

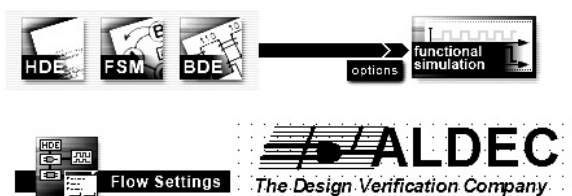
ФГБОУ ВПО
«Воронежский государственный технический университет»

Кафедра конструирования и производства радиоаппаратуры

217-2014

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

к выполнению лабораторных работ № 1-4
по дисциплине «Специализированные БИС и устройства
функциональной электроники в приборостроении»
для студентов направления 12.03.01 «Приборостроение»
очной и заочной форм обучения



Воронеж 2014

Составитель канд. техн. наук А.А. Пирогов
УДК 621.396

Методические указания к выполнению лабораторных работ № 1-4 по дисциплине «Специализированные БИС и устройства функциональной электроники в приборостроении» для студентов направления 12.03.01 «Приборостроение» очной и заочной форм обучения / ФГБОУ ВПО «Воронежский государственный технический университет»; сост. А.А. Пирогов. Воронеж, 2014. 35 с.

В методических указаниях изложены требования и рекомендации по подготовке и выполнению лабораторных работ, к каждой из них приводятся краткие теоретические сведения, домашние и лабораторные задания, с рекомендациями по их выполнению, контрольные вопросы и список рекомендуемой литературы.

Методические указания подготовлены в электронном виде в текстовом редакторе MS Word 2007 и содержатся в файле СБИС и УФЭ_ЛР.doc.

Ил. 25. Табл. 1. Библиогр.: 3 назв.

Рецензент канд. техн. наук, доц. М.А. Ромашенко

Ответственный за выпуск зав. кафедрой д-р техн. наук,
проф. А.В. Муратов

Издается по решению редакционно-издательского совета
Воронежского государственного технического университета

© ФГБОУ ВПО «Воронежский
государственный технический
университет», 2014

1. ЛАБОРАТОРНАЯ РАБОТА № 1

ПРОЕКТИРОВАНИЕ ФУНКЦИОНАЛЬНЫХ УЗЛОВ КОМБИНАЦИОННОГО ТИПА НА ЯЗЫКЕ VHDL

1.1. Общие указания по выполнению лабораторной работы

Целью лабораторной работы является углубление и закрепление знаний студентов в области основных понятий цифровой схемотехники, моделей и параметров логических элементов, а также получение навыков в проектировании программируемых логических интегральных схем (ПЛИС) с использованием систем автоматизированного проектирования (САПР) на языке VHDL.

Лабораторный практикум посвящен проектированию структур функциональных узлов комбинационного и последовательностного типа. В рамках данной лабораторной работы необходимо изучить структуру и основные принципы построения проекта на языке описания аппаратных средств VHDL с использованием программного комплекса Active-HDL, построить модели простейших мультиплексоров и дешифраторов.

На выполнение лабораторной работы отводится восемь часов.

Перед выполнением лабораторной работы студент должен самостоятельно выполнить домашнее задание в соответствии с данными методическими указаниями.

Студент, явившийся на занятия, должен иметь методические указания по данной лабораторной работе. В начале занятия преподаватель проверяет выполнение студентом домашнего задания и наличие заготовки отчета по данной лабораторной работе в его рабочей тетради.

К выполненной работе прилагаются необходимые схемы, эскизы, тексты и результаты проектирования, протоколы работы

с программным комплексом и другие материалы согласно указаниям по оформлению отчета.

1.2. Домашние задания и методические указания по их выполнению

Задание 1. Ознакомиться с моделями простейших логических элементов, системами их параметров. Изучить основные узлы комбинационного типа и их назначение. Для этого необходимо воспользоваться лекциями по курсу и литературой [1, С. 7-22].

Задание 2. Ознакомиться с основными принципами и методами построения моделей функциональных узлов на логическом уровне. При выполнении домашнего задания студент должен ознакомиться с основными сведениями о языке VHDL и способами описания проекта. Для этого необходимо воспользоваться лекциями по курсу и литературой [1, С. 81-91, С. 637-642, С. 673-680].

В рамках данной дисциплины студент должен получить основные навыки в работе со следующими инструментами – языком описания аппаратуры (Verilog, VHDL) и средой компьютерного проектирования (Active-HDL, Xilinx ISE).

На языке описания аппаратуры (HDL) производится описание поведения или структура разрабатываемого устройства, далее осуществляется моделирование, где выявляются и устраняются ошибки проекта. После чего производится синтез и реализация проекта на ПЛИС.

Автоматизированное проектирование – это основное направление развития технологий создания электронной аппаратуры. До недавнего времени основным методом решения подобных задач считались разработки на базе серийно выпускаемых стандартных интегральных микросхем, и большинство САПР ориентировались на реализацию именно этой технологии проектирования.

Наиболее востребованным вариантом являются САПР сквозного проектирования, которые позволяют выполнить всю цепочку разработки: от входного описания до создания эскиза печатной платы. Примерами сквозных САПР являются PCAD, DesignLab и OrCAD.

В последние годы появляются новые технологии проектирования электронной аппаратуры, основанные на современных персональных компьютерах, интегрированных САПР и программируемых микросхемах (ПЛИС). Одним из представителей данных САПР является среда проектирования Active-HDL 7.1. Предлагаемая версия данного пакета занимает лидирующие позиции в мире. Далее рассмотрим основные принципы построения проекта и его моделирование на примере простой модели мультиплексора.

Ввод проекта. В начале проектирования разработчику представлены три окна интегрированной среды Active-HDL 7.1:

- **Design Browser** – окно просмотра проекта, в котором отображается все содержимое рабочего пространства.

- **Console** – окно для интерактивного ввода и/или вывода текстовой информации. Все инструментальные средства пакета выдают в это окно сообщения о своей работе, в том числе предупреждения и ошибки.

- В третьем окне активна закладка **Design Flow Manager** («Менеджер маршрута проектирования»). Обычно данное окно занимает большую часть экрана потому, что именно здесь пользователь создает основные документы проекта. На закладке Design Flow Manager изображен доступный для заданных установок маршрут проектирования. Он ограничивает действия тремя способами описания проекта (HDE, FSM, BDE) и последующим функциональным моделированием (functional simulation).

Если необходимо создать проект в текстовом формате на языке VHDL, Verilog или SystemC, то следует нажать кнопку HDE. В качестве рабочего инструмента будет вызван текстовый редактор HDE (от слов Hardware Description language Editor).

Если необходимо представить описание проекта в виде диаграммы состояний конечного автомата (Finite State Machine), то следует использовать редактор FSM.

В рамках данного лабораторного практикума реализация функциональных узлов будет осуществлена с использованием вручную построенных схем, потому необходимо использовать схемный редактор BDE (Block Diagram Editor). Вызвав BDE, запускается «мастер» создания нового исходного файла **New Source File Wizard**. Необходимо следовать его командам, далее приступаем к созданию проекта.

Создание проекта. На первом этапе необходимо найти и разместить на чертеже схемы графические изображения нужных нам элементов. Чтобы открыть библиотеку элементов, необходимо воспользоваться инструментом **Show Symbols Toolbox** (см. рис. 1).

