

РОСЖЕЛДОР
Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
«Ростовский государственный университет путей сообщения»
(ФГБОУ ВПО РГУПС)
Тихорецкий техникум железнодорожного транспорта
(ТТЖТ – филиал РГУПС)

МЕТОДИЧЕСКИЕ УКАЗАНИЯ
ДЛЯ ВЫПОЛНЕНИЯ
ПРАКТИЧЕСКИХ И ЛАБОРАТОРНЫХ ЗАНЯТИЙ
по МДК 01.01 ЦИФРОВАЯ СХЕМОТЕХНИКА

для специальности
09.02.01 Компьютерные системы и комплексы

2015



УТВЕРЖДАЮ

Заместитель директора по
учебной работе:

« 01 » 09 2015г.

 Н.Ю. Шитикова

Методические указания для выполнения практических и лабораторных занятий по МДК 01.01 Цифровая схемотехника разработаны для студентов очной формы обучения специальности 09.02.01 Компьютерные системы и комплексы.

Организация-разработчик: Тихорецкий техникум железнодорожного транспорта – филиал Федерального государственного бюджетного образовательного учреждения высшего профессионального образования «Ростовский государственный университет путей сообщения» (ТТЖТ – филиал РГУПС)

Разработчик:

Чуркина О.Н., преподаватель ТТЖТ- филиала РГУПС

Рецензенты:

А.Н. Исаев, преподаватель ТТЖТ- филиала РГУПС

А.В. Кравцов, преподаватель ТТЖТ- филиала РГУПС

Рекомендованы цикловой комиссией №12 «Специальностей 09.02.01 и 11.02.06».

Протокол заседания №1 от 01 сентября 2015 г.

СОДЕРЖАНИЕ:

Практические занятия:

1. Представление чисел в различных системах счисления.
2. Представление двоичных чисел в прямом, обратном и дополнительном кодах.
3. Выполнение арифметических операций с многоразрядными двоичными числами
4. Построение логических схем по переключательным функциям
5. Преобразование логических выражений
6. Минимизация логических функций
7. Переход от таблицы истинности к СДНФ и СКНФ
8. Построение схем цифровых логических устройств методом синтеза
9. Построение триггеров на основе логических элементов интегральной схемотехники методом синтеза.
10. Построение делителя частоты с заданным коэффициентом деления
11. Построение схемы преобразователя кодов
12. Исследование принципа работы функциональных схем сумматоров
13. Проектирование многоразрядного двоичного сумматора
14. Проектирование ОЗУ заданной ёмкости и структуры

Лабораторные занятия:

1. Исследование работы логических элементов ТТЛ и КМОП.
2. Исследование работы триггеров
3. Исследование принципа работы функциональных схем регистров
4. Исследование функциональных схем шифраторов и дешифраторов
5. Исследование функциональных схем мультиплексоров и демультиплексоров
6. Исследование работы функциональной схемы цифрового компаратора
7. Исследование работы АЦП и ЦАП
8. Изучение структурной схемы микропроцессора

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Данные методические указания предназначены для студентов по выполнению практических занятий согласно программе МДК 01.01 Цифровая схемотехника для студентов специальности 09.02.01 Компьютерные системы и комплексы с целью закрепления теоретических знаний и практических умений.

Цели и задачи методических указаний:

Обучающийся в ходе освоения дисциплины должен:

уметь:

- использовать типовые средства вычислительной техники и программного обеспечения;
- проводить контроль и анализ процесса функционирования цифровых схемотехнических устройств по функциональным схемам.

знать:

- виды информации и способы ее представления в ЭВМ;
- алгоритмы функционирования цифровой схемотехники.

Правила выполнения практических и лабораторных работ

1. Студент должен прийти на занятие подготовленным к выполнению практической или лабораторной работы (проведения лабораторного или практического занятия).

2. Лабораторные и практические занятия должны быть оформлены в виде отчета, с указанием фамилии, инициалов и шифра студента.

3. Отчет о проделанной работе следует выполнять на листах формата А4 с одной стороны листа. В отчете представить результат работы (выполненного задания): решение, графики, схемы, диаграммы, скриншоты (если необходимо).

4. Каждый отчет должен заканчиваться самостоятельными выводами, поскольку студент должен творчески подходить к полученным экспериментальным данным, используя свои теоретические и практические знания.

5. Вспомогательные расчеты можно выполнять на отдельных листах, а при необходимости на листах отчета.

6. Оценку по практической или лабораторной работе студент получает, если:

- расчеты выполнены правильно и в полном объеме;
- сделаны выводы по результатам работы;
- может пояснить выполнение любого этапа работы;
- отчет выполнен в соответствии с требованиями к выполнению работы,
- отвечает на контрольные вопросы на удовлетворительную оценку и выше.

Практическое занятие №1

Представление чисел в различных системах счисления.

Цель работы: изучить представление чисел в различных системах счисления, способы представления чисел в разрядной сетке ЭВМ.

Оборудование: инструкционные карты.

Теоретические сведения:

Системой счисления называется совокупность приемов наименования и записи чисел.

Базисные числа - некоторые символы (слова или знаки), выбираемые для представления чисел.

Все остальные числа получаются в результате каких-либо операций из базисных чисел данной системы счисления

Людьми использовались различные способы записи чисел, которые можно объединить в несколько групп:

- унарные с.с.,
- непозиционные с.с.
- позиционные с.с.

Система называется позиционной, если значение каждой цифры (ее вес) изменяется в зависимости от ее положения (позиции) в последовательности цифр, изображающих число.

Число к единиц какого-либо разряда, объединяемых в единицу более старшего разряда называют основанием позиционной системы счисления, а сама система счисления называется к-ичной.

Базисные числа некоторых систем счисления:

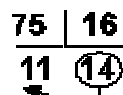
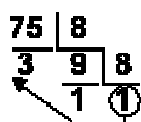
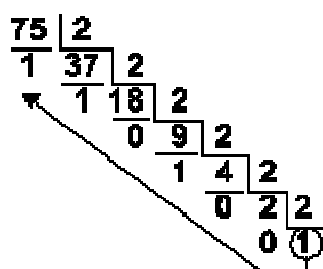
- двоичная (используются цифры 0, 1);
- восьмеричная (используются цифры 0, 1, ..., 7);
- шестнадцатеричная (для первых целых чисел от нуля до девяти используются цифры 0, 1, ..., 9, а для следующих чисел - от десяти до пятнадцати - в качестве цифр используются символы A, B, C, D, E, F).

Перевод чисел из 10-ной системы счисления в другую систему счисления

Правило 1 При переводе целого десятичного числа в систему с основанием q , его необходимо последовательно делить на q до тех пор пока не останется остаток, меньший или равный $q - 1$. Число в системе с основанием q записывается как последовательность остатков от деления, записанных в обратном порядке, начиная с последнего.

Пример:

Перевести число 75 из десятичной системы в двоичную, восьмеричную и шестнадцатеричную:



Ответ: $7510 = 1\ 001\ 0112 = 1138 = 4B16$.

Правило 2 При переводе правильной десятичной дроби в систему счисления с основанием q необходимо сначала саму дробь, а затем дробные части всех последующих произведений последовательно умножать на q , отделяя после каждого умножения целую часть произведения. Число в новой системе счисления записывается как последовательность полученных целых частей произведения. Умножение производится до тех пор, пока дробная часть произведения не станет равной нулю. Это значит, что сделан точный перевод. В противном случае перевод осуществляется до заданной точности (количество знаков после запятой)

Пример:

Перевести число 0,35 из десятичной системы в двоичную, восьмеричную и шестнадцатеричную:

В двоичную

0,	35
	2
0	70
	2
1	40
	2
0	80
	2
1	60
	2
1	20

В восьмеричную

0,	35
	8
2	80
	8
6	40
	8
3	20

В шестнадцатеричную

0,	35
	16
5	60
	16
9	60

Ответ: $0,3510 = 0,010112 = 0,2638 = 0,5916$.

Перевод чисел из любой системы счисления в 10-чную систему счисления

При переводе числа из любой системы счисления в десятичную надо это число представить в виде суммы степеней основания его системы счисления.

Пример

Разряды 3 2 1 0 -1

Число 1 0 1 1, $1_2 = 1 \cdot 2^3 + 1 \cdot 2^1 + 1 \cdot 2^0 + 1 \cdot 2^{-1} = 11,5_{10}$.

Разряды 2 1 0 -1

Число 2 7 6, $5_8 = 2 \cdot 8^2 + 7 \cdot 8^1 + 6 \cdot 8^0 + 5 \cdot 8^{-1} = 190,625_{10}$.

Разряды 2 1 0

Число 1 F 3, $16 = 1 \cdot 16^2 + 15 \cdot 16^1 + 3 \cdot 16^0 = 499_{10}$.

Перевод чисел из 8, 16-чной системы счисления в 2-чную систему счисления

Перевод восьмеричных и шестнадцатеричных чисел в двоичную систему очень прост: достаточно каждую цифру заменить эквивалентной ей двоичной триадой (тройкой цифр) или тетрадой (четверкой цифр).

Пример

$$537,1_8 = 101\ 011\ 111,001_2 ; 1A3,F_{16} = 1\ 1010\ 0011,1111_2$$



Перевод чисел из 2-чной системы счисления в 8,16-чную систему счисления
 Чтобы перевести число из двоичной системы в восьмеричную или шестнадцатеричную, его нужно разбить влево и вправо от запятой на триады (для восьмеричной) или тетрады (для шестнадцатеричной) и каждую такую группу заменить соответствующей восьмеричной (шестнадцатеричной) цифрой.

Пример

$$10101001,10111_2 = 10\ 101\ 001,101\ 110_2 = 251,56_8$$



$$10101001,10111_2 = 1010\ 1001,1011\ 1000_2 = A9,B8_{16}$$



Арифметические действия над числами в любой позиционной системе счисления

Арифметические действия над числами в любой позиционной системе счисления производятся по тем же правилам, что и в десятичной системе счисления, т.к. все они основываются на правилах выполнения действий над соответствующими полиномами.

Пример 1 Сложим числа 15 и 6 в 10-чной, 2-чной, 8-чной, 16-чной системах счисления.

$$\begin{array}{r} 1 \\ + 15 \\ + 6 \\ \hline 21 \\ \hline \end{array} \quad \begin{array}{l} 5+6=11=10+1 \\ 1+1=2 \end{array}$$

$$\begin{array}{r} 111 \\ + 1111 \\ \hline 0110 \\ \hline 10101 \\ \hline \end{array} \quad \begin{array}{l} 1+0=1 \\ 1+1=2=2+0 \\ 1+1+1=3=2+1 \\ 1+1=2=2+0 \end{array}$$

$$\begin{array}{r} 1 \\ + 17 \\ + 6 \\ \hline 25 \\ \hline \end{array} \quad \begin{array}{l} 7+6=13=8+5 \\ 1+1=2 \end{array}$$

$$\begin{array}{r} 1 \\ + F \\ + 6 \\ \hline 15 \\ \hline \end{array} \quad \begin{array}{l} 15+6=21=16+5 \end{array}$$

Пример 2 Вычтем число 59,75 из числа 201,25 в 10-чной, 2-чной, 8-чной, 16-чной системах счисления.

$$\begin{array}{r}
 1 \quad 1 \\
 - \quad 201,25 \\
 \quad 59,75 \\
 \hline
 141,50 \\
 \begin{array}{l}
 5-5=0 \\
 10+2-7-5 \\
 10-9=1 \\
 9-5=4 \\
 7-1=1
 \end{array} \\
 \hline
 1 \quad 1 \quad 1 \\
 - \quad 311,2 \\
 \quad 73,6 \\
 \hline
 215,4 \\
 \begin{array}{l}
 8+2-5=4 \\
 8-3=5 \\
 8-7=1
 \end{array} \\
 \hline
 \end{array}$$

$$\begin{array}{r}
 1 \quad 1 \quad 1 \quad \text{Заемы} \\
 - \quad 11001001,01 \\
 \quad 00111011,11 \\
 \hline
 10001101,10 \\
 \begin{array}{l}
 1-0=1 \\
 0-0=0 \\
 1-1=0 \\
 1-1=0 \\
 2-1=1
 \end{array} \\
 \hline
 1 \quad 1 \\
 - \quad \text{С } 9,4 \\
 \quad 3B,C \\
 \hline
 8D,8 \\
 \begin{array}{l}
 16+4-12=8 \\
 16+8-11=13=D_{16} \\
 12-1-3=8
 \end{array} \\
 \hline
 \end{array}$$

Пример 3 Перемножим числа 115 и 51 в 10-чной, 2-чной, 8-чной системах счисления.

$$\begin{array}{r}
 \times \quad 115 \\
 \quad 51 \\
 \hline
 115 \\
 575 \\
 \hline
 5865
 \end{array}$$

$$\begin{array}{r}
 \times \quad 1110011 \\
 \quad 110011 \\
 \hline
 1110011 \\
 1110011 \\
 \hline
 1011011101001
 \end{array}$$

$$\begin{array}{r}
 \times \quad 163 \\
 \quad 63 \\
 \hline
 531 \\
 1262 \\
 \hline
 13351
 \end{array}$$

Пример 4 Разделим число 30 на число 6 в 10-чной, 2-чной, 8-чной системах счисления.

$$\begin{array}{r}
 - \quad 30 \overline{) 6} \\
 \quad 30 \overline{) 5} \\
 \hline
 0
 \end{array}$$

$$\begin{array}{r}
 - \quad 11110 \overline{) 110} \\
 \quad 110 \overline{) 101} \\
 \hline
 110 \\
 110 \\
 \hline
 0
 \end{array}$$

$$\begin{array}{r}
 - \quad 36 \overline{) 6} \\
 \quad 36 \overline{) 5} \\
 \hline
 0
 \end{array}$$

Упражнения к практическому занятию:

1. Расположите следующие числа в порядке возрастания:

748, 1100102, 7010, 3816

6E16, 1428, 11010012, 10010

2. В какой системе счисления выполнены следующие сложения? найдите основания каждой системы.

$$98+89=121$$

$$1345+2178=35223$$

$$765+576+677=2462$$

$$98+56+79=167$$

$$213*3=1144$$

3. Вычислите значение выражения:

$$2568+10110,12*(608+1210)-1F16$$

$$1AD16-1001011002:10102+2178$$

$$101010+(10616-110111012)*128$$

$$10112*11002:148+(1000002-408)$$

4. Число 110101112 соответствует числу в 8-чной системе счисления:
4948

1258

768

3278

998

5. Число N в двоичной системе счисления записывается в виде некоторой последовательности из десяти двоичных символов. Число единиц в двоичной записи числа N на 7 больше, чем число единиц в двоичной записи числа $N+1$. Число $N+1$ равно:

1100000001

1100000010

1100110000

1100000000

Контрольные вопросы:

1. Системы счисления. Основные понятия
2. Унарные, позиционные и непозиционные системы счисления
3. Аддитивные системы счисления.
4. Арифметические операции над числами в разных системах счисления.
5. Перевод чисел из одной системы счисления в другую.
6. Смешанные системы счисления.

Практическое занятие №2

Представление двоичных чисел в прямом, обратном и дополнительном кодах.

Цель работы: научиться выполнять операции над числами с использованием обратного и дополнительного кодов.

Оборудование: инструкционные карты.

Теоретические сведения:

Сложение (вычитание). Операция вычитания приводится к операции сложения путем преобразования чисел в обратный или дополнительный код. Пусть числа $A \geq 0$ и $B \geq 0$, тогда операция алгебраического сложения выполняется в соответствии с табл.1

Требуемая операция	Необходимое преобразование
$A+B$	$A+B$
$A-B$	$A+(-B)$
$-A+B$	$(-A)+B$
$-A-B$	$(-A)+(-B)$

При выполнении сложения цифр необходимо соблюдать следующие правила:

1. Слагаемые должны иметь одинаковое число разрядов. Для выравнивания разрядной сетки можно дописывать незначащие нули слева к целой части числа и незначащие нули справа к дробной части числа.
2. Знаковые разряды чисел участвуют в сложении так же, как и значащие.
3. Необходимые преобразования кодов производятся с изменением знаков чисел.
4. При преобразовании единицы переноса из старшего знакового разряда, в случае использования обратного кода (ОК), эта единица складывается с младшим числовым разрядом. При использовании дополнительного кода (ДК) единица переноса теряется. Знак результата формируется автоматически, результат представляется в том коде, в котором представлены исходные слагаемые.

Пример 1. Сложить два числа: $A_{10}=7$; $B_{10}=16$.

$A_2=+111$

$B_2=+10000$

Исходные числа имеют различную разрядность, необходимо провести выравнивание разрядной сетки:

$[A_2]_{\text{п}}=[A_2]_{\text{ок}}=[A_2]_{\text{дк}}=0:00111$

$[B_2]_{\text{п}}=[B_2]_{\text{ок}}=[B_2]_{\text{дк}}=0:10000$

Сложение в обратном и дополнительном коде дает один и тот же результат:

$$\begin{array}{r} 0:00111 \\ + 0:10000 \\ \hline C_2=0:10111_2 \\ C_{10}=+23. \end{array}$$

Задание 1: Сложить два числа.

№ по списку в журнале	1 вариант	2 вариант	3 вариант	4 вариант
1,3,5,14	$A_{10}=17; B_{10}=6.$	$A_{10}=9; B_{10}=21.$	$A_{10}=11; B_{10}=19.$	$A_{10}=34; B_{10}=5.$
2,4,6,8,	$A_{10}=7; B_{10}=16.$	$A_{10}=10; B_{10}=27.$	$A_{10}=8; B_{10}=35.$	$A_{10}=34; B_{10}=56.$
7,9,11,13	$A_{10}=6; B_{10}=13.$	$A_{10}=25; B_{10}=52.$	$A_{10}=7; B_{10}=14.$	$A_{10}=6; B_{10}=26.$
15,17,19,30	$A_{10}=9; B_{10}=21.$	$A_{10}=10; B_{10}=27.$	$A_{10}=9; B_{10}=21.$	$A_{10}=9; B_{10}=21.$
10,12,16,18	$A_{10}=9; B_{10}=21.$	$A_{10}=9; B_{10}=21.$	$A_{10}=17; B_{10}=6.$	$A_{10}=10; B_{10}=27.$
20,22,23	$A_{10}=10; B_{10}=7.$	$A_{10}=9; B_{10}=21.$	$A_{10}=9; B_{10}=21.$	$A_{10}=6; B_{10}=26.$
21,24,25,26	$A_{10}=9; B_{10}=21.$	$A_{10}=10; B_{10}=27.$	$A_{10}=17; B_{10}=6.$	$A_{10}=9; B_{10}=21.$
27,1,28,29	$A_{10}=34; B_{10}=5.$	$A_{10}=9; B_{10}=21.$	$A_{10}=9; B_{10}=21.$	$A_{10}=17; B_{10}=6.$

Пример 2. Сложить два числа $A_{10}=+16; B_{10}=-7$ в ОК и ДК.

В соответствии с таблицей 1 должна быть реализована зависимость $A+(-B)$, в которой второй член преобразуется с учетом знака

$$\begin{aligned} [A_2]_{\text{п}} &= [A_2]_{\text{ок}} = [A_2]_{\text{дк}} = 0:10000 \\ [B_2]_{\text{п}} &= 1:111 = 1:00111 \\ [B_2]_{\text{ок}} &= 1:11000 \\ [B_2]_{\text{дк}} &= 1:11001 \end{aligned}$$

Сложение в ОК

$$\begin{array}{r} [A_2]_{\text{ок}} = 0:10000 \\ + [B_2]_{\text{ок}} = 1:11000 \\ \hline 0:01000 \text{ (перенос)} \\ + \quad \quad 1 \\ \hline 0:01001 \end{array}$$

$$C_2=0:01001$$

$$C_{10}=+9.$$

Сложение в ДК

$$\begin{array}{r} [A_2]_{\text{дк}} = 0:10000 \\ + [B_2]_{\text{дк}} = 1:11001 \\ \hline 0:01001 \end{array}$$

$$C_2=0:01001$$

$$C_{10}=+9$$

При сложении чисел в ОК и ДК были получены переносы в знаковый разряд и из знакового. В случае ОК перенос из знакового разряда требует дополнительного прибавления единицы младшего разряда (см.п.4). В случае ДК этот перенос не требуется.

Задание 2: Сложить два числа.

№ по списку в журнале	1 вариант	2 вариант	3 вариант	4 вариант
1,3,5,14	$A_{10}=17; B_{10}=-6.$	$A_{10}=9; B_{10}=-21.$	$A_{10}=11; B_{10}=-19$	$A_{10}=34; B_{10}=-5.$
2,4,6,8,	$A_{10}=-7; B_{10}=16.$	$A_{10}=-10; B_{10}=27.$	$A_{10}=-8; B_{10}=35.$	$A_{10}=-34; B_{10}=56$
7,9,11,13	$A_{10}=-6; B_{10}=13.$	$A_{10}=25; B_{10}=-52.$	$A_{10}=7; B_{10}=-14.$	$A_{10}=6 B_{10}=-26.$
15,17,19,30	$A_{10}=-9; B_{10}=21.$	$A_{10}=-10; B_{10}=27.$	$A_{10}=-9; B_{10}=21.$	$A_{10}=-9; B_{10}=21.$
10,12,16,18	$A_{10}=-9; B_{10}=21.$	$A_{10}=9; B_{10}=-21.$	$A_{10}=17; B_{10}=-6.$	$A_{10}=10; B_{10}=-27$
20,22,23	$A_{10}=10; B_{10}=-7.$	$A_{10}=-9; B_{10}=21.$	$A_{10}=-9; B_{10}=21.$	$A_{10}=-6 B_{10}=26.$
21,24,25,26	$A_{10}=-9; B_{10}=21.$	$A_{10}=10; B_{10}=-27.$	$A_{10}=17; B_{10}=-6.$	$A_{10}=9; B_{10}=-21.$
27,1,28,29	$A_{10}=34; B_{10}=-5.$	$A_{10}=-9; B_{10}=21.$	$A_{10}=-9; B_{10}=21.$	$A_{10}=17; B_{10}=-6.$

Контрольные вопросы:

1. Перечислите формы представления информации в ЭВМ.
2. Поясните отличие знаковых разрядов от значащих.
3. Поясните порядок переноса в ОК и ДК.

Практическое занятие №3

Выполнение арифметических операций с многоразрядными двоичными числами

Цель работы: научиться выполнять арифметические операции над числами
Оборудование: инструкционные карты.

Теоретические сведения:

Коды обоих чисел суммируются по правилам двоичного сложения разряд за разрядом, включая старшие знаковые разряды кодов. Если появляется перенос из знакового разряда, он отбрасывается. Отрицательный результат, полученный в дополнительном коде, должен быть переведен в прямой код.

Пример 7.

$$\begin{array}{rcl} \text{А) слагаемое – А пр.к.} & 0,1101100 & \\ \text{слагаемое – В пр.к.} & +1,1000110 & \\ \text{А д.к.} & 0,1101100 & \\ \text{В д.к.} & + 1,0111010 & \\ \text{А + В д.к.} & 0,0100110 & \text{- результат} \\ & 11111000 & \text{- перенос} \\ & \text{перенос игнорируется} & \\ & 0,0100110 & \text{- сумма} \end{array}$$

$$\begin{array}{rcl} \text{Б) слагаемое А пр.к.} & 0,0110110 & \\ \text{слагаемое В пр.к.} & + 1,1011010 & \\ \text{А д.к.} & 0,0110110 & \\ \text{В д.к.} & + 1,0100110 & \\ \text{А + В д.к.} & 1,1011100 & \\ \text{А + В пр.к.} & 1,0100100 & \text{- сумма} \end{array}$$

Если оба слагаемых имеют одинаковый знак, то может случиться, что их алгебраическая сумма выйдет за границы диапазона чисел, представленных в коде данной разрядности. Это называется переполнением разрядной сетки. При переполнении возможны два случая: либо имеется перенос в знаковый разряд и нет переноса из знакового разряда, либо нет переноса в знаковый разряд и есть перенос из знакового разряда.

Для выполнения операции вычитания предварительно вычитаемое переводится в дополнительный код, а затем осуществляется суммирование кодов чисел.

Пример 8

уменьшаемое	А пр.к.	0,1101101
вычитаемое	В пр.к.	- 0,1000110

А д.к. 0,1101101

В д.к. + 1,0111010

А – В д.к.	10,0100111	- разность
	1111000	- перенос

В арифметико-логическом устройстве ЭВМ арифметические операции выполняются над операндами, представленными с фиксированной или плавающей точкой, в двоичной либо в десятичной системах счисления.

Алгоритм реализации операции сложения двоичных чисел с фиксированной точкой состоит из следующих этапов:

- 1) анализа знакового разряда;
- 2) перевода мантиисы в обратный или дополнительный код, если знак результата отрицательный;
- 3) суммирование кодов слагаемых; анализа переполнения разрядной сетки; при переполнении разрядной сетки результат неверен; преобразования результата в прямой код, если знак результата отрицательный.

Алгоритм выполнения сложения двоичных чисел с плавающей точкой гораздо сложнее, чем с фиксированной точкой, и состоит из следующих этапов:

- 1) выравнивание порядков путем вычитания порядка меньшего числа из порядка большего числа с учетом знака порядка и сдвиг мантиисы на разность порядков;
- 2) анализ знака мантиисы, перевод мантиисы в обратный или дополнительный код, если знак числа отрицателен;
- 3) суммирование кодов мантиисы слагаемых; анализ наличия переполнения разрядной сетки мантиисы результата;
- 4) преобразование результатов в прямой код, если знак результата отрицательный;
- 5) нормализация результата влево с уменьшением порядка, если результат получился ненормализованным.

Задание:

1. Сложить два числа в двоичной системе счисления. Результат проверить при помощи десятичной системы счисления.

- ✓ 1101100000 + 10110110
- ✓ 101110111 + 1000100001
- ✓ 1001000111,01 + 100001101,101
- ✓ 1010101 + 10000101

2. Выполнить вычитание двух чисел в двоичной системе счисления. Результат проверить при помощи десятичной системы счисления.

- ✓ 101010000,10111 - 11001100,01
- ✓ 1011001001 - 1000111011

✓ $1110000110 - 10111101$

✓ $1001000011 - 10110111$

3. Выполнить умножение двух чисел в двоичной системе счисления.
Результат проверить при помощи десятичной системы счисления.

✓ $1011001 \cdot 1011011$

✓ $11001 \cdot 1011100$

✓ $1010101 \cdot 1011001$

✓ $101011 \cdot 100111$

4. Выполнить деление в двоичной системе счисления.

✓ $100111000 : 11000 = 1101$

✓ $111111000 : 101010 = 1100$

Контрольные вопросы:

1. По каким правилам производится сложение в двоичной системе счисления?

2. Чем похожи операции сложения в двоичной и десятичной системе счисления?

3. По каким правилам производится вычитание в двоичной системе счисления?

4. По каким правилам производится умножение в двоичной системе счисления?

5. По каким правилам производится деление в двоичной системе счисления?

Практическое занятие №4

Построение логических схем по переключательным функциям

Цель работы: получить практические навыки синтеза логических схем.

Оборудование: инструкционные карты.

Теоретические сведения:

По заданной функции f требуется построить схему, реализующую данную функцию. Задача синтеза решается неоднозначно. Можно поставить в соответствие заданной функции f целое множество схем. Для построения логической схемы необходимо элементы, предназначенные для выполнения логических операций, указанных в логической функции, располагать в порядке, указанном в булевом выражении.

Пример. Построить логическую схему устройства, реализующего логическую функцию $f = \overline{x_1}x_2x_3 \vee x_1\overline{x_2}x_3 \vee x_1x_2\overline{x_3} \vee x_1x_2x_3$. (рис. 1)

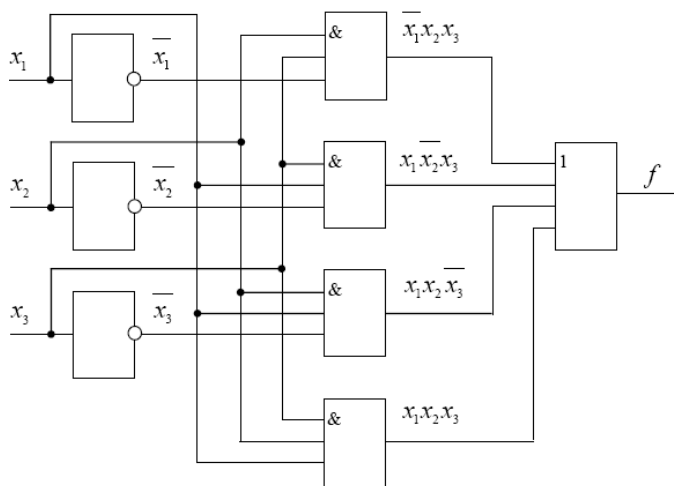


Рис. 1. Пример логической схемы устройства

Синтез логических устройств в заданном базисе.

С целью уменьшения номенклатуры используемых микросхем, часто пользуются функционально полной системой в составе двух логических элементов выполняющих операции И-НЕ, ИЛИ-НЕ. Любую логическую функцию можно записать в заданном базисе логических элементов. Если задан базис И-НЕ, то путем двойного инвертирования исходного выражения или его части и применения теорем де Моргана логическая функция приводится к виду, содержащему только операции логического умножения и инвертирования. Если же задан базис ИЛИ-НЕ, исходную логическую функцию теми же приемами приводят к виду, содержащему только операции логического сложения и инверсии. Далее логическое выражение записывается через условные обозначения выбранных операций.

