

# Реализация аппаратной платформы микропроцессорной системы, проектируемой на основе ядра MicroBlaze, в ПЛИС семейств FPGA фирмы Xilinx

**В предыдущих номерах журнала [5–6] были подробно рассмотрены процедуры создания нового проекта микропроцессорной системы на основе 32-разрядного ядра MicroBlaze [1–4] в САПР Xilinx Embedded Development Kit (EDK) и формирования спецификации аппаратной платформы. Настоящая статья продолжает ознакомление с содержанием этапов проектирования встраиваемых микропроцессорных систем, предназначенных для реализации в ПЛИС семейств FPGA фирмы Xilinx [8]. В данной публикации обсуждаются этапы генерации списка соединений аппаратной части разрабатываемой микропроцессорной системы, синтеза и реализации проекта в кристалле FPGA.**

**Валерий Зотов**

walerry@km.ru

**П**режде чем перейти к рассмотрению очередного этапа проектирования, необходимо выполнить еще одну процедуру, связанную с формированием описания аппаратной части разрабатываемой системы. Для получения законченного описания аппаратных средств проектируемой микропроцессорной системы необходимо кроме спецификации MHS (Microprocessor Hardware Specification) подготовить дополнительную информацию о топологических и временных ограничениях, которая используется средствами синтеза, размещения и трассировки проекта в кристалле.

## **Подготовка файла временных и топологических ограничений проекта аппаратной платформы**

Дополнительная информация, используемая программами синтеза, размещения и трассировки проекта аппаратной части микропроцессорной системы, задается в форме файла временных и топологических ограничений User Constraints File (UCF). Этот файл имеет текстовый формат, каждая строка которого представляет собой выражение, описывающее соответствующий параметр (ограничение проекта). Для создания файла временных и топологических ограничений и внесения в него информации можно использовать встроенный текстовый редактор управляющей оболочки Xilinx Platform Studio (XPS) или специальную программу Constraints Editor, которая входит в состав систем проектирования серии Xilinx ISE (Integrated Synthesis Environment). Программа Constraints Editor на основании данных, указанных разработчиком в диалоговом режиме, автоматически формирует соответствующие выражения

для описания ограничений проекта. Кроме того, для этих целей может использоваться редактор назначения выводов кристалла и топологических ограничений PACE (Pinout and Area Constraints Editor), который также входит в состав САПР серии ISE. Способы создания и редактирования файлов ограничений средствами пакетов проектирования Xilinx ISE подробно рассмотрены в отдельной книге [7], поэтому далее кратко описывается только процесс подготовки файла временных и топологических ограничений при использовании интегрированного текстового редактора управляющей оболочки Xilinx Platform Studio.

Для создания файла UCF во встроенном текстовом редакторе XPS следует выбрать пункт основного меню **File**, после чего в появившемся всплывающем меню активизировать команду **New**, в результате чего откроется новое окно редактирования. Ввод выражений, описывающих временные и топологические ограничения проекта, осуществляется с помощью клавиатуры. После завершения процесса редактирования необходимо сохранить изменения файла ограничений, используя команду **Save As** из меню **File**. При этом на экране монитора отображается стандартная диалоговая панель сохранения файла, в которой необходимо выбрать соответствующий раздел на диске и указать полное название формируемого файла ограничений. Вновь созданный файл UCF должен располагаться в разделе *data* рабочего каталога проекта разрабатываемой микропроцессорной системы. В качестве имени файла ограничений нужно указать название файла проекта создаваемой системы, а в качестве расширения — *ucf*.

В файлах ограничений проектов встраиваемых микропроцессорных систем наиболее часто применя-

ются два типа выражений. Ограничения первого типа позволяют установить соответствие между внешними цепями проектируемой системы и номерами выводов ПЛИС. Выражения второго типа используются для определения максимального значения периода сигнала синхронизации.

Параметр LOC позволяет осуществить закрепление выводов перед трассировкой, а также явно указать конфигурируемый логический блок (Configurable Logic Block, CLB) для реализации элементов проекта. Для привязки внешних цепей проектируемой системы (подключаемых к контактам кристалла) к требуемым выводам ПЛИС используется следующий формат выражения:

```
NET <название_цепи> LOC=<номер_вывода_ПЛИС>;
```

Форма записи номера вывода ПЛИС в этом выражении зависит от типа используемого корпуса микросхемы. Для корпусов типа BGA номер вывода указывается в том же виде, в каком он представлен в документации. При использовании корпусов других типов номер вывода ПЛИС, указанный в документации, в выражении ограничения LOC сопровождается префиксом P. Приведем примеры топологических ограничений, используемых для указания соответствия внешних цепей разрабатываемой микропроцессорной системы и номеров выводов ПЛИС:

```
NET clock_s LOC=C5;
NET dat1 LOC=P10;
```

