Министерство высшего образования и науки Российской Федерации Национальный научно-исследовательский университет ИТМО Факультет программной инженерии и компьютерной техники

Лабораторная работа №4 по дисциплине «Основы профессиональной деятельности».

Вариант №726.

Работу выполнил: Афанасьев Кирилл Александрович, Студент группы Р3106. Преподаватель: Афанасьев Дмитрий Борисович.

Оглавление

Задание	3
Текст исходной программы	3
Описание программы	4
Таблица трассировки выполнения команд	6
Вывод	7

Задание

«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. Вариант 726:

3ED: -	+ 0200	1	3FB:	6E0B	6F0:	AC01		6FE:	0A00
3EE:	EE18	Ì	3FC:	EE0A	6F1:	F001	Ì	6FF:	FB63
3EF:	AE16	Ī	3FD:	AE06	6F2:	F304	ĺ	700:	0043
3F0:	0C00		3FE:	0C00	6F3:	6E0B			
3F1:	D6F0		3FF:	D6F0	6F4:	F201			
3F2:	0800		400:	0800	6F5:	CE06			
3F3:	0700		401:	6E05	6F6:	4E08			
3F4:	6E12		402:	EE04	6F7:	0500			
3F5:	EE11		403:	0100	6F8:	0500			
3F6:	AE0E		404:	ZZZZ	6F9:	6C01			
3F7:	0C00		405:	YYYY	6FA:	6E05			
3F8:	D6F0		406:	XXXX	6FB:	CE01			
3F9:	0800		407:	FB65	6FC:	AE02			
3FA:	0740				6FD:	EC01			

Текст исходной программы

Таблица 1: Текст исходной программы.

		ı	Тиолици 1. Текст исхооной программы.
Адрес	Код команды	Мнемоника	Комментарий
3ED	0200	CLA	MEM(407) = 0
3EE	EE18	ST IP + 0x18	IVIEIVI(407) = 0
3EF	AE16	LD IP + 0x16	Фактический адрес: 406. X -> AC
3F0	0C00	PUSH	X -> 7FF (X в стек)
3F1	D6F0	CALL 0x6F0	Вызов подпрограммы по адресу 0x6F0 (3F2 в стек)
3F2	0800	POP	(3F2 сняли со стека) Снять F(X) со стека
3F3	0700	INC	F(X) + 1 -> AC
3F4	6E12	SUB IP + 0x12	F(X) + 1 - 0 -> AC (MEM(407) = 0)
3F5	EE11	ST IP + 0x11	MEM(407) = F(X) + 1
3F6	AE0E	LD IP + 0xE	Y -> AC
3F7	0C00	PUSH	Y -> 7FF (Y в стек)
3F8	D6F0	CALL 0x6F0	Вызов подпрограммы по адресу 0x6F0 (3F9 в стек)
3F9	0800	POP	Снять F(Y) со стека
3FA	0740	DEC	F(Y) - 1 -> AC
3FB	6E0B	SUB IP + 0xB	F(Y) - 1 - (F(X) + 1) -> AC (MEM(407) = F(X) + 1)

