

Государственный комитет Российской Федерации по высшей школе
Казанский государственный технологический университет

Микропроцессоры и микро-ЭВМ

Методические указания к лабораторным работам

часть 3

Казань 2004.

Методические указания знакомят студентов с работой микропроцессорного отладочного устройства и пульта управления.

Рассмотрены вопросы практического освоения методики программирования в кодах микропроцессора КР580ИК80А.

Руководство предназначено для студентов специальностей 2102,2202 и 1804, изучающих дисциплины "Микропроцессорные средства и системы" и "Электронно-вычислительные машины".

Лабораторная работа 1

ОРГАНИЗАЦИЯ УСЛОВНЫХ ПЕРЕХОДОВ

ЦЕЛЬ РАБОТЫ: изучение программных способов организации условных переходов в микро-ЭВМ.

1. ОБЩИЕ СВЕДЕНИЯ

Организация условных переходов в микро-ЭВМ осуществляется с помощью регистра признаков МП БИС.

Регистр признаков имеет 5 разрядов, каждый из которых устанавливается по определенному правилу в соответствии с выполнением МП БИС последней команды. Этими разрядами являются:

1. Разряд переполнения CY-CARRY. В него записывается 1, если при выполнении арифметической команды или команды сдвига было переполнение аккумулятора, в противном случае в разряд записывается 0.

2. Разряд знака S-SIGN. В него записывается 1, если при выполнении арифметической или логической команды во всех разрядах числа в аккумуляторе имеются 0, в противном случае в разряд записывается 0.

3. Разряд нулевого результата Z-ZERO. В него записывается 1, если при выполнении арифметической или логической команды во всех разрядах числа в аккумуляторе имеются 0, в противном случае в разряд записывается 0.

4. Дополнительный разряд переполнения AC-AUX.CARRY. В него записывается 1, если при выполнении команд в аккумуляторе возникает единица переноса из третьего разряда числа.

5. Разряд четности P-PARITY. В него записывается 1, если при выполнении команды количество единиц в разрядах аккумулятора будет четным.

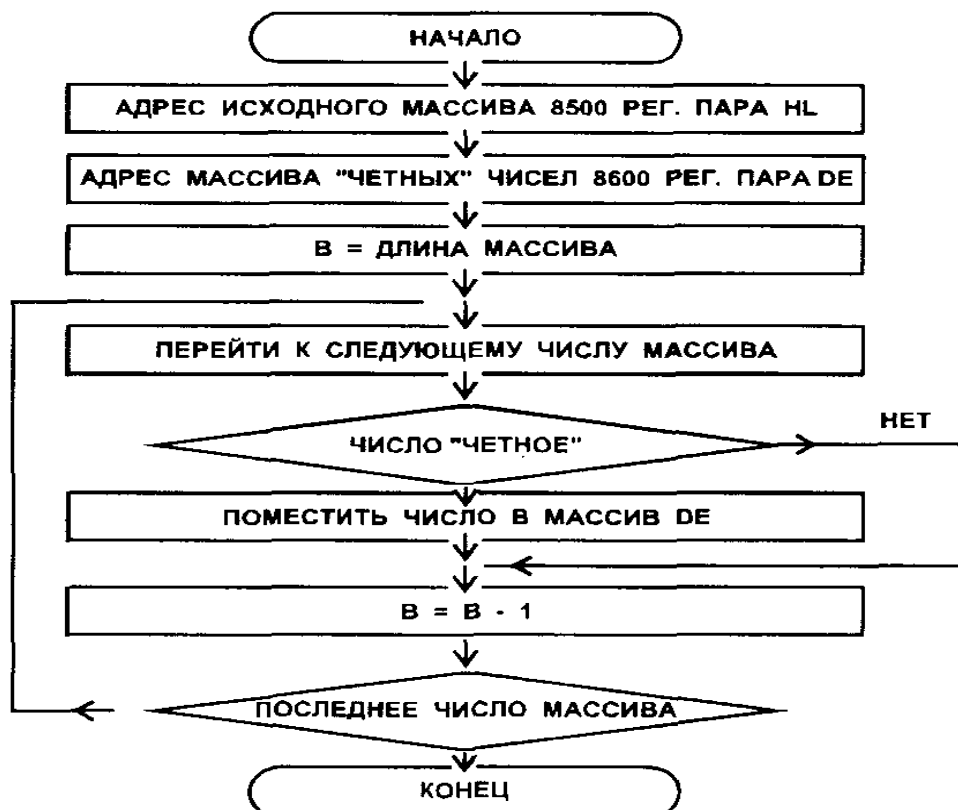
Условные переходы организуются в программах с помощью команд условных переходов. При выполнении этих команд МП БИС проверяет состояние соответствующего разряда регистра состояния. Если при проверке состояния условие не подтверждается, то выполняется следующая по порядку команда программы. Все команды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

JZ			переход если 0;
JNZ			переход если не 0;
JN			переход если отрицательный результат;
JNN			Переход если нет отрицательного результата;
JC			переход если перенос;
JNC			переход если нет переноса;
JPO			переход если число четное;
JPE			переход если число нечетное;

2. ПРОГРАММИРОВАНИЕ С ИСПОЛЬЗОВАНИЕМ УСЛОВНЫХ ПЕРЕХОДОВ

Задание 1: Загрузить программным путем по адресу 8500 массив чисел (Приложение 1).
Сделать выборку числа, которые имеют четное число единиц, записать в массив по адресу 8600.

АЛГОРИТМ



ПОГРАММНАЯ МОДЕЛЬ

HL	85	00	Исходный массив
----	----	----	-----------------

B		Счетчик
---	--	---------

DE	86	00	Массив четных чисел
----	----	----	---------------------

ПРОГРАММА

Таблица 2

Адрес	Машинный код	Метка	Мнемокод	Комментарий
8200	21	M2	LXI H	записать в регистровую пару HL
8201	00		00	адрес исходного массива
8202	85		85	
8203	11		LXI D	записать в регистровую пару DE
8204	00		00	адрес массива четных чисел
8205	86		86	
8206	06		MVI B	загрузить в счетчик B длину массива
8207	nn		nn	длина массива
8208	0E		MVI C	загрузить в регистр C «00»
8209	00		00	
820A	23	MD	INX H	перейти к следующему числу массива HL
820B	7E		MOV A,M	поместить число из массива в Акк
820C	81		ADD C	Сложение Акк+рег.C
820D	EA		JPE, MD	если число четное, идти на MD
820E	13		13	
820F	82		82	
8210	C3		JMP,M1	иначе идти на M1
8211	15		15	
8212	82		82	
8213	12		STAX D	записать число в рег. пару DE
8214	13	M1	INX D	перейти к следующему числу массива DE
8215	05		DCR B	уменьшить счетчик на 1
8216	C2		JNZ, M2	если не 0, идти на M2
8217	0A		0A	
8218	82		82	
8219	E7		RST 32	Останов

3. ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

3.1. Изучить материалы, изложенные в п. 1 и лекции по теме "Команды перехода".

3.2. Получить индивидуальное задание на выполнение программирования.

3.3. Выполнить работу по программированию в соответствии с полученным заданием с практической отработкой программы на устройстве "Э580".

3.4. Оформить отчет.

4. СОДЕРЖАНИЕ ОТЧЕТА

4.1. Задание на выполнение лабораторной работы.

4.2. Программная модель (использование РОНов памяти при решении задачи, алгоритм программа).

4.3. Исходные данные, используемые при решении программы
полученные результаты.

ВОПРОСЫ ДЛЯ САМОПРОВЕРКИ

1. Какие бывают команды перехода?

2. Перечислите команды условных переходов.

3. С помощью каких регистров признаков осуществляется каждая из них

4. Каков формат команд перехода и к какому способу адресации они относятся?

5. По каким условиям записывается 1 в каждый из разрядов регистра состояния МП БИС?

Лабораторная работа 2 МАСКИРОВАНИЕ ДАННЫХ

ЦЕЛЬ РАБОТЫ: изучение программных способов маскирования данных.

1. ОБЩИЕ СВЕДЕНИЯ

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

1. Логического умножения (AND r) числа в аккумуляторе и маски, которое очищает разряд числа, если в соответствующем разряде маски, будет записан 0, и не изменяет его, если в разряде маски записана 1;

2. Логического сложения (OR r) числа в аккумуляторе и маски, которое устанавливает разряд числа в 1, если в таком же разряде маски будет записана 1, и не изменяет его, если в этом разряде записан 0; 3. Логического "исключающего ИЛИ" (XOR r) числа в аккумуляторе и маски, которое инвертирует содержание разряда числа, если в соответствующем разряде маски записана 1, и не изменяет его, если в этом разряде записан 0;

При выполнении всех логических команд задействуются разряды Z и N регистра признаков, разряд C регистра переноса не работает.

Наиболее часто для целей маскирования используется операция "исключающее ИЛИ". При этом бит проверяемых исходных данных обращается в "логический 0", если он совпадает по значению с соответствующим битом маски.

После применения операции XOR можно проверить значение каждого бита результата

путем сдвига проверяемого бита и последующего выяснения значения разряда в регистре признаков. Другое назначение команд XOR в очистке (сбросе) аккумулятора либо регистров. К слову данных "прибавляется" копия этого слова, в результате в соответствующем регистре образуются двоичные 0.

2.ПРОГРАММИРОВАНИЕ С ИСПОЛЬЗОВАНИЕМ МАСКИРОВАНИЯ.

