

Изучаем Active-HDL 7.1.

Урок 1. Знакомство с пакетом

Александр ШАЛАГИНОВ
shalag@vt.cs.nstu.ru

Не правда ли, до боли знакомая «картинка»: истерзанная бокорезами, скальпелем и паяльником печатная плата, хранящая тайны ненайденных ошибок. Вокруг нагромождение перепутанных проводами железных ящиков — блоков питания, генераторов и осциллографов. А рядом ваш верный друг — дымящийся паяльник.

Что это? Ностальгия разработчика аппаратуры старой закалки по «давно минувшим дням»?

Ныне другие времена. Теперь предмет своего творчества вы можете получить, в буквальном смысле «не отходя от компьютера». Да и технология создания аппаратуры выглядит совсем по-другому, напоминая больше стиль работы программиста, чем схемотехника.

Чем же должен владеть современный разработчик? По крайней мере, двумя инструментами — языком описания аппаратуры (типа Verilog или VHDL) и средой компьютерного проектирования (типа Active-HDL, Xilinx ISE или Quartus II).

Что же касается типового маршрута проектирования, то он примерно такой. На языке описания аппаратуры (HDL), похожем на язык программирования, вы описываете поведение или структуру разрабатываемого устройства, моделированием избавляетесь от ошибок в нем, после чего синтезируете и реализуете свой проект. Но не на печатной плате, а на кристалле микросхемы программируемой логики (ПЛИС).

Остается только запрограммировать ее, получив физическую реализацию своего проекта. Заметьте, у вас в руках не виртуальное, а осязаемое, существующее «в железе» устройство. И, что немаловажно, результат порой достигается без всякого паяльника, нужна лишь подходящая отладочная плата.

Предлагаемые «Уроки», по мнению автора, должны помочь освоить, пожалуй, самый популярный среди разработчиков цифровой аппаратуры продукт — интегрированную среду проектирования Active-HDL 7.1. Работать в ней — одно удовольствие, в чем, как мы надеемся, вы убедитесь уже после первого ознакомительного урока. Вам будет предложено, по меньшей мере, 15 уроков, причем каждый из них заканчивается практическим результатом. Согласитесь, промежуточный успех, пусть и самый маленький, служит хорошим стимулом к тому, чтобы сделать следующий шаг. Ну что, начнем?

Введение

Автоматизированное проектирование — это магистральное направление развития тех-

нологий создания электронной аппаратуры. До недавнего времени образцом решения таких задач считались разработки на базе серийно выпускаемых стандартных интегральных микросхем, и большинство САПР ориентировались на реализацию именно этой технологии проектирования.

Идеальным вариантом казались «сквозные» САПР, которые позволяли выполнить всю цепочку проектирования: от входного описания до создания эскиза печатной платы. Примерами сквозных САПР являются PCAD, DesignLab и OrCAD.

В последние годы набирают силу новые технологии проектирования электронной аппаратуры, основанные на современных персональных компьютерах, интегрированных САПР и программируемых микросхемах (ПЛИС).

Яркий представитель таких САПР нового поколения — это интегрированная среда проектирования Active-HDL 7.1. Ее отличает особая легкость и, если можно так выразиться, элегантность в работе. Фирма ALDEC, создатель этого программного продукта, год от года совершенствует свое детище и наделяет его новыми возможностями. Предлагаемая версия данного пакета по многочисленным опросам пользователей и специалистов занимает лидирующие позиции в мире.

К сожалению, серьезным препятствием на пути использования названной системы в образовательных, проектных и коммерческих организациях России является дефицит литературы по этому пакету на русском языке.

Цель предлагаемых «Уроков» — хотя бы немного улучшить сложившуюся ситуацию, побыстрее и без моральных потрясений начать изучение новой и перспективной САПР — интегрированной среды проектирования цифровой аппаратуры Active-HDL 7.1.

Чтобы запустить управляющую оболочку пакета, достаточно щелкнуть на пиктограмме. При установке данного программного продукта эта иконка автоматически помещается на рабочем столе вашего компьютера.

На стартовой панели **Getting Started** (рис. 1) вы можете выбрать из предложенного списка и открыть существующее рабочее пространство (**Open existing workspace**) или создать новое (**Create new workspace**). Мы начнем знакомство с созданием нового рабочего пространства, а потому установим переключатель в положение **Create new workspace**.

