

РОСЖЕЛДОР
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«РОСТОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ПУТЕЙ
СООБЩЕНИЯ»
(ФГБОУ ВО РГУПС)
ТЕХНИКУМ
(ТЕХНИКУМ ФГБОУ ВО РГУПС)

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ
ПО ВЫПОЛНЕНИЮ ПРАКТИЧЕСКИХ РАБОТ**

ОП. 11 СХЕМОТЕХНИКА

ДЛЯ СПЕЦИАЛЬНОСТИ 09.02.04

ИНФОРМАЦИОННЫЕ СИСТЕМЫ (ПО ОТРАСЛЯМ)

Базовая подготовка среднего профессионального образования

Ростов-на-Дону

2016

Рассмотрены
Предметной (цикловой)
комиссией специальности
«Информационные системы»

Прот. от 30.08.2016

Председатель:



Методические указания разработаны
на основании рабочей программы
ОП.11 Схемотехника специальности
09.02.04 Информационные системы
(по отраслям)

Заместитель
директора по УМР



Разработчик: Ливинский С.В., преподаватель техникума ФГБОУ ВО
РГУПС, к.т.н.

Рекомендована объединенной методической комиссией техникума
ФГБОУ ВО РГУПС.

Заключение ОМК № 1 от «26» сентября 2016г.

СОДЕРЖАНИЕ

1. Паспорт.....	5
1.1 Область применения.....	5
1.2 Цели и задачи – требования к результатам освоения дисциплины.....	5
1.3 Количество часов.....	6
2. Структура, содержание и методические указания к выполнению прак- тических работ.....	7
2.1 Распределение компетенций по тематикам практических занятий.....	7
2.2 Методика выполнения практических занятий.....	8
3. Контроль и оценка результатов освоения учебной дисциплины.....	144

1. ПАСПОРТ

1.1 Область применения

Методические указания по выполнению практических занятий дисциплины «Схемотехника» предназначены для изучения дисциплины в учреждениях среднего профессионального образования, реализующих образовательную программу среднего (полного) общего образования, при подготовке квалифицированных рабочих и специалистов среднего звена.

1.2 Цели и задачи – требования к результатам освоения дисциплины

Целью выполнения практических занятий по ОП. 11 «Схемотехника» является освоение следующих компетенций:

КОД	Наименование результата обучения
ПК 1.2	Взаимодействовать со специалистами смежного профиля при разработке методов, средств и технологий применения объектов профессиональной деятельности
ПК 1.3	Производить модификацию отдельных модулей информационной системы в соответствии с рабочим заданием, документировать произведенные изменения.
ОК 1	Понимать сущность и социальную значимость своей будущей профессии, проявлять к ней устойчивый интерес.
ОК 2	Организовывать собственную деятельность, выбирать типовые методы и способы выполнения профессиональных задач, оценивать их эффективность и качество.
ОК 3	Принимать решение в стандартных и нестандартных ситуациях и нести за них ответственность.
ОК 4	Осуществлять поиск и использование информации, необходимой для эффективного выполнения профессиональных задач, профессионального и личностного развития.
ОК 5	Использовать информационно-коммуникационные технологии в профессиональной деятельности.
ОК 6	Работать в коллективе и в команде, эффективно общаться с коллегами, руководством, потребителями.
ОК 7	Брать на себя ответственность за работу членов команды (подчиненных), за результат выполнения заданий.
ОК 8	Самостоятельно определять задачи профессионального и личностного развития, заниматься самообразованием, осознанно планировать повышение квалификации.
ОК 9	Ориентироваться в условиях частой смены технологий в профессиональной деятельности.

В результате выполнения практических занятий обучающийся **должен знать:**

- формы представления информации в схемах автоматики и вычислительной техники;
- основы алгебры логики;
- основы проектирования цифровых устройств;
- условно графические обозначения элементов;
- принцип работы элементов памяти;
- структуру и организацию микропроцессорных наборов;
- принцип построения и работы типовых узлов ЭВМ;
- современное состояние и тенденции развития микросхемотехнических средств вычислительной техники.

В результате выполнения практических занятий обучающийся **должен уметь:**

- производить арифметические вычисления в двоичной системе;
- выполнять синтез логических схем;
- читать условно-графическое обозначение (УГО) и принципиальные схемы;
- увеличивать адресность и разрядность модулей памяти;
- строить функциональные и принципиальные схемы узлов и устройств ЭВМ с соблюдением требований стандартов;
- производить сравнительную оценку проектируемых схем, с учетом основных параметров;
- описать работу синтезированного устройства с помощью таблиц истинности и временных диаграмм;
- составлять алгоритм работы цифрового устройства;
- пользоваться технической и справочной литературой.

1.3 Количество часов:

— практические занятия в количестве **24** часа.

2 СТРУКТУРА, СОДЕРЖАНИЕ И МЕТОДИЧЕСКИЕ УКАЗАНИЯ К ВЫПОЛНЕНИЮ ПРАКТИЧЕСКИХ ЗАНЯТИЙ

2.1 Распределение компетенций по тематикам практических занятий

Практическое занятие 1	Перевод чисел из одной системы счисления в другую.	ОК 1-9 ПК 1.2
Практическое занятие 2	Минимизация функции.	ОК 1-9 ПК 1.2
Практическое занятие 3	Моделирование логических элементов.	ОК 1-9 ПК 1.2, 1.3
Практическое занятие 4	Исследование работы логических элементов.	ОК 1-9 ПК 1.2, 1.3
Практическое занятие 5	Исследование работы RS и D триггеров.	ОК 1-9 ПК 1.2, 1.3
Практическое занятие 6	Исследование работы JK и T триггеров.	ОК 1-9 ПК 1.2, 1.3
Практическое занятие 7	Исследование работы регистров.	ОК 1-9 ПК 1.2, 1.3
Практическое занятие 8	Исследование работы счетчиков.	ОК 1-9 ПК 1.2, 1.3
Практическое занятие 9	Исследование работы дешифратора.	ОК 1-9 ПК 1.2, 1.3
Практическое занятие 10	Исследование работы мультиплексора.	ОК 1-9 ПК 1.2, 1.3
Практическое занятие 11	Исследование работы ОЗУ и ПЗУ.	ОК 1-9 ПК 1.2, 1.3
Практическое занятие 12	Исследование работы АЛУ.	ОК 1-9 ПК 1.2, 1.3

2.2 Методика выполнения практических занятий

Практическое занятие №1

Перевод чисел из одной системы счисления в другую

Цель занятия: получить практические навыки операций в различных системах счисления.

Краткие теоретические сведения

В современной вычислительной технике информация чаще всего кодируется с помощью последовательности сигналов всего двух видов: включено или не включено, намагничено или ненамагничено, высокое или низкое напряжение и т.д. Принято обозначать одно состояние цифрой 0, а другое - 1. Такое представление информации в цифровом виде называют двоичным. Набор (последовательность) из нулей и единиц называют двоичным кодом.

Система счисления - совокупность приемов наименования и обозначения чисел. Системы счисления разделяются на две группы: позиционные и непозиционные. Позиционной называется система счисления, в которой значение цифры зависит от ее места (позиции) в ряду цифр, обозначающих число. Системы, не обладающие этим свойством, называются непозиционными (римская система счисления). Основанием позиционной системы счисления называется число цифр, которое используют при записи.

В ЭВМ часто используется восьмеричная и шестнадцатеричная системы счисления. В восьмеричной системе счисления числа записываются с помощью восьми цифр (0 1 2 3 4 5 6 7). Сама восьмерка записывается двумя цифрами: 10. Для записи чисел в шестнадцатеричной системе необходимо уже располагать шестнадцатью различными символами, используемыми как цифры:

10-я: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

16-я: 0 1 2 3 4 5 6 7 8 9 A B C D E F

Пример 1. Переведем десятичное число 46 в двоичную систему счисления.

Правило: Чтобы перевести целое положительное десятичное число в систему счисления с другим основанием, нужно это число разделить на основание. Полученное частное снова разделить на основание и т.д. до тех пор, пока частное не окажется меньше основания. В результате записать в одну строку последнее частное и все остатки, начиная с последнего.

$$\begin{array}{r}
 \begin{array}{l}
 \begin{array}{r}
 46 \\
 \hline
 46 \\
 \hline
 0
 \end{array}
 \end{array}
 \begin{array}{l}
 \begin{array}{r}
 2 \\
 \hline
 23 \\
 \hline
 22 \\
 \hline
 0
 \end{array}
 \end{array}
 \begin{array}{l}
 \begin{array}{r}
 2 \\
 \hline
 11 \\
 \hline
 10 \\
 \hline
 1
 \end{array}
 \end{array}
 \begin{array}{l}
 \begin{array}{r}
 2 \\
 \hline
 5 \\
 \hline
 4 \\
 \hline
 1
 \end{array}
 \end{array}
 \begin{array}{l}
 \begin{array}{r}
 2 \\
 \hline
 2 \\
 \hline
 2 \\
 \hline
 0
 \end{array}
 \end{array}
 \begin{array}{l}
 \begin{array}{r}
 2 \\
 \hline
 2 \\
 \hline
 1
 \end{array}
 \end{array}
 \end{array}$$

$$46 = 101100_2.$$

Пример 2. Переведем десятичное число 672 в восьмеричную систему счисления.

$$\begin{array}{r}
 \begin{array}{l}
 \begin{array}{r}
 672 \\
 \hline
 672 \\
 \hline
 0
 \end{array}
 \end{array}
 \begin{array}{l}
 \begin{array}{r}
 8 \\
 \hline
 84 \\
 \hline
 80 \\
 \hline
 4
 \end{array}
 \end{array}
 \begin{array}{l}
 \begin{array}{r}
 8 \\
 \hline
 10 \\
 \hline
 8 \\
 \hline
 2
 \end{array}
 \end{array}
 \begin{array}{l}
 \begin{array}{r}
 8 \\
 \hline
 1
 \end{array}
 \end{array}
 \end{array}$$

$$672 = 1240_8.$$

Пример 3. Переведем десятичное число 934 в шестнадцатеричную систему счисления.

$$\begin{array}{r}
 \begin{array}{l}
 \begin{array}{r}
 934 \\
 \hline
 928 \\
 \hline
 6
 \end{array}
 \end{array}
 \begin{array}{l}
 \begin{array}{r}
 16 \\
 \hline
 58 \\
 \hline
 48 \\
 \hline
 10
 \end{array}
 \end{array}
 \begin{array}{l}
 \begin{array}{r}
 16 \\
 \hline
 3
 \end{array}
 \end{array}
 \end{array}$$

$$934 = 3A6_{16}.$$

Пример 4. Переведем в двоичную систему счисления положительную десятичную дробь 0,3.

Правило: Чтобы перевести положительную десятичную дробь в двоичную, нужно дробь умножить на 2. Целую часть произведения взять в качестве первой цифры после запятой в двоичной дроби, а дробную часть вновь умножить на 2. В качестве следующей цифры двоичной дроби взять целую часть этого произведения, а дробную часть произведения снова умножить на 2 и т.д. до получения после запятой заданного количества цифр.

$$\begin{array}{r|l}
 0, & 3 \\
 \hline
 & \times 2 \\
 \hline
 0, & 6 \\
 \hline
 & \times 2 \\
 \hline
 1, & 2 \\
 \hline
 & \times 2 \\
 \hline
 0, & 4 \\
 \hline
 & \times 2 \\
 \hline
 0, & 8 \\
 \hline
 & \times 2 \\
 \hline
 1, & 6
 \end{array}$$

Дробная часть 0,6 уже была на втором шаге вычислений. Поэтому вычисления будут повторяться. Следовательно в двоичной системе счисления число 0,3 представляется периодической дробью:

$$0,3 = 0,0(1001)_2.$$

Пример 5. Переведем в двоичную систему счисления положительную десятичную дробь 0,625.

$$\begin{array}{r|l}
 0, & 625 \\
 \hline
 & \times 2 \\
 \hline
 1, & 250 \\
 \hline
 & \times 2 \\
 \hline
 0, & 500 \\
 \hline
 & \times 2 \\
 \hline
 1, & 0
 \end{array}$$

$$0,625 = 0,101_2.$$

Замечание: Перевод десятичного числа в двоичную систему счисления проводится отдельно для его целой и дробной части.

Пример 6. Переведем в десятичную систему счисления двоичное число 1011,011.

Правило: Чтобы перевести число из двоичной системы в десятичную систему счисления, нужно двоичное число представить в виде суммы степеней двойки с коэффициентами-цифрами и найти эту сумму.

$$1011,011_2 = 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 + 0 \cdot 2^{-1} + 1 \cdot 2^{-2} + 1 \cdot 2^{-3} \\ = 1 \cdot 8 + 1 \cdot 2 + 1 \cdot 1 + 1 \cdot (1/2) + 1 \cdot (1/4) + 1 \cdot (1/8) = 8 + 2 + 1 + 1/2 + 1/4 + 1/8 = 11,375$$

$$1011,011_2 = 11,375_{10}.$$

Пример 7. Переведем в десятичную систему счисления восьмеричное число 511.

$$511_8 = 5 \cdot 8^2 + 1 \cdot 8^1 + 1 \cdot 8^0 = 5 \cdot 64 + 1 \cdot 8 + 1 = 329$$

$$511_8 = 329_{10}.$$

Пример 8. Переведем в десятичную систему счисления шестнадцатеричное число 1151.

$$1 \cdot 16^3 + 1 \cdot 16^2 + 5 \cdot 16^1 + 1 \cdot 16^0 = 1 \cdot 4096 + 1 \cdot 256 + 5 \cdot 16 + 1 = 4096 + 256 + 80 + 1 = 4433.$$

$$1151_{16} = 4433_{10}.$$

Пример 9. Переведем двоичное 1100001111010110 число в восьмеричную форму.

Правило: Для преобразования двоичного числа в восьмеричное необходимо двоичную последовательность разбить на группы по три цифры справа налево и каждую группу заменить соответствующей восьмеричной цифрой. Аналогично поступают и при переводе в шестнадцатеричную систему, только двоичную последовательность разбивают не на три, а на четыре цифры.

Десятичн ое число	Восьмеричн ое число	Двоичн ая запись	Шестнадц ате- ричное число	Двоичн ая запись
0	0	000	0	0000
1	1	001	1	0001
2	2	010	2	0010
3	3	011	3	0011
4	4	100	4	0100
5	5	101	5	0101
6	6	110	6	0110
7	7	111	7	0111
8	10	000	8	1000
9	11	001	9	1001
10	12	010	A	1010
11	13	011	B	1011
12	14	100	C	1100
13	15	101	D	1101
14	16	110	E	1110
15	17	111	F	1111

Переведем наше число в восьмеричную и шестнадцатеричную системы:

1100001111010110

1 100 001 111 010 110

1 4 1 7 2 6

1100 0011 1101 0110

C 3 D 6

Аналогично осуществляется и обратное преобразование: для этого каждую цифру восьмеричного или шестнадцатеричного числа заменяют группой из трех или четырех цифр. Например:

A B 5 1
1010 1011 0101 0001

1 7 7 2 0 4
1 111 111 010 000 100

Порядок выполнения

1. Произвести перевод чисел из одной системы счисления в другую в ручную в соответствии с вариантом задания, предложенным преподавателем.

2. Проверить правильность решения, используя специализированный калькулятор.

3. Оформить отчет по занятию.

Вариант				
1	2	3	4	5
$10101_2 \rightarrow X_{10}$	$11101_2 \rightarrow X_{10}$	$10111_2 \rightarrow X_{10}$	$10001_2 \rightarrow X_{10}$	$11111_2 \rightarrow X_{10}$
$13311_4 \rightarrow X_{10}$	$12112_4 \rightarrow X_{10}$	$11313_4 \rightarrow X_{10}$	$13212_4 \rightarrow X_{10}$	$32312_4 \rightarrow X_{10}$
$239_{10} \rightarrow X_5$	$487_{10} \rightarrow X_5$	$129_{10} \rightarrow X_5$	$912_{10} \rightarrow X_5$	$591_{10} \rightarrow X_5$
$19_{10} \rightarrow X_2$	$78_{10} \rightarrow X_2$	$54_{10} \rightarrow X_2$	$39_{10} \rightarrow X_2$	$48_{10} \rightarrow X_2$
$110101010_2 \rightarrow X_8$	$100101011_2 \rightarrow X_8$	$111101000_2 \rightarrow X_8$	$100111011_2 \rightarrow X_8$	$111101011_2 \rightarrow X_8$
$1011101.001_8 \rightarrow X_{10}$	$1111101.011_8 \rightarrow X_{10}$	$1001101.101_8 \rightarrow X_{10}$	$1111111.011_8 \rightarrow X_{10}$	$1000101.011_8 \rightarrow X_{10}$

Содержание отчета

1. Тема, цель занятия.
2. Решенные варианты заданий.
3. Выводы.

Контрольные вопросы

1. Разновидности систем счисления.
2. Системы счисления, используемые в цифровой технике.
3. Правила действий над числами в различных системах счисления.
4. Назовите числа от 1 до 9 в различных системах счисления.

Практическое занятие №2

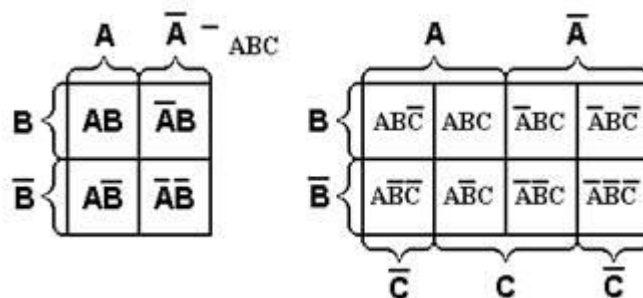
Минимизация функции

Цель занятия: изучить методы минимизации логических функций

Краткие теоретические сведения

Переключательные функции являются математическими моделями некоторых реальных электронных цифровых схем. Для того чтобы синтезировать наиболее оптимальное цифровое устройство (например, имеющее минимальные аппаратные затраты), математическую модель этого устройства, представленную в виде СДНФ или СКНФ, преобразуют к соответствующему виду с использованием приведенных выше законов алгебры логики. Преобразование СДНФ или СКНФ логической функции к минимальному виду аналитической записи называется процессом минимизации ФАЛ. Однако результат минимизации функции с помощью законов и тождественных преобразований булевой алгебры во многом зависит от опыта разработчика и не всегда является очевидным. Поэтому разработан ряд алгоритмов, формализующих и автоматизирующих подобные преобразования.

Рассмотрим один из алгоритмов минимизации, предложенный американский ученым Вейчем. Вейч предложил специальные диаграммы-карты, в которые можно записать все конstituенты единицы, входящие в СДНФ (конstituенты нуля, входящие в СКНФ) той или иной булевой функции. На рисунке 1 в качестве примера приведены диаграммы для минимизаций функций двух, трех и четырех переменных соответственно.



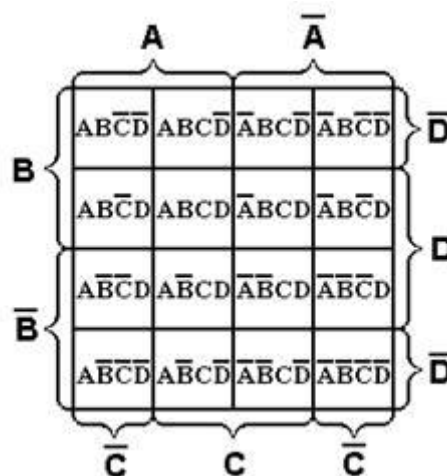


Рисунок 1 - Диаграммы Вейча для функций 2-х, 3-х и 4-х переменных.

Каждой клетке диаграммы в случае минимизации СДНФ соответствует определенная конституента единицы. Метод минимизации с помощью диаграмм Вейча заключается в следующем. Конституенты единицы, входящие в СДНФ булевой функции, заносятся в соответствующие клетки диаграммы. Удобно наличие соответствующей конституенты единицы изображать в клетке диаграммы цифрой 1, а отсутствие - 0. Все диаграммы построены таким образом, что рядом расположенные единицы по горизонтали или вертикали склеиваются между собой в соответствии с законом склеивания алгебры логики. Одну и ту же конституенту единицы можно использовать для склеивания с несколькими другими конституентами единицы с целью получения наиболее простого окончательного выражения. Цель всех операций - получить как можно меньшее число прямоугольников (в том числе квадратов), чтобы число членов СДНФ уменьшилось, получив в итоге МДНФ.

Формировать прямоугольники можно только при включении в них хотя бы одного нового члена. Склеивание можно осуществлять и путем замыкания крайних ребер в «бочку». Таким образом, полученная диаграмма Вейча геометрически образует цилиндр. На рисунке 2 приведены некоторые правила склеивания конституент единицы для функций 2-х и 3-х переменных.

Поскольку для ФАЛ трех переменных диаграмма представляет как бы развертку цилиндра, разрезанного по $\bar{C}$, поэтому единицы, расположенные по

краям диаграммы (например, как это изображено на рисунке 2) считаются расположенными рядом.

Метод минимизации СДНФ с помощью диаграмм Вейча включает в себя следующие шаги:

1. производится занесение в соответствующую диаграмму конституент единицы, входящих в СДНФ минимизируемой функции;
2. Используя приведенные выше правила склеивания, находят простые импликанты функций (простой импликантой называется некоторая конъюнкция, полученная в результате склеивания конституент единицы, не участвующая в склеивании ни с одной другой из конъюнкций);
3. Находится искомая минимальная дизъюнктивная нормальная форма (МДНФ) ФАЛ выбором минимальной совокупности простых импликант, покрывающей все конституенты единицы диаграммы.

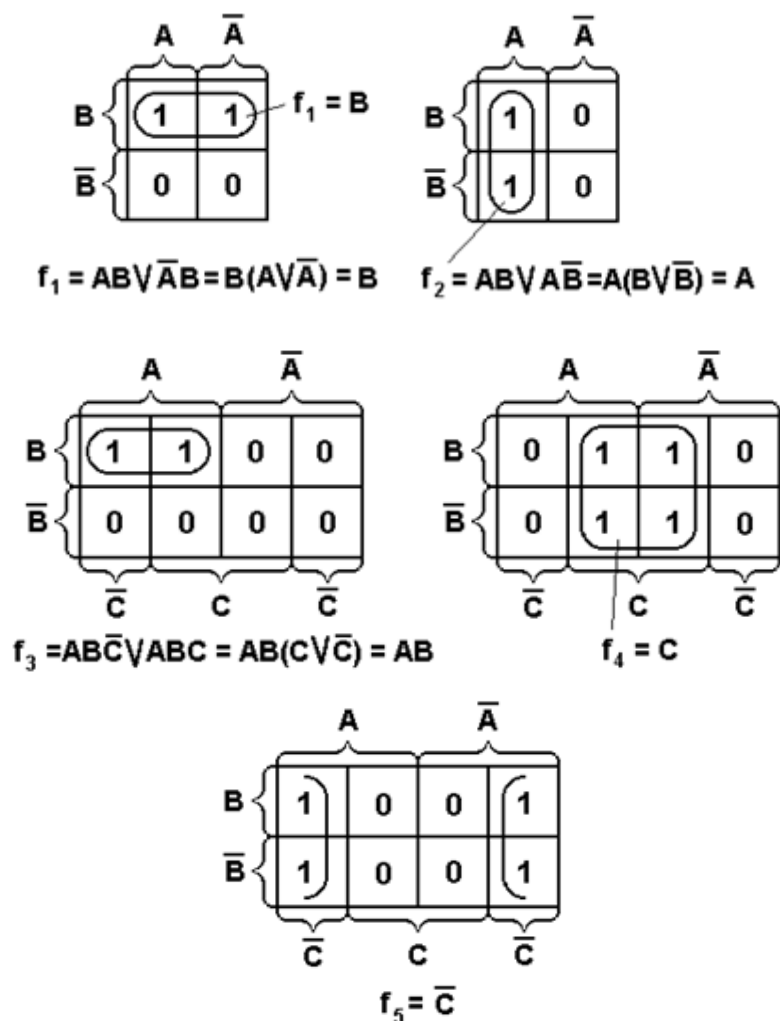


Рисунок 2 - Примеры для иллюстрации правил склеивания

В качестве примера найдем МДНФ функции, заданной таблицей (рисунок 2). Диаграмма Вейча этой функции представлена на рисунке 3. Из рисунка видно, что в результате склеивания образовались две простые импликанты: BC и $\overline{A}\overline{C}$. Импликанта $\overline{A}B$ участвует в склеивании с конъюнкциями BC и $\overline{A}\overline{C}$ и поэтому не является простой. Таким образом, полученная МДНФ имеет вид $f = BC \vee \overline{A}\overline{C}$. В истинности полученного выражения можно убедиться путем подстановки всех наборов переменных A, B и C.

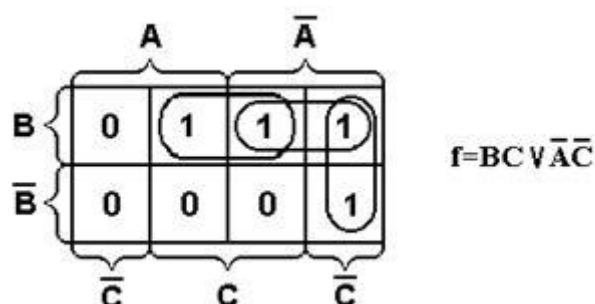


Рисунок 3 - Пример минимизации функции, заданной таблицей (рисунок 2)

Аналогом диаграмм Вейча являются карты Карно. Они позволяют изображать на плоскости прямоугольника конститuenty единицы (нуля) для ФАЛ более четырех переменных. В отличие от диаграмм Вейча, в которых отдельным строкам и столбцам соответствуют отдельные переменные, в картах Карно им можно присваивать значения нескольких переменных. При этом должны быть представлены все возможные комбинации этих переменных, например: AB , $A\overline{B}$, $\overline{A}B$ и $\overline{A}\overline{B}$. Таким образом, общее количество переменных минимизируемой с помощью карты Карно ФАЛ может быть больше, чем в случае использования диаграмм Вейча. Сам процесс минимизации аналогичен описанному на примере диаграмм Вейча.

Минимизация ФАЛ методом диаграмм Вейча и карт Карно в силу своей наглядности широко используется при ручном процессе минимизации логических функций от небольшого количества переменных, обычно не превышающего пяти. В случае, когда количество переменных больше, необходимо ис-

пользовать методы, характеризующиеся однозначностью алгоритма, что позволяет автоматизировать процесс минимизации на ЭВМ. Одним из таких методов является метод Квайна и Мак-Класки. Суть метода состоит в следующем:

1. Все конституенты единицы, записанные в виде двоичных кодов, разбиваются на группы, содержащие одинаковое количество единиц;
2. Склеивание производят между конституентами единицы, расположенными только в соседних группах, т.е. в соседних кубах кубического комплекса функции f ;
3. Все склеенные конституенты, образующие импликанты, отмечают каким-либо признаком, например звездочкой;
4. Не отмеченные при склеивании импликанты являются простыми и заносятся в импликантную матрицу, в первой строке которой записываются все конституенты единицы функции f , а в первом столбце – все простые импликанты;
5. В пересечении строк и столбцов импликантной матрицы проставляется звездочка (или другой выделяющий символ) в тех местах, где соответствующая импликанта покрывает конституенту единицы;
6. Ищутся столбцы импликантной матрицы, имеющие только по одной звездочке;
7. Соответствующие этим звездочкам простые импликанты называются базисными и образуют ядро функции f и обязательно входят в МДНФ;
8. Рассматриваются столбцы импликантной матрицы и выделяются те, которые содержат более одной звездочки;
9. Если среди выделенных столбцов существуют такие, которые покрываются базисными импликантами из ядра функции, то они считаются уже учтенными в записи МДНФ, если же выделенные столбцы базисными импликантами не покрываются, то в МДНФ записываются импликанты соответствующих столбцов с минимальным количеством переменных.

Рассмотрим пример минимизации функции f , заданной таблицей 1.3 методом Квайна и Мак-Класки. Функция принимает единичные значения при

следующих значениях переменных А, В и С - 000, 010, 011, 111, которые и образуют конституенты единицы. Эти конституенты можно записать в четыре группы по количеству в них единиц:

группа 0: 000 *;

группа 1: 010 *;

группа 2: 011 *;

группа 3: 111 *.

Осуществим склеивание конституент из соседних групп. Те конституенты, которые удастся склеить, пометим звездочкой. Результатом склеивания являются импликанты, которые аналогично записываются в группы по количеству в них единиц. Вместо несовпадающих разрядов в записи импликант проставляются прочерки:

группа 1: 0-0;

группа 2: 01-;

группа 3: -11.

Дальнейшее склеивание соседних импликант возможно только при наличии прочерка в одинаковых позициях. В нашем примере это условие не выполняется и, поэтому, образованные импликанты являются простыми. Из полученных простых импликант составляется импликантная матрица, в первой строке которой записываются все конституенты единицы, а в первом столбце – все простые импликанты. Звездочкой отмечаются все пересечения строк и столбцов, в которых импликанты покрывают конституенты единицы.

Простые импликанты	Конституенты единицы			
	000	010	011	111
0-0	*	*		
01-		*	*	
-11			*	*

В матрице находим столбцы, содержащие по одной звездочке. Это столбцы с конституентами 000 и 111. Одиночным звездочкам в них соответствуют импликанты 0-0 и -11. Эти импликанты являются базисными, образуют ядро

логической функции и обязательно входят в состав МДНФ. Далее выделяются столбцы, содержащие более одной звездочки. Такими столбцами являются столбцы с контигуентами 010 и 011. Поскольку эти контигуенты покрываются импликантами 0-0 и -11 из ядра функции, то импликанта 01- является лишней и в МДНФ не входит. В результате имеем в составе МДНФ импликанты 0-0 и -11. Заменяя двоичные коды полученных импликант на обозначения переменных в прямом и инверсном виде, получим окончательную запись МДНФ функции: $f = \overline{A}\overline{C} \vee BC$, что соответствует результату минимизации методом диаграмм Вейча.

Рассмотрим пример минимизации ФАЛ четырех переменных методом Квайна и Мак-Класки. Пусть задана функция в виде последовательности десятичных чисел:

$$f(A,B,C,D) = \vee(1,2,3,5,6,7,10,11,14,15).$$

Запишем контигуенты единицы функции f в виде двоичных кодов:

$$f = 0001 \vee 0010 \vee 0011 \vee 0101 \vee 0110 \vee 0111 \vee 1010 \vee 1011 \vee 1110 \vee 1111.$$

Разобьем все контигуенты единицы по группам из количества единиц.

группа 0: ;

группа 1: 0001 **, 0010 ***;

группа 2: 0011 ****, 0101 **, 0110 **, 1010 ***;

группа 3: 0111 ****, 1011 ***, 1110 **;

группа 4: 1111 ***.

Произведем склеивание контигуент из соседних групп, помечая склеиваемые контигуенты звездочками. Полученные импликанты разобьем по группам:

группа 1: 00-1 *, 0-01 *, 001- **, 0-10 **, -010*;

группа 2: 0-11 ***, -011 **, 01-1 *, 011- **, 101- *, 1-10 **;

группа 3: -111 *, 1-11 **, 111- *.

Продолжим дальнейшее склеивание импликант из соседних групп, причем склеивание возможно только между импликантами с прочерками в одинаковых позициях:

группа 1: 0--1, 0--1, 0-1-, -01-, 0-1-, --10, -01-;

группа 2: --11, --11, -11-, 1-1-, 1-1-.

Перепишем, оставив только неповторяющиеся импликанты:

группа 1: 0--1, 0-1- *, -01- *, --10 *;

группа 2: --11 *, -11- *, 1-1- *.

Продолжим склеивание:

Группа 1: --1-, --1-, --1-.

Не отмеченные звездочками импликанты являются простыми: 0--1 и --1-.

Построим импликантную матрицу:

Простые импликанты	Контституенты единицы									
	0001	0010	0011	0101	0110	0111	1010	1011	1110	1111
0--1	*		*	*		*				
--1-		*	*		*	*	*	*	*	*

Из матрицы видно, что столбцы, содержащие по одной звездочке соответствуют обоим простым импликантам и, следовательно, обе эти импликанты являются базисными, образуют ядро ФАЛ и входят в МДНФ. Заменяя двоичные коды импликант соответствующими переменными, запишем вид МДНФ: $f = \overline{A}D \vee C$. В истинности полученного выражения можно убедиться подстановкой всех возможных значений входных переменных А, В, С, и D.

При минимизации частично определенной ФАЛ недостающие наборы выбираются произвольно исходя из соображения получения наименьшего количества простых импликант, по возможности содержащих минимальное количество переменных. На диаграмме Вейча такие импликанты будут определяться максимально большими прямоугольниками. Поэтому при минимизации частично определенной ФАЛ в диаграмму Вейча заносят все определенные значения функции в виде соответствующих нулей и единиц. В неопределенных ячейках дописываются нули и единицы до образования максимально склеиваемых областей, соответствующих простым импликантам с наименьшим количе-

ством переменных. Например, если для заданной функции трех переменных заполненная диаграмма Вейча имеет вид (прочерками указываются неопределенные значения), представленный на рисунке 4,а, то после дополнения единиц в позициях $\bar{A}BC$ и $A\bar{B}C$, диаграмма Вейча примет вид, представленный на рисунке 4,б. В результате склеивания МДНФ запишется $f = C$.

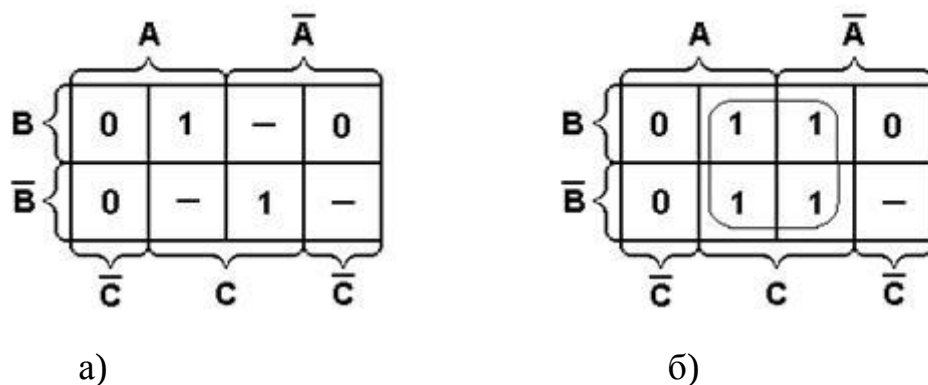


Рисунок 4 - Пример минимизации частично определенной ФАЛ

Порядок выполнения

1. Произвести перевод чисел из одной системы счисления в другую в ручную в соответствии с вариантом задания, предложенным преподавателем.
2. Проверить правильность решения, используя специализированный калькулятор.
3. Оформить отчет по занятию.

Содержание отчета

1. Тема, цель занятия.
2. Решенные варианты заданий.
3. Выводы.

Контрольные вопросы

1. Дайте определение системы счисления и приведите примеры используемых в цифровой технике систем счисления.

2. Отнимите из числа 57_{10} число 83_{10} и из числа 83_{10} число 57_{10} по правилам двоичной арифметики, используя дополнительные коды и операцию сложения.
3. Дайте определение функции алгебры-логики. Приведите различные формы записи для произвольной логической функции трех переменных.
4. Выполните минимизацию логической функции из предыдущего вопроса, используя диаграммы Вейча и метод Квайна и Мак-Класки.
5. Изобразите кубические комплексы произвольной логической функции двух переменных геометрической фигурой.

Практическое занятие №3

Моделирование логических элементов

Тема, цель занятия: получит практические навыки по моделированию имитаторов логических элементов.

Краткие теоретические сведения

Известно, что математической основой цифровых вычислительных устройств является двоичная арифметика, в которой используются всего два числа — 0 и 1. Выбор двоичной системы счисления диктовался требованиями простоты технической реализации самых сложных задач с использованием всего одного базового элемента — ключа, который имеет два состояния: включен (замкнут) или выключен (разомкнут). Если первое состояние ключа принять за условную (логическую) единицу, то второе будет отражать условный (логический) ноль или наоборот. Возможные комбинации показаны на рисунках 1, 2 и 3.

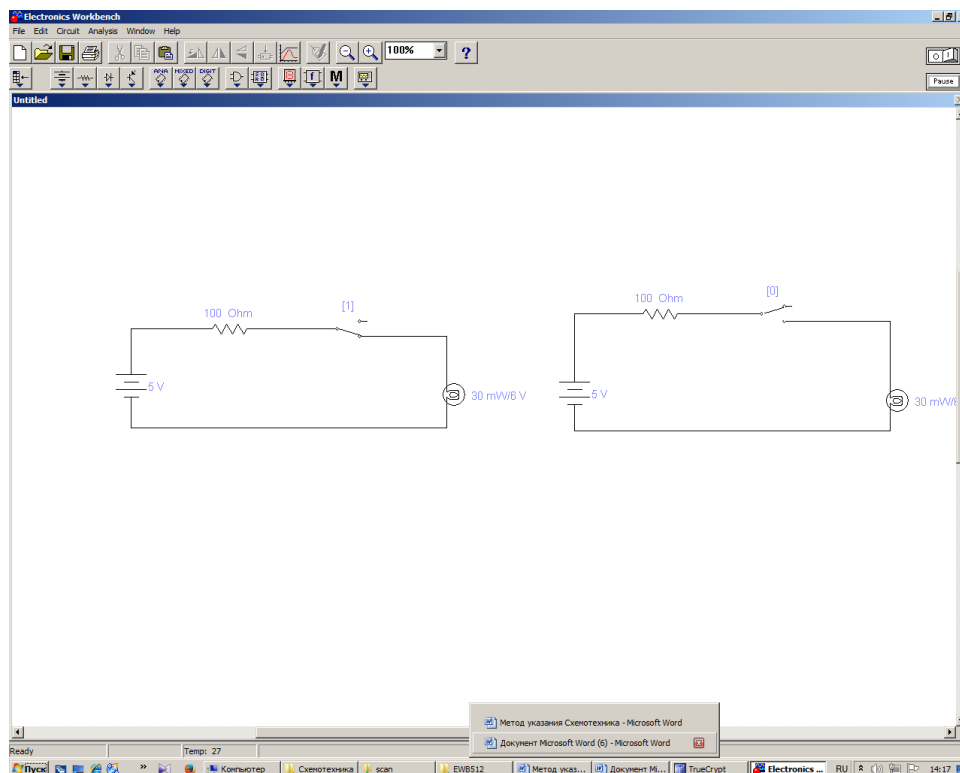


Рисунок 1 - Схема электромеханических имитаторов логической единицы (а) и нуля (б)

На рисунке 1 показаны ключи 1 и 0, управляемые клавишами 1 и 0 соответственно, и вспомогательные устройства в виде батареи 5 В с внутренним сопротивлением 100 Ом и лампа накаливания на 6 В с мощностью 30 мВт, которые позволяют судить о состоянии ключа: если он находится в положении 1, лампа горит (рисунок 1, а), или не горит, если он находится в положении 0 (рисунок 1, б).

