

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

Факультет программной инженерии и компьютерной техники

Лабораторная работа №6

Обмен данными с ВУ по прерыванию

Вариант №12345

Выполнил

Лабин Макар Андреевич

группа Р3131

Проверил

Обляшевский Севастьян Александрович

г. Санкт-Петербург, 2025

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 необрабатываемые прерывания.

Введите номер варианта

1. Основная программа должна инкрементировать содержимое X (ячейки памяти с адресом 046_{16}) в цикле.
2. Обработчик прерывания должен по нажатию кнопки готовности ВУ-3 осуществлять вывод результата вычисления функции $F(X) = -7X + 7$ на данное ВУ, а по нажатию кнопки готовности ВУ-2 выполнить операцию побитового 'И' 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

1. Текст разработанной программы:

| Адрес | Код | Мнемоника | Комментарий |
|------------------------------------|------|-----------|--------------------------------------|
| Векторы прерывания | | | |
| 0 | 0016 | \$FUNC | Вектор прерывания #0 |
| 1 | 0180 | PS0 | |
| 2 | 0010 | \$LAND | Вектор прерывания #1 |
| 3 | 0180 | PS1 | |
| LAND: Функция побитового «И» | | | |
| 10 | 0000 | NOP | Отладочная точка останова. |
| 11 | 0C00 | PUSH | Сохраняем AC. |
| 12 | 1204 | IN 4 | |
| 13 | 2E32 | AND X | $X \& (BY-2) \rightarrow X$ |
| 14 | EE31 | ST X | |
| 15 | 0000 | NOP | Отладочная точка останова. |
| 16 | 0800 | POP | Восстанавливаем AC. |
| 17 | 0B00 | IRET | Возврат из обработки прерывания. |
| FUNC: Функция вывода значения F(X) | | | |
| 18 | 0000 | NOP | Отладочная точка останова. |
| 19 | 0C00 | PUSH | Сохраняем AC. |
| 1A | AE2B | LD X | |
| 1B | 0500 | ASL | |
| 1C | 0500 | ASL | |
| 1D | 4E28 | ADD X | |
| 1E | 4E27 | ADD X | $F(X) = -7 * X + 7 \rightarrow BY-3$ |
| 1F | 4E26 | ADD X | |
| 20 | 0780 | NEG | |
| 21 | 4F07 | ADD #7 | |
| 22 | 1306 | OUT 6 | |
| 23 | 0000 | NOP | Отладочная точка останова. |
| 24 | 0800 | POP | Восстанавливаем AC. |
| 25 | 0B00 | IRET | Возврат из обработки прерывания. |

| Основная программа | | | |
|--------------------|------|----------|--|
| 46 | 0000 | X | Переменная X. |
| 47 | 1000 | DI | Запрещаем прерывания. |
| 48 | AF08 | LD #8 | Привязываем векторы к ВУ. |
| 49 | 1307 | OUT 7 | |
| 4A | 0700 | INC | |
| 4B | 1305 | OUT 5 | |
| 4C | 1100 | EI | Разрешаем прерывания. |
| 4D | 0200 | CLA | Обнуляем X. |
| 4E | EEF6 | ST X | |
| 4F | 1000 | DI | Запрещаем прерывания для атомарности инкремента. |
| 50 | AAF5 | LD X | X++ |
| 51 | 0700 | INC | |
| 52 | 7F0C | CMP #13 | Избегаем переполнения в F(X). |
| 53 | F801 | BLT 1 | |
| 54 | AFEF | LD #0xEF | |
| 55 | EEF0 | ST X | |
| 56 | 1100 | EI | Разрешаем прерывания после обработки инкремента. |
| 57 | CEF7 | JUMP -9 | Зацикливаемся. |

2. Листинг ассемблерного кода:

```
V0:    ORG    0x0
        WORD  $FUNC, 0x180
V1:    WORD  $LAND, 0x180

        ORG    0x10
LAND:   NOP
        PUSH           ; X & (BY-2) -> X
        IN      4
        AND     X
        ST      X
        NOP
        POP
        IRET

FUNC:   NOP
        PUSH           ; F(X) = -7 * X + 7 -> BY-3
        LD      X
        ASL
        ASL
        ADD     X
        ADD     X
        ADD     X
        NEG
        ADD     #7
        OUT     6
        NOP
        POP
        IRET

X:      ORG    0x046
        WORD  0

START:  DI           ; loading int vectors
        LD      #8
        OUT     7
        INC
        OUT     5
        EI
        CLA           ; X++ in loop
        ST      X
CYCLE:  DI
        LD      X
        INC
        CMP     #13
        BLT     REPEAT
        LD      #0xEF
REPEAT: ST      X
        EI
        JUMP    CYCLE
```

Методика проверки

1. Загрузить комплекс программ в память БЭВМ.
2. Изменить значение точки останова по адресам 010, 015, 018, 023 на HLT.
3. Запустить основную программу в автоматическом режиме с адресом 048.
4. Установить «Готовность ВУ-3».
5. Дождаться останова.
6. Записать содержимое АС в момент останова программы.
7. Продолжить выполнение подпрограммы.
8. Дождаться останова.
9. Записать содержимое DR ВУ-3 в момент останова программы.
10. Продолжить выполнение программы.
11. Ввести беззнаковое 8-битное логическое значение в ВУ-2.
12. Установить «Готовность ВУ-2».
13. Дождаться останова.
14. Записать содержимое АС в момент останова подпрограммы.
15. Продолжить выполнение подпрограммы.
16. Дождаться останова.
17. Записать содержимое АС в момент останова подпрограммы.

| АС до прерывания | Ожидаемое DR ВУ-3 | DR ВУ-3 после |
|------------------|-------------------|---------------|
| EF ~ -17 | 126 | 7E ~ 126 |
| 04 ~ 4 | -21 | EB ~ -21 |
| 09 ~ 9 | -56 | C8 ~ -56 |

| АС до прерывания | Значение DR ВУ-2 | Ожидаемое АС | АС после |
|------------------|------------------|--------------|----------|
| 06 | 00 | 00 | 00 |
| F2 | C5 | C0 | C0 |
| 0A | 1E | 0A | 0A |

Заключение

В ходе выполнения лабораторной работы я изучил организацию процесса прерывания программы в БЭВМ, исследовал порядок функционирования ЭВМ при обмене данными в режиме прерывания программы, в результате чего разработал и исследовал работу комплекса программ обмена данными в режиме прерывания программы по выданному преподавателем варианту.