



Министерство образования Республики Беларусь

Учреждение образования
«Гомельский государственный технический
университет имени П. О. Сухого»

Кафедра «Теоретические основы электротехники»

П. П. Изотов

МИКРОПРОЦЕССОРНАЯ ТЕХНИКА

ЛАБОРАТОРНЫЙ ПРАКТИКУМ

**по курсу «Электроника и микропроцессорная
техника» для студентов специальностей 1-36 02 01
«Машины и технология литейного производства»,
1-36 01 05 «Машины и технология обработки
материалов давлением», 1-36 20 02 «Упаковочное
производство (по направлениям)»
дневной и заочной форм обучения**

Электронный аналог печатного издания

Гомель 2008

УДК 004.315(075.8)
ББК 32.973.26-04я73
ИЗ8

*Рекомендовано к изданию научно-методическим советом
механико-технологического факультета ГГТУ им. П. О. Сухого
(протокол № 6 от 19.12.2006 г.)*

Рецензент: зав. каф. «Электроснабжение» ГГТУ им. П. О. Сухого канд. техн.
наук, доц. *А. В. Сычев*

Изотов, П. П.

ИЗ8 Микропроцессорная техника : лаб. практикум по курсу «Электроника и микропроцессорная техника» для студентов специальностей 1-36 02 01 «Машины и технология литейного производства», 1-36 01 05 «Машины и технология обработки материалов давлением», 1-36 20 02 «Упаковочное производство (по направлениям)» днев. и заоч. форм обучения / П. П. Изотов. – Гомель : ГГТУ им. П. О. Сухого, 2008. – 60 с. – Систем. требования: PC не ниже Intel Celeron 300 МГц ; 32 Mb RAM ; свободное место на HDD 16 Mb ; Windows 98 и выше ; Adobe Acrobat Reader. – Режим доступа: <http://gstu.local/lib>. – Загл. с титул. экрана.

ISBN 978-985-420-735-3.

Содержит краткие теоретические сведения, примеры выполнения заданий, контрольные вопросы и литературу для самостоятельного изучения. В приложении приведен набор команд микропроцессора КР580, условные обозначения в наборе команд и коды команд.
Для студентов неэлектротехнических специальностей.

**УДК 004.315(075.8)
ББК 32.973.26-04я73**

ISBN 978-985-420-735-3

© Изотов П. П., 2008
© Учреждение образования «Гомельский
государственный технический университет
имени П. О. Сухого», 2008

ВВЕДЕНИЕ

Развитие цифровых микросхем с «жесткой» логикой привело к появлению микросхем с «программируемой» логикой, на базе которых строятся микропроцессорные системы. В настоящее время микропроцессорные системы применяются начиная от бытовой техники (телевизоры, видеомагнитофоны, электронные игрушки и т. д.) и заканчивая промышленными устройствами (станки с числовым программным управлением, промышленные роботы, гибкие перестраиваемые автоматические линии и т. д.). Столь широкое применение обусловлено тем, что некоторые микропроцессорные системы выполняются в виде интегральных микросхем. Такая микросхема обладает рядом достоинств перед цифровыми микросхемами с «жесткой» логикой:

- многофункциональность, когда изменением программы работы микропроцессора можно добиться реализации различных функций;
- надежность, т. е. вся система выполнена в виде одной микросхемы, за счет чего в устройстве уменьшается количество соединений между элементами и повышается надежность устройства;
- относительно низкая стоимость, так как микросхема многофункциональна, она востребована многими пользователями, а увеличение объема серийного производства уменьшает стоимость изготавливаемой микросхемы.

На рис. 1 представлена структурная схема обобщенной микропроцессорной системы. Микропроцессорная система состоит из микропроцессора, запоминающего устройства и устройств сопряжения микропроцессора с внешними устройствами ввода–вывода. Запоминающее устройство состоит из двух частей: постоянного (ПЗУ) и оперативного (ОЗУ) запоминающего устройства. ОЗУ позволяет микропроцессору читать и записывать данные и программы, но при выключении питания данные в ОЗУ стираются. ПЗУ позволяет микропроцессору только читать данные и программы, но при этом при выключении питания данные в ПЗУ не стираются. Поэтому программа работы микропроцессора всегда хранится в ПЗУ, а промежуточные результаты, полученные в ходе выполнения программы, сохраняются в ОЗУ. Устройства сопряжения состоят из портов ввода–вывода. Порты вывода позволяют микропроцессору записывать данные, а порты ввода читать данные из внешних устройств.

Несмотря на вышеуказанные достоинства, микропроцессорная система обладает и существенным недостатком, который заключается в том, что работа микропроцессорной системы всегда медленнее работы системы, выполненной на базе цифровых микросхем с «жесткой» логикой. Это обусловлено принципом работы микропроцессор-

ной системы. Микропроцессор для выполнения очередной команды должен вначале прочитать команду, а затем ее выполнить. И так для каждой команды. В цифровых микросхемах с «жесткой» логикой команда выполняется сразу, без этапа чтения команды.

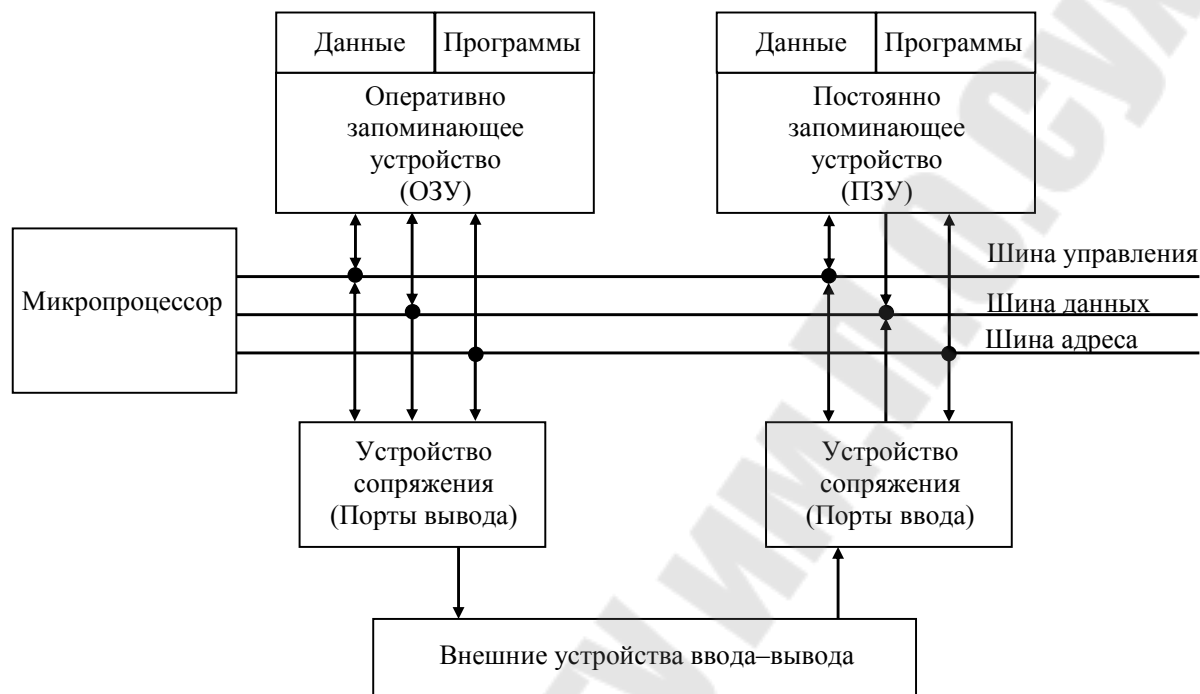


Рис. 1. Обобщенная структурная схема микропроцессорной системы

В лабораторных работах, представленных ниже, рассматривается микропроцессорная система, построенная на базе микропроцессора КР580ВМ80А. На рис. 2 представлена структурная схема микропроцессора КР580ВМ80А. Микропроцессор состоит из регистров данных, арифметическо-логического устройства (АЛУ), регистра признаков (флагов), блока управления и буферов.

Регистры данных. Для хранения участвующих в операциях данных предусмотрено семь 8-разрядных регистров. Регистр А, называемый аккумулятор, предназначен для обмена информацией с внешними устройствами (либо содержимое этого регистра может быть выдано на выход, либо со входа в него может быть принято число), при выполнении арифметических, логических операций и операций сдвига он служит источником операнда (числа, участвующего в операции), в него помещается результат выполнения операции.

Шесть других регистров, обозначенных В, С, D, E, H, L, образуют так называемый блок регистров общего назначения РОН (название связано с тем, что эти регистры могут использоваться для хранения как данных, так и адресов). Эти регистры могут использоваться как

одинокные 8-разрядные регистры. В случаях, когда возникает необходимость хранить 16-разрядные двоичные числа, они объединяются в пары BC, DE, HL.

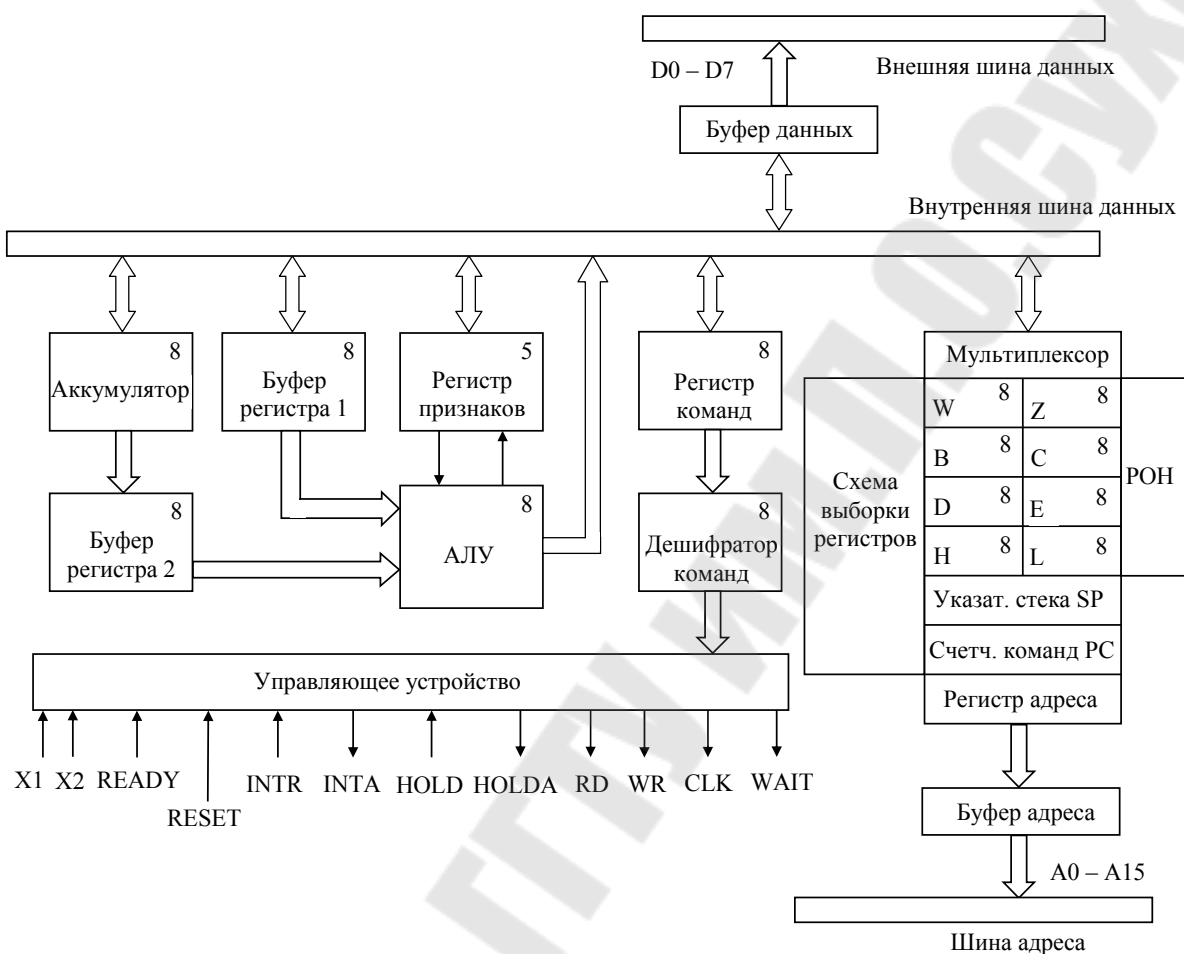


Рис. 2. Структурная схема микропроцессора KP580BM80A: АЛУ – арифметическо-логическое устройство; РОН – регистры общего назначения; X1, X2 – тактовые сигналы; READ – сигнал «Готовность»; RESET – (сброс) установка в исходное состояние; INTR – сигнал «Запрос прерывания»; INTA – сигнал «Разрешение прерывания»; HOLD – сигнал «Захват» (запрос от внешнего устройства на захват шины данных, шины адреса и шины управления); HOLDA – сигнал «Подтверждение захвата» (разрешение внешнему устройству на захват шины данных, шины адреса и шины управления); RD – сигнал «Прием информации»; WR – сигнал «Выдача информации»; 1CLK – сигнал синхронизации; WAIT – сигнал «Ожидание»

Регистры: буферный регистр 1, буферный регистр 2, W, Z используются буферные, программно-недоступные регистры (т. е. регистры, к которым программист при составлении программы не может обращаться).

Указатель стека SP (16-разрядный) служит для адресации особого вида памяти, называемого стеком.

Счетчик команд РС (16-разрядный) предназначен для хранения адреса команды; после выбора из оперативной памяти текущей команды содержимое счетчика увеличивается на единицу, и таким образом формируется адрес очередной команды (при отсутствии безусловных и условных переходов).

При обращении к памяти в качестве адреса используется содержимое регистровой пары HL блока РОН.

При выдаче адреса содержимое регистровой пары HL передается в 16-разрядный регистр адреса, из которого далее через буфер адреса поступает на 16-разрядную шину адреса. С этой шины адрес может быть принят в оперативную или постоянную память. Число кодовых комбинаций 16-разрядного адреса равно 2^{16} , каждая из этих кодовых комбинаций может определять адрес одной из ячеек памяти. Таким образом, обеспечивается возможность обращения к памяти, содержащей до $2^{16} = 2^6 \cdot 2^{10} = 64$ К 8-разрядных слов (байт).

Арифметическо-логическое устройство (АЛУ). В 8-разрядном АЛУ предусмотрена возможность выполнения четырех арифметических операций (сложение с передачей переноса в младший разряд и без учета этого переноса, вычитание с передачей заема в младший разряд и без него), четырех видов логических операций (конъюнкции – логического умножения И, дизъюнкции – логического сложения ИЛИ, сравнения), а также четырех видов циклического сдвига.

При выполнении арифметических и логических операций одним из операндов служит содержимое аккумулятора и результат выполненной операции помещается в аккумулятор. Циклический сдвиг выполняется только над содержимым аккумулятора.