3FD AE06 LD IP + 0x6 Z -> AC 3FE 0C00 PUSH Z -> 7FF (Z в стек) 3FF D6F0 CALL 0x6F0 Bызов подпрограммы по адресу 0x6F0 (400 в стек) 400 0800 POP CH3Tь F(Z) со стека 401 6E05 SUB IP + 0x5 F(Z) - MEM(407) -> AC 402 EE04 ST IP + 0x4 MEM(407) = F(Z) - F(Y) + F(X) + 2 403 0100 HLT Peзультат в 407. Формула выше 404 ZZZZ Z 405 YYYY Y Исходные данные 406 XXXX X 407 F865 R Peзультат работы программы 6F0 AC01 LD SP + 0x1 SP: 7FE + 1 -> 7FF (Загрузить аргумент функции) 6F1 F001 BZS IP + 0x1 Eсли 0-> 6F3 6F2 F304 BPL IP + 0x4 Eсли положительное -> 6F7 6F3 6E0B SUB IP + 0x1 Eсли положительное -> 6F6 6F4 F201 BMI IP + 0x1 Eсли все еще отрицательное -> 6F6 6F5	3FC	EE0A	ST IP + 0xA	MEM(407) = F(Y) - 1 - F(X) - 1 = F(Y) - F(X) - 2					
3FE ОСОО PUSH Z -> 7FF (Z в стек) 3FF D6FO CALL 0x6FO Вызов подпрограммы по адресу 0x6FO (400 в стек) 400 0800 POP Снять F(Z) со стека 401 6E05 SUB IP + 0x5 F(Z) - MEM(407) -> AC 402 EE04 ST IP + 0x4 MEM(407) = F(Z) - F(Y) + F(X) + 2 403 0100 HLT Peзультат в 407. Формула выше 404 ZZZZ Z 405 YYYY Y MCX0Дные данные 406 XXXX X 407 F865 R Peзультат работы программы 6F0 AC01 LD SP + 0x1 SP: 7FE + 1 -> 7FF (Загрузить аргумент функции) 6F1 F001 BZS IP + 0x1 Если положительное -> 6F7 6F2 F304 BPL IP + 0x4 Если положительное -> 6F7 6F3 6E0B SUB IP + 0x8 Мначе фактический адрес: 6FF. Аргумент - МЕМ(6FF) - AC 6F4 F201 BMI IP + 0x1 Если все еще отрицательное -> сложить с мЕМ(6FF). Вернуть все в исходное состояние 6F5 CE06 <		+	-						
3FF D6F0 CALL 0x6F0 Вызов подпрограммы по адресу 0x6F0 (400 в стек) 400 0800 POP Снять F(Z) со стека 401 6E05 SUB IP + 0x5 F(Z) - MEM(407) -> AC 402 EE04 ST IP + 0x4 MEM(407) = F(Z) - F(Y) + F(X) + 2 403 0100 HLT Peзультат в 407. Формула выше 404 ZZZZ Z 405 YYYY Y Исходные данные 406 XXXX X 407 F865 R Результат работы программы 6F0 AC01 LD SP + 0x1 SP: 7FE + 1 -> 7FF (Загрузить аргумент функции) 6F1 F001 BZS IP + 0x1 Если 0 -> 6F3 6F2 F304 BPL IP + 0x4 Если положительное -> 6F7 6F3 6E0B SUB IP + 0xB Иначе фактический адрес: 6FF. Аргумент - МЕМ(6FF) -> AC 6F4 F201 BMI IP + 0x1 Если было отрицательное -> 6F6 6F5 CE06 JUMP IP + 0x6 Иначе -> 6FC 6F6 4E08 ADD IP + 0x8 МЕМ(6FF). Вернуть все в исхо		+							
ЗFF D6F0 CALL 0X6F0 стек) 400 0800 POP Снять F(Z) со стека 401 6E05 SUB IP + 0x5 F(Z) - MEM(407) -> AC 402 EE04 ST IP + 0x4 MEM(407) = F(Z) - F(Y) + F(X) + 2 403 0100 HLT Peзультат в 407. Формула выше 404 ZZZZ Z 405 YYYY Y Исходные данные 406 XXXX X 407 FB65 R Peзультат работы программы 6F0 AC01 LD SP + 0x1 SP: 7FE + 1 -> 7FF (Загрузить аргумент функции) 6F1 F001 BZS IP + 0x1 Eсли 0 -> 6F3 6F2 F304 BPL IP + 0x4 Eсли положительное -> 6F7 6F3 6E0B SUB IP + 0xB Иначе фактический адрес: 6FF. Аргумент - МЕМ(6FF) -> AC 6F4 F201 BMI IP + 0x1 Eсли было отрицательное -> 6F6 6F5 CE06 JUMP IP + 0x6 Иначе -> 6FC 6F6 4E08 ADD IP + 0x8 MEM(6FF). Вернуть все в исходное состояние	3FE	0000	PUSH	, , , , , , , , , , , , , , , , , , ,					
401 6E05 SUB IP + 0x5 F(Z) - MEM(407) -> AC 402 EE04 ST IP + 0x4 MEM(407) = F(Z) - F(Y) + F(X) + 2 403 0100 HLT Pesyльтат в 407. Формула выше 404 ZZZZ Z Исходные данные 406 XXXX X X 407 FB65 R Pesyльтат работы программы 6F0 AC01 LD SP + 0x1 SP: 7FE + 1 -> 7FF (3aгрузить аргумент функции) 6F1 F001 BZS IP + 0x1 Eсли 0 -> 6F3 6F2 F304 BPL IP + 0x4 Eсли положительное -> 6F7 6F3 6E0B SUB IP + 0x8 Иначе фактический адрес: 6FF. Аргумент - MEM(6FF) -> AC 6F4 F201 BMI IP + 0x1 Eсли все еще отрицательное -> 6F6 6F5 CE06 JUMP IP + 0x6 Иначе -> 6FC 6F6 4E08 ADD IP + 0x8 Eсли было отрицательное -> сложить с MEM(6FF). Вернуть все в исходное состояние 6F7 0500 ASL Умножить результат на 2 (уже на 4) 6F9 6C01 SUB SP + 0x1 Вычесть аргумент <t< td=""><td>3FF</td><td>D6F0</td><td>CALL 0x6F0</td><td colspan="6"></td></t<>	3FF	D6F0	CALL 0x6F0						
402 EE04 ST IP + 0x4 MEM(407) = F(Z) - F(Y) + F(X) + 2 403 0100 HLT Peзультат в 407. Формула выше 404 ZZZZ Z 405 YYYY Y Исходные данные 406 XXXX X 407 FB65 R Peзультат работы программы 6F0 AC01 LD SP + 0x1 SP: 7FE + 1 -> 7FF (Загрузить аргумент функции) 6F1 F001 BZS IP + 0x1 Eсли 0 -> 6F3 6F2 F304 BPL IP + 0x4 Eсли положительное -> 6F7 6F3 6E0B SUB IP + 0xB Иначе фактический адрес: 6FF. Аргумент - MEM(6FF) -> AC 6F4 F201 BMI IP + 0x1 Если все еще отрицательное -> 6F6 6F5 CE06 JUMP IP + 0x6 Иначе -> 6FC 6F6 4E08 ADD IP + 0x8 Eсли было отрицательное -> сложить с MEM(6FF). Вернуть все в исходное состояние 6F7 0500 ASL Умножить результат на 2 (уже на 4) 6F9 6C01 SUB SP + 0x1 Вычесть аргумент 6FA 6E05 SUB IP + 0x	400	0800	POP	Снять F(Z) со стека					
403 0100 HLT Результат в 407. Формула выше 404 ZZZZ Z 405 YYYY Y Исходные данные 406 XXXX X X 407 FB65 R Результат работы программы 6F0 AC01 LD SP + 0x1 SP: 7FE + 1 -> 7FF (Загрузить аргумент функции) 6F1 F001 BZS IP + 0x1 Eсли 0 -> 6F3 6F2 F304 BPL IP + 0x4 Eсли положительное -> 6F7 6F3 6E0B SUB IP + 0x8 Иначе фактический адрес: 6FF. Аргумент - МЕМ(6FF) -> AC 6F4 F201 BMI IP + 0x1 Eсли все еще отрицательное -> 6F6 6F5 CE06 JUMP IP + 0x6 Иначе -> 6FC 6F6 4E08 ADD IP + 0x8 Кели было отрицательное -> сложить с МЕМ(6FF). Вернуть все в исходное состояние 6F7 0500 ASL Умножить результат на 2 6F8 0500 ASL Умножить результат на 2 (уже на 4) 6F9 6C01 SUB SP + 0x1 Вычесть мЕМ(700) 6FB CE01 JUMP IP +	401	6E05	SUB IP + 0x5	F(Z) - MEM(407) -> AC					
