Лекция

Базовые компоненты компьютера

КОМПОНЕНТЫ КОМПЬЮТЕРА

- На верхнем уровне компьютер состоит из центрального процессора (CPU), памяти и устройств ввода - вывода, с одним или более модулями каждого типа
- Эти компоненты связаны некоторым способом, чтобы реализовать основную функцию компьютера - выполнить программы

Концепция фон Неймана

- Практически все современные компьютеры следуют концепции, выработанной фон Нейманом. Эта концепция включает три основных момента:

  - данные и команды хранятся совместно в единой подсистеме памяти, способной выполнять операции чтения и записи;

Концепция фон Неймана

- к отдельным элементам данных, хранящимся в памяти, можно обращаться по адресу, характеризующему её положение в общем массиве, независимо от смысла затребованных данных, т.е. независимо от того, являются ли эти данные командой или операндами;
- заданный алгоритм решения задачи реализуется последовательным выполнением элементарных команд в порядке их расположения в памяти, если только иное не будет указано явно.

Базовые компоненты компьютера

- Существует небольшой набор базовых логических элементов, комбинируя которые разными способами, можно создавать средства хранения данных и их логической обработки, в том числе и выполнения арифметических операций

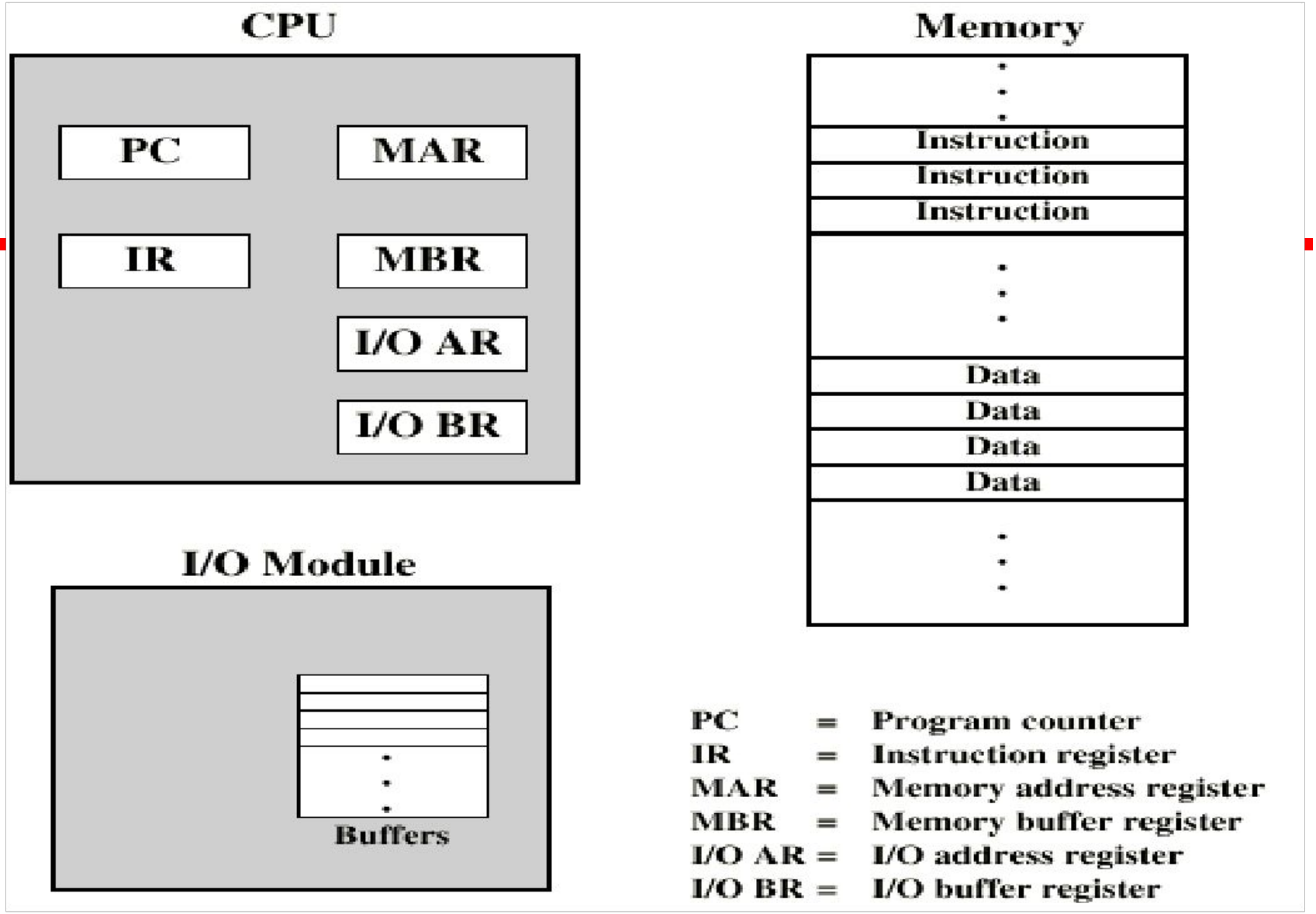


Figure 6.1 - Computer components: top-level view

Центральный процессор

- ЦП должен обмениваться с памятью данными и извлекать из нее команды программы
- Для этого в составе ЦП имеются два регистра (внутренних элемента памяти)

