



Факультет Программной Инженерии и Компьютерной техники

«Основы профессиональной деятельности»

Лабораторная работа №4

Вариант №91743

Выполнил: Малых Кирилл Романович

Группа: Р3132

Преподаватель: Жук Иван Александрович

Санкт-Петербург, 2026 г.

Содержание

1. Порядок выполнения работы	3
2. Описание программы.....	4
3. Трассировка программы	9
4. Вывод по работе	11
5. Список использованной литературы.....	12

1. Порядок выполнения работы

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

279: + 0200	287: 0700	295: 2064	68B: EC01
27A: EE1A	288: 6E0C	-----	68C: 0A00
27B: AE16	289: EE0B	67F: AC01	68D: OFF7
27C: 0C00	28A: AE08	680: F204	68E: 0074
27D: D67F	28B: 0700	681: F003	
27E: 0800	28C: 0C00	682: 7E0A	
27F: 0740	28D: D67F	683: F006	
280: 4E14	28E: 0800	684: F805	
281: EE13	28F: 4E05	685: 4C01	
282: AE11	290: EE04	686: 4C01	
283: 0700	291: 0100	687: 4C01	
284: 0C00	292: ZZZZ	688: 6E05	
285: D67F	293: YYYY	689: CE01	
286: 0800	294: XXXX	68A: AE02	

2. Описание программы

Основная программа:

Адрес	Код команды/данные ячейки	Мнемоника	Описание выполняемой инструкции/хранящихся данных
279	+ 0200	CLA	Очистка аккумулятора $0 \rightarrow AC$
27A	EE1A	ST (IP+26)	Сохранение данных в ячейку памяти из аккумулятора с прямой относительной адресацией: $AC \rightarrow M(27B + 26 = 295)$
27B	AE16	LD (IP+22)	Загрузка из ячейки памяти с прямой относительной адресацией: $M(27C + 22 = 292) \rightarrow AC$
27C	0C00	PUSH	Запись данных из аккумулятора в стек: $AC \rightarrow -(SP)$
27D	D67F	CALL 67F	Вызов подпрограммы: $SP - 1 \rightarrow SP, IP \rightarrow (SP), 67F \rightarrow IP$
27E	0800	POP	Чтение + Удаление элемента из стека: $(SP)+ \rightarrow AC$
27F	0740	DEC	Декремент данных аккумулятора: $AC - 1 \rightarrow AC$
280	4E14	ADD (IP+14)	Сложение с данными из ячейки памяти с прямой относительной адресацией: $AC + M(281 + 14 = 295) \rightarrow AC$
281	EE13	ST (IP+13)	Сохранение в ячейку памяти с прямой относительной адресацией: $AC \rightarrow M(282 + 13 = 295)$
282	AE11	LD (IP+11)	Загрузка данных из ячейки памяти с прямой относительной адресацией: $M(283 + 11 = 294) \rightarrow AC$
283	0700	INC	Инкремент данных аккумулятора: $AC + 1 \rightarrow AC$
284	0C00	PUSH	Запись данных из аккумулятора в стек: $AC \rightarrow -(SP)$
285	D67F	CALL 67F	Вызов подпрограммы: $SP - 1 \rightarrow SP, IP \rightarrow (SP), 67F \rightarrow IP$
286	0800	POP	Чтение + Удаление элемента из стека: $(SP)+ \rightarrow AC$
287	0700	INC	Чтение + Удаление элемента из стека: $(SP)+ \rightarrow AC$
288	6E0C	SUB (IP+12)	Вычитание из аккумулятора данных из ячейки памяти с прямой относительной адресацией $AC - M(289 + 12 = 295) \rightarrow AC$
289	EE0B	ST (IP+11)	Сохранение данных в ячейку памяти с прямой относительной адресацией: $AC \rightarrow M(28A + 11 = 295)$
28A	AE08	LD (IP+8)	Загрузка данных в аккумулятор с прямой относительной адресацией $M(28B + 8 = 293) \rightarrow AC$
28B	0700	INC	Инкремент данных в аккумуляторе $AC + 1 \rightarrow AC$

