

Формирование спецификации аппаратной платформы микропроцессорной системы на основе ядра MicroBlaze, реализуемой в ПЛИС семейств FPGA фирмы Xilinx

В предыдущей публикации цикла [1–5] был детально рассмотрен процесс создания нового проекта микропроцессорной системы на основе тридцатидвухразрядного ядра MicroBlaze в САПР Xilinx Embedded Development Kit (EDK). Следующим этапом разработки встраиваемой микропроцессорной системы, реализуемой в ПЛИС семейств FPGA фирмы Xilinx [6] с использованием ядра MicroBlaze, является формирование спецификации аппаратной платформы. Настоящая статья посвящена изучению различных способов подготовки спецификации аппаратной платформы для разрабатываемой микропроцессорной системы с помощью комплекса средств проектирования Embedded System Tools (EST). Но прежде чем перейти непосредственно к рассмотрению этапа подготовки спецификации аппаратной платформы, необходимо познакомиться с синтаксисом соответствующего файла.

Валерий Зотов

walerry@km.ru

Синтаксис файла спецификации аппаратной платформы разрабатываемой микропроцессорной системы

Спецификация аппаратной платформы встраиваемой микропроцессорной системы MHS (Microprocessor Hardware Specification) представляет собой файл с расширением MHS, в котором в текстовом формате описывается конфигурация и параметры аппаратных средств проектируемой системы. Язык описания спецификации аппаратной платформы имеет много общего с языками описания аппаратуры HDL (Hardware Description Language) высокого уровня VHDL и Verilog. В частности, интерфейсные цепи разрабатываемой системы и ее компонентов описываются в виде портов. Как и в языках HDL высокого уровня, понятие сигнала соответствует физической цепи, которая соединяет компоненты микропроцессорной системы.

Каждый компонент, включаемый в состав спецификации аппаратной платформы, описывается с помощью следующей конструкции:

```
BEGIN <идентификатор_компонента>
Команда1 <параметр1_команды1> = <значение_ параметра1> [<параметр2_команды1> = <значение_ параметра2>, ..., <параметрM_команды1> = <значение_ параметраM>]
...
КомандаN <параметр1_командыN> = <значение_ параметра1> [<параметр2_командыN> = <значение_ параметра2>, ..., <параметрK_командыN> = <значение_ параметраK>]
END
```

В приведенной конструкции квадратными скобками выделены параметры команд, наличие которых зависит от типа команды и не является обязательным.

Ключевое слово BEGIN обозначает начало описания нового компонента системы. Далее следует совокупность команд, с помощью которых указыва-

ются параметры компонента, включаемого в состав спецификации. При необходимости к данной совокупности добавляются команды, задающие границы адресного пространства, которое используется описываемым компонентом. Кроме того, здесь же присутствует группа команд, определяющих подключение данного компонента к шинам и цепям разрабатываемой системы. Общее число команд N в этой конструкции обуславливается в первую очередь типом применяемого компонента. Количество параметров, указываемых в каждой команде, определяется типом этой команды. В дальнейшем параметры команд будем называть ключами, чтобы не путать с параметрами компонентов, включаемых в состав разрабатываемой системы. Блок описания компонента проектируемой системы завершается ключевым словом END.

В текущей версии спецификации аппаратной платформы используется три команды: PARAMETER, PORT и BUS_INTERFACE. С помощью команды PARAMETER определяются значения различных параметров спецификации и применяемых компонентов. Кроме того, эта команда используется для указания диапазона адресного пространства, используемого компонентами разрабатываемой микропроцессорной системы. Все параметры, определяемые командой PARAMETER, можно разделить на две группы: глобальные и локальные. Глобальные параметры относятся ко всей спецификации аппаратной платформы. Команды, которые определяют значения глобальных параметров, располагаются вне блоков описания компонентов BEGIN-END. Примером глобального параметра является параметр VERSION, который указывает номер версии спецификации аппаратной платформы встраиваемой микропроцессорной системы. Значение этого параметра обычно задается в начале файла MHS.

Формат команды, определяющей номер версии спецификации аппаратной платформы, выглядит следующим образом:

```
PARAMETER VERSION = <номер_версии_MHS>
```

Например, в командной строке «PARAMETER VERSION = 2.1.0.» указан номер версии спецификации MHS, принятый в САПР Xilinx EDK 6.2.

