

ГАЖК «Ўзбекистон темир йўллари»

Ташкентский институт инженеров железнодорожного транспорта

Методические указания к выполнению лабораторных работ по дисциплине
**«ЭЛЕКТРОНИКА И МИКРОПРОЦЕССОРНАЯ ТЕХНИКА С ПРИМЕНЕНИЕМ
ПРОГРАММЫ EWB»**

Методическое указание для подготовки бакалавров по направлению

5521300 “ Электротехника, электромеханика, электротехнология”

Ташкент– 2007

УДК 629.4.015.

Данное методическое указание к выполнению практических занятий по дисциплине **«Электроника и микропроцессорная техника с применением программы EWB»** для подготовки магистров по специальности составлено с учетом требований Государственного образовательного стандарта «Общие требования к необходимому содержанию и уровню подготовленности магистров».

Рекомендовано к изданию решением Учебно-методической комиссии Ташкентского института инженеров железнодорожного транспорта

Составители: Д.т.н., проф. Хромова Г. А.,
Ассистент Иксар Е. В.

Рецензенты: - Проректор по научной работе Ибрагимов
Н.Н.

Директор института механики
ИС САН РУз, д.т.н., профессор
Ризаев А.А.

ГАЖК “Ўзбекистон темир йўллари”
Ташкентский институт инженеров железнодорожного института

Разрешаю в печать
Проректор по учебной работе
доцент Ф.Ф. Каримова
«__» _____ 2007 г.

**«ЭЛЕКТРОНИКА И МИКРОПРОЦЕССОРНАЯ ТЕХНИКА
С ПРИМЕНЕНИЕМ ПРОГРАММЫ EWB»**

Методическое указание
для подготовки бакалавров по направлению

5521300 “Электротехника, электромеханика, электротехнология”

Ташкент– 2007

**«ЭЛЕКТРОНИКА И МИКРОПРОЦЕССОРНАЯ ТЕХНИКА
С ПРИМЕНЕНИЕМ ПРОГРАММЫ EWB»**

Ташкент– 2007

Редактор Асадова З.

Подписано в печать

Объем

п.л.

Формат бумаги 60x84 1/16

Тираж

Заказ №

Типография ТашИИТ.

Ташкент, Адылходжаева, 1

Лабораторная работа №1

I. Название лабораторной работы

Исследование логических элементов в программе EWB

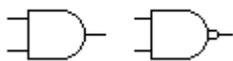
II. Цель лабораторной работы

Ознакомление с программой EWB. Изучение основных логических элементов и моделирование их в поле программы EWB.

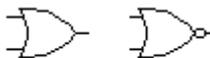
III. Основные сведения

В цифровых устройствах логические состояния представляются двумя уровнями напряжения (потенциалов): высоким, близким к напряжению источника питания, и низким, близким к 0, такой способ представления логических величин называется положительной логикой. Входы и выходы логических элементов в зависимости от уровня сигнала, при котором реализуется функция, подразделяются на прямые и инверсные.

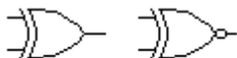
На рис.1 показаны обозначения базовых логических элементов, принятых в программе EWB, что соответствует обозначению элементов, реализующих основные логические функции:



Логические элементы И, И-НЕ



Логические элементы ИЛИ, ИЛИ-НЕ (количество входов)



Логические элементы ИСКЛЮЧАЮЩЕЕ ИЛИ и НЕ И-НЕ

Процесс создания схемы в программе EWB начинается с размещения элементов на рабочем поле (рис.2). Компоненты схемы выбираются из библиотеки программы. Разделы библиотеки могут быть вызваны поочередно с помощью окон, расположенных на верхней панели щелчком мыши. Рассмотрим пример моделирования логического элемента И.

Для моделирования логической схемы в программе EWB выбран логический преобразователь (рис.1). На лицевой панели преобразователя имеются клеммы – индикаторы входов А,В...Н и выхода OUT. Щелчком мыши по входным клеммам А,В...Н активизируем требуемое число входов логического анализатора. В представленной схеме активизированы окна А и В. В результате чего на экране анализатора получим начальную таблицу истинности исследуемого устройства. С помощью логического преобразователя можно проводить не только анализ логических устройств, но и их синтез.

Синтез логического устройства производится щелчком мыши по входам А,В...Н, в результате чего на экране анализатора получим начальную таблицу истинности, в

которой будут представлены все возможные комбинации входных сигналов и соответствующие им значения сигналов в столбце OUT.

В соответствии с заданием в полученную таблицу записываются 0 и 1.

Например, в первой строке при комбинации входных сигналов 0000 нужно на выходе получить не 0, а 1. Для этого ставим курсор мыши на первую строку в столбце OUT, производим щелчок и на клавиатуре введем 1.

Дальнейшие перемещения с целью редактирования остальных строк столбца OUT удобнее производить с помощью клавиш управления курсором.

Используя кнопки панели управления логического преобразователя, получим:

1. Таблицу истинности исследуемого устройства
2. Булево выражение, реализуемое устройством

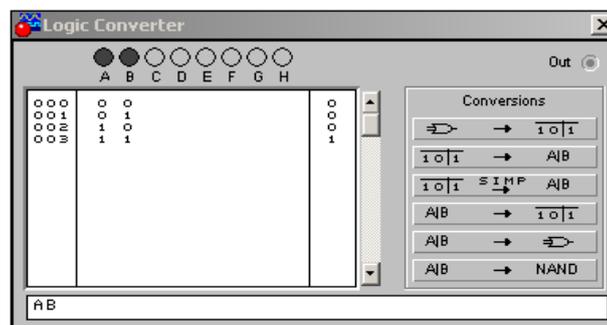
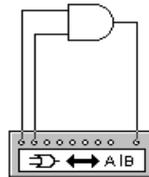


рис. 1

IV. Задание на лабораторную работу

1. В программе EWB исследовать основные логические функции, записать таблицу истинности.
2. Показать, что операция ИЛИ представима операциями И и НЕ.
3. С использованием логического преобразователя произвести синтез логического устройства с выходной комбинацией 10011110 в таблице истинности
4. С использованием логического преобразователя произвести проверку основных аксиом и законов булевой алгебры
5. На элементах И., ИЛИ, НЕ в программе EWB синтезировать устройство сигнал на выходе которого равен 1, только в том случае, когда на его двух входах действуют одинаковые сигналы.
6. В программе EWB синтезировать устройство с четырьмя входами, таким образом, чтобы на выходе появлялся сигнал 1, когда на трех входах будут одновременно сигналы 1.

V. Контрольные вопросы

1. Что является математической основой применения логических элементов?
2. Какие значения может принимать единица измерения информации 1 бит?

3. Что такое положительная логика?
4. На каком законе основывается взаимное преобразование логических элементов?
5. Как производится моделирование схем логических элементов в программе EWB?

Лабораторная работа №2

I. Название лабораторной работы

Исследование всех типов триггеров в программе EWB.

II. Цель лабораторной работы

Изучение основных типов триггеров.

III. Основные сведения

Триггер предназначен для хранения одной логической переменной. В соответствии с этим триггер имеет два состояния. Одно из них обозначается как состояние 0, другое как состояние 1. Воздействуя на входы триггера, его устанавливают в нужное состояние. Триггеры широко используются во многих узлах электронной аппаратуры как самостоятельные схемы, а также в качестве базовых элементов для построения цифровых устройств.

1. **R-S** триггеры. Представляет собой симметричную структуру, состоящую из двух логических элементов ИЛИ–НЕ или И–НЕ, охваченных перекрестной положительной обратной связью. Триггер на элементах И–НЕ, собранный в программе EWB, показан на рис.1.

