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 101422

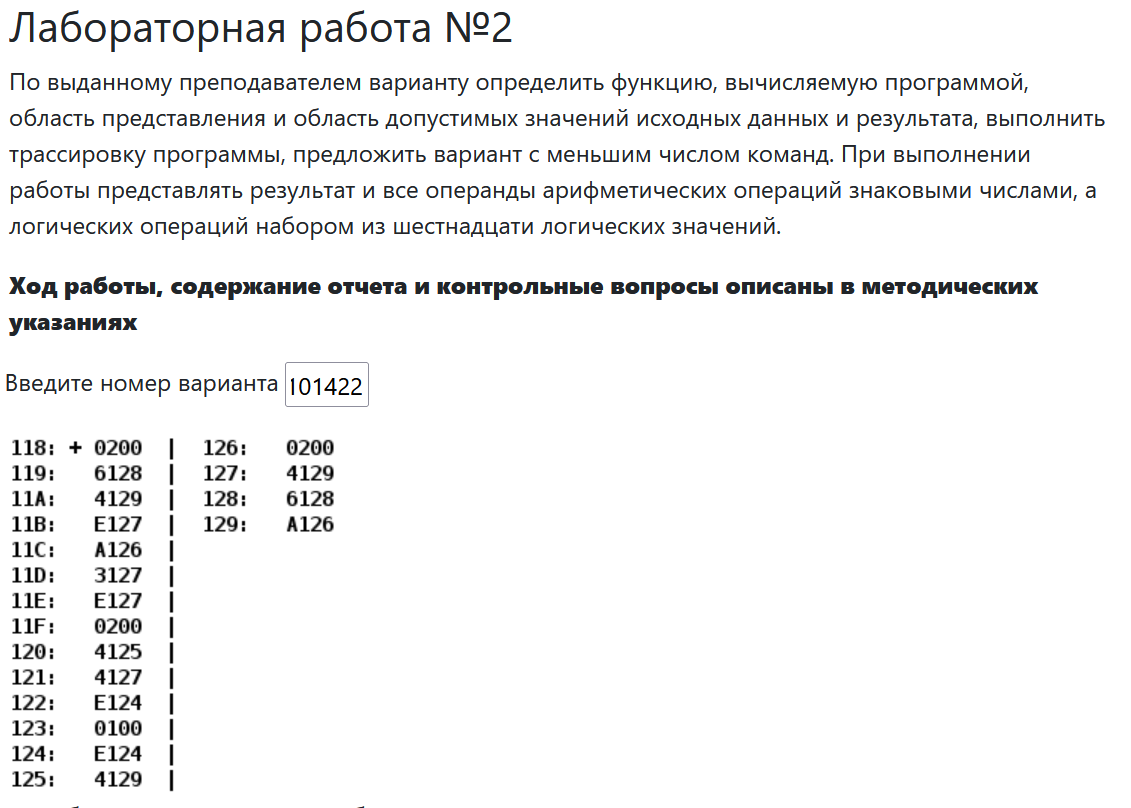
­

Студент: Плотников Алексей Алексеевич, P3113

Преподаватель: Блохина Елена Николаевна

Санкт Петербург 2023

# Задание



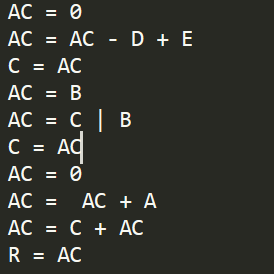
# **Выполнение работы**

## Текст исходной программы

|  |  |  |  |
| --- | --- | --- | --- |
| **Адрес** | **Код команды** | **Мнемоника** | **Комментарий** |
| 118 | 0200 | CLA | Очистить содержимое аккумулятора  0 -> AC |
| 119 | 6128 | SUB 128 | Вычесть из аккумулятора содержимое ячейки памяти 128  AC – (128) -> AC |
| 11A | 4129 | ADD 129 | Добавить содержимое ячейки памяти 129 к аккумулятору  (129) + AC -> AC |
| 11B | E127 | ST 127 | Сохранить содержимое аккумулятора в ячейку памяти 127  AC -> (127) |
| 11C | A126 | LD 126 | Загрузить содержимое ячейки памяти 126 в аккумулятор  (126) -> AC |
| 11D | 3127 | OR 127 | Выполнить над каждым разрядом аккумулятора и ячейки памяти 127 операцию логического сложения  (127) | AC -> AC |
| 11E | E127 | ST 127 | Сохранить содержимое аккумулятора в ячейку памяти 127  AC -> (127) |
| 11F | 0200 | CLA | Очистить содержимое аккумулятора  0 -> AC |
| 120 | 4125 | ADD 125 | Добавить содержимое ячейки памяти 125 к аккумулятору  (125) + AC -> AC |
| 121 | 4127 | ADD 127 | Добавить содержимое ячейки памяти 127 к аккумулятору  (127) + AC -> AC |
| 122 | E124 | ST 124 | Сохранить содержимое аккумулятора в ячейку памяти 124  AC -> (124) |
| 123 | 0100 | HLT | Остановка |

## Описание программы

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

R = ячейка памяти 124

A = ячейка памяти 125

B = ячейка памяти 126

C = ячейка памяти 127

D = ячейка памяти 128

E = ячейка памяти 129