ЗАДАНИЕ 1: ЗАГРУЗИТЬ ПРОГРАММНЫМ ПУТЕМ ПО АДРЕСУ 872D ЧИСЛО 0A, ПО АДРЕСУ 80A4, ЧИСЛО 21. НАЙТИ СЛОВО МАСКИ ТАКОЕ, ЧТОБЫ СУММА ЧИСЕЛ 0A И 21 БЫЛА РАВНА 1F. ПОМЕСТИТЬ ЕГО В ПАМЯТЬ ПОАДРЕСУ 852E, ПРОИЗВЕСТИ ОПЕРАЦИЮ МАСКИРОВАНИЯ КОМАНДОЙ "ИСКЛЮЧАЮЩЕЕ ИЛИ", ПРОВЕРИТЬ РЕЗУЛЬТАТ В АККУМУЛЯТОРЕ.

$$\begin{array}{r}
 \text{РЕШЕНИЕ: } 0A_{16} = 10_{10} = 1010_2 \\
 + 21_{16} = 33_{10} = 100001_2 \\
 \hline
 2B_{16} = 43_{10} = 101011_2 \\
 \text{маска } 34_{16} = 52_{10} = 110100_2 \\
 \hline
 1F_{16} = 31_{10} = 011111_2
 \end{array}$$

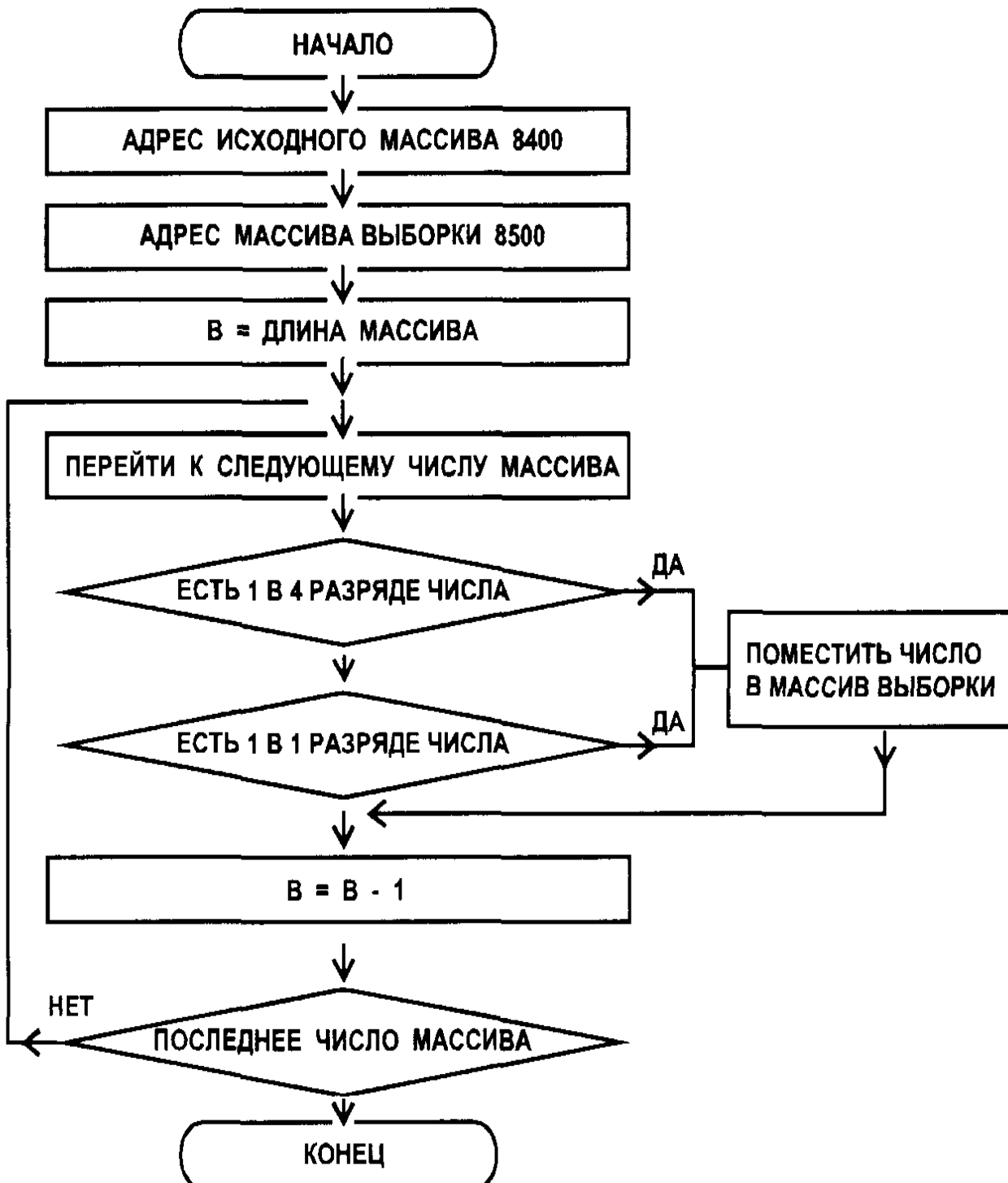
Программа 1.

Таблица 1

Адрес	Машинный код	Метка	Мнемокод	Комментарий
8200	21		LXI H	записать в рег. пару HL
8201	2D		2D	адрес первого слагаемого
8202	87		87	
8203	36		MVI M	загрузить в память первое
8204	0A		0A	слагаемое
8205	7E		MOV A,M	поместить в АКМ первое
8206	21		LXI H	записать в рег. пару HL
8207	A4		A4	адрес второго слагаемого
8208	80		80	
8209	36		MVI M	загрузить в память второе
820A	21		21	слагаемое
820B	86		ADD M	произвести сложение чисел
820C	21		LXI H	записать в рег. пару HL
820D	2E		2E	адрес " маски"
820E	85		85	
820F	36		MVI M	загрузить в память слово "маски"
8210	34		34	
8211	AE		XRA M	произвести операцию "искл.ИЛИ"
8212	00		NOP	конец программы

ЗАДАНИЕ 2: ЗАГРУЗИТЬ ПО АДРЕСУ 8400 МАССИВ ЧИСЕЛ (Приложение 2). СДЕЛАТЬ ВЫБОРКУ: ЧИСЛА, КОТОРЫЕ СОДЕРЖАТ В 5 И 1 РАЗРЯДАХ ЕДИНИЦУ ПОМЕСТИТЬ В МАССИВ ПО АДРЕСУ 8500.

АЛГОРИТМ



ПРОГРАММА 2

Таблица 2

Адрес	Машинный код	Метка	Мнемокод	Комментарий
8200	21	M2	LXI H	записать в рег.пару HL
8201	00		00	адрес исходного массива
8202	84		84	
8203	11		LXI D	записать в рег .пару DE
8204	00		00	адрес массива выборки
8205	85		85	
8206	46		MOV B,M	загрузить в счетчик B длину массива
8207	23		INX H	перейти к следующему числу массива HL
8208	7E		MOV A, M	поместить число из массива в
8209	0E		MVI C	загрузить в регистр C число
820A	10		10	
820B	A1		ANA C	произвести "лог. И" АКМ и
820C	C2		JNZ, M1	если 0, идти на M1
820D	1B		1B	
820E	82		82	
820F	7E		MOV A, M	поместить число из массива в
8210	0E	M3	MVI C	загрузить в регистр C число
8211	02		02	
8212	A1		ANA C	произвести "лог. И" АКМ и
8213	C2		JNZ, M1	если не 0, идти на M1
8214	1B		1B	
8215	82		82	
8216	05		DCR B	уменьшить счетчик на 1
8217	C2		JNZ, M2	если не 0, идти на M2
8218	07		07	
8219	82		82	
821A	E7	M1	RST32	конец
821B	7E		MOV A, M	поместить число из массива в
821C	12		STAX D	записать число в массив DE
821D	13		INX D	перейти к следующему числу массива DE
821 E	C3		JMP, M3	переход на M3
821F	16		16	
8220	82		82	
8221	00		NOP	

3. ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

- 3.1. Изучить материалы, изложенные в п. 1 и лекции по теме "Логические команды".
- 3.2. Получить индивидуальное задание на выполнение программирования.
- 3.3. Выполнить работу по программированию в соответствии с полученным заданием с практической отработкой программы на устройстве "Э-580".
- 3.4. Оформить отчет.

4. СОДЕРЖАНИЕ ОТЧЕТА

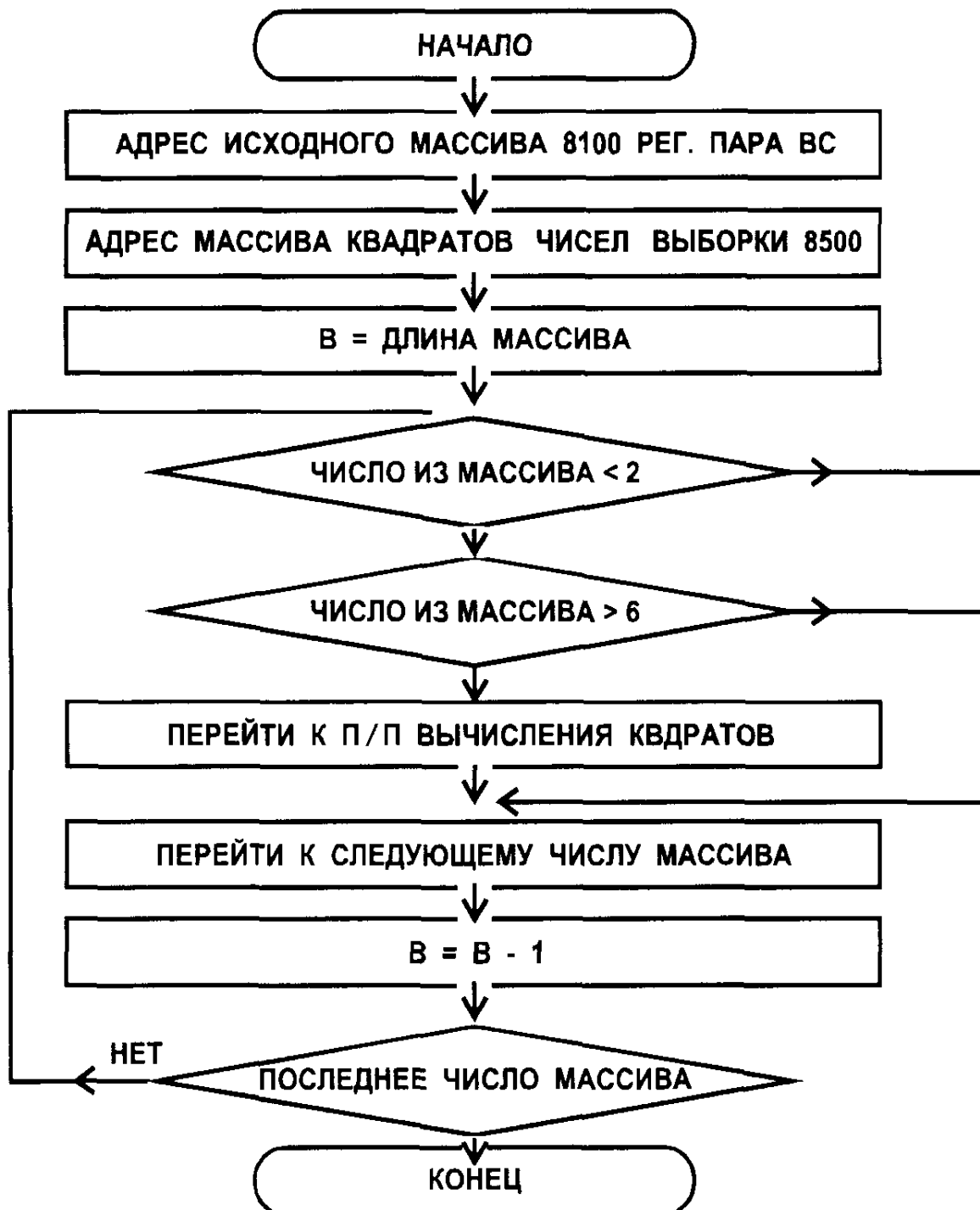
- 4.1 Задание на выполнение лабораторной работы.
- 4.2 Программная модель (использование РОНов и памяти при решении задачи, алгоритм и программа).
- 4.3 Исходные данные, используемые при решении программы и полученные результаты.

