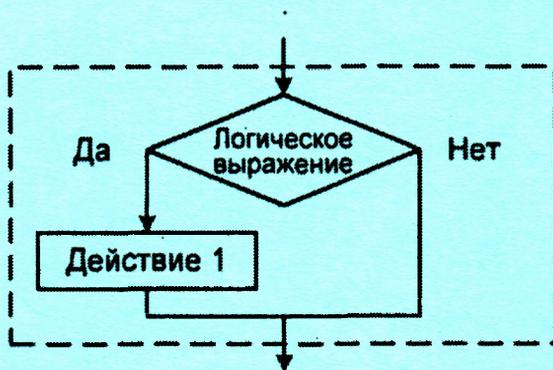


438-2021

ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ АССЕМБЛЕРА ДЛЯ ВОСЬМИРАЗЯДНОГО МИКРОПРОЦЕССОРА

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

к выполнению лабораторных работ по дисциплине
«Цифровые устройства и микропроцессоры»
для студентов направления подготовки 11.03.01 «Радиотехника»
(профиль «Радиотехнические средства передачи, приема и обработки
сигналов») всех форм обучения



Воронеж 2021

УДК 681.325.5(07)
ББК 32.973.2я7

Составители: канд. физ.-мат. наук, доц. В. А. Кондусов,
канд. техн. наук, доц. Е. Д. Алперин

Программирование на языке Ассемблера для восьмиразрядного микропроцессора: методические указания к выполнению лабораторных работ по дисциплине «Цифровые устройства и микропроцессоры» для студентов направления 11.03.01 «Радиотехника» (профиль «Радиотехнические средства передачи, приема и обработки сигналов») всех форм обучения / ФГБОУ ВО «Воронежский государственный технический университет»; сост.: В. А. Кондусов, Е. Д. Алперин. – Воронеж: Изд-во ВГТУ, 2021. – 29 с.

В методических указаниях изложены требования и рекомендации по выполнению лабораторных работ, их объему и содержанию.

Предназначены для студентов направления 11.03.01 «Радиотехника» (профиль «Радиотехнические средства передачи, приема и обработки сигналов») всех форм обучения.

Ил. 5. Табл. 2. Библиогр.: 2 назв.

УДК 681.325.5(07)
ББК 32.973.2я7

Рецензент – А. В. Останков, д-р техн. наук, проф.,
зав. кафедрой радиотехники ВГТУ

*Печатается по решению редакционно-издательского совета
Воронежского государственного технического университета*

ВВЕДЕНИЕ

Цель лабораторного практикума: закрепление теоретических знаний по курсу цифровых устройств и микропроцессоров, приобретение практических навыков по программированию микропроцессоров (МП) и микропроцессорных систем (МПС) в машинных кодах и на языке ассемблера для микроЭВМ на базе микропроцессора КР580.

Тематика лабораторных работ подобрана таким образом, чтобы акцентировать внимание студентов на функциях и характеристиках основных программно-доступных узлов МП и МПС, а также на особенностях команд, микрокоманд и микроопераций.

Знания и умения, полученные в лабораторном практикуме, позволят, в случае необходимости, освоить задачи анализа и проектирования МПС на базе других микропроцессорных комплектов.

Для подготовки к лабораторным работам следует пользоваться приведенными ниже источниками. К выполнению каждой работы допускаются студенты, выдержавшие собеседование с преподавателем по теоретическим и практическим вопросам, относящимся к тематике работы.

ЛАБОРАТОРНАЯ РАБОТА № 1 ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ АССЕМБЛЕРА ДЛЯ ВОСЬМИРАЗЯДНОГО МИКРОПРОЦЕССОРА С ИСПОЛЬЗОВАНИЕМ ЭМУЛЯТОРОВ ПАРТНЁР 01.01 и КР580

Цель работы: ознакомление с работой микропроцессора КР580ВМ80А (i8080), со структурой МП, назначением выводов МП, со структурой памяти МП-системы, с форматами команд МП. Изучение принципов работы микроЭВМ с использованием эмуляторов Партнер 01.01 и КР580.

Основные технические характеристики эмулируемой микро-ЭВМ «Партнер 01.01»

Центральный процессор – КР580ВМ80А; объем памяти – 80 Кбайт; в том числе ОЗУ – 64 Кбайт; ПЗУ – 16 Кбайт; быстродействие выполнения операций типа «регистр-регистр» – 500 тыс. операций в секунду. Программное обеспечение (ПО) включает системные наборы программ: Монитор, Ассемблер, Редактор и Бейсик.

Общие сведения об эмуляторах

Эмулятор «Партнер 01.01» обеспечивает выполнение на IBM – совместимых компьютерах программ, разработанных для 8-разрядных микро-ЭВМ типа

«Партнер 01.01», «Радио 86РК», «Апогей», «Микроша» и др.

Эмулятор КР580 микропроцессорной системы на базе микропроцессора КР580ВМ80А (i8080) позволяет: написание программ на языке ассемблера, используя систему команд МП КР580ВМ80А, их отладку и выполнение в тактовом, командном и сквозном режимах; изучить принципы и порядок выполнения команд.

Желательные системные требования: процессор – не ниже РПД 450МГц; ОЗУ – не менее 128МБ; ОС MS Windows семейства NT.

Назначение системных программ эмулируемой микро-ЭВМ «Партнер 0101

Программа МОНИТОР выполняет функции минимальной операционной системы (ОС) и обеспечивает управление устройствами машины (контроллерами клавиатуры, дисплея), предоставляет пользователю возможность загрузки программ из ПЗУ, запуска их, а также необходимые средства для отладки программ, написанных на языках Ассемблера или Бейсик. Данная версия программы Монитор в основном совместима с аналогичной программой микро-ЭВМ типа «Радио 86 РК» (Микроша, Криста), что позволяет использовать ПО этих машин.

РЕДАКТОР предназначен для редактирования любого текста непосредственно на экране дисплея: исправление ошибок, удаление и ввод новых символов и фрагментов текста.

АССЕМБЛЕР – программа-транслятор для разработки программ на языке Ассемблера микропроцессора КР580ВМ80А. Текст программы с клавиатуры с помощью РЕДАКТОРА заносят в область текстового буфера ОЗУ, а затем с помощью АССЕМБЛЕРА транслируют. Результат трансляции – машинные коды располагаются в другой области ОЗУ, называемой областью трансляции.

Директивы МОНИТОРА

С помощью директив Монитора пользователь может управлять работой машины, просматривать и изменять содержимое ячеек памяти и регистров процессора, отлаживать программу, загружать программы из ПЗУ.

Общая форма записи директив:

<Символ>[параметр1],[параметр2],[параметр3]<Enter>

Имена всех директив состоят из одной латинской буквы, непосредственно за которой могут следовать не более трех параметров, представляющие собой шестнадцатеричные числа. Параметры отделяются между собой запятой. Запятую ставят в тех случаях, когда один из параметров (в том числе и первый) отсутствует. Директивы заканчиваются нажатием клавиши < Enter >. При вводе директив возможны следующие ошибки: длинная строка, неправильное имя директивы, неправильный формат числа. Если эти ошибки произошли, то машина отвечает на них звуковым сигналом, и на экране появляется знак вопроса.

После чего необходимо повторить ввод директивы.

Директива «D»

По директиве «D» на дисплей выводится содержимое заданной области памяти в шестнадцатеричном коде (H-коде)

Форма директивы

<D> [начальный адрес], [конечный адрес]

Один или оба параметра могут быть опущены. В этом случае они принимаются равными нулю.

