Министерство науки и высшего образования Российской Федерации

федеральное государственное автономное образовательное учреждение высшего образования

«Национальный исследовательский университет ИТМО»

*Факультет Программной инженерии и компьютерной техники*

**Лабораторная работа №3**

Выполнение циклических программ

Вариант №3107

Группа: P3131

Выполнил: Хайкин О. И.

Проверил:

Перцев Тимофей Сергеевич

Санкт-Петербург

2022г

Оглавление

[Задание 3](#_Toc96339222)

[Выполнение работы 3](#_Toc96339223)

[Исходная программа 3](#_Toc96339224)

[Назначение программы и реализуемые ею функции 4](#_Toc96339225)

[Область представления исходных данных и результата 4](#_Toc96339226)

[Область допустимых значений исходных данных и результата 4](#_Toc96339227)

[Расположение в памяти ЭВМ программы, исходных данных и результата 5](#_Toc96339228)

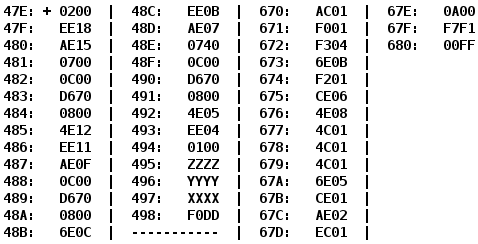
[Адреса первой и последней выполняемых команд 5](#_Toc96339229)

[Таблица трассировки 6](#_Toc96339230)

[Диапазон ячеек памяти для размещения массива 7](#_Toc96339231)

[Вывод 7](#_Toc96339232)

# 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.

# Выполнение работы

## Исходная программа

|  |  |  |  |
| --- | --- | --- | --- |
| Адрес | Код команды | Мнемоника | Комментарии |
| 47E | 0200 | CLA | Очистка аккумулятора |
| 47F | EE18 | ST (IP+18) | Сохранение текущего значения аккумулятора (0) в ячейку 498 |
| 480 | AE15 | LD (IP+15) | Загрузка значения из ячейки 496 в аккумулятор |
| 481 | 0700 | INC | Инкремент |
| 482 | 0C00 | PUSH | Кладём значение на аккумуляторе в стек |
| 483 | D670 | CALL 670 | Вызываем подпрограмму по адресу 670 |
| 484 | 0800 | POP | Достаём значение с вершины стека в аккумулятор |
| 485 | 4E12 | ADD (IP+12) | Сложить значение на аккумуляторе со значением из ячейки 498 |
| 486 | EE11 | ST (IP+11) | Сохранение текущего значения аккумулятора в ячейку 498 |
| 487 | AE0F | LD (IP+F) | Загрузка значения из ячейки 497 в аккумулятор |
| 488 | 0C00 | PUSH | Кладём значение на аккумуляторе в стек |
| 489 | D670 | CALL 670 | Вызываем подпрограмму по адресу 670 |
| 48A | 0800 | POP | Достаём значение с вершины стека в аккумулятор |
| 48B | 6E0C | SUB (IP+C) | Вычесть из значения на аккумуляторе значение из ячейки 498 |
| 48C | EE0B | ST (IP+B) | Сохранение текущего значения аккумулятора в ячейку 498 |
| 48D | AE07 | LD (IP+7) | Загрузка значения из ячейки 495 в аккумулятор |
| 48E | 0740 | DEC | Декремент |
| 48F | 0C00 | PUSH | Кладём значение на аккумуляторе в стек |
| 490 | D670 | CALL 670 | Вызываем подпрограмму по адресу 670 |
| 491 | 0800 | POP | Достаём значение с вершины стека в аккумулятор |
| 492 | 4E05 | ADD (IP+5) | Сложить значение на аккумуляторе со значением из ячейки 498 |
| 493 | EE04 | ST (IP+4) | Сохранение текущего значения аккумулятора в ячейку 498 |
| 494 | 0100 | HLT | Останов |
| 495 | …. | Число (a) |  |
| 496 | …. | Число (b) |  |
| 497 | …. | Число (c) |  |
| 498 | …. | Число (R) | результат |

