

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
РОССИЙСКОЙ ФЕДЕРАЦИИ

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ЯДЕРНЫЙ
УНИВЕРСИТЕТ «МИФИ»

ФАКУЛЬТЕТ АВТОМАТИКИ И ЭЛЕКТРОНИКИ

В.П. Петропавловский, С.Г. Микульский, К.А. Сарксян

ЛАБОРАТОРНЫЙ ПРАКТИКУМ
«ПРОЕКТИРОВАНИЕ ЦИФРОВЫХ УСТРОЙСТВ
НА ПРОГРАММИРУЕМЫХ ЛОГИЧЕСКИХ
ИНТЕГРАЛЬНЫХ СХЕМАХ»

(Виртуальная микроэлектроника)

Под редакцией В.П. Петропавловского

Москва 2012

УДК 004.896:621.382.049.77 (076.5)

ББК 32.97-02я7

П 31

Петропавловский В.П., Микульский С.Г., Сарксян К.А. «Лабораторный практикум «Проектирование цифровых устройств на программируемых логических интегральных схемах». Виртуальная микроэлектроника»: учебное пособие/ Под ред. В.П. Петропавловского. М.: НИЯУ МИФИ, 2012. — 104 с.

Рассматриваются вопросы разработки на базе ПЛИС цифровых схем и автоматов, компьютерного моделирования их работы, заменяющего этап лабораторного макетирования, и их реализации в виде интегральной ИМС. Отличием настоящего пособия от аналогичных книг является использование сквозной технологии схемного и программного (язык VHDL) проектирования электронных устройств на базе пакета MAX+PLUS II фирмы ALTERA, работающего в среде WINDOWS.

Пособие предназначено для студентов, изучающих курсы «Машинные методы проектирования электронных схем», «ПЛИС» и «САПР измерительных систем» по специальности «Электроника и автоматика физических установок», а также аспирантов всех факультетов МИФИ и специалистов государственных и коммерческих организаций, желающих освоить новую виртуальную технологию проектирования цифровой электронной аппаратуры на базе ПЛИС с целью повышения эффективности проектирования и резкого сокращения сроков разработки РЭА.

Рекомендовано УМО «Ядерные физика и технологии» в качестве учебного пособия для студентов и аспирантов учебных заведений по специализации «Электроника и автоматика физических установок».

Рецензент канд. техн. наук В.А. Юхневич

ISBN 978-5-7262-1717-8

© Национальный исследовательский ядерный университет «МИФИ», 2012.

Содержание

Введение.....	6
Работа 1. Разработка принципиальной схемы цифрового устройства с помощью графического редактора Graphic Editor.....	13
1.1. Запуск системы проектирования MAX+PLUS II.....	13
1.2. Методика проектирования принципиальной схемы с помощью графического редактора Graphic Editor.....	15
1.2.1. Создание нового проекта и работа с библиотеками.....	15
1.2.2. Изображение и ввод электронных компонентов схемы, назначение имен входных и выходных контактов ПЛИС.....	20
1.2.3. Выполнение электрических соединений и шин	26
1.2.4. Проверка ошибок и сохранение файла схемы.....	32
1.2.5. Компиляция файла схемы (организация базы данных) с помощью редактора Compiler и выбор типа ПЛИС.....	33
1.3. Структурная схема алгоритма разработки цифровой схемы.....	38
<i>Подготовка к работе</i>	43
<i>Рабочее задание</i>	43
<i>Контрольные вопросы</i>	43
Работа 2. Моделирование работы программируемых логических интегральных схем с помощью редакторов Waveform Editor и Simulator 45	
2.1. Методика моделирования цифровых схем с помощью редакторов Waveform Editor и Simulator.....	45
2.1.1. Загрузка проекта схемы, включение сигнального редактора Waveform Editor и создание файла входных сигналов (Simulator Channel File).....	45
2.1.2. Ввод имен внешних и внутренних контактов ПЛИС для подключения и контроля сигналов.....	48
2.1.3. Создание и редактирование входных сигналов для моделирования работы ПЛИС.....	50
2.1.4. Моделирование работы ПЛИС с помощью программы Simulator, редактирование стимулов и трасс.....	54
2.2. Структурная схема алгоритма моделирования работы ПЛИС.....	56
<i>Подготовка к работе</i>	58
<i>Рабочее задание</i>	58
<i>Контрольные вопросы</i>	59
Работа 3. Программирование ПЛИС: технические и программные средства Starter Kit.....	60
3.1. Технические средства проектирования и отладки работы ПЛИС..61	
3.1.1. Программирование ПЛИС на базе универсальной платы UP-1 фирмы ALTERA.....	61

3.1.2. Интерфейс (линии связи ПЛИС с ПК) типа ByteBlaster.....	66
3.2. Программные средства загрузки и отладки файла принципиальной схемы в ПЛИС с помощью компонента Programmer.....	67
3.2.1. Установки в компоненте Programmer для ПЛИС EPM7128S.....	67
3.2.2. Установки в редакторе Programmer для ПЛИС EPF10K20.....	68
3.3. Программирование ПЛИС.....	69
3.3.1. Назначение номеров выводов.....	69
3.4. Структурная схема алгоритма программирования ПЛИС.....	75
<i>Подготовка к работе</i>	76
<i>Рабочее задание</i>	76
<i>Контрольные вопросы</i>	76
Работа 4. Разработка цифровой схемы на языке VHDL.....	77
4.1. Методика проектирования схем на языке VHDL.....	77
4.1.1. Создание нового проекта	77
4.1.2. Ввод описания цифровой схемы на языке VHDL.....	79
4.2. Оформление программы на языке VHDL с помощью стандартного шаблона типа Templates	80
4.3. Сохранение и компиляция файла.....	82
4.4. Структурная схема алгоритма создания цифровой схемы на языке VHDL.....	84
<i>Подготовка к работе</i>	85
<i>Рабочее задание</i>	85
<i>Контрольные вопросы</i>	85
Список рекомендуемой литературы	86
Приложение 1. Примеры разработки цифровых схем на ПЛИС.....	87
<i>Вариант № 1</i>	87
<i>Вариант № 2</i>	89
<i>Вариант № 3</i>	90
<i>Вариант № 4</i>	92
<i>Вариант № 5</i>	93
Приложение 2. Описание и характеристики технических средств проектирования ПЛИС.....	94
Устройства ввода-вывода лабораторного макета UP1.....	94
Электронные компоненты, относящиеся к микросхеме семейства MAX7000.....	94
Приложение 3. Описание и характеристики отечественных ПЛИС 5576XC1T, 5576XC1T1.....	100
Приложение 4. Установка на ПК пакета MAX+plus II.....	103

Введение

Главными отличиями специалиста в области радиоэлектроники 21-го века от радиоинженеров века предыдущего являются:

- способность принятия решения при разработке электронного прибора или системы: какую часть устройства реализовать *аппаратно*, а какую — *программно*;
- выбор современной микροэлектронной элементной базы для разрабатываемого устройства определяется не тем, *как сделать*, а на *чем* (на каких микросхемах) сделать;
- переход от традиционного схемного мышления при проектировании РЭА к программному (на языке VHDL) методу создания цифровых устройств.

Поясняя эти положения парадигмы проектирования продвинутой радиоэлектроники, хотелось бы, *во-первых*, подчеркнуть необходимость добавления к процессу обучения специалистов не только традиционной аналоговой и импульсной радиоэлектронике, но и принятой на Западе – Computer science – или техническим и программным средствам микропроцессорной техники, включая сюда и теорию — дискретную математику. То есть алгоритм работы цифрового автомата можно реализовать в виде «железа» на *микροэлектронных СБИС* или с помощью *программы*, выполняемой микроконтроллером.

Во-вторых, современные классы цифровой элементной базы СБИС (например, ПЛИС, микроконтроллеры и сигнальные процессоры) имеют библиотеки готовых цифровых схем, позволяющих реализовать до 90 % стандартных задач. Поэтому специалист при решении задачи думает не о схемной реализации устройства, а о том, какую выбрать элементную базу с учетом технического задания и экономических характеристик.

Современное состояние электроники. Традиционный *романтический* образ электронщика конца 20-го века представлял собой человека с дымящимся паяльником в руках, согнувшегося над печатной платой, в которой он что-то выпаивал, регулировал и наблюдал осциллограммы в разных точках электронной схемы, собранной на печатной плате. Соответственно и весь учебный процесс в области радиоэлектроники подстраивался и до настоящего времени основан на таком стереотипе мышления.

Однако, начиная с 90-х годов прошлого века и особенно за последние годы, в мире на базе персональных компьютеров ПК неслышно произошла очередная научно-техническая революция в области микроэлектроники, к которой в нашей стране из-за разных причин вузовская система образования оказалась неготовой. Появление больших БИС и сверхбольших СБИС интегральных микросхем ИМС, на которых были реализованы однокристальные микроЭВМ (микроконтроллеры), программируемые логические ИМС (ПЛИС) и процессоры с цифровой обработкой сигналов (DSP или ЦОС-процессоры), позволило на порядок увеличить как сложность, так и качество разрабатываемой электронной аппаратуры. Поэтому к современному специалисту сегодня предъявляются следующие требования:

- свободная работа с персональными компьютерами и их программными продуктами типа MS Office;
- умение работать в сети Internet;
- работа на уровне администратора в сетях на основе Microsoft Windows;
- конструирование механических конструкций электронных приборов и систем с помощью САПР типа AutoCAD;
- разработка принципиальных схем, моделирование их работы и проектирование печатных плат с помощью САПР типа OrCAD и PCAD ;
- проектирование, моделирование и анализ аналоговых электронных схем с помощью САПР типа SPICE;
- проектирование цифровых устройств на языке VHDL;
- проектирование цифровых устройств на традиционных однокристальных микроконтроллерах и AVR-микроконтроллерах со встроенной Flash-памятью до нескольких мегабайт;
- проектирование сверхскоростных цифровых устройств на RISC-процессорах с цифровой обработкой сигналов (DSP-процессоры типа TMS и ADSP);
- проектирование нестандартных цифровых узлов и автоматов на программируемых логических интегральных микросхемах (ПЛИС) фирм ALTERA и XILINX с помощью САПР типа MAX+PLUS II (QUARTOS) и др.;

- проектирование нового поколения цифровых устройств и автоматов на одном кристалле СБИС с помощью SOC-технологии (system on chip);
- умение работать с современными программными и техническими средствами отладки и проектирования микропроцессорной техники типа внутрисхемных эмуляторов, логических и сигнатурных анализаторов (20—50-канальные осциллографы), паяльных станций для работы с корпусами ИМС на 400—700 выводов и т.д.

Экономический кризис в США, Европе и Юго-Восточной Азии показал, что *ставка на дешевые рабочие руки не имеет никаких перспектив*, так как один универсальный специалист (системный интегратор) в электронике с помощью современных технологий в области САПР и программно-технического инструментария с успехом может заменить большие коллективы *специалистов старших поколений*, придерживающихся «паяльных» технологий.

Таким образом, в современном мире, поделившем все материальные, экономические и финансовые ресурсы, остался последний неподделанный интеллектуальный ресурс в виде специалистов, овладевших продвинутыми технологиями в науке и технике, за которых на Западе уже сегодня ведется ожесточенная борьба. Вполне понятно, что в будущем преуспеют страны, освоившие высокие технологии, а все остальные будут отброшены на обочину истории и обречены на жалкое прозябание в виде поставщиков сырья и дешевой рабочей силы.

Освоение и применение продвинутых технологий проектирования в радиоэлектронике позволит:

- студентам, аспирантам и специалистам промышленности повысить на порядок сложность решаемых задач при проектировании микроэлектронных устройств и автоматов;
- снизить габариты, вес и объем цифровых устройств, а также резко повысить их надежность;
- увеличить быстродействие микроэлектронных систем;
- сократить на порядок сроки разработки, изготовления и отладки радиоэлектронной аппаратуры (РЭА);
- *при решении стандартных задач* снизить требования к квалификации специалистов в области радиоэлектроники и

сделать микроэлектронику доступной для более широкой массы специалистов, в том числе и радиолюбителей.

На сей день ни один из вузов России не может обеспечить освоение студентами перечисленных выше знаний из-за известных объективных, субъективных, а также экономических и социальных причин [2, 3], поэтому будущему специалисту следует надеяться только на себя.

Настоящее пособие является *вторым шагом* [1] по созданию *русской электронной энциклопедии*, с помощью которой современный специалист сможет при соответствующих интеллектуальных и материальных усилиях и затратах стать системным интегратором в радиоэлектронике. В рамках электронной энциклопедии далее предполагается выпуск книг по следующим темам:

- Проектирование, моделирование и анализ аналоговых электронных схем с помощью САПР типа SPICE;
- Проектирование печатных плат на базе пакетов типа PCAD;
- Проектирование цифровых устройств с помощью языка VHDL;
- Проектирование нестандартных цифровых узлов и автоматов на программируемых логических интегральных микросхемах (ПЛИС);
- Проектирование цифровых устройств на однокристальных микроконтроллерах;
- Проектирование сверхскоростных цифровых устройств на DSP-процессорах (типа TMS и SHARC);
- Проектирование нового поколения цифровых устройств и автоматов на одном кристалле СБИС с помощью SOC-технологии (system on chip).

Освоение методики проектирования цифровых устройств и автоматов на программируемых логических интегральных схемах (ПЛИС) является неотъемлемой частью подготовки продвинутого специалиста в области радиоэлектроники 21-го века.

Сущность рассматриваемой ИТ-технологии проектирования цифровых устройств (на примере ПЛИС фирмы Altera) заключается в разработке и загрузке файла принципиальной схемы цифрового устройства в корпус интегральной схемы, расположенной на универсальной плате, которая совместно с персональным компью-

тером ПК входит в программный (пакет MAX+plus II) и аппаратный (лабораторный макет UP-1) комплекс Starter Kit, полностью обеспечивающий все этапы проектирования и отладки цифровых схем (рис. В.1).

Таким образом, принятые в прошлом веке и существующие до сих пор в России, традиционные этапы проектирования, изготовления, отладки и проверки работоспособности лабораторного макета с последующим изготовлением опытного образца цифрового устройства реализуется сегодня в коммерческих центрах и лабораториях с помощью автоматизированного рабочего места (АРМ) на базе программных и технических средств комплекса Starter Kit фирмы Altera.



Рис. В.1. Структурная схема проектирования ПЛИС на базе программных и технических средств фирмы Altera

Предлагаемая IT-технология позволяет не только резко сократить сроки разработки и изготовления стандартных узлов и блоков цифровой аппаратуры от 5—10 мин до нескольких часов, но и повысить качество всех остальных технических характеристик электронной аппаратуры.

Лабораторный практикум посвящен изучению технических и программных средств для проектирования цифровых устройств на **программируемых логических интегральных микросхемах**

(ПЛИС) на базе пакета **MAX+plus II (QUARTUS II)** фирмы **Altera**, обеспечивающего полный цикл создания цифрового устройства: от разработки принципиальной схемы вплоть до программирования интегральной микросхемы (ПЛИС). Дается перечень и описываются характеристики технических средств и их возможности по самостоятельному изготовлению студентами специализированных интегральных микросхем (ИМС), входящих в состав электронных измерительных систем.

САПР MAX+plus II представляет собой интегрированную среду для разработки цифровых устройств на базе программируемых логических интегральных схем (ПЛИС), которая обеспечит студенту (разработчику цифровых устройств) выполнение этапов, включающих в себя:

- создание проектов устройств;
- синтез структур и трассировку внутренних связей ПЛИС;
- подготовку данных для программирования или конфигурирования ПЛИС (компиляцию);
- верификацию проектов (функциональное моделирование и временной анализ);
- программирование или конфигурирование ПЛИС.

Задача данного лабораторного практикума — изучение методики проектирования ПЛИС на базе учебного проекта с его реализацией на лабораторном макете в виде работающего цифрового устройства. Практикум включает в себя четыре лабораторные работы, в процессе выполнения которых студенты осваивают систему MAX+plusII, изучают методы проектирования и моделирования цифровых устройств (с использованием также и языка VHDL), выполняют разработку и реализацию на ПЛИС типовых цифровых узлов и блоков.

Лабораторные работы практикума посвящены таким основным этапам проектирования ПЛИС, как разработка и моделирование работы принципиальной схемы цифрового устройства, выбор типа и корпуса ИМС (привязка к конкретному типу ПЛИС) с последующей компиляцией принципиальной схемы (ПС) в двоичный файл и ее загрузки в ПЛИС.

Особенность этого практикума — использование *продвинутых западных технологий* в виде сертифицированных **программных и**

технических средств фирмы Altera (поставляемых МИФИ как члену ее университетской программы), которые могут применяться на разных этапах самостоятельного изучения процедуры **виртуальной** разработки цифровых устройств с возможностью подключения готовых решений к компьютеру и исследования их характеристик. *Кроме того, в пособии приводятся характеристики лицензионных аналогов ПЛИС фирмы Altera, выпускаемых нашей промышленностью.*

Коллектив авторов выражает глубокую признательность начальнику управления издания литературы НИЯУ МИФИ Татьяне Владимировне Волвенковой, редактору Евгению Григорьевичу Станкевичу, начальнику центра учебно-методического и информационного обеспечения НИЯУ МИФИ Андрею Николаевичу Кармазинскому, коллективу кафедры «Электронные измерительные системы» НИЯУ МИФИ и лично заместителю заведующего кафедрой Белопольскому Владимиру Максимовичу, а также студентам групп А10-08 и Т7-21 за неоценимый вклад в подготовку настоящего издания, а также высказанные замечания и предложения по содержанию и оформлению лабораторных работ.

Работа 1

Разработка принципиальной схемы цифрового устройства с помощью графического редактора Graphic Editor

Цель — освоение методики проектирования принципиальных схем цифровых устройств и автоматов для ПЛИС с помощью графического редактора Graphic Editor пакета MAX+plus II.

При проектировании ПЛИС необходимо:

- установить на персональном компьютере пакет MAX+plus II (см. Приложение 4);
- с помощью графического редактора Graphic Editor пакета MAX+plus II разработать или загрузить из библиотеки принципиальную схему цифрового устройства;
- с помощью редактора Waveform Editor задать входные сигналы для принципиальной схемы и провести моделирование ее работы в редакторе Simulator;
- загрузить принципиальную схему цифрового устройства в ПЛИС с помощью редактора Programmer, который позволяет программировать, конфигурировать, проводить верификацию и тестировать ПЛИС.

1.1. Запуск системы проектирования MAX+PLUS II

Систему MAX+PLUS II можно запустить двумя способами: щелкнув дважды левой кнопкой мыши на пиктограмме MAX+PLUS II или набрав в начальном окне Windows команду Пуск/Все программы/ MAX+plus II. При запуске системы MAX+PLUS II автоматически открывается главное окно Менеджера пакета (рис. 1.1), меню которого охватывает все приложения системы MAX+PLUS II. В самой верхней строке содержится имя последнего проекта, с которым велась работа. Две следующие строки являются обычными для Windows-приложений: строка главного меню (ГМ) и панель инструментов, в левой части которой расположены обычные инструменты Windows (New, Open, Save, Print, Cut,

Copy, Paste, Undo), а в правой — специфические инструменты пакета, с помощью которых осуществляется запуск основных приложений пакета.

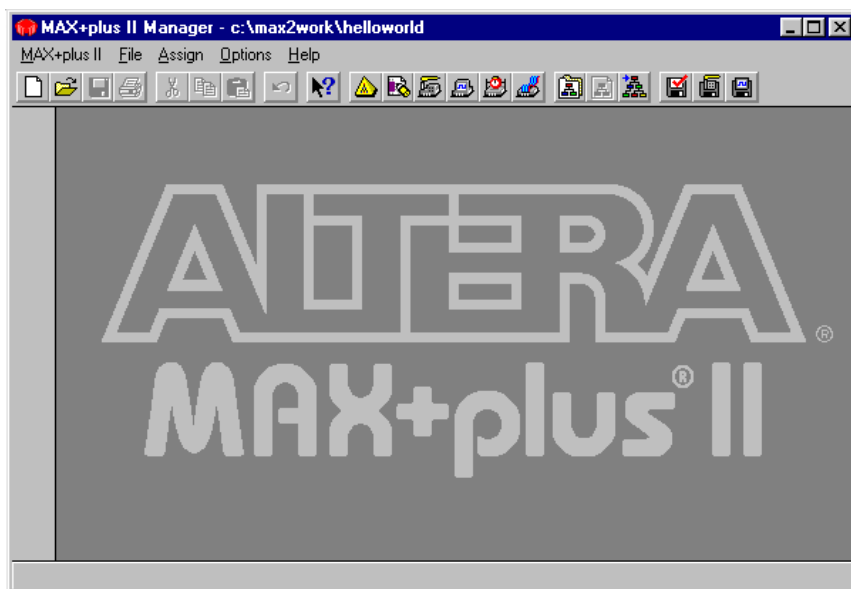


Рис. 1.1. Менеджер проекта

Нажав «мышкой» на первую команду ГМ менеджера **MAX+plus II**, в выпадающем меню можно перейти к любому редактору пакета (рис. 1.2).

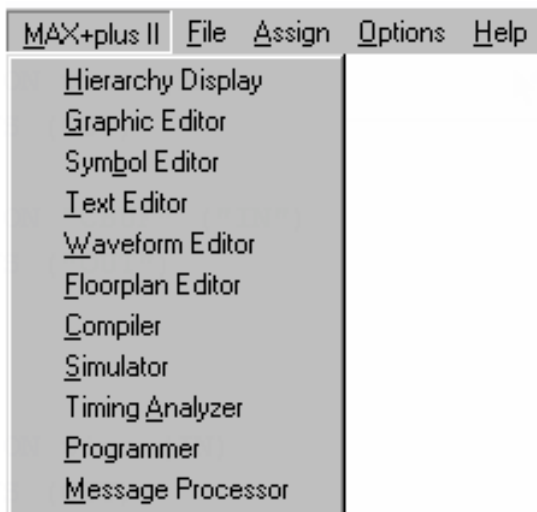


Рис. 1.1. Редакторы САПР MAX+PLUS II

1.2. Методика проектирования принципиальной схемы с помощью графического редактора Graphic Editor

1.2.1. Создание нового проекта и работа с библиотеками

После запуска пакета **MAX+PLUS II**, используя стандартные средства «MS Windows», создадим в каталоге **AL-TERA\MAX2WORK** рабочий каталог под именем **Student file**, а затем в нем — *личный подкаталог*, например **Student 1**.

В качестве примера создадим проект мажоритарной цифровой схемы, представленной на рис.1.3. Принцип работы мажоритарного элемента эквивалентен выборам, когда избирается кандидат, набравший большинство голосов избирателей. В данной схеме на выход **Y** проходит сигнал (например, 1 или 0), поступивший на наибольшее количество входов (например, на A и B или на A и C или на B и C). Поэтому любой сигнал на третьем входе (соответственно C или B или A) на выход **Y** не проходит.

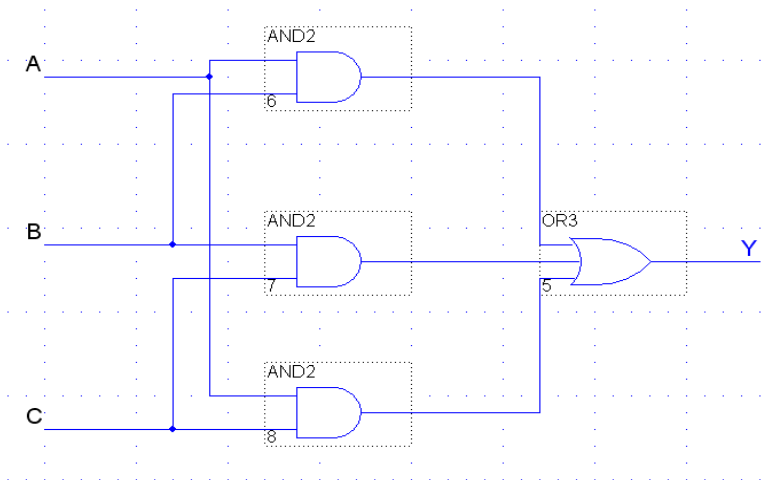


Рис. 1.2. Принципиальная схема мажоритарного элемента

Мажоритарный элемент широко используется при резервировании электронной аппаратуры. В нашем случае можно, например, в счетчике импульсов осуществить трёхкратное резервирование, а выходы трех счетчиков подключить к входам А, В и С мажоритарного элемента. Тогда, в случае сбоя или отказа одного из счетчиков (например, подключенного к входу А), на выход Y мажоритарного элемента будет поступать только информация, хранящаяся в двух других счетчиках (подключенных к входам В и С).

Для создания нового проекта в менеджере системы создадим новый графический файл принципиальной схемы (команда главного меню **File/ New** или активация мышкой крайнего левого значка панели инструментов – белый листок с загнутым уголком). В открывшемся диалоговом окне «New» выбираем пункт «**Graphic Editor file**» и нажимаем кнопку «**OK**» (рис. 1.4), при этом автоматически открывается окно графического редактора (рис. 1.5).

