Федеральное государственное автономное образовательное учреждение высшего

образования «Национальный исследовательский университет ИТМО»

Факультет программной инженерии и компьютерной техники

**Отчет**

**По лабораторной работе №2**

По дисциплине «Основы профессиональной деятельности»

Вариант 24214.

Выполнил: Чураков А. А., группа P3131

Преподаватель: Остапенко Иван.

Санкт-Петербург

~ 2023 ~

Оглавление

[Задание 3](#_Toc151988324)

[Формула 3](#_Toc151988325)

[ОДЗ: 4](#_Toc151988326)

[Трассировка 4](#_Toc151988327)

[Вариант с меньшим кол-вом команд 4](#_Toc151988328)

[Вывод 4](#_Toc151988329)

# Задание

По выданному преподавателем варианту определить функцию, вычисляемую программой, область представления и область допустимых значений исходных данных и результата, выполнить трассировку программы, предложить вариант с меньшим числом команд. При выполнении работы представлять результат и все операнды арифметических операций знаковыми числами, а логических операций набором из шестнадцати логических значений.

Изображение выглядит как текст, Шрифт, типография

Автоматически созданное описание

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№** | **Адрес** | **Код команды** | **Мнемоника** | **Описание** |
| 1 | 199 | 0100 | - | Переменная **R** (итоговый результат) |
| 2 | 19A | 0100 | - | Переменная **Y** |
| 3 | 19B | A19B | - | Переменная **Z** |
| 4 | 19C | +0200 | CLA | Очистить аккумулятор 0 🡪 AC |
| 5 | 19D | 41A5 | ADD **1A5** | Добавить содержимое ячейки памяти 1A5 к аккумулятору. **AC + (1A5) 🡪 AC** |
| 6 | 19E | 419A | ADD **19A** | Добавить содержимое ячейки памяти 19А к аккумулятору **AC + (19A) 🡪 AC** |
| 7 | 19F | E1A4 | ST **1A4** | Записать содержимое аккумулятора в ячейку памяти с адресом 1A4. **AC 🡪 (1A4)** |
| 8 | 1A0 | A19B | LD **19B** | Загрузить содержимое ячейки 19B в аккумулятор.  **(19B) 🡪 AC** |
| 9 | 1A1 | 31A4 | OR **1A4** | Выполнить операцию логического ИЛИ над содержимым аккумулятора и ячейки памяти 1A4, результат записать в аккумулятор. **M | AC 🡪 AC** |
| 10 | 1A2 | E199 | ST **199** | Записать содержимое аккумулятора в ячейку памяти с адресом 199. **AC 🡪 (199)** |
| 11 | 1A3 | 0100 | HLT | Остановка |
| 12 | 1A4 | A19B | - | Переменная **A** (промежуточный результат) |
| 13 | 1A5 | A19B | - | Переменная **X** |

## Формула

1. X 🡪 AC
2. X + Y 🡪 AC
3. A = X + Y
4. Z 🡪 AC
5. Z | A = Z | (X + Y) 🡪 AC
6. **R = Z | (X + Y)**

## Область представления и ОДЗ:

**Область представления**

**R, Z, A** – набор из 16 логических однобитовых значений

**X, Y** – знаковое 16 разрядное число

-215 <= R <= 215-1

-215 <= A <= 215-1

-215 <= Z <= 215-1

**ОДЗ:**

**-32768 <= X+Y <= 32767**

**-215 <= X+Y <= 215-1**

Рассмотрим случаи

0 <= X <= 215-1

-215 <= Y <= 0  
--------------------  
-215 <= X <= 0

0 <= Y <= 215-1  
----------------------  
-214 <= X <= 215-1

-214 <= Y <= 0  
-----------------------  
-215 <= X <= 214-1

0 <= Y <= 214  
-----------------------  
-215 <= X <= 214

0 <= Y <= 214-1

## Трассировка

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Выполняемая команда** | | **Содержимое регистров процессора после выполнения** | | | | | | | | **Ячейка, содержимое которой изменилось после выполнения команды** | |
| Адрес | Код | IP | CR | AR | DR | SP | BR | AC | NZVC | Адрес | Новый код |
| 19C | 0200 | 19D | 0200 | 19C | 0200 | 000 | 0000 | 0000 | 0000 | - | - |
| 19D | 41A5 | 19E | 41A5 | 1A5 | A19B | 000 | 019D | A19B | 1000 |  |  |
| 19E | 419A | 19F | 419A | 19A | 0100 | 000 | 019E | A29B | 1000 |  |  |
| 19F | E1A4 | 1A0 | E1A4 | 1A4 | A29B | 000 | 019F | A29B | 1000 | 1A4 | A29B |
| 1A0 | A19B | 1A1 | A19B | 19B | A19B | 000 | 01A0 | A19B | 1000 |  |  |
| 1A1 | 31A4 | 1A2 | 31A4 | 1A4 | A29B | 000 | 5C64 | A39B | 1000 |  |  |
| 1A2 | E199 | 1A3 | E199 | 199 | A39B | 000 | 01A2 | A39B | 1000 | 199 | A39B |

