

Федеральное государственное автономное образовательное учебное
учреждение высшего образования “Национальный исследовательский
университет ИТМО”

Факультет программной инженерии и компьютерной техники

Лабораторная работа №3
“Выполнение циклических программ”

Вариант №112

Выполнил:

Ситдигов Рафаэль Ильдусович

Группа: Р3131

Проверил:

Цю Тяньшэн

Г. Санкт-Петербург, 2025г.

Оглавление:

Задание	3
Выполнение	4
Трассировка	6
Ответы на теоретические вопросы	8
Заключение	8

Задание:

По выданному преподавателем варианту восстановить текст заданного варианта программы, определить предназначение и составить описание программы, определить область представления и область допустимых значений исходных данных и результата, выполнить трассировку программы.

3FE:	0414		40C:	0380
3FF:	A000		40D:	0400
400:	4000		40E:	AEF2
401:	E000		40F:	0700
402:	+ 0200		410:	EEF0
403:	EEFD		411:	8400
404:	AF03		412:	CEF5
405:	EEFA		413:	0100
406:	AEF7		414:	1200
407:	EEF7		415:	F601
408:	AAF6		416:	0000
409:	0480			
40A:	0380			
40B:	F405			

Шаблон микрокоманд для операции:

Система команд базовой ЭВМ

Код	Команда	Признаки ¹				Описание
		N	Z	V	C	
0XXX						Безадресные команды
0000	NOP	–	–	–	–	Нет операции
0100	HLT	–	–	–	–	Останов
0200	CLA	*	*	0	–	0 → AC
0280	NOT	*	*	0	–	(^AC) → AC
0300	CLC	–	–	–	0	0 → C
0380	CMC	–	–	–	*	(^C) → C
0400	ROL	*	*	*	*	AC и C сдвигается влево. AC15 → C, C → AC0
0480	ROR	*	*	*	*	AC и C сдвигается вправо. AC0 → C, C → AC15
0500	ASL	*	*	*	*	AC сдвигается влево. AC15 → C, 0 → AC0
0580	ASR	*	*	*	*	AC сдвигается вправо. AC0 → C, AC15 → AC14
0600	SXTB	*	*	0	–	Расширение знака мл. байта AC7 → AC15...AC8
0680	SWAB	*	*	0	–	Обмен ст. и мл. байта AC7...AC0 ↔ AC15...AC8
0700	INC	*	*	*	*	AC + 1 → AC
0740	DEC	*	*	*	*	AC – 1 → AC
0780	NEG	*	*	*	*	^AC + 1 → AC
0800	POP	*	*	0	–	(SP)+ → AC
0900	POPF	*	*	*	*	(SP)+ → PS
0A00	RET	–	–	–	–	(SP)+ → IP
0B00	IRET	*	*	*	*	(SP)+ → PS, (SP)+ → IP
0C00	PUSH	–	–	–	–	AC → –(SP)
0D00	PUSHF	–	–	–	–	PS → –(SP)
0E00	SWAP	*	*	0	–	Обмен A и вершины стека
1XXX						Команды ввода-вывода
10XX	DI	–	–	–	–	Запрет прерывания
11XX	EI	–	–	–	–	Разрешение прерываний
12XX	IN REG	–	–	–	–	Чтение из регистров ВУ
13XX	OUT REG	–	–	–	–	Запись в регистры ВУ
18XX	INT NUM	*	*	*	*	Программное прерывание с заданным вектором
XXXX						Адресные команды
2XXX	AND M	*	*	0	–	M & AC → AC
3XXX	OR M	*	*	0	–	M AC → AC
4XXX	ADD M	*	*	*	*	M + AC → AC
5XXX	ADC M	*	*	*	*	M + AC + C → AC
6XXX	SUB M	*	*	*	*	AC – M → AC
7XXX	CMP M	*	*	*	*	Установить флаги по результату AC – M
8XXX	LOOP M	–	–	–	–	M – 1 → M; Если M ≤ 0, то IP + 1 → IP
9XXX						Резерв
AXXX	LD M	*	*	0	–	M → AC
BXXX	SWAM	*	*	0	–	M ↔ AC
CXXX	JUMP M	–	–	–	–	M → IP
DXXX	CALL M	–	–	–	–	SP – 1 → SP, IP → (SP), M → IP
EXXX	ST M	–	–	–	–	AC → M
FXXX						Команды ветвления
F0XX	BEQ (BZS)	–	–	–	–	Переход если равенство (Z==1)
F1XX	BNE (BZC)	–	–	–	–	Переход если неравенство (Z==0)
F2XX	BMI (BNS)	–	–	–	–	Переход если минус (N==1)
F3XX	BPL (BNC)	–	–	–	–	Переход если плюс (N==0)
F4XX	BHIS (BCS)	–	–	–	–	Переход если выше или равно/перенос (C==1)
F5XX	BLO (BCC)	–	–	–	–	Переход если ниже/нет переноса (C==0)
F6XX	BVS	–	–	–	–	Переход если переполнение (V==1)
F7XX	BVC	–	–	–	–	Переход если нет переполнения (V==0)
F8XX	BLT	–	–	–	–	Переход если меньше (N⊕V==1 / N!=V)
F9XX	BGE	–	–	–	–	Переход если больше или равно (N⊕V==0 / N==V)
FAXX						Резерв

Примечания:

1. Значения в столбцах признаков результатов показывают, как изменится соответствующий признак в результате выполнения операции. «–» – команда не влияет на признак, «0» – признак сбросится, «*» – значение признака установится по результату операции.

