

Практический курс сквозного проектирования цифровых устройств на основе ПЛИС фирмы Xilinx









(часть 8)

Валерий Зотов (Москва)

В восьмой части курса основное внимание уделяется применению временных и топологических ограничений в проектах, реализуемых на базе ПЛИС фирмы Xilinx с архитектурой CPLD и FPGA. Представлен формат выражений типичных временных и топологических ограничений. Кратко рассмотрены методы формирования файлов временных и топологических ограничений в САПР серии Xilinx ISE.

КРАТКОЕ ОПИСАНИЕ СТРУКТУРЫ РАБОЧЕГО ОКНА РЕДАКТОРА ВРЕМЕННЫХ ДИАГРАММ TEST BENCH WAVEFORM EDITOR

В верхней строке рабочего окна редактора временных диаграмм *Test Bench Waveform Editor* расположена временная шкала, деления которой соответствуют выбранным ранее единицам измерения длительности тактового сигнала. В левом столбце отображается список названий интерфейсных сигналов устройства. Слева от названия сигнала приведена пиктограмма, обозначающая его тип. Редактором временных диаграмм *Test Bench Waveform Editor* поддерживаются следующие типы сигналов:

-  – сигнал входного порта;
-  – сигнал выходного порта;
-  – сигнал двунаправленного (входного/выходного) порта;
-  – сигнал входного порта векторного типа, представленный в виде шины;
-  – сигнал выходного порта векторного типа, представленный в виде шины;
-  – сигнал двунаправленного (входного/выходного) порта векторного типа, представленный в виде шины;
-  – внутренний сигнал, не связанный с портом;
-  – не связанный с портом внутренний сигнал векторного типа, представленный в виде шины.

Сигналы векторного типа представлены шинами соответствующей разрядности. Численные значения таких

сигналов отображаются непосредственно на изображении временной диаграммы. Эти значения могут быть представлены в двоичном (*Binary*), десятичном (*Decimal*) или шестнадцатеричном (*Hexdecimal*) форматах. Изменение формата представления значений осуществляется с помощью соответствующих команд контекстно-зависимого всплывающего меню, открываемого щелчком правой кнопки мыши при расположении курсора на изображении временной диаграммы выбранного сигнала. Для отображения временных диаграмм всех сигналов, входящих в состав шины, следует поместить курсор мыши на знак «+», расположенный слева от пиктограммы шины, и щёлкнуть левой кнопкой.

На изображении временных диаграмм присутствует указатель конечной точки тестовой последовательности, положение которого устанавливается автоматически на основании информации, указанной с помощью «мастера» инициализации временных параметров процесса моделирования и выбора тактовых сигналов *Initial Timing and Clock Wizard*. Для изменения длительности временных диаграмм тестовых сигналов следует воспользоваться командой *Set End of Test Bench* из всплывающего меню *Test Bench* или контекстно-зависимого меню, открываемого щелчком правой кнопки мыши. При выполнении этой команды на экран выводится диалоговая панель с одноименным заголовком. Новое значение и единицы измерения длительности будут указаны в соответствующи-

х полях редактирования и выбора открывшейся диалоговой панели.

ФОРМИРОВАНИЕ ВРЕМЕННЫХ ДИАГРАММ ТЕСТОВЫХ СИГНАЛОВ ДЛЯ МОДЕЛИРУЕМОГО УСТРОЙСТВА

Формирование временных диаграмм тестовых сигналов для моделируемого устройства может осуществляться несколькими способами. Самым простым и наглядным способом является последовательное изменение значения сигнала на временной диаграмме в соответствующие моменты времени. Чтобы переключить значение сигнала на противоположное (из состояния низкого логического уровня в высокое и наоборот) в какой-либо момент времени, достаточно поместить курсор в требуемое место временной диаграммы и щёлкнуть левой кнопкой мыши.