Максимальное значение периода сигнала синхронизации для соответствующей цепи разрабатываемой микропроцессорной системы задается с помощью параметра *PERIOD*. Полный формат соответствующего выражения ограничения имеет вид:

```
NET <название_цепи_синхронизации> PERIOD=<длительность_периода> [<единицы_измерения>] [[HIGH | LOW][<длительность_первой_фазы_периода> [<единицы_измерения>] ]];
```

где значение HIGH или LOW указывает логический уровень сигнала в первой фазе периода, а элементы, указанные в квадратных скобках, являются необязательными. По умолчанию в качестве единиц измерения длительности установлены наносекунды — нс (ns) и одинаковая продолжительность состояний высокого и низкого уровня периода синхросигнала, благодаря чему получается сокращенный формат записи:

```
NET <название_цепи_синхронизации> PERIOD = <длительность_периода>;
```

Например:

```
NET clock PERIOD=10ns;
```

Таким образом, значение параметра *PERIOD* накладывает ограничение на время распространения сигналов по цепям и логике, подключенных между выходом одного и входом другого синхронного элемента (триггера, регистра или ОЗУ), которые тактируются одним и тем же сигналом синхронизации.

В качестве примера, иллюстрирующего процедуру формирования временных и тополо-

гических ограничений разрабатываемой системы, рассмотрим процесс создания файла UCF для проекта простейшей микропроцессорной системы сбора и обработки данных, спецификация аппаратной платформы которой подготовлена в предыдущей публикации [6]. Для реализации этой системы воспользуемся отладочной платой из инструментального комплекта Spartan-3 Starter Kit. Основным элементом данного отладочного модуля является ПЛИС семейства Spartan-3 XC3S200 в корпусе FT256. В качестве источника тактового сигнала в этом модуле используется кварцевый генератор с частотой 50 МГц. Поэтому для глобальной цепи синхронизации systemClk должно быть указано значение периода, равное 20 нс. Так как выход кварцевого генератора на отладочной плате подключен к выводу T9 ПЛИС XC3S200, этот контакт кристалла должен выполнять функцию входа тактового сигнала systemClk. Для реализации функции входа сигнала принимаемых данных uartRX ядра универсального асинхронного последовательного приемопередатчика назначается вывод ПЛИС T13, а для выполнения функции выхода сигнала передаваемых данных uartTX — вывод R13, так как именно с этими контактами кристалла соединены соответствующие входы микросхемы, выполняющей преобразование уровней сигналов последовательного интерфейса RS-232. Входные цепи ядра параллельного интерфейса ввода-вывода in\_data<0> — in\_data<7>, на которые поступают сигналы входных данных, подключим к восьми ползунковым переключателям, расположенным на отладочной плате. С помощью этих переключателей будут формироваться входные тестовые данные. Соответствующие выходы переключателей подключены к выводам K13, K14, J13, J14, H13, H14, G12 и F12 ПЛИС XC3S200. Для формирования сигнала сброса микропроцессорной системы systemRst можно использовать одну из кнопок, расположенных на отладочной плате, например, кнопку BTN3, подключенную к выводу L14 ПЛИС XC3S200. С учетом всех приведенных выше замечаний окончательное содержимое файла временных и топологических ограничений для микропроцессорной системы сбора и обработки данных system\_tst.ucf должно выглядеть следующим образом:

```
Net systemClk PERIOD = 20ns;
Net uartRX LOC=T13;
Net uartTX LOC=R13;
Net in_data<0> LOC=K13;
Net in_data<1> LOC=K14;
Net in_data<2> LOC=J13;
Net in_data<3> LOC=J14;
Net in_data<4> LOC=H13;
Net in_data<5> LOC=H14;
Net in_data<6> LOC=G12;
Net in_data<7> LOC=F12;
Net systemClk LOC=T9;
Net systemRst LOC=L14;
```

После завершения подготовки спецификации MHS и соответствующего файла временных и топологических ограничений можно перейти к следующему этапу проектирования, в результате выполнения которого должен быть сформирован список соединений аппаратной платформы.