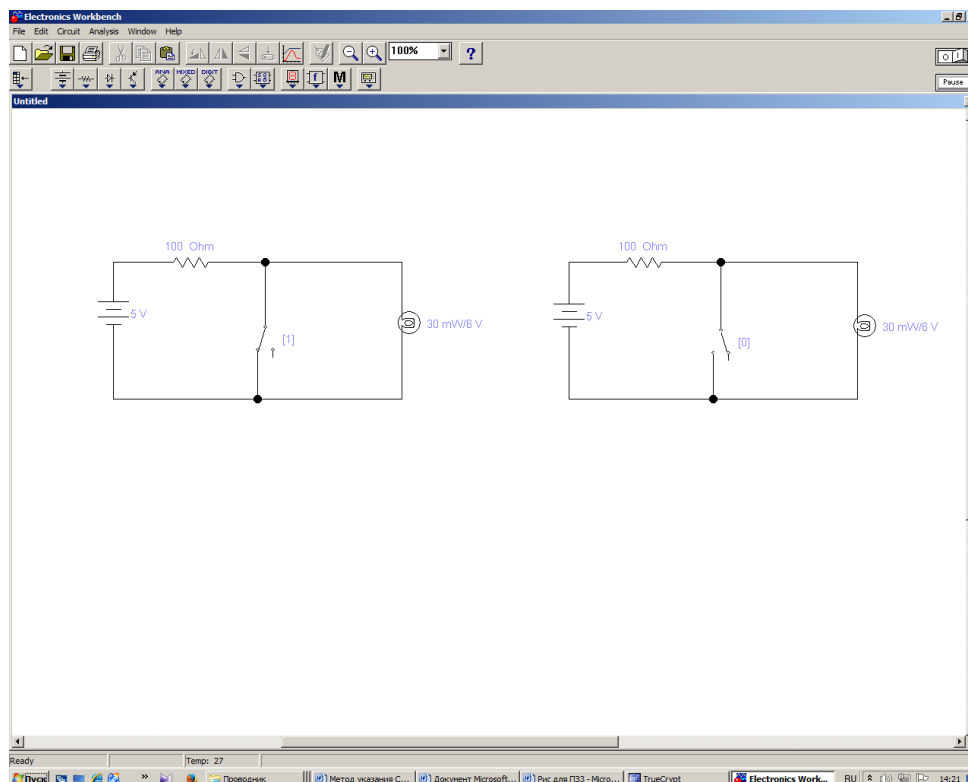


Рисунок 2 - Электромеханические имитаторы логической единицы(а) и нуля (б) в инверсном режиме

Возможно другое расположение ключей по отношению к вспомогательным устройствам, показанное на рисунке 2. В этих схемах состояние индикаторов нуля или единицы противоположно показанному на рисунке 1. При нажатии на клавишу 1 индикатор фиксирует состояние 0 (рисунок 2, а) и наоборот (рисунок 2, б). Следовательно, схемы на рисунке 2 по выходному сигналу (состоянию индикаторных лампочек) обратны (инверсны) по отношению к схемам на рисунке 1. Поэтому такие ключи называют инверторами.

Поскольку в цифровых системах содержится огромное количество ключей (только в одном микропроцессоре их несколько миллионов) и они не могут сообщать друг другу о своем состоянии миганием лампочек, то для взаимного обмена информацией используются электрические сигналы напряжения. При этом ключи, как правило, применяются в инверсном режиме в соответствии со схемами на рисунке 3.

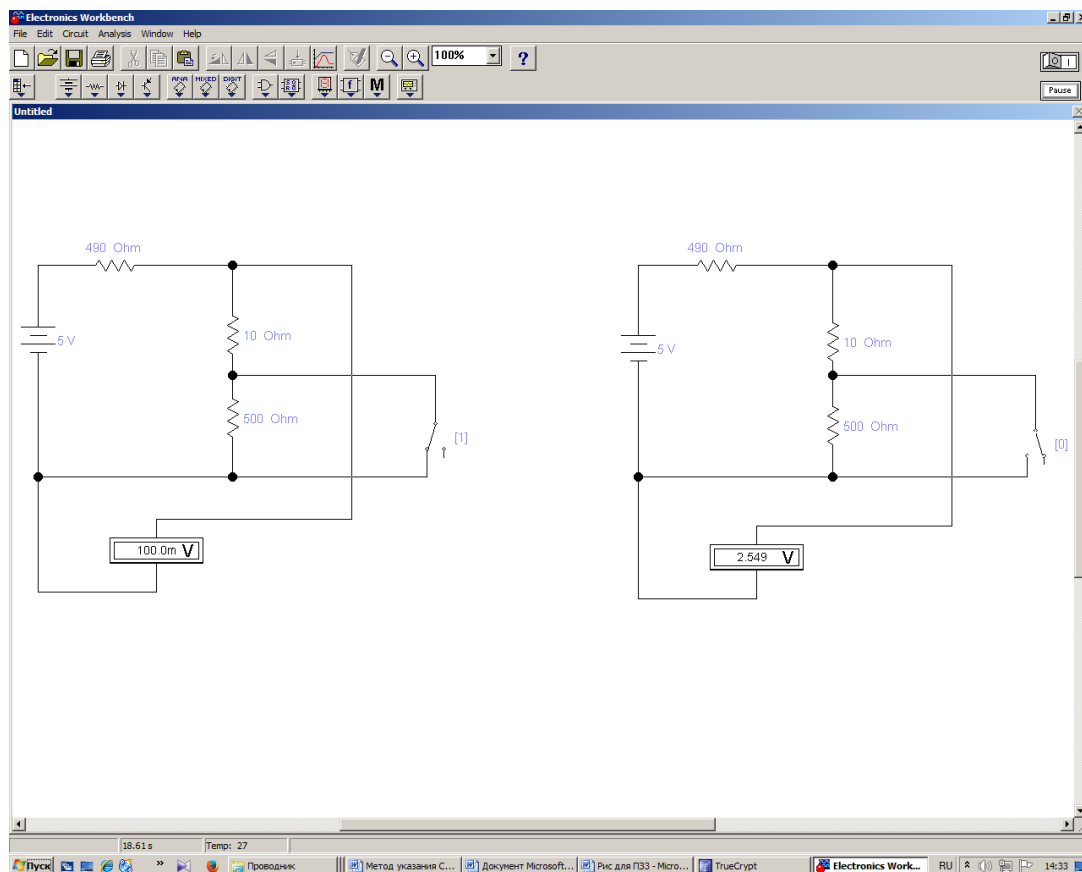


Рисунок 3 - Электромеханические имитаторы логической единицы (а) и нуля (б) в инверсном режиме с индикаторами выходного напряжения

На рисунке 3 сопротивление 490 Ом имитирует внутреннее сопротивление нагрузки ключа (аналог коллекторного сопротивления в транзисторном ключе), сопротивление 10 Ом — сопротивление замкнутого электронного ключа, сопротивление 500 Ом — сопротивление разомкнутого ключа с учетом внешней нагрузки. Как видно из рисунка 3, наличие на выходе логического нуля (инверсия 1) индицируется напряжением 100 мВ (в практических конструкциях может быть и больше), а наличие логической единицы — напряжением 2,55 В (нормируется на уровне 2,4 В). Электронные ключи проектируются таким образом, чтобы при наихудших сочетаниях входных и выходных параметров ключи могли различать сигналы логической единицы и нуля.

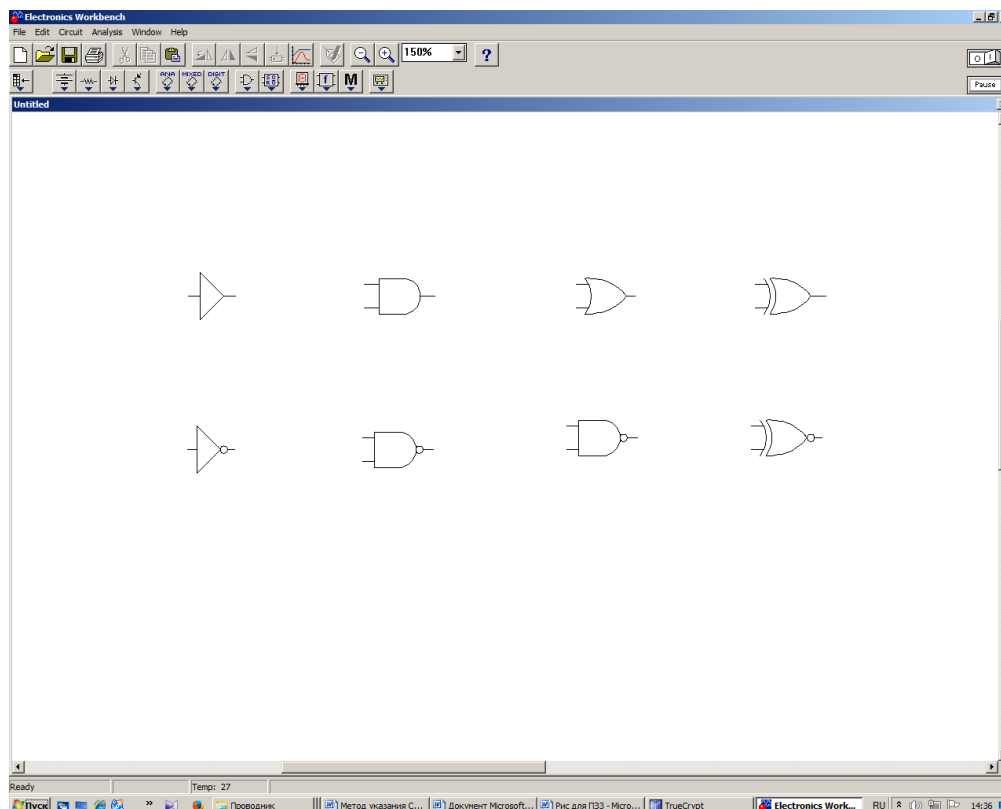


Рисунок 4 - Графические обозначения буферного логического элемента (а), элементов И (AND) (б), ИЛИ (OR) (в), Исключающее ИЛИ (XOR) (г) и их инверсные варианты во втором ряду (NOT, NAND, NOR, XNOR соответственно)

В цифровой технике практические аналоги рассмотренных схем принято называть логическими элементами. При этом в зависимости от выполняемых функций каждый элемент имеет свое название и соответствующее графическое обозначение. На рисунке 4 показаны обозначения базовых логических элементов, принятые в программе EWB 4.1.

Электромеханическим аналогом буферного элемента являются имитаторы на рисунке 3, а логического элемента НЕ (NOT) — на рисунках 2 и 3. Электромеханические аналоги двухвходовых элементов И, И-НЕ показаны на рисунке 5.

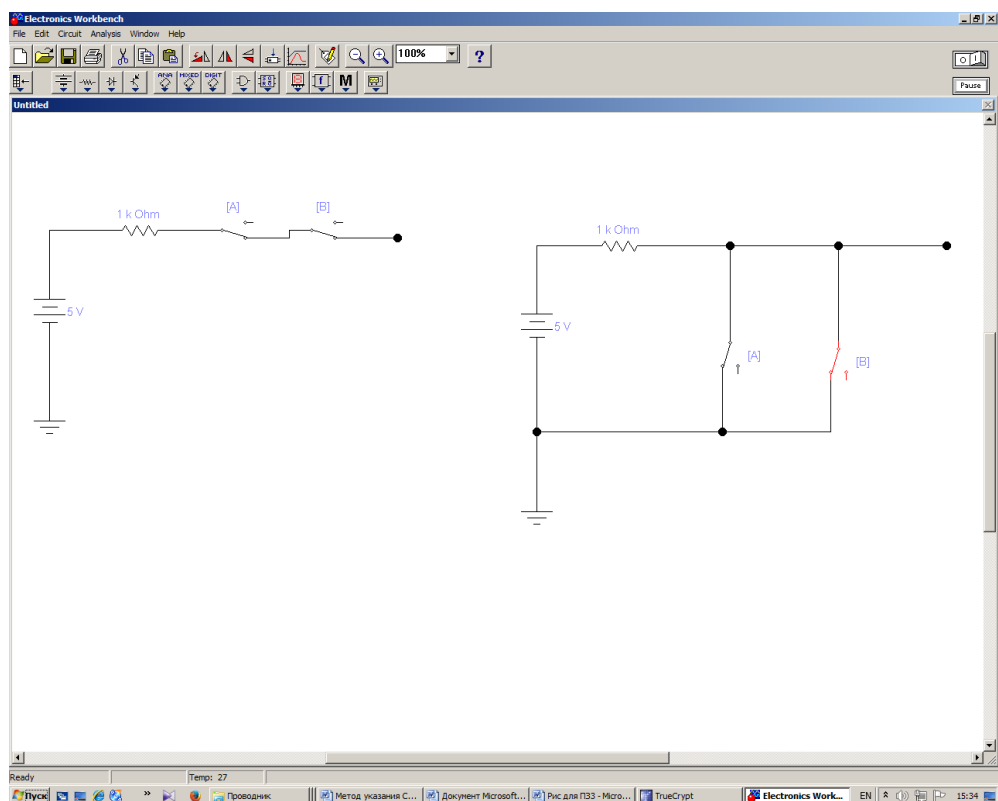


Рисунок 5 - Электромеханические имитаторы двухвходовых элементов

При наличии в программе EWB такого замечательного инструмента, как логический преобразователь, исследования логических схем целесообразно проводить с его помощью. В качестве примера на рисунке 6 приведена схема для исследования элемента Иключающее ИЛИ.

Подключение исследуемого элемента к логическому преобразователю очевидно из рисунка 6. Очевидно также и то, что при наличии двух входов возможны только четыре комбинации входных сигналов, что отображается на экране преобразователя в виде таблицы истинности, которая генерируется после нажатия верхней клавиши.

Для получения булева выражения исследуемого элемента необходимо нажать вторую клавишу.

Это выражение приводится на дополнительном дисплее, расположенном в нижней части лицевой панели, в виде двух слагаемых, соответствующих выходному сигналу ИСТИНА (сигнал логической единицы на выходе OUT). Сопоставление полученного выражения с таблицей истинности убеждает нас в

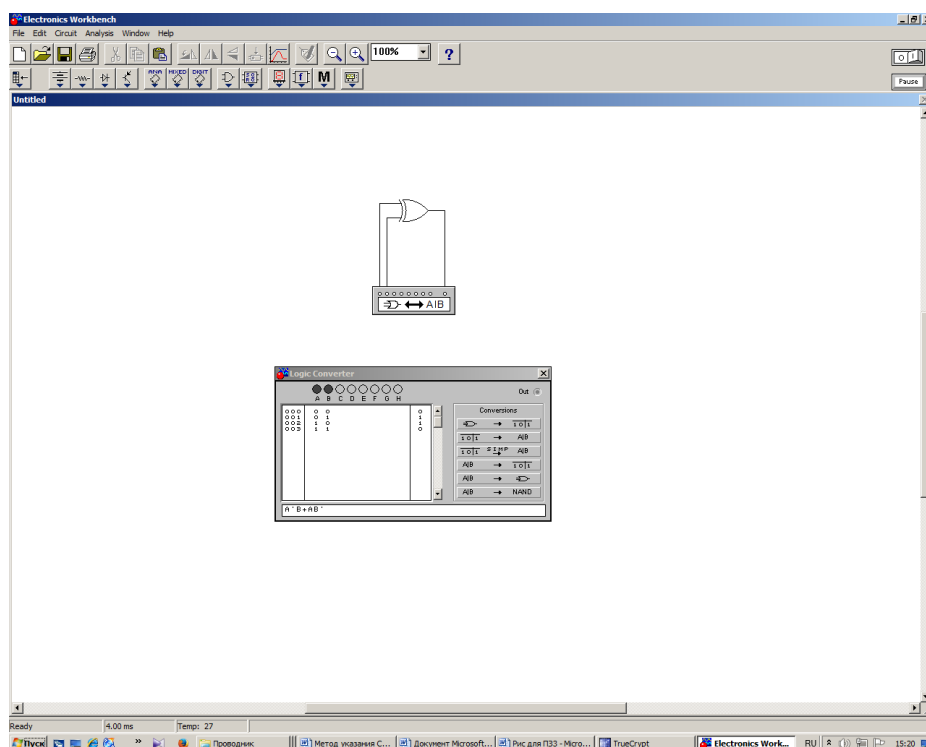


Рисунок 6 - Исследование логического элемента Иключающее ИЛИ
с помощью логического преобразователя

С помощью логического преобразователя можно проводить не только анализ логических устройств, но и их синтез. Допустим, что нам требуется составить схему и булево выражение для логического элемента, у которого выходная комбинация в таблице истинности не 0110, как на рисунке 6, а 1101. Для внесения необходимых изменений отмечаем курсором в столбце OUT подлежащий изменению символ, изменяем его с помощью клавиатуры и затем, перемещаясь по столбцу клавишами управления курсором, изменяем по необходимости символы в других строках. После внесения всех изменений последовательно нажимаем на требуемые клавиши и получаем результат, представленный на рисунке 7. Синтезированное логическое устройство показано в верхнем левом углу рисунка 7, а его булево выражение — на дополнительном дисплее.

В более общем случае для выполнения синтеза целесообразно действовать следующим образом. Щелчком курсора по иконке логического преобразователя непосредственно на линейке приборов раскрываем его лицевую панель. Активизируем курсором клеммы-кнопки А, В, ..., Н (начиная с А), количество которых равно количеству входов синтезируемого устройства. Вносим необходимые изменения в столбец OUT и после нажатия на панели преобразователя указанных выше клавиш управления получаем результат в виде схемы на рабочем поле программы и булево выражение в дополнительном дисплее.

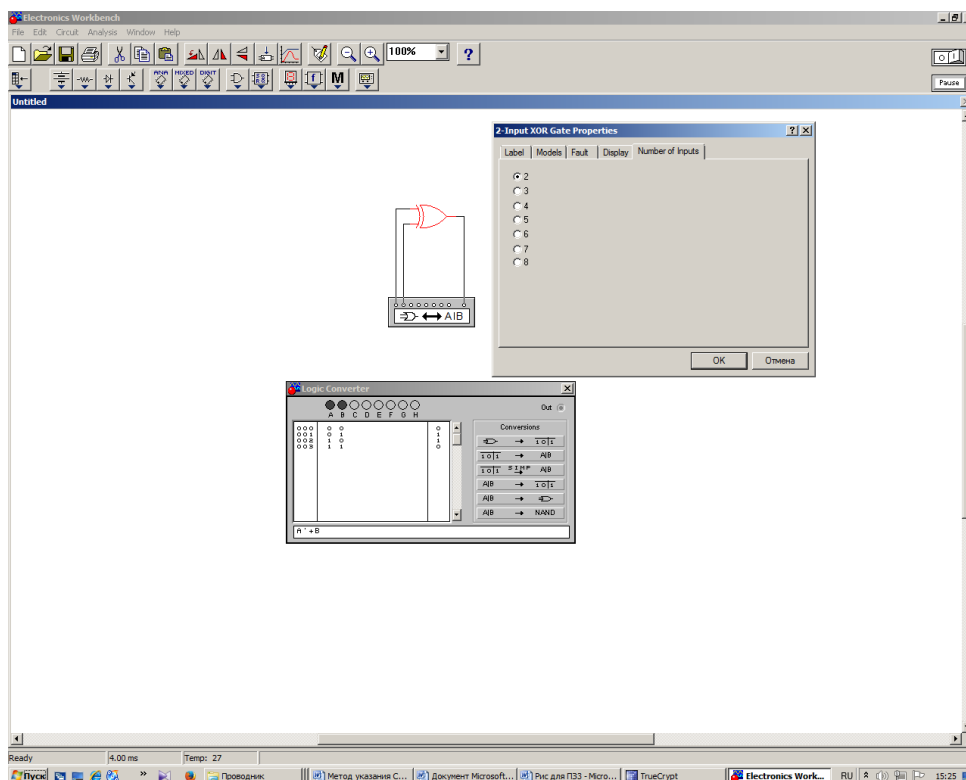


Рисунок 7 - Результат синтеза логического устройства по заданной таблице истинности

В заключение заметим, что для двухвходовых элементов на рисунке 4 можно увеличить количество входов до восьми, открывая двойным щелчком по значку компонента диалоговое окно (рисунок 7). По умолчанию в этом окне указано минимально возможное число входов, равное двум.

Порядок выполнения

1. Провести моделирование электромеханических имитаторов логической единицы и нуля (рисунок 1).
2. Провести моделирование электромеханических имитаторов логической единицы и нуля в инверсном режиме (рисунок 1).
3. Провести моделирование электромеханических имитаторов логической единицы и нуля в инверсном режиме с индикаторами выходного напряжения (рисунок 3).
4. Провести моделирование электромеханических имитаторов двухвходовых элементов (рисунок 5).
5. Исследовать логический элемент Искключающее ИЛИ с помощью логического преобразователя (рисунок 6).
6. Синтезировать логическое устройство по заданной таблице истинности (рисунок 7).
7. Оформить отчет о занятии.

Содержание отчета

1. Тема, цель занятия.
2. Схемы экспериментов.
3. Выводы.

Контрольные вопросы и задания

1. Известно, что единицей измерения информации является бит. Какие значения может принимать эта единица?
2. Проведите моделирование оставшихся без рассмотрения двухвходовых логических элементов на рисунке 4 с использованием логического преобразователя и установите для каждого из них соответствие таблицы истинности и булева выражения.

3. Разработайте схемы электромеханических имитаторов двухвходовых логических элементов на рисунке 4 (за исключением элемента И).

4. Проведите синтез трехвходового логического устройства с выходной комбинацией 10011110 таблицы истинности.

5. Установите различия в булевых выражениях и графических обозначениях логических элементов программы EWB от принятых в отечественной научно-технической литературе.

Практическое занятие №4

Исследование работы логических элементов

Цель занятия: исследование принципов функционирования логических элементов, моделирование логических функций.

Краткие теоретические сведения

Функционирование логических элементов подчинено законам алгебры логики.

Переменные, рассматриваемые в алгебре логики, могут принимать только два значения — 0 или 1. В алгебре логики определены: отношение эквивалентности (обозначается знаком $=$) и операции: сложения (дизъюнкции), обозначаемая знаком $\vee$, умножения (конъюнкции), обозначаемая знаком $\&$ или точкой, и отрицания (или инверсии), обозначаемая надчеркиванием или апострофом. Алгебра логики определяется следующей системой аксиом:

$$\begin{cases} x = 0, \text{ если } x \neq 1, \\ x = 1, \text{ если } x \neq 0. \end{cases} \begin{cases} \bar{0} = 1, \\ \bar{1} = 0. \end{cases}$$

$$\begin{cases} x \vee 1 = 1, \\ 0 \vee 0 = 0, \\ 0 \vee 1 = 1 \vee 0 = 1. \end{cases} \begin{cases} 0 \cdot 0 = 0, \\ 1 \cdot 1 = 1, \\ 1 \cdot 0 = 0 \cdot 1 = 0. \end{cases}$$

Запись логических выражений обычно осуществляют в конъюнктивной или дизъюнктивной нормальных формах. В дизъюнктивной форме логические выражения записываются как логическая сумма логических произведений, в конъюнктивной форме — как логическое произведение логических сумм. Порядок действий такой же, как и в обычных алгебраических выражениях. Логические выражения связывают значение логической функции со значениями логических переменных.

При преобразованиях логических выражений используются логические тождества:

$$\begin{aligned}\bar{x} &= x; & x \vee 1 &= 1; & x \cdot 1 &= x; & x \cdot 0 &= 0; \\ x \vee x &= x; & x \cdot x &= x; & x \vee x \cdot y &= x; \\ xy \vee x\bar{y} &= x; & x \vee y & \quad x \vee \bar{y} &= x; & x\bar{y} &= \bar{x} \vee \bar{y}, & \bar{x} \vee \bar{y} &= \overline{xy}.\end{aligned}$$

Любое логическое выражение, составленное из n переменных $x_1, x_2, \dots, x_n$ с помощью конечного числа операций алгебры логики, можно рассматривать как некоторую функцию n переменных. Такую функцию называют логической. В соответствии с аксиомами алгебры логики функция может принимать в зависимости от значения переменных значение 0 или 1. Функция n логических переменных может быть определена для 2^n значений переменных, соответствующих всем возможным значениям n -разрядных двоичных чисел. Основным интерес представляют следующие функции двух переменных x и y :

- $f_1 \ x, y = x \cdot y$ — логическое умножение (конъюнкция);
- $f_2 \ x, y = x \vee y$ — логическое сложение (дизъюнкция);
- $f_3 \ x, y = \overline{x \cdot y}$ — логическое умножение с инверсией;
- $f_4 \ x, y = \overline{x \vee y}$ — логическое сложение с инверсией;
- $f_5 \ x, y = x \oplus y = x\bar{y} \vee \bar{x}y$ — суммирование по модулю 2;
- $f_6 \ x, y = \overline{x \oplus y} = xy \vee \bar{x}\bar{y}$ — равнозначность.

Физическое устройство, реализующее одну из операций алгебры логики или простейшую логическую функцию, называется логическим элементом. Схема, составленная из конечного числа логических элементов по определенным правилам, называется логической схемой. Основным логическим функциям соответствуют выполняющие их схемные элементы.

Так как область определения любой функции n переменных конечна (2^n значений), такая функция может быть задана таблицей значений $f V_i$, которые она принимает в точках V_i , где $i = 0, 1, \dots, 2^n - 1$. Такие таблицы называют таблицами истинности. В таблице 1 представлены таблицы истинности, задающие указанные выше функции.

Таблица 1

i	Значение переменных		Функции					
	x	y	F1	F2	F3	F4	F5	F6
0	0	0	0	0	1	1	0	1
1	0	1	0	1	1	0	1	0
2	1	0	0	1	1	0	1	0
3	1	1	1	1	0	0	0	1

$i = 2x + y$ — число, образованное значениями переменных.

Если число логических переменных не превышает 5 - 6, преобразования логических уравнений удобно производить с помощью карт Карно или диаграмм Вейча. Цель преобразований - получение компактного логического выражения (минимизация). Минимизацию производят объединением наборов (термов) на карте Карно. Объединяемые наборы должны иметь одинаковые значения функции (все 0 или все 1). Для наглядности рассмотрим пример: пусть требуется найти логическое выражение для мажоритарной функции f_m трех переменных X, Y, Z , описываемой таблицей истинности (таблица 1).

Таблица 2 - Мажоритарная функция

N	X	Y	Z	fm
0	0	0	0	0
1	0	0	1	0
2	0	1	0	0
3	0	1	1	1
4	1	0	0	0
5	1	0	1	1
6	1	1	0	1
7	1	1	1	1

Составим карту Карно. Она представляет собой нечто похожее на таблицу, в которой наименования столбцов и строк представляют собой значения переменных, причем переменные располагаются в таком порядке, чтобы при переходе к соседнему столбцу или строке изменялось значение только одной переменной. Например, в строке XY таблицы (рисунок 1) значения переменных XY могут быть представлены следующими последовательностями: 00, 01, 11, 10 и 00, 10, 11, 01. Таблицу заполняют значениями функции, соответствующими комбинациям значений переменных. Полученная таким образом таблица выглядит, как показано на рисунке 1.

Z \ XY	XY			
	00	01	11	10
0	0	0	1	0
1	0	1	1	1

← XY
↑ YZ
↑ XZ

Рисунок 1 - Карта Карно мажоритарной функции

На карте Карно отмечаем группы, состоящие из $2n$ ячеек (2, 4, 8,...) и содержащие 1, т. к. они описываются простыми логическими выражениями. Три прямоугольника в таблице определяют логические выражения XY , XZ , YZ . Каждый прямоугольник, объединяющий две ячейки, соответствует логическим преобразованиям:

$$XY\bar{Z} \vee XYZ = XY \quad \bar{Z} \vee Z = XY,$$

$$X\bar{Y}Z \vee XYZ = XZ \quad \bar{Y} \vee Y = XZ,$$

$$\bar{X}YZ \vee XYZ = YZ \quad X \vee \bar{X} = YZ.$$

Компактное выражение, описывающее функцию, представляет собой дизъюнкцию полученных при помощи карт Карно логических выражений. В результате получаем выражение в дизъюнктивной форме: .

Для реализации функции мажоритарной логики трех логических переменных необходимо реализовать схему, которая при подаче на ее входы трех сигналов формировала бы на выходе сигнал, равный сигналу на большинстве входов (2 из 3 или 3 из 3). Эта схема полезна для восстановления истинного значения сигналов, поступающих на 3 входа, если возможен отказ на одном из входов. Для реализации функции на элементах «2И-НЕ» преобразуем полученное выражение в базис элементов «И-НЕ», т. е. запишем выражение при помощи операций логического умножения и инверсии. Проверить справедливость каждого из приведенных выражений для f_m можно прямой подстановкой значений X , Y , Z из таблицы 2:

$$\begin{aligned} f_m &= XY \vee YZ \vee XZ = \overline{\overline{X}\overline{Y}} \cdot \overline{\overline{Y}\overline{Z}} \cdot \overline{\overline{X}\overline{Z}} = \overline{\overline{X} \vee \overline{Y}} \vee \overline{\overline{Y} \vee \overline{Z}} \vee \overline{\overline{X} \vee \overline{Z}} = \\ &= \overline{\overline{X}} \cdot \overline{\overline{Y}} \vee \overline{\overline{Y}} \cdot \overline{\overline{Z}} = \overline{\overline{X}} \cdot \overline{\overline{Y}} \cdot \overline{\overline{Z}} \cdot \overline{\overline{X}} \cdot \overline{\overline{Z}}. \end{aligned}$$

Соответствующая схемная реализация приведена на рисунке 2.

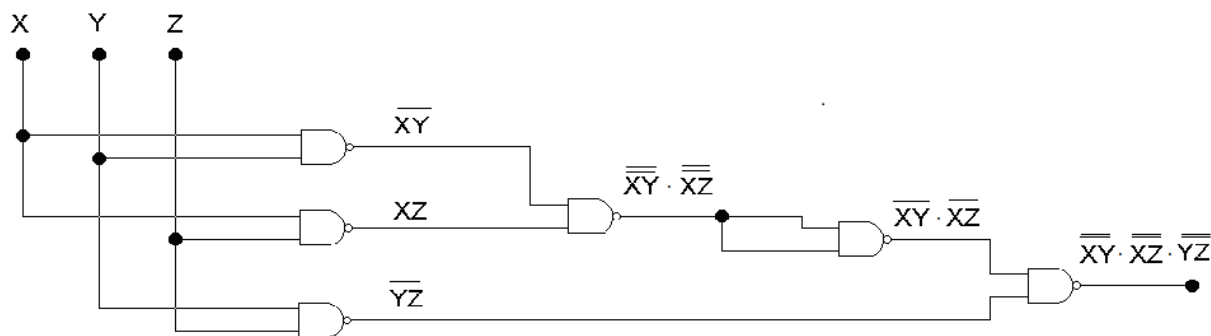


Рисунок 2

Разработанная виртуальная лабораторная установка позволяет выполнить 6 экспериментов:

1. Исследование логической функции «И»:
 - задание уровней логических сигналов;
 - экспериментальное получение таблицы истинности элемента «И»;
 - получение аналитического выражения для функции.
2. Исследование логической функции «И-НЕ»:
 - экспериментальное получение таблицы истинности логического элемента «2И-НЕ», составленного из элементов «2И» и «НЕ»;
 - экспериментальное получение таблицы истинности логического элемента «2И-НЕ».
3. Исследование логической функции «ИЛИ»:
 - экспериментальное получение таблицы истинности элемента «ИЛИ»;
 - получение аналитического выражения для функции.
4. Исследование логической функции «ИЛИ-НЕ»:
 - экспериментальное получение таблицы истинности логического элемента «2ИЛИ-НЕ», составленного из элементов «2ИЛИ» и «НЕ»;
 - экспериментальное получение таблицы истинности логического элемента «2ИЛИ-НЕ».
5. Исследование логических схем с помощью генератора слов:
 - сведения об исследуемой микросхеме;

— экспериментальное получение таблицы истинности логического элемента «2И-НЕ».

6. Реализация логической функции трех переменных:

— синтез схемы, реализующей функцию, заданную логическим выражением;

— синтез схемы, реализующей заданную функцию при помощи логического преобразователя.

Порядок выполнения

Эксперимент 1. Исследование логической функции «И».

а) Задание уровней логических сигналов.

Откройте файл ewb01_01 со схемой, изображенной на рисунке 3.

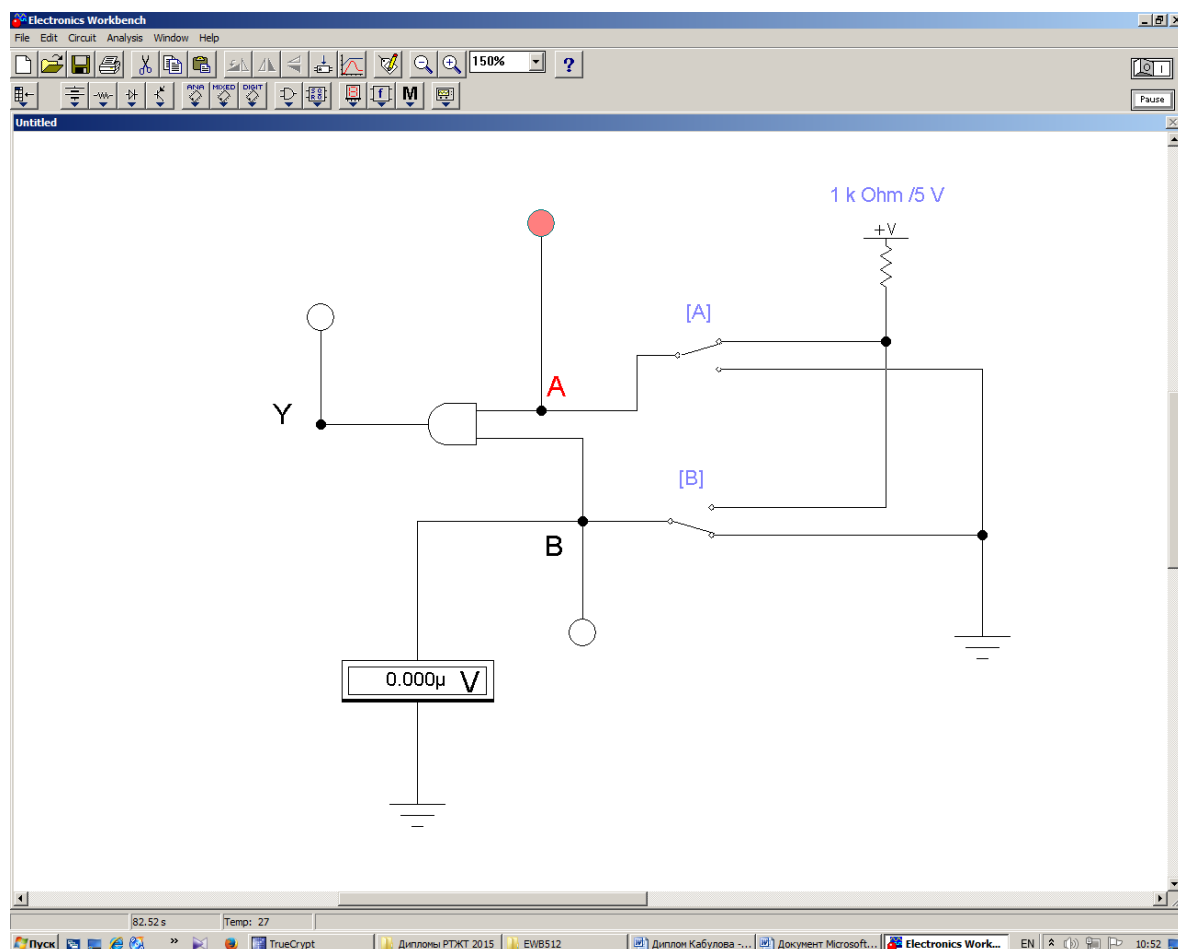


Рисунок 3

В этой схеме два двухпозиционных переключателя A и B подают на входы логической схемы «И» уровни 0 (контакт переключателя в нижнем положении) или 1 (контакт переключателя в верхнем положении). Включите схему. Установите переключатель B в нижнее положение. Измерьте вольтметром напряжение на входе B и определите с помощью логического пробника уровень логического сигнала.

Установите переключатель B в верхнее положение. Определите уровень логического сигнала и запишите показания вольтметра; укажите, какой логический сигнал формируется на выходе Y . Результаты занесите в раздел «Результаты экспериментов».

б) Экспериментальное получение таблицы истинности элемента «И».

Подайте на входы схемы все возможные комбинации уровней сигналов A и B и для каждой комбинации зафиксируйте уровень выходного сигнала Y .

Заполните таблицу истинности логической схемы «И» (таблица 3 в разделе «Результаты экспериментов»).

в) Получение аналитического выражения для функции.

По таблице 3 составьте аналитическое выражение функции элемента «И» и занесите его в раздел «Результаты экспериментов».

Эксперимент 2. Исследование логической функции «И-НЕ».

а) Экспериментальное получение таблицы истинности логического элемента «2И-НЕ», составленного из элементов «2И» и «НЕ».

Соберите схему, изображенную на рисунке 4.

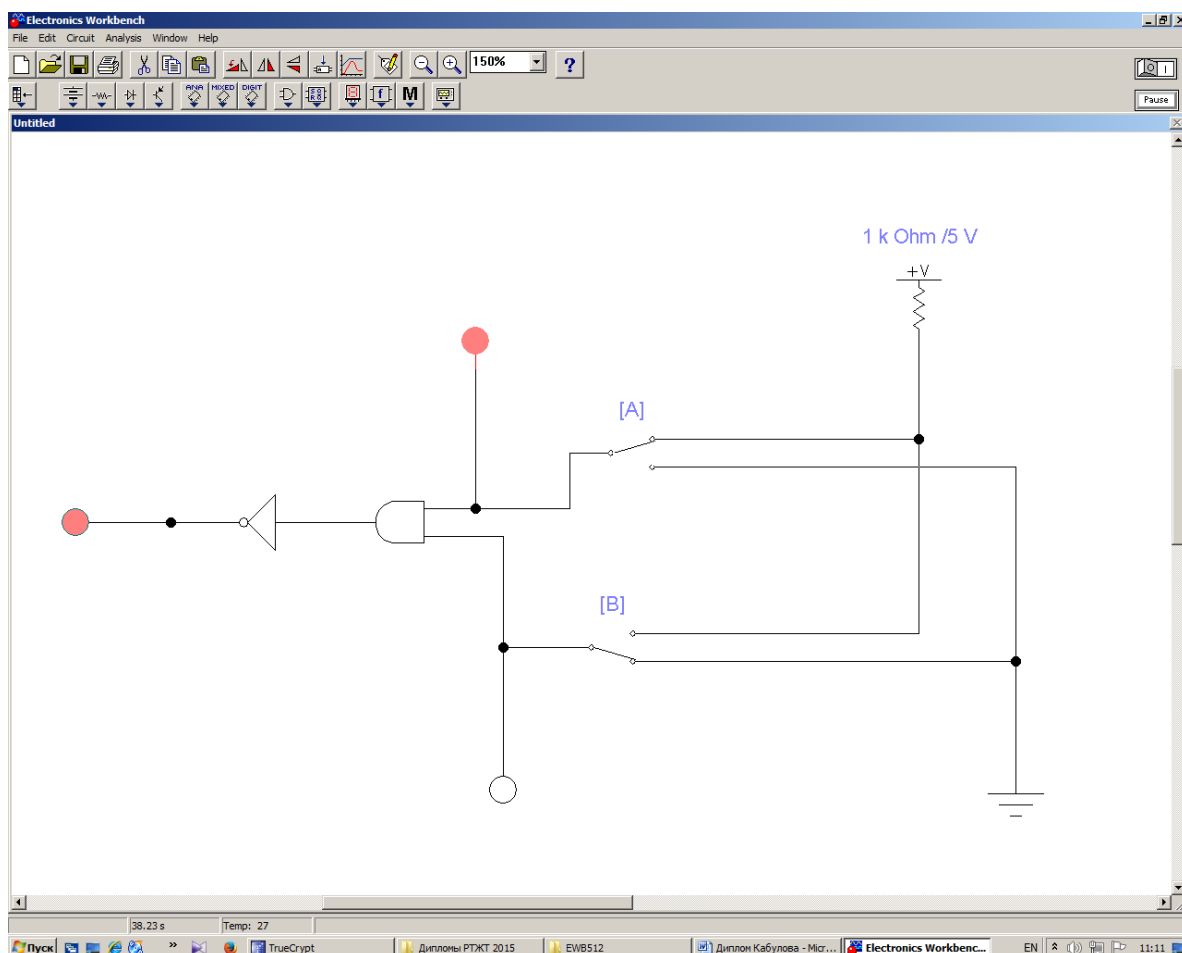


Рисунок 4

Включите схему. Подайте на входы схемы все возможные комбинации уровней входных сигналов и, наблюдая уровни сигналов на входах и выходе с помощью логических пробников, заполните таблицу истинности логической схемы «2И-НЕ» (таблица 4 в разделе «Результаты экспериментов»).