Для явного указания значения сигнала следует щелчком правой кнопки активизировать контекстно-зависимое всплывающее меню и выбрать в нём строку *Set Value* или дважды щёлкнуть левой кнопкой мыши. В результате выполнения указанных действий на экран выводится диалоговая панель с одноименным заголовком, содержащая поле редактирования значения сигнала. Для установки требуемого значения сигнала следует ввести условное обозначение требуемого уровня сигнала, соответствующее списку возможных состояний для данного типа сигналов, после чего подтвердить выбор нажатием кнопки *OK* в нижней части этой диалоговой панели.

Чтобы сформировать последовательность значений для выбранного сигнала, можно воспользоваться соответствующим «мастером» *Pattern Wizard*. Для его активизации необходимо в диалоговой панели *Set Value* нажать кнопку *Pattern Wizard*. В открывшейся диалоговой панели «мастера» *Pattern Wizard* сле-

дует выбрать тип формируемой последовательности значений (алгоритм переключений сигнала) в поле *Pattern Type*. Данное поле выбора находится во встроенной панели *General Pattern Information*. В этой же встроенной панели расположено поле редактирования *Number of Cycles*, в котором указывается число повторений формируемой последовательности значений сигнала.

Содержание выпадающего списка значений поля выбора *Pattern Type* зависит от вида сигнала. Для одиночного сигнала предлагается три типа формируемой последовательности значений, которым соответствуют три варианта, представленные в выпадающем списке: *Pulse*, *Random Signal* и *Toggle*. Для каждого типа последовательности значений сигнала необходимо указать соответствующие временные параметры с помощью одноименных полей редактирования, которые представлены во встроенной панели *Pattern Parameters* диалоговой панели «мастера» *Pattern Wizard*.

Если в поле выбора *Pattern Type* указан вариант *Pulse*, то формируемая последовательность значений соответствует одиночному импульсу. Для этого варианта необходимо указать начальное значение сигнала в поле редактирования *Initial Value*, длительность начальной задержки импульса – в поле редактирования *Initial Delay*, логический уровень импульсного сигнала – в поле редактирования *Pulse Value* и его длительность – в поле редактирования *Pulse Width*.

При выборе варианта *Random Signal* будет сформирована случайная последовательность значений сигнала. В этом случае необходимо определить начальное значение последовательности в поле редактирования *Random Seed* и число циклов тактового сигнала, после которого вычисляется следующее значение последовательности в поле редактирования *Count Every*.

Если выбран вариант *Toggle*, то формируемый сигнал будет иметь форму меандра. Начальное значение этого сигнала указывается в поле редактирования *Initial Value*, альтернативное значение – в поле редактирования *Other Value*, интервал переключения (длительность каждого состояния сигнала) – в поле редактирования *Toggle Every*.

Для сигналов векторного типа, представленных в виде шин, выпадающий список поля выбора *Pattern Type* содержит следующие варианты: *Alternate*,

Count Down, *Count Up*, *Random Bus*, *Shift Left*, *Shift Right*. Выбор варианта *Alternate* предоставляет пользователю возможность определения двух последовательных значений сигнала с указанием их длительности. Значения сигнала задаются в полях редактирования *A* и *B* соответственно. Длительность указанных состояний сигнала, эквивалентная числу периодов тактового сигнала, определяется с помощью полей редактирования *Assign A For* и *Assign B For*.

Чтобы задать поведение выбранного сигнала в виде серии последовательно убывающих значений с указанной длительностью каждого состояния, следует в выпадающем списке поля выбора *Pattern Type* выделить вариант *Count Down*. При этом начальное и конечное значения серии определяются с помощью полей редактирования *Initial Value* и *Terminal Value* соответственно. Значение декремента указывается в поле редактирования *Decrement By*. Длительность каждого состояния сигнала, выраженная числом периодов тактового сигнала, определяется с помощью поля редактирования *Count Every*.

Для определения поведения сигнала в виде серии последовательно возрастающих значений с указанной длительностью каждого состояния необходимо выбрать вариант *Count Up*. В этом случае следует установить требуемые значения тех же параметров, что и для варианта *Count Down*. Однако значение декремента заменяется значением инкремента, которое указывается в поле редактирования *Increment By*.

