

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

**БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ
УНИВЕРСИТЕТ ТРАНСПОРТА**

Кафедра автоматике и телемеханики

К. А. БОЧКОВ, Ю. Ф. БЕРЕЗНЯЦКИЙ

АНАЛИЗ ФУНКЦИОНАЛЬНОЙ СТРУКТУРЫ И СИНТЕЗ ДИСКРЕТНЫХ УСТРОЙСТВ

**Лабораторный практикум
по дисциплине «Теория дискретных устройств»**

Часть I

Гомель 2001

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ
УНИВЕРСИТЕТ ТРАНСПОРТА

Кафедра автоматики и телемеханики

К. А. БОЧКОВ, Ю. Ф. БЕРЕЗНЯЦКИЙ

АНАЛИЗ ФУНКЦИОНАЛЬНОЙ СТРУКТУРЫ И СИНТЕЗ ДИСКРЕТНЫХ УСТРОЙСТВ

Лабораторный практикум
по дисциплине «Теория дискретных устройств»

Часть I

Одобен методическими комиссиями
факультета безотрывного обучения и электротехнического факультета

Гомель 2001

УДК 656.25:681.32 (075.8)
Б 866

Бочков К. А., Березняцкий Ю. Ф.

Б 866 Анализ функциональной структуры и синтез дискретных устройств: Лабораторный практикум по дисциплине «Теория дискретных устройств». Ч. I / Белорус. гос. ун-т трансп. – Гомель: БелГУТ, 2001. – 32 с.

Рассматриваются вопросы анализа функциональной структуры и синтеза дискретных устройств с применением программы «Электронная лаборатория».

Предназначен для студентов третьего курса факультета безотрывного обучения и студентов второго курса электротехнического факультета, обучающихся по специальности Т 11.05 «Автоматика, телемеханика и связь на транспорте».

Р е ц е н з е н т – канд. техн. наук, доцент кафедры «Микропроцессорная техника и информационно-управляющие системы» БелГУТа **С. Н. Харлап**

ВВЕДЕНИЕ

Целью данного цикла лабораторных работ является изучение принципов построения, логической структуры и функциональных особенностей дискретных устройств на базе программы «Электронная лаборатория» (“*Electronics Workbench*”) фирмы “*Interactive Image Technologies*”, а также получение практических навыков анализа и синтеза дискретных устройств различного назначения.

Рассмотрены способы задания функций алгебры логики, преобразование базисов дискретных устройств, получение канонических форм функций алгебры логики, структуры и способы функционирования различных дискретных устройств, их принципы построения.

Лабораторная работа № 1

ИЗУЧЕНИЕ ПРОГРАММЫ «ЭЛЕКТРОННАЯ ЛАБОРАТОРИЯ»

Ц е л ь р а б о т ы. Изучить основные элементы и конструкции программы «Электронная лаборатория», предназначенные для синтеза и анализа дискретных схем, и научиться строить с их помощью простейшие дискретные структуры.

1 Краткие сведения из теории

1.1 Характеристика программы

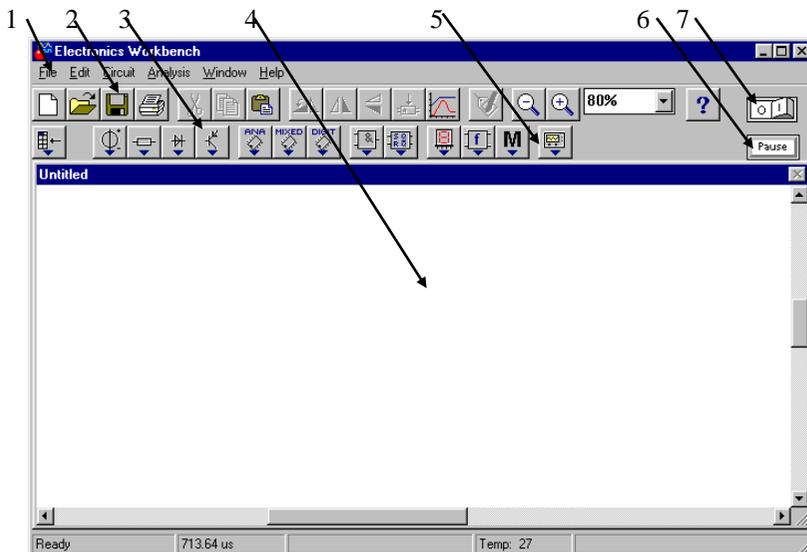
Программа «Электронная лаборатория» предназначена для синтеза и анализа дискретных и аналоговых схем на основе стандартных компонентов, входящих в базовый набор программы, а также при применении созданных пользователем блоков.

Для работы с данной программой не требуется практически никакой подготовки, так как программа имеет интуитивный интерфейс, понятный даже новичку. Однако для работы с дискретной частью программы требуются хотя бы начальные знания по цифровым логическим системам, по логическим элементам И, ИЛИ, НЕ, И-НЕ, ИЛИ-НЕ, триггерам и т. д.

Программа разработана для работы в среде *Windows*.

1.2 Элементы управления

Меню программы (рисунок 1.1) предназначено для выбора действий пользователя по работе с файлами, справочной системой, распечаткой созданных схем, импортом-экспортом данных, редактированием схем, операциями с окнами и др.



1 – панель меню программы; 2 – панель с кнопками наиболее часто используемых команд; 3 – панель базовых элементов программы с ниспадающими меню; 4 – поле для рисования схем; 5 – панель инструментов программы с ниспадающими меню; 6 – кнопка «Пауза»; 7 – кнопка «Запуска/Остановки» процесса моделирования

Рисунок 1.1 – Примерный вид программы «Электронная лаборатория» при запуске

Рассмотрим основные команды программы, с которыми придется оперировать при анализе и синтезе дискретных схем.

Раздел меню «*File*» содержит в себе такие основные команды, как:

- 1) *New* – создать новый файл;
- 2) *Open* – открыть существующий файл;
- 3) *Save* – записать файл;
- 4) *Save as...* – записать файл с другим именем;
- 5) *Revert to Saved* – вернуться к записанному файлу, т. е. отмена всех исправлений в файле;
- 6) *Print* – распечатать файл на принтере;
- 7) *Print setup* – настройка печати;

8) *Exit* – выход из программы и др.

Меню «*Edit*» содержит такие основные команды:

- 1) *Cut* – вырезать, т. е. удаление и перенос выделенного фрагмента схемы в буфер обмена;
- 2) *Copy* – копирование выделенного фрагмента в буфер;
- 3) *Paste* – вставка из буфера фрагмента схемы;
- 4) *Delete* – удаление выделенного фрагмента;
- 5) *Select All* – выделить всю схему для копирования, переноса или удаления;
- 6) *Show Clipboard* – показать содержимое буфера обмена и др.

Меню «*Circuit*» содержит следующие команды:

- 1) *Rotate* – повернуть элемент на 90 градусов по часовой стрелке, повторный выбор – еще на 90 градусов и т. д.;
- 2) *Flip Horizontal* – отражение элемента в горизонтальной плоскости;
- 3) *Flip Vertical* – то же в вертикальной плоскости;
- 4) *Component Properties* – свойства компонента;
- 5) *Create Subcircuit* – создать из выделенного фрагмента схемы блок;
- 6) *Zoom In* – увеличение масштаба;
- 7) *Zoom Out* – уменьшение масштаба;
- 8) *Schematic Options* – опции схемы;
- 9) *Restrictions* – ограничения на модель.

Остальные меню практически не требуются для работы: это команды вызова справки – «*Help*», работы с окнами – «*Window*», опций анализа схемы – «*Analysis*».

1.3 Создание схем

При построении схемы требуется выбрать пиктограмму необходимой группы базовых элементов на панели базовых элементов и щелкнуть на ней левой кнопкой мыши. При этом раскроется ниспадающее меню, в котором можно выбрать необходимый тип элемента, и, удерживая левую кнопку мыши нажатой, перетащить требуемый элемент из ниспадающего меню поля базовых элементов на поле для рисования схем, при этом элемент появляется на поле схемы.

Для того чтобы соединить два элемента между собой, нужно подвести курсор мыши к выводу первого элемента, нажать левую кнопку мыши (при этом у вывода элемента появится небольшой квадратик) и, удерживая кнопку мыши нажатой, перетащить указатель мыши к выводу другого элемента. При этом будет вычерчиваться линия слежения за курсором мыши, и вокруг вывода другого элемента появится такой же небольшой квадратик. Затем кнопка мыши отпускается и провод, соединяющий два вывода, готов. Он будет нарисован в поле схемы.

Для удаления элемента из схемы достаточно навести курсор мыши на

элемент, который требуется удалить, и нажать левую кнопку мыши, при этом элемент окрасится в красный цвет. После чего нажимается клавиша «Delete» («Del») или выбирается пункт «Delete» в меню «Edit» и элемент удаляется из схемы.

В программе также можно задать имя элементу или блоку. В этом случае следует выделить элемент (блок) и выбрать в меню «Circuit» команду «Component Properties» или два раза «щелкнуть» левой кнопкой мыши на выбранном элементе (блоке). После этого появится меню «...Properties», в котором требуется выбрать закладку «Label» и в поле «Label» набрать задуманное имя элемента (желательно английскими буквами). Затем нажимается кнопка “OK”, и название элемента появляется над ним.

Если требуется задать ошибки в работу какого-либо элемента, то в окне «...Properties», вызванном, как и в предыдущем случае, выбирается закладка «Fault». В этой закладке можно выбрать такие неисправности, как: короткое замыкание указанных цифрами выводов элемента (*Short*), обрыв на заданных цифрами выводах (*Open*), задать величину сопротивления утечки (*Leakage*) либо выбрать отметку, обозначающую отсутствие ошибок (*None*).

Для изменения цвета проводника, соединяющего элементы, надо дважды щелкнуть на нем левой кнопкой мыши. При этом появится меню “Wire Properties”, в котором нужно нажать кнопку “Set Wire Color” и в появившемся подокне выбрать желаемый цвет проводника. После этого нажать «OK».

1.4 Создание блоков элементов

Программа «Электронная лаборатория» дает возможность упрощения построения схем путем создания необходимых блоков.