Регистр признаков (флагов). Этот 5-разрядный регистр предназначен для хранения определенных признаков, выявляемых в числе, которое представляет собой результат выполнения некоторых операций. Пять признаков (флагов) этого регистра имеют следующее назначение:

D0 – флаг переноса (CY), устанавливается в состояние, соответствующее переносу из старшего разряда при выполнении арифметических операций и содержимое выдвигаемого из аккумулятора разряда при выполнении операции сдвига;

D2 – флаг четности (P), равен логической 1, если число единиц в разрядах результата четное;

D4 – флаг вспомогательного переноса (AC), хранит перенос, возникающий при выполнении операции из 4-го разряда;

D6 – флаг нуля (Z), устанавливается в состояние логической 1, если результат операции АЛУ или операции приращения содержимого регистра равен нулю;

D7 – флаг знака (S), устанавливается в состояние, соответствующее значению старшего разряда результата операции АЛУ или операции приращения содержимого регистра.

Буферы. Буферы данных и буферы адреса обеспечивают связь центрального процессора с внешними шинами данных и адреса. Особенность буферов состоит в том, что в каждом разряде они используют логические элементы с тремя состояниями. В них, кроме состояний логического нуля и логической 1, предусмотрено еще третье состояние, в котором они имеют практически бесконечное выходное сопротивление и оказываются отключенными от соответствующих шин. Использование таких буферов позволяет процессору отключаться от внешних шин (шин данных и адреса), предоставляя их в распоряжение внешних устройств, а также позволяет использовать одну и ту же шину данных как для приема данных (в качестве входной шины), так и для выдачи данных (в качестве выходной шины). Такое использование шины позволяет сократить число выводов микросхемы.

Блок управления. Состоит из регистра команд, куда принимается первый байт команды, и управляющего устройства, формирующего управляющие сигналы, под действием которых выполняются микрооперации в отдельных узлах. Управляющее устройство содержит выполненную на программируемой логической матрице управляющую память, в которой хранятся микропрограммы отдельных операций. Пользователь не может изменить содержимого управляющей памяти, а значит, и состава команд.

Управляющее устройство принимает и выдает следующие сигналы:

X1, X2 – тактовые сигналы, вырабатываемые внешним генератором и необходимые для работы микропроцессора;

RESET – сигнал «Сброс» в микропроцессоре производит сброс в нуль счетчика команд РС. И если в ячейке памяти с нулевым адресом хранится 1-я команда программы, то с этой команды начинается исполнение команды;

RD – сигнал «Прием информации» (логическая 1) обеспечивает передачу информации от шины данных к микропроцессору;

WR – сигнал «Выдача информации» (логический 0) обеспечивает передачу информации от микропроцессора к шине данных;

HOLD – сигнал «Захват» (логическая 1), запрос от внешнего устройства на использование шины данных, шины адреса и шины управления;

HOLDA – сигнал «Подтверждение захвата» (логическая 1), разрешение внешнему устройству на использование шины данных, шины адреса и шины управления;

READ – сигнал «Готовность» (логический 0) обеспечивает переход микропроцессора в режим ожидания. Если медленное внешнее устройство не успевает подготовить данные, то оно выдает сигнал «Готовность» (логический 0) и процессор переходит в режим ожидания. Когда данные подготовлены, внешнее устройство выдает сигнал «Готовность» (логическая 1), процессор продолжает работу;

WAIT – сигнал «Ожидание» (логическая 1) процессор в режиме ожидания, вызванного сигналом «Готовность» (логический 0);

CLK – сигнал синхронизации;

INTR – сигнал «Запрос прерывания» (логическая 1), прерывается выполнение текущей программы, и процессор переходит в режим обслуживания режима прерывания;

INTA – сигнал «Разрешение прерывания» (логическая 1), процессор сообщает внешним устройствам, что работает в режиме обслуживания прерывания.

ЛАБОРАТОРНАЯ РАБОТА № 1

Команды микропроцессора КР580

Цель работы: при помощи учебного микропроцессорного комплекта (УМК) ознакомиться с системой команд микропроцессора КР580ВМ80А.

Программа работы

1. Освоить навыки работы на учебном микропроцессорном комплекте (УМК).
2. Ознакомиться с командами арифметических и логических операций микропроцессора.
3. Ознакомиться с командами передачи данных.
4. Ознакомиться с командами передачи управления.

Теоретические сведения

Команды микропроцессора делятся на команды арифметических и логических операций, команды передачи данных и на команды передачи управления. Набор команд приведен в Приложении 1.

Команды арифметических операций позволяют осуществлять:

– сложение, вычитание содержимого заданного регистра (В, С, D, E, H, L), ячейки памяти или непосредственных данных из содержимого аккумулятора (А);

– инкрементирование (увеличение на 1), декрементирование (уменьшение на 1) содержимого заданного регистра или ячейки памяти.

Команды логических операций позволяют осуществлять:

– поразрядное логическое умножение И, сложение ИЛИ, исключающее ИЛИ содержимого заданного регистра (В, С, D, E, H, L), ячейки памяти или непосредственных данных с содержимым аккумулятора (А–регистр);

– сравнение содержимого заданного регистра (В, С, D, E, H, L), ячейки памяти или непосредственных данных с содержимым аккумулятора А–регистр;

– циклический сдвиг вправо (деление на 2), влево (умножение на 2) содержимого аккумулятора (А–регистра);

Команды передачи данных позволяют осуществлять:

– пересылку данных из регистра в регистр;

– пересылку данных из ячейки памяти в регистр и наоборот;

– пересылку непосредственных данных в регистр, в ячейку памяти.

Команды передачи управления позволяют осуществлять:

– безусловный переход;

– переход по признаку (если перенос, если не перенос, если нуль, если не нуль, если плюс, если минус, если четно, если нечетно);

- безусловный вызов подпрограммы;
- вызов подпрограммы по признаку;
- возврат из подпрограммы;
- возврат из подпрограммы по признаку.

Примеры применения команд арифметических операций на УМК

Пример 1. Сложение числа 03h (буква h обозначает, что число представлено в шестнадцатеричном коде), находящегося в регистре D, и числа 02h, находящегося в аккумуляторе (A-регистр). (Сумма находится в аккумуляторе (A-регистр).)

Таблица 1

Текст программы

Адрес ячейки памяти	Код	Метка	Мnemonic	Комментарии
0800:	82		ADD D;	Сложение содержимого аккумулятора с регистром D
0801:	00		NOP;	Пустая операция

До выполнения программы:

в аккумуляторе число 02h → 0000 0010b (буква b – в двоичном коде);
в регистре D число 03h → 0000 0011b.

Выполнение команды ADD D:

$$\begin{array}{r}
 0000\ 0010 \\
 +\ 0000\ 0011 \\
 \hline
 0000\ 0101
 \end{array}$$

Результат сложения заносится в аккумулятор 0000 0101b → 05h.

После выполнения программы:

в регистре D число 03h → 0000 0011b;
в аккумуляторе число 05h → 0000 0101b.

Этапы выполнения программы на УМК:

Загрузка чисел: 03 в регистр D, 02 в аккумулятор:

РГ D 03 _ A 02 ВП.

Загрузка программы:

П 0800 ВП

82 _ 00 ВП

Запуск программы: СТ 0800 _ 0801 ВП

Просмотр результата сложения: РГ А

Пример 2. Из числа 06h, находящегося в аккумуляторе, вычитаем число 05h, находящееся в ячейке памяти с адресом 080Ah. (Разность находится в аккумуляторе (A-регистр).)

Таблица 2

Текст программы

Адрес ячейки памяти	Код	Метка	Мнемокод	Комментарии
0800:	96		SUB M;	Из содержимого аккумулятора вычитается число, находящееся в ячейке памяти
0801:	00		NOP;	Пустая операция

Примечание 1. Адрес ячейки памяти должен находиться в регистрах H и L. В регистре H старший байт адреса ячейки памяти, в регистре L младший байт адреса ячейки памяти.

До выполнения программы:
 в аккумуляторе число 06h → 0000 0110b;
 в регистре H число 08h (старший байт адреса ячейки памяти);
 в регистре L число 0Ah (младший байт адреса ячейки памяти);
 в ячейке памяти с адресом 080Ah число 05h → 0000 0101b.

Выполнение команды SUB M:

```

    0000 0110
  - 0000 0101
  -----
    0000 0001
    
```

Разность заносится в аккумулятор 0000 0001b → 01h.

После выполнения программы:
 в аккумуляторе число 01h → 0000 0001b;
 в регистре H число 08h (старший байт адреса ячейки памяти);
 в регистре L число 0Ah (младший байт адреса ячейки памяти);
 в ячейке памяти с адресом 080Ah число 05h → 0000 0101b.

Этапы выполнения программы на УМК:

Загрузка чисел: 08 в регистр H, 0A в регистр L, 06 в аккумулятор:

РГ H 08 _ L 0A _ A 06 ВП

Загрузка числа 05 в ячейку памяти с адресом 080A:

П 080A ВП

05 ВП

Загрузка программы:

П 0800 ВП

96 _ 00 ВП

Запуск программы: СТ 0800 _ 0801 ВП

Просмотр результата: РГ А

Пример 3. Сложение числа А1h (числу А1h соответствует число 161 в десятичном коде), находящегося в аккумуляторе, с непосредственным данным числом 82h (130). Сумма имеет вид трехбайтового числа 123h (291). Поэтому два младших байта 23h находятся в аккумуляторе, а 1 находится в младшем бите D0 регистра признаков (F-регистр).

Примечание 2. Для перевода числа в шестнадцатеричном коде в десятичный код его преобразуют в двоичный код, а затем в десятичный код по формуле:

$$N = a_0 \cdot 2^0 + a_1 \cdot 2^1 + a_2 \cdot 2^2 + a_3 \cdot 2^3 + a_4 \cdot 2^4 + a_5 \cdot 2^5 + a_6 \cdot 2^6 + a_7 \cdot 2^7 + a_8 \cdot 2^8 = \\ = a_0 \cdot 1 + a_1 \cdot 2 + a_2 \cdot 4 + a_3 \cdot 8 + a_4 \cdot 16 + a_5 \cdot 32 + a_6 \cdot 64 + a_7 \cdot 128 + a_8 \cdot 256.$$

Например:

A1h → 1010 0001b → a7a6a5a4 a3a2a1a0

A1h → 1·1 + 0·2 + 0·4 + 0·8 + 0·16 + 1·32 + 0·64 + 1·128 + 0·256 = 161

82h → 1000 0010b → 0·1 + 1·2 + 0·4 + 0·8 + 0·16 + 0·32 + 0·64 + 1·128 + 0·256 = 130

123h → 1 0010 0011b → 1·1 + 1·2 + 0·4 + 0·8 + 0·16 + 1·32 + 0·64 + 0·128 + 1·256 = 291

Таблица 3

Текст программы

Адрес ячейки памяти	Код	Метка	Мнемокод	Комментарии
0800: 0801:	C6 82		ADI 82h;	Сложение содержимого аккумулятора с числом 82h
0802:	00		NOP;	Пустая операция

До выполнения программы:

в аккумуляторе число А1h → 1010 0001b;

в регистре F число 00h.

Выполнение команды ADI 82h:

$$\begin{array}{r} 1010\ 0001 \\ + 1000\ 0010 \\ \hline 1\ 0010\ 0011 \end{array}$$

Результат сложения заносится:

два младших байта в аккумулятор 0010 0011b → 23h;

1 старшего байта в младший бит D0 F-регистра 0000 0011b → 03h.

После выполнения программы:

в регистре F число 03h → 0000 0011b;

в аккумуляторе число 23h → 0010 0011b.

Примечание 3. F–регистр является регистром признаков (флагов) и состоит из восьми битов D0–D7. Назначение этих битов следующее:

D0 – флаг переноса (CY), равен 1, если при выполнении операции был перенос из седьмого разряда арифметическо-логического устройства (АЛУ) или заем в этот разряд;

D1 = 1;

D2 – флаг четности (P), равен 1, если число единиц результата операции четное;

D3 = 0;

D4 – флаг вспомогательного переноса (AC), равен 1, если при выполнении операции был перенос из третьего разряда АЛУ в четвертый;

D5 = 0;

D6 – флаг нуля (Z), равен 1, если результат операции равен нулю;

D7 – флаг знака (S), равен 1, если седьмой бит результата равен 1.

Этапы выполнения программы на УМК:

Загрузка чисел: 00 в регистр F, A1 в аккумулятор:

РГ F 00 _ A A1 ВП

Загрузка программы:

П 0800 ВП

С6 _ 82 _ 00 ВП

Запуск программы:

СТ 0800 _ 0802 ВП

Просмотр результата сложения: РГ F _ A

Пример 4. Увеличить содержимое регистра С на 1. (Результат находится в регистре С.)

Таблица 4

Текст программы

Адрес ячейки памяти	Код	Метка	Мнемокод	Комментарии
0800:	0С		INR C;	Инкрементирование содержимого регистра С
0801:	00		NOP;	Пустая операция

До выполнения программы:

в регистре С число 03h → 0000 0011b.

Выполнение команды INR C:

0000 0011

+ 0000 0001

0000 0100

Результат сложения заносится в регистр С 0000 0100b → 04h.

После выполнения программы:

в регистре С число 04h → 0000 0100b.

Этапы выполнения программы на УМК:

Загрузка числа 03 в регистр С: РГ С 03 ВП

Загрузка программы:

П 0800 ВП

0С _ 00 ВП

Запуск программы: СТ 0800 _ 0801 ВП

Просмотр результата: РГ С

Пример 5. Уменьшить содержимое ячейки памяти с адресом 0802h на 1. (Результат находится в ячейке памяти с адресом 0802h.)

Таблица 5

Текст программы

Адрес ячейки памяти	Код	Метка	Мнемокод	Комментарии
0800:	35		DCR M;	Декрементирование содержимого ячейки памяти
0801:	00		NOP;	Пустая операция

До выполнения программы:

в регистре Н число 08h;

в регистре L число 02h;

в ячейке памяти с адресом 0802h число 05h → 0000 0101b.

Выполнение команды DCR M:

 0000 0101

 0000 0001

0000 0100

Разность заносится в ячейку памяти с адресом 0802h 0000 0100b → 04h.

После выполнения программы:

в регистре Н число 08h;

в регистре L число 02h;

в ячейке памяти с адресом 0802h число 04h → 0000 0100b.

Этапы выполнения программы на УМК:

Загрузка чисел: 08 в регистр Н, 02 в регистр L:

РГ Н 08 _ L 02 ВП

Загрузка числа 05 в ячейку памяти с адресом 0802:

П 0802 ВП

05 ВП

Загрузка программы:

П 0800 ВП

35 _ 00 ВП

Запуск программы: СТ 0800 _ 0801 ВП

Просмотр результата: П 0802 ВП

Примеры применения команд логических операций на УМК

Пример 6. Осуществить поразрядное И – логическое умножение числа 05h, находящегося в аккумуляторе, и числа 03h, находящегося в регистре В. (Результат находится в аккумуляторе.)

