

Разработка спецификации программных средств микропроцессорной системы на основе ядра MicroBlaze, реализуемой в ПЛИС семейств FPGA фирмы Xilinx

В предыдущей части данного цикла публикаций [1–7] был подробно представлен процесс проектирования в САПР Xilinx Embedded Development Kit (EDK) аппаратной части встраиваемой микропроцессорной системы на основе 32-разрядного ядра MicroBlaze. Разработка программного обеспечения встраиваемой системы, реализуемой в ПЛИС семейств FPGA фирмы Xilinx [8], начинается с создания спецификации программной платформы. Процесс формирования спецификации программных средств осуществляется с помощью комплекса средств проектирования Embedded System Tools (EST). Но прежде чем перейти непосредственно к рассмотрению процесса подготовки спецификации программной платформы, необходимо познакомиться с синтаксисом соответствующего файла.

Валерий Зотов

walerry@km.ru

Синтаксис и структура файла спецификации программной платформы, реализуемой в ПЛИС семейств FPGA фирмы Xilinx

Спецификация программной платформы встраиваемой микропроцессорной системы Microprocessor Software Specification (MSS) представляет собой файл с расширением MSS. Данный файл имеет текстовый формат и содержит описание конфигурации и параметров программных компонентов проектируемой системы (операционной системы, библиотек и драйверов). Спецификация MSS определяет драйверы для каждого периферийного модуля, стандартные устройства ввода-вывода и программы обработки прерываний. Спецификация программных средств микропроцессорной системы является исходной информацией для генератора библиотек Library Generator (Libgen).

Язык и структура описания программных средств встраиваемой микропроцессорной системы в общих чертах напоминает язык описания спецификации аппаратной платформы MHS (Microprocessor Hardware Specification), который был рассмотрен в предыдущих публикациях [6].

Каждый компонент, включаемый в состав MSS, описывается с помощью конструкции, которая начинается с ключевого слова BEGIN и заканчивается ключевым словом END. Формат этой конструкции выглядит следующим образом.

```
BEGIN <тип_компонента_спецификации_MSS>
parameter <параметр1> = <значение_параметра1> [, <параметр1_1> = <значение_параметра1_1>]
parameter <параметр2> = <значение_параметра2> [, <параметр2_1> = <значение_параметра2_1>]
...
parameter <параметрN> = <значение_параметраN> [, <параметр2_N> = <значение_параметра2_N>]
END
```

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

Блок описания каждого компонента в спецификации программной платформы разрабатываемой микропроцессорной системы открывает строка, содержащая ключевое слово BEGIN. В этой же строке указывается тип описываемого компонента, который соответствует одному из видов программных средств или процессора. Тип компонента спецификации MSS определяется с помощью соответствующего ключевого слова. В рассматриваемой версии спецификации программных средств для этого зарезервированы четыре варианта ключевого слова: OS (Operating System), driver, library, processor. Ключевое слово OS используется в том случае, если блок описывает параметры операционной системы. Например, следующий блок определяет такие параметры, как идентификатор микропроцессорного ядра, тип и версию используемой операционной системы, идентификаторы стандартных устройств ввода-вывода.

```
BEGIN OS
parameter PROC_INSTANCE = microblaze_core
parameter OS_NAME = standalone
parameter OS_VER = 1.00.a
parameter STDIN = uartlite_core
parameter STDOUT = uartlite_core
END
```

При описании драйвера некоторого IP-компонента проектируемой системы указывается ключевое слово driver. В качестве примера ниже приводится блок описания параметров драйвера для параллельного порта ввода-вывода, подключенного к шине OPB.

```
BEGIN DRIVER
PARAMETER HW_INSTANCE = opb_gpio_core
PARAMETER DRIVER_NAME = gpio
PARAMETER DRIVER_VER = 1.00.a
PARAMETER LEVEL = 1
END
```

Если описываемый компонент спецификации MSS представляет собой библиотеку, то используется ключевое слово library. Например, в следующем блоке указываются параметры используемой библиотеки XilMfs.

```
BEGIN LIBRARY
parameter LIBRARY_NAME = XilMfs
parameter LIBRARY_VER = 1.00.a
parameter NUMBYTES = 100000
parameter BASE_ADDRESS = 0x80f00000
END
```