Лабораторная работа 3

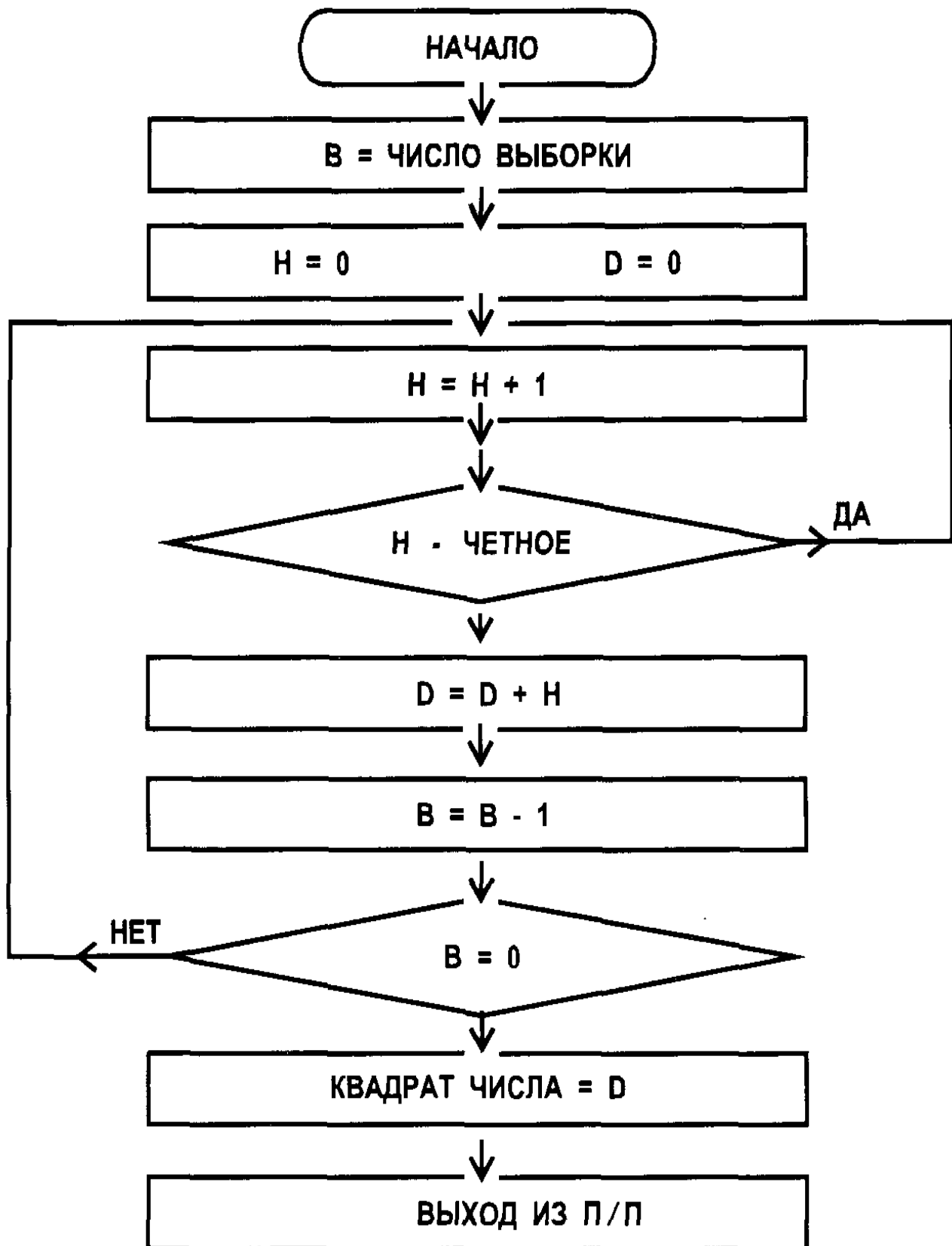
ВОЗВЕДЕНИЕ В КВАДРАТ

ЗАДАНИЕ: ЗАГРУЗИТЬ ПРОГРАММНЫМ ПУТЕМ ПО АДРЕСУ 8100 МАССИВ ЧИСЕЛ (Приложение 3). СДЕЛАТЬ ВЫБОРКУ ЧИСЕЛ ОТ 2 ДО 6, ВОЗВЕСТИ ИХ В КВАДРАТ И ПОМЕСТИТЬ ПО АДРЕСУ 8500.

АЛГОРИТМ



АЛГОРИТМ ПРОГРАММЫ ВЫЧИСЛЕНИЯ КВАДРАТОВ ЧИСЕЛ:



Адрес	Машинный код	Метка	Мнемокод	Комментарий
8200	01	M1	LXI B	записать в рег. пару BC
8201	00		00	адрес исходного массива
8202	81		81	
8203	11		LXI D	записать в рег. пару DE
8204	00		00	адрес массива квадратов выборки
8205	85		85	
8206	26		MVI H	загрузить в счетчик H длину
8207	05		05	массива
8208	0A		LDAX B	поместить число из массива в АКМ
8209	FE		CPI	сравнить его с 02
820A	02		02	
820B	DA		JC, M2	если меньше 2 , идти на M2
820C	17		17	
820D	82		82	
820E	FE		CPI	сравнить его с 06
820F	06		07	
8210	D2		JNC, M2	если больше 6 , идти на M2
8211	17	M2	17	
8212	82		82	
8213	CD		CALL	вызов подпрограммы
8214	00		00	
8215	86		86	
8216	13		INX D	перейти к следующему числу массива DE
8217	03		INX B	перейти к следующему числу массива BC
8218	25		DCR H	уменьшить счетчик на 1
8219	C2		JN2, M1	если не 0, идти на M1
851A	08		08	
821B	82		82	
821C	00		NOP	

ПОДПРОГРАММА ВЫЧИСЛЕНИЯ КВАДРАТОВ

Таблица 3

Адрес	Машинный код	Метка	Мнемокод	Комментарии
8600	CS	M3	PUSH B	сохранить содержимое рег. BC в стеке
8601	47		MOV B.A	записать число выборки в рег. B
8602	E5		PUSH H	сохранить содержимое рег. HL в стеке
8603	26		MVI H	записать в рег. H 00
8604	00		00	
8605	97		SUB A	обнулить АКМ
8606	D5		PUSH D	сохранить содержимое рег. DE в стеке
8607	16		MVI D	записать в рег. D 00
8608	00		00	
8609	24		INR H	$H = H + 1$
860A	7C		MOV A.H	переслать данные из рег. H в АКМ
860B	1F		RAR	осуществить сдвиг вправо
860C	D2		JNC.M3	если нет переноса, идти на M3
860D	09		09	
860E	86		86	
860F	17		RAL	осуществить сдвиг влево
8610	FE		CPI	проверить на переполнение
8611	FF		FF	
8612	82		ADD D	$A = A + D$
8613	57		MOV D.A	записать число из АКМ в рег. D
8614	05		DCR B	уменьшить счетчик на 1
8615	C2		JNZ, M3	если не 0. идти на M3
8616	09		09	
8617	86		86	
8618	D1		POP D	восстановить содержимое рег. DE
8619	12		STAX D	записать в рег. пару DE квадрат числа
861A	E1		POP H	восстановить содержимое рег. HL
861B	C1		POP B	восстановить содержимое рег. BC
861C	C9		RET	возврат из подпрограммы

3. ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

3.1. Изучить материалы, изложенные в пунктах 1,2 и лекции по теме "Стеки подпрограмма".

3.2. Ввести в УОУ "Э580" программу п.2.

3.3. Выполнить программу по командам, используя режим "Отладки". После каждой команды проверить содержимое всех регистров МП БИС.

3.4. Заменить в программе команду POP PSW на команду NOP и проследить, как будет выполняться программа.

3.5. Получить индивидуальное задание на выполнение программирования с использованием стека.

3.6. Выполнить работу по программированию в соответствии с полученным заданием и практической отработкой программы на УОУ "Э580".

3.7. Оформить отчет.

4. СОДЕРЖАНИЕ ОТЧЕТА:

4.1. Задание на выполнение лабораторной работы.

4.2. Программная модель (использование РОНов и памяти при решении задачи, алгоритм и программа).

4.3. Исходные данные, используемые при решении программы и полученные результаты.

ВОПРОСЫ ДЛЯ САМОПРОВЕРКИ

1. В какой последовательности записывается и считывается из стека содержимое аккумулятора и регистра признаков МП БИС при выполнении команды POP PSW и PUSH PSW?

2. Укажите порядок выполнения УОУ "Э580" команды RET.

3. Сравните процесс выполнения УОУ "Э580" команд CALL и RST.

4. В какой последовательности сохраняется и извлекается содержимое регистров МП БИС в подпрограмме?

5. Как будет выполняться программа, если вместо команды POP в ней будет записана команда NOP?

Лабораторная работа 4

ПРОГРАММНАЯ РЕАЛИЗАЦИЯ ДЕЛЕНИЯ ЦЕЛЫХ ДВОИЧНЫХ ЧИСЕЛ

ЦЕЛЬ РАБОТЫ: Освоение способов программной реализации деления целых двоичных чисел.

