Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования «Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики»

Факультет программной инженерии и компьютерной техники

ОТЧЕТ

ПО ЛАБОРАТОРНОЙ РАБОТЕ 2

ИССЛЕДОВАНИЕ РАБОТЫ БЭВМ

ВАРИАНТ 1016

­

Студент: Пышкин Никита Сергеевич, P3113

Преподаватель: Блохина Елена Николаевна

Санкт Петербург 2023

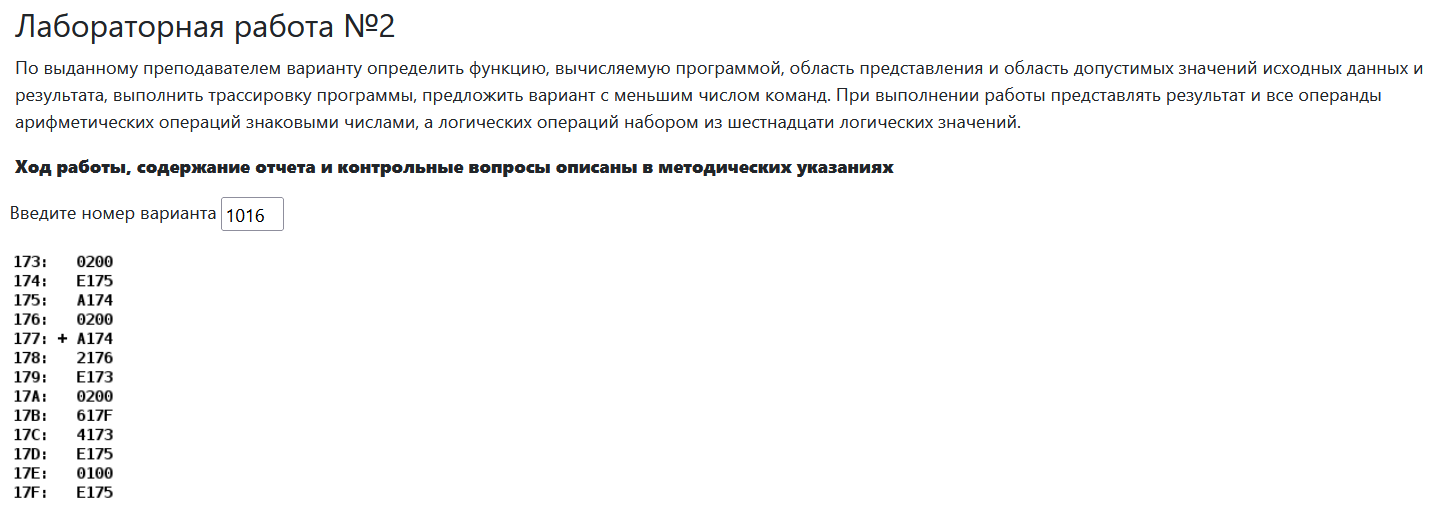
Содержание

[**Задание** 3](#_Toc145935753)

[**Выполнение работы** 6](#_Toc145935754)

[**Заключение** 11](#_Toc145935755)

# **Задание**



# **Выполнение работы**

## Текст исходной программы

|  |  |  |  |
| --- | --- | --- | --- |
| **Адрес** | **Код команды** | **Мнемоника** | **Комментарий** |
| 177 | A174 | LD 174 | Загрузка значения из ячейки B в аккумулятор:  174 -> AC |
| 178 | 2176 | AND 176 | Выполнить логическое “И” для содержимого ячейки памяти 176 и аккумулятора, результат записать в аккумулятор:  176 & AC -> AC |
| 179 | E173 | ST 173 | Сохранить содержимое аккумулятора в ячейку 173:  AC -> 173 |
| 17A | 0200 | CLA | Очистить содержимое аккумулятора:  0 -> AC |
| 17B | 617F | SUB 17F | Вычесть из аккумулятора содержимое ячейки 17F, результат записать в  аккумулятор:  AC – 17F -> AC |
| 17C | 4173 | ADD 173 | Добавить содержимое ячейки 173 к аккумулятору, результат записать в аккомулятор  173 + AC -> AC |
| 17D | E175 | ST 175 | Сохранить содержимое аккумулятора в ячейку 175:  AC -> 175 |
| 17E | 0100 | HLT | Остановка |

## Описание программы

**1) Назначение программы и реализуемая ею функция (формула):**

R = (-E) + (D & B)

