

ФЕДЕРАЛЬНОЕ АГЕНТСТВО ЖЕЛЕЗНОДОРОЖНОГО ТРАНСПОРТА

Федеральное государственное бюджетное образовательное учреждение высшего образования
**«Петербургский государственный университет путей сообщения
Императора Александра I»
(ФГБОУ ВО ПГУПС)**

Петрозаводский филиал ПГУПС

ОДОБРЕНО

на заседании цикловой комиссии

протокол № 10 от 20.06.2017

Председатель цикловой комиссии:

[подпись] (Петрозаводский филиал)

УТВЕРЖДАЮ

Начальник УМО

[подпись]

А.В. Калько

«20» 06

2017 г.

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

**по организации и проведению практических
занятий/лабораторных работ**

По учебной дисциплине
ОП.09. Цифровая схемотехника

Специальность: 27.02.03 Автоматика и телемеханика на транспорте
(железнодорожном транспорте)

Разработчик: Зайцев В.А.

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Методические указания по организации и проведению лабораторных работ разработаны в соответствии с рабочей программой учебной дисциплины ОП.09. Цифровая схемотехника

и предназначены для выполнения лабораторных работ обучающимися.

Лабораторные работы по учебной дисциплине направлены на усвоение знаний, освоение умений и формирование элементов общих компетенций, предусмотренных рабочей программой учебной дисциплины.

В результате освоения учебной дисциплины обучающийся 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;

В результате освоения учебной дисциплины происходит поэтапное формирование элементов общих и/или профессиональных компетенций:

ОК 1. Понимать сущность и социальную значимость своей будущей профессии, проявлять к ней устойчивый интерес.

ОК 2. Организовывать собственную деятельность, выбирать типовые методы и способы выполнения профессиональных задач, оценивать их эффективность и качество.

ОК 9. Ориентироваться в условиях частой смены технологий в профессиональной деятельности

ПК 1.1. Анализировать работу станционных, перегонных, микропроцессорных и диагностических систем автоматики по принципиальным схемам.

ПК 1.2. Определять и устранять отказы в работе станционных, перегонных, микропроцессорных и диагностических систем автоматики.

ПК 1.3. Выполнять требования по эксплуатации станционных, перегонных, микропроцессорных и диагностических систем автоматики.

Рабочей программой предусмотрено выполнение обучающимися практических занятий, включая, как обязательный компонент практические задания с использованием персонального компьютера.

Распределение результатов освоения учебного материала в ходе выполнения лабораторных работ/заданий на практических занятиях происходит в соответствии с таблицей 1.

Таблица 1 – Распределение результатов освоения учебного материала

Раздел, тема	Контрольно-оценочные мероприятия	Кол-во часов	результаты		Поэтапно формируемые элементы общих и профессиональных компетенций
			Усвоенные знания	Освоенные умения	
Тема 1.1 Формы представления числовой информации в цифровых устройствах	Практическое занятие №1 Кодирование чисел различных системах счисления	2	Позиционные системы счисления. Двоичная, восьмеричная, десятичная, шестнадцатеричная системы счисления	Умение переводить числа из одной системы счисления в другую	ОК.1, ОК 2
Тема 1.2 Арифметические операции с кодированным и числами	Практическое занятие №2 Выполнение арифметических операций с многоразрядными двоичными кодированными числами»	2	Правила выполнения арифметических операций с двоичными кодами. Выполнение операции вычитания в ЭВМ	Умение выполнять арифметические операции с двоичными многоразрядными кодированными числами	ОК.1, ОК 2
Тема 2.1 Функциональная логика	Практическое занятие №3 Изучение логических операций	2	Таблицы истинности логических элементов «И», «ИЛИ», «НЕ», проектирование устройств	Умение выполнить соединение логических элементов для реализации заданной логической функции	ОК.1, ОК 2 ОК 9
Тема 2.2 Основы синтеза цифровых логических устройств	Лабораторная работа №1 Исследование логических микросхем.	2	Логические функции реализуемые микросхемами : 2И-НЕ, 2ИЛИ-НЕ, исключающее ИЛИ	Умение определять логические элементы по таблицам истинности и по маркировке микросхем	ОК.1, ОК 2 ОК 9 ПК1.1, ПК 1.2. ПК 1.3
Тема 3.1 Цифровые триггерные схемы	Лабораторная работа №2 Исследование триггеров	2	Синхронные и асинхронные триггеры. RS, D, T, JK триггеры	Умение составлять таблицы состояний триггеров	ОК.1, ОК 2 ОК 9 ПК1.1, ПК 1.2. ПК 1.3
Тема 3.2 Цифровые счетчики импульсов	Лабораторная работа №3 Исследование счетчиков электрических импульсов	2	Принцип работы суммирующих, вычитающих и реверсивных счетчиков.	Уметь конструировать кольцевые счетчики и счетчики с изменяемым модулем счета	ОК.1, ОК 2 ОК 9 ПК1.1, ПК 1.2. ПК 1.3
Тема 3.3 Регистры	Лабораторная работа №4	2	Технические характеристики	Умение выполнять ввод двоичных кодов	ОК.1, ОК 2 ОК 9

	Исследование регистров		параллельных и последовательных регистров	в параллельные и последовательные регистры	ПК1.1, ПК 1.2. ПК 1.3
Тема 4.3 Мультиплексоры и демультимплексоры	Лабораторная работа №5 Исследование мультиплексора и демультимплексора	2	Назначение и структуру мультиплексора и демультимплексора	Умение управлять работой микросхем мультиплексоров и демультимплексоров.	ОК.1, ОК 2 ОК 9 ПК1.1, ПК 1.2. ПК 1.3
Тема 4.4 Комбинационные двоичные сумматоры	Лабораторная работа №6 Исследование сумматора	2	Структура и таблица состояний полусумматора и сумматора	Умение выполнять сложение и вычитание двоичных кодов с помощью микросхем сумматоров	ОК.1, ОК 2 ОК 9 ПК1.1, ПК 1.2. ПК 1.3
Тема 4.5 Цифровые компараторы	Лабораторная работа №7 Исследование работы арифметико-логического устройства	2	Структура АЛУ. Входные и выходные сигналы. Применение АЛУ	Умение составлять коды команд для выполнения действий АЛУ	ОК.1, ОК 2 ОК 9 ПК1.1, ПК 1.2. ПК 1.3
Тема 5.2 Оперативные запоминающие устройства	Лабораторная работа №8 Исследование микросхем оперативной памяти	2	Назначение микросхем ОЗУ. Динамическая, статическая, синхронная, асинхронная память	Умение подключать микросхемы ОЗУ, записывать и считывать информацию из ячеек памяти	ОК.1, ОК 2 ОК 9 ПК1.1, ПК 1.2. ПК 1.3
Тема 5.3 Постоянные запоминающие устройства	Лабораторная работа №9 Исследование микросхем ПЗУ	2	Технические характеристики микросхем ПЗУ. Способы записи и стирания информации	Умение определять тип микросхемы, умение программировать микросхемы ПЗУ	ОК.1, ОК 2 ОК 9 ПК1.1, ПК 1.2. ПК 1.3
Тема 7.3 Программирование микропроцессора	Лабораторная работа №10 Изучение структуры микропроцессорной системы. Эмулятор	2	Понятие микропроцессора и микропроцессорной системы. Назначение и порядок взаимодействия узлов процессора	Умение имитировать работу микропроцессора с помощью эмулятора.	ОК.1, ОК 2 ОК 9 ПК1.1, ПК 1.2. ПК 1.3

Содержание практических занятий/лабораторных работ охватывает весь круг умений и компетенций, на формирование которых направлен(а) учебная дисциплина/МДК/ПМ.

ПЕРЕЧЕНЬ ПРАКТИЧЕСКИХ ЗАНЯТИЙ/ЛАБОРАТОРНЫХ РАБОТ

Практическое занятие №1. Кодирование чисел в различных системах счисления

Практическое занятие №2. Выполнение арифметических операций с многоразрядными двоичными кодированными числами»

Практическое занятие №3 Изучение логических операций

Лабораторная работа №1	Исследование логических микросхем
Лабораторная работа №2	Исследование триггеров
Лабораторная работа №3	Исследование счетчиков электрических импульсов
Лабораторная работа №4	Исследование регистров
Лабораторная работа №5	Исследование мультиплексора и демультиплексора
Лабораторная работа №6	Исследование сумматора
Лабораторная работа №7	Исследование работы арифметико-логического устройства
Лабораторная работа №8	Исследование микросхем оперативной памяти
Лабораторная работа №9	Исследование микросхем ПЗУ
Лабораторная работа №10	Изучение структуры микропроцессорной системы. Эмулятор

КРИТЕРИИ ОЦЕНИВАНИЯ ПРАКТИЧЕСКИХ РАБОТ

При оценке освоенных умений при выполнении практических работ применяется пятибалльная шкала оценивания/ дихотомическая шкала оценивания.

Оценивание практических занятий/лабораторных работ производится в соответствии со следующими нормативными актами:

- Положение о текущем контроле успеваемости и промежуточной аттестации обучающихся;

- Положение о планировании, организации и проведении лабораторных работ и практических занятий.

Практическая работа №1

Тема: Кодирование чисел в различных системах счисления

Цель: Получение практических навыков перевода чисел из одной системы счисления в другую, проведение арифметических операций над числами в различных системах счисления.

Перечень необходимых средств обучения:

Краткие теоретические сведения

Под системой счисления (СС) понимается способ представления любого числа с помощью алфавита символов, называемых цифрами.

СС называется позиционной, если одна и та же цифра имеет различное значение, которое определяется ее местом в числе.



Десятичная СС является позиционной. На рисунке слева значение цифры 9 изменяется в зависимости от, ее положения в числе. Первая слева девятка делает вклад в общее значение десятичного числа 900 единиц, вторая — 90, а третья — 9 единиц.

Римская СС является **непозиционной**. Значение цифры X в числе XXI остается неизменным при вариации ее положения в числе. Количество различных цифр, употребляемых в позиционной СС, называется **основанием** СС. В десятичной СС используется десять цифр: 0, 1, 2, ..., 9; в двоичной СС — две: 0 и 1; в восьмеричной СС — восемь: 0, 1, 2, ..., 7. В СС с основанием Q используются цифры от 0 до Q - 1.

В общем случае в позиционной СС с основанием Q любое число x может быть представлено в виде **полинома**:

$$x = \underbrace{a_n \cdot Q^n + a_{n-1} \cdot Q^{n-1} + \dots + a_1 \cdot Q^1 + a_0 \cdot Q^0}_{\text{целая часть}} + \underbrace{a_{-1} \cdot Q^{-1} + a_{-2} \cdot Q^{-2} + \dots + a_{-m} \cdot Q^{-m}}_{\text{дробная часть}}$$

где в качестве коэффициентов, могут стоять любые цифры, используемые в данной СС.

Принято представлять числа в виде последовательности входящих в полином соответствующих цифр (коэффициентов):

$$x = a_n a_{n-1} \dots a_1 a_0 , a_{-1} a_{-2} \dots a_{-m}$$

Запятая отделяет целую часть числа от дробной части. В ВТ чаще всего для отделения целой части числа от дробной части используют точку. Позиции цифр, отсчитываемые от точки, называют разрядами. В позиционной СС вес каждого разряда отличается от веса (вклада) соседнего разряда в число раз, равное основанию СС. В десятичной СС цифры 1-го разряда — единицы, 2-го — десятки, 3-го — сотни и т. д.

В ВТ применяют позиционные СС с недесятичным основанием: двоичную, восьмеричную, шестнадцатеричную системы и др. Для обозначения используемой СС числа заключают в скобки и индексом указывают основание СС: $(15)_{10}; (1011)_2; (735)_8; (1EA9F)_{16}$. Иногда скобки опускают и оставляют только индекс: $15_{10}; 1011_2; 735_8; 1EA9F_{16}$.

Есть еще один способ обозначения СС: при помощи латинских букв, добавляемых после числа. Например, $15D; 1011B; 735Q; 1EA9FH$.

Установлено, что, чем больше основание СС, тем компактнее запись числа. Так двоичное изображение числа требует примерно в 3,3 раза большего количества цифр, чем его десятичное представление. Рассмотрим два числа: $97D = 1100001B$. Двоичное представление числа имеет заметно большее количество цифр.

Несмотря на то, что десятичная СС имеет широкое распространение, цифровые ЭВМ строятся на двоичных (цифровых) элементах, так как реализовать элементы с десятью четко различимыми состояниями сложно

Шестнадцатеричная и восьмеричная СС используются при составлении программ на языке машинных кодов для более короткой и удобной записи двоичных кодов — команд, данных, адресов и операндов. Перевод из двоичной СС в шестнадцатеричную и восьмеричную СС (и обратно) осуществляется достаточно просто. В табл. 1 приведены некоторые числа, представленные в различных СС.

Правила перевода чисел из одной системы в другую

1. Для перевода двоичного числа в десятичное необходимо его записать в виде многочлена, состоящего из произведений цифр числа и соответствующей степени числа 2, и вычислить по правилам десятичной арифметики:

$$X_2 = A_n \cdot 2^{n-1} + A_{n-1} \cdot 2^{n-2} + A_{n-2} \cdot 2^{n-3} + \dots + A_2 \cdot 2^1 + A_1 \cdot 2^0$$

При переводе удобно пользоваться таблицей степеней двойки:

Таблица 4. Степени числа 2

n (степень)	0	1	2	3	4	5	6	7	8	9	10
2^n	1	2	4	8	16	32	64	128	256	512	1024

Пример . Число 11101000_2 перевести в десятичную систему счисления.

$$11101000_2 = 1 \cdot 2^7 + 1 \cdot 2^6 + 1 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0 = 232_{10}$$

2. Для перевода восьмеричного числа в десятичное необходимо его записать в виде многочлена, состоящего из произведений цифр числа и соответствующей степени числа 8, и вычислить по правилам десятичной арифметики:

$$X_8 = A_n \cdot 8^{n-1} + A_{n-1} \cdot 8^{n-2} + A_{n-2} \cdot 8^{n-3} + \dots + A_2 \cdot 8^1 + A_1 \cdot 8^0$$

При переводе удобно пользоваться таблицей степеней восьмерки:

Таблица 5. Степени числа 8

n (степень)	0	1	2	3	4	5	6
8^n	1	8	64	512	4096	32768	262144

Пример . Число 75013_8 перевести в десятичную систему счисления.

$$75013_8 = 7 \cdot 8^4 + 5 \cdot 8^3 + 0 \cdot 8^2 + 1 \cdot 8^1 + 3 \cdot 8^0 = 31243_{10}$$

3. Для перевода шестнадцатеричного числа в десятичное необходимо его записать в виде многочлена, состоящего из произведений цифр числа и соответствующей степени числа 16, и вычислить по правилам десятичной арифметики:

$$X_{16} = A_n \cdot 16^{n-1} + A_{n-1} \cdot 16^{n-2} + A_{n-2} \cdot 16^{n-3} + \dots + A_2 \cdot 16^1 + A_1 \cdot 16^0$$

При переводе удобно пользоваться таблицей степеней числа 16:

Таблица 6. Степени числа 16

n (степень)	0	1	2	3	4	5	6
16^n	1	16	256	4096	65536	1048576	16777216

Пример . Число $FDA1_{16}$ перевести в десятичную систему счисления.

$$FDA1_{16} = 15 \cdot 16^3 + 13 \cdot 16^2 + 10 \cdot 16^1 + 1 \cdot 16^0 = 64929_{10}$$

4. Для перевода десятичного числа в двоичную систему его необходимо последовательно делить на 2 до тех пор, пока не останется остаток, меньший или равный 1. Число в двоичной системе записывается как последовательность последнего результата деления и остатков от деления в обратном порядке.

Пример. Число 22_{10} перевести в двоичную систему счисления.

22	2
22	11
0	10
	5
	4
	2
	2
	2
	0
	1

$$22_{10} = 10110_2$$

5. Для перевода десятичного числа в восьмеричную систему его необходимо последовательно делить на 8 до тех пор, пока не останется остаток, меньший или равный 7. Число в восьмеричной системе записывается как последовательность цифр последнего результата деления и остатков от деления в обратном порядке.

Пример. Число 571_{10} перевести в восьмеричную систему счисления.

571	8
56	71
11	64
8	8
3	7
	8
	8
	0
	1

$$571_{10} = 1073_8$$

6. Для перевода десятичного числа в шестнадцатеричную систему его необходимо последовательно делить на 16 до тех пор, пока не останется остаток, меньший или равный 15. Число в шестнадцатеричной системе записывается как последовательность цифр последнего результата деления и остатков от деления в обратном порядке.

Пример. Число 7467_{10} перевести в шестнадцатеричную систему счисления.

7467	16
7456	466
11	464
	29
	16
	13
	1

$$7467_{10} = 1D2B_{16}$$

7. Чтобы перевести число из двоичной системы в восьмеричную, его нужно разбить на триады (тройки цифр), начиная с младшего разряда, в случае необходимости

дополнив старшую триаду нулями, и каждую триаду заменить соответствующей восьмеричной цифрой (табл. 3).

Пример. Число 1001011_2 перевести в восьмеричную систему счисления.

$$001\ 001\ 011_2 = 113_8$$

8. Чтобы перевести число из двоичной системы в шестнадцатеричную, его нужно разбить на тетрады (четверки цифр), начиная с младшего разряда, в случае необходимости дополнив старшую тетраду нулями, и каждую тетраду заменить соответствующей восьмеричной цифрой (табл. 3).

Пример. Число 1011100011_2 перевести в шестнадцатеричную систему счисления.

$$0010\ 1110\ 0011_2 = 2E3_{16}$$

9. Для перевода восьмеричного числа в двоичное необходимо каждую цифру заменить эквивалентной ей двоичной триадой.

Пример. Число 531_8 перевести в двоичную систему счисления.

$$531_8 = 101011001_2$$

10. Для перевода шестнадцатеричного числа в двоичное необходимо каждую цифру заменить эквивалентной ей двоичной тетрадой.

Пример. Число $EE8_{16}$ перевести в двоичную систему счисления.

$$EE8_{16} = 111011101000_2$$

11. При переходе из восьмеричной системы счисления в шестнадцатеричную и обратно, необходим промежуточный перевод чисел в двоичную систему.

Пример 1. Число FEA_{16} перевести в восьмеричную систему счисления.

$$\begin{aligned} FEA_{16} &= 111111101010_2 \\ 111\ 111\ 101\ 010_2 &= 7752_8 \end{aligned}$$

Пример 2. Число 6635_8 перевести в шестнадцатеричную систему счисления.

$$\begin{aligned} 6635_8 &= 110110011101_2 \\ 1101\ 1001\ 1101_2 &= D9D_{16} \end{aligned}$$

Рассмотрим правило перевода из **восьмеричной СС в двоичную СС**. Для перевода восьмеричного числа в двоичную СС достаточно заменить каждую цифру восьмеричного числа соответствующим трехразрядным двоичным числом. Затем необходимо удалить крайние нули слева, а при наличии дробной части — и крайние нули справа.

Еще одно правило перевода чисел. Для перевода от **шестнадцатеричной СС к двоичной СС** каждая цифра шестнадцатеричного числа заменяется соответствующим четырехразрядным двоичным числом. У двоичного числа удаляются лидирующие нули (крайние слева), а если имеется дробная часть, то и крайние правые нули.

Пример 1. Перевести число 305.4Q из восьмеричной СС в двоичную СС.

