

**Лекция 7.****Система моделирования цифровых схем – MODELSIM  
и применение ее для проектирования устройств  
на базе ПЛИС ALTERA****СОДЕРЖАНИЕ**

1 Введение.....	2
2 Назначение и основные характеристики пакета ModelSim .....	2
3 Получение и установка свободно распространяемых версий пакета ModelSim .....	3
4 Пользовательский интерфейс пакета ModelSim .....	4
5 Использование ModelSim при проектировании на ПЛИС.....	10
5.1 Общая технология моделирования в MODELSIM .....	11
5.1.1 Создание библиотек.....	11
5.1.2 Подключение библиотек .....	12
5.1.3 Компиляция исходных файлов (Verilog HDL) .....	12
5.1.4 Загрузка проекта.....	12
5.1.5 Добавление входных воздействий.....	13
5.1.6 Наблюдение результатов моделирования.....	13
5.1.7 Запуск и перезапуск моделирования.....	14
5.2 Пример функционально–логического моделирования .....	15
5.3 Пример функционально–временного моделирования .....	17
5.4 Компиляция библиотек ресурсов для работы с ALTERA QUARTUS.....	20

## 1 Введение

Одна из характерных тенденций современного этапа развития технологии проектирования цифровых систем - применение языков описания аппаратуры HDL (Hardware Description Language) высокого уровня, среди которых наибольшее распространение получили VHDL и Verilog.

При этом не только возрастает роль средств моделирования в процессе разработки устройств, но и меняются методы и требования, предъявляемые к процедурам верификации. Для своевременного обнаружения возможных ошибок средства моделирования должны обеспечивать возможность контроля результатов каждого этапа процесса проектирования: создания исходных HDL-описаний, синтеза, размещения и трассировки в кристалл. Такой подход обеспечивает минимальное время разработки устройства и сокращает стоимость этого процесса, так как цена ошибки возрастает с каждым последующим шагом проектирования.

Пакет программных средств ModelSim корпорации Model Technology (одного из подразделений компании Mentor Graphics) в настоящее время является одной из наиболее распространенных систем HDL-моделирования.

## 2 Назначение и основные характеристики пакета ModelSim

- Полная поддержка основных стандартов языков VHDL и Verilog и их расширений (таких как Verilog 2001, VHDL2002).
- Полная совместимость со спецификациями 1.0 - 3.0 стандартного формата описания задержек SDF (Standard Delay Format) обеспечивает возможность обратной аннотации временных параметров. Кроме того, пакет ModelSim удовлетворяет требованиям VCD (Value Change Dump) по формированию стандартных выходных векторов для VHDL и Verilog и PLI (Programming Language Interface).
- Поддержка инновационных технологий и связанных с ними языков моделирования аппаратуры, таких как: System C 2.0, SystemVerilog, PSL, и т.д.
- Единое ядро моделирования пакета SKS (Single Kernel Simulation), обеспечивающее возможность полной отладки <смешанных> проектов, которые одновременно содержат модули, написанные на VHDL и Verilog и др.
- Высокая скорость компиляции и моделирования (по сравнению с другими пакетами моделирования ActiveHDL, NC-Verilog), обеспечивающая минимальное время отладки систем различного уровня сложности. Одним из главных факторов, повышающих производительность, является использование принципа оптимизированной прямой компиляции. В соответствии с этим принципом исходные (VHDL-, Verilog, System C)-описания компилируются в машинно-независимый объектный код, исполняемый на любой поддерживаемой платформе.
- Открытая архитектура программных средств ModelSim, обеспечивающая тесную интеграцию с пакетами САПР <третьих> фирм. Пользователь может выполнять этапы моделирования фактически в рамках основной системы проектирования, в среде которой осуществляется процесс разработки устройства.
- Средства управления пользовательским интерфейсом Tcl (Tool command language) и Tk (Tool kit) предоставляют возможность организации прямого доступа к моделирующему ядру ModelSim, загрузки информации о выполнении процесса моделирования и его результатов в среду используемой САПР и управления работой системы ModelSim через интерфейс применяемых средств проектирования. Возможен также и противоположный метод интеграции с программным обеспечением других фирм, когда в качестве основной системы используется ModelSim, интерфейсная оболочка которой адаптируется для управления выбранным пакетом САПР.

- Наличие защищенного режима компиляции моделей, гарантирующего выполнение требований, предъявляемых к охране интеллектуальной собственности, к которой относятся коммерчески распространяемые модули (IP Core). В обычном режиме разрешается полная отладка моделей с доступом к исходному коду и внутренней структуре объекта. Имеется возможность использования интеллектуальных продуктов распространяемых производителем в скомпилированном виде без передачи исходного кода модели.
- Имеется встроенный индикатор активности кода (анализатор тестового покрытия), который не только повышает эффективность отладки проекта, но и позволяет быстро создавать более полные и надежные тестовые эксперименты. Этот инструмент предоставляет возможность проследить строки исходного кода, которые не <активизировались> в процессе моделирования, и вывести в графической форме соответствующий отчет обо всех файлах проекта. Индикатор активности кода может быть использован как на уровне отдельного блока, так и для всей системы в целом.
- Имеется встроенный анализатор производительности, позволяющий повысить скорость моделирования за счет обнаружения в проекте и последующего устранения факторов, оказывающих отрицательное влияние на быстродействие этого процесса. С помощью этого инструмента можно получить информацию о библиотечных элементах, обработка которых требует значительных временных затрат, фрагментах исходного кода, написанных нерационально с точки зрения скорости его исполнения, неиспользуемых сигналах в выводимых списках, избыточном коде в тестовых последовательностях. Исключение перечисленных элементов из проекта позволяет резко снизить общее время моделирования.
- Возможность работы в различных режимах, в том числе и пакетном. Разработав и отладив некоторый сценарий моделирования в интерактивном режиме, можно оформить его для последующего использования в виде пакетного командного файла.
- Доступный для разработчика визуальный пользовательский интерфейс и наличие подробной справочной системы, сокращающие время освоения пакета моделирования. Средства управления пользовательским интерфейсом Tcl и Tk позволяют выполнить настройку его элементов (панелей кнопок, меню) в соответствии с требованиями каждого конкретного пользователя.

