

Программирование последовательных конфигурационных ПЗУ

фирмы Altera по JTAG-интерфейсу

Для упрощения процесса отладки фирма Altera предлагает метод программирования последовательных конфигурационных ПЗУ с помощью JTAG-интерфейса. Этот метод можно использовать и для загрузки конфигурации в FPGA, и для программирования последовательных конфигурационных ПЗУ.

Виктор Березин

bw73@mail.ru

Роман Золотухо

roman@efo.ru

Микросхемы FPGA фирмы Altera выпускаются по технологии SRAM, в процессе работы конфигурация FPGA хранится в ячейках ОЗУ. Поскольку конфигурационное ОЗУ является энергозависимым, его необходимо загружать каждый раз после включения питания.

В процессе разработки и отладки менять конфигурацию FPGA приходится часто, для этого, как правило, используют загрузку по JTAG-интерфейсу с помощью специальных аппаратных средств — загрузочных кабелей.

После завершения процесса разработки и отладки автономного устройства на основе FPGA необходимо обеспечить его энергонезависимость. Для конфигурации FPGA в автономных устройствах используются два способа: загрузка из внешнего параллельного ПЗУ или микропроцессора (режим Passive Parallel) и загрузка из последовательного конфигурационного ПЗУ (режимы Passive serial и Active Serial) [1]. Фирма Altera выпускает последовательные конфигурационные ПЗУ, поддерживающие различные режимы конфигурации. Выпускаются и однократно программируемые, и репрограммируемые конфигурационные ПЗУ. Репрограммируемые конфигураторы могут быть прошиты непосредственно на плате целевого устройства по последовательному интерфейсу.

Все семейства FPGA фирмы Altera поддерживают режим конфигурации Passive Serial. В процессе отладки устройств этот режим может эмулироваться любым загрузочным кабелем фирмы Altera. Источником тактового сигнала в этом режиме является конфигурационное ПЗУ или загрузочный кабель. Для обеспечения этого режима можно использовать следующие конфигурационные ПЗУ: EPC1441, EPC1, EPC2, EPC4, EPC8, EPC16. Последние три микросхемы образуют семейство Enhanced-конфигураторов — кроме режима Passive Serial эти микросхемы поддерживают также режим Fast Passive Parallel.

Микросхемы EPC1441 и EPC1 являются однократно программируемыми (One-Time Programmable, OTP). Для прошивки этих микросхем служит автономный программатор ASAP2.

Микросхемы EPC2, EPC4, EPC8, EPC16 — репрограммируемые. Они могут быть запрограммированы

непосредственно в целевом устройстве по JTAG-интерфейсу.

Фирма Altera предлагает несколько загрузочных кабелей для конфигурации FPGA и внутрисхемного программирования конфигурационных ПЗУ — MasterBlaster, USB Blaster, ByteBlaster MV и ByteBlaster II. Эти устройства различаются по способу подключения к персональному компьютеру и обладают различными возможностями.

Новые семейства FPGA фирмы Altera — Cyclone, Cyclone II и Stratix II — кроме режимов Passive Parallel и Passive Serial поддерживают еще один режим конфигурации — Active Serial. Режим Active Serial поддерживается последовательными конфигурационными ПЗУ EPCS1, EPCS4, EPCS16, EPCS64. Невысокая цена этих конфигураторов делает их применение весьма оправданным, особенно при серийном выпуске устройств. Далее в статье под «последовательными конфигурационными ПЗУ» будем понимать именно эти конфигураторы с интерфейсом Active Serial.

Режим Active Serial имеет две отличительные особенности:

- источником тактового сигнала конфигурации (DCLK) является FPGA;
- конфигурационное ПЗУ программируется через те же выводы, через которые оно загружает проект в FPGA.

Последовательные конфигурационные ПЗУ с интерфейсом Active Serial могут быть запрограммированы в целевом устройстве с помощью загрузочных кабелей USB Blaster или ByteBlaster II или с помощью внешнего микропроцессора, реализующего программно интерфейс Active Serial (фактически это хорошо известный интерфейс SPI).

Однако применение последовательных конфигурационных ПЗУ со своим собственным интерфейсом программирования вносит в процесс отладки ряд неудобств. Разработчик вынужден либо каждый раз перепрограммировать конфигуратор для того, чтобы загрузить в FPGA отлаживаемый проект, либо использовать два разных интерфейса: JTAG для загрузки отлаживаемого проекта в FPGA и Active Serial для программирования последовательного конфигурационного ПЗУ.

