

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

**УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ
«ВИТЕБСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ
УНИВЕРСИТЕТ»**

МИКРОПРОЦЕССОРНАЯ ТЕХНИКА СИСТЕМ АВТОМАТИЗАЦИИ

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

**к лабораторным работам для студентов
специализации 1-53 01 01-05 «Автоматизация технологических процессов и
производств (легкая промышленность)»
дневной формы обучения**

**Витебск
2013**

микроспроцессорная техника систем автоматизации.

методические указания к лабораторным работам для студентов специализации 1-53 01 01-05 «Автоматизация технологических процессов и производств (легкая промышленность)» дневной формы обучения.

Витебск: Министерство образования Республики Беларусь, УО «ВГТУ», 2013.

Составители: доц., д.т.н. Кузнецов А.А.,
ст. преп. Леонов В.В.

Методические указания составлены в соответствии с программой курса "Микропроцессорная техника систем автоматизации", изучаемого студентами дневной формы обучения специализации 1-53 01 01-05.

Методические указания содержат формулировки целей лабораторных работ, краткие сведения из теории, задания для самостоятельной и лабораторной работы, содержание отчета и контрольные вопросы.

Одобрено кафедрой «Автоматизация технологических процессов и производств» УО «ВГТУ» «10» июня 2013 г., протокол № 10.

Рецензент: проф., к.т.н. Ольшанский В.И.
Редактор: ст. преп. Клименкова С.А.

Рекомендовано к опубликованию редакционно-издательским советом УО «ВГТУ» «27» июня 2013 г., протокол № 5.

Ответственный за выпуск: Букин Ю.А.

Учреждение образования «Витебский государственный технологический университет»

Подписано к печати _____ Формат _____. Уч.-изд. л. _____
Печать ризографическая. Тираж _____ экз. Заказ № _____. Цена _____.

Отпечатано на ризографе учреждения образования «Витебский государственный технологический университет».
Лицензия № 02330/0494384 от 16 марта 2009 г.

СОДЕРЖАНИЕ

2	ОПИСАНИЕ УЧЕБНОЙ МИКРОЭВМ ДЛЯ ИЗУЧЕНИЯ ПРОЕКТИРОВАНИЯ ВЫЧИСЛИТЕЛЬНЫХ УСТРОЙСТВ НА БАЗЕ МИКРОПРОЦЕССОРОВ С ФИКСИРОВАННЫМ НАБОРОМ КОМАНД	4
3	ЛАБОРАТОРНАЯ РАБОТА № 1. Ознакомление и работа с учебной микроЭВМ	9
4	ЛАБОРАТОРНАЯ РАБОТА № 2. Запись и выполнение простых линейных программ	12
5	ЛАБОРАТОРНАЯ РАБОТА № 3. Ввод–вывод и маскирование данных в микропроцессорной системе, организация условных переходов	17
6	ЛАБОРАТОРНАЯ РАБОТА № 4. Организация стека и вызов подпрограмм в микропроцессорной системе	23
7	ЛАБОРАТОРНАЯ РАБОТА № 5. Организация программ двоичной арифметики	30
8	ЛАБОРАТОРНАЯ РАБОТА № 6. Подключение знаковосинтезирующего дисплея и клавиатуры к микроЭВМ	37
	ПРИЛОЖЕНИЯ	
	Микропроцессор серии К580 и его архитектура	46

1 ПОРЯДОК ВЫПОЛНЕНИЯ ЛАБОРАТОРНЫХ РАБОТ И

ПРОГРАММНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ

Г

«

Л

получают навыки программирования 8–разрядных микропроцессоров

Количество лабораторных работ и их распределение по семестрам определяется рабочей программой курса.

Студенты допускаются к занятиям в лаборатории после ознакомления с инструкцией по технике безопасности.

Для работы в лаборатории учебная группа разбивается на бригады из 2...3 человек, которые выполняют лабораторную работу.

Перед выполнением каждой лабораторной работы проводится проверка теоретической подготовки студентов по данной теме.

После проверки подготовки к работе и получения у преподавателя разрешения студенты приступают к выполнению лабораторной работы. Работа выполняется в строгом соответствии с настоящими методическими указаниями.

Воспрещается оставлять без надзора включенные под напряжение лабораторный стенд и измерительные приборы.

При оформлении отчета по лабораторным работам следует:

– на титульном листе указать наименование лабораторной работы, номер учебной группы, Ф. И. О. студента;

– использовать стандартные листы писчей бумаги;

– схемы и графический материал вычерчиваются карандашом с помощью чертежных инструментов, с применением стандартных графических обозначений элементов схем;

– временные диаграммы вычерчиваются на миллиметровой бумаге.

Студент обязан готовиться к каждому очередному занятию по методическим указаниям и литературе, указанной в описании лабораторной работы. В результате подготовки необходимо знать теоретические основы курса в объеме выполняемой лабораторной работы, четко представлять ее цель и последовательность выполнения, схему учебной микроЭВМ, а также принцип действия и назначение всех приборов, применяемых в данной лабораторной работе.

2 ОПИСАНИЕ УЧЕБНОЙ МИКРОЭВМ ДЛЯ ИЗУЧЕНИЯ ПРОЕКТИРОВАНИЯ ВЫЧИСЛИТЕЛЬНЫХ УСТРОЙСТВ НА БАЗЕ МИКРОПРОЦЕССОРОВ С ФИКСИРОВАННЫМ НАБОРОМ КОМАНД

Учебная микроЭВМ (рисунок 1) предназначена для знакомства с особенностями построения микропроцессорных систем с фиксированным набором команд. Она может быть использована для исследования методов программирования и работы БИС, входящих в микропроцессорный комплект серии КР580, как наиболее распространенных и базовых для первичного изучения и знакомства с микропроцессорными средствами. Учебная микроЭВМ может применяться в качестве управляющей при создании и исследовании работы систем управления различными объектами. Она является легко осваиваемым и удобным средством для отладки относительно небольших (до 1 Кбайт) программ пользователя. Открытая конструкция и наличие большого количества средств индикации позволяют наглядно исследовать процесс преобразования и передачи информации в микроЭВМ. В конструкцию стенда входит внутренний источник питания, обеспечивающий ее работу от сети 220 В, 50 Гц.

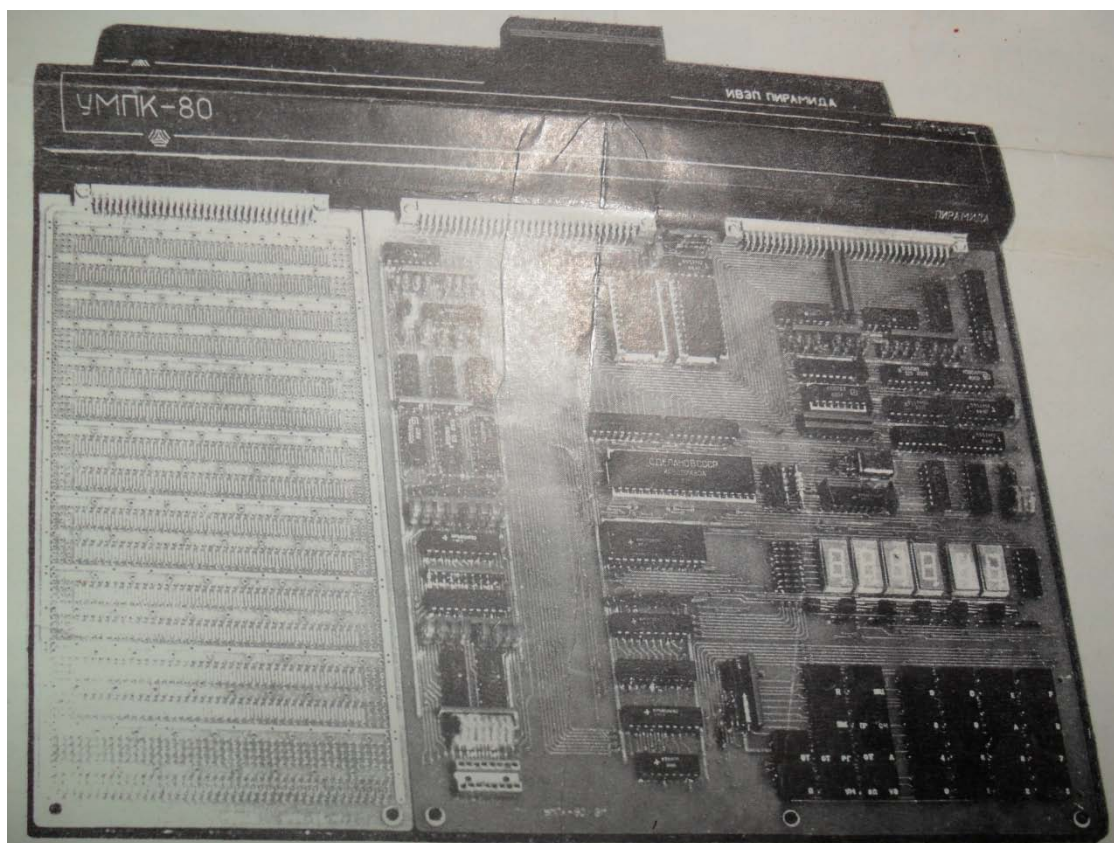


Рисунок 1 – Учебная микроЭВМ

2.1 СТРУКТУРА УЧЕБНОЙ МИКРОЭВМ

На рисунке 2 приведена структура микроЭВМ, где показаны ее отдельные блоки. Схема состоит из блока центрального процессора со схемой тактового питания СхТП; формирователей магистралей данных ФМД, управления ФМУ, адреса ФМА микроЭВМ с магистралями.

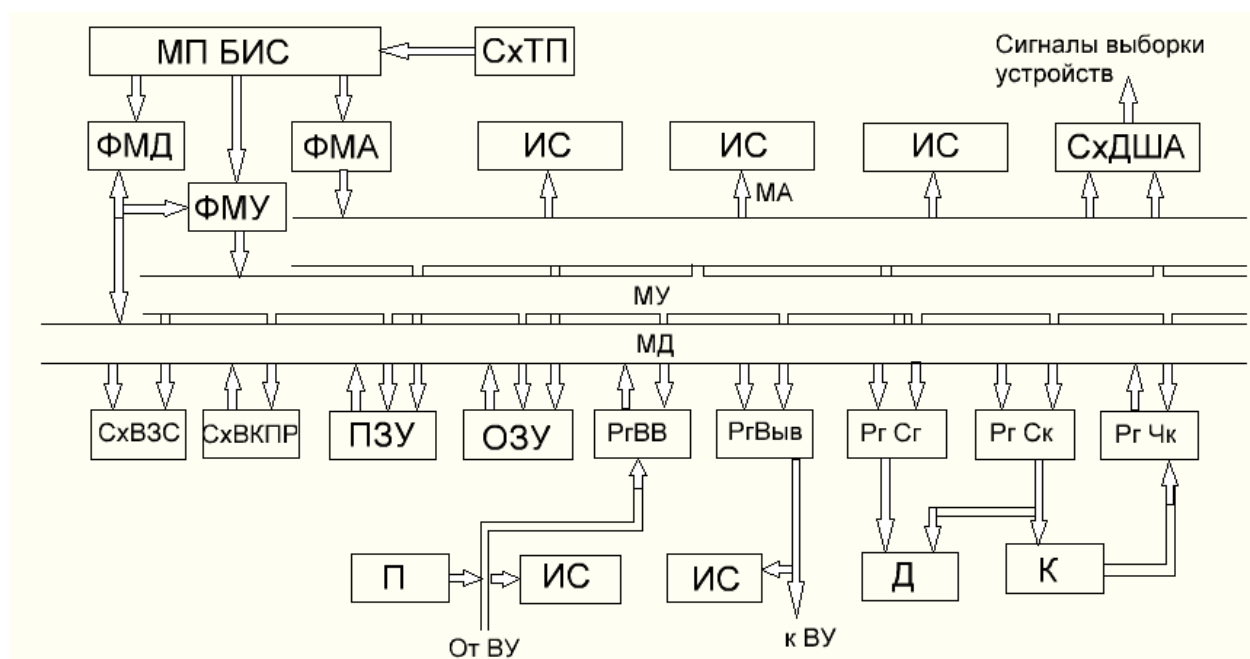


Рисунок 2 – Структура микроЭВМ

- 8–разрядной магистрали данных МД для двунаправленного обмена данными между отдельными узлами микроЭВМ;
- 16–разрядной однонаправленной магистрали адреса МА, управляемой МП БИС и используемой для адресации памяти и отдельных узлов микроЭВМ;
- 5–разрядной однонаправленной магистрали управления МУ, управляемой МП БИС и служащей для определения режима работы микроЭВМ при выполнении каждого машинного цикла;
- блока памяти БП, включающего в себя оперативное запоминающее устройство ОЗУ емкостью 1 Кбайт и постоянное запоминающее устройство ПЗУ емкостью 2 Кбайт;
- регистров ввода РгВВ с переключателями П и вывода РгВыв с индикацией их состояний светодиодами;
- схемы выдачи кода прерывания СхВКПР;
- клавиатуры управления и ввода данных К, состоящей из 9 клавиш управления, переключателя начальной установки микроЭВМ и 16 клавиш для ввода шестнадцатиричных кодов чисел. Для ввода информации в микроЭВМ клавиатура имеет регистр чтения клавиатуры РгЧК;

– 6–разрядный восьмисегментный дисплей Д, четыре разряда которого служат для отображения кода адреса в шестнадцатиричной системе исчисления, а два – для отображения данных, содержащихся по выводимому адресу. Для вывода информации дисплей Д имеет два регистра: регистр сегментов дисплея РгСг и регистр сканирования РгСк и схемы выдачи звукового сигнала СхВЗС. Магистралы учебной микроЭВМ, а также входной и выходной регистры имеют светодиоды ИС, указывающие их состояние.

2.2 РАСПРЕДЕЛЕНИЕ АДРЕСНОГО ПРОСТРАНСТВА УЧЕБНОЙ МИКРОЭВМ

Каждому физическому устройству микроЭВМ ставится в соответствие адрес, и обращение к нему при таком типе адресации аналогично обращению к ячейке памяти с использованием всего набора команд МП БИС.

В таблице 1 приведена карта памяти микроЭВМ, из которой видно, что первые 2 Кбайт адресов занимает ПЗУ, в котором записаны управляющие и демонстрационные программы; адреса с 0800h по 0BFFh отведены ОЗУ; адрес 0B00h является начальным адресом стековой области памяти, а адреса с 0B10h по 0BFFh задействованы для временной записи данных во внутренние регистры МП БИС при работе управляющей программы. Адрес 0800h является также начальным адресом ОЗУ, куда могут записываться исследуемые программы пользователя. Большинство из приведенных в лабораторном практикуме программ используют эту область ОЗУ, которая по адресу 0AFFh в учебной микроЭВМ схемотехнически защищена от случайной записи во время выполнения программ пользователя. Для записи данных при выполнении программ необходимо использовать область ОЗУ с адреса 0B00h по адрес 0BFFh. Знак h после числового представления адреса и данных в дальнейшем означает использование шестнадцатиричной системы счисления.

Таблица 1 – Карта памяти микроЭВМ

Адреса ячеек памяти и портов ввода–вывода	Назначение
1	2
0000h– 07FFh	Управляющая программа
0800h – 0AFFh	Область адресов для записи исследуемых программ
0B00h – 0B00h	Для записи данных исследуемых программ и стек
0B10h–0BFFh	Для записи данных управляющей программы
0C00h – 0FFFh	Резерв

Окончание таблицы 1

1	2
04h	Регистр звукового выхода
05h	Входной регистр
05h	Выходной регистр
06h	Регистр сегментов дисплея
06h	Входной регистр чтения клавиатуры
07h	Регистр сканирования дисплея и клавиатуры

2.3 РЕЖИМЫ РАБОТЫ И АЛГОРИТМ УПРАВЛЯЮЩЕЙ ПРОГРАММЫ УЧЕБНОЙ МИКРОЭВМ

Управляющая программа состоит из программ тестирования отдельных узлов микроЭВМ, начальной установки содержимого внутренних регистров МП БИС и ячеек ОЗУ, сохранения содержимого внутренних регистров МП БИС, выдачи сообщения на дисплей, чтения и декодирования клавиатуры, обслуживания звукового выхода и программ, обеспечивающих следующие режимы:

1. Ожидания ввода команд управления с клавиатуры.
2. Отображения на дисплее адреса любой ячейки и ее содержимого.
3. Ввода с клавиатуры четырехразрядного шестнадцатиричного кода любого адреса с одновременным отображением его на дисплее и, после ввода последней его цифры, автоматического перехода к выполнению режима 2 (начало режима ввода кода адреса с клавиатуры инициируется нажатием клавиши ОТА – отыскание адреса).
4. Обращения к программному счетчику. Выполнение этого режима вызывается нажатием клавиши ПРСч (программный счетчик) и приводит к автоматическому выводу адреса программного счетчика на дисплей с последующим переходом к режиму 2.
5. Вывода на дисплей содержимого внутренних программно–доступных регистров МП БИС с указанием на дисплее названия выводимого регистра. Этот режим устанавливается нажатием клавиши ОTRг (отыскание регистра), при этом на дисплей будет выведено содержимое аккумулятора МП БИС. Последующие нажатия на клавишу ЗпУВ (запись увеличить) приведут к циклическому выводу содержимого всех внутренних регистров МП БИС. Содержимое регистров будет выводиться на дисплей в последовательности: А, FL, В, С, D, E, H, L, SPH, SPL, PCH, PCL.
6. Записи с клавиатуры в ячейки ОЗУ и внутренние программно доступные регистры МП БИС нового кода. При работе микроЭВМ в режимах 2

и 5 после изменения кода чисел, представленных на дисплее данных, запись чисел инициируется нажатием на клавишу ЗпУВ.