Решение.

$$\begin{array}{ccccccc}
 & \text{Переводимое число} & & & \text{Результат} & & \\
 & \underbrace{\hspace{1.5cm}} & & & \underbrace{\hspace{1.5cm}} & & \\
 (3 & 0 & 5. & 4)_8 = & (11000101.1)_2 \\
 \downarrow & \downarrow & \downarrow & \downarrow & & & \\
 011 & 000 & 101. & 100 & & & \\
 \uparrow & & & \uparrow\uparrow & & &
 \end{array}$$

↑

. Заметим, что двоичные числа взяты из табл. 1. (Отмеченные символами « » нули следует отбросить

Таблица 1- Числа от 0 до 15, представленные в различных системах счисления

Системы счисления

Десятичная	Двоичная	Восьмеричная	Шестнадцатеричная
0	000	000	0000
1	001	001	0001
2	010	010	0010
3	011	011	0011
4	100	100	0100
5	101	101	0101
6	110	110	0110

7	111	111	0111
8	1000	1000	1000
9	1001	1001	1001
10	1010	1010	A
11	1011	1011	B
12	1100	1100	C
13	1101	1101	D
14	1110	1110	E
15	1111	1111	F

Рассмотрим еще одно правило. Для перехода от **двоичной СС к восьмеричной** (или **шестнадцатеричной**) СС поступают следующим образом: двигаясь от точки сначала влево, а затем вправо, разбивают двоичное число на группы по три (*четыре*) разряда, дополняя при необходимости пулями крайние левую и правую группы. Затем каждую группу из трех (*четырех*) разрядов заменяют соответствующей восьмеричной (*шестнадцатеричной*) цифрой.

Для перевода **двоичного** числа в **десятичную СС** достаточно представить число в виде полинома, подставить в него известные коэффициенты и вычислить сумму.

Перевод целых чисел из десятичной СС в двоичную, восьмеричную или шестнадцатеричную СС удобно делать с помощью следующего правила:

Для перевода целого числа из S-системы счисления в W-систему счисления нужно последовательно делить это число, а затем получаемые частные на основание W новой СС до тех пор, пока частное не станет меньше W.

При переводе наиболее частой ошибкой является неверная запись результата. Запись двоичного числа следует начинать со старшего значащего разряда (СЗР), а заканчивать записью младшего значащего разряда (МЗР). Следует помнить, что при делении первым получается значение МЗР.

Для перевода правильной дроби из S-системы счисления в СС с основанием W нужно умножить исходную дробь и дробные части получающихся произведений на основание W, представленное в старой S-системе. Целые части получающихся произведений дают последовательность цифр, которая является представлением дроби в W- системе

счисления.

Напомним, что **правильной** называется дробь, числитель которой меньше знаменателя.

Обычно перевод дробей из одной СС в другую производят приближенно. При переводе неправильной дроби переводят отдельно целую и дробную части, руководствуясь соответствующими правилами.

Пример 2. Перевести число 7D2.EH из шестнадцатеричной СС в двоичную СС.

Решение.

$$\begin{array}{ccccccc} \text{Переводимое число} & & & & \text{Результат} & & \\ \hline (7 & D & 2. & E)_{16} & = & (11111010010111)_2 \\ \downarrow & \downarrow & \downarrow & \downarrow & & & \\ 0111 & 1101 & 0010. & 1110 & & & \\ \uparrow & & & \uparrow & & & \end{array}$$

Пример 3. Перевести число 111001100.001B из двоичной СС в восьмеричную СС.

Решение

$$\begin{array}{ccccccc} \text{Переводимое число} & & & & \text{Результат} & & \\ \hline (111 & 001 & 100. & 001)_2 & = & (714.1)_8 \\ \downarrow & \downarrow & \downarrow & \downarrow & & & \\ 7 & 1 & 4. & 1 & & & \end{array}$$

$$\begin{array}{ccccccc} \text{Переводимое число} & & & & \text{Результат} & & \\ \hline (0101 & 1111 & 0001. & 0010)_2 & = & (5F1.2)_{16} \\ \downarrow & \downarrow & \downarrow & \downarrow & & & \\ 5 & F & 1. & 2 & & & \end{array}$$

Пример 4. Перевести число 10111110001.001₂ из двоичной СС в шестнадцатеричную СС.

Решение Для перевода **двоичного** числа в **десятичную** СС достаточно представить число в виде полинома, подставить в него известные коэффициенты и вычислить сумму.

$$\begin{aligned} (11011.11)_2 &= 1 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 + 1 \cdot 2^{-1} + 1 \cdot 2^{-2} = \\ &= 16 + 8 + 0 + 2 + 1 + 0.5 + 0.25 = (27.75)_{10}. \end{aligned}$$

Пример 5. Перевести шестнадцатеричное число 2E5.A₁₆ в десятичную СС. **Решение.**

$$(2E5.A)_{16} = 2 \cdot 16^2 + 14 \cdot 16^1 + 5 \cdot 16^0 + 10 \cdot 16^{-1} = (741.625)_{10}.$$

Перевод целых чисел из десятичной СС в двоичную, восьмеричную или шестнадцатеричную СС удобно делать с помощью следующего правила:

Для перевода целого числа из S-системы счисления в W-систему счисления нужно последовательно делить это число, а затем получаемые частные на основание W новой СС

Двоичная	Восьмеричная	Десятичная	Шестнадцатиричная
A			
	B		
		C	
			D

Вариант	A	B	C	D
1	101010	51	42	2A
2	101110	64	87	57
3	101011	47	204	10D
4	111010	77	155	3B
5	111110	52	41	29
6	1101010	127	52	34
7	1010101	231	63	3F
8	100111	112	37	25
9	1100111	145	63	4C
10	1101000	152	59	2D
11	1101001	164	71	67
12	1100011	133	83	6A
13	100011	116	94	4B
14	101111	104	103	5C
15	1011111	62	117	6D

Практическая работа №2

Тема: Выполнение арифметических операций с многоразрядными двоичными кодированными числами

Цель: Получение практических навыков проведения арифметических операций над числами в различных системах счисления.

Перечень необходимых средств обучения:

Краткие теоретические сведения

Правила выполнения арифметических действий над двоичными числами задаются таблицами сложения, вычитания и умножения (табл. 2), над восьмеричными и шестнадцатеричными числами (Приложения А и Б).

Таблица 2- Арифметические действия над двоичными числами

Таблица 1

Сложение	Вычитание	Умножение
$0 + 0 = 0$	$0 - 0 = 0$	$0 \times 0 = 0$
$0 + 1 = 1$	$1 - 0 = 1$	$0 \times 1 = 0$
$1 + 0 = 1$	$1 - 1 = 0$	$1 \times 0 = 0$
$1 + 1 = 10$	$10 - 1 = 1$	$1 \times 1 = 1$

Правила арифметики во всех позиционных СС аналогичны. В двоичной СС

арифметическое сложение происходит по правилу сложения по модулю два с учетом переноса единицы в старший разряд (см. табл. 2).

В устройствах, реализующих операцию арифметического сложения двоичных чисел, операнды представляют числами определенной разрядности (одинаковой для обоих операндов). При этом неиспользуемые старшие разряды заполняются нулями. Также заполняются нулями младшие разряды дробной части вещественного числа (справа от точки).

Следует заметить, что в реальных ЭВМ чаще всего используются 32-, 64-, 128-разрядные сетки (машинные слова). Однако, для учебных целей при рассмотрении правил выполнения арифметических операций не будем обращать внимание на разрядность операндов (будем использовать разрядность отличающуюся от разрядности реальных ЭВМ).

При сложении вещественных чисел в общем случае перенос осуществляется и из дробной части числа в целую часть.

Рассмотрим правило умножения многоразрядных двоичных чисел.

Умножение двоичных многоразрядных чисел производится путем образования частичных произведений и последующего их суммирования. Каждое частичное произведение равно нулю, если в соответствующем разряде множителя стоит 0, или равно множимому, сдвинутому на соответствующее число разрядов влево, если в разряде множителя стоит 1.

Пример 1. Выполнить операцию арифметического сложения в двоичной системе счисления. *Решение:* Точками показаны переносы.

$$\left[\begin{array}{r} 13 \\ + 7 \\ \hline 20 \end{array} \right]_{10} \rightarrow \left[\begin{array}{r} \overset{\cdot\cdot\cdot\cdot}{01101} \\ + \overset{\cdot\cdot\cdot}{00111} \\ \hline 10100 \end{array} \right]_2$$

Пример 2. Выполнить операцию арифметического сложения двух вещественных чисел в двоичной системе счисления. *Решение:*

$$\left[\begin{array}{r} 55,25 \\ + 19,5 \\ \hline 74,75 \end{array} \right]_{10} \rightarrow \left[\begin{array}{r} \overset{\cdot\cdot\cdot\cdot\cdot\cdot}{0110111.01} \\ + \overset{\cdot\cdot\cdot\cdot}{0010011.10} \\ \hline 1001010.11 \end{array} \right]_2$$

При сложении вещественных чисел в общем случае перенос осуществляется и из дробной части числа в целую часть.

Рассмотрим правило умножения многоразрядных двоичных чисел.

Умножение двоичных многоразрядных чисел производится путем образования частичных произведений и последующего их суммирования. Каждое частичное произведение равно нулю, если в соответствующем разряде множителя стоит 0, или равно множимому, сдвинутому на соответствующее число разрядов влево, если в разряде множителя стоит 1.

Таким образом, **операция умножения** многоразрядных двоичных чисел внутри ЭВМ **сводится к операции сдвига и сложения**. Положение точки, отделяющей целую часть от дробной части, определяется так же, как и при умножении десятичных чисел.

Пример 3. Перемножить в двоичной СС числа 7.5_{10} и 5_{10} . **Решение:**

$$\left[\begin{array}{r} 7,5 \\ \times 5 \\ \hline 37,5 \end{array} \right]_{10} \rightarrow \left[\begin{array}{r} 111.1 \\ \times 101 \\ \hline 1111 \\ + 0000 \\ 1111 \\ \hline 100101.1 \end{array} \right]_2 \rightarrow \begin{array}{r} \text{множимое} \\ \times \text{множитель} \\ \hline \text{1-е част. произв.} \\ + \text{2-е част. произв.} \\ \hline \text{3-е част. произв.} \\ \hline \text{произведение} \end{array}$$

В рассмотренном примере второй разряд множителя равен нулю, по этому второе частичное произведение также равно нулю.

Задания:

2. Выполните арифметические действия с двоичными числами

$E \times F = ?$ и $E + S = ?$

Вариант	E	F	S
1	101010	101	110011
2	101110	11	101011
3	101011	101	101111
4	111010	11	1110101
5	111110	101	110011
6	1101010	11	101011
7	1010101	101	101111
8	100111	11	1110101
9	1100111	101	110011
10	1101000	11	101011
11	1101001	101	101111
12	1100011	11	1110101
13	100011	101	110011
14	101111	11	101011
15	1011111	101	101111

3. Переведите в двоично-десятичный код (ДДК) десятичное число F_{10}

Вариант	F
1	101010
2	101110
3	101011
4	111010
5	111110
6	1101010
7	1010101
8	100111
9	1100111
10	1101000
11	1101001
12	1100011
13	100011
14	101111
15	1011111

Практическая работа №3

Тема: Изучение логических операций.

Цель: Ознакомиться с Булевой алгеброй и логическими элементами . Выполнить проектирование устройства автоматики на логических элементах

Перечень необходимых средств обучения: **Краткие теоретические сведения**

Алгебра логики- это формальный аппарат описания логической стороны процессов, протекающих в цифровых устройствах

Высказывание- некоторое предложение о котором можно сказать истинное оно или ложное.

Z=1 Высказывание истинно.

Z=0 Высказывание ложное.

Логическая переменная (Булева переменная)- такая величина X , которая может принимать только два значения: 0 или 1. Описывает состояние объектов, которые могут находиться в двух состояниях

Логическая функция (функция алгебры логики) - функция $f(x_1, x_2, \dots, x_n)$ принимающая значение «0» или «1» в зависимости от значения логических переменных. Совокупность значений переменных называют **набором**.

Если функция содержит n переменных, то количество наборов $K = 2^n$

Способы задания функции

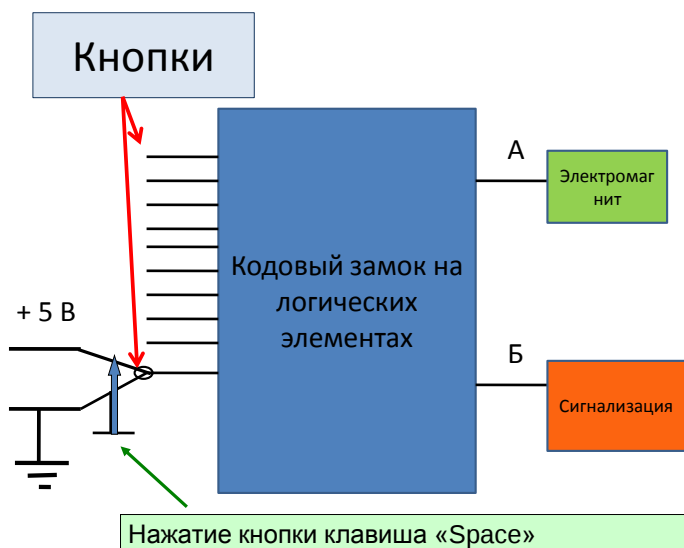
- ✓ **Словесное**
- ✓ **Табличное**
- ✓ **Алгебраическое**
- ✓ **Графическое**
- ✓ **Числовое**
- ✓ Устройства, предназначенные для формирования функций алгебры логики называются логическими или цифровыми устройствами.
- ✓ В электронной технике логической единице соответствует высокий уровень напряжения (5 вольт), логическому нулю – 0 вольт

Задания:

Разработать на основе логических элементов кодовый замок со следующими характеристиками:

Замок содержит 10 кнопок. Одновременное нажатие 4 определенных кнопок (код) вызывает срабатывание электромагнита и замок открывается. Нажатие любой другой кнопки вызывает срабатывание сигнала тревоги и блокирование срабатывания электромагнита (то есть при одновременном нажатии всех кнопок электромагнит не сработает)

При нажатии кнопки на вход устройства поступает сигнал логическая единица. 4 выбранные кнопки (код включения) обеспечивают на выходе А сигнал логической «1» и срабатывание замка. Любое нажатие других кнопок вызывает срабатывание сигнализации (выход Б) и блокировку замка (сигнал 0 на вых А).



Лабораторная работа №1

Тема: Исследование логических микросхем.

Цель: Исследование основных логических элементов и простейших комбинационных устройств на их основе; составление и экспериментальная проверка таблиц истинности для каждого устройства.

Перечень необходимых средств обучения:

Стенд ОАВТ, набор плат

Краткие теоретические сведения

В большинстве современных ЭВМ и цифровых устройств различного назначения информация обрабатывается с помощью двоичного кода, когда информационные сигналы (входные и выходные напряжения) могут принимать только два значения: 1 – “высокий” уровень или 0 – “низкий” уровень. Операции по обработке двоичной информации выполняют логические элементы – электронные схемы, выполняющие простейшие логические операции.

Все возможные логические функции любого числа логических переменных можно образовать с помощью трех основных логических операций: **логического отрицания** (инверсии, операции НЕ), **логического сложения** (дизъюнкции, операции ИЛИ) и **логического умножения** (конъюнкции, операции И). Таблицы истинности, условные графические обозначения элементов, выполняющих эти функции, представлены на рис. 1.

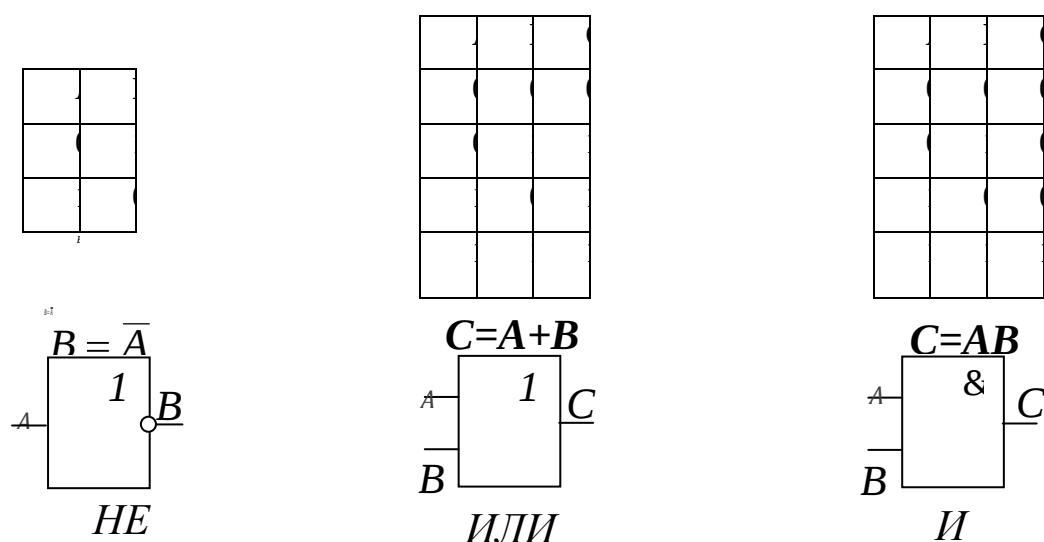


Рис. 1

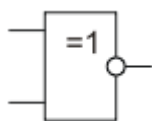
При помощи рассмотренных логических элементов в соответствии с правилами и законами алгебры логики могут быть реализованы сложные логические функции.

С помощью простейших логических элементов можно реализовать более сложные логические функции, такие как **ИСКЛЮЧАЮЩЕЕ ИЛИ**, **НЕРАВНОЗНАЧНОСТЬ**, **ЧЕТНОСТЬ**.

Условно-графическое отображение и таблицы истинности этих элементов приведены ниже.

Эквивалентность (равнозначность), ИСКЛЮЧАЮЩЕЕ_ИЛИ-

НЕ



ИСКЛ-ИЛИ-НЕ

$A \quad B \quad A \leftrightarrow B$

0 0 1

0 1 0

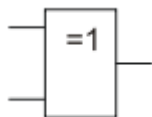
1 0 0

1 1 1

Мнемоническое правило эквивалентности с любым количеством входов звучит так: На выходе будет:

- «1» тогда и только тогда, когда на входе действует **четное** количество,
- «0» тогда и только тогда, когда на входе действует **нечетное** количество

Сложение (сумма) по модулю 2 (Исключающее_ИЛИ, неравнозначность). Инверсия равнозначности.



ИСКЛ-ИЛИ

В англоязычной литературе [XOR](#).

$A \quad B \quad f(AB)$

0 0 0

0 1 1

1 0 1

1 1 0

Мнемоническое правило для суммы по модулю 2 с любым количеством входов звучит так: На выходе будет:

- «1» тогда и только тогда, когда на входе действует **нечётное** количество ,
- «0» тогда и только тогда, когда на входе действует **чётное** количество

В данной лабораторной работе исследуются логические элементы реализованные на микросхемах серии 155 с помощью стенда ОАВТ.

Указанные логические операции выполняются логическими интегральными микросхемами. Логическому нулю соответствует низкий уровень напряжения. Логической единице – высокий уровень напряжения. Для предотвращения ложных срабатываний логическому нулю и логической единице соответствует не конкретное напряжение, а диапазон напряжений. Значение этих диапазонов зависит от технологии изготовления микросхем и напряжения питания, рис 1.

