

ФЕДЕРАЛЬНОЕ АГЕНСТВО ПО ОБРАЗОВАНИЮ  
Министерство образования и науки Российской Федерации  
Государственное образовательное учреждение  
высшего профессионального образования  
«КАЗАНСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

**МИКРОПРОЦЕССОРЫ И МИКРО-ЭВМ**

**Часть 2**

*Методические указания  
к лабораторным работам*

Казань 2007

Составители: доц. Н.Г. Крикун  
доц. Р.Н. Гайнуллин  
доц. М.Ю. Валеев

Микропроцессоры и микро-ЭВМ. Часть 2: Методические указания / Казан. гос. технол. ун-т; Сост.: Н.Г. Крикун, Р.Н. Гайнуллин, М.Ю. Валеев, Казань, 2007, 32с.

В методических указаниях рассмотрены вопросы практического освоения методики программирования в кодах микропроцессора КР580ИК80А.

Предназначены для выполнения лабораторных работ студентами специальностей 220301 – “Автоматизация технологических процессов и производств”, 230102 – “Автоматизированные системы обработки информации и управления”, 140604 – “Электропривод и автоматика промышленных установок и технологических комплексов”, изучающих дисциплины “Микропроцессорные средства”, “Организация ЭВМ и систем”, «Микропроцессоры и микро-ЭВМ» и «Микропроцессорные средства в электроприводе и технологических комплексах».

Подготовлены на кафедре “Автоматизированных систем сбора и обработки информации”.

Печатается по решению методической комиссии специальностей механического и электротехнического профиля.

Рецензенты: д.ф.-м.н., профессор кафедры «Радиофизики» КГУ В.В. Сидоров  
д.х.н, профессор кафедры «Неорганической химии» КГТУ Р.Р. Назмутдинов

ЛАБОРАТОРНАЯ РАБОТА 1  
**ИЗУЧЕНИЕ УЧЕБНО-ОТЛАДОЧНОГО УСТРОЙСТВА**  
**“ЭЛЕКТРОНИКА-580”**

**ЦЕЛЬ РАБОТЫ:** Приобрести практические навыки работы с пультом управления учебно-отладочного устройства "Электроника-580" (Э580), изучить возможности устройства Э580, практически освоить режимы его работы.

Продолжительность лабораторного занятия 4 часа; самостоятельная подготовка - 4 часа.

**1. САМОСТОЯТЕЛЬНАЯ РАБОТА**

1.1. Изучить и практически освоить представление чисел в двоичной и шестнадцатеричной системах счисления.

1.2. Изучить назначение и структуру учебно-отладочного устройства Э580.

1.3. Изучить назначение клавиш пульта управления и режимы работы отладочного устройства.

1.4. Изучить последовательность действий при включении устройства, вводе программы в память отладочного устройства, чтении содержимого ячеек памяти и программно-доступных регистров микропроцессора (МП) КР580ИК80А и записи в них информации.

1.5. Изучить последовательность действий при выполнении программы в пошаговом режиме и режиме с остановом по контрольным точкам.

1.6. Вычислить выражение:  $N = n * 9$ , где **n** Ваш номер в журнале преподавателя.

Полученное число **N** представить в шестнадцатеричной и двоичной системе счисления.

1.7. Ознакомиться с приведенной в таблице 1 программой сложения трех чисел.

1.8. Согласно приведенному выше примеру сложить число, полученное Вами при выполнении операции произведения (пункт 1.6) с числами **N1** и **N2** (см. Приложение 1). Порядок выполнения операции сложения записать в виде программы и оформить в виде таблицы.

В программе нахождения суммы трех чисел для МП К580ИК80 использованы следующие обозначения и исходные данные:

**N** - первое слагаемое

**N1** - второе слагаемое

**N2** - третье слагаемое

**8300** - адрес начала программы

Таблица 1

**Программа сложения трех чисел на Э580**

Адрес	Машинный код	Метка	Мнемокод	Комментарий
8300 8301	3E N		MVI A	загрузка в АКМ числа N
8302 8303	0E N1		MVI C	загрузка в рег. C числа N1
8304 8305	16 N2		MVI D	Загрузка в рег. D числа N2
8306	81		ADD C	A + C
8307	82		ADD D	A + D
8308	00		NOP	Окончание программы

**2. СОДЕРЖАНИЕ И ПОРЯДОК ВЫПОЛНЕНИЯ  
ЛАБОРАТОРНОЙ РАБОТЫ**

1. Включить отладочное устройство.
2. Прочитать содержимое ячейки памяти с адресом 8200.
3. Записать N в ячейку памяти с адресом 8237.
4. Прочитать содержимое ячейки памяти с адресом 1026.
5. Проверить, возможна ли запись числа N в ячейку памяти с адресом 2145.
6. Объяснить, почему невозможна запись числа N в ячейки памяти с адресами менее 8000.
7. Прочитать содержимое всех программно-доступных регистров МП.
8. Записать число N в один из регистров, например в регистр В.
9. Выполнить программу, приведенную в таблице 1 в пошаговом режиме.
10. Записать в регистр-аккумулятор А МП число F8.
11. Составить программу сложения числа F8, записанного в аккумуляторе (АКМ) с

числом N , записанным в ячейке памяти с адресом 8308.

12. Проверить полученный результат.

### **3. СОДЕРЖАНИЕ ОТЧЕТА**

1. Краткое описание цели работы и результатов самостоятельной подготовки к лабораторному занятию.

2. Тексты программ выполнения лабораторной работы.

3. Протокол работы, отражающий последовательность действий на пульте отладочного устройства при выполнении программ.

### **4. КОНТРОЛЬНЫЕ ВОПРОСЫ**

1. Что такое микро-ЭВМ?

2. Что такое микропроцессор? Его существенные отличия от микро-ЭВМ?

3. Нарисуйте структурную схему отладочного устройства и поясните назначение каждого из её элементов.

4. Какие области памяти доступны программисту?

5. Какие адреса памяти относятся к ПЗУ и ОЗУ?

6. В чем заключается принципиальное отличие ПЗУ от ОЗУ?

7. Что происходит при попытке записи данных в ПЗУ?

8. Сохраняется ли в памяти Э580 программа после выключения питания?

9. Что такое программа-монитор и ее функции?

10. Объясните назначение каждой клавиши на пульте управления Э580?

11. Какие признаки можно увидеть на индикаторах пульта управления и как зафиксировать другие признаки?

12. Меняется ли содержание программы, записанной в ячейках памяти с номерами 8310÷842F, если проверить содержимое ячейки 83FD?

## **ЛАБОРАТОРНАЯ РАБОТА 2**

### **ИЗУЧЕНИЕ АРИФМЕТИЧЕСКИХ КОМАНД И КОМАНД ПЕРЕСЫЛКИ ДАННЫХ**

**ЦЕЛЬ РАБОТЫ:** изучить команды пересылки данных и арифметические команды, способы адресации данных при выполнении простых программ по нахождению суммы

ряда чисел и сложении (вычитании) с переносом (заёмом).

Продолжительность лабораторного занятия 4 часа, самостоятельная подготовка - 4 часа.

## **1. САМОСТОЯТЕЛЬНАЯ РАБОТА**

1.1. Ознакомиться с настоящим описанием.

