

Санкт-Петербургский Национальный Исследовательский
Университет ИТМО
Факультет программной инженерии и компьютерной техники

Лабораторная работа №4
По «Основы Профессиональной Деятельности»
Вариант 14313

Выполнил:
Студент группы Р3106
Рубцов Арсений Дмитриевич
Проверил:
Вербовой Александр Александрович

Оглавление

Текст задания.....	3
Описание программы.....	6
Область представления.....	6
Расположение данных в памяти.....	6
ОДЗ.....	7
Таблица трассировки.....	8
Вывод.....	10

Текст задания

По выданному преподавателем варианту восстановить текст заданного варианта программы и подпрограммы (программного комплекса), определить предназначение и составить его описание, определить область представления и область допустимых значений исходных данных и результата, выполнить трассировку программного комплекса.

1A3: + 0200	1B1: 0800	1BF: YYYY	69F: AE02
1A4: EE1C	1B2: 0700	1C0: XXXX	6A0: EC01
1A5: AE18	1B3: 6E0D	1C1: 00A4	6A1: 0A00
1A6: 0700	1B4: EE0C	-----	6A2: 0200
1A7: 0C00	1B5: AE09	695: AC01	6A3: 00AB
1A8: D695	1B6: 0740	696: F203	
1A9: 0800	1B7: 0C00	697: 7E0A	
1AA: 0740	1B8: D695	698: F006	
1AB: 6E15	1B9: 0800	699: F805	
1AC: EE14	1BA: 0740	69A: 4C01	
1AD: AE12	1BB: 6E05	69B: 4C01	
1AE: 0700	1BC: EE04	69C: 4C01	
1AF: 0C00	1BD: 0100	69D: 4E05	
1B0: D695	1BE: ZZZZ	69E: CE01	

Адрес	Код команды	Мненомика	Комментарии	Описание программы
1A3	0200	CLA	Очистка аккумулятора	—
1A4	EE1C	ST IP + 28	Прямая относительная; загрузка AC в ячейку	Обнуляем ячейку ячейку для накопления результата
1A5	AE18	LD IP + 24	Прямая относительная; MEM(1BE) → AC	Загружаем операнд из ячейки 1BE (Z) в AC
1A6	0700	INC	AC + 1	Вызываем подпрограмму и в качестве параметра передаем Z + 1
1A7	0C00	PUSH	AC → 7FF(стек)	
1A8	D695	CALL 0x695	Вызываем подпрограмму по адресу 0x695 (1A9 → SP(7FE))	
1A9	0800	POP	(SP) + → AC	Записываем результат подпрограммы F(Z + 1) - 1 в AC
1AA	0740	DEC	AC — 1 → AC	
1AB	6E15	SUB IP + 0x15	Прямая относительная; AC — result → AC	
1AC	EE14	ST IP + 0x14	Прямая относительная; AC → result	Помещаем F(Z + 1) - 1 в ячейку для накопления результата

1AD	AE12	LD IP + 0x12	Прямая относительная; MEM(1C0) → AC	Загружаем операнд из ячейки 1C0 (X) в аккумулятор
1AE	0700	INC	AC + 1 → AC	Вызываем подпрограмму и в качестве параметра передаем (X + 1)
1AF	0C00	PUSH	AC → 7FF (стек)	
1B0	D695	CALL 0x695	Вызываем подпрограмму по адресу 0x695 (1B1 → SP(7FE))	
1B1	0800	POP	(SP) + → AC	Записываем результат подпрограммы F(X + 1) + 1 в AC
1B2	0700	INC	AC + 1 → AC	
1B3	6E0D	SUB 0xD	Прямая относительная; AC — result → AC	Вычисляем F(X+1) + 1 - (F(Z + 1) - 1) и помещаем в ячейку для накопления результата
1B4	EE0C	ST IP + 0xC	Прямая относительная; AC → result	
1B5	AE09	LD IP + 0x9	Прямая относительная; MEM(1BF) → AC	Загружаем операнд из ячейки 1BF (Y) в аккумулятор
1B6	0740	DEC	AC — 1 → AC	Вызываем подпрограмму и в качестве параметра передаем (Y - 1)
1B7	0C00	PUSH	AC → 7FF (стек)	
1B8	D695	CALL 0x695	Вызываем подпрограмму по адресу 0x695 (1B9 → SP)	
1B9	0800	POP	(SP) + → AC	Записываем результат подпрограммы F(Y - 1) - 1 в AC
1BA	0740	DEC	AC — 1 → AC	
1BB	6E05	SUB IP + 0x5	Прямая относительная; AC — result → AC	Вычисляем F(Y-1) - 1 - (F(X + 1) + 1 - (F(Z + 1) - 1)) И помещаем в ячейку для накопления результата
1BC	EE04	ST IP + 0x4	Прямая относительная; AC → result	
1BD	0100	HLT	Останов	—
1BE	ZZZZ	Z	Значение Z	—
1BF	YYYY	Y	Значение Y	—
1C0	XXXX	X	Значение X	—
1C1	00A4	R	Результат	—

