Национальный Исследовательский Университет ИТМО  
Факультет программной инженерии и компьютерной техники

Лабораторная работа № 4  
«Выполнение комплекса программ»

Выполнил: Лысенко Данила Сергеевич  
Группа: P3110  
Вариант: 4757

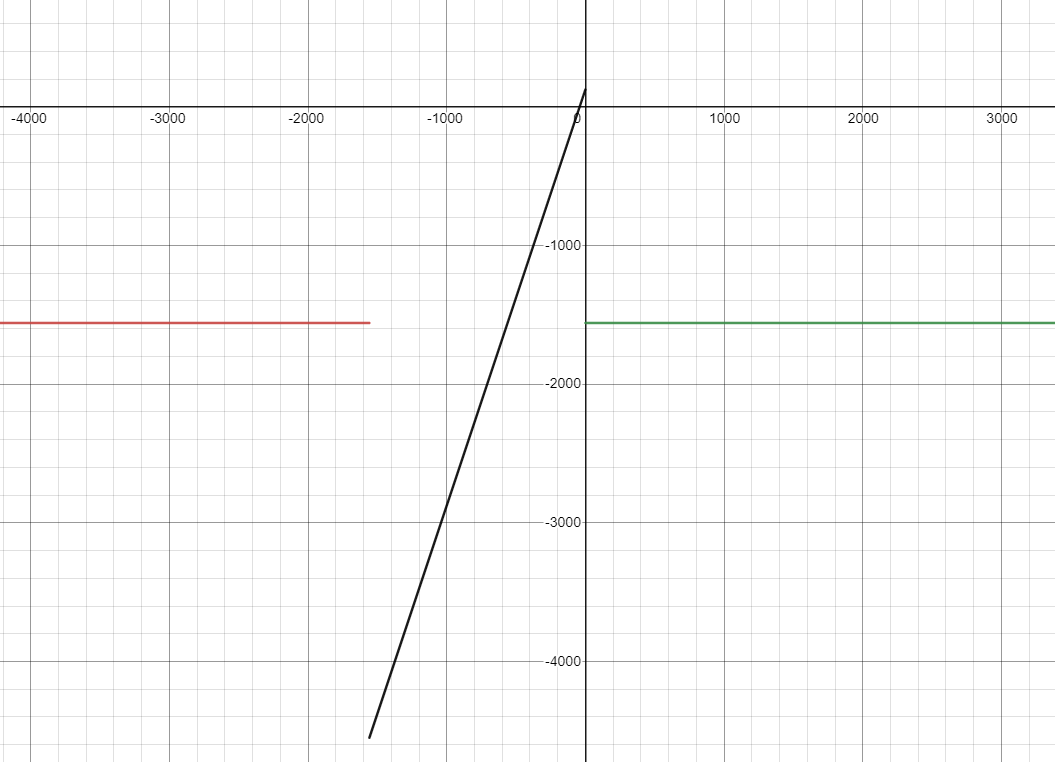
Преподаватель: Перцев Тимофей Сергеевич

Санкт-Петербург  
2021

1. **Текст исходной программы**

|  |  |  |  |
| --- | --- | --- | --- |
| **Адрес** | **Код команды** | **Мнемоника** | **Комментарий** |
| 05D | 0200 | CLA | Очистка значения аккумулятора |
| 05E | EE18 | ST (IP + 24) | Сохранение значения аккумулятора в 077 |
| 05F | AE15 | LD (IP + 21) | Загрузка значения аккумулятора из 075 |
| 060 | 0C00 | PUSH | Запись в стек значения аккумулятора |
| **061** | D6E6 | CALL 6E6 | Вызов подпрограммы 6E6 |
| 062 | 0800 | POP | Чтение из стека в аккумулятор |
| 063 | 4E13 | ADD (IP + 19) | Сложение аккумулятора и ячейки памяти 077 |
| 064 | EE12 | ST (IP + 18) | Сохранение значения аккумулятора в 077 |
| 065 | AE10 | LD (IP + 16) | Загрузка значения аккумулятора из 076 |
| 066 | 0740 | DEC | Декремент аккумулятора |
| 067 | 0C00 | PUSH | Запись в стек значения аккумулятора |
| 068 | D6E6 | CALL 6E6 | Вызов подпрограммы 6E6 |
| 069 | 0800 | POP | Чтение из стека в аккумулятор |
| 06A | 6E0C | SUB (IP + 12) | Вычитание аккумулятора и 077 |
| 06B | EE0B | ST (IP + 11) | Сохранение значения аккумулятора в 077 |
| 06C | AE07 | LD (IP + 7) | Загрузка значения аккумулятора из 074 |
| 06D | 0700 | INC | Инкремент аккумулятора |
| 06E | 0C00 | PUSH | Запись в стек значения аккумулятора |
| 06F | D6E6 | CALL 6E6 | Вызов подпрограммы 6E6 |
| 070 | 0800 | POP | Чтение из стека в аккумулятор |
| 071 | 4E05 | ADD (IP + 5) | Сложение аккумулятора и ячейки памяти 077 |
| 072 | EE04 | ST (IP + 4) | Сохранение значения аккумулятора в 077 |
| 073 | 0100 | HLT | Останов программы |
| 074 | XXXX | X | X |
| 075 | YYYY | Y | Y |
| 076 | ZZZZ | Z | Z |
| 077 | F9E6 | Res | Результат |
|  |  |  |  |
| 6E6 | AC01 | LD (SP + 1) | Загрузка значения из стека 7FF |