Для создания блока следует выделить схему (участок схемы), которую необходимо поместить в блок. Это можно выполнить следующим образом: за пределами схемы нажимается левая кнопка мыши и, не отпуская ее, указатель мыши двигается по диагонали. При этом образуется рамка и выделяются красным цветом все элементы, попавшие в нее.

Затем выбирается команда «Create Subcircuit» в меню «Circuit» или нажимается сочетание клавиш «Ctrl» + «B». После чего появится меню, в котором следует набрать имя создаваемого блока и нажать одну из предлагаемых клавиш «Copy From Circuit» (копировать из схемы), «Move From Circuit» (переместить из схемы), «Replace in Circuit» (заменить в схеме), «Cancel» (отменить). В зависимости от выбранного варианта часть схемы, выбранная для блока, будет либо скопирована из схемы, либо перемещена из нее, либо заменена блоком с указанным именем. Блок будет представлен отдельным окном, которое может быть закрыто путем нажатия кнопки с крестиком в правом верхнем углу (рисунок 1.2). После закрытия окна блока он помещается в специальной области, под названием «Favorites», из которой блок можно извлечь и поместить на поле схемы. Для

этого следует нажать кнопку «*Favorites*», которая размещается первой слева на панели базовых элементов программы. После нажатия кнопки, появится окошко с элементом «*Sub*», на котором следует нажать левую кнопку мыши и, удерживая ее нажатой, переместить мышь на область рисования схемы. Затем надо отпустить кнопку. В результате появится меню «*Choose SUB*», в котором надо выбрать имя созданного ранее блока и нажать кнопку «*Accept*». На поле для рисования схем появится небольшой прямоугольник с выбранным именем. Это и есть сформированный ранее блок. Для того чтобы его рассмотреть, на прямоугольнике можно выполнить двойной щелчок левой кнопкой мыши, и блок развернется в окно со схемой. В этом окне можно сделать необходимые выводы для подключения схемы. Для этого надо подвести курсор к выводу элемента и, когда появится окно выделения на выводе, удерживая левую кнопку мыши, вывести курсор за пределы окна в том месте, где требуется вывод. При этом на границе окна появится небольшой квадратик, и, если закрыть окно, отобразится прямоугольник с выводами, к которым можно подключать другие элементы.

Блокам можно назначать любые имена. В этом случае можно будет создать целую библиотеку цифровых микросхем и цифровых устройств.

На рисунке 1.2 приведен пример создания блока из трех логических элементов с четырьмя выводами для подключения под именем «*Blok*».

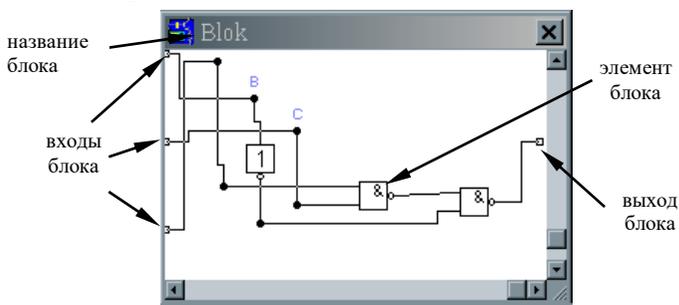


Рисунок 1.2 – Развернутое окно созданного блока

1.5 Инструменты анализа логических схем

Программа имеет в своем составе четыре прибора, которые будут использованы нами для анализа дискретных схем.

1. Мультиметр (*Multimeter*) предназначен для измерения напряжений, токов, сопротивлений и уровней в различных точках схемы (рисунок 1.3). Например, для измерения напряжений на выводах схемы его достаточно перетащить с помощью мыши на поле рисования схем и подключить к исследуемой цепи. Для этого у мультиметра на корпусе есть два вывода: «+» и «-». Чтобы увидеть показания мультиметра, надо два раза быстро нажать левую кнопку мыши на нем.



Рисунок 1.3 – Мультиметр в развернутом виде

2. Генератор кодов (*Word Generator*) служит для выработки различных кодовых последовательностей с числом разрядов до восьми в различных режимах (рисунок 1.4). Генератор имеет выход для тактовых импульсов (вывод “*Clk*”) и восемь выходов для кода. Дополнительно генератор снабжен входом для разрешения работы или запуска генератора по переднему либо по заднему фронту импульса. Для этого на генераторе имеются

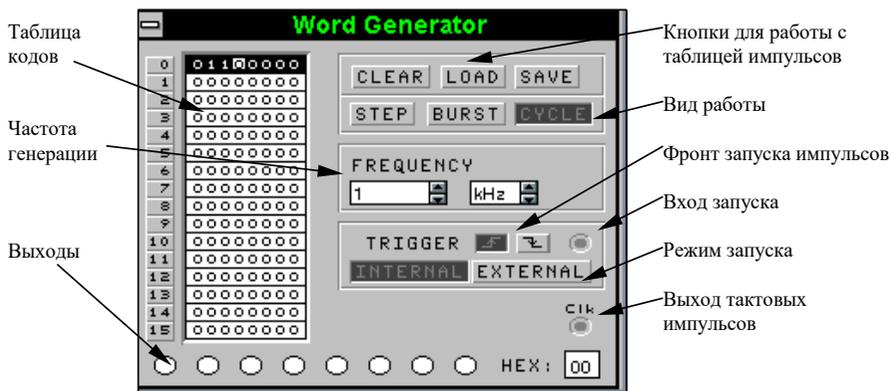


Рисунок 1.4 – Генератор кодовых последовательностей в развернутом виде соответствующие кнопки.

Кнопка “*INTERNAL*” предназначена для запуска генератора от внутреннего источника тактовых импульсов, а кнопка “*EXTERNAL*” - от внешнего источника. Кнопка “*CLEAR*” предназначена для обнуления таблицы, кнопка “*LOAD*” – загрузки ранее записанной таблицы, а “*SAVE*” – для сохранения созданной пользователем таблицы. Кнопка “*STEP*” предназначена для пошаговой генерации импульсов на выходах, кнопка “*BURST*” – для генерации всего набора импульсов и кнопка “*CYCLE*” – для непрерывной генерации набора таблицы импульсов.

Для изменения генерируемых последовательностей надо подвести курсор мыши к требуемому элементу и нажать левую кнопку мыши, при этом элемент выделится, и тогда можно записать логическую “1” или логический “0”. Для быстрого перемещения в таблице можно воспользоваться курсорными клавишами. После создания или редактирования таблицы её можно сохранить под своим именем, нажав с помощью мыши кнопку прибора “*SAVE*”.

3. Логический анализатор (*Logic Analyzer*) - восьмиканальный анализатор логических уровней (рисунок 1.5). Прибор предназначен для визуального наблюдения логических уровней в восьми точках схемы.

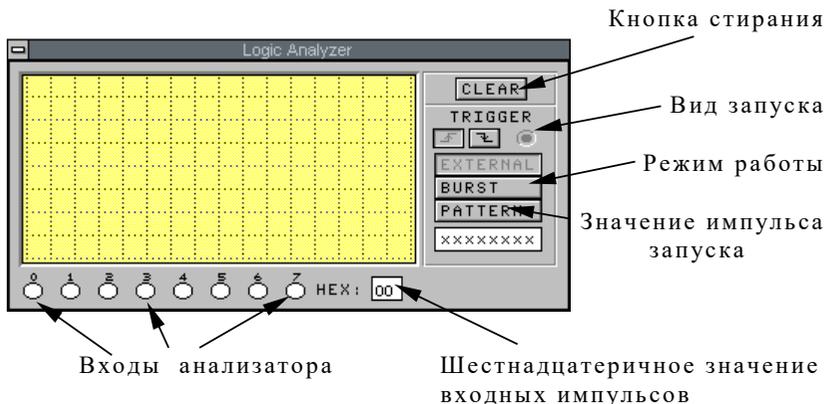


Рисунок 1.5 - Логический анализатор

В анализаторе можно установить вид запуска либо по переднему, либо по заднему фронту импульса, подаваемого на вход запуска “*TRIGGER*”, либо от внутреннего генератора. Кнопка “*EXTERNAL*” предназначена для запуска анализатора от внешнего источника, кнопка “*BURST*” - для полного заполнения экрана анализатора, кнопка “*PATTERN*” - для включения анализатора при поступлении на его входы заданного набора импульсов, при этом “х” - любое значение на данном входе. Для задания набора надо нажать левую кнопку мыши на нужном разряде и изменить этот разряд на логическую “1”, логический “0” либо оставить безразличное состояние “х”.

4. Логический конвертер (преобразователь) (*Logic Converter*) используется для преобразования или нахождения таблицы истинности схемы, преобразования таблицы истинности в функцию алгебры логики, преобразования таблицы истинности в упрощенную логическую формулу, преобразования логической формулы в таблицу истинности, а также для преобразования логической формулы в логическую схему в базисе “И-ИЛИ-НЕ” либо в базисе “И-НЕ” и др.

Для создания таблицы истинности требуется нажать кнопку мыши требуемое количество раз (до восьми) в кружках, обозначенных латинскими буквами “*A, B, C, D, E, F, G и H*”. При этом в поле значений входов появятся последовательные многоразрядные двоичные числа. Чтобы уменьшить число разрядов, следует также нажать левую кнопку мыши в кружках требуемых входов. После этого можно подвести курсор мыши к нужному разряду, нажать левую кнопку мыши, при этом логическое значение выделится квадратиком, и изменить значение на “1” или “0”.

После этого таблицу истинности можно преобразовать в логическую формулу или упрощенную логическую формулу, логическую формулу можно преобразовать в схему в базисе “И-ИЛИ-НЕ” или базисе “И-НЕ”, нажав левой кнопкой мыши на определенную кнопку логического преобразователя.

Для преобразования схем нужно подключить входы схем к входам логического преобразователя, а выход схемы – к выводу “*OUT*” логического преобразователя, после этого можно преобразовывать схему. Для этого надо подвести курсор мыши к прибору, нажав и удерживая левую кнопку мыши, перетащить его в нужное место схемы.

Более подробно работа с логическим конвертером будет изучена в лабораторной работе № 2.

