Федеральное государственное автономное образовательное учреждение высшего образования «Национальный исследовательский университет ИТМО»

Факультет Программной инженерии и компьютерной техники

Дисциплина “Основы профессиональной деятельности”

Отчёт по лабораторной работе №2

Вариант №7347

Выполнил:

Ануфриев Андрей Сергеевич, Р3119

Проверила:

Остапенко Ольга Денисовна

г. Санкт-Петербург

2024 год

Оглавление

[Задание 2](#_Toc184213375)

[Ход выполнения 2](#_Toc184213376)

[1) Текст исходной программы 2](#_Toc184213377)

[2) Описание программы 2](#_Toc184213378)

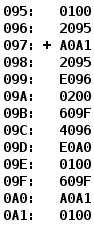
[3) Таблица трассировки 2](#_Toc184213379)

[4) Вариант программы с меньшим числом команд 2](#_Toc184213380)

[Вывод 2](#_Toc184213381)

# Задание

1. 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.



1. Восстановить текст заданного варианта программы, отделить ячейки данных от кода программы, написать назначение программы и реализуемую функцию, которую представить в виде формулы.
2. Во время допуска к работе получить у преподавателя исходные данные для переменных, согласовать вариант программы для исполнения, занести в память базовой ЭВМ заданный вариант программы и, выполняя ее по командам, заполнить таблицу трассировки выполненной программы. Занесение программы с данными, а также запуск программы в пультовом режиме продемонстрировать преподавателю.

# Ход выполнения

## Текст исходной программы

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Адрес | Код | мнем | Что происходит | Комментарии |
| 095 = 1001 0101 | 0100 = a | число |  | Переменная a |
| 096 = 1001 0110 | 2095 = b | число |  | Переменная b |
| 097+= 1001 0111 | A0A1 | **LD 0A1** | Загрузка | 0A1 => AC |
| 098 = 1001 1000 | 2095 | **AND 095** | Логическое умножение | 095 & AC => AC |
| 099 = 1001 1001 | E096 | **ST 096** | сохранение | AC => 096 |
| 09A = 1001 1010 | 0200 | **CLA** | Очистка аккумулятора | 0 => AC |
| 09B = 1001 1011 | 609F | **SUB 09F** | вычитание | AC – 09F => AC |
| 09C = 1001 1100 | 4096 | **ADD 096** | сложение | 096 + AC => AC |
| 09D = 1001 1101 | E0A0 | **ST 0A0** | сохранение | AC=> 0A0 |
| 09E = 1001 1110 | 0100 | **HLT** | останов |  |
| 09F = 1001 1111 | 609F = c | число |  | Переменная c |
| 0A0 = 1010 0000 | A0A1 = d | число |  | Переменная d |
| 0A1 = 1010 0001 | 0100 = e | число |  | Переменная e |
|  |  |  |  |  |

## Описание программы

**По шагам:**

AC = 0A1 = 1010 0001

AC = a(095)&AC = a(095) & e(0A1) = 1001 0101 & 1010 0001

AC => b(096)

b(096) = a(095) & e(0A1)

0 => AC; AC = 0

AC = AC – c(09F) = 0 – (1001 1111)

AC = b(096) + AC = (a(095) & e(0A1)) + (0 - c(09F))

d(0A0) = AC = (a(095) & e(0A1)) + (0 - c(09F))

**Итоговая формула:**

d = (a И e) – с

**Область представления:**

d, c - знаковое, 16-ти разрядное [-215; 215-1]

a, e – логическое значение [0; 216-1]

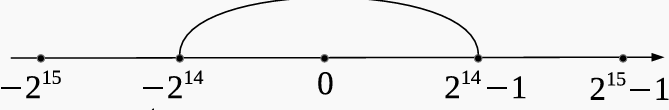
(a И e) - логическое значение [0; 216-1]

