

Практический курс сквозного проектирования цифровых устройств на основе ПЛИС фирмы Xilinx

(часть 25)

Валерий Зотов (Москва)

В двадцатой пятой части курса рассматриваются способы программирования микросхем флэш-памяти с параллельным и последовательным интерфейсом, выпускаемых различными производителями и поддерживаемых модулем *iMPACT* САПР серии Xilinx ISE. Описан процесс непосредственной загрузки конфигурационных данных в последовательные ППЗУ и обратного считывания информации через интерфейс SPI, а также программирования, осуществляемого через порт JTAG-интерфейса ПЛИС FPGA. Приводятся необходимые сведения о выполнении операций программирования конфигурационной флэш-памяти с параллельным интерфейсом BPI.

СПОСОБЫ ПРОГРАММИРОВАНИЯ МИКРОСХЕМ КОНФИГУРАЦИОННОЙ ФЛЭШ-ПАМЯТИ С ПАРАЛЛЕЛЬНЫМ И ПОСЛЕДОВАТЕЛЬНЫМ ИНТЕРФЕЙСОМ

Модуль программирования *iMPACT*, входящий в состав пакета САПР серии Xilinx ISE, позволяет не только загружать конфигурационную последовательность данных в ПЛИС и записывать соответствующую информацию в ПЗУ/ППЗУ, выпускаемые фирмой Xilinx, но и осуществлять программирование микросхем флэш-памяти с параллельным и последовательным интерфейсом от различных производителей. Для выполнения этих операций не требуется дополнительных аппаратных средств (программаторов) – достаточно одного из стандартных загрузочных кабелей, предназначенных для конфигурирования ПЛИС. В настоящее время в модуле программирования *iMPACT* предусмотрена возможность записи конфигурационных данных в микросхемы флэш-памяти с последовательным интерфейсом семейства AT45DB, выпускаемые фирмой Atmel, а также семейств MP25P, M25PE и MP45PE, производимых компанией STMicroelectronics (или фирмой Numonux, образованной в результате слияния отделившихся подразделений компаний Intel и STMicroelectronics).

Процесс программирования элементов конфигурационной флэш-памяти

с последовательным интерфейсом SPI может осуществляться двумя способами. При использовании первого метода (метода прямого программирования) загрузочный кабель подключается непосредственно к контактам порта SPI-интерфейса микросхемы ППЗУ. Второй способ (косвенный метод) заключается в применении ПЛИС, к которой подключен элемент конфигурационной флэш-памяти, в качестве своеобразного моста между JTAG-интерфейсом кристалла FPGA и интерфейсом SPI данного элемента запоминающего устройства. Для этой цели разработан и скомпилирован соответствующий проект (ядро). Этот проект автоматически загружается в ПЛИС при выборе соответствующего режима работы модуля программирования *iMPACT*. Для реализации косвенного метода программирования микросхем флэш-памяти с последовательным интерфейсом SPI используется режим периферийного сканирования *Boundary Scan* (JTAG) *Mode* модуля *iMPACT*.

Запись конфигурационной последовательности данных в микросхемы флэш-памяти с параллельным интерфейсом BPI может осуществляться с помощью модуля программирования *iMPACT* только косвенным методом. При его использовании в ПЛИС, к которой подключен элемент параллельного конфигурационного ППЗУ, автоматически загружается проект (ядро), реализующий функции моста

между JTAG-интерфейсом кристалла FPGA и интерфейсом BPI. Модуль программирования *iMPACT* позволяет осуществлять операции записи конфигурационных данных в микросхемы флэш-памяти с параллельным интерфейсом семейства 28F, выпускаемые фирмой Intel (или Numonux), и ППЗУ других производителей, полностью совместимые с элементами этого семейства.

Из всех семейств кристаллов с архитектурой FPGA, доступных к моменту подготовки данной публикации, косвенный метод программирования элементов конфигурационной флэш-памяти с последовательным и параллельным интерфейсом поддерживался ПЛИС семейств Spartan-3A, Spartan-3AN, Spartan-3A DSP, Virtex-5 LX, Virtex-5 LXT, Virtex-5 SXT, Virtex-5 FXT и Virtex-5 TXT. Кристаллы новых семейств Spartan-6 LX, Spartan-6 LXT, Virtex-6 LXT и Virtex-6 SXT, серийное производство которых начато в начале текущего года, также поддерживают этот способ программирования конфигурационных флэш-ППЗУ. Но при этом следует учитывать, что ПЛИС серий Virtex-6 и Spartan-6 поддерживаются только новой версией системы проектирования Xilinx ISE (Integrated Software Environment/Integrated Synthesis Environment) Design Suite 11, которая уже сейчас доступна разработчикам.

ПОДКЛЮЧЕНИЕ ЗАГРУЗОЧНОГО КАБЕЛЯ ДЛЯ ПРЯМОГО ПРОГРАММИРОВАНИЯ ЭЛЕМЕНТОВ КОНФИГУРАЦИОННОЙ ФЛЭШ-ПАМЯТИ С ПОСЛЕДОВАТЕЛЬНЫМ ИНТЕРФЕЙСОМ SPI И ОБРАТНОГО СЧИТЫВАНИЯ КОНФИГУРАЦИОННОЙ И СТАТУСНОЙ ИНФОРМАЦИИ

Для непосредственного программирования элементов конфигурационной флэш-памяти с последовательным

интерфейсом SPI и обратного считывания конфигурационной и статусной информации на печатной плате разрабатываемого устройства (или на отладочной плате) должен быть предусмотрен разъём, к которому подключаются выводы порта SPI-интерфейса используемого ППЗУ. Все загрузочные кабели, выпускаемые фирмой Xilinx, комплектуются переходным шлейфом с отдельными гибкими выводами, который удобно использовать для прямого программирования микросхем флэш-памяти с последовательным интерфейсом SPI.

