

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

Факультет программной инженерии и компьютерной техники

Лабораторная работа №3

По дисциплине

“Основы профессиональной деятельности”

Вариант: 1080

Выполнил:
Ахроров Кароматуллохон Фирдавсович

Группа: Р3110

Преподаватель:
Блохина Елена Николаевна

Санкт-Петербург, 2024г

Содержание

ЗАДАНИЕ	2
Ход работы.....	3
Описание Программы.....	3
Область представления	4
Область допустимых значений	4
Расположение данных в памяти.....	4
Адреса первой и последней выполняемой команды.....	4
Таблица трассировки	4
Вывод	6

Задание

По выданному преподавателем варианту определить функцию, вычисляемую программой, область представления и область допустимых значений исходных данных и результата, выполнить трассировку программы, предложить вариант с меньшим числом команд. При выполнении работы представлять результат и все операнды арифметических операций знаковыми числами, а логических операций набором из шестнадцати логических значений.

3A0:	03B4		3AE:	7EF4
3A1:	A000		3AF:	F801
3A2:	E000		3B0:	EEF2
3A3:	E000		3B1:	83A2
3A4:	+ AF40		3B2:	CEF9
3A5:	0680		3B3:	0100
3A6:	0500		3B4:	0800
3A7:	EEFB		3B5:	0000
3A8:	AF04		3B6:	0000
3A9:	EEF8		3B7:	F000
3AA:	AEF5			
3AB:	EEF5			
3AC:	AAF4			
3AD:	F003			

1. Ход работы

Текст исходной программы

Адрес	Код команды	Мнемоника	Комментарий
3A4	AF40	LD 40	Прямая загрузка 0040 в АС
3A5	0680	SWAB	Обмен
3A6	0500	ASL	Сдвиг влево
3A7	EEFB	ST (IP-5)	Прямое относительное Сохранение АС в ячейку по адресу IP-5(мин число)
3A8	AF04	LD #04	Прямая загрузка 0004 в АС
3A9	EEF8	ST (IP-8)	Прямое относительное Сохранение АС в ячейку по адресу IP-8(Кол-во элементов массива 4)
3AA	AEF5	LD (IP-11)	Прямая относительная загрузка в АС по адресу IP-11
3AB	EEF5	ST (IP-11)	Прямое относительное Сохранение АС в ячейку по адресу IP-11(в R)
3AC	AAF4	LD (IP-12) +	Косвенная авто инкрементальная загрузка: MEM(IP-12) +=1; MEM(M) -> АС (в адрес текущего элемента)
3AD	F003	BEQ (IP+1)	Если Z == 1, то IP = IP + 1 -> IP
3AE	7EF4	CMP (IP-12)	Флаги по результату АС-R
3AF	F801	BLT IP+1	Если (N≠V == 1 / N!=V), то IP = IP + 1 -> IP
3B0	EEF2	ST (IP-14)	Прямое относительное Сохранение АС в ячейку по адресу IP-14(в R)
3B1	83A2	LOOP 3A2	MEM(3A2) – 1 -> MEM(3A2); Если MEM(3A2) <= 0, то IP + 1 -> IP
3B2	CEF9	JUMP (IP-6)	Прямой относительный прыжок IP-6 -> IP
3B3	0100	HLT	Остановка

2. Описание программы

Программа ищет **максимальный ненулевой элемент** массива из n элементов (хранящихся в памяти по некоторому указателю).

Формула результата

Пусть массив x_1, x_2, \dots, x_n . Тогда

$result = \max(\{-2^{15}\} \cup \{x_i | x_i \neq 0\})$, если существует хотя бы один $x_i \neq 0$, если все $x_i = 0$.

3. Область представления

1. **arr_ptr** (3A1) — 16-разрядный адрес (беззнаковый), указывающий на первый элемент массива.
2. **arr_length** (3A2) — 16-разрядное целое число, используется как счётчик длины массива (беззнаковое).
3. **result** (3A3) — 16-разрядное **знаковое** целое (изначально $0x8000 = -2^{15}$).
4. **arr[i]** — 16-разрядные **знаковые** целые числа, диапазон значений $[-2^{15}..+2^{15}-1]$.

5. Область допустимых значений

1. **arr_length** = 4
 - Чтобы цикл корректно завершился и не выходил за границы памяти, n должно быть в диапазоне $1 \leq n \leq 4$.
 - Если $n=0$, программа может завершиться сразу, но тогда результат останется -2^{15} .
2. **arr[i]** $\in [-2^{15}..+2^{15}-1]$ $[0x000..0x39F]$ \cup $[0x3B4..0x7FF]$
 - При любом значении элемент остаётся валидным, так как программа лишь проверяет «равно ли 0» и « $AC \geq result$ ».
3. **result** (в 3A3) $\in [-2^{15}..+2^{15}-1]$
 - По ходу работы это «плавающее» знаковое число в 16 битах. Изначально -2^{15} .
 - На выходе оно либо останется -2^{15} , если все элементы были 0, либо будет равно какому-то ненулевому значению массива.
4. **Указатель arr_ptr** (3A1) должен указывать на область памяти, где лежат n элементов. То есть $arr_ptr+(n-1)$ не выходит за «легальную» зону памяти машины (например, $[0x000..0x7FF]$ в 16-битной адресации).