### 3 Получение и установка свободно распространяемых версий пакета ModelSim

Система моделирования ModelSim выпускается в двух основных редакциях: Special Edition (SE), функционирующей под управлением операционных систем (ОС) UNIX, Linux, Windows 98, Windows 2000, Windows NT, Windows XP, и Personal Edition (PE), предназначенной только для ОС семейства Windows. Кроме того, имеются специализированные редакции, предназначенные для интеграции с пакетами САПР <третьих> фирм.

Примерами специализированной редакции является ModelSim Xilinx Edition (XE), ориентированная на интеграцию с программным обеспечением фирмы Xilinx или ModelSim Altera Edition ориентированная на интеграцию с программным обеспечением фирмы Altera. Как правило, подобные редакции имеют ряд функциональных ограничений: исходный текст описания должен содержать не более 500 исполняемых выражений; если объем исходного кода превышает указанное значение, то программа моделирования сохраняет работоспособность, но функционирует с очень низкой производительностью и т.д.

Чтобы на практике изучить возможности этой системы моделирования, можно воспользоваться оценочными версиями ModelSim (SE и PE), которые распространяются бесплатно. Лицензии этих версий обеспечивают возможность полноценной работы с пакетом в течение 30 дней с момента установки. Получить бесплатную версию программного обеспечения ModelSim (SE или PE) можно двумя путями. В первом случае следует обратиться

на веб-сайт (<http://www.megrates.ru/>) компании Megrates (Mentor Graphics Technologies) - официального представителя Mentor Graphics в России и СНГ, заполнить регистрационную анкету запроса оценочной версии требуемого программного продукта (в частности, пакета ModelSim). По указанному в анкете адресу будет выслан CD-ROM с соответствующим программным обеспечением. После установки программы необходимо получить файл license.dat, содержащий лицензионный код.

Перед началом работы с пакетом моделирования необходимо активизировать полученный лицензионный код. Для этого следует запустить программу Licensing Wizard из меню Программы/ModelSim..., которая выполнит поиск файла лицензии и автоматически установит полный путь доступа к нему, а также необходимые переменные окружения.

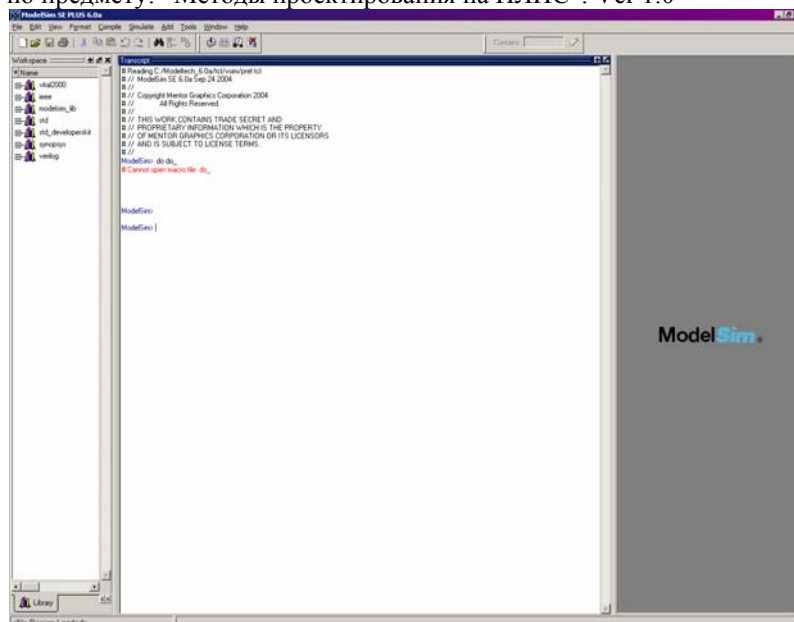
## 4 Пользовательский интерфейс пакета ModelSim

Графический интерфейс системы моделирования ModelSim представлен следующими окнами (возможны варианты в зависимости от версии пакета):

- основное окно (Main window);
- окно исходного кода (Source);
- окно структуры проекта (Structure);
- окно переменных (Variables);
- окно сигналов (Signals);
- окно таблиц (List);
- окно процессов (Process);
- окно временных диаграмм (Wave);
- окно трассировки сигналов (Dataflow).

Основное окно ModelSim является главным элементом пользовательского интерфейса пакета. Оно содержит все необходимые инструменты управления процессами компиляции и моделирования. Открытие вспомогательных окон, используемых для отображения информации о проекте и результатах моделирования в различной форме, и их конфигурирование осуществляется из главного окна. Структура основного окна ModelSim (рис. 1) включает следующие элементы:

- заголовок окна;
- главное меню;
- оперативная панель управления;
- рабочая область;
- консольная область;
- строка состояния.



В заголовке главного окна отображается название программы моделирования с указанием ее версии и редакции.

Главное меню основного окна открывает доступ ко всем командам управления проектом, процессами компиляции и моделирования, отображением результатов моделирования, установки необходимых значений параметров и конфигурирования системы.

Каждый пункт главного меню используется для активизации одноименного всплывающего меню, в котором представлена соответствующая группа команд.

Всплывающее меню File включает команды создания, открытия и закрытия проектов, файлов и наборов данных, а также очистки и сохранения содержимого консольной области в виде файла на диске. Кроме того, в эту группу включены команды установки параметров консольной области и завершения работы.

Выпадающее меню Edit объединяет команды редактирования исходного HDL-кода и установки точек прерывания.

Во всплывающем меню Design представлены команды работы с библиотеками проекта, компиляции исходного кода, инициализации и завершения моделирования. В выпадающем меню View сгруппированы команды, определяющие вид основного окна ModelSim и открывающие вспомогательные окна.

Всплывающее меню Project содержит команды работы с проектом, которые позволяют включать в проект файлы исходных описаний и выполнять компиляцию всех файлов проекта. Команды выпадающего меню Run предназначены для управления процессом моделирования проекта. Команды всплывающего меню Macro используются для запуска процедуры выполнения командных файлов.