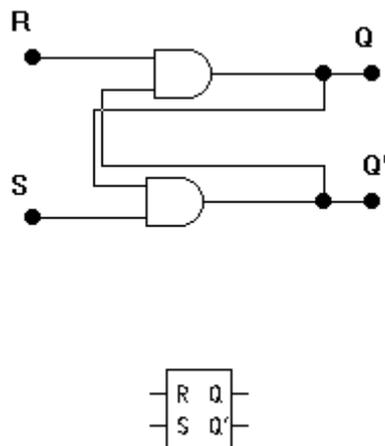


рис. 1

Такой триггер называется симметричным, он может использоваться как бистабильная ячейка, ячейка памяти. Переключение триггера из одного устойчивого состояния в другое происходит при подаче активных сигналов на входы R-S. При подаче на вход R уровня логической 1 первый элемент устанавливается в состояние, в котором на его прямом выходе действует уровень логического 0, следовательно на инверсном – 1, такое состояние триггера считается установкой в 0. Если триггер до подачи сигнала находился в состоянии 1, то при подаче на вход R уровня логического 0, триггер изменит состояние на противоположное. Такое состояние триггера называется установкой в 0. Процесс установления триггера в состояние 1 при подаче на

его вход $S=1$ аналогичен. Одновременная подача уровней лог.1 на оба входа R и S не допускается, так как при этом на обоих входах установится уровень лог. 0, а после снятия со входов активных логических уровней состояние триггера окажется неопределенным. Исследование R-S триггера в программе EWB производится при помощи генератора (Function Generator) и генератора слова (Word Generator) рис 2., которые выбираются в каталоге программ

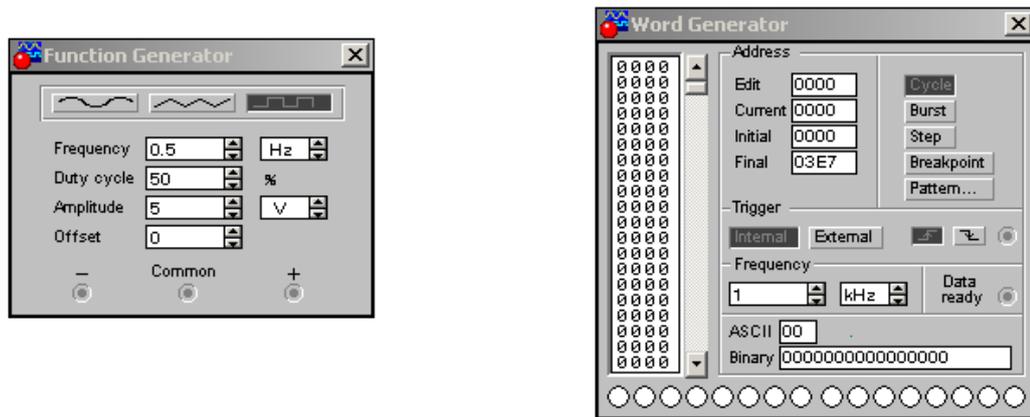


рис. 2

Пример исследования RS триггера показан на рис. 3

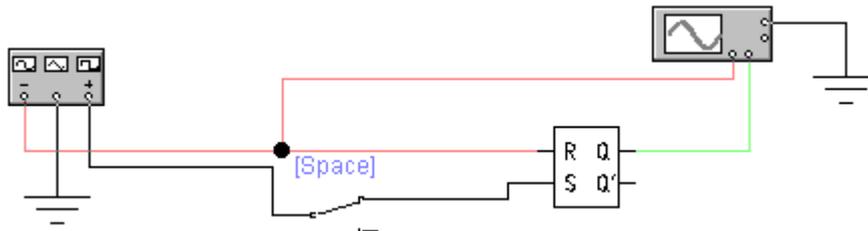


рис. 3

2.D – триггер, в отличие от R-S имеет для установки в состояния 0 и 1 один информационный D вход. D триггер образуется из R-S триггера, если на информационные входы одновременно подать инверсные сигналы (рис. 4).

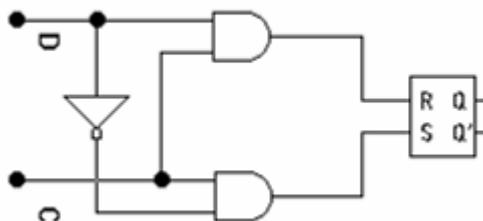


рис. 4

Исследование D триггера производится по схеме рис. 5 в следующем порядке:

1 – на входы предустановки триггера от генератора подаются соответствующие активные сигналы 0 или 1 и далее состояние триггера фиксируется по окну осциллографа.

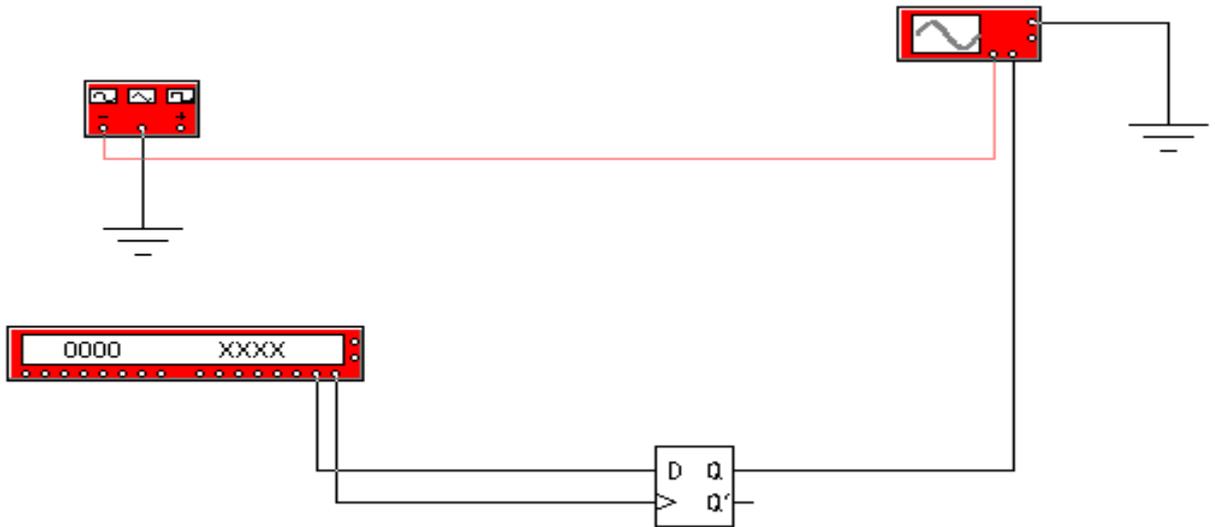


рис. 5

3. JK – триггер. Особенностью JK триггера является то, что при всех комбинациях входных сигналов, кроме $J=K=1$ он срабатывает подобно R-S триггеру. В каждом такте происходит опрокидывание триггера и выходные сигналы меняют свои значения. Это является универсальным свойством JK триггера и дает возможность использовать его как основу для построения основных функциональных схем: счетчиков, регистров.

На рисунке 6 показана схема JK триггера, собранная на основе RS триггера.

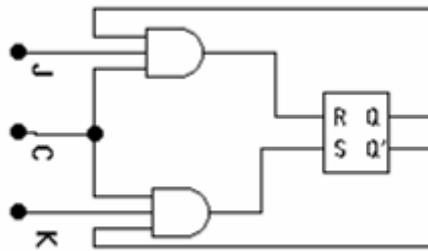


рис. 6

Схема исследования J-K триггера в программе EWB представлена на рис.7

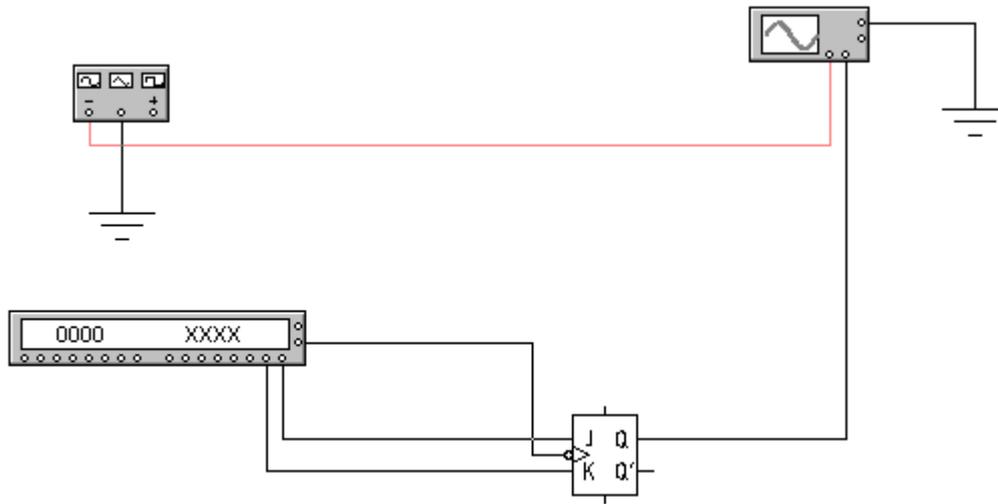


рис. 7

Лабораторная работа №3

I. Название лабораторной работы

Исследование счетчиков в программе EWB.