Таблица 6

Текст программы

Адрес ячейки памяти	Код	Метка	Мнемокод	Комментарии
0800:	A0		ANA B;	Поразрядное И над содержимым аккумулятора и регистра В
0801:	00		NOP;	Пустая операция

До выполнения программы:

в аккумуляторе число 05h → 0000 0101b;

в регистре В число 03h → 0000 0011b.

Выполнение команды ANA B:

$$\begin{array}{r} 0000\ 0101 \\ \& \\ 0000\ 0011 \\ \hline 0000\ 0001 \end{array}$$

Результат заносится в аккумулятор 0000 0001b → 01h.

После выполнения программы:

в регистре В число 03h → 0000 0011b;

в аккумуляторе число 01h → 0000 0001b.

Этапы выполнения программы на УМК:

Загрузка чисел: 03 в регистр В, 05 в аккумулятор:

РГ В 03 _ А 05 ВП

Загрузка программы:

П 0800 ВП

A0 _ 00 ВП

Запуск программы: СТ 0800 _ 0801 ВП

Просмотр результата: РГ А

Пример 7. Осуществить поразрядное ИЛИ – логическое сложение числа 05h, находящегося в аккумуляторе, и числа 03h, которое является непосредственным данным. (Результат находится в аккумуляторе.)

Таблица 7

Текст программы

Адрес ячейки памяти	Код	Метка	Мнемокод	Комментарии
0800: 0801:	F6 03		ORI 03h;	Поразрядное ИЛИ над содержимым аккумулятора и числом 03h
0802:	00		NOP;	Пустая операция

До выполнения программы:

в аккумуляторе число 05h → 0000 0101b.

Выполнение команды ORI 03h:

$$\begin{array}{r} 0000\ 0101 \\ \underline{1\ 0000\ 0011} \\ 0000\ 0111 \end{array}$$

Результат заносится в аккумулятор 0000 0111b → 07h.

После выполнения программы:

в аккумуляторе число 07h → 0000 0111b.

Этапы выполнения программы на УМК:

Загрузка числа 05 в аккумулятор: РГ А 05 ВП

Загрузка программы:

П 0800 ВП

F6 _ 03 _ 00 ВП

Запуск программы: СТ 0800 _ 0802 ВП

Просмотр результата: РГ А

Пример 8. Осуществить поразрядное логическое исключающее ИЛИ числа 05h, находящегося в аккумуляторе, и числа 03h, находящегося в ячейке памяти с адресом 0802h. (Результат находится в аккумуляторе.)

Таблица 8

Текст программы

Адрес ячейки памяти	Код	Метка	Мнемокод	Комментарии
0800:	AE		XRA M;	Поразрядное исключающее ИЛИ над содержимым аккумулятора и ячейкой памяти
0801:	00		NOP;	Пустая операция

До выполнения программы:

в регистре Н число 08h;

в регистре L число 02h;
 в ячейке памяти с адресом 0802h число 03h → 0000 0011b;
 в аккумуляторе число 05h → 0000 0101b.

Выполнение команды XRA M:

$$\begin{array}{r}
 =1 \quad 0000 \ 0101 \\
 \quad \quad 0000 \ 0011 \\
 \hline
 \quad \quad 0000 \ 0110
 \end{array}$$

Результат заносится в аккумулятор 0000 0110b → 06h.

После выполнения программы:

в регистре H число 08h;
 в регистре L число 02h;
 в ячейке памяти с адресом 0802h число 03h → 0000 0011b;
 в аккумуляторе число 06h → 0000 0110b.

Этапы выполнения программы на УМК:

Загрузка чисел: 08 в регистр H, 02 в регистр L:

РГ H 08 _ L 02 ВП

Загрузка числа 03 в ячейку памяти с адресом 0802:

П 0802 ВП

03 ВП

Загрузка программы:

П 0800 ВП

AE _ 00 ВП

Запуск программы: СТ 0800 _ 0801 ВП

Просмотр результата: РГ А

Пример 9. Сравнить содержимое аккумулятора 05h с содержимым регистра E 05h. Содержимое аккумулятора не меняется, а результат сравнения находится в F-регистре.

Таблица 9

Текст программы

Адрес ячейки памяти	Код	Метка	Мnemonic	Комментарии
0800:	BB		CMR E;	Сравнение содержимого аккумулятора и регистра E
0801:	00		NOP;	Пустая операция

Примечание 4. При сравнении:

- если содержимое аккумулятора больше содержимого регистра (B, C, D, E, H, L), ячейки памяти или непосредственных данных, то в F-регистре D0 = 0 → флаг CY (переполнение) и D6 = 0 → флаг Z (нуля);

- если равно, то $D0 = 0$ и $D6 = 1$;
- если меньше, то $D0 = 1$ и $D6 = 0$;
- выставляются в F–регистре флаги: $D2 \rightarrow$ флаг P (четности); $D4 \rightarrow$ флаг AC (вспомогательный перенос); $D7 \rightarrow$ флаг S (знака);
- в F–регистре не меняют значения разряды: $D1 = 1$, $D3 = 0$, $D5 = 0$;
- содержимое аккумулятора не изменяется.

До выполнения программы:

в аккумуляторе число $05h \rightarrow 0000\ 0101b$;

в регистре E число $05h \rightarrow 0000\ 0101b$;

в регистре F число $00h \rightarrow 0000\ 0000b$.

Выполнение команды CMP E:

$05h = 05h$

Результат заносится в F–регистр $0101\ 0110b \rightarrow 56h$ ($D6=1$).

После выполнения программы:

в аккумуляторе число $05h \rightarrow 0000\ 0101b$;

в регистре E число $05h \rightarrow 0000\ 0101b$;

в регистре F число $56h \rightarrow 0101\ 0110b$.

Этапы выполнения программы на УМК:

Загрузка чисел: 05 в аккумулятор, 05 в регистр E, 00 в регистр F:

РГ A 05 _ E 05 _ F 00 ВП.

Загрузка программы:

П 0800 ВП

ВВ _ 00 ВП

Запуск программы: СТ 0800 _ 0801 ВП

Просмотр результата сравнения: РГ F

Пример 10. Умножить содержимое аккумулятора $04h$ на 2. Результат находится в аккумуляторе.

Примечание 5. Если в выражении

$$04h \rightarrow 0000\ 0100b \rightarrow 0 \cdot 1 + 0 \cdot 2 + 1 \cdot 4 + 0 \cdot 8 + 0 \cdot 16 + 0 \cdot 32 + 0 \cdot 64 + 0 \cdot 128 + 0 \cdot 256 = 4,$$

сдвинуть 1 влево, то произойдет умножение на 2:

$$08h \rightarrow 0000\ 1000b \rightarrow 0 \cdot 1 + 0 \cdot 2 + 0 \cdot 4 + 1 \cdot 8 + 0 \cdot 16 + 0 \cdot 32 + 0 \cdot 64 + 0 \cdot 128 + 0 \cdot 256 = 8.$$

Если сдвинуть вправо, то произойдет деление на 2:

$$02h \rightarrow 0000\ 0010b \rightarrow 0 \cdot 1 + 1 \cdot 2 + 0 \cdot 4 + 0 \cdot 8 + 0 \cdot 16 + 0 \cdot 32 + 0 \cdot 64 + 0 \cdot 128 + 0 \cdot 256 = 2.$$

Таблица 10

Текст программы

Адрес ячейки памяти	Код	Метка	Мнемокод	Комментарии
0800:	07		RLC;	Циклический сдвиг влево содержимого аккумулятора
0801:	00		NOP;	Пустая операция

До выполнения программы:
в аккумуляторе число 04h → 0000 0100b.

Выполнение команды RLC:

$$\begin{array}{r} 0000\ 0100 \\ \hline 0000\ 1000 \end{array}$$

Результат остается в аккумуляторе 0000 1000b → 08h.

После выполнения программы:
в аккумуляторе число 08h → 0000 1000b.

Этапы выполнения программы на УМК:

Загрузка числа 04 в аккумулятор: РГ А 04 ВП

Загрузка программы:

П 0800 ВП

07 _ 00 ВП

Запуск программы: СТ 0800 _ 0801 ВП

Просмотр результата: РГ А

Пример 11. Деление содержимого аккумулятора 04h на 2. (Результат находится в аккумуляторе).

Таблица 11

Текст программы

Адрес ячейки памяти	Код	Метка	Мнемокод	Комментарии
0800:	0F		RRC;	Циклический сдвиг вправо содержимого аккумулятора
0801:	00		NOP;	Пустая операция

До выполнения программы:
в аккумуляторе число 04h → 0000 0100b.

Выполнение команды RRC:

$$\begin{array}{r} 0000\ 0100 \\ \hline 0000\ 0010 \end{array}$$

Результат остается в аккумуляторе 0000 0010b → 02h.

После выполнения программы:
в аккумуляторе число 02h → 0000 0010b.

Этапы выполнения программы на УМК:

Загрузка числа 04 в аккумулятор: РГ А 04 ВП

Загрузка программы:

П 0800 ВП

0F _ 00 ВП

Запуск программы: СТ 0800 _ 0801 ВП

Просмотр результата: РГ А

Примеры применения команд передачи данных на УМК

Пример 12. Для выполнения примера 2 необходимо было загрузить в аккумулятор, в регистры H, L и ячейку памяти данные. Составить программу, чтобы эти операции выполнял микропроцессор.

Таблица 12

Текст программы

Адрес ячейки памяти	Код	Метка	Мнемокод	Комментарии
0800: 0801:	3E 06		MVI A, 06h;	Пересылка числа 06h в аккумулятор
0802: 0803:	26 08		MVI H, 08h;	Пересылка числа 08h в регистр H
0804: 0805:	2E 0A		MVI L, 0Ah;	Пересылка числа 0Ah в регистр L
0806: 0807:	36 05		MVI M, 05h;	Пересылка числа 05h в ячейку памяти
0808:	96		SUB M;	Из содержимого аккумулятора вычитается число, находящееся в ячейки памяти
0809:	00		NOP;	Пустая операция

До выполнения программы:

в аккумуляторе любое число;
в регистре H любое число;
в регистре L любое число;
в ячейке памяти с адресом 080Ah любое число.

После выполнения программы:

в аккумуляторе число 01h → 0000 0001b;
в регистре H число 08h;
в регистре L число 02h;
в ячейке памяти с адресом 080Ah число 05h → 0000 0101b.

Этапы выполнения программы на УМК:

Загрузка программы:

П 0800 ВП

3E _ 06 _ 26 _ 08 _ 2E _ 0A _ 36 _ 05 _ 96 _ 00 ВП

Запуск программы: СТ 0800 _ 0809 ВП

Просмотр результата: РГ А

Пример 13. Записать в регистр В число 05h, а затем из регистра В число переписать в регистр С.

Таблица 13

Текст программы

Адрес ячейки памяти	Код	Метка	Мнемокод	Комментарии
0800: 0801:	06 05		MVI B, 05h;	Пересылка числа 05h в регистр В
0802:	48		MOV C, B;	Пересылка числа 05h из регистра В в регистр С
0803:	00		NOP;	Пустая операция

До выполнения программы:

в регистре В любое число;
в регистре С любое число.

После выполнения программы:

в регистре В число 05h;
в регистре С число 05h.

Этапы выполнения программы на УМК:

Загрузка программы:

П 0800 ВП

06 _ 05 _ 48 _ 00 ВП

Запуск программы: СТ 0800 _ 0803 ВП

Просмотр результата: РГ В _ С

Примеры применения команд передачи управления на УМК

Пример 14. Просуммировать три числа 02h, 03h, 04h, находящиеся в ячейках памяти с адресами 0810h, 0811h, 0812h. (Результат суммирования находится в аккумуляторе.)

Таблица 14

Текст программы

Адрес ячейки памяти	Код	Метка	Мнемокод	Комментарии
Начальная установка числовых значений в необходимые регистры				
0800: 0801:	06 03		MVI B, 03h;	Пересылка числа 03h в регистр В (Число 03h – количество сложений)
0802: 0803:	26 08		MVI H, 08h;	Пересылка числа 08h в регистр H

Адрес ячейки памяти	Код	Метка	Мнемокод	Комментарии
0804: 0805:	2E 10		MVI L, 10h;	Пересылка числа 10h в регистр L
0806: 0807:	3E 00		MVI A, 00h;	Пересылка числа 00h в аккумулятор (Обнуление содержимого А-регистра)
0808	00		NOP;	Пустая операция
Сложение чисел				
0809:	86	M1:	ADD M;	Сложение содержимого ячейки памяти и аккумулятора
080A:	23		INX H;	Увеличение на 1 содержимого регистровой пары HL
080B:	05		DCR B;	Уменьшение на 1 содержимого регистра B
080C: 080D: 080E:	C2 09 08		JNZ M1;	Если содержимое регистра B не равно нулю, то перейти на метку M1. Если равно, то продолжить программу
080F:	00		NOP;	Пустая операция

Примечание б. При записи кода команд передачи управления вначале записывается код команды (C2), затем младший байт адреса метки (09), а потом старший байт адреса метки (08).

До выполнения программы:

в аккумуляторе любое число;
в регистре H любое число;
в регистре L любое число;
в регистре B любое число;
в ячейке памяти с адресом 0810h число 02h;
в ячейке памяти с адресом 0811h число 03h;
в ячейке памяти с адресом 0812h число 04h.

После выполнения программы:

в аккумуляторе число 09h;
в регистре H число 08h;
в регистре L число 12h;
в регистре B число 00h;
в ячейке памяти с адресом 0810h число 02h;
в ячейке памяти с адресом 0811h число 03h;
в ячейке памяти с адресом 0812h число 04h.