**ОДЗ:**

Случай 1

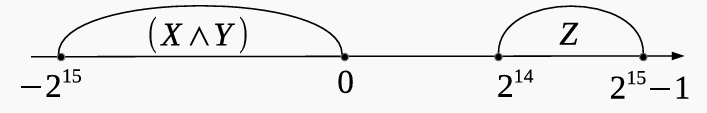
-214 ≤ -c ≤ 214-1

-214 ≤ (a И e) ≤ 214-1



Случай 2

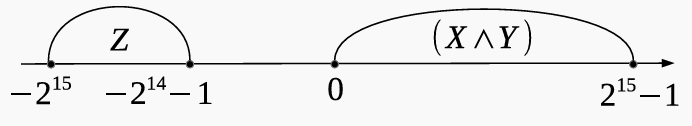
214 ≤ -c ≤ 215-1 (A И E) C



Случай 3

-215 ≤- с ≤ -214-1

C



**Расположение в памяти ЭВМ программы, исходных данных и результатов**

097 – 09E – программа

0A1, 095, 09F - исходный данные

096 – промежуточный результат

0A1 - итоговый результат

**Адреса первой и последней выполняемой команд программы**

097 – адрес первой команды

09E – адрес последней команды

## Таблица трассировки

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Выполняемая команда** | | **Содержимое регистров процессора после выполнения команды** | | | | | | | | **изменённая ячейка** | |
| Адрес | Код | IP | CR | AR | DR | SP | BR | AC | NZVC | Адрес | Новый код |
| 095 | 0100 | - | - | - | - | - | - | - | - | - | - |
| 096 | 2095 | - | - | - | - | - | - | - | - | - | - |
| 097 | A0A1 | 098 | A0A1 | 0A1 | 0100 | 000 | 097 | 0100 | 0000 |  |  |
| 098 | 2095 | 099 | 2095 | 095 | 0100 | 000 | 098 | 0081 | 0000 |  |  |
| 099 | E096 | 09A | E096 | 096 | 0081 | 000 | 099 | 0081 | 0000 | 096 | 0081 |
| 09A | 0200 | 09B | 0200 | 09A | 0200 | 000 | 09A | 0000 | 0100 |  |  |
| 09B | 609F | 09C | 609F | 09F | 609F | 000 | 09B | 4FB0 | 0000 |  |  |
| 09C | 4096 | 09D | 4096 | 096 | 0081 | 000 | 09C | 5013 | 0000 |  |  |
| 09D | E0A0 | 09E | E0A0 | 0A0 | 5013 | 000 | 09D | 5013 | 0000 | 0A0 | 5013 |
| 09E | 0100 | 09F | 0100 | 09E | 0100 | 000 | 09E | 5013 | 0000 |  |  |
| 09F | 609F | - | - | - | - | - | - | - | - | - | - |
| 0A0 | A0A1 | - | - | - | - | - | - | - | - | - | - |
| 0A1 | 0100 | - | - | - | - | - | - | - | - | - | - |

095 a 0100 w 2095 w A0A1 w 2095 w E096 w 0200 w 609F w 4096 w E0A0 w 0100 w 609F w A0A1 w 0100 w

|  |  |  |  |
| --- | --- | --- | --- |
| Адрес | Код | Мнем | Комментарии |
| 001 | 0100 | Число | Переменная e |
| 002 | 0100 | Число | Переменная a |
| 003 | 2095 | Число | Переменная b |
| 004 | A0A1 | число | Переменная d |
| 005 | A001 | LD 001 | 001=>AC |
| 006 | 2002 | AND 002 | 002 & AC => AC |
| 007 | 6003 | SUB 003 | AC – 003=>AC |
| 008 | E004 | ST 004 | AC=>004 |
| 009 | 0100 | HLT | Остановка программы, откл тг |

