

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 №6 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 с системой прерываний; разработать структурную схему и алгоритм работы ЭВМ; составить и отладить микропрограмму командного цикла ЭВМ.

Разработать микропрограмму, выполняющую умножение XY (X и Y – целые числа от 0 до 255) путем Y кратного суммирования множимого X .

2. Определение структуры и программирование

2.1 Форматы данных

X и Y изменяются в пределах от 0 до 255, поэтому любое число можно представить 16 разрядным двоичным кодом без знака

2.2 Программно-доступные регистры

ЭВМ имеет девять программно-доступных регистров: шесть регистров общего назначения ($r0-r5$), программный счетчик – IP ($r6$), регистр признаков – $FLAGS$ ($r7$), содержащий разряд признака нуля (Z), регистр указателя стека – SP ($r8$), регистр адреса таблицы прерываний – ITR ($r9$), а также 8-разрядный регистр маски RM .

2.3 Система команд

Название	Мнемоника	Описание	Изменение признака Z
Суммирование	ADD r r*	$r := r + r^*, IP := IP + 1$	+
Вычитание	SUB r r*	$r := r - r^*, IP := IP + 1$	+
Добавление C	AD r C	$r := r + C, IP := IP + 1$	+
Вычитание C	SB r C	$r := r - C, IP := IP + 1$	+
Чтение в регистр	LD r A	$r := M[A], IP := IP + 1$	-
Запись регистра	MV r A	$M[A] := r, IP := IP + 1$	-
Чтение в регистр с индексацией	LDI r, r*	$r := M[r^*], IP := IP + 1$	-
Запись в стек	PUSH r (SP)	$M[SP] := r, SP := SP - 1, IP := IP + 1$	-
Чтение из стека	POP r (SP)	$SP := SP + 1, r := M[SP], IP := IP + 1$	-
Переход	JMP A	$IP := A$	-
Переход, если нуль	JZ A	Если $Z = 1$, то $IP := A$, иначе $IP := IP + 1$	-
Обращение к подпрограмме	CALL (SP) A	$M[SP] := IP, SP := SP - 1, IP := IP + 1$	-
Загрузка маски	LM A	$RM := M[A], IP := IP + 1$	-
Сдвиг вправо логический	SHR r r*	$r := r^* / 2, IP := IP + 1$	+
Останов	HLT A	$IP := A$, останов	-
Возврат из прерывающей программы	IRET	$SP := SP + 1, RM := M[SP], SP := SP + 1, IP := M[SP]$	-

В описании системы команд приняты следующие обозначения:

- $r, r^* \in \{r0, r1, \dots, r8\}$ – программно-доступные регистры: регистр r^* является источником данных, а регистр r – приемником результата, но может также служить источником второго операнда
- $M[A]$ – ячейка памяти с адресом A
- Знак "+" в описании признаков означает, что устанавливается новое значение признака по результату выполнения команды, а знак "-" свидетельствует о сохранении старого значения признака

3. Кодирование программы и распределение памяти программ и данных

3.1 Коды операций

Название	Мнемоника	Код операции
Чтение в регистр	LD	0x0B
Запись регистра	MV	0x0C
Вычитание	SUB	0x02
Сложение	ADD	0x01
Прибавление C	AD	0x09
Вычитание C	SB	0x0A
Чтение в регистр с индексацией	LDI	0x10
Запись в стек	PUSH	0x03
Чтение из стека	POP	0x04
Переход если нуль	BEQ	0x06
Переход	BR	0x05
Переход если минус	BMI	0x07
Обращение к подпрограмме	CALL	0x08
Возврат из прерывающей программы	IRET	0x12
Загрузка маски	LM	0x11
Останов	HLT	0x00

3.2 Распределение памяти

Служебная программа и таблица прерываний

ОЗУ		
00:	00000000000000101	SA.....
01:	0000000011111111	MS.....
02:	0000000011111111	ASP.....
03:	00000000000001010	ATI.....
04:	0000000000100011	AP.....
05:	0001000100000001	LM AMS.....
06:	1011100000000010	LD RSP ASP..
07:	1011100100000011	LD RATI ATI.
08:	1011011000000100	LD PC AP....
09:	0000000000000000
0A:	0000000001000011	AP0.....
0B:	0000000001100000	AP1.....
0C:	0000000001100010	AP2.....
0D:	0000000001100100	AP3.....
0E:	0000000001100110	AP4.....
0F:	0000000001101000	AP5.....
10:	0000000001101010	AP6.....
11:	0000000001101100	AP7.....

