Санкт-Петербургский Национальный Исследовательский Университет ИТМО

Факультет Программной Инженерии и Компьютерной Техники



Лабораторная работа №4 – «Выполнение комплекса программ»

По дисциплине

«Основы Профессиональной Деятельности»

Вариант №46144

Выполнил студент группы P3118:

Богданова Мария Михайловна

Преподаватель:

          Осипов Святослав Владимирович

Санкт-Петербург

2023 г.

**Оглавление**

[Задание 2](#_Toc87370215)

[Ход работы 3](#_Toc87370216)

[Текст исходной программы 3](#_Toc87370217)

[Описание программы 3](#_Toc87370218)

[Трассировка 4](#_Toc87370219)

[Вариант программы с меньшим количеством команд 4](#_Toc87370220)

[Трассировка с данными числами 5](#_Toc87370221)

[Вывод 5](#_Toc87370222)

## **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.

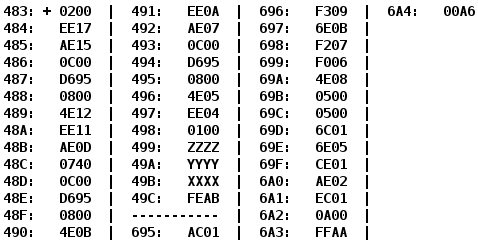


Рис. 1. Список команд заданной программы

## **Ход работы**

### **Текст исходной программы**

|  |  |  |  |
| --- | --- | --- | --- |
| **Адрес** | **Код команды** | **Мнемоника** | **Комментарий** |
| 483 | 0200 | CLA | Обнуление значения аккумулятора |
| 484 | EE17 | ST M | Сохранение значения аккумулятора в ячейку 49С |
| 485 | AE15 | LD X | Загрузка значения ячейки 49B в аккумулятор |
| 486 | 0C00 | PUSH | Декремент SP и запись значения аккумулятора в ячейку с адресом SP |
| 487 | D695 | CALL $P | Вызов подпрограммы по адресу 695 |
| 488 | 0800 | POP | Считывание значения по адресу SP, инкрементация SP |
| 489 | 4E12 | ADD M | Сложение содержимого ячейки 49C с содержимым аккумулятора |
| 48A | EE11 | ST M | Сохранение значения аккумулятора в ячейку 49C |
| 48B | AE0D | LD Z | Загрузка значения ячейки 499 в аккумулятор  Декрементация значения аккумулятора |
| 48C | 0740 | DEC |
| 48D | 0C00 | PUSH | Декремент SP и запись значения аккумулятора в ячейку по адресу SP |
| 48E | D695 | CALL $P | Вызов подпрограммы по адресу 695 |
| 48F | 0800 | POP | Считывание значения по адресу SP, инкрементация SP |
| 490 | 4E0B | ADD M | Сложение содержимого аккумулятора с содержимым ячейки 49С |
| 491 | EE0A | ST M | Сохранение значения аккумулятора в ячейку 49C  Загрузка значения ячейки 49A в аккумулятор |
| 492 | AE07 | LD Y |
| 493 | 0C00 | PUSH | Декремент SP и запись значения аккумулятора в ячейку по адресу SP |
| 494 | D695 | CALL $P | Вызов подпрограммы по адресу 695 |
| 495 | 0800 | POP | Считывание значения по адресу SP, инкрементация SP (верхнего знач. Стека) |
| 496 | 4E05 | ADD M | Сложение значения аккумулятора со значением ячейки 49C |
| 497 | EE04 | ST M | Сохранение значения аккумулятора в ячейку 49С |
| 498 | 0100 | HLT | Остановка работы программы |
| 499 | ZZZZ | Z: word 0x FE83 | Операнд Z |
| 49A | YYYY | Y: word 0x FFD0 | Операнд Y |
| 49B | XXXX | X: word  0x EAC5 | Операнд X |
| 49C | FEAB | M: word 0xFEAB | Результат M |