б) Экспериментальное получение таблицы истинности логического элемента «2И-НЕ».

Соберите схему, изображенную на рисунке 5.

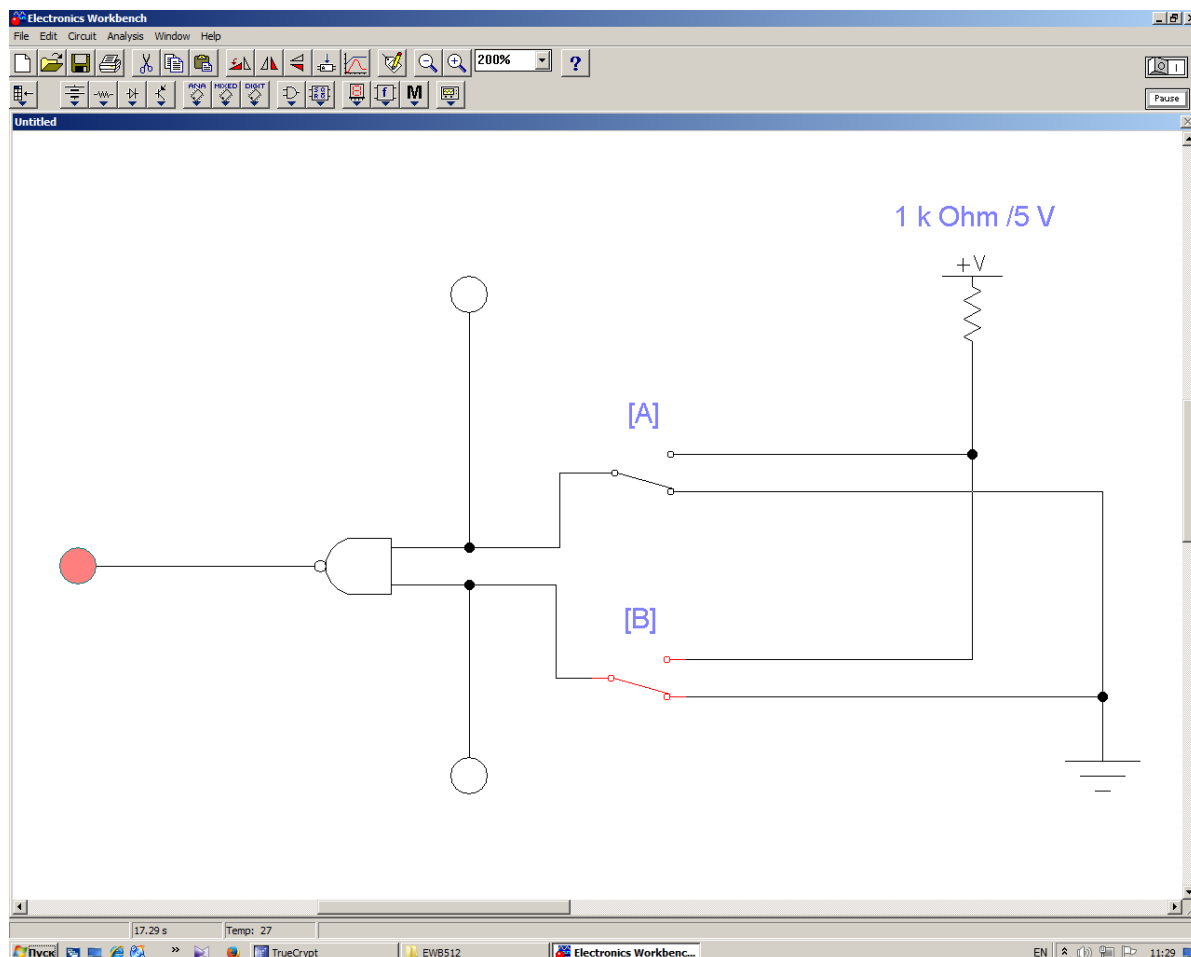


Рисунок 5

Включите схему. Подайте на входы схемы все возможные комбинации уровней входных сигналов и, наблюдая уровни сигналов на входах и выходе с помощью логических пробников, заполните таблицу истинности логической схемы «2И-НЕ» (таблица 5 в разделе «Результаты экспериментов»). Сравните таблицы 4 и 5 между собой.

Эксперимент 3. Исследование логической функции «ИЛИ».

а) Экспериментальное получение таблицы истинности логического элемента «ИЛИ».

Соберите схему, изображенную на рисунке 6.

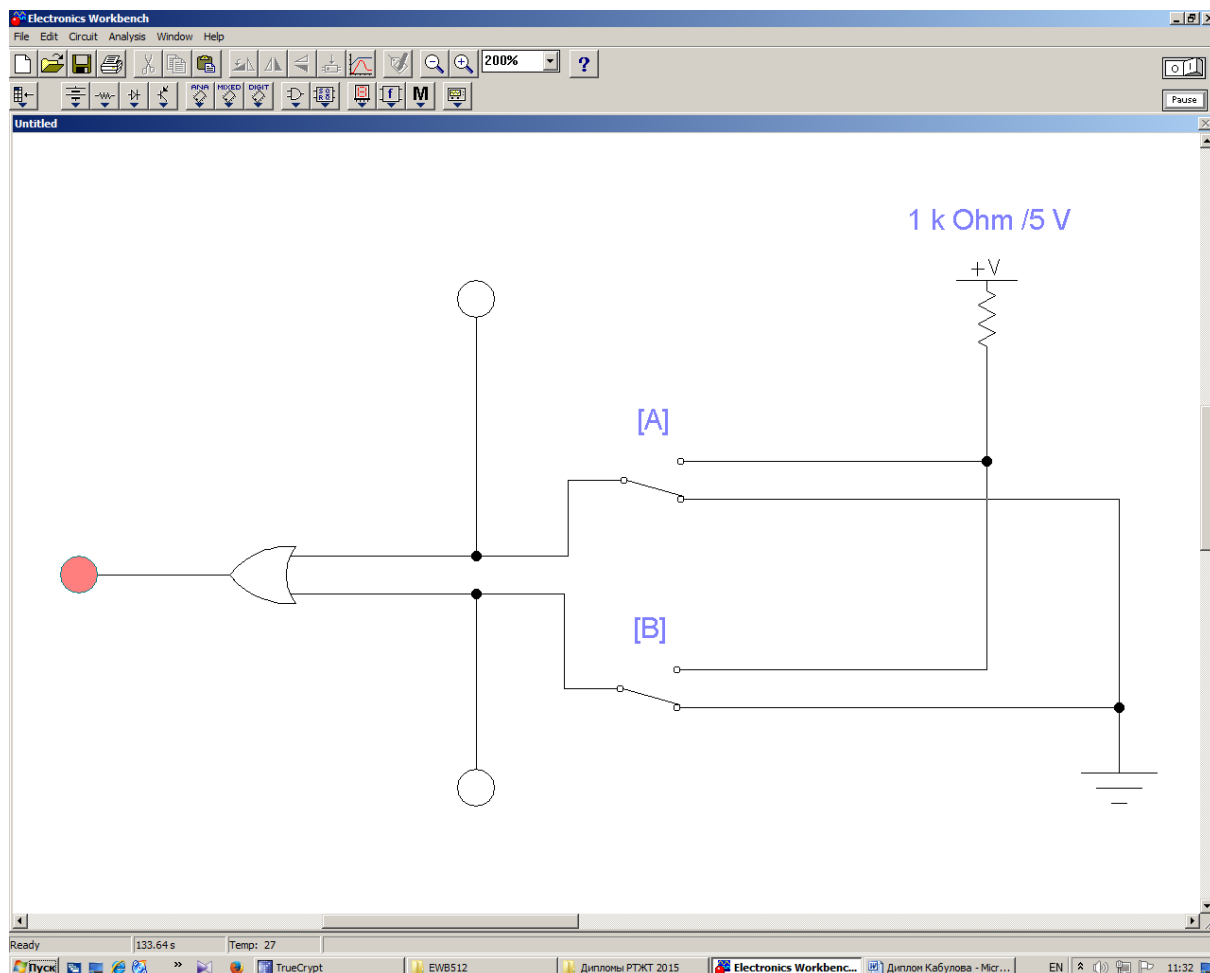


Рисунок 6

Включите схему. Подайте на входы схемы все возможные комбинации уровней входных сигналов и, наблюдая уровни сигналов на входах и выходе с помощью логических пробников, заполните таблицу истинности логической схемы «ИЛИ» (таблица 6 в разделе «Результаты экспериментов»).

б) Получение аналитического выражения для функции.

По таблице 6 составьте аналитическое выражение функции и занесите его в раздел «Результаты экспериментов».

Эксперимент 4. Исследование логической функции «ИЛИ-НЕ».

а) Экспериментальное получение таблицы истинности логического элемента «2ИЛИ-НЕ», составленного из элементов «2ИЛИ» и «НЕ».

Соберите схему, изображенную на рисунке 7.

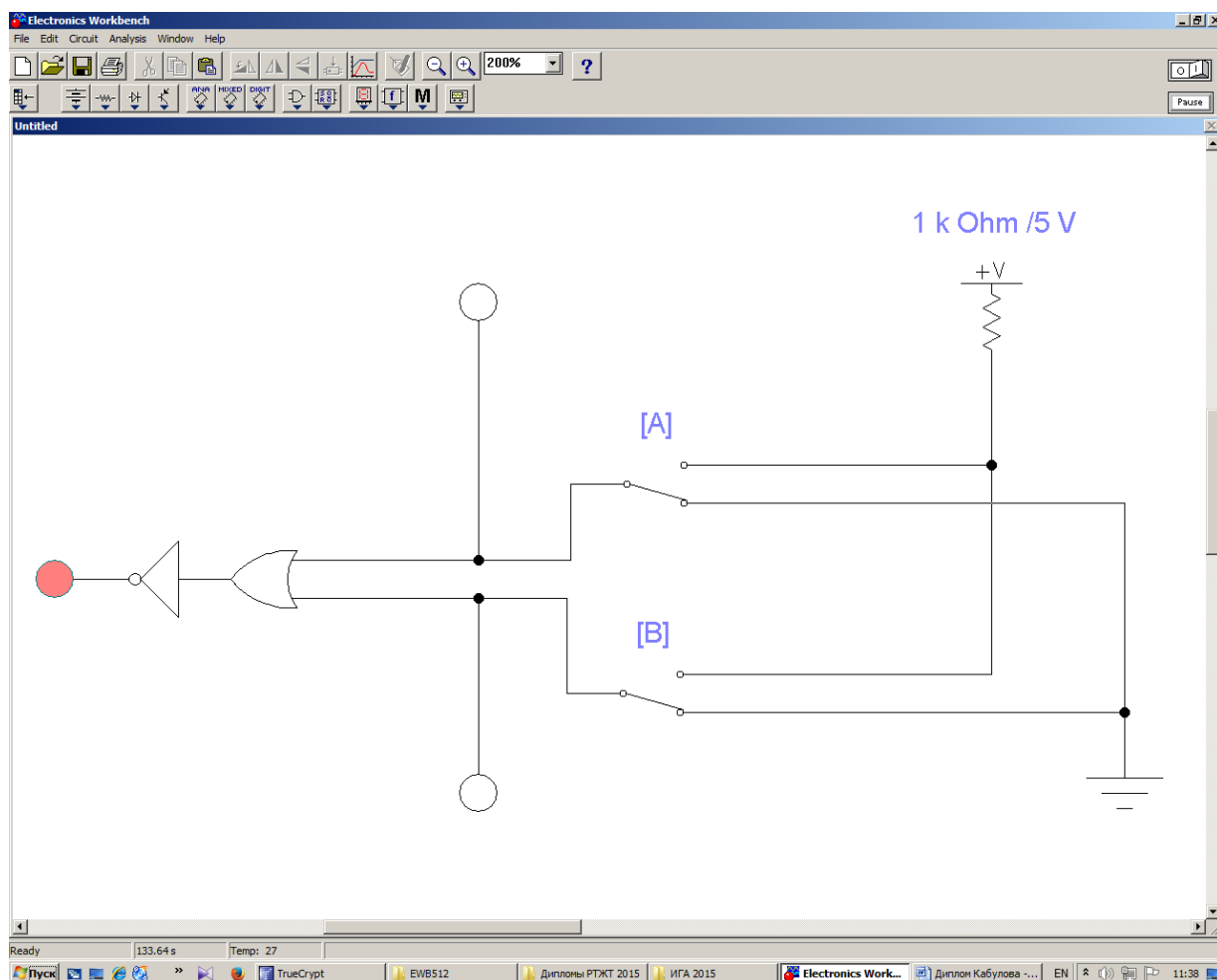


Рисунок 7

Включите схему. Подайте на входы схемы все возможные комбинации уровней входных сигналов и, наблюдая уровни сигналов на входах и выходе с помощью логических пробников, заполните таблицу истинности логической схемы «2ИЛИ-НЕ» (таблица 7 в разделе «Результаты экспериментов»).

б) Экспериментальное получение таблицы истинности логического элемента «2ИЛИ-НЕ».

Соберите схему, изображенную на рисунок 8.

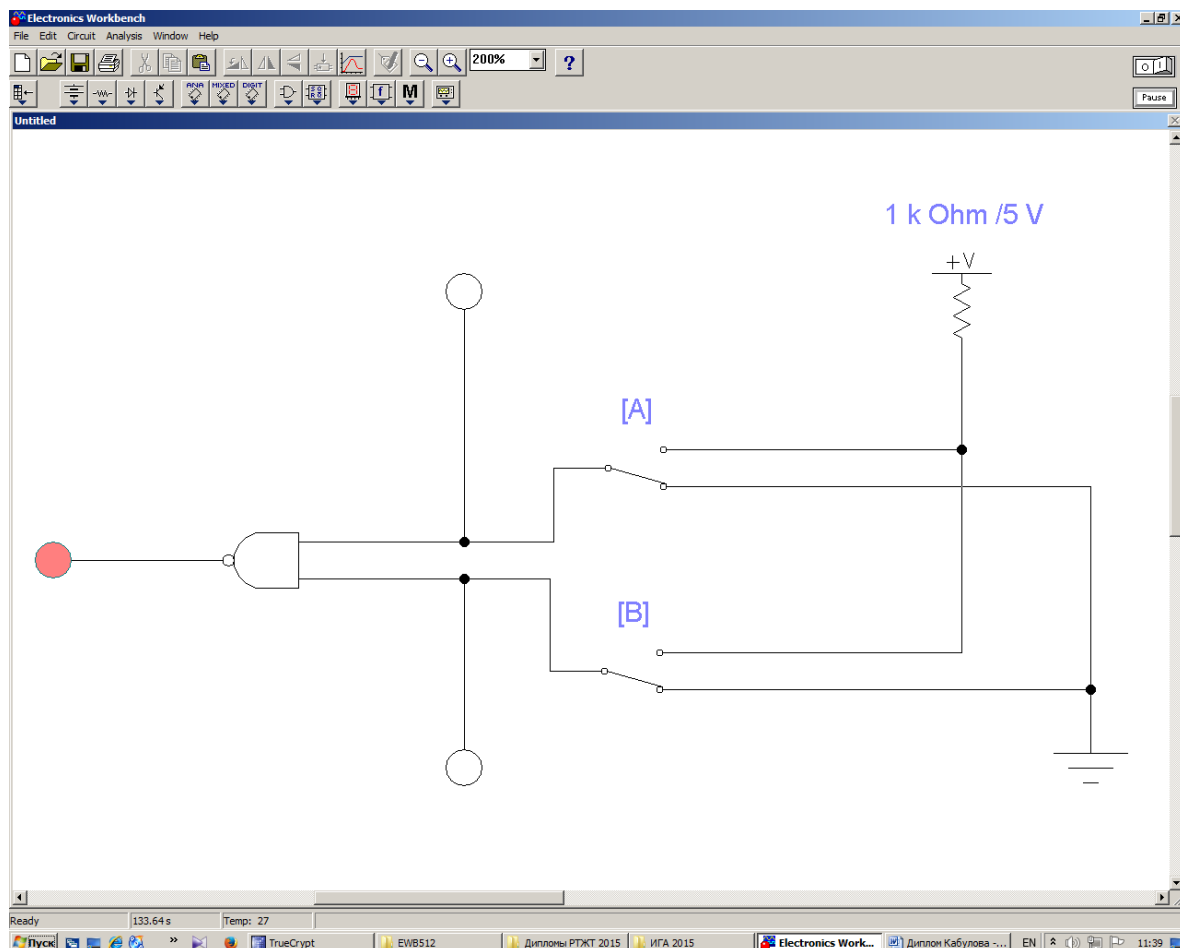


Рисунок 8

Включите схему. Подайте на входы схемы все возможные комбинации уровней входных сигналов и, наблюдая уровни сигналов на входах и выходе с помощью логических пробников, заполните таблицу истинности логической схемы «2ИЛИ-НЕ» (таблица 8 в разделе «Результаты экспериментов»).

Сравните таблицы 7 и 8 между собой.

Эксперимент 5. Исследование логических схем с помощью генератора слов.

а) Сведения об исследуемой микросхеме.

Откройте файл ewb01_02 со схемой, изображенной на рисунке 9.

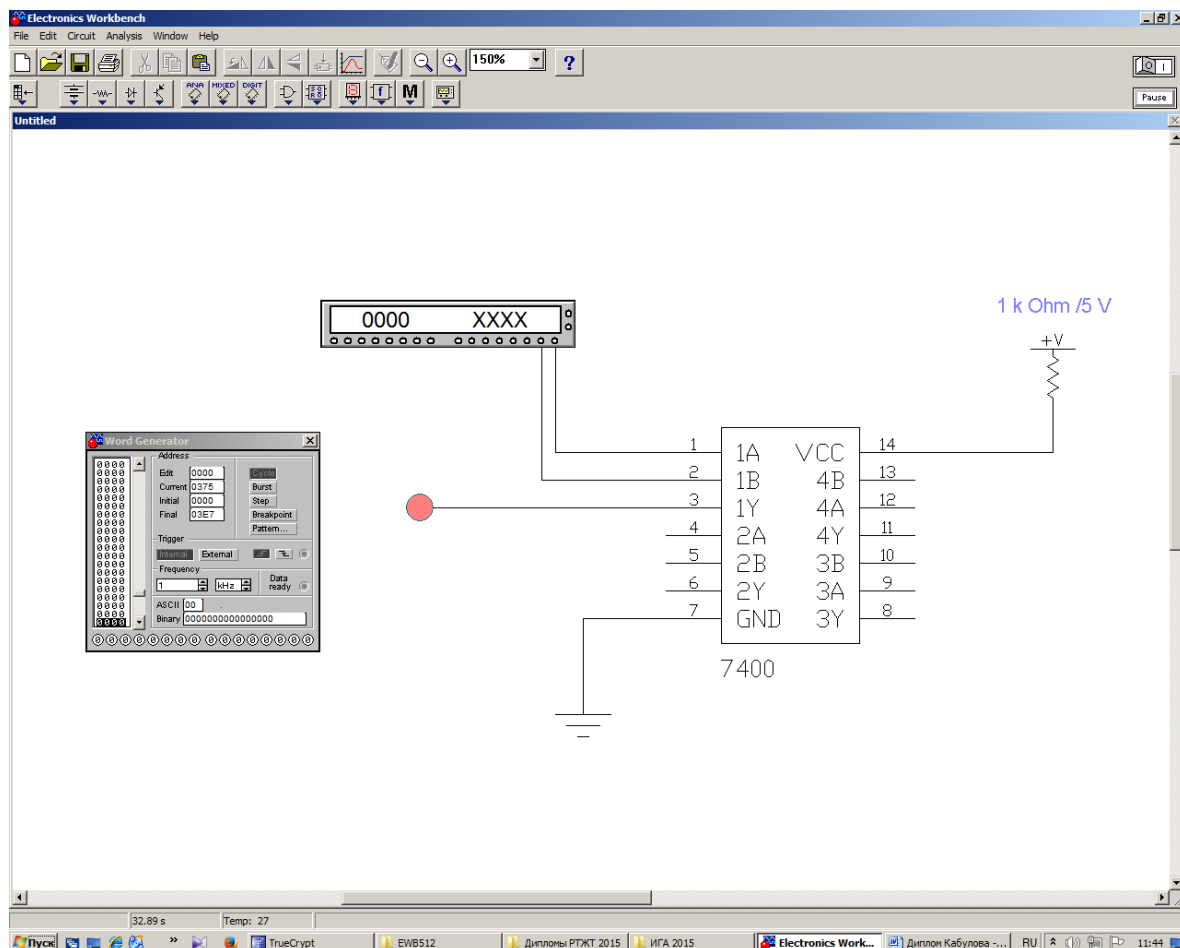


Рисунок 9

Включите схему. Укажите, к каким выводам микросхемы 7400 подключается источник питания, сколько элементов «2И-НЕ» содержит микросхема, сколько элементов используется в данном эксперименте и как обозначены на схеме используемые входы и выходы.

Заполните таблицу сведений о микросхеме (таблица 9 в разделе «Результаты экспериментов»).

б) Экспериментальное получение таблицы истинности логического элемента «2И-НЕ».

Запрограммируйте генератор слов так, чтобы на выходе генератора получать последовательно следующие комбинации: 00, 01, 10, 11. Переведите генератор в режим пошаговой работы нажатием кнопки «Step» на увеличенном изображении генератора. Каждое нажатие кнопки «Step» вызывает переход к очередному слову заданной последовательности, которое подается на выход гене-

ратора. Последовательно подавая на микросхему слова из заданной последовательности, заполните таблицу истинности элемента «2И-НЕ» (таблица 10 в разделе «Результаты экспериментов»).

Указание: значения разрядов текущего слова на выходе генератора отображаются в круглых окнах в нижней части на панели генератора.

Эксперимент 6. Реализация логической функции 3-х переменных.

а) Синтез схемы, реализующей функцию, заданную логическим выражением.

Реализуйте функцию $f = ab \vee bc$ на элементах «2И-НЕ».

Указание. Представьте выражение функции через операции логического умножения и инверсии.

Соберите в Electronics WorkBench схему на элементах «2И-НЕ», соответствующую полученному выражению. Подключите к входам схемы генератор слов, к выходу - логический пробник. Генератор слов запрограммируйте на формирование последовательности из восьми слов, соответствующих числам от 0 до 7:

0=000	1=001	2=010	3=011
4=100	5=101	6=110	7=111

В пошаговом режиме, последовательно подавая на вход полученной схемы все слова последовательности, определите при помощи логического пробника уровень сигнала на выходе схемы. По полученным результатам заполните таблицу 11 в разделе «Результаты экспериментов».

б) Синтез схемы, реализующей заданную функцию при помощи логического преобразователя.

Для получения схемы, реализующей функцию, описываемую логическим выражением $f = ab \vee bc$, можно воспользоваться логическим преобразователем.

Для этого сделайте следующее:

— вызовите логический преобразователь

— введите в нижнее окно панели преобразователя логическое выражение $ab \vee bc$ с клавиатуры (операции «ИЛИ» соответствует знак +, инверсия обозначается апострофом);

— для реализации схемы на элементах «И-НЕ» нажмите клавишу  на панели логического преобразователя.

Логический преобразователь выводит на рабочее поле схему, реализующую функцию, описываемую введенным логическим выражением. Полученная схема приведена на рисунке 10.

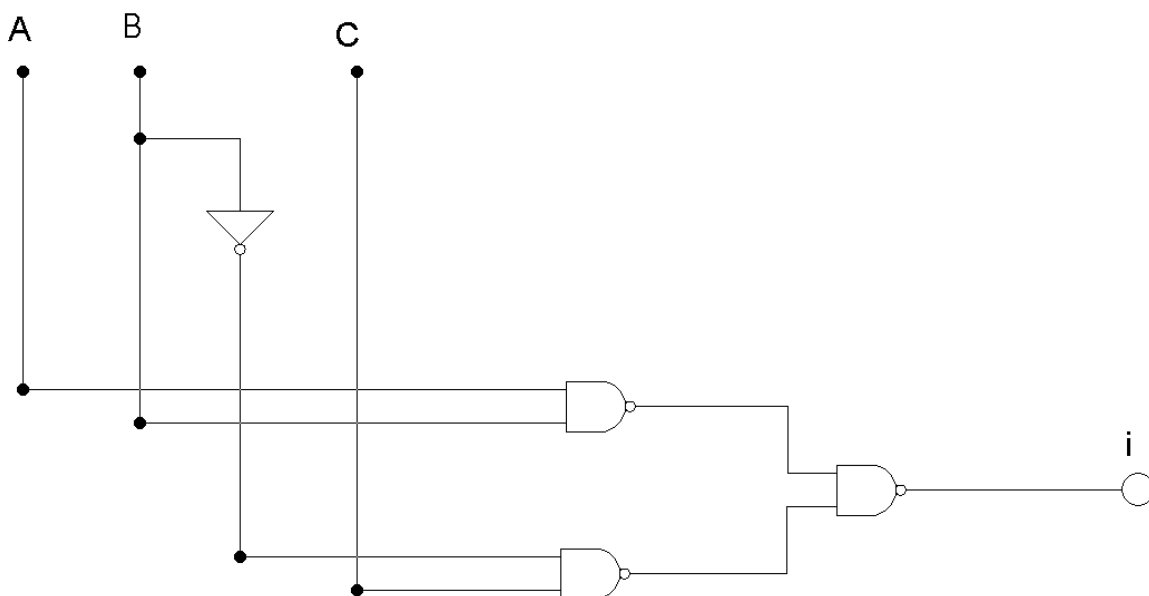


Рисунок 10

К схеме подключите генератор слов, запрограммированный на формирование восьми слов, соответствующих числам от 0 до 7:

0=000	1=001	2=010	3=011
4=100	5=101	6=110	7=111

Переведите генератор слов в пошаговый режим. Включите схему. Последовательно подавая на входы схемы указанные слова и определяя уровень сигнала на выходе схемы логическим пробником, заполните таблицу истинности (таблица 12 в разделе «Результаты экспериментов»).

Вычислите промежуточные значения и занесите их в таблицу истинности (таблица 12 в разделе «Результаты экспериментов»). Они определяют логи-

ческие сигналы на входе третьего элемента «2И-НЕ» в схеме (для контроля результатов вычисления можно к его входам подключить логические пробники).

Результаты экспериментов оформляются в отдельном разделе с одноименным названием.

Содержание раздела «Результаты экспериментов» приведено ниже.

Эксперимент 1. Исследование логической функции «И».

а) Задание уровней логических сигналов.

Привести результаты измерений при следующих положениях элементов схемы:

- напряжение на входе В (ключ В в нижнем положении), В;
- логический сигнал на входе В (ключ В в нижнем положении);
- напряжение на входе В (ключ В в нижнем положении), В;
- напряжение на входе В (ключ В в верхнем положении), В;
- логический сигнал на входе В (ключ В в верхнем положении);
- логический сигнал на выходе Y.

б) Экспериментальное получение таблицы истинности элемента «И»,

Таблица 3

Входы		Выход
A	B	Y
0	0	
0	1	
1	0	
1	1	

в) Получение аналитического выражения для функции.

Аналитическое выражение для функции:

--

Эксперимент 2. Исследование логической функции «И-НЕ».

а) Элемент «2И-НЕ», составленный из «2И» и «НЕ».

Таблица 4

Входы		Выход
A	B	Y
0	0	
0	1	
1	0	
1	1	

б) Элемент «2И-НЕ».

Таблица 5

Входы		Выход
A	B	Y
0	0	
0	1	
1	0	
1	1	

Эксперимент 3. Исследование логической функции «ИЛИ».

а) Экспериментальное получение таблицы истинности логического элемента «ИЛИ».

Таблица 6

Входы		Выход
A	B	Y
0	0	
0	1	
1	0	
1	1	

в) Получение аналитического выражения для функции.

Аналитическое выражение для функции:

--

Эксперимент 4. Исследование логической функции «ИЛИ-НЕ».

а). Элемент «ИЛИ-НЕ», составленный из «ИЛИ» и «НЕ».

Таблица 7

Входы		Выход
A	B	Y
0	0	
0	1	
1	0	
1	1	

б) Элемент «2И-НЕ».

Таблица 8

Входы		Выход
A	B	Y
0	0	
0	1	
1	0	
1	1	

Эксперимент 5. Исследование логических схем с помощью генератора слов.

а) Сведения об исследуемой микросхеме.

Таблица 9

Число элементов «И-НЕ» в микросхеме	
Число исследуемых элементов «2И-НЕ»	
Обозначение выводов для подключения источника питания	
Обозначения используемых входов	
Обозначение используемого выхода	

б) Экспериментальное получение таблицы истинности логического элемента «2И-НЕ» микросхемы 7400.

Таблица 10

Входы		Выход
A	B	Y
0	0	
0	1	
1	0	
1	1	

Эксперимент 6. Реализация логической функции 3-х переменных.

а) Синтез схемы, реализующей функцию, заданную логическим выражением.

Таблица 11

a	b	c	f
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

б) Синтез схемы, реализующей заданную функцию при помощи логического преобразователя.

Таблица 12

a	b	c	$\overline{ab}$	$\overline{ac}$	f
0	0	0			
0	0	1			
0	1	0			
0	1	1			
1	0	0			
1	0	1			
1	1	0			
1	1	1			

Содержание отчета

1. Тема, цель.
2. Схемы экспериментов.
3. Результаты экспериментов.

4. Выводы.

Контрольные вопросы

1. Составьте таблицу истинности элемента «И».
2. Составьте таблицу истинности элемента «ИЛИ».
3. Составьте таблицу истинности элемента «И-НЕ».
4. Составьте таблицу истинности элемента «ИЛИ-НЕ».
5. Что такое конъюнкция?
6. Что такое дизъюнкция?

Практическое занятие №5

Исследование работы RS и D триггеров

Цель занятия: провести моделирования работы RS и D триггеров.

Краткие теоретические сведения

Триггер — цифровой автомат, имеющий два устойчивых состояния равновесия либо 0, либо 1.

Состояние триггера распознается по его выходному сигналу. Под влиянием входного сигнала триггер скачкообразно переходит из одного устойчивого состояния в другое, при этом скачкообразно изменяется уровень напряжения его выходного сигнала. Для удобства использования в схемах вычислительных устройств триггеры обычно имеют два выхода: прямой Q (называется также «выход 1») и инверсный $\bar{Q}$ («выход 0»). В единичном состоянии триггеры на выходе Q имеют высокий уровень сигнала, а в нулевом — низкий. На выходе $\bar{Q}$ наоборот.

Триггер типа RS имеет два входа раздельной установки в нулевое и единичное состояния. Воздействие по входу S (обозначен по первой букве слова set

— установка) приводит триггер в единичное состояние, а воздействие по входу R (от первой буквы слова reset — сброс) — в нулевое. Одновременная подача сигналов S и R не допускается.

Таблица 1 - Таблица истинности переходов асинхронного RS- триггера на элементах «ИЛИ-НЕ»

R	S	Q	Примечание
0	0	Q	Хранение
0	1	1	Установка 1
1	0	0	Установка 0
1	1	-	Запрещено

Триггер образован из двух комбинационных схем «ИЛИ-НЕ», соединенных таким образом, что возникают положительные обратные связи, благодаря которым в устойчивом состоянии выходной транзистор одной схемы «ИЛИ-НЕ» закрыт, а другой открыт (рисунок 1).

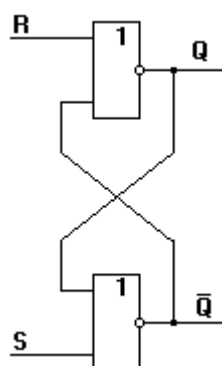


Рисунок 1 - Функциональная схема асинхронного RS-триггера

D-триггер (от слова delay — задержка) принимает информацию по одному входу и реализует функцию временной задержки. D-триггер имеет только режимы установки 1 и 0.

Таблица 2 - Таблица истинности синхронного D-триггера

C	D	Q(t)	Q(t+1)
0	x	0	0
0	x	1	1
1	0	x	0
1	1	x	1

В связи с этим не синхронизируемый D-триггер не применяется, так как его выход будет просто повторять входной сигнал. Синхронизируемый одно-тактный D-триггер задерживает распространение входного сигнала на время паузы между синхросигналами (задержка на половина периода) (рисунок 2).

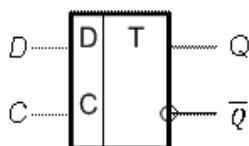


Рисунок 2 - Условно графическое обозначение синхронного D-триггера

Порядок выполнения работы

1. Соберите схемы для проведения экспериментов.

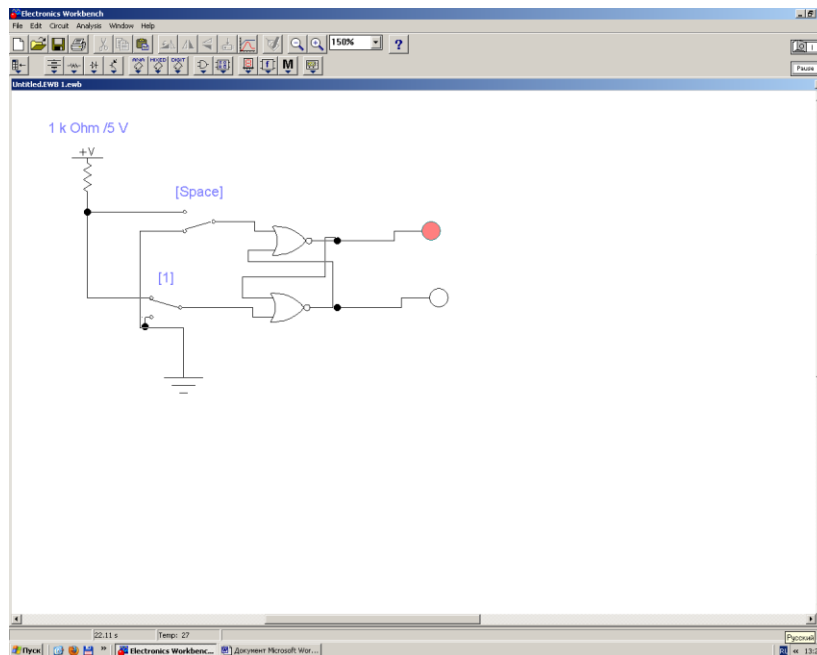


Рисунок 3 – Исследование RS-триггера (схема 1)

При моделировании схемы эксперимента были использованы следующие элементы:

- логические элементы ИЛИ-НЕ (2 штуки);
- источник напряжения, характеризующийся ЭДС;
- переключатели типа однополюсного тумблера (2 штуки);
- заземление;
- логические пробники (2 штуки).

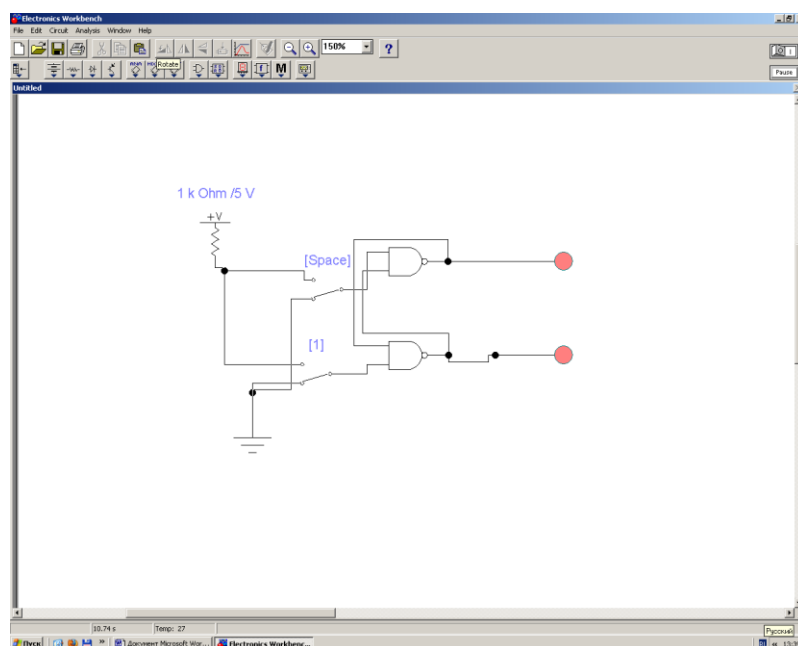


Рисунок 4 - Исследование RS-триггера (схема 2)

При моделировании схемы эксперимента были использованы следующие элементы:

- логические элементы И-НЕ (2 штуки);
- источник напряжения, характеризуемый ЭДС;
- переключатели типа однополюсного тумблера (2 штуки);
- заземление;
- логические пробники (2 штуки).

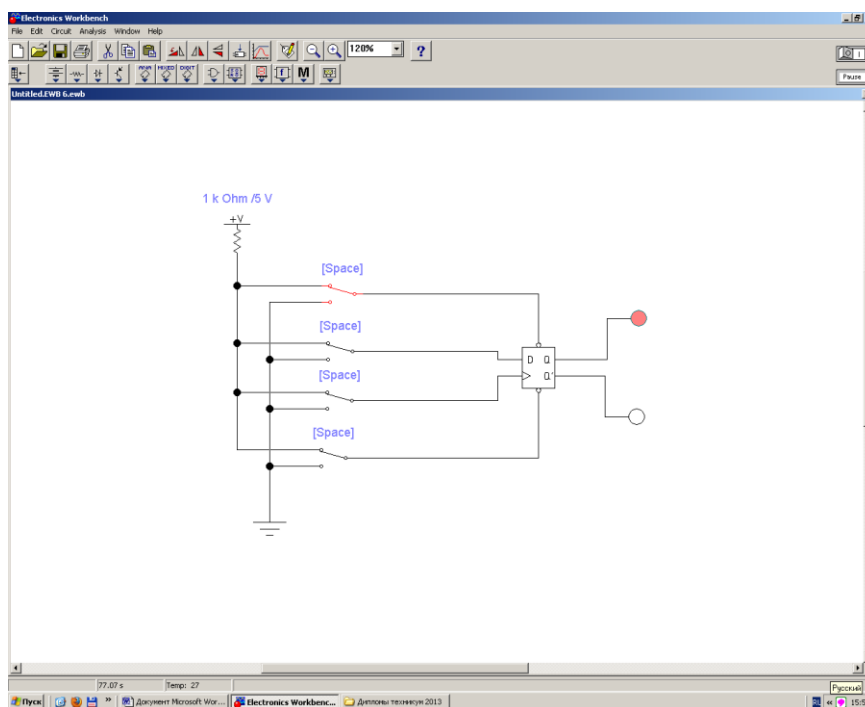


Рисунок 5 - Исследование D-триггера

При моделировании схемы эксперимента были использованы следующие элементы:

- D-триггер;
- источник напряжения, характеризуемый ЭДС;
- переключатели типа однополюсного тумблера (4 штуки);
- заземление;
- логические пробники (2 штуки).