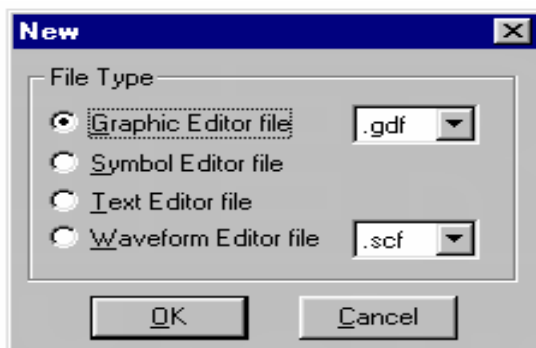


Рис.1.3. Создание и выбор графического файла

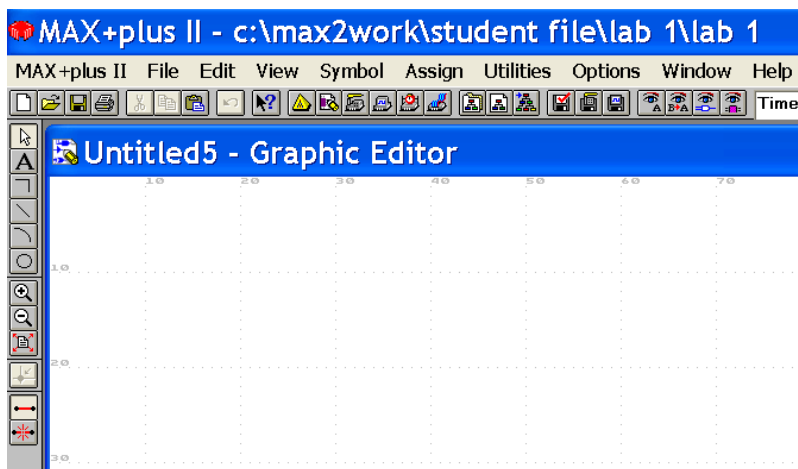


Рис. 1.4. Графический редактор Graphic Editor

Окно редактора имеет ряд дополнительных пунктов основного меню и панель инструментов редактора, расположенную вертикально с левой стороны окна. Часть инструментов редактора, расположенных на вертикальной панели с левой стороны окна, предназначена для создания поясняющих надписей, рисунков и таблиц, которые не являются частью проекта и не обрабатываются компилятором.

Сохраним новый файл проекта (через меню «**File / Save As**») под именем **Lab 1** (расширение **.gdf** новому файлу будет присвоено автоматически) в созданном личном подкаталоге \ **Student 1** (рис 1.6).

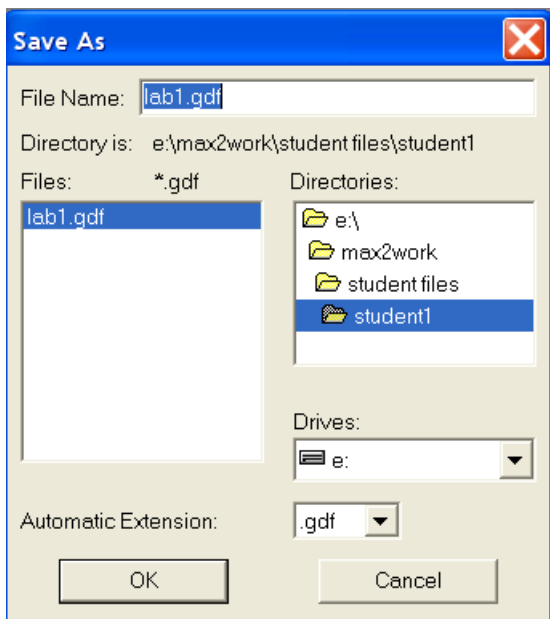


Рис. 1.5. Сохранение файла проекта

Имя файла проекта следует обязательно привязать к имени проекта – это делается путем выбора команды главного меню **File/ Project / Set Project to Current File** (рис. 1.7).

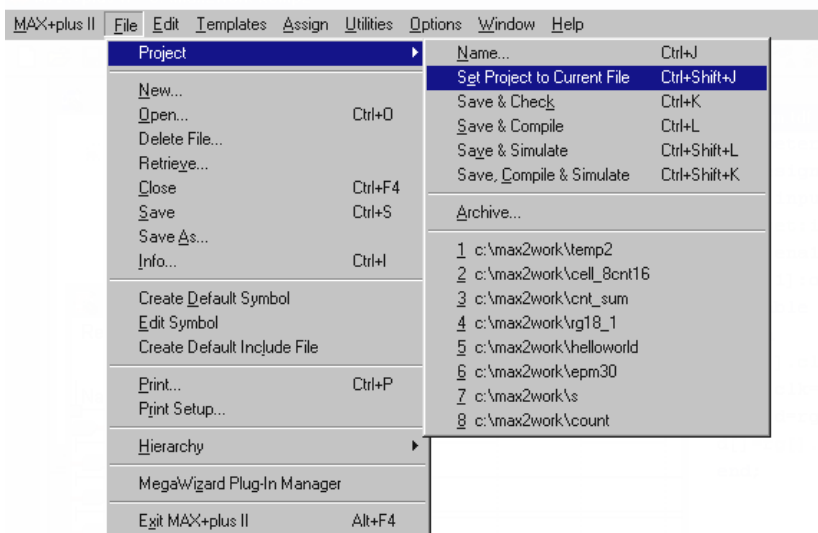


Рис. 1.6. Привязка принципиальной схемы к проекту

При этом проекту присваивается имя графического файла. Таким образом, мы создали новый проект **Lab 1**, в котором будут содержаться все файлы и документы создаваемой ПЛИС.

Существует и другой способ задания имени проекта: с помощью команды главного меню **File/ Project/ Name** введем в появившемся на дисплее окне **Project Name** имя проекта, а в окне **Files** имя графического файла и подтвердим **OK** (рис. 1.8).

Для создания графического проекта можно использовать библиотеки примитивов (**\MAXPLUS\MAX2LIB\PRIM**), макрофункций (**\MAXPLUS\MAX2LIB\MF**) и параметризованных мегафункций (**\MAXPLUS\MAX2LIB\MEGA_LPM**). Примитивы включают большой набор основных логических элементов, триггеров, элементов входа и выхода (**INPUT, OUTPUT, BIDIR**), а также вспомогательные элементы: **GND** (логический ноль или земля), **VCC** (логическая единица или питание ПЛИС).

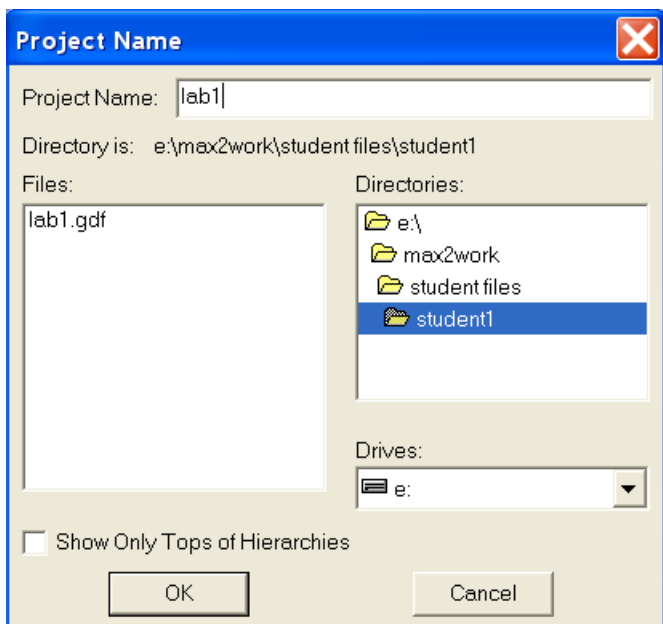


Рис. 1.7. Задание имени проекта и графического файла

Макрофункции включают главным образом эквиваленты логических микросхем 74-й серии. Параметризованные мегафункции позволяют реализовывать многовходовые и многоразрядные элементы цифровой схемотехники (логику, регистры, мультиплексоры и т.д.), вводя ряд параметров в специально обозначенных областях условных графических обозначений этих элементов. Порядок использования мегафункций можно найти в соответствующем разделе электронного справочника системы («**Help**»).

1.2.2. Изображение и ввод электронных компонентов схемы, назначение имен входных и выходных контактов ПЛИС

Проектирование цифровой схемы мажоритарного элемента в графическом редакторе начнем с размещения на схеме изображений электронных компонентов с помощью диалогового окна «**Enter Symbol**», которое можно открыть через меню «**Symbol**» основного меню редактора (команда ГМ **Symbol / Enter Symbol**).

Однако удобнее вызывать это окно двойным щелчком левой клавиши «мыши» по свободному пространству открытого окна редактора, после чего выбранный элемент будет размещён именно в этом месте (позиционирование элемента производится по верхнему левому углу условного обозначения элемента). Выбрать и установить необходимый элемент можно двумя способами:

- Набрать имя элемента схемы совпадений типа **2-И (and2)** в окне «**Symbol Name**» диалогового окна «**Enter Symbol**» и нажать «**OK**» (рис. 1.9).

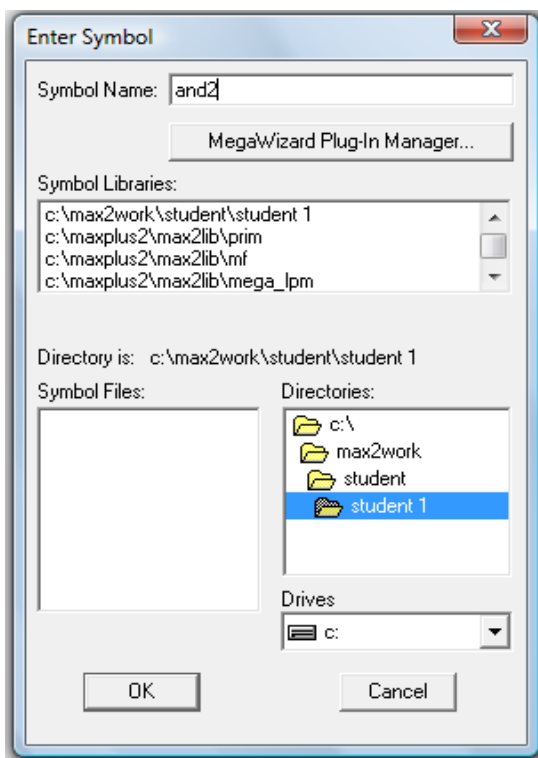


Рис. 1.8. Ввод или выбор электронного компонента *and2*

- Выбрать необходимую библиотеку **altera2/max2lib/prim** в окне **Symbol Libraries** диалогового окна **Enter Symbol** и двойным щелчком левой клавиши «мыши» открыть её. За-

тем аналогичным образом выбрать необходимый элемент **and2** в окне **Symbol File**.

Чтобы ввести в схему еще два элемента **and2**, можно повторить описанную выше процедуру или использовать команды копирования «**Copy**» и вставки «**Paste**». Для этого выделим необходимый символ электронного компонента (ЭК), щелкнув по нему левой кнопкой «мыши», который при этом помещается в красный прямоугольник. Затем необходимо щелкнуть правой кнопкой по символу **and2** и во всплывающем меню (рис. 1.10) выбрать команду «**Copy**». Также можно нажать на иконку «**Copy**» в верхней строке панели инструментов программы или активировать команду ГМ «**Edit/Copy**».

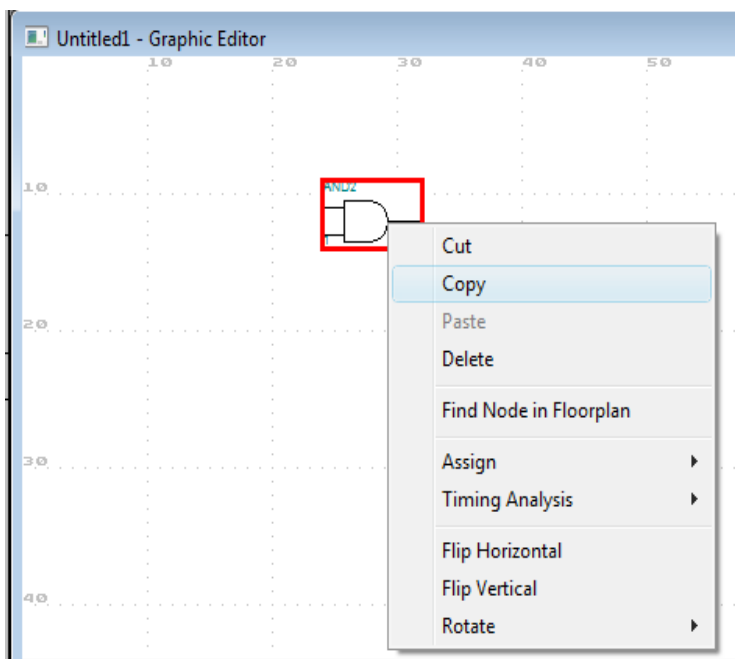


Рис. 1.9. Процедура копирования электронного компонента

Разместить ЭК на схеме можно, нажав курсором в необходимом месте, затем щелкнуть правой кнопкой «мыши» в указанной точке и в выпадающем меню выбрать команду «Вставить» («**Paste**»). Эту операцию можно осуществить, нажав на иконку «**Paste**» в верхней

строке панели инструментов или с помощью команды ГМ «**Edit/Paste**» (рис. 1.11).

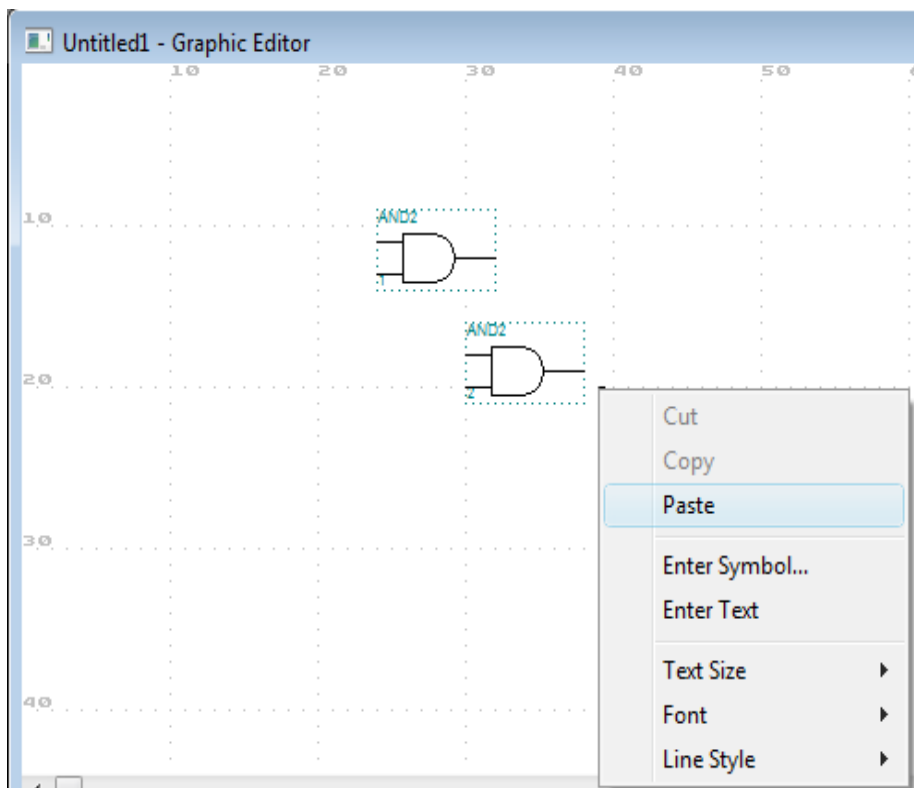


Рис. 1.10. Процедура размещения скопированного электронного компонента

Таким же образом разместим на схеме логический элемент **3-ИЛИ (OR3)**. В результате получим схему, представленную на рис. 1.12.

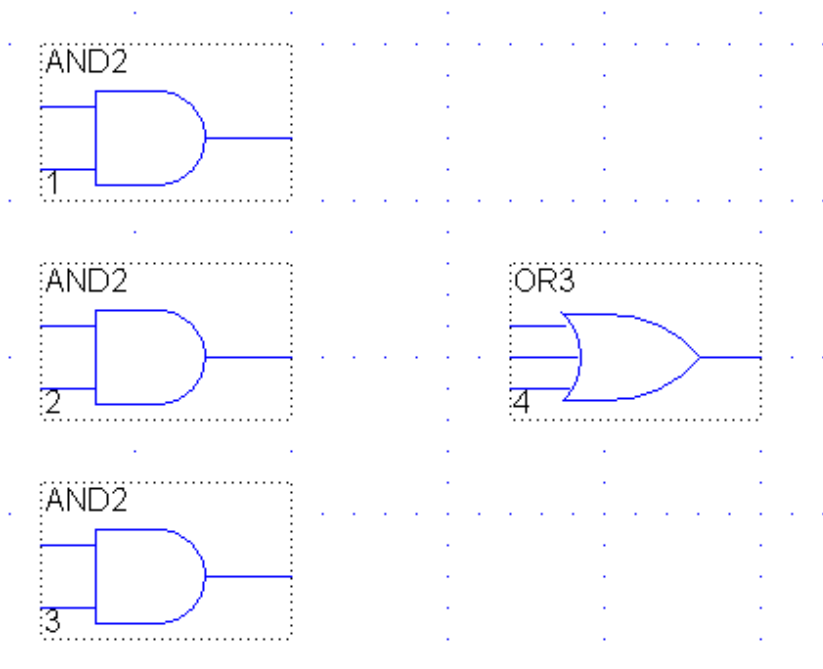


Рис. 1.11. Размещение ЭК мажоритарного элемента

Реализация электронного устройства на ПЛИС, **в отличие от традиционной конструкции, собранной на транзисторах, а также на ИМС малого и среднего уровня интеграции и расположенных как обычно на печатной плате**, позволяет ограничить цифровую схему **со стороны входа и выхода** входными и выходными контактами, которые подключены к выводам (ножкам) корпуса ПЛИС. Поэтому в любую проектируемую схему, реализованную на ПЛИС, необходимо вводить **входные и выходные** (иногда только выходные) **контакты INPUT и OUTPUT** для подачи на ПЛИС и съема с нее электрических сигналов. Для этого в библиотеке **max2lib\prim** выберем символы входа **INPUT** и выхода **OUTPUT** и разместим их на схеме (рис. 1.13).

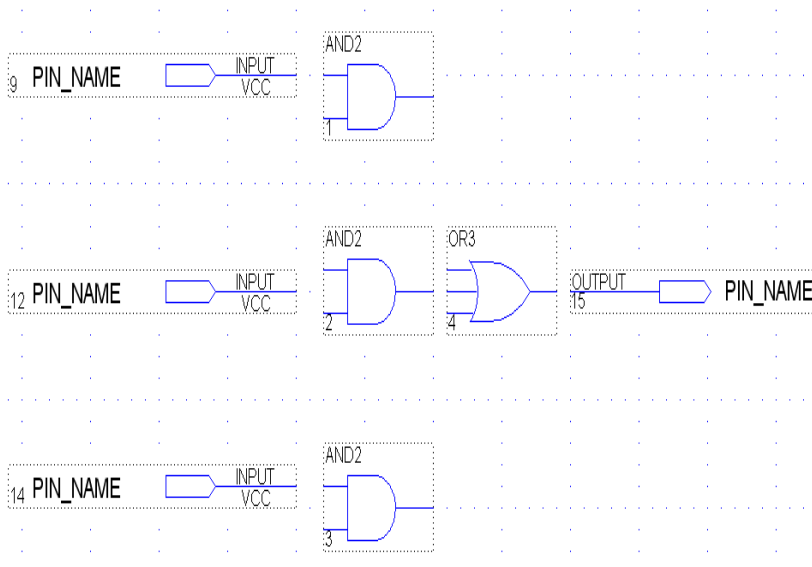


Рис. 1.12. Размещение входных INPUT и выходных OUTPUT контактов ПЛИС

Размещенным на схеме контактам входов и выходов ПЛИС необходимо присвоить имена для компиляции проекта после разработки всей схемы. Каждому контакту ввода-вывода ПЛИС должно быть присвоено своё имя. Для этого необходимо дважды щёлкнуть «мышью» на поле **PIN_NAME** контактов входа-выхода и в появившемся чёрном прямоугольнике набрать новое имя и нажать клавишу **ENTER**. Переход к набору имени следующего элемента ввода может осуществляться по нажатию клавиши **ENTER** (рис. 1.14).

Для редактирования имен входных и выходных контактов необходимо дважды щелкнуть по имени левой кнопкой «мыши» и, удерживая нажатой левую кнопку, выделить необходимую часть имени контакта с последующей коррекцией текста.

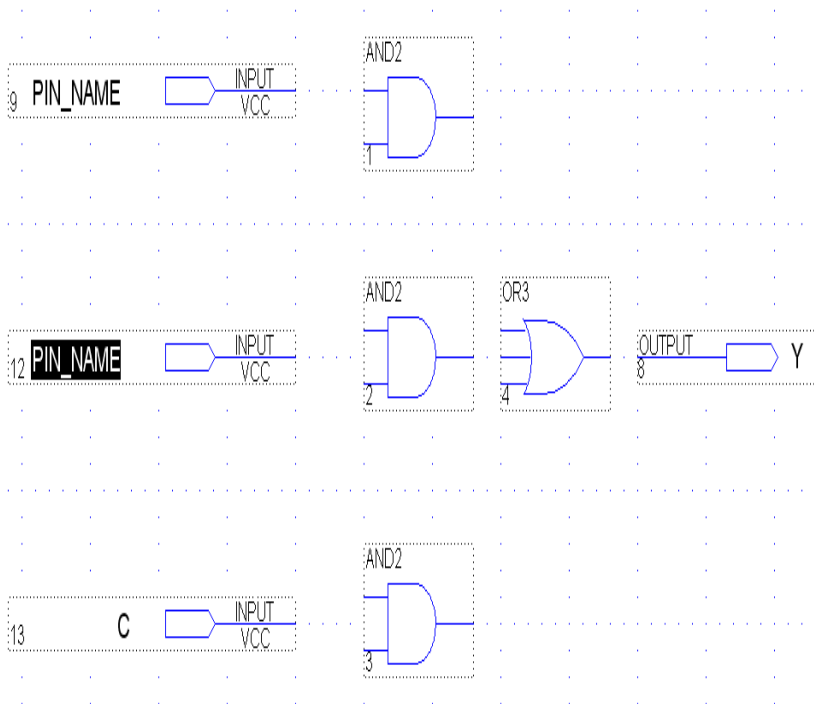


Рис. 1.13. Именование входных INPUT и выходных OUTPUT контактов ПЛИС

1.2.3. Выполнение электрических соединений и шин

Выводы символов элементов можно соединить проводниками (сигнальными линиями) или присваиваемыми одинаковыми именами. Символы соединяются (рис. 1.15) с помощью линий (тип линии задается командой ГМ Options /Line Style) или по заданию имени цепи (для шины — имя шины с указанием числа проводников Bus[7..0]).

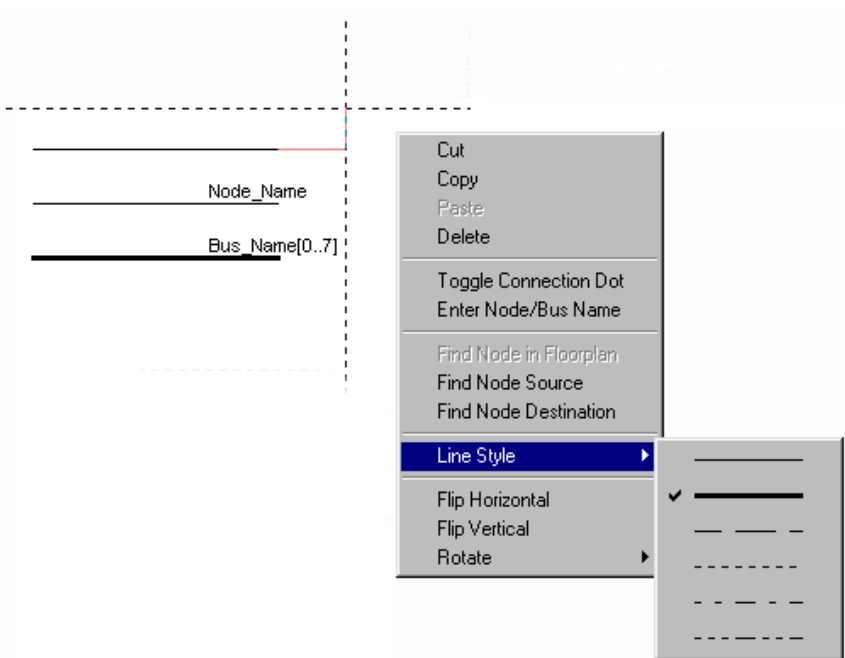


Рис. 1.14. Задание типа линии командой Options /Line Style

Для проведения электрической цепи (проводника) необходимо совместить указатель курсора (конец стрелки) с выводом элемента, при этом курсор автоматически превращается в инструмент рисования ортогональных линий (перекрестие или пунктирный крест на всей схеме). После этого проводится необходимая линия до вывода другого ЭК (при нажатой и удерживаемой левой кнопке мыши). За один приём можно провести два ортогональных отрезка. Если этого недостаточно, то процедуру можно повторять, начиная с конца проведённой линии или с вывода другого элемента (рис. 1.16).

