

Министерство образования и науки Российской Федерации
Федеральное агентство по образованию
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Вятский государственный университет»
Факультет автоматики и вычислительной техники
Кафедра электронных вычислительных машин

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

Вариант 4

Выполнил студент группы ИВТ-31 _____/Крючков И. С/
Проверил _____/Клюкин В.Л./

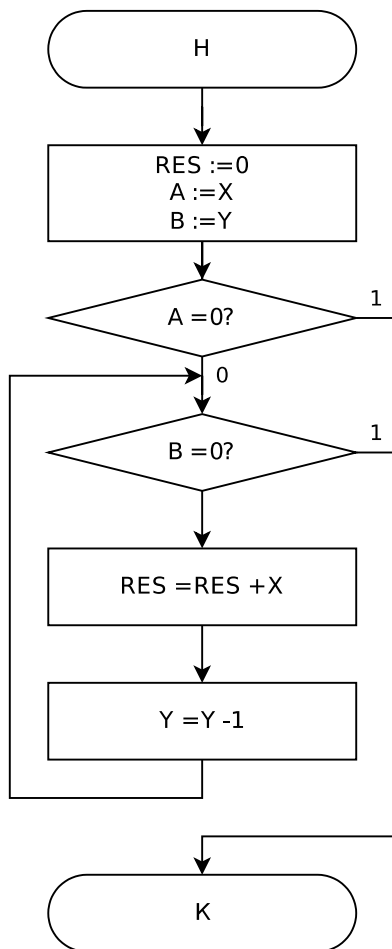
Киров 2022

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. Выполнить умножение: $Z=XY$ (X и Y – целые числа от 0 до 255) путем Y кратного суммирования множимого X

2. Определение структуры и программирование

2.1 Схема алгоритма



2.2 Форматы данных

X и Y изменяются в пределах от 0 до 255, 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 АС	LDA A	AC: = M[A], PC: = PC + 1	+
ОЧИСТКА	CLM A	M[A]: = 00...0, PC: = PC + 1	-
СЛОЖЕНИЕ	ADD A	AX: = AX – M[A], PC: = PC + 1	-
ПЕРЕХОД, ЕСЛИ НУЛЬ	BEQ A	Если PZ = 1, то PC: = A, иначе PC: = PC + 1	-
ПЕРЕХОД	BR A	PC: = A	
ДЕКРЕМЕНТ	DEC A	M[A]: = M[A] - 1, PC: = PC + 1	+
ОСТАНОВ	HLT A	PC: = A, останов	-