Пример: D100,200

Эта директива выводит на дисплей содержимое области памяти от 100H до 200H в шестнадцатеричном коде.

Директива «L»

По директиве «L» на дисплей выводится содержимое области памяти в символьном виде. Не отображаемые символы отображаются в виде точек.

Форма директивы:

<L> [начальный адрес], [конечный адрес]

Один или оба параметра могут быть опущены. В этом случае они равны нулю.

Пример: L200,300

Директива «F»

По директиве «F» содержимое области памяти, определённое первыми двумя параметрами, заполняется значением 3-го параметра.

Форма директивы:

<F>[начальный адрес],[конечный адрес],[код]

Один или все параметра могут быть опущены. В этом случае они равны нулю

Пример: F100, 200,45

В ячейках памяти от 100 до 200 записан код 45.

Директива «M»

По директиве «M» на дисплей выводятся адрес и содержимое ячейки памяти в шестнадцатеричном коде. Можно осуществить просмотр и (или) изменение содержимого ячеек. Форма директивы:

<M> [адрес]

Нажатием клавиши < Enter > отображается содержимое следующей ячейки. Для изменения содержимого ячейки следует (до нажатия клавиши < Enter >) набрать новое значение, а затем нажать < Enter >. Просмотр и изменение заканчивается после ввода символа «.».

Пример: M8AC

08AC 6A 7A В ячейке 08AC произведено изменение

08AD 77

08AE 44 Последняя просмотренная ячейка 08AE

Директива «X»

По директиве X на дисплей выводится содержимое регистров процессора в шестнадцатеричном коде.

Синтаксис директивы: <X>

Первым на дисплей выводится содержимое регистра PC. Просмотр регистровых пар HL, DE, BC, SP, AF производится по нажатию клавиши Enter.

Пример: X

PC 013A При нажатии клавиши Enter
HL поочередно будет выводиться
DE содержимое регистров. Это
BC содержимое можно изменить,
SP записав справа новые значения.
AF

Директива «G»

По директиве G управление передается программе по заданному адресу (происходит запуск и исполнение программы в машинных кодах).

Форма директивы:

<G> [адрес запуска], [адрес останова]

По достижении адреса, заданного вторым параметром, управление получает МОНИТОР и на дисплей выводится этот адрес. Один или оба параметра могут быть опущены. В этом случае первый параметр принимается равным нулю.

Пример: G -запуск с адреса 0

G100 - запуск с адреса 100

G100,103A -выполнение программы с адреса 100 по адрес 103A

Адрес останова не обязателен, если в конце исполняемой программы использовать команду перехода с указанием старта МОНИТОРА: 0F800H (2048D), например: JMP 0F800H; конец программы, выход в МОНИТОР.

Домашнее задание

Задание первое. Изучить организацию функционирования микропроцессорной системы на базе восьмиразрядного микропроцессора [1, С. 84-88, С.95-107, С. 238-242], [2, С.302-318, С.362-391].

На рис. 1 приведена структурная схема МП КР580ВМ80А.

В структуре 8-разрядного микропроцессора можно выделить 3 основных узла: операционный, внутренней памяти и управления. **Операционный узел** включает: 8-разрядное арифметико-логическое устройство АЛУ, обеспечивает выполнение операций обработки данных. Данные могут быть представлены либо двоичным, либо двоично-десятичным кодом; регистр признаков F, фиксирующий признаки, вырабатываемые АЛУ в процессе выполнения команды; ак-

кумулятор (А); регистр временного хранения (Буф. А); регистр временного хранения операндов; десятичный корректор.

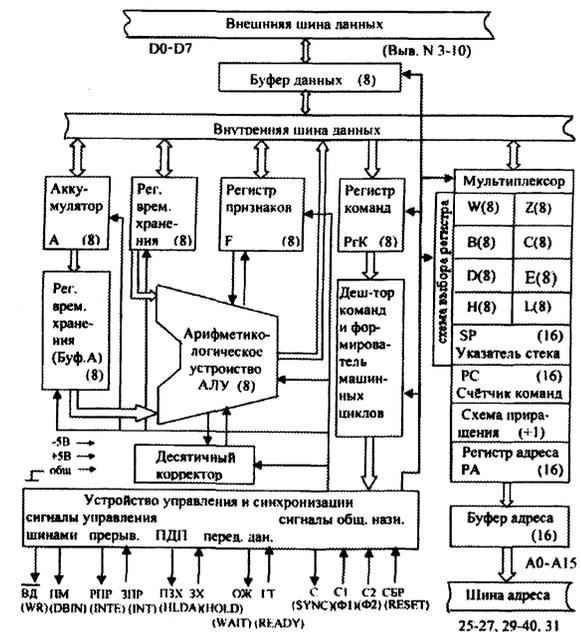


Рис. 1. Архитектура микропроцессора КР580ВМ80А

Узел внутренней памяти включает: шесть 8-разрядных регистров общего назначения В, С, D, E, H, L, которые могут объединяться в парные 16-разрядные регистры BC, DE, HL; два регистра временного хранения W и Z; указатель стека (SP); программный счетчик (PC), предназначенный для хранения адреса очередной команды (при выполнении линейных программ этот адрес автоматически увеличивается на 1, 2, 3 в зависимости от длины выполняемой команды – 1, 2 или 3 байта соответственно);

- счетчик (PC);
- регистр адреса (РА);
- двунаправленный мультиплексор для обмена информацией между АЛУ и блоком регистров узла внутренней памяти по внутренней шине данных;
- однонаправленный 16-разрядный буферный регистр адреса.

Узел управления. Узел управления вырабатывает в определенной последовательности микрооперации, необходимые для выполнения команд в МП. К узлу управления относятся: регистр команд (РгК), предназначенный для хра-

нения первого байта команды, содержащего код операции; дешифратор команд и формирователь машинных циклов; устройство управления и синхронизации, формирующая последовательности управляющих сигналов для работы АЛУ и регистров.

Устройство управления и синхронизации выполняет ряд функций управления и синхронизации: обеспечивает выборку команд и операндов; организует правильное функционирование АЛУ; обеспечивает доступ ко всем регистрам МП; синхронизирует УВВ и управляет их работой; приостанавливает работу МП в режиме ожидания и отключает МП от системной магистрали в режиме ПДП.

Регистры временного хранения, регистр команд, регистры W и Z, и регистр адреса пользователю программно недоступны. Кроме того, МП имеет 16-разрядный однонаправленный 3-стабильный канал адреса A0-A15, 8-разрядный двунаправленный 3-стабильный канал данных D0-D7, четыре входных (RESET, READY, INT, HOLD) и шесть выходных (SYNC, DBIN, READY, WAIT, INTE, HLDA) выводов сигналов управления.

Десятичный корректор облегчает работу с числами, представленными в двоично-десятичной системе счисления.

Буферные регистры данных и адреса используются для буферизации внутренних шин данных и адреса со стороны внешней магистрали.

Мультиплексор обеспечивает подключение к внутренней магистрали МП требуемого регистра из блока регистров.

МП КР580ВМ80А обеспечивает адресацию внешней памяти до 64 Кбайт и подключение до 256 устройств ввода-вывода.