1.2. Изучить способы адресации данных МП К580ИК80А:

1) - непосредственная;

2) - прямая;

3) - косвенная;

4) - регистровая.

1.3. Изучить назначение разрядов регистра признаков МП К580ИК80.

1.4. Изучить назначение группы арифметических команд.

1.5. В соответствии с заданием, выданным преподавателем, подготовить данные с указанием для каждой вновь изучаемой команды следующие её характеристики:

1) - шестнадцатеричный код;

2) - мнемокод;

3) - формат команды;

4) - используемый способ адресации;

5) - действие, выполняемое командой;

6) - влияние результата выполнения команды на состояние соответствующих разрядов регистра признаков.

## **2. СОДЕРЖАНИЕ ЛАБОРАТОРНОЙ РАБОТЫ**

Задачи, решаемые на ЭВМ, не сводятся к обработке отдельного элемента данных с помощью одной операции. Напротив, они требуют обработки многих элементов данных (например, массива или блока данных), которые занимают отдельные ячейки памяти. Программа может выполнять одну и ту же операцию над содержимым ячеек, причем выполнять ее многократно с помощью программных циклов.

Например, необходимо найти сумму ряда чисел. При этом задача состоит в том, чтобы осуществить суммирование нескольких чисел сразу. Эти числа могут представлять собой совокупности входных сигналов, находящихся под управлением системы, число изделий (или число сообщений), изготовленных (или принятых) за определенный промежуток времени. Предположим, что сумма не превышает  $255_{10}$  и для

её хранения достаточно одной 8-битовой ячейки памяти.

В программе суммирования ряда чисел для МП К580ИК80 использованы следующие исходные данные и обозначения:

**B** - регистр-счетчик, указатель длины массива

**NNNN** - адрес ячейки памяти, где указана длина массива

**HL** - регистровая пара, адресный указатель данных

**DDDD** - адрес ячейки памяти, где хранится первый элемент

**8100** - адрес начала программы

Таблица 1

**Программа сложения ряда чисел**

Адрес	Машинный код	Метка	Мнемокод	Комментарий
8100	3A	M2	LDA	загрузка длины ряда чисел в АКМ
8101	NN			из ячейки памяти с адресом
8102	NN			NNNN
8103	47		MOV B,A	загрузка длины ряда чисел в регистр-счетчик
8104	97		SUB A	Обнуление аккумулятора
8105	21		LXI H	загрузка адресного указателя
8106	DD			DDDD в регистровую пару HL
8107	DD			
8108	86		ADD M	сложение элемента массива с АКМ
8109	05		DCR B	уменьшение содержимого регистра-счетчика на единицу
810A	CA	M1	JZ, M1	сложение элементов массива
810B	11			закончено?
810C	81			
810D	23		INX H	переход к следующему адресу
810E	C3		JMP, M2	организация цикла
810F	08			
8110	81			
8111	00		NOP	окончание программы

Перед выполнением программы исходный ряд чисел заносится с помощью командных клавиш в область памяти, начиная с адреса **DDDD** (см. Приложение 2), а его длина - в ячейку памяти с адресом **NNNN**.

В большинстве случаев точность вычислений, обеспечиваемая одной ячейкой памяти, недостаточна. Поэтому рассмотрим арифметику для чисел, занимающих несколько ячеек. Задача состоит в том, чтобы сложить два числа длиной более 8 бит каждое. Исходные числа располагаются в области памяти таким образом, что сначала идут младшие разряды, а затем более старшие разряды. Полученную сумму необходимо поместить в те ячейки памяти, где хранилось первое число.

В программе суммирования двух длинных чисел для МП К580ИК80 использованы следующие исходные данные и обозначения:

**N** - длина чисел в байтах

**B** - регистр-счетчик, указатель длины чисел

**DE** - регистровая пара, адресный указатель первого слагаемого

**NNNN** - адрес ячейки памяти, где хранится младший байт первого числа

**HL** - регистровая пара, адресный указатель второго слагаемого

**DDDD** - адрес ячейки памяти, где хранится младший байт второго числа

**8300** – адрес начала программы

Таблица 2

**Программа сложения двух длинных чисел**

Адрес	Машинный код	Метка	Мнемокод	Комментарий
8300	06	M2	MVI B	Загрузка счетчика
8301	N		LXI D	загрузка адресного указателя первого числа
8302	11			
8303	NN			
8304	NN			
8305	21		LXI H	загрузка адресного указателя второго числа
8306	DD			
8307	DD			
8308	1A		LDAX D	загрузка операнда в АКМ
8309	8E		ADC M	сложение с учетом переноса
830A	12		STAX D	Запоминание результата сложения



Таблица 2 (продолжение)

830B	05		DCR B	Уменьшение счетчика на единицу
830C	CA		JZ, M1	сложение закончено?
830D	14			
830E	83			
830F	13		INX D	переход к следующему байту первого числа
8310	23		INX H	переход к следующему байту второго числа
8311	C3		JMP, M2	организация цикла
8312	08			
8313	83			
8314	00	MI	NOP	окончание программы

Перед выполнением программы исходные числа **F1** и **F2**, начиная с младшего байта, заносятся последовательно байт за байтом с помощью командных клавиш в ячейки памяти, начиная с адресов **NNNN** и **DDDD** соответственно (см. Приложение 3).

### 3. ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

1.1. Подготовленные в процессе самостоятельной работы команды (пересылки, арифметические и др.) необходимо выполнить в следующей последовательности;

1.2. Записать необходимые операнды с помощью командных клавиш отладочного устройства;

1.3. Записать изучаемую команду в память,

1.4. Выполнить команду в пошаговом режиме;

1.5. Проверить и зафиксировать соответствие ожидаемых и наблюдаемых, результатов, т.е. прочитать содержимое соответствующей ячейки памяти или программно-доступного регистра.

1.6. По заданию преподавателя подготовить и выполнить программы сложения (вычитания) ряда чисел и сложения длинных чисел с переносом (заёмом). Исходные данные необходимо взять из Приложений 2 и 3 соответственно.

#### **4. СОДЕРЖАНИЕ ОТЧЕТА**

1. Краткое описание цели работы и результатов самостоятельной подготовки к лабораторному занятию.
2. Тексты программ выполнения лабораторной работы.
3. Протокол, отражающий последовательность действий на пульте отладочного устройства и результаты выполнения команд МП К580ИК80 с записью последовательности нажатых клавиш на пульте отладочного устройства.

#### **5. КОНТРОЛЬНЫЕ ВОПРОСЫ**

1. Что такое команда МП?
2. Что такое система команд МП?
3. Назовите группы команд МП К580ИК80?
4. Из каких частей состоит команда МП?
5. Что такое мнемокод?
6. Назовите способы адресации, используемые в МП К580ИК80А?
7. Для чего служат команды пересылки данных?
8. Назовите значения разрядов регистра признаков МП К580ИК80А?
9. Какие из признаков формируются при выполнении команды пересылки данных?
10. После выполнения команды пересылки данных MOV C,B сохраняются ли исходные данные в регистре В?
11. Приведите примеры команд пересылки данных с прямой, косвенной, регистровой адресацией.
12. В каких байтах располагаются младшие и старшие разряды адреса в командах, содержащих адрес ячейки памяти?
13. В ячейке памяти с адресом 8314 находится число F4. Какие команды можно применить для записи его в регистр С?
14. В регистре D находится число E4. Какие команды можно использовать для записи его в ячейку памяти с адресом 8147?
15. В каких регистрах МП К580ИК80 может храниться адрес ячейки памяти?
16. Для чего служат арифметические (логические команды)?
17. Данные, какой длины может обрабатывать МП К580ИК80А?
18. Где хранятся операнды, и куда помещается результат после выполнения арифметических (логических) команд?
19. Какие разряды в регистре признаков формируются при выполнении

арифметических (логических) команд?