Выпадающее меню Options предоставляет доступ к командам установки значений параметров компиляции, моделирования и графического интерфейса. Всплывающее меню Window содержит команды установки взаимной конфигурации открытых окон пакета ModelSim. Пункт Help основного меню предоставляет справочную информацию о программах пакета.

Оперативная панель управления содержит кнопки быстрого доступа, которые дублируют наиболее часто используемые команды меню пакета ModelSim.

Рабочая область основного окна ModelSim представляет собой эффективный визуальный инструмент управления проектом. Здесь отображается детальная информация о структуре текущего проекта и используемых библиотек. Выполнение необходимых операций осуществляется с помощью команд всплывающих контекстно-зависимых меню, которые выводятся на экран при щелчке правой кнопки мыши на свободном поле или на выделенной строке в рабочей области. Эта область содержит, как правило, несколько страниц, снабженных

закладками с их названиями (Project, Library, Sim), каждая из которых представляет соответствующий вид информации о проекте.

Страница Project предназначена для отображения списка файлов, которые входят в состав текущего проекта. С помощью команд всплывающего контекстно-зависимого меню пользователь может быстро открыть выбранный файл для редактирования, выполнить компиляцию отдельного или всех файлов, добавить или удалить файлы из проекта. Страница Library позволяет просмотреть состав библиотек, используемых в проекте, и выполнить необходимые операции с ними.

В поле выбора, расположенном в верхней части страницы, указывается название библиотеки, структуру которой необходимо просмотреть. Для этого используется выпадающий список доступных библиотек, который выводится при нажатии кнопки, расположенной у правой границы поля выбора. Команды всплывающего контекстно-зависимого меню позволяют редактировать и удалять модули, входящие в состав выбранной библиотеки, перекомпилировать ее, а также создавать новые библиотеки.

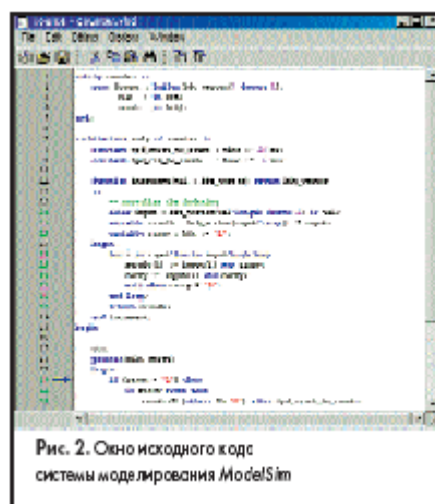
Страница Sim содержит подробную информацию об иерархической структуре проекта, для которого активизирован процесс моделирования. Эта информация может отображаться в сжатом или развернутом виде. Выбор формата отображения осуществляется с помощью команд всплывающего контекстно-зависимого меню. Кроме рассмотренных, в рабочей области могут присутствовать страницы, отражающие структуру открытых наборов данных.

В консольной области основного окна фиксируются выполняемые команды и все сообщения программы, которые появлялись в текущем сеансе работы пакета ModelSim. Кроме того, в нижней части этой области отображается командная строка, в начале которой указывается заголовок, соответствующий текущему режиму работы.

Командная строка позволяет издавать команды управления, используя клавиатуру. Содержимое консольной области может быть сохранено в виде файла на диске. Сохраненные файлы далее могут использоваться в качестве макросов (командных файлов), вызываемых в последующих сеансах работы с программой.

В строке состояния, расположенной в нижней части основного окна, отображается информация о загруженном проекте, текущем времени и итерации моделирования, название выбранного элемента проекта.

Окно исходного кода (рис. 2) позволяет просматривать и редактировать текст HDL-описания выбранного элемента структуры проекта. Это окно автоматически открывается при активизации соответствующей строки в рабочей области основного окна пакета ModelSim и используется в процессе отладки описания выбранного элемента проекта.



Структура окна исходного кода, помимо стандартных элементов управления (строки меню и оперативной панели инструментов), включает панели. В левой панели отображаются номера строк исходного кода и условные обозначения, используемые в процессе отладки. Здесь содержится информация о точках прерывания, исполняемой строке кода и выбранном процессе. В правой панели содержится текст HDL-описания.

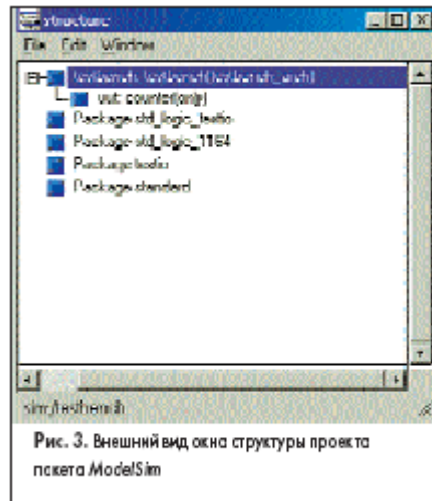


Рис. 3. Внешний вид окна структуры проекта пакета ModelSim

Окно Structure (рис. 3) представляет в иерархической форме информацию о структуре проекта и всех его элементов. Это окно фактически дублирует содержание аналогичной страницы в рабочей области основного окна ModelSim. При развернутой форме отображения в окне структуры содержится подробная информация обо всех HDL-элементах различного уровня иерархии проекта. Выделение элементов проекта в окне структуры влечет за собой отображение соответствующей информации в окнах исходного кода и сигналов.

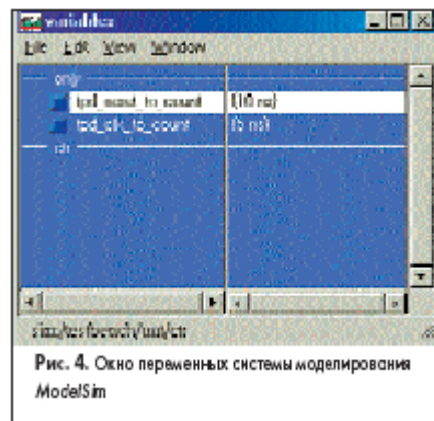


Рис. 4. Окно переменных системы моделирования ModelSim

Окно Variables (рис. 4) предназначено для контроля текущих значений переменных, параметров и констант выбранного процесса. Рабочая область этого окна содержит две панели. В левой панели отображаются названия переменных, параметров и констант, а в правой - их текущие значения. Для элементов составных типов (массивов или записей) используется иерархическая форма представления. Окно переменных предоставляет возможность оперативного изменения отображаемых значений в ходе отладки.

