

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Тульский государственный университет»

Политехнический институт
Кафедра «Промышленная автоматика и робототехника»

Утверждено на заседании кафедры
«Промышленная автоматика и робототех-
ника»

« 17 » января 2023г., протокол № 2

И.о заведующего кафедрой

 О.А.Ерзин

МЕТОДИЧЕСКИЕ УКАЗАНИЯ
по выполнению лабораторных работ
по дисциплине (модулю)
«Программирование микропроцессоров»

основной профессиональной образовательной программы
высшего образования – программы бакалавриата

по направлению подготовки
15.03.02 Технологические машины и оборудование

с направленностью (профилем)
Информационно-измерительные и управляющие системы
технологических машин

Форма обучения: очная

Идентификационный номер образовательной программы: 150302-01-22

Тула 2023 год

Разработчик(и) методических указаний

Зайчиков Игорь Вячеславович, канд.техн.наук, доц.

(ФИО, должность, ученая степень, ученое звание)



(подпись)

(ФИО, должность, ученая степень, ученое звание)

(подпись)

Содержание

Введение

1. Общие требования и методические указания

2. Правила техники безопасности при выполнении лабораторных работ

3. Лабораторная работа №1.

Изучение архитектуры учебного микропроцессорного комплекта УМПК-80

4. Лабораторная работа №2.

Изучение ассемблерных команд, машинных кодов и способов программирования УМПК-80

5. Лабораторная работа №3.

Разработка программ для микро-ЭВМ УМПК-80 с использованием подпрограмм и стека для управления звуковым сигналом.

6. Лабораторная работа №4.

Изучение принципов подключения светодиодного дисплея к микро-ЭВМ

7. Лабораторная работа №5.

Изучение принципов подключения клавиатуры к микро-ЭВМ

8. Лабораторная работа №6.

Разработка арифметических программ на Ассемблере МП БИС серии КР 580

ВВЕДЕНИЕ

Широкое применение микропроцессоров при построении систем управления открыло широкие возможности для робототехники. Встроенные микроЭВМ на основе микропроцессоров позволяют существенно расширить функциональные возможности современных роботов, увеличить их надежность, снизить стоимость, улучшить формы общения человека-оператора и промышленного робота.

Все это делает актуальной задачу качественной подготовки студентов специальности "Робототехнические системы" по дисциплине "Микропроцессорные устройства управления". В рамках этого курса студенты изучают устройство микропроцессора БИС серии K580, устройство микроЭВМ на этом микропроцессоре, его систему команд и программирование на машинном ассемблере. Изучение теоретического материала курса подкреплено проведением практических и лабораторных занятий. Лабораторные работы очень эффективно проводить с использованием учебной микроЭВМ, УМПК-80, построенной на базе микропроцессора БИС K580BM80. С помощью микроЭВМ УМПК-80 студенты изучают особенности работы однокристальных микропроцессоров с фиксированным подбором команд, вырабатывают навыки программирования таких микропроцессоров, исследуют программные и аппаратные средства ввода и отображения информации в микро

Настоящие лабораторные работы являются второй частью цикла работ, посвященных практическому освоению студентами основ построения и программирования микропроцессорных систем и содержат три двухчасовые лабораторные работы. Они являются методическим руководством, позволяющим развить у студентов навыки работы с микропроцессорными системами, построенными на базе микропроцессора с фиксированным набором команд БИС серии K580 и закрепить теоретический материал соответствующего курса. Материал соответствует рабочей программе курса "Микропроцессорные устройства управления", содержит необходимые теоретические сведения, иллюстрированные конкретными примерами и позволяет студентам приобрести навыки самостоятельного программирования микропроцессоров с фиксированным набором команд, широко используемых при построении систем управления цикловых систем управления промышленными роботами.

1. ОБЩИЕ ТРЕБОВАНИЯ И МЕТОДИЧЕСКИЕ УКАЗАНИЯ.

Учебная микроЭВМ УМПК-80 - это сложное техническое устройство, позволяющее со всей наглядностью изучать архитектуру микроЭВМ и исследовать основные информационные процессы, протекающие в ней при выполнении рабочей программы, благодаря наличию светодиодной индикации всех шин микроЭВМ, адресной, данных и управления, а также благодаря наличию режимов работы: покомандного, потактового. Программирование микроЭВМ ведется в машинных кодах микропроцессора БИС K580BM80, и после освоения этой части лабораторного практикума студенты, как правило, имеют хорошие навыки программирования микропроцессоров. Однако следует иметь в виду повышенные требования к внимательности разработчика программ в машинных кодах, к тщательности подготовки и ввода программ и данных в микроЭВМ, поскольку в УМПК-80 нет программы редактора, облегчающей работу программиста.

Каждая лабораторная работа из предложенного цикла содержит теоретический материал. При подготовке к лабораторной работе следует изучить теоретические сведения, уяснить основные положения, законспектировать ключевые моменты, необходимые для выполнения заданий.

По завершения всего объема данного цикла лабораторных работ следует приступить к оформлению отчета. Сдача выполненных работ производится в конце каждого цикла. Отчет по лабораторной работе должен содержать:

1. Название, цель и задачи лабораторной работы.
2. Описание и назначение используемых в работе программных средств.
3. Блок-схемы алгоритмов решаемых задач и тексты разработанных программ с подробным описанием.
4. Результаты выполнения созданных программ в виде последовательности чисел соответствующих каждой модификации программы, таблицы значений, либо подробного описания действий, производимых микроЭВМ по программе пользователя над имитаторами выходных устройств.
5. Ответы на вопросы в заданиях на каждую лабораторную работу.

2 . ПРАВИЛА ТЕХНИКИ БЕЗОПАСНОСТИ ПРИ ВЫПОЛНЕНИИ ЛАБОРАТОРНЫХ РАБОТ.

2.1. При эксплуатации и техническом обслуживании учебной микроЭВМ УМПК-80 обязательно соблюдать "Правила технической эксплуатации электроустановок" и "Правила техники безопасности при эксплуатации электроустановок".

2.2. Наладочные работы, осмотры, ремонт производить только при полном отключения микроЭВМ от сети.

ЗАПРЕЩАЕТСЯ! Заменять предохранители при включенной в сеть микроЭВМ, эксплуатировать микроЭВМ без заземления.

ВНИМАНИЕ! Повторное включение микроЭВМ следует производить не ранее чем через 5 с после погасания светодиода "Перегрузка".

3. Лабораторная работа №1.

Изучение архитектуры учебного микропроцессорного комплекта УМПК-80

1. ЦЕЛЬ И ЗАДАЧИ РАБОТЫ.

Целью работы является освоение функционирования архитектуры учебного микропроцессорного комплекта УМПК-80

Задачей работы является изучение особенностей аппаратуры и правил функционирования имеющихся блоков учебного микропроцессорного комплекта УМПК-80

2. ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

2.1. Функциональная схема учебного микропроцессорного комплекта УМПК-80 (плакат)

2.2. Учебная микроЭВМ УМПК-80. ПБА2.390.039.ПС. Паспорт.

3. ИСПОЛЬЗУЕМОЕ ОБОРУДОВАНИЕ.

Учебная микроЭВМ типа УМПК 80.

4. ЗАДАНИЕ НА РАБОТУ.

1. Ознакомиться и зарисовать функциональную схему учебного микропроцессорного комплекта УМПК-80

2. По паспорту ознакомиться и зафиксировать основные правила функционирования узлов схемы учебного микропроцессорного комплекта УМПК-80

3. Ознакомиться со служебными режимами функционирования схемы учебного микропроцессорного комплекта УМПК-80 для отладки программ.

4. Обратит особое внимание для дальнейшего программирования на устройства звукового сигнала, светодиодного дисплея, клавиатуры, регистровых портов ввода-вывода и системы индикации

5. КОНТРОЛЬНЫЕ ВОПРОСЫ.

1. Какие режимы клавиатуры для отладки программ доступны?

2. Что регистрируют светодиодные индикаторы УМПК-80?

3. Как вводятся машинные коды в память УМПК-80?

4. Какие основные узлы имеются в УМПК-80?

6. СОДЕРЖАНИЕ ОТЧЕТА

Отчёт должен содержать функциональную схему учебного микропроцессорного комплекта УМПК-80 и основные правила функционирования его узлов

7. БИБЛИОГРАФИЧЕСКИЙ СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.

1. Микропроцессоры: в 3-х кн. Кн. 3. Средства отладки, лабораторный практикум, задачник: Учеб. для втузов/Н.В.Воробьев, В.Л.Горбунов, А.В. Горячев и др.; Под ред. Л.Н. Преснухина. - К.: Высш.шк., 1936. - 351 с.

2. Учебная микроЭВМ УМПК-80. ПБА2.390.039.ПС. Паспорт.

4. Лабораторная работа №2.

Изучение ассемблерных команд, машинных кодов и способов программирования УМПК-80

1. ЦЕЛЬ И ЗАДАЧИ РАБОТЫ.

Целью работы является освоение способов адресации, системы команд и программирования однокристалльных микропроцессоров с фиксированным набором команд.

Задачей работы является изучение особенностей команд и их кодировки при использовании ассемблера и Машинных кодов

2. ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Способы адресации

Микропроцессор (МП) KP580BM80 является наиболее распространенным в настоящее время. Для большинства приложений, связанных с разработкой прикладных программ, достаточно рассмотреть программную модель данного МП. Знание программной модели рис.1 и системы команд МП являются необходимой информацией для обеспечения проектирования специализированных контроллеров и управляющих микроЭВМ на базе МП KP580BM80 с применением языка ассемблера.

Отличительной особенностью МП является возможность программного объединения 8 - битных регистров общего назначения (РОН) в 16 -битные регистровые пары. Данные пары могут хранить 16-битные операнды или быть задействованы в качестве указателей памяти. Основным указателем памяти является Н-пара, которая адресуется в командах как РОН с адресом 110. Ячейка памяти, адрес которой определяется содержимым Н-пары, обозначается М. Микропроцессор КР580ВМ80 может адресовать до 256 портов ввода и до 256 портов вывода. Максимальный размер прямо адресуемой памяти достигает 64 кбайта, рис. 2.

Наиболее простой и в тоже время неэкономичной является прямая адресация: в поле операнда содержится полный 16-битовый адрес памяти. Прямая адресация обеспечивает возможность обращения к любой ячейке адресного пространства МП.

Если возникает необходимость включить в программу фиксированные (непосредственные) значения кодов символов (данных) применяется непосредственная адресация. Команда МП при этом содержит код операции 1 байт и непосредственный, операнд - 2 байта. Допускается непосредственно адресовать 16-битные слова (В2 - младший байт, В3 - старший байт), применяемые для установки значений указателей памяти.

При использовании регистровой адресации значением операнда является содержимое адресуемого в команде РОН. Команда МП с таким режимом адресации однобайтовая и выполняется достаточно быстро. В однобайтовых командах могут адресоваться два РОН, так как их адреса, имеют всего по 3 бита.

