

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»
Выполнение комплекса программ
Вариант №6418

Выполнил:
Дядев Владислав Александрович
Группа Р3131
Проверила:
Остапенко Ольга Денисовна

Санкт-Петербург
2025

Содержание

Задание.....	3
Порядок выполнения.....	4
Исходная программа	4
Область допустимых значений	4
Область представления.....	5
Расположение в памяти ЭВМ.....	5
Методика проверки программы	6
Вывод	8

Задание

Вариант №6418

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. Вариант задания представлен на Рисунок 1.

1. Основная программа должна инкрементировать содержимое X (ячейки памяти с адресом $03A_{16}$) в цикле.
2. Обработчик прерывания должен по нажатию кнопки готовности ВУ-1 осуществлять вывод результата вычисления функции $F(X) = -2X + 6$ на данное ВУ, а по нажатию кнопки готовности ВУ-3 вычесть содержимое РД данного ВУ из 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.

Рисунок 1 – Задание

Порядок выполнения

Исходная программа

```
1  ORG 0x0 ; Инициализация векторов прерывания
2  V0: WORD $DEFAULT, 0x180 ; Вектор прерывания #0
3  V1: WORD $INT1, 0x180 ; Вектор прерывания #1
4  V2: WORD $DEFAULT, 0x180 ; Вектор прерывания #2
5  V3: WORD $INT3, 0x180 ; Вектор прерывания #3
6  V4: WORD $DEFAULT, 0x180 ; Вектор прерывания #4
7  V5: WORD $DEFAULT, 0x180 ; Вектор прерывания #5
8  V6: WORD $DEFAULT, 0x180 ; Вектор прерывания #6
9  V7: WORD $DEFAULT, 0x180 ; Вектор прерывания #7
10
11  DEFAULT: IRET ; Обработка прерываний по умолчанию
12
13  ORG 0x03A ; Инициализация необходимых переменных
14  X: WORD ?
15  MIN: WORD 0xFFFF ; Минимальное значение по ОДЗ (~60)
16  MAX: WORD 0x0042 ; Максимальное значение по ОДЗ (66)
17
18  ORG 0x20 ; Начало программы
19  START: DI ; Запрет прерываний
20          CLA
21          LD #0x9 ; Загрузка в регистр MR (1000|0001=1001) – разрешение прерываний и вектор #1
22          OUT 3 ; Разрешение прерываний для ВУ-1
23          LD #0xB ; Загрузка в регистр MR (1000|0011=1011) – разрешение прерываний и вектор #3
24          OUT 7 ; Разрешение прерываний для ВУ-3
25          EI ; Разрешение прерываний
26
27  MAIN: DI ; Запрет прерываний
28          LD SX ; Загружаем X
29          INC ; Инкрементируем
30          CALL CHECK ; Проверим, что значение не вышло за границы ОДЗ или заменяем его
31          ST SX ; Сохраняем новое значение X
32          EI ; Разрешение прерываний
33          JUMP MAIN ; Спин-луп
34
35  CHECK: CMP $MIN ; Сравниваем X с минимумом по ОДЗ
36          BPL CHECK_MAX ; Если X > MIN, то переход на сравнение с максимумом по ОДЗ
37          JUMP LD_MIN ; Иначе будем загружать минимум по ОДЗ в X
38          CHECK_MAX: CMP $MAX ; Сравниваем X с максимумом по ОДЗ
39          BMI RETURN ; Если X < MAX, то переход на возврат, т.к. X лежит в ОДЗ
40          LD_MIN: LD $MIN ; Иначе загружаем минимум по ОДЗ в X
41          RETURN: RET
42
43  INT1: DI ; Обработка прерывания на ВУ-1
44          LD X ; Загрузили X
45          NOP
46          ASL ; Умножение на 2
47          NEG ; Умножение на -1
48          ADD #6 ; Прибавление 6
49          NOP
50          OUT 2 ; Вывод результата функции на ВУ-1
51          EI ; Конец обработки прерывания на ВУ-1
52          IRET ; Возврат из обработки прерывания
53
54  INT3: DI ; Обработка прерывания на ВУ-3
55          IN 6 ; Получение значения с ВУ-3
56          NOP
57          NEG ; Умножаем значение с ВУ-3 на -1
58          ADD X ; Прибавляем X
59          ST X ; Обновляем значение X
60          NOP
61          EI ; Конец обработки прерывания на ВУ-3
62          IRET ; Возврат из обработки прерывания
```

Код также доступен в репозитории на github:

<https://github.com/Alvas07/ITMO/blob/main/1-2%20OPD/Lab6/lab6.asm>

Область допустимых значений

$$F(X) = -2X + 6$$

$$-128 \leq -2X + 6 \leq 127$$

$$-134 \leq -2X \leq 121$$

$$-60 \leq X \leq 67$$

Так как в программе X инкрементируется, то ОДЗ будет:

$$-61 \leq X \leq 66$$

Имеем:

$$X = [0xFFFFD; 0x0042]$$

Область представления

X, MIN, MAX – знаковое 16-ричное число

DR KBY – знаковое 8-разрядное число

Расположение в памяти ЭВМ

Векторы прерываний: 0x0-0xF

Переменные: 0x03A-0x03C

Программа: 0x03D-0x04A

Методика проверки программы

Проверка обработки прерываний:

1. Загрузить текст программы в БЭВМ.
2. Заменить NOP по нужному адресу на HLT.
3. Запустить программу в режиме РАБОТА.
4. Установить «Готовность ВУ-1».
5. Дождаться останова.
6. Записать текущее значение X из памяти БЭВМ:
 1. Запомнить текущее состояние счетчика команд.
 2. Ввести в клавишный регистр значение 0x03A
 3. Нажать «Ввод адреса».
 4. Нажать «Чтение».
 5. Записать значение регистра данных.
 6. Вернуть счетчик команд в исходное состояние.
7. Записать результат обработки прерывания – содержимое DR контроллера ВУ-1
8. Рассчитать ожидаемое значение обработки прерывания
9. Нажать «Продолжение».
10. Ввести в ВУ-3 произвольное число, записать его
11. Установить «Готовность ВУ-3».
12. Дождаться останова.
13. Записать текущее значение X из памяти БЭВМ, также, как и в пункте 6.
14. Нажать «Продолжение».
15. Записать текущее значение X из памяти БЭВМ, также, как и в пункте 6.
16. Рассчитать ожидаемое значение переменной X после обработки прерывания

17. Сравнить значения, полученные в пунктах 15, 16. Если они равные – программа работает верно.

Проверка основной программы:

1. Загрузить текст программы в БЭВМ.
2. Записать в переменную X минимальное по ОДЗ значение (-61)
3. Запустить программу в режиме останова.
4. Пройти нужное количество шагов программы, убедиться, что при увеличении X на 1 после момента, когда он станет равен 66, происходит сброс значения в минимальное по ОДЗ.

Вывод

В ходе данной лабораторной работы я изучил обмен данными с ВУ-1 и ВУ-3 в режиме прерываний. Научился писать программы с прерываниями на Ассемблер БЭВМ.