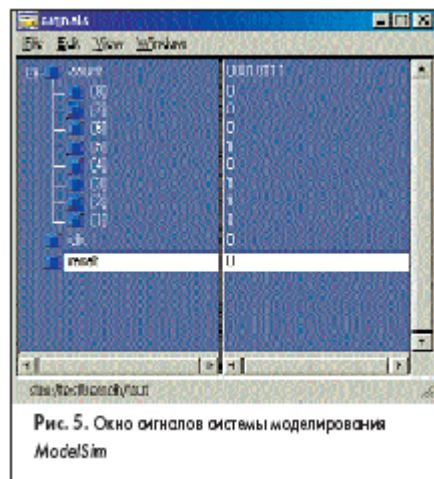
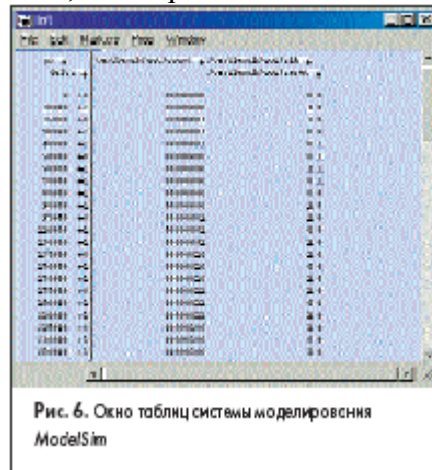


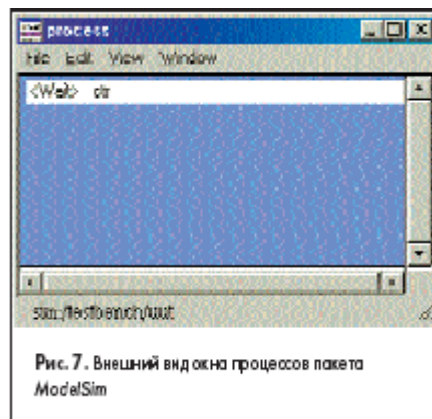
Рис. 5. Окно сигналов системы моделирования ModelSim



Окно Signals (рис. 5) используется для отображения текущего состояния сигналов (на момент завершения шага моделирования) в символьной форме. Структура этого окна, кроме стандартных элементов управления (строки меню, оперативной панели инструментов и строки состояния), включает две панели. Левая панель содержит список идентификаторов сигналов, соответствующих выделенному элементу в окне структуры проекта, а правая - их текущие значения. Сигналы составных типов (массивов или записей) отображаются в виде иерархической структуры, которая может быть представлена в сжатой или развернутой форме. Двойной щелчок левой кнопки мыши на названии сигнала открывает окно исходного кода, представляющее фрагмент описания, в котором объявляется этот сигнал.



Окно List (рис. 6) представляет результаты моделирования в табличной форме. Отображаемая здесь таблица разделена на две области. Левая область содержит последовательность дискретных отсчетов времени моделирования. Каждый столбец в правой области представляет значения сигнала или переменной в соответствующие моменты моделирования. В верхней части этой области отображаются полные названия сигналов с учетом иерархии проекта. Команды меню рассматриваемого окна позволяют сохранять результаты моделирования в текстовом виде в различных форматах.

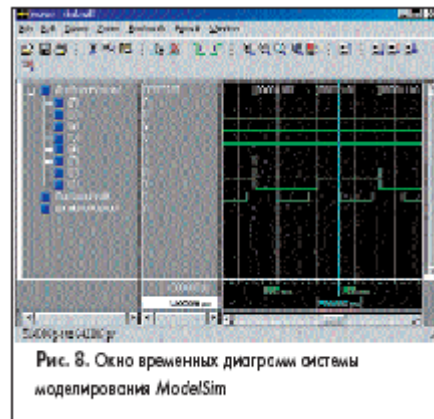


Окно Process (рис. 7) предназначено для отображения списка процессов в выбранном структурном фрагменте проекта или запланированных для выполнения в текущем цикле моделирования. Команды меню View рассматриваемого окна определяют, по какому принципу формируется выводимый список процессов. Перед каждым элементом списка указано обозначение состояния, в котором находится соответствующий процесс. Возможны три варианта обозначений:

- <Ready> - соответствует состоянию процесса, выполнение которого планируется в текущем интервале времени;
- <Wait> - указывает на то, что процесс находится в состоянии ожидания изменения сигнала или указанного временного интервала;



- <Done> - соответствует состоянию процесса, завершившего выполнение оператора ожидания.

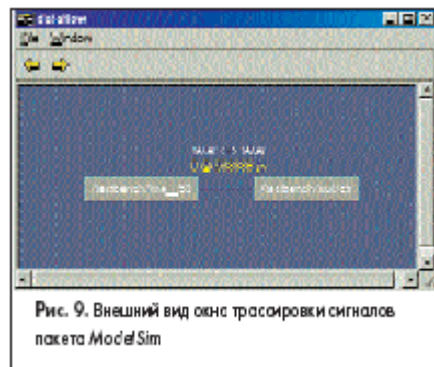


Окно временных диаграмм (рис. 8) используется для представления результатов моделирования в графическом виде. В рабочей области этого окна расположены три панели.

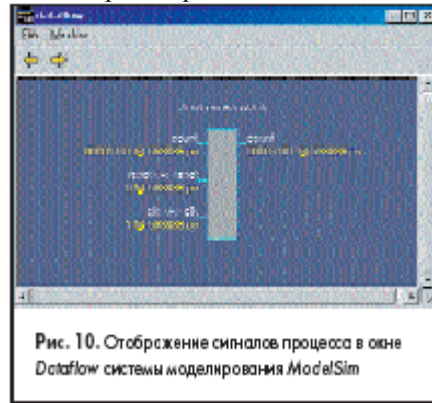
Левая панель содержит список контролируемых сигналов. В средней панели указывается состояние сигналов в момент времени, который фиксируется положением курсора в панели временных диаграмм. В нижней части этой панели выводится значение времени моделирования, на котором установлен курсор.

