

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
«Вятский государственный университет»
Факультет автоматики и вычислительной техники
Кафедра электронных вычислительных машин

Отчет по лабораторной работе №4 по дисциплине
«Организация ЭВМ и систем»
Задание 8

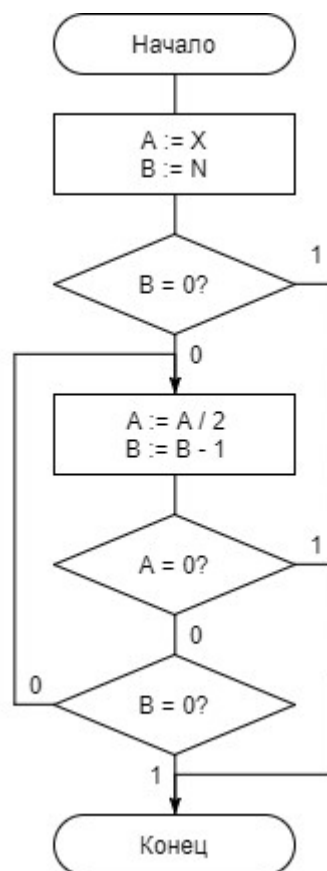
Выполнил студент группы ИВТ-32 _____/Рзаев А. Э./
Проверил преподаватель _____/Вожегов Д. В./

1 Задание

Определить архитектуру ЭВМ, система команд которой состоит из одноадресных команд, использующих прямую адресацию; разработать структурную схему и алгоритм работы ЭВМ; составить и отладить микропрограмму командного цикла ЭВМ; составить и выполнить программу сдвига 16-разрядного двоичного кода X на N разрядов в сторону младших разрядов (сдвиг логический).

2 Определение архитектуры и программирование

2.1 Схема алгоритма



2.2 Форматы данных

X и N изменяются в пределах $0..65536$, поэтому любое число можно представить 16-разрядным двоичным кодом без знака.

2.3 Программно-доступные регистры

Программно-доступными регистрами МЭ, система команд которой

состоит из одноадресных команд, можно считать: аккумулятор AX, счетчик CX, программный счетчик PC и регистр признаков RP, содержащий в простейшем случае разряд признака нуля (Z).

2.4 Система команд

Название	Мнемоника		Изменение признака Z
Загрузка AX	LDA A	$AX := M[A]; PC := PC + 1$	+
Загрузка CX	LDC A	$CX := M[A]; PC := PC + 1$	-
Запись AX	MOV A	$M[A] := AX; PC := PC + 1$	-
Сдвиг AX вправо	SHR	$AX := AX / 2; PC := PC + 1$	+
Переход, если ноль	JZ A	Если $Z = 1, PC := A$, иначе $PC := PC + 1$	-
Цикл	LOOP A	Если $Z = 0, PC := A$, иначе $PC := PC + 1$	-
Останов	HLT	$PC := 0$	-

2.5 Программа

```

1   LDA AN
2   JZ m1
3   LDA AX
4   LDC AN
m0: 5   SHR
6   JZ m2
7   LOOP m0
m2: 8   MOV AX
m1: 9   HLT

```

В программе приняты следующие обозначения:

- AX — адрес ячейки памяти (ЯП), в которой находится число
- AN — адрес ЯП, в которой находится количество сдвигов

