

# ЛАБОРАТОРНАЯ РАБОТА N 1

## ТЕМА РАБОТЫ: Обучение работе на УМК

**Цель работы:** Изучение микропроцессорного комплекта КР580ВМ80А. Основные команды. Правила ввода данных. Управление вводом/выводом. Индикация работы УМК.

### Теоретическое обоснование

Учебный микропроцессорный комплект (УМК) создан на базе микропроцессора КР580ВМ80А и предназначен для подготовки специалистов в области микропроцессорной техники и обучению основам программирования.

Микропроцессор КР580ВМ80А предназначен для выполнения определенного набора команд (представленного в приложении 1) и реализован на одной БИС.

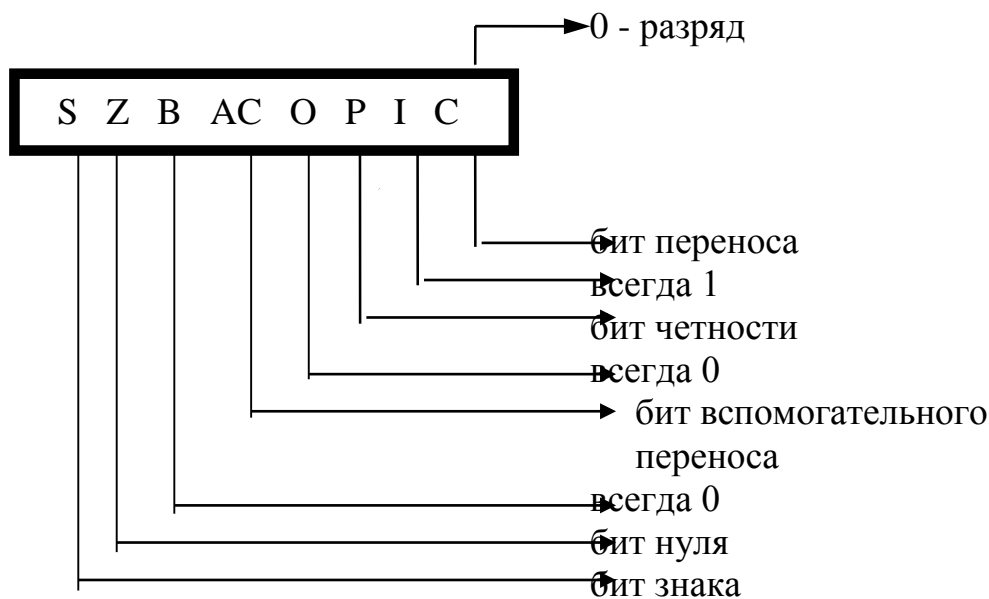
Все команды микропроцессора можно разделить на 8 основных групп:

- команды пересылки данных и загрузки регистров;
- команды работы с памятью;
- арифметические, логические команды и команды сравнения и сдвига;
- команды передачи управления;
- команды манипуляции стеком;
- команды вызова подпрограмм и возврата из них;
- команды ввода/вывода;
- команды обработки прерываний.

Для оперативного хранения данных в микропроцессоре имеется семь 8-разрядных регистров общего назначения:

- А - регистр-накопитель, аккумулятор;
- В - регистр общего назначения;
- С - регистр общего назначения;
- Д - регистр общего назначения;
- Е - регистр общего назначения;
- Н - регистр косвенного адреса;
- L - регистр косвенного адреса.

Для хранения битов условий имеется 8-разрядный регистр признаков PSW. Распределение и назначение битов условий регистров PSW следующее:



Бит переноса устанавливается в 1, если возникает переполнение старшего разряда в байте, в противном случае устанавливается в 0.

Бит четности устанавливается в 1, если количество единиц в байте четное, в противном случае устанавливается в 0.

Бит вспомогательного переноса устанавливается в 1, если возникает перенос из младшей тетрады байта в старшую, в противном случае - в 0.

Бит нуля устанавливается в 1, если результат выполнения операции равен 0, в противном случае в 0.

Бит знака устанавливается в 1, если результат операции отрицательный (ст. разряд байта равен 1), в противном случае, устанавливается в 0.

16-разрядный регистр-указатель стека SP служит для определения нижней границы области стека.

16-разрядный регистр-счетчик команд PC служит для хранения адреса выполненной команды.

### **Двоичная и шестнадцатиричная система счисления.**

В цифровой технике для представления данных используют двоичную систему счисления, т.к. для этих систем используемый цифровой сигнал может иметь только значения - 0 и 1. Т.о., одним разрядом двоичного числа можно представить два числа. Используя два разряда можно представить 4 числа ит. д... В микропроцессоре используются 8-разрядные двоичные слова, называемые байтами. Каждый байт может представить число от 0 до 255, в зависимости от комбинации в нем 0 и 1.

**Например.**

00000000 - 0  
00000001 - 1  
00000010 - 2  
00000011 - 3  
.....  
00001111 - 15  
.....  
11111111 - 255

Для удобства оперирования двоичными данными используют их шестнадцатиричное представление. Для этого байт делят пополам (на тетрады) и каждую тетраду кодируют следующим образом:

0000 - 0	1000 - 8
0001 - 1	1001 - 9
0010 - 2	1010 - A
0011 - 3	1011 - B
0100 - 4	1100 - C
0101 - 5	1101 - D
0110 - 6	1110 - E
0111 - 7	1111 - F

Каждая тетрада представляет 1 из 16 чисел. При этом вес старшей тетрады в байте равен 16. Например, число 10 в шестнадцатиричном коде представляет число 16.

Например:

двоичное	шестнадцатиричное	десятичное
00010000	10	16
00100100	24	36
11111111	FF	255

В дальнейшем используется шестнадцатиричная система счисления для представления данных.

### **Содержание работы:**

- 1.1. Ознакомление с органами управления, ввода, индикации УМК.
- 1.2. Включение и запуск УМК.
- 1.3. Команды системного монитора УМК.
  - 1.3.1. Просмотр и модификация содержимого ячейки памяти.
  - 1.3.2. Просмотр и модификация содержимого регистра микропроцессора (МП).

- 1.3.3. Старт программы.
- 1.3.4. Заполнение массива памяти константой.
- 1.3.5. Подсчет контрольной суммы.
- 1.3.6. Перемещение массива памяти.

## **ВЫПОЛНЕНИЕ РАБОТЫ**

1.1. Ознакомление с органами управления, ввода, индикации.

Откройте крышку дипломата УМК. На лицевой панели УМК размещены кнопки включения/выключения, сброса и прерывания, кнопки управления пошаговым режимом работы УМК, функциональная клавиатура, клавиатура ввода данных, шести сегментный дисплей, светодиоды индикации состояния шины данных, шины адреса, управляющие сигналы МП.

Кнопка включения/выключения УМК СЕТЬ расположена в левой нижней части лицевой панели. На самой кнопке выгравирован символ " ~ ". Эта кнопка служит для включения (нажатое состояние кнопки) и выключения (отжатое состояние кнопки) УМК. Над кнопкой размещены три светодиода, над которыми выгравировано: +5В,-5В,+12В. При перегрузках срабатывает защита блока питания и загорается соответствующий светодиод. В случае необходимо выключить УМК и вызвать специалиста по обслуживанию. Кнопка СБРОС расположена в правой средней части лицевой панели УМК. На кнопке выгравировано СБ. Эта кнопка служит для инициализации системной программы МОНИТОР, и лицевой позиции шестисегментного дисплея появляется символ " - ". УМК готов к приему команд.

Кнопка ПЕРЫВАНИЕ располагается под кнопкой СБРОС. На ней выгравировано ПР. При нажатии на эту кнопку вырабатывается сигнал ЗАПРОС НА ПЕРЫВАНИЕ 7-ого уровня и, если прерывания разрешены (выполнена команда EI), будет закончено выполнение текущей команды, и управление будет передано на адрес 38H (что соответствует седьмому уровню прерываний). Начиная с этого адреса, располагается программа обработки прерывания. Если кнопка ПР будет нажата во время работы программы МОНИТОР (на любой стадии), на дисплей будет выведен символ " ? ". В противном случае на дисплей будет выведен адрес точки прерывания и управление будет передано программе МОНИТОР.

Управление пошаговым режимом работы УМК производится с помощью кнопок, на которых выгравированы надписи: РБ/ЩГ, КМ/ЦК и ШГ. С помощью этих кнопок может быть установлен один из двух режимов работы УМК по шагам. Первый режим -

покомандный. Для установки этого режима необходимо нажать кнопку РБ/ШГ (она останется в нажатом состоянии).

Для выполнения команд необходимо нажать кнопку ШГ. Каждое нажатие кнопки ШГ вызовет выполнение текущей команды. При этом на светодиодах индикации состояние шин данных, адреса и управляющих сигналов, расположенных в центре панели УМК, будут высвечиваться в двоичном коде (каждый светодиод отражает соответствующий разряд), соответственно, адрес и код выполняемой команды, а также управляющие сигналы МП.

Второй режим работы по шагам - работа по командным циклам. Для установки этого режима необходимо нажать кнопки: РБ/ШГ и КМ/ЦК. В этом случае можно проследить ход выполнения команды. При каждом нажатии на кнопку ШГ будет выполнен следующий машинный цикл. При этом на светодиодах индикации будет отражаться информация, соответствующая каждому машинному циклу.

Для работы в автоматическом режиме обе клавиши РБ/ШГ и КМ/ЦК должны быть в отжатом состоянии. Клавиатура УМК располагается в правой нижней части лицевой панели и разделена на две части. В левой части расположены функциональные клавиши. За каждой закреплена определенная функция системной программы МОНИТОР. На них выгравированы соответствующие идентификаторы функции:

П - просмотр и модификация содержимого ячейки памяти;

РГ - просмотр и модификация содержимого регистров МП;

СТ - старт;

КС - подсчет контрольной суммы;

ЗК - заполнение массива памяти константой;

ПМ - перемещение массива памяти;

└┘ - разделитель;

ВП - выполнить.

Первая часть клавиш предназначена для ввода параметров в шестнадцатиричной форме. В дальнейшем подразумевается, что все вводимые и выводимые данные имеют шестнадцатиричный формат. На них выгравированы символы: 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F. Кроме этого на клавишах 4, 5, 6, 7, 8, 9 под цифрами выгравированы идентификаторы регистров микропроцессора.

PH - ст. байт счетчика команд;

PL-мл. байт счетчика команд;

SH-ст. байт указателя стека;

SL-мл. байт указателя стека;

H-регистр H;

L-регистр L.

Для идентификации остальных регистров МП используются клавиши:

- А-регистр А;
- В-регистр В;
- С-регистр С;
- D-регистр D;
- Е-регистр Е;
- F-регистр признаков.

Шестисегментный дисплей расположен в центре лицевой панели УМК над клавиатурой и предназначен для отображения данных в шестнадцатеричной форме (как вводимых, так и результата). При этом левые 4 сегмента используются для отображения данных типа байт. Светодиоды индикации состояния шин адреса, данных и управляющих сигналов МП расположены в центре лицевой панели УМК. Шина адреса состоит из 16 светодиодов, а шина данных - из 8 светодиодов (для отображения в двоичном коде). Над ними выгравированы надписи, соответственно, АДРЕС и ДАННЫЕ. Под каждым светодиодом этих шин выгравированы номера разрядов, которым они соответствуют. Под ними размещены светодиоды управляющих сигналов МП с соответствующими надписями - СОСТОЯНИЕ и названия сигналов.

## **1.2. Включение и запуск УМК.**

Для включения УМК необходимо:

- а) отжать (выключить) кнопку СЕТЬ, если она была включена;
- б) подключить сетевой шнур к сети переменного тока напряжением 220 В и частотой 50 Гц;
- в) нажать кнопку СЕТЬ, она должна остаться в нажатом состоянии.

При этом не должны загораться светодиоды защиты +5В, -5В, +12В

Повторное включение УМК должно производиться не менее, чем через 20 сек после выключения. В противном случае сработает защита блока питания и загорятся соответствующие светодиоды защиты. В этом случае необходимо выключить УМК и дождаться, когда светодиоды защиты погаснут, и только после этого можно повторно включать УМК.

Для запуска УМК необходимо:

- отжать кнопки РБ/ШГ и КМ/ЦК, тем самым перевести УМК в автоматический режим работы;

- нажать на кнопку СБ. При этом в левой части дисплея появится символ " - ", что означает, что УМК готов к работе и находится в режиме приёма команд.

### 1.3. Команды системного монитора УМК.

Для ввода команд в УМК необходимо:

а) на функциональной клавиатуре нажать клавишу, соответствующую выбранной команде. При этом экран дисплея погаснет;

б) ввести параметры команд (если их несколько, то между ними необходимо нажать клавишу РАЗДЕЛИТЕЛЬ). По мере ввода данных, они будут отображаться в соответствующей части дисплея. При вводе параметров необязательно вводить лидирующие нули, вместо 01 можно ввести 1 или вместо 0023 - 23. Длина параметра не ограничена, однако при вводе параметров, отображающих адреса существенными являются только 4 правые позиции, при вводе байтовых значений - только 2;

в) нажать клавишу ВП. Результат выполнения команды появится на дисплее. Если при вводе команды будет допущена ошибка, на экран дисплея будет выведен символ "?", и команда будет снята. Оператор должен повторить ввод.