Подпрограмма:

Адрес	Код команды	Мnemonic	Комментарии	Описание программы
695	AC01	LD SP + 0x1	SP(7FF) → AC	—
696	F203	BMI IP + 0x3	IF N == 1 THEN IP + 3	Если переданный аргумент в AC < 0, то переходим на ячейку 69A
697	7E0A	CMP IP + 0xA	Установка флагов по результату AC - A	Если AC ≤ A, то переход на 69F
698	F006	BEQ IP + 0x6	IF Z == 1 THEN IP + 0x6	
699	F805	BLT IP + 0x5	IF N ⊕ V == 1 THEN IP + 0x5	
69A	4C01	ADD SP + 0x1	Косвенная относительная, со смещением SP; AC + SP(7FF) → AC	Сложение числа с самим собой трижды
69B	4C01	ADD SP + 0x1	Косвенная относительная, со смещением SP; AC + SP(7FF) → AC	
69C	4C01	ADD SP + 0x1	Косвенная относительная, со смещением SP; AC + SP(7FF) → AC	
69D	4E05	ADD IP + 0x5	Прямая относительная; AC + B → AC	Складываем полученное число с константой B
69E	CE01	JUMP IP + 0x1	JUMP (6A0)	Переход на 6A0
69F	AE02	LD IP + 0x2	Прямая относительная; A → AC	Загрузка A (в случае AC ≤ B)
6A0	EC01	ST SP+ 0x1	Косвенная относительная, со смещением SP; AC → SP(7FF)	Значение AC в стек
6A1	0A00	RET	Возврат	
6A2	0200	A	Константа	A = 512
6A3	00AB	B	Константа	B = 171

Описание программы

Назначение программы: нахождение значения функции:

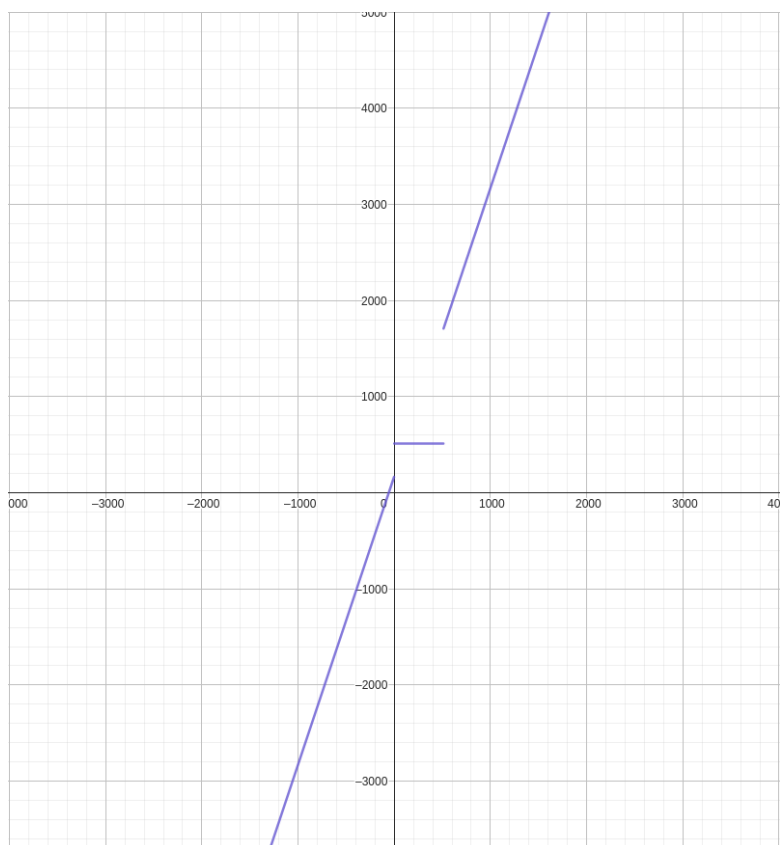
$$R = F(Y-1) - 1 - (F(X+1) + 1 - (F(Z+1) - 1))$$

