COA - System Buses

# Key points:

* An **instruction cycle** consists of an instruction fetch, followed by zero or more operand fetch, followed by zero or more operand stores, followed by an interrupt check (if interrupts are enabled).
* The **major computer system components** (processor, main memory, I/O modules) need to be interconnected in order to exchange data and control signals. The most popular means of interconnection is the use of shared system bus consisting of multiple lines. In contemporary (modern) system, there typically is a hierarchy of buses to improve performance.
* **Key design elements for buses** include arbitration (whether permission to send signals on bus lines is controlled centrally or in a distributed fashion); timing (whether signals on the bus are synchronized to central clock or are sent asynchronously based on the most recent transmission); and width (number of address lines and number of data lines).

# Computer Components

* All contemporary computer design is based on concepts developed by John Von Neumann at the institute for advanced studies, Princeton. Such a design is referred to as *Von Neumann architecture* and is based on three key concepts.
  + Data and instruction are stored in a single read-write memory.
  + The contents of this memory are addressable by location, without regard to the type of data contained there.
  + Execution occurs in a sequential fashion (unless explicitly modified) from one instruction to another.

# Program Concept

* Hardwired systems are inflexible
  + Hardware can do different tasks, given correct control signals.
  + Instead of re-wired, supply a new set of control signals.
* Programs
  + Sequence of steps.
  + For each step, an arithmetic or logical operation is done.
  + For each operation, a different set of control signals is needed.

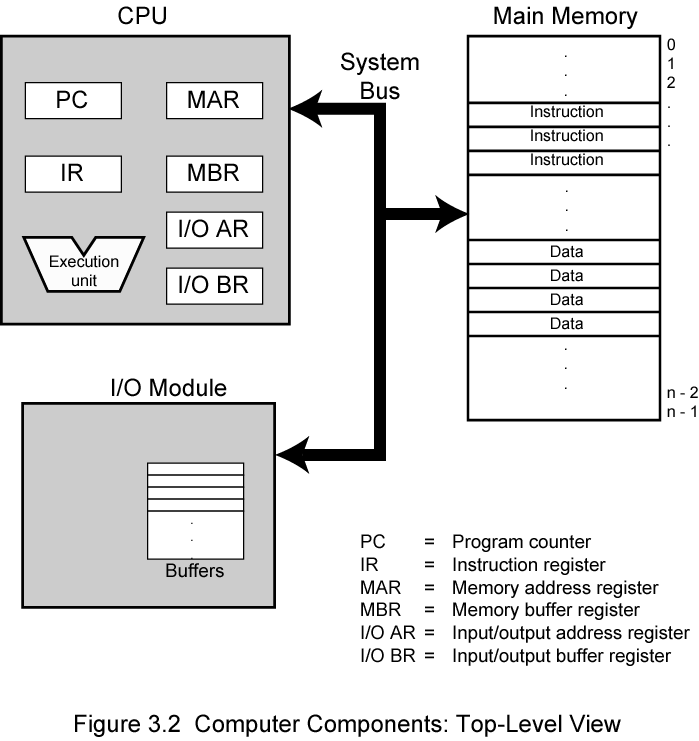
# Function of control unit

* For each operation, a unique code is provided
  + E.g., ADD, MOV
* A hardware segment accepts the code and issues the control signals

# Components

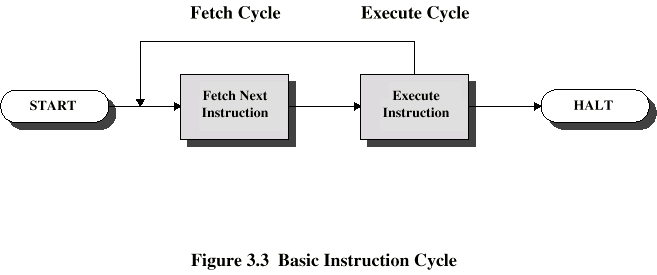
* The control unit (CU) and arithmetic and logical unit (ALU) constitute (create) the central processing unit (CPU).
* Data and instruction need to get into the system and as a result out
  + Input/output
* Temporary storage of code and results is needed
  + Main memory

