

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 №6 по дисциплине
«Организация ЭВМ и систем»
Задание 8

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

Киров 2017

1 Задание

Определить архитектуру ЭВМ с системой прерываний; разработать структурную схему и алгоритм работы ЭВМ; составить и отладить микропрограмму командного цикла ЭВМ.

Разработать микропрограмму, вычисляющую логический сдвиг числа X на N разрядов.

2 Определение архитектуры и программирование

2.1 Формат данных

X и N изменяются в пределах 0...65535, 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

3.2 Коды операций

Название	Мнемоника	Код операции
Суммирование	ADD	0x01
Вычитание	SUB	0x02
Добавление C	AD	0x9
Вычитание C	SB	0xA
Чтение в регистр	LD	0xB
Запись регистра	MV	0xC
Чтение в регистр с индексацией	LDI	0x0E
Запись в стек	PUSH	0x07
Чтение из стека	POP	0x06
Переход	JMP	0x03
Переход, если нуль	JZ	0x04
Обращение к подпрограмме	CALL	0xD
Возврат из подпрограммы	IRET	0x05
Сдвиг вправо логический	SHR	0x08
Останов	HLT	0x00
Загрузка маски	LM	0xF

3.3 Распределение памяти программ и данных

00:	0000000000000110	SIP
01:	0000000011111111	MS
02:	0000000011111111	BP
03:	0000000000001011	IT
04:	0000000000100001	PIP
05:	0000000000000000	
06:	1111000000000001	LM MS.....	
07:	1011100000000010	LD SP BP.....	
08:	1011100100000011	LD ITR IT.....	
09:	1011011000000100	LD IP PIP.....	
0A:	0000000000000000	
0B:	0000000000110100	P0IP (IT)	
0C:	0000000001001010	P1IP.....	
0D:	0000000001001101	P2IP.....	
0E:	0000000001010000	P3IP.....	
0F:	0000000001010011	P4IP.....	
10:	0000000001010110	P5IP.....	
11:	0000000001011001	P6IP.....	
12:	0000000001011100	P7IP.....	

Рисунок 1 – Служебная программа и таблица прерываний

14:	0000000011111110	MP.....
15:	0000000011111101	M0.....
16:	0000000011001011	M1.....
17:	0000000011010111	M2.....
18:	0000000010101111	M3.....
19:	0000000010011101	M4.....
1A:	0000000010111101	M5.....
1B:	0000000011111101	M6.....
1C:	0000000011110011	M7.....
1D:	0000000000000000
1E:	0000000000101110	ARRAY.....
1F:	0000000000000011	N.....
20:	0000000000000110	SUM.....
21:	1111000000010100	LM MP.....
22:	1011010100011110	LD R5 ARRAY.....
23:	1011010000011111	LD R4 N.....
24:	0000001000110011	SUB R3 R3.....
25:	0000111000010101	LDI R1 R5 (L2)....
26:	1001010100000001	AD R5 01.....
27:	0000000100110001	ADD R3 R1.....
28:	1010010000000001	SB R4 01.....
29:	0000010000101011	JZ L1.....
2A:	0000001100100101	JMP L2.....
2B:	1100001100100000	MU R3 SUM (L1)....
2C:	0000000000000000	HLT SIP.....
2D:	0000000000000000
2E:	0000000000000001	T0.....
2F:	0000000000000010	T1.....
30:	0000000000000011	T2.....

Рисунок 2 – Маски и основная программа

32:	0000001111101111	X.....
33:	0000000000000100	N.....
34:	1111000000010101	LM M0 (P0).....
35:	0000011101110000	PUSH FLAGS.....
36:	0000011100010000	PUSH R1.....
37:	0000011100100000	PUSH R2.....
38:	1011000100110010	LD R1 X.....
39:	1011001000110011	LD R2 N.....
3A:	1001001000000000	AD R2 00.....
3B:	0000010001000001	JZ L3.....
3C:	0000100000010001	SHR R1 R1 (L4)....
3D:	0000010001000001	JZ L3.....
3E:	1010001000000001	SB R2 01.....
3F:	0000010001000001	JZ L3.....
40:	0000001100111100	JMP L4.....
41:	1100000101000110	MV R1 RES (L3)....
42:	0000011000100000	POP R2.....
43:	0000011000010000	POP R1.....
44:	0000011001110000	POP FLAGS.....
45:	0000010100000000	IRET.....
46:	0000000000000000	RES.....