28C	0C00	PUSH	Запись данных из аккумулятора в стек: AC → -(SP)
28D	D67F	CALL 67F	Вызов подпрограммы: SP - 1 → SP, IP → (SP), 67F → IP
28E	0800	POP	Чтение + Удаление элемента из стека: (SP)+ → AC
28F	4E05	ADD (IP+5)	Суммирование данных с ячейки памяти с аккумулятором AC + M(290 + 5 = 295) → AC
290	EE04	ST (IP+4)	Сохранение в ячейку памяти с прямой относительной адресацией: AC → M(291 + 4 = 295)
291	0100	HLT	Остановка работы программы
292	ZZZZ	Z	Значение переменной Z
293	YYYY	Y	Значение переменной Y
294	XXXX	X	Значение переменной X
295	2064	R	Результат работы программы

Подпрограмма:

Адрес	Код команды/данные ячейки	Мнемоника	Описание выполняемой инструкции/хранящихся данных
67F	AC01	LD (SP+1)	Загрузка данных из стека в аккумулятор с прямой относительной адресацией M(ST + 1) → AC
680	F204	BMI (4)	Переход, если число отрицательное Если N == 1, то <u>685</u> = IP + 5, иначе IP = IP + 1
681	F003	BEQ (3)	Переход, если равенство Если Z == 1, то <u>685</u> = IP + 4, иначе IP = IP + 1
682	7E0A	CMP (10)	Выставление флагов результата команды AC - M(683 + 10 = <u>68D</u>) → NZVC
683	F006	BEQ (6)	Переход, если равенство Если Z == 1, то <u>68A</u> = IP + 7, иначе IP = IP + 1
684	F805	BLT (5)	Переход, если число меньше Если N \oplus V == 1, то <u>68A</u> = IP + 6 , иначе IP = IP + 1
685	4C01	ADD (ST+1)	Суммировать данные из стека с аккумулятором с прямой относительной адресацией: AC + M(ST + 1) → AC
686	4C01	ADD (ST+1)	Суммировать данные из стека с аккумулятором с прямой относительной адресацией: AC + M(ST + 1) → AC
687	4C01	ADD (ST+1)	Суммировать данные из стека с аккумулятором с прямой относительной адресацией: AC + M(ST + 1) → AC
688	6E05	SUB (IP+5)	Вычитание данных из аккумулятора с прямой относительной адресацией AC - M(689 + 5 = <u>68E</u>) → AC
689	CE01	JUMP (IP+1)	Выполнить безусловный переход IP = IP + 2 = <u>68B</u>

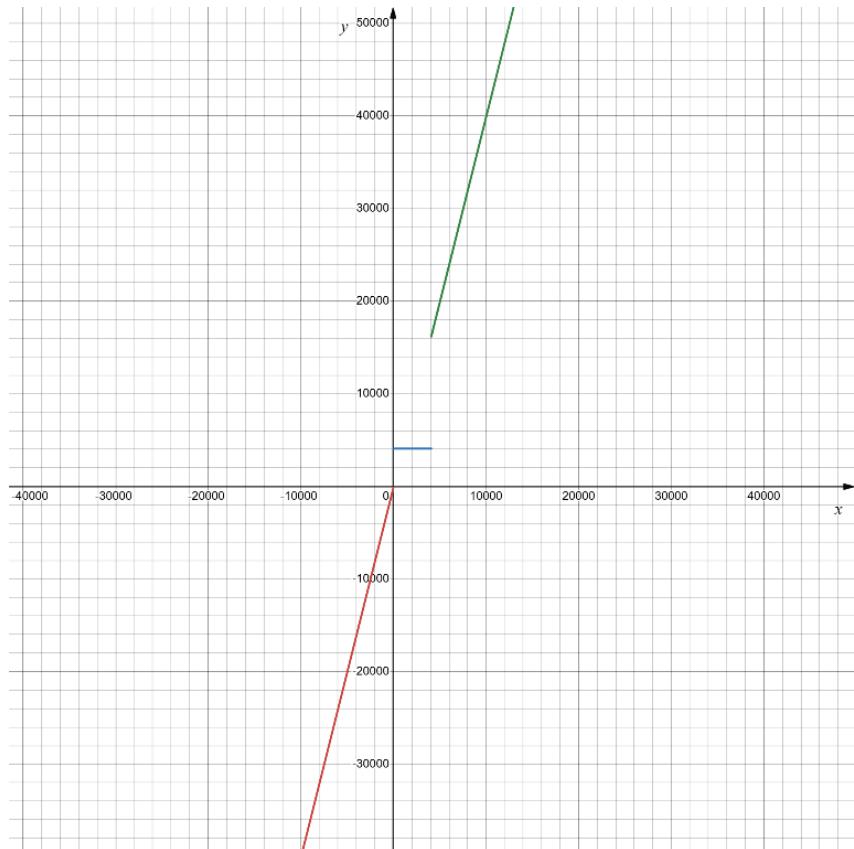
68A	AE02	LD (IP+2)	Загрузка данных в аккумулятор с прямой относительной адресацией $M(68B + 2 = 68D) \rightarrow AC$
68B	EC01	ST (SP+1)	Сохранение данных в стек с прямой относительной адресацией $AC \rightarrow M(SP + 1)$
68C	0A00	RET	Возврат из подпрограммы $(SP)+ \rightarrow IP$
68D	0FF7	0FF7	Константа С (4087)
68E	0074	0074	Константа V (116)

