Федеральное государственное автономное образовательное учреждение

высшего образования «Национальный исследовательский университет

ИТМО»

*Факультет программной инженерии и компьютерной техники*

*Направление подготовки: 09.03.04 – Программная инженерия, Системное и прикладное программное обеспечение.*

*Дисциплина «Основы профессиональной деятельности»*

**Отчет**

**По лабораторной работе №6**

**Обмен данными с ВУ по прерыванию**

**Вариант №** **41106**

Выполнил:

Молчанов Фёдор Денисович

Группа: Р3113

Преподаватель:

Блохина Елена Николаевна

Г. Санкт-Петербург, 2024 г.

Оглавление

[Задание 3](#_Toc166521095)

[Код программы 3](#_Toc166521096)

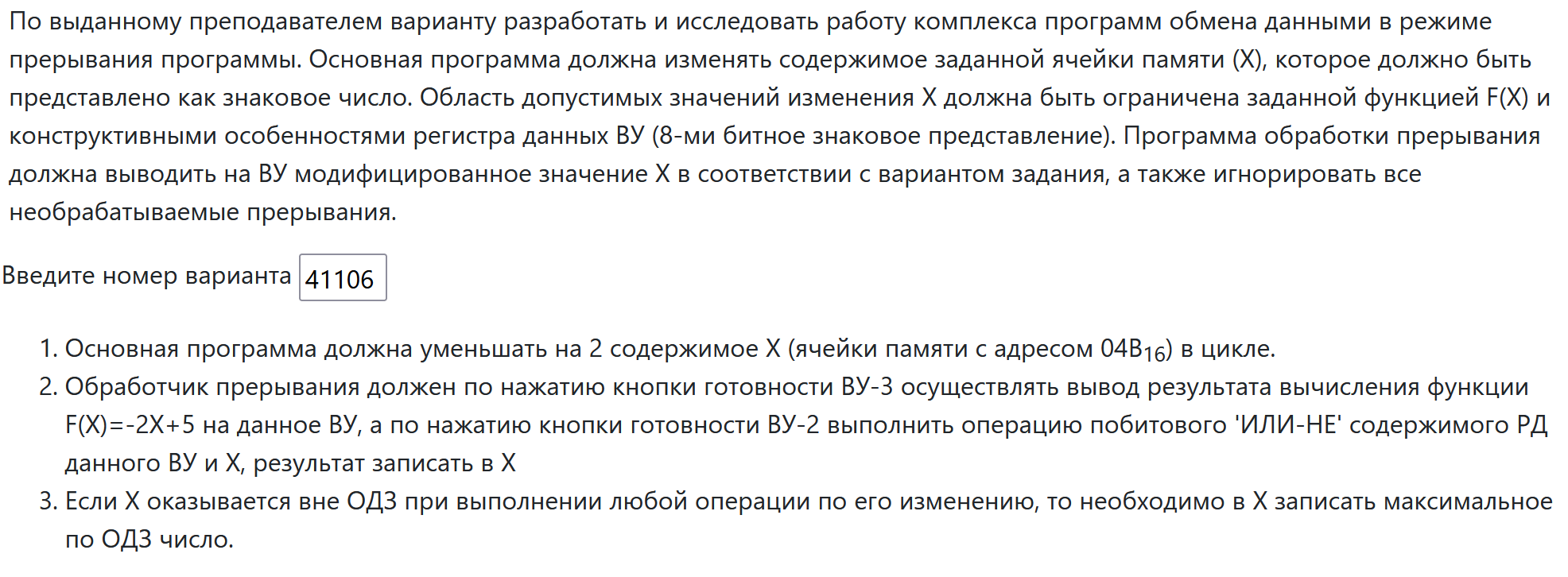
[Описание программы 6](#_Toc166521097)

[Описание и назначение исходных данных 6](#_Toc166521098)

[ОП и ОДЗ исх. Данных 7](#_Toc166521099)

[Методика проверки 7](#_Toc166521100)