В случае выбора варианта *Random Signal* будет сформирована серия случайных последовательных значений сигнала. Для этого необходимо определить начальное значение серии в поле редактирования *Random Seed* и длительность каждого состояния сигнала (выраженную числом циклов тактового сигнала) с помощью поля редактирования *Count Every*.

При выборе варианта *Shift Left* формируемый участок временной диаграммы будет соответствовать последовательности значений сигнала, получаемых поочередным сдвигом начального значения на один разряд влево (в сторону младшего значащего разряда). Начальное значение указывается в поле редактирования *Initial Value*. Значение, используемое для заполнения освобождающегося при сдвиге разряда, задаётся в поле редактирования *Shift In*. Длительность каждого состоя-


ния сигнала, полученного в результате сдвига, определяется с помощью поля редактирования *Shift Every* в виде числа циклов тактового сигнала.

Если в поле выбора *Pattern Type* указан вариант *Shift Right*, то формируемый участок временной диаграммы будет соответствовать последовательности значений сигнала, получаемых путём поочередного сдвига начального значения на один разряд вправо (в сторону старшего значащего разряда). Для этого варианта необходимо указать требуемые значения тех же параметров, что и для варианта *Shift Left*.

Значения сигналов могут задаваться в двоичном (*Binary*), десятичном (*Decimal*) или шестнадцатеричном (*Hexadecimal*) форматах. Выбор формата представления значений сигналов осуществляется с помощью одноименных кнопок с зависимой фиксацией, расположенных во встроенной панели *Radix*.

Завершение работы «мастера» *Pattern Wizard* осуществляется нажатием кнопки *OK*, расположенной в нижней части диалоговой панели. После этого сформированный фрагмент временной диаграммы отображается в рабочей области окна редактора временных диаграмм *Test Bench Waveform Editor*.

Если необходимо изменить значения временных параметров внешнего тактового сигнала, которые были заданы с помощью «мастера» *Initial Timing and Clock Wizard*, то следует выполнить команду *Rescale Timing..* из всплывающего меню *Test Bench* или контекстно-зависимого всплывающего меню. При этом на экран выводится диалоговая панель с заголовком *Rescale Timing*, которая позволяет отредактировать значения всех параметров каждого внешнего тактового сигнала.

Завершив редактирование временных диаграмм тестовых сигналов, следует сохранить их, используя команду *Save* из меню *File* или кнопку  на оперативной панели управления Навигатора проекта. При этом кроме файла временных диаграмм с расширением *tbw*, автоматически создаётся файл с тем же именем и расширением *vhw*, в котором содержится текст описания тестовой системы и соответствующих сигналов на языке VHDL. Тем самым пользователю предоставляется возможность последующего редактирования временных диаграмм тестовых сигналов как в графическом, так и в текстовом виде.

ИСПОЛЬЗОВАНИЕ ВРЕМЕННЫХ И ТОПОЛОГИЧЕСКИХ ОГРАНИЧЕНИЙ В ПРОЕКТАХ, РЕАЛИЗУЕМЫХ НА БАЗЕ ПЛИС ФИРМЫ XILINX

Кроме модулей исходного описания разрабатываемого устройства, в состав нового проекта САПР серии Xilinx ISE необходимо добавить информацию для программ синтеза, размещения и трассировки в кристалле ПЛИС. Эта информация может располагаться непосредственно в модулях исходного описания проекта в форме атрибутов или может быть вынесена в отдельный модуль (файл временных и топологических ограничений проекта *User Constraints File, UCF*). При использовании механизма атрибутов теряется универсальность модулей исходного описания. Для их последующего применения в других проектах, которые могут быть реализованы на базе ПЛИС различных семейств, каждый раз потребуется вносить изменения в выражения, определяющие значения временных и топологических параметров. Вынесение этих параметров в отдельный модуль *UCF* позволяет преодолеть указанный недостаток. Использование файла *UCF* является предпочтительным не только потому, что при этом сохраняется универсальность модулей исходного описания, которые могут использоваться в других проектах. Чтобы внести коррективы в параметры ограничений, например, при изменении типа ПЛИС, не требуется редактирование различных модулей исходного описания, а достаточно скорректировать содержимое файла *UCF*.

