Chapter Two

Top Level View of Computers



Prepared by: Esubalew Mulat

Target Group: 4th Year ECE Students

Material Primarily Adapted from the Works of W. Stallings

Addis Ababa Science and Technology University, October 2018 E.C.

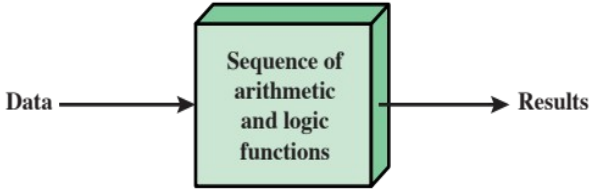
Topics

- Instruction Cycle
- Interrupts
- Interconnection Unit

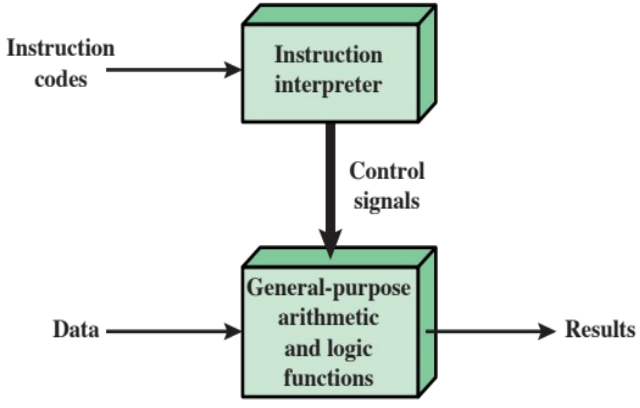
Instruction Cycle

- Contemporary computer designs are based on concepts developed by John von Neumann at the Institute for Advanced Studies, Princeton
- Referred to as the *von Neumann architecture* and is based on three key concepts:
  - Data and instructions 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 the next