Понятие «рабочее пространство» является относительно новым, к нему мы еще вернемся, а пока удовлетворимся простым пояснением. Это еще одна надстройка над совокупностью обычных проектов, своего рода суперпроект или мультипроектная рабочая среда. Вы можете «держать под рукой» сразу несколько проектов и иметь быстрый доступ к их совокупному ресурсу.

На второй панели **New Workspace** (рис. 2) необходимо напечатать имя рабочего прост-

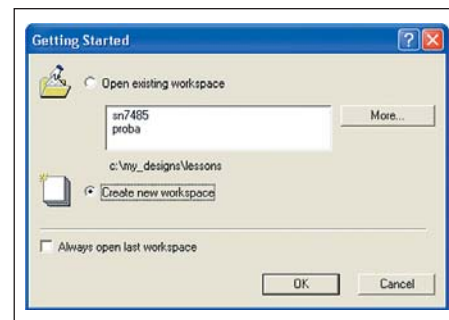


Рис. 1. На стартовой панели устанавливаем переключатель в положение Create new workspace («Создать новое рабочее пространство»)

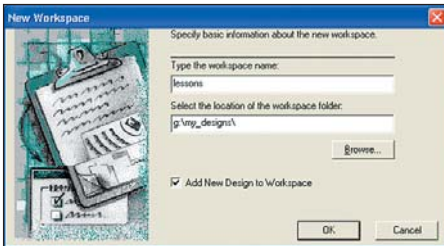


Рис. 2. Задаем имя рабочего пространства и его месторасположение на диске

ранства и указать место, где оно будет расположено. По умолчанию вам будет предложена папка **my_designs**, находящаяся в корне того логического диска, на котором установлена система.

Чтобы не делать потом лишнюю работу, убедитесь, что установлен флажок **Add New Design to Workspace**.

На третьей панели **New Design Wizard** (рис. 3) по умолчанию предлагается режим **Create an Empty Design with Design Flow** («Создать пустой проект с использованием менеджера маршрута проектирования»). Это наиболее разумный вариант, в чем мы позднее убедимся.

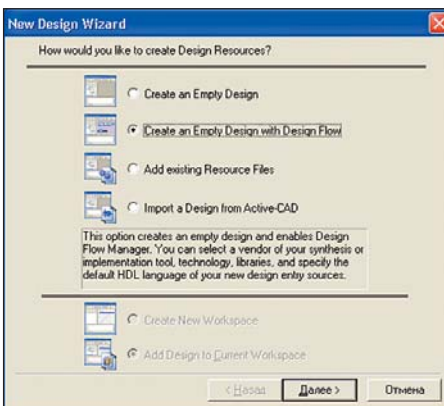


Рис. 3. Выбираем вариант создания проекта с помощью менеджера маршрута проектирования Design Flow Manager

На четвертой панели сохраняем все, как есть (установки по умолчанию), потому что синтез и реализация проекта не поддерживаются данным пакетом, если на вашем компьютере не установлены дополнительные программные средства. Во всяком случае, обсуждение этих вопросов не входит в планы первоначального знакомства с системой **Active-HDL 7.1**.

На пятой панели введем имя проекта, например **Lesson_1** (урок_1). Обратите внимание, синхронно с вводом имени проекта заполняется и нижнее поле, где необходимо задать имя рабочей библиотеки. По умолчанию предполагается, что они будут иметь одинаковые имена, что представляется весьма разумным.

На последней панели мастера **New Design Wizard** вы должны проверить правильность введенных данных и, если нет нужды возвра-