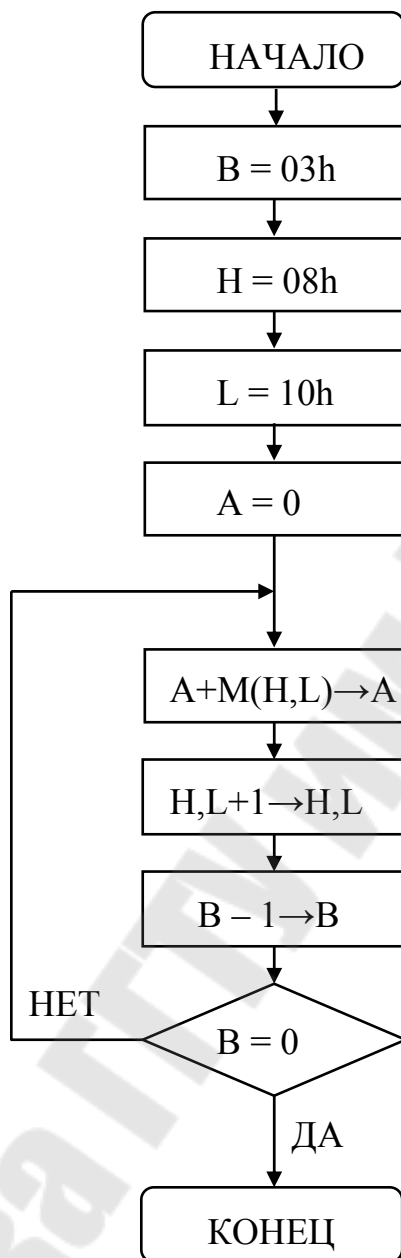


Рис. 3. Алгоритм работы программы

Этапы выполнения программы на УМК:

Загрузка чисел 02, 03, 04 в ячейки памяти с адресами 0810, 0811, 0812:

П 0810 ВП

02 _ 03 _ 04 ВП

Загрузка программы:

П 0800 ВП

06 _ 03 _ 26 _ 08 _ 2E _ 10 _ 3E _ 00 _ 00 _ 86 _ 23 _ 05 _ C2 _ 09 _ 08 _ 00 ВП

Запуск программы: СТ 0800 _ 080F ВП

Просмотр результата: РГ А

Пример 15. Найти в программе Монитора (с адреса 0000h по адрес 07FFh) код команды 0Bh (DCX B) и определить адрес ячейки памяти. Если адрес ячейки памяти более 07FFh, то записать в регистр В число 55h и остановить поиск. (Если в регистре В число 00h, то адрес ячейки памяти, в которой находится код 0Bh, располагается в регистре Н старший байт адреса, а в регистре L младший байт адреса.)

Таблица 15

Текст программы

Адрес ячейки памяти	Код	Метка	Мнемокод	Комментарии
Запись в регистровую пару HL числа FFFFh (- 1)				
0800:	26		MVI H, FFh;	Пересылка числа FFh в регистр H
0801:	FF			
0802:	2E		MVI L, FFh;	Пересылка числа FFh в регистр L
0803:	FF			
Увеличение содержимого регистровой пары HL на 1. Проверка, что регистровая пара HL содержит число 0800h (граница поиска)				
0804:	23	M1:	INX H;	Увеличение на 1 содержимого регистровой пары HL
0805:	3E		MVI A, 08h	Пересылка числа 08h в аккумулятор
0805:	08			
0807:	BC		CMR H;	Сравнение содержимого аккумулятора и содержимого регистра H
0808:	CA		JZ M3;	Если содержимое регистра H равно 08h, то на метку M3. Если нет, то продолжить программу
0809:	14			
080A:	08			
Проверка, что в ячейке памяти число 0Bh				
080B:	3E		MVI A, 0Bh;	Пересылка числа 0Bh в аккумулятор
080C:	0B			
080D:	BE		CMP M;	Сравнение содержимого аккумулятора и ячейки памяти
080E:	C2		JNZ M1;	Если содержимое ячейки памяти не равно 0Bh, то перейти на метку M1.
080F:	04			Если равно, то закончить программу
0810:	08			
Если содержимое ячейки памяти 0Bh, то записать в регистр В число 00h и закончить программу				
0811:	06		MVI B, 00h;	Запись числа 00h в регистр В
0812:	00			
0813:	00	M2:	NOP;	Пустая операция (Конец программы)

Адрес ячейки памяти	Код	Метка	Мнемокод	Комментарии
Если содержимое регистровой пары 0800h, то записать в регистр В число 55h и закончить программу				
0814:	06	M3:	MVI B, 55h;	Запись числа 55h в регистр В
0815:	55			
0816:	C3		JMP M2;	Безусловный переход на метку M2 (Конец программы)
0817:	13			
0818:	08			

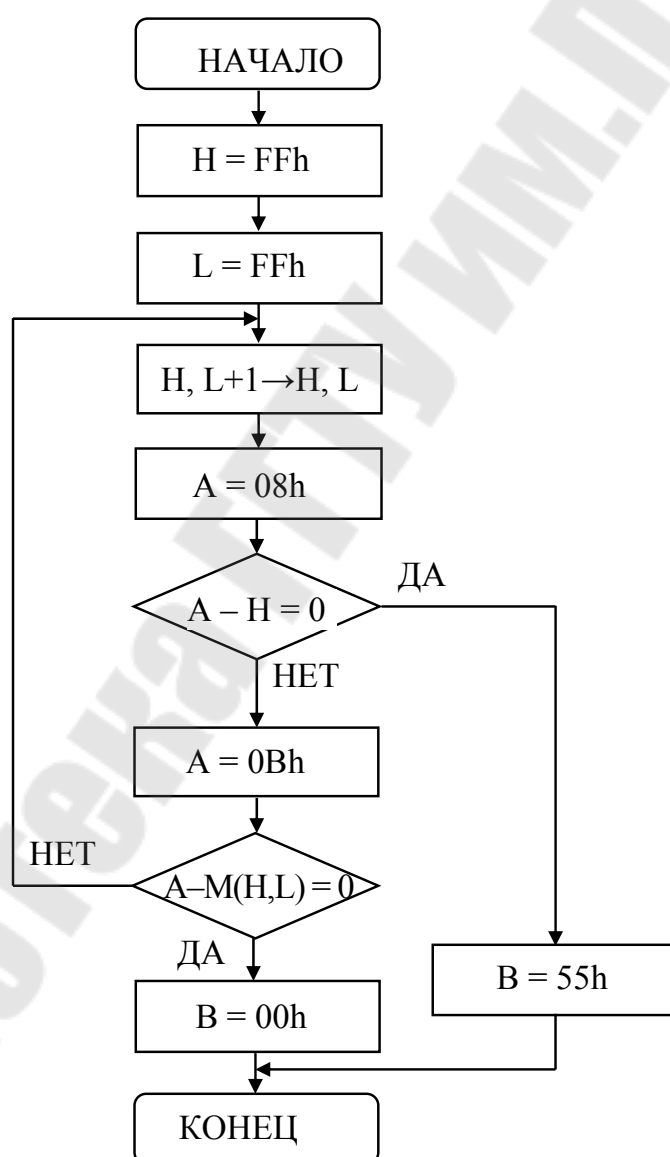


Рис. 4. Алгоритм работы программы

До выполнения программы:
 в аккумуляторе любое число;
 в регистре H любое число;
 в регистре L любое число;
 в регистре B любое число.

После выполнения программы:
 Если код 0Bh найден:
 в регистре B число 00h;
 в регистре H число 01h;
 в регистре L число DDh;
 в аккумуляторе число 0Bh.
 Если код 0Bh не найден:
 в регистре B число 55h;
 в регистре H число 08h;
 в регистре L число 00h;
 в аккумуляторе число 08h.

Этапы выполнения программы на УМК:

Загрузка программы:

П 0800 ВП

26 FF 2E FF 23 3E 08 BC CA 14 08 3E 0B BE _
 C2 06 08 06 00 00 06 55 C3 13 08 ВП

Запуск программы: СТ 0800 0813 ВП

Просмотр результата: РГ В H L ВП

Примечание 7. Чтобы увидеть, как работает программа в примере 15, когда код не найден, измените код по адресу 080Eh следующим образом: 080C: 8F.

Пример 16. Из чисел 05h, 07h, 06h, расположенных в ячейках памяти с адресами 0820h, 0821h, 0822h, найти максимальное число и записать его в регистр E.

Таблица 16

Текст программы

Адрес ячейки памяти	Код	Метка	Мнемокод	Комментарии
Начальная установка числовых значений в необходимые регистры				
0800:	06		MVI B, 02h	Запись числа 02h в регистр B
0801:	02			
0802:	26		MVI H, 08h;	Пересылка числа 08h в регистр H
0803:	08			
0804:	2E		MVI L, 20h;	Пересылка числа 20h в регистр L
0805:	20			

Адрес ячейки памяти	Код	Метка	Мнемокод	Комментарии
Запись в аккумулятор числа, находящегося в ячейке памяти с адресом 0820h				
0806:	7E		MOV A, M;	Запись в аккумулятор числа из памяти
Увеличение содержимого регистровой пары HL на 1. Сравнение содержимого аккумулятора и ячейки памяти				
0807:	23	M1:	INX H;	Увеличение на 1 содержимого регистровой пары HL
0808:	BE		CMR M;	Сравнение содержимого аккумулятора и ячейки памяти
0809: 080A: 080B	DA 11 08		JC M3;	Если содержимое аккумулятора меньше содержимого ячейки памяти, то на M3. Если больше – продолжать программу
080C:	05	M2:	DCR B;	Уменьшение на 1 содержимого регистра B
080D: 080E: 080F:	C2 07 08		JNZ M1;	Если содержимое регистра B не равно нулю, то перейти на метку M1. Если равно, то продолжить программу
0810:	00		NOP;	Пустая операция (Конец программы)
Если содержимое аккумулятора меньше содержимого ячейки памяти, то большее число записать в аккумулятор, а затем в регистр E				
0811:	7E	M3:	MOV A, M;	Запись в аккумулятор числа из памяти
0812:	5E		MOV E, M;	Запись в регистр E числа из памяти
0813: 0814: 0815:	C3 0C 08		JMP M2;	Безусловный переход на метку M2

До выполнения программы:

- в регистре B любое число;
- в регистре H любое число;
- в регистре L любое число;
- в аккумуляторе любое число;
- в регистре E любое число;
- в ячейке памяти с адресом 0820h число 05h;
- в ячейке памяти с адресом 0821h число 07h;
- в ячейке памяти с адресом 0822h число 06h.

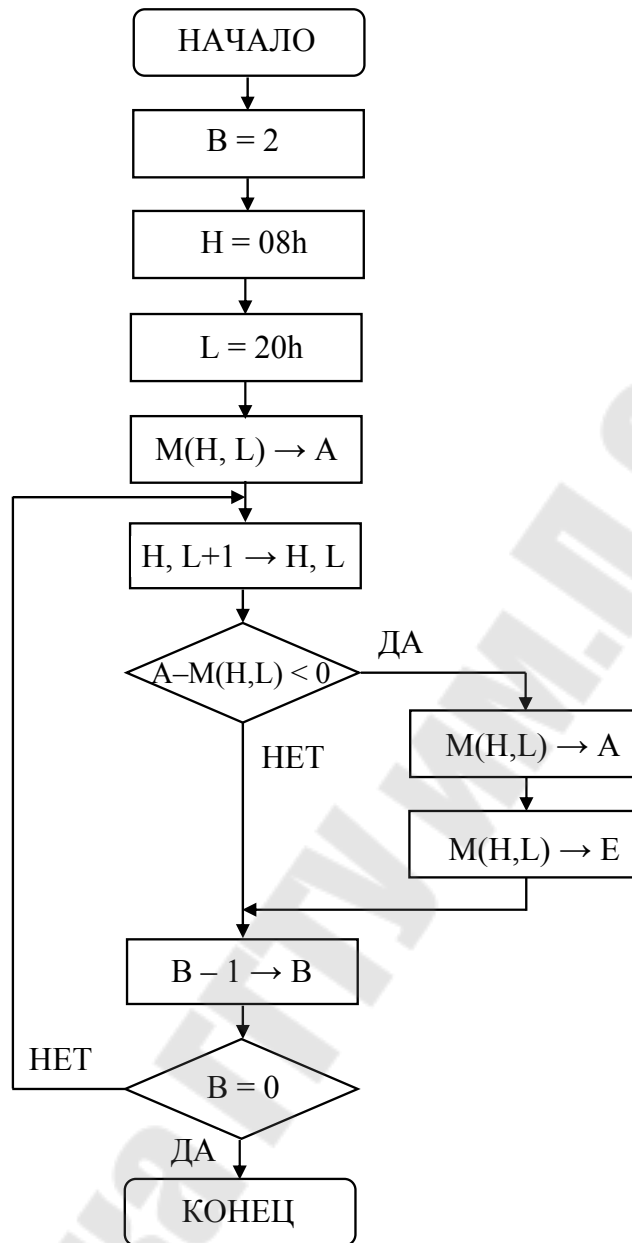


Рис. 5. Алгоритм работы программы

После выполнения программы:

- в регистре В число 00h;
- в регистре Н число 08h;
- в регистре L число 22h;
- в аккумуляторе число 07h;
- в регистре E число 07h;
- в ячейке памяти с адресом 0820h число 05h;
- в ячейке памяти с адресом 0821h число 07h;
- в ячейке памяти с адресом 0822h число 06h.

Этапы выполнения программы на УМК:

Загрузка чисел 05, 07, 06 в ячейки памяти с адресами 0820, 0821, 0822:

П 0820 ВП 05 _ 07 _ 06 ВП

Загрузка программы:

П 0800 ВП 06 _ 02 _ 26 _ 08 _ 2E _ 20 _ 7E _ 23 _ BE _ DA _ 11 _ 08 _ 05 _
C2 _ 07 _ 08 _ 00 _ 7E _ 5E _ C3 _ 0C _ 08 ВП

Запуск программы: СТ 0800 _ 0810 ВП

Просмотр результата: РГ E

Порядок выполнения работы

1. Используя примеры 1–5, ознакомиться с арифметическими командами микропроцессора и научиться составлять программы для микропроцессора с использованием этих команд, а также загружать и выполнять программы на УМК.

2. В соответствии с вариантом задания составить программу для микропроцессора. Загрузить и проверить составленную программу на УМК. При этом должно быть показано состояние регистров и ячеек памяти, используемых в программе, до и после выполнения программы, а также выполнение арифметической команды (примеры 1–5). При составлении программы брать коды команд из Приложения 1.

Вариант 1. Сложить число 04h, находящееся в ячейке памяти с адресом 0810h, с число 05h, находящееся в аккумуляторе.

Вариант 2. Из числа 09h, находящееся в аккумуляторе, вычесть число 05h (непосредственное данное).

Вариант 3. Увеличить на 1 число 06h, находящееся в ячейке памяти с адресом 0810h.

Вариант 4. Уменьшить на 1 число 08h, находящееся в регистре В.

Вариант 5. Из числа 08h, находящееся в аккумуляторе, вычесть число 05h, находящееся в регистре В.

Вариант 6. Сложить число 01h (непосредственное данное) с число 03h, находящееся в аккумуляторе.

Вариант 7. Увеличить на 1 число 07h, находящееся в регистре D.

Вариант 8. Уменьшить на 1 число 09h, находящееся в ячейке памяти с адресом 0812h.

Вариант 9. Сложить число 04h, находящееся в регистре E, с число 03h, находящееся в аккумуляторе.

Вариант 10. Из числа 07h, находящееся в аккумуляторе, вычесть число 05h, находящееся в ячейке памяти с адресом 0815h.

3. Используя примеры 6–11, ознакомиться с логическими командами микропроцессора и научиться составлять программы для микропроцессора с использованием логических команд.

4. В соответствии с вариантом задания составить программу для микропроцессора. Загрузить и проверить составленную программу на УМК.

Вариант 1. Осуществить поразрядное И – логическое умножение числа 09h, находящегося в аккумуляторе, и числа 03h, находящегося в ячейке памяти с адресом 0810h.