- РгАП регистр адреса памяти (MAR), в котором формируется адрес очередной ячейки памяти для выполнения записи или чтения

Центральный процессор

- БРгП **буферный регистр памяти** (MBR), который служит для временного хранения данных, записываемых в память или считанных из памяти

- ПСч **программный счетчик** (PC) хранит адрес очередной команды
- РгК **регистр команд** (IR) служит для временного хранения выполняемой команды

Центральный процессор

- **РгА В/В** регистр адреса внешнего устройства (**I/O AR)** служит для хранения адреса устройства ввода-вывода
- **БРг В/В** регистр данных внешнего устройства (**I/O BR)** — для хранения данных, передаваемых в устройство ввода-вывода или получаемых из него

Устройство оперативной памяти

- Содержит множество ячеек
- Каждая ячейка имеет свой числовой идентификатор - **адрес**
- В каждой ячейке хранится слово, которое может быть интерпретировано или как элемент данных, либо как команда

Модуль ввода-вывода

- Передает данные от внешнего устройства в ЦП или в память и обратно
- В состав модуля ввода-вывода входят внутренние буферы для временного хранения передаваемых данных

Выполнение программы

- Выполнение последовательности элементарных шагов
- Каждый шаг - это выполнение арифметической, логической операции или операции обмена данными
- Для каждой операции требуется индивидуальный набор управляющих сигналов

Выполнение программы

- Каждая команда имеет уникальный код

  например Add, Move и т.д.
- Аппаратный сегмент принимает код и формирует сигналы управления
- Процесс обработки отдельной команды принято называть ***циклом обработки команды***