Используемый загрузочный кабель JTAG-интерфейса с указанным переходным шлейфом следует подключить к свободному порту компьютера, для которого он предназначен, а затем гибкие выводы шлейфа присоединить к соответствующим контактам порта SPI-интерфейса используемого ППЗУ. Для программирования микросхем флэш-памяти, выпускаемых компанией Atmel, необходимо соединить сигнальный контакт TMS/PROG загрузочного кабеля с выводом /CS_N ППЗУ, контакт TDI/DIN – с выводом SI, контакт TDO/DONE – с выводом SO, контакт TCK/CCLK – с выводом SCK. Контакты GND/GND и VREF/VREF загрузочного кабеля следует подключить к выводам микросхем флэш-памяти GND и VCC соответственно. Если в качестве элемента конфигурационной памяти применяется микросхема фирмы STMicroelectronics (или Numonyx), то контакт загрузочного кабеля TDI/DIN должен быть присоединён к выводу S_N ППЗУ, контакт TDO/DONE – к выводу D, контакт TDO/DONE – к выводу Q, контакт TCK/CCLK – к выводу C. Выводы ПЛИС, подключенные к порту SPI-интерфейса программируемой микросхемы флэш-памяти, должны быть переведены в высокоимпедансное состояние при осуществлении операций непосредственной записи данных в ППЗУ и обратного считывания конфигурационной и служебной информации.

После выполнения указанных подключений необходимо подать напряжение питания на плату инструментального модуля, используемого для аппаратной отладки или реализации разрабатываемого устройства. Такая последовательность обеспечивает возможность автоматического обнаружения и инициализации загрузочного кабеля, а также конфигурационных ППЗУ непосредственно в процессе

функционирования программы *iMPACT*. При подключении загрузочного кабеля после запуска модуля программирования необходимо выполнить «вручную» операции установки типа и параметров используемого кабеля.

Активизация модуля *iMPACT* в режиме прямого программирования элементов конфигурационной флэш-памяти с последовательным интерфейсом SPI

Для запуска модуля программирования *iMPACT* в режиме прямого программирования элементов флэш-памяти с последовательным интерфейсом SPI необходимо вначале выполнить те же действия, что и при загрузке конфигурационной последовательности данных в ППЗУ фирмы Xilinx. Затем в открывшейся диалоговой панели с заголовком *Welcome to iPACT* следует переключить в нажатое состояние кнопку *Configure devices*. При этом становится доступным поле выбора режима программирования, расположенное справа от этой кнопки. В выпадающем списке необходимо выделить строку *Using Direct SPI Configuration*, которая соответствует режиму непосредственного программирования микросхем конфигурационной флэш-памяти с последовательным интерфейсом SPI. Выбор требуемого режима

работы модуля программирования завершается нажатием кнопки «Готово» (*Finish*) в нижней части диалоговой панели *Welcome to iMPACT*.

В результате выполнения указанных операций на экране появляется диалоговая панель с заголовком *Add Device*, в которой необходимо указать идентификатор и расположение на диске сформированного ранее файла с расширением *mcs* или *exo*, предназначенного для загрузки в выбранную микросхему флэш-памяти. Следует обратить внимание на то, что этот файл должен быть сгенерирован для последовательного ППЗУ с интерфейсом SPI. Эта процедура была подробно рассмотрена в 22-й части данного курса.

После определения названия требуемого файла программирования, записываемого в конфигурационное ППЗУ с интерфейсом SPI, в области расположения рабочих окон Навигатора проекта (*Project Navigator*) открывается новое окно, на закладке которого отображается название выбранного режима работы программы *iMPACT* – *Direct SPI Configuration*, и выводится диалоговая панель, озаглавленная *Select Device Part Name*. Кроме того, в окне исходных модулей *Sources Window* добавляется страница *Configuration Modes*, предназначенная для выбора режима работы средств программирования. С помощью поля выбора, представленного в диалоговой панели с заголовком *Select Device Part Name*, следует указать условное обо-

значение используемой микросхемы флэш-памяти. Выпадающий список этого поля выбора содержит обозначения всех ППЗУ с последовательным интерфейсом SPI, поддерживаемых модулем программирования *iMPACT*.

Выбранный тип микросхемы флэш-памяти следует подтвердить нажатием клавиши *OK*, расположенной в нижней части диалоговой панели с заголовком *Select Device Part Name*. При успешном выполнении указанных действий в рабочем окне программы *iMPACT – Direct SPI Configuration* появляется условный графический образ (УГО) выбранной микросхемы флэш-памяти с интерфейсом SPI, под которым отображается обозначение этого ППЗУ и идентификатор файла программирования, а в окне процессов *Processes Window* – страница *Configuration Operations*. Эта страница предоставляет доступ к соответствующим командам программирования ППЗУ, обратного считывания и проверки записанных данных.

В окне регистрации консольных сообщений *Transcript Window* управляющей оболочки Навигатора проекта выводится информация о выполнении последовательности команд установки режима работы программного модуля *iMPACT* для непосредственной записи данных в элемент флэш-памяти с интерфейсом SPI, выбора типа микросхемы ППЗУ и файла программирования. Вид этой информации приведен ниже на примере программирования микросхемы M25P16, выпускаемой компанией STMicroelectronics (Numonyx):

```
// *** BATCH CMD : setMode -spi
Selected part: M25P16
INFO:iMPACT:501 - '1': Added Device M25P16 successfully.
-----
'1': Loading file
'D:/Prj_n/Prj_direct_spi/direct_flash_spi.mcs' ...
INFO:iMPACT - Elapsed time = 0
sec.
done.
// *** BATCH CMD : setAttribute
-position 1 -attr readnextdevice
-value ""
```

Процедура автоматической идентификации типа подключенного загрузочного кабеля в режиме прямого программирования микросхем флэш-памяти с интерфейсом SPI осуществляется при первой выполняемой команде. Ход выполнения этой процедуры