2 Порядок выполнения работы

1 Ознакомиться по данным методическим указаниям с правилами работы и основными базовыми элементами программы «Электронная лаборатория».

2 Запустить программу в среде «*Windows*» из командной строки с ключом “*/din*”.

3 Просмотреть состав и научиться работать со всеми компонентами программы.

4 Нарисовать дискретную схему, заданную преподавателем, и выполнить ее анализ, используя средства программы «Электронная лаборатория».

5 Сделать выводы по работе, в которых отразить особенности анализа дискретных схем при помощи программы «Электронная лаборатория».

Содержание отчета

Наименование и цель работы, краткая характеристика программы «Электронная лаборатория», перечень основных элементов анализа программы с кратким их описанием, схема заданного преподавателем блока (построенная средствами программы «Электронная лаборатория»), ответы на заданные преподавателем контрольные вопросы, выводы по работе.

Контрольные вопросы

1 Назначение программы «Электронная лаборатория».

2 Для чего служит логический анализатор в программе?

3 Назначение мультиметра в программе.

4 Выделите особенности логического конвертера.

5 Опишите процедуру создания блока пользователя.

6 Какие Вам известны программные средства для анализа дискретных

схем кроме программы «Электронная лаборатория»? Дайте их краткую характеристику.

7 Охарактеризуйте основные элементы управления программой.

8 В чем заключаются особенности создания дискретных схем при помощи программы «Электронная лаборатория»?

Лабораторная работа № 2

МИНИМИЗАЦИЯ ФУНКЦИЙ АЛГЕБРЫ ЛОГИКИ И ПОСТРОЕНИЕ ДИСКРЕТНЫХ СХЕМ С ИСПОЛЬЗОВАНИЕМ ЛОГИЧЕСКОГО КОНВЕРТЕРА ПРОГРАММЫ «ЭЛЕКТРОННАЯ ЛАБОРАТОРИЯ»

Ц е л ь р а б о т ы. Научиться минимизировать функции алгебры логики (ФАЛ), получать совершенную дизъюнктивную нормальную форму (СДНФ) по таблице истинности (ТИ), строить дискретные схемы по заданным ФАЛ в различных базах, а также изучить способы задания ФАЛ.

1 Краткие сведения из теории

1.1 Способы задания функций алгебры логики

Функцию $f(X_1, X_2, \dots, X_n)$ называют функцией алгебры логики, если она, как и ее переменные, может принимать только два значения: логический 0 и логическую 1. Переменные ФАЛ сопоставляют со значениями сигналов на входах дискретного устройства (ДУ), а значения функции алгебры логики - со значениями сигналов на его выходах.

Реальные ДУ имеют конечное число входов, поэтому число переменных у соответствующих ФАЛ также конечно.

Существует ряд способов задания ФАЛ [1, с.27 - 31]:

1) **т а б л и ч н ы й** - ФАЛ задается таблицей ее значений в зависимости от значений переменных. Совокупность значений называют входным набором. Каждому набору переменных соответствует определенное значение функции. Каждый набор представляет собой n -разрядное двоичное число. При этом общее число наборов составляет $k = 2^n$. Таблицу, в которой для всех наборов переменных приводятся значения ФАЛ, называют таблицей истинности. При количестве переменных, равном n , таблица содержит 2^n строк (по числу наборов), n столбцов (по числу переменных) и один столбец значений функции;

2) **г р а ф и ч е с к и й** - наборам значений переменных ФАЛ сопоставляются точки n -мерного пространства. Множество 2^n наборов определяет множество вершин n -мерного единичного куба. Вершинам куба соответствуют наборы значений переменных и приписаны значения

функции на этих наборах, т. е. областью определения ФАЛ, зависящей от n переменных, является множество вершин единичного n -мерного куба. Куб называют единичным, так как каждое его ребро соединяет вершины, наборы которых различаются одной переменной;

3) **к о о р д и н а т н ы й** - функцию задают в виде координатной карты состояний, которую называют картой Карно. Карты представляют собой прямоугольные таблицы, разделенные горизонтальными и вертикальными линиями на клетки. Общее число клеток соответствует числу наборов функции. Все переменные функции разбивают на две группы. Одна группа переменных определяет выбор строки, другая - столбца. На пересечении строки и столбца находится клетка, в которую записывают значение функции при соответствующем наборе переменных. Разделение переменных на группы выполняется таким образом, чтобы в соседних клетках наборы различались только значением одной переменной;

4) **ч и с л о в о й** - каждому набору переменных ставят в соответствие определенное число в двоичной системе исчисления и присваивают соответствующий номер. Переменным X_1, X_2, \dots, X_n приписывают соответственно веса $2^{n-1}, 2^{n-2}, \dots, 2^1, 2^0$. Функцию задают в виде десятичных номеров тех наборов переменных, на которых она принимает значение 1;

5) **а н а л и т и ч е с к и й** - функция задается в виде алгебраического выражения, показывающего, какие и в какой последовательности должны выполняться логические операции над аргументами функции. Алгебраическое выражение может быть составлено из наборов аргументов, на которых функция принимает значение 1, или из наборов, на которых она принимает значение 0.

Элементарные функции одной или двух переменных реализуются отдельными логическими элементами. Совокупность логических элементов может реализовать различные ФАЛ в зависимости от схемы соединения их между собой.

В устройствах автоматики, телемеханики и связи применяют большое количество ДУ, характеризующихся различными законами функционирования, т. е. реализующих различные ФАЛ. Важным этапом синтеза ДУ является определение способа соединения между собой логических элементов, обеспечивающих работу устройства в соответствии с заданным законом функционирования. На этом этапе требуется представить ФАЛ устройства через функции выбранной полной системы (*базиса*).

Базисом называют *полную систему* функций алгебры логики.

Система функций является *полной*, если она включает в себя по крайней мере одну функцию, не сохраняющую 0, одну функцию, не сохраняющую 1, одну несамоудовлетворяющую, одну немонотонную и одну нелинейную функции (*теорема Поста - Яблонского*).

Свойством сохранения нуля функция $f(X_1, X_2, \dots, X_n)$ обладает, если она на нулевом наборе аргументов равна нулю, т. е. $f(0, 0, \dots, 0) = 0$.

Свойством сохранения единицы функция $f(X_1, X_2, \dots, X_n)$ обладает, если она на единичном наборе аргументов равна единице, т. е. $f(1, 1, \dots, 1) = 1$.

Свойством самодвойственности обладает функция, у которой инвертирование всех аргументов приводит к инверсии значения функции, т. е. $f(\bar{X}_1, \bar{X}_2, \dots, \bar{X}_n) = \bar{f}(X_1, X_2, \dots, X_n)$.

Свойством монотонности обладает функция, значение которой при любом возрастании набора не убывает, т. е.

$$f(a_1, a_2, \dots, a_n) \leq f(a'_1, a'_2, \dots, a'_n),$$

где $a_1 \leq a'_1$, $a_2 \leq a'_2, \dots$, $a_n \leq a'_n$.

Свойством линейности обладает функция, которая может быть представлена полиномом первой степени:

$$f(X_1, X_2, \dots, X_n) = a_0 \oplus a_1 X_1 \oplus a_2 X_2 \oplus \dots \oplus a_n X_n,$$

где a_0, a_1, \dots, a_n – коэффициенты, равные нулю или единице.

Минимальный базис состоит из такого набора функций, исключение из которого любой функции превращает этот набор в неполную систему функций. Наиболее удобным для представления в виде логического выражения функций алгебры логики является базис, содержащий конъюнкцию (умножение), дизъюнкцию (сложение) и инверсию (отрицание) (базис И-ИЛИ-НЕ). Этот базис называется основным. Минимальный базис включает в себя две функции И-НЕ (базис Шеффера) либо ИЛИ-НЕ (базис Пирса). Однако использование трех функций упрощает логическое описание, а в ряде случаев и построение дискретных устройств автоматики, телемеханики и связи.

Любая функция алгебры логики выражается через исходные функции неоднозначно. Поэтому требуется найти такую форму ее представления, которая позволяет построить наиболее простую электрическую схему. При решении этой задачи функцию алгебры логики дискретного устройства вначале оказывается удобным представить в некоторой исходной канонической форме, которую называют нормальной, а затем преобразовать ее так, чтобы она соответствовала наиболее простой электрической схеме с учетом выбранного базиса логических элементов.

Каноническими формами представления функций алгебры логики являются дизъюнктивная нормальная форма и конъюнктивная нормальная форма [1, с. 45 - 46].

Совершенная дизъюнктивная нормальная форма (СДНФ) представляет собой алгебраическое выражение, которое принимает

значение, равное 1 на тех наборах переменных, на которых значение заданной функции равно 1.

Совершенная конъюнктивная нормальная форма (СКНФ) представляет собой алгебраическое выражение, которое принимает значение 0 на тех наборах переменных, на которых значение заданной функции равно 0.

Кроме совершенных форм ФАЛ существуют и простые: дизъюнктивная нормальная форма (ДНФ) и конъюнктивная нормальная форма (КНФ), которые не содержат в своих *конституентах* всех аргументов. Конституентой единицы (нуля) называют функцию, которая принимает значение единицы (нуля) при одном из всех возможных наборов переменных и нуля (единицы) при всех других.

Упрощение ФАЛ может быть осуществлено различными способами: путем использования *законов и аксиом алгебры логики*, применением *карт Карно*, использованием *правил упрощения контактных схем*, использованием *методов Квайна, Квайна – Мак-Класки* и др. [1, с.49 - 63].

1.2 Элементы управления логическим конвертером

Логический конвертер (рисунок 2.1) представляет собой мощное средство программы «Электронная лаборатория», позволяющее по заданной схеме дискретного устройства строить его таблицу истинности, по таблице истинности дискретного устройства получать его ФАЛ, минимизировать полученную ФАЛ, по полученной ФАЛ строить схемы в базисе И-ИЛИ-НЕ и базисе И-НЕ.

Недостатками логического конвертера (ЛК) являются: возможность анализа выходных значений всего одной функции при числе входных аргументов до восьми, а также отсутствие опции для построения схем в базисе ИЛИ-НЕ.

