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 № 1271

*Выполнил:*

Студент группы P3112 Степанов Артур Петрович

*Преподаватель:*

Ткешелашвили Нино Мерабиевна

Санкт-Петербург, 2022

# 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.Программа осуществляет асинхронный ввод данных с ВУ-2.

1. Основная программа должна уменьшать на 2 содержимое X (ячейки памяти с адресом 01A16) в цикле.
2. Обработчик прерывания должен по нажатию кнопки готовности ВУ-1 осуществлять вывод результата вычисления функции F(X)=-7X-5 на данное ВУ, a по нажатию кнопки готовности ВУ-3 выполнить операцию побитового 'Исключающее 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.

Код программы:  
ORG 0x01A

X: WORD ?

SAVED\_A: WORD ?

SAVED\_AB: WORD ?

LEFT: WORD 0xFFED ; левая граница (-128-5)/7 = -19

RIGHT: WORD 0x0011 ; правая граница (127 - 5)/7 = 17

ORG 0x0 ; Инициализация векторов прерывания

V0: WORD $DEFAULT,0x180 ; Вектор прерывания #0

V1: WORD $INT1,0x180 ; Вектор прерывания #1

V2: WORD $DEFAULT,0x180 ; Вектор прерывания #2

V3: WORD $INT2,0x180 ; 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 ; просто возврат

ORG 0x0F0 ; загрузка начальных векторов прерывания

START: DI ; запрет прерываний

CLA

OUT 0x1 ; MR КВУ-0 на вектор 0

OUT 0x5 ; MR КВУ-2 на вектор 0

OUT 0xB ; MR КВУ-4 на вектор 0

OUT 0xD ; MR КВУ-5 на вектор 0

OUT 0x11 ; MR КВУ-6 на вектор 0

OUT 0x15 ; MR КВУ-7 на вектор 0

OUT 0x19 ; MR КВУ-8 на вектор 0

OUT 0x1D ; MR КВУ-9 на вектор 0

LD #0x9 ; разрешить прерывания и вектор №1

OUT 3 ; (1000|0001=1001) в MR КВУ-1

LD #0xB ; разрешить прерывания и вектор №3

OUT 7 ; (1000|0011=1011) в MR КВУ-3

JUMP $PROG

PROG: EI ; разрешили прерывания

CLA

SPINLOOP: EI ; разрешили прерывания

LD $X

DI ; запретили прерывания

SUB #0x2 ; уменьшили на 2

CMP $LEFT ; сравнили с

BLT MIN ; левой границей

CMP $RIGHT ; сравнили с

BGE MIN ; правой границей

ST $X ; сохранили Х

JUMP SPINLOOP ; продолжили бесконечный цикл

MIN: LD $LEFT ; загрузили левую границу ОДЗ

ST $X ; сохранили Х

JUMP SPINLOOP ; продолжили бесконечный цикл

INT1: DI ; обработка прерывания ВУ-1

LD $X ; загрузили Х

ASL ; умножили на 2

ASL ; умножили на 4

ASL ; умножили на 8

SUB $X ; умножили на 7

NEG ; умножили на -7

SUB #0x5 ; вычли 5

OUT 2 ; вывели на ВУ-1

HLT ; проверка работоспособности ВУ-1 NOP

IRET ; возврат из обработки прерывания

INT2: DI ; обработка прерывания ВУ-2

CLA

IN 6 ; загрузили РДВУ-3

ST $SAVED\_A ; сохранили РДВУ-2

AND $X ;умножили на X

ST $SAVED\_AB ;Сохранили

LD $SAVED\_A ;

OR $X ;

NOT ;Получили NOT(AB)

OR $SAVED\_AB ;получили исключающее или

CMP $LEFT ; сравнили с

BLT MINN ; левой границей

CMP $RIGHT ; сравнили с

BGE MINN ; правой границей

ST $X ; сохранили Х

HLT ; проверка работоспособности ВУ-3 NOP

IRET ; возврат из обработки прерывания

MINN: LD $RIGHT ; загрузили левую границу ОДЗ

ST $X ; сохранили Х

HLT ; проверка работоспособности ВУ-3 NOP

IRET ; возврат из обработки прерывания

# Описание программы

1. Основная программа уменьшает на 2 содержимое Х в цикле.
2. Обработчик прерывания по нажатию кнопки готовности ВУ1 выводит результат вычисления функции f(x) = -7x-5 на данное ВУ, а по нажатию кнопки готовности ВУ3 осуществляет прибавление утроенного содержимого регистра данных ВУ к X и записывает результат в Х.
3. Если Х оказывается вне ОДЗ при выполнении любой операции по его изменению, то в Х записывается минимальное по ОДЗ число.
4. ОДЗ: x Є [-19; 17]

    f(x) Є [-128; 127]

1. Расположение в памяти:

Адрес первой команды: 0F0

Адрес результата: 01A

# Методика проверки

1. Загрузить текст программы в БЭВМ.
2. Заменить NOP на HLT.
3. Запустить программу в режиме РАБОТА.
4. Установить «Готовность ВУ1».
5. Дождаться остановки.
6. Записать содержимое IP.
7. Записать текущее значение X из памяти БЭВМ:
   1. Ввести в клавишный регистр значение 001A.
   2. Нажать «ВВОД АДРЕСА».
   3. Нажать «ЧТЕНИЕ».
   4. Записать содержимое DR.
8. Записать результат обработки прерывания - содержимое DR контроллера ВУ-1.
9. Восстановить содержимое счётчика команд:
   1. Ввести полученное на пункте 6 значение в клавишный регистр.
   2. Нажать «ВВОД АДРЕСА».
10. Нажать «ПРОДОЛЖЕНИЕ».
11. Ввести в регистр данных контроллера ВУ-3 произвольное число, записать как содержимое DR контроллера ВУ-3.
12. Установить «Готовность ВУ-3»
13. Дождаться остановки.
14. Записать содержимое IP.
15. Повторить пункт 7.
16. Записать старое значение X из памяти БЭВМ:
    1. Ввести в клавишный регистр значение 001B.
    2. Нажать «ВВОД АДРЕСА».
    3. Нажать «ЧТЕНИЕ».
    4. Записать содержимое DR.
17. Удостовериться что всё идёт по плану: ожидаемые значения совпадают с фактическими
18. Проверим основную программу (присваивается ли минимальное значение по ОДЗ при выходи за границы ОДЗ)
19. Для этого не будем вызывать прерывания от ВУ и, наблюдая за изменением ячейки аккумулятора, убедимся в этом

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| IP | X | -7Х-5 (ожидаемое) | -7Х-5 (фактическое) | РДВУ-3 | IP | Х | РДВУ-3|Х (ожидаемое) | РДВУ-3|Х (фактическое) |
| 118 | -6 | 25 | 25 | 88 | 12D | 0011 | 0011 | 0011 |

# Где A|B = A\*B+!A\*!B

# Вывод:

В ходе выполнения данной лабораторной работы я научился работать с ВУ-1, ВУ-3 освоил команды ввода-вывода, а также познакомился с синтаксисом Ассемблера.