

Продолжение. Начало в № 3 '2009

Изучаем Active-HDL 7.1.

Урок 4. Как работать с редактором временных диаграмм

Александр ШАЛАГИНОВ
shalag@vt.cs.nstu.ru

Само название — редактор временных диаграмм (**Waveform Editor**) — может ввести в заблуждение. Из него следует, что данное инструментальное средство предназначено для создания и редактирования графических описаний сигналов, называемых внешними воздействиями. Как мы убедились на уроках 2 и 3, редактор действительно умеет выполнять такую работу. Но главное его назначение в другом — показывать и анализировать результаты моделирования в графическом формате.

Иными словами, в среде **Active-HDL 7.1** программа **Waveform Editor** объединяет в себе функции графического редактора внешних воздействий и постпроцессора моделирования. Во многих САПР названные функции выполняются разными программами. Например, в пакете **DesignLab 8.0** для этих целей используются два инструмента — **Stimulus Editor** (графический редактор входных сигналов) и **Probe** (постпроцессор моделирования).

Справедливости ради заметим, что в оригинальной документации фирмы **ALDEC** встречается и более точное название — **Waveform Viewer/Editor**. Слово **Viewer**, добавленное в название, как раз и подчеркивает вторую функцию — отображать (просматривать) результаты моделирования.

Итак, на самом деле редактор имеет два режима — просмотр результатов моделирования (**Waveform Viewer**) и редактирование временных диаграмм (**Waveform Editor**). Визуально эти режимы различаются цветом фона, на котором рисуются временные диаграммы.

В режиме просмотра результаты моделирования изображаются на желтом фоне, и пока окно редактора **Waveform Editor** присоединено к имитатору (команда **Connect to Simulator** из меню **Waveform**), а имитатор активен, невозможно вручную изменять графики временных диаграмм. Сказанное не касается самих стимулов, которые всегда доступны для редактирования: как перед началом, так и в процессе интерактивного моделирования. В режиме редактирования графиков цвет фона — белый, и появляется возмож-

ность изменять формы сигналов по своему желанию или рисовать их заново.

Окно редактора сигналов

Окно редактора **Waveform Editor** можно открыть многими способами. Пустое окно проще всего активизировать, щелкнув левой кнопкой мыши по иконке **New Waveform**, расположенной на инструментальной панели **Standard**. Именно так мы действовали на предыдущих уроках. Волновой файл с ранее сохраненными формами сигналов (*.awf) открывается двойным щелчком на его имени из окна **Design Browser**.

Загрузим в интегрированную среду **Active-HDL 7.1** проект **Lesson_1**, созданный на уроке 1 и модифицированный на уроке 3. Постараемся привести информацию, представленную в окне редактора сигналов **Waveform Editor**, к виду, изображенному на рис. 1. Заодно познакомимся более подробно с основными элементами этого окна.

Мы уже знаем (урок 1), что окно редактора **Waveform Editor** условно поделено на две части. В левой половине окна задается список наблюдаемых или редактируемых сигналов, а в правой — их временные диаграммы. Граница между ними определяется числом и шириной столбцов левой части. Отметим, что набор команд всплывающего меню тоже зависит от того, в какой части окна вы щелкнули правой кнопкой мыши.

По умолчанию в левой части размещаются три столбца:

- **Name** — имена объектов (портов, сигналов, переменных);
- **Value** — значение объекта для текущего положения активного курсора (он выделяется красным цветом);
- **Stimulator** — тип стимулятора, прикрепленного к объекту.

