

Министерство образования и науки Российской Федерации
Государственный университет аэрокосмического приборостроения
кафедра "Радиотехнические системы"

Терминология проектирования цифровых устройств на ПЛИС

учебное пособие по дисциплинам

"Программируемые логические схемы",
"Устройства цифровой обработки сигналов"

Санкт-Петербург
2005

Пособие подготовлено на кафедре радиотехнических систем Санкт-Петербургского Государственного университета аэрокосмического приборостроения.

Авторы:

доцент, к.т.н. Роман Александрович Мяльк,
ассистент Владислав Юрьевич Шестаков.

При подготовке пособия использованы следующие материалы:

- Комолов Д.А., Мяльк Р.А., Зобенко А.А., Филлипов А.С. Системы автоматизированного проектирования фирмы Altera Max+Plus II и Quartus II. Краткое описание и самоучитель. – М.: ИП РадиоСофт, 2002 – 352 с.: ил.
- Угюмов Е.П. Цифровая схемотехника. – СПб.: БХВ, 2000.

Содержание

A	4
B	8
C	11
D	15
E	19
F	22
G	26
H	29
I	31
J	34
K	35
L	36
M	40
N	47
O	50
P	52
R	58
S	60
T	67
U	72
V	72
W	75
X	75

A

ACEX 1K

Семейство микросхем фирмы Altera для реализации проектов, требующих низкой стоимости конечного устройства. Это семейство выполнено по SRAM технологии, имеет высокую производительность, достаточное количество ресурсов и встроенную память для реализации сложных проектов. Семейство ACEX 1K включает следующие микросхемы EP1K10, EP1K30, EP1K50 и EP1K100.

Микросхемы семейств ACEX 1K, FLEX 10K и FLEX 6000 могут конфигурироваться совместно, используя FLEX chain (FLEX цепочки) или данные, хранящиеся в одной и той же конфигурационной микросхеме.

ACF

См. Assignment&Configuration File (Файл назначений и конфигурации).

active-high node

Активный высокий уровень сигнала для цепи, узла или контакта. Цепь, узел или контакт, который активизируется высоким уровнем сигнала . 1 в AHDL и Verilog HDL, '1' в VHDL или VCC. Например, `ena`, `clk`.

active-low node

Активный низкий уровень сигнала для цепи, узла или контакта) Цепь, узел или контакт, который активизируется низким уровнем сигнала . 0 в AHDL и Verilog HDL, '0' в VHDL или GND. Например, `clrn`, `prn`, `oen`. В AHDL design files (Проектные файлы на языке AHDL) цепям с активным низким уровнем следует назначать значение по умолчанию VCC с помощью оператора Default Statement.

ADF

См. Altera Design File.

AHDL

См. Altera Hardware Description Language.

Altera Design File

Проектный файл фирмы Altera с расширением (**.adf**). Формат ASCII файла для ввода Boolean equation (Булевых уравнений) в САПР A+PLUS фирмы Altera. ADF файл использует netlist формат и булевы уравнения для описания проекта. Компилятор MAX+PLUS II автоматически преобразует ADF файл в Compiler Netlist File (Файл списка соединений компилятора) с расширением (**.cnf**) во время компиляции проекта.

ADF файл также генерируется при компиляции State Machine File (Файла конечного автомата) с расширением (**.smf**).

Altera Hardware Description Language (AHDL)

Язык описания цифровых устройств фирмы Altera. Модульный язык высокого уровня, полностью интегрированный в САПР MAX+PLUS II. Вы можете создавать AHDL Text Design File (Текстовый проектный файл на языке AHDL) с расширением (**.tdf**), используя MAX+PLUS II Text Editor (Текстовый редактор MAX+PLUS II) или любой другой текстовый редактор, затем компилировать, моделировать и запрограммировать ваши проекты в микросхемы фирмы Altera. Язык AHDL позволяет использовать булевы уравнения, описания конечных автоматов и логики, поставляемую Library of Parameterized Modules (LPM) (Библиотеку параметризованных модулей) и создавать свои собственные параметризованные функции.

MAX+PLUSII Compiler (Компилятор) может создавать Design Output Files (Выходные проектные файлы) с расширением (**.tdo**) на языке AHDL.

Altera Megafunction Partner Program (AMPP)

Партнёрская программа по созданию мегафункций для микросхем фирмы Altera) — программа, предоставляющая её участникам поддержку от фирмы Altera по созданию и распространению мегафункций для микросхем фирмы Altera.

auxiliary file

Вспомогательный файл. Файл, который связан с проектом, созданным с помощью САПР MAX+PLUS II, но не является design file (Проектным файлом) в иерархическом дереве проекта. Большинство вспомогательных файлов не содержат описания проекта. Редактируемые пользователем вспомогательные файлы с таким же именем, как и проект, можно увидеть в окне Hierarchy Display (Дисплея иерархии проекта). Ниже приведён список редактируемых вспомогательных файлов.

Assignment&Configuration File (**.act**)

Assignment&Configuration Output File (**.aco**)

Command File (**.cmd**)

EDIF Command File (**.edc**)

Fit File (**.fit**)

FLEX Chain File (**.fcf**)

Hexadecimal (Intel-format) File (**.hex**)

History File (**.hst**)

Include File (**.inc**)

Jam File (**.jam**)

JTAG Chain File (**.jcf**)

Library Mapping File (**.lmf**)

Log File (**.log**)

Memory Initialization File (**.mif**)
Memory Initialization Output File (**.mio**)
Message Text File (**.mtf**)
Programmer Log File (**.plf**)
Report File (**.rpt**)
Serial Vector Format File (**.svf**)
Simulator Channel File (**.scf**)
Standard Delay Format (SDF) Output File (**.sdo**)
Symbol File (**.sym**)
Table File (**.tbl**)
Tabular Text File (**.ttf**)
Text Design Output File (**.tdo**)
Timing Analyzer Output File (**.tao**)
Vector File (**.vec**)
VHDL Memory Model Output File (**.vmo**)

Не редактируемые вспомогательные файлы:

Compiler Netlist File (**.cnf**)
Hierarchy Interconnect File (**.hif**)
JEDEC File (**.jed**)
Node Database File (**.ndb**)
Programmer Object File (**.pof**)
Raw Binary File (**.rbf**)
Serial Bitstream File (**.sbf**)
Simulator Initialization File (**.sif**)
Simulator Netlist File (**.snf**)
SRAM Object File (**.sof**)

area marquee

Граница выделения. Линии, образующие прямоугольник, который окружает выделенную область в Graphic Editor (Графическом редакторе) или в Symbol Editor (Редакторе символов). Он появляется при использовании Selection tool (Указателя выделения), когда перемещается мышь с нажатой Кнопкой 1.

В Hierarchy Display (Дисплее иерархии проекта) прямоугольная граница видна, когда вы перемещаете мышь с нажатой Кнопкой 1.

area selection

Область выделения. Прямоугольная область, включающая один или несколько соседних объектов. В Graphic Editor (Графическом редакторе) или в Symbol Editor (Редакторе символов) эта область содержится внутри прямоугольной границы, называемой area marquee (Граница выделения). В Waveform Editor (Редакторе временных диаграмм), Floorplan Editor (Редакторе физического размещения) и

Hierarchy Display (Дисплее иерархии проекта) все объекты внутри области выделения “подсвечиваются”.

Область выделения используется для выбора нескольких смежных объектов путем перемещения мыши с нажатой Кнопкой 1. В Waveform Editor (Редакторе временных диаграмм) такими объектами могут быть расположенные рядом nodes (цепи) и groups (группы), временные диаграммы целиком или фрагменты одной или нескольких временных диаграмм. В Floorplan Editor (Редакторе физического размещения) такими объектами могут быть расположенные рядом pins (контакты), nodes (цепи), logic cells (логические ячейки) или bins (карманы). В Hierarchy Display (Дисплее иерархии проекта) выделенными могут быть значки файлов.

В Graphic Editor (Графическом редакторе) или в Symbol Editor (Редакторе символов) символы, дуги, окружности, линии с произвольным углом наклона и текстовые блоки для того, чтобы быть выделенными, должны полностью находиться внутри area marquee (Границы выделения). Если ортогональная линия пересекает границу выделения, то выделяется только часть линии внутри её.

Array

Множество, массив. См. group.

ASCII

American Standard Code for Information Interchange (Американский стандартный код для обмена информацией). Текстовые редакторы, которые используются для редактирования текстовых файлов САПР MAX+PLUS II, например, Text Design File (.tdf), Library Mapping File (.lmf) или Vector File (.vec), должны поддерживать эту систему кодирования текстовых данных.

Assignment

Присвоение. В языках AHDL и VHDL обычно используется в Boolean equation (Булевых выражениях). Значение правой части выражения присваивается символу названию или группе в левой части выражения.

assignment (resource)

Назначение ресурсов. См. resource assignment

Assignment&Configuration File (.acf)

Файл назначений и конфигурации ASCII файл с расширением (.acf). в котором хранится информация о probe (Синонимах имён цепей), pin (Контактах), location (Местоположении), chip (Микросхеме), clique (Группах), logic option (Опциях логики), timing (Временных параметрах), connected pin (Соединенных контактах), local routing (Размещении в смежных логических ячейках), device assignments (Назначениях для микросхемы) и о параметрах Compiler (Компилятора), Simulator (Симулятора), Timing Analyzer (Анализатора временных параметров) для данного проекта.

В ACF файле хранится информация, вводимая с помощью команд меню программных модулей САПР MAX+PLUSII. Информация о назначениях контактов и местоположении цепей может вводиться в окне Floorplan Editor (Редактора физического размещения). ACF файл можно редактировать вручную в окне Text Editor (Текстового редактора).

B

back-annotation

Копирование результатов компиляции. Процесс копирования назначений ресурсов для микросхемы из Fit File (Файл размещения проекта) с расширением (**.fit**), созданного Компилятором, в Assignment&Configuration File (Файл назначений и конфигурации) с расширением (**.acf**). Эта команда сохраняет сделанное размещение проекта при последующих компиляциях.

background process

Фоновый процесс. Программный модуль или команда, которые могут выполняться автоматически, в то время пока вы работаете над другой задачей. Они могут генерировать свои сообщения в окне Message Processor (Процессора сообщений). В САПР MAX+PLUS II следующие программные модули и команды являются фоновыми процессами:

Compiler (Компилятор)

Programmer (Программатор)

Simulator (Симулятор)

Timing Analyzer (Анализатор временных параметров)

Модуль ввода ACF файла

Команда **Import Vector File** (Преобразование векторного файла) из меню File программного модуля Waveform Editor (Редактора временных диаграмм)

Команда **Project Archive** (Архивирование проекта) из меню File, Project.

balloon text

Текст в контуре. Текст, появляющийся в Floorplan Editor (Редакторе физического размещения) с информацией об элементе, находящемся под указателем мыши, например, о pin (Контакте), I/O cell (Ячейке ввода/вывода), logical cell (Логической ячейке), embedded cell (Встроенной ячейке) или assignment bin (Выбранном “кармане”). Информация отображается на экране в следующих форматах:

<node name (Название цепи)>@<cell number (Номер ячейки)>

<pin name (Название контакта)>@<pin number (Номер контакта)> (<pin function (функция контакта)> (<dedicated pin name (специальное имя контакта)>)).

При этом *< pin name (Название контакта)>* или *< node name (Название цепи)>* заменяются на текст *<none>*, если для этого элемента нет назначений. Если данному элементу назначены несколько функций, перечисляются первые две функции, затем следует *etc.* При просмотре результатов компиляции в Floorplan Editor (Редакторе физического размещения) текст *unrouted* (неразведённые) появляется после *node name* (Название цепи) или *pin name* (Название контакта), если их не удалось развести.

batch mode

Пакетный режим. Режим моделирования, при котором команды Simulator (Симулятора) выполняются из Command File (Командного файла) с расширением **(.cmd)**, а не выбираются из меню, отображаемом на экране.

Binary

Бинарный, двоичный. Система счисления по основанию 2. Бинарными или двоичными цифрами являются 0 и 1.

Boolean logic

Булева логика. Логика, которая подчиняется теоремам булевой алгебры (Georg Boole, «The Laws of Thought», 1854). Булева часть проекта может быть реализована в микросхеме с помощью матрицы AND-OR.

Branches

Ветви. Ветви иерархического дерева, представляющие различные уровни иерархии. Ветвь состоит из *design filename* (Имени проектного файла), *file icon* (Значка файла) и *ancillary file icons* (Значков вспомогательных файлов). Пересечения ветвей показываются кнопками «+» и «-». Стрелки соединений направлены от ветвей более высокого уровня к ветвям более низкого уровня.

Breakpoint

Точка остановки. Задаваемое пользователем условие, при выполнении которого процесс моделирования будет прекращён.

buried node

Внутренняя цепь. Цепь, состоящая из комбинаторной или регистровой логики, которая не имеет непосредственного соединения с выходным контактом.

buried register

Внутренний регистр. Регистр в микросхеме Altera, выходы которого не соединены с внешним контактом микросхемы. Внутренний регистр может быть реализован на *I/O cells* (Ячейках ввода/вывода) или на *logic cells* (Логических ячейках), которые не имеют внешнего контакта. Внутренний регистр используется для реализации внутренней логики.

Bus

Шина. Толстая линия в файле Graphic Editor (Графического редактора), которая представляет несколько цепей. Шина служит для передачи нескольких сигналов между компонентами проекта и может содержать от 2 до 256 цепей (т. е. битов).

В AHDL файлах и файлах Waveform Editor (Редактора временных диаграмм) понятие group (группа) является синонимом шины.

В VHDL языке шина – это guarded signal (Защищенный сигнал), который может иметь свои drivers (Устройства), например, источники сигналов, выключатели. В VHDL шина называется array (Массивом) и не ограничивается размером 256 symbolic names (Символьных имени). Примером array type (Массива) является STD_LOGIC_VECTOR. Для получения более подробной информации см. Раздел 3.2.1, *Array Types* (Массивы) в *IEEE Standard VHDL Language Reference Manual*. Поддерживаются только одно- и двух- мерные массивы скалярных элементов.

В Verilog HDL языке шина – это array of nets (Массив сеток), и её размер ограничивается 256 symbolic names (Символьными именами). Для получения более подробной информации см. Раздел 3.3: *Vectors* в *IEEE Standard Hardware Description Language Based on the Verilog Hardware Description Language*.

bus (or group) name

Имя шины или группы. Имя шины или группы размером до 256 nodes (цепей).

Имя шины состоит не более чем из 32 символов. Это могут быть числа или арифметические выражения в квадратных скобках, представляющие один или два диапазона. Двух-диапазонные имена не поддерживаются Waveform Editor (Редактором временных диаграмм). Начало и конец диапазона разделяются двумя точками. Каждое число в диапазоне определяет отдельную node (цепь) или бит.

Пример: bus (Шина) a [4 . . 1] состоит из node (Цепей) a4, a3, a2, и a1.

Пример: bus (Шина) b [2 . . 1] [1 . . 0] состоит из цепей b2_1, b2_0, b1_1 и b1_0.

Sequential name (Последовательное имя), состоящее из разделенного запятыми списка имён, может использоваться в AHDL Text Design File (Проектном файле на языке AHDL) с расширением (.tdf) и в Graphic Design Files (Графическом проектном файл) с расширением (.gdf). Только в TDF файле такой список имён должен быть заключен в круглые скобки. Sequential bus names (Последовательные имена шин) могут включать одно-диапазонные или двух-диапазонные имена шин.

Пример: a[3..0], dout[6..4], z3

Первое имя в серии одно-диапазонных, двух-диапазонном или последовательных имён является самым старшим разрядом (MSB) шины, последнее имя является самым младшим разрядом (LSB).

Arbitrary bus name (Условное имя шины), состоящее не более чем из 32 символов, может использоваться в Waveform Design File (Проектный файл, описанный временными диаграммами) с расширением (**.wdf**), Simulator Channel File (Файле временных диаграмм) с расширением (**.scf**) и в Vector file (Векторном файле) с расширением (**.vec**). Arbitrary bus name (Произвольное имя шины) не размер шины.

bus pinstub

Место подключения шины. Место на границе символа мега- или макрофункции, обозначенное значком “х” в Symbol File (Файле символа) с расширением (**.sym**), который определяет входы или выходы этой мега- или макрофункции. Для того чтобы шина (толстая линия), нарисованная в файле Graphic Editor (Графического редактора), “соединилась” с bus pinstub (Местом подключения шины) мега- или макрофункции, она должна иметь такое же число битов.

ByteBlaster

Кабель параллельной загрузки. Устройство и кабель загрузки конфигурации через параллельный порт LPT. Позволяет программировать и конфигурировать микросхемы фирмы Altera непосредственно на плате. С помощью ByteBlaster можно запрограммировать микросхемы семейств MAX3000, MAX 7000S, 7000AE, 7000B, MAX 9000, конфигурационные микросхемы EPC2 и конфигурировать микросхемы семейств ACEX 1K, FLEX 6000, FLEX 8000, FLEX 10K, КА, КВ, КЕ. Возможно программирование и конфигуриция нескольких микросхем в JTAG цепочках и конфигурация нескольких микросхем во FLEX цепочках.

ByteBlaster соединяется с параллельным портом принтера PC посредством разъема DB25-to-DB25. Штырьковый 10-контактный разъем ByteBlaster соединяется с 10-контактным гнездовым разъёмом на плате.

C

Chip

Чип. Группа элементов, определяемых как единый модуль. Чип назначается реальной микросхеме либо пользователем, либо компилятором.

Вы можете назначить чип для функциональных модулей в design files (Проектных файлах). Элементы, назначаемые одному и тому же чипу, во время компиляции размещаются в одной и той же микросхеме. Термин device (Микросхема) всегда относится к реальной микросхеме программируемой логики, тогда как термин chip (Чип) всегда относится к group of logic functions (Группе логических функций).

Когда Compiler (Компилятор) обрабатывает проект, каждому chip (Чипу) соответствует programming file (Программирующий файл) для выбранной микросхемы.

Classic

Классическое. Семейство микросхем Altera EPLD архитектуры, выполненное по EPROM технологии. MAX+PLUSII поддерживает следующие микросхемы семейства Classic: EP600I, EP610, EP610I, EP900I, EP910, EP910I, EP1800I и EP1810.

Clear

Сброс. Входной сигнал, который сбрасывает в ноль регистр или триггер. Синхронный сигнал Clear срабатывает при каждом положительном или отрицательном фронте сигнала Clock (Тактового сигнала). Асинхронный сигнал Clear срабатывает независимо от сигнала Clock.

Clique

Группа. Группа logic functions (Элементов проекта), определяемая как отдельный, именованный модуль. Compiler (Компилятор) старается разместить членов группы вместе. Назначение clique (Группы) позволяет сгруппировать логику критичного к быстродействию пути, чтобы повысить производительность.

Если возможно, все члены группы назначаются одному и тому же LAB. Если не удаётся разместить членов группы в одном LAB (Массив логических блоков), то они помещаются в один row (Ряд LAB для микросхем семейств ACEX 1K, FLEX 10K, FLEX 8000, FLEX 6000 и MAX 9000).