| 6E7 | F001 | BEQ 1 | Переход, если равенство |
| 6E8 | F308 | BPL 8 | Переход, если плюс |
| 6E9 | 7E0A | CMP (IP + 10) | Установить флаги по результату AC – 6F4 |
| 6EA | F806 | BLT 6 | Переход, если меньше |
| 6EB | F005 | BEQ 5 | Переход, если равенство |
| 6EC | 0500 | ASL | Арифметический сдвиг влево |
| 6ED | 0500 | ASL | Арифметический сдвиг влево |
| 6EE | 6C01 | SUB (SP + 1) | Вычитание аккумулятора и 7FF |
| 6EF | 4E05 | ADD (IP + 5) | Сложение аккумулятора и 6F5 |
| 6F0 | CE01 | JUMP 1 | Переход |
| 6F1 | AE02 | LD (IP + 2) | Загрузка 6F4 в аккумулятор |
| 6F2 | EC01 | ST (SP + 1) | Сохранение значения аккумулятора в 7FF |
| 6F3 | 0A00 | RET | Возврат из подпрограммы |
| 6F4 | F9E9 | A | Константа A |
| 6F5 | 007C | B | Константа B |

1. **Описание исходной программы**
2. Назначение программы: нахождение значения функции:



1. Расположение в памяти БЭВМ программы, исходных данных и результата:

074-076 – переменные

077 – результат

6F4 – константа A

6F5 – константа B

1. Область представления

A, B, X, Y, Z, Res – целые знаковые шестнадцатеричные числа

1. ОДЗ

Для того чтобы определить одз, проанализируем данную функцию. При значении аргумента функции в промежутке [-1558; 0], функция вернет значение выражения 3x+124. При использовании любого значения из заданного промежутка в функции не возникнет переполнения. При оставшихся значениях аргумента функция вернет заданную константу -1559, следовательно, переполнения также не может возникнуть.

Так как основная программа вычисляет следующее выражение:

то минимально мы можем получить -4550 – 124 – 4550 = -9244 > -2^15

а максимально: 124 + 4550 + 124 = 4798 < 2^15 – 1

В обоих случаях переполнение невозможно.

В функцию как аргументы мы передаем значения z-1, y, x+1. Значит, одз:

**Вывод:**

В процессе выполнения лабораторной работы был получен опыт работы с подпрограммами и стеком, разобрался, каким образом реализован стек в БЭВМ, изучил принцип действия команд PUSH, POP, CALL, RET.