#### 1.3.1. Команда ПРОСМОТР И МОДИФИКАЦИЯ СОДЕРЖИМОГО ЯЧЕЙКИ ПАМЯТИ.

Эта команда используется для отображения или записи в память данных. Для выполнения этой команды необходимо:

а) нажать клавишу П;

б) ввести адрес ячейки памяти, например, 800H;

в) нажать клавишу ВП. На дисплее в правой части появится содержимое заданной ячейки

800 XX

г) введите новое значение - 0;

800 0

д) нажмите клавишу РАЗДЕЛИТЕЛЬ, осуществить переход к

следующей ячейке памяти

801 XX

е) введите новое значение - 1

800 1

ж) действуя аналогично пунктам д), е) введите далее следующие данные: 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F;

з) нажмите клавишу ВП.

На этом выполнение данной команды ЗАПИСЬ В ПАМЯТЬ будет закончено. УМК перейдёт в режим ввода следующей команды.

Для проверки правильности выполнения этой команды необходимо выполнить эту же команду без модификации содержимого памяти, т.е. в режиме чтения:

а) нажать клавишу П;

б) ввести адрес ячейки памяти - 800Н;

в) нажать клавишу ВП. На дисплее должно появиться:

800 00;

г) нажмите клавишу РАЗДЕЛИТЕЛЬ

801 01;

д) и т.д. до появления на дисплее

80F 0F;

Таким образом, вы можете просмотреть содержимое ячеек 800Н - 80FH. Оно должно совпадать с тем, что вы ввели раньше;

е) нажмите клавишу ВП.

### **Задание:**

1. Начиная с адреса 810Н введите последовательно в ячейки памяти и проверьте данные:

0FH, 0EH, 0DH, 0CH, 0BH, 0AH, 9, 8, 7, 6, 5, 4, 3, 2, 1.

Примечание: здесь и далее символ Н при обозначении данных означает шестнадцатиричный формат. При вводе данных символ Н вводить не надо.

2. По адресу 820Н введите ААН

“ 821Н “ 55

“ 822Н “ АА

“ 823Н “ 55

“ 824Н “ АА

“ 825Н “ 55

3. Заполните массив памяти 900Н - 90FH значением 0.

4. Запишите в ячейки:

920Н - 80Н      92АН - 80Н      940Н - 02Н

923Н - 40Н      92FH - 08Н      945Н - 01Н

927Н - 20Н      932Н - 04Н      94ЕН - 00Н

### **1.3.2. Команда ПРОСМОТР И МОДИФИКАЦИЯ СОДЕРЖИМОГО РЕГИСТРОВ.**



Эта команда используется как для просмотра, так и модификации регистров микропроцессора.

Изменим содержимое регистров микропроцессора А и В:

а) нажмите клавишу РГ;

б) введите идентификатор регистра А - А (на клавиатуре ввода данных). На дисплее появится содержимое регистра А;

А - ХХ;

введите новое значение:

А - 0А;

г) нажмите клавишу РАЗДЕЛИТЕЛЬ;

д) введите идентификатор регистра В - В:

В - ХХ;

е) введите новое значение - 0В:

В - 0В;

ж) нажмите клавишу ВП.

Для проверки правильности выполнения команды выполните следующие действия:

а) нажмите клавишу РГ;

б) введите идентификатор регистра А - А, должно появиться на дисплее:

А - 0А;

в) нажмите клавишу РАЗДЕЛИТЕЛЬ;

г) введите идентификатор регистра В - В, должно появиться на дисплее:

В - 0В;

д) нажмите клавишу ВП.

Если содержимое регистров А и В будет отлично от 0АН и 0ВН, значит при выполнении команды модификации регистров вы неправильно ввели новые значения. Выполните команду повторно.

### **Задание:**

1. Установите значение регистров:

А - 00Н      РН - 07Н

В - 01Н      РL - 08Н

С - 02Н      SH - 0ВН

D - 03Н      SL - 0АН

Е - 04Н      F - FFН    Н - 05Н    L - 06Н

2. Проверить правильность выполнения команды.

### **1.3.3. Команда СТАРТ ПРОГРАММЫ.**

Эта команда используется для запуска и отладки программ пользователя. Для выполнения этой команды необходимо

предварительно записать в память машинные коды программы, например, коды последовательности пустых команд NOP - 00H.

а) используя команду МОНИТОРА ПРОСМОТР И МОДИФИКАЦИЯ СОДЕРЖИМОГО ЯЧЕЙКИ ПАМЯТИ, запишите в ячейки памяти с адресами 800H - 80EH коды команды NOP - 00H;

б) в ячейку с адресом 80FH запишите FFH, код команды PST 7, выполняющей функцию программного прерывания для прекращения выполнения программы. Теперь выполним команду СТАРТ ПРОГРАММЫ. Для этого необходимо:

- нажать клавишу СТ;
- ввести стартовый адрес программы - 800H;
- установить альтернативные точки останова;
- нажать клавишу РАЗДЕЛИТЕЛЬ, на дисплее появятся символ “\_”;
- ввести адрес первой точки останова - 803H;
- нажать клавишу РАЗДЕЛИТЕЛЬ, на дисплее появится символ “\_”;
- ввести адрес альтернативной точки останова - 805H;
- нажать клавишу ВП, осуществится передача управления на адрес 800H. На дисплее появится адрес первой точки останова 803.

Продолжим выполнение программы, начиная с адреса первой точки останова, адреса 803H, до адреса 805H. Для этого:

- нажмите клавишу СТ;
- нажмите клавишу РАЗДЕЛИТЕЛЬ, это означает, что выполнение программы необходимо продолжить с текущего адреса;
- введите адрес точки останова 805H;
- нажмите клавишу ВП. Выполнение программы будет продолжено.

На дисплее появится 805 - адрес точки останова. Теперь продолжим выполнение программы без указания точки останова. Для запуска программы с начала, адреса 800H, без установки точек останова :

- нажмите клавишу СТ;
- введите стартовый адрес - 800H;
- нажмите клавишу ВП.

На дисплее появится адрес 810. Программа выполнена.

### **Задание:**

1. Запишите в ячейки памяти (830H - 83EH) - 00H.
2. Запишите в ячейки памяти 83FH - FFH.
3. Выполните команду СТАРТ ПРОГРАММЫ с установкой двух точек останова 835H и 83AH.

4. Продолжите выполнение программы до адреса 83СН.
5. Продолжите выполнение программы без указания точек останова.
6. Запустите программу с адреса 830программу с адреса 830Н без указания точек останова.

#### **1.3.4. Команда ЗАПОЛНЕНИЕ МАССИВА ПАМЯТИ КОНСТАНТОЙ.**

Эта команда используется для записи в массив памяти константы. Для выполнения этой команды:

- а) нажмите клавишу ЗК;
- б) введите начальный адрес массива - 800Н;
- в) нажмите клавишу РАЗДЕЛИТЕЛЬ, экран дисплея погаснет;
- г) введите конечный адрес массива 80FH;
- д) нажмите клавишу РАЗДЕЛИТЕЛЬ, последний введенный адрес останется на дисплее;
- е) введите константу ААН;
- ж) нажмите клавишу ВП.

Для проверки правильности выполнения команды заполнения содержимого ячейки памяти (см. п. 1.3.1).

#### **Задание:**

1. Заполните (и проверьте) массив памяти с адресом 840Н - 84FH данными - FFH.
2. Заполните (и проверьте) массив памяти с адресом 93АН - 952Н данными 05H.
3. Заполните (и проверьте) массив памяти с адресом 960Н - 97FH данными С7H.

#### **1.3.5. Команда ПОДСЧЁТ КОНТРОЛЬНОЙ СУММЫ.**

Перед выполнением этой команды заполните массив памяти 840Н-84FH данными 0АН (контрольная сумма будет равна 96Н).

Для выполнения команды ПОДСЧЕТ КОНТРОЛЬНОЙ СУММЫ:

- а) нажмите клавишу КС;
- б) введите начальный адрес массива 840Н;
- в) нажмите клавишу РАЗДЕЛИТЕЛЬ;
- г) введите конечный адрес массива 84FH;
- д) нажмите клавишу ВП. На дисплее появится значение контрольной суммы массива - 96.

**Задание:**

1. Заполните массив памяти (800H - 845H) данными 01H и подсчитайте контрольную сумму.
2. Заполните массив памяти (852H - 8FFH) данными C7H и подсчитайте контрольную сумму.
3. Заполните массив памяти (800H - 9FFH) данными FFH и подсчитайте контрольную сумму.

**1.3.6. Команда ПЕРЕМЕЩЕНИЕ МАССИВА ПАМЯТИ.**

Эта команда используется для пересылки данных из одной области памяти в другую. Например, для перемещения кодов программы.

Перед выполнением этой команды предварительно заполните массив памяти 800H - 83FH данными 55H и подсчитайте контрольную сумму этого массива. Должно получиться 40H.

Для выполнения команды ПЕРЕМЕЩЕНИЯ МАССИВА ПАМЯТИ:

- а) нажмите клавишу ПМ;
- б) введите начальный адрес перемещаемого массива - 800H;
- в) нажмите клавишу РАЗДЕЛИТЕЛЬ;
- г) введите конечный адрес перемещаемого массива - 83FH;
- д) нажмите клавишу РАЗДЕЛИТЕЛЬ;
- е) введите начальный адрес массива, куда осуществляется перемещение - 900H;
- ж) нажмите клавишу ВП.

Для проверки правильности выполнения команды перемещения подсчитайте контрольную сумму нового массива (900H - 93FH). Она должна совпадать с контрольной суммой перемещаемого массива (800H - 83FH) = 40H.

**Задание:**

1. Заполните массив памяти (850H - 8FFH) данными 03H. Подсчитайте контрольную сумму. Переместите этот массив в область с начальным адресом 950H. Проверьте правильность выполнения команды перемещения.
2. Переместите массив памяти (0 - 0FFH) в область с начальным адресом 800H. Проверьте правильность выполнения команды.
3. Переместите массив памяти (800H - 8FFH) в область с начальным адресом 880H. Проверьте правильность выполнения команды перемещения.

**Примечание.** Так как в задании 3 области памяти перекрываются, для сохранения данных используйте промежуточный буфер.

**Контрольные вопросы:**

1. Как осуществляется включение и запуск УМК?
2. Какие команды МП вы знаете?
3. Как осуществляется просмотр и модификация содержимого ячеек памяти?
4. Как осуществляется просмотр и модификация содержимого регистров МП?
5. Как осуществляется старт программы?
6. Как заполняется массив памяти константой?
7. Как производится подсчет контрольной суммы?
8. Как производится перемещение массива памяти?

## Лабораторная работа №2

### Изучение работы УМК в пошаговом режиме

**Цель работы:** Освоение режимов работы УМК по машинным циклам и по командам (установка режимов и применение при выполнении конкретных последовательностей команд).

#### Теоретический обзор.

Существуют два режима работы УМК по шагам:

- режим работы по машинным циклам;
- режим работы по командам.

**Режим работы УМК по машинным циклам** используется для изучения процесса выполнения команд в микропроцессоре.

**Режим работы УМК по командам** используется для изучения команд и отладки программ.

#### Содержание работы.

2.1. Режим работы УМК по машинным циклам.

2.1.1. Установка режима.

2.1.2. Выполнение последовательности команд.

2.2. Режим работы УМК по командам.

2.2.1. Установка режима.

2.2.2. Выполнение программы.

#### Выполнение работы.

##### 2.1. Режим работы УМК по машинным циклам.

2.1.1. Установка режима работы УМК по машинным циклам.

Для этого выполните следующие действия:

- а) включите и приведите в рабочее состояние УМК (см. Лаб. Раб. №1);
- б) нажмите клавишу РБ/ШГ (нажатое состояние);
- с) нажмите клавишу КМ/ЦК (нажатое состояние).

2.1.2. Выполнение последовательности команд.

Рассмотрим пример выполнения следующих команд:

Адрес	Команда	Машинный код	Комментарий
800	NOP	00	; пустая команда
801	MOV A, B	78	; пересылка $B \leftarrow A$
802	ADD C	81	; $A = A + C$
803	RRC	0F	; циклический сдвиг на 1 вправо
804	ANI 0FH	E6 0F	; $A = \text{AND } 0FH$
806	PUSH H	E5	; запись в стек HL
807	LXI H, 900H	21 00 09	; загрузка HL=900H (адрес M)
80A	MOV M, A	77	; запись $M=A$ по адресу HL = 900H
80B	MVI A, 0FFH	3E FF	; загрузка $A = 0FFH$
80D	SUB M	96	; $A=A-M$ по адресу HL
80E	POP H	E1	; $HL \leftarrow$ стек
80F	JMP 800H	C3 00 08	; переход на адрес 800H

2.1.2.1. Запишите в память, начиная с адреса 800H, коды указанной в п. 2.1.2. последовательности команд (используйте команду монитора ПРОСМОТР И МОДИФИКАЦИЯ СОДЕРЖИМОГО ЯЧЕЙКИ ПАМЯТИ).

2.1.2.2. Запустите программу с адреса 800H (используйте команду СТАРТ ПРОГРАММЫ).