Clock

Тактовый сигнал. Сигнал, который тактирует регистры и триггеры.

В триггере или конечном автомате Clock (Тактовый сигнал) является чувствительный к фронту. Выход триггера может измениться только по фронту сигнала Clock. Например, в D триггере входное значение сохраняется и подается на выход по фронту сигнала Clock.

В некоторых случаях, MAX+PLUS II считает вход Latch Enable как Clock, например, при анализе временных параметров в Delay Matrix (Матрице Задержек).

Clock Enable

Разрешение тактового сигнала. Чувствительный к уровню сигнал у триггеров с разрешением, т.е. с индексом “Е”, а именно DFFE, TFFE, SRFFE и JKFFE. Когда этот сигнал имеет низкий уровень, тактовые импульсы на входе триггера игнорируются.

Column

Столбец. Вертикальная линия LABs (Массивов логических блоков), соединённых с помощью FastTrack Interconnect path (Быстрых соединительных линий) в микросхемах семейств ACEX1K, FLEX 10K, FLEX 8000, FLEX 6000 или MAX 9000.

COM или RS-232 port

Порт RS-232. Коммуникационный последовательный порт у PC или у рабочей станции UNIX. Устройство BitBlaster, которое используется для конфигурирования и программирования микросхем на плате, должно подключаться к COM порту.

combinatorial feedback

Комбинаторная обратная связь. Обратная связь от logic cell (Логической ячейки), которая возвращается обратно в логические ячейки микросхемы независимо от тактового сигнала. В этом случае значение с выхода непосредственно попадает на вход комбинаторной логики и предыдущее входное значение теряется.

combinatorial output

Комбинаторный выход. Выход logic cell (Логической ячейки), который попадает на входы независимо от тактового сигнала. В этом случае теряются предыдущие входные значения.

Command File (.cmd)

Командный файл с расширением .cmd. Текстовый файл ASCII, содержащий команды для моделирования в пакетном режиме.

Comment

Комментарий. В Graphic Editor (Графическом редакторе) и Symbol Editor (Редакторе Символов) комментарий представляет собой блок текста, используемый для документирования проекта. Он не связан ни с каким объектом. Комментарий может быть вставлен в любом месте Graphic Editor files (Файлов графического редактора). Комментарий также может быть вставлен внутри границ символа в Symbol Editor file (Файле Редактора символов). Комментарий игнорируется Compiler (Компилятором) и может использоваться для документирования различных секций файла.

В Waveform Editor (Редакторе временных диаграмм) комментарий представляет собой строку текста, которая используется для создания примечаний в поле временных диаграммах. Комментарий не связывается ни с какой временной диаграммой. Он привязывается к тому моменту на временной шкале, где вводится первая буква. В Name field (Поле имени) появляется метка индикации комментария. Когда вводится комментарий, добавляется пустое пространство между временными диаграммами, куда он вставляется. Комментарии игнорируются Compiler (Компилятором).

Во всех текстовых файлах MAX+PLUS II, за исключением VHDL Design Files (Проектные файлы на языке VHDL) с расширением (**.vhd**), Verilog Design Files (Проектные файлы на языке Verilog) с расширением (**.v**), Assignment&Configuration Files (Файлы назначений и конфигурации) с расширением (**.acf**), например, в Report Files (Файлах отчёта) с расширением (**.rpt**), Vector Files (Векторных файлах) с расширением (**.vec**) и в Text Design Files (Текстовых проектных файлах) с расширением (**.tdf**), комментарий представляет собой строку символов, начинающуюся и заканчивающуюся знаком процент (%). Вы можете вставлять комментарии в любом месте текстового файла.

В VHDL Design Files (Проектных файлах на языке VHDL) и в ACF файлах комментарии начинаются с двойного тире (--) и продолжаются до символа End-of-Line (Конец строки). В VHDL TDF файлах возможно использование комментариев VHDL-стиля. Если используется комментарий VHDL-стиля в TDF файле, то нужно разделить двойное тире (--) и предшествующее символьное имя пробелом.

В Verilog Design Files (Проектных файлах на языке Verilog) комментарии начинаются с двух косых черт (//) и продолжаются до символа End-of-Line (Конец строки). Verilog Design Files (Проектные файлы на языке Verilog) и ACF файлы также поддерживают комментарии, состоящие из любой строки, заключённой между символами /* и */.

Compiler Netlist File

Файл списка соединений Компилятора с расширением (**.cnf**). Двоичный файл, который содержит данные из design file (Проектного файла). CNF файл создается модулем Compiler Netlist Extractor (Экстрактором списка соединений Компилятора), который является частью MAX+PLUS II Compiler (Компилятора).

Configuration EPROM

конфигурационное EPROM. Семейство EPROM микросхем с последовательным интерфейсом фирмы Altera, которое разработано для конфигурирования микросхем семейств FLEX 6000, FLEX 8000, FLEX 10K и ACEX 1K. Оно включает микросхемы EPC1, EPC1213, EPC1064, EPC1064V, и EPC1441.

connection dot

Точка соединения. Точка, вводимая на пересечении двух сигнальных линий (цепей или шин) в Graphic Editor file (Файле Графического редактора). Точка соединения показывает, что сигналы логически соединены.

Construct

Элемент языка. Единица в языках AHDL, VHDL, Verilog HDL или EDIF.

continuity checking

Проверка наличия контактов. Тест на наличие контактов между выводами микросхемы и панельки адаптера программатора. Этот тест подтверждает, что микросхема правильно установлена в панельку адаптера программатора.

cutoff node

Отключённая цепь. Цепь, которая исключается из анализа временных параметров. Цепь может быть исключена из анализа временных параметров с помощью команды **Timing Analysis Cutoff** (Исключение из анализа временных параметров).

D

Database

База данных. Одноуровневое представление всех design files (Проектных файлов) иерархического проекта MAX+PLUS II. База данных используется модулями Compiler (Компилятора) во время компиляции.

Decimal

Десятичное представление. Система счисления по основанию 10. Десятичными цифрами являются цифры от 0 до 9.

В языках AHDL, VHDL и Verilog HDL не требуется специального обозначения для десятичных цифр.

default Simulator Channel File

Файл временных диаграмм по умолчанию. с расширением (.scf) Файл, который может содержать все nodes (Цепи) и groups (Группы) из Simulator Netlist File (Файла списка соединений Симулятора) с расширением (.snf) для данного проекта. Он создается автоматически с помощью команды **Enter Nodes from SNF** (Ввод цепей из SNF файла) из меню Node в Waveform Editor (Редакторе временных диаграмм).

default timing tagging

Определение временных параметров по умолчанию. Timing Analyzer (Анализатор временных параметров) использует следующие определения для анализа временных параметров по умолчанию:

Типы временных параметров	Определения по умолчанию
Delay Matrix (Матрица задержек)	Все входные контакты являются sources (Началом измеряемого пути), все выходные контакты являются destinations (Окончанием измеряемого пути).
Setup/Hold (Матрица временных предустановки и удержания сигналов)	Все входные контакты являются sources (Началом измеряемого пути), все информационные и тактовые входы триггеров; входы Latch Enable защёлок; информационные, адресные и Write Enable входы асинхронной RAM памяти являются destinations (Окончанием измеряемого пути).
Registered Performance (Быстродействие регистровой логики)	Все Q выходы триггеров являются sources (Началом измеряемого пути), все информационные и Clock Enable входы триггеров являются destinations (Окончанием измеряемого пути).

Delimiter

Указатель границ. Текстовая строка, символ или ключевое слово, используемое для определения начала или конца statement (Оператора) или construct (Структуры) в текстовом файле.

Например, [и] (квадратные скобки) является указателем границ AHDL group ranges (Диапазонов групп в языке AHDL), а символ % (проценты) является указателем границ для комментариев во многих текстовых файлах MAX+PLUS II.

design file

Проектный файл. Файл, который содержит логику проекта MAX+PLUS II и обрабатывается Compiler (Компилятором). Следующие файлы являются проектными файлами:

Altera Design File (.adf)
 EDIF Input File (.edf)*
 Graphic Design File (.gdf)*
 OrCAD Schematic File (.sch)*
 State Machine File (.smf)
 Text Design File (.tdf)*
 Verilog Design File (.v)
 VHDL Design File (.vhd)*
 Waveform Design File (.wdf)

Xilinx Netlist Format File (.xnf)

Звездочкой (*) отмечены проектные файлы, которые могут быть top-level files (Файлами верхнего уровня) в иерархических проектах. Остальные проектные файлы должны быть или файлами проекта или файлами нижнего уровня иерархического проекта.

destination node

Окончание измеряемого пути. Node (Цепь), которая определена как окончание пути, на котором измеряется задержка распространения сигнала при timing analysis (Анализе временных параметров). Эта цепь определяется с помощью команды **Timing Analysis Destination** из меню Utilities программного модуля Timing Analyzer (Анализатора временных параметров), и может быть любой цепью, которая является входом logic function (Функционального модуля) или контактом.

Device

Микросхема. Микросхема программируемой логики фирмы Altera семейств Classic, MAX 3000, MAX 7000, MAX 9000, FLEX 6000, FLEX 8000, FLEX 10K и ACEX 1K.

Altera также предлагает микросхемы Configuration EPROM, которые используются для конфигурирования микросхем семейств FLEX 6000, FLEX 8000, FLEX 10K и ACEX 1K.

device assignment

Назначение микросхемы. Назначение микросхемы для определённого пользователем блока logic functions (Функциональных модулей), называемого chip (Чип).

device family

Семейство микросхем. Группа микросхем программируемой логики фирмы Altera, имеющих одинаковую внутреннюю структуру. Classic, MAX 3000, MAX 7000, MAX 9000, FLEX 6000, FLEX 8000, FLEX 10K и ACEX 1K – семейства микросхем, поддерживаемых САПР MAX+PLUS II.

device option

Опция микросхемы. Задаваемый параметр, который определяет работу микросхемы. Микросхемы фирмы Altera имеют следующие опции:

Опция	Семейство микросхем
Auto-Restart Configuration on Frame Error	ACEX 1K, FLEX 6000, FLEX 8000, FLEX 10K
Disable Start-Up Time –Out	FLEX 8000
Enable Chip-Wide Output Enable	ACEX 1K, FLEX 6000 и FLEX 10K
Enable Chip-Wide Reset	ACEX 1K, FLEX 6000 и FLEX 10K
Enable DCLK Output in User Mode	FLEX 8000
Enable INIT_DONE Output	ACEX 1K, FLEX 6000 и FLEX 10K
Enable JTAG Support	MAX 3000A, MAX 7000A, MAX 7000B, MAX 7000S, MAX 7000AE, FLEX 6000 и FLEX 8000
Enable LOCL Output	ACEX 1K, FLEX 10K
JTAG User Code	ACEX 1K, FLEX 10K
Low-Voltage I/O	All (все)
Release Clears Before Tri-States	ACEX 1K, FLEX 6000, FLEX 8000, и FLEX 10K
Security Bit	Classic, MAX 3000A, MAX 7000 и MAX 9000
Turbo Bit	Classic, MAX 3000A, MAX 7000 и MAX 9000 (Logic Cell Turbo Bit может быть применена ко всем логическим ячейкам в микросхемах MAX 7000 или MAX 9000.)
Use the configuration device Pull-Up Resistor	ACEX 1K, FLEX 6000 и FLEX 10K
Use Low-Voltage Configuration EPROM	ACEX 1K, FLEX 6000 и FLEX 10K
User Code	MAX 3000A, MAX 7000A, MAX 7000B, MAX 7000S, MAX 7000AE и MAX 9000
User-Supplied Start-Up Clock	ACEX 1K, FLEX 6000, FLEX 8000 и FLEX 10K

dual I/O feedback

Двойная обратная связь по входу/выходу. Наличие обратной связи от контакта и от регистровой или комбинаторной логики у одной и той же логической ячейки.

dynamic models

Динамические модели. Модели, которые представляют комбинаторную логику проекта в timing Simulator Netlist Files (Временных SNF файлах) с расширением (.snf).

Динамические модели для логики создаются в timing SNF файле, когда включена команда **Optimize Timing SNF** из меню Processing программного модуля Compiler (Компилятор). В этом случае вместо обработки комбинаторной логики, Simulator (Симулятор) и Timing Analyzer (Анализатор временных параметров) обращаются к её динамической модели.

Использование динамических моделей позволяет сократить время моделирования, но компилятору требуется дополнительное время для генерации SNF файла.

E

EAB

См. Embedded Array Block

EC

См. embedded cell

EDIF

Electronic Design Interchange Format (Формат для обмена электронными проектами между различными САПРами). Стандартный формат для передачи проекта другим САПРам.

Вы можете создать netlist EDIF 2 0 0, 3 0 0 file (Файл списка соединений EDIF 2 0 0 или 3 0 0) из схемного проекта или из проекта на языке VHDL или Verilog HDL, которые были обработаны соответствующим synthesis tool (Синтезатором), и затем импортировать этот файл в САПР MAX+PLUS II, как EDIF Input File (Входной файл EDIF) с расширением (.edf). MAX+PLUS II поддерживает входные файлы EDIF, содержащие функции из Library of Parameterized Modules (LPM - Библиотеки параметризованных модулей). Компилятор MAX+PLUS II может также создавать один или несколько EDIF Output File (Выходных файлов EDIF) с расширением (.edo) либо в формате EDIF 2 0 0, либо 3 0 0, содержащих информацию для временного или функционального моделирования с помощью стандартных EDIF симуляторов.

EDIF Netlist Reader (Модуль ввода EDIF файлов) и EDIF Netlist Writer (Модуль генерации EDIF файла), входящие в состав компилятора MAX+PLUS II были

одобрены Electronic Industries Association (EIA – Ассоциацией электронной промышленности), что свидетельствует об их соответствии стандарту EDIF 3 0 0 Netlist View.

EDIF Command File

Командный файл EDIF) с расширением **(.edc)**. Текстовый файл ASCII, используемый для настройки формата EDIF Output Files (Выходных файлов EDIF) с расширением **(.edo)**, создаваемых EDIF Netlist Writer (Генератором выходного EDIF файла).

EDIF Input File

Входной файл в формате EDIF с расширением **(.edf)**. Файл списка соединений EDIF версии 2 0 0 или 3 0 0, созданный любым стандартным генератором EDIF файла. Входные файлы в формате EDIF могут компилироваться с помощью Compiler (Компилятора) MAX+PLUS II. MAX+PLUS II поддерживает Входные файлы в формате EDIF, содержащие функции из Library of Parameterized Modules (LPM - Библиотеки параметризованных модулей).

EDIF Output Files

Выходной файл в формате EDIF с расширением **(.edo)**. Файл списка соединений EDIF версии 2 0 0 или 3 0 0, созданный EDIF Netlist Writer (Генератором выходного EDIF файла). Этот файл может быть использован для моделирования проекта другим программным обеспечением для рабочей станции UNIX или PC.

EEPROM

Electrically Erasable Programmable Read-Only Memory (Электрически стираемое ПЗУ) Вид перепрограммируемой полупроводниковой памяти, содержимое которой может быть стерто поданными соответствующим образом электрическими сигналами.

Embedded Array Block (EAB)

Блок встроенной памяти. Физически сгруппированный набор 8 embedded cells (Встроенных ячеек), с помощью которого можно реализовать память (RAM или ROM) или комбинаторную логику в микросхеме семейства FLEX 10K. EAB состоит из embedded cell array (Массива встроенных ячеек) с входными информационными, адресными, управляющими сигналами и выходными информационными, которые могут поступать на вход триггера.

С помощью одного EAB можно реализовать следующие блоки памяти 256x8, 512x4, 1,024x2 или 2,048x1 битов. Каждая встроенная ячейка внутри EAB обеспечивает до 256 битов памяти. Для блоков памяти, указанных выше размеров, у EAB будет 8, 4, 2 или 1 выход, соответственно. Несколько EAB могут быть объединены для создания большего блока памяти.

Данные в EAB поступают по row interconnect paths (Горизонтальным линиям межсоединений) и по dedicated input bus (Специальной входной шине).

embedded cell (EC)

Встроенная ячейка. Элемент памяти, который находится в embedded array (Встроенном массиве) микросхемы FLEX 10K, и с помощью которого может быть реализована память (RAM или ROM) или комбинаторная логика. Embedded Array Block (EAB - Блок встроенной памяти) состоит из группы 8 embedded cells (Встроенных ячеек). С его помощью можно реализовывать следующие блоки памяти 256x8, 512x4, 1,024x2 или 2,048x1 битов. Каждая встроенная ячейка внутри EAB обеспечивает до 256 битов памяти. Для блоков памяти, указанных выше размеров, у EAB будет 8, 4, 2 или 1 выход, соответственно.

Встроенные ячейки имеют идентификатор формата EC<number (Цифра)>_<row letter (Буква, обозначающая строку)>, где <number (Цифра)> изменяется от 1 до 8, а <row letter (Буква, обозначающая строку)> соответствует букве, обозначающей строку в которой находится EAB.

EPLD

Erasable Programmable Logic Device (Стираемая микросхема программируемой логики), т.е. микросхема фирмы Altera, которая является членом семейств Classic, MAX 3000, MAX 7000 и MAX 9000.

EPROM

Erasable Programmable Read-Only Memory (ПЗУ с возможностью стирания информации. Вид перепрограммируемой полупроводниковой памяти, содержимое которой может быть стёрто посредством обработки микросхемы ультрафиолетовым светом соответствующей длины волны.

evaluated function

Числовая функция. Математическая функция, которая вычисляет арифметическое выражение и возвращает значение на основе одного или нескольких аргументов. AHDL Define Statement (Оператор Определить языка AHDL) может использоваться для описания числовых функций. В следующем примере показывается определение числовой функции MAX:

```
DEFINE MAX(a,b) = (a > b) ? a : b;
```

expander product term

Расширитель логического умножения. Проинвертированное значение с выхода элемента AND, которое возвращается в Logic Array Block (LAB – Массив макро-ячеек) микросхем семейств MAX 3000, MAX 7000 или MAX 9000.

Неподключённый expander product term (Расширитель логического умножения), который может использоваться другими логическими ячейками в том же LAB, называется shareable expander (Расширитель, который может быть использован

другими логическими ячейками). Product term (Выход элемента AND), используемый таким образом, называется shared expander (Общий расширитель).

В микросхемах семейств MAX 3000, MAX 7000 и MAX 9000 логическая функция, которая “заимствуется” из соседней логической ячейки того же LAB, называется parallel expander (Параллельным расширителем).

Extension

Расширение — см. filename extension.

F

Family-specific mega- or macrofunction

Мега- или макрофункция для определённого семейства микросхем. Предоставляемая фирмой Altera мега- и макрофункция, которая содержит логику, оптимизированную для архитектуры определённого семейства микросхем.

