

Создание проекта микропроцессорной системы

на основе ядра MicroBlaze, реализуемой в ПЛИС семейств FPGA фирмы Xilinx

Продолжая цикл публикаций [1–4] о семействе 32-разрядных микропроцессорных ядер MicroBlaze™, предназначенных для реализации на основе FPGA фирмы Xilinx, переходим к практическому изучению этапов проектирования встраиваемых систем в рамках пакета Xilinx Embedded Development Kit (EDK). В настоящей статье рассматривается структура процесса разработки микропроцессорных систем, в качестве основы которых используется ядро MicroBlaze, а также последовательность необходимых операций для создания нового проекта в САПР Xilinx EDK версии 6.2.

Валерий Зотов

walerry@euro.ru

Этапы проектирования микропроцессорных систем на основе ядра MicroBlaze, реализуемых в FPGA фирмы Xilinx

Процесс разработки встраиваемых микропроцессорных систем на основе ядра MicroBlaze, предназначенных для реализации в FPGA фирмы Xilinx [5–8], можно интерпретировать в виде двух маршрутов последовательных этапов проектирования. Первый маршрут — разработка аппаратной части встраиваемой микропроцессорной системы. Второй — проектирование программных средств (рис. 1).

Маршрут проектирования аппаратной части системы состоит из следующих этапов:

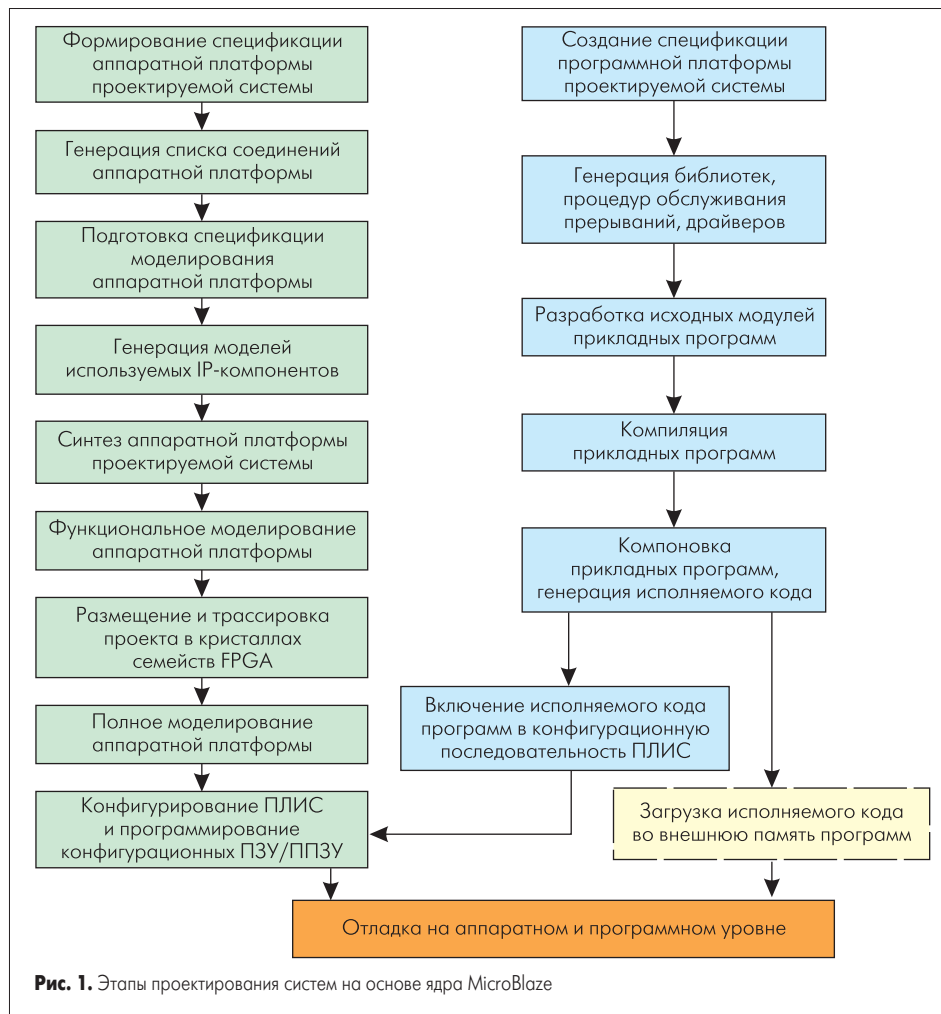
- формирование спецификации аппаратной платформы системы;
- генерация HDL-описаний и списка соединений аппаратной платформы;
- подготовка спецификации моделирования аппаратной части разрабатываемой системы;
- генерация моделей IP-компонентов, используемых в составе проекта, для последующей верификации разрабатываемой системы;
- синтез аппаратной платформы проектируемой системы;
- функциональное моделирование аппаратной части системы;
- размещение и трассировка проекта в кристаллах FPGA;
- полное (временное) моделирование разрабатываемой системы;
- загрузка конфигурационной последовательности в кристалл FPGA и программирование конфигурационных ПЗУ или ППЗУ.