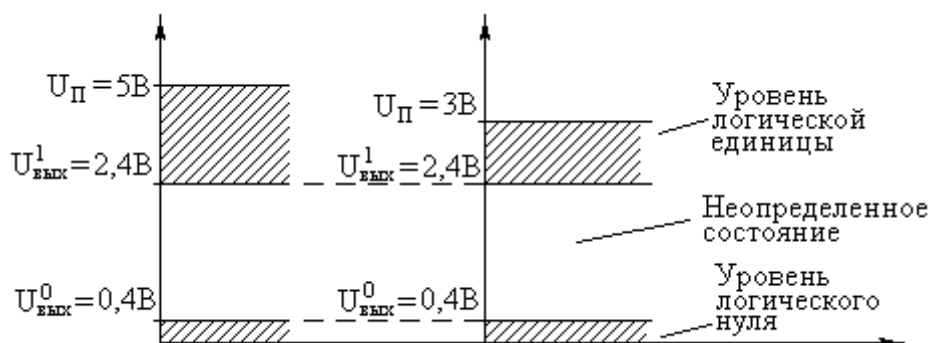
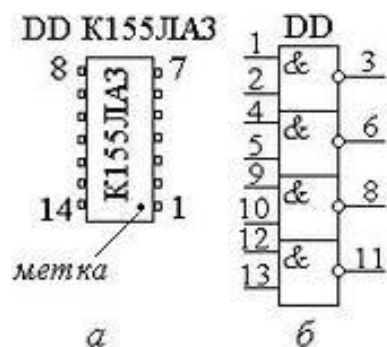


Рис 1 Уровни «1» и «0» для различных серий микросхем

Логические ИМС объединяют в серии. В основе серии лежит базовый элемент **И- НЕ** либо **ИЛИ – НЕ**. От характеристик элемента зависят свойства серии. Российские стандарты логических ИМС: 155 серия; 133 серия; 176 серия, 555 серия и др. В лабораторной работе используются микросхемы 155ЛА3, 155ЛЕ1. Маркировка и технические данные микросхем приведены ниже.

Микросхема 155 ЛА3



Вывод 7 общий (земля)
Вывод 14 + 5 вольт

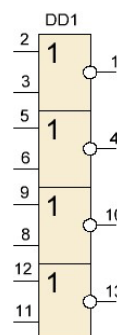


1	Номинальное напряжение питания	5 В ±5 %
2	Выходное напряжение низкого уровня	не более 0,4 В
3	Выходное напряжение высокого уровня	не менее 2,4 В
4	Напряжение на антизвонном диоде	не менее -1,5 В
5	Входной ток низкого уровня	не более -1,6 мА
6	Входной ток высокого уровня	не более 0,04 мА
7	Входной пробивной ток	не более 1 мА
8	Ток короткого замыкания	-18...-55 мА
9	Ток потребления при низком уровне выходного напряжения	не более 22 мА

10	Ток потребления при высоком уровне выходного напряжения	не более 8 мА
11	Потребляемая статическая мощность на один логический элемент	не более 19,7 мВт
12	Время задержки распространения при включении	не более 15 нс
13	Время задержки распространения при выключении	не более 22 нс

Микросхема 155ЛЕ1

1	Номинальное напряжение питания	5 В $\pm 5\%$
2	Выходное напряжение низкого уровня	не более 0,4 В
3	Выходное напряжение высокого уровня	не менее 2,4 В
4	Входной ток низкого уровня	не более -1,6 мА
5	Входной ток высокого уровня	не более 0,04 мА
6	Входной пробивной ток	не более 1 мА
7	Ток потребления при низком уровне выходного напряжения	не более 27 мА
8	Ток потребления при высоком уровне выходного напряжения	не более 16 мА
9	Потребляемая статическая мощность на один логический элемент при низком уровне выходного напряжения	не более 36 мВт
10	Потребляемая статическая мощность на один логический элемент при высоком уровне выходного напряжения	не более 21 мВт



Описание стенда ОАВТ

Фотография и принадлежности стенда для исследования логических микросхем приведены на рис 1

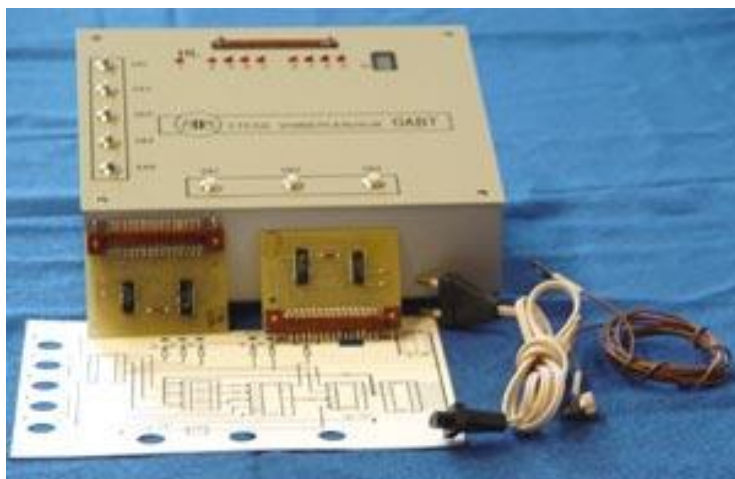


Рис 1 Стенд ОАВТ с принадлежностями

Стенд имеет встроенный источник питания, платы и карты для работы с платами. Переключатели и кнопки позволяют подавать на вход логических элементов высокий уровень напряжения логическая «1» или низкий уровень напряжения логический «0»

На плате расположены цифровые комбинационные схемы малой степени интеграции *K155ЛЕ1*, *K155ЛА3* и другие

Цифровые микросхемы серии *K155* изготавливаются по стандартной технологии биполярных ИС транзисторно-транзисторной логики (ТТЛ). Все микросхемы серии *K155* имеют напряжение питания $5\text{ В} \pm 5\%$, выходной уровень логического 0 не более $0,4\text{ В}$, выходной уровень логической 1 не менее $2,4\text{ В}$, типовую нагрузочную способность – 10 (то есть к выходу микросхемы может быть подключено до 10 других логических микросхем этой серии).

Микросхема *K155ЛЕ1* выполняет функцию *ИЛИ-НЕ* (4 *ИЛИ-НЕ*), *K155ЛА3* – функцию *И-НЕ* (4 *И-НЕ*).

Задания:

1. Вставьте карту П1 в разъем стенда. Логические элементы и устройства, расположенные на плате П I исследуются при подаче на их входы логических сигналов (0 или 1) от тумблеров SA1 –SA5. Уровень сигнала, подаваемого от соответствующего тумблера, определяется по положению подвижного контакта (вверху единица, внизу ноль), уровень выходного сигнала – по свечению индикаторов HL1 – HL9.
2. Установите карту П1.1 на стенд. Включите стенд (тумблер на задней панели). Переключая сигналы на входе элемента («0» и «1») составьте таблицу истинности и определите элементы обозначенные на карте буквой « λ ». Примечание: если обозначение элементов начинается с одинаковой цифры (например D1.1 и D 1.2 то это означает, что D1.2 и D1.1 одинаковые элементы одной микросхемы D1
3. Замените карту 1.1 на карту 1.2. Повторите операции п. 1
4. Выполните операции с той же платой и картами 1.3, 1.4, 1.5, 1.6,
5. Выполните операции с картой 1.7 «Трехразрядное устройство проверки на четность». Индикатор HL9 должен светиться при четной сумме сигналов SA1, SA2, SA3. Проверьте совпадение теоретических и практических результатов.
6. Выполните операции с картой 1.8 «Устройство сравнения двух двухразрядных чисел»

X1 (SA1) X2 (SA4) и Z1 (SA2) Z2 (SA5)

При выполнении условия $X1 X2 = Z1 Z2$ на выходе появляется единица (светится HL1)

7. Подготовьте отчет о проделанной работе. Отчет должен содержать номер карты, таблицы истинности, УГО логических элементов и выводы

Контрольные вопросы

1. Приведите таблицу истинности элемента 2И-НЕ
2. Приведите таблицу истинности элемента 2ИЛИ-НЕ
3. Каким значениям электрического напряжения соответствует логическая единица для микросхем 155 серии? Логический ноль?
4. Что такое комбинационное устройство?
5. Что такое серия ИМС ?

Лабораторная работа №2

Тема: Исследование триггеров

Цель: Исследование RS, T, D триггеров. Ознакомление с элементной базой ТТЛ, выработка навыков составления и исследования схем с применением триггеров, экспериментальная проверка таблиц истинности для каждого устройства.

Перечень необходимых средств обучения:

Стенд ОАВТ, платы триггеров, карты для стенда

Краткие теоретические сведения

Триггер- электронное устройство имеющее два устойчивых состояния.

В одном состоянии на выходе триггера присутствует высокий потенциал («1»), в другом – низкий («0»).

Переход триггера из одного состояния в другое происходит лавинообразно, с приходом переключающего (запускающего) сигнала.

Триггер применяется :

В качестве элемента памяти для хранения одноразрядного числа. В промежутке между переключающими импульсами состояние триггера не изменяется. Триггер «запоминает» последний поступивший сигнал.

Для формирования прямоугольных импульсов из напряжения другой формы. Триггер лавинообразно переходит из одного состояния в другое. Запускающие импульсы могут быть синусоидальными, а на выходе триггера сигнал получается прямоугольной формы.

В качестве делителя частоты на 2. Пара переключающих импульсов на входе вызывает формирование одного импульса на выходе.

Триггер имеет два выхода: прямой Q и инверсный $\overline{Q}$

Состояние триггера оценивается по уровню сигнала на прямом выходе («0» или «1»).

Если на прямом выходе «1» то на инверсном «0» и наоборот

Триггеры имеют различные типы входов:

R (от англ. **Reset**) — ***раздельный вход установки в состояние 0;***
S (от англ. **Set**) — ***раздельный вход установки в состояние 1;***
K — ***вход установки универсального триггера в состояние 0;***
J — ***вход установки универсального триггера в состояние 1;***
T — ***счетный вход;***
D (от англ. **Delay**) — ***информационный вход установки триггера в состояние, соответствующее логическому уровню на этом входе;***
C — ***управляющий (синхронизирующий) вход.***

Наименование триггера определяется типом его входов: например **RS триггер**

В лабораторной работе исследуются: RS триггер, D- триггер, Т- триггер. Платы с триггерами, реализованными на различных логических элементах подключаются к стенду ОАВТ

Стенд имеет встроенный источник питания, элементы индикации, коммутации и карты для работы с платами. Переключатели и кнопки позволяют подавать на вход триггеров высокий уровень напряжения логическая «1» (тумблер включен, положение тумблера вверх) или низкий уровень напряжения логический «0» (тумблер замыкает вход на землю). Аналогичную функцию выполняют кнопки. Высокий уровень сигнала индицируется светодиодами.

На плате расположены триггеры, собранные на логических элементах И-НЕ , ИЛИ-НЕ и в интегральном исполнении на ИМС K155TM2

Задания:

1. Установите в разъем плату II. Установите на пульт плату II – 1 « RS триггер». Подавайте на входы триггера R и S комбинации сигналов «0» и «1» с помощью кнопок SB2 и SB3. Составьте таблицу истинности RS триггера. Входной сигнал «1» индицируется свечением светодиодов HL1 или HL2. Выходные сигналы Q и $\overline{Q}$ индицируются светодиодами HL3 HL4. Зарисуйте схему RS триггера. Данные об используемых микросхемах используйте из лабораторной работы 2.
2. Повторите измерения для карт II – 2 и II -3. На этих платах RS триггер реализован на других элементах.
3. Установите в пульт карту II – 4 «D- триггер». Комбинируя сигналы приходящие на информационный вход D и на вход синхронизации C, составьте таблицу истинности для D-триггера. Зарисуйте электрическую схему D-триггера.
4. Выключите пульт ОАВТ. Вставьте в пульт карту II – 7 Счетный триггер. Соедините проводником на плате контакты Y1 и X2. Кнопкой SB1 подавайте импульсы на вход триггера. Задержите отпускание кнопки. Определите, когда происходит изменение сигнала на выходе счетного триггера при замыкании кнопки (передний фронт импульса) или при размыкании кнопки (задний фронт импульса). Зарисуйте электрическую схему Т- триггера.
5. Оформите отчет о проделанной работе

Контрольные вопросы

1. Что такое триггер?
2. Характеризуйте прямой и инверсный выходы триггера
3. Характеризуйте RS триггер
4. Характеризуйте D- триггер
5. Характеризуйте Т-триггер
6. Характеризуйте синхронные и асинхронные триггеры

Лабораторная работа №3

Тема: Исследование счетчиков электрических импульсов

Цель: Исследование счетчиков электрических импульсов. Ознакомление с элементной базой счетчиков, выработка навыков составления и исследования схем счетчиков, экспериментальная проверка таблиц состояния для каждого устройства

Перечень необходимых средств обучения:

Стенд ОАВТ, платы счетчиков, карты для стенда

Краткие теоретические сведения

Счётчик –предназначен для счета поступающих на его вход импульсов и сохранения информации об их количестве. Счетчик состоит из запоминающих ячеек –триггеров.

- Счётчики могут строиться на двухступенчатых D-триггерах, Т-триггерах и JK-триггерах.
- Ячейки счетчика соединяются между собой таким образом, чтобы каждому числу импульсов соответствовали единичные состояния определенных ячеек
- Совокупность нулей и единиц на выходах **n** ячеек счетчика представляет собой **n** разрядное двоичное число, которое определяет количество поступивших на вход импульсов
- Ячейки счетчика называют его разрядами
- Каждый разряд счетчика может находиться в двух состояниях.
- **модуль счёта (M)** — максимальное число единичных сигналов, которое может быть сосчитано счётчиком. Счётчики обозначают через СТ (от англ. counter).
- **Операция инкремента** - изменение содержимого счетчика на единицу

Классификация счетчиков

- по модулю счёта:
 - двоично-десятичные (декада);
 - Двоичные ($M=2^n$);
 - с произвольным модулем счёта (используется двоичный код, но значение модуля устанавливается за счет обратных связей);
- по направлению счёта:
 - суммирующие;
 - вычитающие;
 - Реверсивные (могут изменять направление счета)

Электрическая схема суммирующего счетчика на Т триггерах приведена на рис 1

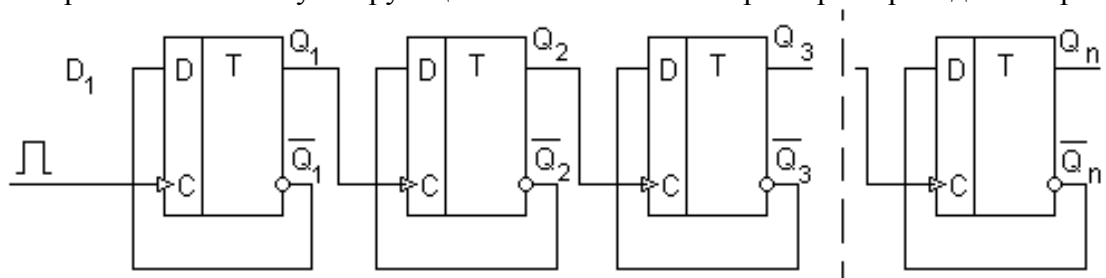


Рис1 Электрическая схема суммирующего счетчика на Т- триггерах

Состояние триггеров суммирующего счетчика приведено в таблице 1

Таблица1 Состояние триггеров суммирующего счетчика

Число поступивших импульсов	Состояние триггеров			
	Q_4	Q_3	Q_2	Q_1
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1

Вычитающий счетчик

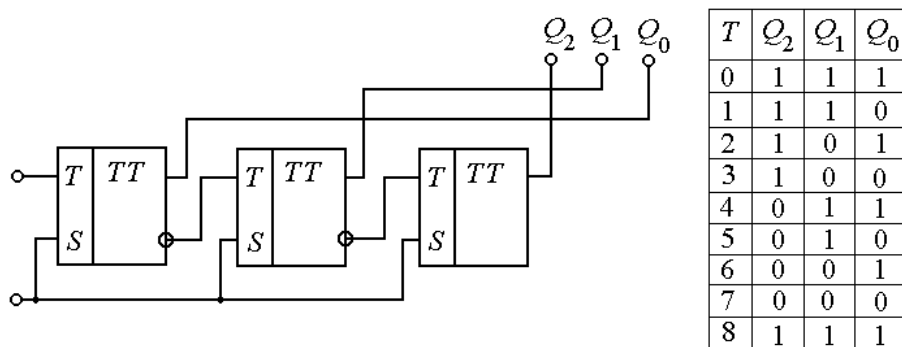
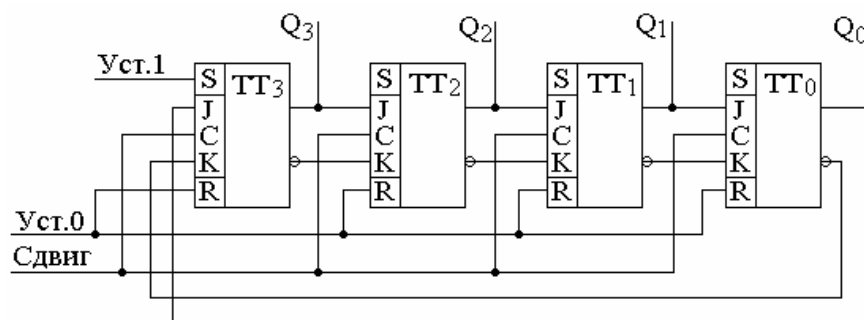


Рис 2 Вычитающий счетчик. Соединение триггеров выполняется по инверсному выходу.

Кольцевой счетчик

Если в счетчике объединить выход последнего триггера с входом первого триггера, то получится кольцевой счетчик. Каждый тактовый импульс перемещает установленное число («1») на одно положение вправо по кольцу. Установка «1» выполняется с помощью входа «S» триггера Q_3 .



Задания:

1. Установите в разъём плату ПЗ (карта III – 1). Соберите схему кольцевого счетчика и исследуйте его работу. Кольцевой четырехразрядный счетчик реализован на базе микросхемы универсального регистра 155 ИР1. Универсальный регистр переводится в последовательный режим при сигнале $V=0$. Поэтому тумблер SA 5 должен находиться в положении «0». Входная информация (вход D) подается кнопкой SB 2, без индикации. Кнопка «отпущена» - $D=1$, кнопка «нажата» - $D=0$. Выходная информация выводится на четырехразрядный дисплей HL6 – HL9.

Для перевода регистра в режим кольцевого счетчика записывают в него число 0001. Для этого не нажимая SB2 («1» на информационном входе) нажимают кнопку SB1 (поступление тактового импульса). Затем соединяют внешней перемычкой штырьки на плате X и Y, что соответствует соединению выхода счетчика со входом. После этого с подачей каждого тактового импульса (нажимая на кнопку SB1) единица будет последовательно смещаться по разрядам счетчика. Заполните таблицу 1. В таблице в столбце SB отмечают количество нажатий кнопки SB1, соответствующего поступлению импульса в счетчик. В столбцах HL фиксируют состояние индикаторов HL1 - HL4. Горящий светодиод соответствует «1», погашенный – «0»

Таблица1

SB 1	HL 1	HL2	HL3	HL4
0				
1				
2				
3				
4				
5				
6				
7				
8				
9				

2. Исследовать схему суммирующего счетчика на четырех триггерах.

Выключить пульт ОАВТ. Снять плату ПЗ. Установить плату П5 (Карта V-1). Включить пульт ОАВТ. Установить тумблер SA 5 в положение «1». Произвести сброс счетчика в нулевое положение кнопкой SB 3.

Счетные импульсы формируемые по нажатию кнопки SB 1 подаются на вход суммирования счетчика. Коэффициент пересчета равен $2^4=16$.

Последовательно нажимая кнопку SB1. Заполните таблицу 2

Таблица 2

SB 1	HL 1	HL2	HL3	HL4
0				
1				
2				
3				
4				
5				
6				
7				
8				
9				
10				
11				
12				
13				
14				
15				
16				

3.Измените коэффициент пересчета счетчика. Для этого соедините выход У4 с входом Х1 счетчика. Выполните операции п.2 и заполните таблицу 3. Определите коэффициент пересчета счетчика. Объясните полученный результат.