Правая панель отображает результаты моделирования в форме временных диаграмм. В нижней секции этой панели расположена ось времени моделирования. Управление масштабом изображения осуществляется с помощью инструментов, расположенных на оперативной панели, и команд меню Zoom этого окна.

Для перемещения по временной оси следует использовать элементы прокрутки этой панели. Окно временных диаграмм предоставляет возможность измерения временных интервалов между моментами переключения сигналов, а также сохранять результаты моделирования в графической форме в виде файла на диске для последующего анализа и использования.



Окно Dataflow позволяет проследить <прохождение> выбранного сигнала в иерархической структуре проекта (рис. 9). Название анализируемого сигнала отображается в центре рабочей области окна. В левой части расположены названия процессов, которые формируют поведение сигнала, а в правой - названия процессов, которые используют этот сигнал в качестве входного, но не изменяют его.



Кроме того, рассматриваемое окно может использоваться для отображения информации о сигналах процесса, выделенного в окне Process (рис. 10). В этом случае в центре рабочей области окна располагается условный графический образ выбранного процесса. Все сигналы, значения которых читаются в процессе, представлены в виде входов, располагающихся с левой стороны условного графического образа, а сигналы, формируемые процессом, - в виде выходов, находящихся с правой стороны условного графического образа.

В окне трассировки рядом с идентификаторами сигналов указываются их значения в момент времени, который зафиксирован положением курсора в окне временных диаграмм. При перемещении этого курсора окно Dataflow отслеживает изменение значений отображаемых сигналов.

## 5 Использование ModelSim при проектировании на ПЛИС

Modelsim - мощная среда моделирования, которая может быть использована для проведения модельных экспериментов с цифровыми системами различного уровня сложности. Тип модельного эксперимента зависит от уровня (способа) представления элементов модели цифровой системы. Таким образом, выделяют два вида моделей цифровых систем:

- функциональные (поведенческие) модели;
- низкоуровневые (gate-level) модели.

Модельные эксперименты с функциональными и функционально–логическими моделями применяются на ранних стадиях проектирования и поэтому являются менее ресурсоемкими с точки зрения проведения комплексных модельных экспериментов, по сравнению с низкоуровневыми gate-level моделями.

Ресурсоемкость gate-level моделей обусловлена тем, что они содержат в себе информацию о проекте на уровне элементарных логических элементов и связей между ними. Для каждого элемента данной модели (лог. элемента или связи) присутствует информация о физическом времени распространения сигнала. Поэтому ресурсоемкость моделирования цифровых систем, функциональность которых описана в базе этих моделей, больше.

Время, затрачиваемое на моделирование функционально–логической модели (для 1 секунды модельного времени) может составлять порядка 10–20 секунд притом, что моделирование gate-level модели может занять порядка 10 – 20 минут.

## 5.1 Общая технология моделирования в MODELSIM

В этом разделе будет описываться технология моделирования цифровых систем с использованием языка VERILOG HDL, применительно к процессу моделирования цифровых систем создаваемых на основе ПЛИС фирмы ALTERA семейство FLEX10KE (ACEX). В качестве основного средства управления средой моделирования MODELSIM будет использован командный язык.

Рассмотрим основные этапы работы с MODELSIM при моделировании цифровой схемы:

- Создание библиотек.
- Подключение библиотек
- Компиляция исходных текстов проекта и текстов TESTBENCH
- Загрузка проекта
- Добавление входных воздействий
- Наблюдение результатов моделирования
- Перезапуск моделирования

### 5.1.1 Создание библиотек

При работе с MODELSIM разделяют два вида библиотек:

- Рабочая библиотека (Working Library)
- Библиотека ресурсов (Resource Library)

Рабочая библиотека предназначена для хранения результатов компиляции рабочих файлов. Под рабочими файлами будем понимать исходные тексты на VERILOG HDL, которые подвергаются периодическому изменению в ходе разработки (и как следствие требуют частой перекомпиляции). Изменения в этих файлах могут быть связаны со всяческими усовершенствованиями, фиксацией ошибок, и т.д.

Библиотека ресурсов содержит в себе редко изменяемые компоненты, прекомпиляция которых происходит достаточно редко. Эта библиотека(библиотеки) может содержать в себе модели логических компонентов, параметризуемых мегафункций и т.д., которые используются в рабочих проектах как неизменные блоки.

*Для создания библиотеки в интерактивном режиме следует:*

1. Choose New > Library (File menu). The Create a New Library dialog box appears.
2. Under Create, select a new library and a logical mapping to it.
3. In the Library Name box, type the library name.
4. Under Library Physical Name, specify the library's name or path.
5. Click OK.

*Для создания библиотеки в режиме командной строки следует набрать:*

```
vlib <library_name>
```

где **library\_name** это название–идентификатор библиотеки.

### 5.1.2 Подключение библиотек

*Для подключения библиотеки в интерактивном режиме следует:*

1. Choose New > Library (File menu). The Create a New Library dialog box appears.
2. Under Create, select a map to an existing library.
3. Under Library Name, enter the name of the library.
4. Under Library Physical Name, specify the library's name or path.
5. Click OK.

*Для подключения библиотеки в режиме командной строки следует набрать:*

**vmap <library\_name> <library name or directory path>**

где **library\_name** – название идентификатор библиотеки,  
**library\_name or directory\_path** - местоположение библиотеки в файловой системе.

### 5.1.3 Компиляция исходных файлов (Verilog HDL)

Порядок компиляции файлов проекта и TESTBENCH не имеет значения. Компилятор исходных файлов является инкрементным, это означает, что из указанного списка компилируемых файлов, для непосредственной компиляции будут выбраны только те файлы, которые были изменены по сравнению с теми, что находятся в рабочей библиотеке.

*Для проведения компиляции в интерактивном режиме следует:*

1. Choose Compile (Compile menu).
2. Specify the working library.
3. Specify the Verilog HDL files and click Compile.

*Для проведения компиляции в режиме командной строки следует набрать:*

**vlog -work <library\_name> <file1>.v <file2>.v**

где **library\_name** – название идентификатор рабочей библиотеки.