Программа выше находит сумму 3-х различных значений одной и той же функции:

$R = -f(z) + 1 + f(x + 1) + 1 + f(y + 1)$, где f – функция, которую реализовывает подпрограмма.

$$f(x) = \begin{cases} 4087, & x \in (0, 4087] \\ 4x - 116, & x \notin (0, 4087] \end{cases}$$

График функции:



ОПИ

X, Y, Z, R, C, V – 16-ти разрядные целые знаковые числа

ОДЗ

$$\begin{aligned} C &= \text{OFF7}_{16} = 4087_{10} \\ V &= \text{0074}_{16} = 116_{10} \end{aligned} \quad \text{– являются константами}$$

Рассмотрим саму функцию. Пусть x – это наша переменная, значение функции от которой мы и будем искать.

Если $x \in (0, 4087]$, то $f(x) = 4087$. При $R \in [-2^{15}, 2^{15} - 1]$ ситуация с переполнением не произойдёт.

При $x \notin (0, 4087]$ ситуация сложнее. Преобразуем формулу для R :

$$R = -f(z) + 1 + f(x+1) + 1 + f(y+1) = f(x+1) + f(y+1) - f(z) + 2;$$

Следовательно: $(f(x+1) + f(y+1) - f(z)) \in [-2^{15} - 2, 2^{15} - 3]$;

Предположим, что все три функции имеют примерно одинаковое ограничение. Тогда получается:

$$\begin{cases} f(x+1) \in \left[\frac{2(-2^{15}-2)}{3}; \frac{2(2^{15}-3)}{3} \right] \\ f(y+1) \in \left[\frac{2(-2^{15}-2)}{3}; \frac{2(2^{15}-3)}{3} \right] \\ f(z) \in \left[\frac{2(-2^{15}-2)}{3}; \frac{2(2^{15}-3)}{3} \right] \end{cases}$$

Раскроем каждую из функций и найдём ОДЗ для каждой из переменной.

$$f(x+1) \in \left[\frac{2(-2^{15}-2)}{3}; \frac{2(2^{15}-3)}{3} \right] \Rightarrow \frac{2(-2^{15}-2)}{3} \leq 4x - 112 \leq \frac{2(2^{15}-3)}{3};$$

$$x \in \left[\frac{2(-2^{15}-2)}{12} + 112; \frac{2(2^{15}-3)}{12} + 112 \right];$$

$$f(y+1) \in \left[\frac{2(-2^{15}-2)}{3}; \frac{2(2^{15}-3)}{3} \right] \Rightarrow \frac{2(-2^{15}-2)}{3} \leq 4y - 112 \leq \frac{2(2^{15}-3)}{3};$$

$$y \in [\frac{2(-2^{15}-2)}{12} + 112; \frac{2(2^{15}-3)}{12} + 112];$$

$$f(z) \in [\frac{2(-2^{15}-2)}{3}; \frac{2(2^{15}-3)}{3}] \Rightarrow \frac{2(-2^{15}-2)}{3} \leq 4z - 116 \leq \frac{2(2^{15}-3)}{3};$$

$$z \in [\frac{2(-2^{15}-2)}{12} + 116; \frac{2(2^{15}-3)}{12} + 116];$$

Итог

$$\begin{cases} X \in [\frac{2(-2^{15}-2)}{12} + 112; \frac{2(2^{15}-3)}{12} + 112]; \\ Y \in [\frac{2(-2^{15}-2)}{12} + 112; \frac{2(2^{15}-3)}{12} + 112]; \\ Z \in [\frac{2(-2^{15}-2)}{12} + 116; \frac{2(2^{15}-3)}{12} + 116]; \\ R \in [-2^{15}, 2^{15}-1] \\ C = 4087 \\ V = 116 \end{cases}$$