**R = ((0-D+E) | B) + A**

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

Область представления:

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

Допустимые значения:

Рассмотрим R:

-215 R 215 – 1

Рассмотрим (0-D+E) ⬄ (E-D):

Учтем, что у нас 0 – D, значит, D не может быть равно -215, следовательно -215 + 1 D 215 – 1

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

E15 D15 = 1

-215 E 215 – 1

-215 + 1 D 215 – 1

Случай 2, когда слагаемые имеют одинаковый знак переполнение возможно, ограничим:

E15 D15 = 0

-214 E 214 – 1

-214 D 214 – 1

Рассмотрим случаи:

* E принимает крайнее отрицательное значение:

E= -215

-215 + 1 D 0

* E принимает крайнее положительное значение:

E = 215 – 1

0 D 215 – 1

* D принимает крайнее отрицательное значение:

D = -215 + 1

-215 E 1

* D принимает крайнее положительное значение:

D = 215 – 1

-1 E 215 – 1

Рассмотрим **((0-D+E) | B) + A:**

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

B15 A15 = 1

0 B 216 – 1

-215 A 215 – 1

Случай 2, когда слагаемые имеют одинаковый знак переполнение возможно, ограничим:

B15 A15 = 0

0 B 214 – 1 , B15 = A15  или (0 B 216 – 1) & (B14 = 0)

-214 A 214 – 1

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

Программа занимает адресса 118-129

По адресам 118-123 находится исходный код программы

По адресам 124-129 находятся переменные

R = ячейка памяти 124

A = ячейка памяти 125

B = ячейка памяти 126

C = ячейка памяти 127

D = ячейка памяти 128

E = ячейка памяти 129

По адресу 124 хранится итоговый результат

По адресу 127 хранится промежуточный результат

**4) Адреса первой и последней команды выполняемой команды**

По адресам 118 и 123 находятся первая и последняя команда программы соответственно