Рисунок 3 – Прерывающая программа P0

4A:	1111000000010110	LM M1.....
4B:	0000010100000000	IRET.....
4C:	0000000000000000
4D:	1111000000010111	LM M2.....
4E:	0000010100000000	IRET.....
4F:	0000000000000000
50:	1111000000011000	LM M3.....
51:	0000010100000000	IRET.....
52:	0000000000000000
53:	1111000000011001	LM M4.....
54:	0000010100000000	IRET.....
55:	0000000000000000
56:	1111000000011010	LM M5.....
57:	0000010100000000	IRET.....
58:	0000000000000000
59:	1111000000011011	LM M6.....
5A:	0000010100000000	IRET.....
5B:	0000000000000000
5C:	1111000000011100	LM M7.....
5D:	0000010100000000	IRET.....

Рисунок 4 – Прерывающие программы P1...P7

3.4 Разработка структуры и алгоритма работы

Алгоритм работы ЭВМ представлен на рисунке 3 в виде укрупненной граф-схемы микропрограммы командного цикла.

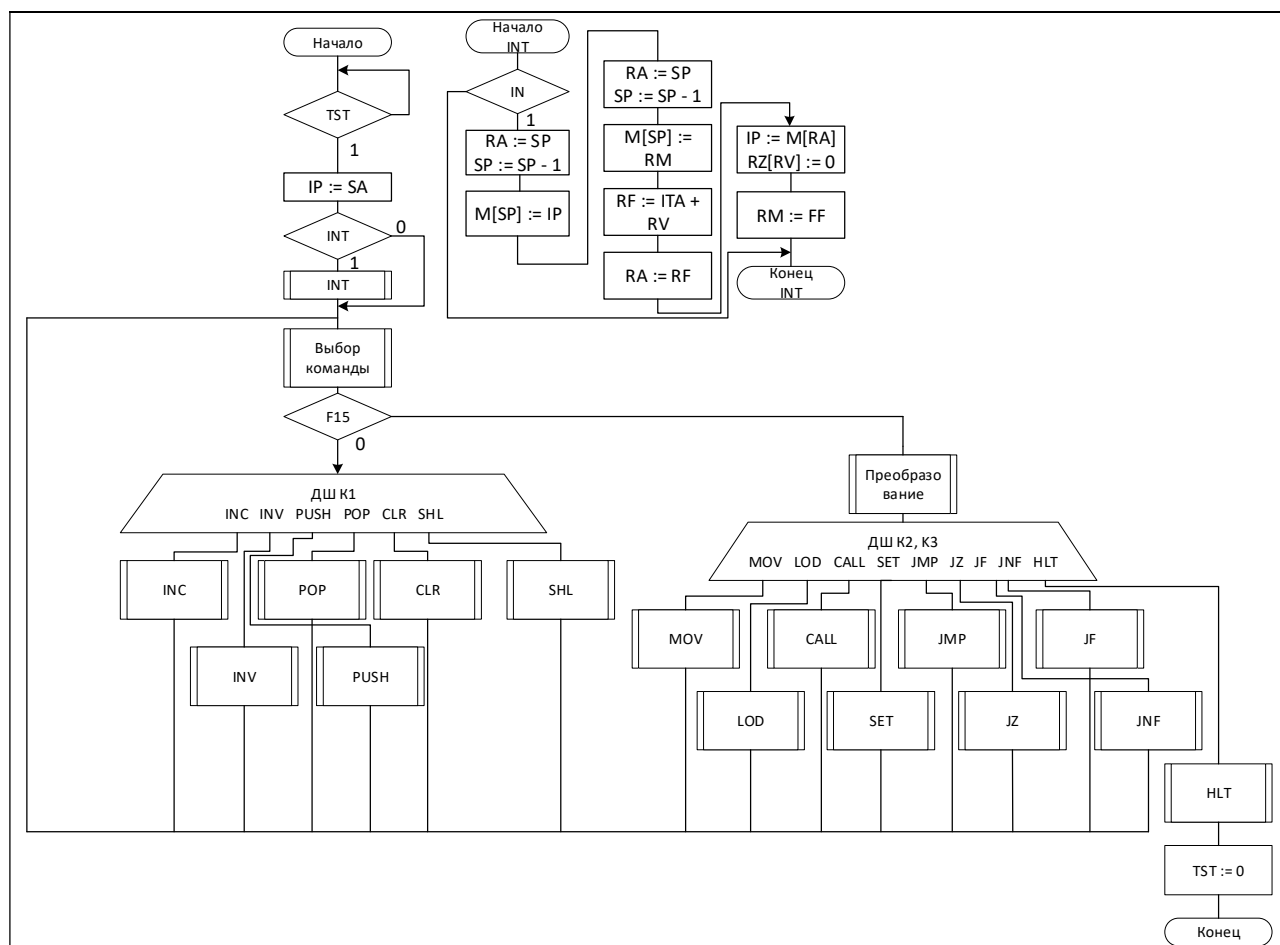


Рисунок 5 – Граф-схема микропрограммного командного цикла учебной ЭВМ

3.5 Форматы команд

Формат				Пример
0	КОП	r^*	r	PUSH
0	КОП	A		JZ
1	КОП	r^*	A	CALL
1	КОП	r^*	imm16	AD

3.6 Распределение регистров

	PЗУ (R0 – R7)		PЗУ (R8 – R15)
0	r0	8	r8 (SP)
1	r1	9	r9 (ITA)
2	r2	10	
3	r3	11	
4	r4	12	

	РЗУ (R0 – R7)			РЗУ (R8 – R15)	
5	r5		13	Буферный регистр команд	
6	r6 (IP)		14	Регистр константы	