Локальные параметры относятся только к соответствующим компонентам проектируемой системы. Команды, определяющие значения локальных параметров, располагаются внутри блоков описания компонентов BEGIN-END. Значение параметра HW_VER указывает номер версии используемого IP-компонента. Команда, с помощью которой определяется значение этого параметра, имеет следующий формат:

```
PARAMETER HW_VER = <номер_версии_IP-компонента>
```

Номера версий соответствующего IP-компонента указаны в файле документации для этого компонента. В качестве примера можно привести команду «PARAMETER HW_VER = 1.00.a», которая указывает на то, что в соответствующем блоке BEGIN-END описывается экземпляр компонента версии 1.00.a.

С помощью параметра INSTANCE задается позиционное обозначение (идентификатор) описываемого экземпляра используемого IP-компонента. Для определения значения этого параметра следует включить в состав соответствующего блока описания BEGIN-END командную строку, формат которой имеет следующий вид:

```
PARAMETER INSTANCE = <идентификатор_экземпляра_IP-компонента>
```

Например, командная строка «PARAMETER INSTANCE = uart1» определяет идентификатор uart1 для описываемого экземпляра IP-компонента, который представляет собой последовательный асинхронный приемопередатчик.

Диапазон адресов памяти, используемый описываемым экземпляром некоторого IP-компонента проектируемой системы, задается в виде базового и максимального значений адреса. Размер адресного пространства, выделяемого для компонента микропроцессорной системы, должен быть равен 2 в степени N. При этом последние N бит в значении базового адреса должны быть равны нулю. Значение базового адреса определяется с помощью параметра C_BASEADDR. Формат команды, определяющей значение этого параметра, выглядит следующим образом:

```
PARAMETER C_BASEADDR = <значение_базового_адреса>
```

Значения адреса, как правило, задаются в шестнадцатеричном представлении. Например, командная строка «PARAMETER C_BASEADDR = 0xFFFF0000» выделяет для описываемого экземпляра компонента диапазон адресного пространства, начинающийся с адреса FFFF0000.

Верхняя граница адресного пространства памяти, используемого компонентом, указывается в виде значения параметра C_HIGHADDR. Команда, предназначенная для определения значения этого параметра, имеет следующий формат:

```
PARAMETER C_HIGHADDR = <значение_верхней_границы_выделяемого_адресного_пространства>
```

Примером использования команды, определяющей верхнюю границу адресного пространства памяти для экземпляра некоторого компонента, является строка «PARAMETER C_HIGHADDR = 0xFFFF00FF».

Команда PORT предназначена для описания взаимосвязей между компонентами проектируемой системы, а также внешнего интерфейса системы. Эта команда может быть как глобальной, так и локальной. Глобальная команда PORT используется для описания внешних портов микропроцессорной системы, а также для определения значений постоянных сигналов. Глобальные команды PORT располагаются вне блоков описания компонентов BEGIN-END. Для описания интерфейсных (внешних) портов разрабатываемой микропроцессорной системы наиболее часто используется следующий формат команды PORT:

```
PORT <идентификатор_порта> = <идентификатор_цепи_сигнала>, DIR=<тип_порта_сигнала>, VEC=[n:m], SIGIS=<категория_порта_сигнала>
```

С помощью ключа DIR указывается тип порта (сигнала), соответствующий направлению передачи данных. Для обозначения входного порта используется одно из следующих значений ключа DIR: INPUT, IN, I. В командах, описывающих выходные порты, ключ DIR должен принимать одно из трех значений: OUTPUT, OUT, O. При описании двунаправленного интерфейсного порта указывается одно из двух возможных значений ключа DIR: INOUT или IO. Необязательный ключ VEC позволяет описывать с помощью команды PORT многоразрядные порты, которые представляются в виде векторов (масивов). При использовании ключа VEC значения n и m задают начальный и конечный индексы вектора, которые определяют разрядность порта. Применение команды PORT для описания интерфейсных портов проектируемой системы иллюстрируют следующие примеры:

- строка «PORT Sys_reset = sys_rst, DIR=IN» описывает входной порт Sys_reset, с которым ассоциируется сигнал sys_rst;
- строка «PORT Gp_io = gpio_io, DIR=INOUT, VEC=[0:31]» определяет тридцатидвухразрядный двунаправленный порт Gp_io, с которым ассоциируется сигнал векторного типа gpio_io.