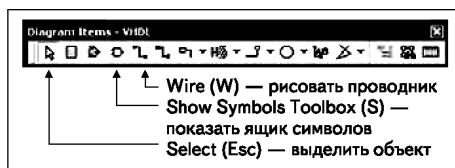


Рис. 1. Панель инструментов среды проектирования Active-HDL

В режиме, заданном по умолчанию, для построения схемы доступна одна библиотека со встроенными символами **Built-in symbols**. Раскрыв ее содержимое, вы увидите набор простейших логических элементов (см. рис. 2).

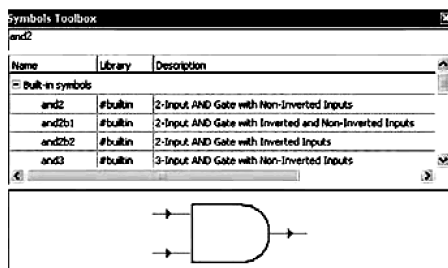


Рис. 2. Диалоговая панель со встроенными символами

Для построения мультиплексора необходимы элементы (and2, inv, or2) размещаются они в зоне проекта известным методом Drag & Drop. Чтобы сделать видимыми позиционные обозначения символов, исполним команду **View Texts** из меню **Diagram**. На открывшейся панели View Texts устанавливается флажок NAME. Позиционные обозначения всех символов на схеме станут видимыми (см. рис. 3).

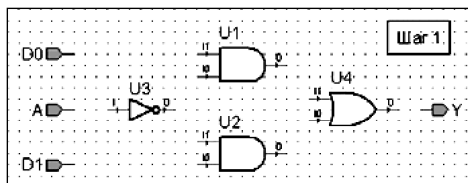


Рис. 3. Символы с указанными позиционными обозначениями

На втором шаге элементы схемы необходимо соединить проводниками в соответствии со структурой мультиплексора. Нажмем на панели инструментов **Diagram Items** кнопку **Wire** или на клавиатуре клавишу **W**.

Поместите курсор в точку, из которой необходимо начать проводник, и щелкните левой кнопкой мыши (ЛКМ). Переместите курсор в точку назначения и опять щелкните ЛКМ. Если проводник имеет сложную форму, придется фиксировать

каждый его сегмент (излом, угол) щелчком мыши. Законченная схема должна выглядеть так, как показано на рис. 4.

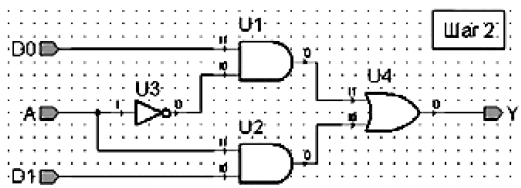


Рис. 4. Окончательная схема мультиплексора

На третьем этапе проводникам схемы надо назначить имена. Цепи, подключенные к терминалам (входным и выходным портам), автоматически получают те же самые названия, что и соответствующий порт. А внутренние проводники система именует по шаблону (NET9, NET13, NET17 и др.), у них есть системные имена. Однако если пользователь собирается контролировать сигналы на внутренних цепях, желательно присвоить им названия.

Чтобы сделать имя видимым на экране монитора, достаточно дважды щелкнуть на проводнике мышью, затем на открывшейся диалоговой панели Wire Properties выбрать закладку View Text и установить флажок NAME. Так же просто заменить системное имя пользовательским. Откройте диалоговую панель Wire Properties и введите в поле Segment желаемое имя.

На четвертом шаге необходимо выполнить электрический контроль схемы (команда **Check Diagram** в выпадающем меню **Diagram**) и компиляцию проекта (команда **Compile** в выпадающем меню **Design**). Система не только проинформирует вас о числе ошибок и предупреждений, но и покажет те места на схеме, где они обнаружены.

На пятом шаге графическое описание проекта должно быть конвертировано в текстовое описание (в формат языка

VHDL). Выполняется данная процедура командой **Generate HDL Code**.

Задание диаграмм входных сигналов. Временные диаграммы внешних воздействий задаются в специальном окне редактора сигналов **Waveform Editor**, который вызывается по пиктограмме New Waveform на инструментальной панели Standard. При запуске редактор Waveform Editor открывает новую закладку поверх ранее построенной схемы, которая состоит из двух частей. Левая часть предназначена для ввода сигналов, правая – для показа временных диаграмм.

Необходимо щелчком правой кнопкой мыши (ПКМ) в левой части окна для выбора в контекстном меню команды Add Signals. Она дублируется также на инструментальной. Откроется одноименная панель со списком всех доступных сигналов. Нажатием кнопки **Add** осуществляется перенос выбранных сигналов в окно редактора Waveform Editor. Выделив один или несколько программируемых сигналов и щелчком на иконке Stimulators исполним одноименную команду **Stimulators**.

Откроется диалоговая панель Stimulators (см. рис. 5). В окне Signals выделим сигнал, в окне Type определим для него тип стимулятора **Clock** и в правом окне зададим частоту (Frequency) 10 МГц. Нажмем кнопку Apply, не закрывая панель, необходимо осуществить программирование всех входных сигналов. Кроме стимулятора Clock возможно задавать входные воздействия по средствам стимуляторов **Formula** (задание входных сигналов по указанной закономерности), **Value** (задание фиксированного значения сигнала на весь временной диапазон моделирования), **Hotkey** (переключение состояний входных сигналов с использованием «горячих клавиш»).

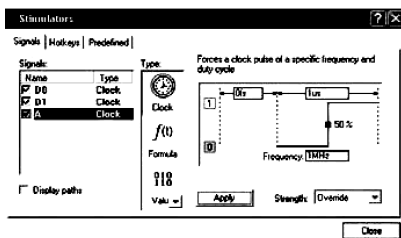


Рис. 5. Диалоговое окно стимулятора Clock

Моделирование проекта. Есть несколько способов запустить схему на моделирование (см. рис. 6). Одним из вариантов является пошаговое моделирование командой **Run For** из меню **Simulation**. Имитатор, получив такую команду, продвинет моделирование на один шаг, величина которого задается в соседнем окне справа от кнопки.

По умолчанию шаг равен 100 ns, но его всегда можно поменять. Повторное исполнение этой команды продвинет время моделирования еще на один шаг и т. д. В паузе перед очередным шагом, возможно, изменить не только его длительность, но и перепрограммировать параметры входных сигналов. Пошаговое моделирование является достаточно эффективным при отладке схемы.

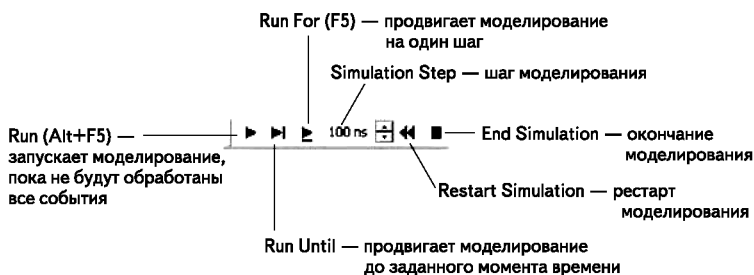


Рис. 6. Инструменты управления процессом моделирования

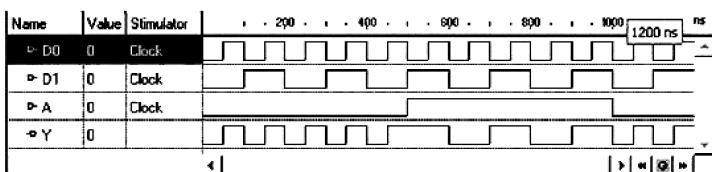


Рис. 7. Пример результатов моделирования

1.3. Вопросы к домашнему заданию

1.3.1. Дать определение идеализованного логического элемента.