Выполнение работы

Текст исходной Программы:

Адрес	Код команды	Мнемоника	Комментарий	Виды адресаций
3FE	0414		Значение А	
3FF	A000		Значение В	
400	4000		Значение С	
401	E000		Значение D	
402	0200	CLA	Записывает ноль в АС	Безадресная
403	EEFD	ST EED	Записывает значение ячейки 401 в АС	Прямая относительная
404	AF03	LD F03	Записывает значение АС в ячейку F03	Прямая абсолютная
405	EEFA	ST EFA	Записывает значение ячейки 400 в АС	Прямая относительная
406	AEF7	LD EF7	Записывает значение АС в ячейку 3FE	Прямая относительная
407	EEE7	ST EE7	Записывает значение ячейки 3FF в АС	Прямая относительная
408	AAF6	LD AF6	Записывает значение АС в ячейку EF6	Прямая относительная
409	0480	ROR	АС и С сдвигаются вправо. AC0 → C, C → AC15	Безадресная
40A	0380	CMC	(^C) → C	Безадресная
40B	F405	BHIS (BCS)	Переход к ячейке 411	Ветвление с равенством
40C	0380	CMC	(^C) → C	Безадресная
40D	0400	ROL	АС и С сдвигаются влево. AC15 → C, C → AC0	Безадресная
40E	AEF2	LD EF2	Записывает значение АС в ячейку EF2	Прямая относительная
40F	0700	INC	AC + 1 → AC	Безадресная
410	EEF0	ST M	Записывает значение АС в М	Прямая относительная
411	8400	LOOP M	M – 1 → M	Прямая абсолютная
412	CEF5	JUMP M	M → IP	Прямая относительная
413	0100	HLT	Останавливает выполнение программы	Безадресная
414	1200		A[0]	
415	F601		A[1]	
416	0000		A[2]	

Назначение программы:

- Подсчет количества ненулевых элементов массива А
- d – результат подсчета
- c – количество элементов массива, т. е. повторения цикла
- a – адрес первого элемента массива
- b – адрес текущего элемента массива

$$MEM(401) = \sum_{j=a}^c \begin{cases} 0 & \text{if } MEM(j) == 0 \\ 1 & \text{if } MEM(j) \neq 0 \end{cases}$$

где c – количество элементов массива, a – адрес первого элемента

Область представления:

- a, b, c, d – 16ти разрядные целые числа в прямом коде
- $A[0], A[1], A[2]$ - 16ти разрядные целые числа в дополнительном коде

Область допустимых значений:

- Элементы массива $A[i] \in [-32768; 32767]$ (т. е. $[-2^{15}; 2^{15}-1]$, так как 16й бит – знаковый)
- $c, d \in [1; 2030]$
- $b \in [a; a + c - 1]$
- $a \in [0; 1407 - c] \cup [1425; 2048 - c]$

Фактическое ОДЗ для a и b :

$a \in [0; 1404] \cup [1425; 2045]$

$b \in [1425; 1427]$

Расположение в памяти ЭВМ исходных данных:

- 3FE, 400, 414, 415, 416 – исходные данные
- 3FF – промежуточный результат
- 401 – итоговый результат
- 402 – 413 – команды

Адреса первой и последней выполняемой команды:

- Адрес первой команды: 402
- Адрес последней команды: 413

Вывод:

- В ходе выполнения данной лабораторной работы я научился работать с циклами, ветвлениями, одномерными массивами, прямой относительной и косвенной адресацией, изучил цикл выполнения таких команд как LOOP и JUMP

Таблица трассировки:

Выполняемая команда		Содержимое регистров процессора после выполнения команды									Ячейка, содержимое которой изменилось после выполнения команды	
Адрес	Код	IP	CR	AR	DR	SP	BR	AC	PS	NZVC	Адрес	Значение
3FE	0414	3FF	0414	3FE	0414	000	03FE	0000	004	0100		
3FF	A000	400	A000	000	0000	000	03FF	0000	004	0100		
400	4000	401	4000	000	0000	000	0400	0000	004	0100		
401	E000	402	E000	000	0000	000	0401	0000	004	0100		
402	0200	403	0200	402	0200	000	0402	0000	004	0100		
403	EEFD	404	EEFD	401	0000	000	FFFD	0000	004	0100	401	0000
404	AF03	405	AF03	404	0003	000	0003	0003	000	0000		
405	EEFA	406	EEFA	400	0003	000	FFFA	0003	000	0000	400	0003
406	AEFA	407	AEFA	401	0000	000	FFFA	0000	004	0100		
407	EEE7	408	EEE7	3EF	0000	000	FFE7	0000	004	0100	3EF	0000
408	AAF6	409	AAF6	000	0000	000	FFF6	0000	004	0100	3FF	A001
409	0480	40A	0480	409	0480	000	0409	0000	004	0100		
40A	0380	40B	0380	40A	0380	000	040A	0000	005	0101		
40B	F405	411	F405	40B	F405	000	0005	0000	005	0101		
411	8400	412	8400	400	0002	000	0001	0000	005	0101	400	0002
412	CEF5	408	CEF5	412	0408	000	FFF5	0000	005	0101		
408	AAF6	409	AAF6	001	0000	000	FFF6	0000	005	0101	3FF	A002
409	0480	40A	0480	409	0480	000	0409	8000	00A	1010		
40A	0380	40B	0380	40A	0380	000	040A	8000	00B	1011		
40B	F405	411	F405	40B	F405	000	0005	8000	00B	1011		
411	8400	412	8400	400	0001	000	0000	8000	00B	1011	400	0001
412	CEF5	408	CEF5	412	0408	000	FFF5	8000	00B	1011		
408	AAF6	409	AAF6	002	0000	000	FFF6	0000	005	0101	3FF	A003
409	0480	40A	0480	409	0480	000	0409	8000	00A	1010		
40A	0380	40B	0380	40A	0380	000	040A	8000	00B	1011		
40B	F405	411	F405	40B	F405	000	0005	8000	00B	1011		
411	8400	413	8400	400	0000	000	FFFF	8000	00B	1011	400	0000
412	CEF5											
413	0100	414	0100	413	0100	000	0413	8000	00B	1011		

Теоретические вопросы:

1. Организация одномерных массивов данных в памяти. Организация и обработка массивов с числом измерений, больше чем одно.
2. Сравнение значений в БЭВМ. Команды условных и безусловного переходов.
3. Организация циклических вычислений. Команда LOOP.
4. Режимы адресации БЭВМ.
5. Описание адресных команд и команд переходов с различными режимами адресации: наименование, назначение, тип команды и вид адресации. Количество и название машинных циклов, потактовое выполнение команд.
6. Количество обращений к памяти команд БЭВМ с различными режимами адресации.
7. Где находятся аргументы программы? Где находится результат? Как они представлены?
8. Какое максимальное количество элементов данных может поддерживать ваша программа?

1. q

2. Q

3. LOOP – признак команды не изменяется. Выполняется условием: если $M > 0$: $M - 1 \rightarrow M$; иначе: $IP + 1 \rightarrow IP$;

4. Безадресная команда – выполняет команду без ссылки на ячейку памяти. (Например: CLA или HLT)

Прямая адресная – Выполняет команду по адресу (Например: ADD)

Прямая относительная - Используется, когда память логически разбивается на блоки (сегменты). Адрес ячейки памяти содержит две составляющих: адрес начала сегмента (базовый адрес) и смещение адреса операнда в сегменте.

Прямая абсолютная

Ветвление с адресом

Косвенная адресация - Адрес, размещённый в адресной части команды, указывает на ячейку, в которой находится адрес операнда. При мнемонической записи команд указание косвенной адресации производится путём заключения адреса в скобки. Например, команда ADD (25) — сложить содержимое A с содержимым ячейки, адрес которой хранится в ячейке 25 (косвенная адресация).

- **Прямая адресация.** Физический адрес операнда совпадает с кодом в адресной части команды. 5
- **Непосредственная адресация.** В команде содержится не адрес операнда, а непосредственно сам операнд. Позволяет повысить скорость выполнения операции. 5
- **Косвенная адресация.** Адресная часть команды указывает адрес ячейки памяти или номер регистра, в которых содержится адрес операнда. 5
- **Относительная адресация.** Используется, когда память логически разбивается на блоки (сегменты). Адрес ячейки памяти содержит две составляющих: адрес начала сегмента (базовый адрес) и смещение адреса операнда в сегменте. 5
- **Регистровая адресация.** Операнд (входной или выходной) находится во внутреннем регистре процессора. Самая быстрая адресация, так как не требует дополнительных циклов обмена по магистрали. 4
- **Автоинкрементная адресация.** После выполнения команды содержимое используемого регистра увеличивается на единицу или на два. Этот метод адресации удобен, например, при последовательной обработке кодов из массива данных, находящегося в памяти. 4
- **Автодекрементная адресация.** Содержимое выбранного регистра уменьшается на единицу или на два перед выполнением команды. Эта адресация также удобна при обработке массивов данных. 4
- **Индексная адресация.** Формирование исполнительного адреса осуществляется путём добавления к базовому адресу содержимого индексного регистра. 1

Итоги

В результате выполнения лабораторной работы был изучен принцип работы Б-ЭВМ. Также изучены команды, которые может выполнять Б-ЭВМ.