20. Для чего используются команды положительного и отрицательного приращения (INR, DCR, INX, DCX)?

21. Как выполняется команда сравнения двух чисел? Где формируется результат выполнения этой команды?

22. Как выполняются команды простого и циклического сдвига содержимого аккумулятора?

23. Над содержимым каких программно-доступных регистров МП К580ИК80А можно выполнить команду сдвига?

### **ЛАБОРАТОРНАЯ РАБОТА 3 ОБРАБОТКА МАССИВОВ ИНФОРМАЦИИ ОРГАНИЗАЦИЯ ЦИКЛОВ**

**ЦЕЛЬ РАБОТЫ:** Освоить методику программирования в кодах микропроцессора К580ИК80А при обработке массивов, выполнении и отладке программ на устройстве "Электроника 580".

Продолжительность лабораторного занятия - 4 часа, самостоятельная подготовка - 4 часа.

#### **1. САМОСТОЯТЕЛЬНАЯ РАБОТА**

1.1. Ознакомиться с настоящим описанием.

1.2. Изучить группу команд передачи управления МП К580ИК80А.

1.3. Изучить рекомендуемую литературу [1;2;3] и разделы конспекта лекций по программированию в кодах МП К580ИК80А.

1.4. В соответствии с вариантом задания, выдаваемого преподавателем, составить программу в кодах МП К580ИК80А и подготовить на неё полную документацию, включающую в себя:

- схему алгоритма;
- распределение памяти;
- текст программы в кодах МП К580ИК80А.

1.5. Результаты выполнения самостоятельной работы представить в виде отчета.

## 2. СОДЕРЖАНИЕ ЛАБОРАТОРНОЙ РАБОТЫ

При программировании часто возникает необходимость в пересылке массива данных из одной области памяти в другую. Такая операция может потребоваться при инициализации элементов массива, при пересылке или формировании строки данных, выводимой на экран дисплея.

Рассмотрим следующую задачу. Массив чисел расположен в области памяти, начиная с адреса **NNNN**, и состоит из **N** элементов. Необходимо переслать массив в другую область памяти, начиная с адреса **DDDD**. В программе для МП К580ИК80А в качестве адресного регистра используется регистровая пара **HL**. В качестве адресных регистров можно также использовать регистры **B** и **C** (регистровая пара **BC**), регистры **D** и **E** (регистровая пара **DE**), но только для выполнения операций перемещения данных в аккумулятор и обратно (с помощью команд **LDAX** и **STAX**). Эти адресные регистры нельзя использовать в командах, выполняющих арифметические и логические операции, а также операции пересылки данных в регистры общего назначения или из регистров общего назначения. Необходимо учитывать, что при каждом проходе программа должна изменять содержимое обоих адресных регистров. Один и тот же набор регистров должен совместно использоваться в качестве счетчиков указателя адреса и рабочих ячеек.

В программе пересылки массива из одной области памяти в другую использованы следующие исходные данные и обозначения:

**N** - количество элементов массива

**D** - регистр-счетчик, указатель длины массива

**HL** - адресный указатель исходного массива

**NNNN** - адрес ячейки памяти, где хранится первое число  
исходного массива

**BC** - адресный указатель результирующего массива

**DDDD** - адрес ячейки памяти, с которой начинается результирующий массив

**8500** - адрес начала программы

Таблица 1

**Пересылка массива информации из одной области памяти в другую**

Адрес	Машинный код	Метка	Мнемокод	Комментарий
8500	16	M2	MVI D	загрузка счетчика
8501	N		LXI H	загрузка адресного указателя исходного массива
8502	21			
8503	NN			
8504	NN		LXI B	загрузка адресного указателя результатирующего массива
8505	01			
8506	DD			
8507	DD		MOV A,M STAX B DCR D JZ, M1	пересылка числа из памяти в АКМ пересылка числа из АКМ в память уменьшение счетчика на единицу пересылка массива закончена?
8508	7E			
8509	02			
850A	15			
850B	CA			
850C	13			
850D	85			
850E	23		INX H	переход к следующему адресу в HL
850F	03		INX B	переход к следующему адресу в BC
8510	C3	M1	JMP, M2	организация цикла
8511	08		NOP	окончание программы
8512	85			
8513	00			

Перед выполнением программы исходный массив чисел заносится в память, с помощью командных клавиш, начиная с адреса NNNN (см. Приложение 4).

В ряде случаев возникает необходимость выбрать из массива информации данные, представляющие собой ряд максимальных, либо минимальных величин. Предположим, что необходимо написать программу для решения следующей задачи. Дан массив **A1**, состоящий из **N** однобайтовых чисел. Необходимо переписать из массива **A1** в массив **B1** все числа в диапазоне от **H1** до **H2**.

В программе выборки из массива чисел заданного интервала использованы

следующие исходные данные и обозначения:

**N** - количество элементов исходного массива

**L** - счетчик, указатель длины исходного массива

**BC** - адресный указатель исходного массива

**NNNN** - адрес ячейки памяти, где хранится первое число исходного массива.

**DE** - адресный указатель результирующего массива

**DDDD** - адрес ячейки памяти, куда будет записано первое число результирующего массива.

**H1** - нижний предел диапазона выборки

**H2** - верхний предел диапазона выборки

**8100** - адрес начала программы

Перед выполнением программы исходный массив чисел **A1** заносится в память с помощью командных клавиш, начиная с адреса **NNNN** (см. Приложение 5)

Таблица 2

**Выборка из массива данных чисел заданного интервала**

Адрес	Машинный код	Метка	Мнемокод	Комментарий
8100	2E	M2	MVI L	загрузка счетчика
8101	N		LXI B	загрузка адресного указателя исходного массива
8102	01			
8103	NN			
8104	NN			
8105	11		LXI D	загрузка адресного указателя результатирующего массива
8106	DD			
8107	DD			
8108	0A		LDAX B	выборка числа из массива A1
8109	FE			
810A	H1		JC, M1	сравнение с нижним пределом - числом H1 переход на метку M1, если число < H1
810B	DA			
810C	15			
810D	81			

Таблица 2 (продолжение)

810E	FE		CPI	сравнение с верхним пределом -
810F	H2			числом H2
8110	D2		JNC, M1	переход на метку M1, если
8111	15			число > H2
8112	81			
8113	12		STAX D	запись числа, попавшего в
				заданный интервал, в память
8114	13		INX D	переход к следующему адресу в
				DE
8115	03	M1	INX B	переход к следующему адресу в
				BC
8116	2D		DCR L	уменьшение счетчика
8117	C2		JNZ, M2	организация цикла
8118	08			
8119	81			
811A	00		NOP	окончание программы

### 3. ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

1. По заданию преподавателя подготовить и выполнить программы пересылки массива информации из одной области памяти в другую и выбрать из массива данных числа заданного интервала. Исходные данные необходимо взять из Приложений 4 и 5 соответственно.