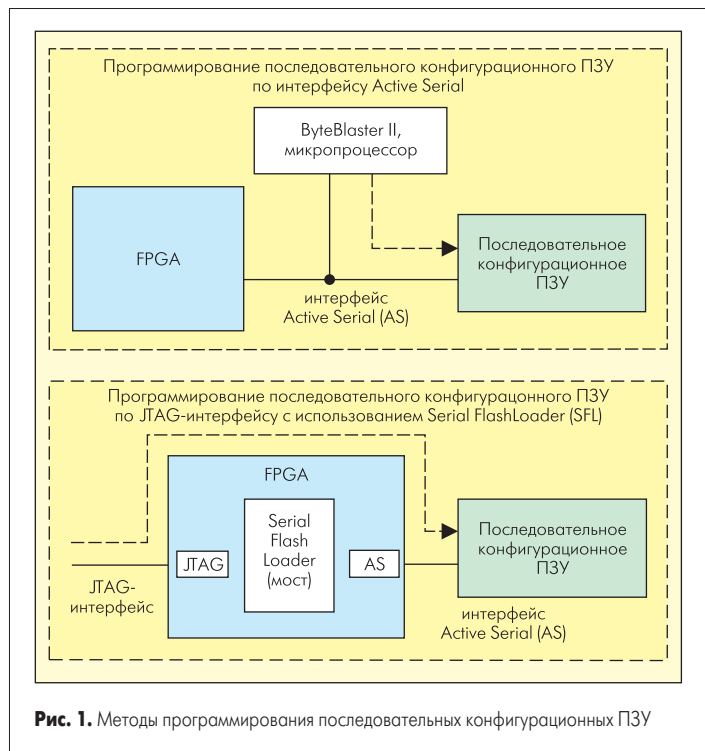


Рис. 1. Методы программирования последовательных конфигурационных ПЗУ

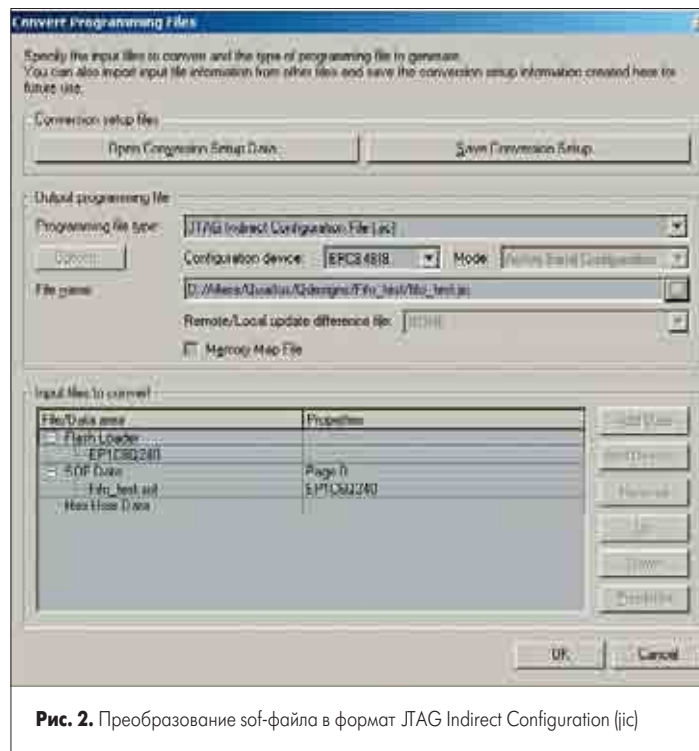


Рис. 2. Преобразование sof-файла в формат JTAG Indirect Configuration (.jic)

Таблица. Методы программирования последовательных конфигурационных ПЗУ

Интерфейс программирования	Преимущества	Недостатки
Active Serial	Простой скоростной последовательный интерфейс	1. Конфигурация FPGA и программирование ПЗУ происходят по различным интерфейсам (два разъема). 2. Выбор загрузочных кабелей для программирования ПЗУ ограничен.
JTAG (с использованием Serial FlashLoader)	1. Конфигурация FPGA и программирование ПЗУ происходят по одному и тому же интерфейсу (один разъем). 2. Для программирования ПЗУ можно использовать любой загрузочный кабель фирмы Altera.	Процесс программирования требует больше времени (по сравнению с Active Serial), так как перед программированием ПЗУ необходимо загрузить в FPGA SFL-образ.