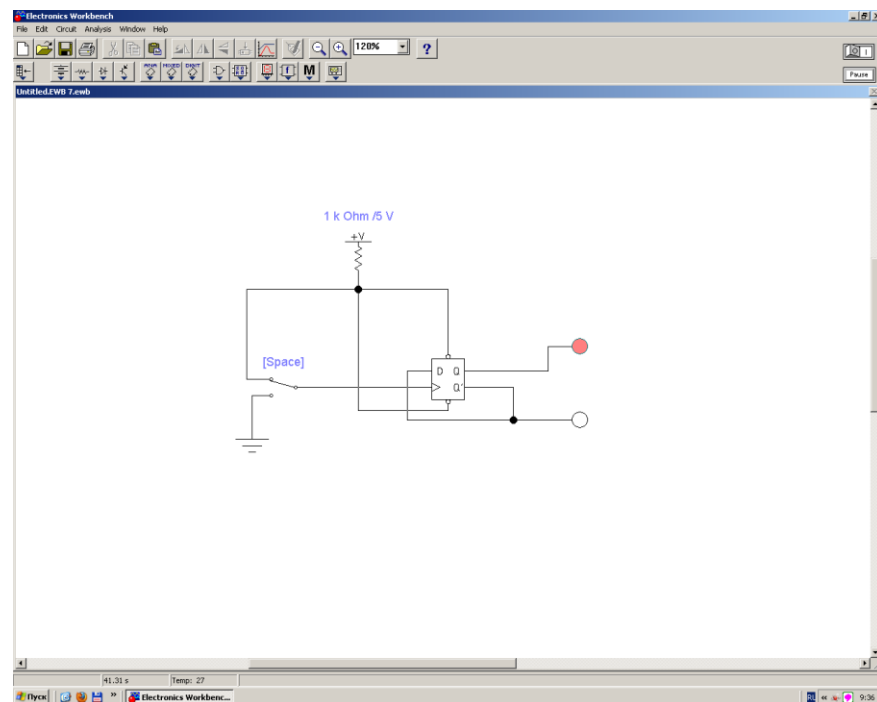


Рисунок 6 – Исследование работы D-триггера в счетном режиме

При моделировании схемы эксперимента были использованы следующие элементы:

- D-триггер;
- источник напряжения, характеризуемый ЭДС;
- переключатель типа однополюсного тумблера;
- заземление;
- логические пробники (2 штуки).

2. Выполните эксперименты в соответствии с приведенным заданием.

Эксперимент 1. Исследование RS-триггера.

1. Откройте файл EWB1 со схемой, изображенной на рисунке 3.
2. Включите схему.
3. Последовательно подайте на схему следующие сигналы: S=0, R=1; S=0, R=0; S=1, R=0; S=0, R=0.
4. Убедитесь в том, что:
 - при S=0, R=1 триггер устанавливается в состояние Q=0;

— при переходе к $S=0, R=0$ триггер сохраняет прежнее состояние выхода $Q=0$;

— при $S=1, R=0$ триггер устанавливается в состояние $Q=1$;

— при переходе к $S=0, R=0$ триггер сохраняет прежнее состояние выхода $Q=1$.

5. Для каждого перехода (изменения состояния или сохранения предыдущего) нарисуйте в разделе «Результаты экспериментов» граф перехода.

6. По результатам эксперимента для схемы (рисунок 3) заполните таблицу функций возбуждения (таблица 3).

Таблица 3 – Таблица функций возбуждения

Q_t	Q_{t+1}	R	S
0		0	0
0		1	0
0		0	1
1		1	0
1		0	0
1		0	1

Эксперимент 2. Исследование RS-триггера.

1. Откройте файл EWB2 со схемой, изображенной на рисунке 4.

2. Включите схему.

3. Последовательно подайте на схему следующие сигналы: $S=1, R=0$; $S=0, R=0$; $S=0, R=1$; $S=0, R=0$.

4. Убедитесь в том, что:

— при $S=1, R=0$ триггер устанавливается в состояние, при котором выход $Q=0$;

— при переходе к $S=R=1$ триггер сохраняет прежнее значение выхода $Q=0$

— при $S=0, R=1$, триггер устанавливается в состояние, при котором $Q=1$;

— при переходе к $S=1, R=1$ прежнее значение выхода $Q=1$ сохраняется.

5. Для каждого перехода (изменения состояния или сохранения предыдущего) нарисуйте в разделе «Результаты экспериментов» граф перехода.

6. По результатам эксперимента для схемы (рисунок 4) заполните таблицу функций возбуждения (таблица 4).

Эксперимент 3. Исследование D-триггера.

1. Откройте файл EWB6 со схемой, изображенной на рисунке 5.

2. Включите схему.

3. Убедитесь в том, что:

— при $R=1, S=0$ триггер устанавливается в 1 ($Q=1, Q^*=0$) независимо от состояния остальных входов;

— при $R=0, S=1$ триггер устанавливается в 0 ($Q=0, Q^*=1$) независимо от состояния остальных входов.

4. Установите $S'=R'=1$, проверьте истинность таблицы функций возбуждения (таблица 5), по результатам эксперимента заполните таблицу 6.

5. Составьте временные диаграммы работы триггера для всех возможных комбинаций Q_t, D_t и зарисуйте их в раздел «Результаты экспериментов».

Таблица 5 – Таблица функций возбуждения

Q_t	Q_{t+1}	D
0	0	0
0	1	1
1	0	0
1	1	1

Таблица 6 – Таблица функций возбуждения

Q_t	Q_{t+1}	J	K
0		0	0
0		0	1
0		1	0
0		1	1
1		0	0
1		0	1
1		1	0
1		1	1

Эксперимент 4. Исследование работы D-триггера в счетном режиме.

1. Соберите схему, изображенную на рисунке 6. Подавая на счетный вход С тактовые импульсы с помощью ключа [С] и определяя состояние выходов триггера при помощи пробников, составьте временные диаграммы работы триггера в счетном режиме и занесите их в раздел «Результаты экспериментов».

3. Оформите отчет по занятию.

Содержание отчета

1. Тема, цель занятия.
2. Схемы экспериментов.
3. Результаты экспериментов.
4. Выводы.

Контрольные вопросы

1. Дайте определение триггеру.
2. Поясните работу RS-триггера.
3. Поясните работу D-триггера.
4. Поясните работу D-триггер в счетном режиме.
5. Приведите таблицу функций возбуждения RS-триггера.
6. Приведите таблицу функций возбуждения D-триггера.

Практическое занятие №6

Исследование JK и T триггеров

Цель занятия: исследование работы JK и T триггеров.

Краткие теоретические сведения

Триггер — цифровой автомат, имеющий два устойчивых состояния равновесия либо 0, либо 1.

Состояние триггера распознается по его выходному сигналу. Под влиянием входного сигнала триггер скачкообразно переходит из одного устойчивого состояния в другое, при этом скачкообразно изменяется уровень напряжения его выходного сигнала. Для удобства использования в схемах вычислительных устройств триггеры обычно имеют два выхода: прямой Q (называется также «выход 1») и инверсный $\bar{Q}$ («выход 0»). В единичном состоянии триггеры на выходе Q имеют высокий уровень сигнала, а в нулевом — низкий. На выходе $\bar{Q}$ наоборот.

T-триггер — является триггером со счетным входом (или счетным триггером). Он изменяет свое состояние на противоположное каждый раз, когда на его вход приходит очередной сигнал. Обозначение триггера пришло от первой буквы английского слова toggle— защелка (рисунок 1).

Таблица 1 - Таблица истинности T-триггер

T	Q(T)	Q(t+1)
0	0	0
0	1	1
1	0	1
1	1	0

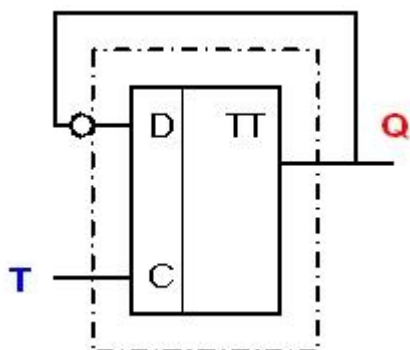


Рисунок 1 - Функциональная схема T-триггера

JK-триггер — универсален, с отдельной установкой нулевого и единичного состояния, в зависимости от соединения его входов он может работать как RS, T, D — триггер. В отличие от триггера типа RS в нем не запрещена одновременная подача сигналов на оба входа. Входы J и K эквивалентны входам S и R установки триггера соответственно в состояния «1» и «0». При объединении входов J и K и при подаче на них счетных импульсов. Вход J при раздельном использовании входов играет роль входа установки в единицу, а вход K — роль входа установки в нуль (рисунок 2).

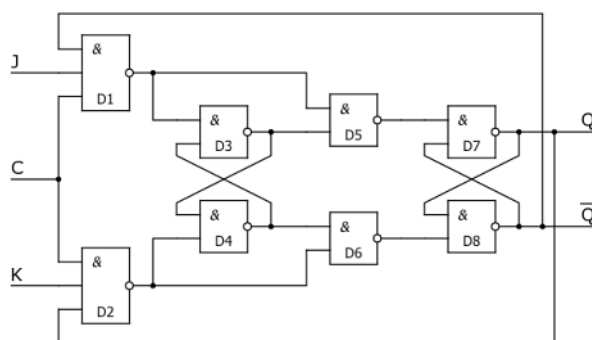


Рисунок 2 - Функциональная схема JK-триггер

Таблица 2 - Таблица истинности JK-триггер

C	K	J	Q(t)	Q(t+1)
0	x	x	0	0
0	x	x	1	1
1	0	0	0	0
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	0
1	1	0	1	0
1	1	1	0	1
1	1	1	1	1

Порядок выполнения работы

1. Смоделируйте следующие схемы лабораторных установок.

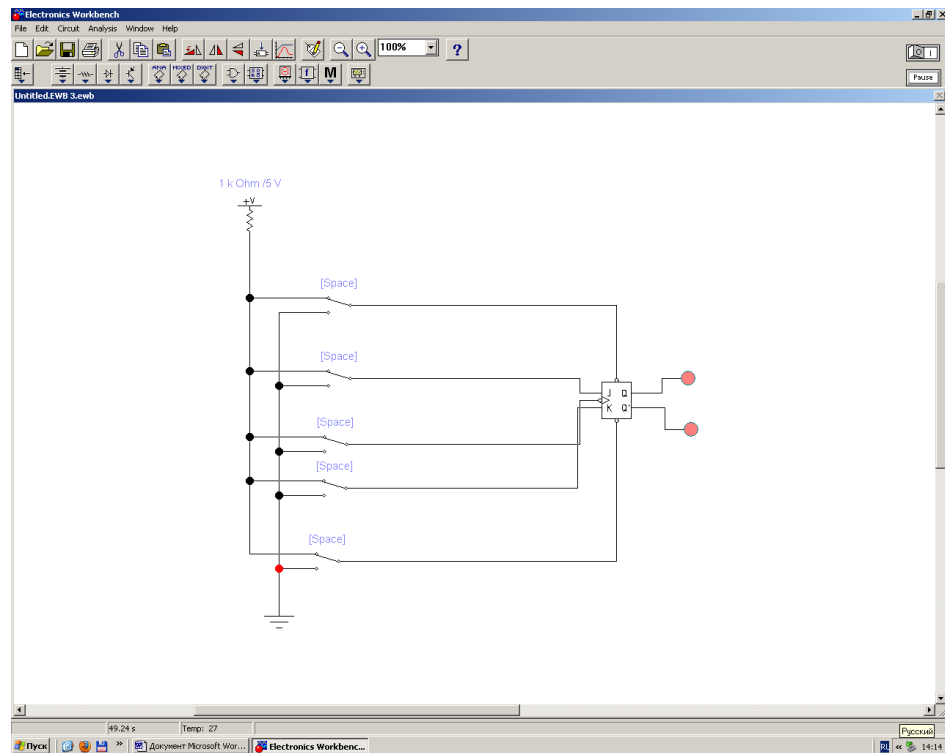


Рисунок 3 – Исследование JK-триггера

При моделировании схемы эксперимента были использованы следующие элементы:

- JK-триггер;
- источник напряжения, характеризуемый ЭДС;
- переключатели типа однополюсного тумблера (5 штук);
- заземление;
- логические пробники (2 штуки).

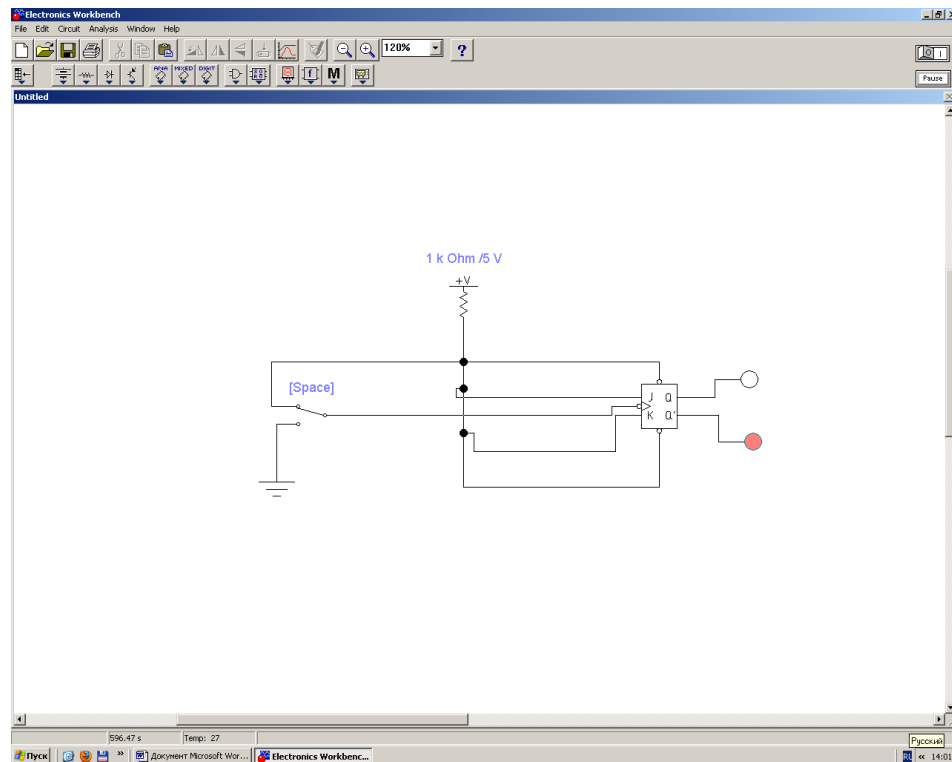


Рисунок 4 - Исследование JK-триггера в счетном режиме (Т-триггер)

При моделировании схемы эксперимента были использованы следующие элементы:

- JK-триггер;
- источник напряжения, характеризуемый ЭДС;
- переключатель типа однополюсного тумблера;
- заземление;
- логические пробники (2 штуки).

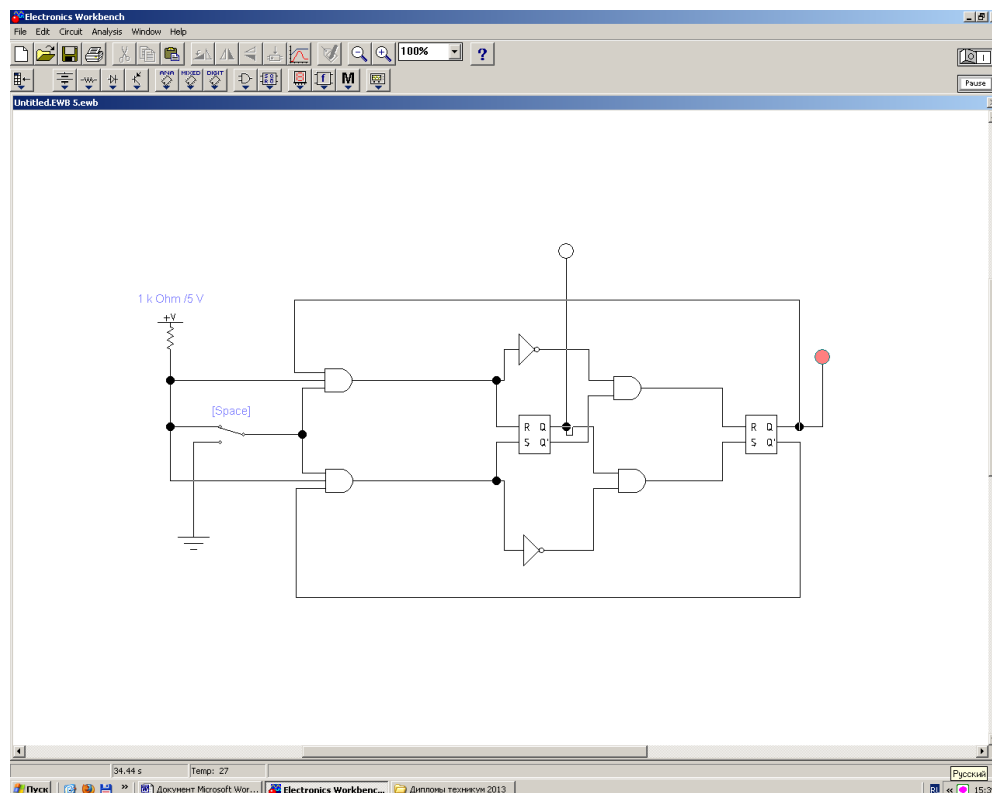


Рисунок 5 - Исследование JK-триггера, построенного на базе логических элементов и RS-триггеров

При моделировании схемы эксперимента были использованы следующие элементы:

- RS-триггер (2 штуки);
- логические элементы И на 3 входа (2 штуки);
- логические элементы И на 2 входа (2 штуки);
- логические элементы НЕ (2 штуки);
- источник напряжения, характеризуемый ЭДС;
- переключатель типа однополюсного тумблера;
- заземление;
- логические пробники (2 штуки).

2. Выполните эксперименты согласно приведенным заданиям.

Эксперимент 1. Исследование JK-триггера.

1. Откройте файл EWB3 со схемой, изображенной на рисунке 3.

2. Включите схему.

3. Убедитесь в том, что:

— при $R=1$, $S=0$ триггер устанавливается в 1 ($Q=1$, $Q'=0$) независимо от состояния остальных входов;

— при $R=0$, $S=1$ триггер устанавливается в 0 ($Q=0$, $Q'=1$) независимо от состояния остальных входов.

4. Установите $S'=R'=1$, проверьте истинность таблицы функций возбуждения (таблица 3), по результатам эксперимента заполните таблицу 4.

Таблица 3 – Таблица функций возбуждения

Q_t	Q_{t+1}	J	K
0	0	×	0
0	1	0	1
1	0	1	0
1	1	0	×

Таблица 4 – Таблица функций возбуждения

Q_t	Q_{t+1}	J	K
0		0	0
0		0	1
0		1	0
0		1	1
1		0	0
1		0	1
1		1	0
1		1	1

Указание: начальное состояние триггера устанавливать кратковременной подачей сигнала $S=0$ для получения $Q=1$ и сигнала $R'=0$ для получения $Q_t=0$. Переход триггера в состояние Q_{t+1} происходит только по отрицательному фронту импульса на счетном входе C, сформированном соответствующим ключом.

Составьте временные диаграммы работы триггера для всех возможных комбинаций Q_t , J_t , K_t и зарисуйте их в раздел «Результаты экспериментов».

Эксперимент 2. Исследование JK-триггера в счетном режиме (Т-триггер).

1. Соберите схему, изображенную на рисунке 4 (файл EWB4).
2. Включите схему.
3. Изменяя состояние входа С соответствующим ключом, зарисуйте в разделе «Результаты экспериментов» диаграммы работы триггера в счетном режиме.

Эксперимент 3. Исследование JK-триггера, построенного на базе логических элементов и RS-триггеров.

1. Откройте файл EWB5 со схемой, изображенной на рисунке 5.
 2. Включите схему.
 3. Изменяя уровень сигнала на входе С, составьте временные диаграммы сигналов на выходах Q1 и Q2 обоих RS-триггеров и зарисуйте их в разделе «Результаты экспериментов».
 4. Укажите режим работы триггера.
 5. Определите моменты изменения сигналов Q1 и Q2 по отношению к моментам изменения сигнала С.
 6. Отрадите различие во временах переключения RS-триггеров на диаграммах.
3. Оформите отчет о результатах работы.

Содержание отчета

1. Тема, цель занятия.
2. Схемы экспериментов.
3. Результаты экспериментов.
4. Выводы.

Контрольные вопросы

1. Дайте определение триггеру.
2. Поясните работу JK-триггера.
3. Поясните работу T-триггера.
4. Приведите таблицу функций возбуждения JK-триггера.
5. Приведите таблицу функций возбуждения T-триггера.

Практическое занятие №7

Исследование работы регистров

Цель занятия: изучить принципы работы регистров различных типов.

Краткие теоретические сведения

Основное назначение регистров — хранение и преобразование многоразрядных двоичных чисел. Регистры наряду со счетчиками и запоминающими устройствами являются наиболее распространенными устройствами цифровой техники. При сравнительной простоте регистры обладают большими функциональными возможностями. Они используются в качестве управляющих и запоминающих устройств, генераторов и преобразователей кодов, счетчиков, делителей частоты, узлов временной задержки. Элементами структуры регистров являются синхронные триггеры D- или JK-типа с динамическим или статическим управлением. Одиночный триггер может запоминать (регистировать) один разряд (бит) двоичной информации. Такой триггер можно считать одноразрядным регистром. Занесение информации в регистр называют операцией ввода или записи. Выдача информации к внешним устройствам характеризует операцию вывода или считывания. Запись информации в регистр не требует его предварительного обнуления.

Понятие «весовой коэффициент» к разрядам регистра в отличие от счетчика неприменимо, поскольку весовая зависимость между отдельными разрядами целиком определяется записанной в регистр информацией. По этой при-

чине на условных изображениях регистров нумерация информационных входов и выходов наносится подряд.

Все регистры в зависимости от функциональных свойств подразделяются на две категории — накопительные (регистры памяти, хранения) и сдвигающие. В свою очередь, сдвигающие регистры делятся по способу ввода и вывода информации на параллельные, последовательные и комбинированные (параллельно-последовательные и последовательно-параллельные), по направлению передачи (сдвига) информации — на однонаправленные и реверсивные.

Наиболее простыми регистрами являются регистры памяти. Их назначение — хранение двоичной информации небольшого объема в течение короткого промежутка времени. Эти регистры представляют собой набор синхронных триггеров, каждый из которых хранит один разряд двоичного числа. Ввод (запись) и вывод (считывание) информации производится параллельным кодом. Ввод обеспечивается тактовым импульсом, с приходом очередного тактового импульса записанная информация обновляется. Считывание производится в прямом или в обратном коде (в последнем случае с инверсных выходов).

Регистры хранения представляют собой наборы триггеров с независимыми информационными входами и обычно общим тактовым входом. В таком качестве используются синхронные триггеры, составленные из микросхем, содержащих в одном корпусе несколько самостоятельных триггеров, например K155TM8 (74175), K155TM9 (74179) и другие, которые можно рассматривать как 4—6-разрядные регистры памяти. Нарращивание разрядности регистров памяти достигается добавлением нужного числа триггеров, тактовые входы которых подсоединяют к шине синхронизации.

Регистр K155IP15 (74173) является библиотечным компонентом EWB и может служить примером устройства хранения с тремя выходными состояниями. Схема его включения приведена рисунке 1.

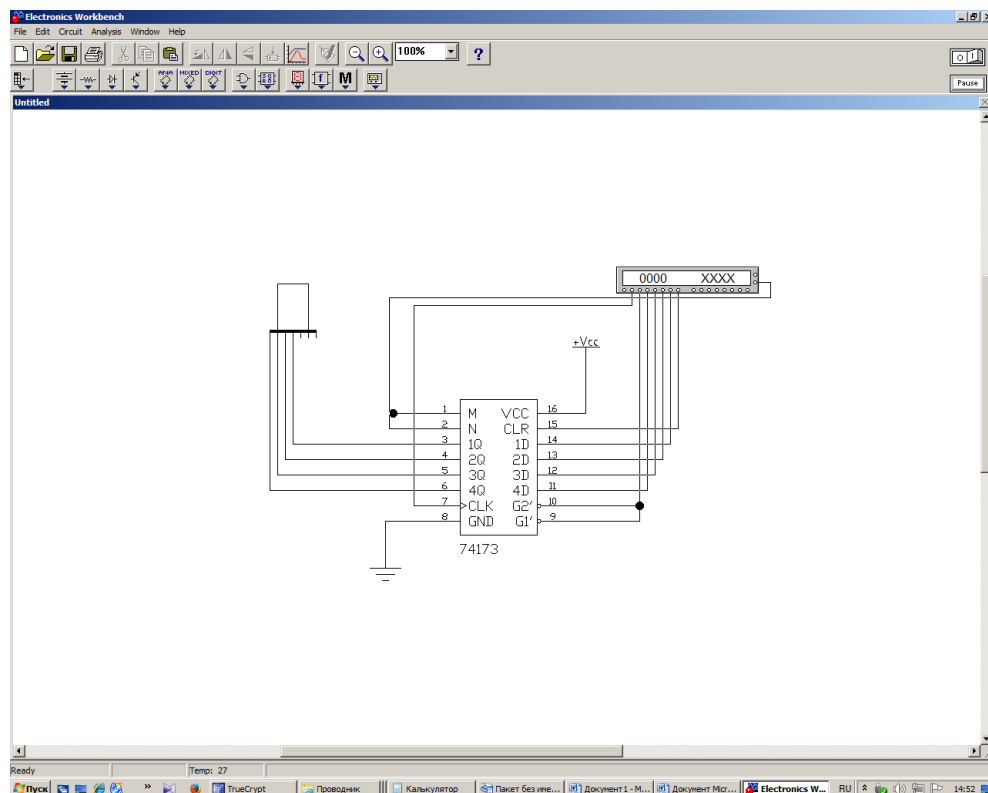


Рисунок 1 - Схема включения регистра 74173

Отметим крайне неудачное расположение выводов регистра 74173 и их несоответствие оригиналу. Как видно из рисунка 1, K155ИР15 — четырехразрядный регистр. Он имеет выходы 1Q...4Q с третьим Z-состоянием (при сигнале 1 на выводах G2, G1), а его входы 1D...4D снабжены логическими элементами разрешения записи путем подачи логического 0 на входы M, N (в EWB ошибочно показаны прямыми). Используется регистр как четырехразрядный источник кода, способный обслуживать непосредственно шину данных цифровой системы.

Загрузка информации в регистр производится синхронно с положительным перепадом тактового импульса, если на входах M, N присутствуют напряжения низкого уровня. Если на одном из этих входов напряжение высокого уровня, после прихода положительного тактового перепада в регистре должны остаться прежние данные. Вход сброса CLR имеет высокий активный уровень. Если на входы G2, G1 подано напряжение активного низкого уровня, данные, содержащиеся в регистре, отображаются на выходах 1Q...4Q, присутствие хотя бы одного напряжения высокого уровня на входах разрешения G2 и G1 вызы-

вает Z-состояние (размыкание) для выходных линий. При этом данные из регистров шину данных систем не проходят, выходы регистра не влияют на работу других аналогичных выходов, присоединенных к проводникам шины. На работу входов сброса CLR и тактового C смена уровней на входах разрешения влияния не оказывает.

Регистр K155IP15 потребляет ток 72 мА и имеет тактовую частоту до 25 МГц; вариант 74LS173 потребляет ток 30 мА, его тактовая частота 30 МГц.

Вторым наиболее распространенным классом регистров являются регистры сдвига, которые отличаются большим разнообразием как в функциональном отношении, так и в отношении схемных решений и характеристик. Регистры сдвига, помимо операции хранения, осуществляют преобразование последовательного двоичного кода в параллельный, а параллельного — в последовательный, выполняют арифметические и логические операции, служат в качестве элементов временной задержки. Своим названием они обязаны характерной для этих устройств операции сдвига. С приходом каждого тактового импульса происходит перезапись (сдвиг) содержимого триггера каждого разряда в соседний разряд без изменения порядка следования единиц и нулей. При сдвиге информации вправо после каждого тактового импульса бит из более старшего разряда сдвигается в младший, а при сдвиге влево — наоборот.

На отечественных схемах символом регистра служат буквы RG. Для регистров сдвига указывается также направление сдвига: > — вправо; < — влево; <-> — реверсивный (двунаправленный).

Работу регистра сдвига рассмотрим на примере библиотечного регистра 74195 (K155IP12), схема включения которого показана на рисунке 2. ИМС 74195 — быстродействующий регистр для выполнения операций сдвига, счета, накопления и взаимного параллельно-последовательного преобразования цифровых слов. Через вход LD/SH загружаются параллельные данные и производится их сдвиг вправо. Если на этом входе присутствует напряжение высокого уровня, через входы первого триггера J и K в регистр вводятся последовательные данные. Вход J имеет высокий активный уровень, вход K — низкий; если

эти входы соединить, получим простой D-вход. Данные сдвигаются в направлении от QA к QB, QC, а затем к QD после каждого положительного перепада на тактовом входе CLK.

Если на входе LD/SH присутствует напряжение низкого (активного) уровня, все четыре триггера регистра запускаются одним тактовым перепадом (от низкого уровня к высокому). Тогда данные от параллельных входов A...D передаются на соответствующие выходы QA...QD. Сдвиг данных влево обеспечивается в схеме, где каждый выход Qn соединен внешней перемычкой со входом D_{n+1}, т.е. схема включения на рис. 9.42 соответствует только режиму приема и хранения данных.

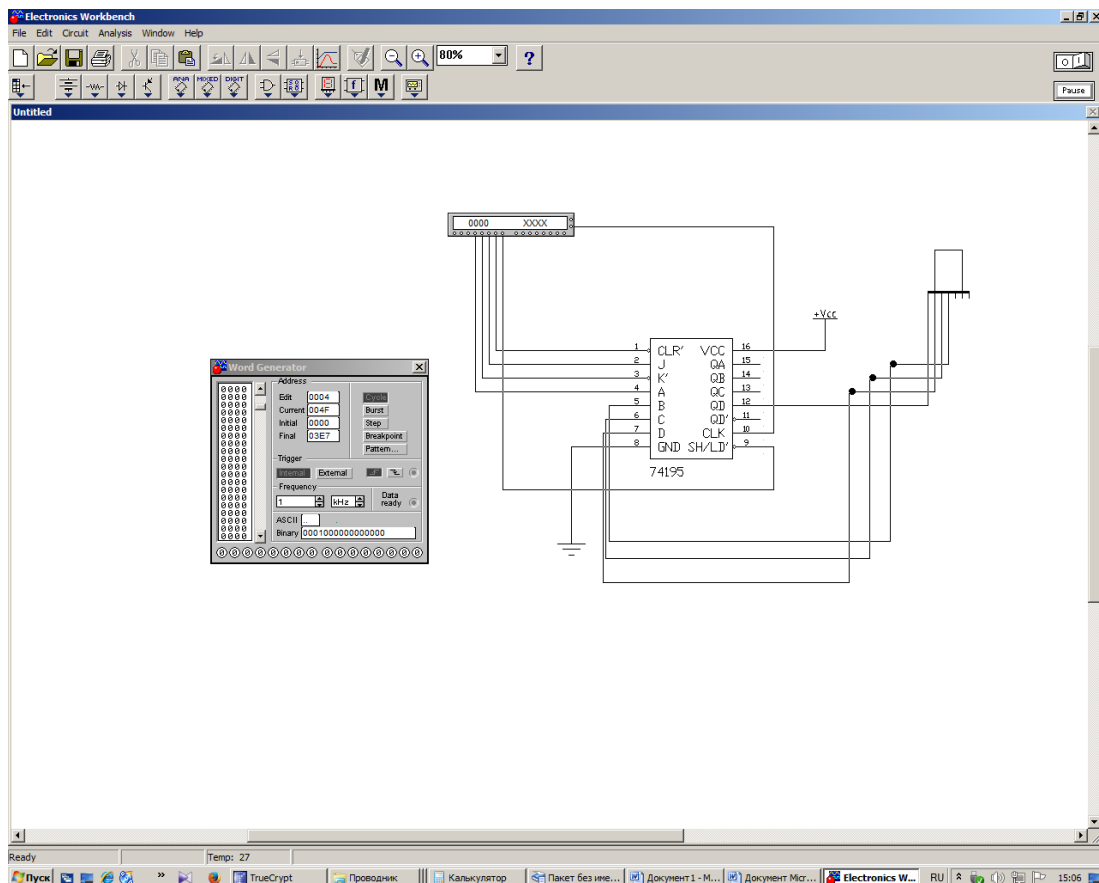


Рисунок 2 - Схема включения регистра 74195

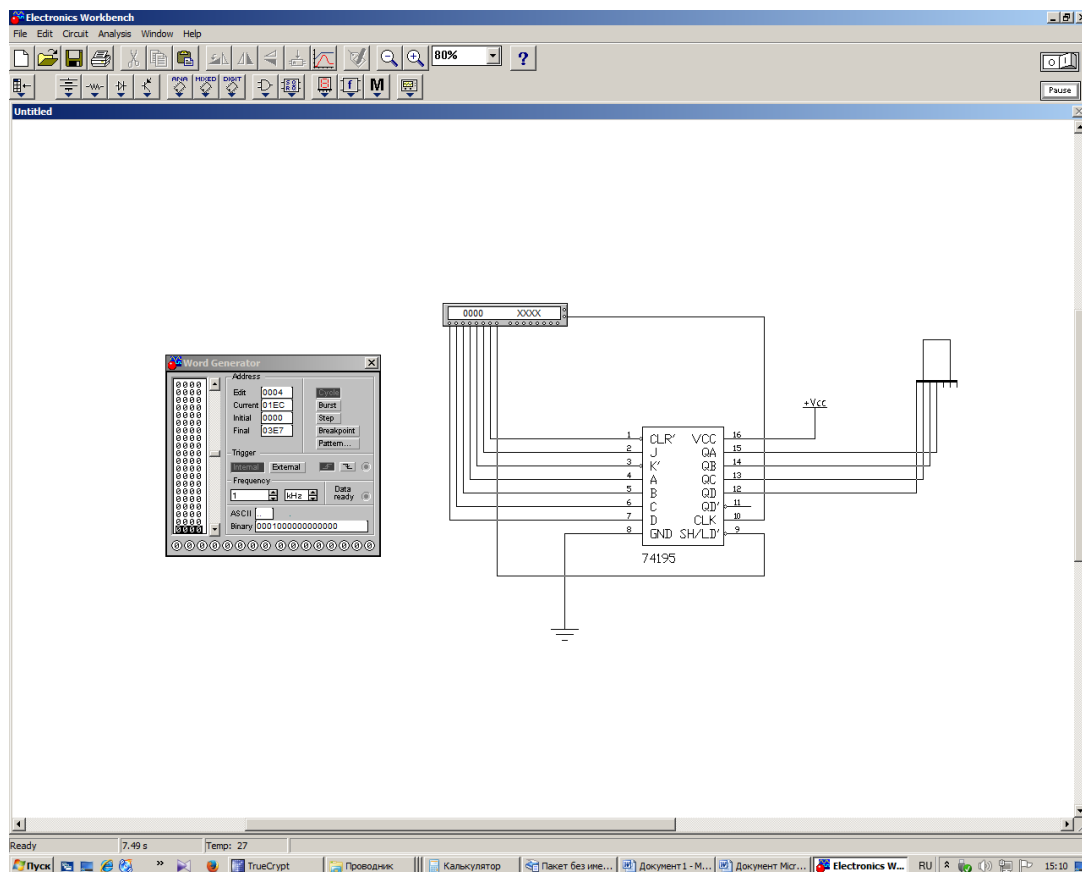


Рисунок 3 - Схема включения регистра 74195 в режиме сдвига

Схема включения ИМС в режиме сдвига и режимы работы генератора слова показаны на рисунке 3. Для режима сдвига напряжение на входе LD/SH надо зафиксировать на высоком уровне. Из-за того, что все операции в регистре ИР12 строго синхронны и запускается он фронтом импульса, логические уровни на входах J, K, Dn, LD/SH можно произвольно изменять до прихода фронта запуска. Низким уровнем на входе CLR всем выходным сигналам присваивается низкий уровень.

Напряжение низкого уровня на входе CLR означает также запрет на действие тактового импульса CLK, для правильного сброса данных надо выбрать момент, когда на входе CLK присутствует напряжение низкого уровня.

Порядок выполнения

1. Соберите схемы, изображенные на рисунках 1 — 3.
2. Проведите моделирование регистра 74133 по схеме на рисунке 1. При

моделировании необходимо выбрать с помощью генератора слова двоичные

комбинации, которые позволяют проверить все режимы его работы.

3. Составить так называемую таблицу состояния, напоминающую таблицу истинности.

4. Проведите моделирование регистра 74195 в режиме приема данных (рисунке 3).

5. Для приведенной на рисунке 4 схемы исследуйте следующие режимы сдвига:

1 — сдвиг и установка по первому каскаду ($JK=11$);

2 — сдвиг и сброс по первому каскаду ($JK=00$);

3 — сдвиг и переключение первого каскада ($JK=10$);

4 — сдвиг и хранение в первом каскаде ($JK=01$).

При этом, как указывалось выше, $CLR=1$, $LD/SH=1$, состояние входа А безразлично.

6. Составить отчет по занятию.

Содержание отчета

1. Тема, цель занятия.
2. Схемы экспериментов.
3. Результаты моделирования.
4. Выводы

Контрольные вопросы

1. Что такое регистр, какие функции он может выполнять?
2. Назовите типы регистров и их возможные применения.
3. Принцип работы параллельного регистра.
4. принцип работы последовательного регистра.

Практическое занятие №8

Исследование работы счетчиков

Цель занятия: изучение структуры и исследование работы суммирующих и вычитающих счетчиков, изучение способов изменения коэффициента пересчета счетчиков, исследование работы счетчиков с коэффициентом пересчета, отличным от 2.

Краткие теоретические сведения

Счетчик — устройство для подсчета числа входных импульсов. Число, представляемое состоянием его выходов по фронту каждого входного импульса, изменяется на единицу. Счетчик можно реализовать на нескольких триггерах. В суммирующих счетчиках каждый входной импульс увеличивает число на его выходе на единицу, в вычитающих счетчиках каждый входной импульс уменьшает это число на единицу. Наиболее простые счетчики - двоичные. На рисунке 1 представлен суммирующий двоичный счетчик и диаграммы его работы.