В файлах временных и топологических ограничений могут определяться значения параметров как локального, так и глобального характера. Объектами локальных ограничений являются отдельные экземпляры элементов описания проекта (цепи, компоненты, контакты). Глобальные параметры относятся ко всему проекту в целом или оказывают влияние на все элементы проекта, тип которых указан в соответствующем выражении. Большинство глобальных ограничений удобнее задавать в виде параметров выполняемого этапа (процесса) с помощью соответствующих диалоговых панелей Навигатора проекта. Поэтому в файлах временных и топологических ограничений целесообразно задавать значения параметров, которые носят локальный харак-

тер, и глобальные ограничения, отсутствующие в диалоговых панелях опций процессов Навигатора проекта.

Полное описание всех временных и топологических параметров, а также синтаксис соответствующих выражений ограничений можно найти в документации САПР серии Xilinx ISE (*Constraints Guide*). В последующих разделах приводится описание параметров и синтаксис выражений ограничений, которые наиболее часто используются в процессе синтеза и реализации проектов на базе ПЛИС семейств как CPLD, так и FPGA.

ОПИСАНИЕ ТИПИЧНЫХ ТОПОЛОГИЧЕСКИХ ПАРАМЕТРОВ И СИНТАКСИС СООТВЕТСТВУЮЩИХ ВЫРАЖЕНИЙ ОГРАНИЧЕНИЙ

Параметр *LOC* позволяет осуществить закрепление выводов перед трассировкой, а также явно указать функциональный (*Function Block, FB*) или конфигурируемый логический блок (*Configurable Logic Block, CLB*), в зависимости от выбранного типа ПЛИС, для реализации элементов проекта. Для привязки «внешних» цепей проекта (подключаемых к контактам кристалла) к требуемым выводам ПЛИС используется следующий формат выражения:

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

Форма записи номера вывода ПЛИС в этом выражении зависит от типа используемого корпуса микросхемы. Для корпусов, относящихся к категории BGA или CSP, номер вывода указывается в том же виде, в каком он представлен в документации. При использовании корпусов других типов номер вывода ПЛИС в выражении ограничения *LOC* сопровождается префиксом P. В качестве примеров топологических ограничений, используемых для указания соответствия внешних цепей разрабатываемого устройства и номеров выводов ПЛИС, приводятся следующие выражения:

```
NET clock LOC=A4;
NET dat1 LOC=P12;
```

Чтобы в кристалле с архитектурой FPGA определить конфигурируемый логический блок CLB, внутри которого требуется разместить

цепь или элемент проекта, следует воспользоваться соответственно форматами:

```
NET <название_цепи> LOC=
CLB_<координаты_конфигурируемо-
го_логического_блока_ПЛИС>;
```

например:

```
NET clock LOC= CLB_R2C3;

INST <обозначение_элемента> LOC=
CLB_<координаты_конфигурируемо-
го_логического_блока_ПЛИС>;
```

например:

```
INST D10 LOC= CLB_R3C4;
```

Для определения функционального блока, внутри которого требуется разместить цепь или элемент проекта в ПЛИС с архитектурой CPLD, следует воспользоваться соответственно форматами:

```
NET <название_цепи> LOC=FB<но-
мер_функционального_блока_ПЛИС>;
```

например:

```
NET clock LOC=FB2;

INST <обозначение_элемента>
LOC=FB<номер_функционального_бло-
ка_ПЛИС>;
```

например:

```
INST D10 LOC=FB3;
```

ОПИСАНИЕ ТИПИЧНЫХ ВРЕМЕННЫХ ПАРАМЕТРОВ И СИНТАКСИС СООТВЕТСТВУЮЩИХ ВЫРАЖЕНИЙ ОГРАНИЧЕНИЙ

Временные ограничения могут устанавливаться как для отдельных элементов схемы (например, цепей), так и для совокупности элементов определённого типа. Такие совокупности называются временными группами. Вначале рассмотрим временные ограничения, относящиеся к единичному элементу схемы.

