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

Факультет Программной инженерии и компьютерной техники

Дисциплина “Основы профессиональной деятельности”

Отчёт по лабораторной работе №6

Вариант №3970

Выполнил:

Ануфриев Андрей Сергеевич, Р3119

Проверила:

Остапенко Ольга Денисовна

г. Санкт-Петербург

2025 год

Оглавление

[Задание 3](#_Toc191798607)

[Ход выполнения 4](#_Toc191798608)

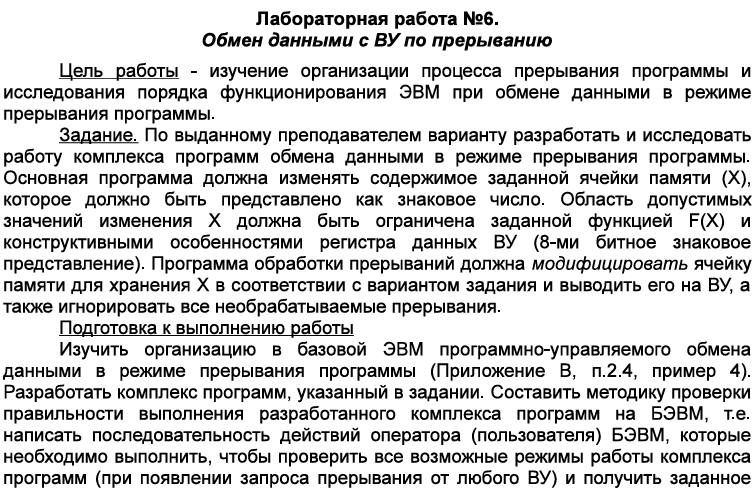
[1) Текст исходной программы 5](#_Toc191798609)

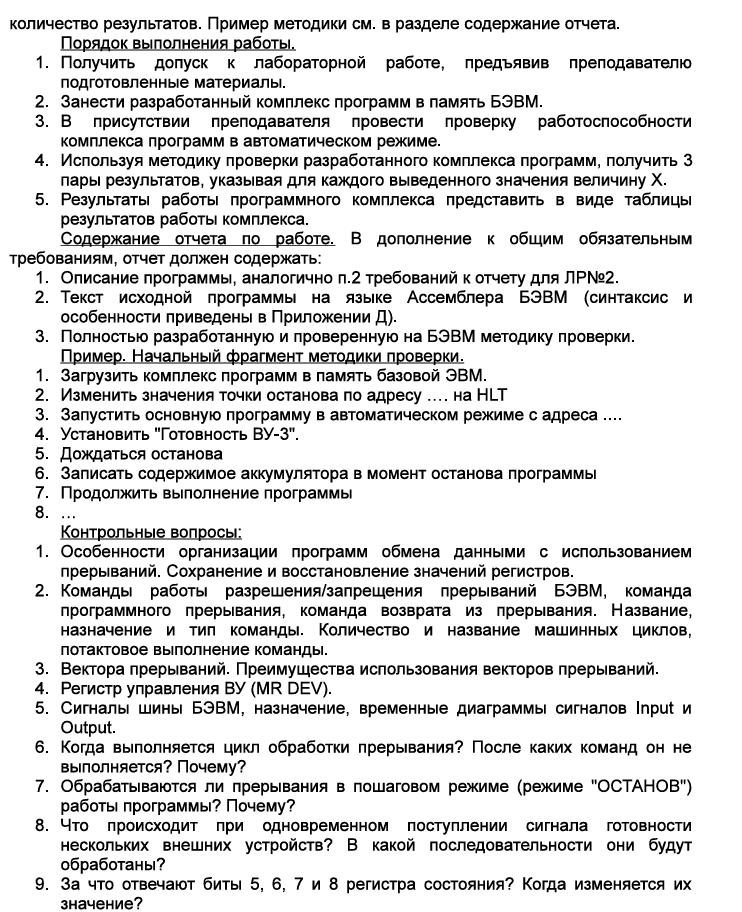
[2) Описание программы 7](#_Toc191798610)

[3)Трассировка 9](#_Toc191798611)

[4)График функции и проверка результата 9](#_Toc191798612)

# Задание





По выданному преподавателем варианту разработать и исследовать работу комплекса программ обмена данными в режиме прерывания программы. Основная программа должна изменять содержимое заданной ячейки памяти (Х), которое должно быть представлено как знаковое число. Область допустимых значений изменения Х должна быть ограничена заданной функцией F(X) и конструктивными особенностями регистра данных ВУ (8-ми битное знаковое представление). Программа обработки прерывания должна выводить на ВУ модифицированное значение Х в соответствии с вариантом задания, а также игнорировать все необрабатываемые прерывания.