## Вариант программы с меньшим числом команд

**По шагам:**

AC = e

AC = AC & a

AC = AC – b

d = AC

**Программа**:

001 0100

002 0100

003 2095

004 A0A1

005+ A001

006 2002

007 6003

008 E004

009 0100

# Доп 1

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Выполняемая команда** | | **Содержимое регистров процессора после выполнения команды** | | | | | | | | **изменённая ячейка** | |
| Адрес | Код | IP | CR | AR | DR | SP | BR | AC | NZVC | Адрес | Новый код |
| 095 | 7FFF | - | - | - | - | - | - | - | - | - | - |
| 096 | 0000 | - | - | - | - | - | - | - | - | - | - |
| 097 | A0A1 | 098 | A0A1 | 0A1 | 0000 | 000 | 0097 | 7FFF | 0000 |  |  |
| 098 | 2095 | 099 | 2095 | 095 | 0000 | 000 | 0098 | 7FFF | 0000 |  |  |
| 099 | E096 | 09A | E096 | 096 | 0000 | 000 | 0099 | 7FFF | 0000 | 096 | 7FFF |
| 09A | 0200 | 09B | 0200 | 09A | 0200 | 000 | 009A | 0000 | 0100 |  |  |
| 09B | 609F | 09C | 609F | 09F | FFFE | 000 | 009B | 0010 | 0000 |  |  |
| 09C | 4096 | 09D | 4096 | 096 | FFFE | 000 | 009C | 8001 | 1010 |  |  |
| 09D | E0A0 | 09E | E0A0 | 0A0 | 0000 | 000 | 009D | 8001 | 1010 | 0A0 | Невозможно ранить 8001 |
| 09E | 0100 | 09F | 0100 | 09E | 0100 | 000 | 009E | 8001 | 1010 |  |  |
| 09F | FFFE | - | - | - | - | - | - | - | - | - | - |
| 0A0 | 0000 | - | - | - | - | - | - | - | - | - | - |
| 0A1 | 7FFF | - | - | - | - | - | - | - | - | - | - |

