

Часть 1

Обзор САПР MAX+plus II

В данной главе приводится обзор САПР MAX+plus II и описание его составных частей.

 Для получения полной и обновленной информации по MAX+plus II пользуйтесь справочной системой (**Help**) САПР MAX+plus II.

1.1. Проектирование с помощью MAX+plus II

САПР MAX+plus II (*Multiple Array Matrix Programmable Logic User*) представляет собой архитектурно-независимую среду проектирования, которая легко приспособливается к конкретным проектным требованиям и может работать на различных компьютерных платформах. MAX+plus II предоставляет различные способы ввода проекта, быструю компиляцию и непосредственное программирование микросхем.

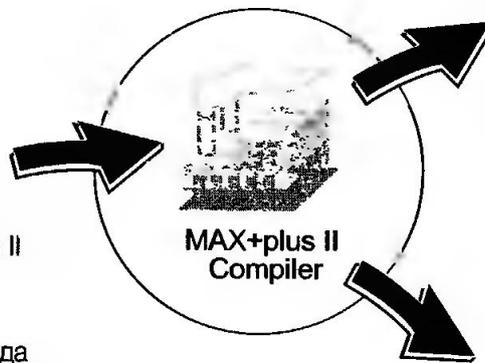
Система проектирования MAX+plus II, как показано на рисунке 1.1, является функционально полным САПР для реализации проектов на микросхемах программируемой логики фирмы Altera семейств Classic, MAX 3000, MAX 5000, MAX 7000, MAX 9000, FLEX 6000, FLEX 8000, FLEX 10K, ACEX1K.

 Для получения информации о поддерживаемых семействах микросхем см. файл MAX+plus II *read.me*.

MAX+plus II предоставляет полный спектр возможностей для проектирования цифрового устройства: различные способы ввода проекта, логический синтез, компиляцию с заданными временными ограничениями, функциональное и временное моделирование, разделение проекта на части и моделирование проекта, выполненного на нескольких микросхемах, временной анализ, автоматическое определение ошибок, программирование и верификацию микросхем. САПР MAX+plus II позволяет вводить и сохранять проекты в виде файлов AHDL (язык описания аппаратуры фирмы Altera), EDIF, Verilog HDL, VHDL, OrCAD. MAX+plus II может использовать netlist файлы фирмы Xilinx для ввода проекта, может создавать выходные SDF (*Standard Delay Format*) файлы для обеспечения удобного интерфейса с другими САПР промышленного стандарта CAE.

Ввод проекта

- Графический редактор MAX+plus II
- Редактор символов MAX+plus II
- Текстовый редактор MAX+plus II
- Редактор временных диаграмм MAX+plus II
- Редактор физического размещения MAX+plus II
- Язык AHDL
- Язык VHDL
- Язык Verilog HDL
- Другие средства для ввода проекта промышленного стандарта CAE



Верификация проекта

- Симулятор MAX+plus II
- Редактор временных диаграмм MAX+plus II
- Временной анализатор MAX+plus II
- Другие средства верификации проекта промышленного стандарта CAE

Программирование микросхем

- Средства программирования фирмы Altera
- Программатор фирмы Data I/O
- Другие промышленные программаторы

Рис. 1.1

MAX+plus II имеет удобный графический интерфейс и простую в использовании справочную систему, содержащую всю необходимую разработчику информацию. MAX+plus II состоит из 11 полностью интегрированных программных модулей, которыми вы пользуетесь при создании своего проекта (рис. 1.2).

Задание многих параметров и выполнение команд, например, открытие файлов, выбор микросхемы, назначение контактов и логических ячеек, компиляция текущего проекта, происходит во многих или во всех программных модулях САПР MAX+plus II одинаково. Если вы освоили работу с одним модулем, то это послужит основой при использовании остальных. Редакторы проекта (графический, текстовый, временных диаграмм) и вспомогательные редакторы (редактор физического размещения и редактор символов) также имеют много общих параметров и команд. Каждый редактор проекта позволяет вам выполнять поиск сигнала или символа, причем во всех редакторах это делается

одинаково. Вы можете в одном иерархическом проекте сочетать различные типы файлов ввода проекта, выбирая такой способ ввода, который больше всего подходит для каждого функционального блока. Обширная библиотека разработанных компанией Altera мега- и макрофункций, включая операционные устройства из библиотеки параметризованных модулей (*Library of Parameterized Modules*), обеспечивает широкий спектр возможностей для ввода проекта.



Рис. 1.2

Вы можете работать с несколькими программными модулями MAX+plus II одновременно. Например, во время компиляции или моделирования одного проекта вы можете вводить другие проекты, используя различные редакторы. Можно просматривать всю иерархию проекта и спокойно перемещаться от одного уровня иерархии к другому, а MAX+plus II будет автоматически запускать соответствующий редактор проекта для каждого файла.

Компилятор является сердцем системы MAX+plus II и обеспечивает эффективную обработку проекта. Вы можете задавать параметры компилятору для достижения наилучшей реализации вашего проекта в выбранной микросхеме. Автоматическое определение местоположения ошибок и обширная документация по ошибкам и предупреждающим

сообщениям делают формальную отладку проектов легкой и быстрой. Вы можете создавать различные выходные файлы для функционального и временного моделирования, моделирования проекта, реализованного на нескольких микросхемах, временного анализа, программирования микросхемы. На каждом этапе процесса проектирования MAX+plus II помогает вам сконцентрироваться на проекте, а не на особенностях пользования программного обеспечения.

Великолепная интеграция программных модулей MAX+plus II позволяет вам максимально повысить эффективность и производительность, предоставляя полный контроль над средой проектирования.

1.2. Процедура проектирования

Упрощенно процесс создания нового проекта от концепции до завершения можно описать следующим образом:

1. Создайте файл нового проекта или иерархический проект из любой комбинации файлов, полученных с помощью различных редакторов проекта САПР MAX+plus II, например, графического, текстового и редактора временных диаграмм.
2. Укажите имя файла высшего уровня в иерархии. Оно будет именем проекта.
3. Назначьте семейство микросхем для проекта. Вы можете либо позволить компилятору выбрать конкретную микросхему для вашего проекта, либо указать тип микросхемы вручную.
4. Откройте окно компилятора (**Compiler**) и нажмите кнопку **Start** для компиляции проекта. Если вы хотите, то можете включить временной экстрактор списка соединений симулятора (**Timing SNF Extractor**), чтобы создать файл со списком соединений для временного моделирования и анализа.
5. Если проект скомпилирован успешно, вы можете по выбору выполнить моделирование и временной анализ:
 - Для запуска временного анализа откройте окно временного анализатора (**Timing Analyzer**), выберите один из режимов анализа и нажмите кнопку **Start**.
 - Для запуска моделирования вы должны сначала создать тестирующие векторы входных переменных. Это можно сделать либо с использованием графического редактора временных диаграмм (**Waveform Editor**), либо с помощью текстового редактора (**Text Editor**). В первом случае результат будет записан в файл временных диаграмм (**Simulator Channel File**) с расширением **.scf**, во втором — в векторный файл (**Vector File**) с расширением **.vec**. После этого откройте окно **Simulator** (симулятор) и нажмите кнопку **Start**.

6. Откройте окно **Programmer** (программатор) и вставьте микросхему в адаптер на базовом модуле программатора (MPU), либо подключите загрузочные устройства MasterBlaster или ByteBlaster MV к технологическому разъему на плате, где распаяна микросхема.
7. Нажмите кнопку **Program** для программирования микросхем, выполненных по технологии EEPROM, или нажмите кнопку **Configure** для конфигурирования микросхем, выполненных по технологии SRAM.

1.3. Запуск MAX+plus II



MAX+plus II можно запустить одним из следующих способов:

- ✓ Щелкните дважды кнопку 1 (левую кнопку мыши) на иконке MAX+plus II. На компьютере с операционной системой Windows эта иконка появляется в группе программ MAX+plus II.

или

- ✓ Введите maxplus2 ↵ в командной строке. Откроется окно менеджера MAX+plus II (рис. 1.3).



Рис. 1.3

☞ Если вы не установили лицензионный файл, то диалоговое окно **License Setup** (меню **Options**) появляется автоматически. Для получения информации о том, как установить лицензионный файл для MAX+plus II, см. раздел "Установка лицензионного файла" на стр. 16.

1.4. Окно MAX+plus II Manager



Окно **MAX+plus II Manager** открывается автоматически при запуске MAX+plus II. Вы можете открыть все остальные программные модули MAX+plus II из меню MAX+plus II, как показано на рис. 1.4.

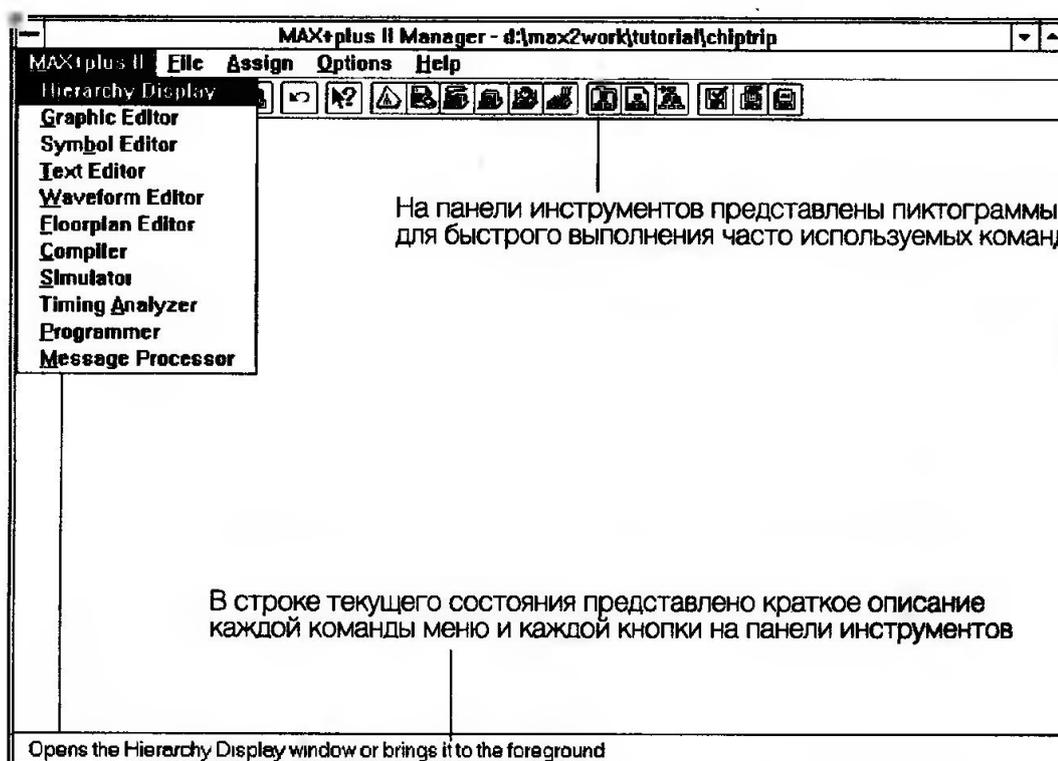


Рис. 1.4

Команды, доступные из меню **MAX+plus II Manager**, также доступны из любого другого программного модуля MAX+plus II. Они позволяют вам открыть файл, компилировать и моделировать текущий проект или переключиться на другой проект. Вы можете определить библиотеки ваших символов и проектов, архивировать резервные копии всех файлов текущего проекта в отдельной директории, подобрать комбинацию взаимодополняющих цветов для элементов экрана и ввести новый лицензионный файл. Вы также можете показать или скрыть панель инструментов и строку состояния и открыть справочную систему MAX+plus II из меню **Help**.

Кроме того, вы можете вводить, редактировать и удалять ресурсы, устройства и назначения параметров, управляющих компиляцией проекта, включая синтез логических схем, разделение проекта на части и разводку. Эти функции доступны независимо от того, открыт ли какой-либо файл проекта или окно приложения. За более подробной информацией об этих функциях обратитесь к разделу "Основные возможности ввода проекта верхнего уровня MAX+plus II" на стр. 36.

••• Для получения полной и обновленной информации по MAX+plus II Manager обратитесь к справочной системе MAX+plus II (Help).

1.5. Составные части (программные модули) MAX+plus II

Программное обеспечение MAX+plus II состоит из 11 программных модулей и MAX+plus II Manager. Одновременно могут быть активизированы несколько приложений ввода проекта, что позволяет вам переключаться между ними посредством щелчка мыши или выбора команды из меню. Также вы можете запустить один из программных модулей в фоновом режиме. Команды, совместно используемые различными программными модулями, работают одинаково, что облегчает вашу задачу проектирования на логическом уровне.

Вы легко можете свернуть окно приложения в иконку, не закрывая приложения, и позже восстановить его. Эта функция позволяет вам не загромождать экран и при этом не снижать эффективность работы.

В таблице 1.1 приведено описание программных модулей MAX+plus II и соответствующие им иконки.

Таблица 1.1

Программные модули MAX+plus II

Иконка	Программные модули
	<p>Hierarchy Display (дисплей иерархии проекта) — вывод на экран текущей иерархии файлов в виде иерархического дерева, ветвями которого являются подпроекты. Вы можете на взгляд определить является ли данный файл проекта графическим или текстовым или это проект, введенный с помощью редактора временных диаграмм, какие файлы открыты в данный момент, какие вспомогательные файлы, редактируемые пользователем, доступны для проекта. Вы также можете открыть или закрыть один или несколько файлов прямо в иерархическом дереве и ввести для них назначения ресурсов.</p>
	<p>Graphic Editor (графический редактор) — позволяет вам выполнить схемный ввод проекта. Наряду с предоставляемыми фирмой Altera базовыми элементами (primitives), мегафункциями и макрофункциями, используемыми для построения стандартных блоков, вы также можете использовать созданные вами символы функциональных узлов.</p>

Окончание табл. 1.1

Иконка	Программные модули
	<p>Symbol Editor (редактор символов) — позволяет вам редактировать существующие и создавать новые символы.</p>
	<p>Text Editor (текстовый редактор) — текстовый редактор позволяет вам создавать и редактировать текстовые файлы проекта на уровне функционально-логического описания на языке AHDL и VHDL. С помощью текстового редактора вы также можете создавать, просматривать и редактировать другие ASCII файлы, используемые различными модулями MAX+plusII. Создание AHDL и VHDL файлов посредством текстового редактора MAX+plusII по сравнению с другими текстовыми редакторами имеет ряд преимуществ: контекстно-зависимая справочная система, синтаксическая раскраска, шаблоны AHDL и VHDL.</p>
	<p>Waveform Editor (редактор временных диаграмм) — выступает в двух ролях: как инструмент ввода проекта и как инструмент для введения тестовых векторов и просмотра результатов моделирования.</p>
	<p>Floorplan Editor (редактор физического размещения) — позволяет вам назначать размещение физических контактов микросхем и ресурсов логических ячеек, используя графическую среду. Вы можете редактировать как расположение контактов, так и назначать расположение отдельных логических ячеек (LC) в логических блоках (LAB). Вы также можете просматривать результаты размещения, выполненного при последней компиляции.</p>
	<p>Compiler (компилятор) — производит логическую обработку проектов и размещение их на микросхемах семейств Altera Classic, MAX 5000, MAX 3000, MAX 7000, MAX 9000, FLEX 8000, FLEX 10K, ACEX1K. Большинство задач он выполняет автоматически. Тем не менее, вы можете задавать все или часть параметров, управляющих процессом компиляции.</p>
	<p>Simulator (симулятор) — позволяет вам тестировать логику работы и временные характеристики вашего цифрового устройства. Возможно функциональное моделирование, моделирование с учетом временных соотношений и моделирование проекта, выполненного на нескольких микросхемах.</p>
	<p>Timing Analyzer (временной анализатор) — анализирует временные характеристики вашего цифрового устройства после того, как оно синтезировано и оптимизировано компилятором.</p>
	<p>Programmer (программатор) — позволяет вам программировать микросхемы Altera и тестировать созданные устройства.</p>
	<p>Message Processor (процессор сообщений) — выводит на экран сообщения об ошибках, предупреждающие и информирующие сообщения о состоянии вашего проекта и позволяет автоматически определять местоположение источника сообщения в исходных файлах проекта, во вспомогательных файлах и в назначениях физического уровня размещения.</p>

На рис. 1.5 показан вывод на экран нескольких окон: **Hierarchy Display** и **Waveform Editor**, а также раздел описания справочной системы MAX+plus II.

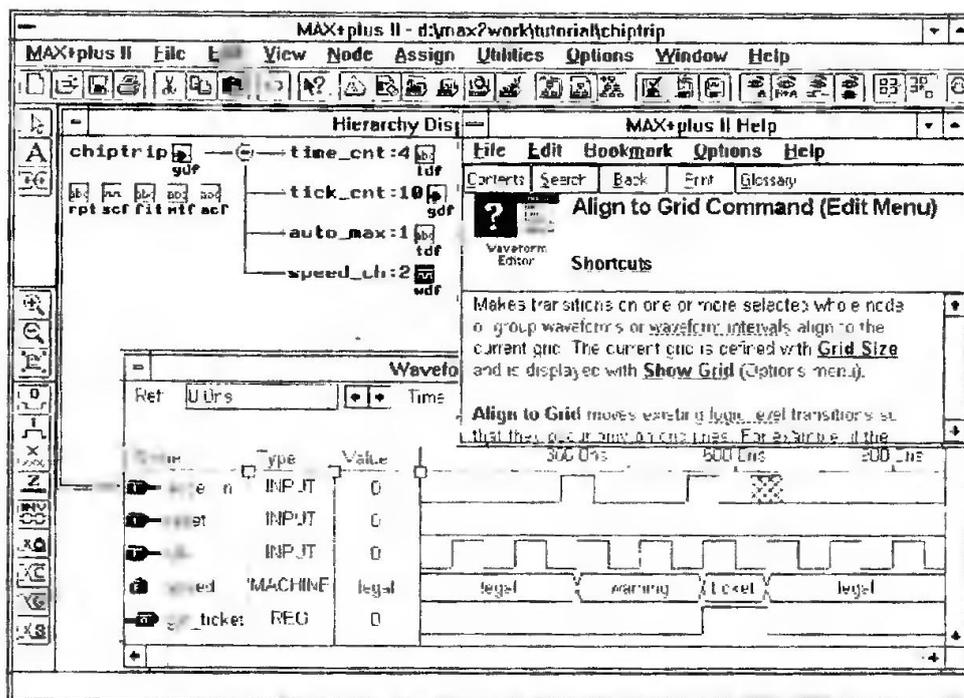


Рис. 1.5

1.6. Файлы проекта, вспомогательные файлы и проекты

До начала работы с MAX+plus II следует хорошо уяснить разницу между файлами проекта, вспомогательными файлами и проектами.

1.6.1. Файлы проектов

Файл проекта — это файл, содержащий функционально-логическое описание проекта для MAX+plus II и служащий исходным описанием для компилятора. Файл проекта может быть графическим, текстовым или файлом в виде временной диаграммы, созданным соответствующим редактором MAX+plus II или каким-либо другим стандартным графическим или текстовым редактором или генератором EDIF, VHDL файлов.

Компилятор может автоматически обрабатывать следующие файлы проекта:

- графические файлы проекта (.gdf);
- текстовые файлы проекта (.tdf);
- файлы проекта временных диаграмм (.wdf);
- VHDL файлы проекта (.vhd);
- Verilog файлы проекта (.v);
- графические файлы OrCAD (.sch);
- входные файлы EDIF (.edf);
- файлы формата списка соединений Xilinx (.xnf);

- файлы проекта Altera (.adf);
- файлы конечного автомата (.smf).

1.6.2. Вспомогательные файлы

Вспомогательные файлы — это файлы, связанные с проектом MAX+plus II, но не являющиеся частью иерархического дерева проекта. Большинство вспомогательных файлов не содержат описаний логики проекта. Некоторые из этих файлов автоматически создаются приложением MAX+plus II, некоторые вводятся пользователем. Примерами вспомогательных файлов являются файлы назначения и конфигурации (.act), символьные файлы (.sym), файлы сообщений (.rpt) и векторные файлы (.vec).

1.6.3. Проекты

Проект состоит из всех файлов в иерархии проекта, включая вспомогательные входные и выходные файлы. Проект называется так же, как и файл проекта на самом высоком уровне иерархического дерева, но без расширения имени файла. MAX+plus II осуществляет компиляцию, моделирование, временной анализ и программирование в данный момент времени только одного проекта, хотя вы всегда можете редактировать файлы, относящиеся к другому проекту. Например, по мере того, как вы компилируете **project1**, вы можете редактировать TDF файл, который является частью **project2**, и сохранить его, однако, если вы хотите его компилировать, то сначала должны определить **project2** в качестве имени текущего проекта.

Каждый проект целесообразно помещать в отдельный подкаталог.

1.7. Справочная система MAX+plus II



Справочная система MAX+plus II обеспечивает полную, обновленную документацию программного обеспечения MAX+plus II. Справочная система научит вас всему, что необходимо знать об основных инструментах программных модулей MAX+plus II, о командах, процедурах, сокращениях, золотых правилах, сообщениях, базовых элементах, мегафункциях, макрофункциях, AHDL и VHDL. Справочная система также предоставляет информацию обо всех микросхемах фирмы Altera и адаптерах, что позволяет вам выбрать подходящую микросхему до начала функционально-логического проектирования. Для получения полезной дополнительной информации справочная система укажет вам другую техническую документацию Altera и даст

советы по наиболее эффективному проектированию с помощью инструментов MAX+plus II.

Каждый раздел справочной системы содержит одно или несколько выделенных слов, называемых *переходами*, которые позволяют перейти к другим темам раздела или к дополнительной информации по текущей теме. По умолчанию переходы показаны зеленым цветом. Для просмотра темы вам нужно поместить указатель мыши на переход и щелкнуть кнопку 1 (левая кнопка мыши). Переход, подчеркнутый сплошной линией, сообщает о том, что это новая тема справочной системы. Переход, подчеркнутый пунктирной линией, сообщает о том, что это элемент словаря. Голубой переход сообщает о примере, списке горячих клавиш и иконок или иллюстрации на самом высоком уровне данной темы справочной системы. Когда вы снова щелкните кнопку 1 мыши, появившаяся тема закроется.