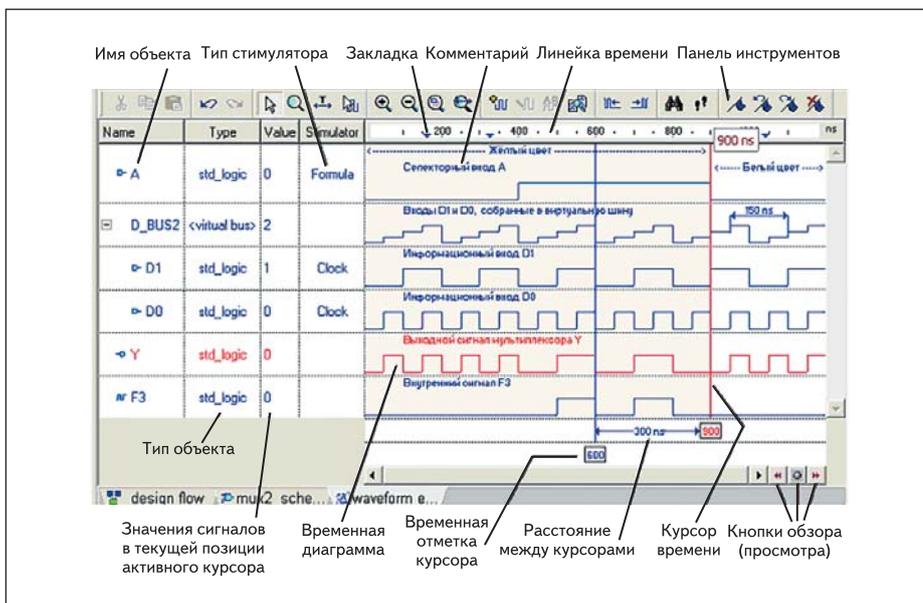


Рис. 1. Окно редактора сигналов Waveform Editor

По желанию пользователя к ним можно добавить еще три:

- **Mode** — графическое обозначение типа объекта (входные, выходные, двунаправленные порты, сигналы, переменные);
- **Hierarchy** — пути иерархии объекта;
- **Type** — тип объекта, например, для VHDL это **std_logic**, **bit**, **integer**, **boolean** и т. п.

Чтобы добавить или удалить столбец, надо щелкнуть правой кнопкой мыши на его имени, а затем выполнить команду **Columns** или **Remove This Column**.

Список сигналов можно сортировать в убывающем или возрастающем порядке по отдельным столбцам. В этом случае справа от имени сортируемого столбца появляется едва заметный треугольник (если хватает места), обращенный вершиной вверх или вниз в зависимости от порядка сортировки. Команды сортировки вызываются из контекстного меню или выполняются неявно при щелчке левой кнопкой мыши (ЛКМ) на имени сортируемого столбца.

Если сортировка не дает нужного результата, можно вручную (методом **Drag & Drop**) расставить графики так, как вы этого хотите. Перед тем, как заняться такой работой, рекомендуется свернуть все виртуальные шины. В любом случае входящие в них сигналы не следует «отрывать» от самой шины и без необходимости «перетасовывать» сигналы в ней. Помните, ручная сортировка возможна лишь в том случае, если выключена автоматическая сортировка, то есть ни в одном столбце нет треугольника автоматической сортировки.

Нередко пользователь минимизирует ширину столбцов, используя команду **Best Fill** (наилучшее заполнение) из всплывающего меню, или вручную сокращает ширину столбца до размеров имени. Эти действия приводят к весьма неприятным последствиям: индикатор автоматической сортировки становится невидимым, и пользователь начинает удивляться, почему вдруг перестала работать ручная сортировка.

Каждый тип объектов (портов, сигналов и переменных) имеет свое графическое обозначение. Например, для объектов, написанных на языке VHDL, они будут выглядеть так, как показано на рис. 2. В принципе эта информация дублируется в двух столбцах: **Mode** и **Name**. Поэтому, если установлен флажок **Show Signal Icons** столбца **Name**.

- ▷ входной порт — in
- ◁ выходной порт — out
- ↔ двунаправленный порт — inout
- лг сигнал
- V= переменная

Рис. 2. Значки, указывающие тип VHDL-объекта (порты, сигналы, переменные)

Способы формирования списка сигналов

Существует три способа формирования списка сигналов, которые мы хотим контролировать в процессе моделирования. О двух из них уже говорилось на предыдущих уроках. Коротко напомним их суть.

Щелчком в левой части окна редактора **Waveform Editor** надо вызвать контекстное меню (урок 1) и исполнить команду **Add signals** (горячая клавиша **Ctrl+I**). Появится полный список объектов верхнего уровня иерархии (**Root**), из которого остается только выбрать необходимые в данном эксперименте сигналы и перенести их в окно волнового редактора. Чтобы получить доступ к сигналам другого иерархического уровня, следует выделить нужный уровень и указать курсором требуемый сигнал.

Для тех, кто предпочитает вводить команды с клавиатуры, существует другой способ (урок 3). Он заключается в использовании макрокоманды **wave**, за которой надо ввести список наблюдаемых сигналов. Текстовая строка с макрокомандой вводится с клавиатуры в окно **Console**, например:

```
wave D0 D1 A Y
```

Третий способ формирования списка контролируемых сигналов заключается в использовании инструмента **Design Browser**. Нам понадобится его средняя закладка **Structure**. Она автоматически активизируется, если исполнить команду **Initialize Simulation** из меню **Simulation**.