(FFFF & FFFF) – FFFF =

= FFFF + (-FFFF) =

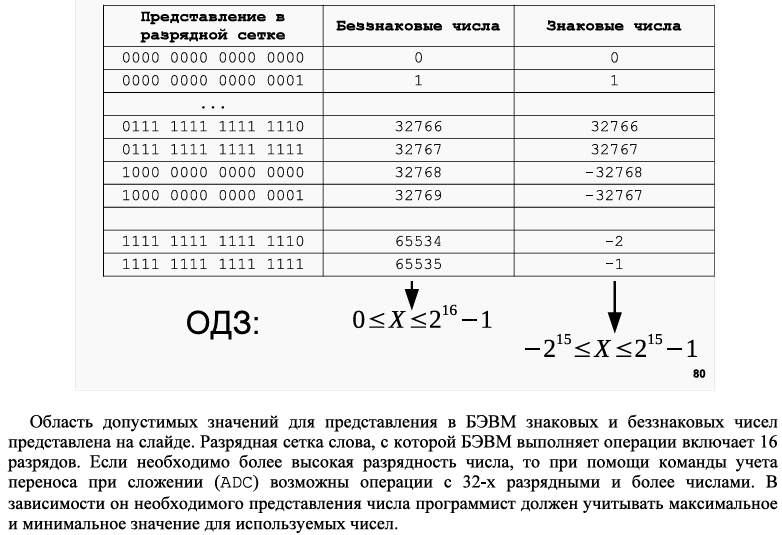
= FFFF + 0000 + 1 = 0000 и переполнение

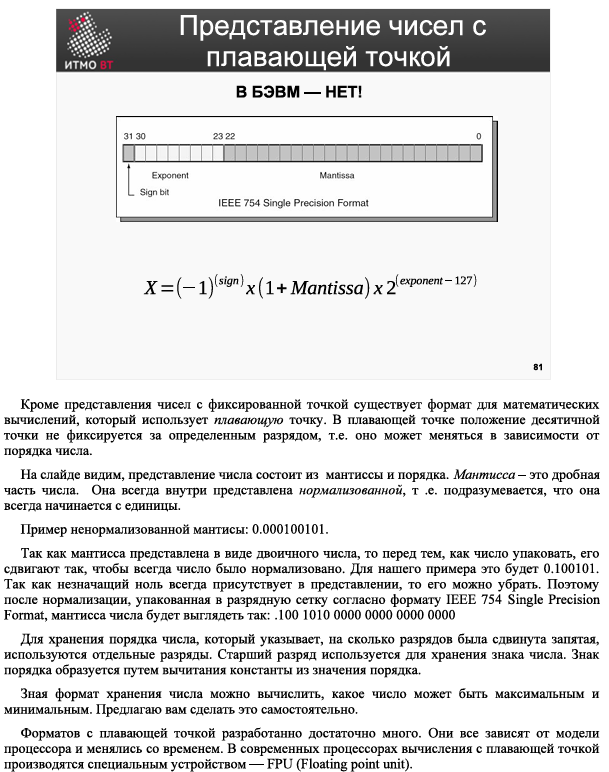
# Вывод

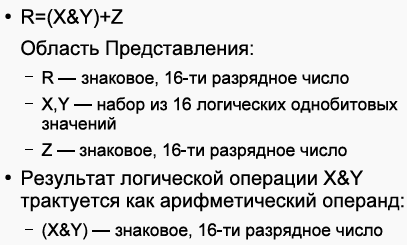
В ходе работы а понял как работает БЭВМ, узнал какие есть команды и на практике попробовал понять какую формулу реализует программа.Научился определять ОП и ОДЗ данных.

# Ответы на вопросы

* + - 1. Форматы представления, область представления область допустимых значений в БЭВМ для знаковых и беззнаковых чисел с фиксированной точкой.

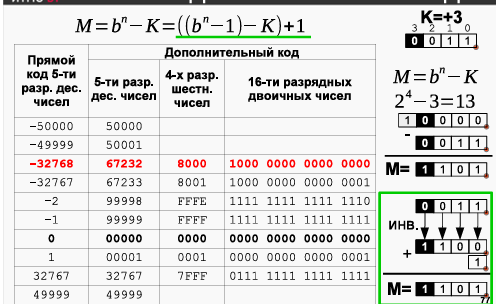






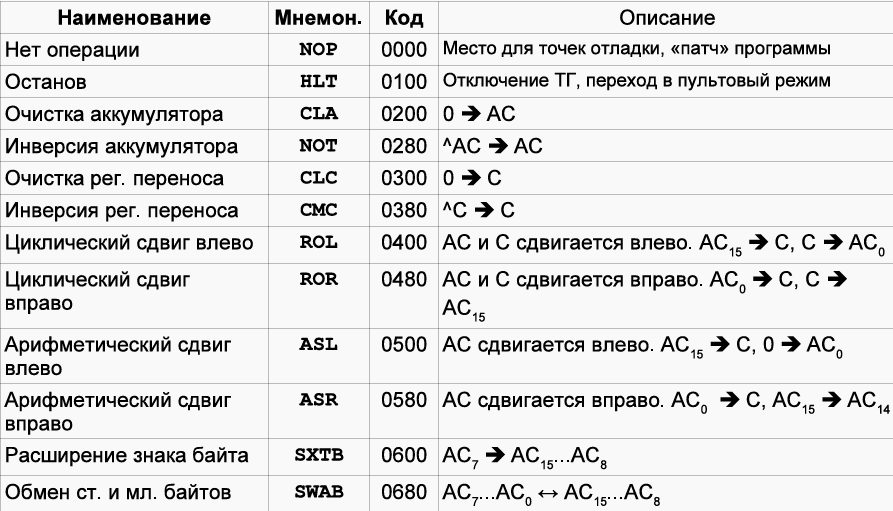
* + - 1. Представление чисел в разрядной сетке в прямом, обратном и дополнительном кодах.