Командный цикл

- Два цикла:
  - Загрузка
  - Выполнение

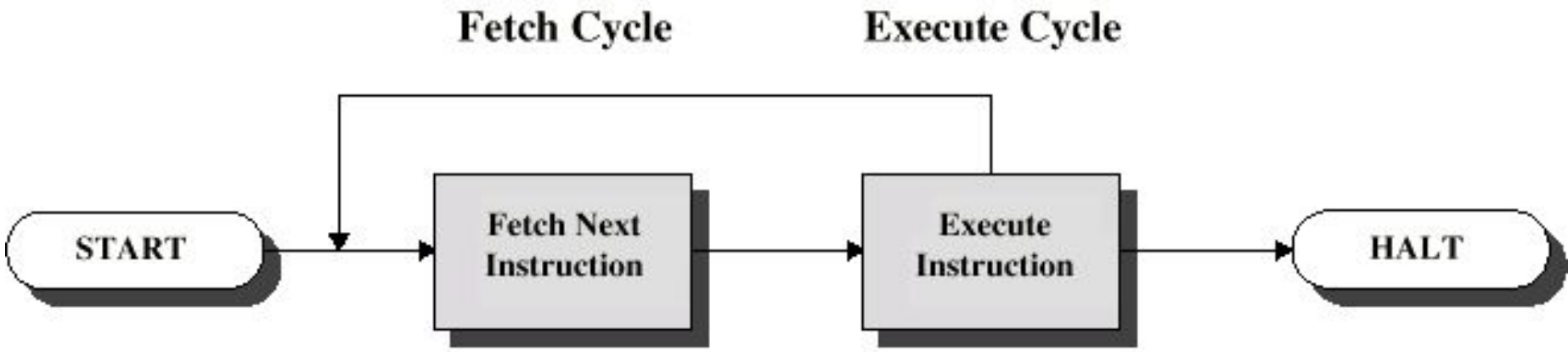


Схема основного цикла обработки команды

Цикл загрузки

- **Программный счетчик (PC)** содержит адрес следующей команды
- Процессор загружает команду из памяти по адресу, указанному в программном счетчике
- Инкремент программного счетчика, если не указан адрес перехода
- Команда загружается в регистр команд (РгК) процессора

Цикл выполнения

- Процессор распознает команду и выполняет необходимые действия
- Часть битов кода команды представляет собой ***код операции,*** определяющий, какая именно операция должна быть выполнена процессором
- Выполняя заданную операцию, процессор будет производить элементарные действия, которые можно разделить на четыре категории

Цикл выполнения

- **Процессор - память**

  Данные перемещаются между ЦП и памятью
- **Процессор – модуль ввода/вывода**

  Данные перемещаются между ЦП и модулем в/в

Цикл выполнения

- **Обработка данных**

  Процессор выполняет заданную арифметическую или логическую операцию над данными
- **Управление**

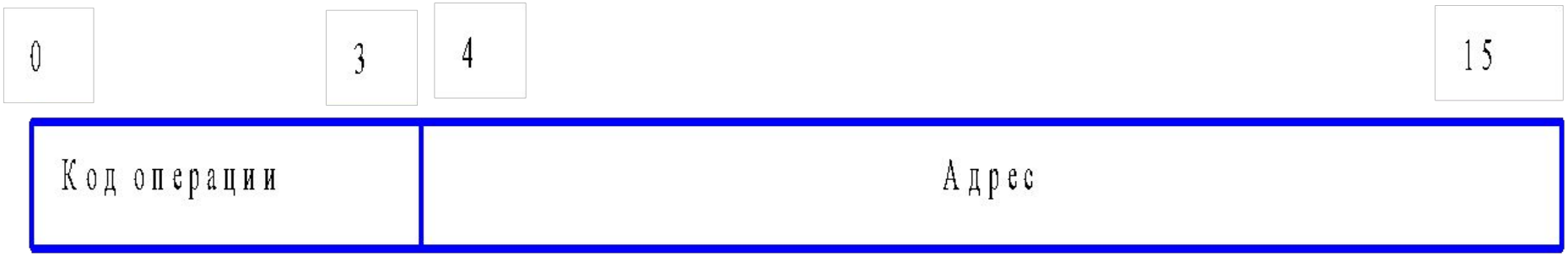
  Изменение естественной последовательности команд программы

  Например, условный или безусловный переход

Постановка задачи

- Выполнить пошаговое сложение кодов двух чисел 3 и 2, расположенных в ячейках 940 и 941 оперативной памяти. Результат разместить в ячейке 941. Набор команд следует разместить в последовательности ячеек памяти, начиная с адреса 300