1. Основная программа должна инкрементировать содержимое X (ячейки памяти с адресом 02A16) в цикле.
2. Обработчик прерывания должен по нажатию кнопки готовности ВУ-1 осуществлять вывод результата вычисления функции F(X)=-5X+1 на данное ВУ, a по нажатию кнопки готовности ВУ-2 выполнить операцию побитового 'И-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 минимальное по ОДЗ число.

# Ход выполнения

## Текст исходной программы

|  |  |  |
| --- | --- | --- |
|  | ORG 0x0 | *Инициализируем векторы прерываний.* |
| V0: | WORD $default, 0X180 | Вектор 0 – стандартный обработчик. |
| V1: | WORD $int1, 0X180 | Вектор 1 установлен на обр. INT1. |
| V2: | WORD $int2, 0X180 | Вектор 2 установлен на обр. INT2. |
| V3: | WORD $default, 0x180 | Вектор 3 – стандартный обработчик. |
| V4: | WORD $default, 0X180 | Вектор 4 – стандартный обработчик. |
| V5: | WORD $default, 0X180 | Вектор 5 – стандартный обработчик. |
| V6: | WORD $default, 0X180 | Вектор 6 – стандартный обработчик. |
| V7: | WORD $default, 0X180 | Вектор 7 – стандартный обработчик. |
| default: | IRET | просто возвращает из прерывания. |
| X: | WORD ? |  |
| ADDR\_X: | WORD 0x02A | Указатель на X. |
| max: | WORD 0x0019 | максимальное значение Х 25 |
| min: | WORD 0xFFE7 | минимальное значение Х -25 |
|  |  |  |
|  | ORG 0x020 |  |
| START: | DI | На время инициализации векторов запретим какие-либо прерывания. |
|  | CLA | Запрет прерываний для неиспользуемых ВУ |
|  | OUT 0x1 |
|  | OUT 0x7 |
|  | OUT 0xB |
|  | OUT 0xD |
|  | OUT 0x11 |
|  | OUT 0x15 |
|  | OUT 0x19 |
|  | OUT 0x1D |
|  | LD #0x9 | Инициализируем прерывание ВУ-1. |
|  | OUT 3 | на вектор 1 (и разрешим его). |
|  | LD #0xA | Инициализируем прерывание ВУ-2. |
|  | OUT 5 | на вектор 2 (и разрешим его). |
|  | LD 0x002A |  |
|  | ST X |  |
|  | EI | разрешаем прерывание |
|  |  |  |
| main: |  | Запрет прерываний чтобы обеспечить автом. операции |
|  | LD ADDR\_X | загрузили X |
|  | INC |  |
|  | CALL check |  |
|  | ST X |  |
|  |  |  |
|  | JUMP main |  |
|  |  |  |
| int1: | NOP | Обработка прерывания на ВУ-1 |
|  | PUSH | x |
|  | LD X |  |
|  | ASL | 2x |
|  | ASL | 4x |
|  | ADD X | 5x |
|  | NEG | -5x |
|  | INC | -5x+1 |
|  | OUT 2 |  |
|  | POP |  |
|  | HLT |  |
|  | IRET |  |
|  |  |  |
| int2: | PUSH | Обработка прерывания на ВУ-2 |
|  | HLT |  |
|  | IN 4 |  |
|  | AND X |  |
|  | NOT |  |
|  | ST X |  |
|  | POP | (не x) И DR |
|  | HLT |  |
|  | IRET |  |
|  |  |  |
| check: |  | Проверка принадлежности X к ОДЗ |
| check\_min: | CMP min | Если x > min переход на проверку верхней границы |
|  | BPL check\_max |
|  | JUMP ld\_min | Иначе загрузка min в аккумулятор |
| check\_max: | CMP max | Проверка пересечения верхней границы X |
|  | BMI return | Если x < max переход |
| ld\_min: | LD min | Загрузка минимального значения в X |
| return: | RET | Метка возврата из проверки на ОДЗ |

## Описание программы

**Назначение программы:**

Обработка вызываемых прерываний

**Описание исходных данный**:

(Минимальное и максимальное возвращаемое значение из прерывание)

**ОПИ:**

X, min, max – 16 битные знаковые числа

DR ВУ - 8 битное знаковое

**ОДЗ:**

-128<=-5x+1<=127

-129<=-5x<=126

-126<=5x<=129

-25<=x<=25

X [-25, 25] это [FFE7, 0019]

* Расположение в памяти ЭВМ программы и исходных данных:
  + Программа располагается в памяти в ячейках между адресами 138 и 15B включительно (без учета строки).
  + Исходные данные должны располагаться в ячейках памяти:
    - STR – 138.
    - Строка: начиная с адреса STR, заканчивая символом 0x00.
  + В программе также используется вспомогательная ячейка, находящаяся по адресу 0x139.
* Первая команда располагается в ячейке по адресу 13A. Последняя – 15B.