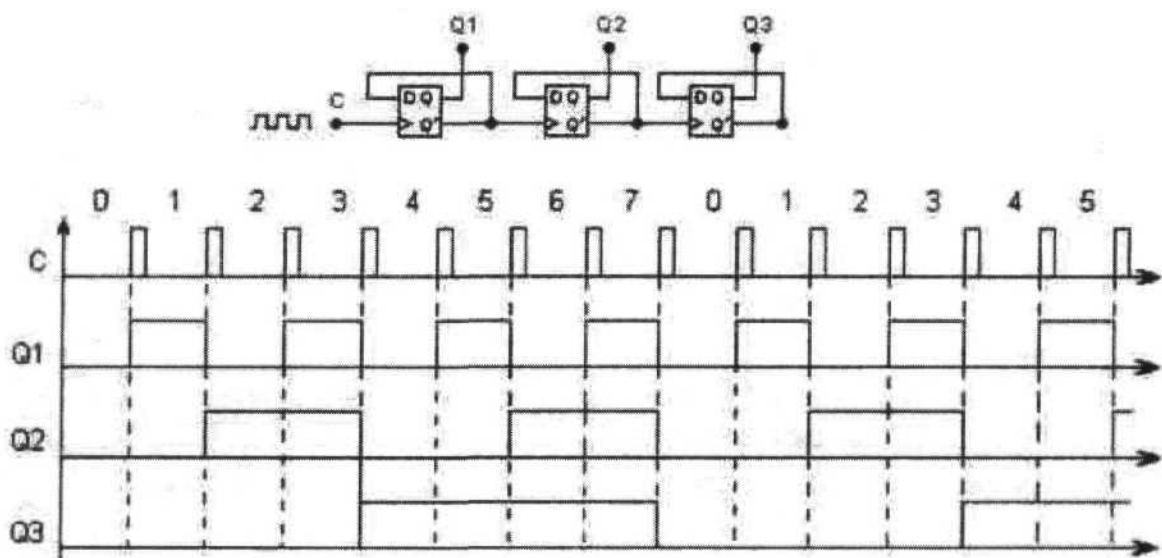


Рисунок 1

Счетчики можно реализовать на триггерах. При этом триггеры соединяют последовательно. Выход каждого триггера непосредственно действует на тактовый вход следующего. Для того, чтобы реализовать суммирующий счетчик,

необходимо счетный вход очередного триггера подключать к инверсному выходу предыдущего. Для того, чтобы изменить направление счета (реализовать вычитающий счетчик), можно предложить следующие способы:

1) Считывать выходные сигналы счетчика не с прямых, а с инверсных выходов триггеров. Число, образуемое состоянием инверсных выходов триггеров счетчика, связано с числом, образованным состоянием прямых выходов триггеров следующим соотношением:

$$N_{\text{ИП}} = 2^n - N_{\text{ИНВ}} - 1,$$

где n — разрядность выхода счетчика.

В таблице 1 приведен пример связи числа на прямых выходах с числом на инверсных выходах триггеров счетчика.

2) Изменить структуру связей в счетчике: подавать на счетный вход следующего триггера сигнал не с инверсного, а с прямого выхода предыдущего, как показано на рисунке 2. В этом случае изменяется последовательность переключения триггеров.

Таблица 1

Состояния прямых выходов			Число	Состояния инверсных выходов			Число
Q3	Q2	Q1	N	Q3'	Q2'	Q1'	N
0	0	0	0	1	1	1	7
0	0	1	1	1	1	0	6
0	1	0	2	1	0	1	5

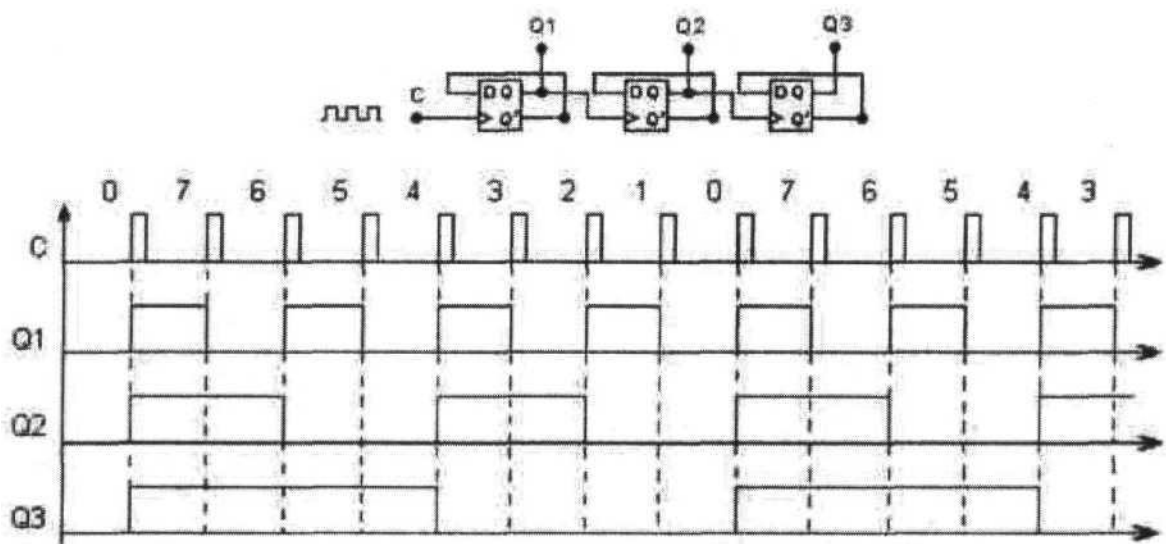


Рисунок 2

Изменение коэффициента пересчета.

Счетчики характеризуются числом состояний в течение одного периода (цикла). Для схем на рисунках 1, 2 цикл содержит $N = 2^3 = 8$ состояний (от 000 до 111). Часто число состояний называют коэффициентом пересчета, который равен отношению числа импульсов N_C на входе к числу импульсов N_{QCT} на выходе старшего разряда за период:

$$K_{сч} = N_C / N_{QCT}.$$

Если на вход счетчика подавать периодическую последовательность импульсов с частотой F_C , то частота F_Q на выходе старшего разряда счетчика будет меньше в раз:

$$K_{сч} = F_C / F_Q.$$

Поэтому счетчики также называют делителями частоты, а величину — коэффициентом деления. Для увеличения величины приходится увеличивать число триггеров в цепочке. Каждый дополнительный триггер удваивает число состояний счетчика и число . Для уменьшения коэффициента можно в качестве выхода счетчика рассматривать выходы триггеров промежуточных каскадов. Например, для счетчика на трех триггерах $K_{сч} = 8$, если взять

выход 2-го триггера, то $K_{сч} = 4$. При этом  является целой степенью числа 2: 2, 4, 8, 16 и т. д.

Можно реализовать счетчик, для которого  — любое целое число. Например, для счетчика на трех триггерах можно сделать  от 2 до 7, но при этом один или два триггера могут быть лишними. При использовании всех трех триггеров можно получить $K_{сч} = 5...7$: $2^2 < K_{сч} < 2^3$. Счетчик с $K_{сч} = 5$ должен иметь 5 состояний, которые в простейшем случае образуют последовательность: {0, 1, 2, 3, 4}. Циклическое повторение этой последовательности означает, что коэффициент деления счетчика равен 5. Для построения суммирующего счетчика с $K_{сч} = 5$ надо, чтобы после формирования последнего числа из последовательности {0, 1, 2, 3, 4} счетчик переходил не к числу 5, а к числу 0. В двоичном коде это означает, что от числа 100 нужно перейти к числу 000, а не 101. Изменение естественного порядка счета возможно при введении дополнительных связей между триггерами счетчика. Можно воспользоваться следующим способом: как только счетчик попадает в нерабочее состояние (в данном случае 101), этот факт должен быть опознан и повлечь последующую выработку сигнала, который перевел бы счетчик в состояние 000. Рассмотрим этот способ более детально. Факт попадания счетчика в нерабочее состояние описывается логическим уравнением:

$$\begin{aligned} F &= 101 \vee 110 \vee 111 = Q_3 \cdot \bar{Q}_2 \cdot Q_1 \vee Q_3 \cdot Q_2 \cdot \bar{Q}_1 \vee Q_3 \cdot Q_2 \cdot Q_1 = \\ &= Q_3 \cdot Q_1 \vee Q_3 \cdot Q_2 \end{aligned}$$

Состояния 110 и 111 также являются нерабочими и поэтому учтены при составлении уравнения. Если на выходе эквивалентной логической схемы $F = 0$, значит счетчик находится в одном из рабочих состояний: $0 \vee 1 \vee 2 \vee 3 \vee 4$. Как только он попадает в одно из нерабочих состояний $5 \vee 6 \vee 7$, формируется сигнал $F = 1$. Появление сигнала $F = 1$ должно переводить счетчик в начальное состояние 000, следовательно, этот сигнал нужно использовать для воздействия на установочные входы триггеров счетчика, которые осуществляли бы сброс счетчика в состояние $Q_1 = Q_2 = Q_3 = 0$. При реализации счетчика

на триггерах со входами установки логическим нулем для сброса триггеров требуется подать на входы сброса сигнал $R = 0$. Для обнаружения факта попадания в нерабочее состояние используем схему, реализующую функцию F и выполненную на элементах И-НЕ. Для этого преобразуем выражение для функции:

Соответствующая схемная реализация приведена на рисунке 3.

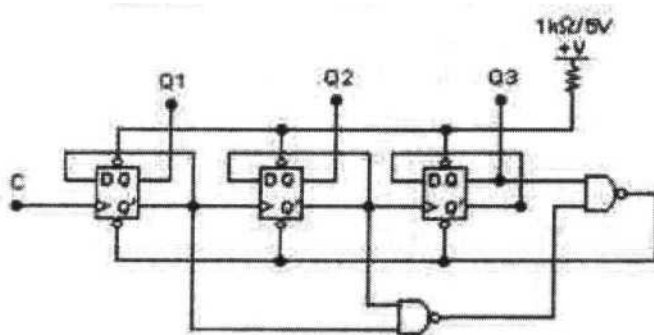


Рисунок 3

Счетчик будет работать следующим образом: при счете от 0 до 4 все происходит как в обычном суммирующем счетчике с $K_{сч} = 8$. Установочные сигналы равны 1 и естественному порядку счета не препятствуют. Счет происходит по положительному фронту импульса на счетном входе C . В тот момент, когда счетчик находится в состоянии 4 (100), следующий тактовый импульс сначала переводит счетчик в состояние 5 (101), что немедленно (задолго до прихода следующего тактового импульса) приводит к формированию сигнала сброса, который поступает на установочные входы сброса триггеров. В результате счетчик сбрасывается в 0 и ждет прихода следующего тактового импульса на счетный вход. Один цикл счета закончился, счетчик готов к началу следующего цикла.

Применяя такие схемы с обратной связью для сброса счетчика, нужно иметь в виду, что операция сброса занимает конечное время, поэтому непосредственно перед сбросом счетчика в 0 на выходе первого триггера появляются кратковременные импульсы, или «иголки». Это не имеет значения при подключении счетчика напрямую к индикатору, но при использовании этого выхода счетчика в качестве источника тактовых импульсов могут возникнуть опре-

деленные проблемы. Схема, в которой это явление устранено, приведена на рисунке 4. Важным отличием является то, что схема обнаруживает не факт попадания в нерабочее состояние 101, а факт попадания в состояние 100 и в следующем такте вырабатывает сигнал сброса.

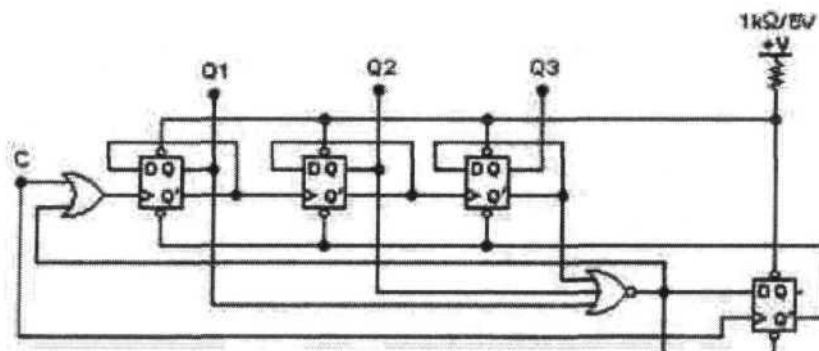


Рисунок 4

Порядок выполнения

Эксперимент 1. Исследование суммирующего счетчика.

Соберите схему, изображенную на рисунке 5.

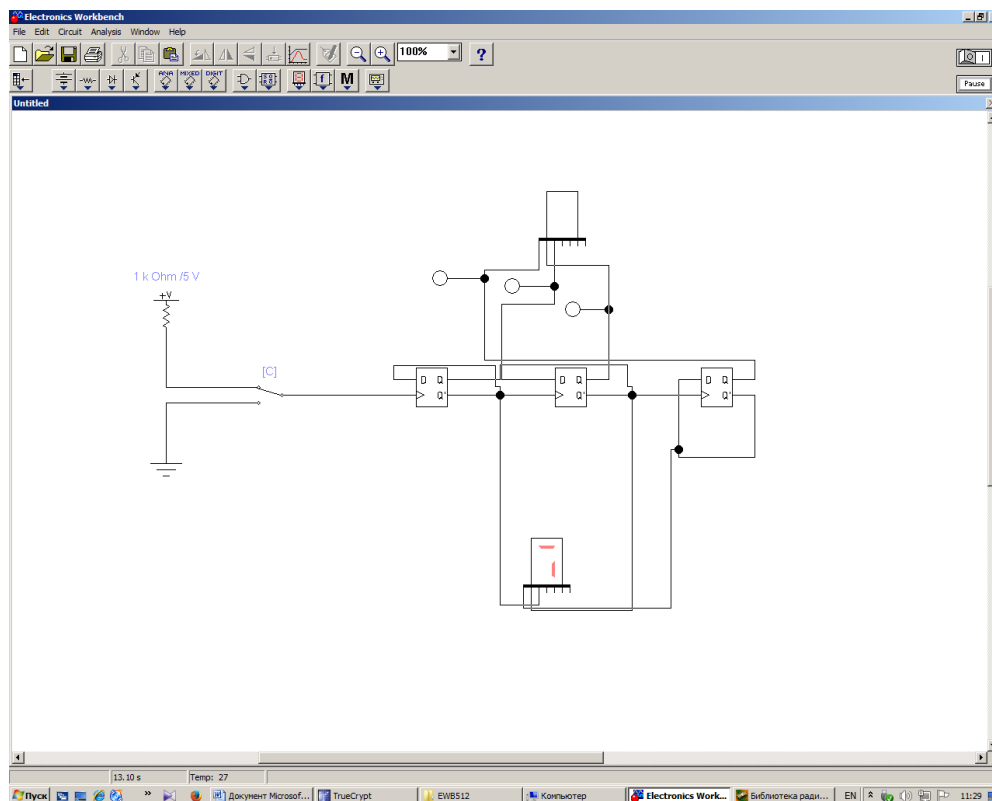


Рисунок 5

Включите схему. Подавая на вход схемы тактовые импульсы при помощи ключа *C* и, наблюдая состояние выходов счетчика при помощи логических пробников, составьте временные диаграммы работы суммирующего счетчика. Определите коэффициент пересчета счетчика. Результаты занесите в раздел «Результаты экспериментов». Обратите внимание на числа, формируемые состояниями инверсных выходов счетчика.

Эксперимент 2. Исследование вычитающего счетчика.

Соберите схему, изображенную на рисунке 6.

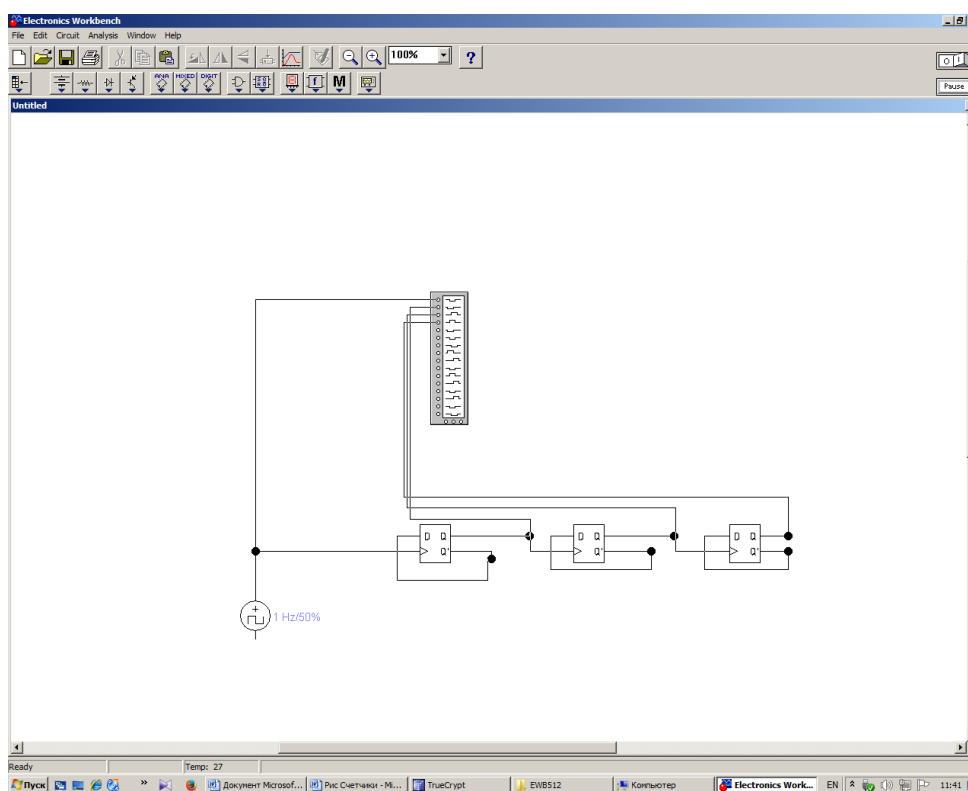


Рисунок 6

Включите схему. Зарисуйте временные диаграммы работы вычитающего счетчика в раздел «Результаты экспериментов».

В схеме на рисунке 6 входы логического анализатора подключите к инверсным входам триггеров. Включите схему. Зарисуйте полученные временные диаграммы в раздел «Результаты экспериментов» и сравните их с диаграммами, полученными в эксперименте 1.

Эксперимент 3. Исследование счетчика с измененным коэффициентом пересчета.

Соберите схему, изображенную на рисунке 7.

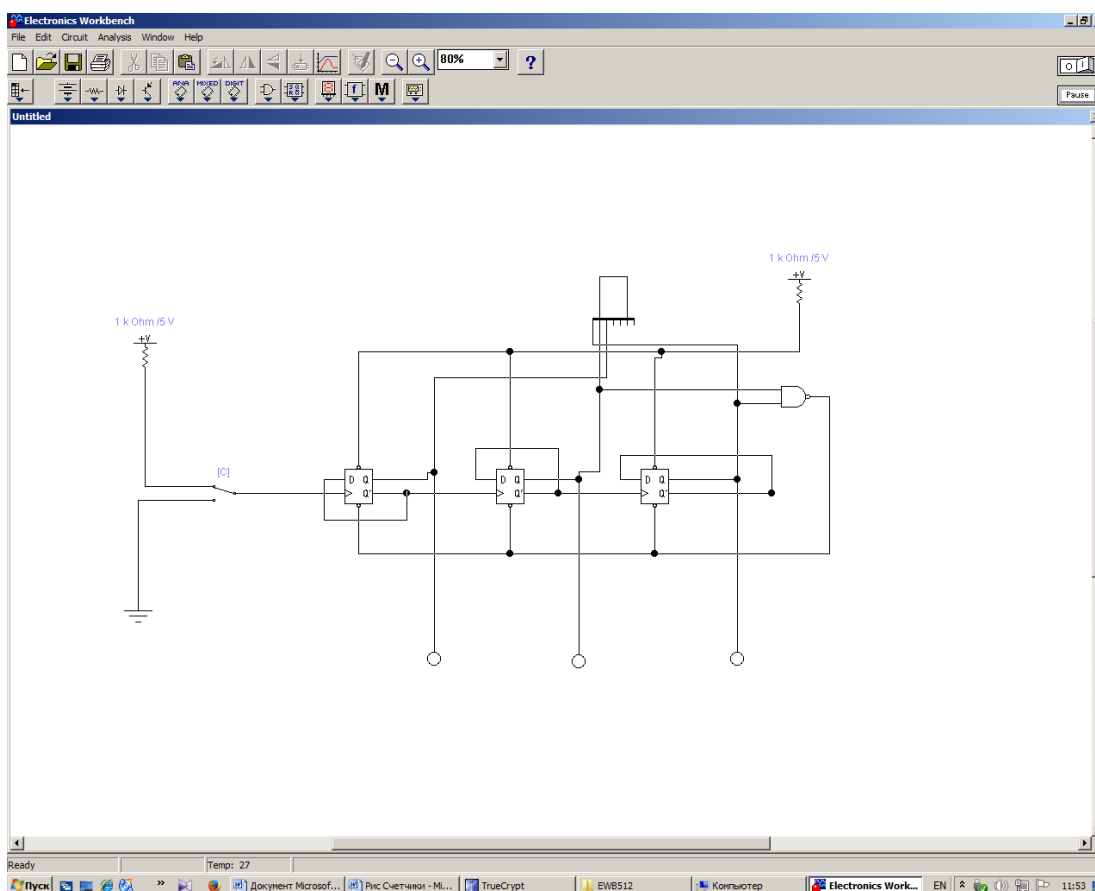


Рисунок 7

Включите схему. Подавая на вход схемы тактовые импульсы при помощи ключа C и наблюдая состояние выходов счетчика при помощи логических пробников, составьте временные диаграммы работы счетчика и определите коэффициент пересчета. Результаты занесите в раздел «Результаты экспериментов».

Измените структуру комбинационной части счетчика в соответствии со схемой на рисунке 3. Подавая на вход схемы тактовые импульсы при помощи ключа C и, наблюдая состояние выходов счетчика при помощи логических пробников, составьте временные диаграммы работы счетчика на 5. Результаты занесите в раздел «Результаты экспериментов».

Эксперимент 4. Исследование регистра Джонсона.

Соберите схему, изображенную на рисунке 8.

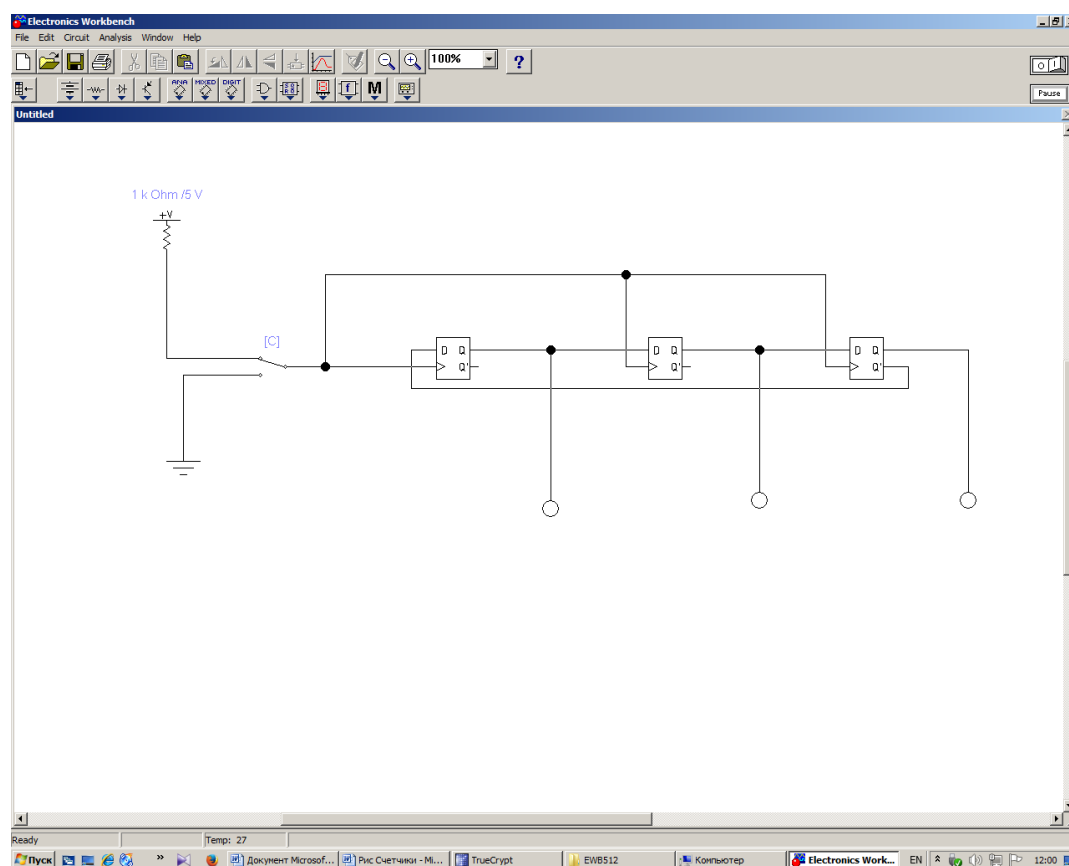


Рисунок 8

Счетное устройство, приведенное на рисунке, получило название регистра Джонсона или регистра с перекрестными связями. Включите схему. Постройте временные диаграммы сигналов на выходах триггеров. Определите коэффициент пересчета регистра Джонсона. Результаты занесите в раздел «Результаты экспериментов».

Эксперимент 5. Исследование регистра Джонсона, реализованного на JK-триггерах.

Соберите схему, изображенную на рисунке 9.

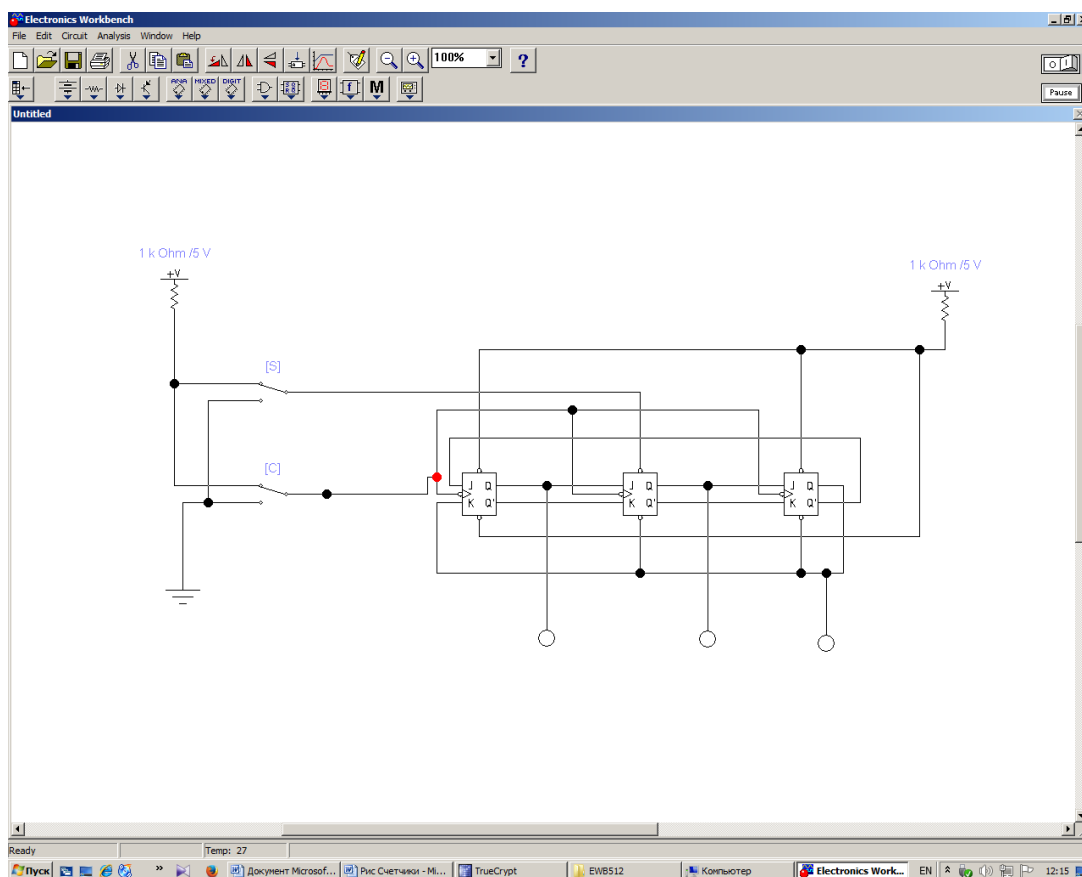


Рисунок 9

Установите ключ S в верхнее положение (на вход S второго триггера подается сигнал логической единицы). Включите схему. Постройте временные диаграммы работы схемы и занесите их в раздел «Результаты экспериментов».

Сравните полученные диаграммы с результатами эксперимента 4.

Установите схему в состояние 000. Подайте при помощи ключа S кратковременный импульс на вход S второго триггера. При этом схема должна установиться в состояние 010. Подавая на вход C схемы тактовые импульсы при помощи соответствующего ключа и наблюдая состояние выходов схемы при помощи логических пробников, составьте временные диаграммы работы устройства. Определите коэффициент пересчета схемы.

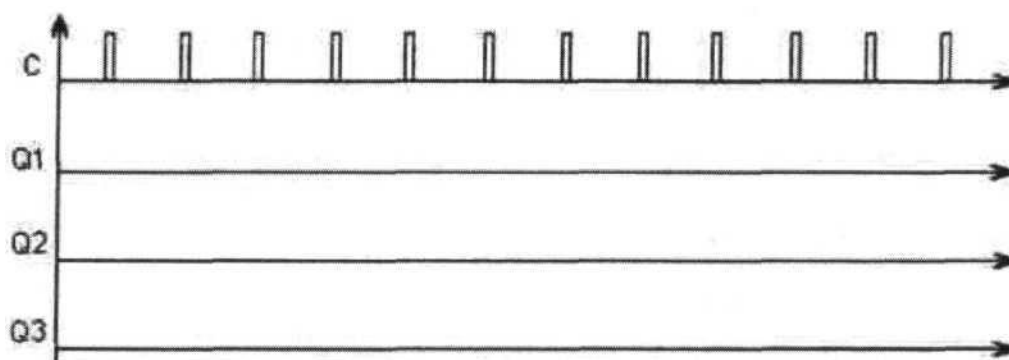
Результаты занесите в раздел «Результаты экспериментов».

Указание. Вернуть схему в прежнее состояние можно подачей кратковременного импульса на вход S второго триггера в момент, когда схема находится в состоянии 101.

Результаты экспериментов

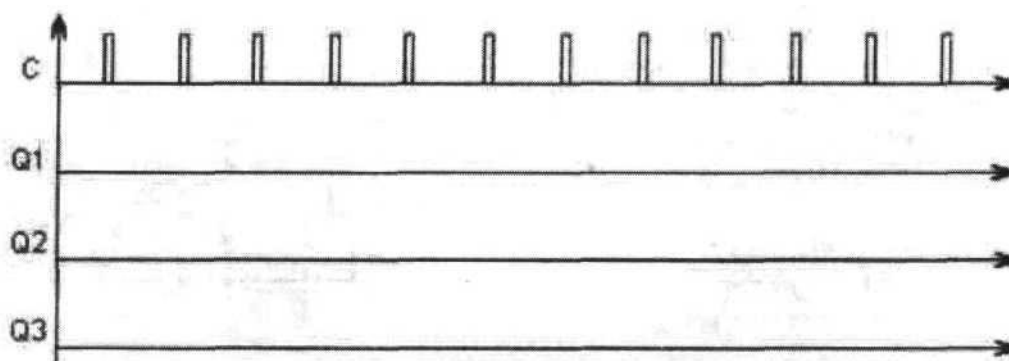
Эксперимент 1. Исследование суммирующего счетчика.

Временные диаграммы.

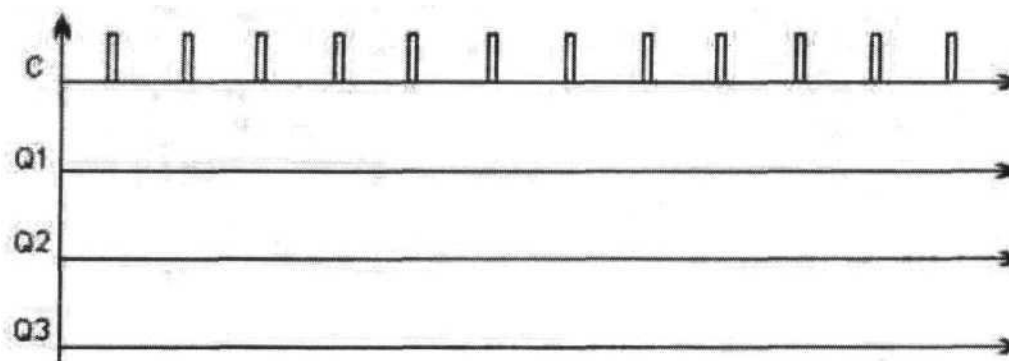


Эксперимент 2. Исследование вычитающего счетчика.

а) Временные диаграммы.

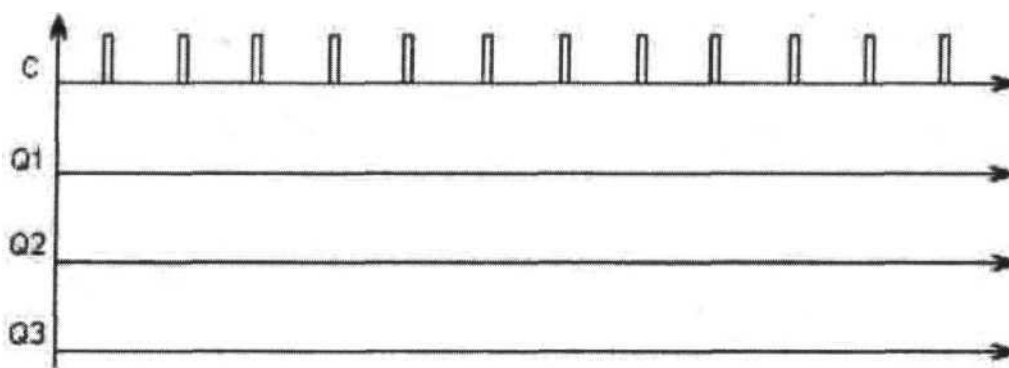


б) Временные диаграммы.

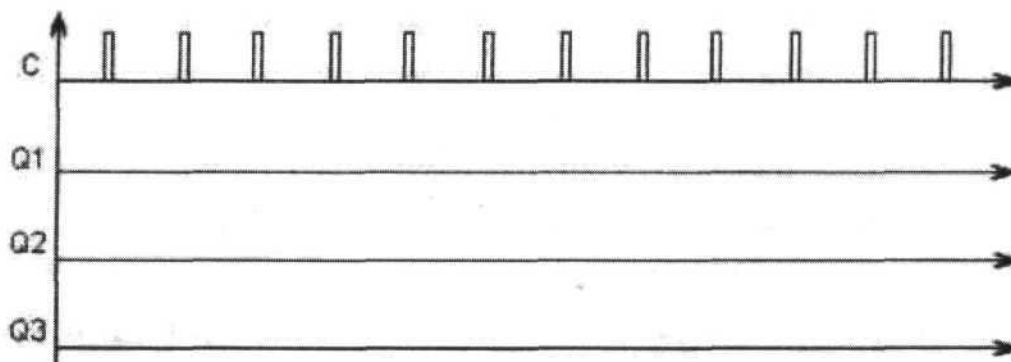


Эксперимент 3. Исследование счетчика с измененным коэффициентом пересчета.

а) Временные диаграммы.

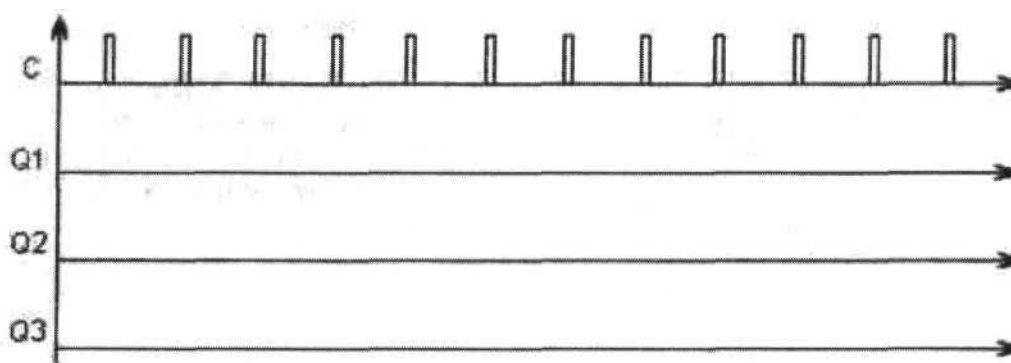


б). Временные диаграммы.



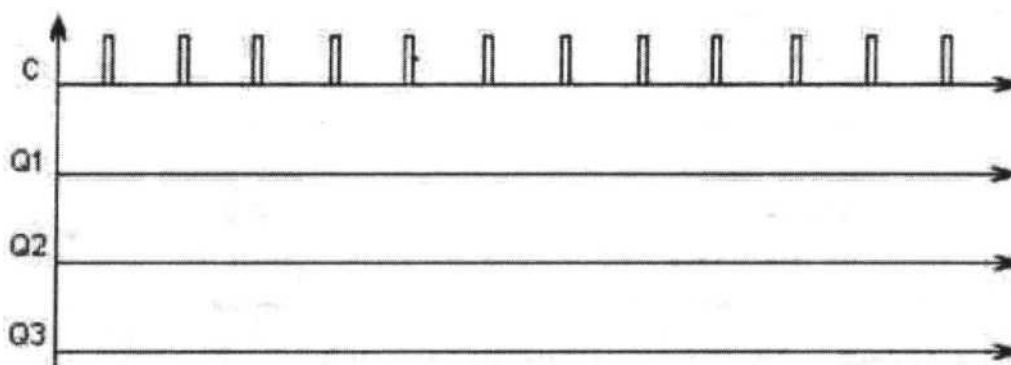
Эксперимент 4. Исследование регистра Джонсона.

Временные диаграммы.

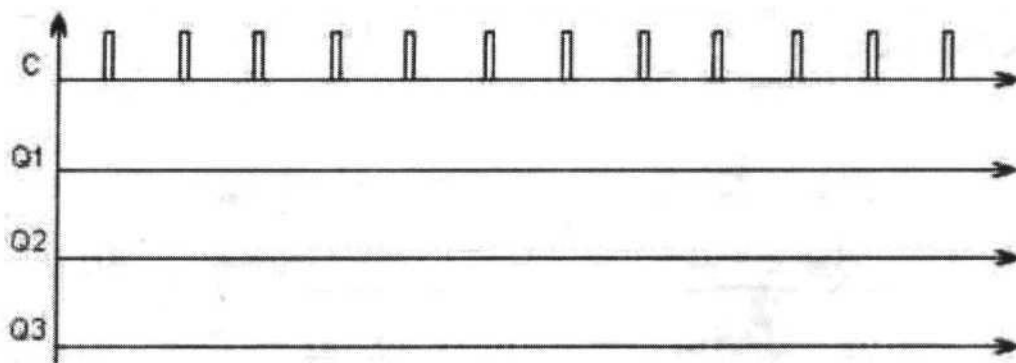


Эксперимент 5. Исследование регистра Джонсона, реализованного на JK-триггерах.