**2) Описание и назначение исходных данных:**

Область представления:

* B, D – набор из 16 однобитовых значений, 0 B, D 216 – 1
* A, E, R – знаковое, 16-ти разрядное число, -215 A, E, R 215 – 1

Результат логической операции (D & B) трактуется как арифмитический операнд:

* (D & B) – знаковое, 16-ти разрядное число, -215 (D & B) 215 – 1

Допустимые значения:

* С R все просто: -215 R 215 – 1
* Теперь рассмотрим A - E:

Прежде всего учтем, что у нас 0 – E, а значит, E не может быть равно -215, следовательно -215 + 1 E 215 – 1

Случай 1, когда слагаемые имеют разные знаки, переполнение возникнуть не может:

A15 E15 = 1

-215 A 215 – 1, -215 + 1 E 215 – 1

Случай 2, когда слагаемые имеют одинаковый знак переполнение возможно, поэтому их нужно ограничить (хоть мы и теряем половину возможных значений A и E)

A15 E15 = 0

-214 A, E 214 – 1

* Отдельно рассмотрим случаи:

1) A принимает крайнее отрицательное значение:

A = -215

-215 + 1 E 0

2) A принимает крайнее положительное значение:

A = 215 – 1

0 E 215 – 1

3) E принимает крайнее отрицательное значение:

E = -215 + 1

-215 A 1

4) E принимает крайнее положительное значение:

E = 215 – 1

-1 A 215 – 1

**3) Расположение в памяти ЭВМ программы, исходных данных и результатов:**

По адресам 173-17F находится исходный код программы

По адресам 174, 176, 17F находятся переменные

По адресу 173 находится промежуточный результат

По адресу 175 находится итоговый результат

По адресам 177 и 17E находятся первая и последняя команда программы соответственно

Новые исходные данные для таблицы трассировки

E= -256 (1111 1111 0000 0000) 17F

B = -2 (1111 1111 1111 1110) 174

D = 34 (0000 0000 0010 0010) 176

## Таблица трассировки

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Выполняемая команда** | | **Содержимое регистров процессора после выполнения команды** | | | | | | | | **Ячейка, содержимое которой изменилось после выполнения команды** | |
| Адрес | Код | IP | CR | AR | DR | SP | BR | AC | NZVC | Адрес | Новый код |
| 177 | A174 | 178 | A174 | 174 | FFFE | 000 | 0177 | FFFE | 1000 |  |  |
| 178 | 2176 | 179 | 2176 | 176 | 0022 | 000 | 0178 | 0022 | 0000 |  |  |
| 179 | E173 | 17A | E173 | 173 | 0022 | 000 | 0179 | 0022 | 0000 | 173 | 0022 |
| 17A | 0200 | 17B | 0200 | 17A | 0200 | 000 | 017A | 0000 | 0100 |  |  |
| 17B | 617F | 17C | 617F | 17F | FF00 | 000 | 017B | 0100 | 0000 |  |  |
| 17C | 4173 | 17D | 4173 | 173 | 0022 | 000 | 017C | 0122 | 0000 |  |  |
| 17D | E175 | 17E | E175 | 175 | 0122 | 000 | 017D | 0122 | 0000 | 175 | 0122 |
| 17E | 0100 | 17F | 0100 | 17E | 0100 | 000 | 017E | 0122 | 0000 |  |  |

## Вариант программы с меньшим числом команд

R = (D & B) – E

|  |  |  |  |
| --- | --- | --- | --- |
| **Адрес** | **Код команды** | **Мнемоника** | **Комментарий** |
| 177 | + A174 | LD 174 | Загрузка значения из ячейки B в аккумулятор:  174 -> AC |
| 178 | 2176 | AND 176 | Выполнить логическое “И” для содержимого ячейки памяти 176 и аккумулятора, результат записать в аккумулятор:  176 & AC -> AC |
| 179 | 617F | SUB 17F | Вычесть из аккумулятора содержимое ячейки 173, результат записать в  аккумулятор:  AC – 17F -> AC |
| 17A | E175 | ST 175 | Сохранить содержимое аккумулятора в ячейку 175:  AC -> 175 |
| 17B | 0100 | HLT | Остановка |

# **Заключение**

В ходе выполнения данной лабораторной работы я изучил основные команды БЭВМ, научился работать с адресами, определять область представления переменных и результатов, составлять таблицу трассировки и переписывать исходный код программы с меньшим числом команд.