II. Цель лабораторной работы

Изучение всех типов счетчиков.

III. Основные сведения

Счетчиком называют цифровое устройство, сигналы на выходе которого отображают число импульсов, поступивших на счётный вход. Счётчик, образованный цепочкой из N триггеров, может посчитать в двоичном коде 2^N импульсов. Каждый триггер такой цепочки называется разрядом счётчика. Число N определяет количество разрядов двоичного числа, которое может быть записано в счётчик. Число K равное 2^N называется модулем (коэффициентом) счёта.

Счётчики классифицируются по следующим параметрам:

- 1 – Коэффициент счёта: двоичные, двоично-десятичные и с программируемым коэффициентом счёта;
- 2 – Направление счёта: суммирующие, вычитающие, реверсивные;
- 3 – Способ организации внутренних связей: последовательный, параллельный, комбинированный перенос и кольцевые счётчики.

Суммирующие, вычитающие реверсивные счётчики последовательного типа

В суммирующем счётчике при поступлении на вход очередного импульса происходит увеличение на единицу хранимого в счётчике числа. Таким образом, в счётчике устанавливается число, которое получается путём суммирования предыдущего разряда с единицей. Вычитающий счётчик действует обратным образом: двоичное число, хранящееся в счётчике, с каждым поступающим импульсом

уменьшается на 1. Реверсивный счётчик может работать в качестве суммирующего и вычитающего режима работы, определяется управляющими сигналами на входах. Увеличение разрядности счётчика осуществляется подключением дополнительных триггеров к соответствующему выходу последнего триггера.

Пример построения суммирующего двоичного счётчика показан на рисунке 8.

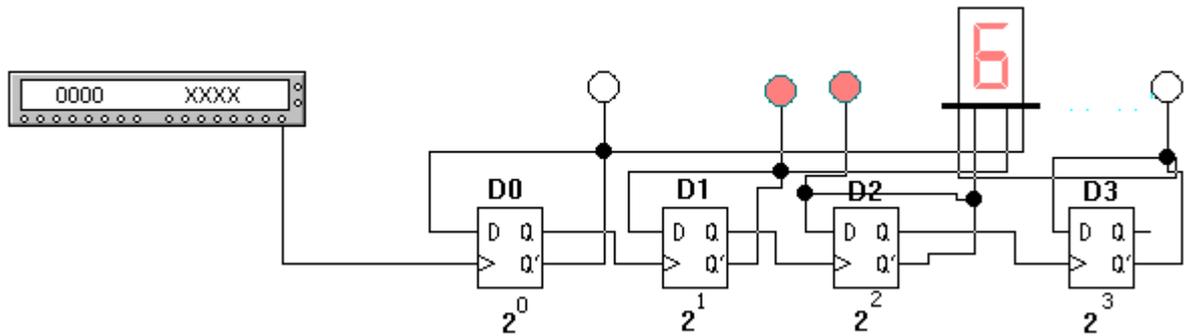


рис. 8

На входы счётчика подаются импульсы с выхода синхросигналов генератора слова, который генерируется при нажатии клавиши STEP (см рис.2) каждый триггер счётчика осуществляет деление на 2, сигнал переноса передаётся последовательно от одного разряда к другому для суммирующего счётчика с прямых входов Q, а для вычитающего с инверсных Q'.

Счётчик с параллельным переносом

В счётчике с параллельным переносом добавлены дополнительные логические элементы для формирования сигнала переноса, схема представлена на рис.9.

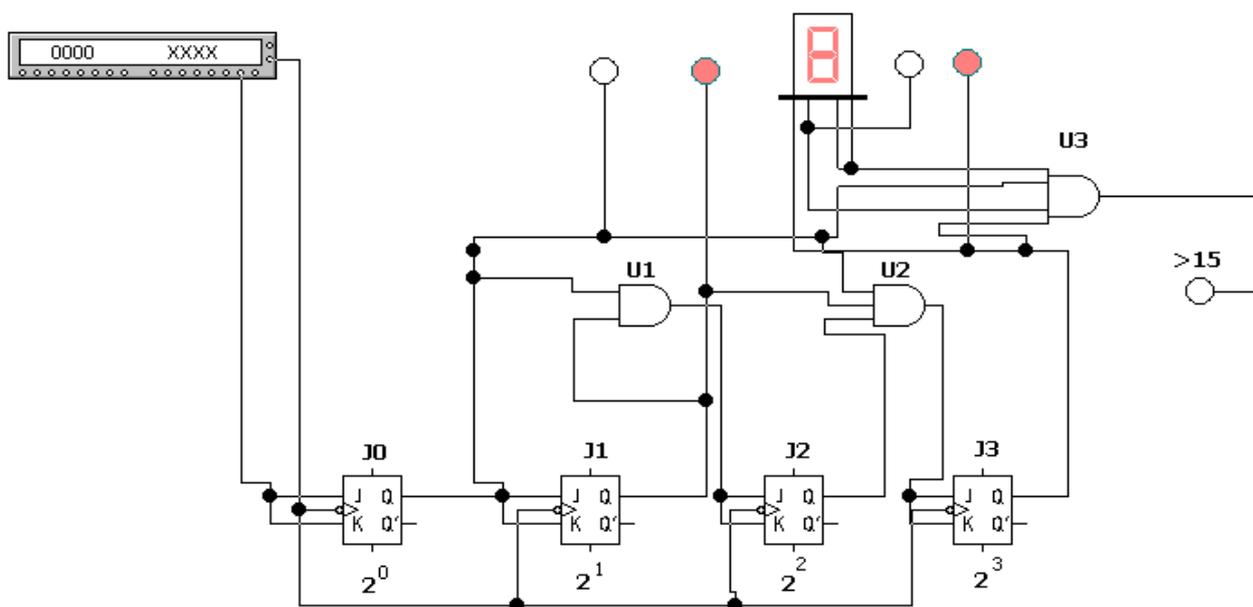


рис. 9.

Проектирование счётчика сводится к определению числа триггеров и организации связей между ними.

IV. Задание на лабораторную работу

1. Построить в программе EWB суммирующий вычитающий, реверсивный счётчики (рис.8 и рис.9) и произвести анализ их работы, построить временные диаграммы работы счётчика.
2. Построить схему суммирующего счётчика с коэффициентом пересчёта по заданию преподавателя.
3. Составить схему реверсивного счётчика с коэффициентом пересчёта 4.

V. Контрольные вопросы

1. Какие бывают счётчики?
2. Чем отличаются суммирующие счётчики от вычитающих?
3. Что такое программируемый счётчик?
4. Как организуется параллельный перенос?

Лабораторная работа № 4

I. Название лабораторной работы

Исследование регистров в программе EWB.

II. Цель лабораторной работы

Изучение всех типов регистров.

III. Основные сведения

Регистры – это устройства, предназначенные для хранения и преобразования многоразрядных двоичных чисел. Они используются в качестве управляющих и запоминающих устройств. Регистры выполняются на триггерах типа D или JK, одиночный триггер запоминает один разряд двоичной информации 1-бит. Занесение информации в регистр называется операцией ввода информации. Выдача информации называется операцией вывода или считывания, запись новой информации в регистр требует его предварительного обнуления.