С помощью клавиш **Shift** или **Ctrl** следует выделить требуемые сигналы и отбуксировать их методом **Drag & Drop** в окно волнового редактора. Если вам нужны все сигналы, то предпочтительнее команда **Select All** или горячая клавиша **Ctrl+A**.

Вероятно, самый быстрый способ достичь желаемого результата заключается в использовании команды **Add to Waveform** из всплывающего меню. При этом не потребуется даже предварительно открывать окно редактора **Waveform Editor** — все будет сделано автоматически.

В ходе модельных экспериментов нередко приходится изменять первоначально сформированный список: удалять одни и добавлять другие сигналы. Конечно, можно воспользоваться командой **Delete**. Но если нет уверенности, что удаляемый сигнал больше не понадобится, то лучше его просто сделать невидимым (команда **Hide/Show** из всплывающего меню).

Но как снова сделать видимым ранее скрытый сигнал? Второе слово в названии команды **Hide/Show** говорит о том, что видимость сигнала можно восстановить. Но ведь прежде его надо увидеть. Кажется, мы оказались в «патовой ситуации». И тем не менее выход

есть. Воспользуйтесь командой **Show Hidden Signals** из выпадающего меню **Waveform**, и все скрытые сигналы появятся на экране монитора.

В пакете **Active-HDL 7.1** есть еще одно более удобное средство, предназначенное для тех же целей, — это диалоговая панель **Signal Browser** (рис. 3). Она открывается одноименной командой из выпадающего меню **Waveform**.

В левом окне панели приводится иерархия проекта, а в правом — список сигналов выбранного уровня, загруженный в окно волнового редактора. Устанавливая (**Show**) или сбрасывая (**Hidden**) флажки рядом с именами сигналов, можно определять оптимальный для текущего эксперимента набор объектов.

Заканчивая теоретическую разминку, скажем несколько слов о виртуальных шинах. В нашем проекте **Lesson_1** реальных шин нет. Но для удобства работы мы можем объединять любые существующие сигналы в виртуальные (не существующие на самом деле) шины. Сказанное касается не только входных, но и выходных сигналов.

Для этого надо выделить объединяемые в шину объекты и выполнить команду **Merge Signals** из контекстного меню. Противоположная команда **Split Signals** вернет все на свои места.

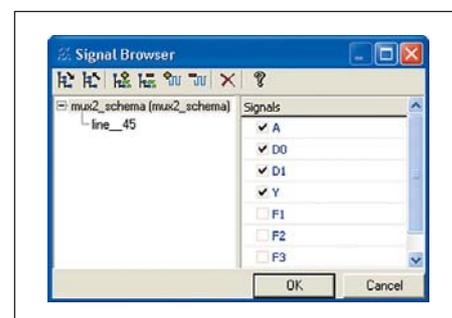


Рис. 3. Диалоговая панель просмотра сигналов Signal Browser, загруженных в сигнальный редактор Waveform Editor

Содержимое созданных шин можно редактировать, например, переставлять местами сигналы, добавлять или удалять их из уже существующей шины. Заметим, что названные операции доступны не только перед началом, но и в ходе модельного эксперимента.

По умолчанию созданные шины будут получать стандартные имена **VBUS0**, **VBUS1**, **VBUS2** и т. д. Однако их всегда можно переименовать. С этой целью надо выделить виртуальную шину и щелкнуть мышью на ее имени, теперь оно доступно для редактирования. Именно так было заменено стандартное название шины **VBUS0** на пользовательское имя **D_BUS2** (рис. 1).

Как следует из рис. 1, шина **D_BUS2** содержит два сигнала — **D1** и **D0**. Чтобы убедиться в этом, достаточно свернуть и снова распаковать шину. Но и без дополнительного экс-

перимента легко заметить, что сигналы, составляющие шину, изображаются иначе — их значки смещены вправо.

Порядок следования объектов в шине такой же, как и в первоначальном списке, из которого они выбирались. Например, если вы хотите, чтобы сигнал **D1** оказался «старшим» разрядом шины **D_BUS2**, то надо позаботиться о том, чтобы он был выше других в названном списке.