7. Увеличения или уменьшения на единицу адреса, представленного на дисплее при работе микроЭВМ в режиме 2. Эти режимы инициируются соответственно нажатием клавиш ЗпУВ и УМ (уменьшить).

8. Запуска программы с адреса, указанного на дисплее, с помощью клавиши П (пуск).

9. Выполнения команд по машинным циклам с отображением информации на магистральных микроЭВМ с помощью светодиодов. Режим устанавливается нажатием клавиши ШЦ (шаг машинного цикла).

10. Выполнения программы по командам. Режим устанавливается нажатием клавиши ШК (шаг команды), при этом после каждой команды управляющая программа автоматически переходит к режиму 4.

11. Остановка программы пользователя. Режим устанавливается при нажатии клавиши СТ (стоп) с автоматическим сохранением в ОЗУ содержимого всех внутренних регистров МП БИС и переходом к выполнению режима 4. Сохранение состояния регистров позволяет анализировать их содержание на момент останова с помощью перехода к режиму 5. После останова программы микроЭВМ может быть переведена на режимы 9 и 10.

12. Возврата микроЭВМ из режима выполнения команд по машинным циклам на управляющую программу с автоматическим восстановлением содержимого внутренних регистров МП БИС и переходом к режиму 4.

Работа микроЭВМ организована таким образом, что после подачи на нее питания запускается программа тестирования всех основных узлов. Эта программа начинается с ячейки ПЗУ, имеющей адрес 0000h. В результате выполнения теста проверяются:

- правильность записанной в ПЗУ информации и возможность ее верного считывания;

- функционирование ОЗУ – на отсутствие ошибок при записи/считывании чисел;

- МП БИС – на правильность выполнения им простейших арифметических и логических операций;

- запись чисел в регистр вывода и работоспособность всех его светодиодов состояния, а также работа всех сегментов индикаторов дисплея.

В случае обнаружения ошибок в работе ОЗУ или ПЗУ подается прерывистый звуковой сигнал и на дисплей выводится соответствующее сообщение: ОЗУ или ПЗУ. При успешном завершении теста микроЭВМ производит начальную установку всех регистров МП БИС и на дисплей выводится сообщение НАЧАЛО, свидетельствующее о готовности ее к работе. После инициализации микроЭВМ переходит в режим ожидания ввода команд управления и может реагировать на нажатие трех клавиш: ОТА (отыскание адреса), ОTRг (отыскание регистра) и ПРСч (программный счетчик), что соответствует режиму 1. Для останова работы исследуемой программы или при

возврате на управляющую программу при работе микроЭВМ в режиме выполнения программы по машинным циклам используется клавиша СТ (стоп). Нажатие этой клавиши в указанных режимах приводит к останову режима, при этом на дисплей выводится содержимое программного счетчика МП БИС. Нажатие клавиши СТ при другом режиме приведет к переходу микроЭВМ в режим ожидания ввода команд управления, при этом на дисплей будет выводиться сообщение НАЧАЛО.

Для удобства отладки программ в микроЭВМ предусмотрена возможность останова выполнения программ по любым заранее заданным адресам с помощью одного из векторов системы прерывания.

Программы, записанные в оставшейся части ПЗУ, можно условно разбить на два типа:

1. Вспомогательные подпрограммы выполнения арифметических и логических операций (например, умножение, деление двух восьмибитных чисел и т. д.), которые могут применяться пользователем при создании более сложных программ.

2. Демонстрационные подпрограммы (например, подпрограмма, используемая для записи и воспроизведения с помощью звукового выхода музыкальных произведений по нотам, коды которых можно хранить в виде чисел в ПЗУ или ОЗУ, подпрограмма "секундомер" и т. д.).

Имеется открытый доступ к магистралям микроЭВМ, что позволяет исследовать ее совместную работу с БИС, входящими в микропроцессорный набор серии К580 в виде дополнительных модулей расширения стенда.

Для успешного выполнения лабораторных работ необходимо предварительное ознакомление с языком программирования и командами МП БИС КР580ВМ80, представленными в Приложении.

3 Лабораторная работа № 1

ОЗНАКОМЛЕНИЕ И РАБОТА С УЧЕБНОЙ МИКРОЭВМ

Цель работы: ознакомиться со структурой учебной микроЭВМ, распределением адресного пространства, органами управления и режимами работы.

После включения учебной микроЭВМ или нажатия клавиши начальной установки R содержимое программного счетчика МП БИС обнуляется, и выполнение программы начинается с нулевой ячейки памяти, где записаны программы тестов всех узлов микроЭВМ.

После выполнения тестов производится начальная установка всех внутренних программно-доступных регистров МП БИС. Начальная установка состоит в том, что в программный счетчик записывается число 0800h (первый адрес ОЗУ), в указатель стека число 0ВВ0h, а остальные регистры МП БИС обнуляются.

Задания для домашней подготовки

1. Ознакомьтесь с описанием учебной микроЭВМ.
2. Ознакомьтесь с типовой минимальной структурой микроЭВМ, методами организации магистралей, подключения памяти и внешних устройств к магистралям.
3. Изучите алгоритм работы управляющей программы стенда.
4. Изучите внутренние регистры МП БИС КР580ВМ80 и их назначение.

Задания к лабораторной работе

Задание 1. Исследовать порядок включения и процессы, происходящие при включении микроЭВМ.

1. Подключить шнур питания к сети.
2. Включить тумблер СЕТЬ.

В результате выполнения тестовых программ светодиоды выходного регистра и сегменты дисплея будут включены на время 2 с, после чего микроЭВМ подает звуковой сигнал и выводит на дисплей сообщение НАЧАЛО. Находясь в этом режиме, микроЭВМ реагирует на нажатия клавиши управления. Из любого места управляющей программы можно вернуть микроЭВМ к начальному состоянию нажатием на клавишу R. В этом случае микроЭВМ начинает выполнять управляющую программу с нулевой ячейки памяти. В результате ее выполнения все ячейки ОЗУ обнуляются, что приводит к уничтожению программы пользователя, записанной ранее в ОЗУ.

3. Нажать на клавишу R и убедиться, что тестовые программы проходят заново.

Задание 2. Исследовать содержимое памяти.

1. Нажать на клавишу ОТА, при этом на дисплее появится сообщение начала выполнения режима отыскания адреса (включены нижние сегменты дисплея).

2. Последовательно нажать на клавиши 0, 8, 0, 0. Убедиться при этом, что каждая цифра будет записана в младший разряд адресного дисплея и произойдет одновременный сдвиг всех знаков на адресном дисплее на один разряд влево. При введении четвертой цифры с клавиатуры микроЭВМ выведет на дисплей число, записанное по этому адресу. На дисплее данных появится число 00. В случае ошибки при вводе кода адреса следует нажать на клавишу ОТА и повторить ввод. Все вводимые с клавиатуры и выводимые на дисплей числа отображаются в шестнадцатеричной системе счисления.

3. Нажать на клавишу ЗпУВ. В этом режиме микроЭВМ увеличит на единицу адрес на адресном дисплее и выведет содержимое указанной ячейки памяти на дисплей данных. Последовательно нажимая на клавишу ЗпУВ, проверить содержимое нескольких ячеек адресного пространства ОЗУ (начиная с адреса 0800h).

4. Нажать на клавишу ОТА и ввести код 0000 (первый адрес ПЗУ). На дисплее данных появится код 26. Последовательно нажимая на клавишу ЗпУВ, просмотреть содержимое нескольких последовательно расположенных ячеек ПЗУ.

5. Убедиться, что содержимое памяти можно просмотреть с помощью последовательного нажатия на клавишу УМ в обратном порядке. Таким образом, использование этого режима позволяет проверить содержимое всех ячеек памяти микроЭВМ.

Задание 3. Записать числа в память микроЭВМ.

1. Включить тумблер СЕТЬ (или при включенной ранее микроЭВМ нажать на клавишу R).

2. Нажать на клавишу ОТА и после этого набрать адрес 0800h. На дисплее данных после ввода адреса появится его содержимое 00h.

3. Нажать на клавишу 1. МикроЭВМ вводит значение цифры нажатой клавиши в младший разряд дисплея данных, при этом в этом разряде дисплея появляется десятичная точка. Она свидетельствует о том, что микроЭВМ находится в режиме ввода данных и число, представленное на дисплее данных, не является истинным содержимым адреса памяти, представленного на адресном дисплее.

4. Нажать на клавишу 2. Убедиться при этом, что цифра 1 младшего разряда дисплея переместилась на старший разряд дисплея данных, а цифра 2 на ее место (десятичная точка осталась в младшем разряде). Проверить, что последующие нажатия на цифровые клавиши приводят к вводу цифры нажатой клавиши в младший разряд дисплея данных и смещению в старший разряд предыдущей введенной цифры.

5. Записать нажатием на клавишу ЗпУВ число, представленное на дисплее по адресу 0800h. При этом на дисплее появится адрес 0801h и его содержимое 00h, а десятичная точка погаснет, показывая, что данные введены и микроЭВМ вышла из режима ввода данных.

6. Записать любое число на дисплей данного при наличии на дисплее адреса 0801h.

7. Нажать на клавишу УМ. На адресном дисплее появится адрес 0800h с его содержимым. Убедиться при этом, что число было записано по этому адресу. Отсутствие десятичной точки в младшем разряде дисплея свидетельствует о том, что это число было извлечено из памяти.

8. Нажать на клавишу ЗпУВ и проверить содержимое адреса 0801h. Убедиться при этом, что число, записанное на дисплее данного, не было записано в память при нажатии на клавишу УМ.

9. Осуществить вывод на дисплей содержимого адреса 0000h (ПЗУ). На дисплее будет 26h. Записать по этому адресу на дисплее любое другое число. Убедиться, что при нажатии на клавишу ЗпУВ запись данных в ПЗУ невозможна и при этом микроЭВМ определяет это действие как ошибку, формирует звуковой сигнал, не увеличивает адрес и выводит на дисплей истинное число, записанное в ПЗУ.

Задание 4. Записать числа в программно-доступные регистры МП БИС.

1. Включить микроЭВМ. После вывода на дисплей сообщения НАЧАЛО нажать на клавишу ОTRг. На дисплее будет выведено содержимое аккумулятора (А) МП БИС. Многократно нажимая на клавишу ЗпУВ, проверить последовательность вывода содержимого внутренних регистров МП БИС на дисплей.

2. Изменить число, записанное в регистре при наличии на дисплее информации о его содержимом, с помощью цифровых клавиш клавиатуры. Нажать на клавишу ЗпУВ.

3. Нажать на клавишу УМ и убедиться при этом в правильности записи числа в регистр.

Задание 5. Осуществить пуск выполнения программы.

1. Для осуществления пуска программы с любого адреса памяти вывести этот адрес на дисплей. Пуск программы производится нажатием на клавишу П (пуск), при этом в программный счетчик МП БИС записывается адрес, указанный на дисплее, и выполнение программы начинается с этого адреса. Записать на дисплей адрес 05B0h, являющийся началом музыкальной программы, записанной в ПЗУ.

2. Нажать на клавишу П, при этом микроЭВМ исполнит мелодию по нотам, записанным в ПЗУ.

3. Убедиться, что выполнение программы может быть остановлено нажатием на клавишу СТ. Для этого повторить пуск музыкальной программы

заново и при выполнении программы нажать на клавишу СТ. При останове программы на дисплей будет выводиться адрес, записанный в программном счетчике, и его содержимое на момент останова. При останове программы проверить возможность осуществления режима вывода на дисплей содержимого регистров МП БИС на момент останова, для чего следует нажать на клавишу ОTRг, при этом на дисплее появится содержимое аккумулятора МП БИС.

4. Нажатием на клавишу ПРСч вывести на дисплей адрес памяти, на котором было прервано выполнение программы. Нажать на клавишу П (пуск) и продолжить выполнение программы.

Отчет должен содержать:

1. Программную модель микропроцессора КР580ВМ80.
2. Распределение адресного пространства.
3. Информацию о содержании внутренних программно-доступных регистров МП БИС после программы начальной установки микроЭВМ.

Задания для самопроверки

1. Что такое карта памяти микроЭВМ?
2. Какие адреса памяти микроЭВМ относятся к ОЗУ и ПЗУ?
3. Расскажите о возможных режимах работы микроЭВМ.
4. Как записать числа в программно-доступные регистры МП БИС или память микроЭВМ?
5. Что происходит в микроЭВМ при попытке записи данных в ПЗУ?
6. Какие узлы проверяются в микроЭВМ в процессе выполнения тестов при ее включении или нажатии на клавишу R?
7. Укажите возможные изменения режимов работы микроЭВМ при выполнении программы по командам.
8. Опишите функции каждой клавиши управления на клавиатуре.

Литература

1. Основы проектирования микропроцессорных устройств автоматики. / Б. М. Каган, В. В. Сташин. – Москва: Энергоатомиздат, 1987.
2. Программы для микропроцессоров. Справ. пособие. / А. Л. Гуртовцев, С. В. Гудыменко. – Минск: Вышэйшая школа, 1989.

4 Лабораторная работа № 2

ЗАПИСЬ И ВЫПОЛНЕНИЕ ПРОСТЫХ ЛИНЕЙНЫХ ПРОГРАММ

Цель работы: исследовать выполнение отдельных команд и простых программ, использовать различные методы адресации в программах, запись программ.

Микропроцессор КР580ВМ80 имеет фиксированный набор команд. Время выполнения команды определяется процессом получения, декодирования и выполнения команды. Это время можно представить состоящим из ряда временных интервалов. Наиболее короткий – временной интервал, равный периоду синхронизации МП БИС, называется машинным тактом. Время, необходимое для извлечения 1 байт информации из памяти или внешнего устройства выполнения команды, определяемой одним машинным словом, называется машинным циклом. Машинный цикл для МП БИС может включать в себя до 5-ти машинных тактов. В зависимости от вида команды время ее выполнения может состоять из 1–5 машинных циклов. Для МП БИС имеется 10 различных типов машинных циклов: извлечение кода команды (цикл М1); чтение данных из памяти; запись данных в память, извлечение данных из стека; запись данных в стек; ввод данных из внешнего устройства; запись данных во внешнее устройство; цикл обслуживания прерывания; останов; обслуживание прерывания в режиме останова. Первым машинным циклом при извлечении любой команды является цикл М1. На каждом машинном цикле МП БИС проверяет состояние сигнала "ГОТОВ" на своем входе. Нулевой сигнал на этом входе приостанавливает нормальную работу МП БИС, при этом на магистралях микроЭВМ присутствует вся информация, передаваемая на рассматриваемом машинном цикле. В учебной микроЭВМ это используется для исследования выполнения команд по машинным циклам. В этом режиме информация на магистралях микроЭВМ отображается светодиодами состояния. Программа записывается в микроЭВМ в последовательных ячейках памяти.

Рассмотрим простейшую программу (программа 2.1), извлекающую число из ячейки памяти по адресу ОВ00h, инвертирующую его и записывающую результат в ячейку памяти по адресу ОВ01h.

Программа 2.1 (в мнемосокодах)

Мнемосокод	Комментарий
LDA ОВ00h	Извлечь число из ячейки памяти с адресом ОВ00h
СМА	Инвертировать число
STA ОВ01h	Записать результат в ячейку памяти по адресу ОВ01h
RST1	Прервать выполнение программы

При записи программ все числа представляются в шестнадцатиричной системе счисления.

Для записи программы в память микроЭВМ необходимо перевести мнемкоды команд в машинные коды. Команды в программе могут быть одно-, двух- или трехбайтные и должны в памяти занимать, соответственно, одну, две или три ячейки памяти.

Программа 2.1 (размещение по адресам памяти)

Адрес	Число	Комментарий
0800	3A	Код команды LDA
0801	00	Младший байт адреса
0802	0B	Старший байт адреса
0803	2F	Код команды CMA
0804	32	Код команды STA
0805	01	Младший байт адреса
0806	0B	Старший байт адреса
0807	CF	Код команды RST 1

Предварительную запись программ удобно проводить в более компактной форме. В программе указывается начальный адрес каждой команды и при этом понимается, что в зависимости от длины (одно-, двух- или трехбайтная) команды в памяти будут занимать от одной до трех последовательных ячеек. При такой записи в левом столбце указываются лишь адреса команд в программе. Это позволяет сократить объем при описании программ и сделать более простым их анализ.

Программа 2.1 (общий вид записи)

Адрес	Машинный код	Метка	Мнемкокод	Комментарий
0800	3A 000B		LDA 0B00h	Извлечь число из ячейки памяти с адресом 0B00h
0803	2F		CMA	Инвертировать число
0804	32 010B		STA 0B01h	Записать результат в ячейку памяти по адресу
0807	CF		RST 1	Прервать выполнение

Здесь используется прямой способ адресации.

Рассмотрим программу, аналогичную программе 2.1, с использованием косвенного способа адресации (программа 2.2).

Программа 2.2

Адрес	Машинный код	Метка	Мнемокод	Комментарий
0800	21 000B		LXIH, OBOOh	Записать в регистровую пару HL число OBOOh
0803	7E		MOV A, M	Извлечь число из ячейки памяти с адресом, указанным в HL
0804	2F		CMA	Инвертировать число в аккумуляторе
0805	23		INX M	Увеличить на 1 число в регистровой паре HL
0806	77		MOV M, A	Записать число из аккумулятора по адресу, указанному в HL
0807	CF		RST 1	Прервать выполнение программы

Задания для домашней подготовки

1. Ознакомьтесь с языком программирования и структурой команд МП БИС KP580BM80.

2. Изучите методы программирования на языке ассемблера и в машинных кодах для МП БИС KP580BM80.

3. Рассмотрите правила выполнения команд INRA (3C), DCRA (3D), ADDA (87), ANAA (A7), ORAA (B7), CMPA (BF), DAA(21).