Для описания параметров микропроцессорного ядра в заголовке блока указывается ключевое слово `processor`. В приведенном ниже блоке определяются следующие параметры процессора: обозначение экземпляра компонента микропроцессорного ядра, идентификатор и версия соответствующего драйвера.

```
BEGIN PROCESSOR
parameter HW_INSTANCE = microblaze_core
parameter DRIVER_NAME = cpu
parameter DRIVER_VER = 1.00.a
parameter XMDSTUB_PERIPHERAL = my_jtag
END
```

В строках, которые следуют после ключевого слова `BEGIN`, располагается совокупность команд `parameter`, с помощью которых указываются параметры программного компонента, включаемого в состав спецификации MSS. Общее число команд `N` в этой конструкции обуславливается, в первую очередь, типом описываемого программного компонента. Количество параметров, определяемых в каждой команде `parameter`, зависит от типа этой команды. В большинстве случаев в каждой строке, содержащей команду `parameter`, определяется значение одного параметра, но можно использовать командные строки, в которых задаются значения двух параметров. Например в команде, описывающей программу обработки прерывания, указывается не только ее название, но и идентификатор соответствующего порта (сигнала) прерывания. Описание каждого компонента спецификации программной платформы проектируемой системы завершается ключевым словом `END`.

Команда `parameter` является основной и наиболее часто используемой в спецификации программных средств, поэтому рассмотрим ее подробнее. Прежде всего следует обратить внимание на то, что все параметры, определяемые командой `parameter`, можно разделить на две группы: глобальные и локальные. Глобальные параметры относятся ко всей спецификации программной платформы в целом. Команды `parameter`, которые определяют значения глобальных параметров, располагаются вне блоков описания программных компонентов `BEGIN-END`. К группе глобальных относятся параметры описания версии спецификации MSS и программ обработки прерываний для глобальных портов системы. Значение глобального параметра `VERSION` указывает номер версии формата спецификаций Platform Specification Format (PSF) встраиваемой микропроцессорной системы. Значение этого параметра обычно задается в начале файла MSS. Формат команды, определяющей номер версии спецификации программной платформы, выглядит следующим образом:

```
parameter VERSION = <номер_версии_MSS>
```

Например, в командной строке «`parameter VERSION = 2.2.0.`» указан номер версии фор-

мата спецификаций PSF, используемый в САПР Xilinx EDK 6.2.

Глобальный параметр `INT_HANDLER` используется для описания программы обслуживания прерывания, ассоциируемого с соответствующим внешним портом проектируемой системы. Формат выражения, в котором определяется значение этого параметра, имеет следующий вид:

```
parameter INT_HANDLER = <идентификатор_программы_обработки_прерывания>, INT_PORT = <идентификатор_внешнего_сигнала(порта)_прерывания>
```

Например, командная строка

```
parameter INT_HANDLER = int_handl_proc, INT_PORT = Sys_Interrupt
```

устанавливает в качестве процедуры обслуживания прерывания, ассоциируемого с внешним портом `Sys_Interrupt`, программу `int_handl_proc`.

Локальные параметры спецификации MSS относятся только к соответствующим компонентам программной платформы проектируемой системы. Команды, определяющие значения локальных параметров, располагаются внутри блоков описания программных компонентов `BEGIN — END`. Рассмотрим подробнее наиболее часто используемые локальные параметры.

С помощью параметра `HW_INSTANCE` указывается позиционное обозначение (идентификатор) экземпляра IP-компонента, с которым ассоциируется описываемый программный компонент. Данный параметр используется при описании драйверов (`DRIVER`) и процессора (`PROCESSOR`). Для определения значения этого параметра следует включить в состав соответствующего блока описания `BEGIN — END` командную строку, формат которой имеет следующий вид:

```
parameter HW_INSTANCE = <идентификатор_экземпляра_IP-компонента>
```

В этом выражении в качестве значения параметра `HW_INSTANCE` должен указываться тот же идентификатор, что задан для соответствующего экземпляра IP-компонента в спецификации аппаратной части проектируемой системы. Например, для описания драйвера последовательного асинхронного приемопередатчика, который обозначен в спецификации MHS как `uart_port`, в соответствующий блок `BEGIN DRIVER — END` спецификации программных средств нужно включить командную строку

```
parameter HW_INSTANCE = uart_port.
```

