МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«Национальный исследовательский университет ИТМО»

ФАКУЛЬТЕТ ПРОГРАММНОЙ ИНЖЕНЕРИИ И  
КОМПЬЮТЕРНОЙ ТЕХНИКИ

**ЛАБОРАТОРНАЯ РАБОТА №6**

по дисциплине

«Основы профессиональной деятельности»

Вариант № 86866

***Выполнил:***Студент группы P3107  
 Пшеничников Артём Дмитриевич

***Принял:***Осипов Святослав Владимирович

**Содержание**

[Задание (Вариант 55655) 3](#_gjdgxs)

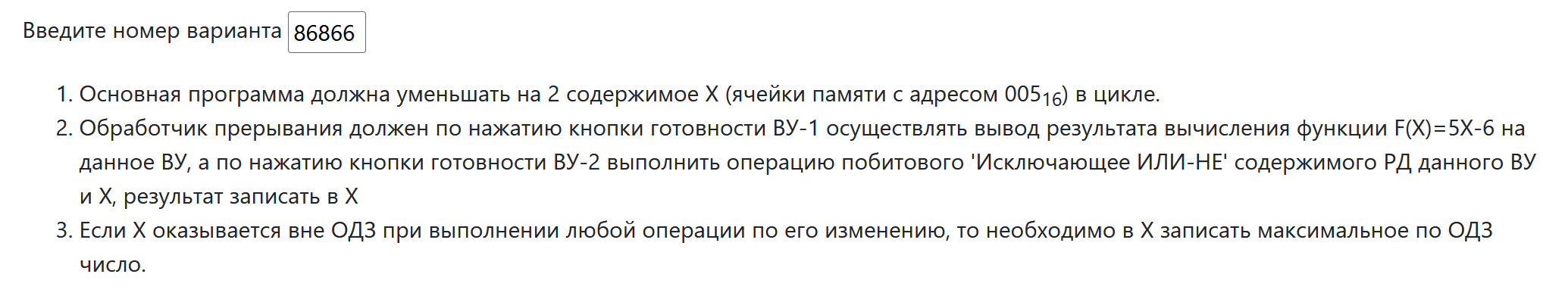
[Текст исходной программы 4](#_arj93mk5uiji)

[Описание программы 6](#_wgcg15252s2g)

[Тестирование 7](#_qllcfdbjyetk)

[Заключение](#_3znysh7) 8

# **Задание (Вариант 86866)**



# 

# **Текст исходной программы**

ORG 0x0  
V0: WORD $DEFAULT, 0x180 ; Вектор прерывания #0  
V1: WORD $INT\_1, 0x180 ; Вектор прерывания #1  
V2: WORD $INT\_2, 0x180 ; Вектор прерывания #2  
V3: WORD $DEFAULT, 0x180 ; Вектор прерывания #3  
V4: WORD $DEFAULT, 0x180 ; Вектор прерывания #4  
V5: WORD $DEFAULT, 0x180 ; Вектор прерывания #5  
V6: WORD $DEFAULT, 0x180 ; Вектор прерывания #6  
V7: WORD $DEFAULT, 0x180 ; Вектор прерывания #7  
  
DEFAULT: IRET ; Дефолтный вектор  
  
X: WORD 0x084 ; Адрес ячейки X (по заданию 5, но оно пересекается с векторами)  
start\_value: WORD 0x0042 ; стартовое значение X  
  
START: NOP  
 LD start\_value ; записали стартовое значение в X  
 CALL clamp\_var  
 ST (X)  
 ;  
 DI ; запрет прерываний  
 LD #0x9 ; Разрешение прерываний и вектор 1  
 OUT 0x3 ; ВУ-1 на вектор INT\_1  
 LD #0xA ; Разрешение прерываний и вектор 2  
 OUT 0x5 ; ВУ-2 на вектор INT\_2  
 EI ; Прерываемся  
 ;  
main: DI  
 LD (X)  
 SUB #0x02  
 CALL clamp\_var  
 ST (X)  
 OUT 0x06 ; добавлен вывод текущего X на ВУ-3  
 EI  
 NOP ; прерываемся только здесь  
 JUMP main  
  
  
min\_var: WORD 0x0000  
max\_var: WORD 0x00FF  
clamp\_var: CMP max\_var ; ограничение переменной в AC от min\_var до max\_var  
 BNC fix\_max\_var ; если максимум строго меньше значения  
 DEC  
 CMP min\_var  
 BNS fix\_min\_var ; если минимум больше или равен значению-1 => строго больше  
 INC  
 JUMP end\_clamp\_var  
fix\_max\_var: LD max\_var  
 JUMP end\_clamp\_var  
fix\_min\_var: LD max\_var ; LD min\_var ; по заданию пишем максимальное число  
 JUMP end\_clamp\_var  
end\_clamp\_var: RET  
  