Таблица 3

SB 1	HL 1	HL2	HL3	HL4
0				
1				
2				
3				
4				
5				
6				
7				
8				
9				
10				
11				
12				
13				
14				
15				
16				

4. Исследуйте работу вычитающего счетчика. Выключите ОАВТ. Снимите перемычку с платы. Установите карту V-2. Включите ОАВТ. Переключите тумблер SA 5 в положение «0» Введите в счетчик предустановленное число. Для этого установите с помощью

тумблеров SA1 – SA4 число и нажмите кнопку SB2. Реализуйте режим обратного счета. Нажимайте кнопку SB1, что соответствует подаче одного импульса на вход счетчика и запишите полученные результаты в таблицу 4, аналогичную таблице 3

5. Оформите отчет о проделанной работе

Контрольные вопросы

1. Что такое счетчик?
2. Что такое модуль (коэффициент пересчета) ?
3. Объясните по предложенной электрической схеме принцип работы суммирующего счетчика на триггерах
4. Объясните по предложенной электрической схеме принцип работы принципа работы реверсивного счетчика.
5. Что такое кольцевой счетчик?
6. Объясните по предложенной электрической схеме принцип работы принципа работы счетчика с предустановкой числа.
7. Объясните по предложенной электрической схеме принцип работы принципа работы счетчика с изменяемым модулем пересчета.

Лабораторная работа №4

Тема: Исследование регистров

Цель: Исследование регистров. Ознакомление с элементной базой микросхем регистров, выработка навыков составления и исследования схем с применением регистров.

Перечень необходимых средств обучения:

Стенд ОАВТ, платы с микросхемами регистров, карты для стенда

Краткие теоретические сведения

Регистром называется последовательностное устройство, предназначенное для записи, кратковременного хранения и последующего считывания информации, представленной в виде n - разрядной кодовой комбинации (двоичного числа, слова), например 10100011

Регистры могут также использоваться в качестве счетчиков и делителей частоты, узлов временной задержки импульсов.

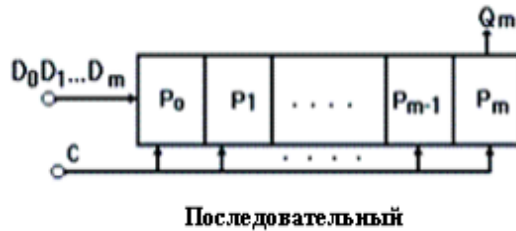
- ✓ Состоит из запоминающих элементов- триггеров
- ✓ В каждом триггере хранится цифра определенного разряда числа
- ✓ Может выполнять сдвиг принятого слова вправо и влево
- ✓ Может преобразовать двоичный код из прямого в обратный (замена 0 на 1 и 0 на 1)
- ✓ Выполнять логические сложение и умножение

Типы регистров (по способу ввода-вывода информации)

- Параллельные. Ввод- вывод слова выполняется параллельно, одновременно во все разряды.
- Последовательные (регистры сдвига). Разряды числа вводятся и выводятся последовательно
- Комбинированные

Последовательный регистр

В последовательном регистре запись и чтение информации выполняется по одному биту. Применение последовательного кода связано с необходимостью передачи большого количества двоичной информации по ограниченному количеству соединительных линий. При параллельной передаче разрядов требуется большое количество соединительных проводников. Если двоичные разряды последовательно бит за битом передавать по одному проводнику, то можно значительно сократить размеры соединительных линий на плате (и размеры корпусов микросхем).



Принципиальная схема последовательного регистра, собранного на основе D-триггеров приведена на рисунке 1

Рассмотрим работу этого регистра.

Можно предположить, что в начале все триггеры регистра находятся в состоянии логического нуля, т.е. $Q_0=0$, $Q_1=0$, $Q_2=0$, $Q_3=0$. Если на входе D-триггера T1 логический 0, то поступление синхрoимпульсов на входы «С» триггеров не меняет их состояния.

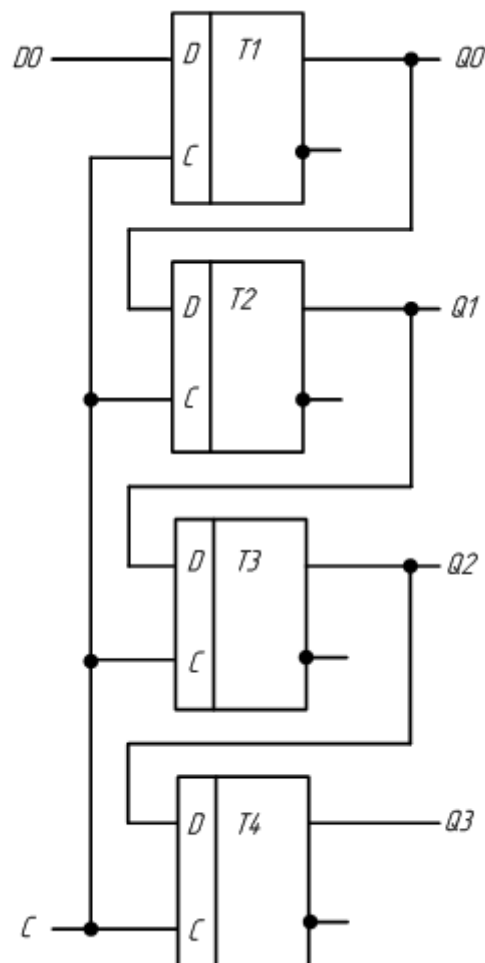


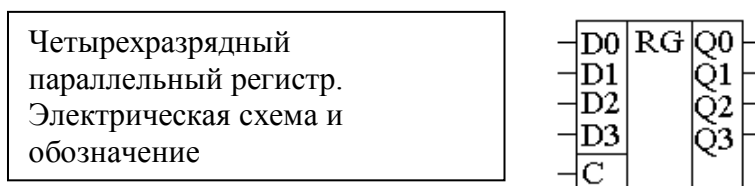
Рисунок 2 Схема последовательного регистра

Как следует из рисунка 2, синхроимпульсы поступают на соответствующие входы всех триггеров регистра одновременно и записывают в них то, что имеет место на их информационных входах. На информационных входах триггеров T2, T3, T4 – уровни логического «0», т.к. информационные входы последующих триггеров соединены с выходами предыдущих триггеров, находящихся в состоянии логического «0», а на вход «D» первого триггера, по условию примера, подается «0» из внешнего источника информации. При подаче на вход «D» первого триггера «1», с приходом первого синхроимпульса, в этот триггер запишется «1», а в остальные триггеры – «0», т.к. к моменту поступления фронта синхроимпульса на выходе триггера T1 ещё присутствовал логический «0». Таким образом, в триггер T1 записывается та информация (тот бит), которая была на его входе «D» в момент поступления фронта синхроимпульса и т.д.

При поступлении второго синхроимпульса логическая «1» с выхода первого триггера, запишется во второй триггер, и в результате происходит сдвиг первоначально записанной «1» с триггера T1 в триггер T2, из триггера T2 в триггер T3 и т.д. Таким образом, производится последовательный сдвиг поступающей на вход регистра информации (в последовательном коде) на один разряд вправо в каждом такте синхроимпульсов.

После поступления четырёх синхроимпульсов регистр оказывается полностью заполненным разрядами числа, вводимого через последовательный ввод «D». В течение следующих четырёх синхроимпульсов производится последовательный поразрядный вывод из регистра записанного числа, после чего регистр оказывается полностью очищенным (регистр окажется полностью очищенным только при условии подачи на его вход уровня «0» в режиме вывода записанного числа).

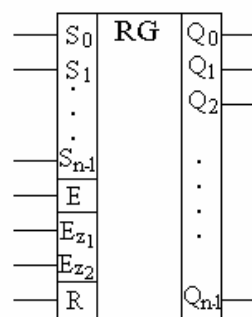
Параллельный регистр



При записи информации в параллельный регистр все биты (двоичные разряды) должны быть записаны одновременно. Поэтому все тактовые входы триггеров, входящих в состав регистра, объединяются параллельно.

Регистры памяти (Регистры хранения)

- Хранят двоичную информацию небольшого объема в течение короткого промежутка времени. Эти регистры представляют собой набор синхронных триггеров, каждый из которых хранит один разряд двоичного числа. Ввод (запись, загрузка) и вывод (считывание) информации производится одновременно во всех разрядах параллельным кодом. Запись обеспечивается тактовым импульсом.
 - С приходом очередного тактового импульса происходит обновление записанной информации.
 - Принцип построения регистра хранения на **RS** –триггерах приведен на рисунке 1
- Каждый триггер служит для записи и хранения одного двоичного разряда



Разряды слова **A** подводятся к **S** – входам триггеров через схемы совпадения "И", управляемые по шине "ЗАПИСЬ" единичными импульсами **E**. При сигнале **E=0** обеспечивается режим хранения записанной информации, то есть новая информация на установочные входы **S** не подается.

Для считывания информации в требуемом коде необходимо на соответствующую шину подать единичный импульс. По окончании считывания на RS – триггеры регистра подается сигнал сброса в "0", после чего регистр готов принимать и хранить следующее слово.

Регистры могут использоваться и для операции сдвига данных с целью преобразования двоичного последовательного кода в параллельный и наоборот. Суть сдвига состоит в том, что по сигналу синхроимпульса происходит одновременная перезапись содержимого каждого триггера в соседний триггер. При этом не меняется само двоичное слово (число), записанное в регистре, оно лишь сдвигается на один разряд и только содержимое последнего триггера $ТТ_{n-1}$ пропадает из регистра, а на вход первого $ТТ_0$ поступает новый бит.

Регистры сдвига реализуются, как правило, на синхронных ***D*** и ***JK*** - триггерах со статическим или динамическим управлением. Такие регистры имеют информационный вход, вход тактовых импульсов и установочный вход. Выходы в регистре сдвига могут быть с триггера каждого разряда для считывания информации параллельным кодом или только с последнего младшего разряда для считывания информации последовательным кодом.

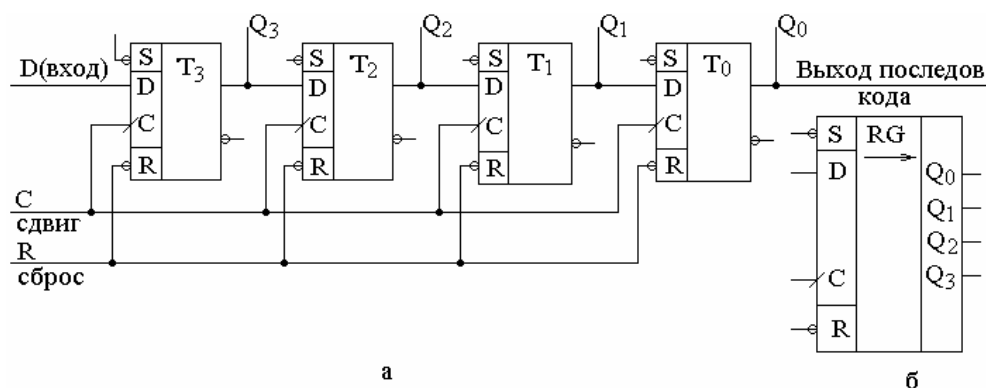


Рис 2 Схема четырехразрядного регистра сдвига вправо на синхронных D -триггерах с прямым динамическим входом С . Стрелка под обозначением регистра RG в основном поле показывает направление сдвига информации.

В схеме прямой выход **Q** каждого предыдущего (левого) триггера соединен со входом **D** последующего триггера.

Перед записью информации регистр устанавливается в нулевое состояние подачей положительного импульса по шине «СБРОС».

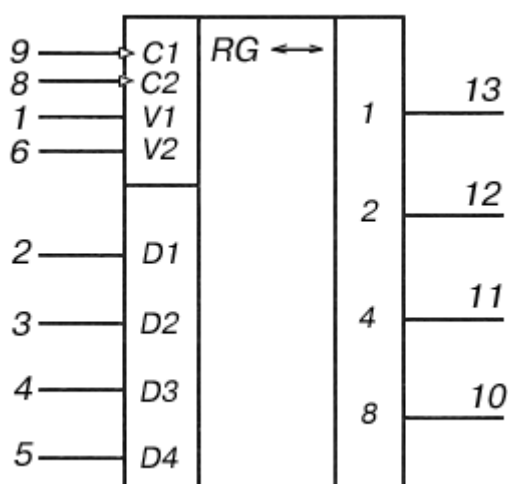
В лабораторной работе проводится исследование параллельного ,последовательного и универсального регистров реализованного на микросхеме K155ИР1. Электрическая схема и характеристики микросхемы приведены ниже

Микросхемы K155ИР1, КМ155ИР1 (7495) — представляют четырехразрядный, сдвиговый регистр. Он имеет последовательный вход данных S1 (вывод 1), четыре параллельных входа DO — D3 (выводы 2 — 5), а также четыре выхода Q0 — Q3 (выводы 13 — 10) от каждого из триггеров . Регистр имеет два тактовых входа С1 и С2. От любого из пяти входов данных код поступит на выходы синхронно с отрицательным перепадом, поданным на выбранный тактовый вход.

Вход разрешения параллельной загрузки PE служит для выбора режима работы регистра K155ИР1, КМ155ИР1 (7495). Если на вход PE дается напряжение высокого уровня, разрешается работа тактовому входу С2. В момент прихода на этот вход отрицательного перепада тактового импульса в регистр загружаются данные от параллельных входов DO — D3.

Если на вход PE подано напряжение низкого уровня, разрешается работа тактовому входу С1. Отрицательные фронты последовательности тактовых импульсов сдвигают данные от последовательного входа S1 на выход Q0, затем на Q1, Q2 и Q3, (вправо). Сдвиг данных по регистру влево получится, если соединить выход Q3 и вход D2, Q2 и D1, Q1 и D0, Регистр надо перевести в параллельный режим, подав на вход PE напряжение высокого уровня. Напряжение на входе регистра K155ИР1, КМ155ИР1 (7495) PE можно менять только если на обоих тактовых входах уровни низкие. Однако если на входе С1 напряжение низкого уровня, перемена сигнала на входе PE от низкого уровня к высокому не меняет состояния выходов.

Обычный вариант микросхемы K155ИР1, КМ155ИР1 (7495) имеет ток потребления 63 мА, Максимальная,тактовая частота 25 МГц, Возможные режимы работы регистра K155ИР1, КМ155ИР1 (7495) следует выбирать по таблице.



- 1 - информационный вход V1;
 2 - вход первого разряда D1;
 3 - вход второго разряда D2;
 4 - вход третьего разряда D3;
 5 - вход четвертого разряда D4;
 6 - вход выбора режима V2;
 7 - общий;
 8 - вход синхронизации C2;
 9 - вход синхронизации C1;
 10 - выход четвертого разряда;
 11 - выход третьего разряда;
 12 - выход второго разряда;
 13 - выход первого разряда;
 14 - напряжение питания;

Задания:

1. Исследуйте работу последовательного регистра. В выключенном состоянии стенда ОАВТ установите в разъем платы ПЗ. Установите карту (трафарет) III-1. Микросхема универсального регистра К155ИР1 переводится в последовательный режим при сигнале управления V=0, поэтому при работе с этой картой тумблер SA 5 должен быть в положении «0». Входная информация подается кнопкой SB2 (без индикации). Кнопка «отпущена» D=1, кнопка «Нажата» D=0. Тактовый импульс (синхроимпульс) задается кнопкой SB 1.

Запишите в последовательный регистр число 1101. Для этого последовательно подавайте на вход нужное число (0 или 1) и посылайте синхроимпульс.

Исследуйте смещение заданного числа в регистре. Для этого при введенном числе 1101 подавайте на вход D «0» и посылайте синхроимпульс. Результаты запишите в таблицу 1

Таблица 1

Тактовый импульс	HL 1	HL 2	HL 3	HL 4
0	1	1	0	1
1				
2				
3				
4				

Объясните полученные результаты

2. Исследуйте работу микросхемы универсального регистра К155ИР1 в режиме параллельного регистра. Для этого установите на стенд ОАВТ карту –шаблон III-2.

Переключите тумблер SA 5 в положение «1». Это соответствует установке сигнала «1» на вход V.

Входная информация индицируется светодиодами HL1 – HL4, выходная HL5 – HL8

Выполните запись числа 1001 в параллельный регистр. Наберите тумблерами SA1 - SA4 это четырехразрядное число. Кнопкой SB1 направьте синхроимпульс на вход

микросхемы. Зафиксируйте число, которое фиксируется на выходах микросхемы. Наберите другое, произвольное число и повторите операцию. Объясните полученные результаты.

Контрольные вопросы

1. Что такое регистр? Для чего он предназначен?
2. Характеризуйте регистр сдвига (последовательный)
3. Характеризуйте параллельный регистр
4. Объясните принцип работы регистра хранения информации на RS триггерах. Электрическая схема прилагается.
5. Объясните принцип работы регистра хранения информации на D триггерах. Электрическая схема прилагается
6. Объясните назначение входов EZ1, EZ2, V в микросхемах регистров.

Лабораторная работа №5

Тема: Исследование мультиплексора и демultipлексора

Цель: Исследование дешифратора – демultipлексора и мультиплексора. Ознакомление с элементной базой микросхем, выработка навыков составления и исследования схем с применением комбинационных устройств.

Перечень необходимых средств обучения:

стенд для изучения работы цифровых микросхем ОАВТ, плата,
технологическая карта

Краткие теоретические сведения

В лабораторной работе исследуется работа дешифратора и демultipлексора на интегральной микросхеме (ИС) К155ИД4, коммутатора на ИС К155КП5 и преобразователя кодов на ИС К155РЕ3.

Дешифратор – это устройство, предназначенное для преобразования двоичного кода в напряжение логической единицы (логического нуля) на том выходе, номер которого совпадает со значением двоичного кода на входе. При N входах в *полном дешифраторе* имеется 2^N выходов, т.е. для каждой комбинации входных сигналов имеется соответствующий выход. Дешифратор, у которого при N входах число выходов меньше 2^N , называется *неполным*. Другое название дешифратора - *декодер*.

Вид микросхемы дешифратора приведен на рис 1

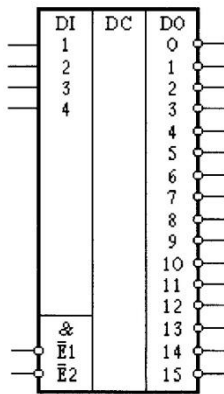
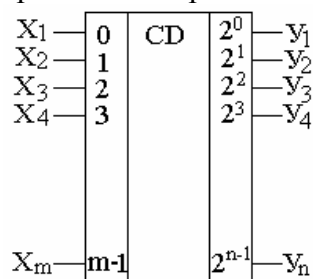


Рис 1 Микросхема дешифратора

Шифратором называется устройство с m входами и n выходами, преобразующее сигнал «1» на одном из входов в n -элементный параллельный код на выходах.

Шифратор выполняет функцию обратную дешифратору. Микросхема шифратора приведена на рис 2



Одно из практических применений шифратора и дешифратора в калькуляторе приведено на рис 3

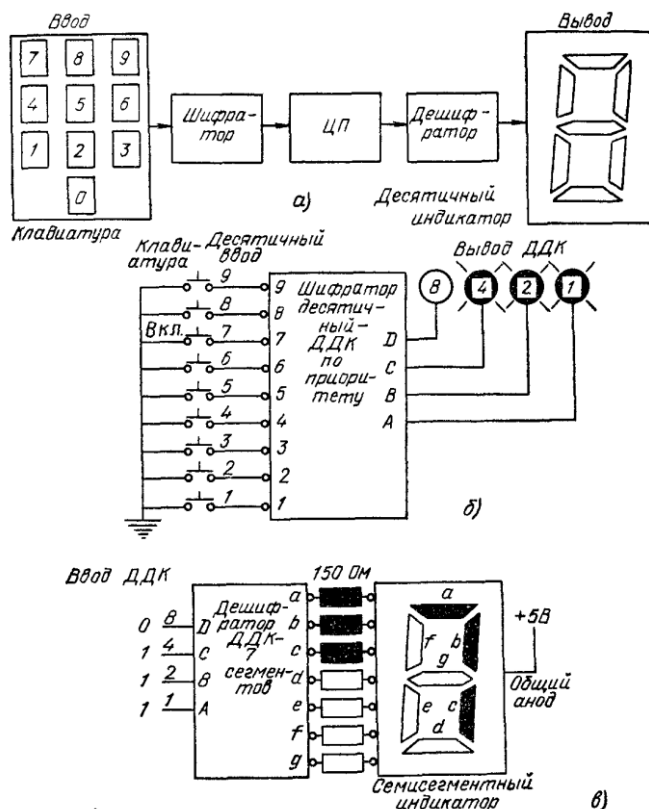
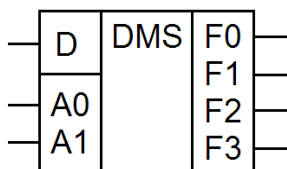


Рис 3 Применение шифратора и дешифратора в калькуляторе.

