Факультет Программной Инженерии и Компьютерной Техники

Лабораторная работа №3

Вариант 7868

Выполнил:

Козаченко Данил Александрович

Группа P3112

Проверил:

Абузов Ярослав Александрович

Содержание

[Задание 3](#_Toc190817587)

[Ход выполнения 4](#_Toc190817588)

[*1) Текст исходной программы 4*](#_Toc190817589)

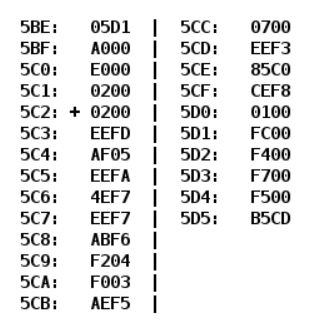
[*2) Описание программы: 5*](#_Toc190817590)

[*3) Таблица трассировки: 6*](#_Toc190817591)

[Заключение 7](#_Toc190817592)

Задание

1. По выданному преподавателем варианту восстановить текст заданного варианта программы, определить предназначение и составить описание программы, определить область представления и область допустимых значений исходных данных и результата, выполнить трассировку программы.
2. Получить у преподавателя номер варианта к лабораторной работе. Изучить способы и средства организации циклических программ с использованием системы команд базовой ЭВМ (приложение В, п.1.7). Восстановить текст заданного варианта программы. Составить описание программы.



1. Получить допуск к лабораторной работе, предъявив преподавателю подготовленные материалы. Получить у преподавателя новые исходные данные. Значения элементов массива из задания используются только для определения функциональности программы! Занести в память базовой ЭВМ заданный вариант программы, новые исходные данные и заполнить таблицу трассировки, выполняя эту программу по командам.
2. Отчет по работе должен быть составлен аналогично лабораторной работе №2, за исключением п. 4 (разработка программы с сокращенным числом команд). Необходимо привести диапазон всех ячеек памяти, где может размещаться массив исходных данных.

Ход выполнения

1. Текст исходной программы представлен в Таблице 1:

Таблица 1

|  |  |  |  |
| --- | --- | --- | --- |
| **Адрес** | **Код команды** | **Мнемоника** | **Комментарии** |
| 5BE | 05D1 | arr\_first | Адрес первого элемента массива |
| 5BF | A000 | arr\_current | Адрес текущего элемента массива |
| 5C0 | E000 | arr\_len | Число необработанных элементов массива |
| 5C1 | 0200 | result | Переменная RESULT |
| 5C2 | 0200 | CLA | Очистка аккумулятора  0 ⇒ AC |
| 5C3 | EEFD | ST (IP-3) | Запись значения содержимого аккумулятора в ячейку памяти 5C1, расположенную на 3 ячейки раньше IP  AC ⇒ M(5C1) |
| 5C4 | AF05 | LD #05 | Запись значения 5 в аккумулятор  5 ⇒ AC |
| 5C5 | EEFA | ST (IP-6) | Запись значения содержимого аккумулятора в ячейку памяти 5C0, расположенную на 6 ячеек раньше IP  AC ⇒ M(5C0) |
| 5C6 | 4EF7 | ADD (IP-9) | Сложение значения содержимого ячейки памяти 5BE, расположенной на 9 ячеек раньше IP и аккумулятора, с последующей записью результата в аккумулятор  M(5BE)+ AC ⇒ AC |
| 5C7 | EEF7 | ST (IP-9) | Запись значения содержимого аккумулятора в ячейку памяти 5BF, расположенную на 9 ячеек раньше IP  AC ⇒ M(5BF) |
| 5C8 | ABF6 | ADD –(IP-10) | Вычитает 1 из значения ячейки 5BF, расположенной на 10 ячеек раньше IP и берёт это значение как адрес ячейки, значение которой складывает с аккумулятором, с последующей записью результата в аккумулятор  M(5D5)+ AC ⇒ AC |
| 5C9 | F204 | BMI (IP+4) | Если N == 1, то IP+4+1 ⇒ IP |
| 5CA | F003 | BEQ (IP+3) | Если Z == 1, то IP+3+1 ⇒ IP |
| 5CB | AEF5 | ADD (IP-11) | Сложение значения содержимого ячейки памяти 5C1, расположенной на 11 ячеек раньше IP и аккумулятора, с последующей записью результата в аккумулятор  M(5C1)+ AC ⇒ AC |
| 5CC | 0700 | INC | Инкремент аккумулятора  AC + 1 ⇒ AC |
| 5CD | EEF3 | ST (IP-13) | Запись значения содержимого аккумулятора в ячейку памяти 5C1, расположенную на 13 ячеек раньше IP  AC ⇒ M(5C1) |
| 5CE | 85C0 | LOOP 5C0 | M(5C0) - 1 ⇒ M(5C0);  Если M(5C0) <= 0, то IP+1 ⇒ IP |
| 5CF | CEF8 | JUMP (IP-8) | Перескок в ячейку 5C8, расположенную на 8 раньше IP  IP-8+1 ⇒ IP |
| 5D0 | 0100 | HLT | Отключение тактового генератора, остановка программы |
| 5D1 | FC00 | arr[0] | Элементы массива |
| 5D2 | F400 | arr[1] |
| 5D3 | F700 | arr[2] |
| 5D4 | F500 | arr[3] |
| 5D5 | B5CD | arr[4] |

