Министерство науки и высшего образования Российской Федерации

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 №2315

Выполнил:

Студент группы P3113

Султанов Артур Радикович

Проверил:

Блохина Елена Николаевна

г. Санкт-Петербург

2023г.

# Оглавление

[**Оглавление 2**](#_7c4xf1yrz0jn)

[**Задание 3**](#_2k7ftafq69on)

[Часть 1. Текст исходной программы 4](#_7jrtmguqfjq3)

[Часть 2. Описание программы 6](#_wrakoyp9d46u)

[Назначение программы 6](#_8u9qzhr2q8h)

[ОПИ, ОДЗ 6](#_xfs5xj3jd99e)

[Расположение данных 6](#_g0h7qzgsb1fn)

[Адреса первой и последней выполняемой команды 7](#_s0ag5ieqiln)

[Часть 3. Методика проверки 7](#_9m2lnrvj8c7k)

[**Заключение 10**](#_smihc5yw6057)

# 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 декрементировать содержимое X (ячейки памяти с адресом 0x036) в цикле.

2. Обработчик прерывания должен по нажатию кнопки готовности ВУ-1 осуществлять вывод результата вычисления функции F(X)=2X+2 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 максимальное по ОДЗ число.

## **Часть 1. Текст исходной программы**

| Адрес | Метка | Код команды | Мнемоника | Комментарии |
| --- | --- | --- | --- | --- |
| 050 | START | 1000 | DI | Запрет прерываний |
| 051 |  | 0200 | CLA | Очистка аккумулятора |
| 052 |  | AF09 | LD #0x9 | Вектор 1 и разрешение прерываний на MR ВУ-1 |
| 053 |  | 1303 | OUT 0x3 |
| 054 |  | AF0B | LD #0xB | Вектор 3 и разрешение прерываний на MR ВУ-3 |
| 055 |  | 1307 | OUT 0x7 |
| 056 |  | 1100 | EI | Разрешение прерываний |
| 057 | DEC\_LOOP | AEDE | LD X | Основная программа. Загрузка X в аккумулятор |
| 058 |  | 0740 | DEC | Декрементация |
| 059 |  | DE08 | CALL NORM\_X | Вызов подпрограммы-нормализатора |
| 05A |  | EEDB | ST X | Сохранение аккумулятора в X |
| 05B |  | CEFB | JUMP DEC\_LOOP | Безусловный переход на 0x57 |
| 05C |  | 0100 | HLT | Останова |
| 05D | F | AC01 | LD &1 | Функция F. Загрузка значения по адресу SP+1 в AC |
| 05E |  | 0500 | ASL | Сдвиг влево (умножение на 2) |
| 05F |  | 4F02 | ADD #2 | Сложение с числом 2 |
| 060 |  | EC01 | ST &1 | Сохранение аккумулятора на стек (SP+1) |
| 061 |  | 0A00 | RET | Возврат |
| 062 | NORM\_X | 7ED4 | CMP X\_MIN | Сравнить переданный (через AC) аргумент с мин.значением по ОДЗ |
| 063 |  | F803 | BLT NORM\_X\_FIX | Если аргумент меньше минимума, то переход на исправление |
| 064 |  | 7ED3 | CMP X\_MAX | Сравнить переданный (через AC) аргумент с макс.значением по ОДЗ |
| 065 |  | F004 | BEQ NORM\_X\_RET | Если аргумент меньше или равен максимуму, то он лежит в ОДЗ. Переход к возврату. |
| 066 |  | F803 | BLT NORM\_X\_RET |
| 067 | NORM\_X\_FIX | 0000 | NOP | Отладочная точка останова (NOP/HLT) |
| 068 |  | AECF | LD X\_MAX | Исправление - загрузка максимального по ОДЗ значения в AC. |
| 069 |  | 0000 | NOP | Отладочная точка останова (NOP/HLT) |
| 06A | NORM\_X\_RET | 0A00 | RET | Возврат |
| 06B | INT1 | 1000 | DI | Запрет прерываний |
| 06C |  | AEC9 | LD X | Загрузить X |
| 06D |  | 0000 | NOP | Отладочная точка останова (NOP/HLT) |
| 06E |  | 0C00 | PUSH | Вызов функции F |
| 06F |  | DEED | CALL F |
| 070 |  | 0800 | POP |
| 071 |  | 0000 | NOP | Отладочная точка останова (NOP/HLT) |
| 072 |  | 1302 | OUT 0x2 | Запись аккумулятора (результата F) в ВУ-1 |
| 073 |  | 1100 | EI | Разрешение прерываний |
| 074 |  | 0B00 | IRET | Возврат из обрабоки прерывания |
| 075 | INT3 | 1000 | DI | Запрет прерываний |
| 076 |  | 1206 | IN 0x6 | Чтение с ВУ-3 в аккумулятор |
| 077 |  | 0000 | NOP | Отладочная точка останова (NOP/HLT) |
| 078 |  | 3EBD | OR X | Логичское ИЛИ-НЕ с X |
| 079 |  | 0280 | NOT |
| 07A |  | DEE7 | CALL NORM\_X | Вызов подпрограммы нормализации X |
| 07B |  | EEBA | ST X | Сохранение аккумулятора в X |
| 07C |  | 0000 | NOP | Отладочная точка останова (NOP/HLT) |
| 07D |  | 1100 | EI | Разрешение прерываний |
| 07E |  | 0B00 | IRET | Возврат из обрабоки прерывания |

