полное комплектование производства импортными и отечественными компонентами и материалами
- Поддержание складов, своевременное анонсирование снятия с производства, подбор аналогов
- Серийное плановое производство
- Тестирование и испытания по методикам и ТУ
- Гарантийный и постгарантийный сервис

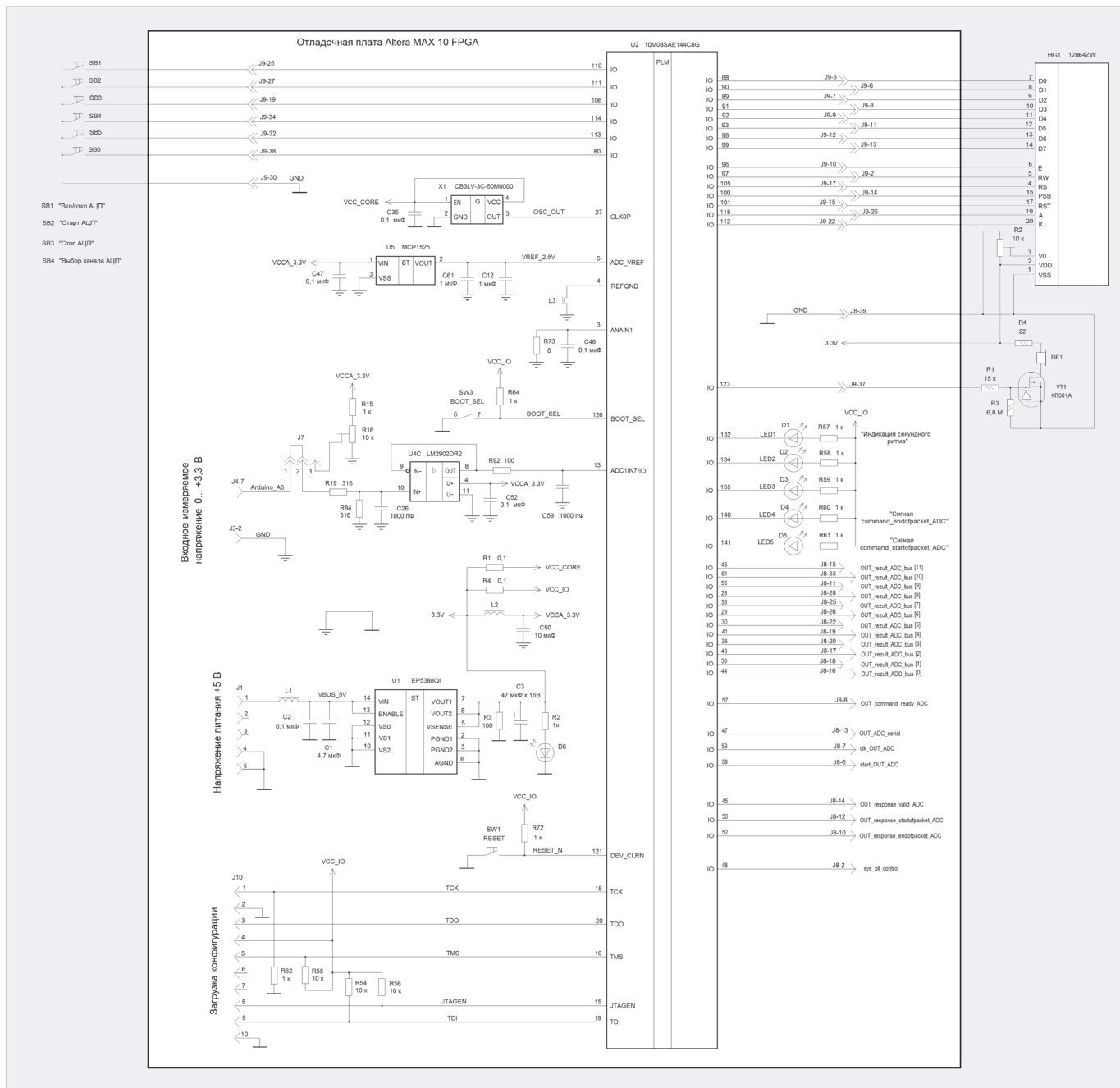


Рис. 10. Принципиальная схема макета, собранного на базе платы Altera MAX 10 FPGA

вых элементов платы: цифровому общему проводу ПЛИС, цифровому общему проводу стабилизатора напряжения +3,3 В, цифровому общему проводу генератора глобального синхросигнала 50 МГц, цифровому общему проводу, разъёмам цифровых сигналов J2, J8, J9. Как показано на схеме, цифровой и аналоговый общие провода на плате соединяются друг с другом. С точки зрения минимизации влияния цифровых шумов на точность аналоговых преобразований по цепям общего провода данное соединение целесообразно выполнить в одной точке платы.

Создание проекта ПЛИС

Процесс создания в инструментальной среде Quartus II проекта ПЛИС,

включающего в себя IP-ядро поддержки встроенного АЦП ПЛИС, имеет ряд специфических особенностей. Рассмотрим создание такого проекта по шагам.

Сначала в главном меню среды Quartus II (автор использовал 64-разрядную среду Quartus II Version 15.0.0 Web Edition) выбираем File → New Project Wizard... В открывшемся окне New Project Wizard на странице под заголовком Directory, Name, Top-Level Entity задаём первичные параметры проекта: путь к директории проекта на диске, имя проекта, а также модуль верхнего уровня проекта. В нашем случае все перечисленные имена совпадают (MAX_10_ADC_1), как показано на рисунке 11. По указанному пути предва-

рительно следует создать каталог проекта, который будет содержать все его исходные файлы. В нашем случае это файлы с исходным текстом на языке описания аппаратных средств Verilog HDL, имеющие расширение *.v. Один из этих файлов имеет имя MAX_10_ADC_1.v. Этот файл содержит модуль верхнего уровня проекта. Другие исходные файлы содержат модули более низкого уровня, реализующие взаимодействие модуля верхнего уровня с ЖКИ, кнопками управления, вывод из ПЛИС данных АЦП в последовательном виде, преобразование результата АЦП в значение температуры, генерацию звука. Все перечисленные исходные файлы можно создавать в любом тексто-

