

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 № 11 от 23.06.2017
Председатель цикловой комиссии:

Sh (Kashanov)

УТВЕРЖДАЮ

Начальник УМО

А.В. Калько А.В. Калько
«23» 06 2017г.

МЕТОДИЧЕСКИЕ УКАЗАНИЯ
по организации и проведению лабораторных работ

По учебной дисциплине

ОП.12. МИКРОПРОЦЕССОРЫ И МИКРОПРОЦЕССОРНЫЕ СИСТЕМЫ

Специальность: 09.02.02 Компьютерные сети

Разработчик: Зайцев В.А.

2017г

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Методические указания по организации и проведению лабораторных работ разработаны в соответствии с рабочей программой учебной дисциплины ОП.12. Микропроцессоры и микропроцессорные системы 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:**

- Работать с микропроцессорными системами;
- Программировать микропроцессорные системы;

знать:

- Назначение, функции, характеристики и состав микропроцессорных систем;
- Системы команд, особенности организации системы прерываний микропроцессорных систем;
- Организацию памяти и доступа к ней.

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 3 | Принимать решение в стандартных и нестандартных ситуациях и нести за них ответственность |
| ОК 4 | Осуществлять поиск и использование информации, необходимой для эффективного выполнения профессиональных задач, профессионального и личностного развития |
| ОК 5 | Использовать информационно-коммуникационные технологии в профессиональной деятельности |
| ОК 6 | Работать в коллективе и команде, эффективно общаться с коллегами, руководством, потребителями |
| ОК 7 | Брать на себя ответственность за работу членов команды (подчиненных), за результат выполнения заданий |
| ОК 8 | Самостоятельно определять задачи профессионального и личностного развития, заниматься самообразованием, осознанно планировать повышение квалификации |
| ОК 9 | Ориентироваться в условиях частой смены технологий в профессиональной деятельности |

ПК 1.2. Осуществлять выбор технологии, инструментальных средств и средств вычислительной техники при организации процесса разработки и исследования объектов профессиональной деятельности

ПК 3.6 Выполнять замену расходных материалов и мелкий ремонт периферийного оборудования, определять устаревшее оборудование и программные средства сетевой инфраструктуры.

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 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 1.3. Логическая структура микропроцессора	Лабораторная работа № 1 Исследование основных логических элементов	2	Технические характеристики логических микросхем	Умение определять электронные компоненты по маркировке; разрабатывать схемы на логических элементах	ОК 2, ОК 3 ОК 4, ОК 5, ПК 1.2. , ПК 3.6
Тема 1.3. Логическая структура микропроцессора	Лабораторная работа № 2 Исследование триггеров	2	Технические данные микросхем триггеров. Входные и выходные сигналы триггеров	Применение триггеров в электронных схемах	ОК 2, ОК 3 ОК 4, ОК 5, ПК 1.2. , ПК 3.6
Тема 1.3. Логическая структура микропроцессора	Лабораторная работа № 3 Исследование регистров	2	Назначение и характеристики регистров. Выпускаемые микросхемы регистров.	Выполнять запись двоичных кодов в регистры. Выполнять сдвиг кода в последовательном регистре.	ОК 2, ОК 3 ОК 4, ОК 5, ПК 1.2. , ПК 3.6
Тема 1.3. Логическая структура микропроцессора	Лабораторная работа №4 Исследование счетчиков импульсов	2	Структура и технические характеристики счётчиков импульсов	Проектирование счетчиков с изменяемым модулем счета	ОК 2, ОК 3 ОК 4, ОК 5, ПК 1.2. , ПК 3.6
Тема 1.3. Логическая структура микропроцессора	Лабораторная работа 5 Исследование дешифратора	2	Характеристики дешифратора. Выпускаемые микросхемы дешифраторов	Применять дешифратор в различных цифровых устройствах	ОК 2, ОК 3 ОК 4, ОК 5, ПК 1.2. , ПК 3.6
Тема 1.3. Логическая	Лабораторная работа №6	2	Характеристики мультиплексоров.	Тестировать и применять	ОК 2, ОК 3 ОК 4, ОК 5,

структура микропроцессора	Исследование мультиплексоров и демультимплексоров		Выпускаемые микросхемы	мультиплексоры в различных цифровых устройствах	ПК 1.2. , ПК 3.6
Тема 1.3. Логическая структура микропроцессора	Лабораторная работа 7 Исследование преобразователя кодов	2	Назначение и характеристики преобразователей кода	Применять преобразователи кода в системах вывода информации	ОК 2, ОК 3 ОК 4, ОК 5, ПК 1.2. , ПК 3.6
Тема 1.3. Логическая структура микропроцессора	Лабораторная работа 8 Исследование сумматора	2	Технические характеристики полусумматоров и сумматоров	Применять сумматоры в составе типовых микросхем	ОК 2, ОК 3 ОК 4, ОК 5, ПК 1.2. , ПК 3.6
Тема 1.3. Логическая структура микропроцессора	Лабораторная работа 9 Исследование арифметико-логического устройства (АЛУ)	2	Назначение и структура АЛУ. Входные и выходные коды АЛУ	Умение определять выходной сигнал АЛУ при различных входных сигналах	ОК 2, ОК 3 ОК 4, ОК 5, ПК 1.2. , ПК 3.6
Тема 2.7 Микросхемы ОЗУ и ПЗУ	Лабораторная работа 10 Исследование микросхемы оперативной памяти	2	Структура и технические характеристики микросхем оперативной памяти Организация памяти и доступа к ней.	Умение выполнять запись и чтение информации в ячейках микросхемы оперативной памяти	ОК 2, ОК 3 ОК 4, ОК 5, ПК 1.2. , ПК 3.6
Тема 2.7 Микросхемы ОЗУ и ПЗУ	Лабораторная работа 11 Исследование микросхемы ПЗУ	2	Принцип хранения и записи информации в микросхеме ПЗУ	Умение использовать программатор для чтения и перезаписи микросхем ПЗУ	ОК 2, ОК 3 ОК 4, ОК 5, ПК 1.2. , ПК 3.6
Тема 3.2 Команды ассемблера для МП Intel	Лабораторная работа 12. Изучение структуры микропроцессорной системы. Эмулятор микропроцессора КР580.	2	Структура и принцип работы микропроцессора. Назначение, функции, характеристики и состав микропроцессорных систем	Умение работать с эмулятором микропроцессорной системы Работать с микропроцессорным и системами;	ОК 2, ОК 3 ОК 4, ОК 5, ПК 1.2. , ПК 3.6
Тема 3.2 Команды ассемблера для МП Intel	Лабораторная работа №13 Программирование процессора КР 580. Выполнение арифметических операций	2	Команды языка Ассемблер, формат и структура команд. Системы команд, особенности организации системы прерываний микропроцессорных систем;	Умение составлять программы на языке Ассемблер и проверять работоспособность с помощью эмулятора Работать с микропроцессорным и системами	ОК 2, ОК 3 ОК 4, ОК 5, ПК 1.2. , ПК 3.6
Тема 3.2 Команды ассемблера для МП Intel	Лабораторная работа №14 Программирование процессора КР 580 Разветвляющиеся программы	4	Команды языка Ассемблер, формат и структура команд. Системы команд, особенности организации системы прерываний микропроцессорных	Умение программировать микропроцессорные системы	ОК 2, ОК 3 ОК 4, ОК 5, ПК 1.2. , ПК 3.6

			систем;		
Тема 4.4 Таймеры	Лабораторная работа №15 Исследование таймера	2	Назначение, электрическая схема и характеристики микросхемы таймера	Умение работать с микропроцессорным и системами собрать устройство с заданным временем включения и выключения	ОК 2, ОК 3 ОК 4, ОК 5, ПК 1.2. , ПК 3.6

Содержание лабораторных работ охватывает весь круг умений и компетенций, на формирование которых направлена учебная дисциплина.

ПЕРЕЧЕНЬ ЛАБОРАТОРНЫХ РАБОТ

Лабораторная работа № 1	Исследование основных логических элементов
Лабораторная работа № 2	Исследование триггеров
Лабораторная работа № 3	Исследование регистров
Лабораторная работа № 4	Исследование счетчиков импульсов
Лабораторная работа № 5	Исследование дешифратора
Лабораторная работа № 6	Исследование мультиплексора
Лабораторная работа № 7	Исследование преобразователя кодов
Лабораторная работа № 8	Исследование сумматора
Лабораторная работа № 9	Исследование арифметико- логического устройства
Лабораторная работа № 10	Исследование микросхемы оперативной памяти
Лабораторная работа № 11	Исследование микросхемы ПЗУ
Лабораторная работа № 12	Изучение структуры микропроцессорной системы. Эмулятор микропроцессора КР580
Лабораторная работа № 13	Программирование процессора КР 580. Выполнение арифметических операций
Лабораторная работа № 14, 15	Программирование процессора КР 580 Разветвляющиеся программы
Лабораторная работа № 16	Исследование таймера

КРИТЕРИИ ОЦЕНИВАНИЯ ЛАБОРАТОРНЫХ 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.

Тема: Исследование основных логических элементов

Цель: Исследование основных логических элементов и простейших комбинационных устройств на их основе. Изучение основ и законов алгебры логики. Ознакомление с элементной базой ТТЛ. выработка навыков составления и исследования схем, реализующих заданную логическую функцию, и простейших комбинационных устройств; составление и экспериментальная проверка таблиц истинности для каждого устройства.

Перечень необходимых средств обучения:

Стенд ОАВТ , комплект 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.

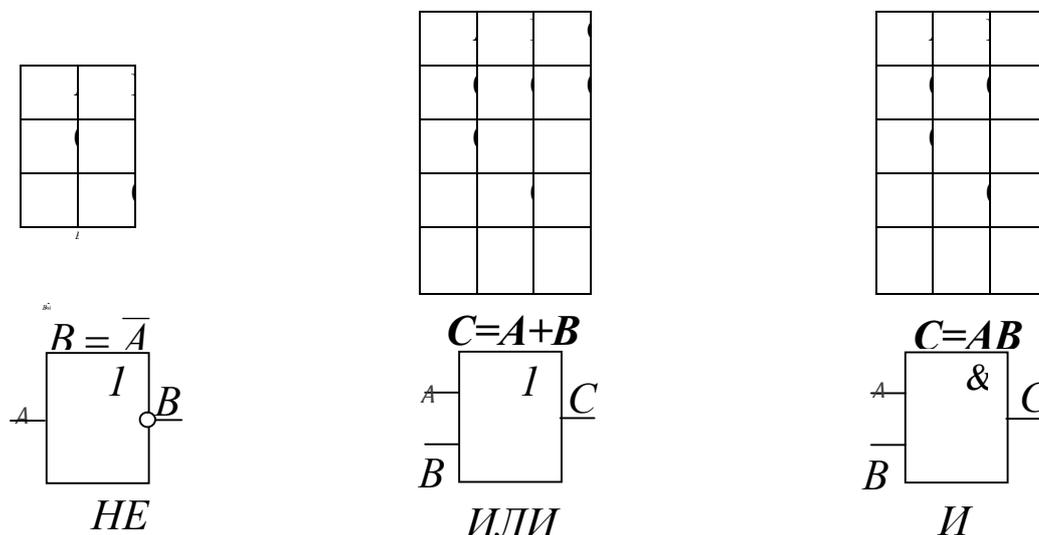


Рис. 1

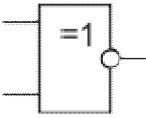
При помощи рассмотренных логических элементов в соответствии с правилами и законами алгебры логики могут быть реализованы сложные логические функции.

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

$A \ B \ A \leftrightarrow B$

0 0 1

0 1 0

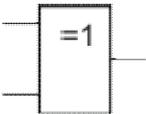
1 0 0

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](#).

$A \ B \ f(AB)$

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.

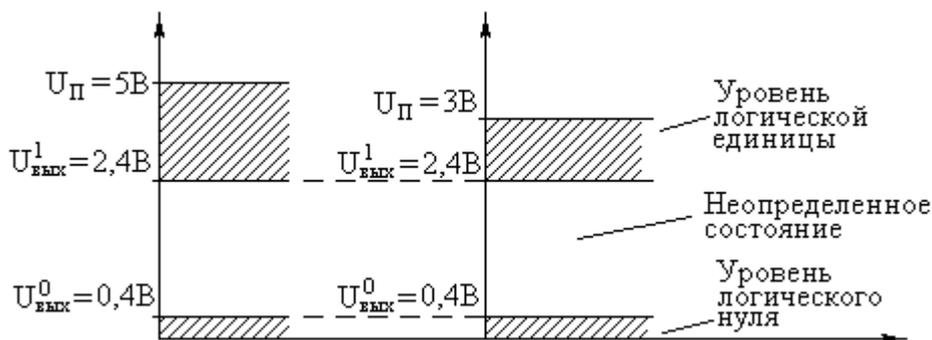
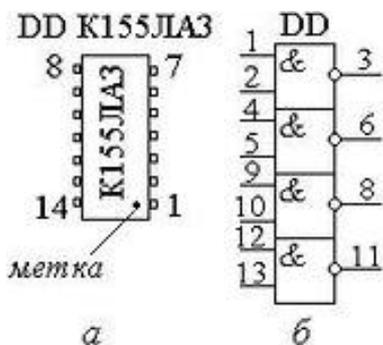


Рис 1 Уровни «1» и «0» для различных серий микросхем

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

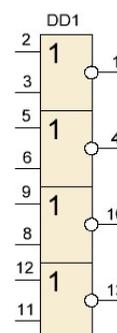


1	Номинальное напряжение питания	5 В \pm 5 %
2	Выходное напряжение низкого уровня	не более 0,4 В
3	Выходное напряжение высокого уровня	не менее 2,4 В
4	Напряжение на антизвонном диоде	не менее -1,5 В
5	Входной ток низкого уровня	не более -1,6 мА
6	Входной ток высокого уровня	не более 0,04 мА
7	Входной 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

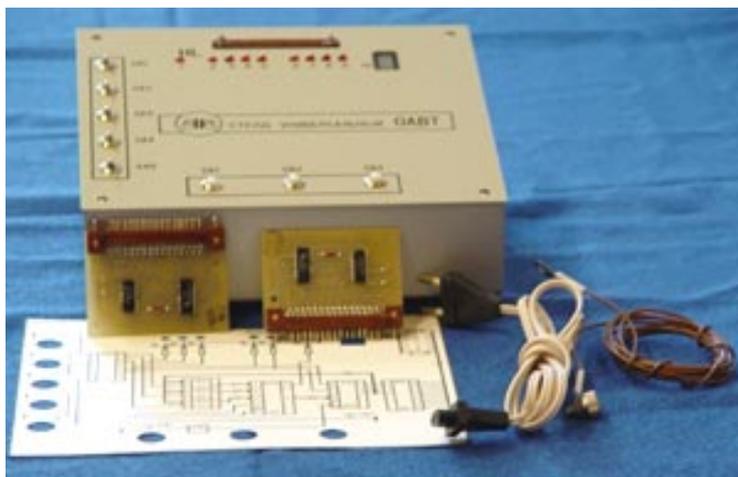


Рис 1 Стенд ОАВТ с принадлежностями

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 \bar{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, T- 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 К155ТМ2

Задания:

1. Установите в разъем плату П. Установите на пульт плату П – 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 \bar{Q} индицируются светодиодами HL3 HL4. Зарисуйте схему RS триггера. Данные об используемых микросхемах используйте из лабораторной работы 2.
2. Повторите измерения для карт П – 2 и П -3. На этих платах RS триггер реализован на других элементах.
3. Установите в пульт карту П – 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 П – 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 T-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 T-триггер
6. Характеризуйте синхронные и асинхронные триггеры

Лабораторная работа №3

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 последовательное 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.