## Вариант с меньшим кол-вом команд

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№** | **Адрес** | **Код команды** | **Мнемоника** | **Описание** |
| 1 | 199 | 0100 | - | Переменная **R** (итоговый результат) |
| 2 | 19A | C001 | - | Переменная **Y** |
| 3 | 19B | 0123 | - | Переменная **Z** |
| 4 | 19C | +0200 | CLA | Очистить аккумулятор |
| 5 | 19D | A1A2 | LD **1A2** | Загрузить содержимое ячейки с адресом 1A2 **1A2 🡪 AC** |
| 6 | 19E | 419A | ADD **19A** | Добавить содержимое ячейки памяти 19А к аккумулятору **AC + (19A) 🡪 AC** |
| 7 | 19F | 319B | OR **19B** | Выполнить операцию логического ИЛИ над содержимым аккумулятора и содержимым ячейки памяти с адресом 19B **AC | 19B 🡪 AC** |
| 8 | 1A0 | E199 | ST **199** | Записать содержимое аккумулятора в ячейку памяти с адресом 199. **AC 🡪 (199)** |
| 9 | 1A1 | 0100 | HLT | Остановка |
| 10 | 1A2 | A19B | - | Переменная **X** |

1. 0 🡪 AC
2. 1A2 🡪 AC *(X 🡪 AC)*
3. 1A2+19A 🡪 AC *(X + Y) 🡪 AC*
4. 19B | (1A2 + 19A) 🡪 AC *Z | (X + Y) 🡪 AC*
5. 19B | (1A2 + 19A) 🡪 199 *R = Z | (X + Y)*

Для других чисел X=C000, Y=C001, Z = 123

[C00016]доп = [11000000000000002]доп = -1638410

[C00116]доп = [1100000000000001]доп = -1638310

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Выполняемая команда** | | **Содержимое регистров процессора после выполнения** | | | | | | | | **Ячейка, содержимое которой изменилось после выполнения команды** | |
| Адрес | Код | IP | CR | AR | DR | SP | BR | AC | NZVC | Адрес | Новый код |
| 19C | 019C | 19D | 0000 | 19C | 019C | 000 | 0000 | 0000 | 0000 | 19C | 019C |
| 19D | A1A2 | 19E | A1A2 | 1A2 | C000 | 000 | 019D | C000 | 1000 |  |  |
| 19E | 419A | 19F | 419A | 19A | C001 | 000 | 019E | 8001 | 1001 |  |  |
| 19F | 319B | 1A0 | 319B | 19B | 0123 | 000 | 7EDC | 8123 | 1001 |  |  |
| 1A0 | E199 | 1A1 | E199 | 199 | 8123 | 000 | 01A0 | 8123 | 1001 | 199 | 8123 |
| 1A1 | 0100 | 1A2 | 0100 | 1A1 | 0100 | 000 | 01A1 | 8123 | 1001 |  |  |

# Вывод

В ходе данной лабораторной работы я познакомился с базовой ЭВМ и командами. Я научился манипулировать памятью ЭВМ и исполнять базовые программы.

**ADD:**

1. Instruction Fetch

IP -> BR, AR

BR + 1 -> IP, MEM(AR) -> DR

DR -> CR

1. Operand Fetch

DR -> AR

MEM(AR) -> DR

1. Execution

       (via ALU):AC + DR -> AC, N, Z, V, C

**LD:**

1. Instruction Fetch

IP -> BR, AR

BR + 1 -> IP, MEM(AR) -> DR

DR -> CR

1. Operand Fetch

DR -> AR

MEM(AR) -> DR

1. Execution

DR -> AC, N, Z, V, C

**OR**:

1. Instruction Fetch

IP -> BR, AR

BR + 1 -> IP, MEM(AR) -> DR

DR -> CR

1. Operand Fetch

DR -> AR

MEM(AR) -> DR

1. Execution

^AC & ^DR -> BR

^BR -> AC, N, Z, V

**CLA:**

1. Instruction Fetch

IP -> BR, AR

BR + 1 -> IP, MEM(AR) -> DR

DR -> CR

1. Operand Fetch

Ничего

1. Execution

0 -> AC

**ST:**

1. Instruction Fetch

IP -> BR, AR

BR + 1 -> IP, MEM(AR) -> DR

DR -> CR

1. Operand Fetch

Тоже ничего

1. Execution

DR -> AR

AC -> DR

DR -> MEM(AR)