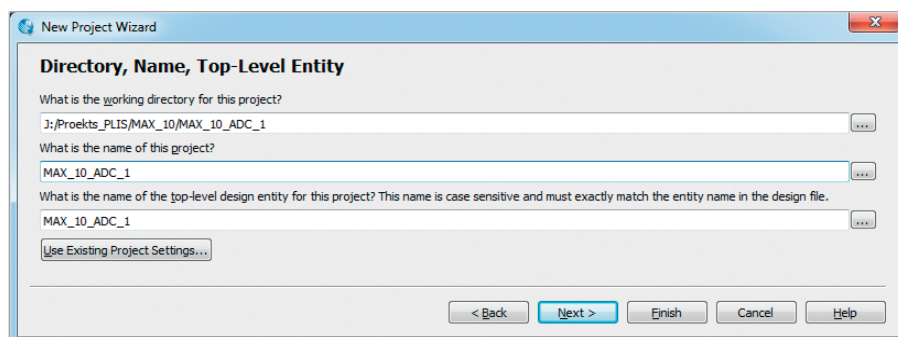


Рис. 11. Задание первичных параметров проекта ПЛИС

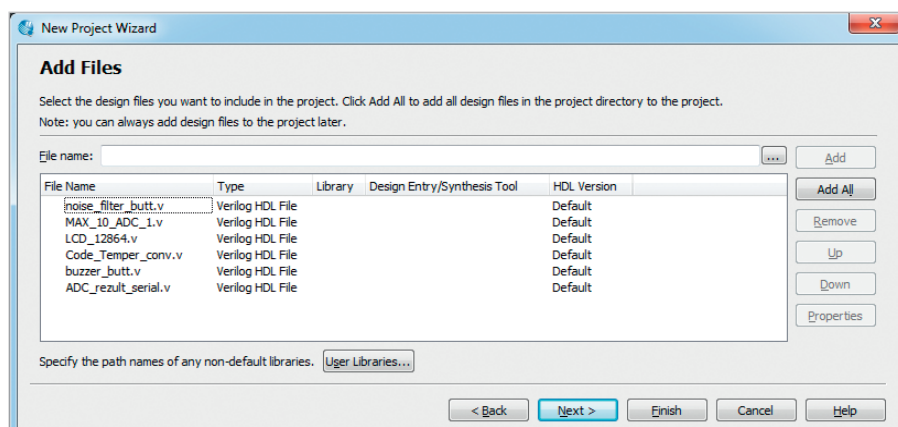


Рис. 12. Включение файлов в состав проекта ПЛИС

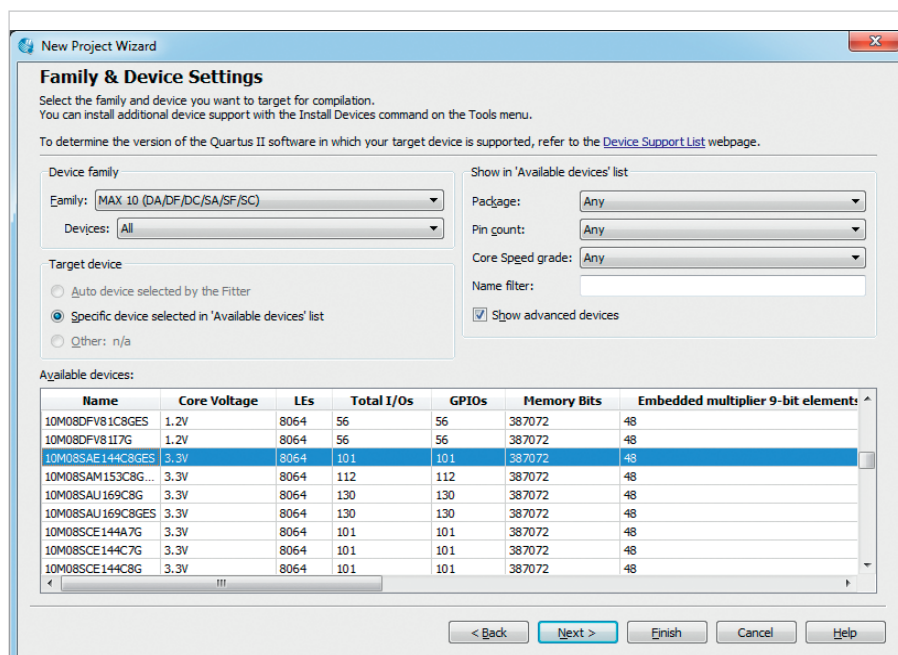


Рис. 13. Выбор ПЛИС для проекта

данных ранее исходных файлов в состав проекта. Для этого нужно нажать на кнопку с многоточием (...) справа от позиции File name, затем в открывшемся окне каталогов и файлов выбрать нужный исходный файл и добавить этот файл в проект нажатием на кнопку Add. Имена и тип всех включённых в состав проекта файлов будут перечислены в окне странице Add Files, как показано на рисунке 12.

Далее необходимо задать аппаратную платформу для нашего проекта ПЛИС. Для этого в окне New Project Wizard нажимаем на кнопку Next, затем в открывшейся странице Family & Device Settings выбираем требуемое семейство ПЛИС (MAX 10) и конкретную модель ПЛИС в этом семействе (10M08SAE144C8GES), как показано на рисунке 13.

Теперь следует задать базовые настройки симуляции для нашего проекта ПЛИС. Для этого в окне New Project Wizard нажимаем на кнопку Next. В открывшейся странице EDA Tool Settings выбираем в строке Simulation в графе Tool Name инструмент симуляции ModelSim-Altera, а в графе Format(s) – язык симуляции Verilog HDL. Остальные настройки на странице оставляем заданными по умолчанию.

В заключение ещё раз в окне New Project Wizard нажимаем на кнопку Next, и на открывшейся странице Summary проверяем корректность всех заданных ранее настроек проекта. При отсутствии замечаний кликаем на кнопке Finish, после чего создание проекта считается завершённым. Созданный проект появляется во вкладке Hierarchy области навигации Project Navigator среды Quartus II, как показано на рисунке 14. Все входящие в проект файлы перечислены во вкладке Files области навигации Project Navigator.

Включение в проект ПЛИС модуля PLL

После создания проекта ПЛИС, перед включением в него модуля АЦП необходимо задать для него источник синхросигнала. Как было сказано в первой части статьи, в качестве синхросигнала для АЦП в ПЛИС семейства MAX10 производителем рекомендуется использовать выходной сигнал встроенной в ПЛИС системы PLL. Для реализации этой рекомендации необходимо включить в проект ПЛИС IP-ядро PLL. В составе ПЛИС 10M08SAE144C8G имеется только один аппаратный модуль PLL, его мы и задействуем. Для этого воспользуемся встроенным в среду Quartus II каталогом IP-ядер IP

вом редакторе, однако удобнее делать это в среде Quartus II, имеющей цветовую подсветку синтаксиса исходных текстов на языках описания аппаратных средств Verilog HDL, VHDL и AHDL. Для создания нового исходного файла следует в главном меню среды Quartus II выбрать File → New..., после чего откроется окно New, в котором можно выбрать тип создаваемого файла.

Далее будет показан порядок включения всех созданных исходных файлов в состав проекта.

