Федеральное государственное автономное образовательное учреждение высшего

образования «Национальный исследовательский университет ИТМО»

Факультет программной инженерии и компьютерной техники

**Отчет**

**По лабораторной работе №4**

**«Исследование работы БЭВМ»**

По дисциплине «Основы профессиональной деятельности»

Вариант 97585

Выполнил: Воронов Г. А., группа P3116

Преподаватель: Остапенко Ольга Денисовна

Санкт-Петербург

2024

Оглавление

[Текст задания. 3](#_gjdgxs)

[Текст программы. 4](#_30j0zll)

[Описание программы. 5](#_1fob9te)

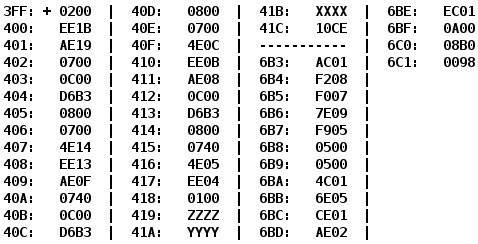
[Таблица трассировки 6](#_3znysh7)

[Вариант с меньшим числом команд. 7](#_2et92p0)

[Вывод. 8](#_tyjcwt)

# **Текст задания.**

По выданному преподавателем варианту восстановить текст заданного варианта программы и подпрограммы (программного комплекса), определить предназначение и составить его описание, определить область представления и область допустимых значений исходных данных и результата, выполнить трассировку программного комплекса.



# **Текст программы.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Адрес** | **Код команды** | **Мнемоника** | **Комментарий** | **Описание** |
| 3FF | 0200 | CLA | Очистка аккумулятора |  |
| 400 | EE1B | ST IP+0x1B | Запись AC в ячейку 0x41c | R := 0 |
| 401 | AE19 | LD IP + 0x19 | Загрузка 0x41b в AC | AC := X |
| 402 | 0700 | INC | AC := AC + 1 | X + 1 |
| 403 | 0C00 | PUSH | Положить на стек значение AC | Передача аргумента подпрограмме |
| 404 | D6B3 | CALL 0x6B3 | Вызов подпрограммы по адресу 0x6B3 | F(X + 1) |
| 405 | 0800 | POP | Взять из стека значение и положить в AC | Получение результата подпрограммы |
| 406 | 0700 | INC | AC : = AC + 1 | F(X + 1) + 1 |
| 407 | 4E14 | ADD IP + 0x14 | AC := AC + [0x41c] |  |
| 408 | EE13 | ST IP + 0x13 | [0x41c] := AC | R := R + F(X + 1) + 1 |
| 409 | AE0F | LD IP + 0xF | AC := [0x419] | AC := Z |
| 40A | 0740 | DEC | AC : = AC - 1 | Z - 1 |
| 40B | 0C00 | PUSH | Положить на стек значение AC | Передача аргумента подпрограмме |
| 40C | D6B3 | CALL 0x6B3 | Вызов подпрограммы | F(Z - 1) |
| 40D | 0800 | POP | Взять из стека значение и положить в AC | Получение результата подпрограммы |
| 40E | 0700 | INC | AC : = AC + 1 | F(Z - 1) + 1 |
| 40F | 4E0C | ADD IP + 0xC | AC := AC + [0x41c] |  |
| 410 | EE0B | ST IP + 0xB | [0x41c] := AC | R := R + F(Z - 1) + 1 |
| 411 | AE08 | LD IP + 0x8 | AC := [0x41a] | AC := Y |
| 412 | 0C00 | PUSH | Положить на стек значение AC | Передача аргумента подпрограмме |
| 413 | D6B3 | CALL 0x6B3 | Вызов подпрограммы | F(Y) |
| 414 | 0800 | POP | Взять из стека значение и положить в AC | Получение результата подпрограммы |
| 415 | 0740 | DEC | AC : = AC - 1 | F(Y) - 1 |
| 416 | 4E05 | ADD IP + 5 | AC : = AC + [0x41c] |  |
| 417 | EE04 | ST IP + 4 | [0x41c] := AC | R := R + F(Y) - 1 |
| 418 | 0100 | HLT | Завершение программы |  |
| 419 |  | Z | Переменная Z |  |
| 41A |  | Y | Переменная Y |  |
| 41B |  | X | Переменная X |  |
| 41C |  | R | Переменная R | R = F(X + 1) + 1 + F(Z - 1) + 1 + F(Y) - 1 |
| … | **Подпрограмма** | | | |
| 6B3 | AC01 | LD SP + 1 | AC := [SP + 1] | AC := arg |
| 6B4 | F208 | BMI IP + 8 | IF AC < 0:  IP := 0x6BD | Если аргумент отрицательный, то вернуть A |
| 6B5 | F007 | BEQ IP + 7 | IF AC == 0:  IP := 0x6BD | Если аргумент == 0, то вернуть A |
| 6B6 | 7E09 | CMP IP + 9 | (AC - [0x6c0]) -> NZVC | Если аргумент больше A, то вернуть A |
| 6B7 | F905 | BGE IP + 5 | IF AC > [0x6c0]:  IP := 0x6BD |
| 6B8 | 0500 | ASL | AC := 2\*AC | AC := 4\*arg |
| 6B9 | 0500 | ASL | AC := 2\*AC |
| 6BA | 4C01 | ADD SP + 1 | AC := AC + [SP + 1] | AC := AC + arg = 5\*arg |
| 6BB | 6E05 | SUB IP + 5 | AC := AC - [0x6c1] | AC := AC - [0x6c1] |
| 6BC | CE01 | JUMP IP + 1 | IP := 0x6BE |  |
| 6BD | AE02 | LD IP + 2 | AC := [0x6C0] |  |
| 6BE | EC01 | ST SP + 1 | [SP + 1] := AC | Вернуть результат подпрограммы.  **5\*arg - B**  или  **A** |
| 6BF | 0A00 | RET | Возврат из подпрограммы |  |
| 6C0 | 08B0 | A | Константа A = 0x8b0 = 2224 | |
| 6C1 | 0098 | B | Константа B = 0x98 = 152 | |