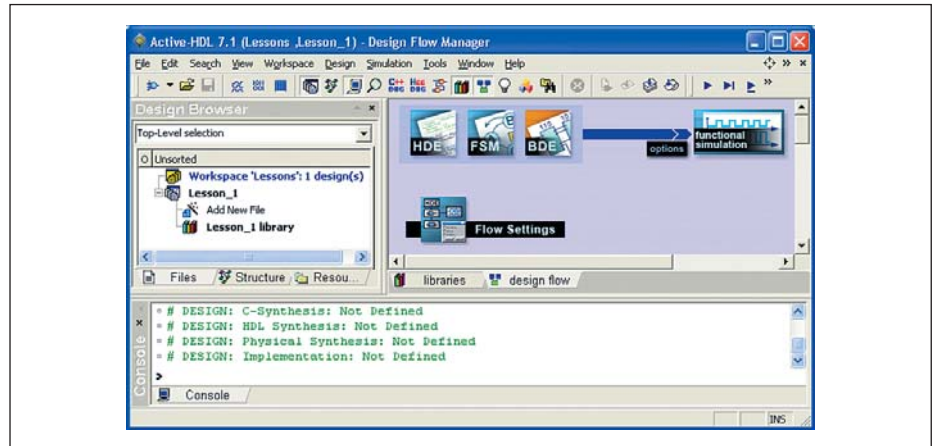



Рис. 4. Управляющая оболочка (Framework) системы проектирования Active-HDL 7.1

щаться к уже пройденным панелям, нажать на кнопку «Готово».

С этого момента собственно и начинается активная работа по созданию проекта. Она выполняется в интегрированной среде проектирования **Active-HDL 7.1** (рис. 4). В строке заголовка видно это название, а правее в круглых скобках высвечивается имя загруженного рабочего пространства и проекта: **Lessons, Lesson_1**. Далее следует название активной закладки — **Design Flow Manager**.

Ввод проекта

В начале проектирования на экране монитора открыты всего три окна интегрированной среды **Active-HDL 7.1**:

- **Design Browser** — окно просмотра проекта, в котором отображается все содержимое рабочего пространства **Lessons**. Пока в рабочем пространстве только один проект **Lesson_1** с пустой рабочей библиотекой **Lesson_1 library**. Горячая клавиша **Alt+I** включает и выключает это окно.
- **Console** — окно для интерактивного ввода и/или вывода текстовой информации. Все инструментальные средства пакета выдают в это окно сообщения о своей работе, в том числе предупреждения и ошибки. Поэтому надо взять себе за правило регулярно сюда заглядывать или закрывает это окно). Более того, вы можете сами вводить в окно **Console** различные команды, подобно тому, как это делается с помощью командной строки **MS-DOS**.
- В третьем окне активна закладка **Design Flow Manager** («Менеджер маршрута проектирования»). Обычно данное окно занимает большую часть экрана потому, что именно здесь пользователь создает основные документы проекта. Вы можете развернуть окно на весь экран, нажав горячую клавишу **Alt+F9** или щелкнув на значке , расположенном в правом верхнем углу экрана. На закладке **Design Flow Manager** изображен доступный для заданных установок мар-

шрут проектирования. Он ограничивает наши действия тремя способами описания проекта (**HDE, FSM, BDE**) и последующим функциональным моделированием (**functional simulation**). Правда, перед моделированием потребуется еще установка некоторых опций (**options**).

Если мы собираемся создавать проект в текстовом формате на языке **VHDL, Verilog** или **SystemC**, то следует нажать кнопку **HDE**. В качестве рабочего инструмента будет вызван текстовый редактор **HDE** (от слов **Hardware Description language Editor**).

Если мы хотим представить описание проекта в виде диаграммы состояний конечного автомата (**Finite State Machine**), то следует щелкнуть на кнопке **FSM**. Однако мы собираемся рисовать схему, а потому нажмем на кнопку **BDE** и вызовем схемный редактор — **Block Diagram Editor**. Заметим, что это далеко не единственный способ начать работу. Итак, щелкнув на кнопке **BDE**, мы вызовем «мастер» создания нового исходного файла **New Source File Wizard**.

На первой диалоговой панели «мастера» нам делать нечего. Надо просто убедиться, что включен флажок **Add the generated file to the design** — «Добавить генерируемый файл к проекту» (а по умолчанию он установлен), и перейти на следующую панель.

На второй панели предлагается выбрать язык описания аппаратуры, на который позднее будет конвертирована наша схема. По умолчанию — это язык **VHDL**. Нас он вполне устраивает, поэтому оставляем все, как есть, и продвигаемся на третью панель. Здесь необходимо ввести имя исходного файла, например **mux2_schema** (двухканальный мультиплексор), где система будет хранить схемное описание проекта. По умолчанию файл получит расширение ***.bde**.