Следует нажать в окне New Project Wizard на кнопку Next. В открывшейся странице Project Type оставляем заданную по умолчанию настройку типа проекта Empty project и снова нажимаем на кнопку Next. В открывшейся странице Add Files производим включение соз-

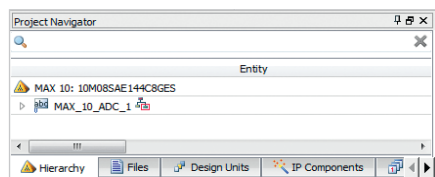


Рис. 14. Созданный в Quartus II проект ПЛИС

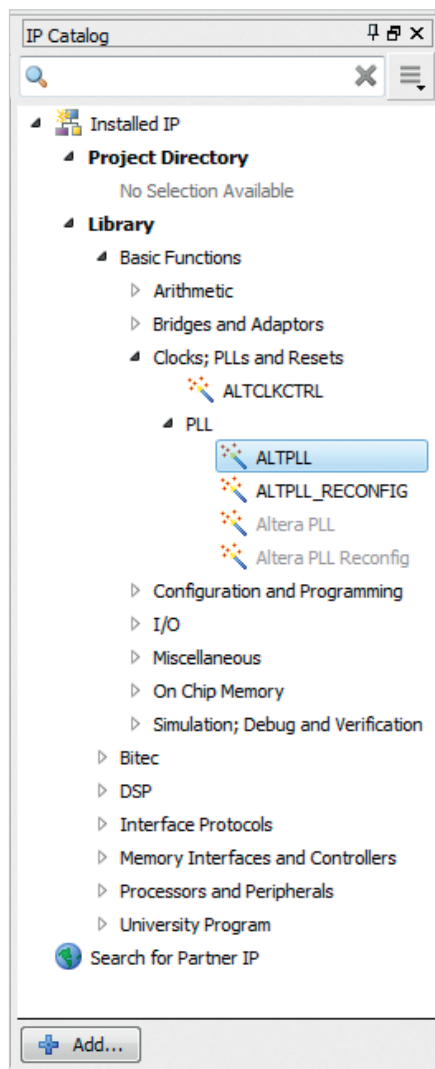


Рис. 15. Включение в состав проекта ПЛИС модуля PLL

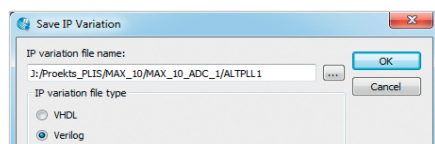


Рис. 16. Задание первичных параметров IP-ядра PLL

Catalog, выбрав в нём ALTPLL (см. рис. 15). После этого откроется окно задания первичных параметров IP-ядра PLL Save IP Variation, в котором необходимо указать путь к директории хранения исходного файла IP-ядра PLL внутри директории проекта ПЛИС, его имя (в нашем случае это ALTPLL1) и язык представления Verilog (см. рис. 16). После нажатия на кнопку ОК запускается инструменталь-

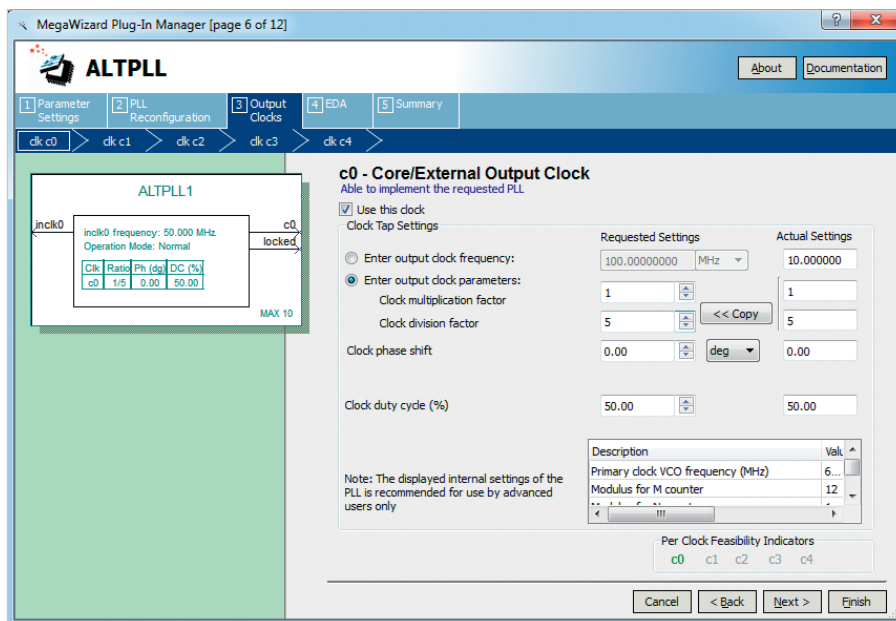


Рис. 17. Задание выходных параметров IP-ядра PLL

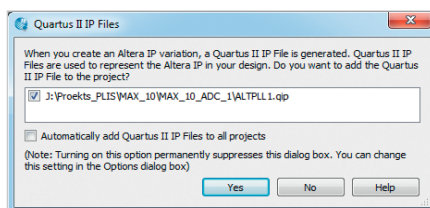


Рис. 18. Запрос на включение IP-ядра PLL в состав проекта ПЛИС

ное средство среды Quartus II для редактирования IP-ядер MegaWizard Plug-In Manager, с помощью которого можно задать все параметры нашего модуля PLL. Открывшееся окно MegaWizard Plug-In Manager имеет несколько вкладок, на каждой из которых задаётся определённая группа параметров. В соответствии с рекомендациями производителя выбираем для синхронизации аппаратного модуля АЦП частоту 10 МГц, при том что частота нашего глобального синхросигнала (внешнего генератора на плате Altera MAX 10 FPGA) составляет 50 МГц. Таким образом, систему PLL необходимо сконфигурировать так, чтобы она обеспечивала деление входной частоты 50 МГц на 5 для получения выходной частоты 10 МГц, имела один выход c0, а также – выход записываемого ею устройства locked. Значение входной частоты PLL (50 МГц) и количества выходов PLL (один выход c0) необходимо задать на вкладке 1 Parameter Settings. Делитель PLL задается на вкладке 3 Output Clocks, как показано на рисунке 17.

Требуемая структурная схема IP-ядра PLL показана на рисунке 17 слева: один вход синхросигнала (incl0), один выход синхросигнала (c0), один выход

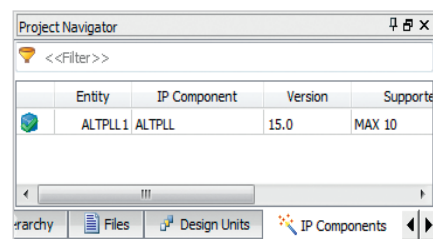


Рис. 19. IP-ядро PLL в составе проекта ПЛИС

записываемого (locked). Все остальные параметры на прочих вкладках MegaWizard Plug-In Manager следует оставить заданными по умолчанию. Для последовательного перемещения по вкладкам при настройке параметров PLL следует использовать кнопки <Back, Next>.

По завершении конфигурирования PLL после нажатия на кнопку Finish открывается окно с запросом на включение сконфигурированного нами IP-ядра PLL (файл ALTPLL1.qip) в состав проекта, как показано на рисунке 18. Если согласиться с запросом на включение (кнопка Yes), то во вкладке Files области навигации Project Navigator проекта ПЛИС появятся файлы IP-ядра PLL (ALTPLL1.qip) и исходного кода модуля PLL (ALTPLL1.v), а во вкладке IP Components – позиция IP-ядра PLL, как показано на рисунке 19. Файл ALTPLL1.v содержит модуль на языке Verilog HDL с исходным кодом мегафункции ALTPLL. Имя и список входов и выходов этого модуля соответствуют ранее заданным параметрам PLL и выглядят, как показано в листинге 1.