- Для получения информации о механизме использования справочной системы (например, копировании или печати тем справочной системы) щелкните на **How to Use Help** (меню **Help**).

1.7.1. Меню Help

Строка меню каждого программного модуля MAX+plus II обеспечивает доступ к меню **Help** (меню справочной системы), как показано на рис. 1.6.

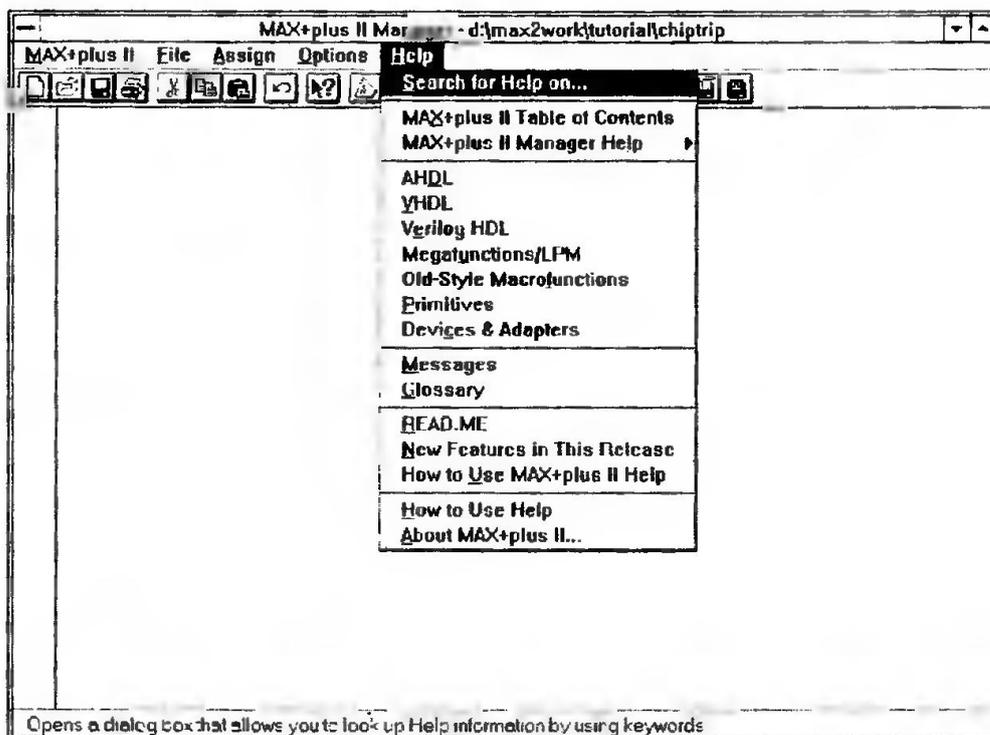


Рис. 1.6

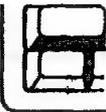
В табл. 1.2 приведено описание всех пунктов меню справочной системы (**Help**) и, где они есть, показаны соответствующие им иконки.

Таблица 1.2

Пункты меню справочной системы MAX+plus II

Пункт меню	Иконка
Search for Help on (поиск по справочной системе) — открывает диалоговое окно, названное Search (поиск), которое позволяет быстро отыскать нужную информацию по обширному индексу справочной системы. Вы можете выбрать слово или фразу, которую вы хотите найти. Как только вы начнете набирать буквы, окно списка автоматически прокрутит слова или фразы, наиболее близкие к тем, которые вы набираете. Далее вы можете составить список подходящих вам тем справочной системы и перейти к теме, которую вы хотите просмотреть.	—
MAX+plus II Table of Contents (таблица содержания MAX+plus II) — полная таблица содержания, в которой перечислены все главные темы, представленные в справочной системе MAX+plus II. К ней также можно получить доступ через кнопку Contents (содержание) на панели кнопок в верхней части окна справочной системы.	—
<Application name> Help (справочная система по <Имя программного модуля>) — открывает подменю по темам справочной системы для текущего программного модуля MAX+plus II:	В табл. 1.1 показаны иконки всех приложений
Table of Contents (таблица содержания) — таблица содержания для текущего программного модуля	—
Introduction (введение) — краткий обзор текущего программного модуля, включая иллюстрации и информацию о том, как начать работу с использованием этого модуля.	
Basic Tools (основные инструменты) — подробное описание элементов, находящихся в окне программного модуля, а также входных и выходных файлов, сопровождаемое иллюстрациями и примерами. В категории основных инструментов в справочной системе также представлена информация о базовых элементах и макрофункциях, о кнопках, полях, и конках и т. д.	
Commands (команды) — описание каждой из команд текущего программного модуля, сопровождаемое иллюстрациями и примерами. Иллюстрации диалоговых окон команд включают появляющиеся объяснения для каждой опции и кнопки диалогового окна.	
Procedures (процедуры) — пошаговые инструкции, сопровождаемые иллюстрациями и примерами решения специфических задач в текущем программном модуле.	
Golden Rules (золотые правила) — подборка важнейших советов и правил по использованию текущего программного модуля.	

Продолжение табл. 1.2

Пункт меню	Иконка
<p>Shortcuts (горячие клавиши и иконки) — последовательность нажатия клавиш клавиатуры или мыши, иконок панели инструментов для быстрого выполнения команд и процедур текущего программного модуля.</p>	
<p>AHDL — справочная система по AHDL, включающая подробные инструкции по разработке проекта, описанию основных элементов структуры проекта, синтаксиса AHDL и руководство по стилю проектирования.</p>	
<p>VHDL — справочная система по VHDL, включающая инструкции по разработке проекта с помощью MAX+plus II VHDL, описания поддерживаемых конструкций языка VHDL, синтаксиса VHDL и руководство по стилю проектирования.</p>	
<p>Verilog HDL — справочная система по Verilog HDL, включающая инструкции по разработке проекта с помощью MAX+plus II Verilog HDL, описания поддерживаемых конструкций языка Verilog HDL, синтаксиса и руководство по стилю проектирования.</p>	
<p>Megafunctions/LPM (мегафункции/БГМ) — список мегафункций, из Библиотеки Параметризованных Модулей (LPM) и функции Altera MegaCore/OpenCore. Если вы выбираете конкретную мегафункцию, то выводятся на экран ее описание, правила обращения, AHDL Function Prototype (AHDL прототип функции), VHDL Component Declaration (средства включения в описания на языке VHDL) и информация о необходимых для этой мегафункции ресурсах микросхемы.</p>	<p>—</p>
<p>Old-Style Macrofunctions (макрофункции микросхем серии 74XXXX) — перечень операционных узлов – функциональных аналогов микросхем серии 74XXXX. Вы можете выбрать один из перечисленных узлов для вывода на экран всех названий макрофункций, которые его реализуют. Если вы выбираете конкретную макрофункцию, то будут выведены на экран ее описание и логические уровни сигналов по умолчанию.</p>	<p>—</p>
<p>Primitives (базовые элементы) — алфавитный список базовых элементов. Если вы выберете конкретный базовый элемент, то будут выведены на экран его описание и функциональная таблица.</p>	<p>—</p>
<p>Devices & Adapters (микросхемы и адаптеры) — список всех текущих микросхем фирмы Altera, поддерживаемых MAX+plus II, и адаптеров для их программирования. Также включены руководства по выбору микросхем для каждого семейства фирмы Altera. Вы можете выбрать одну из микросхем, перечисленных в списке, для вывода на экран ее характеристик, информации о расположении логических ячеек и назначении контактов для каждого типа корпуса.</p>	
<p>Messages (сообщения) — алфавитный список всех сообщений MAX+plus II: информационных, сообщений-предупреждений и сообщений об ошибках. Все сообщения сопровождаются подробными объяснениями возможных причин их появления и рекомендуемыми действиями. В объяснениях сообщений также предусмотрены переходы к дополнительной полезной информации.</p>	

Окончание табл. 1.2

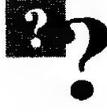
Пункт меню	Иконка
Glossary (словарь) — полный список терминов MAX+plus II и их определений.	
READ.ME (прочти) — копия файла read.me, предоставляемого с MAX+plus II. В ней находится информация о системных требованиях, известных проблемах и способах их решения.	—
New Features in This Release (новые возможности данной версии) — описание всех новых возможностей в текущей версии MAX+plus II, включая поддержку новых микросхем и интерфейсы с другими средствами проектирования стандарта CAE.	
How to Use Help (как пользоваться справочной системой) — информация о механизмах использования приложений справочной системы Windows.	—
How to Use MAX+plus II Help (как пользоваться справочной системой MAX+plus II) — подробная информация о том, как пользоваться справочной системой MAX+plus II, включая описание пунктов меню справочной системы и документацию по условным обозначениям.	
About MAX+plus II (о MAX+plus II) — отображает номер версии MAX+plus II, номер версии текущего приложения, информацию об авторских правах.	—

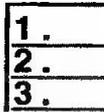
Таблица 1.3

Кнопочная панель окна справочной системы

Название	Функция
Contents (содержание)	Показывает таблицу содержания справочной системы MAX+plus II.
Index (индекс)	Открывает диалоговое окно Help Topics (темы справочной системы),
Back (вернуться)	Возвращает ранее просмотренную информацию, т.е. вы восстанавливаете ваш путь по темам, которые вы уже видели. Кнопка серая, если предыдущей темы нет.
Print (печать)	Выводит на печать одну или несколько копий текущей темы справочной системы.
Glossary (словарь)	Показывает список терминов MAX+plus II и их определения. Вы можете напечатать любой элемент словаря, который вы откроете в этом списке.

1.7.2. С чего начать знакомство со справочной системой?

Справочная система универсальна. Это позволяет вам решить, каким способом вы хотите изучать MAX+plus II. Однако, если вы начинающий пользователь, то для вас самым эффективным может оказаться следующий подход:

- Шаг 1:** Выберите команду **MAX+plus II Table of Contents** (таблица содержания MAX+plus II) из меню **Help** (справочная система) любого программного модуля. Это окно показывает все программные модули и другие темы, включая представляющие их иконки, о которых доступна информация в справочной системе.
- Шаг 2:**  Выберите **Introduction to MAX+plus II** (введение в MAX+plus II) из темы **MAX+plus II Table of Contents**. Этот обзор суммирует возможности MAX+plus II и подсказывает отправные пункты **Where to Start?** (с чего начать?) для знакомящихся с MAX+plus II.
- Шаг 3:**  Из **Introduction to MAX+plus II** (введение в MAX+plus II) вы можете выбрать **How to Use MAX+plus II Help** для получения исходной информации о том, как организована справочная система и какие в ней используются условные обозначения.
- Шаг 4:**  Из меню **Help** (справочной системы) в любом программном модуле выберите **<Application name> Help Procedures** для просмотра списка всех пошаговых процедур, которые вы можете использовать в текущем программном модуле.
- Шаг 5:**  Из меню **Help** (справочной системы) в любом программном модуле выберите **<Application name> Help Golden Rules** для того, чтобы узнать самые важные советы и руководящие указания для текущего программного модуля. Золотые правила позволяют вам максимально использовать преимущества MAX+plus II при создании цифровых устройств с его помощью, предоставляя в сжатом виде самую важную информацию, доступную в остальных темах справочной системы.

 *Каждый программный модуль MAX+plus II имеет свое введение и золотые правила.*

1.7.3. Как сделать запрос справочной системе по конкретной теме?

MAX+plus II предоставляет справку как на основе меню, так и контекстно-зависимую справку. Вы можете запросить справку с помощью мыши или с клавиатуры различными способами:

- В каждом программном модуле MAX+plus II есть меню справочной системы, которое направляет вас к информации о MAX+plus II в общем или к информации, связанной с конкретным программным модулем. Например:
 - для того чтобы открыть индекс MAX+plus II: выберите **Search for Help on** (поиск по справочной системе) из меню **Help**. В открывшемся диалоговом окне выберите с помощью прокрутки или наберите нужное ключевое слово, дважды щелкните кнопку 1

на нем для вывода списка всех связанных с ним тем справочной системы, затем дважды щелкните кнопку 1 на заголовке темы для открытия справки по данной теме. Вам не нужно набирать целые слова, поскольку те буквы, которые вы набираете сопоставляются с перечисленными ключевыми словами;
для того чтобы узнать комбинацию клавиш для быстрого вызова команды: перейдите к теме справочной системы, описывающей эту команду, используя **Search for Help on** (поиск по справочной системе) и щелкните кнопку 1 на голубом переходе **Shortcuts** на заголовке этой темы или перейдите к меню справочной системы и выберите **Shortcuts** из подменю **<Application Name>Help** для вывода на экран таблицы всех горячих клавиш и иконок для текущего программного модуля.

- Когда вы выбираете контекстно-зависимую кнопку справочной системы () на панели инструментов или нажимаете Shift+F1, указатель превращается в вопросительный знак. После этого вы можете щелкнуть кнопку 1 на любом элементе окна или на любой команде меню. Если для данного элемента есть контекстно-зависимая справка, то на экран будет выведена соответствующая информация. Иначе, справочная система покажет список всех элементов, для которых есть контекстно-зависимая справка.
- Когда выделена команда меню или же на экране отображается диалоговое окно команды или сообщение, то нажмите F1 для получения справки по данной теме. Например:
 - для получения мгновенной информации о команде: выделите команду в меню и нажмите F1;
 - для того чтобы увидеть все элементы, для которых есть контекстно-зависимые справки в программном модуле: нажмите F1. Справочная система MAX+plus II покажет список всех элементов данного программного модуля, для которых есть контекстно-зависимая справка. Щелкните кнопку 1 на нужном элементе.

1.8. Ввод проекта

Все инструменты, необходимые для создания проекта, доступны в MAX+plus II. MAX+plus II ускоряет ввод вашего проекта с помощью набора стандартных логических функций, включая базовые элементы, мегафункций, LPM функций и макрофункций микросхем серии 74XXXX. САПР MAX+plus II также обеспечивает множество основных и дополнительных возможностей, которые облегчают ввод и отладку проекта. В САПР MAX+plus II представлены три редактора ввода проекта — **Graphic Editor** (графический редактор), **Text Editor** (текстовый

редактор) и **Waveform Editor** (редактор временных диаграмм). Также представлены два вспомогательных редактора **Floorplan Editor** (редактор физического размещения) и **Symbol Editor** (редактор символов), облегчающие ввод проекта.

MAX+plus II поддерживает различные способы ввода проекта:

- схемные проекты вводятся с помощью MAX+plus II **Graphic Editor** (графического редактора MAX+plus II). Вы также можете открыть, редактировать и сохранить схемы, созданные с помощью редактора схем OrCAD Draft;
- проекты на языке описания аппаратных средств AHDL, VHDL и Verilog HDL вводятся с помощью MAX+plus II **Text Editor** (текстового редактора MAX+plus II) или других стандартных текстовых редакторов;
- проекты, представленные временными диаграммами, вводятся с помощью MAX+plus II **Waveform Editor** (редактора временных диаграмм MAX+plus II);
- **EDIF netlist** и **Xilinx netlist** файлы, сформированные другими САПРами промышленного стандарта EDA, могут быть импортированы в среду MAX+plus II;
- схемные и текстовые файлы проектов, созданные с помощью предыдущих версий САПР фирмы Altera, также могут быть интегрированы в среду MAX+plus II;
- физические ресурсы микросхемы (внешние контакты, ячейки ввода-вывода, внутренние ячейки) для всех типов проектных файлов в текущем проекте могут быть заданы в графическом виде с помощью **Floorplan Editor** (редактор физического размещения). Редактор физического размещения сохраняет назначения для проекта в файле с расширением **.act**, наряду с параметрами настройки для **Compiler** (компилятора), **Simulator** (симулятора) и **Timing Analyzer** (временного анализатора);
- графические символы, представляющие любой тип проектных файлов, могут быть сформированы автоматически в любом из редакторов проекта MAX+plus II. Вы можете редактировать имеющиеся или создавать свои собственные символы с помощью **Symbol Editor** (редактора символов) и использовать их в любом схемном проектном файле.

В иерархическом проекте вы можете на любом уровне иерархии легко смешивать Graphic Design Files (графические файлы проекта) с расширением (**.gdf**), Text Design Files (текстовые файлы проекта) с расширением (**.tdf**), VHDL Design Files (VHDL файлы проекта) с расширением (**.vhd**), Verilog Design Files (Verilog файлы проекта) с расширением (**.v**), EDIF Input Files (входные файлы EDIF) с расширением (**.edf**) и OrCAD Schematic Files (схемные файлы OrCAD) с расширением (**.sch**). Однако

Waveform Design Files (файлы проектов, описанных временными диаграммами) с расширением (.wdf), Xilinx Netlist файлы с расширением (.xnf), Altera Design файлы с расширением (.adf) и State Machine Files (файлы конечных автоматов) с расширением (.smf) не могут быть файлом высшего уровня проекта. См. "Иерархия проекта" на стр. 60. Способы ввода проектов в MAX+plus II показаны на рис 1.7.

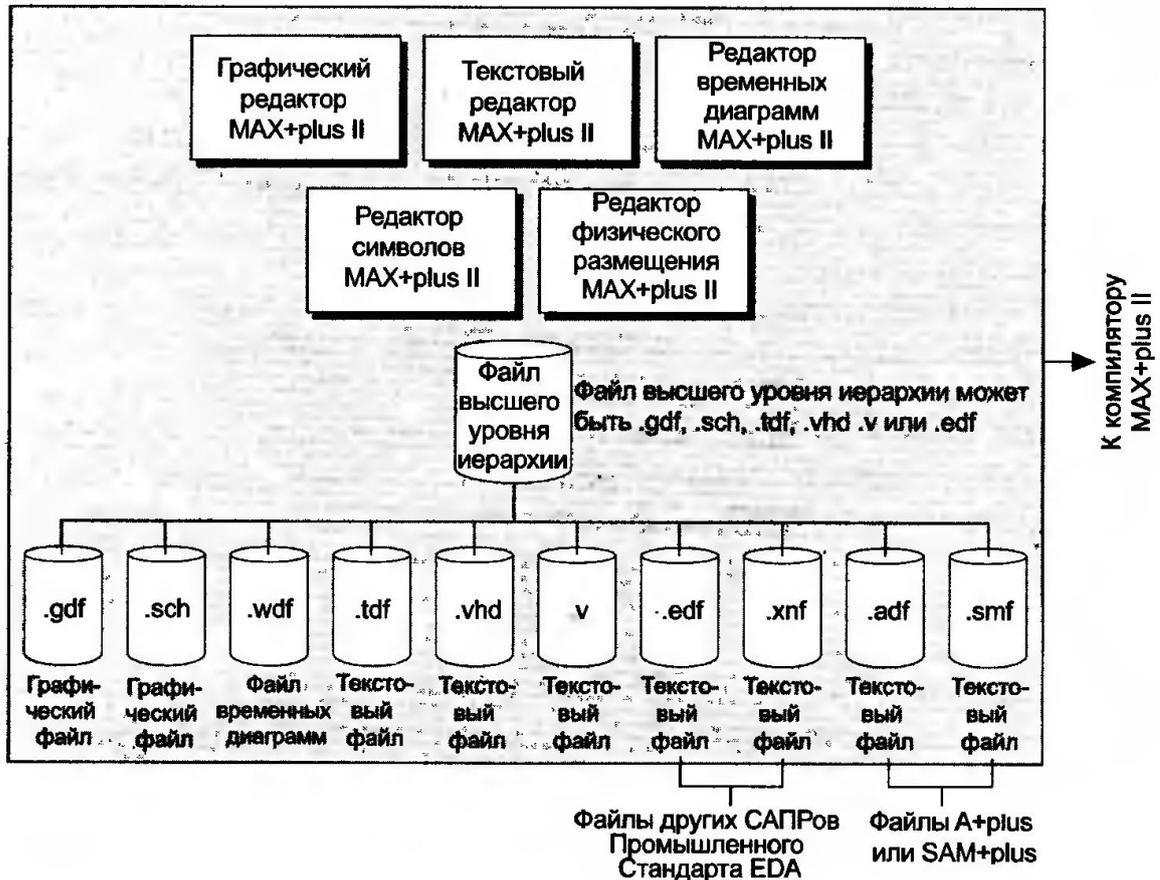


Рис. 1.7

1.8.1. Основные возможности ввода проекта верхнего уровня MAX+plus II

Во всех программных модулях MAX+plus II вы можете задавать ресурсы, микросхемы и настройки параметров, которые управляют процессом компиляции, включая функционально-логический синтез, разделение проекта на части и сборку, пользуясь командами меню **Assign** (назначение).

На рис. 1.8 показано меню **Assign** (назначение). Вы можете ввести назначения для текущего проекта независимо от того, открыт ли какой-либо файл проекта или окно программного модуля. MAX+plus II сохранит информацию для проекта в файле **Assignment & Configuration** (назначения и конфигурация) с расширением .acf. Назначения, сделанные в окне **Floorplan Editor** (редактора физического размещения) также

сохранятся в .acf файле. Кроме того, вы можете редактировать .acf файл проекта вручную с помощью **Text Editor** (текстового редактора).

