

621.382.82(07)

Р 851

№ 2917

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ



**Таганрогский государственный
радиотехнический университет**

РУКОВОДСТВО
К ЛАБОРАТОРНОЙ РАБОТЕ

**ПРОЕКТИРОВАНИЕ
СПЕЦИАЛИЗИРОВАННЫХ СБИС НА ОСНОВЕ ПЛИС
С ИСПОЛЬЗОВАНИЕМ САПР MENTOR GRAPHICS**

Для студентов специальностей 2205, 2002, 2008, 2201, 2203 и
направлений 55.07.00, 55.11.00



**БАЗОВАЯ УЧЕБНО-НАУЧНАЯ ЛАБОРАТОРИЯ НКБ ВС И
УНТЦ ФЭП "ПРОЕКТИРОВАНИЕ ЭЛЕКТРОННЫХ УСТРОЙСТВ"**

Таганрог 2000

УДК 621.382.82-181.2(076.5)

Составители: Б.Г. Коноплев, И.И. Итенберг, С.В. Солодовников, Е.А. Рындин,
А.В. Ковалев, И.Е. Лысенко

Руководство к лабораторной работе "Проектирование специализированных СБИС на основе ПЛИС с использованием САПР Mentor Graphics".
Таганрог: Изд-во ТРТУ, 2000. 43 с.

Цикл лабораторных работ по освоению студентами методов проектирования электронной аппаратуры с использованием рабочей станции Hewlett Packard подготовлен совместно сотрудниками ФЭП и НКБ ВС ТРТУ для базовой учебно-научной лаборатории "Проектирование электронных устройств".

В данной работе излагаются сведения, необходимые для проектирования специализированных СБИС на основе ПЛИС фирмы Altera (MAX 3000, MAX 7000, MAX 9000, FLEX 6000, FLEX 8000, FLEX 10K) с использованием программы MAX+plus II САПР Mentor Graphics. Описаны маршруты проектирования СБИС на основе ПЛИС, включая ввод электрических схем, моделирование, подготовку файлов для программирования ПЛИС.

Ил. 30. Библиогр.: 3 назв.

Рецензент В.Г.Ивченко, канд. техн. наук, доцент кафедры КЭС ТРТУ.

**Коноплев Борис Георгиевич
Итенберг Игорь Ильич
Солодовников Сергей Валерьевич
Рындин Евгений Адальбертович
Ковалев Андрей Владимирович
Лысенко Игорь Евгеньевич**

**Руководство
к лабораторной работе**

**Проектирование
специализированных СБИС на основе ПЛИС
с использованием САПР Mentor Graphics**

Ответственный за выпуск Рындин Е.А.
Редактор Васютина О.К.
Корректор Проценко И.А.

ЛР 020565 от 23 июня 1997 г.
Печать офсетная
Формат 60 * 84/16
Усл. п. л. – 2,7
Заказ N

Подписано к печати
Бумага офсетная
Уч.- изд. л. – 2,5
Тираж 150 экз.

“С”

Издательство Таганрогского государственного
радиотехнического университета
ГСП 17А, Таганрог, 28, Некрасовский, 44
Типография Таганрогского государственного
радиотехнического университета
ГСП 17А, Таганрог, 28, Энгельса, 1

ВВЕДЕНИЕ

Достижения в области микроэлектронных технологий привели к тому, что основу многих современных радиоэлектронных и вычислительных устройств составляют специализированные большие и сверхбольшие интегральные схемы (БИС и СБИС), позволяющие значительно улучшить технико-экономические характеристики аппаратуры конкретного назначения [1].

На практике используют пять способов реализации специализированных СБИС:

- 1) полностью заказные - предполагающие полный цикл проектирования, включающие разработку всех литографических шаблонов на уровне отдельных областей транзисторных структур;
- 2) заказные на основе библиотечных элементов - предполагающие использование заранее разработанных топологических библиотек элементов, узлов и блоков различной сложности и включающие разработку всех литографических шаблонов, но на уровне элементов, узлов и блоков;
- 3) полузаказные на основе базовых матричных кристаллов (БМК) - предполагающие использование заранее изготовленных «полуфабрикатов» - БМК - кристаллов с матрицами так называемых базовых ячеек, каждая из которых содержит набор нескоммутированных элементов (транзисторов, диодов, резисторов и др.), позволяющих посредством разработки только заказных шаблонов металлизации соединить базовые элементы в соответствии с проектируемой схемой для выполнения заданного набора функций;
- 4) на основе постоянных запоминающих устройств (ПЗУ) - предполагающие программирование заранее изготовленных микросхем ПЗУ, содержащих полный дешифратор и программируемый шифратор (поле ячеек памяти), для выполнения определенных функций;
- 5) на основе программируемых логических матриц (ПЛМ) - также предполагающие программирование заранее изготовленных микросхем, содержащих ПЛМ, отличающихся от ПЗУ наличием программируемых дешифраторов, что обеспечивает дополнительную гибкость.

Каждый из перечисленных способов имеет достоинства и недостатки, определяющие область его практического использования. Так, полностью заказные СБИС отличаются наиболее высокими показателями эффективности использования площади кристалла, быстродействия, надежности, но при этом предполагают длительный цикл проектирования (несколько месяцев работы проектного коллектива) и имеют наиболее высокую стоимость. Напротив, СБИС на основе ПЛМ позволяют сократить цикл проектирования до нескольких часов и свести к минимуму затраты на проектирование за счет определенной избыточности в числе логических матриц, длине связей и т.д., что несколько снижает качественные показатели.

Следует отметить, что переход к субмикронным топологическим размерам элементов СБИС (до 0,18 мкм) привел к резкому улучшению характеристик СБИС, в том числе и на основе ПЛМ. При этом степень интеграции, быстродействие, помехоустойчивость и надежность программируемых интегральных схем достигли столь высокого уровня, что в настоящее время во всем мире наблюдается тенденция все более широкого их использования для реализации специализированных СБИС.

Современные микросхемы на основе ПЛМ представляют собой сложные комплексы, включающие блоки ПЛМ, специальные встроенные блоки, мультиплексируемые линии межсоединений, блоки ОЗУ и ПЗУ, дешифраторы, устройства ввода-вывода, преобразователи питающих напряжений и др. Поэтому они получили более емкое название - программируемые логические интегральные схемы (ПЛИС). На рис. 1 показан внешний вид, а на рис. 2 схематически приведена топологическая организация одной из последних разработок фирмы Altera - ПЛИС серии FLEX 10K [2].



Рис. 1. Внешний вид Altera FLEX 10K

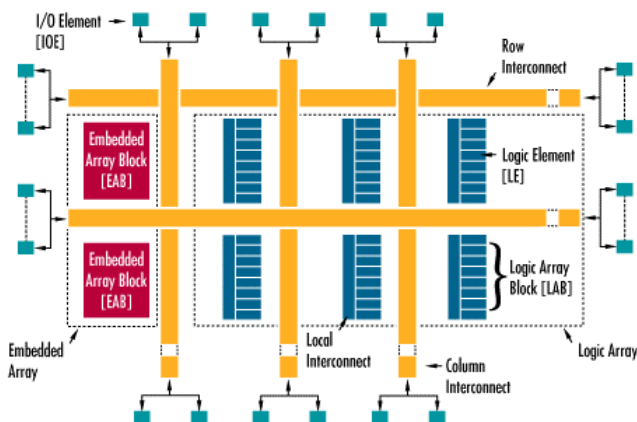


Рис. 2. Топологическая организация ПЛИС серии FLEX 10K:
IOE - элемент ввода-вывода; EAB - встроенный блок; LE - логический элемент; LAB - блок ПЛМ; Row, Column, Local Interconnect - горизонтальные, вертикальные и локальные линии связи

В табл. 1 приведены основные рабочие характеристики некоторых ПЛИС фирмы Altera (серий FLEX 10K и APEX 20K) [2].

Таблица 1

Характеристики ПЛИС фирмы Altera

Параметр	FLEX 10K	FLEX 10KA	FLEX 10KE	APEX 20K	APEX 20KE
Проектная норма, мкм	0,5	0,35	0,25	0,22	0,18
Напряжение питания, В	5	3,3	2,5	2,5	1,8
Макс. Число транзисторов	250 000	250 000	250 000	1 052 000	2 670 000
Макс. число логических элементов	12 160	12 160	12 160	16 640	42 240
Макс. число внешних выводов	-	-	-	502	756

В табл. 2 приведены данные по максимальному быстродействию ПЛИС конкурирующих фирм - мировых лидеров в данной области [2].

Таблица 2

Максимальное быстродействие ПЛИС, производимых фирмами Altera, Xilinx, Lucent

Параметр	Altera FLEX 10K	Xilinx XC4000E-2	Lucent OR2C15A-4S208
Максимальная рабочая частота, МГц	125	52	89

Разработка специализированной СБИС на основе ПЛИС невозможна без систем автоматизированного проектирования (САПР), опирающихся не только на персональные компьютеры, но и на более мощную вычислительную

базу - рабочие станции.

Предлагаемый курс лабораторных работ посвящен изучению подсистемы автоматизированного проектирования специализированных СБИС на основе ПЛИС MAX+plus II, являющейся составной частью одной из самых современных САПР для рабочих станций - Mentor Graphics.

1. ОБЩИЕ СВЕДЕНИЯ О СИСТЕМЕ MAX+plus II

Процедуру разработки нового проекта от концепции до завершения можно упрощенно представить следующим образом [2, 3]:

- 1) создание нового файла или иерархической структуры нескольких файлов проекта с помощью любого сочетания редакторов в системе MAX+plus II, то есть графического, текстового и сигнального редакторов;
- 2) задание имени файла – проекта верхнего уровня в качестве имени проекта;
- 3) назначение семейства ПЛИС для проекта;
- 4) открытие окна компилятора Compiler и выбор кнопки Start для начала компиляции проекта. По желанию пользователя можно подключить модуль извлечения временных параметров проекта Timing SNF Extractor для создания файла, используемого при временном моделировании;
- 5) в случае успешной компиляции возможен временной анализ, для чего следует выполнить следующее:
 - для проведения анализа задержек открыть окно Timing Analyzer, выбрать режим анализа и нажать кнопку Start;
 - для проведения моделирования нужно сначала создать текстовый вектор в файле (.scf), пользуясь сигнальным редактором, или в файле вектора (.vec), пользуясь текстовым редактором. Затем открыть окно отладчика Simulator и нажать кнопку Start;
- 6) открытие окна программатора Programmer с последующим выбором одного из двух способов: использование программатора Master Programming Unit (MPU) или подключение загрузочных устройств BitBlaster, ByteBlaster или FLEX Download Cable к устройству, программируемому в системе;
- 7) выбор кнопки Program для программирования устройств с памятью типа EPROM или EEPROM (электрически перепрограммируемых ПЗУ) либо выбор кнопки Configure для конфигурации устройства с памятью типа SRAM (статического ОЗУ).