1.3.2. Перечислить основные логические элементы, представить таблицы истинности.

1.3.3. Перечислить основные функциональные узлы комбинационного типа цифровой схемотехники, дать краткое описание.

1.3.4. Дать определение двоичного дешифратора, мультиплексора и демультиплексора, представить таблицы состояний, пояснить принцип работы.

1.3.5. Дать определение компаратора кода, сумматора и умножителя, представить таблицы состояний, пояснить принцип работы.

1.4. Лабораторные задания и методические указания по их выполнению

Задание 1. Построить и отладить модели мультиплексоров на логическом уровне, опираясь на пример приведенный в домашнем задании, схемы мультиплексоров представлены на рис. 8 и 9. Для каждой схемы необходимо построить временные диаграммы, демонстрирующие их работу.

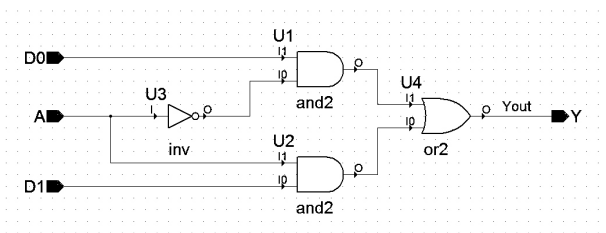


Рис. 8. Схема мультиплексора «2-1»

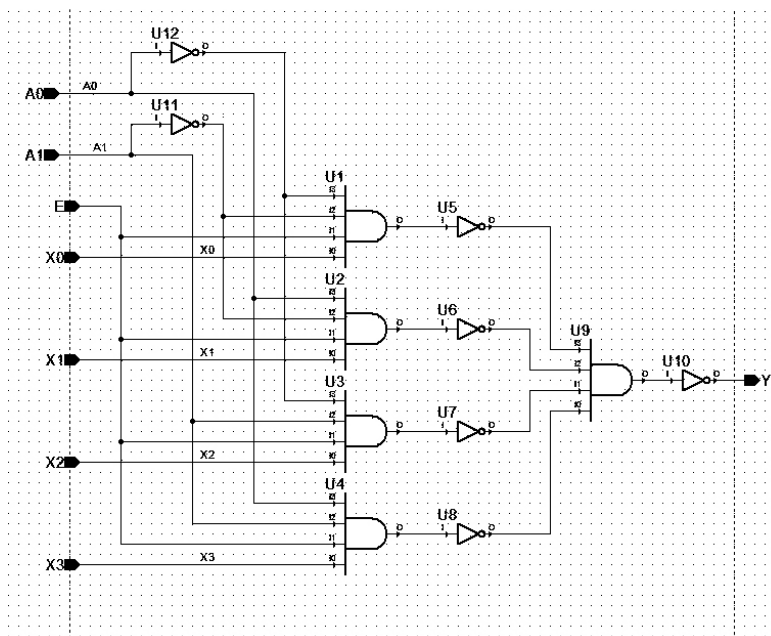


Рис. 9. Схема мультиплексора «4-1»

Задание 2. Построить и отладить модели дешифраторов на логическом уровне, опираясь на пример приведенный в домашнем задании, схемы дешифраторов представлены на рис. 10, 11 и 12. Для каждой схемы необходимо построить временные диаграммы, демонстрирующие их работу и соответствующие их таблицам состояний. Обратите внимание,

что демультиплексор на рис. 10 в отличие от остальных имеет инверсные выходы, но при этом не имеет разрешающего сигнала «Е»

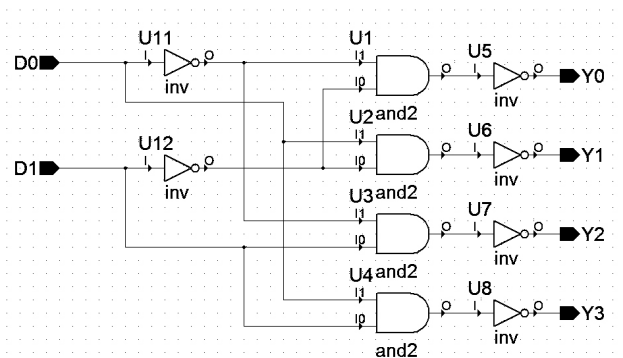


Рис. 10. Схема дешифратора «2-4» с инверсными выходами

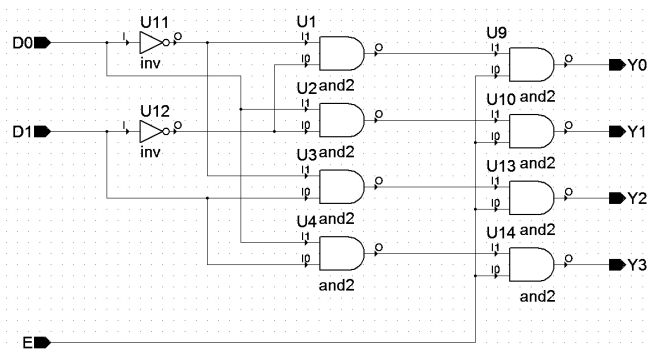


Рис. 11. Схема дешифратора «2-4» с использованием разрешающего сигнала «Е»

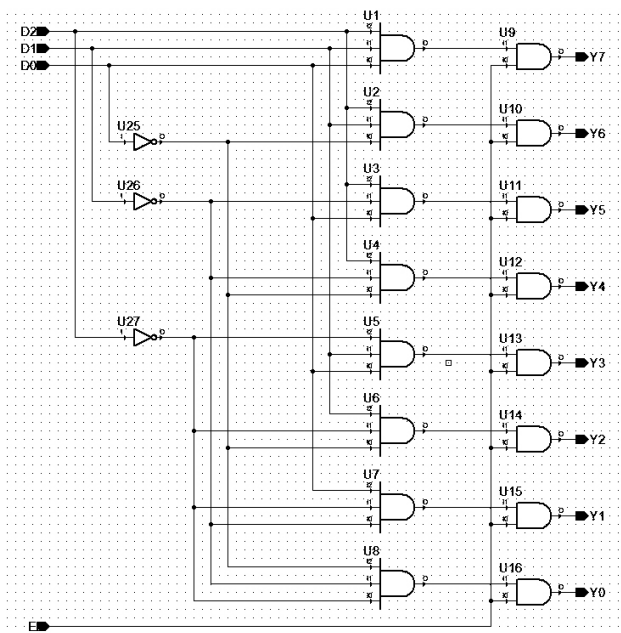


Рис. 12. Схема дешифратора «3-8» с использованием разрешающего сигнала «Е»

1.5. Контрольные вопросы

1.5.1. Какова цель лабораторной работы?

1.5.2. В чем заключается лабораторное задание?

Пояснить ход его выполнения.

1.5.3. Какие данные являлись исходными для выполнения работы?

1.5.4. Пояснить работу полученных моделей функциональных узлов комбинационного типа.

1.5.5. Сформулируйте выводы по данной лабораторной работе.