Вариант 2. Осуществить поразрядное ИЛИ – логическое сложение числа 04h, находящегося в аккумуляторе, и числа 03h, находящегося в регистре E.

Вариант 3. Осуществить поразрядное исключающее ИЛИ числа 06h, находящегося в аккумуляторе, и числа 03h (непосредственное данное).

Вариант 4. Осуществить сравнение числа 06h, находящегося в аккумуляторе, и числа 03h, находящегося в ячейке памяти с адресом 0810h. Расписать биты D0–D7 F–регистра.

Вариант 5. Умножить содержимое аккумулятора 02h на 4.

Вариант 6. Поделить содержимое аккумулятора 08h на 4.

Вариант 7. Осуществить поразрядное И – логическое умножение числа 07h, находящегося в аккумуляторе, и числа 09h (непосредственное данное).

Вариант 8. Осуществить поразрядное ИЛИ – логическое сложение числа 05h, находящегося в аккумуляторе, и числа 03h, находящегося в ячейке памяти с адресом 0810h.

Вариант 9. Осуществить поразрядное исключающее ИЛИ числа 05h, находящегося в аккумуляторе, и числа 01h, находящегося в регистре C.

Вариант 10. Осуществить поразрядное ИЛИ – логическое сложение числа 07h, находящегося в аккумуляторе, и числа 08h, находящегося в регистре C.

5. Используя примеры 12–13, ознакомиться с командами передачи данных в микропроцессоре и научиться составлять программы для микропроцессора с использованием команд передачи данных.

6. Дополнить программу, составленную по п. 2, командами передачи данных (пример 12). Загрузить и проверить составленную программу на УМК.

7. Используя примеры 14–15, ознакомиться с командами передачи управления в микропроцессоре и научиться составлять программы для микропроцессора с использованием команд передачи управления и алгоритмы работы этих программ.

8. В соответствии с вариантом задания составить программу для микропроцессора. Составить алгоритм работы программы. Загрузить и проверить составленную программу на УМК.

Вариант 1. Из чисел 01h, 02h, 03h, 05h, 04h, расположенных в ячейках памяти с адресами 0830h, 0831h, 0832h, 0833h, 0834h, найти максимальное число и записать его в аккумулятор.

Вариант 2. Сложить числа 01h, 02h, 03h, 01h, расположенные в ячейках памяти с адресами 0820h, 0821h, 0822h, 0823h.

Вариант 3. Вычесть из числа 09h три числа 01h, 02h, 03h, расположенные в ячейках памяти с адресами 0820h, 0821h, 0822h.

Вариант 4. Прибавить к числу 02h три числа 01h, 02h, 03h, расположенные в ячейках памяти с адресами 0820h, 0821h, 0822h.

Вариант 5. Найти в программе Монитора код 1Bh и определить адрес ячейки памяти, где записан этот код.

Вариант 6. Осуществить поразрядное И – логическое умножение над тремя числами 01h, 05h, 03h, расположенными в ячейках памяти с адресами 0820h, 0821h, 0822h.

Вариант 7. Осуществить поразрядное ИЛИ – логическое сложение над тремя числами 01h, 05h, 03h, расположенными в ячейках памяти с адресами 0820h, 0821h, 0822h.

Вариант 8. Осуществить поразрядное исключающее ИЛИ над тремя числами 01h, 05h, 03h, расположенными в ячейках памяти с адресами 0820h, 0821h, 0822h.

Вариант 9. Из чисел 02h, 04h, 03h, 08h, 06h, расположенных в ячейках памяти с адресами 0830h, 0831h, 0832h, 0833h, 0834h найти максимальное число и записать его в регистр С.

Вариант 10. Сложить числа 01h, 02h, 03h, 01h, 02h, расположенные в ячейках памяти с адресами 0830h, 0831h, 0832h, 0833h, 0834h.

Вопросы для самоконтроля

1. Начертите структурную схему микропроцессорной системы. Объясните назначение ее элементов.

2. Начертите структурную схему микропроцессора КР580. Объясните назначение ее элементов.

3. Объясните, чем отличаются устройства, выполненные на микросхемах «жесткой» логики, от устройств, выполненных на микросхемах «программируемой» логики.

4. Состав и назначение команд арифметических операций. Приведите пример программы с использованием команд арифметических операций.

5. Состав и назначение команд логических операций. Приведите пример программы с использованием команд логических операций.

6. Состав и назначение команд передачи данных. Приведите пример программы с использованием команд передачи данных.
7. Состав и назначение команд управления. Приведите пример программы с использованием команд управления.
8. Состав и назначение регистра признаков (F–регистра).
9. Куда необходимо записать адрес ячейки памяти, чтобы обратиться к ней в микропроцессоре КР580?
10. Покажите на примере сложение двух чисел в двоичном коде.
11. Покажите на примере вычитание двух чисел в двоичном коде.
12. Покажите на примере поразрядное И – логическое умножение двух чисел в двоичном коде.
13. Покажите на примере поразрядное ИЛИ – логическое сложение двух чисел в двоичном коде.
14. Покажите на примере поразрядное исключающее ИЛИ двух чисел в двоичном коде.
15. Переведите число из двоичного кода в шестнадцатеричный код, а затем – в десятичный код.
16. При выполнении команды сравнения как определить, что числа равны; одно число больше или меньше.
17. При записи команды условного или безусловного перехода как записывается адрес метки. Приведите пример.

ЛАБОРАТОРНАЯ РАБОТА № 2

Работа микропроцессорной системы с внешними устройствами

Цель работы: при помощи учебного микропроцессорного комплекта (УМК) ознакомиться с работой микропроцессорной системы на базе микросхемы КР580ВМ80А по вводу–выводу данных от внешних устройств.

Программа работы

1. Ознакомиться с командами ввода–вывода данных на внешние устройства.
2. Ознакомиться с интерфейсом ввода–вывода параллельной информации – программируемой микросхемой КР580ВВ55А.
3. Ознакомиться с возможностью передачи данных и управления внешним устройством.
4. Ознакомиться с возможностью приема и обработки данных от внешнего устройства.

Теоретические сведения

Микропроцессорная система на базе микросхемы КР580ВМ80А имеет отдельные 16-разрядную шину адреса, 8-разрядную шину данных и шину управления. Шина адреса и шина управления обеспечивают прямую адресацию памяти объемом до 65 536 байт (64 Кбайта), 256 устройств ввода и 256 устройств вывода.

Ячейки памяти и внешнее устройство могут иметь один и тот же адрес (например 83h). Чтобы не было одновременного обращения и к ячейке памяти, и к внешнему устройству в шине управления имеются сигналы I/O_RD и I/O_WR. Когда активен сигнал I/O_RD, идет чтение данных, когда активен сигнал I/O_WR, идет запись данных (в) из микропроцессора в адресуемое внешнее устройство. Когда сигналы I/O_RD и I/O_WR не активны, то идет обращение к адресуемой ячейке памяти. Например, при выполнении команд:

- MOV R, M – пересылка данных из памяти в регистр;
- MOV M, R – пересылка данных и регистра в память;
- MVI M, B2 – пересылка непосредственных данных в память;
- ADD M – сложение содержимых ячейки памяти и аккумулятора;
- ADC M – сложение содержимых ячейки памяти и аккумулятора с переносом;
- SUB M – вычитание содержимого ячейки памяти из содержимого аккумулятора;

SBB M – вычитание содержимого ячейки памяти из содержимого аккумулятора с заемом;

INR M – инкрементирование содержимого ячейки памяти;

DCR M – декрементирование содержимого ячейки памяти;

ANA M – поразрядное И над содержимым ячейки памяти и аккумулятора;

XRA M – поразрядное исключаящее ИЛИ над содержимым ячейки памяти и аккумулятора;

ORA M – поразрядное ИЛИ над содержимым ячейки памяти и аккумулятора;

CMP M – сравнение содержимых ячейки памяти и аккумулятора сигналы I/O_RD и I/O_WR не активны, и идет обращение к адресуемой ячейке памяти. При выполнении команды **OUT B2** будет активен сигнал I/O_WR. При выполнении команды **IN B2** – активен сигнал I/O_RD.

Для передачи данных внешнему устройству необходимо записать эти данные в аккумулятор, а затем выполнить команду **OUT B2**. Вместо B2 записывается адрес внешнего устройства (например **OUT 80h**).

Для чтения данных из внешнего устройства необходимо выполнить команду **IN B2**. После выполнения этой команды данные будут находиться в аккумуляторе. Вместо B2 также записывается адрес внешнего устройства (например **IN 81h**).

Внешние устройства напрямую к шине данных не подключаются, так как выполнение команды **IN** или **OUT** длится не более 5 мксек, и медленные внешние устройства не смогут прочитать данные или выставить данные на шину данных. Целесообразно использовать интерфейс ввода–вывода.

Для внешнего устройства ввода интерфейс читал бы данные с шины данных и сохранял их. Для внешнего устройства вывода интерфейс записывал бы данные и по требованию микропроцессора выдавал их на шину данных. В качестве такого интерфейса широко используется программируемая микросхема ввода–вывода параллельной информации КР580ВВ55А.

Микросхема КР580ВВ55А имеет три порта ввода–вывода А, В, С. Порты А, В, С могут читать данные с шины данных и сохранять их для внешних устройств вывода. Также порты А, В, С могут читать данные с внешнего устройства ввода и по требованию микропроцессора выставлять их на шину данных. При этом порт С может формировать управляющие сигналы. К портам А и В могут быть подключены два 8-разрядных, а к порту С два 4-разрядных внешних устройств ввода–вывода. Микросхема КР580ВВ55А может работать с 4-мя внешними устройствами ввода–вывода.

Чтобы микросхема КР580ВВ55А начала работать, ее необходимо запрограммировать, т. е. указать, как должны работать порты А, В и С на ввод или вывод информации. При программировании записывается управляющее слово в регистр управления.

На платах (ТЭЗах) УМК микросхема КР580ВВ55А имеет:

- адрес порта А – 80h;
- адрес порта В – 81h;
- адрес порта С – 82h;
- адрес управляющего регистра – 83h.

В примерах программ, представленных ниже, порт А программируется на вывод данных (загораются светодиоды), порт В – на ввод данных (положение переключателей вкл. или откл.), порт С не используется.

Примеры применения команд ввода–вывода данных на УМК

Пример 1. Прочитать содержимое ячейки памяти с адресом 81h и записать в регистр В. Прочитать данные из порта В адрес 81h. (Данные из порта В будут находиться в аккумуляторе.)

Таблица 1

Текст программы

Адрес ячейки памяти	Код	Метка	Мнемокод	Комментарии
0800: 0801:	26 00		MVI H, 00h;	Пересылка числа 00h в регистр H
0802: 0803:	2E 81		MVI L, 81h;	Пересылка числа 81h в регистр L
0804:	46		MOV B, M;	Запись в регистр В данных из ячейки памяти
0805: 0806:	3E 82		MVI A, 82h;	Загрузка управляющего слова 82h в А–регистр: порт А – на вывод данных; порт В – на ввод данных; порт С – на вывод данных.
0807: 0808:	D3 83		OUT 83h;	Пересылка управляющего слова 82h в порт управляющего регистра по адресу 83h
0809: 080A:	DB 81		IN 81h;	Чтение данных из порта В по адресу 81h и запись в А–регистр (аккумулятор)
080B:	00		NOP;	Пустая операция

Примечание 1. Для чтения данных из порта В или записи данных в порт А необходимо вначале записать управляющее слово 82h в порт управляющего регистра по адресу 83h.

До выполнения программы:
 в регистре Н любое число;
 в регистре L любое число;
 в регистре В любое число;
 в аккумуляторе любое число.

После выполнения программы:
 в регистре Н число 00h;
 в регистре L число 81h;
 в регистре В число 87h;
 в аккумуляторе число 00h (при не включенных ключах).

Примечание 2. Обратите внимание, что в примере 1 идет обращение по одному адресу 81h и к ячейке памяти (MOV B, M(H,L)), и к устройству ввода – вывода (IN 81h). Данные, полученные из ячейки памяти и из устройства ввода – вывода, разные.

Этапы выполнения программы на УМК:

Загрузка программы:

П 0800 ВП

26 _ 00 _ 2E _ 81 _ 46 _ 3E _ 82 _ D3 _ 83 _ DB _ 81 _ 00 ВП

Запуск программы: СТ 0800 _ 080В ВП

Просмотр результата: РГ А _ В

Пример 2. Включить в порте А, начиная с нижнего, через один светодиоды. (Чтобы светодиоды горели через один, в аккумулятор необходимо записать число 55h → 0101 0101b.)

Таблица 2

Текст программы

Адрес ячейки памяти	Код	Метка	Мнемокод	Комментарии
0800: 0801:	3E 82		MVI A, 82h;	Загрузка управляющего слова 82h в А–регистр:
0802: 0803:	D3 83		OUT 83h;	Пересылка управляющего слова 82h в порт управляющего регистра по адресу 83h
0804: 0805:	3E 55		MVI A, 55h;	Чтобы загорелись через один светодиоды в порту А, загружаем в А–регистр число 55h
0806:	2F		CMA;	Так как светодиоды включаются не логической 1, а логическим 0, то осуществляется инвертирование А–регистра
0807: 0808:	D3 80		OUT 80h;	Пересылка содержимого А–регистра в порт А по адресу 80h
0809:	00		NOP;	Пустая операция

Примечание 3. Для записи данных в порт А вначале необходимо записать эти данные в аккумулятор, а затем выполнить команду OUT 80h. Принимая во внимание, что светодиоды в порте А включаются не логической 1, а логическим 0, то необходимо осуществлять инвертирование содержимого А–регистра.

До выполнения программы:
в аккумуляторе любое число;
в порту А светодиоды не горят.

Выполнение команды СМА:

0101 0101

1010 1010

Результат заносится в аккумулятор 1010 1010b → АAh.

После выполнения программы:
в аккумуляторе число 1010 1010 → АAh;
в порте А, начиная с нижнего, через один горят светодиоды.

Этапы выполнения программы на УМК:

Загрузка программы:

П 0800 ВП

3E _ 82 _ D3 _ 83 _ 3E _ 55 _ 2F _ D3 _ 80 _ 00 ВП

Запуск программы: СТ 0800 _ 0809 ВП

Просмотр результата: РГ А

Пример 3. Прочитать содержимое порта В, прибавить к нему число 02h и переслать скорректированное число в порт А.

