MicroBlaze — семейство тридцатидвухразрядных микропроцессорных ядер,

реализуемых на основе ПЛИС фирмы Xilinx

Семейство восьмиразрядных микропроцессорных ядер PicoBlaze, которое было представлено в первой части цикла публикаций [1–5], посвященного рассмотрению встраиваемых микропроцессорных модулей на основе ПЛИС фирмы Xilinx, применяется, как правило, в системах начального и среднего уровня.
В качестве основы для проектирования «систем на кристалле» более высокого уровня и быстродействия фирма Xilinx предлагает семейство тридцатидвухразрядных микропроцессорных модулей MicroBlaze.

Валерий Зотов

walerry@euro.ru

В отличие от свободно распространяемого семейства микропроцессорных ядер PicoBlaze, MicroBlaze является составной частью пакета Embedded Development Kit (EDK), предлагаемого фирмой Xilinx в качестве основного инструмента разработки и отладки встраиваемых микропроцессорных систем на основе ПЛИС серий FPGA. Элементы семейства MicroBlaze представляют собой встраиваемые микропроцессорные ядра с RISC-архитектурой, которые предназначены для применения в системах, выполняемых соответственно на основе ПЛИС серий Spartan-II, Spartan-IIE, Spartan-3, Virtex, Virtex-E, Virtex-II, Virtex-II Pro.

В настоящей статье рассматриваются основные характеристики микропроцессорного ядра MicroBlaze, его архитектура и организация шинных интерфейсов.

Основные характеристики микропроцессорного ядра MicroBlaze

Особенностями микропроцессорного ядра MicroBlaze являются:

- гибкая архитектура с раздельными шинами данных и команд, которые соответствуют спецификации OPB (On-chip Peripheral Bus) фирмы IBM;
- 32-разрядная шина данных;
- 32-разрядная шина команд;
- прямой доступ к ресурсам блочной памяти ПЛИС Block SelectRAM, реализуемый через шину LMB (Local Memory Bus);
- возможность организации памяти данных микропроцессорной системы на основе ресурсов внутренней блочной памяти ПЛИС Block SelectRAM или внешних запоминающих устройств различного типа;
- диапазон адресного пространства оперативной памяти ядра позволяет использовать в составе проектируемой микропроцессорной системы память данных объемом от 0 до 4 Гбайт;
- возможность использования кэш-памяти с целью повышения производительности разрабатываемой микропроцессорной системы;
- 32-разрядное арифметическо-логическое устройство (АЛУ), реализующее логические функции, операции сложения, вычитания, умножения, деления и сдвига;
- блок регистров общего назначения, включающий 32 32-разрядных регистра;
- 32-разрядная шина адресов;
- система команд, поддерживающая 106 32-разрядных инструкций;
- возможность реализации ППЗУ микропрограмм как на основе внутренней блочной памяти ПЛИС, так и на базе внешней памяти, расположенной вне кристалла;
- диапазон адресного пространства программной памяти ядра позволяет использовать в составе разрабатываемой микропроцессорной системы ППЗУ микропрограмм объемом от 512 кбайт до 4 Гбайт;
- применение трехступенчатого конвейера микрокоманд, обеспечивающее высокую производительность проектируемой системы;
- поддержка прямого, косвенного и непосредственного режимов адресации;
- поддержка 8 входных и 8 выходных интерфейсов, соответствующих спецификации FSL (Fast Simplex Link);
- применение аппаратных умножителей для реализации соответствующих операций АЛУ (в кристаллах, обладающих данным типом ресурсов,

например, в ПЛИС семейств Virtex-II или Spartan-3);

- включение в состав микропроцессорного ядра специальной логики, обеспечивающей возможность аппаратной отладки разрабатываемых программ непосредственно в кристалле через порт JTAG-интерфейса;
- наличие входа сброса (инициализации), позволяющего перевести микропроцессор в начальное состояние;
- полная совместимость компонентов ядра со средствами разработки проектов и программирования ПЛИС ISE (Integrated Synthesis Environment) фирмы Xilinx: Base ISE, Foundation ISE и Alliance ISE;
- возможность моделирования ядра в составе разрабатываемых проектов с помощью системы ModelSim XE [6–9];
- оптимизированная структура микропроцессорного ядра, учитывающая особенности используемых ПЛИС, позволяет минимизировать объем ресурсов кристалла, требуемых для его реализации, что делает возможным размещение в этом же кристалле периферийных модулей проектируемой системы, включая интерфейсы ввода-вывода (в ПЛИС семейства Spartan-3 ядро MicroBlaze занимает 525 секций (slices) и два модуля блочной памяти, что составляет 68% от полного объема логических ресурсов кристалла XC3S50, 27% — кристалла XC3S200 и 15% от логической емкости ПЛИС XC3S400 [10]);
- расширенные интерфейсные возможности микропроцессорного ядра, обеспечивающие оптимальное его сопряжение с периферийными модулями, реализуемыми как на основе свободных логических ресурсов кристалла, так и в виде внешних устройств;
- наличие дополнительных компонентов периферийных модулей микропроцессорной системы, включенных в комплект пакета программных средств разработки EDK в виде IP-ядер, которые позволяют значительно ускорить и упростить процесс разработки «систем на кристалле»;
- высокая производительность, достигающая 125 D-MIPS (в зависимости от типа используемого кристалла);
- интегрированная система разработки и отладки аппаратной части и программных средств встраиваемых микропроцессорных систем, реализуемых на основе кристаллов семейств FPGA фирмы Xilinx, EDK;
- возможность разработки микропроцессорных программ с применением языков высокого уровня C/C++.