Пример. Заданную функцию f перевести в базисы И-НЕ и ИЛИ-НЕ. Исходная ДНФ в базисе И-НЕ имеет вид:

$$f = x_2x_4 + \overline{x_1x_3x_4} + \overline{x_1x_2x_3} = x_2x_4 + \overline{x_1x_3x_4} + \overline{x_1x_2x_3} =$$

$$= (\overline{x_2x_4})(\overline{\overline{x_1x_3x_4}})(\overline{\overline{x_1x_2x_3}}) = (\overline{x_2} | \overline{x_4})(\overline{x_1} | \overline{x_3} | \overline{x_4})(\overline{x_1} | \overline{x_2} | \overline{x_3}).$$

Аналогично, КНФ в базисе ИЛИ-НЕ имеет вид:

$$f = (\overline{x_1 + x_4})(\overline{x_1 + x_2 + x_3})(\overline{x_2 + x_3 + x_4}) =$$

$$= (\overline{x_1 + x_4}) + (\overline{x_1 + x_2 + x_3}) + (\overline{x_2 + x_3 + x_4}) =$$

$$= (x_1 \downarrow x_4) \downarrow (\overline{x_1} \downarrow \overline{x_2} \downarrow \overline{x_3}) \downarrow (\overline{x_2} \downarrow \overline{x_3} \downarrow \overline{x_4}).$$

Задание 1.

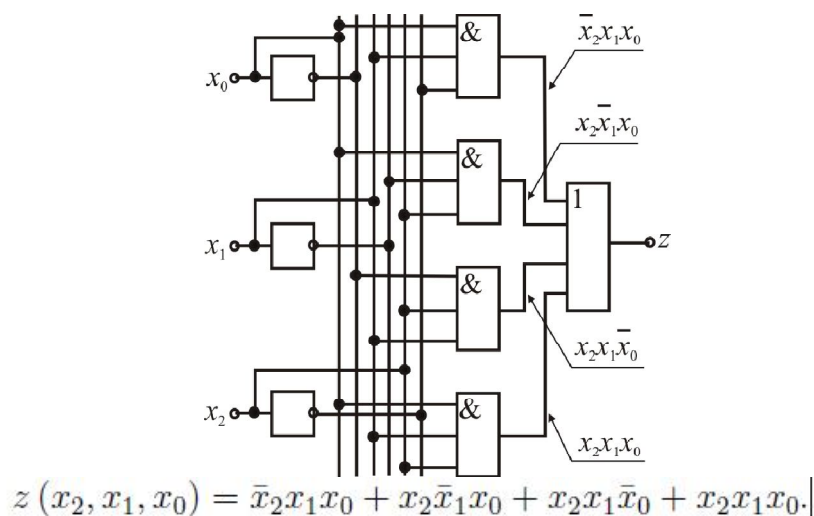
1. Построить структурную схему логического устройства по логической функции трех переменных равной «1», если хотя бы две входные переменные равны «1».

2. Построить структурную схему логического устройства по логической функции трех переменных равной «0», если хотя бы две входные переменные равны «0».

3. Построить структурную схему логического устройства по логической функции четырех переменных равной «1», если хотя бы три входные переменные равны «1».

Пример. Построить структурную схему (1).

Решение: Построить ТИ. Для реализации ФАЛ нам понадобятся три ЛЭ, реализующих операцию НЕ, т.к. исходная ФАЛ формируется тремя переменными, операция дизъюнкции должна быть выполнена четыре раза над тремя переменными, т.о. понадобятся четыре ЛЭ реализующих операцию ЗИ. Последней выполняется операция конъюнкции над четырьмя выражениями, требуется ЛЭ реализующий операцию 4 ИЛИ.



Задание 2.

Пример. Пусть логическая функция задана выражением

$$f = x_1 x_4 + \overline{(x_1 + x_3 + x_4)}(x_2 + x_3)$$

Привести логическую функцию в базис И-НЕ, ИЛИ-НЕ.

а) приводим функцию к базису И-НЕ

$$f = f_1 + \overline{f_2 f_3} = f_1 + \overline{\overline{\overline{f_2 f_3}}} = f_1 + \overline{\overline{(f_2 | f_3)}} = f_1 + \overline{\overline{\overline{(f_2 | f_3)}}} = \overline{\overline{f_1}} + \overline{\overline{\overline{(f_2 | f_3)}}} = \overline{\overline{f_1}} | (\overline{\overline{f_2}} | \overline{\overline{f_3}});$$

$$\overline{\overline{f_1}} = \overline{\overline{x_1 x_4}} = x_1 | x_4;$$

$$\overline{\overline{f_2}} = \overline{\overline{x_1 + x_3 + x_4}} = \overline{\overline{x_1 x_3 x_4}} = \overline{\overline{x_1 x_3 x_4}} = x_1 | x_3 | x_4;$$

$$\overline{\overline{f_3}} = \overline{\overline{x_2 + x_3}} = \overline{\overline{x_2 x_3}} = \overline{\overline{x_2 x_3}} = x_2 | x_3;$$

$$f = (x_1 | x_4) | ((x_1 | x_3 | x_4) | (x_2 | x_3)).$$

б) приводим функцию к базису ИЛИ-НЕ

$$f = f_1 + \overline{f_2 f_3} = f_1 + \overline{\overline{\overline{f_2 f_3}}} = f_1 + \overline{\overline{(f_2 + f_3)}} = f_1 + \overline{\overline{\overline{(f_2 + f_3)}}} = f_1 \downarrow (\overline{\overline{f_2}} \downarrow \overline{\overline{f_3}});$$

$$\overline{\overline{f_1}} = \overline{\overline{x_1 x_4}} = \overline{\overline{x_1 + x_4}} = x_1 \downarrow x_4;$$

$$\overline{\overline{f_2}} = \overline{\overline{x_1 + x_3 + x_4}} = \overline{\overline{x_1 + x_3 + x_4}} = x_1 \downarrow x_3 \downarrow x_4;$$

$$\overline{\overline{f_3}} = \overline{\overline{x_2 + x_3}} = \overline{\overline{x_2 + x_3}} = x_2 \downarrow x_3 \Rightarrow \overline{\overline{f_3}} = x_2 \downarrow x_3;$$

$$f = (x_1 \downarrow x_4) \downarrow ((x_1 \downarrow x_3 \downarrow x_4) \downarrow (x_2 \downarrow x_3)).$$

Индивидуальные задания

Синтезировать в базисе И-НЕ и ИЛИ-НЕ устройства, заданные логической функцией:

$$1) f(x_1, x_2, x_3, x_4) = x_1 x_2 \bar{x}_3 \vee \bar{x}_1 x_3 x_4 \vee \overline{(x_1 x_2)}(\overline{x_3 x_4});$$

$$2) f(x_1, x_2, x_3, x_4) = \overline{(x_1 \vee x_2)} x_3 x_4;$$

$$3) f(x_1, x_2, x_3, x_4) = \overline{(x_1 x_2)} \vee x_3 \vee x_4;$$

$$4) f(x_1, x_2, x_3, x_4) = x_1 \vee x_2 \vee \bar{x}_3 \vee \bar{x}_4;$$

$$5) f(x_1, x_2, x_3, x_4) = x_1 x_2 \bar{x}_3 \bar{x}_4;$$

$$6) f(x_1, x_2, x_3, x_4) = x_1 \vee x_2 \vee \bar{x}_3 \vee x_4 \vee \overline{x_1 x_2};$$

$$7) f(x_1, x_2, x_3, x_4) = (x_1 \vee \bar{x}_2)(x_3 \vee \bar{x}_4);$$

$$8) f(x_1, x_2, x_3, x_4) = x_1 x_2 x_3 \vee \bar{x}_1 x_2 \bar{x}_4 \vee \overline{(\bar{x}_1 \vee x_2)}(\overline{x_3 \vee x_4});$$

$$9) f(x_1, x_2, x_3, x_4) = x_1 x_2 x_3 \vee \overline{x_1 x_2 x_3 \vee x_4};$$

$$10) f(x_1, x_2, x_3, x_4) = (x_1 \vee x_2) \bar{x}_3 \vee \overline{x_1 x_2 x_3 x_4};$$

Контрольные вопросы:

1. Поясните задачу синтеза КЦ.
2. Понятие базиса.

Практическое занятие №5

Преобразование логических выражений

Цель работы: научиться преобразовывать логические выражения с помощью законов алгебры логики.

Оборудование: инструкционные карты.

Теоретические сведения:

Закон двойного отрицания (двойное отрицание исключает отрицание):

$$A = \overline{\overline{A}}.$$

1. Переместительный (коммутативный) закон:

- для логического сложения: $A \vee B = B \vee A$;
- для логического умножения: $A \& B = B \& A$.

Результат операции над высказываниями не зависит от того, в каком порядке берутся эти высказывания.

2. Сочетательный (ассоциативный) закон:

- для логического сложения: $(A \vee B) \vee C = A \vee (B \vee C)$;
- для логического умножения: $(A \& B) \& C = A \& (B \& C)$.

При одинаковых знаках скобки можно ставить произвольно или вообще опускать.

3. Распределительный (дистрибутивный) закон:

- для логического сложения: $(A \vee B) \& C = (A \& C) \vee (B \& C)$;
- для логического умножения: $(A \& B) \vee C = (A \vee C) \& (B \vee C)$.

Закон определяет правило выноса общего высказывания за скобку.

4. Закон общей инверсии (законы де Моргана):

- для логического сложения: $\overline{A \vee B} = \overline{A} \& \overline{B}$;
- для логического умножения: $\overline{A \& B} = \overline{A} \vee \overline{B}$

5. Закон идемпотентности (от латинских слов *idem* — тот же самый и *potens* — сильный; дословно — равносильный):

- для логического сложения: $A \vee A = A$;
- для логического умножения: $A \& A = A$.

Закон означает отсутствие показателей степени.

6. Законы исключения констант:

- для логического сложения: $A \vee 1 = 1$, $A \vee 0 = A$;
- для логического умножения: $A \& 1 = A$, $A \& 0 = 0$.

7. Закон противоречия:

- $A \& \bar{A} = 0$.

Невозможно, чтобы противоречащие высказывания были одновременно истинными.

8. Закон исключения третьего:

- $A \vee \bar{A} = 1$.

Из двух противоречащих высказываний об одном и том же предмете одно всегда истинно, а второе — ложно, третьего не дано.

9. Закон поглощения:

- для логического сложения: $A \vee (A \& B) = A$;
- для логического умножения: $A \& (A \vee B) = A$.

Знание законов логики позволяет проверять правильность рассуждений и доказательств. Основываясь на законах, можно выполнять упрощение сложных логических выражений. Такой процесс замены сложной логической функции более простой, но равносильной ей, называется минимизацией функции.

Некоторые преобразования логических формул похожи на преобразования формул в обычной алгебре (вынесение общего множителя за скобки, использование переместительного и сочетательного законов и т.п.), другие — основаны на свойствах, которыми не обладают операции обычной алгебры (использование распределительного закона для конъюнкции, законов поглощения, склеивания, де Моргана и др.).

Нарушения законов логики приводят к логическим ошибкам и вытекающим из них противоречиям.

Упрощение формул.

Пример 1. Упростить формулу $(A \vee B) \& (A \vee C)$.

Решение:

1. Раскроем скобки: $(A \vee B) \& (A \vee C) = A \& A \vee A \& C \vee B \& A \vee B \& C$;
2. По закону идемпотентности $A \& A = A$, следовательно,
 $A \& A \vee A \& C \vee B \& A \vee B \& C = A \vee A \& C \vee B \& A \vee B \& C$;
3. В высказываниях A и $A \& C$ вынесем за скобки A и используя свойство $A + 1 = 1$, получим

$$A \vee A \& C \vee B \& A \vee B \& C = A \& (1 \vee C) \vee B \& A \vee B \& C = A \vee B \& A \vee B \& C;$$

4. Аналогично предыдущему пункту вынесем за скобки высказывание A.
 $A \vee B \& A \vee B \& C = A \& (1 \vee B) \vee B \& C = A \vee B \& C.$

Таким образом, мы доказали закон дистрибутивности.

Всякую формулу можно преобразовать так, что в ней не будет отрицаний сложных высказываний - все отрицания будут применяться только к простым высказываниям.

Задание:

1. Упростить выражения $F = (A \vee B) \wedge (A \vee B).$
2. Упростить выражения $\overline{AB} + \overline{B}, \quad \overline{BC} + \overline{C}, \quad \overline{AC} + \overline{BC}$ так, чтобы в полученных формулах не содержалось отрицания сложных высказываний.

Контрольные вопросы:

1. Для чего применяются законы алгебры логики.
2. Поясните порядок действий при преобразовании логических выражений.

Практическое занятие №6

Минимизация логических функций

Цель работы: изучить методы минимизации булевых функций.

Оборудование: инструкционные карты.

Теоретические сведения:

1. Метод карт Карно

Метод Карно основан на законе склеивания. Склеиваются наборы, отличающиеся друг от друга значением одного разряда. Такие наборы называются соседними. Карно закодировал клетки своей карты так, что в соседних клетках оказались соседние, а значит, склеивающиеся наборы. Соседними могут быть не только отдельные клетки, которые мы назовем элементарными квадратами Карно, но и целые группы соседних клеток (назовем их прямоугольниками Карно). Под прямоугольником Карно будем понимать некоторую, зачастую разрозненную фигуру покрытия, все соседние клетки которой закодированы соседними наборами. Например, на поле карты для 4-х переменных изображён прямоугольник Карно, состоящий из четырёх элементарных квадратов Карно, описываемых наборами $x_4x_3x_2x_1$, $x_4x_3x_2x_1$, $x_4x_3x_2x_1$, $x_4x_3x_2x_1$. Если над логической суммой этих четырёх наборов произвести последовательно операции склеивания, то мы аналитически получим результат в виде импликанты (под импликантой будем понимать неполный набор) x_3x_1 . Карта Карно позволяет получить этот результат графически значительно быстрее и проще. Для решения этой задачи используем алгоритм графической минимизации. Кстати говоря, сам Карно никакого алгоритма не предложил.

Правило склеивания. Карты Карно

Упрощение выражений булевых функций (минимизация) основывается на понятии несущественности переменных. Переменная называется несущественной на паре наборов, если при изменении ее значения на противоположное булева функция сохраняет свое значение.

Например, для булевой функции трех переменных, $f(1,3,5,6,7)=1$, которая рассматривалась в подразделе 1.3 "Теории переключательных схем", 6-я и 7-я конъюнкции имеют вид: $x_1x_2x_3$, $x_1x_2x_3$. По дистрибутивному закону:

$$x_1x_2x_3 \vee x_1x_2x_3 = x_1x_2 (x_3 \vee x_3) = x_1x_2.$$

Таким образом, две конъюнкции, содержащие несущественную переменную, заменяются одной, в которой несущественная переменная отсутствует.

В кубическом виде склеивание имеет следующую интерпретацию: $\begin{matrix} 1 & 1 \\ 0 & 1 \end{matrix} \begin{matrix} 1 & 1 \end{matrix} = 1 \begin{matrix} 1 & 1 \end{matrix} X$, что соответствует конъюнкции x_1x_2 .

Приведем основные определения, используемые при минимизации булевых функций. Данные определения используют понятия нормальных (канонических) форм булевых функций, введенные в подразделе 1.3.

Число переменных, входящих в элементарную конъюнкцию (для ДНФ) или в элементарную дизъюнкцию (для КНФ) называется ее рангом.

В основе любых методов минимизации лежит операция склеивания. Два элементарных произведения одного ранга (для ДНФ) или элементарных сумм одного ранга (для КНФ) склеиваются, если они различаются только по одной переменной.

Операция $Ax \vee A = A$ называется полным склеиванием, а операция $Ax \vee A = A \vee Ax \vee A$ - неполным склеиванием (для ДНФ).

Операция $(A \vee x) \cdot (A \vee) = A$ называется полным склеиванием, а операция $(A \vee x) \cdot (A \vee) = A \cdot (A \vee x) \cdot (A \vee)$ - неполным склеиванием (для КНФ).

Например, полное склеивание $(x_1 \vee x_2 \vee x_3) \cdot (x_1 \vee x_2 \vee 3) = x_1 \vee x_2$;

Неполное склеивание $x_1x_2x_3 \vee x_1x_23 = x_1x_2 \vee x_1x_2x_3 \vee x_1x_23$.

Импликантой называется элементарное произведение, равное 1 на одном или нескольких наборах, где данная функция равна 1, и равное 0 на всех наборах, где данная функция равна 0. Импликанта покрывает один или несколько минтермов рассматриваемой булевой функции. Обычно, импликанта - это результат склеивания соответствующих минтермов или импликант.

Простая импликанта - это импликанта, которая содержит хотя бы минтерм функции, но перестает быть импликантой после удаления любого аргумента (иными словами, это импликанта, к которой не нельзя применить операцию склеивания).

Сокращенная ДНФ - это дизъюнкция всех простых импликант.

Существенная импликанта - это простая импликанта, образованная склеиванием таких минтермов, что по крайней мере для одного из них эта операция была единственной. Существенные импликанты образуют ядро функции.

Тупиковая ДНФ - это дизъюнкция простых импликант, из которых ни одна не является лишней.

МДНФ (минимальная ДНФ) - тупиковая ДНФ с минимальным числом вхождений переменных (минимальным числом букв) по сравнению с другими тупиковыми формами этой функции.

Имплицентой называется элементарная логическая сумма, равная 0 на одном или нескольких наборах, где данная функция равна 0, и равная 1 на всех наборах, где данная функция равна 1. Имплицента покрывает один или несколько макстермов рассматриваемой булевой функции. Обычно, имплицента - это результат склеивания соответствующих макстермов.

Простая имплицента - это имплицента, которая содержит хотя бы макстерм функции, но перестает быть имплицентой после удаления любого аргумента (иными словами, это имплицента, к которой не нельзя применить операцию склеивания).

Сокращенная КНФ - это конъюнкция всех простых имплицент.

Существенная имплицента - это простая имплицента, образованная склеиванием таких макстермов, что по крайней мере для одного из них эта

операция была единственной. Существенные имплиценты образуют ядро функции.

Тупиковая КНФ - это конъюнкция простых имплицентов, из которых ни одна не является лишней.

МКНФ (минимальная КНФ) - тупиковая КНФ с минимальным числом вхождений переменных (минимальным числом букв) по сравнению с другими тупиковыми формами этой функции.

Правила минимизации с использованием карт Карно

1. В карте Карно группы единиц (для получения ДНФ) и группы нулей (для получения КНФ) необходимо обвести четырехугольными контурами. Внутри контура должны находиться только одноименные значения функции. Этот процесс соответствует операции склеивания или нахождения импликант данной функции.

2. Количество клеток внутри контура должно быть целой степенью двойки (1, 2, 4, 8, 16...).

3. При проведении контуров крайние строки карты (верхние и нижние, левые и правые), а также угловые клетки, считаются соседними (для карт до 4-х переменных).

4. Каждый контур должен включать максимально возможное количество клеток. В этом случае он будет соответствовать простой импликанте.

5. Все единицы (нули) в карте (даже одиночные) должны быть охвачены контурами. Любая единица (нуль) может входить в контуры произвольное количество раз.

6. Множество контуров, покрывающих все 1 (0) функции образуют тупиковую ДНФ (КНФ). Целью минимизации является нахождение минимальной из множества тупиковых форм.

7. В элементарной конъюнкции (дизъюнкции), которая соответствует одному контуру, остаются только те переменные, значение которых не изменяется внутри обведенного контура. Переменные булевой функции входят в элементарную конъюнкцию (для значений функции 1) без инверсии, если их значение на соответствующих координатах равно 1 и с инверсией - если 0. Для значений булевой функции, равных 0, записываются элементарные дизъюнкции, куда переменные входят без инверсии, если их значение на соответствующих координатах равно 0 и с инверсией - если 1.

Если рассматривать запись результатов минимизации в кубическом виде, то при минимизации булевой функции по единичным значениям, каждой конъюнкции ранга R соответствует куб ранга R , где каждой переменной без инверсии соответствует 1 в кубе, переменной с инверсией - 0, а на месте отсутствующей переменной ставится X . Полученное множество кубов образует единичное покрытие $C1$ (соответствующее ДНФ).

При минимизации булевой функции по нулевым значениям и представлении результатов минимизации в кубическом виде, нулевое покрытие $C0$ формируется на основе обратной ДНФ, которая является инверсной функцией по отношению к КНФ (способ построения инверсных

функций и пример инверсных функций для $f(1,3,5,6,7) = 1$ рассмотрен в подразделе 1.3). Отметим, что обратная ДНФ строится на основе КНФ. Таким образом каждой дизъюнкции ранга R (из КНФ) соответствует куб ранга R, где каждой переменной без инверсии соответствует 0 в кубе, переменной с инверсией - 1, а на месте отсутствующей переменной ставится X. Полученное множество кубов образует нулевое покрытие C0.

		X3 X4	00	01	11	10	
X1 X2	00	0	0	1	0		S3
	01	0	1	1	1		S6
	11	1	1	1	1		S2
	10	0	1	1	1		S5
							S4

При данном способе задания таблица истинности функции представляется в виде координатной карты состояний, которая содержит 2^n клеток (по числу входных наборов булевой функции n переменных). Переменные функции разбиваются на две группы так, что одна группа определяет координаты столбца карты, а другая - координаты строки.

При таком способе построения каждая клетка определяется значениями переменных, соответствующих определенному двоичному набору. Внутри каждой клетки карты Карно ставится значение функции на данном наборе. Переменные в строках и столбцах располагаются так, чтобы соседние клетки карты Карно различались только в одном разряде переменных, т.е. были соседними. Поэтому значения переменных в столбцах и в строках карты образуют соседний код Грея. Такой способ представления очень удобен для наглядности при минимизации булевых функций.

Карты Карно были изобретены в 1952 Эдвардом В. Вейч'ем и усовершенствованы в 1953 Морисом Карно, физиком из "Бэлл Лабс". Метод карт Карно применим к минимизации булевых функций до 6-ти переменных (до 4 переменных на плоскости) и до 6 - в трехмерной интерпретации.

Если требуется получить карту Карно для какой-либо функции, сначала надо записать эту функцию в СДНФ, - в совершенной дизъюнктивно нормальной форме, или в виде таблицы истинности.

Каждое слагаемое булева выражения в СДНФ, или каждая единица в столбце функции таблицы истинности, задается на карте Карно единицей в соответствующей клетке. Координаты этой клетки содержат те же входные переменные и их инверсии, что и данное слагаемое СДНФ булева выражения (или данная строка таблицы истинности).

Таблица истинности для четырех переменных включает 16 строк, следовательно карта Карно должна состоять из 16 клеток, как показано на рисунке:

	$\bar{A} \bar{B}$	$\bar{A} B$	$A B$	$A \bar{B}$
$\bar{C} \bar{D}$				
$\bar{C} D$				
$C D$				
$C \bar{D}$				

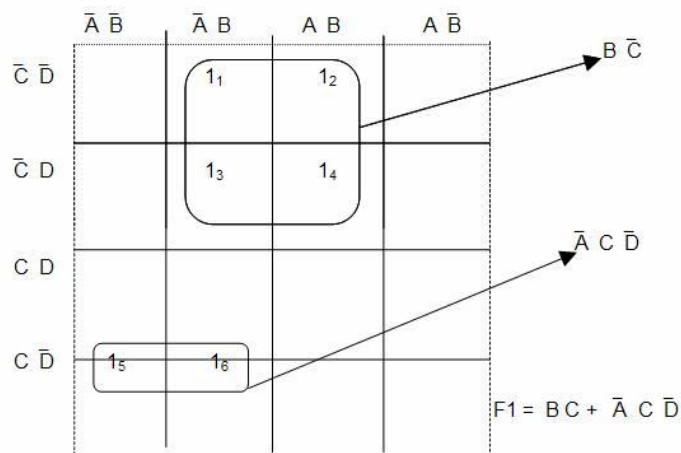
У карты Карно для четырех переменных клетки крайнего левого столбца должны рассматриваться как соседние для клеток крайнего правого столбца, а клетки верхней строки, – как соседние для клеток нижней строки. Другими словами можно сказать, что эта карта расположена на поверхности цилиндра (склеили правый край карты с левым), изогнутого и растянутого так, что его верхний срез соединяется с нижним срезом; при этом цилиндр превращается в тор (бублик).

Правила упрощения заполненной карты Карно для четырех переменных заключаются в следующем :

- соседние две, четыре, или восемь единиц обводят общим контуром;
- контур должен быть прямоугольным без изгибов или наклонов;
- каждый контур превращает все входящие в него единицы в одну, т.е. объединенные таким образом слагаемые СДНФ булева выражения дают одно слагаемое в упрощенном выражении;
- те входные переменные, которые входят в координаты данного контура совместно со своими инверсиями, исключаются из слагаемого, которое дает этот контур в упрощенное выражение.

Примеры упрощения булевых выражений с помощью карты Карно:

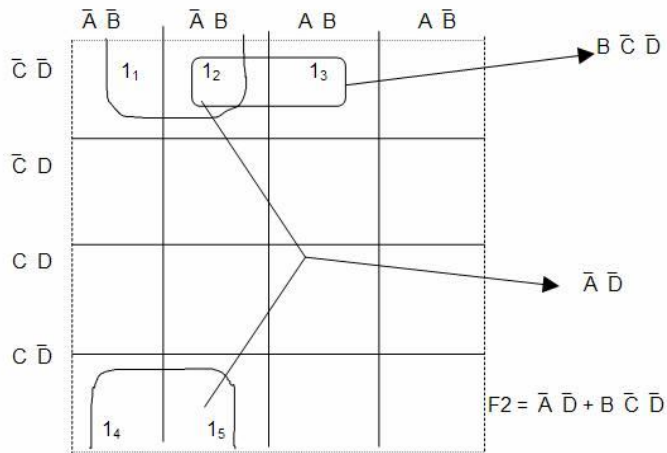
$$1. F1 = \bar{A} \bar{B} \bar{C} \bar{D}_1 + \bar{A} B \bar{C} \bar{D}_2 + \bar{A} B \bar{C} D_3 + A B \bar{C} \bar{D}_4 + \\ + \bar{A} \bar{B} C \bar{D}_5 + \bar{A} B C \bar{D}_6.$$



В первом примере минимизации булевой функции $F1$ нижний контур из двух единиц 15 и 16, соответствующие пятому и sixthому слагаемым в исходном булевом выражении, дает возможность опустить B и $\bar{B}$. После этого в нем остается произведение $\bar{A} C \bar{D}$. В верхнем контуре из четырех единиц 11, 12, 13 и 14, соответствующие первым четырем слагаемым в

исходном булевом выражении попарно опускаются A и $\bar{A}$, D и $\bar{D}$, так что в результате этого верхний контур дает произведение $B C$.

$$2. F2 = \bar{A} \bar{B} \bar{C} \bar{D}_1 + \bar{A} B \bar{C} \bar{D}_2 + A B \bar{C} \bar{D}_3 + \bar{A} B C \bar{D}_4 + \bar{A} B C \bar{D}_5$$



Во втором примере минимизации булевой функции $F2$ контур из двух единиц 12 и 13, соответствующие второму и третьему слагаемым в исходном булевом выражении, дает возможность опустить A и $\bar{A}$. После этого в нем остается произведение $B C D$. В контуре из четырех единиц 11, 12, 14 и 15, соответствующие другим четырем слагаемым из исходного булева выражения, попарно опускаются B и $\bar{B}$, C и $\bar{C}$, так что в результате этого верхний контур дает произведение $\bar{A} D$.

Карта Карно представляется в данном случае свернутой в цилиндр, в котором верхний край совмещается с нижним. Этот пример показывает также, что контура могут накладываться друг на друга.

Правило покрытия 1.

Любой области из 2^k смежных клеток можно поставить в соответствие конъюнкцию $(n-k)$ -го ранга, состоящую из переменных, которые имеют постоянное значение во всех единичных наборах, соответствующих клеткам области. Причем переменная x_i включается в конъюнкции в прямом виде, если эта переменная имеет значение 1 на всех клетках области. Соответственно переменная x_i включается в конъюнкции в инверсном виде, если она имеет значение 0 на всех клетках области. "Покрытая" область на карте обводится контуром. Дизъюнкция конъюнкций, совместно покрывающих все клетки карты, заполненные единицами, есть одна из ДНФ переключательной функции.

Цель минимизации формулы ПФ по карте Карно - "покрыть" все клетки, содержащие единицы, наименьшим числом конъюнкций наименьшего ранга, т. е. "покрыть" наименьшим числом контуров, каждый из которых охватывает как можно большую область смежных клеток, все клетки, содержащие единицы. Дизъюнкция полученных конъюнкций есть одна из тупиковых (возможно, минимальная) ДНФ функции.

Простым импликантам (минималиям) в методе Квайна-Мак-Класки на карте Карно соответствуют области смежных клеток, не являющиеся частью никакой другой области смежных клеток.

Обязательной простой импликанте (экстремали) в методе Квайна-Мак-Класки соответствует область смежных клеток, которая покрывает хотя бы одну единичную клетку, не входящую в состав никакой из других областей смежных клеток.

Правило покрытия 2.

Любой смежной области 2^k клеток, заполненных нулями, можно поставить в соответствие дизъюнкцию $(n-k)$ -го ранга, состоящую из переменных, которые имеют постоянное значение во всех нулевых наборах, соответствующих клеткам области, причем переменная x_i входит в дизъюнкцию в прямом виде, если имеет значение 0 на всех клетках области, и, соответственно в инверсном виде, если она имеет значение 1 на всех клетках области. Конъюнкция минимального числа дизъюнкций, совместно покрывающих все клетки карты, заполненные нулями, есть одна из тупиковых (возможно, минимальных) КНФ переключательной функции.

2. Метод Петрика

Метод Петрика используется для нахождения всех минимальных покрытий конститuent единицы и позволяет получить все тупиковые ДНФ по импликантной матрице. Суть метода заключается в следующем. По импликантной матрице строится так называемое конъюнктивное представление импликантной матрицы. Для этого все простые импликанты обозначаются разными буквами (обычно прописными латинскими). После этого, для каждого i -го столбца импликантной матрицы строится дизъюнкция всех букв, обозначающих строки матрицы, пересечение которых с i -м столбцом отмечено крестиком. Конъюнктивное представление импликантной матрицы образуется как конъюнкция построенных дизъюнкций для всех столбцов матрицы. К конъюнктивному представлению матрицы могут быть применены все соотношения булевой алгебры с целью его упрощения. После раскрытия скобок и выполнения всех возможных поглощений получается дизъюнкция конъюнкций, каждая из которых содержит все импликанты тупиковой ДНФ.