Словесное описание работы микропроцессора

С целью упрощения понимания принципа работы МП, дадим вначале словесное описание его функционирования во время выполнения команд программы:

1. Перед выполнением очередной команды МП содержит ее адрес в программном счетчике РС;
2. МП обращается к памяти по адресу, содержащемуся в РС, и считывает из памяти первый байт очередной команды в регистр команд (PгК);
3. Дешифратор команд декодирует содержащийся в PгК код команды и в результате его декодирования, в частности, «узнает»: какова длина этой команды (1, 2 или 3 байта); где хранятся ее операнды; какие действия нужно выполнить над операндами;
4. В соответствии с полученной от дешифратора команд информацией устройство управления вырабатывает упорядоченную во времени последовательность микроопераций, реализующих предписания команды, в том числе: извлекает операнды из регистров и памяти; выполняет над ними предписанные кодом команды арифметические, логические или другие операции; в зависимо-

сти от длины команды модифицирует содержимое РС на 1, 2 или 3 (при линейном алгоритме); передает управление очередной команде, адрес которой снова находится в программном счетчике РС.

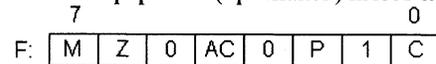
Рассмотрим, например, команду сложения содержимого аккумулятора и регистра В, имеющую мнемоническое обозначение ADD В. Команда ADD В – однобайтовая и имеет код операции 80h.

В начале выполнения этой команды МП выставляет на шину адреса адрес команды, считывает из памяти ее код 80h и помещает его в регистр команд (PгК). После декодирования команды устройство управления (УУ) вырабатывает предписанную командой последовательность управляющих сигналов, приводящую к следующим действиям: содержимое аккумулятора копируется в регистр временного хранения (Буф. А); содержимое регистра В копируется во второй регистр временного хранения; производится суммирование двух регистров временного хранения и результат сложения помещается в аккумулятор А; в зависимости от результата операции модифицируется содержимое регистра флагов F;

Содержимое программного счетчика РС увеличивается на 1, так как команда ADD В является однобайтовой, и теперь программный счетчик содержит адрес следующей команды программы и т.д.

Регистр флагов. Порядок выполнения команд программы зависит от особенностей (признаков) текущих результатов. Для индикации складывающейся ситуации МП формирует двоичные признаки (флаги), отражающие признаки результатов выполнения команд. Значение каждого признака фиксируется в специальном триггере, которые в совокупности образуют регистр кода условия. Содержимое аккумулятора и регистра флагов называют словом состояния программы PSW (Program Status Word).

Регистр флагов (признаков) имеет следующий формат:



В МП КР580ВМ80А имеется 5 флагов: С (Carry) – признак переноса; М (Minus) – признак отрицательного результата; Z (Zero) – признак нуля; Р (Parity) – признак паритета/четности; АС (Auxiliary Carry) признак половинного переноса.

Установка флагов производится при выполнении следующих условий: флаг знака М устанавливается, если знаковый бит результата операции (седьмой разряд аккумулятора) равен 1, иначе сбрасывается; флаг нуля Z устанавливается, если результат операции в аккумуляторе равен нулю, иначе сбрасывается; флаг дополнительного переноса АС устанавливается при наличии переноса из третьего разряда аккумулятора в четвертый, иначе сбрасывается; флаг четности Р устанавливается, если результат операции в аккумуляторе содержит четное число единиц, иначе сбрасывается; флаг переноса С устанавливается

при наличии переноса (при сложении) или заема (при вычитании) из старшего разряда аккумулятора, иначе сбрасывается.

Важнейшими флагами являются: флаг C, позволяющий реализовать на 8-разрядном МП обработку данных длиной в произвольное число байт; флаг Z, используемый для организации циклов и ветвлений; флаг M, используемый для организации ветвлений по знаку результата. Для организации ветвлений в составе команд микропроцессора предусмотрен обширный набор команд условной передачи управления, осуществляющих выбор одного из двух направлений перехода в зависимости от состояния проверяемого флага.

Регистры МП КР580. Блок программно доступных регистров МП ВМ80А отличается большой функциональной неоднородностью. Практически каждый регистр МП выполняет присущую только ему функцию, что сделано с целью более короткого кодирования системы команд микропроцессора. С другой стороны, это привело к усложнению программирования этого МП.

Восьмиразрядный аккумулятор А используется в подавляющем большинстве команд логической и арифметической обработки. Обычно он адресуется неявно и служит как источником операнда, так и приемником результата. Благодаря этому в командах ВМ80А явно указывается только один операнд.

16-разрядный регистр HL, как правило, служит адресным регистром. При косвенной регистровой адресации он хранит 16-разрядный адрес основной памяти. В этом случае к нему ссылаются с помощью мнемоники M (Memory), например:

```
MOV A, M; содержимое ячейки памяти, адрес которой  
; содержится ; в регистрах H и L, заносится в аккумулятор
```

В некоторых командах старший и младший байты 16-разрядного регистра HL могут адресоваться независимо и использоваться как отдельные 8-разрядные регистры данных H (High Byte) и L (Low Byte) соответственно.

Регистры PC и SP выполняют свою обычную функцию счетчика команд и указателя стека.

8-разрядные регистры F, A вместе с 16-разрядными регистрами HL, SP и PC образуют стандартный регистровый набор микропроцессора с аккумулятором. Этот набор расширен четырьмя 8-разрядными регистрами общего назначения (РОН): B, C, D, E, которые в некоторых командах объединяются в 16-разрядные парные регистры BC и DE. Младшими регистрами пары являются соответственно регистры C и E. Введение РОН позволило создать достаточно эффективный микропроцессор с широкими функциональными возможностями.

В командах ссылки на регистр выполняются как явно, когда регистр кодируется трехразрядным полем, так и неявно (с помощью кода операции), когда код операции подразумевает и способ использования регистров по умолчанию.

Пространство памяти и ввода-вывода МП КР580ВМ80А

Адресное пространство памяти, образующее основную память МП КР580ВМ80А, представляет собой линейный одномерный массив из 64 Кб ячеек памяти, так что 16-разрядный адрес дает возможность микропроцессору непосредственно адресовать любой байт памяти. Данные в памяти могут храниться байтами или словами. Слова в памяти хранятся в двух соседних байтах (младший байт слова хранится по младшему адресу). Адресом слова является адрес его младшего байта.

В МП определено 4 способа задания месторасположения данных в памяти: прямой; косвенный регистровый (через 16-разрядные регистры HL, BC и DE); непосредственный; автоинкрементный/автодекрементный через указатель стека SP.

При прямой и непосредственной адресации данных могут быть доступны байты или слова, при косвенной адресации – только байты. Стековая адресация применяется только при работе со словами.

В МП ВМ80А используется изолированное адресное пространство ввода-вывода. Эта область организована в виде одномерного массива из 256 восьмиразрядных портов ввода и 256 восьмиразрядных портов вывода. Допускается только прямой способ доступа к пространству ВВ.

Лабораторные задания

Задание первое. Выполнить директивы Монитора для микроЭВМ «Партнер 01.01». Для выполнения этого задания выполнить следующие действия:

1. С рабочего стола открыть файл «ASM-ED.RKP (эмулятор микроЭВМ «Партнер 01.01»). После запуска эмулятора сразу же запускается программа «МОНИТОР» микроЭВМ «Партнер 0101». На экране ПК появится надпись:

```
ПАРТНЕР – 01.01 (В. 03)
```

```
:->
```

означающая что мы загрузили программу «Монитор» микроЭВМ «Партнер 01.01.» (рис. 2).

2. Выполнить директивы МОНИТОРа: D, L, F, M, X (см. раздел «Директивы Монитора»).