Архитектура микропроцессорного ядра MicroBlaze

Структура микропроцессорного ядра MicroBlaze построена по принципу Гарвардской архитектуры, в основе которой лежит концепция использования раздельных шин данных и команд. Организация магистралей микропроцессора в соответствии с этой концепцией обеспечивает достижение высоких скоростей выполнения операций. Обобщенная структура микропроцессорного ядра MicroBlaze представлена на рис. 1.

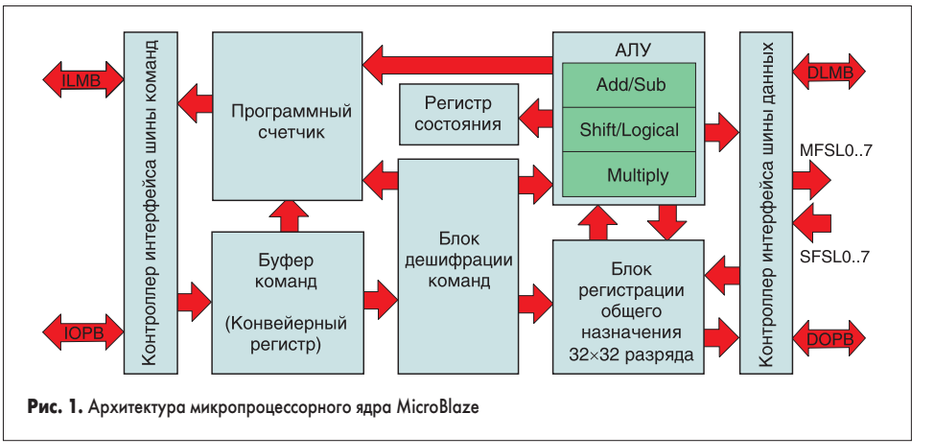

Рис. 1. Архитектура микропроцессорного ядра MicroBlaze

Основными элементами архитектуры микропроцессорного ядра MicroBlaze являются:

- 32-разрядное АЛУ;
- блок регистров общего назначения;
- регистр состояния (статуса);
- программный счетчик (счетчик команд);
- буфер команд (конвейерный регистр);
- блок декодирования команд;
- контроллер интерфейса шины команд;
- контроллер интерфейса шины данных;
- отладочная логика;
- кэш-память команд;
- кэш-память данных;
- схема управления прерываниями.

Последние четыре элемента из перечисленных выше явно не показаны на рис. 1. Отладочная логика и кэш-память команд и данных не являются обязательными элементами архитектуры микропроцессорного ядра MicroBlaze. Необходимость их присутствия определяется разработчиком в соответствии с конфигурацией проектируемой системы.

Блок АЛУ выполняет логические и арифметические операции над 32-разрядными операндами. В качестве операндов может использоваться содержимое любого из 32 регистров общего назначения, а также 16- или 32-разрядные константы, указанные непосредственно в тексте команды. Результат выполнения операции заносится в один из регистров общего назначения, номер которого задается в качестве соответствующего параметра выполняемой инструкции. В кристаллах семейств Virtex-II и Spartan-3 для выполнения инструкций умножения используются блоки аппаратных умножителей, что позволяет значительно сократить время выполнения указанных операций.

Регистр статуса (Machine Status Register, MSR) содержит значения флагов (признаков), формируемых блоком АЛУ при выполнении арифметических и логических инструкций, а также операций ввода-вывода и прерываний. Этот регистр содержит 32 разряда, назначение которых подробно рассматривается в следующем разделе.

