

Федеральное государственное автономное образовательное

учреждение высшего образования

«Национальный исследовательский университет ИТМО»

**Отчёт по лабораторной работе 5**

Предмет: ОПД

Варинат: 1564

Выполнил**:** студент группы Р3115 Храбров Артём Алексеевич

Проверил**:** Блохина Елена Николаевна

Дата сдачи: xx.xx.xx

2025

Задание

1. Программа осуществляет асинхронный вывод данных на ВУ-1
2. Программа начинается с адреса 1EC16. Размещаемая строка находится по адресу 5ED16.
3. Строка должна быть представлена в кодировке КОИ-8.
4. Формат представления строки в памяти: АДР0: ДЛИНА АДР1: СИМВ2 СИМВ1 АДР2: СИМВ4 СИМВ3 ..., где ДЛИНА - 16 разрядное слово, где значащими являются 8 младших бит.
5. Вывод строки начинается со вывода количества символов (1 байт), и должен быть завершен по выводу их необходимого количества.

Код на ассемблере

Текст исходной программы

Программа

|  |  |  |  |
| --- | --- | --- | --- |
| **Адрес** | **Код команды** | **Мнемоника** | **Комментарий** |
| 24E | 0200 | CLA | 0 🡪 AC |
| 24F | EE1A | ST 26A | 0 🡪 26A |
| 250 | AE18 | LD 269 | 269 🡪 AC (Загрузка X) |
| 251 | 0740 | DEC | AC – 1 🡪 AC |
| 252 | 0C00 | PUSH | Помещение в стек |
| 253 | D669 | CALL $669 | Вызов подпрограммы по адресу 669 |
| 254 | 0800 | POP | Извлечение значения из стека |
| 255 | 0700 | INC | AC + 1 🡪 AC |
| 256 | 6E13 | SUB 26A | AC – 26A 🡪 AC |
| 257 | EE12 | ST 26A | AC 🡪 26A |
| 258 | AE0F | LD 269 | 268 🡪 AC (загрузка Y) |
| 259 | 0700 | INC | AC + 1 🡪 AC |
| 25A | 0C00 | PUSH | Помещение в стек |
| 25B | D669 | CALL $669 | Вызов подпрограммы по адресу 669 |
| 25C | 0800 | POP | Извлечение значения из стека |
| 25D | 0700 | INC | AC + 1 🡪 AC |
| 25E | 4E0B | ADD 26B | AC + 26B 🡪 AC |
| 25F | EE0A | ST 26A | AC 🡪 26A |
| 260 | AE06 | LD 267 | 267 🡪 AC (загрузка Z) |
| 261 | 0C00 | PUSH | Помещение в стек |
| 262 | D669 | CALL $669 | Вызов подпрограммы по адресу 669 |
| 263 | 0800 | POP | Извлечение значения из стека |
| 264 | 6E05 | SUB 26A | AC – 26A 🡪 AC |
| 265 | EE04 | ST 26A | AC 🡪 26A |
| 266 | 0100 | HLT | Остановка |

Подпрограмма

|  |  |  |  |
| --- | --- | --- | --- |
| **Адрес** | **Код команды** | **Мнемоника** | **Комментарий** |
| 669 | AC01 | LD (SP + 1) | Загрузка в аккумулятор аргумента (arg) |
| 66A | F207 | BMI (IP + 7) | Переход к адресу 672, если AC ≤ 0 |
| 66B | F006 | BEQ (IP + 6) |
| 66C | 7E08 | CMP (IP + 8) | Переход к адресу 672, если AC ≥ A |
| 66D | F904 | BGE (IP + 4) |
| 66E | 0500 | ASL | AC15 🡪 C; 0 🡪 AC0 (умножение на 2) |
| 66F | 4C01 | ADD (SP + 1) | Сложение AC с arg |
| 670 | 4E05 | ADD (IP + 5) | AC + B |
| 671 | CE01 | JUMP (IP + 1) | Переход к 673 |
| 672 | AE02 | LD (IP + 2) | A 🡪 AC |
| 673 | EC01 | ST (SP +1) | Запись в стек |
| 674 | 0A00 | RET | Возврат |

**Назначение программы**

*Основная программа:*

Вызывает три раза подпрограмму с аргументами (X – 1), (Y + 1), Z и обрабатывает получившиеся резултаты, формируя итоговый.