Не все из перечисленных этапов являются обязательными. В процессе разработки может не проводиться моделирование аппаратной части системы. Поэтому можно исключить этапы подготовки спецификации моделирования, генерации моделей, функционального и временного моделирования. При этом следует все же учитывать, что моделирование аппаратной части системы позволяет повысить эффективность процесса проектирования в целом за счет более раннего обнаружения возможных ошибок.

Маршрут разработки программных средств состоит из следующих этапов:

- создание спецификации программной платформы разрабатываемой системы;
- формирование библиотек, драйверов и процедур обслуживания прерываний в соответствии со спецификацией программной платформы системы;
- разработка исходных модулей прикладных программ (приложений) на языке ассемблера или языках высокого уровня C/C++;
- компиляция исходных модулей прикладных программ;
- компоновка программ, генерация исполняемого кода;
- преобразование исполняемого кода программ в загружаемый код кристаллов FPGA;
- загрузка исполняемого кода программ во внешнюю программную память;
- комплексная отладка прикладных программ для проектируемой системы с использованием аппаратных и программных средств.

Все перечисленные этапы разработки встраиваемых микропроцессорных систем на основе ядра



MicroBlaze выполняются с помощью комплекса средств проектирования Xilinx EDK. Прежде чем перейти непосредственно к детальному изучению представленных этапов, необходимо определить понятие проекта, которое будет использоваться в последующих разделах.

Структура проекта в Embedded System Tools

Под проектом в САПР встраиваемых микропроцессорных систем Embedded System Tools (EST) понимается совокупность модулей (файлов), которые содержат информацию, необходимую для выполнения всех этапов процесса разработки встраиваемых систем на базе ПЛИС Xilinx. В структуре проекта EST можно выделить следующие группы модулей:

- исходные описания аппаратной и программной платформ проектируемой системы в графической или текстовой форме;
- IP-компоненты, разработанные пользователем;
- библиотеки, драйверы и процедуры обработки прерываний;
- прикладное программное обеспечение проектируемой микропроцессорной системы;
- документация, сопровождающая проект;
- промежуточные результаты, используемые в качестве исходных данных для последующих этапов проектирования;
- отчеты о выполнении отдельных этапов проектирования;

- описания тестовых воздействий, необходимых для моделирования проектируемой системы;
- окончательные результаты проектирования, используемые для конфигурирования ПЛИС.

Все модули проекта располагаются в рабочем каталоге (папке), включающем несколько подкаталогов, каждый из которых содержит файлы. Изначально в состав нового проекта входит файл с расширением XMP, в котором указываются параметры проекта, и пустые

файлы спецификаций аппаратной и программной платформ. Затем к проекту добавляются модули описания аппаратной платформы проектируемого устройства и исходные модули прикладных программ. Далее, после выполнения каждого этапа разработки системы в проект включаются результаты, полученные на этом этапе. Кроме того, разработчик может включить необходимую текстовую документацию.

Создание нового проекта в среде управляющей оболочки Xilinx Platform Studio

Работа с комплексом средств автоматизированного проектирования Xilinx Embedded Development Kit начинается с запуска управляющей оболочки Xilinx Platform Studio с помощью кнопки *Пуск (Start)*. После нажатия кнопки *Пуск (Start)* в открывшейся панели меню необходимо выбрать строку *Программы (Programs)*, а затем в предложенном списке выбрать группу программ *Xilinx Embedded Development Kit*, в которой нужно выделить строку *Xilinx Platform Studio* и щелкнуть на ней левой кнопкой мышки. При успешном выполнении указанных операций на экране монитора отображается основное окно *Xilinx Platform Studio*, подробная структура которого рассмотрена ранее [3].

Для создания нового проекта следует выполнить команду *File* основного меню управляющей оболочки Xilinx Platform Studio, а затем во всплывающем меню выбрать строку *New Project* (рис. 2). В результате открывается всплывающее меню следующего уровня иерархии, строки которого представляют разработчику два механизма подготовки нового проекта.

При выборе строки *Base System Builder* (рис. 2) запускается «мастер» формирования новых аппаратных платформ *Base System Build Wizard*, который позволяет проводить весь процесс разработки нового проекта в интерактивном режиме. Данный метод наиболее целесообразно применять при выпол-