1. ОБЩИЕ СВЕДЕНИЯ

Деление двоичных чисел производится так же, как и деление десятичных чисел. Однако деление двоичных чисел осуществляется проще, так как использование только двух цифр (0 и 1) исключает в каждом цикле деления необходимость определения числа делителей, содержащихся в текущей части делимого (остатке). Для определения очередной цифры частного достаточно только сравнить текущую часть делимого с делителем.

Пусть требуется разделить однобайтное число без знака $X=11010011$ на однобайтное число без знака $Y=00001011$. Вручную деление осуществляется следующим образом:

$$\begin{array}{r|l}
 11010011 & 1011 \\
 \underline{1011} & 10011 \\
 10001 & \\
 \underline{1011} & \\
 1101 & \\
 \underline{1011} & \\
 10 &
 \end{array}$$

Рис 1.

Выделяется часть делимого, начиная со старшего разряда таким образом, чтобы она была не меньше делителя и из нее вычитался делитель, а в частное записывается 1. К остатку, полученному из вычитания, сносится следующая цифра делимого и производится сравнение полученного таким образом остатка с делителем. Если остаток не меньше делителя, то в следующий более младший разряд частного записывается 1. Если остаток меньше делителя - записывается 0. Затем к остатку добавляется следующая цифра делителя и снова производится сравнение остатка с делителем для определения следующей цифры частного. Аналогично производится операции, до тех пор, когда используются все цифры делимого. При делении заданных чисел получается частное $Z = 10011$ и остаток $f = 10$.

Из приведенного примера видно, что при ручном делении часть операций фиксируется на бумаге, то есть, выполняется как бы явно (вычитание из текущего остатка делителя, если остаток не меньше делителя), а часть операций выполняется в уме, то есть как бы неявно (выделение текущего остатка с делителем).

При организации автоматического процесса деления все операции должны выполняться явно. Процесс деления тех же чисел с явным выполнением всех операций приведен на рис. 2.

Здесь после образования каждого текущего остатка, полученного добавлением очередной цифры делимого к остатку, производится вычитание из текущего остатка делителя. Если результат вычитания не меньше 0, то в очередной разряд частного записывается 1. Если результат вычитания - меньше 0, то в очередной разряд частного записывается 0.и производится восстановление остатка путем прибавления к отрицательному остатку делителя. Затем к остатку добавляется очередная цифра делимого, и операции определения очередных цифр частного повторяются аналогично.

11010011	1011	
1011	00010011	
1010		0
1011		
11		
1011		0
1000		
1011		
110		
1011		0
101		
1011		
1101		
1011		1
100		
1011		0
111		
1011		
1000		
1011		0
11		
1011		
10001		
1011		1
1101		
1011		1
10		

Рис.2

2. ПРОГРАММИРОВАНИЕ ДЕЛЕНИЯ ЦЕЛЫХ ДВОИЧНЫХ ЧИСЕЛ

Рассмотрим 2 способа реализации программирования деления с явным счетчиком и неявным счетчиком. (Приложение 4)

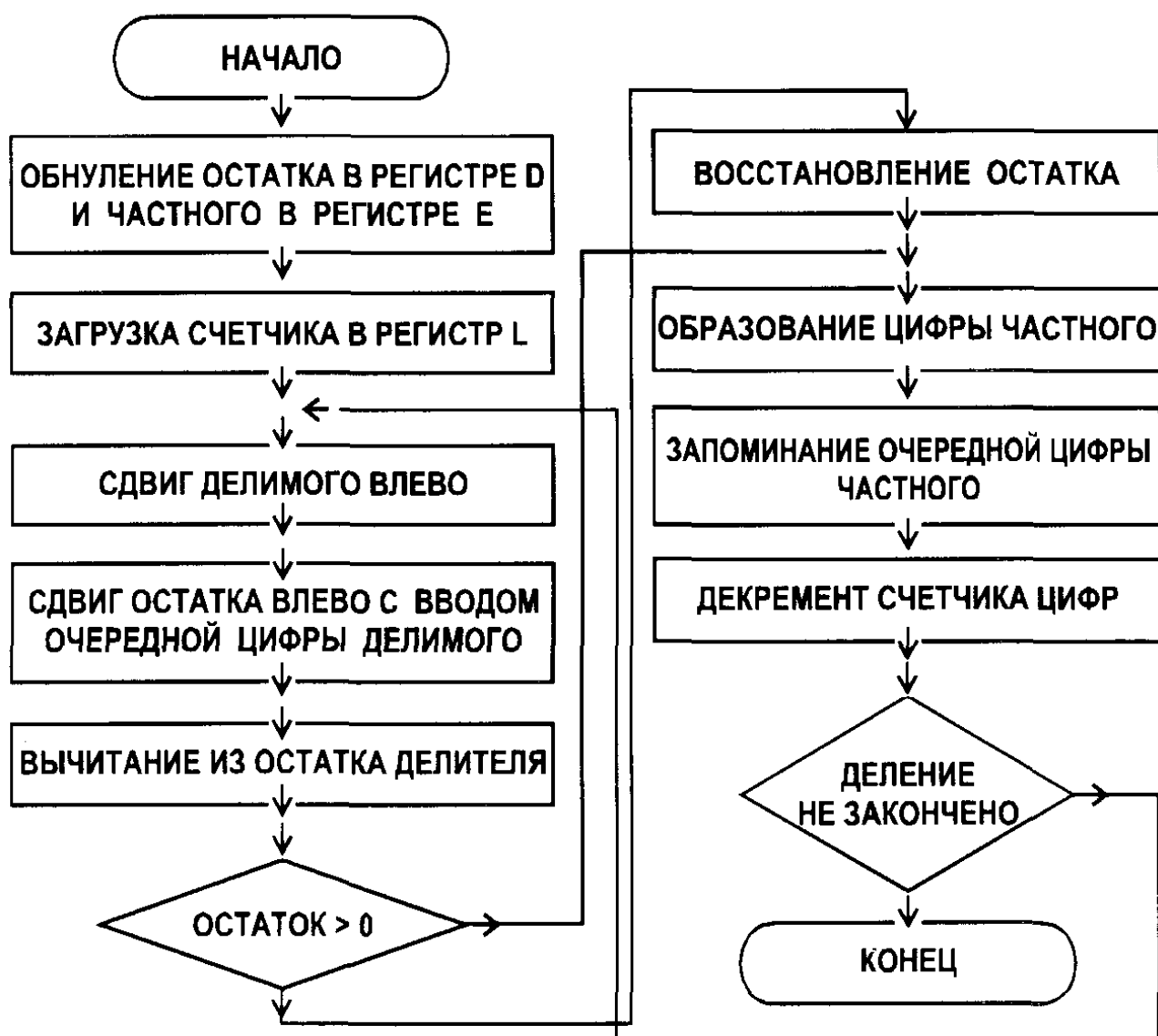
2.1. С явным счетчиком.

В делимое Е частное

С делитель L счетчик

D остаток

АЛГОРИТМ

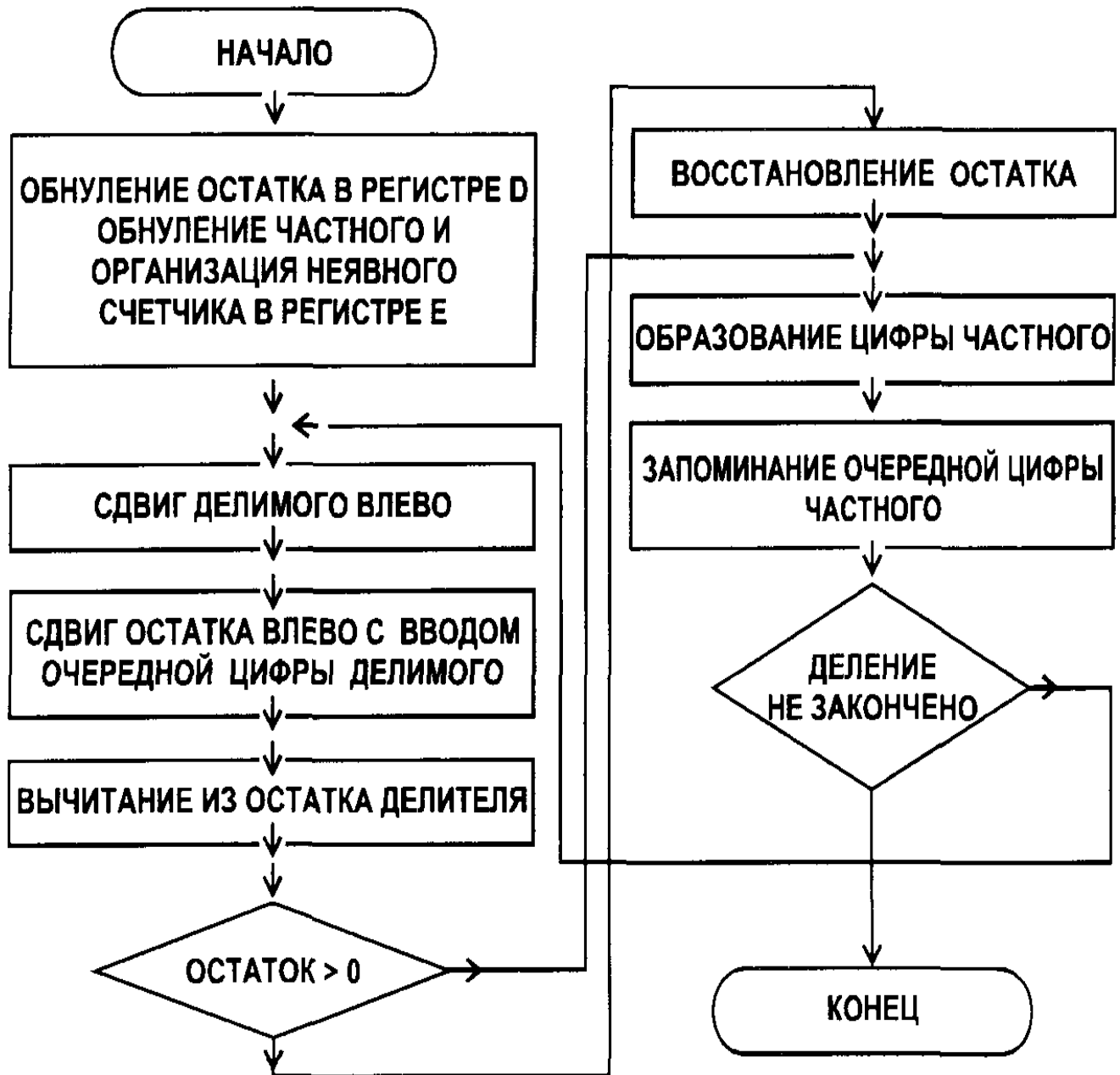


Адрес	Машинный код	Метка	Мнемокод	Комментарий
8200	11	MO	LXID	обнуление остатка и частного
8201	00		00	
8202	00		00	
8203	2E		MVIL	загрузка счетчика цифр частного
8204	08		08	
8205	78		MOVA.B	
8206	17		RAL	сдвиг делимого влево
8207	47		MOV B,A	
8208	7A		MOV AD	
8209	17		RAL	сдвиг остатка влево
820A	91	M1	SUBC	вычитание из остатка делителя
820B	D2		JNC.M1	если остаток положительный,
820C	OF		OF	идти на M1
820D	82		82	
820E	81		ADD C	восстановление остатка
820F	57		MOV D,A	запоминание остатка
8210	3F		CMC	образование цифры частного
8211	7B		MOVA.E	запись очередной цифры частного
8212	17		RAL	
8213	5F		MOVE.A	
8214	2D		DCRL	уменьшить счетчик на 1
8215	C2		JNZ.MO	если не 0, идти на MO
8216	05		05	
8217	82		82	
8218	00		NOP	