Демультимплексор — это логическое устройство, предназначенное для переключения сигнала с одного информационного входа на один из информационных выходов. На схемах демультимплексоры обозначают через **DMX** или **DMS**.



Мультиплексор- устройство обратное по назначению демультимплексору. В соответствии с сигналами на адресном входе соединяется один из входов с выходом мультиплексора.

Аналоговые и цифровые мультиплексоры значительно различаются по принципу работы. Первые электрически соединяют выбранный вход с выходом (при этом сопротивление между ними невелико — порядка единиц/десятков ом). Вторые же не образуют прямого электрического соединения между выбранным входом и выходом, а лишь «копируют» на выход логический уровень ('0' или '1') с выбранного входа. Аналоговые мультиплексоры иногда называют **коммутаторами**.

Задания

1. В выключенном состоянии стенда установите в разъем стенда ОАВТ плату 4.
2. Установите на стенд шаблон IV-1 Дешифратор-демультимплексор.
3. Включите стенд ОАВТ тумблером на задней стенке.
4. Индикаторы HL1- HL8 должны светиться, что соответствует сигналам «1» по всем выходам (Обратите внимание, что выходы инверсные).
5. С помощью тумблеров SA1 –SA3 задавайте номер выхода, на который будет поступать сигнал «0». Соответствующий индикатор погаснет при нажатии кнопки SB2, разрешающей считывание. Запишите в таблицу заданный адрес и номер погасшего индикатора.

Заполните таблицу измерений.

SA3	SA2	SA1	HL1	HL2	HL3	HL4	HL5	HL6	HL7	HL8
0	0	0								
0	0	1								
0	1	0								
0	1	1								
1	0	0								
1	0	1								
1	1	0								
1	1	1								

6. Исследуйте работу мультиплексора. Замените карту - шаблон на стенде ОАВТ . Поставьте карту IV-3.
7. Введите адрес входа, сигнал с которого будет выведен на выход мультиплексора. Для этого с помощью тумблеров SA1-SA3 наберите адрес в двоичном коде (011). Нажмите

кнопку SB3. Код будет введен в регистр RG и поступит на адресный вход мультиплексора.

8. С помощью тумблеров SA1 – SA3 последовательно набирайте сигнал, поступающий на вход дешифратора. Преобразованный из двоичного в десятичный, он будет соответствовать номеру выхода дешифратора, на который поступит «0» при нажатии на кнопку SB2

9. Если номер входа дешифратора, на который поступает «0» совпадет с адресом заданным регистром RG для мультиплексора MUX , сигнал «0» поступит на выход мультиплексора и погасит индикатор HL1

Заполните таблицу измерений.

«0» - индикатор HL1 погашен «1» - индикатор HL1 светится

SA3	SA2	SA1	HL1
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

Лабораторная работа №6

Тема: Исследование сумматора

Цель: Исследование режима работы сумматора. Ознакомление с элементной базой микросхем сумматоров , выработка навыков составления и исследования схем с применением сумматоров.

Перечень необходимых средств обучения:

Стенд ОАВТ, платы с микросхемой сумматора, карты для стенда.

Краткие теоретические сведения

Микросхемы сумматоров предназначены для суммирования двух входных двоичных кодов, то есть выходной код будет равен арифметической сумме двух входных кодов. Например, если один входной код - 7 (0111), а второй - 5 (0101), то суммарный код на выходе будет 12 (1100)

На схемах сумматоры обозначаются буквами SM. В отечественных сериях код, обозначающий микросхему сумматора, - ИМ.

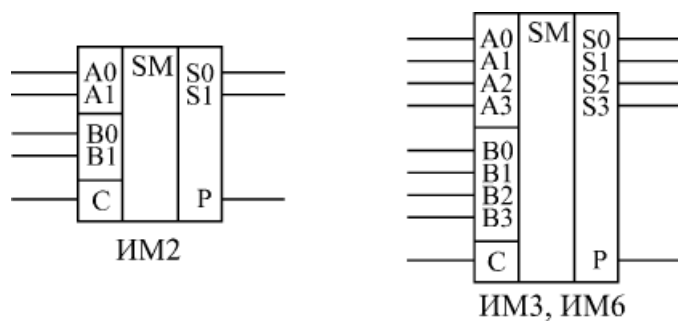
Сложение чисел в последовательных сумматорах осуществляется поразрядно, последовательно во времени. В сумматорах параллельного действия сложение всех разрядов многоразрядных чисел происходит одновременно.

Сумматор выполняет сложение. Он представляет собой последовательное соединение одноразрядных двоичных сумматоров, каждый из которых осуществляет сложение в одном разряде. При этом если сумма двух цифр в данном разряде больше или равна основанию используемой системы счисления, то возникает перенос старшего разряда в соседний сумматор.

При последовательном суммировании используется один, общий для всех разрядов полный сумматор с дополнительной цепью задержки.

Оба слагаемых кодируются последовательностями импульсов, которые синхронно вводятся в сумматор через входы *A* и *B*, начиная с младших разрядов.

Цепь задержки обеспечивает хранение импульса переноса *P_I* на время одного такта, то есть до прихода пары слагаемых следующего разряда, с которыми он будет просуммирован



A и *B* – входные сигналы. *S* – выходной сигнал. Суммируются A_0+B_0 ; A_1+B_1 . *C* – бит переноса от предыдущего сумматора

В лабораторной работе исследуется работа сумматора двух четырехразрядных двоичных чисел на интегральной микросхеме (ИС) К155ИМ3. Для задания двух четырехразрядных слагаемых используются тумблеры SA1 – SA4 (Слагаемое A) и регистр D1 (Слагаемое B). Ввод слагаемого B в регистр может осуществляться как в параллельном (с тумблеров SA1 – SA4), так и в последовательном (с кнопки SB2) режимах. Сигнал начального переноса P_0 подается с кнопки SB3 через инвертор D2.3, имеющийся в основном блоке. Индикация входных операндов A и B осуществляется двоичным восьмиразрядным индикатором (индикаторы HL2 – HL5 для операнда A и индикаторы HL6- HL9 для операнда B). Сигнал переноса в старший (пятый) разряд подается на индикатор HL1. Сумма в четырех первых разрядах дешифруется в блоке индикации и высвечивается в виде шестнадцатиричного числа на семисегментном индикаторе.

Задания:

1. Выключите стенд ОАВТ
2. Установите в разъем плату ПЗ
3. Установите карту – шаблон ПИ-3
4. Установите тумблер SA5 (сигнал «V») в положение «1». Этот режим соответствует параллельному вводу числа в универсальный регистр – микросхему D1. Тумблерами SA1 (младший разряд), SA2, SA3, SA4 (старший разряд) наберите число 0101. Нажав кнопку SB1 (синхроимпульс) введите набранное число в регистр и в сумматор в качестве слагаемого B. Наберите тумблерами SA1 – SA4 второе слагаемое число 0011. Числа, с

которыми проводятся операции называют операндами. При суммировании учитывать сигнал переноса SB3

На индикаторе HG1 высветится сумма этих двоичных чисел в шестнадцатиричном коде, число 8. Выполняем проверку суммирования:

$$0101_2 + 0011_2 = 8_{16} = 1000_2$$

$$5_{10} + 3_{10} = 8_{10}$$

2. Выполните операции сложения чисел

$$0101 + 1001$$

$$0011 + 0100$$

$$0011 + 0111$$

$$0111 + 0111$$

$$1100 + 1011$$

Проверьте с помощью десятичных чисел результаты. Объясните полученные данные

3. Выполните операцию вычитания двух четырехразрядных двоичных чисел. Для этого необходимо перевести вычитаемое в дополнительный код и провести операцию сложения уменьшаемого и дополнительного кода вычитаемого.

Пример вычитания двоичных чисел

$$N_1 = 1100 \quad N_2 = 1001 \quad N_1 - N_2 = ?$$

N_2 - вычитаемое. Определяем дополнительный код этого числа. Для этого:

✓ Инвертируем N_2 (Заменяем «1» на «0» и «0» на «1»). Получаем 0110

✓ Добавляем «1» к младшему разряду. Получаем 0111

$$N_{2\text{доп}} = 0111$$

Выполняем сложение

$$\text{Слагаемое } N_1 \quad 1100_2$$

$$\text{Слагаемое } N_{2\text{доп}} \quad 0111_2$$

$$\text{Сумма } N_1 + N_2 \quad 1\ 0011_2$$

$$\text{Проверка } 1100_2 = 12_{10} \quad 1001_2 = 9_{10}$$

$$12_{10} - 9_{10} = 3_{10} = 0011_2$$

Перенос поступает в знаковый разряд. Если знак не учитывается, как в данной схеме то «1» отбрасывается. Остается 4-разрядное число. 5 разряд отбрасывается.

Результат вычитания 0011_2

Выполните вычитание чисел на стенде ОАВТ

$$1101 - 1001$$

$$1110 - 1011$$

Проверьте полученные результаты

Контрольные вопросы

1. Что такое сумматор? Для чего он предназначен?
2. Характеризуйте полусумматор. Приведите таблицу истинности полусумматора.
3. Объясните принцип работы одноразрядного сумматора с тремя входами.
4. Объясните принцип работы многоразрядного сумматора

Лабораторная работа №7

Тема: Исследование работы арифметико-логического устройства

Цель: Исследование режима работы арифметико-логического устройства .

Ознакомление с элементной базой микросхем АЛУ , выработка навыков составления и исследования схем с применением АЛУ

Перечень необходимых средств обучения:

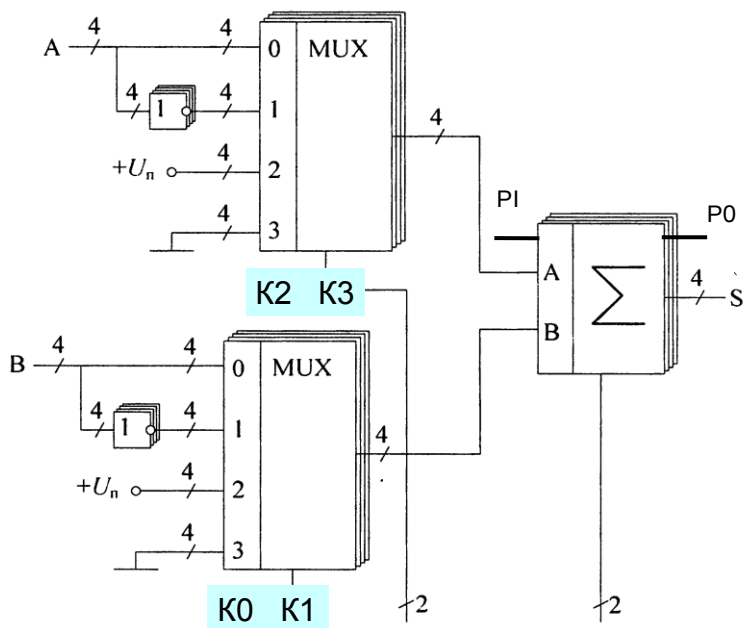
Стенд для изучения АЛУ.

Краткие теоретические сведения

Основу микропроцессора (МП) составляет *арифметико-логическое устройство (АЛУ)*, выполняющее арифметические, логические операции и операции сдвига над данными, представленными в двоичном коде.

В АЛУ имеется два входа для операндов и один выход для результата.

Упрощенная структура АЛУ приведена на рис 1



Структурная схема арифметического устройства

Рис.1 Упрощенная схема четырехразрядного АЛУ

Сигналы управления

Адресный код «00» подключает к входу сумматора «0» вход данных мультиплексора MUX

Код «01» - «1»

Код «10» «2»

Код «11» «3»

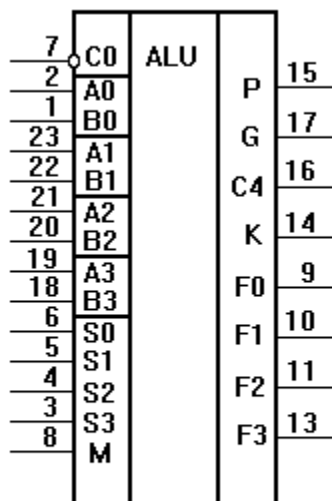
Сигнал переноса сумматора PI

Для данного устройства можно составить таблицу входных и выходных сигналов

Таблица 18.2. Список команд арифметического устройства

Номер строки	Управляющий код					Выполняемая операция
	K0	K1	K2	K3	PI	
0	0	0	0	0	0	$S = A + B$
1	0	0	0	0	1	$S = A + B + 1$
2	0	0	0	1	0	$S = B - A - 1$
3	0	0	0	1	1	$S = B - A$
4	0	0	1	0	0	$S = B - 1$
5	0	0	1	0	1	$S = B$
6	0	0	1	1	0	$S = B$
7	0	0	1	1	1	$S = B + 1$
8	0	1	0	0	0	$S = A - B - 1$
9	0	1	0	0	1	$S = A - B$

Таким образом, подавая на вход сигналы K0,K1,K2,K3,PI можно реализовать устройство выполняющее различные арифметические операции с четырехразрядными числами A и B. В более совершенном варианте устройство способно выполнять операции с логическими операндами. В данной работе исследуется работа микросхемы АЛУ K155ИПЗ, условное графическое обозначение и технические данные которой приведены ниже.



Выводы: $+U_{ИП} - 24$; общий – 12.

A0-A3, B0-B3 - входные сигналы

F0-F3 - выходные сигналы

C0 - входной перенос

C4 - выходной перенос

S0-S3 - выбор функций, выполняемых

арифметико-логическим устройством

M=1 - логические функции

M=0 - арифметико-логические функции

Примечание: входной и выходной переносы в отличие от остальных сигналов представлены инверсным кодом

Рис. 2.12

Для исследования работы АЛУ K155ИПЗ применяется специальный стенд. Управление работой микросхемы (подача сигналов «0» и «1» на входы S0 -

S3 и набор значения операндов A и B) производится с помощью кнопок и тумблеров.

Задания:

1. Изучите назначение кнопок и тумблеров стенда
2. Получите у преподавателя индивидуальное задание для выполнения различных арифметических и логических операций
3. С помощью элементов коммутации (тумблеров и кнопок) произведите выполнение арифметических и логических операций
4. Сравните полученный результат с расчетными значениями
5. Подготовьте отчет о проделанной работе

Контрольные вопросы.

1. Что такое АЛУ? Назначение и применение АЛУ.
2. Объясните, как получить результаты в таблице входных и выходных сигналов АЛУ
3. Объясните принцип работы одноразрядного сумматора с тремя входами.
4. Объясните принцип работы многоразрядного сумматора

Лабораторная работа №8

Тема: Исследование микросхем оперативной памяти

Цель: Исследование принципа работы микросхем оперативной памяти. Ознакомление со способами записи, стирания, чтения информации в микросхемах оперативной памяти. Выработка навыков работы с микросхемами памяти K155PY2.

Перечень необходимых средств обучения:

Стенд для изучения микросхемы памяти 155PY2.

Краткие теоретические сведения

Запоминающие устройства (ЗУ) предназначены для записи, хранения и воспроизведения больших объемов цифровой информации и обмена ею с другими цифровыми устройствами.

Оперативная память (RAM)- это рабочая область для процессора компьютера. Во время работы в ней хранятся программы и данные.

Информация в оперативной памяти сохраняется только при включенном питании или до нажатия кнопки «Reset», то есть энергозависимая.

RAM- (Random Access Memory)- память с произвольным доступом. Это означает, что обращение к данным не зависит от их расположения в памяти (в отличие от файловой системы на диске).

На рис 1 приведена иерархия оперативной памяти

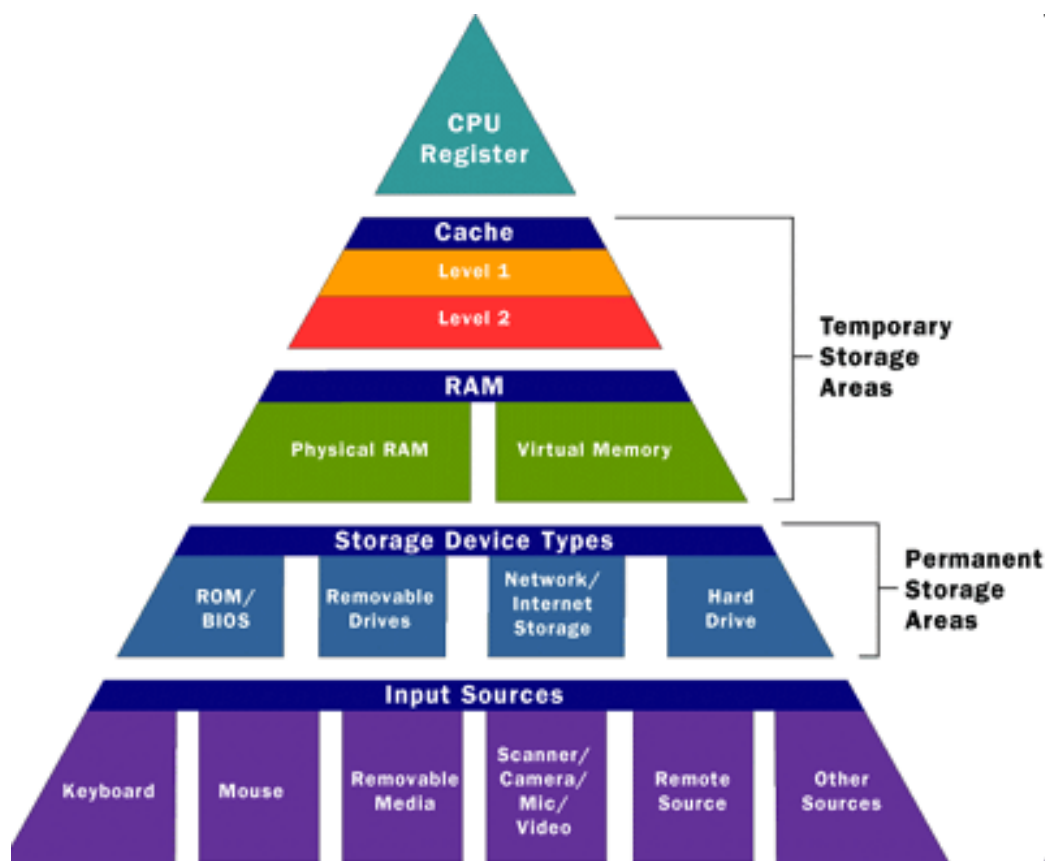


Рис1 Иерархия оперативной памяти

Оперативная память характеризуется следующими параметрами:

1. Информационная емкость — максимально возможный объем хранимой информации. Выражается в битах или байтах (словах). Бит хранится запоминающим элементом (ЗЭ), а слово—запоминающей ячейкой (ЗЯ), т. е. группой ЗЭ, к которым возможно лишь одновременное обращение.

Байт равен 8 битам. Более крупными единицами являются Кбит (килобит) равный $2^{10}(1024)$ бит и Мбит (мегабит) равный $2^{20}(1048576)$ бит.

2. Организация ЗУ— произведение числа хранимых слов на их разрядность.

Это произведение определяет информационную емкость ЗУ, однако при одной и той же информационной емкости организация ЗУ может быть различной.