Таблица 3

Текст программы

Адрес ячейки памяти	Код	Метка	Мнемокод	Комментарии
0800: 0801:	3E 82		MVI A, 82h;	Загрузка управляющего слова 82h в А–регистр
0802: 0803:	D3 83		OUT 83h;	Пересылка управляющего слова 82h в порт управляющего регистра по адресу 83h
0804: 0805:	DB 81		IN 81h;	Чтение данных из порта В по адресу 81h и запись в А–регистр (аккумулятор)
0806: 0807:	C6 02		ADI 02h;	Сложение числа 02h с содержимым А–регистра
0808:	2F		СМА;	Инвертирование А–регистра (аккумулятора)
0809: 080A:	D3 80		OUT 80h;	Пересылка содержимого А–регистра в порт А по адресу 80H
080B:	00		NOP;	Пустая операция

До выполнения программы:

в порту В нижний ключ включен, а остальные ключи разомкнуты:

0000 0001b → 01h;

в аккумуляторе любое число;

в порту А светодиоды не горят.

Выполнение команды ADI 02h:

$$\begin{array}{r} 0000\ 0001 \\ +\ 0000\ 0010 \\ \hline 0000\ 0011 \end{array}$$

Результат сложения заносится в аккумулятор 0000 0011b → 03h.

Выполнение команды SMA:

$$\begin{array}{r} 0000\ 0011 \\ \hline 1111\ 1100 \end{array}$$

Результат заносится в аккумулятор 1111 1100b → FCh.

После выполнения программы:

в порте В нижний ключ включен, а остальные ключи разомкнуты:

0000 0001b → 01h;

в аккумуляторе число 1111 1100b → FCh;

в порте А, начиная с нижнего, горят два светодиода.

Этапы выполнения программы на УМК:

Загрузка программы:

П 0800 ВП

3E _ 82 _ D3 _ 83 _ DB _ 81 _ C6 _ 02 _ 2F _ D3 _ 80 _ 00 ВП

Запуск программы: СТ 0800 _ 080В ВП

Просмотр результата: РГ А

Пример 4. Организовать в порте А на светодиодах эффект «бегущего огня». Эффект заключается в последовательном включении снизу вверх одного светодиода.

Таблица 4

Текст программы

Адрес ячейки памяти	Код	Метка	Мnemonic	Комментарии
0800: 0801:	3E 82		MVI A, 82h;	Загрузка управляющего слова 82h в А-регистр
0802: 0803:	D3 83		OUT 83h;	Пересылка управляющего слова 82h в порт управляющего регистра по адресу 83h

Адрес ячейки памяти	Код	Метка	Мнемокод	Комментарии
0804: 0805:	3E FE		MVI A, FEh;	Чтобы загорелся нижний светодиод, загружаем в А–регистр число FEh
0806: 0807:	D3 80	M1:	OUT 80h;	Пересылка содержимого А–регистра в порт А по адресу 80h
0808:	F5		PUSH PSW;	Чтобы содержимое А–регистра не было утеряно, производим запись в стек содержимого А–регистра и F–регистра (регистра флагов)
Временная задержка				
0809: 080A: 080B:	11 FF 1F		LXI D, 1FFFh;	Чтобы начать отсчет времени, записываем двухбайтовое число 1FFFh в регистровую пару DE: в D–регистр (старший байт) 1Fh, в E–регистр (младший байт) FFh (максимальный промежуток времени, когда число FFFFh)
080C:	1B	M2:	DCX D;	Уменьшение на единицу содержимого регистровой пары DE
080D:	7A		MOV A, D;	Пересылка содержимого D–регистра в А–регистр
080E:	B3		ORA E;	Поразрядное ИЛИ – логическое сложение над E–регистром и А–регистром
080F: 0810: 0811:	C2 0C 08		JNZ M2;	Если содержимое А–регистра не ноль, то перейти на метку M2. Если ноль, то перейти к следующей команде
Продолжение программы				
0812:	F1		POP PSW;	Возвращение из стека содержимого А–регистра и F–регистра
0813:	07		RLC;	Чтобы огоньки двигались вверх, необходим циклический сдвиг влево содержимого А–регистра
0814: 0815: 0816:	C3 06 08		JMP M1;	Чтобы процесс повторялся периодически, осуществляется безусловный переход на метку M1
0817:	00		NOP;	Пустая операция

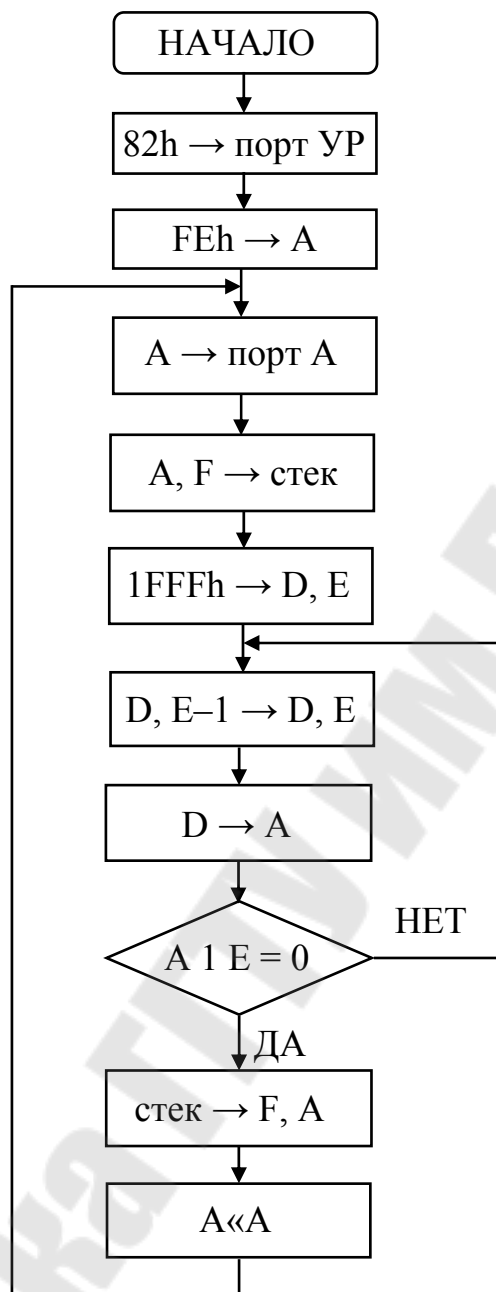


Рис. 1. Алгоритм работы программы: порт УР – порт управляющего регистра; $A \vee E$ – логическое сложение ИЛИ регистра А и регистра Е; $A \ll A$ – циклический сдвиг влево содержимого регистра А

Примечание 4. Программа в примере 4 работает в режиме бесконечного цикла, т. е. у нее нет конца. Чтобы остановить эту программу, необходимо нажать кнопку сброс (СБ) на УМК.

Примечание 5. Стек – область памяти, в которой можно хранить данные без явной адресации, используя одну команду (PUSH). С помощью команды (POP) данные можно извлечь из стека, однако только те, которые были записаны в нем последними. Стек работает по принципу, согласно которому величина, записанная последней, извлекается первой. Адрес стека храниться в указателе стека (регистр S).

Этапы выполнения программы на УМК:

Загрузка программы:

П 0800 ВП

3E _ 82 _ D3 _ 83 _ 3E _ FE _ D3 _ 80 _ F5 _ 11 _ FF _ 1F _ 1B _ 7A _
 B3 _ C2 _ 0C _ 08 _ F1 _ 07 _ C3 _ 06 _ 08 _ 00 ВП

Запуск программы: СТ 0800 _ 0817 ВП

Пример 5. Если включен верхний ключ в порте В, то в порте А на светодиодах – эффект «бегущий огонь». Эффект заключается в последовательном включении снизу вверх одного светодиода. Если не включен верхний ключ либо включены другие ключи в порте В, то в порте А светодиоды выключены.

Таблица 5

Текст программы

Адрес ячейки памяти	Код	Метка	Мнемокод	Комментарии
0800: 0801:	3E 82		MVI A, 82h;	Загрузка управляющего слова 82h в А–регистр
0802: 0803:	D3 83		OUT 83h;	Пересылка управляющего слова 82h в порт управляющего регистра по адресу 83h
0804: 0805:	06 FE		MVI B, FEh;	Чтобы горел один светодиод, в регистр В записывается число FEh
0806: 0807:	DB 81	M3:	IN 81h;	Чтение данных из порта В по адресу 81h и запись в А–регистр (аккумулятор)
0808: 0809:	FE 80		CPI 80h;	Сравнение содержимого А–регистра с числом 80h
080A: 080B: 080C:	C2 20 08		JNZ M4;	Если число 80h не равняется содержимому А–регистра, то переход на метку М4. Если число 80h равняется, то продолжить программу
Подпрограмма 1 организации в порте А эффекта «бегущего огня»				
080D:	78		MOV A, B;	Запись данных из регистра В в аккумулятор
080E: 080F:	D3 80		OUT 80h;	Пересылка содержимого А–регистра в порт А по адресу 80h

Продолжение табл. 5

Адрес ячейки памяти	Код	Метка	Мнемокод	Комментарии
0810:	F5		PUSH PSW;	Чтобы содержимое А–регистра не было утеряно, производим запись в стек содержимого А–регистра и F–регистра (регистра флагов)
Временная задержка подпрограммы 1				
0811: 0812: 0813:	11 FF 1F		LXI D, 1FFFh;	Чтобы начать отсчет времени, записываем двухбайтовое число 1FFFh в регистровую пару DE: в D–регистр (старший байт) 1Fh, в E–регистр (младший байт) FFh (максимальный промежуток времени, когда число FFFFh)
0814:	1B	M2:	DCX D;	Уменьшение на единицу содержимого регистровой пары DE
0815:	7A		MOV A, D;	Пересылка содержимого D–регистра в A–регистр
0816:	B3		ORA E;	Поразрядное ИЛИ – логическое сложение над E–регистром и A–регистром
0817: 0818: 0819:	C2 14 08		JNZ M2;	Если содержимое A–регистра не ноль, то перейти на метку M2. Если ноль, то перейти к следующей команде
Продолжение подпрограммы 1				
081A:	F1		POP PSW;	Возвращение из стека содержимого A–регистра и F–регистра
081B	07		RLC;	Чтобы огоньки двигались вверх, необходим циклический сдвиг влево содержимого A–регистра
081C:	47		MOV B, A	Запись данных из аккумулятора в регистр B
081D: 081E: 081F:	C3 06 08		JMP M3;	Чтобы процесс повторялся периодически, осуществляется безусловный переход на метку M3
Подпрограмма 2 в порте A светодиоды не горят				
0820: 0821:	3E FF	M4:	MVI A, FFh;	Чтобы не горели светодиоды, загружаем в A–регистр число FFh

Адрес ячейки памяти	Код	Метка	Мнемокод	Комментарии
0822: 0823:	D3 80		OUT 80h;	Пересылка содержимого А-регистра в порт А по адресу 80h
0824: 0825:	06 FE		MVI B, FEh;	Запись числа FEh в регистр В
0826: 0827: 0828:	C3 06 08		JMP M3;	Чтобы процесс повторялся периодически, осуществляется безусловный переход на метку М3
0829:	00		NOP;	Пустая операция

Этапы выполнения программы на УМК:

Загрузка программы:

П 0800 ВП 3E 82 D3 83 06 FE DB 81 FE 80 C2 20
 08 78 D3 80 F5 11 FF 1F 1B 7A B3 C2 14 08 F1
 07 47 C3 06 08 3E FF D3 80 06 FE C3 06 08
 00 ВП

Запуск программы: СТ 0800 _ 0829 ВП

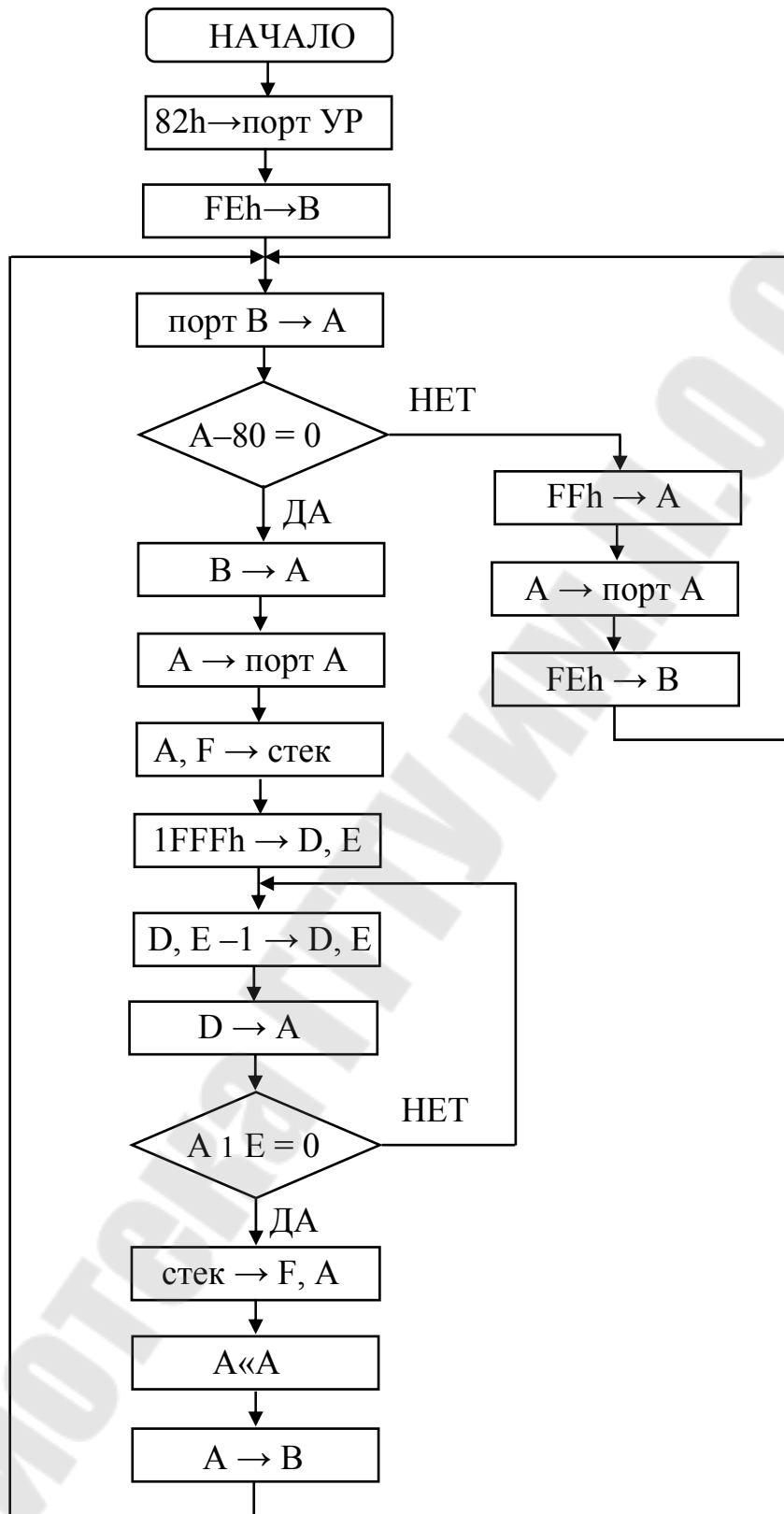


Рис. 2. Алгоритм работы программы

Порядок выполнения работы

1. Используя примеры 1–3, ознакомьтесь с командами ввода–вывода данных в микропроцессоре и научитесь составлять программы для микропроцессора с использованием этих команд, а также загружать и выполнять программы на УМК.