Маски и основная программа

13:	0000000011111111	M0.....
14:	0000000000000000	M1.....
15:	0000000000000000	M2.....
16:	0000000000000000	M3.....
17:	0000000000000000	M4.....
18:	0000000000000000	M5.....
19:	0000000000000000	M6.....
1A:	0000000000000000	M7.....
1B:	0000000000000000
1C:	0000000000000000
1D:	0000000000000000
1E:	0000000000000000
1F:	0000000000000000
20:	0000000000101110	AM.....
21:	0000000000000011	N.....
22:	0000000000001000	S.....
23:	0001000100010010	LM AMP.....
24:	1011010100100000	LD R5 AAM....
25:	1011010000100001	LD R4 AN.....
26:	0000001000110011	SUB R3 R3.....
27:	0001000000010101	LDI R1 (R5)+..
28:	0000000100110001	ADD R3 R1.....
29:	1010010000000001	SB R4 1.....
2A:	0000011000101100	BEQ.....
2B:	0000010100100111	BR.....
2C:	1100001100100010	MV R3 AS.....
2D:	0000000000000101	HLT.....
2E:	0000000000000011	T1.....
2F:	0000000000000000	T2.....
30:	0000000000000101	T3.....

Прерывающая программа P0

```

40: 00000000000000111 |X.....
41: 0000000000000010 |Y.....
42: 0000000000000000 |Z.....
43: 0001000100010011 |LM AM0.....
44: 0000001101110000 |PUSH RP.....
45: 0000001100000000 |PUSH R0.....
46: 0000001100010000 |PUSH R1.....
47: 0000001100100000 |PUSH R2.....
48: 0000001000100010 |R2:=0 (Z).....
49: 1011000001000000 |LD R0 AX.....
4A: 0000011001010001 |BEQ X=0?.....
4B: 1011000101000001 |LD R1 AY.....
4C: 0000011001010001 |BEQ Y=0?.....
4D: 0000000100100000 |ADD R2 R0 (Z:=Z + X)..
4E: 1010000100000001 |R1 - 1 (Y:=Y-1).....
4F: 0000011001010001 |BEQ (Y = 0?).....
50: 0000010101001101 |BR.....
51: 1100001001000010 |MV R2 AZ.....
52: 0000010000100000 |POP R2.....
53: 0000010000010000 |POP R1.....
54: 0000010000000000 |POP R0.....
55: 0000010001110000 |POP RP.....
56: 0001001000000000 |IRET.....