3. Быстродействие ЗУ

Оценивается временами считывания (чтения), записи и длительностью цикла чтения/записи. **Время считывания** — интервал между моментами подачи сигнала чтения и появления цифровой информации, соответствующей заданному адресу, на выходе ЗУ.

Время записи — интервал после подачи сигнала записи, достаточный для установления ЗЭ (ЗЯ) в состояние, задаваемое входной цифровой информацией

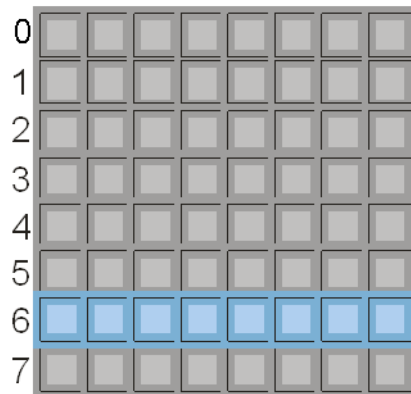
Цикл (чтения или записи)— минимально допустимый интервал между последовательными чтениями или записями.

Длительность цикла может превышать суммарное время чтения и записи, т. к. после этих операций может потребоваться время для восстановления необходимого начального состояния ЗУ

Пространство памяти представлено в виде ячеек (прямоугольников), которые состоят из определённого количества строк и столбцов. Один такой "прямоугольник" называется страницей, а совокупность страниц называется банком.

Структура запоминающего устройства

- ЗУ состоит из запоминающего массива и электронного обрамления.
- Запоминающий массив (накопитель) содержит запоминающие элементы, каждый из которых может хранить «0» или «1».
- N-разрядное слово хранится в N запоминающих элементах составляющих ячейку.
- Ячейка имеет свой адрес в накопителе, например выделенная ячейка имеет адрес 6_{10}



Шины компьютера

Существует 3 типа шин:

- Шина адреса (*направляются адреса памяти или внешних устройств*)
- Шина данных (*Передаются данные и инструкции от одной части системы к другой*)
- Шина управления (*Передаются сигналы управления - готовность принять данные, подтверждение приема и т.д.*)

Кроме сигналов адреса и данных на микросхемы памяти поступают сигналы:

- Разрешения чтения и записи
- Выбор чипа (кристалла)

В некоторых микросхемах могут использоваться другие виды сигналов

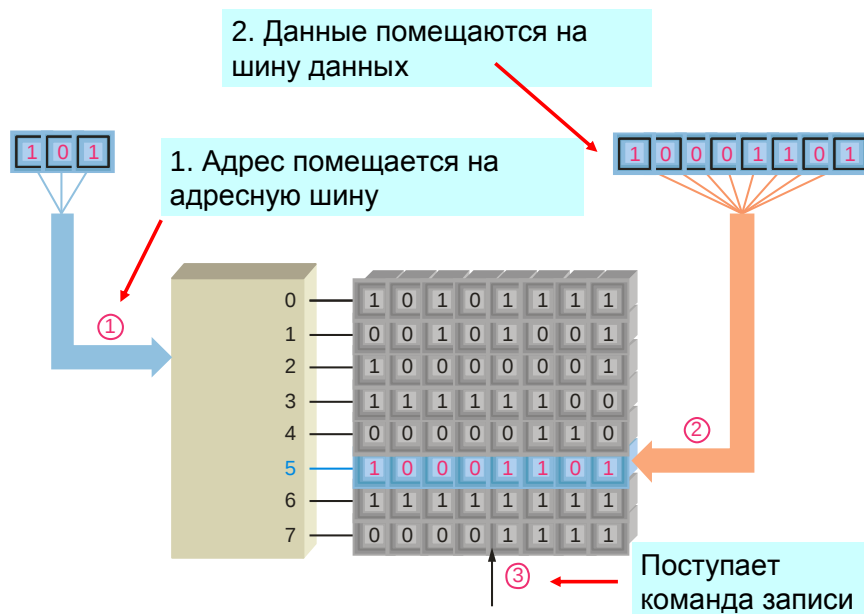
Read Enable (RE) и Write Enable (WE) Эти сигналы посылаются процессором для управления перемещением данных из памяти или в память.

Chip Select (CS) или Chip Enable (CE) используется для декодирования адреса. Все другие сигналы не воспринимаются, если Chip Select не активен.

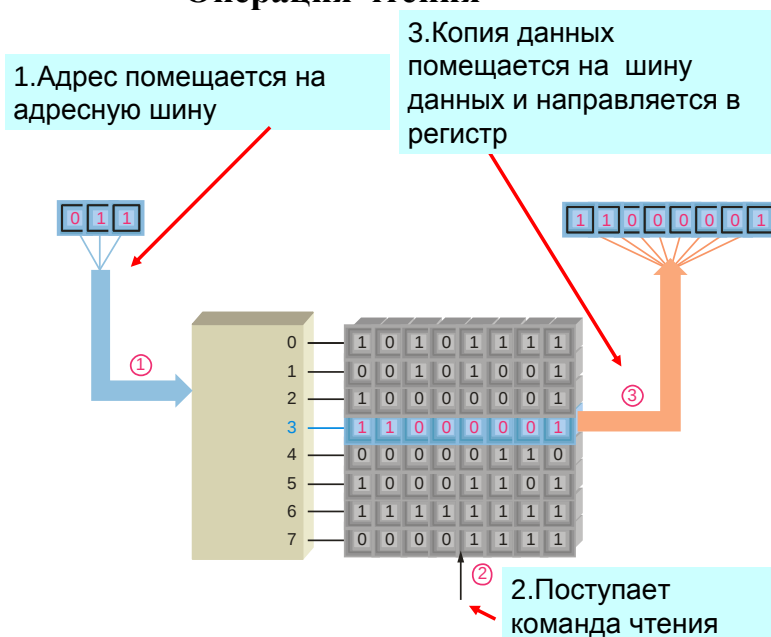
Output Enable (OE) активен во время операции чтения. Он подключает память к шине данных.

Перемещение информации из памяти и в память

Операция Записи



Операция чтения



Организация ЗУ

Произведение числа хранимых слов на их разрядность определяет информационную емкость ЗУ, однако при одной и той же информационной емкости организация ЗУ может быть различной, например ОЗУ 16 х 4 бит (16 ячеек по 4 бита (элемента) в ячейке). В качестве примера на рис 2 приведена структура микросхемы оперативной памяти

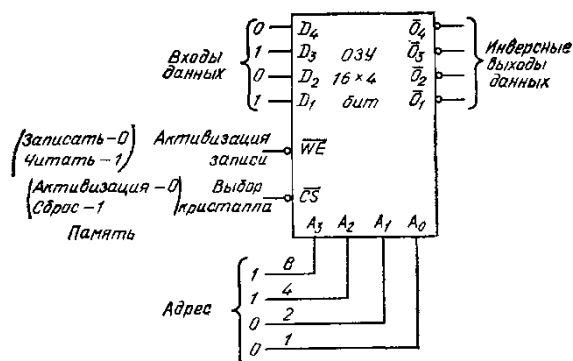


Рис. 3.25. Логическая схема ОЗУ 16×4 бит

Рис 2 Структура микросхемы ОЗУ

Структура ОЗУ 16×4 бит

Адрес	Бит D	Бит C	Бит B	Бит A
Слово 0				
Слово 1				
...	...	...	...	...
Слово 11				
Слово 12	0	1	0	1
Слово 13				

Определяется адресом 1100₂
(12₁₀)

В этой лабораторной работе исследуется микросхема оперативной памяти 7489 (Российский аналог 155PY2). Цоколевка микросхемы приведена на рис 3

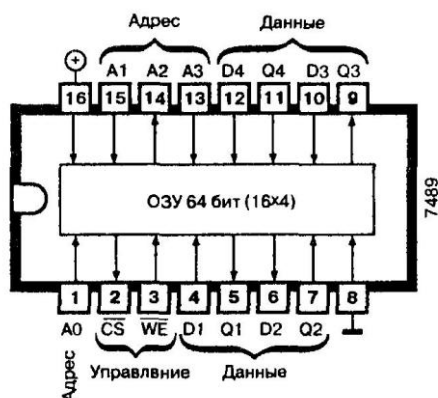


Рис3 Микросхема 7489

Задания

1. Ознакомьтесь с техническими характеристиками микросхемы оперативной памяти K155PY2 Приложение 1 методических указаний к лабораторной работе.
2. Ознакомьтесь с устройством стенда, назначением тумблеров и кнопок на лицевой панели.
3. Включите стенд через источник питания +5 Вольт
4. Выполните очистку ячеек памяти микросхемы. Для этого:

Наберите на счетчике СТ номер ячейки в двоичном коде например

0
1
1
1
1

Там где «1» должен гореть светодиод; младший разряд находится в верхней части стенда.

Тумблерами D1 – D4 выставьте слова (число) «0000»

Последовательно нажимая кнопку «W» n раз, введите 0000 во все ячейки микросхемы (от 0000 до 1111). При нажатии кнопки выполняется запись заданного числа в выбранную ячейку и к адресу ячейки прибавляется «1». Далее операция повторяется.

Проверьте результат очистки. Для этого:

Последовательно нажимая кнопку «С» подавайте сигнал на вход «Чтение» микросхемы.

При этом на счетчике СТ индицируется номер ячейки, а на выходе микросхемы Q1 – Q4 индицируется четырехразрядное число, содержащееся в ячейке. В случае, если операция выполнена правильно, во всех ячейках будет содержаться «0000»

5. Получите индивидуальное задание от преподавателя: номера ячеек и двоичные четырехразрядные числа, которые должны быть записаны в эти ячейки. Выполните запись. Результаты проверьте и продемонстрируйте преподавателю.

5. Оформите отчет о проделанной работе

Контрольные вопросы

1. Характеризуйте оперативную память
2. Что такое время доступа?
3. Характеризуйте процесс записи информации в ячейку памяти
4. Характеризуйте процесс чтения информации из ячейки памяти

Приложение 1

Характеристики микросхемы 7489 (аналог 155PY2)

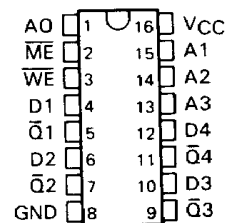
SN7489

64-BIT RANDOM-ACCESS READ/WRITE MEMORY

D1416, DECEMBER 1972—REVISED FEBRUARY 1984

- For Application as a "Scratch Pad" Memory with Nondestructive Read-Out
- Fully Decoded Memory Organized as 16 Words of Four Bits Each
- Fast Access Time . . . 33 ns Typical
- Diode-Clamped, Buffered Inputs
- Open-Collector Outputs Provide Wire-AND Capability
- Typical Power Dissipation . . . 375 mW
- Compatible with Most TTL Circuits

SN7489 . . . J OR N PACKAGE
(TOP VIEW)



logic symbol

Лабораторная работа №9

Тема: Исследование микросхем ПЗУ

Цель: Изучить устройство и маркировку микросхем ПЗУ
 способы программирования микросхем
 Научиться выполнять прошивку микросхем с помощью программатора «Мастер 32»

Перечень необходимых средств обучения:

компьютер, программатор «Мастер 32», микросхемы ПЗУ

Краткие теоретические сведения

Память — это сложная структура, включающая в себя множество элементов. Память предназначена для запоминания, хранения каких либо массивов информации: наборов, таблиц, групп цифровых кодов. Каждый код хранится в отдельном элементе памяти, называемом ячейкой памяти. Основная функция любой памяти и состоит в выдаче этих кодов на выходы микросхемы по внешнему запросу. Основным параметром памяти это объем, то есть количество кодов, которые могут в ней храниться, и разрядность этих кодов. В зависимости от способа занесения (записи) информации и от способа ее хранения, память разделяется на следующие основные типы:

Постоянная память (ПЗУ — постоянное запоминающее устройство, **ROM** — Read Only Memory — память только для чтения), в которую информация заносится на этапе изготовления микросхемы. Информация в памяти не пропадает при выключении ее питания, поэтому ее еще называют энергонезависимой памятью.

Оперативная память (ОЗУ — оперативное запоминающее устройство, **RAM** — Random Access Memory — память с произвольным доступом), запись информации в которую наиболее проста и может производиться пользователем сколько угодно раз на протяжении всего срока службы микросхемы. Информация в памяти пропадает при выключении ее питания

Для обозначения количества ячеек памяти используются следующие специальные единицы измерения:

1К — 1 килобайт, 1024 байт
 1М — 1 мегабайт, 1024 килобайт
 1Г — 1 гигабайт, 1024 мегабайт

Принцип организации памяти записывается следующим образом:

сначала пишется количество ячеек, а затем через знак умножения — разрядность кода, хранящегося в одной ячейке. Например, организация памяти 64Кх8 означает, что память имеет 64К (то есть 65536) ячеек и каждая ячейка — восьмиразрядная. А организация памяти 4М х 1 означает, что память имеет 4М (то есть 4194304) ячеек, причем каждая ячейка имеет всего один разряд. Общий объем памяти измеряется в байтах (килобайтах, мегабайтах, гигабайтах) или в битах.

Существует несколько разновидностей ПЗУ, предназначенных для различных целей.

Типы программируемых микросхем:

PROM (programmable read-only memory — программируемая память только для чтения) — чип памяти, данные в который могут быть записаны всего один раз. То, что однажды записано в PROM, сохраняется в нем навсегда. В отличие от основной памяти, PROM сохраняет данные даже в том случае, когда компьютер выключен. Микросхемы PROM изначально производятся чистыми, а для записи данных используют программаторы. При записи информации в микросхеме определенным образом пережигаются перемычки между элементами, поэтому информацию перезаписать невозможно..

EPROM (erasable programmable read-only memory — стираемая программируемая память, используемая только для чтения) — специальный тип PROM, который может очищаться под действием ультрафиолетовых лучей. В микросхемах EPROM имеется небольшое кварцевое окно для стирания светом. После стирания микросхема EPROM может быть перепрограммирована. Общий вид одной из микросхем EPROM приведен на рис 1

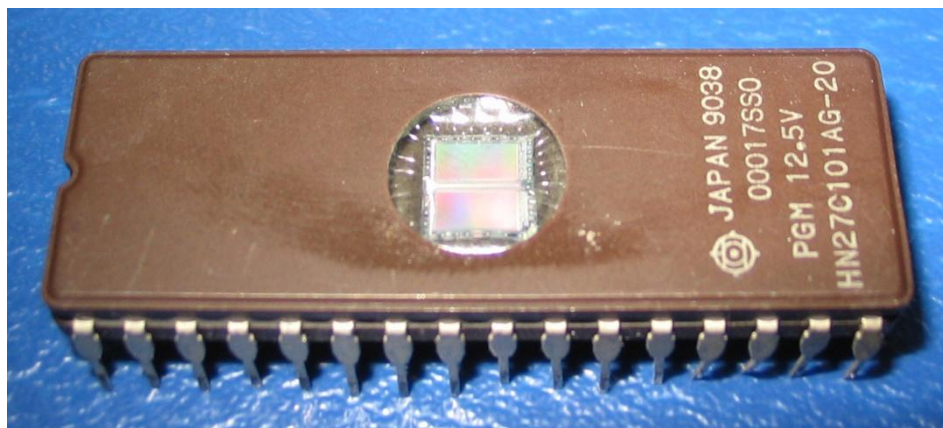


Рис.1 Микросхема EPROM с ультрафиолетовым стиранием.

EEPROM (electrically erasable programmable read-only memory — электрически стираемая программируемая память, используемая только для чтения) — специальный тип PROM, который может быть очищен электрическим разрядом. Подобно другим типам PROM, EEPROM сохраняет данные и при выключенном питании компьютера. Специальный тип EEPROM, называемый Flash memory или Flash EEPROM, может быть перезаписан непосредственно в компьютере, без применения дополнительных устройств типа программатора.

Общий вид микросхемы EEPROM на материнской плате компьютера приведен на рис 2

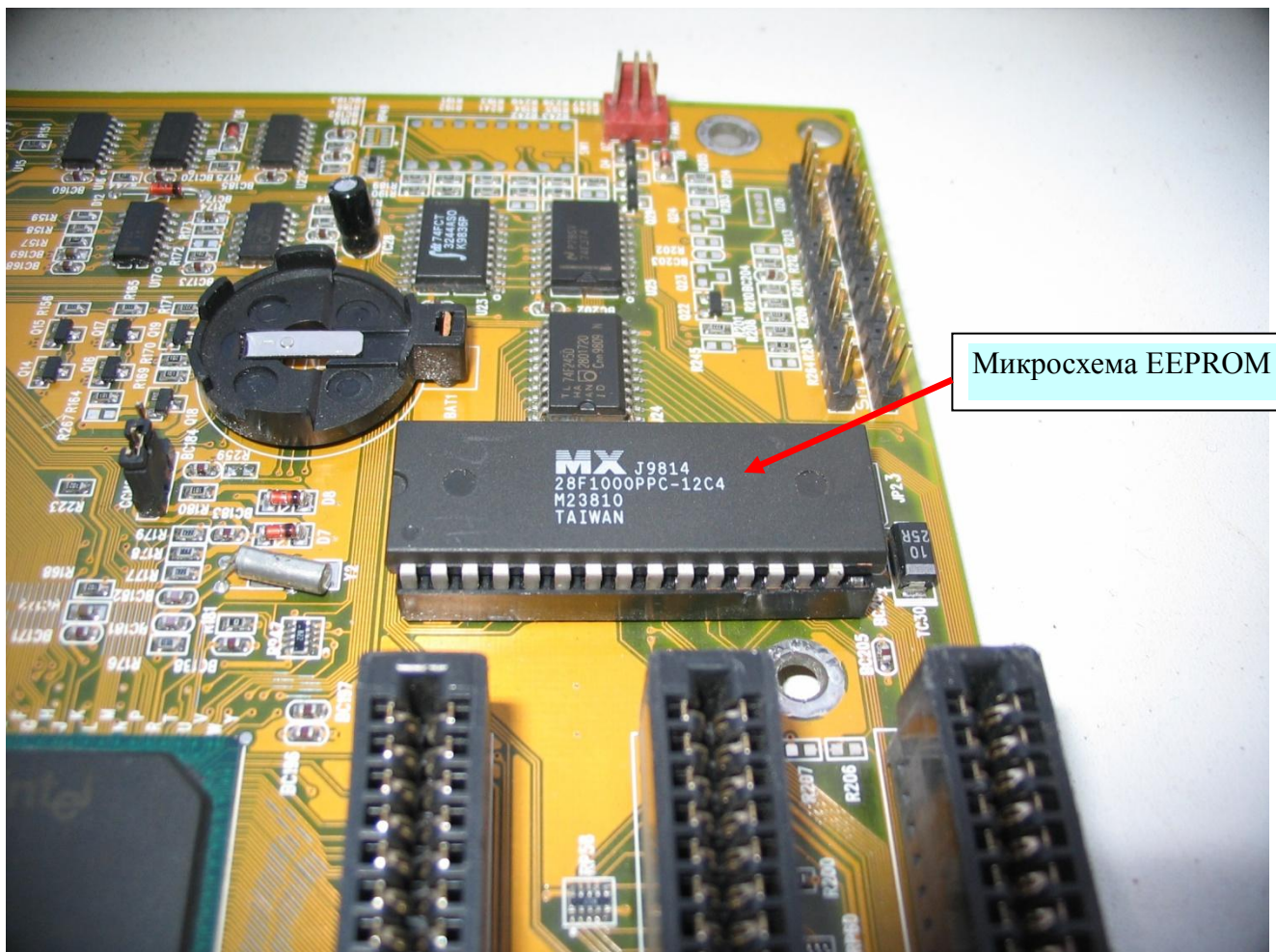


Рис 2 Микросхема EEPROM

Маркировка микросхем ПЗУ

- 27Сxxx- микросхема EPROM с ультрафиолетовым стиранием. XXX- цифры означающие емкость микросхемы в килобитах.
- 28 Fxxxx- микросхема EEPROM. Записывается и стирается электрическими импульсами. За счет использования полезной площади кристалла электрическими цепями стирания, плотность записи информации уменьшилась
- 29Fxxx[микросхема Flash EEPROM. Записывается и стирается электрическими импульсами. По сравнению с EEPROM изменена система записи и стирания информации.