|  |  |  |  |
| --- | --- | --- | --- |
| Адрес | Код команды | Мнемоника | Комментарии |
| 670 | AC01 | LD (SP+1) | Загрузка значения из предпоследней ячейки стека в аккумулятор |
| 671 | F001 | BEQ IP+1 | Операция ветвления, совершающая переход на ячейку 673, если (Z==1) |
| 672 | F304 | BPL IP+4 | Операция ветвления, совершающая переход на ячейку 677, если (N!=1) |
| 673 | 6E0B | SUB (IP+B) | Вычесть из значения на аккумуляторе значение из ячейки 67F |
| 674 | F201 | BMI IP+1 | Операция ветвления, совершающая переход на ячейку 676, если (N==1) |
| 675 | CE06 | JUMP (IP+6) | Прыжок, переходит к выполнению команды по адресу 67C |
| 676 | 4E08 | ADD (IP+8) | Сложить значение на аккумуляторе со значением из ячейки 67F |
| 677 | 4C01 | ADD (SP+1) | Сложить значение на аккумуляторе со значением из предпоследней ячейки стека |
| 678 | 4C01 | ADD (SP+1) | Сложить значение на аккумуляторе со значением из предпоследней ячейки стека |
| 679 | 4C01 | ADD (SP+1) | Сложить значение на аккумуляторе со значением из предпоследней ячейки стека |
| 67A | 6E05 | SUB (IP+5) | Вычесть из значения на аккумуляторе значение из ячейки 680 |
| 67B | CE01 | JUMP (IP+1) | Прыжок, переходит к выполнению команды по адресу 67D |
| 67C | AE02 | LD (IP+2) | Загрузка значения из ячейки 67F в аккумулятор |
| 67D | EC01 | ST (SP+1) | Сохранение текущего значения аккумулятора в предпоследнюю ячейку стека |
| 67E | 0A00 | RET | Возвращение из подпрограммы к основной программе |
| 67F | F7F1 | число (d) | -30705 |
| 680 | 00FF | Число (e) | 255 |

## Назначение программы и реализуемые ею функции

Назначение: высчитывание результата по формуле R=f(c)-f(b+1)+f(a), где a, b и с – числа, а f(x) определена как:

, где d и e – определённые числа.

## Область представления исходных данных и результата

Результат (498): знаковое 16-разрядное число

Исходные числа (495-497): знаковые 16-разрядные числа

Константы (67F-680): знаковые 16-разрядные числа

## Область допустимых значений исходных данных и результата

**Результат (R):** 0≤R≤L, где L – длина массива.

**Элементы массива (Mi):** -215 ≤ Mi ≤ 215-1(любые знаковые числа)

**Длина массива чисел (L):** длина массива – число положительное, то есть 1≤L≤127. (длина задаётся через команду AFXX, где XX – длина массива, отсюда верхняя граница, причём один бит идёт под знак числа)

*P.S: технически L может быть нулём или отрицательным числом, но такое значение воспримется программой как L=1. Если допустить это, то -128≤L≤127.*

**Адрес первого элемента массива (A):** любая ячейка памяти БЭВМ, незанятая данными/командами, т.е. A ∊ [000,245-L][257,7FF] (условие про 245-L нужно, чтобы массив не «налез» на ячейки, занятые программой)

*P.S: ничто не мешает массиву дойти до адреса 7FF и продолжиться с адреса 000*

## Расположение в памяти ЭВМ программы, исходных данных и результата

**Исходные данные:**

245 – адрес первого элемента массива

247 – длина массива (технически загружается с использованием команды AFXX, а не вносится как исходные данные)

257-25A – массив чисел (FEED, 4713, 0000, FFFF)