## Формирование списка соединений

Данный этап включает в себя две фазы. В первой фазе осуществляется генерация модулей HDL-описаний в соответствии с подготовленной спецификацией аппаратной платформы проектируемой системы. Вторая фаза представляет собой процедуру синтеза аппаратной части встраиваемой микропроцессорной системы, в процессе которой производится трансформация исходных модулей HDL-описания в список цепей (netlist), выполненный на низком логическом уровне. Элементы низкоуровневого описания, формируемого в процессе синтеза, должны соответствовать архитектуре семейства ПЛИС, выбранного для реализации проекта. Синтезированный список цепей должен быть максимально адаптирован к ресурсам используемого кристалла, что обеспечивает его наиболее эффективное отображение средствами размещения и трассировки на физическом уровне. В процессе синтеза выполняется оптимизация полученного списка цепей в соответствии с выбранным критерием. Основным результатом фазы синтеза является формирование файла NGC, который представляет собой описание проекта аппаратной части системы на низком логическом уровне в двоичном формате. Процедура синтеза выполняется теми средствами, которые были выбраны при определении параметров проекта с помощью опции *Synthesis Tool* [5]. В качестве инструмента синтеза, применяемого по умолчанию, используются средства Xilinx Synthesis Tool (XST), которые являются составной частью системы автоматизированного проектирования Foundation ISE.

Этап генерации списка соединений аппаратной платформы выполняется в автоматическом режиме. Для активизации этого процесса следует выбрать команду *Generate Netlist* из всплывающего меню **Tools** управляющей оболочки Xilinx Platform Studio или воспользоваться кнопкой быстрого доступа , расположенной на оперативной панели управления XPS. Информация о ходе выполнения каждой фазы рассматриваемого этапа проектирования отображается во встроенном окне консольных сообщений на странице с закладкой *Output*. В случае обнаружения ошибок соответствующие сообщения выводятся на странице *Errors* окна консольных сообщений, а предупреждения — на странице *Warnings*.

При успешном завершении всех процедур данного этапа создается файл NGC, который содержит результаты синтеза проекта.

Следующим этапом разработки является функциональное моделирование аппаратной части. Этот процесс выполняется в рамках пакета моделирования ModelSim и практически не отличается от моделирования различных цифровых устройств, проектируемых на основе ПЛИС фирмы Xilinx. Выполнение этапов функционального и временного моделирования рассмотрено в соответствующей литературе [7]. Кроме того, этапы моделирования не являются строго обязательными, поэтому переходим непосредственно к этапу реализации проекта аппаратной платформы в кристалле FPGA.

### Реализация проекта в кристалле ПЛИС FPGA

Этап реализации (*Implementation*) проекта аппаратной части встраиваемой микропроцессорной системы, реализуемой на базе ПЛИС семейств FPGA, включает в себя три фазы: трансляция (*Translate*), отображение логического описания проекта на физические ресурсы кристалла (*MAP*), размещение и трассировка (*Place and Route*). В процессе трансляции выполняется объединение всех списков соединений в формате EDIF, входящих в состав проекта, и информации обо всех ограничениях, которая содержится в файле UCF (User Constraints File). Результатом фазы трансляции является формирование логического описания проекта в терминах примитивов низкого уровня Xilinx с учетом временных и топологических ограничений, выполненного в формате NGD (Native Generic Database). Во второй фазе рассматриваемого этапа логическое описание проекта, полученное на предыдущем шаге, проецируется на физические ресурсы кристалла FPGA выбранного типа. При этом выполняется оптимизация в соответствии с выбранным критерием и заданными ограничениями. В процессе размещения и трассировки выбирается наилучшее расположение конфигурируемых логических блоков, реализующих соответствующие функции проектируемой системы, и выполняются необходимые соединения с учетом временных и топологических ограничений.

В результате выполнения этапа создается двоичный файл, который описывает использование физических ресурсов кристалла для реализации компонентов (функций) разрабатываемой системы и выполнения необходимых соединений между ними. Этот файл затем используется в качестве исходного для генерации конфигурационной последовательности ПЛИС.

Процесс реализации проекта аппаратной части системы выполняется средствами пакета САПР Foundation ISE. Управление этим процессом может осуществляться как в среде XPS, так и непосредственно в рамках оболочки пакета Foundation ISE — *Навигатора проекта* (*Project Navigator*). Требуемый способ управления процессом реализации проекта выбирается при установке параметров проекта EDK с помощью переключателя *Implementation Tool Flow* [5].

Если для управления процессом реализации проекта был выбран первый способ (непосредственно из оболочки Xilinx Platform Studio), то для выполнения этого этапа следует выбрать команду *Generate Bitstream* из всплывающего меню *Tools* или нажать кнопку  на оперативной панели управления XPS. Далее все необходимые процедуры выполняются в автоматическом режиме, включая формирование конфигурационной последовательности, загружаемой в кристалл. При этом используются значения параметров программ трансляции, компоновки, размещения и трассировки, а также генерации конфигурационной последовательности, установленные по умолчанию. Информационные сообще-

ния этих программ выводятся на странице *Output* встроенного окна консольных сообщений XPS, а предупреждения и сообщения об ошибках — на страницах *Warnings and Errors* соответственно.

