Министерство науки и высшего образования Российской Федерации Федеральное государственное бюджетное образовательное учреждение высшего образования

«Рязанский государственный радиотехнический университет

имени В.Ф. Уткина»

Кафедра «Электронные вычислительные машины»

Отчет о лабораторной работе № 1

**Структура процессора и состав микрокоманд**

по дисциплине

«ЭВМ и периферийные устройства»

**Выполнили:**

 студенты группы 445

бригада № 1

Андрюшин М.А.

Гукало Д.Г.

**Проверили:**

к.т.н., доц. каф. ЭВМ

Устюков Д.И.

ст. пр. каф. ЭВМ

Кижаев О.В.

 Рязань 2025

Цель работы

Изучение процессора на уровне структурной схемы, ознакомление со структурой микрокоманд (МК) и порядком ввода данных, кодирование и выполнение МК.

Практическая часть

1) В состав микропрограммируемого процессора (МП) входят операционный блок, блок микропрограммного управления (БМУ), оперативная память (RAM) и микропрограммная память (MPM). Структурная схема микропрограммируемого процессора приведена на рисунке 1;

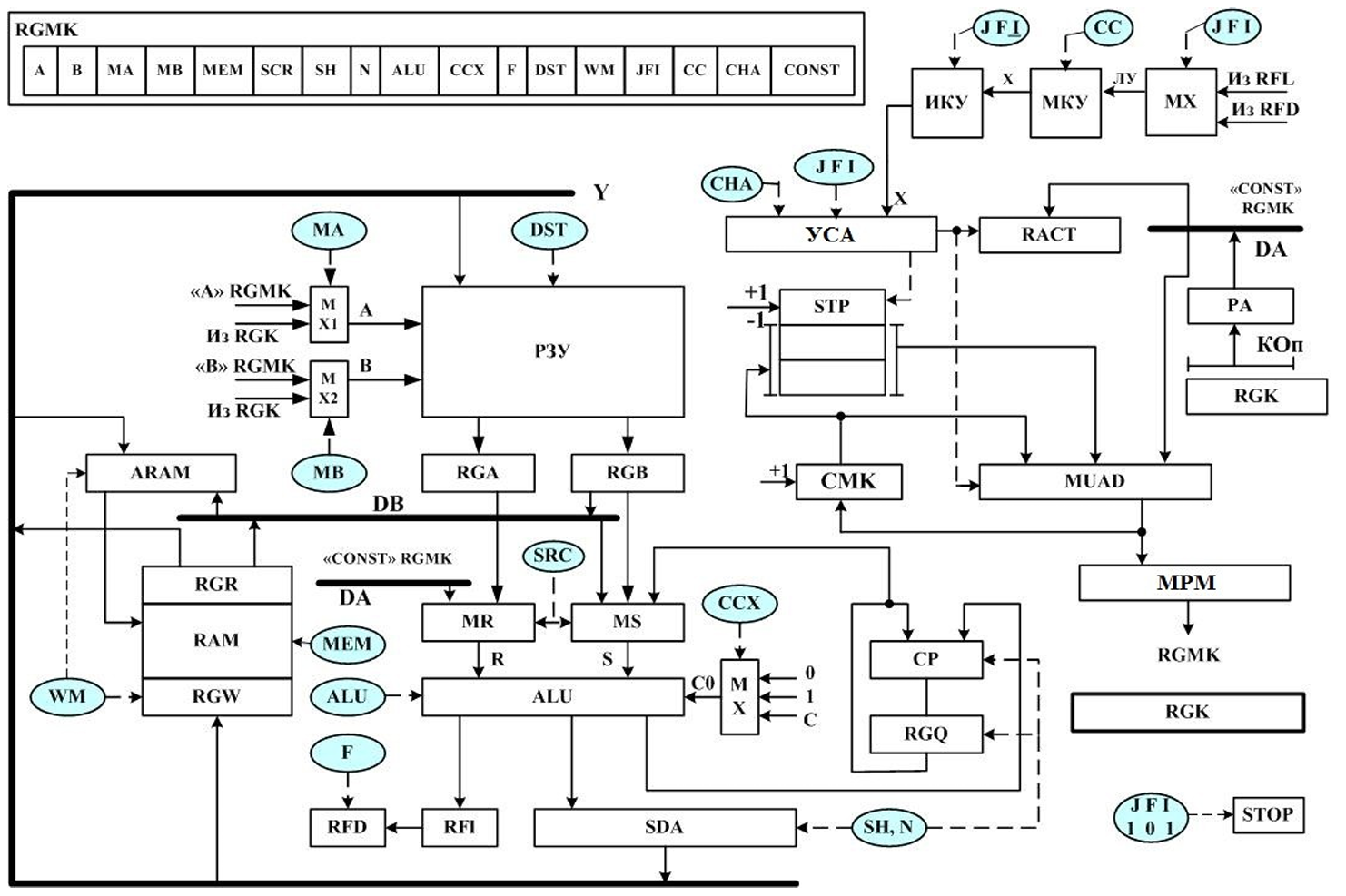


Рисунок 1 — Структурная схема процессора

2) структура микрокоманды и описание полей. Микрокоманда содержит 64 бит и включает в себя 17 полей. Структура микрокоманды представлена в табл. 1.

Таблица 1— Структура микрокоманды

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Поле | A | B | МА | MB | МЕМ | SRC | SH | N | ALU |
| Бит | 4 | 4 | 2 | 2 | 3 | 3 | 4 | 4 | 4 |
| Def | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 6 |

Продолжение таблицы 1

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Поле | CCX | F | DST | WM | JFI | СС | СНА | CONST |
| Бит | 2 | 1 | 3 | 2 | 3 | 3 | 4 | 16 |
| Def | 0 | 0 | 0 | 0 | 0 | 0 | 7 | 0 |

* Поля А и В задают адреса регистров РЗУ.
* Поля МА и МВ управляют выбором адресов РЗУ из микрокоманды или регистра команд RGK. По умолчанию МА=МВ=0, что соответствует выбору адресов из микрокоманды.