Значение параметра `PROC_INSTANCE` указывает позиционное обозначение (идентификатор) экземпляра микропроцессорного ядра, с которым ассоциируется описываемая операционная система. Указываемый идентификатор должен совпадать с позиционным обозначением соответствующего экземпляра микропроцессорного ядра в спецификации MHS. Формат выражения, используемого в блоке описания `BEGIN OS — END` для определения этого значения, выглядит следующим образом:

```
parameter PROC_INSTANCE = <идентификатор_экземпляра_микропроцессорного_ядра>
```

Параметр `OS_NAME` предназначен для указания используемой операционной системы. Для определения значения этого параметра необходимо в блок описания операционной системы `BEGIN OS — END` добавить выражение

```
parameter OS_NAME = <обозначение_используемой_операционной_системы>
```

Если разрабатываемое программное обеспечение встраиваемой микропроцессорной системы не предусматривает использование какой-либо операционной системы, то для параметра `OS_NAME` следует указать значение `standalone`.

Номер версии применяемой операционной системы задается в форме значения параметра `OS_VER`. Для определения этого значения в блоке описания `BEGIN OS — END` должна присутствовать командная строка, формат которой имеет следующий вид:

```
parameter OS_VER = <номер_версии_операционной_системы>
```

Номер версии операционной системы указывается в формате `k.m.n.a`, где `k`, `m`, `n` являются цифровыми символами, а `a` — алфавитным символом.

Параметр `DRIVER_NAME` используется для назначения каждому компоненту спецификации аппаратной части проектируемой системы соответствующего драйвера. Формат команды, определяющей значение этого параметра, выглядит следующим образом:

```
parameter DRIVER_NAME = <обозначение_драйвера_устройства>
```

Так как драйверы некоторых устройств имеют несколько версий, необходимо уточнить номер используемой версии драйвера, используя параметр `DRIVER_VER`. Выражение, используемое для определения значения параметра `DRIVER_VER`, имеет формат

```
parameter DRIVER_VER = <номер_версии_драйвера_устройства>
```

Номер версии драйвера задается в том же формате, что и номер версии операционной системы.

Локальный параметр `INT_HANDLER` используется для описания программы обслуживания прерывания, ассоциируемого с соответствующим портом периферийного устройства. Формат командной строки, в которой устанавливается значение этого параметра, имеет такой вид:

```
parameter INT_HANDLER = <идентификатор_программы_обработки_прерывания>, INT_PORT = <идентификатор_сигнала(порта)_прерывания_периферийного_устройства>
```

В этой командной строке должен быть указан тот же идентификатор порта прерывания периферийного устройства, что и в спецификации аппаратной части проектируемой системы для этого компонента.

Для описания используемой библиотеки необходимо указать ее название и номер версии.

Название применяемой библиотеки задается с помощью параметра `LIBRARY_NAME`. В блок описания используемой библиотеки `BEGIN LIBRARY — END` должна быть включена командная строка, определяющая значение этого параметра:

```
parameter LIBRARY_NAME = <название_используемой_библиотеки>
```

Номер версии применяемой библиотеки определяется с помощью параметра `LIBRARY_VER`. Значение этого параметра указывается в том же формате, что и номер версии операционной системы. Для этого используется выражение

```
parameter LIBRARY_VER = <номер_версии_библиотеки>
```

Для идентификации стандартных устройств ввода и вывода в блоке описания опций операционной системы используются параметры `STDIN` и `STDOUT` соответственно. В качестве значений этих параметров указываются идентификаторы периферийных IP-компонентов, выполняющих роль стандартных устройств ввода и вывода. Формат командной строки для определения параметров `STDIN` и `STDOUT` выглядит следующим образом.

```
parameter STDIN = <идентификатор_устройства_ввода>
parameter STDOUT = <идентификатор_устройства_вывода>
```

Например, строки

```
parameter STDIN = uart_RS232
parameter STDOUT = uart_RS232
```

определяют в качестве стандартных устройств ввода и вывода последовательный асинхронный приемопередатчик `uart_RS232`.