Ключ SIGIS применяется для указания категории, к которой относится сигнал, ассоциируемый с описываемым портом. В текущей версии спецификации MHS предопределены три категории сигналов и соответствующие им значения ключа SIGIS:

- CLK соответствует группе сигналов синхронизации;

- INTERRUPT соответствует группе сигналов прерывания;
- RST соответствует группе сигналов сброса. Например, командная строка «PORT Sys_clock = sys_clk, DIR=IN, SIGIS=CLK» описывает входной порт Sys_clock, с которым ассоциируется тактовый сигнал sys_clk.

Для определения значений постоянных сигналов применяется формат команды PORT, который имеет следующий вид:

```
PORT <идентификатор_постоянного_сигнала> = <значение_сигнала>, DIR=<тип_порта_сигнала>
```

Значение сигнала может задаваться в двоичном или шестнадцатеричном представлении. Например, в командной строке «PORT const1 = 0b1010, DIR=OUTPUT, VEC=[0:3]» значение постоянного сигнала const1 указывается в двоичном формате, а в команде «PORT const2 = 0xC, DIR=OUTPUT, VEC=[0:3]» — в шестнадцатеричном.

С помощью локальных команд PORT описывается подключение портов компонентов микропроцессорной системы. Эти команды не имеют параметров (ключей), определяющих тип и размерность сигналов. Локальные команды PORT располагаются внутри блоков описания компонентов BEGIN-END. Формат локальных команд PORT выглядит следующим образом:

```
PORT <идентификатор_порта_IP-компонента> = <идентификатор_цепи_сигнала>
```

Если какой-либо порт IP-компонента должен оставаться неподключенным, то в соответствующей команде PORT вместо идентификатора цепи указывается пара двойных кавычек («»). Для обозначения цепей питания и земли зарезервированы идентификаторы net_vcc и net_gnd соответственно. Для многоразрядных портов, описываемых с помощью векторов, идентификаторы net_vcc и net_gnd автоматически расширяются в соответствии с требуемой разрядностью. Использование локальной команды PORT демонстрируют следующие примеры:

- Команда «PORT Clk = sys_clk» описывает подключение порта Clk некоторого IP-компонента к цепи sys_clk.
- Команда «PORT = net_gnd» описывает подключение порта In_reset некоторого IP-компонента к цепи «земля».
- Команда «PORT = ""» описывает порт Out_en, который находится в неподключенном состоянии.

Команда BUS_INTERFACE используется для описания взаимосвязей компонентов, осуществляемых на основе шинной архитектуры. Для подключения к шинам различные компоненты микропроцессорной системы используют одни и те же группы цепей (сигналов). При использовании команды PORT для их описания потребовалось бы достаточно большое количество командных строк, повторяющихся для каждого компонента, что привело бы к существенному увеличению времени подготовки и объема файла спецификации аппаратной платформы. Избежать этого позволяет применение для

описания подключения компонентов к шинам проектируемой микропроцессорной системы команды BUS_INTERFACE. Формат команды BUS_INTERFACE выглядит следующим образом:

```
BUS_INTERFACE <тип_шинного_интерфейса> = <идентификатор_шины>
```

Тип шинного интерфейса в команде BUS_INTERFACE указывается в виде соответствующего условного обозначения. Для различных вариантов (режимов) подключения компонентов к стандартным шинным интерфейсам, применяемым во встраиваемых микропроцессорных системах на основе ядер фирмы Xilinx, зарезервированы следующие условные обозначения:

- SDCR соответствует подключению компонента к шине DCR в подчиненном (Slave) режиме;
- SLMB соответствует подключению компонента к шине LMB в подчиненном (Slave) режиме;
- MSOPB соответствует подключению компонента к шине OPB в ведущем-подчиненном (Master-slave) режиме;
- SOPB соответствует подключению компонента к шине OPB в подчиненном (Slave) режиме;
- MPLB соответствует подключению компонента к шине PLB в ведущем (Master) режиме;
- MSPLB соответствует подключению компонента к шине PLB в ведущем-подчиненном (Master-slave) режиме;
- SPLB соответствует подключению компонента к шине PLB в подчиненном (Slave) режиме.

Например, командная строка «BUS_INTERFACE SOPB = *myorb*» описывает подключение компонента в подчиненном режиме к шине *myorb*, которая соответствует стандарту OPB.