Максимальное значение периода сигнала синхронизации для соответствующей цепи проекта задаётся с помощью параметра *PERIOD*. Полный формат выражения ограничения имеет вид:

```
NET <название_цепи_синхронизации>
PERIOD=<длительность_периода>
[<единицы_измерения>] [{HIGH |
LOW}]<длительность_первой_фазы_пе-
риода> [<единицы_измерения>] [];
```

где значение *HIGH* или *LOW* указывает логический уровень сигнала в первой фазе периода. По умолчанию в качестве единиц измерения длительности установлены наносекунды и одинаковая продолжительность состояний высокого и низкого уровня периода синхросигнала, в результате чего получается сокращённый формат записи:

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

например:

```
NET clock PERIOD=25 ns
```

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

Параметр *OFFSET* позволяет установить предельные временные соотношения между тактовым сигналом и связанными с ним сигналами входных и выходных цепей, подключаемых к выводам кристалла. Синтаксис соответствующего выражения ограничения выглядит следующим образом:

```
NET <название_цепи_подключае-
мой_к_выводу_ПЛИС> OFFSET =
{IN|OUT} <длительность_задержки>
[<единицы_измерения>]
{BEFORE|AFTER} <название_це-
пи_синхронизации>;
```

Например:

```
NET dat_input OFFSET = IN 30
BEFORE clock;
```

задаёт максимальное время установления сигнала для входной цепи *dat_input* по отношению к тактовому сигналу *clock* равным 30 нс;

```
NET dat_output OFFSET = OUT 15
AFTER clock_sys;
```

устанавливает максимальное значение задержки выходного сигнала *dat_output* по отношению к тактовому сигналу *clock_sys* – 15 нс.

Временные группы могут создаваться различными способами. Наиболее универсальным является использование параметра *TIMEGRP*. В общем случае используется следующий формат выражения:

```
TIMEGRP <название_создавае-
мой_группы> = <условное_обозначе-
ние_предопределенной_группы>
[(<названия_компонентов>)]
[EXCEPT <условное_обозначение_ти-
па_элемента> (<названия_исключае-
мых_компонентов>)];
```

Например, выражение

```
TIMEGRP FFR= RISING FFS;
```

формирует временную группу *FFR*, объединяющую все триггеры, тактируемые фронтом сигнала синхронизации; выражение

```
TIMEGRP FFND5 = FFS EXCEPT
FFS("D5");
```

создаёт временную группу *FFND5*, в которую входят все триггеры за исключением *D5*.

Примерами предопределённых групп являются: *FFS* – триггеры, *RAMS* – синхронные ОЗУ, *LATCHES* – защёлки, *PADS* – выводы (контакты) кристалла.

Для создания новых временных групп на основе уже имеющихся рекомендуется воспользоваться соответствующим форматом выражения ограничения:

```
TIMEGRP <название_создавае-
мой_группы> = <название_времен-
ной_группы1> [, <название_времен-
ной_группы2> ,...] [EXCEPT <назва-
ние_исключаемой_группы_1> [, <назва-
ние_исключаемой_группы2> ,...]];
```

Например, выражение

```
TIMEGRP NEW_GROUP = GROUP1,
GROUP4 EXCEPT GROUP5;
```

включает в состав создаваемой группы элементы, которые входят в состав временных групп *GROUP1* и *GROUP4*, исключая элементы, находящиеся в группе *GROUP5*.

Ограничения для временных групп записываются с помощью ключевого слова *TIMESPEC*. Наиболее часто используются следующие конструкции групповых временных ограничений. Первая синтаксическая конструкция устанавливает значение параметра *PERIOD*, рассмотренного выше, для указанной временной группы:

```
TIMESPEC <TS_идентификатор_специ-
фикации> = PERIOD <название_вре-
менной_группы> <длительность_пе-
риода> [<единицы_измерения>]
[{HIGH | LOW}]<длительность_пер-
вой_фазы_периода> [<единицы_изме-
рения>] [];
```

Идентификатор временной спецификации всегда должен начинаться с *TS_*, например:

```
TIMESPEC TS_syn = PERIOD
SYNC_GROUP 55 HIGH 30
```

