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**

Обмен данными с ВУ по прерыванию

Вариант №3107

Группа: P3131

Выполнил: Хайкин О. И.

Проверил:

Перцев Тимофей Сергеевич

Санкт-Петербург

2022г

Оглавление

[Задание 3](#_Toc101107830)

[Выполнение работы 3](#_Toc101107831)

[Составленная программа (ассемблер) 3](#_Toc101107832)

[Назначение программы и реализуемые ею функции 4](#_Toc101107833)

[Область представления исходных данных и результата 4](#_Toc101107834)

[Область допустимых значений исходных данных и результата 4](#_Toc101107835)

[Расположение в памяти ЭВМ программы, исходных данных и результата 5](#_Toc101107836)

[Адреса первой и последней выполняемых команд 5](#_Toc101107837)

[Методика проверки 5](#_Toc101107838)

[Вывод 6](#_Toc101107839)

# 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 уменьшать на 2 содержимое X (ячейки памяти с адресом 01216) в цикле.
2. Обработчик прерывания должен по нажатию кнопки готовности ВУ-3 осуществлять вывод результата вычисления функции F(X)=-4X+6 на данное ВУ, a по нажатию кнопки готовности ВУ-2 прибавить содержимое РД данного ВУ к Х, результат записать в X
3. Если Х оказывается вне ОДЗ при выполнении любой операции по его изменению, то необходимо в Х записать максимальное по ОДЗ число.

# Выполнение работы

## Составленная программа (ассемблер)

ORG 0x0 ; Настройка векторов прерываний

V0: WORD $DEFAULT,0x180

V1: WORD $DEFAULT,0x180

V2: WORD $INT2,0x180

V3: WORD $INT3,0x180

V4: WORD $DEFAULT,0x180

V5: WORD $DEFAULT,0x180

V6: WORD $DEFAULT,0x180

V7: WORD $DEFAULT,0x180

ORG 0x012

X: WORD 0x0000 ; Ячейка для работы основной программы

LOWEST: WORD 0xFFE2 ; Наименьшее значение ОДЗ

HIGHEST: WORD 0x0021 ; Наибольшее значение ОДЗ

DEFAULT: IRET ; "Пустое" прерывание

START: DI ; Запрещаем прерывания

CLA ; Очистка аккумулятора (загрузка значения 0)

OUT 0x1

OUT 0x3

OUT 0xB

OUT 0xD

OUT 0x11

OUT 0x15

OUT 0x19

OUT 0x1D ; Запрет прерываний всем ВУ кроме ВУ-2 и ВУ-3

LD #0xA ; 1010 - значение для MR КВУ-2

OUT 0x5

LD #0xB ; 1011 - значение для MR КВУ-3

OUT 0x7

MAIN: EI ; Разрешаем прерывания

LD X ; Загружаем в аккумулятор X

CYCLE: SUB #0x2 ; Уменьшение значения на 2

PUSH

CALL CHECK ; Вызов подпрограммы проверки ОДЗ

POP

ST X ; Сохранение значения X

NOP ; Точка отладки (конец цикла)

JUMP CYCLE ; Цикл (вечный)

INT2: NOP ; Точка отладки, начало прерывания с ВУ-2 (сложить DR ВУ-2 с X)

IN 4 ; Читаем значение на ВУ-2

ADD X ; Складываем с X

PUSH

CALL CHECK ; Вызов подпрограммы проверки ОДЗ

POP

ST X ; Сохранение значения X

NOP ; Точка отладки, конец прерывания с ВУ-2

IRET ; Возврат из прерывания

INT3: NOP ; Точка отладки, начало прерывания с ВУ-3 (вывести -4X+6)

PUSH ; Сохраняем в стек текущее значение на аккумуляторе

LD X ; Загружаем текущий X

NEG ; -X

ASL ; -2X

ASL ; -4X

ADD #0x6 ; -4X+6

OUT 6 ; Вывод

POP ; Возвращаем значение аккумулятора

NOP ; Точка отладки, конец прерывания с ВУ-3

IRET ; Возврат из прерывания

CHECK: LD &1 ; Загружаем переданное на проверку число

CMP LOWEST ; Если меньше нижней границы, то

BMI NEWX ; ставим максимальное по ОДЗ значение

CMP HIGHEST ; Если больше верхней границы, то

BPL NEWX ; ставим максимальное по ОДЗ значение

JUMP RETURN ; Если значение входит в ОДЗ, то на выход из подпрограммы

NEWX: LD HIGHEST ; Загрузка максимального по ОДЗ значения

RETURN: ST &1 ; Сохранение в стек

RET ; Возврат из подпрограммы

## Назначение программы и реализуемые ею функции

Назначение: основная программа уменьшает на 2 содержимое X. По прерыванию с ВУ-2 к значению X прибавляется содержимое РД этого ВУ. По прерыванию с ВУ-3 на него выводится значение функции f(x) = -4X+6

## Область представления исходных данных и результата

Исходные данные:

Нижняя граница X (метка LOWEST): знаковое число

Верхняя граница X (метка HIGHEST): знаковое число

## Область допустимых значений исходных данных и результата

Нижняя граница X (метка LOWEST): любое\* число, меньше верхней границы

Верхняя граница X (метка HIGHEST): любое\* число, больше нижней границы

\* нижняя и верхняя границы X определяются из функции и технических ограничений ВУ (8 разрядов под знаковое число результата). Т.е.:

-2^7<=F(x)<=2^7-1

-128<=-4x+6<=127

-134<=-4x<=121

-121<=4x<=134

-30<=x<=33

Т.е. верхняя граница = 33, нижняя граница = -30

## Расположение в памяти ЭВМ программы, исходных данных и результата

**Исходные данные:**

013 – Нижняя граница

014 – Верхняя граница

**Промежуточные данные:**

012 – X

**Программа:**

015 – 049

**Результат:**

…

## Адреса первой и последней выполняемых команд

Первая – 016,

Последняя – …

## Методика проверки

1. Загрузить комплекс программ в память базовой ЭВМ (или скомпилировать код в ассемблере)
2. Выполнить шаги необходимой проверки.

**Проверка работы «основной программы»:**

1. Заменить точку отладки по адресу 02B на команду HLT (или заменить команду NOP с комментарием «конец цикла» на HLT в коде ассемблера и скомпилировать)
2. Запустить программу в режиме «Работа» с адреса 016 (при компиляции с использованием ассемблера адрес начала выставлен автоматически)
3. Дождаться останова
4. Записать значение на аккумуляторе
5. Продолжить выполнение программы
6. Дождаться очередного останова
7. Сравнить значение на аккумуляторе с записанным. Корректное выполнение программы приведёт к уменьшению этого значения на 2.
8. Продолжить выполнение программы
9. При желании повторять шаги 6-8, пока значение на аккумуляторе не приблизится к значению нижней границы
10. Удостовериться, что при выходе значения на аккумуляторе за пределы ОДЗ, значение автоматически изменится на максимальное по ОДЗ до очередного останова.

**Проверка работы прерывания с ВУ-2**

1. Заменить точки отладки по адресам 02D и 034 на команды HLT (или заменить команды NOP с комментариями «начало прерывания с ВУ-2» и «конец прерывания с ВУ-2» на HLT в коде ассемблера и скомпилировать)
2. Запустить программу в режиме «Работа» с адреса 016 (при компиляции с использованием ассемблера адрес начала выставлен автоматически)
3. Установить конкретное значение на РД ВУ-2. Записать его.
4. Установить «Готовность ВУ-2».
5. Дождаться останова
6. Записать текущее значение на аккумуляторе. В зависимости от того, в какой момент исполнения программы было вызвано прерывание, значение на аккумуляторе может быть либо равно значению X, либо быть на 2 меньше его.
7. Вручную сложить значение на РД ВУ-2 и значение на аккумуляторе. Предсказать итоговое значение с учётом ОДЗ (если результат сложения выходит за рамки ОДЗ, то за результат сложения берётся максимальное по ОДЗ число). Учесть возможность отличия значения на аккумуляторе от реального значения X (сосчитать сумму таким же методом для РД ВУ-2 и значения на аккумуляторе+2). Записать получившиеся результаты.
8. Продолжить выполнение программы
9. Дождаться останова
10. Сравнить значение на аккумуляторе с записанными результатами. При корректном выполнении программы значение на аккумуляторе должно совпасть с одним из сосчитанных
11. Продолжить выполнение программы
12. При желании повторить пункты 3-11 с другими значением на РД ВУ-2

**Проверка работы прерывания с ВУ-3**

1. Заменить точки отладки по адресам 036 и 03F на команды HLT (или заменить команды NOP с комментариями «начало прерывания с ВУ-3» и «конец прерывания с ВУ-3» на HLT в коде ассемблера и скомпилировать)
2. Запустить программу в режиме «Работа» с адреса 016 (при компиляции с использованием ассемблера адрес начала выставлен автоматически)
3. Установить «Готовность ВУ-3»
4. Дождаться останова
5. Записать текущее значение на аккумуляторе. В зависимости от того, в какой момент исполнения программы было вызвано прерывание, значение на аккумуляторе может быть либо равно значению X, либо быть на 2 меньше его.
6. Вручную сосчитать результат применения функции f(x)=-4x+6 к записанному значению. Учесть возможность отличия значения на аккумуляторе от реального значения X (сосчитать значение функции для значения на аккумлутяоре+2). Записать получившиеся результаты.
7. Продолжить выполнение программы
8. Дождаться останова
9. Сравнить значение на РД ВУ-3 с сосчитанными вручную результатами. При корректном выполнении программы значение на РД ВУ-3 должно совпасть с одним из записанных.
10. Убедиться, что значение на аккумуляторе совпадает с записанным на предыдущем останове
11. Продолжить выполнение программы
12. При желании повторить пункты 3-11

# Вывод

В ходе выполнения шестой лабораторной работы я изучил организацию процесса прерывания программы и исследовал порядок функционирования БЭВМ при обмене данными в режиме прерывания программы