3 Кодирование программы и распределение памяти программ и данных

3.1 Коды операций

Название	Мнемоника	Код операции
Загрузка AX	LDA A	0x01

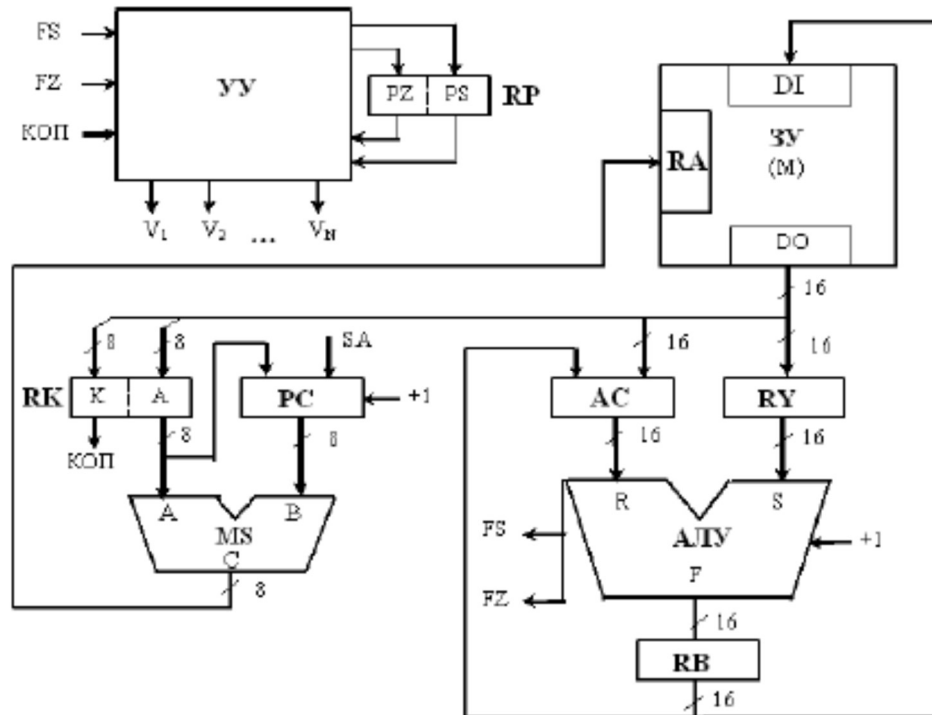
Название	Мнемоника	Код операции
Загрузка CX	LDC A	0x07
Запись AX	MOV A	0x02
Сдвиг AX вправо	SHR	0x04
Переход, если ноль	JZ A	0x05
Цикл	LOOP A	0x03
Останов	HLT	0x00

3.2 Распределение памяти программ и данных

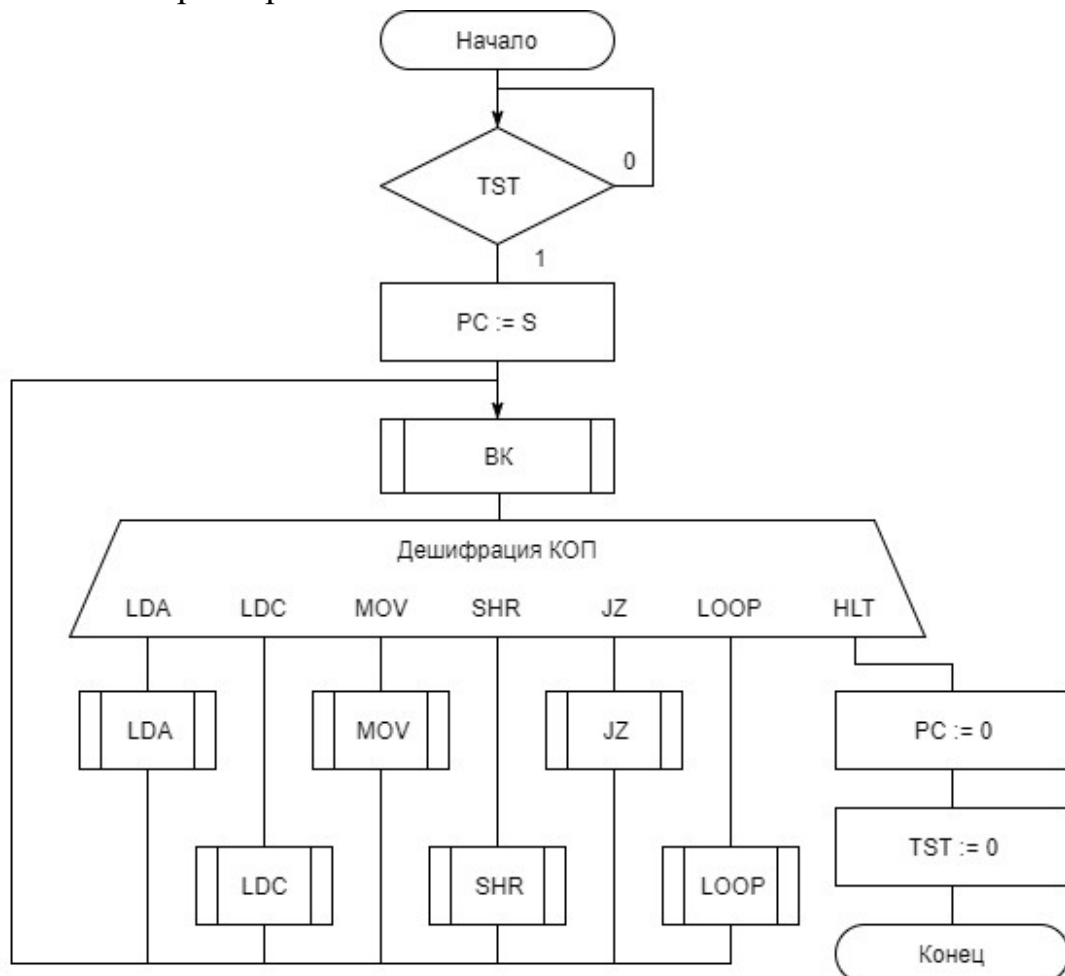
Адрес	Код	Мнемоника	Комментарии
00	0005	S	Адрес начала программы
01		X	Число
02		N	Количество сдвигов
03			Свободная ячейка
04			Свободная ячейка
05	0102	LDA AN	Загрузка N в AX для проверки на 0
06	050D	JZ m1	Завершение, если N = 0
07	0101	LDA AX	Загрузка X в AX
08	0702	LDC AN	Загрузка N в CX
09	0400	SHR	Сдвиг AX на один разряд вправо
0A	050C	JZ m2	Если X = 0, записать значение в память
0B	0309	LOOP m0	Повторить цикл CX раз
0C	0201	MOV AX	Записать AX в память
0D	0000	HLT	Завершение