404 ZZZZ Z 405 YYYY Y 406 XXXX X 407 FB65 R Peзультат работы программы 6F0 AC01 LD SP + Ox1 SP: 7FE + 1 -> 7FF (Загрузить аргумент функции) 6F1 F001 BZS IP + Ox1 Если 0 -> 6F3 6F2 F304 BPL IP + Ox4 Если положительное -> 6F7 6F3 6E0B SUB IP + Ox8 Иначе фактический адрес: 6FF. Аргумент -	402	EE04	ST IP + 0x4	MEM(407) = F(Z) - F(Y) + F(X) + 2					
405YYYYY406XXXXX407FB65RРезультат работы программы6F0AC01LD SP + 0x1SP: 7FE + 1 -> 7FF (Загрузить аргумент функции)6F1F001BZS IP + 0x1Если 0 -> 6F36F2F304BPL IP + 0x4Если положительное -> 6F76F36E0BSUB IP + 0xBИначе фактический адрес: 6FF. Аргумент - МЕМ(6FF) -> AC6F4F201BMI IP + 0x1Если все еще отрицательное -> 6F66F5CE06JUMP IP + 0x6Иначе -> 6FC6F64E08ADD IP + 0x8Если было отрицательное -> сложить с МЕМ(6FF). Вернуть все в исходное состояние6F70500ASLУмножить результат на 26F80500ASLУмножить результат на 2 (уже на 4)6F96C01SUB SP + 0x1Вычесть аргумент6FA6E05SUB IP + 0x5Вычесть мЕМ(700)6FBCE01JUMP IP + 0x1Перепрыгнуть на 1 команду6FCAE02LD IP + 2MEM(6FF) -> AC6FDEC01ST SP + 1Сохранить результат вместо аргумента6FE0A00RETВзять с вершины стека адрес и перейти обратно в программу6FFFB63Константа 1	403	0100	HLT	Результат в 407. Формула выше					
406 XXXXX X 407 FB65 R Результат работы программы Подпрограмма 6F0 AC01 LD SP + 0x1 SP + 0x1 SP + 7FE + 1 -> 7FF (Загрузить аргумент функции) 6F1 F001 BZS IP + 0x1 ECли 0 -> 6F3 6F2 F304 BPL IP + 0x4 ECли положительное -> 6F7 6F3 6E0B SUB IP + 0xB MEM(6FF) -> AC 6F4 F201 BMI IP + 0x1 ECли все еще отрицательное -> 6F6 6F5 CE06 JUMP IP + 0x6 Иначе -> 6FC 6F6 4E08 ADD IP + 0x8 MEM(6FF). Вернуть все в исходное состояние 6F7 0500 ASL Умножить результат на 2 6F8 0500 ASL Умножить результат на 2 (уже на 4) 6F9 6C01 SUB SP + 0x1 Bычесть аргумент 6FA 6E05 SUB IP + 0x5 Bычесть MEM(700) 6FB CE01 JUMP IP + 0x1 Перепрыгнуть на 1 команду 6FC AE02 LD IP + 2 MEM(6FF) -> AC 6FD EC01 ST SP + 1 Cохранить результат вместо аргумента 6FE 0A00 RET B3ять с вершины стека адрес и перейти обратно в программу 6FF FB63 Константа 1	404	ZZZZ	Z						
407FB65RРезультат работы программыПодпрограмма6F0AC01LD SP + 0x1SP: 7FE + 1 -> 7FF (Загрузить аргумент функции)6F1F001BZS IP + 0x1Если 0 -> 6F36F2F304BPL IP + 0x4Если положительное -> 6F76F36E0BSUB IP + 0xBИначе фактический адрес: 6FF. Аргумент - МЕМ(6FF) -> AC6F4F201BMI IP + 0x1Если все еще отрицательное -> 6F66F5CE06JUMP IP + 0x6Иначе -> 6FC6F64E08ADD IP + 0x8Если было отрицательное -> сложить с МЕМ(6FF). Вернуть все в исходное состояние6F70500ASLУмножить результат на 26F80500ASLУмножить результат на 2 (уже на 4)6F96C01SUB SP + 0x1Вычесть аргумент6FA6E05SUB IP + 0x5Вычесть МЕМ(700)6FBCE01JUMP IP + 0x1Перепрыгнуть на 1 команду6FCAE02LD IP + 2МЕМ(6FF) -> AC6FDEC01ST SP + 1Сохранить результат вместо аргумента6FE0A00RETВзять с вершины стека адрес и перейти обратно в программу6FFFB63Константа 1	405	YYYY	Υ	Исходные данные					
Подпрограмма FREE SP: 7FE + 1 → 7FF (Загрузить аргумент функции) 6F0 AC01 LD SP + 0x1 SP: 7FE + 1 → 7FF (Загрузить аргумент функции) 6F1 F001 BZS IP + 0x1 ECли 0 → 6F3 6F2 F304 BPL IP + 0x4 ECЛИ ПОЛОЖИТЕЛЬНОЕ → 6F7 6F3 6E0B SUB IP + 0x8 Иначе фактический адрес: 6FF. Аргумент - МЕМ(6FF) → AC 6F4 F201 BMI IP + 0x1 ECЛИ ВСЕ ФЩЕ ОТРИЦАТЕЛЬНОЕ → 6F6 6F5 CE06 JUMP IP + 0x6 Иначе → 6FC 6F6 4E08 ADD IP + 0x8 ECЛИ БЫЛО ОТРИЦАТЕЛЬНОЕ → СЛОЖИТЬ С 6F7 0500 ASL Умножить результат на 2 6F8 0500 ASL Умножить результат на 2 (уже на 4) 6F9 6C01 SUB SP + 0x1 Вычесть аргумент 6FA 6E05 SUB IP + 0x5 Вычесть МЕМ(700) 6FB CE01 JUMP IP + 0x1 Перепрыгнуть на 1 команду 6FC AE02 LD IP + 2 МЕМ(6FF) → AC 6FD EC01 ST SP + 1 Сохранить результат вместо аргумент	406	XXXX	X						
6F0 ACO1 LD SP + 0x1 SP: 7FE + 1 -> 7FF (Загрузить аргумент функции) 6F1 F001 BZS IP + 0x1 ECли 0 -> 6F3 6F2 F304 BPL IP + 0x4 ECли положительное -> 6F7 6F3 6E0B SUB IP + 0xB Иначе фактический адрес: 6FF. Аргумент - MEM(6FF) -> AC 6F4 F201 BMI IP + 0x1 ECли все еще отрицательное -> 6F6 6F5 CE06 JUMP IP + 0x6 Иначе -> 6FC 6F6 4E08 ADD IP + 0x8 ECли было отрицательное -> сложить с MEM(6FF). Вернуть все в исходное состояние 6F7 0500 ASL Умножить результат на 2 6F8 0500 ASL Умножить результат на 2 (уже на 4) 6F9 6C01 SUB SP + 0x1 Вычесть аргумент 6FA 6E05 SUB IP + 0x5 Вычесть МЕМ(700) 6FB CE01 JUMP IP + 0x1 Перепрыгнуть на 1 команду 6FC AE02 LD IP + 2 MEM(6FF) -> AC 6FD EC01 ST SP + 1 Сохранить результат вместо аргумента 6FE 0A00 RET Взять с верши	407	FB65	R	Результат работы программы					
6F1 F001 BZS IP + 0x1 Если 0 → 6F3 6F2 F304 BPL IP + 0x4 Если положительное → 6F7 6F3 6E0B SUB IP + 0xB Иначе фактический адрес: 6FF. Аргумент - МЕМ(6FF) → AC 6F4 F201 BMI IP + 0x1 Если все еще отрицательное → 6F6 6F5 CE06 JUMP IP + 0x6 Иначе → 6FC 6F6 4E08 ADD IP + 0x8 Если было отрицательное → сложить с МЕМ(6FF). Вернуть все в исходное состояние 6F7 0500 ASL Умножить результат на 2 6F8 0500 ASL Умножить результат на 2 (уже на 4) 6F9 6C01 SUB SP + 0x1 Вычесть аргумент 6FA 6E05 SUB IP + 0x5 Вычесть МЕМ(700) 6FB CE01 JUMP IP + 0x1 Перепрыгнуть на 1 команду 6FC AE02 LD IP + 2 МЕМ(6FF) → AC 6FD EC01 ST SP + 1 Сохранить результат вместо аргумента 6FE 0A00 RET Взять с вершины стека адрес и перейти обратно в программу 6FF FB63 Константа 1	Подпр	ограмма							