2.2. С неявным счетчиком

В делимоеС делительD остатокE частное и неявный счетчик

АЛГОРИТМ



Адрес	Машинный код	Метка	Мнемокод	Комментарий
8200	11	MO	LXID	обнуление остатка и частного и организация неявного счетчика
8201	01		01	
8202	00		00	сдвиг делимого влево
8203	78		MOVA, B	
8204	17		RAL	
8205	47		MOV B, A	
8206	7A		MOV A, D	сдвиг остатка влево
8207	17		RAL	
8208	91		SUBC	вычитание из остатка делителя
8209	D2		JNC,M1	если остаток положительный, идти на M1
820A	00	M1	OD	восстановление остатка
820B	82		82	
820C	81		ADD C	запоминание остатка
820D	57		MOV D,A	образование цифры частного
820E	3F		CMC	запись очередной цифры частного
820F	7B		MOVA, E	сдвиг частного влево
8210	17		RAL	запоминание частного
8211	5F		MOVE.A	если не последний бит, идти на MO
8212	D2		JNC.MO	
8213	03		03	
8214	82		82	
8215	00		NOP	

3. ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

3.1. Изучить материалы, изложенные в пунктах 1,2.

3.2. Получить задание на выполнение программирования.

3.3. Выполнить работу по программированию в соответствии с полученным заданием с практической отработкой программы на устройстве "Э-580".

3.4. Оформить отчет.

4. СОДЕРЖАНИЕ ОТЧЕТА

4.1. Задание на выполнение лабораторной работы.

4.2. Программная модель (использование РОНов и памяти при решении задачи, алгоритм и программа).

4.3. Исходные данные, используемые при решении программы и полученные результаты.

ВОПРОСЫ ДЛЯ САМОПРОВЕРКИ

1. На чем основан алгоритм программы деления чисел?

2. Чем отличается программы 1 и 2?
3. Каким образом организуется образование цифры частного в программах 1 и 2?
4. Объясните организацию неявного счетчика в программе 2?

Лабораторная работа 5

ПРОГРАММНАЯ РЕАЛИЗАЦИЯ УМНОЖЕНИЯ ДВОИЧНЫХ ЧИСЕЛ.

ЦЕЛЬ РАБОТЫ: Освоение организации программы двоичных чисел. (Приложение 5)

1. ОБЩИЕ СВЕДЕНИЯ

Существует несколько алгоритмов умножения чисел. При неявном алгоритме умножение можно заменить многократным сложением, на пример $14 \cdot 3 = 14 + 14 + 14$. Существенный недостаток этого способа - значительная длительность процесса вычисления. При втором алгоритме умножение осуществляется в столбец. Этот алгоритм применим для умножения двоичных чисел. Пусть требуется умножить число 0110 (6) на число 0011 (3). Умножение в столбец производится аналогично умножению десятичных чисел:

$$\begin{array}{r}
 0110 = 6 \\
 0011 = 3 \\
 \hline
 0110 \\
 + 0110 \\
 0000 \\
 0000 \\
 \hline
 00010010 = 18
 \end{array}$$

2. ПРОГРАММИРОВАНИЕ УМНОЖЕНИЯ ЦЕЛЫХ ДВОИЧНЫХ ЧИСЕЛ.

При вычислении результата по второму алгоритму необходимо осуществить многократное суммирование со сдвигом влево множимого при одновременной проверке содержимого разрядов множителя, начиная со стороны его младшего разряда. При этом если в очередном разряде множителя записана 1, то множимое добавляется 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.

Входные параметры:

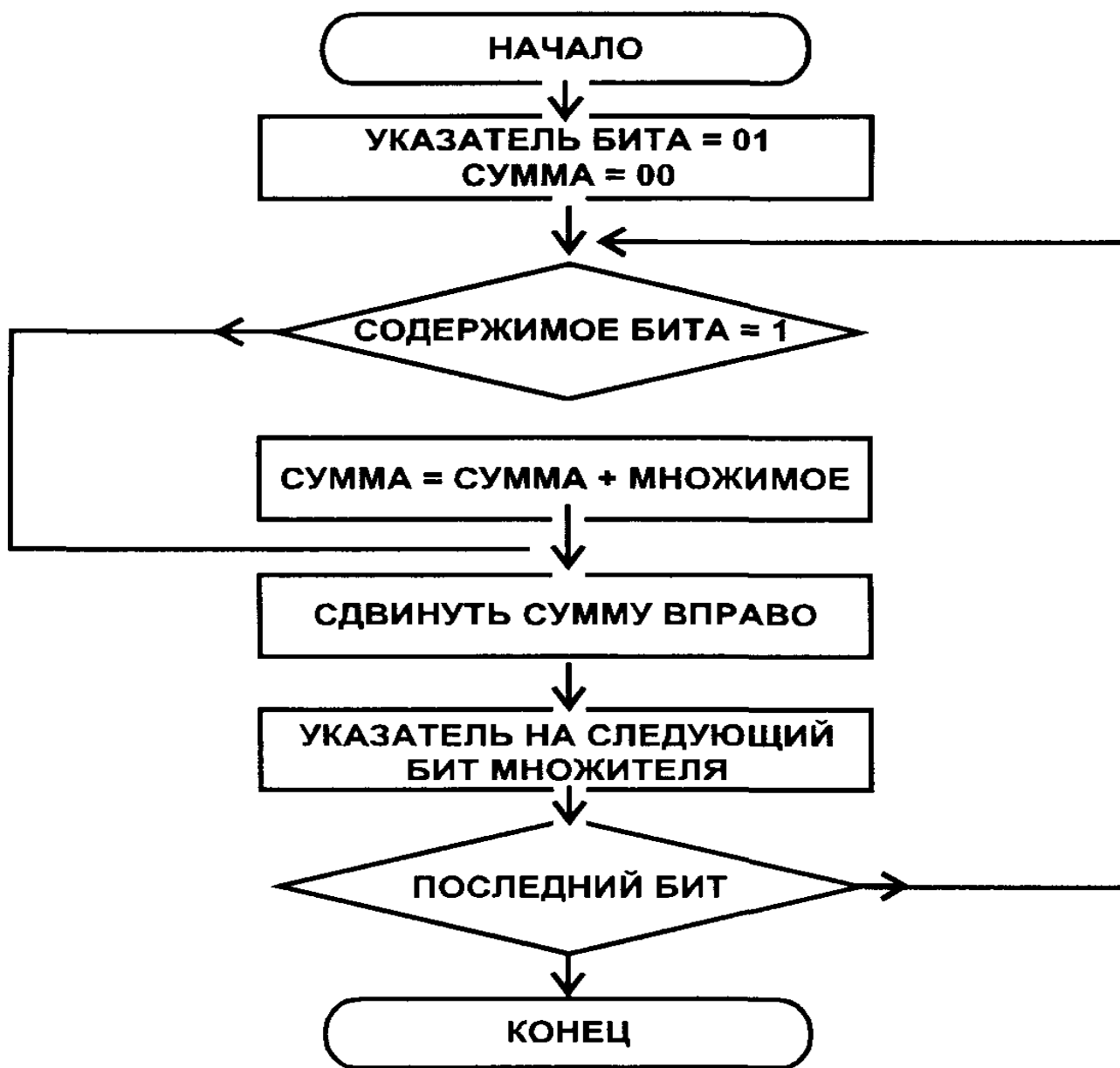
D множимое

E множитель

D остаток

BC результат перемножения

АЛГОРИТМ



ПРОГРАММА

Адрес	Машинный код	Метка	Мнемокод	Комментарий
8200	01	M1	LXIB	очистить содержимое регистров ВС
8201	00		00	
8202	00		00	
8203	3E		MVIA	загрузить в аккумулятор указатель разряда
8204	01		01	
8205	A7		ANAA	очистить флаг C
8206	F5		PUSH	сохранить указатель разряда в стеке
8207	A3		ANAE	проверить содержимое очередного разряда множителя
8208	78		MOV A, B	загрузить в аккумулятор старший байт суммы
8209	CA		JZ.M2	если в очередном разряде записан 0, идти на M2
820A	0D	M2	0D	
820B	82		82	
820C	82		ADDD	прибавить множимое к сумме
820D	1F		RAR	сдвинуть сумму вправо
820 E	47		MOV B,A	сохранить содержимое аккумулятора в регистре B
820F	79		МСЛ/ДС	загрузить в аккумулятор младший байт суммы
8210	1F		RAR	сдвинуть число в аккумуляторе вправо
8211	4F		MOV C,A	сохранить содержимое аккумулятора в регистре C
8212	F1		POP PSW	подучить из стека указатель разряда
8213	17		PAL	указатель на следующий разряд
8214	D2		JNC M1	если разряд не последний, переход M1
8215	06		06	
8216	82		82	
8217	00		NOP	

3. ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

3.1. Изучить материалы, изложенные в пунктах 1,2.

3.2. Получить задание на выполнение программирования.

