**Министерство науки и высшего образования Российской Федерации**

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧЕРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»**

**Факультет ПИиКТ**

**Дисциплина: Основы профессиональной деятельности**

**Лабораторная работа №4**

**Выполнение комплекса**

**программ**

**Вариант 2824**

Выполнил: Михайлов Петр Сергеевич

Группа: Р3111

Преподаватель: Остапенко Ольга Денисовна

Санкт-Петербург 2025г.

Содержание

[Задание 3](#_Toc191537569)

[Определение функции, вычисляемой программой 4](#_Toc191537570)

[1. Текст исходный программы 4](#_Toc191537571)

[2. Описание программы 5](#_Toc191537572)

[3. Расположение в БЭВМ программы, исходных данных и результатов 5](#_Toc191537573)

[4. Область представления 5](#_Toc191537574)

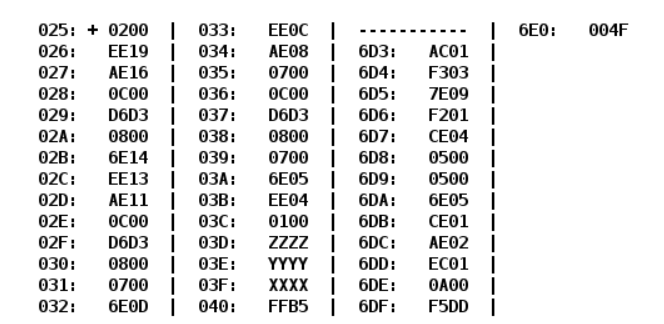
[5. Область допустимых значений 6](#_Toc191537575)

[Трассировка программы 7](#_Toc191537576)

[Заключение 8](#_Toc191537577)

# Задание

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.



# Определение функции, вычисляемой программой

1. Текст исходный программы

|  |  |  |  |
| --- | --- | --- | --- |
| **Адрес** | **Код команды** | **Мнемоника** | **Комментарии** |
| 025 | 0200 | CLA | Очистить аккумулятор:  0 ⇒ AC |
| 026 | EE19 | ST IP+19 | Сохранить содержимое аккумулятора в ячейку памяти **026** + 1 + 19 = **040**:  AC ⇒ (**040**) |
| 027 | AE16 | LD IP+16 | Загрузка в аккумулятор ячейки **027** + 1 + 16 = **3E**:  **(3E)** = Y ⇒ AC |
| 028 | 0C00 | PUSH | Вызов функции: AC ⇒ - (SP) |
| 029 | D6D3 | CALL 6D3 | F(Y): SP – 1 ⇒ SP, IP ⇒ (SP), 6D3 ⇒ IP |
| 02A | 0800 | POP | Загрузка результата в аккумулятор: (SP) + ⇒ AC |
| 02B | 6E14 | SUB IP+14 | Вычитание **02B** + 1 + 14 = **040** из аккумулятора:  AC – **(040)** = R ⇒ AC |
| 02C | EE13 | ST IP+13 | Сохранить содержимое аккумулятора в ячейку памяти **02C** + 1 + 13 = **040**:  AC ⇒ (**040**) = R |
| 02D | AE11 | LD IP+11 | Загрузка в аккумулятор ячейки **02D** + 1 + 11 = **3F**:  **(3F)** = X ⇒ AC |
| 02E | 0C00 | PUSH | Вызов функции: AC ⇒ - (SP) |
| 02F | D6D3 | CALL 6D3 | F(X): SP – 1 ⇒ SP, IP ⇒ (SP), 6D3 ⇒ IP |
| 030 | 0800 | POP | Загрузка результата в аккумулятор: (SP) + ⇒ AC |
| 031 | 0700 | INC | Инкремент аккумулятора: AC + 1 ⇒ AC |
| 032 | 6E0D | SUB IP+0D | Вычитание **032** + 1 + 0D = **040** из аккумулятора:  AC – **(040)** = R ⇒ AC |
| 033 | EE0C | ST IP+0C | Сохранить содержимое аккумулятора в ячейку памяти **033** + 1 + 0C = **040**:  AC ⇒ (**040**) = R |
| 034 | AE08 | LD IP+08 | Загрузка в аккумулятор ячейки **034** + 1 + 08 = **3D**:  **(3D)** = Z ⇒ AC |
| 035 | 0700 | INC | Инкремент аккумулятора: AC + 1 ⇒ AC |
| 036 | 0C00 | PUSH | Вызов функции: AC ⇒ - (SP) |
| 037 | D6D3 | CALL 6D3 | F(Z+1): SP – 1 ⇒ SP, IP ⇒ (SP), 6D3 ⇒ IP |
| 038 | 0800 | POP | Загрузка результата в аккумулятор: (SP) + ⇒ AC |
| 039 | 0700 | INC | Инкремент аккумулятора: AC + 1 ⇒ AC |
| 03A | 6E05 | SUB IP+05 | Вычитание **03A** + 1 + 05 = **040** из аккумулятора:  AC – **(040)** = R ⇒ AC |
| 03B | EE04 | ST IP+04 | Сохранить содержимое аккумулятора в ячейку памяти **03B** + 1 + 04 = **040**:  AC ⇒ (**040**) = R |
| 03C | 0100 | HLT | Останов |
| 03D | ZZZZ | - | Значение Z |
| 03E | YYYY | - | Значение Y |
| 03F | XXXX | - | Значение X |
| 040 | FFB5 | - | Результат |