Однако само по себе добавление IP-ядра PLL в проект ПЛИС автоматически не обеспечивает его использования в проекте. Для его использова-



ИСПЫТАТЕЛЬНАЯ ЛАБОРАТОРИЯ

ЭЛЕКТРОННОЙ КОМПОНЕНТНОЙ БАЗЫ

АО «ТЕСТПРИБОР» ПРОВОДИТ СЕРТИФИКАЦИОННЫЕ ИСПЫТАНИЯ ЭКБ

Преимущества:

- ✓ Безупречное качество всех выполняемых работ
- ✓ Конфиденциальность
- ✓ Независимая оценка
- ✓ Современное оборудование
- ✓ Квалифицированный инженерно-технический персонал

ВИДЫ РАБОТ

- Анализ применяемой ЭКБ, сбор и разработка технической документации
- 100% входной контроль и идентификация продукции
- 100% отбраковочные испытания
- 100% диагностический неразрушающий контроль
- Разработка программного обеспечения и методик сертификационных испытаний
- Климатические испытания
- Механические испытания
- Разработка и изготовление технологической оснастки

Реклама



+7 (495) 657-87-37



tp@test-expert.ru
www.test-expert.ru



125480, г. Москва,
ул. Планерная, д. 7А

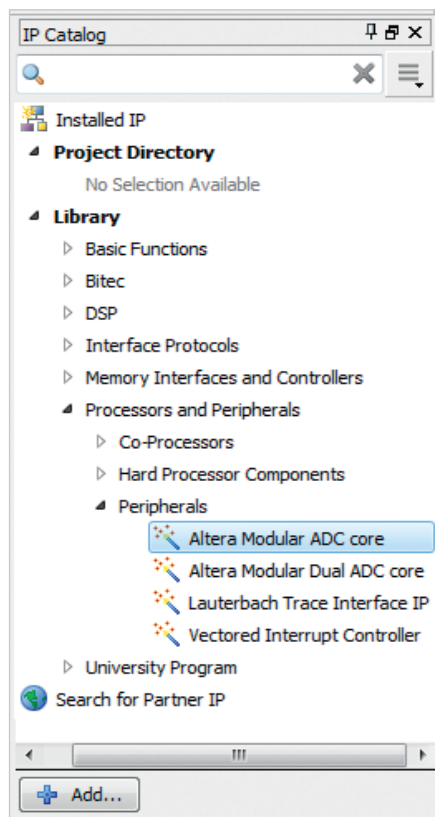


Рис. 20. Выбор IP-ядра модуля АЦП для включения в проект ПЛИС в среде Quartus II

Листинг 1

```
module ALTPLL1 (
    inclk0,
    c0,
    locked);
```

Листинг 2

```
ALTPLL1 ALT_PLL_1(
    .inclk0(sys_clk),
    .c0(clk_pll),
    .locked(locked_pll)
);
```

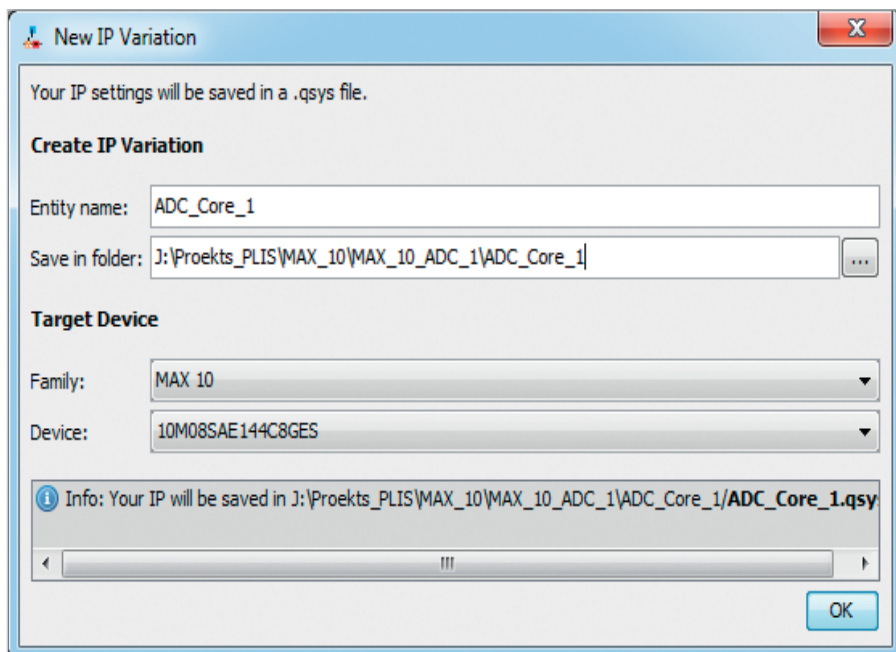


Рис. 21. Создание файла с расширением .qsys

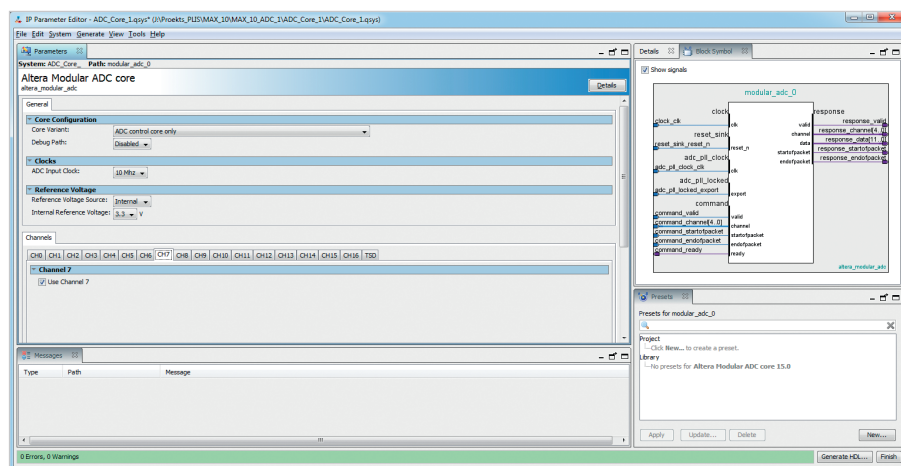


Рис. 22. Настройка параметров АЦП в редакторе IP-ядер IP Parameter Editor

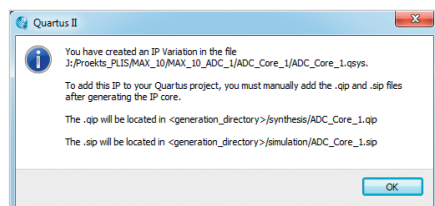


Рис. 23. Порядок действий по включению сгенерированного HDL-кода поддержки АЦП в проект ПЛИС

ния в состав исходного текста модуля верхнего уровня проекта ПЛИС (файл MAX_10_ADC_1.v) необходимо вручную включить фрагмент кода, устанавливающий связь между сигналами модуля

верхнего уровня проекта и модуля PLL (см. листинг 2). На листинге 2:

- ALT_PLL_1 – имя модуля нижнего уровня, используемого в модуле верхнего уровня (это имя можно задать произвольно);
- sys_clk – частота глобального синхросигнала, поступающего на вход PLL в проекте ПЛИС (50 МГц);
- clk_pll – выходная частота PLL, которая в дальнейшем будет использоваться для синхронизации модуля АЦП в проекте ПЛИС (10 МГц);
- locked_pll – сигнал запирающего модуля АЦП, поступающий в него из модуля PLL.