**Подпрограмма:**

|  |  |  |  |
| --- | --- | --- | --- |
| 695 | AC01 | P: LD &01 | Загрузка в аккумулятор значения ячейки по адресу SP + 1 |
| 696 | F309 | BPL I | Переход к ячейке 6A0, если число не отрицательно |
| 697 | 6E0B | SUB I | Вычитание значения ячейки 6A3 из аккумулятора |
| 698 | F207 | BMI I | Переход к ячейке 6A0, если число отрицательно |
| 699 | F006 | BEQ I | Переход к ячейке 6A0, если значения аккумулятора равно нулю |
| 69A | 4E08 | ADD I | Прибавление значения ячейки 6A3 к аккумулятору |
| 69B | 0500 | ASL | Циклический сдвиг значения аккумулятора влево (умножение на 2) |
| 69C | 0500 | ASL | Циклический сдвиг значения аккумулятора влево (умножение на 2) |
| 69D | 6C01 | SUB &01 | Вычитание значения ячейки по адресу SP+1 из значения аккумулятора |
| 69E | 6E05 | SUB U | Вычитание значение ячейки 6A4 из аккумулятора |
| 69F | CE01 | JUMP B | Переход к ячейке 6A1 |
| 6A0 | AE02 | LD I | Загрузка значения ячейки 6A3 в аккумулятор |
| 6A1 | EC01 | B: ST &01 | Сохранение значения аккумулятора в ячейку по адресу SP + 1 |
| 6A2 | 0A00 | RET | Возврат из подпрограммы, инкремент SP |
| 6A3 | FFAA | I: word 0xFFAA | Константа I |
| 6A4 | 00A6 | U: word 0x00A6 | Константа U |

### **Описание программы**

р

**Назначение программы:**

Программа выдает результат работы функции на координаты X, Y, Z.

R = f(X) + f(Z-1)+f(Y)

**Интерпретация программы в псевдокоде:**

x - одна из координат, для которой выполняется подпрограмма

M = 0;

X = ?; -5435 = 111.1 0101 0011 1011 =111. 0 1010 1100 0101 = EAC5 =

Y = ?; -48 = 1111 1111 11.11 0000 = 1111 1111 11.01 0000 = FFD0

Z = ?; 643 = 1111 11.10 1000 0011 = FE83

func(x):

if x >= 0:

return I;

if x <= I:

return I;

else:

return 3\*x – u;

M+= func(X);

M+= func(Z-1);

M+=func(Y);

**Расположение в памяти БЭВМ программы, исходных данных и результатов:**

483 – 49C - программа

499 – 49B - переменные

49C - результат

695 – 6A2 - подпрограмма

6A3,6A4 – константы

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

483 – адрес первой инструкции,

498 – адрес последней инструкции.

