ФГБОУ ВПО «СПбНИУ ИТМО»

*Факультет программной инженерии и компьютерной техники*

*Дисциплина “Основы профессиональной деятельности”*

**Лабораторная работа №6**

**3420 вариант**

Выполнил:

Мантуш Даниил Валерьевич,

группа Р3119

Санкт-Петербург

2025

# Оглавление

[Оглавление 2](#_Toc192349688)

[Задание 2](#_Toc192349689)

[Основные этапы вычисления 2](#_Toc192349690)

[1.1 Ассемблер 2](#_Toc192349691)

[1.2 Описание программы 3](#_Toc192349692)

[1.3 Область допустимых значений 3](#_Toc192349693)

[1.4 Расположение данных в памяти 4](#_Toc192349694)

[2.0 Таблица трассировки 4](#_Toc192349695)

[Вывод 6](#_Toc192349696)

# Задание

1. Основная программа должна увеличивать на 2 содержимое X (ячейки памяти с адресом 051) в цикле.
2. Обработчик прерывания должен по нажатию кнопки готовности ВУ-1 осуществлять вывод результата вычисления функции F(X)=4X на данное ВУ, a по нажатию кнопки готовности ВУ-3 выполнить операцию побитового '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 минимальное по ОДЗ число.

# Основные этапы вычисления

## Ассемблер

## ORG 0x0051

## X: WORD ?

## LEFT: WORD 0xFFE0 ; левая граница ОДЗ = -32

## RIGHT: WORD 0x001F ; правая граница ОДЗ = 31

## 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 $INT3,0x180 ; 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:

## IN 4

## IRET ; просто возврат

## ORG 0x0F0 ; загрузка начальных векторов прерывания

## START: DI ; запрет прерываний

## CLA

## LD #8

## 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 №2

## OUT 7 ; (1000|0011=1011) в MR КВУ-3

## JUMP $PROG

## PROG: EI ; разрешили прерывания

## CLA

## SPINLOOP: EI ; разрешили прерывания

## LD $X

## HLT

## DI ; запретили прерывания

## ADD #0x2 ; увеличили 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:

## HLT

## ASL ; умножили на 2

## ASL ; умножили на 4

## OUT 2 ; вывели на ВУ-1

## HLT ; проверка работоспособности ВУ-1

## IRET ; возврат из обработки прерывания

## INT3: HLT

## IN 6 ; загрузили РДВУ-3

## OR $X

## NOT

## ST $X

## POP

## HLT

## IRET ; возврат из обработки прерывания

## MINN: LD $LEFT ; загрузили левую границу ОДЗ

## ST $X ; сохранили Х

## HLT ; проверка работоспособности ВУ-3

## IRET ; возврат из обработки прерывания

## 1.2 Описание программы

## Основной цикл:

## Начальное X = 0.

## Увеличение на 2: 0, 2, 4, ..., 30, 32.

## При X = 32 → выход за ОДЗ, X сбрасывается в -32.

## Далее: -32, -30, -28, ..., 30, и цикл повторяется.

## ВУ-1: F(X) = 4X:

## X = 0: 4\*0 = 0.

## X = 31: 4\*31 = 124 (в пределах 8 бит).

## X = -32: 4\*(-32) = -128 (в пределах 8 бит).

## X не может превысить 31 или упасть ниже -32 из-за ОДЗ.

## ВУ-3: NOT (X XOR РД ВУ-3):

## X = 5 (00000101), РД ВУ-3 = 3 (00000011):

## X XOR РД = 00000110 (6).

## NOT 6 = 11111001 (-7).

## X = -7 (в пределах ОДЗ).

## X = 31 (00011111), РД ВУ-3 = -32 (11100000):

## X XOR РД = 11111111 (-1).

## NOT (-1) = 00000000 (0).

## X = 0 (в пределах ОДЗ).

## X = 31, РД ВУ-3 = 32 (00100000):

## X XOR РД = 00111111 (63).

## NOT 63 = 11000000 (-64).

## X = -32 (сброс в минимальное значение ОДЗ).

## 1.3 Область допустимых значений

* X ограничен диапазоном -32..31:
  + Минимум: -32 (0xFFE0).
  + Максимум: 31 (0x001F).
* Это гарантирует, что F(X) = 4X всегда укладывается в 8-битный диапазон -128..127.

Основная программа:

* Увеличение X на 2 теперь проверяется на диапазон -32..31.
* Если X выходит за пределы (например, становится 32 или -33), присваивается -32.

Обработчик ВУ-3:

* Операция NOT (X XOR РД) также проверяет результат на ОДЗ -32..31.

ВУ-1:

* Вывод F(X) = 4X остается без изменений, так как ограничение ОДЗ применяется только к X, а не к выводу (по условию).

## 2.0 Метод обработки прерываний

Проверка обработки прерываний:

1. Загрузить текст программы в БЭВМ.

2. Запустить программу в режиме РАБОТА.

3. Установить «Готовность ВУ-1».

4. Дождаться останова.

5. Записать текущее значение X из памяти БЭВМ:

1. Запомнить текущее состояние счетчика команд.

2. Ввести в клавишный регистр значение 0x010

3. Нажать «Ввод адреса».

4. Нажать «Чтение».

5. Записать значение регистра данных.

6. Вернуть счетчик команд в исходное состояние.

7. Нажать «Продолжение» 2 раза.

8. Записать результат обработки прерывания – полученное значение F(x) из DR контроллера ВУ-1.

9. Нажать «Продолжение».

10. Ввести в ВУ-3 произвольное число, записать его.

11. Установить «Готовность ВУ-3».

12. Дождаться останова.

13. Записать текущее значение X из памяти БЭВМ:

1. Запомнить текущее состояние счетчика команд.

2. Ввести в клавишный регистр значение 0x010

3. Нажать «Ввод адреса».

4. Нажать «Чтение».

5. Записать значение регистра данных.

6. Вернуть счетчик команд в исходное состояние.

14. Нажать «Продолжение» 2 раза.

15. Записать результат обработки прерывания – XOR из DR контроллера ВУ-3.

16. Рассчитать ожидаемое значение переменной X после обработки прерывания и сравнить.

Проверка основной программы:

1. Загрузить текст программы в БЭВМ.

2. Записать в переменную X минимальное по ОДЗ значение (-32)

3. Запустить программу в режиме останова.

4. Пройти нужное количество шагов программы, убедиться, что при увеличении X на 2, до момента, когда он равен 31, происходит сброс значения в минимальное по ОДЗ.

# Вывод

Во время выполнения данной лабораторной работы я изучил обмен данными между ВУ-1 и ВУ-3 в режиме прерываний, циклы прерывания и исполнения новых команд.