отображается на экране дисплея с помощью всплывающего окна индикации и сопровождается соответствующей информацией в окне регистрации консольных сообщений *Transcript Window*. Протокол выполнения процедуры автоматического определения загрузочного кабеля, подключаемого к порту USB персонального компьютера, который применяется для программирования элементов флэш-памяти с интерфейсом SPI, выглядит следующим образом:

```
GUI --- Auto connect to cable...
// *** BATCH CMD : setCable -port
auto
AutoDetecting cable. Please wait.
PROGRESS_START - Starting Operation.
Connecting to cable (Usb Port -
USB21).
Driver file xusb_emb.sys found.
Driver version: src=1029,
dest=1029.
Driver windrvr6.sys version =
8.1.1.0. WinDriver v8.11 Jungo
(c) 1997 - 2006 Build Date: Oct
16 2006 X86 32bit SYS 12:35:07,
version = 811.
Cable PID = 0008.
Max current requested during enu-
meration is 60 mA.
Cable Type = 3, Revision = 0.
Setting cable speed to 6 MHz.
Cable connection established.
File version of
D:/Xilinx/10.1/ISE/data/xusb_xlp.
hex = 1302.
Downloading
D:/Xilinx/10.1/ISE/data/xusb_xlp.
hex.
Downloaded firmware version =
1302.
Type = 0x0004.
ESN option: 00000000000000.
PLD file version = 0012h.
PROGRESS_END - End Operation.
```

При неудачном завершении процедуры автоматической идентификации загрузочного кабеля, осуществляемой модулем программирования *iMPACT*, в окне регистрации консольных сообщений *Transcript Window* отображаются подробные сведения о последовательном сканировании всех доступных параллельных портов (LPT) и USB-портов компьютера, после чего выводится соответствующее предупреждение об отсутствии подключенного загрузочного кабеля. В этом слу-

чае необходимо проверить правильность подключения гибких выводов загрузочного кабеля к выводам разъёма порта интерфейса SPI микросхемы конфигурационной флэш-памяти, а также наличие напряжения питания.

ПРЯМОЕ ПРОГРАММИРОВАНИЕ МИКРОСХЕМ КОНФИГУРАЦИОННОЙ ФЛЭШ-ПАМЯТИ С ПОСЛЕДОВАТЕЛЬНЫМ ИНТЕРФЕЙСОМ SPI С ПОМОЩЬЮ ЗАГРУЗОЧНОГО JTAG-КАБЕЛЯ

Прежде чем приступить к записи конфигурационной последовательности данных в ППЗУ, рекомендуется проверить и установить требуемые параметры процесса программирования. Для этого следует выбрать команду *Set Programming Properties* из всплывающего меню *Edit* или из контекстно-зависимого всплывающего меню, открываемого щелчком правой кнопки мыши. При выполнении этой команды на экран выводится диалоговая панель с заголовком *Device Programming Properties*, которая предназначена для определения значений параметров процесса программирования микросхем флэш-памяти. В этой диалоговой панели представлено два параметра: *Verify* и *Erase Before Programming*.

С помощью параметра *Verify* разрешается или запрещается автоматическое выполнение операции проверки записанной информации по окончании процесса программирования ППЗУ с последовательным интерфейсом SPI. По умолчанию для этого параметра предлагается значение «включено», при котором осуществляется автоматический контроль данных, загруженных в элемент флэш-памяти в текущем сеансе программирования этой микросхемы.

Значение параметра *Erase Before Programming* определяет необходимость предварительной очистки содержимого ППЗУ с последовательным интерфейсом SPI перед осуществлением процесса программирования данной микросхемы. Значение «включено», установленное по умолчанию для этого параметра, соответствует режиму автоматического выполнения операции «стирания» конфигурационных данных, находящихся в ППЗУ перед его программированием.

После установки требуемых значений параметров процесса прямого программирования микросхем конфигурационной флэш-памяти с после-

довательным интерфейсом необходимо подтвердить их нажатием клавиши *OK*, которая расположена в нижней части диалоговой панели *Device Programming Properties*.

Для выполнения всех операций программирования ППЗУ с последовательным интерфейсом SPI и обратного чтения конфигурационных и статусных данных следует использовать соответствующие команды, которые представлены на странице *Configuration Operations* в окне процессов *Processes Window*, во всплывающем меню *Operations* управляющей оболочки Навигатора проекта и контекстно-зависимом всплывающем меню. Эти команды становятся доступными после выделения щелчком левой кнопки мыши УГО *SPI PROM*, представленного в рабочем окне *iMPACT*. Контекстно-зависимое всплывающее меню открывается щелчком правой кнопкой мыши при расположении курсора на условном изображении соответствующего элемента флэш-ППЗУ.

Процесс прямого программирования выбранной микросхемы конфигурационной флэш-памяти с последовательным интерфейсом SPI иници-

ируется с помощью команды *Program*. Ход выполнения процесса записи данных в ППЗУ отображается с помощью всплывающего окна индикации и сопровождается соответствующими уведомлениями в рабочем окне программы *iMPACT* и окне регистрации консольных сообщений *Transcript Window*. Ниже представлен протокол выполнения операции программирования микросхемы конфигурационной памяти M25P16 с последующим контролем записанной информации:

```
// *** BATCH CMD : Program -p 1
-e -v
Validating device...
'1': IDCODE is '202015' (in hex).
'1': ID Check passed.
Device validated successfully.
'1': Erasing Device.
PROGRESS_START - Starting Operation.
'1': Programming Flash.
'1': Reading device contents...
done.
'1': Verification completed.
PROGRESS_END - End Operation.
Elapsed time = 121 sec.
```

ПРЯМОЕ ЧТЕНИЕ КОНФИГУРАЦИОННЫХ ДАННЫХ И СТАТУСНОЙ ИНФОРМАЦИИ ИЗ МИКРОСХЕМ ФЛЭШ-ПАМЯТИ С ПОСЛЕДОВАТЕЛЬНЫМ ИНТЕРФЕЙСОМ SPI С ПОМОЩЬЮ МОДУЛЯ ПРОГРАММИРОВАНИЯ *iMPACT* И ЗАГРУЗОЧНОГО JTAG-КАБЕЛЯ