3.3. Выполнить работу по программированию в соответствии с полученным заданием с практической отработкой программы на устройстве "Э-580".

3.4. Оформить отчет.

4. СОДЕРЖАНИЕ ОТЧЕТА

4.1. Задание на выполнение лабораторной работы

4.2. Программная модель (использование РОНов и памяти при решении задачи, алгоритм и программа).

4.3. Исходные данные, используемые при решении программы и полученные результаты.

ВОПРОСЫ ДЛЯ САМОПРОВЕРКИ

1. Приведите примеры существующих алгоритмов умножения

2. На чем основан алгоритм работы программы умножения'

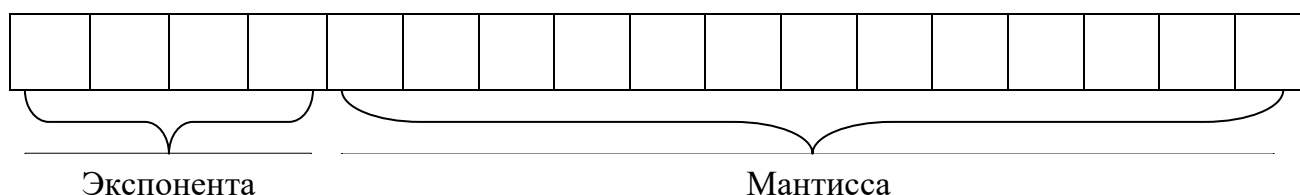
3. Для чего в программе используется команда ANA A?

4. Что такое "сдвиг влево", "сдвиг вправо"?

Лабораторная работа №6

СЛОЖЕНИЕ ЧИСЕЛ С ПЛАВАЮЩЕЙ ЗАПЯТОЙ

Числа с плавающей запятой состоят из двух основных частей: экспоненты и мантиссы. Экспонента означает степень числа, мантисса – значащие цифры. Для работы будем использовать следующее представление чисел с плавающей запятой:



Для перевода числа с плавающей запятой из десятичной системы счисления в шестнадцатеричную используем следующий метод: сначала найдем шестнадцатеричное представление экспонента, затем – мантиссы. Старший разряд экспоненты отвечает за ее знак. Таким образом, мы можем проводить вычисления с числами при основании системы счисления от -8 до 7.

Пусть, например, нам нужно перевести число 3,15 в шестнадцатеричную систему счисления. Производим следующие действия:

$$3_{10} = 11_2; \quad \begin{array}{ll} 0,15 * 2 = 0,3; & 0 \\ 0,3 * 2 = 0,6; & 0 \\ 0,6 * 2 = 1,2; & 1 \\ 0,2 * 2 = 0,4; & 0 \\ 0,4 * 2 = 0,8; & 0 \\ 0,8 * 2 = 1,6; & 1 \\ 0,6 * 2 = 1,2; & 1 \\ 0,2 * 2 = 0,4; & 0 \\ 0,4 * 2 = 0,8; & 0 \\ 0,8 * 2 = 1,6; & 1 \\ 0,6 * 2 = 1,2. & 1 \end{array}$$

$$3,15_{10} = 11,0010011001_2 = 1,10010011001 * 2^1.$$

Отбрасываем целую часть мантиисы (1), экспонента равна $1_{10} = 0001_2$, получаем двоичное представление десятичного числа 3,15:

$$000110010011001_2 = 1933_{16}.$$

Для сложения двух чисел с плавающей запятой необходимо сравнить их экспоненты, выяснив при этом, насколько одно число больше другого. Затем с помощью операции сдвига оба числа приводятся к одинаковой экспоненте, их мантиисы складываются, образуя при этом мантиису суммы. Искомая сумма получается при объединении экспоненты и мантиисы.

Предположим, нам необходимо сложить два числа: 3,15 и 0,72.

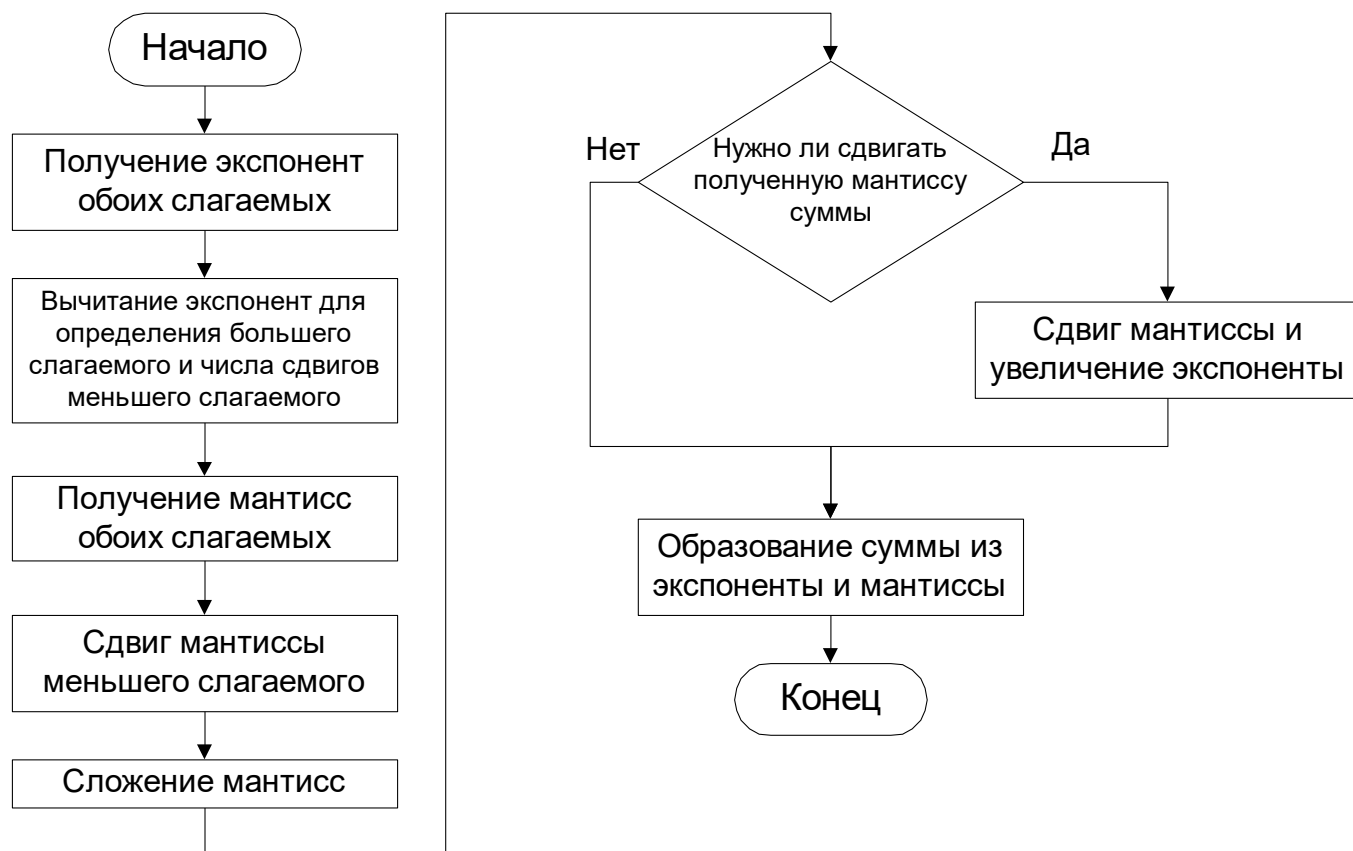
$$\begin{array}{ll} 3,15_{10} = 1933_{16}, & \\ 3,15 + 0,72 = 3,87; & 0,72_{10} = F70A_{16}, \\ 3,87_{10} = 1EF5_{16}. & \end{array}$$

Найдем экспоненты наших слагаемых. Это $1_{16} = 0001_2$ и $F_{16} = 1111_2$. Так как старший бит указывает на знак числа, то $0001 > 1111$ (то есть $1 > -1$). Их разность и будет признаком того, во сколько раз второе слагаемое меньше первого. В нашем случае разность равна двум; это означает, что мантиису второго слагаемого нужно сдвинуть вправо на две позиции, чтобы получить у обоих слагаемых одинаковые экспоненты.

Мантииса первого слагаемого: $933_{16} = 100100110011_2$. Таким образом, мы имеем число $1,100100110011_2$. Мантииса второго слагаемого: $70A_{16} = 011100001010_2$. После двукратного сдвига получаем число $0,010111000010_2$. Складывая оба слагаемых, получаем: $1,100100110011 + 0,010111000010 = 1,111011110101$. Мантиисы суммы: $111011110101_2 = EF5_{16}$.

Так как экспонента суммы равна 1, получаем ответ: $1EF5$.

Схема алгоритма:



Программная модель

DE

 Первое слагаемое

HL

 Второе слагаемое и сумма

Программа:

Адрес	Код	Метка	Команда	Описание
8200	7A		MOV A,D	Копирование данных из регистра D в регистр A
8201	CD		CALL	Вызов подпрограммы с начальным адресом 8300
8202	00		00	
8203	83		83	
8204	47		MOV B,A	Копирование данных из регистра A в регистр B
8205	7C		MOV A,H	Копирование данных из регистра H в регистр A
8206	CD		CALL	Вызов подпрограммы с начальным адресом 8300
8207	00		00	
8208	83		83	
8209	90		SUB B	Вычитание содержимого регистра B из содержимого аккумулятора
820A	4F		MOV C,A	Копирование данных из регистра A в регистр C
820B	E6		ANI	Логическое умножение содержимого аккумулятора с числом 80
820C	80		80	
820D	CA		JZ, M1	Если 0, перейти на M1
820E	19		19	
820F	82		82	
8210	79		MOV A,C	Копирование данных из регистра C в регистр A
8211	EB		XCHG	Обмен данными между регистровыми парами DE и HL
8212	3D		DCR A	Уменьшение содержимого аккумулятора на единицу
8213	2F		CMA	Инвертирование аккумулятора
8214	4F		MOV C,A	Копирование данных из регистра A в регистр C
8215	C5		PUSH B	Запись содержимого регистровой пары BC в стек
8216	C3		JMP, M2	Переход на M2
8217	1D		1D	