4 Разработка структуры и алгоритма работы

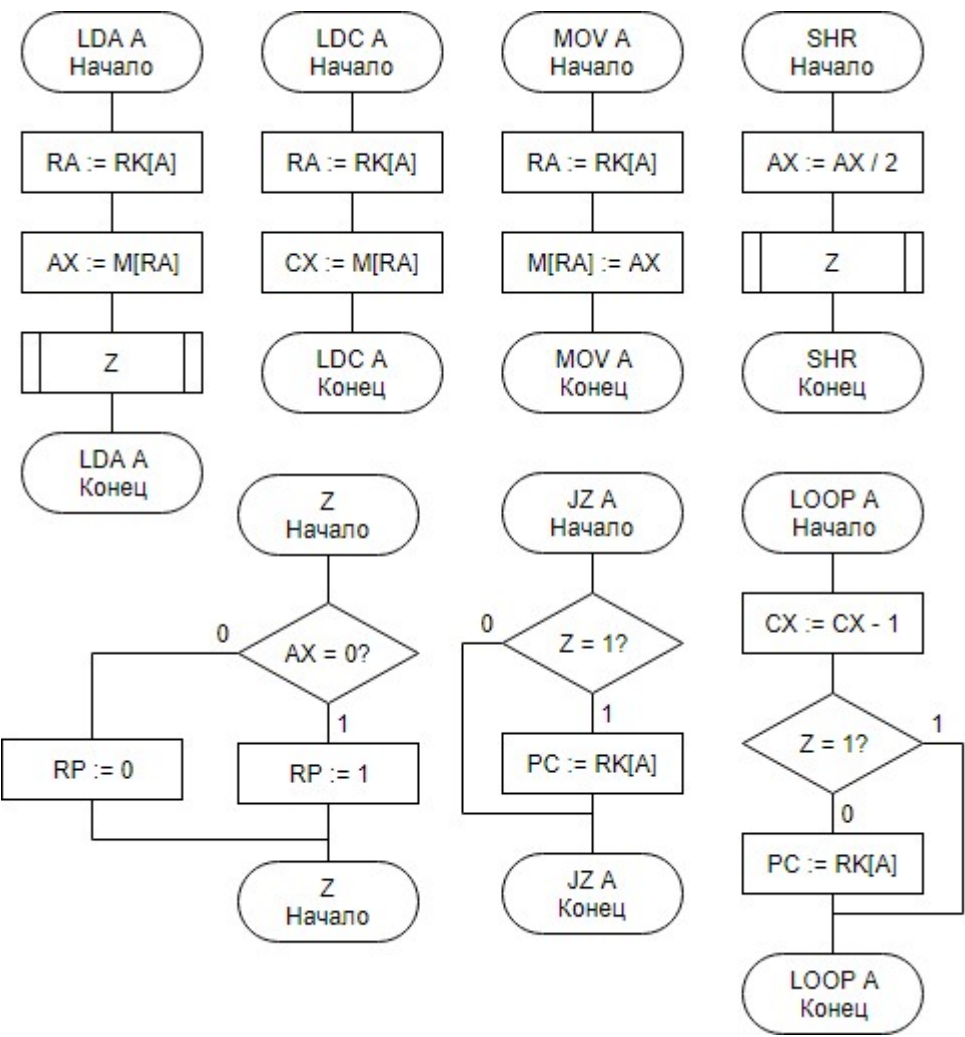
4.1 Структура учебной ЭВМ



4.2 Алгоритм работы ЭВМ



4.3 Алгоритм работы подмикропрограмм

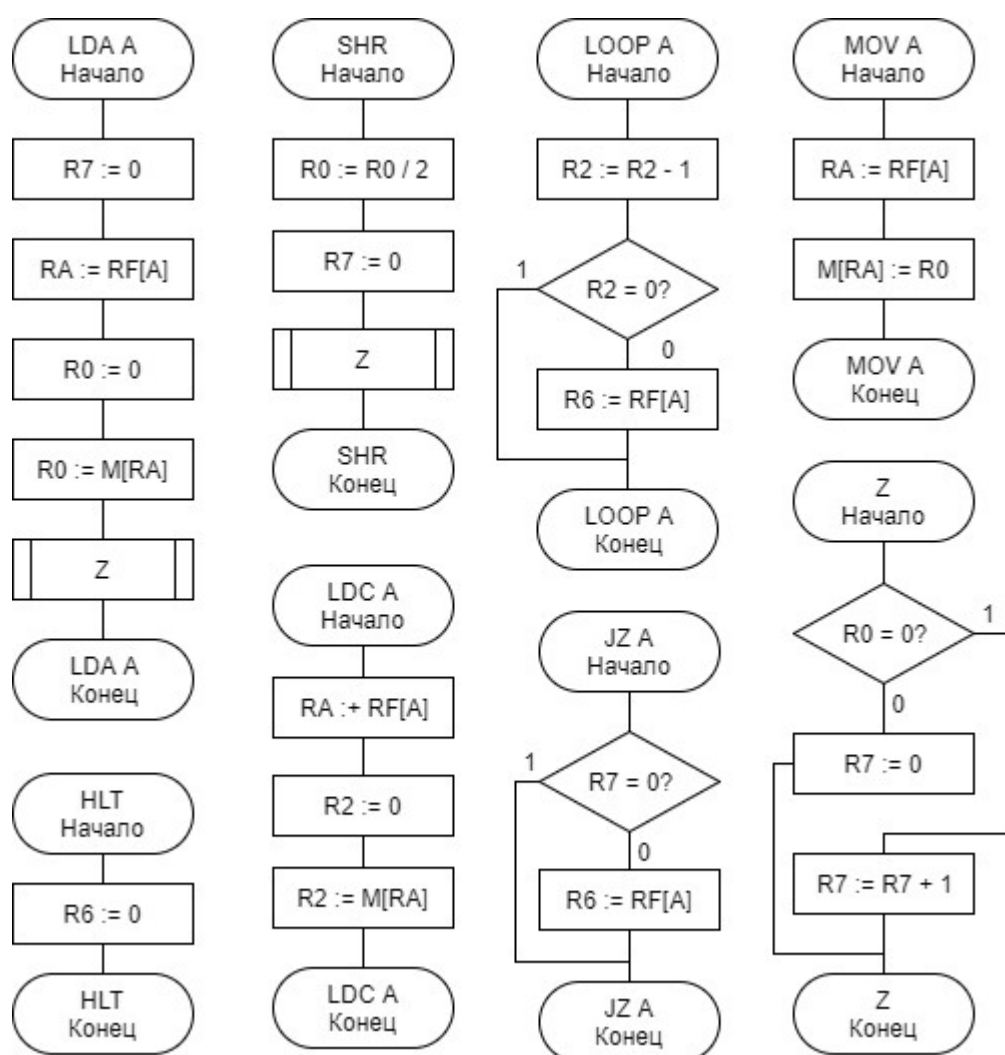


5 Микропрограммная реализация ЭВМ

5.1 Распределение регистров

РЗУ (R0 – R7)			РЗУ (R8 – R15)	
0	AX		8	
1			9	
2	CX		10	
3			11	
4			12	
5			13	Регистр для данных из ЗУ
6	PC		14	Регистр константы
7	RP	Z	15	Счетчик адреса ЗУ
RA	Адрес ЗУ		RQ	

5.2 Граф-схемы подмикропрограмм операций



5.3 Коды операций и начальные адреса подмикропрограмм

Мнемоника	Код операции	Адрес первой микрокоманды
LDA A	0x01	0x0F
MOV A	0x02	0x1A
LOOP A	0x03	0x21
SHR	0x04	0x2A
JZ A	0x05	0x2E
LDC A	0x07	0x32
HLT	0x00	0x1E

5.4 Микропрограмма командного цикла (выборка команды и установка признаков)

№	МИ	РЗУ		Упр. АЛУ			Упр. ОЗУ			Шина	МИ	Упр. усл.			Упр. УУ		
	I8-0	A	B	C0	^OE	SC	^CS	^W	^EA	D11-0	I3-10	A	U	^CCE	C0	^RLD	^OE
00	571	E	E	0	0	00	1	1	1	006	C	00	0	0	1	1	0
RE:=0111111111111111; PA/CIQ:=6																	
01	533	0	E	0	0	00	1	1	1	001	9	00	0	0	1	1	0
RE – сдвиг вправо; PA/CIQ:=PA/CIQ-1																	
02	143	0	6	0	0	00	1	1	0	000	E	00	0	0	1	1	0
RA:=0																	