  
  
cntr: WORD ?  
var\_a: WORD 0x05  
var\_b: WORD 0x06  
calculate\_func: LD var\_a ; Подсчёт функции a\*X - b, X - в стеке  
 ST cntr ; записали на сколько умножить  
 CLA  
loop1: ADD &1 ; сложили в цикле  
 LOOP cntr  
 JUMP loop1  
 SUB var\_b ; минус b  
 ST &1 ; записали в стек  
 RET  
  
buffer: WORD ? ; ; 1 2 3 4 5  
not\_xor: NOP ; исключающее ИЛИ-НЕ (00-1, 01-0, 10-0, 11-1) (not ((a or b) and (not (a and b))))  
 LD &1  
 NOP ; Breakpoint  
 AND &2 ; 5  
 NOT ; 4  
 ST buffer  
 LD &1  
 OR &2 ; 2  
 AND buffer ; 3  
 NOT ; 1  
 ST &2 ; запись в стек  
 POP  
 SWAP ; убрали ненужный аргумент  
 RET  
  
  
 ; прерывание по ВУ-1  
INT\_1: NOP ; подсчёт функции и вывод на ВУ-1  
 PUSH  
 LD (X) ; загружаем X в AC  
 PUSH  
 NOP ; Breakpoint-1-1  
 CALL calculate\_func ; Подсчёт функции  
 POP  
 NOP ; Breakpoint-1-2  
 CALL clamp\_var  
 OUT 0x02 ; вывод на ВУ-1  
 NOP ; Breakpoint-1-3  
 POP  
 IRET  
  
mask: WORD 0x00FF ; прерывание по ВУ-2  
INT\_2: NOP ; подсчёт исключающего ИЛИ-НЕ и запись в X  
 PUSH  
 LD (X) ; загружаем X в AC  
 PUSH  
 NOP ; Breakpoint-2-1  
 CLA  
 IN 0x04  
 SXTB  
 PUSH  
 NOP ; Breakpoint-2-2  
 CALL not\_xor ; Подсчёт исключающего ИЛИ-НЕ  
 POP  
 NOP ; Breakpoint-2-3  
 ; AND #0xFF ; обрежем мусор почему не работает?  
 AND mask ; обрежем мусор  
 CALL clamp\_var  
 ST (X)  
 NOP ; Breakpoint-2-4  
 POP  
 IRET

# **Описание программы**

**Назначение программы:**

1. Декрементация ячейке X на 2 в цикле
2. По готовности ВУ-1 вывод на ВУ-1 результата F(X)=5X-6
3. По готовности ВУ-2 запись в X NOT-XOR(РД2, X)
4. При вычислении X проверять ОДЗ, если X вне ОДЗ, то X = max(ОДЗ)

**Область представления:**

X – адрес ячейки

start**\_**value – начальное значение ячейки X

min**\_**var – нижняя граница ОДЗ

max**\_**var – верхняя граница ОДЗ

**Область определения:**

Последняя исполняемая команда = 0x060 < X < Макс. Размер стека = 0x7FB

0 <= start**\_**value <= 215 – 1 (при записи в X применится ОДЗ)

min**\_**var = 0x0000 (по заданию)

max**\_**var = 0x00FF (по заданию)

Расположение в памяти ЭВМ программы: 000-060 + переменная X

Расположение данных: X

Расположение результата: ВУ-1, X

Первая выполняемая команда: 013

Последняя выполняемая команда: 060

# **Тестирование**

Тест 1: точки Breakpoint-1-x

Запуск 1: 25\*5-6 = 119

1-1: 0019

1-2: 0077

1-3: 0077

Запуск 2: превышено

1-1: 00E3

1-2: 0469

1-3: 00FF

Тест 2: точки Breakpoint-2-x

Запуск 1:

1-1: 00A7 // 1010 0111

1-2: 0057 // 0101 0111

1-3: FF0F // 0000 1111

1-4: 000F

Запуск 2:

1-1: 004F // 0100 1111

1-2: FFDA // 1101 1010

1-3: 006A // 0110 1010

1-4: 006A

# **Заключение**

В ходе выполнения лабораторной работы я освоил принципы работы с прерываниями в БЭВМ, научился управлять изменением данных в памяти, взаимодействовать с внешними устройствами и работать с подпрограммами в прерываниях.