Форматы команд и данных



а) Формат команды

б) Формат представления целого числа

Форматы команд и данных

- Как команды, так и данные представлены 16-разрядными словами

- Формат команды позволяет представить множество $2^{4} = 16$ различных кодов инструкций и $2^{12} = 4096$ (4K) слов памяти при прямой адресации

Набор регистров ЦП

- **Счетчик команд PC,** в котором формируется адрес очередной команды
- **Регистр команды IR,** в который считывается и в котором хранится в процессе выполнения очередная команда
- **Аккумулятор AC -** рабочий регистр для временного хранения данных

Набор команд

- **0001 -** загрузка AC из памяти $\mathbf{(1)_h}$;
- **0010 -** сохранение содержимого AC в памяти $\mathbf{(2)_h}$;
- **0101 -** добавление к содержимому AC числа, считанного из памяти $\mathbf{(5)_h}$.

Распределение памяти

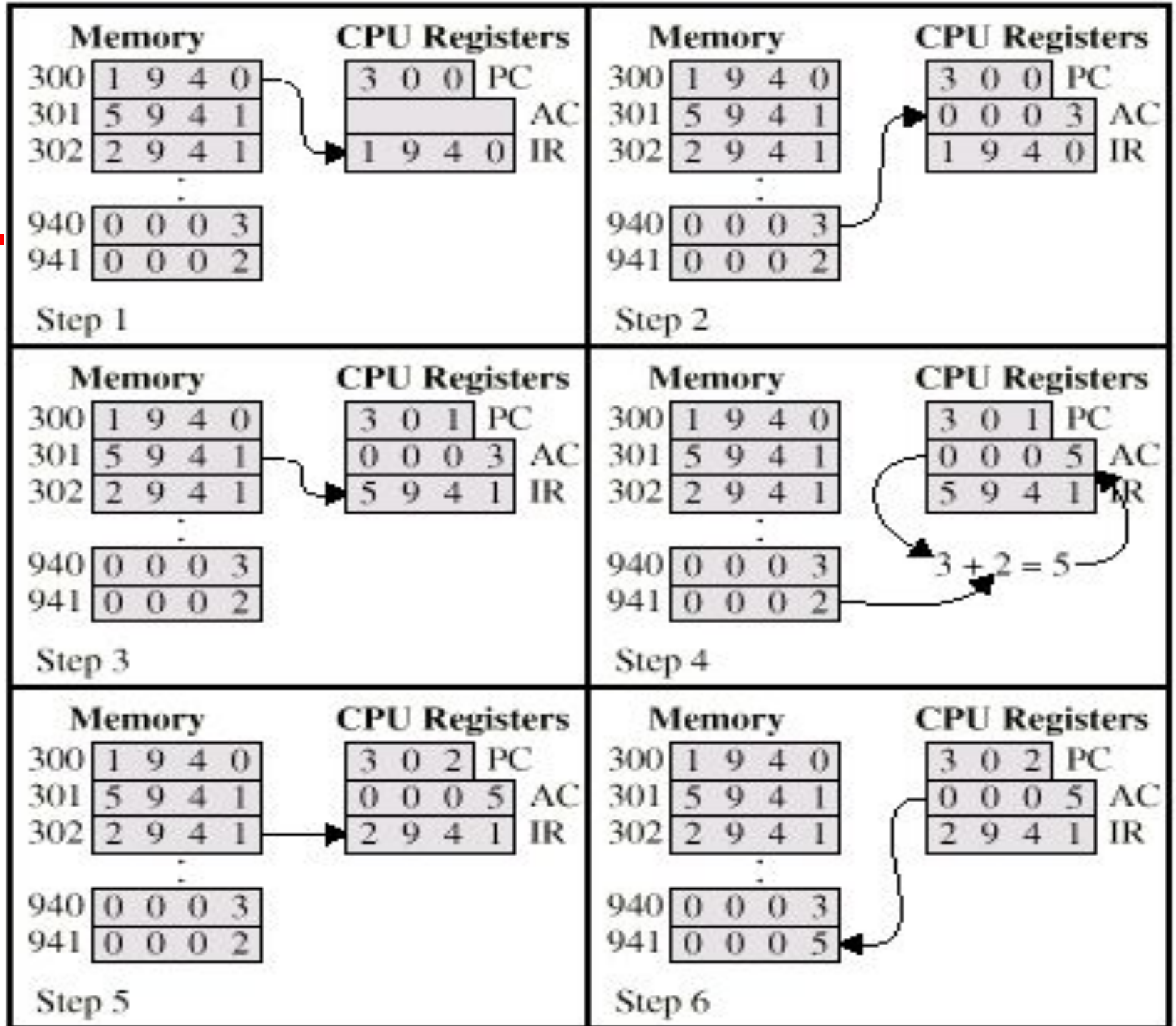
Данные

Адрес	Операнд
940	0003
941	0002

Распределение памяти

Программа

Адрес	Команда
300	1 940
301	5 941
302	2 941


Memory
CPU Registers
300 1 9 4 0
301 5 9 4 1
302 2 9 4 1
940 0 0 0 3
941 0 0 0 2
3 0 0 PC
AC
1 9 4 0 IR