4. Рассмотрите результат выполнения программы 2.1 при записи по адресу 0803h команд, приведенных в п. 3 задания. Результаты выполнения программы при различных командах, записанных в программе 2.1 по адресу 0803h, занести в таблицу 2.1.

Таблица 2.1

Число, записанное по адресу OBOOh	Команда, записанная по адресу 0803h	Число, записанное по адресу OBOlh
-----------------------------------	-------------------------------------	-----------------------------------

5. Видоизмените и запишите программу 2.2 так, чтобы при ее выполнении исследуемое число первоначально записывалось по адресу OBOOh.

6. Разработайте программы: а) увеличения на 5 числа, записанного по адресу OBOOh, и записи результата по адресу OBOlh (программа 2.3); б) сложения чисел, записанных по адресам OBOOh и OBAOh, и записи результата по адресу OBOlh (программа 2.4); в) сравнения чисел в адресах OBOOh и OBOlh и записи большего из них в регистр В (программа 2.5).

Задания к лабораторной работе

Задание 1. Исследовать программу 2.1.

1. Ввести в микроЭВМ программу 2.1.
2. Записать по адресу OBOOh исследуемое число.
3. Осуществить пуск программы 2.1 с адреса 0800h. Проверить результаты выполнения программы путем исследования числа, записанного по адресу OBOlh.

4. Исследовать процесс выполнения программы по командам. После выполнения каждой команды проанализировать содержимое всех программно-доступных регистров МП БИС.

5. Исследовать процесс выполнения команд в программе 2.1 по машинным циклам. Обратит внимание на последовательность передачи и преобразования информации в микроЭВМ при выполнении каждой команды.

6. Заменяя в программе 2.2 команду CMA на команды INRA (3C), DCRA (3D), ADDA (87), ANAA (A7), ORAA (B7), CMPA (BF), DAA(27), исследовать результат выполнения указанных команд по числу, записанному по адресу OBOlh. Проверить таблицу 2.1, заполненную при выполнении домашнего задания.

Задание 2. Исследовать программу 2.2.

1. Ввести в микроЭВМ программу 2.2.
2. Записать по адресу OBOOh исследуемое число.
3. Осуществить пуск программы с адреса 0800h. Проверить результат выполнения программы по числу, записанному по адресу OBOlh.

4. Исследовать процесс выполнения команды MOV A, M по машинным циклам.

5. Ввести и исследовать выполнение микроЭВМ видоизмененной программы 2.2, позволяющей первоначально записывать исследуемое число по адресу OBOOh.

Задание 3. Исследовать программу 2.3.

1. Ввести в микроЭВМ программу 2.3, разработанную при домашней подготовке.

2. Осуществить пуск программы 2.3 и проверить результат ее выполнения по числу, записанному по адресу OBOlh при числах 05h, FEh, записанных по адресу OBOOh.

Задание 4. Исследовать программу 2.4.

1. Ввести в микроЭВМ программу 2.4, разработанную при домашней подготовке.

2. Осуществить пуск программы 2.4 и проверить результат ее выполнения по числу, записанному по адресу $0B01h$, последовательно записывая по адресам $0B00h$ и $0BA0h$, соответственно, числа $0Bh$ и $B0h$, FEh и $B5h$ осуществляя пуск программы.

3. Видоизменить и исследовать программу 2.4 для случая, когда сумма двух чисел будет превышать восьмиразрядное число.

Задание 5. Исследовать программу 2.5.

1. Ввести в микроЭВМ программу 2.5, разработанную при домашней подготовке.

2. Записать по адресам $0B00h$ и $0B01h$ исследуемые числа.

3. Осуществить пуск программы с адреса $0800h$. Проверить результат выполнения программы по числу, записанному в регистре В.

Отчет должен содержать:

1. Заполненную таблицу 2.1 для случаев выполнения программы 2.1 при использовании команд, представленных в п. 4 задания для домашней подготовки.

2. Видоизмененную программу 2.2, записанную в машинных кодах, позволяющую первоначально записывать исследуемое число по адресу $0B00h$.

3. Разработанные в процессе домашней подготовки программы 2.3, 2.4, 2.5, записанные в машинных кодах; результаты исследований работы программы в соответствии с п. 3, 4, 5 заданий.

Задания для самопроверки

1. За сколько машинных тактов выполняется каждая команда в программах 2.1 и 2.2?

2. Укажите все возможные способы адресации, используемые при составлении программы по п. 6 а задания для домашней подготовки.

3. При выполнении каких команд, приведенных в п. 4 задания для домашней подготовки, используются разряды регистра признаков МП БИС?

4. Сформулируйте правила выполнения МП БИС команд, приведенных в п. 4 задания для домашней подготовки.

Литература

1. Микропроцессоры. Основы построения микроЭВМ./ В.Л. Горбунов, Д.И. Панфилов, Д.Л. Преснухин. – Москва: Высшая школа, 1984.

2. Микропроцессоры и микроЭВМ./ И. В. Прангишвили. – Москва: Энергия, 1979.

5 Лабораторная работа №3

ВВОД – ВЫВОД И МАСКИРОВАНИЕ ДАННЫХ В МИКРОПРОЦЕССОРНОЙ СИСТЕМЕ И ОРГАНИЗАЦИЯ УСЛОВНЫХ ПЕРЕХОДОВ

Цель работы: исследовать методы подключения и организации обмена информацией с простейшими устройствами ввода-вывода. Изучить программные способы маскирования данных и организации условных переходов в микроЭВМ.

К командам ввода-вывода МП БИС КР580ВМ80 относятся команды IN(A1) и OUT(A1). При выполнении команды IN(A1) микроЭВМ считывает число из входного устройства с адресом (A1) и записывает его в аккумулятор. При выполнении команды OUT(A1) МП БИС записывает число из аккумулятора в выходное устройство с адресом (A1). С помощью этих команд микроЭВМ может обмениваться информацией не более чем с 256 внешними устройствами.

В качестве простейших устройств ввода-вывода могут использоваться 8-разрядные регистры (например, КР1533ИР23, КР1533ИР37). Обмен данными между микроЭВМ и внешними устройствами может производиться как в определенных местах в программе, так и по сигналам прерывания. В последнем случае подпрограмма обмена данными с внешними устройствами будет вызываться за счет перевода микроЭВМ в режим обслуживания прерывания. В качестве устройств ввода-вывода могут применяться и более сложные схемы, например программируемое устройство ввода-вывода информации в параллельном коде (КР580ВВ55).

Схема подключения к микроЭВМ входного устройства, выполненного на базе ИМС КР1533АП6 с переключателями, приведена на рисунке 3.1. При замкнутых переключателях на входы ИМС подается "0", а при разомкнутых "1". Переключатели используются для имитации передачи данных от внешнего устройства. К входному устройству подключены светодиоды (Н1 – Н8) для индикации чисел, поступающих в него. На рисунке 3.2 приведена схема подключения выходного устройства к микроЭВМ, построенная на базе схемы КР1533ИР37. Светодиоды Н1 – Н8 указывают число, записанное в выходном устройстве.

Простейшая программа (программа 3.1) перезаписи числа со входного устройства (с адресом 05h) в выходное устройство (также с адресом 05h) имеет вид:

Программа 3.1

Адрес	Код	Метка	Мнемокод	Комментарий
0800	DB05	CNT:	IN 05h	Записать число из входного устройства с адресом 05h в аккумулятор
0802	D3 05		OUT 05h	Записать число из аккумулятора в выходное устройство с адресом 05h
0804	OUT 05h		JMP CNT	Переход в точку программы с меткой CNT

Во многих случаях при выполнении программ необходимо проверять или изменять (маскировать) состояние одного или нескольких разрядов числа в аккумуляторе. Это можно осуществить с помощью следующих команд логических операций:

1. Логического умножения числа в аккумуляторе и маски, которое очищает разряд числа, если в соответствующем разряде маски будет записан "0", и не изменяет его, если в разряде маски записана "1".

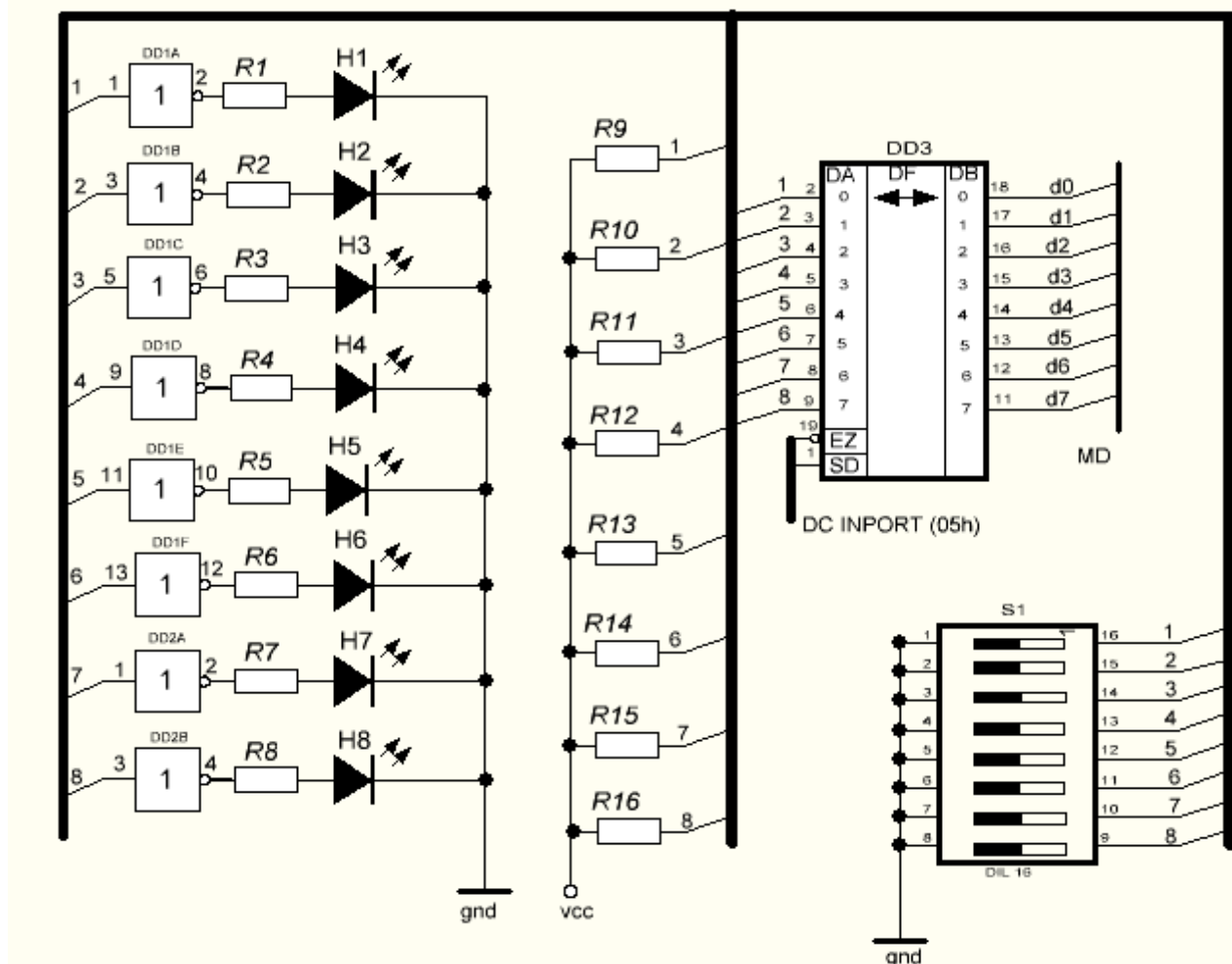


Рисунок 3.1 – Схема подключения к микроЭВМ входного устройства

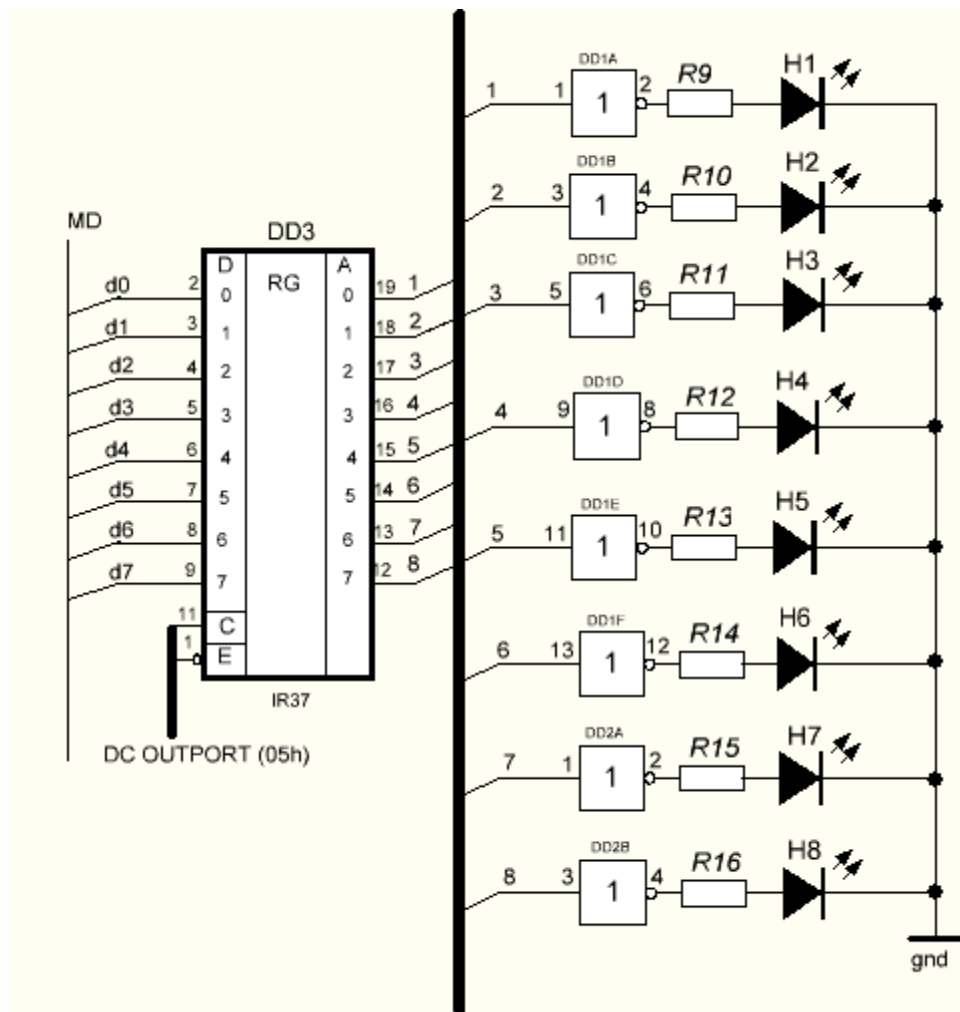


Рисунок 3.2 – Схема подключения выходного устройства к микроЭВМ

2. Логического сложения числа в аккумуляторе и маски, которое устанавливает разряд числа в "1", если в таком же разряде маски будет записана "1", и не изменяет его, если в этом разряде записан "0".

3. Логического "исключающего ИЛИ" числа в аккумуляторе и маски, которое инвертирует содержание разряда числа, если в соответствующем разряде маски записана "1", и не изменяет его, если в этом разряде записан "0".

Проведение логических операций возможно также с содержимым аккумулятора и внутренними регистрами МП БИС. В этом случае команды однобайтные. При выполнении всех логических команд задействуются разряды Z, S, P, AC регистра признаков (в разряд CY записывается 0), назначение которых приведено в Приложении. Это позволяет проверять состояние любого разряда числа, записанного во входном устройстве.

Программа 3.2, приведенная ниже, помещает результат маскирования в выходное устройство.

Программа 3.2

Адрес	Код	Метка	Мнемокод	Комментарий
0800	DB05	CNT:	IN 05h	Записать число из входного устройства с адресом 05h в аккумулятор
0802	E6 20		ANI 20h	Выполнить логическую операцию
0804	D3 05		OUT 05h	Записать число из аккумулятора в выходное устройство с адресом 05h
0806	C3 00 08		JMP CNT	Переход в точку программы с меткой CNT

Условные переходы организуются в программах с помощью команд условных переходов. При выполнении этих команд МП БИС проверяет состояние соответствующего разряда регистра признаков. Если при проверке состояния разряда регистра признаков условие не подтверждается, то выполняется следующая по порядку команда программы. Все команды условных переходов трехбайтные: первый байт содержит код команды, второй и третий байты – адрес передачи управления. Команды условных переходов позволяют строить ветвящиеся алгоритмы и в зависимости от текущего значения результата выполнения программы переходить на различные участки программы.

Программа (программа 3.3) служит для определения "1" в пятом разряде числа, записанного во входном устройстве. Программа использует операции маскирования и команды условных переходов.

Программа 3.3

Адрес	Код	Метка	Мнемокод	Комментарий
0800	DB05	WAIT:	IN 05h	Записать число из входного устройства с адресом 05h в аккумулятор
0802	E6 20		ANI 20h	Проверить состояние пятого разряда числа
0804	CA 00 08		JZ WAIT	Переход в точку программы с меткой WAIT, если в пятом разряде был "0", (Z = 0)
0807	CF		JMP CNT	Переход в точку программы с меткой CNT

В представленных ранее программах имел место лишь один цикл, в котором работала микроЭВМ. Программа ожидания появления "1" во втором и пятом разрядах числа, записанного во входном устройстве (программа 3.4), содержит два цикла.