### 5.1.4 Загрузка проекта

После компиляции измененных файлов проекта, для начала моделирования проект необходимо загрузить. Необходимо отметить, что процедуре загрузки подлежит только тот модуль проекта (не файл, а модуль!) который является модулем высшего уровня (имеется в виду высший уровень проектной иерархии), находящийся в рабочей библиотеке.

Если планируется провести функциональное или функционально–логическое моделирование, дополнительных опций при загрузке проекта не требуется. В случае проведения временного моделирования (Gate-Level Timing Simulation), следует загрузить дополнительно файл SDF описывающий физические временные задержки на элементах, указав какой уровень проверок будет использован (MIN, TYP, MAX).

1. Choose Simulate (Simulate menu).
2. Specify the highest-level design unit or testbench and click Load.

Для загрузки проекта в режиме командной строки следует набрать:

**vsim work.<top-level\_design\_unit>**

где **top-level\_design\_unit** – главный модуль проекта.

### 5.1.5 Добавление входных воздействий

При моделировании проекта цифровой схемы, обычно используется специальная модель TESTBENCH, которая берет под свое управление все входы проверяемого/наблюдаемого модуля. Однако средствами MODELSIM можно добавить недостающие входные воздействия или переопределить существующие (без изменения TESTBENCH), если таковые имеются с использованием команды

**force <signal name to be forced> <value> <time>, <value> <time>**

более подробную информацию по данной команде можно получить из командной строки MODELSIM

**help force**

или из документа

C:\Modeltech\_6.0c\docs\pdf\se\_cmds.pdf

### 5.1.6 Наблюдение результатов моделирования

Пакет программ MODELSIM имеет много различных инструментов для наблюдения результатов моделирования. Одним из наиболее популярных является окно просмотра временных диаграмм цифровых сигналов.

Чтобы его вызвать из командной строки сделаем следующее:

**view wave**

для просмотра значений сигналов, нужно их добавить в окно просмотра временных диаграмм, для этого можно воспользоваться командой:

**add wave**

или сделать это с использованием графических средств среды.

более подробную информацию по данной команде можно получить в из командной строки

**help add wave**

или

C:\Modeltech\_6.0c\docs\pdf\se\_cmds.pdf

В опциях этой команды имеется возможность гибко настраивать способ отображения всех сигналов при моделировании.

### 5.1.7 Запуск и перезапуск моделирования

Ключом к запуску процесса моделирования является команда

**run**

к которой, как и к другим, ранее рассмотренным командам, есть обширный файл помощи.

**run 200us**

запускает процесс моделирования на 200us модельного времени.

Если в процессе моделирования вы решили прервать процесс, для этого существует команда

**restart**

которая прерывает процесс моделирования.

## 5.2 Пример функционально–логического моделирования

Рассмотрим пример на основе счетчика (файлы в каталоге FUNC\_MODELING):

ФАЙЛ	Назначение	Содержимое
do_func	главный командный файл	<pre>vlib work vlog -f vlog_func_cmd.cmd vsim -L {D://!OVD/MODELSIM/FUNC/altera_mf/} -L {D://!OVD/MODELSIM/FUNC/220model/} work.my_tb  # do wave.do add wave /my_tb/counter_inst/*  run -all</pre>
vlog_func_cmd.cmd	вспомогательный файл параметров	<pre>-reportprogress 300 -work work -lint my_tb.v counter.v</pre>
counter.v	файл-модуль цифровой системы	<pre>module counter ( input          clk, input          reset_b, input          cnt_en, output reg [3:0] cnt );  always @(posedge clk, negedge reset_b)   if (~reset_b) begin     cnt &lt;= 0;   end else begin     if (cnt_en) begin       cnt &lt;= cnt + 1'b1;     end   end endmodule</pre>
my_tb.v	TESTBENCH	<pre>module my_tb;  reg clk = 0; reg reset_b = 0; reg cnt_en = 0;  always begin   #20;   clk &lt;= ~clk; end  initial begin   cnt_en &lt;= #200 1;   #100;   reset_b &lt;= 1'b1;    #10_000;   \$display ("End of counter experiment");   \$stop; end  counter counter_inst(   .clk(clk),   .reset_b(reset_b),   .cnt_en(cnt_en),   .cnt() );  endmodule</pre>

Для запуска моделирования следует зайти в каталог, где лежат эти файлы и запустить из командой строки **modelsim.exe**

После этого в командной строке MODELSIM запустить командный файл:

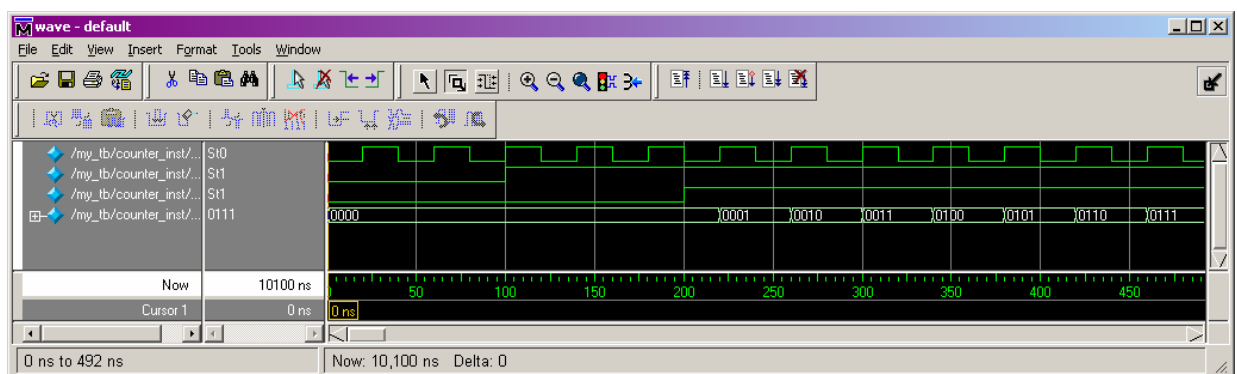
**do do\_func**



Результатом работы этого командного файла будет:

```
# Reading C:/Modeltech_6.0c/tcl/vsim/pref.tcl
# // ModelSim SE 6.0c Feb  2 2005
# //
# // Copyright Mentor Graphics Corporation 2005
# // All Rights Reserved.
# //
# // THIS WORK CONTAINS TRADE SECRET AND
# // PROPRIETARY INFORMATION WHICH IS THE PROPERTY
# // OF MENTOR GRAPHICS CORPORATION OR ITS LICENSORS
# // AND IS SUBJECT TO LICENSE TERMS.
# //
do do_func
# Model Technology ModelSim SE vlog 6.0c Compiler 2005.02 Feb  2 2005
# -- Compiling module my_tb
# -- Compiling module counter
#
# Top level modules:
#   my_tb
# vsim -L D://OVD/MODELSIM/FUNC/altera_mf/ -L D://OVD/MODELSIM/FUNC/220model/ work.my_tb
# Loading work.my_tb
# Loading work.counter
# End of counter experiment
# Break at my_tb.v line 19
# Simulation Breakpoint: Break at my_tb.v line 19
# MACRO ./do_func PAUSED at line 8
```

В окне просмотра временных диаграмм будет следующее:



### 5.3 Пример функционально–временного моделирования

Рассмотрим пример на основе счетчика (файлы в каталоге GATE\_MODELING/counter):

ФАЙЛ	Назначение	Содержимое
counter.qpf	QUARTUS PROJECT FILE	<pre> QUARTUS VERSION = "4.2" DATE = "17:49:08 February 21, 2005" PROJECT REVISION = "counter"                     </pre>
counter.qsf	QUARTUS SETTINGS FILE	<pre> set_global_assignment -name ORIGINAL_QUARTUS_VERSION "4.2 SP1" set_global_assignment -name PROJECT_CREATION_TIME_DATE "17:49:08 FEBRUARY 21, 2005" set_global_assignment -name LAST_QUARTUS_VERSION "4.2 SP1" set_global_assignment -name DEVICE_FILTER_SPEED_GRADE ANY set_global_assignment -name FAMILY ACEX1K set_global_assignment -name TOP_LEVEL_ENTITY counter set_global_assignment -name <b>DEVICE "EP1K100QC208-3"</b> set_global_assignment -name FITTER_EFFORT "STANDARD FIT" set_global_assignment -name ERROR_CHECK_FREQUENCY_DIVISOR 1 set_global_assignment -name EDA_SIMULATION_TOOL "ModelSim (Verilog)" set_global_assignment -name LOGICLOCK_INCREMENTAL_COMPILE_ASSIGNMENT off set_global_assignment -name EDA_OUTPUT_DATA_FORMAT VERILOG -section_id eda_simulation set_global_assignment -name EDA_TIME_SCALE "1 ps" -section_id eda_simulation                     </pre>
counter.v	RTL модель счетчика В моделировании не участвует. Используется для автоматического синтеза.	<pre> module counter ( input          clk, input          reset_b, input          cnt_en, output reg [3:0] cnt );  always @(posedge clk, negedge reset_b) if (~reset_b) begin cnt &lt;= 0; end else begin if (cnt_en) begin cnt &lt;= cnt + 1'b1; end end  endmodule                     </pre>
simulation/modelsim/counter.vo	Список целей сгенерированный в результате логического синтеза исходной RTL-модели. Также на языке VERILOG HDL.	<pre> `timescale 1 ps / 1 ps  module counter ( cnt_en, reset_b, clk, cnt); input cnt_en; input reset_b; input clk; output [3:0] cnt;  wire gnd = 1'b0; wire vcc = 1'b1;  tri1 devclrn; tri1 devpor; tri0 devoe; // synopsys translate_off initial \$sdf_annotate("counter_v.sdo"); // synopsys translate_on  wire \cnt_en-dataout ; wire \reset_b-dataout ; wire \clk-dataout ; wire \cnt_rtl_0 wysi_counter counter_cell[0]~COUT ; wire \cnt_rtl_0 wysi_counter counter_cell[1]~COUT ; wire \cnt_rtl_0 wysi_counter counter_cell[2]~COUT ; wire \cnt_rtl_0 wysi_counter q[3] ; wire \cnt_rtl_0 wysi_counter q[2] ; wire \cnt_rtl_0 wysi_counter q[1] ; wire \cnt_rtl_0 wysi_counter q[0] ;  // atom is at PIN_182 flex10ke_io \cnt_en~I ( .datain(gnd), .clk(gnd), .ena(vcc), .aclr(gnd), .oe(gnd), .devclrn(devclrn), .devpor(devpor), .devoe(devoe), .dataout(\cnt_en-dataout ), .padio(cnt_en)); // synopsys translate_off defparam \cnt_en~I .operation_mode = "input"; defparam \cnt_en~I .reg_source_mode = "none"; defparam \cnt_en~I .feedback_mode = "from_pin"; .....                     </pre>
simulation/modelsim/counter_v.sdo	Файл физических задержек распространения сигналов в сгенерированном списке цепей.	<pre> // // Device: Altera <b>EP1K100QC208-3 Package PQFP208</b> // // // This SDF file should be used for ModelSim (Verilog) only // (Delayfile (SDFVersion "2.1")                     </pre>



Для запуска моделирования следует зайти в каталог, где лежат данные файлы и запустить из командой строки **modelsim.exe**