* Поле МЕМ управляет чтением и записью памяти. Оно состоит из трех бит. Первый бит - признак обращения к памяти, второй - определяет режим чтение/запись, третий - размер данных (байт или слово).
* Поле SRC управляет выбором источников операндов. По умолчанию SRC=1, что соответствует выбору операндов из RGA и RGB.
* Поле SH управляет работой сдвигателей SDA и CP. Поле N указывает количество разрядов, на которое надо сдвигать содержимое SDA и/или CP. По умолчанию SH=0 и N=0 - передача с выхода ALU на шину Y без сдвига.
* Поле ALU управляет операциями АЛУ. По умолчанию ALU=6, что означает сложение операнда R с входным переносом С0, определяемым полем CCX.
* Поле ССХ управляет входным переносом. По умолчанию значение ССХ=0.
* Поле F - поле фиксации флажков. При значении F=1 текущее значение флажков заносится в RFD.
* Поле DST управляет записью данных с выходов SDA и RGR в РЗУ по адресу В. По умолчанию DST=0 (без записи).
* Поле WM управляет записью в память.
* Поле JFI участвует совместно с полем СС в формировании условий перехода. Старший бит J=1 - признак безусловного перехода, бит F указывает, что условие перехода определяется флажками из RFD, бит I означает инверсию формируемого условия. При JFI=101 происходит остановка работы и формирование признака STOP. По умолчанию JFI=0.
* Поле СС управляет формированием условий перехода. Обозначения переходов соответствуют условным переходам МП i8086.
* Поле СНА обеспечивает формирование адреса следующей микрокоманды и содержит набор инструкций схемы УПМ.
* Поле CONST содержит 16-битовую константу, подключаемую к шине DA операционного блока, или адрес перехода, подаваемый на шину DA блока микропрограммного управления. По умолчанию CONST=0.

**Задание**

Написать и отладить микропрограмму, которая вычисляет следующее выражение: (AX – BX) / 16 & [100h] + 31h → [100h]

Изображение выглядит как снимок экрана, текст, Шрифт, число

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 2 — Микропрограммная память

Результаты выполнения задания представлены в таблицах 3 - 7.