Блок регистров общего назначения содержит 32 регистра, обозначаемых по порядку R0 — R31, длина каждого из которых составляет 32 разряда. При использовании этих регистров в разрабатываемой программе следует придерживаться соглашений, установленных интерфейсом Application Binary Interface (ABI), который определяет правила создания приложений на языке ассемблера для микропроцессорного ядра MicroBlaze. Компилятор программ, написанных на языке высокого уровня C/C++, при их трансляции автоматически выполняет требования, предусмотренные ABI. Согласно классификации, принятой в ABI, все регистры общего назначения условно разделяются на три группы:

- регистры, предназначенные для хранения временных (промежуточных) данных, которые не сохраняются при вызове функций (подпрограмм);
- регистры, содержимое которых восстанавливается после вызова функций (подпрограмм) в состояние, предшествующее обращению к функции (подпрограмме);
- выделенные регистры, предназначенные для хранения строго определенных данных.

К первой группе относятся регистры с номерами R3 — R12. Регистры R3 и R4 предназначены для записи возвращаемых значений вызываемой функции, которые передаются в вызывающую программу (функцию). Регистры R5 — R12 используются для передачи параметров между подпрограммами. Вторую группу образуют регистры с номерами R19 — R31. Информация, которая хранится в этих регистрах, при вызове функций (подпрограмм) записывается в стек. После выполнения функции (подпрограммы) данные, помещенные в стек перед ее вызовом, перезаписываются из стека в соответствующие регистры. В третью группу входят регистры с номерами R0 — R2 и R13 — R18. Регистр с идентификатором R0 всегда содержит нулевое значение. Любые попытки записи иных данных в этот регистр игнорируются, поэтому всегда из него можно считать нулевое значение. В регистр R1 записывается значение указателя стека. Поэтому его содержимое обновляется при вызове функций (подпрограмм) и возврате из них. Регистр R14 предназначен для хранения адреса возврата при вызове процедуры обслуживания прерывания. Регистр R15 используется для записи адреса возврата при обращении к подпрограмме. В регистр R16 заносится значение адреса возврата при обработке системных прерываний отладчика, вызванных непредвиденной ситуацией. Регистр R17 предназначен для записи адреса возврата из процедуры обработки исключений. Регистр R18 использу-

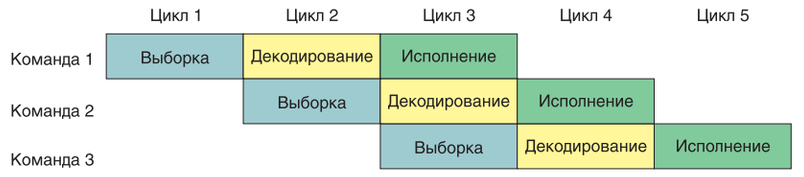

Рис. 2. Конвейерная обработка команд в микропроцессорном ядре MicroBlaze

ется для хранения временных (промежуточных) данных при выполнении ассемблерных операций.

Схема управления прерываниями формирует комбинацию внутренних сигналов, необходимых для выполнения процедуры обработки прерываний. Микропроцессорное ядро MicroBlaze изначально содержит единственную цепь сигнала прерывания. Для расширения системы прерываний следует использовать дополнительную логику, которая реализуется на основе свободных ресурсов кристалла.

Контроллеры интерфейсов шин данных и команд предназначены для организации взаимодействия микропроцессорного ядра и других компонентов проектируемой системы, включая программную и оперативную память, через соответствующие магистрали.

Буферный (конвейерный) регистр команд используется для организации конвейерной обработки инструкций, которая позволяет повысить производительность микропроцессорного ядра за счет параллельного выполнения нескольких команд. Принцип конвейерной обработки основан на расщеплении процесса обработки команды на несколько фаз и совмещении во времени выполнения различных фаз некоторой последовательности инструкций. В микропроцессорном ядре MicroBlaze процесс обработки команды разбит на три фазы:

- выборка инструкции из программной памяти;
- декодирование команды;
- исполнение инструкции.

Каждая из перечисленных фаз выполняется в течение одного машинного цикла. При этом выигрыш в производительности ядра достигается за счет одновременного выполнения (на протяжении одного машинного цикла) соответствующих фаз трех инструкций, которые последовательно выбираются из программной памяти. В течение одного машинного цикла производится выборка новой команды, декодирование предыдущей инструкции и исполнение команды, которая предшествовала двум перечисленным инструкциям (рис. 2).

Очевидно, что применение конвейерной обработки команд дает положительный эффект только при последовательной выборке инструкций из программной памяти. При выполнении команды передачи управления в программе две последующие инструкции, которые находятся в стадии выборки и дешифрации, становятся неактуальны и далее не выполняются. Для повторного заполнения конвейера новой последовательностью команд после передачи управления требуется три машинных цикла. Чтобы сократить временные потери, вызванные необходимостью перезагрузки конвейера после выполнения инструкций, управляющих последовательностью выполнения операций в программе, используется следующий метод. При исполнении команд передачи управления производится сброс только той инструкции в конвейере, которая находится в стадии выборки. Команда, находящаяся в фазе декодирования, продолжает выполняться. Таким образом, фактическая передача управления происходит только после завершения исполнения декодируемой инструкции. Для практического применения этого метода предусмотрены специальные варианты команд, управляющих последовательностью выполнения операций в программе. В конце мнемонического идентификатора таких команд присутствует символ D (Delay), указывающий на то, что исполнение этих команд осуществляется с дополнительной задержкой.