На этом добавление IP-ядра PLL в проект ПЛИС можно считать завершённым.

Включение в проект ПЛИС модуля АЦП

Включение модуля АЦП в проект ПЛИС в среде Quartus II также осуществляется через каталог IP-ядер IP Catalog, как показано на рисунке 20. В каталоге IP Catalog выбирается одинарное или двойное IP-ядро АЦП в соответствии с тем, сколько аппаратных модулей АЦП содержит используемая в проекте ПЛИС и сколько их требуется пользователю. В нашем случае выбираем одинарное IP-ядро Altera Modular ADC core. После выбора нужного IP-ядра пользователю будет предложено задать местонахождение и имя файла с расширением .qsys, содержащего настройки модуля АЦП в проекте ПЛИС (см. рис. 21). В нашем случае задаём имя файла ADC_Core_1 (имя можно выбрать произвольно), указываем путь к нему в пределах каталога нашего проекта ПЛИС и нажимаем на кнопку ОК, после чего автоматически запускается редактор IP-ядер IP Parameter Editor. В открывшемся окне настроек модуля АЦП, в редакторе IP Parameter Editor (см. рис. 22) задаём все необходимые параметры модуля АЦП:

- вариант IP-ядра Core Variant (в нашем случае – вариант ядра только с управлением АЦП без функций упорядочения и автосохранения оцифрованных выборок – ADC Control core only);
- наличие отладочных путей Debug Path (отсутствуют – Disabled);
- значение частоты входного синхросигнала АЦП ADC Input Clock (10 МГц);
- источник опорного напряжения АЦП Reference Voltage Source (внутренний ИОН АЦП ПЛИС – Internal);
- значение напряжения внутреннего ИОН Internal Reference Voltage (3,3 В).

Далее в разделе Channels задаём, какие каналы АЦП (выделенные, универсальные, встроенный температурный датчик TSD) будут включены, при необходимости задаём функцию входного делителя напряжения для тех каналов, которые её поддерживают. Задав все требуемые параметры АЦП, нажимаем на кнопку Generate HDL. После этого откроется окно задания начальных условий генерации HDL-кода: языка (выбираем Verilog), наличия модели для симуляции (выбираем None) и пути к выходным файлам HDL-кода (задаём директорию ADC_Core_1, находящуюся внутри директории нашего проекта ПЛИС). После задания всех перечисленных условий нажимаем на кнопку Generate и запускаем процесс генерации в открывшемся окне Generate. После корректного завершения генерации HDL-кода в этом окне выдаётся соответствующее сообщение: «Generate: completed successfully». Нажатие на кнопку Close приводит к открыванию окна с инструкциями по включению сгенерированного HDL-кода в наш проект ПЛИС (см. рис. 23). Согласно порядку действий, указанному в инструкции, необходимо вручную включить в состав проекта ПЛИС созданный на этапе генерации файл ADC_Core_1.qip, путь к которому показан на рисунке 23. Для этого во вкладке Files области навигации Project Navigator проекта ПЛИС нажатиями сначала правой, а потом левой кнопки мыши активируем команду добавления/удаления файлов из проекта ПЛИС Add/Remove Files in project. С помощью указанной команды добавляем в проект файл ADC_Core_1.qip, после чего во вкладке Files области навигации Project Navigator нашего проекта ПЛИС появляются файл ADC_Core_1.qip и несколько производных от него исходных файлов более низкого уровня иерархии (см. рис. 24). Каждый из этих файлов содержит исходный код, реа-

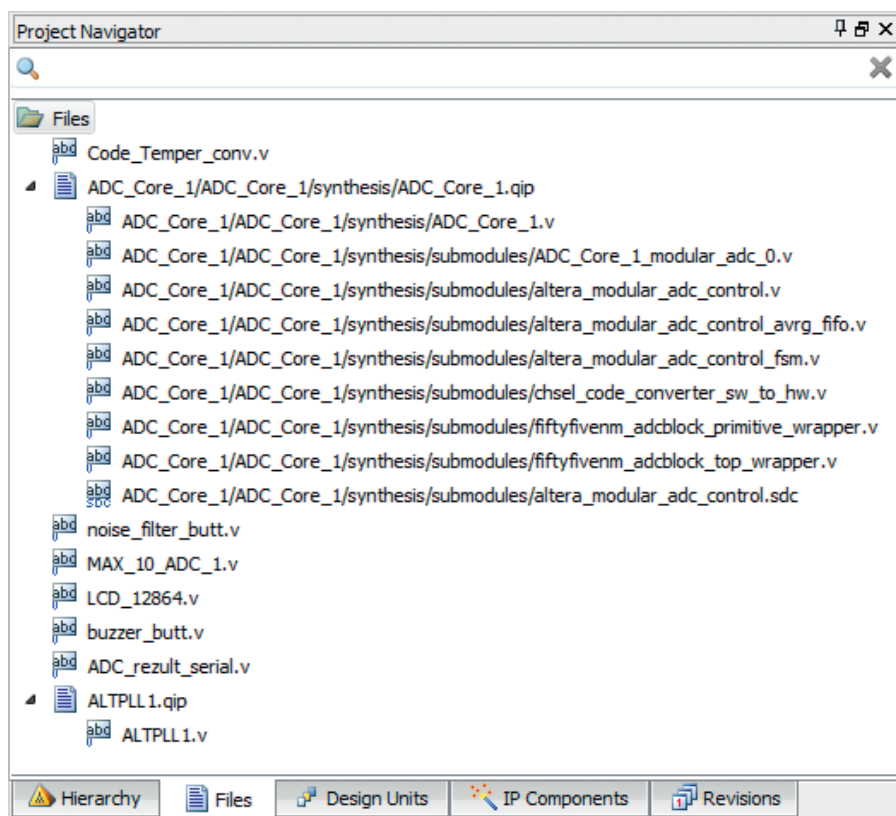


Рис. 24. Добавленные в проект ПЛИС исходные файлы IP-ядра АЦП

Листинг 3

```
module ADC_Core_1 (
    input wire      adc_pll_clock_clk,      // adc_pll_clock.clk
    input wire      adc_pll_locked_export,  // adc_pll_locked.export
    input wire      clock_clk,              // clock.clk
    input wire      command_valid,          // command.valid
    input wire [4:0] command_channel,       // .channel
    input wire      command_startofpacket,  // .startofpacket
    input wire      command_endofpacket,    // .endofpacket
    output wire     command_ready,          // .ready
    input wire      reset_sink_reset_n,     // reset_sink.reset_n
    output wire     response_valid,         // response.valid
    output wire [4:0] response_channel,     // .channel
    output wire [11:0] response_data,       // .data
    output wire     response_startofpacket, // .startofpacket
    output wire     response_endofpacket    // .endofpacket
);
```

лизирующий обслуживание какой-либо функции аппаратного модуля АЦП: взаимодействие модуля АЦП с шиной или с памятью, перекодировку результата АЦП и т.п. Нас среди этих файлов интересует только один – файл ADC_Core_1.v, содержащий модуль верхнего уровня IP-ядра АЦП. Имя и список входов и выходов этого модуля соответствуют ранее заданным нами параметрам АЦП (см. листинг 3).

