

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

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

1 Задание

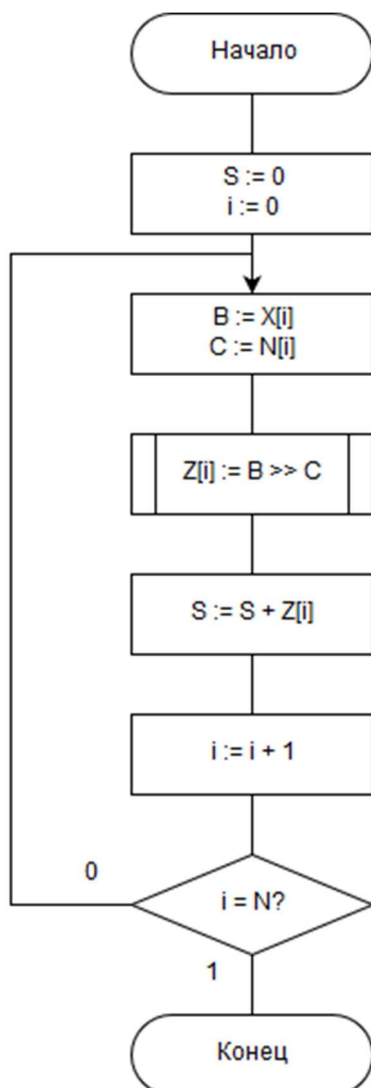
Определить архитектуру, разработать и отладить микропрограмму командного цикла ЭВМ, составить и выполнить программу вычисления суммы частных S :

$$S = \sum_{i=1}^N x_i \gg n_i,$$

где $x_i \gg n_i$ – логический сдвиг числа x_i вправо на n_i разрядов. Для сдвига чисел использовать подпрограмму на основе программы, составленной при выполнении предыдущей лабораторной работы. Обмен данными между программой и подпрограммой должен производиться через стек.

2 Определение архитектуры и программирование

2.1 Схема алгоритма



2.2 Форматы данных

X и N изменяются в пределах 0..65535, поэтому любое число можно представить 16-разрядным двоичным кодом без знака.

2.3 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), а также регистр указателя стека – SP (r8).

2.4 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	$M[SP] := IP, SP := SP - 1, IP := IP + 1$	-
Возврат из подпрограммы	RET (SP)	$SP := SP + 1, IP := M[SP]$	-
Сдвиг вправо логический	SHR r r*	$r := r^* / 2, IP := IP + 1$	+
Останов	HLT A	$IP := A$, останов	-

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

2.5 Программа

В программе приняты следующие обозначения:

- BP – адрес ячейки памяти, в которой находится адрес начала накопителя стека
- ARRAY – адрес ячейки памяти, в которой находится начальный адрес массива исходных данных $(x_0, n_0, x_1, n_1, \dots, x_n, n_n)$
- PROC – начальный адрес подпрограммы деления чисел нацело
- SUM – адрес ячейки, в которую записывается сумма S
- IPS – начальный адрес программы суммирования частных

	LD SP BP	Загрузка регистра указателя стека SP
	LD R5 ARRAY	Загрузка адреса массива ARRAY в регистр r5
	LD R4 N	Загрузка числа повторений цикла N в r4
	SUB R3 R3	Очистка регистра r3 для суммы S
m1:	LDI R1 [R5]	Чтение X_i в r1
	AD R5 01	Увеличение r5 на 1
	LDI R2 [R5]	Чтение N_i в r2
	AD R5 01	Увеличение r5 на 1
	CALL PROC	Обращение к подпрограмме по адресу PROC
	ADD R3 R1	Суммирование
	SB R4 01	Вычитание единицы из числа повторений цикла
	JZ m2	Если $N=0$, то переход на метку m2
	JMP m1	Переход на метку m1
m2:	MV R3 SUM	Запись суммы S адресу SUM
	HLT IP	Загрузка IP и останов