В тесте спецификации аппаратной платформы разрабатываемой системы могут располагаться комментарии. Комментарием считается строка, начинающаяся с символа #. Комментарий может занимать отдельную строку или располагаться в командной строке. В последнем случае комментарием считается та часть строки, которая следует после символа #. Применение комментариев повышает информативность файла спецификации аппаратной платформы и облегчает поиск требуемых разделов или компонентов. В качестве примера ниже приведен комментарий, занимающий полную строку, и комментарий, расположенный непосредственно в строке, содержащей команду.

```
# -----Assign constant signals-----
PARAMETER C_HAS_IPR = 1 # Interrupt Pending Register present
```

Типовая структура спецификации аппаратной платформы проектируемой микропроцессорной системы, как правило, содержит следующие разделы:

- команда, указывающая номер версии спецификации MHS;
- описание внешнего интерфейса (глобальных портов) разрабатываемой системы;

- определение портов, подключаемых к цепям питания и «земли»;
- декларация и определение значений постоянных сигналов;
- блоки описания микропроцессорного ядра и компонентов, включаемых в состав разрабатываемой системы.

Создание спецификации аппаратной платформы в среде управляющей оболочки Xilinx Platform Studio

Процесс создания и редактирования спецификации аппаратной платформы проектируемой микропроцессорной системы в среде управляющей оболочки Xilinx Platform Studio может осуществляться в трех режимах: текстовом, графическом и интерактивном. Так как спецификация MHS представляет собой текстовый файл, то для его подготовки и модификации можно использовать текстовый редактор, который входит в состав средств проектирования Xilinx EST. Но создавать новую спецификацию аппаратной платформы с чистого листа в текстовом режиме сложно и трудоемко. При наборе текста команд файла MHS велика вероятность появления различных ошибок. Текстовый режим целесообразно использовать в основном для редактирования значений отдельных параметров или в тех случаях, когда новая спецификация аппаратной платформы формируется на основе уже существующего файла MHS.

Графический метод подготовки спецификации аппаратной платформы заключается в использовании редактора структурных схем Processor Block Diagram Editor (PBD Editor), который является одним из инструментов САПР Xilinx EST. Этот способ отличается большей наглядностью по сравнению с текстовым методом. Проектируемая система представляется в редакторе PBD Editor в виде структурной схемы, элементами которой являются IP-компоненты. Значения всех

необходимых параметров этих компонентов определяются с помощью контекстно-зависимых всплывающих меню.

Для создания с нуля новой спецификации аппаратной платформы удобнее всего воспользоваться интерактивным методом. При использовании этого способа состав проектируемой системы, взаимосвязи компонентов и значения всех параметров определяются в диалоговом режиме с помощью соответствующей многостраничной панели. Чтобы приступить к формированию новой спецификации аппаратной платформы в интерактивном режиме, необходимо выполнить команду Project основного меню управляющей оболочки Xilinx Platform Studio, а затем во всплывающем меню выбрать строку Add/Edit Cores. Можно также воспользоваться аналогичной командой из контекстно-зависимого всплывающего меню, отображаемого на экране при щелчке правой кнопкой мыши на строке *MHS File*, которая находится на странице System встроенного окна проекта разрабатываемой системы, в разделе Project Files. В результате выполненных операций на экран выводится диалоговая панель, вид которой показан на рис. 1.

Панель диалога создания и редактирования спецификации аппаратной платформы проектируемой системы содержит четыре страницы, снабженных закладками с их названиями: *Peripherals*, *Bus Connections*, *Ports*, *Parameters*. Страница *Peripherals* позволяет определить состав компонентов, заносимых в спецификацию проектируемой системы, и указать их общие параметры. Вся информация о компонентах, включенных в спецификацию MHS, представляется в форме таблицы, которая содержит шесть колонок (рис. 1). Для каждого экземпляра компонента, входящего в состав разрабатываемой системы, отводится отдельная строка таблицы. Первоначально при создании новой спецификации в этой таблице представлена только строка с названиями колонок.