Как и в случае с PLL, само по себе добавление IP-ядра АЦП в проект ПЛИС автоматически не обеспечивает его использование в проекте. Для его использования в исходный текст модуля верхнего уровня проекта ПЛИС (файл MAX_10_ADC_1.v) необходимо вручную включить фрагмент кода, устанавливающий связь между сигналами

модуля верхнего уровня проекта и модуля АЦП (см. листинг 4).

После включения в проект ПЛИС модуля АЦП во вкладке IP Components области навигации Project Navigator проекта ПЛИС появится позиция IP-ядра АЦП (см. рис. 25). На этом добавление IP-ядра модуля АЦП в проект ПЛИС можно считать завершённым.

Теперь можно запустить компиляцию проекта ПЛИС с АЦП, выбрав в главном меню Quartus II Processing → Start Compilation. В результате корректного завершения компиляции проекта ПЛИС вид открытых окон в среде Quartus II будет примерно соответствовать рисунку 26.

В нижней части окна (в окне сообщений Messages) должна содержаться итоговая информация об отсутствии ошибок компиляции (0 errors). В автома-

Листинг 4

```
// Модуль АЦП верхнего уровня
ADC_Core_1 ADC_Core_Core_1(
    .adc_pll_clock_clk(clk_pll), // вход синхросигнала АЦП
    .adc_pll_locked_export(locked_pll), // вход запирающего PLL, поступающий с соответствующего выхода PLL
    .clock_clk(sys_clk), // вход глобального синхросигнала
    .command_valid(command_valid_ADC), // вход валидности команды (активный уровень высокий)
    .command_channel(command_channel_ADC), // входы задания канала - пятиразрядное слово [4:0]
    .command_startofpacket(command_startofpacket_ADC), // вход разрешения преобразований (активный уровень высокий)
    .command_endofpacket(command_endofpacket_ADC), // вход запрещения преобразований (активный уровень высокий)
    .command_ready(command_ready_ADC), // выход готовности результата преобразования (активный уровень высокий)
    .reset_sink_reset_n(sys_rst_n), // вход сброса модуля АЦП (активный уровень низкий)
    .response_valid(response_valid_ADC), // выход возврата сигнала валидности (активный уровень высокий)
    .response_channel(response_channel_ADC), // выходы возврата входов задания канала - пятиразрядное слово [4:0]
    .response_data(dout_ADC), // выходы возврата данных преобразования - двенадцатизрядное слово [11:0]
    .response_startofpacket(response_startofpacket_ADC), // выход возврата сигнала разрешения преобразований (активный уровень высокий)
    .response_endofpacket(response_endofpacket_ADC) // выход возврата сигнала запрещения преобразований (активный уровень высокий)
);
```

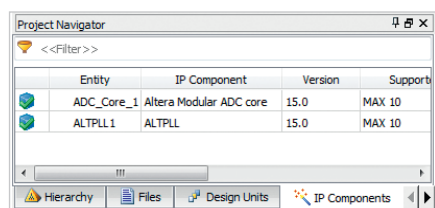


Рис. 25. IP-ядра PLL и АЦП в составе проекта ПЛИС

тически открывшемся окне Compilation Report отображается информация о задействованных в проекте ресурсах ПЛИС в абсолютном и процентном

отношениях. Как можно видеть из содержания Compilation Report, в нашем проекте ПЛИС задействованы все имеющиеся в этой ПЛИС модули PLL и ADC (100%).

Далее работа с проектом ПЛИС с АЦП не отличается от работы с любым другим проектом ПЛИС без АЦП. В частности, необходимо определить для всех цифровых линий ввода/вывода модуля верхнего уровня проекта соответствующие им выводы ПЛИС. Для этого в главном меню Quartus II нужно выбрать Assignments → Pin Planner и в открывшемся одноименном окне

задать для каждой линии ввода/вывода вывод ПЛИС в соответствии с принципиальной схемой макета (см. рис. 10). Кроме этого, для всех линий необходимо задать в столбце I/O Standard значение «3.3-V LVTTL», а в столбце Current Strength – значение «8mA(default)». Все остальные параметры задействованных в проекте выводов ПЛИС можно оставить заданными по умолчанию. Необходимо заметить, что таким образом определяются только цифровые линии проекта. Разрешение для использования выделенных и универсальных ана-

advantex UNO-2XM 100 кГц – 21 ГГц

Малошумящий синтезатор частот

Уровень фазового шума: -140 дБн/Гц
при отстройке 10 кГц @1 ГГц

Время перестройки: < 60 мкс

Шаг перестройки: 0.0001 Гц

Мощность выходного сигнала:
-5..+15 дБм, шаг 0,5 дБ

Опорный сигнал: 1-250 МГц

Выход опорной частоты: 10/100 МГц

Рабочий диапазон: -40...+60 °C

Габаритные размеры:
30,5 × 87,5 × 195,0 мм

Сделано в России
www.advantex.ru

Реклама

логовых входов АЦП в ПЛИС нами уже было выполнено ранее, при настройке IP-ядра АЦП в разделе Channels редактора Parameter Editor (см. рис. 22).

Далее необходимо включить на цифровых входах ПЛИС, используемых в нашем макете для подключения к плюсу питания резисторы. Для этого в главном меню Quartus II нужно выбрать Assignments → Assignments Editor и в открывшемся одноименном окне добавить к имеющимся там позициям ещё шесть позиций: key[5]–key[0] для каждого входа подключения кнопок (см. рис. 27). В столбце Assignment Name каждой из этих позиций необходимо задать значение «Weak Pull-Up Resistor», а в столбце Value – значение «On», как показано на рисунке.

После всех перечисленных действий следует ещё раз откомпилировать проект ПЛИС, чтобы учесть в создаваемой конфигурации все сделанные изменения. В результате корректной компиляции будет сгенерирован файл, который подлежит записи в конфигурационную память ПЛИС.

Запись конфигурации в ПЛИС

Выходной файл конфигурации откомпилированного проекта ПЛИС должен быть записан в специальную энергонезависимую конфигурационную память, которая, как правило, представляет собой отдельную микросхему. Физически эту микросхему следует располагать непосредственно рядом с ПЛИС и подсоединять к ней через специальный цифровой интерфейс. Сразу после подачи питания на ПЛИС и микросхему конфигурационной памяти в целевом устройстве производится автоматическая загрузка конфигурации из памяти в ПЛИС, после чего сконфигурированная ПЛИС начинает функционировать. Возможно также состояние, когда в саму ПЛИС конфигурация загружена, а её конфигурационная память пуста. В этом случае конфигурация в ПЛИС загружается сторонними средствами и сохраняется только до тех пор, пока на неё подано питание.

Все ПЛИС семейства MAX 10 не требуют наличия внешней памяти, поскольку содержат встроенную энергонезависимую конфигурационную память. В связи с этим запись конфигурации в конфигурационную память ПЛИС MAX 10 имеет некоторую специфику.

