## Лабораторная работа № 1

**Тема:** организация ЭВМ

**Цель:** изучение структуры учебной ЭВМ и исполнение команд;

**Техническое обеспечение:** персональный компьютер, программная модель учебной ЭВМ;

**Основные понятия:** структура ЭВМ, состав процессора, регистры состояния, ОЗУ, операнд, цикл команды, микрокоманда.

Общие положения

Для изучения базовых понятий архитектуры ЭВМ взята программную модель учебной ЭВМ. Модель можно рассматривать как некий собирательный образ простой ЭВМ, которая иллюстрирует принципы организации командного цикла, механизмы взаимодействия процессора и внешних устройств, работы с кэш-памятью и архитектурные особенности, которой, соответствуют тенденциям развития современных ЭВМ.

Программирование может осуществляться как в машинных кодах, так и на языке Ассемблер. При кодировании команд и данных использована десятичная система счисления.

Структура учебной ЭВМ

Модель учебной ЭВМ включает процессор, оперативную (ОЗУ) и сверхоперативную память (РОН), устройство ввода (УВв) и устройство вывода (УВыв). Процессор, в свою очередь, состоит из центрального устройства управления (УУ), арифметического устройства (АУ) и системных регистров (CR, PC, SP, RA, RB и др.), рисунок 1.

При желании можно подключить дополнительно модели кэш-памяти, контроллера клавиатуры, символьного дисплея, блока таймеров, тоногенератор, и контроллер семисегментной индикации. Программирование и подключение внешних устройств рассматривается в лабораторной работе №6.