Имена объекта проекта (**entity**) и его архитектурного тела (**architecture body**) могут совпадать с названием файла, поэтому второе и третье поля заполнять не обязательно.

На четвертой панели «мастер» предлагает ввести интерфейсные порты, соответствующе-

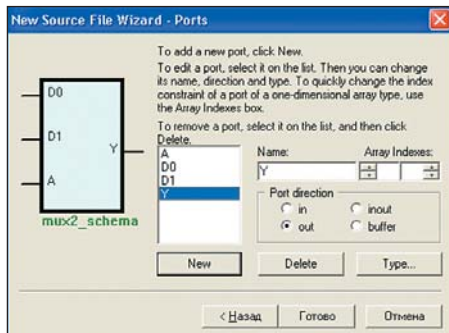


Рис. 5. Вводим данные об интерфейсных портах проекта

щие входам и выходам проектируемого устройства. Мультиплексор на два входа, который мы собираемся проектировать, имеет два информационных входа **D0** и **D1**, селекторный вход **A** и выход **Y**. Нажимая на кнопку **New** или клавишу **Enter**, введем описание портов, как показано на рис. 5.

Кроме имени порта необходимо указать его направление (**in** — входной, **out** — выходной, **inout** — двунаправленный) и тип (**bit**, **boolean**, **integer**, **std_logic** и т. д.). По умолчанию тип порта установлен как **std_logic**, что соответствует нашим желаниям. Щелкните по кнопке **Type**, и вы убедитесь в сказанном.

Нажатие на кнопку «Готово» заканчивает ввод портов, и управление передается схемному редактору **Block Diagram Editor**.

В главном окне среды **Active-HDL 7.1** появляется новая закладка с именем открытого в этом редакторе файла **mux2_schema.bde**. Мы уже знаем, что это схемный файл. Обратите внимание, он добавлен и в окно просмотра проекта **Design Browser** (рис. 6). Такие файлы помечаются специальным значком с изображением символа



Рис. 6. В проект добавлен схемный файл mux2_schema.bde

В рабочей области схемного редактора видны только графические изображения портов. Схему мультиплексора еще предстоит нарисовать. Этим мы сейчас и займемся. Для удобства рисования схемы развернем окно редактора **Block Diagram Editor** на полный экран (горячая клавиша **Alt+F9**) и отыщем инструментальную панель **Diagram Items – VHDL** — «Элементы схемы VHDL» (рис. 7). При создании простой схемы нам

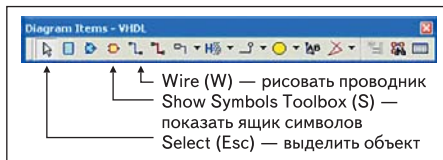


Рис. 7. Инструментальная панель Diagram Items – VHDL

потребуется всего три пиктограммы, отмеченные комментарием на рис. 7.

Первый шаг

На первом шаге необходимо найти и разместить на чертеже схемы графические изображения нужных нам элементов.

Чтобы увидеть «инструментальный ящик с символами», щелкнем по иконке **Show Symbols Toolbox** или нажмем на горячей клавише **S** (рис. 7).

Появится диалоговая панель с одноименным названием (рис. 8). В режиме, заданном по умолчанию, для рисования схемы доступна только одна библиотека со встроенными символами **Built-in symbols**.

Раскрыв ее содержимое, вы увидите весьма скудный набор простейших логических элементов, там нет даже триггеров. Однако для рисования схемы мультиплексора их вполне достаточно.

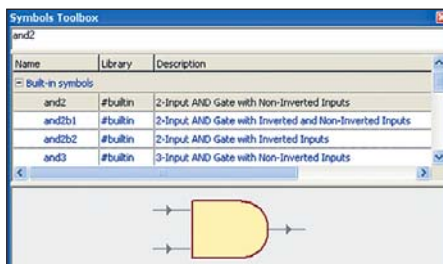


Рис. 8. Инструментальный ящик символов (Symbols Toolbox)

Найдем подходящие для построения мультиплексора элементы (**and2**, **inv**, **or2**) и отбуксируем их один за другим на чертеж схемы (методом буксировки **Drag & Drop**).