2.5 Программа

	1	CLM RES
	2	LDA X
	3	BEQ X
	4	LDA Y
	5	BEQ Y
m0	6	ADD X
	7	DEC Y
	8	BEQ m1
	9	BR m0
m1	10	HLT

3. Кодирование программы и распределение памяти программ и данных

3.1 Коды операций

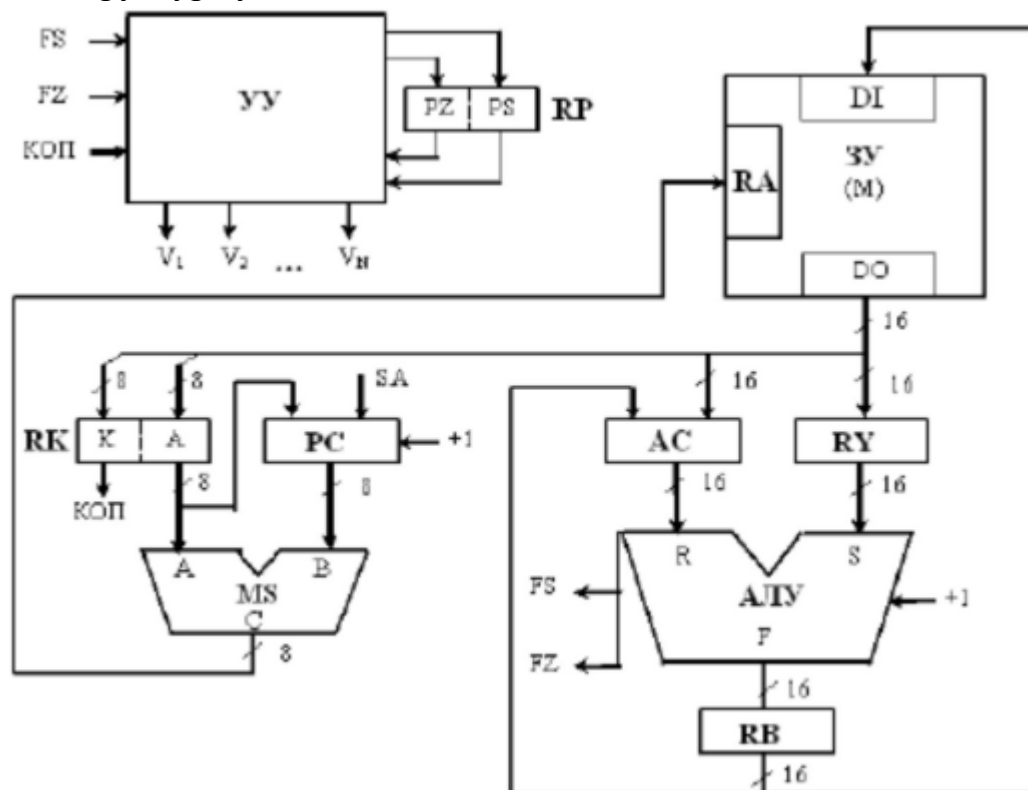
Название	Мнемоника	Код операции
ЗАГРУЗКА АС	LDA A	0x06
ОЧИСТКА	CLM A	0x02
СЛОЖЕНИЕ	ADD A	0x04
ПЕРЕХОД, ЕСЛИ НУЛЬ	BEQ A	0x07
ПЕРЕХОД	BR A	0x05
ДЕКРЕМЕНТ	DEC A	0x03
ОСТАНОВ	HLT A	0x00