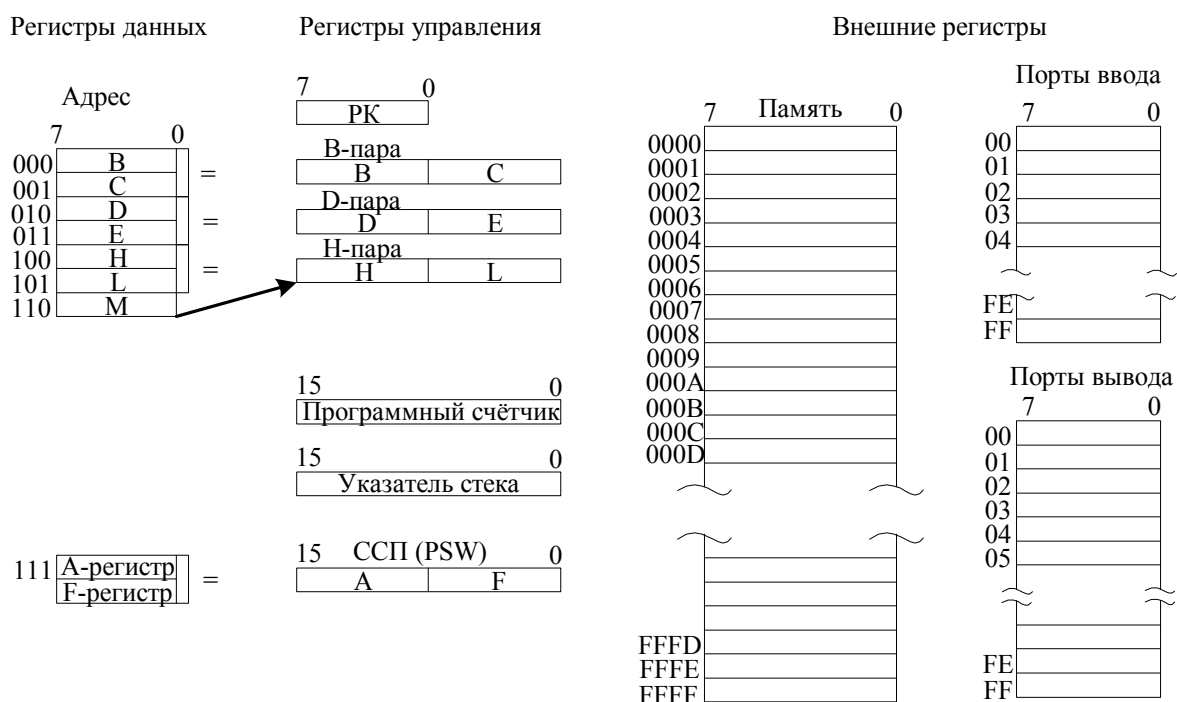


Рис. 1. Программная модель микропроцессора КР580ВМ80

В командах с неявной адресацией подразумевается, что операнд находится в определённом внутреннем регистре МП и не требует специальной адресации. Так все команды сдвига МП КР580ВМ80 оперируют содержимым аккумулятора. В операциях преобразования двоичных данных подразумевается, что одним из операндов является содержимое аккумулятора, и что результат операции помещается в аккумулятор. Неявная адресация применяется в унитарных операциях. В качестве примера можно привести команду инвертирования аккумулятора.

Косвенная адресация позволяет компактно адресовать все пространство памяти МП. В этом режиме поле операнда определяет не регистр с данными, а указатель памяти, содержащий адрес операнда. В МП серии КР580 основным указателем памяти служит Н-пара (регистры Н и L), адресуемая как РОН с адресом 110. В командах загрузки и запоминания аккумулятора в качестве указателей памяти могут применяться В- и D-пары. В однобайтовых командах загрузки в стек и извлечения из стека также используется косвенная адресация: память адресуется указателем стека. Косвенная адресация широко применяется при обработке данных типа "вектор" (последовательное расположение данных в памяти). В указатель памяти сначала загружается начальный адрес последовательности данных, а доступ к последующим элементам данных осуществляется с помощью инкремента указателя памяти.

Следует отметить, что в МП серии КР580 отсутствует индексная адресация. Данный способ адресации реализуется программным путем.

ПРИМЕР 1.1. Команда (с прямой адресацией) пересылки содержимого аккумулятора в ячейку памяти с адресом $\langle B_3 \rangle \langle B_2 \rangle$. Мнемоническое обозначение команды STA $\langle B_2 \rangle \langle B_3 \rangle$. Эта команда трёхбайтовая:

КОП	Адрес	
00110010	$\langle B_2 \rangle$	$\langle B_3 \rangle$

1 байт

2 байт

3 байт

Следует обратить внимание на то, что вторым байтом команды STA является младший байт адреса ячейки, памяти, т.е. если надо записать содержимое аккумулятора в ячейку памяти с адресом 0800, то соответствующая команда записывается в виде STA 00 08

В результате ее выполнения произойдет следующее перемещение информации внутри микроЭВМ;
[содержимое Ак] → [ячейку памяти адресом 0800]

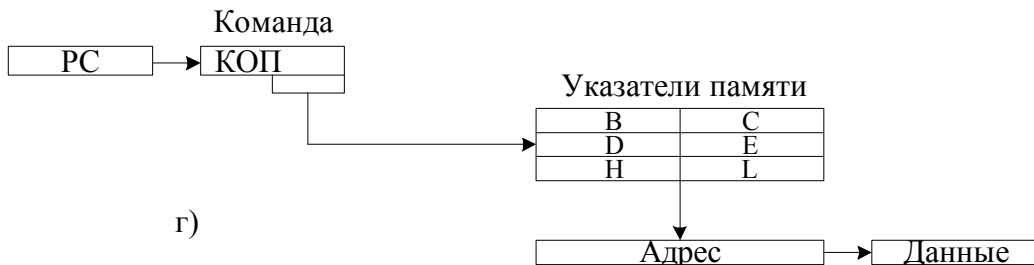
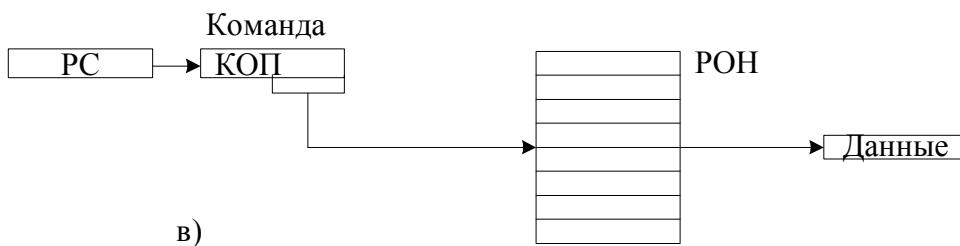
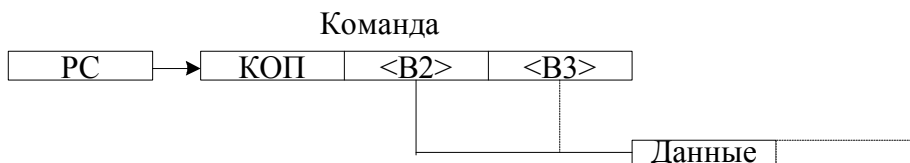
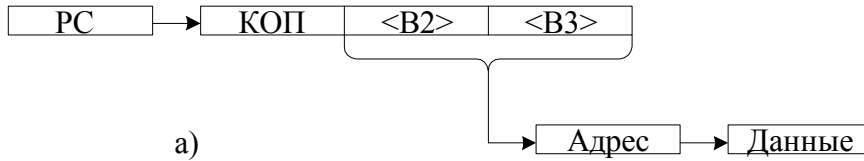


Рис. 2. Способы адресации МП КР580ВМ80

РС - программный счётчик;

КОП - код операции;

а - прямая адресация;

б - непосредственная адресация;

в - регистровая адресация;

г - косвенная адресация;

ПРИМЕР 1.2. Команда о прямой адресацией LDA<B₂><B₃> предназначена для загрузки аккумулятора содержимым ячейки, указанном во 2-м и 3-м байтах команды. Записать эту команду для случая, когда адрес ячейки памяти 0BD0, зарисовать схему перемещения информации в этом случае.

ПРИМЕР 1.3. Трехбайтовая команда с прямой адресацией <B₂> <B₃> предназначена для записи в две последовательно расположенные ячейки памяти содержимого регистровой пары HL. Причем содержимое регистра L заносится в ячейку памяти, адресуемую 2-м и 3-м байтами команды, а содержимое регистра H - в ячейку, адресуемую инкрементированным значением 2-го и 3-го битов. По команде SHLD B0 08

будут выполнены следующие действия:

[L] - ячейку памяти с адресом 08B0

[H] - ячейку памяти с адресом 08B1

ПРИМЕР 1.4. Команда с прямой адресацией LHLD $\langle B_2 \rangle \langle B_3 \rangle$ предназначена для загрузки регистровой пары HL содержимым двух рядом расположенных ячеек памяти. Записать эту команду для случая, когда старший адрес равен 08C1, и зарисовать схему передачи информации.

ПРИМЕР 1.5. Двухбайтная команда с непосредственной адресацией MVIB $\langle B_2 \rangle$ предназначена для загрузки регистра "B" содержимым второго байта команды.

КОП	Операнд
00DDD110	$\langle B_2 \rangle$
1-й байт	2-й байт

Фрагмент КОП DDD может принимать в двоичном коде значения от 000 до 111, адресуя тем самым один из восьми регистров общего назначения (РОН). В нашем случае произойдет следующая пересылка информации:

$[\text{содержимое 2-го байта команды}] \rightarrow [B]$

ПРИМЕР 1.6. Записать команду загрузки аккумулятора числом DC, используя команду с непосредственной адресацией, зарисовать схему передачи информации.

ПРИМЕР 1.7. Двухбайтные арифметические операции с непосредственной адресацией ADI $\langle B_2 \rangle$ и SUI $\langle B_2 \rangle$ предназначены для прибавления к аккумулятору и вычитания из аккумулятора соответственно второго байта команды. Схемы информационных процессов при выполнении этих команд таковы:

ADI $\langle B_2 \rangle$:

$[Ak] + [\langle B_2 \rangle] \rightarrow [Ak]$

SUI $\langle B_2 \rangle$:

$[Ak] - [\langle B_2 \rangle] \rightarrow [Ak]$

ПРИМЕР 1.8. Двухбайтные арифметические операции с непосредственной адресацией ACI $\langle B_2 \rangle$ и SBI $\langle B_2 \rangle$ предназначены для прибавления к аккумулятору с учётом переноса и вычитания из аккумулятора с учетом заёма соответственно второго байта команды. Схемы информационных процессов при выполнении этих команд таковы:

ACI $\langle B_2 \rangle$:

$[Ak] + [\langle B_2 \rangle] + [C] \rightarrow [Ak]$

SBI $\langle B_2 \rangle$:

$[Ak] - [\langle B_2 \rangle] - [C] \rightarrow [Ak]$

Здесь $[C] = 1$, если триггер переноса (заема) во флаговом регистре установлен в единичное состояние, и $[C] = 0$ в противном случае.

ПРИМЕР 1.9. Написать программу, используя изученные команды и способы адресации, выполняющие следующие действия:

к содержимому ячейки памяти 08B0 прибавить число D3,

результат занести в ячейку памяти 08B1, а затем вычесть из него с учетом заема содержимое ячейки 08B2.

ПРИМЕР 1.10. Однобайтная команда MOV R_1, R_2 с регистровой адресацией предназначена для пересылки содержимого одного регистра в другой. Схема выполняемой ею операции такова:

$[R_2] \rightarrow [R_1]$

Формат команды MOV R_1, R_2

КОП
01DDD SSS
1 байт

где фрагмент DDD принимает значения в двоичном

коде - от 000 до 111, адресуя тем самым 8 РОНов - приемников информации, а фрагмент SSS, изменяясь в тех же пределах, адресует 8 РОНов - источников информации.