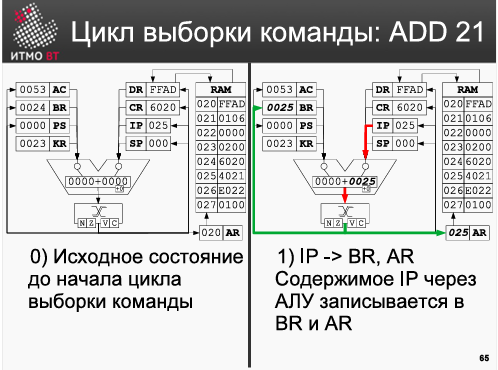
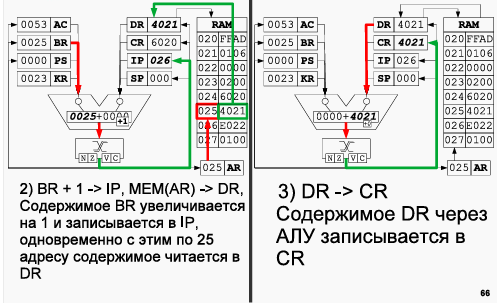
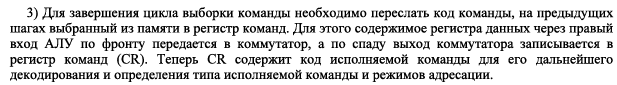


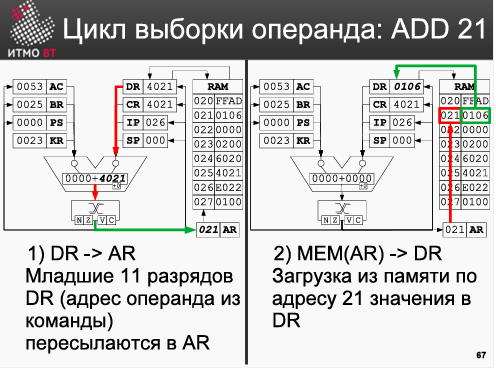
* + - 1. Адресные и безадресные команды БЭВМ.





* + - 1. Описание команды, находящейся по указанному адресу: наименование, назначение, тип команды и вид адресации. Кол-во и название машинных циклов, потактовое выполнение команд.





* + - 1. Потактовое выполенение некоторых команд

**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

**SUB:**

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) + 1) -> 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

**AND:**

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 -> 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

**SWAM:**

1. Instruction Fetch

IP -> BR, AR

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

DR -> CR

1. Operand Fetch

Вроде как ничего, но не уверен

1. Execution

DR - > BR

AC -> DR

BR -> AC, N, Z, V, DR -> MEM(AR) -> DR

**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)

* + - 1. Регистры

**Регистры:**

AC - аккумулятор, регистр общего назначения, в основном команды кладут свои результаты именно туда. 16-разрядный.

BR - буферный регистр(вспомогательный регистр для промежуточных данных, используется командами, чьи вычисления требует несколько этапов). 16-разрядный.

PS(Program State) - регистр состояния процессора, туда кидаются в том числе флаги NZVC. 9-разрядный.

IR(Input Register) - клавишный регистр, туда попадает то, что мы вводим с пульта. 16-разрядный.

DR(Data Register) - регистр данных, может записывать данные в ячейки RAM или наоборот брать содержимое ячеек из RAM. 16-разрядный.