1) Трассировка работы программы в таблице 3 представлена для набора исходных данных №1:

[100h] = 0xAAAA

AX = 0x0100

BX = 0x0010

Таблица 3 — Трассировка программы №1

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| СМК | РЗУ | | | | RGA | RGB | ALU | SDA | RFI | | | | | | RGR | RGW | ARAM |
| AX | BX | CX | DX | N | Z | V | C | P | M |
| 000 | 0100 | 0010 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0 | 0 | 0 | 0 | 0 | 0 | 0000 | 0000 | 0000 |
| 001 | 0100 | 000F | 0000 | 0000 | 0100 | 0010 | 00F0 | 000F | 0 | 0 | 0 | 0 | 0 | 0 | 0000 | 0000 | 0100 |
| 002 | 0100 | 000F | 0000 | 0000 | 0100 | 0100 | 0100 | 0100 | 0 | 0 | 0 | 0 | 1 | 0 | 0000 | 0000 | 0100 |
| 003 | 0100 | 000F | 0000 | 0000 | 0100 | 0100 | 0100 | 0100 | 0 | 0 | 0 | 0 | 1 | 0 | AAAA | 0000 | 0100 |
| 004 | 0100 | 000F | 0000 | 0000 | 000F | 0000 | 000A | 000A | 0 | 0 | 0 | 0 | 0 | 0 | AAAA | 0000 | 0100 |
| 005 | 0100 | 000F | 0000 | 0000 | 0100 | 000A | 003B | 003B | 0 | 0 | 0 | 0 | 1 | 0 | AAAA | 003B | 0100 |

Проверка вычисления выражения вручную по микрокомандам:

1. (0x0100–0x0010) / 0x0010 = 0xF

2. 0xF & 0xAAAA = 0xA

3. 0xA + 0x31 = 0x3B

2) Трассировка работы программы в таблице 4 представлена для набора исходных данных №2:

[100h] = 0xAF00

AX = 0x0104

BX = 0x00AA

Таблица 4 — Трассировка программы №2

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| СМК | РЗУ | | | | RGA | RGB | ALU | SDA | RFI | | | | | | RGR | RGW | ARAM |
| AX | BX | CX | DX | N | Z | V | C | P | M |
| 000 | 0104 | 00AA | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0 | 0 | 0 | 0 | 0 | 0 | 0000 | 0000 | 0000 |
| 001 | 0104 | 0005 | 0000 | 0000 | 0104 | 00AA | 005A | 0005 | 0 | 0 | 0 | 0 | 0 | 0 | 0000 | 0000 | 0000 |
| 002 | 0104 | 0005 | 0000 | 0000 | 0104 | 0104 | 0100 | 0100 | 0 | 0 | 0 | 0 | 1 | 0 | 0000 | 0000 | 0100 |
| 003 | 0104 | 0005 | 0000 | 0000 | 0104 | 0104 | 0104 | 0104 | 0 | 0 | 0 | 0 | 0 | 0 | AF00 | 0000 | 0100 |
| 004 | 0104 | 0005 | 0000 | 0000 | 0005 | 0000 | 0009 | 0009 | 0 | 1 | 0 | 0 | 0 | 0 | AF00 | 0000 | 0100 |
| 005 | 0104 | 0005 | 0000 | 0000 | 0104 | 0104 | 0031 | 0031 | 0 | 0 | 0 | 0 | 1 | 0 | AF00 | 0031 | 0100 |

Проверка вычисления выражения вручную по микрокомандам:

1. (0x0104–0x00AA) / 0x0010 = 0x5

2. 0x5 & 0xAF00 = 0x0

3. 0x0 + 0x31 = 0x31

3) Трассировка работы программы в таблице 5 представлена для набора исходных данных №3:

[100h] = 0x00FF

AX = 0x1011

BX = 0x0111

Таблица 5 — Трассировка программы №3

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| СМК | РЗУ | | | | RGA | RGB | ALU | SDA | RFI | | | | | | RGR | RGW | ARAM |
| AX | BX | CX | DX | N | Z | V | C | P | M |
| 000 | 1011 | 0111 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0 | 0 | 0 | 0 | 0 | 0 | 0000 | 0000 | 0000 |
| 001 | 1011 | 00F0 | 0000 | 0000 | 1011 | 0111 | 0F00 | 00F0 | 0 | 0 | 0 | 0 | 0 | 0 | 0000 | 0000 | 0000 |
| 002 | 1011 | 00F0 | 0000 | 0000 | 0104 | 1011 | 0100 | 0100 | 0 | 0 | 0 | 0 | 1 | 0 | 0000 | 0000 | 0100 |
| 003 | 1011 | 00F0 | 0000 | 0000 | 0104 | 0104 | 1011 | 1011 | 0 | 0 | 0 | 0 | 1 | 0 | 00FF | 0000 | 0100 |
| 004 | 1011 | 00F0 | 0000 | 0000 | 00F0 | 0000 | 00F0 | 00F0 | 0 | 0 | 0 | 0 | 0 | 0 | 00FF | 0000 | 0100 |
| 005 | 1011 | 00F0 | 0000 | 0000 | 1011 | 00F0 | 0121 | 0121 | 0 | 0 | 0 | 0 | 1 | 0 | 00FF | 0121 | 0100 |