# 3)Трассировка

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Адр | Знчн | IP | CR | AR | DR | SP | BR | AC | PS | NZVC | Адр | Знчн |
| 13A | 0200 | 13A | 0000 | 000 | 0000 | 000 | 0000 | 0000 | 004 | 0100 |  |  |
| 13A | 0200 | 13B | 0200 | 13A | 0200 | 000 | 013A | 0000 | 004 | 0100 |  |  |
| 13B | AEFC | 13C | AEFC | 138 | 05B4 | 000 | FFFC | 05B4 | 000 | 0000 |  |  |
| 13C | EEFC | 13D | EEFC | 139 | 05B4 | 000 | FFFC | 05B4 | 000 | 0000 | 139 | 05B4 |
| 13D | 1203 | 13E | 1203 | 13D | 1203 | 000 | 013D | 0540 | 000 | 0000 |  |  |
| 13E | 2F40 | 13F | 2F40 | 13E | 0040 | 000 | 0040 | 0040 | 000 | 0000 |  |  |
| 13F | F0FD | 140 | F0FD | 13F | F0FD | 000 | 013F | 0040 | 000 | 0000 |  |  |
| 140 | A8F8 | 141 | A8F8 | 5B4 | CFD1 | 000 | FFF8 | CFD1 | 008 | 1000 |  |  |
| 141 | 0680 | 142 | 0680 | 141 | 0680 | 000 | 0141 | D1CF | 008 | 1000 |  |  |
| 142 | 1302 | 143 | 1302 | 142 | 1302 | 000 | 0142 | D1CF | 008 | 1000 |  |  |
| 143 | 7F00 | 144 | 7F00 | 143 | 0000 | 000 | 0000 | D1CF | 009 | 1001 |  |  |
| 144 | F008 | 145 | F008 | 144 | F008 | 000 | 0144 | D1CF | 009 | 1001 |  |  |
| 145 | 1203 | 146 | 1203 | 145 | 1203 | 000 | 0145 | D140 | 009 | 1001 |  |  |
| 146 | 2F40 | 147 | 2F40 | 146 | 0040 | 000 | 0040 | 0040 | 001 | 0001 |  |  |
| 147 | F0FD | 148 | F0FD | 147 | F0FD | 000 | 0147 | 0040 | 001 | 0001 |  |  |
| 148 | AAF0 | 149 | AAF0 | 5B4 | CFD1 | 000 | FFF0 | CFD1 | 009 | 1001 | 139 | 05B5 |
| 149 | 1302 | 14A | 1302 | 149 | 1302 | 000 | 0149 | CFD1 | 009 | 1001 |  |  |
| 14A | 7F00 | 14B | 7F00 | 14A | 0000 | 000 | 0000 | CFD1 | 009 | 1001 |  |  |
| 14B | F001 | 14C | F001 | 14B | F001 | 000 | 014B | CFD1 | 009 | 1001 |  |  |
| 14C | CEF0 | 13D | CEF0 | 14C | 013D | 000 | FFF0 | CFD1 | 009 | 1001 |  |  |
| 13D | 1203 | 13E | 1203 | 13D | 1203 | 000 | 013D | CF40 | 009 | 1001 |  |  |
| 13E | 2F40 | 13F | 2F40 | 13E | 0040 | 000 | 0040 | 0040 | 001 | 0001 |  |  |
| 13F | F0FD | 140 | F0FD | 13F | F0FD | 000 | 013F | 0040 | 001 | 0001 |  |  |
| 140 | A8F8 | 141 | A8F8 | 5B5 | C62C | 000 | FFF8 | C62C | 009 | 1001 |  |  |
| 141 | 0680 | 142 | 0680 | 141 | 0680 | 000 | 0141 | 2CC6 | 001 | 0001 |  |  |
| 142 | 1302 | 143 | 1302 | 142 | 1302 | 000 | 0142 | 2CC6 | 001 | 0001 |  |  |
| 143 | 7F00 | 144 | 7F00 | 143 | 0000 | 000 | 0000 | 2CC6 | 001 | 0001 |  |  |
| 144 | F008 | 145 | F008 | 144 | F008 | 000 | 0144 | 2CC6 | 001 | 0001 |  |  |
| 145 | 1203 | 146 | 1203 | 145 | 1203 | 000 | 0145 | 2C40 | 001 | 0001 |  |  |
| 146 | 2F40 | 147 | 2F40 | 146 | 0040 | 000 | 0040 | 0040 | 001 | 0001 |  |  |
| 147 | F0FD | 148 | F0FD | 147 | F0FD | 000 | 0147 | 0040 | 001 | 0001 |  |  |
| 148 | AAF0 | 149 | AAF0 | 5B5 | C62C | 000 | FFF0 | C62C | 009 | 1001 | 139 | 05B6 |
| 149 | 1302 | 14A | 1302 | 149 | 1302 | 000 | 0149 | C62C | 009 | 1001 |  |  |
| 14A | 7F00 | 14B | 7F00 | 14A | 0000 | 000 | 0000 | C62C | 009 | 1001 |  |  |
| 14B | F001 | 14C | F001 | 14B | F001 | 000 | 014B | C62C | 009 | 1001 |  |  |
| 14C | CEF0 | 13D | CEF0 | 14C | 013D | 000 | FFF0 | C62C | 009 | 1001 |  |  |
| 13D | 1203 | 13E | 1203 | 13D | 1203 | 000 | 013D | C640 | 009 | 1001 |  |  |
| 13E | 2F40 | 13F | 2F40 | 13E | 0040 | 000 | 0040 | 0040 | 001 | 0001 |  |  |
| 13F | F0FD | 140 | F0FD | 13F | F0FD | 000 | 013F | 0040 | 001 | 0001 |  |  |
| 140 | A8F8 | 141 | A8F8 | 5B6 | 0000 | 000 | FFF8 | 0000 | 005 | 0101 |  |  |
| 141 | 0680 | 142 | 0680 | 141 | 0680 | 000 | 0141 | 0000 | 005 | 0101 |  |  |
| 142 | 1302 | 143 | 1302 | 142 | 1302 | 000 | 0142 | 0000 | 005 | 0101 |  |  |
| 143 | 7F00 | 144 | 7F00 | 143 | 0000 | 000 | 0000 | 0000 | 005 | 0101 |  |  |
| 144 | F008 | 14D | F008 | 144 | F008 | 000 | 0008 | 0000 | 005 | 0101 |  |  |
| 14D | 0100 | 14E | 0100 | 14D | 0100 | 000 | 014D | 0000 | 005 | 0101 |  |  |