6F2 F304 BPL IP + 0x4 Если положительное -> 6F7 6F3 6E0B SUB IP + 0xB Иначе фактический адрес: 6FF. Аргумент - MEM(6FF) -> AC 6F4 F201 BMI IP + 0x1 Если все еще отрицательное -> 6F6 6F5 CE06 JUMP IP + 0x6 Иначе -> 6FC 6F6 4E08 ADD IP + 0x8 Если было отрицательное -> сложить с MEM(6FF). Вернуть все в исходное состояние 6F7 0500 ASL Умножить результат на 2 6F8 0500 ASL Умножить результат на 2 (уже на 4) 6F9 6C01 SUB SP + 0x1 Вычесть аргумент 6FA 6E05 SUB IP + 0x5 Вычесть МЕМ(700) 6FB CE01 JUMP IP + 0x1 Перепрыгнуть на 1 команду 6FC AE02 LD IP + 2 MEM(6FF) -> AC 6FD EC01 ST SP + 1 Сохранить результат вместо аргумента 6FE 0A00 RET Взять с вершины стека адрес и перейти обратно в программу 6FF FB63 Константа 1	6F0	AC01	LD SP + 0x1	SP: 7FE + 1 -> 7FF (Загрузить аргумент функции)					
6F36E0BSUB IP + 0xBИначе фактический адрес: 6FF. Аргумент - MEM(6FF) -> AC6F4F201BMI IP + 0x1Если все еще отрицательное -> 6F66F5CE06JUMP IP + 0x6Иначе -> 6FC6F64E08ADD IP + 0x8Если было отрицательное -> сложить с MEM(6FF). Вернуть все в исходное состояние6F70500ASLУмножить результат на 26F80500ASLУмножить результат на 2 (уже на 4)6F96C01SUB SP + 0x1Вычесть аргумент6FA6E05SUB IP + 0x5Вычесть MEM(700)6FBCE01JUMP IP + 0x1Перепрыгнуть на 1 команду6FCAE02LD IP + 2MEM(6FF) -> AC6FDEC01ST SP + 1Сохранить результат вместо аргумента6FE0A00RETВзять с вершины стека адрес и перейти обратно в программу6FFFB63Константа 1	6F1	F001	BZS IP + 0x1	Если 0 -> 6F3					
6F3 6E0B SUB IP + 0XB MEM(6FF) -> AC 6F4 F201 BMI IP + 0x1 Eсли все еще отрицательное -> 6F6 6F5 CE06 JUMP IP + 0x6 Иначе -> 6FC 6F6 4E08 ADD IP + 0x8 Eсли было отрицательное -> сложить с MEM(6FF). Вернуть все в исходное состояние 6F7 0500 ASL Умножить результат на 2 6F8 0500 ASL Умножить результат на 2 (уже на 4) 6F9 6C01 SUB SP + 0x1 Вычесть аргумент 6FA 6E05 SUB IP + 0x5 Вычесть МЕМ(700) 6FB CE01 JUMP IP + 0x1 Перепрыгнуть на 1 команду 6FC AE02 LD IP + 2 МЕМ(6FF) -> AC 6FD EC01 ST SP + 1 Сохранить результат вместо аргумента 6FE 0A00 RET Взять с вершины стека адрес и перейти обратно в программу 6FF FB63 Константа 1	6F2	F304	BPL IP + 0x4	Если положительное -> 6F7					
6F5 CE06 JUMP IP + 0x6 Иначе -> 6FC 6F6 4E08 ADD IP + 0x8 Eсли было отрицательное -> сложить с MEM(6FF). Вернуть все в исходное состояние 6F7 0500 ASL Умножить результат на 2 6F8 0500 ASL Умножить результат на 2 (уже на 4) 6F9 6C01 SUB SP + 0x1 Вычесть аргумент 6FA 6E05 SUB IP + 0x5 Вычесть MEM(700) 6FB CE01 JUMP IP + 0x1 Перепрыгнуть на 1 команду 6FC AE02 LD IP + 2 MEM(6FF) -> AC 6FD EC01 ST SP + 1 Сохранить результат вместо аргумента 6FE 0A00 RET Взять с вершины стека адрес и перейти обратно в программу 6FF FB63 Константа 1	6F3	6E0B	SUB IP + 0xB						
6F64E08ADD IP + 0x8Если было отрицательное -> сложить с МЕМ(6FF). Вернуть все в исходное состояние6F70500ASLУмножить результат на 26F80500ASLУмножить результат на 2 (уже на 4)6F96C01SUB SP + 0x1Вычесть аргумент6FA6E05SUB IP + 0x5Вычесть МЕМ(700)6FBCE01JUMP IP + 0x1Перепрыгнуть на 1 команду6FCAE02LD IP + 2МЕМ(6FF) -> AC6FDEC01ST SP + 1Сохранить результат вместо аргумента6FE0A00RETВзять с вершины стека адрес и перейти обратно в программу6FFFB63Константа 1	6F4	F201	BMI IP + 0x1	Если все еще отрицательное -> 6F6					
6F64E08ADD IP + 0X8MEM(6FF). Вернуть все в исходное состояние6F70500ASLУмножить результат на 26F80500ASLУмножить результат на 2 (уже на 4)6F96C01SUB SP + 0x1Вычесть аргумент6FA6E05SUB IP + 0x5Вычесть MEM(700)6FBCE01JUMP IP + 0x1Перепрыгнуть на 1 команду6FCAE02LD IP + 2MEM(6FF) -> AC6FDEC01ST SP + 1Сохранить результат вместо аргумента6FE0A00RETВзять с вершины стека адрес и перейти обратно в программу6FFFB63Константа 1	6F5	CE06	JUMP IP + 0x6	Иначе -> 6FC					
6F8 0500 ASL Умножить результат на 2 (уже на 4) 6F9 6C01 SUB SP + 0x1 Вычесть аргумент 6FA 6E05 SUB IP + 0x5 Вычесть МЕМ(700) 6FB CE01 JUMP IP + 0x1 Перепрыгнуть на 1 команду 6FC AE02 LD IP + 2 MEM(6FF) -> AC 6FD EC01 ST SP + 1 Сохранить результат вместо аргумента 6FE 0A00 RET Взять с вершины стека адрес и перейти обратно в программу 6FF FB63 Константа 1	6F6	4E08	ADD IP + 0x8						
6F9 6C01 SUB SP + 0x1 Вычесть аргумент 6FA 6E05 SUB IP + 0x5 Вычесть MEM(700) 6FB CE01 JUMP IP + 0x1 Перепрыгнуть на 1 команду 6FC AE02 LD IP + 2 MEM(6FF) -> AC 6FD EC01 ST SP + 1 Сохранить результат вместо аргумента 6FE 0A00 RET Взять с вершины стека адрес и перейти обратно в программу 6FF FB63 Константа 1	6F7	0500	ASL	Умножить результат на 2					
6FA 6E05 SUB IP + 0x5 Вычесть МЕМ(700) 6FB CE01 JUMP IP + 0x1 Перепрыгнуть на 1 команду 6FC AE02 LD IP + 2 MEM(6FF) -> AC 6FD EC01 ST SP + 1 Сохранить результат вместо аргумента 6FE 0A00 RET Взять с вершины стека адрес и перейти обратно в программу 6FF FB63 Константа 1	6F8	0500	ASL	Умножить результат на 2 (уже на 4)					
6FB CE01 JUMP IP + 0x1 Перепрыгнуть на 1 команду 6FC AE02 LD IP + 2 MEM(6FF) -> AC 6FD EC01 ST SP + 1 Сохранить результат вместо аргумента 6FE 0A00 RET Взять с вершины стека адрес и перейти обратно в программу 6FF FB63 Константа 1	6F9	6C01	SUB SP + 0x1	Вычесть аргумент					
6FC AE02 LD IP + 2 MEM(6FF) -> AC 6FD EC01 ST SP + 1 Сохранить результат вместо аргумента 6FE 0A00 RET Взять с вершины стека адрес и перейти обратно в программу 6FF FB63 Константа 1	6FA	6E05	SUB IP + 0x5	Вычесть МЕМ(700)					
6FD EC01 ST SP + 1 Сохранить результат вместо аргумента 6FE 0A00 RET Взять с вершины стека адрес и перейти обратно в программу 6FF FB63 Константа 1	6FB	CE01	JUMP IP + 0x1	Перепрыгнуть на 1 команду					
6FE 0A00 RET Взять с вершины стека адрес и перейти обратно в программу 6FF FB63 Константа 1	6FC	AE02	LD IP + 2	MEM(6FF) -> AC					
6FE 0A00 RET в программу 6FF FB63 Константа 1	6FD	EC01	ST SP + 1	Сохранить результат вместо аргумента					
	6FE	0A00	RET						
700 0043 Константа 2	6FF	FB63		Константа 1					
	700	0043		Константа 2					