### 4. СОДЕРЖАНИЕ ОТЧЕТА

1. Краткое описание цели работы и результатов самостоятельной подготовки к лабораторному занятию.

2. Тексты программ выполнения лабораторной работы.

3. Протокол, отражающий последовательность действий на пульте отладочного устройства и результаты выполнения команд МП К580ИК80А с записью последовательности нажатых клавиш на пульте отладочного устройства.

## **5. КОНТРОЛЬНЫЕ ВОПРОСЫ**

1. Назовите назначение разрядов регистра признаков МП К580ИК80А?
2. По каким условиям записывается 1 в каждый из разрядов регистра признаков МП К580ИК80А?
3. Какие функции выполняют команды перехода?
4. Какие существуют способы изменения последовательности выполнения команд?
5. Как осуществляется выполнение команды перехода?
6. В каких байтах располагаются младшие и старшие разряды в командах перехода?
7. По алгоритму задачи необходимо принять решение "больше". Какими командами можно это реализовать?
8. По алгоритму задачи необходимо принять решение "равно". Какими командами можно это реализовать?
9. Как организуется цикл в системе команд МП К580ИК80А?
10. Какое максимальное число команд может быть в теле цикла?
11. Какое максимальное число команд имеет параметр цикла?
12. Как организуется работа с массивами?
13. Какая область памяти отладочного устройства может использоваться под запись массива данных?
14. С какой целью следует оставлять "пустые" команды в программе?
15. Назовите способы разработки и отладки программ для микропроцессоров и микро-ЭВМ?
16. Охарактеризуйте достоинства и недостатки используемой в отладочном устройстве технологии разработки и отладки программ?



## ЛАБОРАТОРНАЯ РАБОТА 4

### ПОДПРОГРАММА И СТЕК

ЦЕЛЬ РАБОТЫ: Исследование особенностей записи и обращения к подпрограммам, изучение методов использования стека при разработке программ с помощью учебно-отладочного устройства "Электроника 580".

Продолжительность лабораторного занятия - 4 часа, самостоятельная подготовка - 4 часа.

#### 1. СОДЕРЖАНИЕ ЛАБОРАТОРНОЙ РАБОТЫ

Учитывая ограниченные возможности памяти при разработке программ, необходимо стремиться сделать их как можно короче. С этой целью часть программы, которая неоднократно повторяется, или программа, которая часто используется, могут быть использованы 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.

Для вызова подпрограмм и возврата из них используются команды **CALL<A2>,<AI>** и **RET**.

При работе с подпрограммами используется стековая память. Стек - специально организованная область ОЗУ, используемая для временного сохранения данных или адресов. Число, записанное в стек последним, извлекается из него первым.

Команда **RET** помещает в программный счетчик последнее записанное на данный момент в стеке число. После этого выполнение программы будет осуществляться с этого адреса. Любая подпрограмма должна заканчиваться 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 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 (аналогично командам условного перехода) без использования дополнительных команд.

Помимо команд вызова подпрограмм и возврата из них, со стеком можно обмениваться информацией с помощью команд **PUSH<R>** (записать в стек

содержание обозначенного регистра МП БИС) и **POP<R>** (записать данные из стека в обозначенный регистр МП БИС)

**Программа исследования процесса выполнения команд вызова и возврата из подпрограмм, а также команд работы со стеком**

Таблица 1

Адрес	Машинный код	Метка	Мнемокод	Комментарий
8200	31	M1	LXI SP	записать в указатель стека SP
8201	C0		C0	адрес 0BC0
8202	0B		0B	
8203	CD		CALL, M1	вызвать подпрограмму M1
8204	07		07	
8205	82		82	
8206	CF		RST 1	остановить выполнение подпрограммы
8207	F5		PUSH	записать слово состояние МП БИС в стек
8208	C5		PUSH B	записать содержимое рег. BC в стек
8209	D5		PUSH D	записать содержимое рег. DE в стек
820A	E5		PUSH H	записать содержимое рег. HL в стек
820B	3E		MVI A	записать в рег. A число 05
820C	05		05	
820D	47		MOVB,A	записать число из рег. A в B
820 E	87		ADD A	удвоить содержимое аккумулятора
820F	5F		MOVE,A	записать число из рег. A в E
8210	67		MOV H,A	записать число из рег. A в H
8211	E1		POP H	записать число из стека в рег. HL
8212	D1		POP D	записать число из стека в рег. DE
8213	C1		POP B	записать число из стека в рег. BC
8214	F1		POP PSW	записать слово состояние из стека в МП БИС
8215	C9		RET	возврат подпрограммы

### 3. ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

1. По заданию преподавателя подготовить и выполнить программу, указанную в таблице 1.

### 4. СОДЕРЖАНИЕ ОТЧЕТА

1. Краткое описание цели работы и результатов самостоятельной подготовки к лабораторному занятию.

2. Тексты программ выполнения лабораторной работы.

3. Протокол, отражающий последовательность действий на пульте отладочного устройства и результаты выполнения команд МП K580ИК80А с

записью последовательности нажатых клавиш на пульте отладочного устройства.

## ПРИЛОЖЕНИЕ 1

### Варианты заданий к лабораторной работе 1

N п/п	Первое слагаемое N	Второе слагаемое N1	Третье слагаемое N2
1		1D	25
2		A2	13
3		24	3F
4		61	2E
5		35	3A
6		3E	4A
7		28	2C
8		3B	54
9		34	11
10		A1	17
11		31	12
12		5D	0F
13		29	1E
14		26	19
15		14	42
16		3A	17
17		24	44
18		39	24
19		23	2D
20		34	45
21		21	12
22		1F	3D
23		2A	2A
24		41	52
25		13	31

ПРИЛОЖЕНИЕ 2

**Варианты заданий к лабораторной работе 2**

N п/п	NNNN	DDDD	Исходный ряд чисел
1	8203	832F	12, 23, 1D, 2F, 19, 3D, 0F
2	8132	8263	26, 36, 2F, 3D, 22
3	8233	8553	24, 1A, 29, 31, 42, 0D
4	8317	8400	2E, 1D, 35, 27, 1C
5	8267	8154	39, 34, 0D, 1F, 3C, 21
6	8102	8300	34, 24, 45, 2C, 23
7	8430	8521	62, 1D, 1C, 1A, 1E, 24
8	8100	832D	A4, 1F, 23, 0D
9	8523	8200	17, 25, 3D, 1A, 23, 41
10	8423	8145	27, 2F, 1A, 3D, 09
11	841D	8323	24, 12, 2A, 32, 07, ID
12	8705	8305	21, 36, 18, 3A, 12, 1E
13	8320	8100	13, 35, 2A, 2D, 2F, 19
14	831F	8053	14, 34, 23, 3B, IF
15	8424	812A	11, 21, 31, 23, 41, 0D
16	8503	8401	3A, 24, 4A, 16, 14, 07
17	8204	8312	2F, 2A, 2D, 2C, 2E
18	843F	8700	14, 25, 3A, 2D, 48, OF
19	8303	8500	61, 2F, ID, 1A, 2D, IF
20	8319	8473	23, 43, 2F, 1D, 2E
21	81FA	831C	3A, 3D, 2A, 11, 18, 04
22	8316	8400	31, 08, 3F, 24, 5A, 11
23	8306	8419	11, 45, 34, 43, 1E
24	8003	8520	38, 2F, 4A, 23, 19, 08
25	8302	8408	24, 16, 07, 2F, 41, 2E