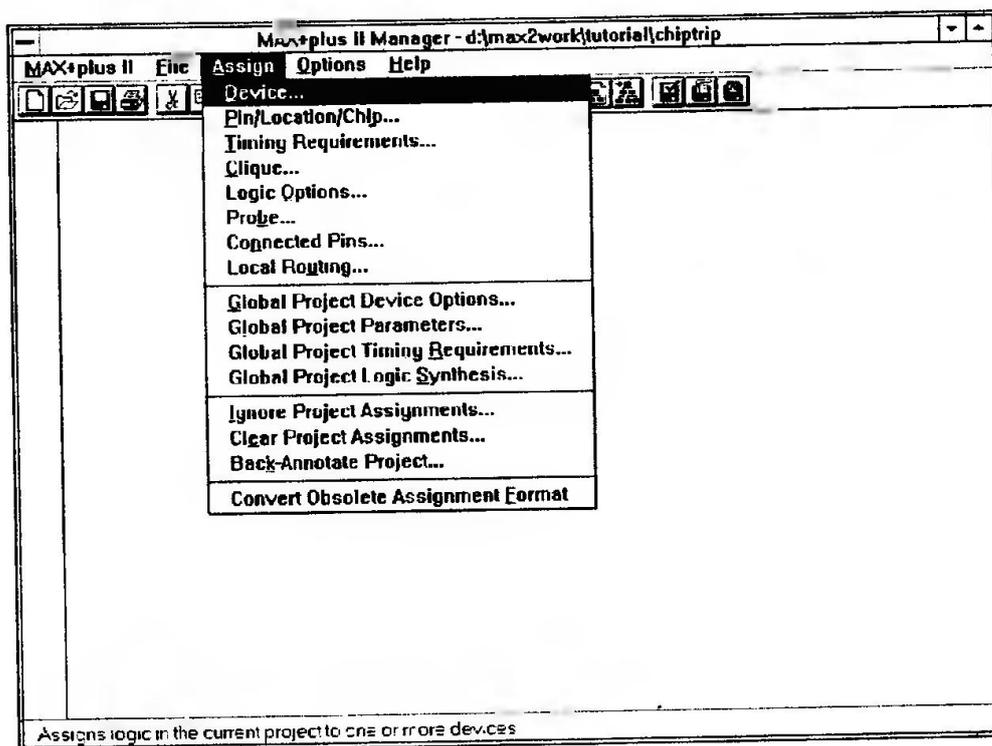


Рис. 1.8

Ниже приведены функции, которые являются общими для всех программных модулей MAX+plus II:

- *Device, resource, and probe assignments* (назначения микросхемы, ресурсов и синонима имени цепи);
- *Back-annotation* (копирование результатов компиляции);
- *Global project device options* (параметры микросхемы для текущего проекта);
- *Global project parameters* (назначение параметров для текущего проекта);
- *Global project timing requirements* (требуемые характеристики быстродействия для текущего проекта);
- *Global project logic synthesis* (параметры логического синтеза для текущего проекта).

Device, resource, and probe assignments (назначения микросхемы, ресурсов и синонима имени цепи)

Управляемыми ресурсами микросхемы фирмы Altera являются выводы СБИС и логические ячейки, которые выполняют определенные пользователем задачи. Вы можете сами назначить различные режимы работы ресурсам микросхемы, и компилятор MAX+plus II реализует именно тот режим, который вам нужен. Возможны следующие назначения:

- *Clique assignment* (назначение группы) — устанавливает, какие функциональные модули должны оставаться вместе в процессе синтеза, разводки и других операций, выполняемых компилятором. Объединение функциональных модулей в группу гарантирует, что они будут размещены в одном Logic Array Block (LAB) (логическом блоке), Embedded Array Block (EAB) (встроенном блоке), в одной и той же строке LAB;
- *Chip assignment* (назначение кристалла) — устанавливает, какие функциональные модули должны выполняться в каких микросхемах, когда проект разделен на несколько микросхем;
- *Pin assignment* (назначение выводов СБИС) — указывает размещение входов или выходов на конкретном контакте микросхемы;
- *Location assignment* (назначение местоположения) — назначает размещение отдельных элементов проекта, например, базового элемента или мегафункции в конкретном месте внутри кристалла, например, в определенных логических ячейках, I/O ячейках (ячейках ввода-вывода), во встроенных ячейках, в LAB, EAB, в определенной строке или колонке;
- *Probe assignment* (назначение синонима имени цепи) — назначает легкое для запоминания, уникальное имя для внутреннего сигнала проекта;
- *Connected pin assignment* (назначение соединенных контактов) — определяет, каким образом два или более контактов соединяются внешне на вашей монтажной плате. Эта информация также полезна для временного моделирования и моделирования множества взаимосвязанных проектов;
- *Local routing assignment* (размещение в смежных логических блоках) — назначает соединение указанных пользователем цепей через локальные шины соединений (действует только для FLEX6000);
- *Device assignment* (назначение микросхемы) — назначает микросхему для реализации проекта. Вы можете назначить конкретную микросхему с требуемым типом корпуса, быстродействием и рабочим диапазоном температур. Вы также можете, установив "AUTO" режим, дать возможность компилятору выбрать микросхему из указанного вами семейства микросхем;
- *Logic option assignment* (назначение параметров логического синтеза) — управляет логическим синтезом отдельных частей проекта во время компиляции. Задается конкретный тип логического синтеза и/или отдельные установки.
Altera предоставляет большое число логических опций, а также тройку "готовых" типов синтеза, каждый из которых представляет собой набор заданных параметров логического синтеза, объединенных своим названием. Вы можете использовать эти типы

синтеза или создавать свои собственные под конкретную задачу. Это позволяет вам разрабатывать ваши типы синтеза для конкретного семейства микросхем, чтобы воспользоваться особенностями их архитектуры;

- *Timing assignment* (временное назначение) — управляет функционально-логическим синтезом и размещением отдельных функциональных модулей проекта для достижения желаемой задержки распространения сигнала от входа до выхода t_{PD} , задержки на выходе по отношению к синхронизирующему сигналу t_{CO} , времени предустановки для синхронизируемого сигнала t_{SU} и частоты синхронизирующего сигнала f_{MAX} .

••• За полной информацией по всем микросхемам обратитесь к спецификациям для отдельных микросхем на текущем CD Altera Digital Library.

Обратитесь к разделу **Devices & Adapters** в справочной системе MAX+plus II по вопросам о расположении контактов для всех имеющихся в настоящее время корпусов микросхем фирмы Altera.

Back-annotation (копирование результатов компиляции)

После того, как вы ввели ваш проект в целом, вы можете его компилировать с помощью MAX+plus II, затем сохранить, т. е. обратно аннотировать назначения ресурсов, которые компилятор сделал во время самой последней компиляции для того, чтобы вы могли получить такую же разводку при последующих компиляциях проекта.

Global project device options (параметры микросхемы для текущего проекта)

Вы можете задать общие функции микросхемы, которые компилятор будет учитывать при обработке проекта. Вы можете зарезервировать часть пользовательских выводов и логических ячеек для дальнейшего расширения проекта, задать настройки для функций микросхемы и для контактов двойного назначения. Например, вы можете задать установку Security Bit (бита защиты), который защищает микросхемы, выполненные по EPROM или EEPROM технологии от считывания.

Global project parameters (назначение параметров для текущего проекта)

Вы можете задать имена и значения параметров во всех параметризованных функциях вашего проекта для использования их компилятором.

Global project timing requirements (требуемые характеристики быстродействия для текущего проекта)

Вы можете ввести общие временные требования для всего проекта, такие как задержка распространения сигнала от входа до выхода t_{PD} , задержка сигнала на выходе относительно синхронизирующего сигнала t_{CO} , время предустановки относительно синхронизирующего сигнала t_{SU} и частота синхронизирующего сигнала f_{MAX} . Вы также можете задать условия подсчета этих временных характеристик.

Global project logic synthesis (параметры логического синтеза для текущего проекта)

Вы можете выбрать общие для всех блоков проекта настройки логического синтеза, управляющие компиляцией проекта. Вы можете задать один из стандартных типов синтеза, оптимизировать проект по скорости или аппаратным затратам, разрешить компилятору автоматическое назначение таких сигналов, как *Clock*, *Clear*, *Preset*, *Output*. Вы также можете задать компилятору стандартный или многоуровневый синтез, указать варианты автоматического кодирования состояний конечного автомата, разрешить раздельное использование логической и регистровой частей логической ячейки (LC) (функция *Register Packing*). Кроме того, вы можете установить быстродействие логических ячеек и I/O ячеек (ячеек ввода/вывода), задать размещение операционных узлов во встроенных блоках памяти (EAB).

1.8.2. Общие функции редакторов

Многие функции используются совместно всеми пятью редакторами MAX+plus II или тремя редакторами ввода проекта (графическим, текстовым и редактором временных диаграмм), облегчая вам работу по проектированию логики. Например, стандартные функции, такие как сохранение и открытие файла, возможны во всех редакторах проектирования.

Кроме того, общими для программных модулей редакторов MAX+plus II являются приведенные ниже функции.

Symbol & Include File Generation (создание символа и соответствующего ему файла включения)

Вы можете автоматически создать и обновить символ для любого типа проектного файла, поддерживаемого MAX+plus II, с помощью команды **Create Default Symbol** (создать символ по умолчанию) из меню **File** (файл). **Symbol File** (файл символа) носит такое же имя,

что и файл проекта, и имеет расширение **.sym**. Он может быть включен в состав любого GDF или OrCAD Schematic File (схемного файла) с расширением **.sch**, который находится выше в иерархии проекта. Вы также можете модифицировать созданный символ.

Аналогично созданию символа вы можете автоматически создавать и обновлять Include File (файл включения), содержащий AHDL Function Prototype (AHDL прототип функции), для любого **.tdf** файла проекта MAX+plus II. Команда **Create Default Include File** (создать файл включения) из меню **File** создает Include File (файл включения) с тем же именем, что и имя файла проекта, но с расширением **.inc**. Вы можете использовать AHDL Include Statement для связывания Include File (файла включения) и его Function Prototype (прототип функции) в TDF иерархического проекта. Компилятор MAX+plus II также использует информацию в **Include File** для обработки их копий в Verilog Design Files (Verilog файлах проекта) с расширением (**.v**).

Node Location (определение места расположения цепи)

Вы можете выбрать цепь во вспомогательном файле или в текущем физическом размещении и определить место ее описания в исходный файл проекта с помощью команды **Find Node in Design File** (найти цепь в файле проекта) из меню **Utilities**. Аналогично вы можете выбрать цепь или группу в файле проекта или во вспомогательном файле и определить место ее расположения в окне редактора физического размещения с помощью команд из меню **Utilities Find Node in Floorplan** (найти цепь в топологической структуре) или **Find Clique in Floorplan** (найти группу в топологической структуре). Команды **Find Node in Floorplan** (найти цепь в топологической структуре) и **Find selected macrofunction in Floorplan** (найти группу в топологической структуре) также доступны в **Hierarchy Display** (дисплее иерархии проекта).

Hierarchy Traversal (прохождение по дереву иерархии проекта)

Вы можете перемещаться вниз или вверх в текущей иерархии. MAX+plus II открывает выбранные файлы и автоматически запускает соответствующий редактор.

Context-Sensitive Menu Commands (команды контекстно-зависимого меню)

Команды контекстно-зависимого меню доступны на появляющихся меню во всех редакторах и в **Hierarchy Display** (дисплее иерархии проекта). Эти меню появляются автоматически, когда вы щелкаете кнопку 2 (правую кнопку мыши) на выбранном с помощью курсора

объекте. Меню будет содержать команды, относящиеся к выбранному объекту или объектам.

Timing Analysis (анализ временных параметров)

Вы можете пометить источники и приемники сигналов и определить задержки их прохождения, времена предустановки и удержания, максимальную тактовую частоту для каждого синхросигнала в проекте. Вы также можете выделять по одному сигналу для детального анализа пути его прохождения.

Find & Replace Text (найти и заменить текст)

Вы можете найти и заменить текст, включая имена цепей и их синонимов, в любом файле проекта текущей иерархии.

Undo, Cut, Copy, Paste & Delete (отменить, вырезать, копировать, вставить и удалить)

Вы можете отменить последний шаг редактирования и отменить отмену. Вы можете также копировать, вырезать, вставлять и удалять одну или несколько выделенных позиций в другие программные модули MAX+plus II или приложения Windows.

Print (печать)

Вы можете напечатать текущий файл частично или целиком, выбрать принтер или плоттер и определить их конфигурацию.

1.8.3. Графический редактор MAX+plus II



Графический редактор MAX+plus II (**Graphic Editor**), показанный на рис. 1.9, предоставляет среду проектирования: что видишь на экране, то и получишь в проекте. Открыть окно графического редактора вы можете путем выбора **Graphic Editor** (графический редактор) из меню MAX+plus II. Команда **New** (новое) из меню **File** (файл) открывает новое, без названия, окно графического редактора.

Графический редактор представляет собой программу ввода схемы, позволяющую вам быстро и легко вводить даже сложные проекты. Обширная библиотека базовых элементов, мегафункций и макрофункций, включая Library of Parameterized Modules (LPM) (библиотеку параметризованных модулей), предоставляет основные функциональные блоки для реализации проекта. Создавая символы, вы можете строить ваши собственные библиотеки специальных функций.

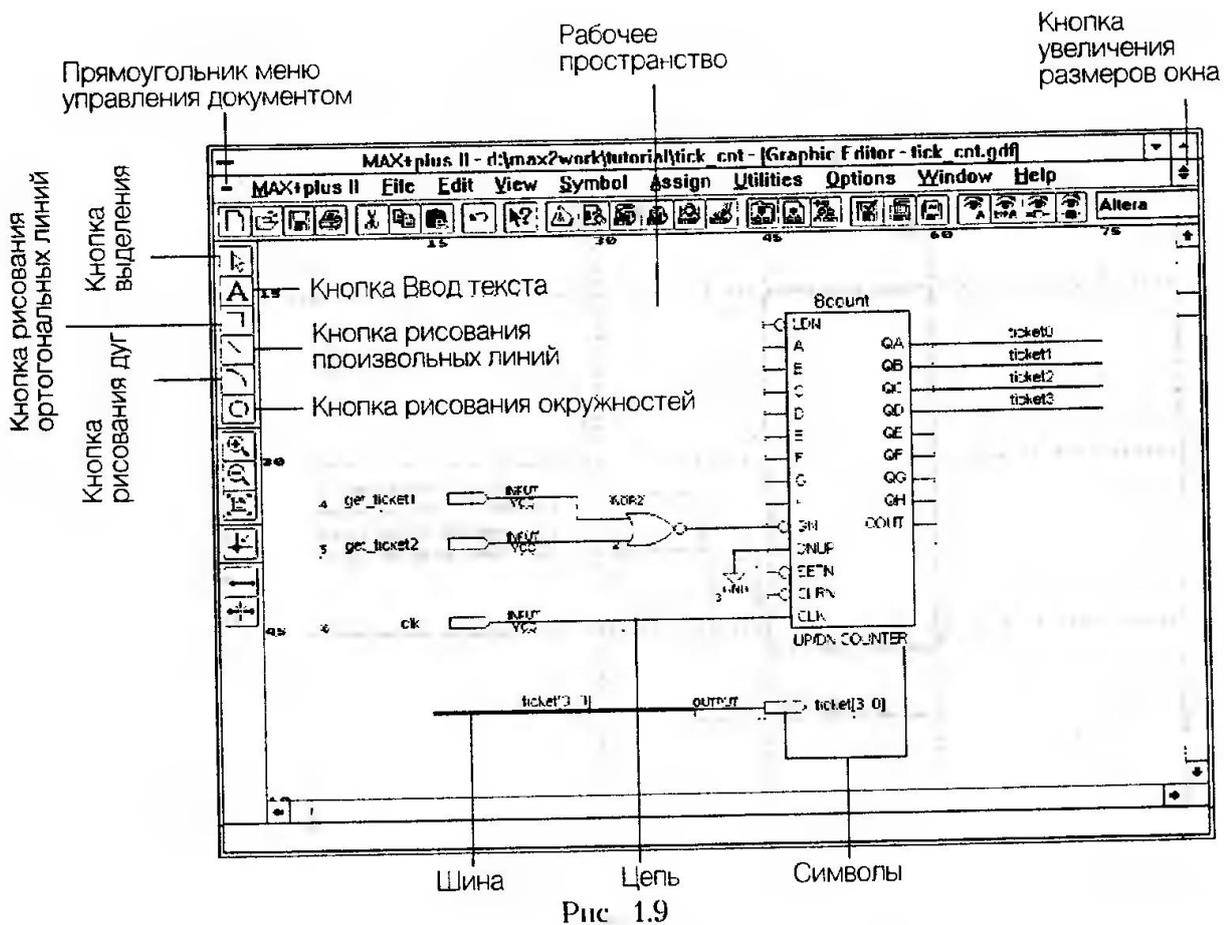


Рис. 1.9

Графический файл проекта (.gdf) или схемный файл OrCAD (.sch), созданные с помощью графического редактора, могут включать любую комбинацию символов базовых элементов, мегафункций и макрофункций, символов ваших собственных библиотек. Символы могут представлять любой тип проектного файла, включая другие GDF и OrCAD схемные файлы, AHDL текстовые файлы проектов (.tdf), VHDL файлы проектов (.vhd), Verilog файлы проектов (.v), файлы проектов временных диаграмм (.wdf), EDIF входные файлы (.edf), файлы формата списка соединений Xilinx (.xnf), файлы проектов Altera (.adf) и файлы конечных автоматов (.smf).

Графический редактор имеет следующие особенности:

- кнопка выделения, показанная на рис. 1.9, облегчает ввод проекта. Она позволяет вам перемещать и копировать элементы, вводить новые символы. Если вы поместили курсор на основание контакта или конец линии, автоматически включается режим рисования ортогональных линий. Когда вы щелкаете на текст, например, название контакта или цепи, автоматически включается режим ввода текста;
- символы соединяются друг с другом тонкими линиями, называемыми цепями, или толстыми линиями, называемыми шинами. Шины представляют собой совокупность цепей. Присвоив имя цепи, вы

тем самым соединяете ее с другими цепями или символами, имеющими такое же имя. Шины соединяются с помощью имен. Их графическое соединение является вспомогательным;

- если это необходимо в вашем проекте, вы можете проинвертировать любой бит порта символа мегафункции или макрофункции. "Кружок NOT" (символ операции НЕ) появляется автоматически для индикации инвертированного порта;
- вы можете выбрать и редактировать множество объектов в выделенной вами прямоугольной области. Когда вы перемещаете выбранные элементы при включенной функции Rubberbanding (неразрывность цепей) сигнальные связи сохраняются;
- для каждого символа вы можете просмотреть все назначения и параметры. Для облегчения моделирования вы также можете назначить группы контактов, соединенных внешне на плате;
- базовые элементы, мегафункции и макрофункции, предоставляемые фирмой Altera, уменьшают время ввода проекта. Вы также можете создать свои собственные библиотеки специальных функций. После редактирования символов вы можете автоматически обновить выбранный символ или все символы в файле графического редактора.

Графический редактор MAX+plus II предоставляет множество других возможностей. Например, вы можете увеличивать или уменьшать масштаб изображения на экране таким образом, чтобы на экран помещался весь файл проекта или же только его часть. Вы также можете выбрать различные шрифты для текста, размеры шрифтов, типы линий и расстояние между линиями разметки. Вы можете копировать, вырезать, вставлять и удалять один или несколько выбранных элементов, изменять их расположение зеркально относительно горизонтали или вертикали, поворачивать их на 90, 180 или 270 градусов, задавать размер и ориентацию листа чертежа.

 *За полной информацией о функциях и возможностях графического редактора MAX+plus II обратитесь к справочной системе MAX+plus II.*

1.8.4. Редактор символов MAX+plus II



Редактор символов MAX+plus II, показанный на рис. 1.10, дает вам возможность просматривать, создавать и редактировать символ, который отображает цифровое устройство. Открыть окно редактора символов вы можете путем выбора **Symbol Editor** (редактор символов) из меню MAX+plus II. Команда **New** (новое) из меню **File** (файл) открывает новое, без названия, окно редактора символов.

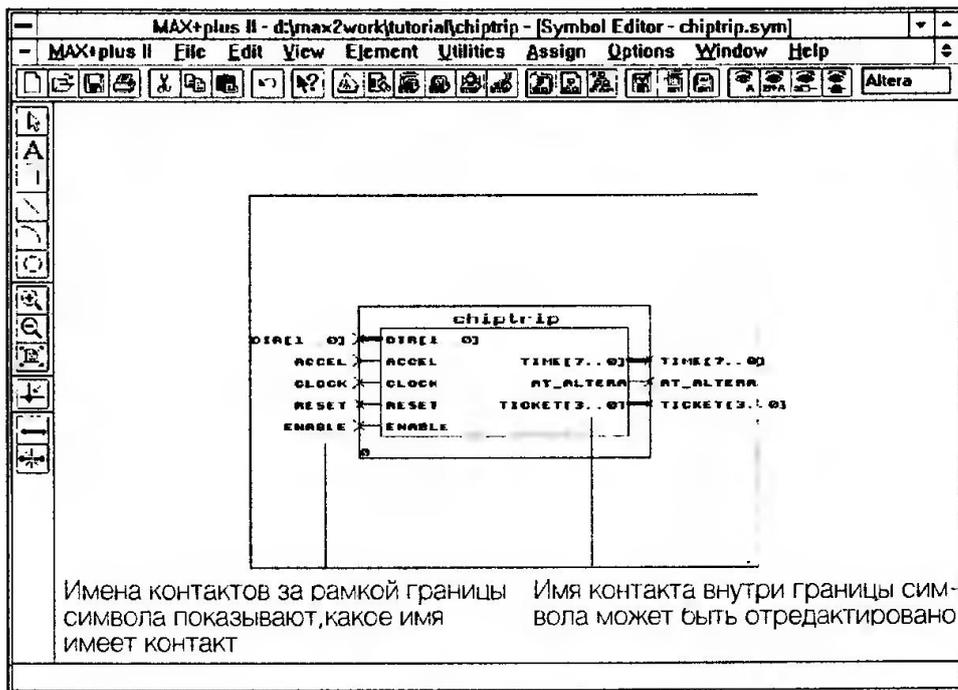


Рис. 1.10

Файл символа называется так же, как и файл проекта, который он представляет, но с расширением **.sym**. команда **Create Default Symbol** (создать символ по умолчанию), доступная из меню **File** (файл) графического, текстового и редактора временных диаграмм, создает символ для любого проектного файла.