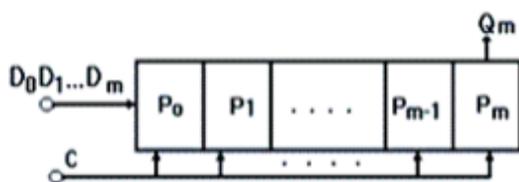
- ✓ Состоит из запоминающих элементов- триггеров
- ✓ В каждом триггере хранится цифра определенного разряда числа
- ✓ Может выполнять сдвиг принятого слова вправо и влево
- ✓ Может преобразовать двоичный код из прямого в обратный (замена 0 на 1 и 0 на 1)
- ✓ Выполнять логические сложение и умножение

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).



Последовательный

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 синхроимпульсов на входы «С» триггеров не меняет их состояния.

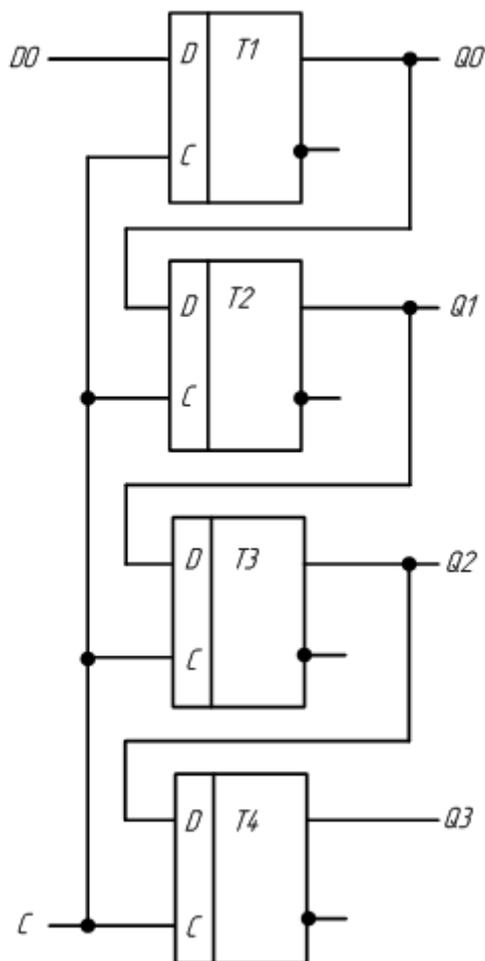


Рисунок 2 Схема последовательного регистра

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.

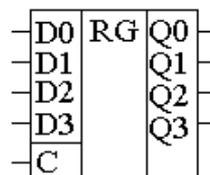
При поступлении второго синхроимпульса логическая «1» с выхода первого триггера, запишется во второй триггер, и в результате происходит сдвиг первоначально записанной «1» с триггера T1 в триггер T2, из триггера T2 в триггер T3 и т.д. Таким образом,

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

Четырёхразрядный параллельный регистр. Условное графическое обозначение



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. Для записи **n** – разрядного слова необходимо **n** триггеров. Перед записью информации все триггеры устанавливаются в нулевое состояние подачей положительного единичного импульса по шине УСТ. "0".

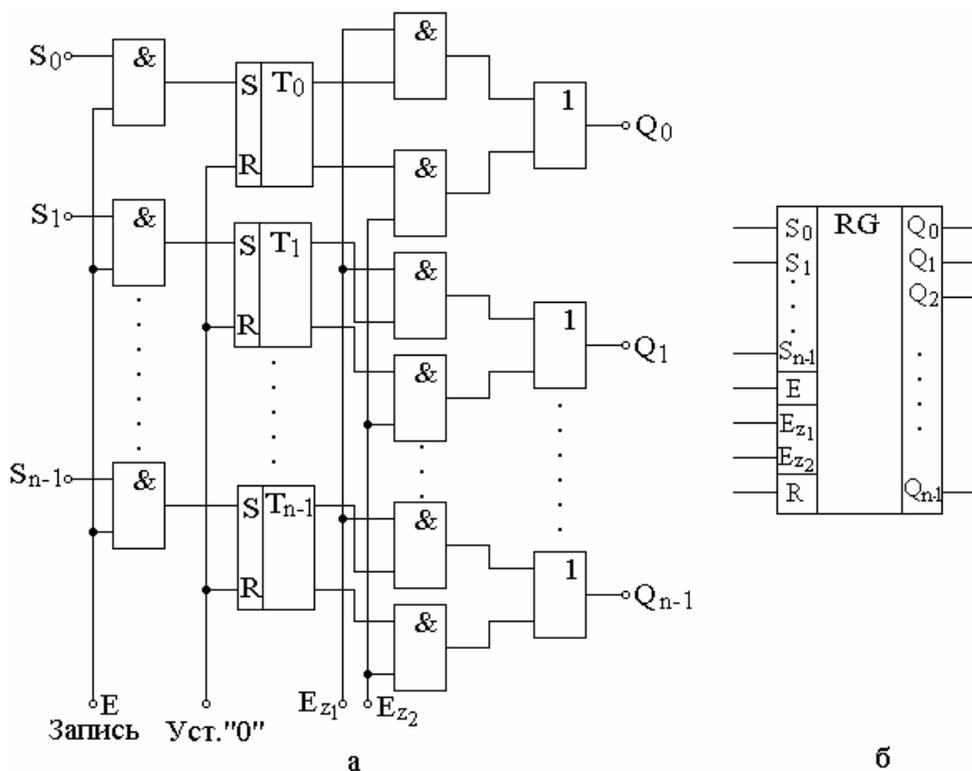


Рис 1 Регистр на RS триггерах

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.

Вывод информации из регистра может осуществляться в прямом и обратном коде через схемы совпадения, управляемые сигналами E_{z1} и E_{z2} .

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 сдвига

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 TT_{n-1} пропадает из регистра, а на вход первого TT_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.

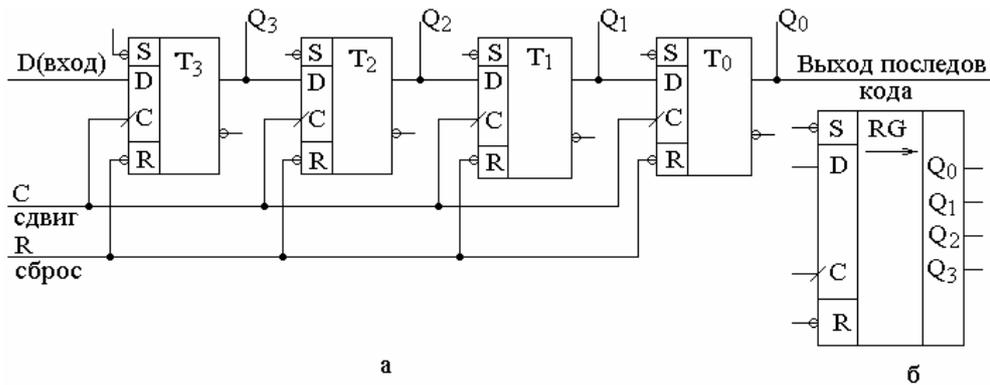


Рис 2 Схема четырехразрядного регистра сдвига вправо на синхронных D-триггерах с прямым динамическим входом С. Стрелка под обозначением регистра RG в основном поле показывает направление сдвига информации.

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».

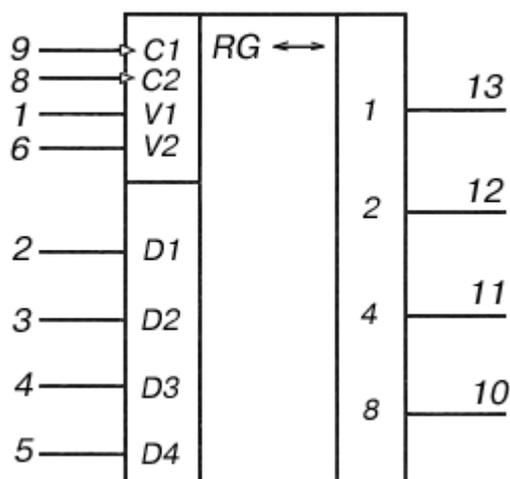
В лабораторной работе проводится исследование параллельного, последовательного и универсального регистров реализованного на микросхеме K155IP1. Электрическая схема и характеристики микросхемы приведены ниже

Микросхемы K155IP1, KM155IP1 (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 C1 и C2. 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 K155IP1, KM155IP1 (7495). Если на вход PE дается напряжение высокого уровня, разрешается работа тактовому входу C2. 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 C1. 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 K155IP1, KM155IP1 (7495) PE можно менять только если на обоих тактовых входах уровни низкие. Однако если на входе C1 напряжение низкого уровня, перемена сигнала на входе PE от низкого уровня к высокому не меняет состояния выходов.

Обычный вариант микросхемы K155IP1, KM155IP1 (7495) имеет ток потребления 63 мА, Максимальная, тактовая частота 25 МГц, Возможные режимы работы регистра K155IP1, KM155IP1 (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 плату 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 № 4

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, T-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 счёта)

Электрическая схема суммирующего счетчика на Т триггерах приведена на рис 1

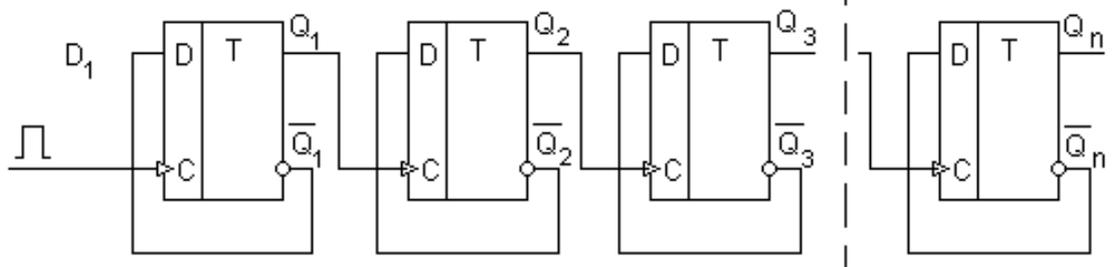


Рис1 Электрическая схема суммирующего счетчика на Т- триггерах

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

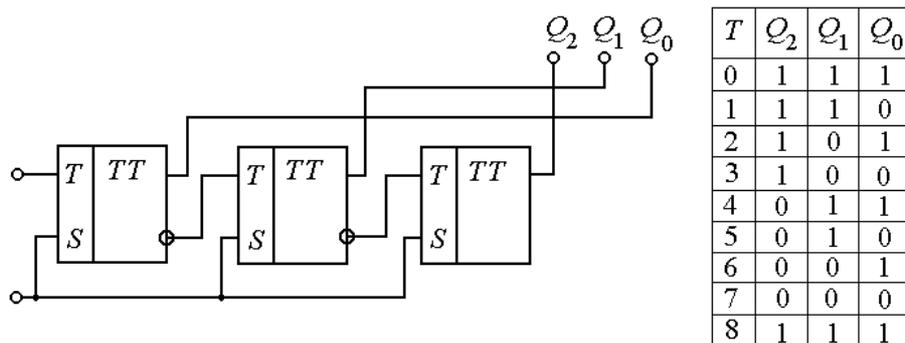
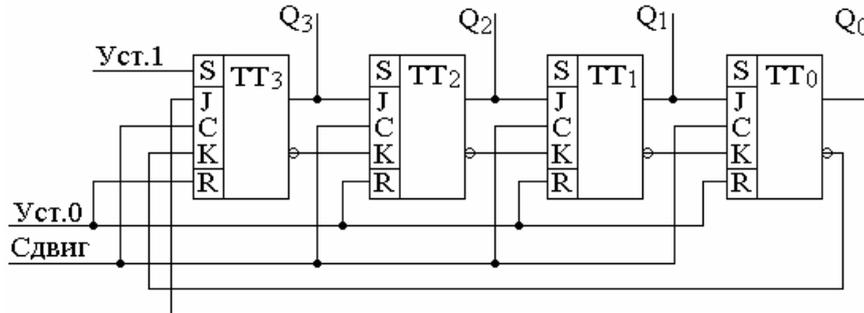


Рис 2 Вычитающий счетчик. Соединение триггеров выполняется по инверсному выходу.

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 Q3.



Задания

1. Установите в разъем 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	HL 2	HL 3	HL 4
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 №5

Тема: Исследование дешифратора

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

Перечень необходимых средств обучения:

Стенд ОАВТ, платы с комбинационными устройствами, карты для стенда.

Краткие теоретические сведения

В лабораторной работе исследуется работа дешифратора на интегральной микросхеме (ИС) K155ИД4 и K155 ИД1.