5. Расположение данных в памяти

- 3A0, 3A2, 38A, 3B4, 3B5, 3B6, 3B7 – исходные данные;
- 3A1 – промежуточный результат;
- 3A3 – итоговый результат;
- 3A4 – 3B3 – команды

6. Адреса первой и последней выполняемой команды

- Адрес первой команды: 3A4
- Адрес последней команды: 3B3

7. Таблица трассировки

Arr[0]=0, Arr[1]=-2, Arr[2]=4, Arr[3]=96, arr_length=4

Выполняемая команда		Содержание регистров в процессоре после выполнения команды								Ячейка, содержимое которой изменилось после выполнения команды	
Адрес	Код	IP	CR	AR	DR	SP	BR	AC	NZVC	Адрес	Новый код
3A4	AF40	3A5	AF40	3A4	0040	000	0040	0040	0000		
3A5	0680	3A6	0680	3A5	0680	000	03A5	4000	0000		
3A6	0500	3A7	0500	3A6	4000	000	03A6	8000	1010		
3A7	EEFB	3A8	EEFB	3A3	8000	000	FFFB	8000	1010	3A3	8000
3A8	AF04	3A9	AF04	3A8	0004	000	0004	0004	0000		
3A9	EEF8	3AA	EEF8	3A2	0004	000	FFF8	0004	0000	3A2	0004
3AA	AEF5	3AB	AEF5	3A0	03B4	000	FFF5	03B4	0000		
3AB	EEF5	3AC	EEF5	3A1	03B4	000	FFF5	03B4	0000	3A1	03B4
3AC	AAF4	3AD	AAF4	3B4	0000	000	FFF4	0000	0100	3A1	03B5
3AD	F003	3B1	F003	3AD	F003	000	0003	0000	0100		
3B1	83A2	3B2	83A2	3A2	0003	000	0002	0000	0100	3A2	0003
3B2	CEF9	3AC	CEF9	3B2	03AC	000	FFF9	0000	0100		
3AC	AAF4	3AD	AAF4	3B5	FFFE	000	FFF4	FFFE	1000	3A1	03B6
3AD	F003	3AE	F003	3AD	F003	000	03AD	FFFE	1000		
3AE	7EF4	3AF	7EF4	3A3	8000	000	FFF4	FFFE	0001		
3AF	F801	3B0	F801	3AF	F801	000	03AF	FFFE	0001		
3B0	EEF2	3B1	EEF2	3A3	FFFE	000	FFF2	FFFE	0001	3A3	FFFE
3B1	83A2	3B2	83A2	3A2	0002	000	0001	FFFE	0001	3A2	0002
3B2	CEF9	3AC	CEF9	3B2	03AC	000	FFF9	FFFE	0001		
3AC	AAF4	3AD	AAF4	3B6	0004	000	FFF4	0004	0001	3A1	03B7
3AD	F003	3AE	F003	3AD	F003	000	03AD	0004	0001		
3AE	7EF4	3AF	7EF4	3A3	FFFE	000	FFF4	0004	0000		
3AF	F801	3B0	F801	3AF	F801	000	03AF	0004	0000		
3B0	EEF2	3B1	EEF2	3A3	0004	000	FFF2	0004	0000	3A3	0004
3B1	83A2	3B2	83A2	3A2	0001	000	0000	0004	0000	3A2	0001
3B2	CEF9	3AC	CEF9	3B2	03AC	000	FFF9	0004	0000		
3AC	AAF4	3AD	AAF4	3B7	0060	000	FFF4	0060	0000	3A1	03B8
3AD	F003	3AE	F003	3AD	F003	000	03AD	0060	0000		
3AE	7EF4	3AF	7EF4	3A3	0004	000	FFF4	0060	0001		
3AF	F801	3B0	F801	3AF	F801	000	03AF	0060	0001		
3B0	EEF2	3B1	EEF2	3A3	0060	000	FFF2	0060	0001	3A3	0060
3B1	83A2	3B3	83A2	3A2	0000	000	FFFF	0060	0001	3A2	0000
3B3	0100	3B4	0100	3B3	0100	000	03B3	0060	0001		

8. Вывод

Во время выполнения лабораторной работы я научился работать в БЭВМ с массивами, ветвлением и циклами. Я изучил прямую и косвенную адресацию и цикл выполнения таких команд, как LOOP и JUMP. Так же научился потактовое чтение команд.

AA04

IF

Ip-> Br,Ar

Br+1-> IP,Mem(ar)->Dr

Dr->CR

AF

SXT_Cr(07)->Br

Br+IP->Ar

Mem(Ar)->Dr

Dr + 1-> Mem(ar)

Dr-1-> Dr

LOOP

OF

Dr->Ar

Mem(ar)->Dr

Exe

Dr-1->Dr, Dr-> Mem(ar)

Dr-1-> Br

Если Br(15)==1 то Ip+1