Задание второе. Изучить назначение системной программы РЕДАКТОР. РЕДАКТОР предназначен для редактирования любого текста непосредственно на экране дисплея: исправление ошибок, удаление и ввод новых символов и фрагментов текста. РЕДАКТОР при необходимости компонует из отдельных частей полный текст. Стереть ненужные символы можно клавишей Delete, при этом подвести курсор под символ. При наборе программы необходимо использовать информацию прил. 1-3.

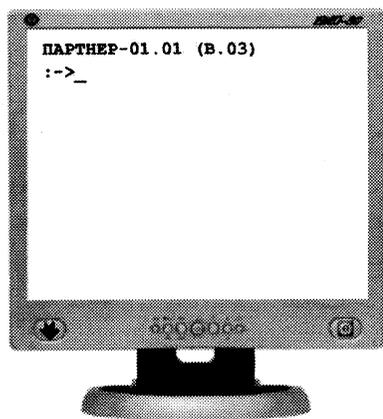


Рис. 2

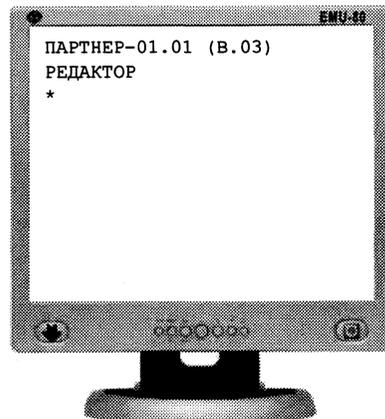


Рис. 3

Запуск РЕДАКТОРА:

1. Загрузить программу «МОНИТОР», выполнив действия пункта 1 первого задания;

2. Нажать последовательно клавиши ПК <G> и <Enter>.

При первом запуске Редактора необходимо очистить буфер ОЗУ с помощью последовательного нажатия клавиш Esc и N латинского алфавита. Такое действие будем именовать директивой и обозначать Esc и N. По этой директиве очистится экран и на запрос NEW? Можно ответить Y, в этом случае текстовый буфер в памяти компьютера очистится и РЕДАКТОР перейдет в режим ввода строки (в начале первой строки появится символ «->»), в противном случае установится режим редактирования (в начале строки – символ «*»).

Режим ввода строки используется для ввода текста с клавиатуры компьютера. Строка может состоять не более чем из 63 символов. Набор строки завершают нажатием клавиши Enter, после чего она пересылается в текстовый буфер. Допущенную при вводе ошибку можно исправить, сместив курсор назад до нужного места. Устранив ошибку, его можно вернуть в любую позицию в строке. Символ можно удалить из строки установкой курсора под соответствующим знаком и нажатием на клавишу Del.

После нажатия Enter в текстовый буфер заносится символ, под которым находился курсор, и все символы, находящиеся слева от него. Чтобы перейти из режима ввода строки в режим редактирования необходимо нажать на клавишу PgUp.

Для перехода в режим ввода строки достаточно нажать на клавиши Esc + E + ↓ (поочередное нажатие).

Из Редактора, находящегося в режиме редактирования (*) можно вызвать Ассемблер дважды нажав на клавишу PgUp.

Из Ассемблера переход в Редактор в режим редактирования осуществляется нажатием на клавишу PgUp.

В табл. 1 приведены основные директивы РЕДАКТОРА.

Таблица 1

Директивы РЕДАКТОРА

Директива	Выполняемое действие
ESC+N	НАЧАЛО РАБОТЫ, очистка экрана, варианты: - при ответе Y на запрос NEW? - очистка текстового буфера и установка режима ввода строки; - при нажатии <Enter> на запрос NEW? - установка режима редактирования без очистки текстового буфера (в начале строки – символ «*»).
Page Up	Выход в АССЕМБЛЕР
ESC и E (одновременно)	Выход из АССЕМБЛЕРА в МОНИТОР.
РЕЖИМ ВВОДА СТРОК.	
ENTER	Ввод в память набранной строки.
→,←	Перемещение курсора для исправления строки.
Page Up	Завершение ввода строк.
РЕДАКТИРОВАНИЕ ТЕКСТА	
Home	Перемещение курсора к началу первой строки
→,↓,←,↑	Перемещение курсора к месту редактирования
↓	Перевод РЕДАКТОРА (в режим ввода строки («->»)) и фиксирование отредактированного текста программы.
Ctrl и E	Вход в монитор не стирая экран памяти

Задание третье. Изучить назначение системной программы АССЕМБЛЕР и основные директивы АССЕМБЛЕРА.

АССЕМБЛЕР - программа - транслятор для разработки программ на языке Ассемблера микропроцессора КР580ВМ80А. Текст программы с клавиатуры с помощью РЕДАКТОРА заносят в область текстового буфера ОЗУ, а затем с помощью АССЕМБЛЕРА транслируют. Результат трансляции - машинные коды располагаются в другой области ОЗУ, называемой областью трансляции. Вход в АССЕМБЛЕР из РЕДАКТОРА производится двойным нажатием клавиши Page Up.

Требования к программам

1. Программа на языке Ассемблера, предназначенная для трансляции, должна состоять из строк, которые имеют следующий формат записи:
метка : код операнды ; комментарий.

Пример: MET1: XRA A ; ОБНУЛЕНИЕ АККУМУЛЯТОРА

2. Метки должны состоять из последовательности латинских букв и цифр число которых не более пяти и начинаться с буквы. Если имя содержит более пяти знаков, то воспринимаются только первые пять из них. В качестве меток нельзя использовать имена регистров A,B,C,D,E,H,L. и мнемонику команд процессора.

3. В программах можно использовать псевдооператор ORG, END, DB, DW, DS, EQU. Имена меток перед псевдооператорами должны заканчиваться ":", за исключением EQU.

4. После шестнадцатеричного числа ставят латинскую букву H. В случае, если число начинается с буквы, впереди добавляют цифру 0.

Трансляция начинается при нажатии на одну из следующих клавиш:

«1» – трансляция с выводом на экран протокола трансляции, представляющего собой строки исходного текста программы, перед которыми в шестнадцатеричной системе выводятся коды ошибок, адреса размещения команд и данных в ОЗУ и машинные коды транслируемой программы;

«2» – после трансляции программы на дисплее отображается»-перечень встретившихся в ней имен меток в алфавитном порядке и их в шестнадцатеричные адреса;

«3» – текст программы транслируется в машинные коды и выводится сообщение о результатах трансляции: число ошибок и два шестнадцатеричных числа: старший адрес программы в области загрузки и в области трансляции (ограниченный символами «/»).

Выйти из Ассемблера в Монитор можно при одновременном нажатии на клавиши **Ctrl** и **E**. Директивы АССЕМБЛЕРА приведен в табл. 2.

Таблица 2

Директивы АССЕМБЛЕРА.

Директива	Выполняемое действие
1	Трансляция программы с отображением на экране протокола трансляции.
2	Трансляция программы с отображением на экране таблицы меток.
3	Трансляция программы с отображением на экране количества ошибок и адресов транслированной программы.
Page Up	Выход в РЕДАКТОР (в режим редактирования (*)) .
Ctrl и E (одновременно)	Выход в МОНИТОР

Примеры листинга оттранслированных и выполненных программ представлен на рис. 4. Выполните набор этих программ и получите аналогичный результат. Сделайте выводы.