**Варианты заданий к лабораторной работе 2**

N п/п	NNNN	DDDD	F1 – первое слагаемое	F2 - второе слагаемое
1	8165	8231	12A3DE	23D6A8
2	8234	8421	A4E6FF	18D5A4
3	8110	8311	35D6E5	8567E4
4	8200	8012	AD3312	342EA6
5	8234	8500	E6D556	D34520
6	8100	8435	E4A5D2	3F56A5
7	8131	8015	86DE54	9A56D3
8	8310	8200	D46618	341198
9	8402	8300	37E529	19075D
10	8238	8423	539AE7	76D5A3
11	82A4	831F	8D45F1	D3E45C
12	8209	8410	4B56C4	A41209
13	80AE	8126	7DAED2	1F549A
14	8239	8100	3FDD24	96F235
15	8450	8502	73550F	23097E
16	8005	8412	DE5319	1120F8
17	8403	8509	9F5CD7	4980DE
18	8201	8321	9302F4	18DAF3
19	8111	8300	83041F	1EF5D5
20	8415	8500	D63690	2017AE
21	8152	8542	350971	5AE6DD
22	82F3	81ED	3290D5	56DEAF
23	854D	8203	4539DE	3216D7
24	8398	82AE	EF2890	1078DE
25	8502	8124	2605ED	54DEA6

**Варианты заданий к лабораторной работе 3**

N п/п	NNNN	DDDD	Исходный массив
1	8022	8402	23, 5A, 32, D4, 27, 1D, 38, 01, CF
2	8501	8321	ED, 3D, 4D, 49, 9D, AA, F5, ED, 98
3	8306	8534	ED, DF, 6A, D7, 98, 02, B3, 36
4	8102	8422	28, 3F, ED, 6D, AE, A6, 87, 90
5	8541	8027	D2, 25, 87, 02, DF, 67, DF, 46, AD
6	8521	8306	45, 89, DF, 24, 91, 34, E5, 67, DA
7	8400	8201	13, 2F, 6E, A7, 83, 45,10, AE
8	8302	845F	67, 01, DE, 34, 91, AD, ED, 76
9	8411	8509	11, 34, 2D, AE, 85, DE, 29, 01, 67
10	8310	82EF	34, 87, 51, AE, 5C 44, 98
11	8205	8342	46, 93, 56, 02, 83, 47,10, 88
12	8511	8453	D3, A7, 23, F4, 65,19, OF, 12
13	8054	8078	57, AF, D6, 7D, 9A, 90, 29, 02, 34
14	8200	8456	23, DE, 37, 83, CB, 93, 76
15	8230	8209	87, 03, 67, 71, 6A, 27
16	8322	8211	09, ED, 37, FD, 87, 01, 35
17	8267	81D5	ED, 03, 45, DE, BE, 89,10, 90
18	8241	8156	87, 39, 02, 09, 73, ED, EF, 94, 34
19	8200	8342	67, 45, DF, 20, 64, 30
20	8276	8165	48, 71, 93, 28, F4, 38, 98
21	826D	8312	EF, D4, 38, 93, B5, 2E, 9E, DD
22	8203	8123	45, 94, 03, EE, F9, B4, 83
23	8310	8310	97, 40, ED, 46, 98, 40, 07
24	82F4	8100	54, AE, DC EB, 98, 94, 03, 32
25	8276	8198	11, 87, 34, AE, 09, C4, 36, 20

**Варианты заданий к лабораторной работе 3**

№ п/п	NNNN	DDDD	Исходный массив	H1	H2
1	8032	8442	23,5A,32,D4,27,1D,38	IF	C1
2	8304	8322	ED,3D,4D,49,9D,AA,F5	45	F1
3	830E	8531	ED,DF,6A,D7,98,02,B3	13	D6
4	810D	8442	28,3F,ED,6D,AE,A6,87	2E	A8
5	85F1	8127	D2,25,87,02,DF,67,DC	33	CD
6	8561	8326	45,89,DF,24,91,34,E5	35	9D
7	8408	8200	13,2F,6E,A7,83,45,10	6A	D3
8	8392	845F	67,01,DE,34,91,AD,ED	23	ED
9	8415	8579	11,34,2D,AE,85,DE,29	19	CF
10	831E	828F	34,87,51,AE,5C,44,98	24	A9
11	8255	8302	46,93,56,02,83,47,10	39	D3
12	85F1	8473	D3,A7,23,F4,65,19,0F	21	EE
13	80E4	8038	57,AF,D6,7D,9A,90,29	34	ED
14	8204	8422	23,DE,37,83,CB,93,76	09	78
15	8231	8202	87, 03, 67, 71,6A, 27, 3D	12	68
16	8320	8221	09, ED 37, FD 87,01, 35	1E	EF
17	8264	8125	ED 03,45, DE ,BE 89,10	31	AD
18	8211	8116	87,39,02,09,73, ED,EF	29	DA
19	8200	8312	67,45,DF,20,64,30,AE	32	DE
20	8274	8125	48,71,93,28, F4,38,98	31	F2
21	825D	8322	EF ,D4, 38,93, B5, 2E, 9E	45	ED
22	8201	8143	45,94,03, EE, F9, B4 ,83	23	BE
23	8312	8310	97,40, ED, 46,98,45 ,07	21	D9
24	82A4	8140	54, AE, DC ,EB, 98,94,03	10	C8
25	8216	8148	11,87,34, AE, 09, C4, 36	1A	E6

## Система команд МП КР580ИК80А

Мнемоника	Описание команды	Код команды								Длина Коман- ды, байт	Число Тактов	Флаги условий
		D7	D6	D5	D4	D3	D2	D1	D0			S Z A C P CY
MOV R1, R2	Передача из регистра R2 в регистр R1	0	1	D	D	D	S	S	S	1	5	- - - - -
MOV M, R	Передача из регистра в память	0	1	1	1	0	S	S	S	1	7	- - - - -
MOV R, M	Передача из памяти в регистр	0	1	D	D	D	1	1	0	1	7	- - - - -
MVI R	Передача байта в регистр	0	0	D	D	D	1	1	0	2	7	- - - - -
MVI M	Передача байта в память	0	0	1	1	0	1	1	0	2	10	- - - - -
LXI RP	Загрузка парных регистров B-C, D-E, H-L, S-P	0	0	R	R	0	0	0	1	3	10	- - - - -
LDAX RP	Загрузка аккумулятора По адресу, указанному Парой регистров B-C или D-E	0	0	R	R	1	0	1	0	1	7	- - - - -
STAX RP	Занесение содержимого аккумулятора по адресу, указанному парой регистров B-C или D-E	0	0	R	R	0	0	1	0	1	7	- - - - -
LDA	Загрузка аккумулятора по адресу, указанному в команде	0	0	1	1	1	0	1	0	3	13	- - - - -
STA	Занесение содержимого аккумулятора по адресу, указанному в команде	0	0	1	1	0	0	1	0	3	13	- - - - -
LHLD	Загрузка регистров L, H из двух соседних ячеек, начиная с адреса, указанного в команде	0	0	1	0	1	0	1	0	3	16	- - - - -
SHLD	Занесение содержимого регистров L, H в две соседние ячейки, начиная с адреса, указанного в команде	0	0	1	0	0	0	1	0	3	16	- - - - -
XCHG	Обмен данными между парами регистров H-L и D-E	1	1	1	0	1	0	1	1	1	4	- - - - -
XTHL	Обмен данными между SP и H-L	1	1	1	0	0	0	1	1	1	18	- - - - -