7	r7 (FLAGS)	Z	15	Счетчик адреса ЗУ RK[A]	
RA	Адрес ЗУ		RQ		

3.7 Коды операций

Название	Мнемоника	Код операции
Суммирование	ADD	0x01
Вычитание	SUB	0x02
Добавление C	AD	0x9
Вычитание C	SB	0xA
Чтение в регистр	LD	0xB
Запись регистра	MV	0xC
Чтение в регистр с индексацией	LDI	0x0E
Запись в стек	PUSH	0x07
Чтение из стека	POP	0x06
Переход	JMP	0x03
Переход, если нуль	JZ	0x04
Обращение к подпрограмме	CALL	0xD
Возврат из подпрограммы	IRET	0x05
Сдвиг вправо логический	SHR	0x08
Останов	HLT	0x00
Загрузка маски	LM	0xF

3.8 Микропрограмма командного цикла

00:	571	E	E	0	0	00	1	1	1	006	C	000	0	0	1	1	0	00	1
01:	533	0	E	0	0	00	1	1	1	001	9	000	0	0	1	1	0	00	1
02:	143	0	6	0	0	00	1	1	0	000	E	000	0	0	1	1	0	00	1
03:	337	0	6	0	1	00	0	1	1	000	E	000	0	0	1	1	0	00	1
04:	133	0	0	0	1	00	1	1	1	040	3	100	1	0	1	1	0	00	1
05:	203	6	6	1	0	00	1	1	0	000	E	000	0	0	1	1	0	00	1
06:	337	0	C	0	1	00	0	1	1	008	3	001	1	0	1	1	0	00	0
07:	345	E	F	0	1	00	1	1	1	000	2	000	0	0	1	1	0	00	1
08:	345	E	F	0	1	00	1	1	1	003	C	000	0	0	1	1	0	00	1
09:	533	0	C	0	0	00	1	1	1	009	9	000	0	0	1	1	0	00	1
0A:	131	C	C	0	0	00	1	1	1	000	2	000	0	0	1	1	0	00	0
0B:	343	0	7	0	1	00	1	1	1	000	E	000	0	0	1	1	0	00	1
0C:	133	0	0	0	1	00	1	1	1	004	3	000	0	0	0	0	0	01	1
0D:	303	0	7	1	1	00	1	1	1	004	3	000	0	1	1	1	0	00	1