Чтобы сделать видимыми позиционные обозначения символов, исполним команду **View Texts** из меню **Diagram**. На открывшейся панели **View Texts** установим флажок **NAME**. Позиционные обозначения всех символов на схеме станут видимыми (рис. 9).

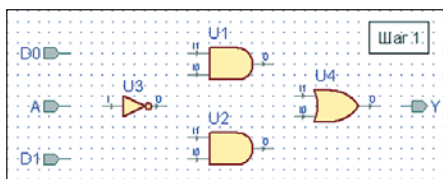


Рис. 9. Размещаем на чертеже схемы логические элементы мультиплексора и визуализируем их позиционные обозначения

Второй шаг

На втором шаге элементы схемы необходимо соединить проводниками в соответствии со структурой мультиплексора.

Нажмем на панели инструментов **Diagram Items** кнопку **Wire** или на клавиатуре клавишу **W**. По изменившейся форме курсора станет ясно, что активизировался режим рисования проводников схемы.

Поместите курсор в точку, из которой вы собираетесь начать проводник, и щелкните левой кнопкой мыши (**ЛКМ**). Переместите курсор в точку назначения и опять щелкните **ЛКМ**. Если проводник имеет сложную форму, придется фиксировать каждый его сегмент (излом, угол) щелчком мыши (о других способах создания соединений мы поговорим на следующих уроках). Законченная схема должна выглядеть так, как показано на рис. 10.

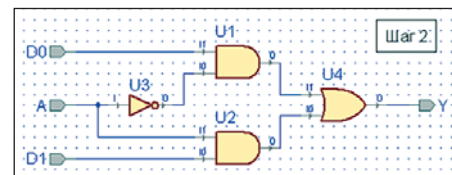


Рис. 10. Выполняем электрические соединения в схеме мультиплексора

Третий шаг

На третьем шаге проводникам схемы надо назначить имена (рис. 11). Следует, правда, заметить, что в простых схемах, как наша, этот шаг можно вообще пропустить.

Действительно, цепи, подключенные к терминалам (входным и выходным портам), автоматически получают те же самые названия, что и соответствующий порт. А внутренние проводники система именует по шаблону, например, **NET9**, **NET13**, **NET17** и др., так что у них тоже есть системные имена.

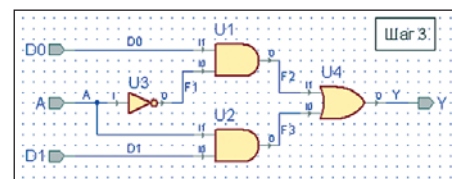


Рис. 11. Проводникам схемы назначаем имена

Однако если пользователь собирается контролировать сигналы на внутренних цепях, желательно присвоить им какие-то осмысленные названия, хотя бы такие: **F1**, **F2** и **F3** (рис. 11). Тогда их станет легче находить.


Чтобы сделать имя видимым на экране монитора, достаточно дважды щелкнуть на проводнике мышью, затем на открывшейся диалоговой панели **Wire Properties** выбрать закладку **View Text** и установить флажок **NAME**. Так же просто заменить системное имя пользовательским. Откройте диалоговую панель **Wire Properties** и введите в поле **Segment** желаемое имя.

Четвертый шаг

На четвертом шаге полезно (хотя и не обязательно) выполнить электрический контроль схемы (команда **Check Diagram** в выпадающем меню **Diagram**). Эта процедура весьма полезна, если у вас остаются сомнения относительно правильности нарисованной схемы. Система не только проинформирует вас о числе ошибок и предупреждений, но и покажет те места на схеме, где они обнаружены.

Пятый шаг

На пятом шаге, который тоже можно пропустить, графическое описание проекта (в нашем примере — схема) должно быть конвертировано в текстовое описание (в нашем примере — в формат языка VHDL).

Выполняется такая работа командой **Generate HDL Code**, которую можно активизировать щелчком по иконке  или из выпадающего меню **Diagram**. Кроме того, эта команда «висит» и на ПКМ всплывающего меню. Выполнив ее, вы обнаружите в окне просмотра проекта **Design Browser** новый файл **mux2_schema.vhd** (рис. 12).