А теперь приступим к делу. Закроем окно волнового редактора с загруженным в него файлом **mux2_schema.awf**. Исполним команду **Initialize Simulation** из меню **Simulation**. В окне просмотра проекта **Design Browse** активизируется средняя закладка **Structure**.

Откроем список сигналов верхнего уровня проекта. Для этого надо щелкнуть ЛКМ на строке **mux2_schema (mux2_schema)** в верхнем окне **Design Browser**. В нижнем окне появится полный список объектов верхнего уровня иерархии проекта (**top level**).

Выделим их целиком (горячая клавиша **Ctrl+A**) и исполним команду **Add to Waveform** из всплывающего меню. Откроется окно редактора **Waveform Editor**, и все выделенные сигналы будут автоматически перенесены в него.

Вызовем диалоговую панель **Signal Browser** (команда с тем же именем из меню **Waveform**) и сделаем невидимыми сигналы **F1** и **F2**. Перетащим сигнал **D1** так, чтобы он оказался выше сигнала **D0**, и объединим их в виртуальную шину (команда **Merge Signals** из всплывающего меню). По умолчанию редактор присвоит ей имя **VBUS0**.

Зададим шине другое имя — **D_BUS2**. Щелкнем на значке «плюс» рядом с ее именем, чтобы увидеть входящие в нее объекты. На сигналы **A**, **D1** и **D0** «повесим» стимуляторы и запрограммируем их так же, как мы это делали на первом уроке.

Выполним моделирование на интервале 1200 ns (команда **run 1200 ns** в окне **Console**). Вернемся к началу (команда **restart**) и повторим моделирование на интервале 900 ns. Новое моделирование (желтый фон) выполняется поверх результатов предыдущего модельного эксперимента (белый фон).

Задание комментария к временным диаграммам

Добавим комментарий к созданным временным диаграммам. Комментарий особенно полезен, если предполагается знакомство с вашим проектом других лиц. В любом случае он улучшает восприятие результатов моделирования. Кроме того, по тексту комментария может выполняться поиск нужных участков временных диаграмм.

Выделим сигнал, для которого мы собираемся задавать пояснения, и поставим курсор в то место, где должен располагаться комментарий. Найдем в контекстном меню команду **Insert Comment** и исполним ее.

В качестве альтернативы можно щелкнуть по иконке с тем же названием . Введите с клавиатуры текстовую строку комментария и нажмите **ENTER**. При необходимости уже существующий комментарий всегда можно отредактировать. Аналогичным образом добавьте пояснения к остальным временным диаграммам в соответствии с рис. 1.

Редактирование свойств сигналов

Нередко возникает необходимость выделить какую-либо временную диаграмму, чтобы она сразу «бросалась в глаза». Это означает, что она должна иметь иные свойства, чем у остальных объектов. На рис. 1 таким объектом является выходной сигнал **Y**. Щелкнем по нему ПКМ и выполним команду **Properties** («Свойства»).

На открывшейся диалоговой панели активизируем закладку **Display** и поменяем заданный по умолчанию синий цвет графика на красный. Кроме цвета можно изменить размеры, толщину графика и даже его форму.

Ради эксперимента поменяем форму у виртуальной шины **D_BUS2**, активизировав кнопку выбора  **Analog** на закладке **Display** диалоговой панели «Свойства». Теперь ее значения выглядят как ступенчато нарастающая функция (рис. 1).

Для завершения работы осталось измерить два временных интервала, показанных на рис. 1, и поставить на измерительной линейке несколько закладок (небольших треугольников синего цвета).

С закладками вообще все просто. Достаточно щелкнуть ЛКМ в точке на временной линейке, где вы хотите поместить закладку. Повторный щелчок на закладке удалит ее.

Можно поставить любое число закладок. Позднее мы узнаем, что, опираясь на закладки, очень удобно просматривать длинные временные диаграммы и быстро находить нужные ее участки.

Измерять временные интервалы мы научились на втором уроке, и сейчас вам представилась возможность самостоятельно выполнить эти операции.

Режимы просмотра временных диаграмм

При сложных схемах результаты моделирования не помещаются на экране монитора. Не оправдывает себя и надежда на стандартные средства **Windows** — полосы прокрутки. Очень трудно и неудобно «искать иголку в стоге сена».

Когда число сигналов велико, и они не помещаются на один экран, полезно открыть несколько окон редактора **Waveform Editor**. Самое ценное заключается в том, что во всех окнах автоматически обеспечивается *синхронизация* процесса моделирования.

