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 №6

### по дисциплине

### «ОСНОВЫ ПРОФЕССИОНАЛЬНОЙ ДЕЯТЕЛЬНОСТИ»

### Вариант №3416

##### ***Выполнил:*** Студент группы P3134 Баянов Равиль Динарович

#### Преподаватель:

##### Бострикова Дарья

##### Константиновна

Санкт-Петербург

2023

**Оглавление**

[**Задание** 3](#_Toc133369992)

[**Текст исходной программы** 4](#_Toc133369993)

[**Описание программы** 8](#_Toc133369994)

[**Вывод** 10](#_Toc133369996)

# **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.

**Вариант:**

Основная программа должна увеличивать на 3 содержимое X (ячейки памяти с адресом 03516) в цикле.

Обработчик прерывания должен по нажатию кнопки готовности ВУ-3 осуществлять вывод результата вычисления функции F(X)=-7X-9 на данное ВУ, a по нажатию кнопки готовности ВУ-2 вычесть Х из утроенного содержимого РД данного ВУ, результат записать в Х

Если Х оказывается вне ОДЗ при выполнении любой операции по его изменению, то необходимо в Х записать минимальное по ОДЗ число.

# **Текст исходной программы**

|  |  |  |
| --- | --- | --- |
| **Адрес** | **Мнемоника** | **Комментарии** |
| - | ORG 0x0 | Инициализация векторов прерывания |
| 000, 001 | V0: WORD $DEFAULT, 0x180 | Вектор прерывания #0 |
| 002, 003 | V1: WORD $DEFAULT, 0x180 | Вектор прерывания #1 |
| 004, 005 | V2: WORD $INT2, 0x180 | Вектор прерывания #2 (Вектор прерывания для ВУ 2) |
| 006, 007 | V3: WORD $INT3, 0x180 | Вектор прерывания #3 (Вектор прерывания для ВУ 3) |
| 008, 009 | V4: WORD $DEFAULT, 0x180 | Вектор прерывания #4 |
| 00A,  00B | V5: WORD $DEFAULT, 0x180 | Вектор прерывания #5 |
| 00C,  00D | V6: WORD $DEFAULT, 0x180 | Вектор прерывания #6 |
| 00E,  00F | V7: WORD $DEFAULT, 0x180 | Вектор прерывания #7 |
| 010 | DEFAULT: IRET | Просто возврат (Обработка прерывания по умолчанию) |
| - | ORG 0x035 | Загрузка начальных векторов прерывания |
| 035 | X: WORD 0x0000 | Переменная X |
| 036 | MIN: WORD 0xFFEE | Минимальное допустимое значение X |
| 037 | MAX: WORD 0x0012 | Максимальное допустимое значение X |
| 038 | START: DI | Запрет на прерывания |
| 039 | CLA | Обнуление аккумулятора AC |
| 03A | OUT 0x1 | Запрет прерываний для неиспользуемых ВУ |
| 03B | OUT 0x3 |
| 03C | OUT 0xB |
| 03D | OUT 0xD |
| 03E | OUT 0x11 |
| 03F | OUT 0x15 |
| 040 | OUT 0x19 |
| 041 | OUT 0x1D |
| 042 | LD #0xA | Загрузка в аккумулятор MR (1000|0010=1010) |
| 043 | OUT 5 | Разрешение прерываний для ВУ 2 |
| 044 | LD #0xB | Загрузка в аккумулятор MR (1000|0011=1011) |
| 045 | OUT 7 | Разрешение прерываний для ВУ 3 |
| 046 | EI | Разрешение прерываний |
| 047 | PROG: DI | Запрет прерываний |
| 048 | LD X | Загрузка переменной X в аккумулятор AC |
| 049 | ADD #3 | Увеличение переменной X на 3 |
| 04A | CALL CHECK | Переход к подпрограмме проверки ОДЗ переменной X |
| 04B | ST X | Сохранение новой переменной X в ячейку |
| 04C | EI | Разрешение прерываний |
| 04D | JUMP PROG | Безусловный переход (START) |
| 04E | VU2: WORD 0x0000 | Ячейка для записи содержимого регистра данных ВУ 2 |
| 04F | INT2: DI | Обработка прерывания на ВУ 3 (Запрет прерываний) |
| 050 | NOP | Отладочная остановка (NOP/HLT) |
| 051 | IN 4 | Ввод с ВУ 2 в аккумулятор AC |
| 052 | ST VU2 | Сохранение содержимого регистра данных ВУ 2 в ячейке VU2 |
| 053 | ASL | Увеличение в 2 раза аккумулятора AC |
| 054 | ADD VU2 | Добавление ячейки значения ячейки VU2 (Прибавление значения из регистра данных ВУ 2) |
| 055 | SUB X | Вычитание из аккумулятора AC значения переменной X |
| 056 | ST X | Сохранение нового значения X в ячейку памяти |
| 057 | NOP | Отладочная остановка (NOP/HLT) |
| 058 | EI | Разрешение прерываний |
| 059 | IRET | Возврат из обработки прерываний |
| 05A | INT3: DI | Запрет прерываний |
| 05B | NOP | Отладочная остановка (NOP/HLT) |
| 05C | PUSH | Сохраняем значение аккумулятора AC в стек ST |
| 05D | LD X | Загрузка переменной X в аккумулятор AC |
| 05E | ASL | Увеличение в 2 раза аккумулятора AC |
| 05F | ADD X | Сложение значение аккумулятора AC со значением переменной X |
| 060 | ASL | Увеличение в 2 раза аккумулятора AC |
| 061 | ADD X | Сложение значение аккумулятора AC со значением переменной X |
| 062 | ADD #9 | Прибавление к аккумулятору AC значение 9 |
| 063 | NEG | Отрицание значения аккумулятора AC |
| 064 | OUT 6 | Вывод на ВУ 3 |
| 065 | LD X | Загрузка значения переменной X в аккумулятор AC |
| 066 | NOP | Отладочная остановка (NOP/HLT) |
| 067 | POP | Вынимаем значение стека в аккумулятор AC |
| 068 | EI | Разрешение прерываний |
| 069 | IRET | Возврат из обработки прерываний |
| 06A | CHECK:  CMP MIN | Сравнение переменной X с минимальным допустимым значением переменной X |
| 06B | BPL CMAX | Если переменная X> MIN переход на проверку максимального допустимого значения X |
| 06C | JUMP LDMIN | Безусловный переход на LDMIN |
| 06D | CMAX: CMP MAX | Сравнение переменной X с максимальным допустимым значением переменной X |
| 06E | BMI RETURN | Если переменная X <MAX переход на RETURN |
| 06F | LDMIN: LD MIN | Загрузка минимального допустимого значения X в аккумулятор AC |
| 070 | RETURN: RET | Возврат из проверки допустимых значений переменной X |