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

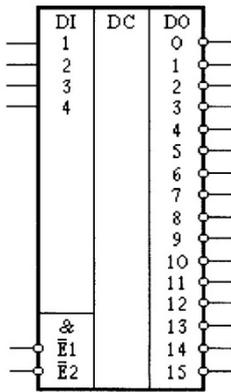
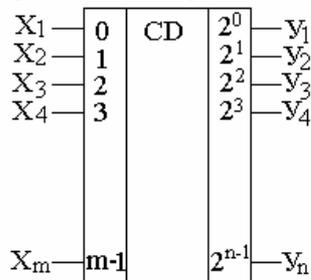


Рис 1 Микросхема дешифратора

Шифратором называется устройство с t 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

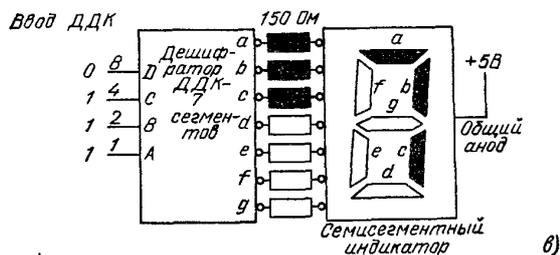
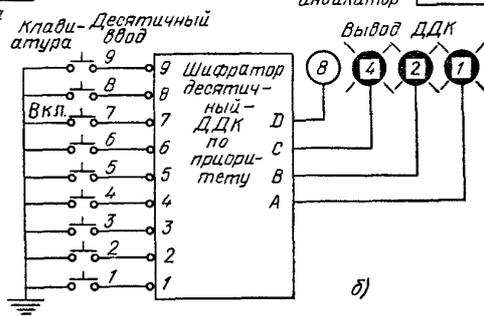
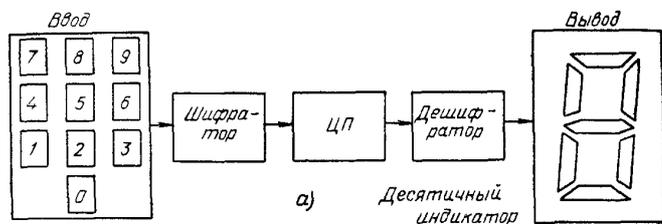


Рис 3 Применение шифратора и дешифратора в калькуляторе.

Задания

1. В выключенном состоянии стенда установите в разъем стенда OABT плату 4.
 2. Установите на стенд шаблон IV-1 Дешифратор-демультиплексор.
 3. Включите стенд OABT 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								

Лабораторная работа № 6

Тема: Исследование мультиплексора

Цель: Исследование микросхемы мультиплексора

Перечень необходимых средств обучения:

Краткие теоретические сведения

Мультиплексор (от англ. multiplex – многократный) – это устройство, предназначенное для коммутации в желаемом порядке информации, поступающей с нескольких входных шин на одну выходную.

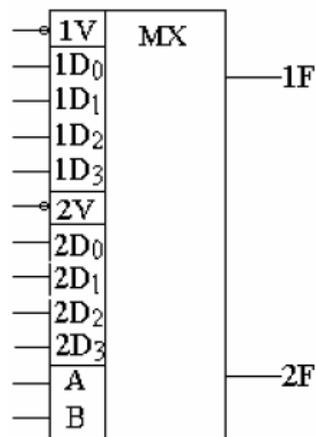
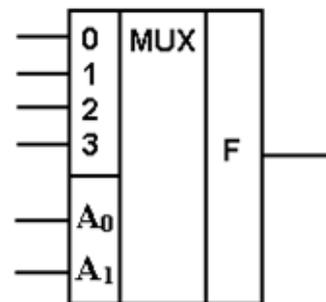
Мультиплексоры обладают двумя группами входов: информационными и управляющими

- К **информационным входам** подводятся информационные каналы.
- К **управляющим входам** относятся адресные и разрешающие (стробирующие) входы. На адресные входы подается кодовая комбинация, в соответствии с которой тот или иной информационный вход подключается к выходу. Если мультиплексор имеет *n* адресных входов, то число информационных входов равно 2^n
- **Разрешающий (стробирующий) вход** управляет одновременно всеми информационными входами независимо от состояния адресных входов.

- Запрещающий сигнал на этом входе блокирует действия всего устройства.
- Наличие разрешающего входа расширяет функциональные возможности мультиплексора, позволяя синхронизировать его работу с работой других узлов. Разрешающий вход также используется для наращивания разрядности мультиплексора.

Таблица функционирования мультиплексора

Адрес		Выход
A1	A0	F
0	0	Вход 0
0	1	Вход 1
1	0	Вход 2
1	1	Вход 3



Микросхема мультиплексора 155 КП2

Задания:

1. В выключенном состоянии стенда установите в разъем стенда ОАВТ плату 4.
2. Исследуйте работу мультиплексора. Установите карту IV-3 - на стенде ОАВТ .
3. 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 №7

Тема: Исследование преобразователя кодов

Цель: Исследование режима работы преобразователя кодов. Ознакомление с элементной базой микросхем преобразователей кодов , выработка навыков составления и исследования схем с применением преобразователей кодов.

Перечень необходимых средств обучения:

Стенд ОАВТ, платы с микросхемой преобразователя кодов, карты для стенда.

Краткие теоретические сведения

Кодовой комбинацией (словом) называется совокупность двоичных цифр (нулей и единиц), образующих двоичное число, соответствующее определенному информационному символу.

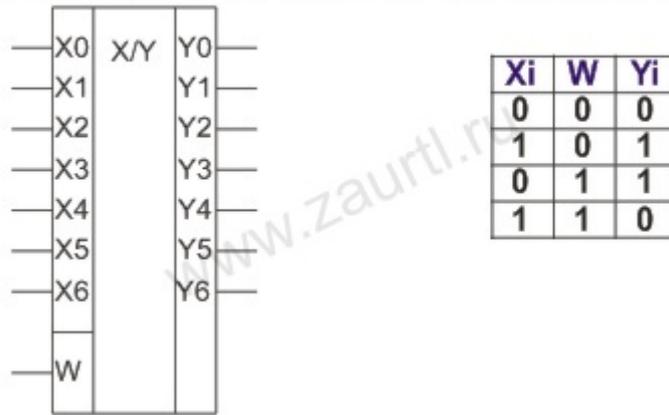
Число двоичных цифр (нулей и единиц) в кодовой комбинации (слове) называется **длиной кодовой комбинации (слова)**. При этом каждая двоичная цифра называется **элементом двоичного кода**.

- Преобразователем кодов или кодопреобразователем называется цифровой узел с m входами и n выходами, преобразующий один параллельный двоичный код в другой.

- Оператор, выполняемый преобразователем кодов, обычно задается в виде таблицы истинности. Каждая переменная Y_i , выходного кода рассматривается как логическая функция всех переменных X входного кода.

Пример преобразователя кодов:

преобразователь кода и таблица его функционирования



При сигнале «1» на входе W выполняется инверсия входных сигналов.

В качестве примера преобразователя кодов, выпускаемых в виде ИС, можно привести схемы, обеспечивающие преобразование информации из двоичного в двоично-десятичный код. Частным случаем преобразователей кода являются шифраторы и дешифраторы.

В лабораторной работе исследуется преобразователь кода для семисегментного индикатора на микросхеме памяти 155 PE3 (рис 1).

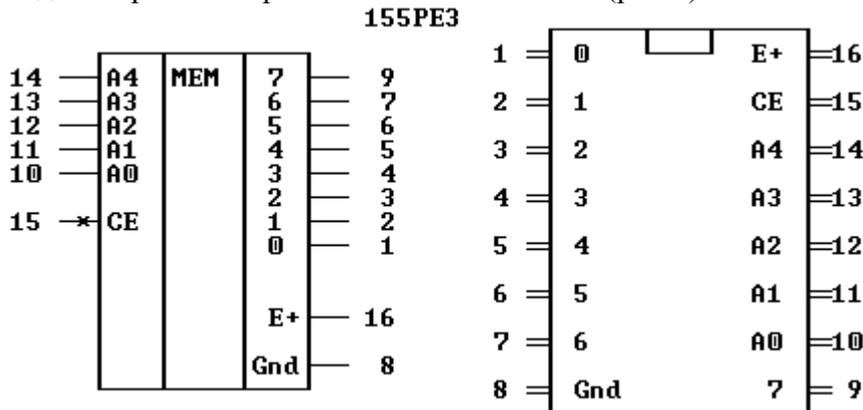


Рис 1 Микросхема 155 PE3

Семисегментный индикатор (рис 2) состоит из светодиодных сегментов. При подаче напряжения на сегмент он светится. Таким образом, подключая различные сегменты можно индицировать различные цифры и буквы.

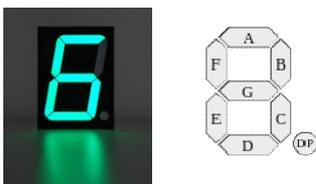


Рис. 2 Семисегментный индикатор

Таблица истинности преобразователя семисегментного кода.

Отображаемые цифры и буквы	Входная комбинация (двоичный код)				Выходная комбинация (семисегментный код)							
	X3	X2	X1	X0	g	f	e	d	c	b	a	
0	0	0	0	0	0	1	1	1	1	1	1	1
1	0	0	0	1	0	0	0	0	1	1	0	0
2	0	0	1	0	1	0	1	1	0	1	1	1
3	0	0	1	1	1	0	0	1	1	1	1	1
4	0	1	0	0	1	1	0	0	1	1	0	0
5	0	1	0	1	1	1	0	1	1	0	0	1
6	0	1	1	0	1	1	1	1	1	0	0	1
7	0	1	1	1	0	0	0	0	1	1	1	1
8	1	0	0	0	1	1	1	1	1	1	1	1
9	1	0	0	1	1	1	0	1	1	1	1	1
A	1	0	1	0	1	1	1	0	1	1	1	1
B	1	0	1	1	1	1	1	1	1	0	0	0
C	1	1	0	0	0	1	1	1	0	0	0	1
D	1	1	0	1	1	0	1	1	1	1	1	0
F	1	1	1	0	1	1	1	1	0	0	0	1
G	1	1	1	1	1	1	1	0	0	0	0	1

Способ подключения микросхемы 155PE3 для преобразования кода приведен на рис 3

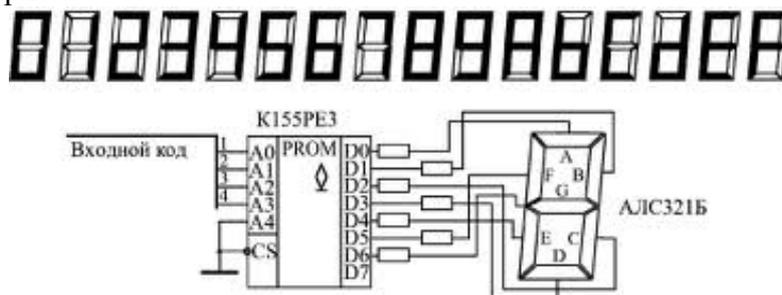


Рис 3 Подключение микросхемы 155PE3 к семисегментному индикатору

Задания

1. Соедините комплектующие элементы на плате для беспаявого монтажа. Схема прилагается. В качестве преобразователя кодов используется микросхема 514ИД2. Для индикации выходных сигналов используется микросхема АЛС324Б.
2. Подавайте на вход 514 ИД2 двоичные коды от 0000 до 1001 (10 кодов)
3. Заполните таблицу истинности для преобразователя кодов.

Лабораторная работа №8

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.

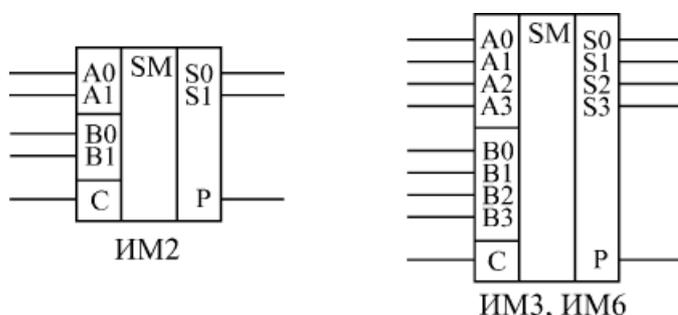
Сложение чисел в последовательных сумматорах осуществляется поразрядно, последовательно во времени. В сумматорах параллельного действия сложение всех разрядов многоразрядных чисел происходит одновременно.

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 В в регистр может осуществляться как в параллельном (с тумблеров SA1 – SA4), так и в последовательном (с кнопки SB2) режимах. Сигнал начального переноса P₀ подается с кнопки SB3 через инвертор D2.3 , имеющийся в основном блоке. Индикация входных операндов А и В осуществляется двоичным восьмиразрядным индикатором (индикаторы HL2 – HL5 для операнда А и индикаторы HL6- HL9 для операнда В). 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 III-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 В. 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

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

N₂- вычитаемое. Определяем дополнительный код этого числа. Для этого:

✓ Инвертируем N₂ (Заменяем «1» на «0» и «0» на «1»). Получаем 0110

✓ Добавляем «1» к младшему разряду. Получаем 0111

$$N_{2\text{доп}} = 0111$$

Выполняем сложение

$$\begin{array}{l} \text{Слагаемое } N_1 \quad \quad 1100_2 \quad \quad \text{Проверка } 1100_2 = 12_{10} \quad 1001_2 = 9_{10} \\ \text{Слагаемое } N_{2\text{доп}} \quad 0111_2 \quad \quad 12_{10} - 9_{10} = 3_{10} = 0011_2 \\ \text{Сумма } N_1 + N_2 \quad \rightarrow 1\ 0011_2 \end{array}$$

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 0011₂

Выполните вычитание чисел на стенде ОАВТ

$$1101 - 1001$$

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 № 9

Тема: Исследование 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:

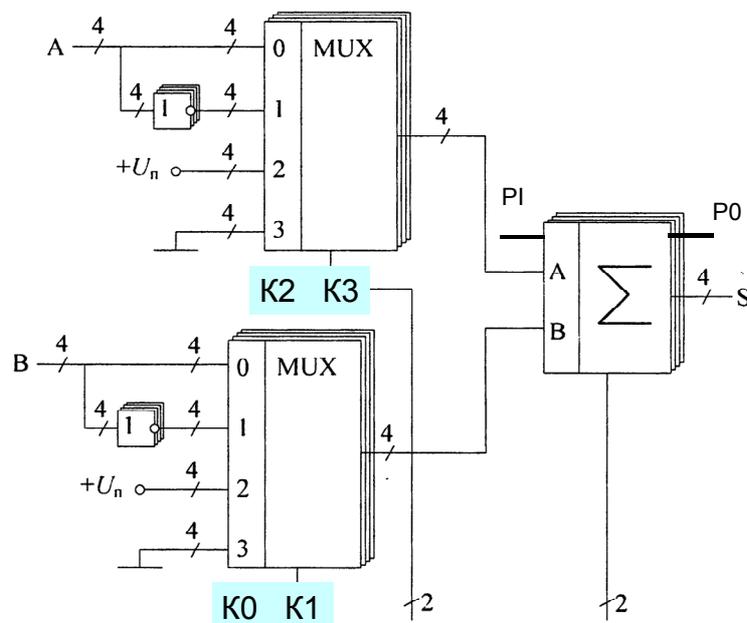
Стенд для изучения АЛУ.