Результат работы этих мега или макрофункции, будет одинаковым, независимо от семейства микросхем. Однако, реальные primitives (Базовые элементы) и nodes (Цепи), используемые внутри файла мега- или макрофункции, могут отличаться от семейства к семейству, чтобы используя преимущества различных семейств обеспечить более высокую производительность и эффективную реализацию проекта.

fan-in и fan-out

Входы и Выходы: Fan-in (Входы) – сигналы, поступающие на входы logic cell (Логической ячейки). Fan-out (Выходы) – сигналы, распространяющиеся с выхода logic cell (Логической ячейки).

FastTrack Interconnect

Скоростные линии межсоединений. Выделенные линии межсоединений, которые перекрывают всю длину или ширину кристалла микросхем семейств ACEX 1K, FLEX 6000, FLEX 8000, MAX 9000 или FLEX 10K. Эти линии позволяют сигналам распространяться между всеми Logic Array Blocks (LAB – Массивами макроячеек) микросхемы.

FCF

См. FLEX Chain File.

file icon

Значок файла. Значок, который появляется в окне программного модуля MAX+PLUS II и представляет файл текущего иерархического дерева проекта. Двойной щелчок Кнопки 1 на значке открывает файл, который он представляет.

В окне Hierarchy Display (Дисплея иерархии проекта) значок файла показывает, с помощью какого редактора MAX+PLUS II можно открыть данный файл. Расширение имени файла отображается под значком файла. Имя файла отображается слева от значка.

В окне Compiler (Компилятора) значки показывают входные и выходные файлы для текущего проекта.

Filename

Имя файла. Имя проектного файла, вспомогательного файла или какого-либо другого файла, без расширения.

Имя файла может содержать до 32 символов плюс 3 символа для расширения. Полный путь плюс имя файла и его расширение могут содержать до 128 символов.

Поскольку Windows 3.1 и Windows for Workgroups 3.11 поддерживают только 8-буквенное имя файла, MAX+PLUS II по умолчанию преобразует более длинные имена файлов во всех операционных системах Windows в 8-символьные. Информация о таких преобразованиях находится в файле **maxplus2.idx** каждой директории, содержащей длинные имена файлов. Возможно изменить это условие и задать поддержку длинных имен файлов, доступную в Windows NT и Windows 95, путем установки переменной `USE_WINNT_LONG_FILENAMES` (Использовать длинные имена), находящейся в разделе `[system]` файла **maxplus2.ini**, значения ON (Включено).

В окне Hierarchy Display (Дисплея иерархии проекта) имя файла, значок файла и его расширение представляют файл в текущем иерархическом дереве проекта.

filename extension

Расширение имени файла. Одна, две или три буквы, которые следуют за точкой (.) после имени файла.

В окне Hierarchy Display (Дисплея иерархии проекта) расширение имени файла, имя файла и значок файла представляют файл в текущем иерархическом дереве проекта.

FitFile

Файл размещения с расширением **(.fit)**. Файл ASCII, генерируемый компилятором, который документирует назначения pin (Контактов), logic cell (Логических ячеек), I/O cell (Ячеек ввода/вывода), embedded cell (Встроенных ячеек), chip (Чипа) и device (Микросхемы), сделанные во время последней компиляции. Назначения записываются в синтаксисе Assignment&Configuration File **(.acf)** файла.

Файл размещения может использоваться для сохранения результатов последней компиляции и для функционального тестирования Simulator (Симулятором) и Programmer (Программатором). Для сохранения результатов последней компиляции назначения из Файла размещения должны быть скопированы в ACF файл проекта с помощью команды **Back-Annotate Project** из меню Assign.

С помощью команды **Last Compilation Floorplan** (Физическое размещение последней компиляции) из меню Layout программного модуля Floorplan Editor (Редактора физического размещения) можно отобразить на экране информацию из Файла размещения.

FLEX Chain File

Файл цепочки FLEX с расширением **(.fcf)**. Файл ASCII, который хранит имена файлов программирования для конфигурирования нескольких микросхем семейств ACEX 1K, FLEX 6000, FLEX 8000 или FLEX 10K в режиме Passive Serial (Пассивный последовательный). FCF файл сохраняет информацию, вводимую с помощью команды **Multi-Device FLEX Chain Setup** (Задание FLEX цепочек для нескольких микросхем) из меню FLEX программного модуля Programmer (Программатор).

FLEX 6000

Семейство микросхем фирмы Altera с архитектурой Flexible Logic Element Matrix. Это семейство выполнено по SRAM технологии, имеет высокую производительность и достаточное количество ресурсов для реализации сложных проектов. Семейство FLEX 6000 включает следующие микросхемы EPF6010A, EPF6016/EPF6016A, EPF6024A. Altera не рекомендует использовать это семейство в новых проектах.

FLEX 8000

Семейство микросхем фирмы Altera с архитектурой Flexible Logic Element Matrix. Это семейство выполнено по SRAM технологии, имеет высокую производительность и достаточное количество ресурсов для реализации сложных проектов. Семейство FLEX 8000 включает следующие микросхемы EPF8282V, EPF8282A, EPF8282AV, EPF8452A EPF8636A, EPF8820A, EPF81188A и EPF81500A. Altera не рекомендует использовать это семейство в новых проектах.

FLEX 10K

Семейство микросхем фирмы Altera с архитектурой Flexible Logic Element Matrix. Это семейство выполнено по SRAM технологии, имеет высокую производительность, достаточное количество ресурсов и встроенную память для реализации сложных проектов. Семейство FLEX 10K включает следующие микросхемы EPF10K10/EPF10K10A, EPF10K20, EPF10K30/EPF10K30A, EPF10K30E, EPF10K40, EPF10K50/EPF10K50V, EPF10K50E/EPF10K50S, EPF10K70, EPF10K100/EPF10K100A/EPF10K100B, EPF10K100E, EPF10K130V, EPF10K130E, EPF10K200E/EPF10K200S, EPF10K250A.

FLEX 10KA, FLEX 10KB и FLEX 10KE являются улучшенными версиями микросхем этого семейства. Они совместимы функционально и по внешним контактам с предыдущими моделями микросхем. FLEX 10KA является 3.3В версией а FLEX 10KB и FLEX 10KE 2.5В версией семейства FLEX 10K. Микросхемы FLEX 10KE имеют dual-port RAM (Двух-портовое ОЗУ).

Микросхемы EPF10K100GC503-3DX и EPF10K200EFC672-1X имеют схему phase-locked loop для реализации функций ClockLock и ClockBoost.

flipflop или register

Триггер. Запоминающее устройство (ячейка) для хранения одного бита данных, которое срабатывает по фронту тактирующего сигнала. Положительный фронт тактирующего сигнала изменяет выходное значение триггера, в зависимости от входных данных. Это значение сохраняется до следующего положительного фронта тактирующего сигнала или до тех пор, пока триггер не будет сброшен или установлен сигналами Clear и Preset.

В зависимости от архитектуры микросхемы, триггер может настраиваться для работы, как чувствительная к уровню Latch (Защелка) или как срабатывающий по фронту D, T, JK или SR триггер.

В языке Verilog HDL, понятие register (регистр) используется для описания абстрактного устройства хранения данных, которое Compiler (Компилятор) MAX+PLUS II реализовывает на триггерах микросхемы.

fMAX

Максимальная тактовая частота. Максимальная частота Clock (Тактового сигнала), которая может быть достигнута без нарушения внутренних времён setup (Предустановки) и hold (Удержания). **fMAX** Является также назначаемым параметром, который определяет минимально допустимую тактовую частоту для проекта. В MAX+PLUS II вы можете назначить требуемую частоту **fMAX** для проекта в целом или для любого входного контакта (INPUT или INPUTC), двунаправленного контакта (BIDIR или BIDIRC) или для register (Регистра, триггера).

Function Prototype

Прототип функции. Определяет порты (контакты) primitive (Базового элемента), мегафункции или макрофункции в языке AHDL. Прототип функции содержит имяфункции и список ее входов и выходов. Для мега- и макрофункций прототип может содержать параметры, которые определяют свойства функции. Function Prototypes (Прототипы функций) определяются в Function Prototype Statement (Операторе прототипа функции). Они часто хранятся в Include Files (Включаемых файлах) с расширением (.inc). Включаемые файлы, содержащие Прототипы функций для поставляемых фирмой Altera мега- и макрофункций, находятся в директориях \maxplus2\max2lib\mega_lpm и \maxplus2\max2inc, создаваемых во время инсталляции САПР MAX+PLUS II. (На рабочих станциях UNIX директория **maxplus2** является подкаталогом директории /usr.)

Для использования мега- или макрофункции в проекте, описанном на языке AHDL, её логика должна быть определена в проектном файле и должен быть объявлен её прототип. Объявление Прототипов функций для primitives (Базовых элементов) является необязательным. Затем вы можете использовать эту функцию или в операторе Instance Declaration (Объявление функции) или указав её имя в тексте.

При использовании Module Instantiation (Модульная реализация) в языке Verilog HDL компилятор MAX+PLUS II использует информацию из AHDL Include Files (Включаемого файла AHDL), который содержит Прототипы функций для реализации logic function.

G

GDF

См. Graphic Design File

Glitch или spike

Импульсные помехи или иголки. Пульсирующие значение сигнала, которое появляется при изменении логического уровня два или более раза за короткий промежуток времени.

Когда Simulator (Симулятор) находится в режиме timing (Временного) или linked (Объединённого) моделирования, можно задать ширину импульса и проверить проект на наличие импульсов, которые короче заданного значения. Обнаружение импульсных помех невозможно в режиме функционального моделирования.

global signal

Глобальный сигнал. Сигнал, определяемый внешним контактом или логикой, который распространяется по global routing (Специальному маршруту) на кристалле

прежде чем выполнит определённую функцию. Clock (Тактовый сигнал), Preset (Установка триггера), Clear (Сброс триггера) и Output Enable (Разрешение выходных сигналов) могут быть глобальными сигналами.

Глобальный сигнал, определяемый логикой, доступен только в микросхемах семейств ACEX 1K, FLEX 6000 и FLEX 10K.

Глобальный сигнал может быть установлен несколькими способами:

Во время ввода проекта с помощью primitive (Базового элемента) GLOBAL. Можно использовать dedicated (Специальный) внешний контакт для задания глобального сигнала. Для этого его нужно соединить с primitive (Базовым элементом) GLOBAL. Также возможно использовать выход logic function (Функционального модуля) в качестве глобального сигнала. Для этого его нужно соединить с primitive (Базовым элементом) GLOBAL. Использование глобального сигнала, определяемого логикой, не освобождает для других целей dedicated (Специальный) внешний контакт.

С помощью опции *Automatic Global* в диалоговом окне **Global Project Logic Synthesis** (Параметры логического синтеза для текущего проекта) из меню Assign (Установить). Компилятор выбирает сигнал, распространяющийся с внешнего контактом и используемый большинством триггеров, как глобальный Clock (Тактовый сигнал), Preset (Предустановка) или Clear (Сброс), а сигнал, используемый большинством буферов TRI, как глобальный Output Enable (Разрешение выходных сигналов).

С помощью логической опции *Global Signal* в диалоговом окне **Individual Logic Options**, которое можно открыть из диалогового окна **Logic Options** меню Assign (Установить). Когда эта опция включена для внешнего контакта или для одно-выходного logic function (Функционального модуля), то это эквивалентно использованию primitive (Базового элемента) GLOBAL. Выключение этой логической опции предохраняет внешний контакт от использования в качестве глобального сигнала.

GND

Входное напряжение низкого уровня (Ground - земля).

По умолчанию GND является неактивным значением сигнала. В AHDL Text Design File (Текстовом проектном файле на языке AHDL) с расширением (.tdf), GND является предопределенной константой и ключевым словом. В VHDL Design File (Проектном файле на языке VHDL) с расширением (.vhd) GND представляется '0'. В Verilog Design File (Проектном файле на языке Verilog) с расширением (.v) GND представляется 0. В файле графического редактора GND является символом. GND представляется как низкий (0) логический уровень в Simulator (Симуляторе) и Waveform Editor (Редакторе временных диаграмм).

Graphic Design File

Графический проектный файл с расширением (**.gdf**). Схемный проектный файл, созданный с помощью Graphic Editor (Графического редактора) MAX+PLUS II.

OrCAD Schematic File (Схемный файл OrCAD) с расширением (**.sch**) автоматически преобразуется в GDF файл и используется Graphic Editor (Графическим редактором) и Compiler (Компилятором) MAX+PLUS II как GDF файл.

Gray code

Код Грея. Схема подсчета, в которой у последовательных значений счёта изменяется только один бит. В противоположность ему, двоичная последовательность счёта не исключает изменения нескольких бит у последовательных значений счёта. При изменении только одного бита восприимчивость шумов в схеме уменьшается.

group или array

Группа или массив. В AHDL языке группой является объединение до 256 символьных имён, которые рассматриваются как единое целое. Имя группы может быть определено с помощью однодиапазонного, двухдиапазонного или последовательного формата имени группы.

В VHDL языке группой называется массив, который не ограничивается 256 символьными именами. Примером array type (Массива) является STD_LOGIC_VECTOR и BIT_VECTOR. Для получения более подробной информации см. Раздел 3.2.1, *Array Types* (Массивы) в **IEEE Standard VHDL Language Reference Manual**. Поддерживаются только одно- и двух- мерные массивы скалярных элементов.

В языке Verilog HDL группой называется массив, который ограничивается 256 символьными именами. Примером array types (Типов массивов) являются запоминающие устройства (которые являются массивами register elements или слов) и массивы gate instances и registers. Elements, instances или registers в массиве задаются в определённом порядке. За дополнительной информацией обратитесь к разделу 3.3: *Vectors*, разделу 3.8: *Memories* и разделу 7: *Gate and Switch Level Modeling* в руководстве **IEEE Standard Hardware Description Language Based on the Verilog Hardware Description Language**.

В Waveform Editor (Редакторе временных диаграмм) и Simulator (Симуляторе) группой является объединение до 256 цепей, которое представляется одной диаграммой. В этих программных модулях для имени группы может быть использован arbitrary (Условный) или single-range (Однодиапазонный) формат имени группы.

group name

Имя группы (шины) — см. bus name (Имя шины).

Н

hard logic function

Аппаратный функциональный модуль. Функциональный модуль в проектном файле, который не может быть удалён во время логического синтеза и может быть назначен физическим ресурсам, например, device (Конкретной микросхеме), pin (Контакту), logic cell (Логической ячейке) или I/O cell (Ячейке ввода/вывода). В Graphic Design Files (Графических проектных файлах) с расширением **(.gdf)** и в Text Design Files (Текстовых проектных файлах) с расширением **(.tdf)**, аппаратными базовыми элементами/портами являются INPUT, INPUTC, OUTPUT, OUTPUTC, BIDIR, BIDIRC, LCELL, MCELL, DFF, DFFE, TFF, TFFE, JKFF, JKFFE, SRFF, SRFFE и LATCH. Однако, базовые элементы INPUT и INPUTC, не влияющие на выходные значения проекта, не считаются аппаратными функциональными модулями. Если primitives (Базовые элементы) SOFT, TRI и OPNDRN не удаляются во время логического синтеза, они также являются аппаратными базовыми элементами. Мегафункции или макрофункции, содержащие аппаратные базовые элементы, рассматриваются как аппаратные функциональные модули.

В Waveform Design Files (Проектных файлах, описанных временными диаграммами) с расширением **(.wdf)**, аппаратными функциональными модулями являются входные и выходные node (Цепи), внутренние цепи registered (Регистрового) и combinatorial (Комбинаторного) типа.

Hexadecimal

Шестнадцатеричная система счисления. Система счисления по основанию 16.

Шестнадцатеричными цифрами являются цифры от 0 до 9 и символы от A до F.

Шестнадцатеричные числа обозначаются следующим образом:

Язык	Обозначение
AHDL	X"<последовательность цифр от 0 до 9, от A до F>" Или H"<последовательность цифр от 0 до 9, от A до F>"
VHDL	16#<последовательность цифр от 0 до 9, от A до F>#
Verilog HDL	'h<последовательность цифр от 0 до 9, от A до F>

Примеры:

H"123AECF" (AHDL)

16#FF# (VHDL)

'h837FF (Verilog HDL)

Hexadecimal (Intel-Format) File

Интел HEX файл с расширением **(.hex)**. Текстовый файл ASCII в Интел HEX формате.

Compiler (Компилятор) и Simulator (Симулятор) MAX+PLUS II могут использовать Hex файлы для задания исходного значения памяти (например ROM).

Компилятор MAX+PLUS II автоматически создает выходные Hex файлы, содержащие данные для конфигурирования микросхем семейства FLEX 8000 в режиме Active Parallel Up (APU) и для конфигурирования микросхем семейств ACEX 1K, FLEX 6000 и FLEX 10K в режиме Passive Serial (PS).

После компиляции вы также можете создать Hex файлы, поддерживающие другие режимы конфигурирования микросхем семейств ACEX 1K, FLEX 6000, FLEX 8000 и FLEX 10K.

Если в вашем проекте есть память, и вы используете Hex файлы для задания её исходного содержимого, то следует назвать этот файл именем, отличным от имени проекта и любого chip name (Имени чипа) внутри проекта. Поскольку Компилятор автоматически генерирует выходные Hex файлы для микросхем семейств ACEX 1K, FLEX 6000, FLEX 8000 и FLEX 10K. Они могут переписать содержимое файлов для инициализации памяти.

hierarchical node or symbol name

Иерархическое имя цепи или символа. Уникальное имя цепи или символа, которое основано на его местоположении в иерархии проектных файлов и net ID (Идентификационном) номере. В языках AHDL, VHDL – это имя logic function (Функционального модуля), в Verilog HDL – это instance name.

Каждая цепь и символ в проекте имеют иерархическое имя. Вы также можете назначить node name (Имя цепи) или probe name (Синоним имени) для цепи.

Hierarchy Interconnect File

Файл иерархических соединений с расширением **(.hif)** Файл ASCII, созданный модулем Компилятора Netlist Extractor (Экстрактором списка соединений). Этот файл определяет иерархические соединения проектных файлов.

History File

Файл истории с расширением **(.hst)**. Файл ASCII, созданный Simulator (Симулятором) MAX+PLUS II, в который записываются все команды, кнопки и опции, используемые во время моделирования, а также сообщения во время их выполнения.

hold time

Время удержания сигнала. У flipflop-триггера (D-nhbuuthf) – это минимальный промежуток времени, в течение которого информационный входной сигнал или Clock Enable (Разрешение тактовых сигналов) должен остаться неизменным после активного фронта Clock (Тактового сигнала) данного триггера.

У latch (Защелки) – это минимальный промежуток времени, в течение которого сигнал на D входе должен остаться неизменным после активного фронта сигнала Latch Enable (Разрешение защёлки).

У блока асинхронного RAM (ОЗУ) – это минимальный промежуток времени, в течение которого сигналы на информационных и адресных входах должны оставаться неизменными после активного фронта сигнала Write Enable (Разрешение записи).