Рассмотрим табл. 3, строки которой соответствуют простым импликантам функции f , а столбцы — конъюнкциям совершенной ДНФ (СДНФ). В каждую клетку записываем единицу, если соответствующая простая импликанта поглощает элементарную конъюнкцию и нуль — в противном случае. Такая таблица называется «импликантной таблицей».

Согласно определению, каждая тупиковая ДНФ определяется таким набором строк, что в таблице, образованной этими строками в каждом столбце имеется одна единица, причём из этого набора нельзя удалить ни одной строки так, чтобы при этом ни один столбец не стал нулевым.

Таблица 3

СДНФ Сокр. ДНФ		10 00	1100	1010	0101	0011	1110	1011	0111
P_1	1 <u>0</u>	1	1	1	0	0	1	0	0
P_2	101 <u> </u>	0	0	1	0	0	0	1	0
P_3	<u> </u> 001	0	0	0	0	1	0	1	0
P_4	0 <u>11</u>	0	0	0	0	1	0	0	1
P_5	01 <u> </u> 1	0	0	0	1	0	0	0	1

Пусть в общем случае в таблице имеется N столбцов и m строк. Поставим в соответствие простым импликантам сокращённой ДНФ переменные $P_1 \dots P_m$. Фиксируем некоторую дизъюнкцию простых импликант. Будем считать, что $P_i = 1$, если i -я простая импликанта входит в эту дизъюнкцию и $P_i = 0$, в противном случае. Запишем в виде формалы условие того, что рассматриваемая дизъюнкция является ДНФ функции. Для этого необходимо, чтобы в каждом столбце таблицы была хотя бы одна единица, т.е.

$$\prod_{j=1}^N (a_{1j}P_1 \vee \dots \vee a_{mj}P_m) = 1,$$

где a_{ij} — элемент матрицы (таблицы), стоящий в i -й строке и j -м столбце, $i = \overline{1, m}$, $j = \overline{1, N}$.

Эту формулу можно трактовать как КНФ некоторой двоичной функции от переменных $P_1 \dots P_m$, которая принимает значение 1 только на тех наборах переменных, которые соответствуют некоторым ДНФ исходной функции, и значение 0 — на наборах, которые соответствуют наборам импликант, не являющихся ДНФ исходной функции.

Заметим, что функция $\varphi(P_1, \dots, P_m)$ монотонна, так как формула 3 не содержит переменных с отрицаниями. Поэтому согласно утверждению 3 для нахождения её сокращённой ДНФ достаточно раскрыть скобки в формуле 3, а затем произвести все поглощения. Наконец, остаётся заметить, что в силу указанного выше свойства этой функции, её простые импликанты и только они будут давать тупиковые ДНФ исходной функции f .

Для табл. 3 функция φ равна:

$$\begin{aligned} \varphi(P_1, P_2, P_3, P_4, P_5) &= \\ &= P_1 \cdot P_1 \cdot (P_1 \vee P_2) \cdot P_5 \cdot (P_3 \vee P_4) \cdot P_1 \cdot (P_2 \vee P_3) \cdot (P_4 \vee P_5) = \\ &= P_1 \cdot (P_1 \vee P_2) \cdot P_5 \cdot (P_3 \vee P_4) \cdot (P_2 \vee P_3) \cdot (P_4 \vee P_5) = \\ &= P_1 \cdot P_5 (P_3 \vee P_2 P_4) = P_1 P_3 P_5 \vee P_1 P_2 P_4 P_5. \end{aligned}$$

Отсюда $P_1 P_3 P_5$ даёт для f тупиковую форму:

$$x_1 \bar{x}_4 \vee \bar{x}_2 x_3 x_4 \vee \bar{x}_1 x_2 x_4,$$

а $P_1 P_2 P_4 P_5$ даёт:

$$x_1 \bar{x}_4 \vee x_1 \bar{x}_2 x_3 \vee \bar{x}_1 x_3 x_4 \vee \bar{x}_1 x_2 x_4.$$

Метод Петрика используется для нахождения всех минимальных покрытий конституент единицы и позволяет получить все тупиковые ДНФ по импликантной матрице. Суть метода заключается в следующем. По импликантной матрице строится так называемое конъюнктивное представление импликантной матрицы. Для этого все простые импликанты обозначаются разными буквами (обычно прописными латинскими). После этого, для каждого i -го столбца импликантной матрицы строится дизъюнкция всех букв, обозначающих строки матрицы, пересечение которых с i -м столбцом отмечено крестиком. Конъюнктивное представление импликантной матрицы образуется как конъюнкция построенных дизъюнкций для всех столбцов матрицы. К конъюнктивному представлению матрицы могут быть применены все соотношения булевой алгебры с целью его упрощения. После раскрытия скобок и выполнения всех возможных поглощений получается дизъюнкция конъюнкций, каждая из которых содержит все импликанты тупиковой ДНФ.

Пример. Задана импликантная матрица (табл. 4.) Найти методом Петрика все тупиковые ДНФ булевой функции f , описываемой данной матрицей.

Таблица 4						
	Конституенты единицы					
Простые импликанты	$/x_1/x_2/x_3x_4$	$/x_1/x_2x_3x_4$	$/x_1x_2/x_3x_4$	$/x_1x_2x_3x_4$	$x_1x_2x_3/x_4$	$x_1x_2x_3x_4$
$/x_1x_4$	X	X	X	X		
$x_2x_3x_4$				X		X
$x_1x_2x_3$					X	X

Имеющиеся простые импликанты обозначим буквами:

$$/x_1x_4 = A, \quad x_2x_3x_4 = B, \quad x_1x_2x_3 = C.$$

Тогда конъюнктивное представление w матрицы имеет вид

$$w = A * A * A * (A \vee B) * C (B \vee C).$$

Упростим его.

$$w = A * (A \vee B) * C (B \vee C) = AC.$$

Тупиковая ДНФ содержит две простые импликанты: $A = /x_1x_4$ и $C = x_1x_2x_3$ и имеем вид $f = /x_1x_4 \vee x_1x_2x_3$.

Задание:

Найти минимальные ДНФ и КНФ булевых функций:

$F(x,y,z,t) = 1$ на наборах 0,3,4,6,8,9,10,11,13,15

$F(x,y,z,t) = 1$ на наборах 1,2,4,6,7,9,10,11,13,15

$F(x,y,z,t) = 1$ на наборах 1,3,5,7,8,10,12,14,15

$F(x,y,z,t) = 1$ на наборах 5,7,8,9,10,11,12,14,15

$F(x,y,z,t) = 1$ на наборах 1,8,10,11,13,14,15

$F(x,y,z,t) = 1$ на наборах 0,1,2,3,5,6,9,10,12,14,15

Практическое занятие №7

Переход от таблицы истинности к СДНФ и СКНФ

Цель работы: научиться осуществлять переход от таблицы истинности к СДНФ и СКНФ.

Оборудование: инструкционные карты.

Теоретические сведения:

Известны два способа задания логических функций: с помощью формулы и с помощью таблицы истинности. По формуле легко составляется таблица. На практике при конструировании различных электронных устройств часто возникает обратная задача — от таблицы истинности перейти к формуле, чтобы на ее основе построить функциональную схему. Переменные структурной формулы соответствуют входам функциональной схемы. Значения переменных в таблице истинности соответствуют значениям входов функциональной схемы. Введем следующие определения. Элементарной конъюнкцией называется конъюнкция нескольких переменных, взятых с отрицанием или без отрицания, причем среди переменных могут быть одинаковые. Элементарной дизъюнкцией называется дизъюнкция нескольких переменных, взятых с отрицанием или без отрицания, причем среди переменных могут быть одинаковые. Всякую дизъюнкцию элементарных конъюнкций назовем дизъюнктивной нормальной формой (ДНФ). Всякую конъюнкцию элементарных дизъюнкций назовем конъюнктивной нормальной формой (КНФ). Совершенной дизъюнктивной нормальной формой (СДНФ) называется ДНФ, в которой нет одинаковых элементарных конъюнкций и все конъюнкции состоят из одного и того же набора переменных, в который каждая переменная входит только один раз (возможно, с отрицанием). Совершенной конъюнктивной нормальной формой (СКНФ) называется КНФ, в которой нет одинаковых элементарных дизъюнкций и все дизъюнкции состоят из одного и того же набора переменных, в который каждая переменная входит только один раз (возможно, с отрицанием).

Алгоритм получения СДНФ по таблице истинности.

1. Отметить те строки таблицы истинности, в последнем столбце которых стоят 1:

X	Y	F(X,Y)
0	0	0
0	1	1*
1	0	1*
1	1	0

2. Выписать для каждой отмеченной строки конъюнкцию всех переменных следующим образом: если значение некоторой переменной в данной строке равно 1, то в конъюнкцию включать саму эту переменную, если равно 0, то ее отрицание: $\bar{X} \wedge Y$ — для 2-й строки; $X \wedge \bar{Y}$ — для 3-й строки.

3. Все полученные конъюнкции связать в дизъюнкцию: $(\bar{X} \wedge Y) \vee (X \wedge \bar{Y})$
(1*)

Алгоритм получения СКНФ по таблице истинности.

1. Отметить те строки таблицы истинности, в последнем столбце которых стоит 0:

X	Y	F(X,Y)
0	0	0*
0	1	1
1	0	1
1	1	0*

2. Выписать для каждой отмеченной строки дизъюнкцию всех переменных следующим образом: если значение некоторой переменной в данной строке равно 0, то в дизъюнкцию включать саму эту переменную, если равно 1, то ее отрицание: $X \vee Y$ — для 1-й строки; $\bar{X} \vee \bar{Y}$ — для 4-й строки.

3. Все полученные дизъюнкции связать в конъюнкцию: $(X \vee Y) \wedge (\bar{X} \vee \bar{Y})$
(2*)

Если мы хотим построить формулу некоторой функции по таблице истинности этой функции, то всегда можно получить СКНФ или СДНФ этой функции.

Задание:

1. Является ли высказывание $(X \rightarrow Y) \leftrightarrow (Y \rightarrow X)$ тавтологией. Выписать СКНФ и СДНФ.
2. Установить эквивалентны ли высказывания. Выписать СКНФ или СДНФ.

а) $X1 = \overline{A \wedge \bar{B} \vee C}$ $X2 = \overline{A \wedge \bar{B} \vee \bar{C}}$ $X3 = (\bar{A} \vee B) \wedge \bar{C}$

б) $X1 = \overline{X \wedge Y}$ $X2 = \overline{X \vee Y}$ $X3 = \bar{X} \vee \bar{Y}$

3. Является ли высказывание $(X \rightarrow Y) \leftrightarrow (Y \rightarrow X)$ тавтологией. Выписать **СКНФ** и **СДНФ**.
4. Установить эквивалентны ли высказывания. Выписать **СКНФ** или **СДНФ**.

а) $X1 = \overline{A \wedge \bar{B} \vee C}$ $X2 = \overline{A \wedge \bar{B} \vee \bar{C}}$ $X3 = (\bar{A} \vee B) \wedge \bar{C}$

б) $X1 = \overline{X \wedge Y}$ $X2 = \overline{X \vee Y}$ $X3 = \bar{X} \vee \bar{Y}$

Контрольные вопросы:

1. Поясните назначение СКНФ, СДНФ.
2. Объясните алгоритм перехода от таблицы истинности к КНФ.

Практическое занятие №8

Построение схем цифровых логических устройств методом синтеза

Цель работы: получить практические навыки синтеза логических схем

Оборудование: инструкционные карты.

Теоретические сведения:

По заданной функции f требуется построить схему, реализующую данную функцию. Задача синтеза решается неоднозначно. Можно поставить в соответствие заданной функции f целое множество схем. Для построения логической схемы необходимо элементы, предназначенные для выполнения логических операций, указанных в логической функции, располагать в порядке, указанном в булевом выражении.

Пример. Построить логическую схему устройства, реализующего логическую функцию $f = \overline{x_1}x_2x_3 \vee x_1\overline{x_2}x_3 \vee x_1x_2\overline{x_3} \vee x_1x_2x_3$. (рис. 1)

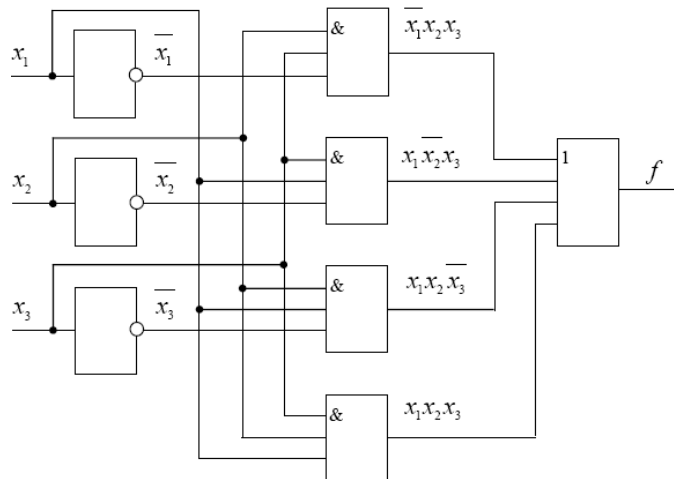


Рис. 1. Пример логической схемы устройства

Синтез логических устройств в заданном базисе.

С целью уменьшения номенклатуры используемых микросхем, часто пользуются функционально полной системой в составе двух логических элементов выполняющих операции И-НЕ, ИЛИ-НЕ. Любую логическую функцию можно записать в заданном базисе логических элементов. Если задан базис И-НЕ, то путем двойного инвертирования исходного выражения или его части и применения теорем де Моргана логическая функция приводится к виду, содержащему только операции логического умножения и инвертирования. Если же задан базис ИЛИ-НЕ, исходную логическую функцию теми же приемами приводят к виду, содержащему только операции логического сложения и инверсии. Далее логическое выражение записывается через условные обозначения выбранных операций.

Пример. Заданную функцию f перевести в базисы И-НЕ и ИЛИ-НЕ. Исходная ДНФ в базисе И-НЕ имеет вид:

$$f = x_2x_4 + \overline{x_1x_3x_4} + \overline{x_1x_2x_3} = x_2x_4 + \overline{x_1x_3x_4} + \overline{x_1x_2x_3} =$$

$$= (\overline{x_2x_4})(\overline{\overline{x_1x_3x_4}})(\overline{\overline{x_1x_2x_3}}) = (\overline{x_2} | \overline{x_4})(\overline{x_1} | \overline{x_3} | \overline{x_4})(\overline{x_1} | \overline{x_2} | \overline{x_3}).$$

Аналогично, КНФ в базисе ИЛИ-НЕ имеет вид:

$$f = \overline{(x_1 + x_4)(x_1 + x_2 + x_3)(x_2 + x_3 + x_4)} =$$

$$= \overline{(x_1 + x_4)} + \overline{(x_1 + x_2 + x_3)} + \overline{(x_2 + x_3 + x_4)} =$$

$$= (x_1 \downarrow x_4) \downarrow (\overline{x_1} \downarrow \overline{x_2} \downarrow \overline{x_3}) \downarrow (\overline{x_2} \downarrow \overline{x_3} \downarrow \overline{x_4}).$$

Задание 1.

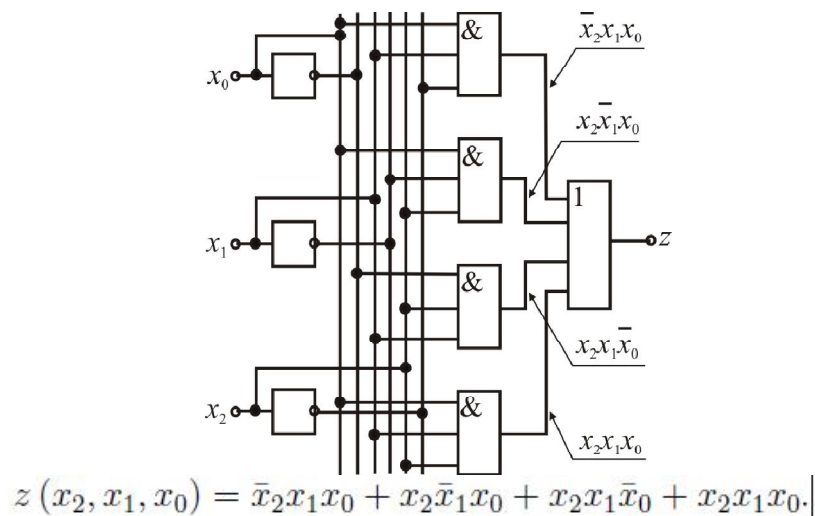
4. Построить структурную схему логического устройства по логической функции трех переменных равной «1», если хотя бы две входные переменные равны «1».

5. Построить структурную схему логического устройства по логической функции трех переменных равной «0», если хотя бы две входные переменные равны «0».

6. Построить структурную схему логического устройства по логической функции четырех переменных равной «1», если хотя бы три входные переменные равны «1».

Пример. Построить структурную схему (1).

Решение: Построить ТИ. Для реализации ФАЛ нам понадобятся три ЛЭ, реализующих операцию НЕ, т.к. исходная ФАЛ формируется тремя переменными, операция дизъюнкции должна быть выполнена четыре раза над тремя переменными, т.о. понадобятся четыре ЛЭ реализующих операцию ЗИ. Последней выполняется операция конъюнкции над четырьмя выражениями, требуется ЛЭ реализующий операцию 4 ИЛИ.



Задание 2.

Пример. Пусть логическая функция задана выражением

$$f = x_1 x_4 + \overline{(x_1 + x_3 + x_4)}(x_2 + x_3)$$

Привести логическую функцию в базис И-НЕ, ИЛИ-НЕ.

а) приводим функцию к базису И-НЕ

$$f = f_1 + \overline{f_2 f_3} = f_1 + \overline{\overline{f_2 f_3}} = f_1 + \overline{(f_2 | f_3)} = f_1 + \overline{\overline{\overline{(f_2 | f_3)}}} = \overline{\overline{f_1}} + \overline{\overline{\overline{(f_2 | f_3)}}} = \overline{f_1} | (\overline{f_2} | \overline{f_3});$$

$$\overline{f_1} = \overline{x_1 x_4} = x_1 | x_4;$$

$$\overline{f_2} = \overline{x_1 + x_3 + x_4} = \overline{x_1 x_3 x_4} = \overline{x_1 x_3} | \overline{x_4} = x_1 | x_3 | x_4;$$

$$\overline{f_3} = \overline{x_2 + x_3} = \overline{x_2 x_3} = x_2 | x_3;$$

$$f = (x_1 | x_4) | ((x_1 | x_3 | x_4) | (x_2 | x_3)).$$

б) приводим функцию к базису ИЛИ-НЕ

$$f = f_1 + \overline{f_2 f_3} = f_1 + \overline{\overline{f_2 f_3}} = f_1 + \overline{(f_2 + f_3)} = f_1 + \overline{\overline{\overline{(f_2 + f_3)}}} = f_1 \downarrow (\overline{f_2} \downarrow \overline{f_3});$$

$$\overline{f_1} = \overline{x_1 x_4} = \overline{x_1} \downarrow \overline{x_4} = x_1 \downarrow x_4;$$

$$\overline{f_2} = \overline{x_1 + x_3 + x_4} = \overline{x_1} \downarrow \overline{x_3} \downarrow \overline{x_4} = x_1 \downarrow x_3 \downarrow x_4;$$

$$\overline{f_3} = \overline{x_2 + x_3} = \overline{x_2} \downarrow \overline{x_3} \Rightarrow \overline{f_3} = x_2 \downarrow x_3;$$

$$f = (x_1 \downarrow x_4) \downarrow ((x_1 \downarrow x_3 \downarrow x_4) \downarrow (x_2 \downarrow x_3)).$$

Индивидуальные задания

Синтезировать в базисе И-НЕ и ИЛИ-НЕ устройства, заданные логической функцией:

$$1) f(x_1, x_2, x_3, x_4) = x_1 x_2 \bar{x}_3 \vee \bar{x}_1 x_3 x_4 \vee \overline{(x_1 x_2)}(\overline{x_3 x_4});$$

$$2) f(x_1, x_2, x_3, x_4) = \overline{(x_1 \vee x_2)} x_3 x_4;$$

$$3) f(x_1, x_2, x_3, x_4) = \overline{(x_1 x_2)} \vee x_3 \vee x_4;$$

$$4) f(x_1, x_2, x_3, x_4) = x_1 \vee x_2 \vee \bar{x}_3 \vee \bar{x}_4;$$

$$5) f(x_1, x_2, x_3, x_4) = x_1 x_2 \bar{x}_3 \bar{x}_4;$$

$$6) f(x_1, x_2, x_3, x_4) = x_1 \vee x_2 \vee \bar{x}_3 \vee x_4 \vee \overline{x_1 x_2};$$

$$7) f(x_1, x_2, x_3, x_4) = (x_1 \vee \bar{x}_2)(x_3 \vee \bar{x}_4);$$

$$8) f(x_1, x_2, x_3, x_4) = x_1 x_2 x_3 \vee \bar{x}_1 x_2 \bar{x}_4 \vee \overline{(\bar{x}_1 \vee x_2)}(\overline{x_3 \vee x_4});$$

$$9) f(x_1, x_2, x_3, x_4) = x_1 x_2 x_3 \vee \overline{x_1 x_2 x_3} \vee x_4;$$

$$10) f(x_1, x_2, x_3, x_4) = (x_1 \vee x_2) \bar{x}_3 \vee \overline{x_1 x_2 x_3 x_4};$$

Контрольные вопросы:

1. Поясните задачу синтеза КЦ.
2. Понятие базиса.

Практическое занятие №9

Построение триггеров на основе логических элементов интегральной схемотехники методом синтеза

Цель работы: научиться строить триггеры на основе логических элементов интегральной схемотехники методом синтеза.

Оборудование: инструкционные карты.

Теоретические сведения:

1.1 Цифровые устройства последовательностного типа

Все цифровые устройства (ЦУ) принято разбивать на два класса: комбинационные ЦУ (КЦУ) и последовательностные ЦУ (ПЦУ).

Для ПЦУ значения выходных переменных определяются не только входными переменными в данный момент, но и их значениями в предшествующие моменты времени. Т.е. ПЦУ должно обладать памятью. Поэтому за ПЦУ закрепились также следующие наименования: ЦУ с памятью, многотактные ЦУ, конечные автоматы. Свойство запоминания информации обеспечивается наличием у ПЦУ различных устойчивых внутренних состояний $Q_1, Q_2, \dots, Q_r$, каждое из которых характеризуется определенной комбинацией сигналов во внутренних цепях ПЦУ. По аналогии со входными и выходными переменными внутренние переменные (состояния) кодируются двоичными L -разрядными числами.

1.2 Триггеры

Триггеры являются простейшими ПЦУ. Отличительными особенностями триггеров являются:

1) число внутренних устойчивых состояний равно двум, чему соответствует одна переменная в прямой (Q) или инверсной форме ($\bar{Q}$);

2) число выходов у триггера также равно двум, один из них называют прямым, другой – инверсным. Причем значения выходов равны соответствующим значениям внутренней переменной. Поэтому для триггеров принято прямой выход обозначать – Q , а инверсный – $\bar{Q}$. Состояние триггера определяется по уровню напряжения на его прямом выходе. Если это напряжение уровня логической единицы, т.е. $Q=1$ (при этом $\bar{Q}=0$), то говорят, что триггер находится в единичном состоянии (в триггер записана “1”). Если же $Q=0$ ($\bar{Q}=1$) – триггер находится в нулевом состоянии (записан “0”).

1.3. Пример выполнения задания:

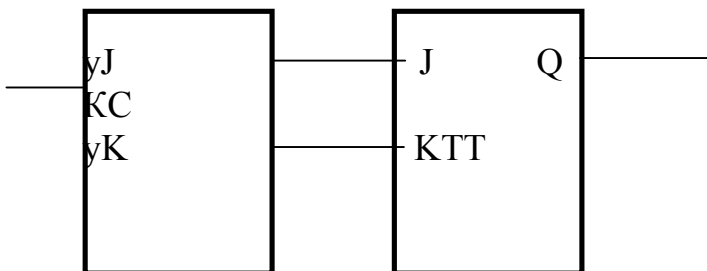
Преобразовать JK триггер в D-триггер.

1. Необходимо составить таблицы истинности двух триггеров: исходного и необходимого:

J	K	Q^t	Q^{t+1}	
0	0	0	0	Режим хранения
0	0	1	1	
0	1	0	0	Режим сброса
0	1	1	0	
1	0	0	1	Режим установки
1	0	1	1	
1	1	0	1	Режим инверсии
1	1	1	0	

D	Q^t	Q^{t+1}
0	0	0
0	1	0
1	0	1
1	1	1

2. строим предполагаемую схему:

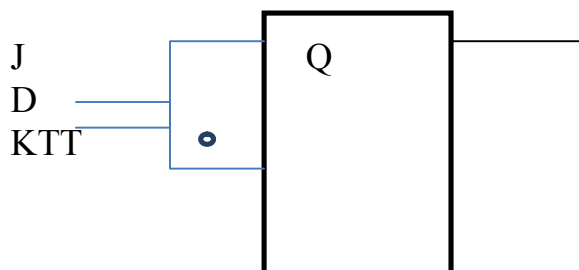


3. Составляем таблицу истинности необходимого триггера, опираясь на предполагаемую схему (при необходимости составляем карту Карно):

D	Q^t	yJ	yK
0	0	0	1
0	1	0	1
1	0	1	0
1	1	1	0

$yJ = D$
 $yK = \neg D$

4. Строим новый триггер



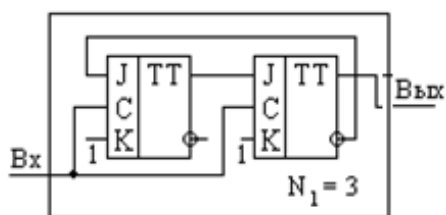
Задание:

- 1.Получить из JK триггера Т- триггер.
- 2.Получить из RS триггера Т- триггер.

Контрольные вопросы

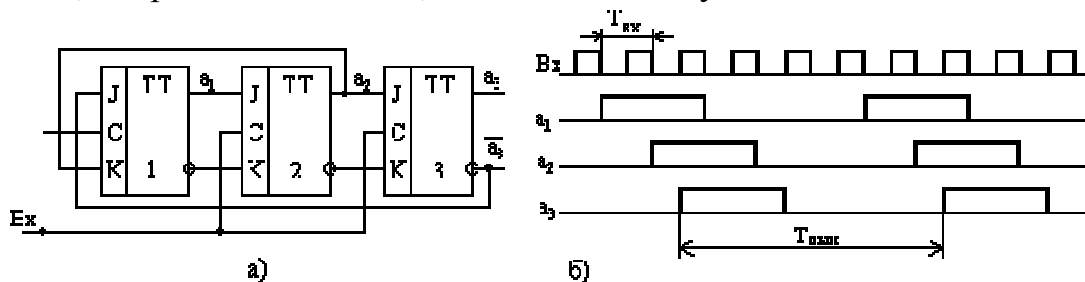
1. Какой триггер называется D-триггером? Т-триггером?
2. Что такое запрещенная комбинация сигналов на входах триггера?
3. Почему JK-триггер называется универсальным?
4. Что такое прямой и инверсный входы?
5. Как построить Т-триггер на основе RS-, D- и JK-триггеров?

Последовательное соединение делителей частоты с коэффициентом деления, равным двум, при котором выход каждого из делителей подключен к входу следующего. На выходе каждого делителя частота следования импульсов вдвое ниже, чем на входе. Так, если частота следования импульсов на входе первого делителя $f_{\text{вх}}$, то на выходе первого делителя она равна $f_{\text{вых1}} = f_{\text{вх}}/2$, на выходе второго $f_{\text{вых2}} = f_{\text{вых1}}/2 = f_{\text{вх}}/2^2$, на выходе третьего делителя $f_{\text{вых3}} = f_{\text{вых2}}/2 = f_{\text{вх}}/2^3$ и т. д. При n каскадах подобного деления частота выходной последовательности окажется равной $f_{\text{вых}} = f_{\text{вх}}/2^n$, т. е. будет осуществляться деление частоты в $N = 2^n$ раз.

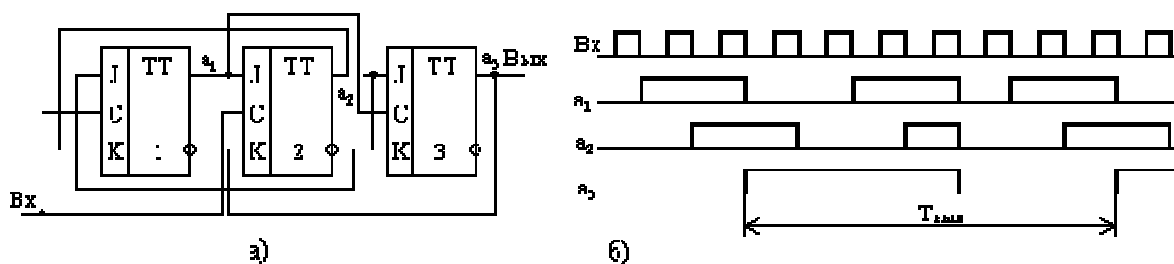


Делитель частоты с коэффициентом деления $N = 3$. Счетчик с периодом цикла $N = 3$ имеет простейшие межтриггерные связи без использования логических элементов. Этот счетчик может одновременно служить и делителем частоты с коэффициентом $N = 2$.

Делитель частоты с коэффициентом деления $N = 5$. Как видно из временной диаграммы, на выходах всех триггеров образуется последовательность импульсов с частотой, в 5 раз более низкой, чем частота импульсов на входе делителя.



Делитель частоты с коэффициентом деления $N = 7$.



Каскадные делители частоты. В тех случаях, когда коэффициент деления N не является простым числом и может быть представлен произведением вида $N = N_1 \cdot N_2 \cdot N_3 \cdot \dots \cdot N_k$, схема делителя строится в виде каскадного соединения делителей, имеющих коэффициенты деления $N_1, N_2, N_3, \dots, N_k$. Таким образом, имея набор схем, реализующих коэффициенты деления, которые представляют собой простые числа, можно каскадным их соединением получать делители с разнообразными коэффициентами деления.