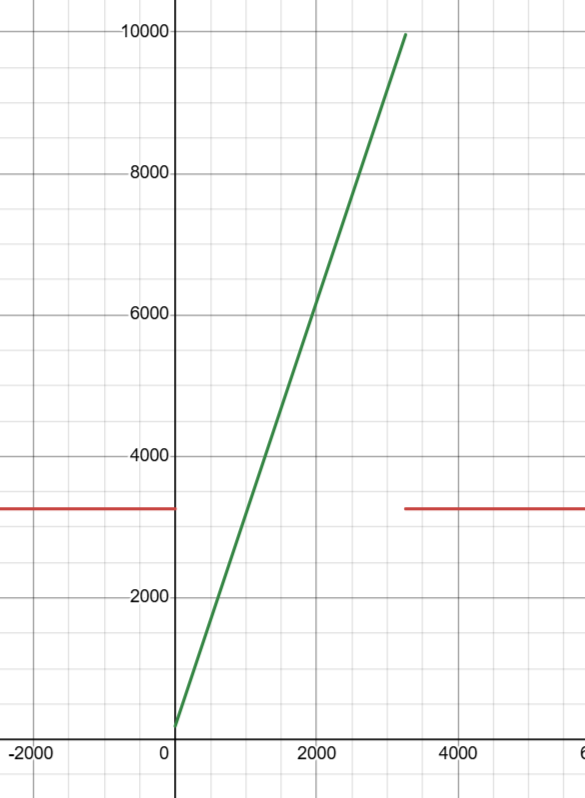
*Подпрограмма:*

Реализует функциюF(x) =

*Комплекс программ:*

Вычисляет значение выраженияR = F(X – 1) + F(Y + 1) – F(Z) + 2

График реализуемой подпрограммой функции



**Область представления**

* X, Y, Z, R – целые 16 разрядные знаковые числа [-215; 215 – 1]
* A = 3259, B = 187 – целые 16 разрядные знаковые числа
* F(X - 1), F(Y + 1), F(Z) - целые 16 разрядные знаковые числа[-215; 215 -1]

Область допустимых значений:

Рассмотрим функцию F(x):

1. Если x ∈ (-∞; 0] ⋃ [A; +∞), то вернётся значение A = 3259. Переполнения не произойдет.
2. Если x ∈ (0; A), то вернётся 3x + B

Функция монотонно возрастает, тогда max и min значения:

Fmin = F(1) = B + 3 = 190

Fmax = F(A – 1) = 3A + B - 3 = 9961

Получается, что все значения функции находятся в промежутке [190; 9961]

Программа вычисляет результат по этой формуле R = F(X – 1) + F(Y + 1) – F(Z) + 2. Ее max и min значения:

Rmax = 9961 + 9961 - 190 + 2 = 19734

Rmin = 190 + 190 – 9961 + 2 = -9589

-215 < -9589 < 19734 < 215 – 1

Получается, что все значения резултата лежат в допустимых границах и не вызовут переполнения.

Итоговая ОДЗ:

Расположение в памяти и назначение исходных данных

Основная программа:

Расположение программы: **24E** – **266**

X: **Ячейка 269** – входные данные

Y: **Ячейка 268** – входные данные

Z: **Ячейка 267** – входные данные

Подпрограмма:

Расположение подпрограммы: **669** – **674**

A: **Ячейка** **675 –** Параметр подпрограммы

B: **Ячейка 676 –** Параметр подпрограммы

Комплекс программ:

R: **Ячейка 26A** – результат работы комплекса программ

Адреса первой и последней исполняемой команды

Основная программа: Первая – **24E,** последняя - **266**

Подпрограмма: Первая – **669**, последняя – **674**

Трассировка программы

X = 215610 = 086С16

Y = -98710 = FC2516

Z = 458710 = 11EB16

Таблица трассировки программы:

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **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 | Новый код |
| 24E | 0200 | 24F | 0200 | 24E | 0200 | 000 | 024E | 0000 | 004 | 0100 |  |  |
| 24F | EE1A | 250 | EE1A | 26A | 0000 | 000 | 001A | 0000 | 004 | 0100 | 26A | 0000 |
| 250 | AE18 | 251 | AE18 | 269 | 11EB | 000 | 0018 | 11EB | 000 | 0000 |  |  |
| 251 | 0740 | 252 | 0740 | 251 | 0740 | 000 | 0251 | 11EA | 001 | 0001 |  |  |
| 252 | 0C00 | 253 | 0C00 | 7FF | 11EA | 7FF | 0252 | 11EA | 001 | 0001 | 7FF | 11EA |
| 253 | D669 | 669 | D669 | 7FE | 0254 | 7FE | D669 | 11EA | 001 | 0001 | 7FE | 0254 |
| 669 | AC01 | 66A | AC01 | 7FF | 11EA | 7FE | 0001 | 11EA | 001 | 0001 |  |  |
| 66A | F207 | 66B | F207 | 66A | F207 | 7FE | 066A | 11EA | 001 | 0001 |  |  |
| 66B | F006 | 66C | F006 | 66B | F006 | 7FE | 066B | 11EA | 001 | 0001 |  |  |
| 66C | 7E08 | 66D | 7E08 | 675 | 0CBB | 7FE | 0008 | 11EA | 001 | 0001 |  |  |
| 66D | F904 | 672 | F904 | 66D | F904 | 7FE | 0004 | 11EA | 001 | 0001 |  |  |
| 672 | AE02 | 673 | AE02 | 675 | 0CBB | 7FE | 0002 | 0CBB | 001 | 0001 |  |  |
| 673 | EC01 | 674 | EC01 | 7FF | 0CBB | 7FE | 0001 | 0CBB | 001 | 0001 | 7FF | 0CBB |
| 674 | 0A00 | 254 | 0A00 | 7FE | 0254 | 7FF | 0674 | 0CBB | 001 | 0001 |  |  |
| 254 | 0800 | 255 | 0800 | 7FF | 0CBB | 000 | 0254 | 0CBB | 001 | 0001 |  |  |
| 255 | 0700 | 256 | 0700 | 255 | 0700 | 000 | 0255 | 0CBC | 000 | 0000 |  |  |
| 256 | 6E13 | 257 | 6E13 | 26A | 0000 | 000 | 0013 | 0CBC | 001 | 0001 |  |  |
| 257 | EE12 | 258 | EE12 | 26A | 0CBC | 000 | 0012 | 0CBC | 001 | 0001 | 26A | 0CBC |
| 258 | AE0F | 259 | AE0F | 268 | FC25 | 000 | 000F | FC25 | 009 | 1001 |  |  |
| 259 | 0700 | 25A | 0700 | 259 | 0700 | 000 | 0259 | FC26 | 008 | 1000 |  |  |
| 25A | 0C00 | 25B | 0C00 | 7FF | FC26 | 7FF | 025A | FC26 | 008 | 1000 | 7FF | FC26 |
| 25B | D669 | 669 | D669 | 7FE | 025C | 7FE | D669 | FC26 | 008 | 1000 | 7FE | 025C |
| 669 | AC01 | 66A | AC01 | 7FF | FC26 | 7FE | 0001 | FC26 | 008 | 1000 |  |  |
| 66A | F207 | 672 | F207 | 66A | F207 | 7FE | 0007 | FC26 | 008 | 1000 |  |  |
| 672 | AE02 | 673 | AE02 | 675 | 0CBB | 7FE | 0002 | 0CBB | 000 | 0000 |  |  |
| 673 | EC01 | 674 | EC01 | 7FF | 0CBB | 7FE | 0001 | 0CBB | 000 | 0000 | 7FF | 0CBB |
| 674 | 0A00 | 25C | 0A00 | 7FE | 025C | 7FF | 0674 | 0CBB | 000 | 0000 |  |  |
| 25C | 0800 | 25D | 0800 | 7FF | 0CBB | 000 | 025C | 0CBB | 000 | 0000 |  |  |
| 25D | 0700 | 25E | 0700 | 25D | 0700 | 000 | 025D | 0CBC | 000 | 0000 |  |  |
| 25E | 4E0B | 25F | 4E0B | 26A | 0CBC | 000 | 000B | 1978 | 000 | 0000 |  |  |
| 25F | EE0A | 260 | EE0A | 26A | 1978 | 000 | 000A | 1978 | 000 | 0000 | 26A | 1978 |
| 260 | AE06 | 261 | AE06 | 267 | 086C | 000 | 0006 | 086C | 000 | 0000 |  |  |
| 261 | 0C00 | 262 | 0C00 | 7FF | 086C | 7FF | 0261 | 086C | 000 | 0000 | 7FF | 086C |
| 262 | D669 | 669 | D669 | 7FE | 0263 | 7FE | D669 | 086C | 000 | 0000 | 7FE | 0263 |
| 669 | AC01 | 66A | AC01 | 7FF | 086C | 7FE | 0001 | 086C | 000 | 0000 |  |  |
| 66A | F207 | 66B | F207 | 66A | F207 | 7FE | 066A | 086C | 000 | 0000 |  |  |
| 66B | F006 | 66C | F006 | 66B | F006 | 7FE | 066B | 086C | 000 | 0000 |  |  |
| 66C | 7E08 | 66D | 7E08 | 675 | 0CBB | 7FE | 0008 | 086C | 008 | 1000 |  |  |
| 66D | F904 | 66E | F904 | 66D | F904 | 7FE | 066D | 086C | 008 | 1000 |  |  |
| 66E | 0500 | 66F | 0500 | 66E | 086C | 7FE | 066E | 10D8 | 000 | 0000 |  |  |
| 66F | 4C01 | 670 | 4C01 | 7FF | 086C | 7FE | 0001 | 1944 | 000 | 0000 |  |  |
| 670 | 4E05 | 671 | 4E05 | 676 | 00BB | 7FE | 0005 | 19FF | 000 | 0000 |  |  |
| 671 | CE01 | 673 | CE01 | 671 | 0673 | 7FE | 0001 | 19FF | 000 | 0000 |  |  |
| 673 | EC01 | 674 | EC01 | 7FF | 19FF | 7FE | 0001 | 19FF | 000 | 0000 | 7FF | 19FF |
| 674 | 0A00 | 263 | 0A00 | 7FE | 0263 | 7FF | 0674 | 19FF | 000 | 0000 |  |  |
| 263 | 0800 | 264 | 0800 | 7FF | 19FF | 000 | 0263 | 19FF | 000 | 0000 |  |  |
| 264 | 6E05 | 265 | 6E05 | 26A | 1978 | 000 | 0005 | 0087 | 001 | 0001 |  |  |
| 265 | EE04 | 266 | EE04 | 26A | 0087 | 000 | 0004 | 0087 | 001 | 0001 | 26A | 0087 |
| 266 | 0100 | 267 | 0100 | 266 | 0100 | 000 | 0266 | 0087 | 001 | 0001 |  |  |