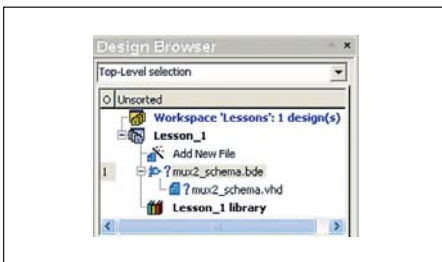



Рис. 12. В проекте появился новый текстовый файл **mux2_schema.vhd**

Познакомиться с его содержимым можно, если щелкнуть по иконке . Рассмотрим автоматически сгенерированное VHDL-описание мультиплексора **mux2_schema**:

```
library IEEE;
use IEEE.std_logic_1164.all;

entity mux2_schema is
port(A, D0, D1 : in STD_LOGIC;
Y : out STD_LOGIC);
end mux2_schema;
architecture mux2_schema of mux2_schema is

signal F1, F2, F3 : STD_LOGIC;


begin
F2 <= F1 and D0;
F3 <= A and D1;
F1 <= not(A);
Y <= F3 or F2;



end mux2_schema;
```

Еще проще открыть этот файл, дважды щелкнув на нем ЛКМ в окне **Design Browser** (рис. 12). Сравнивая графическое и текстовое описание мультиплексора, вы без труда обнаружите их полное соответствие.

Заметим, правда, что описания компонентов **and2**, **inv** и **or2** представлены в данном


случае в потоковой форме (это особенность элементов встроенной библиотеки).

Закончив создание схемы, обратим свой взор на окно просмотра проекта **Design Browser** (рис. 12). Там слева от имени схемного файла по-прежнему красуется подозрительный знак вопроса . Значит, что-то не так.


Вероятно, вы уже догадываетесь, в чем причина: система подсказывает, что созданный проект необходимо откомпилировать. Щелкнем на этой строке ПКМ и исполним команду **Compile** (она дублируется горячей клавишей **F11** и иконкой  на инструментальной панели). Если компиляция пройдет успешно, то вместо знака «вопрос» появится зеленая «галочка»: .


Задание диаграмм входных сигналов

Мысленно поздравив себя с первым успехом, двинемся дальше. Предстоит еще задать временные диаграммы на входах мультиплексора, без них симулятор отказывается работать.

Временные диаграммы внешних воздействий задаются в специальном окне редактора сигналов **Waveform Editor**, который вызывается щелчком по пиктограмме **New Waveform**  на инструментальной панели **Standard**. Дословно **Waveform** переводится как «форма волны», «сигнал». Отсюда и название — редактор сигналов, или сигнальный редактор.

При запуске редактор **Waveform Editor** открывает новую закладку поверх ранее нарисованной схемы, которая состоит из двух частей. Левая часть предназначена для ввода сигналов, правая — для показа временных диаграмм.

Щелкнем ПКМ в левой части окна и выберем в контекстном меню команду **Add Signals**. Она дублируется также на инструментальной панели иконкой . Откроется одноименная панель со списком всех доступных сигналов: **D0**, **D1**, **A**, **Y**, **F1**, **F2**, **F3**. С помощью клавиш **Shift** или **Ctrl** сформируем список сигналов, которые мы хотим наблюдать на экране монитора, например **D0**, **D1**, **A** и **Y**. Нажатием кнопки **Add** перенесем выбранные сигналы в окно редактора **Waveform Editor** (рис. 13) и закроем панель **Add Signals** (кнопка **Close**).

Выделим один или несколько программируемых сигналов и щелчком на иконке **Stimulators**  исполним одноименную команду **Stimulators**. С тем же успехом ее можно выполнить из контекстного меню.

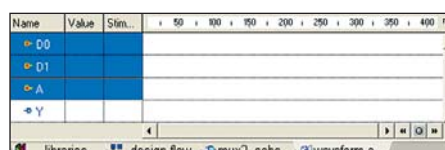


Рис. 13. Задаем список сигналов, которые система будет выводить на экран монитора

Для входных сигналов **D0**, **D1** и **A** зададим временные диаграммы. В пакете есть много различных способов выполнить такую работу. Подробно мы познакомимся с ними на втором уроке. А сейчас воспользуемся одним из самых простых стимуляторов типа **Clock**.