Внутренние генерируемые сигналы должны соответствовать временам удержания сигналов триггера, защёлки и асинхронного RAM (ОЗУ). У пользователя нет возможности задавать эти времена.



I/O cell или I/O element

Ячейка ввода/вывода. Register (Триггер), который находится на периферии микросхем семейств ACEX 1K, FLEX 10K, FLEX 8000 или MAX 9000. Ячейка ввода/вывода имеет малое setup time (Время предустановки сигнала) для обеспечения максимальной скорости передачи данных.

В версиях MAX+PLUS II до 5.0 ячейки ввода/вывода были известны, как периферийные регистры.

I/O feedback

Обратная связь контакта ввода/вывода. Обратная связь от внешнего выходного контакта микросхемы фирмы Altera. Она позволяет использовать выходной контакт в качестве входного.

I/O type

Тип ввода/вывода. Направление распространения сигнала для node (Цепи), pin (Контакта) или state machine (Конечного автомата).

В Graphic Editor (Графическом редакторе) или Symbol Editor (Редакторе символов) pins (Контакты) и pinstubs (Контакты символов) могут иметь следующие типы: входной, выходной или двунаправленный.

В AHDL языке тип порта ввода/вывода может быть input (Входным), output (Выходным), buried (Внутренний), machine input (Входом автомата) или machine output (Выходом автомата).

В Waveform Editor (Редакторе временных диаграмм) тип ввода/вывода node (Цепи) может быть input (Входным), output (Выходным), buried (Внутренний). Входной и выходной типы могут соответствовать внешним контактам, внутренний всегда представляет логику, которая не соединяется напрямую с внешним контактом.

ICR

См. in-circuit reconfigurability.

in-circuit reconfigurability (ICR)

Возможность изменения конфигурации на плате (в работающей схеме). Способность микросхем, выполненных по технологии SRAM, например, ACEX 1K, FLEX 6000, FLEX 8000 и FLEX 10K фирмы Altera, загружать конфигурационные данные при включении питания системы или во время обычной работы системы, после того как они распаяны на печатной плате.

Переконфигурация на плате может быть осуществлена неограниченное число раз. Данные могут загружаться из локальной PROM, например, Altera Configuration EPROM (Конфигурационные микросхемы фирмы Altera), или загружаться под управлением внешнего контроллера или программного модуля Programmer (Программатор) MAX+PLUS II. Programmer (Программатор) позволяет конфигурировать одну или несколько микросхем ACEX 1K и FLEX 10K в цепочке JTAG и одну или несколько микросхем ACEX 1K, FLEX 6000, FLEX 8000 или FLEX 10K в цепочке FLEX. Микросхемы ACEX 1K, FLEX 6000 и FLEX 10K можно конфигурировать в одной FLEX цепочки. Микросхемы FLEX 8000 нужно конфигурировать отдельно от других FLEX микросхем.

in-system programmability (ISP)

Возможность программирования в системе (на плате). Способность микросхем, выполненных по EEPROM технологии, например, MAX 3000A, MAX 7000S, MAX 7000AE, MAX 7000A, MAX 7000B и MAX 9000 быть запрограммированными после того, как они распаяны на печатной плате.

Programmer (Программатор) MAX+PLUS II поддерживает программирование на плате через последовательный кабель загрузки BitBlaster и параллельный кабель загрузки ByteBlaster. Также он позволяет программировать несколько микросхем в цепочке JTAG.

Include File

Включаемый файл с расширением **(.inc)**. Текстовый файл ASCII, который может быть вставлен в Text Design File (Текстовый проектный файл) с расширением **(.tdf)** посредством оператора AHDL Include Statement. Включаемые файлы могут содержать Function Prototype (Прототипы функций), Define (Определения), Parameters (Параметры) или Constant Statements (Константы). Включаемые файлы, содержащие Function Prototypes (Прототипы функций) предоставляемых фирмой Altera мега- и макрофункций находятся соответственно в директориях **\maxplus2\max2lib\mega_lpm** и **\maxplus2\max2inc**, создаваемых во время инсталляции. (На рабочих станциях UNIX директория **maxplus2** является подкаталогом директории **/usr**.)

При использовании Module Instantiation в языке Verilog HDL, компилятор использует информацию из AHDL Include Files (Включаемых файлов AHDL), содержащих прототипы функций для реализации функциональных модулей.

insertion point

Точка вставки. Место, куда вставляется текст или графический объект.

В диалоговом окне или в окне Text Editor (Текстового редактора) точка вставки выглядит как мигающая вертикальная черточка. В Graphic Editor (Графическом редакторе) или Symbol Editor (Редакторе символов) она выглядит как мигающий квадратик.

Когда вы набираете текст, он появляется слева от точки вставки, которая перемещается, по мере того как вы печатаете. Когда вводятся или вставляются символы, верхний левый угол элемента появляется в точке вставки.

Instance

Реализация функционального модуля. Использование logic function (Функционального модуля) в design file (Проектном файле).

В Graphic Editor (Графическом редакторе) функциональный модуль представляется (net) ID number (Идентификационным номером), который располагается в нижнем левом углу символа. В Waveform Editor (Редакторе временных диаграмм) - это имя node (Цепи). В AHDL языке функциональные модули объявляются или в Instance Declaration (Объявление модуля), где объявляются переменные типа <primitive> (Базовый элемент), <megafunction> (Мегафункция), или <macrofunction> (Макрофункция), или непосредственно в тексте описания. В языке VHDL функциональные модули объявляются в помощью Component Instantiation Statement. Registers (регистры) также могут быть реализованы с помощью оператора Register Inference. В Verilog HDL функциональные модули объявляются с помощью операторов Module Instantiations и Gate Instantiations.

В Hierarchy Display (Дисплее иерархии проекта) мега- или макрофункции представляются именем функции и net ID number (Идентификационным номером),

разделённые двоеточием. В AHDL Variable Declaration и VHDL Component Instantiation Statement функциональный модуль представляется именем и названием модуля, разделённые двоеточием. В операторах Module или Gate Instantiation языка Verilog HDL функциональный модуль представляется именем и следующим за ним названием модуля.

interactive mode

Интерактивный режим. Режим моделирования, в котором экранные опции, кнопки и выполняемые команды меню выбираются с помощью клавиатуры или мыши.

ISP

См. in-system programmability.

J

Jam File

Jam файл с расширением **(.jam)**. Файл ASCII на языке Jam для программирования и тестирования микросхем. Он позволяет запрограммировать, верифицировать и проверить отсутствие записанной информации для одной или нескольких микросхем в JTAG цепочке. Jam файлы используются для программирования микросхем с помощью встраиваемых процессоров. Микросхемы MAX 3000A, MAX 7000A, MAX 7000S, MAX 7000AE и MAX 9000 фирмы Altera могут быть запрограммированы с помощью Jam файлов. Цепочка JTAG может содержать любую другую микросхему, которая соответствует спецификации IEEE 1149.1 JTAG, включая ACEX 1K, FLEX 10K, FLEX 6000 и некоторые микросхемы FLEX 8000.

JAM файлы можно создавать с помощью команды **Create Jam or SVF File** (Создать Jam или SVF файл) из меню File программного модуля Programmer (Программатор) или Compiler (Компилятор).

JCF

см. JTAG Chain File.

JEDEC File

Стандартный файл для программирования с расширением **(.jed)** Файл ASCII, который содержит информацию для программирования микросхем. JEDEC файлы являются стандартным промышленным форматом для передачи информации между системой подготовки данных и программатором микросхем. Компилятор MAX+PLUS II во время компиляции проекта автоматически создает JEDEC файлы для микросхем Classic и EPM5032.

Programmer (Программатор) MAX+PLUS II может использовать JEDEC файлы, созданные с помощью MAX+PLUS II, MAX+PLUS (DOS), A+PLUS или PLDshell Plus для программирования микросхем фирмы Altera, перечисленных выше. Programmer (Программатор) также может сохранять данные и функциональные тестовые вектора в формате JEDEC файла.

JTAG boundary-scan testing

Тестирование с использованием JTAG интерфейса или периферийное сканирование. Тестирование, при котором внутренняя часть микросхемы отключается от её внешних контактов ввода/вывода. Такое тестирование возможно благодаря Joint Test Action Group (JTAG) Boundary-Scan Test (BTS) архитектуре, которая используется во всех MAX 3000, MAX 7000A, MAX 7000B, MAX 7000S, кроме EPM7032S и EPM7064S, MAX 7000AE, MAX 9000, FLEX 8000, кроме EPF8452A и EPF81188A, ACEX 1K, FLEX 10K микросхемах. Последовательные данные посылаются в boundary-scan cells (Ячейки периферийного сканирования) микросхемы, наблюдаемые данные выдвигаются и сравниваются с ожидаемыми результатами. Периферийное сканирование позволяет эффективно тестировать печатные платы.

Полная или частичная реализация JTAG BST в микросхемах ACEX 1K, MAX 3000A, MAX 7000A, MAX 7000B, MAX 7000S, MAX 7000AE, MAX 9000, FLEX 10K позволяет программировать и конфигурировать их в JTAG цепочках.

JTAG chain

JTAG цепочка. См. multi-device JTAG chain.

JTAG Chain File

Файл JTAG цепочки) с расширением (**.jcf**). Файл ASCII, в котором хранится информация о номинале микросхемы, о порядке микросхем в JTAG цепочки, имена программирующих файлов, для программирования или конфигурирования одной или нескольких микросхем в JTAG цепочке. JCF файл сохраняет информацию, вводимую с помощью команды программных модулей Compiler (Компилятора) или Programmer (Программатора) **Create Jam or SVF File** из меню File или команды **Multi-Device JTAG Chain Setup** из меню JTAG.

K

Keyword

(Ключевое слово) Слова, которые зарезервированы для реализации синтаксических конструкций в входных файлах MAX+PLUS II, включая AHDL Text Design Files (Текстовые файлы проектов) с расширением (**.tdf**), Assignment&Configuration Files (Файлы назначений и конфигурации) с расширением (**.acf**), Command Files (Командные файлы) с расширением (**.cmd**), EDIF Command

Files (Командные файлы EDIF) с расширением (**.edc**), Library Mapping Files (Файлы карты библиотек) с расширением (**.lmf**), VHDL Design Files (Проектные файлы на языке VHDL) с расширением (**.vhd**), Verilog Design Files (Проектные файлы на языке Verilog) с расширением (**.v**) и Vector Files (Векторные файлы) с расширением (**.vec**). Например, ключевое слово OF нельзя использовать в качестве символического имени, незаключённого в кавычки, в файле AHDL.

L

LAB

См. Logic Array Block.

latch

(Защёлка) Чувствительное к уровню тактируемое запоминающее устройство, которое хранит один бит данных. Отрицательный фронт сигнала Latch Enable (Разрешение защёлки) фиксирует содержимое защёлки в соответствии с входным значением до следующего положительного фронта сигнала Latch Enable (Разрешение защёлки).

Latch Enable

(Разрешение защёлки) Чувствительный к уровню сигнал, который управляет защёлкой. Когда у него высокий уровень, входной сигнал передаётся на выход, когда - низкий, на выходе сохраняется последнее значение.

LC

см. logic cell (Логическая ячейка).

least significant bit (LSB)

(Младший значимый бит) Младший разряд двоичного числа в шине или группе. Например, LSB для шины или группы с названием $a[31..0]$ это $a[0]$ (или a_0).

Library Mapping File

Файл карты библиотек с расширением (**.lmf**). Текстовый файл ASCII, используемый для создания карты ячеек EDIF Input Files (Входных EDIF файлов) с расширением (**.edf**) или символов OrCAD Schematic Files (Схемного OrCAD файла) с расширением (**.sch**) соответствующих primitives (Базовым элементам), мегафункциям или макрофункциям MAX+PLUS II.

Library of Parameterized Modules (LPM)

Библиотека параметризованных модулей. Технологически независимая библиотека функциональных модулей. С помощью параметров они оптимально настраиваются для конкретной задачи. Компилятор MAX+PLUS II поддерживает

LPM модули, используемые в схемных, AHDL, VHDL, Verilog HDL и EDIF входных файлах.

LMF

См. Library Mapping File.

Load

Загрузка. Входной сигнал, который загружает данные в регистр. При синхронном Load сигнале данные загружаются в регистр по каждому положительному или отрицательному фронту тактового сигнала. При асинхронном сигнале данные загружаются независимо от тактового сигнала.

local routing

Размещение в смежных логических ячейках. Эта опция доступна для микросхем FLEX 6000. Она назначает выходные цепи узла в том же самом или в соседнем LAB. Использование этой опции гарантирует, что данные цепи будут соединены с помощью shared local interconnect (Общей локальной линии межсоединений), которая имеется у микросхем FLEX 6000. Её использование позволяет сократить задержку распространения сигнала и достигнуть максимальной производительности проекта.

Location

Местоположение. Общий термин, который относится к назначаемым физическим ресурсам микросхемы фирмы Altera.

Вы можете определить местоположение logic function (Функционального модуля) назначив для него один из следующих ресурсов микросхемы:

- конкретную logic cell (Логическую ячейку);
- конкретную I/O cell (Ячейку ввода/вывода);
- конкретную embedded cell (Встроенную ячейку);
- LAB (Массив логических ячеек), EAB (Блок встроенных ячеек), row (Строка LAB) или column (Столбец LAB).

При назначении logic function (Функциональному модулю) общего местоположения, такого как LAB (Массив логических ячеек), EAB (Блок встроенных ячеек), row (Строка LAB) или column (Столбец LAB), компилятор выбирает наиболее подходящие logic cell (Логические ячейки) или embedded cell (Встроенные ячейки) внутри LAB, EAB, row и column для реализации данной логики.

Log File

Файл регистрации с расширением (.log). Текстовый файл ASCII, создаваемый Simulator (Симулятором) MAX+PLUS II. В этот файл записываются все команды, кнопки и экранные опции, используемые во время сеанса интерактивного моделирования.

logic function или Design Entity

Функциональный модуль Primitive (Базовый элемент). Мегафункция, макрофункция или конечный автомат, которые могут быть представлены либо именем либо символом в design file (Проектном файле).

Logic Array Block (LAB)

Массив логических ячеек. Физически сгруппированный набор логических ресурсов в микросхеме фирмы Altera. LAB состоит из массива логических ячеек или макроячеек. Сигнал, доступный какой-либо логической ячейки в LAB, доступен всему массиву логических ячеек.

В микросхемах семейства Classic вся логика в LAB имеет общий Clock (Тактовый сигнал) и сигналы поступают или с global bus (Общей шины) или с dedicated input bus (Специальных входных контактов). В микросхемах MAX 3000, MAX 7000 сигналы в LAB поступают из Programmable Interconnect Array (PIA – Программируемой матрицы межсоединений) и с dedicated input bus (Специальных входных контактов). В микросхемах ACEX 1K, FLEX 6000, FLEX 8000, MAX 9000 и FLEX 10K сигналы в LAB поступают с row FastTrack Interconnect (Горизонтальных шин межсоединений).

logic cell (LC)

Логическая ячейка. Общий термин для основных блоков микросхем фирмы Altera, на которых реализуется проект. В микросхемах Classic, MAX 3000, MAX 7000 и MAX 9000 логическая ячейка называется macrocell (Макроячейкой) и состоит из двух частей комбинаторной логики и конфигурируемого register (Триггера). Комбинаторная логика используется для реализации логических функций. В микросхемах ACEX 1K, FLEX 6000, FLEX 8000 и FLEX 10K логическая ячейка называется logic element (Логическим элементом) и состоит из look-up table (LUT – четырёх входового ОЗУ), используемого для реализации логических функций четырёх переменных и конфигурируемого register (Триггера).

Register (Триггер) может быть сконфигурирован как latch (Защёлка), D, T, JK, SR триггер или зашунтирован для реализации только комбинаторной логики. Сигнал с триггера может поступать на входы других логических ячеек, на вход той же самой логической ячейки, реализуя обратную связь, либо на внешний выходной или двунаправленный контакт микросхемы.

Вы можете назначить logic function (Функциональный модуль) конкретной логической ячейке. Вы также можете его назначить LAB (Массиву логических ячеек), row (Строке LAB) или column (Столбцу LAB), чтобы данный модуль был размещён в соответствующем LAB, строке или столбце.

В микросхемах ACEX 1K, FLEX 10K, FLEX 8000, FLEX 6000 и MAX 9000 номер логической ячейки имеет следующий формат LC<number (Номер)>_<LAB name (Имя LAB)>, где <number (Номер)> - это цифра от 1 до 8, а <LAB name (Имя LAB)> состоит из буквы, обозначающей строку, и цифры, обозначающей номер столбца LAB. В микросхемах Classic, MAX 3000 и MAX 7000 логические ячейки

имеют формат номера LC<number (Номер)>, где < number (Номер)> состоит из цифр и букв.

Микросхемы ACEX 1K, FLEX 10K, FLEX 8000 и MAX 9000 имеют специальные логические ячейки, называемые I/O cells (Ячейки ввода/вывода), на периферии микросхемы.

logic cell Turbo Bit

См. Turbo Bit.

logic element

См. logic cell

logic level

Логический уровень. Входной и выходной логический уровень nodes (Цепей) и groups (Групп) определяется в соответствии со следующими обозначениями:

Обозначение	Логический уровень
0	Низкий логический уровень (GND)
1	Высокий логический уровень (VCC)
X	Неопределенный (недопускается при инициализации)
Z	Высокое сопротивление (нет входа от контакта), например, используется “выходной” логикой двунаправленного контакта, когда его “входная” логика работает на ввод сигнала.
От 0 до 9, от A до F	Используется для groups (Групп) и интерпретируются как двоичные, десятичные, шестнадцатеричные или восьмеричные значения в зависимости от заданной системы исчисления. Первый бит является старшим значимым, последний – младшим.

logic option

Логическая опция или параметр) Опция, которая управляет процессом логического синтеза для одного или нескольких logic function (Функциональных модулей).

Доступно множество логических опций. Логические опции могут назначаться для отдельных logic functions (Функциональных модулей). Кроме этого группа назначений логических опций, называемая logic synthesis style (Стилем логического синтеза), так же может быть применена к отдельным функциональным модулям. По умолчанию стиль логического синтеза применяется ко всему проекту. Логическая опция logic cell Turbo Bit может включаться и выключаться для всей микросхемы.

Логические опции также могут быть назначены в качестве параметров для мегафункции или макрофункции.

1. Некоторые логические опции недоступны при стандартном синтезе. Все логические опции доступны при многоуровневом синтезе.
2. Логическая опция игнорируется, если она неприменима к выбранному семейству микросхем.

logic synthesis style

Стиль логического синтеза. Комбинация установок опций логического синтеза, которые сохраняются под одним именем.

Стиль логического синтеза может быть индивидуальным для различных семейств микросхем, так как установки опций изменяются в зависимости от архитектуры выбранного семейства микросхем.