Instruction Cycle

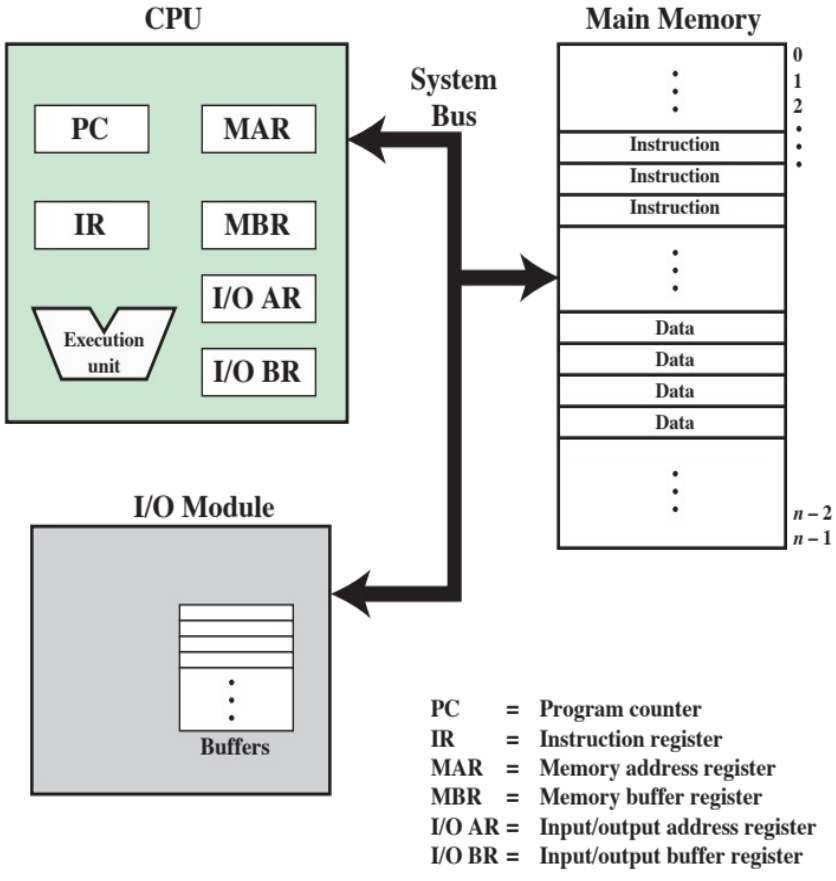


(a) Programming in hardware

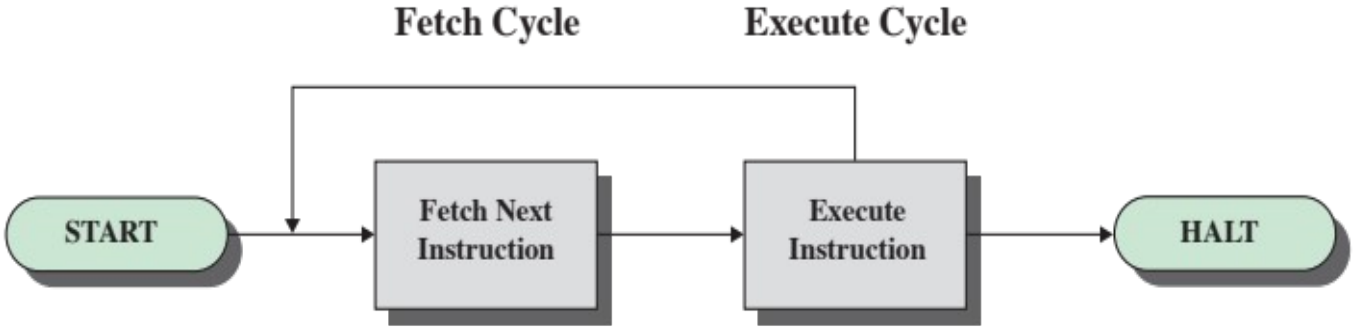


(b) Programming in software

Instruction Cycle



Instruction Cycle



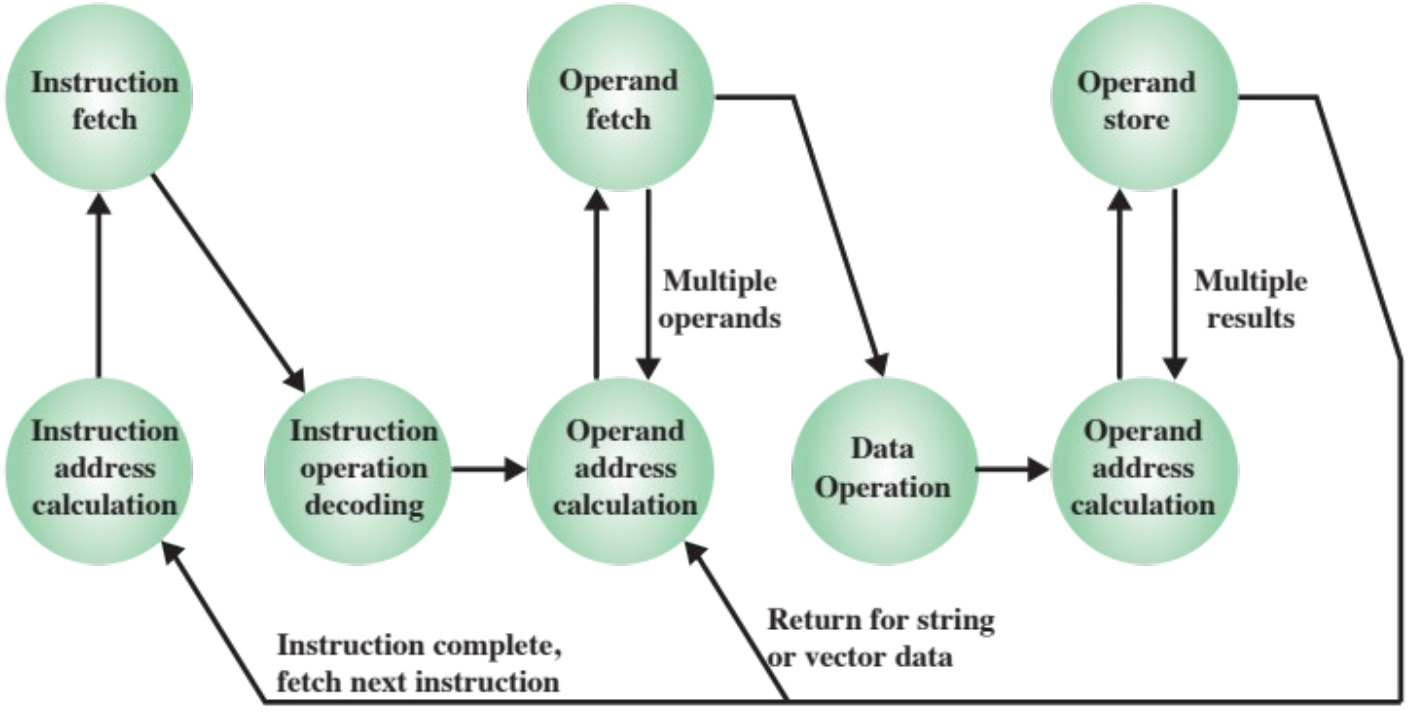
Fetch Cycle

- At the beginning of each instruction cycle the processor fetches an instruction from memory.
- The program counter (PC) holds the address of the instruction to be fetched next.
- The processor increments the PC after each instruction fetch so that it will fetch the next instruction in sequence.
- The fetched instruction is loaded into the instruction register (IR).
- The processor interprets the instruction and performs the required action.

