

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

Факультет программной инженерии и компьютерной техники

Лабораторная работа по основам профессиональной деятельности №6  
Вариант №6607

Выполнил:

Студент группы Р3106

Мельник Фёдор Александрович

Проверил:

Ткешелашвили Нино Мерабиевна,

Преподаватель-практик ФПИиКТ

Санкт-Петербург, 2025

# Оглавление

Текст задания .....	3
Описание программы .....	4
Назначение программы .....	4
Текст программы.....	4
Область допустимых значений .....	6
Расположение данных в памяти.....	6
Область представления .....	6
Методика проверки программы .....	6
Вывод .....	7

## Текст задания

По выданному преподавателем варианту разработать и исследовать работу комплекса программ обмена данными в режиме прерывания программы. Основная программа должна изменять содержимое заданной ячейки памяти ( $X$ ), которое должно быть представлено как знаковое число. Область допустимых значений изменения  $X$  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  $X$  в соответствии с вариантом задания, а также игнорировать все необрабатываемые прерывания.

Введите номер варианта

1. Основная программа должна уменьшать на 2 содержимое  $X$  (ячейки памяти с адресом  $00B_{16}$ ) в цикле.
2. Обработчик прерывания должен по нажатию кнопки готовности ВУ-1 осуществлять вывод результата вычисления функции  $F(X) = -6X - 1$  на данное ВУ, а по нажатию кнопки готовности ВУ-3 прибавить утроенное содержимое РД данного ВУ к  $X$ , результат записать в  $X$ .
3. Если  $X$  оказывается вне ОДЗ при выполнении любой операции по его изменению, то необходимо в  $X$  записать максимальное по ОДЗ число.

# Описание программы

## Назначение программы

1. Основная программа должна уменьшать на 2 содержимое X (ячейки памяти с адресом 00B<sub>16</sub>) в цикле.
2. Обработчик прерывания должен по нажатию кнопки готовности ВУ-1 осуществлять вывод результата вычисления функции  $F(X) = -6X - 1$  на данное ВУ, а по нажатию кнопки готовности ВУ-3 прибавить утроенное содержимое РД данного ВУ к X, результат записать в X
3. Если X оказывается вне ОДЗ при выполнении любой операции по его изменению, то необходимо в X записать максимальное по ОДЗ число.

## Текст программы

```
ORG 0x0
V0: WORD $default, 0X180
V1: WORD $int1, 0X180
V2: WORD $default, 0x180
V3: WORD $int3, 0X180
V4: WORD $default, 0X180
V5: WORD $default, 0X180
V6: WORD $default, 0X180
V7: WORD $default, 0X180

ORG 0x020
X: WORD ?

temp: WORD 0x0;
max: WORD 0x0015 ; 21, максимальное значение X
min: WORD 0xFFEA ; -22, минимальное значение X
default: IRET ; Обработка прерывания по умолчанию

START: DI
      CLA
      OUT 0x1 ; Запрет прерываний для неиспользуемых ВУ
      OUT 0x5
      OUT 0xB
      OUT 0xD
      OUT 0x11
      OUT 0x15
      OUT 0x19
      OUT 0x1D
LD #0x9
      OUT 0x3 ; Разрешение прерываний для 1 ВУ
```

```

        LD #0xB
        OUT 0x7          ; Разрешение прерываний для 3 ВУ
        EI
main:    DI              ; Запрет прерываний чтобы обеспечить атом. операции
        LD X
        DEC
DEC      DEC
        CALL check
        ST X
NOP
        EI              ; Разрешение прерываний для обработки
        JUMP main
int1:    DI              ; Обработка прерывания на ВУ-1
        LD X
        ASL
        ASL
        ADD X
        ADD X
        NEG
        SUB #1
        OUT 0x2
        NOP
        EI
        IRET
int3:    DI              ; Обработка прерывания на ВУ-3
        IN 0x6
        ST temp
        ASL
        ADD temp
        ADD X
        CALL check
        ST X
        NOP
        EI
        IRET

check:   ; Проверка ОДЗ
check_min:  CMP min
        BPL check_max   ; Если число больше MIN, проверяем верхнюю границу
        JUMP ld_max
check_max:  CMP max
        BMI return      ; Если число меньше MAX, возвращаем это число неизменным
ld_max:    LD max
return: RET