2. ЛАБОРАТОРНАЯ РАБОТА № 2

ПРОЕКТИРОВАНИЕ ИЕРАРХИЧЕСКИХ МОДУЛЕЙ НА ЯЗЫКЕ VHDL

2.1. Общие указания по выполнению лабораторной работы

Целью лабораторной работы является углубление и закрепление знаний студентов в области функциональных узлов комбинационного типа цифровой схемотехники. В рамках данной лабораторной работы необходимо построить на логическом уровне иерархическую модель дешифратора адреса м мультимплексора с использованием программного комплекса Active-HDL на языке описания аппаратуры VHDL.

На выполнение лабораторной работы отводится восемь часов.

Перед выполнением лабораторной работы студент должен самостоятельно выполнить домашнее задание в соответствии с данными методическими указаниями.

Студент, явившийся на занятия, должен иметь методические указания по данной лабораторной работе. В начале занятия преподаватель проверяет выполнение студентом домашнего задания и наличие заготовки отчета по данной лабораторной работе в его рабочей тетради.

К выполненной работе прилагаются необходимые схемы, эскизы, тексты и результаты проектирования, протоколы работы с программным комплексом и другие материалы согласно указаниям по оформлению отчета.

2.2. Домашние задания и методические указания по их выполнению

Задание 1. Ознакомиться со структурой и принципом работы основных узлов комбинационного типа (двоичные

дешифраторы, мультиплексоры и демультимплексоры, компараторы и сумматоры, матричные умножители). Для этого необходимо воспользоваться лекциями по курсу и литературой [1, С. 73-95, С. 104-122].

Задание 2. Изучить методы построения иерархических блоки (ИБ) *FUB (Functional user block)*. ИБ применяются при проектировании схем в программном комплексе Active-HDL для тех же целей, что и символы, но в отличие от последних они имеют несколько специфических свойств. Есть возможность создавать и редактировать ИБ непосредственно на схеме, сохранять иерархический блок в библиотеке элементов, что позволяет уже построенный и отлаженный блок использовать при создании новых проектов. Такой подход к проектированию логических схем в значительной мере снижает трудоемкость и время разработки.

2.3. Вопросы к домашнему заданию

1.3.1. Пояснить схемотехническую реализацию дешифраторов.

1.3.2. Пояснить схемотехническую реализацию мультиплексоров.

1.3.3. Как осуществляется наращивание размерности дешифраторов?

1.3.4. Как осуществляется наращивание размерности мультиплексоров?

2.4. Лабораторные задания и методические указания по их выполнению

Задание 1. Построить и отладить иерархическую модель дешифратора «5-32» на логическом уровне с использованием программного комплекса Active-HDL на языке описания аппаратуры VHDL. Обозначение и схема дешифратора представлена на рис. 13 и 14 соответственно. Для данной схемы

необходимо построить временную диаграмму, демонстрирующую работу, а также полученную модель необходимо сохранить в виде стандартного элемента библиотеки.

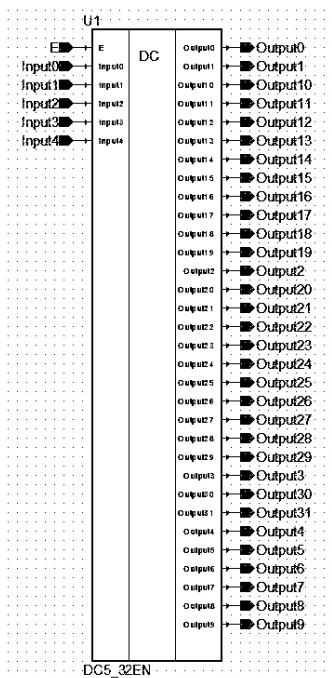


Рис. 13. Обозначение дешифратора «5-32»

Иерархическая модель дешифратора «5-32» строится на основе, полученных в первой лабораторной работе моделей простейших дешифраторов «2-4» и «3-8». Данная иерархическая модель должна также содержать разрешающий сигнал «Е».

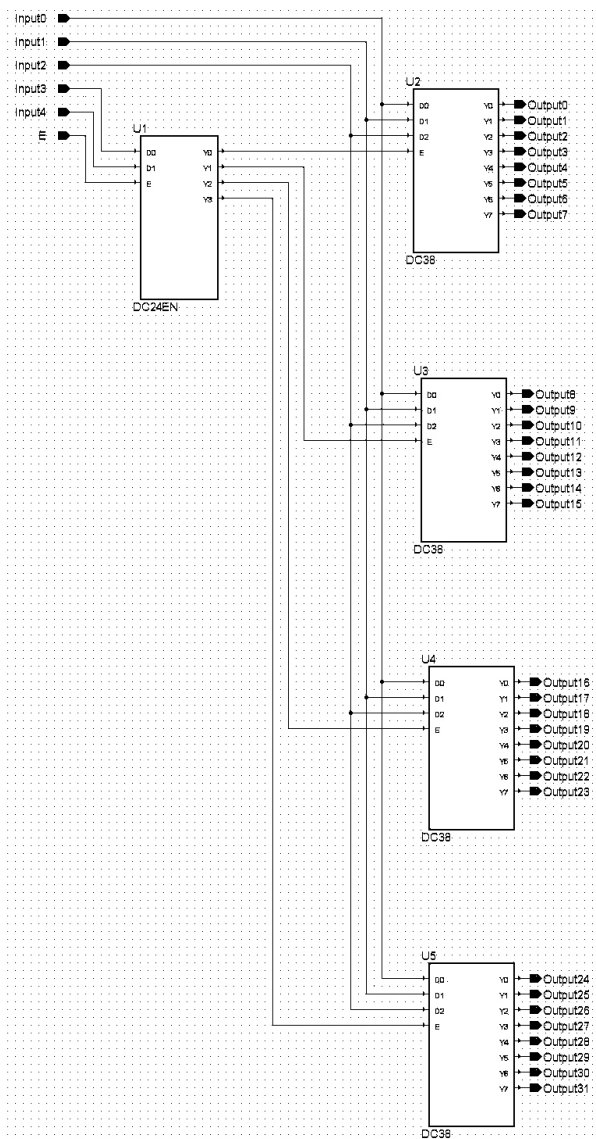


Рис. 14. Схема иерархической модели дешифратора «5-32»

Задание 2. Построить и отладить иерархическую модель мультиплексора «16-1» на логическом уровне с использованием программного комплекса Active-HDL на языке описания аппаратуры VHDL. Обозначение и схема мультиплексора представлена на рис. 15 и 16 соответственно. Для данной схемы необходимо построить временную диаграмму, демонстрирующую работу, а также полученную модель необходимо сохранить в виде стандартного элемента библиотеки.

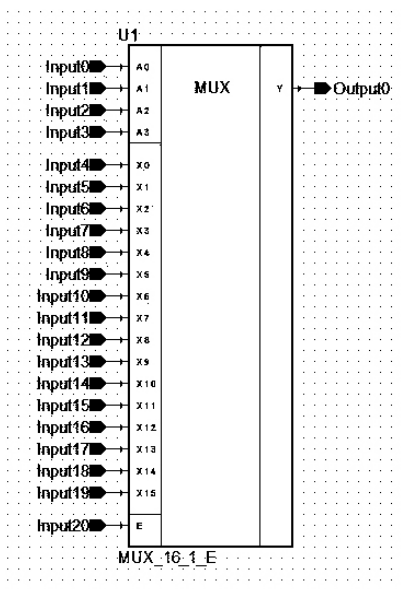


Рис. 15. Обозначение мультиплексора «16-1»

Иерархическая модель мультиплексора «16-1» строится на основе, полученной в первой лабораторной работе модели мультиплексора «4-1». Данная иерархическая модель должна также содержать разрешающий сигнал «Е».