Задание:

Построить временную диаграмму и схему делителя с заданным коэффициентом деления:

- а) $N=6$,
- б) $N=8$,
- в) $N=9$,
- г) $N=10$.

Контрольные вопросы:

1. Каково назначение делителей?
2. Каков принцип действия делителей?
3. Каким образом, можно получить делители с разнообразными коэффициентами деления?

Практическое занятие №11

Построение схемы преобразователя кодов

Цель работы: ознакомиться с методами преобразования двоичных кодов, используемых в системах кодирования и передачи сообщений.

Оборудование: инструкционные карты.

Теоретические сведения:

Преобразователь кодов – это логическое устройство, осуществляющее преобразование входных переменных в совокупность выходных кодов в зависимости от алгоритма преобразования. Преобразователи могут быть реализованы на диодах, на интегральных микросхемах малой и средней степени интеграции, оперативных запоминающих устройствах (ОЗУ), постоянных запоминающих устройствах (ПЗУ) и на других элементах. При помощи преобразователей кодов и логических элементов производятся логические операции над булевыми переменными.

Одним из наиболее распространённых преобразователей кодов является дешифратор.

Дешифраторами называют комбинационные логические структуры, преобразующие входные кодовые комбинации в выходные сигналы, распределяемые по индивидуальным цепям. Применяя некоторую совокупность многовыходовых схем «И», можно построить дешифратор, представляющий собой логическую схему с n входных и M выходных шин.

Каждая кодовая комбинация, подаваемая на вход дешифратора, возбуждает только один из выходов. Кодовые комбинации, как правило, представлены двоичным или двоично-десятичным кодом. Более сложные коды предварительно преобразуют в двоичные коды.

При дешифрации n -разрядного двоичного кода и реализации всех комбинаций этого кода число выходов дешифратора определяется мощностью данного кода: $M = 2^n$.

Функционирование двоичного дешифратора можно описать с помощью логических выражений:

$$f_0 = \overline{X}_n \cdot \overline{X}_{n-1} \cdot \dots \cdot \overline{X}_2 \cdot \overline{X}_1$$

$$f_1 = \overline{X}_n \cdot \overline{X}_{n-1} \cdot \dots \cdot \overline{X}_2 \cdot X_1$$

.....

$$f_{M-1} = X_n \cdot X_{n-1} \cdot \dots \cdot X_2 \cdot X_1$$

(I)

где $X_1, X_2, \dots, X_n$ – сигналы на входах дешифратора;

$f_0, f_1, \dots, f_{M-1}$ – сигналы на выходах дешифратора.

Выражения (I) являются исходными для синтеза дешифраторных и специальных логических схем.

Таким образом, дешифратор представляет собой не что иное, как совокупность схем совпадений, формирующих управляющий сигнал только на одном из выходов, в то время как на остальных выходах этот сигнал отсутствует.

В зависимости от требуемых характеристик и назначения в системах кодирования и передачи сообщений используются несколько разновидностей дешифраторов.

ЛИНЕЙНЫЙ ДЕШИФРАТОР

Логическая схема, реализующая независимое решение каждого из выражений (I), представляет собой линейный дешифратор. Простейший дешифратор на два входа $n = 2$ показан на рис. 1.

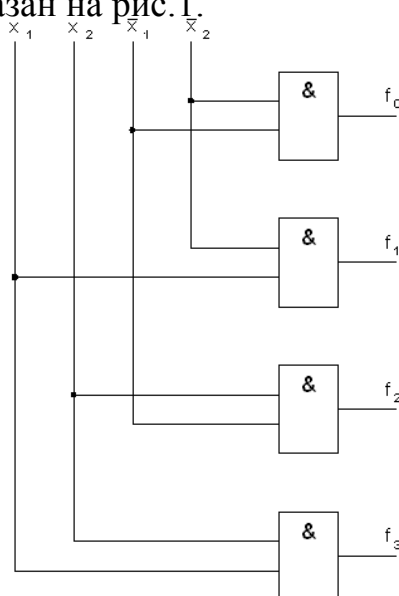


Рис. 1.

Переключательные функции, описывающие работу такого узла, имеют в соответствии с (I) вид ($f_0 = \overline{X_2} \cdot \overline{X_1}$, $f_1 = \overline{X_2} \cdot X_1$, $f_2 = X_2 \cdot \overline{X_1}$, $f_3 = X_2 \cdot X_1$). Функция каждого из выходов прямоугольного дешифратора реализуется на одном вентиле, имеющем определённое количество входов. Учитывая, что каждая микросхема может содержать четыре двухвходовых, три трёхвходовых, два четырёхвходовых или один восьмивходовой вентиль и, зная число входов дешифратора, можно рассчитать затраты, необходимые на его реализацию. Так, например, количество необходимых для реализации линейного дешифратора вентилях равно числу его выходов: $B = 2^n$, а число входов каждого вентиля равно n . При построении дешифратора необходимо

учитывать, что источники как прямого, так и инверсного сигналов этого разряда подключены к половине элементов «И» дешифратора, т.е. требуемый коэффициент разветвления источника по выходу должен быть не менее $M/2$. Если элемент источника не обладает достаточным коэффициентом разветвления по выходу, приходится для его увеличения включать дополнительные элементы. Задержка формирования сигналов на выходе линейного дешифратора составит $\tau_{\text{ср}}$ при парафазных входах и $2\tau_{\text{ср}}$ при однофазных (здесь $\tau_{\text{ср}}$ – среднее время задержки вентиля).

МНОГОСТУПЕНЧАТЫЕ ДЕШИФРАТОРЫ

Многоступенчатые дешифраторы применяются с целью сокращения затрат оборудования при организации дешифраторов с большим числом выходов. Многоступенчатые дешифраторы можно разделить на пирамидальные и собственно многоступенчатые дешифраторы.

Пирамидальные дешифраторы. Общий подход при построении пирамидального дешифратора заключается в разбиении входных переменных на группы. Младшие разряды входного слова дешифрируются на первой ступени при помощи линейного дешифратора (DC). Далее подключаются старшие разряды, каждый из которых в совокупности со значениями младших разрядов образует последующие ступени дешифратора.

Функциональная схема такого дешифратора с $n = 5$ показана на рис.2. Она представляет собой четырёхкаскадную схему. На первый каскад подаются младшие разряды. Второй, третий и четвёртый каскады выполнены на элементах «И», управление которыми производится старшими разрядами соответственно. Можно отметить, что наращивание производится по одному разряду или по одинаковым группам (например, по 2 разряда).

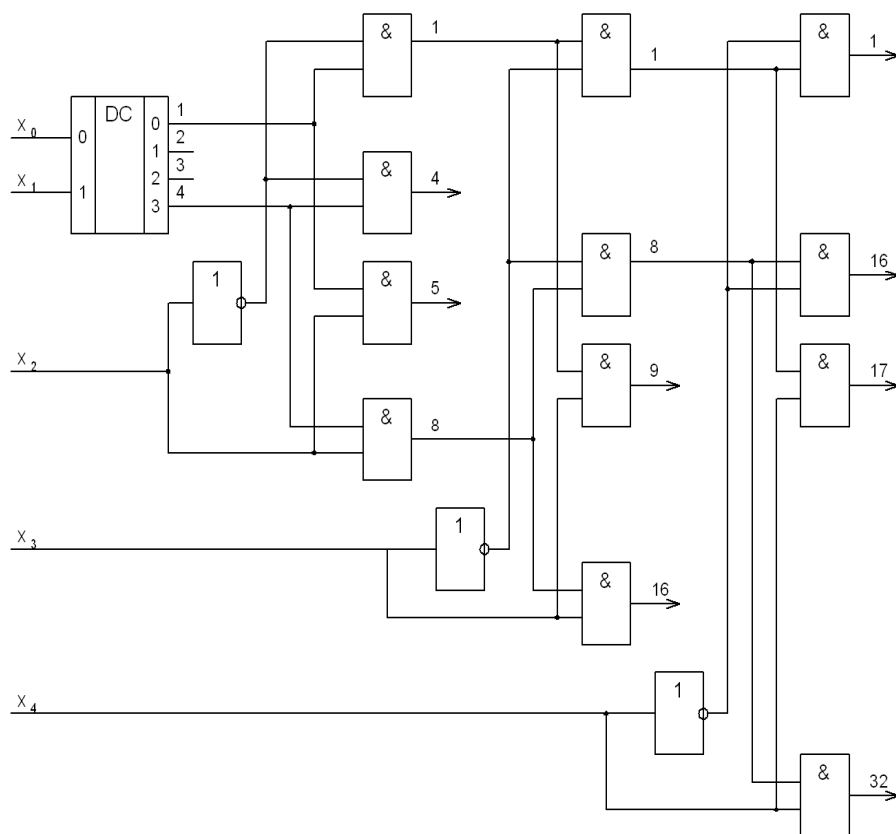


Рис. 2.

Многоступенчатые дешифраторы. При построении собственно многоступенчатого дешифратора необходимо разбить число его входов на две равные группы при чётном n . При нечётном n - на две группы, у которых число входов отличается на единицу: $(n-1)/2$ и $(n+1)/2$. В дальнейшем группы, имеющие более трёх входов, разбиваются по тому же принципу. Использование данного правила позволяет сократить затраты на оборудование до минимума.

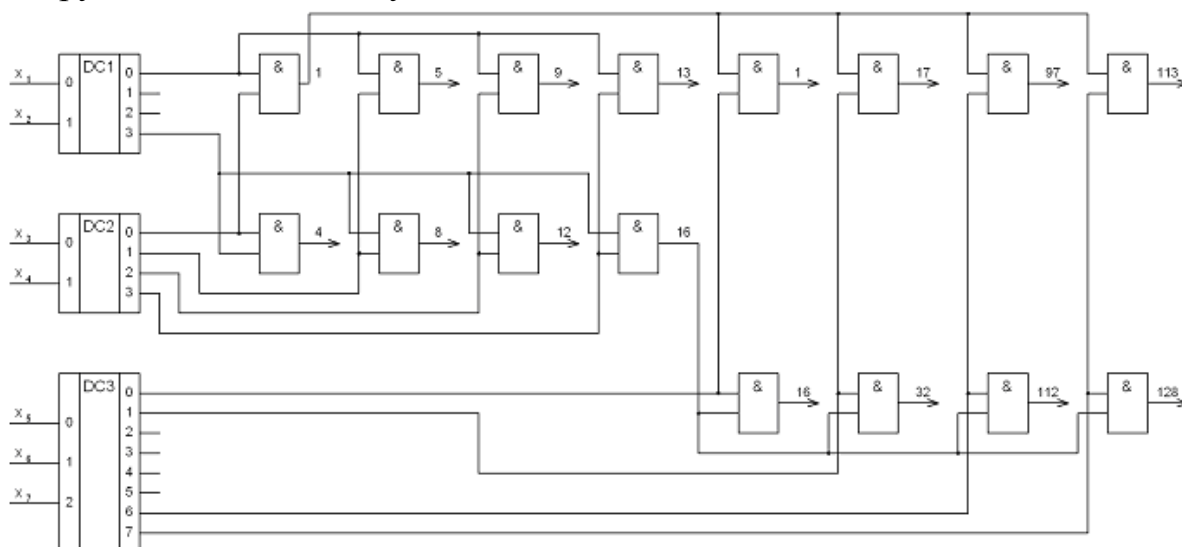


Рис.3

На рис. 3 показана схема построения многоступенчатого дешифратора на 7 входов. Разбиение на группы с использованием приведённого выше правила производится следующим образом. Первая группа имеет четыре входа, вторая – три. Затем первая группа разбивается на следующие две группы по два входа каждая, т.е. окончательный вариант разбиения 2-2-3. Первая ступень включает в себя три линейных дешифратора DC1÷DC3, имеющих согласно разбиению соответственно 2-2-3 входа. Вторая ступень включает в себя преобразователь кода X/Y, выполненный на двухвходовых схемах «И». Совпадение сигналов выявляется с помощью дешифратора, выполненного по матричной схеме, когда каждый из четырёх выходов DC1 подключен к схемам «И», вторые входы которых подключены к каждому из четырёх выходов DC2, что обеспечивает $4 \times 4 = 16$ выходов. Третья ступень включает в себя такой же преобразователь кода, у которого каждые восемь входов схем «И» подключены соответственно к каждому из шестнадцати выходов преобразователя второй ступени (выбор строки). Другие входы схем «И» подключены по шестнадцать выходов к каждому из восьми выходов дешифратора DC3 (выбор столбца).

ДЕШИФРАТОРЫ ДЛЯ КОРРЕКТИРУЮЩИХ КОДОВ

Рассмотренные виды дешифраторов относятся к классу полных дешифраторов, т.к. число выходных шин связано с числом входных соотношением $M = 2^n$.

При использовании неполных дешифраторов, у которых $M < 2^n$, за счёт увеличения избыточности могут быть реализованы дешифраторы для кодов с обнаружением и исправлением ошибок.

Для построения дешифратора с обнаружением ошибок нужны дополнительные узлы, осуществляющие обнаружение ошибки. При обнаружении ошибки эти функциональные узлы должны формировать сигналы, запрещающий дешифрацию сообщения. В случае, когда в ОЗУ записана нулевая кодовая комбинация, что указывает на обнаружение ошибки в принятом сообщении, осуществляется запрет дешифрации при её выявлении.

Для построения дешифратора с исправлением ошибок необходимо выбрать кодовые комбинации с соответствующим кодовым расстоянием $d = S + r + 1$, найти совокупность комбинаций-спутников, отображающих результат наложения всевозможных векторов S -кратной ошибки на разрешённые комбинации. Каждая разрешённая комбинация будет иметь,

таким образом, по $M_c = \sum_{i=1}^S C_n^i$ комбинаций-спутников, которые должны быть объединены на схемах «ИЛИ». Сигналы с выходов схем «ИЛИ» будут являться сигналами управления соответствующих объектов.

ПРЕОБРАЗОВАТЕЛИ ДВОИЧНЫХ КОДОВ

При помощи преобразователей двоичных кодов могут быть реализованы логические узлы, выполняющие различные операции (схема равнозначности, сумматоры, схемы сложения по модулю два и др.), а также преобразователи одного кода в другой.

Широкое применение находят преобразователи двоичных кодов в код управления семисегментным индикатором. Они могут быть реализованы на логических элементах, образующих комбинационную схему, на микросхемах типа К514ИД1 и К514ИД2 и др.

В лабораторной работе преобразователь реализован на ОЗУ, в ячейки памяти которого заносятся кодовые комбинации, осуществляющие включение соответствующих сегментов. Адресация ячеек памяти производится в зависимости от цифры, которую требуется высветить на индикаторе.

Расположение сегментов показано на схеме стенда (рис. 4). Для того, чтобы на индикаторе высветились цифры от 0 до 9, необходимо с выхода дешифратора получить сигналы, которые включали бы сегменты, соответствующие передаваемой цифре. Например, включение первого сегмента происходит при цифрах 0, 2, 3, 5, 6, 8, 9 и описывается при помощи выражения

$$f_1 = \bar{X}_4 \bar{X}_3 \bar{X}_2 \bar{X}_1 \vee \bar{X}_4 \bar{X}_3 X_2 \bar{X}_1 \vee \bar{X}_4 \bar{X}_3 X_2 X_1 \vee \bar{X}_4 X_3 \bar{X}_2 X_1 \vee \bar{X}_4 X_3 X_2 \bar{X}_1 \vee X_4 \bar{X}_3 \bar{X}_2 \bar{X}_1 \vee X_4 \bar{X}_3 \bar{X}_2 X_1$$

При использовании минимизации данное выражение можно привести к виду $f_1 = \bar{X}_1 X_2 \bar{X}_4 \vee \bar{X}_3 X_2 \bar{X}_4 \vee \bar{X}_2 \bar{X}_3 X_4 \vee \bar{X}_1 \bar{X}_2 \bar{X}_3 \vee X_1 \bar{X}_2 X_3 \bar{X}_4$.

В дальнейшем данное выражение реализуется при помощи логических схем «И», для его реализации необходимо использовать четыре трёхвходовых, одну четырёхвходовую и одну пятивходовую схемы. Преобразователи, необходимые для включения остальных сегментов цифры, строятся аналогично рассмотренному.

Задание:

1. Описать схему линейного дешифратора.
2. Описать схему пирамидального дешифратора.

Контрольные вопросы:

1. Для чего предназначены преобразователи кодов.
2. Приведите примеры преобразователей кодов.

Практическое занятие №12

Исследование принципа работы функциональных схем сумматоров

Цель работы: изучить принцип работы функциональных схем сумматоров.

Оборудование: инструкционные карты, лабораторный стенд.

Теоретические сведения:

Суммирование многоразрядных двоичных чисел $A=a_n a_{n-1} \dots a_0$ и $B=b_n b_{n-1} \dots b_0$ производится путем их поразрядного сложения с переносом между разрядами. Поэтому основным узлом многоразрядных сумматоров является комбинационный одноразрядный сумматор, который выполняет арифметическое сложение трех одноразрядных чисел (цифр): цифры данного разряда первого слагаемого (a_i), цифры данного разряда второго слагаемого (b_i) и цифры (1 или 0) переноса из соседнего младшего разряда (p_i). В результате сложения для каждого разряда получаются две цифры – сумма для этого разряда (S_i) и перенос в следующий старший разряд (p_{i+1}).

Условное графическое изображение одноразрядного сумматора и его таблица истинности (функционирования) приведены на рис. 1.

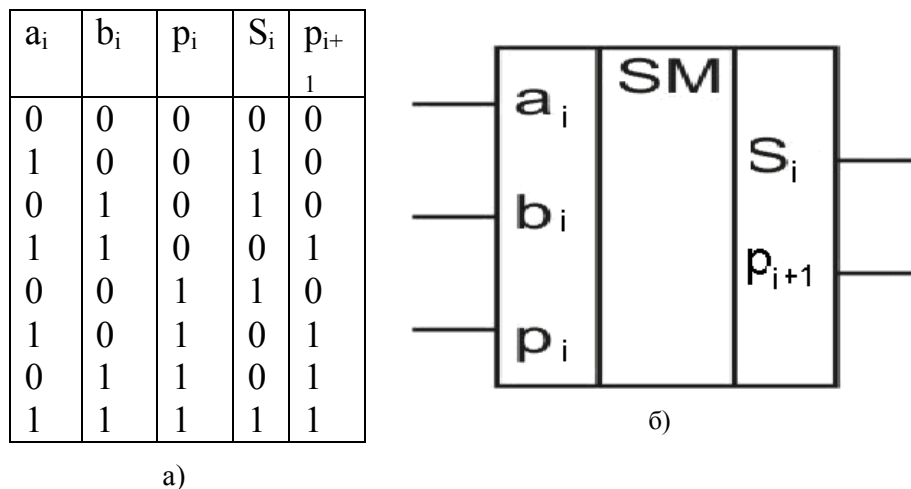


Рис. 1. Условное обозначение (б) и таблица истинности (а) одноразрядного сумматора

Для синтеза схемы одноразрядного сумматора запишем выражения для S_i и p_{i+1} (выходов сумматора):

$$S_i = a_i \bar{b}_i \bar{p}_i + \bar{a}_i b_i \bar{p}_i + \bar{a}_i \bar{b}_i p_i + a_i b_i p_i = (\bar{a}_i \bar{b}_i + a_i b_i) \bar{p}_i + (a_i \bar{b}_i + \bar{a}_i b_i) p_i =$$

$$= (a_i \oplus b_i) \bar{p}_i + (\bar{a}_i \oplus \bar{b}_i) p_i = (a_i \oplus b_i) \oplus p_i$$

Схема одноразрядного сумматора, построенная в соответствии с выражениями (1) приведена на рис. 2.

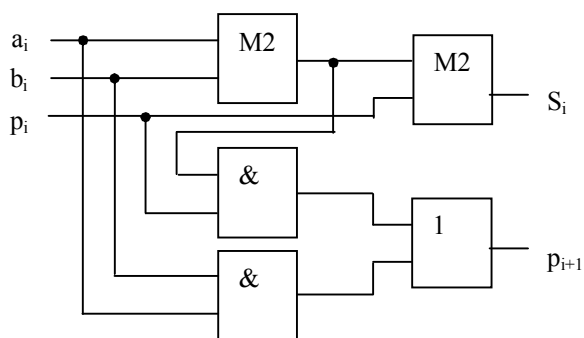


Рис. 2. Схема одноразрядного сумматора

Многоразрядный параллельный сумматор может быть составлен из одноразрядных сумматоров, число которых равно числу разрядов слагаемых, путем соединения выхода, на котором формируется сигнал переноса данного разряда, с входом для сигнала переноса соседнего старшего разряда. Такой способ организации переноса называется

последовательным. Пример построения 3-разрядного параллельного сумматора демонстрирует рис. 3. В сумматорах этого типа перенос распространяется последовательно от разряда к разряду по мере образования суммы в каждом разряде. При наиболее неблагоприятных условиях переноса, например, при сложении чисел 11...11 и 00...01 будет иметь место «пробег» единицы переноса через весь сумматор от самого младшего к самому старшему разряду. Поэтому в наихудшем случае время распространения переноса

$$T_{\text{зд.р.пер.}} = n \cdot t_{\text{зд.р.пер.}}$$

где $t_{\text{зд.р.пер.}}$ – время задержки распространения переноса в одном разряде;

n – число разрядов сумматора. Данный тип сумматора наиболее прост с точки зрения схемы цепей распространения переноса, но имеет сравнительно

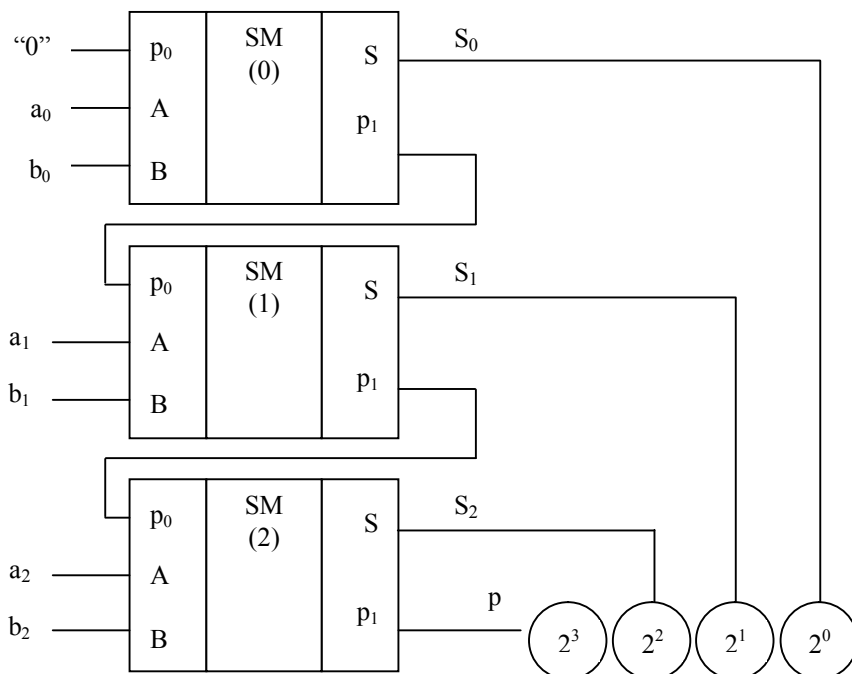


Рис. 3. Функциональная схема 3-разрядного параллельного

низкое быстродействие.

Более высоким быстродействием обладают сумматоры с параллельным переносом, в которых сигналы переноса формируются во всех разрядах одновременно. Этой цели служат специальные схемы ускоренного переноса.

Двоичные сумматоры - вычитатели

Теперь, когда мы знаем, что двоичные сумматоры можно использовать как для сложения, так и для вычитания, спроектируем схему универсального устройства – сумматора - вычитателя, положив в ее основу схему вычитателя (рис. 4, б). Чтобы эта схема работала как 3-разрядный сумматор, достаточно временно (условно) исключить из нее 3 инвертора и на вход переноса младшего разряда подать “0”. В преобразованном виде эта схема (рис. 5) вместо инверторов содержит три логических элемента М2 (сумма по модулю 2). При подаче 0 на вход V логического элемента М2 информационные биты каждого разряда двоичного числа $b_2b_1b_0$ проходят через этот элемент без инверсии. Таким образом, при установке 0 на управляющем входе схема складывает двоичные числа $a_2a_1a_0$ и $b_2b_1b_0$. Результат появляется на выходных индикаторах. Кроме того, логический 0 на управляющем входе V поступает на вход переноса младшего разряда двоичного сумматора.

Чтобы схема работала как 3-разрядный вычитатель, на управляющем входе V нужно установить уровень логической 1. В этом случае логический элемент М2 действует как инвертор сигналов на входах В одноразрядных сумматоров. Кроме того, логическая 1 на управляющем входе поступает на вход переноса младшего разряда двоичного сумматора.

Задание

- 2.1. Используя ЛЭ, расположенные на лабораторном стенде, спроектировать схему и исследовать работу (снять таблицу функционирования) одноразрядного сумматора.
- 2.2. Исследовать работу (снять таблицу функционирования) ИС 2-разрядного сумматора К155ИМ2.
- 2.3. На базе ИС К155ИМ2 спроектировать схему 4-разрядного двоичного сумматора – вычитателя и выполнить следующие арифметические операции $A+B$ и $C-D$ (значения А, В, С, D, соответствующие вашему варианту, приведены в табл.).

№ бригады	1	2	3	4	5
-----------	---	---	---	---	---

A	2	3	3	4	5
B	2	2	3	2	1
C	6	7	5	5	4
D	5	4	1	3	4

Контрольные вопросы

1. Представьте операнды (слагаемые – при сложении; уменьшаемое и вычитаемое – при вычитании) в двоичном обратном коде и выполните следующие операции:

$$\begin{array}{llll}
 \text{а)} (+7) & \text{б)} (+8) & \text{в)} (+3) & \text{г)} (+13) \\
 -(+1) & -(-5) & -(-8) & -(-10)
 \end{array}$$

2. Представьте операнды в двоичном дополнительном коде и выполните те же операции, что и в пункте 1.

3. Дайте определение одноразрядного сумматора и спроектируйте его схему в ОФПН логических элементов. Сравните потребные для этого аппаратные затраты (количество ИС) с затратами, необходимыми для схемы, приведенной на рис. 2.

4. Укажите достоинства и недостатки двоичных сумматоров с последовательным переносом.

5. На базе ИС К155ИМ2 спроектируйте схему 8-разрядного сумматора - вычитателя.

Практическое занятие №13

Проектирование ОЗУ заданной ёмкости и структуры

Цель работы: изучить правила увеличения разрядности ИМС памяти.

Оборудование: инструкционные карты.

Теоретические сведения:

Как правило, *разрядность* данных одной ИМС памяти, оказывается недостаточной для хранения слова большой разрядности. Особенно это относится к *ОЗУ*, имеющим более сложную структуру запоминающего элемента, занимающего большее, нежели у *ПЗУ*, место на кристалле ИМС и хранящего часто лишь одноразрядные слова. Необходимая *разрядность* при этом обеспечивается соединением нескольких однотипных ИМС по следующим правилам:

- на адресные шины всех ИМС параллельно подается один и тот же код адреса;
- управляющие сигналы (выбора кристалла и управление записью и чтением) подаются на все ИМС одновременно;
- каждая ИМС хранит свой разряд слова данных.

На [рис. 13.1](#) приведена схема построения блока *ОЗУ* информационного объема 1Кх8 на базе ИМС информационного объема 1Кх1 каждая. На адресные линии каждой ИМС подключаются младшие линии шины адреса (для примера она принята 16-разрядной).

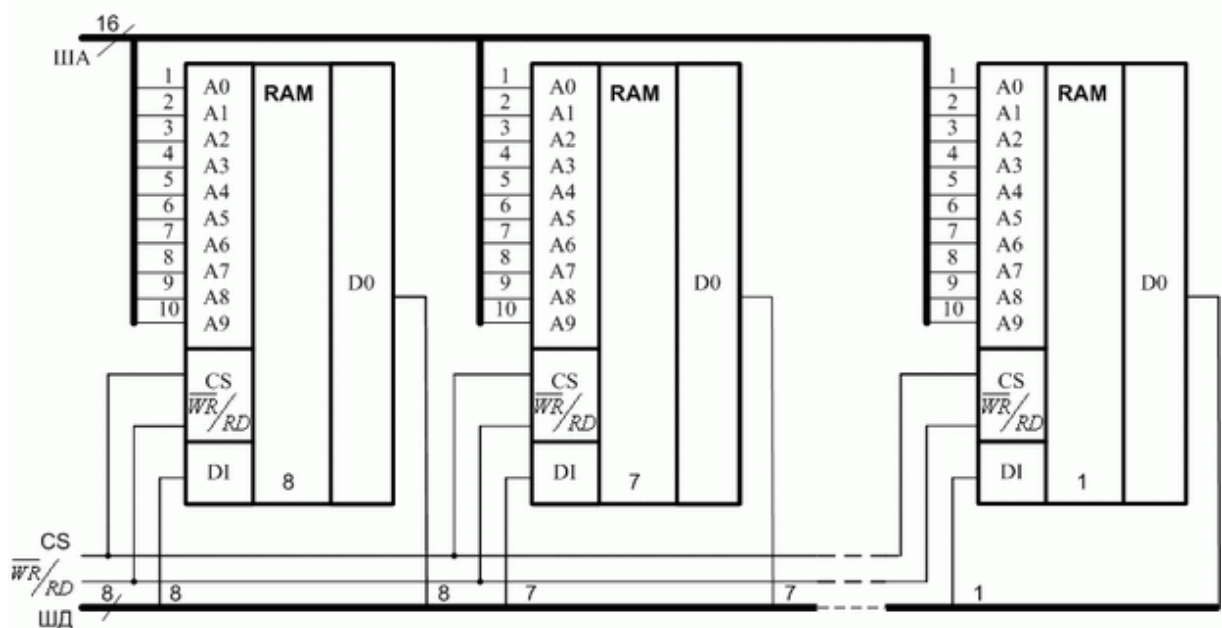


Рис. 13.1. Достижение требуемой разрядности двоичного числа в схемах памяти

Все ИМС группы работают одновременно, поскольку сигналы выбора кристалла CS объединены в одну линию. Аналогично и режим работы всех микросхем будет одинаковым, поскольку линии записи/чтения также объединены. При этом каждая ИМС хранит свой разряд данных. Все линии данных со всех ИМС образуют в данном случае 8-разрядную шину данных ШД.