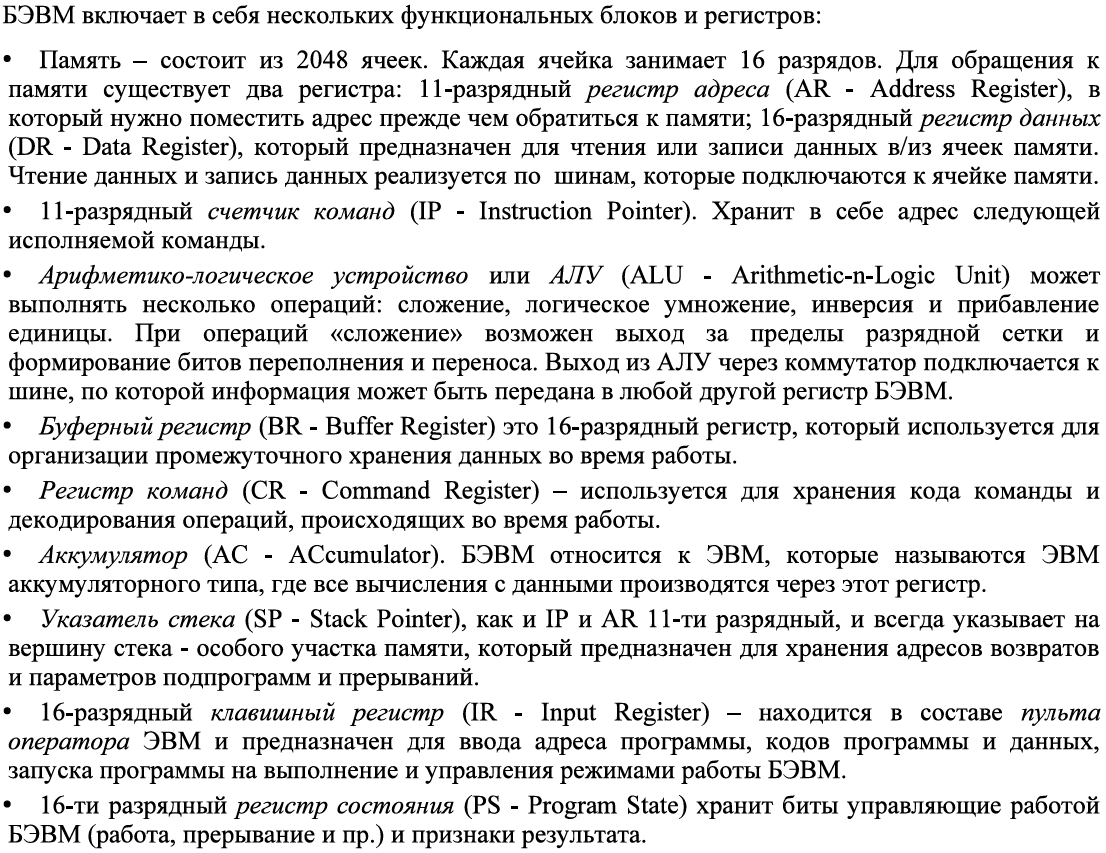
CR(Command Register) - регистр комманд, содержит текущую исполняемую команду.  16-разрядный.

IP(Instruction Pointer) - счетчик команд, содержит адрес инструкции, которая будет исполняться следующей, по умолчанию инкрементируется на каждом шаге. 11-разрядный.

SP(Stack Pointer) - указатель стека, содержит адрес вершины стека, используется для некоторых команд, нужен для передачи аргументов в подпрограммы. 11-разрядный.

AR(Address Register) - регистр адреса, работает с RAM, указывая переданный из IP адрес ячейки, с которой мы сейчас работаем. 11-разрядный.

В регистре информация хранится на триггерах.



* + - 1. Циклы команд

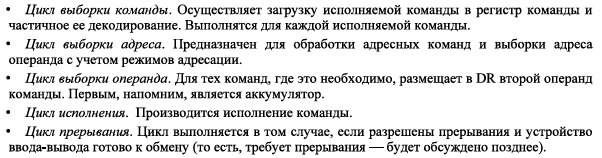
1. Instruction Fetch - цикл выборки команды. Обычно, код команды извлекается из памяти и помещается в регистр команд (CR). Счётчик команд (IP) инкрементируется при помощи буферного регистра.