При выборе второго способа, когда управление процессом реализации проекта осуществляется с помощью *Навигатора проекта* пакета Foundation ISE, разработчик может установить требуемые параметры соответствующих процедур, например, выбрать критерий оптимизации при размещении и трассировке проекта аппаратной части в кристалле. В этом случае прежде всего необходимо скопировать файл NGC, содержащий результаты синтеза аппаратной платформы, в рабочий каталог соответствующего проекта для пакета Foundation ISE. Этот проект автоматически создается средствами XPS в разделе, указанном в параметрах проекта Xilinx EDK (в поле редактирования *NPL File*) [5]. Процесс копирования файла NGC выполняется с помощью команды *Export to ProjNav*, которая находится во всплывающем меню *Tools*. Далее нужно активизировать графическую оболочку *Навигатора проекта* пакета Foundation ISE. Для этого следует, не закрывая окна XPS, дважды щелкнуть левой кнопкой мыши на пиктограмме , расположенной на Рабочем столе Windows, или использовать альтернативный метод запуска программ в среде Windows 2000/XP. В последнем случае нужно в меню кнопки *Пуск* выполнить команду *Программы/Xilinx ICE 6/Project Navigator*. При успешном выполнении указанных операций на экране монитора отображается основное окно *Навигатора проекта*, структура которого детально рассмотрена в отдельной публикации [7]. Далее необходимо открыть проект разрабатываемой системы, который был создан автоматически средствами Xilinx EDK. Для этого следует выполнить команду *File/Open Project*, в результате исполнения которой на экране отображается стандартная диалоговая панель открытия существующего проекта. В этой панели необходимо указать рабочий каталог и название проекта, которые были заданы в параметрах проекта Xilinx EDK. После закрытия диалоговой панели во встроенном окне исходных модулей (файлов) проекта (*Sources in Project*) отображаются строки с названием проекта, параметрами ПЛИС, используемой для реализации разрабатываемой системы, и названием файла NGC с результатами синтеза аппаратной платформы, который был сформирован на предыдущем этапе. При выделении строки с названием файла NGC во встроенном окне процессов (*Processes for Source*) открывается список процедур, необходимых для выполнения этапа реализации. Представленный в окне процессов список процедур соответствует основным фазам данного этапа, которые были перечислены выше. На рис. 1 в качестве примера приведен вид рабочей области *Навигатора проекта* при открытии проекта аппаратной части микропроцессорной системы сбора и обработки данных.

В список исходных модулей проекта в САПР Foundation ISE необходимо включить файл временных и топологических ограничений UCF, сформированный ранее. Для этого сле-

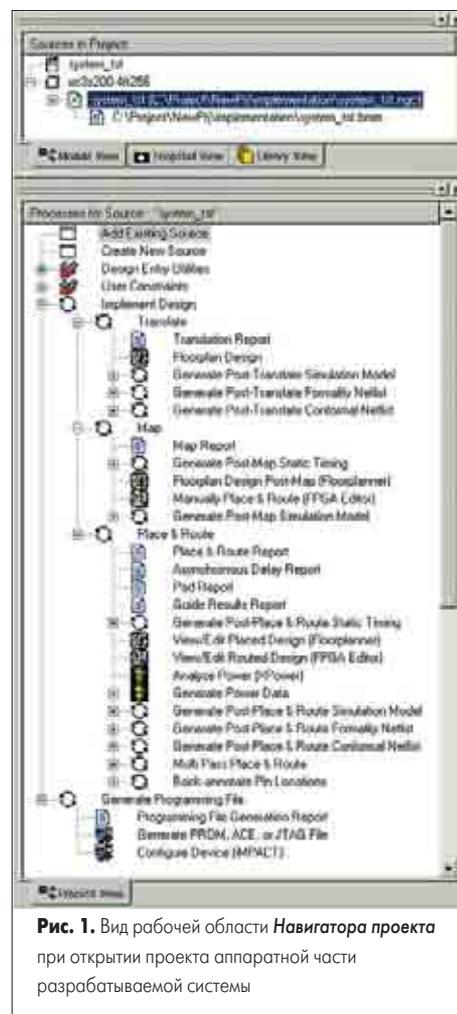


Рис. 1. Вид рабочей области *Навигатора проекта* при открытии проекта аппаратной части разрабатываемой системы

дует выбрать пункт *Project* основного меню *Навигатора проекта*, а затем в появившемся всплывающем меню выполнить команду *Add Source* или *Add Copy of Source*. Исполнение этих команд начинается с вывода стандартной диалоговой панели открытия файла, в которой нужно указать название и расположение требуемого файла ограничений UCF, находящегося в разделе *data* рабочего каталога проекта разрабатываемой микропроцессорной системы. При использовании команды *Add Source* этот файл добавляется в состав текущего проекта САПР Foundation ISE, но не переносится в его рабочий каталог. Команда *Add Copy of Source* выполняет аналогичную функцию и отличается только тем, что файл UCF, включаемый в состав проекта, копируется в рабочий каталог текущего проекта.