Система MAX+plus II содержит 11 приложений и главную управляющую программу. Различные приложения, обеспечивающие создание проекта, могут быть активизированы мгновенно, что позволяет пользователю переключаться между ними щелчком мыши или с помощью команд меню. В это же

время может работать одно из фоновых приложений, например компилятор, программа моделирования, анализатор синхронизации и программатор. Одни и те же команды разных приложений работают одинаково, что облегчает задачу разработки логического дизайна.

Перед тем как начать работать в системе MAX+plus II, следует понять разницу между файлами проекта, вспомогательными файлами и проектами.

Файл проекта – это графический, текстовый или сигнальный файл, созданный с помощью графического или сигнального редакторов системы MAX+plus II или в любом другом, использующем промышленные стандарты, схемном или текстовом редакторе либо при помощи программы netlist writer, имеющейся в пакетах, поддерживающих стандартные форматы описания аппаратуры EDIF (Electronic Design Interchange Format), VHDL (Very High Speed Integrated Circuit Hardware Description Language) и Verilog HDL. Этот файл содержит функциональное описание проекта MAX+plus II и обрабатывается компилятором.

Вспомогательные файлы — это файлы, связанные с проектом MAX+plus II, но не являющиеся частью его иерархического дерева. Большинство таких файлов не содержит функционального описания проекта. Некоторые из них создаются автоматически приложением системы MAX+plus II, другие — пользователем. Примерами вспомогательных файлов являются файлы назначений и конфигурации (.acf), символьные файлы (.sym), файлы отчета (.rpt) и файлы тестовых векторов (.vec).

Проект состоит из всех файлов иерархической структуры проекта, в том числе вспомогательных и выходных файлов. Именем проекта является имя файла проекта верхнего уровня без расширения. Система MAX+plus II выполняет компиляцию, тестирование, анализ синхронизации и программирование сразу целого проекта, хотя пользователь может в это время редактировать файлы этого проекта в рамках другого проекта. Для каждого проекта желательно создавать отдельный подкаталог в рабочем каталоге системы MAX+plus II.

В системе MAX+plus II легко доступны все инструменты для создания проекта. Разработка проекта ускоряется за счёт имеющихся стандартных функций, в том числе примитивов, мегафункций, библиотеки параметризованных модулей (LPM) и макрофункций устаревшего типа микросхем 74-й серии.

В системе MAX+plus II есть три редактора для разработки проекта: графический, текстовый и сигнальный, а также два вспомогательных редактора: редактор базового плана кристалла и символьный редактор.

В иерархической структуре проекта на любом уровне допускается смешанное использование файлов, созданных в различных редакторах системы MAX+plus II.

2. РЕДАКТОРЫ СИСТЕМЫ MAX+plus II

Система MAX+plus II включает в себя четыре редактора создания проекта: графический редактор (Graphic Editor), символьный редактор (Symbol Editor), текстовый редактор (Text Editor) и сигнальный редактор (Waveform Editor). Данные редакторы имеют следующие общие функции: создание, сохранение и открытие файла; вывод данных на печать; поиск узлов; поиск и замена текста; отмена последнего шага редактирования, его возвращения; удаление, копирование, вставка выделенных фрагментов; обмен фрагментами между приложениями системы MAX+plus II или приложениями Windows; всплывающие окна меню [2, 3].

Графический редактор позволяет создавать проекты на основе библиотек УГО (рис.3). Созданные в данном редакторе проекты могут содержать любую комбинацию УГО, мегафункции и макрофункции. Графический редактор поддерживает следующие форматы файлов: .gdf и .sch.

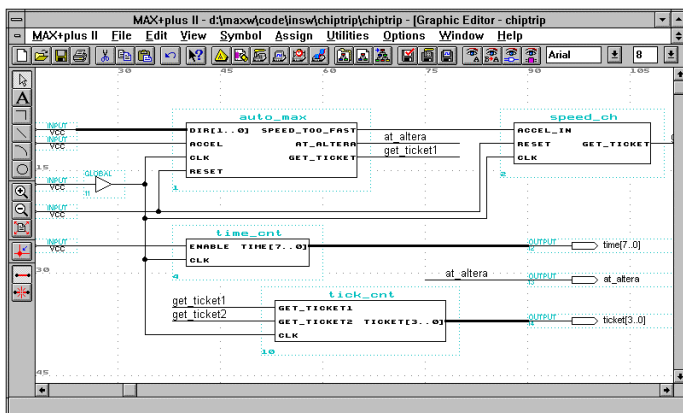


Рис.3. Рабочее окно графического редактора

Графический редактор позволяет увеличивать или уменьшать масштаб изображения проекта на экране, выбирать размер шрифта, задавать стили линий, получать зеркальное отображение, поворачивать выделенные фрагменты на 90, 180 и 270 градусов, задавать размер и ориентацию текущего листа схемы.

Символьный редактор позволяет создавать и редактировать УГО (символ). Символьный файл имеет то же имя, что и проект, с расширением .sym.

Вызов редактора осуществляется по команде Create Default Symbol в меню File (рис.4).

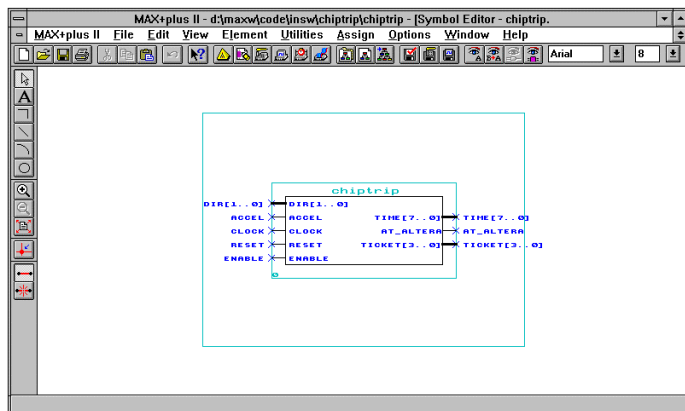


Рис.4. Рабочее окно символьного редактора

Символьный редактор позволяет переопределять символ, представляющий файл проекта, создавать и редактировать выходы и их имена, используя входные, выходные и двунаправленные выходы, а также задавать значения параметров и устанавливать их по умолчанию, вводить комментарии.

Текстовый редактор позволяет создавать текстовые файлы проекта на языках описания аппаратуры: AHDL (Altera Hardware Description Language) – .tdf, VHDL – .vhd, Verilog HDL – .v. При этом ввод исходных данных о проекте осуществляется не в виде схемы, выполненной в графическом редакторе (см. рис. 3), а в виде текстового описания проекта, что позволяет приблизить разработку проекта к процессу программирования, повысить наглядность представления информации, сократить вероятность ошибок и сроки проектирования. Более подробно данные возможности будут рассмотрены в лабораторной работе, посвященной разработке VHDL-описаний проектов СБИС.

В текстовом редакторе также можно работать со следующими форматами файлов: .acf, .aco, .adf, .cmd, .edc, .edf, .fit, .hst, .lmf, .log, .mif, .mio, .mtf, .plf, .rpt, .sdo, .smf, .tao, .tdf, .tdo, .tdx, .tff, .vec, .vho, .vmo, .vo, .xnf и с произвольным файлом формата ASCII.

Данный текстовый редактор имеет следующие встроенные функции: ввод файла проекта, их компиляции и отладки с выдачей сообщения об ошибках и их локализацией в исходном тексте или в тексте вспомогательных файлов. Кроме того, данный редактор содержит шаблоны языковых конструкций

для AHDL, VHDL и Verilog HDL. В текстовом редакторе можно редактировать файлы конфигурации, а также делать установки для компилятора, программы моделирования и временного анализатора (рис.5).

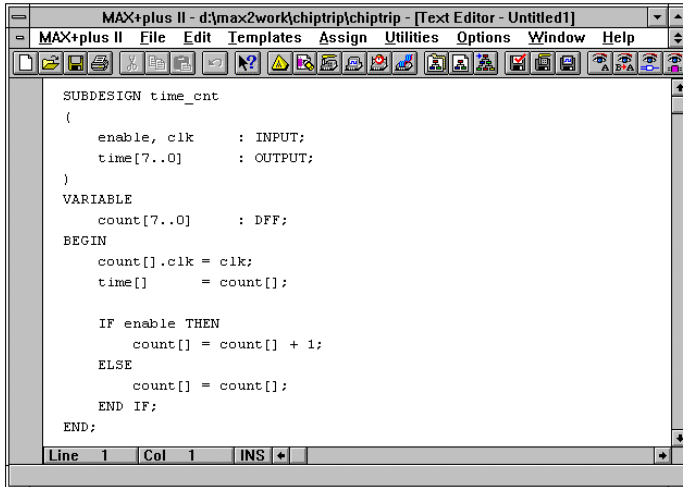


Рис.5. Рабочее окно текстового редактора

Сигнальный редактор позволяет создавать описание проекта, вводить тестовые векторы и просматривать результаты тестирования (рис.6). Данный редактор поддерживает формат файлов сигнальных проектов .wdf, содержащих временные диаграммы, а также формат файлов тестирования .scf, содержащих входные векторы для функциональной отладки.

Сигнальный редактор является альтернативой графического или текстового редакторов. С его помощью можно графическим способом задавать комбинации входных и выходных логических уровней. Созданный файл формата .wdf содержит логические входы, выходы комбинаторной логики, счетчиков и т.д.

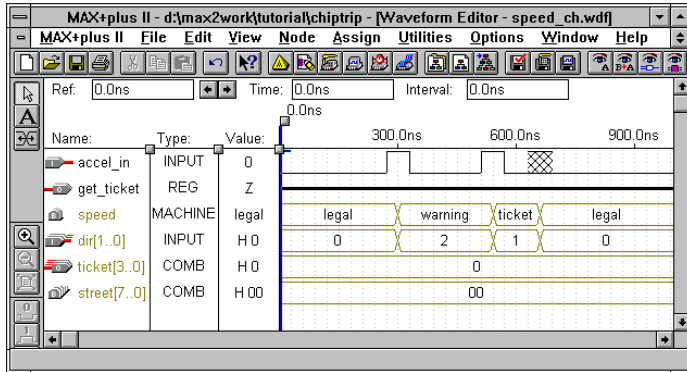


Рис.6. Рабочее окно сигнального редактора

Сигнальный редактор системы MAX+plus II позволяет выполнять следующие встроенные функции: редактировать временные диаграммы целиком или частично; создавать и редактировать временные диаграммы отдельных узлов. Позволяет создавать группы содержащие от 2 до 256 узлов, и объединять их с уже существующими; отображать значение группы в двоичной, десятичной, шестнадцатеричной или восьмеричной системе счисления с преобразованием или без в код Грея; копировать, вставлять, удалять и перемещать выбранную часть или весь сигнал как в пределах одной группы, так и между группами. Можно инвертировать, вставлять, переписывать, повторять, расширять или сжимать интервал сигнала с любым логическим уровнем, тактовым сигналом, последовательностью счета или именем состояния; задавать сетку для выравнивания переходов между логическими уровнями; вводить комментарии в любом месте файла проекта; изменять масштаб отображения проекта. Для облегчения тестирования можно делать наложение любых выводов как одного файла, так и нескольких файлов для сравнения сигналов.