Содержимое ПЗУ обычно изображается в виде специальной таблицы, называемой картой прошивки памяти. В таблице показывается содержимое всех ячеек памяти, причем в каждой строке записывается содержимое 16 (или 32) последовательно идущих (при нарастании кода адреса) ячеек. При этом, как правило, используется 16-ричное кодирование.

Рис. 3. Пример карты прошивки ПЗУ.

Адрес	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
10	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
20	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
30	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
40	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
50	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
60	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
70	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
80	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
90	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
A0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
B0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
C0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
D0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
E0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
F0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF

ПЗУ как универсальная комбинационная микросхема

Одно из самых распространенных применений микросхем ПЗУ — замена ими сложных комбинационных схем. Такое решение позволяет существенно упростить проектируемое устройство и снизить количество используемых комбинационных микросхем, а иногда даже уменьшить потребляемый ток и увеличить быстродействие схемы.

Суть предлагаемого подхода сводится к следующему. Если рассматривать адресные входы микросхемы ПЗУ как входы комбинационной схемы, а разряды данных — как выходы этой комбинационной схемы, то можно сформировать любую требуемую таблицу истинности данной комбинационной схемы. Для этого всего лишь надо составить таблицу прошивки ПЗУ, соответствующую нужной таблице истинности. В этом случае не надо ни подбирать логические элементы, ни оптимизировать их соединения, ни думать о том, можно ли вообще построить заданную комбинационную схему из стандартных микросхем. Важно только, чтобы количество требуемых входов не превышало количества адресных разрядов ПЗУ, а количество требуемых выходов не превышало разрядности шины данных ПЗУ.

ПЗУ в генераторах импульсных последовательностей

Следующее важнейшее применение ПЗУ — это построение генераторов сложных последовательностей цифровых импульсов. Такие генераторы широко используются в самых разных измерительных системах, в устройствах автоматики, в телевизионных системах, в схемах управления линейными или матричными индикаторами и т.д.

Задача в данном случае ставится следующим образом. Необходимо сформировать последовательность из нескольких сигналов различной длительности, сдвинутых относительно друг друга на различные временные интервалы. Причем последовательность эта может быть как разовой (однократно начинающейся по внешнему сигналу), так и периодической, непрерывно повторяющейся.

Принцип работы микросхемы ПЗУ

Основа ПЗУ - накопитель информации, занимающий почти всю площадь кристалла. Вдоль и поперек кристалла проложены непересекающиеся шины, образующие прямоугольную матрицу, в узлы которой включены запоминающие элементы - МОП-транзисторы специальной структуры. Логические сигналы на одну систему шин поступают с выхода дешифратора части адресных разрядов. При этом высокий уровень устанавливается только на одной из этих шин, и, следовательно, если какие-либо транзисторы в выбранном ряду открыты, то такой же уровень установится и на соответствующих вертикальных шинах.

Сигналы вертикальных шин поступают на входы мультиплексора, подключающего к выходу микросхемы только одну из шин. Какая шина будет подключена к выходу, зависит от значения разрядов адреса, неиспользуемых для дешифрации горизонтальных шин. Следовательно, при

каждой комбинации сигналов на адресных входах БИС выходной сигнал определяется состоянием одного из транзисторов, находящегося на перекрестии шин, выбранных кодами X и Y. Запоминающая способность ПЗУ с электрическим программированием и стиранием информации ультрафиолетовым излучением определяется особенностями транзисторных структур, входящих в матрицу запоминающих элементов. В отличие от обычных МОП-транзисторов они имеют затвор из поликристаллического кремния, расположенный в слое его окисла, благодаря чему такой затвор и получил название "плавающего". Запись информации (заряд затвора) происходит в результате приложения между стоком и истоком большого отрицательного напряжения. Электроны, получающие в электрическом поле достаточную энергию, преодолевают потенциальный барьер на границе полупроводника и окисла и дрейфуют к плавающему затвору, заряжая его отрицательно. Этот заряд создает в канале электрическое поле, стремящееся открыть МОП-транзистор.

После отключения программирующего напряжения заряд затвора не исчезает, поскольку двуокись кремния настолько хороший изолятор, что даже по истечении 10 лет на нем сохраняется более 70% заряда, а электрических соединений, через которые затвор мог бы разрядиться, нет. Однако если МОП-транзистор с плавающим затвором подвергнуть воздействию ультрафиолетовых лучей, электроны получают энергию, достаточную для того, чтобы "перескочить" обратно на подложку. Затвор при этом разрядится, и запоминающий элемент возвратится в свое первоначальное не запрограммированное состояние. Остается лишь добавить, что подачей программирующих напряжений на транзисторы матрицы управляет специальный блок, также расположенный на кристалле РПЗУ.

Процесс программирования РПЗУ достаточно прост и состоит в подаче на адресные входы адресов ячеек, в которые необходимо занести информацию, установке на выходах микросхемы данных для программирования и подаче на специальный вход PGM импульса управления программированием.

Принцип работы ячейки памяти ПЗУ приведен на рис 4 и рис 5

Ячейка Flash памяти

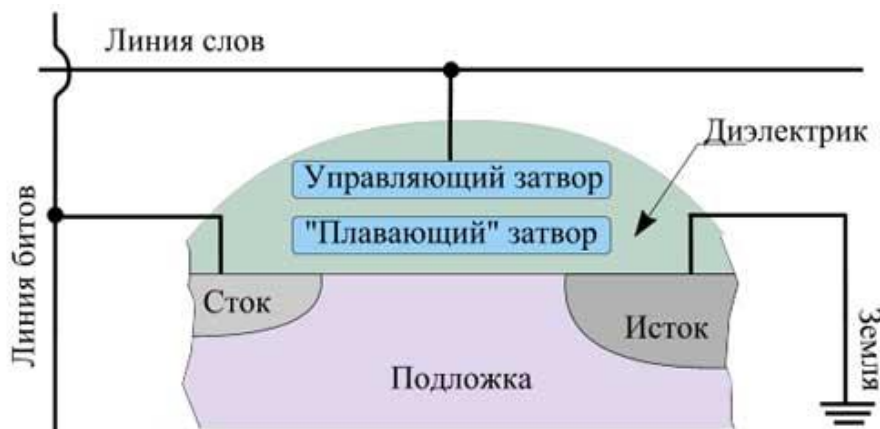
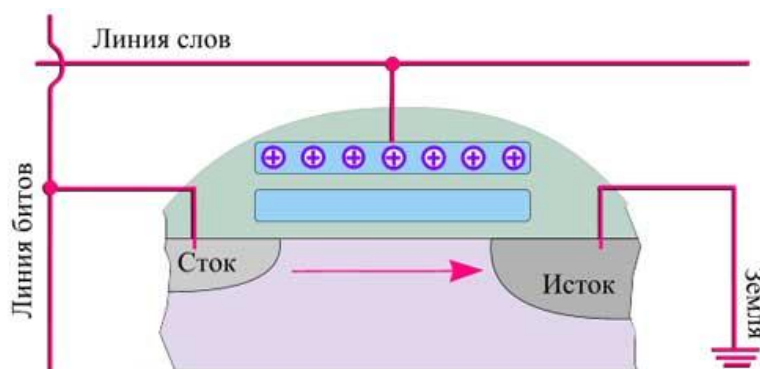
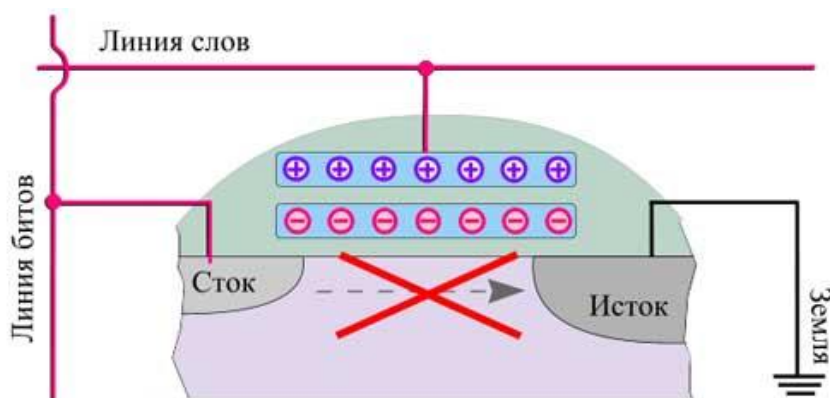


Рис 4 Ячейка Flash памяти



При чтении, в отсутствие заряда на "плавающем" затворе, под воздействием положительного поля на управляющем затворе, образуется n-канал в подложке между истоком и стоком, и возникает ток.



Наличие заряда на "плавающем" затворе меняет вольт-амперные характеристики транзистора таким образом, что при обычном для чтения напряжении канал не появляется, и тока между истоком и стоком не возникает.

Рис 5 Запись «0» и «1» в ячейку памяти

Срок хранения информации

Изоляция в ячейке неидеальна, заряд постепенно изменяется. Рекомендуемый максимальный срок хранения заряда, заявляемый большинством производителей для бытовых изделий — 10—20 лет.

Специфические внешние условия могут катастрофически сократить срок хранения данных. Например, повышенные температуры или радиационное (гамма-лучевое и высокоэнергичными частицами) облучение.

Микросхемы EPROM и EEPROM используются в компьютерной технике для записи BIOS системы и карт расширения, в принтерах и видеокартах. При выполнении ремонтных работ

часто возникает необходимость восстановить прошивку микросхемы или заменить её новой версией. В этом случае применяется программатор

Универсальный программатор "Мастер"



Программатор: это отдельное устройство на базе популярного микроконтроллера семейства MCS-51, с управлением от персонального компьютера, к которому программатор подключается через COM-порт и внешний блок питания. Такая реализация позволяет создавать надежные устройства с широкими функциональными возможностями, причем добавление поддержки новых микросхем осуществляется развитием программного обеспечения. Отличительными особенностями программаторов "Мастер" является использование высококачественных универсальных ZIF-панелей Aries в самом программаторе и панелей Wells на разнообразных адаптерах для корпусов микросхем SOIC, PLCC, TSOP, VSOP, PSOP, TQFP.

"Мастер-32" разрабатывался для сервисных центров по ремонту и обслуживанию различной электронной техники(телевизоры, радио и сотовые телефоны, АОНы) и для ремонта компьютерной техники.

Подключение программатора

Программатор подключается к компьютеру через последовательный интерфейс при выключенном компьютере. Поддерживается связь по любому из портов COM1-COM4. Скорость обмена выбирается программно и может составлять 9600, 57600 или 115200 б/с. Для подключения программатора к компьютеру необходимо кабелем соединить свободный порт компьютера с портом программатора.

После того как программатор подключен к компьютеру, необходимо включить его в сеть. Светодиод индицирует готовность программатора к работе. Далее нужно на ПК запустить управляющую программу и приступить к работе.

Устанавливать микросхемы в колодку необходимо выравняя их по первой ножке.

Первая ножка у рычажка ZIF-панели.



Внимание ! Подключение программатора к персональному компьютеру возможно только после его отключения от сети.

Внимание ! Перед установкой микросхемы в колодку программатора нужно программно выбрать этот тип схемы

Программа Masterwp

Работа с программой

В дальнейшем будет использоваться понятие "буфер". Буфером называется область памяти в которой находится считанная информация. Эта информация является исходной для программирования ПЗУ.

Программа предоставляет пользователю следующие возможности:

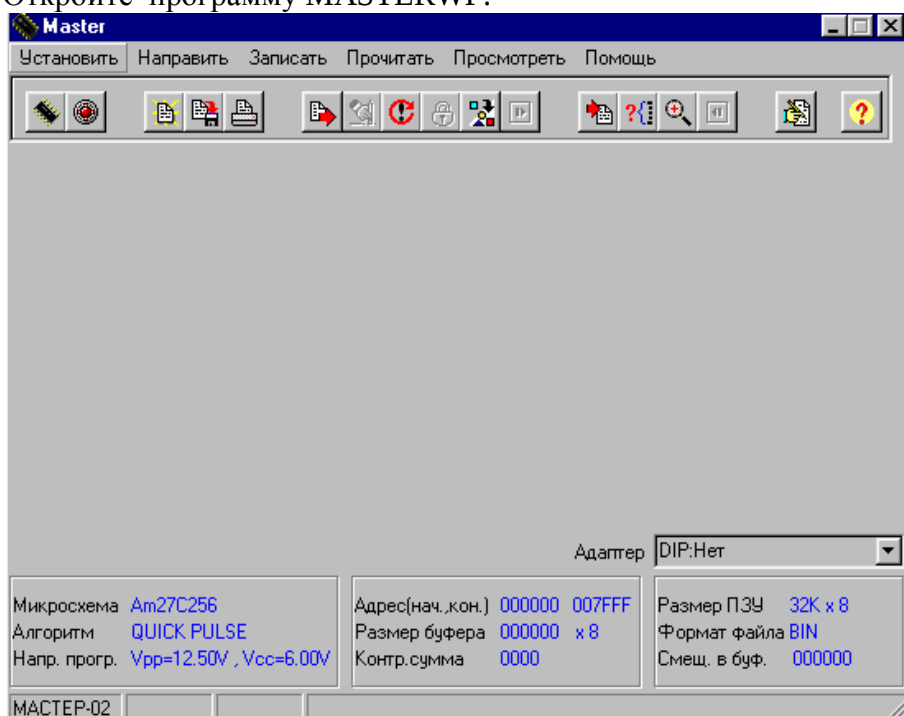
- установить тип ПЗУ (из списка);
- выбрать последовательный порт;
- установить скорость обмена;
- установить начальный и конечный адреса в ПЗУ;
- проверить ПЗУ на чистоту;
- прочитать ПЗУ в буфер;
- прочитать ПЗУ в файл;
- прочитать идентификационный код;
- прочитать файл в буфер;
- установить тип файла (BIN / HEX);
- установить смещение от начала файла;
- установить смещение в буфере;
- установить биты секретности;
- запрограммировать содержимое буфера в ПЗУ;
- запрограммировать ПЗУ содержимым файла;
- сохранить содержимое буфера в файле;
- сравнить содержимое ПЗУ с буфером;
- просмотреть и отредактировать содержимое ПЗУ.

Часто возникает потребность прошить микросхему ПЗУ, то есть перепрограммировать ее.

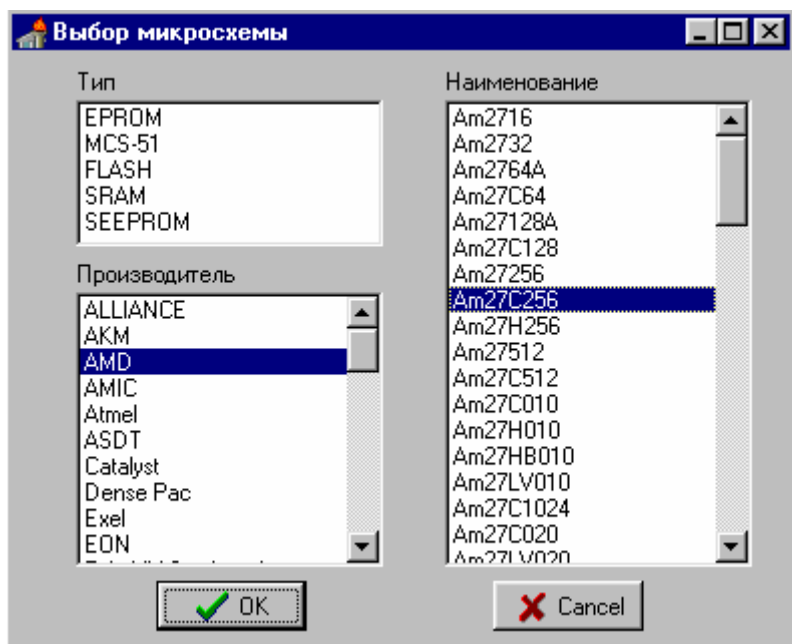
Чтобы информацию с одной микросхемы переписать на другую следует выполнить следующие действия.

Вставьте в программатор микросхему, с которой нужно считать информацию.

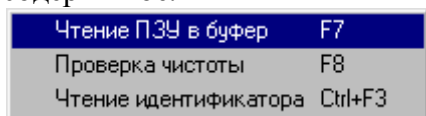
Откройте программу MASTERWP:



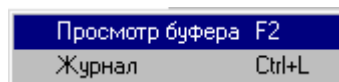
Чтобы начать работу с микросхемами, нужно в меню «установить» выбрать имя и фирму микросхемы. И затем в этом меню выбрать последовательный порт – COM1 или COM2.



После этого в меню «прочитать» выбрать «Чтение ПЗУ в буфер». При этом всё содержимое микросхемы будет скопировано в буфер, откуда можно смотреть и программировать его содержимое.

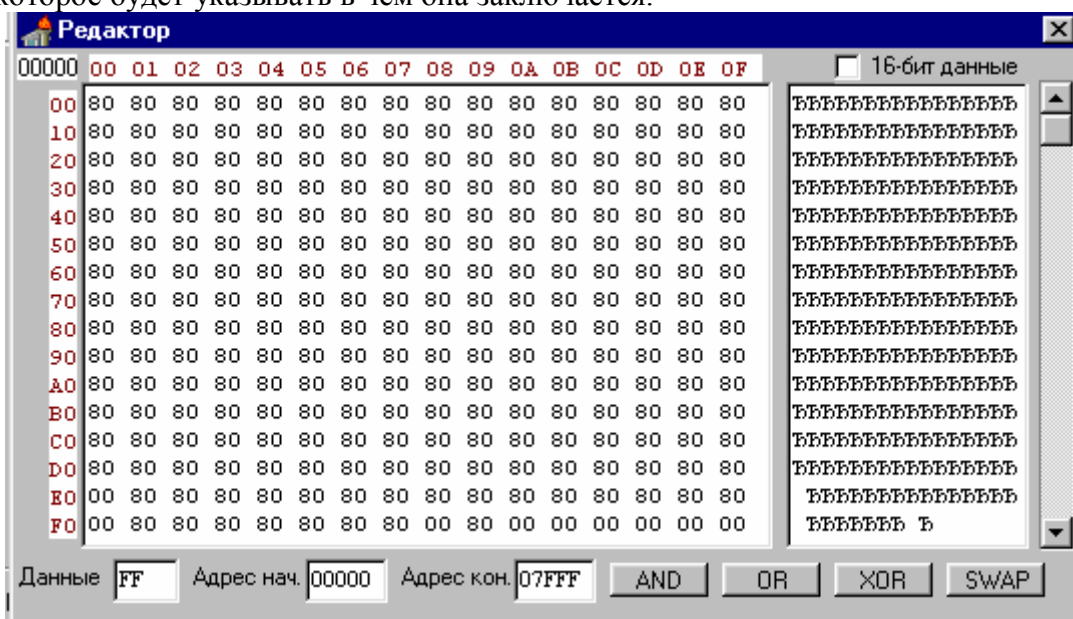


При нажатии меню «просмотр» -> просмотр буфера



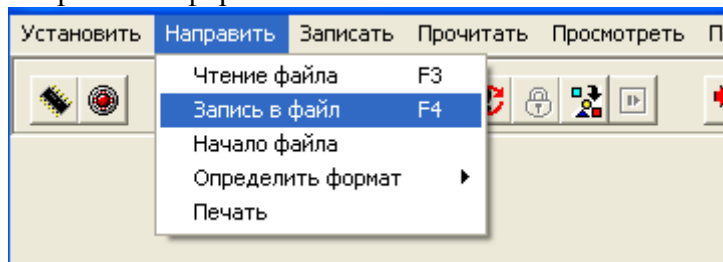
должна появиться следующая картина.