Окончание таблицы.

Описание программы

• Назначение программы: выполнение вычисления по формуле:

$$R = F(Z) - F(Y) + F(X) + 2,$$
 где $F(x) = \begin{cases} 3x - 67, & x > 0; x < -1181 \\ -1181, -1181 \le x \le 0 \end{cases}$ – подпрограмма, выполняющая вычисление по формуле. График функции $F(x)$:

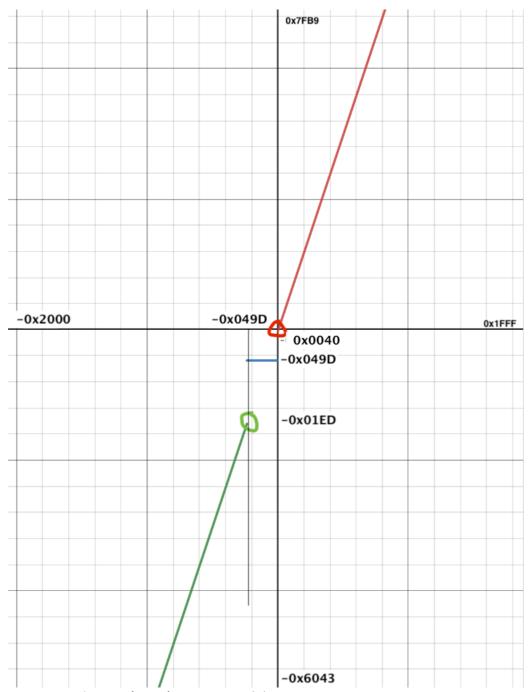


Рисунок 1. График функции F(х).