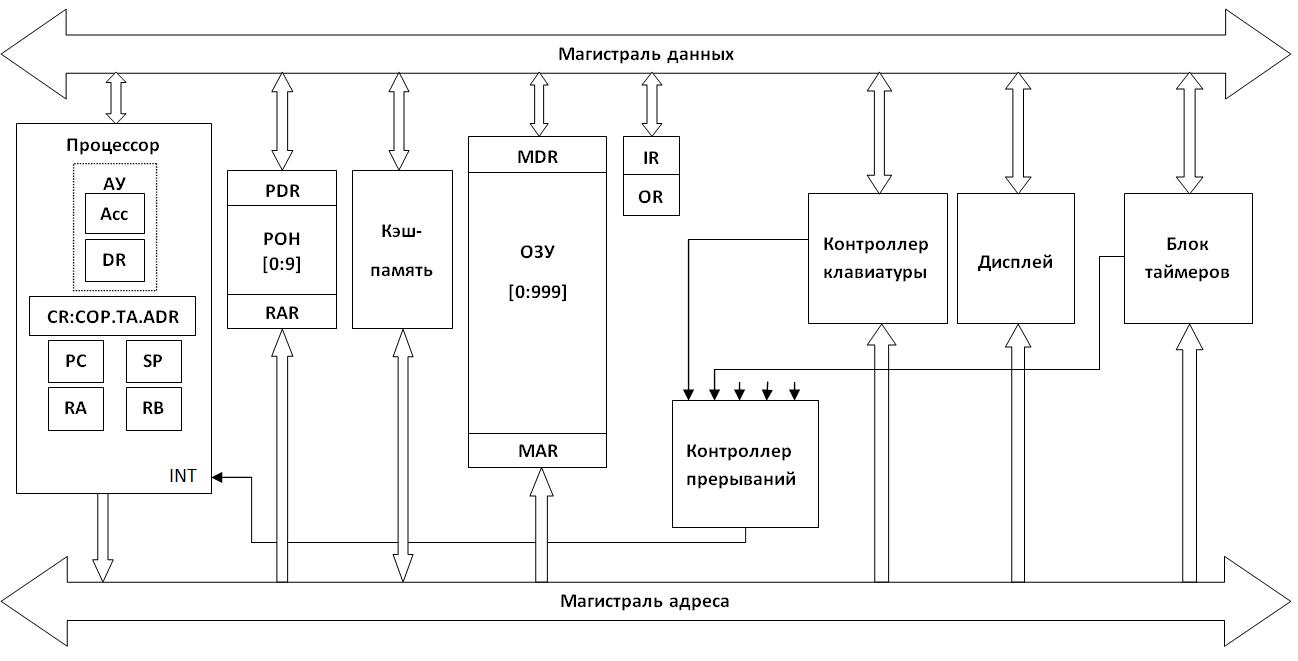
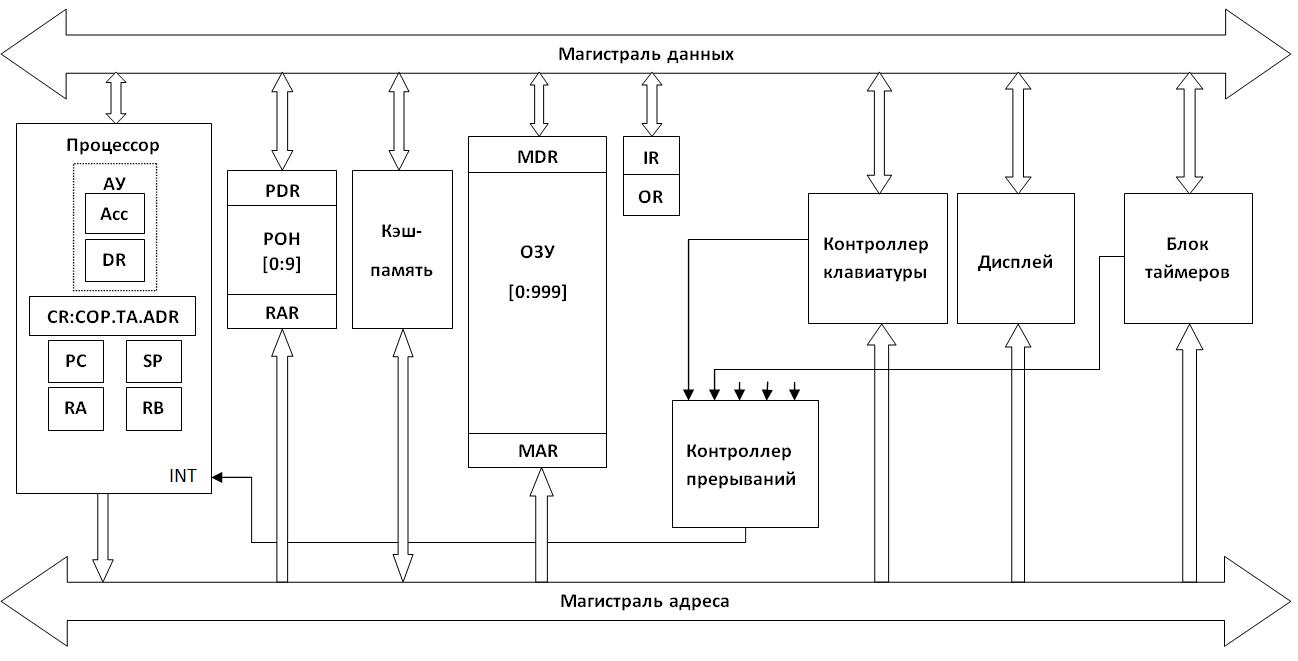


Рисунок 1 – Структурная схема учебной ЭВМ

Состав процессора учебной ЭВМ

**Арифметическое устройство (АУ)**: АУ выполняет арифметические или логические операции над входными данными. Какая именно арифметическая операция должна быть выполнена определяется кодом операции (СОР) регистра команд процессора (CR). Результат выполненной операции помещается в аккумулятор (Асс).

**Аккумулятор (Аccumulator)**: регистр, для хранения одного из операндов арифметической или логической операции. Также, через аккумулятор (Асс) производятся операции ввода и вывода.

Изначально оба операнда могут хранится в ОЗУ, и до выполнения операции один из них нужно загрузить в аккумулятор. После выполнения команды обработки результат помещается в аккумулятор, и при необходимости его можно сохранить в ячейке памяти.