Чтобы упростить процесс отладки, фирма Altera предлагает альтернативный метод программирования последовательных конфигурационных ПЗУ с интерфейсом Active Serial. Этот метод заключается в использовании JTAG-интерфейса и для загрузки конфигурации в FPGA, и для программирования последовательных конфигурационных ПЗУ [2].

Суть метода заключается в следующем: для программирования конфигурационного ПЗУ в FPGA по JTAG-интерфейсу загружается специально подготовленный проект — Serial FlashLoader (SFL). SFL представляет собой мост между интерфейсами JTAG и Active Serial и преобразует поток данных, поступающих по JTAG-интерфейсу, в формат, необходимый для программирования последовательного конфигурационного ПЗУ.

Применение этого метода позволяет использовать для программирования последовательных конфигурационных ПЗУ любой загрузочный кабель фирмы Altera, а также несколько упростить разрабатываемое устройство (поскольку отпадает необходимость устанавливать отдельный разъем для программирования конфигурационного ПЗУ).

Поддержка такого метода программирования последовательных конфигурационных ПЗУ осуществляется в САПР Quartus II начиная с версии 4.1 Service Pack 1.

Оба метода программирования последовательных конфигурационных ПЗУ представлены на рис. 1.

Преимущества и недостатки обоих методов программирования последовательных конфигурационных ПЗУ представлены в таблице.

Процесс программирования последовательных конфигурационных ПЗУ по JTAG-интерфейсу включает три этапа.

Этап 1. Создание файла для программирования по JTAG-интерфейсу.

Для программирования последовательных конфигурационных ПЗУ по JTAG-интерфейсу используются файлы в форматах JTAG Indirect Configuration (.jic) или JAM Standard Test And Programming Language, STAPL (.jam).

Jic-файл формируется в САПР Quartus II путем преобразования файла для конфигурации FPGA — SRAM Object File (.sof).

Преобразование происходит следующим образом:

1. В меню «File» выбирается пункт «Convert Programming Files».
2. В открывшемся окне (рис. 2) в поле «Programming file type» выбирается формат «JTAG Indirect Configuration File (.jic)».
3. В поле «Configuration device» из списка выбирается используемая в устройстве микросхема последовательного конфигурационного ПЗУ.
4. В поле «File name» вводится имя выходного jic-файла.
5. В секции «Input file to convert» активизируется раздел «SOF Data» и указывается входной sof-файл для преобразования (кнопка «Add File»). При этом в разделе «SOF Data»

отображается тип целевой микросхемы FPGA (информация о нем содержится в sof-файле).

6. В секции «Input file to convert» активизируется раздел «Flash Loader» и выбирается используемая в устройстве микросхема FPGA (кнопка «Add Device»).

7. После того как выполнены все установки, запускается процесс преобразования (кнопка «Ok»).

При желании разработчик может преобразовать jic-файл в формат JAM STAPL. Для преобразования нужно сделать следующее:

1. В САПР Quartus II открыть окно программатора (меню «Tools», пункт «Programmer»).
2. Выбрать jic-файл в качестве входного (кнопка «Add File»).
3. В меню «File» выбрать пункт «Create/Update», затем во всплывающем меню выбрать «Create JAM, SVF or ISC File».

Окно создания jam-файла представлено на рис. 3.

Структура полученных в результате таких преобразований jic- или jam-файлов представлена на рис. 4. Файл состоит из двух секций. Первая содержит Serial FlashLoader (или SFL-