Вторая конструкция предназначена для задания временных соотношений между элементами сформированных или предопределённых временных групп:

```
TIMESPEC<TS_идентификатор_специ-
фикации> = FROM <название_вре-
менной_группы1> TO <название_вре-
менной_группы2> <длительность_за-
держки> [<единицы_измерения>];
```

Например:

```
TIMESPEC TS_del = FROM FF_GROUP
TO PAD_GROUP 25 ns;
```

Максимальное значение задержки распространения сигнала для соответствующей цепи проекта задаётся с помощью параметра *MAXDELAY*. Полный формат выражения ограничения имеет вид:

```
NET <название_цепи> MAXDELAY=<макс-
имальное_значение_задержки> <еди-
ницы_измерения>.
```

В качестве единиц измерения длительности могут использоваться наносекунды, микросекунды, миллисекунды, пикосекунды. Пример использования временного ограничения *MAXDELAY*:

```
NET data_in MAXDELAY=5.0 ns;.
```

Для указания максимально допустимого разброса задержек распространения сигнала по цепи, которая содержит одно или несколько ответвлений, предназначен параметр *MAXSKEW*. Так как ветви цепи могут иметь различную длину и топологию, возникает временной перекося изменения сигнала данной цепи на входах элементов, которые к ней подключены. В ряде случаев, если разброс запаздывания сигнала в различных ветвях цепи превышает некоторую величину, такая ситуация может привести к сбою функционирования проектируемого устройства. Значение параметра *MAXSKEW* задаёт максимально допустимую величину временного перекося. Синтаксис соответствующего выражения, устанавливающего данное ограничение, имеет следующий вид:

```
NET <название_цепи> MAXSKEW=<максимальное_значение_разброса_задержки> [<единицы_измерения>],
```

например:

```
NET data_load MAXSKEW = 3 ns
```

В качестве единиц измерения временного разброса используются те же обозначения, что и в выражении, определяющем максимальное значение задержки распространения сигнала *MAXDELAY*. По умолчанию, если единицы измерения не указаны явно, используются наносекунды.

Параметр *NODELAY* позволяет минимизировать задержку сигналов на информационных входах триггеров, расположенных в блоках ввода-вывода кристалла. По умолчанию при конфигурировании триггеров, входящих в состав блоков ввода-вывода ПЛИС семейств FPGA, устанавливается дополнительная задержка по входам данных, которая компенсирует запаздывание тактового сигнала. С помощью выражения ограничения, синтаксис которого приведён ниже, эта задержка исключается для указанного элемента. Параметр *NODELAY* может быть задан для триггера, расположенного в соответствующем блоке ввода-вывода, или для цепи, подключенной к контакту кристалла. Поэтому для данного параметра используются два варианта формата выражения ограничения:

```
INST <обозначение_триггера_блока_ввода-вывода> NODELAY;
```

например:

```
INST in5_ff NODELAY;
NET <название_цепи_подключенной_к_выводу_кристалла> NODELAY;
```

например:

```
NET data_in_5 NODELAY;
```

ОПИСАНИЕ ТИПИЧНЫХ ВСПОМОГАТЕЛЬНЫХ ПАРАМЕТРОВ И СИНТАКСИС СООТВЕТСТВУЮЩИХ ВЫРАЖЕНИЙ ОГРАНИЧЕНИЙ

Для более точного расчёта задержек распространения сигналов можно использовать параметр *VOLTAGE*, который позволяет учесть зависимость временных параметров кристалла от напряжения питания проектируемого устройства. В случае установки данного ограничения производится масштабирование всех задержек распространения сигналов в кристалле в соответствии с указанным значением напряжения питания. Синтаксис выражения, устанавливающего значение параметра *VOLTAGE*, имеет следующий вид:

```
VOLTAGE = <значение_напряжения_источника_питания> [V];
```

где V указывает единицы измерения напряжения (В). Например,

```
VOLTAGE = 5.
```

Следует обратить внимание на то, что параметр *VOLTAGE* может применяться только для ПЛИС, предназначенных для работы в коммерческом диапазоне напряжения питания. Если указывается значение, которое выходит за пределы диапазона, допустимого для выбранного типа кристаллов, ограничение игнорируется. При этом выводится сообщение об ошибке, а для проекта устанавливается типовое значение напряжения питания, принятое по умолчанию для выбранного семейства кристаллов.