а) Временные диаграммы.



б) Временные диаграммы.



Составьте отчет по занятию.

Содержание отчета

1. Тема, цель занятия.
2. Схемы экспериментов.
3. Результаты экспериментов.
4. Выводы.

Контрольные вопросы

1. Почему при подключении счетных входов триггеров к инверсным выходам предыдущих каскадов счетчик на D-триггерах работает как суммирующий, а при подключении к прямым — как вычитающий?
2. В каком режиме будет работать счетчик на JK-триггерах при подключении счетных входов триггеров к прямым выходам предыдущих каскадов? Как изменится режим работы счетчика при подключении счетных входов триггеров к инверсным выходам?
3. Какой коэффициент пересчета имеет регистр Джонсона?
4. Какими способами можно изменить коэффициент пересчета счетчика?

5. Сколько триггеров должен содержать счетчик с коэффициентом пересчета $K_{сч} = 3, 5, 7, 9, 10, 12, 14, 15, 24, 30$?

6. В двоичном счетчике коэффициент пересчета равен 8, число триггеров — 3. При поступлении тактовых импульсов на счетный вход счетчик изменяет своё состояние в следующей последовательности:

000-001-010-011-100-101-110-111-000. Сколько триггеров в счетчике изменяют свое состояние одновременно на каждом из переходов? Действительно ли триггеры изменяют своё состояние одновременно? Как происходит переход счетчика из состояния 111 в состояние 000? Какой из триггеров первым изменит своё состояние? Что послужит причиной переключения второго триггера? Как развивается процесс изменения состояния триггеров при переходе счетчика из состояния 011 в состояние 100?

7. Цифровые часы в метро реализованы на основе счетчиков. Иногда можно

заметить, что четное число секунд на табло часов сохраняется заметно дольше, чем нечетное (возможна и обратная закономерность). Почему это происходит?

8. Какую разрядность должен иметь счетчик, отсчитывающий секунды и десятки секунд при наличии генератора импульсов частотой 10 кГц?

Практическое занятие №9

Исследование дешифраторов

Цель работы: ознакомление с принципом работы дешифраторов, исследование влияния управляющих сигналов на работу дешифраторов, реализация и исследование функциональных модулей на основе дешифраторов.

Краткие теоретические сведения

Комбинационной схемой называется логическая схема, реализующая однозначное соответствие между значениями входных и выходных сигналов. Для реализации комбинационных схем используются логические элементы, выпускаемые в виде интегральных схем. В этот класс входят интегральные схемы дешифраторов, шифраторов, мультиплексоров, демультиплексоров, сумматоров.

В практическом занятии для удобства вместо схемы дешифратора используется схема демультиплексора, это возможно благодаря сходству алгоритмов работы.

Дешифратор — логическая комбинационная схема, которая имеет n информационных входов и 2^n выходов. Каждой комбинации логических уровней на входах будет соответствовать активный уровень на одном из 2^n выходов. Обычно n равно 2, 3 или 4. На рисунке 1 изображен дешифратор с $n = 3$, активным уровнем является уровень логического нуля.



Рисунок 1

На входы C, B, A можно подать следующие комбинации логических уровней: 000, 001, 010...111, всего 8 комбинаций. Схема имеет 8 выходов, на одном из которых формируется низкий потенциал, на остальных - высокий. Номер этого единственного выхода, на котором формируется активный (нулевой) уровень, соответствует числу N , определяемому состоянием входов C, B, A следующим образом: $N = C2^2 + B2^1 + A2^0$ град. Например, если на входы подана комбинация логических уровней 011, то из восьми выходов микросхемы ($Y_0, Y_1, \dots, Y_7$) на выходе с номером $n = 3$ установится нулевой уровень сигнала ($Y_3=0$), а все остальные выходы будут иметь уровень логической единицы.

Этот принцип формирования выходного сигнала можно описать следующим образом:

$$Y_i = \begin{cases} 0, & \text{если } i = k; \\ 1, & \text{если } i \neq k; \\ k = 2^2 \cdot C + 2^1 \cdot B + 2^0 \cdot A. \end{cases}$$

Видно, что уровень сигнала на выходе Y_3 описывается выражением:

$$Y_3 = \overline{\overline{C} \cdot \overline{B} \cdot \overline{A}} = 0.$$

В таком же виде можно записать выражения для каждого выхода дешифратора:

$$\begin{aligned} Y_0 &= \overline{\overline{C} \cdot \overline{B} \cdot \overline{A}}, Y_1 = \overline{\overline{C} \cdot \overline{B} \cdot A}, Y_2 = \overline{\overline{C} \cdot B \cdot \overline{A}}, Y_3 = \overline{\overline{C} \cdot B \cdot A}, Y_4 = \overline{C \cdot \overline{B} \cdot \overline{A}}, \\ Y_5 &= \overline{C \cdot \overline{B} \cdot A}, Y_6 = \overline{C \cdot B \cdot \overline{A}}, Y_7 = \overline{C \cdot B \cdot A}. \end{aligned}$$

Помимо информационных входов A, B, C дешифраторы обычно имеют дополнительные входы управления G . Сигналы на этих входах, например, разрешают функционирование дешифратора или переводят его в пассивное состояние, при котором, независимо от сигналов на информационных входах, на всех выходах установится уровень логической единицы. Можно сказать, что существует некоторая функция разрешения, значение которой определяется состояниями управляющих входов. Разрешающий вход дешифратора может быть прямым или инверсным. У дешифраторов с прямым разрешающим входом активным уровнем является уровень логической единицы, у дешифраторов с инверсным входом — уровень логического нуля. На рисунке 1 представлен дешифратор с одним инверсным входом управления. Принцип формирования выходного сигнала в этом дешифраторе с учетом сигнала управления описывается следующим образом:

$$Y_i = \begin{cases} 1 \cdot G, & \text{если } i = k; \\ 1, & \text{если } i \neq k; \\ k = 2^2 \cdot C + 2^1 \cdot B + 2^0 \cdot A. \end{cases}$$

У дешифратора с несколькими входами управления функция разрешения, как правило, представляет собой логическое произведение всех разрешающих

сигналов управления. Например, для дешифратора 74138 с одним прямым входом управления G1 и двумя инверсными G2A и G2B (рисунок 2) функции выхода Y_i и разрешения G имеют вид:

$$Y_i = \begin{cases} 1 \cdot \bar{G}, & \text{если } i = k; \\ 1, & \text{если } i \neq k; \\ k = 2^2 \cdot C + 2^1 \cdot B + 2^0 \cdot A. \end{cases}$$

$$G = G1 \cdot \overline{G2A} \cdot \overline{G2B}.$$



Рисунок 2.

Обычно входы управления используются для каскадирования (увеличения разрядности) дешифраторов или при параллельной работе нескольких схем на общие выходные линии.

Использование дешифратора в качестве демультиплексора.

Дешифратор может быть использован и как демультиплексор — логический коммутатор, подключающий входной сигнал к одному из выходов. В этом случае функцию информационного входа выполняет один из входов разрешения, а состояние входов C, B и A задает номер выхода, на который передается сигнал со входа разрешения.

Порядок выполнения

Эксперимент 1. Исследование принципа работы дешифратора 3x8 в основном режиме.

1. Постройте схему, изображенную на рисунке 3.

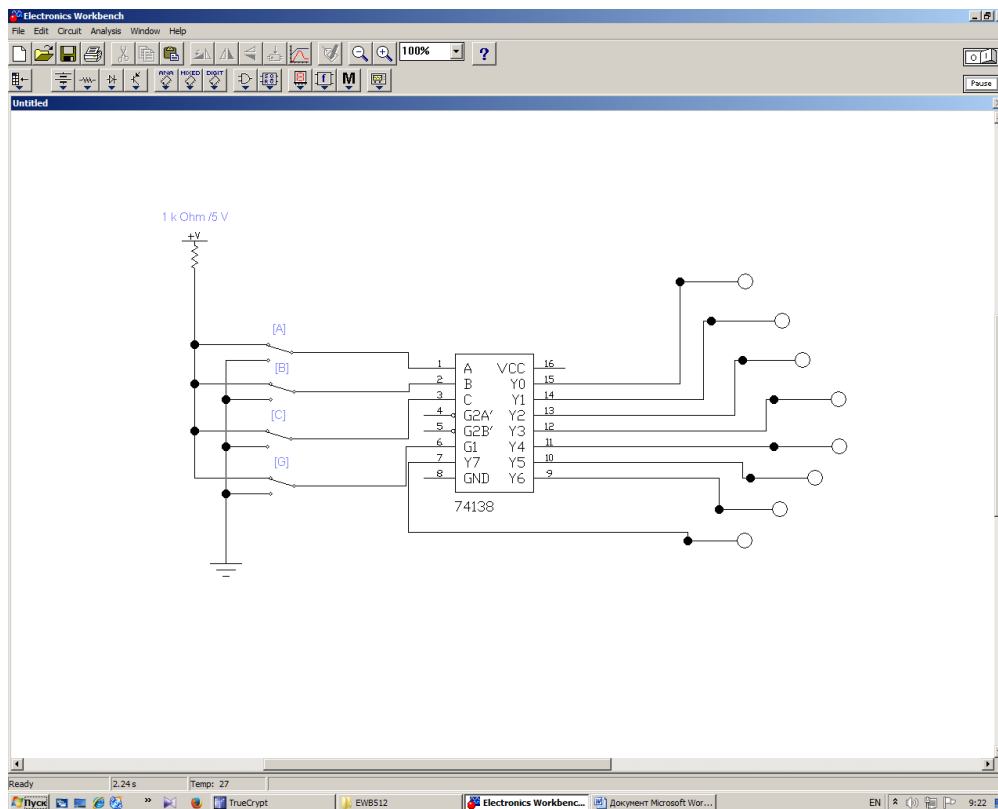


Рисунок 3

2. Включите схему.
3. Подайте на вход G уровень логической единицы. Для этого клавишей G ключ G установить в верхнее положение.
4. Определите и запишите уровни сигналов на выходах Y0...Y7 в таблицу истинности при $G = 1$ (таблица 1 в разделе «Результаты экспериментов»).
5. Подайте на вход G уровень логического нуля (ключ G установите в нижнее положение). Убедитесь, что дешифратор перешел в рабочий режим и на одном из выходов установился уровень логического нуля.
6. Подавая все возможные комбинации уровней логических сигналов на входы A, B, C с помощью одноименных ключей и определяя с помощью логических пробников уровни логических сигналов на выходе схемы, заполните таблицу истинности дешифратора при $G=0$ (таблица .1. в разделе «Результаты экспериментов»).

Эксперимент 2. Исследование принципа работы дешифратора 3х8 в режиме 2х4.

1. В схеме на рисунке 3 подключите вход С к общему проводу (земле), задав $C=0$ (рисунок 4).

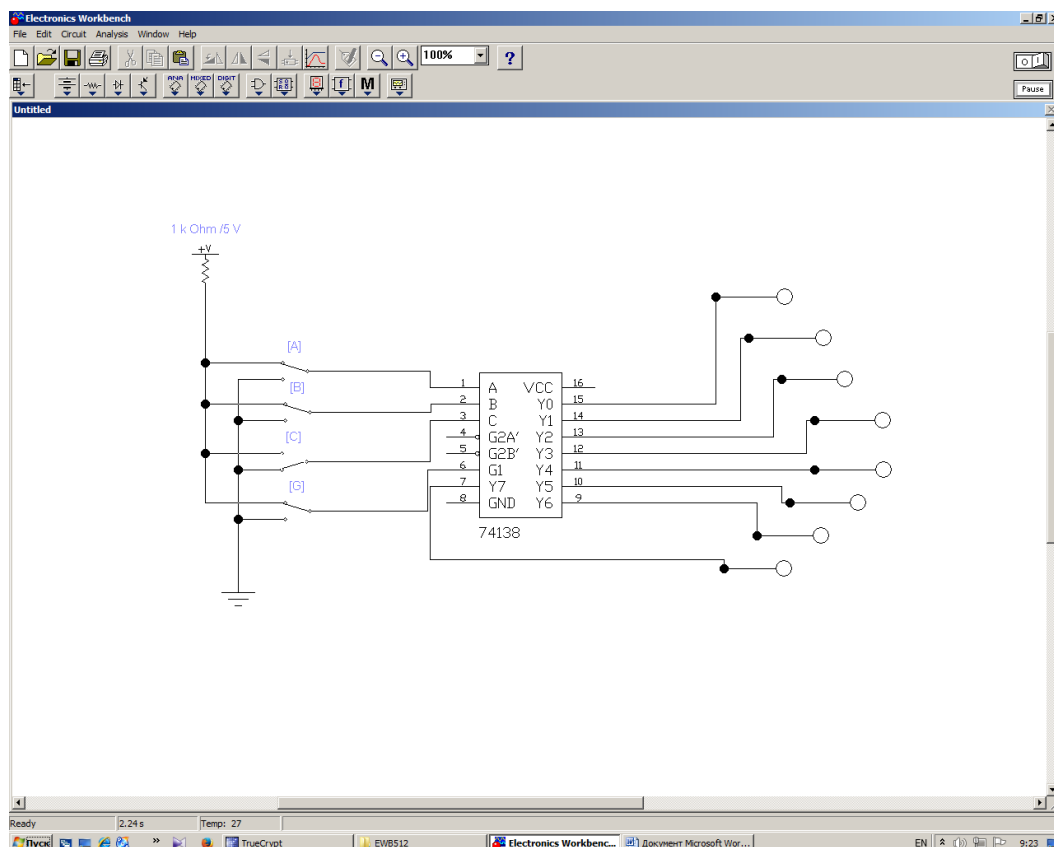


Рисунок 4

2. Изменяя уровни сигналов на входах В и А и наблюдая уровни сигналов на выходах схемы, с помощью пробников заполните таблицу истинности дешифратора (таблица 2 в разделе «Результаты экспериментов»).

3. Укажите выходы, на которых уровень сигнала не меняется.

4. Прodelайте пункт 1 при $C=1$, для чего вход С подключите к источнику логической единицы. Заполните таблицу истинности дешифратора (таблица 3 в разделе «Результаты экспериментов»).

5. Прodelайте пункт 1, заземлив вход В ($B=0$), а на входы А и С подавая все возможные комбинации логических уровней. Заполните таблицу истинности (таблица 4 в разделе «Результаты экспериментов»), там же укажите номера выходов, на которых уровень логического сигнала не изменяется.

Эксперимент 3. Исследование работы дешифратора в качестве демультиплексора.

1. Соберите схему, изображенную на рисунке 5.

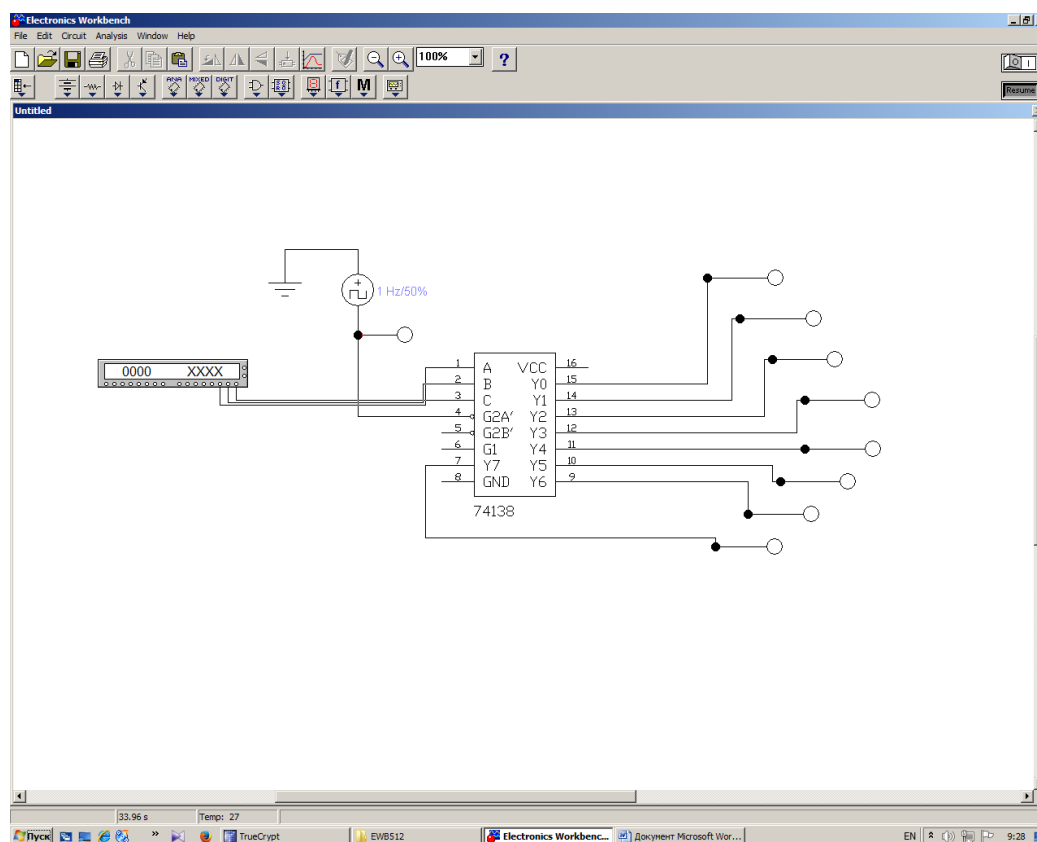


Рисунок 5

2. Включите схему.

3. В пошаговом режиме работы генератора слов подайте на входы С, В, А демультиплексора слова, эквивалентные числам от 0 до 7. Наблюдая при помощи логических пробников уровни сигналов на выходах, заполните таблицу функционирования (таблица 5 в разделе «Результаты экспериментов»). Убедитесь, что изменяющийся сигнал на входе G поочередно появляется на выходах дешифратора.

Эксперимент 4. Исследование дешифратора 3x8 с логической схемой на выходе.

1. Соберите схему, изображенную на рисунке 6.

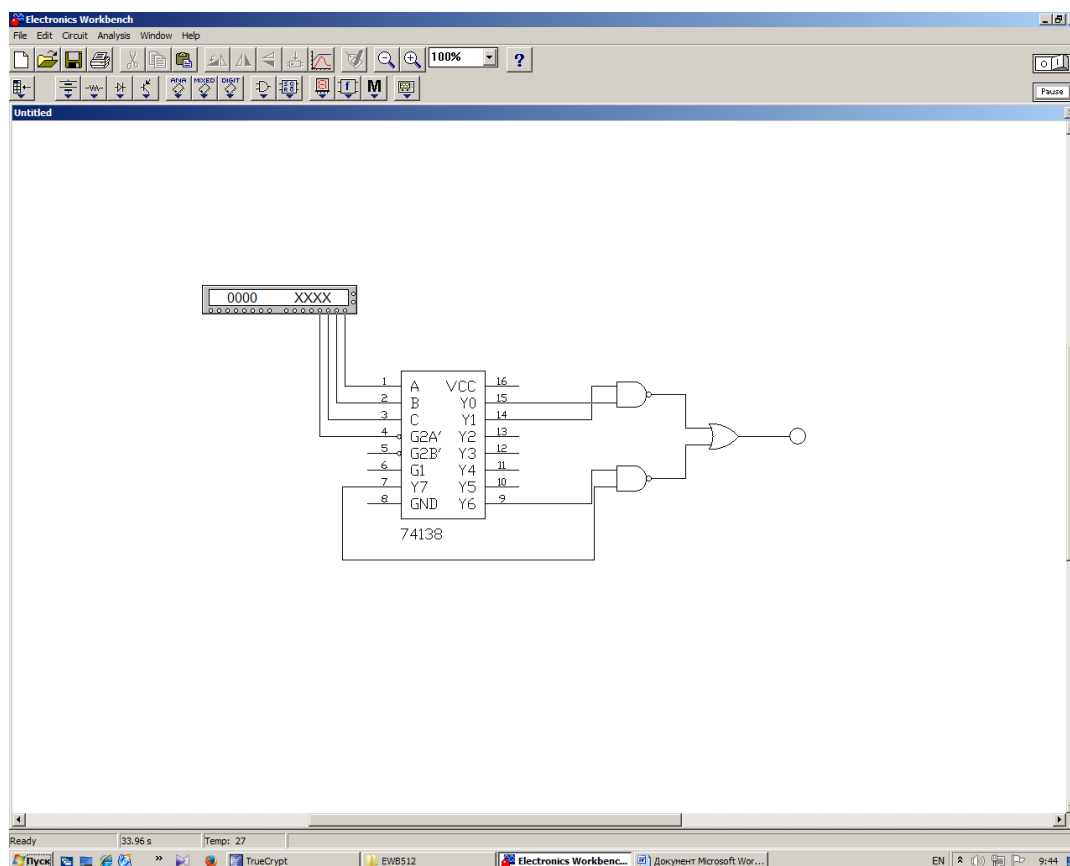


Рисунок 6

2. Включите схему.
3. Установите генератор слов в пошаговый режим.
4. Последовательно подавая слова от генератора на вход схемы и, наблюдая уровень логического сигнала на выходе схемы, с помощью логического пробника, составьте таблицу истинности функции F , реализуемой схемой на выходе (таблица 6 в разделе «Результаты экспериментов»).
5. По таблице запишите аналитическое выражение функции и занесите полученное выражение в раздел «Результаты экспериментов».

Эксперимент 5. Исследование микросхемы 74138.

1. Соберите схему, изображенную на рисунке 7.

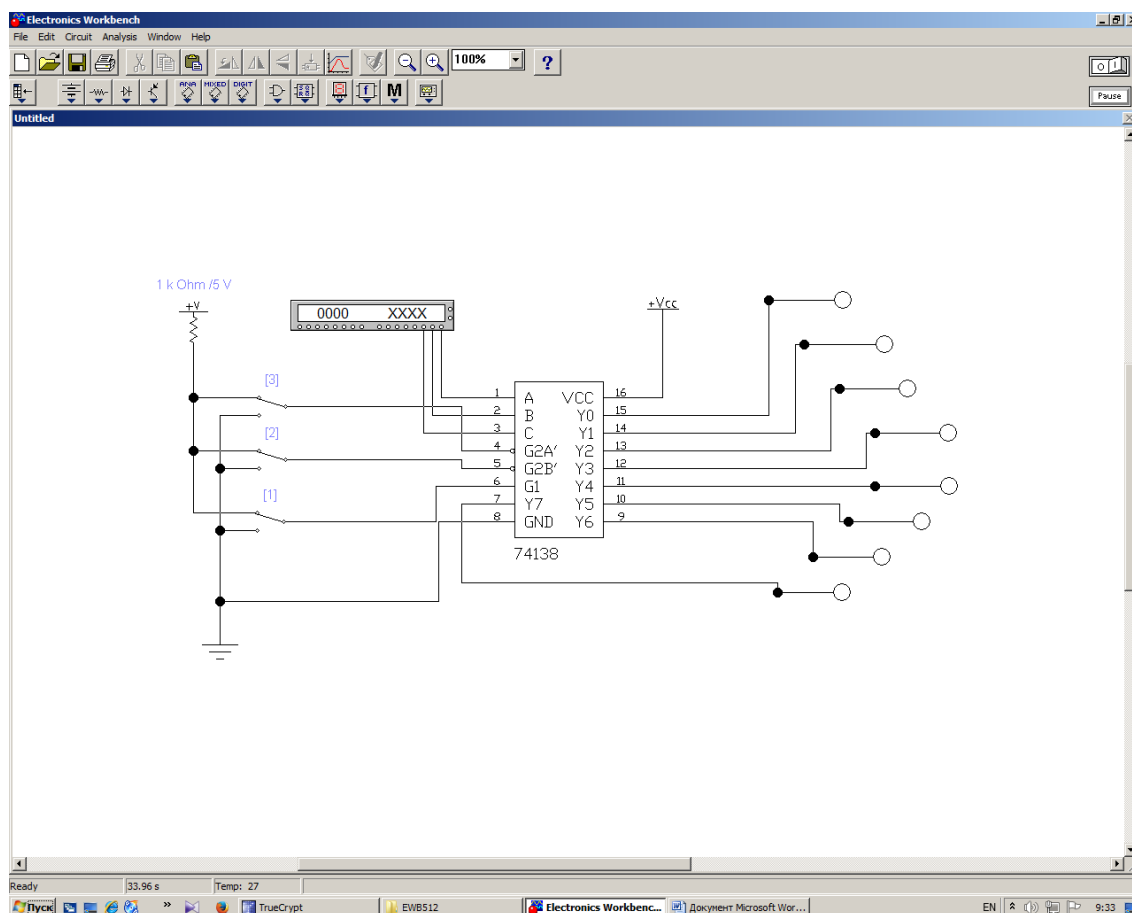


Рисунок 7

2. Установите генератор слов в пошаговый режим.
3. Включите схему.
4. С помощью соответствующих ключей установите состояние управляющих входов $G1=0$, $G2A=G2B=1$. Подавая на входы A, B, C слова от генератора слов и наблюдая состояние выходов с помощью логических пробников, заполните таблицу функционирования дешифратора 74138 (таблица 7 в разделе «Результаты экспериментов»).
5. Повторите операции пункта а) при $G1=G2A=1$, $G2B=0$.
6. Заполните таблицу функционирования дешифратора 74138 (таблица 8 в разделе «Результаты экспериментов»).
7. Повторите операции пункта а) при $G1=1$, $G2A=G2B=0$.
8. Заполните таблицу функционирования дешифратора 74138 (таблица 9 в разделе «Результаты экспериментов»).

Эксперимент 6. Исследование микросхемы 74138 с помощью логического анализатора.

1. Соберите схему, изображенную на рисунке 8.

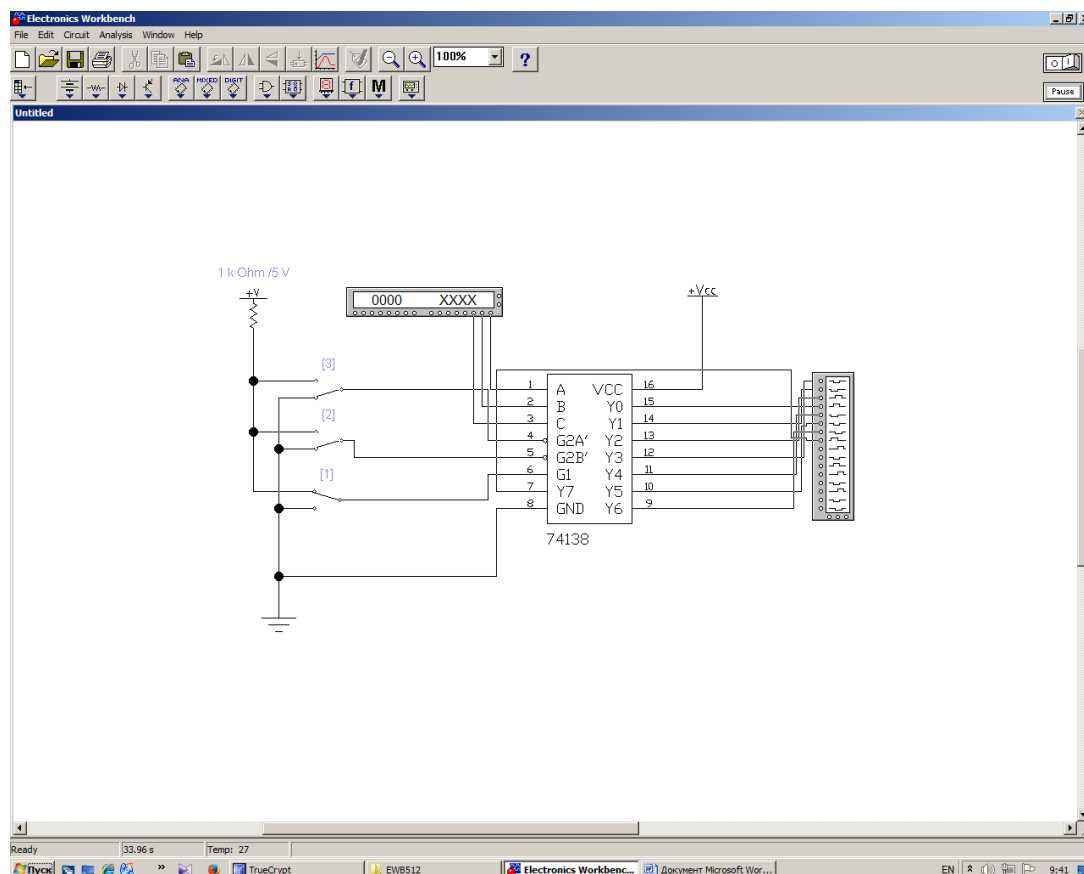


Рисунок 8

2. Установите генератор слов в пошаговый режим.
3. Включите схему.
4. С помощью соответствующих ключей установите состояние управляющих входов $G1=1$, $G2A=G2B=0$.
5. Подавая слова от генератора слов, получите временные диаграммы работы дешифратора на экране логического анализатора и зарисуйте их в разделе «Результаты экспериментов».
6. Сопоставьте временные диаграммы с таблицей 9.

Результаты экспериментов

Эксперимент 1. Исследование работы дешифратора 3x8 в основном режиме.

Таблица 1

С	В	А	G	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
0	0	0	1								
0	1	1	1								
0	0	1	0								
0	1	0	0								
0	1	1	0								
1	0	0	0								
1	0	1	0								
1	1	0	0								
1	1	1	0								

Для простоты заполнения таблицы истинности в ней можно отмечать только выводы с низким уровнем сигнала.

Эксперимент 2. Исследование дешифратора 3x8 в режиме 2x4.

Таблица 2

С	В	А	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
0	0	0								
0	0	1								
0	1	0								
0	0	1								

Выводы, которых уровень сигнала не изменяется _____

Таблица 3

С	В	А	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
1	0	0								
1	0	1								
1	1	0								
1	0	1								

Выводы, на которых уровень сигнала не изменяется _____

Таблица 4

С	В	А	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
0	0	0								
0	0	1								
1	0	0								
1	0	1								

Выводы, на которых уровень сигнала не изменяется _____

Эксперимент 3. Исследование работы дешифратора в качестве демультимплексора.

Таблица 5

С	В	А	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
0	0	0								
0	0	1								
0	1	0								
0	0	1								
1	0	1								
1	0	1								
1	0	1								
1	0	1								

Выводы с изменяющимся сигналом отмечать в клетках таблицы как G.

Эксперимент 4. Исследование дешифратора 3х8 с логической схемой на выходе.

Таблица 6

Аналитическое выражение для функции

--

Эксперимент 5. Исследование микросхемы 74138.

Таблица 7

С	В	А	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
0	0	0								
0	0	1								
0	1	0								
0	1	1								
1	0	0								
1	0	1								
1	1	0								
1	1	1								

Таблица 8

С	В	А	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
0	0	0								
0	0	1								
0	1	0								
0	1	1								
1	0	0								
1	0	1								
1	1	0								
1	1	1								

Таблица 9

С	В	А	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
0	0	0								
0	0	1								
0	1	0								
0	1	1								
1	0	0								
1	0	1								
1	1	0								
1	1	1								

Эксперимент 6. Исследование микросхемы 74138 с помощью логического анализатора.

Диаграммы на экране логического анализатора (рисунок 9).

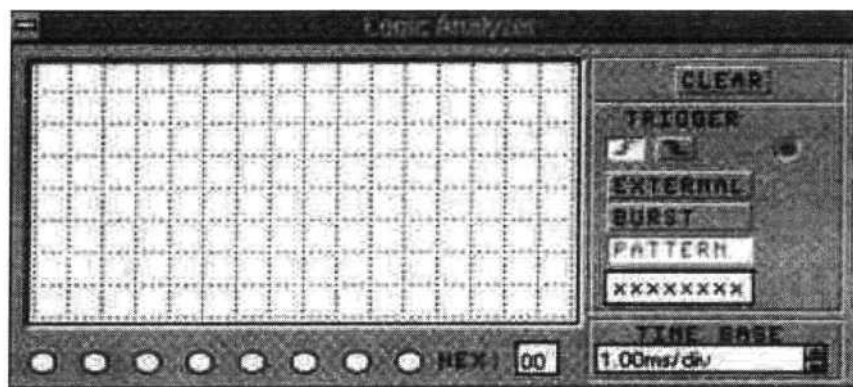


Рисунок 9

Содержание отчета

1. Тема, цель занятия.
2. Схемы экспериментов.
3. Результаты экспериментов.
4. Выводы.

Контрольные вопросы

1. Какие логические функции выполняет дешифратор?
2. Каково назначение входов управления в дешифраторе, как влияет сигнал управления на выходные функции дешифратора?
3. Какие дополнительные логические элементы необходимы для реализации логических функции n аргументов на основе дешифратора с прямыми выходами? А с инверсными?
4. Как выглядит схема дешифратора 2×4 , выполненная в базисе И, ИЛИ, НЕ? Входы дешифратора A, B , выходы Y_0, Y_1, Y_2, Y_3 . Сколько элементов каждого типа для этого требуется?
5. Как надо видоизменить схему дешифратора 2×4 в предыдущем случае, чтобы оснастить её прямым управляющим входом? Инверсным? Обозначьте входы дешифратора A, B , управляющий вход G или $\bar{G}$, выходы Y_0, Y_1, Y_2, Y_3 .

6. Как из двух дешифраторов 2х4 сделать один дешифратор 3х8?
7. Как на основе нескольких дешифраторов 2х4 с управляющим входом сделать дешифратор 4х16? Сколько дешифраторов 2х4 потребуется для решения этой задачи, если не использовать другие элементы?
8. Как на основе дешифратора 2х4 сделать схему, фиксирующую совпадение двух бит ($A=B=1$, $A=B=0$) и реализующую функцию $F = AB \vee \overline{AB}$?
9. Как на основе дешифратора сделать логическую схему, реализующую функцию

Практическое занятие №10

Исследование мультиплексоров

Цель работы: ознакомление с принципом работы мультиплексора, реализация и исследование функциональных модулей на основе мультиплексоров.

Краткие теоретические сведения

Мультиплексор — комбинационная логическая схема, представляющая собой управляемый переключатель, который подключает к выходу один из информационных входов данных. Номер подключаемого входа равен числу (адресу), определяемому комбинацией логических уровней на входах управления. Кроме информационных и управляющих входов, схемы мультиплексоров содержат вход разрешения, при подаче на который активного уровня мультиплексор переходит в активное состояние. При подаче на вход разрешения пассивного уровня мультиплексор перейдет в пассивное состояние, для которого сигнал на выходе сохраняет постоянное значение независимо от значений информационных и управляющих сигналов. Число информационных входов у мультиплексоров обычно 2, 4, 8 или 16. На рисунке 1 представлен мультиплексор 8х1 с

инверсным входом разрешения G , прямым Y и инверсным W -выходами ($W = \bar{Y}$).

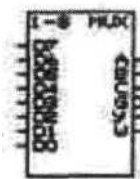


Рисунок 1

Уравнение мультиплексора. Функционирование мультиплексора, представленного на рисунке 1, описывается характеристическим уравнением, связывающим сигнал на выходе (Y) с разрешающим (G), входными информационными ($D0...D7$) и управляющими (A, B, C) сигналами:

$$Y = \left(\bar{C} \cdot \bar{B} \cdot \bar{A} \cdot D0 \vee \bar{C} \cdot \bar{B} \cdot A \cdot D1 \vee \bar{C} \cdot B \cdot \bar{A} \cdot D2 \vee \bar{C} \cdot B \cdot A \cdot D3 \vee \right. \\ \left. \vee C \cdot \bar{B} \cdot \bar{A} \cdot D4 \vee C \cdot \bar{B} \cdot A \cdot D5 \vee C \cdot B \cdot \bar{A} \cdot D6 \vee C \cdot B \cdot A \cdot D7 \right) \cdot \bar{G}.$$

Как видно из уравнения, на мультиплексоре можно реализовать логические функции, для чего нужно определить, какие сигналы и логические константы следует подавать на входы мультиплексора.

Реализация заданной функции с помощью мультиплексора. Логическая функция n переменных определена для 2^n комбинация значений переменных. Это позволяет реализовать функцию n -переменных на мультиплексоре, имеющем n -управляющих и 2^n информационных входов. В этом случае каждой комбинации значений аргументов соответствует единственный информационный вход мультиплексора, на который подается значение функции. Например, требуется реализовать функцию

Эта функция определена только для 8 комбинаций значений переменных, поэтому для её реализации можно использовать мультиплексор 8×1 с тремя управляющими входами. Составим таблицу истинности функции (таблица 1).

Таблица 1

N	c	b	a	F1
0	0	0	0	1
1	0	0	1	0
2	0	1	0	0
3	0	1	1	1
4	1	0	0	0
5	1	0	1	0
6	1	1	0	1
7	1	1	1	1

Из таблицы видно, что для реализации функции на мультиплексоре необходимо подать на информационный вход мультиплексора с номером N сигнал, значение которого равно соответствующему значению функции $F1$, т. е. на входы с номерами 1, 2, 4, 5 следует подать уровень логического нуля, а на остальные — уровень логической единицы. Таким образом, при подаче комбинации логических уровней на управляющие входы мультиплексора, к его выходу подключится вход, значение сигнала на котором равно соответствующему значению функции. Схемная реализация приведена на рисунке 2.

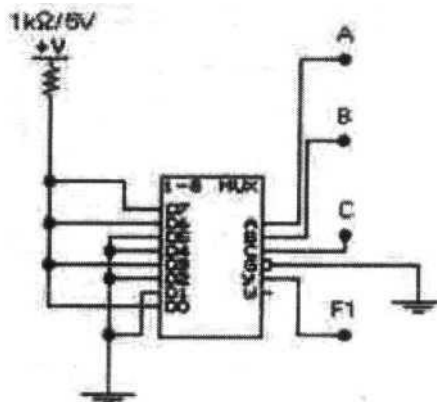


Рисунок 2

При реализации логических функций на информационные входы можно подавать не только константы, но и изменяющиеся входные сигналы. Так, например, рассмотрим другой способ реализации функции $F1$, рассмотренной

выше. Для этого минимизируем выражение функции:

x

Составим таблицу истинности функции в зависимости от значений переменных a и b (таблица 2).

Таблица 2

N	b	a	F1
0	0	0	c
1	0	1	0
2	1	0	c
3	1	1	1

Заданную такой таблицей функцию реализуют, как и в предыдущем случае, подав на вход с номером N сигнал, значение которого соответствует значению функции F1. В данном случае сигналы c и c' , соответствующие переменной c , подаются на информационные входы, как указано в таблице истинности. При этом сокращается число управляющих входов. Схемная реализация такого способа задания функции представлена на рисунке 3.

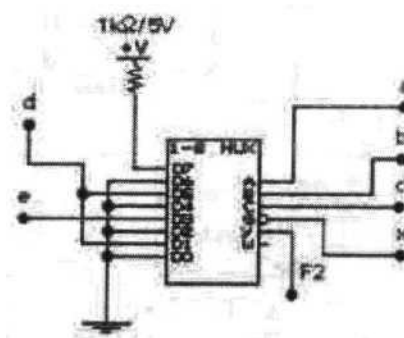


Рисунок 3

Так как используются только два адресных входа, управляющий вход C можно заземлить. При этом состояние информационных входов $D4...D7$ безразлично. Уровень сигнала на выходе схемы определяется комбинацией уровней сигналов в точках A, B, C , соответствующих переменным a, b, c . Схема на рисунке 3 по существу представляет собой мультиплексор 4×1 с двумя управляющими и четырьмя информационными входами.