Расположение данных в памяти:

Основная программа:

279-291 – инструкции;

292-294 – входные данные;

295 – ячейка результат;

Подпрограмма:

67F-68C – инструкции

68D, 68E – константы.

3. Трассировка программы

$$Z = 5000_{10} = 1388_{16},$$

$$Y = 0_{10} = 0_{16},$$

$$X = 5000_{10} = 1388_{16}$$

Выполняемая команда				Содержимое регистров процессора после выполнения команды					Ячейка, чьё содержимое изменилось после выполнения команды			
Адрес	Код Команды	IP	CR	AR	DR	SP	BR	AC	PS	NZVC	Адрес	Новый код
292	1388	-	-	-	-	-	-	-	-	-	-	-
293	0	-	-	-	-	-	-	-	-	-	-	-
294	1388	-	-	-	-	-	-	-	-	-	-	-
295	R	-	-	-	-	-	-	-	-	-	-	-
68D	OFF7	-	-	-	-	-	-	-	-	-	-	-
68E	0074	-	-	-	-	-	-	-	-	-	-	-
279	0200	279	0000	000	0000	000	0000	0000	004	0100		
279	0200	27A	0200	279	0200	000	0279	0000	004	0100		
27A	EE1A	27B	EE1A	295	0000	000	001A	0000	004	0100	295	0000
27B	AE16	27C	AE16	292	1388	000	0016	1388	000	0000		
27C	0C00	27D	0C00	7FF	1388	7FF	027C	1388	000	0000	7FF	1388
27D	D67F	67F	D67F	7FE	027E	7FE	D67F	1388	000	0000	7FE	027E
67F	AC01	680	AC01	7FF	1388	7FE	0001	1388	000	0000		
680	F204	681	F204	680	F204	7FE	0680	1388	000	0000		
681	F003	682	F003	681	F003	7FE	0681	1388	000	0000		
682	7E0A	683	7E0A	68D	OFF7	7FE	000A	1388	001	0001		
683	F006	684	F006	683	F006	7FE	0683	1388	001	0001		
684	F805	685	F805	684	F805	7FE	0684	1388	001	0001		
685	4C01	686	4C01	7FF	1388	7FE	0001	2710	000	0000		
686	4C01	687	4C01	7FF	1388	7FE	0001	3A98	000	0000		
687	4C01	688	4C01	7FF	1388	7FE	0001	4E20	000	0000		
688	6E05	689	6E05	68E	0074	7FE	0005	4DAC	001	0001		
689	CE01	68B	CE01	689	068B	7FE	0001	4DAC	001	0001		
68B	EC01	68C	EC01	7FF	4DAC	7FE	0001	4DAC	001	0001	7FF	4DAC
68C	0A00	27E	0A00	7FE	027E	7FF	068C	4DAC	001	0001		
27E	0800	27F	0800	7FF	4DAC	000	027E	4DAC	001	0001		
27F	0740	280	0740	27F	0740	000	027F	4DAB	001	0001		
280	4E14	281	4E14	295	0000	000	0014	4DAB	000	0000		
281	EE13	282	EE13	295	4DAB	000	0013	4DAB	000	0000	295	4DAB
282	AE11	283	AE11	294	1388	000	0011	1388	000	0000		
283	0700	284	0700	283	0700	000	0283	1389	000	0000		
284	0C00	285	0C00	7FF	1389	7FF	0284	1389	000	0000	7FF	1389
285	D67F	67F	D67F	7FE	0286	7FE	D67F	1389	000	0000	7FE	0286
67F	AC01	680	AC01	7FF	1389	7FE	0001	1389	000	0000		
680	F204	681	F204	680	F204	7FE	0680	1389	000	0000		

