

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

Факультет Программной Инженерии и Компьютерной Техники

Лабораторная работа №3

Выполнение циклических программ

Вариант 1605

Выполнил:

Григорьев Даниил Александрович

Группа Р3116

Преподаватель:

Ткешелашвили Нино Мерабиевна

Санкт-Петербург 2024

Содержание

Задание	3
Функция	<i>Ошибка! Закладка не определена.</i>
ОП и ОДЗ	5
Область представления:	Ошибка! Закладка не определена.
Область допустимых значений	Ошибка! Закладка не определена.
Трассировка программы	7
Вариант с меньшим числом команд	<i>Ошибка! Закладка не определена.</i>
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.

20F:	0223		21D:	7EF4
210:	0200		21E:	F801
211:	4000		21F:	EEF2
212:	0200		220:	8211
213:	+ AF40		221:	CEF9
214:	0680		222:	0100
215:	0500		223:	02A8
216:	EEFB		224:	0C00
217:	AF05		225:	0706
218:	EEF8		226:	1000
219:	AEF5		227:	0315
21A:	EEF5			
21B:	AAF4			
21C:	F303			

Таблица команд

Адрес	Код команды	Мнемоника	Комментарии
20F	0223	A	Указатель на первый элемент массива
210	0200	B	Указатель на элемент массива
211	4000	C	Счетчик (кол-во элементов массива)
212	0200	R	Результат
213	+AF40	LD #40	Прямая загрузка 0040 -> AC
214	0680	SWAB	Обмен младшего и старшего байта (AC7...AC0 <-> AC 15...AC8)
215	0500	ASL	AC сдвигается влево, AC15 -> C, C -> AC0
216	EEFB	ST (IP-5)	Прямое относительное сохранение (AC -> (212))
217	AF05	LD #5	Прямая загрузка 0005 -> AC
218	EEF8	ST (IP-8)	Прямое относительное сохранение (AC -> M (211))
219	AEF5	LD (IP-11)	Прямая относительная загрузка (M (20F) -> AC)
21A	EEF5	ST (IP-11)	Прямое относительное сохранение (AC -> M (210))
21B	AAF4	LD (IP-12)+	Косвенная автоинкрементная загрузка (M (210) -> AC, MEM (210) += 1)
21C	F303	BPL (IP + 3)	Если N==0, то IP + 3 + 1 -> IP
21D	7EF4	CMP (IP-12)	Установить флаги по результату AC – M (212)
21E	F801	BLT (IP + 1)	Если N xor V == 1 (N != V), то IP + 1 + 1 -> IP
21F	EEF2	ST (IP-14)	Прямое относительное сохранение (AC -> M (212))
220	8211	LOOP 211	M (211) -1 -> M (211), Если M (211) <= 0, то IP + 1 -> IP
221	CEF9	JUMP (IP-7)	Прямой относительный прыжок IP – 7 + 1 -> IP
222	0100	HLT	Останов
223	02A8	-	Числа массива
224	0C00	-	
225	0706	-	
226	1000	-	
227	0315	-	

Описание программы

Программа выполняет поиск и сохранение в ячейку 212 максимальное отрицательное число массива

(если в массиве отрицательных чисел нет, то в 212 ячейке будет 8000)

Расположение в памяти БЭВМ программы, исходных данных и результатов:

20F (A) – указатель на первый элемент массива

210 (B) – указатель на текущий элемент массива

211 (C) – количество элементов массива (по нему идет цикл)

212 (R) – ячейка результата (будет содержаться максимальное отрицательное число массива либо 8000, если отрицательных чисел в массиве нет)