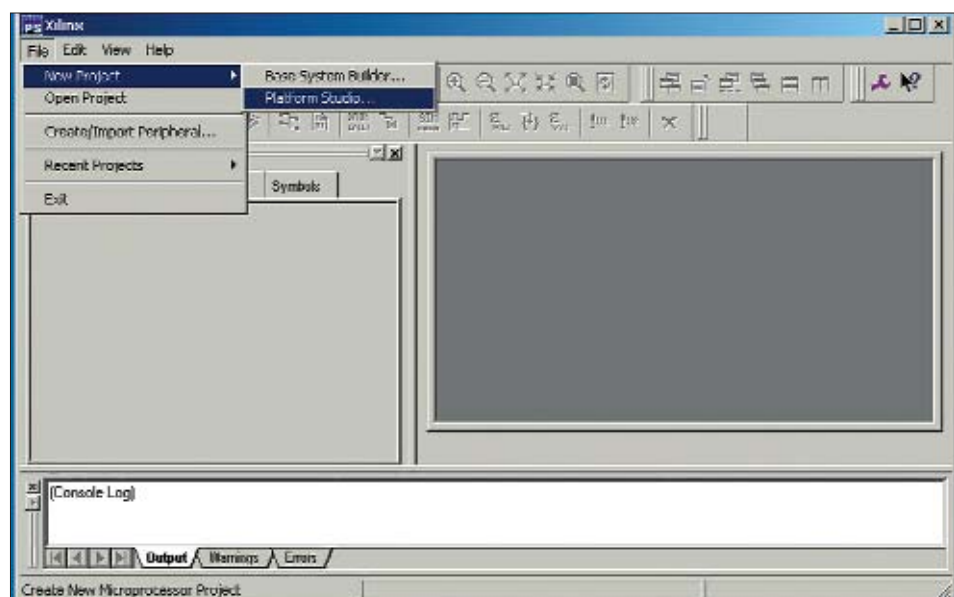


Рис. 2. Последовательность действий по созданию нового проекта в Xilinx Embedded Development Kit

нении проектов, для аппаратной реализации которых используются отладочные и демонстрационные платы, поддерживаемые пакетом Xilinx EDK [3]. В общем случае применяется стандартный способ создания нового проекта разрабатываемой микропроцессорной системы, доступ к которому открывает строка *Platform Studio...* во всплывающем меню (рис. 2). При выборе этой строки на экран монитора выводится диалоговая панель (рис. 3). В этой панели для создания нового проекта обязательно должны быть определены следующие исходные данные:

- название проекта;
- диск и каталог, в котором предполагается расположить проект;
- семейство ПЛИС, на базе которого разрабатывается система;
- тип кристалла, используемого для реализации проектируемой системы;
- тип корпуса выбранной ПЛИС;
- категория быстродействия выбранного кристалла.

Кроме того, в этой же панели можно указать дополнительно следующую необязательную информацию:

- название уже существующего файла спецификации аппаратной платформы встраиваемой микропроцессорной системы MHS (Microprocessor Hardware Specification), который может использоваться, например, в качестве шаблона в создаваемом проекте;
- возможность использования в новом проекте депозитария собственных (разработанных пользователем) IP-ядер периферийных модулей, библиотек и драйверов;
- полное название каталогов, которые используются в качестве депозитария IP-компонентов, библиотек и драйверов.

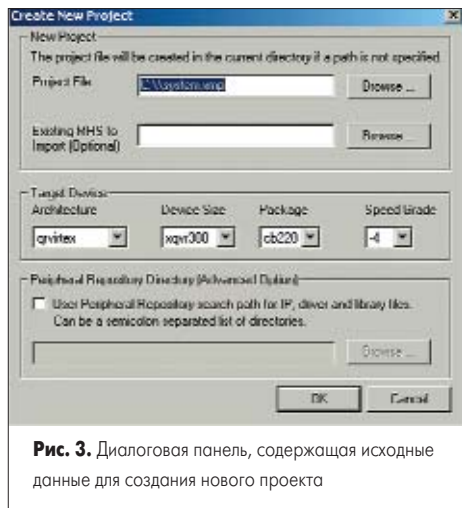



Рис. 3. Диалоговая панель, содержащая исходные данные для создания нового проекта

В первую очередь рекомендуется указать полное название файла создаваемого проекта, включающее путь доступа к разделу (папке), в котором будет располагаться рабочий каталог проекта. Полное наименование файла проекта указывается в поле редактирования *Project File* (рис. 3). Целесообразно хранить все проекты в специально сформированной для этих целей папке, например, *C:\Project*. Таким образом, рабочий каталог нового проекта должен создаваться внутри этой папки. Рекомендуется задавать мнемо-

нические имена проектов и их рабочих каталогов, чтобы впоследствии было удобно ориентироваться при поиске требуемого проекта. По умолчанию в поле редактирования *Project File* предлагается в качестве нового проекта файл *system.xmp*, располагающийся в корневом каталоге диска, на котором находится комплекс средств автоматизированного проектирования Xilinx EDK. Определить местоположение создаваемого проекта можно двумя способами: используя клавиатуру или панель навигации по диску компьютера. При выборе первого метода следует поместить курсор на поле редактирования *Project File* и щелкнуть левой кнопкой мышки, после чего ввести с клавиатуры имя диска, каталога и название файла проекта. При этом указываемый каталог на диске должен быть создан заранее. Для выбора каталога, который уже существует, удобнее воспользоваться вторым способом, нажав кнопку *Browse...*, расположенную справа от поля редактирования *Project File* (рис. 3). В открывшейся панели навигации (рис. 4) следует с помощью мышки выбрать требуемый диск и каталог, а затем подтвердить сделанный выбор нажатием кнопки *OK*. Для создания нового раздела нужно щелкнуть левой кнопкой мышки на пиктограмме , которая находится в верхней части панели навигации. При этом в текущем разделе создается пустой каталог с названием «Новая папка». Для изменения названия новой папки следует прежде всего выделить его щелчком левой кнопкой мышки. Затем следует набрать текст нового названия с клавиатуры, завершив ввод нажатием клавиши *Enter*. Далее следует выполнить выбор новой папки щелчком левой кнопкой мышки. После закрытия панели навигации выбранные параметры автоматически отображаются в поле редактирования *Project File*. В качестве примера создадим проект с названием *system_tst*, расположив его рабочий каталог в папке *C:\Project*.