Программа 3.4

Адрес	Код	Метка	Мnemonic	Комментарий
0800	DB05	WAIT:	IN 05h	Записать число из входного устройства с адресом 05h в аккумулятор
0802	E604		ANI 00000100b	Проверить состояние второго разряда числа
0804	CA 00 08		JZ WAIT 1	Переход в точку программы с меткой WAIT1, если в пятом разряде был "0", (Z = 0), если во втором разряде был «0», (Z = 0)
0807	3E FF		MVI A, FFh	Если «1», (Z = 1), то зажечь
0809	D3 05		OUT 05h	светодиоды выходного регистра
080B	DB 05	WAIT:2	IN 05h	Записать число из входного устройства с адресом 05h в аккумулятор
080D	E6 20		ANI 00100000b	Выключен ли пятый переключатель?
080F	CA 0B 08		JZ WAIT2	Если нет, переход в точку программы с меткой WAIT2
0812	3E 00		MVI A, 00h	Если да, погасить светодиоды выходного регистра
0814	D3 05		OUT 05h	
0816	C3 00 08		JMP WAIT1	Повторить программу

Задания для домашней подготовки

1. Изучите способы организации обмена информацией между микроЭВМ и внешними устройствами.
2. Рассмотрите схемы подключения устройств ввода–вывода данных при различных способах обмена.
3. Ознакомьтесь с командами ввода–вывода МП БИС КР580ВМ80, а также временными диаграммами их выполнения.
4. Изучите группу логических команд и команд условной передачи управления.
5. Ознакомьтесь с разрядами регистра признаков МП БИС и правилами записи в них "1".
6. Ознакомьтесь с программами 3.1; 3.2; 3.3; 3.4.
7. Самостоятельно разработайте программы:

а) включения светодиодов выходного устройства, если число, записанное во входном устройстве, больше 3;

б) включения светодиодов выходного устройства, если число, записанное во входном устройстве, больше 3, но меньше 8.

8. Видоизмените программу 3.3 так, чтобы микроЭВМ реагировала на "0" в пятом разряде при записанных "1" во всех остальных разрядах.

Задания к лабораторной работе

Задание 1. Исследовать программу 3.1.

1. Ввести в микроЭВМ программу 3.1. Осуществить пуск программы.
2. Убедиться, что при выполнении программы микроЭВМ постоянно переписывает данные с входного устройства в выходное. Для этого с помощью переключателей входного устройства изменить числа, записанные в нем. Информация о числах в устройствах ввода-вывода может отображаться светодиодами.

Задание 2. Исследовать программу 3.2.

1. Ввести в микроЭВМ программу 3.2. Осуществить пуск программы и исследовать результат ее выполнения по числу, записанному в выходное устройство.
2. Заменяя в программе двухбайтную команду ANI(D) на однобайтные ANA A, XRA A, ORA A, исследовать результат их выполнения по числу, записанному в выходном устройстве.

Задание 3. Исследовать программу 3.3.

1. Ввести в микроЭВМ программу 3.3. Осуществить пуск программы и убедиться, что при ее выполнении микроЭВМ реагирует лишь на те числа во входном устройстве, которые содержат "1" в пятом разряде.
2. Исследовать видоизмененную программу 3.3, позволяющую микроЭВМ реагировать на "0" в пятом разряде при записанных "1" во всех остальных разрядах.

Задание 4. Исследовать программу 3.4.

1. Ввести в микроЭВМ программу 3.4. Осуществить пуск программы и убедиться, что при наличии "1" лишь во втором разряде числа входного регистра светодиоды выходного регистра включены и микроЭВМ работает в цикле WAIT2 ожидания появления единицы в пятом разряде числа.
2. Записать "0" во второй разряд входного устройства при включенных светодиодах входного устройства. Записать "1" в пятый разряд входного устройства и убедиться, что светодиоды выходного устройства выключаются и микроЭВМ находится при выполнении цикла WAIT1 программы.

3. Установить "1" одновременно во втором и пятом разрядах числа во входном устройстве и проверить, что микроЭВМ последовательно выполняет оба цикла (WAIT1, WAIT2) программы.

Задание 5. Исследовать самостоятельно программы, разработанные при выполнении п. 7 задания для домашней подготовки.

Отчет должен содержать:

1. Самостоятельно разработанные и исследованные в процессе выполнения лабораторной работы программы, указанные в п. 7 задания для домашней подготовки.

2. Полный перечень команд передачи управления по условию для МП БИС КР580ВМ80.

3. Полный перечень команд логических операций для МП БИС КР580ВМ80.

Задания для самопроверки

1. С помощью каких команд микроЭВМ может осуществлять ввод–вывод информации?

2. За сколько машинных тактов осуществляется ввод–вывод данных по командам IN(A1), OUT(A1)?

3. Укажите достоинства и недостатки различных методов адресации к внешним устройствам.

4. Перечислите виды логических операций, выполняемые МП БИС.

Литература

1. Микропроцессоры. Основы построения микроЭВМ. / В. Л. Горбунов, Д. И. Панфилов, Д. Л. Преснухин. – Москва: Высш. шк., 1984.

2. Микропроцессоры и микроЭВМ. / И. В. Прангишвили – Москва: Энергия, 1979.

3. Микропроцессоры и микроЭВМ: Пер. с англ. / Соучек Б. – Москва: Сов. радио, 1979.

6 Лабораторная работа № 4

ОРГАНИЗАЦИЯ СТЕКА И ВЫЗОВ ПОДПРОГРАММ В МИКРОПРОЦЕССОРНОЙ СИСТЕМЕ

Цель работы: исследовать особенности записи и обращения к подпрограммам, изучить методы использования стека при создании программ.

Память микроЭВМ, построенной на основе МПК серии К580, а также память большинства из используемых микроконтроллеров имеет ограниченный объем, как правило, не более 65536 однобайтных ячеек. Учитывая ограниченные возможности памяти, при разработке программ стремятся максимально уменьшить их объем. С этой целью часть программы, которая неоднократно повторяется, или программа, которая часто используется, оформляются в виде подпрограмм: последовательностей команд, выполнение которых может быть вызвано из любого места программы любое количество раз. Процесс передачи управления к подпрограмме называется ее вызовом. Данные и адреса, требуемые для работы подпрограммы, называются входными параметрами. Результаты работы подпрограммы, передаваемые по окончании ее работы в основную программу, называются выходными параметрами.

Для вызова подпрограмм и возврата из них используются команды CALL (A2) (A1) и RET.

Команда CALL (A2) (A1) загружает в программный счетчик МП БИС содержимое байтов (A2) (A1), записанных в последующих двух ячейках памяти после адреса, где записан код команды CALL (CDh). Содержимое байта (A2) записывается в младший байт PCL программного счетчика, а третий байт (A1) команды в старший байт PCN программного счетчика, при этом МП БИС автоматически сохраняет в стеке адрес следующей после команды CALL (A2) (A1) команды основной программы, к которому перейдет МП БИС после выполнения подпрограммы.

Стеком называется специально организованная область ОЗУ, задействованная в микроЭВМ для временного хранения данных или адресов. Число, записанное в стек последним, извлекается из него первым.

Команда RET (C9h) помещает в программный счетчик последнее, записанное на данный момент в стеке число. После чего выполнение основной программы будет осуществляться с этого адреса. Любая подпрограмма должна оканчиваться командой RET.

Автоматическое сохранение и восстановление адреса основной программы при выполнении подпрограмм позволяет сделать подпрограммы вложенными, то есть осуществить вызов одной подпрограммы из другой. Уровень вложенности для данной микроЭВМ определяется лишь размером стека.

Существуют также команды условного вызова подпрограмм и возврата из них. Они позволяют вызвать подпрограмму и возвратиться из нее по определенному состоянию заданных разрядов регистра признаков (аналогично командам условных переходов). Все команды условного вызова подпрограммы являются трехбайтными, во втором и третьем байтах, которых сообщается начальный адрес подпрограммы. Команды вызова подпрограмм и возврата из них используют стек и внутренний регистр МП БИС SP (STACKPOINTER) для адресации к стеку.

Помимо команд вызова подпрограмм и возврата из них, со стеком можно обмениваться информацией с помощью команд PUSH (R) (записать в стек содержимое обозначенного регистра МП БИС) и POP (R) (записать данные из стека в обозначенный регистр МП БИС).

При записи в стек содержимого пары регистров или программного счетчика по адресу SP-1 записывается содержимое старшего регистра из указанной пары или старший байт РСН программного счетчика, а по адресу SP-2 в стек записывается содержимое младшего регистра из указанной пары или младший байт РСЛ программного счетчика.

При записи из стека данных в пару регистров или программный счетчик в младший регистр пары или РСЛ записывается число из адреса, указанного в указателе стека SP, а в старший регистр пары или РСН – число, записанное по адресу SP+1. В результате выполнения команды содержимое указателя стека SP увеличивается на 2. Данные в памяти не изменяются, а лишь происходит их чтение и увеличение содержимого SP.

При разработке программ необходимо первоначально назначать область стека, записывая в SP адрес с помощью команды LXI SP (A2) (A1) или команды SPHL.

Все операции со стеком должны быть сбалансированы, то есть каждая подпрограмма должна содержать равное количество команд PUSH (R) и POP (R) и оканчиваться командой RET. В противном случае выполнение команды RET в конце подпрограммы приведет к записи в программный счетчик случайного числа из стека. Адрес возврата в основную программу будет потерян и нарушится последовательность ее выполнения.

Как правило, в начале каждой программы сохраняют в стеке содержимое всех задействованных при ее выполнении регистров с помощью команд PUSH (R).

В конце подпрограммы восстановление содержимого регистров осуществляется с помощью команд POP (R) и в обратной последовательности по отношению к их записи в стек.

Обычно в виде подпрограмм записываются многократно используемые фрагменты программ, например, подпрограмма выдачи звукового сигнала, подпрограмма обслуживания клавиатуры и дисплея и т. д.

Алгоритм работы простой подпрограммы временной задержки приведен на рисунке 4.1.



Рисунок 4.1 – Алгоритм работы простой подпрограммы временной задержки

Общее время задержки вычисляется по формуле $Td = t1 + (t2 + t3 + t4) N + t5$, где N – число, первоначально записанное в счетчике. В качестве счетчика выбран регистр B , в который записывается число N из регистра C .

Команда NOP нужна для увеличения времени выполнения цикла, а, следовательно, и общей задержки. Вместо команды NOP может быть записана любая последовательность команд, выполнение которых не изменяет содержимого регистров микропроцессора. Время записи числа N в регистр B и возврата из подпрограммы $t1 + t5$ фиксировано и в цикл не входит. Минимальная задержка для приведенной подпрограммы определяется при $N=1$ и равна $Tdmin = t1 + t2 + t3 + t4 + t5$. Максимальная задержка имеет место при $N=256$ и, соответственно, равна $Tdmax = t1 + (t2 + t3 + t4) 256 + t5$.

Время t для каждой команды определяется числом машинных тактов МП БИС, за которые она выполняется. Один такт соответствует периоду синхронизации МП БИС.

Подпрограмма DLY (подпрограмма 4.1) представляет подпрограмму временной задержки, записанную в соответствии с алгоритмом, представленным на рисунке 4.1. Рассмотрим пример использования подпрограммы временной задержки при организации звуковых сигналов в микроЭВМ.

В микроЭВМ звуковые сигналы могут формироваться простейшей схемой, на вход которой со звуковой частотой записываются по очереди 0 и 1. Устройство формирования звуковых сигналов имеет адрес 04h (бит DO). Схема алгоритма работы подпрограммы генерации звуковых колебаний (подпрограмма 4.2) приведена на рисунке 4.2.

Подпрограмма 4.1

Адрес	Код	Метка	Мнемокод	Комментарий
0900	41		MOV B,C	Записать число из регистра C в B
0901	00	DLY	NOP	Нет операции
0902	05		DCR B	Уменьшить число в регистре B на 1
0903	C2 0109		JNZ DLY	Если B не равно 0, то идти переход на метку DLY
0906	C9		RET	Возврат из подпрограммы

Подпрограмма 4.2

Адрес	Код	Метка	Мнемокод	Комментарий
0A00	AF	BPP	XRA A	Очистить аккумулятор
0A01	D3 04		OUT 04h	Записать 0 в выходной регистр
0A03	CD 0009		CALL DLY	Вызвать подпрограмму задержки
0A06	2F		CMA	Записать FF в аккумулятор
0A07	D3 04		OUT 04h	Записать FF в выходной регистр
0A09	CD 0009		CALL DLY	Вызвать подпрограмму задержки
0A0C	C9		RET	Возврат из подпрограммы

Программа 4.3

Адрес	Код	Метка	Мнемокод	Комментарий
0800	DB05	MAIN	IN 05	Записать число из входного регистра в аккумулятор
0802	4F		MOVC, A	Записать число в регистр C
0803	CDOOOA		CALL BPP	Вызвать подпрограмму BPP
0806	C3 0008		JMP MAIN	Продолжать



Рисунок 4.2 – Схема алгоритма работы подпрограммы генерации звуковых колебаний

При использовании подпрограмм за счет изменения входных параметров можно влиять на конечный результат выполнения подпрограмм. Рассмотрим это на примере регулируемой временной задержки, подпрограмма которой находится в ПЗУ микроЭВМ. Программа состоит из двух подпрограмм: DELB (адрес 0430) и DELA (адрес 0429).

Подпрограмма DELB осуществляет регулируемую временную задержку, и входным параметром ее является двухбайтное число, записанное в паре регистров (B, C). Это число и определяет длительность задержки в миллисекундах. Частным случаем подпрограммы DELB является подпрограмма DELA, осуществляющая фиксированную задержку в 1 мс и не имеющая входных параметров. Таким образом, если необходимо иметь в программе фиксированную задержку в 1 мс, то можно обращаться с помощью команды CALL к подпрограмме DELA. При необходимости получения определенной заданной временной задержки в программе необходимо записать соответствующее число в регистры B, C, а затем вызвать подпрограмму DELB.

Задания для домашней подготовки

1. Ознакомьтесь с командами вызова и возврата из подпрограммы CALL (A2) (A1) и RET, а также аналогичными командами условного вызова и возврата.

2. Определите, при каких числах, записанных в регистре B, подпрограмма 4.1 будет осуществлять минимальное и максимальное время задержки. Определите эти времена, если машинный такт $T = 0,5$ мкс.

3. Разработайте подпрограмму 5– и 10–секундной временной задержки.

4. Составьте программу, последовательно включающую светодиоды выходного устройства на время, соответственно, 10 и 5 сек.

5. Разберите программу 4.4, определяющую, какой из восьми переключателей входного устройства (адрес 05h) микроЭВМ установлен в положение 0:

а) составьте алгоритм работы программы 4.4;

б) запишите подпрограмму, используемую в программе 4.4;

в) установите, в каком регистре МП БИС содержится информация о номере переключателя входного устройства, установленном в 0;

г) определите, как будет работать программа 4.4, если на входном устройстве будут установлены в положение 0 не один, а несколько переключателей.

Программа 4.4

Адрес	Код	Метка	Мнемокод	Комментарий
0800	31 B00B		LXI SP, 0BB0	Записать в указатель стека SP адрес 0BB0

Адрес	Код	Метка	Мнемокод	Комментарий
0803	DB 05	WAITC	IN	Получить число из входного устройства
0805	FE FF		CPI FF	Содержит ли какой-либо разряд 0?
0807	CA 0308		JZW AITC	Если нет, то ждать
080D	CF		RST1	Окончить
0810	06 FF	IDSW	MVI B, FF	Записать в FF
0812	04	SRCH	INR B	Увеличить содержимое регистра B на 1
0813	0F		RRC	Циклически сдвинуть число в аккумуляторе вправо
0814	DA 1208		JC SRCH	Если C = 1, то продолжать
0817	C9		RET	Возврат из подпрограммы

Задания к лабораторной работе

Задание 1. Исследовать процесс выполнения команд вызова из подпрограммы, а также команд работы со стеком.

1. Ввести в микроЭВМ подпрограмму 4.5.
2. Выполнить программу 4.5 по командам. После каждой команды проверить содержимое всех регистров МП БИС.
3. Выполнить команды CALL STDY, PUSH PSW, POP, RET по машинным циклам.
4. Заменить в подпрограмме 4.5 команду POPPSW на команду NOP и проследить, как будет выполняться подпрограмма 4.5. Объяснить изменения.

Подпрограмма 4.5

Адрес	Машинный код	Метка	Мнемокод	Комментарий
0800	31B00B		LXI SP, 0B00h	Записать в указатель стека SP адрес 0B00h
0803	CD 0708		CALL STDY	Вызвать подпрограмму STDY
0806	CF		RST1	
0807	F5	STDY	PUSH PSW	Записать слово состояния МП БИС в стек
0808	C5		PUSH B	Сохранить содержимое регистров BC в стеке
0809	D5		PUSH D	Сохранить содержимое регистров DE в стеке

Адрес	Машинный код	Метка	Мнемокод	Комментарий
080A	E5		PUSH H	Сохранить содержимое регистров HL в стеке
080B	3E 05		MVI A, 05	
080D	47		MOV B, A	
080E	87		ADD A	
080F	5F		MOV E, A	
0810	67		MOV H,A	
0811	E1		POP H	Восстановить содержимое регистров HL
0812	D1		POP D	Восстановить содержимое регистров DE
0813	C1		POP B	Восстановить содержимое регистров BC
0814	F1		POP PSW	Восстановить содержимое слова состояния
0815	C9		RET	

Задание 2. Исследовать программу временной задержки на примере работы программы генерации звуковых колебаний (программа 4.3).

1. Ввести в микроЭВМ полный текст программы (4.3) для генерации звуковых сигналов с учетом подпрограмм 4.1 и 4.2.
2. Установить на входном устройстве число 0.
3. Осуществить пуск программы.
4. Проследить за изменением тона звука, формируемого микроЭВМ в процессе выполнения программы, увеличивая число, записанное во входном устройстве.

Задание 3. Исследовать программу регулируемой временной задержки на примере программы, последовательно включающей и выключающей светодиоды выходного устройства на время, соответственно, 10 и 5 сек.