- Описание исходных данных:
 - X, Y, Z Аргументы функции F.
 - R Результат работы программы
 - о ОПИ:
 - X, Y, Z, R 16-разрядное знаковое целое число.

о ОДЗ:
$$-2^{15} \le R \le 2^{15} - 1$$

$$-0x2000 \le X, Y, Z \le 0x1FFF$$

$$-2^{15} \le F(Z) - F(Y) + F(X) \le 2^{15} - 3$$

$$F(X), F(Y), F(Z) \in [-0x6043; -0x0E1D] \cup$$

$$\cup \{-0x049D\} \cup [-0x0040; 0x7FB9]$$

- Расположение в памяти ЭВМ программы, исходных данных и результатов:
 - о Программа располагается в памяти в ячейках между адресами 3ED и 407 включительно, подпрограмма располагается между адресами 6F0 и 700 включительно.
 - \circ Исходные данные должны располагаться в ячейках памяти: X-406, Y-405, Z-404.
 - Результат работы программы будет находиться в ячейке по адресу 407.
 - о Для хранения неизменяемых данных (констант) используются ячейки памяти с адресами 6FF (Значение константы: 0xFB63) и 700 (Значение константы: 0x0043).
- Первая команда располагается в ячейке по адресу 3ED. Последняя 403.

Таблица трассировки выполнения команд

Таблица 2: Трассировка выполнения команд.

