

**Федеральное государственное автономное образовательное
учреждение высшего образования «Национальный исследовательский
университет ИТМО
Факультет программной инженерии и компьютерной техники**

**Основы Профессиональной Деятельности
Работа с БЭВМ
Лабораторная работа №4
Вариант 41408**

Выполнил:
Студент группы Р3116
Брагин Роман Андреевич

Проверила:
Блохина Елена Николаевна

г. Санкт-Петербург

2024 г

1.Текст исходной программы

Адрес	Код команды	Мнемоника	Комментарии
15B	0200	CLA	0 \square AC очистка аккумулятора
15C	EE19	ST IP + 19	Сохранение с прямой относительной адресацией AC \rightarrow IP + 19 загружаем 0 в 176 ячейку(результата)
15D	AE15	LD IP +15	Загрузка с прямой относительной адресацией (IP +15) \square AC; 173 ячейка (переменная Z) в AC
15E	0C00	PUSH	SP - 1 \square SP, AC \square SP; 0-1получаем 7FF мы записываем (значение переменной Z) из аккумулятора в ячейку 7FF
15F	D73A	CALL 73A	SP - 1 \square SP, IP \square SP, 73A \square IP; делаем SP-1получаем 7FE, записываем 7FE (из IP) в SP как значение, 73A \square IP вызов подпрограммы
160	0800	POP	SP \square AC, SP +1 \square SP; значение ячейки из 7FF f(Z) записываем в AC, SP(7FF+1) = 0;
161	4E14	ADD IP +14	Сложение с прямой относительной адресацией (IP +14) + AC \square AC; значение константы C1 + AC (f(Z)) \square AC (C1 +f(Z));
162	EE13	ST IP +13	Сохранение с прямой относительной адресацией AC \square IP +13; сохраняем значение (C1 +f(Z)) в ячейку 176
163	AE11	LD IP +11	Загрузка с прямой относительной адресацией (IP +11) \square AC; загружаем X в AC
164	0740	DEC	AC - 1 \square AC; из X в AC - 1 получаем X-1
165	0C00	PUSH	SP - 1 \square SP, AC \square SP; записываем X-1 в 7FF
166	D73A	CALL 73A	SP - 1 \square SP, IP \square SP; вызываем подпрограмму и делаем SP - 1 = 7FE
167	0800	POP	SP \square AC, SP +1 \square SP; (7FF (значение f(X-1)) \square AC, SP+1 = 0)
168	4E0D	ADD IP +0D	Сложение с прямой относительной адресацией (IP +0D) + AC \square AC; (f(Z)+C) из 176 + AC (f(X-1)) = AC(f(Z)+f(X-1) +C)
169	EEOC	ST IP +0C	Сохранение с прямой относительной адресацией AC \square IP +0C; AC(f(Z)+f(X-1) +C) \square в 176 ячейку
16A	AE09	LD IP + 9	Загрузка с прямой относительной адресацией (IP + 9) \square AC; загружаем Y в AC
16B	0740	DEC	AC - 1 \square AC; Y-1 в AC
16C	0C00	PUSH	SP - 1 \square SP, AC \square SP; SP - 1 = 7FF, Y-1 загружаем в ячейку 7FF
16D	D73A	CALL 73A	SP - 1 \square SP, IP \square (SP); вызываем подпрограмму и делаем SP = 7FE
16E	0800	POP	SP \square AC, SP +1 \square SP; загружаем f(Y-1) в AC
16F	0740	DEC	AC - 1 \square AC; делаем f(Y-1) - 1 в AC
170	6E05	SUB IP+5	AC - IP+5 \square AC; f(Y-1) - 1 - f(Z)+f(X-1) +C
171	EE04	ST IP +04	Сохранение с прямой относительной адресацией AC \square IP +04; загружаем результат f(Y) - 1 - f(Z)+f(X-1) +C
172	0100	HLT	Остановка; конец программы