1. Ввести в микроЭВМ программу, разработанную в п. 5 домашнего задания.
2. Осуществить пуск программы и проверить ее выполнение.
3. Изменить числа, записанные в регистрах BC, в подпрограмме регулируемой временной задержки (4. 4). Проверить возможность изменения длительности задержки.

Задание 4. Исследовать программу 4.5.

1. Ввести программу 4.5 в микроЭВМ.
2. Установить на входном устройстве переключатели в положение, соответствующее числу FFh. Осуществить пуск программы и убедиться, что

микроЭВМ будет находиться в режиме ожидания появления 0 в любом разряде входного устройства

3. Установить 0 с помощью переключателей в любом из разрядов входного устройства. Проверить содержимое регистров МП БИС после окончания выполнения программы.

4. Осуществить повторный запуск программы при наличии нулей в двух разрядах входного устройства. Какое число будет записано в регистре В МП БИС после окончания выполнения программы?

Отчет должен содержать:

1. Ответы на вопросы п. 1 и п. 2 домашнего задания.
2. Полный перечень команд вызова и возврата из подпрограмм для МП БИС КР580ВМ80.
3. Разработанные в процессе домашней подготовки программы 5– и 10–секундной задержки.
4. Перечень команд работы со стеком для МП БИС КР580ВМ80.

Задания для самопроверки

1. Укажите количество машинных тактов выполнения команды CALL (A2) (A1).
2. В какой последовательности записывается и считывается из стека содержимое аккумулятора и регистра признаков МП БИС при выполнении команды PUSH PSW и POP PSW?
3. С помощью каких команд можно задать или переобозначить область памяти, отведенную под стек?
4. Укажите порядок выполнения микроЭВМ команды RET.
5. В какой последовательности сохраняется и извлекается содержимое регистров МП БИС в подпрограммах?
6. Какой минимальный адрес будет записан в указателе стека SP МП БИС при выполнении подпрограммы 4.5?
7. Как будет выполняться подпрограмма 4.5, если вместо команды POP В в ней будет записана команда NOP?

Литература

1. Микропроцессоры. Основы построения микроЭВМ. / В. Л. Горбунов, Д. И. Панфилов, Д. Л. Преснухин. – Москва: Высш.шк., 1984.
2. Микропроцессоры и микроЭВМ. / И. В. Прангишвили. – Москва: Энергия, 1979.
3. Микропроцессоры и микроЭВМ: Пер. с англ. / Б. Соучек – Москва: Сов. радио, 1979.

7 Лабораторная работа №5

ОРГАНИЗАЦИЯ ПРОГРАММ ДВОИЧНОЙ АРИФМЕТИКИ

Цель работы: изучить способы организации и исследовать программы выполнения арифметических операций.

Из двух способов определения чисел с фиксированной и плавающей точкой первый получил наибольшее распространение при программировании микроЭВМ на МП БИС КР580ВМ80. Это связано с отсутствием специальных команд, позволяющих работать МП БИС с числами с плавающей точкой.

Проведение арифметических операций сложения, вычитания, умножения, деления вычисления специальных функций рассмотрим на примерах соответствующих подпрограмм.

Программа 5. 1 (MAIN) позволяет осуществлять сложение массива однобайтных чисел с получением двухбайтного результата. Слагаемые должны быть расположены в последовательных адресах памяти. Входными параметрами для подпрограммы ADDB являются адрес первого слагаемого, записанный в регистровой паре HL, и число слагаемых, записанное в регистре В. Выходным параметром программы MAIN является сумма, старший байт которой записан в регистре С, а младший – в аккумуляторе А. Алгоритм программы сложения состоит в том, что после прибавления каждого элемента массива определяется переполнение аккумулятора (разряд С = 1), и если это имеет место, то содержимое регистра С увеличивается на 1. Таким образом, за счет суммирования единиц переноса получается старший байт суммы.

Подпрограмма 5.1

Адрес	Машинный код	Метка	Мнемокод	Комментарий
0800	21 000B	MAIN	LXI H, 000h	Записать в HL адрес первого слагаемого
0803	06 05		MVI B, 05h	Загрузить в В количество слагаемых
0805	CD 0908		CALL ADDB	Вызвать подпрограмму сложения
0808	CF		RST 1	Прервать выполнение программы
0809	AF	ADDB	XRA A	Очистить аккумулятор
080A	4F		MOV C, A	Очистить счетчик переносов
080B	86	CNT	ADD M	Прибавить к аккумулятору число из массива слагаемых

Адрес	Машинный код	Метка	Мнемокод	Комментарий
080C	D2 1008		JNC TRM	Если нет переноса, то переход на метку программы TRM
080F	0C		INR C	Увеличить регистр C на 1
0810	23	TRM	INX H	Указать на следующий адрес слагаемого
0811	05		DCR B	Уменьшить счетчик слагаемых
0812	C2 0B08		JNC CNT	Если слагаемые не все, то переход на метку программы CNT
0815	C9		RET	

МикроЭВМ может проводить арифметические операции с числами с двойной или большей длиной машинного слова. Так как МП БИС имеет 8-разрядное АЛУ, то операции с такими числами должны проводиться по байтам, начиная с младших байтов. При суммировании (вычитании) младших байтов чисел необходимо применять команду ADD (SUB), а для сложения (вычитания) остальных – команду ADC (SBB), которая будет учитывать состояние разряда CY регистра признаков МП БИС.

Программа 5.2 находит разность чисел, имеющих одинаковую длину. Входные параметры: в регистре C длина чисел в байтах, в регистровой паре HL – адрес младшего байта вычитаемого, в регистровой паре DE – адрес младшего байта уменьшаемого. Каждое из чисел записывается в последовательных адресах памяти, начиная с младших байтов. В том случае, если уменьшаемое меньше вычитаемого, будет подаваться звуковой сигнал.

Выполнение команд MOV M, A; INX H; INX D не воздействует на разряд CY. В программе SBN подпрограмма звукового сигнала (BEEP) начинается с адреса 0012h. При исследовании программы SBN необходимо перед началом выполнения ее осуществить начальную установку всех входных параметров.

Программа 5.2

Адрес	Машинный код	Метка	Мнемокод	Комментарий
0900	AF	SBN	XRA A	Очистить A и флаг CY
0901	1A	CNT	LDA XD	Записать в аккумулятор уменьшаемое
0902	9E		SBB M	Вычесть из содержимого аккумулятора вычитаемое
0903	77		MOV M, A	Записать разность на место вычитаемого

Адрес	Машинный код	Метка	Мнемокод	Комментарий
0904	23		INX H	Указать на следующий байт вычитаемого
0905	13		INX D	Указать на следующий байт уменьшаемого
0906	0D		DCR C	Уменьшить содержимое счетчика длины числа
0907	C2 0109		JNZ CNT	Если не последний байт, то переход на метку программы CNT
090A	D0		RNC	Если байт старший и результат без заема ($C = 0$), то возврат
090B	CD 1200		CALL BEEP	Если был заем ($C = 1$), подать сигнал
090E	CF		RST 1	Прервать выполнение программы

Умножение чисел

Существует несколько алгоритмов умножения чисел. В одних случаях умножение можно заменить многократным сложением, например, $14 * 3 = 14 + 14 + 14$. Существенным недостатком этого способа является значительная длительность процесса вычисления. Более оптимальным является алгоритм, при котором умножение осуществляется в столбец. Этот алгоритм применим и для умножения двоичных чисел, например,

$$\begin{array}{r}
 0110 = 6_{10} \\
 \underline{0011 = 3_{10}} \\
 0110 \\
 0110 \\
 0000 \\
 \underline{0000} \\
 00010010 = 18_{10}
 \end{array}$$

При вычислении результата необходимо осуществить многократное суммирование со сдвигом влево умножаемого при одновременной проверке содержимого разрядов множителя, начиная со стороны его младшего разряда.

При этом если в очередном разряде множителя записана 1, то умножаемое прибавляется к сумме и сдвигается влево на один разряд, а если в разряде записан 0, то произойдет только сдвиг умножаемого. Сдвиг умножаемого влево можно заменить сдвигом суммы вправо. По этому алгоритму работает подпрограмма MUL умножения двух однобайтных чисел с получением двухбайтного результата, расположенная в ПЗУ по адресу 04E1h.

Входные параметры: регистр D – умножаемое, регистр E – множитель. Результат перемножения записывается в регистры B и C.

Деление чисел

Деление двоичных чисел, как и чисел, представленных в любой другой системе счисления, основывается на последовательном вычитании делителя из делимого и остатков от деления. Однако двоичное деление реализуется проще, так как использование только двух цифр (0 и 1) исключает в каждом цикле деления необходимость определения числа делителей, содержащихся в текущем значении делимого или остатка (достаточно только сравнить их). Программа 5.3 (DIVB) построена по этому алгоритму. Входными параметрами этой программы являются делимое (в регистре E) и делитель (в регистре D); выходными параметрами – частное (в регистре H) и остаток (в регистре C).

<u>Программа 5.3</u>				
Адрес	Машинный код	Метка	Мнемокод	Комментарий
0900	21 0800	DIVB	LXI H, 0008h	Загрузить счетчик битов (регистр L) и очистить регистр частного H
0903	0E 00		MVI C, 00h	Очистить регистр промежуточного делимого
0905	7B	MXTB	MOV A, E	Загрузить делимое в аккумулятор
0906	17		RAL	Сдвинуть старший бит в разряд CY
0907	5F		MOV E, A	Возвратить делимое в регистр E
0908	79		MOV A, C	Загрузить в аккумулятор промежуточное делимое из регистра C
0909	17		RAL	Сдвинуть разряд CY в младший бит
090A	92		SUB D	Вычесть из содержимого аккумулятора делитель
090B	D2 0F09		JNC NOADD	Если C = 1, восстановить содержимое аккумулятора
090E	82		MOV A, C	
090F	4F	NOADD	MOV C, A	Возвратить промежуточное данное в регистр C
0910	3F		CMC	Инвертировать CY

Адрес	Машинный код	Метка	Мнемокод	Комментарий
0911	7C		MOV A,H	Сдвинуть разряд CY в младший
0912	17		RAL	бит регистра частного H
0913	67		MOV H, A	
0914	2D		DCR L	Проверены ли все 8 разрядов?
0915	C2 0509		JNZ MXTB	Если нет, продолжать
0918	CF		RST 1	Если да, прервать выполнение программы

Вычисление специальных функций

Для вычисления специальных функций ($\sin x$, $\cos x$, $\operatorname{tg} x$, $\ln x$, x^2) применяются специальные алгоритмы. Функции $\sin x$, $\cos x$, $\ln x$ можно вычислить, воспользовавшись их разложением в ряд:

$$\begin{aligned} \sin x &= x - (x^3)/3! + (x^5)/5! - (x^7)/7! + \dots \text{ для любого } x \text{ (рад)} \\ \cos x &= x - (x^2)/2! + (x^4)/4! - (x^6)/6! + \dots \text{ для любого } x \text{ (рад)} \\ \ln(1+x) &= x + (x^2)/2! + (x^3)/3! - (x^4)/4! + \dots \text{ для } 0 < x < 1 \end{aligned}$$

Число членов ряда определяется из условия получения требуемой точности. Для вычисления функции x^2 с точностью до целых чисел можно применить алгоритм, основанный на том, что квадрат числа можно определить сложением последовательности нечетных чисел:

Число	Сумма нечетных чисел		Результат
1	1	=	1
2	1 + 3	=	4
3	1 + 3 + 5	=	9
4	1 + 3 + 5 + 7	=	16
5	1 + 3 + 5 + 7 + 9	=	25

Исходя из приведенного примера видно, что какое число необходимо возвести в квадрат, такое же количество последовательных нечетных чисел, начиная с 1, необходимо сложить.

Вычисление специальных функций по приведенным выражениям занимает длительное время и обеспечивает низкую точность. Это обусловлено сравнительно небольшой длиной машинного слова и ограниченным

быстродействием МП БИС. Поэтому в тех случаях, когда ставятся жесткие требования по быстродействию и точности, применяется вычисление функций с помощью таблиц. Проиллюстрируем этот метод на примере программы вычисления квадрата числа x (программа 5.4).

Программа SQ определяет квадрат чисел от 0 до 10 включительно. Входной параметр программы – число x . Оно записывается в регистр L . Выходной параметр – значение x^2 в регистре H .

Программа 5.4

Адрес	Машинный код	Метка	Мнемокод	Комментарий
0900	26 00	SQ	MVI H, 00h	Очистить регистр H
0902	11 000A		LXI D, SQTБ	Загрузить начальный адрес таблицы
0905	19		DAD D	Получить адрес элемента
0906	66		MOV H, M	Загрузить его в регистр H
0907	CF		RST 1	Прервать выполнение программы
0A00	00	SQTБ	00h	
0A01	01		01h	
0A02	04		04h	Таблица квадратов чисел
0A03	09		09h	
0A04	10		10h	
0A05	19		19h	
0A06	24		24h	
0A07	31		31h	
0A08	40		40h	
0A09	51		51h	
0A0A	64		64h	

Задание для домашней подготовки

1. Изучите группу арифметических команд МП БИС КР580ВМ80.
2. Ознакомьтесь с правилами выполнения команды DAA десятичной коррекции аккумулятора.
3. Изучите программы, приведенные выше. Рассмотрите результат выполнения каждой программы на конкретных числовых примерах.
4. Составьте программу для исследования результата перемножения двух чисел на основе подпрограммы MUL, расположенной в ПЗУ.
5. Разработайте программу сложения двух 8-разрядных чисел с получением результата в двоично-десятичном коде (результат записать в выходной регистр).
6. Оцените время выполнения подпрограммы 5.3.

Задания к лабораторной работе

Задание 1. Исследовать программу сложения однобайтных чисел с получением двухбайтного результата.

1. Ввести в микроЭВМ программу 5.1.
2. Записать в микроЭВМ последовательность из пяти чисел, начиная с адреса ОВ00.
3. Осуществить пуск программы и проверить ее выполнение по данным, записанным в регистре С и аккумуляторе МП БИС.
4. Изменить программу 5.1 так, чтобы результат выполнения записывался по адресу ОВ06h и ОВ07h. Проверить результат ее выполнения.
5. Изменить в программе указатель количества слагаемых в сумме и выполнить программу заново.

Задание 2. Исследовать программу вычитания двух чисел, имеющих одинаковую длину.

1. Ввести в микроЭВМ программу 5.2.
2. Записать в регистры DE и HL, соответственно, начальные адреса младших байтов уменьшаемого и вычитаемого. Вычитаемое должно быть записано в области ОЗУ, где нет защиты от случайной записи во время выполнения программ (адреса ОВ00h – ОВВ0h). Записать в регистр С длину числа в байтах. Для первого случая записать в регистр С число 01h (рассматривается вычитание двух 8–разрядных чисел).
3. Записать по адресам, указанным в регистрах HL и DE, уменьшаемое и вычитаемое. Выбрать при этом уменьшаемое большим вычитаемого.
4. Осуществить пуск программы 5.2 и исследовать результат ее выполнения по числу, записанному по адресу, где находится вычитаемое.
5. Видоизменить программу 5.2 так, чтобы результат вычисления разности двух 8–ми разрядных чисел записывался в выходной регистр.
6. Изменить числа, записанные по адресам, указанным в регистрах HL и DE так, чтобы уменьшаемое было меньше вычитаемого.
7. Осуществить пуск программы 5.2 и проследить, что при этом получается.
8. Исследовать в аналогичной последовательности результат вычитания двухбайтных чисел.

Задание 3. Исследовать программу умножения двух 8–разрядных чисел с получением 16–разрядного результата.

1. Ввести в микроЭВМ программу для исследования результата умножения двух чисел, разработанную в п. 5 домашнего задания.
2. Осуществить пуск программы и проверить результат умножения двух чисел по числу, записанному в регистрах ВС.

Задание 4. Исследовать программы деления двух 8–разрядных чисел.

1. Ввести в микроЭВМ программу 5.3.
2. Записать в регистры E и D , соответственно, делимое и делитель.
3. Осуществить пуск программы и проверить результат деления двух чисел по содержимому регистров H и C .

Задание 5. Исследовать программу вычисления квадрата числа с помощью таблицы.

1. Ввести в микроЭВМ программу 5.4.
2. Записать в регистр L число, квадрат которого необходимо вычислить.
3. Осуществить пуск программы и проверить результат вычисления квадрата числа по содержимому регистра H .
4. Видоизменить программу так, чтобы результат вычисления квадрата числа записывался в выходной регистр.
5. Ввести в программу 5.4 проверку на значение числа, квадрат которого определяется в результате выполнения программы. Если задаваемое число больше 10, то микроЭВМ должна указывать на это, например, подачей звукового сигнала.

Задание 6. Исследовать программу сложения двух 8–разрядных чисел с получением результата в двоично–десятичном коде.

1. Ввести в микроЭВМ программу, разработанную в п. 5 домашнего задания.
2. Осуществить пуск программы и проверить результат сложения следующих чисел: $33 + 25$; $38 + 25$; $98 + 25$.

Отчет должен содержать:

1. Полный перечень арифметических команд МП БИС КР580ВМ80.
2. Программу сложения двух 8–разрядных чисел с получением результата в двоично–десятичном коде, разработанную в п. 5 домашней подготовки.
3. Программу исследования результата умножения двух 8–разрядных чисел, разработанную в п. 4 домашней подготовки.
4. Результаты выполнения арифметических операций по всем заданиям.

Задания для самопроверки

1. Какие команды арифметических операций может выполнять МП БИС КР580ВМ80?
2. Сформулируйте правило выполнения МП БИС команды ДАА.
3. После каких команд можно выполнять десятичную коррекцию числа в аккумуляторе (выполнять команду ДАА)?
4. На чем основаны алгоритмы работы подпрограмм умножения и деления чисел?

5. В чем преимущество табличного способа вычисления функций?
6. Представьте числа от 0 до 20 в двоично–десятичном коде.