Основные возможности символьного редактора MAX+plus II:

- вы можете настроить символ, представляющий файл проекта. Вы можете вводить и задавать параметры контактов и их имена. Вы можете выбрать для отображения полное имя контакта в символе или изменить его, сделав более компактным или информативным. Полное название контакта и название, отображаемое в файле графического редактора, могут различаться;
- имена контактов автоматически копируются за границей символа для того, чтобы у вас была видимая ссылка на название, которое соответствует данному контакту. Если вы передвигаете название контакта внутри символа, идентичное название за символом, которое не может быть передвинуто относительно контакта, помогает вам отслеживать соответствие контакта и его названия;
- вы можете задать параметры и их значения по умолчанию;
- линии разметки помогают вам расположить объекты точно по сетке;
- вы можете вставлять комментарии или полезные замечания в символ. Они также будут появляться при вводе символа в файл графического редактора.

- За полной информацией о функциях и возможностях символьного редактора MAX+plus II обратитесь к справочной системе MAX+plus II.

1.8.5. Текстовый редактор MAX+plus II



Текстовый редактор MAX+plus II, показанный на рис. 1.11, является гибким инструментом для создания текстовых файлов проекта (.tdf) на языке AHDL (язык описания аппаратных средств фирмы Altera), VHDL файлов проектов (.vhd) на языке VHDL (язык описания аппаратных средств сверхбыстродействующих интегральных схем) и Verilog файлы проектов (.v) на языке Verilog HDL (языке описания аппаратных средств Verilog). С помощью текстового редактора MAX+plus II вы также можете просматривать, вводить и редактировать любые другие ASCII файлы. Открыть окно текстового редактора вы можете путем выбора **Text Editor** (текстовый редактор) из меню MAX+plus II. Команда **New** (новое) из меню **File** (файл) открывает новое, без названия, окно текстового редактора.

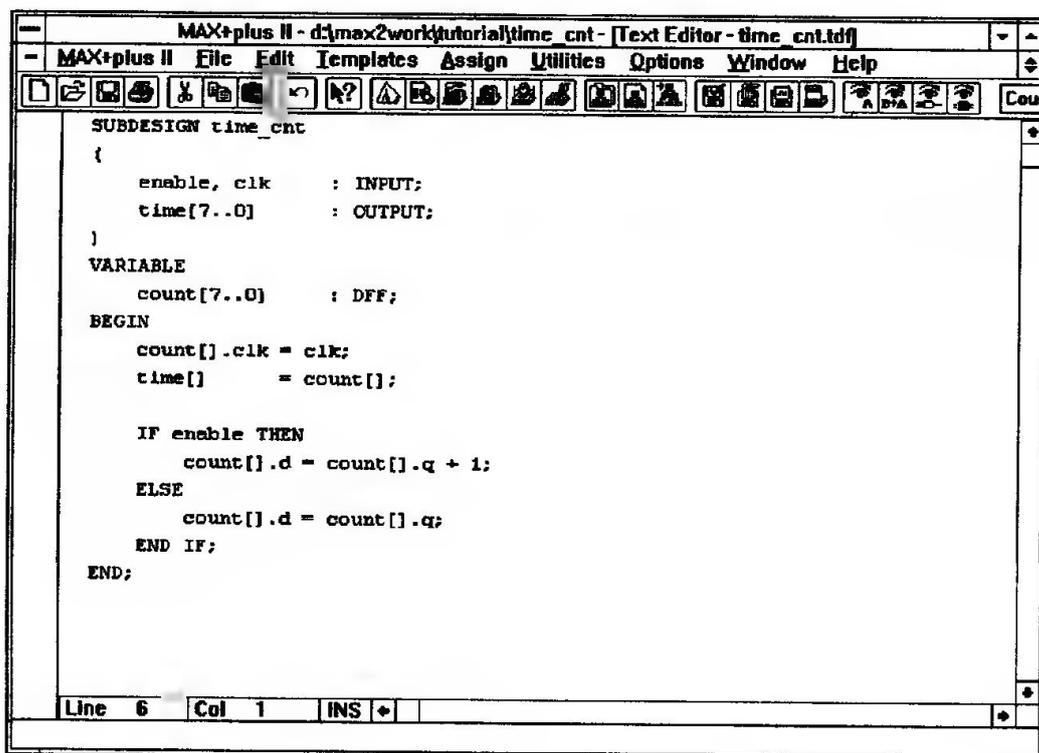


Рис. 1.11

Хотя для создания AHDL, VHDL и Verilog HDL проектных файлов вы можете использовать любой текстовый ASCII редактор, использо-

вание текстового редактора MAX+plus II дает вам возможность вводить проект, компилировать и отлаживать его с помощью одного САПР MAX+plus II .

Основные возможности текстового редактора MAX+plus II:

- языки AHDL, VHDL, Verilog HDL и текстовый редактор полностью интегрированы в систему MAX+plus II. Вы можете обрабатывать проекты, описанные с помощью языков AHDL, VHDL или Verilog HDL, компилятором, и процессор сообщений автоматически определит местонахождение любых синтаксических ошибок в текстовом редакторе. Текстовый редактор также предоставляет шаблоны для конструкций языков AHDL, VHDL и Verilog HDL. (См. "Язык описания аппаратных средств фирмы Altera" на стр. 53, "VHDL" на стр. 55 и "Verilog HDL" на стр. 57);
- вы можете включить синтаксическую функцию раскрашивания текста для того, чтобы ясно видеть структуру AHDL, VHDL, Verilog HDL файлов и других вспомогательных текстовых файлов;
- вы можете автоматически найти определенную секцию или комментарий, используя выбранный разграничитель. Это позволяет вам легко передвигаться по файлу проекта;
- вы можете использовать возможность редактирования "перетащить и оставить" для перемещения выбранного текста на новое место внутри файла;
- вы можете вручную редактировать Assignment & Configuration Files (файлы назначений и конфигурации) с расширением (.acf), которые задают синонимы имен цепей, ресурсы и назначения микросхем, а также параметры для компилятора, симулятора, анализатора временных параметров и программатора;
- вы можете создавать векторные файлы с расширением (.vec), которые используются в качестве входных файлов для моделирования, функционального тестирования или ввода проекта с помощью временных диаграмм. Вы также можете создать Command Files (командные файлы) с расширением (.cmd) для использования их симулятором MAX+plus II, командные файлы EDIF с расширением (.edc) и Library Mapping Files (файлы карты библиотек) с расширением (.lmf) для использования их компилятором MAX+plusII. Вы также можете редактировать любые другие ASCII файлы;
- при запуске компиляции или моделирования процессор сообщений MAX+plus II автоматически определяет местоположение любой синтаксической ошибки в текстовых файлах;
- с помощью контекстно-зависимой справки вы можете получить справку по элементам синтаксиса AHDL, ключевым словам и операторам. Вы также можете получить справку по всем предоставляемым фирмой Altera базовым элементам, мегафункциям и макрофункциям, которые встречаются в AHDL, VHDL и Verilog HDL

файлах проектов. Кроме того, вы можете получить контекстно-зависимую справку по ключевым словам и элементам синтаксиса в других текстовых файлах, таких как Assignment & Configuration Files (файлы назначений и конфигурации), Vector Files (векторные файлы), Command Files (командные файлы) и EDIF Command Files (командные файлы EDIF);

- текстовый редактор MAX+plus II предоставляет также и другие возможности. Например, вы можете найти, вырезать, копировать, вставить и удалить текст, выбирать различные шрифты текста и их размер, устанавливать количество пробелов при табуляции, использовать автоматический отступ.

 *За полной информацией о функциях и возможностях текстового редактора MAX+plus II обратитесь к справочной системе MAX+plus II.*

1.8.6. Редактор временных диаграмм MAX+plus II



Редактор временных диаграмм MAX+plus II, показанный на рис. 1.12, выступает в двух ролях: как инструмент ввода проекта и как инструмент для задания тестовых векторов и просмотра результатов моделирования. Вы можете создавать Waveform Design Files (файлы проектов, описанных временными диаграммами) с расширением (.wdf), которые содержат логическое описание проекта, и Simulator Channel Files (файлы временных диаграмм) с расширением (.scf), которые содержат входные векторы для моделирования и функционального тестирования. Открыть окно редактора временных диаграмм вы можете путем выбора **Waveform Editor** (редактор временных диаграмм) из меню MAX+plus II. Команда **New** (новое) из меню **File** (файл) открывает новое, без названия, окно редактора временных диаграмм.

Ввод проекта с помощью временной диаграммы является альтернативой для графического и текстового ввода проекта. Вы создаете проект с помощью комбинации логических уровней на входе и желаемых уровней на выходе в виде графической временной диаграммы. WDF может содержать логические входы, входы конечного автомата, а также комбинаторные и регистровые выходы, и выходы конечного автомата. Также могут использоваться внутренние сигналы для того, чтобы определить интересующие выходы.

Ввод проекта в виде временных диаграмм является подходящим для схем с хорошо определенной последовательностью входов и выходов, таких как конечные автоматы, счетчики и регистры. WDF удобны для

описания комбинаторных схем, декодеров, циклограмм и других периодических устройств.

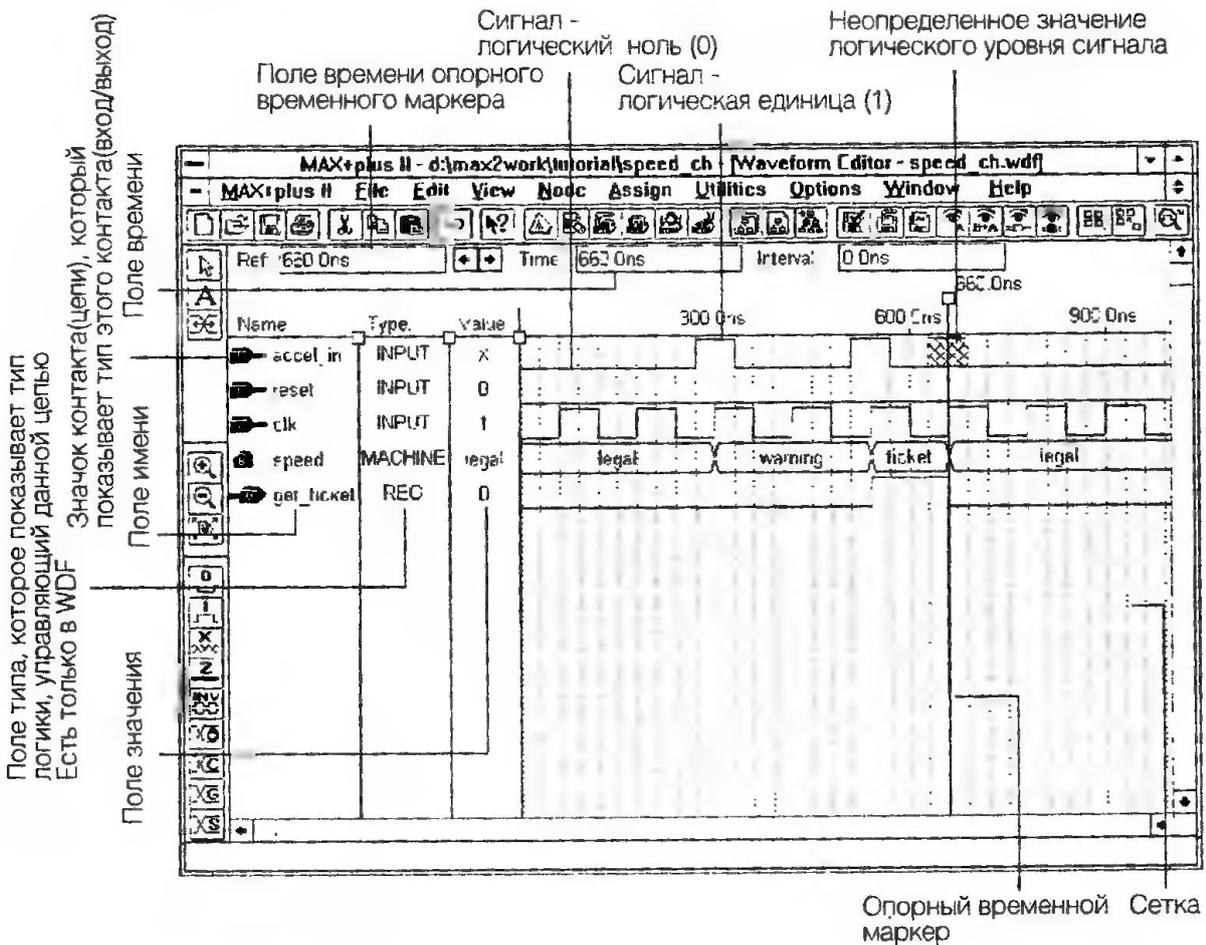


Рис. 1.12

Редактор временных диаграмм предоставляет множество возможностей. Вы легко можете трансформировать временную диаграмму полностью или частично, создавать и редактировать цепи и группы. С помощью небольшого набора простых команд вы можете создавать ASCII Table File (табличный файл ASCII) с расширением (.tbl) или импортировать ASCII Vector File (векторный файл ASCII) с расширением (.vec) для создания SCF или WDF файлов. Вы также можете сохранять WDF как SCF с целью моделирования или конвертировать SCF в WDF для использования его в качестве файла проекта.

Основные возможности редактора временных диаграмм MAX+plus II:

- вы можете создавать или редактировать цепь (узел), которая может представляться как входной контакт, выходной контакт или внутренний сигнал устройства;
- при создании WDF файла вы можете задать тип логики, соответствующей каждой цепи (узлу). Она может быть следующих типов: входной контакт, регистровая, комбинаторная или конечный автомат;

- вы можете задать высокий (1), низкий (0), неопределенный или высокоимпедансный (Z) логический уровень сигнала по умолчанию в логической цепи (узле), или любое исходное состояние по умолчанию в цепи (узле) типа конечного автомата;
- для упрощения создания тестового вектора вы можете добавить какой-либо один или все цепи (узлы) из Simulation Netlist File (файл со списком соединений для симулятора) с расширением (.snf) для полностью откомпилированного проекта, к SCF;
- вы можете объединять от 2 до 256 цепей в группу (шину) или отменить такое объединение, чтобы видеть каждую цепь, входящую в группу. Группы также могут быть объединены в другие группы. Значение сигнала в группе (шине) может отображаться в двоичной, десятичной, шестнадцатиричной или восьмеричной системе счисления с возможностью преобразования в код Грея;
- вы можете копировать, вставлять, перемещать или удалять выбранный фрагмент временной диаграммы, целую временную диаграмму для цепи (узла) или для всей группы (шины), то есть имя группы (шины) или цепи (узла) и ее временную диаграмму. С помощью одной операции вы можете редактировать множество интервалов или всю временную диаграмму. Вы также можете инвертировать, вставить, переписать, повторить, растянуть или сжать участок временной диаграммы любой длины с различными логическими уровнями, счетными последовательностями, именами состояний автоматов;
- вы можете определить и по желанию вывести на экран сетку для выравнивания фронтов логических сигналов до или после того, как они созданы;
- вы можете вставлять комментарии между временными диаграммами в любом месте внутри файла;
- вы можете изменять масштаб по любой шкале;
- для того, чтобы показать разницу в результатах моделирования, вы можете наложить второй файл временной диаграммы и осуществить сравнение с результатами, которые представлены в текущем файле. Второй файл может быть получен при тестировании реальной микросхемы.

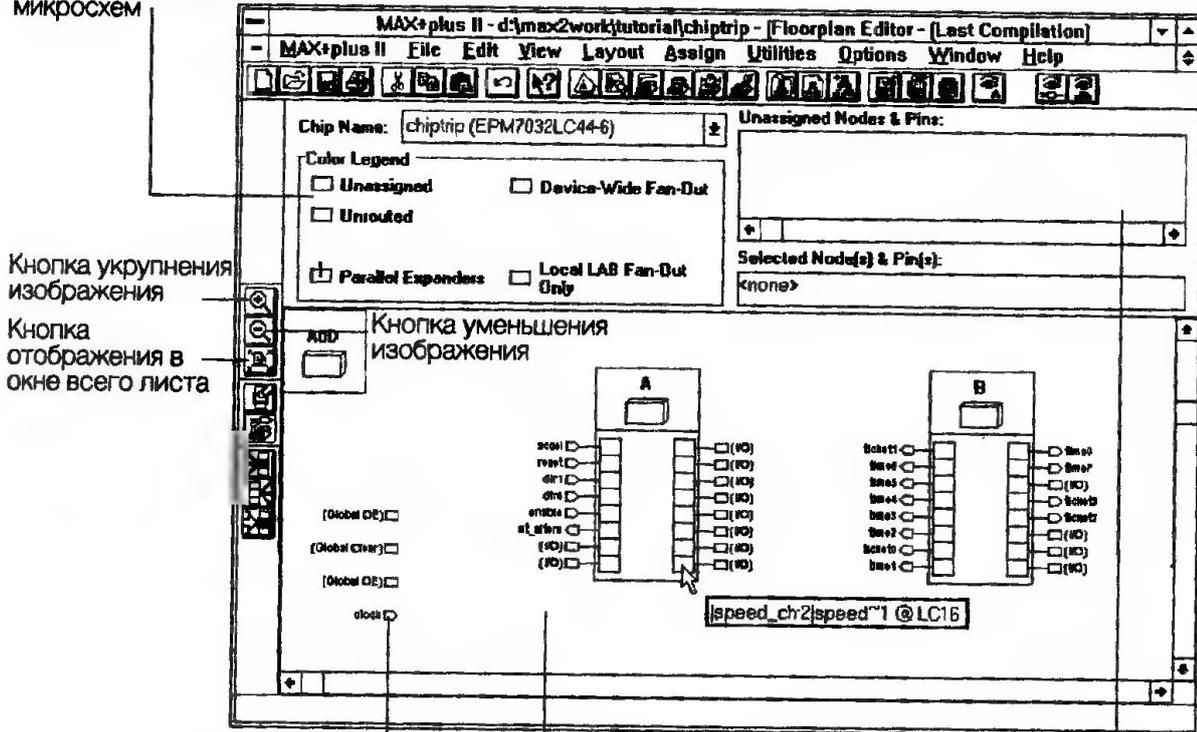
••• *За дополнительной информацией по редактированию временных диаграмм обратитесь к подразделу "Редактор временных диаграмм MAX+plus II" в разделе "Проверка проекта" на стр. 78. За полной информацией о функциях и возможностях редактора временных диаграмм MAX+plus II обратитесь к справочной системе MAX+plus II.*

1.8.7. Редактор физического размещения MAX+plus II



Вы можете использовать редактор физического размещения, показанный на рис. 1.13, для назначения физических ресурсов микросхемы и просмотра размещения и соединения логических ячеек, выполненного компилятором. Открыть окно редактора физического размещения вы можете путем выбора **Floorplan Editor** (редактор физического размещения) из меню MAX+plus II.

Легенда показывает цвета, обозначающие размещенные, неразмещенные цепи и использование внутренних ресурсов различных семейств микросхем



Кнопка укрупнения изображения

Кнопка отображения в окне всего листа

Кнопка уменьшения изображения

Выделенные общие контакты для некоторых микросхем показываются отдельно от LAB

В данный момент на экране LAB View (вид логических блоков)

Вы можете переместить название цепи или контакта на изображение микросхемы, чтобы назначить их внешнему контакту микросхемы, логической ячейке, логическим блокам, ячейке ввода/вывода, ячейке встроенной памяти, строке, колонке или чипу, в зависимости от выбранного представления

Рис. 1.13

Редактор физического размещения предоставляет удобные средства для ввода и редактирования назначений физических ресурсов микросхемы вашего проекта. Возможны два варианта изображения микросхемы:

- **Device View** (вид микросхемы), показывающий все контакты на корпусе микросхемы и их функции;
- **LAB View** (вид логических блоков), показывающий внутреннее содержание микросхемы, включая все LAB (логические блоки)

и отдельные LC (логические ячейки) внутри каждого LAB. В микросхемах, которые включают Embedded Array Blocks (EAB) (блоки встроенной памяти), вы можете просмотреть отдельную ячейку внутри каждого EAB.

Для микросхем MAX 9000, FLEX 6000, FLEX 8000, FLEX 10K и ACEX 1K отображается местоположение I/O cell (ячейки ввода вывода). Также отображаются внешние контакты микросхемы по периметру кристалла.

Редактор физического размещения показывает список имен неназначенных цепей и контактов вашего проекта в поле Unassigned Nodes & Pins. Каждое имя имеет значок, который вы можете переместить на изображение внешнего контакта микросхемы, логической ячейки, ячейки ввода/вывода или ячейки встроенной памяти при отображении на экране Device View (вида микросхемы) или LAB View (вида логических блоков). Вы также можете переместить цепь или контакт, имеющие назначения, обратно в список неназначенных цепей и контактов или в другое место на кристалле.

Вы также можете сделать более общее назначение для всего LAB, EAB или микросхемы и затем дать возможность компилятору выбрать наиболее подходящее местоположение внутри LAB, EAB или микросхемы. В микросхемах семейств MAX 9000, FLEX 6000, FLEX 8000, FLEX 10K и ACEX 1K вы также можете назначать цепи и контакты для строк или колонок. При этом внутри bin (кармана) соответствующего LAB, EAB, микросхеме, строке или колонке выводится число, показывающее количество назначенных ему цепей или контактов.

Основные возможности редактора физического размещения MAX+plus II:

- вы вводите назначения физических ресурсов в графической среде drag-and-drop (перетащить и оставить). Назначения компилятора также могут быть скопированы и отредактированы;
- цветная легенда обозначает размещенные и неразмещенные цепи, использование внутренних ресурсов различных семейств микросхем, контакты VCC, GND и зарезервированные контакты;
- вы можете просматривать и редактировать ваши текущие назначения для проекта, которые сохраняются в файле Assignment & Configuration с расширением (.act). Вы также можете вывести на экран нередатируемое (только для просмотра) размещение вашего проекта при последней компиляции, которое хранятся в Fit File (файле размещения) с расширением (.fit), независимо от того, было ли это размещение успешным. Любой из неразмещенных элементов выделяется цветом в списке названий неразмещенных цепей и контактов. Узлы, которые уже размещены, но не соединены, указываются красным цветом;

- вы можете вывести на экран входные и выходные связи для любого выбранного элемента (элементов) или связи между элементами в выбранной группе. Вы также можете просмотреть подробную статистику об использовании линий соединений для выбранного элемента (элементов);
- **Report File Equation Viewer** (окно для просмотра уравнений из файла отчета) показывает вам для выбранного элемента или элементов логическое уравнение и имена всех цепей и контактов, которые являются входными или выходными переменными в этом уравнении. Вы можете выбрать входные или выходные переменные и посмотреть их логические уравнения. Таким образом, вы можете проследить распространение сигнала через весь кристалл от входных контактов до выходных или наоборот;
- имя логической функции, назначенной контакту или логической ячейке, автоматически выводится на экран в белом прямоугольнике, когда на нем (ней) установлен курсор;
- если множество элементов размещены в одну и ту же область, то вы можете просмотреть список всех элементов и выбрать один из них для редактирования;
- вы можете назначить одно и то же имя для выходного контакта одной микросхемы и для входного другой, чтобы управлять разделением проекта на части в проекте, реализованном на нескольких микросхемах;
- функция компилятора **smart recompile** (интеллектуальная перекомпиляция) позволяет вам быстро перекомпилировать ваш проект после размещения, сделанного в редакторе физического размещения.