Execute Cycle

- Processor-memory
  - data transfer between CPU and main memory
- Processor I/O
  - Data transfer between CPU and I/O module
- Data processing
  - Some arithmetic or logical operation on data
- Control
  - Alteration of sequence of operations
  - E.g. jump
- Combination of above

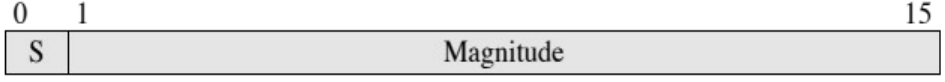
Instruction Cycle State Diagram



Hypothetical Machine



(a) Instruction format



(b) Integer format

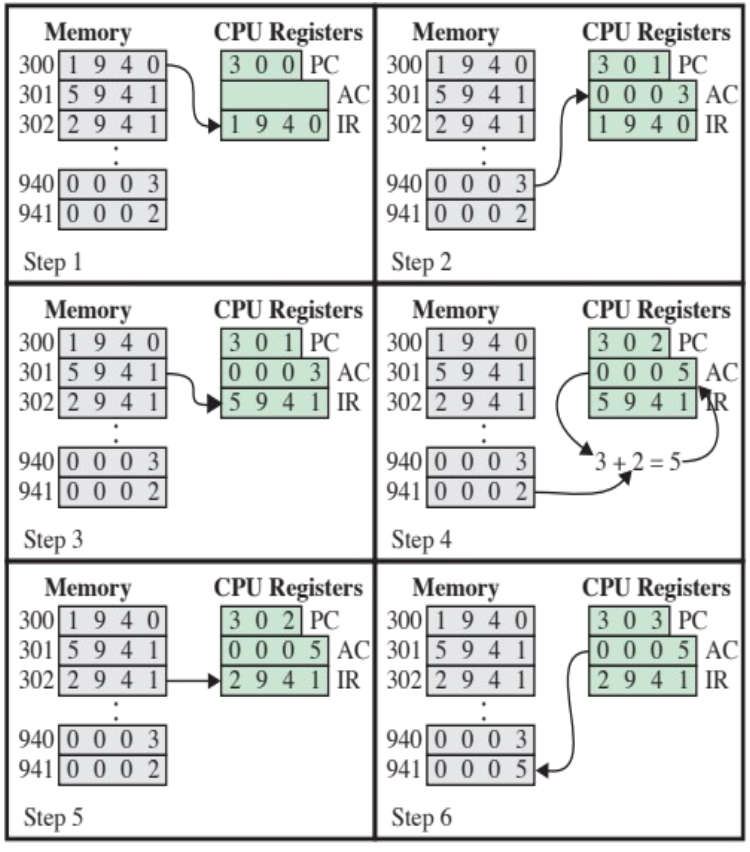
Program Counter (PC) = Address of instruction
Instruction Register (IR) = Instruction being executed
Accumulator (AC) = Temporary storage

(c) Internal CPU registers

0001 = Load AC from Memory
0010 = Store AC to Memory
0101 = Add to AC from Memory