# Computer Components: Top Level View



# 3.2 Computer function

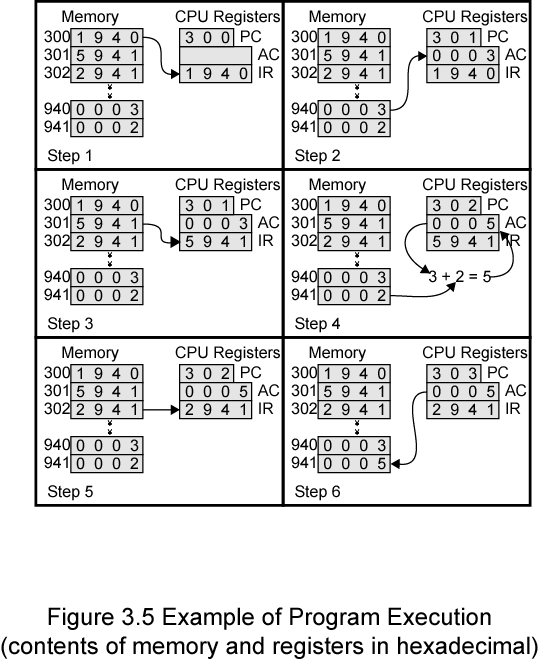
* Instruction processing consists of two steps: the processor reads (**fetch**es) instructions from memory one at a time, and **execute**s each instruction. Program execution consists of repeating the process of instruction fetch and instruction execution. The instruction execution may involve several operations and depends on the nature of the instruction.



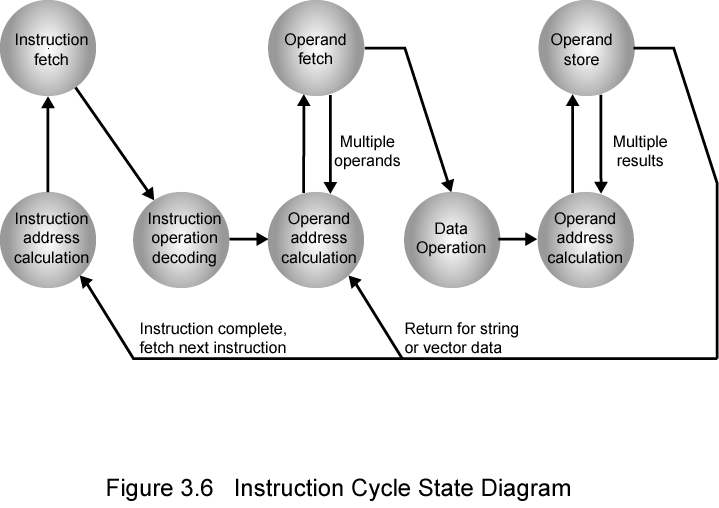
* Program Counter (PC) holds address of next instruction to fetch
* Processor fetches instruction from memory location pointed to by PC
* Increment PC
  + Unless told otherwise
* Instruction loaded into Instruction Register (IR). The instruction contains bits that specify the action the processor is to take.
* Processor interprets instruction and performs required actions

These actions fall into four categories.

1. **Processor – memory:** data may be transferred from processor to memory or from memory to processor.
2. **Processor – I/O module:** data may be transferred from or to peripheral device by transferring between the processor and an I/O module.
3. **Data processing:** the processor may perform arithmetic or logic operation on data.
4. **Control:** alteration the sequence of execution.



# Instruction Cycle State Diagram



# Method of Arbitration:

In all but the simplest systems, more than one module may need control of the bus. For example an I/O module may need to read or write directly to memory, without sending the data to the processor. As only one unit at a time can successfully transmit over the bus, some method of arbitration is needed.

Two methods of arbitration are: 1) Centralized 2) Distributed

## Centralized:

A single hardware device, referred to as a bus controller or arbiter, is responsible for allocating time on the bus. The device may be a separate module or part of the processor.

## Distributed:

Each module contains access control logic and modules act together to share the bus.

With both methods of arbitration, the purpose is to designate one device, either the processor or an I/O module, as master. The master may initiate a data transfer (e.g., read or write) with some other device, which acts as slave for this particular exchange.