Увеличение информационного объема при фиксированной разрядности данных

В том случае, когда разрядность всего двоичного числа поддерживается одной ИМС, а количество слов, сохраняемых ею, недостаточно для обеспечения требуемого информационного объема объединяют несколько таких ИМС по следующим правилам:

- одноименные разряды ШД всех ИМС включаются параллельно;
- младшие разряды ША подключаются параллельно ко всем ИМС;
- сигнал выбора кристалла у каждой ИМС свой, он приходит с дешифратора, на входы которого подаются старшие разряды ША, следующие за подключенными параллельно ко всем ИМС;
- сигнал выбора режима подается одновременно на все ИМС.

Построенная по этим правилам схема ПЗУ информационного объема $1K \times 8$ на базе ИМС информационного объема 256×8 каждая показана на [рис. 13.2](#). Здесь младшие восемь разрядов ША подаются одновременно на все ИМС. Следующие два старших разряда ША – 9-й и 10-й – поступают на дешифратор, генерирующий сигналы выбора кристалла для каждой ИМС. Данному подключению соответствует **карта памяти** ([табл. 13.1](#)) – таблица, содержащая начальные и конечные адреса, поддерживаемые каждой ИС схемы памяти. В рассматриваемом примере начальным адресом памяти является адрес 0000_{16} . При подаче этого кода младшие восемь разрядов поступают на все ИС, но в активный режим переводится только первая ИМС, так как код 00_2 на входе дешифратора инициирует наличие активного (единичного) сигнала только на выходе "0" дешифратора, который в качестве сигнала выбора кристалла подается на вход CS первой ИМС.

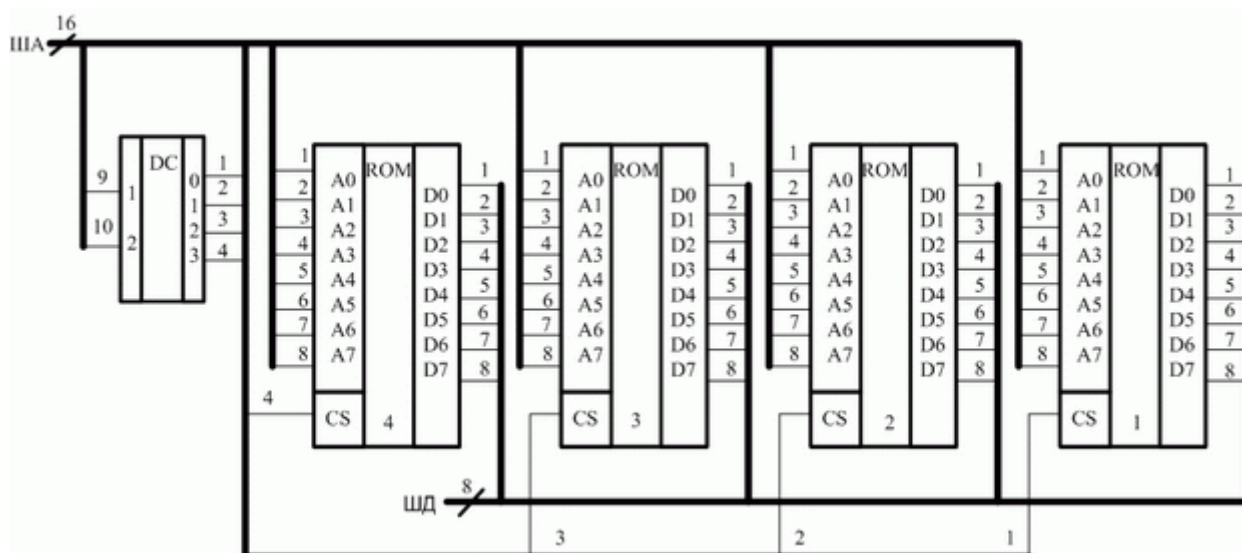


Рис. 13.2. Достижение требуемого информационного объема памяти с фиксированной длиной двоичного числа

На входах *CS* остальных ИМС *ROM* в это время пассивный сигнал логического 0, поэтому они находятся в режиме хранения информации. В таблице указываются только младшие и старшие адреса, поддерживаемые каждой ИМС. Поэтому для всех ИМС в младшем адресе восемь младших разрядов равны нулю, а в старшем адресе – восемь старших разрядов равны 1. Для обеспечения непрерывности массива адресов за старшим адресом, поддерживаемым первой ИМС, следует младший адрес, поддерживаемый второй ИМС. Естественно, они отличаются на 1. Это положение справедливо и для остальных ИМС (табл. 13.1).

Таблица 13.1. Карта памяти для схемы ПЗУ информационного объема 1Кх8 на базе ИМС информационного объема 256х8 каждая		
В двоичном коде	В шестнадцатеричном коде	Активная ИМС (блок) памяти
0000 0000 0000 0000 0000 0000 1111 1111	0000 00FF	1
0000 0001 0000 0000 0000 0001 1111 1111	0100 01FF	2
0000 0010 0000 0000 0000 0010 1111 1111	0200 02FF	3
0000 0011 0000 0000 0000 0011 1111 1111	0300 03FF	4

Краткие итоги

Если разрядности одной ИМС недостаточно для обеспечения нужной разрядности чисел, производится наращивание разрядности по шине данных.

В случае, когда недостаточным является количество адресов одной ИМС памяти, осуществляется наращивание по шине адреса.

Контрольные вопросы:

1. В каком случае нужно соединять несколько ИМС памяти по шине данных?
2. В каком случае нужно соединять несколько ИМС памяти по шине адреса?
3. Какая ИМС служит для обеспечения выбора рабочего блока памяти?
4. Чем ограничивается наращивание разрядности по шине адреса?
5. Чем ограничивается наращивание разрядности по шине данных?

Лабораторное занятие №1

Исследование работы логических элементов ТТЛ и КМОП

Цель работы: изучить схемы элементов ТТЛ и КМОП.

Оборудование: инструкционные карты.

Теоретические сведения:

Все, выпускаемые в настоящее время, стандартные ЦИС малой и средней степени интеграции изготавливаются на основе биполярных или МОП транзисторов. По схемотехническому и конструктивно-технологическому исполнению биполярные микросхемы подразделяют на типы: транзисторно - транзисторные логические ИС (ТТЛ) и их модификации (ТТЛ с диодами Шоттки, Шоттки транзисторная логика и др.); эмиттерно-связанные логические ИС (ЭСЛ). ЦИС на МОП транзисторах подразделяются на:

р - каналные (р-МОП);

п - каналные (п-МОП);

комплементарные на взаимодополняющих п- и р-канальных транзисторах (КМОП).

Микросхемы ТТЛ являются основой для построения аппаратуры среднего и высокого быстродействия и в настоящее время получили наибольшее распространение.

Электрическая схема базового элемента ТТЛ со сложным инвертором приведена на [рис.3.1.](#)

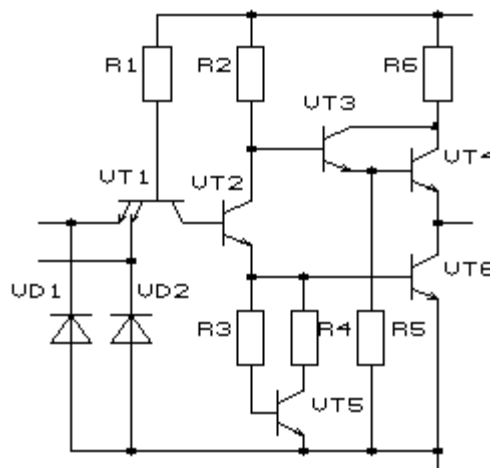


Рисунок 3.1 Электрическая схема элемента ТТЛ.

Резистор R1 и транзистор VT1 реализуют логическую функцию И, транзистор VT2 с резистором R2 и цепочкой, образованной транзистором VT5 и резисторами R3 и R4, выполняют роль фазорасщепляющего каскада для выходного буферного каскада, построенного на транзисторах VT3...VT6. Резистор R5 предназначен для рассасывания объемного заряда из базы транзистора VT4 при выключении инвертора, а резистор R6 - ограничивает “сквозные” токи, протекающие в схеме при ее переключении. Диоды VD1 и VD2 - антизвонные и предназначены для ограничения амплитуды отрицательных выбросов на входах микросхемы, что позволяет предотвратить ложные срабатывания.

Входная характеристика базового элемента ТТЛ приведена на [рис.3.2](#), а его передаточная характеристика - на [рис. 3.3](#).

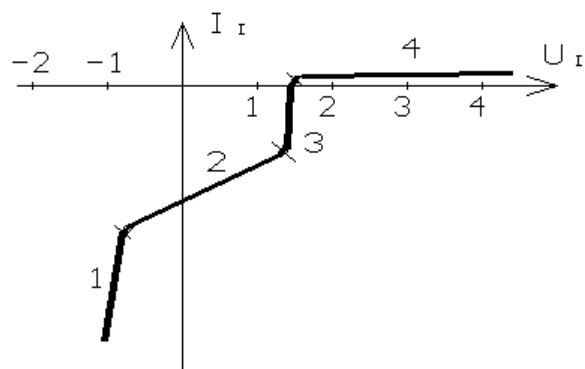


Рисунок 3.2. Входная характеристика ТТЛ- элемента.

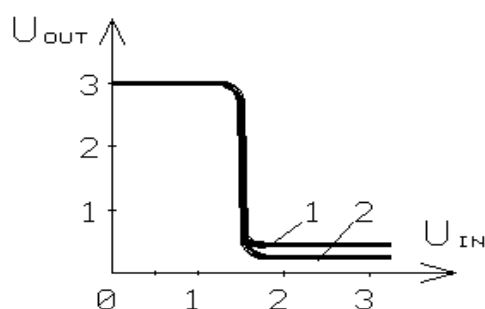


Рисунок 3.3. Передаточная характеристика ТТЛ элемента.

На входной характеристике ТТЛ элемента можно выделить четыре участка: участок 1 определяется антизвонным диодом и характеризует способность микросхемы ограничивать выбросы отрицательной полярности; участок 2 представляет собой прямую линию, наклон которой может быть определен

согласно выражению 3.1.

$$I_{\text{в}} = \frac{V_{\text{с}} - U_{\text{в}} - U^*}{R_{\text{л}}} (1 + M_{\text{э}} b_{\text{л}}) \quad (3.1)$$

где $V_{\text{с}}$ - напряжение питания; $U_{\text{в}}$ - входное напряжение; U^* - прямое падение напряжения на переходе база - эмиттер многоэмиттерного транзистора; $M_{\text{э}}$ - количество эмиттеров, на которые подано напряжение высокого уровня; $b_{\text{л}}$ - инверсный коэффициент усиления транзистора VT1.

Участок 3 соответствует моменту переключения тока базы МЭТ из эмиттера в коллектор и определяет напряжение порога переключения (U_{TH}). Участок 4 входной характеристики определяет входной ток высокого уровня $I_{\text{в}}$, который может быть определен по формуле (3.2)

$$I_{\text{в}} = b_{\text{л}} I_{\text{Рл}} \quad (3.2)$$

$$\text{где } I_{\text{в}} = \frac{V_{\text{с}} - U_{\text{в}} - U^*}{R_{\text{л}}} \quad (3.3)$$

$U_{\text{в}}$ - входное напряжение низкого уровня, поданное на остальные эмиттеры.

На передаточной характеристика условно показано выходное напряжение низкого уровня ($U_{\text{ОЛ}}$) для двух случаев: коэффициент разветвления $K_{\text{раз}} = 10$ (участок 1) и $K_{\text{раз}} = 1$ (участок 2). В общем случае выходное напряжение низкого уровня может быть определено по формуле 3.4, напряжение высокого уровня - в соответствии с выражением 3.5.

$$U_{\text{ОЛ}} = U_0 + (r_{\text{с}} + 4 m f_{\text{T}} / b_{\text{Н}} I_{\text{б}}) I_{\text{Л}} \quad (3.4)$$

$$U_{\text{ОН}} = V_{\text{с}} - (R_2 / (1 + b_{\text{Н}})) I_{\text{Л}} - 2 U^* \quad (3.5)$$

где $b_{\text{Н}}$ - нормальный коэффициент усиления транзисторов VT3, VT4, включенных по схеме Дарлингтона; $r_{\text{с}}$ - объемное сопротивление тела коллектора транзистора VT6; $m f_{\text{T}} / b_{\text{Н}} I_{\text{б}}$ - динамическое сопротивление коллектор-эмиттер транзистора VT6 в режиме насыщения; $I_{\text{б}}$ - ток базы транзистора VT6; $I_{\text{Л}}$ - ток нагрузки.

В настоящее время микросхемы ТТЛ выпускаются только с диодами Шоттки, использование которых позволяет существенно повысить быстродействие ЦИС при одновременном снижении потребляемой мощности. Имеется множество модификаций ТТЛ элементов, наибольший интерес среди которых представляют буферы с тремя состояниями и с

открытым коллектором. Вентили с тремя состояниями ([рис.3.4](#)) являются основой для построения двунаправленных приемопередатчиков, широко используемых при проектировании микропроцессорных систем. Если на вход EZ подан высокий уровень напряжения, то буфер с тремя состояниями инвертирует информацию, поступающую на вход А. Если же на входе EZ низкий уровень напряжения, то выходные транзисторы VT4 и VT6 закрыты, что соответствует высокоимпедансному или третьему состоянию.

Электрическая схема логического элемента ТТЛ типа с открытым коллектором приведена на [рис.3.5](#), важной особенностью которых является то, что они позволяют реализовывать логическую функцию “Монтажное И” и работать на нестандартную нагрузку.

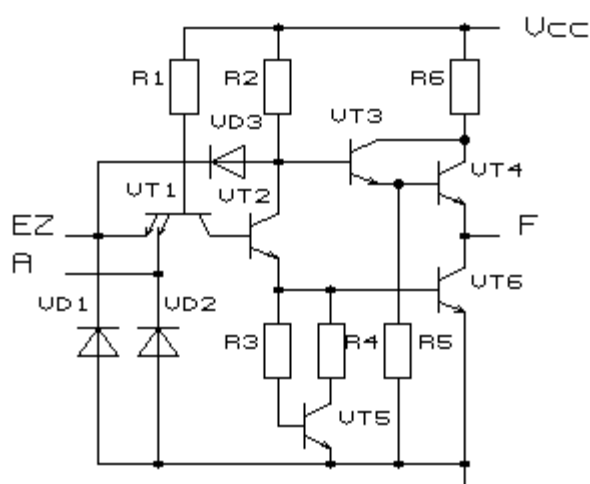


Рисунок 3.4. Электрическая схема буфера с тремя состояниями

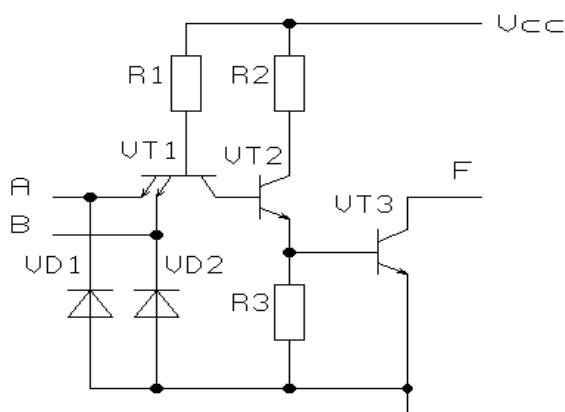


Рисунок 3.5. Электрическая схема вентилля с открытым коллектором

Работа этого элемента ничем не отличается от функционирования базового элемента за исключением того, что к коллектору выходного транзистора

необходимо подключать сопротивление нагрузки, второй вывод которого подсоединен к положительному выводу источника питания.

При проектировании аппаратуры на элементах ТТЛ возникают вопросы, связанные с передачей сигналов между блоками и внутри них, режимом неиспользуемых входов и т.д. Существуют общие рекомендации по решению этих вопросов, которые и будут рассмотрены ниже.

Неиспользуемые входы элементов типа И рекомендуется подключать к положительному выводу источника питания через резистор с сопротивлением 1 кОм. Через один резистор можно подключать не более 20 входов.

Для устранения помех, возникающих в шинах питания и земли в следствии протекания сквозных токов в сложных инверторах, необходимо вблизи разъема и по площади печатной платы устанавливать блокировочные конденсаторы. Емкость конденсатора, устанавливаемого вблизи разъема, рассчитывается из соотношения: 0.1 мкФ на каждые 100 мА потребляемого тока. Дополнительно, на каждые 5 - 10 ЦИС устанавливается блокировочный конденсатор емкостью 0.047 - 0.1 мкФ по площади печатной платы.

При передаче коротких импульсов, длительностью менее 100 нс, с короткими фронтами даже в пределах печатной платы могут возникнуть существенные изменения в его форме. Для предотвращения искажений рекомендуется выход элемента, формирующего короткие импульсы, подключать к общей шине через резистор с сопротивлением 100 Ом. При передаче сигналов между блоками на расстояние до 3 м связи между ними рекомендуется выполнять витой парой, а при длине более 3 м - коаксиальным кабелем с волновым сопротивлением 100 Ом с применением согласующих сопротивлений на передающем и приемном концах.

Элементы КМОП типа в настоящее время являются наиболее перспективными логическими схемами по нескольким причинам: во-первых благодаря своему сверхнизкому потреблению. Во-вторых, благодаря успехам микроэлектронной технологии быстродействие КМОП схем достигло уровня быстродействующих ТТЛ- схем. И в-третьих, возможность работы КМОП схем в широком диапазоне питающих напряжений.

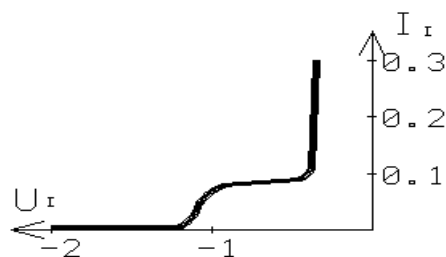


Рис. 3.7. Входная характеристика элемента ЭСЛ.

Типовая электрическая схема КМОП инвертора приведена на [рис.3.8](#).

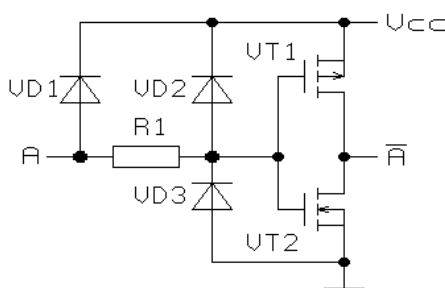


Рисунок 3.8. Электрическая схема КМОП инвертора.

Диоды VD1, VD2, VD3 и резистор R1 образуют схему защиты входа от воздействия статического электричества. Передаточная характеристика КМОП инвертора приведена на [рис.3.9](#). Если входное напряжение $U_{вх} < U_{порн}$, то транзистор VT2 закрыт, а транзистор VT1 открыт. Ток через транзистор VT1 не течет, то $U_{остр} = 0$ и на выходе $U_{вых} = E$.

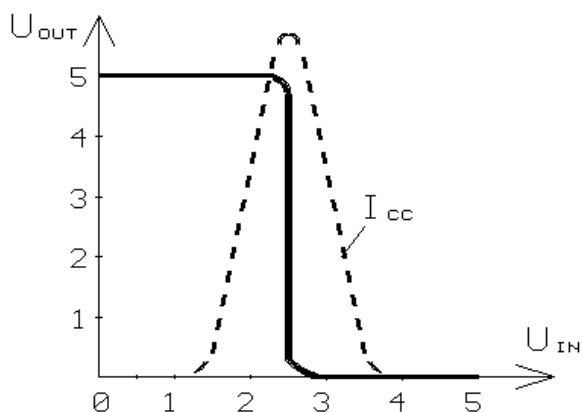


Рисунок 3.9 Передаточная характеристика КМОП инвертора

Когда потенциал входа увеличивается до уровня $U_{вх} = U_{порн}$, открывается транзистор VT2 и в схеме начинает протекать ток:

$$I_{cp} = I_{cn} = 0,5 B_n (U_{вх} - U_{порп}) , (3.6)$$

где B_n - относительная крутизна n-канального транзистора (пропорциональна W/L) и равна 60-100 мкА/В². Падение напряжения на VT1 увеличивается, но пока рабочие точки VT1 находятся на крутом участке выходной характеристики, выходное напряжение уменьшается незначительно.

С дальнейшим ростом $U_{вх}$, происходит увеличение $I_{cn}=I_{cp}$ и рабочие точки транзистора VT1 попадают в пологую область характеристик, когда ток I_{cp} достигает величины тока насыщения p-канальных транзисторов, определяемых выражением:

$$I_{cp} = 0,5 B_p (E - U_{вх} - U_{порр})^2 (3.7)$$

Падение $U_{остр}$ возрастает, а потенциал $U_{вых}$ падает. Значение $U_{вх}$ равное порогу переключения $U_{тн}$ определяется из условия $I_{cn}=I_{cp}$. Решая (3.6) и (3.7) относительно $U_{вх}$ получим:

$$U_{порр} = [U_{порп} + \overline{B_p / B_n} (E - U_{порп})] / (1 + \overline{B_p / B_n}) (3.8)$$

После переключения транзистор VT2 работает в крутой области характеристик и потенциал на выходе $U_{вых} = U_{остп}$. Когда потенциал $U_{вх}$ достигает значения $E - U_{порр}$, транзистор VT1 запирается и на выходе устанавливается потенциал: $U^0 = 0$.

Ширина зоны неопределенности имеет относительно малое значение $U_n < 0,1$ В. При типовых значениях пороговых напряжений $U_{порп}$, $U_{порр} = 1-2$ В помехоустойчивость составляет 2 и более В. Напряжение питания должно быть больше $E > U_{порп} + U_{порр}$.

КМОП инвертор потребляет ток только при входных напряжениях $U_{порп} < u_i < U_{вх} < (E - U_{порр})$. Максимальное значение $I_{сс}$ определяется выражением:

$$I_{сс\max} = 0,5 B_p [(E - U_{порп} - U_{порр}) / (1 + \overline{B_p / B_n})]^2 (3.9)$$

В статическом режиме $I_{сс}=0$. Мощность потребления КМОП схем существенно образом зависит от частоты и емкости нагрузки и определяется выражением:

$$P_{сс} = f C_n E^2 (3.10)$$

Входное сопротивление КМОП схем лежит в пределах $10^3 \dots 10^5$ МОм и определяется токами утечки.

Контрольные вопросы:

1. Назовите элементы транзисторно-транзисторной логики (ТТЛ), поясните схему.
2. Назовите элементы КМОП-логики, поясните схему.
3. Основные параметры логических элементов
4. Элементы какого типа являются наиболее перспективными логическими схемами и почему.

Лабораторное занятие №2

Исследование работы триггеров

Цель работы: изучить функциональные свойства триггеров.

Оборудование: инструкционные карты.

Теоретические сведения:

Триггеры - это простейшие представители цифровых устройств последовательностного типа т.е. устройств или автоматов, обладающих памятью. Последовательностные устройства характеризуются определенным числом внутренних состояний. В каждый конкретный момент времени оно может находиться только в одном из возможных состояний. Переход устройства из одного состояния в другое осуществляется под действием внешних управляющих сигналов. Однако, значение выходного сигнала нельзя определить только по состоянию входных, поскольку оно зависит не только от входной информации, но и от предыдущего состояния устройства. Триггером называется устройство способное находиться в одном из двух устойчивых состояний и скачкообразно переходить из одного в другое под действием внешних управляющих сигналов. Данные состояния триггера определяются как состояние 0 и состояние 1. Триггер может находиться в любом из состояний неограниченный промежуток времени, до поступления внешнего воздействия или отключения питания. Простейший триггер представляет собой одноразрядную ячейку памяти. В общем случае он снабжается определенной входной комбинационной схемой. Триггер снабжается двумя выходами: прямым Q и инверсным $\bar{Q}$. Состояние сигналов на данных выходах может быть только противофазным. Говоря о состоянии триггера подразумевают значение выходного сигнала на выходе Q . При наличии уровня лог. 1 на прямом выходе Q говорят, что: «триггер находится в состоянии 1», либо: «триггер установлен», либо «триггер взведен». При наличии же на данном выходе лог. 0 оперируют понятиями: «триггер находится в состоянии 0», либо: «триггер сброшен». Существующие типы триггеров могут быть классифицированы по различным признакам. Наиболее часто триггеры классифицируют по типу используемых информационных входов. Различают следующие типы основных информационных входов триггера:

- R – раздельный вход сброса триггера ($Q=0$);
- S – раздельный вход установки триггера ($Q=1$);
- K – вход сброса универсального триггера ($Q=0$);
- J – вход установки универсального триггера ($Q=1$);
- T – счетный вход триггера;

D – информационный вход переключения триггера в состояние, соответствующее логическому уровню на этом входе;
C – управляющий или синхронизирующий вход.

Кроме этих основных входов некоторые триггеры могут снабжаться входом V. Вход V блокирует работу триггера и он сколь угодно долго может сохранять ранее записанную в него информацию.

С точки зрения типа используемых входов различают RS-, D-, T-, JK-, VD-, VT- триггеры.

По виду реакции на входные сигналы триггеры подразделяют на асинхронные и синхронные

Асинхронный триггер изменяет свое состояние непосредственно в момент изменения сигнала на его информационных входах. Синхронный триггер изменяет свое состояние лишь в строго определенные (тактовые) моменты времени, соответствующие действию активного сигнала на его синхронизирующем входе C.

УГО асинхронного RS – триггера представлено на рисунке 1.1.

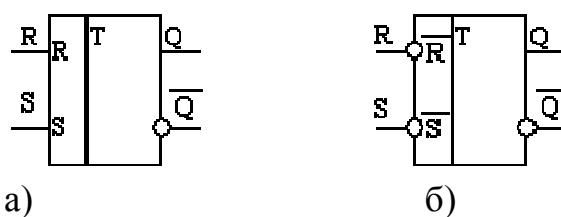


Рисунок 1.1. УГО асинхронного RS – триггера с прямыми и инверсными входами.

По виду активного логического сигнала, действующего на синхронизирующем входе триггеры подразделяют на статические – управляемые уровнем, и динамические – управляемые перепадом входного сигнала. При этом информационные входы могут быть прямыми и инверсными. Для переключения триггера на его прямой вход необходимо подать сигнал X, а на инверсный сигнал. Так, определения "синхронный RS-триггер с прямыми статическими входами" и "синхронный RS-триггер со статическими входами" означают, что рассматриваемый триггер имеет два информационных входа: вход установки S, вход сброса R и синхронизирующий вход C. Переключение триггера происходит в моменты времени, определяемые появлением активного логического сигнала на входе синхронизации ($C = 1$), причем для переключения на входы R или S необходимо подать высокий логический уровень, т. е. сигнал лог.1 ($R = 1$ или $S = 1$). Эти входы соответственно обозначают: C, R, S. УГО данного триггера приведено на рисунке 1.2.а. Если триггер определен как "синхронный RS-триггер с инверсными статическими

входами", то в этом случае, он также имеет два информационных входа: S, R и синхронизирующий вход C. Однако переключение триггера будет происходить при наличии на входе C сигнала лог.0. И активными уровнями для входов R и S будут сигналы лог. 0. УГО данного триггера приведено на рисунке 1.2.б.

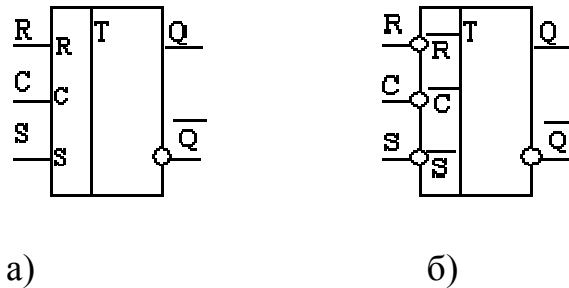


Рисунок 1.2. УГО синхронного RS – триггера с прямыми и инверсными входами.

На УГО RS триггеров допускается менять местами входы R и S. Для триггеров других типов это не допустимо. Простейшие асинхронные RS триггеры строятся на двух элементах 2ИЛИ-НЕ либо 2И-НЕ. Поскольку активными сигналами для элементов ИЛИ, ИЛИ-НЕ являются лог. 1, то триггер, построенный на таких элементах, будет чувствителен к сигналам высокого уровня. Он будет переключаться при подаче на его входы лог.1. Для элементов и, И-НЕ активными уровнями являются уровни лог.0. Поэтому триггер, созданный из данных элементов, будет переходить в новое состояние при поступлении на его входы сигналов лог.0. Функциональные схемы данных триггеров приведены на рисунке 1.3.а и 1.3.б – соответственно. Из рисунка видно, что в триггерах меняется положение входов R и S.

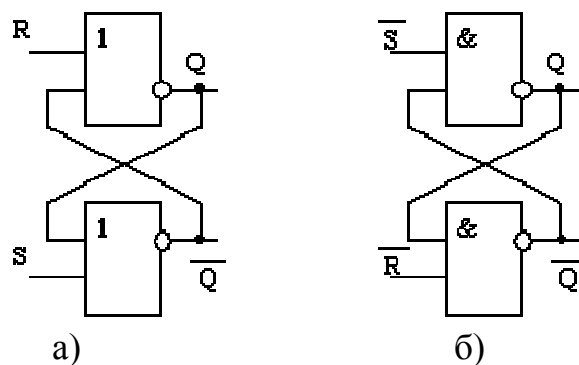


Рисунок 1.3. Функциональные схемы асинхронных RS триггеров: а - на элементах ИЛИ-НЕ; б – на элементах И-НЕ.