Подпрограмма:

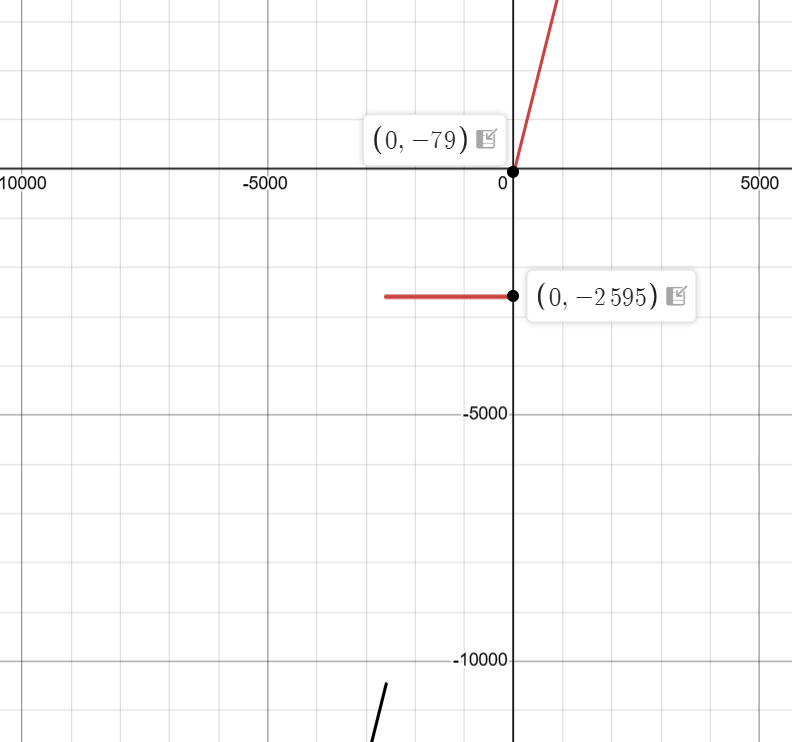
|  |  |  |  |
| --- | --- | --- | --- |
| **Адрес** | **Код команды** | **Мнемоника** | **Комментарии** |
| 6D3 | AC01 | LD SP+1 | Загрузка аргумента |
| 6D4 | F303 | BPL IP+3 | Переход на **6D4** + 1 +3 = **6D8**, если плюс  N == 0 |
| 6D5 | 7E09 | CMP IP+9 | Установка флага AC – {**6D5** + 1 + 9 = **6DF** = P} |
| 6D6 | F201 | BMI IP+1 | Переход на **6D6** + 1 +1 = **6D8**, если минус, то есть  AC – P <= 0 или P >= AC  N == 1 |
| 6D7 | CE04 | JUMP IP+4 | Прыжок на **6D7** + 1 + 4 = **6DC** |
| 6D8 | 0500 | ASL | Умножение на 4 |
| 6D9 | 0500 | ASL |
| 6DA | 6E05 | SUB IP+5 | Вычитание **6DA** + 1 + 5 = **6E0** = Q |
| 6DB | CE01 | JUMP IP+1 | Прыжок на **6DB** + 1 + 1 = **6DD** |
| 6DC | AE02 | LD IP+2 | Загрузка **6DC** + 1 + 2 = **6DF** = P |
| 6DD | EC01 | ST SP+1 | Сохранение результата |
| 6DE | 0A00 | RET | Возврат |
| 6DF | F5DD | F5DD | Константа P = -2595 |
| 6E0 | 004F | 004F | Константа Q = 79 |

2. Описание программы

Вычислим значение программы (формула):

Исходная функция имеет вид:

График функции:



3. Область представления

X, Y, Z, R, P, Q – целые знаковые шестнадцатеричные числа

4. Область допустимых значений

P = F5DD16 = -259510

Q = 004F16 = 7910

Для того чтобы определить ОДЗ, проанализируем данную функцию. При значении аргумента функции в промежутке , функция вернет значение выражения P. При использовании любого значения из заданного промежутка в функции не возникает переполнения.