SPHL	Занесение содержимого регистра H-L в SP	1	1	1	1	1	0	0	1	1	5	- - - - -
PUSH RP	Ввод содержимого регистров B-C, D-E или H-L в стэк	1	1	R	R	0	1	0	1	1	11	- - - - -
PUSH PSW	Ввод PSW в стэк	1	1	1	1	0	1	0	1	1	11	- - - - -
POP RP	Выдача данных из стека в регистры B-C, D-E, H-L	1	1	R	R	0	0	0	1	1	10	- - - - -
POP PSW	Выдача данных из стека в аккумулятор и регистр признаков	1	1	1	1	0	0	0	1	1	10	++ + + +
ADD R	Сложение содержимого регистра и аккумулятора	1	0	0	0	0	S	S	S	1	4	++ + + +
ADC R	То же, но с учётом переноса CY	1	0	0	0	0	S	S	S	1	4	++ + + +
ADD M	Сложение содержимого ячейки памяти и аккумулятора	1	0	0	0	0	1	1	0	1	7	++ + + +
ADC M	То же, но с учётом переноса CY	1	0	0	0	1	1	1	0	1		++ + + +
ADI	Сложение байта с содержимым аккумулятора	1	1	0	0	0	1	1	0	2	7	++ + + +
ACI	Сложение байта с содержимым аккумулятора с учётом переноса CY	1	1	0	0	1	1	1	0	2	7	++ + + +
DAD RP	Сложение содержимого пар регистров B-C, D-E, H-L, SP с содержимым пары H-L	0	0	R	R	1	0	1	0	1	10	- - - - +
SUB R	Вычитание содержимого регистра из содержимого аккумулятора	1	0	0	1	0	S	S	S	1	4	++ + + +
SBB R	То же, но с заёмом	1	0	0	1	1	S	S	S	1	4	++ + + +
SUB M	Вычитание содержимого ячейки памяти из содержимого аккумулятора	1	0	0	1	0	1	1	0	1	7	++ + + +
SBB M	То же, но с заёмом	1	0	0	1	1	1	1	0	1	7	++ + + +
SUI	Вычитание байта из содержимого аккумулятора	1	1	0	1	0	1	1	0	2	7	++ + + +
SBI	То же, но с учётом заёма	1	1	0	1	1	1	1	0	2	7	++ + + +
INR R	Увеличение содержимого регистра на единицу	0	0	D	D	D	1	0	0	1	5	++ + + -

INR M	Увеличение содержимого ячейки памяти на единицу	0	0	1	1	0	1	0	0	1	10	++ + + -
DCR R	Уменьшение содержимого регистра на единицу	0	0	D	D	D	1	0	1	1	5	++ + + -
DCR M	Уменьшение содержимого ячейки памяти на единицу	0	0	1	1	0	1	0	1	1	10	++ + + -
INX RP	Увеличение содержимого парных регистров B-C, D-E, H- L, SP на единицу	0	0	R	R	0	0	1	1	1	5	-- - - -
DCR RP	Уменьшение содержимого парных регистров B-C, D-E, H- L, SP на единицу	0	0	R	R	0	0	1	1	1	5	-- - - -
ANA R	Поразрядное логическое умножение содержимого регистра и аккумулятора	1	0	1	0	0	S	S	S	1	4	++ 0 + 0
ANA M	Поразрядное логическое умножение содержимого ячейки памяти и аккумулятора	1	0	1	0	0	1	1	0	1	7	++ 0 + 0
ANI	Поразрядное логическое умножение содержимого аккумулятора и байта	1	1	1	0	0	1	1	0	2	7	++ 0 + 0
XRA R	Поразрядное исключающее ИЛИ над содержимым регистра или аккумулятора	1	0	1	0	1	S	S	S	1	4	++ 0 + 0
XRA M	Поразрядное исключающее ИЛИ над содержимым ячейки памяти и аккумулятора	1	0	1	0	1	1	1	0	1	7	++ 0 + 0
XRI	Поразрядное исключающее ИЛИ над содержимым аккумулятора и байтом	1	1	1	0	1	1	1	0	2	7	++ 0 + 0
ORA R	Поразрядное логическое сложение содержимого регистра или аккумулятора	1	0	1	1	0	S	S	S	1	4	++ 0 + 0
ORA M	Поразрядное логическое сложение содержимого ячейки памяти и аккумулятора	1	0	1	1	0	1	1	0	1	7	++ 0 + 0

ORI	Поразрядное логическое сложение содержимого ячейки памяти и аккумулятора	1	0	1	1	0	1	1	0	2	7	++ 0 + 0
CMP R	Сравнение содержимого регистра и аккумулятора	1	0	1	1	1	S	S	S	1	4	++ + + +
CMP M	Сравнение содержимого ячейки памяти и аккумулятора	1	0	1	1	1	1	1	0	1	7	++ + + +
CPI	Сравнение байта с содержимым аккумулятора	1	1	1	1	1	1	1	0	2	7	++ + + +
RLC	Циклический сдвиг содержимого аккумулятора влево	0	0	0	0	0	1	1	1	1	4	- - - - +
RRC	То же, но вправо	0	0	0	0	1	1	1	1	1	4	- - - - +
RAL	Циклический сдвиг содержимого аккумулятора влево через перенос	0	0	0	1	0	1	1	1	1	4	- - - - +
RAR	То же, но вправо	0	0	0	1	1	1	1	1	1	4	- - - - +
CMA	Инвертирование Аккумулятора	0	0	1	0	1	1	1	1	1	4	- - - - -
STC	Установка флага переноса CY в единицу	0	0	1	1	0	1	1	1	1	4	- - - - 1
CMC	Инвертирование флага переноса	0	0	1	1	1	1	1	1	1	4	- - - - C
DAA	Двоично-десятичная коррекция содержимого аккумулятора	0	0	1	0	0	1	1	1	1	4	++ + + +
JMP	Безусловный переход	1	1	0	0	0	0	1	1	3	10	- - - - -
JC	Переход при наличии переноса	1	1	0	1	1	0	1	0	3	10	- - - - -
JNC	Переход при отсутствии переноса	1	1	0	1	0	0	1	0	3	10	- - - - -
JZ	Переход при нуле	1	1	0	0	1	0	1	0	3	10	- - - - -
JNZ	« « отсутствию нуля	1	1	0	0	0	0	1	0	3	10	- - - - -
JP	« « плюсе	1	1	1	1	0	0	1	0	3	10	- - - - -
JM	« « минусе	1	1	1	1	1	0	1	0	3	10	- - - - -
JPE	« « четности	1	1	1	0	1	0	1	0	3	10	- - - - -
JPO	« « нечетности	1	1	1	0	0	0	1	0	3	10	- - - - -
PCHL	Занесение в счетчик команд содержимого регистра H-L	1	1	1	0	1	0	0	1	1	5	- - - - -
CALL	Вызов подпрограммы	1	1	0	0	1	1	0	1	3	17	- - - - -
CC	То же, но при переносе	1	1	0	1	1	1	0	0	3	11/17	- - - - -