Редактор физического размещения MAX+plus II предоставляет множество других функций. Например, вы можете увеличивать или уменьшать изображение на экране, чтобы видеть всю микросхему или ее часть. Вы также можете копировать, вырезать, вставлять или удалять одно или несколько выбранных размещений и проводить поиск по имени группы, имени цепи или по номерам контактов, логических ячеек, ячеек ввода/вывода, ячеек встроенной памяти.

1.8.8. Язык описания аппаратных средств фирмы Altera



Язык описания аппаратных средств фирмы Altera (AHDL) является высокоуровневым модульным языком, который полностью интегрируется в систему. Вы можете использовать текстовый редактор MAX+plus II или другой текстовый редактор для создания AHDL Text Design Files (текстовых файлов проекта AHDL) с расширением (.tdf), которые вы компилируете и моделируете в MAX+plus II (рис.1.14).

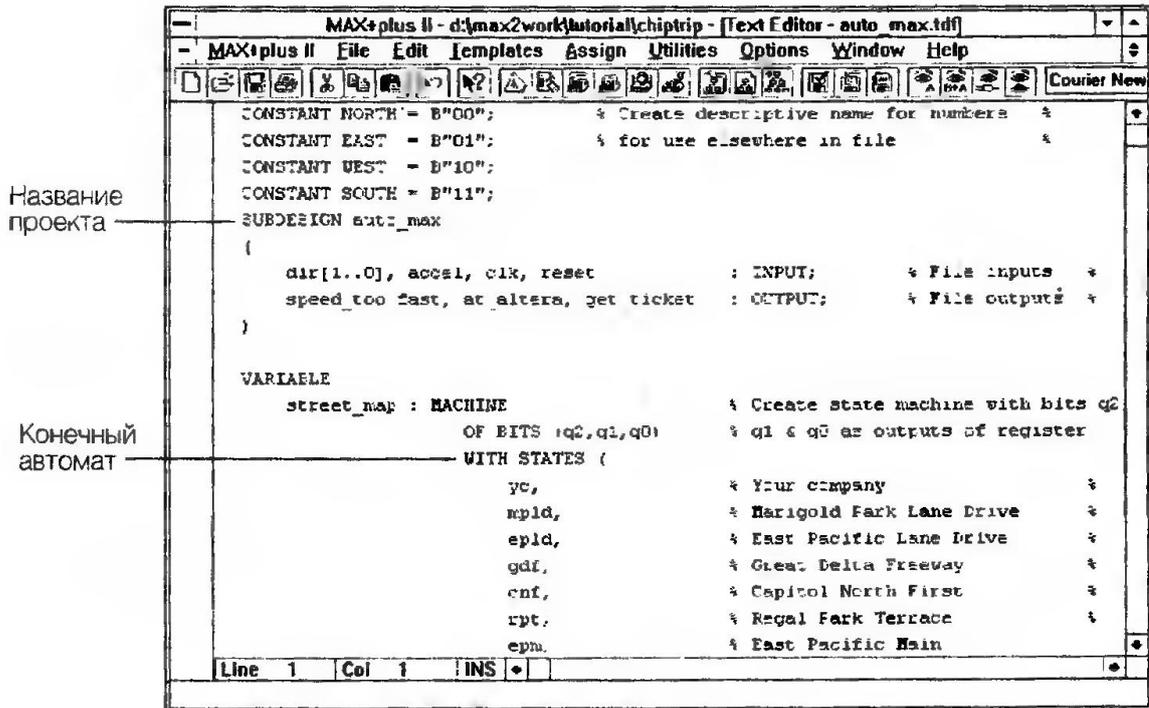


Рис. 1.14

Вы можете использовать любой текстовый редактор ASCII для создания TDF. Однако когда вы вводите файлы AHDL с помощью текстового редактора MAX+plus II, вы можете использовать преимущества ввода проекта, компиляции и отладки, которые есть в САПР MAX+plus II. Например, вы можете воспользоваться преимуществами шаблонов AHDL, использовать синтаксическую раскраску для облегчения просмотра различных разделов файла, получить контекстно-зависимую справку о синтаксических элементах AHDL, ключевых словах и предложениях, а также о базовых элементах, мегафункциях и макрофункциях. Вы можете произвести назначения ресурсов и параметров, использовать возможность автоматического определения местоположения ошибок во время и после компиляции.

AHDL состоит из различных элементов и операторов, которые описывают логику. Ниже выделены основные возможности AHDL, которые делают его идеальным инструментом для описания конечных автоматов, таблиц истинности, булевых уравнений, логики и групповых операторов:

- вы можете использовать функции из библиотеки параметризованных модулей (LPM) для реализации логики. LPM предоставляет комбинационные схемы, арифметические модули, элементы памяти для реализации цифровых устройств;
- в качестве альтернативы использования LPM функций, вы можете описывать цифровые устройства с помощью булевых выражений и уравнений, макрофункций и таблиц истинности.

- вы можете создать свои собственные параметризованные проекты в AHDL, используя операторы языка;
- вы можете сохранять часто используемые константы, оценочные функции, параметры и прототипы функций в Include Files (файлах включения) с расширением (.inc) и вставлять их в любой TDF;
- AHDL удобен для проектирования конечных автоматов. Вы можете сами определить способ кодирования состояний или позволить сделать это компилятору. Вы также можете импортировать и экспортировать AHDL конечные автоматы между TDF и другими файлами проектов в иерархии;
- в процессе компиляции MAX+plus II может генерировать AHDL Text Design Export Files (экспортные файлы проекта на языке AHDL) с расширением (.tdx) и Text Design Output Files (выходные текстовые файлы проекта) с расширением (.tdo). Независимо от вашего первоначального метода ввода проекта, вы можете переименовать этот файл в TDF и использовать его вместо первоначального файла (или файлов) проекта.

☛ Обратитесь к справочной системе MAX+plus II или к справочнику MAX+plus II AHDL для получения полной информации по AHDL.

1.8.9. VHDL



Язык описания аппаратуры скоростных ИС (VHDL) представляет собой высокоуровневый модульный язык, полностью интегрированный в САПР MAX+plus II. VHDL является языком описания аппаратуры промышленного стандарта, описывающим входы, выходы, поведение и назначение схем. Этот язык определяется Стандартами IEEE 1076-1987 и 1076-1993. Вы можете использовать текстовый редактор MAX+plus II или другой текстовый редактор для создания VHDL файлов проектов (.vhd) в синтаксических элементах VHDL 1987 или 1993, которые вы компилируете и моделируете в MAX+plus II (рис. 1.15).

Вы можете использовать любой текстовый редактор ASCII для создания VHDL файла проекта (.vhd). Однако, когда вы вводите VHDL файл проекта посредством текстового редактора MAX+plus II, вы можете воспользоваться возможностями ввода проекта, компиляции и отладки, которые доступны только в редакторах MAX+plus II. Например, вы можете воспользоваться шаблонами VHDL, использовать контекстно-зависимую справку MAX+plus II для получения информации о базовых элементах, мегафункциях и макрофункциях фирмы Altera, использовать синтаксическую раскраску для облегчения просмотра различных участков файла, произвести назначения ресурсов и параметров микросхемы

и использовать возможность автоматического поиска местоположения ошибок во время и после компиляции.

```

-- An up/down counter
PROCESS (clk)
  VARIABLE cnt      : INTEGER RANGE 0 TO 255;
  VARIABLE direction : INTEGER;
BEGIN
  IF (up_down = '1') THEN
    direction := 1;
  ELSE
    direction := -1;
  END IF;

  IF (clk'EVENT AND clk = '1') THEN
    cnt := cnt + direction;
  END IF;

  qd <= cnt;

END PROCESS;

```

Line 99 Col 5 INS

Рис. 1.15

VHDL файлы проектов могут содержать любую комбинацию конструкций, поддерживаемых MAX+plus II. Они также могут содержать базовые элементы, мегафункции и макрофункции фирмы Altera, т.е. файлы проектов нижнего уровня иерархии, а также мега и макрофункции, определяемые пользователем.

Компилятор MAX+plus II может создавать выходные файлы VHDL (.vho), содержащие информацию о функциональных и временных параметрах проекта после синтеза. Эти файлы могут быть импортированы в симуляторы промышленного стандарта. Информация о временных параметрах также может быть записана в Standard Delay Format Output Files (выходные файлы в формате стандартной задержки) с расширением .sdo.

- Для получения полной и обновленной информации о поддержке VHDL САПР MAX+plus пользуйтесь справочной системой САПР MAX+plus II (Help) или руководством MAX+plus II VHDL.

1.8.10. Verilog HDL



Язык описания аппаратуры Verilog является высокоуровневым модульным языком, полностью интегрированным в САПР MAX+plus II. Verilog HDL является языком описания аппаратуры промышленного стандарта, описывающим входы и выходы, поведение и назначение схем. Этот язык определяется стандартом IEEE Std 1364. Вы можете использовать текстовый редактор MAX+plus II или другой текстовый редактор для создания Verilog файлов проектов (.v), которые вы компилируете и моделируете в MAX+plus II (рис. 1.16).

```

MAX+plus II - c:\max2work\verilog\counters - [Text Editor - counters.v]
MAX+plus II File Edit Templates Assign Utilities Options Window Help
Courier

// MAX+plus II Verilog Example
// Efficient Counter Inference
// Copyright (c) 1997 Altera Corporation

module counters (d, clk, clear, ld, enable, up_down,
                qa, qb, qc, qd, qe, qf, qg,
                qh, qi, qj, qk, ql, qa, qn);

    input  [7:0] d;
    input  clk, clear, ld, enable, up_down;
    output [7:0] qa, qb, qc, qd, qe, qf, qg;
    output [7:0] qh, qi, qj, qk, ql, qa, qn;

    reg    [7:0] qa, qb, qc, qd, qe, qf, qg;
    reg    [7:0] qh, qi, qj, qk, ql, qa, qn;

    integer direction;

    // An enable counter
    always @(posedge clk)
    begin
        if (enable)
            qa = qa + 1;
    end

    // A synchronous load counter
    always @(posedge clk)
    begin
        if (ld)
            qa = d;
    end

```

Рис. 1.16

Вы можете использовать любой текстовый редактор ASCII для создания Verilog файла проекта (.v). Однако, когда вы вводите Verilog файл проекта с помощью текстового редактора MAX+plus II, вы можете воспользоваться уникальными возможностями ввода проекта, компиляции и отладки, которые доступны только в редакторах MAX+plus II. Например, вы можете воспользоваться шаблонами Verilog HDL, использовать контекстно-зависимую справку MAX+plus II для получения информации о базовых элементах, мегафункциях и макрофункциях фирмы Altera, использовать синтаксическую раскраску для облегчения просмотра различных участков файла, произвести назначения ресурсов и параметров микросхемы и использовать возможность автоматического поиска местоположения ошибок во время и после компиляции.

Verilog HDL файлы проектов могут содержать любую комбинацию конструкций, поддерживаемых MAX+plus II. Они также могут содержать базовые элементы, мегафункции и макрофункции фирмы Altera, а также функциональные модули, определяемые пользователем.

Компилятор MAX+plus II может создавать выходные файлы Verilog (.vo), содержащие информацию о функциональных и временных параметрах проекта после синтеза. Эти файлы могут быть импортированы в симуляторы промышленного стандарта для моделирования вашего проекта.

Информация о временных параметрах также может быть записана в Standard Delay Format Output Files (выходные файлы в формате стандартной задержки) с расширением .sdo.

••• Для получения полной и обновленной информации о поддержке Verilog HDL САПР MAX+plus II пользуйтесь справочной системой САПР MAX+plus II (**Help**) или руководством **MAX+plus II Verilog HDL**.

1.8.11. Primitives (базовые элементы), мегафункции и макрофункции

Altera предоставляет библиотеки функциональных модулей — базовые элементы, мегафункции и макрофункции микросхем серий 74XXXX — включая модули, оптимизированные для архитектуры отдельных семейств микросхем. Во время инсталляции все функциональные модули копируются в подкаталоги \maxplus2\max2lib и \maxplus2\vhdlm, где *m* это "87" или "93". (На рабочих станциях UNIX директория maxplus2 является подкаталогом директории /usr.)

САПР MAX+plus II содержит информацию обо всех функциональных модулях, включая входные логические уровни по умолчанию, AHDL прототипы функций, VHDL объявления компонентов и функциональные таблицы модулей. Вы просто выбираете кнопку контекстно-зависимой справки на панели инструментов или нажимаете клавиши **Shift+F1** в графическом или текстовом редакторе, затем щелкаете кнопку 1 на символе функционального элемента или его имени для открытия соответствующей темы справочной системы.

Базовые элементы

Базовые элементы, а именно буфер, триггер, триггер-защелка, вход/выход и логические вентили, являются простейшими функциональными блоками, используемыми для проектирования цифровых схем с помощью САПР MAX+plus II. Они могут использоваться в GDF, AHDL, VHDL и Verilog HDL файлах проектов.

Базовые элементы в файлах проектов HDL являются подмножеством символов базовых элементов, используемых в GDF файлах. Остальные базовые элементы могут быть описаны различными операторами и средствами языков описания аппаратуры.

Базовые элементы являются элементами языка AHDL. VHDL объявления компонентов для базовых элементов содержатся в файле `maxplus2.vhd` в подкаталоге `\maxplus2\vhdlmn\altera`, где *mn* это "87" или "93".

В графическом редакторе при рисовании схемы вы можете соединить базовый элемент с шиной. Компилятор автоматически преобразует такое соединение в массив базовых элементов. Количество базовых элементов в нем будет соответствовать числу проводников в шине. Массивы базовых элементов являются альтернативой использования параметризованных функций.

Мегафункции

Мегафункции представляют собой сложные или высокоуровневые блоки, которые могут быть использованы вместе с базовыми элементами и другими мега- и макрофункциями для создания проекта.

Многие мегафункции, включая функции из библиотеки параметризованных модулей (LPM), являются параметризованными по своей сути. В них можно задать разрядность, необходимые управляющие сигналы и другие параметры, которые влияют на поведение и реализацию мегафункции. Возможность задавать параметры может сильно упростить ввод проекта. Мегафункции могут свободно использоваться в GDF и во всех файлах проектов HDL. Когда компилятор анализирует все цифровое устройство, он автоматически использует логику, специализированную для выбранного семейства микросхем и удаляет все неиспользуемые элементы для обеспечения оптимального использования ресурсов при реализации данного проекта.

Макрофункции микросхем серий 74XXXX

Макрофункции микросхем серий 74XXXX представляют собой сложные или высокоуровневые блоки, которые могут быть использованы вместе с базовыми элементами и другими мега- и макрофункциями для создания проекта. Они могут использоваться в GDF и во всех файлах проектов HDL. Когда компилятор анализирует цифровое устройство, он автоматически использует мегафункции, специализированные для выбранного семейства микросхем, и удаляет все неиспользуемые вентили и триггеры для обеспечения оптимального использования ресурсов при реализации данного проекта. Все входы макрофункций имеют заданные по умолчанию уровни входных сигналов, и неиспользуемые контакты могут оставаться несоединенными.

Многие макрофункции имеют шинные эквиваленты, которые функционально идентичны, но имеют сигнальные линии, сгруппированные в шины.

Макрофункции микросхем 74XXXX не являются параметризованными. Однако некоторые параметры, специфичные для микросхем фирмы Altera, могут применяться к макрофункциям для определения способа их реализации.

 Altera рекомендует использовать мегафункции LPM вместо эквивалентных макрофункции микросхем 74XXXX. LPM и другие параметризованные функции легче в использовании, они масштабируются и более эффективно реализуются в микросхеме.

1.9. Иерархия проекта



Дисплей иерархии проекта MAX+plus II показывает иерархический проект в виде дерева иерархии, где файлы проектов более низкого уровня представлены в виде ветвей (рис. 1.17). В одном проекте могут использоваться различные способы ввода проектов.

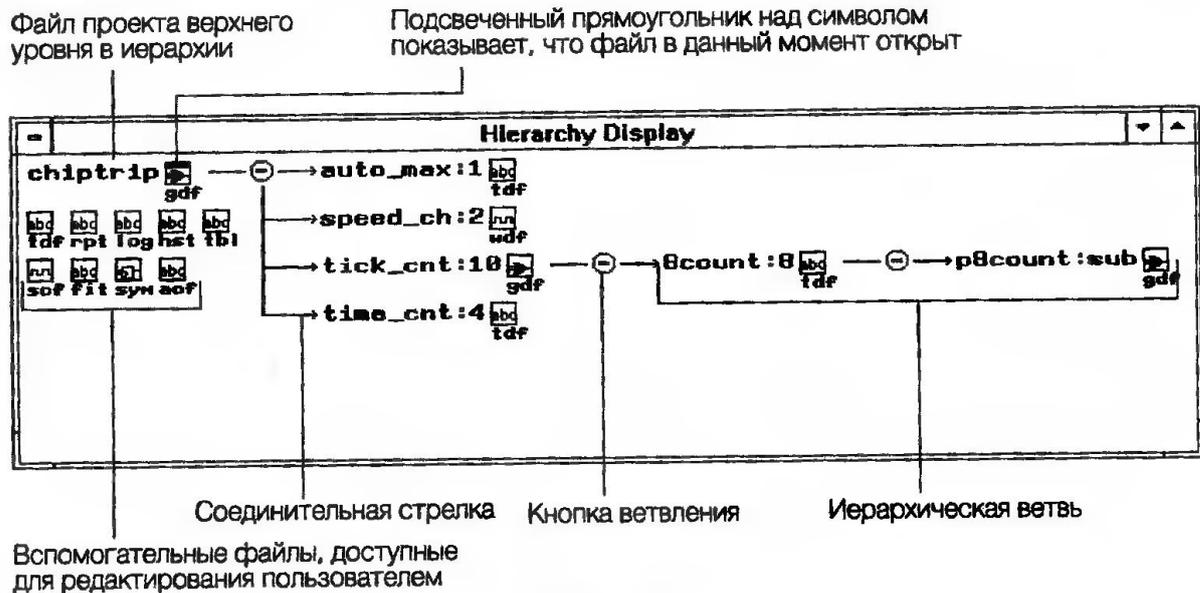


Рис. 1.17

Когда вы открываете дисплей иерархии проекта, то он показывает полную иерархию файлов проектов, называемую "иерархическим деревом" текущего проекта. Если один или несколько файлов в иерархии открыты, то верхняя часть символа этого файла отображается в виде подсвеченного прямоугольника. Если проект скомпилирован, то дисплей иерархии проекта показывает всю иерархию файлов проектов и все вспомогательные файлы для файла проекта верхнего уровня.

Из окна дисплея иерархии проекта вы можете открыть любой файл, который в нем указан. При этом автоматически будут открываться соответствующие редакторы. Вы также можете увеличить или уменьшить масштаб изображения на экране для того, чтобы увидеть всю иерархию или ее часть.

Дисплей иерархии проекта предоставляет следующие возможности:

- вы легко можете открыть любой файл проекта или вспомогательный файл в текущей иерархии из окна дисплея иерархии проекта;
- кнопки ветвления на пересечениях ветвей иерархического дерева позволяют вам скрывать или отображать ветви более низкого уровня;
- все имена файлов сопровождаются значками соответствующих редакторов MAX+plus II и расширениями имен;
- если файл открыт, то над значком файла отображается подсвеченный прямоугольник. При закрытии файла прямоугольник исчезает;
- вы можете выбрать файл проекта и просмотреть его физическую реализацию в окне **LAB View** редактора физического размещения;
- вы можете выбрать файл проекта и назначить ресурсы для него. Такое действие аналогично назначению ресурсов для символа в графическом редакторе;
- вы можете отобразить на экране любую из множества открытых иерархий;
- вы можете отобразить иерархию в горизонтальном или вертикальном представлении;
- вы можете вывести на печать текущее иерархическое дерево или любую комбинацию файлов проекта и вспомогательных файлов.

••• Для получения полной информации обо всех функциях и возможностях дисплея иерархии проекта MAX+plus II пользуйтесь справочной системой САПР MAX+plus II (**Help**).

1.10. Обработка проекта

САПР MAX+plus II обрабатывает проекты для микросхем Altera Classic, MAX 3000, MAX 5000, MAX 7000, MAX 9000, FLEX 6000, FLEX 8000, FLEX 10K и ACEX1K. САПР MAX+plus II компилирует проекты автоматически, но вы можете задать подробную спецификацию процесса компиляции. На рис. 1.18 показано, как САПР MAX+plus II компилирует проекты.