Для удобства работы в последних версиях пакета **Active-HDL** появилась возможность

добавлять один и тот же сигнал не только в разные окна, но и в одно окно.

Если пришла идея сэкономить место за счет высоты (**Height**) временных диаграмм, то вас ждет разочарование — по умолчанию в свойствах объектов уже установлено ее минимальное значение: **Height = 15**. Конечно, можно увеличить число видимых на экране графиков, если удалить комментарий к ним. Но, как говорится, это «палка о двух концах». В любом случае решение остается за вами.

Когда велико время моделирования, то можно попытаться «сжать» временные диаграммы, используя операции масштабирования (рис. 4).

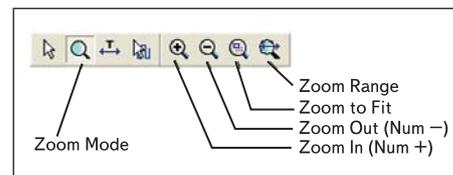


Рис. 4. Команды масштабирования временных диаграмм

Самая радикальная команда **Zoom to Fit** («Покажи все») может не дать желаемого результата, если эксперимент слишком длинный. Большое количество переключений какого-либо сигнала сольется в сплошную полосу, и график потеряет информативность. Однако в любом случае эта команда полезна, чтобы сделать осмысленными последующие действия.

Например, можно переключиться в режим **Zoom Mode** (иконка ) и при нажатой ЛКМ выделить желаемую область наблюдения. В момент отпускания кнопки указанная область будет распахнута на все окно редактора.

Очень удобно масштабировать графики с помощью клавиш **Num+** и **Num-**. Они расположены на клавиатуре у самого края, и до них удобно добираться правой рукой. Немаловажно, что названные клавиши сохраняют активность в любых режимах работы волнового редактора.

Следует также помнить, что изменение масштаба выполняется относительно активного курсора (отмеченного красным цветом). Другими словами, этот курсор при любом увеличении остается виден в окне редактора, сохраняя свое текущее местоположение.

Кроме операций масштабирования, в среде **Active-HDL** есть весьма эффективный инструмент, позволяющий быстро, а главное осмысленно, перемещаться по временным диаграммам. Такая работа выполняется с помо-

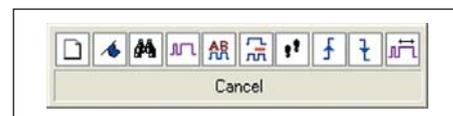


Рис. 5. Панель выбора режимов просмотра временных диаграмм

щью кнопок обзора (просмотра), расположенных в правом нижнем углу волнового редактора . Щелкнув на средней кнопке, вы увидите возможные режимы просмотра (рис. 5).

Вы можете «пролистывать» временные диаграммы по страницам (экранам) , продвигаться по оси времени скачками от одной закладки к другой , просматривать результаты моделирования, «опираясь» на события , комментарии , метки несовпадения , фронты или срезы сигналов, заданные ограничения . Двигаться можно в любом направлении, используя кнопки **Browse Next** (следующий просмотр) или **Browse Previous** (предыдущий просмотр) .

Обратите внимание на цвет кнопок перемещения. Он зависит от выбранного режима, например, при движении «по событиям» цвет кнопок коричневый , а при перемещении по закладкам — синий .

Некоторые пользователи предпочитают применять для этих целей не кнопки перемещения, а управляющие клавиши «стрелка влево» и «стрелка вправо», которые становятся активными при нажатой клавише **Ctrl**.

Кроме того, движение по событиям возможно с помощью иконок **Previous Event** и **Next Event** , расположенных на инструментальной панели **Waveform Toolbar**.

В реальных цифровых схемах всегда существуют ограничения на определенные временные параметры, такие как время предустановки или удержания, минимальная длительность сигнала сброса, сдвига или счета и т. п.

Операции, связанные с их проверкой, называются контролем временных соотношений в схеме. Моделированием такие проверки выполняются особенно эффективно.

Допустим, в нашем проекте выходной сигнал **Y** (точнее, его единичный уровень) не должен быть короче 60 ns (заметьте, что это совершенно абстрактное требование). Щелкнем мышью по кнопке обзора и выберем режим **Browse by constraint** («Просмотр ограничений»). Появится диалоговая панель с таким же названием, где мы зададим контролируемое ограничение (рис. 6).