**Область представления:**

X, Y, Z – целочисленные 16-значные значения

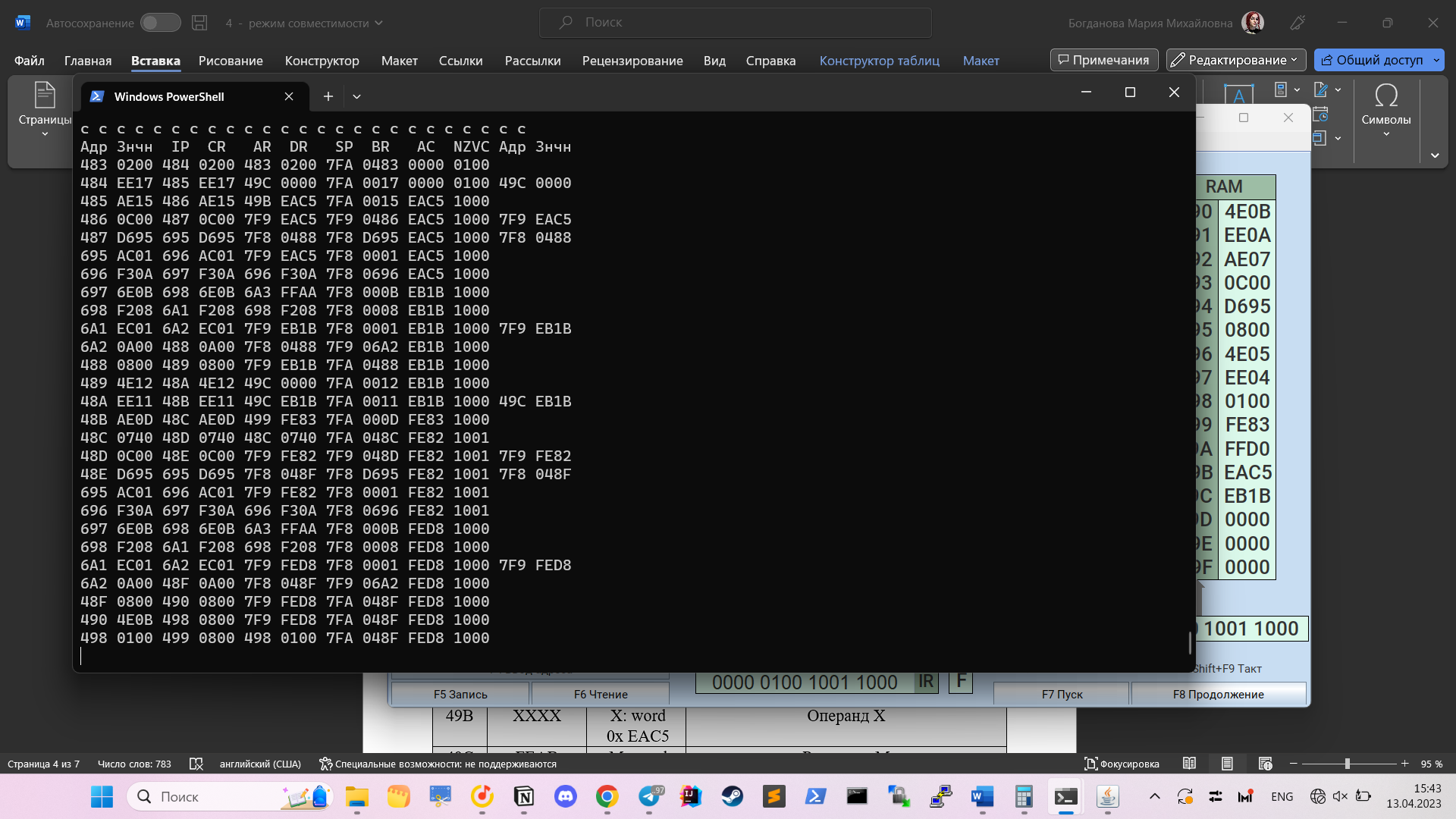
I, U – целочисленные 16-значные константы