$$R = F(Y-1) - 1 - (F(X+1) - F(Z+1) + 2)$$

$$R = F(Y-1) - F(X+1) + F(Z+1) - 3$$

$$F(x) = \begin{cases} 3x + 171; & x < 0, x > 512 \\ 512; & 0 \leq x \leq 512 \end{cases}$$

График функции:



Область представления

- X, Y, Z, R, A, B — целые знаковые шестнадцатиразрядные числа.

Расположение данных в памяти

Адрес первой выполняемой инструкции — 1A3, адрес последней - 1BD

Основная программа располагается в ячейках: 1A3 — 1C1.

- 1A3 — 1BD — инструкции
- 1BE — 1C0 — исходные данные (X - 1C0, Y - 1BF, Z - 1BE)
- 1C1 — результат (R)

Подпрограмма расположена в ячейках: 695 — 6A3:

- 695 — 6A1 — инструкции
- 6A2, 6A3 — константы (A, B)

ОДЗ

(const) A = 512

(const) B = 171

Для результата программы R:

$$-2^{15} \leq R \leq 2^{15} - 1$$

Для определения ОДЗ функции, проанализируем данную функцию. При значении аргумента в промежутке $[0, 512]$, функция вернет значение 512. При использовании любого значения из этого промежутка в функции не возникнет переполнения.

При оставшихся значениях аргумента функция вернет выражение $3x+171$. На промежутках $[-2^{15}; 0) \cup (512; 2^{15} - 1)$ эта функция монотонно возрастает, поэтому рассмотрим минимальное и максимальное значение:

$$F_{\min} = F(-2^{15}) = 3 * (-2^{15}) + 512 = -97792$$

$$F_{\max} = F(2^{15} - 1) = 3 * 2^{15} - 1 + 512 = 98815$$

Это означает, что на всем промежутке значений аргумента, результат функции будет находиться на отрезке $[-97792; 98815]$, откуда следует возможное возникновение переполнения, значит требуется ограничить значение операнда таким образом, чтобы в результате значение функции не выходило переполнение.

=> функция не будет иметь переполнения на промежутке: $[-10921; 10865]$

$$F_{\min} = F(-10922) = 3 * (-10922) + 171 = -32595$$

$$F_{\max} = F(10865) = 3 * (10865) + 171 = 32766$$

Данные значения являются максимальными (до возникновения переполнения), но программа вычисляет выражение следующего вида

$$R = F(Y - 1) - F(X + 1) + F(Z + 1) - 3, \text{ где}$$

$$R_{\min} = F(-10922) - F(10865) + F(-10922) = -32595 - 32766 - 32595 = -97956 < -2^{15}$$

$$R_{\max} = F(10865) - F(-10922) + F(10865) = 32766 + 32595 + 32766 = 98127 > 2^{15} - 1$$

В обоих случаях переполнение возможно, значит для операндов $X + 1$, $Y - 1$, $Z + 1$ одз следующее:

$$\begin{cases} -3640 \leq X \leq 3580 \\ -3639 \leq Y \leq 3581 \\ -3640 \leq Z \leq 3580 \end{cases}$$