Если стиль логического синтеза для текущего проекта не определен, т.е., если он не назначен командой **Global Project Logic Synthesis** (Параметры логического синтеза для текущего проекта) из меню **Assign**, то компилятор MAX+PLUS II будет использовать “по умолчанию” параметры стиля **Normal**. Для их просмотра откройте диалоговое окно **Define Synthesis Style** (Определение стиля синтаксиса), выберите стиль в окошке *Style* и нажмите кнопку **Use Default** (Использовать по умолчанию).

logical operator

Логический оператор. Оператор, выполняющий логические операции с nodes (Цепями), groups (Группами) или числами.

Логическими операторами языка AHDL являются NOT (!), AND (&), NAND (!&), OR (#), NOR (!#), XOR (\$) и XNOR (!\$).

Логическими операторами языка VHDL являются AND, NAND, OR, NOR, XOR и NOT.

Логическими операторами языка Verilog HDL являются and (&&) и or (||) .

LPM

См. Library of Parameterized Modules.

LSB

См. least significant bit.

M

Macrocell

См. logic cell.

Macrofunction

Макрофункция. Крупный блок, который может использоваться совместно с вентилями, триггерами и мегафункциями в design files (Проектных файлах) MAX+PLUS II.

Altera рекомендует в новых проектах отдавать предпочтение использованию мегафункций вместо эквивалентных макрофункций. Мегафункции лучше масштабируются и синтезируются.

Altera предлагает более 300 макрофункций 74XXXX серии в директории `\maxplus2\max2lib` и ее подкаталогах, создаваемых во время инсталляции. AHDL Include Files (Включаемые файлы) с расширением (**.inc**) для этих макрофункций находятся в директории `\maxplus2\max2inc`. VHDL Component Declaration (Объявления компонент VHDL) для макрофункций, поддерживаемых VHDL, находятся в `maxplus2 package` в библиотеке **altera**, которая размещается в подкаталоге `\maxplus2\vhdl\nn`, где *nn* или “87” или “93”. На рабочих станциях UNIX каталог **maxplus2** является подкаталогом директории `/usr`.

Для просмотра файла, содержащего логику макрофункции, выделите её символ в Graphic Editor (Графическом редакторе) или её имя в Text Editor (Текстовом редакторе) и выберите команду **Hierarchy Down** (Просмотр иерархии вниз) из меню File.

MAX 3000

Семейство микросхем второго поколения фирмы Altera, основанное на архитектуре Multiple Array MatriX. Эта архитектура используется в семействах MAX 3000A, MAX 7000, MAX 7000A, MAX 7000B, MAX 7000S, and MAX 7000AE. Семейство MAX 3000 выполнено по EEPROM технологии и включает микросхемы EPM3032A, EPM3064A, EPM3128A и EPM3256A.

MAX 3000A являются улучшенной версией MAX 7000 и предназначены для проектов, требующих низкой стоимости конечного устройства. Микросхемы MAX 3000A позволяют использовать до шести управляемых логикой или внешним контактом сигналов Output Enable (Разрешение выхода) и несколько Clock (Тактовых сигналов) с возможностью инверсии. MAX 3000A имеют in-system programming (Возможность программирования на плате) благодаря наличию JTAG BST схемы.

Altera рекомендует использовать MAX 3000A, MAX 7000A, MAX 7000B, MAX 7000E, MAX 7000S и MAX 7000AE микросхемы вместо MAX 7000.

MAX 5000

Семейство микросхем первого поколения фирмы Altera, основанное на архитектуре Multiple Array MatriX. Семейство выполнено по EPROM технологии и включает микросхемы EPM5032, EPM5064, EPM5128, EPM5128A, EPM5130 и EPM5192.

MAX 7000, MAX 7000A, MAX 7000B, MAX 7000E, MAX 7000S и MAX 7000AE

Семейства микросхем второго поколения фирмы Altera, основанные на архитектуре Multiple Array MatriX. Они выполнены по EEPROM технологии и включает микросхемы EPM7032, EPM7032V, EPM7032S, EPM7032AE, EPM7064, EPM7064S, EPM7064AE, EPM7096, EPM7128A, EPM7128E, EPM7128S, EPM7160E, EPM7160S, EPM7192E, EPM7192S, EPM7256A, EPM7256E, EPM7256S и EPM7512AE.

Микросхемы MAX 7000A, MAX 7000B, MAX 7000E, MAX 7000S и MAX 7000AE являются улучшенными версиями микросхем MAX 7000 и совместимы с ними функционально и по контактам. Микросхемы MAX 7000A, MAX 7000B, MAX 7000E, MAX 7000S и MAX 7000AE отличаются от MAX 7000 тем, что они позволяют использовать до шести управляемых логикой или внешним контактом сигналов Output Enable (Разрешение выхода), быстрый ввод и несколько Clock (Тактовых сигналов) с возможностью инверсии. Микросхемы MAX 7000A, MAX 7000B, MAX 7000S и MAX 7000AE имеют in-system programming (Возможность программирования на плате) благодаря наличию JTAG BST схемы. MAX 7000, MAX 7000E и MAX 7000S имеют напряжение питания ядра 5В, MAX 7000A и MAX 7000AE – 3,3В, а MAX 7000B – 2,5В.

Altera рекомендует использовать микросхемы MAX 7000A, MAX 7000B, MAX 7000E, MAX 7000S и MAX 7000AE вместо микросхем MAX 7000.

MAX 9000

Семейство микросхем третьего поколения фирмы Altera на основе архитектуры Multiple Array MatriX. Оно выполнено по EEPROM технологии и включает микросхемы EPM9560, EPM9560A, EPM9480, EPM9400, EPM9320 и EPM9320A.

Микросхемы MAX 9000A представляют собой улучшенную версию микросхем MAX 9000 и совместимы с ними функционально и по контактам. Микросхемы MAX 9000A отличаются от микросхем MAX 9000 тем, что они имеют дополнительные 16 битов для user code (Кода пользователя).

Микросхемы MAX 9000 с суффиксом “F” имеют фиксированный алгоритм программирования, и, следовательно, могут программироваться Serial Vector Format Files (Файлами в формате последовательных векторов).

MAX+PLUS (DOS)

Multiple Array MatriX Programmable Logic User System Устаревшая версия САПР фирмы Altera для DOS ОС. MAX+PLUS САПР фирмы Altera для работы с устаревшими семействами микросхем Classic и MAX 5000. Graphic Design Files (Графические проектные файлы) с расширением (.gdf), созданные для MAX+PLUS, автоматически преобразуются и обрабатываются компилятором MAX+PLUS II. AHDL Text Design Files (Текстовые проектные файлы на языке

AHDL) с расширением **(.tdf)** компилируются непосредственно. Programmer (Программатор) MAX+PLUS II может программировать микросхемы Classic и MAX 5000 файлами JEDEC Files с расширением **(.jed)** и Programmer Object Files с расширением **(.pof)**, созданными MAX+PLUS II.

САПР MAX+PLUS больше не предлагается фирмой Altera.

MAX+PLUS II Message File

Файл сообщений MAX+PLUS II с расширением **(.mmf)**. Двоичный файл, созданный MAX+PLUS II, который содержит сообщения, сгенерированные программным модулем или командой MAX+PLUS II, которые работают в фоновом режиме, например, Compiler (Компилятор) или Programmer (Программатор). Этот файл используется для вывода на экран сообщений Message Processor (Процессором сообщений) и для определения местоположения источника сообщения в design files (Проектных файлах) и в ancillary files (Вспомогательных файлах).

maxplus2.idx file

Текстовый файл, который создаётся автоматически при сохранении файлов, и показывает соответствие между именами файлов, состоящими из более чем восьми символов, и сгенерированным 8-символьными именами файлов.

MAX+PLUS II создает файл **maxplus2.idx** в каждой директории, где вы сохраняете файлы с именами, состоящими из более чем 8 символов. Этот файл автоматически обновляется каждый раз, когда вы сохраняете файл с длинным именем.

maxplus2.ini file

Текстовый файл, создаваемый во время инсталляции, который содержит параметры, влияющие на работу программных модулей MAX+PLUS II. В этот файл записываются все опции, которые вы устанавливаете во время работы, чтобы в дальнейшем они использовались автоматически.

MegaCore и OpenCore megafunctions

Мегафункции MegaCore и OpenCore) Мегафункции MegaCore и OpenCore являются верифицированными проектными файлами для сложных системных функций, которые могут быть приобретены у фирмы Altera. Они протестированы и оптимизированы под архитектуры микросхем ACEX 1K, FLEX 10K, FLEX 8000, FLEX 6000, MAX 9000, MAX 7000 и MAX 3000. MegaCore фирмы Altera состоит из нескольких различных design files (Проектных файлов). Post-synthesis AHDL design file (Проектный файл на языке AHDL, полученный после логического синтеза) используется для fitting (размещения) мегафункции в выбранной микросхеме фирмы Altera. Так же имеются VHDL или Verilog HDL functional simulation models (Модели для функционального моделирования на языке VHDL или Verilog HDL), которые используются при проектировании и моделировании с помощью стандартных средств моделирования EDA.

Мегафункции OpenCore представляют собой MegaCore, которые вы можете использовать в своём проекте для оценки эффективности их применения до их покупки. Если же вы покупаете мегафункцию, то можно будет генерировать programming files (Программирующие файлы) и выходные файлы EDIF, VHDL и Verilog HDL для post-compilation simulation (Моделирования после компиляции) с помощью других EDA средств.

MegaCore/OpenCore мегафункции находятся на сайте фирмы Altera <http://www.altera.com>.

Если ваш лицензионный файл для MegaCore функций включает разрешение просмотра исходного проектного файла, вы можете его просмотреть путем выделения символа мегафункции в Graphic Editor (Графическом редакторе) или её имени в Text Editor (Текстовом редакторе) и выбора команды **Hierarchy Down** (Просмотр иерархии вниз) из меню File.

Megafunction

Мегафункция. Крупный блок, который может использоваться совместно с вентилями, триггерами и макрофункциями 74XXXX серии в design files (Проектных файлах) MAX+PLUS II.

Altera предлагает библиотеку мегафункций, включая функции из Library of Parameterized Modules (LPM - Библиотеки параметризованных модулей) версии 2.1.0, в директории `\maxplus2\max2lib\mega_lpm`, создаваемой во время инсталляции. AHDL Include Files (Включаемые файлы AHDL) с расширением (**.inc**) для этих мегафункций находятся в директории `\maxplus2\max2lib\mega_lpm`. VHDL Component Declaration (Объявления компонент VHDL) для LPM и других мегафункций находятся в `lpm_components package` в библиотеке **lpm** и в `megacore package` в библиотеке **altera**, соответственно. Обе библиотеки размещаются в директории `\maxplus2\vhdlnn`, где **nn** - это "87" или "93". (На рабочих станциях UNIX директория **maxplus2** является подкаталогом директории **/usr**.) Для просмотра файла, содержащего логику мегафункции, выделите её символ в Graphic Editor (Графическом редакторе) или её имя в Text Editor (Текстовом редакторе) и выберите команду **Hierarchy Down** (Просмотр иерархии вниз) из меню File.

memory bit и memory word

Бит памяти и слово памяти. Бит памяти – это адресуемая ячейка в блоке памяти (RAM или ROM).

Слово памяти – это группа битов в блоке RAM или ROM.

Например, слово памяти `content5_[4..0]` состоит из отдельных битов памяти `content5_4`, `content5_3`, `content5_2`, `content5_1` и `content5_0`.

Memory Initialization File

Файл инициализации памяти с расширением **(.mif)**. Файл, который определяет начальное содержание блока памяти (RAM или ROM), т.е. значение для каждого адреса. Данный файл используется во время компиляции и моделирования проекта.

Memory initialization Output File

Выходной файл инициализации памяти с расширением **(.mio)**. Файл ASCII, генерируемый компилятором при создании Text Design Export File **(.tdo)** для проекта. TDO файл, в котором имеется RAM или ROM, всегда содержит MIO файл для каждого сегмента памяти.

MIO файл определяет адреса и значения, используемые для инициализации RAM или ROM памяти, аналогично Memory Initialization File (Файлу инициализации памяти) с расширением **(.mif)**.

Вы можете переименовать файл MIO в MIF и использовать его с TDO файлом, который сохраняется как Text Design File (Текстовый проектный файл) с расширением **(.tdf)**.

Memory segment или segment

Сегмент памяти. Физическая реализация памяти RAM или ROM в микросхеме. Сегмент памяти содержит последовательность адресуемых битов.

В микросхемах FLEX 10K сегмент памяти состоит из набора битов, которые размещаются в одной embedded cell (Встроенной ячейке). Каждая встроенная ячейка позволяет реализовать до 256 битов памяти. Для создания блока памяти могут потребоваться несколько сегментов памяти.

Message Text File

Текстовый файл сообщений. с расширением **(.mtf)**. ASCII файл, который содержит текст сообщений, отображаемых в окне Message Processor (Процессора сообщений).

MIF

См. Memory Initialization File.

MMF

См. MAX+PLUS II Message File.

most significant bit (MSB)

Старший разряд. Старший разряд двоичного числа в шине или в группе. Например, MSB для шины с названием $a[31..0]$ будет $a[31]$.

MSB

См. most significant bit.

MTF

См. Message Text File.

multi-device FLEX chain

FLEX-цепочка нескольких микросхем. Последовательность микросхем, в которой конфигурационные данные передаются от микросхемы к микросхеме при Passive Serial (Пассивной последовательной схеме загрузки).

Programmer (Программатор) MAX+PLUS II может конфигурировать несколько микросхем ACEX 1K, FLEX 6000, FLEX 8000 или FLEX 10K во FLEX цепочке.

multi-device JTAG chain

JTAG цепочка нескольких микросхем. Последовательность микросхем, в которой данные для программирования или конфигурации передаются от микросхемы к микросхеме используя схему JTAG BST.

Programmer (Программатор) MAX+PLUS II может программировать или конфигурировать несколько микросхем ACEX 1K, MAX 3000A, MAX 7000S, MAX 7000A, MAX 7000B, MAX 7000AE, MAX 9000, FLEX 10K в JTAG цепочке. JTAG цепочка может содержать любую комбинацию микросхем фирмы Altera и микросхем других производителей, соответствующих спецификации IEEE 1149.1 JTAG, включая некоторые микросхемы FLEX 8000.

MAX +PLUS II может создавать Jam Files (**.jam**), Jam Byte-Code Files (**.jbc**) и Serial Vector Format Files (**.svf**), которые позволяют программировать одну или несколько микросхем MAX 3000A, MAX 7000S, MAX 7000A, MAX 7000B, MAX 7000AE и MAX 9000 в JTAG цепочке. Файлы SVF могут использоваться Automatic Test Equipment (Программаторами АТЕ типа). Jam и JBC файлы используются при программировании под управлением встроенного процессора.

multi-level synthesis (Многоуровневый синтез) Логический синтез, который использует преимущества всех доступных логических опций, включая опции, перечисленные в диалоговых окнах **Define Synthesis Style** и **Advanced Options** из меню Assign. Такой тип логического синтеза позволяет fit (Размещать) сложные проекты без вмешательства пользователя.

Многоуровневый синтез может быть выбран с помощью диалогового окна **Global Project Logic Synthesis** из меню Assign. Такой тип синтеза доступен для семейств микросхем ACEX 1K, MAX 5000, MAX 7000, MAX 9000, FLEX 6000, FLEX 8000 и FLEX 10K. Для семейств ACEX 1K, FLEX 6000, FLEX 8000 и FLEX 10K возможен только такой тип синтеза.

N

name characters

Символы в именах. В MAX+PLUS II символы от A до Z, от a до z, от 0 до 9, ко-
сая черта (/), тире (-) и подчеркивание (_) являются разрешенными для задания
имён breakpoint (Точки остановки), chip (Чипа), clique (Группы), file (Файла),
group (Шины), node (Цепи), parameter (Параметра), pin (Контакта), pinstub (Места
подключения), probe (Синонима имени), logic synthesis style (Стиля логического
синтеза), quoted symbolic name (Символьного имени в кавычках) и unquoted sym-
bolic name (Символьного имени без кавычек). Исключения перечислены ниже.
Использование регистра клавиатуры имеет значение только для Verilog HDL
файлов.

Элемент	Исключения для символов имён
Filename (Имя файла)	Не разрешается использование косой черты (/). Для рабочих станций UNIX имеет значение регистр клавиатуры.
Single-range group (bus) name (Простое имя шины)	Не разрешается использование косой черты (/), идентификатор шины не может оканчиваться циф- рой. За именем следует числовой диапазон или арифметическое выражение в скобках. Начало и конец числового диапазона отделяются двумя точ- ками. Например, шина a[3..1] состоит из цепей a3, a2 и a1. В файлах Графического редактора последовательные имена шин могут содержать не- сколько простых имён шин. Например, a[8..0], d[6..4].
Dual-range group (bus) name (Двух-диапазонное имя ши- ны)	Такие же, как и для простых имён шин. Имеется два числовых диапазона или арифметических вы- ражения в скобках. Например, a[6..3] [4..0].
Sequential group (bus) name (Последовательное имя ши- ны)	Состоит из последовательности имён цепей, разде- лённых запятыми и заключенных в круглые скобки. Например, шина (a, b, c) состоит из цепей a, b, и c. В файлах Графического редактора круглые скобки не используются.
Unquoted symbolic name (Символьное имя без кавы- чек в языке AHDL)	Не разрешается использование тире (-). Название не может состоять только из цифр. Нельзя исполь- зовать ключевые слова AHDL.
Verilog HDL identifiers (Идентификаторы Verilog HDL)	Не разрешается использование косой черты (/) и тире (-). Имя не должно начинаться с цифры. Имеет значение регистр клавиатуры. Нельзя использовать

Элемент	Исключения для символов имён
	ключевые слова Verilog HDL.
VHDL names (Имена VHDL)	Не разрешается использование косой черты (/) и тире (-). Имя должно начинаться с буквы, не должно оканчиваться подчеркиванием (_) и не может содержать два подчеркивания (__) подряд. Нельзя использовать ключевые слова VHDL.
ACF names (ACF имена)	Имена, которые содержат символы: косую черту (/), тире (-), вертикальную черту (), двоеточие (:) и точку (.), должны быть заключены в двойные кавычки ("").

net ID number

См. symbol ID number.

Network

Сеть. Группа соединенных node (Цепей) или bus lines (Линий шины), включая цепи и шины, соединенные посредством имён.

Node

Цепь (Узел). Проводник, по которому распространяется сигнал логических компонентов проектного файла. В Verilog HDL, цепи называются “nets”.

В файлах Graphic Editor (Графического редактора) цепи представляются в виде линий, в текстовых файлах цепи - символьные имена, в файлах Waveform Editor (Редактора временных диаграмм) цепи - временные диаграммы.

Node Database File (Файл базы данных цепей) с расширением (.ndb) Файл, содержащий базу данных имён цепей проекта. В нём можно редактировать назначения ресурсов и probe (Синонимов имён) с помощью команд меню Assign и Floorplan Editor (Редактора физического размещения). Модули компилятора Compiler Netlist Extractor (Экстрактор списка соединений Компилятора) и Database Builder (Построитель базы данных) создают Node Database File (Файл базы данных цепей проекта) во время обработки проекта.

