Факультет Программной Инженерии и Компьютерной Техники

Лабораторная работа №2

Исследование работы БЭВМ

Вариант 317962

Выполнил:

Петров Вячеслав Маркович

Группа P3108

Принял:

Остапенко Ольга Денисовна

Содержание

[Задание 3](#_Toc150723544)

[Функция 4](#_Toc150723545)

[ОП и ОДЗ 4](#_Toc150723546)

[Область представления: 4](#_Toc150723547)

[Область допустимых значений 4](#_Toc150723548)

[Трассировка программы 4](#_Toc150723549)

[Вариант с меньшим числом команд 5](#_Toc150723550)

[Вывод 6](#_Toc150723551)

# 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

Автоматически созданное описание

|  |  |  |  |
| --- | --- | --- | --- |
| Адрес | Код команды | Мнемоника | Комментарии |
| 066 | 0200 | - | Переменная A |
| 067 | 6068 | - | Переменная B |
| 068 | 2081 | - | Переменная C  (промежуточный результат) |
| 069 | E068 | - | Переменная D |
| 06A | 0200 | **CLA** | Очистить аккумулятор:  0 => AC |
| 06B | 3067 | **OR 067** | Выполнить операцию логического «ИЛИ» над содержимым ячейки памяти **067** и аккумулятором, результат записать в аккумулятор:  ^(^AC & ^(**067**)) => AC |
| 06C | 3069 | **OR 069** | Выполнить операцию логического «ИЛИ» над содержимым ячейки памяти **069** и аккумулятором, результат записать в аккумулятор:  ^(^AC & ^(**069**)) => AC |
| 06D | E068 | **ST 068** | Сохранить содержимое аккумулятора в ячейку памяти (**068**):  AC => (**068**) |
| 06E | A066 | **LD 066** | Загрузить содержимое ячейки **066** в аккумулятор:  (**066**) => AC |
| 06F | 6068 | **SUB 068** | Выполнить операцию вычитания содержимого ячейки памяти (**068**) из аккумулятора:  AC - (**068**) => AC |
| 070 | E068 | **ST 068** | Сохранить содержимое аккумулятора в ячейку памяти (**068**):  AC => (**068**) |
| 071 | 0200 | **CLA** | Очистить аккумулятор:  0 => AC |
| 072 | 0280 | **NOT** | Инвертировать содержимое аккумулятора:  ^AC => AC |
| 073 | 2081 | **AND 081** | Выполнить операцию логического «ИЛИ» над содержимым ячейки памяти **081** и аккумулятором, результат записать в аккумулятор:  AC & (**081**) => AC |
| 074 | 3068 | **OR 068** | Выполнить операцию логического «ИЛИ» над содержимым ячейки памяти **068** и аккумулятором, результат записать в аккумулятор:  ^(^AC & ^(**068**)) => AC |
| 075 | E068 | **ST 068** | Сохранить содержимое аккумулятора в ячейку памяти (**068**):  AC => (**068**) |
| 076 | 0200 | **CLA** | Очистить аккумулятор:  0 => AC |
| 077 | 6080 | **SUB 080** | Выполнить операцию вычитания содержимого ячейки памяти (**080**) из аккумулятора:  AC - (**080**) => AC |
| 078 | 6068 | **SUB 068** | Выполнить операцию вычитания содержимого ячейки памяти (**068**) из аккумулятора:  AC - (**068**) => AC |
| 079 | E068 | **ST 068** | Сохранить содержимое аккумулятора в ячейку памяти (**068**):  AC => (**068**) |
| 07A | A07E | **LD 07E** | Загрузить содержимое ячейки **07E** в аккумулятор:  (**07E**) => AC |
| 07B | 3068 | **OR 068** | Выполнить операцию логического «ИЛИ» над содержимым ячейки памяти **068** и аккумулятором, результат записать в аккумулятор:  ^(^AC & ^(**068**)) => AC |
| 07C | E07F | **ST 07F** | Сохранить содержимое аккумулятора в ячейку памяти (**07F**):  AC => (**07F**) |
| 07D | 0100 | **HLT** | Останов |
| 07E | 6068 | - | Переменная E |
| 07F | 3068 | - | Итоговый результат R |
| 080 | 6068 | - | Переменная F |
| 081 | 2081 | - | Переменная G |

# Функция

R = ((0000 0000 0000 00002 – F) - (A – (B ∨ D) ∨ (1111 1111 1111 11112 & G))) ∨ E

Упростим:

R = (– F – ((A – (B ∨ D)) ∨ G)) ∨ E

# ОП и ОДЗ

## Область представления:

-R- знаковое, 16-ти разрядное число

-A, C- знаковое, 16-ти разрядное

-B, D – набор из 16 однобитных значений

-(A & B)- знаковое, 16-ти разрядное число

-A - (A & B) – знаковое, 16-ти разрядное

Для логических операций: [0;65535]

Для арифметических операций: [-32768;32767]