Описание работы триггера может задано в виде таблицы истинности или в виде функции алгебры логики (ФАЛ). Отличительной особенностью описания триггеров, как представителей класса последовательностных устройств, является использование в качестве дополнительной входной

переменной значения сигнала Q_n , т.е. предыдущего значения выходного сигнала триггера. Обобщенная таблица истинности асинхронных RS триггеров, построенных на логике ИЛИ-НЕ и И-НЕ представлена в таблице 1.1.

Таблица 1.1. Таблица истинности RS триггеров.

Значения исходных параметров			Состояние выхода			
			Логика ИЛИ-НЕ		Логика И-НЕ	
R	S	Qn	Qn+1		Qn+1	
0	0	0	0	Qn	запрет	
0	0	1	1			
0	1	0	1	1		0
0	1	1	1			
1	0	0	0	0		1
1	0	1	0			
1	1	0	запрет			Qn
1	1	1				

Триггер типа RS не допускает одновременно наличие на входах R и S активных сигналов. В этом случае не выполняется условие его функционирования, поскольку на выходах Q и логические уровни перестают быть взаимно инверсными и принимают одинаковые значения. Данные комбинации считаются запрещенными. Логическое выражение, определяющее функционирование асинхронного RS – триггера с прямыми входами, имеет вид:

$$Q_{n+1} = S \vee \bar{R}Q_n \quad (1.1)$$

а для триггера с инверсными входами:

$$Q_{n+1} = \bar{S}VRQ_n \quad (1.2)$$

Синхронный RS – триггер снабжен дополнительным входом синхронизации C. ФАЛ синхронных RS – триггеров с прямыми и инверсными входами легко получить из выражений (1.1) и (1.2):

$$Q_{n+1} = \bar{C}Q_n \vee C(S \vee \bar{R}Q_n) \quad (1.3)$$

$$Q_{n+1} = \bar{C}Q_n \vee C(\bar{S} \vee RQ_n) \quad (1.4)$$

В качестве примера на рисунке 1.4 приведена функциональная схема синхронного RS триггера на элементах ИЛИ-НЕ.

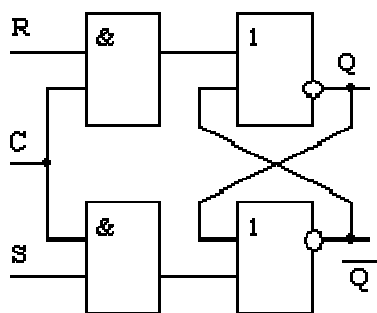


Рисунок 1.4 Функциональная схема синхронного RS триггера на элементах ИЛИ-НЕ.

Триггер – защелка или D-триггер обычно снабжен только одним информационным входом D. Этот триггер может быть только синхронным. Информация со входа D переписывается на выход триггера только по сигналу синхронизации C. До тех пор, пока синхросигнал активен, триггер пропускает входную информацию со входа D на выход Q. При пассивном синхросигнале триггер не чувствителен к изменениям информационного сигнала. Работа D-триггера описывается следующим выражением:

$$Q_{n+1} = \bar{C}Q_n \vee CD \quad (1.5)$$

Функциональная схема D – триггера приведена на рисунке 1.5, а и условное графическое изображения - на рисунке 1.6.

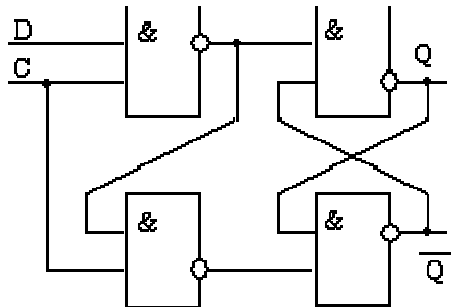


Рисунок 1.5 Функциональная схема D триггера

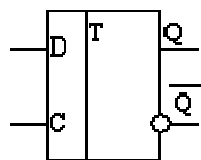


Рисунок 1.6. УГО D – триггера.

Счетный или T-триггер переключается из исходного состояния в противоположное под действием синхросигнала. Из этого следует, что триггер должен быть синхронным. Для его реализации попробуем воспользоваться D-триггером. Построим T-триггер на базе D-триггера, приведенного на рисунке 1.6. Полученная схема приведена на рисунке 1.7. Поскольку D триггер был построен на базе RS-триггера, то следовательно и

Т-триггер можно сделать на его основе. Этот вариант схемы приведен на рисунке 1.8.

Данные схемы переключаются с приходом каждого сигнала активного уровня на вход Т. Схемы достаточно просты, однако они не позволяют реализовать Т-триггер. Поскольку каждая из них не сможет пребывать в устойчивом состоянии в период действия активного сигнала на входе Т. В этом случае схемы переходят в автогенераторный режим. Для исключения данного эффекта Т-триггер строят на базе двухступенчатого триггера.

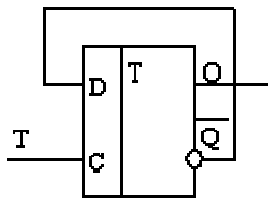


Рисунок 1.7. Реализация Т-триггера на основе D – триггера.

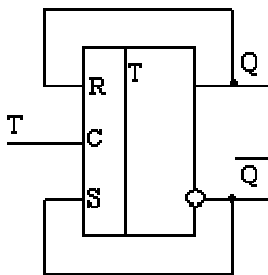


Рисунок 1.8. Реализация Т-триггера на основе RS – триггера.

Двухступенчатые или двухтактные триггеры состоят из двух триггеров. Первый (находящийся левее) называется «ведущий», а второй- «ведомый». Данная структура получила наименование: MS-триггер (Master-Slave). В MS-триггерах не возникает эффекта автогенерации, поскольку каждая ступень данного триггера переключается в то время, когда другая не активна. На рисунке 1.9 представлен вариант построения двухступенчатого триггера на базе D-триггеров, а на рисунке 1.10 – на базе RS-триггеров. Из иллюстраций видно, что ведущий (Master) триггер переключается при наличии сигнала активного уровня (лог.1) на входе С. На его информационные входы поступают сигналы с выходов ведомого. Ведомый (Slave) триггер в этот момент переключится не может, так как на него подается инвертированный сигнал синхронизации. В данном случае он имеет уровень лог. 0. По завершению действия на входе С ведущего триггера активного сигнала и переходе его в уровень лог.0 триггер перестает реагировать на изменения информационных сигналов. Он фиксирует записанную в него информацию. Однако теперь разрешена работа ведомому триггеру, поскольку на его входе синхронизации присутствует активный уровень лог. 1. Триггер воспринимает сигналы с выходов ведущего и переключается в новое состояние, которое и считается состоянием MS-триггера.

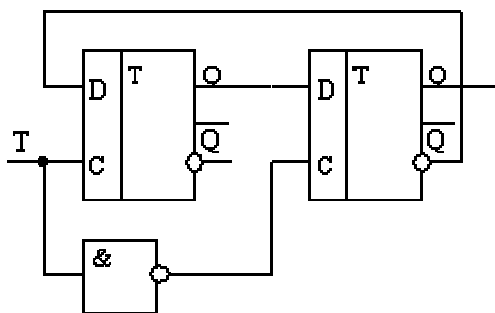


Рисунок 1.9. Реализация двухступенчатого Т-триггера на основе D – триггера.

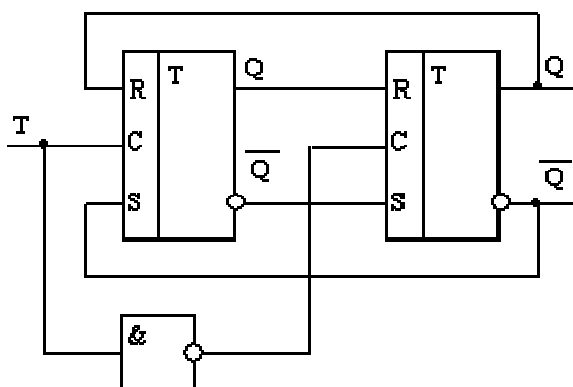


Рисунок 1.10. Реализация двухступенчатого Т-триггера на основе RS – триггера.

ФАЛ Т-триггера имеет вид:

$$Q_{n+1} = \bar{C}Q_n \vee (\bar{Q}_n T \vee Q_n \bar{T}) \quad (1.6)$$

Представленные структуры можно «скрыть» за УГО Т-триггера общего вида. Подобное УГО представлено на рисунке 1.11. По характерному внешнему признаку – наличию только одного информационного входа Т можно предположить, что данный триггер асинхронный. Однако это не так. Из рисунков 1.9 и 1.10 хорошо видно, что информационный вход Т по сути является синхронизирующим входом С. Поэтому триггеры данного типа будем называть «псевдоасинхронные Т-триггеры».

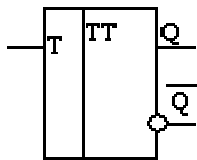


Рисунок 1.11. УГО псевдоасинхронного Т – триггера.

Те Т-триггеры, которые имеют специализированные обособленные информационный и синхронизирующий входы будем считать синхронными. Уго данного триггера приведено на рисунке 1.12. Идентификационным

признаком двухступенчатых триггеров является наличие в поле функциональной спецификации УГО двух символов «ТТ».

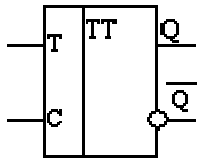


Рисунок 1.12. УГО синхронного Т – триггера.

Универсальный JK триггер также относится к классу двухступенчатых. Он может быть только синхронным. По логике своей работы JK-триггер наиболее близок к RS-триггеру, но в отличие от последнего не имеет запрещенных входных комбинаций. Вход J является входом установки триггера в состояние лог. 1, а вход К – в состояние лог. 0. Наличие активных уровней сигналов на данных входах не нарушает логику работы триггера. В этом случае он функционирует как Т-триггер. Функциональная схема JK-триггера приведена на рисунке 1.13, а УГО – на рисунке 1.14. Универсальный JK-триггер может быть использован в качестве базового для реализации на нем всех типов триггеров.

$$Q_{n+1} = \bar{C}Q_n \vee C(\bar{Q}_n J \vee Q_n \bar{K}) \quad (1.7)$$

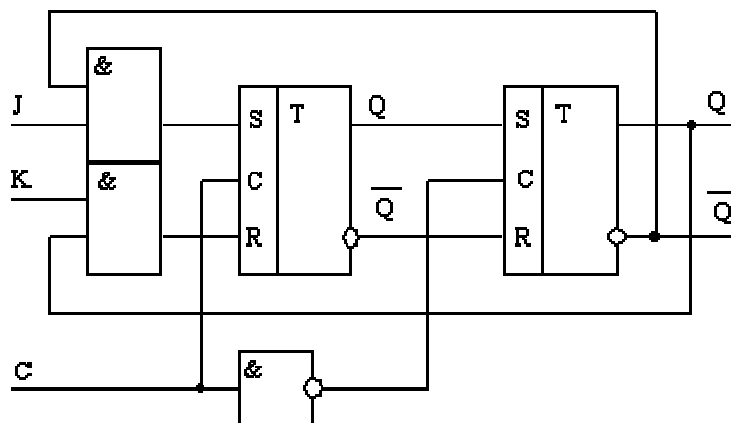


Рисунок 1.13. Функциональная схема JK–триггера.

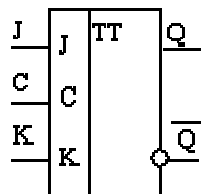


Рисунок 1.14. УГО JK–триггера.

Задание:

1. Начертить функциональные схемы и УГО RS, D, T, JK триггеров, привести их ФАЛ
2. Построить таблицы истинности указанных типов триггеров.
3. Исследовать работу интегрального JK-триггера, составить его таблицу истинности. Определить приоритеты входов.
4. Реализовать на интегральных JK-триггерах RS, D, T триггеры и начертить соответствующие функциональные схемы.
5. При выполнении лабораторной работы руководствоваться методическими указаниями п.6.

Контрольные вопросы :

1. В чем отличие комбинационных и последовательностных логических устройств?
2. Какие триггеры называются асинхронными, а какие синхронными?
3. Чем вызвана необходимость введения двухступенчатости триггеров?
4. Какие типы триггеров должны быть двухступенчатыми?
5. Какой тип триггеров называется «универсальным» и почему?
6. Какой тип триггеров имеет запрещенные комбинации входных сигналов и почему?

Лабораторное занятие №3

Исследование принципа работы функциональных схем регистров

Цель работы: Изучение функциональных свойств универсальных сдвиговых регистров на примере ИМС К155 ИР1.

Оборудование: инструкционные карты.

Теоретические сведения :

Регистр - это микроэлектронное устройство, предназначенное для хранения и преобразования информации, представленной в двоичной форме. Информация представляется двоичными последовательностями (сигналов) в виде «слов» определенной длины. Слова могут иметь длину кратную 4 битам, т.е. 4, 8, 12, 16 и т.д.

Чтобы регистр мог «запомнить» слово информации, он должен состоять из соответствующего числа ячеек, каждая из которых должна иметь два устойчивых состояния: лог.0 и лог.1. Эти состояния должны сохраняться сколь угодно долго, но могут быть изменены в любой момент времени. В качестве ячеек обычно используются синхронные D - триггеры, состояние которых отображает слово информации, а каждый триггер хранит 1 бит информации и соответствует одному разряду слова. В зависимости от наличия и характера связей между ячейками различают: 1) регистры хранения информации; 2) сдвиговые регистры и 3) регистры специального назначения.

Регистры, предназначенные для хранения информации могут работать в двух режимах - записи и хранения информации. При записи «содержимое» ячеек изменяется на новое: в соответствии со значением разрядов входного слова информации. В режиме хранения состояние («содержимое») ячеек регистра остается неизменным пока сохраняется этот режим. Причем во время хранения регистр как бы становится «нечувствительным» к изменению входных информационных сигналов. Такие регистры используются для построения оперативных запоминающих устройств (ОЗУ) ЭВМ. Между ячейками таких регистров взаимных связей нет. Регистры сдвига (сдвиговые регистры) могут работать в трех режимах: 1) записи; 2) хранения и 3) сдвига (последовательной выдачи) информации. В лабораторной работе изучаются эти регистры, поэтому рассмотрим их принцип действия подробнее. На рисунке 3.1 приведено УГО 4-х разрядного сдвигового регистра К155ИР1, подлежащего исследованию. В таблице 3.1 представлено назначение выводов ИМС регистра.

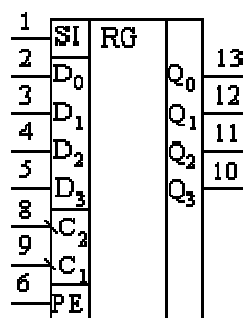


Рисунок 3.1. УГО 4-х разрядного универсального сдвигового регистра К155ИР1

Таблица 3.1. Назначение выводов ИМС К155ИР1

Сигнал	Назначение
SI	последовательный вход
D0 - D3	входы данных для параллельной записи
C1	синхронизация сдвига
C2	синхронизация записи
PE	выбор режима 0 – сдвиг 1 - запись
Q0 - Q3	Выходы

Регистры имеет четыре группы входов:

1. Информационные входы параллельной записи D0...Di;
2. Информационные входы последовательной записи: DR - при сдвиге вправо (вниз); DL - при сдвиге влево (вверх) для ИМС К155ИР13 и SI при сдвиге вправо для ИМС К155ИР1;
3. Входы управления режимом работы (выбора режима) V1 и V2 для ИМС К155ИР13 и PE для ИМС К155ИР1;
4. Входы общего назначения C, C1, C2 используемые для синхронизации и R - вход сброса.

Рассмотрим более подробно особенности организации и работы регистра К155ИР1.

Сдвиговый регистр К155ИР1 имеет разрядность равную 4 и осуществляет однонаправленный сдвиг информации вправо, имеет отдельные входы синхронизации в различных режимах и один вход выбора режима.

Регистр не имеет входа сброса R. Однако функция сброса может быть легко реализована путем записи в регистр числа 0000.

Входы регистра C1 и C2 являются обратными динамическими, поэтому состояние ячеек регистра может изменяться только в момент спада синхронизирующих импульсов на данных входах (1→0).

Вход выбора режима PE служит для управления регистром согласно таблице 3.1. При подаче на данный вход лог.1 регистр переводится в режим записи информации по входам D0 ...D3, вход SI не оказывает влияния на состояние ячеек. В регистре не предусмотрен режим хранения информации. Регистр находится в данном состоянии только во время отсутствия активных сигналов на входах синхронизации C1 и C2. В режиме записи сигналов с информационных входов D0 ...D3 спад сигнала на входе C2 приведет к тому, что слово информации поразрядно одновременно будет занесено в регистр. И на выходах Q0 ...Q3 сигналы примут соответствующие значения (параллельная запись). Если же на вход C2 попадут несколько импульсов, то будет осуществляться повторная запись одного и того же слова информации. А если входное слово успеет измениться, то в регистр будет записано слово, которое оказалось на входах D0 ...D3 в момент последнего тактового импульса. Режим «сдвига» устанавливается подачей на вход PE лог. 0. При этом тактовые импульсы должны подаваться на вход C1. По мере поступления на вход C1 тактовых импульсов, содержимое ячеек регистра «сдвигается» в сторону старших разрядов слова информации. Если принять, что выход Q0 соответствует младшему, а Q3 - самому старшему разрядам слова, то при подаче первого (с момента установки режима) тактового импульса содержимое ячейки Q0 будет переписано в ячейку Q1, содержимое же Q1 - в ячейку Q2 и т.д. Таким образом, слово информации, записанное или хранимое в регистре, первым тактовым импульсом сдвигается (относительно ячеек) в сторону старшего разряда. Очевидно, что второй тактовый импульс сдвинет слово уже на 2 разряда и т.д. При этом с выхода Q3 за 4 такта можно вывести (получить) все слово, начиная со старшего разряда последовательно во времени. Одновременно в ячейку Q0 будет записываться сигнал со входа SI. На рисунке 3.1 УГО регистра показано вертикально и выходы Q0 ...Q3 расположены сверху - вниз. Поэтому сдвиг слова осуществляется «вниз». Если же УГО начертить горизонтально, то выходы Q0...Q3 будут располагаться слева - направо, а сдвиг слова - «вправо». Поэтому Вход SI называется входом последовательной записи информации при сдвиге вправо, а Q3 - выходом регистра с последовательным выводом информации. Регистры с таким алгоритмом функционирования получили название универсальных, так как их можно использовать для хранения и преобразования информации, а также для деления частоты импульсов, построения фазовращателей и распределителей импульсов и выполнения математических операций с двоичными числами. Например, сдвиг на один разряд в сторону старших разрядов записанного в регистр слова, можно рассматривать, как умножение на 2 двоичного числа, отображаемого этим словом.

Задание:

1. Ознакомиться с принципом действия универсального регистра K155ИР1, режимами его работы, УГО, уяснить состав и назначение выводов регистра по рисунку 3.1. и таблице 3.1.

Контрольные вопросы

1. Почему регистры K155ИР13, K155ИР1 называются универсальными? Объясните принцип действия регистров.
2. Как установить режим работы регистра? Какие режимы имеют универсальные регистры? Регистры хранения информации?
3. Что означает режим сдвига «вправо», «влево»? Как они реализуются и зачем используются?
4. Какими особенностями обладают сдвиговые регистры при работе по схеме «замкнутого кольца»? Для чего такое включение регистра можно использовать?
5. Как построить 12-разрядный сдвиговый регистр на основе ИМС K155ИР13 и K155ИР1? В каких режимах может работать такой регистр?
6. Как организовать режим сдвига «влево» у регистра K155ИР1? Приведите схему включения.
7. Каким образом универсальный регистр можно использовать для деления частоты импульсов?

Лабораторное занятие №4

Исследование функциональных схем шифраторов и дешифраторов

Цель работы: исследовать функциональные схемы шифраторов и дешифраторов.

Оборудование: инструкционные карты.

Теоретические сведения:

Дешифратор (декодер)

Дешифратор (декодер) служит для преобразования n-разрядного позиционного двоичного кода в единичный выходной сигнал на одном из 2^n выходов. При каждой входной комбинации сигналов на одном из выходов появляется 1. Таким образом, по единичному сигналу на одном из выходов можно судить о входной кодовой комбинации. Таблица истинности для декодера с двумя входами изображена в таблице 2.1.

Таблица 2.1 – Таблица истинности двухразрядного дешифратора

x1	x2	y0	y1	y2	y3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

Например, устройство должно иметь 4 выхода. Для каждого выхода записываем логическое выражение. На основе СДНФ:

$$y_0 = x_1 \cdot x_2$$

$$y_1 = x_1 \cdot \bar{x}_2$$

$$y_2 = \bar{x}_1 \cdot x_2$$

$$y_3 = \bar{x}_1 \cdot \bar{x}_2$$

По этой системе выражений несложно построить схему требуемого дешифратора (рисунок 2.1).

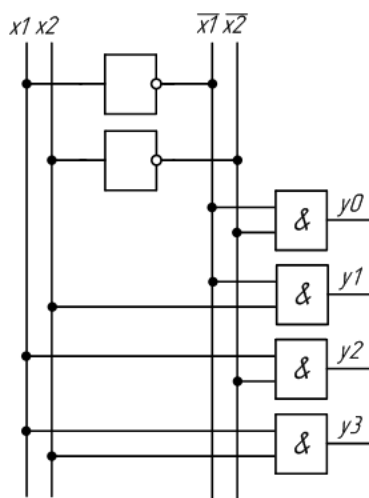


Рисунок 2.1 – Схема дешифратора

Условное графическое обозначение такого дешифратора изображено на рисунке 2.2.

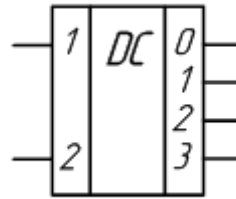


Рисунок 2.2 – Условное графическое обозначение дешифратора

2.2 Шифратор (кодер)

Шифратор выполняет функцию, обратную декодеру (дешифратору), то есть преобразует непозиционный (унитарный) двоичный 2^n разрядный код в n разрядный позиционный код. При подаче на один из входов единичного сигнала на выходе формируется соответствующий двоичный код. Составим таблицу истинности шифратора при $n = 2$.

Таблица 2.2 – Таблица истинности шифратора при $n = 2$

x_1	x_2	x_3	x_4	y_1	y_0
1	0	0	0	0	0
0	1	0	0	0	1
0	0	1	0	1	0
0	0	0	1	1	1

Синтезируем шифратор. Для этого запишем систему его собственных функций:

$$y_1 = x_1 \cdot x_2 \cdot x_3 \cdot x_4 + x_1 \cdot x_2 \cdot x_3 \cdot x_4$$

$$y_0 = x_1 \cdot x_2 \cdot x_3 \cdot x_4 + x_1 \cdot x_2 \cdot x_3 \cdot x_4$$

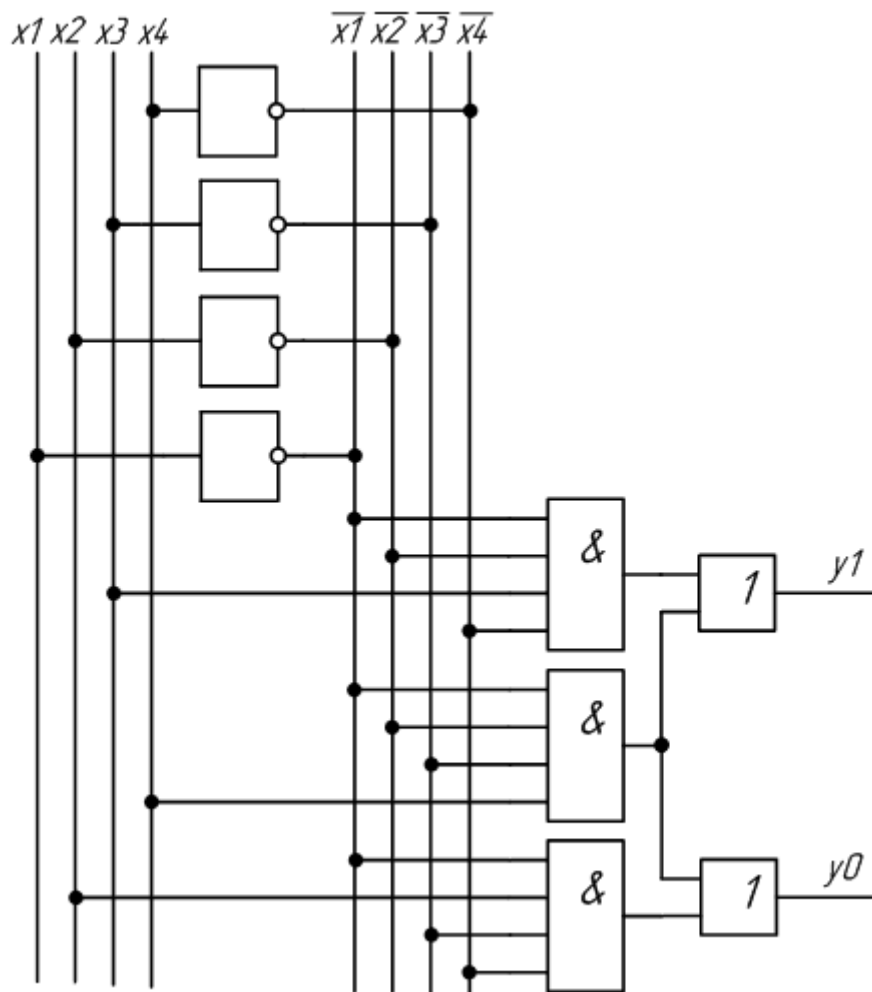


Рисунок 2.3 – Схема шифратора

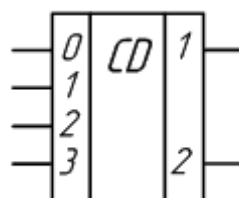
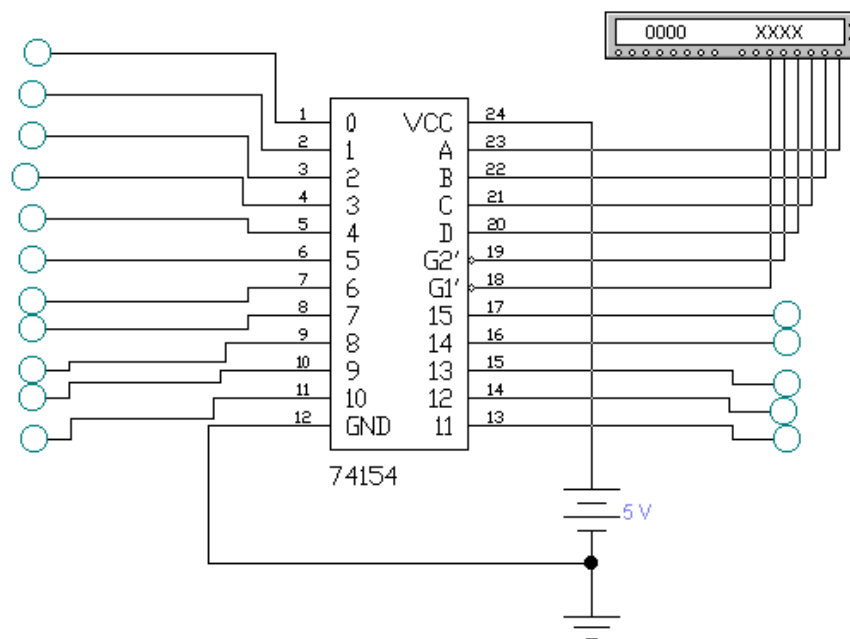


Рисунок 2.4 – Условное графическое обозначение шифратора

Задание:

Исследование шифратора:

Дешифраторы как самостоятельные изделия электронной техники имеют 4, 8 или 16 выходов. Если требуется большее число выходов, дешифраторы наращиваются в систему.



Соберите схему рис. 2.5 и наберите код на генераторе слова как показано на рис. 2.6. С помощью осциллографа просмотрите и зарисуйте входные и выходные сигналы шифратора и поясните его работу.

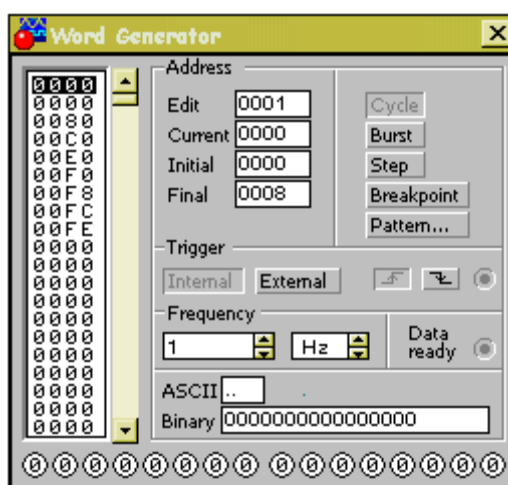


Рис. 2.6. Панель генератора слова с установками для схемы на рис. 1.

2). Подготовьте выходные кодовые комбинации генератора слова в схеме рис. 2, обеспечивающие поочередное включение индикаторов на выходе дешифратора, начиная с выхода 0.

3). Переведите дешифратор рис.2 в режим демультиплексора и выполните задание по п. 2.

Поясните результаты, полученные в пунктах 2 и 3.

Контрольные вопросы:

1. Что такое шифратор и для решения каких задач он используется?
2. Изобразите схему шифратора.
3. Что такое дешифратор и для решения каких задач он используется?

Лабораторное занятие №5

Исследование функциональных схем мультиплексоров и демультиплексоров.

Цель работы: теоретически исследовать функциональные схемы мультиплексоров и демультиплексоров

Оборудование: инструкционные карты.