CNC	То же, но при отсутствии переноса	1	1	0	0	1	1	0	0	3	11/17	- - - - -
CZ	Вызов подпрограммы при нуле	1	1	0	0	1	1	0	0	3	11/17	- - - - -
CNZ	То же, но при отсутствии нуля	1	1	0	0	0	1	0	0	3	11/17	- - - - -
CP	Вызов подпрограммы при плюсе	1	1	1	1	0	1	0	0	3	11/17	- - - - -
CM	То же, но при минусе	1	1	1	1	1	1	0	0	3	11/17	- - - - -
CPE	Вызов подпрограммы при чётности	1	1	1	0	1	1	0	0	3	11/17	- - - - -
CPO	То же, но при нечётности	1	1	1	0	0	1	0	0	3	11/17	- - - - -
RET	Возврат	1	1	0	0	1	0	0	1	1	10	- - - - -
RC	« « при ереносе	1	1	0	1	1	0	0	0	1	5/11	- - - - -
RNC	« « отсутствии переноса	1	1	0	1	0	0	0	0	1	5/11	- - - - -
RZ	« « нуле	1	1	0	0	1	0	0	0	1	5/11	- - - - -
RNZ	« « отсутствии нуля	1	1	0	0	0	0	0	0	1	5/11	- - - - -
RP	« « плюсе	1	1	1	1	0	0	0	0	1	5/11	- - - - -
RM	« « минусе	1	1	1	1	1	0	0	0	1	5/11	- - - - -
RPE	« « четности	1	1	1	0	1	0	0	0	1	5/11	- - - - -
RPO	« « нечетности	1	1	1	0	0	0	0	0	1	5/11	- - - - -
RST	Повторный запуск	1	1	N	N	N	1	1	1	1	11	- - - - -
IN	Ввод	1	1	0	1	1	0	1	1	2	10	- - - - -
OUT	Вывод	1	1	0	1	0	0	1	1	2	10	- - - - -
EI	Разрешить прерывание	1	1	1	1	1	0	1	1	1	4	- - - - -
DI	Запретить прерывание	1	1	1	1	0	0	1	1	1	4	- - - - -
NOP	Отсутствие операции	0	0	0	0	0	0	0	0	1	4	- - - - -
HLT	Останов	0	1	1	1	0	1	1	0	1	7	- - - - -

### Приложение 7

Команда	Описание операции	В/М	T <sup>80</sup>	T <sup>85</sup>	F
Группа команд передачи данных					
MOV r1, r2	r1 ← r2 { <u>M</u> ove}	1/1	5	4	-
MOV r, M	r ← M(rp H)	1/2	7	7	-
MOV M, r	M(rp H) ← r	1/2	7	7	-
MVI r, d8	r ← d8 { <u>M</u> ove <u>i</u> mmEDIATE}	2/2	7	7	-
MVI M, d8	M(rp H) ← d8	2/3	10	10	-
LXI rp, d16	rp ← d16, rp=B,D,H или SP { <u>L</u> oad <u>i</u> mmEDIATE}	3/3	10	10	-
LDA addr	A ← M(addr) { <u>L</u> oad <u>d</u> irect <u>a</u> ccumulator}	3/4	13	13	-
STA	M(addr) ← A { <u>S</u> ore <u>d</u> irect <u>a</u> ccumulator}	3/4	13	13	-
LHLD addr	L ← M(addr), H ← M(addr+1) { <u>L</u> oad <u>H</u> and <u>L</u> direct}	3/5	16	16	-
SHLD addr	M(addr) ← L, M(addr+1) ← H { <u>S</u> ore <u>H</u> and <u>L</u> direct}	3/5	16	16	-
LDAX rp	A ← M(rp), rp=B или D { <u>L</u> oad <u>a</u> ccumulator <u>i</u> ndirect}	1/2	7	7	-
STAX rp	M(rp) ← A, rp=B или D { <u>S</u> ore <u>a</u> ccumulator <u>i</u> ndirect}	1/2	7	7	-
XCHG	H ↔ D, L ↔ E {Exchange}	1/1	4	4	-

Группа команд арифметических операций					
ADD r	$A \leftarrow A + r$ { <u>A</u> dd}	1/1	4	4	+
ADI d8	$A \leftarrow A + d8$ { <u>A</u> dd <u>i</u> mmEDIATE}	2/2	7	7	+
ADD M	$A \leftarrow A + M(rp\ H)$	1/2	7	7	+
ADC r	$A \leftarrow A + r + CY$ { <u>A</u> dd with <u>c</u> arry}	1/1	4	4	+
ADC M	$A \leftarrow A + M(rp\ H) + CY$	1/2	7	7	+
ACI d8	$A \leftarrow A + d8 + CY$ { <u>A</u> dd with <u>c</u> arry <u>i</u> mmEDIATE}	2/2	7	7	+
SUB r	$A \leftarrow A - r$ { <u>S</u> ubtract }	1/1	4	4	+
SUI d8	$A \leftarrow A - d8$ { <u>S</u> ubtract <u>i</u> mmEDIATE}	2/2	7	7	+
SUB M	$A \leftarrow A - M(rp\ H)$	1/2	7	7	+
SBB r	$A \leftarrow A - r - CY$ { <u>S</u> ubtract with <u>b</u> orrow }	1/1	4	4	+
SBI d8	$A \leftarrow A - d8 - CY$	2/2	7	7	+
SBB M	$A \leftarrow A - M(rp\ H) - CY$ { <u>S</u> ubtract <u>i</u> mmEDIATE}	1/2	7	7	+
INR r	$r \leftarrow r + 1$ { <u>I</u> ncrEment}	1/1	5	5	$\Delta$
INR M	$M(rp\ H) \leftarrow M(rp\ H) + 1$	1/3	10	10	$\Delta$
DCR r	$r \leftarrow r - 1$ { <u>D</u> ecrEment}	1/1	5	5	$\Delta$
DCR M	$M(rp\ H) \leftarrow M(rp\ H) - 1$	1/3	10	10	$\Delta$
INX rp	$rp \leftarrow rp + 1, rp=B, D, H \text{ или } SP$ { <u>I</u> ncrEment rp}	1/1	5	5	–
DCX rp	$rp \leftarrow rp - 1, rp=B, D, H \text{ или } SP$ { <u>D</u> ecrEment rp}	1/1	5	5	–
DAD rp	$rp\ H \leftarrow rp\ H + rp, rp=B, D, H \text{ или } SP$ {Double precision add}	1/1	10	10	$\nabla$
DAA	Десятичная коррекция {Decimal adjust accumulator}	1/1	4	4	+
Группа команд логических операций					
ANA r	$A \leftarrow A \& r$ { <u>A</u> nd register with <u>a</u> ccumulator }	1/1	4	4	*
ANI d8	$A \leftarrow A \& d8$ { <u>A</u> nd <u>i</u> mmEDIATE with accumulator }	2/2	7	7	*
ANA M	$A \leftarrow A \& M(rp\ H)$ { <u>A</u> nd memory with <u>a</u> ccumulator }	1/2	7	7	*
XRA r	$A \leftarrow A \oplus r$ { <u>E</u> xclusive <u>o</u> r register with <u>a</u> ccumulator }	1/1	4	4	#
XRI d8	$A \leftarrow A \oplus d8$ { <u>E</u> xclusive <u>o</u> r <u>i</u> mmEDIATE with accumulator }	2/2	7	7	#
XRA M	$A \leftarrow A \oplus M(rp\ H)$ { <u>E</u> xclusive <u>o</u> r memory with <u>a</u> ccumulator }	1/2	7	7	#