В зависимости от функциональных свойств регистры подразделяются: накопительные регистры, регистры памяти, хранения и сдвигающие.

Сдвигающие регистры подразделяются:

- по способу ввода и вывода информации на параллельные, последовательные и комбинированные.
- по направлению передачи информации – однонаправленные и реверсивные.

Регистры памяти предназначены для хранения в течении короткого промежутка времени, ввод и вывод информации в такой регистр производится параллельным кодом. С приходом очередного тактового импульса записанная информация обновляется.

Регистры хранения представляют собой набор триггеров с независимыми информационными входами и общим тактовым входом. Нарастивание разрядности регистра осуществляется добавлением определённого числа триггеров. В регистрах памяти тактовые входы всех триггеров подсоединены к шине синхронизации, что даёт возможность хранить информацию до появления импульса синхронизации.

Последовательные регистры

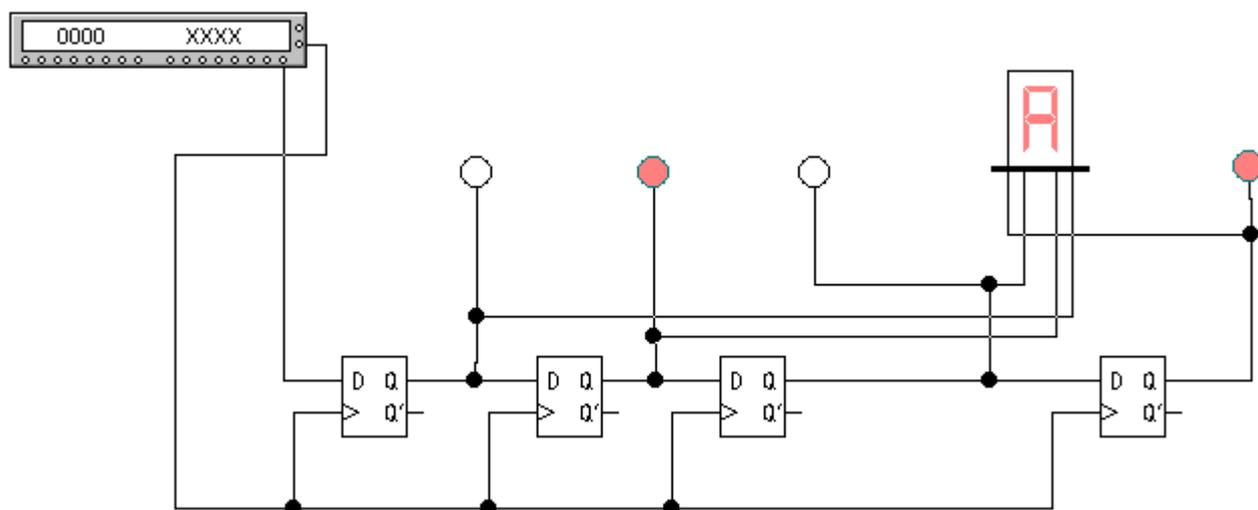


рис. 10

На рисунке 10 представлен пример построения регистра хранения информации на библиотечных D триггерах в программе EWB. Выход триггера каждого разряда подключен к входу D триггера соседнего более младшего разряда. Таким образом, при низком уровне синхронизирующего сигнала, хранящееся в триггере значение разряда числа передается на вход триггера соседнего справа разряда и производит в нем подготовку управляющих цепей. В момент положительного фронта синхронизирующего сигнала каждый из триггеров устанавливается в состояние, соответствующее действующему на входе D сигналу, и число в регистре оказывается сдвинутым на один разряд вправо. Для осуществления сдвига влево необходимо в

сдвиговом регистре изменить связи между триггерами, подключая выход триггера ко входу D триггера соседнего, более старшего разряда.

Параллельные регистры

В параллельном регистре предназначенное для хранения число подаётся одновременно всеми разрядами, т.е. в параллельной форме. При этом для каждого разряда числа предусматривается два входа, на один из которых поступает прямой код, на другой – инверсный. Пример построения такого регистра на R-S триггерах показан на рис.11.

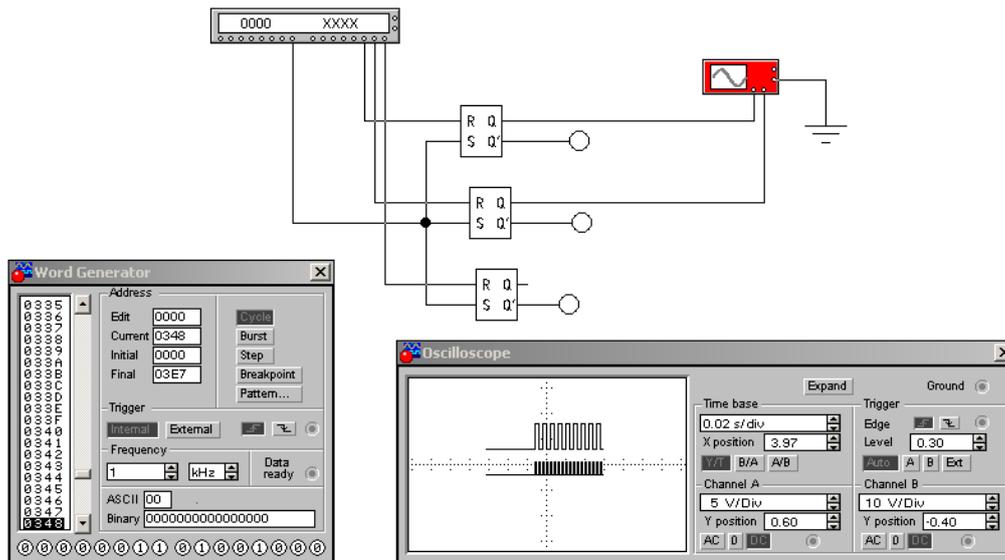


рис. 11

IV. Задание на лабораторную работу

1. Построить 8 разрядный параллельный регистр.
2. Построить схему 4 разрядного сдвигающего регистра на D триггерах, пояснить принцип работы регистра на примере продвижения единицы, построить временные диаграммы сигналов на выходах регистра.

V. Контрольные вопросы

1. Что из себя представляет регистр?
2. Назовите типы регистров.
3. Построить в программе EWB 8-разрядный сдвиговый регистр на библиотечных D триггерах.
 - а). Пояснить принцип работы регистра на примере продвижения одной единицы.
 - б). Построить временные диаграммы сигналов на выходах регистра при подаче шести сдвигающих импульсов.
 - в). Можно ли в регистре использовать J-K триггеры?
4. Построить 8-разрядный параллельный регистр в программе EWB.
 - а) Пояснить принцип работы регистра.
 - б) Построить временные диаграммы сигналов на выходах регистра.
5. Как можно использовать сдвигающий регистр в качестве распределителя импульсов (пояснить схемой)?

Лабораторная работа № 5

I. Название лабораторной работы

Исследование шифраторов и дешифраторов в программе EWB.

II. Цель лабораторной работы

Изучение принципа работы, построения и применения дешифраторов в цифровых устройствах.

III. Основные сведения

Шифратор (кодер) осуществляет преобразование десятичных чисел в двоичную систему счисления. Подача сигнала на один из входов приводит к появлению на выходах n-разрядного двоичного числа, соответствующего номеру возбужденного входа. Шифраторы широко используются в разнообразных устройствах ввода информации в цифровые системы.

Для ввода информации используется клавиатура. Нажатие десятичной клавиши соответствует генерации соответствующего кода. Поскольку в клавиатуре невозможно нажатие сразу нескольких клавиш, в шифраторах используется принцип приоритета старшего разряда, т.е. при нажатии, например, клавиш 9,5,2 на выходе шифратором будет генерироваться код 1001, соответствующий 9. Шифратор может быть построен на практически всех представленных в библиотеке EWB логических элементах, кроме того в библиотеке программы имеется микросхема ИМС-74148, что соответствует аналогу ИМС-155ИВ1 – шифратор приоритетов.