Выделим контролируемый сигнал **Y**, поставим курсор на начало координат (иконка **Go to Time** или горячая клавиша **Ctrl+G**) и, щелкая по кнопке просмотра **Next constraint** , начнем искать участки временных диаграмм, где единичный уровень сигнала **Y** оказывается короче 60 ns.

Мы можем предсказать ожидаемый результат: пока на выход **Y** передается сигнал

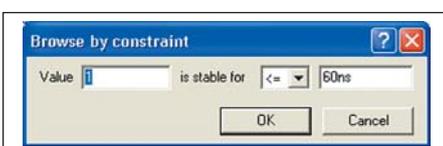


Рис. 6. Установка контролируемого ограничения (уровень '1' выходного сигнала **Y** не должен быть меньше 60 ns)

D0 частотой 10 МГц (длительность положительных импульсов 50 ns), волновой редактор будет фиксировать нарушение временных интервалов. Как только на выход начнет передаваться сигнал **D1** частотой 5 МГц (длительность импульсов 100 ns), редактор перестанет находить нарушения. Описанный режим очень удобно использовать и в процессе поиска коротких импульсных помех, называемых рисками сбоя или иголками (**Glitches**).

Среди рассмотренных режимов просмотра особое место занимает режим поиска сигналов по значению или комментариев по заданному тексту. Активизируется этот режим щелчком по иконке **Find** — третья слева на рис. 5. Надо сказать, что это далеко не единственный способ запустить данный режим. Например, команда **Find** «висит» и на горячей клавише **Ctrl+F**.

Командой **Find** открывается диалоговая панель с тем же названием, где можно задать условия поиска (рис. 7).

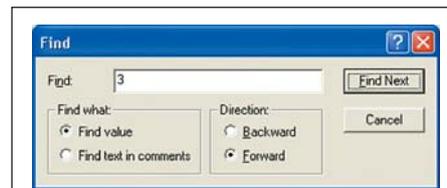


Рис. 7. Задание условий поиска — значение 3 в прямом направлении

В данном случае мы собираемся искать значение 3 в прямом направлении (**Forward**). Такое значение иногда принимает виртуальная шина **D_BUS2**. Если не выделен ни один сигнал, то поиск будет осуществляться по всем временным диаграммам, если выделен один или несколько объектов, то — только в заданном подмножестве. Поставив в поле **Find** значение **X**, мы с минимальными усилиями про-

верим, нет ли в нашей схеме некорректного поведения и неизвестных **X**-состояний.

С помощью описанного способа трудно, а порой и невозможно отыскать конкретное состояние схемы, особенно когда оно повторяется неоднократно. А вот поставив в качестве метки уникальный комментарий, мы одним щелчком мыши попадем в требуемую точку на временных диаграммах. Например, введя в поле **Find** шаблон для поиска «белый», редактор сразу поместит курсор в точку 900 ns (рис. 1), где вставлен комментарий «Белый цвет».

Создание виртуальных сигналов с помощью логических выражений

Повысить «разрешающую способность поиска» можно с помощью логических выражений, создаваемых пользователем. В таких выражениях разрешается использовать только реальные сигналы или созданные ранее виртуальные шины. Объединять их в математические выражения можно с помощью различных логических операций (**not, and, nand, or, nor, xor, xnor**), арифметических операций (+, -), операций сдвига (>>, <<) и операций отношения (=, /=, <, >, <=, >=). При этом разрешаются скобочные формы записи.

Например, вы хотите отыскать участки временных диаграмм, на которых все входные сигналы (то есть **A**, **D1** и **D0**) принимают единичные значения. Выделим названные сигналы, щелкнув ПКМ в левой части окна волнового редактора и исполним команду **Create Expression Signal**. Откроется диалоговая панель с тем же названием (рис. 8) и с уже готовым булевским выражением. Дело в том, что по умолчанию все выбранные сигналы редактор объединяет операциями конъюнкции **and**.