#### СТ800ВП

На шине адреса появится адрес первой выполняемой команды 800, на шине данных - код выполняемой команды NOP - 00 ( все светодиоды шины данных погашены). После нажатия клавиши ШГ микропроцессор перейдет к выполнению следующей команды, т.к. первая команда NOP состоит из одного машинного цикла и для её выполнения достаточно выполнить один шаг. Таким образом, нажимая клавишу ШГ, можно проследить ход выполнения всей введённой последовательности команд. Результаты выполнения этого задания сведены в таблицу:

Команда	Шина адреса	Шина данных Код Данные	№ шага	Порядковый № машинного цикла	Примечание
NOP	800	00	0	1	чтение кода команды
MOV A,B	801	78	1	1	выполнение пред. команды чтение кода команды
ADD C	802	81	2	1	выполнение пред. команды чтение кода команды
RRC	803	0F	3	1	выполнение пред. команды чтение кода команды
ANI 0FH	804	E6	4	1	выполнение пред. команды чтение кода команды
	805	0F	5	2	чтение 2 байта команды
PUSH H	806	E5	6	1	выполнение пред. команды чтение кода команды
	BC9	12	7	2	запись H (12) по УС
	BC8	34	8	3	запись L (34) по УС-1
LXI H, 900H	807	21	9	1	выполнение пред. команды чтение кода команды
	808	00	10	2	чтение 2 байта команды
	809	09	11	3	чтение 3 байта команды



Команда	Шина адреса	Шина данных Код Данные	№ шага	Порядковый № машинного цикла	Примечание
MOV M,A	80A	77	12	1	выполнение пред. команды
	900	03	13	2	чтение кода команды запись в ячейку памяти по адресу 900H содержимого регистра A (03)
MVI A, 0FFH	80B	3E	14	1	выполнение пред. команды
	80C	FF	15	2	чтение кода команды чтение 2 байта команды
SUB M	80D	96	16	1	выполнение пред. команды
	900	03	17	2	чтение кода команды чтение ячейки памяти по адресу 900H
POP H	80E	E1	18	1	выполнение пред. команды
	BC8	34	19	2	чтение кода команды запись в H из стека по UC = BC8H
	BC9	12	20	3	запись в L из стека по UC=UC+1=BC9H
JMP 800H	80F	C3	21	1	вып. пред. команды
	810	00	22	2	чтение кода команды чт. мл. байта адр. пер.
	811	08	23	3	чт. ст. байта адр. перехода

Последняя выполняемая команда передаёт управление на начало введённой последовательности команд. Чтобы закончить выполнение данного задания, нажмите клавишу СБРОС, УМК перейдёт в режим приёма команд.

### **Задание на самостоятельную работу к разделу 2.1.**

1. Для следующей последовательности команд запишите машинные коды.

Адрес	Команда	Машинный код	Комментарий
800	LXI B, 940H	01 40 09	; загрузка BC = 940H
803	LXI D, 960H	11 60 09	; загрузка DE = 960H
806	MVI L, 10H	2E 10	; загрузка L = 10H
808	LDAX B	0A	; загрузка A=M по адресу BC
809	STAX D	12	; запись M=A по адресу DE
80A	INX B	03	; BC = BC+1
80B	INX D	13	; DE = DE+1
80C	DCR L	2D	; L = L-1
80D	JMP 800H	C3 00 08	; переход по адресу 800H

2. Выполните действия, аналогичные пунктам 2.1.2.1. и 2.1.2.2. для этих команд.

Примечание: закончить выполнение последовательности команд командой JMP 800H (на шине адреса должен появиться опять адрес 800H).

## **2.2. Режим работы УМК по командам.**

- 2.2.1. Установка режима работы УМК по командам.

Для этого отожмите клавишу КМ/ЦК.

- 2.2.2. Выполнение последовательности команд.

Рассмотрим работу УМК в покомандном режиме на примере последовательности команд:

Адрес	Команда	Машинный код	Комментарий
800	LXI H, 900H	21 00 09	; загрузка HL=900H (адрес M)
803	MVI M, 0AAH	36 AA	; запись M=0AA по адресу HL
805	MVI A, 55H	3E 55	; загрузка A = 55H
807	ADD M	86	; A=A+M по адресу HL
808	STA 901H	32 01 09	; запись M=A по адресу 901H
80B	LXI D, 902H	11 02 09	; загрузка DE=902H
80E	LDAX D	1A	; загрузка A=M по адресу DE
80F	ORA M	B6	; A = A OR M по адресу HL
810	LXI B, 903H	01 03 09	; загрузка BC = 903H
813	STAX B	02	; запись M=A по адресу BC
814	JMP 800H	C3 00 08	; переход на адрес 800H

2.2.2.1. Выполните действия согласно пунктам 2.1.2.1.,2.1.2.2. Результаты выполнения этого задания сведены в таблицу:

Команда	Шина адреса	Шина данных	№ шага	Примечание
LXI H, 900H	800	21	0	; чтение кода команды
MVI M, 0AAH	803	36	1	; чтение кода команды ; выполнение пред. команды
MVI A, 55H	805	3E	2	; чтение кода команды ; выполнение пред. команды
ADD M	807	86	3	; чтение кода команды ; выполнение пред. команды
STA 901H	808	32	5	; чтение кода команды ; выполнение пред. команды
LXI D, 902H	80B	11	6	; чтение кода команды ; выполнение пред. команды
LDAX D	80E	1A	7	; чтение кода команды ; выполнение пред. команды

Команда	Шина адреса	Шина данных	№ шага	Примечание
ORA M	80F	B6	8	; чтение кода команды ; выполнение пред. команды
LXI B, 903H	811	01	9	; чтение кода команды ; выполнение пред. команды
STAX B	813	02	10	; чтение кода команды ; выполнение пред. команды
JMP 800H	814	C3	11	; чтение кода команды ; выполнение пред. команды

После выполнения последней команды (JMP 800H) управление будет передано вновь на начало программы. Для завершения выполнения задания нажмите кнопку СБРОС.

### Задание на самостоятельную работу к разделу 2.2.

1 Запишите машинные коды для следующей последовательности команд:

Адрес	Команда	Машинный код	Комментарий
800	LXI H, 900H	21 00 09	; загрузка HL=900H (адрес)
803	LXI B, 920H	01 20 09	; загрузка BC = 920H (адрес)
806	LXI D, 940H	11 40 09	; загрузка DE = 940H (адрес)
809	LDAX B	0A	; загрузка A=M по адресу BC
80A	SUB M	96	; A=A2-M по адресу H
80B	STA 950H	32 50 09	; запись M=A по адресу 950H
80E	LDAX D	1A	; загрузка A=M по адресу DE
80F	ANA M	A6	; A = A AND M по адресу HL
810	RAL	17	; сдвиг влево на 1 регистр A
811	STA 951H	32 51 09	; запись M=A по адресу 951H
814	LDA 950H	3A 50 09	; загрузка A=M по адресу 950H
817	MOV L, A	6F	; пересылка L ← A

Адрес	Команда	Машинный код	Комментарий
818	LDA 951H	3A 51 09	; загрузка A=M по адресу 951H
81B	MOV H, A	67	; пересылка H ← A
81C	SHLD 952H	22 52 09	; запись HL по адресу 952H-953H
81F	JMP 800H	C3 00 08	; переход на адрес 800H

2 Выполните действия, аналогичные пункту 2.2.2.1. для этой последовательности команд. Результаты сведите в таблицу.

### **Контрольные вопросы.**

1. Какие режимы работы предусмотрены для данного микропроцессорного комплекта?
2. Какие действия необходимо предпринять для установки режима работы по машинным циклам?
3. Для чего используется режим работы по командам?
4. Какой из режимов позволяет более детально изучить работу команд микропроцессора?

## Лабораторная работа №3

### Изучение регистров микропроцессора. Команды загрузки регистров. Команды пересылки.

**Цель работы:** Изучение видов регистров микропроцессора, освоение их использования при программировании (загрузка регистров различных типов, пересылка данных из одного регистра в другой).

#### Теоретический обзор.

В микропроцессоре КР 580ВМ80А для программирования доступны следующие регистры:

- 16-разрядный счётчик команд (PC), который содержит адрес выполняемой команды;
- 16-разрядный регистр - указатель стека (SP), который определяет адрес специализированной области ОЗУ - стека;
- 8-разрядный регистр-накопитель (A), который используется для хранения и накопления результата в арифметических, логических операциях, а также операциях ввода-вывода и сдвига. Кроме того, он может быть использован в качестве регистра общего назначения для хранения данных;
- шесть 8-разрядных регистров общего назначения B, C, D, E, H, L;
- 8-разрядный регистр признаков PSW (F), который содержит биты условий:
  - C - перенос,
  - AC - вспомогательный перенос ,
  - S - знак,
  - Z - ноль,
  - P - чётность.

Эти биты устанавливаются в зависимости от результата операции при выполнении арифметических, логических команд, команд сдвига и сравнения.

Распределение битов условий в байте признаков следующее:

7 разряд

← 0 разряд

S	Z	0	AC	0	P	1	C
---	---	---	----	---	---	---	---

Регистры общего назначения могут использоваться для манипуляции 16-разрядными данными. Для этого регистры объединяются в пары следующим образом: В - С, D - E, H - L. В каждой паре первый регистр используется для хранения старшего байта (например, В), а второй - для хранения младшего байта (например, С). В пару объединяются также регистры PSW и А.

### Содержание работы.

- 3.1. Команды загрузки регистров общего назначения.
- 3.2. Команды загрузки регистров 16-разрядными (двухбайтовыми) данными.
- 3.3. Команды загрузки регистра указателя стека.
- 3.4. Команды пересылки.
- 3.5. Команды загрузки счётчика команд.
- 3.6. Просмотр регистра признаков.

### Выполнение работы.

#### 3.1. Команды загрузки регистров общего пользования.

Общий вид команды:

MVI R, B

где R - идентификатор одного из регистров А, В, С, D, E, H, L;

B - непосредственный операнд, байтовое число.

- 3.1.1. Запишите в память, начиная с адреса 800H, следующую последовательность команд:

Адрес	Команда	Машинный код	Комментарий
800	MVI A, 0	3E 00	; загрузка регистра А = 00H
802	MVI B, 1	06 01	; загрузка регистра В = 01H
804	MVI C, 2	0E 02	; загрузка регистра С = 02H
806	MVI D, 3	16 03	; загрузка регистра D = 03H
808	MVI E, 4	1E 04	; загрузка регистра E = 04H
80A	MVI H, 5	26 05	; загрузка регистра H = 05H

80С	MVI L, 6	2E 06	; загрузка регистра L = 06H
-----	----------	-------	-----------------------------

3.1.2. Выполните эту последовательность, используя команду:

СТ800 80ЕВП

(на дисплее появится адрес останова 80Е).

3.1.3. Используя команду ПРОСМОТР И МОДИФИКАЦИЯ РЕГИСТРОВ, посмотрите содержимое регистров общего назначения А, В, С, D, E, H, L.

Значения регистров должны быть следующими:

A = 00

B = 01

C = 02

D = 03

E = 04

H = 05

L = 06

## 3.2. Команды манипуляции 16-разрядными данными.

Общий вид команды:

LXI R, <BHVL>

где R - идентификатор регистра B, D, H;

BH - старший байт 16-разрядного операнда;

VL - младший байт 16-разрядного операнда.

3.2.1. Запишите в память, начиная с адреса 800H, коды следующей последовательности команд:

Адрес	Команда	Машинный код	Комментарий
800	LXI B, 3132H	01 32 31	; загрузка пары регистров BC = 3132H
803	LXI D, 3334H	11 34 33	; загрузка пары регистров DE = 3334H
806	LXI H, 3536H	21 36 35	; загрузка пары регистров HL = 3536H

Примечание: в памяти располагается сначала младший байт операнда, затем - старший.



3.2.2. Выполните эту последовательность команд:

СТ800 809ВП

3.2.3. Посмотрите содержимое регистров. Значения регистров должны быть следующими:

B = 31H

C = 32H

D = 33H

E = 34H

H = 35H

L = 36H

### 3.3. Команды загрузки регистра указателя стека

3.3.1. Команда непосредственной загрузки регистра указателя стека имеет вид:

LXI SP, BHBL

где BHBL - значение операнда;

BH - старший байт;

BL - младший байт.

Запишите в память по адресу 800H коды команды:

Адрес	Команда	Машинный код	Комментарий
800	LXI SP, 0B10H	31 00 0B	; загрузка указателя стека SP=0B10H

3.3.2. Выполните эту команду:

СТ800 803ВП

3.3.3. Посмотрите содержимое регистра указателя стека. Значение старшего байта должно быть SH = 0BH, значение младшего байта SL = 10H.

3.3.4. Команда косвенной загрузки регистра указателя стека имеет вид:

SPHL

По этой команде в указатель стека загружается содержимое регистровой пары HL. Поэтому, чтобы в указатель стека загрузить, например, число 0B30H, его предварительно надо загрузить в регистровую пару HL.

3.3.5. Запишите в память по адресу 800H коды следующих команд:

Адрес	Команда	Машинный код	Комментарий
800	LXI H, 0B30H	21 30 0B	; загрузка HL = 0B30H
803	SPHL	F9	; загрузка SP = HL

3.3.6. Выполните эти команды:

CT800 804BP

3.3.7. Посмотрите содержимое регистра указателя стека. Значение старшего байта должно быть SH = 0BH, значение младшего байта - SL = 30H.