В блоке описания микропроцессорного ядра используется ряд специфических параметров. Параметр `XMDSTUB_PERIPHERAL` позволяет задать идентификатор периферийного устройства, используемого для осуществления процесса отладки. Формат выражения, определяющего значение этого параметра, имеет следующий вид:

```
parameter XMDSTUB_PERIPHERAL = <идентификатор_устройства_отладки>
```

Параметр `COMPILER` предназначен для выбора компилятора, который должен применяться для компиляции используемых библиотек и драйверов. Значение этого параметра должно соответствовать типу микропроцессорного ядра. При использовании стандартного компилятора для ядра `MicroBlaze` необходимо в качестве значения параметра `COMPILER` указать `mb-gcc`. Формат соответствующей командной строки выглядит таким образом:

```
parameter COMPILER = <тип_компилятора>
```

С помощью параметра `ARCHIVER` выбирается программа архивации объектных файлов в библиотеки. Выбор архиватора зависит от типа используемого микропроцессорного ядра. Чтобы указать стандартный архиватор для ядра `MicroBlaze`, в качестве значения параметра `ARCHIVER` следует задать `mb-ag`, используя командную строку следующего формата:

```
parameter ARCHIVER = <название_программы_архивации>
```

В блоке описания процессора могут также применяться параметры `COMPILER_FLAGS` и `EXTRA_COMPILER_FLAGS`, которые задают опции компиляции драйверов и библиотек. При использовании стандартного компилятора и архиватора данные параметры не рекомендуется указывать.

Значение параметра `CORE_CLOCK_FREQ_HZ` определяет тактовую частоту микропроцессорного ядра. Формат выражения, в котором указывается это значение, представлен ниже:

```
parameter CORE_CLOCK_FREQ_HZ = <значение_тактовой_частоты_микропроцессорного_ядра>
```

Значение тактовой частоты задается в герцах.

В тесте спецификации программных средств, также как и в спецификации аппаратной платформы, могут использоваться комментарии. Применение комментариев делает структуру файла `MSS` более прозрачной и облегчает поиск параметров, значение которых необходимо скорректировать. Комментарием считается часть любой строки, которая следует после символа `#`. Комментарий может занимать полную строку, если в начале ее указан символ `#`. Примеры комментариев различного вида в файле `MSS` демонстрируют следующие строки:

```
# ----- Assign drivers -----
parameter DRIVER_NAME = uarlite # DRIVER for RS-232
```

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

- команда, указывающая номер версии спецификации `MSS`;
- описание используемой операционной системы;
- определение параметров микропроцессорного ядра;
- блоки описания драйверов периферийных устройств;
- блоки описания используемых библиотек.

Создание спецификации программных средств в среде управляющей оболочки **Xilinx Platform Studio**

Процесс создания и редактирования спецификации программной платформы проектируемой микропроцессорной системы в среде управляющей оболочки `Xilinx Platform Studio` может осуществляться в двух режимах: текстовом и интерактивном. Так как файл спецификации `MSS` имеет текстовый формат, то для его подготовки и внесения изменений можно использовать любой текстовый редактор, например, который входит в состав средств проектирования `Xilinx EST`. Однако процедура создания абсолютной новой спецификации программной платформы в текстовом редакторе занимает достаточно много времени и не исключает появления различных ошибок. Текстовый режим целесообразнее использовать, в основном, для редактирования значений отдельных параметров или в тех случаях, когда новая спецификация программных средств

разрабатываемой микропроцессорной системы формируется на основе уже существующего файла `MSS`.

Более эффективным способом создания «с нуля» новой спецификации программной платформы является интерактивный метод. При использовании этого метода описание программных компонентов проектируемой системы и определение значений всех параметров выполняются в табличной форме с помощью соответствующей многостраничной диалоговой 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 *Software Platform Settings*. Можно также воспользоваться аналогичной командой из контекстно-зависимого всплывающего меню, отображаемого на экране при щелчке правой кнопкой мыши на строке, в которой указан идентификатор микропроцессорного ядра, находящейся на странице *System* встроенного окна проекта разрабатываемой системы, в разделе *System BSP*. Вид диалоговой панели представлен на рис. 1.

Диалоговое окно создания и редактирования спецификации программных средств содержит три вкладки: *Software Platform*, *Processor and Driver Parameters*, *Library/OS Parameters*.

Страница *Software Platform* позволяет указать состав компонентов, заносимых в спецификацию программной платформы проектируемой системы (драйверов, библиотек, операционной системы), и определить значения их общих параметров. Все сведения о компонентах, включенных в спецификацию `MSS`, заносятся в соответствующие таблицы, расположенные на этой странице (рис. 1). Для каждого типа программных средств, используемых в разрабатываемой системе, предусмотрена своя таблица, где каждому компоненту отводится отдельная строка.