Для отображения иерархической структуры файлов проекта в виде дерева с ветвями, представляющими собой подпроекты, система MAX+plus II содержит дисплей иерархии (Hierarchy Display). В иерархии текущего проекта отображается имя и иконка файла для каждого подпроекта. Кроме того, дисплей иерархии показывает вспомогательные файлы, связанные с текущей иерархией (рис.7) [2].

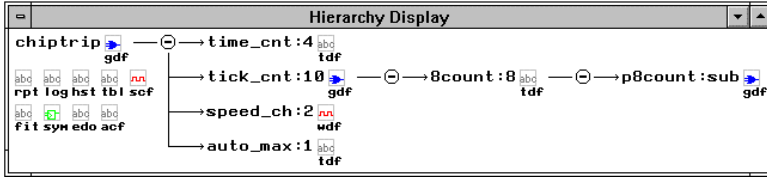


Рис. 7. Дисплей иерархии

Для назначения ресурсов физических устройств и просмотра результатов разводки, выполненных компилятором, система MAX+plus II содержит редактор базового плана кристалла (Floorplan Editor) (рис.8).

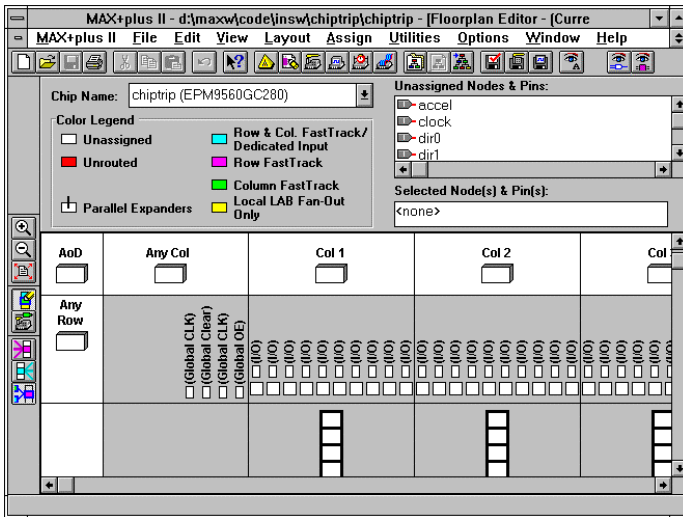


Рис.8. Рабочее окно редактора базового плана кристалла

Редактор базового плана кристалла может представлять два типа изображения [2, 3]:

- 1) Device View, показывающий все контакты устройства и их функции;
- 2) LAB View, показывающий внутреннюю часть устройства, в том числе все логические структурные блоки и отдельные логические элементы.

Для компиляции созданных проектов система MAX+plus II содержит компилятор (Compiler) (рис.9). При компиляции проекта извлекается информация об иерархических связях между файлами проекта и производится проверка на ошибки ввода описания проекта. Если проект слишком большой для

реализации в одном устройстве, компилятор автоматически разбивает его на части для реализации в нескольких устройствах того же самого семейства, при этом число соединений между устройствами минимизируется. Способ реализации проекта отражается в файле отчета компилятора .grf.

Компилятор может автоматически обрабатывать следующие файлы проекта: графические файлы проекта (.gdf); текстовые файлы проекта на языке AHDL (.tdf); сигнальные файлы проекта (.wdf); файлы проекта на языке VHDL (.vhd); файлы проекта на языке Verilog (.v); схемные файлы OrCAD (.sch); входные файлы EDIF (.edf); файлы формата Xilinx Netlist (.xnf); файлы проекта Altera (.adf); файлы цифрового автомата (.smf).

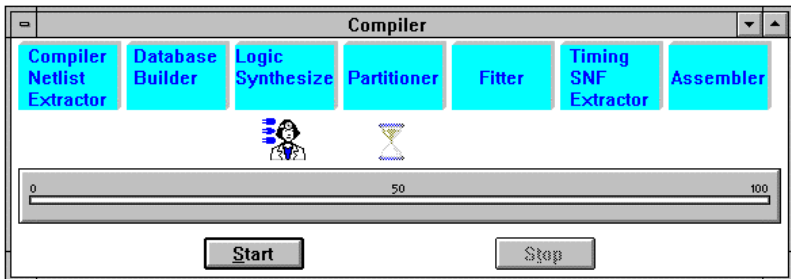


Рис.9. Рабочее окно компилятора

Компилятор системы MAX+plus II позволяет создавать файлы, используемые программатором одного или нескольких устройств для программирования ПЛИС. Кроме того, разработчик проекта может сам задавать стиль логического синтеза проекта по умолчанию и другие параметры логического синтеза в рамках всего проекта, что позволяет производить логический синтез в соответствии с потребностями разработчика. Кроме того, разработчик может ввести требования по синхронизации в рамках всего проекта, точно задать разбиение большого проекта на части для реализации в нескольких устройствах и выбрать варианты параметров применяемых устройств.

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 (рис.10).

Для тестирования логических операций и внутренней синхронизации проекта в системе MAX+plus II содержится программа моделирования (Simulator) (рис.11). Она позволяет разработчику моделировать проект прежде, чем он будет реализован в устройстве, что существенно сокращает время разработки проекта. Кроме того, разработчик может производить моделиро-

вание проекта независимо от количества используемых устройств, требуемых для его реализации.

Для анализа исполнения проекта в системе MAX+plus II используется временной анализатор (Timing Analyzer) (рис.12). Данный анализатор позволяет разработчику анализировать работу проектируемой логической цепи после того, как она была синтезирована и оптимизирована компилятором. Разработчик может исследовать все пути прохождения сигналов в проектируемой логической цепи, определить критические задержки в цепях.

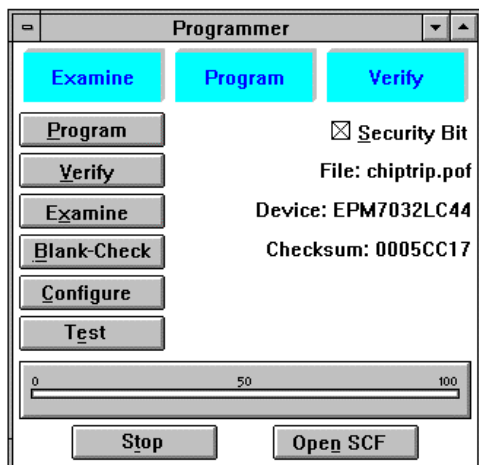


Рис.10. Рабочее окно программатора

Для выдачи на экран сообщений об ошибках, предупреждающих и информационных сообщений, в системе MAX+plus II используется генератор сообщений (Message Processor). Генератор сообщений взаимодействует со всеми приложениями системы MAX+plus II, что позволяет определять и корректировать ошибки, получать предупреждающие сообщения и подсказки в любой момент времени (рис.13).

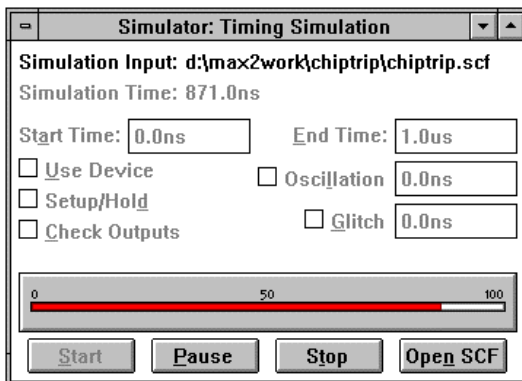


Рис.11. Рабочее окно программы моделирования

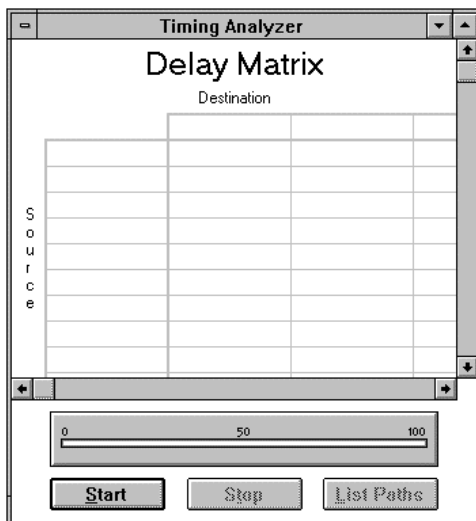


Рис.12. Рабочее окно временного анализатора

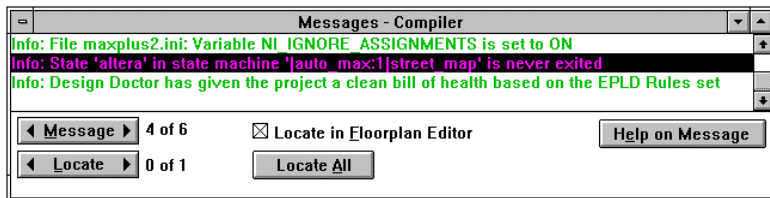


Рис.13. Рабочее окно генератора сообщений

3. КОМПИЛЯЦИЯ ПРОЕКТА В СИСТЕМЕ MAX+plus II

Система MAX+plus II позволяет вводить, редактировать и удалять типы назначений ресурсов, устройств и параметров управления компиляцией проекта с помощью команд из меню Assign (рис.14) [2].



Рис.14. Меню Assign

Разработчик может выполнять назначения для своего проекта независимо от того, открыт ли какой-нибудь файл проекта или окно приложений. Доступны следующие типы назначений [2, 3]:

- 1) Clique... – задает логические функции, которые должны оставаться вместе в одном и том же блоке логической структуры LAB, блоке ячеек EAB, в одном ряду или устройстве;
- 2) Pin/Location/Chip... – назначает конкретному контакту или нескольким

контактам кристалла вход или выход одной логической функции (назначает единственную логическую функцию конкретной ячейке кристалла) задает логические функции, которые должны быть реализованы в одном и том же устройстве в случае разделения проекта на несколько устройств.