Если перед компиляцией включена команда программного модуля Compiler (Компилятор) **Preserve All Node Name Synonyms** из меню Processing, то данный файл не будет содержать всех возможных форм названий цепей проекта.

Если нечаянно удалить этот файл, то нужно повторно компилировать проект прежде, чем станет возможным использование большинства команд меню Assign и функций Floorplan Editor (Редактора физического размещения). Если Node Database File (Файл базы данных цепей) создан с помощью команды **Project Save&Check** из меню File, то во Floorplan Editor (Редакторе физического размещения) будут видны только pins (Контакты). Необходима полная компиляция проекта для того, чтобы buried nodes (Внутренние цепи) стали видны во Floorplan Editor (Редакторе физического размещения).

node или net name

Имя цепи или имя сети. Имя, данное сигналу в design file (Проектном файле). Оно может содержать до 32 символов. Можно использовать следующие символы: буквы от A до Z, от a до z; цифры от 0 до 9, косую черту (/), тире (-) и подчеркивание (_). Иерархические имена цепей могут содержать до 128 символов, включая вертикальную черту (|), двоеточие (:) и точку (.). Регистр клавиатуры не важен.

Существуют некоторые ограничения для имён VHDL Design Files (Проектных файлов на языке VHDL) с расширением (**.vhd**), Verilog Design Files (Проектных файлов на языке Verilog) с расширением (**.v**) и для символьных имён без кавычек в AHDL Text Design Files (Текстовых проектных файлах на языке AHDL) с расширением (**.tdf**).

node type

Тип цепи. Тип логики, которая формирует сигналы для node (Цепи) или group (Шины) в Waveform Design File (Проектном файле, описанным временными диаграммами) с расширением (**.wdf**) или в Vector File (Векторном файле) с расширением (**.vec**). Определяются четыре типа логики:

Тип	Описание
INPUT	Цепь или шина, на которую сигнал поступает с входного контакта.
COMB	Цепь или шина, на которую сигнал поступает с комбинаторной логики, например, вентиля AND.
REG	Цепь или шина, на которую сигнал поступает с триггера логической ячейки микросхемы.
MACH	Цепь, на которую сигнал поступает с конечного автомата.

Normal logic synthesis style

Обычный стиль логического синтеза. Параметры логического синтезатора, при которых он пытается реализовать проект используя минимальное количество ресурсов микросхемы без внесения существенных дополнительных временных задержек.

Для отображения на экране параметров данного стиля, выберите его в диалоговом окне **Define Synthesis Style**. Доступ к нему можно получить через диалоговые окна **Logic Options** или **Global Project Logic Synthesis** из меню Assign.

Если стиль логического синтеза для текущего проекта не определен, т.е., если он не назначен командой **Global Project Logic Synthesis** (Параметры логического синтеза для текущего проекта) из меню Assign, то компилятор MAX+PLUS II будет использовать “по умолчанию” параметры стиля Normal. Для их просмотра откройте диалоговое окно **Define Synthesis Style** (Определение стиля синтаксиса), выберите стиль в окошке *Style* и нажмите кнопку **Use Default** (Использовать по умолчанию).



object-by-object selection

Выделение несовпадающих объектов. Процесс выделения нескольких несовпадающих объектов. Первый объект выделяется щелчком Кнопки 1 на нём. Возможно добавление или удаление объектов к выделенной группе, если удерживать нажатой клавишу Shift во время щелчка Кнопкой 1 на объекте.

В Graphic Editor (Графическом редакторе) такое выделение может быть использовано для графических и текстовых блоков, в Waveform Editor (Редакторе временных диаграмм) для nodes (Цепей) и groups (Шин), во Floorplan Editor (Редакторе физического размещения) для pins (Контактов), nodes (Цепей), logic cells (Логических ячеек) или bins (Карманов), в Hierarchy Display (Дисплее иерархии проекта) для значков файлов.

В Graphic Editor (Графическом редакторе) несколько объектов, выделенных с помощью прямоугольной области, могут быть добавлены к уже существующему выделению, если удерживалась нажатой клавиша Shift до время перемещения мыши с нажатой Кнопки 1.

Octal

Восьмеричная система счисления. Система счисления по основанию 8. Восьмеричные цифры – это цифры от 0 до 7.

Восьмеричные числа обозначаются следующим образом:

Язык	Обозначение
AHDL	O ”< последовательность цифр от 0 до 7>” Или Q ”< последовательность цифр от 0 до 7>”
VHDL	8 #< последовательность цифр от 0 до 7>#
Verilog HDL	’o < последовательность цифр от 0 до 7>

Пример:

Q”4671223” (AHDL)

8#4671223# (VHDL)

’o4671223 (Verilog HDL)

one-hot encoding

Единичное кодирование. Способ двоичного кодирования, при котором только один бит имеет значение 1. Четыре разрешенных значения 0001, 0010, 0100, и 1000 вместе представляют собой пример единичного кода так, как в каждом из этих четырех значений только один бит установлен в 1.

Вы можете реализовать единичное кодирование вручную. Команда **Global Project Logic Synthesis** из меню Assign имеет опцию *One-Hot State Machine Encoding*, которая выполняет единичное кодирование для всего проекта. Altera рекомендует использовать эту опцию вместо единичного кодирования вручную.

Данная опция доступна как при многоуровневом, так и при стандартном синтезе. Она игнорируется, если неприменима к выбранному семейству микросхем.

OrCAD Library File

Файл библиотеки OrCAD с расширением **(.lib)**. Двоичный файл, содержащий информацию, которая описывает каким образом символы отображаются в OrCAD Schematic File (Схемном файле OrCAD) с расширением **(.sch)**.

Graphic Editor (Графический редактор) MAX+PLUS II использует OrCAD Library File, сгенерированный САПР OrCAD, чтобы импортировать OrCAD Schematic File. OrCAD Library File для каждого OrCAD Schematic File должен содержать все библиотеки для OrCAD символов, используемых в схеме. Этот OrCAD Library File должен быть скопирован в ту же директорию, где находится OrCAD Schematic File.

OrCAD Schematic File (Схемный файл OrCAD) с расширением **(.sch)** Схемный проектный файл, созданный с помощью графического редактора OrCAD Draft. Его можно открыть и редактировать в MAX+PLUS II, а затем сохранить, как

Graphic Design File (Графический проектный файл) с расширением **(.gdf)** или как OrCAD Schematic File (Схемный файл OrCAD) с расширением **(.sch)**. OrCAD Schematic File может обрабатываться Compiler (Компилятором) MAX+PLUS II.

Oscillation

Нестабильный сигнал, быстрые переколебания сигнала. Нестабильный логический уровень сигнала. Если Simulator (Симулятор) в режиме временного моделирования или моделирования нескольких проектов, то возможно задать временной интервал стабилизации сигнала и проверить проект на наличие сигналов, которые не стабилизируются в течение заданного периода. Если Simulator (Симулятор) находится в режиме функционального моделирования, то возможно проверить проект на наличие nil-period oscillation (Изменение логического уровня сигнала в течение очень короткого промежутка времени).

Output Enable

Разрешение выхода. Высокий логический уровень этого сигнала разрешает выдачу информации на выходные цепи или контакты функционального модуля.

В микросхемах семейства MAX 7000 (кроме MAX 7000E) сигнал с контакта global Output Enable (Общие разрешение выхода) с активным низким уровнем должен инвертироваться при подключении ко входу Output Enable с активным высоким логическим уровнем primitive (Базового элемента) TRI. Для остальных семейств микросхем активным может быть либо низкий, либо высокий уровень сигнала.

Для микросхем MAX 9000, Fitter (Разводчик) автоматически вставляет дополнительные primitives (Базовые элементы) LCELL, чтобы обеспечить правильную полярность для контакта non-global Output Enable (Разрешение выхода, не являющегося общим) или для сигнала Output Enable, формируемого logic cell (Логической ячейкой).

P

Parameter

Параметр. Атрибут мегафункции или макрофункции, который определяет логику, создаваемую или используемую для реализации функционального модуля, т.е. характеристика, определяющая размер, поведение и его реализацию в микросхеме. Параметры определяют primitives (Базовые элементы) или подпроекты, необходимые для реализации логики данного функционального модуля.

Параметризованная function (Функция в языке описания или функциональный модуль в схеме) – это функция, поведение которой определяется одним или несколькими параметрами. Некоторые логические функции, такие как, например, функции или модули Library of Parameterized Modules (LPM - Библиотеки параметризованных модулей) являются изначально параметризованными и требуют задания параметров при использовании.

Параметры могут быть заданы при использовании любой мегафункции в MAX+PLUS II, чтобы определить её размеры и реализацию. Некоторые параметры также могут применяться к макрофункциям old-style (74XXXX серии) для определения их реализации. MAX+PLUS II позволяет назначать global (Общие) значения параметров по умолчанию для всего проекта.

parameterized module

Параметризованный модуль. Logic function (Функциональный модуль), который использует параметры для масштабируемости, настройки и эффективной реализации. MAX+PLUS II поддерживает различные параметризованные модули, также называемые параметризованными функциями, включая функции Library of Parameterized Modules (LPM - Библиотеки параметризованных модулей).

Функции LPM обеспечивают архитектурно-независимый ввод проекта для всех микросхем, поддерживаемых САПР MAX+PLUS II. Compiler (Компилятор) MAX+PLUS II обрабатывает функции LPM, используемые в графических, AHDL, VHDL, Verilog HDL и EDIF входных файлах.

Pin

Контакт, вывод. Реальный входной или двунаправленный контакт микросхемы фирмы Altera.

В файлах Graphic Editor (Графического редактора) контакт представляется символами INPUT, INPUTC, OUTPUT, OUTPUTC, BIDIR или BIDIRC. В Text Design File (Текстовом проектном файле) с расширением (.tdf) контакт представляется как port (Порт) INPUT, OUTPUT или BIDIR. В VHDL Design file (Проектном файле на языке VHDL) с расширением (.vhd) контакт представляется как port (Порт) IN, OUT или INOUT. В Verilog Design File (Проектном файле на языке Verilog) с расширением (.v) контакт представляется как port (Порт) input, output и inout. В Waveform Design File (Проектном файле, описанным временными диаграммами) с расширением (.wdf) контакт представляется как входная, выходная или двунаправленная цепь.

Можно назначить символ контакта или порт проектного файла внешнему контакту микросхемы с конкретным номером. Такое назначение можно сделать и для row (Строки линий межсоединений) и для column (Колонки линий межсоединений), чтобы внешние контакты находились в заданном месте.

pin number

Номер контакта. Номер внешнего контакта микросхемы, используемый для назначения ему входного или выходного сигнала проектного файла.

Для обозначения контактов BGA и PGA корпусов используются буквы и цифры.

Pinstub

Место подключения контакта. В Graphic Editor (Графическом редакторе) и Symbol Editor (Редакторе символов) это место на границе символа, обозначенное значком "x" в Symbol File (Символьном файле) с расширением (.sym) и имя, ко-

торое соответствует входу или выходу primitive (Базового элемента), мегафункции или макрофункции в проектном файле, описывающим данный символ. Линия (цепь), нарисованная в схеме, должна соединиться с местом подключения, чтобы компилятор распознал соединение между логикой текущего файла и логикой primitive (Базового элемента), мегафункции или макрофункции.

При редактировании символа можно добавлять или удалять pinstubs (Места подключения).

Pinstubs (Места подключения) в файлах Graphic Editor (Графического редактора) синонимичны портам в AHDL Function Prototypes (Прототипах функций AHDL) и в VHDL Component Declarations (Объявлениях компонентов VHDL). Они также синонимичны портам, перечисленным в Subdesign Sections текстовых проектных файлов с расширением **(.tdf)** более низкого уровня, в Entity Declarations VHDL проектных файлов с расширением **(.vhd)** более низкого уровня, в Module Declarations, Module Instantiations и Gate Instantiations проектных файлов на языке Verilog с расширением **(.v)**.

pinstub name

Имя контакта. Символьное имя, которое определяет вход или выход logic function (Функционального модуля).

В Symbol Editor (Редакторе символов) “видимое” имя контакта появляется снаружи и внутри символа. Оно может быть аббревиатурой или псевдонимом “полного” имени входного, выходного или двунаправленного контакта проектного файла мегафункции, макрофункции или Function Prototype (Прототипа функции) базового элемента.

При создании pinstub (Места подключения) в Symbol Editor (Редакторе символов) вы можете определить отображать или нет “видимое” pinstub name (Имя контакта) в файле Graphic Editor (Графического редактора). Наличие или отсутствие каждого pinstub (Места подключения) может быть задано при редактировании символа в Graphic Editor (Графическом редакторе) с помощью команды **Edit Ports/Parameters** из меню Symbol.

Pinstubs (Места подключения) в файлах Graphic Editor (Графического редактора) синонимичны портам в AHDL Function Prototypes (Прототипах функций AHDL) и в VHDL Component Declarations (Объявлениях компонентов VHDL). Они также синонимичны портам, перечисленным в Subdesign Sections текстовых проектных файлов с расширением **(.tdf)** более низкого уровня, в Entity Declarations VHDL проектных файлов с расширением **(.vhd)** более низкого уровня, в Module Declarations, Module Instantiations и Gate Instantiations проектных файлов на языке Verilog с расширением **(.v)**.

PLF

См. Programmer Log File.

PLS-ES

Устаревшая версия САПР MAX+PLUS II для PC.

POF

См. Programmer Object File.

Port

Порт. Символьное имя, представляющее вход или выход primitive (Базового элемента) или design file (Проектного файла).

В языке AHDL имя порта в Subdesign Section представляет вход или выход для текущего файла. Это же имя порта появляется в Function Prototype (Прототипе функции) для данной функции. Когда используется primitive (Базовый элемент) или проектный файл более низкого уровня в Instance Declaration или непосредственно в тексте его порты используются для соединения с другими функциями TDF файла. После объявления функционального модуля его входы и выходы обозначаются в формате *<instance name (Имя модуля)>.<port name (Имя порта)>* в Logic Section. При использовании функционального модуля непосредственно в тексте для соединения его портов с другими модулями TDF файла указываются или имена портов или порядок их следования.

В VHDL имя порта в Entity Declaration представляет вход или выход текущего файла. При использовании primitive (Базового элемента) или проектного файла более низкого уровня в Component Instantiation его порты соединяются с сигналами посредством Port Map Aspects.

В Verilog HDL порт в Module Declaration представляет вход или выход текущего файла. При использовании проектного файла более низкого уровня в Module Instantiation его порты соединяются по порядку или по имени с портами Module Declaration реализуемого модуля. Аналогично, при реализации базового элемента с помощью Module Instantiation, его порты используются для соединения по порядку с другими функциями в файле. Вентильные примитивы в Verilog HDL имеют порты, называемые “терминалами”. Когда используется вентильный примитив в Gate Instantiation, его терминалы соединяются по порядку с терминалами реализуемого вентиля.

Имя порта в AHDL Subdesign Section, VHDL Entity Declaration, в Verilog HDL Module или в Gate Declaration синонимично имени контакта в Graphic Design File (Графическом проектном файле) с расширением **(.gdf)** или в Waveform Design File (Проектном файле, описанном временными диаграммами) с расширением **(.wdf)**. Имя порта, которое появляется при реализации функционального модуля, синонимично pinstub name (Имени контакта) символа в файле Graphic Editor (Графического редактора).

Preset

Предустановка (установка в единицу). Асинхронный входной сигнал, который устанавливает высокий логический уровень на выходе register (Триггера) независимо от входных сигналов.

Primitive

Базовый библиотечный элемент. Один из базовых функциональных блоков, используемых для реализации проектов с помощью САПР MAX+PLUS II. Базовые элементы используются в Graphic Design Files (Графических проектных файлах) с расширением (**.gdf**), AHDL Text Design Files (Текстовых проектных файлах на языке AHDL) с расширением (**.tdf**), VHDL Design Files (Проектных файлах на языке VHDL) с расширением (**.vhd**) и в Verilog Design Files (Проектных файлах на языке Verilog) с расширением (**.v**).

Базовые элементы для графического редактора включают buffers (Буферные устройства), flipflops (Триггеры), latch (Защёлки), input (Входы), output (Выходы) и логические функции. Символы базовых элементов находятся в директории **\maxplus2\max2lib\prim**, создаваемой во время инсталляции.

Primitives (Базовые элементы) buffers (Буферных устройств), flipflops (Триггеров), latch (Защёлок) для языков AHDL, VHDL и Verilog HDL являются подмножеством их символов, используемых в файлах Graphic Editor (Графического редактора). Остальные функции реализуются логическими операторами, портами и другими конструкциями этих языков. Function Prototypes (Прототипы функций) для базовых элементов AHDL встроены в САПР MAX+PLUS II. Component Declarations для базовых элементов VHDL находится в **maxplus2 package** директории **\maxplus2\max2vhdl nn \altera**, где **nn** это “87” или “93”.

На рабочих станциях UNIX директория **maxplus2** является подкаталогом директории **/usr**.

primitive array

Массив базовых элементов. Один базовый элемент, подключённый к шине или к нескольким цепям. Такое подключение означает использование нескольких базовых элементов. Их количество будет соответствовать числу подключённых цепей.

Probe

Синоним имени. Уникальное имя, назначаемое цепи, например, входу или выходу базового элемента, мегафункции или макрофункции, которое может быть использовано вместо полного иерархического имени цепи в MAX+PLUS II. Таким образом, синоним имени обеспечивает короткое имя для идентификации цепи.

product term

Терм-произведение. Два или более сомножителя в булевом выражении дизъюнктивной нормальной формы, соединенном оператором AND составляют терм, где “product” (произведение) означает “logical product” (логическое произведение).

Programmer Log File

Файл протокола работы Программатора с расширением **(.plf)**. Файл ASCII, создаваемый программным модулем Programmer (Программатор), в который записываются команды и сообщения во время программирования.

Programmer Object File

Объектный программирующий файл) с расширением **(.pof)** Двоичный файл, создаваемый модулем компилятора Assembler. Он содержит данные, используемые программным модулем Programmer (Программатор) для программирования микросхем Altera. Programmer (Программатор) MAX+PLUS II может сохранять функциональные тестовые вектора в POF файле.

programming file

Программирующий файл) Файл, содержащий данные для программирования микросхем фирмы Altera. Программные модули MAX+PLUS II Compiler (Компилятор) и Programmer (Программатор) могут создавать программирующие файлы. MAX+PLUS II создаёт программирующие файлы следующих форматов:

- FLEX Chain File **(.fcf)**
- Hexadecimal (Intel-Format) File **(.hex)**
- Jam Byte-Code File **(.jbc)**
- Jam File **(.jam)**
- JEDEC File **(.jed)**
- JTAG Chain File **(.jcf)**
- Programmer Object File **(.pof)**
- Raw Binary File **(.rbf)**
- Serial Bitstream File **(.sbf)**
- Serial Vector Format File **(.svf)**
- SRAM Object File **(.sof)**
- Tabular Text File **(.tff)**

Файлы POFs, SOFs, JEDEC Files, Jam Files, JBC Files, JCFs и FCFs используются для программирования или конфигурирования микросхем с помощью программного модуля Programmer (Программатор) MAX+PLUS II. Тестовые вектора для функционального тестирования могут сохраняться в POF или JEDEC File файлах. Все остальные форматы используются для программирования и конфигурирования микросхем другими средствами.