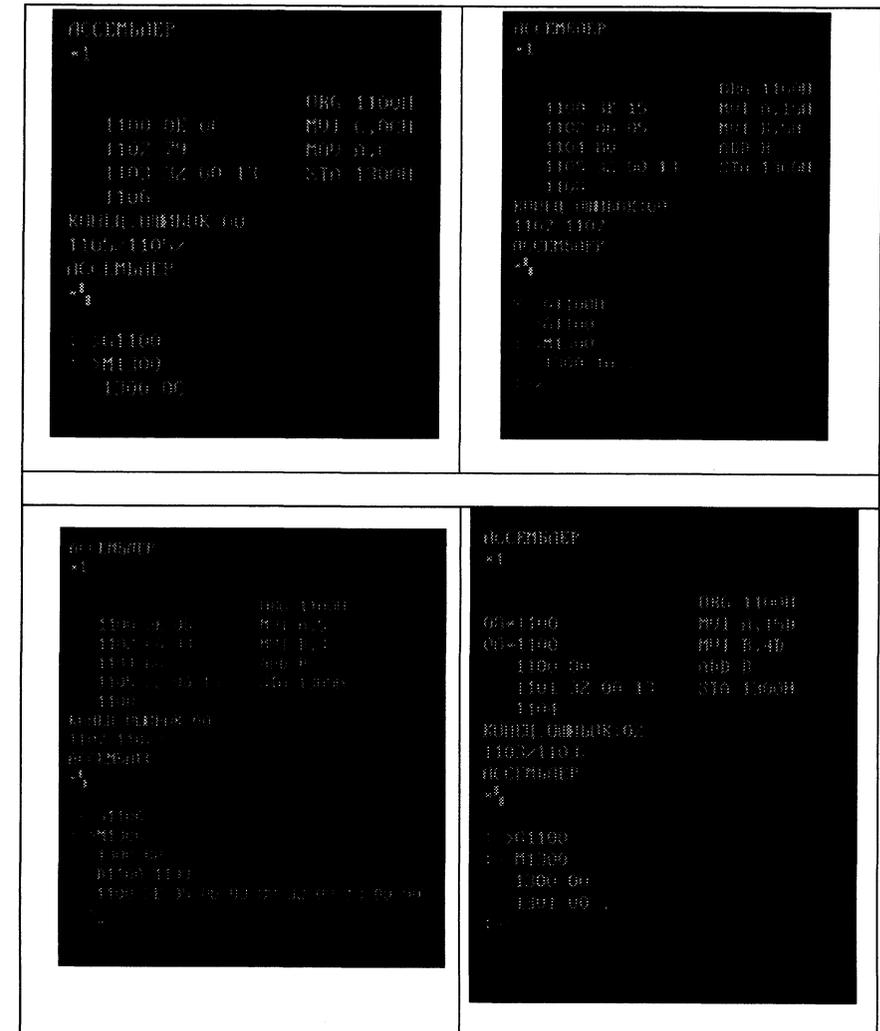


Рис. 4

Задание четвертое. Текст программы, представленной ниже с помощью РЕДАКТОРА, занесите в область текстового буфера ОЗУ:

```
ORG 1100H
MVI C,0CH
MOV A,C
STA 1300H
```

а затем с помощью АССЕМБЛЕРА оттранслируйте. Запишите результат трансляции – машинные коды вместе с программой. Если результат трансляции без ошибок, выполните программу и убедитесь в правильности результата.

В задании 4 вставьте необходимый комментарий, используя прил. 1и 2.

Задание пятое. Текст программы, представленной ниже с помощью РЕДАКТОРА, занесите в область текстового буфера ОЗУ:

```
ORG 1100H
MVI C,0CH
MOV A,C
STA 1300H
```

Перейдите из режима ввода строки в режим редактирования и сделайте изменение в тексте программы, а затем выполните ее и посмотрите результат. В задании 5 оставьте необходимый комментарий.

Задание шестое. С рабочего стола загрузить эмулятор восьмиразрядной ЭВМ KP580. На экране монитора появится главное окно программы (рис. 5):

Содержимое главного окна программы:

1. Главное меню программы;
2. Структурная схема МП-системы;
3. Таблица содержимого ОЗУ МП-системы;
4. Внешние периферийные устройства, подключенные к портам МП-системы;
5. Панель редактирования значений выбранной (текущей) ячейки ОЗУ МП-системы;
6. Панель редактирования значений содержимого выбранного регистра общего назначения (РОН) МП-системы;
7. Группа кнопок «Сброс» для обнуления всех ячеек ОЗУ и регистров общего назначения (РОН) МП-системы;
8. Панель системы команд МП KP580BM80A (скрытый вид);
9. Группа кнопок «Выполнение» для выполнения программы МП-системой в сквозном, командном и тактовом режимах.

В поле окна 3 разместите программу пятого задания в машинных кодах начиная с адреса 1100H. Используя группу кнопок поля 9 выполнить программу в сквозном, командном и тактовом режимах. Посчитайте количество байт этой программы и время её выполнения для тактовой частоты процессора равной 2 МГц.

Контрольные вопросы по выполненной работе

1. Основные принципы работы микро-ЭВМ
2. Назначение основных узлов (на уровне программно-доступных микро-процессора KP580BM80A.
3. Программное обеспечение микро-ЭВМ «Партнер 01.01».
4. Назначение эмулятора «ПАРТНЕР01.01». Требования к аппаратному обеспечению. Требования к ОС и её особенности. Основы работы с программой.
5. Назначение программы МОНИТОР и основные директивы МОНИТОРА.
6. Назначение программы РЕДАКТОР и основные директивы РЕДАКТОРА.
7. Запуск РЕДАКТОРА. Основные режимы работы РЕДАКТОРА: режим ввода строк, режим редактирования. Основные действия, выполняемые в этих режимах. Переход из одного режима в другой.
8. Исправление ошибок в тексте или программе.
9. Назначение программы АССЕМБЛЕР и основные директивы. АССЕМБЛЕРА.
10. Требования к программам, написанным на языке АССЕМБЛЕРА.
11. Запуск программ Ассемблер, Монитор, Редактор, Бейсик.
12. Переход из Ассемблера в Редактор.

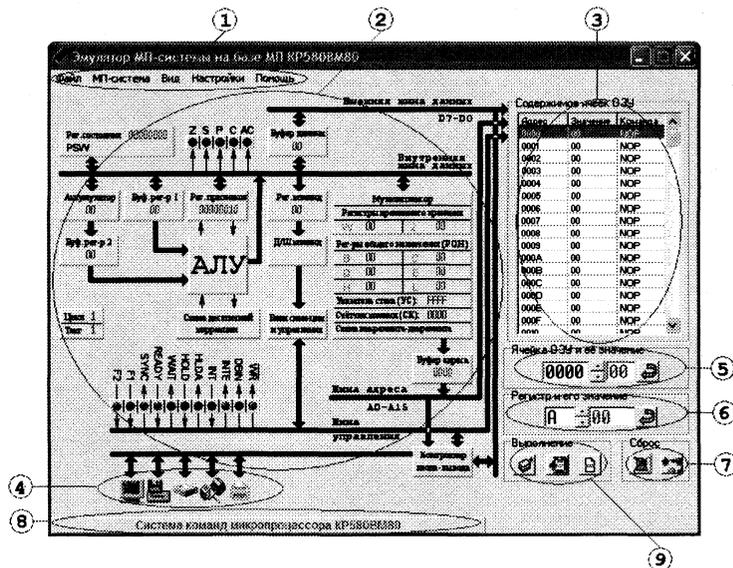


Рис. 5. Главное окно программы

Требования к выполнению отчёта

Отчёт выполняется на стандартных листах бумаги формата А4 (210*297 мм) и должен содержать следующее: структурную схему микропроцессора К580ВМ80А, результат выполнения всех заданий вместе с программой проверки.