Поскольку адрес одного из операндов предопределен, в командах обработки данных достаточно указать явно местоположение только второго операнда.

**Регистр данных (Data Register):** этот регистр призван компенсировать разницу в быстродействии ОЗУ и процессором. Используется для временного хранения данных при обмене информацией между памятью и процессором. Когда операнд находится в регистре данных (DR), с ним можно произвести арифметическую или логическую операцию. Содержимое регистра данных (DR) может быть, например, сложено с содержимым аккумулятора (Асс), а полученный результат занесен в аккумулятор (Асс).

В состав устройства управления (УУ) входят:

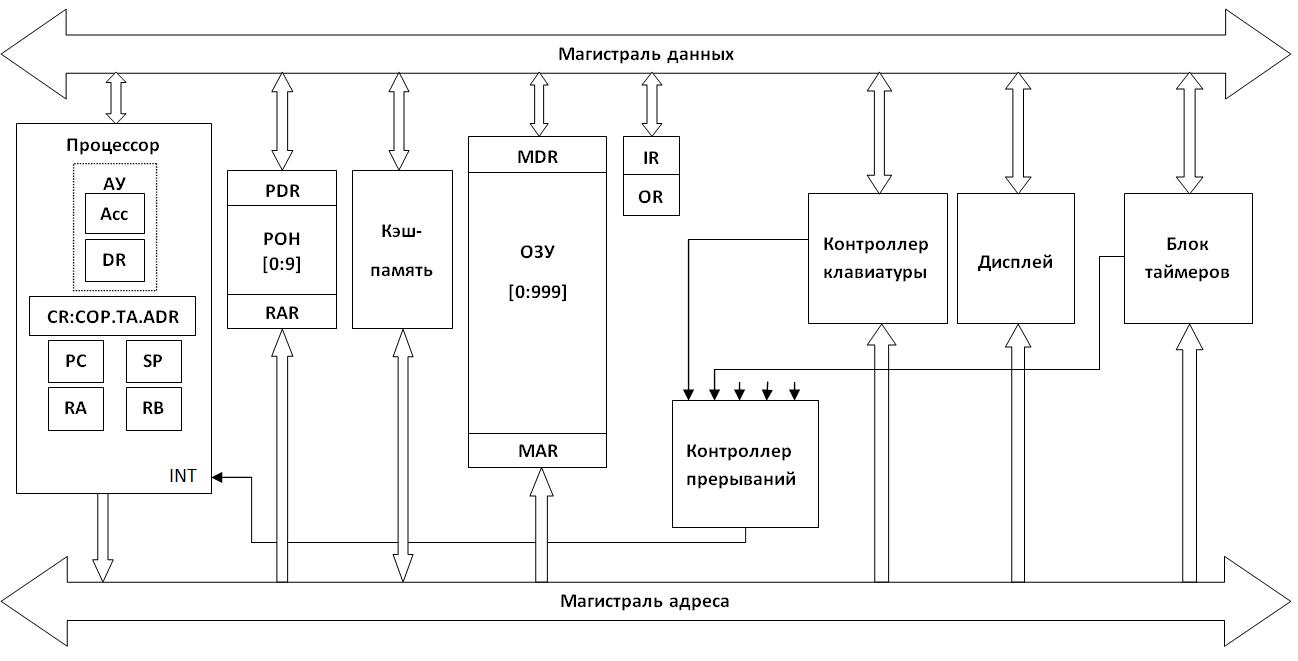
**PC** – счетчик адреса команды, содержащий адрес текущей команды;

**CR** – регистр команды, содержащий код команды;

**RB** – регистр базового адреса, содержащий базовый адрес данных;

**SP** – указатель стека, содержащий адрес верхушки стека;

**RA** – регистр адреса, содержащий адрес при косвенной адресации.

**Счетчик адреса команды (PC, Program Counter)**: используется для хранения адреса команды, непосредственно выполняемой ЭВМ. Перед началом вычислений в счетчик адреса команды (РС) заносится адрес ячейки памяти, где хранится команда, которая должна быть выполнена. Чтобы изменить порядок вычислений, достаточно в счетчик адреса команды (РС) занести адрес перехода.