ПРИМЕР 1.11. Однобайтные арифметические операции с регистровой адресацией ADDR (ADC R) и SUB R (SBB R) предназначены для прибавления и соответственно вычитания из аккумулятора содержимого регистра "R". В скобках указаны аналогичные команды, но учитывающие при выполнении операции значение бита переноса или заёма соответственно. Схемы информационных процессов при выполнении этих команд следующие:

ADD R:

$[Ak] + [R] \rightarrow [Ak]$

ADC R:

$$[Ak]+[R]+[C] \rightarrow [Ak]$$

SUB R:

$$[Ak]-[R] \rightarrow [Ak]$$

SBB R:

$$[Ak]-[R]-[C] \rightarrow [Ak]$$

ПРИМЕР 1.12. Однобайтные команды STAX RP и LDAX RP с косвенной адресацией предназначены для пересылки содержимого аккумулятора в ячейку памяти (и наоборот), адрес которой указан в регистровой паре "RP". Формат этих команд следующий:

КОП	LDAX RP	КОП
00RP1010		00RP0010
1 байт		1 байт

Здесь фрагмент "RP" принимает значения от 00 до 11 в двоичном коде, определяя одну из трех регистровых пар.

Схемы этих операций следующие:

STAX RP

$[Ak] \rightarrow [в\ ячейку\ памяти,\ адрес\ которой\ хранится\ в\ регистровой\ паре\ RP]$.

LDAX RP

$[содержимое\ ячейки\ памяти,\ адрес\ которой\ хранится\ в\ регистровой\ паре\ RP] \rightarrow [Ak]$

КОНТРОЛЬНЫЕ ПРИМЕРЫ:

1. $[0800]+[B]-DC \rightarrow [0801]$
2. $[C]-[0802]+2B \rightarrow [B]$
3. $[Ak]-[D]+[0803] \rightarrow [0804]$, адрес 0803 хранится в "BC"
4. $[Ak]+[E]-[0805] \rightarrow [B]$
5. $[0810, 0811]-[HL] \rightarrow [BC]$
6. $[HL]+[BC]-[DE] \rightarrow [0812, 0813]$
7. $[B]+[C]-[D] \rightarrow [0801]$, адрес 0801 хранится в "DE"
8. $[08CD]+[H]-12 \rightarrow [H]$
9. $[08C1]-[L]+23 \rightarrow [08C2]$, адрес 08C2 хранится в "HL"
10. $[D]-[Ak]+56 \rightarrow [08CB]$
11. $[H]-[08D8]+[CD] \rightarrow [B]$
12. $[E]-[B]+[Ak]-[08BB] \rightarrow [H]$

Система команд

Микропроцессор KP580BM80 является представителем семейства процессоров с фиксированным набором команд. Его система команд содержит 78 команд (табл.1), которые можно подразделить на 6 групп по функциональному назначению:

1. Команды пересылки информации;
2. Команды арифметических операций;
3. Команды логических операций и сдвигов;
4. Команды передачи управления;
5. Команды ввода/вывода;
6. Команды управления микропроцессором;

Рассмотрим подробнее эти группы команд.

Команды пересылки информации осуществляют двухсторонний обмен между внешней памятью и регистрами микропроцессора или между различными регистрами блока РОН. При этом содержимое источника информации и состояние триггеров флагового регистра не изменяются.

Команды арифметических операций обеспечивают возможность сложения и вычитания 8- битных операндов. Операции умножения и деления, а также операции с другими форматами данных реализуются с помощью программ. Состояния флажков при выполнении арифметических операций указаны в табл.1. Особенностью микропроцессора серии KP580 являются арифметические операции с 16- битными операндами. В них операнды считаются целыми без знака, и значения всех флажков не изменяются. Команды инкремента INX и декремента DCX позво-

ляют увеличить или уменьшить содержимое регистров (B,C), (D,E), (H, L) и указателя стека SP на единицу. Команда двойного сложения DAD суммирует содержимое регистровой пары (H, L) и адресуемого 16-битного регистра. Необходимо отметить, что выполнение команды DAD может устанавливать флажок переноса C.

Команды логических операций и сдвигов. Логические операции являются поразрядными, т.е. выполняются для всех 8 бит операндов. Не адресуемый операнд находится в аккумуляторе, туда же загружается результат операции. По результату операции модифицируются состояния всех флажков, кроме флажка переноса C, который никогда не может быть установлен в единицу и поэтому принудительно сбрасывается. Логические команды данной группы обеспечивают операции поразрядной конъюнкции, поразрядной дизъюнкции операндов, а также операцию "исключающего или". Команды сдвига обеспечивают сдвиг операнда влево или вправо только на один разряд. В зависимости от того, что помещается в освобождающийся при сдвиге бит и как используется выдвигающийся бит и флажок переноса, вводятся несколько команд сдвига: логического RLC, RRC и арифметического RAL и RAR.

Выполнение данных команд можно проследить по табл.1.

Команды передачи управления делятся на команды безусловной передачи управления, условной передачи управления и вызова подпрограмм.

Команда безусловной передачи управления JMP состоит из байта кода операции и двух байтов полного адреса и позволяет передать управление любой ячейке памяти адресного пространства.

Команды условной передачи управления позволяют осуществить передачу управления только при удовлетворении некоторого условия, заданного в коде операции. Если условие не удовлетворяется, то передачи управления не происходит, а выполняется следующая по порядку команда. Проверяемым условием является текущее состояние одного из флажков, указываемых в коде операции. В системе команд микропроцессора KP580BM80 предусмотрено восемь команд разветвления: JZ, JNZ, JC, JNC, JM, JP, JPE, JPO.

Команда вызова подпрограммы CALL состоит из байта - кода операции и двух байт - адреса перехода, представляющего собой начальный адрес подпрограммы. При выполнении вызова следует временно заполнить адрес команды, следующий за командой CALL (адрес возврата), загрузить адрес перехода команды CALL в программный счетчик PC, а после выполнения подпрограммы передать адрес возврата в PC. Последняя функция выполняется командой - RET - командой возврата из подпрограммы. Следует отметить, что для заполнения команды CALL, необходим 16-битный регистр для временного запоминания адреса возврата. Если вызванная подпрограмма вызывает еще одну, а эта вторая - третью и т.д., то проблема сводится к запоминанию адресов возврата в порядке вызовов подпрограмм и возврату их в программный счетчик PC по мере выполнения подпрограмм. Для этого необходим блок регистров, работающий таким образом, что последний включенный в него адрес возврата извлекается первым. Такой блок регистров называется стеком. В микропроцессоре KP580BM80 для стека выделяется область адресного пространства, отмеченная указателем стека, а глубина его определяется выражением: $N = \text{FFFF} - \langle \text{SP}_0 \rangle$.

где $\langle \text{SP}_0 \rangle$ - начальное значение указателя стека;

FFFF - максимальное значение адреса в шестнадцатеричной системе счисления в адресном пространстве микропроцессора KP580BM80.

Для работы со стеком имеется пара команд PUSH и POP (см. табл.1). В системе команд рассматриваемого микропроцессора имеются команды обращения к подпрограммам, которые проверяют единичное и нулевое значение каждого из четырех флажков:

флажок Z: вызовы CZ, CNZ; возможны RZ, RNZ;
 флажок C: вызовы CC, CNC; возвраты RC, RNC;
 флажок S: вызовы CM, CP; возвраты RM, RP;
 флажок P: вызовы CPE, CPO; возвраты RPE, RPO;

Команда ввода/вывода IN и OUT. Второй байт этих команд представляет собой 8-битный адрес порта ввода/вывода, а получателем может быть только аккумулятор.

Команды управления микропроцессором применяются для задания режима работы микропроцессора. Команда HLT – останов вызывает прекращение выполнения программы и переводит микропроцессор в состояние останова. После этого вывести его из состояния останова можно только сигналом общего сброса RESET.

Команды EI - разрешения прерываний и DI - запрещения прерываний позволяют программно заказывать или отменять режим прерываний.

Команда NOP не производит никаких действий, кроме инкремента программного счётчика.

Команды пересылки информации

ПРИМЕР 2.1. Команда MOV R, M. (Move from memory) загружает регистр R (с номером DDD) содержимым ячейки памяти, адрес которой хранится в регистровой паре HL.

Схема команды:

$\langle \text{HL} \rangle \rightarrow [\text{R}]$

Код операции:

01DDD110

Эта команда однобайтная, выполняется за два цикла.

Варианты команды:

ПРИМЕР 2.2.
(Move to memory) за-
мента, адрес которой
вой паре HL, содер-
номером (SSS).

Схема команды:

$[R] \rightarrow [(HL)]$

Код операции:

01110SSS

Эта команда однобайтная, выполняется за два цикла.

Варианты команды:

MOV A, M
MOV B, M
MOV C, M
MOV D, M
MOV E, M
MOV H, M
MOV L, M

Код операции

7E
46
4E
56
5E
66
6E

Команда MOV M, R.
писывает в ячейку па-
хранится в регистро-
жимое регистра R с

ПРИМЕР 2.3.
(Move register) загру-
мером (DDD) содер-
номером (SSS).

Содержимое R_2

Схема команды:

$[R_2] \rightarrow [R_1]$

Код операции:

01DDSSS

Эта команда однобайтная, выполняется за один цикл.

Некоторые варианты этой команды:

MOV M, A
MOV M, B
MOV M, C
MOV M, D
MOV M, E
MOV M, H
MOV M, L

Код операции

77
70
71
72
73
74
75

Команда MOV R_1, R_2 .
жает регистр R_1 с но-
жимым регистра R_2 с

не меняется.

MOV B, A
MOV A, B
MOV C, A
MOV A, C
MOV C, B
MOV B, C

Код операции

47
78
4F
79
48
41

Изучить команды LDA $\langle B_2 \rangle \langle B_3 \rangle$, STA $\langle B_2 \rangle \langle B_3 \rangle$, LXI $\langle B_2 \rangle, \langle B_3 \rangle$, LHLD $\langle B_2 \rangle, \langle B_3 \rangle$, SHLD $\langle B_2 \rangle, \langle B_3 \rangle$ самостоя-
тельно.

Арифметические команды

ПРИМЕР 2.4. Команда ADD R складывает содержимое аккумулятора с содержимым регистра R (с номером SSS) и засылает результат в аккумулятор.

Схема команды:

$[Ак] + [R] \rightarrow [Ак]$

Код операции:

10000SSS

Эта команда однобайтная и выполняется за один цикл.

Варианты команды:

ADD A
ADD B
ADD C
ADD D
ADD E
ADD H
ADD L

Код операции

87
80
81
82
83
84
85

ПРИМЕР 2.5. Команда ADD M складывает содержимое
аккумулятора с содержимым ячейки памяти, адрес которой находится
в регистрах H, L. Результат засылается в аккумулятор.

Схема команды:

$(A_k) + [H, L] \rightarrow [A_k]$

Код операции:

10000110 (86)

Эта команда однобайтная и выполняется за два цикла.

ПРИМЕР 2.6. Команда ADC R складывает содержимое аккумулятора с содержимым регистра R с номером (SSS) и с содержимым разряда "C" регистра признаков. Результат заносится в аккумулятор.

Схема команды:

$(A_k) + (R) + (C) \rightarrow (A_k)$

Код операции:

10001SSS

Эта команда однобайтная, выполняется за один цикл.

Варианты команды:

	Код операции
ADC A	8F
ADC B	88
ADC C	89
ADC D	8A
ADC E	8B
ADC H	8C
ADC L	8D

ПРИМЕР 2.7. Команда ADC M складывает содержимое аккумулятора с содержимым ячейки памяти, адрес которой записан в регистрах HL и содержимым разряда "C" регистра признаков. Результат заносится в аккумулятор.

Схема команды:

$(A_k) + [H, L] + (C) \rightarrow (A_k)$

Код операции:

10001110 (8E)

Эта команда однобайтная, выполняется за один цикл.

Команды вычитания SUB R и SBB R изучить самостоятельно.