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

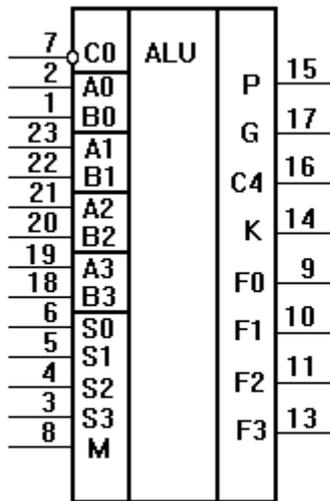
Для данного устройства можно составить таблицу входных и выходных сигналов

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 К155ИП3 применяется специальный стенд. Управление работой микросхемы (подача сигналов «0» и «1» на входы S0 - S3 и набор значения операндов А и В) 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
3. Объясните принцип работы одноразрядного сумматора с тремя входами.
4. Объясните принцип работы многоразрядного сумматора

Лабораторная работа № 10

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

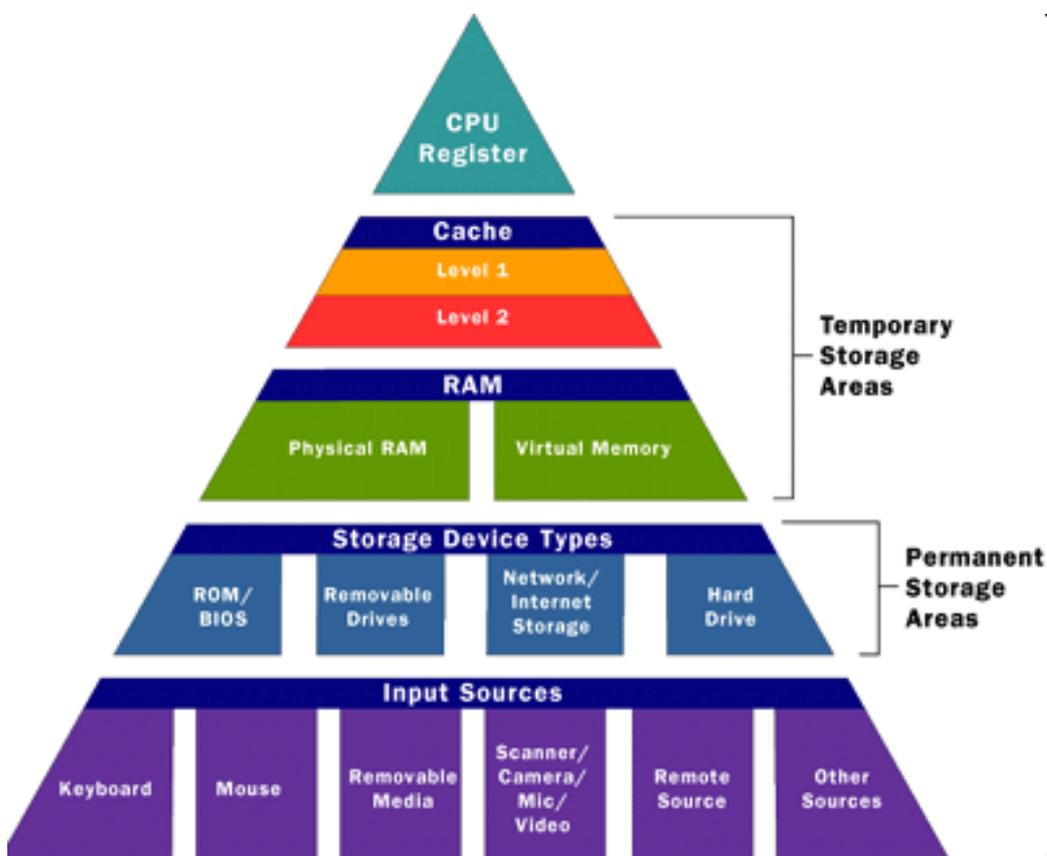


Рис1 Иерархия оперативной памяти

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*

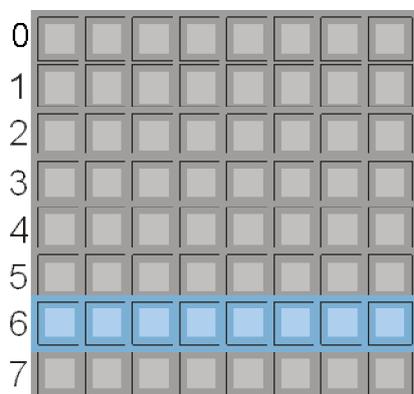
Цикл (чтения или записи)- минимально допустимый интервал между последовательными чтениями или записями.

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 b_{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

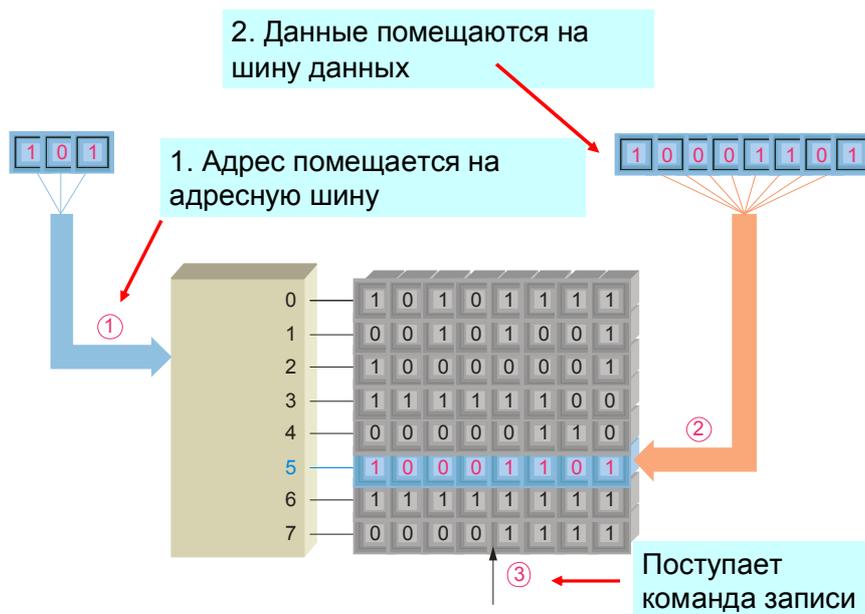
Read Enable (RE) и Write Enable (WE) Эти сигналы посылаются процессором для управления перемещением данных из памяти или в память.

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

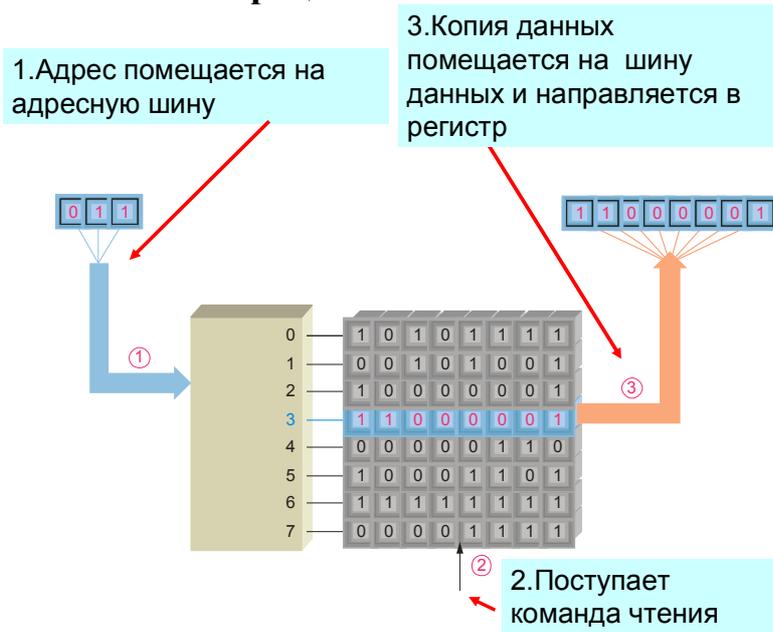
Output Enable (OE) активен во время операции чтения. Он подключает память к шине данных.

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 x 4 бит (16 ячеек по 4 бита (элемента) в ячейке). В качестве примера на рис 2 приведена структура микросхемы оперативной памяти

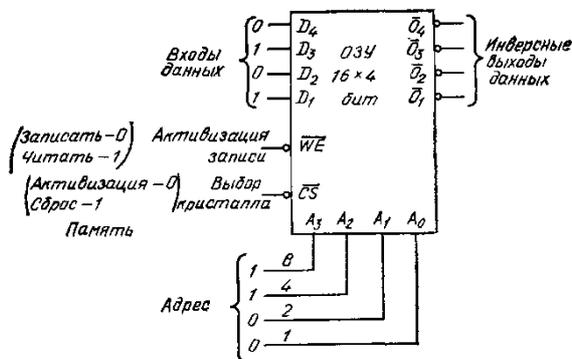


Рис. 3.25. Логическая схема ОЗУ 16x4 бит

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 1100_2
(12_{10})

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

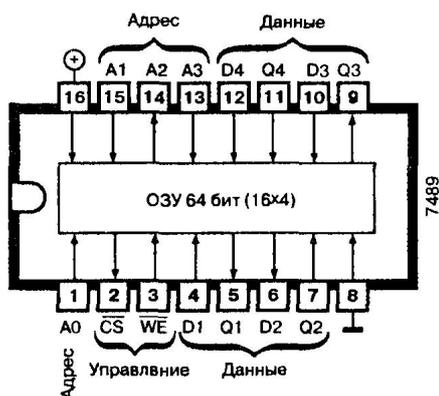


Рис3 Микросхема 7489

Задания:

1. Ознакомьтесь с техническими характеристиками микросхемы оперативной памяти К155PY2 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

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

Лабораторная работа № 11

Тема: Исследование микросхемы 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

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 x 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

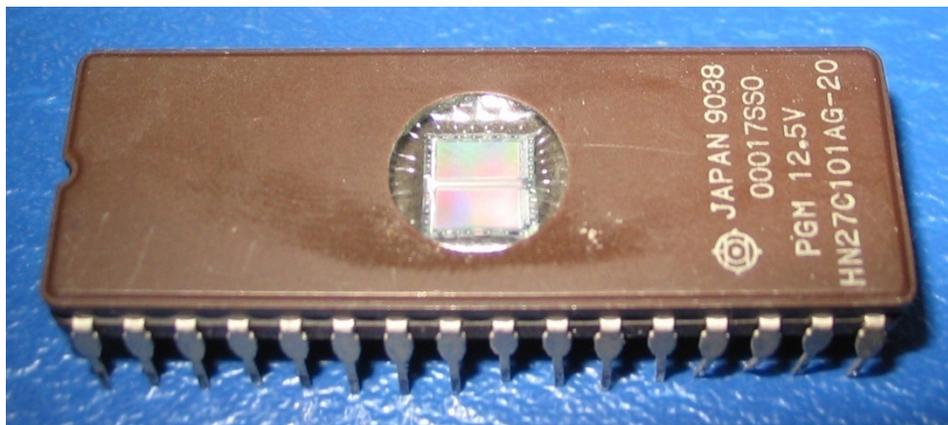


Рис.1 Микросхема EPROM с ультрафиолетовым стиранием.