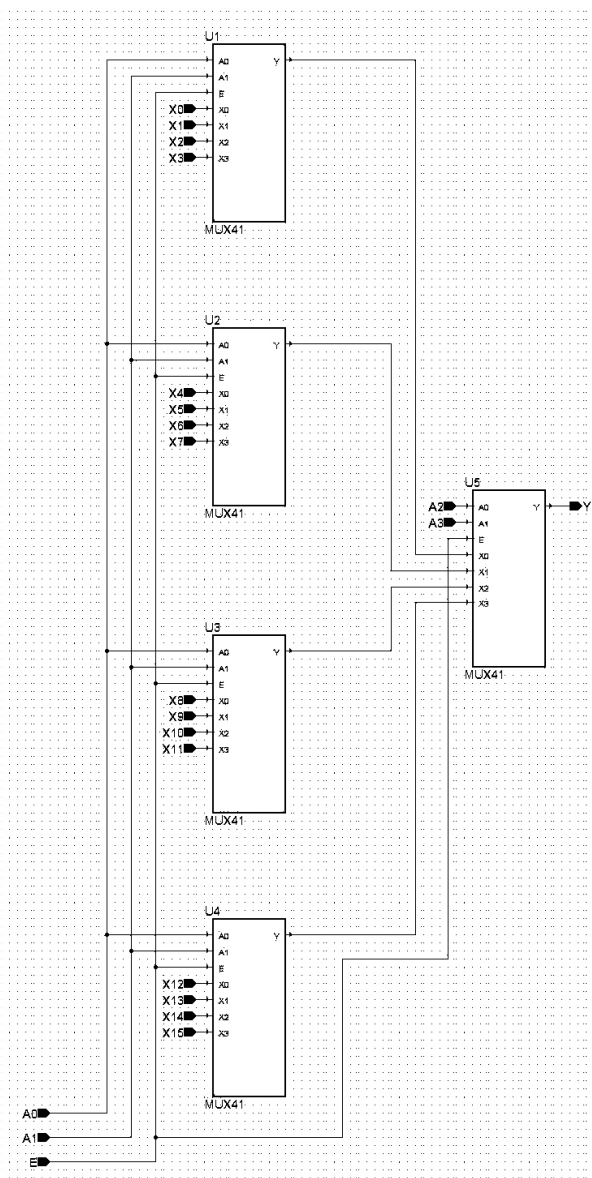


Рис. 16. Схема иерархической модели мультиплексора «16-1»

2.5. Контрольные вопросы

2.5.1. Какова цель лабораторной работы?

2.5.2. В чем заключается лабораторное задание?

Пояснить ход его выполнения.

2.5.3. Какие данные являлись исходными для выполнения работы?

2.5.4. Пояснить работу полученной иерархической модели дешифратора.

2.5.5. Пояснить работу полученной иерархической модели мультиплексора.

2.5.6. Сформулируйте выводы по данной лабораторной работе.

3. ЛАБОРАТОРНАЯ РАБОТА № 3

ПРОЕКТИРОВАНИЕ КОМПАРАТОРА КОДА С ИСПОЛЬЗОВАНИЕМ ИЕРАРХИЧЕСКИХ МОДУЛЕЙ И МНОГОРАЗРЯДНЫХ ШИН ДАННЫХ НА ЯЗЫКЕ VHDL

3.1. Общие указания по выполнению лабораторной работы

Целью лабораторной работы является углубление и закрепление знаний студентов в области работы, структуры и систем параметров функциональных узлов комбинационного типа, а также получение навыков в проектировании компараторов кода с использованием языка VHDL. В рамках данной лабораторной работы необходимо построить модели компаратора для двух и четырех разрядных слов с использованием иерархических блоков и многоразрядных шин данных на языке описания аппаратных средств VHDL с применением программного комплекса Active-HDL.

На выполнение лабораторной работы отводится восемь часов.

Перед выполнением лабораторной работы студент должен самостоятельно выполнить домашнее задание в соответствии с данными методическими указаниями.

Студент, явившийся на занятия, должен иметь методические указания по данной лабораторной работе. В начале занятия преподаватель проверяет выполнение студентом домашнего задания и наличие заготовки отчета по данной лабораторной работе в его рабочей тетради.

К выполненной работе прилагаются необходимые схемы, эскизы, тексты и результаты проектирования, протоколы работы с программным комплексом и другие материалы согласно указаниям по оформлению отчета.

3.2. Домашние задания и методические указания по их выполнению

Задание 1. Ознакомиться с принципом работы, назначением и схемными конструкциями компараторов. Для этого необходимо воспользоваться лекциями по курсу и литературой [1, С. 100-103].

Компараторы кодов применяются для сравнения двух входных кодов и выдачи на выходы сигналов о результатах этого сравнения (о равенстве или неравенстве кодов). Компараторы (устройства сравнения) определяют отношения между двумя словами. Основными отношениями, через которые можно выразить остальные, можно считать два – «равно» и «больше».

Если используется одиночная микросхема, то для ее правильной работы достаточно подать единицу на вход $A = B$, а состояния входов $A < B$ и $A > B$ не важны, на них можно подать как нуль, так и единицу. Назначение выходов понятно из их названия, а полярность выходных сигналов положительная (активный уровень – единица). Если микросхемы компараторов

кодов каскадируются (объединяются) для увеличения числа разрядов сравниваемых кодов, то надо выходные сигналы микросхемы, обрабатывающей младшие разряды кода, подать на одноименные входы микросхемы, обрабатывающей старшие разряды кода (см. рис. 17).

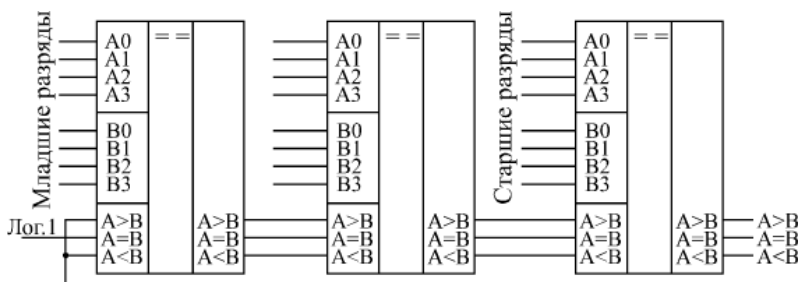


Рис. 17. Каскадирование компараторов кодов

Задание 2. Ознакомиться с методами трассировки в программном комплексе Active-HDL, с построением проводников и многоразрядных шин данных.

Способы построения проводников. Проводники в пакете Active-HDL 7.1 создаются двумя способами. Первый способ считается классическим, так как реализован во многих современных САПР. Он носит название «метод последовательных точек». Суть метода в том, что проектировщик щелкает ЛКМ в точках, где начинается проводник, где он меняет свое направление и где заканчивается. В процессе построения создается временная трасса, и есть возможность, двигаясь в обратном направлении по уже проложенному пути, удалять только что созданные сегменты, а для его завершения потребуется двойной щелчок мышью.

Второй метод можно назвать методом «прижатой кнопки»: в процессе построения проводника левая кнопка мыши остается все время в нажатом состоянии. Прижимается ЛКМ в точке, где необходимо начать проводник и, не отпуская ее, ведете указатель мыши в точку назначения. При смене

направления, в месте, где надо сделать угол (излом), следует нажать клавишу Space (пробел). Закончив создание проводника, просто отпустите кнопку мыши.