Команда	Описание операции	В/М	T <sup>80</sup>	T <sup>85</sup>	F
ORA r	$A \leftarrow A \vee r$ { <u>O</u> r register with <u>a</u> ccumulator }	1/1	4	4	#
ORI d8	$A \leftarrow A \vee d8$ { <u>O</u> r <u>i</u> mmEDIATE with accumulator }	2/2	7	7	#
ORA M	$A \leftarrow A \vee M(rp\ H)$ { <u>O</u> r memory with <u>a</u> ccumulator }	1/2	7	7	#
CMP r	$A - r$ } Z=1 при A=r, d8, M { <u>C</u> ompare }	1/1	4	4	+
CPI d8	$A - d8$ } CY=1 при A<r, d8, M { <u>C</u> ompare <u>i</u> mmEDIATE }	2/2	7	7	+
CMP M	$A - M(rp\ H)$ }	1/2	7	7	+

RLC		{Rotate accumulator left in carry}	1/1	4	4	▽
RRC		{Rotate accumulator right in carry}	1/1	4	4	▽
RAL		{Rotate accumulator left through carry}	1/1	4	4	▽
RAR		{Rotate accumulator right through carry}	1/1	4	4	▽
CMA	$A \leftarrow \bar{A}$	{Complement accumulator}	1/1	4	4	–
CMC	$CY \leftarrow \bar{CY}$	{Complement carry}	1/1	4	4	▽
STC	$CY \leftarrow 1$	{Set carry}	1/1	4	4	▽
Группа команд передачи управления						
JMP addr	$PC \leftarrow \text{addr}$	{Jump}	3/3	10	10	–
Jcond addr	$PC \leftarrow \text{addr}$	{Jump conditional}	3/3	10	7/10 <sup>1</sup>	–
PCHL	$PC \leftarrow HL$	{Load Program Counter with H and L}	1/1	5	4	–
CALL addr	$SP \leftarrow SP - 1, M(SP) \leftarrow PCh,$	{Call}	3/5	17	18	–
Ccond addr	$SP \leftarrow SP - 1, M(SP) \leftarrow PCl; PC \leftarrow \text{addr}$	{Call conditional}	3/3/5 <sup>2</sup>	11/17 <sup>1</sup>	9/18 <sup>1</sup>	–
RET	$PCl \leftarrow M(SP), SP \leftarrow SP + 1,$	{Return}	1/3	10	10	–
Rcond	$PCh \leftarrow M(SP), SP \leftarrow SP + 1$	{Return conditional}	1/1/3 <sup>2</sup>	5/11 <sup>1</sup>	6/12 <sup>1</sup>	–
RST n	$SP \leftarrow SP - 1, M(SP) \leftarrow PCh,$ $SP \leftarrow SP - 1, M(SP) \leftarrow PCl; PC \leftarrow 8 \times n (n = 0 \dots 7)$	{Restart}	1/3	11	12	–
Группа команд управления стеком, вводом-выводом и состоянием МП						
PUSH rp	$SP \leftarrow SP - 1, M(SP) \leftarrow prh, SP \leftarrow SP - 1, M(SP) \leftarrow rpl$	{Push}	1/3	11	12	–
PUSH PSW	$SP \leftarrow SP - 1, M(SP) \leftarrow A, SP \leftarrow SP - 1, M(SP) \leftarrow F$		1/3	11	12	–
POP rp	$rpl \leftarrow M(SP), SP \leftarrow SP + 1, rph \leftarrow M(SP), SP \leftarrow SP + 1$	{Pop}	1/3	10	10	–
POP PSW	$F \leftarrow M(SP), SP \leftarrow SP + 1, A \leftarrow M(SP), SP \leftarrow SP + 1$		1/3	10	10	+
XTHL	$L \leftrightarrow M(SP), H \leftrightarrow M(SP+1)$	{Exchange Top of Stack with H and L}	1/5	18	16	–
SPHL			1/1	5	6	–
Команда	Описание операции		B/M	T <sup>80</sup>	T <sup>85</sup>	F
IN port	$A \leftarrow \text{ПО}(\text{port})$	{Input}	2/3	10	10	–
OUT port	$\text{ПО}(\text{port}) \leftarrow A$	{Output}	2/3	10	10	–
EI <sup>3</sup>	INTE = 1 после следующей команды	{Enable Interrupts}	1/1	4	4	–
DI <sup>3</sup>	INTE = 0 после данной команды	{Disable Interrupts}	1/1	4	4	–
HLT	Остановка процессора; $PC \leftarrow PC + 1$	{Halt}	1/1	7	5	–
NOP	Пустая операция	{No-operation}	1/1	4	4	–
SIM <sup>4</sup>	Установка маски прерываний	{Set Interrupt Mask}	1/1	–	4	–
RIM <sup>4</sup>	Чтение маски прерываний	{Read Interrupt Mask}	1/1	–	4	–
Примечание: <sup>1</sup> X из X/Y при невыполнении условия cond и Y при невыполнения условия cond; <sup>2</sup> X из Z/X/Y при невыполнении условия cond и Y при невыполнения условия cond; <sup>3</sup> в МП 8085А вместо триггера INTE данные команды управляются флагом (триггером) IE; <sup>4</sup> только для МП 8085А						

### **Библиографический список**

1. Нарышкин А.К. Цифровые устройства и микропроцессоры. М.: Academia, 2006.
2. Пухальский Г.Х. Проектирование микропроцессорных систем. Учебное пособие для вузов. СПб: Политехника, 2001, 544 с.
3. Костров Б.В. Ручкин В.Н. Микропроцессорные системы и микроконтроллеры: Учебное пособие для вузов. М: ТЕХБУК, 2007.
4. Микропроцессоры. / Под ред. Л.Н. Преснухина. М.: Высш. школа, 1986. Т. 1. 347 с.
5. Микропроцессоры: Справочное пособие / Под ред. Ю.А. Овечкина. М.: Судостроение, 1987. 519 с.

# МИКРОПРОЦЕССОРЫ И МИКРО-ЭВМ

## Часть 2

Составители: Н.Г. Крикун  
Р.Н. Гайнуллин  
М.Ю. Валеев

Редактор В.С. Дука  
Корректор Ю.Е. Стрыхарь

Лицензия № 020404 от 6.03.97 г.

Подписано в печать . . . 2007.	Формат 60x84 1/16.
Бумага писчая.	Печать офсетная.
уч.-изд. л.	Тираж экз.
	Заказ . «С» .
	усл. печ. л.

Издательство Казанского государственного  
технологического университета

Офсетная лаборатория Казанского государственного  
технологического университета

420015, Казань, К. Маркса, 68.