

Федеральное государственное автономное образовательное учреждение высшего
образования «Национальный исследовательский университет ИТМО»

Факультет Программной Инженерии и Компьютерной Техники

Лабораторная работа №3
Вариант 7869

Выполнил:
Лежнев Никита Сергеевич
Группа Р3112
Проверил:
Абузов Ярослав Александрович

Санкт-Петербург 2024г.

Содержание

Задание	3
Описание программы	5
Расположение данных в памяти	5
Адрес первой и последней выполняемой инструкции	5
Область представления	5
Область допустимых значений	5
Таблица трасировки	6
Вывод	8

Задание

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

Введите номер варианта **7869**

219:	022F		227:	F404
21A:	0200		228:	0400
21B:	4000		229:	0400
21C:	E000		22A:	4EF1
21D:	+ 0200		22B:	EEF0
21E:	EEFD		22C:	821B
21F:	AF04		22D:	CEF5
220:	EEFA		22E:	0100
221:	4EF7		22F:	F403
222:	EEF7		230:	E21C
223:	ABF6		231:	0480
224:	0480		232:	1003
225:	F406			
226:	0480			

Адрес	Код команды	Мнемоника	Комментарии
219	022F	A	Адрес первого элемента
21A	0200	B	Адрес следующего элемента для проверки
21B	4000	N	Количество элементов массива
21C	E000	R	Результат
21D	0200	CLA	Очистка аккумулятора
21E	EEFD	ST IP-3	Сохранение аккумулятора AC -> R
21F	AF04	LD #04	Прямая загрузка 0004 -> AC
220	EEFA	ST IP-6	Сохранение аккумулятора AC -> N
221	4EF7	ADD (IP - 9)	Сложение аккумулятора AC = AC + A
222	EEF7	ST IP-9	Сохранение аккумулятора AC -> B
223	ABF6	LD IP-A	B -= 1; Загрузка MEM(B) -> AC
224	0480	ROR	Циклический сдвиг вправо
225	F406	BHIS(BCS) IP+6	Если C == 1: IP + 6 + 1 -> IP(22C)
226	0480	ROR	Циклический сдвиг вправо
227	F404	BHIS(BCS) IP+4	Если C == 1: IP + 4 + 1 -> IP(22C)
228	0400	ROL	Циклический сдвиг влево
229	0400	ROL	Циклический сдвиг влево
22A	4EF1	ADD (IP-F)	Сложение AC = AC + MEM(R)
22B	EEF0	ST IP-16	Сохранение аккумулятора AC -> R
22C	821B	LOOP 21B	N - 1 -> N; если N <= 0: IP - 1 -> IP
22D	CEF5	JUMP IP-B	IP - B + 1 -> IP
22E	0100	HLT	Остановка программы.
22F	F403		Элемент массива
230	E21C		Элемент массива
231	0480		Элемент массива
232	1003		Элемент массива

Описание программы

Программа считает сумму всех чисел в массиве, которые делятся на 4.

Расположение данных в памяти

A(219) - Адрес первого элемента

B(21A) - Адрес следующего элемента для проверки

N(21B) - Количество элементов массива

R(21C) - Результат

22F...232 - массив с данными

Адрес первой и последней выполняемой инструкции

21D - адрес первой инструкции

22E - адрес последней инструкции

Область представления

A, B - 11ти разрядные целые числа

N, R - 16ти разрядные целые числа

Элементы массива - 16ти разрядные целые числа

Область допустимых значений

Значение элементов массива $[2^{15}; 2^{15} - 1]$

$N \in [1; 127]$

$R \in [0; FFFF]$

$A \in [0; 219 - N] \cup [22F; 7FF - N]$

$B \in [A; A + N - 1]$

Фактическое ОДЗ:

$A \in [0; 215] \cup [22F; 7FB]$

$B \in [22F; 232]$

Таблица трасировки

Адр	Знчн	IP	CR	AR	DR	SP	BR	AC	NZVC	Адр	Знчн
21D	200	21E	200	21D	200	0	021D	0	100		
21E	EEFD	21F	EEFD	21C	0	0	FFFD	0	100	21C	0
21F	AF04	220	AF04	21F	4	0	4	4	0		
220	EEFA	221	EEFA	21B	4	0	FFFA	4	0	21B	4
221	4EF7	222	4EF7	219	022F	0	FFF7	233	0		
222	EEF7	223	EEF7	21A	233	0	FFF7	233	0	21A	233
223	ABF6	224	ABF6	232	1003	0	FFF6	1003	0	21A	232
224	480	225	480	224	480	0	224	801	11		
225	F406	22C	F406	225	F406	0	6	801	11		
22C	821B	22D	821B	21B	3	0	2	801	11	21B	3
22D	CEF5	223	CEF5	22D	223	0	FFF5	801	11		
223	ABF6	224	ABF6	231	480	0	FFF6	480	1	21A	231
224	480	225	480	224	480	0	224	8240	1010		
225	F406	226	F406	225	F406	0	225	8240	1010		
226	480	227	480	226	480	0	226	4120	0		
227	F404	228	F404	227	F404	0	227	4120	0		
228	400	229	400	228	400	0	228	8240	1010		
229	400	22A	400	229	400	0	229	480	11		
22A	4EF1	22B	4EF1	21C	0	0	FFF1	480	0		
22B	EEF0	22C	EEF0	21C	480	0	FFF0	480	0	21C	480
22C	821B	22D	821B	21B	2	0	1	480	0	21B	2
22D	CEF5	223	CEF5	22D	223	0	FFF5	480	0		
223	ABF6	224	ABF6	230	E21C	0	FFF6	E21C	1000	21A	230
224	480	225	480	224	480	0	224	710E	0		
225	F406	226	F406	225	F406	0	225	710E	0		
226	480	227	480	226	480	0	226	3887	0		
227	F404	228	F404	227	F404	0	227	3887	0		
228	400	229	400	228	400	0	228	710E	0		
229	400	22A	400	229	400	0	229	E21C	1010		

Адр	Знчн	IP	CR	AR	DR	SP	BR	AC	NZVC	Адр	Знчн
22A	4EF1	22B	4EF1	21C	480	0	FFF1	E69C	1000		
22B	EEF0	22C	EEF0	21C	E69C	0	FFF0	E69C	1000	21C	E69C
22C	821B	22D	821B	21B	1	0	0	E69C	1000	21B	1
22D	CEF5	223	CEF5	22D	223	0	FFF5	E69C	1000		
223	ABF6	224	ABF6	22F	F403	0	FFF6	F403	1000	21A	022F
224	480	225	480	224	480	0	224	7A01	11		
225	F406	22C	F406	225	F406	0	6	7A01	11		
22C	821B	22E	821B	21B	0	0	FFFF	7A01	11	21B	0
22E	100	22F	100	22E	100	0	022E	7A01	11		

Вывод

Благодаря данной работе я научился работать с циклами, одномерными массивами, изучил цикл выполнения таких команд как LOOP и JUMP.