### 3.4. Команды пересылки.

Общий вид команды:

MOV R1, R2

где R1 - идентификатор регистра получателя: A, B, C, D, E, H, L;

R2 - идентификатор регистра источника: A, B, C, D, E, H, L.

3.4.1. Запишите в память, начиная с адреса 800H, коды следующей последовательности команд:

Адрес	Команда	Машинный код	Комментарий
800	MVI A, FFH	3E FF	; загрузка регистра A = FF
802	MOV B, A	47	; пересылка B ← A
803	MOV C, B	48	; пересылка C ← B
804	MOV D, C	51	; пересылка D ← C
805	MOV E, D	5A	; пересылка E ← D
806	MOV H, E	63	; пересылка H ← E
807	MOV L, H	6C	; пересылка L ← H

3.4.2. Выполните эту последовательность команд:

CT800 808BP

3.4.3. Просмотрите содержимое регистров. Их значения должны быть равны FFH.

### 3.5. Команда загрузки счётчика команд PCHL

По этой команде в счётчик команд записывается содержимое пары регистров HL. Таким образом, для того, чтобы загрузить в счётчик команд адрес 0900H, необходимо сначала это число загрузить в регистровую пару HL и затем выполнить команду PCHL.

3.5.1. Запишите в память по адресу 800H коды следующих команд:

Адрес	Команда	Машинный код	Комментарий
800	LXI H, 0900H	21 00 09	; загрузка HL = 0900H
803	PCHL	E9	; загрузка счётчика команд PC = HL. Переход на адрес 0900H

#### Задание на самостоятельную работу.

1. Напишите и выполните программу загрузки регистров по следующим правилам:

1 вариант	2 вариант	3 вариант
B := F0H	B := 23H	B := BBH
C := 33H	C := 45H	C := CCH
D := EEH	D := 10H	D := DDH
E := AAH	E := 62H	E := EEH
H := 00H	H := A5H	H := 12H
L := 19H	L := 97H	L := 34H
A := FFH	A := 0EH	A := AAH

Проверьте правильность выполнения программы.

2. Напишите и выполните программу загрузки регистровых пар:

1 вариант	2 вариант	3 вариант
BC := FFFFH	BC := 789AH	BC := BBCCH
DE := 0123H	DE := F0E9H	DE := DDEEH
HL := 55AAH	HL := 56DCH	HL := 1234H

Проверьте правильность работы программы.

3. Напишите и выполните программу загрузки регистра указателя стека:

а) командой непосредственной загрузки:

SP := 0820H

SP := 0840H

SP := 0AFFH

б) используя команду косвенной загрузки:

SP := 0825H

SP := 089FH

SP := 0A5AH

Для проверки результатов выполняйте программу покомандно с остановкой после каждой команды (см. Лаб. раб. №1, п.1.3.3.) и смотрите значение регистра SP.

4. Напишите и выполните программу пересылки, предварительно загрузив регистры указанными значениями:

1 вариант	2 вариант	3 вариант
B ← A (значением 00)	D ← H (55H)	D ← H (BCH)
C ← L (22H)	E ← C (81H)	E ← L (03H)
H ← B (EEH)	L ← D (FFH)	A ← B (7AH)

Проверьте правильность выполнения этих программ.

5. Напишите и выполните программу перехода на адрес:

1 вариант	2 вариант	3 вариант
0860H	0923H	0A00H
0810H	091F	0A21H
085FH	0953	0A3FH

с адреса 800H.

### Контрольные вопросы.

1. Какие типы регистров используются в микропроцессоре?
2. Биты каких условий содержит регистр признаков?
3. Как размещается в памяти 16-разрядный операнд?
4. Какие команды используются для загрузки указателя стека и счётчика команд?
5. Какова структура команды пересылки?

## Лабораторная работа №4

### Изучение методов адресации памяти. Команды работы с памятью.

**Цель работы:** Изучение непосредственного и косвенного методов адресации к памяти. Освоение команд чтения и записи в память при использовании различных способов адресации.

#### Теоретический обзор.

Память представляет собой последовательность ячеек размером в один байт. Каждая ячейка имеет свой адрес в диапазоне от 0 до 65535. Для удобства обычно используется шестнадцатеричное значение адреса, тогда диапазон адресации составляет 0000H - FFFFH.

В микропроцессорной системе применяется два метода адресации к памяти : непосредственный и косвенный.

При **непосредственной адресации** адрес ячейки памяти указывается в самой команде во втором и третьем байтах команды. В общем виде команда выглядит следующим образом:

КОП АНАL

где КОП - код операции (чтение или запись),

АНАL - адрес ячейки памяти,

АН - старший байт адреса,

AL - младший байт адреса.

В памяти такая команда будет размещена в следующем порядке:

КОП

AL

АН,

то есть после байта кода операции располагается сначала младший байт адреса, а затем - старший.

**Косвенная адресация** предполагает, что адрес ячейки памяти будет располагаться в регистровых парах HL, BC, DE. Для каждой конкретной команды работы с памятью закреплена своя регистровая пара. Таким образом, прежде чем

выполнить такую команду, необходимо сначала задать адрес в соответствующей регистровой паре. Например,

LXI H, 800H

MOV M, A ; запись в память регистра A по адресу HL

или

LXI D, 900H

STAX D ; запись в память регистра A по адресу DE

### Содержание работы

- 4.1. Команды записи в память по непосредственному адресу.
- 4.2. Команды чтения памяти по непосредственному адресу.
- 4.3. Команды чтения/записи памяти при адресации через регистровую пару HL.
- 4.4. Команды чтения/записи памяти при адресации через регистр. пары BC, DE.

### Выполнение работы

#### 4.1. Команды непосредственной записи в память.

Существуют две команды непосредственной записи в память:

STA ANAL	запись в память по непосредственному адресу ANAL содержимого регистра A.
SHLD ANAL	запись в память содержимого регистровой пары HL. Причём по адресу ANAL будет записано содержимое регистра L, а по адресу ANAL+1 будет записано содержимое регистра H.

4.1.1. Запишите в память, начиная с адреса 800H, коды следующих команд:

Адрес	Команда	Машинный код	Комментарий
800	STA 880H	32 80 08	; запись в память содержимого регистра A по адресу 880H

803	SHLD 890H	22 90 08	; запись в память содержимого регистра L по адресу 890H, содержимого регистра H по адресу 891H.
-----	-----------	----------	---

4.1.2. Посредством команды монитора ПРОСМОТР И МОДИФИКАЦИЯ РЕГИСТРОВ установите следующие значения регистров:

A = C7H, H = 55H, L = AAH.

Выполните введённые команды:

ST800 806ВП

4.1.3. Посмотрите содержимое ячеек памяти с адресами 880H, 890H, 891H.

Их значения должны быть:

880H = C7H

890H = AAH

891H = 55H

### Задание на самостоятельную работу к разделу 4.1.

1. Напишите и выполните программу записи данных в память из регистра A в соответствии с нижеприведённой таблицей. Для этого используйте команду загрузки регистра A и команду записи в память регистра A по непосредственному адресу.

Адрес	900	905	90C	912	916	91F
Данные	00	01	02	04	08	10

2. Проверьте правильность работы программы.
3. Напишите и выполните программу записи данных из регистровой пары HL в соответствии с нижеприведённой таблицей. Для этого используйте команду загрузки регистровой пары HL и команду записи в память регистровой пары HL по непосредственному адресу.

Адрес	920	921	925	936	92C	92D
Данные	31	32	4F	50	FD	01

4. Проверьте правильность работы программы.

5. Напишите и выполните программу записи данных в память в соответствии с таблицей:

Адрес	940	941	943	947	94A	94B	94F	951
Данные	00	01	03	07	0A	0B	0F	51

6. Проверьте правильность работы программы.

#### 4.2. Команды чтения памяти по непосредственному адресу.

Аналогично командам непосредственной записи существуют две команды непосредственного чтения из памяти:

LDA ANAL	чтение памяти по непосредственному адресу ANAL в регистр A
LHLD ANAL	чтение памяти по непосредственному адресу ANAL в регистровую пару HL. При этом в регистр L будет записано содержимое ячейки с адресом ANAL, в регистр H - содержимое ячейки с адресом ANAL+1.

- 4.2.1. Запишите в память по адресу 800H коды следующих команд:

Адрес	Команда	Машинный код	Комментарий
800	LDA 880H	3A 80 08	; чтение в регистр A содержимого ячейки с адресом 880H
803	LHLD 890H	2A 90 08	; чтение в регистр L содержимого ячейки с адресом 890H, в регистр H содержимого ячейки с адресом 891H

- 4.2.2. Выполните эти команды:

СТ800 806ВП

- 4.2.3. Посмотрите содержимое регистров. Оно должно быть следующим:

A = C7H

H = 55H

L = AAH



## Задание на самостоятельную работу к разделу 4.2.

1. Напишите и выполните программу загрузки регистров В, С, D, E, H, L из памяти в соответствии с нижеприведённой таблицей. Используйте команды чтения памяти в регистр А по непосредственному адресу и команды пересылки.

Адрес	900	905	90С	912	916	91F
Регистр	В	С	D	E	H	L
Результат	00	01	02	04	08	10

2. Проверьте правильность работы программы.
3. Напишите и выполните программу загрузки регистров В, С, D, E, H, L из памяти в соответствии с нижеприведённой таблицей. Используйте команды чтения памяти в регистровую пару HL по непосредственному адресу и команды пересылки.

Адрес	920	921	925	926	92С	92D
Регистр	В	С	D	E	H	L
Результат	31	32	4F	50	1F	01

4. Проверьте правильность работы программы
5. Напишите и выполните программу перезаписи данных из одних ячеек памяти в другие в соответствии с таблицей:

Адрес ячейки	исх. 940	941	943	947	94A	94B	94F	951
Адрес запис. ячейки	960	961	963	967	96A	94B	94F	971
Результат	00	01	03	07	0A	0B	0F	51

6. Проверьте правильность работы программы

### 4.3. Команды чтения/записи памяти при адресации через регистровую пару HL.

Общий вид команды:

MOV M, R	запись в память содержимого регистра
MOV R, M	загрузка регистра из памяти

где R - регистр общего пользования: A, B, C, D, E, H, L.

4.3.1. Запишите в память, начиная с адреса 800H, коды следующей программы:

Адрес	Команда	Машинный код	Комментарий
800	MVI A, 0AAH	3E AA	; загрузка регистров
802	MVI B, 0BBH	06 BB	
804	MVI C, 0CCH	0E CC	
806	MVI D, 0DDH	16 DD	
808	MVI E, 0EEH	1E EE	
80A	LXI H, 900H	21 00 09	; загрузка HL = 900H (адрес M)
80D	MOV M, A	77	; запись M=A, по адресу HL
80E	LXI H, 901H	21 01 09	; и т. д.
811	MOV M, C	71	
812	LXI H, 902H	21 02 09	
815	MOV M, B	70	
816	LXI H, 903H	21 03 09	
819	MOV M, E	73	
81A	LXI H, 904H	21 04 09	
81D	MOV M, D	72	
81E	LXI H, 905H	21 05 09	
821	MOV M, H	74	
822	LXI H, 906H	21 06 09	
825	MOV M, L	75	

4.3.2. Выполните эту последовательность команд:

### СТ800 826ВП

4.3.3. Проверьте правильность выполнения программы. Значения ячеек памяти должны быть следующими:

900H = AAH

901H = CCH

902H = BBH

903H = EEH

904H = DDH

905H = 09H

906H = 06H

4.3.4. Запишите в память коды следующей программы:

Адрес	Команда	Машинный код	Комментарий
800	LXI H, 900H	21 00 09	; загрузка HL = 900H (адрес M)
803	MOV E, M	5E	; чтение E = M по адресу HL
804	LXI H, 901H	21 01 09	; и т. д.
807	MOV D, M	56	
808	LXI H, 902H	21 02 09	
80B	MOV C, M	4E	
80C	LXI H, 903H	21 03 09	
80F	MOV B, M	46	
810	LXI H, 904H	21 04 09	
813	MOV A, M	7E	
814	LXI H, 905H	21 05 09	
817	MOV H, M	66	
818	LXI H, 906H	21 06 09	
81B	MOV L, M	6E	

4.3.5. Выполните эту последовательность команд:

### СТ800 81СВП

4.3.6. Проверьте содержимое регистров. Оно должно быть следующим:

регистр A = DDH

регистр B = EEH

регистр C = BBH

регистр D = CCH

регистр E = AАН

регистр H = 09H

регистр L = 06H

### **Задание на самостоятельную работу к разделу 4.3.**

1. Напишите и выполните программу записи в память содержимого регистров в соответствии с таблицей:

Регистр	Адрес ячейки памяти	Содержимое
A	920	FF
B	921	EF
C	922	DF
D	923	CF
E	924	BF
H	925	09
L	926	26

2. Напишите и выполните программу чтения содержимого памяти и заполните таблицу:

Регистр	Адрес ячейки памяти	Содержимое	Должно быть
B	920		FF
C	921		EF
D	922		DF
E	923		CF
A	924		BF
H	925		09
L	926		26