Для управления всеми процедурами этапа реализации, выполняемыми в автоматическом режиме, используются соответствующие параметры. Их значения могут быть заданы поочередно для каждой фазы в отдельности (например, перед ее активизацией) или сразу для всего процесса в целом. В первом случае следует выделить в окне процессов (рис. 1) строку с названием соответствующей фазы, после чего нажать кнопку , расположенную на оперативной панели *Навигатора проекта*, или воспользоваться командой *Properties* контекстно-зависимого всплывающего меню, которое выводится при щелчке правой кнопкой мыши. Затем в появившейся диалоговой панели нужно установить требуемые значения параметров. При втором способе в окне процессов (рис. 1) выделяется

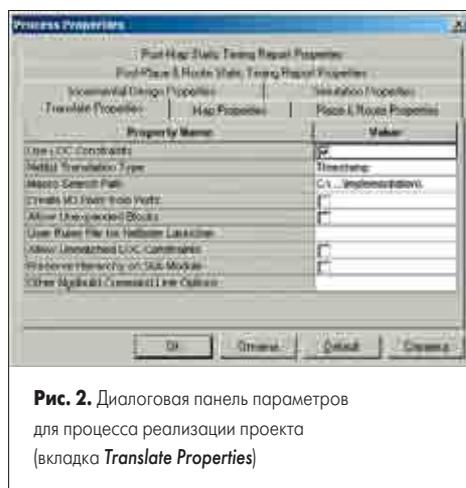


Рис. 2. Диалоговая панель параметров для процесса реализации проекта (вкладка *Translate Properties*)

строка с названием этапа *Implement Design* и далее выполняется та же последовательность действий, что и в первом случае. При использовании данного способа диалоговая панель параметров содержит семь вкладок: *Translate Properties*, *MAP Properties*, *Place and Route Properties*, *Incremental Design Properties*, *Simulation Properties*, *Post-Map Static Timing Report Properties*, *Post-Place & Route Timing Report Properties* (рис. 2). Назначение большинства параметров, представленных на этих страницах, детально рассмотрено в книге [7]. Поэтому далее подробно обсуждаются только те параметры процесса реализации, на которые рекомендуется обратить особое внимание при выполнении данного этапа разработки системы.

На странице *Translate Properties*, содержащей параметры процедуры трансляции проекта, следует убедиться, что для параметра *Use LOC Constraints* установлено значение «включено». В этом случае процесс трансляции будет выполняться с учетом топологических ограничений, описываемых с помощью выражений  $LOC = \dots$ , которые присутствуют в файле UCF.

Открыв страницу *MAP Properties*, на которой сосредоточены параметры процедуры распределения ресурсов CLB кристалла для реализации проектируемой системы, необходимо с помощью параметра *Optimization Strategy (Cover Mode)* выбрать стратегию оптимизации, осуществляемой во время выполнения данной фазы. На этой фазе производится назначение функциональных генераторов CLB для реализации соответствующих функций системы. В выпадающем списке возможных значений этого параметра представлено четыре элемента: *Area*, *Speed*, *Balanced*, *Off*. Значение *Area*, установленное по умолчанию, определяет в качестве критерия оптимизации минимизацию количества используемых таблиц преобразования (LUT) и, следовательно, конфигурируемых логических блоков. При выборе значения *Speed* целью оптимизации является уменьшение задержек распространения сигналов за счет сокращения количества уровней логики. Значение *Balanced* позволяет сочетать рассмотренные выше стратегии оптимизации. Использование значения *Off* запрещает оптимизацию при распределении ресурсов CLB кристалла.

Далее следует перейти к странице *Place and Route Properties*, которая объединяет параметры управления процедурами размещения

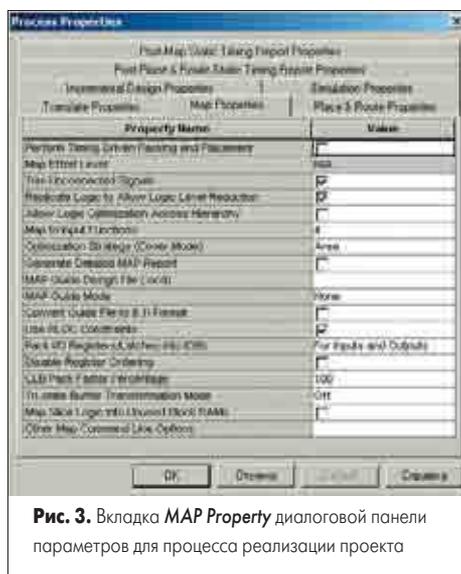


Рис. 3. Вкладка *MAP Property* диалоговой панели параметров для процесса реализации проекта