M – целочисленный 16-значный результат

**Область допустимых значений**

### Трассировка с данными числами

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Выполненная команда | | | Содержимое регистров процессора после выполнения команды. | | | | | | | | | Ячейка, содержимое к-рой изменилось после выполнения команды | | |
| Адр | Код | IP | | CR | AR | DR | SP | BR | AC | NZVC | Адр | | Новый |
| 483 | 0200 | 484 | | 0200 | 483 | 0200 | 7FA | 0483 | 0000 | 0100 |  | |  |
| 484 | EE17 | 485 | | EE17 | 49C | 0000 | 7FA | 0017 | 0000 | 0100 | 49C | | 0000 |
| 485 | AE15 | 486 | | AE15 | 49B | EAC5 | 7FA | 0015 | EAC5 | 1000 |  | |  |
| 486 | 0C00 | 487 | | 0C00 | 7F9 | EAC5 | 7F9 | 0486 | EAC5 | 1000 | 7F9 | | EAC5 |
| 487 | D695 | 695 | | D695 | 7F8 | 0488 | 7F8 | D695 | EAC5 | 1000 | 7F8 | | 0488 |
| 695 | AC01 | 697 | | AC01 | 7F9 | EAC5 | 7F8 | 0001 | EAC5 | 1000 |  | |  |
| 696 | F30A | 697 | | F30A | 696 | F30A | 7F8 | 0696 | EAC5 | 1000 |  | |  |
| 697 | 6E0B | 698 | | 6E0B | 6A3 | FFAA | 7F8 | 000B | EB1B | 1000 |  | |  |
| 698 | F208 | 6A1 | | F208 | 698 | F208 | 7F8 | 0008 | EB1B | 1000 |  | |  |
| 6A1 | EC01 | 6A2 | | EC01 | 7F9 | EB1B | 7F8 | 0001 | EB1B | 1000 | 7F9 | | EB1B |
| 6A2 | 0A00 | 488 | | 0A00 | 7F8 | 0488 | 7F9 | 06A2 | EB1B | 1000 |  | |  |
| 488 | 0800 | 489 | | 0800 | 7F9 | EB1B | 7FA | 0488 | EB1B | 1000 |  | |  |
| 489 | 4E12 | 48A | | 4E11 | 49C | 0000 | 7FA | 0012 | EB1B | 1000 |  | |  |
| 48A | EE11 | 48B | | EE11 | 49C | EB1B | 7FA | 0011 | EB1B | 1000 | 49C | | EB1B |
| 48B | AE0D | 48C | | AE0D | 499 | FE83 | 7FA | 000D | FE83 | 1000 |  | |  |
| 48C | 0740 | 48D | | 0740 | 48C | 0740 | 7FA | 048C | FE82 | 1001 |  | |  |
| 48D | 0C00 | 48E | | 0C00 | 7F9 | FE82 | 7F9 | 048D | FE82 | 1001 | 7F9 | | FE82 |
| 48E | D695 | 695 | | D695 | 7F8 | 048F | 7F8 | D695 | FE82 | 1001 | 7F8 | | 049F |
| 695 | AC01 | 696 | | AC01 | 7F9 | FE82 | 7F8 | 0001 | FE82 | 1001 |  | |  |
| 696 | F30A | 697 | | F10A | 696 | F30A | 7F8 | 0696 | FE82 | 1001 |  | |  |
| 697 | 6E0B | 698 | | 6E0B | 6A3 | FFAA | 7F8 | 000B | FED8 | 1000 |  | |  |
| 698 | F208 | 6A1 | | F208 | 698 | F208 | 7F8 | 0008 | FED8 | 1000 |  | |  |
| 6A1 | EC01 | 6A2 | | EC01 | 7F9 | FED8 | 7F8 | 0001 | FED8 | 1000 | 7F9 | | FED8 |
| 6A2 | 0A00 | 48F | | 0A00 | 7F8 | 048F | 7F9 | 06A2 | FED8 | 1000 |  | |  |
| 48F | 0800 | 490 | | 0800 | 7F9 | FED8 | 7FA | 048F | FED8 | 1000 |  | |  |
| 490 | 4E0B | 498 | | 0800 | 7F9 | FED8 | 7FA | 048F | FED8 | 1000 |  | |  |
| 498 | 0100 | 499 | | 0800 | 498 | 0100 | 7FA | 048F | FED8 | 1000 |  | |  |



Задание:

\_stdcall

Int16 func (int16 a, int16 b){

Int16 x = 27;

Int16 y = -94;

Int16 z = 8;

Return (x+y-z)&(a+b);

}

Вид:

Int16 func (int16 a, int16 b){

Int16 x = 27;

Int16 y = -94;

Int16 z = 8;

Int16 temp = x+y-z

Return temp&(a+b);

}

org 0x10

START:

CLA

LD #10

PUSH

LD #9

PUSH

CALL P

HLT

P: CLA

PUSH

PUSH

PUSH

PUSH

LD #27

ST &3

LD #0xA2

ST &2

LD #8

ST &1

CLA

LD &3

ADD &2

SUB &1

ST &0

CLA

LD &5

ADD &6

AND &0

SWAP

POP

SWAP

POP

SWAP

POP

SWAP

POP

ST &3

SWAP

POP

SWAP

POP

RET

b 6 3

a 5 2

ret 4 1

x 3

y 2

z 1

temp 0

## Вывод

В ходе выполнения данной лабораторной я исследовала работу с подпрограммами, стеком и командами косвенной относительной адресацией со смещением относительно SP базовой ЭВМ, изучила различные виды адресации, а также научилась работать с массивами при помощи команд цикла и ветвления.