Проверка вычисления выражения вручную по микрокомандам:

1. (0x1011–0x0111) / 0x0010 = 0xF0

2. 0xF0 & 0x00FF = 0xF0

3. 0xF0 + 0x31 = 0x121

4) Трассировка работы программы в таблице 6 представлена для набора исходных данных №4:

[100h] = 0x1031

AX = 0x0011

BX = 0x0122

Таблица 6 — Трассировка программы №4

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| СМК | РЗУ | | | | RGA | RGB | ALU | SDA | RFI | | | | | | RGR | RGW | ARAM |
| AX | BX | CX | DX | N | Z | V | C | P | M |
| 000 | 0011 | 0122 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0 | 0 | 0 | 0 | 0 | 0 | 0000 | 0000 | 0000 |
| 001 | 0011 | FFFF | 0000 | 0000 | 0011 | 0122 | FEEF | FFFE | 1 | 0 | 0 | 0 | 0 | 0 | 0000 | 0000 | 0000 |
| 002 | 0011 | FFFF | 0000 | 0000 | 0011 | 0011 | 0100 | 0100 | 0 | 0 | 0 | 0 | 1 | 0 | 0000 | 0000 | 0100 |
| 003 | 0011 | FFFF | 0000 | 0000 | 0011 | 0011 | 0011 | 0011 | 0 | 0 | 0 | 0 | 0 | 0 | 1031 | 0000 | 0100 |
| 004 | 0011 | FFFF | 0000 | 1020 | FFFF | 0000 | 1020 | 1020 | 0 | 0 | 0 | 0 | 0 | 0 | 1031 | 0000 | 0100 |
| 005 | 0011 | FFFF | 0000 | 1020 | 0011 | 1020 | 1051 | 1051 | 0 | 0 | 0 | 0 | 0 | 0 | 1031 | 1051 | 0100 |

Проверка вычисления выражения вручную по микрокомандам:

1. (0x0011–0x0122) / 0x0010 = 0xFEEF

2. 0xFEEF & 0x1031 = 0x1020

3. 0x1020 + 0x31 = 0x1051

5) Трассировка работы программы в таблице 7 представлена для набора исходных данных №5:

[100h] = 0x1001

AX = 0x1000

BX = 0x0001

Таблица 7 — Трассировка программы №5

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| СМК | РЗУ | | | | RGA | RGB | ALU | SDA | RFI | | | | | | RGR | RGW | ARAM |
| AX | BX | CX | DX | N | Z | V | C | P | M |
| 000 | 1000 | 0001 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0 | 0 | 0 | 0 | 0 | 0 | 0000 | 0000 | 0000 |
| 001 | 1000 | 00FF | 0000 | 0000 | 1000 | 0001 | 0FFF | 00FF | 0 | 0 | 0 | 0 | 0 | 0 | 0000 | 0000 | 0000 |
| 002 | 1000 | 00FF | 0000 | 0000 | 1000 | 1000 | 0100 | 0100 | 0 | 0 | 0 | 0 | 1 | 0 | 0000 | 0000 | 0100 |
| 003 | 1000 | 00FF | 0000 | 0000 | 1000 | 1000 | 1000 | 1000 | 0 | 0 | 0 | 0 | 1 | 0 | 1001 | 0000 | 0100 |
| 004 | 1000 | 00FF | 0000 | 0000 | 00FF | 0001 | 0001 | 0001 | 0 | 0 | 0 | 0 | 1 | 0 | 1001 | 0000 | 0100 |
| 005 | 1000 | 00FF | 0000 | 0000 | 1000 | 0001 | 0032 | 0032 | 0 | 0 | 0 | 0 | 1 | 0 | 1001 | 0032 | 0100 |

Проверка вычисления выражения вручную по микрокомандам:

1. (0x1000–0x0001) / 0x0010 = 0xFF

2. 0xFF & 0x1000 = 0x1

3. 0x1 + 0x31 = 0x32

Заключение

В ходе выполнения лабораторной работы была изучена структура процессора intel8086. Изучена структура микрокоманд, получены навыки работы с эмулятором процессора intel8086 и составления микропрограмм.