- 3) Probe... – присваивает уникальное имя входу или выходу логической функции;
- 4) Connected Pins... – задает внешнее соединение двух или более контактов на схеме разработчика;
- 5) Local Routing... – присваивает коэффициент разветвления по выходу логическому элементу;
- 6) Device... – назначает тип ПЛИС, на которой реализуется проект;
- 7) Logic Options... – управляет синтезом отдельных логических функций во время компиляции;
- 8) Timing Requirements... – управляет синтезом и подгонкой отдельных логических функций для получения требуемых характеристик для времени задержки вход-неподрегистренный выход (t_{PD}), синхросигнал-выход (t_{CO}), синхросигнал-время установки (t_{SU}) и для частоты синхросигнала (f_{MAX}).

Можно определить глобальные настройки устройства для компилятора, с целью использования их для всех устройств при обработке проекта [2, 3]:

- 1) Global Project Parameters – позволяет задавать имена и глобальные установки, которые могут быть использованы компилятором для параметров всех функций в проекте;
- 2) Global Project Timing Requirements – позволяет ввести глобальные требования по синхронизации для проекта, задавая общие характеристики для времени задержки вход-неподрегистренный выход (t_{PD}), синхросигнал-выход (t_{CO}), синхросигнал-время установки (t_{SU}) и для частоты синхросигнала (f_{MAX});
- 3) Global Project Logic Synthesis – позволяет сделать глобальные установки для компилятора в части логического синтеза проекта;
- 4) Global Project Device Options – позволяет определять настройки выбранного устройства для всех устройств, использованных в текущем проекте.

4. ПРИМЕР ПРОЕКТИРОВАНИЯ СПЕЦИАЛИЗИРОВАННОЙ ИС

В качестве примера рассмотрим маршрут проектирования с помощью программы MAX+plus II ИС четырехразрядного сдвигового регистра (рис. 15) на двухтактных D-триггерах (рис. 16).

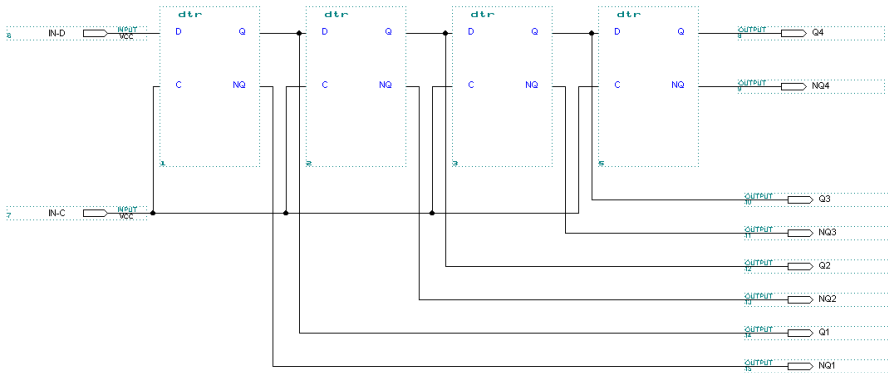


Рис. 15. Схема четырехразрядного сдвигового регистра

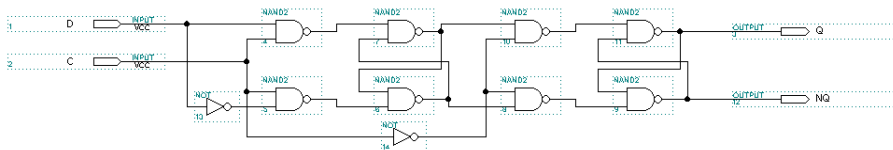


Рис. 16. Схема двухтактного D-триггера

Основные этапы проектирования:

- 1) запуск программы MAX+plus II;
- 2) ввод имени проекта;
- 3) ввод электрической схемы проектируемого устройства;
- 4) ввод входных сигналов для функционально-логического моделирования;
- 5) выбор типа ПЛИС для реализации проекта;
- 6) проверка корректности ввода электрической схемы;
- 7) компиляция схемы проекта;
- 8) функционально-логическое моделирование проектируемого устройства;
- 9) расчет задержек прохождения сигналов со входов на выходы проектируемого устройства;
- 10) просмотр базового плана кристалла ПЛИС с размещенными элементами схемы;

1) сохранение файлов проекта;

12) формирование файлов для программирования ПЛИС.

Для запуска программы MAX+plus II необходимо:

- 1) ввести логическое имя пользователя и пароль, указанные администратором сети;
- 2) запустить файл maxstart.exe в каталоге, указанном преподавателем.

Для ввода имени проекта следует щелкнуть левой кнопкой манипулятора «мышь» на пункте меню File\Project\Name... или нажать комбинацию клавиш Ctrl+J (рис. 17).

В появившемся окне необходимо указать диск (Drives:) и каталог (Directories:) для размещения файлов проекта, а также ввести имя проекта (Project Name:) (например, sd_reg без расширения), после чего щелкнуть «мышью» на кнопке ОК (рис. 18) или нажать ENTER на клавиатуре.

Ввод электрической схемы проектируемого устройства необходимо начинать с первого уровня иерархии. Нижний (нулевой) уровень представлен логическими вентилями. В рассматриваемом примере требуется осуществить ввод схемы D-триггера (1-й уровень) на основе элементов 2И-НЕ, НЕ (0-й уровень) (см. рис. 16).

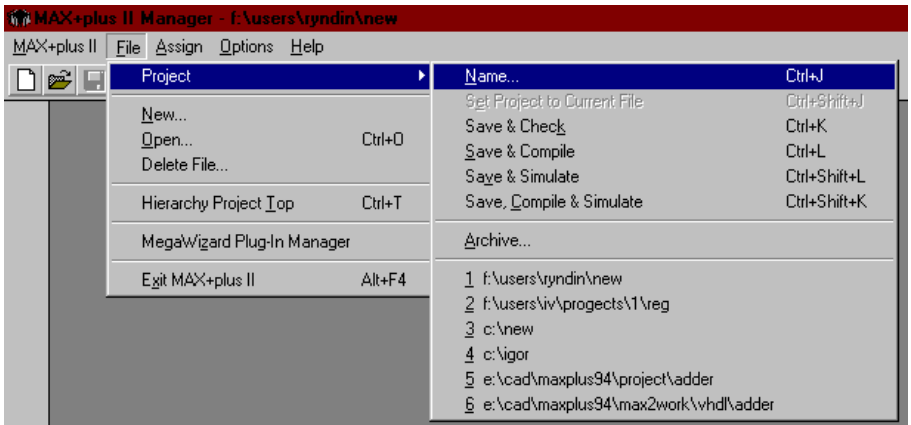


Рис. 17. Ввод имени проекта

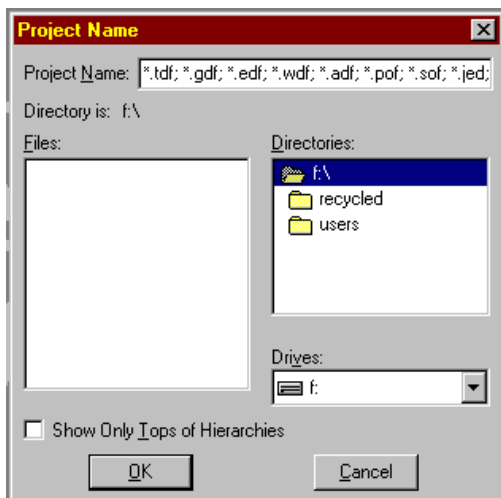


Рис. 18. Окно для ввода имени проекта

Для ввода схемы D-триггера необходимо:

- 1) войти в графический редактор, щелкнув левой кнопкой «мыши» на пункте меню MAX+plus II\Graphic Editor;
- 2) в появившемся пустом окне щелкнуть правой кнопкой «мыши» на любом свободном месте и в появившемся динамическом меню выбрать пункт Enter Symbol;
- 3) в появившемся диалоговом окне (рис. 19) выбрать двойным щелчком левой кнопки «мыши» библиотеку maxplus2\max2lib\prim в списке под заголовком Symbol Libraries и библиотечный элемент nand2 (2И-НЕ) в списке Symbol Files. При этом в окне графического редактора появится условное изображение элемента 2И-НЕ;
- 4) навести курсор «мыши» на изображение элемента, нажать левую кнопку манипулятора и, удерживая ее нажатой, переместить изображение элемента в требуемую точку в соответствии со схемой (см. рис. 16), после чего отпустить кнопку;
- 5) навести курсор «мыши» на изображение элемента, нажать Ctrl на клавиатуре и левую кнопку «мыши» и, удерживая их, скопировать изображение элемента в требуемую точку в соответствии со схемой (см. рис. 16), после чего отпустить кнопки;

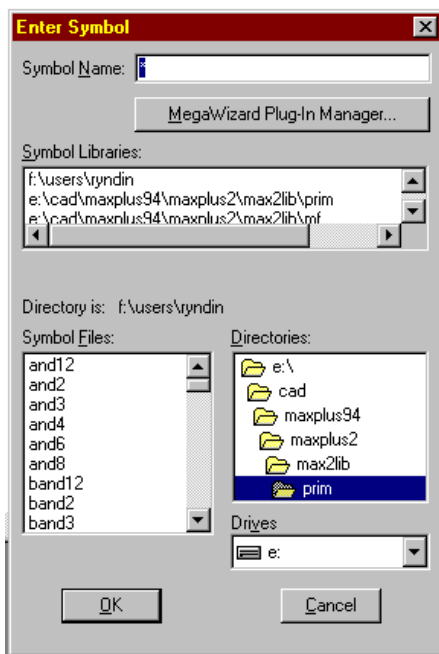


Рис. 19. Выбор библиотечного элемента

- 6) повторить пункт 5 для всех элементов 2И-НЕ D-триггера (см. рис. 16);
- 7) повторить пункты 2 – 5 для элементов not (НЕ), input (Вход), output (Выход), разместив их в соответствии со схемой (см. рис. 16);
- 8) дважды щелкнуть левой кнопкой «мыши» на заголовке PIN_NAME элемента input и ввести с клавиатуры имя входа (например, D);
- 9) повторить пункт 8 для всех входов (input) и выходов (output) схемы;
- 10) провести межэлементные соединения. Навести курсор «мыши» на изображение соответствующего контакта элемента, нажать левую кнопку и, удерживая ее, провести соединение. В месте изгиба соединительной линии следует отпускать левую кнопку «мыши», не сдвигая курсор, вновь нажать ее, после чего вести линию далее в ином направлении. Проводить ветвящиеся линии рекомендуется от соединяемого внешнего вывода к узлу. При этом изображения узлов проставляются автоматически. При необходимости переместить отдельные участки линий или отдельные уже соединенные в схему элементы, следует навести курсор «мыши» на переме-

щаемый элемент изображения, нажать левую кнопку и, удерживая ее, осуществить перемещение. При этом все линии связи перемещаются соответствующим образом автоматически;

- 11) сохранить файл схемы D-триггера, щелкнув левой кнопкой «мыши» на пункте меню File\Save и введя с клавиатуры имя файла (например, dtr.gdf). Файл сохранится в каталоге проекта по умолчанию.