Step 1
0 0 0 3 AC
Step 2
3 0 1 PC
0 0 0 3 AC
5 9 4 1 IR
Step 3
0 0 0 5 AC
3 + 2 = 5
Step 4
3 0 2 PC
2 9 4 1 IR
Step 5
Step 6

Пошаговое выполнение команд

- 1 В **счетчике команд PC** содержится число 300 — адрес первой команды фрагмента
- Эта команда на фазе извлечения считывается в **регистр команд IR**
- При считывании команды одновременно выполняется **приращение адреса в PC**

Пошаговое выполнение команд

- 2 Первые 4 разряда в IR — **код операции** — указывают ЦП, что нужно выполнить **чтение** ячейки памяти по адресу, заданному в остальной части команды (940), и результат **записать в аккумулятор AC**
- 3 Следующая команда извлекается из ячейки памяти, заданной содержимым PC, а само содержимое PC увеличивается

Пошаговое выполнение команд

- 4 Первые 4 разряда новой команды задают **сложение** содержимого АС с числом, считанным из ячейки памяти, адрес которой указан в адресной части команды. Вычисленная **сумма остается в АС**.
- 5 Извлекается **следующая команда**.
- 6 Содержимое АС **записывается** в память по адресу, указанному в адресной части команды — в ячейку 941.