**Регистр команд (CR, Command Register)**: используется для хранения кода команды, непосредственно выполняемой машиной. Для того, чтобы приступить к выполнению команды, код команды необходимо извлечь из памяти и разместить в регистре команд (CR), где код операции команды хранится в течение всего времени выполнения команды. Регистр команд(СR) содержит следующие поля: СОР — код операции; ТА — тип адресации; ADR — адрес или непосредственный операнд, рисунок 3.

**Регистр базового адреса (RB, Register Base)**: хранит указатель базового адреса некоторого объекта в памяти. Другими словами, в регистр базового адреса можно записать адрес ячейки памяти (базовый адрес), а затем выполнить команду, операнд которой располагается по этому адресу памяти.

*Базовый адрес* – это адрес начала некоторой структуры данных (например, массива данных).

**Указатель стека (SP, Stack Pointer):** регистр, который содержит адрес вершины стека. *Вершина стека* – это ячейка, куда была произведена последняя по времени запись.

**Регистр исполнительного адреса (RA, Register Address)**: содержит исполнительный адрес при косвенной адресации. *Исполнительный адрес* – это код номера ячейки, по которому будет записан или считан операнд.

Регистры Acc, DR, IR, OR, CR и имеют длину 6 десятичных разрядов, регистры PC, SP, RA и RB – 3 разряда.

Регистры состояния

При завершении выполнения операции арифметическое устройство (АУ) вырабатывает сигналы признаков результата: **Z, S, OV**. Сигналы признаков результата отражает регистр состояния или, регистр флагов.

*Регистр флагов* – это регистр, предназначенный для фиксации и хранения признаков (флагов) результата последней выполненной арифметической или логической операции. Флаги устанавливаются в каждом цикле АУ и информируют о состоянии результата предыдущей операции:

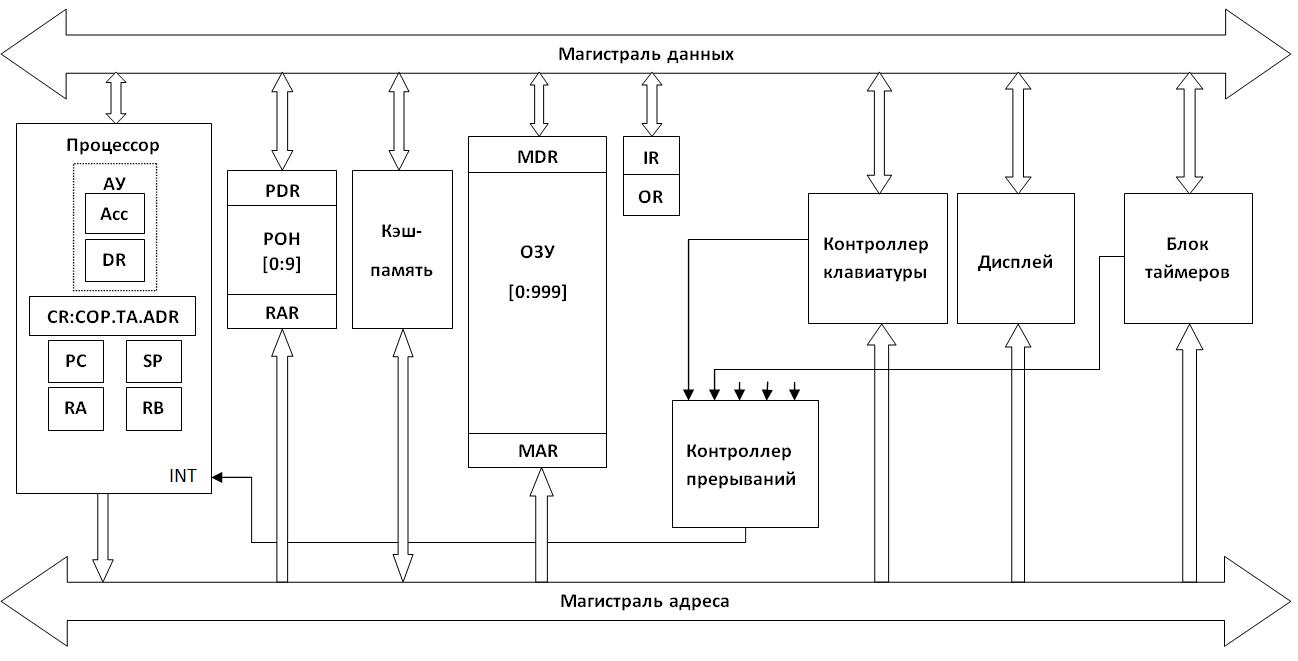
* Результат равен нулю. В регистре установится в единичное состояние флаг **Z (Zero)**;
* Результат отрицательный. В регистре установится в единичное состояние флаг **S (SIgn)**;
* Переполнение. При переполнении разрядной сетки установится в единичное состояние флаг **OV (Overflow**.