Для ввода электрической схемы сдвигового регистра (второй уровень иерархии) на основе D-триггеров необходимо предварительно создать условное графическое изображение (Symbol) D-триггера.

Для создания условного графического изображения необходимо:

- 1) войти в символьный редактор, щелкнув левой кнопкой «мыши» на пункте меню MAX+plus II\Symbol Editor;
- 2) в появившемся окне символьного редактора (рис. 20) навести курсор «мыши» на левую границу графического изображения и дважды щелкнуть левой кнопкой;
- 3) в появившемся диалоговом окне (рис. 21) ввести с клавиатуры имя входа (например, D) в поле Full Pinstub Name, щелкнуть левой кнопкой «мыши» на заголовке Input Pin в поле I/O Type и на заголовке Used в поле Default Status, установить флажок Show Visible Pinstub Name in Graphic Editor, после чего щелкнуть левой кнопкой «мыши» на ОК или нажать клавишу ENTER на клавиатуре;
- 4) навести курсор «мыши» на появившееся изображение входа в виде символа × (рис. 22), нажать левую кнопку и, удерживая ее, переместить изображение входа в требуемую точку;
- 5) повторить пункты 2 - 4 для всех входов и выходов D-триггера, учитывая, что изображения выходов удобнее вводить, щелкая «мышью» в пункте 2 на правой границе условного графического изображения элемента. Имена входов и выходов должны совпадать с соответствующими именами на схеме, введенной в графическом редакторе (файл dtr.gdf);

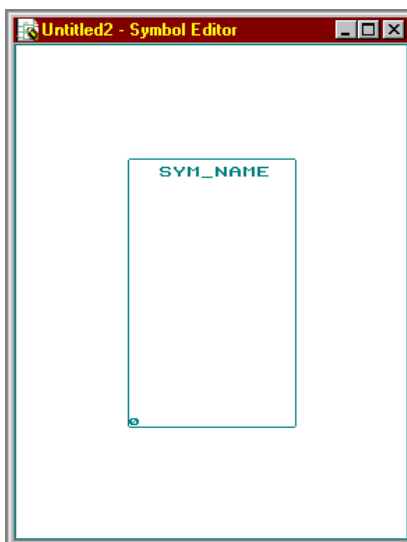


Рис. 20. Окно символьного редактора

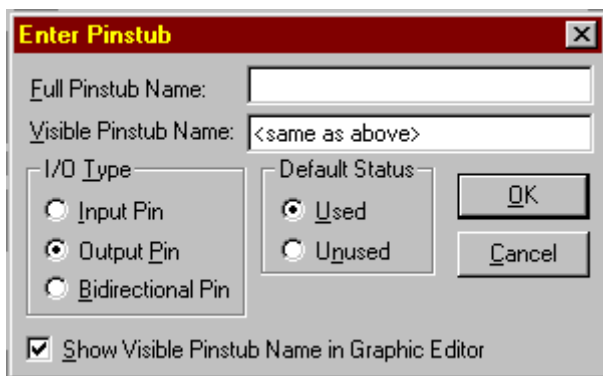


Рис. 21. Диалоговое окно символьного редактора

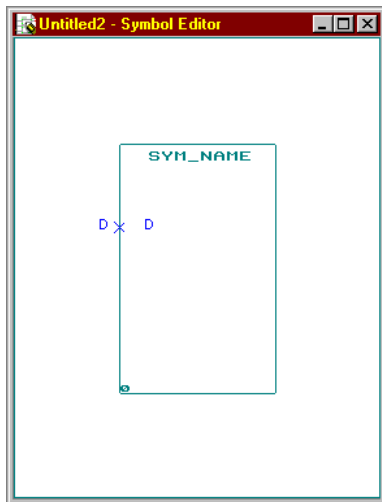


Рис. 22. Изображение входа в символьном редакторе

- 6) при необходимости удалить изображение входа или выхода необходимо навести курсор «мыши» на удаляемое изображение, щелкнуть правой кнопкой и в появившемся динамическом меню щелкнуть левой кнопкой на пункте Delete;
- 7) сохранить файл условного графического изображения D-триггера, щелкнув левой кнопкой «мыши» на пункте меню File\Save и введя с клавиатуры имя файла (например, dtg.sym). Имена соответствующих файлов с расширениями .sym и .gdf должны совпадать. Файл сохранится в каталоге проекта по умолчанию.

Ввод электрической схемы сдвигового регистра на основе D-триггеров (см. рис. 15) осуществляется в графическом редакторе аналогично вводу схемы D-триггера. Отличия состоят в следующем:

- 1) в пункте 3 необходимо выбрать библиотечный элемент dtg в каталоге проекта;
- 2) в пунктах 8 и 9 имена входов и выходов регистра должны отличаться от имен входов и выходов на схеме триггера (в файле dtg.gdf);
- 3) в пункте 11 сохранить схему регистра в файле с именем проекта (в приведенном примере – sd_reg.gdf).

Для ввода входных сигналов необходимо:

- 1) войти в редактор сигналов, щелкнув левой кнопкой «мыши» на пункте меню MAX+plus II\Waveform Editor;

- 2) в появившемся окне редактора (рис. 23) дважды щелкнуть левой кнопкой «мыши» на свободной строке (ниже заголовка Name:);

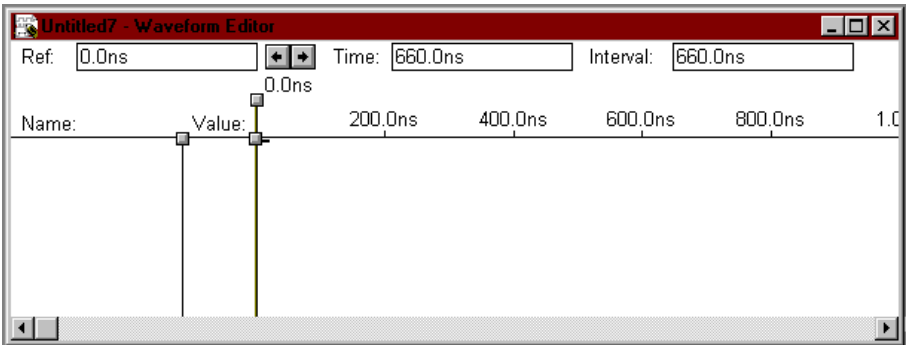


Рис. 23. Окно редактора сигналов

- 3) в появившемся диалоговом окне щелкнуть левой кнопкой «мыши» на кнопке List;
- 4) щелкнуть левой кнопкой «мыши» на имени входа из списка, появившегося под заголовком Nodes and Gropes from SNF, после чего щелкнуть левой кнопкой «мыши» на ОК или нажать клавишу ENTER на клавиатуре;
- 5) повторить пункты 2 – 4 для всех входов и выходов схемы, которые предполагается использовать при моделировании;
- 6) определить все входные сигналы, щелкая левой кнопкой «мыши» на строке с именем входного сигнала, а затем на одной из следующих кнопок:



- уменьшить масштаб изображения;



- увеличить масштаб изображения;



- установить низкий уровень логического нуля;



- установить высокий уровень логической единицы;



- инвертировать логический уровень;



- установить неопределенный логический уровень;



- установить высокоимпедансное Z-состояние;



- установить периодическое изменение логического уровня.

При установке периодического изменения логического уровня в появившемся диалоговом окне необходимо установить начальное значение уровня в поле под заголовком Starting Value (0 или 1) и период следования импульсов под заголовком Multiplied By (длительность импульсов умножается на введенное целое число ≥ 1);

- 7) сохранить файл сигналов, щелкнув левой кнопкой «мыши» на пункте меню File\Save и введя с клавиатуры имя проекта (в данном примере sd_reg.scf). Файл сохранится в каталоге проекта по умолчанию.

Для выбора типа ПЛИС необходимо:

- 1) щелкнуть левой кнопкой «мыши» на пункте меню Assign\Device;
- 2) в появившемся диалоговом окне (рис. 24) щелкнуть левой кнопкой «мыши» на выбранной серии ПЛИС в списке под заголовком Device Family (для просмотра списка следует щелкнуть левой кнопкой «мыши» на стрелку справа от поля Device Family), после чего щелкнуть левой кнопкой «мыши» на выбранном типе ПЛИС из данной серии в списке под заголовком Devices;
- 3) щелкнуть левой кнопкой «мыши» на ОК или нажать клавишу ENTER на клавиатуре.

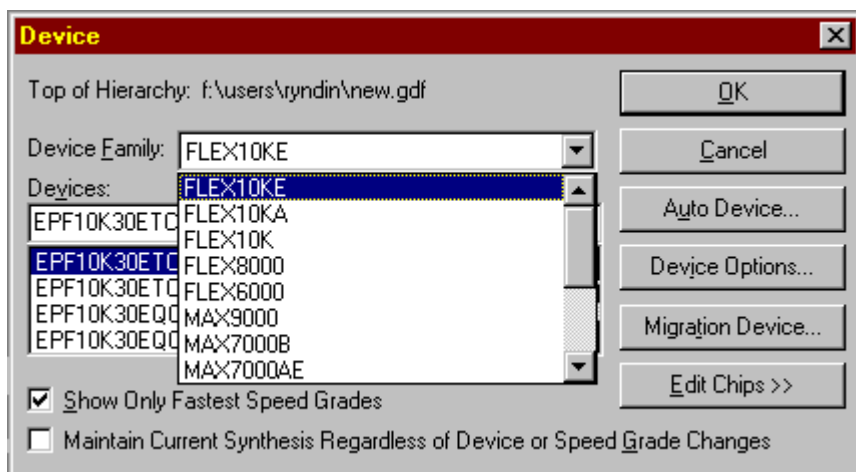


Рис. 24. Диалоговое окно выбора типа ПЛИС

Для проверки корректности ввода электрической схемы проекта, компиляции и функционально-логического моделирования необходимо щелкнуть левой кнопкой «мыши» на пункте меню File\Project\Save, Compile & Simulate или нажать комбинацию клавиш Ctrl+Shift+K.