(d) Partial list of opcodes

Hypothetical Machine…



Interrupts

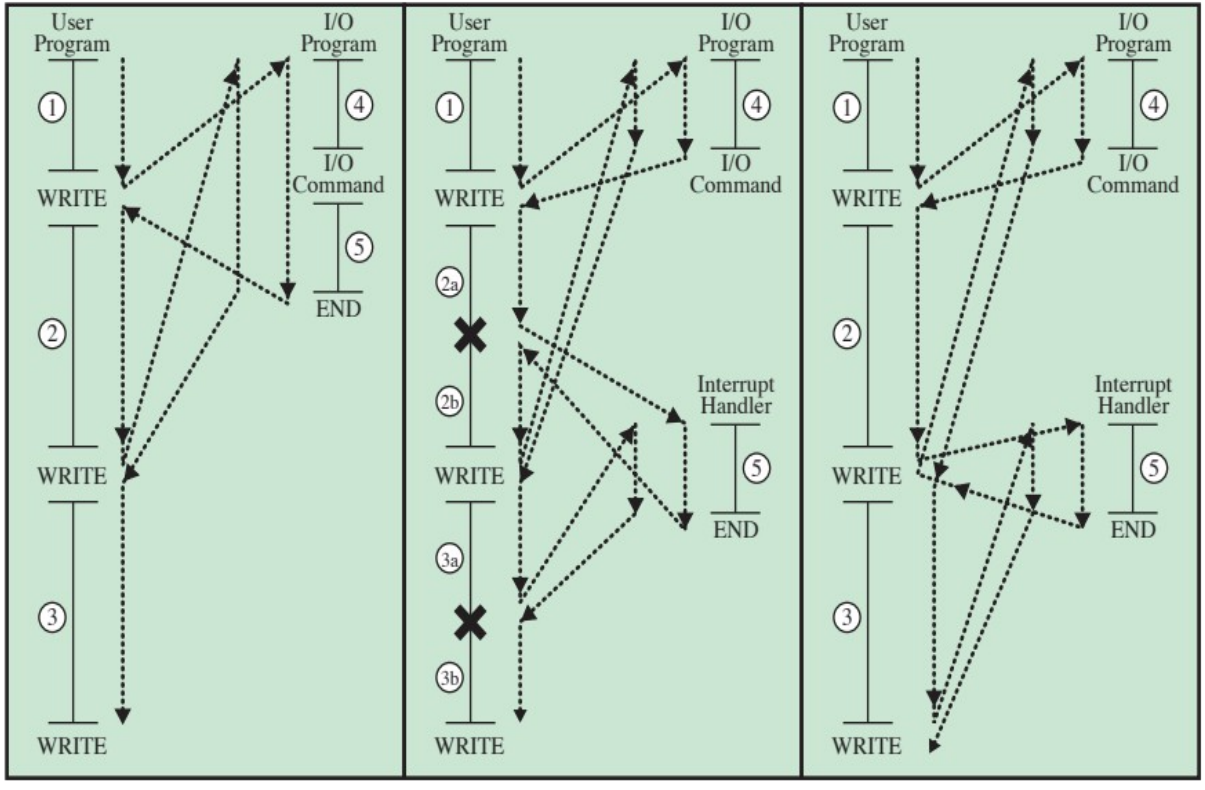
Interrupts

- Mechanism by which other modules (e.g. I/O) may interrupt normal sequence of processing.

- Interrupt Sources:

  - Program
    —e.g. overflow, division by zero
  - Timer
    —Generated by internal processor timer
    —Used in pre-emptive multi-tasking
  - I/O
    —from I/O controller
  - Hardware failure
    —e.g. memory parity error