При оставшихся значениях аргумента функция вернет выражение , что означает, что функция не переполняется на промежутке , а в других случаях будет переполнение.

Так как основная программа вычисляет следующее выражение:

,

то минимально мы можем получить: ,

а максимально .

В обоих случаях переполнение возможно. Значит, крайние значения нужно поделить еще на три, так как идет сложение результатов трех программ.

В функцию, как аргументы, мы передаем значения Z+1, X, Y. Значит, ОДЗ:

Если P и Q можно менять, то:

1. При :
2. При

# Трассировка программы

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Выполняемая команда | | Содержимое регистров процессора после выполнения команды | | | | | | | | | Ячейка, содержимое которой изменилось после выполнения команды | |
| Адрес | Код команды | IP | CR | AR | DR | SP | BR | AC | PS | NZVC | Адрес | Новый код |
| 025 | 0200 | 026 | 0200 | 025 | 0200 | 000 | 0025 | 0000 | 004 | 0100 |  |  |
| 026 | EE19 | 027 | EE19 | 040 | 0000 | 000 | 0019 | 0000 | 004 | 0100 | 040 | 0000 |
| 027 | AE16 | 028 | AE16 | 03E | 0002 | 000 | 0016 | 0002 | 000 | 0000 |  |  |
| 028 | 0C00 | 029 | 0C00 | 7FF | 0002 | 7FF | 0028 | 0002 | 000 | 0000 | 7FF | 0002 |
| 029 | D6ED | 6ED | D6ED | 7FE | 002A | 7FE | D6ED | 0002 | 000 | 0000 | 7FE | 002A |
| 6ED | AC01 | 6EE | AC01 | 7FF | 0002 | 7FE | 0001 | 0002 | 000 | 0000 |  |  |
| 6EE | F303 | 6F2 | F303 | 6EE | F303 | 7FE | 0003 | 0002 | 000 | 0000 |  |  |
| 6F2 | 0500 | 6F3 | 0500 | 6F2 | 0002 | 7FE | 06F2 | 0004 | 000 | 0000 |  |  |
| 6F3 | 0500 | 6F4 | 0500 | 6F3 | 0004 | 7FE | 06F3 | 0008 | 000 | 0000 |  |  |
| 6F4 | 6E05 | 6F5 | 6E05 | 6FA | 004F | 7FE | 0005 | FFB9 | 008 | 1000 |  |  |
| 6F5 | CE01 | 6F7 | CE01 | 6F5 | 06F7 | 7FE | 0001 | FFB9 | 008 | 1000 |  |  |
| 6F7 | EC01 | 6F8 | EC01 | 7FF | FFB9 | 7FE | 0001 | FFB9 | 008 | 1000 | 7FF | FFB9 |
| 6F8 | 0A00 | 02A | 0A00 | 7FE | 002A | 7FF | 06F8 | FFB9 | 008 | 1000 |  |  |
| 02A | 0800 | 02B | 0800 | 7FF | FFB9 | 000 | 002A | FFB9 | 008 | 1000 |  |  |
| 02B | 6E14 | 02C | 6E14 | 040 | 0000 | 000 | 0014 | FFB9 | 009 | 1001 |  |  |
| 02C | EE13 | 02D | EE13 | 040 | FFB9 | 000 | 0013 | FFB9 | 009 | 1001 | 040 | FFB9 |
| 02D | AE11 | 02E | AE11 | 03F | 0003 | 000 | 0011 | 0003 | 001 | 0001 |  |  |
| 02E | 0C00 | 02F | 0C00 | 7FF | 0003 | 7FF | 002E | 0003 | 001 | 0001 | 7FF | 0003 |
| 02F | D6ED | 6ED | D6ED | 7FE | 0030 | 7FE | D6ED | 0003 | 001 | 0001 | 7FE | 0030 |
| 6ED | AC01 | 6EE | AC01 | 7FF | 0003 | 7FE | 0001 | 0003 | 001 | 0001 |  |  |