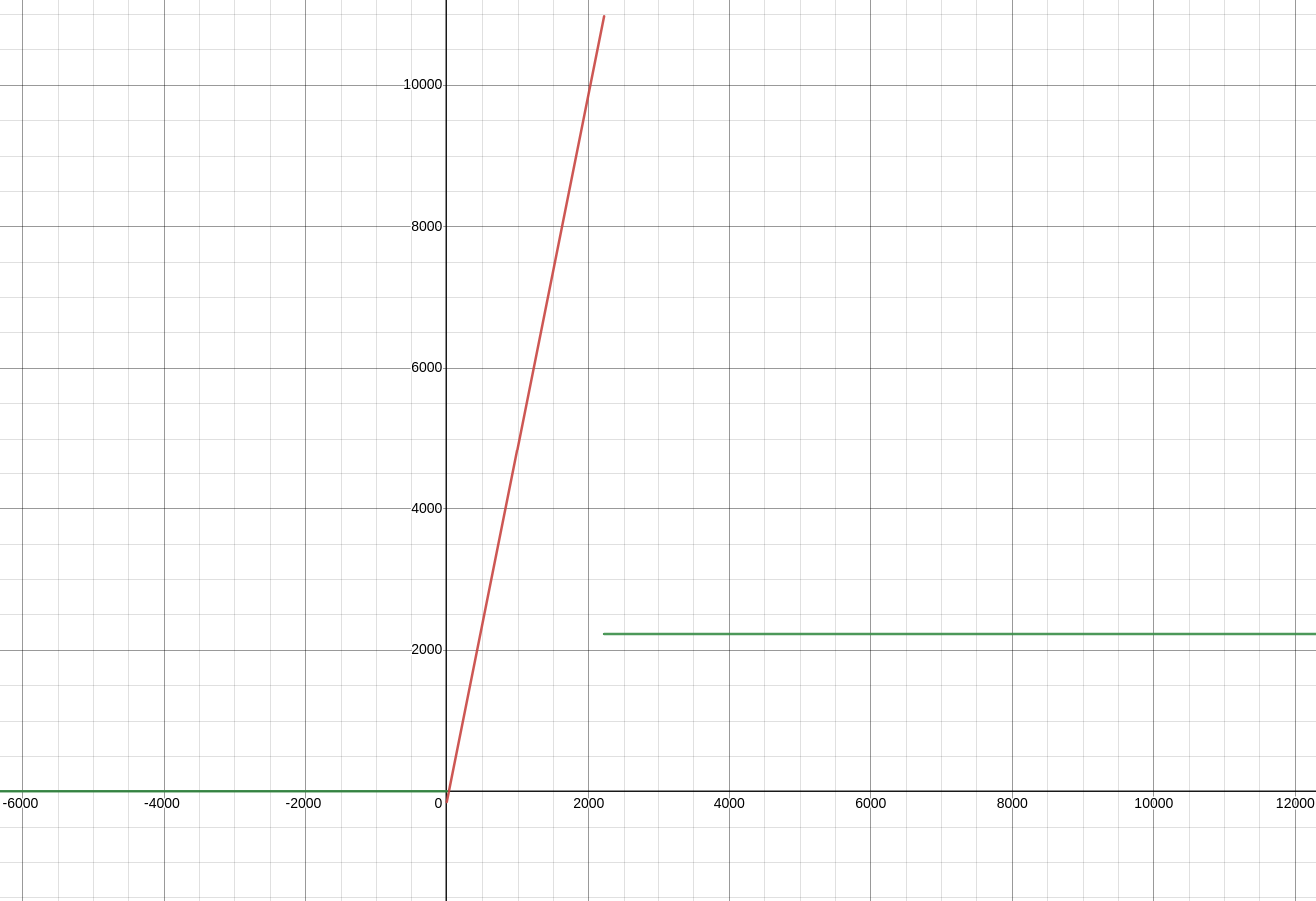
# **Описание программы.**

Назначение программы: нахождение значения выражения

*R = F(X + 1) + 1 + F(Z - 1) + 1 + F(Y) - 1*

*R = F(X + 1) + F(Z - 1) + F(Y) + 1*

**График:**

****

**Область представления:**

* X, Y, Z, R, A, B — целые знаковые числа

**Область допустимых значений:**

A = 2224

B = 152

Рассмотрим ОДЗ функции F(x):

* Если , то F(x) = 2224 и, соответственно, R = 2224 + 2224 + 2224 + 1. Переполнения не возникнет.
* Если , то F(x) = 5\*x — 152.
  + Максимальное значение: Fmax = F(2224) = 5\*2224 - 152 = 10968
  + Минимальное значение: Fmin = F(1) = 5\*1 - 152 = -147

Может возникнуть переполнение в случае:

R = F(2224) + F(2224) + F(2224) + 1 = 10968\*3 + 1 = 32905 >= 215

Тогда:

-215 <= *R < 215*

*-215 <= F(X + 1) + F(Z - 1) + F(Y) + 1 < 215*

-10922 = - *<= F(X + 1), F(Z - 1), F(Y) < =10922*

*- <=X+1, Z - 1, Y <*

*-3691 <= X+1, Z-1, Y <= 3691*

***-3692 <= X <= 3690***

***-3690 <= Z <= 3692***

***-3691 <= Y <= 3691***

или

F(Y) < 215 - F(Z - 1) - F(X + 1) - 1

F(Y) < 215 - 10968\*2 - 1