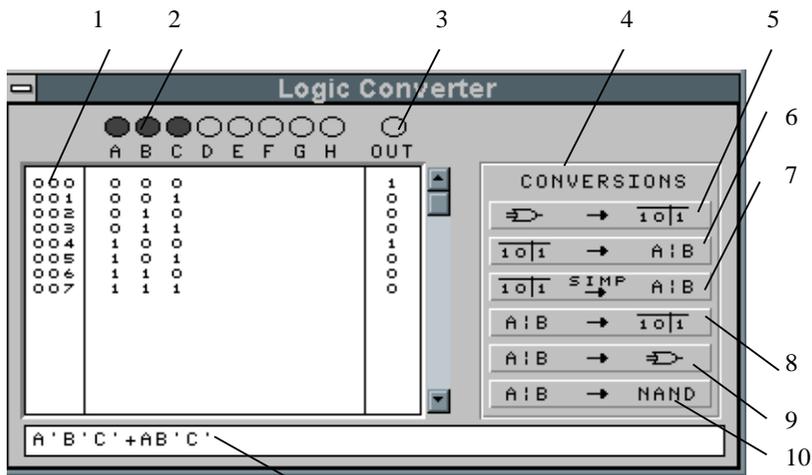
Для получения таблицы истинности для заданной схемы с помощью ЛК достаточно соединить соответствующие входы ЛК (поз. 2 на рисунке 2.1) со входами анализируемой дискретной схемы (при этом окно ЛК должно быть свернуто в пиктограмму), а выход этой схемы соединить со входом анализа выходных значений *OUT* (поз. 3 на рисунке 2.1) и нажать кнопку, соответствующую позиции 5 на рисунке 2.1. В результате в окне конвертера будет высвечена полученная ТИ.

Для заданий ФАЛ в окне 11 ЛК необходимо знать следующее.

Самым старшим является разряд А конвертера, а самым младшим - разряд Н.

При задании формул логическое сложение задается символами «|» или «+», отрицание – символом «'». При умножении двух аргументов они пишутся друг за другом без каких-либо символов. Для того чтобы выполнить инверсию суммы двух аргументов, их необходимо

предварительно взять в скобки. Например, выражение $\overline{A \vee B \cdot C}$ в окне 11 задания формул ЛК должно выглядеть следующим образом: «(A|BC)» или



11

1 – номер входного набора; 2 – активные аргументы; 3 – выходное значение функции; 4 – панель конвертирования; 5 – кнопка для построения таблицы истинности по заданной схеме; 6 – кнопка для получения логического выражения в форме СДНФ по таблице истинности; 7 – кнопка для упрощения ТИ; 8 – кнопка для построения ТИ по заданной формуле; 9 – кнопка для построения схемы в базе И-ИЛИ-НЕ по формуле, заданной в окне 11; 10 – кнопка для построения схемы в базе И-НЕ по формуле, заданной в окне 11

Рисунок 2.1 – Окно логического конвертера

«(A+BC)».

2 Порядок выполнения работы

1 Изучить по учебнику, данным методическим указаниям и конспекту лекций способы задания функций алгебры логики, способы минимизации ФАЛ, построение ТИ, получение СДНФ, СКНФ, ДНФ, КНФ функций алгебры логики по ТИ, преобразование базисов.

2 Используя данное пособие, ознакомиться с элементами управления логического конвертера.

3 Построить при помощи программы «Электронная лаборатория» дискретную схему без ее упрощения по формуле, взятой из таблицы 2.1, в соответствии с заданным вариантом.

4 Упростить вручную заданную ФАЛ.

5 Получить ТИ построенной схемы, используя ЛК.

6 По ТИ, используя соответствующее свойство ЛК, найти СДНФ.

7 Вручную найти СКНФ для имеющейся таблицы истинности.

8 На основе СДНФ построить схему ДУ в базисе И-ИЛИ-НЕ и базисе И-НЕ.

9 Используя ЛК, упростить СДНФ и построить по упрощенной ФАЛ схему ДУ в базисах И-НЕ и ИЛИ-НЕ.

10 Сделать выводы по работе.

Т а б л и ц а 2.1 – Варианты заданий

Вариант	Функция алгебры логики	Вариант	Функция алгебры логики
1	$(\overline{AB} \vee \overline{CA})(\overline{AB} \vee C)$	16	$C\overline{A} \vee \overline{B} \vee (\overline{CA} \vee ABC)\overline{A}$
2	$\overline{\overline{AB}} \vee \overline{AC} \vee AB(\overline{AC} \vee \overline{AC})$	17	$ABC \vee \overline{A}(B \vee C) \vee \overline{ABC}$
3	$\overline{ABC} \vee \overline{AB}(C \vee \overline{A})$	18	$\overline{\overline{AB}} \vee \overline{AB} \vee (\overline{A} \vee \overline{CB})$
4	$(\overline{A} \vee \overline{C})\overline{ABC} \vee \overline{AB}$	19	$A \vee (\overline{A} \vee \overline{B} \vee \overline{C})(\overline{A} \vee \overline{BC} \vee \overline{CB})$
5	$(\overline{AB} \vee \overline{C})(\overline{ABC} \vee \overline{\overline{ABC}})$	20	$A \vee (\overline{A} \vee \overline{B} \vee \overline{C})(\overline{A} \vee \overline{B} \vee C)$
6	$ABC \vee \overline{ABC}(\overline{AB} \vee \overline{ABC})$	21	$ABC \vee \overline{AB}(\overline{A} \vee \overline{B} \vee \overline{C})(\overline{A} \vee \overline{C})$
7	$A(A \vee \overline{ABC}) \vee \overline{\overline{ABC}}$	22	$AB \vee \overline{AB} \vee \overline{AC} \vee BC(\overline{A} \vee \overline{CB})$
8	$B(A \vee \overline{BC}) \vee ABC(\overline{A} \vee BC)$	23	$\overline{A} \vee \overline{AB} \vee \overline{ABC} \vee A(\overline{A} \vee \overline{C})$
9	$(\overline{A} \vee B)(\overline{A} \vee C) \vee ABC \vee \overline{AC}$	24	$A \vee \overline{\overline{BC}} \vee (A \vee \overline{B})(\overline{\overline{A} \vee \overline{B}})$
10	$\overline{\overline{ABC}} \vee \overline{ABC} \vee \overline{AC}$	25	$\overline{A} \vee \overline{AB} \vee \overline{BC}(\overline{A} \vee \overline{BC})$
11	$\overline{ABC} \vee \overline{\overline{ABC}} \vee (AB \vee \overline{CA})$	26	$\overline{AA} \vee \overline{B}(\overline{B} \vee \overline{CB})(\overline{A} \vee \overline{BC})$
12	$(\overline{AC} \vee \overline{AB})(\overline{AB} \vee \overline{C})$	27	$A(\overline{A} \vee \overline{ABC})(\overline{A} \vee \overline{\overline{BC}})$
13	$A \vee (\overline{A} \vee B)(\overline{A} \vee \overline{CB}) \vee \overline{BA}$	28	$ABC \vee \overline{A}(BC \vee \overline{BC}) \vee \overline{C}$
14	$A \vee \overline{AB} \vee \overline{BC} \vee ABC(\overline{A} \vee BC)$	29	$(A \vee B)(\overline{A} \vee B)(\overline{AB} \vee \overline{C}) \vee \overline{A}$
15	$A \vee AB \vee \overline{AC}(\overline{\overline{AB}} \vee \overline{\overline{B}})$	30	$\overline{ABC} \vee \overline{\overline{ABC}} \vee (\overline{AB} \vee \overline{C})$

Содержание отчета

Наименование и цель лабораторной работы, краткие сведения из теории,

полученная ТИ, полученные схемы (неупрощенные и упрощенные), полученные СДНФ и СКНФ, выражение, полученное при минимизации заданной ФАЛ вручную с подробным выводом, выражение минимизированной ФАЛ при использовании ЛК, ответы на заданные преподавателем контрольные вопросы, выводы по работе.

Контрольные вопросы

- 1 Что называется функцией алгебры логики?
- 2 Приведите основные законы алгебры логики.
- 3 Приведите основные аксиомы алгебры логики.
- 4 Что называется СДНФ?
- 5 Что называется СКНФ?
- 6 В чем заключается отличие между СДНФ и СКНФ?
- 7 Чем отличаются совершенные формы ФАЛ от несовершенных?
- 8 Что называется минимальным базисом?
- 9 Сформулируйте теорему Поста-Яблонского.
- 10 Сформулируйте свойство сохранения нуля ФАЛ.
- 11 Сформулируйте свойство сохранения единицы ФАЛ.
- 12 Сформулируйте свойство самодвойственности ФАЛ.
- 13 Сформулируйте свойство линейности ФАЛ.
- 14 Сформулируйте свойство монотонности ФАЛ.
- 15 Сформулируйте закон де Моргана.
- 16 Какие способы задания ФАЛ Вам известны?

Лабораторная работа № 3

ИЗУЧЕНИЕ ШИФРАТОРОВ, ДЕШИФРАТОРОВ И ПРЕОБРАЗОВАТЕЛЕЙ КОДОВ

Ц е л ь р а б о т ы. Изучить назначение, принципы построения и структуры шифраторов, дешифраторов и преобразователей кодов.

1 Краткие сведения из теории

Шифратором (кодером) называется устройство, преобразующее сигнал логической 1 на одном из входов в соответствующую кодовую комбинацию на выходах. Шифраторы используют, например, для ввода информации в цифровые системы. В таких устройствах при нажатии выбранной клавиши подается сигнал на определенный вход шифратора, и на его выходе появляется двоичное число, соответствующее данной клавише.

Для синтеза шифратора строится таблица истинности (ТИ), на основании

которой получают аналитические зависимости выходов от входов. Затем выражения преобразовывают (если надо) к требуемому базису и строят схему.

П р и м е р. Необходимо построить шифратор (Ш) для перевода десятичных чисел от 3 до 9 в двоичный код 8421 (таблица 3.1) в базе ИЛИ-НЕ.