Interrupts

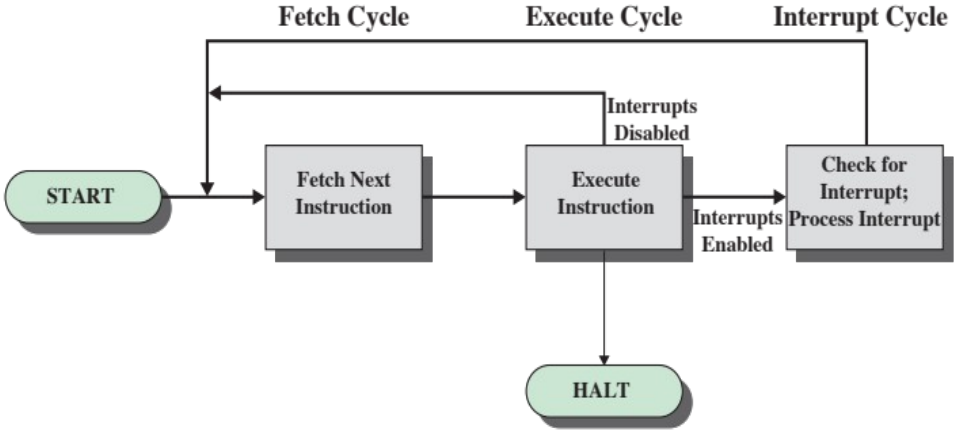


(a) No interrupts (b) Interrupts; short I/O wait (c) Interrupts; long I/O wait

✖ = interrupt occurs during course of execution of user program

Interrupt Cycle

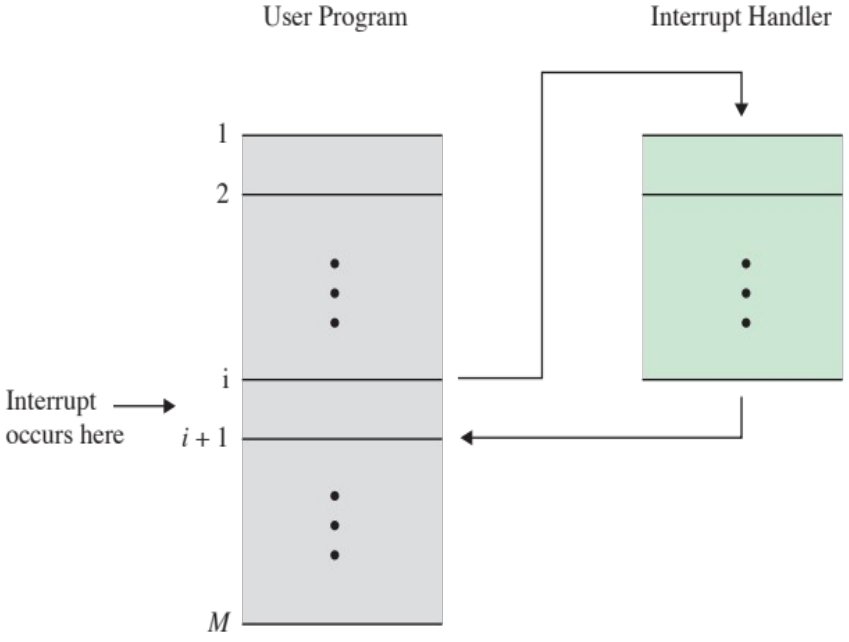
- Added to instruction cycle
  - Processor checks for interrupt
- Indicated by an interrupt signal
- If no interrupt, fetch next
- instruction



Interrupts…

If interrupt pending:

- Suspend execution of current program
- Save context
- Set PC to start address of interrupt handler routine
- Process interrupt
- Restore context and continue interrupted program



Multiple Interrupts

- Disable interrupts

—Processor will ignore further interrupts while processing one interrupt

—Interrupts remain pending and are checked after first interrupt has been processed

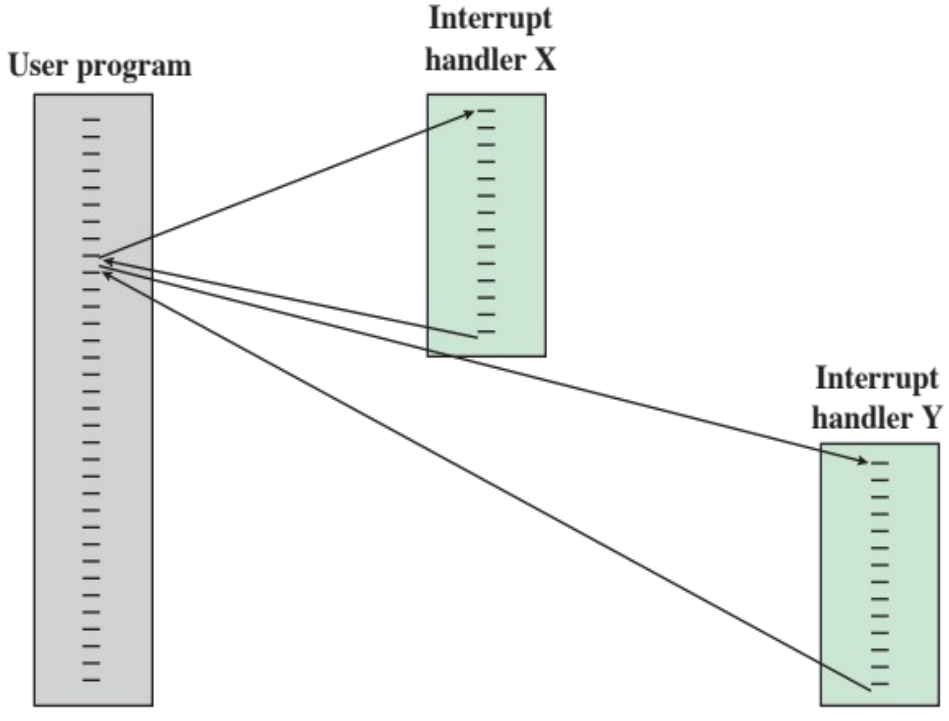
—Interrupts handled in sequence as they occur