Проектирование шин. Шины размещаются на схеме по тем же правилам, что и проводники. Им требуется задавать кроме имени еще и ширину шины. Например, по умолчанию шины именуются как *BUS3(7:0)*. В круглых скобках указывается, что шина содержит восемь проводников, членов шины. Они имеют имена *BUS3(7)...* *BUS3(0)*. Не забывайте, что левая граница диапазона индексов, задающих ширину шины, всегда соответствует наиболее значащему разряду. В нашем примере старшим разрядом шины будет *BUS3(7)*.

К шине нельзя подключать проводники, не являющиеся членами шины. Для назначения имен шинам используются те же самые инструменты, что и для проводников. Поэтому мы и здесь обойдемся без особых комментариев.

3.3. Вопросы к домашнему заданию

3.3.1. Дать определение компаратора кода.

3.3.2. Пояснить принцип каскадного построения компараторов кода.

3.3.3. Пояснить способы построения проводников и многоразрядных шин данных.

3.4. Лабораторные задания и методические указания по их выполнению

Задание 1. Построить и отладить модель компаратора для двухразрядных слов. Схема данного компаратора представлена на рисунке ниже.

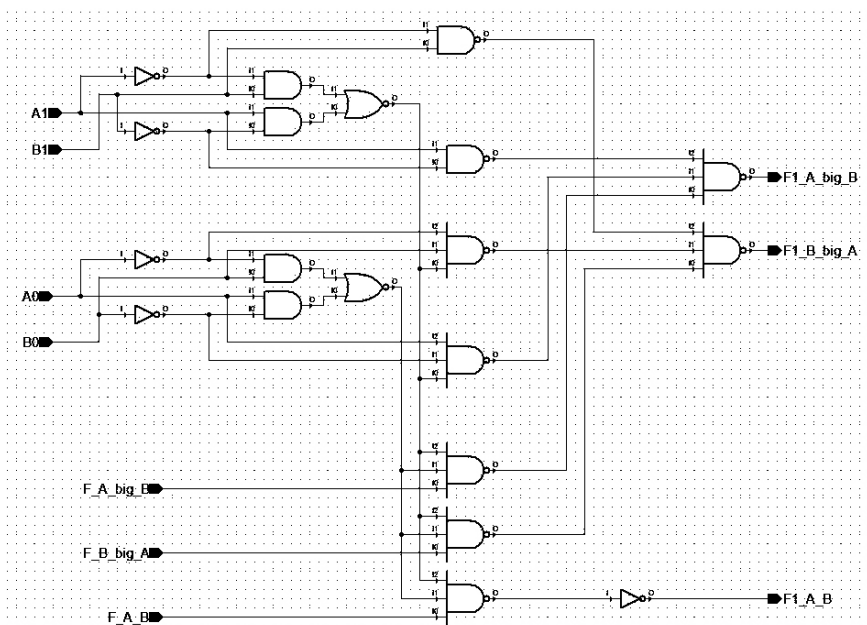


Рис. 18. Схема компаратора кодов для двухразрядных слов

Задание 2. Построить и отладить иерархическую модель компаратора кода для четырехразрядных слов, опираясь на пример приведенный в домашнем задании, с использованием программного комплекса Active-HDL на языке описания аппаратуры VHDL. Схема компаратора представлена на рис. 19. Для данной схемы необходимо построить временную диаграмму, демонстрирующую ее работу.

Данная иерархическая модель компаратора строится на основе, полученной в первом задании модели компаратора для двухразрядных слов. Сравниваемые слова поразрядно объединить шинами данных $A(3:0)$ и $B(3:0)$.

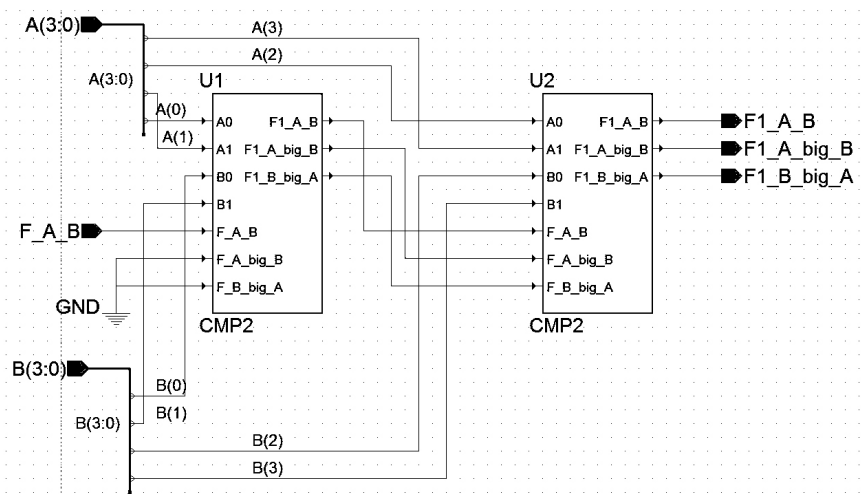


Рис. 19. Схема каскадной иерархической модели компаратора кодов для четырехразрядных слов

3.5. Контрольные вопросы

3.5.1. Какова цель лабораторной работы?

3.5.2. В чем заключается лабораторное задание?

Пояснить ход его выполнения.

3.5.3. Какие данные являлись исходными для выполнения работы?

3.5.4. Пояснить работу полученной модели компаратора.

3.5.5. Сформулируйте выводы по данной лабораторной работе.

4. ЛАБОРАТОРНАЯ РАБОТА № 4

ПРОЕКТИРОВАНИЕ СТРУКТУРЫ ОПЕРАТИВНОГО ЗАПОМИНАЮЩЕГО УСТРОЙСТВА СТАТИЧЕСКОГО ТИПА НА ЯЗЫКЕ VHDL

4.1. Общие указания по выполнению лабораторной работы

Целью лабораторной работы является углубление и закрепление знаний студентов в области проектирования цифровых устройств с использованием языков высокого уровня описания аппаратных средств. В рамках данной лабораторной работы необходимо построить на логическом уровне модель оперативного запоминающего устройства (ОЗУ) статического типа с использованием программного комплекса Active-HDL на языке описания аппаратуры VHDL.

На выполнение лабораторной работы отводится восемь часов.

Перед выполнением лабораторной работы студент должен самостоятельно выполнить домашнее задание в соответствии с данными методическими указаниями.

Студент, явившийся на занятия, должен иметь методические указания по данной лабораторной работе. В начале занятия преподаватель проверяет выполнение студентом домашнего задания и наличие заготовки отчета по данной лабораторной работе в его рабочей тетради.

К выполненной работе прилагаются необходимые схемы, эскизы, тексты и результаты проектирования, протоколы работы с программным комплексом и другие материалы согласно указаниям по оформлению отчета.

4.2. Домашние задания и методические указания по их выполнению

Задание 1. Ознакомиться с основными типами, структурами и системами параметров запоминающих устройств (ЗУ). Для этого необходимо воспользоваться лекциями по курсу и литературой [1, С. 227-247].

Задание 2. Рассмотреть структуру и организацию ОЗУ статического типа.

Для выполнения данного задания за основу примем структуру ОЗУ статического типа на 16 одноразрядных слов. Данное ОЗУ реализовано по схеме двухкоординатной выборки. Схема получена при помощи двух дешифраторов адреса выбора столбцов DC_y и строк DC_x матрицы запоминающих ячеек (ЗЯ), на пересечении которых и расположена искомая запоминающая ячейка памяти (см. рис. 20).