ПРИМЕР 2.8. Команда INR R увеличивает содержимое регистра R на 1 и результат засылает в регистр R.

Схема команды:

$(R) + 1 \rightarrow (R)$

Код операции:

00DDD100

Эта команда однобайтная, выполняется за один цикл.

Изучить команду DCR R самостоятельно.

ПРИМЕР 2.9. Команда INX RP увеличивает содержимое пары регистров на 1, и результат записывает в пару регистров RP.

Схема команды:

$(RP) + 1 \rightarrow (RP)$

Код операции:

00RP0011

Эта команда однобайтная, выполняется за один цикл.

ПРИМЕР 2.10. Команда DAD RP складывает содержимое регистровой пары RP с содержимым регистровой пары HL и результат засылает в H,L.

Схема команды:

$(RP) + (HL) \rightarrow (HL)$

Код операции:

00RP001

Эта команда однобайтная, выполняется за три цикла.

Команду INR M изучить самостоятельно.

Логические команды

ПРИМЕР 2.11. Команда ANA R есть логическая операция "И" (конъюнкция - соединение) над содержимым регистров и аккумулятора, результат заносится в аккумулятор (перенос сбрасывается).

Схема команды:

$(A_k) \wedge (R) \rightarrow (A_k)$

Код операции:

10100SSS

Эта команда однобайтная, выполняется за один цикл.

ПРИМЕР 2.12. Команда ANI <B₂> есть логическая операция "И" над содержимым аккумулятора и второго байта команды, результат заносится в аккумулятор (перенос сбрасывается).

Схема команды:

$$(A_k) \wedge \langle B_2 \rangle \rightarrow (A_k)$$

Код операции:

11100110

Эта команда двухбайтная, выполняется за два цикла.

ПРИМЕР 2.13. Команда ORA R есть логическая операция "ИЛИ" (дизъюнкция - разделение) над содержимым аккумулятора и регистра R, результат заносится в аккумулятор (перенос сбрасывается).

Схема команды:

$$(A_k) \vee (R) \rightarrow (A_k)$$

Код операции:

10110SSS

Эта команда однобайтная, выполняется за один цикл.

ПРИМЕР 2.14. Команда CMP R сравнивает содержимое аккумулятора и регистра R. Содержимое аккумулятора не изменяется. Разряды признаков устанавливаются по результату вычитания: $(A_k) - (R)$

если $(A_k) = (R)$, то $z = 1$

если $(A_k) < (R)$, то $c = 1$, иначе $\bar{z} \cdot \bar{c} = 1$

Код операции:

10111

Эта команда однобайтная, выполняется за один цикл.

ПРИМЕР 2.15. Команда RLC циклически сдвигает влево на 1 бит содержимое аккумулятора.

Схема команды:

$$(A_n) \rightarrow (A_{n+1}); (A_7) \rightarrow (A_0); (A_7) \rightarrow (C).$$

Код операции:

00000111 (07)

Эта команда однобайтная, выполняется за один цикл.

ПРИМЕР 2.16. Команда RRC циклически сдвигает вправо содержимое аккумулятора.

Схема команды:

$$(A_{n+1}) \rightarrow (A_n); (A_0) \rightarrow (A_7); (A_0) \rightarrow (C).$$

Код операции:

00001111 (08)

Эта команда однобайтная, выполняется за один цикл.

ПРИМЕР 2.17. Команда RAL сдвигает влево на 1 бит содержимое аккумулятора через разряд переноса.

Схема команды:

$$(A_n) \rightarrow (A_{n+1}); (A_7) \rightarrow (C); (C) \rightarrow (A_0).$$

Эта команда однобайтная, выполняется за один цикл.

ПРИМЕР 2.18. Команда CMA берет обратный код содержимого аккумулятора. Содержимое регистра признаков не меняется.

Схема команды:

$$(\bar{A}) \rightarrow (A)$$

Код операции:

00101111 (28)

Эта команда однобайтная, выполняется за один цикл.

Самостоятельно изучить команды XRA R, RAR.

Контрольные вопросы

1. Назовите группы команд микропроцессора K580BM80.
2. Опишите функции команд.
3. Запишите команду непосредственной загрузки регистровой пары.
4. Назовите и охарактеризуйте известные вам команды арифметических операций.
5. Перечислите команды пересылки информации.
6. Что такое стек и для чего он предназначен?
7. Опишите механизм обращения к подпрограмме.
8. Опишите механизм возврата из подпрограммы.
9. Перечислите и охарактеризуйте логические команды.
10. В чем смысл команд сдвига?

Занятие 3. Система команд (продолжение)

Команды передачи управления

Команды ветвления (переходов).

ПРИМЕР 3.1. Команда JMP <B₂><B₃> передаёт управление команде, адрес которой определяется содержимым третьего и второго байтов команды.

Схема команды:

<B₂><B₃> → (CT1) Код операции:

110000 (C3)

Эта команда трёхбайтная, выполняется за три цикла.

ПРИМЕР 3.2. Команда JNZ <B₂><B₃> передаёт управление команде, адрес которой задан в третьем и втором байтах команды по признаку "не ноль", в противном - случае команде, следующей за этой командой.

Схема команды:

Если (z)=0, то <B₃><B₂> → (CT1)

иначе (CT1)+3 → (CT1)

Код операции:

11000010 (C2)

CCC

Эта команда трёхбайтная, выполняется за три цикла. CCC определяет признак перехода и от его значения возможны следующие команды:

<u>Команды пер- грамме.</u>	JNZ - если не ноль,	Z=0, CCC → 000;	<u>перехода к подпро-</u>
	JZ - если ноль,	Z=1, CCC → 001;	
ПРИМЕР 3.3.	JNC - если не перенос,	C=0, CCC → 010;	Команда CALL
	JC - если перенос,	C=1, CCC → 011;	
<B ₂ ><B ₃ > заносит	JPO - если нечётно,	P=0, CCC → 100;	содержимое
	JPE - если чётно,	P=1, CCC → 101;	
счётчика ко- ресу указателя стека	JP - если плюс,	S=0, CCC → 110;	манд в стек по ад- (SP), содержимое
	JM - если минус,	S=1, CCC → 111.	
регистра SP			(стека)
уменьшает на 2 и			выполня-
ет переход к команде, записанной по адресу, указанному в третьем и втором байтах команды.			

Схема команды:

(CT1) → [(SP-1)], [(SP-2)]

(SP)-2 → (SP); <B₃><B₂> → (CT1)

Код операции:

11001101 (CD)

Эта команда трёхбайтная, выполняется за пять циклов.

ПРИМЕР 3.4. Команда CNZ <B₂><B₃> осуществляет обращение к подпрограмме по признаку "не ноль".

Схема команды:

Если (z)=0, то (CT1) → [(SP-1)], [(SP-2)]

(SP)-2 → (SP); <B₃><B₂> → (CT1)

иначе (CT1)+3 → (CT1)

Код операции:

11000100 (C4)

CCC

CCC – определяет признак перехода и от его значения возможны команды:

CNZ - если не ноль,	Z=0, CCC → 000;
CZ - если ноль,	Z=1, CCC → 001;
CNC - если не перенос,	C=0, CCC → 010;
CC - если перенос,	C=1, CCC → 011;
CPO - если нечётно,	P=0, CCC → 100;
CPE - если чётно,	P=1, CCC → 101;
CP - если плюс,	S=0, CCC → 110;
CM - если минус,	S=1, CCC → 111.

Все эти команды трёхбайтные и выполняются за три-пять циклов.

Команды возврата из подпрограммы

ПРИМЕР 3.5. Команда RET осуществляет возврат к основной программе, адрес которой записан в верхней паре ячеек стека и увеличение указателя стека на два.

Схема команды:

[SP], [(SP)+1] → (CT1);

$(SP)+2 \rightarrow (SP)$.

Код операции:

11001001 (C9)

Эта команда трёхбайтная, выполняется за три цикла.

ПРИМЕР 3.6. Команда RNZ осуществляет возврат из подпрограмм по признаку "не ноль".

Схема команды:

Если $(z)=0$, то $[(SP)], [(SP)+1] \rightarrow (CTI)$

$(SP)+2 \rightarrow (SP)$

иначе $(CTI)+1 \rightarrow (CTI)$

Код операции:

11000 000
CCC

CCC – определяет признак перехода и от его значения возможны команды:

RNZ - если не ноль,	Z=0,	CCC \rightarrow 000;
RZ - если ноль,	Z=1,	CCC \rightarrow 001;
RNC - если не перенос,	C=0,	CCC \rightarrow 010;
RC - если перенос,	C=1,	CCC \rightarrow 011;
RPO - если нечётно,	P=0,	CCC \rightarrow 100;
RPE - если чётно,	P=1,	CCC \rightarrow 101;
RP - если плюс,	S=0,	CCC \rightarrow 110;
RM - если минус,	S=1,	CCC \rightarrow 111.

Все эти команды трёхбайтные и выполняются за один-три цикла.

Команды управления

К этой группе команд относят помимо отмеченных выше и команды обращения к стеку.

ПРИМЕР 3.7. Команда PUSH RP содержимое регистровой пары заносят побайтно в стек.

Схема команды:

$[R] \rightarrow [(SP)-1], [P] \rightarrow [(SP)-2],$

где R - старший регистр пары (четный номер);

P - младший регистр пары (нечетный номер).

Код операции:

11RP0101

Эта команда однобайтная, выполняется за три цикла.

ПРИМЕР 3.8. Команда PUSH PSW помещает в стек содержимое аккумулятора и регистра признаков.

Схема команды:

$[Ак] \rightarrow [(SP)-1]; (RGF) \rightarrow [(SP)-2]; (SP)-2 \rightarrow (SP)$

Код операции:

11110101 (F5)

Эта команда однобайтная, выполняется за три цикла.

ПРИМЕР 3.9. Команда POP RP побайтно загружает в пару регистров два верхних элемента стека.

Схема команды:

$[SP] \rightarrow (P); [(SP)+1] \rightarrow R; (SP)+2 \rightarrow (SP)$

Код операции:

11RP0001

Эта команда однобайтная, выполняется за три цикла.

ПРИМЕР 3.10. Команда POP PSW содержимое двух верхних ячеек стека заносит в аккумулятор и регистр признаков.

Схема команды:

$[SP] \rightarrow (RGF); [(SP)+1] \rightarrow (Ак); (SP)+2 \rightarrow (SP)$

Код операции:

11110001 (F1)

Эта команда однобайтная, выполняется за три цикла.

ПРИМЕР 3.11. Команда SPHL загружает в регистр указатель - стека содержимое регистровой пары HL.

Схема команды:

$(HL) \rightarrow [SP]$

Код операции:

11111001 (F9)

3. ИСПОЛЬЗУЕМОЕ ОБОРУДОВАНИЕ.

Учебная микроЭВМ типа УМПК 80.

4. ЗАДАНИЕ НА РАБОТУ.

Ознакомиться с группами команд микропроцессора:

1. Команды пересылки информации;
2. Команды арифметических операций;
3. Команды логических операций и сдвигов;
4. Команды передачи управления;
5. Команды ввода/вывода;
6. Команды управления микропроцессором;

5. Контрольные вопросы

1. Перечислите команды переходов.
2. По каким условиям осуществляются переходы?
3. Опишите схему команды CALL <B₂><B₃>.
4. Какие команды обращения к подпрограмме по условию вы знаете?
5. Для чего предназначена команда, RET и какие действия она производит?
6. Назовите команды возврата из подпрограммы по условию.
7. Каков формат команд передачи управления?
8. Назовите команды работы со стеком.