2. Address Fetch - цикл выборки адреса. Существует только для адресных команд. Всегда заканчивается выборкой адреса операнда(Клименков сказал пока не забивать им голову, так что подробнее об этом цикле будет UPD позже)

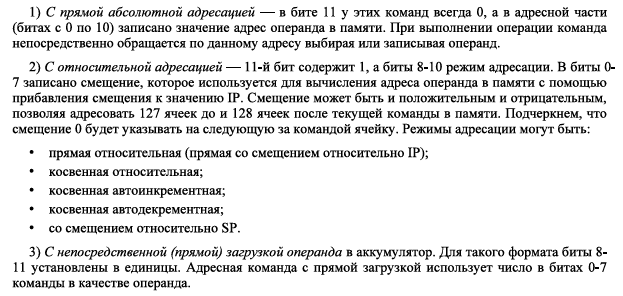
3. Operand Fetch - цикл выборки операнда. Для адресных команд помещает данные из ячейки в регистр данных (DR) для последующих операций.

4. Execution - цикл исполнения. Различен для каждой команды. Подробнее о циклах исполнения команд можно узнать через java -jar -Dmode=decoder bcomp-ng.jar

5. Interruption - цикл прерывания. Нужен для работы с внешними устройствами.

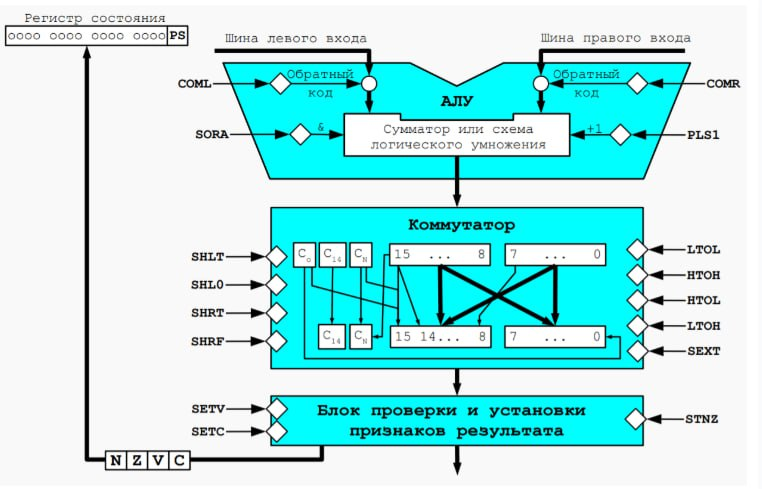


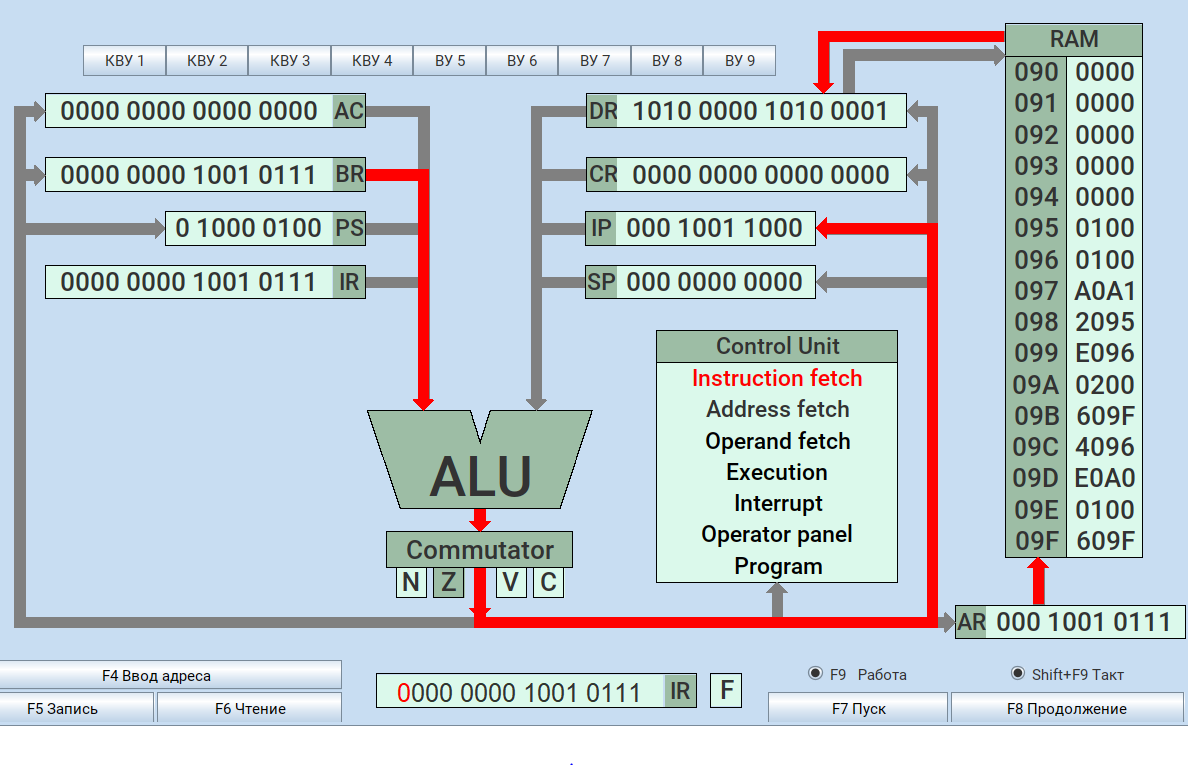
* + - 1. Адресация



* + - 1. Какую формулу реализует программа? Как можно её упростить?
      2. Где находится результат? Как они представлены? Какие дополнительные ячейки использует программа?
      3. Какое кол-во обращений к ячейкам памяти при выполнении безадресной команды? На каких циклах оно выполняется?

Обращение к ячейкам памяти при выполнении безадресной команды выполняется **на первом этапе цикла выполнения команды** — при выборке исполняемой команды из ОЗУ.