Если функцию можно представить в виде произведения одночлена на многочлен, то её также можно реализовать при помощи мультиплексора. Как следует из уравнения мультиплексора, сигнал, соответствующий одночлену, нужно подать на вход разрешения. Например, требуется реализовать функцию $F2$, описываемую следующим выражением:

$$F2 = x \cdot d \cdot c \cdot \bar{b} \cdot \bar{a} \vee d \cdot \bar{b} \cdot a \vee e \cdot \bar{c} \cdot b \cdot a \vee c \cdot b \cdot a .$$

При реализации данной функции на мультиплексоре сигнал, соответствующий переменной x , следует подать на его разрешающий вход. Рассмотрим, какие сигналы необходимо подать на управляющие входы мультиплексора. Выражение в скобках можно рассматривать как некоторую функцию f пяти переменных: a, b, c, d, e , из которых наиболее часто используются переменные a, b, c . Поэтому сигналы, соответствующие этим переменным, нужно подать на управляющие входы мультиплексора. Определим, какие сигналы следует подать на информационные входы, чтобы реализовать функцию f . Для этого составим таблицу истинности функции в зависимости от значений переменных a, b, c (таблица 3).

Таблица 3

N	c	b	a	f
0	0	0	0	0
1	0	0	1	d
2	0	1	0	0
3	0	1	1	e
4	1	0	0	0
5	1	0	1	d
6	1	1	0	0
7	1	1	1	1

Из таблицы видно, что на информационные входы с номерами $N = 0, 2, 4, 6$ нужно подать уровень логического нуля. Сигнал, соответствующий переменной d , нужно подать на входы с номерами $N = 1, 5$, сигнал, соответствующий переменной e — на вход с номером 3.

Порядок выполнения

Эксперимент 1. Исследование мультиплексор

1. Соберите схему, приведенную на рисунке 4.

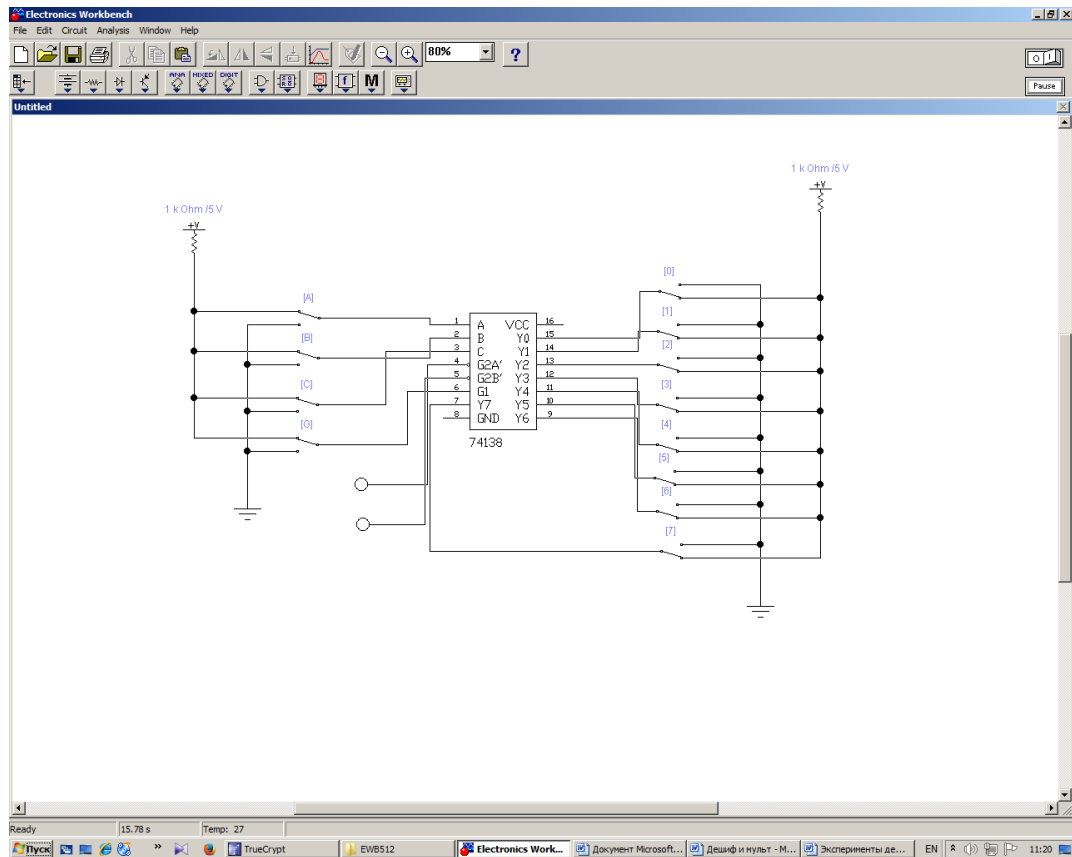


Рисунок 4

2. Включите схему.

3. С помощью ключа G установите на входе G мультиплексора уровень логического нуля. Поочередно подавая все возможные комбинации логических уровней при помощи ключей A, B, C на соответствующие входы мультиплексора, для каждой комбинации с помощью логических пробников определите, переключение какого из ключей в левой части схемы изменяет состояние выходов мультиплексора.

4. Обозначение соответствующего входа мультиплексора запишите в таблицу 4 в разделе «Результаты экспериментов», указав при этом, как передается входной сигнал на выходы мультиплексора (напрямую или с инверсией).

Например, если переключение ключа 4 изменяет состояние выходов мультиплексора, в таблице в строке с соответствующей комбинацией уровней сигналов на входах A, B, C следует записать для выхода Y — D4, для выхода W — D4.

5. Установите при помощи ключа G уровень логической единицы на входе G микросхемы. В раздел «Результаты экспериментов» запишите обозначения выводов, которые при переключении соответствующих ключей в левой части схемы не влияют на состояние выходов микросхемы.

Эксперимент 2. Исследование мультиплексора с помощью генератора слов.

1. Соберите схему, приведенную на рисунке 5.

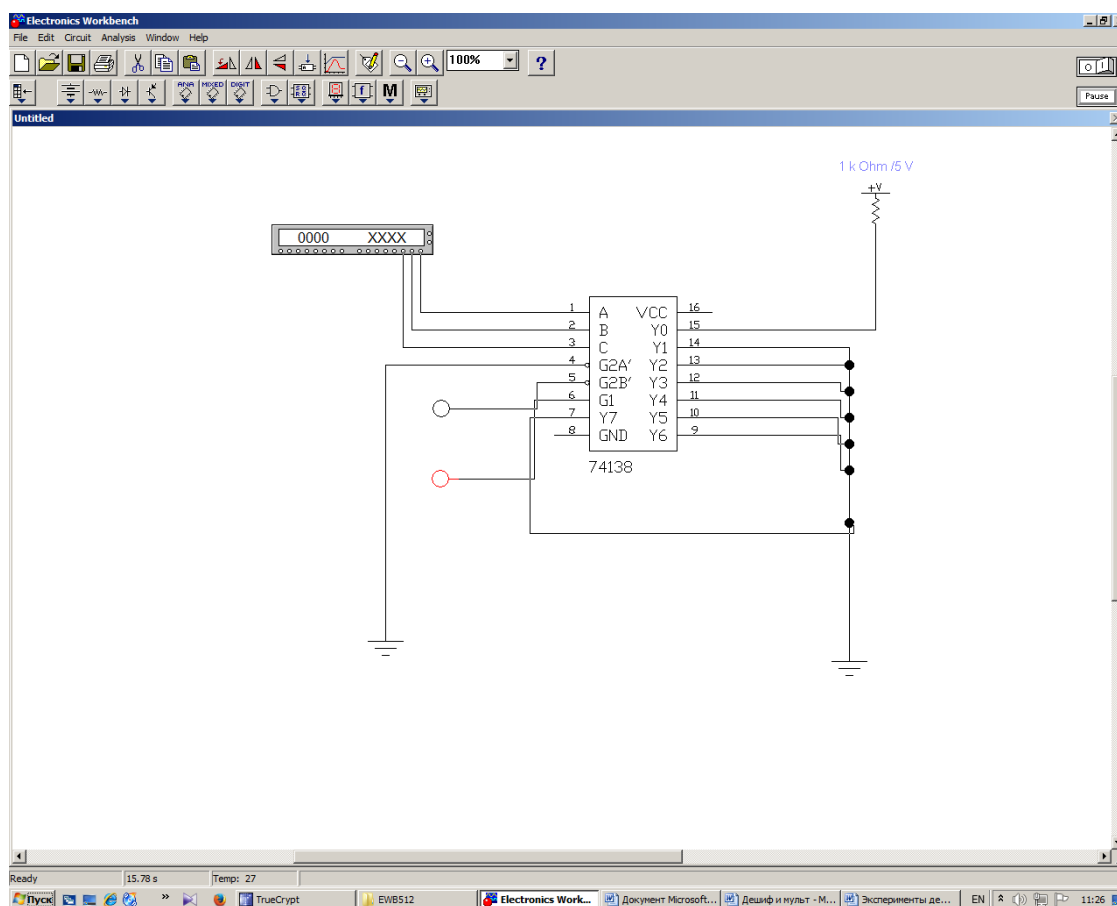


Рисунок 5

2. Включите схему.

3. Подавая в пошаговом режиме слова от генератора слов на входы мультиплексора и наблюдая уровни сигналов на выходах Y и W при помощи логических пробников, заполните таблицу 5 в разделе «Результаты экспериментов».

Эксперимент 3. Реализация заданной функции с помощью мультиплексора.

1. Определите значение функции F1 для каждой комбинации значений аргументов и заполните графу F1,а в разделе «Результаты экспериментов».

2. Соберите схему, приведенную на рисунке 6.

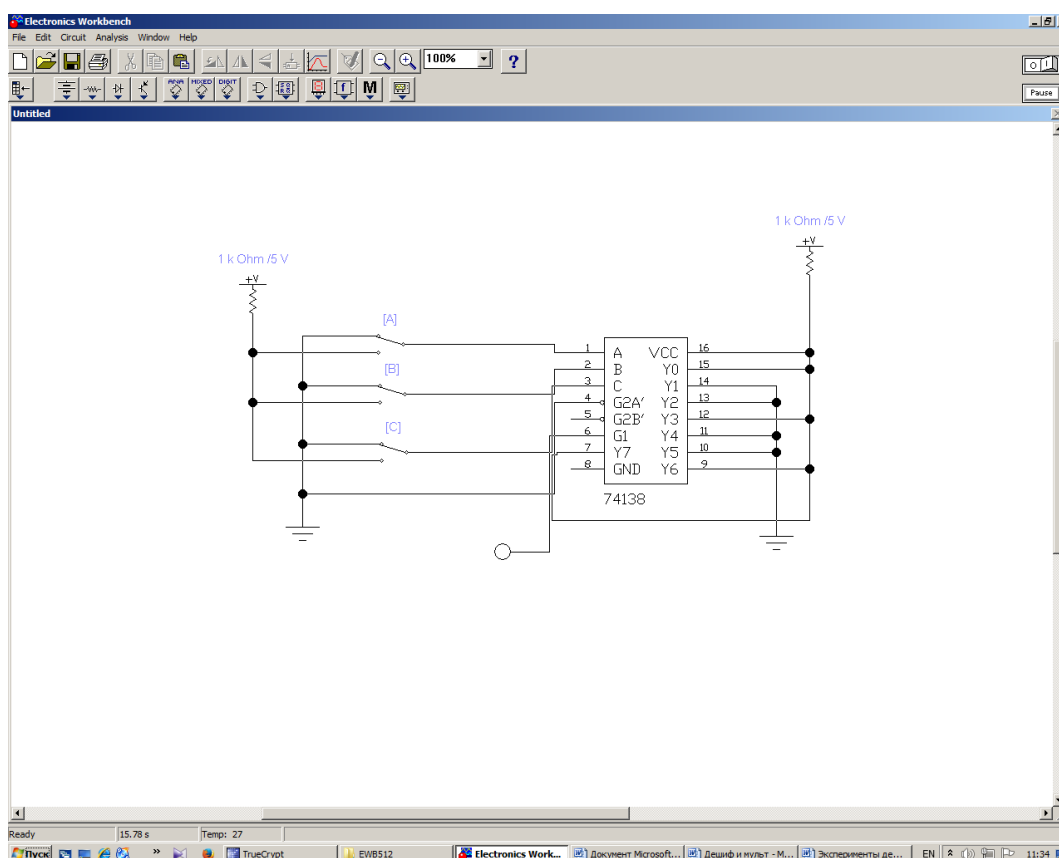


Рисунок 6

3. Включите схему.

4. Подайте при помощи ключей A, B, C все возможные комбинации логических сигналов на входы мультиплексора и, определяя уровень сигнала на выходе Y логическим пробником F1, заполните графу F1,а в таблице 6 в разделе

ле «Результаты экспериментов». Убедитесь, что функция, реализуемая мультиплексором, описывается выражением: $F1 = \bar{C} \cdot \bar{B} \cdot \bar{A} \vee B \cdot A \vee C \cdot B$.

5. Соберите схему, приведенную на рисунке 7.

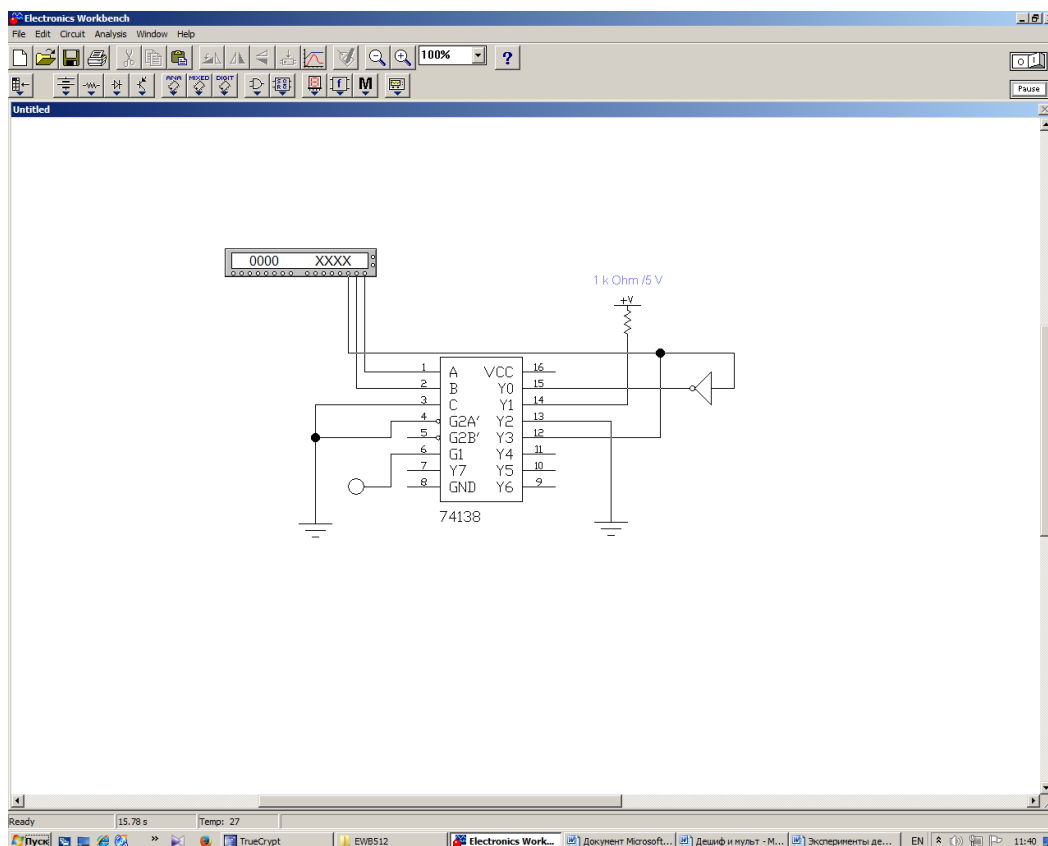


Рисунок 7

6. Включите схему.

7. Подавая в пошаговом режиме слова от генератора слов на входы мультиплексора и наблюдая уровень сигнала на выходе Y логическим пробником F1, заполните графу F1,б в таблице 7 в разделе «Результаты экспериментов». Убедитесь, что сигнал на выходе также определяется функцией F1.

8. Соберите схему, приведенную на рисунке 8.

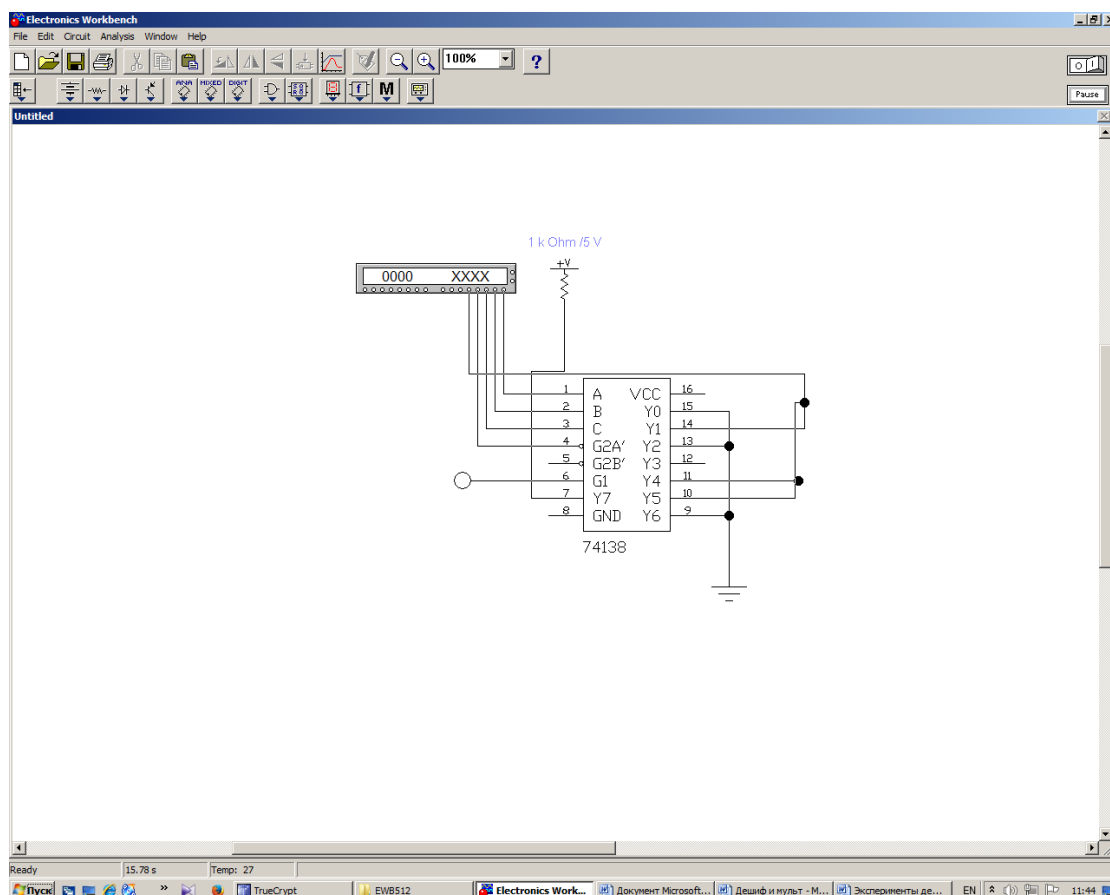


Рисунок 8

9. Последовательно подавая на входы схемы все возможные комбинации уровней логических сигналов, убедитесь, что уровень логической единицы на выходе появляется только в случаях, когда на входах схемы действуют комбинации, описываемые шестнадцатеричными эквивалентами 07H, 09H, 0CH, 0DH, 0FH, 13H, 17H, 19H, 1BH, 1CH, 1DH, 1FH, при которых функция F2 принимает значение 1.

Эксперимент 4. Исследование мультиплексора 74153.

1. Соберите схему, приведенную на рисунке 9.

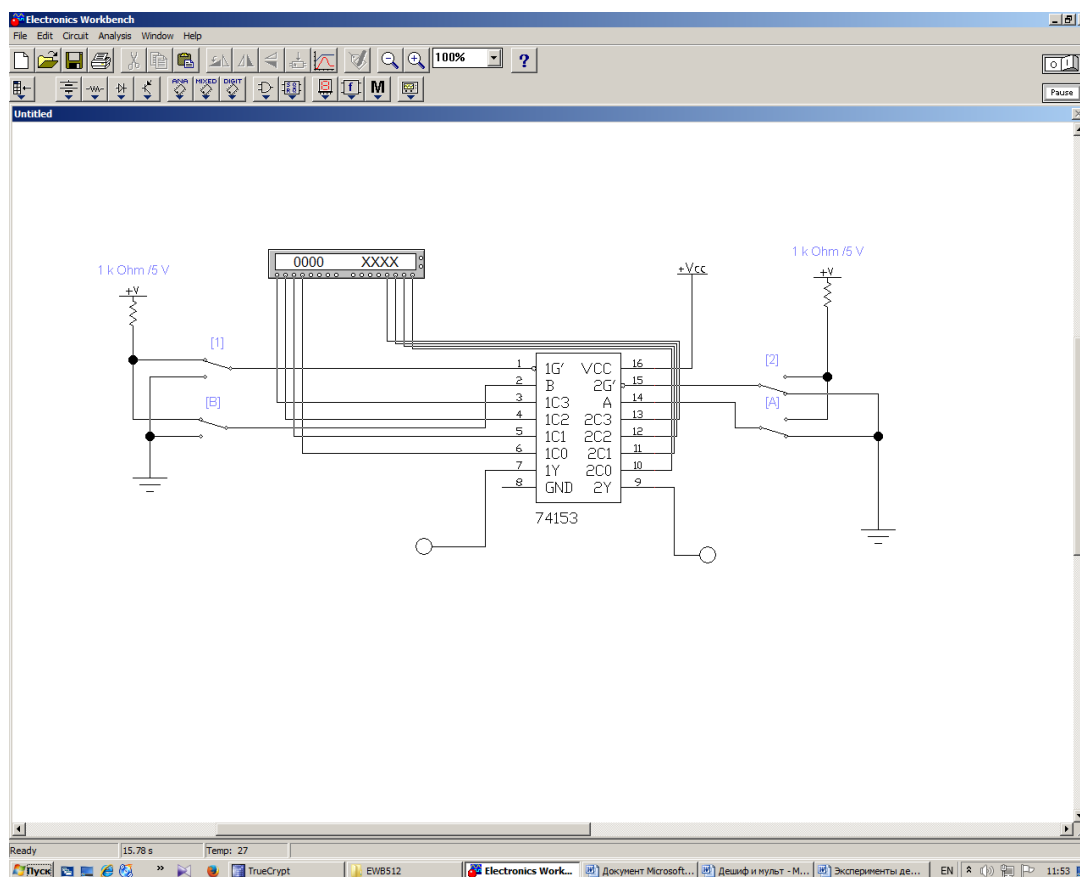


Рисунок 9

2. Исследуйте работу сдвоенного четырехканального мультиплексора (микросхема 74153).

3. Составьте таблицу функционирования схемы для выходов Y1 и Y2. Для этого установите ключами 1 и 2 уровень логического нуля на входах G1 и G2.

4. Затем в пошаговом режиме последовательно подайте от генератора все слова последовательности для каждой комбинации логических уровней на входах A и B. Для каждого шага определите входы, сигналы с которых проходят на выходы микросхемы. Обозначения входов занесите в таблицу 8 в разделе «Результаты экспериментов».

5. При помощи ключа 1 установите уровень логической единицы на входе 1G микросхемы.

6. Подавая на входы микросхемы слова от генератора, определите, какой из выходов микросхемы перестал реагировать на изменение состояния входов. Запишите обозначение этого вывода в раздел «Результаты экспериментов», в).

7. Повторите действия пункта б), установив на входе 2G уровень логической единицы, а на входе 1G — уровень логического нуля.

Результаты экспериментов

Эксперимент 1. Исследование мультиплексора.

а)

Таблица 4

A	B	C	Y	W
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

б)

Выводы, не влияющие на состояние выходов микросхемы: _____

Эксперимент 2. Исследование мультиплексора с помощью генератора слов.

Таблица 5

A	B	C	Y	W
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

Эксперимент 3. Реализация заданной функции с помощью мультиплексо-
ра.

Таблица 6

A	B	C	F1расч	F1a)	F1б)
0	0	0			
0	0	1			
0	1	0			
0	1	1			
1	0	0			
1	0	1			
1	1	0			
1	1	1			

Эксперимент 4. Исследование мультиплексора 74153.

а)

Таблица 7

A	B	1Y	2Y
0	0		
0	1		
1	0		
1	1		

б)

Обозначение вывода, не реагирующего на изменение состояния входов:

в)

Обозначение вывода, не реагирующего на изменение состояния входов:

Содержание отчета

1. Тема, цель занятия.
2. Схемы экспериментов.
3. Результаты экспериментов.
4. Выводы.

Контрольные вопросы

1. Функцию какого электрического устройства выполняет мультиплексор для логических сигналов?
2. Каким аналитическим уравнением описывается работа мультиплексора 2х1 с управляющим входом? В уравнении используйте следующие обозначения: входы - А, В, выход - Y, разрешающий вход G. Какие и в каком количестве логические элементы требуются для реализации этого уравнения?
3. Как реализовать схему мультиплексора 2х1 с управляющим входом на элементах И-НЕ?
4. Как можно на основе двух мультиплексоров 2х1 сделать один мультиплексор 4х1? Какие дополнительные элементы понадобятся для этого?
5. Функции скольких переменных можно реализовать без дополнительных элементов (за исключением инверторов) на одном мультиплексоре 4х1? На мультиплексоре 8х1? На мультиплексоре 16х1?

6. При какой форме аналитического представления логической функции, предназначенной для реализации на мультиплексоре, управляющий вход G может быть использован для подачи одного из входных сигналов?

7. Какими логическими уравнениями описывается работа микросхемы сдвоенного мультиплексора 74153?

Практическое занятие № 11

Исследование работы ОЗУ и ПЗУ

Цель занятия: изучить структуру и принципы функционирования ОЗУ и ПЗУ.

Краткие теоретические сведения

Оперативные запоминающие устройства (ОЗУ) являются неотъемлемой частью микропроцессорных систем различного назначения. ОЗУ делятся на два класса: статические и динамические. В статических ОЗУ запоминание информации производится на триггерах, а в динамических — на конденсаторах емкостью порядка 0,5 пФ. Длительность хранения информации в статических ОЗУ не ограничена, тогда как в динамических она ограничена временем саморазряда конденсатора, что требует специальных средств регенерации и дополнительных затрат времени на этот процесс.

Конструктивно любое ОЗУ состоит из двух блоков — матрицы запоминающих элементов и дешифратора адреса. По технологическим соображениям матрица чаще всего имеет двухкоординатную дешифрацию адреса — по строкам и столбцам. На рисунке 1 показана матрица 16-битного статического ОЗУ.

решения счетчика адреса (Addr_cnt) или такой же вход дешифраторов, подключенных к выходам счетчика.

При записи в ячейку памяти (рисунок 2) на соответствующей разрядной шине устанавливается 1 или 0, на входе WR/RD' устанавливается сигнал 1 и после стробирования счетчика или дешифраторов адреса сигналом CS срабатывают элементы 2И U1, U2. Положительный перепад сигнала с элемента U2 поступает на тактовый вход D-триггера U4, 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 WR/RD' устанавливается 0, при этом срабатывают элементы 111, U3, U5 и на вход РАЗРЕШЕНИЕ ВЫХОДА буферного элемента U6 поступает разрешающий сигнал, в результате чего сигнал с Q-выхода D-триггера передается на разрядную шину D00...D03. Для проверки функционирования ячейки памяти используется генератор слова (рисунок 3).

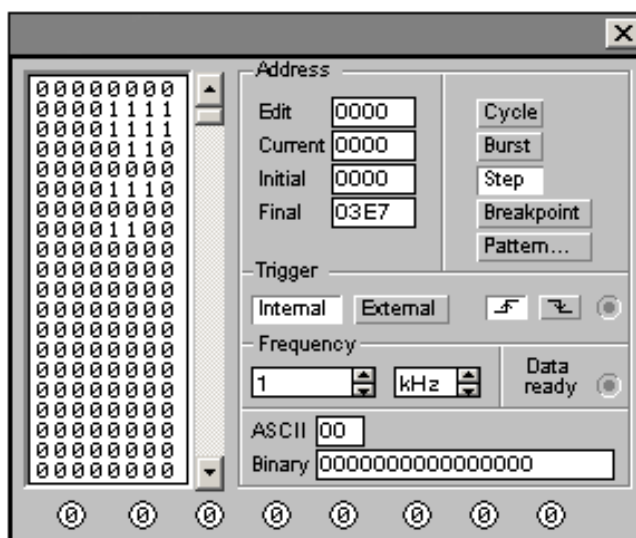


Рисунок 3 - Лицевая панель генератора слова с установками для схемы на рисунке 2

Современные запоминающие устройства статического типа отличаются высоким быстродействием и в микропроцессорных системах используются ограниченно из-за сравнительно высокой стоимости. В таких системах они используются только в качестве так называемой кэш-памяти. Cache (запас) обо-

значает быстродействующую буферную память между процессором и основной памятью, служащую для частичной компенсации разницы в скорости процессора и основной памяти — в нее заносятся наиболее часто используемые данные. Когда процессор первый раз обращается к ячейке памяти, ее содержимое параллельно копируется в кэш, и в случае повторного обращения может быть с гораздо большей скоростью из нее извлечено. При записи в память информация попадает в кэш и одновременно копируется в память (схема Write Through — прямая или сквозная запись) или копируется через некоторое время (схема Write Back — обратная запись). При обратной записи, называемой также буферизованной сквозной записью, информация копируется в память в первом же свободном такте, а при отложенной (Delayed Write) — когда для помещения в кэш нового значения не оказывается свободной области; при этом в основное ОЗУ вытесняются сравнительно редко используемые данные. Вторая схема более эффективна, но и более сложна за счет необходимости поддержания соответствия содержимого кэша и основной памяти.

Кэш-память состоит из области данных, разбитой на блоки (строки), которые являются элементарными единицами информации при работе кэша, и области признаков (tag), описывающей состояние строк (свободна, занята, помечена для дозаписи и т.п.). В основном используются две схемы организации кэша: с прямым отображением (direct mapped), когда каждый адрес памяти может кэшироваться только одной строкой (в этом случае номер строки определяется младшими разрядами адреса), и n-связный ассоциативный (n-way associative), когда каждый адрес может кэшироваться несколькими строками. Ассоциативный кэш более сложен, однако позволяет более гибко кэшировать данные; наиболее распространены четырехсвязные системы кэширования.

Микропроцессоры 486 и выше имеют также внутренний (Internal) кэш объемом 8... 16 Кбайт. Он также обозначается как Primary (первичный) или L1 (Level I — первый уровень) в отличие от внешнего (External), расположенного на плате и обозначаемого Secondary (вторичный) или L2. В большинстве процессоров внутренний кэш работает по схеме с прямой записью, а в 486 (процес-

сор Intel P24D и последние DX4-100, AMD DX4-120, 5x86) и Pentium он может работать и с отложенной записью. Последнее требует специальной поддержки со стороны системной платы, чтобы при обмене по DMA (прямое обращение к памяти устройств ввода-вывода) можно было поддерживать согласованность данных в памяти и внутреннем кэше. Процессоры Pentium Pro имеют также встроенный кэш второго уровня объемом 256 или 512 Кбайт.

В микропроцессорных системах в качестве ОЗУ чаще всего используются динамические ОЗУ с запоминающим конденсатором, которые отличаются большим многообразием. Приведем данные по наиболее распространенным типам таких ОЗУ.

В динамической памяти ячейки выполнены на основе областей с накоплением зарядов, занимающих гораздо меньшую площадь, нежели триггеры, и практически не потребляющих энергии при хранении информации. При записи бита в такую ячейку в ней формируется электрический заряд, который сохраняется в течение нескольких миллисекунд; для постоянного сохранения заряда ячейки необходимо регенерировать (перезаписывать) ее содержимое. Ячейки микросхем динамической памяти также организованы в виде прямоугольной матрицы; при обращении к микросхеме на ее входы вначале подается адрес строки матрицы, сопровождаемый сигналом RAS (Row Address Strobe — строб адреса строки), затем, через некоторое время — адрес столбца, сопровождаемый сигналом CAS (Column Address Strobe — строб адреса столбца). При каждом обращении к отдельной ячейке регенерируются все ячейки выбранной строки, поэтому для полной регенерации матрицы достаточно перебрать адреса строк. Ячейки динамической памяти имеют сравнительно малое быстродействие (десятки — сотни наносекунд), но большую удельную плотность (порядка нескольких мегабайт на корпус) и меньшее энергопотребление.

Обычные ОЗУ называют часто асинхронными, так как установка адреса и подача управляющих сигналов могут выполняться в произвольные моменты времени, необходимо только соблюдение временных соотношений между этими сигналами. В них включены так называемые охранные интервалы, необхо-

димые для установления сигналов. Существуют также синхронные виды памяти, получающие внешний синхросигнал, к импульсам которого жестко привязаны моменты подачи адресов и обмена данными; они позволяют более полно использовать внутреннюю конвейеризацию и блочный доступ.

FPM DRAM (Fast Page Mode DRAM — динамическая память с быстрым страничным доступом), активно используется в последнее время. Память со страничным доступом отличается от обычной динамической памяти тем, что после выбора строки матрицы и удержания сигнала RAS допускает многократную установку адреса столбца, стробируемого сигналом CAS, а также быструю регенерацию по схеме «CAS прежде RAS». Первое позволяет ускорить блочные передачи, когда весь блок данных или его часть находятся внутри одной строки матрицы, называемой в этой системе страницей, а второе — снизить затраты времени на регенерацию памяти.

EDO (Extended Data Out — расширенное время удержания данных на выходе) фактически представляют собой обычные микросхемы FPM, на выходе которых установлены регистры-защелки данных. При страничном обмене такие микросхемы работают в режиме простого конвейера: удерживают на выходах данных содержимое последней выбранной ячейки, в то время как на их входы уже подается адрес следующей выбираемой ячейки. Это позволяет примерно на 15% по сравнению с FPM ускорить процесс считывания последовательных массивов данных. При случайной адресации такая память ничем не отличается от обычной.

BEDO (Burst EDO — EDO с блочным доступом) — память на основе EDO, работающая не одиночными, а пакетными циклами чтения/записи. Современные процессоры благодаря внутреннему и внешнему кэшированию команд и данных обмениваются с основной памятью преимущественно блоками слов максимальной ширины. При наличии памяти BEDO отпадает необходимость постоянной подачи последовательных адресов на входы микросхем с соблюдением необходимых временных задержек, до- ' статочно стробировать переход к очередному слову отдельным сигналом.

SDRAM (Synchronous DRAM — синхронная динамическая память) — память с синхронным доступом, работающая быстрее обычной асинхронной (FPM/EDO/BEDO). Кроме синхронного доступа, SDRAM использует внутреннее разделение массива памяти на два независимых банка, что позволяет совмещать выборку из одного банка с установкой адреса в другом. SDRAM также поддерживает блочный обмен. Основное преимущество SDRAM состоит в поддержке последовательного доступа в синхронном режиме, где не требуется дополнительных тактов ожидания. При случайном доступе SDRAM работает практически с той же скоростью, что и FPM/EDO.

PB SRAM (Pipelined Burst SRAM — статическая память с блочным конвейерным доступом) — разновидность синхронных SRAM с внутренней конвейеризацией, за счет которой примерно вдвое повышается скорость обмена блоками данных.

Кроме основного ОЗУ, устройством памяти снабжается и устройство отображения информации — видеодисплейная система. Такая память называется видеопамью и располагается на плате видеоадаптера.

Видеопамью служит для хранения изображения. От ее объема зависит максимально возможное разрешение видеокарты — $A \times B \times C$, где A — количество точек по горизонтали, B — по вертикали, C — количество возможных цветов каждой точки. Например, для разрешения $640 \times 480 \times 16$ достаточно иметь видеопамью 256 Кбайт, для $800 \times 600 \times 256$ — 512 КБ, для $1024 \times 768 \times 65536$ (другое обозначение — $1024 \times 768 \times 64k$) — 2 Мбайт и т.д. Поскольку для хранения цветов отводится целое число разрядов, количество цветов всегда является целой степенью 2 (16 цветов — 4 разряда, 256 — 8 разрядов, 64k — 16 и т.д.).

В видеоадаптерах используются следующие типы видеопамью.

FPM DRAM (Fast Page Mode Dynamic RAM — динамическое ОЗУ с быстрым страничным доступом) — основной тип видеопамью, идентичный используемой в системных платах. Наиболее распространенные микросхемы FPM DRAM — четырехразрядные DIP и SOJ, а также шестнадцатиразрядные SOJ.

VRAM (Video RAM — видео-ОЗУ) — так называемая двухпортовая DRAM с поддержкой одновременного доступа со стороны видеопроцессора и центрального процессора компьютера. Позволяет совмещать во времени вывод изображения на экран и его обработку в видеопамати, что сокращает задержки и увеличивает скорость работы.

EDO DRAM (Extended Data Out DRAM — динамическое ОЗУ с расширенным временем удержания данных на выходе) — память с элементами конвейеризации, позволяющей несколько ускорить обмен блоками данных с видеопаматью.

SGRAM (Synchronous Graphics RAM — синхронное графическое ОЗУ) — вариант DRAM с синхронным доступом, когда все управляющие сигналы изменяются одновременно с системным тактовым синхросигналом, что позволяет уменьшить временные задержки.

WRAM (Window RAM — оконное ОЗУ) — EDO VRAM, в котором окно, через который обращается видеоконтроллер, сделано меньшим, чем окно для центрального процессора.

MDRAM (Multibank DRAM — многобанковое ОЗУ) — вариант DRAM, организованный в виде множества независимых банков объемом по 32 Кбайт каждый, работающих в конвейерном режиме.

Увеличение скорости обращения видеопроцессора к видеопамати, кроме повышения пропускной способности адаптера, позволяет повысить максимальную частоту регенерации изображения, что снижает утомляемость глаз оператора.

Микросхемы памяти имеют четыре основные характеристики — тип, объем, структуру и время доступа. Тип обозначает статическую или динамическую память, объем показывает общую емкость памяти, а структура — количество ячеек памяти и разрядность каждой ячейки. Например, 28/32-выводные DIP-микросхемы SRAM имеют 8-разрядную структуру (8к×8, 16к×8, 32к×8, 64к×8, 128к×8), кэш объемом 256 Кбайт состоит из восьми микросхем 32кх8 или четырех микросхем 64к×8 (речь идет об области данных, дополнительные

микросхемы для хранения признаков могут иметь другую структуру). Две микросхемы по 128кx8 поставить уже нельзя, так как нужна 32-разрядная шина данных, что могут обеспечить только четыре микросхемы. Распространенные PB SRAM в 100-выводных корпусах PQFP имеют 32-разрядную структуру 32кx32 или 64кx32 и используются по две или по четыре в платах для Pentium.