В случаях, когда эти условия не выполняются, соответствующие сигналы имеют нулевое значение.

Например: команда SUB вычитает один операнд из другого и на основе полученной разности устанавливаются флаги. Если разность равна нулю, то установится в единичное состояние флаг Z; если разность отрицательна, в единичное состояние устанавливается флаг S. Если разделить (команда DIV) операнд на ноль, произойдет переполнение, в единичное состояние установится флаг OV.

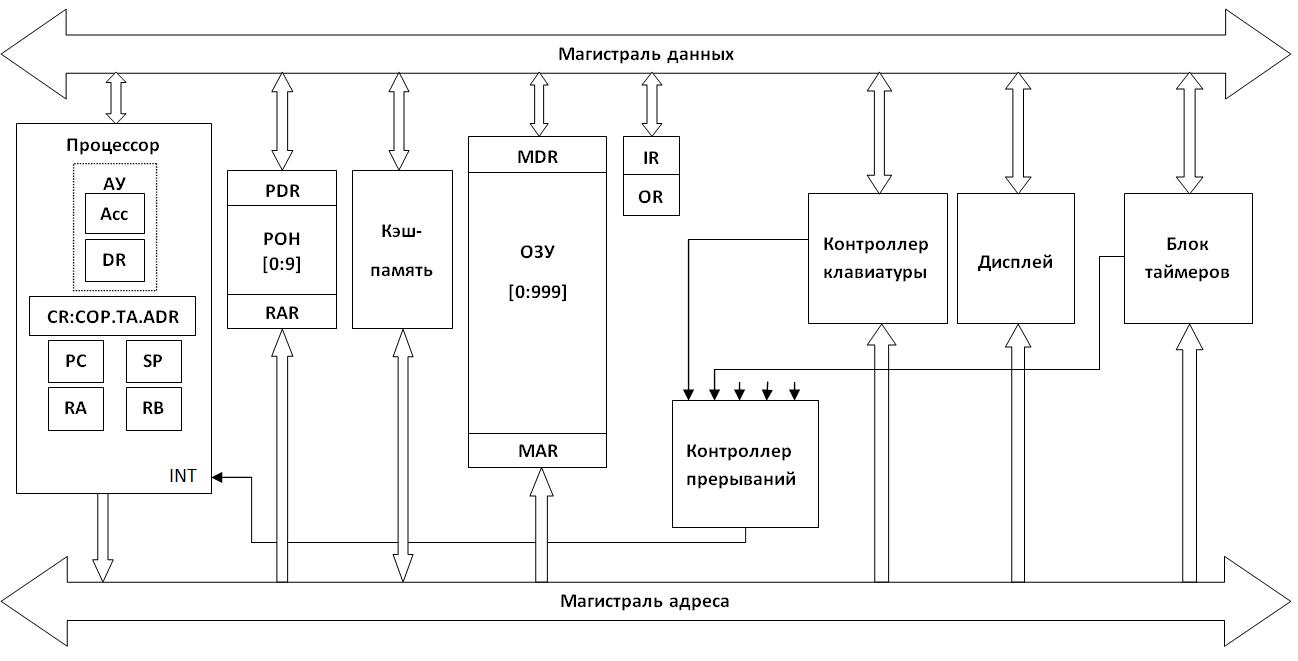
Коды условий очень важны поскольку используются при сравнениях и условных переходах. Флаговый регистр может хранить не только коды условий. Его содержимое в разных машинах может быть разным. Дополнительные поля могут указывать режим машины, уровень приоритета процессора, статус разрешения прерываний.