Текст исходной ПП			
73A	AC01	LD &1	Загрузка с смещением относительно стека SP+1 \square AC; SP (7FE) + 1 = значение из ячейки 7FF(Z) \square AC
73B	F303	BPL IP +3	IF AC \geq 0, IP+3 \square IP; если Z, то идем в 73F иначе в 73C
73C	7E08	CMP IP + 8	Установить флаги по значению AC – (IP + 8); (Z – 00F0 (-240)) из 745 (z+240)
73D	F201	BMI IP + 1	IF N == 1, IP + 1 \square IP; если число $>$ -240, то в 73F
73E	CE03	JUMP IP + 3	IP + 3 \square IP; идем в 742
73F	4C01	ADD &1	AC + (SP+1) \square AC; (значение Z + значение Z) = 2Z \square AC
740	6E05	SUB IP+5	AC – (IP+ 05) \square AC; 2Z –D8(216) (значение из 746) \square AC (2Z –D8(216))
741	CE01	JUMP IP + 1	IP + 1 \square IP; IP +=1 = 743
742	AE02	LD IP + 2	Загрузка с прямой относительной адресацией (IP + 2) \square AC; берем -240 из ячейки 745
743	EC01	LD &1	Сохранение с смещением относительно стека AC \square SP+1; перезаписываем Z в 7FF
744	0A00	RET	Возврат из подпрограммы и увеличивает SP +1 \square SP

Адрес	Значения	Значения для трассировки	Комментарии
173	0	-900 (FC7C)	Переменная Z
174	0	-80 (FFBO)	Переменная Y
175	0	50 (50)	Переменная X
176	00D7		результат G
745	FF10		константа для подпрограммы C1
746	00D8		константа для подпрограммы C2

2.Описание

Программа предназначена для подсчета значений, которые зависят от X, Y, Z;

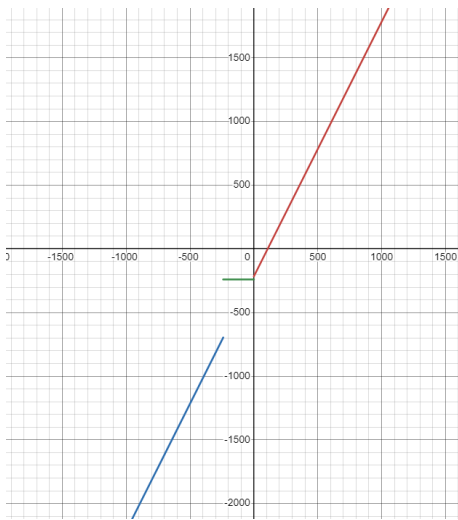
Подпрограмма предназначена для вычисления значения функции:

$f(u) =$

$\{f(u) = 2u - 216, \text{ если } u \geq 0 \quad f(u) = -240, \text{ если } -240 \leq u < 0 \quad f(u) = 2u - 216, \text{ если } u < -240\}$

И вместе они считают $G = f(Y-1) - 1 - f(Z) + f(X-1)$;

График функции



3. Область представления и ОДЗ:

Область представления:

X Y Z – элементы массива, G – результат, C1, C2 – константы

X - знаковое 16-разрядное число, диапазон - $[-2^{15}, 2^{15} - 1]$

Y - знаковое 16-разрядное число, диапазон - $[-2^{15}, 2^{15} - 1]$

Z - знаковое 16-разрядное число, диапазон - $[-2^{15}, 2^{15} - 1]$

C1 - знаковое 16-разрядное число, диапазон - $[-2^{15}, 2^{15} - 1]$

C2 - знаковое 16-разрядное число, диапазон - $[-2^{15}, 2^{15} - 1]$

G - знаковое 16-разрядное число, диапазон - $[-2^{15}, 2^{15} - 1]$

ОДЗ:

$$-2^{15} \leq G = f(Y-1) - 1 - f(Z) + f(X-1) \leq 2^{15} - 1$$

$$-10\,922 \leq f(y-1) \leq 10\,922$$

$$-10\,922 \leq f(x-1) \leq 10\,922$$

$$-10\,922 \leq f(z) \leq 10\,922$$

$$-5352 \leq y \leq 5570$$

$$-5352 \leq x \leq 5570$$

$$-5353 \leq z \leq 5569$$

$$C1, C2 = \text{const}$$

Программа:

Адрес первой выполняемой команды – 15В, адрес последней выполняемой команды - 172.

Подпрограмма:

Адрес первой выполняемой команды – 73А, адрес последней выполняемой команды - 744.