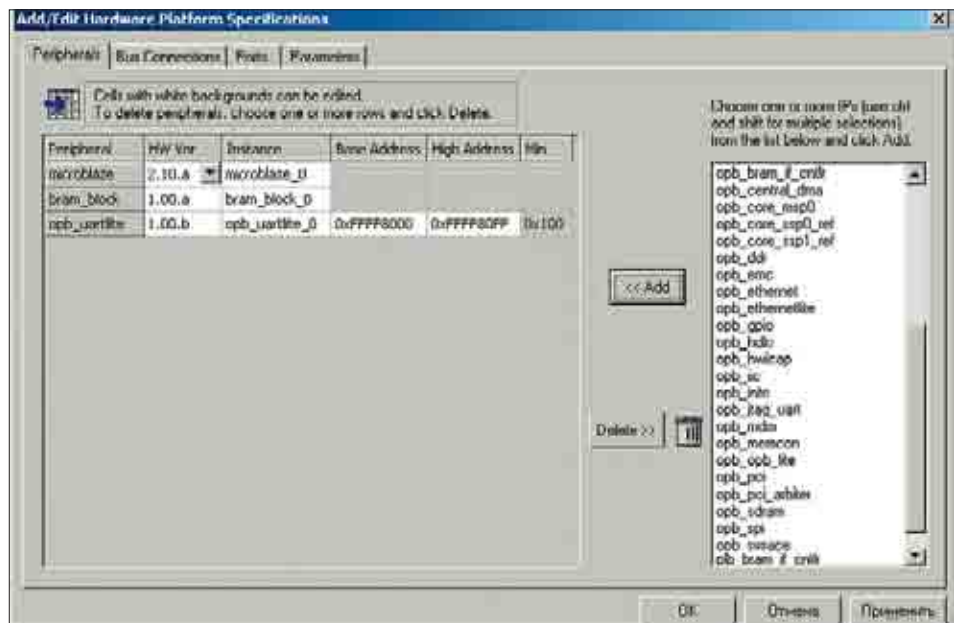


Рис. 1. Диалоговая панель создания и редактирования спецификации аппаратной платформы проектируемой системы (страница Peripherals)

В ячейках, расположенных в колонке с названием *Peripherals*, указывается идентификатор типа компонента, добавленного в состав спецификации. Эти ячейки заполняются автоматически при внесении нового компонента в спецификацию MHS. В колонке *HW Ver* содержится информация о версии аппаратных средств для каждого компонента спецификации. Если какой-либо IP-компонент представлен в нескольких версиях, то соответствующая ячейка этого столбца представляет собой поле выбора, в правой части которого находится кнопка управления выпадающим списком возможных вариантов. При нажатии на эту кнопку на экран выводится список доступных версий данного компонента. Для выбора необходимой версии следует щелкнуть левой кнопкой мыши на соответствующей строке этого списка, после чего номер этой версии будет отображаться в таблице. В ячейках, относящихся к столбцу с названием *Instance*, указывается идентификатор экземпляра компонента, включаемого в состав разрабатываемой микропроцессорной системы. Каждая ячейка, расположенная в этой колонке, представляет собой поле редактирования. Когда новый экземпляр компонента включается в состав спецификации MHS, средства проектирования автоматически присваивают ему некоторый идентификатор. Для изменения имени экземпляра компонента нужно прежде всего активизировать соответствующую ячейку, расположив на ее поле курсор и щелкнув левой кнопкой мыши. Затем, используя клавиатуру, ввести текст идентификатора. В колонке *Base Address* содержатся значения базового адреса компонентов, для которых выделяется адресное пространство. При включении экземпляра такого компонента в состав спецификации для него в соответствующую ячейку таблицы автоматически заносится значение базового адреса, принятое по умолчанию. Каждая ячейка столбца *Base Address* представляет собой поле редактирования соответствующего параметра. Поэтому для изменения значения базового адреса достаточно активизировать соответствующую ячейку и ввести новое значение, используя клавиатуру. В ячейках колонки *High Address* содержатся значения, определяющие верхнюю границу адресного пространства, выделяемого для соответствующего компонента. Изменение значений верхних границ адресного пространства, установленных по умолчанию, осуществляется тем же способом, что и для базовых адресов. При этом следует помнить, что размер диапазона адресного пространства, которое резервируется для некоторого элемента спецификации, должен соответствовать значению, указанному в документации для этого компонента.