Вывод

В процессе выполнения работы я познакомился с реализацией выполнения подпрограмм в БЭВМ и особенностями передачи параметров подпрограмме, возврате значений.

ДОП

Модернизировать КП следующим образом

Элементы массива имеют длину 13 бит, располагаются в памяти без плотной упаковки. ПП реализует ту же функцию. Элементы массива знаковые числа, знак в старшем значащем бите (в 13).

Необходимо учитывать, что при такой упаковке в оставшихся битах элементов массива может находится мусор.

Модернизация КП

267 – X – M[0]

268 – Y – M[1]

269 – Z – M[2]

26A - результат

26B – адрес первого элемента массива (267)

26C – переменная для прохода по массиву (счётчик)

26D –– ячейка для промежуточного хранения элемента массива

26E – маска получения знака 0001 0000 0000 0000

26F – маска очистки старших 3 битов 0001 1111 1111 1111

270 – маска для отрицательного числа 1110 0000 0000 0000

Программа

|  |  |  |  |
| --- | --- | --- | --- |
| **Адрес** | **Код команды** | **Мнемоника** | **Комментарий** |
| 24E | 0200 | CLA | 0 🡪 AC |
| 24F | EE1A | ST 26A | 0 🡪 26A (обнуление реультата) |
| 250 | AF03 | LD #03 | Загрузка длины массива |
| 251 | E26C | ST 26C | Сохранение счетчика цикла |
| 252 | AA18 | LD IP + 18 | Автоинкрементная загрузка из 26B |
| 253 | E26D | ST 26D | Промежуточное хранение элемента массива |
| 254 | 226E | AND 26E | С маской 000100.. Получаем знак числа |
| 255 | F003 | BEQ IP + 3 | Если == 0 то число +, переход к 259. Если знак минус, то 13 бит будет 1 |
| 256 | A26D | LD 26D | Загрузить обратно элемент массива |
| 257 | 226F | AND 26F | С маской 0001111... |
| 258 | 4270 | ADD 270 | Добавить 11100000 |
| 259 | CE02 | JUMP (IP + 2) | Перейти на 25С |
| 25A | A26D | LD 26D | Загрузить обратно элемент массива |
| 25B | 226F | AND 26F | С маской 0001111... |
|  | 0С00 | PUSH | Помещение в стек |
|  | D669 | CALL $669 | Вызов подпрограммы по адресу 669 |
|  | 0800 | POP | Получение со стека |
| 25C | 426A | ADD 26A | Сложение с результатом |
| 25D | E26A | ST 26A | Сохранение результата |
| 25E | 826C | LOOP 26C | цикл |
| 25F | C252 | JUMP 252 | Переход на начало цикла |
| 260 | 0100 | HLT | Остановка |
| …. | … | … | … |
| 267 | A001 |  | X 1010 0000 0000 0001 = 1 |
| 268 | 9FFС |  | Y 1001 1111 1111 1100 = -4 |
| 269 | E005 |  | Z 1110 0000 0000 0101 = 5 |
| 26A |  |  | Результат должен быть 2 |
| 26B | 0267 |  | Адрес 1 элемента массива |
| 26C |  |  | Счётчик цикла |
| 26D |  |  | Ячейка промежуточного хранения элемента |
| 26E | 1000 |  | Маска получения знака |
| 26F | 1FFF |  | Маска очистки старших 3 бит |
| 270 | E000 |  | Маска установки в 1 старших 3 бит |