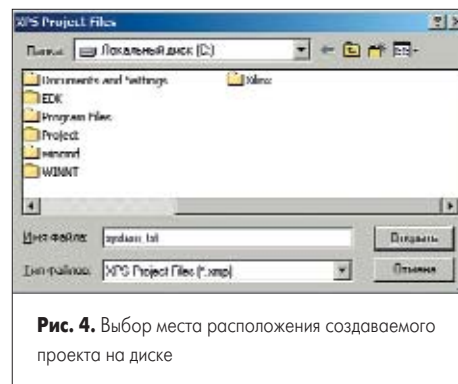


Рис. 4. Выбор места расположения создаваемого проекта на диске

Семейство ПЛИС, тип кристалла, тип корпуса ПЛИС и категория быстродействия кристалла определяются с помощью соответствующих полей выбора, расположенных внутри встроенной панели *Target Device*.

Для определения семейства ПЛИС, на базе которого проектируется встраиваемая микропроцессорная система, следует воспользоваться кнопкой управления выпадающим списком, расположенной в правой части поля выбора семейства *Architecture* (рис. 3). При нажатии на эту кнопку отображается

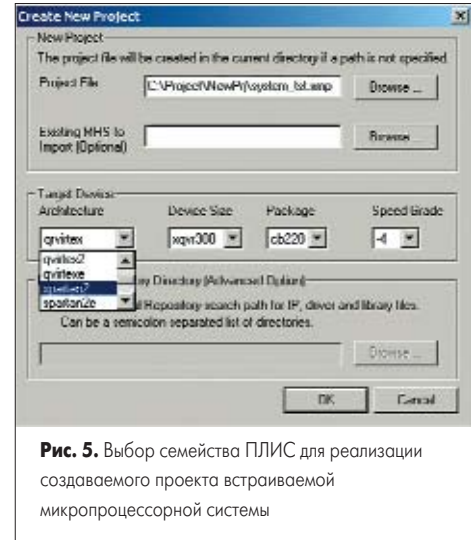


Рис. 5. Выбор семейства ПЛИС для реализации создаваемого проекта встраиваемой микропроцессорной системы

полный список семейств ПЛИС, поддерживаемых пакетом Xilinx EDK (рис. 5).

Чтобы выбрать требуемое семейство ПЛИС, необходимо щелкнуть левой кнопкой мышки на соответствующей строке выпадающего списка. Если нужное семейство не представлено в отображаемой части списка, то следует воспользоваться элементами прокрутки, которые расположены с правой стороны списка (рис. 5). После выбора одной из строк списка название соответствующей серии ПЛИС автоматически отображается в поле выбора семейства *Architecture*. Для примера, в создаваемом проекте *system_tst* укажем семейство *Spartan-II*.

В поле выбора типа кристалла, предназначенного для реализации проектируемой микропроцессорной системы, *Device Size* автоматически отображается тип ПЛИС, установленный по умолчанию для выбранного семейства. Для его изменения необходимо выполнить последовательность действий, аналогичную процедуре установки семейства ПЛИС. Вначале нужно нажать кнопку активизации выпадающего списка доступных кристаллов выбранного семейства — в правой части поля выбора типа кристалла *Device Size* (рис. 6).

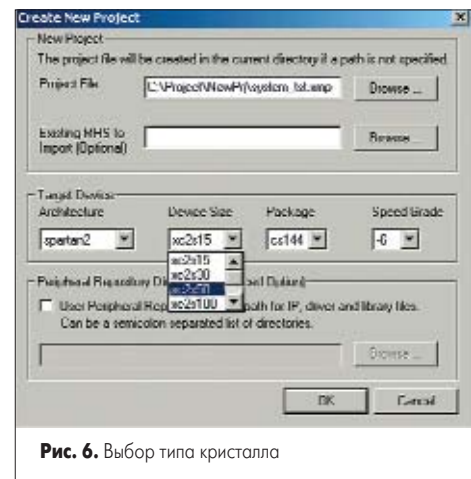


Рис. 6. Выбор типа кристалла

В предложенном списке нужно найти строку, содержащую требуемый тип кристалла, и щелкнуть на ней левой кнопкой мышки. При необходимости для просмотра всего списка ПЛИС можно использовать инструменты прокрутки, расположенные в правой части выпадающего списка. Выделенный элемент

списка автоматически отображается в поле выбора типа кристалла *Device Size*. Для реализации проекта *system_tst*, создаваемого в качестве примера, выберем кристалл XC2S50.