и трассировки проекта. Прежде всего, нужно с помощью параметра *Place and Route Effort Level (Overall)* указать требуемый уровень эффективности процедур размещения и трассировки проекта в кристалл. Этот параметр позволяет за счет выбора более совершенных алгоритмов размещения и трассировки (соответственно за счет увеличения времени выполнения рассматриваемого этапа) достичь более высоких результатов. И наоборот, выбирая менее сложные алгоритмы, можно сократить время выполнения процесса размещения и трассировки, но получить результаты, далекие от оптимальных. Для сложных проектов снижение времени, необходимого для выполнения размещения и трассировки, может привести к появлению неразведенных цепей и увеличению задержек распространения сигналов. В выпадающем списке возможных значений этого параметра представлено три элемента, расположенных в порядке возрастания эффективности: *Standard*, *Medium*, *High*. Значение *Standard* позволяет минимизировать время размещения и трассировки за счет низкого уровня оптимизации этих процессов. При выборе значения *High* достигаются наилучшие результаты, но требуется максимальное время для выполнения всех процедур. Можно также установить различный уровень эффективности для каждой фа-



Рис. 4. Вкладка *Place and Route Properties* диалоговой панели параметров размещения и трассировки проекта

зы этого процесса, используя параметры *Place Effort Level (Overrides Overall Level)* и *Router Effort Level (Overrides Overall Level)*, которые предназначены для отдельной установки уровня эффективности процедур размещения и трассировки соответственно.

Выбирая значение параметра *Extra Effort (Highest PAR level only)*, можно предусмотреть возможность выделения дополнительного времени для выполнения процедур размещения и трассировки, необходимого для удовлетворения различных временных ограничений проекта. Этот параметр используется только в случае, если для *Place and Route Effort Level (Overall)* задано значение *High*. По умолчанию установлено значение *None*, которое запрещает выделение дополнительного времени. При выборе других значений (*Normal* или *Continue on Impossible*) допускается увеличение времени выполнения процедур размещения и трассировки.

Значение параметра *Place and Route Mode* позволяет выбрать режим выполнения процесса размещения и трассировки. Для данного параметра рекомендуется использовать значение *Normal Place and Route*, установленное по умолчанию. Это значение соответствует нормальному режиму трассировки и размещения, при котором выполняются все процедуры с учетом параметров, указанных разработчиком или установленных по умолчанию. При выборе значения *Reentrant Route* осуществляется многократное повторение процедур трассировки с целью оптимизации результатов. Но этот режим можно устанавливать только в том случае, если хотя бы один раз полностью пройдена стадия размещения и трассировки.

Для того чтобы средствами размещения и трассировки принимались во внимание временные ограничения проекта, указанные в файле UCF, необходимо выбрать для параметра *Use Timing Constraints* значение «включено».

После установки всех требуемых значений параметров для процедур реализации проекта следует подтвердить их нажатием кнопки ОК в нижней части диалоговой панели (рис. 2–4). Далее следует активизировать процесс реализации или поочередно каждую его фазу. Запуск процесса в полном объеме осуществляется двойным щелчком левой кнопки мыши на строке *Implement Design* в окне процедур *Навигатора проекта* (рис. 1). Информация о ходе его выполнения отображается в окне консольных сообщений данной управляющей оболочки. Завершение выполнения каждой фазы процесса отмечается соответствующей пиктограммой в строке с ее названием в окне процессов и сопровождается отчетом о полученных результатах. Для просмотра отчета о выполнении фазы трансляции следует дважды щелкнуть левой кнопкой мыши на строке *Translation Report*. Доступ к отчету о результатах отображения логического описания проекта на физические ресурсы кристалла осуществляется двойным щелчком левой кнопки мыши на строке *MAP Report*. В этом отчете рекомендуется обратить внимание на раздел *Design Summary*, который представляет статистику ресурсов ПЛИС, используемых для реализации проекта аппаратной части системы. После завершения заключи-

тельной фазы (*Place-and-Route*) процесса реализации следует открыть отчет о ее выполнении, дважды щелкнув левой кнопкой мыши на строке *Place-and-Route Report* в окне процессов *Навигатора проекта* (рис. 1), и убедиться в отсутствии неразведенных цепей и сообщений об ошибках. Кроме того, целесообразно проверить отчет *Pad Report*, который описывает назначение всех выводов кристалла после загрузки конфигурационных данных проекта. В этом отчете следует проконтролировать выполнение директив закрепления выводов, которые были внесены в файл временных и топологических ограничений.