681	F003	682	F003	681	F003	7FE	0681	1389	000	0000		
682	7E0A	683	7E0A	68D	OFF7	7FE	000A	1389	001	0001		
683	F006	684	F006	683	F006	7FE	0683	1389	001	0001		
684	F805	685	F805	684	F805	7FE	0684	1389	001	0001		
685	4C01	686	4C01	7FF	1389	7FE	0001	2712	000	0000		
686	4C01	687	4C01	7FF	1389	7FE	0001	3A9B	000	0000		
687	4C01	688	4C01	7FF	1389	7FE	0001	4E24	000	0000		
688	6E05	689	6E05	68E	0074	7FE	0005	4DB0	001	0001		
689	CE01	68B	CE01	689	068B	7FE	0001	4DB0	001	0001		
68B	EC01	68C	EC01	7FF	4DB0	7FE	0001	4DB0	001	0001	7FF	4DB0
68C	0A00	286	0A00	7FE	0286	7FF	068C	4DB0	001	0001		
286	0800	287	0800	7FF	4DB0	000	0286	4DB0	001	0001		
287	0700	288	0700	287	0700	000	0287	4DB1	000	0000		
288	6E0C	289	6E0C	295	4DAB	000	000C	0006	001	0001		
289	EE0B	28A	EE0B	295	0006	000	000B	0006	001	0001	295	0006
28A	AE08	28B	AE08	293	0000	000	0008	0000	005	0101		
28B	0700	28C	0700	28B	0700	000	028B	0001	000	0000		
28C	0C00	28D	0C00	7FF	0001	7FF	028C	0001	000	0000	7FF	0001
28D	D67F	67F	D67F	7FE	028E	7FE	D67F	0001	000	0000	7FE	028E
67F	AC01	680	AC01	7FF	0001	7FE	0001	0001	000	0000		
680	F204	681	F204	680	F204	7FE	0680	0001	000	0000		
681	F003	682	F003	681	F003	7FE	0681	0001	000	0000		
682	7E0A	683	7E0A	68D	OFF7	7FE	000A	0001	008	1000		
683	F006	684	F006	683	F006	7FE	0683	0001	008	1000		
684	F805	68A	F805	684	F805	7FE	0005	0001	008	1000		
68A	AE02	68B	AE02	68D	OFF7	7FE	0002	OFF7	000	0000		
68B	EC01	68C	EC01	7FF	OFF7	7FE	0001	OFF7	000	0000	7FF	OFF7
68C	0A00	28E	0A00	7FE	028E	7FF	068C	OFF7	000	0000		
28E	0800	28F	0800	7FF	OFF7	000	028E	OFF7	000	0000		
28F	4E05	290	4E05	295	0006	000	0005	OFFD	000	0000		
290	EE04	291	EE04	295	OFFD	000	0004	OFFD	000	0000	295	OFFD
291	0100	292	0100	291	0100	000	0291	OFFD	000	0000		

4. Вывод по работе

В ходе данной лабораторной работы с впервые познакомился с подпрограммами. Сначала я изучил, как вызывать эти подпрограммы, записывать данные в стек и извлекать их потом. Я узнал о PIC программах и как правильно их организовывать внутри памяти. Было поначалу весело, но потом всё поменялось. Не так я всё это представлял, когда мы начинали. Мне грезились деньги, машины, женщины, уважение, свобода. Я всё это даже получил, более или менее, но вместе с тем пришли тюрьма, постоянный страх и кровь моих товарищей. Я держался на плаву сколько мог, но шансов становилось всё меньше. Теперь это был только вопрос времени.

5. Список использованной литературы

- 1) Кириллов В.В. Архитектура **базовой ЭВМ** – СПб: СПбГУ ИТМО, 2010. – 144 с.
 - 2) Клименков С.В. – “ОПД 2020_21, #1-4. Архитектура и система команд БЭВМ”.
- URL: https://vkvideo.ru/video-34559124_456239019
- 3) Клименков С.В. – “Как построить ферму мобов в майнкрафте”
- URL: https://vkvideo.ru/video-34559124_456239020
- 4) Цист Яна, – Принцип работы Стэка в современных вычислительных машинах / М.: Издательство «Глубинная Глубина», 2024. — 128 с.