Выполняемая команда Содержимое регистров процессора после выполнения которой изменилось после выполнения команды Адрес команды Код команды IP CR AR DR SP BR AC NZVC Адрес моманды команды 3ED 0200 3EE 0200 3ED 0200 000 03ED 0000 0100 407 0000 3EE EE18 3EF EE18 407 0000 000 03ED 0000 0100 407 0000 3EF AE14 3F0 AE14 404 4AD0 000 0014 4AD0 0000 7FF 4AD0 0000 <td< th=""><th colspan="9">1 иолица 2. Трассировка выполнения комано.</th></td<>	1 иолица 2. Трассировка выполнения комано.											
Выполняемая команда Содержимое регистров процессора после выполнения команды команды изменилось после выполнения команды изменилось после выполнения команды Адрес команды Код команды IP CR AR DR SP BR AC NZVC Aдрес команды команды команды 3ED 0200 3ED 0200 000 03ED 0000 0100 407 0000 3EE EE18 3FF EE18 407 0000 000 0018 0000 0100 407 0000 3FF AE14 3F0 AE14 404 4AD0 000 0014 4AD0 0000 7FF 4AD0 7FF 0001 4AD0 </td <td></td> <td></td> <td colspan="7" rowspan="2"></td> <td>Яче</td> <td>йка,</td>										Яче	йка,	
Выполняемая команда Содержимое регистров процессора после выполнения команды изменилось после выполнения команды Адрес команды IP CR AR DR SP BR AC NZVC Адрес Новый код команды 3ED 0200 3EE 0200 3ED 0200 000 03ED 0000 0100 407 0000 3EE EE18 3EF EE18 407 0000 000 0018 0000 0100 407 0000 3EF AE14 3F0 AE14 404 4AD0 000 0014 4AD0 0000 7FF 0000												
КОМАНДА КОМАНДА ИЗМЕНИЛОСЬ после выполнения команды Адрес команды команды КОД команды IP CR AR DR SP BR AC NZVC Адрес команды команды команды команды 3ED 0200 3ED 0200 000 03ED 0000 0100 407 0000 3EE EE18 3EF EE18 407 0000 000 0018 0000 0100 407 0000 3EF AE14 3F0 AE14 404 4AD0 000 0014 4AD0 0000 7FF 03F2 7FF 03F0 4AD0 0000 7FF 03F2 7FF	Выполняемая		Содержимое регистров процессора после выполнения							•		
Адрес Код команды Код команды IP CR AR DR SP BR AC NZVC Адрес команды Новый код 3ED 0200 3EE 0200 3ED 0200 000 03ED 0000 0100 407 0000 3EE EE18 3EF EE18 407 0000 000 0018 0000 0100 407 0000 3EF AE14 3F0 AE14 404 4AD0 000 0014 4AD0 0000 7FF 03F2 7FF 06F0 4AD0 0000 7FF 03F2 7FF 0001 4AD0				•	•	-		•				
Адрес команды Код команды IP CR AR DR SP BR AC NZVC Адрес команды код Новый код 3ED 0200 3EE 0200 3ED 0200 000 03ED 0000 0100 407 0000 3EE EE18 3EF EE18 407 0000 000 0018 0000 0100 407 0000 3EF AE14 3F0 AE14 404 4AD0 000 0014 4AD0 0000 7FF 4AD0 0000 0000 7FF 4AD0 0000 0000												
Адрес команды Код команды IP CR AR DR SP BR AC NZVC Адрес код Новый код 3ED 0200 3EE 0200 3ED 0200 000 03ED 0000 0100 407 0000 3EE EE18 3EF EE18 407 0000 000 0018 0000 0100 407 0000 3EF AE14 3F0 AE14 404 4AD0 000 0014 4AD0 0000 7FF 03F2 7FF 0001 4AD0 0000 0000 0000 0												
Адрес команды IP CR AR DR SP BR AC NZVC Адрес код 3ED 0200 3ED 0200 000 03ED 0000 0100 — 3EE EE18 3EF EE18 407 0000 000 0018 0000 0100 407 0000 3FF AE14 3F0 AE14 404 4AD0 000 0014 4AD0 0000 7FF 03F2 7FF 001 4AD0 0000 7FF 03F2 7FF 0001 4AD0 0000 7FF 03F2 7FF 06F1 4AD0 7FF 0001 </td <td></td> <td>Кол</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td>KOW</td> <td></td>		Кол									KOW	
3EE EE18 3EF EE18 407 0000 000 0018 0000 0100 407 0000 3EF AE14 3F0 AE14 404 4AD0 000 0014 4AD0 0000 7FF 03F2 7FE 06F0 4AD0 0000 7FE 03F2 6FF 06F0 06F0 7FF 04D0 7FF 0001 4AD0 0000<	Адрес		IP	CR	AR	DR	SP	BR	AC	NZVC	Адрес	
3EF AE14 3F0 AE14 404 4AD0 000 0014 4AD0 0000 7FF 4AD0 7FF 03F0 4AD0 0000 7FF 4AD0 3F1 D6F0 6F0 D6F0 7FE 03F2 7FE D6F0 4AD0 0000 7FE 03F2 6F0 AC01 6F1 AC01 7FF 4AD0 7FE 0001 4AD0 0000 7FE 03F2 6F0 AC01 6F1 AC01 7FF 4AD0 7FE 0001 4AD0 0000 7FE 03F2 6F0 AC01 6F1 F001 6FE 0001 4AD0 0000	3ED	0200	3EE	0200	3ED	0200	000	03ED	0000	0100		
3FO 0C00 3F1 0C00 7FF 4AD0 7FF 03F0 4AD0 0000 7FF 4AD0 3F1 D6F0 6F0 D6F0 7FE 03F2 7FE D6F0 4AD0 0000 7FE 03F2 6F0 AC01 6F1 AC01 7FF 4AD0 7FE 0001 4AD0 0000 9 6F1 F001 6F2 F001 6F1 F001 7FE 06F1 4AD0 0000 9 6F2 F304 6F2 F304 7FE 06F1 4AD0 0000 9 9 6F7 0500 6F8 0500 6F7 4AD0 7FE 06F7 95A0 1010 9 9 95A0 6FB 06F7 95A0 1010 9 9 95A0 6FB 06F7 95A0 1010 9 9 95A0 7FE 06F8 2B40 0011 9 9 9 9	3EE	EE18	3EF	EE18	407	0000	000	0018	0000	0100	407	0000
3F1 D6F0 6F0 D6F0 7FE O3F2 7FE D6F0 4AD0 0000 7FE O3F2 6F0 AC01 6F1 AC01 7FF 4AD0 7FE 0001 4AD0 0000 ————————————————————————————————————	3EF	AE14	3F0	AE14	404	4AD0	000	0014	4AD0	0000		
6F0 AC01 6F1 AC01 7FF 4AD0 7FE 0001 4AD0 0000 6F1 F001 6F2 F001 6F1 F001 7FE 06F1 4AD0 0000 6F2 F304 6F7 F304 6F2 F304 7FE 0004 4AD0 0000 6F7 0500 6F8 0500 6F7 4AD0 7FE 06F7 95A0 1010 6F8 0500 6F9 0500 6F8 95A0 7FE 06F8 2B40 0011 6F9 6C01 6FA 6C01 7FF 4AD0 7FE 0001 E070 1000 6FA 6E05 6FB 6E05 700 0043 7FE 0001 E070 1001 6FB CE01 6FB 06FD 7FE 0001 E02D 1001 7FF E02D 6FE 0A00 3F2 0A00 7FE E02D 7FF	3F0	0C00	3F1	0C00	7FF	4AD0	7FF	03F0	4AD0	0000	7FF	4AD0
6F1 F001 6F2 F001 6F1 F001 7FE 06F1 4ADO 0000	3F1	D6F0	6F0	D6F0	7FE	03F2	7FE	D6F0	4AD0	0000	7FE	03F2
6F2 F304 6F7 F304 6F2 F304 7FE 0004 4AD0 0000 6F7 0500 6F8 0500 6F7 4AD0 7FE 06F7 95A0 1010 6F8 0500 6F9 0500 6F8 95A0 7FE 06F8 2B40 0011 6F9 6C01 6FA 6C01 7FF 4AD0 7FE 0001 E070 1000 6FA 6E05 6FB 6E05 700 0043 7FE 0005 E02D 1001 6FB CE01 6FB 06FD 7FE 0001 E02D 1001 6FD EC01 6FE EC01 7FF E02D 7FE 0001 E02D 1001 6FE 0A00 3F2 0A00 7FF E02D 000 03F2 E02D 1001 3F3 0700 <td< td=""><td>6F0</td><td>AC01</td><td>6F1</td><td>AC01</td><td>7FF</td><td>4AD0</td><td>7FE</td><td>0001</td><td>4AD0</td><td>0000</td><td></td><td></td></td<>	6F0	AC01	6F1	AC01	7FF	4AD0	7FE	0001	4AD0	0000		
6F7 0500 6F8 0500 6F7 4ADO 7FE 06F7 95AO 1010 6F8 0500 6F9 0500 6F8 95AO 7FE 06F8 2B4O 0011 6F9 6C01 6FA 6C01 7FF 4ADO 7FE 0001 E07O 1000 6FA 6E05 6FB 6E05 700 0043 7FE 0005 E02D 1001 6FB CE01 6FB 6EB 06FD 7FE 0001 E02D 1001 6FD EC01 6FE EC01 7FF E02D 7FE 0001 E02D 1001 7FF E02D 6FE 0A00 3F2 0A00 7FF E02D 000 03F2 E02D 1001 3F3 0700 3F4 0700 3F3 0700 000 03F3 E02E 1001	6F1	F001	6F2	F001	6F1	F001	7FE	06F1	4AD0	0000		
6F8 0500 6F9 0500 6F8 95A0 7FE 06F8 2B40 0011 6F9 6C01 6FA 6C01 7FF 4AD0 7FE 0001 E070 1000 6FA 6E05 6FB 6E05 700 0043 7FE 0005 E02D 1001 6FB CE01 6FB 06FD 7FE 0001 E02D 1001 6FD EC01 6FE EC01 7FF E02D 7FE 0001 E02D 1001 7FF E02D 6FE 0A00 3F2 0A00 7FF E02D 7FF 06FE E02D 1001 3F2 0800 3F3 0800 7FF E02D 000 03F2 E02D 1001 3F3 0700 3F4 0700 3F3 0700 000 03F3 E02E 1001 <td< td=""><td>6F2</td><td>F304</td><td>6F7</td><td>F304</td><td>6F2</td><td>F304</td><td>7FE</td><td>0004</td><td>4AD0</td><td>0000</td><td></td><td></td></td<>	6F2	F304	6F7	F304	6F2	F304	7FE	0004	4AD0	0000		
6F9 6C01 6FA 6C01 7FF 4ADO 7FE 0001 E070 1000	6F7	0500	6F8	0500	6F7	4AD0	7FE	06F7	95A0	1010		
6FA 6EO5 6FB 6EO5 700 0043 7FE 0005 EO2D 1001	6F8	0500	6F9	0500	6F8	95A0	7FE	06F8	2B40	0011		
6FB CE01 6FD CE01 6FB 06FD 7FE 0001 E02D 1001	6F9	6C01	6FA	6C01	7FF	4AD0	7FE	0001	E070	1000		
6FD EC01 6FE EC01 7FF E02D 7FE 0001 E02D 1001 7FF E02D 6FE 0A00 3F2 0A00 7FE 03F2 7FF 06FE E02D 1001	6FA	6E05	6FB	6E05	700	0043	7FE	0005	E02D	1001		
6FE 0A00 3F2 0A00 7FE 03F2 7FF 06FE E02D 1001 3F2 0800 3F3 0800 7FF E02D 000 03F2 E02D 1001 3F3 0700 3F4 0700 3F3 0700 000 03F3 E02E 1000 3F4 6E12 3F5 6E12 407 0000 000 0012 E02E 1001	6FB	CE01	6FD	CE01	6FB	06FD	7FE	0001	E02D	1001		
3F2 0800 3F3 0800 7FF E02D 000 03F2 E02D 1001 3F3 0700 3F4 0700 3F3 0700 000 03F3 E02E 1000 3F4 6E12 3F5 6E12 407 0000 000 0012 E02E 1001	6FD	EC01	6FE	EC01	7FF	E02D	7FE	0001	E02D	1001	7FF	E02D
3F3 0700 3F4 0700 3F3 0700 000 03F3 E02E 1000 3F4 6E12 3F5 6E12 407 0000 000 0012 E02E 1001	6FE	0A00	3F2	0A00	7FE	03F2	7FF	06FE	E02D	1001		
3F4 6E12 3F5 6E12 407 0000 000 0012 E02E 1001	3F2	0800	3F3	0800	7FF	E02D	000	03F2	E02D	1001		
	3F3	0700	3F4	0700	3F3	0700	000	03F3	E02E	1000		
3F5 EE11 3F6 EE11 407 E02E 000 0011 E02E 1001 407 E02E	3F4	6E12	3F5	6E12	407	0000	000	0012	E02E	1001		
	3F5	EE11	3F6	EE11	407	E02E	000	0011	E02E	1001	407	E02E