223 – 227 – это массив

213 – 222 – это исполняемые инструкции

Область представления

A, B – 11 разрядные, адрес БВЭМ

C – 8 разрядное знаковое целое число

R – 16 разрядное знаковое целое число

Числа массива – 16 разрядные знаковые целые числа

Область допустимых значений

$C \in [1, 127]$, $R \in [-32768, 32767]$

A, B $\in [0, 20F-C]$ объедин [223, 7FF], Если $arr[i] \in [-32768, 32767]$

Адреса первой и последней исполняемых команд

213 – адрес первой команды

222 – адрес последней исполняемой команды

Полученные числа и программа

Adr = 200
 N = 3
 Arr = [19, -44, -7]
 R = -7

Адрес	Код команды	Мнемоника	Комментарии
200*	0013	Arr[0]	Числа массива
211*	FFD4	Arr[1]	
212*	FFF9	Arr[2]	
...			
20F*	0200	A	Указатель на первый элемент массива
210	0200	B	Указатель на элемент массива
211	4000	C	Счетчик (кол-во элементов массива)
212	0200	R	Результат
213	+AF40	LD #40	Прямая загрузка 0040 -> AC
214	0680	SWAB	Обмен младшего и старшего байта (AC7...AC0 <-> AC 15...AC8)
215	0500	ASL	AC сдвигается влево, AC15 -> C, C -> AC0
216	EEFB	ST (IP-5)	Прямое относительное сохранение (AC -> (212))
217*	AF03	LD #5	Прямая загрузка 0005 -> AC
218	EEF8	ST (IP-8)	Прямое относительное сохранение (AC -> M (211))
219	AEF5	LD (IP-11)	Прямая относительная загрузка (M (20F) -> AC)
21A	EEF5	ST (IP-11)	Прямое относительное сохранение (AC -> M (210))
21B	AAF4	LD (IP-12)+	Косвенная автоинкрементная загрузка (M (210) -> AC, MEM (210) += 1)
21C	F303	BPL (IP + 3)	Если N==0, то IP + 3 + 1 -> IP
21D	7EF4	CMP (IP-12)	Установить флаги по результату AC – M (212)
21E	F801	BLT (IP + 1)	Если N xor V == 1 (N != V), то IP + 1 + 1 -> IP
21F	EEF2	ST (IP-14)	Прямое относительное сохранение (AC -> M (212))
220	8211	LOOP 211	M (211) -1 -> M (211), Если M (211) <= 0, то IP + 1 -> IP
221	CEF9	JUMP (IP-7)	Прямой относительный прыжок IP – 7 + 1 -> IP
222	0100	HLT	Останов

Трассировка программы

Выполняемая команда		Содержимое регистров процессора после выполнения команды									Ячейка, содержимое которой изменилось после выполнения команды	
Адрес	Код команды	IP	CR	AR	DR	SP	BR	AC	PS	NZVC	Адрес	Новый код
213	AF40	214	AF40	213	0040	0	0040	0040	000	0000		
214	0680	215	0680	214	0680	0	0214	4000	000	0000		
215	0500	216	0500	215	4000	0	0215	8000	00A	1010		
216	EEFB	217	EEFB	212	8000	0	FFFB	8000	00A	1010	212	8000
217	AF05	218	AF05	217	0005	0	0005	0005	000	0000		
218	EEF8	219	EEF8	211	0005	0	FFF8	0005	000	0000	211	0005
219	AEF5	21A	AEF5	20F	0200	0	FFF5	0200	000	0000		
21A	EEF5	21B	EEF5	210	0200	0	FFF5	0200	000	0000	210	0200
21B	AAF4	21C	AAF4	200	0013	0	FFF4	0013	000	0000	210	0201
21C	F303	220	F303	21C	F303	0	0003	0013	000	0000		
220	8211	221	8211	211	0004	0	0003	0013	000	0000	211	0004
221	CEF9	21B	CEF9	221	021B	0	FFF9	0013	000	0000		
21B	AAF4	21C	AAF4	201	FFD4	0	FFF4	FFD4	008	1000	210	0202
21C	F303	21D	F303	21C	F303	0	021C	FFD4	008	1000		
21D	7EF4	21E	7EF4	212	8000	0	FFF4	FFD4	001	0001		
21E	F801	21F	F801	21E	F801	0	021E	FFD4	001	0001		
21F	EEF2	220	EEF2	212	FFD4	0	FFF2	FFD4	001	0001	212	FFD4
220	8211	221	8211	211	0003	0	0002	FFD4	001	0001	211	0003
221	CEF9	21B	CEF9	221	021B	0	FFF9	FFD4	001	0001		
21B	AAF4	21C	AAF4	202	FFF9	0	FFF4	FFF9	009	1001	210	0203
21C	F303	21D	F303	21C	F303	0	021C	FFF9	009	1001		
21D	7EF4	21E	7EF4	212	FFD4	0	FFF4	FFF9	001	0001		
21E	F801	21F	F801	21E	F801	0	021E	FFF9	001	0001		
21F	EEF2	220	EEF2	212	FFF9	0	FFF2	FFF9	001	0001	212	FFF9
220	8211	221	8211	211	0002	0	0001	FFF9	001	0001	211	0002
221	CEF9	21B	CEF9	221	021B	0	FFF9	FFF9	001	0001		
21B	AAF4	21C	AAF4	203	0000	0	FFF4	0000	005	0101	210	0204
21C	F303	220	F303	21C	F303	0	0003	0000	005	0101		
220	8211	221	8211	211	0001	0	0000	0000	005	0101	211	0001
221	CEF9	21B	CEF9	221	021B	0	FFF9	0000	005	0101		
21B	AAF4	21C	AAF4	204	0000	0	FFF4	0000	005	0101	210	0205
21C	F303	220	F303	21C	F303	0	0003	0000	005	0101		
220	8211	222	8211	211	0000	0	FFFF	0000	005	0101	211	0000
222	0100	223	0100	222	0100	0	0222	0000	005	0101		

Вывод

В ходе работы над лабораторной работой я изучил, как работать в БЭВМ с массивами, а также с переадресацией, циклами и ЛУМРами. Попробовал поработать с ветвлениями и изучил их метод взаимодействия в программе.