Т а б л и ц а 3.1 – Примеры различных двоичных кодов

Десятичные цифры	Двоичные коды					
	8421	2421	2 из 5	с избытком 3	3а+2	7421
0	0000	0000	11000	0011	00010	0000
1	0001	0001	01100	0100	00101	0001
2	0010	0010	00110	0101	01000	0010
3	0011	0011	00011	0110	01011	0011
4	0100	0100	10001	0111	01110	0100
5	0101	1011	10100	1000	10001	0101
6	0110	1100	01010	1001	10100	0110
7	0111	1101	00101	1010	10111	1000
8	1000	1110	10010	1011	11010	1001
9	1001	1111	01001	1100	11101	1010

Строим ТИ (таблица 3.2).

Т а б л и ц а 3.2 – Таблица истинности шифратора

Входы	Выходы (код 8421)			
	X4	X3	X2	X1
Y3	0	0	1	1
Y4	0	1	0	0
Y5	0	1	0	1
Y6	0	1	1	0
Y7	0	1	1	1
Y8	1	0	0	0
Y9	1	0	0	1

Записываем выражения ДНФ каждого из выходов путем сложения тех аргументов, на которых выход будет находиться в состоянии 1:

$$X4 = Y8 \vee Y9; \quad X3 = Y4 \vee Y5 \vee Y6 \vee Y7; \quad X2 = Y3 \vee Y6 \vee Y7; \quad X1 = Y3 \vee Y5 \vee Y7 \vee Y9.$$

Затем преобразуем выражения к базису ИЛИ-НЕ по закону двойного

$$\text{отрицания: } X4 = Y8 \vee Y9 = \overline{\overline{Y8} \downarrow \overline{Y9}}; \quad X3 = Y4 \vee Y5 \vee Y6 \vee Y7 = \overline{\overline{Y4} \downarrow \overline{Y5} \downarrow \overline{Y6} \downarrow \overline{Y7}};$$

$$X2 = Y3 \vee Y6 \vee Y7 = \overline{\overline{Y3} \downarrow \overline{Y6} \downarrow \overline{Y7}}; \quad X1 = Y3 \vee Y5 \vee Y7 \vee Y9 = \overline{\overline{Y3} \downarrow \overline{Y5} \downarrow \overline{Y7} \downarrow \overline{Y9}} \quad \text{и}$$

строим схему (рисунок 3.1).

Дешифратором (декодером) называется устройство, распознающее различные кодовые комбинации, т. е. преобразующее двоичное число в сигнал логической единицы на одном из выходов, соответствующем десятичной цифре поданной на вход двоичной комбинации. Число входов дешифратора определяется числом символов кодовой комбинации n , а число выходов

$$m = 2^n.$$

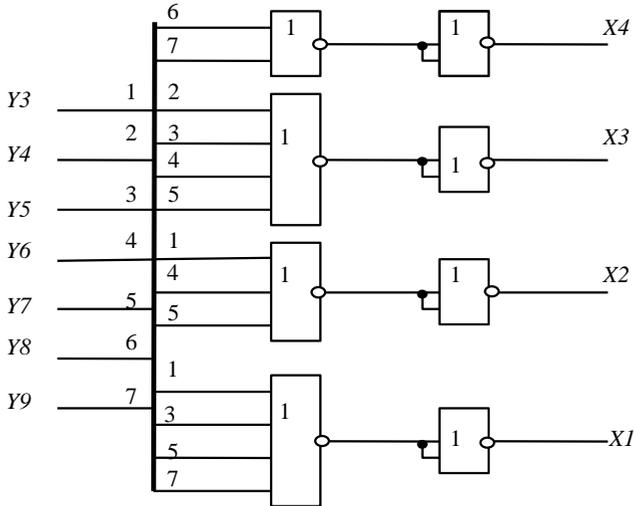


Рисунок 3.1 – Схема шифратора в базе ИЛИ-НЕ

Дешифраторы по принципу построения делятся на *одноступенчатые (линейные)* и *многоступенчатые*. Одноступенчатые дешифраторы выполняют прямое преобразование входных сигналов, поданных в двоичном (параллельном) коде, в выходной сигнал на одном из выходов.

Синтез линейного дешифратора можно выполнить по той же ТИ, что и шифратора, только поменять в ней места входы и выходы.

Пример. Построить линейный дешифратор для перевода двоичных чисел от 3 до 8 кода 8421 в десятичный код в базе И-НЕ.

Строим ТИ (таблица 3.3).

Таблица 3.3 – ТИ дешифратора

Входы				Выходы
X4	X3	X2	X1	Y
0	0	1	1	Y3
0	1	0	0	Y4
0	1	0	1	Y5
0	1	1	0	Y6

0	1	1	1	Y7
1	0	0	0	Y8

Получаем функции каждого выхода путем записи КНФ для каждого из выходов: $Y3 = \overline{X4X3X2X1}$; $Y4 = \overline{X4X3X2X1}$; $Y5 = \overline{X4X3X2X1}$; $Y6 = \overline{X4X3X2X1}$; $Y7 = \overline{X4X3X2X1}$; $Y8 = \overline{X4X3X2X1}$.

Преобразуем выражения к базису И-НЕ: $Y3 = \overline{\overline{\overline{X4X3X2X1}}} = \overline{\overline{X4}|\overline{X3}|X2|X1}$;
 $Y4 = \overline{\overline{\overline{X4X3X2X1}}} = \overline{\overline{X4}|\overline{X3}|\overline{X2}|X1}$; $Y5 = \overline{\overline{\overline{X4X3X2X1}}} = \overline{\overline{X4}|\overline{X3}|\overline{X2}|X1}$;
 $Y6 = \overline{\overline{\overline{X4X3X2X1}}} = \overline{\overline{X4}|\overline{X3}|X2|X1}$; $Y7 = \overline{\overline{\overline{X4X3X2X1}}} = \overline{\overline{X4}|\overline{X3}|X2|X1}$;
 $Y8 = \overline{\overline{\overline{X4X3X2X1}}} = \overline{\overline{X4}|\overline{X3}|\overline{X2}|X1}$ и строим схему (рисунок 3.2).

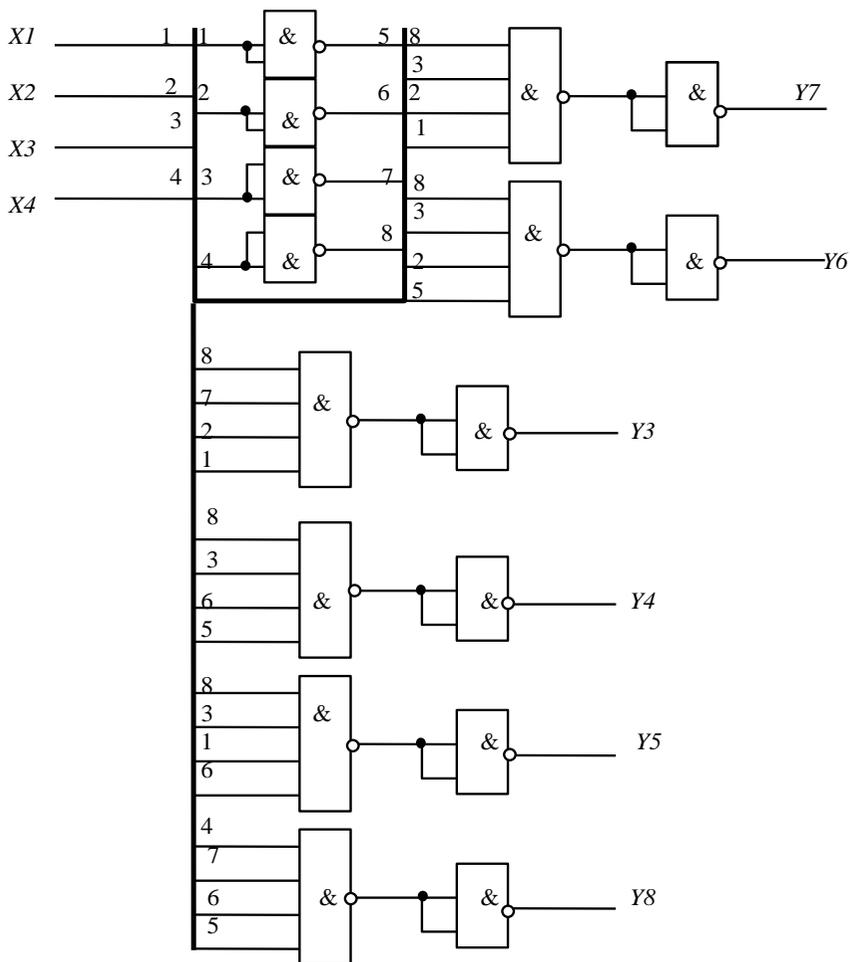


Рисунок 3.2 – Схема дешифратора в базисе И-НЕ

С увеличением символов входной кодовой комбинации возрастает число возможных входных наборов (исходя из соотношения $m = 2^n$) и, соответственно, число входов. Например, если в кодовой комбинации будет два разряда, то количество входных наборов равно четырем, и дешифратор должен иметь два входа, а для его синтеза необходимо использовать двухвходовые логические элементы. Если же кодовая комбинация содержит три разряда, то количество комбинаций возрастает до восьми, и дешифратор будет иметь три входа. При этом в схеме потребуется использовать уже трехвходовые логические элементы.

В тех случаях, когда количество входов логических элементов, используемых для построения дешифраторов, меньше числа символов в кодовой комбинации, применяют многоступенчатые дешифраторы – *прямоугольные (матричные)* и *пирамидальные*, которые имеют ступенчатую структуру (рисунок 3.3).

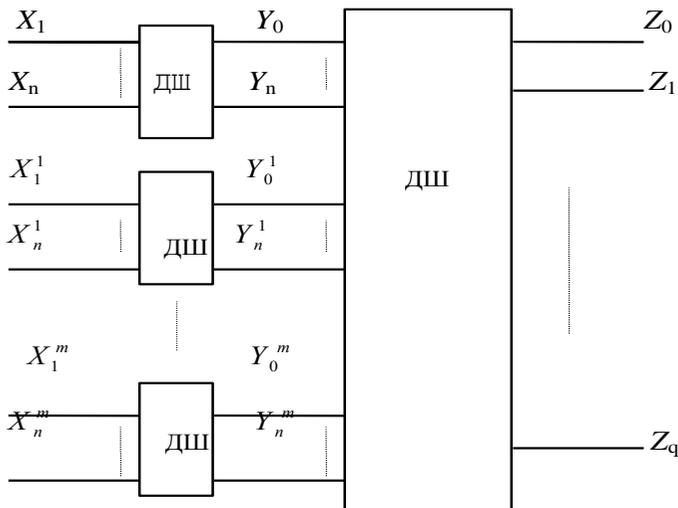


Рисунок 3.3 – Структура многоступенчатого дешифратора

Прямоугольные (матричные) дешифраторы (рисунок 3.4) содержат первую ступень из нескольких линейных дешифраторов и вторую ступень, представляющую собой схемы совпадения (элементы И).