2.6 Распределение программно-доступных регистров ЭВМ

	Регистры ЭВМ	
r0		Не используется
r1	X_i	Число X_i (результат Z_i)
r2	N_i	Кол-во сдвигов N_i
r3	S	Сумма S
r4	N	Число повторений цикла N

	Регистры ЭВМ		
r5	ARRAY		Адрес массива
r6	IP		Программный счетчик
r7	FLAGS	Z	Регистр признаков
r8	SP		Регистр указателя стека

Для выполнения логических сдвигов чисел вправо основная программа обращается к подпрограмме логического сдвига чисел. Перед обращением число X_i помещается в регистр r1, количество сдвигов N_i – в регистр r2. Подпрограмма возвращает в основную программу результат Z_i с помощью регистра r1.

```

AD R2 00      Проверка Ni на равенство нулю
JZ m3         Если Ni = 0, то переход на метку m3
m4: SHR R1 R1  Логический сдвиг r1 на один разряд вправо
JZ m3         Если r1 = 0, то переход на метку m3
SB R2 01      Уменьшение кол-ва сдвигов Ni на 1
JZ m3         Если Ni = 0, то переход на метку m3
JMP m4        Переход на метку m4
m3: RET       Возврат из подпрограммы

```

3 Кодирование программы и распределение памяти программ и данных

3.1 Форматы команд

Команды ЭВМ имеют четыре формата и в зависимости от признака формата (Ф) и кода операции (К) делятся на четыре группы.

Ф					
15	14..12	11..8	7..4	3..0	
0	K1		r	r*	ADD, SUB, LDI, PUSH, POP, SHR, RET
0	K2		A		JMP, JZ, HLT
1	K3	r	C		AD, SB
1	K4	r	A		LD, MV, CALL,

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

Название	Мнемоника	Код операции
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	RET	0x05
Сдвиг вправо логический	SHR	0x08
Останов	HLT	0x00

3.3 Распределение памяти

Адрес	Код	Мнемоника	Комментарии
-------	-----	-----------	-------------

Основная программа и данные

00	0010	IPS	Адрес начала программы
01	0040	BP	Начальный адрес стека
02	0006	ARRAY	Адрес начала массива
03	0004	N	Кол-во пар элементов в массиве
04	0000	SUM	Результат – сумма сдвигов
05			Пустая ячейка

Массив исходных чисел

06		X0	
07		Y0	
08		X1	
09		Y1	
0A		X2	
0B		Y2	
0C		X3	
0D		Y3	

Основная программа

10	B801	LD SP BP	Загрузка регистра указателя стека SP
11	B502	LS R5 ARRAY	Загрузка адреса массива ARRAY в r5
12	0403	LD R4 N	Загрузка числа повторений цикла N
13	0233	SUB R3 R3	Очистка регистра для суммы SUM

Адрес	Код	Мнемоника	Комментарии
14	0E15	LDI R1 [R5]	Чтение Xi в регистр r1
15	9501	AD R5 01	Прибавление единицы к r5
16	0E25	LDI R2 [R5]	Чтение Ni в регистр r2
17	9501	AD R5 01	Прибавление единицы к r5
18	D020	CALL PROC	Обращение к подпрограмме по адресу PROC
19	0131	ADD R3 R1	Суммирование
1A	A401	SB R4 01	Вычитание единицы из числа N
1B	041D	JZ M2	Если N = 0, то переход на метку m2
1C	0314	JMP M1	Переход на метку m3
1D	C304	MV R3 SUM	Запись суммы по адресу ARRAY
1E	0010	HLT IP	Загрузка IP и останов

