#### МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

# ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики»

### ФАКУЛЬТЕТ ПРОГРАММНОЙ ИНЖЕНЕРИИ И КОМПЬЮТЕРНОЙ ТЕХНИКИ

#### ЛАБОРАТОРНАЯ РАБОТА №3

по дисциплине «Основы профессиональной деятельности»

Вариант № 3104

#### Выполнил:

Студент группы Р3131 Дворкин Борис Александрович

### Преподаватель:

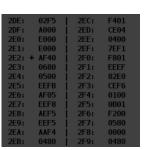
Клименков Сергей Викторович

# Содержание

Текст задания	3
Описание программы	4
Таблица трассировки	
Вывод	
<u>DDIDOZ</u>	

## Текст задания

Восстановить текст заданного варианта программы, определить предназначение и описание программы, определить область представления и ОДЗ исходных данных и результата, выполнить трассировку программы.



Адрес	Код команды	Мнемоника	Комментарии
2DE	02F5	arr_first_elem	Адрес первого элемента
2DF	A000	arr_last_elem	Адрес текущего элемента (начиная с последнего)
2E0	E000	arr_length	Количество элементов массива
2E1	E000	result	Результат
2E2	AF40	LD #0x40	Прямая загрузка 040 -> АС
2E3	0680	SWAB	Обмен ст. и мл. байтов в АС
2E4	0500	ASL	Арифметический сдвиг АС влево
2E5	EEFB	ST IP-5	Прямое относительное сохранение АС -> М (2Е1)
2E6	AF05	LD #0x05	Прямая загрузка 005 -> АС
2E7	EEF8	ST IP-8	Прямое относительное сохранение АС -> М (2Е0)
2E8	AEF5	LD IP-B	Прямая относительная загрузка 02F5 -> AC
2E9	EEF5	ST IP-B	Прямое относительное сохранение AC -> M (2DE)
2EA	AAF4	LD (IP-C)+	Косвенная относительная автоинкрементная загрузка: 3н(2DE) += 1; 3н(2DE) -> AC
2EB	0480	ROR	Циклический сдвиг АС вправо
2EC	F401	BCS IP+1	Если C==1, то IP+1+1 -> IP
2ED	CE04	BR IP+4	IP+4+1 -> IP
2EE	0400	ROL	Циклический сдвиг АС влево
2EF	7EF1	CMP IP-F	Прямое относительное сравнение АС-М(2Е1)
2F0	F801	BLT IP+1	Если N⊕V==1, то IP+1+1 -> IP
2F1	EEEF	ST IP-11	Прямое относительное сохранение АС -> М (2Е0)
2F2	82E0	LOOP 0x2E0	M(2E0) - 1 -> M(2E0); Если (2E0) <= 0, то IP + 1 -> IP
2F3	CEF6	BR IP-A	Переход IP-A+1 -> IP
2F4	0100	HLT	Останов
2F5	0B01		
2F6	F200		
2F7	0580		"Элементы массива"
2F8	0000		
2F9	0480		

### Описание программы

Программа находит максимальный нечётный элемент массива и сохраняет информацию о нём в биты ячейки результата. Формула результата:

$$_{\text{MEM(2E1)}} = \sum_{i=0}^{\text{MEM(2E0)}} \begin{cases} 2^i \text{ если MEM } (2F5+i) \vdots 2 \\ 0 \text{ если MEM } (2F5+i) \vdots 2 \end{cases}$$

### Область представления

- 🛮 arr\_first\_elem, arr\_last\_elem, arr\_length, result 16-ти разрядные целые числа в прямом коде.
- 🛮 arr[i] 16-ти разрядные целые числа в дополнительном коде

### Область допустимых значений

- $\square$  arr\_length  $\in$  [1; 16] (т. к. при arr\_length > 16 битов результата не будет хватать для данных о делимости элементах и он будет ошибочным)
- $\square$  result  $\in [0; 2^{16} 1]$  (T. K. max = 1 + 2 + 4 + ... +  $2^{15}$ )
- 2 arr\_first\_elem ε [0; 2DE arr\_length] υ [2F5; 7FF arr\_length]
- ② arr\_last\_elem ε [arr\_first\_elem; arr\_first\_elem + arr\_length 1]
- $\square$  Элементы массива arr[i]  $\epsilon$  [-32768; 32767] (т. е. [-2<sup>15</sup>; 2<sup>15</sup>-1])

### Расположение данных в памяти

- 2 2DE, 2DF, 2E0, 2F5, 2F6, 2F7, 2F8, 2F9 исходные данные;
- ☑ 2DE промежуточный результат;
- 2E1 итоговый результат;
- 2E2 2F4 команды.

### Адреса первой и последней выполняемой команды

- 🛮 Адрес первой команды: 2Е2
- Адрес последней команды: 2F4

Таблица трассировки: gimme numbers pls 0-0

### Вывод

Во время выполнения лабораторной работы я научился работать в БЭВМ с массивами, ветвлением и циклами. Я изучил различные виды адрессаций и цикл выполнения таких команд, как LOOP и JUMP.