Количество линейных дешифраторов определяется числом групп, на которые разбивается кодовая комбинация. В приведенном на рисунке 3.4 прямоугольном дешифраторе входная комбинация разбита на две группы: X_1, X_2 и X_3, X_4 . Поэтому первая ступень содержит два линейных дешифратора – ДШ1 и ДШ2, которые имеют по четыре входа каждый. Вторая ступень – матричный дешифратор МД, собранный на двухвходовых элементах И, на первые входы которых подаются сигналы с линейного дешифратора ДШ1, а на вторые – с дешифратора ДШ2. Недостатком подобных дешифраторов является большое число элементов И. Поэтому часто вторая ступень МД строится не на логических элементах И, а на полупроводниковых диодах.

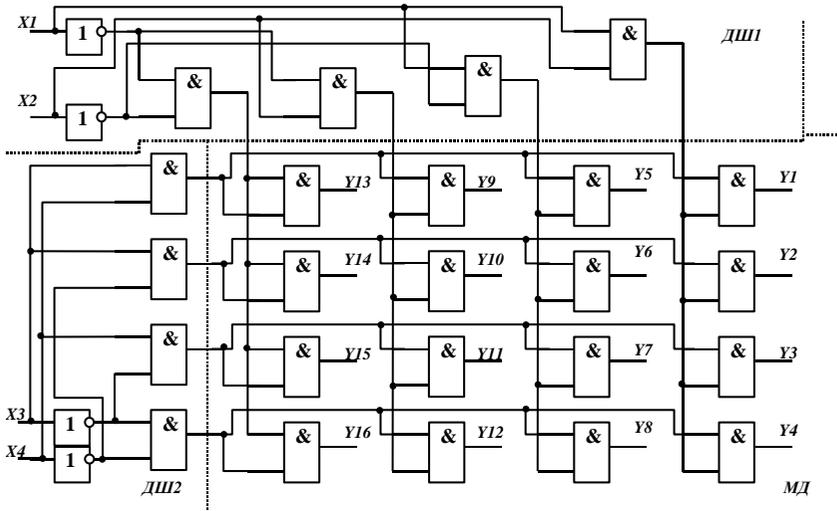


Рисунок 3.4 – Структура четырехразрядного матричного дешифратора на двухвходовых логических элементах

Пирамидальные дешифраторы (рисунок 3.5) имеют число ступеней на единицу меньше числа символов кодовой комбинации, а число элементов И в i -й ступени равно 2^{i+1} . В таких дешифраторах, как правило, используются двухвходовые элементы И.

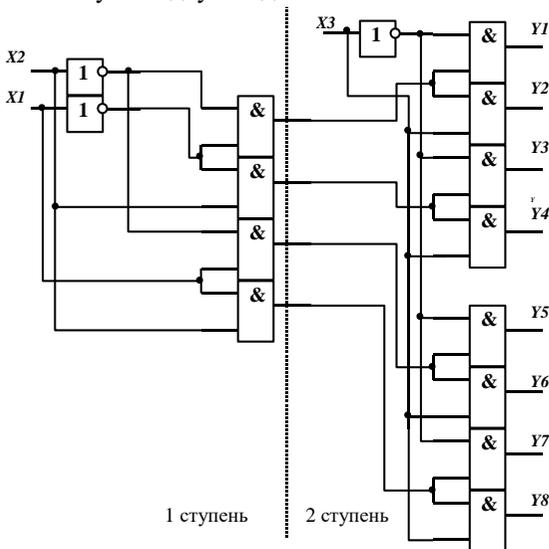


Рисунок 3.5 –
Пирамидальный дешифратор

На рисунке 3.5 изображена схема пирамидального дешифратора

трехсимвольного кода. Число ступеней будет две, так как число символов три (X_1, X_2, X_3). Количество элементов И в первой ступени – $2^{1+1} = 4$, а во второй – $2^{2+1} = 8$. Недостатком пирамидальных дешифраторов является снижение быстродействия при большом количестве ступеней. Однако по количеству двухвходовых элементов И они экономичнее, чем прямоугольные дешифраторы.

Преобразователем кодов (кодопреобразователем) называется устройство для преобразования одного вида двоичного кода в другой. Преобразование одного кода в другой можно осуществлять последовательным соединением дешифратора и шифратора. В этом случае m -элементная комбинация преобразуется дешифратором в сигнал логической 1 на соответствующем выходе, а шифратор из этого сигнала формирует требуемую n -элементную комбинацию. Существует также метод построения преобразователя кодов (ПК), основанный на использовании свойства независимости входов и выходов дискретного устройства.

Синтез ПК с использованием свойства независимости осуществляется по следующей методике:

1) определяется количество двоичных чисел, которые необходимо перевести из одного кода в другой;

2) составляется ТИ, в которой описываются входные и выходные коды двоичного числа;

3) далее синтез выполняется по методике построения ДУ с несколькими выходами и не полностью определяемыми функциями (так как каждый выход Y_m не зависит от другого, то для каждого выхода составляется своя логическая функция и затем строится схема на общих входах). Обязательным этапом в данном случае является минимизация функций выходов одним из известных методов.

П р и м е р. Построить ПК в базе И-НЕ для перевода чисел от 5 до 8 кода 8421 в код 7421 (коды см. таблицу 3.2).

Необходимо перевести четыре двоичных числа. Составляем ТИ (таблица 3.4).

Т а б л и ц а 3.4 – ТИ преобразователя кодов

Десятичные цифры	Входы				Выходы			
	X_4	X_3	X_2	X_1	Y_4	Y_3	Y_2	Y_1
	8	4	2	1	7	4	2	1
5	0	1	0	1	0	1	0	1
6	0	1	1	0	0	1	1	0
7	0	1	1	1	1	0	0	0
8	1	0	0	0	1	0	0	1

Пользуясь свойством независимости выходов, запишем функции каждого выхода в тех строках, где выход равен единице в виде СДНФ:

$$Y_4 = \overline{X_4 X_3 X_2 X_1} \vee X_4 \overline{X_3 X_2 X_1}; \quad Y_3 = \overline{X_4 X_3 X_2 X_1} \vee \overline{X_4 X_3 X_2 X_1}; \quad Y_2 = \overline{X_4 X_3 X_2 X_1};$$

$$Y_1 = \overline{X_4 X_3 X_2 X_1} \vee X_4 \overline{X_3 X_2 X_1}.$$

Затем упрощаем (в данном примере упрощение не производится) полученные выражения одним из подходящих (заданных) методов минимизации и преобразуем их к базису И-НЕ. Для этого используем аксиомы и законы алгебры логики.

$$Y_4 = \overline{\overline{X_4 X_3 X_2 X_1} \cdot X_4 \overline{X_3 X_2 X_1}} = (\overline{X_4} | X_3 | X_2 | X_1) | (X_4 | \overline{X_3} | \overline{X_2} | \overline{X_1}),$$

$$Y_3 = \overline{\overline{X_4 X_3 X_2 X_1} \cdot \overline{X_4 X_3 X_2 X_1}} = (\overline{X_4} | X_3 | \overline{X_2} | X_1) | (\overline{X_4} | X_3 | X_2 | \overline{X_1}),$$

$$Y_2 = \overline{\overline{X_4 X_3 X_2 X_1}} = \overline{X_4} | X_3 | X_2 | \overline{X_1};$$

$$Y_1 = \overline{\overline{X_4 X_3 X_2 X_1} \cdot X_4 \overline{X_3 X_2 X_1}} = (\overline{X_4} | X_3 | \overline{X_2} | X_1) | (X_4 | \overline{X_3} | \overline{X_2} | \overline{X_1}).$$

По полученным выражениям строим схему ПК, представленную на рисунке 3.6.

2 Порядок выполнения работы

1 Изучить по учебнику, данным методическим указаниям и конспекту лекций назначение, принцип действия, структуру и способы синтеза шифраторов, дешифраторов и преобразователей кодов.

2 При помощи программы «Электронная лаборатория» выполнить синтез заданного устройства: шифратора, дешифратора или преобразователя кодов (при синтезе преобразователя кодов обязательным этапом является минимизация выходных функций перед построением схемы), в соответствии с заданием, приведенным в таблице 3.5.

3 Выполнить проверку правильности работы построенного устройства в соответствии с ТИ, используя средства программы «Электронная лаборатория».

4 Сделать выводы по работе.

Содержание отчета

Наименование и цель лабораторной работы, краткие сведения из теории, ТИ заданного устройства, выражения, связывающие между собой входы и выходы устройства (с учетом преобразования базисов, если это требуется по варианту задания), схема полученного устройства в требуемом базисе, ответы на заданные преподавателем контрольные вопросы.

Контрольные вопросы

1 Что называется шифратором?

- 2 Что называется дешифратором?
- 3 Что называется преобразователем кодов?
- 4 В чем заключаются особенности матричных дешифраторов?
- 5 В чем заключаются особенности пирамидальных дешифраторов?
- 6 Где применяются шифраторы, дешифраторы и преобразователи кодов?
- 7 В чем заключается отличие двух способов синтеза преобразователей кодов?
- 8 Сформулируйте принцип независимости входов и выходов дискретного устройства.
- 9 Для чего осуществляется минимизация функций алгебры логики, связывающих между собой входы и выходы дискретных устройств?
- 10 Приведите примеры интегральных микросхем шифраторов и дешифраторов, выпускаемых промышленностью.