1. Описание программы:

***Программа пошагово:***

5C1 = 000016

AC = 000516

5C0 = 000516

AC = Адрес последнего элемента массива + 1

5BF = Адрес последнего элемента массива + 1

do:

AC = Элемент массива + AC

Если AC > 0, то

5C1 = AC + 5C1 + 1

5C0 = 5C0 – 1

while 5C0 > 0

***Программа реализуют следующую функцию:***

Адрес последнего элемента массива + 1 + накопление суммы элементов массива.

Если сумма больше 0, то накопление полученных результатов + 1 в переменную RESULT

***Область представления:***

* + RESULT, arr[i] – знаковое, 16-ти разрядное число
  + arr\_len – беззнаковое, 16-ти разрядное число
  + arr\_first, arr\_current– беззнаковое, 11-ти разрядное число

***Область допустимых значений:***

* arr\_len ∈ [1..558]
* arr\_first ∈ [0.. 5BE16 - arr\_len] ∪ [5D116 .. 7FF16 - arr\_len]
* arr[i] ∈ [-215 .. 215 - 1]
* **Расположение в памяти ЭВМ программы, исходных данных и результатов**

5C2-5D0 – программа;

5BE, 5D1–5D5 – исходные данные;

5BF, 5C0 – промежуточные данные;

5C1 – итоговый результат

* **Адреса первой и последней выполняемой команд программы**

5C2 – адрес первой команды

5D0 – адрес последней команды

1. Таблица трассировки:

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Выполняемая команда** | | **Содержимое регистров процессора после выполнения команды** | | | | | | | | **Ячейка, содержимое которой изменилось после выполнения команды** | |
| Адрес | Код | IP | CR | AR | DR | SP | BR | AC | NZVC | Адрес | Новый код |
| 5BE | 05D1 | - | - | - | - | - | - | - | - | - | - |
| 5BF | A000 | - | - | - | - | - | - | - | - | - | - |
| 5C0 | E000 | - | - | - | - | - | - | - | - | - | - |
| 5C1 | 0200 | - | - | - | - | - | - | - | - | - | - |
| 5C2 | 0200 | 5C3 | 0200 | 5C2 | 0200 | 000 | 05C2 | 0000 | 0100 | - | - |
| 5C3 | EEFD | 5C4 | EEFD | 5C1 | 0000 | 000 | FFFD | 0000 | 0100 | 5C1 | 0000 |
| 5C4 | AF05 | 5C5 | AF05 | 5C4 | 0005 | 000 | 0005 | 0005 | 0000 | - | - |
| 5C5 | EEFA | 5C6 | EEFA | 5C0 | 0005 | 000 | FFFA | 0005 | 0000 | 5C0 | 0005 |
| 5C6 | 4EF7 | 5C7 | 4EF7 | 5BE | 05D1 | 000 | FFF7 | 05D6 | 0000 | - | - |
| 5C7 | EEF7 | 5C8 | EEF7 | 5BF | 05D6 | 000 | FFF7 | 05D6 | 0000 | 5BF | 05D6 |
| 5C8 | ABF6 | 5C9 | ABF6 | 5D5 | B5CD | 000 | FFF6 | B5CD | 1000 | 5BF | 05D5 |
| 5C9 | F204 | 5CE | F204 | 5C9 | F204 | 000 | 0004 | B5CD | 1000 | - | - |
| 5CE | 85C0 | 5CF | 85C0 | 5C0 | 0004 | 000 | 0003 | B5CD | 1000 | 5C0 | 0004 |
| 5CF | CEF8 | 5C8 | CEF8 | 5CF | 05C8 | 000 | FFF8 | B5CD | 1000 | - | - |
| 5C8 | ABF6 | 5C9 | ABF6 | 5D4 | F500 | 000 | FFF6 | F500 | 1000 | 5BF | 05D4 |
| 5C9 | F204 | 5CE | F204 | 5C9 | F204 | 000 | 0004 | F500 | 1000 | - | - |
| 5CE | 85C0 | 5CF | 85C0 | 5C0 | 0003 | 000 | 0002 | F500 | 1000 | 5C0 | 0003 |
| 5CF | CEF8 | 5C8 | CEF8 | 5CF | 05C8 | 000 | FFF8 | F500 | 1000 | - | - |
| 5C8 | ABF6 | 5C9 | ABF6 | 5D3 | F700 | 000 | FFF6 | F700 | 1000 | 5BF | 05D3 |
| 5C9 | F204 | 5CE | F204 | 5C9 | F204 | 000 | 0004 | F700 | 1000 | - | - |
| 5CE | 85C0 | 5CF | 85C0 | 5C0 | 0002 | 000 | 0001 | F700 | 1000 | 5C0 | 0002 |
| 5CF | CEF8 | 5C8 | CEF8 | 5CF | 05C8 | 000 | FFF8 | F700 | 1000 | - | - |
| 5C8 | ABF6 | 5C9 | ABF6 | 5D2 | F400 | 000 | FFF6 | F400 | 1000 | 5BF | 05D2 |
| 5C9 | F204 | 5CE | F204 | 5C9 | F204 | 000 | 0004 | F400 | 1000 | - | - |
| 5CE | 85C0 | 5CF | 85C0 | 5C0 | 0001 | 000 | 0000 | F400 | 1000 | 5C0 | 0001 |
| 5CF | CEF8 | 5C8 | CEF8 | 5CF | 05C8 | 000 | FFF8 | F400 | 1000 | - | - |
| 5C8 | ABF6 | 5C9 | ABF6 | 5D1 | FC00 | 000 | FFF6 | FC00 | 1000 | 5BF | 05D1 |
| 5C9 | F204 | 5CE | F204 | 5C9 | F204 | 000 | 0004 | FC00 | 1000 | - | - |
| 5CE | 85C0 | 5D0 | 85C0 | 5C0 | 0000 | 000 | FFFF | FC00 | 1000 | 5C0 | 0000 |
| 5D0 | 0100 | 5D1 | 0100 | 5D0 | 0100 | 000 | 05D0 | FC00 | 1000 | - | - |
| 5D1 | FC00 | - | - | - | - | - | - | - | - | - | - |
| 5D2 | F400 | - | - | - | - | - | - | - | - | - | - |
| 5D3 | F700 | - | - | - | - | - | - | - | - | - | - |
| 5D4 | F500 | - | - | - | - | - | - | - | - | - | - |
| 5D5 | B5CD | - | - | - | - | - | - | - | - | - | - |

Заключение

Я изучил способы организации циклических программ и исследовал порядок функционирования БЭВМ при выполнении циклических программ и обработки одномерных массивов, изучил принцип работы разных режимов адресации.