Блок декодирования команд на основании данных, поступающих с выходов программной памяти (а точнее, с выходов конвейерного регистра), формирует совокупность управляющих сигналов, необходимых для выполнения соответствующей операции, которые подаются на все блоки микропроцессорного ядра.

Управление последовательностью выполнения команд в составе программы осуществляется с помощью программного счетчика. Сигналы на его выходах образуют адрес выборки следующей команды. Эти сигналы передаются на адресные входы программной памяти. Режим работы программного счетчика (счет или загрузка) определяется типом текущей выполняемой инструкции и состоянием сигналов на входах управления микропроцессорного ядра. В основном режиме, при отсутствии прерываний, остановов и команд управления последовательностью выполнения программы, содержимое программного счетчика автоматически увеличивается на единицу при исполнении текущей операции. Таким образом реализуется последовательная выборка и выполнение команд программы.

В процессе выполнения команд управления ходом программы в программный счетчик производится загрузка значения, соответствующего адресу, по которому осуществляется передача управления в программе. После исполнения команды перехода программный счетчик продолжает работу в инкрементном режиме, но, уже начиная с нового значения, которое было записано при ее выполнении.

При появлении активного уровня сигнала на входе прерывания микропроцессорного ядра процесс выполнения текущей команды принудительно прекращается, а в программный счетчик загружается адрес вектора соответствующей процедуры обработки (0x00000010). Перед этим значение адреса инструкции, которая выполнялась в момент поступления запроса прерывания, сохраняется в регистре общего назначения R14. Для запрета будущих прерываний, запросы которых могут поступить во время выполнения процедуры обслуживания, флаг Interrupt Enable (IE) Flag регистра статуса сбрасывается в состояние низкого логического уровня. После завершения процесса обработки прерывания происходит автоматическое восстановление значения адреса невыполненной команды в программном счетчике. При этом флаг IE устанавливается в состояние высокого логического уровня, разрешающее обработку прерываний.

Если при выполнении микропроцессорной программы генерируется исключение, то процесс исполнения текущей инструкции прерывается. Адрес этой невыполненной команды записывается в регистр общего назначения R17, а в программный счетчик загружается адрес вектора процедуры обработки исключения (0x00000008). По этому адресу расположена инструкция передачи управления соответствующей подпрограмме, которая должна выполняться при возникновении события, вызывающего исключение.

В начальный момент времени функционирования микропроцессорного ядра, а также при подаче внешнего сигнала сброса выходы программного счетчика устанавливаются в нулевое состояние (низкого логического уровня). Таким образом, в этих случаях управление передается команде программы, расположенной по адресу 0x00000000.

Отладочный модуль реализуется на основе логики периферийного сканирования кристаллов FPGA. При включении этого модуля в состав микропроцессорного ядра обеспечивается возможность отладки разрабатываемых приложений в кристалле через порт JTAG, используя соответствующие программные средства EDK. В процессе отладки можно установить требуемое число контрольных точек, выполнять отлаживаемую программу в пошаговом режиме, считывать данные и производить запись в память и все регистры, включая регистр состояния и программный счетчик.

Для повышения производительности проектируемой системы в ряде случаев в состав микропроцессорного ядра MicroBlaze может быть включена быстродействующая кэш-память команд и данных. Размер этой памяти может составлять 2, 4, 8, 16, 32 и 64 кбайт.

Архитектура микропроцессорного ядра MicroBlaze предоставляет возможность выбора одного из трех вариантов реализации программной памяти, в которой хранится последовательность выполняемых команд. Первый вариант предполагает организацию памяти программ в виде запоминающего устройства, реализуемого на основе ресурсов блочной памяти кристалла Block SelectRAM. Основным недостатком этого варианта является ограниченный объем программной памяти, который определяет-

ся физическими ресурсами блочной памяти ПЛИС. Второй вариант заключается в применении внешнего (по отношению к кристаллу FPGA) ППЗУ для хранения программ. В этом случае объем программной памяти может достигать максимально допустимого значения (4 Гбайт) при включении соответствующего числа микросхем ППЗУ. Для третьего варианта организации программной памяти характерно совместное использование внутренней блочной памяти ПЛИС и внешнего ППЗУ. При этом используются различные шины команд для каждого типа программной памяти. Более подробно различные варианты конфигурации ППЗУ программ будут рассмотрены при описании шинных интерфейсов микропроцессорного ядра MicroBlaze.