Пример построения схемы шифратора на ИМС- 74148 показан на рис. 12.

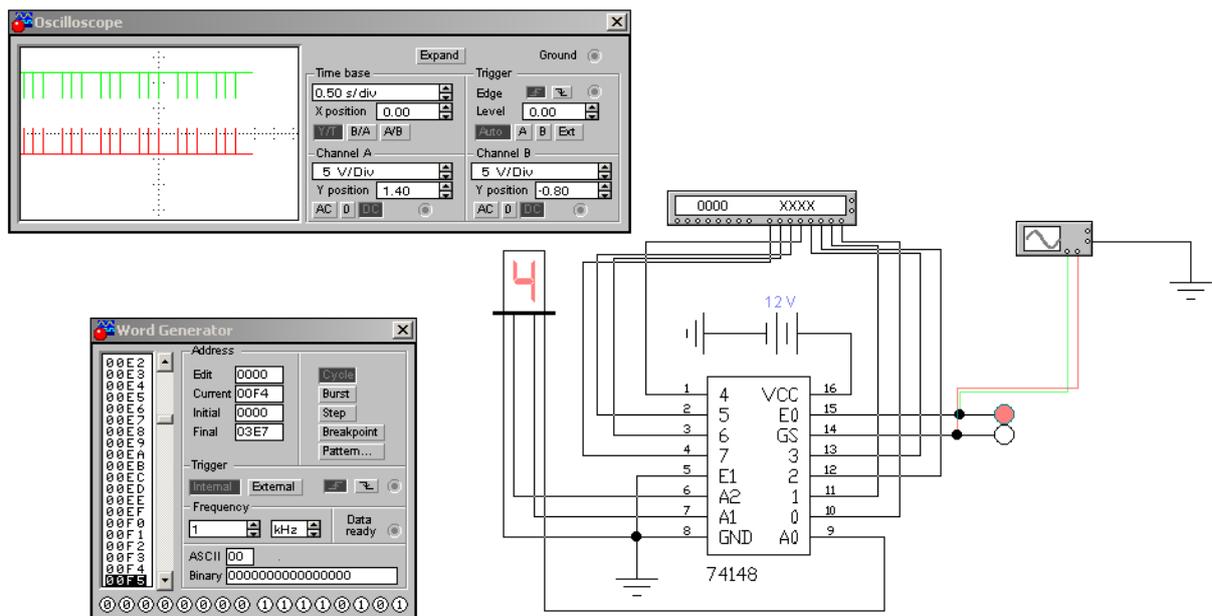


рис. 12

Дешифратор предназначен для обратного преобразования двоичных чисел в десятичные. Дешифраторы, как самостоятельные изделия электронной техники изготавливаются на 4, 8, 16 выходов. При необходимости количество входов наращивается. В качестве примера рассмотрим построение дешифратора элементами И-НЕ. Для подачи информационных сигналов на входы дешифратора используется логический генератор слова (рис.13).

Схема дешифратора содержит три инвертора и четыре элемента 3И-НЕ.

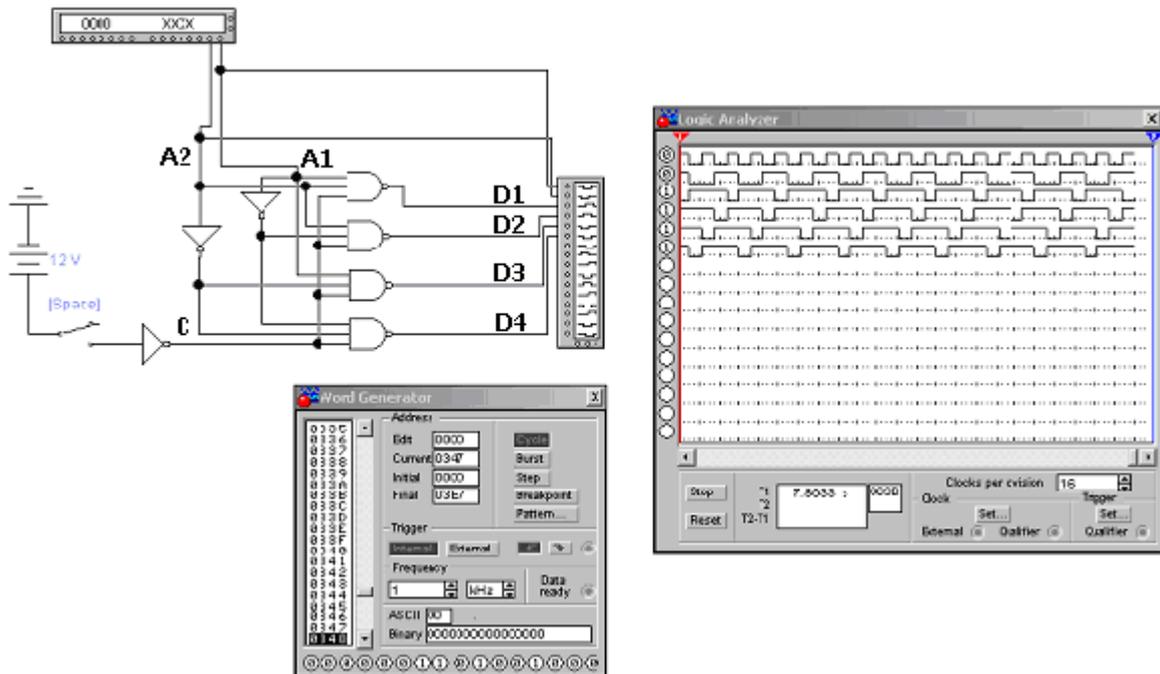


рис. 13

IV. Задание на лабораторную работу

1. Проведите моделирование дешифратора с двумя входами и четырьмя выходами в программе EWB.
2. Модернизируйте схему дешифратора 2*4 с использованием логического преобразователя и генератора слова библиотеки EWB, дополнив ее еще одним входом, проведите испытание.
3. Построить прямоугольный дешифратор с 4 входами и 16 выходами.

V. Контрольные вопросы

1. Что из себя представляет шифратор, при решении каких задач он используется?
2. Какой дешифратор называется линейным?
3. Какой дешифратор называется прямоугольным?
4. Какие логические элементы могут использоваться для построения дешифратора?

Лабораторная работа № 6

I. Название лабораторной работы

Исследование мультиплексора и демультиплексора в программе EWB.

II. Цель лабораторной работы

Изучение принципа построения, назначения и применения мультиплексора в цифровых устройствах.

III. Основные сведения

Устройство, которое осуществляет выборку одного из нескольких входов, и подключает его к своему выходу, называется мультиплексором. Мультиплексор имеет несколько информационных входов (D_0, D_1, \dots), адресные входы ($A_0, A_1 \dots$), вход для подачи стробирующего сигнала C и один выход Q . На рис. 14 показано символическое изображение мультиплексора с четырьмя информационными входами. Каждому информационному входу мультиплексора присваивается номер, называемый адресом. При подаче стробирующего сигнала на вход C мультиплексор выбирает один из выходов, адрес которого задается двоичным кодом на адресных входах и подключает его к выходу. Таким образом, подавая на адресные входы адреса различных информационных входов можно передавать цифровые сигналы с этих входов на выход Q . При отсутствии стробирующего сигнала связь между информационными входами и выходом отсутствует. Таким образом, сигнал высокого уровня на выходе Q появится при совпадении сигналов на соответствующих входах A , D и C .