При подключении микросхем конфигурационной флэш-памяти с последовательным интерфейсом SPI непосредственно к разъёму JTAG-порта загрузочного кабеля модуль программирования *iMPACT* не только предоставляет возможность записи информации в ППЗУ, но и позволяет выполнять операции обратного считывания конфигурационных данных и их контроля, чтение информации о состоянии этого элемента, а также очистку содержимого запоминающего устройства. Для этого предусмотрены соответствующие команды *Readback*, *Verify*, *Get Device Checksum*, *Blank Check* и *Erase*. Эти команды имеют то же назначение, что и в случае применения ППЗУ фирмы Xilinx серий XC18V00 и *Platform Flash*, но процесс их выполнения отличается при использовании

микросхем флэш-памяти с последовательным интерфейсом SPI. Поэтому далее в настоящем разделе приводится информация о выполнении каждой из перечисленных выше команд.

Команда *Readback* предназначена для считывания конфигурационной информации, записанной в выбранный элемент флэш-памяти, и сохранения её в виде файла на диске в формате MCS. При активизации этой команды на экран выводится стандартная диалоговая панель, используемая для определения названия создаваемого файла, с заголовком *Save ReadbackFile*. Последующий ход процесса обратного считывания из ППЗУ записанной информации отображается с помощью соответствующей всплывающей панели индикации в рабочем окне программы *iMPACT* и следующих уведомлений в окне регистрации консольных сообщений *Transcript Window*:

```
// *** BATCH CMD : ReadbackToFile
-p 1 -file
"D:/Prj_n/Prj_direct_spi/old_flash_spi.mcs"
Validating device...
'1': IDCODE is '202015' (in hex).
'1': ID Check passed.
Device validated successfully.
'1': Performing readback on device...
PROGRESS_START - Starting Operation.
'1': Reaback completed successfully.
PROGRESS_END - End Operation.
Elapsed time = 186 sec.
```

Для проверки достоверности конфигурационных данных, содержащихся в микросхеме флэш-памяти с последовательным интерфейсом SPI, используется команда *Verify*. При выборе этой команды осуществляется чтение содержимого ППЗУ и сравнение считанных данных с информацией, записанной в указанном файле программирования. Результат выполнения этой команды отображается в виде соответствующей всплывающей панели индикации в рабочем окне программы *iMPACT* и следующей информации в окне регистрации консольных сообщений *Transcript Window*:

```
// *** BATCH CMD : Verify -p 1
Validating device...
'1': IDCODE is '202015' (in hex).
'1': ID Check passed.
Device validated successfully.
'1': Reading device contents...
PROGRESS_START - Starting Operation.
```

```
done.
'1': Verification completed.
PROGRESS_END - End Operation.
Elapsed time = 30 sec.
```

Команда *Get Device Checksum* инициирует операцию вычисления контрольной суммы конфигурационных данных, записанных во флэш-ППЗУ с последовательным интерфейсом SPI, и сравнения полученного результата со значением контрольной суммы файла «прошивки», который указан в рабочем окне программы *iMPACT* для выбранного элемента. Результаты вычисления контрольной суммы и сравнения с образцовым значением выводятся в окне регистрации консольных сообщений *Transcript Window* в следующем виде:

```
// *** BATCH CMD : Checksum -p 1
Validating device...
'1': IDCODE is '202015' (in hex).
'1': ID Check passed.
Device validated successfully.
'1': Reading device contents...
done.
All Checksum calculations are performed on the address range spanned by the configuration file.
'1': Calculated checksum matches expected checksum, 000012c06
```

С помощью команды *Blank Check* считывается информация о состоянии микросхемы конфигурационной флэш-памяти с последовательным интерфейсом SPI, подключенной к загрузочному кабелю. Полученные данные о статусе ППЗУ (содержит этот элемент памяти конфигурационную последовательность или нет) отображаются в виде всплывающей информационной панели в рабочем окне программы *iMPACT* и окне регистрации консольных сообщений *Transcript Window* управляющей оболочки Навигатора проекта. Если микросхема флэш-памяти не содержит конфигурационных данных (находится в незапрограммированном состоянии), то результат выполнения команды *Blank Check* выглядит следующим образом:

```
// *** BATCH CMD : BlankCheck -p 1
Validating device...
'1': IDCODE is '202015' (in hex).
'1': ID Check passed.
Device validated successfully.
'1': Performing blank check on device...
```

```
PROGRESS_START - Starting Operation.
'1': Part is blank.
PROGRESS_END - End Operation.
Elapsed time = 181 sec.
```

При наличии конфигурационной информации в ППЗУ результат выполнения команды *Blank Check*, отображаемый в окне регистрации сообщений *Transcript Window*, имеет следующий вид:

```
// *** BATCH CMD : BlankCheck -p 1
Validating device...
'1': IDCODE is '202015' (in hex).
'1': ID Check passed.
Device validated successfully.
'1': Performing blank check on device...
Failed at address, 0
'1': Part is not blank.
PROGRESS_END - End Operation.
Elapsed time = 0 sec.
PROGRESS_START - Starting Operation.
```

Для очистки содержимого микросхемы конфигурационной флэш-памяти с последовательным интерфейсом SPI (перевод в незапрограммированное состояние) предназначена команда *Erase*. Процесс выполнения операции «стирания» конфигурационной информации в ППЗУ сопровождается выводом всплывающей панели индикации в рабочем окне программы *iMPACT* и соответствующей информации в окне регистрации консольных сообщений *Transcript Window*, которая выглядит следующим образом:

```
// *** BATCH CMD : Erase -p 1
Validating device...
'1': IDCODE is '202015' (in hex).
'1': ID Check passed.
Device validated successfully.
'1': Erasing Device.
PROGRESS_START - Starting Operation.
PROGRESS_END - End Operation.
Elapsed time = 33 sec.
```

