Федеральное государственное автономное образовательное учреждение

высшего образования «Национальный исследовательский университет ИТМО»

*Факультет программной инженерии и компьютерной техники*

**Лабораторная работа №6**

**По дисциплине**

**“Основы профессиональной деятельности”**

Вариант: 65342

Выполнил:

Темешев Тимур Аленович

Группа: Р3112

Преподаватель:

Осипов Святослав Владимирович

Санкт-Петербург, 2023г

Оглавление

[Задание 2](#_Toc97076875)

[Ход работы 2](#_Toc97076876)

[Текст исходной программы 2](#_Toc97076877)

[Описание программы 4](#_Toc97076878)

[Трассировка с данными числами 5](#_Toc97076879)

[Вывод 5](#_Toc97076880)

[Список литературы 5](#_Toc97076881)

## 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 01816) в цикле.
2. Обработчик прерывания должен по нажатию кнопки готовности ВУ-3 осуществлять вывод результата вычисления функции F(X)=-6X-1 на данное ВУ, a по нажатию кнопки готовности ВУ-2 выполнить операцию побитового маскирования, оставив 3-х младших разряда 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 работы

### Текст исходной программы

            ORG     0x000 ;инициализация векторов прерываний

V0: WORD    $DEFAULT, 0x180

V1: WORD    $INT1, 0x180 ;Вектор прерываний для ВУ 1

V2: WORD    $INT2, 0x180    ;Вектор прерываний для ВУ 2

V3: WORD    $DEFAULT, 0x180

V4: WORD    $DEFAULT, 0x180

V5: WORD    $DEFAULT, 0x180

V6: WORD    $DEFAULT, 0x180

V7: WORD    $DEFAULT, 0x180

ORG 0x02D

X:  WORD    0x0000  ;Переменная X

MIN: WORD   0xFFEB  ;Нижняя граница значений X

MAX: WORD   0x0015  ;Верхняя граница значений X

DEFAULT:    IRET        ;Обработка прерывания по умолчанию

X\_ADDRESS: WORD $X

TEMP: WORD ?

START:  DI

        CLA

        LD #0x9

        OUT 3 ;разрешение прерываний для 1 ВУ

        LD #0xA

        OUT 5 ;разрешение прерываний для 2 ВУ

        EI

MAIN:

        EI

        LD X

        ST TEMP

        INC

        CALL CHECK

        PUSH

        LD TEMP

        PUSH

        LD X\_ADDRESS

        PUSH

        CALL $CAS

        JUMP MAIN

INT1: ;обработка прерывания на ВУ-1

    PUSH

    LD X

    ASL

    ADD X

    ASL

    INC

    INC

    NEG

    OUT 2

    LD X

    NOP

    POP

    IRET

INT2: ;обработка прерывания на ВУ-2

    PUSH

    NOP

    IN 4

    AND X

    ST X

    NOP

    POP

    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 возврат

LD\_MIN:

    LD MIN ;Загрузка минимального значения в X

RETURN: RET ;Метка возврата из проверки на ОДЗ

DEREF: WORD ?

CAS:

    PUSHF

    DI

    LD &2

    ST DEREF

    LD (DEREF)

    CMP &3

    BNE FAIL

SUCCES: LD &4

    ST (DEREF)

    LD #0x1

    JUMP EXIT

FAIL:CLA

EXIT: POPF

SWAP

ST &3

SWAP

SWAP

POP

SWAP

POP

SWAP

POP

RET

; RET 1- новое  0 - старое

### Описание программы

**Назначение программы:**

Программа циклически увеличивает значение ячейки памяти на 1 и обрабатывает прерывания.

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

Вектор прерываний: 0x000 – 0x00F

Переменные: 0x2D – 0x062

Программа: 0x030 – 0x04D

**Область представления:**

X, MIN, MAX – знаковое 16-ричное целое число

**Область допустимых значений**

# *Методика проверки программы*

Проверка обработки прерываний:

1. Загрузить текст программы в БЭВМ.
2. Заменить все NOP на HLT.
3. Запустить программу в автоматическом режиме с адреса 0x033
4. Установить «Готовность ВУ-2».
5. Дождаться останова.
6. Записать текущее значение X из памяти БЭВМ:
   * + 1. Запомнить текущее состояние счетчика команд.
       2. Ввести в клавишный регистр значение 0x02D
       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 КВУ-2
8. Рассчитать ожидаемое значение обработки прерывания
9. Нажать «Продолжение».
10. Установить «Готовность ВУ-1».
11. Дождаться останова.
12. Записать текущее значение X из памяти БЭВМ (аналогично п.6).AC(0..7)
13. Нажать «Продолжение».
14. Записать текущее значение X из памяти БЭВМ (аналогично п.6).AC(DR + X)
15. Рассчитать ожидаемое значение переменной X после обработки прерывания

(если значение X выходит за пределы ОДЗ, тогда в X будет записано минимальное по

ОДЗ значение, т.е. FFEB)

Проверка основной программы:

1. Загрузить текст программы в БЭВМ.
2. Записать в переменную X минимальное по ОДЗ значение (-21 = FFEB)
3. Запустить программу в режиме останова.
4. Пройти нужное количество шагов программы, убедиться, что при увеличении

X на 1, до того момента, когда он равен 22, происходит сброс значения в минимальное по ОДЗ

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Прерывание ВУ-1 | | | Прерывание ВУ-2 | | | |
| AC(0..7) | -6x-2 | DR | AC(0..7) | DR КВУ-2 | AC(DR & X) | Результат AC(0..7) |
| 10 | 9E | 9E | 11 | EF | 1 | 1 |
| 11 | 98 | 98 | 11 | FF | 11 | 11 |

### Трассировка с данными числами

----------------------------------------------------------------

## Вывод

В ходе выполнения работы я ознакомился с устройством обмена по прерываниям, изучил процесс прерывания. Также закрепил знания в написании программ на ассемблере БЭВМ.

## Список литературы

1. Тейнсли Д. Linux и UNIX программирование в shell. Руководство разработчика. BHV, «Ирина», Киев, 2001. – Режим доступа: <https://rtfm.co.ua/uploads/books/Linux_i_UNIX_proghrammirovaniie_v_shell_-_Devid_Tieinsli.pdf>
2. Операционная система Linux: Курс лекций. Учебное пособие / Г. В .Курячий, К. А. Маслинский — М. : ALT Linux; Издательство ДМК Пресс, 2010. — 348 с. : ил. ; 2-е изд., исправленное.— (Библиотека ALT Linux). Режим доступа:

<https://docs.altlinux.org/books/altlibrary-linuxintro2.pdf>