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

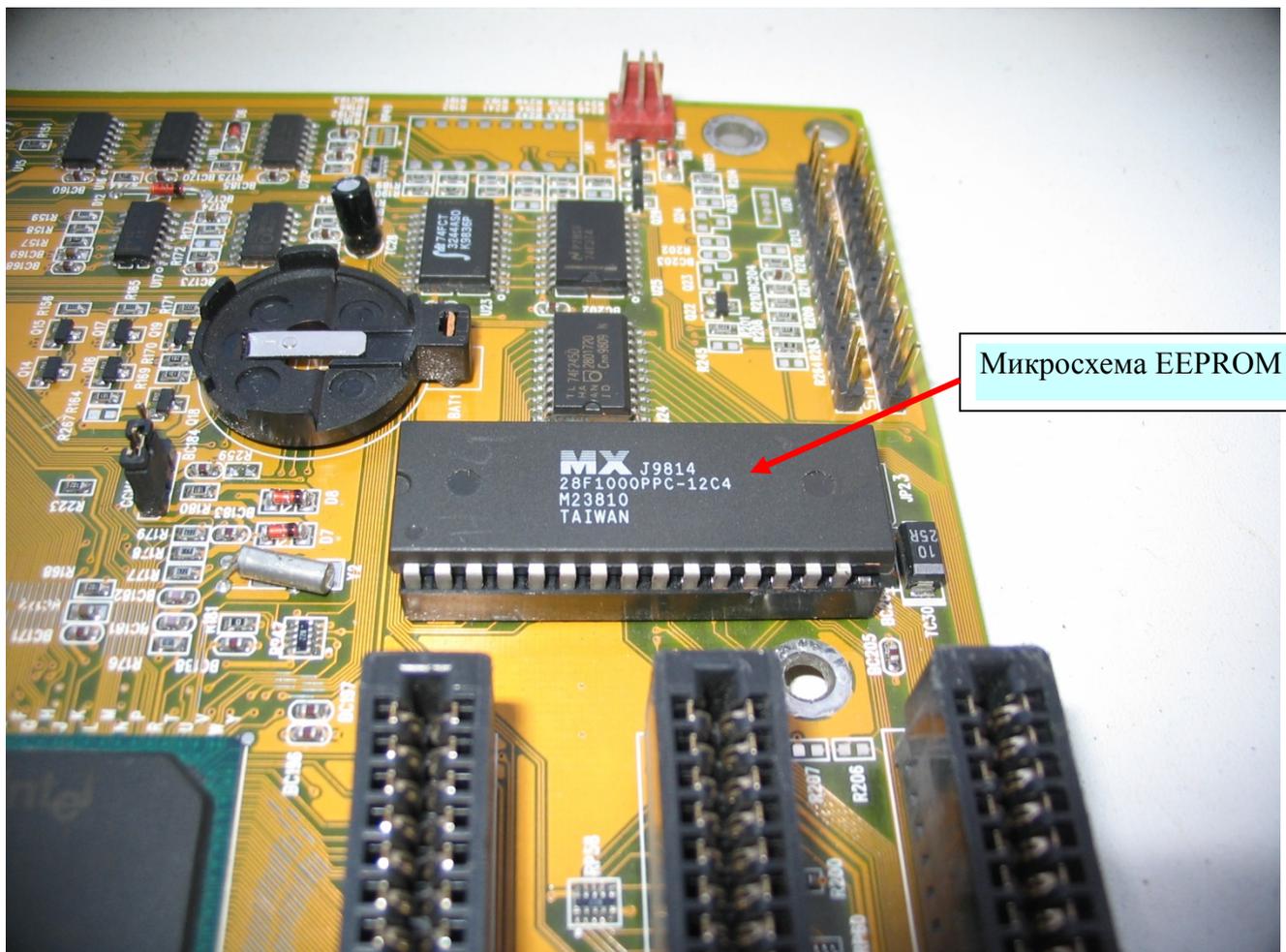


Рис 2 Микросхема EEPROM

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															
10	FF															
20	FF															
30	FF															
40	FF															
50	FF															
60	FF															
70	FF															
80	FF															
90	FF															
A0	FF															
B0	FF															
C0	FF															
D0	FF															
E0	FF															
F0	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.

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 Agies 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 СОМ1-СОМ4. 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 выравнивая 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.

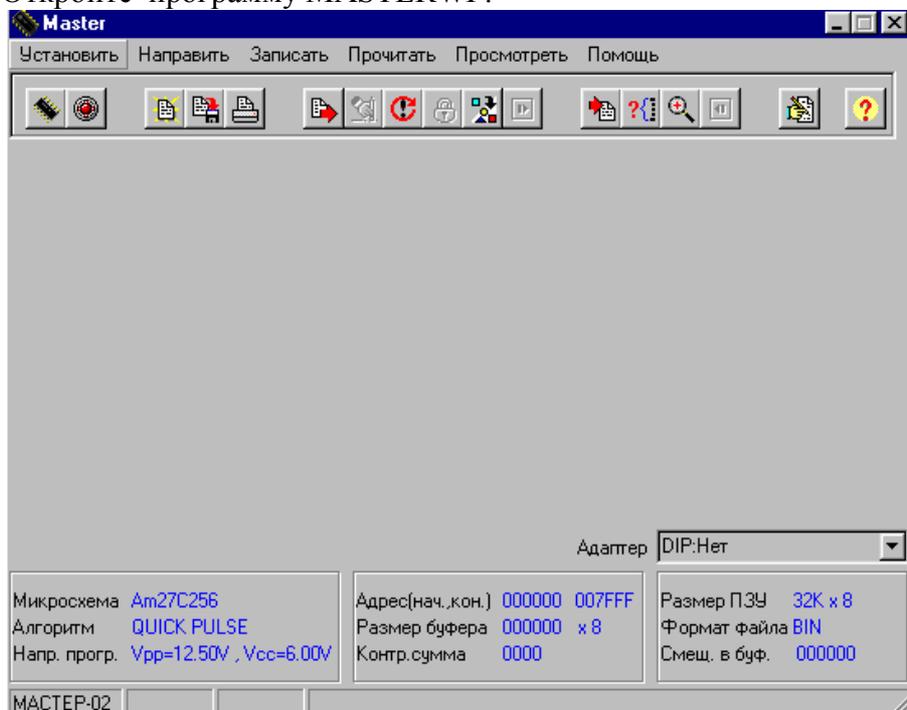
Программа предоставляет пользователю следующие возможности:

- 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.

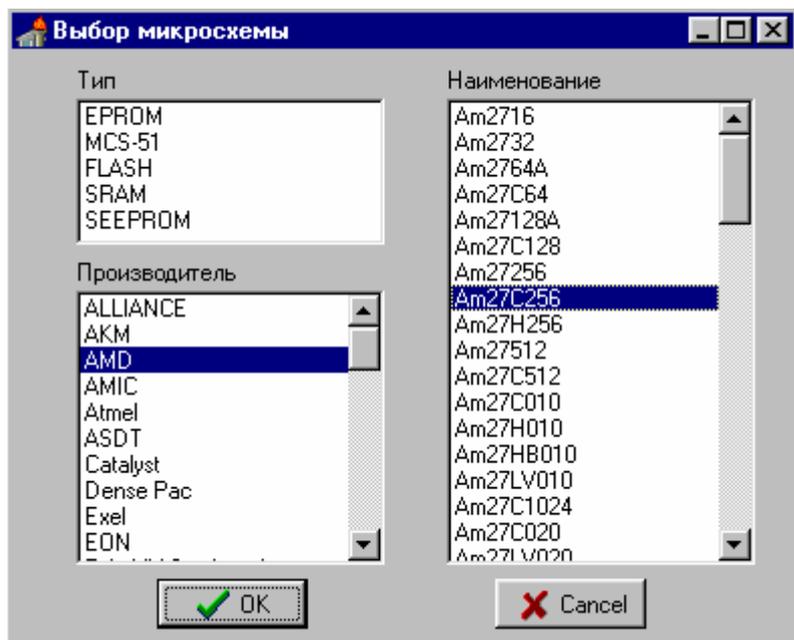
В лабораторной работе необходимо прошить микросхему ПЗУ, то есть перепрограммировать ее. Чтобы информацию с одной микросхемы переписать на другую следует выполнить следующие действия.

Вставьте в программатор микросхему, с которой нужно считать информацию.

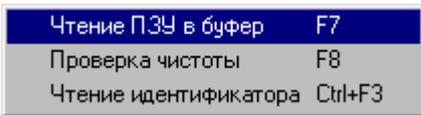
Откройте программу MASTERWP:



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.

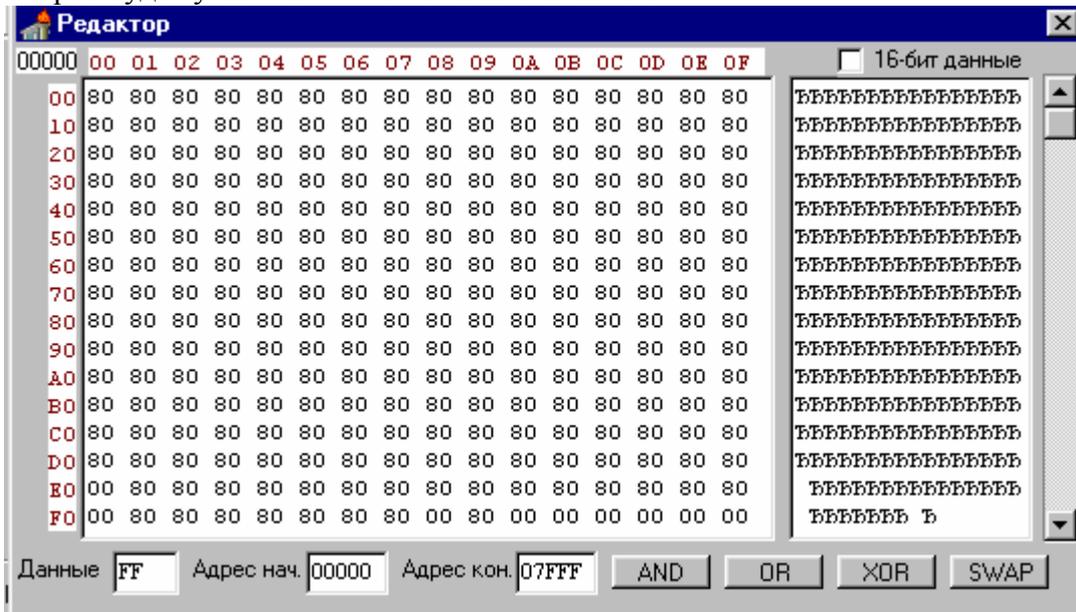


При нажатии меню «просмотр» -> просмотр буфера

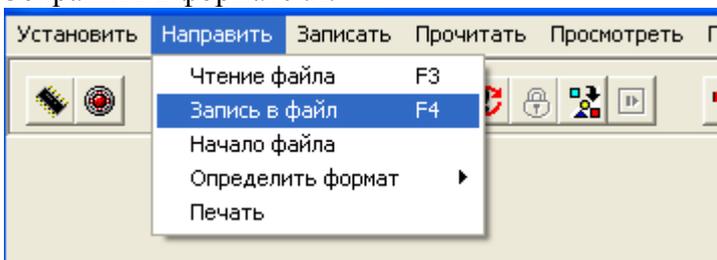


должна появиться следующая картина.

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

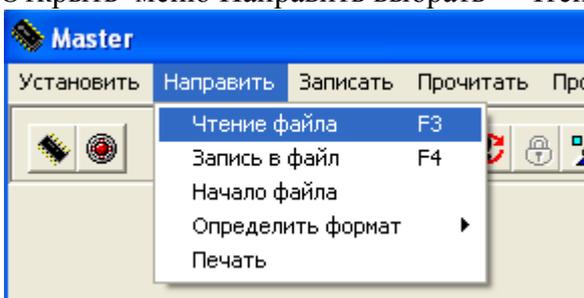


Открыть меню Направить -> Записать в файл
Сохранить в формате txt



Вставить в программатор другую микросхему (которую нужно перепрограммировать)

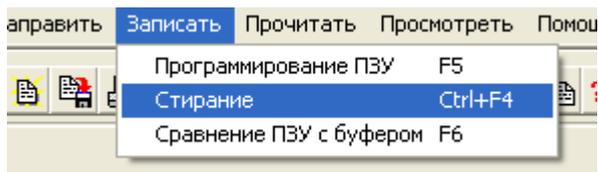
В меню Установить выбрать имя и фирму микросхемы
Открыть меню Направить выбрать ->Чтение файла:



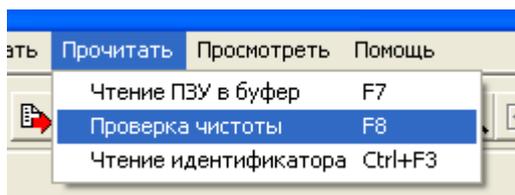
При этом произойдет чтение сохраненного буфера.

Чтобы перепрограммировать ПЗУ необходимо для начала стереть микросхему.

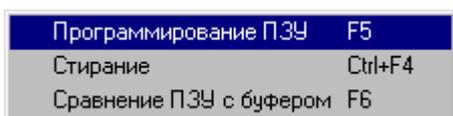
Для этого выбрать меню Запись->стирание.



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.

Выполнение работы

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 № 12

Тема: Изучение структуры микропроцессорной системы.
Эмулятор микропроцессора КР580.

Цель: приобретение навыков работы с эмулятором ассемблера для процессора КР580ВМ80, .

Перечень необходимых средств обучения:

Персональный компьютер.

Программное обеспечение – эмулятор микропроцессорной системы на базе КР580ВМ80.

Методические указания к практическому занятию.

Краткие теоретические сведения

Микропроцессор структурно состоит из двух блоков:

- ✓ Операционного блока
- ✓ Управляющего блока

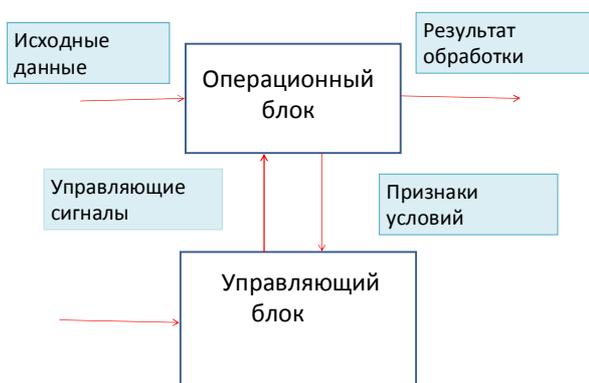


Рис 1 Структурная блок-схема микропроцессора

Операционный блок

Это устройство в котором выполняются операции:

- ✓ Прием и временное хранение исходных данных
- ✓ Преобразование и передача результата обработки.
- ✓ Проверка соответствия результата обработки заранее обусловленным признакам (отрицательные числа, нули, четность и т.п.) и направление результата проверки управляющему блоку

Операционный блок включает в себя: регистры, сумматоры, каналы передачи информации, мультиплексоры для коммутации каналов, шифраторы, дешифраторы и т.д. Ряд этих устройств составляет блок АЛУ (арифметико-логическое устройство)

Управляющее устройство

- ✓ Координирует действия узлов операционного блока.
- ✓ Вырабатывает последовательность управляющих сигналов, вид которых зависит от кода текущей операции и признаков результата предыдущей операции.
- ✓ Под действием управляющих сигналов в узлах операционного устройства выполняются требуемые действия.

Структура МПС является магистрально-модульной. В такой структуре имеется группа магистралей (шин), к которым подключаются различные модули (блоки), обменивающиеся между собой информацией поочередно, в режиме разделения времени. Термин "шины" относится к совокупности цепей (линий), число которых определяет разрядность шины.