e(0A1)&a(095)=> b(096)

0-c(09F)+b(096)=>d(0A0)

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Выполняемая команда** | | **Содержимое регистров процессора после выполнения команды** | | | | | | | | **изменённая ячейка** | |
| Адрес | Код | IP | CR | AR | DR | SP | BR | AC | NZVC | Адрес | Новый код |
| 00A | 7FFF | - | - | - | - | - | - | - | - | - | - |
| 00B | 0000 | - | - | - | - | - | - | - | - | - | - |
| 00D | A0A1 | 098 | A0A1 | 0A1 | 0000 | 000 | 0097 | 7FFF | 0000 |  |  |
| 00E | 2095 | 099 | 2095 | 095 | 0000 | 000 | 0098 | 7FFF | 0000 |  |  |
| 010 | E096 | 09A | E096 | 096 | 0000 | 000 | 0099 | 7FFF | 0000 | 096 | 7FFF |
| 011 | 0200 | 09B | 0200 | 09A | 0200 | 000 | 009A | 0000 | 0100 |  |  |
| 012 | 609F | 09C | 609F | 09F | FFFE | 000 | 009B | 0010 | 0000 |  |  |
| 09C | 4096 | 09D | 4096 | 096 | FFFE | 000 | 009C | 8001 | 1010 |  |  |
| 09D | E0A0 | 09E | E0A0 | 0A0 | 0000 | 000 | 009D | 8001 | 1010 | 0A0 | Невозможно ранить 8001 |
| 09E | 0100 | 09F | 0100 | 09E | 0100 | 000 | 009E | 8001 | 1010 |  |  |
| 09F | FFFE | - | - | - | - | - | - | - | - | - | - |
| 0A0 | 0000 | - | - | - | - | - | - | - | - | - | - |
| 0A1 | 7FFF | - | - | - | - | - | - | - | - | - | - |