### **4.4. Команды чтения/записи при адресации через регистровые пары BC, DE.**

STAX B	запись содержимого регистра A в память, адрес в регистровой паре BC
STAX D	запись содержимого регистра A в память, адрес в регистровой паре DE
LDAX B	чтение содержимого памяти в регистр A, адрес в регистровой паре BC
LDAX D	чтение содержимого памяти в регистр A, адрес в регистровой паре DE

4.4.1. Запишите в память, начиная с адреса 800H, коды программы:

Адрес	Команда	Машинный код	Комментарий
800	LXI B, 900H	01 00 09	; загрузка BC = 900H
803	MVI A, 0FH	3E 0F	; загрузка A = 0FH
805	STAX B	02	; запись M=A по адресу BC
806	LXI D, 910H	11 10 09	; загрузка DE = 910H
809	MVI A, 0F0H	3E F0	; загрузка A = F0H
80B	STAX D	12	; запись M=A по адресу DE

4.4.2. Выполните эту последовательность команд:

CT800 80СВП

4.4.3. Посмотрите содержимое ячеек памяти 900H и 910H. Оно должно быть следующим:

900H = 0FH

910H = F0H

4.4.4. Запишите в память, начиная с адреса 800H, коды программы:

Адрес	Команда	Машинный код	Комментарий
800	LXI D, 900H	11 00 09	; загрузка DE = 900H
803	LDAX D	1A	; чтение A=M по адресу DE
804	MOV L, A	6F	; пересылка $L \leftarrow A$

805	LXI B, 910H	01 10 09	; загрузка BC = 910H
808	LDAX B	0A	; чтение A=M по адресу BC
809	MOV H, A	67	; пересылка H←A

4.4.5. Выполните эту последовательность команд:

CT800 80ABП

4.4.6. Посмотрите содержимое регистров H, L. Оно должно быть следующим:

H = F0H

L = 0FH

### Задание на самостоятельную работу к разделу 4.4.

1. Напишите и выполните программу записи данных в две области памяти, используя для адресации регистровую пару BC и регистровую пару DE в соответствии с таблицей:

Адреса 1 области памяти	910	912	915	919	921	927
Адреса 2 области памяти	930	932	935	939	941	947
Данные	10	12	15	19	21	27

2. Проверьте правильность работы программы.

3. Напишите и выполните программу перезаписи данных из одной области памяти (адресуйте через BC) в другую область памяти (DE) в соответствии с таблицей:

Адреса исходной области памяти	930	932	935	939	941	947
Адреса записываемой области памяти	960	961	962	963	964	965

Результат	10	12	15	19	21	27
-----------	----	----	----	----	----	----

4. Проверьте правильность работы программы.

### **Контрольные вопросы.**

1. Какие методы адресации существуют в данной микропроцессорной системе?
2. Как размещается в памяти команда, использующая непосредственную адресацию?
3. Перечислите команды непосредственной записи в память и чтения из памяти.
4. Действие каких команд осуществляется через регистровые пары BC, DE?
5. Какие регистры используются в команде пересылки?

## **Лабораторная работа №5**

### **Изучение арифметических команд**

**Цель работы:** Изучение команд микропроцессора, выполняющих арифметические действия с операндами (сложение, вычитание, инкремент, декремент). Применение этих команд для вычисления арифметических выражений.

#### **Задание на лабораторную работу:**

изучить:

- 1 Команды сложения 8-разрядных чисел.
- 2 команды вычитания 8-разрядных чисел.
- 3 Команды двойного сложения.
- 4 Команды инкремента.
- 5 Команды декремента.

#### **Ход работы**

В микропроцессорной системе КР580ВМ80А предусмотрены следующие команды двоичной арифметики:

- сложение 8-разрядной чисел;
- сложение 16-разрядной чисел;
- вычитание 8-разрядных чисел;
- инкремент;
- декремент.

Все арифметические операции с 8-разрядными операндами предполагают, что один из операндов размещается в регистре-аккумуляторе, а другой либо в регистре, либо в памяти (при этом адрес ячейки задаётся в регистровой паре HL), либо является непосредственным числом, заданным в самой команде. Вычитание производится всегда из регистра-аккумулятора. Результат арифметической операции записывается в аккумулятор. Кроме того, по результату арифметических операций сложения и вычитания устанавливаются биты признаков: C - переноса, Z- нуля, S - знака, P - чётности, AC - вспомогательного переноса.

Команды сложения 16-разрядных чисел, так называемые команды двойного сложения, предусматривают, что один из операндов находится в регистровой паре HL, а второй - либо в DE, либо в BC. Результат записывается в HL. Кроме того, по результату операции устанавливается, либо сбрасывается бит переноса - C.

Команды инкремента увеличивают содержимое регистров, ячейки памяти по адресу в HL, и регистровых пар на единицу. Команда инкремента регистра и памяти изменяет биты признаков: Z, S, P, AC. Инкремент регистровой пары не затрагивает биты признаков.

Команды декремента уменьшают содержимое регистров, ячейки памяти по адресу в HL, и регистровых пар на единицу. Биты признаков изменяются аналогично команде инкремента.

#### **1 Команды сложения 8-разрядных чисел.**



D R	AD	сложение с регистром: A, B, C, D, E, H, L;
D M	AD	сложение с ячейкой памяти (адрес в HL);
I B	AD	сложение с непосредственным числом, B - байт;
C R	AD	сложение с регистром A, B, C, D, E, H, L плюс бит переноса C ;
C M	AD	сложение с ячейкой памяти (адрес в HL) плюс бит переноса C ;
I B	AC	сложение с непосредственным числом, B - байт, плюс бит переноса C;

5.1.1. Запишите в память, начиная с адреса 800H, коды программы, реализующей выражение:

$$A = A + B + M + 1$$

Адрес	A да	Команда	Машинный код	Комментарий
00	8	ADD B	80	; A = A+B
01	8	LXI H, 900H	21 00 09	; загрузка HL = 900H (адрес M)
04	8	ADD M	86	; A = A + M
05	8	ADI 1	C6 01	; A = A + 1

5.1.2. Выполните программу, предварительно задавая исходные значения в соответствии с таблицей, и проверьте полученные результаты (результат операции в регистре A, биты условий в регистре F):

СТ800 807ВП

A (исх.)	00	00	00	F0	FF	5	5
B	00	02	10	0E	00	A	A
M	00	03	45	00	00	F	F
A (рез.)	01	06	56	FF	00	F	F
F	02	06	06	86	57	8	8
						6	

5.1.3. Запишите в память, начиная с адреса 800H, коды программы сложения 16-разрядных чисел, используя команды 8-разрядного сложения:

$$HL = DE + BC$$

Адрес	A да	Команда	Машинный код	Комментарий
00	8	MOV A, C	79	

01	8 E	ADD	83	; сложение младших байтов, установка бита переноса в случае переполнения
02	8 V L, A	MO	6F	; занесение младшего байта в L
03	8 V L, B	MO	78	
04	8 D	ADC	8A	; сложение старших байт с учетом переноса
05	8 V H, A	MO	67	; занесение старшего байта в H

5.1.4 Выполните программу, предварительно задавая исходные значения в соответствии с таблицей. Проверьте полученные результаты.

СТ800 806ВП

BC	00	02	F0	81	8	F
01		C5	00	37	09F	FFF
DE	00	03	0F	72	8	0
FE		F1	FF	D9	121	001
HL	00	06	FF	F4	0	0
FF		B6	FF	10	1C0	000
F	46	06	86	82	0	5
					3	7

### Задание на самостоятельную работу к разделу 5.1.

1. Напишите и выполните программу, реализующую вычисление выражения

$C = D + E$  и заполните таблицу:

D	10	FF	C7	1	A	A	E
E	80	01	8	9	4	5	F
C				9	5	0	
F							

2. Напишите и выполните программу сложения двух ячеек памяти по правилу

$M1 = M2 + M3$  и заполните таблицу.

Адрес M1 = 900H

Адрес M2 = 901H

Адрес M3 = 902H

M	0	FE	D5	2	2	6	19
2				2	1		
M	F0	02	C2	B	B	9	33
3				B	5		
M							
1							
F							

3. Напишите и выполните программу сложения

$$HL = BC + E + 4000H$$

и заполните таблицу. Исходные значения возьмите произвольно.

C	

## 5.2. Команды вычитания 8 - разрядных чисел

B R	SU	вычитание регистра: A, B, C, D, E, H, L;
B M	SU	вычитание ячейки памяти (адрес HL);
I B	SU	вычитание непосредственного числа, B - байт;
B R	SB	вычитание регистра: A, B, C, D, E, H, L минус бит переноса C;
B M	SB	вычитание ячейки памяти ( адрес HL ) минус бит переноса C;
B	SBI	вычитание непосредственного числа минус бит переноса C.

5.2.1. Запишите в память, начиная с адреса 800H, коды программы, реализующей вычисление выражения

$$A = A - B - M - 1;$$

рес	Ад	Команда	Машинный код	Комментарий
	800	SUB B	90	; A = A - B
	801	LXI H, 900H	21 00 09	; загрузка HL = 900H (адрес M)
	804	SUB M	96	; A = A - M
	805	SBI 1	DE 01	; A = A - 1

5.2.2. Выполните программу, предварительно задав исходные значения в соответствии с таблицей. Проверьте полученные результаты.

СТ800 807ВП

(исх.)	A	FF	00	01	5	2	0	05
	B	01	FF	01	0	2	0	06
	M	01	00	00	4	0	0	FF
(рез.)	A	FC	00	FF	0	0	F	FF
	F	96	56	87	6	5	7	87

5.2.3. Запишите в память, начиная с адреса 800H, коды программы вычитания 16-разрядных чисел:

HL = DE - BC

Адрес	А	Команда	Машинный код	Комментарий
0	80	MOV A, E	7B	
1	80	SUB C	91	; вычитание младших байтов E - C
2	80	MOV L, A	6F	; если E < C, перенос = 1
3	80	MOV A, D	7A	
4	80	SBB B	98	; вычитание ст. байтов с учётом переноса: D - B - бит переноса
5	80	MOV H, A	67	

5.2.4. Выполните программу, предварительно задавая исходные значения в соответствии с таблицей. Проверьте полученные результаты.

СТ800 806ВП

DE	0000	FFFF	0FF0	8000
BC	FFFF	0FF0	0001	7FFF
HL	0001	F00F	0FEF	0001
F	47	96	16	46

### Задание на самостоятельную работу к разделу 5.2.

1. Напишите и выполните программу, реализующую вычисление выражения

$C = D - E - 10H$

и заполните таблицу. Исходные значения возьмите произвольно.

D	
E	
C	
F	

2. Напишите и выполните программу вычитания двух ячеек памяти M1 = M2 - M3 и заполните таблицу. Исходные значения возьмите произвольно.

Адрес M1 = 900H

Адрес M2 = 901H

Адрес M3 = 902H

2	M	
3	M	
1	M	
F		

3. Напишите и выполните программу вычитания  $HL = BC - E - 0FFFH$  и заполните таблицу. Исходные значения возьмите произвольно.

C	B	
	E	
L	H	
	F	

### 5.3. Команды двойного сложения.

DAD H	сложение $HL = HL + HL$
DAD B	сложение $HL = HL + BC$
DAD D	сложение $HL = HL + DE$

5.3.1. Запишите в память, начиная с адреса 800H, коды программы, реализующей сложение:

$$HL = BC + DE$$

Адрес	А	Команда	Машинный код	Комментарий
00	8	MOV H, B	60	; пересылка $H \leftarrow B$
01	8	MOV L, C	69	; пересылка $L \leftarrow C$
02	8	DAD D	19	; $HL = HL + DE$

5.3.2. Выполните программу, предварительно задавая исходные значения в соответствии с таблицей. Проверьте полученные результаты.

СТ800 803ВП

C	B	0	7	80	55	EC	FF
	D	7	8	80	AA	13	80
E		FFF	000	00	55	47	00
L	H	7	F	00	FF	FF	7F
	F	4	4	47	46	47	47
		6	6				

5.3.3. Запишите в память, начиная с адреса 800H, коды программы вычитания адреса элемента массива по его порядковому номеру и чтения элемента массива в регистр A (в регистре C находится номер элемента (0 - 256), в HL - базовый адрес массива).

Адрес	А	Команда	Машинный код	Комментарий
00	8	LXI H, 900H	21 09	; загрузка в HL базового адреса массива

03	8	MVI B, 0	06 00	; загрузка B = 0 (старший байт смещения)
05	8	DAD B	09	; HL = HL+ BC
06	8	MOV A, M	7E	; чтение элемента массива в регистр A

5.3.4. Выполните эту программу, предварительно задавая номер элемента массива в регистре C. Заполните таблицу.

C	0	1	5	0	2	9
H (адрес элемента массива)				F		
A						

5.3.5. Сравните полученные результаты (регистр A) с содержимым соответствующих ячеек памяти, используя команду монитора ПРОСМОТР И МОДИФИКАЦИЯ СОДЕРЖИМОГО ЯЧЕЙКИ ПАМЯТИ.

### Задание на самостоятельную работу к разделу 5.3.

1. Напишите и выполните программу заполнения массива по заданному индексу элемента массива в соответствии с таблицей.

Базовый адрес массива	00	9				
Номер элемента	0	0	3	0	1	8 F 1
Адрес элемента						
Содержимое элемента массива	0	0	1	0	2	3 4 5 0

2. Проверьте правильность работы программы.