03	337	0	6	0	1	00	0	1	1	000	E	00	0	0	1	1	0
R6:=SA (PC:=S)																	
04	203	6	6	1	0	00	1	1	0	000	E	00	0	0	1	1	0
RA:=R6; R6:=R6+1 (RA:=PC; PC:=PC+1)																	
05	245	E	F	0	1	00	0	1	1	000	2	00	0	0	1	1	0
Переход по КОП, RF:=K[A] (RK[A]:=K[A])																	
06	303	0	7	0	1	00	1	1	1	004	3	00	0	1	1	1	0
Переход к микрокоманде 04																	

5.5 Микропрограмма командного цикла (выполнение операций)

№	МИ	РЗУ		Упр. АЛУ			Упр. ОЗУ			Шина	МИ	Упр. усл.			Упр. УУ		
	18-0	A	B	C0	^OE	SC	^CS	^W	^EA	D11-0	I3-I0	A	U	^CCE	C0	^RLD	^OE
0F	343	7	7	0	1	00	1	1	1	000	E	00	0	0	1	1	0
R7 := 0 (LDA A)																	
10	133	F	F	0	0	00	1	1	0	000	E	00	0	0	1	1	0
RA := RF[A]																	
11	343	0	0	0	1	00	1	1	1	000	E	00	0	0	1	1	0
R0 := 0																	
12	335	0	0	0	1	00	0	1	1	000	E	00	0	0	1	1	0
R0 := M[RA]																	
13	133	0	0	0	1	00	1	1	1	016	3	00	0	0	1	1	0
Переход, если R0 != 0																	
14	303	7	7	1	1	00	1	1	1	000	E	00	0	0	1	1	0
R7 := R7 + 1																	
15	133	0	0	0	1	00	1	1	1	017	3	00	1	1	1	1	0
Переход к МК 17																	
16	361	7	7	0	1	00	1	1	1	000	E	00	0	0	1	1	0
R7 := 0																	
17	133	0	0	0	1	00	1	1	1	006	3	00	1	1	1	1	0
Переход к МК 06																	