При этом если вы что – то делаете неправильно внизу должно появиться сообщение об ошибке, которое будет указывать в чем она заключается.



Открыть меню Направить -> Записать в файл

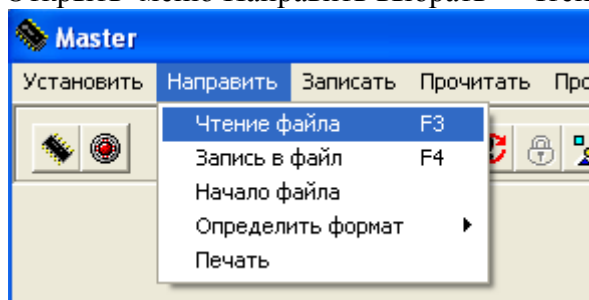
Сохранить в формате txt



Вставить в программатор другую микросхему (которую нужно перепрограммировать)

В меню Установить выбрать имя и фирму микросхемы

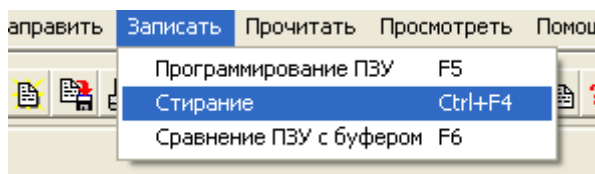
Открыть меню Направить выбрать ->Чтение файла:



При этом произойдет чтение сохраненного буфера.

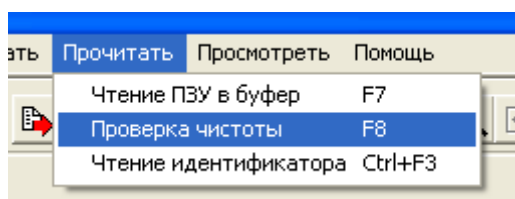
Чтобы перепрограммировать ПЗУ необходимо для начала стереть микросхему.

Для этого выбрать меню Запись->стирание.



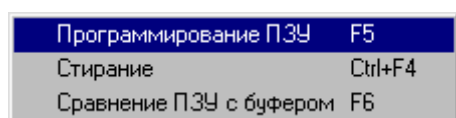
Еще один важный аспект это проверка чистоты. Микросхема считается чистой если не содержит никаких данных. Если данные останутся то запись будет невозможна.

Для этого нужно выбрать меню Прочитать->проверка чистоты.



После проведенных вышеуказанных пунктов можем Программировать.

Для этого в меню Записать выбрать пункт Программирование ПЗУ:



При этом произойдет чтение сохраненного буфера и непосредственно запись буфера на микросхему.

Задания

1. Изучите техническое описание программатора «Мастер 32». Обратите внимание на порядок подключения программатора к компьютеру, положение микросхемы в разъёме.
2. Запишите в файл содержимое предложенной микросхемы ПЗУ.
3. Проверьте на чистоту вторую микросхему.
4. Установите режим записи на программаторе
5. Запишите информацию из файла во вторую микросхему.
6. Проверьте полученную прошивку
7. Внесите в ручном режиме изменения в содержимое микросхемы (свою фамилию англ буквами).
8. Составьте отчёт о проделанной работе

Лабораторная работа №10

Тема: Программирование микропроцессора КР580. Изучение работы эмулятора.

Цель: Исследование режима работы микропроцессора. Ознакомление с блок-схемой процессора, выработка навыков составления и наладки программ для процессора КР580 на языке ассемблер с применением эмулятора

Перечень необходимых средств обучения: персональный компьютер, программа эмулятор КР580, таблица команд микропроцессора.

Краткие теоретические сведения

Микропроцессор (МП) - это программно-управляемое электронное цифровое устройство, предназначенное для обработки цифровой информации и управления процессом этой обработки, выполненное на одной или нескольких интегральных схемах с высокой степенью интеграции электронных элементов.

Упрощенная структура микропроцессора представлена на рис 1

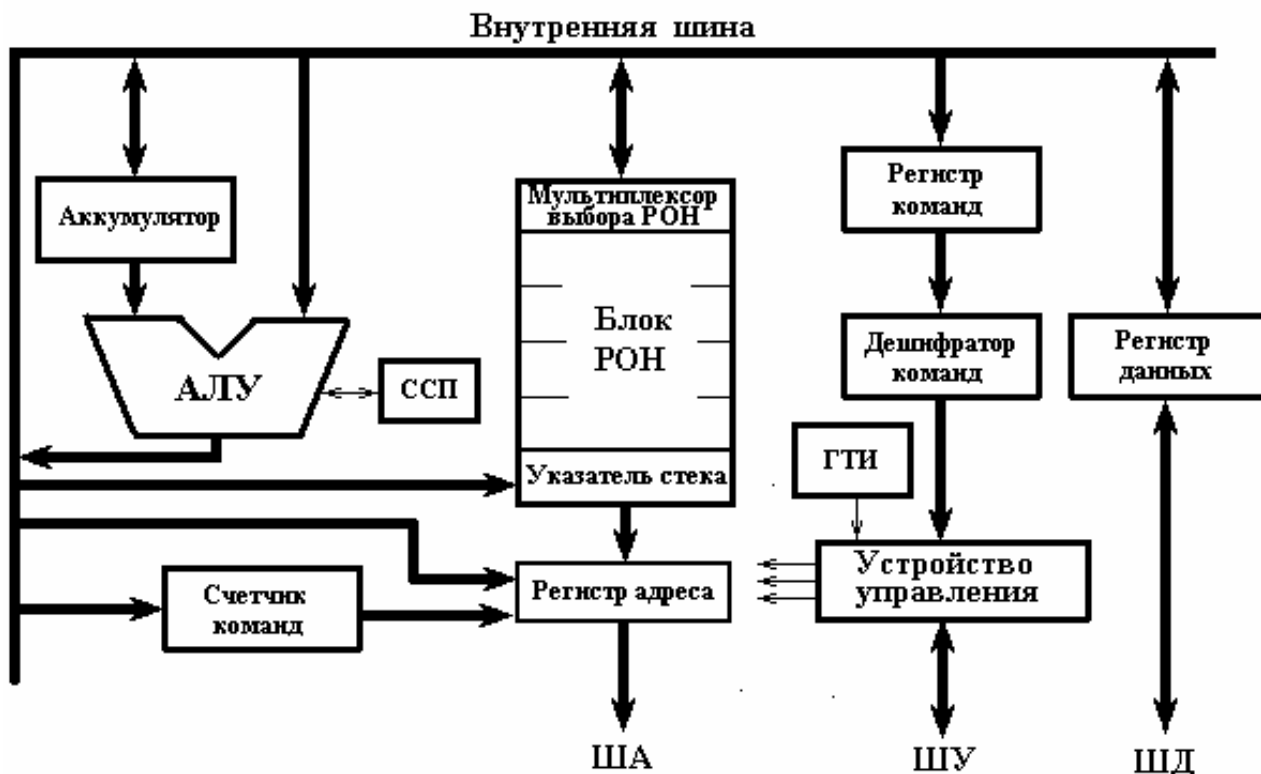


Рис 1 Упрощенная структура микропроцессора

Шина адресов **ША** (англ. **AB** - Address Bus),

Шина данных **ШД** (англ. **DB** -Data Bus)

Шина управления **ШУ** (англ. **CB** - Control Bus).

АЛУ- арифметико-логическое устройство

ССП- *регистр слова состояния процессора* .

- При выполнении операции в АЛУ на ее входы поступают операнды, из ячеек внешней памяти, подключаемой к МП. Туда же можно помещать и результат операции.
- Так как обращения к внешней памяти являются медленными в составе МП имеется несколько *рабочих регистров*. Один из этих регистров называется **аккумулятором** или регистром **А**.

В него помещается один из операндов, участвующих в операции, а также результат операции, формируемый в АЛУ.

- Использование такого регистра упрощает адресацию в различных командах, позволяет не указывать в команде адрес одного операнда и адрес регистра результата
- Остальные рабочие регистры используемые в операциях называются *регистрами общего назначения (РОН)*.
- Один из операндов часто хранится в одном из РОН, другой же – в аккумуляторе.

Для указания МП необходимых действий при выполнении программы используются команды, поступающие из внешней памяти по шине данных. Для приема и хранения текущей команды используется *регистр команд* (РК). Для идентификации команды используется *дешифратор команд* (ДК), в функции которого входит определение типа текущей команды и запуск такой последовательности действий МП, которая ее выполнит.

Процесс управления всеми элементами МП выполняет *устройство*

управления (УУ), которое принимает внешние сигналы и формирует ответные, управляющие и синхронизирующие сигналы, необходимые для работы внешних устройств.

Для формирования синхронизирующих сигналов в МП имеется *генератор тактовых импульсов* (ГТИ). Совокупность входных и выходных внешних управляющих сигналов называют *шиной управления* (ШУ).

- Адрес текущей команды формируется *счетчиком команд* (СК). Содержание СК загружается в *регистр адреса* (РА) перед чтением очередной команды или данных. В этом случае после каждого обращения к памяти состояние СК и, соответственно, РА возрастает. Такой метод формирования адреса рассчитан на то, что элементы программы (команды и данные) располагаются в последовательных ячейках памяти, начиная с некоторого, заранее определенного адреса, например, нулевого.

- В процессе выполнения программы может возникнуть необходимость изменения последовательного характера задания адреса. Например, если при проверке результата выполнения последней команды требуется одна из двух различных последовательностей команд, или требуется обращение к непоследовательно расположенной ячейке данных, то имеется возможность занесения новых данных как непосредственно в счетчик команд для изменения хода программы, так и в регистр адреса для обращения к произвольной ячейке памяти.

В этом случае прибегают к организации *стека* - набора внутренних или внешних ячеек памяти, используемых для временного хранения обрабатываемых данных. При этом загрузка и выгрузка стека происходит в последовательные ячейки.

Система команд микропроцессора

- Это набор допустимых для данного процессора управляющих кодов и способов адресации данных.
- Система команд жестко связана с конкретным типом процессора, поскольку определяется аппаратной структурой блока дешифрации команд, и обычно не обладает переносимостью на другие типы процессоров. Состав команд не нормализован и может быть от 8 до 250

Команды могут быть:

- Арифметические
- Логические
- Передачи данных
- Вызова подпрограмм
- Возврата из подпрограмм
- Прочие

Код команды ничем не отличается от обычных данных в двоичном коде, размещенных в памяти вычислителя. Конкретный двоичный код воспринимается и обрабатывается процессором как команда в том случае, когда он попадает в процессор в фазе чтения кода команды.

В двоичном коде команды существуют группы разрядов – **поля** – с различным функциональным назначением

•



- 1 - поле кода операции (КОП) объемом 1 или 2 байта;
2 - поле адресной части команды (АЧ) объемом от 1 до 4 байт

- **КОП** - код операции - двоичный код, однозначно указывающий процессору на выполнение конкретных действий (пересылка, сложение и т.п.), и определяющий при этом форму задания адресов операндов;

- **АЧ** - адресная часть - двоичное число, которое может представлять собой адрес (адреса) операндов, значение операнда, адрес следующей команды (адрес перехода, передачи управления).

Пример программы в машинных кодах

Адрес, Н-код	Двоичное содержимое		Адрес, Н-код	Содержимое, Н-код	
2000	0011 1110	← Начало программы	2000	3E	← Начало программы
2001	1011 0100		2001	B4	
2002	0010 1111		2002	2F	
2003	0011 0010		2003	32	
2004	0000 0000		2004	00	
2005	0010 0001		2005	21	
2006	0111 0110	← Конец программы	2006	76	← Конец программы
2007			2007		
		а)			б)

- Для перехода от человеческого языка к машинному применяются языки программирования. В данной работе используется язык программирования низкого уровня Ассемблер.
- Команды обозначаются сокращенными словами и фразами (**мнемоникой**). Например: **CMA** (Complement accumulator) – дополнить аккумулятор, то есть инвертировать содержимое аккумулятора; **ADD** (Addition)- сложить.

Программа, переводящая мнемонику в машинные коды называется транслятором.

Общий принцип составления программы с применением ассемблера приведен на рис 2. Буквы команды **CMA** переведены в соответствующий ASCII код. Затем три кода ASCII переведены в код команды инверсии содержимого аккумулятора.

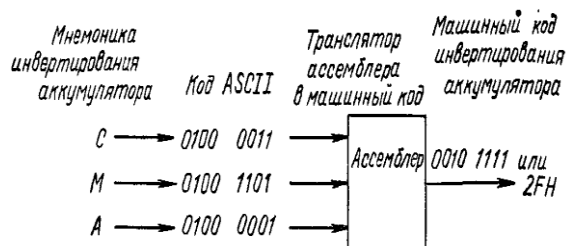


Рис 2 Преобразование программы с языка Ассемблер

Принцип создания программы на языке Ассемблер

- Каждая команда программы пишется в отдельной строке.
- Каждая строка разбивается на четыре поля, образующие четыре колонки.
- У каждого поля свое назначение:
Первое поле — поле метки .
Второе поле — поле команды (мнемокода).
Третье поле предназначено для параметров команды.
Четвертое поле — поле комментариев. *Заполнять это поле не обязательно. Оно занимает пространство до конца страницы. Сюда, если нужно, можно поместить комментарии к команде.*

Для того, чтобы транслятор отличал комментарий от самой программы, в начале комментария необходимо поставить символ «точка с запятой». Этот символ называют

признаком комментария. Встретив признак комментария, транслятор прекращает дальнейшую обработку строки и переходит к обработке следующей. В результате все, что стоит после точки с запятой до конца текущей строки игнорируется.

Пример написания одной строки программы на языке Ассемблер.
Выполнить действие: «переместить содержимое регистра R0 в аккумулятор».
Строка программы, реализующая эту команду, выглядит следующим образом:

MOV A,R0 ; Это команда перемещения из R0 в аккумулятор

Первое поле в данном случае пустое. То есть, в приведенной строке отсутствует метка.
Во втором поле записан мнемокод команды **MOV**.

В третьем поле записаны параметры команды. *Команда MOV имеет два параметра. В Ассемблере параметры команд принято записывать через запятую. Для улучшения внешнего вида текста можно вставлять как до, так и после каждого из параметров любое количество пробелов.*

- В рассматриваемой команде в качестве параметров выступают имена источника и получателя информации.
- Во всех подобных командах параметры записываются в следующем порядке: сначала записывается имя приемника информации («куда переместить»), а затем имя источника («откуда взять»).

Достоинства языка ассемблер

Требует минимум ресурсов (память)

Выполняет программу за минимальное время (количество операций- оптимизировано)

Недостатки языка Ассемблер

- Большое количество команд
- Сложности с составлением программы сложнее читать и понимать программу на Ассемблере по сравнению с языками программирования высокого уровня
- Как правило, меньшее количество доступных библиотек по сравнению с современными индустриальными языками программирования.
- Отсутствует переносимость программ на компьютеры с другой архитектурой и системой команд.

В лабораторной работе необходимо составить программу для микропроцессора модели КР 580 на языке Ассемблер и произвести её отладку. Прилагается описание команд ассемблера для КР580 и программа эмулятор работы микропроцессора. Эмулятор имитирует работу микропроцессора с программой. Есть возможность проверить содержимое ячеек и регистров и выполнить отладку (проверку и исправление программы).

Общий вид рабочего стола ПК с открытой программой эмулятора приведен на рис 3

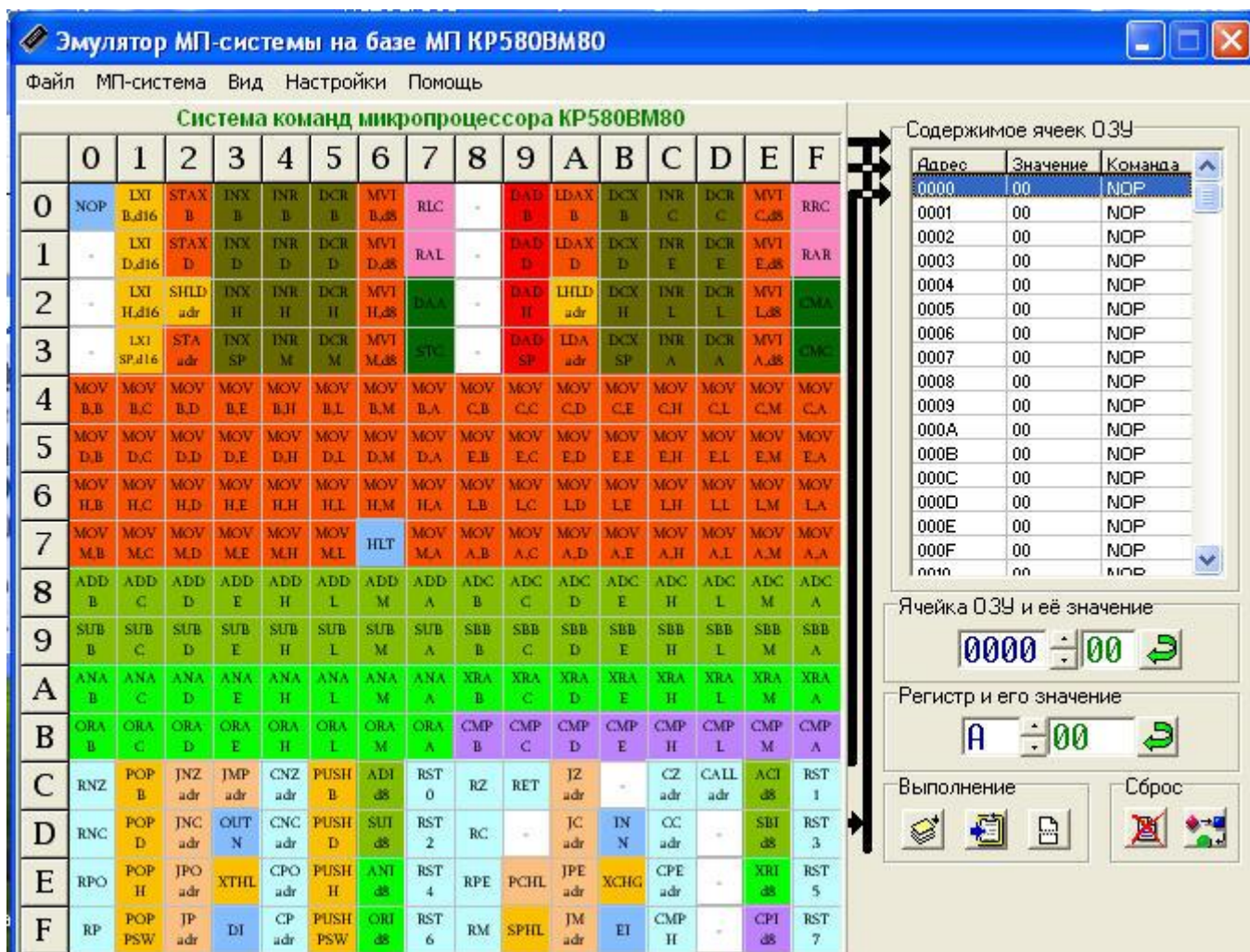


Рис 4 Система команд микропроцессора КР 580

Задания:

1. Изучите предложенное вам задание для составления программы.
2. Запустите программу «Эмулятор микропроцессора КР580» с помощью ярлыка на рабочем столе
3. Изучите правила работы с эмулятором (меню «Помощь»)
4. Изучите команды ассемблера для микропроцессора КР580 (файл на рабочем столе компьютера или распечатка).
4. Составьте программу для выполнения задания на языке Ассемблер
5. Введите программу в ячейки и регистры эмулятора процессора
6. Запустите пошаговый режим выполнения программы. Проверьте выполнение программы и, если необходимо, внесите поправки.
7. Пригласите преподавателя, запустите программу и продемонстрируйте полученный результат