3. Напишите и выполните программу перезаписи содержимого массива 1, заданного в первом пункте, в массив 2 в соответствии с таблицей:

Базовый адрес массива 2	10	9				
Номер элемента массива 2		2	4			
Адрес элемента массива 2						
Содержимое элемента массива 2						

4. Проверьте правильность работы программы.

### 5.4. Команды инкремента.

R R	IN	увеличение на 1 содержимого регистра: A, B, C, D, E, H, L;
R M	IN	увеличение на 1 содержимого ячейки памяти, адрес M в HL;

X R	IN	увеличение на 1 содержимого пары регистров: BC, DE, HL, SP (указателя стека). В команде указывается идентификатор старшего регистра, например, INX B
-----	----	--

5.4.1. Запишите в память по адресу 800H код команды:

Адрес	Команда	Машинный код	Комментарий
800	INR E	1C	; E = E+1

5.4.2. Выполните данную команду для следующих исходных значений регистра E и проверьте полученные результаты:

СТ800 801ВП

E (исх.)	00	0F	F0	FF
E (рез.)	01	10	F1	00
F	02	12	82	56

5.4.3. Запишите в память, начиная с адреса 800H, коды команд:

Адрес	Команда	Машинный код	Комментарий
800	LXI H, 900H	21 00 09	; загрузка HL = 900H (адрес M)
803	INR M	34	; M = M+ 1

5.4.4. Выполните данную последовательность команд для следующих исходных значений содержимого ячейки памяти. Проверьте полученные результаты.

СТ800 804ВП

M (исх.)	00	0F	F0	FF
M (рез.)	01	10	F1	00
F	02	12	82	56

5.4.5. Запишите в память по адресу 800H код команды:

Адрес	Команда	Машинный код	Комментарий
800	INX D	13	; DE = DE + 1

5.4.6. Выполните команду для следующих исходных значений пары регистров DE. Проверьте полученные результаты.

E (исх.)	0000	0F0F	00FF	FFF
E (рез.)	0001	0F10	0100	0000
F	56	056	56	56

Так как команда инкремента пары регистров не затрагивает биты признаков, значение регистра признаков остаётся равным значению регистра признаков в последнем задании (5.4.4.).

### Задание на самостоятельную работу к разделу 5.4.

1. Напишите и выполните программу заполнения массива памяти (900H - 904H), соответственно, данными (00 - 04), используя команды инкремента пары регистров и регистра.

#### 5.5. Команды декремента

R	DCR	уменьшение на 1 содержимого регистра A, B, C, D, E, H, L;
M	DCR	уменьшение на 1 содержимого ячейки памяти, адрес M в HL;
R	DCX	уменьшение на 1 содержимого пары регистров BC, DE, HL, SP. В команде указывается старшего регистра, например, DCX B

5.5.1. Запишите в память по адресу 800H код команды:

Адрес	Адрес	Команда	Машинный код	Комментарий
800	8	DCR C	0D	; C = C-1

5.5.2. Выполните эту команду для следующих значений регистра C. Проверьте полученные результаты.

СТ800 801ВП

C (исх.)	00	01	10	11
C (рез.)	FF	00	0F	10
F	86	56	06	12

5.5.3. Запишите в память, начиная с адреса 800H, коды команд:

Адрес	Адрес	Команда	Машинный код	Комментарий
800	8	LXI H, 900H	21 00 09	; загрузка HL = 900H (адрес M)
803	8	DCR M	35	; M = M-1

5.5.4. Выполните эти команды для следующих исходных значений содержимого ячейки памяти. Проверьте полученные результаты.

СТ800 804ВП

M (исх.)	00	01	10	11
M (рез.)	FF	00	0F	10
F	86	56	06	12

5.5.5. Запишите в память по адресу 800H код команды:

Адрес	Адрес	Команда	Машинный код	Комментарий
800	8	DCX H	2B	; HL = HL - 1



5.5.6. Выполните команду для следующих исходных значений пары регистров HL. Проверьте полученные результаты.

СТ800 801ВП

HL (исх.)	0000	1000	FFFF	0001
HL (рез.)	FFFF	0FFF	FFFE	0000
F	12	12	12	12

Команда декремента пары регистров не затрагивает биты признаков.

### **Задание на самостоятельную работу к разделу 5.5.**

1. Напишите и выполните программу заполнения массива памяти (90FH-90AH) данными, соответственно, (0FH - 0AH), используя команды декремента пары регистров и регистра.

### **Контрольные вопросы.**

1. Где хранится результат арифметической операции?
2. Как размещаются операнды и результат при выполнении команды двойного сложения?
3. Как изменяется содержимое PSW при выполнении операции декремента содержимого ячейки, регистра, регистровой пары?
4. Перечислите команды сложения и вычитания 8-разрядных чисел.
5. Какие команды используются для изменения на единицу значения операнда?

## Лабораторная работа №9

### Изучение команд безусловного и условного переходов.

**Цель работы:** изучение команд безусловного перехода и условных переходов по признакам Z (ноль), C (перенос), P (чётность), S (знак). Освоение использования этих команд для решения различных практических задач (реализация бесконечного цикла, работа с массивами, с содержимым ячеек памяти).

#### Теоретические основы.

В системе команд микропроцессора КР580ВМ80А предусмотрены команды изменения последовательности выполнения команд для организации циклов, обработки условий, передачи управления и т.д. . Существуют два типа команд перехода: безусловный и условный.

При выполнении команд безусловного перехода осуществляется передача управления по адресу, заданному во втором и третьем байтах команды, либо по адресу, заданному в регистровой паре.

Команды условного перехода выполняются в том случае, если установлен или сброшен соответствующий бит признака, в противном случае команда игнорируется и выполняется следующая за ней команда.

Существуют команды условного перехода для следующих битов признаков:

- бита нуля;
- бита переноса;
- бита знака;
- бита чётности.

Для каждого бита признака предусмотрены две команды перехода: переход по установленному биту признака (равен единице) и переход по сброшенному биту признака (равен нулю).

Соответствие выполнения команды и признаков приведены в таблице 9.1.

Признак Команда	Ноль		Перенос		Чётность		Знак	
	Z		C		P		S	
	1	0	1	0	1	0	1	0
JZ	да	-	-	-	-	-	-	-
JNZ	-	да	-	-	-	-	-	-
JC	-	-	да	-	-	-	-	-
JNC	-	-	-	да	-	-	-	-
JPE	-	-	-	-	да	-	-	-
JPO	-	-	-	-	-	да	-	-
JM	-	-	-	-	-	-	да	-
JP	-	-	-	-	-	-	-	да

Таблица 9.1.

### Содержание работы.

- 9.1. Команды безусловного перехода.
- 9.2. Команды перехода по признаку НОЛЬ.
- 9.3. Команды перехода по признаку ПЕРЕНОС.
- 9.4. Команды перехода по признаку ЧЁТНОСТЬ.
- 9.5. Команды перехода по признаку ЗНАК.

### Выполнение работы.

#### 9.1. Команды безусловного перехода

JMP ADR	безусловный переход по адресу, указанному во 2 и 3 байтах команды
PCHL	безусловный переход по адресу, заданному в HL

**9.1.1.** Запишите в память, начиная с адреса 800H, коды программы, реализующей бесконечный цикл:

Адрес    Команда    Машинный код    Комментарий

800	NOP	00	; пустая команда
801	NOP	00	; пустая команда
802	NOP	00	; пустая команда
803	JMP 800	C3 00 08	; безусловный переход на начало программы

**9.1.2.** Выполните программу:

СТ800ВП

Данная программа будет выполнять бесконечный цикл.

**9.1.3.** Нажмите кнопку ПРЕРЫВАНИЕ.

На дисплее отобразится адрес точки прерывания (в пределах от 800H до 803H).

**9.1.4.** Запишите в память, начиная с адреса 800H, коды программы, осуществляющей переход по адресу, записанному в массиве (адрес является элементом массива). Адрес выбирается в соответствии с заданным индексом массива. Массив адресов размещается в памяти, начиная с адреса 810H и содержит 5 элементов. Индекс задаётся в регистре C.

Адрес	Команда	Машинный код	Комментарий
800	LXI D, 810H	11 10 08	; загрузка DE (нач. адрес таблицы - база)
803	MOV L, C	69	; пересылка L ← C (индекс массива)
804	MOV H, 0	26 00 00	; загрузка HL (индекс элемента массива)
806	DAD H	29	; HL = HL*2 (смещение в таблице)
807	DAD D	19	; HL = база массива + смещение
808	MOV A, M	7E	; пересылка A ← M (мл. байт адреса перехода)
809	INX H	23	
80A	MOV H, M	66	; пересылка A ← M (ст. байт адреса перехода)
80B	MOV L, A	6F	; пересылка L ← A (мл. байт адреса перехода)
80C	PCHL	E9	; переход, загрузка PC = HL
810	DW 900H	00 09	; массив адресов перехода
812	DW 902H	02 09	
814	DW 904H	04 09	
816	DW 906H	06 09	
Адрес	Команда	Машинный код	Комментарий

**9.1.5.** Выполните программу поэтапно, предварительно задавая исходные значения и адрес точки останова (второй адрес в команде монитора СТАРТ ПРОГРАММЫ) в соответствии с таблицей, и заполните свободные места в ней.

## СТ800 ТОВП

С	00	01	02	03	04
T01 = 807H смещение (HL)					
T02 = 808H адрес элемента массива (HL)					
T03 = 80CH адрес перехода (HL)					
T04	900	902	904	906	908

### Задание на самостоятельную работу к разделу 9.1.:

1. Напишите и выполните программу, реализующую бесконечный цикл суммирования содержимого массива памяти с адресами (0-3FFH).

2. Напишите и выполните программу, осуществляющую переход по адресу, который вычисляется по формуле:

$$AP = BA + I * 16, \text{ где}$$

AP - адрес перехода;

BA - базовый адрес, BA = 920H;

I - индекс, задаётся в регистре E.

Устанавливая точки останова в выполняемой программе, заполните таблицу:

I=E	0	2	5	7	10
CM = I * 16					
AP					

### 9.2. Команды перехода по признаку Z - НОЛЬ.

JZ ADR	переход, если $Z = 1$ ;
JNZ ADR	переход, если $Z = 0$ .

**9.2.1.** Запишите в память, начиная с адреса 800H, программу заполнения 10H ячеек памяти нулями:

Адрес	Команда	Машинный код	Комментарий
800	MVI C, 10H	0E 10	; загрузка C=10H (длина массива)
802	LXI H, 900H	21 00 09	; загрузка HL=900H (начальный адрес массива)
805	MVI M, 0	36	; загрузка M=0
807	INX H	23	; HL=HL+1 (следующий адрес)
808	DCR C	0D	; C=C-1 (длина массива)
809	JNZ 805H	C2 05 08	; переход, если $\neq 0$

**9.2.2.** Выполните программу:

СТ800 80СВП

**9.2.3.** Проверьте результаты выполнения программы в соответствии с таблицей:

C	H	M(900H) - M(90F)
00	910	00

**9.2.4.** Запишите в память, начиная с адреса 800H, программу замены данных в массиве памяти (900H - 90FH), отличных от 00H, на 00H.

Адрес	Команда	Машинный код	Комментарий
800	MVI C, 10H	0E 10	; загрузка C=10H (длина массива)
802	LXI H, 900H	21 00 09	; загрузка HL=900H (начальный адрес массива)
805	MOV A, M	7E	; чтение $A \leftarrow M$
806	CPI 0	FE 00	; проверка условия A=0
808	JZ 80DH	CA 0D 08	; переход, если A=0
80B	MVI M, 0	36 00	; если $A \neq 0$ , загрузка M=0
80C	INX H	23	; HL = HL+1 (следующий адрес)
80E	DCR C	0D	; C=C-1 (длина массива -1)
80F	JNZ 805H	C2 05 08	; продолжать, если длина массива $\neq 0$

9.2.5. Выполните программу:

СТ800 812ВП

9.2.6. Проверьте результаты выполнения программы в соответствии с таблицей:

C	H	M(900H) - M(90FH)
00	910	00

### Задание на самостоятельную работу к разделу 9.2.:

1. Напишите и выполните программу заполнения массива памяти (900H-9FFH) следующим образом: в чётные ячейки (900H, 902H, 904H и т.д.) записать 55H, в нечётные (901H, 903H, 905H и т.д.) записать ААН.
2. Напишите и выполните программу подсчёта контрольной суммы массива памяти (900H-9FFH) по модулю 256, т.е. без учёта переполнения байта.

### 9.3. Команды перехода по признаку C - ПЕРЕНОС

JC ADR	переход, если C=1
JNC ADR	переход, если C=0

9.3.1. Запишите в память, начиная с адреса 800H, программу подсчёта нулей в байте. Исходное значение задано в регистре C.

Адрес	Команда	Машин. код	Комментарий
800	MOV A, C	79	
801	MVI B, 8	06 08	; загрузка B=8 (количество разрядов в байте)
803	MVI E, 0	1E 00	; загр. E=0 (исходное значение кол.-ва нулей)
805	RAR	1F	; значение мл.бита в бит переноса и сдвиг вправо
806	JC 80AH	DA 0A 08	; если перенос =1, обойти инкрем. Счётчика нулей
809	INR E	1C	; E=E+1 (инкремент счётчика нулей)
Адрес	Команда	Машин. код	Комментарий
80A	DCR B	05	; B=B-1 (следующий разряд)
80B	JNZ 805H	C2 05 08	; переход на проверку следующего разряда байта

**9.3.2.** Выполните программу, предварительно задавая исходные значения в соответствии с таблицей. Проверьте результаты.