```

## Область допустимых значений

$$-128 \leq -6(X) - 1 \leq 127$$

$$-127 \leq -6(X) \leq 128$$

$$21.167 \geq X \geq -21.333$$

$$-22 \leq X \leq 21$$

$$21 = 0x0015$$

$$-22 = 0xFFEA$$

## Расположение данных в памяти

Вектора прерываний: 0x000 – 0x00F

Переменные: 0x020 – 0x025

Программа: 0x026-0x05B

## Область представления

X, min, max – знаковое, 16ти разрядное

DR KBY – знаковое, 8ми разрядное

## Методика проверки программы

Проверка обработки прерываний на ВУ-1:

1. Загрузить текст программы в БЭВМ во вкладку Ассемблер.
2. Заменить NOP на HLT в блоке int1.
3. Нажать кнопку «Компилировать»
4. Запустить программу в режиме РАБОТА.
5. Открыть ВУ-1 в верхней части окна. Нажать кнопку «Готов»
6. Дождаться останова.
7. Записать текущее значение X из памяти БЭВМ:
  - 1) Ввести в клавишный регистр значение 0x20
  - 2) Нажать «Ввод адреса»
  - 3) Нажать «Чтение»
  - 4) Записать значение регистра данных
8. Записать результат обработки прерывания – содержимое DR контроллера ВУ-1
9. Рассчитать самостоятельно ожидаемое значение обработки прерывания по формуле  $(F(x) = -6x - 1)$
10. Сравнить значения, полученные в пунктах 8, 9. Если они равные – программа работает верно

Проверка обработки прерываний на ВУ-3:

1. Загрузить текст программы в БЭВМ.
2. Заменить NOP на HLT в блоке int3.
3. Заменить значение переменной X, записать его.

4. Нажать кнопку «Компилировать»
5. Ввести в ВУ-3 произвольное число, записать его.
6. Установить «Готовность ВУ-3».
7. Запустить программу в режиме РАБОТА.
8. Дождаться останова.
9. Записать значение X из памяти БЭВМ:
  - 1) Ввести в клавишный регистр значение 0x20
  - 2) Нажать «Ввод адреса».
  - 3) Нажать «Чтение».
  - 4) Записать значение регистра данных.
9. Рассчитать самостоятельно ожидаемое значение переменной X после обработки прерывания (выполнить  $3DR + X$ )
10. Сравнить значения, полученные в пунктах 8 и 9. Если они равные – программа работает верно

Проверка основной программы:

1. Загрузить текст программы в БЭВМ.
  2. Заменить NOP на HLT в блоке main.
  3. Заменить значение переменной X на минимальное по ОДЗ значение (0xFFEA).
  4. Запустить программу в режиме РАБОТА (F9, F7) и убедиться, что при декрементировании X, если он выходит за рамки ОДЗ, происходит сброс значения в максимальное по ОДЗ.
  5. Повторить пункты 3-4 с другим числом, выходящим за пределы ОДЗ.
- \* Если из блока main вообще убрать NOP, то программа будет циклично декрементировать переменную X. Можно убедиться в этом, убрав NOP.

Прерывание ВУ-1			Прерывание ВУ-3			
X	Ожидание -6X-1	DR	X	DR КВУ-3	Ожидание 3DR + X	Результат
-6 <sub>10</sub> (0xFFFFA)	0x0023	0x0023	0x7	0x1	0x000A	0x000A
-4 <sub>10</sub> (0xFFFFC)	0x0017	0x0017	0xFFFFC	0x2	0x0002	0x0002

Основная программа		
AC	Ожидание	AC
0xFFEA	0x15	0x15
0xFFEE	0x15	0x15

## Вывод

Во время выполнения лабораторной работы я научился работать с вводом и выводом с использованием прерываний. Также я изучил циклы выполнения новых команд и закрепил навык написания программ на ассемблере.