Рис. 3. Преобразование jic-файла в формат JAM STAPL

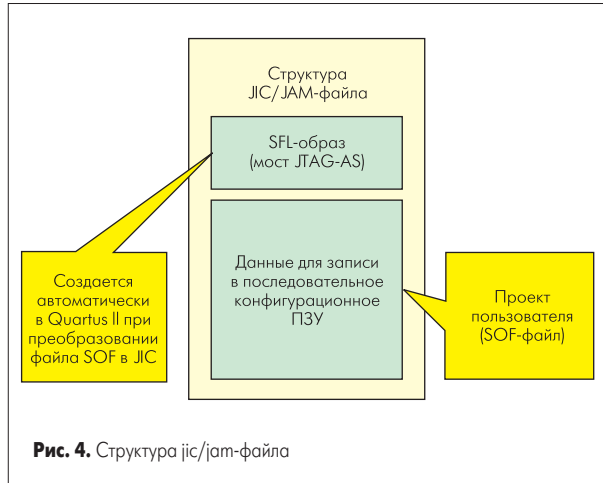


Рис. 4. Структура jic/jam-файла

образ) — мост между интерфейсами JTAG и Active Serial. SFL-образ создается в пакете Quartus II автоматически в процессе преобразования sof-файла в формат JTAG Indirect Configuration (jic). Вторая секция содержит блок данных для прошивки последовательного конфигурационного ПЗУ, который переносится в jic-файл из sof-файла.

Рассмотрим ситуацию, когда в устройстве используется несколько микросхем FPGA (каждая со своим собственным конфигурационным ПЗУ), которые образуют JTAG-цепочку. В этом случае для каждой FPGA нужно создать ее собственный jic-файл. Если разработчик желает использовать формат JAM STAPL, то при преобразовании он может объединить jic-файлы для всех микросхем цепочки в один выходной jam-файл, который позволит запрограммировать все последовательные конфигурационные ПЗУ, подключенные к микросхемам FPGA, образующим цепочку.

Этап 2. Загрузка SFL-образа в FPGA и программирование последовательного конфигурационного ПЗУ.

Для прошивки последовательных конфигурационных ПЗУ по JTAG-интерфейсу с использованием jic-файла нужно воспользоваться программатором из состава пакета Quartus II (меню «Tools», пункт «Programmer»). В окне программатора нужно открыть jic-файл (кнопка «Add File»). При этом будут отображены типы целевых FPGA и конфигурируемого ПЗУ.

Операции «Program/Configure» в окне программатора должны быть активизированы для обоих устройств — и FPGA, и последовательного конфигурационного ПЗУ. Если микросхема FPGA не будет сконфигурирована как мост между интерфейсами JTAG и Active Serial (то есть не будет содержать SFL-образа), то попытка запрограммировать последовательное конфигурационное ПЗУ будет безуспешной.

Для того чтобы запрограммировать конфигурационные ПЗУ для нескольких FPGA, образующих JTAG-цепочку, надо в окне программатора открыть jic-файлы для соответствующих микросхем цепочки и активизировать операции «Program/Configure» для каждого файла.

Если в программаторе пакета Quartus II открыть jam-файл, то активизация режима «Program/Configure» для этого файла автоматически вызовет нужную последовательность действий — сначала произойдет загрузка

образа SFL в FPGA, затем программирование конфигурируемого ПЗУ (другими словами, активизировать процессы загрузки и программирования по отдельности в этом случае не нужно).

При использовании jam-файла разработчик может пользоваться не только программатором пакета Quartus II, но и любым JAM-плеером, поддерживающим STAPL (такой плеер может быть реализован как на персональном компьютере, так и во встраиваемом микропроцессоре на целевой плате).

На рис. 5 представлена ситуация, когда в целевом устройстве имеется JTAG-цепочка из двух FPGA — EP1C20 и EP1C6. Для прошивки конфигурируемого ПЗУ первой из этих FPGA сформирован jam-файл, для второй — jic-файл. Для того чтобы изменить содержимое конфигурационного ПЗУ, подключенного к EP1C6 (в нашем случае это EPCS4SI8), нужно активизировать операции «Program/Configure» только для jic-файла (в этом случае при выполнении программирования микросхема EP1C20 будет «прозрачной» для JTAG-интерфейса).

Этап 3. Реконфигурация FPGA (загрузка нового проекта, прошитого в конфигурационное ПЗУ)

Для того чтобы после окончания процесса программирования последовательного конфигурационного ПЗУ загрузить новый про-

ект в FPGA, нужно на вход nCONFIG подать импульс низкого логического уровня либо просто выключить и включить питание устройства. При этом будет активизирован процесс конфигурации в режиме Active Serial, и проект, записанный в ПЗУ, загрузится в FPGA, заменяя находящийся там в момент программирования ПЗУ Serial FlashLoader.