8218	82		82	
8219	79	M1	MOV A,C	Копирование данных из регистра C в регистр A
821A	80		ADD B	Сложение содержимого регистра B с содержимым аккумулятора
821B	47		MOV B,A	Копирование данных из регистра A в регистр B
821C	C5		PUSH B	Запись содержимого регистровой пары BC в стек
821D	06	M2	MVI B	Запись числа 0F в регистр B
821E	0F		0F	
821F	0E		MVI C	Запись числа 10 в регистр C
8220	10		10	
8221	7C		MOV A,H	Копирование данных из регистра H в регистр A
8222	A0		ANA B	Логическое умножение содержимого аккумулятора с содержимым рег. B
8223	A9		XRA C	Логическое сложение содержимого аккумулятора с содержимым рег. C
8224	67		MOV H,A	Копирование данных из регистра A в регистр H
8225	7A		MOV A,D	Копирование данных из регистра D в регистр A
8226	A0		ANA B	Логическое умножение содержимого аккумулятора с содержимым рег. B
8227	A9		XRA C	Логическое сложение содержимого аккумулятора с содержимым рег. C
8228	57		MOV D,A	Копирование данных из регистра A в регистр D
8229	C1		POP B	Восстановление содержимого регистровой пары BC из стека
822A	79		MOV A,C	Копирование данных из регистра C в регистр A
822B	C6		ADI	Сложение содержимого аккумулятора с нулем
822C	00		00	
822D	CA		JZ, M3	Если 0, идти на M3
822E	3E		3E	
822F	82		82	
8230	7A	M5	MOV A,D	Копирование данных из регистра D в регистр A
8231	1F		RAR	Сдвиг содержимого аккумулятора вправо
8232	57		MOV D,A	Копирование данных из регистра A в регистр D
8233	7B		MOV A,E	Копирование данных из регистра E в регистр A

8234	1F		RAR	Сдвиг содержимого аккумулятора вправо
8235	5F		MOV E,A	Копирование данных из регистра А в регистр Е
8236	D2		JNC, M4	Если нет переноса, идти на М4
8237	3A		3A	
8238	82		82	
8239	3F		CMC	Инвертирование флага переноса
823A	0D	M4	DCR C	Уменьшение содержимого регистра С на единицу
823B	C2		JNZ, M5	Если не 0, идти на М5
823C	30		30	
823D	82		82	
823E	7D	M3	MOV A,L	Копирование данных из регистра L в регистр А
823F	8B		ADC E	Сложение содерж. аккумулятора с содерж. рег. Е с учетом переноса
8240	6F		MOV L,A	Копирование данных из регистра А в регистр L
8241	7C		MOV A,H	Копирование данных из регистра H в регистр А
8242	8A		ADC D	Сложение содерж. аккумулятора с содерж. рег. D с учетом переноса
8243	67		MOV H,A	Копирование данных из регистра А в регистр H
8244	E6		ANI	Логическое умножение содержимого аккумулятора с числом 20
8245	20		20	
8246	CA		JZ, M6	Если 0, идти на М6
8247	50		50	
8248	82		82	
8249	7C		MOV A,H	Копирование данных из регистра H в регистр А
824A	1F		RAR	Сдвиг содержимого аккумулятора вправо
824B	67		MOV H,A	Копирование данных из регистра А в регистр H
824C	7D		MOV A,L	Копирование данных из регистра L в регистр А
824D	1F		RAR	Сдвиг содержимого аккумулятора вправо
824E	6F		MOV L,A	Копирование данных из регистра А в регистр L
824F	03		INX B	Увеличение содержимого регистра В на единицу
8250	7C	M6	MOV A,H	Копирование данных из регистра А в регистр L
8251	E6		ANI	Логическое умножение содержимого аккумулятора с числом 0F

8252	0F		0F	
8253	67		MOV H,A	Копирование данных из регистра А в регистр L
8254	78		MOV A,B	Копирование данных из регистра А в регистр L
8255	0F		RRC	Сдвиг содержимого аккумулятора вправо
8256	0F		RRC	Сдвиг содержимого аккумулятора вправо
8257	0F		RRC	Сдвиг содержимого аккумулятора вправо
8258	0F		RRC	Сдвиг содержимого аккумулятора вправо
8259	AC		XRA H	Логическое сложение содержимого аккумулятора с содержимым рег. H
825A	67		MOV H,A	Копирование данных из регистра А в регистр L
825B	00		NOP	Конец программы
8300	C5		PUSH B	Запись содержимого регистровой пары BC в стек
8301	0E		MVI C	Запись в регистр C числа F0
8302	F0		F0	
8303	A1		ANA C	Логическое умножение содержимого аккумулятора с содержимым рег. C
8304	1F		RAR	Сдвиг содержимого аккумулятора вправо
8305	1F		RAR	Сдвиг содержимого аккумулятора вправо
8306	1F		RAR	Сдвиг содержимого аккумулятора вправо
8307	1F		RAR	Сдвиг содержимого аккумулятора вправо
8308	47		MOV B,A	Копирование данных из регистра А в регистр B
8309	0E		MVI C	Запись в регистр C числа 08
830A	08		08	
830B	A1		ANA C	Логическое умножение содержимого аккумулятора с содержимым рег. C
830C	CA		JZ, M	Если 0, идти на M
830D	14		14	
830E	83		83	
830F	78		MOV A,B	Копирование данных из регистра B в регистр A
8310	0E		MVI C	Запись в регистр C числа F0
8311	F0		F0	
8312	A9		XRA C	Логическое сложение содержимого аккумулятора с содержимым рег. C

8313	47		MOV B,A	Копирование данных из регистра А в регистр В
8314	78		MOV A,B	Копирование данных из регистра В в регистр А
8315	C1	М	POP B	Восстановление содержимого регистровой пары BC из стека
8316	C9		RET	Возврат из подпрограммы

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		D	D	D	S	S	S	1	5	- - - - -
MOV M, R	Передача из регистра в память	0		1	1	0	S	S	S	1	7	- - - - -
MOV R, M	Передача из памяти в регистр	0		D	D	D	1	1	0	1	7	- - - - -
MVI R	Передача байта в регистр	0		D	D	D	1	1	0	2	7	- - - - -
MVI M	Передача байта в память	0		1	1	0	1	1	0	2	10	- - - - -
LXI RP	Загрузка парных регистров B-C, D-E, H-L, S-P	0		R	R	0	0	0	1	3	10	- - - - -
LDAX RP	Загрузка аккумулятора По адресу, указанному Парой регистров B-C или D-E	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		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 В-С, D-E, H- L, SP на единицу	0	0	R	R	0	0	1	1	1	5	-- - - -
DCR RP	Уменьшение содержимого парных регистров В-С, 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	- - - - \bar{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 наличие 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	- - - - -


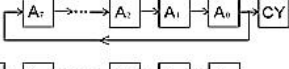

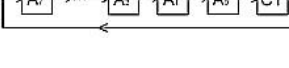
JP	« « плюсе	1	1	1	1	0	0	1	0	3	10	- - - - -
JM	« « минусе	1	1	1	1	1	0	1	0	3	10	- - - - -
JPE	« « четности	1	1	1	0	1	0	1	0	3	10	- - - - -
JPO	« « нечетности	1	1	1	0	0	0	1	0	3	10	- - - - -
PCHL	Занесение в счетчик команд содержимого регистра H-L	1	1	1	0	1	0	0	1	1	5	- - - - -
CALL	Вызов подпрограммы	1	1	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 четности	1	1	1	0	1	1	0	0	3	11/17	- - - - -
CPO	То же, но при нечетности	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 2

Команда	Описание операции	В/М	T ⁸⁰	T ⁸⁵	F
---------	-------------------	-----	-----------------	-----------------	---

<i>Группа команд передачи данных</i>					
MOV r1, r2	$r1 \leftarrow r2$ { <u>M</u> ove}	1/1	5	4	-
MOV r, M	$r \leftarrow M(rp\ H)$	1/2	7	7	-
MOV M, r	$M(rp\ H) \leftarrow r$	1/2	7	7	-
MVI r, d8	$r \leftarrow d8$ { <u>M</u> ove <u>i</u> mmEDIATE}	2/2	7	7	-
MVI M, d8	$M(rp\ H) \leftarrow d8$	2/3	10	10	-
LXI rp, d16	$rp \leftarrow d16, rp=B, D, H \text{ или } SP$ { <u>L</u> oad <u>i</u> mmEDIATE}	3/3	10	10	-
LDA addr	$A \leftarrow M(addr)$ { <u>L</u> oad <u>d</u> irect <u>a</u> ccumulator}	3/4	13	13	-
STA	$M(addr) \leftarrow A$ { <u>S</u> to <u>r</u> e <u>d</u> irect <u>a</u> ccumulator}	3/4	13	13	-
LHLD addr	$L \leftarrow M(addr), H \leftarrow M(addr+1)$ { <u>L</u> oad <u>H</u> and <u>L</u> direct}	3/5	16	16	-
SHLD addr	$M(addr) \leftarrow L, M(addr+1) \leftarrow H$ { <u>S</u> to <u>r</u> e <u>H</u> and <u>L</u> direct}	3/5	16	16	-
LDAX rp	$A \leftarrow M(rp), rp=B \text{ или } D$ { <u>L</u> oad <u>a</u> ccumulator indirect}	1/2	7	7	-
STAX rp	$M(rp) \leftarrow A, rp=B \text{ или } D$ { <u>S</u> to <u>r</u> e <u>a</u> ccumulator indirect}	1/2	7	7	-
XCHG	$H \leftrightarrow D, L \leftrightarrow E$ {Exchange}	1/1	4	4	-
<i>Группа команд арифметических операций</i>					
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> nc <u>r</u> ement}	1/1	5	5	Δ
INR M	$M(rp\ H) \leftarrow M(rp\ H) + 1$	1/3	10	10	Δ
DCR r	$r \leftarrow r - 1$ { <u>D</u> ec <u>r</u> ement}	1/1	5	5	Δ
DCR M	$M(rp\ H) \leftarrow M(rp\ H) - 1$	1/3	10	10	Δ
INX rp	$rp \leftarrow rp + 1, rp=B, D, H \text{ или } SP$ { <u>I</u> nc <u>r</u> ement <u>r</u> p}	1/1	5	5	—
DCX rp	$rp \leftarrow rp - 1, rp=B, D, H \text{ или } SP$ { <u>D</u> ec <u>r</u> ement <u>r</u> p}	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	∇
DAA	Десятичная коррекция {Decimal adjust accumulator}	1/1	4	4	+
<i>Группа команд логических операций</i>					
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	$A \leftarrow A \oplus d8$ { Exclusive or immediate with accumulator }	2/2	7	7	#
XRA M	$A \leftarrow A \oplus M(rp\ H)$ { Exclusive or memory with accumulator }	1/2	7	7	#