Теоретические сведения:

1.1 Мультиплексоры

Мультиплексор – это комбинационная многовходовая схема с одним выходом. Входы мультиплексора подразделяются на информационные $D_0, D_1, \dots, D_{n-1}$ и управляющие (адресные) $A_0, A_1, \dots, A_{k-1}$. Обычно $2^k = n$, где k и n – число адресных и информационных входов соответственно. Двоичный код, поступающий на адресные входы, определяет (выбирает) один из информационных входов, значение переменной с которого передается на выход y , т.е. мультиплексор реализует функцию:

$$y = D_i, \text{ если } \sum_{i=0}^{k-1} A_i \cdot 2^i = i \quad (1)$$

Таблица 1

$A_1 A_0$	$D_0 D_1 D_2 D_3$	y
0 0	0 * * *	0
0 0	1 * * *	1
0 1	* 0 * *	0
0 1	* 1 * *	1
1 0	* * 0 *	0
1 0	* * 1 *	1
1 1	* * * 0	0
1 1	* * * 1	1

* – значение входа не влияет на значение y

Таблица функционирования, описывающая работу мультиплексора, имеющего, например, $n = 4$ информационных (D_0, D_1, D_2, D_3) и $k = 2$ адресных (A_0, A_1) входов, представлена в табл. 1.

Вариант схемной реализации мультиплексора “4-1” (“четыре в один”, т.е. коммутирующего данные от одного из четырех входов на единственный выход) и его условное графическое изображение представлены на рис. 1.

Здесь мультиплексор построен как совокупность двухвходовых конъюнкторов данных (их число равно числу информационных входов), управляемых выходными сигналами дешифратора, дешифрирующего двоичный адресный код. Выходы конъюнкторов объединены схемой ИЛИ.

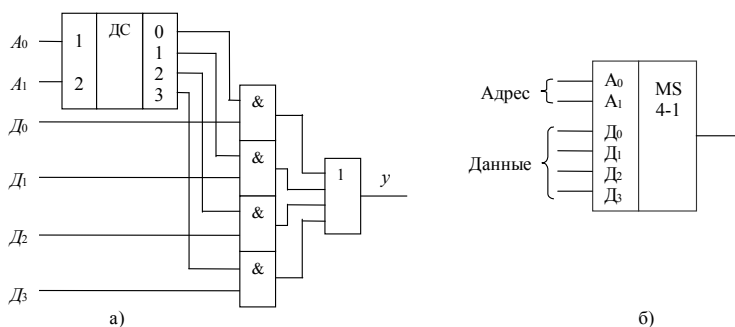


Рис. 1. Схема мультиплексора с дешифратором (а) и УГО (б)

В интегральном исполнении применяется более простая схема, в которой конъюнкторы дешифратора одновременно выполняют и функцию конъюнкторов данных. Работа мультиплексора при этом описывается соотношением

$$y = D_0 \overline{A_1 A_0} + D_1 \overline{A_1} A_0 + D_2 A_1 \overline{A_0} + D_3 A_1 A_0. \quad (2)$$

Из (2) следует, что при любом значении адресного кода все слагаемые, кроме одного равны нулю. Ненулевое слагаемое равно D_i , где i – значение текущего адресного кода.

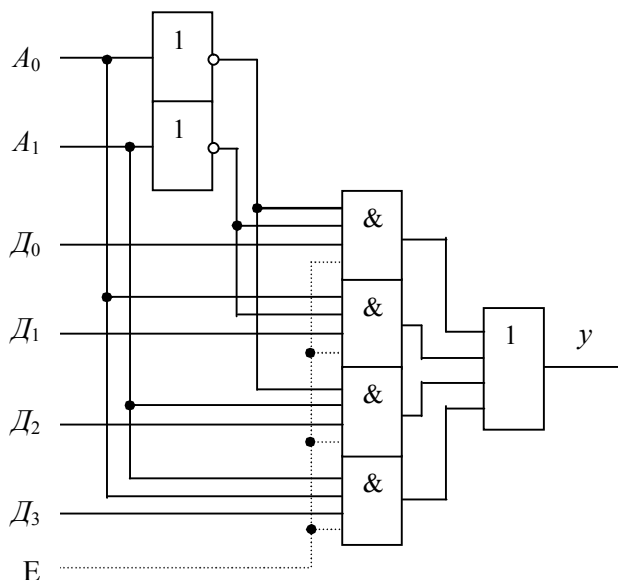


Рис. 2. Схема мультиплексора “4-1”

В соответствии с этим соотношением строятся реальные схемы мультиплексоров, одна из которых для мультиплексора “четыре в один” приведена на рис. 2. Как правило, схема дополняется входом разрешения работы – E (показан пунктирной линией). При отсутствии разрешения работы ($E=0$) выход y становится нулевым и не зависит от комбинации сигналов на информационных и адресных входах мультиплексора.

Мультиплексоры 4-1, 8-1, 16-1 выпускаются в составе многих серий цифровых интегральных схем и имеют буквенный код КП. Например, К555КП1 – мультиплексор 2-1 (в данном корпусе размещаются четыре мультиплексора), К555КП12 – мультиплексор 4-1 (в одном корпусе размещаются два мультиплексора) и т.д.

В тех случаях, когда функциональные возможности ИС мультиплексоров не удовлетворяют разработчиков по числу информационных входов, прибегают к их каскадированию с целью наращивания числа входов до требуемого значения. Наиболее универсальный способ наращивания размерности мультиплексора состоит в построении пирамидальной структуры, состоящей из нескольких мультиплексоров. При этом первый ярус схемы представляет собой столбец, содержащий столько мультиплексоров, сколько необходимо для получения нужного числа информационных входов. Все мультиплексоры этого столбца коммутируются одним и тем же адресным кодом, составленным из соответствующего числа младших разрядов общего адресного кода. Старшие разряды адресного кода используются во втором ярусе, мультиплексор которого обеспечивает поочередную работу мультиплексоров первого яруса на общий выход.

Пирамидальная схема, выполняющая функцию мультиплексора “16-1” и построенная на мультиплексорах “4-1”, показана на рис. 3.

1.2.

т.е.

(y₀,

...

мультиплексоров, $2^k = n$. Двоичный код, поступающий на адресные входы, определяет один из n выходов, на который передается значение переменной с информационного входа (D), т.е. демультиплексор реализует следующие функции:

$$(3) \quad \begin{cases} D, & \text{если } \sum_{i=0}^{k-1} A_i \cdot 2^i = i, \\ 0, & \text{если } \sum_{i=0}^{k-1} y_i = i. \end{cases} \quad i = 0, 1, 2, \dots, n-1; (3)$$

Таблица функционирования демультиплексора, имеющего n = 4 информационных выходов (y₀, y₁, y₂, y₃) и k = 2 адресных входов (A₀, A₁), представлена в табл. 2.

Таблица 2

D	A ₀	A ₁	y ₀	y ₁	y ₂	y ₃	D	A ₀	A ₁	y ₀	y ₁	y ₂	y ₃
0	0	0	0	0	0	0	0	1	0	0	0	0	0
1	0	0	1	0	0	0	1	1	0	0	0	1	0
0	0	1	0	0	0	0	0	1	1	0	0	0	0
1	0	1	0	1	0	0	1	1	1	0	0	0	1

Уравнения, описывающие работу демультиплексора:

$$y_0 = D \cdot \overline{A_1} \cdot \overline{A_0}; y_1 = D \cdot \overline{A_1} \cdot A_0; y_2 = D \cdot A_1 \cdot \overline{A_0}; y_3 = D \cdot A_1 \cdot A_0. \quad (4)$$

Схема демультиплексора, построенная по данным уравнениям и его графическое изображение представлены на рис. 4.

Демультиплексоры

Демультиплексор – схема, выполняющая обратную функции мультиплексора, это комбинационная схема, имеющая один информационный вход (D), n информационных выходов y₁, ..., y_{n-1}) и k управляющих (адресных) входов (A₀, A₁, A_{k-1}). Обычно, также как и

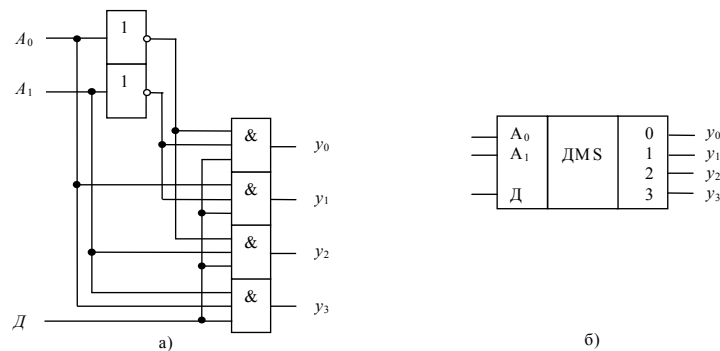


Рис. 4. Схема демультиплексора "1-4" (а)
и его условное изображение (б)

Функция демультиплексора легко реализуется с помощью дешифратора, если его вход “Разрешение” (Е) использовать в качестве информационного входа демультиплексора, а входы 1, 2, 4 ... - в качестве адресных входов демультиплексора $A_0, A_1, A_2, \dots$. Действительно, при активном значении сигнала на входе Е избирается выход, соответствующий коду, поданному на адресные входы. Поэтому ИС дешифраторов, имеющих разрешающий вход, иногда называют не просто дешифраторами, а дешифраторами-демультиплексорами (например, К155ИД4, К531ИД7 и др.).

1.3 Применение мультиплексоров и демультиплексоров

1.3.1. Термином “мультиплексирование” называют процесс передачи данных от нескольких источников по общему каналу, а устройство, осуществляющее на передающей стороне операцию сведения данных в один канал, принято называть мультиплексором. Подобное устройство способно осуществлять временное разделение сигналов, поступающих от нескольких источников, и передавать их в канал (линию) связи друг за другом в соответствии со сменой кодов на своих адресных входах.

На приемной стороне обычно требуется выполнить обратную операцию – демультиплексирование, т.е. распределение порций данных, поступивших по каналу связи в последовательные моменты времени, по своим приемникам. Эту операцию выполняет демультиплексор. Совместное использование мультиплексора и демультиплексора для передачи данных от n источников к n приемникам по общей линии иллюстрирует рис. 5. (В общем случае число источников данных не равно числу приемников).

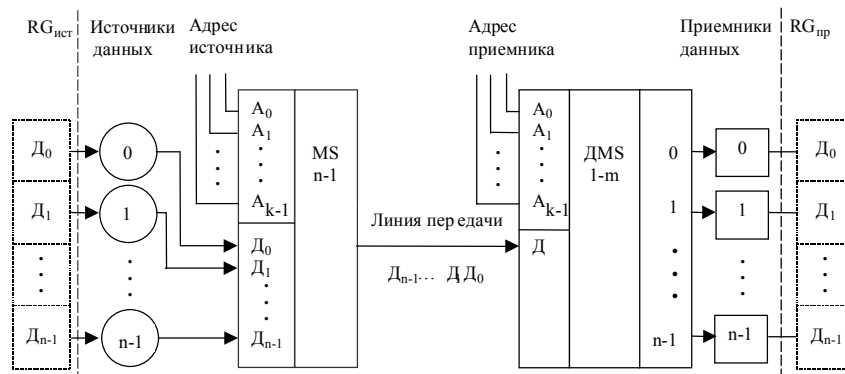


Рис. 5. Схема объединения мультиплексора и демультиплексора
для последовательной передачи данных

1.3.2. Если в схеме (рис. 5) n различных источников и приемников заменить n -разрядными источником и приемником, например, регистрами $RG_{\text{ист.}}$ и $RG_{\text{пр.}}$ (изображены пунктирными линиями), то схема может быть использована для преобразования n -разрядного параллельного кода на передающей стороне в последовательный код (с помощью мультиплексора) и последовательного кода в параллельный на приемной стороне (с помощью демультимплексора). При подобном применении мультиплексора и демультимплексора в качестве их адресных кодов используются выходные сигналы двоичного счетчика, последовательно формирующего на своих выходах двоичные коды чисел от 0 до $n-1$.

1.3.3. Мультиплексор можно использовать в качестве универсального логического элемента для реализации любой логической функции от числа аргументов, равного числу адресных входов мультиплексора. Покажем это на примере логической функции, заданной своей таблицей истинности (табл. 3).

Таблица 3

№	a	b	c	y	№	a	b	c	y
0	0	0	0	0	4	1	0	0	0
1	0	0	1	0	5	1	0	1	1
2	0	1	0	0	6	1	1	0	1
3	0	1	1	1	7	1	1	1	0

Выбираем мультиплексор, имеющий три адресных (по числу аргументов функции) и восемь информационных входов. Для реализации заданной функции информационные входы мультиплексора соединим с уровнями логических “1” и “0” в такой последовательности, которая полностью копирует последовательность единиц и нулей функции в таблице истинности (рис. 6).

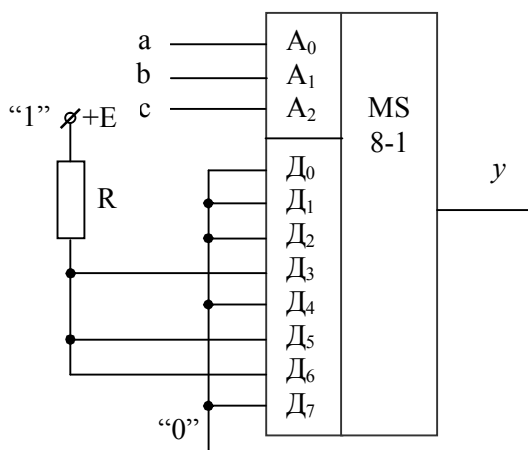


Рис. 6. Применение мультиплексора для реализации логической функции

При этом не требуется ни записи СДНФ, ни ее минимизации. Кстати, функция, заданная табл. 3 (четность числа единиц в трехразрядном слове), не упрощается, поэтому для своей реализации, например, в базисе ЛЭ “И-НЕ” требует четырех ЛЭ “ЗИ-НЕ” и трех инверторов, т.е. в сумме потребуется три ИС. В то же время для реализации схемы по рис. 6 требуется всего одна ИС мультиплексора “8-1”. По этой причине, способ реализации функций трех или большего числа аргументов с помощью ИС мультиплексоров весьма популярен у разработчиков.

Контрольные вопросы:

1. Дайте определение мультиплексора и демультиплексора.
2. Перечислите применения мультиплексоров и демультиплексоров.
3. В чем суть каскадирования мультиплексоров?

Лабораторное занятие №7

Исследование работы функциональной схемы цифрового компаратора

Цель работы: изучение правил выполнения операции сравнения двоичных чисел и исследование принципов построения цифровых компараторов.

Оборудование: персональный компьютер, инструкционные карты, Electronics Workbench.

Теоретические сведения:

Компаратором (устройством сравнения) называют функциональный узел, обеспечивающий сравнение двух чисел А и В. Если А и В – n-разрядные двоичные числа, то компаратор именуют цифровым.

Простейшие компараторы формируют на выходе однобитовый сигнал равенства, или неравенства сравниваемых чисел А и В. Эти отношения используются как логические условия в микропрограммах, в устройствах контроля и диагностики ЭВМ, в устройствах автоматики компараторы используются для сигнализации о выходе величин за установленные пределы и т.д.

Компараторы строятся на основе поразрядных операций над одноименными разрядами обоих слов. Слова равны, если попарно равны все одноименные их разряды. Признак (условие) равенства i-х разрядов сравниваемых слов А и В:

$$r_{ip} = a_i b_i + \bar{a}_i \bar{b}_i = \overline{a_i \bar{b}_i + \bar{a}_i b_i} = \overline{a_i \oplus b_i} = 1 \quad (1)$$

Условие неравенства i-х разрядов:

$$r_{in} = a_i \bar{b}_i + \bar{a}_i b_i = a_i \oplus b_i = 1 \quad (2)$$

Схемная реализация приведенных условий изображена на рис. 1, а.

Схема n-разрядного компаратора на равенство показана на рис.1, б.

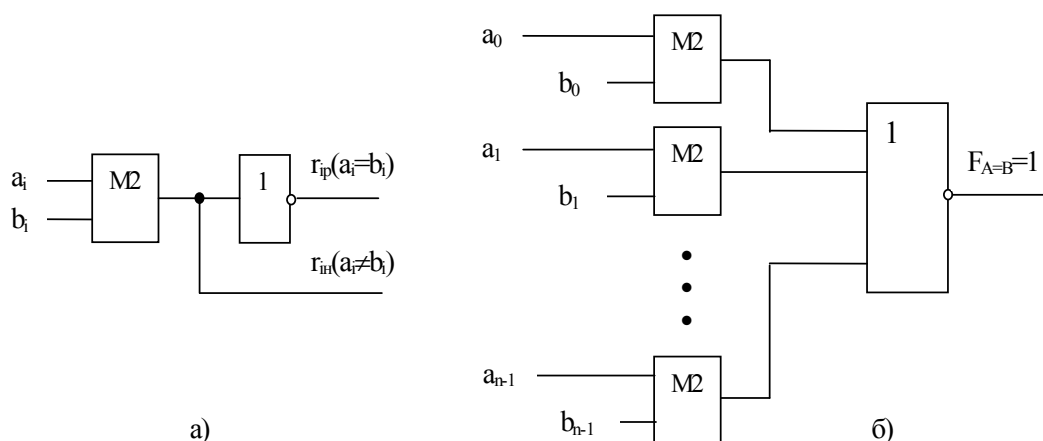


Рис. 1. Схема компаратора 2-х переменных (а) и n-разрядных слов (б)

Более сложные компараторы выявляют не только факт равенства двух n-разрядных чисел, но и сравнивают числа по значению. Такие компараторы

имеют три выхода: “A>B”, “A=B”, “A<B”, и в зависимости от соотношения величин A и B активный уровень (- уровень логической 1) появляется на одном из этих выходов.

Построить такой компаратор можно на базе двоичного сумматора, выполнив на нем операцию вычитания A-B и проанализировав полученный результат. Для этого на сумматор нужно число B подать в дополнительном коде (см. лабораторную работу №4 “Двоичные сумматоры”). Тогда выходной перенос сумматора (p_1) будет равен 0 лишь в том случае, когда A строго меньше B. Равенство разности 0 является признаком того, что A=B. Единица переноса при нулевой сумме указывает на то, что A строго больше B. Сказанное иллюстрируют следующие примеры:

$$\begin{array}{r} \text{A>B} \\ -A \quad -13 \quad +1101 \\ -B \quad -12 \quad +0100 \\ \hline 1.0001 \\ p_1=1 \quad s \neq 0 \end{array}$$

$$\begin{array}{r} \text{A=B} \\ -A \quad -12 \quad +1100 \\ -B \quad -12 \quad +0100 \\ \hline 1.0000 \\ s=0 \end{array}$$

$$\begin{array}{r} \text{A<B} \\ -A \quad -11 \quad +1011 \\ -B \quad -12 \quad +0100 \\ \hline 0.1111 \\ p_1=0 \end{array}$$

Примечание. Вычитание из числа A числа $B=12_{10}=1100_2$ заменено прибавлением к A дополнительного кода числа B, равного 0100₂.

Правила справедливы, если числа A и B рассматриваются как положительные величины, без знака. Если же их старшие разряды трактуются как знаки, то правила будут несколько иные. Их легко вывести самостоятельно, если есть навыки обращения с обратными и дополнительными кодами.

Схема, реализующая описанный алгоритм, изображена на рис. 2.

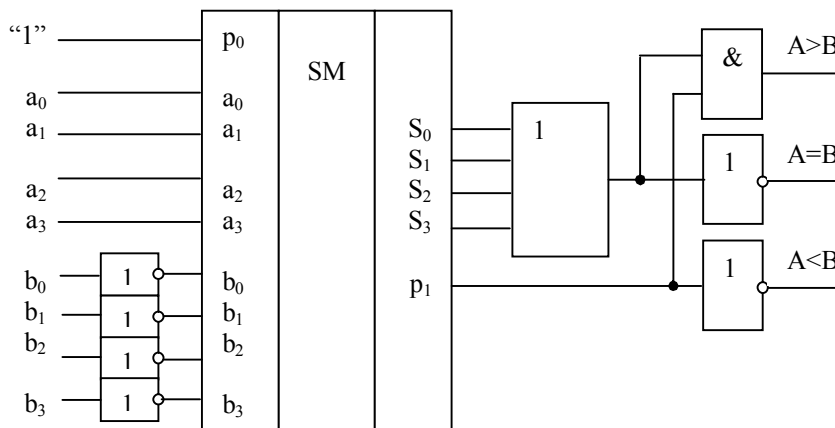


Рис. 2. Схема 4-разрядного компаратора на базе двоичного сумматора

Примером компаратора двоично-кодированных чисел может служить ИС 4-разрядного компаратора К555СП1 (рис. 3). Компаратор имеет 11 входов. Четыре пары входов $a_i b_i$ ($i=0,1,2,3$) используются для подачи на них соответствующих разрядов сравниваемых чисел, входы A<B, A=B, A>B позволяют каскадировать несколько ИС компараторов для увеличения разрядности сравниваемых чисел. Компаратор имеет три выхода

результатов сравнения: $A > B$, $A = B$ и $A < B$. При каскадировании выходы $A > B$, $A = B$ и $A < B$ схемы, сравнивающей младшие разряды, следует присоединить к одноименным входам последующего каскада. Этим способом с помощью двух компараторов СП1 можно сравнивать два восьмиразрядных слова. Нетрудно подсчитать необходимое число каскадов для любой большей длины сравниваемых слов.

Все возможные комбинации поразрядных соотношений входных кодов, а также уровней на входах каскадирования сведены в таблицу, где показаны

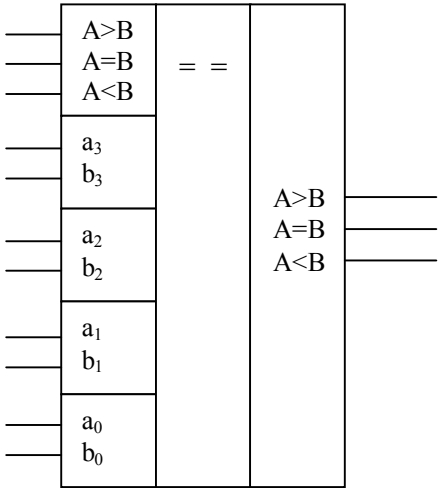


Рис.3. Условное изображение
ИС компаратора K555СП1

соответствующие результирующие уровни на выходах $A > B$, $A = B$ и $A < B$ (табл. 1).

Таблица 1

	Входы сравнения данных				Входы наращивания каскадов			Выходы		
	a_3, b_3	a_2, b_2	a_1, b_1	a_0, b_0	$I(A > B)$	$I(A < B)$	$I(A = B)$	$A > B$	$A < B$	$A = B$
1.	$a_3 > b_3$	x	x	x	x	x	x	B	H	H
2.	$a_3 < b_3$	x	x	x	x	x	x	H	B	H
3.	$a_3 = b_3$	$a_2 > b_2$	x	x	x	x	x	B	H	H
4.	$a_3 = b_3$	$a_2 < b_2$	x	x	x	x	x	H	B	H
5.	$a_3 = b_3$	$a_2 = b_2$	$a_1 > b_1$	x	x	x	x	B	H	H
6.	$a_3 = b_3$	$a_2 = b_2$	$a_1 < b_1$	x	x	x	x	H	B	H
7.	$a_3 = b_3$	$a_2 = b_2$	$a_1 = b_1$	$a_0 > b_0$	x	x	x	B	H	H
8.	$a_3 = b_3$	$a_2 = b_2$	$a_1 = b_1$	$a_0 < b_0$	x	x	x	H	B	H
9.	$a_3 = b_3$	$a_2 = b_2$	$a_1 = b_1$	$a_0 = b_0$	H	H	B	H	H	B

Задание:

2.1. На базе 2-х входных ЛЭ “М2” (К155ЛП5) спроектировать схему и исследовать работу n-разрядного компаратора на равенство, ($n=2$ для 1-й и 2-й бригад; $n=3$ для 3-й, 4-й и 5-й бригад).

Порядок выполнения пункта 2.1 задания. При фиксированном значении одного из сравниваемых чисел, например А, равном номеру вашей бригады, установить значение второго числа (В) равным, большим и меньшим А на единицу. Результаты сравнения свести в таблицу.

2.2. На базе 4-х разрядного двоичного сумматора спроектировать и исследовать работу компаратора с тремя выходами: $A=B$, $A>B$ и $A<B$.

Порядок выполнения пункта 2.2 задания. Четырехразрядный двоичный сумматор построить из двухразрядных сумматоров К155ИМ2.

При фиксированном значении одного из сравниваемых чисел, например А, равном номеру вашей бригады плюс 3, установить значение второго числа (В) равным, большим и меньшим А на 2. Результаты сравнения свести в таблицу.

2.3. Исследовать работу ИС 4-х разрядного компаратора К555СП1.

Порядок выполнения пункта 2.3 задания аналогичен порядку выполнения п. 2.2. Кроме того, для фиксации равенства $A=B$ на входы наращивания ($A>B$) и ($A<B$) следует подать напряжение низкого уровня, а на вход $I(A=B)$ – напряжение высокого уровня (см. строку 9 табл. 1).

Контрольные вопросы:

1. Приведите определение цифрового компаратора и перечислите его применения.

2. Запишите условия равенства (неравенства) одноименных разрядов сравниваемых чисел.

3. Чему равно значение выхода схемы (рис. 1, б) при а) $A=B$, б) $A<B$ и в) $A>B$?

4. Используя ИС К555СП1 спроектируйте схему 8-ми разрядного цифрового компаратора.

Лабораторное занятие №8

Исследование работы АЦП и ЦАП

Цель работы: Изучение схем и принципа работы цифро-аналоговых и аналого-цифровых преобразователей

Оборудование: персональный компьютер, инструкционные карты, ElectronicsWorkbench.

Теоретические сведения:

1.1 Цифро-аналоговые преобразователи

Цифро-аналоговые преобразователи (ЦАП) служат для преобразования информации из цифровой формы в аналоговый сигнал – суммирование токов и напряжений. ЦАП широко применяется в различных устройствах автоматики для связи цифровых ЭВМ с аналоговыми элементами и системами. Принцип работы ЦАП состоит в суммировании аналоговых сигналов, пропорциональных весам разрядов входного цифрового кода, с коэффициентами, равными нулю или единице в зависимости от значения соответствующего разряда кода. ЦАП преобразует цифровой двоичный код $Q_4Q_3Q_2Q_1$ в аналоговую величину, обычно напряжение $U_{\text{вых}}$. Каждый разряд двоичного кода имеет определенный вес i -го разряда вдвое больше, чем вес $(i-1)$ -го. Работу ЦАП можно описать следующей формулой:

$$U_{\text{вых}} = e * (Q_1 * 1 + Q_2 * 2 + Q_3 * 4 + Q_4 * 8 + \dots), \quad (1),$$

где e - напряжение, соответствующее весу младшего разряда,

Q_i - значение i -го разряда двоичного кода (0 или 1).

Например, для числа 1001: $U_{\text{вых}} = e * (1 * 1 + 0 * 2 + 0 * 4 + 1 * 8) = 9 * e$,

для числа 1100: $U_{\text{вых}} = e * (0 * 1 + 0 * 2 + 1 * 4 + 1 * 8) = 12 * e$.

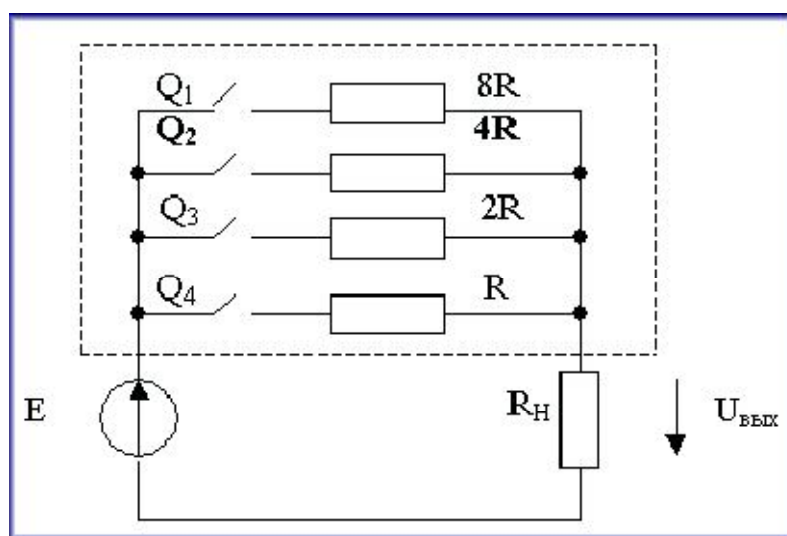


Рис. 1. Схема цифро-аналогового преобразователя

Упрощенная схема реализации ЦАП представлена на рис.1. В схеме i -й ключ замкнут при $Q_i=1$, при $Q_i=0$ – разомкнут. Регистры подобраны таким образом, что $R \gg R_n$. Эквивалентное сопротивление обведенного пунктиром двухполюсника $R_{эк}$ и сопротивление нагрузки R_n образуют делитель напряжения, тогда $U_{вых} = E R_n / R_{эк} + R_n \gg E R_n / R_{эк}$ (2)

Проводимость двухполюсника $1 / R_{эк}$ равна сумме проводимостей ветвей (при $Q_i=1$ i – ветвь включена, при $Q_i=0$ – отключена): $1 / R_{эк} = Q_1 / 8R + Q_2 / 4R + Q_3 / 2R + Q_4 / R$ (3)

Подставив (3) в (2), получаем выражение, идентичное (1)

$U_{вых} = (8E R_n / R) * (Q_1 * 1 + Q_2 * 2 + Q_3 * 4 + Q_4 * 8)$ Очевидно, что $e = 8E R_n / R$. Выбором e можно установить требуемый масштаб аналоговой величины.

2. Аналогово-цифровые преобразователи. В информационных и управляющих системах часть (или вся) информация от датчиков бывает представлена в аналоговой форме. Для ее ввода в цифровые ЭВМ и цифровое управляющее устройство широко применяются аналогово-цифровые преобразователи (АЦП). В большинстве случаев АЦП выполняют преобразование входного напряжения или тока в двоичный цифровой код. Существуют различные типы АЦП. Мы остановимся лишь на тех типах, которые получили в настоящее время наибольшее распространение.