В таблицу *Drivers* заносится информация о драйверах всех периферийных устройств, которые были включены в состав спецификации аппаратной платформы разрабатываемой системы. Эта таблица содержит пять колонок. В ячейках, расположенных в колонке с названием *Peripheral*, указываются идентификаторы типа компонентов, которые перечислены в спецификации `MHS`. В колонке *HW Ver* содержится информация о версии аппаратных средств для каждого компонента, указанного в предыдущей колонке. В ячейки, относящиеся к столбцу с названием *Instance*, вносится идентификатор экземпляра компонента, включенного в состав спецификации `MHS` разрабатываемой микропроцессорной системы. В ячейках, которые относятся к столбцу *Driver*, указывается драйвер для соответствующего экземпляра компонента. Каждая 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 драйверов для соответствующего типа периферийного устройства. Для выбора необходимого вари-

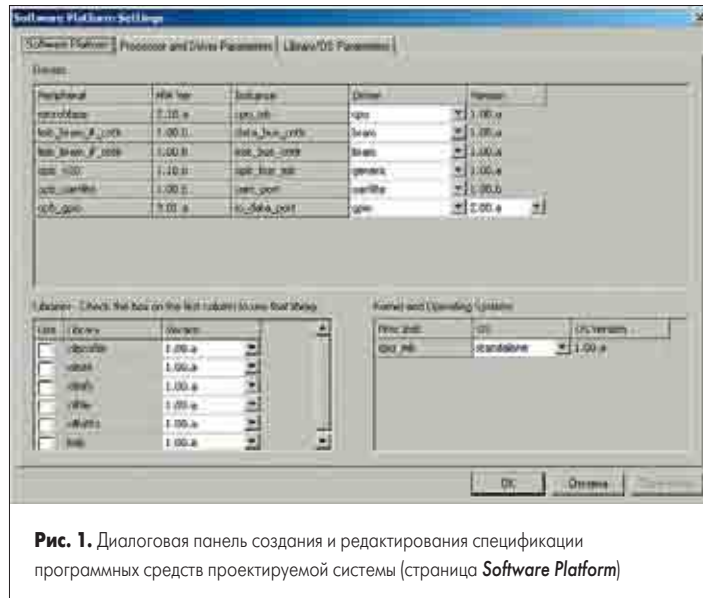


Рис. 1. Диалоговая панель создания и редактирования спецификации программных средств проектируемой системы (страница **Software Platform**)

анта следует щелкнуть левой кнопкой мыши на соответствующей строке этого списка, после чего название выбранного драйвера будет отображаться в таблице. Колонка *Version* содержит сведения об используемых версиях драйверов. Если какой-либо драйвер 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 существующих версий данного драйвера. Для выбора необходимой версии следует щелкнуть левой кнопкой мыши по соответствующей строке списка, после чего номер этой версии будет отображаться в текущей ячейке таблицы. Первоначально, при создании новой спецификации MSS, в таблице *Drivers* представлены только исходные сведения в первых трех колонках, которые заполняются автоматически на основании информации, содержащейся в спецификации аппаратной части микропроцессорной системы.

Таблица *Libraries* позволяет определить состав и версии применяемых библиотек. Каждая строка этой таблицы содержит основную информацию о соответствующей библиотеке. Данная таблица включает в себя три колонки. В ячейках, которые составляют столбец *Library*, перечисляются названия всех доступных библиотек. В колонке *Use* указывается информация об использовании соответствующей библиотеки. Каждая ячейка этого столбца представляет собой поле индикатора состояния. Для того чтобы добавить описание библиотеки в спецификацию программных средств, нужно перевести индикатор в состояние «включено» (отмеченное маркером), расположив на его поле курсор и щелкнув левой кнопкой мыши. Ячейки столбца *Version* позволяют выбрать версию используемой библиотеки. Выбор версии библиотеки осуществляется тем же способом, что и версии драйвера.