СТ800 80ЕВП

С	00	DB	FF	03	9A	55
Е	00	06	08	02	04	04

**9.3.3.** Запишите в память, начиная с адреса 800H, программу, осуществляющую сравнение двух 16-разрядных величин на больше или равно по следующему правилу:

HL >= DE ?

Если HL < DE C=1,

иначе C=0.

Адрес	Команда	Машинный код	Комментарий
800	MVI C, 0	0E 00	; загрузка C = 0, для случая HL >=DE
802	MOV A, L	7D	; сравнение младших байтов
803	SUB E	93	; если L < E, перенос установки в 1
804	MOV A, H	7C	
805	SBB D	9A	; сравнение старших байтов с учётом переноса
806	JNC 80BH	D2 0B 08	; если HL >=DE, перенос = 0, C=0
809	MVI C, 1	0E 01	; если HL < DE, перенос =1, C=1

**9.3.4.** Выполните программу, предварительно задавая исходные значения в соответствии с таблицей. Проверьте результаты.

СТ800 80ВВП

Н	0000	FFFF	8002	5A7F	3A55	0000
DE	0001	FFFF	7FFF	6080	C201	FFFF
С	01	00	00	01	01	01

**Задание на самостоятельную работу к разделу 9.3.:**

1. Напишите и выполните программу подсчёта количества единиц и нулей в байтах массива памяти, начальный и конечный адреса которого задаются, соответственно, в парах регистров HL и DE.



Заполните таблицу:

HL (нач. адр.)	0000	0800	0000
DE (кон. Адр.)	03FF	0AFF	FFFF
кол. единиц			
кол. нулей			

#### 9.4. Команды перехода по признаку P - ЧЁТНОСТЬ.

JPE ADR	переход, если P=1;
JPO ADR	переход, если P=0.

**9.4.1.** Запишите в память, начиная с адреса 800H, программу дополнения байта до чётности в старшем разряде. Исходное число находится в регистре C.

Адрес	Команда	Машинный код	Комментарий
800	MOV A, C	79	; пересылка $A \leftarrow 0$ (исходный байт)
801	ANI 7FH	E67F	; обнуление старшего разряда
803	ORA A	B7	; $A \text{ OR } A$ - установка бита чётности
804	JPO 809H	E2 09 08	; переход, если исходный байт чётный
807	ORI 80H	F6 80H	; дополнить до чётности
809	MOV C, A	4F	; результат

**9.4.2.** Выполните программу, задавая исходные значения в соответствии с таблицей. Проверьте результат.

C (исходное значение)	00	FF	B6	80	CD	75
C (результат)	00	FF	36	00	4D	F5

#### Задание на самостоятельную работу к разделу 9.4.:

1. Напишите и выполните программу дополнения байта до нечётности и заполните таблицу:

C (исходное значение)	00	FF	B6	80	CD	75
-----------------------	----	----	----	----	----	----

С (результат)						
---------------	--	--	--	--	--	--

2. Напишите и выполните программу подсчёта в массиве памяти количества чётных и нечётных байт. Начальный и конечный адреса задаются парами регистров HL и DE соответственно. Заполните таблицу:

HL	0000	0800	0C00
DE	03FF	0AFF	FFFF
Кол-во чётных			
Кол-во нечётных			

### 9.5. Команды перехода по признаку S - ЗНАК

JM ADR	переход, если S = 1 (число <0)
JP ADR	переход, если S = 0 (число >=0)

**9.5.1.** Запишите в память, начиная с адреса 800H, программу выделения из массива (100H - 10FH) цифровых символов с кодами 30H...39H и записи их в другой массив (начальный адрес 900H).

Адрес	Команда	Машинный код	Комментарий
800	LXI H, 100H	21 00 01	; загрузка HL - нач. адрес исх. массива
803	MVI B, 0FH	06 0F	; загрузка B = 0FH (длина массива)
Адрес	Команда	Машинный код	Комментарий
805	LXI D, 900H	11 00 09	; загрузка DE (нач. адрес выходного массива)
808	MOV A, M	7E	; чтение A = M (эл-т массива). Проверить ; условие 30H <= A <= 39H
809	SBI 30H	DE 30	; A = A-30H, если A<30H, S=1
80B	JM 805H	FA 05 08	; переход к следующему элементу
80D	SBI 0AH	DE 0A	; A>=30H, A = A - 0A, если A>=0, S=0

80F	JP 805H	F2 05 08	; переход к след. элементу
812	MOV A, M	7E	; 30H <= A <= 39H, перепис. в массив вывода
813	STAX	12	
814	INX D	13	; адрес след. элемента массива вывода
815	INX H	23	; адрес след. элемента исходного массива
816	DCR B	05	; длина массива - 1 = 0 ?
817	JNZ 808H	C2 08 08	; если нет, продолжать.

**9.5.2.** Выполните программу. Проверьте правильность работы программы, просмотрев выходной массив (900H - 90FH). Значения элементов массива должны соответствовать условию:

30H <= элемент массива <= 30H

### **Задание на самостоятельную работу к разделу 9.5.:**

1. Напишите и выполните программу, формирующую из одного массива два в соответствии со следующими требованиями:

	Нач. адрес	Конеч. Адрес	Элемент массива
Исх. Массив	0	7FH	
1 массив	900H		00H <= Эл.м.<=7FH
2 массив	980H		80H <= Эл.м.<=FFH

и подсчитайте, соответственно, количество элементов в каждом массиве. Проверьте правильность работы программы, просмотрев соответствующее количество ячеек обоих массивов.

### **Контрольные вопросы.**

1. Перечислите признаки, используемые в командах условных переходов.
2. Что означает выражение “сброшенный бит признака” ?
3. Чему будет равен счётчик команд после выполнения операции JMP 920H ?
4. Каков будет результат работы программы из пункта 9.2.1., если команду JNZ заменить на JZ ?
5. Какого знака должен быть регистр A, чтобы осуществился переход по команде JP 930H ?

## Лабораторная работа №10

### Изучение команд манипуляции стеком. Вызов подпрограммы и возврат из неё.

**Цель работы:** освоение основных команд работы со стеком (запись в стек, восстановление из стека, обмен со стеком). Изучение использования стека при операциях с подпрограммами (вызов подпрограммы, возврат из подпрограммы).

#### Теоретические основы.

Стек - это специальная область ОЗУ, используемая для сохранения и восстановления данных, а также адресов возврата при вызове подпрограммы. Нижняя граница области стека определяется 16-разрядным регистром-указателем стека УС. В микропроцессорной системе предусмотрены 3 типа операций со стеком:

- I. запись в стек;
- II. восстановление из стека;
- III. обмен со стеком.

I. **Запись в стек** осуществляется при сохранении содержимого пар регистров (BC, DE, HL, PSWA), а также при вызове подпрограммы (при этом в стек помещается адрес возврата). Запись в стек производится следующим образом:

- a) из указателя стека вычитается 1;
- b) по адресу, содержащемуся в указателе стека, записывается старший байт (старший регистр или старший байт адреса);
- c) из указателя стека вычитается 1;
- d) записывается младший байт.

Например, при сохранении содержимого пар регистров BC указатель стека и сам стек будут выглядеть:

		значение указателя стека	область стека
Значение после записи →		УС - 2	С
		УС - 1	В

Исходное значение → УС 

--

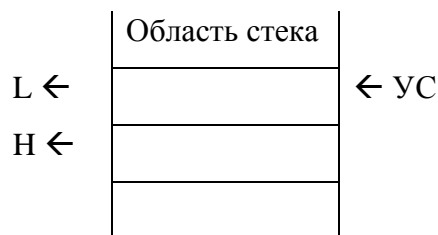
II. **Восстановление из стека** содержимого пар регистров и возврат из подпрограммы осуществляются в обратном порядке:

- a) по адресу, содержащемуся в указателе стека, считывается младший байт;
- b) к указателю стека прибавляется 1;
- c) считывается содержимое старшего байта;
- d) к указателю стека прибавляется 1.

При восстановлении из стека содержимого пары регистров ВС значения указателя стека и область стека будут иметь вид:

		значения УС	область стека
Исходное значение	→	УС	С
		УС + 1	В
Значение после восстановления	→	УС + 2	

III. При **обмене со стеком** осуществляется обмен содержимого верхнего элемента стека и пары регистров HL. Значение указателя стека при этом не изменяется.



Восстановление содержимого пар регистров из стека должно производиться в обратном порядке записи. Так, например, если в стек была произведена запись в последовательности ВС - DE - HL, то восстановление должно производиться в обратном порядке: HL - DE - ВС. Таким образом, стек работает по принципу “первым пришёл, последним ушёл”.

При **вызове подпрограммы** в стек записывается адрес команды, следующей за командой вызова подпрограммы (аналогично операции записи в

стек). Затем, при возврате из подпрограммы, из стека в счётчик команд РС записывается адрес возврата. Необходимо следить за тем, чтобы непосредственно перед выполнением команды возврата из подпрограммы верхним элементом стека являлось значение адреса возврата, то есть, чтобы между командами вызова подпрограммы и возврата из неё не было загрузки стека без восстановления.

## Содержание работы.

**10.1.** Команды записи в стек, восстановления и обмена стеком.

**10.2.** Команды вызова подпрограмм и возврата из них.

## Порядок выполнения работы.

### **10.1. Команды записи в стек, восстановления и обмена со стеком.**

В таблице 10.1. приведены команды работы со стеком - записи, восстановления, обмена.

PUSH B	запись в стек регистровой пары BC
PUSH D	DE
PUSH H	HL
PUSH PSW	F, A;
POP B	восстановление регистр. пары BC
POP D	DE
POP H	HL
POP PSW	F, A;
XTHL	обмен содержимого верхнего элемента стека и HL

Таблица 10.1.

**10.1.1.** Запишите в память, начиная с адреса 800H, коды программы сохранения содержимого регистров BC, DE, HL, F, A:

Адрес	Команда	Машинный код	Комментарий
800	LXI SP, 920H	31 20 0B	; загрузка SP = 920H (область стека)
803	PUSH B	C5	; запись BC в стеке
804	PUSH D	D5	; запись DE в стеке
805	PUSH H	E5	; запись HL в стеке
806	PUSH PSW	F5	; запись F, A

**10.1.2.** Выполните программу и заполните таблицу:

СТ800 807ВП

BC	DE	HL	PSW A	УС

**10.1.3.** Измените содержимое регистров микропроцессора B, C, D, E, H, L, A, F.

**10.1.4.** Запишите программу восстановления регистров:

Адрес	Команда	Машинный код	Комментарий
800	POP PSW	F1	; восстановление F и A
801	POP H	E1	; восстановление регистр. пары HL
802	POP D	D1	; DE
803	POP B	C1	; BC

**10.1.5.** Выполните программу и заполните таблицу:

СТ800 804ВП

BC	DE	HL	F, A	УС

**Примечание:** значения регистров B, C, D, E, H, L, A, F должны совпадать с соответствующими значениями таблицы из пункта 10.1.2. Значение указателя стека УС = 0920H.

**10.1.6.** Запишите в память программу пересылки данных из одного массива в другой, используя для адресации регистровую пару HL и команду обмена стеком:

Адрес	Команда	Машинный код	Комментарий
800	LXI H, 900H	21 00 09	; загрузка HL = 900H (адрес 2 массива)
803	LXI SP, 0B00H	31 00 0B	; загр. SP = B00H (нижняя граница стека)
806	PUSH H	E5	; запись в стек HL (адрес 2 массива)
807	LXI H, 0	21 00 00	; загрузка HL = 0 (адрес 1 массива)
80A	MVI C, 80H	0E 80	; загрузка C= 80H (длина массива)
80C	MOV A, M	7E	; пересылка $A \leftarrow M$ (элемент 1 массива)
80D	INX H	23	; HL = HL+1 (увеличить адрес 1 массива)
80E	XTHL	E3	; обмен стека и HL. HL - адрес 2 массива, ; стек - адрес 1 массива
80F	MOV M, A	77	; запись $M \leftarrow A$ (элемент - во 2 массив)
810	INX H	23	; HL = HL+1 (увеличить адрес 2 массива)
811	XTHL	E3	; обмен стека и HL. HL - адрес 1 массива, ; стек - адрес 2 массива
812	DCR C	0D	; C = C-1 (уменьшить длину массива)
813	JNZ 80CH	C2 0C 08	; продолжить, если длина отлична от нуля
816	POP D	1	; очистка стека

**10.1.7.** Выполните программу. Подсчитайте контрольную сумму (КС) исходного массива (0 - 7FH) и КС второго массива (900H - 97FH), они должны совпадать. Заполните таблицу:

СТ800 817ВП

HL адрес 1 массива	DE адрес 2 массива	C длина	КС 1 массива	КС 2 массива
0000	0900	80		



## Задание на самостоятельную работу к разделу 10.1.

1. Напишите и выполните программу обмена регистровыми парами, используя команды записи в стек и восстановления из стека, следующим образом:

BC → DE

DE → HL

HL → BC

Проверьте результаты.