Оперативное запоминающее устройство

В ячейках оперативного запоминающего устройства (ОЗУ) хранятся команды и данные. Емкость ОЗУ составляет 1000 ячеек. По сигналу MWr (Memory Write) выполняется запись содержимого регистра данных памяти (MDR) в ячейку памяти с адресом, указанным в регистре адреса памяти (MAR). По сигналу MRd (Memory Read) происходит считывание — содержимое ячейки памяти с адресом, содержащимся в MAR памяти, передается в MDR памяти. Ячейки ОЗУ имеют длину 6 десятичных разрядов.

**Регистр данных памяти (**MDR**, Memory Data Register):** используется в качестве буфера между памятью и остальными регистрами процессора; через него пересылаются в процессор команды (операнды) и передаются в память результаты обработки.

**Регистр адреса памяти (**MAR**, Memory Address Register):** предназначен для хранения адреса ячейки памяти вплоть до завершения операции (считывание или запись) с этой ячейкой. Наличие регистра адреса памяти позволяет компенсировать различия в быстродействии оперативной памяти и прочих устройств машины.

**Сверхоперативная память с прямым доступом (СОЗУ):** не является частью ОЗУ. Содержит десять регистров общего назначения R0—R9 (РОН). Фактически РОН – это небольшая регистровая память, доступ к которой осуществляется специальными командами. Обычно в РОН размещают многократно используемые адреса, счетчики циклов и др. Доступ к ним осуществляется (аналогично доступу к ОЗУ) через регистры RAR (Register Address Register) и RDR (Register Data Register).

Размер регистров обычно фиксирован: в модели учебной ЭВМ, РОН имеют длину 6 десятичных разрядов. К любому регистру можно обратиться, указав его номер. С учетов возможного размещения операндов выделяют три подвида команд:

* регистр-регистр: операнды находятся только в регистрах;
* регистр-память: один из операндов размещается в регистре, а второй в основной памяти;
* память –память: оба операнда хранятся в основной памяти.

Исполнение команд

В процессе исполнения команд устройство управления ЭВМ производит анализ и пересылку команд, отдельных ее частей (кода операции, признака адресации и адреса) или операнда из одного регистра ЭВМ в другой ее регистр, АЛУ, память или устройство ввода-вывода. Эти действия (микрооперации) протекают в определенной временной последовательности и скоординированы между собой. *Операнд* – это данные, которыми оперирует машинная команда. К наиболее общим типам операндов можно отнести: адреса, числа, символы и логические данные.

**Цикл команды.** Для реализации одной команды требуется выполнить определенное количество микрокоманд, каждая из которых инициируется одним тактовым импульсом. *Микрокоманда* – это микрооперация реализующая элементарные действия над операционными элементами процессора.

Общее число тактовых импульсов, требуемых для выполнения команды, определяет время ее выполнения, называемое *циклом команды*. Цикл команды обычно включает один или несколько машинных циклов.

Допустим, с помощью учебной ЭВМ необходимо получить сумму чисел 45 и 25. Аккумулятор (Асс) содержит первый операнд – число 45. Второй операнд (число 25) находится непосредственно в адресной части команды (ADD #25).

Для выполнения команды ADD #25 требуется два машинных цикла: выборка команды и исполнение команды. Общее число тактовых импульсов – семь, рисунок 3. Рассмотрим подробнее каким образом происходит выполнение команды сложения ADD #25.