Запоминающий элемент (ЗЭ) ОЗУ предназначен для хранения 1 бита информации, а запоминающая ячейка – 1 слова. Слово в данном случае примем одноразрядное, соответственно разрядность ячейки памяти – 1 бит. Запоминающий элемент представляет собой RS-триггер, в данном случае на элементах ИЛИ-НЕ (см. рис. 21).

Триггер имеет два входа – установки в единицу (S) и установки в ноль (R), два выхода, но в данном случае необходим один X_{out} .

Схема запоминающей ячейки на одно слово представлена на рис. 22. Рассмотрим далее сигналы и блоки данной ячейки.

A_x, A_y – адресные входы соответственно дешифраторов строк и столбцов матрицы;

CS – сигнал разрешающий или запрещающий работу ЗЯ;

RW – сигнал чтение/запись ЗЯ (при подаче «1» – чтение, при «0» – запись);

$RESET$ – сброс, установка ЗЯ в ноль;

X_{in}, X_{out} – соответственно вход и выход ячейки памяти;

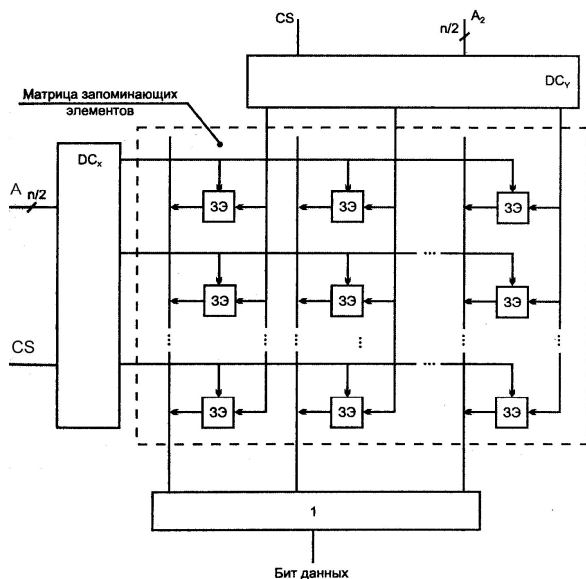


Рис. 20. Структура двухкоординатной выборки ОЗУ статического типа

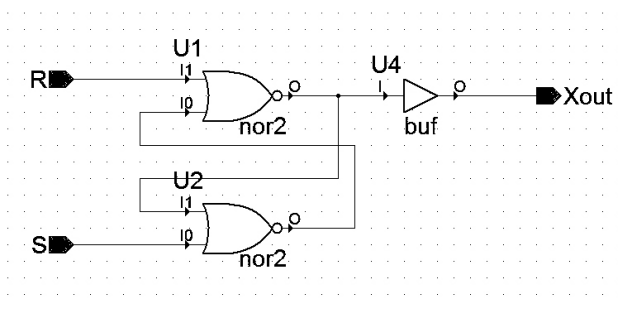


Рис. 21. Схема запоминающего элемента

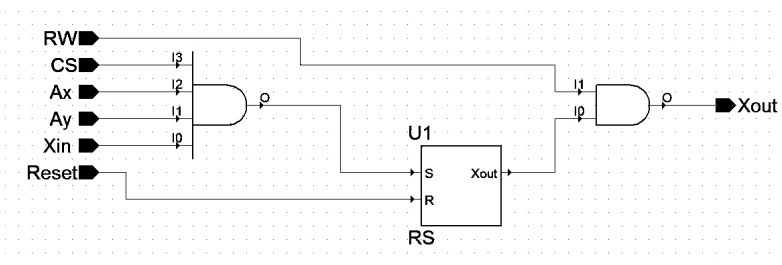


Рис. 22. Запоминающая ячейка ОЗУ статического типа с двухкоординатной выборкой

Запись в ячейку осуществляется при наличии единиц на Ax , Ay и CS . Слово здесь продолжает храниться до прихода следующей команды. Чтение осуществляется при наличии единицы на RW , что приводит к срабатыванию блока *and4* и прохождению слова на выход ячейки памяти $Xout$. Обнуление ЗЯ происходит при подачи на CS «нуля», а на $RESET$ «единицы».

4.3. Вопросы к домашнему заданию

- 1.3.1. Перечислить основные типы и параметры ЗУ.
- 1.3.2. Статические ОЗУ и их классификация.
- 1.3.3. Какие основные структуры запоминающих устройств вы знаете?

4.4. Лабораторные задания и методические указания по их выполнению

Задание 1. На завершающем этапе лабораторного практикума осуществляется построение модели адресного ОЗУ статического типа. Студенты должны построить, отладить и продемонстрировать работоспособность модели ОЗУ на логическом уровне используя полученные в ходе лабораторных

работ модели отдельных ее модулей, опираясь на пример приведенный в домашнем задании.

В первом задании необходимо реализовать модель ОЗУ статического типа построенного по схеме однокоординатной выборки (см. рис. 24). Для решения данной задачи используется запоминающая ячейка с одним адресным входом (см. рис. 23).

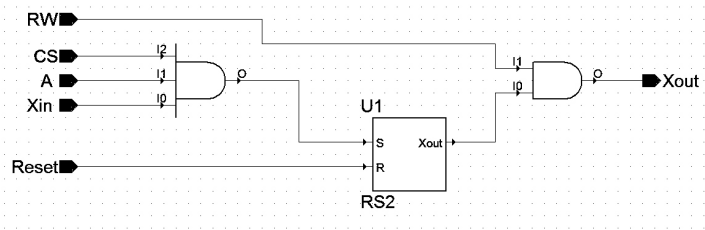


Рис. 23. Запоминающая ячейка ОЗУ статического типа с однокоррдинатной выборкой

В качестве дешифратора адреса используется построенный ранее дешифратор «2-4» с разрешающим входом «Е». Спроектированную ячейку проверить на работоспособность, последовательно выполняя итерации, приведенные в таблице ниже.

Порядок работы запоминающей ячейки

Вход	Итерация 1	Итерация 2	Итерация 3	Итерация 4	Итерация 5	Итерация 6
Reset	1	0	0	0	1	0
RW	0	1	0	1	0	1
CS	0	0	1	0	0	0
A	0	0	1	0	0	0
Xin	0	0	1	0	0	0
	Сброс ячейки	Режим чтения	Режим записи	Режим чтения	Сброс ячейки	Режим чтения

Задание 2. Опираясь на схему двухкоординатной выборки, представленной в домашнем задании, необходимо построить модель с использованием 16 запоминающих ячеек

(см. рис. 25). Ячейки памяти в данном случае будут соответствовать схеме рис. 22 и иметь два адресных входа.