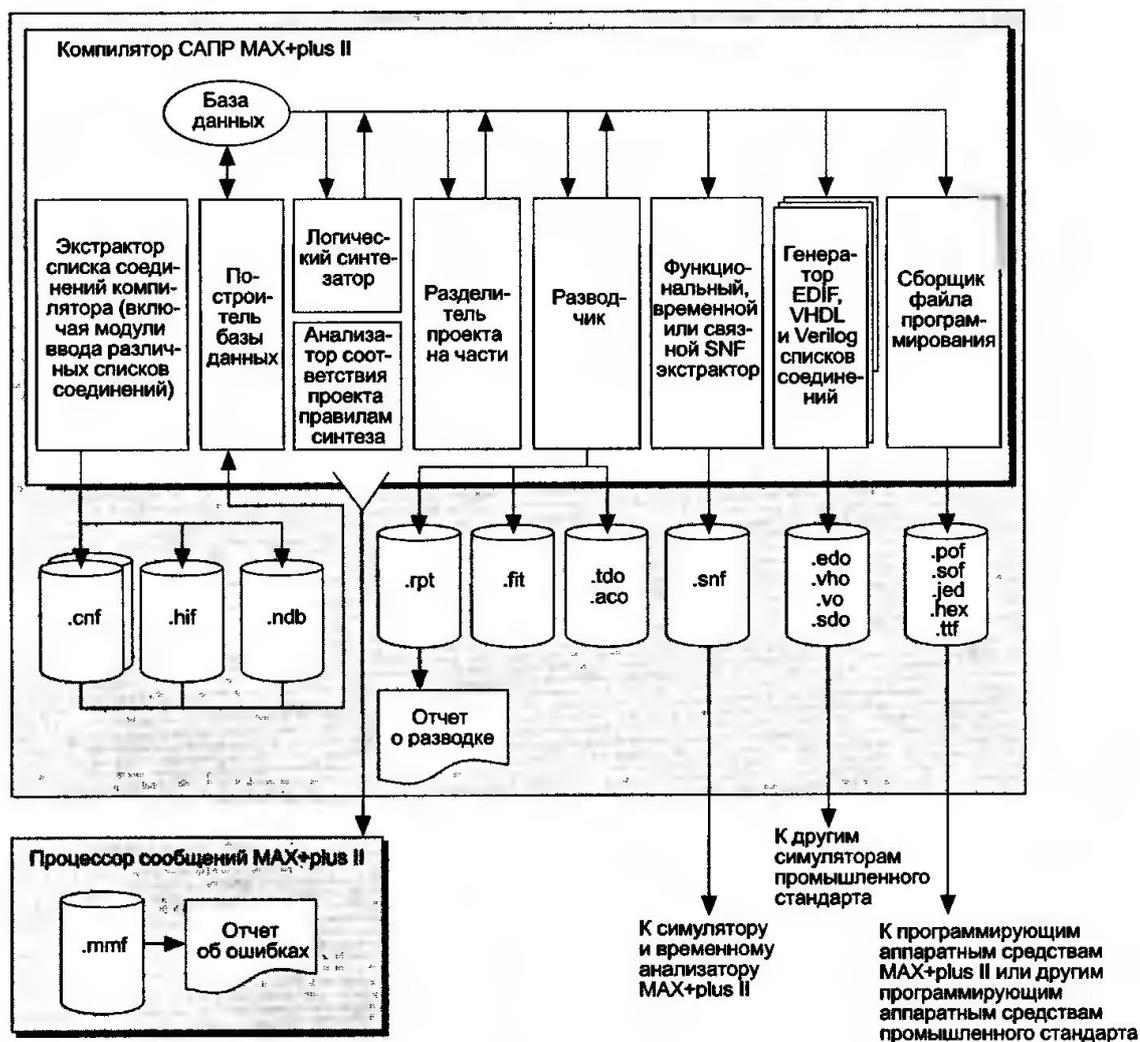


Рис. 1.18

1.10.1. Компилятор MAX+plus II



Компилятор MAX+plus II состоит из ряда модулей и утилит, проверяющих проект на ошибки, синтезирующих логику, подбирающих одну или несколько подходящих микросхем фирмы Altera и генерирующих выходные файлы для моделирования, временного анализа и программирования микросхем. Компилятор связывает программные модули ввода проекта

(графический редактор, текстовый редактор, редактор временных диаграмм, редактор символов и редактор физического размещения) с программными модулями последующей обработки (временной анализатор, симулятор и программатор). На рис. 1.19 показано окно компилятора MAX+plus II.

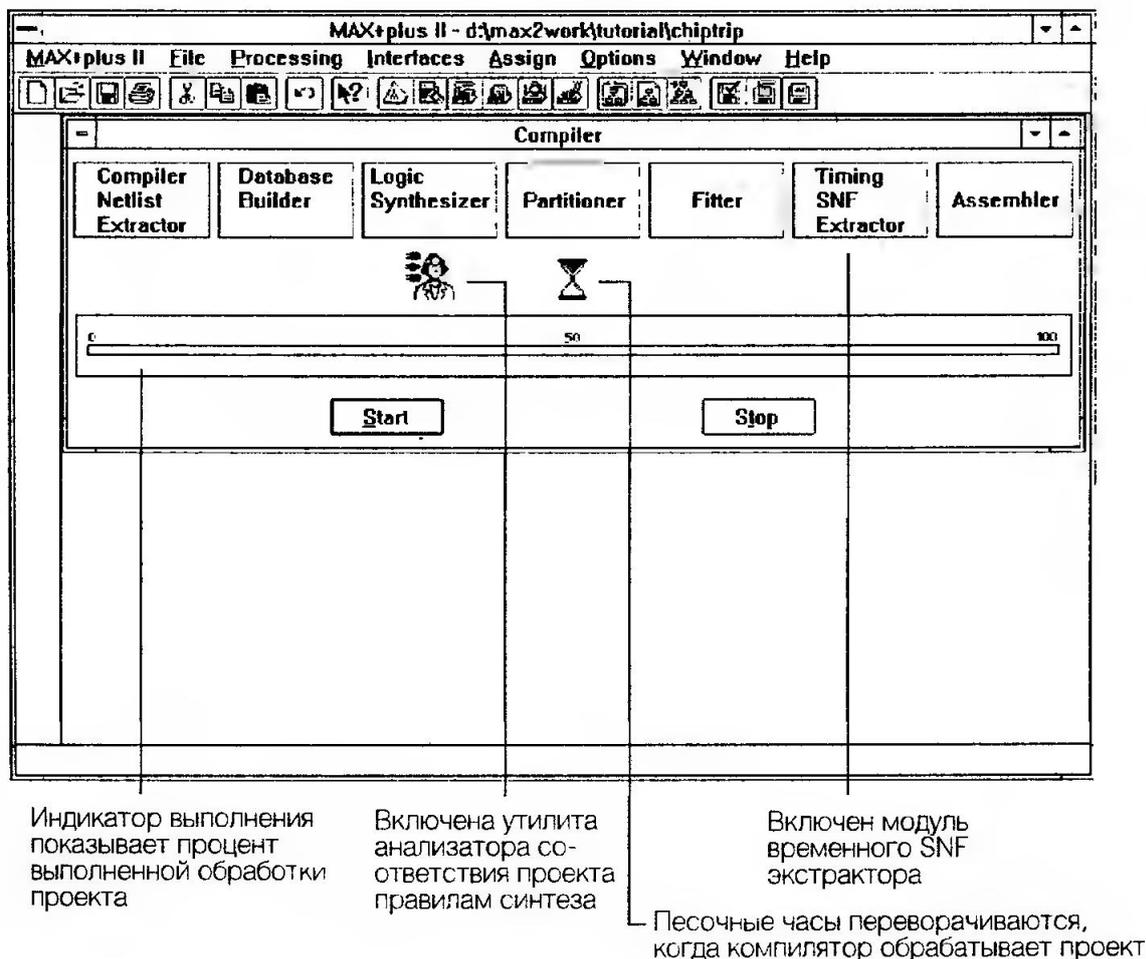


Рис. 1.19

Входные файлы компилятора

Компилятор MAX+plus II использует следующие файлы:

- графические файлы проекта (**.gdf**), созданные с помощью графического редактора MAX+plus II;
- текстовые файлы проекта (**.tdf**), созданные на языке описания аппаратных средств фирмы Altera (AHDL);
- VHDL файлы проекта (**.vhd**), созданные в синтаксисе VHDL 1987 или 1993;
- Verilog файлы проекта (**.v**), созданные в синтаксисе Verilog HDL;
- входные файлы EDIF версии 200 или 300 (**.edf**), созданные любым стандартным генератором **EDIF netlist** файлов;
- файлы проектов временных диаграмм (**.wdf**), созданные с помощью редактора временных диаграмм MAX+plus II;

- OrCAD схемные файлы (**.sch**), созданные с помощью схемного редактора OrCAD Draft или с помощью графического редактора MAX+plus II;
- файлы формата списка соединений фирмы Xilinx (**.xnf**), созданные с помощью программного обеспечения фирмы Xilinx;
- файлы проектов Altera (**.adf**), созданные с помощью A+plus – первого САПР Altera. ADF файл для описания проекта использует формат списка соединений и булевы логические выражения. Во время компиляции проекта компилятор MAX+plus II автоматически преобразует ADF в файл списка соединений компилятора (**.cnf**);
- файлы конечных автоматов (**.smf**), которые содержат описание конечных автоматов, созданные для использования с Altera A+plus или SAM+plus. Во время компиляции компилятор MAX+plus II автоматически преобразует SMF в файл проекта Altera (**.adf**) и файл списка соединений компилятора (**.cnf**);
- шестнадцатиричные (формата Intel) файлы (**.hex**) и/или файлы исходного состояния памяти (**.mif**), содержащие первоначально загружаемые данные для блоков памяти;
- командные файлы EDIF (**.edc**), используемые для задания параметров выходных EDIF файлов (**.edo**), созданных компилятором MAX+plus II;
- файл назначений и конфигурации (**.acf**), в котором хранятся параметры и назначения для данного проекта, настройки компилятора, симулятора и временного анализатора.

 *Информация о назначениях и конфигурации версий MAX+plus II до 5.0, которая хранилась в файлах TDF, Probe & Resource Assignment (**.prb**), и <имя проекта>.ini, может быть автоматически преобразована в формат ACF.*

- символные файлы (**.sym**), созданные с помощью редактора символов MAX+plus II;
- файлы включения (**.inc**), импортируемые в AHDL текстовый файл проекта (**.tdf**) посредством AHDL Include Statement (оператора включения в языке AHDL). Файл включения замещает вызвавший его оператор включения. Файл включения может содержать операторы языка AHDL Function Prototype, Define, Parameters или Constant Statements. Компилятор также использует AHDL прототипы функций в файлах включения для обработки логических функций в Verilog файлах проектов (**.v**);
- Library Mapping Files (файлы карты библиотек) (**.lmf**), используемые для составления карты ячеек во входных файлах EDIF и OrCAD схемных файлах, соответствующих базовым элементам и макрофункциям MAX+plus II.

Процесс компиляции

Компилятор сначала извлекает информацию, определяющую иерархические связи между файлами проектов, и проверяет проект на основные ошибки при его вводе. Он создает карту всего проекта и объединяет все файлы проектов в базу данных, которая может быть эффективно обработана.

Компилятор использует различные алгоритмы для повышения быстродействия вашего проекта и минимизации используемых ресурсов микросхемы. Если ваш проект слишком велик для одной микросхемы, компилятор может автоматически разделить его на несколько микросхем из одного и того же семейства. При этом число соединений между микросхемами будет минимизировано. Файл отчета (.rpt) показывает, как будет реализован проект на одной или нескольких микросхемах.

Также компилятор создает файлы для программатора, которые программатор MAX+plus II или другой программатор промышленного стандарта использует для программирования одной или нескольких микросхем фирмы Altera.

Хотя компилятор может компилировать проект при минимальном вашем вмешательстве, он также позволяет вам задавать различные параметры обработки проекта. Например, вы можете определить стиль логического синтеза проекта по умолчанию и задать различные настройки логического синтеза для всего проекта. Вы можете задать требования к характеристикам быстродействия проекта, определить, как разделить большой проект на несколько микросхем и установить разнообразные опции для выбранной микросхемы. Вы также можете указать сколько контактов и логических ячеек должны оставаться свободными во время текущей компиляции для обеспечения дополнительных функций вашего устройства в будущем.

Выполнение компиляции

Вы можете начать компиляцию проекта из любого программного модуля MAX+plus II или из модуля компилятора. Компилятор автоматически обрабатывает все входные файлы для текущего проекта, и вы можете отслеживать процесс компиляции в окне компилятора:

- песочные часы отсчитывают время и переворачиваются, указывая на то, что компилятор работает;
- поля модулей поочередно подсвечиваются по мере того, как компилятор заканчивает выполнение каждой стадии обработки;
- значки, представляющие файлы, появляются под генерирующими их полями модулей компилятора. Вы можете дважды щелкнуть кнопку 1 мыши на значке для открытия соответствующего файла;
- процент выполненной обработки проекта показан движущейся к 100% полоской в индикаторе выполнения;

- во время разделения проекта на части и разводки кнопка **Stop** в окне компилятора преобразуется в кнопку **Stop/Show Status**, которую вы можете нажать для открытия диалогового окна и просмотра статуса проекта на этих этапах обработки;
- при обнаружении любой ошибки или потенциальной проблемы во время компиляции автоматически открывается окно процессора сообщений, где выводятся информационные сообщения, сообщения об ошибках и предупреждения. Вы можете получить справку о том, как эту ошибку можно исправить, определить местоположение источника сообщения в файлах проектов или в редакторе физического размещения.

Компилятор может запускаться в фоновом режиме. Вы можете свернуть его, пока он обрабатывает проект, и продолжить работу с другими файлами. Индикатор выполнения под свернутой иконкой компилятора позволяет вам видеть продвижение процесса компиляции, при этом вы можете уделять основное внимание другой задаче.

1.10.2. Модули компилятора и выходные файлы

Компилятор MAX+plus II обрабатывает проект с помощью следующих модулей и утилит:

- **Compiler Netlist Extractor** (экстрактор списка соединений компилятора, включая встроенные модули ввода списков соединений EDIF, VHDL, Verilog и XNF);
- **Database Builder** (построитель базы данных);
- **Logic Synthesizer** (логический синтезатор);
- **Partitioner** (разделитель проекта на части);
- **Fitter** (разводчик);
- **Functional SNF Extractor** (функциональный SNF экстрактор);
- **Timing SNF Extractor** (временной SNF экстрактор);
- **Linked SNF Extractor** (связывающий SNF экстрактор);
- **EDIF Netlist Writer** (генератор EDIF списка соединений);
- **Verilog Netlist Writer** (генератор Verilog списка соединений);
- **VHDL Netlist Writer** (генератор VHDL списка соединений);
- **Assembler** (сборщик файла программирования);
- **Design Doctor Utility** (утилита анализатора соответствия проекта правилам синтеза).

Compiler Netlist Extractor (экстрактор списка соединений компилятора, включая встроенные модули ввода списков соединений EDIF, VHDL, Verilog и XNF)

Экстрактор списка соединений компилятора преобразует каждый файл проекта в один или несколько двоичных Compiler Netlist Files

(файлов списка соединений компилятора) (.cnf). Из-за того, что экстрактор списка соединений компилятора разрешает использовать любые значения параметров в параметризованных функциях, содержимое CNF может изменяться при последующих компиляциях, если изменялись значения параметров. Экстрактор списка соединений компилятора также создает Hierarchy Interconnect File (файл иерархической схемы соединений) (.hif), который документирует иерархические соединения между файлами всего проекта и обеспечивает информацию, необходимую для того, чтобы показать иерархическое дерево проекта в дисплее иерархии проекта. Кроме того, экстрактор списка соединений компилятора создает Node Database File (файл базы данных цепей) (.ndb), который содержит названия цепей проекта для базы данных назначений ресурсов.

Встроенные модули ввода списков соединений EDIF, VHDL, Verilog и XNF автоматически преобразовывают информацию о проекте из входных файлов EDIF (.edf), файлов проектов VHDL (.vhd), файлов проектов Verilog (.v) и из файлов формата списка соединений Xilinx (.xnf) в некоторый формат, используемый САПР MAX+plus II. Модуль ввода списка соединений EDIF обрабатывает входные EDIF файлы с помощью файлов карты библиотек (.lmf), которые определяют соответствие между логическими функциями других САПР промышленного стандарта EDA и функциями MAX+plus II. Модуль ввода списка соединений XNF дополнительно создает Text Design Export File (текстовый файл экспорта проекта) (.tdx), который содержит проект на языке AHDL, эквивалентный проекту из файла формата списков соединений Xilinx (.xnf).

Database Builder (построитель базы данных)

Построитель базы данных использует HIF для связывания CNF, описывающих проект. Используя информацию из HIF файла, построитель базы данных копирует каждый CNF файл в единую, одноуровневую базу данных проекта. Таким образом, эта база данных содержит информацию об электрических соединениях проекта.

По мере создания базы данных построитель базы данных проверяет логическую законченность проекта и синтаксические ошибки (т.е. цепь без источника или приемника). Большинство ошибок обнаруживается и может быть легко исправлено на этой стадии обработки. Каждый модуль компилятора последовательно обрабатывает и обновляет эту базу данных.

Вы можете использовать возможность **smart recompile** (интеллектуальная перекомпиляция) для создания расширенной базы данных проекта, которая ускорит последующие компиляции. Эта база данных позволит вам при изменении назначений физических ресурсов микросхемы, например, назначения контактов или логических ячеек, пере-

компилировать проект без повторного построения базы данных и синтеза логики проекта. С помощью функции **total recompile** (полная перекомпиляция) вы можете полностью перекомпилировать проект.

Logic Synthesizer (логический синтезатор)

Модуль логического синтезатора применяет ряд алгоритмов, устраняющих избыточную логику и оптимально использующих ресурсы микросхемы определенного семейства, при реализации вашего проекта. При этом он старается достичь заданных пользователем требований к проекту. Кроме того, логический синтезатор ищет несвязанные цепи. Если он их находит, то удаляет все элементы, связанные с этой цепью.

Используя параметры или один из трех стилей синтеза, вы можете управлять результатами логического синтеза.

Вы можете задать параметры и определить стили логического синтеза из любого программного модуля MAX+plus II.

••• За более полной информацией обратитесь к *Global Project Timing Requirements* (требуемые характеристики быстродействия для текущего проекта) и *Global Project Logic Synthesis* (параметры логического синтеза для текущего проекта) на стр. 40.

Partitioner (разделитель проекта на части)

Если проект не разместить в одной микросхеме, разделитель проекта на части разбивает базу данных, обновленную логическим синтезатором, на части для размещения в нескольких микросхемах одного и того же семейства, пытаясь реализовать проект на минимальном количестве микросхем. Проект разделяется вдоль границ логических ячеек так, чтобы число внешних контактов было минимально.

Когда выполняются модули разделителя проекта на части и разводчика, вы можете приостановить компиляцию. Компилятор в этом случае выведет на экран информацию о текущем статусе процессов деления проекта на части и разводки, включая сравнение требуемых и доступных ресурсов так, чтобы вы могли решить, продолжать или нет компиляцию.

Fitter (разводчик)

Используя базу данных, обновленную разделителем проекта на части, разводчик согласовывает требуемые ресурсы для проекта с ресурсами одной или нескольких микросхем. Он назначает для каждого функционального модуля наилучшее местоположение логических ячеек и подбирает соответствующие пути соединений и внешние контакты для них. Разводчик пытается согласовать ваши назначения ресурсов из файла *Assignment & Configuration* (назначений и конфигурации) (**.acf**) с доступными ресурсами. Опции разводчика позволяют вам задавать

разные алгоритмы для достижения лучшей разводки, например, посредством автоматической вставки логических ячеек или ограничения количества входных связей ячеек. Если разводчик не смог развести проект, то он предлагает вам отклонить часть, либо все ваши назначения или завершить компиляцию.

Независимо от того, разведен проект или нет, разводчик создает Report File (файл отчета) (.rpt), в котором документируется информация о разводке, названия входных и выходных контактов, временные параметры проекта и неиспользованные ресурсы для каждой микросхемы проекта. Вы можете дополнительно включить разделы файла отчета, которые показывают назначения пользователя, файловую иерархию, соединения логических ячеек и логические уравнения.

Компилятор также автоматически создает Fit File (файл разводки) (.fit), который документирует назначения ресурсов микросхемы для всего проекта и информацию о трассировке. Независимо от того, успешно или нет завершилась разводка, вы можете просмотреть ее результаты с помощью редактора физического размещения. Вы также можете скопировать результаты последней разводки из FIT файла в ACF файл проекта для редактирования с помощью команды **Back-annotate**.

Вы можете дополнительно дать команду разводчику создать AHDL Text Design Output Files (AHDL текстовые выходные файлы проекта) (.tdo) для оптимально разведенного проекта. Так как для каждой микросхемы создается свой файл проекта, то проект, реализованный на нескольких микросхемах, вы можете разделить на несколько проектов, соответствующих этим микросхемам. Затем вы можете редактировать логику для отдельной микросхемы, сохраняя TDO файл для этой микросхемы в качестве текстового файла проекта (.tdf). При перекомпиляции логики для этой микросхемы сохранятся результаты предыдущей компиляции для остальной части проекта.

Functional SNF Extractor (функциональный SNF экстрактор)

Функциональный SNF экстрактор создает функциональный файл списка соединений симулятора (.snf), необходимый для функционального моделирования. Компилятор создает этот файл перед синтезом проекта, следовательно, он содержит все цепи, представленные в первоначальных файлах проектов. Функциональный SNF файл не содержит информации о временных параметрах, но быстро генерируется. Этот файл создается только в том случае, если проект компилируется без ошибок.

Timing SNF Extractor (временной SNF экстрактор)

Временной SNF экстрактор создает временной файл списка соединений симулятора (.snf), который содержит временные параметры проекта. Этот файл используется для временного моделирования и для

временного анализа. Генераторы списков соединений EDIF, Verilog и VHDL также используют временные SNF файлы и SDO файлы для создания EDIF, Verilog HDL и VHDL выходных файлов. Временной SNF файл создается только в том случае, если проект компилируется без ошибок.

Вы можете дать указание компилятору создать оптимизированный SNF файл, содержащий динамические модели для различных типов комбинаторной логики. Создание оптимизированного SNF файла увеличивает время компиляции, но он может сберечь ваше время в процессе моделирования и временного анализа.