Структура регистра статуса MSR микропроцессорного ядра MicroBlaze

Из 32 разрядов регистра статуса 24 бита (с 1-го по 24-й) в текущей версии ядра MicroBlaze не используются. Эти разряды регистра состояния зарезервированы для дополнительных флагов, которые могут появиться в последующих версиях ядра. Здесь и далее используется нумерация разрядов регистра, начинающаяся с нуля. Таким образом, последний разряд регистра имеет порядковый номер 31.

Нулевой разряд регистра состояния, обозначаемый как Carry Copy Flag (CC), представляет собой копию флага переноса или займа Carry Flag. Копирование значения флага переноса или займа Carry Flag в нулевой разряд MSR производится автоматически при чтении регистра статуса.

В 25-й разряд регистра состояния записывается значение флага Division by Zero Flag (DBZ). Этот флаг переключается в установленное состояние (состояние высокого логического уровня) в случае, если при выполнении операции происходит деление на нуль. При отсутствии признака деления на нуль флаг DBZ находится в состоянии низкого логического уровня.

26-й разряд регистра статуса содержит значение флага Instruction Cache Enable Flag (ICE), который указывает на возможность использования кэш-памяти команд. Если этот флаг находится в установленном состоянии, то разрешается использование кэш-памяти команд. Сброшенное состояние флага ICE указывает на запрет использования кэш-памяти команд.

27-й разряд MSR предназначен для записи состояния флага FSL Error Flag (FSL), сигнализирующего о наличии и отсутствии ошибок интерфейса FSL. Данный флаг переключается в состояние высокого логического уровня при обнаружении ошибки интерфейса FSL. При отсутствии ошибок флаг FSL находится в сброшенном состоянии.

В 28-м разряде регистра статуса содержится значение флага Break in Progress Flag (BIP), который информирует об останове в процессе выполнения программы. Процесс выполнения программы может быть временно приостановлен как программным, так и аппаратным способом. В первом случае останов осуществляется внутри программы с помощью соответствующих инструкций (brk и brki). Во втором случае приостановка процесса выполнения программы инициируется извне сигналами, которые поступают со входа Ext_Brk или Ext_NM_Brk. Когда микропроцессорное ядро переходит в режим останова, флаг BIP устанавливается в состояние высокого логического уровня. В противном случае этот флаг находится в сброшенном состоянии.

В 29-й разряд записывается состояние флага переноса-займа Carry Flag (C). Этот флаг устанавливается в состояние высокого логического уровня в том случае, если в результате операции сложения происходит перенос или заем при выполнении вычитания. При отсутствии переноса (или займа) из самого старшего разряда значение флага C соответствует состоянию низкого логического уровня.

30-й разряд регистра статуса содержит значение флага Interrupt Enable Flag (IE), который информирует о режиме обработки прерываний. Если этот флаг находится в установленном состоянии, то обработка прерываний разрешена. Сброшенное состояние флага IE соответствует режиму запрета прерываний.

В 31-й разряд регистра состояния заносится значение флага Buslock Enable Flag (BE), который сигнализирует о возможности включения режима Buslock шины данных OPB. Значение низкого логического уровня этого флага указывает на запрет указанного режима шины данных OPB. Установленное состояние флага BE соответствует разрешающему значению.

Шинные интерфейсы микропроцессорного ядра MicroBlaze

Каждая из шин микропроцессорного ядра MicroBlaze (шина данных и шина команд) фактически разделяется на две независимые магистрали:

- глобальную шину, соответствующую спецификации OPB;
- локальную шину, соответствующую спецификации LMB.

В микропроцессорной системе, разрабатываемой на основе ядра MicroBlaze, шины данных и команд могут быть реализованы как в полном составе, так и в виде только локальной LMB или только глобальной магистрали OPB. На рис. 3 показаны шесть возможных вариантов конфигурации шинных интерфейсов микропроцессорного ядра MicroBlaze. Здесь и далее используется следующая система обозначений: DOPB — шина данных, соответствующая спецификации OPB; DLMB — локальная шина данных, соответствующая спецификации LMB; IOPB — шина команд, соответствующая спецификации OPB; ILMB — локальная шина команд, соответствующая спецификации LMB.