ПРОГРАММИРОВАНИЕ МИКРОСХЕМ КОНФИГУРАЦИОННОЙ ФЛЭШ-ПАМЯТИ С ПОСЛЕДОВАТЕЛЬНЫМ ИНТЕРФЕЙСОМ SPI ЧЕРЕЗ ПОРТ JTAG-ИНТЕРФЕЙСА ПЛИС FPGA

При косвенном методе программирования микросхем конфигурацион-

ной флэш-памяти с последовательным интерфейсом SPI следует присоединить загрузочный кабель к разъёму порта JTAG-интерфейса ПЛИС FPGA, к которой подключено это ППЗУ. Далее необходимо активизировать модуль программирования *iMPACT* в режиме периферийного сканирования *Boundary-Scan (JTAG) Mode*. Для этого в стартовой диалоговой панели данной программы с заголовком *Welcome to iMPACT* необходимо зафиксировать в нажатом состоянии кнопку *Configure devices using Boundary-Scan (JTAG)*. При этом рекомендуется установить режим автоматической идентификации подключенного загрузочного кабеля и цепочки периферийного сканирования, указав в выпадающем списке соответствующего поля выбора вариант *Automatically connect to cable and identify Boundary-Scan chain*. После закрытия диалоговой панели *Welcome to iMPACT* осуществляется процесс обнаружения загрузочного кабеля и цепочки периферийного сканирования. Информация о результатах процесса определения состава цепочки периферийного сканирования, отображаемая в окне регистрации консоль-

ных сообщений *Transcript Window*, имеет следующий вид:

```
Attempting to identify devices in
the boundary-scan chain configura-
tion...
// *** BATCH CMD : Identify
PROGRESS_START - Starting Operation.
Identifying chain contents
... '1': : Manufacturer's ID
=Xilinx xcf04s, Version : 15
INFO:iMPACT:1777 -
Reading
D:/Xilinx/10.1/ISE/xcf/data/xcf04
s.bsd...
INFO:iMPACT:501 - '1': Added De-
vice xcf04s successfully.
-----
'2': : Manufacturer's ID =Xilinx
xc3s700a, Version : 0
INFO:iMPACT:1777 -
Reading D:/Xilinx/10.1/ISE/spar-
tan3a/data/xc3s700a.bsd...
INFO:iMPACT:501 - '1': Added De-
vice xc3s700a successfully.
-----
done.
PROGRESS_END - End Operation.
Elapsed time = 2 sec.
```

При успешном определении ПЛИС в цепочке периферийного сканирования её УГО отображается в рабочем окне программы *iMPACT* и на экран выводится диалоговая панель с заголовком *Assign New Configuration File*. В этой диалоговой панели, предназначенной для выбора файла загружаемой конфигурационной последовательности, необходимо указать идентификатор соответствующего файла с расширением *bit* и переключить в нажатое состояние кнопку *Enable Programming of SPI Flash Device Attached to this FPGA*. Установка данной кнопки во включенное состояние предоставляет возможность программирования микросхемы конфигурационной флэш-памяти с последовательным интерфейсом SPI, которая подключена к ПЛИС, через порт JTAG-интерфейса кристалла FPGA. После нажатия клавиши «Открыть» (*Open*) в диалоговой панели с заголовком *Assign New Configuration File* она закрывается, и на экране появляется стандартная диалоговая панель открытия файла, озаглавленная *Add PROM File*. В этой диалоговой панели необходимо определить название используемого файла «про-

шивки», содержимое которого должно быть записано в ППЗУ.

После выбора требуемого файла, сгенерированного для микросхемы конфигурационной флэш-памяти с последовательным интерфейсом SPI, открывается следующая диалоговая панель, которая имеет заголовок *FPGA SPI Flash Association*. В этой диалоговой панели находится поле выбора *SPI Flash*, которое предназначено для определения конкретного типа используемой микросхемы флэш-памяти. В выпадающем списке этого поля выбора представлены условные обозначения всех ППЗУ с последовательным интерфейсом SPI, поддерживаемых модулем программирования *iMPACT*. Указав требуемый тип микросхемы флэш-памяти в поле выбора *SPI Flash*, необходимо подтвердить его нажатием клавиши *OK* в нижней части диалоговой панели с заголовком *FPGA SPI Flash Association*.

Далее выполнение операций косвенного программирования и обратного считывания информации рассматривается на примере микросхемы флэш-памяти с последовательным интерфейсом AT45DB161D, выпускаемой фирмой Atmel.

После определения названия файла «прошивки» в рабочем окне программы *iMPACT* добавляется условный графический образ флэш-ППЗУ с последовательным интерфейсом SPI, соединённый с УГО ПЛИС, а в окне регистрации консольных сообщений *Transcript Window* появляется следующая информация:

```
// *** BATCH CMD : identifyMPM
// *** BATCH CMD : assignFile -p
1 -file
"D:/Prj_n/Prj_indirect_spi/indirect_flash_spi.bit"
'1': Loading file '
D:/Prj_n/Prj_indirect_spi/indirect_flash_spi.bit ' ...
done.
INFO:iMPACT:2257 - Startup Clock
has been changed to 'JtagClk' in
the bitstream stored in memory,
but the original bitstream file
remains unchanged.
-----
INFO:iMPACT:501 - '1': Added De-
vice xc3s700a successfully.
-----
Selected part: AT45DB161D
// *** BATCH CMD : attachflash
-position 1 -spi "AT45DB161D"
// *** BATCH CMD : assignfile-
toattachedflash -position 1 -file
"D:/Prj_n/Prj_indirect_spi/indi-
```

```
rect_flash_spi.mcs"
// *** BATCH CMD : attachflash
-position 1 -spi "AT45DB161D"
```

Кроме того, на экране автоматически отображается диалоговая панель с заголовком *Device Programming Properties*. Эта панель содержит несколько страниц, содержащих параметры ПЛИС и ППЗУ, входящих в состав цепочки периферийного сканирования. Выбор требуемой страницы осуществляется с помощью интерактивного списка *Category*, который отображается в левой части этой диалоговой панели. Для просмотра и установки параметров программирования микросхемы конфигурационной флэш-памяти с последовательным интерфейсом SPI, подключенной к ПЛИС, необходимо в данном списке выделить строку *Device N (Attached Flash*, тип микросхемы ППЗУ), где *N* – порядковый номер элемента в цепочке периферийного сканирования. При этом открывается страница, на которой представлено два параметра, относящихся к ППЗУ, – *Verify* и *Erase Before Programming*. Назначение этих параметров было рассмотрено выше в разделе, посвящённом прямому программированию микросхем конфигурационной флэш-памяти с последовательным интерфейсом SPI.