# Проверка результата

|  |  |  |
| --- | --- | --- |
| Входные данные | Ожидаемый результат | Вывод на ВУ |
| 00D = 13 | 13\*(-5)+1 = -64 = 1100 0000 | 1100 0000 |
| FFE8 = -8 | -9 \* (-5) +1 = 46 = 1110 1001 |  |
|  |  |  |

# Ответы на вопросы

ORG 0x0

V0: WORD $DEFAULT, 0x180

V1: WORD $INT1, 0x180

V2: WORD $INT2, 0x180

V3: WORD $DEFAULT, 0x180

V4: WORD $DEFAULT, 0x180

V5: WORD $DEFAULT, 0x180

V6: WORD $DEFAULT, 0x180

V7: WORD $DEFAULT, 0x180

ORG 0x045

X: WORD ?

max: WORD 0x0016

min: WORD 0xFFE7

DEFAULT: IRET

START:

DI

CLA

OUT 0x1

OUT 0x7

OUT 0xB

OUT 0xD

OUT 0x11

OUT 0x15

OUT 0x19

OUT 0x1D

LD #0x9

OUT 3

LD #0xA

OUT 5

ST X

EI

main:

DI

LD X

INC

CALL check

ST X

EI

JUMP main

INT1:

NOP

PUSH

LD X

ASL

ASL

ADD X

NEG

INC

OUT 2

POP

IRET

INT2:

DI

IN 4

NOP

AND X

NOT

ST X

NOP

EI

IRET

check:

check\_min: CMP min

BPL check\_max

JUMP ld\_min

check\_max: CMP max

BMI return

ld\_min: LD min

return: RET

ДОП

; Инициализация векторов прерываний

ORG 0x0

V0: WORD $DEFAULT, 0x180 ; ВУ-0 (таймер)

V1: WORD $DEFAULT, 0x180 ; ВУ-1 (устройство вывода)

V2: WORD $DEFAULT, 0x180 ; ВУ-2 (устройство ввода)

V3: WORD $DEFAULT, 0x180 ; ВУ-3 (устройство ввода-вывода)

V4: WORD $DEFAULT, 0x180 ; ВУ-4

V5: WORD $DEFAULT, 0x180 ; ВУ-5 (текстовый принтер)