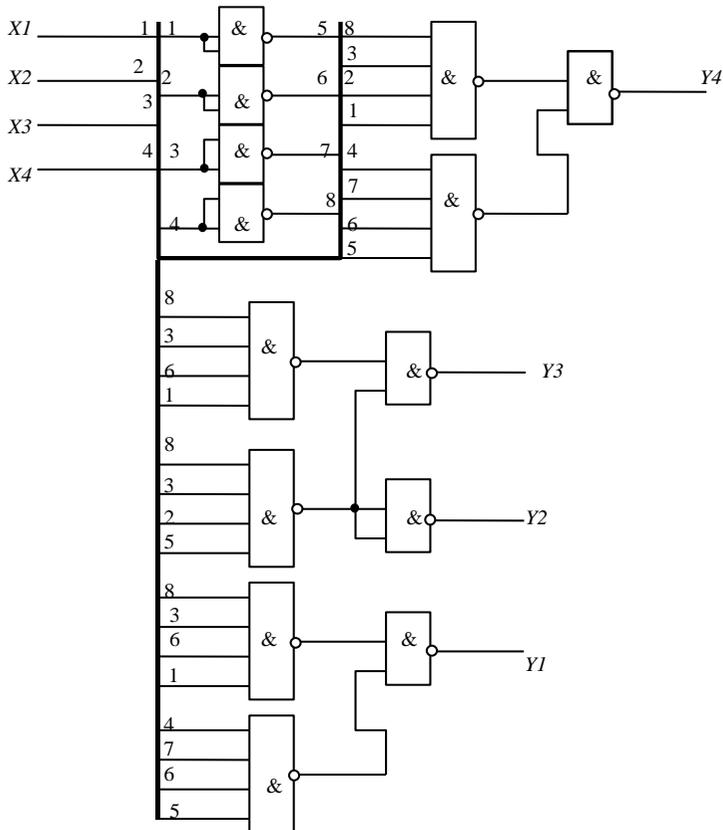


Рисунок 3.6 – Схема ПК в базисе И-НЕ

Таблица 3.5 – Варианты заданий

Вариант	Тип устройства для цифр в десятичном эквиваленте	Базис реализации	Код	
			на входе	на выходе
1	ПК от 0 до 5	И-ИЛИ-НЕ	8421	2 из 5
2	Ш от 3 до 7	И-НЕ	Десятичный	с избытком 3
3	ДШ от 3 до 9	ИЛИ-НЕ	8421	Десятичный
4	ПК от 2 до 9	ИЛИ-НЕ	7421	3а+2
5	Ш от 0 до 6	И-НЕ	Десятичный	7421
6	ДШ от 2 до 8	И-ИЛИ-НЕ	7421	Десятичный
7	ПК от 1 до 7	И-НЕ	2421	с избытком 3
8	Ш от 0 до 5	И-НЕ	Десятичный	2421
9	ДШ от 0 до 5	ИЛИ-НЕ	2421	Десятичный
10	ПК от 2 до 7	И-ИЛИ-НЕ	2 из 5	8421
11	Ш от 2 до 9	И-ИЛИ-НЕ	Десятичный	2 из 5
12	ДШ от 2 до 9	ИЛИ-НЕ	2 из 5	Десятичный
13	ПК от 4 до 9	ИЛИ-НЕ	3а+2	7421
14	Ш от 3 до 9	И-НЕ	Десятичный	3а+2
15	ДШ от 1 до 8	И-НЕ	3а+2	Десятичный
16	ПК от 1 до 8	И-ИЛИ-НЕ	2 из 5	3а+2
17	Ш от 1 до 8	И-НЕ	Десятичный	2421
18	ДШ от 1 до 6	И-НЕ	2421	Десятичный
19	ПК от 1 до 6	ИЛИ-НЕ	3а+2	8421
20	Ш от 1 до 6	ИЛИ-НЕ	Десятичный	3а+2
21	ДШ от 0 до 6	И-НЕ	3а+2	Десятичный
22	ПК от 0 до 6	И-ИЛИ-НЕ	2421	с избытком 3
23	Ш от 4 до 9	ИЛИ-НЕ	Десятичный	с избытком 3
24	ДШ от 0 до 4	И-ИЛИ-НЕ	с избытком 3	Десятичный
25	ПК от 0 до 4	И-ИЛИ-НЕ	2 из 5	3а+2
26	Ш от 0 до 4	И-НЕ	Десятичный	2 из 5
27	ДШ от 0 до 7	ИЛИ-НЕ	2 из 5	Десятичный
28	ПК от 0 до 7	ИЛИ-НЕ	с избытком 3	7421
29	Ш от 0 до 7	И-НЕ	Десятичный	2421
30	ДШ от 0 до 4	И-НЕ	8421	Десятичный

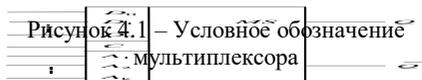
Лабораторная работа № 4

ИЗУЧЕНИЕ МУЛЬТИПЛЕКСОРОВ И ДЕМУЛЬТИПЛЕКСОРОВ

Ц е л ь р а б о т ы. Изучить назначение, структуру и принципы построения мультиплексоров и демультимплексоров.

1 Краткие сведения из теории

Мультиплексор (рисунок 4.1) – это устройство, предназначенное для коммутации одного из информационных входов D_0, D_1, \dots, D_n , адрес которого задается на управляющих входах A_1, A_2, \dots, A_k (причем $n = 2^k - 1$) на выход Q , при подаче сигнала синхронизации на вход C (если последний имеется) [1, с.78 - 79].



Каждый из входов данных имеет свой номер, который задается на информационных входах.

Набор переменных, поступающих на управляющие входы, задает двоичное число N_i вида $x_1x_2x_3\dots x_k$. При подаче сигнала на вход C выходная переменная мультиплексора Q повторяет переменную информационного входа D_{N_i} с номером N_i , задаваемым двоичным кодом на управляющих входах. При отсутствии синхронизирующего сигнала ($C = 0$) связь между информационными входами и выходом отсутствует ($Q = 0$).

При необходимости коммутации n направлений в одно и наличии мультиплексоров с n входами каждому направлению присваивают свой номер от 0 до n . Кодовые комбинации номеров поступают на управляющие входы мультиплексора, а сигналы коммутируемых линий – на информационные входы, причем линия с номером N_i подается на вход N_i . Если число входов мультиплексора больше числа коммутируемых линий, часть входов со старшими номерами не будет использоваться. На неиспользуемые входы подается константа 0.

П р и м е р. Функционирование синхронного мультиплексора при трех

адресных входах определяется таблицей истинности (ТИ) (таблица 4.1). Необходимо составить логическое выражение для выхода Q и построить схему мультиплексора в базе И-ИЛИ-НЕ.

Т а б л и ц а 4.1 – ТИ мультиплексора

Входы				Выход
A1	A2	A3	C	Q
0	0	0	1	D0
0	0	1	1	D1
0	1	0	1	D2
0	1	1	1	D3
1	0	0	1	D4
1	0	1	1	D5
1	1	0	1	D6
1	1	1	1	D7
~	~	~	0	0

Функция выхода в виде совершенной дизъюнктивной нормальной формы (СДНФ) получается как сложение конъюнкций каждой из строк ТИ, и будет иметь вид: $Q = C(D0\overline{A1}\overline{A2}\overline{A3} \vee D1\overline{A1}\overline{A2}A3 \vee D2\overline{A1}A2\overline{A3} \vee D3\overline{A1}A2A3 \vee D4A1\overline{A2}\overline{A3} \vee D5A1\overline{A2}A3 \vee D6A1A2\overline{A3} \vee D7A1A2A3)$.

По полученной функции строим схему мультиплексора (рисунок 4.2).

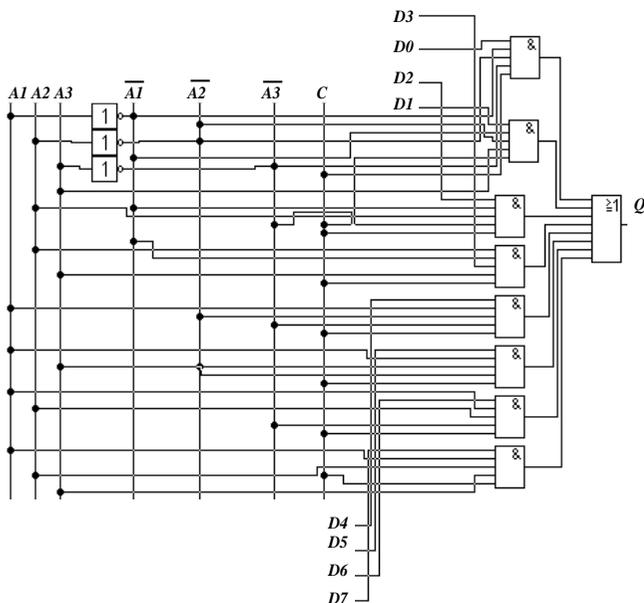


Рисунок 4.2 - Схема мультиплексора в базе И-ИЛИ-НЕ

Если требуется построить устройство с числом входов, большим, чем имеется в одном мультиплексоре, осуществляют каскадное включение мультиплексоров. На рисунке 4.3 приведена схема каскадной коммутации с 14 направлений в одно на мультиплексорах с восьмью и четырьмя входами. В первом каскаде коммутируются линии с номерами от 0 до 7 и от 8 до 13 двумя восьмивходовыми мультиплексорами, в одном из которых используется только шесть входов. Затем группа из двух выходных линий первого каскада коммутируется микросхемой мультиплексора с четырьмя входами.

Демультимплексор (рисунок 4.4) – это устройство, предназначенное для коммутации входа данных D на один из выходов Q_1, Q_2, \dots, Q_n , адрес которого выбирается при помощи управляющих входов A_1, A_2, \dots, A_k (причем $n = 2^k - 1$), при подаче сигнала синхронизации на вход C (если последний имеется). Демультимплексор имеет принцип действия, обратный мультиплексору [1, с.83 - 84].