6. СОДЕРЖАНИЕ ОТЧЕТА

Отчёт должен содержать описание одной команды для каждой группы команд

7. БИБЛИОГРАФИЧЕСКИЙ СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.

3. Микропроцессоры: в 3-х кн. Кн. 3. Средства отладки, лабораторный практикум, задачник: Учеб. для втузов/Н.В.Воробьёв, В.Л.Горбунов, А.В. Горячев и др.; Под ред. Л.Н. Преснухина. - К.: Высш.шк., 1936. - 351 с.
Учебная микроЭВМ УМПК-80. ПБА2.390.039.ПС. Паспорт.

5. Лабораторная работа №3.

Разработка программ для микро-ЭВМ УМПК-80 с использованием подпрограмм и стека для управления звуковым сигналом.

1. ЦЕЛЬ И ЗАДАЧИ РАБОТЫ.

Целью работы является дальнейшее освоение программирования однокристальных микропроцессоров с фиксированным набором команд.

Задачей работы является изучение особенностей записи и обращения к подпрограммам, а также методов использования стека при создании программ.

2. ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ.

Учитывая ограниченные возможности памяти при разработке, нужно стараться сделать их как можно короче. С этой целью часть программы, которая неоднократно повторяется, или программа, которая часто используется, могут быть оформлены 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, называют входными параметрами. Результаты работы программы, передаваемые по окончании ее работы в основную программу, называются выходными параметрами.

Для вызова подпрограммы и возврата из нее используются команда CALL <A₂><A₁> и RET.

Команда CALL <A₂><A₁> загружает в программный счетчик МП БИС содержимое байтов <A₂><A₁>, записанных вслед за кодом команды CALL (СД). Содержимое байта <A₂> записывается в младший байт программного счетчика (PCL), а содержимое <A₁> - в старший байт (PCH), при этом МП БИС автоматически сохраняет в стеке адрес основной программы, к которому она будет обращаться после выполнения подпрограммы.

Стек - специально организованная область ОЗУ, задействованная в микроЭВМ для временного хранения данных или адресов. Число, записанное в стек последним, извлекается первым.

Команда RET(C9) помещает в программный счетчик последнее записанное на данный момент в стеке число. После этого выполнение программы будет осуществляться с этого адреса. Любая подпрограмма должна кончаться командой RET. Автоматическое сохранение и восстановление адреса основной программы при выполнении подпрограмм позволяет сделать подпрограммы вложенными, т.е. осуществлять вызов одной подпрограммы из другой. Уровень вложенности определяется 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. Все команды условного вызова трехбайтные. Команды вызова подпрограмм и возврата из них используют стек и указатель стека МП БИС (SP) для адресации обращения 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 обозначенные регистры МП БИС). Эти команды однобайтные, и в них содержится указание пары регистров МП БИС.

При записи в стек содержимого пары регистров или программного счетчика по адресу SP-1 записывается содержимое старшего регистра из указанной пары или стерший байт программного счетчика (PCH), а по адресу SP-2 в стек записывается содержимое младшего регистра из указанной пары или младший байт программного счетчика (PCL).

При записи из стека данных в пару регистров или программный счетчик в младший регистр пары или в PCL записывается число из адреса, записанного в указателе стека SP, а в старший регистр пары или PCH - число, записанное по адресу SP+1.

При разработке программ необходимо первоначально назначить область стека, записывая в SP адрес с помощью команд LXI SP <A₂> <A₁> или SHPL.

Все операции со стеком должны быть сбалансированы, т.е. каждая подпрограмма должна иметь равное количество команд PUSH <R> и POP <R>, и заканчиваться командой RET.

Как правило, перед каждым обращением к подпрограмме сохраняют в стеке содержимое всех задействованных в вызывающей программе регистров с помощью команды PUSH <R>. В конце подпрограммы восстановления осуществляется с помощью команд POP <R> и в обратной последовательности по отношению к их записи в стек.

Алгоритм работы подпрограммы временной задержки представлен на рис. 1а. Общее время задержки вычисляется по формуле:

$$T = t_1 + (t_2 + t_3 + t_4) \cdot N + t_5 \quad (1)$$

где: N - число, первоначально записанное в счетчике.

В качестве счетчика выбран регистр "B", в который из регистра "C" записывается число " N ". Команда HOP нужна для увеличения времени выполнения цикла.

Подпрограмма 1 временной задержки, записана в соответствии с представленным на рис. 1а алгоритмом.

Подпрограмма 1.

Адрес	Машинный код	Метка	Мнемокод	Комментарий
0900	41		MOV B, C	$C \rightarrow B$
0901	00	DLY	NOP	Нет операции
0902	05		DCR B	$B-1 \rightarrow B$
0903	C2 0109		JNZ DLY	Переход к DLY если $\langle B \rangle \neq 0$
0906	C9		RET	Возврат

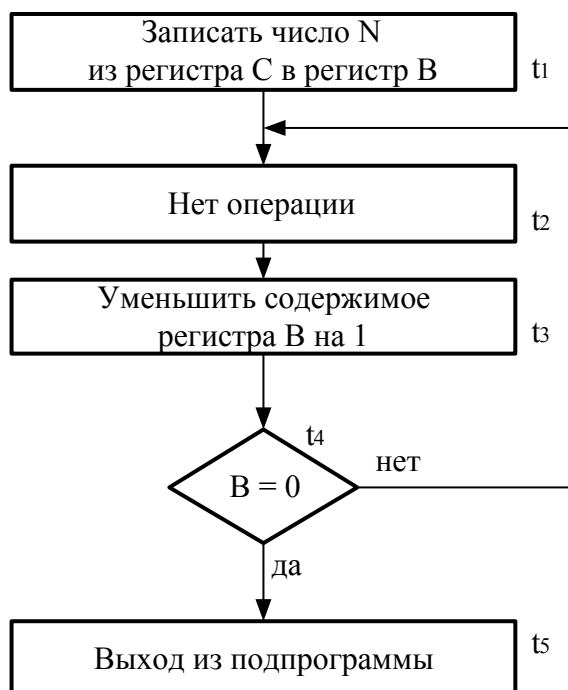


Рис. 1. Алгоритм простой программы временной задержки

Рассмотрим пример использования подпрограммы временной задержки при организации звуковых сигналов в микроЭВМ. В микроЭВМ звуковые сигналы могут формироваться простейшей схемой рис. 2, на вход которой со звуковой частотой записывается по очереди "0" и "1". Пусть устройство звуковых сигналов имеет адрес 80. Схема алгоритма работы подпрограммы генерации звуковых колебаний (подпрограмма 2) представлена на рис.3.

Подпрограмма 2.

Адрес	Машинный код	Метка	Мнемокод	Комментарий
0A00	AF	BPP	XRA A	0 → Ак
0A01	D380		OUT BP	0 → вых. устр.
0A03	CD 0009		CALL DLY	Вызов п/п задержки
0A06	2F		CMA	1 → Ак
0A07	D380		OUT BP	1 → вых. устр.
0A09	CD 0009		CALL DLY	Вызов п/п задержки
0A0C	C9		RET	Возврат

Программа 3 (MAIN) представляет собой программу генерации сигналов с частотой, задаваемой с входного регистра.

Программа 3.

Адрес	Машинный код	Метка	Мнемокод	Комментарий
0800	ДВ20	MAIN	IN 20	Входной регистр → Ак
0802	4F		MOV C, A	Ак → С
0803	CD 000A		CALL BPP	Вызов п/п BPP
0806	C3 0008		JNP MAIN	Продолжить

6. В чём отличие выполнения микроЭВМ команд CALL и RET?
 7. В какой последовательности сохраняется и извлекается содержимое регистров МП БИС в подпрограммах?

6. СОДЕРЖАНИЕ ОТЧЕТА

Отчёт должен содержать

1. Схему формирования звуковых сигналов в микроЭВМ
2. Алгоритмы программ
3. Тексты отлаженных программ

7. БИБЛИОГРАФИЧЕСКИЙ СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Микропроцессоры: В 3-х кн. Кн.3. Средства отладки, лабораторный практикум, задачник: Учеб. для втузов/Н.В.Воробьёв, В.Л.Горбунов, Л.В.Горячев и др.; Под ред. Л.Н.Преснухина,- М.: Высш. шк., 1986.-351 с.
2. Учебная микроЭВМ УМПК-80. ПБА2.390.039.ПС. Паспорт

6. Лабораторная работа №4.

Изучение принципов подключения светодиодного дисплея к микро-ЭВМ

1. ЦЕЛИ ЗАДАЧИ РАБОТЫ.

Целью работы является изучение программно-аппаратурных методов подключения дисплея к микроЭВМ. Задачей работы является разработка программ управления светодиодным дисплеем.

2. ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ.

Подключение дисплея к микроЭВМ рассмотрим методы подключения дисплея, состоящего из шести ячеек (светодиодных индикаторов типа АЛС 324). Каждый индикатор, рис.5а, имеет семь светодиодов для отображения семи сегментов цифр, и восьмой - для отображения десятичной точки, рис.5 б. Индикатор отражает цифры от 0 до 9, а также некоторые буквы латинского алфавита.

Пронумеруем ячейки дисплея так, как показано на рис.5в.

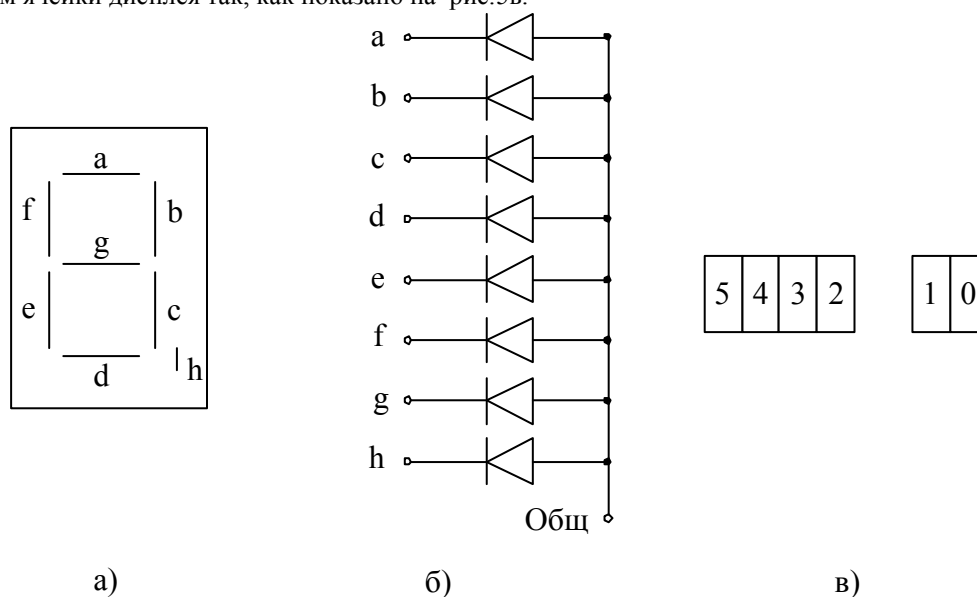


Рис. 5 Внешний вид а), схема семисегментной светодиодной матрицы б), и нумерация ячеек дисплея в).

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: P_7C_7 (адрес 38_{16}) для записи семисегментного кода и P_7C_8 (адрес 28_{16}) для записи номера индикатора рис.6.

Одинаковые сегменты каждой ячейки индикатора связаны общей шиной, которая соединена с одним из транзисторных ключей VT_1-VT_8 на выходе регистра P_7C_7 . Общие аноды индикатора подключены к одному из транзисторных ключей VT_9-VT_{14} на выходе регистра сканирования P_7C_8 .