## **Часть 2. Описание программы**

### Назначение программы

1. Основная программа должна декременитирует X в цикле.

2. По нажатию кнопки готовности ВУ-1 должен осуществиться вывод результата вычисления функции F(X)=2X+2 на данное ВУ

3. По нажатию кнопки готовности ВУ-3 в X должен записаться результат побитового 'ИЛИ-НЕ' содержимого РД данного ВУ и X.

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

### ОПИ, ОДЗ

(ограничено F: (-65) \* 2 + 2 = -128, 62 \* 2 + 2 = 126)

### Расположение данных

| Адрес | Значение |
| --- | --- |
| 0x36 | X |
| 0x37 | X\_MIN |
| 0x38 | X\_MAX |
| 0x39 | X\_MASK |

### Адреса первой и последней выполняемой команды

| Адрес первой выполняемой команды | Адрес последней выполняемой команды |
| --- | --- |
| 0x50 | - |

## **Часть 3. Методика проверки**

Исходный код в удобном для копирования формате: <https://raw.githubusercontent.com/sultanowskii/itmo-edu/master/opd/lab6/task/task.bcomp>

**Проверка 1: Основная программа**

1. В исходном коде программы изменить NOP на HLT по метке NORM\_X\_FIX и по NORM\_X\_FIX+2
2. Загрузить получившийся код в БЭВМ (скомпилировать)
3. Поменять STOP на RUN
4. Запустить программу в автоматическом режиме (START)
5. Следить за регистром AC в графическом интерфейсе - он должен уменьшаться на 1
6. Дождаться точки останова
7. Убедиться, что в аккумуляторе лежит значение 0xFFBE. Это значит, что подпрограмма “исправления” X заметила вылет за минимальное значение и теперь исправит X
8. Продолжить выполнение программы, дождаться следующей точки останова
9. В аккумуляторе должно находиться максимальное по ОДЗ значение (0x3E), т.к. подпрограмма должна исправить вылет за ОДЗ, внеся в X его максимальное значение
10. Продолжить выполнение программы
11. Проделать шаги 5-7
12. Программа остановлена. Проверка завершена

**Проверка 2: ВУ-1**

1. В исходном коде программы изменить NOP на HLT по меткам INT1+2 и INT1+6
2. В исходном коде программы поменять значение X на необходимое (должно лежать в вышеуказанном ОДЗ), изменив значение ячейки по адресу 0x36 (метка X)
3. Загрузить получившийся код в БЭВМ (скомпилировать)
4. Поменять STOP на RUN
5. Установить “Готовность ВУ-1” (заранее, чтобы программа обработала прерывание именно для изначального, введенного значения X)
6. Запустить программу в автоматическом режиме (START)
7. Дождаться точки останова
8. Убедиться, что в аккумуляторе лежит изначально введенное значение X
9. Продолжить выполнение программы, дождаться следующей точки остановы
10. Убедиться, что в аккумуляторе лежит F=(X \* 2) + 2
11. Программа остановлена. Проверка завершена.

**Проверка 3: ВУ-3**

1. В исходном коде программы изменить NOP на HLT по меткам INT3+2 и INT3+7
2. В исходном коде программы поменять значение X на необходимое (должно лежать в вышеуказанном ОДЗ), изменив значение ячейки по адресу 0x36 (метка X)
3. Загрузить получившийся код в БЭВМ (скомпилировать)
4. Поменять STOP на RUN
5. Ввести интересующее значение Y в ВУ-3. Установить “Готовность ВУ-3” (заранее, чтобы программа обработала прерывание именно для изначального, введенного значения X)
6. Запустить программу в автоматическом режиме (START)
7. Дождаться точки останова
8. Убедиться, что в аккумуляторе лежит Y
9. Продолжить выполнение программы, дождаться следующей точки остановы
10. Убедиться, что в аккумуляторе лежит . Если не вписывается в ОДЗ, проверить, что в аккумуляторе лежит максимальное значение X по ОДЗ.
11. Программа остановлена. Проверка завершена.

# Заключение

В рамках данной лабораторной работы я познакомился с прерываниями и векторами прерывания (обработкой) в БЭВМ, а также вводом-выводом, управляемым прерываниями.