Цикл команды - диаграмма состояний

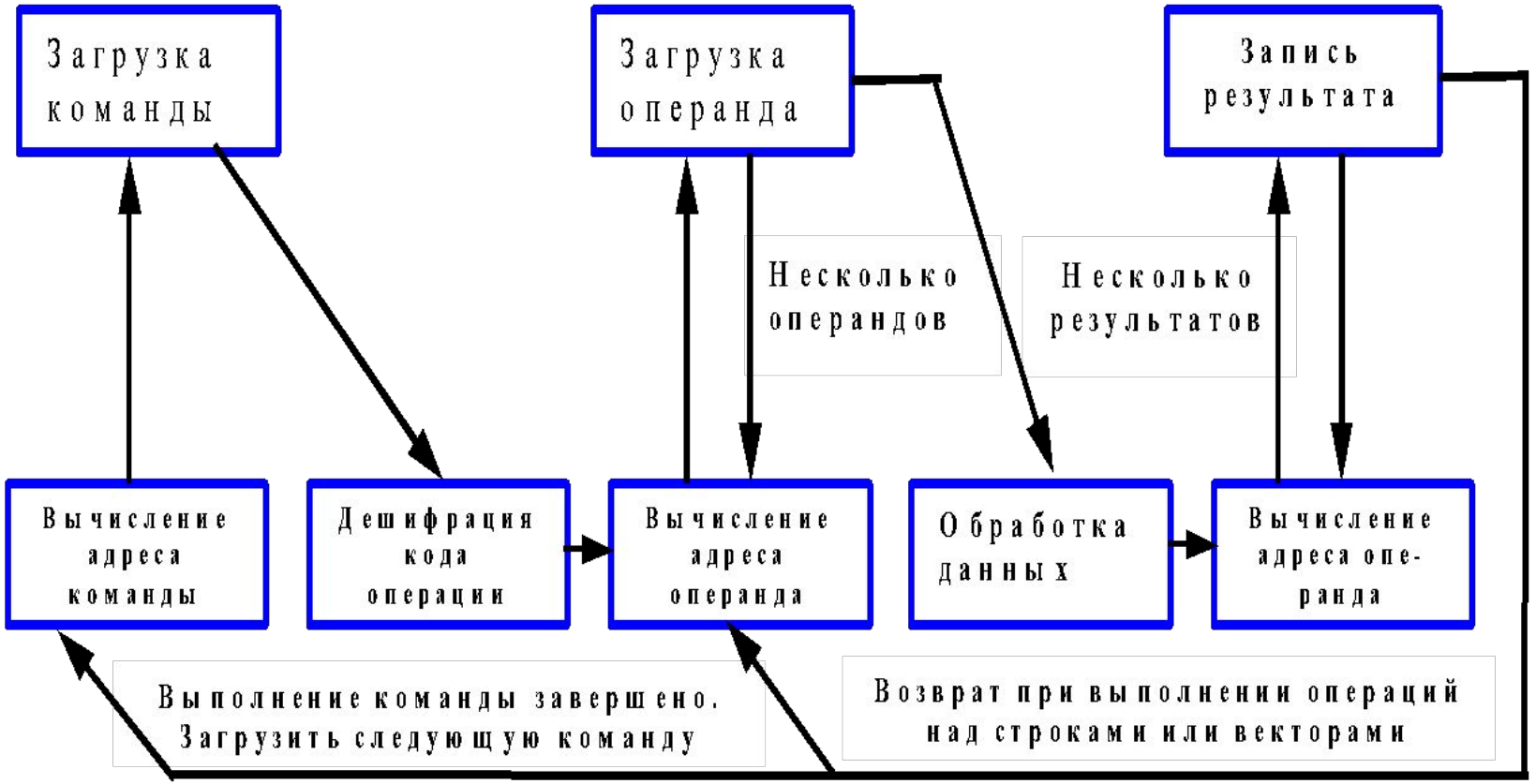


Рисунок 7.6 – Диаграмма состояний основного цикла обработки команды

Этапы диаграммы состояний

- **Вычисление адреса команды**

  При естественном порядке выполнения команд программы адрес следующей команды образуется в результате прибавления константы к адресу текущей команды

  Если, например, длина команды 16 разрядов, и такую же разрядность имеют слова в памяти, то адрес каждой очередной команды отличается от адреса предыдущей на 1.

Вычисления адреса команды

- Если же оперативная память организована таким образом, что адресуемой единицей данных является 8-разрядный байт, то следует адрес в PC увеличивать на 2.

Этапы диаграммы состояний

- **Загрузка команды**

  Считывание команды из ячейки памяти по адресу, заданному в PC, в регистр процессора
- **Дешифрация кода операции**

  Анализ кода операции и выяснение типа затребованной операции, количества и типов участвующих в ней операндов (или операнда)

Этапы диаграммы состояний

- **Вычисление адреса операнда**

  Если в операции используются операнды, хранящиеся в памяти или передаваемые через подсистему ввода-вывода, то на этой фазе определяются их "физические" адреса
- ***Загрузка операнда***

  Считывание операнда из заданной ячейки памяти или из модуля ввода-вывода

Этапы диаграммы состояний

- **Обработка данных**

  Выполнение операции, предусмотренной кодом операции
- **Запись результата**

  Запись результата операции в заданную ячейку памяти или передача в модуль ввода-вывода