Откроется диалоговая панель **Stimulators** (рис. 14). В окне **Signals** выделим первый сигнал **D0**, в окне **Type** определим для него тип стимулятора **Clock** и в правом окне зададим частоту (**Frequency**) 10 МГц. Наждем кнопку **Apply** и, не закрывая панель, перейдем к программированию второго сигнала **D1**.

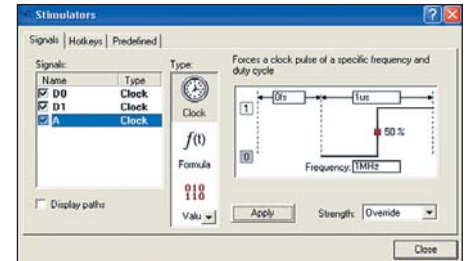




Рис. 14. Программируем внешние воздействия **D0**, **D1** и **A**

Для него зададим частоту вдвое меньше — 5 МГц. Наконец, для третьего входа **A** назначим частоту 1 МГц. Покончив с описанием входных сигналов, закроем панель **Stimulators** (кнопка **Close**).

Моделирование проекта

Теперь все готово для моделирования. Выполним необязательную инициализацию схемы (команда **Initialize Simulation** из меню **Simulation**). Рядом с именами сигналов в столбце **Value** появятся начальные значения всех сигналов (рис. 15). Выходной сигнал **Y** получит значение **U** (**Uninitialized** — не инициализировано). Если сейчас вы переключитесь на закладку **mux2_schema.bde**, то увидите, что и на схеме появились те же самые значения сигналов.

Есть несколько способов запустить схему на моделирование (рис. 16). Проще всего выполнить команду **Run For** из меню **Simulation**. Она дублируется кнопкой  и горячей клавишей **F5**. Имитатор, получив такую команду, продвинет моделирование на один шаг, величина которого задается в соседнем окне справа от кнопки .

По умолчанию шаг равен 100 ns, но его всегда можно поменять. Повторное исполнение

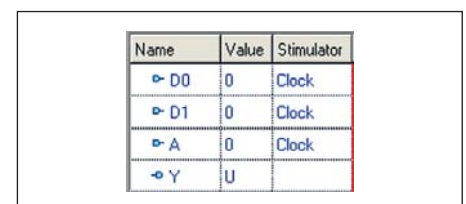


Рис. 15. Выполняем начальную инициализацию схемы

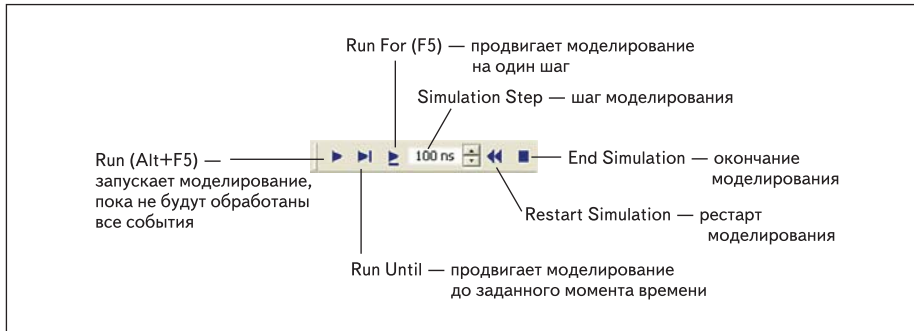


Рис. 16. Кнопки управления процессом моделирования

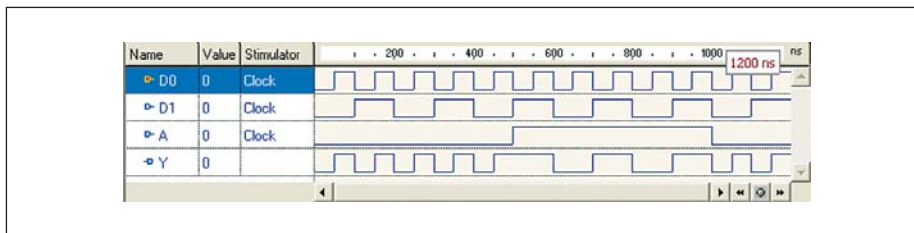


Рис. 18. Результаты моделирования мультиплексора mux2_schema

этой команды продвинет время моделирования еще на один шаг и т. д. В паузе перед очередным шагом вы можете изменить не только его длительность, но и перепрограммировать параметры входных сигналов. Такое пошаговое моделирование весьма полезно при отладке схемы.