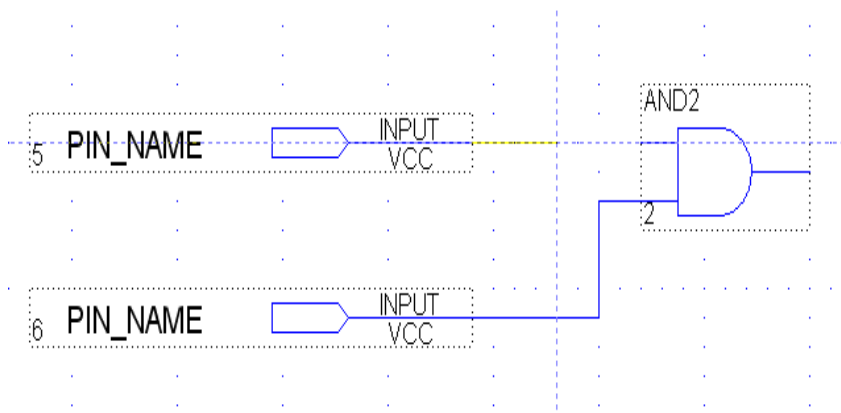


Рис. 1.15. Соединение линией связи ЭК. Перекрестие (пунктирный крест) немного не доведено до входа ЭК

Для соединения пересекающихся линий можно щелкнуть два раза левой кнопкой «мыши» по их перекрестию, а для удаления соединения — повторить эту операцию.

То же самое можно сделать с помощью «иконки» соединения пересекающихся линий, расположенной слева внизу на вертикальной инструментальной панели. Достаточно с помощью «мышки» указать курсором на пересечение линий и, нажимая на «иконку», соединять и разъединять пересекающиеся линии (рис. 1.17).

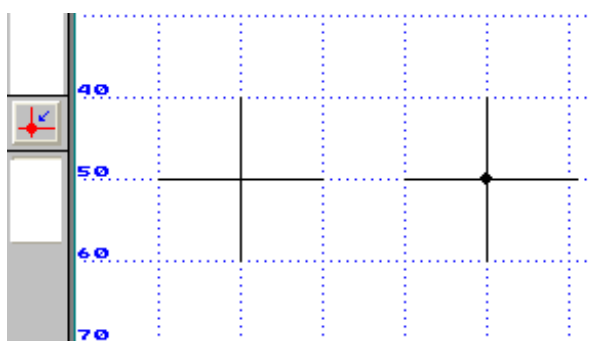


Рис. 1.16. Соединение и разъединение пересекающихся линий

Для соединения элементов необязательно проводить сплошные соединительные линии между выводами. Достаточно выводам, ко-

которые должны быть соединены между собой, присвоить одинаковые имена. **Поименованный вывод** ЭК является его электрическим проводником или цепью («Node»). Для присвоения проводнику имени необходимо сначала провести короткую сигнальную линию. Так как по окончании рисования линии она остаётся выделенной (выделенные элементы обозначаются красным цветом или красной рамкой), то имя проводника можно вводить сразу. Если линия не выделена (имеет чёрный цвет), то необходимо выделить её щелчком левой кнопки мыши в том месте, где должно начинаться имя проводника, и набрать на клавиатуре требуемое имя, например a-a или b-b (рис. 1.18).

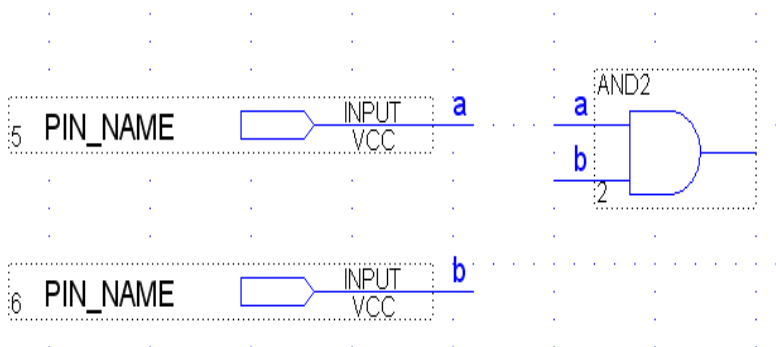


Рис. 1.17. Соединение двух ЭК путем присвоения их выводам одинаковых имён

Этот способ широко используется при соединении ЭК с помощью **шин**, изображаемых в виде толстых линий, внутри которых находится несколько проводников, осуществляющих передачу сигналов между ЭК схемы. При этом число проводников в шине может составлять от 2 до 256. Существует два варианта соединения ЭК:

- при одинаковом обозначении входов и выходов проводников шины, ей присваивается имя с указанием количества проводников и их номера – например, **B[3..0]**, что означает наличие в шине 4 проводников, входы и выходы которых обозначаются одними именами: **B3, B2, B1, B0** (рис. 1.19,a);

- при разном обозначении входов и выходов проводников шины, ей присваивается имя с указанием количества проводников, а также номера их входов и выходов в двух квадратных скобках, например **B[3..0][7..4]**, что означает наличие в шине проводников, входы которых именуются как **B3, B2, B1, B0** (первая скобка), а имена их выходов (**B7, B6, B5, B4**) размещаются во второй скобке (рис.1.19,б).

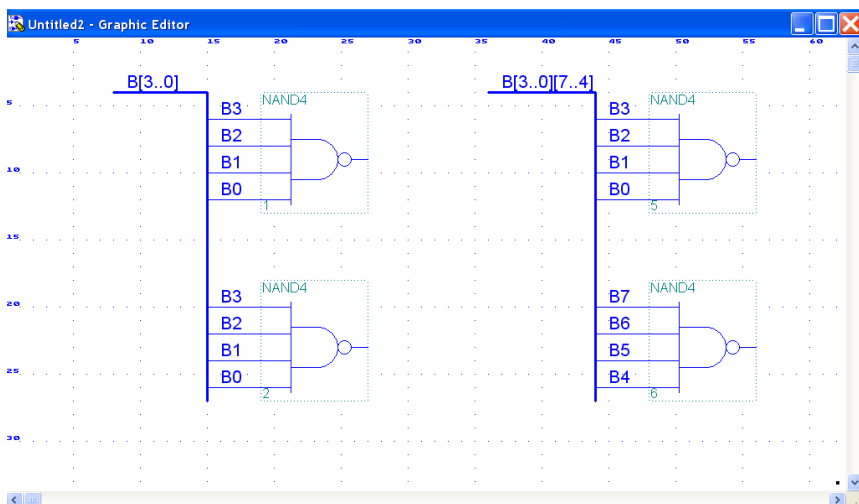


Рис. 1.18. Соединение ЭК с помощью шин:

а — с одинаковыми именами входных и выходных цепей;

б — с разными именами входных и выходных цепей

Таким образом, во втором случае шина состоит из проводников, входы и выходы которых обозначаются как B3-7, B2-6, B1-5, B0-4.

Установленные элементы и группы элементов (любую выделенную область окна) можно удалять, перемещать, поворачивать, копировать и размножать в пределах окна редактора, а также переносить в другие окна через буфер обмена. Все эти операции можно выполнять с использованием пунктов меню «**Edit**» (в основном или главном меню редактора), предварительно выделив требуемую область или элемент.

Выделение отдельных элементов, включая и соединительные линии, производится щелчком левой кнопки «мыши» на изображе-

нии символа элемента. Выделение области возможно при нажатой и удерживаемой левой кнопке. Выделенная область обозначается рамкой чёрного цвета, а все выделенные элементы – красным цветом.

Для удаления, копирования, вырезания, поворота и некоторых других операций с элементами можно использовать выпадающее контекстное меню, которое вызывается щелчком правой кнопки «мыши» по выбранному элементу. Для перемещения выделенных областей в пределах окна используется способ «**Drag&Drop**», а для размножения выделенных областей и элементов – «**Drag&Drop**» при нажатой и удерживаемой клавише **CTRL**.

На основе перечисленных способов соединения между собой ЭК мажоритарного элемента нарисует окончательно его схему, представленную на рис. 1.20.

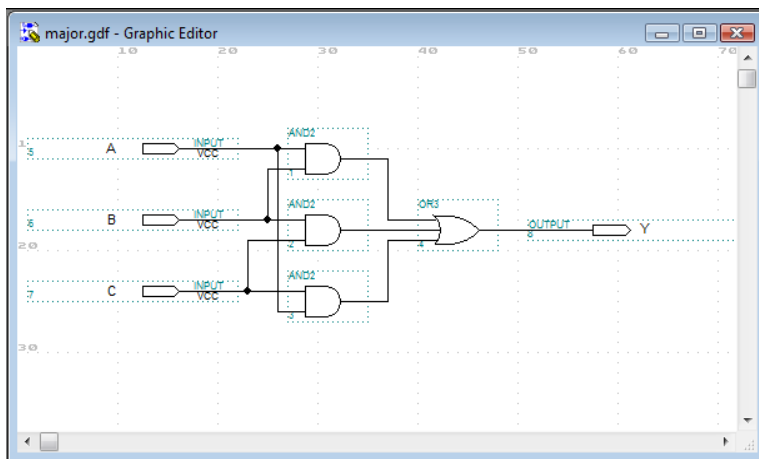


Рис. 1.19. Принципиальная схема мажоритарного элемента

1.2.4. Проверка ошибок и сохранение файла схемы

После создания схемы необходимо сохранить и проверить файл схемы на наличие ошибок с помощью команды **File/Project /Save & Check** (рис. 1.21). По этой команде файл сохраняется и открывается окно компилятора **Compiler** пакета **MAX+plus II**, модуль (программа) которого **Compiler Netlist Extractor** (выделение или

извлечение списка проводов схемы) проверяет схему на ошибки и выводит сообщение об их числе и предупреждения.

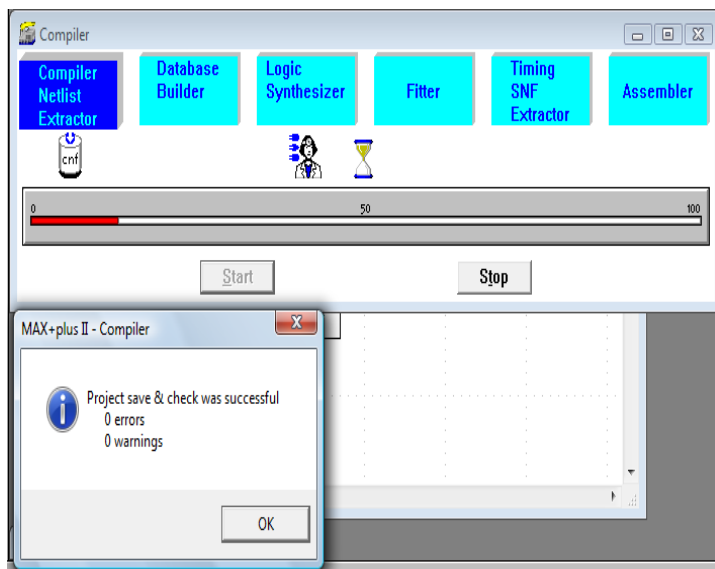


Рис. 1.20. Сохранение файла схемы и проверка ошибок

В случае отсутствия ошибок нажимаем в окне сообщений левой кнопкой мыши на **ОК**, а затем выходим из компилятора (активируем в правом углу иконку «крест»).

При появлении ошибок необходимо выделить «мышью» сообщение и активировать в окошке кнопку **«Locate»** для нахождения источника ошибки или нажать на кнопку **«Help on message»** для получения справки.

1.2.5. Компиляция файла схемы (организация базы данных) с помощью редактора **Compiler** и выбор типа ПЛИС

Компилятор **Compiler** пакета MAX + PLUS II состоит из нескольких модулей и утилит, которые проверяют цифровую схему на наличие ошибок, синтезируют ее логику и размещают в одном

или нескольких корпусах ПЛИС, генерируют выходные двоичные файлы для симулятора, временного анализатора и программатора (рис. 1.22).

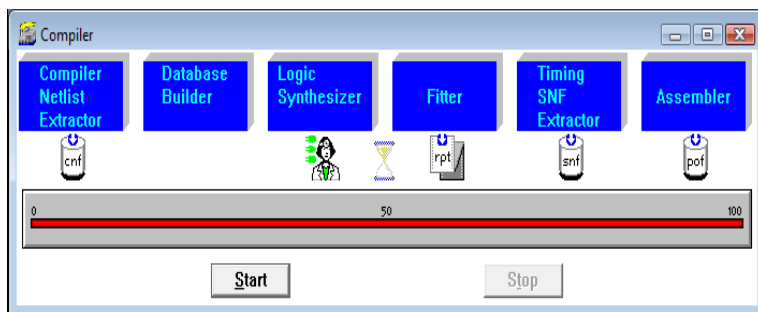


Рис. 1.21. Компилятор *Compiler* пакета *MAX + PLUS II*

Компилятор создает базу данных для моделирования, временного анализа и программирования (записи в ПЛИС) разработанной цифровой схемы. Запуск компилятора можно осуществить из любого приложения пакета *MAX + PLUS II* или с помощью команды вызова окна компилятора в основном меню по команде ГМ **MAX + PLUS II/Compiler** (рис. 1.23).

Процесс компиляции схемы состоит из:

- указания семейства и типа ПЛИС с помощью команды ГМ (рис. 1.24) компилятора **Assign /Device** (в открывшемся диалоговом окне (рис. 1.25) выбрать необходимое семейство и тип ПЛИС);
- выбора режима «интеллектуальная компиляция» или ускоренной работы компилятора с помощью команды ГМ **Processing/Smart Recompile**;

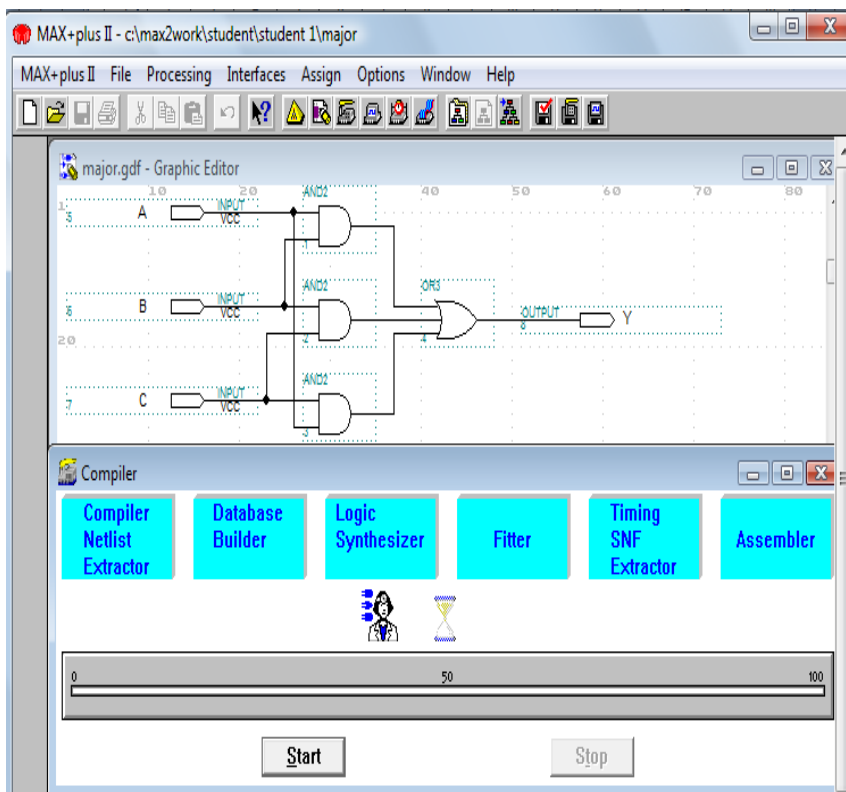


Рис. 1.22. Запуск компилятора в пакете MAX + PLUS II

- подключения утилиты проверки схемы с помощью команды ГМ **Processing /Design Doctor** и настройки параметров ПЛИС – с помощью команды ГМ **Processing/ Design Doctor Settings**, в диалоговом окне настроек которой указывается класс ПЛИС (ОЗУ-FLEX или ПЗУ-EPLD, см. рис. 1.26);
- установки, в случае необходимости, бита секретности (рис. 1.27) для ПЛИС в диалоговом окне **MAX Global Project Device Options**, вызываемого командой ГМ **Assign/Global Project Device Options**;

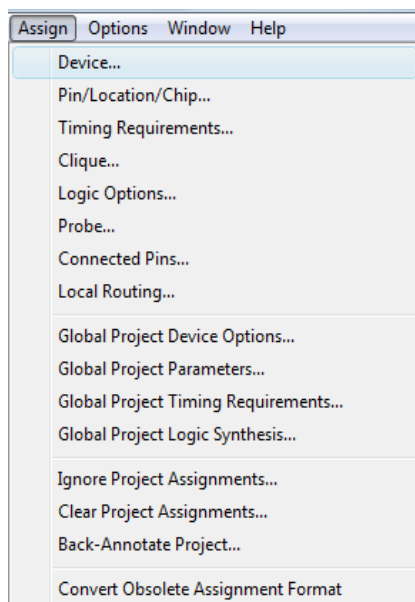


Рис. 1.23. Привязка корпуса ПЛИС к схеме

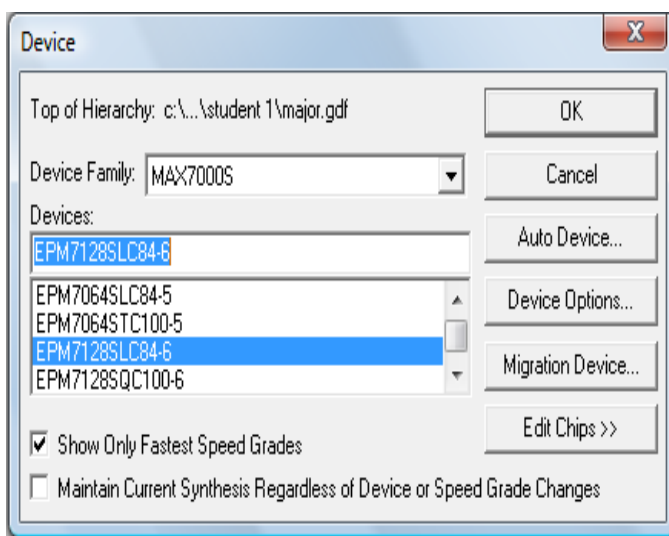


Рис. 1.24. Выбор семейства и типа ПЛИС

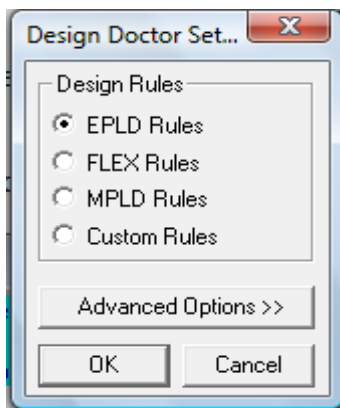


Рис. 1.25. Указание класса ПЛИС

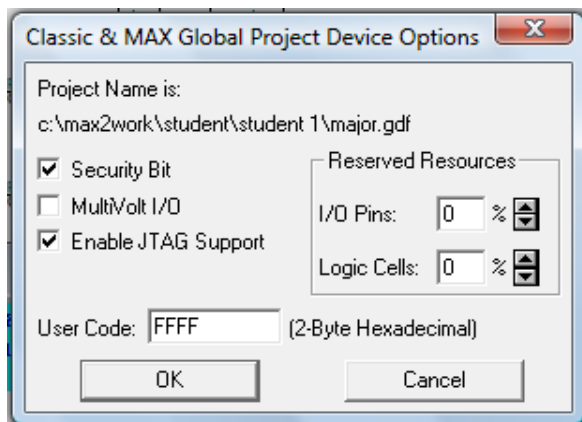


Рис. 1.26. Установка бита защиты ПЛИС

- установки стиля (обычно **NORMAL**) логического синтеза в диалоговом окне **Project Logic Synthesis**, вызываемого командой ГМ **Assign/Global Project Logic Synthesis**;
- создания файла установки и выбора связей схемы для **моделирования** ее работы с помощью команды ГМ **Processing/Timing SNF Extractor**;
- установки разделов файла отчета о результатах работы компилятора с помощью команды ГМ **Processing/Report File Settings**.

После выполнения указанных процедур необходимо запустить компиляцию для обработки разработанной схемы, нажав «мышью» на кнопку **Start**, в результате чего компилятор выводит на экран дисплея четыре сообщения, в которых выводится информация о схеме и выбранным типе ПЛИС (рис. 1.28).

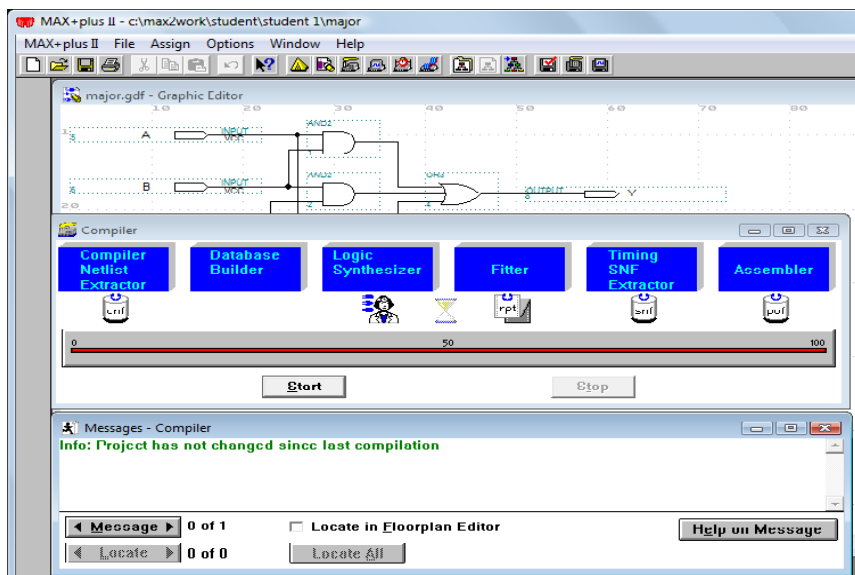
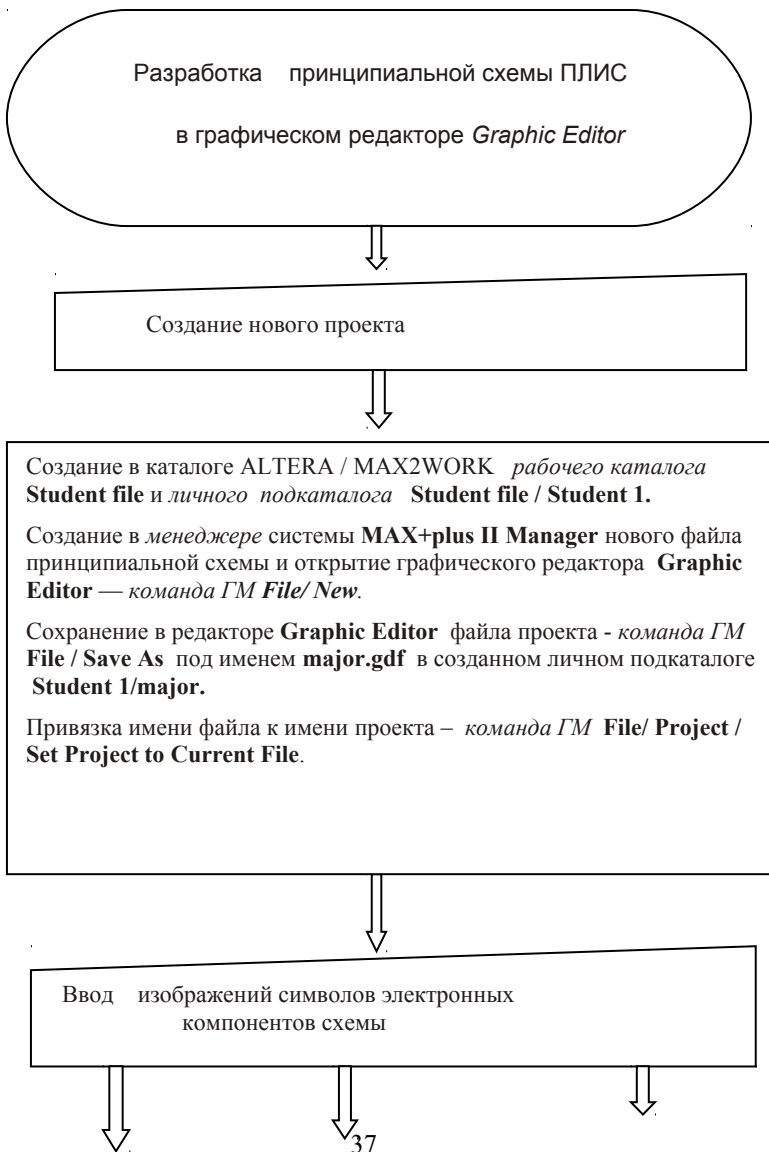
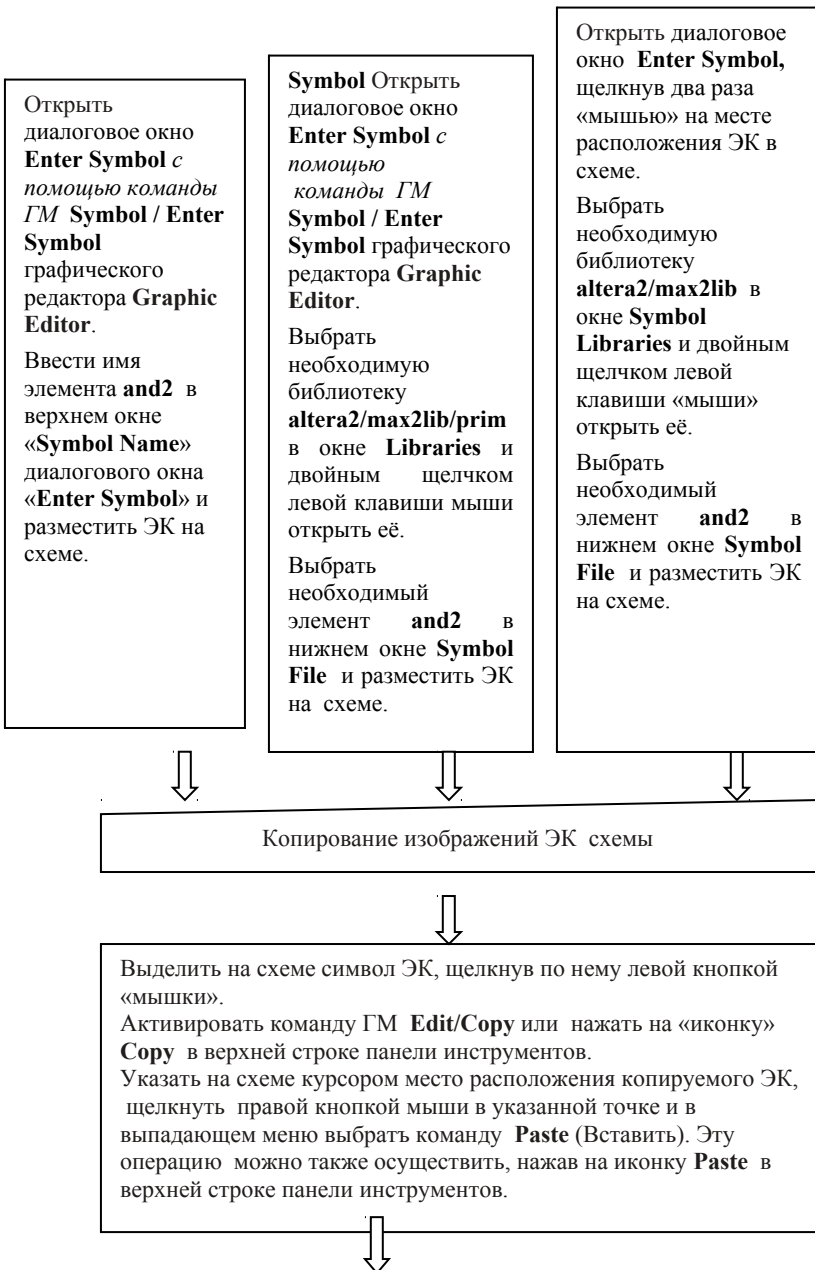