		T	l		l		l	l			
3F6	AE0E	3F7	AE0E	405	FEED	000	000E	FEED	1001		
3F7	0C00	3F8	0C00	7FF	FEED	7FF	03F7	FEED	1001	7FF	FEED
3F8	D6F0	6F0	D6F0	7FE	03F9	7FE	D6F0	FEED	1001	7FE	03F9
6F0	AC01	6F1	AC01	7FF	FEED	7FE	0001	FEED	1001		
6F1	F001	6F2	F001	6F1	F001	7FE	06F1	FEED	1001		
6F2	F304	6F3	F304	6F2	F304	7FE	06F2	FEED	1001		
6F3	6E0B	6F4	6E0B	6FF	FB63	7FE	000B	038A	0001		
6F4	F201	6F5	F201	6F4	F201	7FE	06F4	038A	0001		
6F5	CE06	6FC	CE06	6F5	06FC	7FE	0006	038A	0001		
6FC	AE02	6FD	AE02	6FF	FB63	7FE	0002	FB63	1001		
6FD	EC01	6FE	EC01	7FF	FB63	7FE	0001	FB63	1001	7FF	FB63
6FE	0A00	3F9	0A00	7FE	03F9	7FF	06FE	FB63	1001		
3F9	0800	3FA	0800	7FF	FB63	000	03F9	FB63	1001		
3FA	0740	3FB	0740	3FA	0740	000	03FA	FB62	1001		
3FB	6E0B	3FC	6E0B	407	E02E	000	000B	1B34	0001		
3FC	EE0A	3FD	EEOA	407	1B34	000	000A	1B34	0001	407	1B34
3FD	AE08	3FE	AE08	406	BEAD	000	8000	BEAD	1001		
3FE	0C00	3FF	0C00	7FF	BEAD	7FF	03FE	BEAD	1001	7FF	BEAD
3FF	D6F0	6F0	D6F0	7FE	0400	7FE	D6F0	BEAD	1001	7FE	0400
6F0	AC01	6F1	AC01	7FF	BEAD	7FE	0001	BEAD	1001		
6F1	F001	6F2	F001	6F1	F001	7FE	06F1	BEAD	1001		
6F2	F304	6F3	F304	6F2	F304	7FE	06F2	BEAD	1001		
6F3	6E0B	6F4	6E0B	6FF	FB63	7FE	000B	C34A	1000		
6F4	F201	6F6	F201	6F4	F201	7FE	0001	C34A	1000		
6F6	4E08	6F7	4E08	6FF	FB63	7FE	0008	BEAD	1001		
6F7	0500	6F8	0500	6F7	BEAD	7FE	06F7	7D5A	0011		
6F8	0500	6F9	0500	6F8	7D5A	7FE	06F8	FAB4	1010		
6F9	6C01	6FA	6C01	7FF	BEAD	7FE	0001	3C07	0001		
6FA	6E05	6FB	6E05	700	0043	7FE	0005	3BC4	0001		
6FB	CE01	6FD	CE01	6FB	06FD	7FE	0001	3BC4	0001		
6FD	EC01	6FE	EC01	7FF	3BC4	7FE	0001	3BC4	0001	7FF	3BC4
6FE	0A00	400	0A00	7FE	0400	7FF	06FE	3BC4	0001		
400	0800	401	0800	7FF	3BC4	000	0400	3BC4	0001		
401	6E05	402	6E05	407	1B34	000	0005	2090	0001		
402	EE04	403	EE04	407	2090	000	0004	2090	0001	407	2090
403	0100	404	0100	403	0100	000	0403	2090	0001		-
	1	_ ·• ·	1		1	1 330	1 5 .00				L

Окончание таблицы.

Вывод

Во время выполнения данной лабораторной работы я ознакомился с организацией работы программных комплексов в БЭВМ, устройством стека и способах его управления, а также способах передачи аргументов между подпрограммами.