Файлы JEDEC File созданные A+PLUS и PLDshell Plus могут быть использованы для программирования микросхем Classic. Programmer (Программатор)

MAX+PLUS II может сохранять данные, считанные с микросхемы, в формате POF, JEDEC File или Jam File файлов.

Project

Проект. Проект состоит из всех файлов связанных с отдельной разработкой, включая все файлы subdesign (Подпроектов) и ancillary (Вспомогательные файлы), созданные пользователем или программным обеспечением MAX+PLUS II. Имя проекта - это тоже самое имя, что и имя проектного файла верхнего уровня иерархии, но без расширения.

MAX+PLUS II может выполнять компиляцию, моделирование, анализ временных параметров и программирование только одного заданного проекта.

propagation delay

Задержка распространения (транспортная задержка). Время, необходимое для распространения сигнала между контактами и цепями микросхемы.

R

Radix

Основание системы счисления. Основание системы счисления. В MAX+PLUS II логические уровни шин и числовые значения вводятся и отображаются в двоичной, десятичной, шестнадцатеричной и восьмеричной системах счисления.

RAM

Оперативное запоминающее устройство с произвольной выборкой (ОЗУ). RAM можно реализовать с помощью Embedded Array Blocks (EAB-блоков встроенной памяти) в микросхемах семейств ACEX 1K и FLEX 10K или с помощью flipflops (Триггеров) или latches (Защелок) в микросхемах других семейств.

Range

Диапазон. Последовательность чисел или арифметических выражений, заключённые в скобки, которые определяют ширину шины. Первый бит диапазона является старшим, последний - младшим. Начало и конец диапазона разделяются двумя точками в графическом редакторе и в языке AHDL, двоеточием в языке Verilog HDL. Например, шина $a[2..0]$ состоит из цепей $a2$, $a1$ и $a0$. $a2$ – это старший бит, $a0$ - младший.

Raw Binary File

Двоичный программирующий файл с расширением (**.rbf**). Двоичный файл, содержащий конфигурационные данные для микросхем семейств ACEX 1K, FLEX 6000, FLEX 8000 и FLEX 10K. Он является двоичным эквивалентом Tabular Text File (Табличного текстового файла) с расширением (**.tff**).

RBF файл может использоваться в следующих схемах конфигурации Passive Parallel Synchronous (PPS), Passive Parallel Asynchronous (PPA) и Passive Serial (PS).

Register

Триггер или регистр. См. flipflop.

register packing

Объединение функций. Особенность логической ячейки микросхем семейств ACEX 1K, MAX 9000 и FLEX 10K, позволяющая реализовать в одной и той же логической ячейке комбинаторную схему и триггер с общим входом.

Вы можете реализовать объединение функций вручную путем назначения двух logic functions (Функциональных модулей) одной логической ячейке. Кроме того, команда **Global Project Logic Synthesis** (Логический синтез текущего проекта) из меню Assign имеет опцию *Automatic Register Packing* (Автоматическое объединение функций), позволяющую компилятору автоматически осуществлять объединение функций для соответствующих пар функциональных модулей. Altera рекомендует использовать опцию *Automatic Register Packing* (Автоматическое объединение функций) вместо назначений вручную логических ячеек.

Данная опция доступна как при многоуровневом, так и при стандартном синтезе. Она игнорируется, если не применима к выбранному семейству микросхем.

registered feedback

Регистровая обратная связь. Обратная связь с выхода триггера или защелки.

registered output

Регистровый выход. Выход триггера или защелки, подключённый к внешнему выходному контакту микросхемы.

registered performance

Быстродействие регистровой логики. Минимальный период тактового сигнала или максимальная тактовая частота схемы, рассчитанная Timing Analyzer (Анализатором временных параметров) MAX+PLUS II.

Минимальный период тактового сигнала равен максимальной задержке от Q выхода триггера до D или Clock Enable входа триггера плюс внутреннее время предустановки и задержка распространения в триггере. Вычисленные фазовые сдвиги тактирующего сигнала могут увеличить минимальный период тактового сигнала. Максимальная частота тактового сигнала обратно пропорциональна его минимальному периоду.

Анализатор временных параметров не вычисляет максимальную тактовую частоту для отдельного триггера.

Report File

Файл отчета с расширением **(.rpt)**. Текстовый файл ASCII, создаваемый модулем компилятора Fitter (Разводчик), показывающий использование ресурсов микросхемы данным проектом. Если генерируется ошибка модулями компилятора, предшествующими Partioner (Разделитель проекта на части), то данный файл не создается. Если Partioner (Разделитель проекта на части) генерирует ошибку, то в большинстве случаев Report File (Файл отчета) создается.

Reset

Установка исходного состояния (Сброс в "0"). Асинхронный входной сигнал с высоким активным уровнем, который устанавливает на выходе триггера низкий логический уровень или переводит конечный автомат в начальное состояние, независимо от остальных входов.

Resource

Ресурсы. Части микросхемы Altera, которые реализует определяемую пользователем задачу, например, контакты, логические ячейки.

resource assignment

Назначение ресурсов. Назначение logic function (Функциональному модулю) проекта определённого pin (Контакта), logic cell (Логической ячейки), I/O cell (Ячейки ввода/вывода), embedded cell (Встроенной ячейки), LAB (Массива логических ячеек), EAB (Блока встроенной памяти), row (Строки LAB), column (Столбца LAB) или chip (Чипа). При этом функциональному модулю назначаются физические ресурсы микросхемы.

Назначение ресурсов также может определять clique (Группы), logic option (Логические опции), connected pin (Соединенные контакты), timing requirement (Временные параметры) или local routing (Размещение в смежных логических ячейках) для отдельного функционального модуля проекта. В этом случае задаются параметры компиляции функционального модуля.

Row

Строка LAB. Горизонтальная линия LAB (Массивов логических ячеек), соединенных при помощи FastTrack Interconnect (Линий межсоединений) в микросхемах семейств ACEX 1K, FLEX 6000, FLEX 8000, FLEX 10K или MAX 9000.

RS-232 port

Порт RS-232. См. COM port.

S

SCF

См. Simulator Channel File.

SDF Output File

См. Standard Delay Format Output File.

secondary input

Вторичный вход. Входные сигналы Clock, Preset, синхронный и асинхронный Reset (Clear), синхронный и асинхронный Load триггера или конечного автомата в design file (Проектном файле).

Security Bit

Бит защиты. Бит, который предохраняет микросхемы фирмы Altera, выполненные по технологии EPROM или EEPROM от считывания, запрограммированной информации. Также он предохраняет от непреднамеренного перепрограммирования микросхемы.

Бит защиты может устанавливаться или нет для всего проекта в целом.

Segment

См. memory segment

Serial Bitstream File

Файл ASCII с расширением (**.sbf**), содержащий данные для конфигурирования микросхем семейств FLEX 6000, FLEX 8000 или FLEX 10K с помощью Bit-Blaster. Этот файл может быть создан с помощью команды **Convert SRAM Object Files**

Конвертировать SOF файлы из меню File программного модуля Compiler (Компилятор) или Simulator (Симулятор).

Serial Vector Format File

Файл ASCII с расширением (**.svf**), содержащий данные для программирования одной или нескольких микросхем на оборудовании АТЕ-типа (Automatic Test Equipment). Микросхемы MAX 3000A, MAX 7000A, MAX 7000B, MAX 7000S, MAX 7000AE, MAX 9000, EPC2 и EPC16 фирмы Altera могут программироваться с помощью SVF файлов. Цепочка JTAG может содержать любые микросхемы, поддерживающие спецификацию IEEE 1149.1 JTAG, включая микросхемы семейств ACEX 1K, FLEX 10K, FLEX 6000 и некоторые микросхемы FLEX 8000. Создавать SVF файлы можно с помощью команды **Create Jam or SVF File** (Создать Jam или SVF файл) из меню File программного модуля Compiler (Компилятор) или Simulator (Симулятор).

setup time

Время предустановки сигнала. Для flipflop (Триггера) – это минимальный временной интервал между появлением сигнала на информационном или Clock Enable входе и положительным фронтом тактового сигнала.

Для Latch (Защёлки) – это минимальный временной интервал между появлением сигнала на информационном входе и положительным фронтом сигнала Latch En-

able. Временной анализ времени предустановки и удержания для защёлок возможен только у микросхем MAX 5000. В других семействах микросхем защёлки реализуются с помощью комбинаторной логики и обратной связи.

Для асинхронных блоков RAM – это минимальный временной интервал между появлением сигналов на информационных и адресных входах и положительным или отрицательным фронтом сигнала Write Enable.

Времена предустановок триггеров, защёлок и асинхронных блоков RAM определяются структурой микросхемы и не могут быть заданы пользователем.

shared local interconnect

Общие локальные линии межсоединений. Локальные линии межсоединений микросхем семейства FLEX 6000, позволяющие сигналам распространяться с минимальными задержками между логическими ячейками одного и того же или смежных LAB (Массивов логических ячеек), или между периферийными логическими ячейками микросхемы и контактами ввода/вывода. Эти линии обеспечивают самые быстрые межсоединения в микросхемах семейства FLEX 6000.

SIF

См Simulator Initialization File.

Simulator Channel File

Файл временных диаграмм с расширением (**.scf**). Графический файл с временными диаграммами, который является как входным, так и выходным для Simulator (Симулятора). Он содержит временные диаграммы для входных цепей, которые определяют моделируемые значения, и временные диаграммы для внутренних цепей и выходных цепей, которые моделируются. Временные диаграммы в файле представляются высокими (1), низкими (0), высоко-импедансными (Z) и неопределёнными (X) логическими уровнями.

SCF файл может быть создан и просмотрен с помощью Waveform Editor (Редактора временных диаграмм). Simulator (Симулятор) автоматически создает или обновляет SCF файл во время моделирования. SCF файл может использоваться для получения входных векторов при функциональном тестировании в Programmer (Программаторе).

Simulator Initialization File

Файл инициализации Симулятора с расширением (**.sif**). Файл, сохраняющий значения и сигналы для node (Цепи), group (Шины) и memory (Памяти), введенные с помощью команд симулятора **Initialize Nodes/Groups** и **Initialize Memory** из меню Initialize или команд GROUP INIT и NODE INIT из Command File (Командного файла) с расширением (**.cmd**). SIF файл позволяет повторно использовать предварительно сохранённые сигналы цепей или шин.

Simulator Netlist File

Файл списка соединений Симулятора с расширением **(.snf)**. Двоичный файл, содержащий данные для функционального и временного моделирования, анализа временных параметров или моделирования проекта, реализованного на нескольких микросхемах. Три модуля компилятора создают разные типы SNF файлов, содержащих информацию, необходимую для различных режимов моделирования и анализа временных параметров.

Timing SNF Extractor (Экстрактор временного SNF файла) создает временной SNF файл, содержащий все данные, необходимые для временного моделирования и полного анализа временных параметров.

Functional SNF Extractor (Экстрактор функционального SNF файла) создает функциональный SNF файл, содержащий все данные, необходимые для функционального моделирования.

Linked SNF Extractor (Экстрактор SNF файла для нескольких проектов) создает объединённый SNF файл, связывающий данные из временных, функциональных и объединённых SNF файлов других предварительно скомпилированных проектов. Если все SNF файлы являются временными SNF файлами, то объединённый SNF файл также может использоваться для полного анализа временных параметров.

В данный момент времени для одного и того же проекта может существовать только один тип SNF файла.

single-range group (or bus) name

Простое имя шины. Имя шины состоит из идентификатора, который может содержать до 32 символов, за которым следует диапазон, заданный числами или арифметическими выражениями в квадратных скобках. Начало и конец диапазона разделяются двумя точками. Каждое число в диапазоне представляет отдельную цепь (или “бит шины”). Шина может содержать до 256 цепей. Идентификатор не может заканчиваться цифрой.

Например: шина `a[4..1]` состоит из цепей `a4`, `a3`, `a2` и `a1`.

В файлах Graphic Editor (Графического редактора) имя шины может состоять из последовательности нескольких простых имён шины. Первая цепь последовательности или первая цепь в диапазоне является старшим битом шины, последняя цепь последовательности или последняя цепь диапазона является младшим битом.

Например: `a[8..0]`, `b1`, `dout[6..4]`

SNF

См. Simulator Netlist File.

SOF

См. SRAM Object File.

source node

Источник сигнала. Цепь, которая обозначена как источник сигнала для анализа временных параметров. Источник сигнала задаётся с помощью команды **Timing Analysis Source** из меню Node программного модуля Timing Analyzer (Анализатор временных параметров). Им может быть любая цепью, которая является выходом primitive (Базового элемента), мегафункции, макрофункции или входного контакта.

Spike

Импульсная помеха или иголка. См. glitch.

SRAM Object File

Объектный SRAM файл с расширением **(.sof)**. Двоичный файл, генерируемый модулем компилятора Assembler, который содержит данные для конфигурирования микросхем ACEX 1K, FLEX 6000, FLEX 8000 или FLEX 10K фирмы Altera.

Standard Delay Format Output File

Выходной файл формата стандартной задержки) с расширением **(.sdo)**. Выходной файл, содержащий информацию о временных задержках, позволяющей выполнять back-annotation (Сохранение результатов) моделирования с помощью симуляторов VHDL, которые используют библиотеки моделирования, совместимые с VITAL версий 2.2b и 3.0, back-annotation (Сохранение результатов) моделирования симуляторов языка Verilog HDL, временной анализ и повторный синтез используя средства EDIF моделирования и синтеза. Standard Delay Format (SDF) является промышленным стандартом.

Модули компилятора MAX+PLUS II EDIF, VHDL и Verilog Netlist Writer могут создавать SDF Output File в формате SDF версии 2.1 или 1.0.

standard synthesis

Логический синтез со стандартными опциями. Стандартный. Логический синтез, включающий следующие логические опции:

- Fast I/Q (Быстрый ввод/вывод)
- Global Signal (Общий сигнал)
- Hierarchical Synthesis (Иерархический синтез)
- Insert Additional Logic Cell (Вставка дополнительной логической ячейки)
- Minimization (Full and Partial) (Минимизация полная и частичная)
- NOT Gate Push-Back (Сдвиг назад вентиля НЕ)
- Parallel Expanders (Параллельные расширители)
- Slow Slew Rate (Медленная скорость нарастания выходного напряжения)
- SOFT Buffer Insertion (Вставка SOFT буфера)
- Turbo Bit (Включить Turbo Bit)
- Use LPM for AHDL Operators (Использование LPM для операторов AHDL)
- XOR Synthesis (Синтез XOR)

Стандартный синтез включает только эти логические опции. Остальные опции, перечисленные в диалоговых окнах **Define Synthesis Style** и **Advanced Options** из меню Assign, доступны только при многоуровневом синтезе. Стандартный синтез возможен только для микросхем семейств Classic, MAX 5000, MAX 7000 и MAX 9000.

state bit

Бит состояния. Выход триггера, используемого конечным автоматом для хранения значения одного бита состояния конечного автомата.

state machine

Конечный автомат. Последовательная схема, которая проходит через ряд состояний. Конечный автомат может быть определен с помощью Waveform Design File (Проектного файла, описанного временными диаграммами) с расширением (**.wdf**), State Machine File (Файла конечного автомата) с расширением (**.smf**), Vector File (Векторного файла) с расширением (**.vec**), VHDL Design File (Проектного файла на языке VHDL) с расширением (**.vhd**), Verilog Design File (Проектного файла на языке Verilog) с расширением (**.v**) или с помощью описания в State Machine Declaration (Объявлении конечного автомата) в AHDL Text Design File (Проектном файле на языке AHDL) с расширением (**.tdf**).

sub-project

См. super-project.

Subdesign

Подпроект. Проектный файл более низкого уровня в проекте MAX+PLUS II, т.е., мегафункция или макрофункция фирмы Altera или функция, созданная пользователем.

Библиотеки мега- и макрофункций фирмы Altera находятся в подкаталогах **mega_lpm** и **mf** директории **\maxplus2\max2lib**. AHDL Include Files (Включаемые файлы AHDL) с расширением (**.inc**) для этих функций находятся в директориях **\maxplus2\max2lib\ mega_lpm** и **\maxplus2\max2inc**, соответственно. Component Declaration (Объявления компонентов) для функций, поддерживаемых VHDL, находятся в **maxplus2** и **megacore packages** в библиотеке **altera** и в **lpm_components packages** в библиотеке **lpm**. И та, и другая библиотеки находятся в директории **\maxplus2\vhdlnn**, где *nn* это “87” или “93”. На рабочих станциях UNIX директория **maxplus2** является подкаталогом директории **/usr**.

subdesign name или entity name

Имя подпроекта. Имя, представляющее название подпроекта. В AHDL имя подпроекта представляет собой символьное имя в кавычках или без, которое должно быть таким же, как и имя Text Design File (Текстового проектного файла) с расширением (**.tdf**). В VHDL entity name является идентификатором.

Имя подпроекта без кавычек (AHDL):

Максимальная длина	32 символа
Разрешенные символы	a-z, A-Z, 0-9 и подчеркивание (<u>)</u> Имя без кавычек не может быть зарезервированным идентификатором или ключевым словом AHDL.

Имя подпроекта в кавычках (AHDL):

Максимальная длина	32 символа
Разрешенные символы	a-z, A-Z, 0-9, тире (-) и подчеркивание (<u>)</u>

Идентификатор (VHDL):

Максимальная длина	32 символа
Разрешенные символы	a-z, A-Z, 0-9 и подчеркивание (<u>)</u> Идентификатор не может начинаться с цифры или подчеркивания и не может заканчиваться подчёркиванием, не может содержать двух подчеркиваний (<u><u>)</u>)</u> подряд. Он не может быть ключевым словом.

Идентификатор (Verilog HDL):

Максимальная длина	32 символа
Разрешённые символы	a-z, A-Z, 0-9 и подчеркивание (<u>)</u> Идентификатор не может начинаться с цифры. Идентификаторы учитывают регистр клавиатуры (прописные и строчные буквы). Нельзя использовать ключевые слова Verilog HDL.

На рабочих станции UNIX имена файлов и, следовательно, имена подпроектов учитывают регистр клавиатуры (прописные и строчные буквы).

sum-of-products

Дизъюнктивная нормальная форма (сумма произведений). Булево выражение, состоящее суммы логических произведений.

super-project и sub-project

Иерархический-проект и подпроект. Иерархический-проект состоит из проектного файла верхнего уровня, который содержит символы или функциональные модули, представляющие отдельные проекты.