При этом последовательно появляются следующие окна:

- 1) окно компилятора с перечисленными в верхней части этапами компиляции (рис. 25);
- 2) окно сообщений компилятора об ошибках (Error) и замечаниях (Warning) (рис. 26);
- 3) окно подсистемы функционально-логического моделирования (рис. 27);
- 4) окно сообщений подсистемы функционально-логического моделирования (рис. 28).

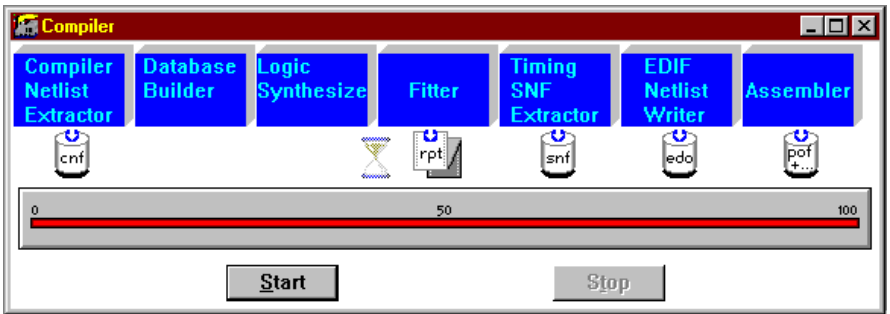


Рис. 25. Окно компилятора

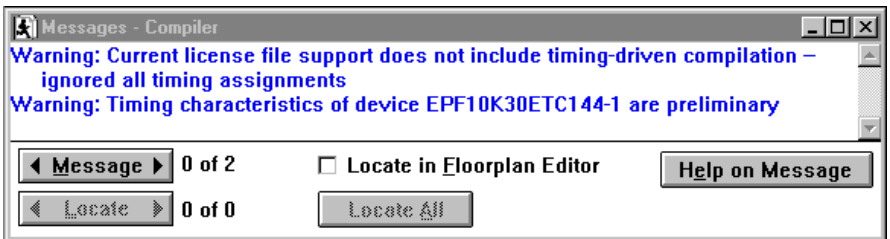


Рис. 26. Окно сообщений компилятора

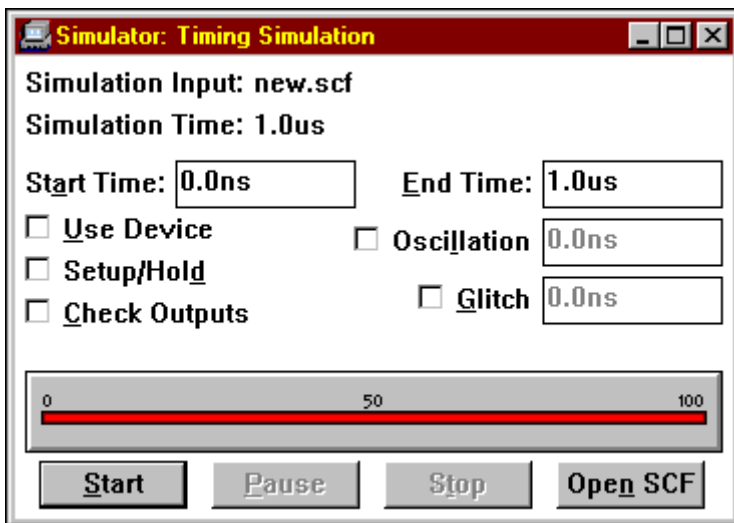


Рис. 27. Окно подсистемы функционально-логического моделирования

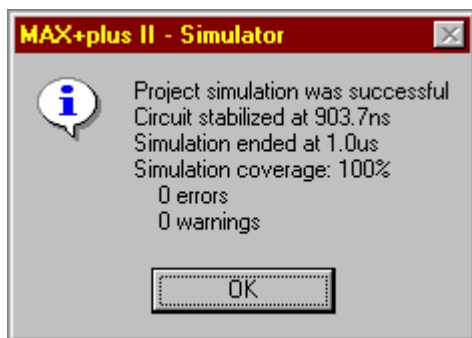


Рис. 28. Окно сообщений подсистемы функционально-логического моделирования

Для просмотра результатов моделирования необходимо щелкнуть левой кнопкой «мыши» на ОК в окне сообщений (см. рис. 28) или нажать клавишу ENTER на клавиатуре, после чего щелкнуть левой кнопкой «мыши» на кнопке Open SCF в окне подсистемы моделирования (см. рис. 27). При этом появится окно редактора сигналов с результатами моделирования (рис. 29).

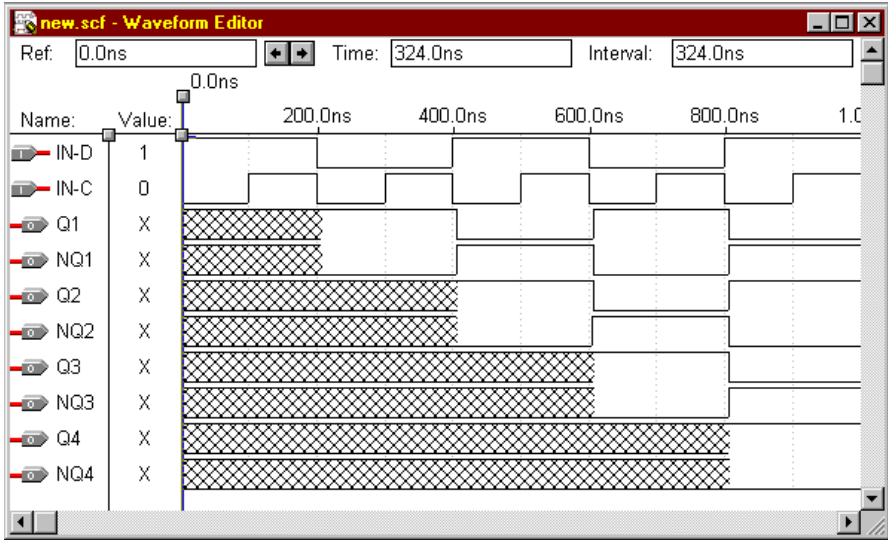


Рис. 29. Результаты функционально-логического моделирования

Для расчета задержек прохождения сигналов со входов на выходы проектируемого устройства следует щелкнуть левой кнопкой «мыши» на пункте меню MAX+plus II\Timing Analyzer. При этом появится таблица задержек сигналов (рис. 30), а результаты автоматически сохранятся в текстовом файле с именем проекта и расширением .tao (в рассматриваемом примере sd_reg.tao). Файл с таблицей задержек можно просмотреть любым текстовым редактором, в том числе и встроенным в MAX+plus II. Он вызывается щелчком левой кнопки «мыши» на пункте меню MAX+plus II\Text Editor.

Для просмотра базового плана кристалла ПЛИС с размещенными элементами схемы проекта следует щелкнуть левой кнопкой «мыши» на пункте меню MAX+plus II\Floorplan Editor. При этом появится окно базового плана ПЛИС (рис. 31).

Для просмотра цоколевки внешних выводов ПЛИС необходимо щелкнуть левой кнопкой «мыши» на пункте меню Layout\Device View (рис. 32).

Подготовка файлов для программирования ПЛИС в соответствии с разработанным проектом осуществляется автоматически после активизации левой кнопкой «мыши» пункта меню MAX+plus II\Programmer.

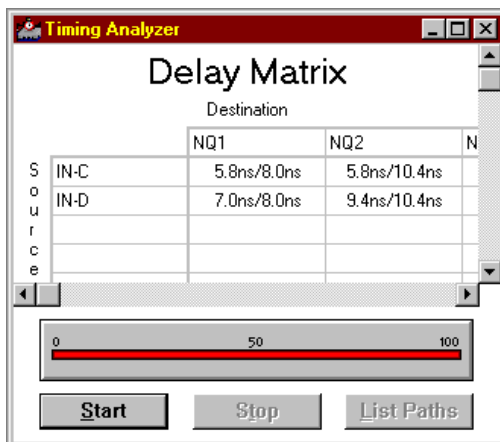


Рис. 30. Таблица задержек сигналов

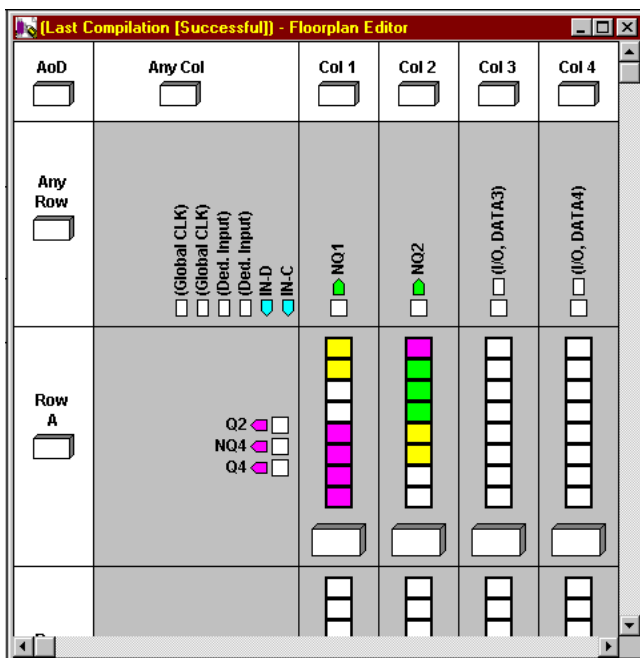


Рис. 31. Базовый план ПЛИС

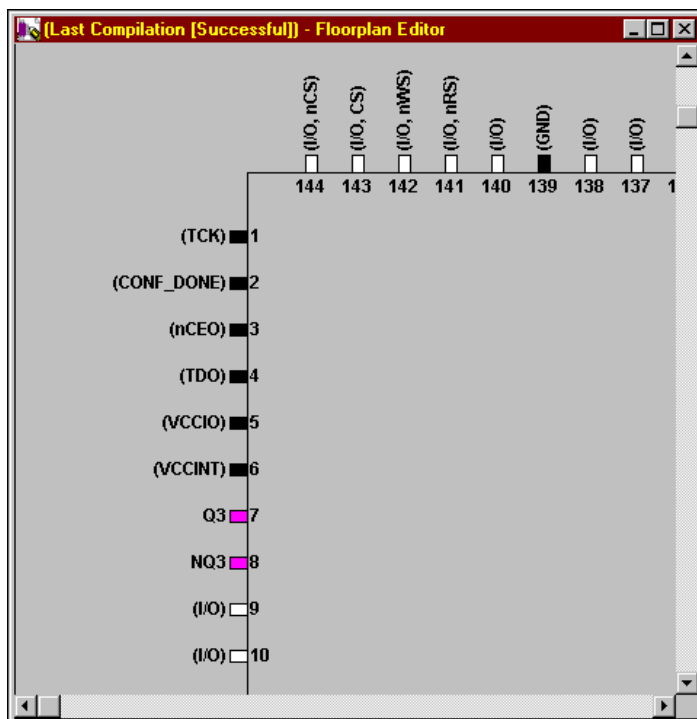


Рис. 32. Цоколевка ПЛИС

5. ПОРЯДОК ВЫПОЛНЕНИЯ ЛАБОРАТОРНОЙ РАБОТЫ

1. Ознакомиться со сведениями о системе MAX+plus II, изложенными в настоящем руководстве.
2. Получить у преподавателя вариант задания по лабораторной работе (варианты заданий приведены в приложении).
3. Выполнить последовательно все этапы проектирования, изложенные в разделе 4.
4. Показать результаты проектирования преподавателю.
5. Сохранить файлы проекта на дискете.