2. В соответствии с вариантом задания составить программу для микропроцессора. Загрузить и проверить составленную программу на УМК. При этом должно быть показано состояние регистров и ячеек памяти, используемых в программе, до и после выполнения программы, а также выполнение арифметической команды (примеры 1–3).

Вариант 1. При помощи ключей выставить в порте В число 07h. Прочитать содержимое порта В, вычесть из него число 04h и переслать скорректированное число в порт А.

Вариант 2. При помощи ключей выставить в порте В число 02h. Прочитать содержимое порта В, увеличить на 1 и переслать скорректированное число в порт А.

Вариант 3. При помощи ключей выставить в порте В число 08h. Прочитать содержимое порта В, уменьшить на 1 и переслать скорректированное число в порт А.

Вариант 4. При помощи ключей выставить в порте В число 07h. Прочитать содержимое порта В. Над содержимым порта В и числом 03h осуществить поразрядное логическое умножение – И. Скорректированное число переслать в порт А.

Вариант 5. При помощи ключей выставить в порте В число 03h. Прочитать содержимое порта В. Над содержимым порта В и числом 04h осуществить поразрядное логическое сложение – ИЛИ. Скорректированное число переслать в порт А.

Вариант 6. При помощи ключей выставить в порте В число 07h. Прочитать содержимое порта В. Над содержимым порта В и числом 02h осуществить поразрядное логическое исключающее ИЛИ. Скорректированное число переслать в порт А.

Вариант 7. При помощи ключей выставить в порте В число 07h. Прочитать содержимое порта В и осуществить циклический сдвиг влево. Скорректированное число переслать в порт А.

Вариант 8. При помощи ключей выставить в порте В число 06h. Прочитать содержимое порта В и осуществить циклический сдвиг вправо. Скорректированное число переслать в порт А.

Вариант 9. При помощи ключей выставить в порте В число 06h. Прочитать содержимое порта В, вычесть из него число 03h и переслать скорректированное число в порт А.

Вариант 10. При помощи ключей выставить в порте В число 04h. Прочитать содержимое порта В. К содержимому порта В прибавить число 03h. Скорректированное число переслать в порт А.

3. Используя примеры 4–5, ознакомиться с использованием команд ввода–вывода данных в программах, работающих в режиме бесконечного цикла, и научиться составлять подобные программы для микропроцессора, а также загружать и выполнять эти программы на УМК.

4. В соответствии с вариантом задания составить программу, работающую в режиме бесконечного цикла, для микропроцессора. Загрузить и проверить составленную программу на УМК. При этом должен быть составлен алгоритм работы программы (примеры 4–5).

Вариант 1. Если в порте В число 03h, то в порте А на светодиодах эффект «бегущего огня». Эффект заключается в последовательном включении снизу вверх двух светодиодов. Если в порте В другое число, то в порте А светодиоды выключены.

Вариант 2. Если в порте В число 01h, то в порте А на светодиодах эффект «мигающего огня». Эффект заключается в последовательном включении снизу четырех светодиодов, а затем сверху – четырех светодиодов. Если в порте В другое число, то в порте А светодиоды выключены.

Вариант 3. Если в порте В число 07h, то в порте А на светодиодах эффект «мигающего огня». Эффект заключается в последовательном включении–выключении всех светодиодов. Если в порте В другое число, то в порте А светодиоды выключены.

Вариант 4. Если в порте В число 08h, то в порте А на светодиодах эффект «мигающего огня». Эффект заключается в последовательном включении выключении через два светодиода. Если в порте В другое число, то в порте А светодиоды выключены.

Вариант 5. Если в порте В число 04h, то в порте А на светодиодах эффект «бегущего огня». Эффект заключается в последовательном включении сверху вниз двух светодиодов. Если в порте В другое число, то в порте А светодиоды выключены.

Вариант 6. Если в порте В число 05h, то в порте А на светодиодах эффект «бегущего огня». Эффект заключается в последовательном включении сверху вниз трех светодиодов. Если в порте В другое число, то в порте А светодиоды выключены.

Вариант 7. Если в порте В число 02h, то в порте А на светодиодах эффект «бегущего огня». Эффект заключается в последовательном включении сверху вниз одного светодиода. Если в порте В другое число, то в порте А светодиоды выключены.

Вариант 8. Если в порте В число 09h, то в порте А на светодиодах эффект «бегущего огня». Эффект заключается в последовательном включении снизу вверх трех светодиодов. Если в порте В другое число, то в порте А светодиоды выключены.

Вариант 9. Если в порте В число 40h, то в порте А на светодиодах эффект «бегущего огня». Эффект заключается в последовательном включении сверху вниз четырех светодиодов. Если в порте В другое число, то в порте А светодиоды выключены.

Вариант 10. Если в порте В число 60h, то в порте А на светодиодах эффект «бегущего огня». Эффект заключается в последовательном включении снизу вверх четырех светодиодов. Если в порте В другое число, то в порте А светодиоды выключены.

Вопросы для самоконтроля

1. Какие команды микропроцессора позволяют обратиться к памяти, а какие к внешним устройствам ввода–вывода?

2. Какая шина адреса и шина данных у микропроцессора КР580ВМ80А? К какому максимальному количеству памяти и внешних устройств может обратиться этот микропроцессор?

3. Какая микросхема используется в лабораторной работе в качестве интерфейса ввода–вывода. Что необходимо сделать, чтобы эта микросхема начала работать как интерфейс ввода–вывода?

4. Будут ли получены одинаковые данные при обращении по одному адресу (например, 81h) к памяти и внешнему устройству? Приведите пример такой программы.

5. Как осуществить запись данных в порт А? Приведите пример программы.

6. Как осуществить чтение данных из порта В? Приведите пример программы.

7. Как осуществляется запись управляющего слова в интерфейс ввода–вывода в лабораторной работе? Приведите пример программы.

8. Какая особенность при «загорании» светодиодов в порте А в лабораторной работе? Приведите пример программы, которая позволяет «зажечь» четыре нижних светодиода в порте А.

9. Что такое стек? Как записать данные в стек и как извлечь данные из стека?

10. Какие преимущества и недостатки при обращении к стеку и памяти?

ЛИТЕРАТУРА

1. Янсен, Й. Курс цифровой электроники : в 4 т. Т. 4. Микрокомпьютеры / Й. Янсен. – Москва : Мир, 1987. – 406 с.
2. Калабеков, Б. А. Микропроцессоры и их применение в системах передачи и обработки сигналов : учеб. пособие для вузов / Б. А. Калабеков. – Москва : Радио и связь, 1988. – 368 с.
3. Гуртовцев, А. Л. Программы для микропроцессоров : справ. пособие / А. Л. Гуртовцев, С. В. Гудыменко. – Минск : Выш. шк., 1989. – 352 с.
4. Токхайм, Р. Л. Микропроцессоры. Курс и упражнения / Р. Л. Токхайм ; пер. с англ. ; под ред. В. Н. Грачева. – Москва : Энергоатомиздат, 1988. – 336 с.

ПРИЛОЖЕНИЕ

Таблица П.1.1

Набор команд микропроцессора КР580

Формат, байт	Время, такт	Код	Мnemonic	Наименование команды	Описание операций	Биты признаков				
						S	Z	AC	P	CY
1	5	*	MOV R1, R2	Пересылка данных из регистра в регистр	$(R1) \leftarrow (R2)$	-	-	-	-	-
1	7	*	MOV R, M	Пересылка данных из памяти в регистр	$(R) \leftarrow ((H, L))$	-	-	-	-	-
1	7	*	MOV M, R	Пересылка данных из регистра в память	$((H, L)) \leftarrow (R)$	-	-	-	-	-
2	7	*	MVI M, B2	Пересылка непосредственных данных в регистр	$(R) \leftarrow (B2)$	-	-	-	-	-
2	10	36	MVI M, B2	Пересылка непосредственных данных в память	$((H, L)) \leftarrow (B2)$	-	-	-	-	-
3	10	*	LXI RR, B2B3	Непосредственная загрузка регистровой пары	$(RRH) \leftarrow (B3),$ $(RRL) \leftarrow (B2)$	-	-	-	-	-
3	13	3A	LDA B2B3	Прямая загрузка A-регистра	$(A) \leftarrow ((B3, B2))$	-	-	-	-	-
3	13	32	STA B2B3	Прямое запоминание содержимого A-регистра	$((B3, B2)) \leftarrow (A)$	-	-	-	-	-
1	7	*	LDAX RR	Косвенная загрузка A-регистра	$(A) \leftarrow ((RR))$	-	-	-	-	-
1	7	*	STAX RR	Косвенное запоминание содержимого A-регистра	$((RR)) \leftarrow (A)$	-	-	-	-	-
3	16	2A	LHLD B2B3	Прямая загрузка H-пары регистров	$(L) \leftarrow ((B3, B2)),$ $(H) \leftarrow ((B3, B2)+1)$	-	-	-	-	-
3	16	22	SHLDY B2B3	Прямое запоминание содержимого H-пары регистров	$((B3, B2)) \leftarrow (L),$ $((B3, B2)+1) \leftarrow (H)$	-	-	-	-	-

Продолжение табл. П.1.1

Формат, байт	Время, такт	Код	Мnemonic	Наименование команды	Описание операций	Биты признаков				
						S	Z	AC	P	CY
1	4	EB	XCHG	Обмен между H- и D-парами регистров	(H) ↔ (D), (L) ↔ (E)	-	-	-	-	-
<i>Команды арифметических операций</i>										
1	4	*	ADD R	Сложение содержимых регистра и A-регистра	$(A) \leftarrow (A) + (R)$	+	+	+	+	+
1	7	86	ADD M	Сложение содержимых ячейки памяти и A-регистра	$(A) \leftarrow (A) + ((H, L))$	+	+	+	+	+
2	7	C6	ADI B2	Сложение непосредственных данных и содержимого A-регистра	$(A) \leftarrow (A) + (B2)$	+	+	+	+	+
1	4	*	ADC R	Сложение содержимых регистра и A-регистра с переносом	$(A) \leftarrow (A) + (R) + (CY)$	+	+	+	+	+
1	7	8E	ADC M	Сложение содержимых ячейки памяти и A-регистра с переносом	$(A) \leftarrow (A) + ((H, L)) + (CY)$	+	+	+	+	+
2	7	CE	ACI B2	Сложение непосредственных данных и A-регистра с переносом	$(A) \leftarrow (A) + (B2) + (CY)$	+	+	+	+	+
1	4	*	SUB R	Вычитание содержимого регистра из содержимого A-регистра	$(A) \leftarrow (A) - (R)$	+	+	+	+	+
1	7	96	SUB M	Вычитание содержимого ячейки памяти из содержимого A-регистра	$(A) \leftarrow (A) - ((H, L))$	+	+	+	+	+

Продолжение табл. П.1.1

Формат, байт	Время, такт	Код	Мnemonic	Наименование команды	Описание операций	Биты признаков				
						S	Z	AC	P	CY
2	7	D6	SUI B2	Вычитание непосредственных данных из содержимого A-регистра	$(A) \leftarrow (A) - (B2)$	+	+	+	+	+
1	4	*	SBB R	Вычитание содержимого регистра из содержимого A-регистра с заемом	$(A) \leftarrow (A) - (R) - (CY)$	+	+	+	+	+
1	7	9E	SBB M	Вычитание содержимого ячейки памяти из содержимого A-регистра с заемом	$(A) \leftarrow (A) - ((H, L)) - (CY)$	+	+	+	+	+
2	7	DI	SBI B2	Вычитание непосредственных данных из содержимого A-регистра с заемом	$(A) \leftarrow (A) - (B2) - (CY)$	+	+	+	+	+
1	5	*	INR R	Инкрементирование содержимого регистра	$(R) \leftarrow (R) + 1$	+	+	+	+	-
1	10	34	INR M	Инкрементирование содержимого ячейки памяти	$((H, L)) \leftarrow ((H, L)) + 1$	+	+	+	+	-
1	5	*	INX RR	Инкрементирование содержимого регистровой пары	$(RR) \leftarrow (RR) + 1$	-	-	-	-	-
1	5	*	DCR R	Декрементирование содержимого регистра	$(R) \leftarrow (R) - 1$	+	+	+	+	-
1	10	35	DCR M	Декрементирование содержимого ячейки памяти	$((H, L)) \leftarrow ((H, L)) - 1$	+	+	+	+	-
1	5	*	DCX RR	Декрементирование содержимого регистровой пары	$(RR) \leftarrow (RR) - 1$	-	-	-	-	-

Продолжение табл. П.1.1

Формат, байт	Время, такт	Код	Мnemonic	Наименование команды	Описание операций	Биты признаков				
						S	Z	AC	P	CY
1	10	*	DAD RR	Сложение содержимых регистровой пары и N-пары регистров	$((H, L)) \leftarrow ((H, L)) + (RR)$	-	-	-	-	+
1	4	27	DAA	Десятичная коррекция содержимого A-регистра	Если $(AL) > 9$ или $(AC-1)$, то $(A) \leftarrow (A) + 6$; если затем $(AH) > 9$ или $(CY)=1$, то $(A) \leftarrow (A) + 6 * 2^4$	+	+	+	+	-
<i>Команды логических операций</i>										
1	4	*	ANA R	Поразрядное И над содержимым регистра и A-регистра	$(A) \leftarrow (A) \wedge (R)$	+	+	0	+	0
1	7	A6	ANA M	Поразрядное И над содержимым ячейки памяти и A-регистра	$(A) \leftarrow (A) \wedge ((H, L))$	+	+	0	+	0
2	7	E6	ANI B2	Поразрядное И над непосредственными данными и содержимым A-регистра	$(A) \leftarrow (A) \wedge (B2)$	+	+	0	+	0
1	4	*	XRA R	Поразрядное ИСКЛЮЧАЮЩЕЕ ИЛИ над содержимым регистра и A-регистра	$(A) \leftarrow (A) \vee (R)$	+	+	0	+	0
1	7	AЕ	XRA M	Поразрядное ИСКЛЮЧАЮЩЕЕ ИЛИ над содержимым ячейки памяти и A-регистра	$(A) \leftarrow (A) \vee ((H, L))$	+	+	0	+	0