Рис. 1.27. Результаты компиляции цифровой схемы

1.3. Структурная схема алгоритма разработки цифровой схемы

На рис. 1.28 представлена структурная схема алгоритма основных этапов проектирования принципиальной схемы.





Ввод и именование входных и выходных контактов
ПЛИС



Выбрать символы входа **INPUT** и выхода **OUTPUT** в библиотеке **max2lib/prim** и разместить их на схеме.

Присвоить имена контактам ввода-вывода ПЛИС, дважды щёлкнув левой кнопкой мышки на их поле **PIN_NAME** и введя имя контакта.



Выполнение и размещение электрических соединений



Задать тип линии (линия связи или шина) командой ГМ **Options /Line Style**.

Для проведения электрической цепи соединить курсором выводы ЭК схемы.

Для соединения или размыкания пересекающихся линий щёлкнуть два раза левой кнопкой «мышки» по их перекрестию.



Выполнение и размещение шин



- 1) при одинаковом обозначении входов и выходов проводников шины, ей присваивается имя с указанием количества проводников и их номера **B[3..0]**, входы и выходы которых обозначаются одними именами - **B3, B2, B1, B0**.
- 2) при разном обозначении входов и выходов проводников шины ей присваивается имя с указанием количества проводников, а также номера их входов и выходов в двух квадратных скобках - **B[3..0][7..4]**, входы которых именуются как **B3, B2, B1, B0** (первая скобка), а имена их выходов (**B7, B6, B5, B4**) размещаются во второй скобке.

Проверка ошибок и сохранение файла схемы

Сохранить и проверить файл схемы на наличие ошибок с помощью команды **File/Project /Save & Check**.
Нажать **OK** в случае отсутствия ошибок.
При появлении ошибок, выделить «мышкой» сообщение и активировать в окошке кнопку «**Locate**» для нахождения источника ошибки или нажать на кнопку «**Help on message**» для получения справки.

Компиляция файла схемы (организация базы данных) и выбор типа ПЛИС

Запуск компилятора Compiler командой ГМ MAX + PLUS II/Compiler.
Указание семейства и типа ПЛИС с помощью команды ГМ компилятора Assign /Device.

выбор режима «интеллектуальная компиляция» или ускоренной работы компилятора с помощью команды ГМ Processing /Smart Recompile.

подключение утилиты проверки схемы с помощью команды ГМ Processing /Design Doctor и настройки параметров ПЛИС – с помощью команды ГМ Processing/ Design Doctor Settings, в диалоговом окне настроек которой указывается класс ПЛИС (ОЗУ-FLEX или ПЗУ-EPLD).

Установка, в случае необходимости, бита секретности для ПЛИС в диалоговом окне MAX Global Project Device Options, вызываемого командой ГМ Assign/Global Project Device Options.

Создание файла установки и выбора связей схемы для моделирования ее работы с помощью команды ГМ Processing /Timing SNF Extractor.

Установка разделов файла отчета о результатах работы компилятора с помощью команды ГМ Processing /Report File Settings.

Запуск процесса компиляции нажатием кнопки Start в окне компилятора Compiler.

Активировать кнопку Message для вывода информации об возможных ошибках и замечаниях по созданной базе проекта схемы.

Исправить ошибки и замечания в схеме.



Конец работы и выход из пакета
MAX+PLUS II

Рис. 1.28. Структурная схема алгоритма проектирования принципиальной схемы

Подготовка к работе

- Установить пакет **MAX+PLUS II** на собственном персональном компьютере или освоить работу с пакетом в лаборатории кафедры.
- Ознакомиться с графическим редактором **Graphic Editor** пакета **MAX+PLUS II**, изучив соответствующие разделы в рекомендуемой литературе и лекциях.
- Ознакомиться с особенностями работы компилятора **Compiler** пакета **MAX+plus II**.

Рабочее задание

- Выполнить под руководством преподавателя проектирование учебной схемы с помощью графического редактора **Graphic Editor**.
- Получить от преподавателя или выбрать из справочника принципиальную схему цифрового устройства.
- Разработать самостоятельно цифровую схему с помощью графического редактора **Graphic Editor**.
- Произвести компиляцию цифровой схемы с помощью компилятора **Compiler**, показать результаты проектирования преподавателю и сохранить файлы проекта.
- Оформить отчет в печатной форме с результатами проектирования принципиальной схемы.

Контрольные вопросы

1. Способы реализации специализированных СБИС. Назовите их достоинства и недостатки.
2. Какова структурная организация ПЛИС?
3. Перечислите основные характеристики современных ПЛИС.
4. Алгоритм проектирования специализированных СБИС на основе ПЛИС с использованием системы MAX+plus II.
5. Назначение и особенности графического редактора (Graphic Editor).

6. Назначение и особенности символьного редактора (Symbol Editor).
7. Назначение и особенности текстового редактора (Text Editor).
8. Назначение и особенности сигнального редактора (Waveform Editor).
9. Назначение и особенности редактора базового плана (Floorplan Editor).
10. Подготовка и порядок проведения компиляции проекта.
11. Моделирование проекта в системе MAX+plus II.
12. Создание нового проекта и работа с библиотеками.
13. Ввод электронных компонентов схемы.
14. Изображение электронных компонентов схемы.
15. Ввод имен входных и выходных контактов ПЛИС.
16. Назначение имен внутренних контактов ПЛИС.
17. Выполнение электрических соединений.
18. Изображение шин на схеме.
19. Проверка ошибок схемы.
20. Сохранение файла схемы.
21. Компиляция файла схемы (организация базы данных).
22. Выбор типа ПЛИС.
23. Структурная схема алгоритма разработки цифровой схемы.
24. Как установить на персональном компьютере пакет MAX+plus II ?

Работа 2

Моделирование работы программируемых логических интегральных схем с помощью редакторов Waveform Editor и Simulator

Цель — моделирование работы цифрового устройства путем виртуального подключения входных сигналов к модели созданной схемы и проверке ее работы с выводом полученных результатов на дисплей для их анализа и коррекции.

Вместо таких традиционных этапов, как изготовление печатной платы с последующей распайкой на ней электронных компонентов (которые нужно где-то достать или купить), организацией рабочего места с необходимой измерительной аппаратурой и привлечением лаборанта (радиомонтажника) для отладки цифрового устройства, предлагается виртуальная IT-технология моделирования и отладки работы цифрового устройства.

Методика моделирования работы цифровых устройств осуществляется на базе пакета **MAX+PLUS II** с помощью редакторов **Waveform Editor** и **Simulator**.

2.1. Методика моделирования цифровых схем с помощью редакторов Waveform Editor и Simulator

2.1.1. Загрузка проекта схемы, включение сигнального редактора Waveform Editor и создание файла входных сигналов (Simulator Channel File)

Для проверки работоспособности спроектированной в предыдущей лабораторной работе схемы сформируем входные сигналы (аналоги генераторов импульсов) с помощью сигнального редактора **Waveform Editor** в виде файла с расширением **Simulator Channel File (.scf)**. При этом сигнальный редактор является:

- инструментом для разработки проекта;
- устройством формирования и подключения моделей импульсных сигналов к модели ПЛИС для наблюдения результатов тестирования разработанной модели схемы в виде осциллограмм.

Для этого откроем сигнальный редактор с помощью команды **MAX+plus II / Waveform Editor** главного меню (ГМ) менеджера пакета (рис. 2.1).

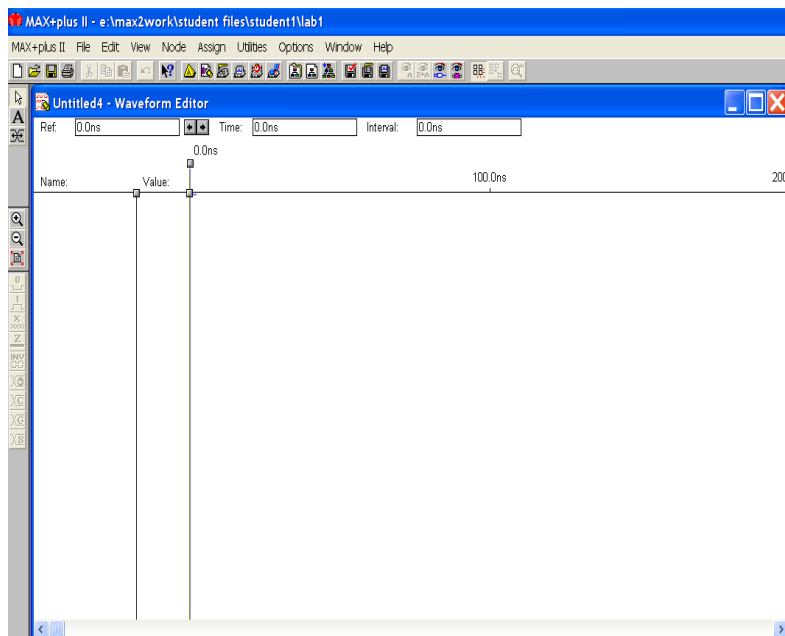


Рис. 2.1. Сигнальный редактор Waveform Editor

Однако перед этим необходимо не забыть о загрузке проекта разработанной в первой лабораторной работе схемы **Lab1** совместно с базой данных для моделирования ее работы и загрузки в ПЛИС. Подключение проекта осуществляется из менеджера пакета **MAX+plus II** с помощью команды ГМ **File / Project / Name**. В открывшемся окошке из директории **ALTERA \ MAX2WORK \ Student file \ Student 1 ** необходимо ввести имя проекта **Lab1.gdf** и нажать на клавишу **OK** (рис. 2.2).

Для создания и подключения к проекту файла входных сигналов с расширением **Lab1.scf** в главном меню менеджера выбираем команду **File / New**, а затем в появившемся окошке назначаем тип файла **Waveform Editor File** и нажимаем кнопку **OK** (рис. 2.3).

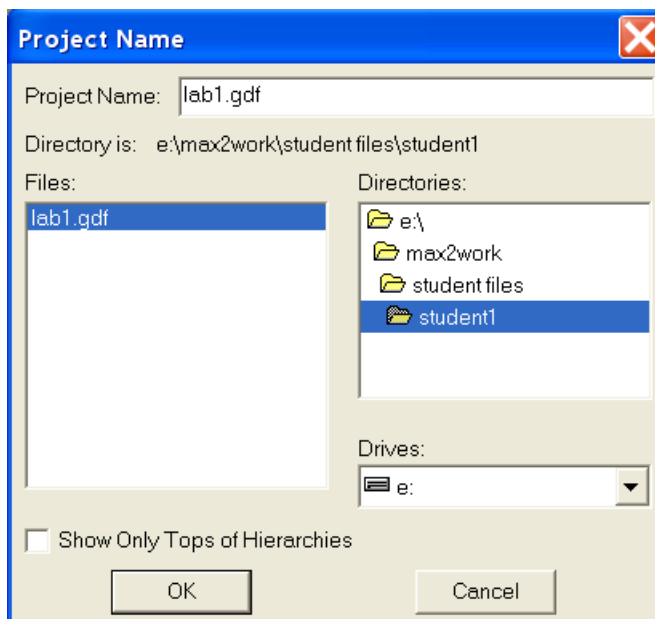


Рис. 2.2. Ввод проекта схемы Lab1.gdf

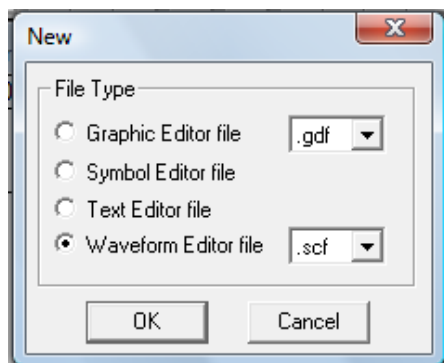


Рис. 2.3. Выбор файла входных сигналов Lab1.scf

2.1.2. Ввод имен внешних и внутренних контактов ПЛИС для подключения и контроля сигналов

Теперь необходимо ввести в сигнальный редактор имена точек схемы, к которым необходимо подключить входные и выходные сигналы. Для этого в первой колонке **Name** окна редактора **Waveform Editor** два раза щёлкнем «мышью» в поле под именем **Name** и в появившемся окне **Insert Node** (рис. 2.4) (это окно также можно вызвать командой ГМ **Node / Insert Node**) в поле **Node Name** введем имя первого входа **A**. В поле **I/O Type** можно выбрать тип входного (**Input Pin**) или выходного (**Output Pin**) сигнала, а также «скрытого сигнала» или точки внутри схемы, не подключенной к входным или выходным выводам корпуса ПЛИС (**Buried Node**). Так как **A** — это входной контакт ПЛИС, то отметим ее как **Input Pin** и нажмем **OK**. Также можно ввести названия входных и выходных точек схемы, нажав кнопку **List**, и выбрав их из списка в поле **Node & Groups from SNF** (рис. 2.5). Аналогично вводу входа **A** вводим входы **B** и **C**.

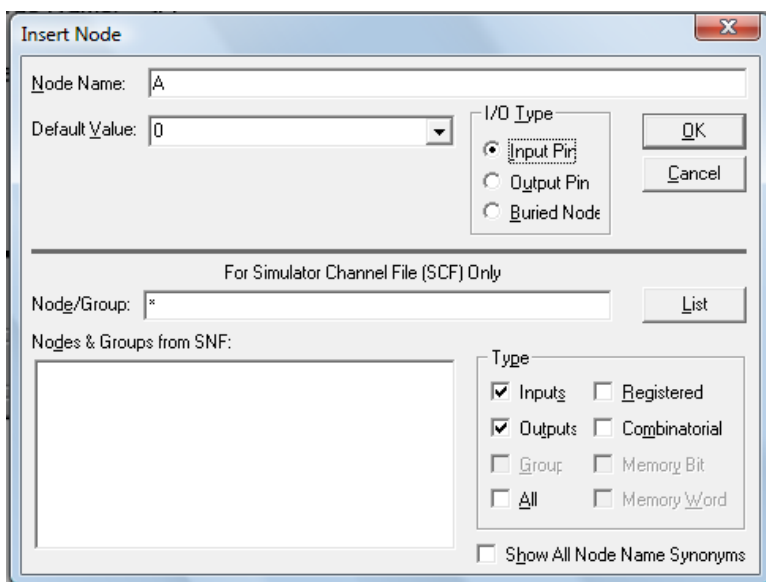


Рис. 2.4. Окно Insert Node

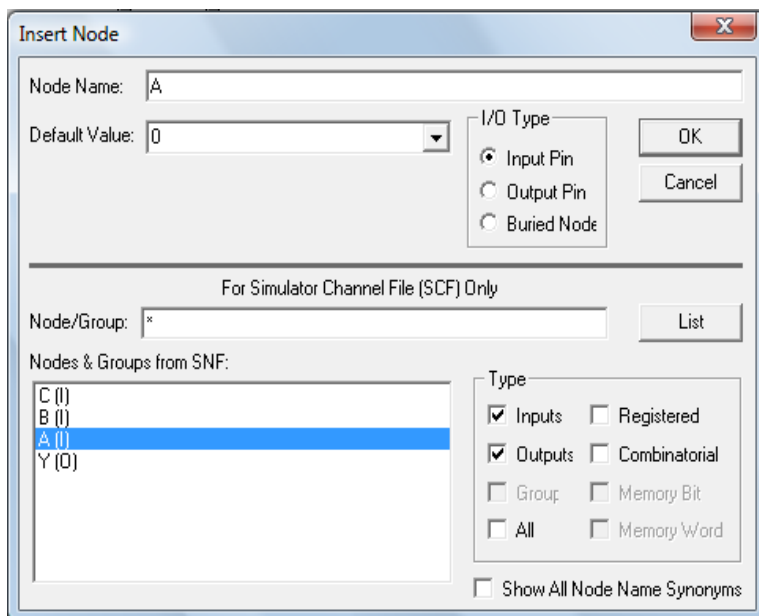


Рис. 2.5. Ввод точки подключения сигнала из списка

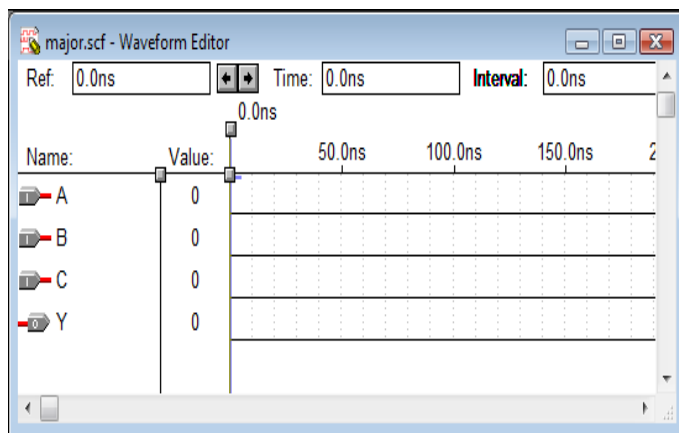


Рис. 2.6. Окно сигнального редактора с введёнными именами выводов ПЛИС

Для выхода **Y** повторяем те же действия, что и при введении входов А, В, и С, только в поле **I/O Type** указываем **Output Pin**. В результате проделанных операций на дисплее появятся имена

(**Name**) контактов схемы с нулевыми значениями (**Value**) осциллограмм (рис. 2.6).

В редакторе возможно одновременное введение всех отмеченных на схеме входных и выходных контактов схемы путем выполнения команды ГМ **Node / Enter Nodes from SNF**. В открывшемся диалоговом окне **Enter Nodes from SNF** необходимо с помощью клавиши **List** загрузить в поле окошка **Available Nodes & Groups** имена всех контактов схемы, а затем перевести их с помощью стрелки из окошка **Available Nodes & Groups** в окно **Selected Nodes & Groups** (рис. 2.7).

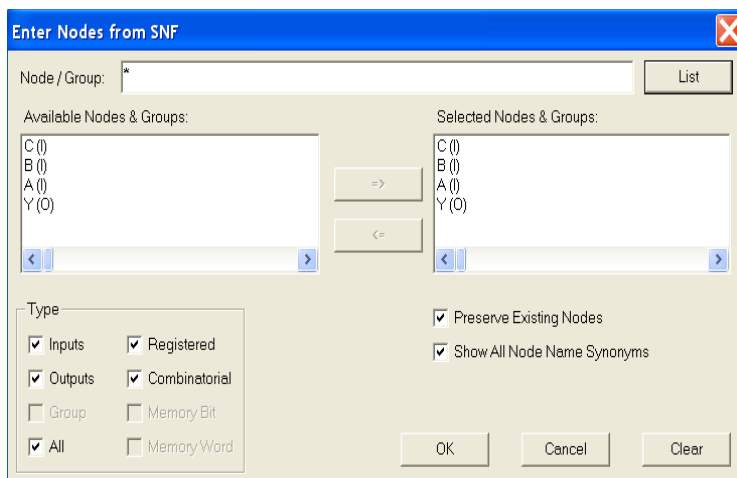


Рис. 2.7. Одновременный ввод в сигнальный редактор всех имен входных и выходных контактов схемы

После нажатия кнопки **ОК** сигнальный редактор выводит на экран имена всех контактов схемы с нулевыми осциллограммами (см. рис. 2.6).

2.1.3. Создание и редактирование входных сигналов для моделирования работы ПЛИС

Теперь необходимо подать три сигнала на входы А, В и С. Перед этим рекомендуется сделать следующие установки: в главном меню (ГМ) **File/ Endtime** выбираем величину временного интервала или конец времени моделирования, равный $1 \text{ us} = 1 \text{ мкс} = 1000 \text{ нс}$, а с помощью команды ГМ **View / Fit in Window** выводим

шкалу времени на экран. В главном меню **Options/ Show Grid/ Gridsize** устанавливаем величину единицы измерения сетки, равную $10\text{ ns} = 10\text{ нс}$. Теперь мы должны ввести осциллограммы входных сигналов, что можно осуществить двумя способами.

Для того, чтобы ввести сигнал на входе **A**, щелчком левой кнопкой «мыши» по этому выводу (на осциллограмме входа **A** появится **чёрная полоса**), затем с помощью команды ГМ **Edit/ Overwrite/ Clock** (рис. 2.8) в диалоговом окне **Overwrite Clock** введем **1** в подменю **Multiplied By** (период сигнала будет $2 \times 1\text{нс}$) и подтверждаем нажатием кнопки **OK**.

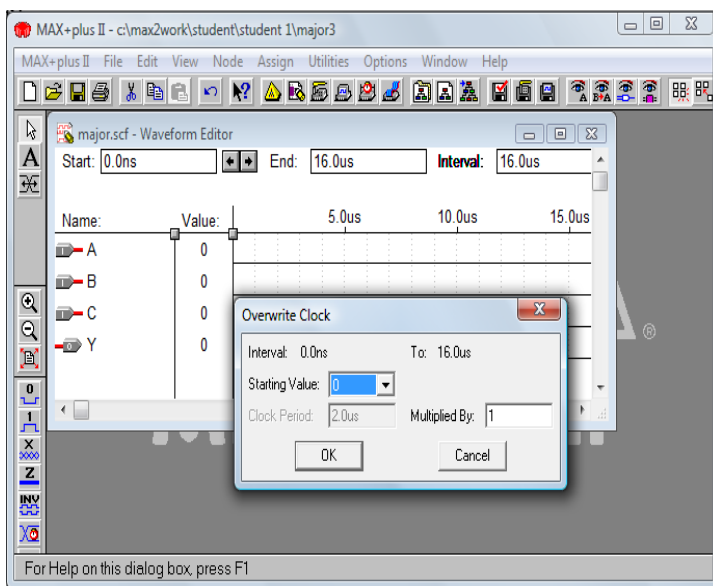


Рис. 2.8. Окно *Overwrite Clock*

Затем для входов **B** и **C** выполняем такие же действия, только в окне **Multiplied By** для входа **B** введем **2**, а для входа **C** — **4** (рис. 2.9). Итоговый файл сигнального редактора показывает сигналы с частотами F , $F/2$ и $F/4$.