**Код на ассемблере**

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

DEFAULT: IRET

ORG 0x035

X: WORD 0x0000

MIN: WORD 0xFFEE

MAX: WORD 0x0012

START: DI

CLA

OUT 0x1

OUT 0x3

OUT 0xB

OUT 0xD

OUT 0x11

OUT 0x15

OUT 0x19

OUT 0x1D

LD #0xA

OUT 5

LD #0xB

OUT 7

EI

PROG: DI

LD X

ADD #3

CALL CHECK

ST X

EI

JUMP PROG

VU2: WORD 0x0000

INT3: DI

LD X

ASL

ADD X

ASL

ADD X

ADD #9

NEG

OUT 6

LD X

NOP

EI

IRET

INT2: DI

NOP

IN 4

ST VU2

ASL

ADD VU2

SUB X

ST X

NOP

EI

IRET

CHECK:

CMP MIN

BPL CMAX

JUMP LDMIN

CMAX: CMP MAX

BMI RETURN

LDMIN: LD MIN

RETURN: RET

# **Описание программы**

Программа циклически увеличивает значение переменной X на 3 и обрабатывает прерывания.

***Расположение в памяти БЭВМ программы, исходных данных и результатов:***

Векторы прерываний: 0x000 – 0x00F

Переменные и константы: 0x035 – 0x037, 04E

Основная программа: 0x038 – 0x070

#### Область представления:

X, MIN, MAX – знаковое 16-ричное число

VU2 – беззнаковое 8-ричное число

#### Область допустимых значений:

-27 ≤ F(X) ≤ 27 - 1

-27 ≤ -7X – 1 ≤ 27 - 1

-27 + 1≤ -7X ≤ 27

(-27 + 1) / 7 ≤ -X ≤ 27 / 7

-18≤ -X ≤ 18

18≥ X ≥ -18

***Методика проверки:***

Проверка основной программы:

1. Загрузить текст программы в БЭВМ.

2. Записать в переменную X минимальное по ОДЗ значение (-18)

3. Запустить программу в режиме останова.

4. Пройти нужное количество шагов программы, убедиться, что при увеличении

X на 3, до того момента, когда он равен 18, происходит сброс значения в

минимальное по ОДЗ

Проверка обработки прерываний:

1. Загрузить текст программы в БЭВМ.

2. Заменить NOP на HLT.

3. Запустить программу в режиме РАБОТА.

4. Установить «Готовность ВУ-3».

5. Дождаться останова.

6. Записать текущее значение X из памяти БЭВМ:

1. Запомнить текущее состояние счетчика команд.

2. Ввести в клавишный регистр значение 0x035

3. Нажать «Ввод адреса».

4. Нажать «Чтение».

5. Записать значение регистра данных.

6. Вернуть счетчик команд в исходное состояние.

7. Записать результат обработки прерывания – содержимое DR контроллера ВУ-3

8. Рассчитать ожидаемое значение обработки прерывания

9. Нажать «Продолжение».

10. Ввести в ВУ-2 произвольное число, записать его

11. Установить «Готовность ВУ-2».

12. Дождаться останова.

13. Записать текущее значение X из памяти БЭВМ (аналогично п.6).

14. Нажать «Продолжение».

15. Записать текущее значение X из памяти БЭВМ (аналогично п.6).

16. Рассчитать ожидаемое значение переменной X после обработки прерывания

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Прерывание ВУ 2(При вводе 0 в ВУ 2) | | | Прерывание ВУ 3 | | |
| AC | Ожидание | AC | AC | Ожидание | AC |
| 010 | 010 | 010 | 010 | -910 | -910 |
| -1010 | 1010 | 1010 | 110 | -1610 | -1610 |
| -2010 | 2010 | 2010 | 210 | -2310 | -2310 |

Основная программа

|  |  |  |
| --- | --- | --- |
| AC | Ожидание | AC |
| 010 | -1810 | -1810 |
| -1810 | -1810 | -1810 |
| 1810 | -1810 | -1810 |

# **Вывод**

Изучил организацию процесса прерывания программы и исследовал порядок функционирования ЭВМ при обмене данными в режиме прерывания программы.