Литература

1. Микропроцессоры. Основы построения микроЭВМ. / В. Л. Горбунов, Д. И. Панфилов, Д. Л. Преснухин. – Москва: Высш.шк., 1984.
2. Микропроцессоры и микроЭВМ. / И. В. Прангишвили. – Москва: Энергия, 1979.
3. Микропроцессоры и микроЭВМ: Пер. с англ. / Б. Соучек – Москва: Сов. радио, 1979.

8 Лабораторная работа № 6

ПОДКЛЮЧЕНИЕ ЗНАКОСИНТЕЗИРУЮЩЕГО ДИСПЛЕЯ И КЛАВИАТУРЫ К МИКРОЭВМ

Цель работы: изучить программно–аппаратные методы подключения дисплея и клавиатуры к микроЭВМ.

В качестве устройства вывода информации, удобного для восприятия, часто используется дисплей. Рассмотрим методы подключения дисплея, состоящего из шести ячеек (восьмисегментных индикаторов), представляющих собой восемь светодиодов с общим катодом в одном корпусе. Каждый индикатор (рисунок 6.1) имеет семь светодиодов для отображения сегментов цифр, а восьмой светодиод отображает десятичную точку. Индикатор может отображать цифры от 0 до 9, а также некоторые буквы.

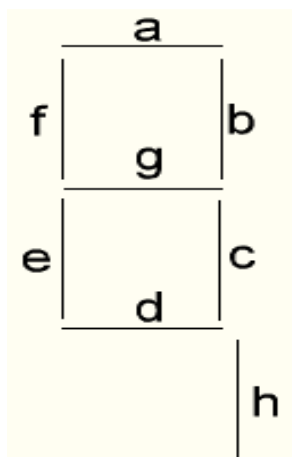


Рисунок 6.1 – Индикатор

Для уменьшения схемотехнического обеспечения, необходимого для подключения дисплея к микроЭВМ, часто применяют мультиплексный режим работы индикаторов. При этом для вывода на дисплей информации используют два выходных регистра: PгСг (адрес 06h) для записи кода выводимого символа и PгСк (адрес 07h) для записи номера индикатора (рисунок 6.2).

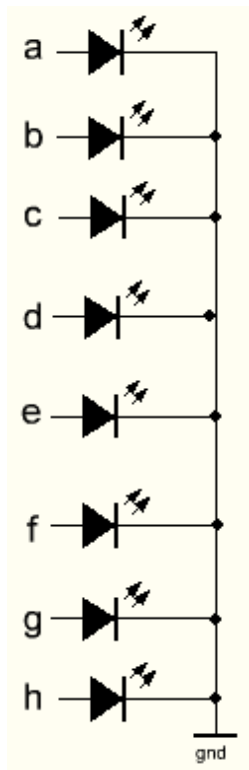


Рисунок 6.2 – Электрическая схема индикатора

Одинаковые сегменты каждой ячейки индикатора связаны общей шиной, которая соединена с выходами регистра PгСг (Seg). Общие катоды индикатора подключены к одному из выходов регистра сканирования PгСк (Scan).

Включение индикатора и его сегментов при записи единицы в соответствующие разряды выглядит так:

для регистра PгСг:

Номер разряда регистра сегментов дисплея	0	1	2	3	4	5	6	7
Включенный сегмент	a	b	c	d	e	f	g	h

для регистра PгСк:

Номер разряда регистра цифр дисплея	0	1	2	3	4	5	6	7
Включаемая цифра дисплея	0	1	2	3	4	5	–	–

Сигналы поданы на регистры Scan и Seg таким образом (рисунок 6.3), что

при поступлении на входы регистров E сигналов выборки от дешифратора адреса DC Seg или DCS can, данные с магистрали MD, подключенной к входам регистров DO – D7, записываются в соответствующий регистр и появляются на его выходах Q0 – Q7.

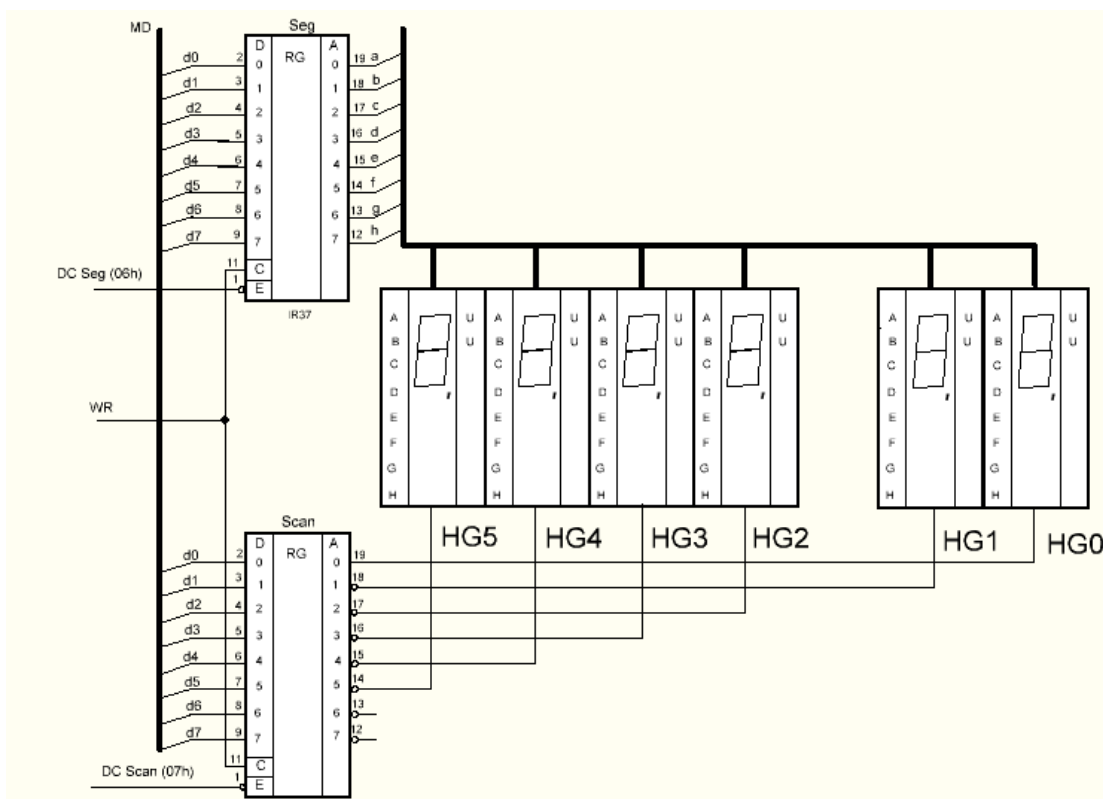


Рисунок 6.3 – Электрическая схема дисплея на сегментных индикаторах

Запись данных осуществляется с помощью команд OUT 06h и OUT 07h для Seg и Scan, соответственно. Например, при записи в регистр сегментов кода 0000 0110, а в регистр сканирования 0010 0000, ток проходит по цепи: лог. 1 выходов Q1 и Q2 регистра Seg → аноды b и c индикатора → общий катод индикатора → лог. 0 выхода Q5 регистра Scan, что приводит к высвечиванию на индикаторе HG5 цифры 1.

Если теперь с помощью программы высвечивать по очереди все ячейки дисплея, записывая их код в регистр Scan, то при достаточной частоте переключений информация на дисплее будет казаться выводимой одновременно.

Программа (6.1) включает сегменты HG5, задаваемые кодом со входного устройства микроЭВМ.

Программа 6.1

Адрес	Машинный код	Метка	Мнемокод	Комментарий
0800	3E 20		MVI A, 20h	Загрузить в аккумулятор число 0010 0000

Адрес	Машинный код	Метка	Мнемокод	Комментарий
0802	D3 07		OUT 07h	Включить цифру 5
0804	DB 05	CNT	IN 05h	Считать данные из входного устройства
0806	D3 06		OUT 06h	Записать их в регистр сегментов дисплея
0808	C3 0408		JMP CNT	Продолжить

Организация мультиплексного режима работы дисплея

При мультиплексном режиме работы вывод информации на каждый индикатор дисплея производится микроЭВМ последовательно. Цифра или символ на индикаторе высвечиваются, некоторый промежуток времени, задаваемый подпрограммой задержки. При большой частоте сканирования индикаторов на цифровом дисплее получается устойчивое изображение.

Схема алгоритма программы, обеспечивающей мультиплексный режим работы дисплея, приведена на рисунке 6.4 (программа 6.2). Код цифр для вывода в каждый разряд дисплея хранится в последовательных ячейках памяти с адресами 0900h– 0905h. При этом полагается, что код цифры для вывода на нулевой индикатор дисплея записан по адресу 0900. Начальный адрес подпрограммы временной задержки 0430h.

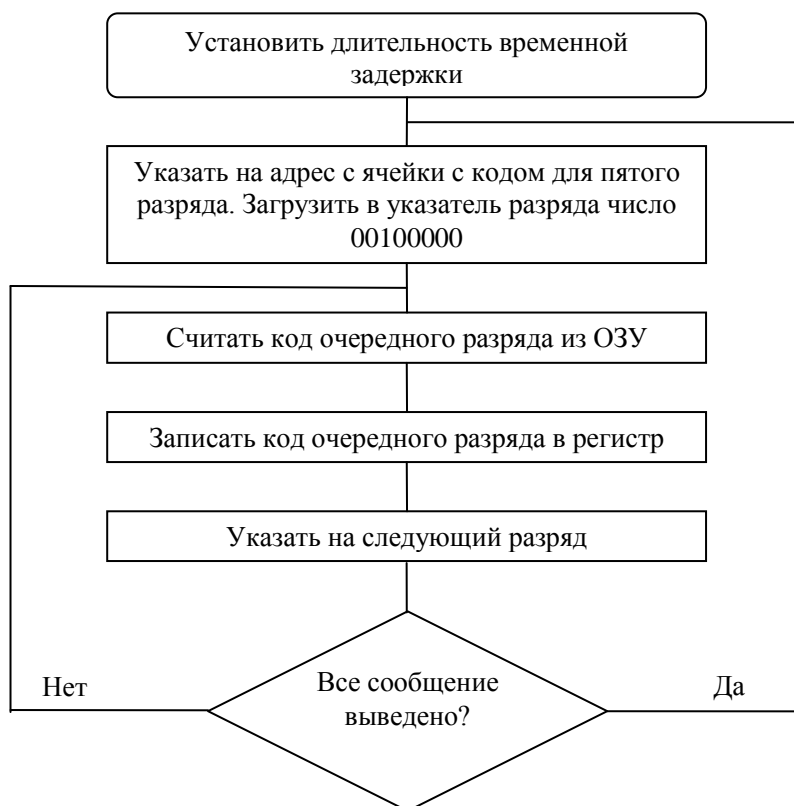


Рисунок 6.4 – Схема алгоритма программы, обеспечивающей мультиплексный режим работы дисплея

Программа 6.2

Адрес	Машинный код	Метка	Мнемокод	Комментарий
0800	01 0004	DSO	LXI B, 0400h	Загрузить в регистры В, С длительность задержки
0803	AF		XRA A	Очистить аккумулятор
0804	21 0509	CNT1	LXI H, 0905h	Указать на адрес кода индикатора HG5
0807	16 20		MVI D, 20h	Загрузить указатель разряда в регистр D
0809	7E	CNT2	MOV A, M	Получить из ОЗУ код очередного разряда
080A	D3 06		OUT 06h	Записать его в регистр сегментов дисплея
080C	7A		MOV A, D	Загрузить в аккумулятор указатель разряда
080D	D3 07		OUT 07h	Включить нужную цифру
080F	1F		RAR	Указать на следующий разряд индикатора
0810	67		MOV D, A	Сохранить указатель разряда в регистре D
0811	CD 3004		CALL DELB	Вызвать подпрограмму временной задержки
0814	AF		XRA A	Очистить аккумулятор
0815	D3 07		OUT 07h	Выключить разряд
0817	2D		DCR L	Уменьшить на 1 содержимое регистра L
0818	B2		ORA D	Все ли сообщение выведено?
0819	C2 0908		JNZ CNT2	Если нет, продолжить
081C	C3 0408		JMP CNT1	Если да, то начать сначала

Подключение клавиатуры к микроЭВМ

Клавиатура является одним из широко распространенных устройств ввода данных и управляющих воздействий в микроЭВМ. С помощью клавиатуры в учебной микроЭВМ можно вводить программу в ОЗУ, инициировать различные режимы ее работы (пуск программы с заданного адреса, останов программы, выполнение программы по машинным циклам, вывод содержания регистров МП БИС на дисплей и т. д.).

Во всех случаях при организации ввода информации с клавиатуры в микроЭВМ перед разработчиком ставится ряд задач, к основным из которых можно отнести:

- 1) определение факта нажатия клавиши на клавиатуре;
- 2) нахождение номера нажатой клавиши;
- 3) осуществление передачи управления на соответствующую подпрограмму.

Наиболее удобно организовывать клавиатуру в виде матрицы размером n и m , где n и m – соответственно, число строк и столбцов. При таком способе организации к микроЭВМ можно подключать $n \times m$ клавиш. Сопряжение клавиатуры с микроЭВМ производят с помощью устройств ввода-вывода данных. Для иллюстрации метода сопряжения клавиатуры с микроЭВМ рассмотрим клавиатуру 3x3, представленную на рисунке 6.4 (где СхДША – схема дешифрации адреса, УВВ – устройство ввода, УВыв – устройство вывода), строки которой подключены к трем младшим разрядам устройства вывода УВыв, а столбцы подключены к трем младшим разрядам устройства ввода УВВ.

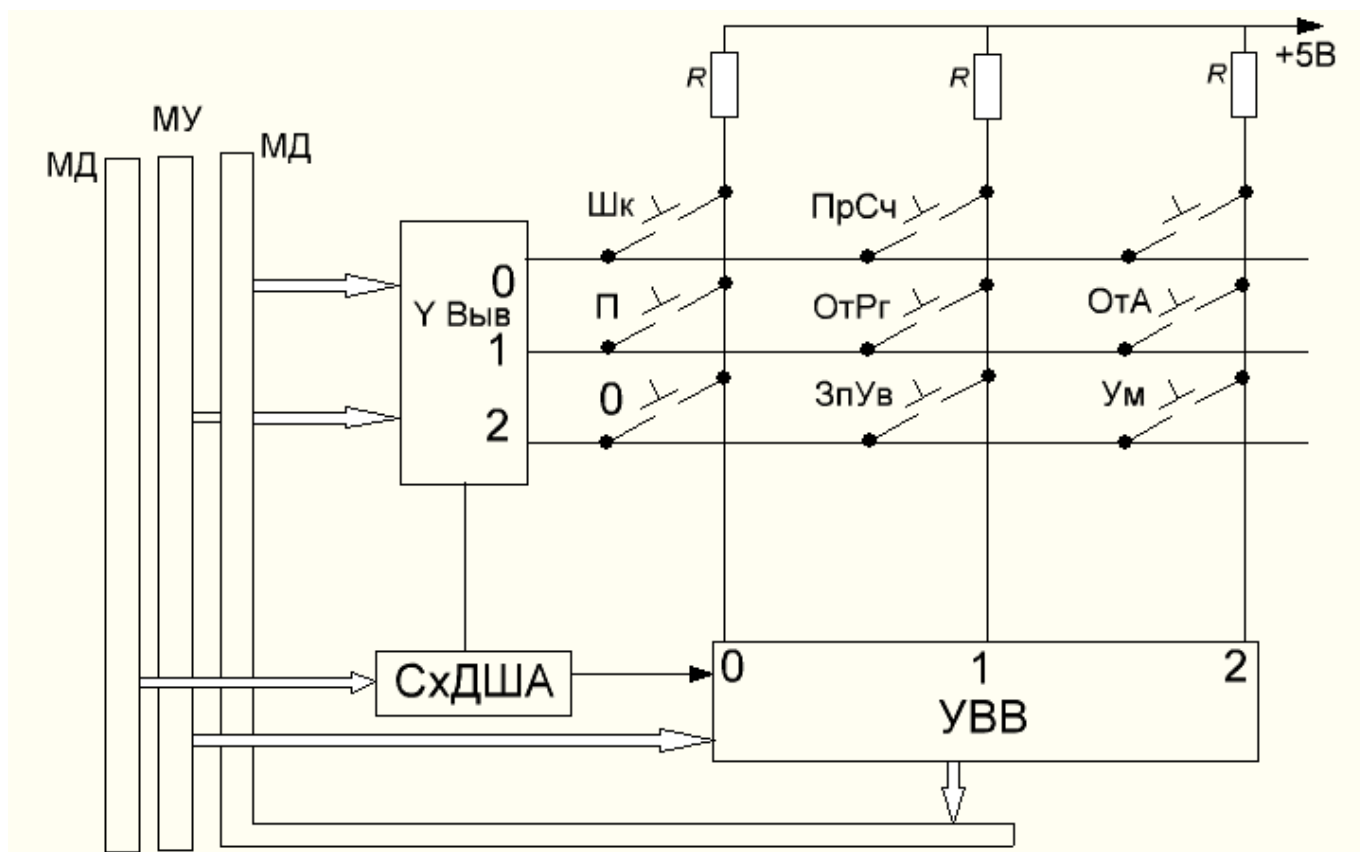


Рисунок 6.4 – Клавиатура 3x3

При программном способе дешифрации нажатой клавиши определение факта нажатия на клавишу может быть осуществлено с помощью такой последовательности операций:

1. Записать логические нули в разряды выходного устройства.
2. Считать содержимое разрядов входного устройства.
3. Повторить снова, если во всех разрядах входного устройства записаны единицы.

Программа 6.3 определяет факт нажатия на одну из клавиш без указания ее номера.