Рисунок 6 – Микропрограмма командного цикла

34:	303	8	8	1	0	00	1	1	0	000	E	000	0	0	1	1	0	00	1
35:	337	0	A	0	1	00	0	1	1	000	E	000	0	0	1	1	0	00	1
36:	133	0	A	0	0	00	1	1	1	000	E	000	0	0	1	1	0	00	1
37:	303	8	8	1	0	00	1	1	0	000	E	000	0	0	1	1	0	00	1
38:	337	0	6	0	1	00	0	1	1	004	3	000	0	1	1	1	0	00	1
40:	213	8	8	0	0	00	1	1	0	000	E	000	0	0	1	1	0	00	1
41:	133	6	6	0	0	00	0	0	1	000	E	000	0	0	1	1	0	00	1
42:	213	8	8	0	0	00	1	1	0	000	E	000	0	0	1	1	0	00	1
43:	133	0	0	0	1	00	0	0	1	000	E	000	0	0	1	1	0	00	1
44:	305	9	F	0	1	00	1	1	1	000	E	000	0	0	1	1	0	00	1
45:	133	F	F	0	0	00	1	1	0	000	E	000	0	0	1	1	0	00	1
46:	337	6	6	0	1	00	0	1	1	000	E	000	0	0	1	1	0	00	1
47:	171	A	A	0	0	00	1	1	1	005	3	000	0	1	1	1	0	00	1

Рисунок 7 – Подмикропрограммы операций (IRET и INT)