Коррекция задержек распространения сигналов с учётом температуры, при которой предполагается эксплуатация проектируемого устройства, осуществляется с помощью параметра *TEMPERATURE*. При использовании этого ограничения производится масштабирование всех задержек распространения сигналов в кристалле в соответствии с указанным значением температуры.

Формат соответствующего выражения ограничения имеет вид:

```
TEMPERATURE = <значение_температуры> [<единицы_измерения>];
```

В качестве единиц измерения температуры могут указываться обозначения С (градусы по Цельсию), F (по Фаренгейту), К (по Кельвину). Например,

```
TEMPERATURE = 25 C;
```

Ограничение *TEMPERATURE* может устанавливаться только в проектах, для реализации которых используются кристаллы, предназначенные для работы в коммерческом диапазоне температур.

Большинство ПЛИС семейств FPGA обладает возможностью программирования нагрузочной способности выходных каскадов для каждого блока ввода-вывода. Для этого используется параметр *DRIVE*, который позволяет установить требуемую мощность для каждого выходного контакта кристалла. В качестве объектов, для которых указывается параметр *DRIVE*, могут выступать выходные компоненты блоков ввода-вывода (выходные буферы *OBUF*, выходные триггеры *OFD*) или цепи, подключаемые к выходным контактам ПЛИС. Соответствующие форматы выражений ограничений выглядят следующим образом:

```
INST <обозначение_выходного_элемента_блока_ввода-вывода> DRIVE = <значение_выходного_тока_в_мА>;
```

например,

```
INST out_buf1 DRIVE = 24;
```

```
NET <название_цепи_подключенной_к_выходу_кристалла> DRIVE = <значение_выходного_тока_в_мА>;
```

например,

```
NET data_out DRIVE = 24
```

Для кристаллов серий XC4000XV, XC4000XLA и Spartan™XL допустимыми значениями выходного тока являются 12 и 24 мА. При использовании ПЛИС серий Spartan-II, Virtex™, Virtex-E, Virtex-II и Virtex-II Pro список возможных значений содержит шесть позиций: 2, 4, 6, 8, 12, 16 и 24 мА. По умолчанию используется значение выходного тока 12 мА.

ОПИСАНИЕ ТИПИЧНЫХ ПАРАМЕТРОВ ИНИЦИАЛИЗАЦИИ ЭЛЕМЕНТОВ ЗАПОМИНАЮЩИХ УСТРОЙСТВ И СИНТАКСИС СООТВЕТСТВУЮЩИХ ВЫРАЖЕНИЙ ОГРАНИЧЕНИЙ

Инициализация элементов запоминающих устройств (ОЗУ, ПЗУ, таблиц преобразования, регистров) осуществляется с помощью параметра INIT, формат выражения которого имеет следующий вид:

```
INST <обозначение_элемента_памяти> INIT = <значение_определяющее_содержимое_элемента_памяти>;
```

например,

```
INST rom1 INIT = 4076
```

Значение, определяющее содержимое элемента запоминающего устройства, указывается в шестнадцатеричном виде. При этом количество знаков шестнадцатеричного числа должно соответствовать разрядности элемента запоминающего устройства.

Формат выражения инициализации триггеров и регистров имеет следующий вид:

```
INST <позиционное_обозначение_триггера_или_регистра> INIT={1 | 0 | S | R};
```

где S – соответствует режиму установки, а R – режиму сброса, например,


```
INST dd5 INIT=1.
```

Все значения параметров, указываемые в файле временных и топологических ограничений, не должны выходить за пределы допустимых диапазонов для выбранного типа ПЛИС.