Включение индикатора и его сегментов при записи единицы в соответствующие разряды выглядит так:
 для регистра P_7C_7 (адрес 38_{16}): Номер разряда регистра сегментов ... 0 1 2 3 4 5 6 7

Включенный сегмент..... а в с d e f g h

для регистра P_7C_8 (адрес 28_{16}):

Номер разряда регистра сканирования...0 1 2 3 4 5 6 7

Включается ячейка дисплея..... 0 1 2 3 4 5 - -

Сигналы заведены на регистры PrCr и PrCk (K589IP12 так, что при поступлении на вход BM1 сигнала выборки от дешифратора адреса данные с шины данных (ШД), подключенной к входам D₀-D₇ записываются в регистр и появляются на его входах Q₀-Q₇. Таким образом, например, при записи в регистр сегментов числа 00000110 отпираются транзисторные ключи VT₂ и VT₃, а при записи в регистр сканирования числа 00100000 отпирается транзисторный ключ VT₁₄ и ток проходит по цепи: +5В - VT₁₄ - HG1 - b и c - VT₂ и VT₃ - земля, при этом на левом индикаторе высветится 1. Если теперь с помощью программы высвечивать по очереди все ячейки, записывая их код в регистр сканирования и включая необходимую цифру с помощью регистра сегментов, то при достаточно высокой частоте переключений можно получать устойчивое изображение выводимой информации на дисплее.

0800	3E 20		MVI A, 20	00100000 → Ак
0802	D3 28		OUT SCAN	Включить цифру 5
0804	DB 20	CNT	IN 20	Считать данные С входного устройства
0806	D3 38		OUT DSP	Записать их в регистр сегментов
0808	C3 0408		JMP CNT	Продолжить

При мультиплексном режиме работы вывод информации на каждый индикатор дисплея производится микроЭВМ последовательно.

3. МСПОЛЬЗУЕМОЕ ОБОРУДОВАНИЕ.

Учебная микроЭВМ типа УМП-80 и макет клавиатуры.

4. ЗАДАНИЕ НА РАБОТУ.

1. Изучить схемы подключения дисплея к микроЭВМ.
2. Изучить программное обеспечение, поддерживающее работу микроЭВМ с дисплеем.
3. Исследуйте программу 6 при организации вывода различных цифр и букв.
4. Разработайте программу мультиплексного включения ячеек светодиодного дисплея

5. КОНТРОЛЬНЫЕ ВОПРОСЫ.

1. Как организуется вывод информации на дисплей?
2. Как надо видоизменить программу 6, чтобы по очереди загорелись сегменты а, в, с, d, e, f, g, h?

6. СОДЕРЖАНИЕ ОТЧЕТА

Отчёт должен содержать

1. Схему подключения дисплея к микроЭВМ.
2. Алгоритмы программ
3. Тексты отлаженных программ

7. БИБЛИОГРАФИЧЕСКИЙ СПИСОК РЕКОМЕНДУЕМЫХ ИСТОЧНИКОВ.

1. Микропроцессоры: в 3-х кн. Кн. 3. Средства отладки лабораторный практикум и задачник: Учеб. для втузов/Н.В. Воробьев, В.Л. Горбунов, А.В. Горячев и др.; Под редакцией Л.Н. Преснухина.-М.: Вьющая шк., 1986. - 351 с.
2. МикроЭВМ: в 3 кн. Практич. пособие/Под редакцией Л.Н.Преснухина, Кн.7. Учебные стенды/1).И.Волков, В.Л.Горбунов, Д.И.Панфилов, С.Г. Шальгин - М.: Высш. шк., 1988.-224 с,
3. Учебная микроЭВМ УМПК-80. ПБА2.390.039.ПС. Паспорт.

7. Лабораторная работа №5.

Изучение принципов подключения клавиатуры к микро-ЭВМ

1. ЦЕЛИ ЗАДАЧИ РАБОТЫ.

Целью работы является изучение программно-аппаратурных методов подключения клавиатуры к микроЭВМ. Задачей работы является разработка программ управления сканированием клавиатуры.

2. ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ.

Подключение клавиатуры к микроЭВМ. Клавиатура является одним из широко распространённых устройств ввода данных и управляющих воздействий в микроЭВМ. С помощью клавиатуры можно вводить программу в ОЗУ, инициализировать различные режимы работы микроЭВМ.

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 клавиш на клавиатуре; 2) нахождение номера нажатой клавиши; 3) осуществление передачи управления на соответствующую подпрограмму. Первые две задачи являются специфическими при организации ввода информации с клавиатуры. Последняя, как правило, решается программными методами.

Наиболее удобно реализовать клавиатуру в виде матрицы размером $n \times m$, где n и m - соответственно число строк и столбцов. Для иллюстрации методы сопряжения клавиатуры с микроЭВМ рассмотрим случай, когда $n = m = 3$ рис. 7 (где: Сх ДША-схема дешифрации адреса, Увв - устройство ввода, Увыв - устройство вывода).

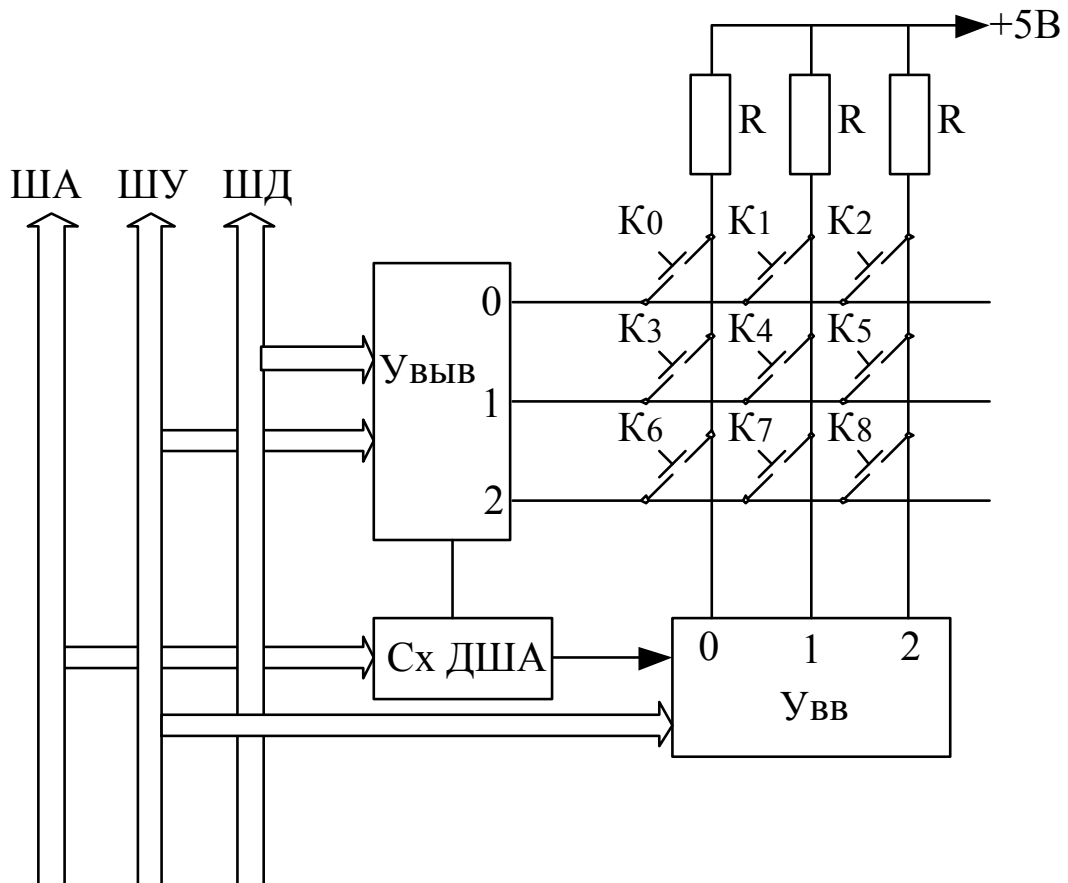


Рис.7 Схема подключения клавиатуры к микроЭВМ.

Строки клавиатуры подключены к трем младшим разрядам устройства вывода (символьное обозначение адреса KBDOT), а столбцы подключены к трем младшим разрядам устройства ввода (символьное обозначение адреса KBDIN).

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

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

Программа 7, написанная в соответствии с приведенным алгоритмом, позволяет определить факт нажатия на одну из клавиш, но не указывает номер нажатой клавиши.

Программа 7.

Адрес	Машинный код	Метка	Мнемокод	Комментарий
0800	3E F8	WAIT K	MVI, A, 11111000	Записать 0 в младшие разряды <i>Ак</i>
0802	D3 KBDOT		OUT KBDOT	Записать 0 в выходное устройство
0804	DB KBDIN		IN KBDIN	Получить число из выходного устройства
0806	E6 07		ANI 00000111	Очистить старшие 5 разрядов аккумулятора
0808	FE 07		CPI 00000111	Есть ли в младших 3-х разрядах <i>Ак</i> ноль?
080A	CA 0008		JZ WAIT K	Если нет, то идти на повторение
080D	C3 0D08	DONE	JMP DONE	Конец

Определить номер нажатой клавиши можно с помощью алгоритма, приведенного на рис.8.

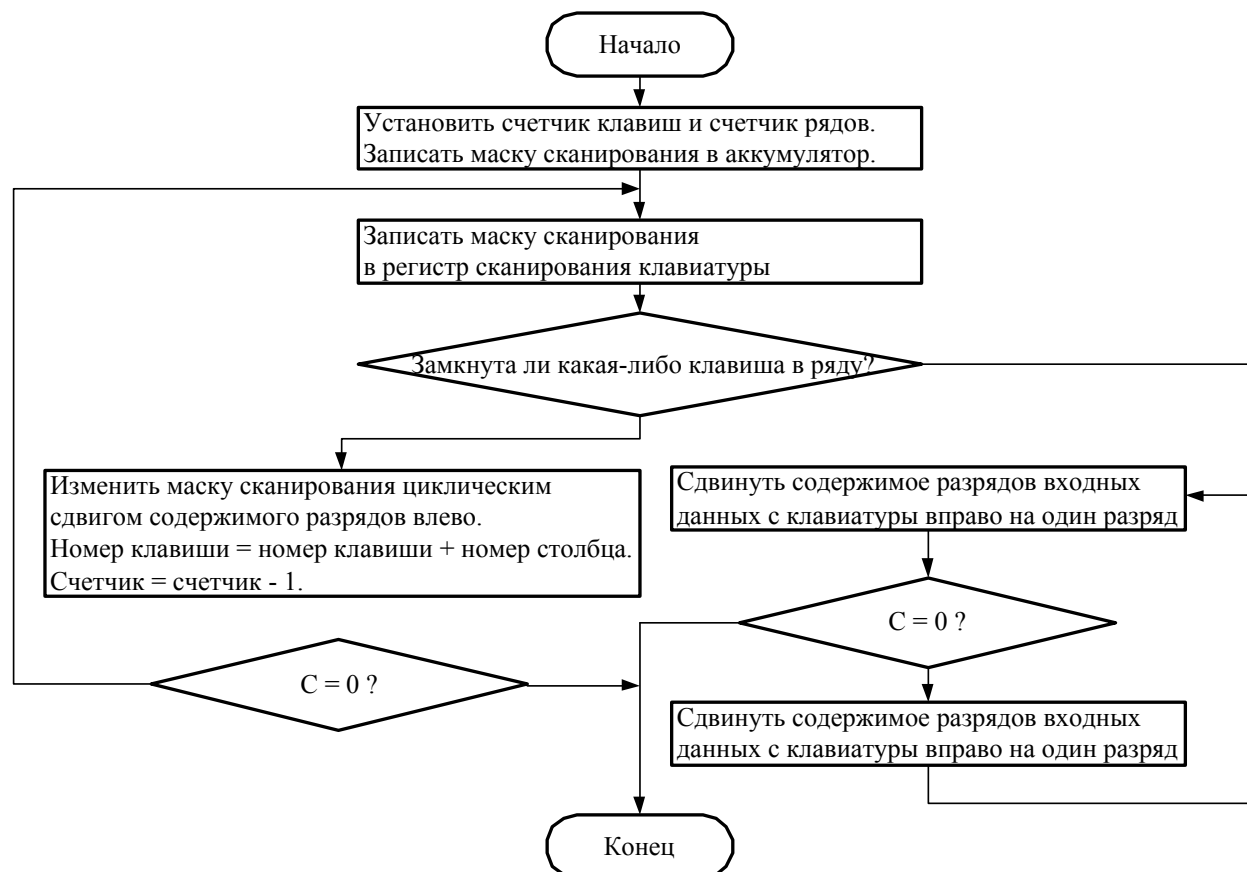


Рис. 8. Схема алгоритма определения нажатой клавиши при сканировании клавиатуры микроЭВМ

Алгоритм основывается на последовательности записи нуля в каждую из строк матрицы клавиатуры. При наличии нуля в каждом ряду микроЭВМ определяет факт нажатия на клавиши, находящиеся в столбцах клавиатуры, принадлежащих анализируемому ряду. Если какая-то клавиша нажата, то определяется ее номер по номеру разряда, в котором записан ноль. Программа 8 реализует описанный выше алгоритм. Номер нажатой клавиши определяется по номеру строки клавиатуры, в которой обнаружена нажатая клавиша, и номеру разряда входного устройства, в котором записан ноль.