Linked SNF Extractor (объединенный SNF экстрактор)

Объединенный SNF экстрактор создает объединенный файл списка соединений симулятора (.snf), который содержит функциональные и временные параметры для моделирования нескольких проектов, размещенных на одной плате. Этот файл объединяет информацию из временных SNF или функциональных SNF файлов нескольких проектов. Проекты, которые объединяются, могут использовать микросхемы из различных семейств. Если объединенный SNF файл содержит информацию о временных параметрах всех проектов, то вы можете использовать его для выполнения временного анализа. Такой файл создается только в том случае, если проект компилируется без ошибок.

☛ *За более подробной информацией о симуляторе MAX+plus II обратитесь к разделу "Проверка проекта" на стр. 74 или к справочной системе MAX+plus II.*

EDIF Netlist Writer (генератор EDIF списка соединений)

Компилятор MAX+plus II может работать с большинством других САПР промышленного стандарта CAE, которые могут считывать файлы списков соединений в стандартном формате EDIF 200 или 300. Генератор EDIF списка соединений создает один или несколько выходных файлов EDIF (.edo), содержащих функциональную и временную информацию после синтеза проекта. Информация о временных параметрах также может быть записана в выходные файлы стандартного формата задержки с расширением (.sdo). Эти файлы могут использоваться симуляторами промышленного стандарта. Выходные файлы EDIF и SDO создаются только в том случае, если проект компилируется без ошибок.

Verilog Netlist Writer (генератор Verilog списка соединений)

Генератор Verilog списка соединений создает один или несколько выходных файлов Verilog (.vo), содержащих функциональную и временную информацию после синтеза проекта. Информация о временных

параметрах также может быть записана в выходные файлы стандартного формата задержки с расширением (.sdo). Эти файлы могут использоваться симуляторами промышленного стандарта Verilog HDL. Выходные файлы Verilog HDL и SDO создаются только в том случае, если проект компилируется без ошибок.

VHDL Netlist Writer (генератор VHDL списка соединений)

Генератор VHDL списка соединений создает один или несколько выходных файлов VHDL (.vho) в синтаксисе VHDL 1987 или 1993, содержащих функциональную и временную информацию после синтеза проекта. Информация о временных параметрах также может быть записана в выходные файлы стандартного формата задержки с расширением (.sdo). Эти файлы могут использоваться симуляторами промышленного стандарта VHDL. Выходные файлы VHDL и SDO создаются только в том случае, если проект компилируется без ошибок.

Assembler (сборщик файла программирования)

Сборщик файла программирования преобразует сделанные разводчиком назначения логических ячеек, контактов и микросхемы в загрузочный модуль для программирования микросхемы (микросхем) в виде одного или нескольких двоичных Programmer Object Files (объектных файлов программатора) (.pof) или SRAM Object Files (объектных файлов SRAM) (.sof). Для некоторых микросхем компилятор также создает JEDEC файлы (.jed), Tabular Text Files (табличные текстовые файлы) (.tff) и Hexadecimal (Intel-format) Files (шестнадцатичные файлы) (.hex). Файлы POF, SOF и JEDEC используются средствами программирования фирмы Altera или других фирм для программирования или конфигурирования микросхем. Файлы HEX и TTF могут использоваться для конфигурирования микросхем FLEX 6000, FLEX 8000, FLEX 10K и ACEX1K другими способами. Сборщик файла программирования создает файлы программирования только в том случае, если проект компилируется без ошибок.

После завершения компиляции компилятор и программатор MAX+plus II позволяют вам создавать дополнительные файлы программирования для использования с другими средствами программирования. Например, вы можете создать Serial Bitstream Files (последовательные файлы битовых потоков) (.sbf) и Raw Binary Files (необработанные двоичные файлы) (.rbf) для конфигурирования микросхем FLEX 6000, FLEX 8000, FLEX 10K и ACEX1K. Вы также можете создавать Serial Vector Format Files (последовательные векторные файлы) (.svf) и Jam файлы (.jam) для программирования микросхем на оборудовании с автоматическим тестированием или под управлением встраиваемого процессора, соответственно.

Design Doctor Utility (утилита анализатора соответствия проекта правилам синтеза)



Утилита анализатора соответствия проекта правилам синтеза проверяет каждый файл проекта на наличие в нем структур, могущих вызвать проблемы функциональной надежности на уровне системы. Обычно они выявляются после того, как проект пошел в производство. Вы можете выбрать одну из трех predetermined настроек или сами задать параметры для анализатора соответствия проекта правилам синтеза.

Нарушение правил синтеза может происходить при наличии в проекте несинхронизированных входов, пульсаций тактовых импульсов, многоуровневой логики на синхростоках, входах асинхронной установки и сброса, гонок сигналов. Справочная система MAX+plus II объясняет, какие могут быть нарушения правил синтеза, чтобы помочь вам определить, что нужно изменить в данном файле проекта.

 *За более полной информацией обо всех функциях и возможностях компилятора обратитесь к справочной системе MAX+plus II.*

1.11. Обнаружение ошибок и определение их местонахождения



Процессор сообщений MAX+plus II (рис. 1.20) имеет связь со всеми программными модулями MAX+plus II, выдающими сообщения об ошибках, информационные сообщения и предупреждения по мере того, как вы обрабатываете, проверяете или программируете проект. Если вы дважды щелкните на сообщении кнопку 1 мыши, то процессор сообщений автоматически откроет файл проекта, вспомогательный файл или физическое размещение проекта, которые содержат источник сообщения, и выделит источник ошибки.

Процессор сообщений предоставляет следующие возможности:

- если сообщение имеет несколько источников, то процессор сообщений позволяет вам найти местоположение каждого из них;
- вы можете определить местонахождение источника сообщения в файлах проектов и во вспомогательных файлах (например, в SCF файле), или на физическом уровне в редакторе физического размещения;
- когда вы выводите список путей сигналов с помощью кнопки **List Paths** во временном анализаторе MAX+plus II, процессор

- сообщений выводит на экран сообщения, показывающие все задержки распространения между парами цепей;
- если вы хотите определить местонахождение сообщений на физическом уровне размещения проекта, то вы можете определить местонахождение каждого их источников сообщения последовательно, или определить местонахождение всех источников одновременно. Определение местонахождения всех источников одновременно — удобный способ для просмотра критических временных путей;
 - вы можете печатать текущие сообщения или сохранить их в ASCII Message Text File (текстовом файле сообщений ASCII) (.mtf);
 - если сообщение выделено в окне процессора сообщений и вы выберете кнопку **Help on Message** (справка о сообщении), то вы получаете подробную информацию о возможной причине сообщения и о действиях по ее устранению.

Окно процессора сообщений открывается во время компиляции. если генерируется сообщение

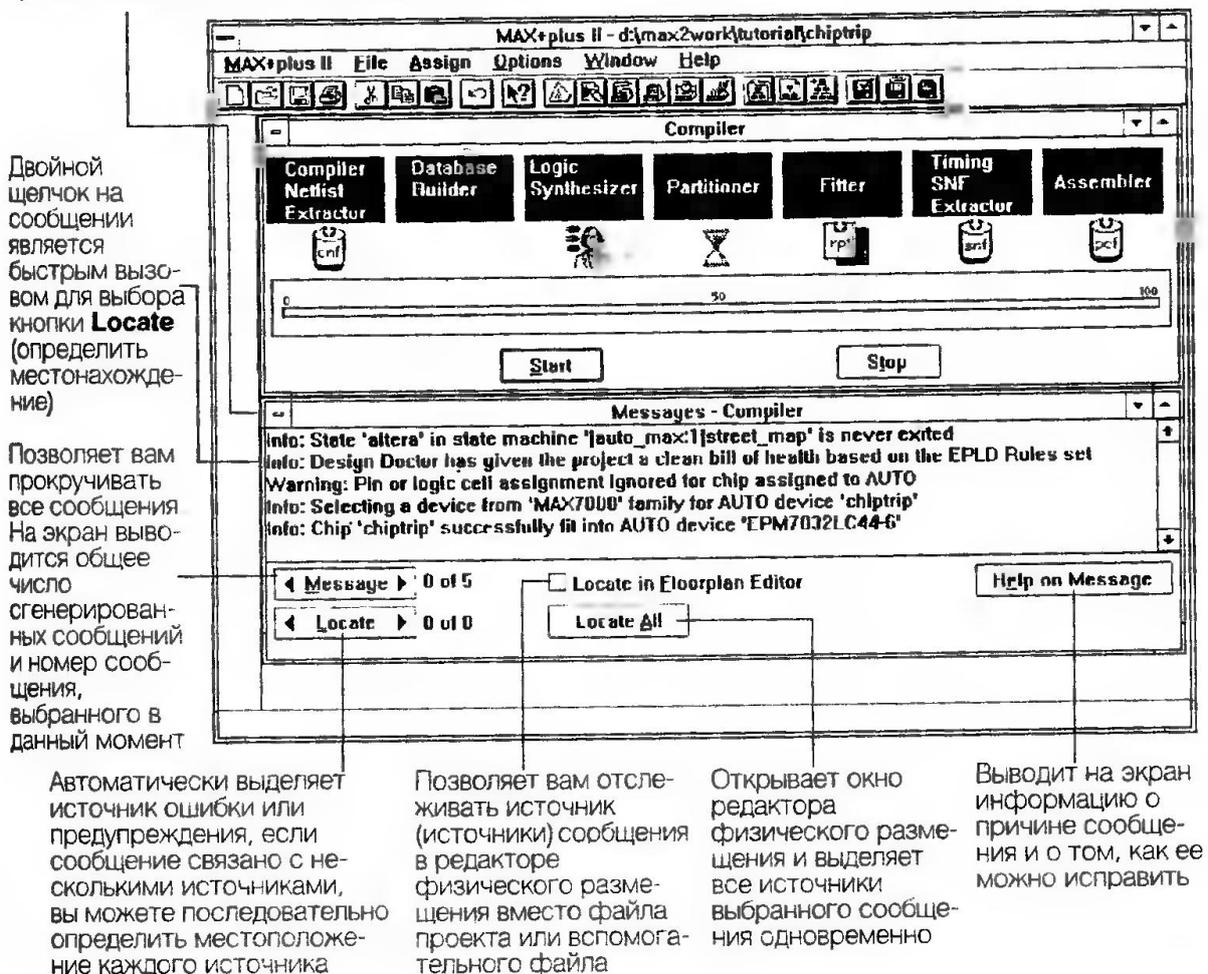


Рис. 1.20

За полной информацией о функциях и возможностях процессора сообщений обратитесь к справочной системе MAX+plus II.

1.12. Проверка проекта

Вы можете проверить скомпилированный проект с помощью трех программных модулей MAX+plus II Simulator (симулятор), Timing Analyzer (временной анализатор) и Waveform Editor (редактор временных диаграмм) (рис. 1.21).

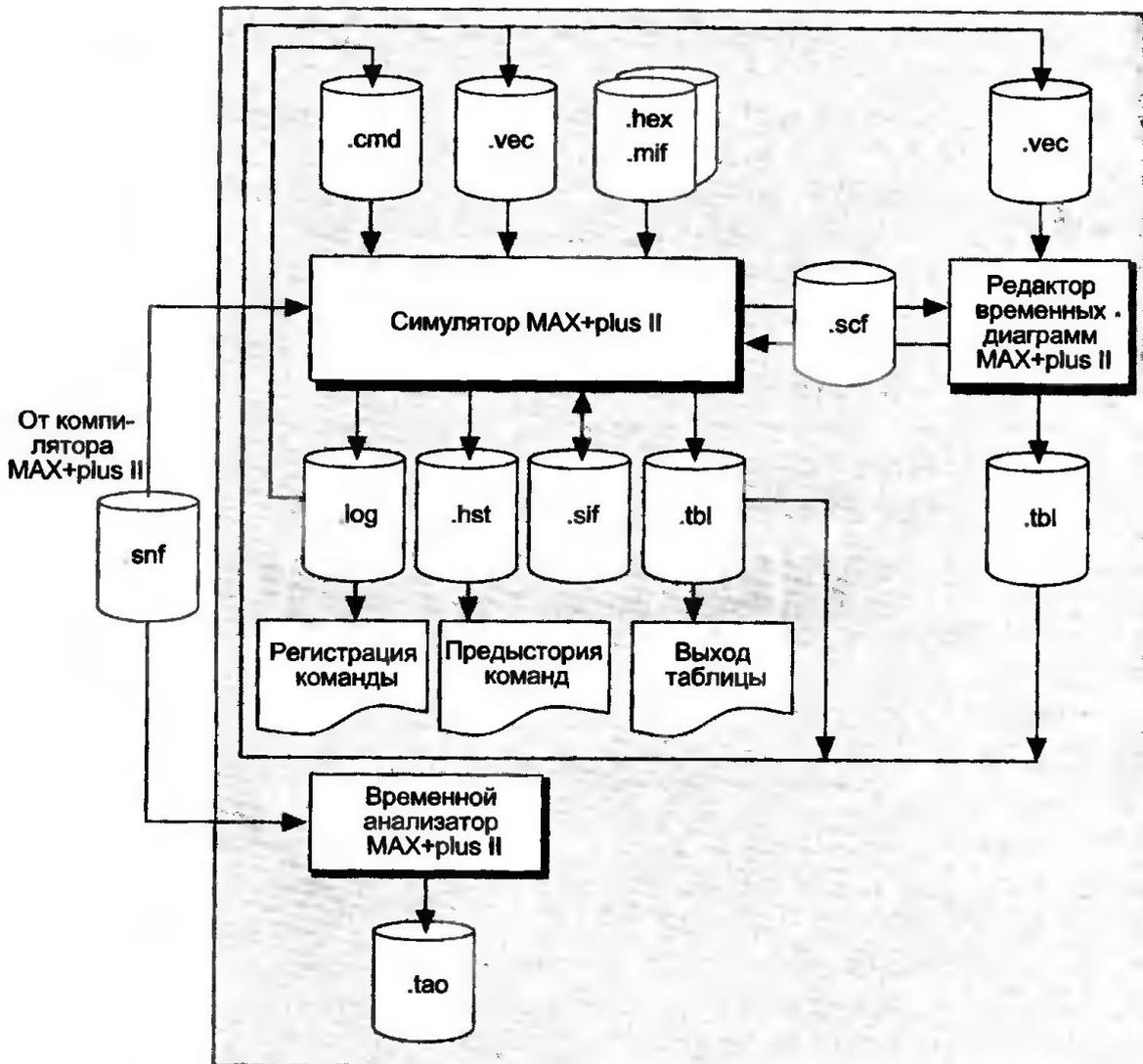


Рис. 1.21

1.12.1. Симулятор MAX+plus II



Симулятор MAX+plus II проверяет логические операции и внутренние временные соотношения проекта, позволяя вам смоделировать работу проекта перед тем, как запрограммировать его в микросхему. Вы можете запустить симулятор либо в интерактивном режиме, либо в режиме пакетной обработки. На рис. 1.22 показано окно симулятора.

Временной SNF, созданный во время компиляции, загружается автоматически

Когда вы первый раз открываете симулятор, SCF или векторный файл с тем же именем, что и проект, загружается автоматически

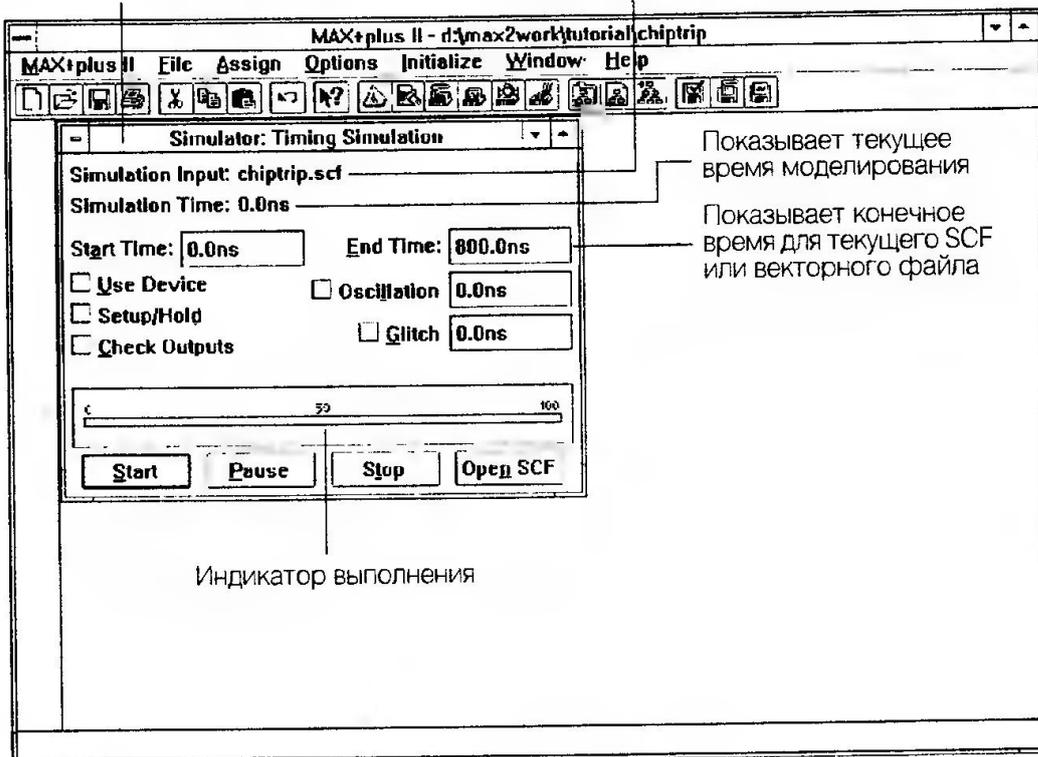


Рис. 1.22

Для моделирования проекта вы должны сначала скомпилировать его и дать указание компилятору создать файл списка соединений симулятора (.snf) для функционального, временного или объединенного моделирования. Затем функциональный, временной или объединенный SNF файл для текущего проекта загружается автоматически при открытии симулятора.

Симулятор использует графический файл временных диаграмм Simulator Channel File (.scf) или векторный файл ASCII (.vec) в качестве источника входных сигналов. Для проектов, содержащих память, вы можете назначить ее исходное значение с помощью Hexadecimal Intel-format Files (шестнадцатиричных файлов) (.hex) или Memory Initialization Files (файлов инициализации памяти) (.mif). Редактор временных диаграмм может автоматически создавать SCF файл по умолчанию, который вы затем можете редактировать для задания желаемых входных сигналов. Если вы создали векторный файл, то симулятор автоматически создает из него SCF файл.

••• Для получения дополнительной информации о формате векторного файла обратитесь к *Vector File Format* в справочной системе MAX+plus II. Вы также можете обратиться к разделу "Занятие 10: Моделирование проекта" на стр. 171 в самоучителе по MAX+plus II.

Симулятор позволяет вам сравнить значения выходных сигналов, полученных в результате моделирования, с их значениями в SCF файле, например, определенные пользователем ожидаемые значения выходных сигналов или их значения от предыдущего моделирования. С помощью некоторых средств программирования вы можете выполнить функциональное тестирование микросхемы, при котором будут сравниваться выходные сигналы на контактах микросхемы с их значениями, полученными при моделировании.

Используя возможности симулятора, вы можете проверить ваш проект на наличие импульсных выбросов, нестабильных сигналов или нарушений времен предустановки и удержания. Как только моделирование закончено, вы можете открыть редактор временных диаграмм для просмотра обновленного SCF файла или сохранения значения выходных сигналов в Table File (табличном файле) (.tbl). Для просмотра TBL файлов используется текстовый редактор.

Функциональное моделирование

Компилятор MAX+plus II создает функциональный SNF файл перед тем, как он синтезирует проект. Следовательно, при функциональном моделировании все введенные в проекте цепи могут быть промоделированы.

Во время функционального моделирования симулятор игнорирует все задержки распространения. Поэтому выходные сигналы изменяются одновременно с входными.

Временное моделирование

Компилятор MAX+plus II создает временной SNF файл после того, как проект полностью синтезирован и оптимизирован. Следовательно, временной SNF файл содержит только те цепи, которые не были исключены во время логического синтеза.

Симулятор использует для моделирования проекта информацию из временного SNF файла, который создается с учетом параметров микросхемы. Эти параметры берутся из Device Model Files (файлов моделей микросхем) (.dmf), входящих в состав САПР MAX+plus II.

Если проект уже разделен на две или несколько микросхем, компилятор создает SNF файлы для проекта в целом и для каждой микросхемы. Однако временное моделирование выполняется только для всего проекта.

Вы можете ускорить временное моделирование, давая компилятору указание создать оптимизированный SNF файл, содержащий динамические модели, представляющие различные типы комбинаторной логики. При этом время компиляции увеличивается, однако получившийся в результате SNF файл может уменьшить время моделирования, так как симулятор может обратиться к динамической модели вместо интерпретации всей логики в комбинаторной схеме.

Объединенное моделирование

Когда компилятор MAX+plus II создает объединенный SNF файл, он объединяет функциональные и/или временные SNF файлы для нескольких отдельных проектов. Отдельные "подпроекты" в объединенном SNF файле могут быть реализованы на микросхемах фирмы Altera из различных семейств. Из-за того, что функциональные SNF скомпилированы не полностью, вы можете добавить подпроекты, представляющие логику, реализованную на микросхемах других фирм.

Вы можете использовать объединенный SNF файл для выполнения моделирования на уровне платы. Если объединенный SNF файл содержит только временную информацию, вы можете использовать его для выполнения временного анализа с помощью временного анализатора MAX+plus II.

Основные особенности симулятора

Вместе с другими программными модулями MAX+plus II, симулятор позволяет вам выполнять следующие действия:

- назначать ожидаемые значения выходных сигналов, чтобы сравнить их с результатами моделирования;
- моделировать отдельные или групповые цепи. Вы можете объединить биты, кодирующие состояния конечного автомата в группу, и ссылаться на них с помощью имен состояний автомата;
- определять интервалы времени, на которых появляются импульсные выбросы или нестабильные сигналы;
- регистрировать нарушения времен предустановки и удержания для триггеров;
- выполнять функциональное тестирование микросхемы. Вы можете проверить, эквивалентны ли функционально выходные сигналы, полученные в результате моделирования, и сигналы на контактах микросхемы;
- задавать условия для контрольных точек, вызывающих приостановку симулятора, когда эти условия выполняются;
- составлять список имен цепей и их начальных значений перед моделированием;
- задавать начальные значения для блоков памяти (RAM или ROM) перед моделированием;
- сохранять начальные значения для цепей и памяти в Simulator Initialization File (файле начальных значений симулятора) (.sif) или загружать их из него;
- сохранять все команды симулятора, которые вы выполняете в интерактивном режиме, в ASCII Log File (файле регистрации) (.log), который имеет такой же формат, как Command File (командный файл) (.cmd), используемый при работе симулятора

в пакетном режиме. С помощью файла регистрации вы можете повторить предыдущее моделирование. Кроме этого, вы можете записать команды симулятора и их действия в ASCII History File (файл истории) (.hst).