Подпрограмма и стек

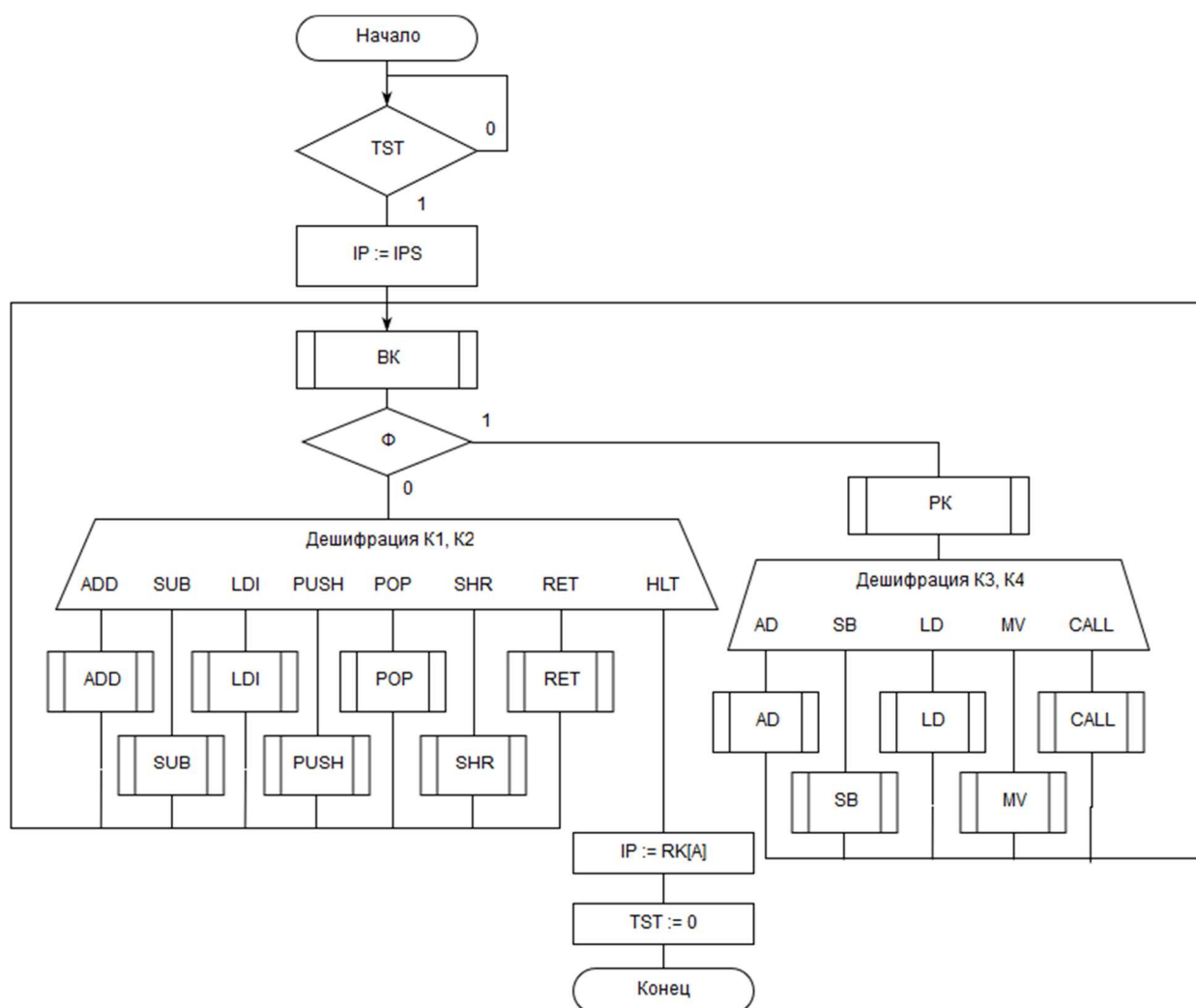
20	9200	AD R2 00	Проверка Ni на равенство нулю
21	0427	JZ M3	Если Ni = 0, то переход на метку m3
22	0811	SHR R1 R1	Логический сдвиг r1 на один разряд вправо
23	0427	JZ M3	Если r1 = 0, то переход на метку m3
24	A201	SB R2 01	Уменьшение кол-ва сдвигов Ni на 1
25	0427	JZ M3	Если Ni = 0, то переход на метку m3
26	0322	JMP M4	Переход на метку m4
27	0500	RET	Возврат из подпрограммы