Конфигурация 1 соответствует варианту реализации шин данных и команд в полном составе (IOPB+ILMB+DOPB+DLMB). На рис. 4 показана типовая схема подключения стандартных периферийных модулей и памяти, соответствующая этой конфигурации. Глобальная шина команд, соответствующая спецификации OPB, используется для подключения внешней программной памяти через соответствующий контроллер. Локальная шина команд, соответствующая спецификации LMB, обеспечивает быстрый доступ к памяти программ, реализуемой на основе блочной памяти кристалла. Внешняя память данных и периферийные модули (реализуемые как внутри, так и вне ПЛИС) подключаются к глобальной шине данных DOPB. Локальная шина данных DLMB выполняет функции сопряжения микропроцессорного ядра MicroBlaze и оперативной памяти, реализуемой на базе блочной памяти кристалла. Данную конфигурацию целесообразно использовать в случае, когда необходим большой объем программной и оперативной памяти, превосходящей ресурсы блочной памяти ПЛИС. При этом для хранения крити-

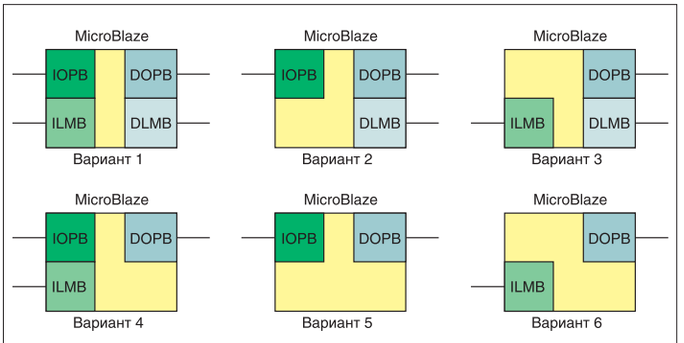

Рис. 3. Варианты конфигурации шинных интерфейсов микропроцессорного ядра MicroBlaze

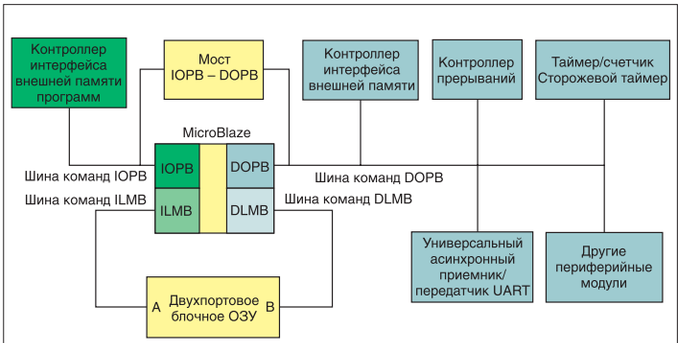

Рис. 4. Вариант 1 конфигурации микропроцессорной системы, проектируемой на основе ядра MicroBlaze

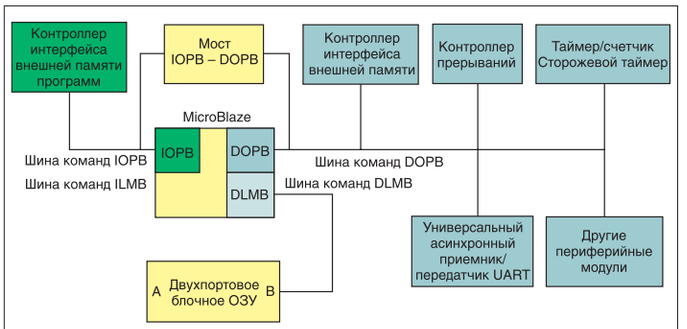

Рис. 5. Вариант 2 конфигурации микропроцессорной системы, проектируемой на основе ядра MicroBlaze

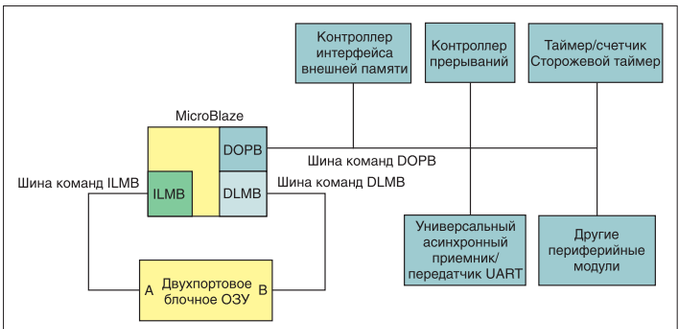

Рис. 6. Вариант 3 конфигурации микропроцессорной системы, проектируемой на основе ядра MicroBlaze

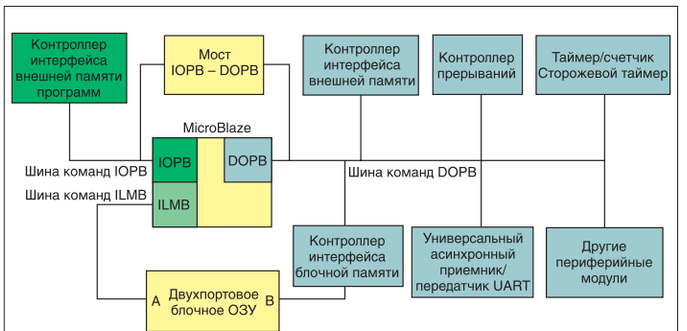

Рис. 7. Вариант 4 конфигурации микропроцессорной системы, проектируемой на основе ядра MicroBlaze

