

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

Мегафакультет компьютерных технологий и управления  
Факультет программной инженерии и компьютерной техники

**Отчёт**  
**по лабораторной работе №6**  
**по дисциплине «Основы профессиональной**  
**деятельности»**

Вариант 91813

Группа: Р3118

Студент: Кожухин Иван Алексеевич

Преподаватель: Осипов Святослав Владимирович

Санкт-Петербург  
2023

# Содержание

<b>1</b>	<b>Задание</b>	<b>1</b>
<b>2</b>	<b>Выполнение задания</b>	<b>2</b>
2.1	Код программы на языке Ассемблера БЭВМ . . . . .	2
2.2	Область представления и допустимых значений . . . . .	5
2.3	Методика проверка программы . . . . .	6
<b>3</b>	<b>Вывод</b>	<b>8</b>

# 1 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 ( $X$ ), 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 необрабатываемые прерывания.

Текст варианта:

1. Основная программа должна уменьшать на 3 содержимое  $X$  (ячейки памяти с адресом  $04C_{16}$ ) 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 'И-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.

## 2 Выполнение задания

### 2.1 Код программы на языке Ассемблера БЭВМ

```
ORG 0x0
```

```
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
default: IRET
```

```
disable_redundant_int:
```

```
CLA
```

```
OUT 0x1
```

```
OUT 0x3
```

```
OUT 0xB
```

```
OUT 0xE
```

```
OUT 0x12
```

```
OUT 0x16
```

```
OUT 0x1A
```

```
OUT 0x1E
```

```
RET
```

```
enable_necessary_int:
```

```
LD #0xA
```

```
OUT 0x5
```

```
LD #0xB
```

```
OUT 0x7
```

```
RET
```

```
ORG 0x4C
```

```
x: WORD 20
```

```
x_addr: WORD 0x4C
```

```
max: WORD 21
```

```
min: WORD -22
```

```
not_x: WORD 0
```

```

start:
DI
CALL disable_redundant_int
CALL enable_necessary_int
CLA
EI
main_loop:
LD x
PUSH
SUB #3
CALL AC_check
PUSH
PUSH
PUSH
PUSH
LD &3
ST &2
LD &4
ST &1
LD x_addr
ST &0
CALL compare_and_set
JUMP main_loop

AC_check:
CMP max
BMI min_check
JUMP correction
min_check:
CMP min
BPL check_passed
correction:
LD max
DEC
check_passed: RET

deref: WORD 0

compare_and_set:
PUSHF
DI
LD &2
ST deref

```

```

LD (deref)
CMP &3
BEQ then
JUMP exit
then:
LD &4
ST (deref)
exit:
POPF
SWAP
ST &3
POP
POP
POP
RET

int2:
PUSH
LD x
NOP
NOT
ST not_x
IN 0x4
AND not_x
CALL AC_check
ST x
NOP
POP
IRET

int3:
PUSH
LD x
NOP
ADD x
ADD x
ASL
ADD #4
NOP
OUT 0x6
POP
IRET

```

## 2.2 Область представления и допустимых значений

Один из индикаторов ВУ мы будем использовать как знаковый регистр, остальные - как числовые. Так как на ВУ-3 всего 8 регистров, то область допустимых значений для функции  $F(X)$  имеет следующий вид:

$$-2^7 \leq F(X) \leq 2^7 - 1$$

$$-128 \leq F(X) \leq 127$$

Так как  $F(X) = 6X + 4$ , то ОДЗ для  $X$  имеет следующий вид:

$$\frac{-128-4}{6} \leq X \leq \frac{127-4}{6}$$

$$-22 \leq X \leq 20$$

Таким образом, когда значение  $X$  будет выходить за допустимые границы, ему будет присваиваться значение 20.

## 2.3 Методика проверка программы

Проверка обработки прерываний:

1. Загрузить код в БЭВМ, скомпилировать;
2. Заменить инструкцию NOP по нужному адресу на HLT;
3. Запустить программу в режиме «Работа»;
4. Нажать кнопку «Готов» на ВУ-3;
5. Дождаться остановки;
6. Записать текущее значение X:
  - i) Запомнить текущее состояние регистра IP;
  - ii) Ввести в клавишный регистр адрес X (0x4C);
  - iii) Нажать «Ввод адреса», нажать «Чтение»;
  - iv) Записать значение регистра DR;
  - v) Вернуть регистр IP в ранее зафиксированное состояние;
7. Записать результат обработки прерывания (содержимое DR ВУ-3);
8. Рассчитать ожидаемое значение обработки прерывания;
9. Нажать «Продолжение»;
10. Ввести на ВУ-2 произвольную комбинацию нулей и единиц;
11. Нажать кнопку «Готов» на ВУ-2;
12. Дождаться остановки;
13. Записать текущее значение X (см. п. 6);
14. Нажать «Продолжение»;
15. Записать текущее значение X (см. п. 6);
16. Рассчитать ожидаемое значение обработки прерывания.

Проверка основной программы:

1. Загрузить код в БЭВМ, используя максимальное значение X как исходное, скомпилировать;
2. Запустить программу в режиме «Останов»;
3. Пройти достаточное число итераций программы, чтобы убедиться в корректности изменения X, а также в правильность коррекции при выходе X за ОДЗ.



Прерывание ВУ-3		
АС	Ожидаемое значение	DR ВУ-3
0x14 (20)	0x7C (124)	0x7C (124)
0x2 (2)	0x10 (16)	0x10 (16)
0xFFED (-19)	0xFF92 (-110)	0xFF92 (-110)

  

Прерывание ВУ-2			
АС	DR ВУ-2	АС после вычислений	Сохранённый результат
0x5 (5)	0x35A	0x35A (858)	0x14 (20)
0x5 (5)	0x1	0x0	0x0
0x5 (5)	0x42	0x42 (66)	0x14 (20)

  

Основная программа		
АС	Ожидаемое значение	Сохранённый результат
0x14 (20)	0x11 (17)	0x11 (17)
0x0	0xFFFFD (-3)	0xFFFFD (-3)
0xFFEA (-22)	0x14 (20)	0x14 (20)

Рис. 1: Ход проверки

### 3 Вывод

В ходе выполнения лабораторной работы были изучены: организация программно-управляемого обмена данными в режиме прерывания программы в БЭВМ; команды работы разрешения/запрещения прерываний в БЭВМ; вектора прерывания и преимущества их использования.



Рис. 2: Изображение в стиле «интернет-мэм»