Примечание. Файлы проекта являются отчетом о выполнении лабораторной работы.

6. КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Способы реализации специализированных СБИС. Достоинства и недостатки.
2. Структурная организация ПЛИС.
3. Основные характеристики современных ПЛИС.
4. Маршрут 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.

ЛИТЕРАТУРА

1. Коноплев Б.Г. Проектирование СБИС и их применение в ЭВА: Конспект лекций. Ч. 2. - Таганрог: ТРТИ, 1991. - 45 с.
2. <http://www.altera.com>. Altera Devices. - 1999.
3. Стешенко В. Занятие 2: система проектирования MAX+plus II ALTERA / Chip News. №9(42). 1999. С. 15-18.

ВАРИАНТЫ ЗАДАНИЙ К ЛАБОРАТОРНОЙ РАБОТЕ

Вариант №1

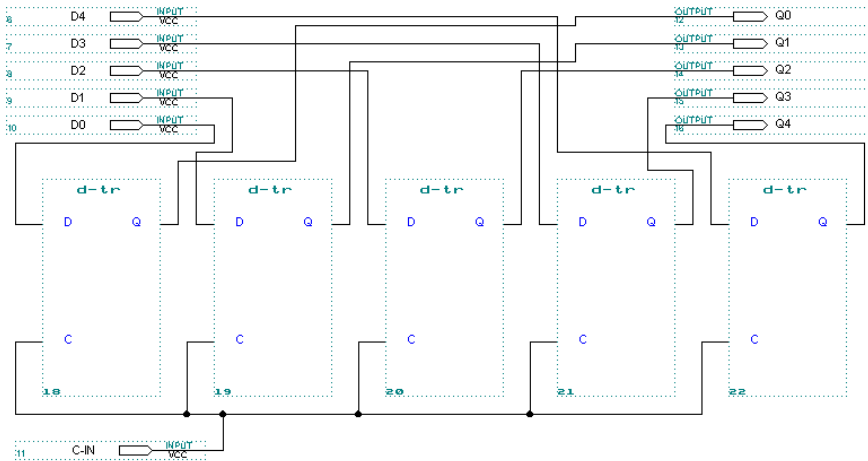


Рис. I. Регистр

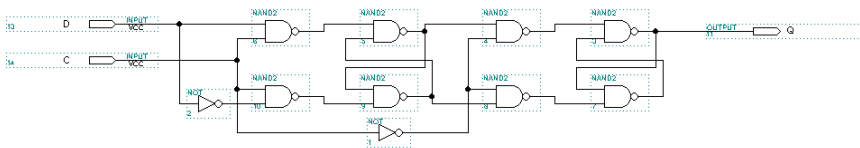


Рис. II. D-триггер

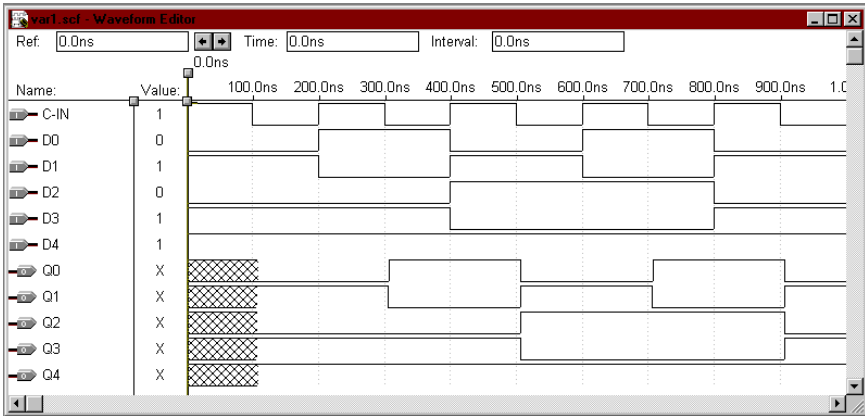


Рис. III. Результаты моделирования регистра

Вариант №2

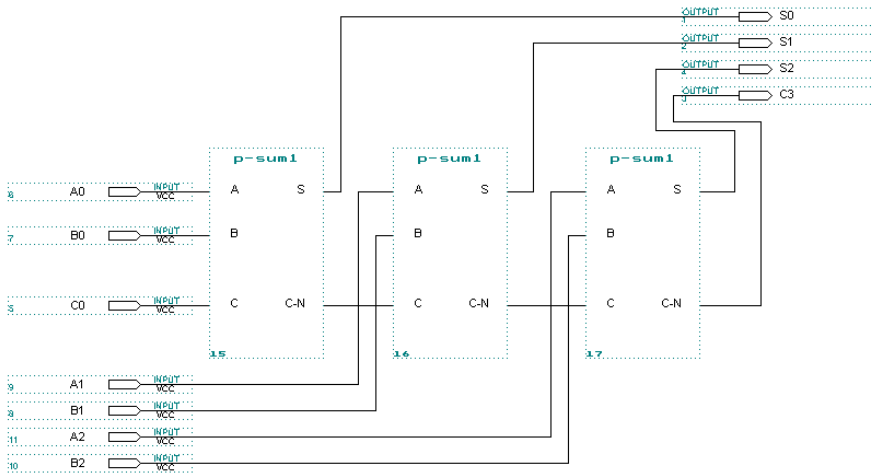


Рис. IV. Трехразрядный сумматор на элементах исключающее ИЛИ, И-НЕ

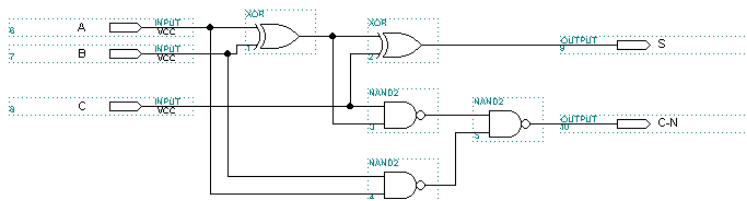


Рис. V. Сумматор на элементах исключающее ИЛИ, И-НЕ

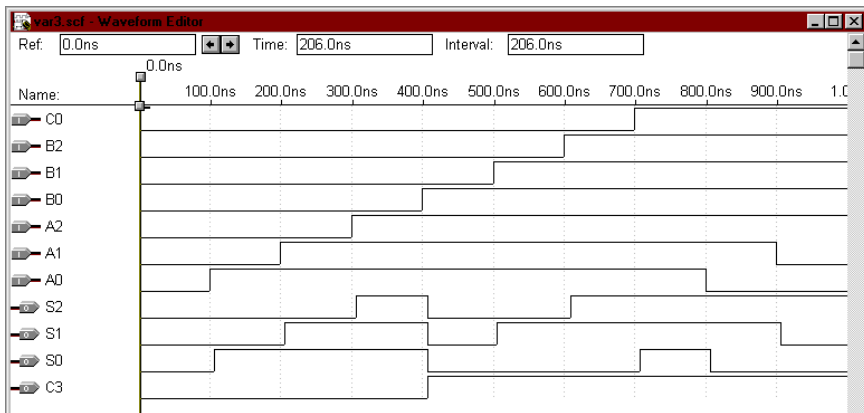


Рис. VI. Результаты моделирования трехразрядного сумматора на элементах исключающее ИЛИ, И-НЕ