4 Разработка алгоритма работы и микропрограммная реализация ЭВМ

4.1 Распределение регистров

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

4.2 Граф-схема микропрограммы командного цикла ЭВМ



4.3 Микропрограмма командного цикла (выборка команды и установка признаков)

№	МИ		РЗУ		Упр. АЛУ			Упр. ОЗУ			Шина	МИ	Упр. усл.			Упр. УУ			Упр. РК		
	I8-0	A	B	C0	OE	SC	CS	W	EA	D11-0	I3-I0	A	U	CCE	C0	RLD	OE	M	L	OE	
00	571	E	E	0	0	00	1	1	1	006	C	00	0	0	1	1	0	00	1	1	
RE:=0111111111111111; PA/ЦЦ:=6																					
01	533	0	E	0	0	00	1	1	1	001	9	00	0	0	1	1	0	00	1	1	
RE – сдвиг вправо; PA/ЦЦ:=PA/ЦЦ-1																					
02	143	0	6	0	0	00	1	1	0	000	E	00	0	0	1	1	0	00	1	1	
RA:=0																					
03	337	0	6	0	1	00	0	1	1	000	E	00	0	0	1	1	0	00	1	1	
R6:=IPS (IP:=IPS)																					
04	203	6	6	1	0	00	1	1	0	000	E	00	0	0	1	1	0	00	1	1	
RA:=R6; R6:=R6+1 (RA:=IP; IP:=IP+1)																					
05	337	0	C	0	1	00	0	1	1	007	3	01	1	0	1	1	0	00	0	1	
RK:=K; RC:=K																					

№	МИ	РЗУ		Упр. АЛУ			Упр. ОЗУ			Шина	МИ	Упр. усл.			Упр. УУ			Упр. РК		
	I8-0	A	B	C0	OE	SC	CS	W	EA	D11-0	I3-I0	A	U	CCE	C0	RLD	OE	M	L	OE
25	337	0	0	0	1	00	0	1	1	004	3	00	0	1	1	1	0	01	1	1
r:=M[SP]																				
26	113	0	7	0	1	00	1	1	1	004	3	00	0	0	1	1	0	00	1	1
Переход, если R7[0]=0 (JZ A)																				
27	334	F	6	0	1	00	1	1	1	004	3	00	0	1	1	1	0	00	1	1
R6:=RF (IP:=RK[A]) (JMP A)																				
28	213	8	8	0	0	00	1	1	0	000	E	00	0	1	1	1	0	00	1	1
RA:=SP, SP:=SP-1 (CALL A)																				
29	133	0	6	0	0	00	0	0	1	000	E	00	0	1	1	1	0	00	1	1
M[SP]:=r																				
2A	334	F	6	0	1	00	1	1	1	004	3	00	0	1	1	1	0	00	1	1
R6:=RF (IP:=RK[A])																				
2B	303	8	8	1	0	00	1	1	0	000	E	00	0	0	1	1	0	00	1	1
SP:=SP+1; RA:=SP (RET)																				
2C	337	0	6	0	1	00	0	1	1	004	3	00	0	1	1	1	0	00	1	1
IP:=M[SP]																				
2D	334	F	6	0	1	00	1	1	1	004	3	00	0	1	1	1	1	00	1	1
IP:=RK[A], останов (HLT A)																				
2E	534	0	0	0	1	00	1	1	1	00A	3	00	0	1	1	1	0	11	1	1
r:=r* / 2 (SHR r r*)																				
2F	134	0	0	0	0	00	1	1	0	000	E	00	0	0	1	1	0	10	1	1
RA:=r* (LDI r r*)																				
30	337	0	0	0	1	00	0	1	1	004	3	00	0	1	1	1	0	01	1	1
r:=M[r*]																				

5 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. По сравнению с предыдущей лабораторной работой, система команд была расширена. В дополнение к прямой были добавлены следующие виды адресации; регистровая, неявная регистровая преинкрементная и постдекрементная, непосредственная. Введение различных видов адресации усложнило командный цикл, однако сделало написание программы удобнее и понятнее для программиста.