Федеральное государственное автономное образовательное учреждение высшего образования «Национальный исследовательский университет ИТМО»

Факультет программной инженерии и компьютерной техники

### Лабораторная работа №6

Вариант №8829

Выполнил

Макогон Ярослав Вадимович

Номер группы: P3118

Проверил

Ермаков М.К.

**Содержание**

[Задание 3](#_Toc198367257)

[Решение 4](#_Toc198367258)

[Описание программы 5](#_Toc198367259)

[Методика проверки 6](#_Toc198367260)

[Выводы 7](#_Toc198367261)

### 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 01C16) в цикле.
2. Обработчик прерывания должен по нажатию кнопки готовности ВУ-3 осуществлять вывод результата вычисления функции F(X)=5X+7 на данное ВУ, a по нажатию кнопки готовности ВУ-2 выполнить операцию побитового 'Исключающее ИЛИ' 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.

### Решение

ORG 0x0

V0: WORD $DEFAULT, 0x180

V1: WORD $DEFAULT, 0x180

V2: WORD $V2FUN, 0x180

V3: WORD $V3FUN, 0x180

V4: WORD $DEFAULT, 0x180

V5: WORD $DEFAULT, 0x180

V6: WORD $DEFAULT, 0x180

V7: WORD $DEFAULT, 0x180

DEFAULT:

IRET

ORG 0x01D

DL: WORD 0xFFE5; down limit

UL: WORD 0x19; up limit

START: CLA

LD #0xA ; 1010

OUT 5

LD #0xB ; 1011

OUT 7

EI

NOP

CYCLE:

DI

LD $X

INC

ST $X

**CMP** DL

**BMI** SETDL

LD $X

**CMP** UL

**BPL** SETDL

JUMP OUTX

SETDL: LD $DL;

OUTX: ST $X;

EI

NOP

JUMP CYCLE

HLT

TMP: WORD ?

V2FUN:

IN 4

ST $TMP

LD $TMP

NOT

**AND** $X

SWAM $X

NOT

**AND** $TMP

OR $X

ST $X

IRET

V3FUN:

LD $X

**ADD** $X

**ADD** $X

**ADD** $X

**ADD** $X

**ADD** #0x7

OUT 6

IRET

ORG 0x01C

X: WORD 0

### Описание программы

|  |  |
| --- | --- |
| Область представления данных | X, DL, UL, TMP – восьмибитные знаковые числа |
| Область допустимых значений | DL, UL, TMP ∈ [-128, 127]  F(X) = 5X + 7  F(X) : [-128, 127]  X : [-27, 24] |
| Расположение в памяти исходных данных и результатов | X в 0x01C  DL в 0x011  UL в 0x012 |
| Адреса первой и последней выполняемой команды | 0x013 – первая команда  0x025 – последняя команда  (без учета обработки прерываний) |
| Назначение программы | Прибавляет 1 к X в спинлупе. При выходе из X одз замена его значения на минимальное. При готовности ВУ-3 вывод 5X+7 на него. П24 ри готовности ВУ-2 - запись результата операции побитового искл. или между X и значением на этом ВУ в X. |

### 

### Методика проверки

Для ВУ-2:

1. Загрузить текст программы в БЭВМ.
2. Заменить NOP по нужному адресу на HLT.
3. Заменить X на целевой -1.
4. Посчитать значение побитового искл. Или между x и значением на ВУ-2 вручную.
5. Запустить программу в режиме РАБОТА.
6. Дождаться останова.
7. Ввести данные на ВУ-2
8. Установить «Готовность ВУ-2».
9. Продолжить выполнение
10. Дождаться останова.
11. Проверить значение X в ячейке 0x01C
12. Сравнить его с посчитанным вручную.

Для ВУ-3:

1. Загрузить текст программы в БЭВМ.
2. Заменить NOP по нужному адресу на HLT.
3. Заменить X на целевой -1.
4. Посчитать значение функции F(X) = 5X + 7 на данном X вручную
5. Запустить программу в режиме РАБОТА.
6. Дождаться останова.
7. Установить «Готовность ВУ-3».
8. Продолжить выполнение
9. Дождаться останова.
10. Проверить значение X в ячейке 0x01C
11. Сравнить его с посчитанным вручную.

|  |  |  |  |
| --- | --- | --- | --- |
| **ВУ-2** | | | |
| **X (После +1)** | **Значение на ВУ-2** | **Ожидание в X** | **Реальное X** |
| 24(10) = 00011000(2) = 18(16) | 10100100(2) = A4(16) | 10111100(2) = BC(16) | 00BC(16) |

|  |  |  |
| --- | --- | --- |
| **ВУ-3** | | |
| **X (После +1)** | **Ожидание на ВУ-3** | **Реальн. На ВУ-3** |
| 24(10) = 00011000(2) = 18(16) | 127(10) = 7FFF(16) | 0111 1111(2) = 7FFF(16) |

### Выводы

* Узнал, как организованы прерывания в БЭВМ
* Изучил методы отладки кода в БЭВМ