F(Y) < 10831, F(Z - 1) <= 10968, F(X + 1) <= 10968

0 < Y <= 2196, 0 < Z - 1, X + 1 <= 2224

**0 < Y <= 2196**

**1 < Z <= 2225**

**-1 < X <= 2223**

или

0 < Z - 1 <= 2196, 0 < Y, X + 1 <= 2224

**1 < Z <= 2197**

**0 < Y <= 2224**

**-1 < X <= 2223**

или

0 < X+ 1 <= 2196, 0 < Y, Z - 1 <= 2224

**-1 < Z <= 2195**

**0 < Y <= 2224**

**1 < Z <= 2225**

**Расположение в памяти БЭВМ**

**Основная программа:**

* [3FF; 418] — инструкции
* [419; 41B] — исходные данные
* 41C — результат

**Подпрограмма:**

* [6B3; 6BF] — инструкции
* [6С0; 6C1] — константы

**Адреса первой и последней выполняемой инструкции программы:**

* 3FF и 418 — для основной программы
* 6B3 и 6BF — для подпрограммы

# **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** | **PS** | **NZVC** | **Адрес** | **Новое значени** |
|  |  |  |  |  |  |  |  |  |  |  |  |  |

# **Вывод.**

Во время выполнения лабораторной работы я научился работать в БЭВМ с массивами, ветвлением и циклами. Я изучил прямую и косвенную адресацию и цикл выполнения таких команд, как LOOP и JUMP.