- Define priorities

—Low priority interrupts can be interrupted by higher priority interrupts

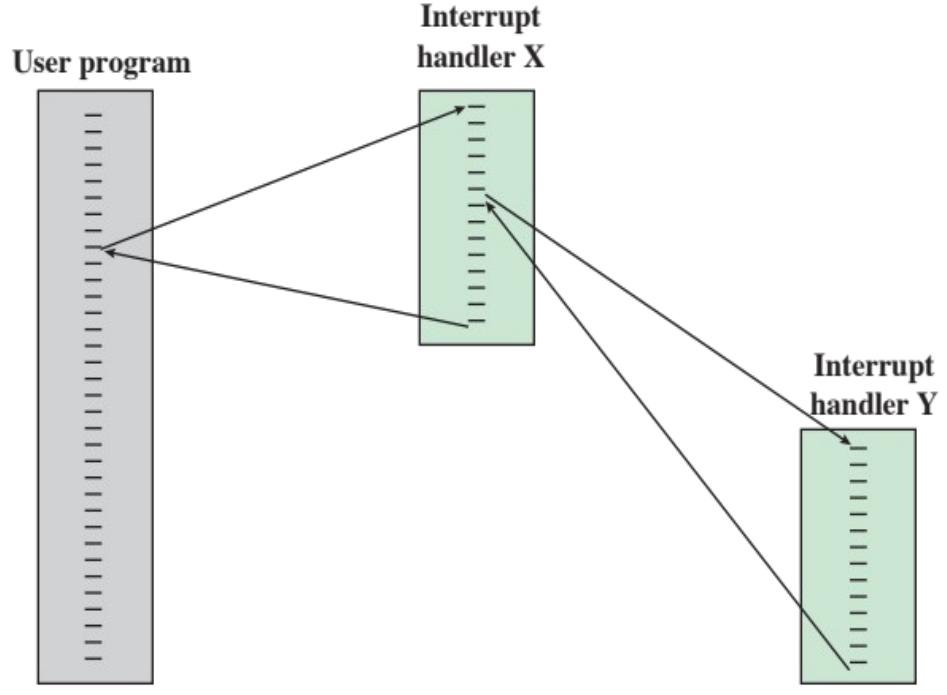
—When higher priority interrupt has been processed, processor returns to previous interrupt

Multiple Interrupts: Sequential



(a) Sequential interrupt processing

Multiple Interrupts: Nested



(b) Nested interrupt processing

Interconnection Units

- All the units must be connected
- Different type of connection for different type of unit
  - Memory
  - Input/Output
  - CPU

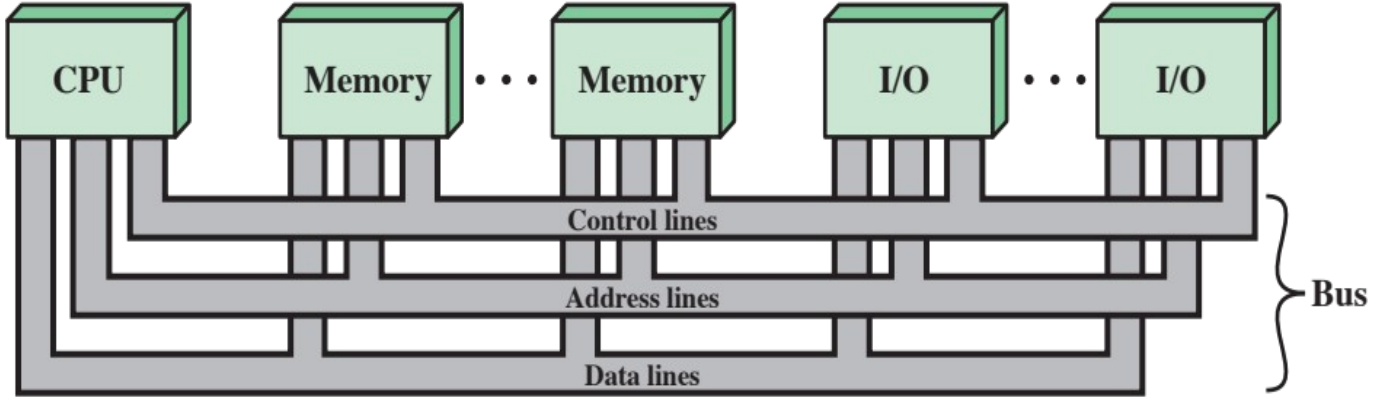
Bus

- There are a number of possible interconnection systems
- Single and multiple BUS structures were common for decades.