После установки требуемых значений параметров программирования флэш-ППЗУ необходимо выделить его УГО в рабочем окне программы *iMPACT*. При этом открывается доступ к командам, представленным на странице *Configuration Operations* в окне процессов *Processes Window* и во всплывающем меню *Operations* управляющей оболочки Навигатора проекта. В рассматриваемом режиме программный модуль *iMPACT* поддерживает те же команды, что и при непосредственном способе программирования микросхем конфигурационной флэш-памяти с последовательным интерфейсом SPI. Но процесс выполнения этих команд отличается от аналогичных операций, осуществляемых при непосредственном подключении загрузочного JTAG-кабеля к порту интерфейса SPI ППЗУ. Протокол процесса программирования микросхемы конфигурационной флэш-памяти с последовательным интерфейсом SPI через порт JTAG-интерфейса ПЛИС FPGA, активизируемого командой *Program*, имеет следующий вид:

```
// *** BATCH CMD : Program -p 1
-spionly
Boundary-scan chain validated
successfully.
INFO:iMPACT - Downloading core
file
D:/Xilinx/10.1/ISE/spartan3a/data
/xc3s700a_spi.cor.
PROGRESS_START - Starting Opera-
tion.
'1': Reading status register con-
tents...
CRC error : 0
INFO:iMPACT:2219 - Status regis-
ter values:
INFO:iMPACT - 0011 1111 1100 1100
INFO:iMPACT:2492 - '1': Completed
downloading core to device.
'1': Core downloaded success-
fully.
INFO:iMPACT - Address 0x00000000
is in sector 0.
INFO:iMPACT - Address 0x0005364B
is not located at the start of a
sector boundary. The whole sector
will be erased.
INFO:iMPACT - Address 0x0005364B
is in sector 3.
INFO:iMPACT - Address 0x00000000
is in sector 0.
INFO:iMPACT - Address 0x0005364B
is not located at the start of a
sector boundary. The whole sector
will be erased.
INFO:iMPACT - Address 0x0005364B
is in sector 3.
INFO:iMPACT - Address 0x00000000
is in page 0.
INFO:iMPACT - Address 0x0005364B
is in page 646.
'1': Programming completed suc-
cessfully.
PROGRESS_END - End Operation.
```

ЧТЕНИЕ КОНФИГУРАЦИОННЫХ ДАННЫХ И СТАТУСНОЙ ИНФОРМАЦИИ ИЗ МИКРОСХЕМ КОНФИГУРАЦИОННОЙ ФЛЭШ-ПАМЯТИ С ПОСЛЕДОВАТЕЛЬНЫМ ИНТЕРФЕЙСОМ SPI С ПОМОЩЬЮ МОДУЛЯ ПРОГРАММИРОВАНИЯ iMPACT ЧЕРЕЗ ПОРТ JTAG-ИНТЕРФЕЙСА ПЛИС FPGA

Модуль программирования *iMPACT* позволяет осуществлять операции обратного считывания конфигурационной и статусной информации из ППЗУ с последовательным интерфейсом SPI через порт JTAG-интерфейса ПЛИС FPGA с помощью тех же команд, что и при не-

посредственным подключении загрузочного кабеля к порту SPI микросхемы флэш-памяти. Но процесс выполнения этих операций существенно отличается от рассмотренного выше прямого чтения конфигурационных и статусных данных. Поэтому в настоящем разделе приводятся только протоколы выполнения команд обратного считывания конфигурационной и статусной информации, а также очистки содержимого ППЗУ с интерфейсом SPI через порт JTAG-интерфейса ПЛИС FPGA.

Информация с результатами контроля содержимого конфигурационной флэш-памяти с последовательным интерфейсом, осуществляемого с помощью команды *Verify* через порт JTAG-интерфейса ПЛИС FPGA, отображается в окне регистрации сообщений *Transcript Window* в следующем виде:

```
// *** BATCH CMD : Verify -p 1
-spionly
Boundary-scan chain validated
successfully.
INFO:iMPACT - Downloading core
file
```

```
D:/Xilinx/10.1/ISE/spartan3a/data
/xc3s700a_spi.cor.
PROGRESS_START - Starting Opera-
tion.
'1': Reading status register con-
tents...
CRC error : 0
value of VSEL pin 0 : 1
value of VSEL pin 1 : 1
value of VSEL pin 2 : 1
value of MODE pin M0 : 1
value of MODE pin M1 : 0
value of MODE pin M2 : 0
POST_CRC_ERR error : 0
INFO:iMPACT:2219 - Status regis-
ter values:
INFO:iMPACT - 0011 1111 1100 1100
INFO:iMPACT:2492 - '1': Completed
downloading core to device.
'1': Core downloaded success-
fully.
INFO:iMPACT - Address 0x00000000
is in page 0.
INFO:iMPACT - Address 0x0005364B
is in page 646.
'1': Verifying device...done.
'1': Verification completed suc-
cessfully.
PROGRESS_END - End Operation.
```