**Выборка команды:**

1) Из счетчика адреса команды РС адрес ячейки памяти (000) выполняемой команды заносится в регистр адреса ОЗУ (MAR), где адрес ячейки хранится до завершения операции, рисунок 3, *а*;

2) Код команды (231025) считывается из ячейки памяти в регистр данных ОЗУ MDR, рисунок 3, *б*;

3) Далее, содержимое MDR (231025) считывается в регистр команд процессора CR, рисунок 3,*в*;

4) Счетчик адреса команды РС увеличивается на 1. Т.е. в РС заносится адрес следующей команды, которая должна быть выполнена; Таким образом завершается машинный цикл «Выборка команды».

Безадресные или команды ввода-вывода окончательно исполняются в этом же цикле, т.е. это одноцикловые команды.

**Исполнение команды:**

1) Из адресной части ADR регистра команд процессора CR считывается операнд (число 25) в регистр данных DR, рисунок 3, д;

2) Затем, из регистра команд процессора CR, код операции (23) считывается АЛУ, рисунок 3, е;

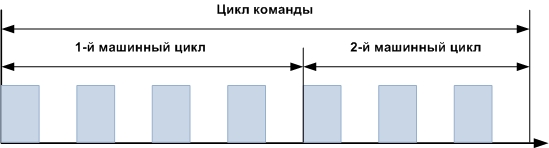
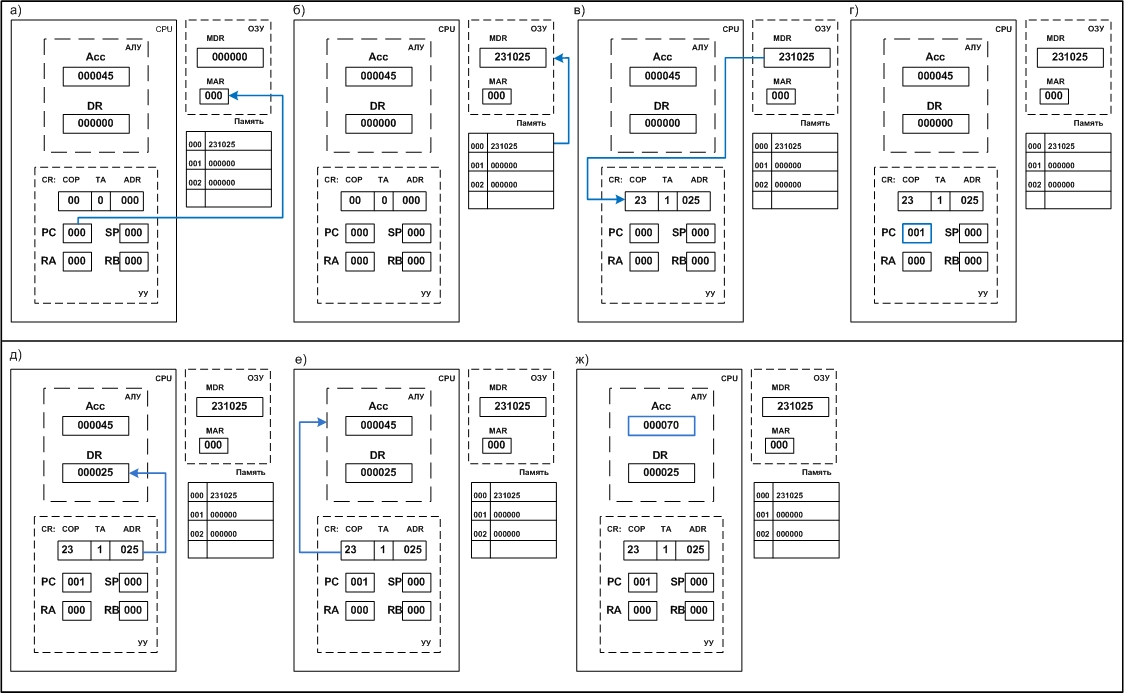
3) Содержимое регистра данных процессораDR суммируется с содержимым аккумулятора Асс и результат помещается в аккумулятор Асс, рисунок 3, ж.