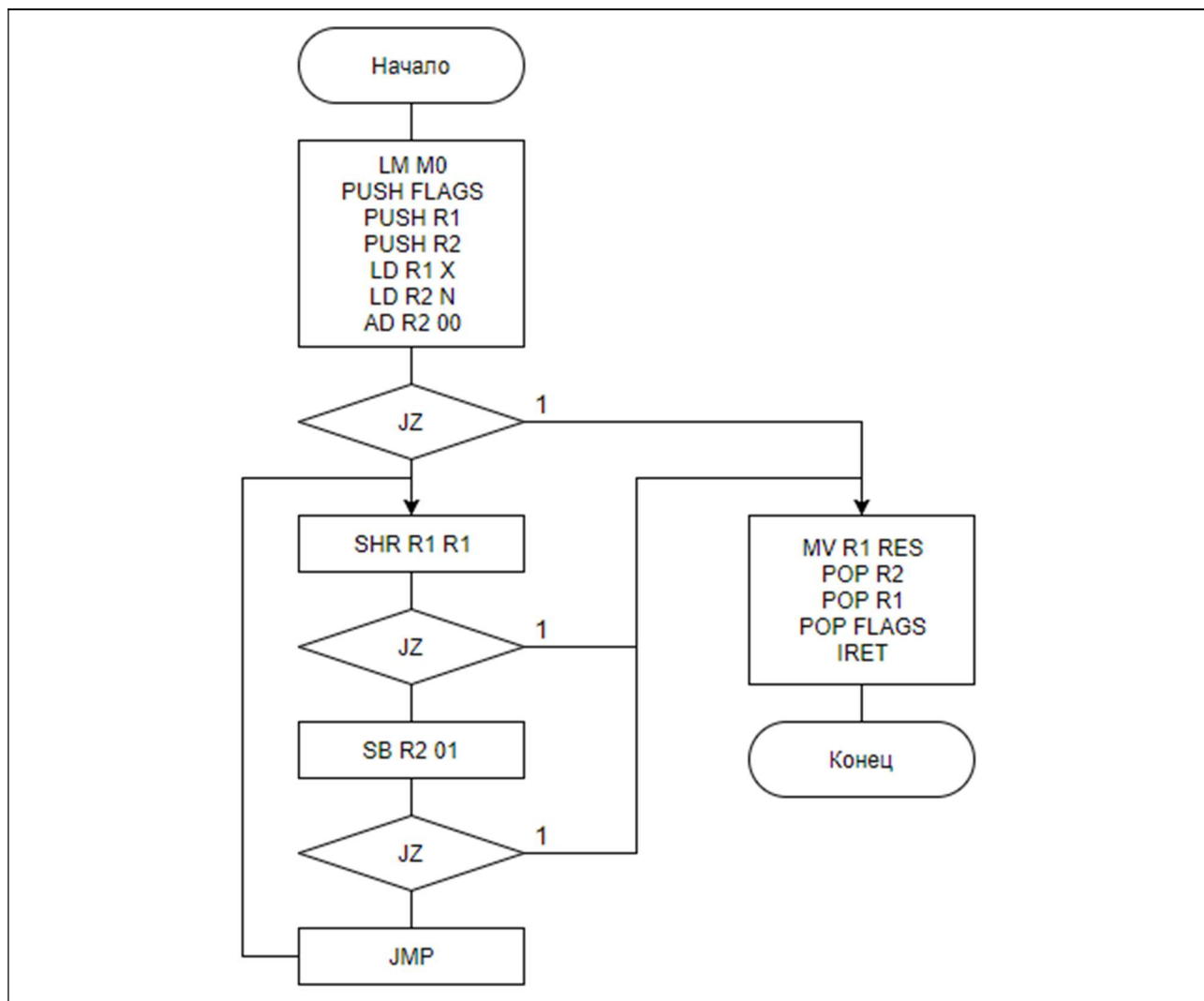


Рисунок 8 – Алгоритм решения задачи

4 Ввод и отладка микропрограммы командного цикла и программы решения задачи

Диаграмма переключения представлена на рисунке 9.

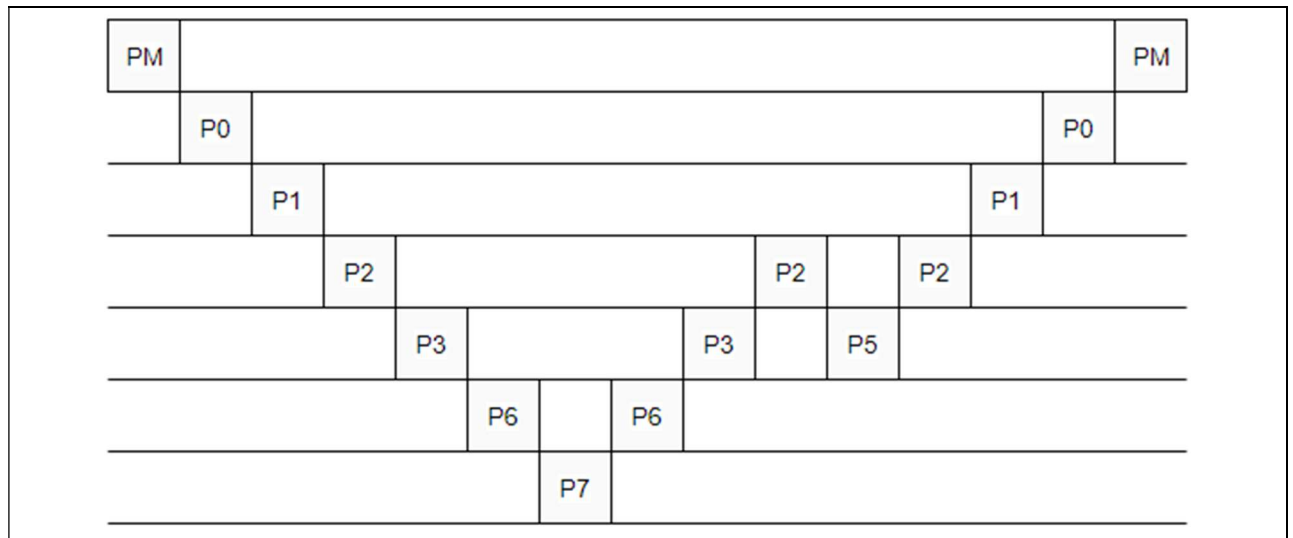


Рисунок 9 – Диаграмма переключений

5 Выводы

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.