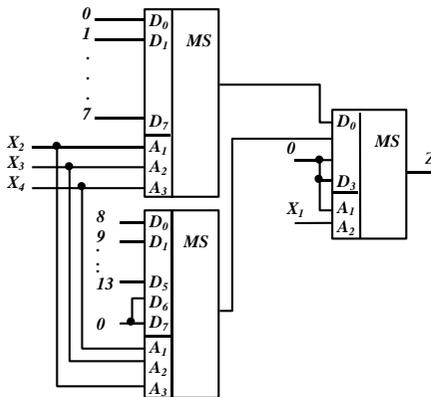


Рисунок 4.3 - Схема каскадного включения мультиплексоров

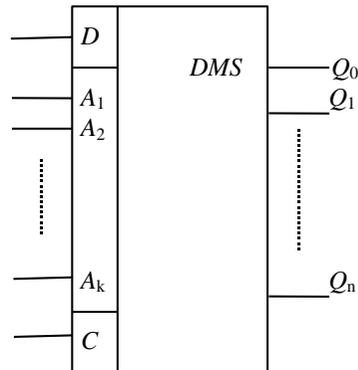


Рисунок 4.4 – Условное обозначение демультимплексора

Пр и м е р. Необходимо синтезировать синхронный демультимплексор на 4 выхода данных, функционирование которого задано таблицей 4.2.

Т а б л и ц а 4.2 – ТИ демультимплексора

Входы			Выходы			
A_1	A_2	C	Q_0	Q_1	Q_2	Q_3
0	0	1	D	0	0	0
0	1	1	0	D	0	0
1	0	1	0	0	D	0
1	1	1	0	0	0	D
~	~	0	0	0	0	0

Функции выходов составляются для каждого из выходов в отдельности на основе принципа независимости входов и выходов. Они имеют следующий вид: $Q0 = D\overline{A1}\overline{A2}C$; $Q1 = D\overline{A1}A2C$; $Q2 = DA1\overline{A2}C$; $Q3 = DA1A2C$.

Согласно формулам строим схему, которая представлена на рисунке 4.5.

С использованием мультиплексоров и демультиплексоров можно строить схемы коммутации с n направлений в m . В схеме, приведенной на рисунке 4.6, по заданным адресам любой из входов может быть подключен к любому из выходов.

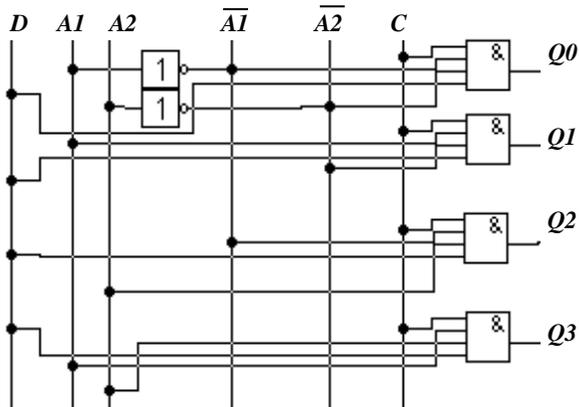


Рисунок 4.5 - Схема демультиплексора в базисе И-ИЛИ-НЕ

Число выходов увеличивают каскадным включением демультиплексоров. К каждому выходу демультиплексора первого каскада может быть подключен демультиплексор второго каскада.

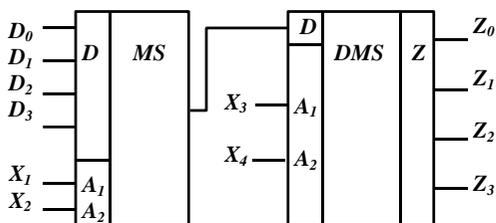


Рисунок 4.6 – Схема коммутации с несколькими входами и выходами

Демультиплексор может использоваться как дешифратор. В этом случае на вход демультиплексора подается константа $D = 1$, а на адресные входы – принимаемая кодовая комбинация. В зависимости от значения переменных на адресных входах логическая 1 будет лишь на одном из выходов демультиплексора.

При синтезе асинхронных мультиплексоров и демультиплексоров в их

таблицах истинности не учитывают переменную C (вход синхронизации).

2 Порядок выполнения работы

1 Изучить теоретический материал по учебнику и данным методическим указаниям.

2 Построить схему мультиплексора (демультиплексора) в соответствии с заданием, приведенным в таблице 4.3.

Т а б л и ц а 4.3 – Варианты заданий

Вариант	Тип устройства	Наличие синхронизации	Базис реализации	Количество входов данных мультиплексора (выходов демультиплексора)
1	Мультиплексор	да	И-НЕ	4
2	Мультиплексор	да	И-НЕ	5
3	Мультиплексор	да	И-НЕ	6
4	Мультиплексор	да	И-НЕ	7
5	Мультиплексор	да	И-НЕ	8
6	Мультиплексор	да	ИЛИ-НЕ	4
7	Мультиплексор	да	ИЛИ-НЕ	5
8	Мультиплексор	да	ИЛИ-НЕ	6
9	Мультиплексор	нет	И-ИЛИ-НЕ	9
10	Мультиплексор	нет	ИЛИ-НЕ	7
11	Мультиплексор	нет	ИЛИ-НЕ	8
12	Мультиплексор	нет	И-ИЛИ-НЕ	4
13	Мультиплексор	нет	И-ИЛИ-НЕ	5
14	Мультиплексор	нет	И-ИЛИ-НЕ	6
15	Мультиплексор	нет	И-ИЛИ-НЕ	7
16	Демультиплексор	да	И-ИЛИ-НЕ	8
17	Демультиплексор	да	И-НЕ	4
18	Демультиплексор	да	И-НЕ	5
19	Демультиплексор	да	И-НЕ	6
20	Демультиплексор	да	И-НЕ	7
21	Демультиплексор	да	И-НЕ	8
22	Демультиплексор	да	ИЛИ-НЕ	4
23	Демультиплексор	да	ИЛИ-НЕ	5
24	Демультиплексор	нет	ИЛИ-НЕ	6
25	Демультиплексор	нет	ИЛИ-НЕ	7
26	Демультиплексор	нет	ИЛИ-НЕ	8
27	Демультиплексор	нет	И-ИЛИ-НЕ	8
28	Демультиплексор	нет	И-ИЛИ-НЕ	5
29	Демультиплексор	нет	И-ИЛИ-НЕ	6
30	Демультиплексор	нет	И-ИЛИ-НЕ	7

3 Проверить правильность работы собранной схемы. Для этого подать на входы кодовые комбинации в соответствии с таблицей истинности заданного мультиплексора (демультиплексора) и зафиксировать выходные значения. При построении схем и верификации работы построенных устройств необходимо пользоваться средствами программы «Электронная лаборатория».

4 Сделать выводы об особенностях применения мультиплексоров и демультиплексоров при синтезе дискретных устройств на их базе.

Содержание отчета

Название и цель работы, условные обозначения мультиплексора и демультиплексора, ТИ и схемы построенного мультиплексора (демультиплексора) в совокупности со схемами проверки работоспособности, ответы на контрольные вопросы (по заданию преподавателя), выводы об особенностях применения заданного мультиплексора (демультиплексора).

Контрольные вопросы

- 1 Что называется мультиплексором?
- 2 Что называется демультиплексором?
- 3 Опишите алгоритм синтеза мультиплексора.
- 4 Опишите алгоритм синтеза демультиплексора.
- 5 В чем заключается основное отличие между мультиплексором и демультиплексором?
- 6 Для чего необходима синхронизация в мультиплексорах (демультиплексорах)?
- 7 Каким образом осуществляется подача входной информации, если число входов мультиплексора больше числа коммутируемых линий?
- 8 Для чего используется каскадное включение мультиплексоров?
- 9 Как осуществляется каскадное включение мультиплексоров?
- 10 Опишите применение демультиплексора в качестве дешифратора.

РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА

- 1 Сапожников В. В. и др. Дискретные устройства железнодорожной автоматики, телемеханики и связи: Учебник для вузов ж.-д. трансп. - М.: Транспорт, 1988. - 255 с.
- 2 Бочков К. А., Кострома Т. В., Сементюта А. Н. Синтез дискретных устройств: Методические указания для выполнения курсового проекта по курсу «Теоретические основы автоматики и телемеханики» / Белорус. ин-т инж. ж.-д. трансп. - Гомель: БелИИЖТ, 1991. - 46 с.
- 3 Сапожников В. В., Бочков К. А., Трохов В. Г., Кузьмин О. И. Теория дискретных устройств железнодорожной автоматики, телемеханики и связи: Методические указания к контрольной работе для студентов III курса специальности «Автоматика, телемеханика и связь на железнодорожном транспорте». – Гомель: БелИИЖТ, 1990. – 41 с.
- 4 Поспелов Д. А. Логические методы анализа и синтеза схем. М.: Энергия, 1974. - 368 с.

СОДЕРЖАНИЕ

Введение	3
<i>Лабораторная работа № 1.</i> Изучение программы «Электронная лаборатория»	3
.....	
<i>Лабораторная работа № 2.</i> Минимизация функций алгебры логики и построение дискретных схем с использованием логического конвертера программы «Электронная лаборатория»	11
.....	17
<i>Лабораторная работа № 3.</i> Изучение шифраторов, дешифраторов и преобразователей кодов.....	27
<i>Лабораторная работа № 4.</i> Изучение мультиплексоров и демультимплексоров	32
Рекомендуемая литература	

Учебное издание

Константин Афанасьевич БОЧКОВ
Юрий Федорович БЕРЕЗНЯЦКИЙ

Анализ функциональной структуры и синтез дискретных устройств

Лабораторный практикум по дисциплине «Теория дискретных устройств»

Часть I

Редактор Т. М. Р и з е в с к а я
Технический редактор В. Н. К у ч е р о в а
Корректор Н. А. Д а ш к е в и ч

Подписано в печать 4. 01. 2001 г. Формат бумаги 60x84_{1/16}.
Бумага газетная. Гарнитура Таймс. Печать офсетная.
Усл. печ. л. 2, 09. Уч.-изд. 1, 99. Тираж 250 экз.
Зак. № 2365. Изд. № 3581

Редакционно-издательский отдел БелГУТа, 246653, г. Гомель, ул. Кирова, 34.
Лицензия ЛВ № 57 от 22.10.97 г.

Типография БелГУТа, 246022, г. Гомель, ул. Кирова, 34.
Лицензия ЛП № 360 от 26.07.99 г.