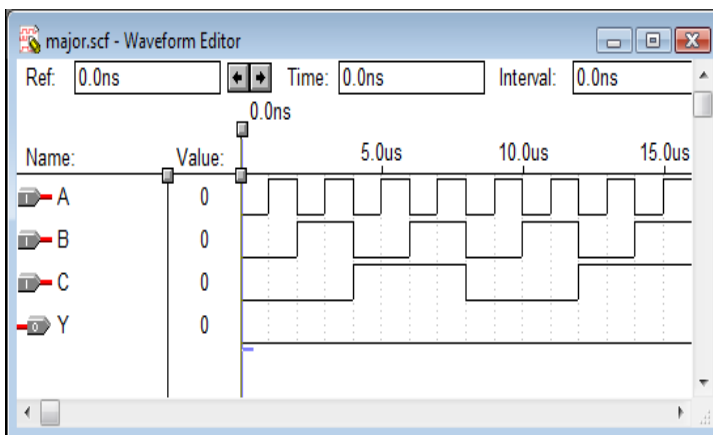


Рис. 2.9. Сигналы входов

Вводить сигналы также можно «**вручную**», используя специальную панель инструментов редактора **Waveform Editor** (рис. 2.10). Для этого на осциллограмме необходимо выделить (нажав на левую кнопку «мыши») интересующую нас область сигнала (рис. 2.11). Для выделенной области указываем «мышью» соответствующее значение на панели инструментов: означает низкий логический уровень; — высокий логический уровень; — неопределённый уровень; — уровень с высоким импедансом; — инверсия выделенного сигнала; — вызов окна **Overwrite Clock**; — вызов окна **Overwrite Count Value**, в котором можно задать параметры сигналов отсчёта.



Рис. 2.10. Панель инструментов

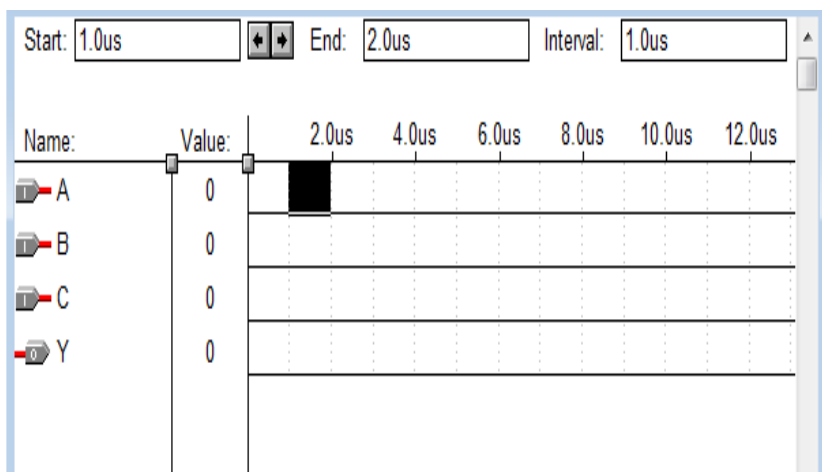


Рис. 2.11. Выделение области сигнала

В итоге можно задать такие же входные сигналы в виде осциллограмм, как на рис. 2.9.

В ГМ выберем **File/ Save**, чтобы сохранить файл с созданными входными сигналами.

2.1.4. Моделирование работы ПЛИС с помощью программы Simulator, редактирование стимулов и трасс

После создания файла **Lab1.scf** моделей (стимулов) входных сигналов можно приступить к моделированию работы созданной принципиальной схемы мажоритарного элемента **Lab1.gdf** с помощью редактора моделирования **Simulator** для получения результатов моделирования в форме осциллограммы выходного сигнала. Для этого в главном меню откроем редактор моделирования с помощью команды **MAX_PLUS II / Simulator** и в открывшемся окне редактора нажмем левой кнопкой «мышки» на клавишу **Start** (рис. 2.12). После окончания процесса моделирования необходимо вывести полученные результаты на экран, нажав «мышкой» на кнопку **Open SCF**, а затем для вывода шкалы времени осциллограмм на экран активировать команду ГМ **View / Fit in Window** (рис. 2.13). Следующим шагом является сохранение этого файла. Для этого выбираем в главном меню **File/ Save As**, вводим в окне имя **Lab2** и нажимаем **OK** (рис. 2.14)

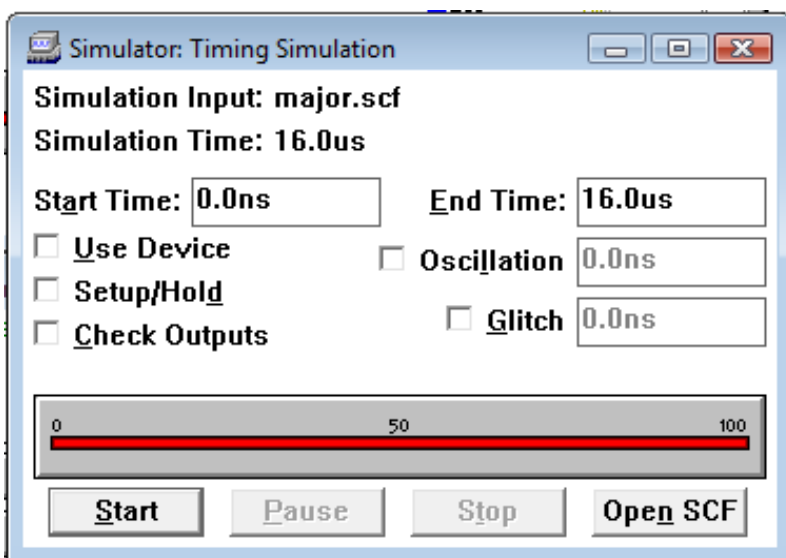


Рис. 2.12. Окно Timing Simulation редактора моделирования Simulator

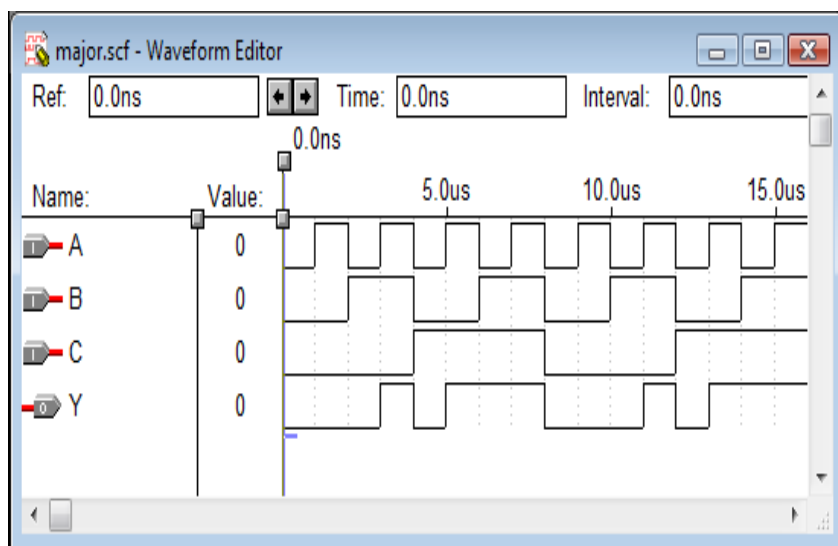


Рис. 2.13. Результаты моделирования принципиальной схемы

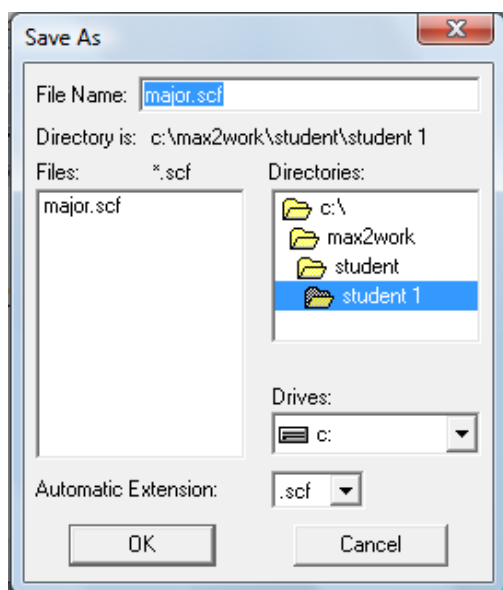
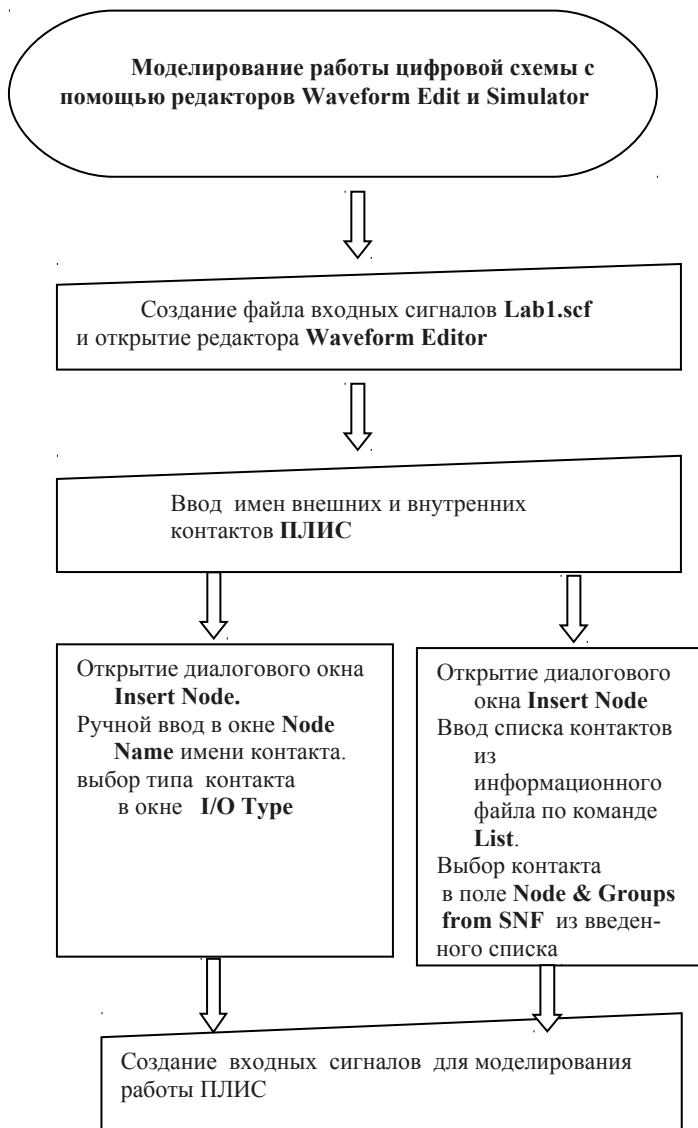


Рис. 2.14. Сохранение результатов моделирования в виде файла Lab2.scf

2.2. Структурная схема алгоритма моделирования работы ПЛИС

На рис. 2.15 представлена структурная схема алгоритма моделирования работы цифровой схемы.





Выбор временного интервала моделирования, равный 1 us,
- команда ГМ **File/ Endtime**
Ввод шкалы времени осциллограмм-команда ГМ **View / Fit in Window**
Установка масштаба сетки — 10 ns. Команда ГМ **Options/ Show Grid/ Gridsize**



Ввод тактовых импульсов

Открыть диалоговое окно **Overwrite Clock**:

- щёлкнув «мышью» по контакту ПЛИС, чтобы выделить **черную полосу** осциллограммы;
- активировать иконку **Overwrite Clock** в открывшейся левой инструментальной панели или с помощью команды ГМ **Edit/Overwrite/Clock**.

В окне **Multipled By** ввести коэффициент умножения периода и кликнуть мышкой **OK**



Ввод импульсов разной длительности и частоты следования

Выделить мышкой заданную область осциллограммы в виде **черной полосы** и в появившейся слева инструментальной панели указать тип импульсного сигнала (0, 1, тактовые импульсы и т.д.)

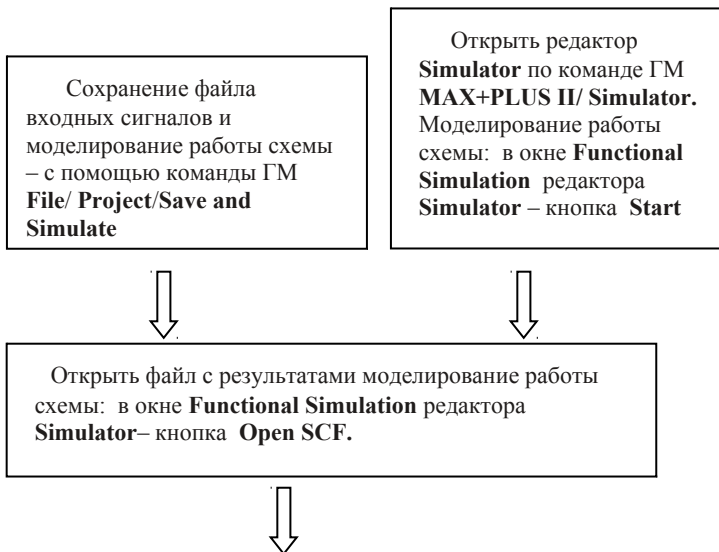


Сохранение файла – команда ГМ **File/Save**.



Моделирование работы схемы





Подготовка к работе

Ознакомиться с редакторами **Waveform Editor** и **Simulator** пакета **MAX+PLUS II**, изучив соответствующие разделы в рекомендуемой литературе и лекциях.

Рабочее задание

1. Запустить пакет **MAX+PLUS II** и открыть сигнальный редактор **Waveform Editor**.
2. Провести под руководством преподавателя моделирование учебной схемы с помощью редакторов **Waveform Editor** и **Simulator**.
3. Загрузить проект схемы цифрового устройства, разработанной в первой лабораторной работе.

4. По указанию преподавателя ввести осциллограммы входных сигналов для схем из приложения 1 в редактор **Waveform Editor**.
5. Выполнить самостоятельно моделирование работы цифрового устройства из приложения 1 при помощи редакторов **Waveform Editor** и **Simulator**.
6. Оформить отчет в печатной форме с результатами моделирования.

Контрольные вопросы

1. Какова методика моделирования цифровых схем с помощью редакторов Waveform Editor и Simulator?
2. Загрузка проекта схемы.
3. Перечислить особенности работы сигнального редактора Waveform Editor.
4. Перечислить этапы создания файла входных сигналов (Simulator Channel File).
5. Порядок ввода имен внешних контактов ПЛИС для подключения и контроля сигналов.
6. Особенности ввода имен внутренних контактов ПЛИС для подключения и контроля сигналов.
7. Редактирование входных сигналов для моделирования работы ПЛИС.
8. Особенности работы редактора моделирования Simulator.
9. Моделирование работы ПЛИС с помощью редактора Simulator.
10. Редактирование стимулов и трасс.
11. Структурная схема алгоритма моделирования работы ПЛИС.

Работа 3

Программирование ПЛИС: технические и программные средства Starter Kit

Цель — реализация разработанной и промоделированной цифровой схемы в виде работающей интегральной схемы.

Данная операция осуществляется путем загрузки файла принципиальной схемы цифрового устройства, разработанного в первой и второй лабораторных работах, в корпус ПЛИС, расположенного на плате лабораторного макета типа **UP-1** фирмы Altera, который совместно с персональным компьютером входит в комплекс **Starter Kit**, полностью обеспечивающий все этапы проектирования и отладки разрабатываемых цифровых схем (рис. 3.1).

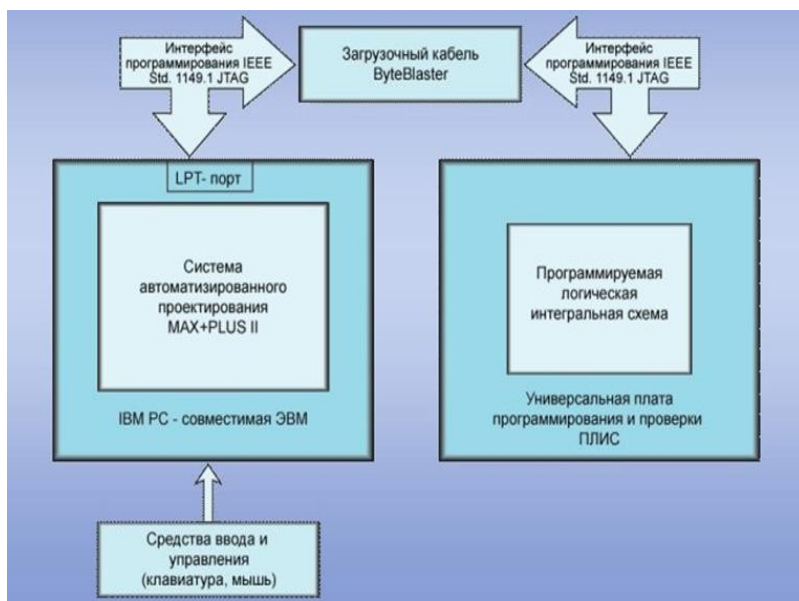


Рис. 3.1. Структурная схема проектирования ПЛИС на базе программных и технических средств фирмы Altera

Загрузка готового проекта в ПЛИС в системе **MAX+PLUS II** выполняется с помощью программатора **Programmer**, который позволяет программировать, конфигурировать, проводить верификацию и тестировать ПЛИС фирмы Altera.

3.1. Технические средства проектирования и отладки работы ПЛИС

3.1.1. Программирование ПЛИС на базе универсальной платы UP-1 фирмы ALTERA

Лабораторная плата **UP1** фирмы Altera предназначена для отработки вопросов практической реализации учебных проектов цифровых устройств на ПЛИС фирмы Altera. На плате расположены две микросхемы: одна из семейства **FLEX10K** (тип **EPF10K20**), вторая – из семейства **MAX7000** (тип **EPM7128S**). Каждой микросхеме отведены аппаратные ресурсы, включающие элементы конфигурирования, элементы индикации, нефиксируемые кнопочные выключатели, DIP-переключатели, задающие на выводах микросхем логические уровни «0» и «1», разъемы расширения для подключения внешних устройств и ряд других элементов. Плата имеет выходной разъем для подключения стандартного монитора **VGA** и входной разъем **PS/2 Mini-Din** для подключения через последовательный интерфейс устройств типа «мыши» и клавиатуры. Монитор **VGA** и последовательные устройства подключаются только к микросхеме семейства **FLEX10K**.

Загрузка файла разработанной схемы (цифрового устройства) в микросхемы осуществляется через установленный на плате входной разъем интерфейса **JTAG**. Наличие второго (выходного) разъема **JTAG** обеспечивает конфигурирование нескольких плат **UP1**, объединенных в **JTAG**-цепочку. Кроме того, для хранения и загрузки конфигурации микросхемы **EPF10K20** может быть использовано конфигурационное ПЗУ типа **EPC1**, для которого предусмотрен соответствующий установочный разъем. Загрузка конфигурации из персонального компьютера осуществляется с помощью устройства **ByteBlaster**. Устройство **ByteBlaster** подключается к параллельному порту персонального компьютера и соединяется кабелем с входным разъемом **JTAG** платы **UP1**. Плата **UP1** имеет встроенный стабилизатор напряжения питания на 5 В, работающий от внешнего нестабилизированного источника питания постоянного тока. Внешний вид и расположение электронных компонентов на плате **UP-1** показаны на рис. 3.2 и 3.3.

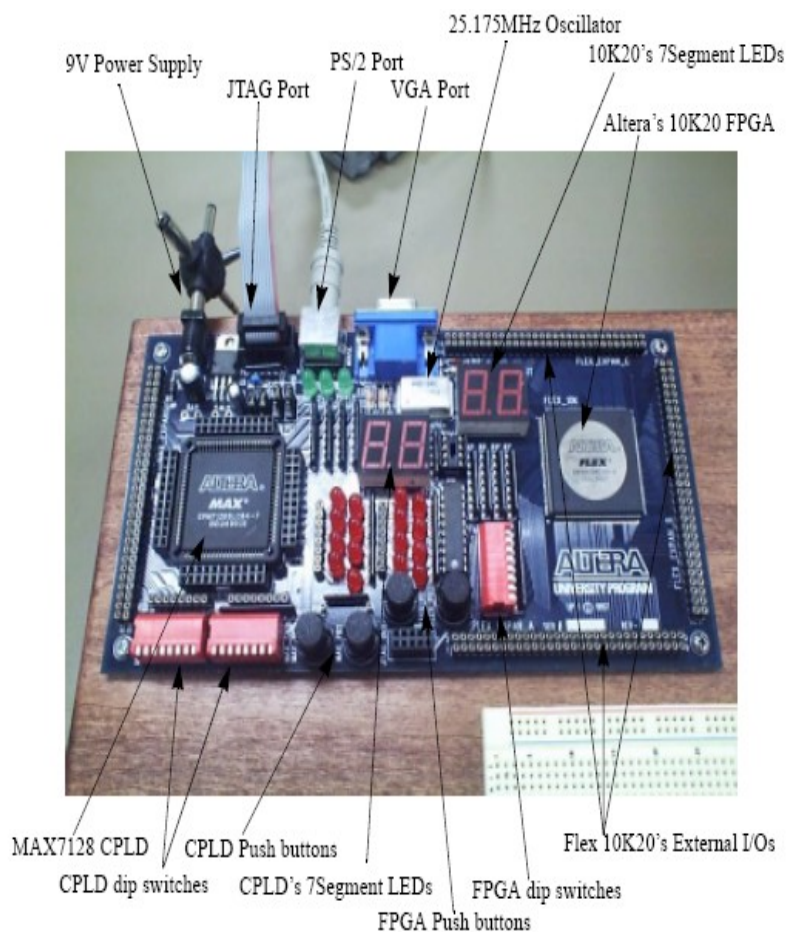


Рис. 3.2. Внешний вид платы UP-1

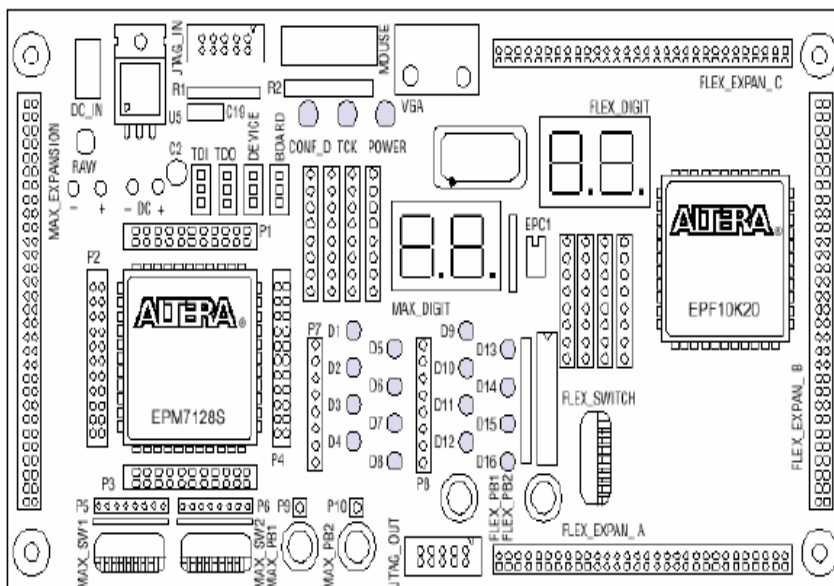


Рис. 3.3. Расположение электронных компонентов на плате UP-1

Для установки режимов работы плата **UP1** содержит четыре трёхконтактных переключателя, выполненных в виде штыревых контактов, замыкаемых перемычками (джамперами).

Переключатели на плате имеют следующие обозначения:

- **TDI** – вход данных,
- **TDO** – выход данных,
- **DEVICE** – микросхема,
- **BOARD** – плата.

Надписями **C1**, **C2**, **C3** на рис. 3.4 обозначены штыревые контакты переключателей. Возможны различные режимы конфигурации: программирование первой микросхемы **EPM7128S**, конфигурирование второй микросхемы **EPF10K20**, совместное конфигурирование обеих микросхем, конфигурирование нескольких плат **UP1**, соединенных через входные-выходные разъемы **JTAG_IN** и **JTAG_OUT**. Режимы конфигурирования и соответствующие им положения перемычек приведены в табл. 3.1.

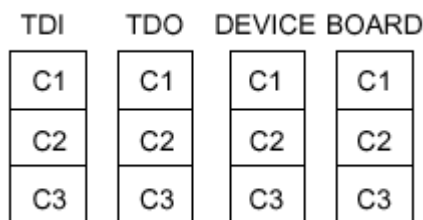


Рис. 3.4. Переключатели выбора режимов конфигурирования

Таблица 3.1

Установка режимов конфигурирования.