В таблице *Kernel and Operating Systems* указывается операционная система для каждого экземпляра микропроцессорного ядра. В первой колонке *Proc Inst* этой таблицы отображается идентификатор экземпляра микропроцессорного ядра. В ячейках второй колонки *OS* необходимо выбрать требуемую операционную систему, воспользовавшись кнопкой активизации выпадающего списка. В третьей колонке (*OS Version*) содержатся сведения о номере версии выбранной операционной системы. Если используемая операционная система имеет несколько версий, то соответствующая ячейка этого столбца представляет собой поле выбора, в правой части которого находится кнопка управления выпадающим списком возможных вариантов. Определив состав компонентов спецификации MSS и значения их общих параметров, нужно перейти к следующей странице диалоговой панели создания и редактирования спецификации программных средств проектируемой системы.

Страница *Processor and Driver Parameters* используется для определения значений параметров процессора и драйверов периферийных устройств. Вид этой страницы показан на рис. 2.

На этой странице представлены две таблицы, имеющие аналогичную структуру. Значения основных параметров микропроцессорного ядра заносятся в таблицу *Processor Parameters*. В колонке *Instance* этой

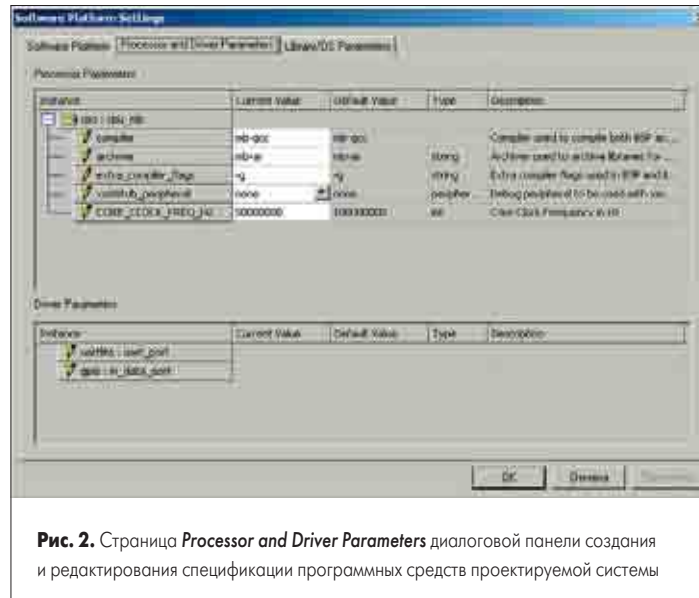


Рис. 2. Страница **Processor and Driver Parameters** диалоговой панели создания и редактирования спецификации программных средств проектируемой системы

таблицы отображается идентификатор экземпляра микропроцессорного ядра, под которым перечисляются его параметры. В ячейках столбца *Current Value* указываются текущие значения соответствующих параметров микропроцессорного ядра. Ячейки, входящие в состав колонки *Default Value*, содержат значения параметров, установленные по умолчанию. В столбце *Type* отображается тип значений соответствующих параметров. В ячейках колонки *Description*, приводятся краткие комментарии, поясняющие назначение параметров.

Таблица *Driver Parameters* предназначена для определения значений параметров драйверов периферийных устройств, которые были включены в состав спецификации аппаратной платформы. В первой колонке таблицы перечисляются идентификаторы периферийных устройств и параметры их драйверов (если таковые имеются). Остальные колонки этой таблицы выполняют те же функции, что и в таблице *Processor Parameters*.

После определения всех параметров процессора и драйверов периферийных устройств нужно перейти к странице с названием *Library/OS Parameters*, вид которой представлен на рис. 3.

Данная страница используется для установки требуемых значений параметров библиотек и операционной системы. Определение значений этих параметров осуществляется в табличной форме. Таблица *Library and OS Parameters* имеет ту же структуру, что и таблицы *Processor Parameters* и *Driver Parameters*. В первой колонке таблицы *Library and OS Parameters* отображается идентификатор микропроцессорного ядра и соответствующая операционная система. Ниже перечисляются параметры операционной системы, значения которых указываются в ячейках столбца *Current Value*. Для выбора требуемых значений следует использовать кнопку, расположенную в правой части этих ячеек, которая открывает доступ к выпадающему списку возможных вариантов.