| 6EE | F303 | 6F2 | F303 | 6EE | F303 | 7FE | 0003 | 0003 | 001 | 0001 |  |  |
| 6F2 | 0500 | 6F3 | 0500 | 6F2 | 0003 | 7FE | 06F2 | 0006 | 000 | 0000 |  |  |
| 6F3 | 0500 | 6F4 | 0500 | 6F3 | 0006 | 7FE | 06F3 | 000C | 000 | 0000 |  |  |
| 6F4 | 6E05 | 6F5 | 6E05 | 6FA | 004F | 7FE | 0005 | FFBD | 008 | 1000 |  |  |
| 6F5 | CE01 | 6F7 | CE01 | 6F5 | 06F7 | 7FE | 0001 | FFBD | 008 | 1000 |  |  |
| 6F7 | EC01 | 6F8 | EC01 | 7FF | FFBD | 7FE | 0001 | FFBD | 008 | 1000 | 7FF | FFBD |
| 6F8 | 0A00 | 030 | 0A00 | 7FE | 0030 | 7FF | 06F8 | FFBD | 008 | 1000 |  |  |
| 030 | 0800 | 031 | 0800 | 7FF | FFBD | 000 | 0030 | FFBD | 008 | 1000 |  |  |
| 031 | 0700 | 032 | 0700 | 031 | 0700 | 000 | 0031 | FFBE | 008 | 1000 |  |  |
| 032 | 6E0D | 033 | 6E0D | 040 | FFB9 | 000 | 000D | 0005 | 001 | 0001 |  |  |
| 033 | EE0C | 034 | EE0C | 040 | 0005 | 000 | 000C | 0005 | 001 | 0001 | 040 | 0005 |
| 034 | AE08 | 035 | AE08 | 03D | 0001 | 000 | 0008 | 0001 | 001 | 0001 |  |  |
| 035 | 0700 | 036 | 0700 | 035 | 0700 | 000 | 0035 | 0002 | 000 | 0000 |  |  |
| 036 | 0C00 | 037 | 0C00 | 7FF | 0002 | 7FF | 0036 | 0002 | 000 | 0000 | 7FF | 0002 |
| 037 | D6ED | 6ED | D6ED | 7FE | 0038 | 7FE | D6ED | 0002 | 000 | 0000 | 7FE | 0038 |
| 6ED | AC01 | 6EE | AC01 | 7FF | 0002 | 7FE | 0001 | 0002 | 000 | 0000 |  |  |
| 6EE | F303 | 6F2 | F303 | 6EE | F303 | 7FE | 0003 | 0002 | 000 | 0000 |  |  |
| 6F2 | 0500 | 6F3 | 0500 | 6F2 | 0002 | 7FE | 06F2 | 0004 | 000 | 0000 |  |  |
| 6F3 | 0500 | 6F4 | 0500 | 6F3 | 0004 | 7FE | 06F3 | 0008 | 000 | 0000 |  |  |
| 6F4 | 6E05 | 6F5 | 6E05 | 6FA | 004F | 7FE | 0005 | FFB9 | 008 | 1000 |  |  |
| 6F5 | CE01 | 6F7 | CE01 | 6F5 | 06F7 | 7FE | 0001 | FFB9 | 008 | 1000 |  |  |
| 6F7 | EC01 | 6F8 | EC01 | 7FF | FFB9 | 7FE | 0001 | FFB9 | 008 | 1000 | 7FF | FFB9 |
| 6F8 | 0A00 | 038 | 0A00 | 7FE | 0038 | 7FF | 06F8 | FFB9 | 008 | 1000 |  |  |
| 038 | 0800 | 039 | 0800 | 7FF | FFB9 | 000 | 0038 | FFB9 | 008 | 1000 |  |  |
| 039 | 0700 | 03A | 0700 | 039 | 0700 | 000 | 0039 | FFBA | 008 | 1000 |  |  |
| 03A | 6E05 | 03B | 6E05 | 040 | 0005 | 000 | 0005 | FFB5 | 009 | 1001 |  |  |
| 03B | EE04 | 03C | EE04 | 040 | FFB5 | 000 | 0004 | FFB5 | 009 | 1001 | 040 | FFB5 |
| 03C | 0100 | 03D | 0100 | 03C | 0100 | 000 | 003C | FFB5 | 009 | 1001 |  |  |

# Заключение

Во время выполнения данной лабораторной работы я познакомился с работой стека и подпрограммам, а также научился использовать новые команды – CALL и RET.