Рисунок 2.2 – Цикл команды

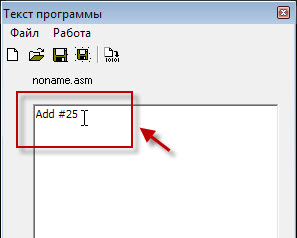
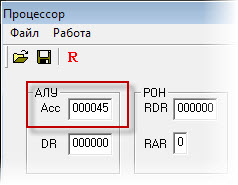
Рисунок 3 – Исполнение команды

Порядок выполнения работы

1 Ознакомится с теоретическим материалом Открыть папку «Модель ПК»; привести в исполнение файл *CompModel.exe;*

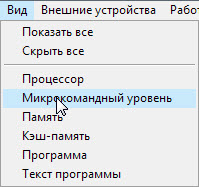
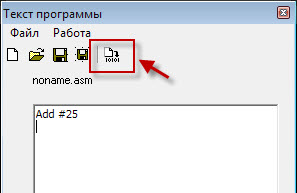
2Ввести в аккумулятор число 45, окно «Процессор»:

3. В окно «Текст программы» ввести команду в соответствии с заданным вариантом, таблица 2.



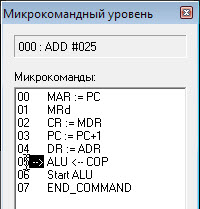
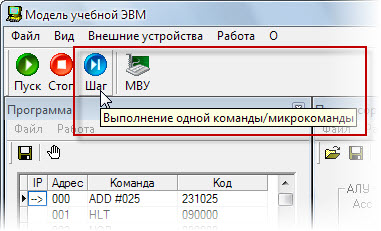
4. Выполнить компиляцию: для этого , в окне «Текст программы» нажать кнопку **Компилировать**. О корректном вводе команды свидетельствует сообщение: «Ошибок не обнаружено» в окне «Текст программы». Окно «Программа» содержит дизассемблированный мнемокод и машинный код команды.

5. Активировать режим микрокоманд. В основном окне «Модель учебной ЭВМ» выбрать меню **Вид → Микрокомандный уровень.**



6. Привести модель учебной ЭВМ в состояние работа, посредством команды Шаг. Выполните команду в режиме **Шаг**.

7. Наблюдайте процесс выполнения команды на уровне микрокоманд. В окно «Микрокомандный уровень» выводится: мнемокод выполняемой команды, список микрокоманд, ее реализующих, и указатель на текущую выполняемую микрокоманду.



8. Определить общее число тактовых импульсов выполнения команды и количество машинных циклов. Показать какие микрокоманды относятся к каждому машинному циклу. Зарисовать диаграмму. Записать микрокоманды и указать машинные циклы. Количество тактовых импульсов на диаграмме соответствует выполняемым микрокомандам.

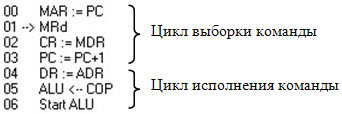


Таблица 2 – Список команд по вариантам

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Номер варианта** | **Команды** | **Первый операнд (Асс)** | **Номер варианта** | **Команды** | **Первый операнд (Асс)** |
| 1 | ADD #10 | 20 | 6 | RD #30 | - |
| 2 | SUB #5 | 30 | 7 | DIV #20 | 200 |
| 3 | MUL #2 | 15 | 8 | MUL #7 | 7 |
| 4 | DIV #5 | 10 | 9 | SUB #15 | 30 |
| 5 | RD #15 | - | 10 | ADD #100 | 100 |

## Форма и содержание отчета

Отчет выполняется на листах формата А4, шрифт Times New Roman, размер шрифта 14 и должен содержать:

1. Титульный лист;
2. Номер и тема лабораторной работы;
3. Формулировка и вариант задания (табл.2.2);
4. Микрокоманды мнемокода с указанием машинных циклов;
5. Диаграмма цикла команды.

Контрольные вопросы

1. Назначение аккумулятора?
2. Если результат выполненной арифметической операции равен нулю, какой флаг установится в единичное состояние?
3. Что определяет цикл команды? Какие машинные циклы может включать? Сколько машинных циклов требуется для выполнения безадресной команды?