Щелкнув на кнопке **OK**, вы обнаружите это выражение в списке сигналов волнового редактора. Более того, вам не придется повторять моделирование — временные диаграм-

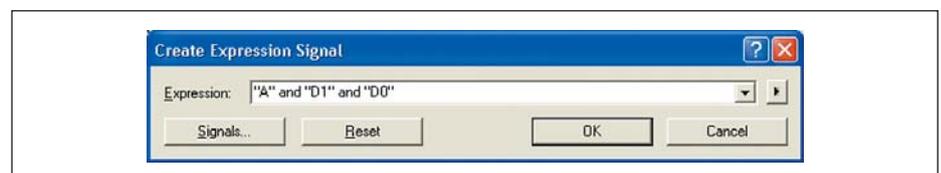


Рис. 8. Автоматическая генерация логического выражения

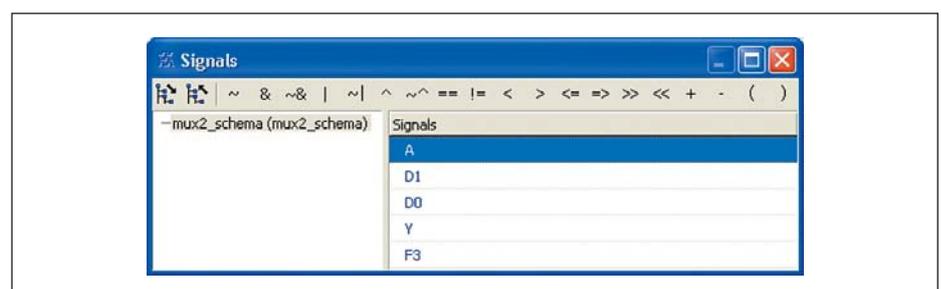


Рис. 9. Список сигналов и операций (строка сверху), доступных для формирования математических выражений

мы для полученного выражения будут построены автоматически.

Этот же результат можно получить другим способом, написав, например, такое выражение: («A»=1) and («D_BUS2» = 3). Щелкнув на кнопке **Signals**, вы увидите все доступные сигналы (рис. 9), которые можно включить в создаваемое математическое выражение.

К сожалению, виртуальные шины в этом списке отсутствуют, хотя они тоже доступны для использования.

Вы можете записать нужное выражение вручную, вводя его в поле **Expression** (рис. 8) с клавиатуры. Однако эффективнее это делать с помощью мыши. Двойной щелчок на имени сигнала «перекидывает» его в формируемое выражение, а одиночным щелчком туда же добавляется значок выбранной операции.

Рассмотрим еще один пример использования выражений, но теперь уже для функциональной верификации проекта. На уроке 1 мы спроектировали схему мультиплексора **mux2_schema** и сейчас попробуем выяснить, правильно ли он работает.

Известно, что такой мультиплексор должен функционировать в соответствии с булевым выражением $Y = (\text{not } A \text{ and } D0) \text{ or } (A \text{ and } D1)$. С помощью только что описанного механизма введем это выражение (точнее, его правую часть) в окно волнового редактора и сравним с выходным сигналом Y. Понятно, что для правильно спроектированной схемы они должны совпадать.

Выделим оба сравниваемых графика, щелкнем ПКМ и исполним команду **Compare Selected** («Сравнить выделенные сигналы»). Описанные действия вызовут диалоговую панель (рис. 10), на которой требуется уточнить условия выполняемой операции сравнения.

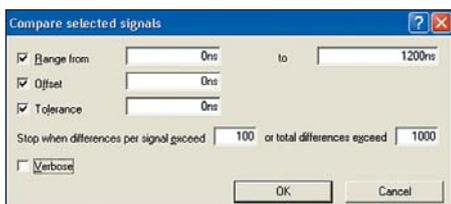


Рис. 10. Задание условий для сравнения выделенных сигналов

На рис. 10 видно, что мы указали временной диапазон, на котором будут сравниваться два сигнала (**Range from** 0 ns to 1200 ns), смещение одного графика относительно другого (**Offset** 0 ns) и допуск, в пределах которого различия между сигналами не будут приниматься во внимание (**Tolerance** 0 ns).

В нашем примере равенство будет фиксироваться только при полном совпадении графиков (смещение и допуск равняются нулю), что мы и наблюдаем в окне **Console** (рис. 11). Фраза «0 difference(s) found» говорит о том, что различий в сравниваемых графиках не обнаружено.