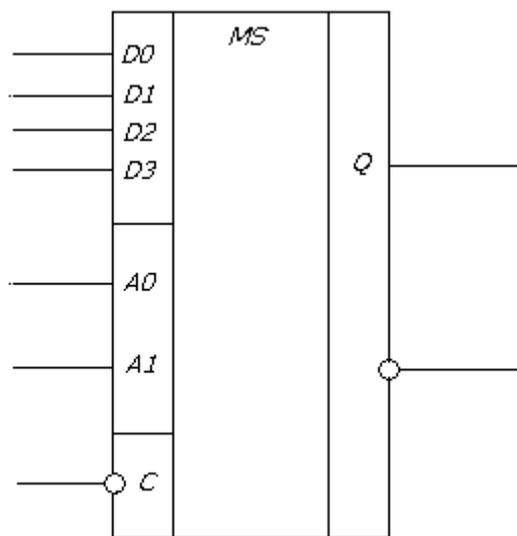


рис. 14

Моделирование мультиплексора в программе EWB показано на рис. 15. Для моделирования использован логический преобразователь, его выходной сигнал описывается булевым выражением $Y = BC + AC$, т.е. сигнал из канала A проходит на выход при адресном входе $C=0$, а из канала B , при $C=1$, что соответствует логике работы мультиплексора.

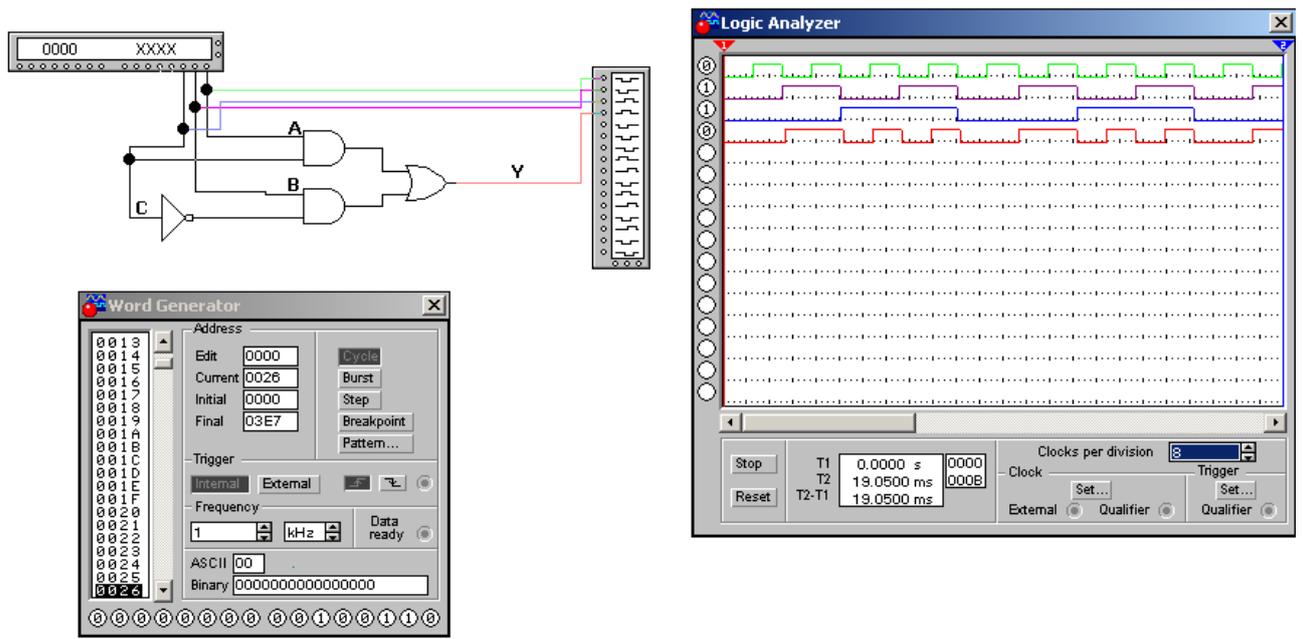


рис. 15

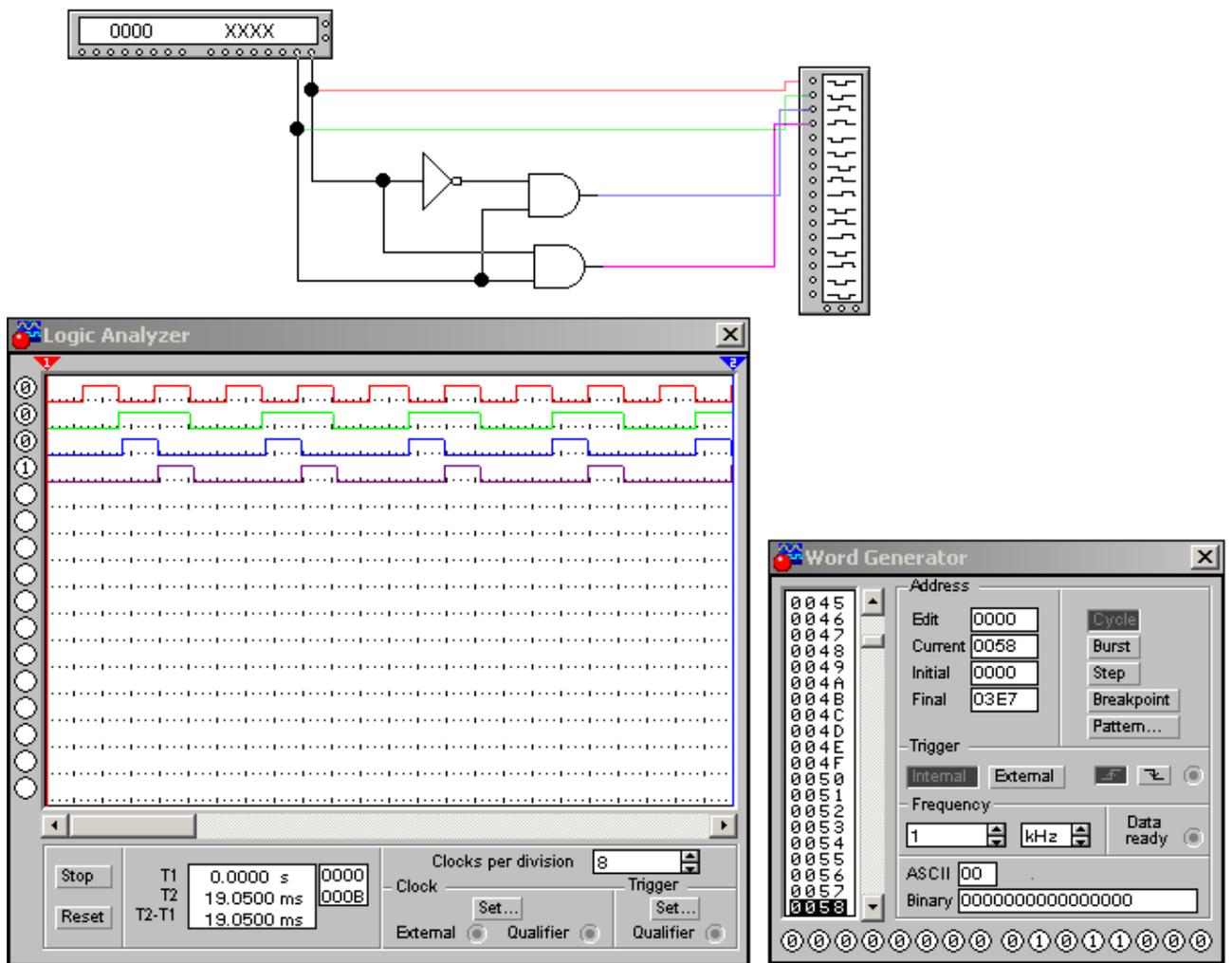


рис. 16

Демультимплексоры в функциональном отношении противоположны мультиплексорам. С их помощью сигналы с одного информационного входа распределяются в требуемой последовательности по нескольким выходам. Выбор нужной шины обеспечивается установкой соответствующего кода на адресных входах.

На рис. 16 представлен пример моделирования двухканального мультиплексора. Схема содержит два элемента И и один элемент НЕ. При $A=0$ сигнал с информационного входа передается на выход $Y1$, а при $A=1$ – на выход $Y2$.