Если вы предпочитаете выполнить непрерывное моделирование на всем интересующем вас интервале модельного времени, то можно воспользоваться командой **Run Until...** из меню **Simulation**. Она дублируется кнопкой на инструментальной панели (рис. 16). При этом откроется окно с одноименным названием (рис. 17), где и следует задать желаемое время, например 1.2us (1,2 мкс).

На инструментальной панели есть еще одна «весьма опасная» команда **Run (Alt+F5)**. Она запускает моделирование на неопределенное время, пока не будут обработаны все планируемые события (**transactions**) в схеме.

Если хотя бы один входной сигнал описан как периодический (**Clock**), события на входах будут генерироваться бесконечно, и имитатор не остановится до тех пор, пока время моделирования не достигнет максимального значения **TIME'HIGH**.

Лучший выход из создавшейся ситуации — воспользоваться командой принудительного останова, щелкнув на иконке **Break** , или выполнить команду **Restart** . Возможно, по этой причине команда **Run** отсутствовала на инструментальной панели более ранних версий пакета.

Заметим, что управлять моделированием можно не только из графического интерфейса пользователя, но и из окна консоли (**Console**). Наберите в командной строке следующий текст: **run 1200ns** и нажмите **Enter**. Моделирование сразу продвинется на заданное время (рис. 18).

На временных диаграммах видно, что пока на селекторном входе **A** низкий уровень, на выход передается входной сигнал **D0**. Если **A** принимает значение '1' (момент времени 500 ns), на выходе появляется сигнал **D1**.

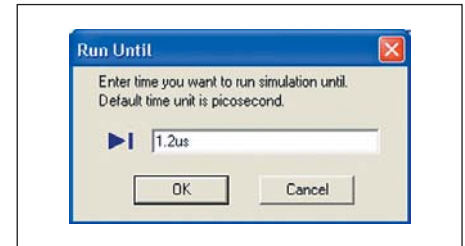


Рис. 17. Задаем финальное время моделирования

Именно так и должен работать мультиплексор (коммутатор) на два входа.

Сохраним результаты моделирования. С этой целью щелкнем ПКМ на названии закладки с временными диаграммами и исполним команду **Save**. Редактор предложит сохранить результаты в файле **Waveform Editor 1.awf**. Лучше отказаться от такого предложения и ввести более информативное имя сигнального файла, например **mux2_schema.awf**.

В среде **Active-HDL 7.1** результаты моделирования можно представить не только в графической, но и в табличной форме. Вернитесь к началу моделирования (команда **Restart**) и выполните команду **New List** (иконка). Откроется новая закладка **list1**, на которую надо поместить интересующие нас сигналы **D0**, **D1**, **A** и **Y** (команда **Add Signals** или кнопка). Выполнив несколько шагов моделирования, вы увидите результаты в табличной форме (рис. 19).

Time	Delta	D0	D1	A	Y
0.000	0	0	0	0	U
0.000	2	0	0	0	0
50.000 ns	0	1	0	0	0
50.000 ns	2	1	0	0	1
100.000 ns	0	0	1	0	1
100.000 ns	2	0	1	0	0
150.000 ns	0	1	1	0	0
150.000 ns	2	1	1	0	1
200.000 ns	0	0	0	0	1
200.000 ns	2	0	0	0	0

Рис. 19. Табличная форма представления результатов моделирования (фрагмент)

Просматривая различные закладки (схему, временные диаграммы, таблицу), вы обнаружите, что они синхронно заполняются данными по мере того, как продвигается моделирование в пошаговом режиме.

Выводы

Краткое знакомство с типовым маршрутом проектирования в среде **Active-HDL** состоялось. Как говорится, лиха беда начало. Кажется, оно у нас получилось вполне успешным.

Теперь, следуя пословице «Повторенье — мать ученья», попробуйте самостоятельно проделать ту же самую работу еще раз, выбрав в качестве объекта проектирования любую схему, которая вам понравится.

На следующем уроке мы познакомимся с различными способами описания внешних воздействий.