3.2 Распределение памяти программ и данных

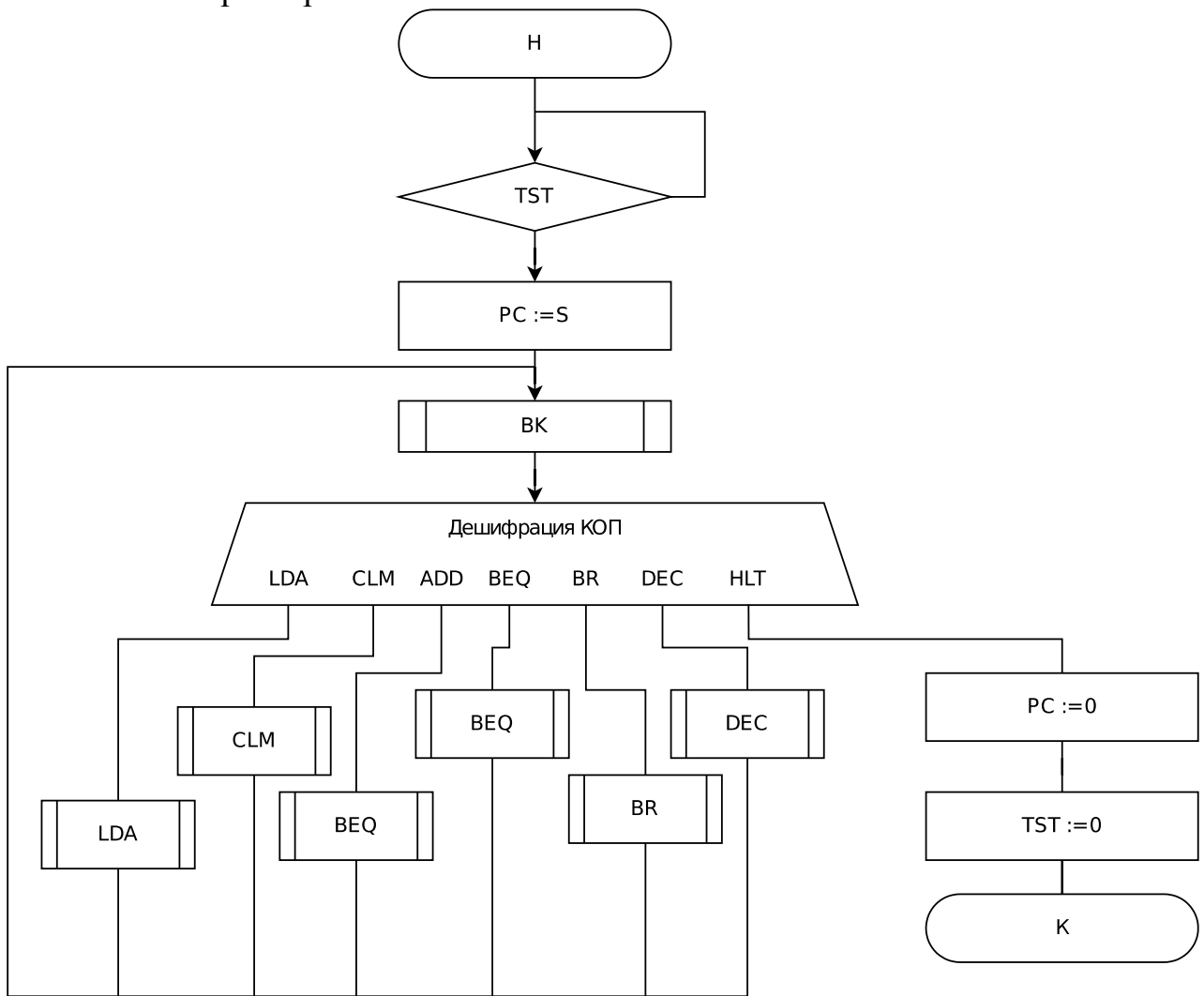
ОЗУ	Комментарии	ПНА
00: 0006	SA.....	00: 1C
01: 0003	X.....	01: 0E
02: 0000	Y.....	02: 0C
03: 000F	Z.....	03: 18
04: 0000	P.....	04: 12
05: 0000	05: 16
06: 0203	Z:=0.....	06: 0F
07: 0601	AC:=X.....	07: 15
08: 070F	X = 0 - ?.....	08: 00
09: 0602	AC:=Y.....	09: 00
0A: 070F	Y = 0 - ?.....	0A: 00
0B: 0401	Z:=Z+X.....	0B: 00
0C: 0302	AC:=AC-1.....	0C: 00
0D: 070F	AC = 0?.....	0D: 00
0E: 050B	->0B.....	0E: 00
0F: 0003	STOP.....	0F: 00