Программа 8

Адрес	Машинный код	Метка	Мнемокод	Комментарий
0800	0600		MVI B, 00	Обнулить счётчики клавиши
0802	0EFE		MVI C, FE	Установка маски сканирования строк
0804	1603		MVI D, 03	Установить счётчик строк
0806	79	FROW	MOV A, C	Записать маску сканирования в аккумулятор
0807	D3 KBDOT		OUT KBDOT	Записать маску сканирования в выходное устройство
0809	07		RLC	Изменить маску сканирования
080A	4F		MOV C, A	Сохранить маску в регистре C
080B	DB KBDIN		IN KBDIN	Получить число из входного устройства
080D	E6 07		ANI 00000111	Маскировать 5 старших разрядов входного устройства
080F	C2 1F 08		JNZ FCOL	Если да, то идти на
0814	78		MOV A, B	Изменить содержимое счётчика
0815	C6 03		ADI 3	номера клавиши
0817	47		MOV B, A	
0818	15		DCR D	Уменьшить содержимое счётчика строк
0819	C2 06 08		JNZ FROW	Если не последняя строка, то повторить
0810	C3 27 08		JMP DONE	Идти на окончание
081F	1F	FCOL	RAR	
0820	D2 27 08		JNC DONE	Определение номера разряда в ко-

0823	04		INR B	тором записан 0 определение но-
0824	C3 1F 08		JMP FCOL	мера нажатой клавиши
0827	C3 27 08	DONE	JMP DONE	Конец

3. МСПОЛЬЗУЕМОЕ ОБОРУДОВАНИЕ.

Учебная микроЭВМ типа УМП-80 и макет клавиатуры.

4. ЗАДАНИЕ НА РАБОТУ.

1. Изучить схемы подключения клавиатуры к микроЭВМ.
2. Изучить программное обеспечение, поддерживающее работу микроЭВМ с клавиатурой.
3. Исследуйте программу 7 обслуживания клавиатуры, предварительно подключив к микроЭВМ макет клавиатуры.
4. Дополните программу 8 так, чтобы номер нажатой клавиши отображался на одной из ячеек дисплея.

5. КОНТРОЛЬНЫЕ ВОПРОСЫ.

1. Какая, из клавиш будет определяться нажатой с помощью программы 8, если на клавиатуре будут одновременно нажаты: а) клавиши 3 и 5; б) клавиши 2 и 8?
2. Видоизмените программу 8 так, чтобы сканирование клавиатуры происходило постоянно.

6. СОДЕРЖАНИЕ ОТЧЕТА

Отчёт должен содержать

1. Схему подключения клавиатуры к микроЭВМ.
2. Алгоритмы программ
3. Тексты отлаженных программ

7. БИБЛИОГРАФИЧЕСКИЙ СПИСОК РЕКОМЕНДУЕМЫХ ИСТОЧНИКОВ.

1. Микропроцессоры: в 3-х кн. Кн. 3. Средства отладки лабораторный практикум и задачник: Учеб. для втузов/Н.В. Воробьев, В.Л. Горбунов, А.В. Горячев и др.; Под редакцией Л.Н. Преснухина.-М.: Вьющая шк., 1986. - 351 с.
2. МикроЭВМ: в 3 кн. Практич. пособие/Под редакцией Л.Н.Преснухина, Кн.7. Учебные стенды/1).И.Волков, В.Л.Горбунов, Д.И.Панфилов, С.Г. Шальгин - М.: Высш. шк., 1988.-224 с,
3. Учебная микроЭВМ УМПК-80. ПБА2.390.039.ПС. Паспорт.

Лабораторная работа №6

Разработка арифметических программ на ассемблере МП БИС серии КР580.

1. ЦЕЛИ И ЗАДАЧИ РАБОТЫ.

Целью работы является усвоение навыков программирования вычислительных задач на ассемблере МП БИС серии КР580.

Задачей работы является разработка подпрограмм, реализующих арифметические действия и вычисления некоторых функций.

2. ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ.

Из двух способов представления чисел с фиксированной или с плавающей точкой первый получил наибольшее распространение при программировании микроЭВМ на МП БИС серии КР580. Это связано с отсутствием специальных команд, позволяющих работать МП БИС с числами с плавающей точкой. В свою очередь, двоичное восьмиразрядное число с фиксированной точкой можно представить как двоичное число со знаком, имеющее значение от -128_{10} до $+127_{10}$. При этом отрицательные числа представляются в дополнительном коде, а старший седьмой разряд числа используется как знаковый. Такое представление чисел не позволяет выполнять арифметические операции с использованием переноса при сложении и заёма при вычитании.

Число с фиксированной точкой можно представить также двоичными числами без знака, имеющими значения от 0 до 255_{10} .

Для МП БИС КР580ВМ80 можно представлять также числа в виде двоично-десятичного числа – BINARI – CODED DECIMAL (BCD) при котором каждый полубайт кодирует десятичную цифру. Такое представление позволяет закодировать в 1 байте числа от 0 до 99_{10} .

Рассмотрим программу 4 сложения массива однобайтных чисел с получением двухбайтного результата.

Программа 4.

Адрес	Машинный код	Метка	Мнемокод	Комментарий
-------	--------------	-------	----------	-------------

0800	21 0COB	MAIN	LXI H, 0B00	Адрес 1-го го \rightarrow HL
0803	06 05		MVI B, 05	$N \rightarrow B$
0805	CD 0908		CALL ADDDB	Вызов п/п сложения
0808	CF		RST1	Прервать программу
0809	AF	ADDB	XRA A	$0 \rightarrow Ak$.
080A	4F		MOV C, A	Очистить счётчик переносов
080B	86	CNT	ADD M	$Ак + \langle M \rangle \rightarrow Ак$
080C	D2 1008		JNC TRM	Если нет переноса, то идти на TRM
080F	0C		INR C	$C+1 \rightarrow C$
0810	23	TRM	INX H	$HL+1 \rightarrow HL$
0811	05		DCR B	$B-1 \rightarrow B$
0812	D2 0B08		JNC CNT	Если не все слагаемые, то идти на CNT
0815	C9		RET	Возврат.

Слагаемые должны быть расположены в последовательных адресах памяти. Входными параметрами подпрограммы ADDDB является адрес первого слагаемого, записанный в регистрах H и L, и число слагаемых, записанные в регистре "B". Выходными параметрами программы MAIN является сумма, старший байт которой записан в регистре "C", а младший в аккумуляторе "Ак". Алгоритм сложения состоит в том, что после прибавления каждого элемента массива определяется переполнение аккумулятора (разряд C-1), и если это имеет место, то содержимое регистра "C" увеличивается на "1". Таким образом, за счет суммирования единиц переноса получается старший байт, рис. 4а.

МикроЭВМ может проводить арифметические операции с числами двойной или большей длиной машинного слова. Так как МП БИС имеет 8-разрядное АПУ, то операции с такими числами должны проводиться по байтам, начиная с младших байтов, а далее с учетом переноса (при сложении) или заёма (при вычитании) единицы в старший разряд. Такие команды, учитывающие значения разряда C регистра признаков, присутствуют в системе команд МП БИС КР580ВМ80: ADC и SBB. Рассмотрим программу умножения чисел. Существует несколько алгоритмов, реализующих эти операцию. При первом алгоритме умножение заменяют многократным сложением, например: $14 \times 3 = 14 + 14 + 14$.

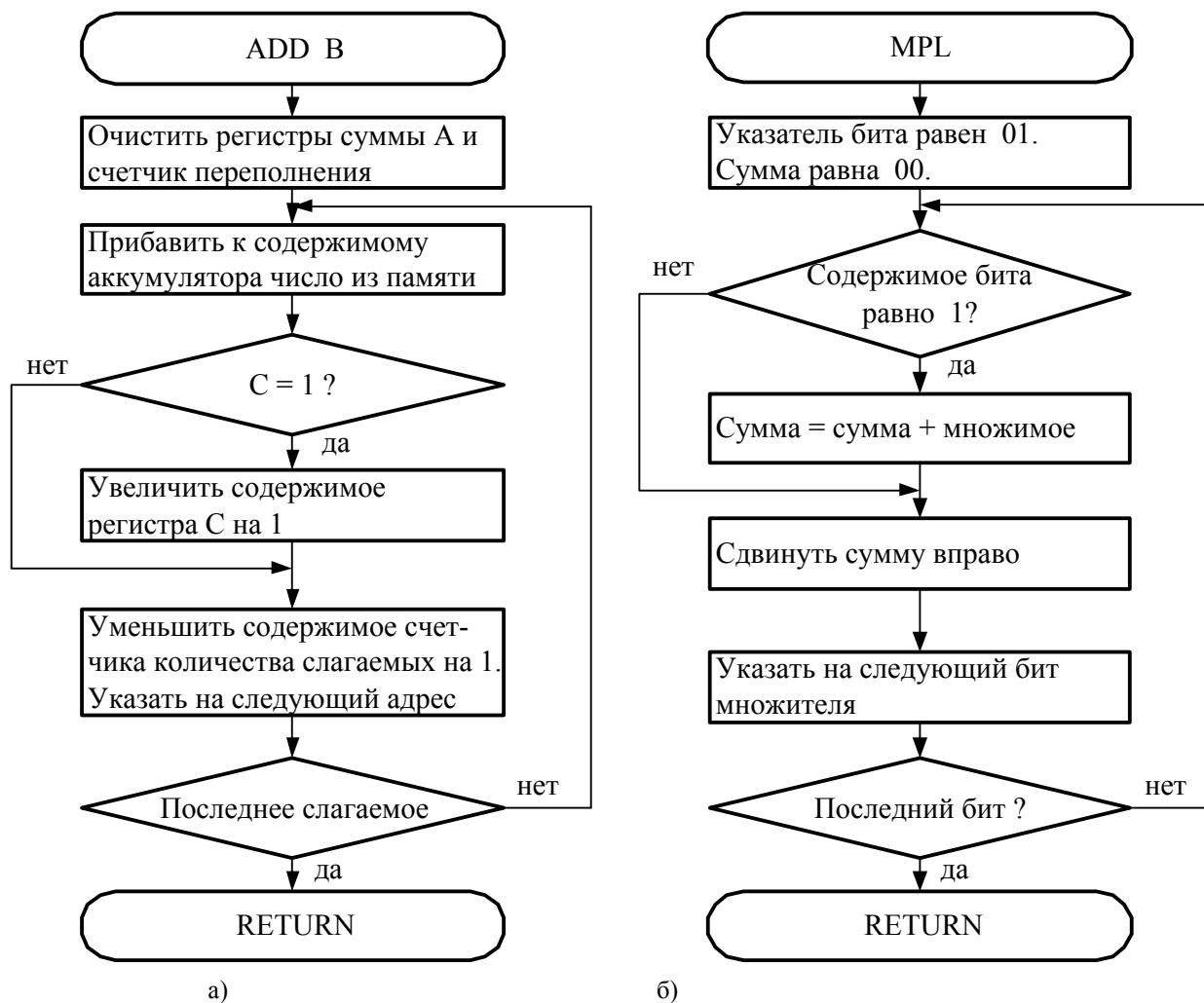


Рис. 4. Алгоритмы подпрограмм сложения (а) и умножения (б).