Тип корпуса ПЛИС указывается в поле выбора *Package* (рис. 3). Для доступа к полному списку типов корпусов, в которых выпускается требуемый кристалл, следует воспользоваться кнопкой активизации выпадающего списка, расположенной в правой части поля выбора *Package*. В открывшемся списке (рис. 7) нужно щелкнуть левой кнопкой мышки на строке, в которой представлено условное обозначение искомого типа корпуса ПЛИС. После этого выбранный тип корпуса микросхемы автоматически отображается в поле *Package*. В качестве примера в новом проекте *system_tst* выберем кристалл в корпусе FG256.

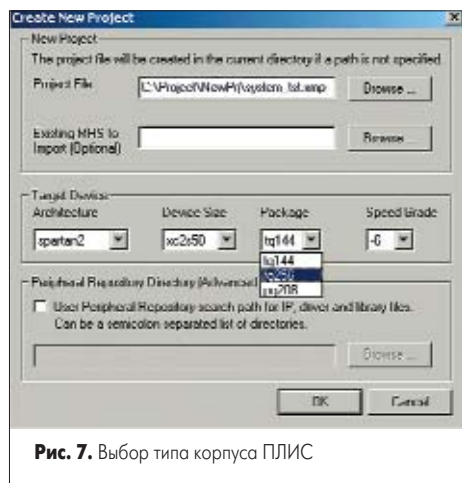


Рис. 7. Выбор типа корпуса ПЛИС

Для определения категории быстродействия кристалла, на базе которого планируется реализация проектируемой микропроцессорной системы, следует воспользоваться полем выбора *Speed Grade* (рис. 3). При нажатии на кнопку активизации выпадающего списка, расположенную в правой части этого поля выбора, на экран выводятся обозначения возможных градаций быстродействия используемого типа ПЛИС (рис. 8). Требуемая категория быстродействия кристалла указывается щелчком левой кнопки мышки на строке, в которой содержится условное обозначение соответствующего класса быстродействия ПЛИС. Для реализации проекта *system_tst* выберем кристалл XC2S50, имеющий класс быстродействия -5.

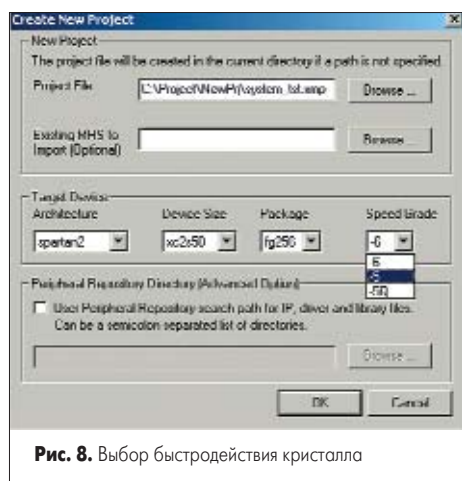


Рис. 8. Выбор быстродействия кристалла

При создании нового проекта можно импортировать файл спецификации аппаратной платформы MHS из других проектов. Для указания названия уже существующего файла MHS щелкните левой кнопкой мышки на поле *Existing MHS to Import* (рис. 3), после чего введите с клавиатуры полное имя соответствующего файла, включая полный путь доступа к нему. Для поиска файла спецификации можно также воспользоваться панелью навигации по дискам компьютера. Для ее активизации следует нажать кнопку *Browse...*, расположенную справа от поля редактирования *Existing MHS to Import*. В открывшейся панели, вид которой аналогичен показанному на рис. 4, следует выбрать диск, каталог и название файла, после чего подтвердить сделанный выбор нажатием кнопки *OK*. После закрытия панели навигации полное название импортируемого файла спецификации автоматически отображается в поле редактирования *Existing MHS to Import*.

Если в новом проекте предполагается применение пользовательских IP-ядер периферийных модулей, библиотек и драйверов, то во встроенной панели дополнительных параметров *Advanced Option* (рис. 3) следует, прежде всего, включить опцию *User Peripheral Repository search path for IP, driver and library files*, щелкнув левой кнопкой мышки на поле индикатора состояния, который расположен слева от названия опции. После этого станет доступным поле редактирования названий каталогов, в которых должен проводиться поиск IP-ядер периферийных модулей, библиотек и драйверов. Названия каталогов могут быть введены непосредственно в этом поле (после его активизации) с помощью клавиатуры. При вводе нескольких названий каталогов их нужно разделять знаком «точка с запятой». Каталоги могут быть также указаны с помощью панели навигации по дискам компьютера, которая выводится на экран кнопкой *Browse...*, расположенной справа от поля редактирования.