IV. Задание на лабораторную работу

1. Используя методику анализа двухканального мультиплексора, с помощью логического анализатора постройте и исследуйте четырехканальный мультиплексор.
2. Используя методику анализа двухканального мультиплексора, с помощью логического анализатора постройте и исследуйте четырехканальный демультимплексор.
3. Подготовьте схему и проведите испытания.

V. Контрольные вопросы

1. Что из себя представляет мультиплексор, каково его назначение?
2. Что такое демультимплексор, для решения каких задач его можно применять?

Лабораторная работа № 7

I. Название лабораторной работы

Исследование преобразователя кодов и цифрового компаратора в программе EWB.

II. Цель лабораторной работы

Назначение и принцип построения компаратора и преобразователя кодов.

III. Основные сведения

Цифровым компаратором называется устройство, предназначенное для сравнения двух чисел представленных в виде двоичного или двоично-десятичного кода. Компараторы определяют равенство или неравенство чисел. И результат сравнения отображается в виде логического сигнала на соответствующем выходе компаратора. Цифровые компараторы применяются для выявления нужного числа (слова) в любых цифровых последовательностях, например: для отметки времени в электронных часах, для выявления условных переходов в вычислительных устройствах, адресных селекторах. На рис. 17 представлена схема одноразрядного компаратора. Компаратор состоит из 2-х элементов НЕ, четырёх элементах И, и одного элемента ИЛИ-НЕ, число в двоичном коде формируется логическим преобразователем и подаётся на входы А и В компаратора. В случае неравенства чисел А и В на выходе компаратора появляется 1. В случае равенства 0, что отображается таблицей истинности.

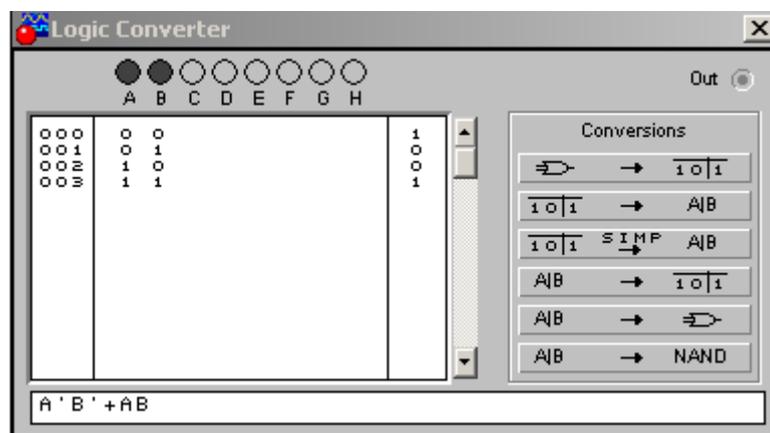
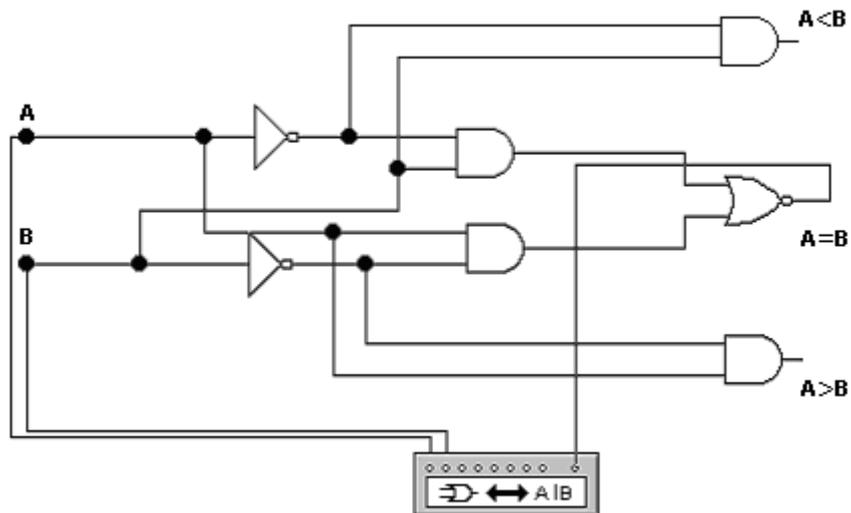


рис. 17

Устройство контроля четности

Операция контроля четности повышает надежность передачи и обработки информации. Операция заключается в суммировании по модулю 2 всех разрядов с целью выяснения четности числа, например, если при передаче кода 1001 произойдет сбой во 2 разряде, то мы получим код 1101. Обнаружение ошибки производится следующим образом. На передающей стороне передаваемый код дополняется контрольным битом до четного или нечетного числа единиц в суммарном коде. В случае нечетного кода дополнительный бит формируется таким образом, чтобы сумма всех единиц в передаваемом коде, включая контрольный бит, была нечетной. При контроле четности все происходит наоборот. Например, в числе 0111 число единиц нечетно. Поэтому при контроле нечетности дополнительный бит должен быть нулем, а при контроле четности единицей. На практике чаще всего используется контроль нечетности, поскольку он позволяет фиксировать полное пропадание информации (случай нулевого кода во всех информационных разрядах). На приемной стороне производится проверка кода четности. Если он правильный, то прием разрешается, в противном случае включается сигнализация ошибки или посылается передатчику запрос на повторную передачу. Схема формирователя бита четности для 4 разрядного кода представлена на рис. 18.

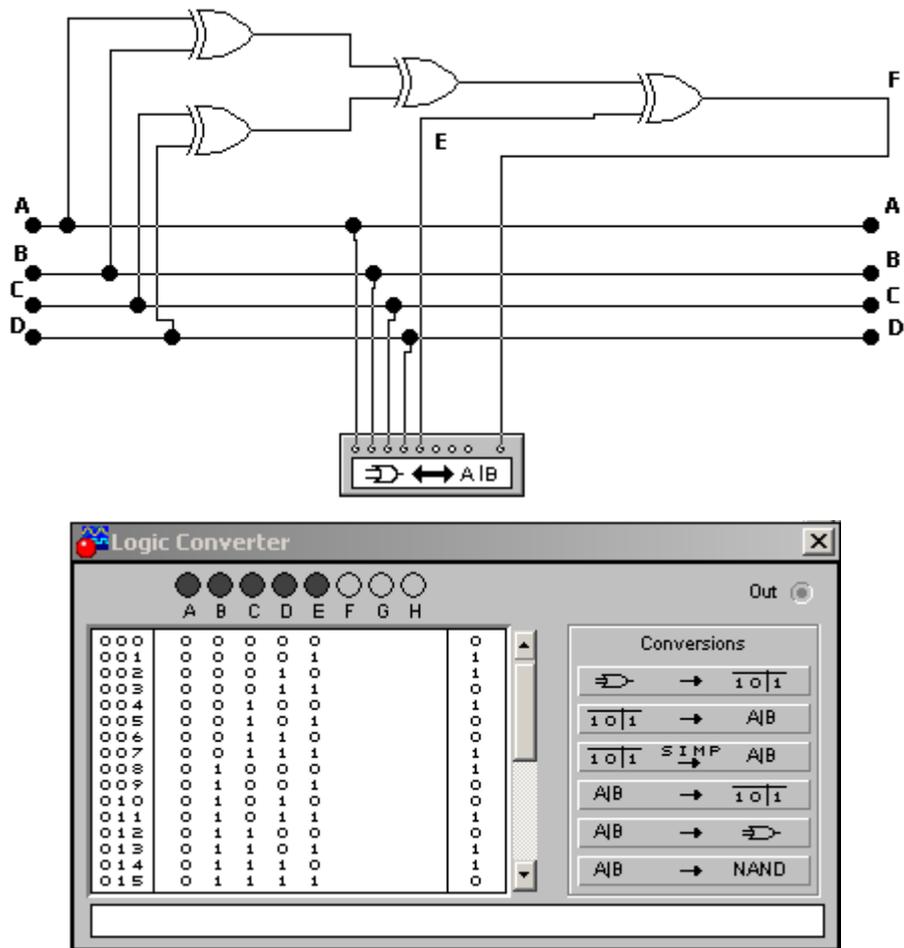


Рис. 18.