Программную поддержку записи конфигурации в конфигурационную память и в саму ПЛИС семейства MAX 10

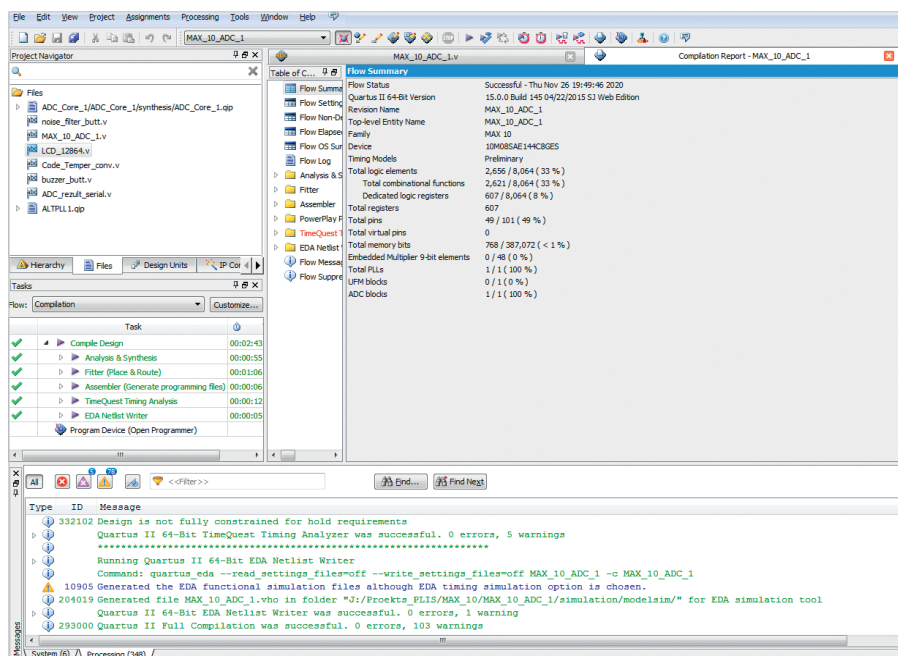


Рис. 26. Корректное завершение компиляции проекта ПЛИС

	tatu	From	To	Assignment Name	Value	Enabled	Entity	Comment
90	✓		OUT led_2	I/O Standard	3.3-V LVTTTL	Yes	MAX_10_ADC_1	
91	✓		OUT led_1	I/O Standard	3.3-V LVTTTL	Yes	MAX_10_ADC_1	
92	✓		OUT clk_OUT_ADC	I/O Standard	3.3-V LVTTTL	Yes	MAX_10_ADC_1	
93	✓		IN key[5]	I/O Standard	3.3-V LVTTTL	Yes	MAX_10_ADC_1	
94	✓		IN key[4]	I/O Standard	3.3-V LVTTTL	Yes	MAX_10_ADC_1	
95	✓		IN key[3]	I/O Standard	3.3-V LVTTTL	Yes	MAX_10_ADC_1	
96	✓		IN key[2]	I/O Standard	3.3-V LVTTTL	Yes	MAX_10_ADC_1	
97	✓		IN key[1]	I/O Standard	3.3-V LVTTTL	Yes	MAX_10_ADC_1	
98	✓		IN key[0]	I/O Standard	3.3-V LVTTTL	Yes	MAX_10_ADC_1	
99	✓		IN key[5]	Weak Pull-Up Resistor	On	Yes	MAX_10_ADC_1	
100	✓		IN key[4]	Weak Pull-Up Resistor	On	Yes	MAX_10_ADC_1	
101	✓		IN key[3]	Weak Pull-Up Resistor	On	Yes	MAX_10_ADC_1	
102	✓		IN key[2]	Weak Pull-Up Resistor	On	Yes	MAX_10_ADC_1	
103	✓		IN key[1]	Weak Pull-Up Resistor	On	Yes	MAX_10_ADC_1	
104	✓		IN key[0]	Weak Pull-Up Resistor	On	Yes	MAX_10_ADC_1	
105		<<new>>	<<new>>	<<new>>				

Рис. 27. Разрешение внутренних подтягивающих резисторов на входах ПЛИС

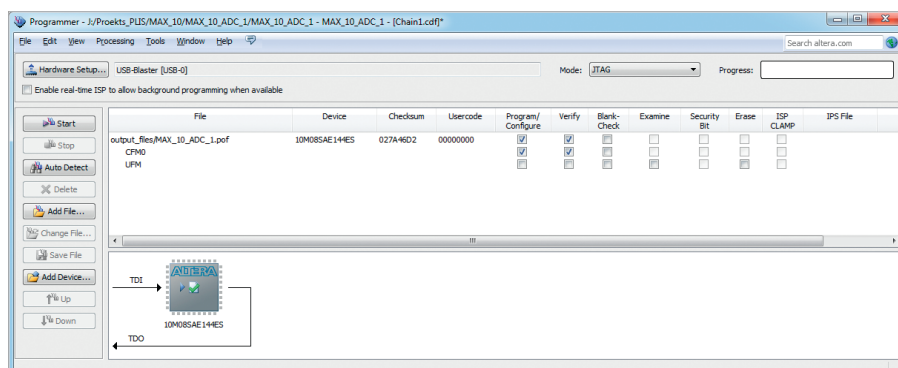


Рис. 28. Загрузка конфигурации в ПЛИС MAX 10

обеспечивает встроенный в среду Quartus II инструмент программирования (Programmer), запустить который можно, выбрав в главном меню Tools → Programmer. В открывшемся одноименном окне необходимо задать настройки, показанные на рисунке 28.

Физически запись в конфигурационную память и в ПЛИС производится с помощью аппаратного программатора USB-Blaster [2]. Автор использовал один

из недорогих клонов этого устройства китайского производства, которые во множестве доступны на рынке. Программатор USB-Blaster следует подключить к ПК через порт USB, а также подключить к целевой ПЛИС через интерфейс JTAG (разъем J10 на плате Altera MAX 10 FPGA). Подключение к ПК необходимо выполнить до запуска инструмента программирования Programmer среды Quartus II. На целевую ПЛИС (отладочную пла-



ВАШ ИНФОРМАЦИОННЫЙ ПОПУТЧИК!

Полосковые дисплеи для транспорта

- ЖК-дисплеи серии SPANPIXEL™ с яркостью до 3000 кд/м²
- Размеры по диагонали от 6,2 до 65"
- Разрешение до 4K2K
- Угол обзора 178° (во всех плоскостях)
- Диапазон рабочих температур (некоторых моделей) –30...+85°C
- Возможна разработка под заказ
- Ресурс до 100 000 часов

ProCHIP
POWERED BY ProSOFT

ОФИЦИАЛЬНЫЙ ДИСТРИБЬЮТОР

АКТИВНЫЙ КОМПОНЕНТ ВАШЕГО БИЗНЕСА
(495) 232-2522 • INFO@PROCHIP.RU • WWW.PROCHIP.RU



Реклама

ту) до её загрузки должно быть подано питающее напряжение.

После запуска инструмента программирования Programmer он должен автоматически распознать подключённый к ПК аппаратный программатор USB-Blaster (в левом верхнем углу окна Programmer справа от кнопки Hardware Setup должна появиться надпись USB-Blaster [USB-0]), как показано на рисунке 28. В случае, если автоматического распознавания не произошло (присутствует надпись «No Hardware»), необхо-

димо в окне Programmer кликнуть на кнопке Hardware Setup и в открывшемся окне вручную задать для программирования подключённый к ПК аппаратный программатор – выбрать из всплывающего меню в поле Currently selected hardware позицию USB-Blaster [USB-0]. Дополнительная информация по вопросам подключения программатора к ПК содержится в [2].