При достижении успешных результатов размещения и трассировки можно перейти непосредственно к этапу формирования конфигурационной последовательности для проекта аппаратной платформы разрабатываемой микропроцессорной системы.

### Генерация конфигурационной последовательности

Процесс создания конфигурационной последовательности для ПЛИС семейств FPGA выполняется средствами пакета Foundation ISE в автоматическом режиме. Все необходимые директивы конфигурирования и последующей активизации кристалла задаются с помощью параметров рассматриваемого процесса перед его выполнением. Вызов диалоговой панели, в которой расположены эти опции, осуществляется уже привычными способами. В окне процессов (рис. 1) щелчком левой кнопки мыши следует выделить строку *Generate Programming File*, после чего нажать кнопку , расположенную на оперативной панели управления *Навигатора проекта*, или воспользоваться командой **Properties** контекстно-зависимого всплывающего меню, которое выводится при щелчке правой кнопки мыши. Диалоговая панель параметров процесса генерации конфигурационной последовательности, вид которой показан на рис. 5, содержит четыре вкладки с названиями: *General Options*, *Configuration Options*, *Startup Options*, *Readback Options* [7].

На странице *General Options*, содержащей основные параметры процесса генерации конфигурационной последовательности, следует убедиться, что для параметра *Create Bit File* установлено значение «включено». Этот параметр используется для управления процессом создания двоичного файла последовательно битового потока (bitstream).



Рис. 5. Диалоговая панель параметров процесса генерации конфигурационной последовательности (вкладка **General Options**)

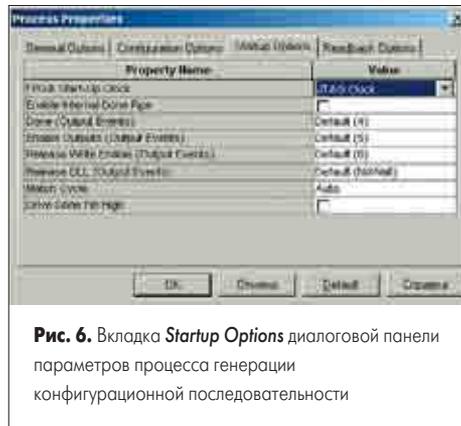


Рис. 6. Вкладка **Startup Options** диалоговой панели параметров процесса генерации конфигурационной последовательности

Далее необходимо открыть страницу *Startup Options*, где расположены параметры управления фазой активизации логики после конфигурирования кристалла (рис. 6), и выбрать источник сигнала синхронизации, используемого в процессе конфигурирования. Выбор источника сигнала синхронизации осуществляется с помощью параметра *Start-Up Clock*. Выпадающий список содержит три возможных значения: *CCLK*, *User Clock*, *JTAG Clock*. Если формируемый файл конфигурационной последовательности предназначен для последующего программирования ППЗУ, то для параметра *Start-Up Clock* следует установить значение *CCLK*. При этом в качестве сигнала синхронизации процесса активизации логики будет использоваться тактовый сигнал *CCLK*, вырабатываемый внутри кристалла FPGA. Значение *JTAG Clock* следует выбирать при загрузке конфигурационной последовательности непосредственно в кристалл FPGA в режиме периферийного сканирования через JTAG-порт. Этот режим целесообразно использовать в процессе отладки проектируемой системы.

Для установки защиты от копирования конфигурационных данных нужно активизировать страницу *Readback Options*, которая представляет параметры, используемые в процессе обратного считывания конфигурационных данных из кристалла (рис. 7). Параметр *Security* предназначен для выбора требуемого варианта защиты конфигурационных данных. В выпадающем списке представлены три возможных значения этого параметра: *Enable Readback and Reconfiguration*, *Disable Readback and Reconfiguration*. Значение *Enable Readback and Reconfiguration*, принятое по умолчанию, разрешает обратное считывание загруженной конфигурационной последовательности и реконфигурирование кри-

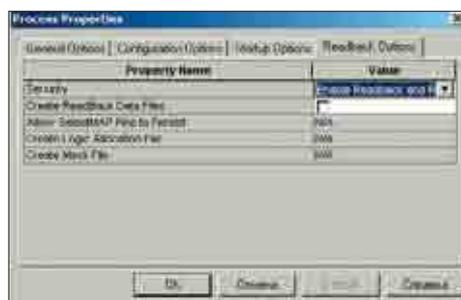


Рис. 7. Вкладка **Readback Options** диалоговой панели параметров процесса генерации конфигурационной последовательности

сталла. Выбор значения *Disable Readback and Reconfiguration* запрещает чтение конфигурационных данных из кристалла FPGA. Значение *Disable Readback and Reconfiguration* устанавливает защиту от считывания конфигурационной последовательности и реконфигурирования кристалла.