Программа 6.3

Адрес	Машинный код	Метка	Мнемокод	Комментарий
0800	3E F8	WAITK	MVI A, F8h	Записать 0 в младшие три разряда аккумулятора
0802	D3 07		OUT 07h	Записать 0 в выходное устройство
0804	DB06		IN06h	Получить число с выходного устройства
0806	E6 07		ANI 07h	Очистить старшие пять разрядов аккумулятора
0808	FE07		CPI 07h	Есть ли в младших трех разрядах аккумулятора 0?
080A	CA 0008		JZ WAITK	Если нет, то переход
080D	3EFF		MVI A, FFh	Зажечь светодиоды выходного устройства
080F	D3 05		OUT 05h	
0811	CF		RST1	

Алгоритм определения номера нажатой клавиши основывается на последовательной записи нуля в каждый из рядов матрицы клавиатуры. При наличии нуля в каждом ряду микроЭВМ определяет факт нажатия на клавиши, находящиеся в столбцах клавиатуры, принадлежащих анализируемому ряду. Если какая-либо клавиша нажата, то определяется ее номер по номеру разряда, в котором записан 0. Программа 6.4 определяет номер нажатой клавиши по номеру ряда клавиатуры, в котором обнаружена нажатая клавиша, и номеру разряда входного устройства, в котором записан 0.

Программа 6.4

Адрес	Машинный код	Метка	Мнемокод	Комментарий
0800	0EFE	START	MVI C, FEh	Начальная установка маски сканирования

Адрес	Машинный код	Метка	Мнемокод	Комментарий
0802	16 03		MVI D, 03h	Число рядов сканирования
0804	79	FROW	MOV A, C	Записать маску
0805	D3 07		OUT 07h	сканирования в выходное устройство
0807	DB06		IN 06h	Получить число из входного устройства
0809	E6 07		ANI 07h	Маскирование пяти младших разрядов
080B	FE07		CPI 07h	числа и проверка их на 0
080D	C2 1A08		JNZ FCOL	Если да, то переход
0810	79		MOV A, C	Изменить маску
0811	07		RLC	сканирования
0812	4F		MOV C, A	
0813	15		DCR D	
0814	C2 0408		JNZ FROW	Если не конец, то сканировать
0817	C3 0008		JMP START	дальше, иначе начать сначала
081A	47	FCOL	MOV B, A	Сохранить введенный код
081B	79		MOV A, C	Сдвинуть в старшую
081C	07		RLC	тетраду код
081D	07		RLC	сканирования
081E	07		RLC C	Сформировать
081F	07		RLC	код нажатой
0820	E6 70		ANI 70h	клавиши и
0822	B0		ORA B	переслать его
0823	D3 05		OUT 05h	в выходное устройство
0825	C3 0008		JMP START	Повторить

Задания для домашней подготовки

1. Ознакомьтесь со схемами подключения клавиатуры и дисплея к микроЭВМ, приведенными на рисунках 6.1 – 6.4.
2. Изучите принцип мультиплексного вывода информации на дисплей.
3. Изучите принцип определения номера нажатой клавиши на клавиатуре.
4. Изучите программы 6.1 – 6.4.
5. Разработайте программу мультиплексного последовательного вывода лишь одной цифры, задаваемой с входного устройства микроЭВМ на разные ячейки дисплея. Для задания времени высвечивания цифры на каждой ячейке дисплея используйте подпрограмму временной задержки. Определите, на что будет влиять время задержки.

6. Дополните программу 6.4 так, чтобы номер нажатой клавиши отображался в одной из ячеек дисплея.

7. Разработайте программу передачи управления по заданным адресам в зависимости от нажатой клавиши на клавиатуре. Адреса передачи управления в зависимости от номера клавиши на клавиатуре задаются таблицей 6.1. Адрес передачи управления записывается в регистры *H, L*.

Таблица 6.1 – Адреса передачи управления в зависимости от номера клавиши на клавиатуре

Символ	Код	Символ	Код
<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>
0	3F	A	77
1	06	B	7C
2	5B	C	39
3	4F	D	5E
4	66	E	79
5	6D	F	71
6	7D	H	76
7	07	L	38
8	7F	P	73
9	6F	П	54

Задания к лабораторной работе

Задание 1. Исследовать программу 6.1.

1. Ввести в микроЭВМ программу 6.1.
2. Осуществить пуск программы. Проследить изменения во включаемых сегментах пятой цифры дисплея, меняя число на входном устройстве.
3. Записать и проверить коды символов и цифр, приведенных в таблице 6.1.

Задание 2. Исследование программы вывода информации на дисплей.

1. Ввести в микроЭВМ программу 6.2.
2. Записать по адресам 0900 – 0905 коды букв для вывода на дисплей сообщения НАЧАЛО.
3. Осуществить пуск программы и проверить правильность вывода сообщения на дисплей.
4. Ввести в микроЭВМ разработанную в п. 5 задания для домашней подготовки программу.
5. Осуществить пуск программы и проверить, что на дисплей выводится лишь одна цифра.

6. Установить в программе время включенного состояния цифры на каждой ячейке дисплея, равное 1 с (время задается подпрограммой задержки; проследить изменения в информации, выводимой на дисплей).

Задание 3. Исследовать программу обслуживания клавиатуры.

1. Ввести в микроЭВМ программу 6.4.
2. Осуществить пуск программы и проверить содержимое регистра В после каждого нажатия на клавишу.
3. Ввести в микроЭВМ программу, разработанную в п. 7 задания для домашней подготовки.
4. Осуществить пуск программы и проверить соответствие адреса, записанного в регистры HL, номеру нажатой клавиши.

Отчет должен содержать:

1. Схему подключения исследуемой клавиатуры к микроЭВМ.
2. Программы, разработанные в п. 5, 6, 7 задания для домашней подготовки.

Задания для самопроверки

1. Какие коды необходимо записать по адресам 0900 – 0905 для вывода на дисплей чисел 1, 2, 3, 4, 5, 6?
2. Как следует изменить программу 6.2, чтобы изображение на дисплее начало равномерно перемещаться?
3. Как следует изменить программу 6.2 для изменения направления сканирования индикаторов дисплея?

Литература

1. Технические средства микропроцессорных систем: Пер. с англ. / Дж. Кофрон – Москва: Мир, 1983.
2. Применение микропроцессорных устройств и микроЭВМ. / В. Л. Горбунов, Д. И. Панфилов. – Москва: Машиностроение, 1983.

ПРИЛОЖЕНИЯ

Микропроцессор серии К580 и его архитектура (программная модель и набор команд)

Микропроцессор (МП) серии К580 представляет собой однокристалльный 8-разрядный процессор, который размещен в 40-контактном корпусе, имеет классическую трехшинную структуру с отдельными шинами адреса, данных, управления и выполняет фиксированный набор команд. Существует несколько модификаций этого МП, различающихся исполнением корпуса, быстродействием и некоторыми другими техническими характеристиками. Данный микропроцессор является аналогом зарубежного прототипа Intel8080. Дальнейшее развитие этого микропроцессора привело к появлению на зарубежном рынке более мощных 8-разрядных МП Intel8085 и Z80, которые выполняют также команды МП К580 (их отечественные прототипы К1821ВМ85 и К1818ХМЗ, соответственно).

Для разработки программ необходимо знание архитектуры микропроцессора и МП системы. Архитектура отражает возможности прикладного использования микропроцессора (а не его техническую реализацию) и содержит описание программной модели МП системы и набора команд, с помощью которых обеспечивается программный доступ к элементам модели. Под программной моделью МП системы понимается совокупность программно-доступных элементов (регистров), объединенных в систему посредством укрупненных программных направленных связей и дополнительных элементов, обеспечивающих достижение функциональной законченности и целостного представления модели (рис. П.1). Программная модель МП системы на базе МП К580 содержит непосредственно модели МП памяти М (MEMORY) и портов ввода-вывода I/O (INPUT/ OUTPUT), объединенных тремя шинами: данных (ШД), адреса (ША) и управления (ШУ). Порты ввода-вывода используются для сопряжения с различными системными (внутренними и внешними, или периферийными) устройствами.

Модель МП содержит следующие элементы.

1. Шесть 8-разрядных регистров общего назначения (РОН) с однобуквенными именами В, С, D, E, H, L. Эти регистры программно-доступны как автономно, так и попарно: как три 16-разрядные регистровые пары с однобуквенными именами по первому, старшему регистру пары В – (BC), D – (DE), и H – (HL). Регистры и регистровые пары используются для временного хранения промежуточных данных, адресов и косвенной адресации основной памяти М (в качестве указателей памяти). Регистровая пара H преимущественно используется как указатель памяти в однобайтных командах МП причем ячейка памяти, адрес которой указан в H-паре, называется M-регистром. Этот регистр в функциональном отношении эквивалентен регистрам МП, но имеет большее время доступа.

2. 16-разрядный регистр указателя стека SP(STACK POINTER) для хранения адреса вершины программного стека, размещаемого в оперативной памяти М и обеспечивающего необходимую глубину вложения подпрограмм при обработке многоуровневых прерываний и модульном построении программ.

3. 16-разрядный регистр счетчика команд, или программный счетчик PC(PROGRAM COUNTER) – для хранения адреса текущей команды выполняемой программы. При естественном ходе выборки команд, последовательно размещенных в памяти, содержимое счетчика увеличивается от команды к команде на число, равное количеству ячеек, занимаемых в памяти выполненной командой.

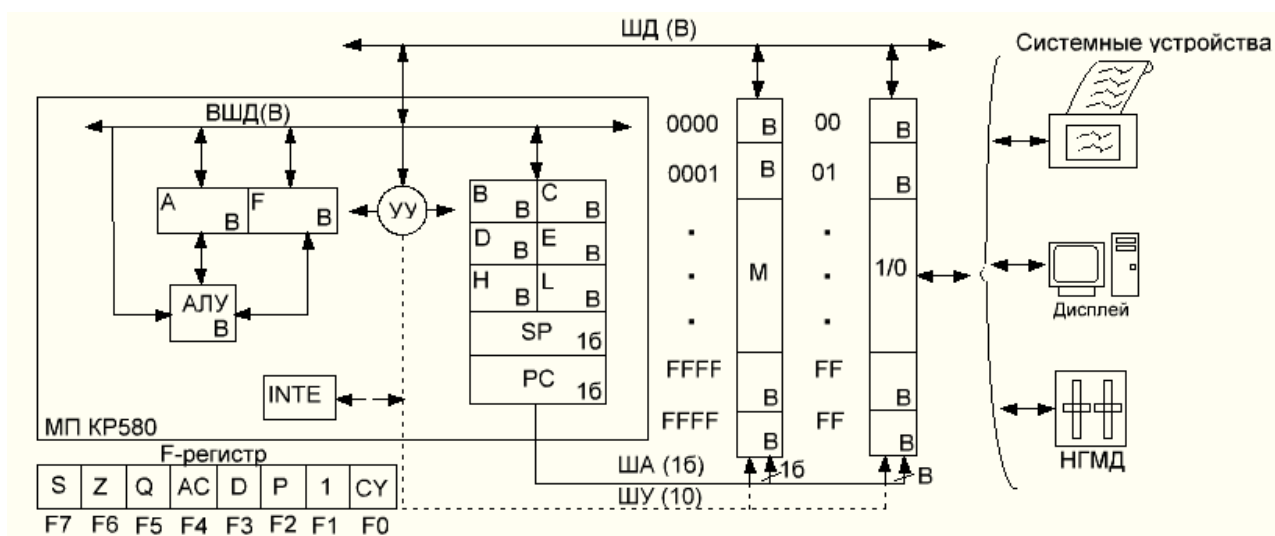


Рисунок П.1 – Программная модель МП системы

4. 8-разрядный А-регистр, или аккумулятор (накопитель), – основной рабочий регистр, используемый во всех арифметико-логических операциях, командах ввода-вывода данных и др. Совместно с F-регистром образует регистровую пару (AF) слова состояния процессора PSW (PROCESSOR STATUS WORD). Этот байт данных отражает результаты текущих преобразований в 8-разрядном арифметико-логическом устройстве (АЛУ);

5. 8-разрядный F-регистр, или регистр признаков (флагов), для хранения двоичных признаков (бит), отражающих некоторые особенности результата выполнения операции в АЛУ. Признаки могут использоваться последующими командами как для изменения естественной последовательности выборки команд из памяти (передача управления, или переход по программе), так и для модификации обрабатываемых данных. F-регистр фиксирует 5 различных признаков (3 разряда регистра не используются и содержат биты – константы):

S (SIGN) – бит знака, равен единице, если седьмой бит байта результата равен единице, то есть результат – отрицательное число;

Z (ZERO) – бит нуля, равен единице, если результат операции равен нулю;

AC (AUXILIARY CARRY) – бит вспомогательного переноса, равен единице, если при выполнении операции был перенос из третьего разряда АЛУ в четвертый, используется в команде десятичной коррекции содержимого А-регистра;

P (PARITY) – бит паритета, равен единице, если число единиц результата операции четное;

CY (CARRY) – бит переноса, равен единице, если при выполнении операции был перенос из седьмого разряда АЛУ или заем в этот разряд (разряды нумеруются, начиная с младшего, нулевого разряда, размещаемого в регистре в крайней правой позиции). Признак используется при обработке данных увеличенной разрядности;

6. 1-разрядный регистр (триггер) разрешения прерывания INTE (INTERRUPT ENABLE). Если триггер установлен в нуль, МП не реагирует на запросы прерывания; если в единицу – прерывание разрешено. После приема запроса прерывания или сброса системы триггер INTE автоматически сбрасывается в нуль и для разрешения обработки последующих запросов прерывания его необходимо программно вновь установить в единицу;

7. Связи между элементами МП и другими элементами МП системы:

– внутренняя 8-разрядная шина данных ВШД (8) – для передачи данных между элементами МП; внешняя двунаправленная 8-разрядная шина данных ШД (8) – для обмена данными между МП, памятью и портами ввода-вывода;

– однонаправленная 16-разрядная шина адреса ША (16) – для адресации памяти, портов; 10-разрядная шина управления ШУ (10), связанная с внутренним устройством управления (УУ) микропроцессора и предназначенная для временной коммутации элементов МП системы.

Модель памяти (М) представляет собой упорядоченную и пронумерованную последовательность 8-разрядных структурных элементов – ячеек памяти, или внешних регистров. 8-разрядное двоичное слово, хранимое в ячейке памяти (регистре), называется байтом, а отдельный двоичный разряд слова – битом. Номер ячейки памяти является ее адресом, 16-разрядная шина адреса МП позволяет обращаться к адресному пространству (максимальной совокупности адресуемых ячеек памяти) размером $2^{16} = 65536 = 64$ Кбайт ($K=1024$ – общепринятая константа). Для нумерации адресов памяти используется шестнадцатеричная система счисления с цифрами 0, 1, ..., 9, А, В, С, D, E, F. Каждый байт может быть представлен двумя полубайтами (старшей и младшей тетрадами бит), значения которых однозначно кодируются указанными шестнадцатеричными цифрами. При этом адрес любой ячейки памяти представляется 4-разрядным, а ее содержимое – 2-разрядным шестнадцатеричными числами (0000, 0001, ..., FFFF) и (00,01, ..., FF).

В модели памяти время доступа к содержимому любой ячейки памяти не зависит от значения ее адреса, и над каждой ячейкой памяти может быть выполнена пара операций: запись байта в ячейку и чтение ее содержимого. Память такого типа называют запоминающим устройством с произвольной

выборкой (ЗУПВ) или оперативным запоминающим устройством (ОЗУ) (оперативной памятью). Эта память используется для хранения программ, исходных, промежуточных и результирующих данных.

В реальных МП системах фактическое количество ячеек памяти, емкость, или рабочее пространство памяти, может быть меньше адресного пространства и, кроме того, разделено на части по каким-либо конструктивным или функциональным признакам. Например, часть памяти может быть предназначена только для операции чтения хранимой информации. Такая память – постоянное (или перепрограммируемое постоянное – ППЗУ) запоминающее устройство (ПЗУ) – используется для хранения программ и констант. Для учета при программировании подобных реальных ограничений на использование адресного пространства применяют карту памяти – графическое распределение рабочего пространства памяти между отдельными блоками последовательных ячеек, сгруппированных по выделенным признакам.

Модель портов ввода-вывода представляет собой, как и модель памяти, упорядоченную и пронумерованную последовательность 8-разрядных регистров. В системе посредством однобайтных шестнадцатеричных адресов (00, 01, ..., FF) адресуются до 256 портов ввода/вывода информации. Каждое системное устройство обменивается информацией (данными, адресами, управляющими сигналами) с МП путем посылки байта информации через соответствующий порт в А-регистр или приема байта из А-регистра. В качестве системных устройств ввода-вывода могут рассматриваться как периферийные устройства типа, например, АЦПУ, НГМД или дисплея, так и внутренние устройства типа, например, программируемого контроллера прерываний КР580ВН59 или программируемого параллельного адаптера интерфейса КР580ВВ55.

Взаимодействие между элементами МП, памяти М и портов ввода-вывода I/O в программной модели МП системы сводится к выполнению четырех операций: записи информации в М и I/O из регистра МП и ее чтения из М и I/O в регистр МП. Последовательность функциональных взаимодействий между элементами программной модели МП системы определяется набором команд МП.

Выполнение команды в МП системе, как и в большинстве вычислительных систем, состоит из двух крупных фаз: выборки адресованной команды из памяти и ее выполнения.

Длительность командного цикла индивидуальна для каждой команды и может быть выражена количеством тактов или периодов синхронизации МП. Зная частоту генератора, можно определить реальное время выполнения команды и фрагментов программ. Длительность команд, оперирующих в МП системе с содержимым внутренних регистров, существенно короче, чем команд, использующих внешние регистры. Поэтому при необходимости минимизировать время выполнения программы желательно основную обработку информации вести с использованием внутренних регистров.