Подготовив таким образом систему к записи, необходимо задать файл конфигурации для записи в конфигу-

рационную память и в ПЛИС. Для этого в окне Programmer нужно нажать на кнопку Add File... и в открывшемся окне выбрать требуемый файл. В нашем случае указанный файл по умолчанию находится в подкаталоге output_files и имеет имя MAX_10_ADC_1.pof. После его выбора в столбце Device верхней части окна Programmer появится обозначение целевой ПЛИС (10M08SAE144ES), далее отображаются аббревиатуры областей памяти, доступных для загрузки в ПЛИС: встроенная конфигурационная флеш-память (CFM), встроенная пользовательская флеш-память (UFM). Необходимо поставить галочки в соответствующих пунктах столбцов Program/Configure и Verify, как показано на рисунке 28. Показанное на рисунке расположение галочек означает, что запись конфигурационного файла MAX_10_ADC_1.pof будет произведена как в саму ПЛИС, так и в её встроенную конфигурационную флеш-память, а после окончания записи будет произведена верификация – проверка соответствия содержимого ПЛИС и содержимого памяти CFM содержимому файла. Пользовательская флеш-память UFM в нашем проекте ПЛИС не задействована.

Иногда требуется записать конфигурацию только в ПЛИС, не затрагивая её встроенную конфигурационную память, например при отладке проекта ПЛИС в целях экономии ресурса циклов стирания-записи памяти CFM. В этом случае в качестве файла конфигурации следует выбрать файл с расширением *.sof. Такой файл (MAX_10_ADC_1.sof) также по умолчанию создаётся в подкаталоге output_files нашего проекта ПЛИС. При выборе этого файла загрузка в CFM и UFM становится недоступной.

Для запуска процесса записи с заданными параметрами необходимо нажать на кнопку Start. При заданных на рисунке 28 параметрах запись вместе с верификацией выполняется достаточно долго – несколько минут. Ход и результаты процесса записи отображаются в виде нарастающей шкалы в области Progress окна Programmer.

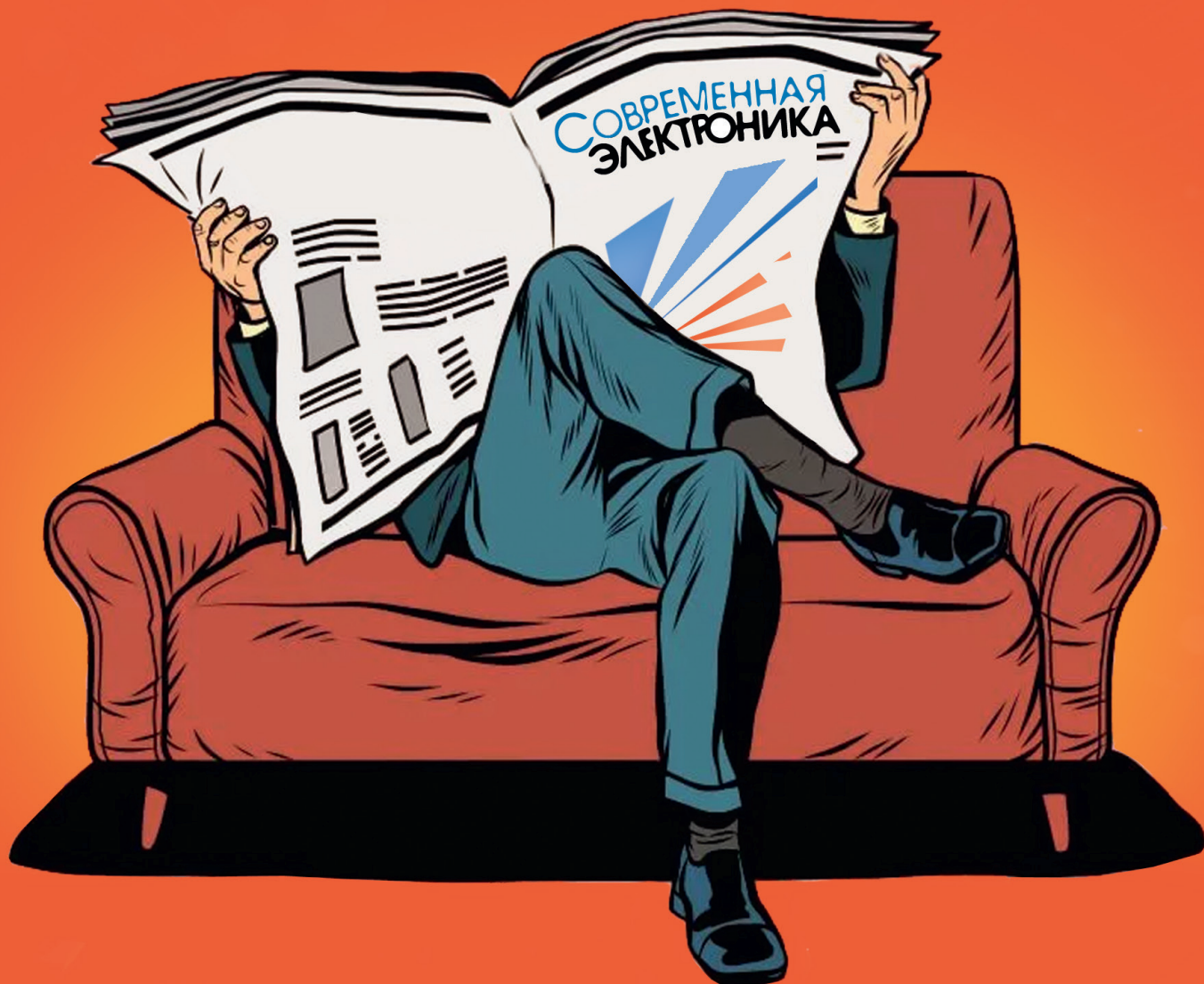
Литература

1. URL: https://www.intel.com/content/www/us/en/programmable/products/boards_and_kits/dev-kits/altera/kit-max-10-evaluation.html.
2. USB-Blaster Download Cable User Guide. UG-USB81204 2015.05.04. URL: https://www.intel.com/content/dam/altera-www/global/en_US/pdfs/literature/ug/archives/ug-usb-blstr-15.0.pdf.





ЧИТАЙТЕ КАК ВАМ УДОБНО



ПЕЧАТНУЮ

ПОДПИСКА С ГАРАНТИРОВАННОЙ ДОСТАВКОЙ

ИЛИ ЭЛЕКТРОННУЮ ВЕРСИЮ ЖУРНАЛА

**СОВРЕМЕННАЯ
ЭЛЕКТРОНИКА**

**ПОДПИСКА
НА ЖУРНАЛ**

онлайн: www.soel.ru • +7 495 232-0087 • info@soel.ru
на почте: по каталогу «Урал-пресс» (индекс для печатной версии 36280)