- Example:

- Control/ Address/ Data bus (PC)
- Unibus (DEC-PDP)

Bus…


CPU
Memory
Memory
I/O
I/O
Control lines
Address lines
Data lines
Bus

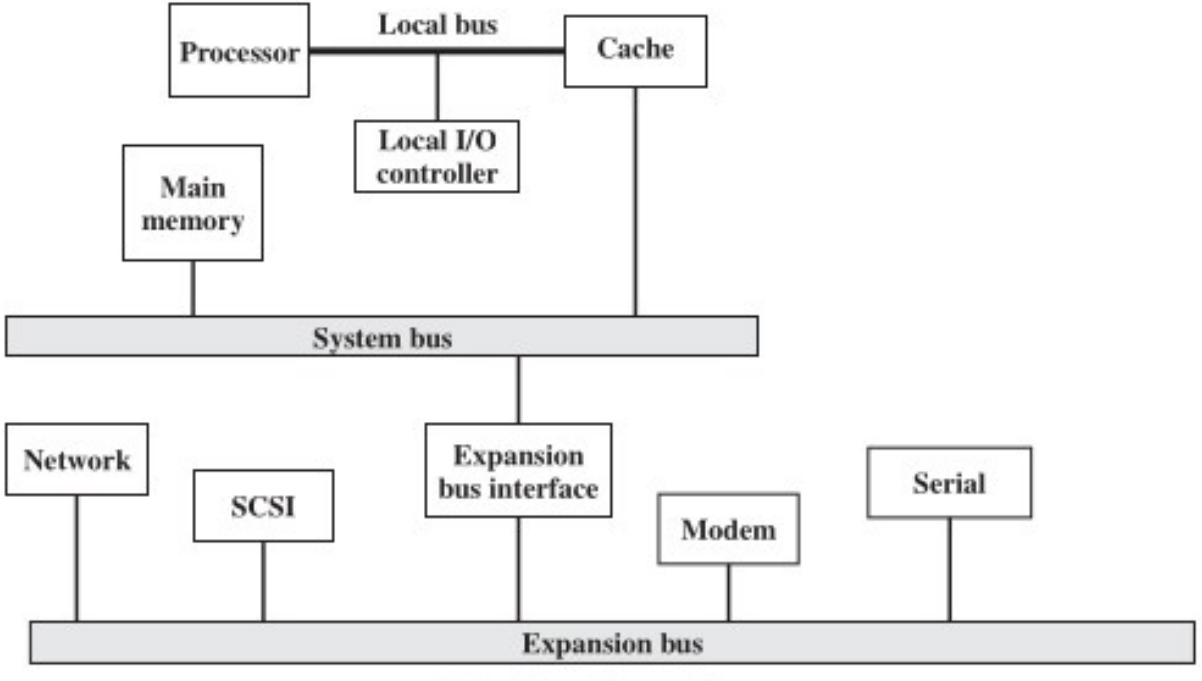
Bus

- If a great number of devices are connected to a single bus, performance will suffer.
- There are two main causes:

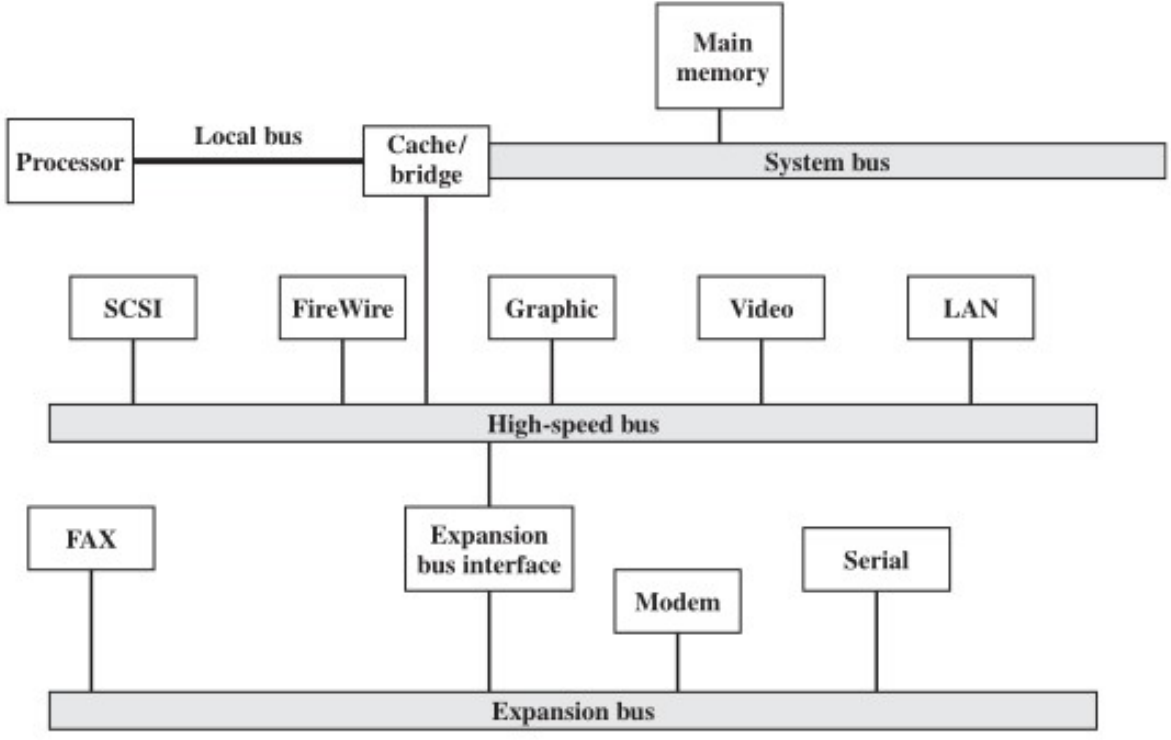
  - Higher Propagation delay.
  - BUS will be a bottleneck.