V6: WORD $IRQ\_HANDLER, 0x180 ; ВУ-6 (бегущая строка)

V7: WORD $DEFAULT, 0x180 ; ВУ-7 (7-сегментный индикатор)

V8: WORD $DEFAULT, 0x180 ; ВУ-8 (клавиатура)

V9: WORD $IRQ\_HANDLER, 0x180 ; ВУ-9 (цифровая клавиатура) - наш обработчик

; Область данных

ORG 0x40

DIGIT: WORD ? ; Для хранения введенной цифры

; Обработчик по умолчанию

DEFAULT: IRET

; Основная программа

ORG 0x50

START:

DI

CLA

OUT 0x1

OUT 0x7

OUT 0xB

OUT 0xE

OUT 0x16

OUT 0x1A

; Инициализация регистров управления ВУ

LD #0xE

OUT 0x12

LD #0xF

OUT 0x1E

EI ; Разрешаем прерывания

MAIN\_LOOP:

CLA ; Очищаем аккумулятор

BR MAIN\_LOOP ; Бесконечный цикл ожидания прерывания

; Обработчик прерывания для ВУ-9 (цифровая клавиатура)

IRQ\_HANDLER:

PUSH ; Сохраняем аккумулятор

PUSHF ; Сохраняем регистр состояния

IN 0x19 ; Читаем регистр данных ВУ-9 (DR#19)

AND #0x0F ; Оставляем только младшие 4 бита (0-9)

ST DIGIT ; Сохраняем цифру

hlt

; Выводим цифру в бегущую строку (ВУ-6)

LD DIGIT

OUT 0x14 ; DR ВУ-6 (адрес 0x14)

hlt

POPF ; Восстанавливаем регистр состояния

POP ; Восстанавливаем аккумулятор

IRET ; Возврат из прерывания

;Длинна 12, каждое число 20 бит. (на каждое по 2 ячейки) найти сумму индексов кратных 3.

ORG 0x0

START\_ADR: WORD 0x0100

STOP\_ADR: WORD ?

CUR\_ADR: WORD ?

LENGHT: WORD 0xC

RESULT\_1: WORD 0x0

RESULT\_2: WORD 0x0

X1: WORD 0x0

X2: WORD 0x0

FFF0: word 0xfff0

START:

CLA

ST RESULT\_1

ST RESULT\_2

LD START\_ADR

ADD LENGHT

ST STOP\_ADR

LD START\_ADR

ADD #0x4

ST CUR\_ADR

PROG:

CALL FUNC

LD CUR\_ADR

CMP STOP\_ADR

BPL STOP

JUMP PROG

STOP:

HLT

FUNC:

LD (CUR\_ADR)+

ADD RESULT\_1

ST RESULT\_1

LD (CUR\_ADR) ;загрузили 2

CALL CHECK

ADC RESULT\_2

ST RESULT\_2

LD CUR\_ADR

ADD #0x4

ST CUR\_ADR

CLA

RET

CHECK:

AND #0x8

CMP #0x1

BPL MINUS

LD (CUR\_ADR)+

RET

MINUS:

LD (CUR\_ADR)+

OR FFF0

RET

ORG 0x0100

WORD 0x0

WORD 0x0

WORD 0x0

WORD 0x0

WORD 0xffff

WORD 0xf

WORD 0x0

WORD 0x0

WORD 0x0

WORD 0x0

WORD 0x2

WORD 0x0

ORG 0x0

V0: WORD $DEFAULT, 0x180

V1: WORD $INT1, 0x180

V2: WORD $DEFAULT , 0x180

V3: WORD $DEFAULT, 0x180

V4: WORD $DEFAULT, 0x180

V5: WORD $DEFAULT, 0x180

V6: WORD $DEFAULT, 0x180

V7: WORD $DEFAULT, 0x180

ORG 0x045

X: WORD ?

max: WORD 0x0016

min: WORD 0xFFE7

DEFAULT: IRET

START:

DI

CLA

OUT 0x1

OUT 0x7

OUT 0xB

OUT 0xD

OUT 0x11

OUT 0x15

OUT 0x19

OUT 0x1D

LD #0x9

OUT 3

LD #0xA

OUT 5

ST X

EI

main:

DI

LD X

INC

CALL check

ST X

EI

JUMP main

INT1:

LD X

NOP

HLT

ASL

ASL

ADD X

NEG

INC

OUT 2

HLT

NOP

IRET

INT2:

IN 4

NOP

HLT

AND X

NOT

ST X

HLT

NOP

IRET

check:

check\_min: CMP min

BPL check\_max

JUMP ld\_min

check\_max: CMP max

BMI return

ld\_min: LD min

return: RET