Режим	TDI	TDO	DEVICE	BOARD
Программирование микросхемы	C1&C2	C1&C2	C1&C2	C1&C2
Конфигурирование микросхемы	C2&C3	C2&C3	C1&C2	C1&C2
Совместное конфигурирование обеих микросхем (1)	C2&C3	C1&C2	C2&C3	C1&C2
Объединение нескольких плат UP1 (2)	C2&C3	OPEN	C2&C3	C2&C3

При совместном конфигурировании устройств в JTAG-цепочке действуют следующие правила:

- первым устройством в JTAG-цепочке должна быть микросхема **EPF10K20**, вторым – микросхема **EPM7128S**;
- последняя плата в JTAG-цепочке не должна устанавливаться в режим объединения плат.

Перед программированием или конфигурированием микросхем необходимо выполнить следующие действия:

- установить режим конфигурирования с помощью соответствующих переключателей;
- подключить кабель загрузочного устройства к параллельному порту компьютера и разъему **JTAG_IN** платы **UP1** (при этом компьютер должен быть выключен);
- включить и установить необходимые параметры программирования/конфигурирования в САПР MAX+PLUSII.

Возможны следующие режимы конфигурирования:

- программирование только микросхемы EPM7128S;
- конфигурирование только микросхемы EPF10K20;
- конфигурирование / программирование обеих микросхем;
- соединение нескольких учебных плат.

Программирование микросхемы семейства MAX7000

1. Установка режима конфигурирования на плате UP1

- Чтобы запрограммировать только микросхему EPM7128S в JTAG-цепочке, установите переключатели переключателей **TDI**, **TD0**, **DEVICE** и **BOARD** в соответствии с рис. 3.5.

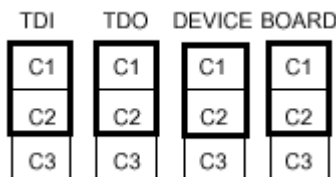


Рис.3.5. Установка режима
программирования микросхемы
EPM7128S

2. Подключение загрузочного устройства ByteBlaster

Подсоедините кабель загрузки **ByteBlaster** к параллельному порту персонального компьютера и к разъёму **JTAG_IN** на плате **UP1**.

Программирование микросхемы семейства FLEX10K

1. Установка режима конфигурирования на плате UP1

Чтобы конфигурировать только устройство EPF10K20 в JTAG-цепочке, установите переключатели переключателей **TD1**, **TD0**, **DEVICE** и **BOARD** в соответствии с рис. 3.6.

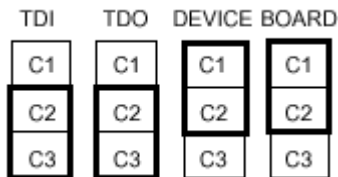


Рис. 3.6. Установка режима
программирования микросхемы
EPF10K20

2. Подключение загрузочного устройства ByteBlaster

Подсоедините кабель загрузки ByteBlaster к параллельному порту персонального компьютера и к разъёму JTAG_IN на учебной плате.

3.1.2. Интерфейс (линии связи ПЛИС с ПК) типа ByteBlaster

ПЛИС на плате Altera UP1 программируется с помощью редактора **Programmer** пакета **MAX+plus II** и ленточного кабеля **ByteBlaster** (рис. 3.7). Устройство **ByteBlaster** подключается к параллельному порту персонального компьютера и соединяется кабелем с входным разъемом **JTAG** платы **UP1**.

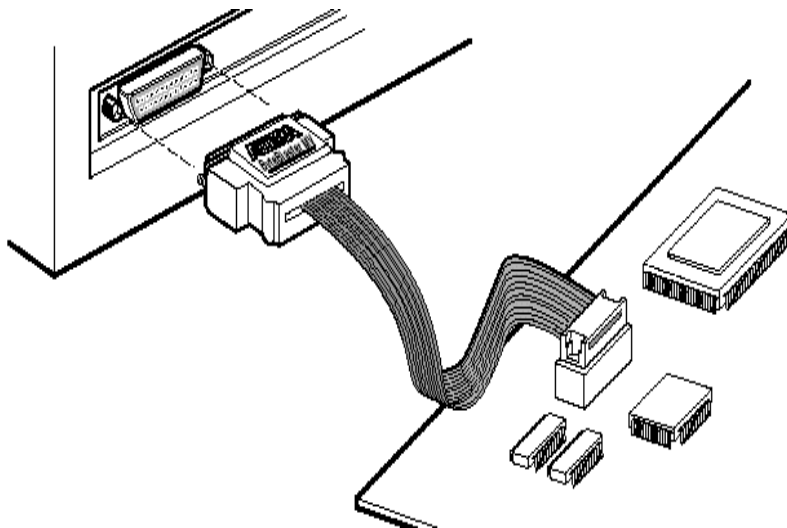


Рис. 3.7. Подключение к ПК интерфейса (линия связи) ByteBlaster

3.2. Программные средства загрузки и отладки файла принципиальной схемы в ПЛИС с помощью компонента Programmer

3.2.1. Установки в компоненте Programmer для ПЛИС EPM7128S

Установка опций JTAG в пакете MAX+PLUSII

1. Включите **Multi-Device** в меню **JTAG**, если необходимо программировать более чем одно устройство EPM7128S.
2. Выберите **Multi-Device** в меню **JTAG**.
3. Выберите **EPM7128S** из списка в **Device Name** в диалоговом окне **Multi-Device** в меню **JTAG**.
4. Введите название программируемого файла для **EPM7128S** устройства в окне **Programming File Name**. Для просмотра директории вашего компьютера и выбора соответствующего файла может использоваться кнопка **Select Programming**.
5. Нажмите **Add** для добавления устройства и программируемого файла в окно **Devices Name & Programming File Names**. Нумерация имён устройств слева показывает порядок устройства в **JTAG**-цепочке. Программируемый файл устройства отображен на той же самой строке как имя устройства. Если никакой программируемый файл не связан с устройством, " <ни один> " отображен рядом с именем устройства.
6. Для проверки **ByteBlaster** кабеля, кода **JTAG ID** и полной инструкции **JTAG**-цепочки нажмите кнопку **Detect JTAG Chain Info**. Прямо над кнопкой будет выведено сообщение об обнаружении кабеля **ByteBlaster**. Проверьте, соответствует ли это сообщение информации в окне **Devices Name & Programming File Names**.
7. Чтобы сохранить текущие установки **JTAG Chain File (.jcf)**, нажмите **Save JCF**. В диалоговом окне **Save JCF** введите название файла в **File Name**, затем выберите каталог в окне каталогов, после чего нажмите **OK**.

8. Нажмите **ОК**, чтобы сохранить ваши изменения.
9. Нажмите кнопку **Program** в окне редактора **Programmer** пакета **MAX+PLUSII**.

3.2.2. Установки в редакторе **Programmer** для ПЛИС **EPF10K20**

Установка опций **JTAG** в пакете **MAX+PLUSII**

1. Включите **Multi-Device** в меню **JTAG**, если необходимо программировать более чем одно устройство **EPF10K20**.
2. Выберите **Multi-Device** в меню **JTAG**.
3. Выберите **EPF10K20** из списка в **Device Name** в диалоговом окне **Multi-Device** в меню **JTAG**.
4. Введите название программируемого файла для микросхемы **EPF10K20** в окне **Programming File Name**. Для просмотра директории вашего компьютера и выбора соответствующего файла может использоваться кнопка **Select Programming**.
5. Нажмите **Add** для добавления устройства и программируемого файла в окно **Devices Name & Programming File Names**. Нумерация имен устройств слева показывает порядок устройства в **JTAG**-цепочке. Программируемый файл устройства отображен на той же самой строке как имя устройства. Если никакой программируемый файл не связан с устройством, " <ни один> " отображен рядом с именем устройства.
6. Для проверки **ByteBlaster** кабеля, кода **JTAG ID** и полной инструкции **JTAG**-цепочки нажмите кнопку **Detect JTAG Chain Info**. Прямо над кнопкой будет выведено сообщение об обнаружении кабеля **ByteBlaster**. Проверьте, что это сообщение соответствует информации в окне **Devices Name & Programming File Names**.
7. Чтобы сохранить текущие установки **JTAG Chain File (.jcf)**, нажмите **Save JCF**. В диалоговом окне **Save JCF** введите название файла в **File Name**, затем выберите каталог в окне каталогов, нажмите **ОК**.
8. Нажмите **ОК**, чтобы сохранить ваши изменения.
9. Нажмите кнопку **Configure** в окне редактора **Programmer** пакета **MAX+PLUSII**.

3.3. Программирование ПЛИС

Для программирования ПЛИС в плате **UP-1**, установите *перемычки* на программирование ПЛИС типа **EPM7128S**. Подключите кабель **ByteBlaster** к параллельному порту компьютера и **JTAG**-входу платы **UP-1**. Подключите питание платы с помощью 9-вольтового адаптера.

3.3.1. Назначение номеров выводов

Конечным этапом этой работы должно стать программирование ПЛИС. Прежде чем запрограммировать цифровую схему в ПЛИС, назначим входы и выходы контактов этой ПЛИС. Для этого необходимо перекомпилировать файл проекта и запрограммировать ПЛИС. Перед выполнением этого пункта необходимо проверить наличие файла в проекте (команда ГМ **FILE/Project/Set Project to Current file**).

Существует два способа назначения номеров выводов.

1. Для того чтобы назначить номер выводу, необходимо щёлкнуть по выводу, чтобы выделить его. Затем, нажав на правую кнопку «мыши», выбрать **Assign, Pin/Location/Chip**. Также это можно сделать через команду **Assign** главного меню.
2. Назначим номера выводов в диалоговом окне, как показано на рис. 3.8. Введем **A** в окно **Node Name**, **12** — в окно **Pin** и нажмем **Add**. Введем **B** в окно **Node Name**, **16** — в окно **Pin** и нажмем **Add**. Повторим эту процедуру, пока не будут введены все номера, как в табл. 3.2. Когда всё будет сделано, нажмем **OK**.

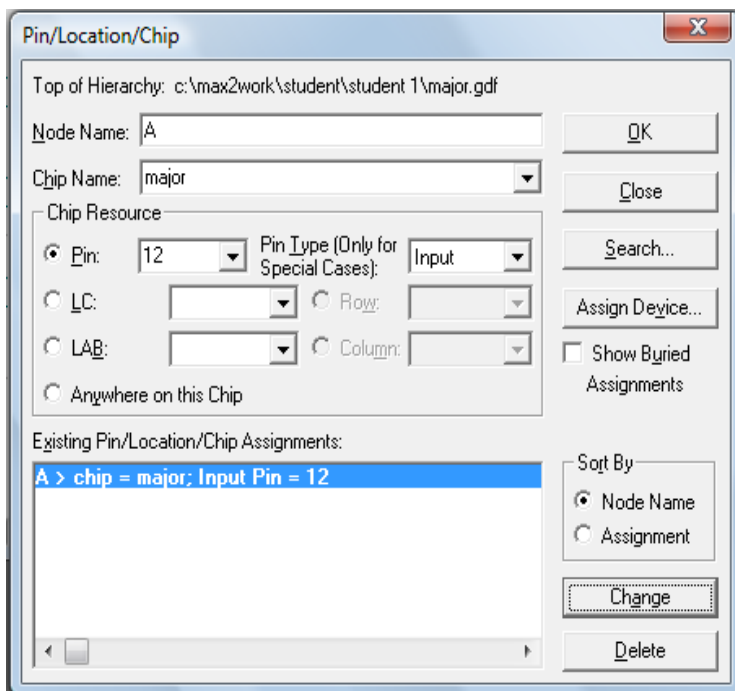


Рис. 3.8. Диалоговое окно Pin/Location/Chip

Таблица 3.2

Ввод номеров контактов

Pin Name	Pin Number
A	12
B	16
C	18
Y	4

Также можно назначить номера выводам, редактируя **Assignment and Configuration File (acf)**. Для этого выберем в ГМ команды **File/Open**. В открывшемся диалоговом окне **Open**, в нашей папке выбираем файл **major** с расширением **.acf** (рис. 3.9).

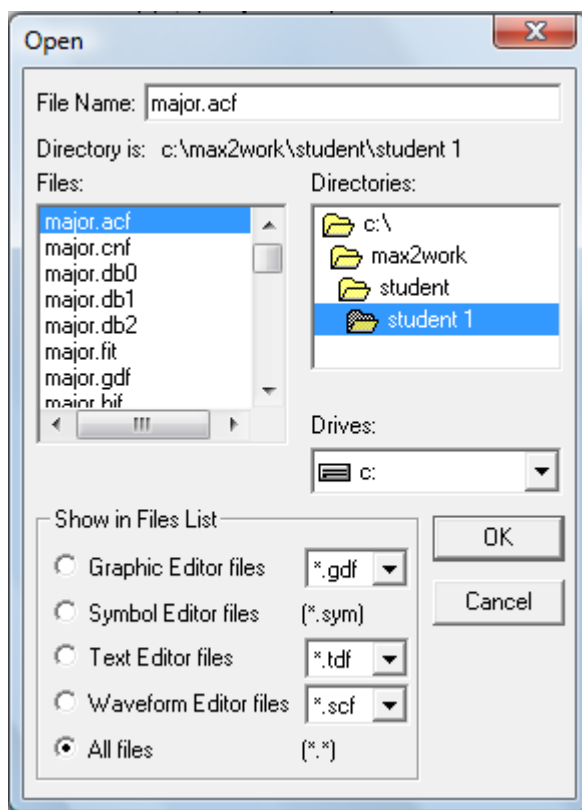


Рис. 3.9. Диалоговое окно *Open*

Отредактируем этот файл, привязывая выходы нашей схемы к «ножкам» программируемой ПЛИС, как в табл. 3.2 (рис. 3.10).

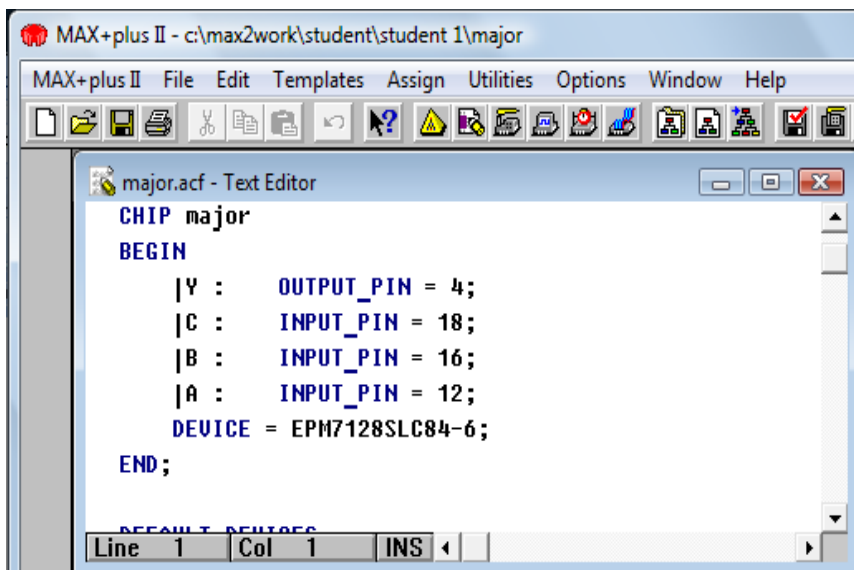


Рис. 3.10. Файл Assignment and Configuration File (acf)

На рис. 3.11 показаны выводы, так как они появятся в файле .gdf.

Откроем файл проекта, который необходимо загрузить в плату (**major.gdf**). Установим проект в текущий файл. Откроем программатор **Programmer** из ГМ **MAX+plus II**. Если вы ранее не программировали ПЛИС, то необходимо установить конфигурацию ПЛИС. Выберем **Hardware Setup** в ГМ **Options** (рис. 3.12). Выберем **ByteBlaster** в диалоговом окне **Hardware Setup**.

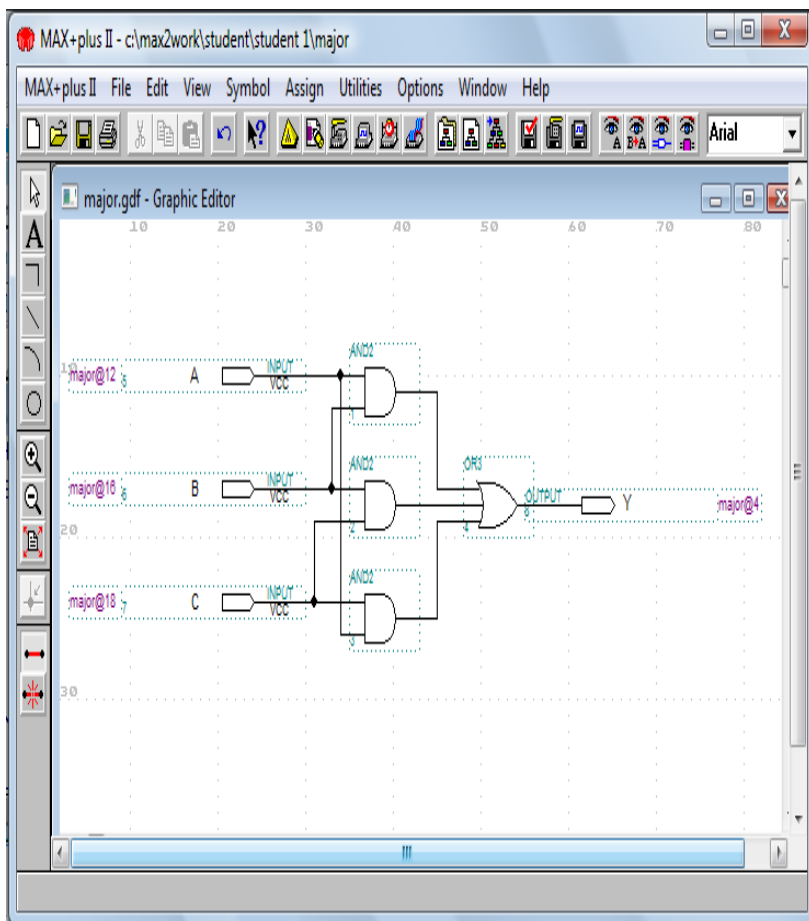


Рис. 3.11. Файл .gdf после назначения выводов

Так как текущий проект был скомпилирован для устройства **MAX7000S**, **pof**-файл будет доступен автоматически. Появится диалоговое окно программатора (рис. 3.13). Для загрузки схемы в ПЛИС нажмем кнопку **Program**.

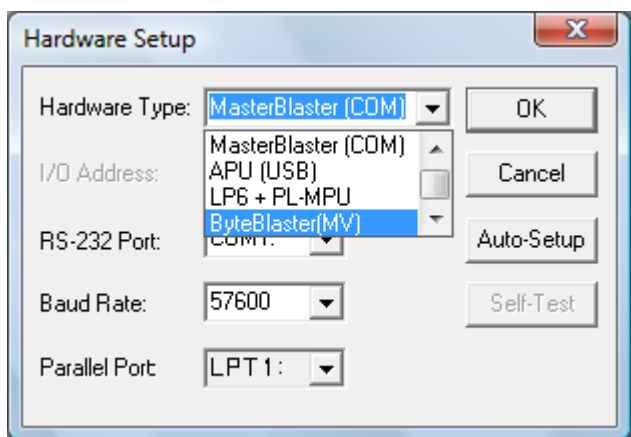


Рис. 3.12. Диалоговое окно Hardware Setup

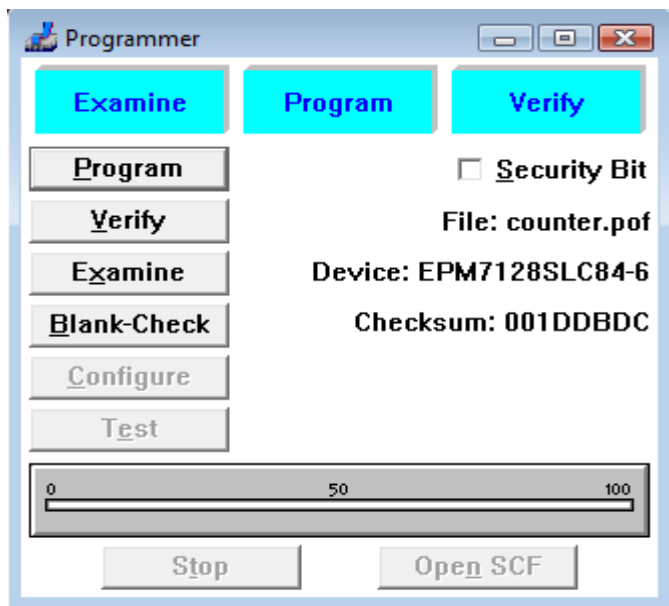
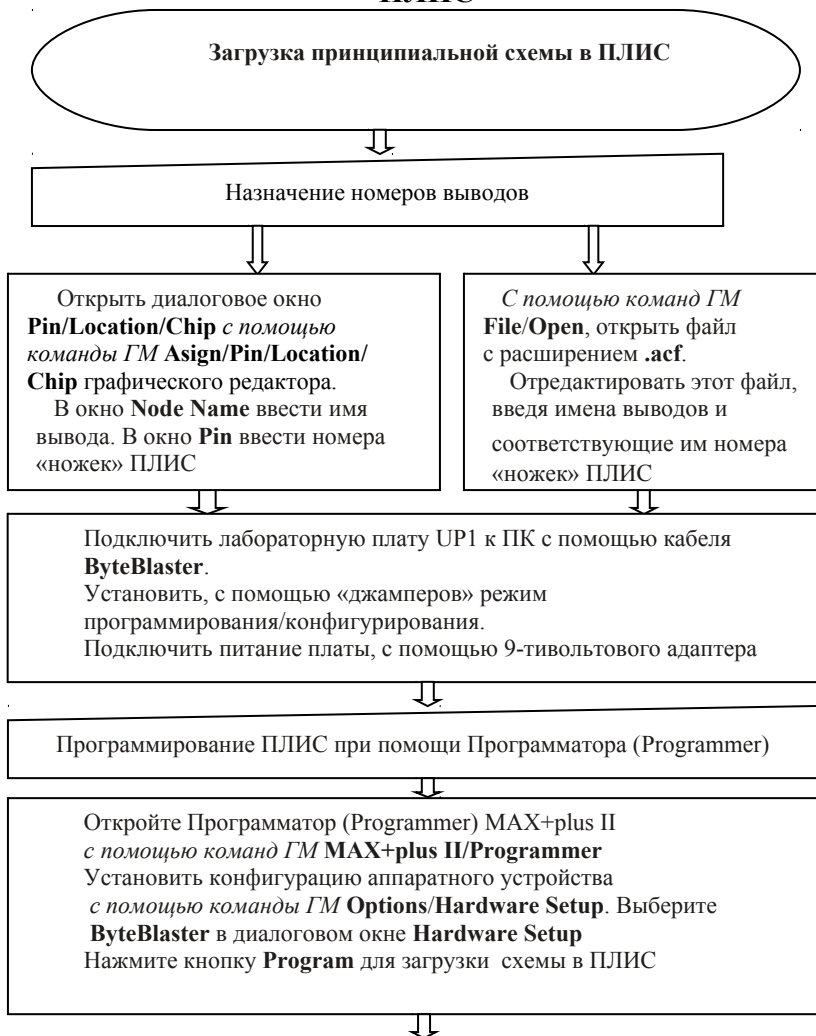


Рис. 3.13. Диалоговое окно Programmer MAX7000S

3.4. Структурная схема алгоритма программирования ПЛИС



Подготовка к работе

Ознакомиться с описанием лабораторного макета **UP1** фирмы Altera и компонентом **Programmer** пакета **MAX+PLUS II**, изучив соответствующие разделы в рекомендуемой литературе и лекциях.

Рабочее задание

1. Выполнить под руководством преподавателя программирование ПЛИС с помощью компонента **Programmer**.
2. Загрузить самостоятельно в ПЛИС цифровую схему, разработанную в работах 1 и 2 с помощью компонента **Programmer**.
3. Произвести редактирование входных и выходных контактов ПЛИС и выполнить программирование повторно.
4. Оформить отчет в печатной форме.

Контрольные вопросы

1. Какие ПЛИС находятся на лабораторном макете UP1?
2. Возможно ли программирование нескольких макетов UP1 последовательно?
3. Каковы особенности и способы назначения номеров выводов?
4. Порядок подключения лабораторной платы UP1 к персональному компьютеру и установки режимов программирования и конфигурирования ПЛИС.
5. Программирование ПЛИС при помощи компонента Programmer.
6. Ввод имен входных и выходных контактов ПЛИС.
7. Назначение имен внутренних контактов ПЛИС.
8. Как осуществляются электрические соединения в ПЛИС?
9. Структурная схема алгоритма программирования ПЛИС.
10. Проверка ошибок программирования ПЛИС.
11. Сохранение результатов программирования ПЛИС.
12. Выбор типа ПЛИС.
13. Информационная защита ПЛИС.

Работа 4

Разработка цифровой схемы на языке VHDL

Цель — изучение программного способа разработки цифровых устройств на базе языка описания работы цифровых схем типа VHDL.