Результаты операции вычисления контрольной суммы конфигурационных данных, содержащихся во флэш-ППЗУ с последовательным интерфейсом SPI, и сравнения со значением контрольной суммы файла программирования, выполняемой при выборе команды *Get Device Checksum*, выводятся в окне регистрации сообщений *Transcript Window* в следующей форме:

```
// *** BATCH CMD : Checksum -p 1
-spionly
Boundary-scan chain validated
successfully.
INFO:iMPACT - Downloading core
file
D:/Xilinx/10.1/ISE/spartan3a/data
/xc3s700a_spi.cor.
PROGRESS_START - Starting Opera-
tion.
'1': Reading status register con-
tents...
CRC error : 0
value of VSEL pin 0 : 1
value of VSEL pin 1 : 1
value of VSEL pin 2 : 1
value of MODE pin M0 : 1
value of MODE pin M1 : 0
```

```
value of MODE pin M2 : 0
POST_CRC_ERR error : 0
INFO:iMPACT:2219 - Status register values:
INFO:iMPACT - 0011 1111 1100 1100
'1': Core downloaded successfully.
INFO:iMPACT - Address 0x00000000 is in page 0.
INFO:iMPACT - Address 0x0005364B is in page 646.
'1': Reading device contents...done.
All Checksum calculations are performed on the address range spanned by the configuration file.
'1': Calculated checksum matches expected checksum, 000009c96
PROGRESS_END - End Operation.
```

Считывание конфигурационной информации, содержащейся в выбранной микросхеме флэш-памяти с последовательным интерфейсом SPI, осуществляемое с помощью команды *Readback*, сопровождается следующими сообщениями:

```
// *** BATCH CMD : ReadbackToFile
-p 1 -file " D:/Prj_n/Prj_indirect_spi/old_flash_spi.mcs.mcs"
-spionly
Boundary-scan chain validated successfully.
INFO:iMPACT - Downloading core file D:/Xilinx/10.1/ISE/spartan3a/data/xc3s700a_spi.cor.
PROGRESS_START - Starting Operation.
'1': Reading status register contents...
CRC error : 0
value of VSEL pin 0 : 1
value of VSEL pin 1 : 1
value of VSEL pin 2 : 1
value of MODE pin M0 : 1
value of MODE pin M1 : 0
value of MODE pin M2 : 0
POST_CRC_ERR error : 0
INFO:iMPACT:2219 - Status register values:
INFO:iMPACT - 0011 1111 1100 1100
INFO:iMPACT:2492 - '1': Completed downloading core to device.
INFO:iMPACT - '1': Checking done pin...done.
'1': Core downloaded successfully.
'1': Performing readback on device...done.
PROGRESS_END - End Operation.
```

Сведения о состоянии запрограммированной микросхемы конфигурационной флэш-памяти с последовательным интерфейсом SPI, полученные с помощью команды *Blank Check*, отображаются в окне регистрации консольных сообщений *Transcript Window* в следующем виде:

```
// *** BATCH CMD : BlankCheck -p
1 -spionly
Maximum TCK operating frequency for this device chain: 10000000.
Validating chain...
Boundary-scan chain validated successfully.
PROGRESS_START - Starting Operation.
INFO:iMPACT - Address 0x00000000 is in page 0.
INFO:iMPACT - Address 0x0005364B is in page 646.
'1': Performing blank check on device...Blank check failed on page 0.
'1': Part is not blank.
done.
PROGRESS_END - End Operation.
```

Если выбранный элемент ППЗУ не содержит конфигурационных данных (находится в незапрограммированном состоянии), то при выполнении команды *Blank Check* в окне регистрации сообщений *Transcript Window* появится следующая информация:

```
// *** BATCH CMD : BlankCheck -p
1 -spionly
Maximum TCK operating frequency for this device chain: 10000000.
Validating chain...
Boundary-scan chain validated successfully.
PROGRESS_START - Starting Operation.
INFO:iMPACT - Address 0x00000000 is in page 0.
INFO:iMPACT - Address 0x0005364B is in page 646.
'1': Performing blank check on device...done.
'1': Part is blank.
PROGRESS_END - End Operation.
```

Процесс выполнения команды *Erase*, осуществляющей «стирание» содержимого выбранной микросхемы флэш-памяти с последовательным интерфейсом SPI, подключенной к ПЛИС, сопровождается следующими сообщениями:

```
/ *** BATCH CMD : Erase -p 1
-spionly
Boundary-scan chain validated successfully.
INFO:iMPACT - Address 0x00000000 is in sector 0.
INFO:iMPACT - Start block = 0 for address 0x00000000.
INFO:iMPACT - Address 0x0005364B is not located at the start of a sector boundary. The whole sector will be erased.
INFO:iMPACT - Address 0x0005364B is in sector 3.
INFO:iMPACT - End block = 95 for address 0x0005364B.
INFO:iMPACT - Address 0x00000000 is in sector 0.
INFO:iMPACT - Address 0x0005364B is not located at the start of a sector boundary. The whole sector will be erased.
INFO:iMPACT - Address 0x0005364B is in sector 3.
INFO:iMPACT - Address 0x00000000 is in sector 0.
INFO:iMPACT - Address 0x0005364B is not located at the start of a sector boundary. The whole sector will be erased.
INFO:iMPACT - Address 0x0005364B is in sector 3.
PROGRESS_START - Starting Operation.
'1': Erasing sector protect register...'1': Erasing device...done.
'1': Erasure completed successfully.
PROGRESS_END - End Operation.
```

ПРОГРАММИРОВАНИЕ МИКРОСХЕМ КОНФИГУРАЦИОННОЙ ФЛЭШ-ПАМЯТИ С ПАРАЛЛЕЛЬНЫМ ИНТЕРФЕЙСОМ ВРІ ЧЕРЕЗ ПОРТ JTAG-ИНТЕРФЕЙСА ПЛИС FPGA

Процесс программирования микросхем конфигурационной флэш-памяти с параллельным интерфейсом ВРІ через порт JTAG-интерфейса ПЛИС FPGA осуществляется, в основном, таким же образом, как и для ППЗУ с последовательным интерфейсом SPI. Вначале необходимо активизировать модуль программирования *iMPACT* в режиме периферийного сканирования *Boundary-Scan(JTAG) Mode*. Затем, после обнаружения загрузочного кабеля и цепочки периферийного сканирования, в диалоговой пане-

ли *Assign New Configuration File* необходимо указать идентификатор файла загружаемой конфигурационной последовательности и зафиксировать в нажатом положении кнопку *Enable Programming of BPI Flash Device Attached to this FPGA*. При этом становится доступным косвенный метод программирования микросхемы конфигурационной флэш-памяти с параллельным интерфейсом BPI, которая подключена к ПЛИС, через порт JTAG-интерфейса этого кристалла FPGA.