- За более полной информацией обо всех функциях и возможностях симулятора MAX+plus II обратитесь к справочной системе MAX+plus II.

1.12.2. Редактор временных диаграмм MAX+plus II



Редактор временных диаграмм MAX+plus II (рис. 1.23) используется в качестве инструмента ввода проекта и в качестве инструмента для задания значений входных сигналов и просмотра результатов моделирования. Вы можете создать Waveform Design Files (файлы проектов, описанных временными диаграммами) (.wdf), содержащие логику проекта, и Simulator Channel Files (файлы временных диаграмм) (.scf), содержащие входные вектора для моделирования.

Поле **Name**, показывающее названия цепей и групп

Значок, изменяющий размеры полей

Неопределенный (X) логический уровень выходных и внутренних цепей

Метка входной цепи

Метка выходной цепи

Метка групповой внутренней цепи

H показывает, что для этой группы основание системы счисления — HEX

Для группы сигналов показывается их значение в выбранной системе счисления

Name	Value	250.0ns	500.0ns	750.0ns
reset	0			
enable	0			
dir1	0			
dir0	0			
clock	0			
accel	0			
at_altera	X			
f time_cnt:4 count[7..0]	H XX		XX	
auto_max:1 street_map	H X		X	
l speed_ch 2 speed	H X		X	

Рис. 1.23

Для моделирования проекта вы должны задать значения входных сигналов. Это можно сделать графически с помощью временных диаграмм, заданных в SCF файле, или с помощью текстового описания в ASCII Vector File (векторном файле) (.vec).

В SCF файле вы задаете временные диаграммы входных сигналов, которые будут управлять моделированием, и названия внутренних и выходных цепей, сигналы в которых будут смоделированы. До процесса моделирования внутренние и выходные цепи имеют неопределенные логические уровни, но вы можете их отредактировать, чтобы задать ожидаемые значения этих сигналов.

Редактор временных диаграмм может использовать Simulator Netlist File (файл списка соединений симулятора) (.snf) для создания SCF файла по умолчанию, содержащего часть или все цепи в компилируемом проекте. Вы можете редактировать этот SCF файл в соответствии с вашими требованиями или вы можете создать SCF файл "с нуля". Кроме того, вы можете импортировать Vector File (векторный файл) для автоматического создания его графического эквивалента в виде временных диаграмм.

С помощью редактора временных диаграмм вы можете просмотреть выходные сигналы в SCF файле, полученные в результате моделирования. Симулятор создает SCF файл автоматически, если источником входных векторов является векторный файл, и наоборот, SCF, который был источником входных векторов, просто обновляется во время моделирования. Временные диаграммы внутренних и выходных цепей переписываются на основе моделирования входных сигналов.

За более подробной информацией обратитесь к разделу "Редактор временных диаграмм MAX+plus II" на стр. 48.

За полной информацией о редакторе временных диаграмм MAX+plus II обратитесь к справочной системе MAX+plus II.

1.12.3. Временной анализатор MAX+plus II



С помощью временного анализатора MAX+plus II (рис. 1.24) вы можете анализировать быстродействие проекта после того, как он оптимизирован компилятором. Вы можете отслеживать все пути сигналов в проекте, определяя пути с наибольшей задержкой распространения и пути, которые ограничивают быстродействие проекта.

Временной анализатор использует информацию из временного SNF файла, генерируемого компилятором. Временной анализатор может также использовать объединенный SNF файл, который связывает временные SNF файлы других проектов.

цепей по умолчанию, доступные для каждого типа анализа. Более того, вы можете выбрать для анализа пути с задержкой меньшей или большей определенного значения, отсечь пути, которые вас не интересуют, так чтобы в анализ включались только те сигналы, которые ведут к данной цепи. Вы также можете исключить из анализа пути задержки сигналов *I/O pin feedback* (обратная связь через контакт ввода-вывода), *Clear* (сброс), *Preset* (установка) и ограничить количество путей для синхросигнала в дисплее максимального быстрогодействия.

После того, как временной анализатор закончит анализ, вы можете выбрать источник или приемник сигнала и получить список всех, связанных с ними путей задержек. Процессор сообщений автоматически показывает выбранные цепи в исходном файле проекта или на физическом уровне.

Вы можете сохранить результаты временного анализа в Timing Analyzer Output File (выходном файле временного анализатора) (.tao).

 *За более полной информацией обо всех функциях и возможностях временного анализатора MAX+plus II обратитесь к справочной системе MAX+plus II.*

1.13. Программирование микросхемы

Altera предоставляет аппаратные и программные средства, необходимые для программирования, конфигурирования и верификации микросхем фирмы Altera (рис. 1.25). Аппаратные средства включают в себя дополнительную плату программатора (для 486 или Pentium PC), которая запускает Altera Master Programming Unit (MPU) (базовый модуль программатора фирмы Altera). MPU выполняет проверку на наличие электрического контакта между программирующим адаптером и микросхемой. С помощью соответствующего программирующего адаптера MPU также поддерживает функциональное тестирование микросхемы. Вы можете применять векторы, созданные для моделирования, к программируемой микросхеме для ее функционального тестирования.

Altera также поддерживает возможность In-system programmability (ISP) (программирования в системе) и In-circuit Reconfiguration (ICR) (переконфигурирования микросхемы) с помощью кабеля загрузки FLEX (для PC), параллельного кабеля загрузки ByteBlaster (для PC) и последовательного кабеля загрузки BitBlaster (для PC и рабочих станций UNIX). FLEX Download Cable может соединять адаптер для программирования конфигурационных микросхем, вставленный в MPU, с FLEX микросхемами на плате. ByteBlaster соединяет параллельный порт (порт принтера), а последовательный кабель загрузки BitBlaster соединяет стандартный RS-232 порт (называемый COM портом PC) с микросхе-

мами, распаянными на плате. Эти кабели позволяют вам программировать или конфигурировать одну или несколько ICR или ISP-совместимых микросхем во FLEX или JTAG цепочках (табл. 1.4).

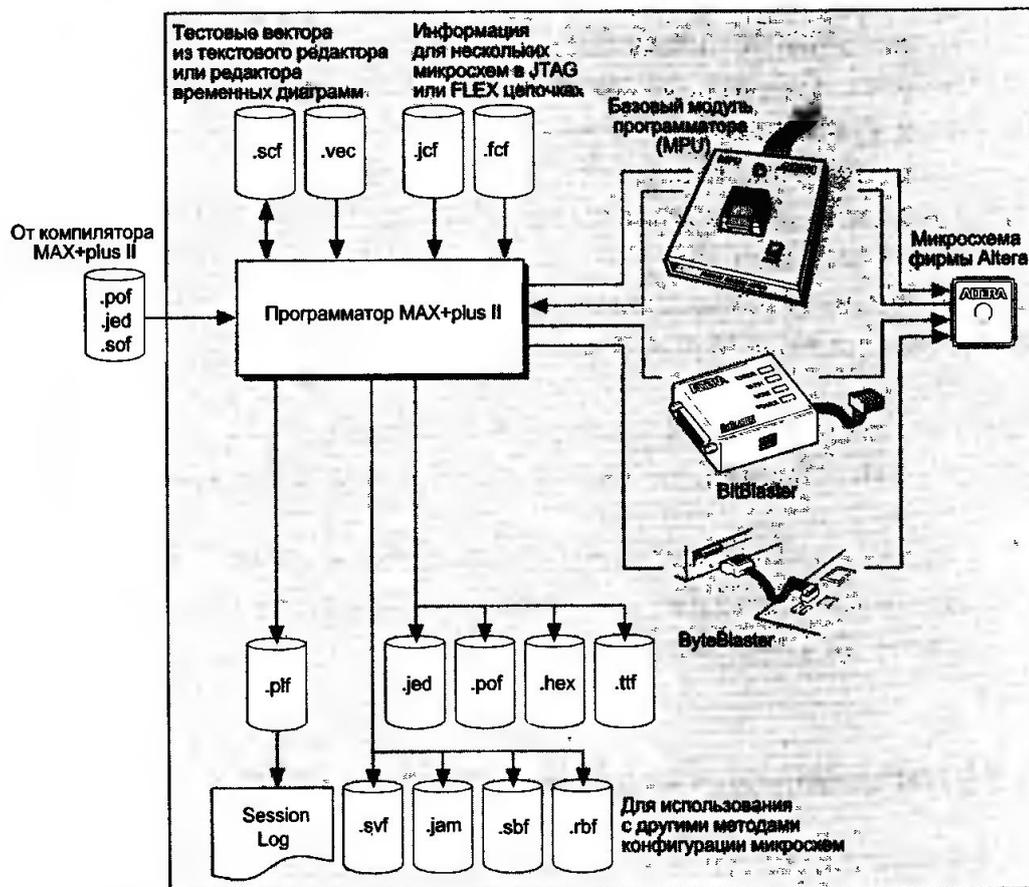


Рис. 1.25

Таблица 1.4

Аппаратные средства программирования фирмы Altera

Аппаратные средства программирования фирмы Altera	PC	Рабочая станция UNIX	Микросхемы Classic & MAX 5000	Микросхемы MAX 7000 (за исключением MAX 7000S)	Микросхемы MAX 7000S & MAX 9000	Микросхемы FLEX 6000, FLEX 8000 & FLEX 10K	Программирование/Реконфигурирование в системе
Дополнительная плата программатора, PL-MPU (базовый модуль программатора) и адаптеры для микросхем	X		X	X	X		
Кабель загрузки FLEX	X					X	X

Окончание табл. 1.4

Аппаратные средства программирования фирмы Altera	PC	Рабочая станция UNIX	Микросхемы Classic & MAX 5000	Микросхемы MAX 7000 (за исключением MAX 7000S)	Микросхемы MAX 7000S & MAX 9000	Микросхемы FLEX 6000, FLEX 8000 & FLEX 10K	Программирование/Реконфигурирование в системе
Кабель загрузки BitBlaster	X	X			X	X	X
Кабель загрузки ByteBlaster	X				X	X	X

••• За дополнительной информацией по конфигурированию микросхем семейств FLEX 6000, FLEX 8000 и FLEX 10K, выполненных по SRAM технологии, обратитесь к *Application Note 87 (конфигурирование микросхем FLEX 6000)*, *Application Note 33 (конфигурирование микросхем FLEX 8000)*, *Application Note 38 (конфигурирование нескольких микросхем FLEX 8000)*, *Application Note 59 (конфигурирование микросхем FLEX 10K)*.

1.13.1. Программатор MAX+plus II



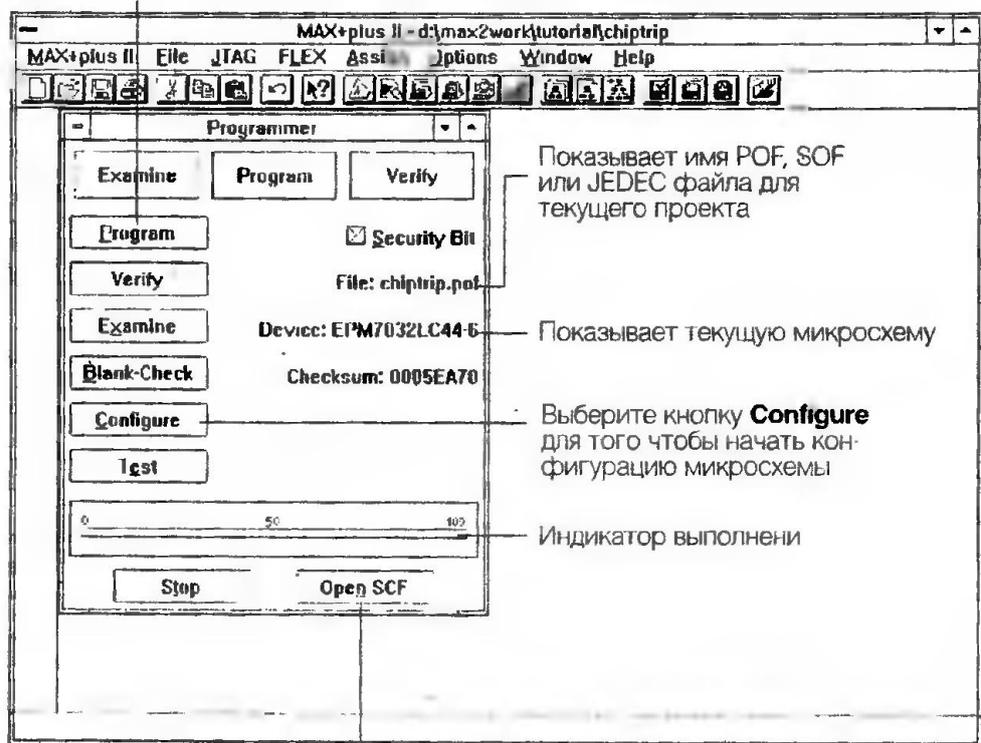
Программатор MAX+plus II, показанный на рис. 1.26, использует файлы программирования, созданные компилятором для программирования микросхем фирмы Altera. Он позволяет вам программировать (**Program**), конфигурировать (**Configure**), сравнить содержимое запрограммированной микросхемы с файлом программирования (**Verify**), считать содержимое микросхемы (**Examine**), проверять микросхемы на отсутствие записанной информации (**Blank-check**), проводить функциональное тестирование микросхемы (**Functionally test**).

Аппаратные средства программирования фирмы Altera дают вам возможность программировать микросхемы Altera с помощью файла (файлов) программирования, создаваемых компилятором. Когда вы открываете окно программатора, автоматически загружается файл программирования для текущего проекта. Программатор MAX+plus II принимает следующие форматы файлов:

- Programmer Object File (объектный файл программатора) (.pof) для программирования микросхем Altera Classic, MAX 3000, MAX 5000, MAX 7000 и MAX 9000, а также Configuration EPROM и FLASH, используемых для конфигурирования микросхем FLEX 6000, FLEX 8000, FLEX 10K и ACEX1K;

- SRAM Object File (объектный файл SRAM) (.sof) для конфигурирования микросхем Altera FLEX 6000, FLEX 8000, FLEX 10K и ACEX1K;
- JEDEC File (.jed) для программирования микросхем Altera Classic и микросхем EPM5016 и EPM5031 из семейства MAX 5000;
- JTAG Chain File (файл цепочки JTAG) (.jcf) описывает порядок, в котором POF, SOF и JEDEC файлы должны программировать или конфигурировать несколько микросхем на плате, соединенных посредством JTAG интерфейса;
- FLEX Chain File (файл цепочки FLEX) (.fcf), описывающий порядок, в котором SOF файлы должны конфигурировать несколько микросхем семейства FLEX, входящих во FLEX цепочку.

Выберите кнопку **Program** для того, чтобы начать программирование микросхемы



Открывает SCF файл, содержащий вектора для функционального тестирования

Рис. 1.26

Программатор и компилятор также могут создавать Hexadecimal (Intel-Format) Files (.hex), Tabular Text Files (.ttf), Serial BitBlaster File (.sbf), Raw Binary Files (.rbf), Serial Vector Format File (.svf) и Jam Files (.jam) для конфигурирования и программирования ISP- и ICR-совместимых микросхем другими способами программирования.

Для начала программирования или конфигурирования вы выберете кнопку **Program** или **Configure**. При возникновении любой ошибки или проблемы в процессе программирования или конфигурирования в окне процессора сообщений появляется список сообщений. По каждому сообщению вы можете получить справку о вызвавших его причинах.

Программатор MAX+plus II выполняет следующие действия:

- программирует данные из POF или JEDEC файлов в "чистые" микросхемы семейств Classic, MAX 5000, MAX 7000 или MAX 9000. После этого микросхемы работают согласно вашему проекту. При использовании режима JTAG-цепочек, несколько микросхем программируются, используя дополнительную информацию из JCF файла;
- загружает данные конфигурации из SRAM Object File (объектного файла SRAM) (.sof) или JEDEC File (.jed) для конфигурирования микросхем семейств FLEX 6000, FLEX 8000 или FLEX 10K. При использовании режима JTAG-цепочек несколько микросхем программируются, используя дополнительную информацию из JCF файла. При использовании FLEX цепочек микросхемы конфигурируются, используя дополнительную информацию из FCF файла;
- создает и читает JCF и FCF файлы, которые описывают порядок микросхем при программировании или конфигурировании микросхем в JTAG или FLEX цепочках соответственно;
- преобразует формат файла POF в JEDEC и, наоборот, сохраняет в файле вектора для функционального тестирования микросхем, так что вы можете программировать и тестировать микросхему аппаратными и программными средствами других фирм;
- сравнивает содержимое файла программирования с содержимым запрограммированной микросхемы;
- считывает содержимое запрограммированной микросхемы и сохраняет его в POF или JEDEC файлах;
- осуществляет проверку на отсутствие записанной информации, чтобы убедиться, что микросхема "чистая" или полностью "стерта";
- включает и выключает установку бита защиты перед тем, как проект программируется в микросхему. Когда бит защиты включен, информация из микросхемы не может быть прочитана. Микросхемы, выполненные по EPROM технологии, также не могут быть перепрограммированы;
- тестирует запрограммированную микросхему с помощью входных векторов, которые могут находиться в текущем файле программирования, в SCF или Векторном файле, и сообщает, являются ли логические уровни выходных сигналов в файле функционально эквивалентными сигналам на контактах микросхемы;
- создает Programmer Log File (файл журнала программатора) (.plf), который записывает команды и сообщения во время программирования для использования их при последующем программировании.

 За полной информацией обо всех функциях и возможностях программатора обратитесь к справочной системе MAX+plus II.

1.14. Режим командной строки MAX+plus II

Вы можете запустить **Compiler** (компилятор), **Timing Analyzer** (анализатор временных параметров) и **Simulator** (симулятор) с помощью командной строки в операционных системах UNIX, Microsoft Windows NT 4.0, Microsoft Windows 98 и Windows 2000.

Для запуска MAX+plus II с помощью командной строки, наберите:

```
maxplus2 -h | -v | {<batch option(s)> (пакетная опция или опции)}[<I/O option(s)> (опция или опции ввода/вывода)]<project name> (название проекта)>} ←
```

Для одного проекта могут использоваться несколько пакетных опций (**batch options**) и опций ввода/вывода (**I/O options**). Несколько проектов могут обрабатываться с помощью одной и той же командной строки. <project name> (название проекта) указывает, что опции для данного проекта закончились.

Возможны следующие <batch options> (пакетные опции):

Batch Options (пакетные опции):	Описание работы:
-h <i>или</i> -help	Выводит на экран список опций, доступных в командной строке
-v <i>или</i> -version	Выводит на экран номер версии MAX+plus II
-c <i>или</i> -compile	Запускает Compiler (компилятор)
-ta_delay	Запускает Timing Analyzer (анализатор временных параметров) для расчета Delay Matrix (матрицы задержек)
-ta_setup	Запускает Timing Analyzer (анализатор временных параметров) для расчета Setup/Hold Matrix (матрицы времен предустановки и удержания сигналов)
-ta_reg	Запускает Timing Analyzer (анализатор временных параметров) для расчета Registered Performance (быстродействия регистровой логики)
-s <i>или</i> -simulate	Запускает Simulator (симулятор)
-i <i>или</i> -ignore_errors	Игнорирует ошибки Compiler (компилятора), Simulator (симулятора), Timing Analyzer (анализатора временных параметров) и продолжает обработку других проектов, указанных в командной строке, даже если предыдущий проект не был обработан.

Доступные <I/O options> (опции ввода/вывода) показаны ниже. Если вы укажете в опции пустые кавычки ("") для <filename> (имя

файла)>, то это будет означать, что <filename (имя файла)> соответствует <project name (имени проекта)>.

I/O Options (опции ввода/вывода):	Описание работы:
-tao "<filename (имя файла)>"	Сохраняет результаты работы Timing Analyzer (анализатора временных параметров) в файле <имя файла>.tao. Если эта опция не используется, то файл <имя проекта>.tao генерируется автоматически.
-scf "<filename (имя файла)>"	Использует файл <имя файла>.scf, как источник входных векторов при моделировании. Если эта опция не указана, то используется файл с именем, заданным переменной SIMULATION_INPUT_FILE в файле <имя проекта>.acf.
-vec "<filename (имя файла)>"	Использует файл <имя файла>.vec в качестве источника входных векторов при моделировании.
-cmd "<filename (имя файла)>"	Использует файл <имя файла>.cmd в качестве источника команд при моделировании.
-tbl "<filename (имя файла)>"	Сохраняет результаты работы Simulator (симулятора) в файле <имя файла>.tbl.
-hst "<filename (имя файла)>"	Записывает "историю" работы Simulator (симулятора) в файл <имя файла>.hst.

 Вы можете использовать символы > или >> для записи предупреждений и сообщений об ошибках в ASCII файл. (В UNIX используйте >! или >>! вместо > или >>, если установлена переменная noclobber).

Ниже приведен пример командной строки, с помощью которой компилируется проект **upcntr**, компилируется проект **chiptrip**, запускается **Timing Analysis** (анализатор временных параметров) для определения **Registered Performance** (быстродействия регистровой логики) проекта **chiptrip** и моделируется проект **chiptrip**, используя файл **test.scf** в качестве источника входных векторов:

```
maxplus2 -c upcntr -c -ta_reg -s -scf "test.scf" chiptrip ↵
```

Следующий пример компилирует проект **chiptrip** и записывает все сообщения в файл **message.out**:

```
maxplus2 -c chiptrip > message.out ↵
```