

Санкт-Петербургский национальный исследовательский
университет информационных технологий, механики и
оптики

Мегафакультет компьютерных технологий и управления
Факультет программной инженерии и компьютерной
техники



Лабораторная работа №6
по основам профессиональной деятельности

Вариант: 72853

Группа: P3114

Студент: Лагус

Максим Сергеевич

Преподаватель: Перминов Илья Валентинович

г. Санкт-Петербург

Апрель, 2022

Задание:

Лабораторная работа №6

По выданному преподавателем варианту разработать и исследовать работу комплекса программ обмена данными в режиме прерывания программы. Основная программа должна изменять содержимое заданной ячейки памяти (X), которое должно быть представлено как знаковое число. Область допустимых значений изменения X 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 X в соответствии с вариантом задания, а также игнорировать все необработываемые прерывания.

Введите номер варианта

1. Основная программа должна инкрементировать содержимое X (ячейки памяти с адресом 031_{16}) в цикле.
2. Обработчик прерывания должен по нажатию кнопки готовности ВУ-3 осуществлять вывод результата вычисления функции $F(X)=6X+4$ на данное ВУ, а по нажатию кнопки готовности ВУ-2 выполнить операцию побитового '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 минимальное по ОДЗ число.

Выполнение работы:

Текст исходной программы

```
1      ORG      0X000; Инициализация начальных значений векторов прерываний
2
3  V0:  WORD    $DEFAULT, 0X180
4  V1:  WORD    $DEFAULT, 0X180
5  V2:  WORD    $INT2, 0X180
6  V3:  WORD    $INT3, 0X180
7  V4:  WORD    $DEFAULT, 0X180
8  V5:  WORD    $DEFAULT, 0X180
9  V6:  WORD    $DEFAULT, 0X180
10 V7:  WORD    $DEFAULT, 0X180
11
12 DEFAULT: IRET
13
14
15      ORG      0X015
16  X:  WORD    0X031; Адрес элемента, с которым будет работать программа
17
18  START: DI          ; Запрещаем прерывания чтобы обеспечить корректную инициализацию
19        CLA          ; регистров прерываний для контроллеров ВУ
20
21        OUT          0X001; Запрещаем прерывания для неиспользуемых ВУ, записывая
22        OUT          0X003; 0 в MR
23        OUT          0X00B
24        OUT          0X00F
25        OUT          0X013
26        OUT          0X017
27        OUT          0X01B
28
29        OUT          0X01F
30
31        LD          #0X0A ; Разрешаем прерывания для ВУ-2, и задаём второй вектор
32        OUT          0X005; (1000|0010) = (1010) = 0XA
33
34        LD          #0X0B ; Разрешаем прерывания для ВУ-3, и задаём третий вектор
35        OUT          0X007; (1000|0011) = (1011) = 0XB
36
37        JUMP       $PROG; Переходим к основной программе
38
39  PROG:  ORG      0X040
40        DI          ; Запрещаем прерывания, чтобы обеспечить атомарность операций
41        LD          (X)
42        INC          ; Увеличиваем переменную, и вызываем подпрограмму проверки ОДЗ
43        CALL       $CHECK
44        ST          (X)
45        NOP         ; Точка останова для отладки
46        EI
47        JUMP       PROG ; Повторяем основной цикл выполнения
48
49  MIN:  WORD    0XFFD6 ; Максимальное и минимальное возможные значения для
50  MAX:  WORD    0X2A ; обрабатываемого числа при заданной функции F(X)
51
52  CHECK: NOP         ; Проверка принадлежности к ОДЗ
53        NOP         ; Точка отладки
54
```

```

55      CMP    MAX    ; Загрузка минимума, если X >= MAX
56      BLT    CHECKM
57      JUMP    LDMIN
58
59 CHECKM: CMP    MIN    ; Загрузка минимума, если X < MIN
60      BGE    RETN
61      HLT
62      JUMP    LDMIN
63
64 RETN:  RET
65
66 LDMIN: LD     MIN    ; Загрузка минимального значения, если X вышел за пределы ОДЗ
67      RET
68
69
70 INT2:  DI          ; Обработка прерываний для ВУ-2
71      NOP          ; Точка отладки
72
73      IN     0X004; Вычисление логического ИЛИ-НЕ
74      OR     (X)
75      NOT
76      ST     (X)
77      CALL   $CHECK
78
79      NOP
80      IRET
81

```

```

81
82 VAR1:  WORD    ?    ; Вспомогательная переменная для вычисления значения функции
83
84 INT3:  DI          ; Обработка прерываний для ВУ-3
85      NOP          ; Точка отладки
86
87      LD     (X)    ; Вычисление F(X) = 6X + 4
88      ASL
89      ST     VAR1
90      ASL
91      ADD    VAR1
92      ADD    #0X04
93
94      OUT    0X006; Вывод полученного значения на ВУ-3
95      HLT
96      IRET

```

Описание программы

1) Расположение программы в памяти

Вектора прерываний: 0x00 - 0x10

Программа и некоторые переменные: 0x40 - 0x78

2) Область представления

X - ячейка, содержащая адрес переменной X, которая представляется как 16-разрядное знаковое число

MIN, MAX - переменные, содержащие знаковые 16-разрядные числа

3) Область допустимых значений

По заданию, $F(X) \in [-128; 127]$, так как значение этой функции выводится на ВУ-3, которое ограничено 8-битной разрядной сеткой

Тогда: $-128 \leq 6x + 4 \leq 127$

$-42 \leq x \leq 41$

Значит, $X \in [-42; 41]$.

4) Методики проверки

Проверка обработки прерываний

- 1) Загрузить и скомпилировать текст программы в БЭВМ
- 2) Заменить NOP на HLT в начале и в конце подпрограмм обработки прерываний
- 3) Запустить основную программу
- 4) Установить "Готовность ВУ-3"
- 5) Дождаться останова
- 6) Записать содержимое аккумулятора в момент останова программы
- 7) Посчитать значение выражения $6 \cdot AC + 4$
- 8) Продолжить выполнение программы и дождаться следующего останова
- 9) Сравнить выведенное на ВУ-3 значение с посчитанным ранее результатом
- 10) Ввести в ВУ-2 число $0x1C$
- 11) Установить "Готовность ВУ-2"
- 12) Записать полученные значений во время остановов, до и после обработки прерывания, сравнить с ожидаемыми

Проверка удовлетворения ОДЗ

- 1) Загрузить и скомпилировать текст программы в БЭВМ
- 2) Заменить NOP на HLT в начале и в конце подпрограммы проверки ОДЗ
- 3) Запустить основную программу в автоматическом режиме
- 4) Дождаться переполнения области допустимых значений и срабатывания точки останова
- 5) Проверить присвоение переменной минимального значений

Трассировка программы