IV. Задание на лабораторную работу

1. Исследовать схему компаратора с помощью логического анализатора в программе EWB, подсоединяя клемму OUT последовательно ко всем выходам компаратора, получить на логическом анализаторе таблицу истинности для каждого выхода.
2. Построить схему формирования бита четности 8 разрядного кода. При помощи логического анализатора получить таблицу истинности.

V. Контрольные вопросы

1. Какие функции выполняет цифровой компаратор?
2. В каких устройствах может использоваться цифровой компаратор?
3. Составить схему 3 разрядного компаратора, используя схему 3 входных логических элементов.

Лабораторная работа № 8

I. Название лабораторной работы

Исследование микросхем памяти в программе EWB.

II. Цель лабораторной работы

Изучение полупроводниковых запоминающих устройств.

III. Основные сведения

По выполняемым функциям различают следующие типы запоминающих устройств:

Оперативное запоминающее устройство (ОЗУ).

Постоянное запоминающее устройство (ПЗУ).

Перепрограммируемое запоминающее устройство (ППЗУ).

ОЗУ используется при необходимости выбрать и обновить хранимую информацию в высоком темпе работы процессора цифрового устройства. В ОЗУ предусматриваются 3 режима работы:

1 – хранение при отсутствии обращения к ЗУ.

2 – режим чтения хранимых слов.

3 – режим записи (время чтения и записи слова в ОЗУ составляют доли миллисекунды).

ОЗУ используются для хранения исходных данных промежуточных, промежуточных и конечных результатов обработки данных. Представляет из себя набор регистров.

ПЗУ предназначено для хранения постоянного хранения информации и представляет из себя магнитный диск с набором реализуемых программ.

ППЗУ представляет из себя программатор и предназначен для отладки программы.

Запоминающее устройство содержит N ячеек, в каждой из которых хранится слово с определённым числом разрядов. Номер ячейки называется адресом. Количество информации, которое может храниться в ЗУ называется емкостью.

Быстродействие ЗУ характеризуется временем выборки и циклом записи, определяемым минимально допустимым временем подачи сигнала выборки при записи и моментом когда допустимо последующее обращение к памяти.

Оперативные Запоминающие Устройства (ОЗУ)

ОЗУ строятся из набора однотипных микросхем и являются неотъемлемой частью микропроцессорных систем различного назначения, они делятся на 2 класса: статические и динамические. В статических ОЗУ запоминание информации производится на триггерах, а в динамических – на конденсаторах емкостью 0,5 пФ. Конструктивно ОЗУ выполнено в виде двух блоков – матрицы запоминающих элементов и дешифратора адреса. Матрица состоит из элементов памяти, расположенных вдоль строк и столбцов и имеет двухкоординатную дешифрацию адреса – по строкам и столбцам. При этом в выбранной ячейке памяти срабатывает двухвходовой элемент И и подготавливает цепи чтения записи информации на входных или выходных разрядных шинах. При записи в ячейку памяти (см. рис. 19) на соответствующей разрядной шине данных устанавливается 1 или 0 на входе А устанавливается сигнал 1, и после стробирования сигналом С срабатывают элементы 2И. Положительный импульс с элемента V2 поступает на тактовый вход D-триггера, в результате чего в нем записывается 1 или 0, в зависимости от уровня сигнала на его D- входе.

При чтении из ячейки памяти на входе А устанавливается 0, при этом срабатывают элементы V1, V3, V5, и на вход буферного элемента V6 поступает разрешающий сигнал, в результате сигнал с выхода триггера передается на разрядную шину данных.

Микросхемы памяти имеют четыре основные характеристики: тип, объём, структура и время доступа. На рис.20 представлено условное графическое изображение микросхемы памяти.

Рис. 20

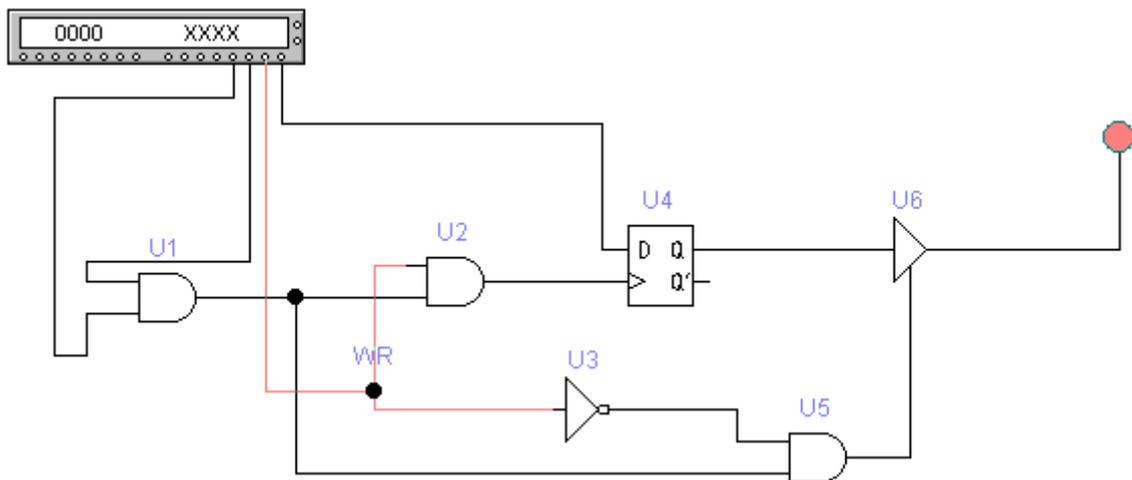


рис. 19. Схема ячейки памяти mem-i ОЗУ.
Постоянные Запоминающие Устройства (ПЗУ)

ПЗУ делятся на: ПЗУ, программируемые маской на заводе- изготовителе, и ПЗУ, программируемые пользователем. Рассмотрим ПЗУ второго типа, которое состоит из дешифратора и подключенных к его выходам схем ИЛИ с плавкими перемычками(рис21).

Ячейка ПЗУ работает следующим образом: В исходном состоянии транзисторы T1...T4 закрыты, и при подключенной к T6 нагрузке на его выходе D0 формируется сигнал логической единицы (+5В). При подаче на входы А, В дешифратора заданной кодовой комбинации, на вход разрешения Е сигнала логической единицы, один из транзисторов T1...T4 откроется и на выходе D0 сформируется сигнал логического нуля. Так, например, при A=B=1 откроется транзистор T4 и сигнал логической единицы с его эмиттера через перемычку S4 поступит на делитель на резисторах R2, R3, транзистор T6 откроется, и на его выходе сформируется сигнал логического нуля. Очевидно, что при любой другой двоичной комбинации будет происходить тоже самое, до тех пор, пока не будет разрушаться соответствующая перемычка. Пережигание перемычек составляет суть программирования и осуществляется отдельно для каждого разряда (каждой ячейки) следующим образом:

– На входы А, В (рис 21) подаётся двоичная комбинация, соответствующая адресу пережигаемой перемычки в программируемом разряде.

– к выходу ячейки D_x через резистор нагрузки (около 300 Ом) подключается источник напряжения 12,5В, в результате чего стабилитрон D пробивается и транзистор T5 открывается.

– На вход разрешения E на короткое время подаётся сигнал логической единицы, при этом через один из открытых транзисторов T1...T2 и T5 протекает ток, достаточный для пережигания соответствующей перемычки.

– Источник 12,5В отключается после раскрытия соответствующей подсхемы, таким образом соответствующая перемычка пережигается.

Заключительным этапом программирования серийно изготавливаемых микросхем ПЗУ в промышленных условиях является электротермотренировка, которая производится в течение 168 часов при повышенной температуре, после чего производится дополнительный контроль записываемой информации. Если обнаруживается ошибка, производится повторное программирование. Модель ПЗУ представлена на рис.22. ПЗУ пережиганием перемычек применяется в качестве специализированных дешифраторов. В основном, в микропроцессорных системах используются ПЗУ с ультрафиолетовым стиранием