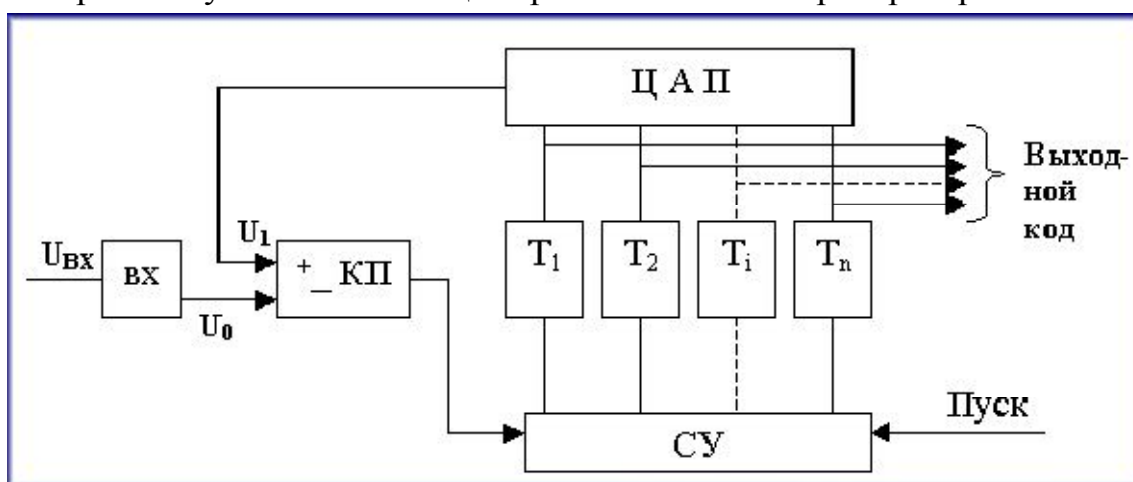


Рис. 2. Схема АЦП последовательного приближения

2.1. АЦП последовательного приближения (АЦППП). Структурная схема АЦППП приведена на рис. 2. Схема работает следующим образом. Входной аналоговый сигнал $U_{вх}$ перед началом преобразования запоминается схемой выборки – хранения ВХ, что необходимо, так как в процессе преобразования необходимо изменение аналогового сигнала. Далее по команде —Пуск с помощью сдвигового регистра последовательно во

времени каждый триггер T_i , начиная со старшего разряда, переводит в положение 1 соответствующий разряд ЦАП. Напряжение U_1 (или ток) с выхода ЦАП сравнивается с входным аналоговым сигналом с помощью компаратора КП. Если $U_0 > U_1$, на выходе компаратора сохраняется низкий уровень и в триггере сохраняется единица, при $U_0 < U_1$ срабатывает компаратор и переводит триггер в положение 0. После окончания цикла на выходах триггеров получается двоичный код, соответствующий (при идеальных элементах) U_0 с точностью до половины младшего разряда. Погрешность АЦП определяется неточностью ЦАП, зоной нечувствительности и смещением нуля компаратора, а также погрешностью схемы выборки – хранения. Поскольку в такой схеме ошибка в каком – либо разряде в дальнейшем не корректируется, необходимо, чтобы время на —взвешивание каждого разряда было достаточно для затухания переходного процесса до уровня, соответствующего половине младшего разряда, и чтобы при разбалансе $U_1 - U_0$ на это значение компаратора успел сработать. Общее время преобразования $t_{пр} = t_{вх} + n(t_{з,к} + t_y + t_{ц}) + t_{сб}$, где $t_{вх}$ – время, необходимое для фиксации $U_{вх}$ схемой ВХ; n – число разрядов; $t_{з,к}$ – время задержки, вносимое компаратором; t_y – время установления U_1 на входе ЦАП; $t_{ц}$ – время задержки цифровых элементов в схеме управления и срабатывания триггера; $t_{сб}$ – время, необходимое для сброса ЦАП в исходное состояние, включая время, необходимое для синхронизации с началом такта. Наибольшую долю в $t_{пр}$ обычно вносит t_y , наибольшая величина которого может быть оценена следующим образом: $t_y = (1+n)T_3 \ln 2$, где T_3 – эквивалентная постоянная времени на входе ЦАП. Если на его выходе включен ОУ, который полезен для уменьшения выходного сопротивления и ускорения тем самым переходного процесса, то $T_3 \gg 1/2\pi f_{ср}$ ($f_{ср}$ – частота среза ОУ по контуру обратной связи). При 12 – разрядном АЦП и использовании быстродействующего ЦАП с $t_y = 100$ нс время $t_{пр}$ близко к 1,5 мкс. В большинстве случаев $t_{пр}$ такого преобразователя достигает 10 – 100 мкс.

2.2. АЦП параллельного типа (АЦПП). Существенное уменьшение $t_{пр}$ удастся получить в АЦП параллельного типа. Его структурная схема приведена на рис.3. Здесь входная аналоговая величина U_0 с выхода схемы ВХ сравнивается с помощью $2^{n+1} - 1$ компараторов с $2(2^n - 1)$ эталонными уровнями, образованными делителями из резисторов равного сопротивления. При этом срабатывают m младших компараторов, образующих на выходах схем И-НЕ нормальный единичный код, затем который с помощью специального дешифратора ДШ преобразуется в двоичный выходной сигнал. Погрешность АЦПП определяется неточностью и нестабильностью

эталонного напряжения, резистивного делителя и погрешностями компараторов. Значительную роль могут играть входные токи компараторов, если делитель недостаточно низкоомный. На рис. 3 приведена структурная схема АЦП параллельного типа.

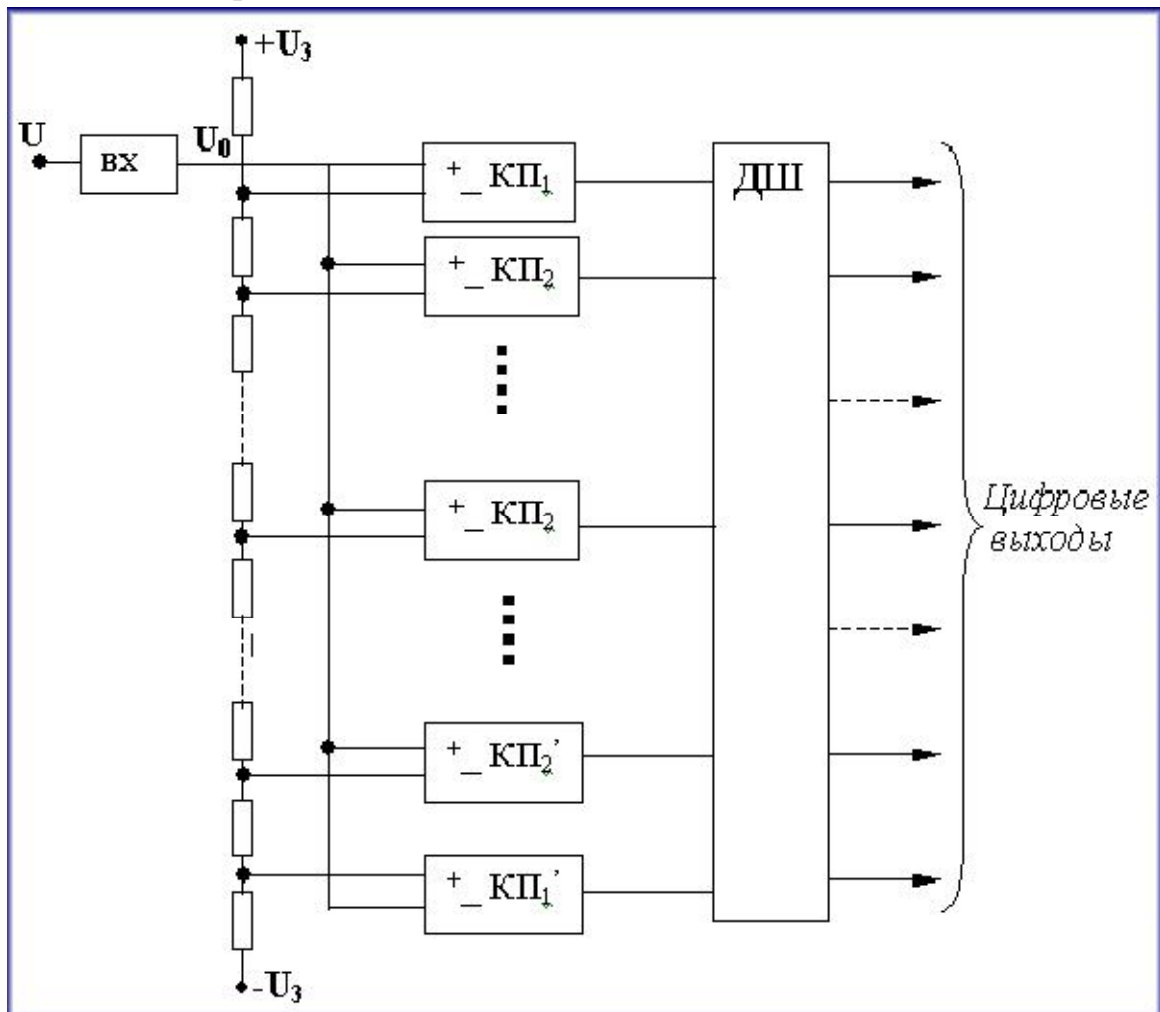


Рис. 3. Схема АЦП параллельного типа

Время преобразования складывается из следующих составляющих: $t_{пр} = t_{вх} + t_{з,k} + a t_{л,сi}$, где $t_{л,сi}$ – Время задержки логических схем; k – число последовательно включенных логических схем. При использовании компараторов со стробированием АЦПП может быть без схемы ВХ. При этом он обеспечивает наибольшее быстродействие по сравнению с любыми другими АЦП.

На рис. 4, показана схема для преобразование аналоговой величины (напряжения) в цифровой код и обратное преобразование цифрового кода в аналоговую величину.

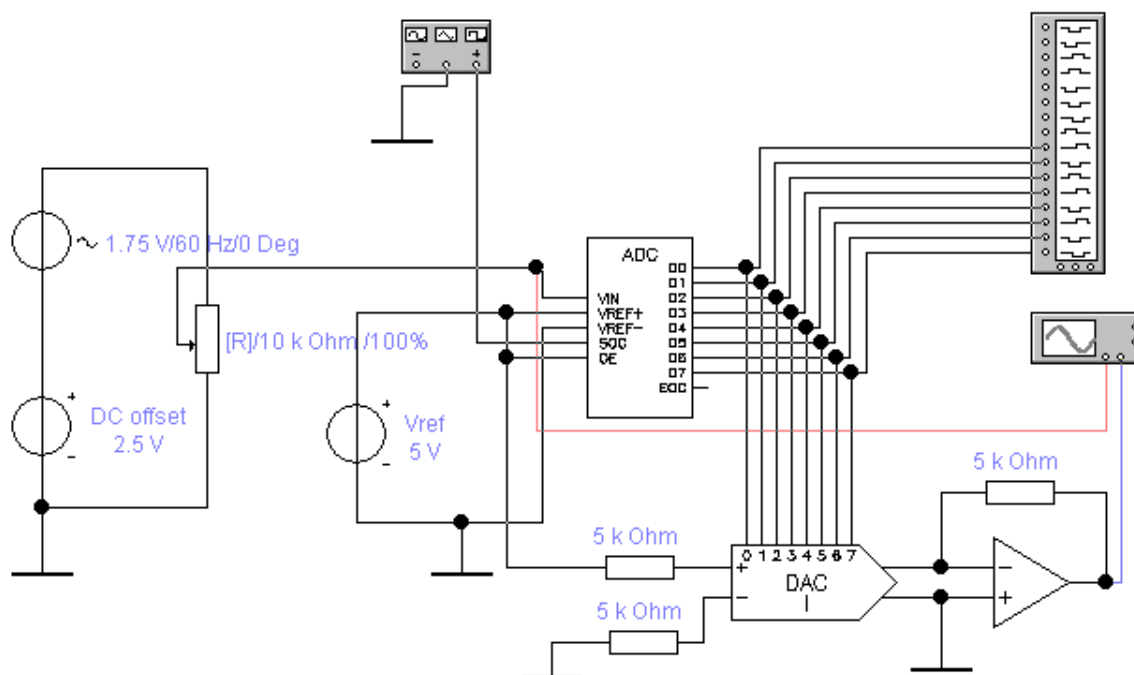


Рис. 4. АЦП и ЦАП.

Процесс работы схемы показан на временной диаграмме. Временная диаграмма, иллюстрирующая работу АЦП и ЦАП, приведена на рисунке 3.3.4.5.

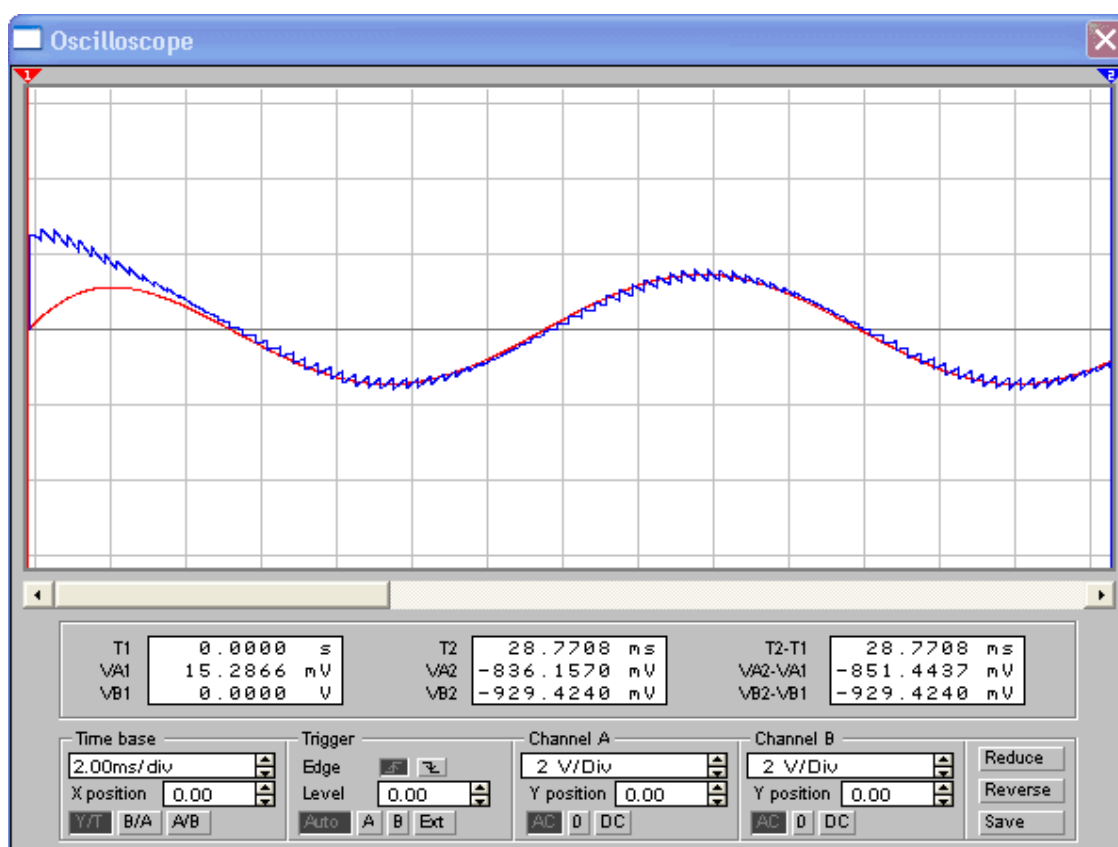


Рис. 5. Диаграмма работы АЦП и ЦАП.

Задание:

1. Используя пакет Electronics Workbench спроектировать схему представленную на рис. 4 , и провести анализ изменения напряжения, меняя частоту и амплитуду, изменить время замыкания и размыкания каждого из ключей.

Контрольные вопросы:

1. Поясните области применения АЦП и ЦАП.
2. Пояснить схему АЦП последовательного типа и последовательного типа.

Лабораторное занятие №9

Изучение структурной схемы микропроцессора

Цель работы: исследование выполнения отдельных команд и простых программ; исследование процесса выполнения команд по машинным циклам, запись программ в машинных кодах и на языке ассемблер.

Оборудование: учебный микропроцессорный комплект (УМК), дополнительная макетная плата (плата ТЭЗ М1 из комплекта УМК), цифровой осциллограф.

Теоретические сведения:

Микропроцессор Intel8080 представляет собой однокристалльный 8-разрядный микропроцессор. Структурная схема микропроцессора представлена на рис. 1. Микропроцессор имеет отдельные 8-разрядную шину данных и 16-разрядную шину адреса, которая обеспечивает адресное пространство памяти 64К байт и отдельные адресные пространства ввода и вывода по 256 байт.

Микропроцессор состоит из устройства управления (УУ), блока регистров общего назначения (РОН), арифметико-логического устройства (АЛУ), схемы управления обменом (СУО), буферных схем данных и адреса (БУ и БД). Блоки микропроцессора объединены через 8-разрядную внутреннюю шину данных.

УУ выполняет функции выборки команды, ее декодирования и выполнения, прием и выдачу управляющих сигналов для составных частей микропроцессора и системы.

Из системы УУ **получает** сигналы:

C1 и C2 – две не перекрывающиеся (т.е. не совпадающие во времени) периодические последовательности импульсов синхронизации;

INT – сигнал требования прерывания;

HLD – сигнал требования прямого доступа в память;

RESET – сигнал сброса (установка микропроцессора в исходное состояние);

RDY – сигнал готовности от составных частей МП системы.

УУ **выдает** в систему сигналы:

INTE – разрешение прерывания;

HLDA – разрешение прямого доступа в память;

WAIT – ожидание (микропроцессор находится в состоянии ожидания);

SYN – сигнал начала машинного цикла (выдается в начале каждого цикла).

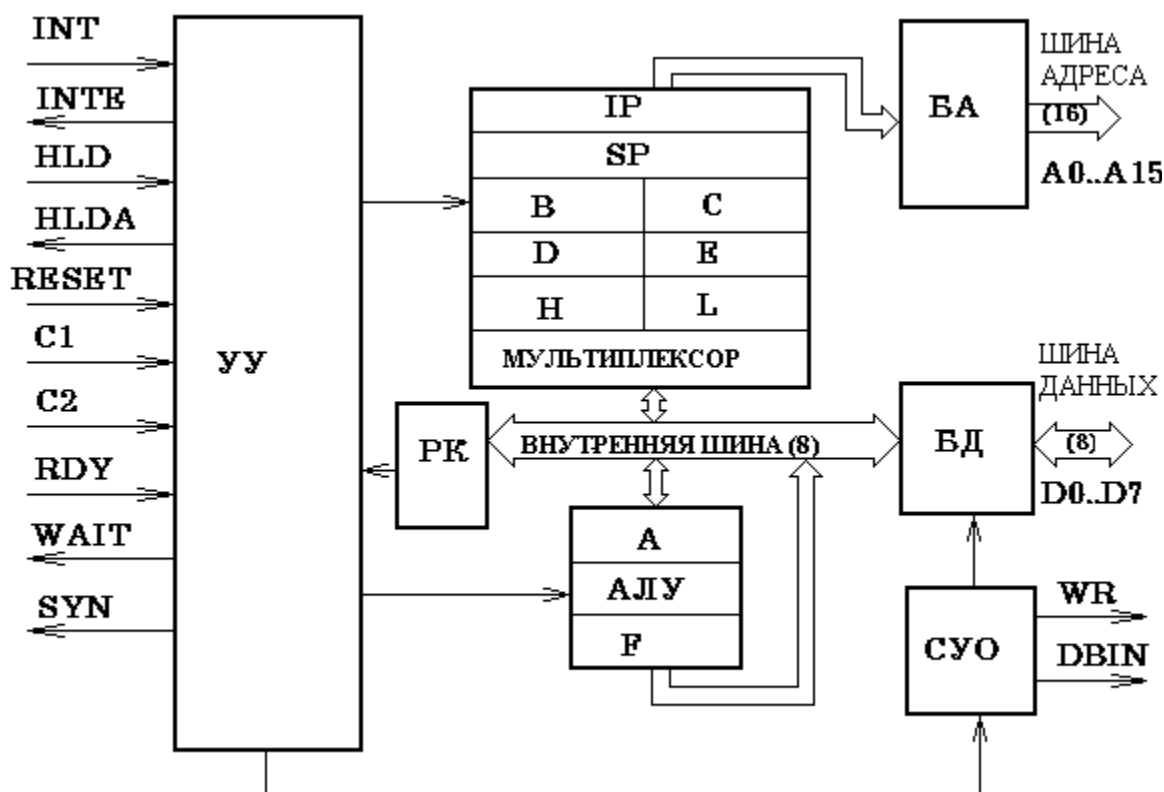


Рис.1. Функциональная схема микропроцессора Intel8080

Выполняемая команда хранится в специальном регистре, который называется регистром команд (ПК).

Арифметико-логическое устройство совместно с аккумулятором (А) и регистром признаков (F) выполняет арифметические и логические операции с двоичными числами (они носят название операндов). В аккумуляторе хранится один из операндов перед выполнением операции и результат после выполнения операции. В регистр признаков помещаются признаки событий, происходящих при выполнении операций (например, перенос из старшего значащего разряда, нулевой результат операции, отрицательный результат и т.д.).

Блок РОН включает 6 8-разрядных регистров (B,C,D,E,H,L), которые могут использоваться попарно как 3 16-разрядных (BC, DE, HL), 16-разрядный регистр-счетчик команд (IP) и 16-разрядный регистр – указатель стека (SP). Схема управления обменом выдает сигналы, синхронизирующие обмен данными между микропроцессором и МП системой. Сигнал WR сопровождает данные, выводимые из МП, а сигнал DBIN – синхронизирует прием информации. Этих сигналов явно недостаточно для управления всей системой, поэтому в начале каждого машинного цикла по шине данных передается служебная информация о типе операции, которая будет выполнена в текущем цикле. Эта информация сопровождается сигналом SYN, принимается специальной дополнительной схемой (она называется системный контроллер), которая, используя эту информацию, формирует все необходимые сигналы шины управления:

MRDC - чтение данных из памяти,

MWTC - запись данных в память, IORC - чтение данных из порта ввода, IOWC - запись данных в порт вывода.

Буферные схемы БА и БД обеспечивают необходимую нагрузочную способность шин адреса и данных, а также возможность отключения шин от внешней, по отношению к МП, системы (путем перевода выходов в третье состояние).

Микропроцессор имеет фиксированный набор из 244 команд. Выполнение команд происходит по машинным циклам. Каждый цикл – это выполнение элементарной операции по управлению шинами МП системы. Время выполнения команды определяется процессом получения, декодирования и выполнения команды. В зависимости от вида команды время ее выполнения может составлять 1-5 машинных циклов. Машинный цикл состоит из 3-5 машинных тактов. Такт – это один период синхронизирующих импульсов. При типовой частоте импульсов синхронизации, равной 2 МГц, длительность одного такта равна 500 нс.

Для КР580ВМ80А имеется 10 различных типов машинных циклов: выборка кода команды (цикл М1), чтение данных из памяти, запись данных в память, чтение данных из стека, запись данных в стек, ввод данных из внешнего устройства, запись данных во внешнее устройство, цикл обслуживания прерываний, останов, обслуживание прерывания в режиме останова. Первым машинным циклом при выполнении любой команды является цикл М1. Тип выполняемого цикла, как указывалось выше, определяется информацией, выдаваемой на шину данных в начале каждого цикла. Соответствие типа цикла и сигналов в каждом разряде шины данных приведено в таблицах 2 и 3 книги УМК.ЭКСПЛУАТАЦИОННАЯ ДОКУМЕНТАЦИЯ.(лист 11). На шине УМК (плата ТЭЗ М1) разряду D0 соответствует сигнал ST0, разряду D1 – ST1 и т.д.

В каждом машинном цикле МП проверяет состояние сигнала «Готов»(RDY) на своем входе. При нулевом сигнале на этом входе работа МП приостанавливается. В УМК это используется для выполнения программы в пошаговом режиме. Одно нажатие кнопки «ШГ» приводит к выполнению одной команды или одного машинного цикла программы.

2. Исследование выполнения команд по машинным циклам

2.1. Присоединить к УМК дополнительную плату ТЭЗ М1. Общий провод платы М1 соединить с входом заземления, а контакт ST5 – с входом внешнего запуска осциллографа. Переключить осциллограф в режим внешней синхронизации. Пригласить преподавателя для проверки схемы. **ВНИМАНИЕ! НЕЛЬЗЯ ВКЛЮЧАТЬ СОЕДИНЕННУЮ СХЕМУ БЕЗ РАЗРЕШЕНИЯ ПРЕПОДАВАТЕЛЯ. ЭТО МОЖЕТ ПРИВЕСТИ К ВЫХОДУ ИЗ СТРОЯ УМК.**

2.2. Включить УМК и осциллограф. Поставить переключатели разверток по X – 1 В/дел, по Y – 1 мкс/дел. Добиться появления на экране линии развертки.

2.3.1. С адреса 0800Н ввести в УМК последовательность шестнадцатеричных чисел 3A 00 09 C3 00 08. Эта последовательность

представляет собой две команды. Первая (3A 00 09) – команда извлечения числа из памяти и передача его в аккумулятор микропроцессора. Число 3A – код команды, 00 09 – адрес в памяти пересылаемого числа (этот адрес равен 0900H – в МП действует соглашение, что для двухбайтовых чисел младший байт расположен по меньшему адресу, старший – по большему). Вторая команда (С3 00 08) передает управление по адресу 0800H, т.е. приведенная программа – это бесконечное выполнение команды 3A 00 09.

2.3.2. Переключить УМК в пошаговый режим работы (кнопки РБ/ШГ и КМ/ЦК нажаты). Запустить программу, подав команду СТ 0800 ВП. Она остановится на первом шаге, причем на индикаторах будет высвечиваться информация. Нажимая кнопку ШАГ, несколько раз выполним программу, информацию занесем в таблицу 1.

Таблица 1. Шаг	Адрес	Данные	Состояние	Тип цикла	Комментарий
1	0000100000 000000	00111010	10100010	M1	Загрузка команды
2	0000100000 000001	00000000	10000010	чтение	Чтение мл. байта адреса
3	0000100000 000010	00001001	10000010	чтение	Чтение ст. байта адреса
4	0000100100 000000	10111111	10000010	чтение	Чтение из памяти по адресу 0900H
5	0000100000 000011	11000011	10100010	M1	Загрузка команды
6	0000100000 000100	00000000	10000010	чтение	Чтение мл. байта адреса
7	0000100000 000101	00001000	10000010	чтение	Чтение ст. байта адреса
8	0000100000 000000	00111010	10100010	M1	Повтор шага 1

Примечание. На шаге 4 информация на индикаторах Данные может отличаться от приведенной в таблице.

Из анализа таблицы 1 видим, что команда 3A 00 09 выполняется за 4 цикла (цикл M1 и 3 цикла чтения памяти). В первом цикле происходит загрузка кода команды 3A (находится в памяти по адресу 0800H) в РК микропроцессора. Во втором цикле читается младший байт адреса 00 (в памяти по адресу 0801H). В третьем цикле читается старший байт адреса 09 (в памяти по адресу 0802H). В четвертом цикле число, находящееся по адресу 0900 (в нашем случае оно равно 0BFH), считывается в аккумулятор микропроцессора.

2.3.3. Снять временную диаграмму выполнения команды 3A0009.

2.3.3.1. Переключить УМК в непрерывный режим работы (отжать кнопки КМ/ЦК и РБ/ШГ).

2.3.3.2. Снять и зарисовать временные диаграммы сигналов на контактах ST5, F2, IORC, IOWC, MRDC, MWTC, A0, A1, A8, D0. Все сигналы по времени на осциллографе привязаны к началу цикла M1 и должны быть

зарисованы в одном временном масштабе. Сигнал F2 совпадает с сигналом синхронизации C2, и его период равен одному такту. Пример временной диаграммы приведен на рис.2.

2.3.3.3. Проанализируем нашу временную диаграмму. Она показывает, что команда выполняется за 13 периодов сигнала F2, т.е. за 13 тактов, причем цикл M1 выполняется за 4 такта, а циклы чтения за 3. Из сигналов синхронизации обмена информацией активным (активный уровень этих сигналов – низкий) бывает только сигнал MRDC (сопровождение чтения из памяти), что и следовало ожидать. Активным уровнем сигналов на шине адреса является низкий уровень, а на шине данных - высокий.

2.3.4. Записать в УМК с адреса 0800H 22 00 09 C3 00 08. Выполнить программу в пошаговом режиме и результат занести в таблицу, аналогичную таблице 1. Снять временную диаграмму выполнения команды 22 00 09.

Контрольные вопросы:

1. За сколько машинных циклов выполняется команда? За сколько тактов?
2. Какие циклы используются при выполнении команды?
3. Какие сигналы синхронизации обмена информацией бывают активны при выполнении команды?
4. Какие действия, по-вашему, выполняет команда?

Перечень рекомендуемых учебных изданий, дополнительной литературы, Интернет-ресурсов.

1. Калабеков Б.А. Цифровые устройства и микропроцессорные системы. М.: Горячая линия — Телеком, 2000
2. Браммер Ю.А, Пащук И.Н. Импульсные и цифровые устройства М.: Высшая школа, 1999
3. В.В. Мукосеев, И.Н. Сидоров «Маркировка и обозначение радиоэлементов. Системы цветовой и буквенно-цифровой маркировки отечественных и зарубежных радиоэлектронных элементов. Справочник. М: Горячая линия — Телеком, 2001
4. Бирюков С.А. Применение цифровых микросхем серий ТТЛ и КМОП. — М.: ДМК, 1999
5. Тарабрин Б.В. и др. Интегральные микросхемы. Справочник М.: Радио и связь, 1997
6. Панфилов Д.И. и др. Электротехника и электроника в экспериментах и упражнениях. Практикум на ElectronicsWorkbench — Т.2: Электроника. — М.: ДОДЭКА, 2000
7. Мизерная З.А. Цифровые системы передачи Методическое пособие М.; Маршрут, 2003
8. Мизерная З.А. Цифровые устройства, микропроцессоры и их программирование. Альбом М.; Маршрут