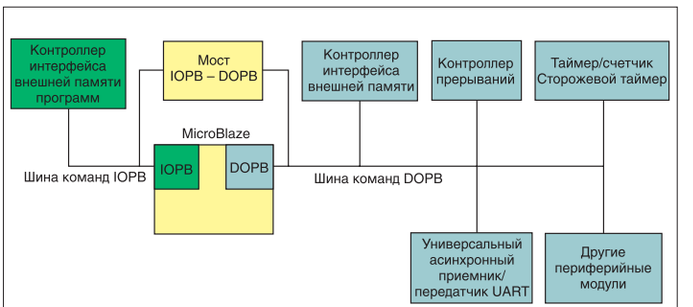

Рис. 8. Вариант 5 конфигурации микропроцессорной системы, проектируемой на основе ядра MicroBlaze

ческих сегментов программного кода и данных используется блочная память кристалла.

Конфигурация 2 (IOPB+DOPB+DLMB) отличается от предыдущей отсутствием локальной шины команд ILMB (рис. 5). При этом программная память полностью выполняется в виде внешнего ППЗУ, взаимодействие с которым осуществляется по глобальной шине команд IOPB через соответствующий контроллер. Сопряжение оперативной памяти и периферийных модулей с микропроцессорным ядром выполняется по той же схеме, что и в конфигурации 1. Конфигурация 2 применяется при больших объемах программного кода и данных, выходящих за пределы ресурсов блочной памяти кристалла FPGA, и реализации памяти программ вне ПЛИС. При использовании данной конфигурации разработчик может задействовать максимальный объем блочной памяти кристалла для организации оперативной памяти с высокой скоростью доступа по шине DLMB.

В конфигурации 3 (ILMB+DOPB+DLMB) по сравнению с конфигурацией 1 исключена глобальная шина команд IOPB (рис. 6). ППЗУ программ в этом случае в полном объеме реализуется на основе блочной памяти ПЛИС. Взаимодействие микропроцессорного ядра MicroBlaze с оперативной памятью и периферийными модулями осуществляется так же, как и в конфигурациях, рассмотренных выше. Конфигурация 3 эффективна при небольшом объеме программного кода, для хранения которого достаточно ресурсов внутренней блочной памяти кристалла. При этом достигается высокая скорость выборки команд программы за счет использования локальной шины команд ILMB.

В отличие от конфигурации 1 в конфигурации 4 (IOPB+ILMB+DOPB) отсутствует локальная шина данных DLMB (рис. 7). Шина команд и программная память организованы так же, как и в конфигурации 1. Внешняя оперативная память и периферийные модули подключаются к глобальной шине данных DOPB. Для отладки программного кода может применяться дополнительный контроллер BRAM Memory Controller, обеспечивающий доступ к внутренней блочной памяти через глобальную шину данных DOPB. Данная конфигурация используется в тех случаях, когда объем разрабатываемой программы и данных превосходит емкость внутренней блочной памяти кристалла. Критические фрагменты программного кода записываются во внутреннее ППЗУ, реализуемое на основе блочной памяти ПЛИС.

В конфигурации 5 (IOPB+DOPB) шины данных и команд представлены только глобальными магистралями, соответствующими спецификации OPB (рис. 8). Для хранения кода программ и данных используется внешняя память. Сопряжение памяти данных и периферийных модулей с микропроцессорным ядром осуществляется теми же способами, что и в конфигурации 1. Конфигурация 5 применяется, как правило, при больших объемах программного кода и данных, которые не содержат критических секций.

Наиболее компактной является конфигурация 6 (ILMB+DOPB), в которой задействованы только локальная шина команд ILMB и глобальная шина данных DOPB (рис. 9). Программная память в данной конфигурации формируется на базе блочной памяти кристалла FPGA, доступ к которой производится по быстрой локальной шине ILMB, что обеспечивает минимальное время выборки команд. Конфигурацию 6 целесообразно использовать при небольшом объеме программного кода, не превосходящем емкости блочной памяти ПЛИС.

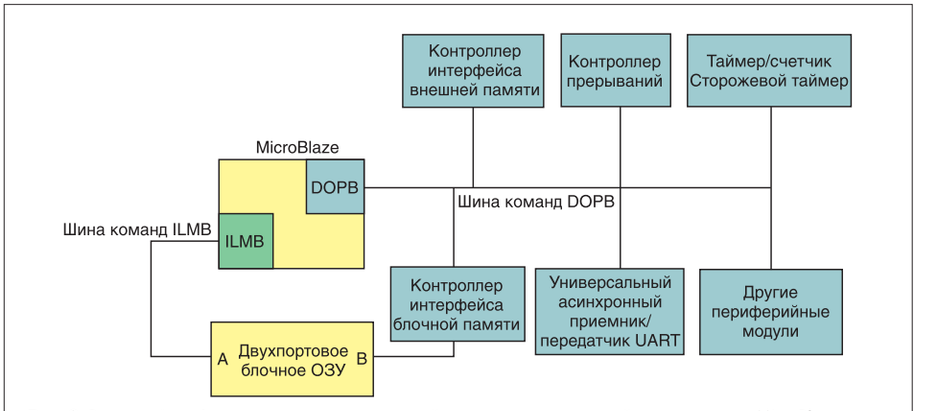

Рис. 9. Вариант 6 конфигурации микропроцессорной системы, проектируемой на основе ядра MicroBlaze