Процедура определения всех необходимых параметров создаваемого проекта завершается нажатием клавиши *OK* (рис. 3). Если при указании параметров нового проекта не было задано название импортируемого файла спецификации аппаратной части встраиваемой микропроцессорной системы, то после нажатия кнопки *OK* на экран выводится диалоговая панель (рис. 9). В этой панели содержится предупреждение об отсутствии названия импортируемого файла MHS и запрос подтверждения создания нового проекта с пустым файлом MHS. При отрицательном ответе (нажатии кнопки *No*) осуществляется возврат к диалоговой панели, показанной на рис. 3.

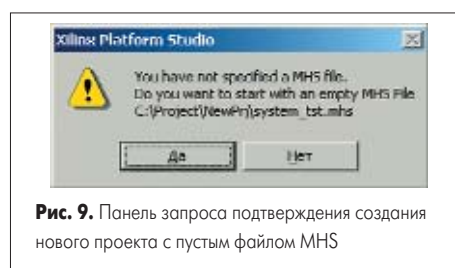


Рис. 9. Панель запроса подтверждения создания нового проекта с пустым файлом MHS

Для формирования нового проекта необходимо утвердительно ответить на этот запрос, нажав кнопку *Yes* (рис. 9). В этом случае на экране отображается информационная панель с уведомлением о способах добавления IP-ядер в спецификацию аппаратной платформы. Вид этой информационной панели представлен на рис. 10. Для ее закрытия следует воспользоваться кнопкой *OK*, расположенной в нижней части панели.

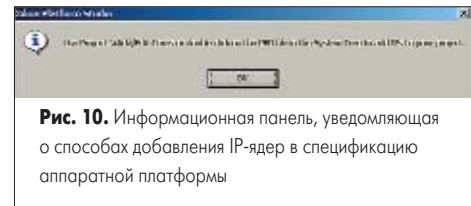


Рис. 10. Информационная панель, уведомляющая о способах добавления IP-ядер в спецификацию аппаратной платформы

В случае успешного создания нового проекта его название отображается в заголовке окна Xilinx Platform Studio, которое приобретает вид, показанный на рис. 11. Во встроенном окне проекта, на странице *System*, в разделе *Project Files* отображается список названий пустых исходных файлов проекта. Раздел *Project Options* содержит информацию об установленных значениях основных параметров проекта.

Выбрав в главном меню пункт *Options*, а затем во всплывающем меню строку *Project Options*, можно внести изменения в значения основных параметров проекта. В результате на экран выводится диалоговая панель *Project Options* (рис. 12), которая позволяет установить и отредактировать значения всех параметров проекта. Для открытия этой панели можно также воспользоваться аналогичной командой из контекстно-зависимого всплывающего меню, которое появится при щелчке правой кнопкой мышки на строке, содержащей значение одного из параметров проекта. Самым быстрым способом доступа к панели установки параметров является двойной щелчок левой кнопкой мышки на строке *Project Options* во встроенном окне проекта.

Диалоговая панель установки параметров проекта содержит три страницы, снабженные закладками с их названиями: *Device and Repository*, *Hierarchy and Flow* и *HDL and Simulation*. Каждая из этих страниц содержит соответствующую группу параметров. Страница *Device and Repository* позволяет изменить семейство ПЛИС, тип кристалла, тип корпуса ПЛИС и категорию быстродействия кристалла, а также название каталогов, в которых должен проводиться поиск IP-ядер периферийных модулей, библиотек и драйверов. Установка значений этих параметров выполняется теми же способами, что и в диалоговой панели, представленной на рис. 3. Кроме того, на этой странице предусмотрена возможность определения названия собственного пользовательского командного (Make) файла, содержащего все необходимые опции и директивы управления процессом проектирования. Указанный файл будет использоваться вместо Make-файла, генерируемого средствами EST. Полное название альтернативного командного Make-файла, включающее путь доступа к нему, указывается в поле редактирования *Custom Makefile*. Определе-