30-контактные SIMM имеют 8-разрядную структуру и используются с процессорами 286, 386SX и 486SLC по две, а с 386DX, 486DLC и обычными 486DX — по четыре. 72-контактные SIMM имеют 32-разрядную структуру и могут использоваться с 486DX по одной, а с Pentium и Pentium Pro — по две. 168-контактные DIMM имеют 64-разрядную структуры и используются в Pentium и Pentium Pro по одной. Установка модулей памяти или микросхем кэша в количестве больше минимального для данной системной (материнской) платы позволяет ускорить работу с ними, используя принцип чередования (Interleave).

Время доступа характеризует скорость работы микросхемы и обычно указывается в наносекундах после тире в конце наименования. На более медленных микросхемах могут указываться только первые цифры (-7 вместо -70, -15 вместо -150), на более быстрых статических «-15» или «-20» обозначает реальное время доступа к ячейке. Часто на микросхемах указывается минимальное из всех возможных времен доступа, например, распространена маркировка 50 EDO DRAM вместо 70, или 45 — вместо 60, хотя такой цикл достижим только в блочном режиме, а в одиночном режиме микросхема по-прежнему имеет время доступа 70 или 60 нс. Аналогичная ситуация имеет место в маркировке PB SRAM: 6 вместо 12, и 7 вместо 15. Микросхемы SDRAM обычно маркируются временем доступа в блочном режиме (10 или 12 нс).

ИМС памяти реализуются в корпусах следующих типов:

— DIP (Dual In line Package — корпус с двумя рядами выводов) — классические микросхемы, применявшиеся в блоках основной памяти IBM PC/XT и ранних PC/AT, сейчас применяются в блоках кэш-памяти;

— SIP (Single In line Package — корпус с одним рядом выводов) — микросхема с одним рядом выводов, устанавливаемая вертикально;

— SIMM (Single In line Memory Module — модуль памяти с одним рядом контактов) — модуль памяти, вставляемый в зажимный разъем; применяется во всех современных платах, а также во многих адаптерах, принтерах и прочих устройствах. SIMM имеет контакты с двух сторон модуля, но все они соединены между собой, образуя как бы один ряд контактов. На SIMM в настоящее время устанавливаются преимущественно микросхемы FPM/EDO/BEDO;

— DIMM (Dual In line Memory Module — модуль памяти с двумя рядами контактов) — модуль памяти, похожий на SIMM, но с отдельными контактами (обычно 2x84), за счет чего увеличивается разрядность или число банков памяти в модуле. Применяется в основном в компьютерах Apple и новых платах P5 и P6. На DIMM устанавливаются микросхемы EDO/BEDO/SDRAM;

— CELP (Card Edge Low Profile — невысокая карта с ножевым разъемом на краю) — модуль внешней кэш-памяти, собранный на микросхемах SRAM (асинхронный) или PB SRAM (синхронный). По внешнему виду похож на 72-контактный SIMM, имеет емкость 256 или 512 Кбайт. Другое название — COAST (Cache On A STick — буквально «кэш на палочке»).

Модули динамической памяти кроме основных ячеек памяти могут иметь дополнительные ячейки для хранения битов четности (Parity) для байтов данных; такие SIMM иногда называют 9- и 36-разрядными модулями (по одному биту четности на байт данных). Биты четности служат для контроля правильности считывания данных из модуля, позволяя обнаружить часть ошибок. Модули с битами четности имеет смысл применять лишь там, где нужна очень высокая надежность. Для обычных применений подходят и тщательно проверенные модули без битов четности, однако при условии, что системная плата поддерживает такие типы модулей.

Проще всего определить тип модуля по маркировке и количеству микросхем памяти на нем: например, если на 30-контактном SIMM две микросхемы одного типа и еще одна — другого, то две первых являются основными (каждая — по четыре разряда), а третья предназначена для хранения битов четности (она одноразрядная).

В 72-контактном SIMM с двенадцатью микросхемами восемь из них хранят данные, а четыре — биты четности. Модули с количеством микросхем 2, 4 или 8 не имеют памяти для хранения битов четности.

Иногда на модули ставится так называемый имитатор четности — микросхема-сумматор, выдающая при считывании ячейки всегда правильный бит четности. В основном это предназначено для установки таких модулей в платы, где проверка четности не отключается.

72-контактные SIMM имеют четыре специальных линии PD (Presence Detect — обнаружение наличия), на которых при помощи перемычек может быть установлено до 16 комбинаций сигналов. Линии PD используются в некоторых материнских платах для определения наличия модулей памяти в разъемах и их параметров (объема и быстродействия). В большинстве универсальных плат производства «третьих фирм», как и выпускаемые ими SIMM, линии PD не используются.

В модулях DIMM в соответствии со спецификацией JEDEC технология PD реализуется при помощи перезаписываемого ПЗУ с последовательным доступом (Serial EEPROM) и носит название Serial Presence Detect (SPD). ПЗУ представляет собой 8-выводную микросхему, размещенную в углу платы DIMM, а его содержимое описывает конфигурацию и параметры модуля. Системные платы с набором микросхем (чипсетами) 440LX/BX могут использовать SPD для настройки системы управления памятью. Некоторые системные платы могут обходиться без SPD, определяя конфигурацию модулей обычным путем.

Постоянные запоминающие устройства (ПЗУ) делятся на четыре типа:

- масочные, программируемые на заводе-изготовителе с применением специальных масок;
- однократно программируемые потребителем путем пережигания нишромовых или поликремневых перемычек;
- многократно программируемые потребителем со стиранием записанной информации ультрафиолетовым излучением;
- многократно программируемые потребителем с электрическим стира-

нием информации.

Рассмотрим ПЗУ второго типа, которое состоит из дешифратора $px2n$ и подключенных к его выходам схем ИЛИ с плавкими перемычками (рисунок 2.4). ПЗУ содержит дешифратор 2×4 в виде подсхемы `pzti_dcd` (A, B — кодовые входы, E — вход разрешения, активный сигнал высокого уровня), к выходам которых можно подключить четыре элемента 4ИЛИ с дополнительными устройствами. На рисунке 4 показаны два таких элемента, выполненных в виде отдельных подсхем `pzu_un1` и `pzu_un2`. Хотя эти элементы одинаковы, наращивание их на схеме путем копирования исключено из-за наличия пережигаемых перемычек — при наличии одноименных подсхем пережигание перемычки в одной подсхеме автоматически приведет к пережиганию такой же перемычки в другой. Поскольку программа не позволяет копировать подсхемы с их переименованием, все их приходится выполнять полностью. На схеме D0, D1 — выходы младшего и первого разрядов.

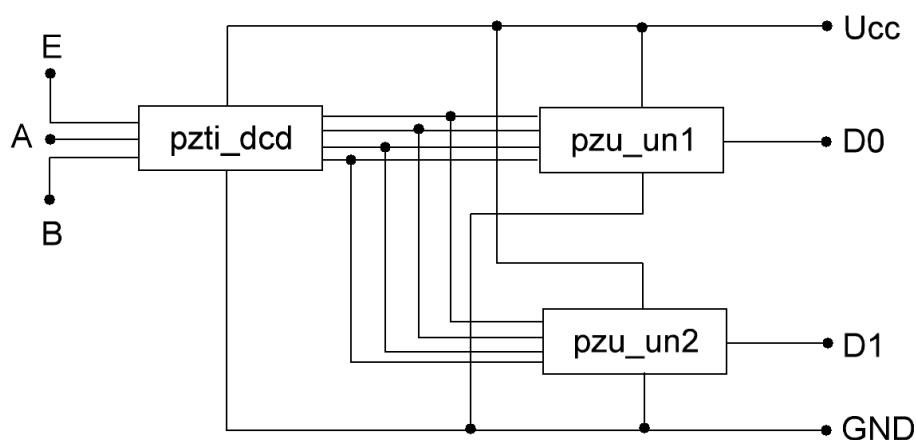


Рисунок 4 - Схема двухразрядного ПЗУ

Схема дешифратора `pzu_dcd` показана на рисунке 5.

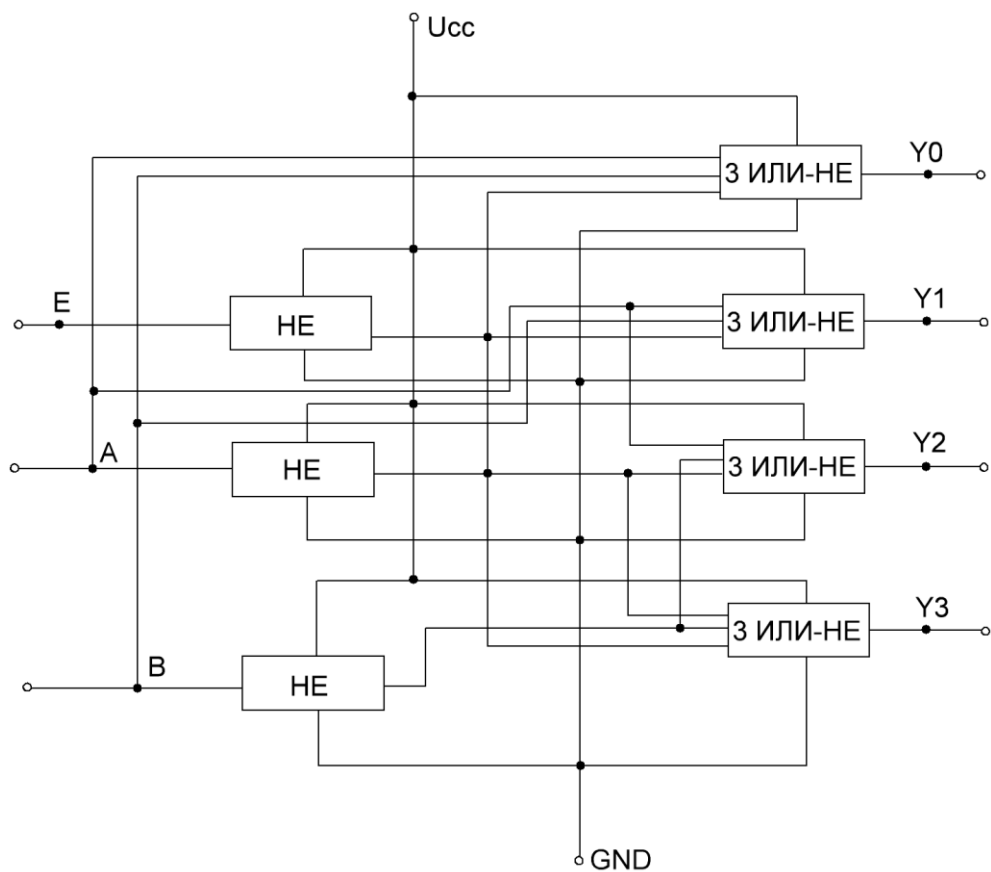


Рисунок 5 - Внутренняя структура подсхемы дешифратора

Дешифратор выполнен на трех элементах НЕ и четырех элементах 3 ИЛИ-НЕ на транзисторах (рисунок 6).

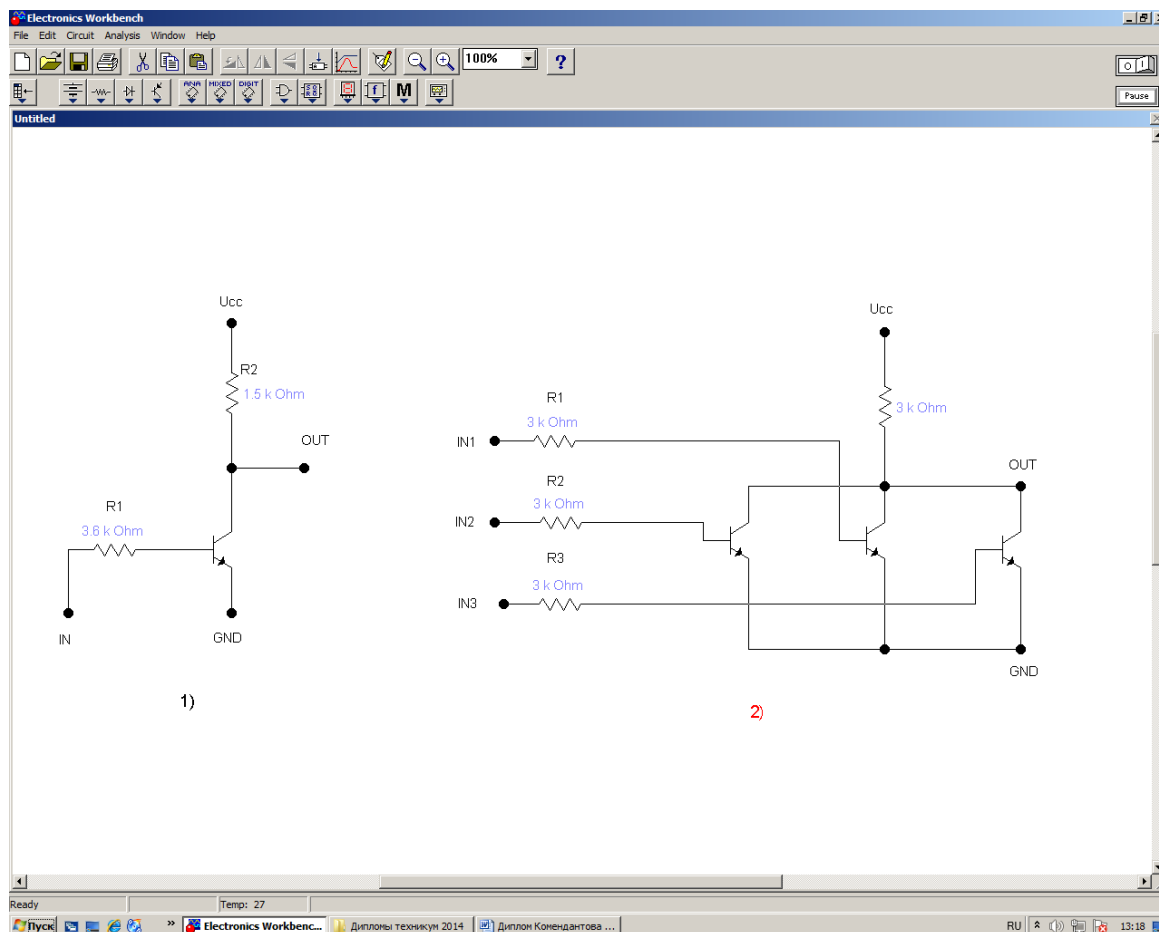


Рисунок 6 - Схемы элементов НЕ (1) и ИЛИ-НЕ (2)

Необходимость выполнения элементов дешифратора на транзисторах объясняется тем, что используемые в программе EWB математические модели цифровых ИМС не всегда позволяют подключать к ним обычные транзисторные схемы и, в частности, применяемые в рассматриваемом ПЗУ ячейки памяти в виде подсхемы rzu_uni. Ее внутренняя структура аналогична структуре ячейки памяти, используемой в ПЗУ K155PE3 (рисунок 7).

D0 формируется сигнал логической единицы (около +5 В). При подаче на входы А, В дешифратора заданной кодовой комбинации, а на вход разрешения Е — сигнала логической единицы, один из транзисторов Т1...Т4 откроется и на выходе D0 сформируется сигнал логического нуля. Так, например, при А=В=1 откроется транзистор Т4 и сигнал логической единицы с его эмиттера через переключку S4 поступит на делитель на резисторах R2, R3, транзистор Т6 откроется, и на его выходе сформируется сигнал логического нуля. Очевидно, что и при любой другой двоичной комбинации будет происходить то же самое до тех пор, пока не будет разрушена соответствующая переключка.

Пережигание переключек составляет суть программирования и осуществляется отдельно для каждого разряда (каждой ячейки) следующим образом:

- на входы А, В (рисунок 4) подается двоичная комбинация, соответствующая адресу пережигаемой переключки в программируемом разряде (в ячейке rzu_unx , где x — номер ячейки);

- к выходу ячейки D x через резистор нагрузки (его сопротивление для конкретных ИМС указывается в документации, для К155РЕЗ составляет около 300 Ом) подключается источник напряжения 12,5 В, в результате чего стабилитрон D пробивается и транзистор Т5 открывается;

- на вход разрешения Е на короткое время подается сигнал логической единицы, при этом через один из открытых транзисторов Т1 ...Т2 и Т5 протекает ток, достаточный для пережигания соответствующей переключки (длительность разрешающего сигнала на входе Е в промышленных программаторах может автоматически увеличиваться после нескольких неудачных попыток программирования одной и той же ячейки);

- источник 12,5 В отключается, и после раскрытия соответствующей подсхемы можно убедиться, что переключка действительно разрушена (в промышленных программаторах этот процесс сводится к проверке записи программируемой ячейки, и при отрицательном результате производится повторное программирование при большей длительности разрешающего сигнала).

Заключительным этапом программирования серийных микросхем ПЗУ в промышленных условиях является электротермотренировка, которая проводится чаще всего в течение 168 часов при повышенной температуре, после чего производится дополнительный контроль записанной информации. Если при этом обнаруживается ошибка, допускается повторное программирование. Если ошибка снова повторяется, микросхема бракуется.

Для моделирования процесса программирования к программируемой схеме необходимо подключить дополнительные элементы. Моделирование целесообразно начинать с одноразрядного ПЗУ (рисунок 8).

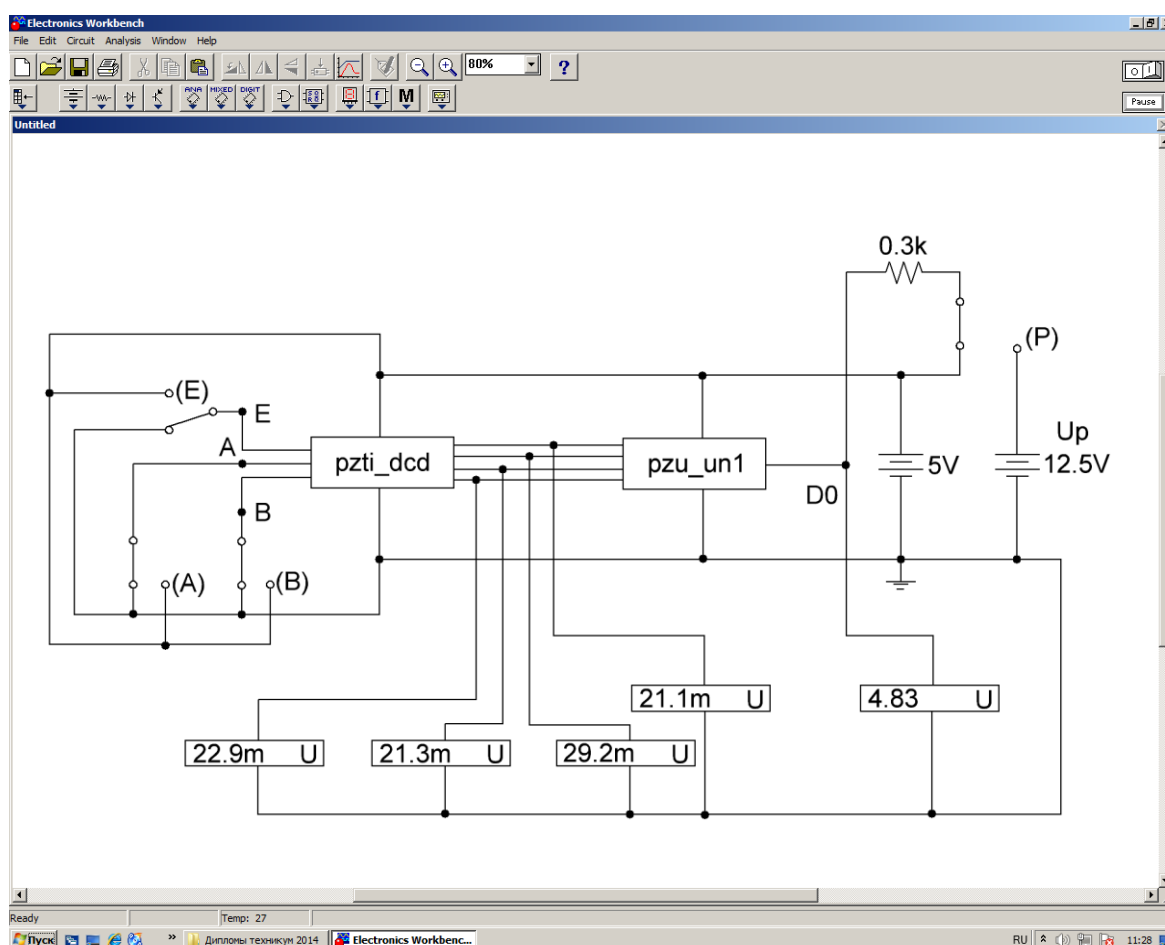


Рисунок 8 - Модель с дополнительными элементами

Следует отметить, что рассматриваемая модель ПЗУ (как на рисунке 4, так и на рисунке 8) достаточно капризна и при некоторых комбинациях входных сигналов моделирование не выполняется. Признаком невозможности моделирования является отсутствие следа от включателя питания (в верхнем пра-

вом углу экрана) окошка с индикацией временных интервалов отсчета. По истечении некоторого времени может быть выдана рекомендация изменить установку погрешности моделирования (по умолчанию она равна 1%). Целесообразно установить ее максимально возможной (10%) в меню Circuit (команда Analysis Options, параметр Tolerance). Целесообразно также поварьировать сопротивлениями входных резисторов и резисторов нагрузки элементов НЕ и ИЛИ-НЕ (рисунок 2.6), а также попробовать изменить параметры транзисторов. В крайнем случае можно ограничиться простейшим случаем — обойтись без дешифратора и использовать только одну ячейку памяти на рисунке 7, подключив к выходу и к одному из ее входов дополнительные элементы, как показано на рисунке 6.

ПЗУ с пережигаемыми перемычками используются чаще всего в качестве специализированных дешифраторов, например для селекции UVB.

ПЗУ с ультрафиолетовым стиранием используются в микропроцессорных системах для хранения управляющих программ, в частности, для размещения BIOS (Basic Input/Output System — основная система ввода/вывода, записанная в ПЗУ, отсюда ее полное название ROM BIOS). BIOS представляет собой набор программ проверки и обслуживания аппаратуры компьютера и выполняет роль посредника между операционной системой (ОС) и аппаратурой. BIOS получает управление при включении системной платы, тестирует саму плату и основные блоки компьютера — видеоадаптер, клавиатуру, контроллеры дисков и портов ввода/вывода, настраивает чипсет платы и загружает внешнюю ОС. При работе под управлением DOS/Windows 3.x/95/98 BIOS управляет основными устройствами, при работе под OS/2, Unix, Windows NT BIOS практически не используется, выполняя лишь начальную проверку и настройку.

Обычно на системной плате установлено только ПЗУ с системным (Main System) BIOS, отвечающим за саму плату и контроллеры FDD (флоппи-дисков), HDD (жестких дисков), портов и клавиатуры; в системный BIOS практически всегда входит System Setup — программа настройки системы. Видеоадаптеры и контроллеры HDD с интерфейсом ST-506 (MFM) и SCSI имеют собственные

BIOS в отдельных ПЗУ; их также могут иметь и другие платы — интеллектуальные контроллеры дисков и портов, сетевые карты и т.п.

Обычно BIOS для современных системных плат разрабатывается одной из специализированных фирм: Award Software, American Megatrends (AMI), реже: Phoenix Technology, Microid Research; в данное время наиболее популярны BIOS фирмы Award. Некоторые производители плат (например, IBM, Intel и Acer) сами разрабатывают BIOS для них. Иногда для одной и той же платы имеются версии BIOS разных производителей, в этом случае допускается копировать прошивки или заменять микросхемы ПЗУ; в общем же случае каждая версия BIOS привязана к конкретной модели платы.

Раньше BIOS помещался в однократно программируемые ПЗУ либо ПЗУ с ультрафиолетовым стиранием; сейчас в основном выпускаются платы с электрически перепрограммируемыми ПЗУ (Flash ROM), которые допускают перепрограммирование BIOS средствами самой платы. Это позволяет исправлять заводские ошибки в BIOS, изменять заводские установки по умолчанию, программировать собственные экранные заставки и т.п.

Тип микросхемы ПЗУ обычно можно определить по маркировке: 27xxxx — обычное ПЗУ, 28xxxx или 29xxxx — перепрограммируемые. Если на корпусе микросхемы 27xxxx есть прозрачное окно — это ПЗУ с ультрафиолетовым стиранием; если его нет — это однократно программируемое ПЗУ, которое можно лишь заменить на другое.

Видео-ПЗУ (Video ROM) — постоянное запоминающее устройство, в которое записаны видео-BIOS, экранные шрифты, служебные таблицы и т.п. ПЗУ не используется видеоконтроллером напрямую, к нему обращается только центральный процессор, в результате выполнения программ, записанных в ПЗУ, происходят обращения к видеоконтроллеру и видеопамяти. На многих современных видеокартах устанавливаются электрически перепрограммируемые ПЗУ (EEPROM, Flash ROM), допускающие перезапись пользователем под управлением специальной программы из комплекта карты.

ПЗУ необходимо только для первоначального запуска видеоадаптера и работы в режиме DOS, Novell Netware и других ОС, функционирующих преимущественно в текстовом режиме; ОС Windows, OS/2 и им подобные, работающие через собственные видеодрайверы, не используют ПЗУ для управления адаптером либо используют его только при выполнении программ для DOS.

При создании видео-BIOS все разработчики придерживаются рекомендаций VESA и VBE. VESA (Video Electronics Standards Association — ассоциация стандартизации видеоэлектроники) — организация, выпускающая различные стандарты в области электронных видеосистем и их программного обеспечения. VBE (VESA BIOS Extension — расширение BIOS в стандарте VESA) — дополнительные функции видео-BIOS по отношению к стандартному видео-BIOS для VGA, позволяющие запрашивать у адаптера список поддерживаемых видеорежимов и их параметров (разрешение, цветность, способы адресации, развертка и т.п.) и изменять эти параметры для согласования адаптера с конкретным монитором. По сути, VBE является унифицированным стандартом программного интерфейса с VESA-совместимыми картами, при работе через видео-BIOS он позволяет обойтись без специализированного драйвера видео-карты.

Порядок выполнения

Задание 1. Исследование ячейки оперативного запоминающего устройства.

Порядок выполнения работы:

- соберите схему, изображенную на рисунке 2;
- наберите на генераторе слова установки, показанные на рисунке 3;
- осуществите запись и считывание информации (контроль и запись информации осуществляйте по индикатору в схеме рисунка 2);
- поясните работу элементов ячейки при записи и считывании информации.

Задание 2. Исследование постоянного запоминающего устройства.

Порядок выполнения работы:

- соберите схему, изображенную на рисунке 8;
- проведите моделирование процесса программирования ПЗУ с пережигаемыми перемычками;
- соберите схему, изображенную на рисунке 4;
- проведите моделирование процесса программирования одной из его ячеек памяти.

Содержание отчета

1. Тема, цель занятия.
2. Схемы лабораторных установок.
3. Результаты моделирования.
4. Выводы.

Контрольные вопросы

1. Структура ОЗУ.
2. Структура ПЗУ.
3. Виды ОЗУ.
4. Виды ПЗУ.
5. Характеристики ОЗУ и ПЗУ.

Практическое занятие №12

Исследование работы АЛУ

Цель занятия: изучить структуру и принципы функционирования АЛУ.

Краткие теоретические сведения

ИМС арифметико-логического устройства (АЛУ) 74181 (К155ИПЗ) возможно использовать в качестве четырехразрядного сумматора. Эта ИМС обеспечивает 32 режима работы АЛУ в зависимости от состояния управляющих сигналов на входах M, S0...S3. Показанная на рисунке 1 схема на базе этой ИМС позволяет оперативно реализовать все упоминавшиеся режимы.

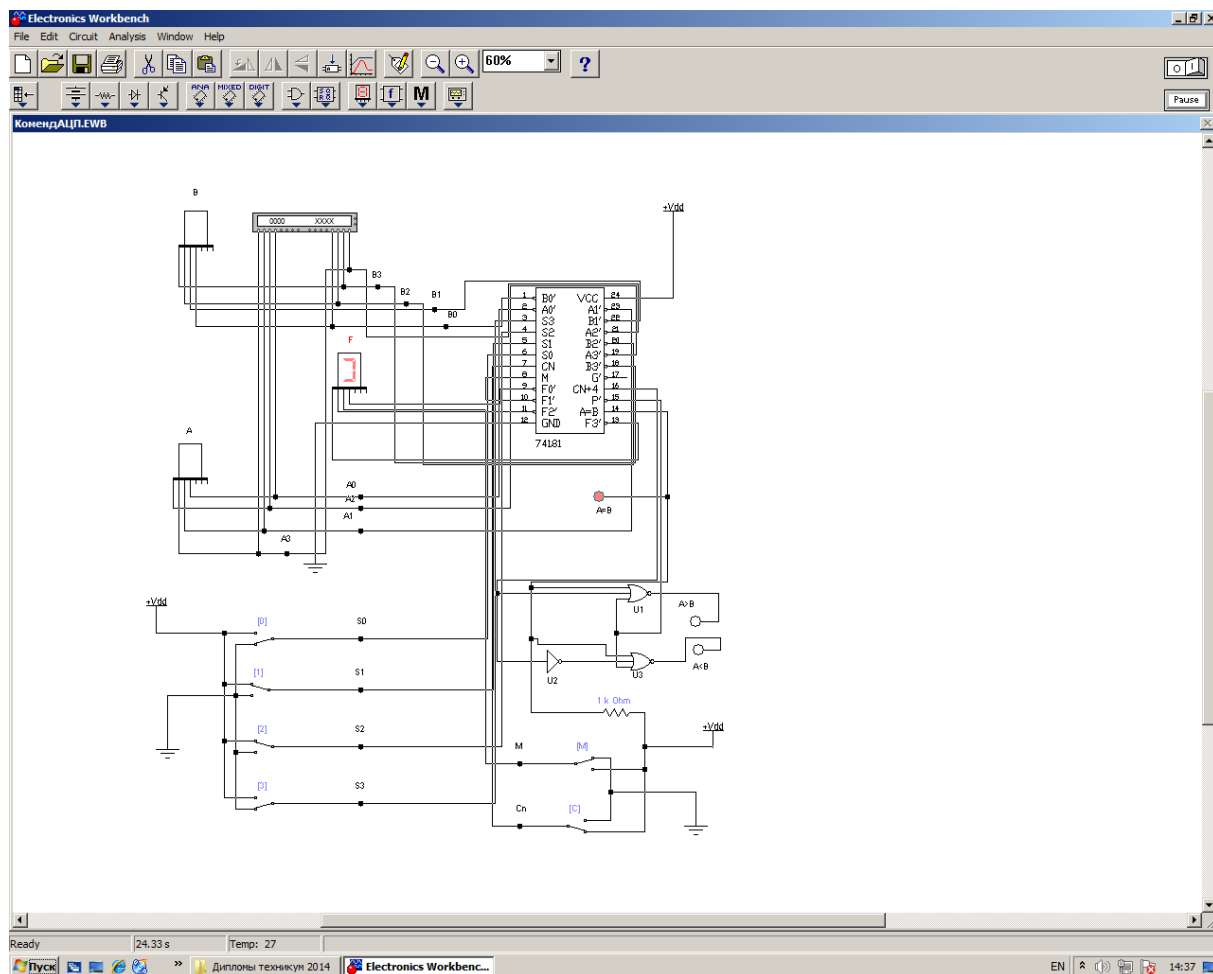


Рисунок 1 – АЛУ на ИМС 74181

Возможные режимы задаются с помощью переключателей 0, 1, 2, 3 для подачи сигналов 0 («земля») или 1 (+5 В) на входы управления S0, S1, S2, S3. В положении переключателя M, показанном на рисунке 2.9 (сигнал 0 на входе M), выполняются 16 арифметических операций (16 комбинаций сигналов S0...S3) с учетом переноса по входу Сп (переключатель С в показанном на рисунке 2.9 положении) или без учета переноса (сигнал 0 на входе Сп переключателя С).

При переводе ключа М в другое положение (на входе М сигнал 1) выполняются 16 логических операций, задаваемых теми же переключателями 0...3.

Значения четырехразрядных операндов А и В задаются с помощью генератора слова и в шестнадцатеричном коде отображаются на алфавитно-цифровых индикаторах. На выходах F0...F3 результат суммирования отображается индикатором F. При коде 1111 на этих выходах и при равенстве операндов выход $A=B$ переводится в единичное состояние. Поскольку этот выход представляет собой каскад с открытым коллектором, то на него подается питание +5 В через резистор 1 кОм. Выход $A=B$ совместно с выходом переноса CN+4 и выходом Р подтверждения переноса используются для формирования признаков $A>B$ и $A<B$ с помощью дополнительных логических элементов U1, U2, U3.

Изменяя состояния сигналов на управляющих входах, можно промоделировать большинство функций АЛУ, используемых в микропроцессорах. Приведем перечень этих функций.

Логические функции (на входе М сигнал 1); выполняются поразрядно, переносы не учитываются.

Код 0000 на входах S3, S2, S1, S0; при этом выполняется логическая функция A' — данные со входов А передаются на выходы F с инверсией, может быть использована в команде CMA (здесь и далее используется мнемоника команд микропроцессоров семейства 80xx фирмы Intel);

0001 — $(A+B)'$ — поразрядная операция ИЛИ с инверсией над операндами А и В;

0010 — $A'B$ — операция И инвертированного операнда А и операнда В;

0011 — 0 — нет операции;

0100 — $(AB)'$ — операция И с инверсией;

0101 — B' — инверсия операнда В;

0110 — $A \oplus B$ — операция Исключающее ИЛИ, команда XRA;

0111 — AB' — операция И над операндами А и инверсией В;

1000 — $A'+B$ — операция ИЛИ над инверсией А и операндом В;

1001 — $(A+B)'$ — операция ИЛИ с инверсией;

1010 — B — передача на выход операнда B ;
 1011 — AB — операция И, команда ANA ;
 1100 — 1;
 1101 — $A+B'$ — операция ИЛИ над инверсией B и операндом A ;
 1110 — $A+B$ — операция ИЛИ, команда ORA ;
 1111 — A — передача на выход операнда A .

Арифметические операции ($M=0$) без переноса ($Cp=1$) и с переносом ($Cp=0$, данные приводятся в круглых скобках):

0000 — A — передача на выход операнда ($A+1$ — суммирование операнда с 1 переноса, команда инкремента).

0001 — $A+B$ — операция суммирования без учета переноса, команда $ADD ((A+B)+1$ — суммирование с учетом переноса, команда ADC);

0010 — $A+B'$ — операция суммирования операнда A с инверсией операнда B без учета переноса ($(A+B')+1$ — то же, но с учетом переноса);

00H — -1 (0);

0100 — $A+AB' (A+(AB')+1)$;

0101 — $(A+B)+AB'((A+B)+AB'+1)$;

0H0 — $A-B-1$, команда $SBB (A-B$, команда $SUB)$;

01H — $AB'-1((AB)')$;

1000 — $A+AB(A+B+1)$;

1001 — $A+B$, команда $ADD (A+B+1)$;

1010 — $(A+B')+AB ((A+B')+AB+1)$;

10H — $AB-1 (AB)$;

1100 — $A+A(A+A+1)$;

1101 — $(A+B)+A((A+B)+A+1)$;

1110 — $(A+B')+A ((A+B')+A+1)$;

1111 — $A-1 (A)$

Порядок выполнения

- соберите схему, изображенную на рисунке 1;
- проведите моделирование режимов АЛУ;
- дополните операции без комментариев описанием выполняемых ими функций;
- проанализируйте систему команд микропроцессора 180080 (KP580ИК80) и возможность использования в них логических функций и арифметических операций ИМС 74181.

3. КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Контроль и оценка результатов освоения дисциплины осуществляется преподавателем в процессе проведения контрольной работы, практических занятий, тестирования, а также выполнения обучающимися индивидуальных заданий, исследований. Итоговый контроль проводится в форме дифференцированного зачета.

Результаты обучения (освоенные умения, усвоенные знания)	Формы и методы контроля и оценки результатов обучения
Умения:	
производить арифметические вычисления в двоичной системе	экспертное наблюдение и оценка на практических занятиях
выполнять синтез логических схем	экспертное наблюдение и оценка на практических занятиях
читать условно-графическое обозначение (УГО) и принципиальные схемы	экспертное наблюдение и оценка на практических занятиях
увеличивать адресность и разрядность модулей памяти	экспертное наблюдение и оценка на практических занятиях, контрольной работе, при тестировании
строить функциональные и принципиальные схемы узлов и устройств ЭВМ с соблюдением требований стандартов	экспертное наблюдение и оценка на практических занятиях
производить сравнительную оценку проектируемых схем, с учетом основных параметров	экспертное наблюдение и оценка на практических занятиях
описать работу синтезированного устройства с помощью таблиц	экспертное наблюдение и оценка на практических занятиях

истинности и временных диаграмм	
описать работу синтезированного устройства с помощью таблиц истинности и временных диаграмм	экспертное наблюдение и оценка на практических занятиях
пользоваться технической и справочной литературой	экспертное наблюдение и оценка на практических занятиях
Знания:	
формы представления информации в схемах автоматики и вычислительной техники	экспертное наблюдение и оценка на практических занятиях, контрольной работе, при тестировании, выполнении домашнего задания, на зачете
основы алгебры логики	экспертное наблюдение и оценка на практических занятиях, контрольной работе, при тестировании, выполнении домашнего задания, на зачете
основы проектирования цифровых устройств	экспертное наблюдение и оценка на практических занятиях, контрольной работе, зачете, при тестировании
условно графические обозначения элементов	экспертное наблюдение и оценка на зачете, при тестировании, выполнении домашнего задания
принцип работы элементов памяти	экспертное наблюдение и оценка на практических занятиях, при тестировании, выполнении домашнего задания, на зачете
структуру и организацию микропроцессорных наборов	экспертное наблюдение и оценка при тестировании, выполнении домашнего задания, на зачете
принцип построения и работы типовых узлов ЭВМ	экспертное наблюдение и оценка на практических занятиях при тестировании, выполнении домашнего задания, на зачете
современное состояние и тенденции развития микросхемотехнических средств вычислительной техники	экспертное наблюдение и оценка при тестировании, выполнении домашнего задания, на зачете

4. РЕКОМЕНДОВАННЫЕ ИСТОЧНИКИ

Основные источники:

1. Угрюмов Е. П. «Цифровая схемотехника», СПб, 2014 г.
2. Пятибратов А. П. и др. «Вычислительные системы, сети и телекоммуникации», М., Финансы и статистика, 2012 г.
3. Ерофеев Ю. Н. «Основы Импульсной техники», М., Высшая школа, 1999г.
4. Алексеенко А. Г. «Основы микросхемотехники», М., ЮНИАМЕДИА-СТАЙЛ, 2009 г.

Дополнительные:

- 1.Алексеенко А. Г., Иванухин И. И. «Микросхемотехника», М., Радио и связь, 2010 г.
- 2.Ефремов Б. Д. и др. «Вычислительные машины системы», М., Высшая школа, 2007 г.
3. Новиков Ю. В. «Основы цифровой схемотехники», М., Мир, 2010 г.
4. Каябеков Б.А. Цифровые устройства и микропроцессорные системы: Учебник для техникумов связи. - М.: Горячая линия - Телеком, 2010. - 336с: ил.

Интернет-ресурсы:

www.net.e-publish.ru

www.sut.ru

www.dokanet.net