Недостаток этого способа значительная длительность процесса вычисления. При втором алгоритме умножение осуществляется в столбец, например:

$$\begin{array}{r}
 \begin{array}{r}
 \times 0110 \\
 0011 \\
 \hline
 0110 \\
 0110 \\
 0000 \\
 0000 \\
 \hline
 0010010
 \end{array}
 \begin{array}{l}
 =6_{10} \\
 =3_{10} \\
 \\
 \\
 \\
 \\
 =18_{10}
 \end{array}
 \end{array}$$

При вычислении результата по второму алгоритму необходимо осуществить многократное суммирование со сдвигом влево множимого при одновременной проверке содержимого разрядов множителя, начиная с его младшего разряда. При этом если в очередном разряде множителя записана 1, то множимое прибавляется к сумме и сдвигается влево на один разряд, а если в разряде записан 0, то произойдет только сдвиг множимого. Сдвиг множимого влево можно заменить сдвигом суммы вправо. По этому алгоритму, рис. 4 б, работает подпрограмма 5 умножения двух однокбайтных чисел с получением одного двухбайтного результата. Входные параметры программы: регистр D- множимое, регистр E - множитель. Результат записывается в регистры В и С.

Программа 5.

Адрес	Машинный код	Метка	Мнемокод	Комментарий
04E1	010000	MPL	LXI B, 0000	0 → BC
04E4	3E01		MVI A, 01	Загрузить в A_K указатель разряда
04E6	A7		ANA A	Очистить флаг C
04E7	F5	MPL1	PUSH PSW	Сохранить указатель разряда в стеке
04E8	A3		ANA E	Проверить содержимое очередного разряда множителя
04E9	78		MOV A, B	Старший байт суммы в A_K
04EA	CAEE04		JZ MPL2	Если в разряде 0, то идти в MPL2
04ED	82		ADD D	Прибавить множимое к сумме
04EE	1F	MPL2	RAR	Сдвиг суммы вправо
04EF	47		MOV B, A	Сохранить содержимое A_K в регистре B
04F0	79		MOV A, C	Младший байт суммы в A_K
04F1	1F		RAR	Сдвиг вправо
04F2	4F		MOV C, A	Сохранить содержимое A_K в регистре C
04F3	F1		POP PSW	Получить из стека указатель разряда
04F4	17		RAL	Указатель на следующий разряд
04F5	D2E704		JNC MPL1	Если разряд не последний, то продолжать на MPL1
04F8	C9		RET	Возврат

5. ИСПОЛЬЗУЕМОЕ ОБОРУДОВАНИЕ.

Учебная микроЭВМ типа УМПК 80.

6. ЗАДАНИЕ НА РАБОТУ.

5. Набрать, и отладить и произвести вычисления с программами 4 и 5.
6. Ознакомиться с правилами выполнения команды десятичной коррекции - DAA.
7. Разработайте, выберите и отладьте программу последовательного вычитания нескольких чисел.
8. Разработайте программу деления двоичных однобайтных чисел.
9. Разработайте программу вычисления некоторых функций методом разложения в ряд Маклорена:

$$\sin X = X - \frac{X^3}{3!} + \frac{X^5}{5!} - \frac{X^7}{7!} + \dots \text{ для любого } X \text{ (рад)}$$

$$\ln(1 + X) = X - \frac{X^2}{2!} + \frac{X^3}{3!} - \frac{X^4}{4!} + \dots \text{ для } 0 < X \leq 1$$

5. КОНТРОЛЬНЫЕ ВОПРОСЫ.

1. Какие команды арифметических операций может выполнять МП БИС KP580BM80?
2. На чем основаны алгоритмы работы подпрограмм умножения и деления чисел?
3. Представьте числа от 0 до 20 в двоично-десятичном коде.
4. Как оценить максимальное время выполнения подпрограммы умножения двух чисел (программа 5).

6. СОДЕРЖАНИЕ ОТЧЕТА

Отчёт должен содержать

1. Описание процедуры сложения чисел.
2. Алгоритмы программ
3. Тексты отлаженных программ

7. БИБЛИОГРАФИЧЕСКИЙ СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.

ОСНОВНОЙ

1. Корнеев В.В., Киселев А.В. Современные микропроцессоры. М.: НОЛИДЖ. 2003. -360с.
2. Баранов В.Н. Применение микроконтроллеров AVR: схемы, алгоритмы, программы. М.:Додэка-XXI. 2004.- 288с.
3. Новиков Ю.В., Скоробогатов П.К. Основы микропроцессорной техники: Учебное пособие. М.: Интернет-Университет Информационных Технологий: Бином. 2006.- 359с.
4. Говоров А.А., Фомичев А.А., Панарин В.М., Соколовский Р.В. Малоканальные микропроцессорные контроллеры в автоматических системах регулирования: учебное пособие для вузов. ТулГУ. - Тула: Издательство ТулГУ, 2005.- 360с.
5. Костин А.А. Алгоритмы динамического программирования для анализа сигналов и изображений : Авто-реф.дис...канд.физ.-мат.наук:05.13.18 / А.А.Костин; ТулГУ. — Тула, 2001. — 20с.
6. Вьюгин Михаил Владимирович. Наборы конечных объектов с заданными информационными соотношениями между ними : Автореф. дис. канд. физ.-мат. наук. : Спец. : 01.01.06 / М. В. Вьюгин ; Московский государственный университет имени М. В. Ломоносова. — Москва, 2004. — 15 с. : ил. — Библиогр.: с. 15 (4 назв.).
7. Бойт, К. Цифровая электроника / К.Бойт;пер.с нем.М.М.Ташлицкого. — М. : Техносфера, 2007. — 472с. : ил. — (Мир электроники). — ISBN 978-5-94836-124-6 /в пер./ : 339.26.
8. Новожилов, О.П. Основы микропроцессорной техники. Т.1 : учеб.пособие:в 2 т. / О.П.Новожилов. — М. : Радио-Софт, 2007. — 432с. : ил. — ISBN 5-93037-165-2 : 314.62.
9. Новожилов, О.П. Основы микропроцессорной техники. Т.2 : учеб.пособие:в 2 т. / О.П.Новожилов. — М. : Радио-Софт, 2007. — 336с. : ил. — ISBN 5-93037-167-9 : 287.96.
10. Абрамов, В.М. Электронные элементы устройств автоматического управления:Схемы.Расчет.Справочные данные / В.М.Абрамов. — М. : Академкнига, 2006. — 680с. : ил. — Библиогр.в конце кн. — ISBN 5-94628-222-0 /в пер./ : 368.10.
11. Белов, А.В. Самоучитель по микропроцессорной технике / А.В.Белов. — 2-е изд.,перераб.и доп. — СПб. : Наука и Техника, 2007. — 256с. : ил. — (Радиолюбитель). — Библиогр.в конце кн. — ISBN 978-5-94387-190-0 : 101.15.
12. Нарышкин, А.К. Цифровые устройства и микропроцессоры : учеб.пособие для вузов / А.К.Нарышкин. — М. : Академия, 2006. — 320с. : ил. — (Высшее профессиональное образование:Радиоэлектроника). — Библиогр.в конце кн. — ISBN 5-7695-1618-6 /в пер./ : 217.80.
13. Джексон, Р.Г. Новейшие датчики : монография / Р.Г.Джексон;пер.с англ.под ред.В.В.Лучинина. — М. : Техносфера, 2007. — 384с.
14. Сергиенко, А.Б. Цифровая обработка сигналов : учебник для вузов / А.Б.Сергиенко. — 2-е изд. — М.[и др.] : Питер, 2006. — 751с. —
15. Свердлов, С.З. Языки программирования и методы трансляции : учеб.пособие для вузов / С.З.Свердлов. — М.[и др.] : Питер, 2007. — 638с. : ил. + 1 опт.диск(CD ROM). — (Учебное пособие). — Библиогр.в конце кн. — ISBN 978-5-469-00378-6 /в пер./ : 229.50.
16. Баев, Б.П. Микропроцессорные системы бытовой техники : учебник для вузов / Б.П.Баев. — М. : Горячая линия-Телеком, 2005. — 480с. : ил. — Б
17. «Радио», журнал. 1986г.
18. Вениаминов В.Н., Лебедев О.Н., Мирошниченко А.И. Микросхемы и их применение: Справ. Пособие.- 3-е изд.,перераб. И доп. — М.: Радио и связь, 1989. 240с.: ил.- (Массовая радиобиблиотека. Вып. 1143).
19. Хвощ С.Т. и др. Микропроцессоры и микро-ЭВМ в системах автоматического управления: Справочник/ С.Т. Хвощ, Н.Н. Варлинский, Е.А. Попов; Под общ. Ред. С.Т. Хвоща.- Л.; Машиностроение. Ленингр. отделение, 1987. 640с.: ил.
20. Шило В.Л. Популярныe цифровые микросхемы: Справочник. 2-е изд., испр. — Челябинск: Металлургия, Челябинское отд., 1989. — 352 с.: ил.- (Массовая радиобиблиотека. Вып. 111).

ДОПОЛНИТЕЛЬНЫЙ

1. Микропроцессорные системы автоматического управления / В.А.Бессекерский, Н.Б.Ефимов, С.И.Зиятдинов и др.; Под общ. ред. В.А.Бессекерского.-Л.; Машиностроение. Ленингр. отделение, 1988.365 с.
2. Остром К., Виттенмарк Б. Системы управления с ЭВМ: Пер. с англ. -М. : Мир, 1987.-480 с.
3. Системы автоматического управления с микро-ЭВМ / В.Н. Дроздов, И.В.Мирошник, В.И.Скорубский.-Л.; Машиностроение Ленингр. отделение, 1989.-284 с.
4. Микропроцессоры: в 3-х кн. Кн. 3. Средства отладки лабораторный практикум и задачник: Учеб. для втузов/Н.В. Воробьев, В.Л. Горбунов, А.В. Горячев и др.; Под редакцией Л.Н. Преснухина.-М.: Вьющая шк., 1986. - 351 с.
5. МикроЭВМ: в 3 кн. Практич. пособие/Под редакцией Л.Н.Преснухина, Кн.7. Учебные стенды/1).И.Волков, В.Л.Горбунов, Д.И.Панфилов, С.Г. Шальгин - М.: Высш. шк., 1988.-224 с,
6. Учебная микроЭВМ УМПК-80. ПБА2.390.039.ПС. Паспорт.