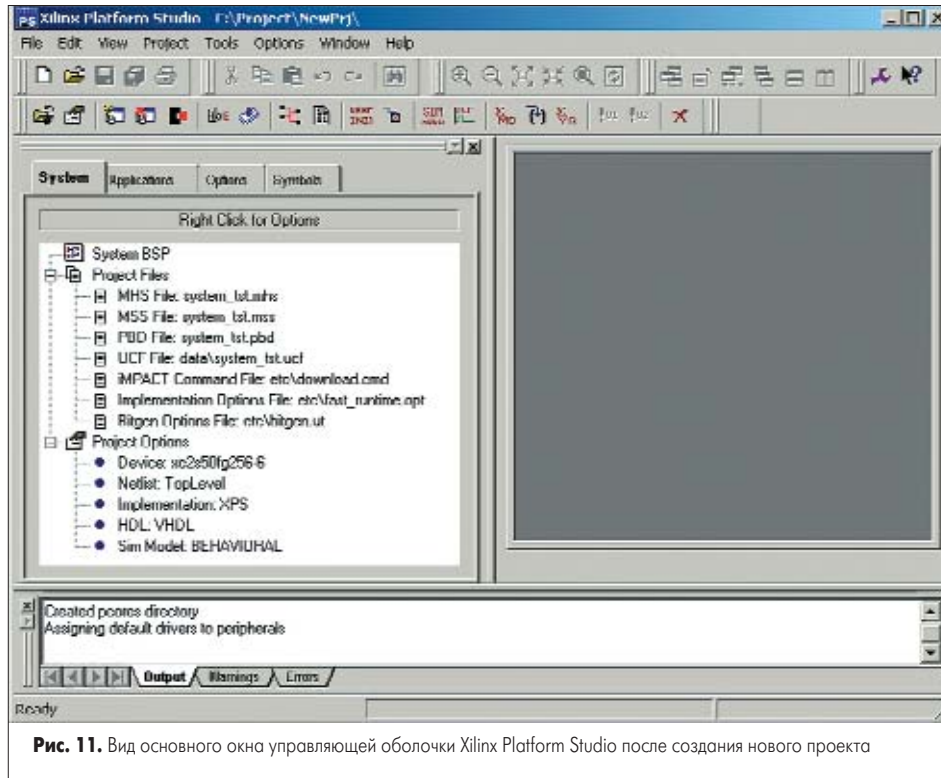


Рис. 11. Вид основного окна управляющей оболочки Xilinx Platform Studio после создания нового проекта

ние имени файла может осуществляться с помощью клавиатуры (после активизации поля редактирования) или панели навигации по дискам компьютера, которая выводится на экран при нажатии кнопки с пиктограммой в виде многоточия (...), расположенной справа от поля редактирования *Custom Makefile*.

На странице *Hierarchy and Flow* диалоговой панели *Project Options* представлены параметры, определяющие иерархию, средства синтеза и способ управления процессом размещения и трассировки проекта в кристалле (рис. 13).

Иерархический уровень разрабатываемого проекта устанавливается с помощью двух кнопок с зависимой фиксацией, которые расположены во встроенной панели *Design Hierarchy*. Если создаваемый проект соответствует верхнему уровню иерархии, то в нажатом состоянии должна находиться кнопка *This is the top level of my design*. В случае, когда разрабатываемый проект является одним из модулей, входящим в состав системы (про-

екта более высокого уровня иерархии), должна быть выбрана кнопка *This is a sub-module in my design*. При этом становится активным поле редактирования *Top Instance*, в котором с помощью клавиатуры можно указать позиционное обозначение компонента, представляющего проектируемый модуль, в составе проекта системы верхнего уровня иерархии. По умолчанию считается, что создаваемый проект соответствует верхнему уровню иерархии, то есть выбрана кнопка *This is the top level of my design*.

Поле выбора *Synthesis Tool* для просмотра всех доступных инструментов синтеза и выбора наиболее подходящего следует воспользоваться кнопкой активизации выпадающего списка, которая расположена в правой части поля *Synthesis Tool*. По умолчанию используются средства синтеза Xilinx Synthesis Tool (XST), которые входят в состав системы автоматизированного проектирования Foundation ISE.

Способ управления процессом размещения и трассировки проекта разрабатываемой системы в кристалле определяется с помощью двух кнопок с зависимой фиксацией, которые расположены во встроенной панели *Implementation Tool Flow*. Если выбрана кнопка *XPS (Xflow)*, то этап размещения и трассировки проекта осуществляется в среде управляющей оболочки САПР EST, Xilinx Platform Studio. В случае выбора кнопки *ISE (ProjNav)* процесс размещения и трассировки выполняется в среде управляющей оболочки пакета Foundation ISE, Project Navigator. При этом проект, разрабатываемый средствами EST, может быть включен в виде модуля в состав проекта, выполняемого в среде САПР Foundation ISE. Для этого нужно щелчком левой кнопкой мышки включить опцию *Add modules to existing NPL file* (рис. 13). Затем в поле редактирования *NPL File* следует указать полное название соответствующего файла проекта (с расширением NPL) для па-

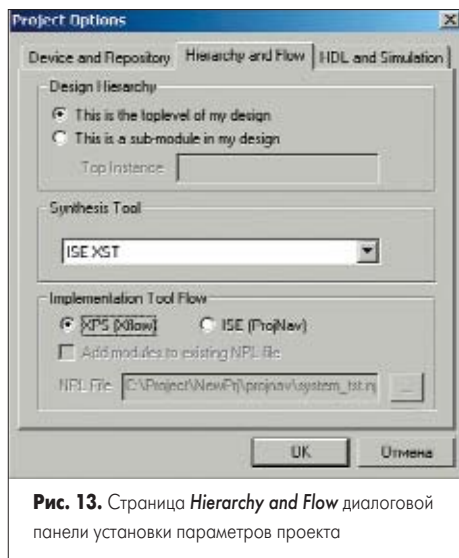


Рис. 13. Страница *Hierarchy and Flow* диалоговой панели установки параметров проекта