- Most computer systems use multiple buses, generally laid out in a hierarchy.

Bus…


Processor
Local bus
Cache
Local I/O controller
Main memory
System bus
Network
SCSI
Expansion bus interface
Modem
Serial
Expansion bus

Bus…


Main memory
Processor
Local bus
Cache/ bridge
System bus
SCSI
FireWire
Graphic
Video
LAN
High-speed bus
FAX
Expansion bus interface
Modem
Serial
Expansion bus

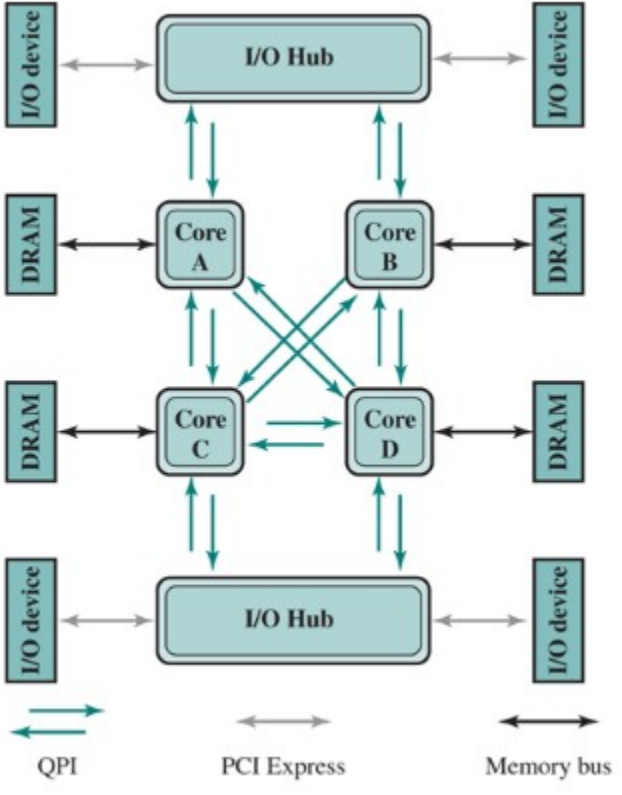
Bus…

Bus Design

- Type: Dedicated, Multiplexed
- Bus Width
- Method of Arbitration: Centralized, Distributed
- Timing: Asynchronous vs Synchronous
- Data Transfer Type

Point to Point Connections

- The shared bus architecture was the standard approach for decades.
- Contemporary systems increasingly rely on point-to-point interconnections.



Point to Point Interconnections

- Why point-to-point interconnects?
- Electrical constraints are encountered with increasing the frequency of wide synchronous buses.
- At higher and higher data rates, it becomes increasingly difficult to perform the synchronization and arbitration functions in a timely fashion.
- With the advent of multicore chips, it was found that the use of a conventional shared bus chip magnified the difficulties of increasing bus data rate and reducing bus latency.
- Point-to-point interconnect has lower latency, higher data rate, and better scalability.

Materials to Watch

- https://www.youtube.com/watch?v=jFDMZpkUWCw

Thank you,

Questions?