4. Разработка структуры и алгоритма работы

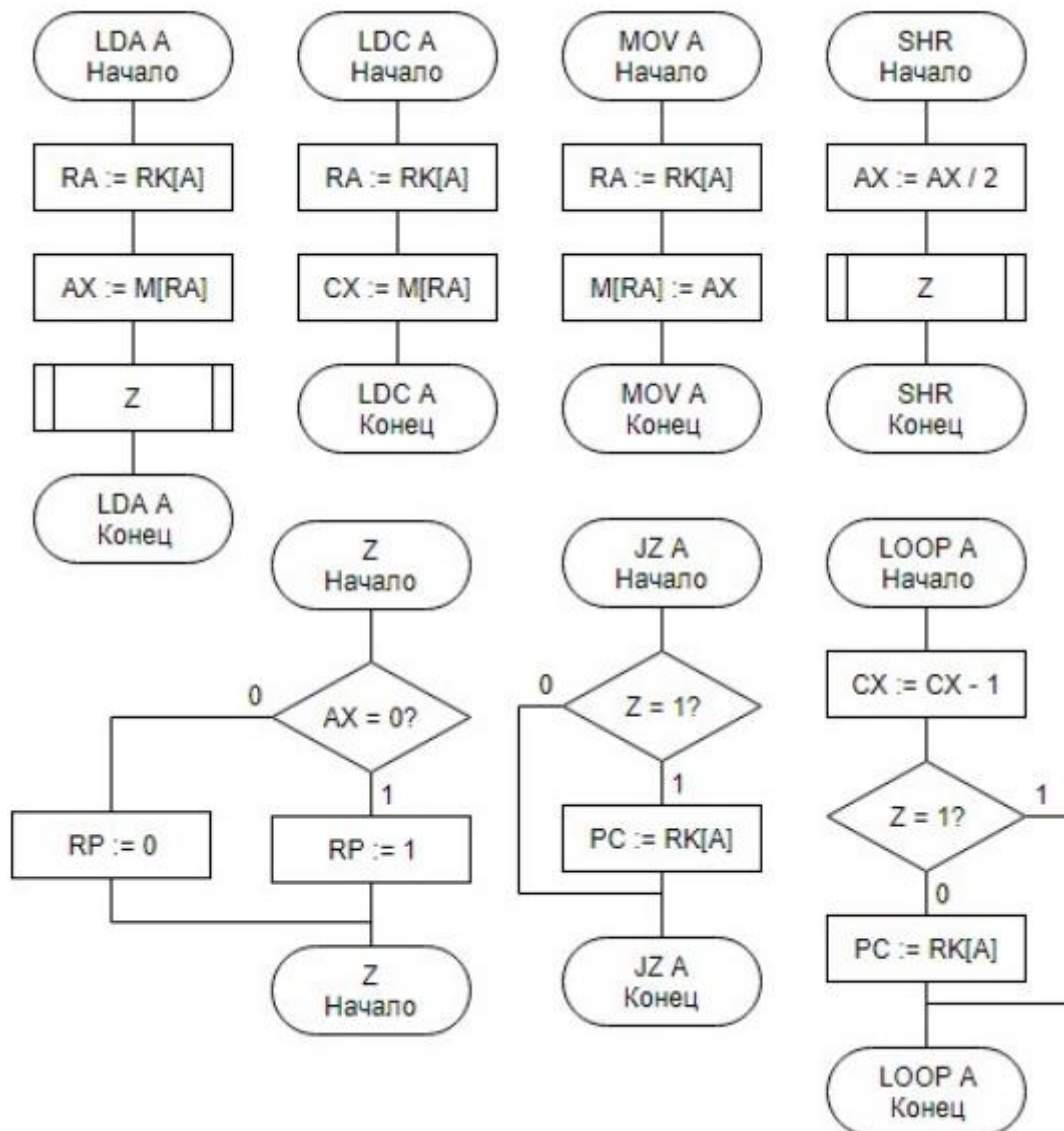
4.1 Структура учебной ЭВМ



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	AC		8	Регистр результата	
1			9		
2			A		
3			B		
4			C		
5			D	Регистр для данных из ЗУ	
6	PC		E	Регистр константы	
7	PR	Z	F	Счетчик адреса ЗУ	
RA	Адрес ЗУ		RQ		

5.2 Коды операций и начальные адреса подмикропрограмм

Мнемоника	Код операции	Адрес первой микрокоманды
LDA A	0x06	0x0F
CLM A	0x02	0x0C
ADD A	0x04	0x12
BEQ A	0x07	0x15
BR A	0x05	0x16
DEC A	0x03	0x18
HLT A	0x00	0x1C

5.3 Микропрограмма командного цикла (выборка команды и установка признаков)

№	МИ	РЗУ		Упр. АЛУ			Упр. ОЗУ			Шина	МИ	Упр. усл.			Упр. УУ		
МК	I8-0	A	B	CO	^OE	SC	^CS	^W	^EA	D11-0	I3-0	A	U	^CCE	CO	^RLD	^OE
00:	571	E	E	0	0	00	1	1	1	006	C	00	0	0	1	1	0
01:	533	0	E	0	0	00	1	1	1	001	9	00	0	0	1	1	0
02:	143	0	1	0	0	00	1	1	0	000	E	00	0	0	1	1	0
03:	337	0	6	0	1	00	0	1	1	000	E	00	0	0	1	1	0
04:	203	6	6	1	0	00	1	1	0	000	E	00	0	0	1	1	0
05:	245	E	F	0	1	00	0	1	1	000	2	00	0	0	1	1	0
06:	343	0	7	0	1	00	1	1	1	000	E	00	0	0	1	1	0
07:	133	0	0	0	1	00	1	1	1	00A	3	00	1	0	1	1	0
08:	133	0	0	0	1	00	1	1	1	004	3	01	0	0	1	1	0
09:	503	0	7	1	1	01	1	1	1	004	3	00	0	1	1	1	0
0A:	303	0	7	1	1	00	1	1	1	004	3	00	0	1	1	1	0