ЛАБОРАТОРНАЯ РАБОТА № 2 ИЗУЧЕНИЕ КОМАНД РАБОТЫ С РЕГИСТРАМИ МИКРОПРОЦЕССОРА И ПАМЯТЬЮ

Общие указания

Цель работы: изучение команд загрузки регистров, команд пересылки, команд работы с памятью.

Исследуются программы, позволяющие осуществлять загрузку регистров, пересылку информации из одного регистра в другой, команды записи или чтения памяти.

Домашние задания и методические указания по их выполнению

Задание первое. Ознакомьтесь с языком программирования и структурой команд загрузки регистров, команд пересылки и команд работы с памятью МП КР580ВМ80А [1, С. 84-88, С.95-107], [2, С.302-318, С.362-391], [Приложения 1-3 методических указаний].

Команды этой группы осуществляют обмен данными между регистрами блока РОН и памятью МП-системы. Признаки состояния процессора не изменяются в процессе выполнения этих команд.

MOV R1, R2 – передача данных из регистра R2 на регистр R1.

MOV R, M – передача данных из ячейки памяти с адресом, который находится в регистрах H (старший полуадрес) и L (младший полуадрес) в регистр R.

MOV M, R – передача данных из регистра R в ячейку памяти с адресом, который находится в регистрах H и L.

MVI R, data – содержимое второго байта команды передается в регистр R.

MVI M, data – содержимое второго байта команды передается в ячейку памяти адресуемой содержимым регистров H и L.

LXI гр, data16 – загрузить непосредственный 16-разрядный операнд в регистровую пару гр, например BC,DE,HL,SP, причем содержимое второго (старшего) байта команды передается в младший регистр регистровой пары(C,E,L,P), а третий (младший) байт команды передается в старший регистр регистровой пары (B,D,H,S). В команде указывается только старший регистр регистровой пары (B,D,H). После трансляции команда в памяти располагается следующим образом: КОП, младший байт операнда по младшему адресу, старший байт операнда по старшему адресу.

LDA addr – загрузить аккумулятор (регистр A) содержимым ячейки памяти с адресом addr (второй и третий байт команды).

STA addr – записать содержимое аккумулятора A передать в ячейку памяти с адресом addr.

LHLD addr – загрузка регистров H и L из памяти. При этом в регистр L будет записано содержимое ячейки с адресом addr, в регистр H содержимое ячейки с адресом addr +1.

SHLD addr – записать в память содержимое регистровой пары HL, причем по адресу addr, будет записано содержимое регистра L, а по адресу addr+1 содержимое регистра H.

LXI SP, BHBL – загрузка регистра указателя стека, где BHBL -значение операнда (BH-старший байт, BL-младший-байт).

SPHL – в указатель стека загружается содержимое регистровой пары HL.

PCHL – в счетчик команд записывается содержимое пары регистров HL. Содержимое регистра H занести в восемь старших разрядов счётчика команд РС. Содержимое регистра L занести в восемь младших разрядов счётчика команд РС.

LDAX гр – загрузить A с косвенной адресацией (содержимое ячейки памяти, адрес которой содержится в паре регистров (только в BC или DE) загружается в регистр A).

STAX гр – записать в память содержимое A по адресу, который содержится в паре регистров (только в BC или DE).

XCHG – обменять содержимое пары регистров H и L с содержимым регистров D и E.

Задание второе. Составьте на АССЕМБЛЕРЕ и в машинных кодах программу загрузки регистров общего назначения A, B, C, D, E, H, L соответственно числами 1, 2, 3, 4, 5, 6, 7.

Для выполнения задания необходимо использовать команды загрузки регистров общего пользования, общий вид которых:

MVI R, data, где R -регистр – место назначения: A, B, C, D, E, H, L,M; data – однобайтовый 8-разрядный непосредственный (0...255) операнд – число/имя.

Для примера составим программу записи числа 5 в регистры A, B.

Адрес	Команда	Машинный код	Комментарий
800	MVI A,5H	3E 05	Загр.рег. A = 05H
802	MVI B,5H	06 05	Загр.рег. B = 05H

Задание третье. Составьте программу загрузки для регистров BC, DE, HL соответственно 16-разрядными данными 2122H, 2223H, 2324H.

Для выполнения задания необходимо использовать команду LXI гр, data16. LXI – загрузить непосредственный 16-разрядный операнд в регистровую пару гр (data16 – двухбайтовый операнд).

Для примера составим программу записи числа 4856H в пары регистров BC и DE:

Адрес	Команда	Машинный код	Комментарий
805	LXI B, 4856H	01 56 48	;Загр.пары рег. BC=4856H
808	LXI D, 4856H	11 56 48	;Загр.пары рег. DE=4856H

Задание четвертое. Составьте программу загрузки регистра указателя стека и счетчика команд данными 0891H, 0924H, 09ACH. Задание по загрузке регистра указателя стека выполнить двумя способами с использованием команды непосредственной загрузки: LXI SP, BHBL где BHBL – значение операнда; BH – старший байт; BL – младший байт и с использованием команды косвенной загрузки SPHL.

В первом случае программа загрузки числа 0891H будет иметь вид:

Адрес	Команда	Машинный код	Комментарий
810	LXI SP, 0891H	31 91 08	;Загрузка указателя стека SP= 0891H

Во втором случае при использовании команды SPHL в указатель стека загружается содержимое регистровой пары HL. Поэтому, чтобы в указатель стека загрузить, например число 12A3H, его предварительно надо загрузить в регистровую пару HL:

Адрес	Команда	Машинный код	Комментарий
815	LXI H, 12A3H	21 A3 12	;Загр.HL =12A3H
818	SPHL	F9	;Загр. SP = HL

Для записи числа в счетчик команд необходимо использовать команду PCHL, по которой в счетчик команд записывается содержимое пары регистров HL, например:

Адрес	Команда	Машинный код	Комментарий
820	LXI H,0891H	21 91 08	;Загр.HL= 0891H
823	PCHL	E9	;Загр.сч.команд PC=HL.

Задание пятое. Составьте программу перехода числа 22H из аккумулятора A последовательно в регистры B, C, D, E, H, L.

Для выполнения задания используются команды пересылки, общий вид которых 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. Например, программа пересылки числа 22H из аккумулятора в регистры B, C будет иметь вид:

Адрес	Команда	Машинный код	Комментарий
820	MVI A,22H	3E 22	;Загр.рег. A = 22H
822	MOV B,A	47	;Пересылка B ← A
823	MOV C,B	48	;Пересылка C ← B

Задание шестое. Составьте программу загрузки числа 55H в ячейки памяти с адресами 900, 901, 902, 903.

Существуют две команды непосредственной записи в память: STA addr-запись содержимого аккумулятора (регистра A) в память по непосредственному адресу addr и вторая команда SHLD addr-запись в память содержимого регистровой пары HL в две соседние ячейки памяти, причем по адресу addr будет записано содержимое регистра L, а по адресу addr +1 будет записано содержимое регистра H, например:

Адрес	Команда	Машинный код	Комментарий
810	STA 910H	32 10 09	;MEM(910H) ← A
813	SHLD 910H	22 10 09	;MEM(910H)←L, ;MEM(911H)←H

Задание седьмое. Составьте программы загрузки регистров B, C, D, E, H, L числами соответственно 1, 2, 3, 4, 5, 6, используя команды чтения из памяти LDA addr – чтение памяти по непосредственному адресу addr в регистр A и LHLD addr – чтение памяти по непосредственному адресу addr в регистровую пару HL. При этом в регистре L будет записано содержимое ячейки с адресом addr, в регистре H содержимое ячейки с адресом addr +1. Например, программа загрузки (чтения) в регистры A, L, H содержимого ячеек памяти 860H, 870H и 871H будет иметь вид:

Адрес	Команда	Машинный код	Комментарий
810	LDA 860H	3A 60 08	;A←MEM(860H)
813	LHLD 870H	2A 70 08	;L←MEM(870H), H←MEM(871H)

Задание восьмое. Составьте программы записи и чтения содержимого регистров A, B, C,D, E, используя команды чтения/записи памяти при адресации через регистровую пару HL. Общий вид команд: MOV M, R - запись в па-

мять содержимого регистра; MOV R, M - загрузка регистра из памяти, где R - регистр общего пользования: A, B, C, D, E.