Подпроектом является любой отдельный проект, который используется как часть другого проекта. Иерархический-проект можно использовать, как подпроект в другом иерархическом-проекте более высокого уровня.

SVF File

См. Serial Vector Format Fail.

Symbol File

Символьный файл с расширением **(.sym)**. Графический файл, создаваемый Symbol Editor (Редактором символов) или модулем Compiler Netlist Extractor (Экстрактор списка соединений компилятора). Он соответствует проектному файлу, т.е. мегафункции, макрофункции или primitive (Базовому элементу) MAX+PLUS II с таким же именем и может использоваться в Graphic Design Files (Графических проектных файлах) с расширением **(.gdf)**.

Если включён модуль компилятора Linked SNF Extractor (Экстрактор SNF файла для нескольких проектов), то во время компиляции символ в файле Graphic Editor (Графического редактора) соответствует Simulator Netlist File с расширением **(.snf)**, а не проектному файлу.

symbol ID number или **net ID number** (Идентификационный номер символа или цепи) Число, однозначно идентифицирующие каждую node (Цепь) и символ в проектном файле.

В Graphic Editor (Графическом редакторе) это число находится в нижнем левом углу символа и соответствует порядковому номеру символа при его вводе в файл графического редактора. В остальных типах проектных файлов компилятор присваивает ID number (Идентификатор цепи) при компиляции проекта. В окне Hierarchy Display (Дисплея иерархии проекта) к имени каждого проектного файла более низкого уровня добавляется двоеточие (:) и ID number (Идентификационный номер) или имя мега-, макрофункции на языке AHDL, VHDL или Verilog HDL.

T

Table File

Табличный файл с расширением **(.tbl)**. Файл ASCII, который содержит в табличном формате все входные вектора и выходные логические уровни текущего Vector File (Векторного файла) с расширением **(.vec)** или Simulator Channel File (Файла временных диаграмм) с расширением **(.scf)**. Table File (Табличный файл) может создаваться Simulator (Симулятором) или Waveform Editor (Редактором временных диаграмм).

Tabular Text File

Табличный текстовый файл с расширением **(.tff)**. Текстовый файл ASCII в табличном формате, содержащий данные для конфигурации микросхем ACEX 1K, FLEX 6000, FLEX 8000 и FLEX 10K. TTF файл является эквивалентом Raw Binary File с расширением **(.rbf)** в десятичной системе исчисления.

Компилятор MAX+PLUS II автоматически создает TTF файлы, содержащие данные для последовательных Passive Parallel Synchronous (PPS), Passive Parallel Asynchronous (PPA) и Passive Serial (PS) конфигурационных схем микросхем FLEX 8000, для PS конфигурационной схемы микросхем ACEX 1K и FLEX 10K,

для PS и Passive Serial Asynchronous (PSA) конфигурационных схем микросхем FLEX 6000.

После компиляции можно создать TTF файлы, поддерживающие другие конфигурационные схемы микросхем ACEX 1K, FLEX 6000, FLEX 8000 и FLEX 10K.

t_{co} clock to output delay

Задержка между тактовым сигналом и выходом. Время, требуемое для получения истинного сигнала на выходном контакте триггера после изменения тактового сигнала на входном контакте этого триггера. Это время показывает задержку распространения сигнала между внешними контактами.

t_{co}

Время, которое можно задать, чтобы определить максимально допустимую задержку между изменением тактового импульса и появлением сигнала на выходе. В MAX+PLUS II можно задать требуемое t_{co} для всего проекта или для отдельного входного (INPUT, INPUTC), выходного (OUTPUT, OUTPUTC) или двунаправленного (BIDIR, BIDIRC) контакта.

TDF

См. Text Design File.

ternary operator

Троичный (тернарный) оператор. Оператор языка AHDL, который выполняет одно из двух действий в зависимости от значения арифметического выражения. Троичный оператор имеет следующий формат:

<expr 1(выражение1)> ? <expr 2(выражение2)> : <expr 3(выражение3)>

Если значение первого выражения является ненулевым (истинным), то вычисляется второе выражение, и оно является результатом троичного оператора. В противном случае, вычисляется третье выражение, и оно является результатом троичного оператора.

Text Design Export File

Экспортный текстовый проектный файл с расширением (**.tdx**). Текстовый файл ASCII на языке AHDL, который может создаваться при компиляции Xilinx Netlist Format File (Файла формата списка соединений Xilinx) с расширением (**.xnf**). Он будет содержать такую же логику, что и файл XNF.

Text Design Export File может сохраняться как Text Design File (Текстовый проектный файл) с расширением (**.tdf**) и использоваться для замены соответствующего XNF файла в проекте.

Text Design File

Текстовый проектный файл с расширением (**.tdf**). Текстовый файл ASCII, написанный на языке AHDL. Text Design Export File с расширением (**.tdx**) (Экспортный текстовый проектный файл) и Text Design Output File (**.tdo**) (Выходной тек-

стовый проектный файл) могут сохраняться как TDF файлы и компилироваться MAX+PLUS II.

Text Design Output File

Выходной текстовый проектный файл с расширением (**.tdo**). Текстовый файл ASCII, создаваемый компилятором MAX+PLUS II, содержащий описание на языке AHDL эквивалентное логики проекта.

Компилятор создает TDO файл, также как и Assignment & Configuration Output File (Выходной файл назначений и конфигураций) с расширением (**.aco**) при компиляции проекта, если включена команда **Generate AHDL TDO File** из меню Processing программного модуля Compiler (Компилятор).

Вы можете сохранить файл TDO как Text Design File (Текстовый проектный файл) с расширением (**.tdf**) и повторно его компилировать. Чтобы оставить сделанные назначения для микросхемы, нужно сохранить Assignment & Configuration Output File (Выходной файл назначений и конфигураций) с расширением (**.aco**) как Assignment & Configuration File (Файл назначений и конфигураций) с расширением (**.acf**). Файлы TDO облегчают сохранение сделанного логического синтеза проекта.

time unit

Единица времени. Единица времени в MAX+PLUS II.

Допустимы следующие единицы времени:

Единица времени	Сокращение
ns	Наносекунда
ms	Миллисекунда
us	Микросекунда
s	Секунда
mhz	Мегагерц

Значение времени не разделяется пробелом с единицей времени. Если вы не задаёте единицу времени, то в зависимости от контекста по умолчанию принимается либо ns, либо mhz.

Timing Analyzer Output File

Выходной файл Анализатора временных параметров с расширением (**.tao**). Текстовый файл ASCII, который используется для сохранения результатов анализа временных параметров, отображаемых Timing Analyzer (Анализатором временных параметров) MAX+PLUS II.

timing assignment

Назначение временных параметров. Назначения, определяющие желаемое быстроедействие одного или нескольких logic function (Функциональных модулей).

Возможно назначение временных параметров t_{PD} , t_{SU} , t_{CO} и f_{MAX} и “timing cuts” (Исключение временных путей). Возможно назначить временные параметры от-

дельным logic functions (Функциональным модулям) или всему проекту в целом, как временные параметры по умолчанию. Назначения временных параметров влияет на компиляцию проекта только для микросхем семейств ACEX 1K, FLEX 6000, FLEX 8000 и FLEX 10K.

timing cuts

Исключение временных путей. Способ задания временных параметров, при котором исключаются некоторые пути распространения сигналов, показывая компилятору, что он не должен учитывать задержки распространения сигналов между указанными цепями пытаясь достичь заданного быстродействия проекта.

t_{PD} input to non-registered output delay

Задержка между входом и выходом комбинаторной логики. Время распространения сигнала от входного контакта через комбинаторную логику до выходного контакта.

t_{PD} - это время, которое можно задать, чтобы определить максимально допустимую задержку распространения сигнала от входа до выхода комбинаторной логики. В MAX+PLUS II можно задать требуемое t_{PD} для всего проекта или для отдельного входного (INPUT, INPUTC), выходного (OUTPUT, OUTPUTC) или двунаправленного (BIDIR, BIDI RC) контакта.

tri-state buffer

Буфер с третьим состоянием. Буфер с входным, выходным и управляющим сигналом Output Enable. Если на входе Output Enable высокий логический уровень, то на выходе такой же сигнал, что и на входе. Если на входе Output Enable низкий логический уровень, то выход находится в состоянии высокого сопротивления. Буфер с третьим состоянием реализуется с помощью primitive (Базового элемента) TRI.

В Graphic Editor (Графическом редакторе) шины с третьим состоянием могут быть реализованы с помощью символа базового элемента TRI и толстой линии, в AHDL файле с помощью переменной TRI_STATE_NODE в файле AHDL.

t_{SU} clock setup time

Время предустановки. Величина времени, на которую данные на информационном или Enable входе триггера должны появиться раньше активного фронта тактирующего сигнала.

t_{SU} - это время, которое можно задать, чтобы определить допустимое значение времени предустановки сигнала. В MAX+PLUS II можно задать требуемое t_{SU} для всего проекта или для отдельного входного (INPUT, INPUTC) или двунаправленного (BIDIR, BIDI RC) контакта.

TTF

См. Tabular Text File

Turbo Bit и logic cell Turbo Bit

Бит повышения быстродействия. Бит для управления быстродействием и потребляемой мощностью микросхемы фирмы Altera. Логическая опция Turbo Bit применима к мегафункциям, макрофункциям и контактам. Если Turbo Bit установлен, то быстродействие увеличивается, если он сброшен, то уменьшается потребляемая мощность. Значение Turbo Bit может определяться в проектном файле или компилятором.

Доступность Turbo Bit отличается для разных семейств микросхем:

Семейство микросхем Altera	Доступность Turbo Bit
Classic	Применима ко всей микросхеме (назначается как опция микросхемы)
MAX 5000	Недоступна
MAX 7000	Применима к отдельным логическим ячейкам внутри микросхемы (назначается как логическая опция)
MAX 9000	Применима к отдельным логическим ячейкам внутри микросхемы (назначается как логическая опция)
FLEX 6000	недоступна
FLEX 8000	недоступна
FLEX 10K	недоступна

В микросхемах MAX 7000E опция *Turbo Bit* (Повышение быстродействия) и опция выходного контакта *Slow Slew Rate* (Медленная скорость нарастания фронта) управляются одним битом. Следовательно, для выходного контакта может быть включена только одна из этих опций. Если обе опции включены или обе опции выключены, то компилятор использует значение опции *Slow Slew Rate* и игнорирует *Turbo Bit* для выходных контактов. Для входных контактов и внутренних логических ячеек компилятор использует значение *Turbo Bit* и игнорирует значение *Slow Slew Rate*.

two's compliment

Дополнительный код. Представление двоичных чисел, при котором отрицательному числу соответствует его инверсия плюс 1. В AHDL считается, что в арифметических операторах используются числа в дополнительном коде. В VHDL числа в дополнительном коде должны иметь signed data type (Тип чисел со знаком).

U

user libraries

Библиотеки пользователя. Одна или несколько директорий, которые содержат ваши собственные мегафункции, макрофункции, Symbol Files (Символьные файлы) с расширением **(.sym)**, AHDL Include Files (Включаемые файлы AHDL) с расширением **(.inc)** или предварительно скомпилированные определённые пользователем VHDL packages.

Компилятор автоматически обращается в эти определённые пользователем библиотеки при компиляции проекта. Команда компилятора **VHDL Netlist Setting** из меню Interface определяет проектные VHDL библиотеки для текущего проекта. С помощью команды **User Libraries** (Библиотеки пользователя) из меню Options можно определить директории, содержащие другие ваши библиотеки.

V

Variable

Переменная. Символическое имя, представляющее функциональный узел. В AHDL переменная может представлять конечный автомат, primitive (Базовый элемент), мегафункцию или макрофункцию. Переменная объявляется в Variable Section. В VHDL переменные имеют единственное текущее значение. Они объявляются и используются только в processes (Процессах) и subprograms (Подпрограммах). Переменная VHDL объявляется в Variable Declaration. Её значение можно изменить с помощью Variable Assignment Statement.

VCC

Напряжение высокого уровня (Напряжение питания).

По умолчанию VCC является активным значением сигнала. В AHDL Text Design File (Текстовом проектном файле на языке AHDL) с расширением **(.tdf)**, VCC является предопределённой константой и ключевым словом. В VHDL Design File (Проектном файле на языке VHDL) с расширением **(.vhd)** VCC представляется '1'. В Verilog Design File (Проектном файле на языке Verilog) с расширением **(.v)** VCC представляется 1. В файле графического редактора VCC является символом. VCC представляется как высокий (1) логический уровень в Simulator (Симуляторе) и Waveform Editor (Редакторе временных диаграмм).

Vector

Вектор. Вектор определяет логические уровни для отдельной node (Цепи) проекта. Simulator (Симулятор) использует векторы для моделирования поведения

проекта. Programmer (Программатор) и Simulator (Симулятор) используют векторы для функционального тестирования.

Векторы для моделирования и функционального тестирования могут быть определены в Vector Files (Векторных файлах) с расширением (.vec) или в Simulator Channel Files (Файлах временных диаграмм) с расширением (.scf). Вектора для функционального тестирования также могут быть сохранены в программирующих файлах.

Вектора для ввода проекта могут быть определены в Waveform Design File (Проектном файле, описанном временными диаграммами) с расширением (.wdf).

Vector File

Векторный файл с расширением (.vec). Файл ASCII, содержащий вектора, задающие логические уровни входных nodes (Цепей) проекта. Simulator (Симулятор) использует этот файл для тестирования логических операций проекта. Programmer (Программатор) и Simulator (Симулятор) используют векторный файл для функционального тестирования.

Кроме того, векторный файл можно конвертироваться в Waveform Design File (Проектный файл, описанный временными диаграммами) с расширением (.wdf).

Verilog Design File

Проектный файл на языке Verilog с расширением (.v). Файл Verilog HDL, создаваемый с помощью Text Editor (Текстового редактора) MAX+PLUS II или любого другого стандартного текстового редактора. Verilog Design File компилироваться с помощью Compiler (Компилятора) MAX+PLUS II.

Verilog HDL

Язык описания аппаратных средств (HDL).

Можно создать Verilog Design File с расширением (.v) с помощью Text Editor (Текстового редактора) MAX+PLUS II или любого другого стандартного текстового редактора и компилировать его непосредственно с помощью Compiler (Компилятора) MAX+PLUS II.

Также можно создать EDIF 2 0 0 или 3 0 0 netlist файл из проектного файла Verilog HDL, обработанного с помощью средств синтеза Verilog HDL, и затем импортировать этот файл в MAX+PLUS II как EDIF Input File (Входной файл EDIF) с расширением (.edf). Кроме того, возможно обрабатывать Verilog Design File непосредственно в MAX+PLUS II с помощью средств фирмы Synopsys. Для этого нужно включить команду **Synopsys Compiler** из меню Interfaces программного модуля Compiler (Компилятор). Компилятор MAX+PLUS II также может генерировать Verilog Output File (Выходной файл Verilog) с расширением (.vo), содержащий информацию для моделирования с помощью стандартных симуляторов Verilog HDL.

Verilog Output File

Выходной файл Verilog. Стандартный netlist файл на языке описания аппаратных средств (HDL) Verilog, генерируемый модулем компилятора Verilog Netlist Write (Генератор Verilog HDL файла). Этот файл может передаваться Verilog HDL симулятору для моделирования. Verilog Output File не может компилироваться компилятором MAX+PLUS II.

VHDL

Very High Speed Integrated Circuit Hardware Description Language. Язык описания аппаратных средств сверхбыстродействующих интегральных схем (VHSIC).

Возможно создавать VHDL Design File (Проектный файл на языке VHDL) с расширением **(.vhd)** с помощью Text Editor (Текстового редактора) MAX+PLUS II или любого другого стандартного текстового редактора и компилировать его непосредственно MAX+PLUS II. Возможно создать EDIF 2 0 0 или 3 0 0 netlist файл из проектного файла VHDL, обработанного с помощью средств синтеза VHDL, и затем импортировать этот файл в MAX+PLUS II как EDIF Input File (Входной файл EDIF) с расширением **(.edf)**. Компилятор MAX+PLUS II может также генерировать VHDL Output File (Выходной VHDL файл) с расширением **(.vho)**, содержащий информацию для моделирования с помощью симулятора VHDL, и VHDL Memory Model Output File (Выходной файл моделей памяти) с расширением **(.vmo)**, содержащий модели блоков памяти RAM или ROM.

VHDL Design File

Проектный файл на языке VHDL с расширением **(.vhd)**. Текстовый файл ASCII, написанный на языке VHDL. VHDL Design File может компилироваться компилятором MAX+PLUS II.

VHDL Memory Model Output File

Выходной файл моделей памяти VHDL) с расширением **(.vmo)**. Стандартный VHDL netlist файл, создаваемый компилятором и содержащий VHDL simulation models (VHDL модели). Компилятор автоматически создает VHDL Memory Model Output File при генерации EDIF Output File с расширением **(.edo)** для проекта, содержащего один или несколько блоков RAM или ROM.

VHDL Output File

Выходной VHDL файл) с расширением **(.vho)**. Стандартный VHDL netlist файл, создаваемый модулем компилятора VHDL Netlist Writer (Генератор VHDL netlist файла). Этот файл может передаваться средствам VHDL моделирования. VHDL Output File не может обрабатываться компилятором MAX+PLUS II.

VITAL - VHDL Initiative Toward ASIC Libraries

Стандартный промышленный формат для VHDL библиотек моделирования. MAX+PLUS II обеспечивает VHDL cell simulation models (VHDL модели ячеек), соответствующие VITAL версии 2.2b и версии 3.0 (VITAL 95). Кроме того, мо-

дули EDIF, VHDL и Verilog Netlist Writer компилятора MAX +PLUS II могут генерировать Standard Delay Format (SDF) Output File с расширением **(.sdo)** для использования с VITAL-совместимыми библиотеками моделирования.

VMO File

См. VHDL Memory Model Output File.

W

Waveform Design File

Проектный файл, описанный временными диаграммами) с расширением **(.wdf)**. Графический файл проекта, описанного временными диаграммами, создаваемый с помощью Waveform Editor (Редактора временных диаграмм) MAX+PLUS II. Этот файл описывает логику с помощью временных диаграмм, соответствующих высокому (1), низкому (0), неопределённому (X) и высокоимпедансному (Z) логическим уровням. Он также может включать конечные автоматы с временными диаграммами, представляющими имена их состояний.

waveform interval

Участок временной диаграммы. В Waveform Editor (Редакторе временных диаграмм) участок временной диаграммы node (Цепи) или шины, представляющий её логические уровни или имена состояния за определённый промежуток времени.

WDF

См. Waveform Design File.

X

Xilinx Netlist Format (XNF) File

Файлы формата списка соединений Xilinx с расширением **(.xnf)**. Netlist файл, создаваемый программным обеспечением фирмы Xilinx. XNF файлы, создаваемые утилитой Xilinx LCA2XNF могут обрабатываться непосредственно компилятором MAX+PLUS II. XNF файл может определять всю логику проекта или использоваться на нижнем уровне иерархического проекта.