№	МИ	РЗУ		Упр. АЛУ			Упр. ОЗУ			Шина	МИ	Упр. усл.			Упр. УУ		
	I8-0	A	B	C0	^OE	SC	^CS	^W	^EA	D11-0	I3-I0	A	U	^CCE	C0	^RLD	^OE
1A	133	F	F	0	0	00	1	1	0	000	E	00	0	0	1	1	0
RA := RF[A] (MOV A)																	
1B	133	0	0	0	0	00	0	0	1	006	3	00	0	1	1	1	0
M[RA] := R0																	
1E	361	6	6	0	0	00	1	1	1	000	3	00	0	1	1	1	0
R6 := 0 (HLT)																	
21	313	2	2	0	0	00	1	1	1	006	3	00	1	0	1	1	0
R2 := R2 - 1 (LOOP A) Переход к МК 06, если R2 = 0																	
22	334	F	6	0	1	00	1	1	1	004	3	00	0	1	1	1	0
R6 := RF[A] Переход к МК 04																	
2A	533	0	0	0	1	00	1	1	1	000	E	00	0	0	1	1	0
R0 := R0 / 2 (SHR)																	
2B	343	7	7	0	1	00	1	1	1	013	3	00	0	1	1	1	0
R7 := 0. Переход к МК 13, обновление регистра признаков																	
2E	133	7	7	0	1	00	1	1	1	006	3	00	1	0	1	1	0
Переход, если R7 != 0 (JZ A)																	
2F	334	F	6	0	1	00	1	1	1	004	3	00	0	1	1	1	0
R7 := RF[A], переход к МК 04																	
32	133	F	F	0	0	00	1	1	0	000	E	00	0	0	1	1	0
RA := RF[A] (LDC A)																	
33	343	2	2	0	1	00	1	1	1	000	E	00	0	0	1	1	0
R2 := 0																	
34	335	2	2	0	1	00	0	1	1	006	3	00	0	1	1	1	0
R2 := M[RA]																	

6 Расчет производительности и быстродействия

Допустим, что среднее число циклов в программе сдвига $C = 8$, вероятность обнуления операнда до окончания цикла $p_1 = 0.004$, вероятность того, что $N = 0$, $p_2 = 0.004$; кроме того, будем считать, что время обращения к ЗУ включено во время выполнения микрокоманд.

Тип команды	Ср. число команд, h_i	Ср. число микрокоманд, b_i	$h_i * b_i$	Вероятность команды p_i	$b_i * p_i$
LDA	2.992	7.5	22.44	0.097	0.73
LDC	0.996	3	2.988	0.032	0.1
MOV	0.996	2	1.992	0.032	0.06
JZ	8.984	1.5	13.476	0.290	0.44
SHR	7.984	5.5	43.912	0.258	1.42

Тип команды	Ср. число команд, h_i	Ср. число микрокоманд, b_i	$h_i * b_i$	Вероятность команды p_i	$b_i * p_i$
LOOP	7.984	1.5	11.976	0.258	0.39
HLT	1	1	1	0.032	0.032
	$H = 30.936$		$R = 97.784$		$r = 3.17$

Примем $\tau = 100$ нс. Тогда среднее время выполнения команды составит $T_V = r\tau \approx 317$ нс, а быстродействие $V = 1/T_V = 3.15$ млн. команд/сек. Аналогично среднее время решения задачи – $T_W = R\tau \approx 9.8$ мкс, производительность – $W = 1/T_W \approx 102$ тыс. задач/сек.

7 Вывод

В ходе лабораторной работы была разработана и изучена учебная ЭВМ; разработана и реализована система команд, написана программа решения задачи, которая была помещена в ОЗУ. При выборке данных из ОЗУ старшие 8 бит указывали на код операции, который затем поступал в преобразователь начального адреса – так осуществлялся механизм вызова нужной микропрограммы, а младшие 8 бит указывали на адрес данных. Так была реализована прямая адресация.