```

Прерывающие программы P1 – P7

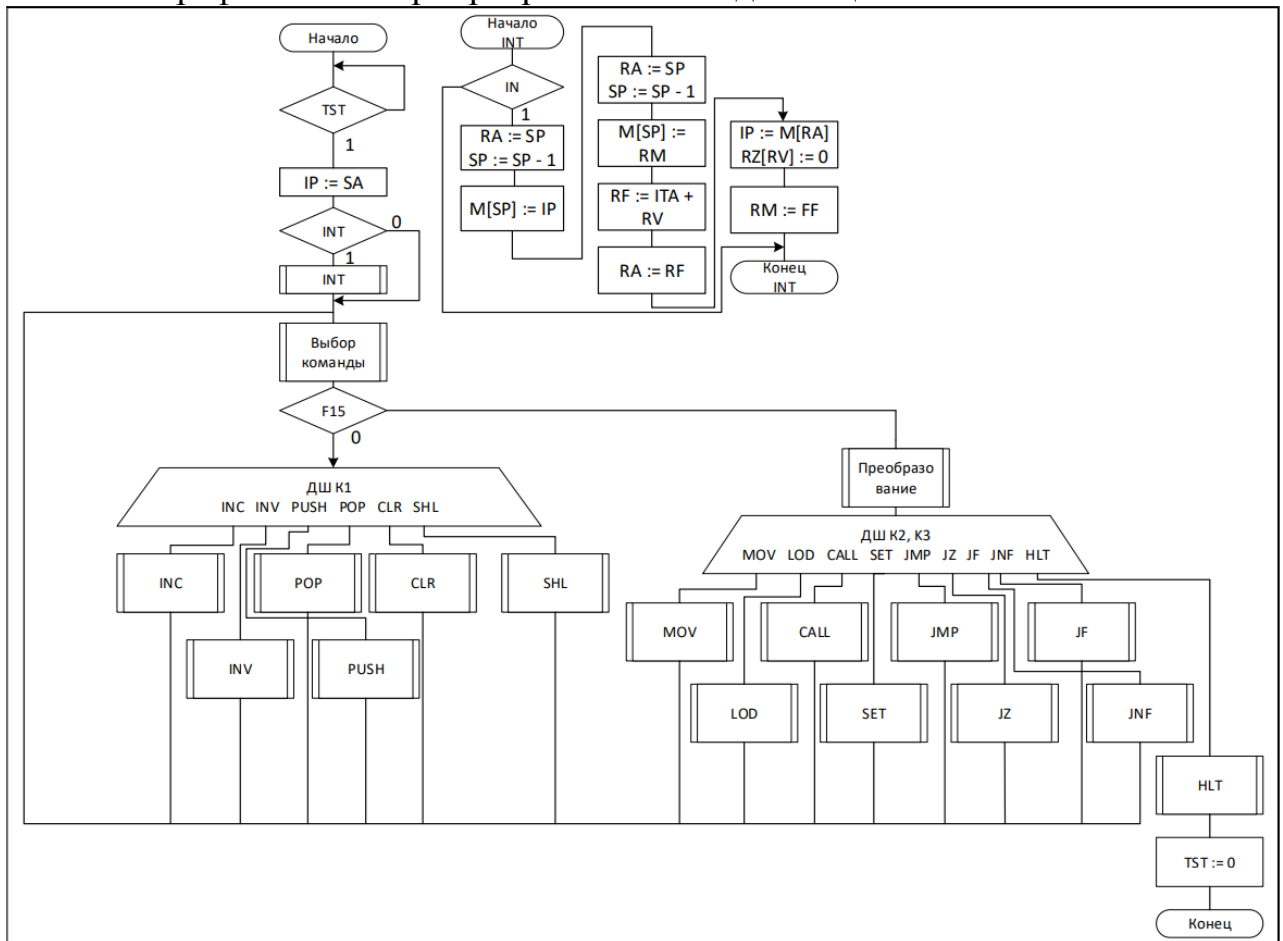
```

60: 0001000100010100 |LM AM1.....
61: 0001001000000000 |IRET.....
62: 0001000100010101 |LM AM2.....
63: 0001001000000000 |IRET.....
64: 0001000100010110 |LM AM3.....
65: 0000000000000000 |IRET.....
66: 0001000100010111 |LM AM4.....
67: 0001001000000000 |IRET.....
68: 0001000100011000 |LM AM5.....
69: 0001001000000000 |IRET.....
6A: 0001000100011001 |LM AM6.....
6B: 0001001000000000 |IRET.....
6C: 0001000100011010 |LM AM7.....
6D: 0001001000000000 |IRET.....

```

4. Разработка структуры и алгоритма работы

4.1 Граф-схема микропрограммы командного цикла



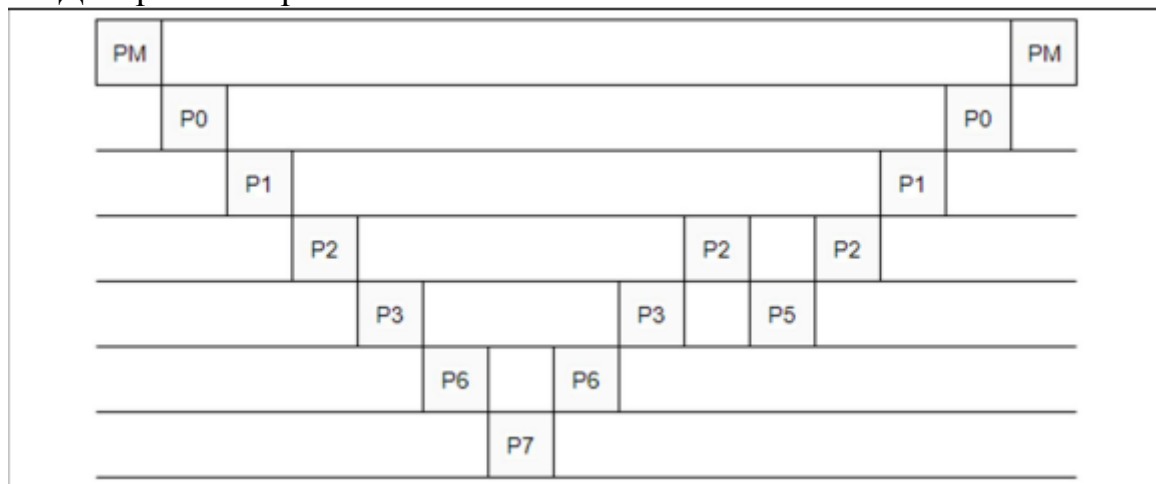
4.2 Микропрограмма командного цикла (выборка команды и установка признаков)