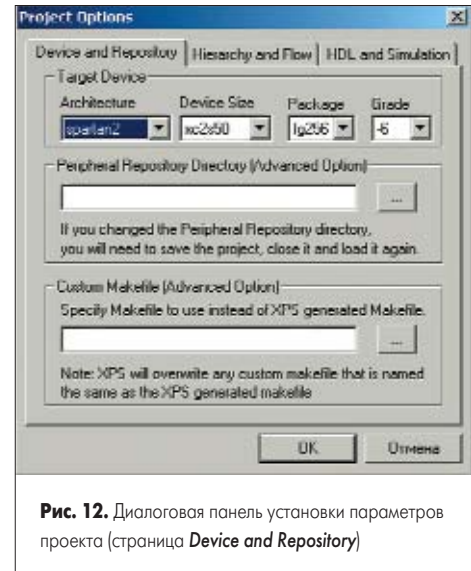


Рис. 12. Диалоговая панель установки параметров проекта (страница *Device and Repository*)

кета Foundation ISE. Имя файла проекта может задаваться теми же способами, что и название командного файла *Custom Makefile* на странице *Device and Repository*.

Страница *HDL and Simulation* (рис. 14) содержит три группы кнопок с зависимой фиксацией и два поля редактирования, определяющих значения параметров моделирования проекта. Первая группа кнопок с зависимой фиксацией, расположенная во встроенной панели *HDL*, предназначена для выбора языка описания аппаратуры HDL, который используется в процессе генерации моделей и описаний аппаратной части проектируемой системы. С помощью этих кнопок разработчик может выбрать один из языков высокого уровня: VHDL или Verilog. По умолчанию используется язык VHDL.

Вторая группа кнопок с зависимой фиксацией, находящаяся внутри встроенной панели *Simulator Compile Script*, позволяет указать систему моделирования, для которой будет сформирован соответствующий файл сценария. Если в активном состоянии зафиксирована кнопка *ModelSim*, то будет создан файл сценария моделирования для системы ModelSim XE. При выборе кнопки *NCsim* файл сценария формируется для одноименной системы моделирования. Кнопка *None* позволяет отключить генерацию файла сце-

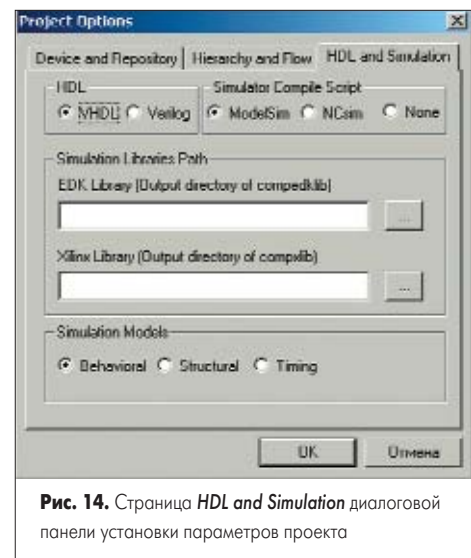


Рис. 14. Страница *HDL and Simulation* диалоговой панели установки параметров проекта

нария моделирования. В качестве системы моделирования, выбранной по умолчанию, используется пакет ModelSim XE.

Тип формируемой модели определяется с помощью третьей группы кнопок с зависимой фиксацией *Simulation Models*:

- *Behavioral* — создание функциональной модели разрабатываемой системы на поведенческом уровне;
- *Structural* позволяет сформировать модель на структурном уровне;
- *Timing* — генерация полной (временной) модели.

По умолчанию генерируется поведенческая модель проектируемой системы.

В процессе моделирования встраиваемых микропроцессорных систем, разрабатываемых с помощью комплекса средств Xilinx EDK, используются две группы библиотек Xilinx Library и EDK Library. Прежде чем перейти непосредственно к верификации проекта, необходимо скомпилировать эти библиотеки для выбранной системы моделирования. При этом можно указать названия ка-

талогов, в которых планируется разместить скомпилированные библиотеки. Для определения названий этих каталогов следует воспользоваться полями редактирования *EDK Library* и *Xilinx Library* соответственно, которые расположены внутри встроеной панели *Simulation Libraries Path* (рис. 14).

Все выполненные изменения параметров проекта вступают в силу только после нажатия кнопки *OK* в нижней части диалоговой панели (рис. 12–14).

После определения всех необходимых параметров создаваемого проекта можно перейти к следующему этапу разработки встраиваемой микропроцессорной системы на основе ядра MicroBlaze — созданию спецификации аппаратной платформы, который будет рассмотрен в следующей публикации цикла.

Литература

1. Зотов В. MicroBlaze — семейство 32-разрядных микропроцессорных ядер, реали-

зуемых 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. Кнышев Д. А., Кузелин М. О. ПЛИС фирмы «Xilinx»: описание структуры основных семейств — М.: Издательский дом «Додэка-XXI». 2001.
6. Кузелин М. ПЛИС фирмы Xilinx: семейство Spartan™-II // Компоненты и технологии. 2001. № 3.
7. Кузелин М. ПЛИС фирмы Xilinx: семейство Virtex™-II // Chip News. Инженерная микроэлектроника. 2002. № 2.
8. Spartan-3 FPGA. Handbook. Xilinx Inc. 2003.