При составлении программы надо иметь в виду, что предварительно необходимо записать в регистровую пару HL адрес ячейки памяти, в которую производится запись-чтение данных. Общий вид команды: LXI H,data16 - загрузить непосредственный 16-разрядный операнд в регистровую пару HL (data16 – двухбайтовый операнд). Например, программа записи в ячейку памяти с адресом 900H содержимого регистра A будет иметь вид:

Адрес	Команда	Машинный код	Комментарий
800	LXI H,900H	21 00 09	;загр. HL=900H, адр.М
803	MOV M, A	77	;зп. M=A, по адр. HL

Программа чтения данных из ячейки памяти с адресом 900H в регистр A будет иметь вид:

Адрес	Команда	Машинный код	Комментарий
800	LXI H,900H	21 00 09	;загр. HL=900H, адр.М
803	MOV A, M	7E	;чт. A=M по адр. HL

Лабораторные задания

1. С помощью РЕДАКТОРа занесите в область текстового буфера ОЗУ программу загрузки регистров общего пользования A, B, C, D, E, H, L соответственно числам 1,2,3, 4, 5, 6, 7 а затем с помощью АССЕМБЛЕРа оттранслируйте. Запишите результат трансляции - машинные коды вместе с листингом программы. Если результат трансляции без ошибок выполните программу. Просмотрите содержимое регистров общего назначения.

2. Запишите в память программу загрузки пар регистров BC, DE, HL соответственно 16-разрядными данными 2122H, 2223H, 2324H. Выполните эту программу и проверьте правильность записи данных в регистры общего назначения.

3. Запишите в память программу загрузки регистра указателя стека и счетчика команд данными 0891H, 0924H, 09ACH. Выполните программу и проверьте содержимое указанных регистров.

4. Запишите в память программу пересылки числа AA из аккумулятора A последовательно в регистры B, C, D, E, H, L. Выполните ее и проверьте правильность работы.

5. Запишите в память программу загрузки числа 55H в ячейки памяти с адресами 900H, 901H, 902H, 903H. Выполните ее. Посмотрите содержимое ячеек памяти с адресами 900H, 901H, 902H, 903H. Оно должно быть равно 55H.

6. Запишите в память программы загрузки регистров B, C, D, E, H, L числами соответственно 1, 2, 3, 4, 5, 6 с использованием команд чтения памяти по

непосредственному адресу LDA addr и LHLD addr. Выполните их. Просмотрите содержимое регистров и убедитесь в правильности записи.

7. Запишите в память программы записи и чтения содержимого регистров A, B, C, D, E, используя команды чтения/записи памяти при адресации через регистровую пару HL. Выполните их, предварительно записав в регистры A, B, C, D, E, или соответственно в ячейки памяти числа 10, 11, 12, 13, 14. Просмотрите содержимое регистров (или памяти) и убедитесь в правильности ее работы.

Оформление отчёта

Отчет должен содержать:

1. Тексты программ в соответствии с пунктами домашнего задания.
2. Пояснения к исправлениям ошибок в тексте программы, выявленным в процессе их отладки.
3. Выводы.

Контрольные вопросы по выполненной работе

1. Перечислите регистры, доступные для программирования в микропроцессоре КР580ВМ80А.
2. Объясните отличие регистра аккумулятора от других регистров общего назначения.
3. Какие команды используются для загрузки регистров общего назначения?
4. Какие команды используются при непосредственной записи в память?
5. Приведите команды обращения к памяти с косвенной адресацией. [1, С. 84-88, С.95-107], [2, С.302-318, С.362-391].

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Новожилов, О. П. Основы микропроцессорной техники: учеб. пособие [Текст] / О. П. Новожилов. – В 2-х томах. – Т.1. – М.: ИП РадиоСофт, 2007. – 432 с.
2. Мышляева И. М. Цифровая схемотехника: учебник для сред. проф. образования [Текст] / И. М. Мышляева. – М.: Академия, 2005. – 400 с.

**СИСТЕМА КОМАНД МИКРОПРОЦЕССОРА K580
(ШЕСТНАДЦАТЕРИЧНАЯ КОДИРОВКА)**

I. ГРУППА КОМАНД ПЕРЕСЫЛКИ (признаки F не формируются)

1. Команда MOV – пересылка из регистра в регистр, из регистра в память (MEM), из памяти (MEM) в регистр (R – регистр-источник).

	Регистр-источник операнда								Примечания
	A	B	C	D	E	H	L	M	
MOV A,R	7F	78	79	7A	7B	7C	7D	7E	Межрегистровая передача A←R для R=A,B,C,D, E,H, L. Передача из памяти A←MEM(HL) для R=M
MOV B,R	47	40	41	42	43	44	45	46	
MOV C,R	4F	48	49	4A	4B	4C	4D	4E	
MOV D,R	57	50	51	52	53	54	55	56	
MOV E,R	5F	58	59	5A	5B	5C	5D	5E	
MOV H,R	67	60	61	62	63	64	65	66	
MOV L,R	6F	68	69	6A	6B	6C	6D	6E	
MOV M,R	77	70	71	72	73	74	75	-	Передача в память MEM(HL)←R

2. Команда MVI – пересылка непосредственного операнда в регистр или память, data – однобайтовый операнд.

Команда	Регистр – место назначения								Примечание
	A	B	C	D	E	H	L	M	
MVI R,data	3E	06	0E	16	1E	26	2E	36	R←data для R≠M, MEM(HL)←data; для R=M
MVI M,data									

3. Команда LXI – загрузить непосредственный 16-разрядный операнд в регистровую пару rp, (data16 – двухбайтовый операнд).

Команда	Адресуемая регистровая пара				Примечание
	BC	DE	HL	SP	
LXI rp,data16	01	11	21	31	rp←data16

4. Команды загрузки регистровых пар addr – 16-разрядный адрес, находящийся во 2-м и 3-м байтах команды, (rp – регистровая пара).

Команда	Адресуемая пара, rp				Примечания
	A	BC	DE	HL	
LDAX rp		0A	1A		A←MEM(rp)
LHLD addr				2A	L←MEM(addr); H←MEM(addr+1)
LDA addr	3A				A←MEM(addr)
STAX rp		02	12		MEM(rp)←A
SHLD addr				22	MEM(addr)←L; MEM(addr+1)←H
STA addr	32				MEM(addr)←A
XCHG				EB	H↔D; L↔E

II. ГРУППА АРИФМЕТИЧЕСКИХ И ЛОГИЧЕСКИХ КОМАНД

1. Команды операций над содержимым аккумулятора A и регистра.

Они вырабатывают все признаки F (команды INR и DCR, не вырабатывают признак CF). Для R=M все команды выполняют действия с ячейкой памяти, на которую указывает содержимое регистровой пары HL. (A←A + MEM(HL)).