4.1. Методика проектирования схем на языке VHDL

4.1.1. Создание нового проекта

В качестве альтернативы схемному проектированию цифрового устройства используется текстовый ввод схемы на таких языках, как, например, **AHDL** (**A**ltera **H**ardware **D**escription **L**anguage) или **VHDL** (**V**ery **H**igh **S**peed **I**ntegrated **C**ircuit **H**ardware **D**escription **L**anguage). Разработчик создаёт текстовый файл в соответствии с синтаксисом языка и использует компилятор для создания программ, аналогичных графическим файлам.

Язык типа **AHDL** представляет собой *систему программирования* типа **Assembler** и, хотя очень прост в использовании, имеет более узкое распространение по сравнению с *языком высокого уровня* **VHDL**. Компилятор MAX+plus II поддерживает оба языка, но ввиду того, что на **VHDL** пишутся программы также и для микроконтроллеров и DSP-процессоров, рассмотрим особенности проектирования цифровых схем на этом языке.

Каждый файл VHDL состоит из двух разделов: ввода данных схемы (**entity**) и структуры программы (**architecture**). В начале программы разработчик определяет внешние аспекты VHDL, то есть входные и выходные имена и имя функции. Во второй части (**architecture**) определяются внутренние связи, или как входы и выходы цифровой схемы будут вести себя относительно друг друга и относительно других сигналов или функций.

Рассмотрим структуру проекта VHDL на примере мажоритарно-го элемента, изображённого на рис. 4.1.

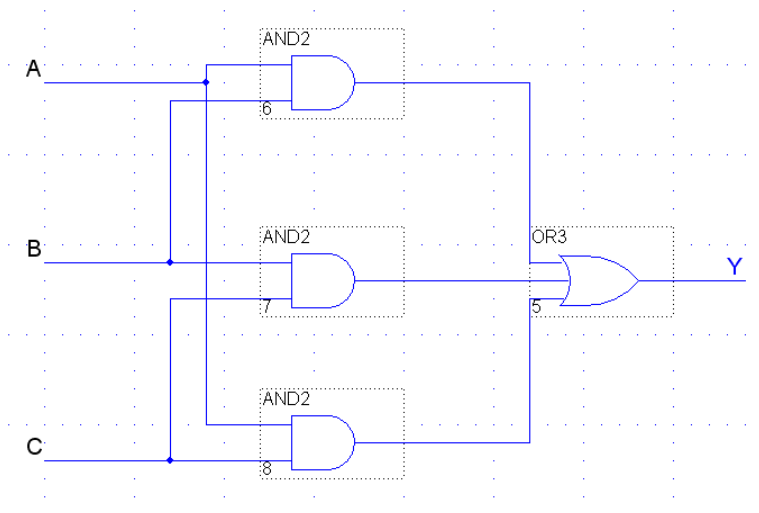


Рис. 4.1. Схема мажоритарного элемента

Для создания файла текстового редактора в главном меню выберем **File/New** и в открывшемся диалоговом окне **New** выбираем **Text Editor file** (рис. 4.2). Нажимаем **OK**.

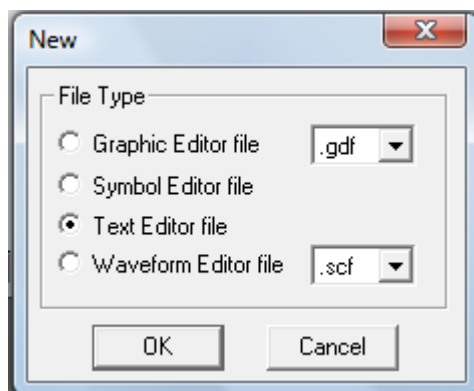


Рис. 4.2. Окно New

4.1.2. Ввод описания цифровой схемы на языке VHDL

Теперь необходимо в текстовом редакторе ввести описание цифровой схемы на языке VHDL. Сделать это можно двумя способами.

В первом случае необходимо составить на языке **VHDL** описание цифровой схемы (рис. 4.3). Полный файл VHDL для мажоритарного элемента показан ниже. Двойные тире в начале двух первых строк указывают, что эти строки являются комментариями.

В программе под именем *major.vhd* после деклараций указывается библиотека *LIBRARY ieee*, содержащая стандартные логические элементы (вентили) *ieee.std_logic_1164.ALL* (155-я серия российской логики), а затем в разделе *ENTITY* входы *a*, *b*, *c* и выход *y* схемы, которые далее будет необходимо привязать к входным и выходным контактам (портам) ПЛИС. В следующем разделе *ARCHITECTURE* записывается алгоритм работы логики мажоритарного элемента $y \leq (a \text{ and } b) \text{ or } (b \text{ and } c) \text{ or } (a \text{ and } c)$, после чего указывается на конец программы *END majority*:

```
-- major.vhd
-- VHDL реализация мажоритарного элемента
--Library contains standard VHDL logic types
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;
--Entity defines inputs and outputs
ENTITY major2 IS
PORT (
a, b, c : IN STD LOGIC;
y       : OUT STD LOGIC);
END major2;
--Architecture describes input/output relationship
ARCHITECTURE majority OF major2 IS
BEGIN
y <= (a and b) or (b and c) or (a and c);
END majority;
```

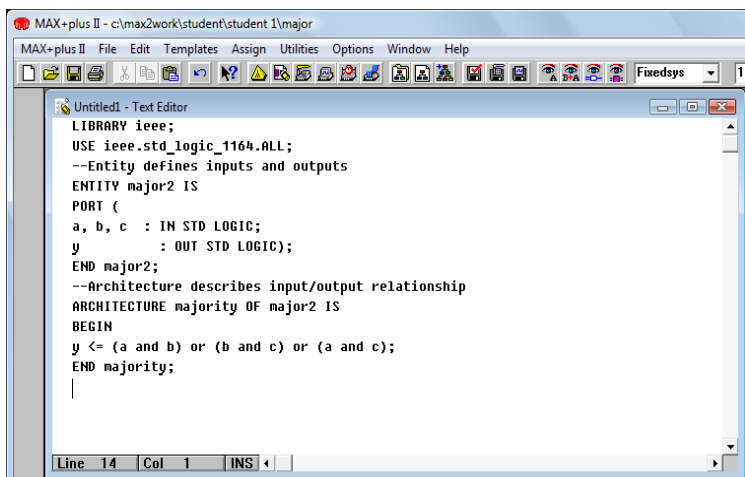


Рис. 4.3. Описание схемы на языке VHDL

Во втором случае необходимо использовать для написания программы стандартный шаблон типа **VHDL Templates**.

4.2. Оформление программы на языке VHDL с помощью стандартного шаблона типа Templates

В пакете MAX+plus II имеется стандартный шаблон **Templates** для ускорения составления программы на языке VHDL. Активируя команду ГМ **Templates**, выделим в подменю команду **VHDL Template** (рис. 4.4) и проанализируем появившийся на дисплее шаблон для написания программы (рис. 4.5).

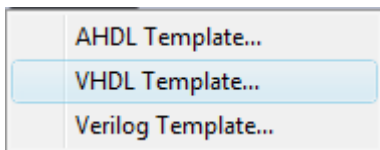


Рис. 4.4. Меню Template

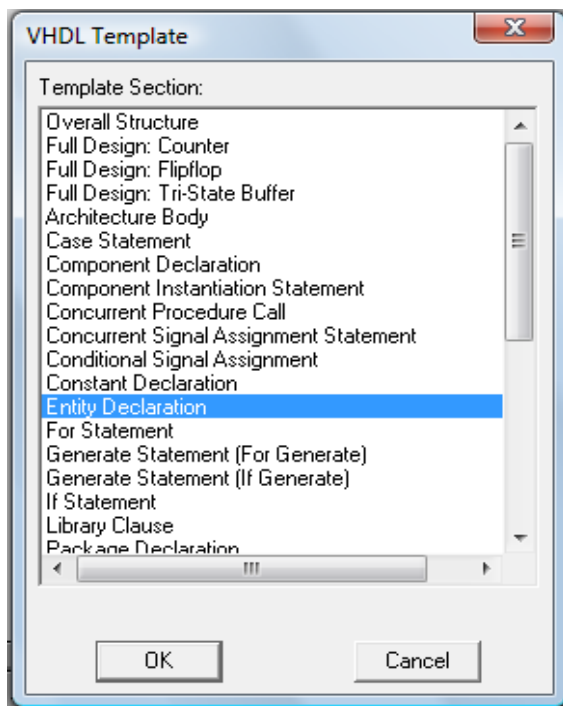


Рис. 4.5. Диалоговое окно VHDL Template

Для написания программы в диалоговом окне поочередно работаем с шаблонами **Entity Declaration**, **Library Clause** и **architecture Body** и заполняем пропуски необходимой информацией. В качестве примера активируем в шаблоне команду **Entity Declaration** и запишем часть программы на языке VHDL:

```
ENTITY __entity_name IS
  GENERIC(__parameter_name : string := __default_value;
    __parameter_name : integer:= __default_value);
  PORT(
    __input_name, __input_name          : IN    STD_LOGIC;
    __input_vector_name                 :        IN
    STD_LOGIC_VECTOR(__high downto __low);
    __bidir_name, __bidir_name          :        INOUT
    STD_LOGIC;
    __output_name, __output_name : OUT  STD_LOGIC);
END __entity_name;
```


Удалим в заготовке линии, которые нам не нужны, и заменим имена вводов и выводов цифровой схемы. У нашего мажоритарного элемента входы a, b, c и выход y. Таким образом, мы можем модифицировать шаблон:

```
ENTITY major IS
    PORT (
        a, b, c : IN STD_LOGIC;
        y : OUT STD_LOGIC);
END major;
```

Аналогично, активируя в шаблоне команду **Architecture Body**, составим программу работы мажоритарного элемента:

```
--Architecture describes input/output relationship
ARCHITECTURE majority OF major IS
BEGIN
    y <= (a and b) or (b and c) or (a and c);
END majority;
```

4.3. Сохранение и компиляция файла

После ввода текстового описания, файл необходимо сохранить. Для этого в главном меню выбираем **File/Save As**. В открывшемся диалоговом окне Save As вводим название файла **major** и выбираем автоматическое расширение **.vhd** (рис. 4.6). Для проверки файла на ошибки в главном меню выбираем **File/Project/Set Project to Current File**. После этого выбираем **File/Project/Save and Compile**, и файл будет откомпилирован и сохранён.

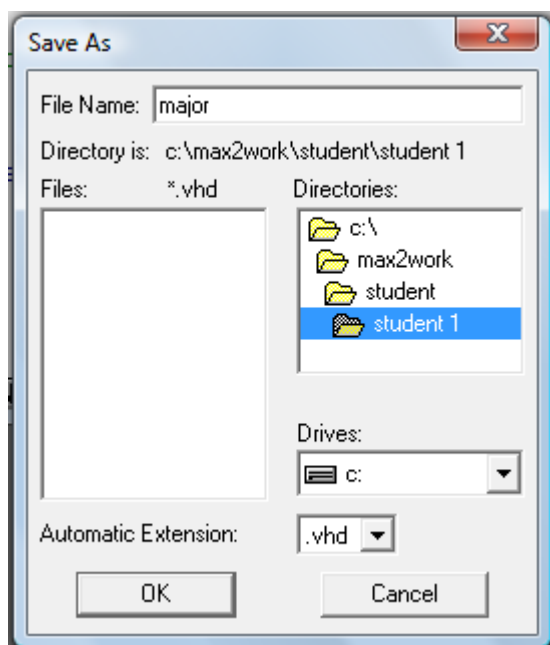
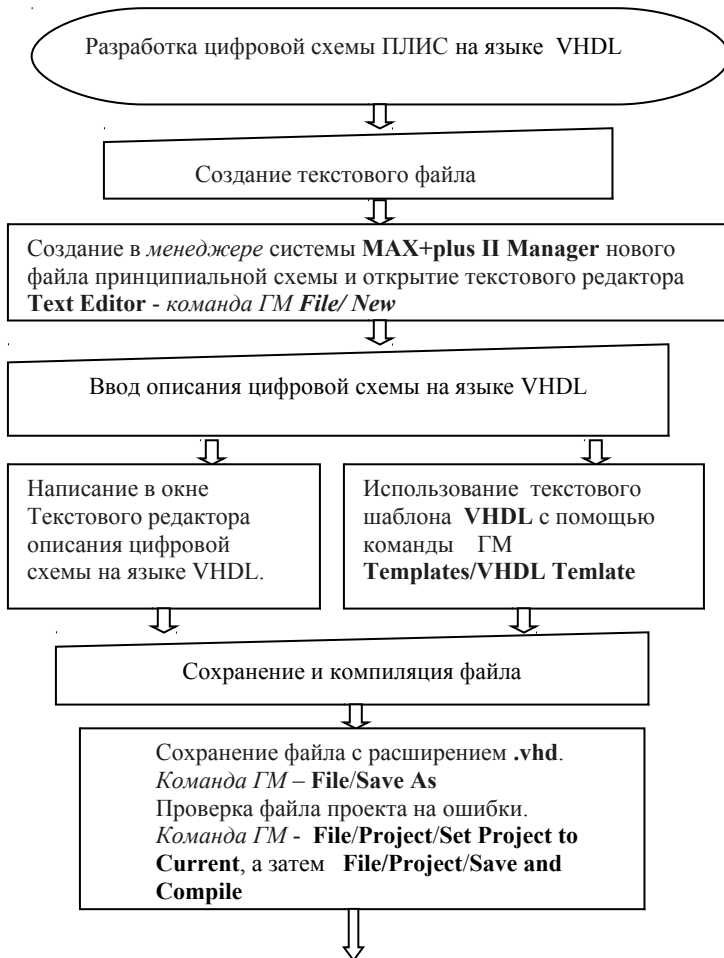


Рис. 4.6. Окно Save As

Следующим шагом является моделирование работы мажоритарного элемента, рассмотренное во второй лабораторной работе.

4.4. Структурная схема алгоритма создания цифровой схемы на языке VHDL



Подготовка к работе

Ознакомиться с текстовым редактором **Text Editor** и основами программирования ПЛИС с помощью языка высокого уровня типа **VHDL** в пакете **MAX+PLUS II**, изучив соответствующие разделы в рекомендуемой литературе и лекциях.

Рабочее задание

При помощи шаблона **Templates** создать на языке высокого уровня **VHDL** в текстовом редакторе **Text Editor** схему, выбранную из справочника или в приложении 1. Промоделировать работу цифрового устройства, описанного на языке **VHDL**.

Контрольные вопросы

1. Из каких структур состоит файл VHDL?
2. В чем отличие языков AHDL и VHDL?
3. Можно ли совмещать в одном проекте элементы, созданные в графическом и текстовом редакторах?
4. Методика проектирования принципиальной схемы на языке VHDL.
5. Создание нового проекта и работа с библиотеками.
6. Программирование ввода в схему электронных компонентов.
7. Программирование электрических соединений схемы.
8. Именованые цепей схемы.
9. Программный ввод шин.
10. Программирование позиционных обозначений на схеме.
11. Проверка ошибок и правил выполнения принципиальных схемы.
12. Структурная схема алгоритма проектирования принципиальной схемы на языке VHDL.

Список рекомендуемой литературы

1. *Петропавловский В.П., Шалагинов А.В., Азаров Д.А.* Компьютерное проектирование электронных схем на базе САПР типа OrCAD. Современные электронные цифровые технологии / Под ред. В.П. Петропавловского: Лабораторный практикум. М.: МИФИ, 2005.
2. Конспект лекций по курсу «Виртуальная микроэлектроника».
3. *Суворова Е. А., Шейнин Ю. Е.* Язык VHDL для проектирования систем на СБИС. СПб.: СПбГУАП, 2001.
4. *Бродин В.Б., Калинин А.В.* Системы на микроконтроллерах и БИС программируемой логики. М.: ЭКОМ, 2002.
5. Системы автоматизированного проектирования фирмы Altera MAX+II и Quartus II /Комолов Д.А. и др.: Краткое описание и самоучитель. М.: ИП РадиоСофт, 2002.
6. *Ашихмин А.С.* Цифровая схемотехника. Шаг за шагом. М.:Диалог МИФИ, 2008.

Приложение 1

Примеры разработки цифровых схем на ПЛИС

Вариант № 1

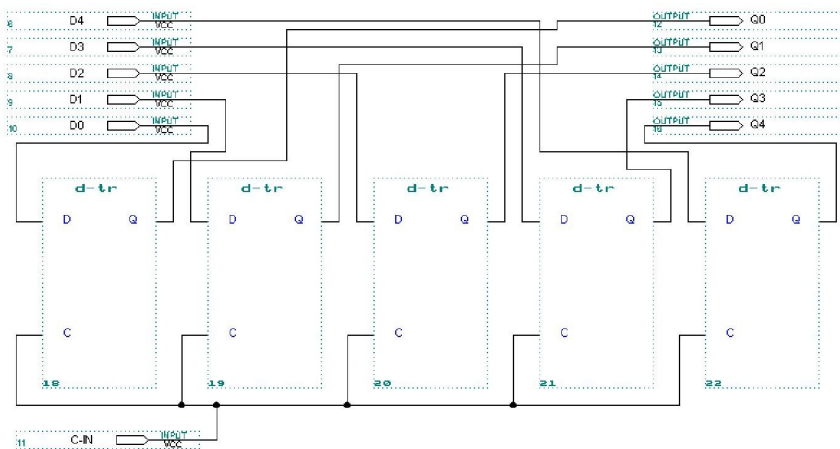


Рис. П.1.1. Регистр

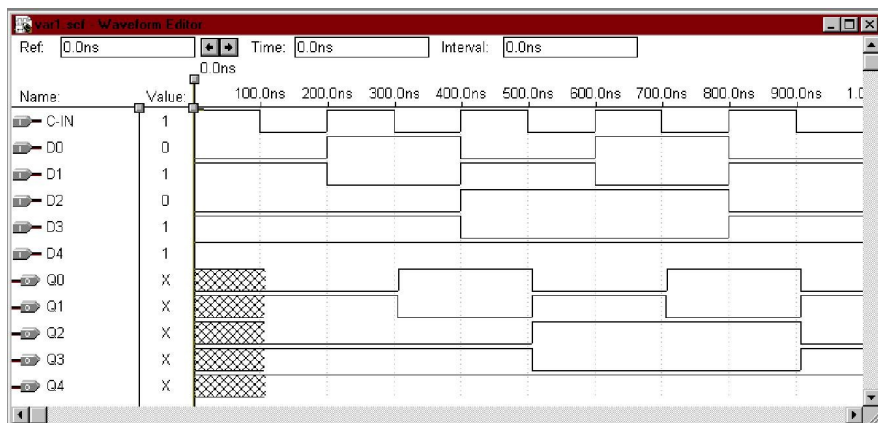


Рис. П.1.2. Результаты моделирования регистра

Вариант № 2

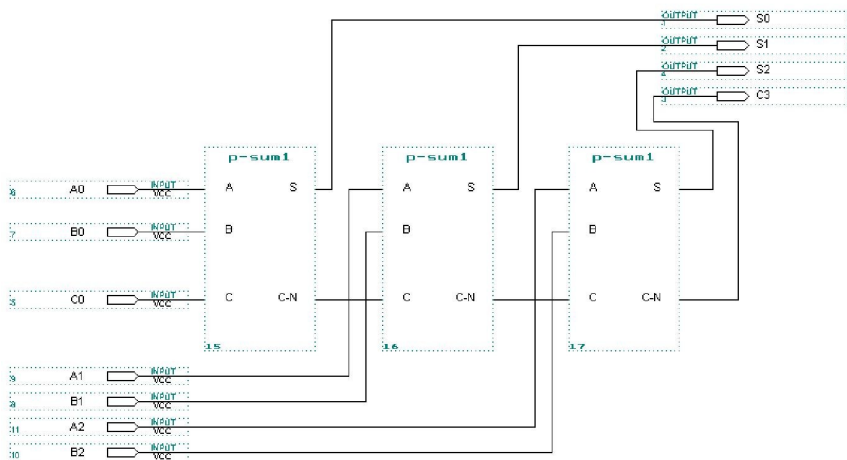


Рис. П.1.3. Трехразрядный сумматор на элементах исключающее ИЛИ, И-НЕ

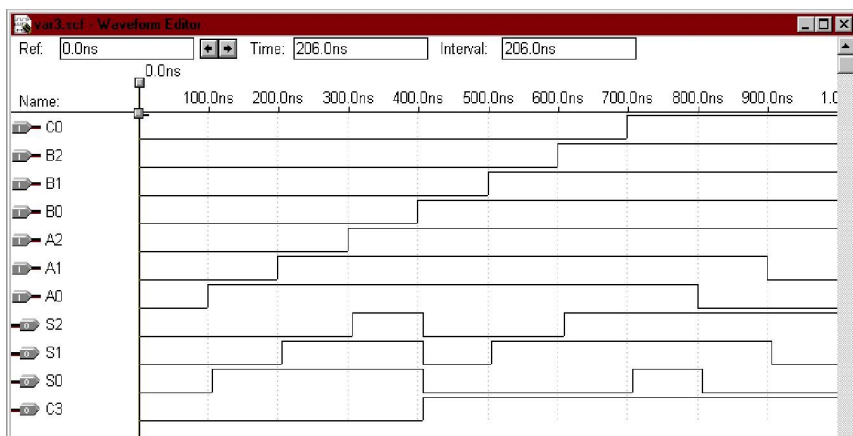


Рис. П.1.4. Результаты моделирования трехразрядного сумматора на элементах исключающее ИЛИ, И-НЕ