- Стандартной является структура МП с шинами адресов ША (англ. АВ - Adress Bus), данных ШД (англ. DB -Data Bus) и управления ШУ (англ. СВ - Control Bus).
- ГТИ- генератор тактовых импульсов
- ССП- регистр слова состояния процессора
- РОН- регистры общего назначения

Упрощенная схема микропроцессора приведена на рис 2

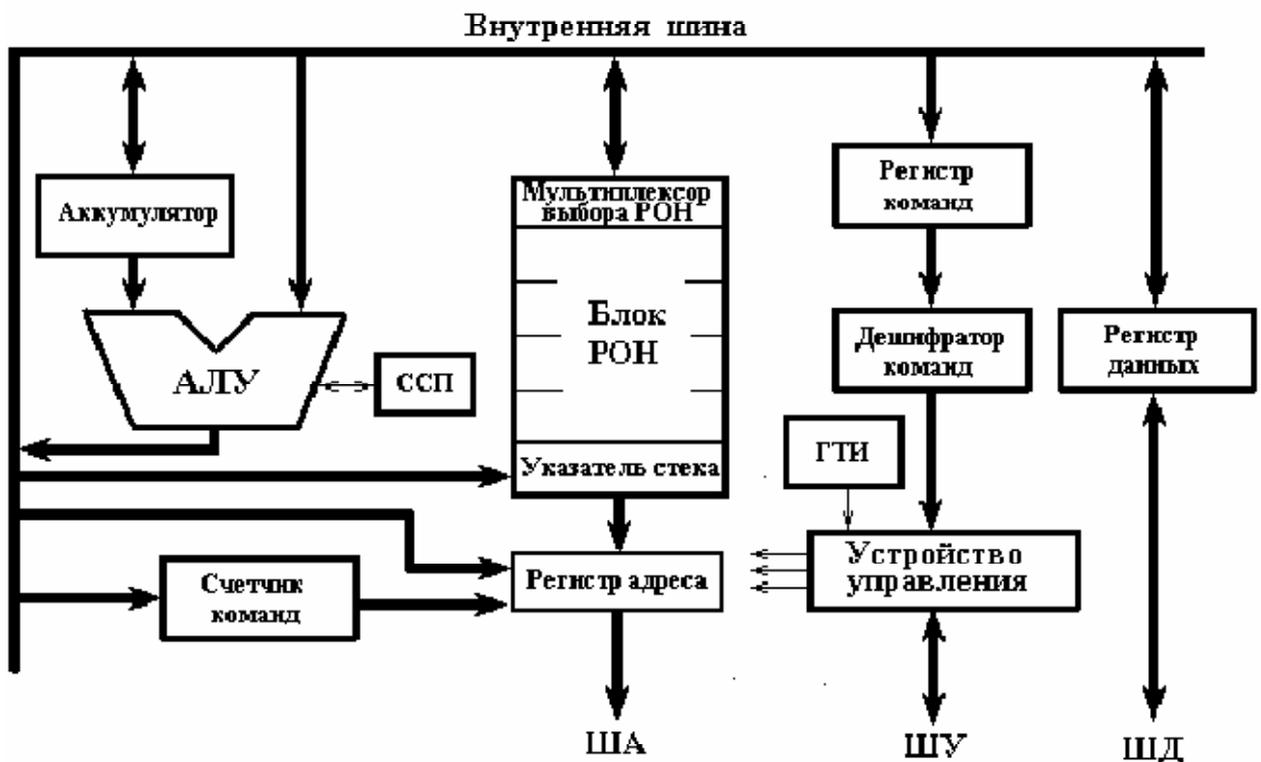


Рис 2 Упрощенная схема микропроцессорной системы

Регистр флагов

- При работе с операндами в АЛУ возникают различные состояния, связанные, например, с нулевым и отрицательным результатами, с переполнением разрядной сетки и т. п.
- Чтобы фиксировать эти события используется специальный регистр, в котором значения отдельных бит соответствуют различным событиям: состояние переноса, значение знака, нулевой результат и др. Такой регистр получил название регистра слова состояния процессора (ССП).

Биты событий называют флажками(флагами), а регистр иногда называют регистром флагов.

Флаги этого регистра используются для принятия решений, когда вводятся команды ветвлений

Аккумулятор

- При выполнении операции в АЛУ на его 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

Регистры общего назначения

- Остальные рабочие регистры используемые в операциях называются *регистрами общего назначения (РОН)*.
- Один из операндов часто хранится в одном из РОН, другой же – в аккумуляторе

Регистр команд и дешифратор команд

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

Регистр данных / адреса.

- Составляется из двух 8-разрядных регистров которые могут быть использованы вместе или раздельно. Обозначены Н и L соответственно старшему и младшему (HIGH и LOW) байтам.
- Регистры HL являются универсальными. Могут хранить данные, адрес и др.

Устройство управления

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)*.

Счетчик команд

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.

Эмулятор микропроцессора КР580

Изучение работы и программирование реального микропроцессора КР580 может быть выполнено с помощью программы эмулятора. Эмулятор имитирует работу микропроцессора и подключенных к нему внешних устройств: оперативной памяти, портов ввода / вывода и т.п.

Такая структура называется микропроцессорной системой.



Запуск эмулятора выполняется с помощью ярлыка на рабочем столе.

После запуска появляется окно программы (рис 1)

Содержимое главного окна программы:

1. Главное меню программы
2. Структурная схема микропроцессорной системы
3. Таблица содержимого ОЗУ микропроцессорной системы (МП-системы)
4. Внешние периферийные устройства, подключенные к портам МП-системы
5. Панель редактирования значения выбранной ячейки ОЗУ
6. Панель редактирования значения выбранного регистра общего назначения
7. Группа кнопок «Сброс» для обнуления всех ячеек ОЗУ и регистров
8. Панель системы команд микропроцессора КР580ВМ80А
9. Группа кнопок «Выполнение» для выполнения программы в сквозном, командном и тактовом режиме

Структурная схема микропроцессорной системы (МПС) расположена в центральной части главного окна программы, как показано на рисунке 1, п.2, и содержит следующие элементы:

- Регистр слова состояния микропроцессора (PSW) и его значение, представленное в двоичной системе счисления, а также, расшифровку этого значения, представленного в словесной форме;
- Буфер данных МПС и его значение, представленное в шестнадцатеричной системе счисления;
- Регистр-аккумулятор (А) и его значение, представленное в шестнадцатеричной системе счисления;
- Буферные регистры 1 и 2 и их значения, представленные в шестнадцатеричной системе счисления;
- Регистр признаков (флагов) и его значение, представленное в двоичной системе счисления, а также, индикаторы расшифровки флагов: Z, S, P, C, AC;
- Регистр команд и его значение, представленное в шестнадцатеричной системе счисления;
- Дешифратор команд, индицирующий мнемонику текущей выполняемой команды, закреплённой на регистре команд;

- Счётчики машинных микроциклов и микротактов, индицирующие свои текущие значения в десятичной системе счисления;
- Блок АЛУ;
- Блок десятичной коррекции значения регистра-аккумулятора;
- Блок синхронизации и управления;
- Буфер адреса и его значение, представленное в шестнадцатеричной системе счисления;
- Блок регистров общего назначения и их значения, представленные в шестнадцатеричной системе счисления. Регистры В, С, D, E, H, L;
- Блок регистров временного хранения и их значения, представленные в шестнадцатеричной системе счисления. Регистры W, Z;
- Схема инкремента/декремента, индицирующее своё соответствующее действие условными обозначениями «+1» и «-1» соответственно;
- Регистр-указатель стека и его значение, представленное в шестнадцатеричной системе счисления;
- Регистр-счётчик команд и его значение, представленное в шестнадцатеричной системе счисления;
- Контролер ввода/вывода;
- Индикаторы состояния и тактирования микропроцессора: F1, F2, SYNC, READY, WAIT, HOLD, HLDA, INT, INTE, DBIN, WR;
- Порты от 00h до 04h для монитора, дисковод, жёсткого диска, сетевого адаптера и принтера соответственно;
- Все элементы связаны между собой шинами: данных, адреса, управления, внутренней шиной данных и шиной внешних устройств (портов) в соответствии со структурной схемой.

Следующие элементы структурной схемы носят активный характер, позволяющий, при помощи щелчка мыши на их значениях, отобразить и редактировать последнее в панели редактирования значений регистров ([рисунок 1, п.6](#)):

- Аккумулятор;
- Регистры блока РОН: В, С, D, E, H, L;
- Регистры временного хранения: W, Z;
- Указатель стека;
- Счётчик команд.

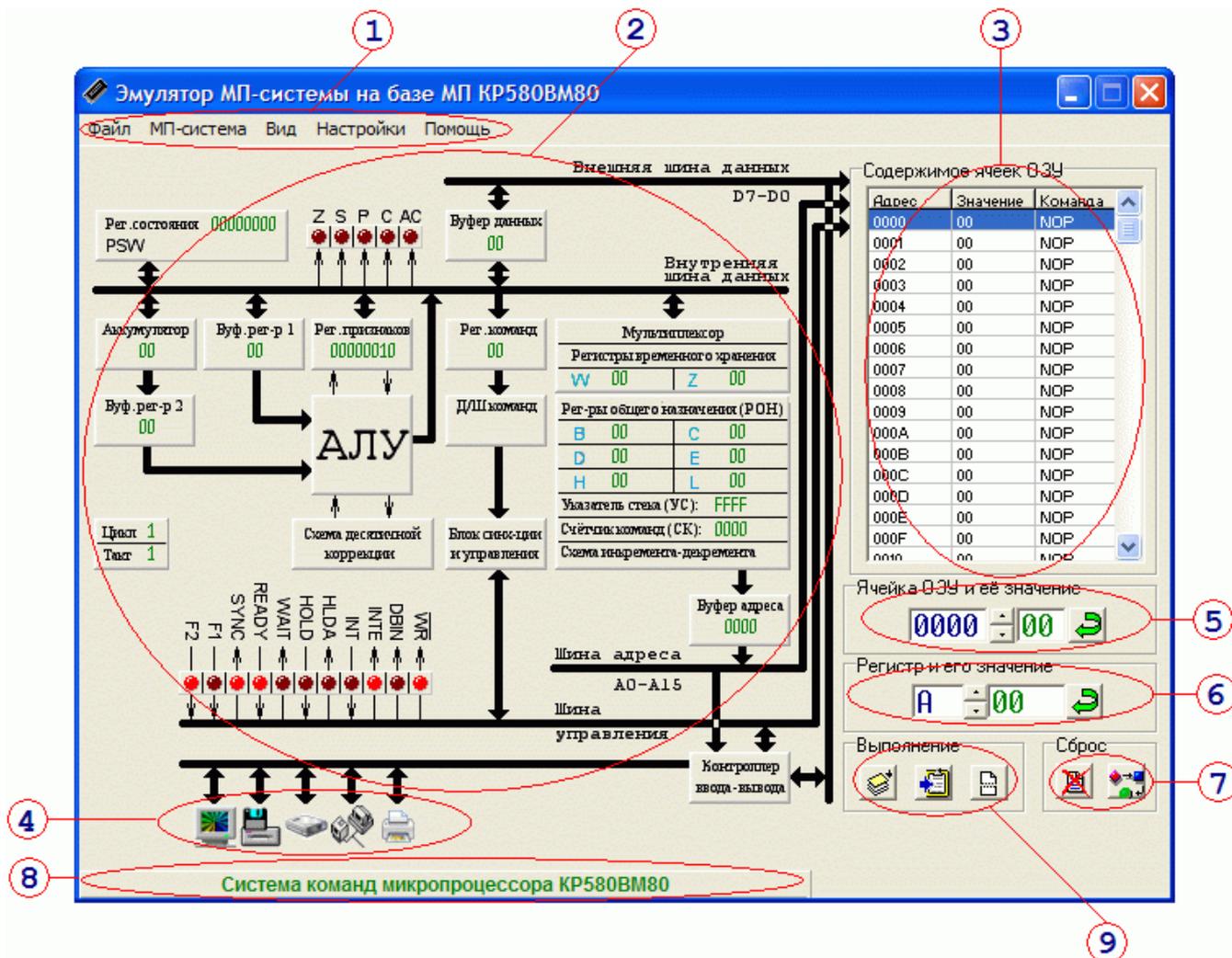


Рисунок 1 – Главное окно программы

Основные принципы работы с программой

Начало работы с программой в основном заключается в написании или загрузке программы на ассемблере в эмулятор. Для этого можно воспользоваться либо панелью системы команд программы, либо панелью редактирования значений ячеек ОЗУ эмулятора, либо загрузить образ ОЗУ с носителя. При необходимости, можно заполнить соответствующими значениями регистры общего назначения эмулятора.

После чего, для подробного изучения каждого такта конкретной команды, можно воспользоваться кнопкой тактированного выполнения команды ;

Для выполнения программы в командном режиме, (к примеру, для отладки программы) используется кнопка командного режима .

Соответственно, для выполнения программы целиком, следует пользоваться кнопкой программного режима .

Если выполнение программы завершается командой останова 76h HLT, устанавливается флаг останова микропроцессора HLDA и выдаётся соответствующее сообщение. Работу с эмулятором можно продолжить, сняв флаг при помощи пункта меню "МП-система→Снять флаг HLDA", либо воспользовавшись клавишей F12 на клавиатуре, либо произвести сброс РОН (см. ниже).

Для очистки (обнуления) РОН воспользуйтесь кнопкой "Сброс" , для сброса ОЗУ - , либо пунктом меню "Файл→Новый (очистить память и регистры)"

После написания программы на языке Ассемблера, её можно сохранить в виде образа ОЗУ и РОН эмулятора на какой-либо носитель, и при следующей надобности, также загрузить в эмулятор. Загружать и сохранять можно не только весь образ, но и часть ОЗУ эмулятора.

Для удобства работы с написанными программами в виде таблиц или текстовых документов, предусмотрены возможности экспорта части содержимого ОЗУ и/или РОН эмулятора в MS Excel, MS Word и текстовый файл.

Также предусмотрена возможность печати части содержимого ОЗУ и/или РОН эмулятора на принтере.

Задания

А) Загрузка чисел и команд в регистры и ячейки оперативной памяти

1. Выполните редактирование ячеек оперативной памяти. Введите в ячейку 0002 число 78_{16}

2. Выполните редактирование регистра общего пользования. Введите в регистр В число 26_{16}

3. Введите в аккумулятор А число 58 с помощью команды. Для этого введите в ячейку 0000

код команды 3E соответствующий мнемокоду команды MVI A. В следующую ячейку 0001

введите размещаемое число- 58.