Периферийные компоненты для микропроцессорного ядра MicroBlaze

Пакет программных средств разработки встраиваемых микропроцессорных систем EDK помимо ядра MicroBlaze содержит набор компонентов периферийных модулей, которые представлены в форме IP-ядер (Intellectual Property). При проектировании «систем на кристалле» на основе микропроцессорного ядра MicroBlaze, разработчик может использовать следующие IP-модули:

- таймер-счетчик;
- сторожевой таймер;
- универсальный асинхронный приемник-передатчик UART;
- контроллер прерываний Interrupt Controller;
- контроллер стандартного интерфейса ввода-вывода;
- контроллер интерфейса памяти SDRAM;
- контроллер интерфейса памяти DDR SDRAM;
- контроллер интерфейса памяти Flash Memory;
- контроллер интерфейса блочной памяти BlockRAM;
- контроллер интерфейса Serial Peripheral Interface, соответствующего спецификации фирмы Motorola;
- универсальный асинхронный приемник-передатчик UART, использующий для коммуникации JTAG-порт.

Все перечисленные выше IP-модули включены в состав EDK в полнофункциональном варианте. Кроме того, в комплект EDK входят ознакомительные версии (которые могут использоваться только на протяжении некоторого ограниченного периода времени) следующих IP-ядер:

- универсальные асинхронные приемники-передатчики UART, совместимые со стандартами UART 16550 и UART 16450 фирмы National Semiconductor;
- контроллер сетевого интерфейса Ethernet 1Gb (1Gb Ethernet Controller);
- контроллеры сетевого интерфейса Ethernet Media Access Controller (EMAC) и EMAC Lite 10/100 Mbps, соответствующие спецификации IEEE Std. 802.3 Media Independent Interface;
- контроллер интерфейса High Level Data Link Control.

IP-модули, представленные в виде ознакомительных версий, поставляются отдельно, как самостоятельные продукты. С полным списком IP-ядер, предлагаемых фирмой Xilinx для проектирования микропроцессорных систем на основе ПЛИС различных семейств, можно ознакомиться на сайте **ww w .xilinx.c om**.

На этом завершается описание архитектурных особенностей микропроцессорного ядра MicroBlaze. В следующей публикации будет представлена система команд данного ядра.

Литература

1. Зотов В. PicoBlaze — семейство восьмиразрядных микропроцессорных ядер, реализуемых на основе ПЛИС фирмы Xilinx // Компоненты и Технологии. 2003. № 4.
2. Зотов В. Система команд микропроцессорного ядра PicoBlaze, реализуемого на основе ПЛИС семейств Spartan-II, Spartan-IIE, Virtex, Virtex-E // Компоненты и Технологии. 2003. № 5.
3. Зотов В. Особенности микропроцессорного ядра PicoBlaze, предназначенного для применения в проектах, реализуемых на основе ПЛИС семейства Virtex-II // Компоненты и Технологии. 2003. № 6.
4. Зотов В. Особенности микропроцессорного ядра PicoBlaze, предназначенного для применения в проектах, реализуемых на основе ПЛИС семейства CoolRunner-II // Компоненты и Технологии. 2003. № 7.
5. Зотов В. Разработка программ на языке ассемблера для семейства микропроцессорных ядер PicoBlaze // Компоненты и Технологии. 2003. № 8.
6. Зотов В. Проектирование цифровых устройств на основе ПЛИС фирмы Xilinx в САПР WebPack ISE. М.: Горячая линия — Телеком. 2003.
7. Зотов В. ModelSim — система HDL-моделирования цифровых устройств // Компоненты и Технологии. 2002. № 6.
8. Зотов В. Функциональное моделирование цифровых устройств, проектируемых на базе ПЛИС фирмы Xilinx в среде САПР WebPACK ISE // Компоненты и Технологии. 2002. № 7.
9. Зотов В. Временное моделирование цифровых устройств, проектируемых на базе ПЛИС фирмы Xilinx в среде САПР WebPACK ISE // Компоненты и Технологии. 2002. № 8.
10. Spartan-3 FPGA Handbook. Xilinx, Inc. 2003.