Функциональные возможности набора команд определяются, в первую очередь, форматом данных, команд и способами адресации операндов в командах. Для МП характерна малая разрядность основных структурных элементов (регистров), что приводит к использованию в МП преимущественно одноадресных, реже двухадресных (типа, например, команд межрегистровых пересылок) команд и различных способов не прямой адресации памяти, сокращающих длину командных слов и размеры программ.

Формат данных (базового информационного слова), принятый в МП КР580, – 8-разрядное двоичное слово (V_7, \dots, V_0) – байт, где V_7 – старший, а V_0 – младший разряды байта. Хранение и пересылка всех данных в МП системе реализуются в виде последовательности байтов (побайтно), а для представления данных повышенной разрядности используются программно-организуемые многобайтные слова. Байт может использоваться для представления как целых двоичных чисел без знака в диапазоне от 0 до 255, так и чисел со знаком: целых положительных в диапазоне от 0 до 127 и отрицательных в диапазоне от -1 до -128 при представлении чисел в дополнительном коде. Интерпретация числового значения байта (число со знаком или без него) осуществляется программным путем. Для чисел со знаком старший, седьмой разряд байта интерпретируется как знаковый бит: если он равен 0, число положительное; если 1 – отрицательное. Кроме рассмотренных представлений байта в виде двоичных или шестнадцатеричных чисел возможна их интерпретация в качестве двоично–десятичных чисел (байт содержит две десятичные двоично-кодированные цифры) или алфавитно–цифровых символов, используемых для обмена с периферийными устройствами.

Формат команды (командного слова) зависит от типа операции и может быть одно-, двух- или трехбайтным. Байты многобайтной команды обязательно размещаются в соседних ячейках памяти, и адрес первого байта является адресом команды в целом. Первый байт команды представляет код операции, второй и третий байты – данные или адрес. Заметим, что в трехбайтных командах старший и младший байты адреса или 16-разрядных данных меняются местами, то есть вначале следует младший, а затем старший байт. Это является особенностью МП КР580, которую необходимо учитывать при программировании. К однобайтным относятся все команды межрегистровых пересылок (или команды типа регистр-регистр), ряд команд обращения к памяти, арифметических и логических операций, команды сдвига, возврата из подпрограммы, обращения к стеку, разрешения и запрещения прерываний; к двухбайтным – команды с непосредственными данными и ввода-вывода; к трехбайтным – команды переходов, вызова подпрограмм, загрузки регистровых пар и прямой записи в память. В МП КР580 используются пять различных способов адресации.

Прямая адресация, при которой второй и третий байты команды содержат прямой адрес операнда в памяти.

Регистровая, или неявная, адресация, при которой адрес регистра источника и (или) приемника операнда определяется кодом команды.

Регистровая косвенная адресация, при которой адрес операнда находится в регистровой паре, адресуемой кодом команды.

Непосредственная адресация, при которой операнд размещается во втором байте – для двухбайтной команды или во втором и третьем байтах – для трехбайтной команды.

Стековая адресация, при которой адрес определяется указателем стека. Она отличается от регистровой косвенной адресации тем, что при обращении к памяти происходит запись или чтение двух байтов, а содержимое указателя стека автоматически, соответственно, уменьшается или увеличивается на 2. Информация в стеке хранится в том порядке, в котором туда поступает, а извлекается в обратном порядке по принципу: “последним пришел – первым вышел”.

Набор команд МП КР580 содержит 78 базовых команд, различающихся мнемоническим обозначением кода операции, а в целом включает 244 различные их модификации. Условные обозначения, используемые для описания набора команд, приведены в таблице П.1. В таблице П.2 дано описание набора команд. В таблицах П.3 – П.5 представлены шестнадцатеричные значения кода операций команд с регистровой адресацией.

Команды МП К580 подразделяются на 5 функционально специализированных групп:

1. Команды передачи данных используются для пересылки данных из регистра в регистр, из памяти в регистр (регистровую пару) и из регистра (регистровой пары) в память.

2. Команды арифметических операций применяются для двоичных операций сложения, вычитания, инкрементирования и декрементирования содержимого регистра (регистровой пары) или ячейки памяти, а также для сложения двоично-десятичных чисел.

3. Команды логических операций выполняют операции И, ИЛИ, ИСКЛЮЧАЮЩЕЕ ИЛИ (неравнозначность), сравнения и сдвига.

4. Команды передачи управления используются при условных и безусловных переходах в программе, вызовах подпрограмм и возвратах из них, рестартах подпрограмм при организации многоуровневых векторных прерываний.

5. Команды стека, ввода-вывода и управления применяются для управления прерыванием, останова МП, ввода-вывода данных и обращения к стеку, его указателю и вершине.

Особый тип образуют команды рестарта RST0, ..., RST7 из четвертой группы. Эти однобайтные команды формируются, как правило, специальными аппаратными средствами (контроллерами прерываний) в режиме прерывания работы МП. В этот режим МП входит, получив сигнал прерывания от системного устройства. При этом МП заканчивает цикл выполнения текущей

команды (в программном счетчике содержится уже адрес следующей команды программы) и считывает с ШД (8) код команды рестарта, который выставляется контроллером прерываний.

Действие этой аппаратной команды аналогично действию команды вызова подпрограммы, но отличается, во-первых, тем, что ограничено возможностью вызова в соответствии с вектором прерывания лишь одной из восьми подпрограмм, размещаемых в первых 64 ячейках оперативной памяти, и, во-вторых, тем, что при выполнении команды рестарта не происходит, как при выполнении любой другой команды, инкрементирования содержимого программного счетчика. Благодаря этому в стек записывается адрес очередной команды прерываемой программы, а по команде возврата (в подпрограмме обслуживания прерывания) этот адрес извлекается из стека и управление передается в ту точку прерванной программы, куда поступило прерывание и где продолжается выполнение основной программы.

Команды рестарта – это аппаратные аналоги команд вызова подпрограмм, инициализируемые сигналом прерывания в любой точке и в любой момент времени выполняемой программы. При таком неожиданном вызове подпрограммы необходимо обеспечить сохранение в стеке слова состояния процессора и содержимого РОН для возобновления после обслуживания прерывания нормального хода прерванной программы. МП КР580 обладает также возможностью отработки в режиме прерывания не только команд рестарта, но любой последовательности команд, аппаратно формируемой в этом режиме контроллером прерываний. В частности, они воспринимают аппаратно формируемую трехбайтную команду CALLB2B3, что позволяет осуществлять вызов подпрограммы из любой области памяти.

Набор команд МП КР580 ориентирован на решение сложных задач управления данными, в частности, создания систем, работающих в реальном масштабе времени. Эффективному решению этих задач способствует широкий спектр команд передачи данных, передачи управления, вызова и рестарта подпрограмм, управления прерыванием и обращения к стеку программируемой глубины.

Таблица П.1 – Условные обозначения в наборе команд

Обозначение	Комментарий
<i>1</i>	<i>2</i>
<i>R1, R2, R</i>	Один из регистров <i>A, B, C, D, E, H, L</i>
<i>M</i>	Ячейка памяти, адресуемая содержимым <i>H</i> – пары регистров
<i>RM1, RM2, RM</i>	Один из регистров <i>A, B, C, D, E, H, L</i> , ячейка памяти <i>M</i> или байт <i>B2</i>
<i>RR</i>	Одна из регистровых пар <i>B – (B, C), D – (D, E), H – (H, L)</i> , или указатель стека <i>SP</i>

Окончание таблицы П.1

1	2
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 H – паре регистров
$0,1$	Состояния бита
\leftarrow	Оператор присваивания: элемент слева от символа \leftarrow заменяется элементом, находящимся справа от него
\leftrightarrow	Обмен элементами
$\wedge \vee$	Логические операции И, ИЛИ, ИСКЛЮЧАЮЩЕЕ ИЛИ
\bar{X}	Операция инверсии значения X
$+, -, *$	Знаки алгебраических сложения, вычитания, умножения. В колонке "Биты признаков" знак "+" означает, что бит устанавливается в зависимости от результата операции, а знак – что бит не изменяется в данной операции
$\frac{n_1}{n_2}$	Количество тактов, где n_1 – число тактов при невыполнении условия, n_2 – при его выполнении

Таблица П.2 – Набор команд микропроцессора КР580

Формат, байтов	Время, тактов	Код	Мнемокод	Наименование команды	Описание операций					
						S	Z	AC	P	CY
1	2	3	4	5	6	7	8	9	10	11
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 R, 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 таблицы П.2

1	2	3	4	5	6	7	8	9	10	11
<i>Команды арифметических операций</i>										
1	4	EB	XGHG	Обмен между H и D-парами регистров	$(H) \leftrightarrow (D),$ $(L) \leftrightarrow (E)$	-	-	-	-	-
1	4	*	ADD R	Сложение содержимых регистра и A-регистра	$(A) \leftarrow (A) + (R)$	+	+	+	+	+
1	7	86	ADD M	Сложение содержимых ячейки в памяти в 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	$(A) \leftarrow (A) + (R) + (C$ $Y)$	+	+	+	+	+
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 таблицы П.2

1	2	3	4	5	6	7	8	9	10	11
2	7	D6	SUI B2	Вычитание непосредственных данных из содержимого A-регистра	$(A) \leftarrow (A) - (B2)$	+	+	+	+	+
1	4	*	SUB 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	DE	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 таблицы П.2

1	2	3	4	5	6	7	8	9	10	11
1	10	*	DAD RR	Сложение содержимых регистровой пары и H – 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 $(A) \leftarrow (A) + 6 \cdot 2^4$	+	+	+	+	+
<i>Команды логических операций</i>										
1	4	*	ANAR	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	AE	XRA M	Поразрядное ИСКЛЮЧАЮЩЕЕ ИЛИ над содержимым ячейки памяти и A-регистра	$(A) \leftarrow (A) \vee ((H,L))$	+	+	0	+	0

Продолжение таблицы П.2

1	2	3	4	5	6	7	8	9	10	11
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	ORIB2	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	CMPM	Сравнение содержимых ячейки памяти и A-регистра	—”—”—	+	+	+	+	0
2	7	EF	CPIB2	Сравнение непосредственных данных с содержимым A-регистра	—”—”—	+	+	+	+	0

Продолжение таблицы П.2

1	2	3	4	5	6	7	8	9	10	11
1	4	07	RLC	Циклический сдвиг влево содержимого А-регистра	$(A_{i+1}) \leftarrow (A_i),$ $(A_0) \leftarrow (A_7),$ $(CY) \leftarrow (A_7)$	-	-	-	-	+
1	4	17	RAL	Циклический сдвиг влево содержимого А-регистра через перенос	$(A_{i+1}) \leftarrow (A_i),$ $(A_0) \leftarrow (CY),$ $(CY) \leftarrow (A_7)$	-	-	-	-	+
1	4	0F	RRC	Циклический сдвиг вправо содержимого А-регистра	$(A_i) \leftarrow (A_{i+1}),$ $(A_7) \leftarrow (A_0),$ $(CY) \leftarrow (A_0)$	-	-	-	-	+
1	4	1F	RAR	Циклический сдвиг вправо содержимого А-регистра через перенос	$(A_i) \leftarrow (A_{i+1}),$ $(A_7) \leftarrow (CY),$ $(CY) \leftarrow (A_0)$	-	-	-	-	+
1	4	2F	CMA	Дополнение содержимого А-регистра	$(A) \leftarrow (\overline{A})$	-	-	-	-	-
1	4	3F	CMC	Дополнение содержимого признака переноса	$(CY) \leftarrow (\overline{CY})$	-	-	-	-	+
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	$(PC) \leftarrow (B3, B2)$	-	-	-	-	-
3	10	DA	JC B2B3	Переход, если перенос	Если условие верно, то $(PC) \leftarrow (B3, B2)$, в противном случае $(PC) \leftarrow (PC) + 3$	-	-	-	-	-
3	10	D2	JNC B2B3	Переход, если не перенос	— ” —	-	-	-	-	-

Продолжение таблицы П.2

1	2	3	4	5	6	7	8	9	10	11
3	10	CA	JZ B2B3	Переход, если нуль	__"__"	-	-	-	-	-
3	10	C2	JNZ B2B3	Переход, если не нуль	__"__"	-	-	-	-	-
3	10	F2	JPB2B3	Переход, если плюс	__"__"	-	-	-	-	-
3	10	FA	JMB2B3	Переход, если минус	__"__"	-	-	-	-	-
3	10	EA	JPEB2B3	Переход, если четно	__"__"	-	-	-	-	-
3	10	E2	JPOB2B3	Переход, если нечетно	__"__"	-	-	-	-	-
3	17	CD	CALL B2B3	Безусловный вызов подпрограммы	((SP)-1)←(PCH), ((SP)-2)←(PCL), (SP)←(SP)-2, (PC)←(B3,B2)	-	-	-	-	-
3	11/17	DC	CC B2B3	Вызов подпрограммы, если перенос	Если условие верно, то ((SP)-1)←(PCH), ((SP)-2)←(PCL), (SP)←(SP)-2, (PC)←(B3,B2), в противном случае (PC)←(PC) + 3	-	-	-	-	-
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	__"__"	-	-	-	-	-

Продолжение таблицы П.2

1	2	3	4	5	6	7	8	9	10	11
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	(PCL) \leftarrow ((SP)), (PCH) \leftarrow ((SP)+1), (SP) \leftarrow (SP)+2	-	-	-	-	-
1	5/11	D8	RC	Возврат, если перенос	Если условие верно, то (PCL) \leftarrow ((SP)), (PCH) \leftarrow ((SP)+1), (SP) \leftarrow (SP)+2, в противном случае (PC) \leftarrow (PC)+1	-	-	-	-	-
1	5/11	DO	RNC	Возврат, если не перенос	__"__"	-	-	-	-	-
1	5/11	C8	RZ	Возврат, если нуль	__"__"	-	-	-	-	-
1	5/11	CO	RNZ	Возврат, если не нуль	__"__"	-	-	-	-	-
1	5/11	FO	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	__"__"	-	-	-	-	-
1	11	C7	RST0	Рестарт по 0-му уровню	((SP)-1) \leftarrow (PCH), ((SP)-2) \leftarrow (PCL), (SP) \leftarrow (SP)-2, (PC) \leftarrow 8·N	-	-	-	-	-

Продолжение таблицы П.2

1	2	3	4	5	6	7	8	9	10	11
1	11	CF	RST1	Рестарт по 1–му уровню	$\underline{\quad} \text{''} \underline{\quad} \text{''} \underline{\quad}$	–	–	–	–	–
1	11	D7	RST2	Рестарт по 2–му уровню	$\underline{\quad} \text{''} \underline{\quad} \text{''} \underline{\quad}$	–	–	–	–	–
1	11	DF	RST3	Рестарт по 3–му уровню	$\underline{\quad} \text{''} \underline{\quad} \text{''} \underline{\quad}$	–	–	–	–	–
1	11	E7	RST4	Рестарт по 4–му уровню	$\underline{\quad} \text{''} \underline{\quad} \text{''} \underline{\quad}$	–	–	–	–	–
1	11	EF	RST5	Рестарт по 5–му уровню	$\underline{\quad} \text{''} \underline{\quad} \text{''} \underline{\quad}$	–	–	–	–	–
1	11	F7	RST6	Рестарт по 6–му уровню	$\underline{\quad} \text{''} \underline{\quad} \text{''} \underline{\quad}$	–	–	–	–	–
1	11	FF	RST7	Рестарт по 7–му уровню	$\underline{\quad} \text{''} \underline{\quad} \text{''} \underline{\quad}$	–	–	–	–	–
1	5	E9	PCHL	Запись содержимого Н – пары регистров	(PCH) \leftarrow (H), (PCL) \leftarrow (L)	–	–	–	–	–
<i>Команды стека, ввода–вывода и управления</i>										
1	11	*	PUSHRR	Запись в стек содержимого регистровой пары	((SP) -1) \leftarrow (RRH), ((SP) -2) \leftarrow (RRL), (SP) \leftarrow (SP) -2	–	–	–	–	–
1	11	F5	PUSHPSW	Запись в стек слова состояния процессора	((БЗ) -1) \leftarrow (А)б (БЗ) \leftarrow (БЗ) -2 б ((БЗ) -2) \leftarrow (А)	–	–	–	–	–
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)), (А) \leftarrow ((SP) $+1$), (SP) \leftarrow (SP) $+2$	+	+	+	+	–
1	18	E3	XTHL	Обмен между вершиной стека и Н–парой регистров	(L) \leftarrow ((SP)), (H) \leftarrow ((SP) $+1$)	–	–	–	–	–
1	5	F9	SPHL	Запись содержимого Н – пары регистров в указатель стека	(SP) \leftarrow (H, L)	–	–	–	–	–
2	10	DB	INB2	Ввод данных	(А) \leftarrow ((B2))	–	–	–	–	–

Продолжение таблицы П.2

<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>	<i>7</i>	<i>8</i>	<i>9</i>	<i>10</i>	<i>11</i>
2	10	D3	OUT B2	Вывод данных	$((B2)) \leftarrow (A)$	–	–	–	–	–
1	4	FB	EI	Разрешение прерывания	$(INTE) \leftarrow 1$	–	–	–	–	–

Таблица П.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

¹Команда MOV_R, R эквивалентна NOP.

²Команда MOV_M, M запрещена, поскольку ее код совпадает с HLT.

Таблица П.4 – Коды команд с регистрами

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

Таблица П.5 – Коды команд с регистровыми парами

Команда	RR			
	BC	DE	HL	PSW
LXIRR	01	11	21	31
LDAXRR	0A	1A	–	–
STAXRR	02	12	–	–
INX RR	03	13	23	33
DCXRR	0B	1B	2B	3B
DAD RR	09	19	29	39
PUSH RR	C5	D5	E5	–
POP RR	C1	D1	E1	–