Команда	Описание операции		В/М	T ⁸⁰	T ⁸⁵	F
ORA r	A←A ∨ r	{ Or register with accumulator }	1/1	4	4	#
ORI d8	A←A ∨ d8	{ Or immediate with accumulator }	2/2	7	7	#
ORA M	A←A ∨ M(rp H)	{ Or memory with accumulator }	1/2	7	7	#
CMP r	A – r	} Z=1 при A=r, d8, M { Compare } CY=1 при A<r, d8, M { Compare immediate }	1/1	4	4	+
CPI d8	A – d8		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 ← \overline{A}	{ Complement accumulator }	1/1	4	4	–
CMC	CY ← \overline{CY}	{ Complement carry }	1/1	4	4	∇
STC	CY ← 1	{ Set carry }	1/1	4	4	∇
Группа команд передачи управления						
JMP addr	PC ← addr	{ Jump }	3/3	10	10	–
Jcond addr	PC ← addr	{ Jump conditional }	3/3	10	7/10 ¹	–
PCHL	PC ← HL	{ Load Program Counter with H and L }	1/1	5	4	–
CALL addr	} SP ← SP – 1, M(SP) ← PCh, { Call } } SP ← SP – 1, M(SP) ← PCl; PC ← addr { Call conditional }		3/5	17	18	–
Ccond addr			3/3/5 ²	11/17 ¹	9/18 ¹	–
RET	} PCl ← M(SP) , SP ← SP + 1, { Return } } PCh← M(SP), SP ← SP + 1 { Return conditional }		1/3	10	10	–
Rcond			1/1/3 ²	5/11 ¹	6/12 ¹	–
RST n	} SP ← SP – 1, M(SP) ← PCh, { Restart } } SP ← SP – 1, M(SP) ← PCl; PC ← 8 ×n (n= 0 . . . 7)		1/3	11	12	–
Группа команд управления стеком, вводом-выводом и состоянием МП						
PUSH rp	SP←SP – 1, M(SP)←prh, SP←SP – 1, M(SP)←rpl	{ Push }	1/3	11	12	–
PUSH PSW	SP←SP – 1, M(SP) ← A, SP←SP – 1, M(SP)←F		1/3	11	12	–
POP rp	rpl←M(SP), SP←SP + 1, rph ← M(SP), SP←SP+1	{ Pop }	1/3	10	10	–
POP PSW	F←M(SP), SP←SP + 1, A← M(SP), SP←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	–
IN	port	$A \leftarrow PO(port)$ {Input}	2/3	10	10	–
OUT	port	$PO(port) \leftarrow A$ {Output}	2/3	10	10	–
EI ³		INTE = 1 после следующей команды {Enable Interrupts}	1/1	4	4	–
DI ³		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 ⁴		Установка маски прерываний {Set Interrupt Mask}	1/1	–	4	–
RIM ⁴		Чтение маски прерываний {Read Interrupt Mask}	1/1	–	4	–

Примечание: ¹ X из X/Y при невыполнении условия cond и Y при невыполнения условия cond; ² X из Z/X/Y при невыполнении условия cond и Y при невыполнения условия cond; ³ в МП 8085А вместо триггера INTE данные команды управляются флагом (триггером) IE; ⁴ только для МП 8085А

Приложение 3 (лаб. №1)

№	Массив чисел
---	--------------

1	14, E2, CE, F5, A9, 4, 6E
2	E8, D6, 9C, 45, 5B, EE, BA
3	9A, B8, 80, DB, 69, 1C, 70
4	81, C9, A7, F8, 19, 6E, 51
5	8E, BD, 6B, F0, 3B, A0, 1F
6	DB, 56, 5E, F3, 8E, F6, FD
7	DA, 1C, 8, EB, BA, 9, 7C
8	22, 3B, 3A, 83, F2, A7
9	68, DF, 80, C, 48, 32
10	44, 89, E4, 71, 6A, EA, 1B
11	C6, A8, 56, 86, 53, A5
12	6A, CB, 3C, FB, 88, AC
13	5B, 60, 47, 52, 2D, C, B9
14	D6, 76, E3, C5, 63, A
15	4B, 65, 35, 66, 8D, 68
16	0, DC, E1, 4D, B7, 36, 21
17	85, EE, 2B, 87, 8D, 6E, 3
18	A6, D, 5D, CA, AC, BB, 1F
19	57, 7E, B9, 48, D4, 52, D0
20	D3, AB, 1D, C8, 43, 1, F
21	75, A0, 21, 16, DD, 48, 58
22	2D, AF, DF, 84, F5, 8C, 47
23	99, 34, 4D, A0, B3, 68, B9
24	BC, 74, 37, F8, 1A, 21
25	B9, 54, A8, E7, 9, AD, 38
26	18, CE, 19, 60, 44, BE, 1F
27	91, E5, 65, 3B, 40, 4C
28	B4, 42, D1, E2, 28, 3
29	E0, 35, 42, 22, 23, F, 68
30	5A, 5C, 25, 36, 98, C9

Приложение 2 (лаб. №2.2)

№	Массив чисел
1	CB, 21, D3, D2, B4, C4
2	5A, A3, CE, E6, DA, B1, 2D
3	ED, 91, 22, E3, 94, 15
4	5B, 30, A4, 63, D3, 7F
5	E0, A7, BE, E6, AE, FC, B2
6	2F, D1, D4, 69, E0, 62, 50
7	A6, 27, 14, 76, 28, 5B
8	71, 4B, 48, EC, EA, 31, 50
9	3C, B7, EC, 9F, A4, F6, 94
10	3F, B3, D1, E8, 13, 33
11	47, 3F, 7B, C5, 49, F1
12	F7, 56, 30, 9A, D0, E4
13	9C, C9, 9A, 93, 6A, 82
14	5D, 96, E9, 20, 0, 32
15	85, D0, DA, 7A, 37, C2
16	52, 3, C2, 37, 4B, 8F, A0
17	90, DA, 33, EF, 95, EA, BC
18	27, EA, 38, 8F, D3, 76
19	73, 9, 3F, 96, 55, C0, C3
20	C, 10, 60, 71, 46, 71
21	5D, 59, 29, 37, 20, DA
22	F6, 2, 6, 9E, 92, 1E, 4D
23	55, C0, 27, 86, 52, 22, 88
24	B6, 83, F4, 46, 29, 4C, B7
25	ED, 45, 29, 46, A, 7C
26	19, 1C, F7, 5, B7, 28
27	A3, 61, 4, 81, E6, 80, A3
28	C0, E0, 1, E5, 92, 47, CC
29	13, 43, A3, 82, 8, 9B
30	B3, 4A, 21, E1, 77, 9

Приложение 3 (лаб. №3)

№	Массив чисел
1	7, 8, 6, 2, 9, 3, 5
2	4, 1, 8, 7, 2, 9
3	3, 4, 6, 8, 7, 1
4	9, 2, 3, 6, 4, 7, 8
5	1, 9, 5, 3, 2, 6
6	4, 7, 1, 5, 9, 3, 2
7	6, 8, 7, 1, 5, 4, 9
8	2, 6, 3, 7, 1, 5, 8
9	4, 9, 2, 3, 6, 7, 5
10	1, 8, 9, 2, 4, 6, 7
11	5, 3, 8, 1, 2, 9
12	4, 6, 3, 5, 1, 8
13	9, 4, 6, 2, 5, 3
14	1, 9, 4, 8, 6, 5
15	3, 2, 9, 1, 8, 4, 6
16	5, 7, 2, 3, 1, 8
17	9, 4, 7, 5, 3, 1
18	8, 2, 4, 7, 5, 9, 3
19	6, 8, 1, 4, 2, 5, 7
20	3, 6, 8, 1, 4, 9, 2
21	5, 3, 7, 8, 1, 4
22	9, 6, 5, 7, 8, 3, 4
23	1, 9, 6, 5, 2, 8
24	3, 1, 7, 6, 5, 9
25	8, 2, 1, 7, 6, 5
26	9, 8, 2, 1, 3, 7
27	6, 9, 8, 2, 5, 3
28	1, 6, 9, 8, 7, 5, 2
29	3, 4, 6, 9, 1, 7, 8
30	2, 3, 4, 6, 5, 1, 9

Приложение 4 (лаб. №4.1)

№	Делимое	Делитель
1	7C	4
2	DC	7
3	22	5
4	9C	26
5	87	1E
6	66	5
7	54	6
8	C7	30
9	A8	26
10	CA	15
11	77	13
12	77	1D
13	19	4
14	9E	22
15	56	D
16	F0	4C
17	96	21
18	22	9
19	9D	1A
20	E6	3B
21	FE	40
22	E4	D
23	90	4
24	57	16
25	EA	19
26	C1	5
27	42	8
28	A0	B
29	43	10
30	4C	4

Приложение 5 (лаб. 5)

№	Множимое	Множитель
1	DA	F8
2	E4	B9
3	AA	EA
4	87	97
5	F2	2C
6	37	AF
7	41	4F
8	D7	E6
9	D1	98
10	6A	F6
11	AD	AA
12	8E	DB
13	4D	66
14	AD	15
15	B3	76
16	E6	DB
17	22	86
18	12	6C
19	F1	F2
20	81	53
21	71	4A
22	C6	7D
23	59	82
24	2C	BD
25	2B	1D
26	5C	F9
27	82	36
28	8E	B8
29	31	3E
30	6E	45

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Микропроцессоры: Справочное пособие/ Под ред. Ю.А.Овечкина. Л.:Судостроение, 1987.519 с.
2. Гилмор Ч. Введение в микропроцессорную технику. М.: Мир, 1984.331 с.

3. Микропроцессоры/ Под ред. Л.Н.Преснухина. М.: Высшая школа,1986. Т.3.350 с.
4. Богумирский С.П. Руководство пользователя ПЭСМ в 2-х томах. СП: "Ассоциация OLIKO", 1992.764с.
5. Мячев И.С. Интерфейсы средств вычислительной техники. М.: Радио и связь, 1993.254с.