Схема включения FPGA семейств Cyclone и Cyclone II и последовательного конфигурационного ПЗУ представлена на рис. 6. Состояние входов MSEL[1..0] FPGA определяет режим конфигурации Active Serial. Во время загрузки SFL-образа в FPGA состояние входов MSEL[1..0] игнорируется, так как загрузка конфигурации в FPGA по JTAG-интерфейсу имеет наивысший приоритет. После завершения процесса программирования новый проект будет загружаться в FPGA в режиме Active Serial.

Литература

1. Altera Configuration Handbook. http://www.altera.com/literature/hb/cfg/config_handbook.pdf.
2. Using the Serial FlashLoader With the Quartus II Software. Altera Application Note 370. <http://www.altera.com/literature/an/an370.pdf>.

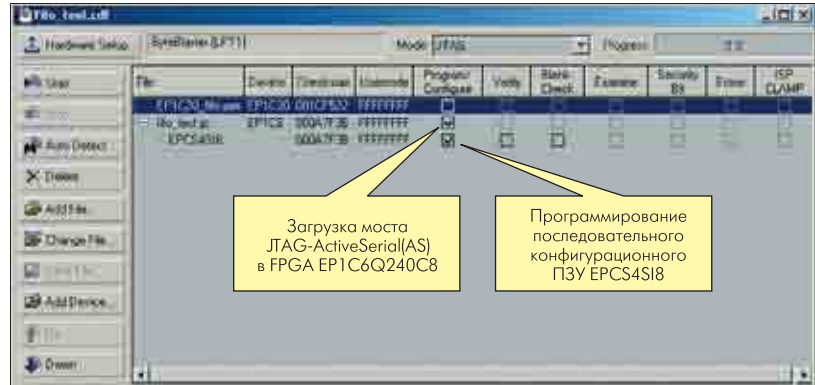


Рис. 5. Программирование последовательных конфигурационных ПЗУ по JTAG-интерфейсу в пакете Quartus II (FPGA образуют JTAG-цепочку)

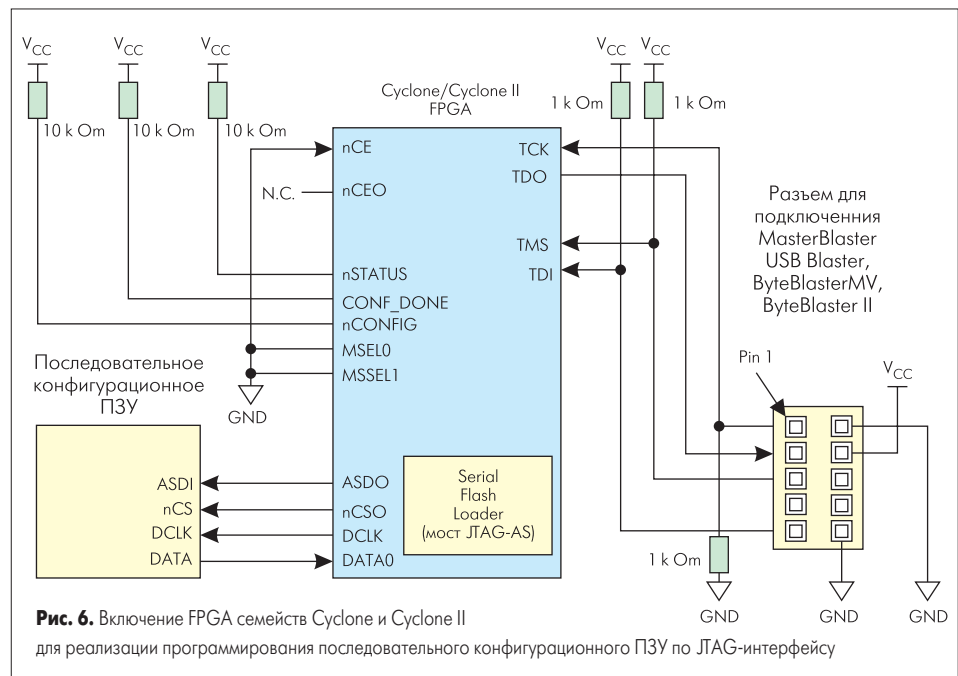


Рис. 6. Включение FPGA семейств Cyclone и Cyclone II для реализации программирования последовательного конфигурационного ПЗУ по JTAG-интерфейсу