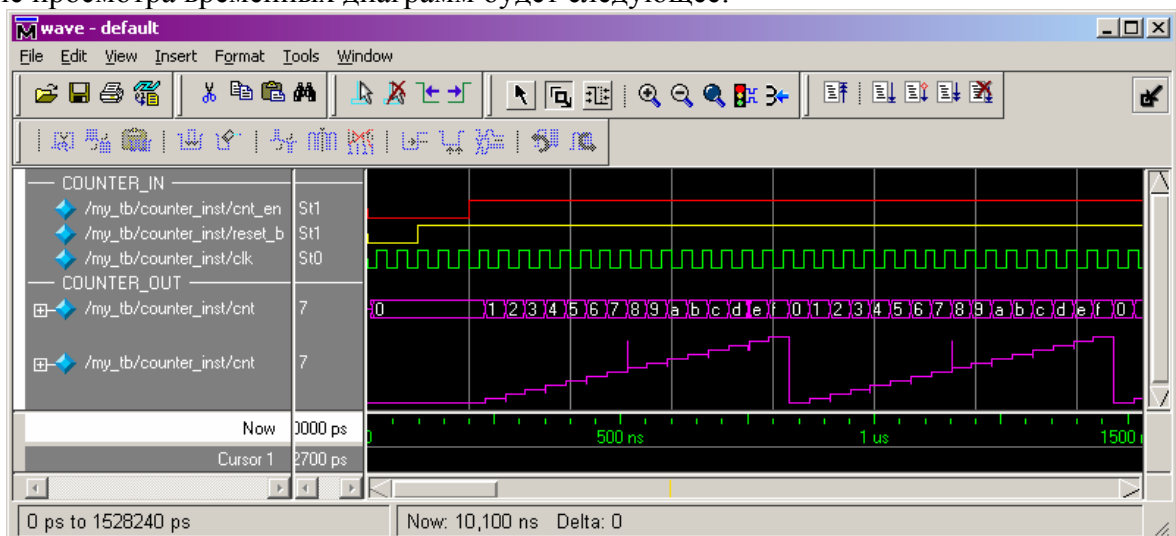
После этого в командной строке MODELSIM запустить командный файл:

**do do\_gate**

Результатом работы этого командного файла будет:

```
# Reading C:/Modeltech_6.0c/tcl/vsim/pref.tcl
# // ModelSim SE 6.0c Feb  2 2005
# //
# // Copyright Mentor Graphics Corporation 2005
# // All Rights Reserved.
# //
# // THIS WORK CONTAINS TRADE SECRET AND
# // PROPRIETARY INFORMATION WHICH IS THE PROPERTY
# // OF MENTOR GRAPHICS CORPORATION OR ITS LICENSORS
# // AND IS SUBJECT TO LICENSE TERMS.
# //
do do_gate
# Model Technology ModelSim SE vlog 6.0c Compiler 2005.02 Feb  2 2005
# -- Compiling module my_tb
#
# Top level modules:
#   my_tb
# Model Technology ModelSim SE vlog 6.0c Compiler 2005.02 Feb  2 2005
# -- Compiling module counter
#
# Top level modules:
#   counter
# vsim -L D:/!OVD/MODELSIM/FUNC/altera_mf/ -L D:/!OVD/MODELSIM/GATE/flex10ke_atoms -sdfmin
counter=counter_v.sdo work.my_tb work.counter
# Loading work.my_tb
# Loading work.counter
# Loading D:/!OVD/MODELSIM/GATE/flex10ke_atoms.flex10ke_io
# Loading D:/!OVD/MODELSIM/GATE/flex10ke_atoms.flex10ke_asynch_io
# Loading D:/!OVD/MODELSIM/GATE/flex10ke_atoms.flex10ke_atoms.dffe_io
# Loading D:/!OVD/MODELSIM/GATE/flex10ke_atoms.flex10ke_lcell
# Loading D:/!OVD/MODELSIM/GATE/flex10ke_atoms.flex10ke_asynch_lcell
# Loading D:/!OVD/MODELSIM/GATE/flex10ke_atoms.flex10ke_lcell_register
# Loading D:/!OVD/MODELSIM/GATE/flex10ke_atoms.PRIM_DFFE
# Loading counter_v.sdo
# Loading counter_v.sdo
# Loading counter_v.sdo
# ** Note: (vsim-3587) SDF Backannotation Successfully Completed.
# Time: 0 ps Iteration: 0 Region: /my_tb File: my_tb.v
# End of counter experiment
# Break at my_tb.v line 20
# Simulation Breakpoint: Break at my_tb.v line 20
# MACRO ./do_gate PAUSED at line 10
```

В окне просмотра временных диаграмм будет следующее:



## 5.4 Компиляция библиотек ресурсов для работы с ALTERA QUARTUS

В предыдущих примерах для моделирования мы использовали специальные командные файлы, в которых мы подключаем внешние библиотеки ресурсов.

```
vsim -L {D://OVD/MODELSIM/FUNC/altera_mf/} -L {D://OVD/MODELSIM/FUNC/220model/} work.my_tb
vsim -L {D://OVD/MODELSIM/GATE/flex10ke_atoms} -sdfmin counter=counter_v.sdo work.my_tb work.counter
```

Для использования библиотеки ресурсов в проекте, их необходимо предварительно скомпилировать и выделить для них специальное место в файловой системе. Данные библиотеки компилируются один раз и могут не изменяться до тех пор, пока не потребуется обновить САПР или использовать другое семейство ПЛИС.

Создадим такую библиотеку:

В качестве места хранения библиотеки выберем каталог: **C:\WORK\MODEL\_LIB**

1. Создаем каталог **C:\WORK\MODEL\_LIB**
2. В данном каталоге создаем подкаталоги:
  - a. **FUNC** [библиотека функциональных моделей]
  - b. **GATE** [библиотека временных моделей]
3. Копируем файлы из каталога **ALTERA/QUARTUS/EDA/SIM\_LIB/** так чтобы в каталоге **FUNC** оказались файлы: **220model.v altera\_mf.v**, а в каталоге **GATE** оказались файлы: **flex10ke\_atoms.v max\_atoms.v**
4. Создаем в каталоге **C:\WORK\MODEL\_LIB** файл **build** который имеет следующее содержание:

```
# компиляция ресурсных библиотек
# библиотека функциональных моделей
cd func

vlib 220model
vlog -reportprogress 300 -work 220model 220model.v

vlib altera_mf
vlog -reportprogress 300 -work altera_mf altera_mf.v nopli.v

cd ..

# библиотека gate-level моделей

cd gate

vlib max_atoms
vlog -reportprogress 300 -work max_atoms max_atoms.v

vlib flex10ke_atoms
vlog -reportprogress 300 -work flex10ke_atoms flex10ke_atoms.v

exit
```

5. Запускаем из командной строки **modelsim.exe** (находясь в каталоге **C:\WORK\MODEL\_LIB**)
6. Запускаем из командной строки командный файл **do build**.

После чего начнется компиляция библиотек, которая по завершении процесса предложит вам выйти из MODELSIM.

[все файлы можно найти в приложении к лекции в соответствующих каталогах]