Санкт-Петербургский Национальный Исследовательский  
Университет ИТМО

Факультет программной инженерии и компьютерной техники

**Лабораторная работа №6**

По “Основы профессиональной деятельности”

Вариант 14718

*Выполнила:*

Брель Мария Владимировна P3107

*Преподаватель:*

Вербовой Александр Александрович

Санкт-Петербург

2024

**Оглавление**

[Задание 3](#__RefHeading___Toc12835_2286687196)

[Основные этапы вычисления 4](#__RefHeading___Toc12837_2286687196)

[1.1 Программа на ассемблере 4](#__RefHeading___Toc12839_22866871961)

[1.2 Область представления 6](#__RefHeading___Toc12843_2286687196)

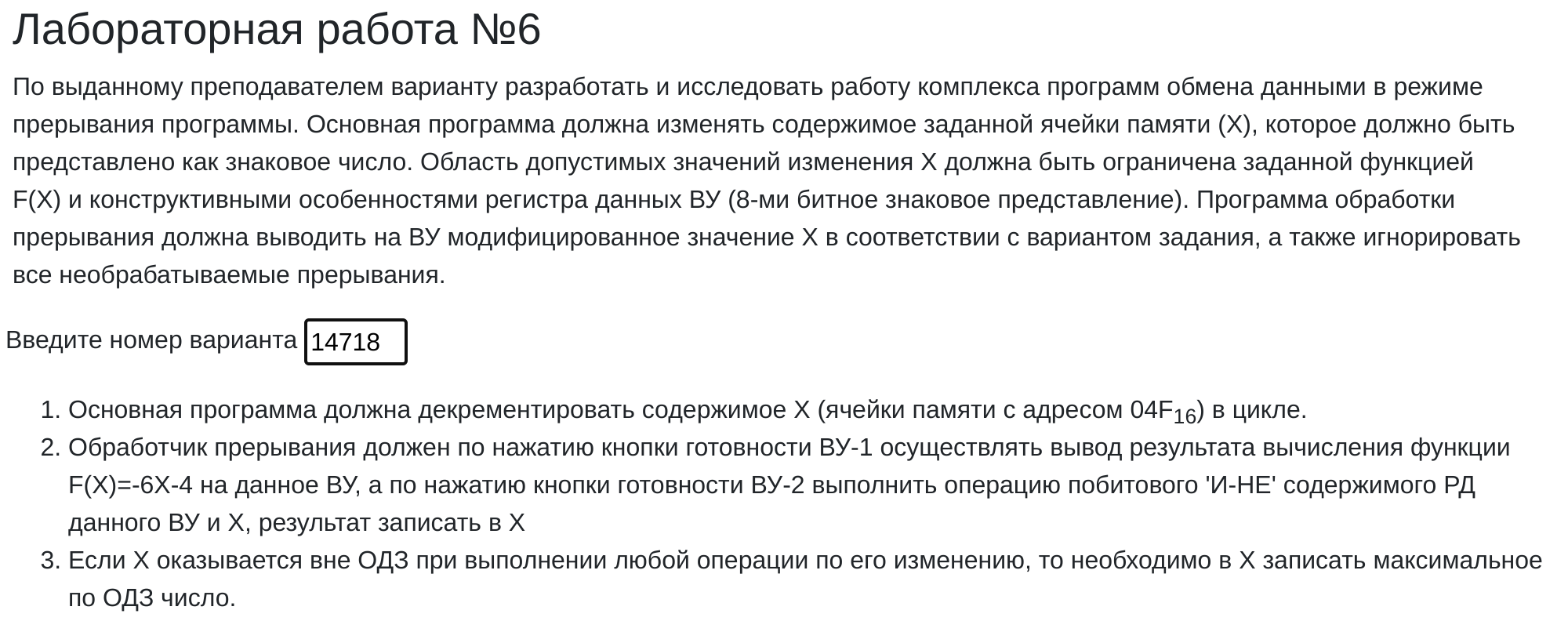
[1.3 Область допустимых значений 6](#__RefHeading___Toc12843_22866871961)

[1.4 Расположение данных в памяти 6](#__RefHeading___Toc12845_2286687196)

[1.5 Методика проверки программы 7](#__RefHeading___Toc12845_22866871961)

[Вывод 9](#__RefHeading___Toc12853_2286687196)