В правой части страницы *Peripherals* диалоговой панели создания и редактирования спецификации аппаратной платформы расположено встроенное окно, в котором отображается список доступных IP-компонентов. Чтобы выбрать требуемый компонент, необходимо поместить курсор мыши на соответствующую строку списка и щелкнуть левой

кнопкой. Если нужный тип компонентов не представлен в отображаемой части списка, то следует воспользоваться элементами прокрутки, которые расположены с правой стороны списка (рис. 1). Можно сразу выделить группу компонентов, удерживая нажатой клавишу CTRL или Shift при щелчке левой кнопкой мыши на строках списка. Для того чтобы экземпляры компонентов, выбранных из предложенного списка, были добавлены в спецификацию MHS, нужно нажать кнопку <<Add, которая расположена в диалоговой панели между таблицей и списком (рис. 1). После этого в таблицу будут добавлены строки, соответствующие экземплярам выделенных компонентов. Если какой-либо компонент был ошибочно включен в спецификацию, то для его удаления нужно, прежде всего, выделить соответствующую строку таблицы, расположив на ней курсор и щелкнув левой кнопкой мыши. Затем необходимо нажать кнопку Delete>>, находящуюся несколько ниже кнопки <<Add, в результате чего на экран будет выведена диалоговая панель, в которой содержится запрос подтверждения удаления компонента из спецификации аппаратной платформы (рис. 2). В этой панели также отображается предупреждение о том, что вместе с удаляемым экземпляром компонента будет утеряна вся информация о его параметрах и подключении к шинам.

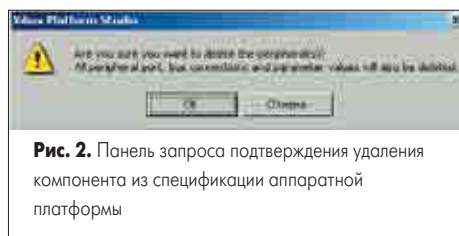


Рис. 2. Панель запроса подтверждения удаления компонента из спецификации аппаратной платформы

При нажатии кнопки ОК, расположенной в нижней части панели запроса (рис. 2), все выделенные строки таблицы будут удалены. Для предотвращения удаления выбранных компонентов следует воспользоваться кнопкой Отмена (Cancel).

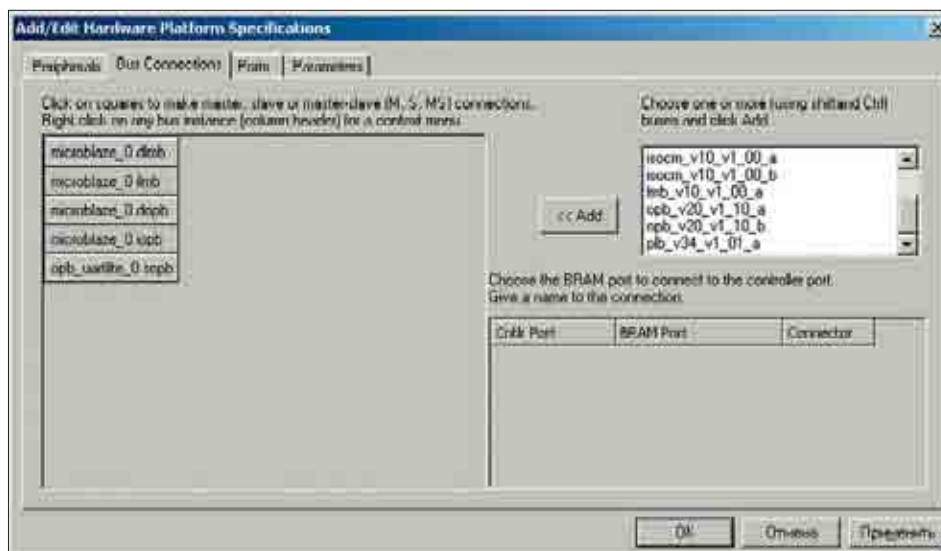


Рис. 3. Страница Bus Connections диалоговой панели создания и редактирования спецификации аппаратной платформы

После того как в состав спецификации аппаратной платформы проектируемой системы включены все необходимые компоненты, следует описать их взаимодействие на основе шинной архитектуры. Для этого нужно перейти к странице *Bus Connections*. Данная страница предназначена для формирования конфигурации шин разрабатываемой системы. Подключение компонентов к шинам проектируемой системы описывается в форме таблицы, которая располагается в левой части диалоговой панели (рис. 3).

