

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 № 86866

Выполнил:

Студент группы Р3107

Пшеничников Артём Дмитриевич

Принял:

Осипов Святослав Владимирович

Содержание

Задание (Вариант 55655)

Текст исходной программы

Описание программы

Таблица трассировки

Заключение

Задание (Вариант 86866)

Введите номер варианта

1. Основная программа должна уменьшать на 2 содержимое X (ячейки памяти с адресом 005₁₆) в цикле.
2. Обработчик прерывания должен по нажатию кнопки готовности ВУ-1 осуществлять вывод результата вычисления функции $F(X)=5X-6$ на данное ВУ, а по нажатию кнопки готовности ВУ-2 выполнить операцию побитового 'Исключающее ИЛИ-НЕ' содержимого РД данного ВУ и X, результат записать в X
3. Если X оказывается вне ОДЗ при выполнении любой операции по его изменению, то необходимо в X записать максимальное по ОДЗ число.

Текст исходной программы

```
    ORG 0x0
V0: WORD $DEFAULT, 0x180 ; Вектор прерывания #0
V1: WORD $INT_1, 0x180   ; Вектор прерывания #1
V2: WORD $INT_2, 0x180   ; Вектор прерывания #2
V3: WORD $DEFAULT, 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: IRET ; Дефолтный вектор

X:      WORD 0x084 ; Адрес ячейки X (по заданию 5, но оно пересекается с векторами)
start_value: WORD 0x0042 ; стартовое значение X

START: NOP
      LD start_value ; записали стартовое значение в X
      CALL clamp_var
      ST (X)

      ;
      DI             ; запрет прерываний
      LD #0x9        ; Разрешение прерываний и вектор 1
      OUT 0x3         ; ВУ-1 на вектор INT_1
      LD #0xA        ; Разрешение прерываний и вектор 2
      OUT 0x5         ; ВУ-2 на вектор INT_2
      EI             ; Прерываемся
      ;

main:  DI
      LD (X)
      SUB #0x02
      CALL clamp_var
      ST (X)
      OUT 0x06        ; добавлен вывод текущего X на ВУ-3
      EI
      NOP             ; прерываемся только здесь
      JUMP main

min_var: WORD 0x0000
max_var: WORD 0x00FF
clamp_var: CMP max_var ; ограничение переменной в АС от min_var до max_var
           BNC fix_max_var ; если максимум строго меньше значения
           DEC
           CMP min_var
           BNS fix_min_var ; если минимум больше или равен значению-1 => строго больше
           INC
           JUMP end_clamp_var
fix_max_var: LD max_var
            JUMP end_clamp_var
fix_min_var: LD max_var ; LD min_var ; по заданию пишем максимальное число
            JUMP end_clamp_var
end_clamp_var: RET

cntr: WORD ?
var_a: WORD 0x05
var_b: WORD 0x06
calculate_func: LD var_a ; Подсчёт функции a*X - b, X - в стеке
               ST cntr ; записали на сколько умножить
               CLA
loop1: ADD &1 ; сложили в цикле
      LOOP cntr
      JUMP loop1
      SUB var_b ; минус b
      ST &1 ; записали в стек
      RET

buffer: WORD ? ; ;
```

1 2 3 4 5

```

not_xor: NOP                ; исключающее ИЛИ-НЕ (00-1, 01-0, 10-0, 11-1) (not ((a or b) and (not (a and b))))
        LD &1
        AND &2              ; 5
        NOT                 ; 4
        ST buffer
        LD &1
        OR &2               ; 2
        AND buffer          ; 3
        NOT                 ; 1
        ST &2              ; запись в стек
        POP
        SWAP                ; убрали ненужный аргумент
        RET

; прерывание по BY-1
INT_1:  NOP                ; подсчёт функции и вывод на BY-1
        LD (X)              ; загружаем X в AC
        PUSH
        CALL calculate_func ; Подсчёт функции
        POP
        CALL c_lamp_var
        OUT 0x02            ; вывод на BY-1
        IRET

mask:   WORD 0x00FF        ; прерывание по BY-2
INT_2:  NOP                ; подсчёт исключающего ИЛИ-НЕ и запись в X
        LD (X)              ; загружаем X в AC
        PUSH
        IN 0x04
        PUSH
        CALL not_xor        ; Подсчёт исключающего ИЛИ-НЕ
        POP
        ; AND #0xFF ; обрежем мусор почему не работает?
        AND mask            ; обрежем мусор
        CALL c_lamp_var
        ST (X)
        IRET

```

Описание программы

Назначение программы:

1. Декрементация ячейке X на 2 в цикле
2. По готовности ВУ-1 вывод на ВУ-1 результата $F(X)=5X-6$
3. По готовности ВУ-2 запись в X NOT-XOR(РД2, X)
4. При вычислении X проверять ОДЗ, если X вне ОДЗ, то $X = \max(\text{ОДЗ})$

Область представления:

X – адрес ячейки
start_value – начальное значение ячейки X
min_var – нижняя граница ОДЗ
max_var – верхняя граница ОДЗ

Область определения:

Последняя исполняемая команда = $0x060 < X < \text{Макс. Размер стека} = 0x7FB$
 $0 \leq \text{start_value} \leq 2^{15} - 1$ (при записи в X применится ОДЗ)
min_var = $0x0000$ (по заданию)
max_var = $0x00FF$ (по заданию)

Расположение в памяти ЭВМ программы: 000-060 + переменная X

Расположение данных: X

Расположение результата: ВУ-1, X

Первая выполняемая команда: 013

Последняя выполняемая команда: 060

Таблица трассировки

При трассировке не работают прерывания

Заключение

В ходе выполнения лабораторной работы я освоил принципы работы с прерываниями в БЭВМ, научился управлять изменением данных в памяти, взаимодействовать с внешними устройствами и работать с подпрограммами в прерываниях.