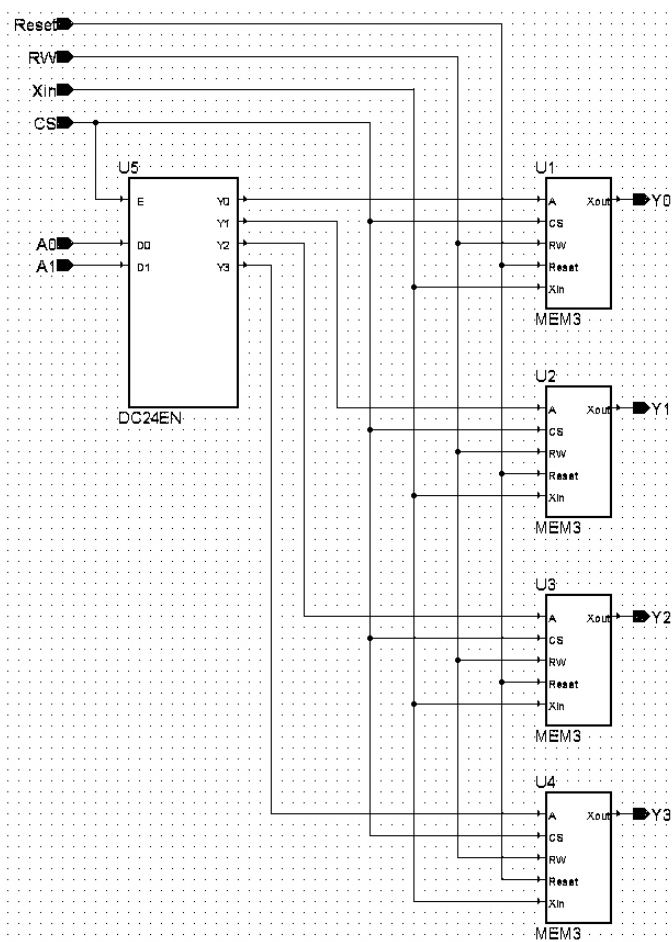


Рис. 24. Схема ОЗУ статического типа на 4 запоминающие ячейки

В качестве двух дешифраторов адреса используется построенные ранее дешифраторы «2-4» с разрешающим входом «Е». Полученные модели ОЗУ необходимо откомпилировать и проверить их работоспособность. Реализовать циклы

запись/чтение запоминающих ячеек в соответствии со следующими итерациями: Сброс (*RESET*), включение схемы (*CS*), адресная запись 1 бита информации в каждую ячейку памяти, адресное считывание данных, сброс ячеек (*RESET*).

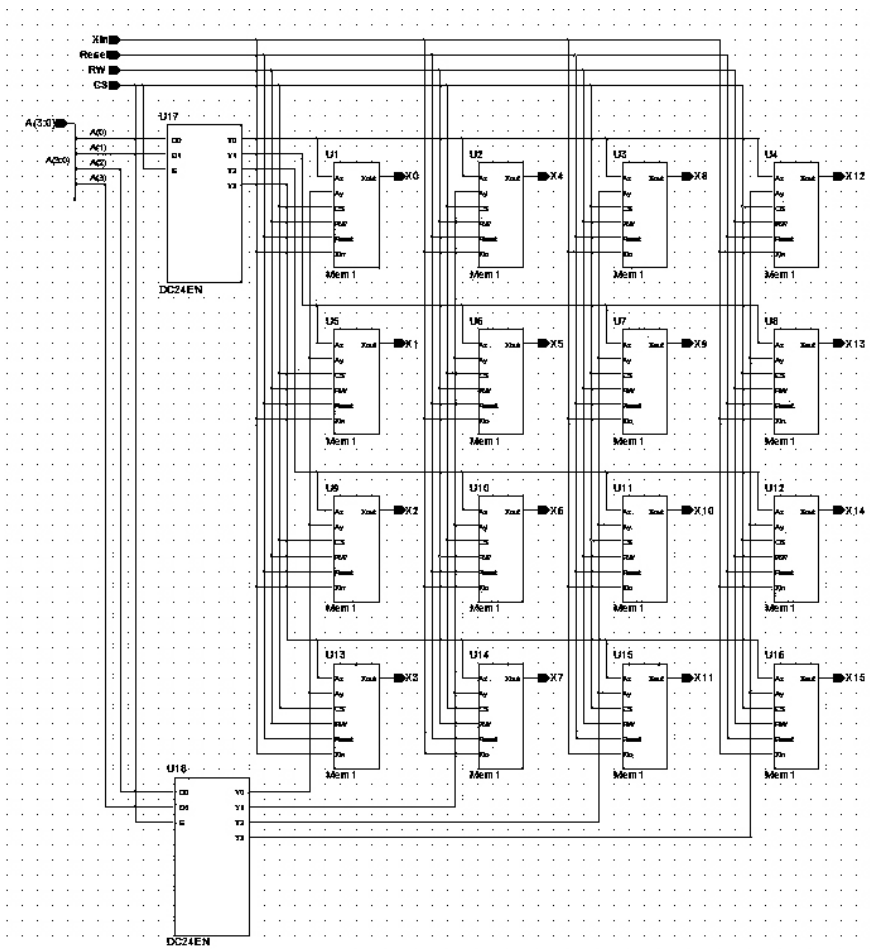


Рис. 25. Схема ОЗУ статического типа на 16 запоминающих ячейках

4.5. Контрольные вопросы

1.5.1. Какова цель лабораторной работы?

1.5.2. В чем заключается лабораторное задание?

Пояснить ход его выполнения.

1.5.3. Какие данные являлись исходными для выполнения работы?

1.5.4. Пояснить работу полученных моделей ОЗУ, продемонстрировать запись/чтение данных.

1.5.5. Сформулируйте выводы по данной лабораторной работе.

УКАЗАНИЯ ПО ОФОРМЛЕНИЮ ОТЧЁТА

Отчет по каждой лабораторной работе должен содержать наименование и цель работы, краткие теоретические сведения, ход и результаты выполнения лабораторного задания, где приводятся исходные данные и результаты работы с программным комплексом с необходимыми пояснениями. Отчет завершается кратким перечнем приобретенных навыков и выводами о результатах проделанной работы. Оформление отчета выполняется в соответствии со стандартом ВГТУ.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Угрюмов, Е.П. Цифровая схемотехника [Текст] : учеб. пособие / Е.П. Угрюмов. – 2-е изд., перераб. и доп. – СПб. : БХВ-Петербург, 2004. – 800 с.
2. Солонина, А.И. Основы цифровой обработки сигналов [Текст] : учеб. пособие / А.И. Солонина, Д.А. Ухладович, С.М. Арбузов, Е.Б. Соловьева. – СПб. : БХВ-Петербург, 2005. – 768 с.
3. Солонина, А. И. Алгоритмы и процессоры цифровой обработки сигналов [Текст] : учеб. пособие / А.И. Солонина, Д.А. Улахович, Л.А. Яковлев. – СПб. : БХВ-Петербург, 2002. – 464 с.

СОДЕРЖАНИЕ

Лабораторная работа № 1

Проектирование функциональных узлов комбинационного типа на языке VHDL 1

Лабораторная работа № 2

Проектирование иерархических модулей на языке VHDL 13

Лабораторная работа № 3

Проектирование компаратора кода с использованием иерархических модулей и многоразрядных шин данных на языке VHDL 19

Лабораторная работа № 4

Проектирование структуры оперативного запоминающего устройства статического типа на языке VHDL 25

Указания по оформлению отчёта 33

Библиографический список 34

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

к выполнению лабораторных работ № 1-4
по дисциплине «Специализированные БИС и устройства
функциональной электроники в приборостроении»
для студентов направления 12.03.01 «Приборостроение»
очной и заочной форм обучения

Составитель
Пирогов Александр Александрович

В авторской редакции

Подписано к изданию 20.10.2014.
Уч.-изд. л. 2,1.

ФГБОУ ВПО «Воронежский государственный
технический университет»
394026 Воронеж, Московский просп., 14