После того, как указаны все программные компоненты и определены значения их параметров, следует подтвердить выполненные изменения в диалоговой панели создания и редактирования спецификации программной платформы нажатием кнопки *Применить (Apply)* или *ОК*, которые находятся в нижней части этой панели (рис. 1–3). При этом

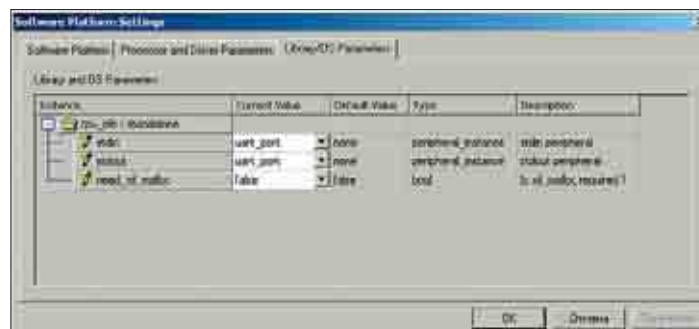


Рис. 3. Страница **Library/OS Parameters** диалоговой панели создания и редактирования спецификации программных средств проектируемой системы

вся информация, заданная в табличной форме, записывается в виде соответствующих команд в файл спецификации MSS. Использование кнопки *ОК* приводит к закрытию диалоговой панели.

В качестве примера создадим спецификацию программных средств для проекта простейшей микропроцессорной системы сбора и обработки данных, разработка аппаратной части которой рассмотрена в предыдущих публикациях [5–7]. Для этого нужно заполнить все указанные выше таблицы, как показано на рис. 1–3. После закрытия диалоговой панели создания и редактирования спецификации программной платформы рекомендуется проверить содержимое файла `system_tst.mss`. Чтобы открыть этот файл, нужно во встроенном окне проекта разрабатываемой системы, на странице *System*, в разделе *Project Files* расположить курсор на строке *MSS File* и дважды щелкнуть левой кнопкой мыши. Содержимое файла `system_tst.mss` для микропроцессорной системы сбора и обработки данных должно выглядеть следующим образом:

```
PARAMETER VERSION = 2.2.0

BEGIN OS
PARAMETER OS_NAME = standalone
PARAMETER OS_VER = 1.00.a
PARAMETER PROC_INSTANCE = cpu_mb
PARAMETER STDIN = uart_port
PARAMETER STDOUT = uart_port
END

BEGIN PROCESSOR
PARAMETER DRIVER_NAME = cpu
PARAMETER DRIVER_VER = 1.00.a
PARAMETER HW_INSTANCE = cpu_mb
```

```
PARAMETER COMPILER = mb-gcc
PARAMETER ARCHIVER = mb-ar
PARAMETER CORE_CLOCK_FREQ_HZ = 50000000
END

BEGIN DRIVER
PARAMETER DRIVER_NAME = bram
PARAMETER DRIVER_VER = 1.00.a
PARAMETER HW_INSTANCE = data_bus_cntlr
END

BEGIN DRIVER
PARAMETER DRIVER_NAME = bram
PARAMETER DRIVER_VER = 1.00.a
PARAMETER HW_INSTANCE = inst_bus_cntlr
END

BEGIN DRIVER
PARAMETER DRIVER_NAME = generic
PARAMETER DRIVER_VER = 1.00.a
PARAMETER HW_INSTANCE = opb_bus_mb
END

BEGIN DRIVER
PARAMETER DRIVER_NAME = uartlite
PARAMETER DRIVER_VER = 1.00.b
PARAMETER HW_INSTANCE = uart_port
END

BEGIN DRIVER
PARAMETER DRIVER_NAME = gpio
PARAMETER DRIVER_VER = 2.00.a
PARAMETER HW_INSTANCE = in_data_port
END
```

На основании сформированной спецификации программных средств далее выполняется генерация соответствующих драйверов и библиотек. Эта процедура, а также процесс разработки и компиляции прикладных программ для встраиваемой микропроцессорной системы, реализуемой в ПЛИС семейств FPGA фирмы Xilinx с использованием ядра MicroBlaze, будут рассмотрены в следующей публикации. ■

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. Реализация аппаратной платформы микропроцессорной системы, проектируемой на основе ядра MicroBlaze, в ПЛИС семейств FPGA фирмы Xilinx // Компоненты и технологии. 2004. № 9.
8. Кузелин М. О., Кнышев Д. А., Зотов В. Ю. Современные семейства ПЛИС фирмы Xilinx: Справочное пособие. М.: Горячая линия-Телеком. 2004.