4. Установите курсор на ячейку 0000 и нажмите кнопку выполнения команды в главном окне программы. Убедитесь, что нужное число установлено в регистр А

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Что такое эмулятор?
2. Что такое микропроцессорная система?
3. Что такое машинный код?
4. Характеризуйте аккумулятор микропроцессора
5. Что такое РОН?
6. Что такое регистр флагов?
7. Как изменить содержимое регистров процессора?
8. Как отредактировать содержимое ячейки памяти?

Лабораторная работа № 13

Тема: Программирование процессора КР 580. Выполнение арифметических операций

Цель: Исследование системы команд микропроцессора КР580.

Выработка навыков применения команд для процессора КР580 на языке ассемблер с применением эмулятора для выполнения арифметических операций.

Перечень необходимых средств обучения:

персональный компьютер, программа эмулятор КР580 , таблица команд микропроцессора.

Краткие теоретические сведения

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
2000	0011 1110
2001	1011 0100
2002	0010 1111
2003	0011 0010
2004	0000 0000
2005	0010 0001
2006	0111 0110
2007	

а)

Адрес, Н-код	Содержимое, Н-код
2000	3E
2001	B4
2002	2F
2003	32
2004	00
2005	21
2006	76
2007	

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 знаком 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 изучить систему команд ассемблера для КР580 и программу эмулятор работы микропроцессора.

Эмулятор имитирует работу микропроцессора с программой. Общий вид рабочего стола ПК с открытой программой эмулятора приведен на рис 3

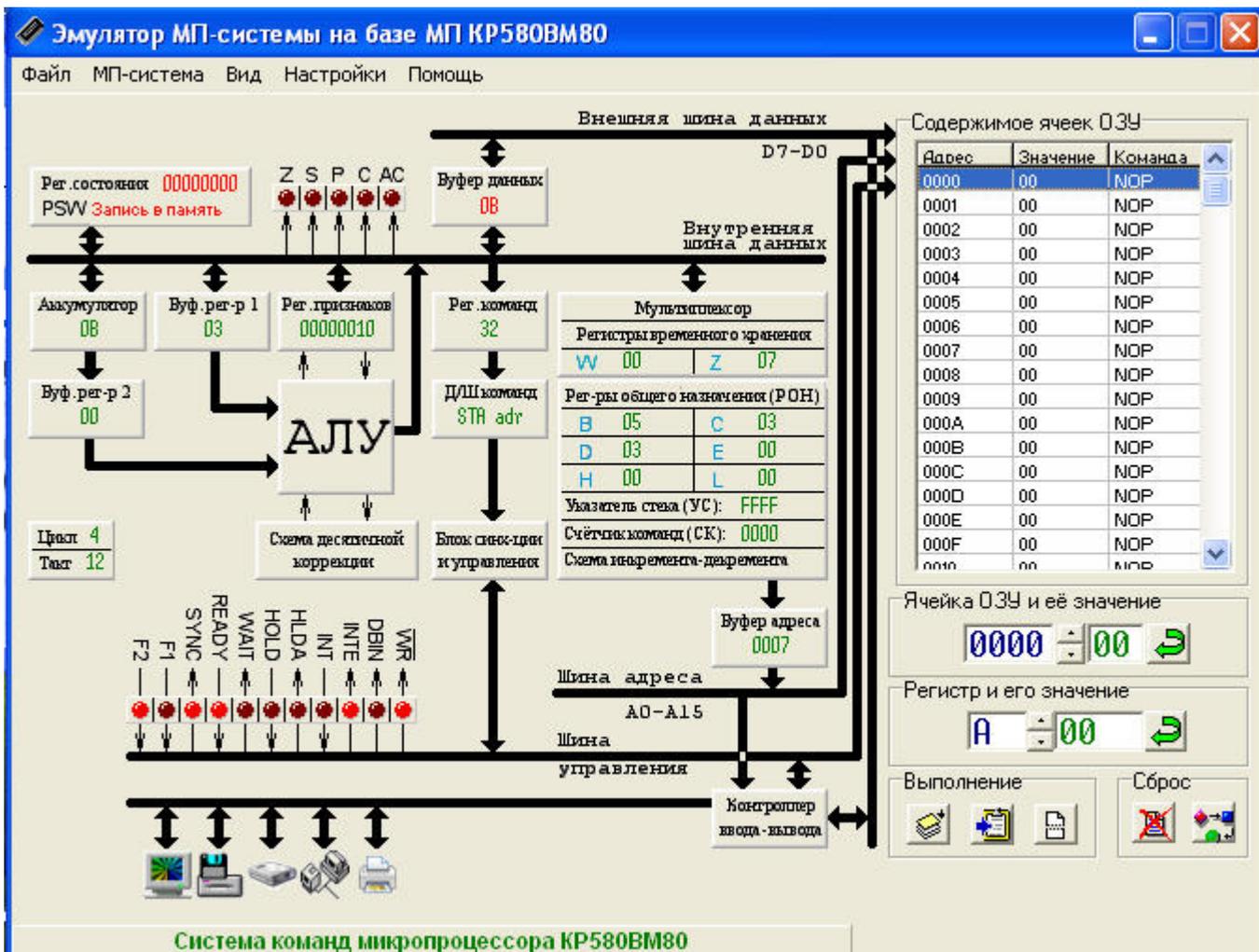


Рис 3 Общий вид рабочего стола эмулятора

Система команд микропроцессора (коды команд и мнемоника) открывается из строчки меню «Система команд микропроцессора КР580ВМ80 и приведена на рис 4.

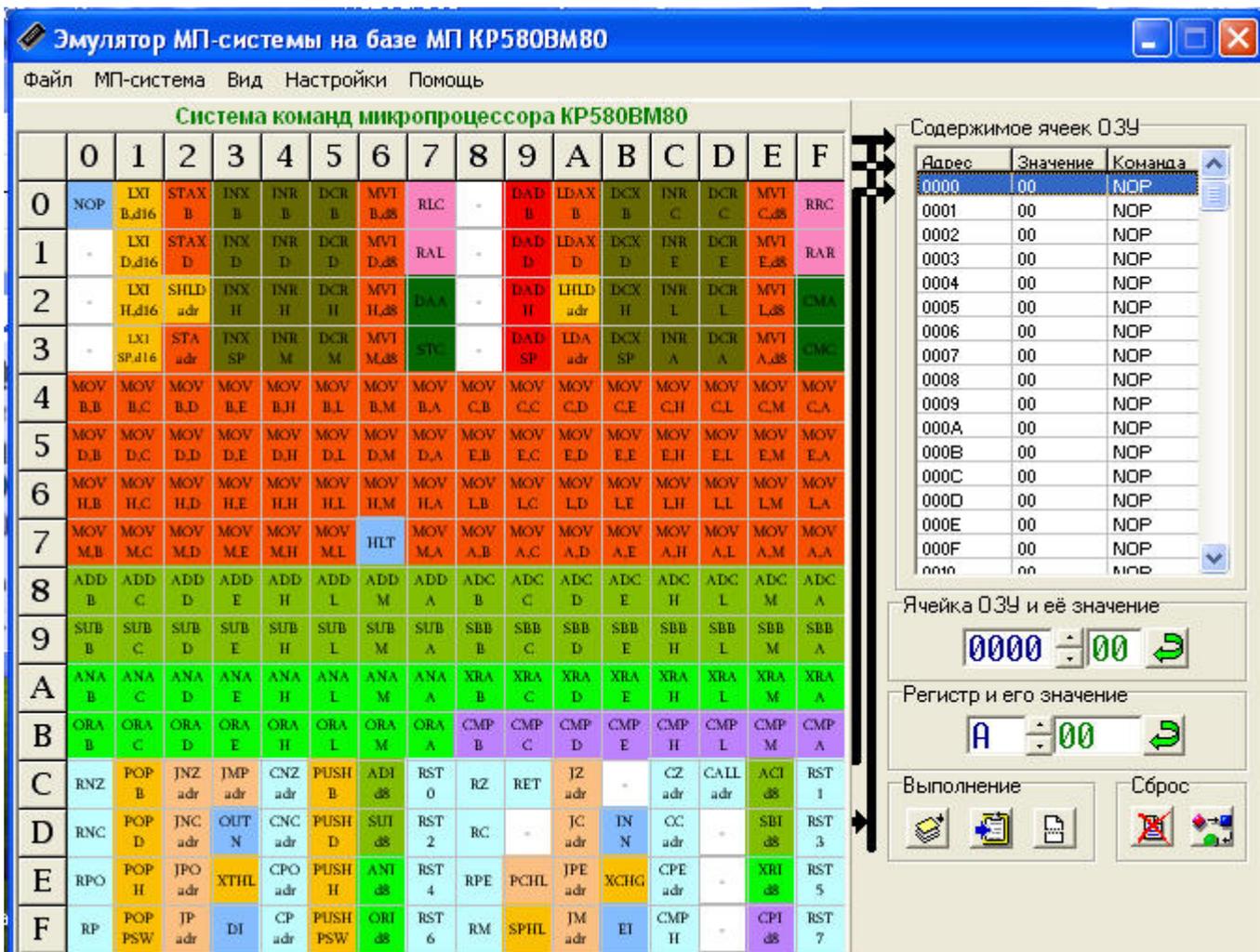


Рис 4 Система команд микропроцессора KP 580

Для того, чтобы поместить нужную команду в ячейку памяти необходимо ввести его вручную в ячейку или выполнить перетаскивание с помощью мыши. Проверка работы команды производится с помощью кнопки выполнения

Выполнение работы

1. Запустите программу «Эмулятор микропроцессора KP580» с помощью ярлыка на рабочем столе
2. Изучите правила работы с эмулятором (меню «Помощь»)
3. Изучите команды ассемблера для микропроцессора KP580 (файл на рабочем столе компьютера или распечатка).
4. Вызовите таблицу кодов команд (как показано на рис 1). Изучите команды загрузки операндов в регистры и ячейки памяти (группы команд с мнемокодами LDA, , SHLD, LHLD). Изучите команды перемещения операндов MVI ,STA, MOV Чтобы получить информацию выберите команду и нажмите на неё правой кнопкой мыши.
5. Разместите в ячейках оперативной памяти начиная с ячейки 0000 команды и проверьте их выполнение:
 - 5.1 Загрузить командой в аккумулятор (регистр A) число 12

- 5.2 Запишите вручную в ячейку памяти 0020 число 11. Переместите командой число из этой ячейки в аккумулятор.
- 5.3 Запишите вручную в ячейки памяти числа. В ячейку 0021 число 17 в ячейку 0022 число 18. С помощью команды переместите эти числа из ячеек памяти в регистры H и L
- 5.4 Переместите командой числа из регистров H и L в ячейки 0023 и 0024
- 5.5 Переместите командой число из регистра H в регистр A
- 5.6 Переместите число из аккумулятора в ячейку 0025
6. Вызовите таблицу кодов команд. Изучите команды арифметического сложения и вычитания (группы команд с мнемосодами ADD, SUB).
 Выполнить вычисление:

$$Y = X1 + X2 - X3$$
 Результат Y разместить в ячейку 0034
 Начальные условия: Числа $X1=7$, $X2=2$, $X3=3$ разместить соответственно в ячейки 0030 – 0032
7. Подготовьте отчет о проделанной работе. Приведите в отчете составленные программы

Лабораторная работа №14

Тема: Программирование процессора КР 580 Разветвляющиеся программы

Цель: Изучение принципов программирования микропроцессоров на языке ассемблер.

Изучение принципов создания разветвляющихся программ

Перечень необходимых средств обучения:

Персональный компьютер, программа Эмулятор микропроцессора КР580.

Краткие теоретические сведения

Для некоторых программ порядок выполнения зависит от полученного промежуточного результата. В этом случае необходимо выполнить ветвление программы. В языке Ассемблер предусмотрены команды для организации ветвления.

Стандартный процессор содержит регистр признаков (регистр флагов). Каждый бит этого регистра принимает значение «0» или «1» в зависимости от полученного результата (содержимого регистра аккумулятор). Таблица состояний регистра флагов приведена ниже.

D7	D6	D5	D4	D3	D2	D1	D0
----	----	----	----	----	----	----	----

S	Z	0	AC	0	P	1	CY (C)
---	---	---	----	---	---	---	-----------

Здесь:

S – флаг знака (принимает значение старшего разряда результата). $S = 1$ – если самый значащий бит результата равен единице, т. е. число отрицательное, иначе $S = 0$.

Z – флаг нуля. $Z = 1$, если результат операции нулевой, иначе $Z = 0$.

AC – флаг вспомогательного переноса (переноса между тетрадами байта). Если при выполнении операций происходит перенос из младшей тетрады в старшую, то $AC = 1$, иначе $AC = 0$. Для команд логического умножения признак вспомогательного переноса (AC) принимает значение 4-го разряда результата (аккумулятора): $(AC) \text{ f } A(3)$

P – флаг четности (или паритета). Если число единиц в байте результата четно, то $P = 1$, иначе $P = 0$. Не следует путать понятие паритета с понятием четности в общеупотребительном смысле (для чисел, представленных в десятичной системе счисления).

CY (C) – флаг переноса (или заема), $CY = 1$ – если операция привела к переносу из старшего разряда (или заему в старший разряд), иначе $CY = 0$.

Для выполнения условий ветвления часто используется группа команд сравнения **СМР**. Команды меняют флаг в зависимости от результата сравнения и создают условия для передачи управления в другую ячейку. Характеристика команды приведена ниже.