**Результат:**

248

**Команды:**

249-256

**Временные данные:**

246 – текущий элемент массива

## Адреса первой и последней выполняемых команд

249 – первая

256 – последняя

## Таблица трассировки

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Выполняемая команда | | Содержимое регистров после выполнения команды | | | | | | | | Ячейка, содержимое которой изменилось после выполнения команды | |
| Адрес | Код | IP | CR | AR | DR | SP | BR | AC | NZVC | Адрес | Новый код |
| 249 | 0200 | 24A | 0200 | 249 | 0200 | 000 | 0249 | 0000 | 0100 |  |  |
| 24A | EEFD | 24B | EEFD | 248 | 0000 | 000 | FFFD | 0000 | 0100 | 248 | 0000 |
| 24B | AF04 | 24C | AF04 | 24B | 0004 | 000 | 0004 | 0004 | 0000 |  |  |
| 24C | EEFA | 24D | EEFA | 247 | 0004 | 000 | FFFA | 0004 | 0000 | 247 | 0004 |
| 24D | AEF7 | 24E | AEF7 | 245 | 0257 | 000 | FFF7 | 0257 | 0000 |  |  |
| 24E | EEF7 | 24F | EEF7 | 246 | 0257 | 000 | FFF7 | 0257 | 0000 | 246 | 0257 |
| 24F | AAF6 | 250 | AAF6 | 257 | FEED | 000 | FFF6 | FEED | 1000 | 246 | 0258 |
| 250 | F203 | 254 | F203 | 250 | F203 | 000 | 0003 | FEED | 1000 |  |  |
| 254 | 8247 | 255 | 8247 | 247 | 0003 | 000 | 0002 | FEED | 1000 | 247 | 0003 |
| 255 | CEF9 | 24F | CEF9 | 255 | 024F | 000 | FFF9 | FEED | 1000 |  |  |
| 24F | AAF6 | 250 | AAF6 | 258 | 4713 | 000 | FFF6 | 4713 | 0000 | 246 | 0259 |
| 250 | F203 | 251 | F203 | 250 | F203 | 000 | 0250 | 4713 | 0000 |  |  |
| 251 | AEF6 | 252 | AEF6 | 248 | 0000 | 000 | FFF6 | 0000 | 0100 |  |  |
| 252 | 0700 | 253 | 0700 | 252 | 0700 | 000 | 0252 | 0001 | 0000 |  |  |
| 253 | EEF4 | 254 | EEF4 | 248 | 0001 | 000 | FFF4 | 0001 | 0000 | 248 | 0001 |
| 254 | 8247 | 255 | 8247 | 247 | 0002 | 000 | 0001 | 0001 | 0000 | 247 | 0002 |
| 255 | CEF9 | 24F | CEF9 | 255 | 024F | 000 | FFF9 | 0001 | 0000 |  |  |
| 24F | AAF6 | 250 | AAF6 | 259 | 0000 | 000 | FFF6 | 0000 | 0100 | 246 | 025A |
| 250 | F203 | 251 | F203 | 250 | F203 | 000 | 0250 | 0000 | 0100 |  |  |
| 251 | AEF6 | 252 | AEF6 | 248 | 0001 | 000 | FFF6 | 0001 | 0000 |  |  |
| 252 | 0700 | 253 | 0700 | 252 | 0700 | 000 | 0252 | 0002 | 0000 |  |  |
| 253 | EEF4 | 254 | EEF4 | 248 | 0002 | 000 | FFF4 | 0002 | 0000 | 248 | 0002 |
| 254 | 8247 | 255 | 8247 | 247 | 0001 | 000 | 0000 | 0002 | 0000 | 247 | 0001 |
| 255 | CEF9 | 24F | CEF9 | 255 | 024F | 000 | FFF9 | 0002 | 0000 |  |  |
| 24F | AAF6 | 250 | AAF6 | 25A | FFFF | 000 | FFF6 | FFFF | 1000 | 246 | 025B |
| 250 | F203 | 254 | F203 | 250 | F203 | 000 | 0003 | FFFF | 1000 |  |  |
| 254 | 8247 | 256 | 8247 | 247 | 0000 | 000 | FFFF | FFFF | 1000 | 247 | 0000 |
| 256 | 0100 | 257 | 0100 | 256 | 0100 | 000 | 0256 | FFFF | 1000 |  |  |

## Диапазон ячеек памяти для размещения массива

[257,7FF][000,244]. Массив может продолжаться после достижения ячейки 7FF (конца памяти БЭВМ) с ячейки 000. Максимальная длина массива – 127 ячеек.

# Вывод

В ходе выполнения третьей лабораторной работы я изучил средства управления вычислительным процессом БЭВМ и научился использовать команды с относительной адресацией.