Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Рязанский государственный радиотехнический университет имени В.Ф. Уткина»

Кафедра «ЭВМ»

ЭВМ и периферийные устройства

Отчёт о лабораторной работе №2

**«СТРУКТУРА ПРОЦЕССОРА И СОСТАВ МИКРОКОМАНД»**

|  |
| --- |

**Выполнили:**

ст. гр. 245

Бригада №1

Бекренев Владислав

Луковкин Иван

**Проверил:**

ст. пр. Устюков Д.И.

асс. Тарасов А.С.

Рязань 2023

**Цель работы:** изучение процессора на уровне структурной схемы, ознакомление со структурой микрокоманд (МК) и порядком ввода данных, кодирование и выполнение МК.

**Теоретическая часть**

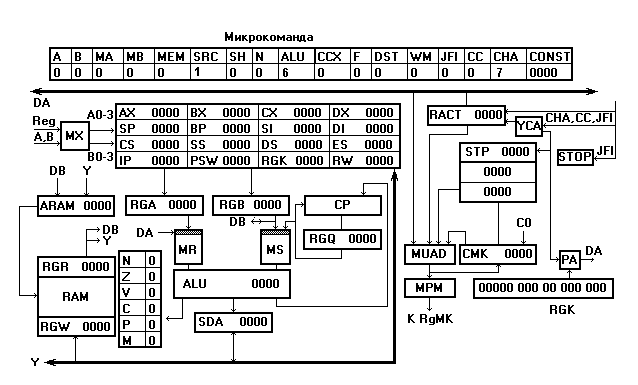


Рисунок 1 - Структура микропрограммируемого процессора

****

Рисунок 2 - Структура микрокоманды с описанием назначения ее полей

Поля **А** и **В** задают адреса регистров РЗУ.

Поля **МА** и **МВ** управляют выбором адресов РЗУ из микрокоманды или регистра команд RGK.

Поле **МЕМ** управляет чтением и записью памяти.

Поле **SRC** управляет выбором источников операндов.

Поле **SH** управляет работой сдвигателей SDA и CP. **N** – величина сдвига.

Поле **ALU** управляет операциями арифметико-логическим устройством.

Поле **ССХ** управляет входным переносом.

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.

Поле **WM** управляет записью в память.

Поле **JFI** участвует совместно с полем СС в формировании условий перехода (условный/безусловный переход).

Поле **СС** управляет формированием условий перехода (определяет выбор условий).

Поле **СНА** обеспечивает формирование адреса следующей микрокоманды.

Поле **CONST** содержит 16-битовую константу, используемую в ходе микроопераций для вычислений/как маску/для адресации ОП.

**Практическая часть**

Задание (Вариант 1): (AX – BX) / 16 & [100h] + 31h → [100h]

Результаты выполнения задания представлены в таблицах 1-2. Трассировка работы программы в таблице 2 представлена для набора исходных данных A.

Исходные данные:

1. AX = 0505h; BX = 0101h; [100h] = FFFFh
2. AX = 0303h; BX = 0202h; [100h] = 0000h
3. AX = 0606h; BX = 0404h; [100h] = 0100h
4. AX = 0202h; BX = 0101h; [100h] = 0101h
5. AX = 0707h; BX = 0101h; [100h] = 0001h

Рисунок 3 - Код программы

Таблица 1 - Память микрокоманд

| Адрес  МК | A | B | MA | MB | MEM | SRC | SH | N | ALU | CCX | F | DST | WM | JFI | CC | CHA | CONST |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 000 | (AX - BX) / 16 | | | | | | | | | | | | | | | | |
| 0 | 3 | 0 | 0 | 0 | 1 | 1 | 4 | 2 | 1 | 0 | 4 | 0 | 0 | 0 | 7 | 0000 |
| 001 | Обращение в ARAM к [100h] ячейке | | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 5 | 5 | 0 | 0 | 6 | 0 | 0 | 0 | 2 | 0 | 0 | 7 | 0100 |
| 002 | BX & RGR | | | | | | | | | | | | | | | | |
| 3 | 3 | 0 | 0 | 0 | 3 | 0 | 0 | 9 | 0 | 0 | 4 | 0 | 0 | 0 | 7 | 0000 |
| 003 | BX + CONST | | | | | | | | | | | | | | | | |
| 0 | 3 | 0 | 0 | 7 | 5 | 0 | 0 | 3 | 0 | 0 | 0 | 1 | 5 | 0 | 7 | 0031 |

Таблица 2 - Трассировка программы

| СМК | РЗУ | | | | RGA | RGB | ALU | SDA | RFI | | | | | | RGR | RGW | ARAM |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| AX | BX | CX | DX | N | Z | V | C | P | M |
|  | 0505 | 0101 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0 | 0 | 0 | 0 | 0 | 0 | 0000 | 0000 | 0000 |
| 000 | 0505 | 0040 | 0000 | 0000 | 0505 | 0101 | 0404 | 0040 | 0 | 0 | 0 | 0 | 1 | 0 | 0000 | 0000 | 0000 |
| 001 | 0505 | 0040 | 0000 | 0000 | 0505 | 0505 | 0100 | 0100 | 0 | 0 | 0 | 0 | 1 | 0 | FFFF | 0000 | 0100 |
| 002 | 0505 | 0040 | 0000 | 0000 | 0040 | 0040 | 0040 | 0040 | 0 | 0 | 0 | 0 | 1 | 0 | FFFF | 0000 | 0100 |
| 003 | 0505 | 0040 | 0000 | 0000 | 0505 | 0040 | 0071 | 0071 | 0 | 0 | 0 | 0 | 0 | 0 | FFFF | 0071 | 0100 |

**Экспериментальная часть:**

Проверка вычислений:

1. AX = 0505h; BX = 0101h; [100h] = FFFFh
   1. 0505h - 0101h = 0404h
   2. 0404h / 16 = 0040h
   3. 0040h & FFFFh = 0040h
   4. 0040h + 0031h = 0071h
2. AX = 0303h; BX = 0202h; [100h] = 0000h
   1. 0303h - 0202h = 0101h
   2. 0101h / 16 = 0010h
   3. 0010h & 0000h = 0000h
   4. 0000h + 0031h = 0031h
3. AX = 0606h; BX = 0404h; [100h] = 0100h
   1. 0606h - 0202h = 0404h
   2. 0404h / 16 = 0040h
   3. 0040h & 0100h = 0000h
   4. 000h + 0031h = 0031h
4. AX = 0202h; BX = 0101h; [100h] = 0101h
   1. 0202h - 0101h = 0101h
   2. 0101h / 16 = 0010h
   3. 0010h & 0101h = 0000h
   4. 0000h + 0031h = 0031h
5. AX = 0707h; BX = 0101h; [100h] = 0001h
   1. 0707h - 0101h = 0606h
   2. 0606h / 16 = 0060h
   3. 0060h & 0101h = 0000h
   4. 0000h + 0031h = 0031h

Вывод: программа работает в соответствии с заданием на всех протестированных наборах исходных данных.

**Заключение**

В ходе выполнения лабораторной работы были изучен процессор на уровне структурной схемы, структура микрокоманд (МК) и порядок ввода данных, кодирование и выполнение МК.