В первой колонке этой таблицы представлены все компоненты, которые были включены в состав спецификации MHS с помощью страницы *Peripherals*. Следующие столбцы этой таблицы соответствуют шинам разрабатываемой системы. Первоначально при формировании новой спецификации аппаратной платформы не определен состав шин микропроцессорной системы. Поэтому в таблице присутствует только первая колонка (см. рис. 3). Чтобы добавить описания шин в состав спецификации MHS, нужно воспользоваться списком, который представлен во встроенном окне, расположенном в правой части страницы *Bus Connections* диалоговой панели. В этом списке содержатся идентификаторы всех доступных стандартов шин, которые могут использоваться в архитектуре микропроцессорных систем, проектируемых на основе ядер MicroBlaze и PowerPC. В предложенном списке нужно найти и выделить строку, содержащую требуемый тип шины, поместить на нее курсор мыши и щелкнуть левой кнопкой. При необходимости для просмотра всех вариантов стандартов шин можно использовать инструменты прокрутки, расположенные в правой части списка. Для выделения группы шин нужно удерживать нажатой клавишу Ctrl или Shift при щелчке левой кнопкой мыши на соответствующих строках списка.

Для того чтобы описания шин, соответствующих выбранным из предложенного списка стандартам, были включены в специфика-

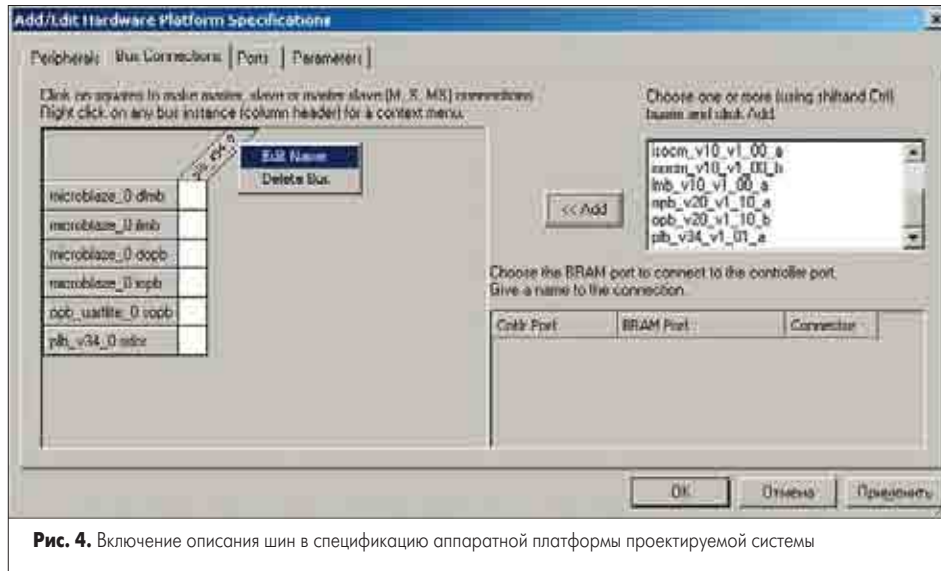


Рис. 4. Включение описания шин в спецификацию аппаратной платформы проектируемой системы

кацию MHS, нужно нажать кнопку <<Add, расположенную на странице *Bus Connections* между таблицей и списком (рис. 3). В результате выполненных действий в таблицу автоматически добавляются столбцы, которые соответствуют типам шин, отмеченным в списке (рис. 4). При этом в верхней части таблицы появляется строка, в которой отображаются идентификаторы шин, установленные по умолчанию.

Для того чтобы изменить названия шин, следует воспользоваться командой *Edit Name*, которая находится в контекстно-зависимом всплывающем меню. Это меню открывается щелчком правой кнопки мыши при расположении курсора в ячейке таблицы, в которой содержится изменяемый идентификатор шины. В результате выполнения команды *Edit Name* на экране отображается диалоговая панель редактирования идентификатора шины, показанная на рис. 5. В поле редактирования этой диалоговой панели приводится текущее название шины. Ввод нового идентификатора шины осуществляется с помощью клавиатуры и завершается нажатием кнопки ОК, которая расположена в нижней части диалоговой панели (рис. 5). Вслед за подтверждением новое название шины автоматически отображается в соответствующей ячейке таблицы.