5.4 Микропрограмма командного цикла (выполнение операций)

№	МИ	РЗУ		Упр. АЛУ			Упр. ОЗУ			Шина	МИ	Упр. усл.			Упр. УУ		
МК	I8-0	A	B	CO	^OE	SC	^CS	^W	^EA	D11-0	I3-0	A	U	^CCE	CO	^RLD	^OE
0B:	---	-	-	-	-	--	-	-	-	---	-	--	-	-	-	-	-
0C:	133	0	F	0	0	00	1	1	0	000	E	00	0	0	1	1	0
0D:	143	0	F	0	0	00	0	0	1	004	3	00	0	1	1	1	0
0E:	---	-	-	-	-	--	-	-	-	---	-	--	-	-	-	-	-
0F:	133	0	F	0	0	00	1	1	0	000	E	00	0	0	1	1	0
10:	337	0	0	0	1	00	0	1	1	006	3	00	0	1	1	1	0
11:	---	-	-	-	-	--	-	-	-	---	-	--	-	-	-	-	-
12:	133	0	F	0	0	00	1	1	0	000	E	00	0	0	1	1	0
13:	305	8	8	0	1	00	0	1	1	004	3	00	0	1	1	1	0
14:	---	-	-	-	-	--	-	-	-	---	-	--	-	-	-	-	-
15:	113	0	7	0	1	00	1	1	1	004	3	00	0	0	1	1	0
16:	334	F	6	0	1	00	1	1	1	004	3	00	0	1	1	1	0
17:	---	-	-	-	-	--	-	-	-	---	-	--	-	-	-	-	-
18:	133	0	F	0	0	00	1	1	0	000	E	00	0	0	1	1	0
19:	337	0	0	0	1	00	0	1	1	000	E	00	0	0	1	1	0
1A:	313	0	0	0	0	00	0	0	1	006	3	00	0	1	1	1	0
1B:	---	-	-	-	-	--	-	-	-	---	-	--	-	-	-	-	-
1C:	133	0	F	0	0	00	1	1	0	000	E	00	0	0	1	1	0
1D:	134	8	0	0	0	00	0	0	1	000	E	00	0	0	1	1	0
1E:	343	0	8	0	1	00	1	1	1	000	E	00	0	0	1	1	0
1F:	334	F	6	0	1	00	1	1	1	000	3	00	0	1	1	1	0

6. Расчет производительности и быстродействия

Допустим, что среднее число циклов в программе $C = 5$, вероятность обнуления операнда до окончания цикла $p_1 = 0.005$, вероятность того, что $X = 0$, $p_2 = 0.004$; 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	2.3	7.5	17.25	0.097	0.7275
CLM	0.996	3	2.988	0.042	0.126
ADD	5.994	2.5	14.985	0.290	0.725
BEQ	7.996	2	15.992	0.256	0.512
BR	6.442	1.5	9.663	0.032	0.048
DEC	5.994	1.5	8.991	0.290	0.435
HLT	1	1	1	0.032	0.032
	$H = 30.772$		$R = 70.869$		$r = 2.6$

Примем $t = 100$ нс. Тогда среднее время выполнения команды составит $T_v = rt = 260$ нс, а быстродействие $V = 1 / T_v = 3.84$ млн. команд/сек. Аналогично среднее время решения задачи $T_w = R_t = 7$ мкс., производительность - $W = 142$ тыс. задач в секунду.

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.