# Задание



# Код программы

org 0x0

v0: word $default, 0x180

v1: word $default, 0x180

v2: word $int2, 0x180

v3: word $int3, 0x180

v4: word $default, 0x180

v5: word $default, 0x180

v6: word $default, 0x180

v7: word $default, 0x180

org 0x10

max\_odz: word 0x003A ; 58

min\_odz: word 0x00c5 ;-59

default: iret

org 0x20

start:

di

cla

;запрет прерываний

out 0x1

out 0x3

out 0xB

out 0xD

out 0x11

out 0x15

out 0x19

out 0x1D

;разрешение прерываний для ву2 и ву3

LD #0xA ;(1000 + 0010 = 1010 = A)

out 0x5

LD #0xB ;(1000 + 0011 = 1011 = B)

out 0x7

;надо было как-то инциализировать начало. можно было просто cla

main:

ei

cla

;скорее всего неоптимизированно, ну да ладно, И ТАК СОЙДЕТ)))

declp:

di ;чтобы не прерывалось во время исполнения вычитания и проверки на одз

ld x

sub #0x2

call check

st x

nop ;для отладки. после первого цикла можно сразу вызывать прерывание

ei

jump declp

org 0x4b

x: word 0x004A ;максимальное одз = 58 в 10 СС

org 0x50

int3:

hlt ;проверить AC

st x

asl

neg

add #0x5

hlt

out 0x6

hlt

ld x

;здесь проверка не требуется, так как результат будет выведен на ВУ3

iret

int2:

;"побитовое или-не" это стрелка пирса !(A|B) = NOT (a OR b)

ld x

hlt ;проверить AC перед прерыванием

in 0x4

or $x

not

hlt

call check

st x

iret

check:

nop

cmp min\_odz

blt write\_max

nop

cmp max\_odz

bpl write\_max

nop

ret

write\_max:

ld $max\_odz

st x

ret

# Описание программы

1. Основная программа должна уменьшать на 2 содержимое X (ячейки памяти с адресом 04B16) в цикле.
2. Обработчик прерывания должен по нажатию кнопки готовности ВУ-3 осуществлять вывод результата вычисления функции F(X)=-2X+5 на данное ВУ, a по нажатию кнопки готовности ВУ-2 выполнить операцию побитового 'ИЛИ-НЕ' содержимого РД данного ВУ и Х, результат записать в Х
3. Если Х оказывается вне ОДЗ при выполнении любой операции по его изменению, то необходимо в Х записать максимальное по ОДЗ число.

# Описание и назначение исходных данных

0x0-0xF векторы прерывания, из них:

v2: word $int2, 0x180

v3: word $int3, 0x180

остальные векторы это метки на $default, 0x180

0x10 max\_odz: word 0x007f ; 127 в дес. СС

0x11 min\_odz: word 0xff7f ;-128 в дес. СС

0x4b x: word ? ячейка памяти, над которой производятся операции вычитания и которая записывается в память в определённых случаях (см. описание программы)

# ОП и ОДЗ исх. Данных

ОП:

X, min\_odz, max\_odz: 8-и разрядное знаковое число

DR КВУ2, КВУ3: 8-и разрядное знаковое число

F(x) = -2x + 5

ОДЗ:

# Методика проверки

ВУ-2:

1. Загрузить текст программы в БЭВМ
2. Заменить все NOP на HLT
3. Запустить программу в режиме РАБОТА
4. Дождаться останов
5. Установить желаемое значение для операции «НЕ-ИЛИ» в DR ВУ-2
6. Установить готовность ВУ-2
7. Нажать «ПРОДОЛЖЕНИЕ»
8. Дождаться останов
9. Сверить, ожидаемое ли значение получилось в регистре AC БЭВМ

ВУ-3:

1. Установить готовность ВУ-3
2. Нажать «ПРОДОЛЖЕНИЕ»
3. Дождаться останов
4. Проверить содержимое AC
5. Нажать «ПРОДОЛЖЕНИЕ»
6. Дождаться останов
7. Сверить, ожидаемое ли значение вывелось на ВУ-3