Команды сравнения:

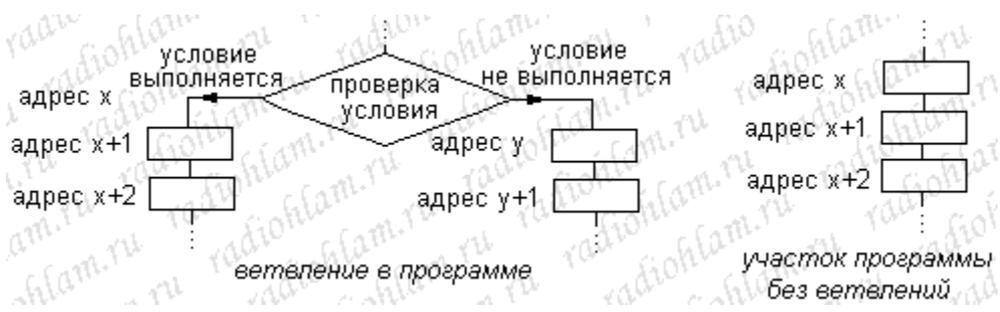
СМР R – операция $(A) - (R)$;

если $(A) = (R)$, то $Z = 1$;

если $(A) < (R)$, то $C = 1$;

Где **Z** и **C** флаги признаков в регистре

Ветвление - это такое место в программе, после которого в зависимости от какого либо условия может начать выполняться тот или иной код. То есть, счетчик команд в результате выполнения команды ветвления может быть установлен по двум различным адресам, в зависимости от исхода проверки какого-либо условия.



- Условный переход – переход к заданной ячейке памяти при выполнении какого либо условия ($Z=0$, $Z=1$, $S=1$ и т.д.)
- Безусловный переход – переход к выполнению подпрограммы в ходе выполнения программы.
- В микропроцессоре команды выполняются последовательно из ячейки в ячейку если нет команды перехода.
- В случае перехода к блоку программы А затем будет выполняться и блок Б. Необходимо предусмотреть переход от блока А к блоку С

Команды «Прыжка» Начинаются с «J». Пример характеристик одной из таких команд приведен ниже.

JMP_ V2_ V3 – записать информацию из второго и третьего байта команды в счетчик команд

Безусловный переход по адресу.

Управление передается команде, чей адрес указан во 2-ом и 3-ем байте текущей команды.

Выполнение работы:

1. Запустите программу Эмулятор КР580
2. Вызовите меню системы команд эмулятора. Ознакомьтесь с характеристиками группы команд CMP
3. Выполните операцию сравнения двух произвольных чисел для одной из команд. Проверьте результат.
4. Вызовите меню системы команд эмулятора. Ознакомьтесь с характеристиками группы команд Jxx (JMP, JNZ, JNC и т.д). Эти команды позволяют «прыгать» (Jump) и передавать управление в другую ячейку
5. Составить программу:

Составить программу:

Ввести числа X1, X2, X3 в ячейки 0030- 0032 соответственно

Если $X1 = X3$ то $Y = X1 + X3$

Иначе $Y = X1 + X2$

Результат У записать в ячейку 0035

Для проверки ввести числа 05,07, 05

Дополнит. проверка ввести 05, 07, 06

Проверьте результат запустив программу на эмуляторе.

6. Составьте программу в соответствии с заданием:

- В ячейках памяти 0030 – 0032 находятся 3 числа. Расположить эти числа в порядке возрастания в ячейках 0035- 0037.
- При проверке числовой материал получить у преподавателя

Контрольные вопросы

1. Что представляет собой регистр флагов?
2. Для чего применяется регистр флагов?
3. Что такое разветвляющиеся программы?

Лабораторная работа № 15

Тема: Исследование таймера

Цель: Изучить работу микросхемы таймера NE555

Перечень необходимых средств обучения:

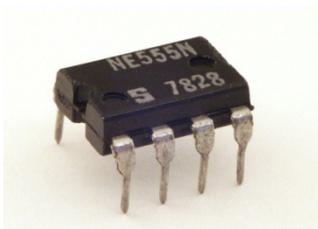
Плата для сборки без пайки, комплектующие электронные компоненты, источник питания

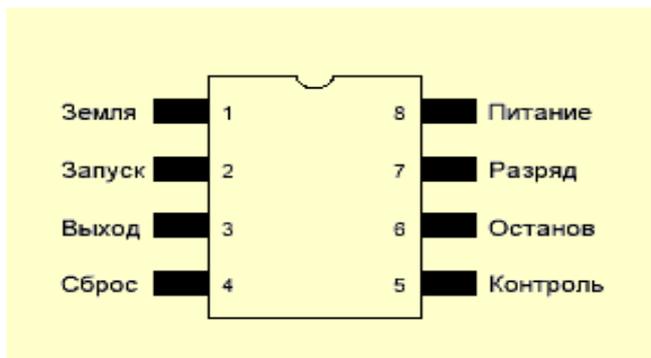
Краткие теоретические сведения

Микросхемы таймеров предназначены:

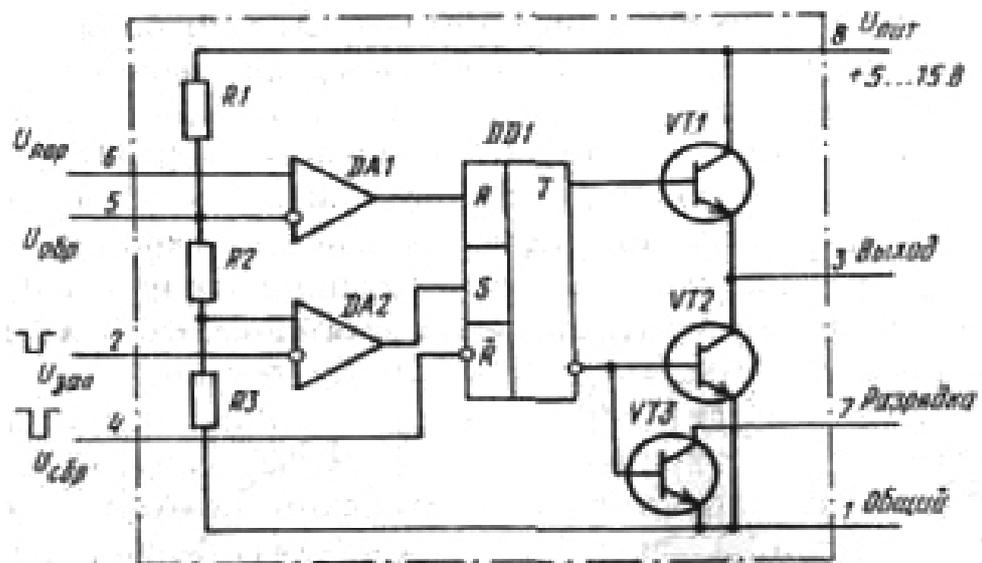
- ✓ Для включения и выключения устройств через определенные промежутки времени
- ✓ Для создания последовательности импульсов прямоугольной формы различной длительности и скважности. Период импульсов определяется параметрами подключаемых к микросхеме внешних компонентов: конденсатора и резистора

Первая микросхема - таймер была выпущена в 1971 году компанией Signetics Corporation под названием «Интегральный таймер NE555». Удачные технические решения позволили выпускать её практически без изменений 45 лет

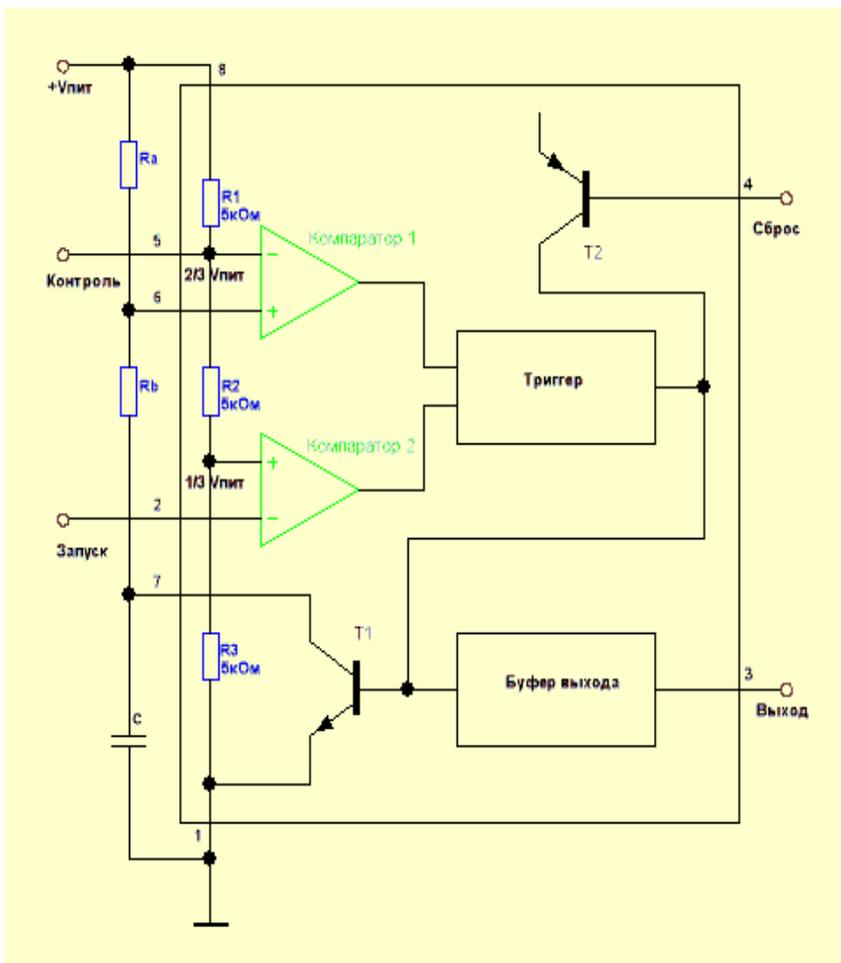




Цоколевка микросхемы



Электрическая схема NE555



Функциональная схема микросхемы таймера

Назначение выводов микросхемы

- 1. Земля.** Вывод, который подключается к минусу питания и к общему проводу схемы.
 - 2. Запуск.** Вход компаратора №2. При подаче на этот вход импульса низкого уровня (не более $1/3 V_{пит}$) таймер запускается и на выходе устанавливается напряжение высокого уровня на время, которое определяется внешним сопротивлением R ($R_a + R_b$) и конденсатором C - это так называемый режим моностабильного мультивибратора. Ток, потребляемый входом, не превышает 500 нА
 - 3. Выход.** Выходное напряжение меняется вместе с напряжением питания и равно $V_{пит} - 1,7 \text{ В}$ (высокий уровень на выходе). При низком уровне выходное напряжение равно примерно $0,25 \text{ В}$ (при напряжении питания $+5 \text{ В}$). Переключение между состояниями низкий - высокий уровень происходит приблизительно за 100 нс .
 - 4. Сброс.** При подаче на этот вывод напряжения низкого уровня (не более $0,7 \text{ В}$) происходит сброс выхода в состояние низкого уровня не зависимо от того, в каком режиме находится таймер на данный момент. Для предотвращения случайных сбросов этот вывод рекомендуется подключить к плюсу питания, пока в нем нет необходимости.
 - 5. Контроль.** Этот вывод позволяет получить доступ к опорному напряжению компаратора №1, которое равно $2/3 V_{пит}$. Обычно, этот вывод не используется и его рекомендуется подключить к общему проводу через конденсатор $0,01 \text{ мкФ}$ для уменьшения уровня помех.
- Подачей напряжения на этот вывод можно управлять длительностью выходных импульсов таймера и отказаться от RC времязадающей цепочки.
- 6. Останов.** Этот вывод используется для остановки таймера и приведения выхода в состояние низкого уровня.

При подаче импульса высокого уровня таймер останавливается, и выход сбрасывается в состояние низкого уровня

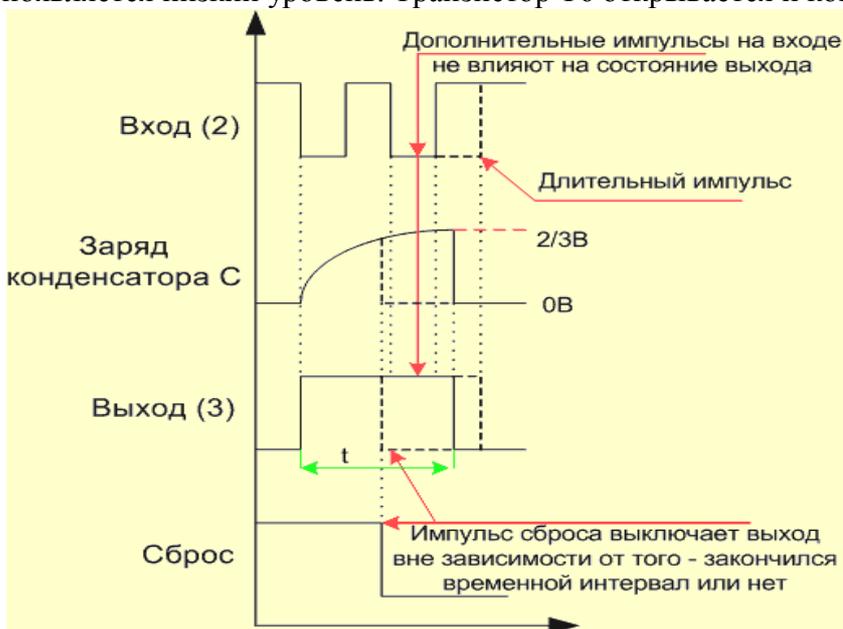
7. Разряд. Этот вывод может применяться как вспомогательный выход. Нагрузочная способность его примерно такая же, как и у обычного выхода таймера.

8. Плюс питания. Напряжение питания таймера может находиться в пределах 4,5-16 вольт

Принцип работы таймера

На микросхему подано питание. Вход находится в состоянии высокого уровня, на выходе - низкий уровень, конденсатор С разряжен.

Подаем импульс низкого уровня. Выход таймера переключается в состояние высокого уровня. Транзистор Т6 закрывается и конденсатор начинает заряжаться через резистор R. Все то время пока конденсатор заряжается, выход таймера остается во включенном состоянии - на нем сохраняется высокий уровень напряжения. Как только конденсатор зарядится до $2/3$ напряжения питания, выход микросхемы выключается и на нем появляется низкий уровень. Транзистор Т6 открывается и конденсатор С разряжается.



Задания

1. Выполните сборку стенда в соответствии с электрической схемой
2. Подключите стенд к источнику питания
3. Наблюдайте переключение таймера. Оцените длительность импульсов.
4. Выключите питание стенда и замените конденсатор на конденсатор большей емкости
5. Включите питание. Оцените длительность импульсов.
6. Сделайте выводы