Вариант № 3

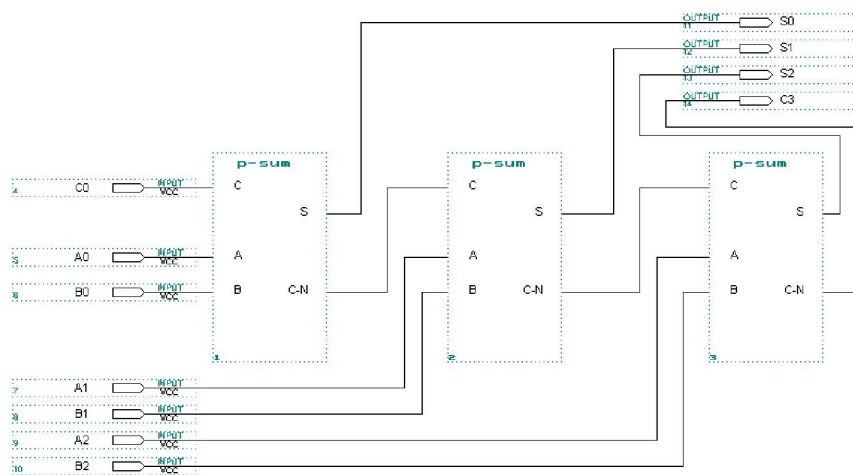


Рис. П.1.5. Трехразрядный сумматор на элементах И-НЕ

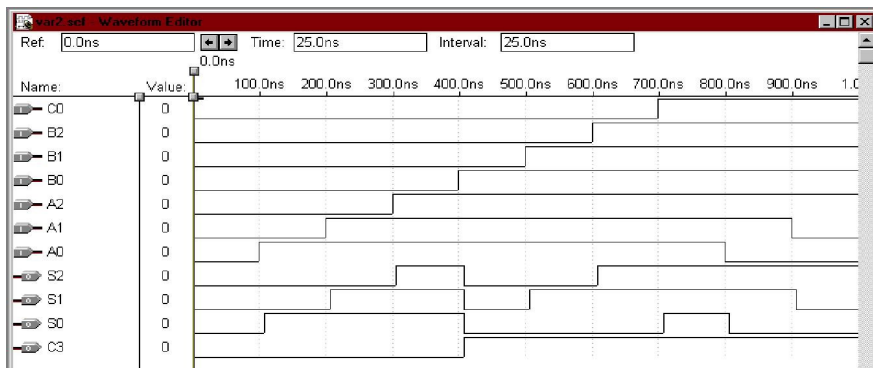


Рис. П.1.6. Результаты моделирования трехразрядного сумматора на элементах И-НЕ

Вариант № 4

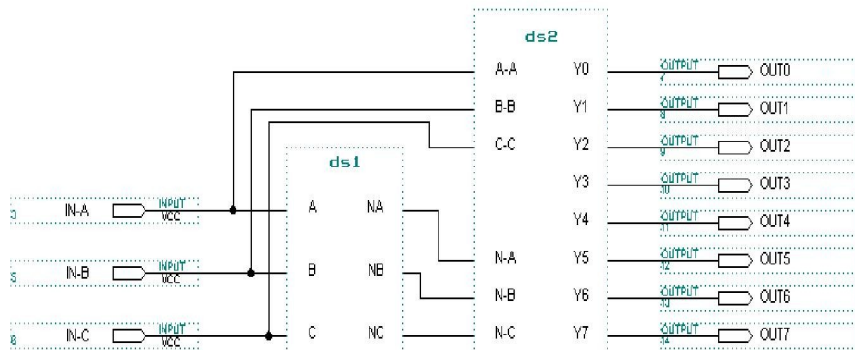


Рис. П.1.7. Дешифратор на элементах И-НЕ

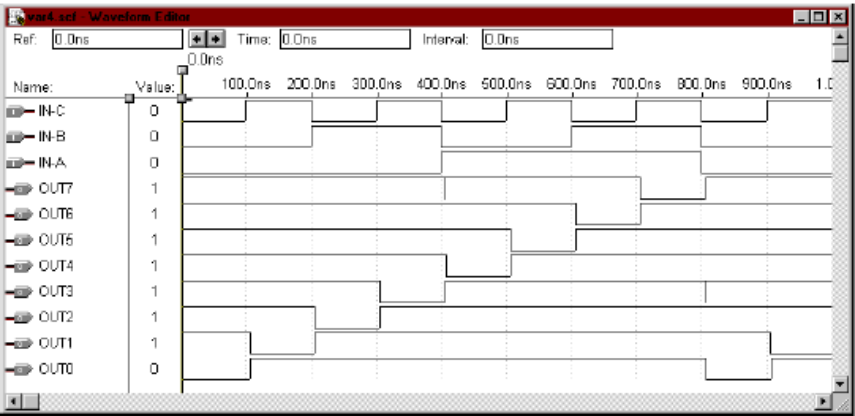


Рис. П.1.8. Результаты моделирования дешифратора на элементах И-НЕ

Вариант № 5

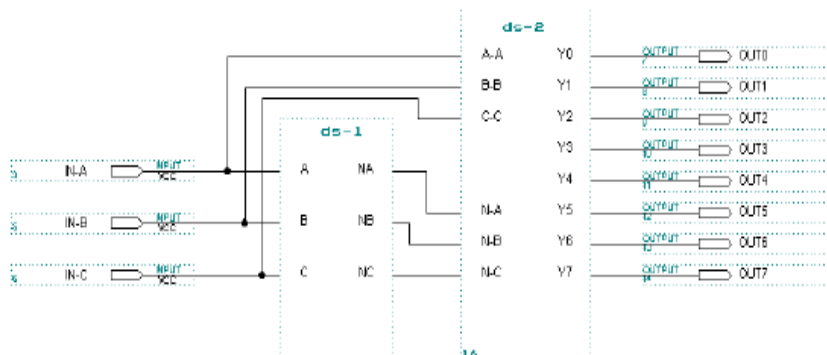


Рис. П.1.9. Дешифратор на элементах ИЛИ-НЕ

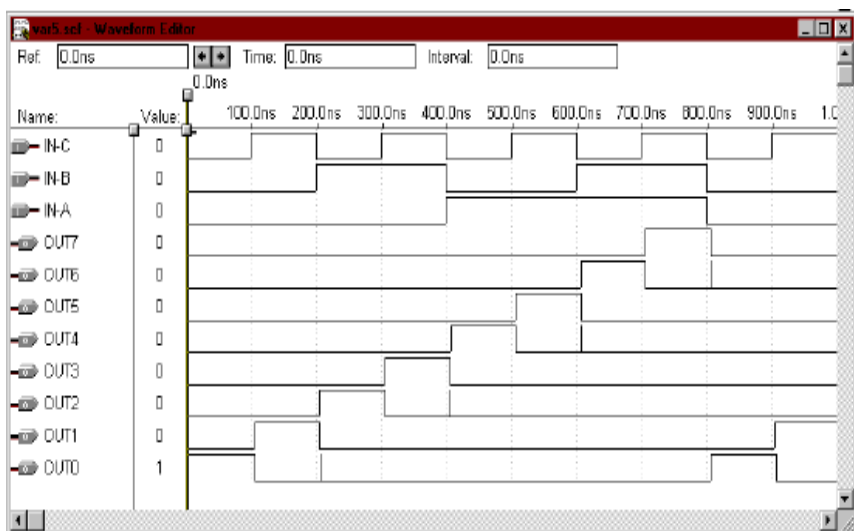


Рис. П.1.10. Результаты моделирования дешифратора на элементах ИЛИ-НЕ

Приложение 2

Описание и характеристики технических средств проектирования ПЛИС

ПЛИС EPM7128S семейства MAX7000 имеет 84 вывода. (Схема цоколевки представлена на рис. П.2.1) Микросхемы MAX 7000, кроме 44-выводных, поддерживают MultiVolt I/O интерфейс, который позволяет их подключать к системам с различным питанием. 5 В микросхемы во всех корпусах могут работать с 3,3 В или 5,0 В устройствами. Микросхемы имеют наборы контактов питания для внутреннего ядра и входных буферов (VCCINT) и для выходных драйверов (VCCIO). ПЛИС EPF10K20, семейства FLEX10K имеет 240 выводов. Расположение выводов схемы приведено на рис. П.2.2, а цоколевка — в табл. П.2.1 .

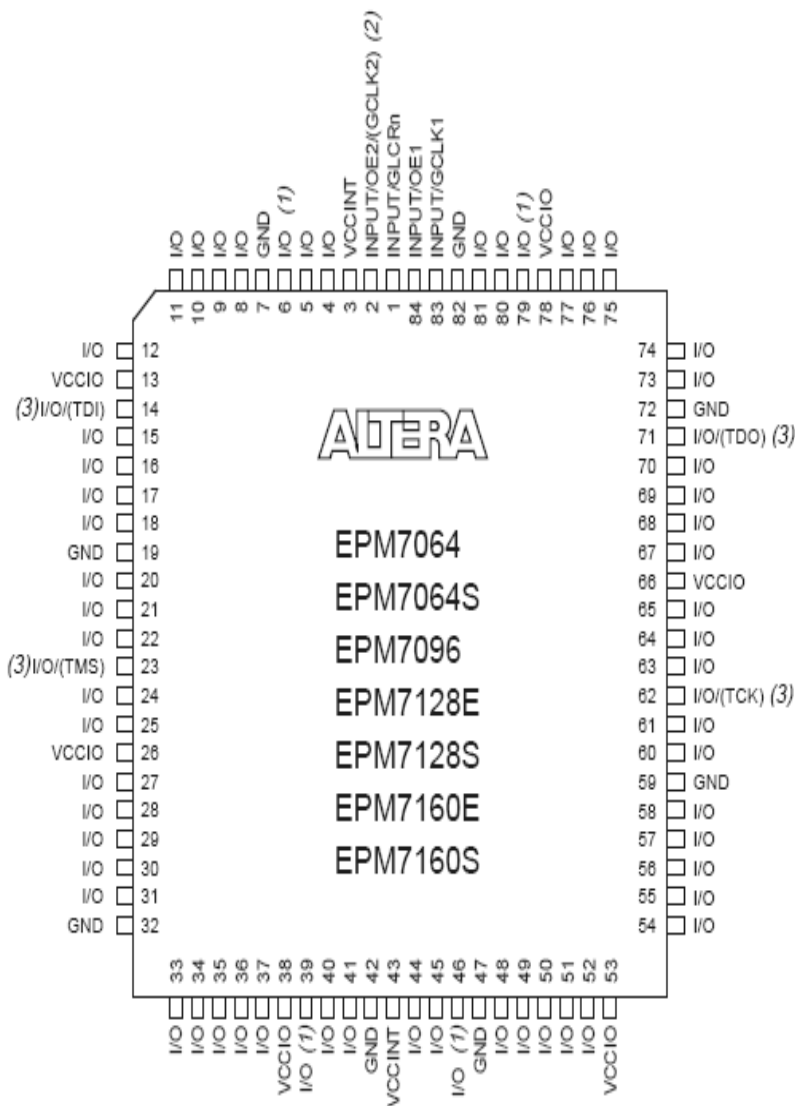
Устройства ввода-вывода лабораторного макета UP1

Электронные компоненты, относящиеся к микросхеме семейства MAX7000

Микросхеме EPM7128S доступны следующие ресурсы:

- установочный разъем («кровать») для корпуса **PLCC** с 84 контактами;
- разъем **JTAG** подключения загрузочного кабеля **ByteBlaster**;
- два оперативных нефиксируемых кнопочных переключателя;
- два 8-разрядных задающих DIP-переключателя уровней «0» и «1»;
- контрольные контактные гнезда сигнальной шины;
- 16 светодиодов сигнальной шины;
- знакосинтезирующий индикатор, включающий два семисегментных знакоместа;
- выход тактового генератора (25,175 МГц);
- разъем расширения с 42 контактами ввода/вывода и контакты глобальных синхросигналов **CLR**, **OE1**, и **OE2/GCLK2**.

- два кнопочных нефиксируемых переключателя **MAX_PB1** и **MAX_PB2** соединены с контактами на плате. При нажатии переключатели замыкают указанные контакты на общий провод через резистор 10 кОм. Переключатели могут быть подключены к контактам микросхемы **ЕРМ7128S** соединительными проводами;
- задающие DIP-переключатели **MAX_SW1** и **MAX_SW2** содержат по восемь переключателей, которые позволяют задавать сигналы логического уровня на связанных с ними контактах. Разомкнутое состояние выключателя соответствует логической «1», замкнутое – логическому «0». Переключатели могут быть подключены к контактам микросхемы **ЕРМ7128S** соединительными проводами;
- плата **UP1** содержит 16 контрольных гнезд и 16 светодиодов (рис. П.2.3). Ток через светодиоды задается резисторами номиналом 330 Ом. Зажигание каждого светодиода осуществляется логическим «0». Светодиоды D1, D2, ..., D16 соединены, соответственно, с контрольными гнездами 1, 2, ..., 16;
- Знакосинтезирующий индикатор микросхемы **ЕРМ7128S** обозначен надписью **MAX_DIGIT**. Индикатор включает два знакоместа, выполненных на семисегментных индикаторах с общим катодом. Соответственно зажигание сегментов осуществляется логическим "0". Обозначение и подключение сегментов индикаторов приведено на рис. П.2.4.



84-Pin PLCC

Рис. П.2.1. Расположение выводов EPM7128S

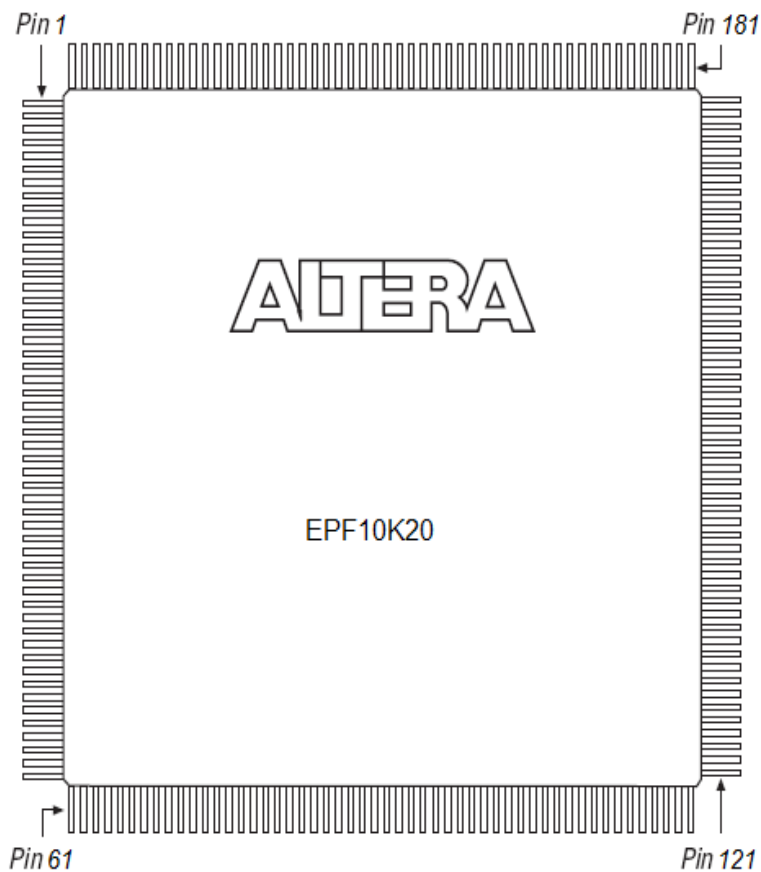


Рис. П.2.2. Расположение выводов EPF10K20

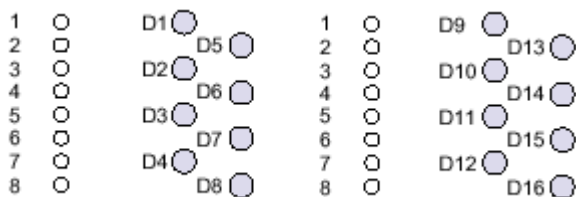


Рис. П.2.3. Расположение контрольных гнезд и светодиодов на плате UPI

Цоколевка ПЛИС EPF10K20

Pin Name	240-Pin PQFP/RQFP
MSEL0 (2)	124
MSEL1 (2)	123
nSTATUS (2)	60
nCONFIG (2)	121
DCLK (2)	179
CONF_DONE (2)	2
INIT_DONE (3)	26
nCE (2)	178
nCEO (2)	3
nWS (4)	238
nRS (4)	236
nCS (4)	240
CS (4)	239
RDYnBSY (4)	23
CLKUSR (4)	11
DATA7 (4)	190
DATA6 (4)	188
DATA5 (4)	186
DATA4 (4)	185
DATA3 (4)	183
DATA2 (4)	182
DATA1 (4)	181
DATA0 (2), (5)	180

TDI (2)	177
TDO (2)	4
TCK (2)	1
TMS (2)	58
TRST (2)	59
Dedicated Inputs	90, 92, 210, 212
Dedicated Clock Pins	91, 211
DEV_CLRn (3)	209
DEV_OE (3)	213
VCCINT	5, 16, 27, 37, 47, 57, 77, 89, 96, 112, 122, 130, 140, 150, 160, 170, 189, 205, 224
VCCIO	(16)
GNDINT	10, 22, 32, 42, 52, 69, 85, 93, 104, 125, 135, 145, 155, 165, 176, 197, 216, 232
GNDIO	—
No Connect (N.C.) (7)	—
Total User I/O Pins (8)	189

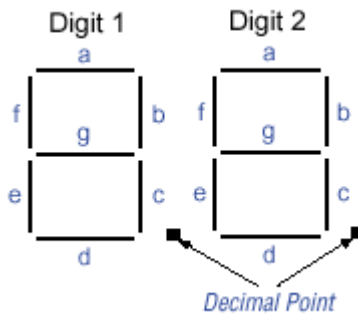


Рис. П.2.4. Знакосинтезирующий индикатор микросхемы EPM7128S

Приложение 3

Описание и характеристики отечественных ПЛИС 5576XC1T, 5576XC1T1

На рис. П.3.1 показана первая отечественная современная ПЛИС на 50 тыс. вентилей типа 5576XC1T, имеющая следующие характеристики:

- питание микросхемы $3,3 \pm 0,3$ В;
- 5-вольтовая толерантность входов/выходов;
- встроенный блок тестирования (Joint Test Action Group, JTAG) с использованием периферийного сканирования схемы (BST), совместимый с IEEE Std. 1149.1-1990;
- встроенная система реконфигурации (ICR) осуществляется посредством внешних конфигурационных устройств, интеллектуальным контроллером или JTAG портом;
- 240-выводной металлокерамический корпус для поверхностного монтажа.



Рис. П.3.1. ПЛИС на 50 тыс. вентилей
типа 5576XC1T

Основные функциональные параметры ПЛИС 5576XC1T1:

<i>Параметр</i>	<i>Значение</i>
Типовая логическая емкость, вент.	50000
Количество логических элементов	2880
Количество логических блоков	360
Емкость встроенной памяти, бит	20480
Количество триггеров	3184
Количество пользовательских выводов	189

Эксплуатационные характеристики:

Диапазон рабочих температур, °C	от минус 60 до плюс 85
Частота переключения 8-разрядного счетчика, МГц	> 100
Выходное напряжение низкого уровня, В (UCC = 3 В, Iol=12мА)	< 0,45
Выходное напряжение высокого уровня, В (UCC =3В, Ioh=-4мА)	> 2,4
Входное напряжение, В, не более	5,5
Ёмкость нагрузки, пФ	< 100

ПЛИС с повышенной нагрузочной способностью 5576XC2T

- питание микросхемы $3,3 \pm 0,3$ В;
- 5-вольтовая толерантность входов-выходов;
- встроенный блок тестирования (Joint Test Action Group, JTAG) с использованием периферийного сканирования схемы (BST), совместимый с IEEE Std. 1149.1-1990;
- встроенная система реконфигурации (ICR) осуществляется посредством внешних конфигурационных устройств, интеллектуальным контроллером или JTAG портом;
- программируемая нагрузочная способность элементов ввода/вывода (12-24 мА);
- 108-выводной металлокерамический корпус для поверхностного монтажа.

Основные функциональные параметры

<i>Параметр</i>	<i>Значение</i>
Типовая логическая емкость, вент.	2500
Количество логических элементов	208
Количество логических блоков	26
Количество триггеров	272
Количество пользовательских выводов	68

Конфигурационное ПЗУ 5576PC1Y для ПЛИС 5576XC1T, 5576XC1T1

Специально для этих ПЛИС серии 5576 фирма «Миландр» выпустила конфигурационное ПЗУ (Flash EEPROM) 5576PC1Y (рис. П.3.1). Ближайший аналог фирмы **Altera** – EPC4. Емкость этого устройства в аналогичном корпусе составляет 4 Мбит.



*Рис. П.3.2. Конфигурационное ПЗУ
(Flash EEPROM) 5576PC1Y*

Приложение 4

Установка на ПК пакета MAX+plus II

Для разработки схем и устройств на ПЛИС рассмотрим, как установить на компьютере пакет MAX+Plus II фирмы Altera.

Для загрузки пакета выполните следующие операции:

- зайдите на веб-сайт компании Altera (<http://www.altera.com/>);
- в верхнем меню выберите пункт Support → Downloads;
- в меню слева выберите самый нижний пункт в последнем подменю Legacy Software – MAX+Plus II;
- выберите пункт MAX+Plus II Student Edition (студенческая версия);
- нажмите на ссылку student102.exe. Несмотря на то, что данная программа является бесплатной, для её загрузки с сайта разработчика необходима регистрация. Однако предусмотрена возможность однократного доступа без необходимости регистрации. Unlicensed Features Unlicensed Features Для этого поставьте переключатель в группе “Don't have an account?” в положение Get One-Time Access и укажите свой адрес электронной почты. Галочку под полем ввода можно не ставить. Нажмите кнопку “Get One-Time Access”;
- сохраните загруженный файл в удобном Вам месте (например на диске C);

Для установки САПР MAX+Plus II выполните следующие шаги:

- запустите на выполнение загруженный ранее файл student102.exe, нажмите кнопку Next;
- на экране появится лицензионное соглашение, которое необходимо прочитать и принять, нажав кнопку Yes;
- нажмите кнопку Next. Наберите в окне своё имя и название организации, например, «Вася Иванов», “НИЯУ МИФИ”;
- нажав на кнопку Next, на экране можно выбрать компоненты программы, которые необходимо устанавливать, а также путь к папке, в которую устанавливается программа;

- рекомендуется оставить все галочки установленными, а путь к папке установки можно при необходимости откорректировать с помощью кнопки Browse;
- затем на экране выясняется необходимость установки программы MAX+Plus II Tutorial (учебник для пользователей), рекомендуется оставить галочку установленной и указать путь и каталог размещения учебника на личном ПК;
- нажмите кнопку Next два раза, после чего начнётся установка программы;
- по завершении установки в меню “Пуск” появится пункт «MAX+Plus II Student Edition»;
- запустите программу выбором пункта «MAX+PlusII Baseline» в меню Пуск → Программы → MAX+Plus II Student Edition, при первом запуске примите лицензионное соглашение, а во втором окне ответьте No.

После установки пакета необходимо получить и загрузить лицензию MAX+Plus II фирмы Altera для того, чтобы разблокировать весь необходимый для работы пакета набор программ. Для этого выполните следующие операции:

- зайдите на страницу сайта фирмы Altera <http://www.altera.com/download/licensing/lic-university.html>;
- выберите пункт «Version 10.2, 10.1 or 9.23» и нажмите Continue;
- введите в поле «Enter your hard disk volume serial number» серийный номер тома C:, который можно посмотреть, открыв пакет MAX+Plus II, войдя в меню Options → License Setup и нажав кнопку System Info (рис. П.4.1).

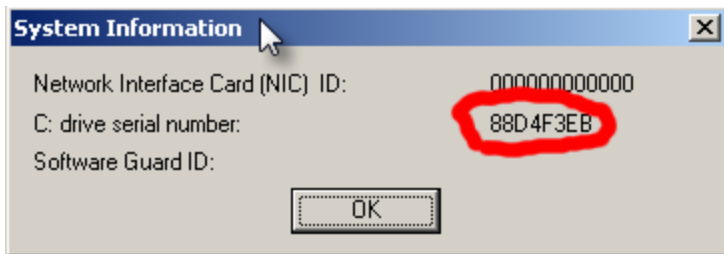


Рис. П.4.1. Местоположение серийного номера тома диска (C:), необходимого для получения лицензии

Два раза нажмите кнопку Continue, пока не появится форма регистрации.

ModelSim-Altera

Resource Centers

- Overview
- Installation & Licensing
- Scripting
- Board Design & I/O
- Design Entry & Planning
- Synthesis & Netlist Viewers
- Incremental Compilation
- Optimisation
- Power Management
- TimeQuest Timing Analyzer
- Classic Timing Analyzer
- Simulation & Verification
- On-Chip Debugging
- HardCopy Design
- EDA Tool Support

Software Resources

- OS Support
- Driver Installation

Download & Licensing

- Download
- Licensing

Please complete the information in the box below and click the **Continue** button.

You didn't enter your first name. Please enter it here:
Kirill

You didn't enter your last name. Please enter it here:
Sarkyan

You didn't enter your address. Please enter it here:
dot@kkursor.ru

You didn't enter your city. Please enter it here:
Moscow

You didn't enter your state. Please enter it here:
-

You didn't enter your country. Please enter it here:
Russian Federation

You didn't enter your zip code. Please enter it here:
1

You didn't enter your phone number. Please enter it here:
+7 905 123 4567

You did not enter a valid email address. You must enter a valid email address to obtain a license file.
Re-enter email address:
dot@kkursor.ru

Continue

Рис. П.4.2. Регистрационная форма на веб-сайте Altera

Заполните регистрационную форму своими данными и нажмите кнопку «Continue». Ответьте на вопросы анкеты и нажмите «Finish». Если система ответила «Thank you for choosing the MAX+PLUS II BASELINE development software.», значит, все данные приняты системой и на указанный на шаге почтой e-mail отправлено письмо с лицензионным файлом license.dat. Сохраните этот файл в удобном месте (например в папке C:\MaxPlus2). Запустите MAX+Plus II. Зайдите в меню Options → License Setup. Нажмите кнопку Browse и найдите файл license.dat, полученный по вашей почте (например, e-mail). Нажмите «OK». Активация лицензии завершена, если содержимое списков программ из правого окна Unlicensed Features перейдет в левое окно Licensed Features (рис. П.4.3).

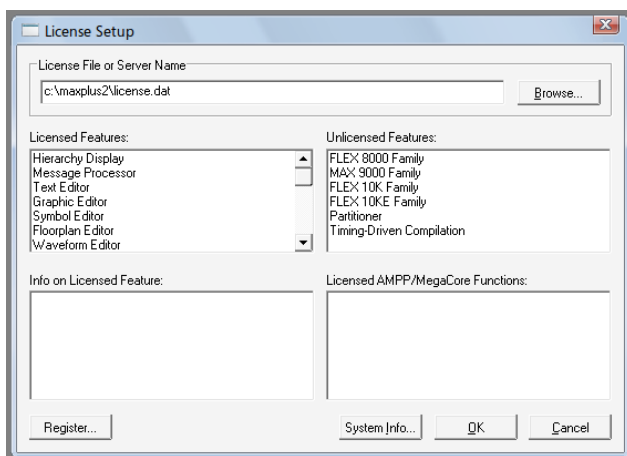


Рис. П.4.3. Меню установки лицензионных файлов License Setup

На этом установка и настройка пакета MAX+PLUS II завершена и можно приступить к работе с ним.

*Владимир Петрович Петропавловский
Станислав Геннадиевич Микульский
Кирилл Артемович Сарксян*

ЛАБОРАТОРНЫЙ ПРАКТИКУМ

«ПРОЕКТИРОВАНИЕ ЦИФРОВЫХ УСТРОЙСТВ НА ПРОГРАММИРУЕМЫХ ЛОГИЧЕСКИХ ИНТЕГРАЛЬНЫХ СХЕМАХ»

(Виртуальная микроэлектроника)

Учебное пособие

Редактор Е.Г. Станкевич

Вёрстка книги полностью соответствует оригинал-макету,
предоставленному авторами.

Подписано в печать 27.08.2012
Печ. л. 6,75. Уч.-изд. л. 6,75

Формат 60x84 1/16.
Тираж 120 экз.

Изд. № 016-1.

Заказ №

Национальный исследовательский ядерный университет «МИФИ»,
Типография НИЯУ МИФИ. 115409, Москва, Каширское шоссе, 31