Ком-да	Регистр-источник операнда								Примечания
	A	B	C	D	E	H	L	M	
ADD R	87	80	81	82	83	84	85	86	A←A + R
ADC R	8F	88	89	8A	8B	8C	8D	8E	A←A + R + CF
SUB R	97	90	91	92	93	94	95	96	A←A - R
SBB R	9F	98	99	9A	9B	9C	9D	9E	A←A - R - CF
ANA R	A7	A0	A1	A2	A3	A4	A5	A6	A←A∧R лог. И
XRA R	AF	A8	A9	AA	AB	AC	AD	AE	A←A⊕R иск. ИЛИ
ORA R	B7	B0	B1	B2	B3	B4	B5	B6	A←A∨R лог. ИЛИ
CMP R	BF	B8	B9	BA	BB	BC	BD	BE	A - R опер.срав.
INR R	3C	04	0C	14	1C	24	2C	34	R←R + 1
DCR R	3D	05	0D	15	1D	25	2D	35	R←R - 1

2. Команды операций с регистровыми парами. Эти команды используют операнды двойной длины (2 байта). Из них только команда двойного сложения DAD формирует признак CF.

Команда	Адресуемая регистровая пара, rp				Примечание
	BC	DE	HL	SP	
DAD rp	09	19	29	39	(HL)←(HL) + (rp)
INX rp	03	13	23	33	rp←rp + 1
DCX rp	0B	1B	2B	3B	rp←rp - 1

3. Команды операций, выполняемых с непосредственным однобайтовым операндом (data). Команды формируют все признаки F.

Команда	Код	Примечания
ADI data	C6	A←A + data
ACI data	CE	A←A + data + CF
SUI data	D6	A←A - data
SBI data	DE	A - data - CF
ANI data	E6	A←A ^ data логическое И
XRI data	EE	A←A ⊕ data искл. ИЛИ
ORI data	F6	A←A ∨ data логическое ИЛИ
CPI data	FE	A - data сравнение

4. Команды специальных операций и сдвига аккумулятора А.

Команда	Код	Примечание	
DAA	27	Десятичная коррекция АКК, Формирует все признаки	
CMA	2F	Инвертирование А. Признаки F не формирует	
STC	37	Установка признака C _F , C _F ←1	
CMC	3F	Инвертирование признака C _F	Формируется только признак C _F
RLC	07	Циклический сдвиг А влево	
RRC	0F	Циклический сдвиг А вправо	
RAL	17	Сдвиг А влево	
RAR	1F	Сдвиг А вправо	

III. ГРУППА КОМАНД ПЕРЕДАЧИ УПРАВЛЕНИЯ

Команды перехода (JMP), вызова подпрограмм (CALL), возврата из подпрограмм (RET) и повторного пуска (RST) не формируют признаков F. Команда безусловного перехода по косвенному адресу PCHL - E9.

Команда Условия	JMP	CALL	RET	n	RST n
	безусловная	C3	CD		
NZ по ненулю (Z=0)	C2	C4	C0	0	C7
Z по нулю (Z = 1)	CA	CC	C8	1	CF
NC по непереносу (C _F =1)	D2	D4	D0	2	D7
C по переносу (P=0)	DA	DC	D8	3	DF
PO по нечетности (P=0)	E2	E4	E0	4	E7
PE по четности (P=1)	EA	EC	E8	5	EF
P по знаку плюс (S=0)	F2	F4	F0	6	F7
M по знаку минус (S=1)	FA	FC	F8	7	FF

IV. ГРУППА КОМАНД УПРАВЛЕНИЯ СТЕКОМ, ВВОДОМ-ВЫВОДОМ, СОСТОЯНИЕМ МП

1. Команды управления стеком. Признаки формируют только команды POP PSW

Команда	Адресуемая регистровая пара				Примечание
	BC	DE	HL	PSW	
PUSH rp	C5	D5	E5	F5	MEM (SP-1) ← rh MEM (SP-2) ← rl SP=SP-2
POP rp	C1	D1	E1	F1	rh ← MEM (SP) rl ← MEM (SP+1) SP=SP+2
XTHL			E3		L ↔ MEM (SP) H ↔ MEM (SP+1)
SPHL			F9		SP ← HL

Примечание. Старший байт (rh) регистровой пары первым помещается в стек. Младший байт (rl) регистровой пары первым извлекается из стека.
2. Команды ввода (IN), вывода (OUT), прерывания (DI,EI) и остановка (HLT), (port – адрес порта).

Команда	Код	Примечание
OUT port	D3	Вывод из порта 00 – FF
IN port	DB	Ввод в порт 00 – FF
DI	F3	Запрет прерывания
EI	FB	Разрешение прерывания
NOP	00	Без операции
HLT	76	Останов

ПРИЛОЖЕНИЕ 2

СООТВЕТСТВИЕ КЛАВИШ

Клавиша	Значение	Значение с удерживаемым Shift
—	-	=
+ =	;	+
[X	[
] Ъ]	
: ж	:	*
/ ю	/	?
·	.	>
, Б	,	<
@	@	
1	1	!
2	2	"
3	3	#
4	4	α
5	5	%
6	6	&
7	7	'
8	8	(
9	9)
0	0	
CapsLock (дважды нажать)	Рус/Лат	Набор русских букв на латинской клавиатуре
Отпустить Shift/ нажать Shift	Лат/Рус	Набор лат/рус

ПРИЛОЖЕНИЕ 3
НАБОР РУССКИХ БУКВ НА ЛАТИНСКОЙ КЛАВИАТУРЕ

Рус	А	Б	В	Г	Д	Е	Ё	Ж	З	И	Й	К	Л	М	Н
Лат	A	B	W	G	D	E		V	Z	I	J	K	L	M	N
Рус	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь
Лат	O	P	R	S	T	U	F	H	C		{	}		Y	X
Рус	Э	Ю	Я	:	,	.	/	;							
Лат			Q	:	<	>	?	+							

ОГЛАВЛЕНИЕ

Введение.....	3
Лабораторная работа № 1. Программирование на языке Ассемблера для восьмиразрядного микропроцессора с использованием эмуляторов ЕМУ-80 и КР580.....	3
Лабораторная работа № 2. Изучение команд работы с регистрами микропроцессора и памятью.....	18
Библиографический список.....	23
Приложение 1. Система команд микропроцессора К580 (шестнадцатеричная кодировка).....	24
Приложение 2. Соответствие клавиш	27
Приложение 3. Набор русских букв на латинской клавиатуре.....	28

ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ АССЕМБЛЕРА
ДЛЯ ВОСЬМИРАЗЯДНОГО МИКРОПРОЦЕССОРА

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

к выполнению лабораторных работ по дисциплине
«Цифровые устройства и микропроцессоры» для студентов направления
11.03.01 «Радиотехника» (профиль «Радиотехнические средства передачи,
приема и обработки сигналов») всех форм обучения

Составители:

Кондусов Василий Ананьевич
Алперин Евгений Данилович

Редактор Е. А. Четвертухина

Подписано в печать 02.12.2021.

Формат 60x84 1/16. Бумага для множительных аппаратов.
Уч.-изд. л. 1,7. Усл. печ. л. 1,6. Тираж 63 экз. Заказ № 188.

ФГБОУ ВО «Воронежский государственный технический университет»
394026 Воронеж, Московский проспект, 14

Участок оперативной полиграфии издательства ВГТУ
394026 Воронеж, Московский проспект, 14