Еще раз проверяем ОДЗ

Делаем ТТ с новыми данными в 10-формате

А=-90 A = FFA6

Б=2048 Б = 0800

Е=2089 E = 0829

Д=-786 Д = FCEE

## Новая 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 |
| XXX | XXXX | XXXX | XXXX | XXXX | XXXX | XXXX | XXXX | XXXX | XXXX | XXX | XXXX |
| 118 | 0200 | 0119 | 0200 | 0118 | 0200 | 0000 | 0118 | 0000 | 0100 |  |  |
| 119 | 6128 | 011A | 6128 | 0128 | FCEE | 0000 | 0119 | 0312 | 0000 |  |  |
| 11A | 4129 | 011B | 4129 | 0129 | 0829 | 0000 | 011A | 0B3B | 0000 |  |  |
| 11B | E127 | 011C | E127 | 0127 | 0B3B | 0000 | 011B | 0B3B | 0000 | 127 | 0B3B |
| 11C | A126 | 011D | A126 | 0126 | 0800 | 0000 | 011C | 0800 | 0000 |  |  |
| 11D | 3127 | 011E | 3127 | 0127 | 0B3B | 0000 | F4C4 | 0B3B | 0000 |  |  |
| 11E | E127 | 011F | E127 | 0127 | 0B3B | 0000 | 011E | 0B3B | 0000 | 127 | 0B3B |
| 11F | 0200 | 0120 | 0200 | 011F | 0200 | 0000 | 011F | 0000 | 0100 |  |  |
| 120 | 4125 | 0121 | 4125 | 0125 | FFA6 | 0000 | 0120 | FFA6 | 1000 |  |  |
| 121 | 4127 | 0122 | 4127 | 0127 | 0B3B | 0000 | 0121 | 0AE1 | 0001 |  |  |
| 122 | E124 | 0123 | E124 | 0124 | 0AE1 | 0000 | 0122 | 0AE1 | 0001 | 124 | 0AE1 |
| 123 | 0100 | 0124 | 0100 | 0123 | 0100 | 0000 | 0123 | 0AE1 | 0001 |  |  |

## 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 |
| XXX | XXXX | XXXX | XXXX | XXXX | XXXX | XXXX | XXXX | XXXX | XXXX | XXX | XXXX |
| 118 | 0200 | 0119 | 0200 | 0118 | 200 | 0000 | 0118 | 0000 | 0100 |  |  |
| 119 | 6128 | 011A | 6128 | 0128 | 6128 | 0000 | 0119 | 9ED8 | 1000 |  |  |
| 11A | 4129 | 011B | 4129 | 0129 | A126 | 0000 | 011A | 3FFE | 0011 |  |  |
| 11B | E127 | 011C | E127 | 0127 | 3FFE | 0000 | 011B | 3FFE | 0011 | 127 | 3FFE |
| 11C | A126 | 011D | A126 | 0126 | 200 | 0000 | 011C | 0200 | 0001 |  |  |
| 11D | 3127 | 011E | 3127 | 0127 | 3FFE | 0000 | C001 | 3FFE | 0001 |  |  |
| 11E | E127 | 011F | E127 | 0127 | 3FFE | 0000 | 011E | 3FFE | 0001 | 127 | 3FFE |
| 11F | 0200 | 0120 | 0200 | 011F | 200 | 0000 | 011F | 0000 | 0101 |  |  |
| 120 | 4125 | 0121 | 4125 | 0125 | 4129 | 0000 | 0120 | 4129 | 0000 |  |  |
| 121 | 4127 | 0122 | 4127 | 0127 | 3FFE | 0000 | 0121 | 8127 | 1010 |  |  |
| 122 | E124 | 0123 | E124 | 0124 | 8127 | 0000 | 0122 | 8127 | 1010 | 124 | 8127 |
| 123 | 0100 | 0124 | 0100 | 0123 | 100 | 0000 | 0123 | 8127 | 1010 |  |  |

## Вариант программы с меньшим числом команд

**R = (B | (E-D)) + A**

|  |  |  |  |
| --- | --- | --- | --- |
| **Адрес** | **Код команды** | **Мнемоника** | **Комментарий** |
| 118 | A129 | LD 129 | Загрузить содержимое ячейки памяти 129 в аккумулятор  (129) -> AC |
| 119 | 6128 | SUB 128 | Вычесть из аккумулятора содержимое ячейки памяти 128  AC – (128) -> AC |
| 11A | 3126 | OR 126 | Выполнить над каждым разрядом аккумулятора и ячейки памяти 126 операцию логического сложения  (126) | AC -> AC |
| 11B | 4125 | ADD 125 | Добавить содержимое ячейки памяти 125 к аккумулятору  (125) + AC -> AC |
| 11C | E124 | ST 124 | Сохранить содержимое аккумулятора в ячейку памяти 124  AC -> (124) |
| 11D | 0100 | HLT | Остановка |

# **Заключение**

В ходе выполнения лабораторной работы изучил ряд команд БЭВМ, научился работать с командами с абсолютной адресацией, определять область представления переменных и результатов, составлять таблицу трассировки и переписывать исходный код программы с меньшим числом команд.