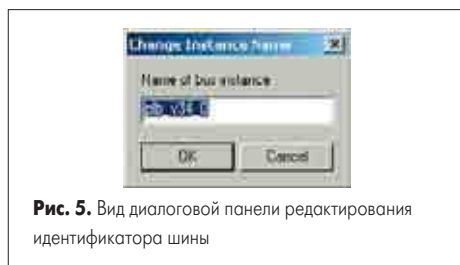


Рис. 5. Вид диалоговой панели редактирования идентификатора шины

Если какая-либо шина была ошибочно включена в состав спецификации MHS, то для ее удаления нужно использовать команду *Delete Bus*, которая располагается в том же контекстно-зависимом всплывающем меню, что и команда *Edit Name*. При выборе команды *Delete Bus* появляется панель запроса подтверждения удаления шины из спецификации аппаратной платформы, вид которой

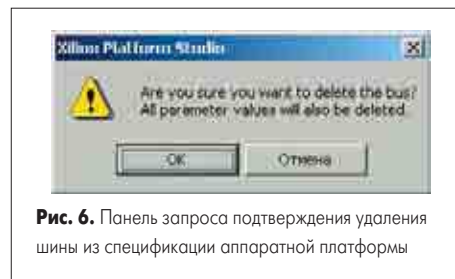


Рис. 6. Панель запроса подтверждения удаления шины из спецификации аппаратной платформы

показан на рис. 6. В этой панели также содержится предупреждение о том, что вместе с удаляемым экземпляром шины будет утеряна вся информация о его параметрах. При утвердительном ответе на вопрос (нажатии кнопки ОК) выбранная шина будет удалена из спецификации MHS.

После того как завершено определение состава и названий шин разрабатываемой системы, необходимо указать способ подключения компонентов к этим шинам. Для этого следует поочередно в каждом столбце сформированной на странице *Bus Connections* таблицы поместить курсор в ячейки, соответствующие подключаемым к этой шине компонентам, и щелкнуть левой кнопкой. Данные ячейки располагаются на пересечении столбца, который соответствует используемой для подключения шине, и строки, содержащей идентификатор подключаемого экземпляра компонента. Большинство компонентов спецификации MHS может быть сопряжено с шиной определенного типа. Этот тип, как правило, указывается в идентификаторах таких компонентов. Например, компонент OPB_UARTLITE представляет собой последовательный асинхронный приемопередатчик, предназначенный для подключения к шине, которая соответствует спецификации OPB (On-chip Peripheral Bus). При отсутствии ошибок в выборе шины и компонента при щелчке левой кнопкой мыши в соответствующей ячейке таблицы отображается одно из трех возможных условных обозначений варианта подключения к шине. Подключение компонента, функционирующего в ведущем (Master) режиме обозначается символом M, в подчиненном (Slave) — символом S, в ведущем-подчиненном (Master-slave) — символами MS.

Сопряжение модулей внутренней блочной памяти ПЛИС Block SelectRAM с контроллером описывается в форме таблицы, которая находится на странице *Bus Connections* под списком стандартов шин. Эта таблица содержит три колонки. В колонке с названием *Ctrlr Port* указывается идентификатор порта контроллера, к которому подключается модуль блочной памяти. В ячейках, расположенных в колонке *BRAM Port*, необходимо указать порт модуля блочной памяти, через который осуществляется сопряжение с контроллером. Каждая из ячеек этого столбца представляет собой поле выбора, в правой части которого находится кнопка управления выпадающим списком. Для определения требуемого порта блочной памяти необходимо нажать эту кнопку и выбрать из предложенного списка тип порта, расположив курсор на строке с его названием и щелкнув левой кнопкой мыши. Колонку *Connector* образуют ячейки, в которых указывается идентификатор, используемый для обозначения подключения блочной памяти. Каждая из этих ячеек является полем редактирования, в котором отображается идентификатор, присваиваемый по умолчанию. Для изменения этого идентификатора необходимо активизировать соответствующую ячейку, расположив на ней курсор и щелкнув левой кнопкой мыши, а затем ввести требуемый текст с помощью клавиатуры.

После завершения описания внутренних соединений элементов разрабатываемой системы с помощью шин можно перейти к следующему шагу — описанию подключения тех портов компонентов, которые не сопряжены с шинами. Эта процедура осуществляется с помощью страницы *Port* диалоговой панели создания и редактирования спецификации аппаратной платформы проектируемой системы.

Окончание следует

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. Кузелин М. О., Кнышев Д. А., Зотов В. Ю. Современные семейства ПЛИС фирмы Xilinx. Справочное пособие. М.: Горячая линия-Телеком. 2004.