Таблица трассировки

A = 0x0691

B = 0xF015

X = 0x049B

Y = 0xFD64

Z = 0x0077

Адрес	Знач	IP	CR	AR	DR	SP	BR	AC	NZVC	Адрес	Знач
1A3	0200	1A4	0200	1A3	0200	000	01A3	0000	0100		
1A4	EE1C	1A5	EE1C	1C1	0000	000	001C	0000	0100	1C1	0000
1A5	AE18	1A6	AE18	1BE	0077	000	0018	0077	0000		
1A6	0700	1A7	0700	1A6	0700	000	01A6	0078	0000		
1A7	0C00	1A8	0C00	7FF	0078	7FF	01A7	0078	0000	7FF	0078
1A8	D695	695	D695	7FE	01A9	7FE	D695	0078	0000	7FE	01A9
695	AC01	696	AC01	7FF	0078	07FE	0001	0078	0000		
696	F203	697	F203	696	F203	7FE	0696	0078	0000		
697	7E0A	698	7E0A	6A2	0691	7FE	000A	0078	1000		
698	F006	699	F006	698	F006	7FE	0698	0078	1000		
699	F805	69F	F805	699	F805	7FE	0005	0078	1000		
69F	AE02	6A0	AE02	6A2	0691	7FE	0002	0691	0000		
6A0	EC01	6A1	EC01	7FF	0691	7FE	0001	0691	0000	7FF	0691
6A1	0A00	1A9	0A00	7FE	01A9	7FF	06A1	0691	0000		
1A9	0800	1AA	0800	7FF	0691	000	01A9	0691	0000		
1AA	0740	1AB	0740	1AA	0740	000	01AA	0690	0001		
1AB	6E15	1AC	6E15	1C1	0000	000	0015	0690	0001		
1AC	EE14	1AD	EE14	1C1	0690	000	0014	0690	0001	1C1	0690
1AD	AE12	1AE	AE12	1C0	049B	000	0012	049B	0001		
1AE	0700	1AF	0700	1AE	0700	000	01AE	049C	0000		
1AF	0C00	1B0	7FF	049C	01AF	7FF	049C	0000	0000	7FF	049C
1B0	D695	695	D695	7FE	01B1	7FE	D695	049C	0000	7FE	01B1
695	AC01	696	AC01	7FF	049C	7FE	0001	049C	0000		

696	F203	697	F203	696	F203	7FE	0696	049C	0000		
697	7E0A	698	7E0A	6A2	0691	7FE	000A	049C	1000		
698	F006	699	F006	698	F006	7FE	0698	049C	1000		
699	F805	69F	F805	699	F805	7FE	0005	049C	1000		
69F	AE02	6A0	AE02	6A2	0691	7FE	0002	0691	0000		
6A0	EC01	6A1	EC01	7FF	0691	7FE	0001	0691	0000	7FF	0691
6A1	0A00	1B1	0A00	7FE	01B1	7FF	06A1	0691	0000		
1B1	0800	1B2	0800	7FF	0691	000	01B1	0691	0000		
1B2	0700	1B3	0700	1B2	0700	000	01B2	0692	0000		
1B3	6E0D	1B4	6E0D	1C1	0690	000	000D	0002	0001		
1B4	EE0C	1B5	EE0C	1C1	0002	000	000C	0002	0001	1C1	0002
1B5	AE09	1B6	AE09	1BF	FD64	000	0009	FD64	1001		
1B6	0740	1B7	0740	1B6	0740	000	01B6	FD63	1001		
1B7	0C00	1B8	0C00	7FF	FD63	7FF	01B7	FD63	1001	7FF	FD63
1B8	D695	695	D695	7FE	01B9	7FE	D695	FD63	1001	7FE	01B9
695	AC01	696	AC01	7FF	FD63	7FE	0001	FD63	1001		
696	F203	69A	F203	696	F203	7FE	0003	FD63	1001		
69A	4C01	69B	4C01	7FF	FD63	7FE	0001	FAC6	1001		
69B	4C01	69C	4C01	7FF	FD63	7FE	0001	F829	1001		
69C	4C01	69D	4C01	7FF	FD63	7FE	0001	F58C	1001		
69D	4E05	69E	4E05	6A3	F015	7FE	0005	E5A1	1001		
69E	CE01	6A0	CE01	69E	06A0	7FE	0001	E5A1	1001		
6A0	EC01	6A1	EC01	7FF	E5A1	7FE	0001	E5A1	1001	7FF	E5A1
6A1	0A00	1B9	0A00	7FE	01B9	7FF	06A1	E5A1	1001		
1B9	0800	1BA	0800	7FF	E5A1	000	01B9	E5A1	1001		
1BA	0740	1BB	0740	1BA	0740	000	01BA	E5A0	1001		
1BB	6E05	1BC	6E05	1C1	0002	000	0005	E59E	1001		
1BC	EE04	1BD	EE04	1C1	E59E	000	0004	E59E	1001	1C1	E59E
1BD	0100	1BE	0100	1BD	0100	000	01BD	E59E	1001		

Вывод

Во время выполнения лабораторной работы я узнал как работает стек в БЭВМ, научился вызывать подпрограммы, а так же изучил команды PUSH и POP..