# Задание



# Основные этапы вычисления

## 1.1 Программа на ассемблере

|  |
| --- |
| ORG 0x0  V0: WORD $default, 0X180  V1: WORD $int1, 0X180  V2: WORD $int2, 0X180  V3: WORD $default, 0x180  V4: WORD $default, 0X180  V5: WORD $default, 0X180  V6: WORD $default, 0X180  V7: WORD $default, 0X180  ORG 0x04F  x: WORD ?  max: WORD 0x0015  min: WORD 0xFFEA  ; Обработка прерываний по умолчанию  ; Сбрасывает готовность со всех неиспользуемых ВУ и выходит из  прерывания  default: IRET  start: DI  CLA  OUT 0x1  OUT 0x7  OUT 0xB  OUT 0xD  OUT 0x11  OUT 0x15  OUT 0x19  OUT 0x1D  LD #0x9  OUT 0x3  LD #0xA  OUT 0x5  EI  ; Основная программа  main: DI  LD X  EI  DEC  CALL check  DI  ST X  EI  JUMP main  ; Обработка прерывания ВУ-1  int1: CALL check  PUSH  NOP ; Отладочная точка останова  ASL  ASL  ASL  NEG  SUB #0x4  OUT 0x2  NOP ; Отладочная точка останова  POP  IRET  ; Обработка прерывания ВУ-2  int2: CALL check  PUSH  NOP ; Отладочная точка останова  CLA  IN 0x4  SXTB  NOP ; Отладочная точка останова  AND (SP + 0)  NOT  NOP ; Отладочная точка останова  CALL check  ST (SP + 0)  NOP ; Отладочная точка останова  POP  IRET  check:  check\_min: CMP min  BGE check\_max  LD max  JUMP return  check\_max: CMP max  BLT return  LD max  return: RET |

## 1.2 Область представления

•X, min, max – знаковое 16-разрядное целое число

•DR (регистр данных) ВУ-1 и ВУ-2 – знаковое 8-разрядное целое число

## 1.3 Область допустимых значений

−128 ≤ −6X − 4 ≤ 127

−124 ≤ −6X ≤ 131

−22 ≤ X ≤ 21

–22 = 1111 1111 1110 1010 = FFEA

21 = 0000 0000 0001 0101 = 003D

Значит X ∈ [– 22; 21]

## 1.4 Расположение данных в памяти

Вектора прерываний: 0x000 – 0x00F

Переменные: 0x04F – 0x051

Основная программа: 0x062 – 0x06A

Подпрограмма обработки прерываний с ВУ-1: 0x06B – 0x076

Подпрограмма обработки прерываний с ВУ-2: 0x077 – 0x085

Подпрограмма обработки для проверки ОДЗ: 0x086 – 0x08E

## **1.5 Методика проверки программы**

***Проверка обработки прерываний:***

1. Загрузить комплекс программ в память БЭВМ
2. Во всех точках останова заменить NOP на HLT
3. Запустить БЭВМ в режиме РАБОТА
4. Ввести в регистр данных ВУ-2 тестовые данные
5. Установить готовность ВУ-2
6. Дождаться остановки программы
7. Запомнить текущее значени AC (считаем что это Х)
8. Нажать кнопку ПРОДОЛЖЕНИЕ
9. Дождаться остановки программы
10. Сверить значение в аккумуляторе со значением введенным на ВУ-2
11. Нажать кнопку ПРОДОЛЖЕНИЕ
12. Дождаться остановки программы
13. Записать значение из акумулятора, сравнив его с регистром данных ВУ-2
14. Нажать кнопку ПРОДОЛЖЕНИЕ
15. Дождаться остановки программы
16. Рассчитать ожидаемы результат вычисления выражения NOT((DR ВУ-2) AND X) и сравнить его с полученным в аккумуляторе
17. Нажать кнопку ПРОДОЛЖЕНИЕ
18. Дождаться остановки программы
19. Проверить корректность приведения вычисленного значения в ОДЗ
20. Нажать кнопку ПРОДОЛЖЕНИЕ
21. Установить готовность ВУ-1
22. Дожаться остановки программы
23. Записать значение переменной X из аккумулятора
24. Нажать кнопку ПРОДОЛЖЕНИЕ
25. Дождаться остановки программы
26. Рассчитать ожидаемый результат вычисления функции F(X) и сравнить его с полученным в аккумуляторе и на ВУ-1

***Проверка основной программы***

1. Загрузить комплекс программы в память БЭВМ
2. Запустить БЭВМ в режиме РАБОТА
3. Дождаться достижения крайних значений аккумулятора для проверки корректности приведения к ОДЗ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Прерывание в ВУ-2 | | | | |
| DR ВУ-2 | X до | X без одз | X после | X ожидаемое |
| 10(16) | 000A(10) | FFFF(-1) | FFFF(-1) | FFFF(-1) |
| 34(52) | FFFE(-2) | FFCB(-34) | 0015(21) | 0015(21) |
| 7F(127) | 0010(16) | FFEF(-10) | FFEF(-10) | FFEF(-10) |

|  |  |  |  |
| --- | --- | --- | --- |
| Прерывание в ВУ-1 | | | |
| X | F(X) ожидаемое | F(X) полученное | DR ВУ-1 |
| 000A(10) | FFAC(-64) | FFAC(-64) | AC(-64) |
| FFFC(-3) | 001C(28) | 001C(28) | 001C(28) |
| 0014(20) | FF5C(-92) | FF5C(-92) | 5C(-92) |

|  |  |  |
| --- | --- | --- |
| Основная программа | | |
| AC до | AC ожидаемая | AC после |
| FFEA(-22) | 0015(21) | 0015(21) |
| 0014(20) | 0013(19) | 0013(19) |
| 0000(0) | FFFF(-1) | FFFF(-1) |

# Вывод

В ходе работы я изучила организацию процесса прерывания программы и исследовала порядок функционирования БЭВМ при обмене данными с внешними устройствами в режиме прерывания программы