Продолжение табл. П.1.1

Формат, байт	Время, такт	Код	Мnemonic	Наименование команды	Описание операций	Биты признаков				
						S	Z	AC	P	CY
2	7	EE	XRI B2	Поразрядное ИСКЛЮЧАЮЩЕЕ ИЛИ над непосредственными данными и содержимым A-регистра	$(A) \leftarrow (A) \nabla (B2)$	+	+	0	+	0
1	4	*	ORA R	Поразрядное ИЛИ над содержимым регистра и A-регистра	$(A) \leftarrow (A) \vee (R)$	+	+	0	+	0
1	7	B6	ORA M	Поразрядное ИЛИ над содержимым ячейки памяти и A-регистра	$(A) \leftarrow (A) \vee ((H, L))$	+	+	0	+	0
2	7	F6	ORI B2	Поразрядное ИЛИ над непосредственными данными и содержимым A-регистра	$(A) \leftarrow (A) \vee (B2)$	+	+	0	+	0
1	4	*	CMP R	Сравнение содержимых регистра и A-регистра	Если $(A) - (RM) = 0$, то $(CY) = 0, (Z) = 1$; если > 0 , то $(CY) = 0, (Z) = 0$; если < 0 , то $(CY) = 1, (Z) = 0$	+	+	+	+	0
1	7	BE	CMP M	Сравнение содержимых ячейки памяти и A-регистра						
2	7	FE	CPI B2	Сравнение непосредственных данных с содержимым A-регистра						
1	4	07	RLC	Циклический сдвиг влево содержимого A-регистра	$(A_{i+1}) \leftarrow (A_i), (A_0) \leftarrow (A_7), (CY) \leftarrow (A_7)$	-	-	-	-	+
1	4	17	RAL	Циклический сдвиг влево содержимого A-регистра через перенос	$(A_{i+1}) \leftarrow (A_i), (A_0) \leftarrow (CY), (CY) \leftarrow (A_7)$	-	-	-	-	+

Продолжение табл. П.1.1

Формат, байт	Время, такт	Код	Мnemonic	Наименование команды	Описание операций	Биты признаков				
						S	Z	AC	P	CY
1	4	OF	RRC	Циклический сдвиг вправо содержимого A-регистра	$(A_i) \leftarrow (A_{i+1}), (A_7) \leftarrow (A_0), (CY) \leftarrow (A_0)$	-	-	-	-	+
1	4	1F	RAR	Циклический сдвиг вправо содержимого A-регистра через перенос	$(A_i) \leftarrow (A_{i+1}), (A_7) \leftarrow (CY), (CY) \leftarrow (A_0)$	-	-	-	-	+
1	4	2F	CMA	Дополнение (инвертирование) содержимого A-регистра	$(A) \leftarrow (\bar{A})$	-	-	-	-	-
1	4	3F	CMC	Дополнение содержимого признака переноса	$(CY) \leftarrow (\bar{CY})$	-	-	-	-	+
1	4	37	STC	Установка в единицу признака переноса	$(CY) \leftarrow 1$	-	-	-	-	1
<i>Команды передачи управления</i>										
3	10	C3	JMP B2B3	Безусловный переход		-	-	-	-	-
3	10	DA	JC B2B3	Переход, если перенос	$(PC) \leftarrow (B2, B3)$ Если условие верно, то $(PC) \leftarrow (B2, B3)$, в противном случае $(PC) \leftarrow (PC) + 3$	-	-	-	-	-
3	10	D2	JNC B2B3	Переход, если не перенос		-	-	-	-	-
3	10	CA	JZ B2B3	Переход, если нуль		-	-	-	-	-
3	10	C2	JNZ B2B3	Переход, если не нуль		-	-	-	-	-
3	10	F2	JP B2B3	Переход, если плюс		-	-	-	-	-
3	10	FA	JM B2B3	Переход, если минус		-	-	-	-	-
3	10	EA	JPE B2B3	Переход, если четно		-	-	-	-	-
3	10	E2	JPO B2B3	Переход, если нечетно		-	-	-	-	-

Продолжение табл. П.1.1

Формат, байт	Время, такт	Код	Мnemonic	Наименование команды	Описание операций	Биты признаков				
						S	Z	AC	P	CY
3	17	CD	CALL B2B3	Безусловный вызов подпрограммы	$((SP)-1) \leftarrow (PCH)$, $((SP)-2) \leftarrow (PCL)$, $(SP) \leftarrow (SP) - 2$, $(PC) \leftarrow (B2, B3)$	-	-	-	-	-
3	11/17	DC	CC B2B3	Вызов подпрограммы, если перенос	Если условие верно, то $((SP)-1) \leftarrow (PCH)$, $((SP)-2) \leftarrow (PCL)$, $(SP) \leftarrow (SP) - 2$, $(PC) \leftarrow (B2, B3)$, в противном случае $(PC) \leftarrow (PC) + 3$ $(PCL) \leftarrow ((SP))$, $(PCH) \leftarrow ((SP) + 1)$, $(SP) \leftarrow (SP) + 2$	-	-	-	-	-
3	11/17	D4	CNC B2B3	Вызов подпрограммы, если не перенос		-	-	-	-	-
3	11/17	CC	CZ B2B3	Вызов подпрограммы, если нуль		-	-	-	-	-
3	11/17	C4	CNZ B2B3	Вызов подпрограммы, если не нуль		-	-	-	-	-
3	11/17	F4	CP B2B3	Вызов подпрограммы, если плюс		-	-	-	-	-
3	11/17	FC	CM B2B3	Вызов подпрограммы, если минус		-	-	-	-	-
3	11/17	EC	CPE B2B3	Вызов подпрограммы, если четно		-	-	-	-	-
3	11/17	E4	CPO B2B3	Вызов подпрограммы, если нечетно		-	-	-	-	-
1	10	C9	RET	Возврат		-	-	-	-	-
1	5/11	D8	RC	Возврат, если перенос		Если условие верно, то $(RLC) \leftarrow ((SP))$, $(PCH) \leftarrow ((SP)) + 1$, $(SP) \leftarrow (SP) + 2$, в противном случае $(PC) \leftarrow (PC) + 1$	-	-	-	-
1	5/11	D0	RNC	Возврат, если не перенос	-		-	-	-	-
1	5/11	C8	RZ	Возврат, если нуль	-		-	-	-	-
1	5/11	C0	RNZ	Возврат, если не нуль	-		-	-	-	-
1	5/11	F0	RP	Возврат, если плюс	-		-	-	-	-
1	5/11	F8	RM	Возврат, если минус	-		-	-	-	-
1	5/11	E8	RPE	Возврат, если четно	-		-	-	-	-
1	5/11	E0	RPO	Возврат, если нечетно	-		-	-	-	-

Продолжение табл. П.1.1

Формат, байт	Время, такт	Код	Мnemonic	Наименование команды	Описание операций	Биты признаков				
						S	Z	AC	P	CY
1	11	C7	RST 0	Рестарт по 0-му уровню	$((SP) - 1) \leftarrow (PCH),$ $((SP) - 2) \leftarrow (PCL),$ $(SP) \leftarrow (SP) - 2,$ $(PC) \leftarrow 8 * N$	-	-	-	-	-
1	11	CF	RST 1	Рестарт по 1-му уровню		-	-	-	-	-
1	11	D7	RST 2	Рестарт по 2-му уровню		-	-	-	-	-
1	11	DF	RST 3	Рестарт по 3-му уровню		-	-	-	-	-
1	11	E7	RST 4	Рестарт по 4-му уровню		-	-	-	-	-
1	11	EF	RST 5	Рестарт по 5-му уровню		-	-	-	-	-
1	11	F7	RST 6	Рестарт по 6-му уровню		-	-	-	-	-
1	11	FF	RST 7	Рестарт по 7-му уровню		-	-	-	-	-
1	5	E9	PCHL	Запись содержимого N-пары регистров	$(PCH) \leftarrow (H),$ $(PCL) \leftarrow (L)$	-	-	-	-	-
<i>Команды стека, ввода-вывода и управления</i>										
1	11	*	PUSH RR	Запись в стек содержимого регистровой пары	$((SP) - 1) \leftarrow (RRH),$ $((SP) - 2) \leftarrow (RRL),$ $(SP) \leftarrow (SP) - 2$	-	-	-	-	-
1	11	F5	PUSH PSW	Запись в стек слова состояния процессора	$((SP) - 1) \leftarrow (A),$ $(SP) \leftarrow ((SP) - 2),$ $((SP) - 2) \leftarrow (F)$	-	-	-	-	-
1	11	*	POP RR	Чтение из стека содержимого регистровой пары	$(RRL) \leftarrow ((SP)),$ $(RRH) \leftarrow ((SP) + 1),$ $(SP) \leftarrow (SP) + 2$	-	-	-	-	-
1	11	F1	POP PSW	Чтение из стека слова состояния процессора	$(F) \leftarrow ((SP)),$ $(A) \leftarrow ((SP) + 1),$ $(SP) \leftarrow (SP) + 2$	+	+	+	+	-

Окончание табл. П.1.1

Формат, байт	Время, такт	Код	Мnemonic	Наименование команды	Описание операций	Биты признаков				
						S	Z	AC	P	CY
1	18	E3	XTHL	Обмен между вершиной стека и H-парой регистров	$(L) \leftarrow ((SP)),$ $(H) \leftarrow ((SP) + 1)$	-	-	-	-	
1	5	F9	SPHL	Запись содержимого H-пары регистров указатель стека	$(SP) \leftarrow (H, L)$	-	-	-	-	
2	10	DB	IN B2	Ввод данных	$(A) \leftarrow ((B2))$	-	-	-	-	
2	10	D3	OUT B2	Вывод данных	$((B2)) \leftarrow (A)$	-	-	-	-	
1	4	FB	EI	Разрешение прерывания	$(INTE) \leftarrow 1$	-	-	-	-	
1	4	F3	DI	Запрет прерывания	$(INTE) \leftarrow 0$	-	-	-	-	
1	7	76	HLT	Останов	$(PC) \leftarrow (PC) + 1$ СТОП	-	-	-	-	
1	4	00	NOP	Пустая операция	$(PC) \leftarrow (PC) + 1$	-	-	-	-	

Условные обозначения в наборе команд

Обозначение	Комментарий
R1, R2, R	Один из регистров A, B, C, D, E, H, L
M	Ячейка памяти, адресуемая содержимым N-пары регистров
RM1, RM2, RM	Один из регистров A, B, C, D, E, H, L; ячейка памяти M или байт B2
RR	Одна из регистровых пар B-(B, C), D-(D, E), H-(H, L) или указатель стека SP
RRH, RRL	Старший и младший регистры регистровой пары RR
PCH, PCL	Старший и младший регистры программного счетчика PC
RH, RL	Старший и младший полубайты регистра (R)
R_i	Разряд i -регистра (R), где $i = 0, \dots, 7$
B2, B3	Второй и третий байты команды
N	Номер уровня (вектор) рестарта, где $N = 0, \dots, 7$
(X)	Содержимое элемента (адреса) X, где X-регистр, ячейка памяти, регистровая пара, байты команды или бит признака
((X))	Содержимое элемента, адресуемого по содержимому элемента X. Например, ((H, L)) – содержимое ячейки памяти, адрес которой указан в N-паре регистров
0, 1	Состояние бита
\leftarrow	Оператор присваивания: элемент слева от символа « \leftarrow » заменяется элементом, находящимся справа от него
\leftrightarrow	Обмен элементами
\wedge	Логические операции И, ИЛИ, ИСКЛЮЧАЮЩЕЕ ИЛИ
\bar{X}	Операция дополнения (инверсии) значения X
+, -, *	Знаки алгебраических сложения, вычитания и умножения. В колонке «Биты признаков» знак «+» означает, что бит устанавливается в зависимости от результата операции, а знак «-» – что бит не изменяется в данной операции
n1/n2	Количество тактов, где n1 – число тактов при невыполнении условия, n2 – при его выполнении; $n1, n2 = \{4, 5, 7, 10, 11, 13, 16, 17, 18\}$

Таблица П.1.3

Коды команд MOV RM1, RM2

RM1	RM2							
	B	C	D	E	H	L	M	A
B	40 ¹	41	42	43	44	45	46	47
C	48	49 ¹	4A	4B	4C	4D	4E	4F
D	50	51	52 ¹	53	54	55	56	57
E	58	59	5A	5B ¹	5C	5D	5E	5F
H	60	61	62	63	64 ¹	65	66	67
L	68	69	6A	6B	6C	6D ¹	6E	6F
M	70	71	72	73	74	75	76 ²	77
A	78	79	7A	7B	7C	7D	7E	7F ¹

Примечание. ¹ Команда MOV R, R эквивалентна команде NOP.

² Команда MOV M, M запрещена, ее код совпадает с кодом команды HLT.

Таблица П.1.4

Коды команд с регистрами

Команда	RM							
	B	C	D	E	H	L	M	A
ADD RM	80	81	82	83	84	85	86	87
ADC RM	88	89	8A	8B	8C	8D	8E	8F
SUB RM	90	91	92	93	94	95	96	97
SBB RM	98	99	9A	9B	9C	9D	9E	9F
ANA RM	A0	A1	A2	A3	A4	A5	A6	A7
XRA RM	A8	A9	AA	AB	AC	AD	AE	AF
ORA RM	B0	B1	B2	B3	B4	B5	B6	B7
CMP RM	B8	B9	BA	BB	BC	BD	BE	BF
INR RM	04	0C	14	1C	24	2C	34	3C
DCR RM	05	0D	15	1D	25	2D	35	3D
MVI RM, B2	06	0E	16	1E	26	2E	36	3E

Таблица П.1.5

Коды команд с регистровыми парами

Команда	RR			
	B	D	H	P
LXI RR	01	11	21	31
LDAX RR	0A	1A	–	–
STAX RR	02	12	–	–
INX RR	03	13	23	33
DCX RR	0B	1B	2B	3B
DAD RR	09	19	29	39
PUSH RR	C5	D5	E5	–
POP RR	C1	D1	E1	–

СОДЕРЖАНИЕ

Введение.....	3
Лабораторная работа № 1. Команды микропроцессора КР580	9
Лабораторная работа № 2. Работа микропроцессорной системы с внешними устройствами	33
Литература	48
Приложение	49

Учебное электронное издание комбинированного распространения

Изотов Петр Павлович

МИКРОПРОЦЕССОРНАЯ ТЕХНИКА

**Лабораторный практикум
по курсу «Электроника и микропроцессорная
техника» для студентов специальностей 1-36 02 01
«Машины и технология литейного производства»,
1-36 01 05 «Машины и технология обработки
материалов давлением», 1-36 20 02 «Упаковочное
производство (по направлениям)»
дневной и заочной форм обучения**

Электронный аналог печатного издания

Редактор С. Н. Санько
Компьютерная верстка М. В. Лапицкий

Подписано в печать 07.10.2008 г.

Формат 60x84/16. Бумага офсетная. Гарнитура «Таймс».
Цифровая печать. Усл. печ. л. 3,49. Уч.-изд. л. 3,35.
Изд. № 5.

E-mail: ic@gstu.gomel.by
<http://www.gstu.gomel.by>

Издатель и полиграфическое исполнение:
Издательский центр учреждения образования
«Гомельский государственный технический
университет имени П. О. Сухого».
ЛИ № 02330/0131916 от 30.04.2004 г.
246746, г. Гомель, пр. Октября, 48.