2. Напишите и выполните программу побайтного сложения двух массивов, без учёта переноса, и записи результатов в первый массив, используя для адресации обоих массивов регистровую пару HL.

Адреса массивов и значение указателя стека следующие:

1 массив	0 - FFH
2 массив	100H - 1FFH
УС	B20H

## 10.2. Команды вызова подпрограммы и возврата из неё

Команды работы с подпрограммами приведены в таблице 10.2.

CALL ADR	вызов подпрограммы по адресу ADR, указанному во втором и третьем байтах команды, причём при записи кодов команды вызова в памяти сначала записывается младший байт адреса, а затем старший, по аналогии с записью кодов команд загрузки регистровых пар;
RET	возврат из подпрограммы.

Таблица 10.2.

**10.2.1.** Запишите в память программу, использующую в своём теле вызовы подпрограммы, осуществляющей сравнение HL и DE.

Адрес	Команда	Машинный код	Комментарий
800	LXI H, 900H	21 00 09	; загрузка HL (начальный адрес массива)
803	LXI D, 97FH	11 7F 09	; загрузка DE (конечный адрес массива)
806	LXI SP, 0B00H	31 00 0B	; загр.УС = B00H (нижняя граница стека)
809	MVI B, 0	06 00	; загрузка B = 0 (исходное значение КС)
80B	MOV A, B	78	; подсчёт КС массива памяти
80C	ADD M	86	
80D	MOV B, A	47	
80E	INX H	23	; HL = HL+1 (адрес следующей ячейки)
80F	CALL 816H	CD 16 08	; вызов подпрограммы
812	JNC 80BH	2 0B 08	
815	NOP		; пустая команда
816	MOV A, E	7B	; подпрограмма сравнения:
817	SUB L	95	; если HL > DE то бит переноса = 1,
818	MOV A, D	7A	; иначе бит переноса = 0
819	SBB H	9C	
81A	RET	C9	; возврат из п/п

**10.2.2.** Выполните программу поэтапно, устанавливая точки останова (ТО) в соответствии с приведённой ниже таблицей. Запишите в таблицу значения указателя стека в точках останова.

СТ800 ТОВП

	ТО1 = 80FH	ТО2 = 816H	ТО3 = 81AH	ТО4 = 812H	ТО5 = 815H
УС					

В точке останова ТО3 посмотрите содержимое ячеек с адресами УС, УС+1. В них должны находиться младший и старший байты адреса:

М (УС)	12
М (УС+1)	08

**10.2.3.** Запишите в память программу, содержащую вызов подпрограммы с передачей параметров через стек:

Адрес	Команда	Машинный код	Комментарий
800	LXI SP, 0B00H	31 00 0B	; загрузка UC = B00H
803	LXI H, 5678H	21 78 56	; загрузка HL = 5678H
803	LXI B, 0123H	01 23 01	; загрузка BC = 0123H
809	LXI D, 9ABCH	11 BC 9A	; загрузка DE = 9ABCH
80C	PUSH H	E5	; запись значений регистров в стек
80E	PUSH D	D5	
80F	CALL 813H	CD 13 08	; вызов подпрограммы (п/п)
812	NOP		
813	POP H	E1	; загрузка HL (адрес возврата из п/п)
814	POP D	D1	; восстановление DE
815	POP B	C1	; восстановление BC
816	XTHL	E3	; обмен стека и HL (в стеке адрес ; возврата, в HL - восстановл. значение)
817	RET	C9	; возврат из п/п

**10.2.4.** Выполните программу и проверьте значения регистров в соответствии с таблицей:

СТ800 812ВП

BC	DE	HL	UC
0123	9ABC	5678	0A00

**Задание на самостоятельную работу к разделу 10.2.**

1. Напишите и выполните программу, содержащую вызов подпрограммы, выполняющей сложение четырёх двухбайтовых величин. Исходные величины передаются в подпрограмму через стек, возврат результата в регистровой паре HL. Исходные значения двухбайтовых величин возьмите произвольно.
2. Напишите и выполните программу в соответствии со следующим алгоритмом:
  - а) заполнить массив (900H - 9FFH) константой EEH;

- b) подсчитать контрольную сумму этого массива по модулю 256 (без учёта переноса);
- c) заполнить массив (A00H - AFFH) константой 11H;
- d) подсчитать контрольную сумму.

Подсчёт контрольной суммы оформите как подпрограмму, с передачей в качестве входных параметров адреса начала и конца массива, а в качестве выходного параметра - значение контрольной суммы массива.

### **Контрольные вопросы.**

1. Что из себя представляет и для чего используется указатель стека ?
2. Как указатель стека УС обозначается в программах, работающих со стеком ?
3. Пусть начальное значение УС - 1000H. Чему будет равен указатель стека после записи в стек трёх 16-разрядных величин ? Обоснуйте ответ.
4. Пусть в стеке содержится адрес команды, следующей за командой вызова подпрограммы. Значение УС при этом равно 1000H. Чему будет равен указатель стека после возврата из подпрограммы ?
5. Перечислите основные команды работы со стеком.
6. Опишите механизм использования стека при работе с подпрограммами.
7. Для чего в программе 10.2.3. после команды вызова подпрограммы стоит команда, не выполняющая никаких действий ?

## Лабораторная работа №11

### Изучение команд ввода-вывода

**Цель работы:** освоить принципы взаимодействия микропроцессора с внешними устройствами, основанные на командах ввода и вывода данных. В качестве примера изучить работу программы-драйвера, осуществляющей приём данных с клавиатуры и вывод их на дисплей.

#### Теоретические основы.

Для организации взаимодействия микропроцессора с внешними устройствами предусмотрены две команды: команда ввода и команда вывода 8-разрядных данных.

По **команде ввода** осуществляется считывание данных из внешнего порта, адрес которого задан во втором байте команды, в регистр-аккумулятор микропроцессора. Например:

```
IN PORT1,
```

где PORT1 - адрес порта ввода.

По **команде вывода** осуществляется вывод данных из регистра-аккумулятора микропроцессора во внешний порт, адрес которого указывается во втором байте команды. Например:

```
OUT PORT2,
```

где PORT2 - адрес порта вывода.

Используя эти команды, пользователь имеет возможность создавать программы-драйверы, обеспечивающие взаимодействие микропроцессора с периферийной техникой.

#### Содержание работы.

11.1. Программа-драйвер, осуществляющая приём данных с клавиатуры УМК и вывода их на шестисегментный дисплей.

#### Выполнение работы

### 11.1. Программа-драйвер.

Исходные данные приведены в следующей таблице:

PORT A	порт номера индикатора содержит: 20H - 0 индикатор (самый правый) 10H- 1 индикатор 08H- 2 индикатор 04H- 3 индикатор 02H- 4 индикатор 01H- 5 индикатор Адрес: PORTA = F8H
PORT B	порт вывода данных - выводится код отображаемого символа. Адрес: PORTB = F9H
PORT C	порт состояния клавиатуры. Адрес: PORTC = FAH

Для отображения символов на дисплее необходимо сначала вывести в PORTA номер соответствующего индикатора, а затем в PORTB вывести код символа. Для того, чтобы отображаемые символы постоянно присутствовали на индикаторах дисплея, его надо постоянно регенерировать посредством вывода соответствующих данных в PORTA и PORTB. Для этой цели используется буфер регенерации, в котором хранятся коды отображаемых символов. Для того, чтобы погасить индикатор, в PORTA надо вывести номер этого индикатора, а в PORTB - 0. При нажатии на клавишу в порту состояния устанавливаются соответствующие биты, говорящие о том, что есть данные для ввода.

**11.1.1.** Запишите в память коды программы ввода данных с клавиатуры и отображения их на экране дисплея:

Адрес	Команда	Машинный код	Комментарий
800	LXI H, 8AЕH	21 AE 08	; обнулить буфер регенерации
803	MVI C, 6	0E 06	
805	MVI M, 0	36	
807	INX H	23	
808	DCR C	0D	
809	JNZ 805	C2 05 08	
80C	LXI H, 08AЕH	21 AE 08	;загр. HL (адр. буфера регенерации дисплея)
80F	MVI B, 20H	06 20	;загрузка B (номер мл. индикатора)
811	MOV A, B	78	
812	OUT PORTA	D3 F8	; вывод номера индикатора
814	MOV A, M	7E	; чтение A = M (код выводимого символа)
815	OUT PORTB	D3 F9	; вывод символа на дисплей
817	IN PORTÑ	DB F9	; поймать момент
819	ANI 74H	E6 74	; нажатия клавиши
81B	CPI 74H	FE 74	
81D	MVI A, 0	3E 00	; сбросить выведенный
81F	OUT PORT B	D3 F9	; символ
821	JNZ 82EH	C2 2E 08	; переход, если клавиша нажата
824	INX H	23	;HL=HL+1 (адрес след. выводимого символа)
825	MOV A, B	78	; вычисление номера след. индикатора
826	RRC	0F	
827	MOV B, A	47	
828	JNC 811H	D2 11 08	; цикл регенерации экрана и проверки ввода символа, след. индикатор
82B	JMP 80CH	C3 0C 08	; перейти к след. индикатору
82E	CALL 865H	CD 65 08	; вызов подпрограммы задержки (борьба с дребезгом)
831	IN PORTC	DB FA	; чтение введённых данных
833	MOV C, A	4F	
834	IN PORTC	DB FA	; поймать момент

Адрес	Команда	Машинный код	Комментарий
836	ANI 74H	E6 74	; отпущения клавиши
838	CPI 74H	FE 74	
83A	JNZ 834H	C2 34 08	
83D	CALL 865H	CD 65 08	; вызов подпрограммы задержки
840	CALL 86FH	CD 6F 08	; вызов п/п преобразования кода клавиши в код ASCII. Результат возвращается в A
843	LXI H, 89EH	21 9E 08	; преобразование кода ASCII принятого символа в код вывода этого символа
846	CPI A	FE 41	; HL - адрес таблицы кодов символов
848	JC 84DH	DA 4D 08	; A - код ASCII символа - индекс в таблице кодов символов
84B	SUI 7	D6 07	
84D	ANI 0FH	E6 0F	
84F	MOV E, A	5F	
850	MVI D, 0	16 00	
852	DAD D	19	
853	MOV E, M	5E	; E - код выводимого символа
854	LXI H, 8AЕH	21 AE 08	; записать в буфер вывода
857	MVI D, 6	16 06 08	; сдвиг влево буфера вывода
859	MOV A, M	7E	
85A	MOV M, E	73	
85B	MOV E, A	5F	
85C	INX H	23	
85D	DCR D	15	
85E	JNZ 859H	C2 59 08	
861	JMP 80CH	C3 0C 08	; продолжать
864	NOP	00	
865	LXI D, 1000H	11 00 10	; подпрограмма задержки на 10 мс
868	DCX D	1B	; DE = DE - 1
869	MOV A, E	7B	



Адрес	Команда	Машинный код	Комментарий
86A	ORA D	B2	
86B	JNZ 868H	C2 68 08	
86E	RET	C9	
86F	MOV A, C	79	; преобразование
870	ANI 10H	E6 10	;кода принятого
872	JNZ 877H	C2 77 08	;символа с информационной
875	MVI C, 0	0E 00	;клавиатуры в код ASCII,
877	MOV A, C9EH	79	;а коды функциональных
878	ANI 64H	E6 64	;клавиш, соответственно,
87A	RAR	1F	;в коды 0 -7
87B	RAR	1F	
87C	RAR	1F	
87D	MOV C, A	4F	
87E	MOV A, B	78	
87F	ANI 3	E6 03	
881	JNZ 897H	C2 97 08	
884	MOV A, B	78	
885	RAR	1F	
886	RAR	1F	
887	RAR	1F	
888	CPI 4	FE 04	
88A	JNZ 88EH	C2 8E 08	
88D	DCR A	3D	
88E	ADD C	81	
88F	ORI 0	F6 30	
891	CPI 3AH	FE 3A	
893	RC	D8	
894	ADI 7	C6 07	
896	RET	C9	
897	DCR A	3D	

Адрес	Команда	Машинный код	Комментарий
898	MOV B, A	47	
899	MOV A, C	79	
89A	RRC	0F	
89B	ADD B	80	
89C	RET	C9	
89D	NOP	00	
89E	DB 3FH, 6H, 5BH, 4FH		; таблица кодов вывода символов
8A2	DB 66H, 6DH, 7DH, 7H		
8A6	DB 7FH, 6FH, 77H, 7CH		
8AA	DB 39H, 5EH, 79H, 71H		
8AE			; буфер регенерации

**11.1.2.** Подсчитайте контрольную сумму кодов введённой программы (800H - 80DH). Она должна быть равна 44H.

**11.1.3.** Запустите программу:

СТ800ВП

При нажатии на клавишу клавиатуры данных (правая часть кнопок), на самом правом индикаторе дисплея будет отображаться введённый символ. Ранее введённый символ переместится на следующий индикатор (влево).

**11.1.4.** Для выхода из программы нажмите кнопку ПРЕРЫВАНИЕ.

### **Контрольные вопросы.**

1. Для чего служит регистр-аккумулятор процессора ?
2. С какими внешними устройствами может взаимодействовать микропроцессор?
3. Какие порты и с какой целью используются в программе ?
4. Для чего предназначен буфер регенерации ?