## Область допустимых значений

# Трассировка программы

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Выполняемая  команда | | Содержимое регистров после выполнения команды | | | | | | | | | Ячейка, содержимое которой изменилось после выполнения команды | |
| Адрес | Код  Команды | IP | CR | AR | DR | SP | BR | AC | PS | NZVC | Адрес | Новый  код |
| 123 | E126 | - | - | - | - | - | - | - | - | - | - | - |
| 124 | 4124 | - | - | - | - | - | - | - | - | - | - | - |
| 125 | A125 | - | - | - | - | - | - | - | - | - | - | - |
| 126 | 0100 | - | - | - | - | - | - | - | - | - | - | - |
| 06A | +0200 | 06A | 0000 | 000 | 0000 | 000 | 0000 | 0000 | 004 | 0100 |  |  |
| 06A | 0200 | 06B | 0200 | 06A | 0200 | 000 | 006A | 0000 | 004 | 0100 |  |  |
| 06B | 3067 | 06C | 3067 | 067 | 6068 | 000 | 9F97 | 6068 | 000 | 0000 |  |  |
| 06C | 3069 | 06D | 3069 | 069 | E068 | 000 | 1F97 | E068 | 008 | 1000 |  |  |
| 06D | E068 | 06E | E068 | 068 | E068 | 000 | 006D | E068 | 008 | 1000 | 068 | E068 |
| 06E | A066 | 06F | A066 | 066 | 0200 | 000 | 006E | 0200 | 000 | 0000 |  |  |
| 06F | 6068 | 070 | 6068 | 068 | E068 | 000 | 006F | 2198 | 000 | 0000 |  |  |
| 070 | E068 | 071 | E068 | 068 | 2198 | 000 | 0070 | 2198 | 000 | 0000 | 068 | 2198 |
| 071 | 0200 | 072 | 0200 | 071 | 0200 | 000 | 0071 | 0000 | 004 | 0100 |  |  |
| 072 | 0280 | 073 | 0280 | 072 | 0280 | 000 | 0072 | FFFF | 008 | 1000 |  |  |
| 073 | 2081 | 074 | 2081 | 081 | 2081 | 000 | 0073 | 2081 | 000 | 0000 |  |  |
| 074 | 3068 | 075 | 3068 | 068 | 2198 | 000 | DE66 | 2199 | 000 | 0000 |  |  |
| 075 | E068 | 076 | E068 | 068 | 2199 | 000 | 0075 | 2199 | 000 | 0000 | 068 | 2199 |
| 076 | 0200 | 077 | 0200 | 076 | 0200 | 000 | 0076 | 0000 | 004 | 0100 |  |  |
| 077 | 6080 | 078 | 6080 | 080 | 6068 | 000 | 0077 | 9F98 | 008 | 1000 |  |  |
| 078 | 6068 | 079 | 6068 | 068 | 2199 | 000 | 0078 | 7DFF | 003 | 0011 |  |  |
| 079 | E068 | 07A | E068 | 068 | 7DFF | 000 | 0079 | 7DFF | 003 | 0011 | 068 | 7DFF |
| 07A | A07E | 07B | A07E | 07E | 6068 | 000 | 007A | 6068 | 001 | 0001 |  |  |
| 07B | 3068 | 07C | 3068 | 068 | 7DFF | 000 | 8200 | 7DFF | 001 | 0001 |  |  |
| 07C | E07F | 07D | E07F | 07F | 7DFF | 000 | 007C | 7DFF | 001 | 0001 | 07F | 7DFF |
| 07D | 0100 | 07E | 0100 | 07D | 0100 | 000 | 007D | 7DFF | 001 | 0001 |  |  |

# Вариант с меньшим числом команд

Программа:

123: E126

124 : 4124

125 : A125

126 : + A125

127 : 212E

128 : E123

129: A124

12A: 6126

12B: E123

12C: 0100

12D: 0100

|  |  |  |  |
| --- | --- | --- | --- |
| Адрес | Код команды | **Мнемоника** | Комментарии |
| 123 | E126 | **-** | Итоговый результат R |
| 124 | 4124 | **-** | Переменная A |
| 125 | A125 | **-** | Переменная B |
| 126 | A125 | **LD 125** | Загрузить содержимое ячейки памяти **125** в аккумулятор:  (**125**) => AC |
| 127 | 212E | **AND 12E** | Выполнить операцию логического «И» над содержимым ячейки памяти **12D** и аккумулятором, результат записать в аккумулятор:  AC & (**12D**) => AC |
| 128 | E126 | **ST 123** | Сохранить содержимое аккумулятора в ячейку памяти (**123**):  AC => (**123**) |
| 129 | A124 | **LD 124** | Загрузить содержимое ячейки памяти **124** в аккумулятор:  (**124**) => AC |
| 12A | 6126 | **SUB 126** | Выполнить операцию вычитания содержимого ячейки памяти (**126**) из аккумулятора:  AC – (**126**) => AC |
| 12B | E123 | **ST 123** | Сохранить содержимое аккумулятора в ячейку памяти (**123**):  AC => (**123**) |
| 12C | 0100 | **HLT** | Останов |
| 12D | 0100 | **-** | Переменная D |

Предложенный мной вариант помогает сэкономить две ячейки памяти.

# Вывод

В ходе работы над лабораторной работой я познакомился со структурой БЭВМ, узнал, как устроены и связаны его основные элементы, научился определять ОДЗ, узнал структуру и виды команд, как представлены данные в памяти БЭВМ, написал свою программу, эквивалентную по выполнению заданной, тем самым сэкономив две ячейки памяти.