Далее в стандартной диалоговой панели открытия файла с заголовком *Add PROM File* следует указать название используемого файла программирования, содержимое которого должно быть записано в ППЗУ. Этот файл должен быть сформирован специально для микросхемы конфигура-

ционной флэш-памяти с параллельным интерфейсом BPI. Процесс подготовки файлов программирования для ППЗУ этого типа был подробно рассмотрен в двадцать второй части настоящего курса. После выбора требуемого файла «прошивки» открывается очередная диалоговая панель, озаглавленная *FPGA BPI Flash Association*. С помощью поля выбора *BPI Flash*, которое расположено в этой диалоговой панели, необходимо определить тип применяемой микросхемы параллельной флэш-памяти. Выпадающий список данного поля выбора содержит условные обозначения всех ППЗУ с параллельным интерфейсом BPI, поддерживаемых модулем программирования *iMPACT*.

После выбора условного обозначения используемой микросхемы флэш-памя-

ти параллельного типа необходимо выполнить ту же последовательность действий, что и при программировании ППЗУ с интерфейсом SPI через порт JTAG-интерфейса ПЛИС FPGA. При этом поддерживаются все команды записи и верификации данных, обратного считывания конфигурационной и статусной информации, а также очистки содержимого элемента флэш-памяти, которые были рассмотрены в предыдущих разделах. Подробная информация о результатах выполнения этих операций отображается в окне регистрации консольных сообщений *Transcript Window*.

На этом завершается изучение процесса программирования элементов основных типов энергонезависимой памяти, применяемых для хранения конфигурационной информации ПЛИС с архитектурой FPGA. ☺

Новости мира News of the World Новости мира

Бинарная система сдерживает мощность квантовых компьютеров

Следует отказаться от традиционной бинарной системы при разработке сверхбыстрых квантовых компьютеров. Переход на новую пятирежимную систему позволит упростить построение сверхмощных машин. Об этом заявляют Мэтью Нили (Matthew Neeley) и его коллеги из Университета Калифорнии (University of California) в Санта-Барбаре.

До сих пор разработка квантовых компьютеров следовала традиционной бинарной вычислительной модели. Она кодирует всю информацию при помощи компонентов, которые могут находиться в двух состояниях: 0 или 1. Но существуют и другие возможности. Нили объясняет, что можно использовать тернарную систему с тремя цифрами, и тогда основные единицы должны стать троичными, или тритам (ternary digit), т.е. будут, по сути, трёхпозиционные переключатели. Один трит будет содержать больше информации, чем обычный бит.

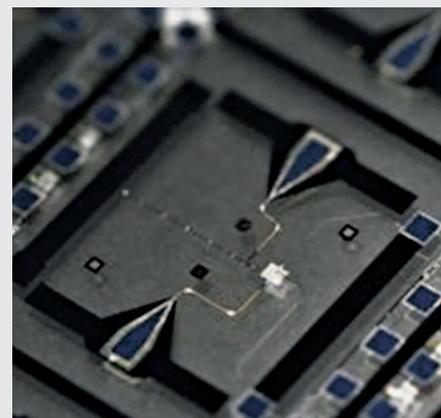
Команда Нили уже построила квантовый компьютер, чьи стандартные блоки имеют пять основных состояний. До сих пор стандартные блоки квантовых компьютеров были бинарными квантовыми битами (кубитами), которые кодировали два положения в квантовый спин атомов, электронов или протонов. Способность таких частиц игнорировать обычную логику и существование сразу в нескольких квантовых состояниях должны однажды

позволить квантовым компьютерам выполнять огромное количество вычислений одновременно.

Исследователи использовали сверхпроводящий алюминий и кремниевую схему на сапфировой подложке для получения пятирежимных кубитов или кудитов (qudit). В кудитах содержится ещё больше информации, чем в кубитах, что позволит производить вычисления с меньшим количеством кудитов, – поясняют учёные. Возбуждая микроволновые фотоны пяти различных частот в схеме, они смогли способствовать их переходам между пятью дискретными энергетическими уровнями. Также был разработан метод квантового измерения, который позволяет различать между собой эти уровни.

Однако один кудит мало что даёт. Джонатан Хоум (Jonathan Home) из Национального института стандартов и технологий (National Institute of Standards and Technology, NIST) в Боулдере заявил, что команде Нили необходимо расширить эту базовую систему, чтобы два или более кудитов могли передавать информацию между собой, позволяя осуществлять более сложные вычислительные операции. «Разработка такого рода систем, где два кудита взаимодействуют, но по-прежнему сохраняют интересные свойства пятирежимной системы, и будет главной задачей», – говорит Хоум.

Потенциальные возможности квантовых компьютеров уже вызывают интерес Управления перспективных исследова-



ний (Advanced Research Projects Agency) США, которое надеется использовать их для взлома кодов. Это управление уже оказывает поддержку исследованиям Хоума в области квантовых компьютеров, работающих при комнатной температуре, чтобы бинарные кубиты могли взаимодействовать и обмениваться информацией. Последние результаты показывают, что ионы магния могут использоваться для остановки взаимной дестабилизации кубитов из-за передачи тепла, равно как и их квантовых состояний.

Этот трюк использует плотно сжатую цепочку ионов бериллия в качестве кубитов, в то время как прилегающие ионы магния поглощают все тепло. Тепло, как правило, приводит к разрушению квантовой информации. «И это откроет путь крупномасштабным квантовым вычислениям, поскольку это служит важной задаче – передаче информации», – говорит Хоум.

newscientist.com