4.Трассировка

Выполняемая команда		Содержимое регистров процессора после выполнения команды								Ячейка, содержимое которой изменилось	
Адрес	Код	IP	CR	AR	DR	SP	BR	AC	NZVC	Адрес	Новый код
15B	0200	15C	0200	15B	0200	000	015B	0000	0100		
15C	EE19	15D	EE19	176	0000	000	0019	0000	0100	176	0000
15D	AE15	15E	AE15	173	FC7C	000	0015	FC7C	1000		
15E	0C00	15F	0C00	7FF	FC7C	7FF	015E	FC7C	1000	7FF	FC7C
15F	D73A	73A	D73A	7FE	0160	7FE	D73A	FC7C	1000	7FE	0160
73A	AC01	73B	AC01	7FF	FC7C	7FE	0001	FC7C	1000		
73B	F303	73C	F303	73B	F303	7FE	073B	FC7C	1000		
73C	7E08	73D	7E08	745	FF10	7FE	0008	FC7C	1000		
73D	F201	73F	F201	73D	F201	7FE	0001	FC7C	1000		
73F	4C01	740	4C01	7FF	FC7C	7FE	0001	F8F8	1001		
740	6E05	741	6E05	746	00D8	7FE	0005	F820	1001		
741	CE01	743	CE01	741	0743	7FE	0001	F820	1001		
743	EC01	744	EC01	7FF	F820	7FE	0001	F820	1001	7FF	F820
744	0A00	160	0A00	7FE	0160	7FF	0744	F820	1001		
160	0800	161	0800	7FF	F820	000	0160	F820	1001		
161	4E14	162	4E14	176	0000	000	0014	F820	1000		
162	EE13	163	EE13	176	F820	000	0013	F820	1000	176	F820
163	AE11	164	AE11	175	0050	000	0011	0050	0000		
164	0740	165	0740	164	0740	000	0164	004F	0001		
165	0C00	166	0C00	7FF	004F	7FF	0165	004F	0001	7FF	004F
166	D73A	73A	D73A	7FE	0167	7FE	D73A	004F	0001	7FE	0167
73A	AC01	73B	AC01	7FF	004F	7FE	0001	004F	0001		
73B	F303	73F	F303	73B	F303	7FE	0003	004F	0001		
73F	4C01	740	4C01	7FF	004F	7FE	0001	009E	0000		
740	6E05	741	6E05	746	00D8	7FE	0005	FFC6	1000		
741	CE01	743	CE01	741	0743	7FE	0001	FFC6	1000		
743	EC01	744	EC01	7FF	FFC6	7FE	0001	FFC6	1000	7FF	FFC6
744	0A00	167	0A00	7FE	0167	7FF	0744	FFC6	1000		
167	0800	168	0800	7FF	FFC6	000	0167	FFC6	1000		
168	4E0D	169	4E0D	176	F820	000	000D	F7E6	1001		
169	EE0C	16A	EE0C	176	F7E6	000	000C	F7E6	1001	176	F7E6

16A	AE09	16B	AE09	174	FFB0	000	0009	FFB0	1001		
16B	0740	16C	0740	16B	0740	000	016B	FFAF	1001		
16C	0C00	16D	0C00	7FF	FFAF	7FF	016C	FFAF	1001	7FF	FAFF
16D	D73A	73A	D73A	7FE	016E	7FE	D73A	FFAF	1001	7FE	016E
73A	AC01	73B	AC01	7FF	FFAF	7FE	0001	FFAF	1001		
73B	F303	73C	F303	73B	F303	7FE	073B	FFAF	1001		
73C	7E08	73D	7E08	745	FF10	7FE	0008	FFAF	0001		
73D	F201	73E	F201	73D	F201	7FE	073D	FFAF	0001		
73E	CE03	742	CE03	73E	0742	7FE	0003	FFAF	0001		
742	AE02	743	AE02	745	FF10	7FE	0002	FF10	1001		
743	EC01	744	EC01	7FF	FF10	7FE	0001	FF10	1001	7FF	FF10
744	0A00	16E	0A00	7FE	016E	7FF	0744	FF10	1001		
16E	0800	16F	0800	7FF	FF10	000	016E	FF10	1001		
16F	0740	170	0740	16F	0740	000	016F	FF0F	1001		
170	6E05	171	6E05	176	F7E6	000	0005	0729	0001		
171	EE04	172	EE04	176	0729	000	0004	0729	0001	176	0729
172	0100	173	0100	172	0100	000	0172	0729	0001		

Вывод: после данной лабораторной работы, я изучил команды iRET CALL PUSH POP и узнал, что такое подпрограмма и что она делает.