Установив требуемые значения параметров процесса генерации конфигурационной последовательности, следует подтвердить их нажатием клавиши **OK** в нижней части диалоговой панели (рис. 5–7). Затем для активизации процесса формирования файла конфигурирования необходимо дважды щелкнуть левой кнопкой мыши на строке *Generate Programming File*, расположенной в окне процедур *Навигатора проекта* (рис. 1). Информация о ходе его выполнения отображается в окне консольных сообщений и строке состояния Навигатора проекта. При успешном завершении этого процесса, отмеченном соответствующей пиктограммой в строке *Generate Programming File* в окне процессов, создается файл конфигурационного битового потока (имеющий расширение *.bit*) для аппаратной части разрабатываемой микропроцессорной системы. После окончания этапа реализации можно закрыть окно *Навигатора проекта* и вернуться в среду управляющей оболочки XPS для выполнения дальнейших этапов разработки.

Далее необходимо скопировать полученный файл конфигурационного битового потока из рабочего каталога проекта *CAIP Foundation ISE* в раздел проекта *Xilinx EDK*. Для этого следует активизировать пункт *Tools* основного меню управляющей оболочки XPS и в появившемся всплывающем меню выбрать команду *Import from ProjNav*. При этом на экран выводится диалоговая панель, вид которой показан на рис. 8. В этой панели, в поле редактирования *BIT File* необходимо указать полное название копируемого конфигурационного файла, включающее полный путь доступа к нему, а также в поле редактирования *BMM File* — полное имя файла, который описывает содержимое блочной памяти ПЛИС. Названия этих файлов могут быть введены с помощью клавиатуры после активизации соответствующего поля редактирования. Для поиска этих файлов можно также воспользоваться кнопками с пиктограммой в виде многоточия (...), которые расположены справа от каждого поля. При нажатии на эти кнопки на экран выводится стандартная диалоговая панель открытия файла, используя которую нужно выбрать требуемые файлы. Копирование указанных файлов выполняется сразу после нажатия клавиши **OK**, расположенной в нижней части диалоговой панели (рис. 8).

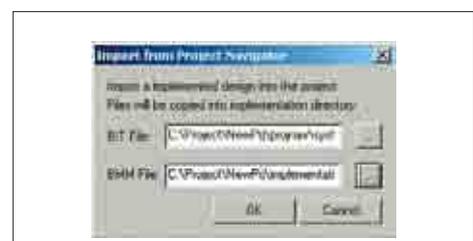


Рис. 8. Диалоговая панель копирования файла конфигурационной последовательности в рабочий каталог проекта Xilinx EDK

На этом фактически завершается процесс разработки аппаратной платформы встраиваемой микропроцессорной системы на основе 32-разрядного ядра MicroBlaze. Сформированный файл конфигурационной последовательности необходимо дополнить информацией о содержимом блочной памяти ПЛИС, на основе которой реализуется ППЗУ программ. Вопросы создания программного обеспечения для проектируемой микропроцессорной системы будут рассмотрены в следующей публикации цикла. ■

### **Литература**

1. Зотов В. MicroBlaze — семейство тридцатидвухразрядных микропроцессорных ядер, реализуемых на основе ПЛИС фирмы Xilinx // Компоненты и технологии. 2003. № 9.
2. Зотов В. Система команд микропроцессорного ядра MicroBlaze // Компоненты и технологии. 2004. № 1–3.
3. Зотов В. Embedded Development Kit — система проектирования встраиваемых микропроцессорных систем на основе ПЛИС серий FPGA фирмы Xilinx // Компоненты и технологии. 2004. № 4.
4. Зотов В. Организация памяти микропроцессорного ядра MicroBlaze // Компоненты и технологии. 2004. № 5.
5. Зотов В. Создание проекта микропроцессорной системы на основе ядра MicroBlaze, реализуемой в ПЛИС семейств FPGA фирмы Xilinx // Компоненты и технологии. 2004. № 6.
6. Зотов В. Формирование спецификации аппаратной платформы микропроцессорной системы на основе ядра MicroBlaze, реализуемой в ПЛИС семейств FPGA фирмы Xilinx // Компоненты и технологии. 2004. №№ 7–8.
7. Зотов В. Проектирование цифровых устройств на основе ПЛИС фирмы Xilinx в САПР WebPack ISE. М.: Горячая линия — Телеком. 2003.
8. Кузелин М. О., Кнышев Д. А., Зотов В. Ю. Современные семейства ПЛИС фирмы Xilinx. Справочное пособие. М.: Горячая линия — Телеком. 2004.