Подготовка основы нового модуля временных и топологических ограничений проекта

Файл временных и топологических ограничений проекта имеет текстовый формат, каждая строка которого представляет собой выражение, описывающее соответствующий параметр. Для внесения информации в файл UCF можно использовать встроенный HDL-редактор или специальную программу *Constraints Editor* пакета САПР серии Xilinx ISE, которая на основании данных, указанных разработчиком в диа-

логовом режиме, автоматически формирует соответствующие выражения для описания ограничений проекта. Для этих целей может использоваться редактор назначения выводов кристалла и топологических ограничений *PACE (Pinout and Area Constraints Editor)*.

Процедура создания нового файла UCF стартует при нажатии кнопки  на оперативной панели или при выборе команды *New Source* из раздела *Project* основного меню Навигатора проекта. В открывшейся диалоговой панели *Select Source Type* следует выделить строку *Implementation Constraints File*, которая в качестве типа нового модуля задаёт файл временных и топологических ограничений проекта.

Затем в этой же панели необходимо указать название нового файла UCF в поле редактирования *File Name* и нажать кнопку *Next*. После этого на экран выводится следующая панель диалога с заголовком *Associate Source*, которая позволяет выбрать модуль исходного описания проекта, с которым ассоциируется создаваемый файл временных и топологических ограничений. Как правило, в большинстве случаев выбирается модуль описания верхнего уровня иерархии проекта.


Процесс выбора завершается нажатием кнопки *Next* в нижней части диалоговой панели *Associate Source*. После чего на экране появляется информационная панель *Summary*, в которой отображаются все параметры формируемого нового модуля временных и топологических ограничений.

Редактирование модуля временных и топологических ограничений проекта

Чтобы приступить к редактированию файла UCF, необходимо в окне исходных модулей Навигатора проекта выделить строку с его названием, после чего в окне процессов развернуть строку *User Constraints*.

Для изменения файла UCF во встроенном текстовом редакторе следует дважды щёлкнуть левой кнопкой мыши на строке *Edit Constraints (Text)*, в результате чего открывается новое рабочее окно редактирования. Ввод выражений временных и топологических ограничений осуществляется с помощью клавиатуры. Для сокращения времени подготовки файла временных и топологических ограничений, а также для минимизации син-

тактических ошибок рекомендуется использовать шаблоны HDL-редактора, представленные в папке UCF. В данной папке необходимо открыть один из разделов, CPLD или FPGA, в зависимости от семейства ПЛИС, выбранного для реализации проектируемого устройства. После этого в открытом разделе следует выбрать соответствующую группу шаблонов. Дальнейший механизм применения шаблонов был подробно рассмотрен в предыдущей статье данного цикла.

Чтобы вступили в силу все выполненные изменения в модуле временных и топологических ограничений (для учёта в процессах синтеза, размещения и трассировки проекта в кристалле), необходимо после редактирования сохранить файл UCF на диске, воспользовавшись командой *Save* из всплывающего меню *File* или кнопкой , расположенной на оперативной панели управления Навигатора проекта.

В качестве примера законченного модуля временных и топологических ограничений приведём текст файла UCF для проекта счётчика Джонсона, реализуемого на основе ПЛИС семейства XC9500XL в корпусе CS48.

```
NET left LOC=G7;
NET right LOC=B2;
NET stop LOC=F2;
NET clk LOC=A7;
NET q<0> LOC=B4;
NET q<1> LOC=C6;
NET q<2> LOC=F6;
NET q<3> LOC=G5;
NET clk PERIOD = 50ns;
```

Для модификации файла UCF с помощью редактора временных и топологических ограничений следует дважды щёлкнуть левой кнопкой мыши на строке *Create Timing Constraints*, в результате чего открывается рабочее окно программы *Constraints Editor*.

Чтобы воспользоваться средствами программы PACE для назначения «внешним» цепям проекта соответствующих контактов кристалла и ввода других топологических ограничений, необходимо расположить курсор мыши на строку *Assign Package Pins* и дважды щёлкнуть левой кнопкой. Указанная программа позволяет выполнить процедуру закрепления выводов в более наглядном виде по сравнению с текстовым редактором.

Продолжение следует