Вариант №3

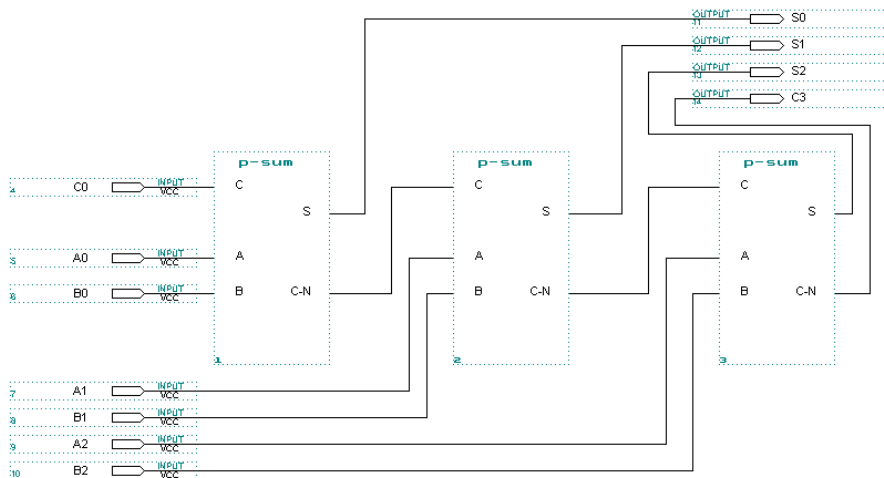


Рис. VII. Трехразрядный сумматор на элементах И-НЕ

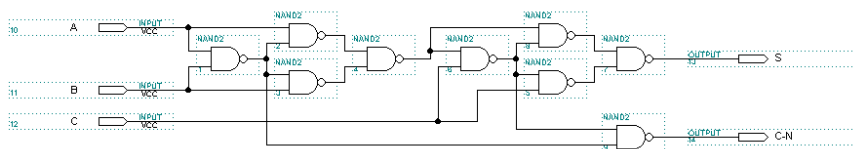


Рис. VIII. Сумматор на элементах И-НЕ

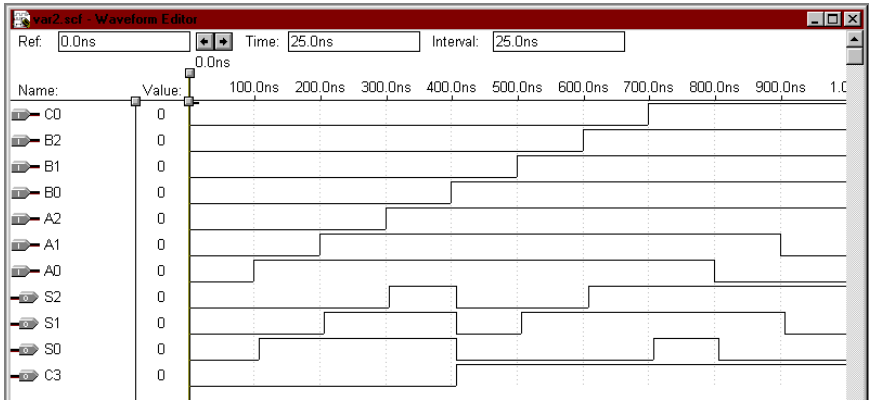


Рис. IX. Результаты моделирования трехразрядного сумматора на элементах И-НЕ

Вариант №4

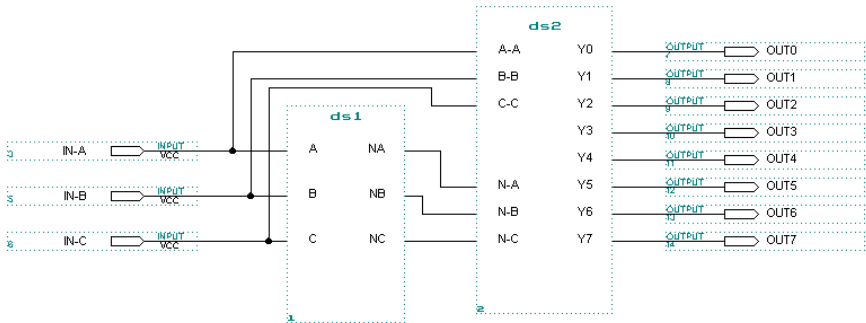


Рис. X. Дешифратор на элементах И-НЕ

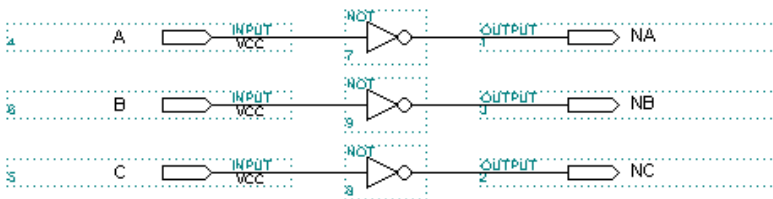


Рис. XI. Блок ds1

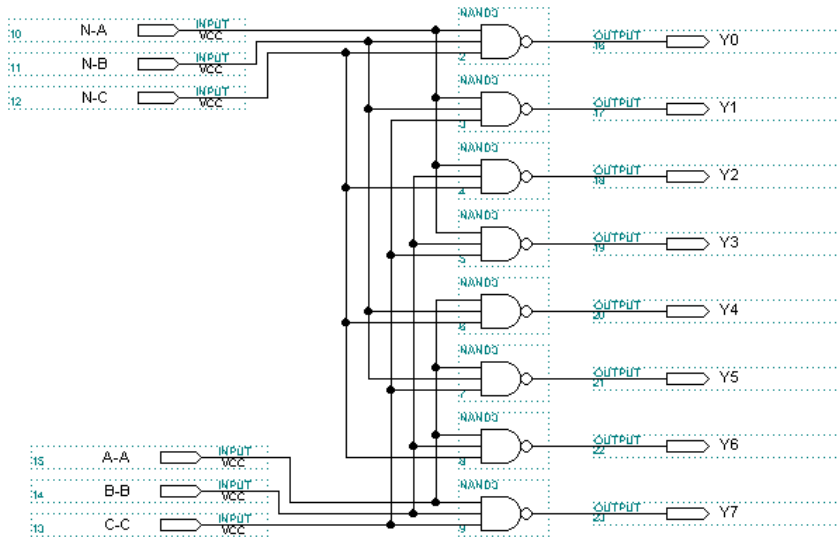


Рис. XII. Блок ds2

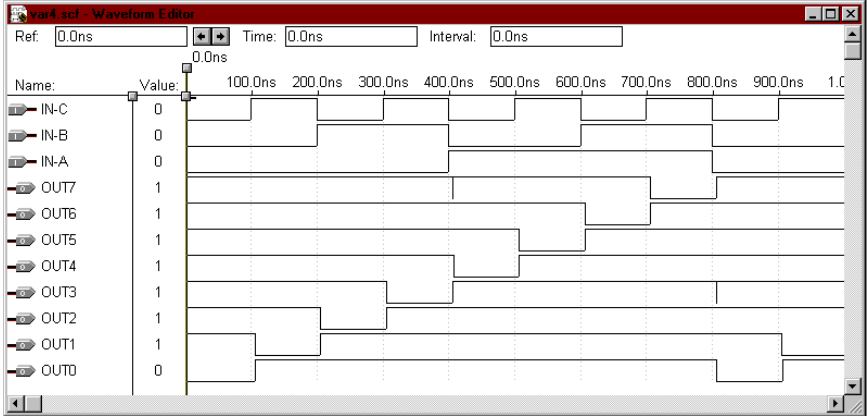


Рис. XIII. Результаты моделирования дешифратора на элементах И-НЕ

Вариант №5

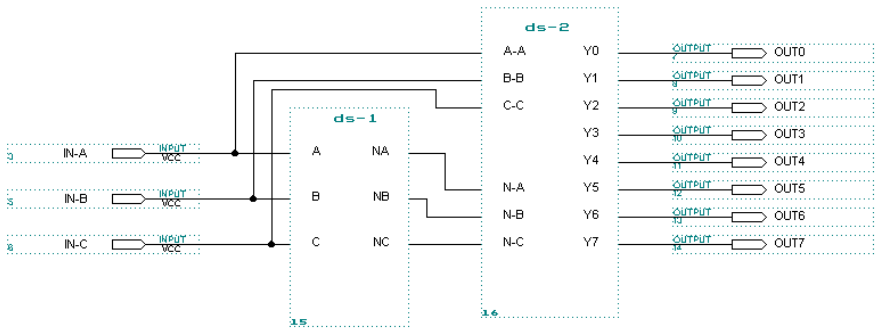


Рис. XIV. Дешифратор на элементах ИЛИ-НЕ

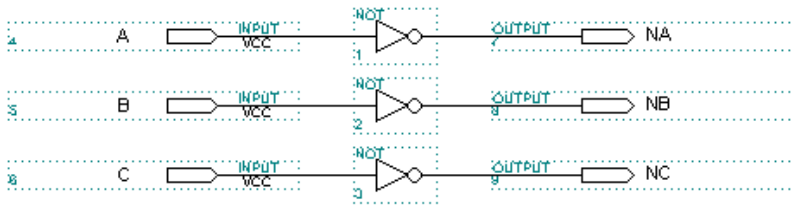


Рис. XV. Блок ds-1

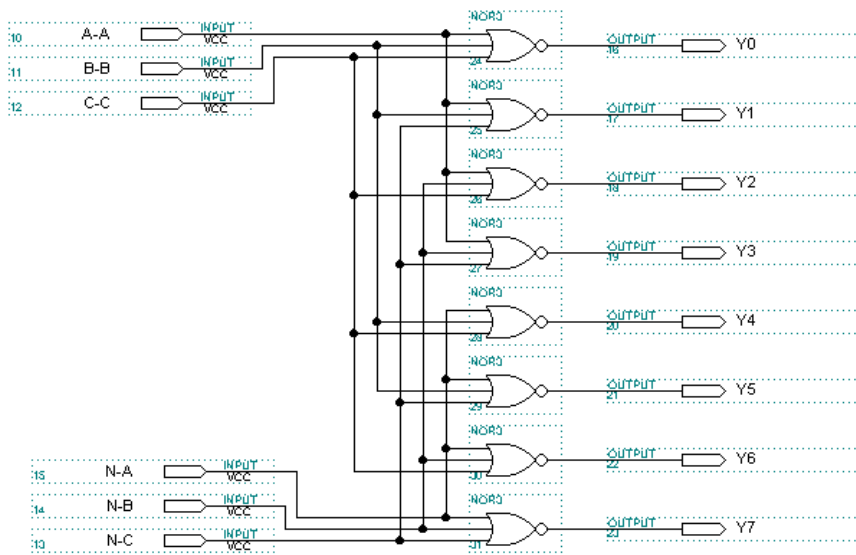


Рис. XVI. Блок ds-2

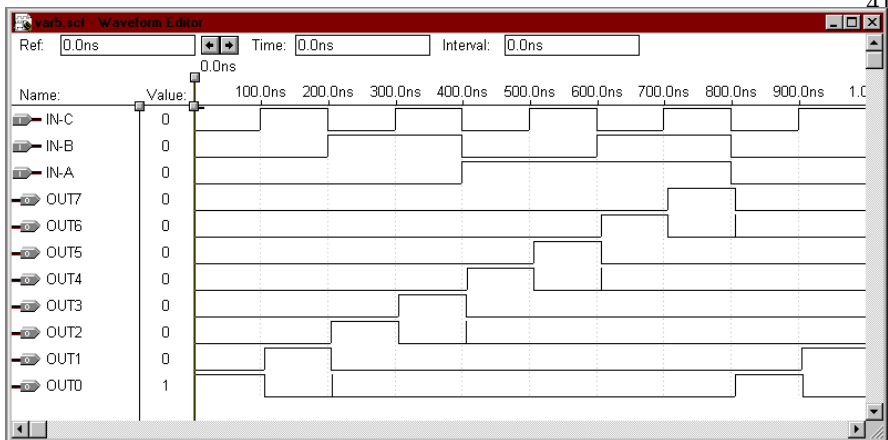


Рис. XVII. Результаты моделирования дешифратора на элементах ИЛИ-НЕ

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
1. ОБЩИЕ СВЕДЕНИЯ О СИСТЕМЕ MAX+plus II	6
2. РЕДАКТОРЫ СИСТЕМЫ MAX+plus II	8
3. КОМПИЛЯЦИЯ ПРОЕКТА В СИСТЕМЕ MAX+plus II	16
4. ПРИМЕР ПРОЕКТИРОВАНИЯ СПЕЦИАЛИЗИРОВАННОЙ ИС	18
5. ПОРЯДОК ВЫПОЛНЕНИЯ ЛАБОРАТОРНОЙ РАБОТЫ	31
6. КОНТРОЛЬНЫЕ ВОПРОСЫ	32
ЛИТЕРАТУРА	32
ПРИЛОЖЕНИЕ. ВАРИАНТЫ ЗАДАНИЙ К ЛАБОРАТОРНОЙ РАБОТЕ ..	33