00:	571	E E	0	0	00	1	1	1	006	C	000	0	0	1	1	0	00	1	1	0	1	0	1
01:	533	0 E	0	0	00	1	1	1	001	9	000	0	0	1	1	0	00	1	1	0	1	1	1
02:	143	0 6	0	0	00	1	1	0	000	E	000	0	0	1	1	0	00	1	1	0	1	1	1
03:	337	0 6	0	1	00	0	1	1	000	E	000	0	0	1	1	0	00	1	1	0	1	1	1
04:	203	6 6	1	0	00	1	1	0	03F	3	100	1	0	1	1	0	00	1	1	0	1	1	1
05:	337	0 C	0	1	00	0	1	1	007	3	001	1	0	1	1	0	00	0	1	0	1	1	1
06:	345	E F	0	1	00	1	1	1	000	2	000	0	0	1	1	0	00	1	0	0	1	1	1
07:	345	E F	0	1	00	1	1	1	003	C	000	0	0	1	1	0	00	1	0	0	1	1	1
08:	533	0 C	0	0	00	1	1	1	008	9	000	0	0	1	1	0	00	1	1	0	1	1	1
09:	131	C C	0	0	00	1	1	1	000	2	000	0	0	1	1	0	00	0	1	0	1	1	1
0A:	343	0 7	0	1	00	1	1	1	000	E	000	0	0	1	1	0	00	1	1	0	1	1	1
0B:	133	0 0	0	1	00	1	1	1	00E	3	000	1	0	1	1	0	01	1	1	0	1	1	1
0C:	133	0 0	0	1	00	1	1	1	004	3	001	0	0	1	1	0	01	1	1	0	1	1	1
0D:	503	0 7	1	1	01	1	1	1	004	3	000	0	1	1	1	0	00	1	1	0	1	1	1
0E:	303	0 7	1	1	00	1	1	1	004	3	000	0	1	1	1	0	00	1	1	0	1	1	1

5.4 Микропрограмма командного цикла (IRET, INT)

37:	303	8 8	1	0	00	1	1	0	000	E	000	0	1	1	1	0	00	1	1	0	1	1	1
38:	133	0 0	0	1	00	0	1	1	000	E	000	0	1	1	1	0	00	1	1	0	1	0	1
39:	303	8 8	1	0	00	1	1	0	000	E	000	0	1	1	1	0	00	1	1	0	1	1	1
3A:	337	0 6	0	1	00	0	1	1	004	3	000	0	1	1	1	0	00	1	1	0	1	1	1
3B:
3C:	133	0 F	0	0	00	1	1	0	000	E	000	0	1	1	1	0	00	1	1	0	1	1	1
3D:	133	0 F	0	1	00	0	1	1	004	3	000	0	1	1	1	0	00	0	1	0	1	0	1
3E:
3F:	213	8 8	0	0	00	1	1	0	000	E	000	0	1	1	1	0	00	1	1	0	1	1	1
40:	113	0 6	0	0	00	0	0	1	000	E	000	0	1	1	1	0	00	1	1	0	1	1	1
41:	213	8 8	0	0	00	1	1	0	000	E	000	0	1	1	1	0	00	1	1	0	1	1	1
42:	133	0 0	0	1	00	0	0	1	000	E	000	0	1	1	1	0	00	1	1	0	1	1	0
43:	305	9 F	0	1	00	1	1	1	000	E	000	0	1	1	1	0	00	1	1	0	0	1	1
44:	133	0 F	0	0	00	1	1	0	000	E	000	0	1	1	1	0	00	1	1	1	1	1	1
45:	337	0 6	0	1	00	0	1	1	000	E	000	0	1	1	1	0	00	1	1	0	1	1	1
46:	171	0 0	0	0	00	1	1	1	004	3	000	0	1	1	1	0	00	1	1	1	1	0	1

5.5 Диаграмма переключения



6. Вывод

В ходе лабораторной работы была разработана и изучена учебная ЭВМ. В ходе выполнения лабораторной работы была изучена система и добавлена система прерываний. Добавленная система прерываний может получать 8 запросов на прерывание. Запросы имеют приоритет, выражаемый позицией

запроса в регистре запросов от 7 до 0, чем ниже значение, тем выше приоритет. Для запрета или разрешения обработки запроса существует маска. Так, если некоторый бит маски равен 1, то соответствующий запрос (располагающийся в соответствующем бите регистра запросов) обработан не будет. Таким образом были запрещены любые прерывания в служебной программе. После обработки бит обработанного запроса устанавливается в 0 для устранения возможности повторной обработки запроса