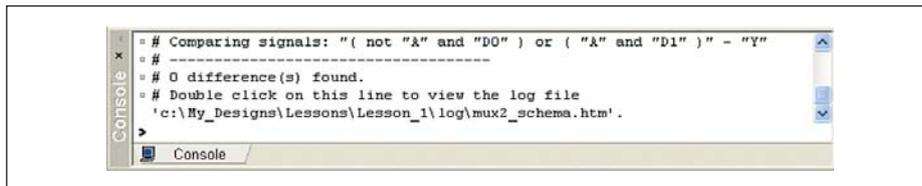


Рис. 11. Сообщение о результатах сравнения выведено в окно Console (различий между реальным и эталонным сигналами не обнаружено)

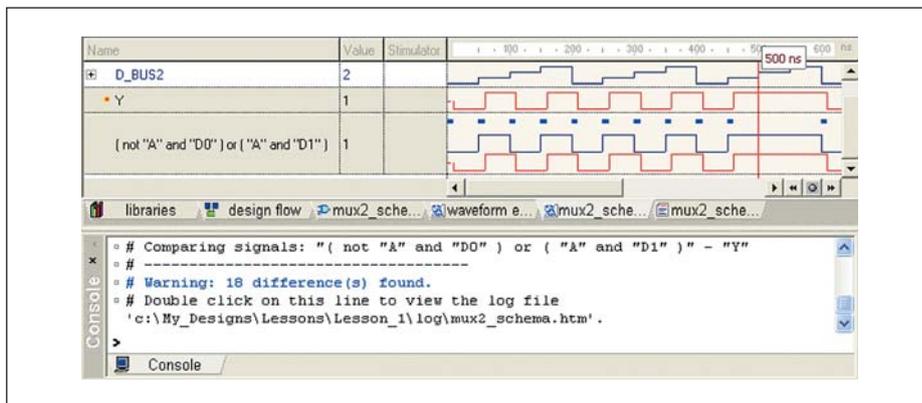


Рис. 12. Результаты сравнения реального выходного сигнала Y мультиплексора mux2_schema и эталонного выражения, описывающего его функционирование

Вряд ли вы получили удовлетворение от проведенного эксперимента. Конечно, он подтвердил правильность ранее выполненного проекта, и это радует. Но хотелось бы увидеть, каким образом в пакете фиксируются отличия.

Придется еще раз повторить эксперимент, добавив в проект временную задержку, с которой мультиплексор реагирует на входные воздействия.

Найдем в окне **Design Browser** файл **mux2_schema.vhd** с VHDL: описанием схемы мультиплексора и двойным щелчком ЛКМ отроем его.

Предпоследнюю строку в файле отредактируем так, чтобы назначение для выхода Y выполнялось с задержкой 10 ns:

```
Y <= F3 or F2 after 10 ns;
```

Фраза “after 10 ns” как раз и означает, что выход Y получит новое значение не в тот момент, когда выполняется данный оператор, а спустя 10 ns относительно текущего модельного времени.

Чтобы появилась возможность редактировать этот файл, надо снять с него флажок «Только чтение».

Сохраним отредактированный файл. Измененный файл откомпилируем еще раз и повторим модельный эксперимент (рис. 12).

Участки временных диаграмм, на которых сравниваемые сигналы не совпадают, волновой редактор отмечает жирными линиями синего цвета. Причина их появления очевидна — это влияние задержек, внесенных нами в проект.

Для сложных цифровых устройств наличие задержек затрудняет оценку правильности их функционирования. Например, два варианта схемной реализации проекта могут работать одинаково, но иметь разные задержки. Из-за них не будет совпадения реакций, а значит, и уверенности в том, что обе реализации правильны.

Чтобы получить адекватные результаты, следует выполнить сравнение с «точностью до задержек». Другими словами, надо разрешить сигналам немного отличаться друг от друга так, чтобы система этого не чувствовала. В среде **Active-HDL 7.1** предусмотрена такая возможность.

В поле **Tolerance** на панели **Compare Selected Signals** (рис. 10) можно задать допуск на разрешенные отличия.

Введите в это поле значение 10 ns и повторите моделирование. Теперь отличия сравниваемых графиков длительностью 10 ns и менее волновой редактор не регистрирует. Такого же результата можно достичь, просто сдвинув (**Offset**) эталонный сигнал на величину задержки (в примере на 10 ns).

Ограничение чувствительности сравниваемых сигналов позволяет также исключить из этого процесса и риски сбоя — импульсные помехи малой длительности (**glitch** — иголки).

Более подробный отчет о результатах сравнения можно получить, если дважды щелкнуть в окне **Console** на строке, начинающейся словами: **Double click on this line to view the log file...**

Продолжение следует