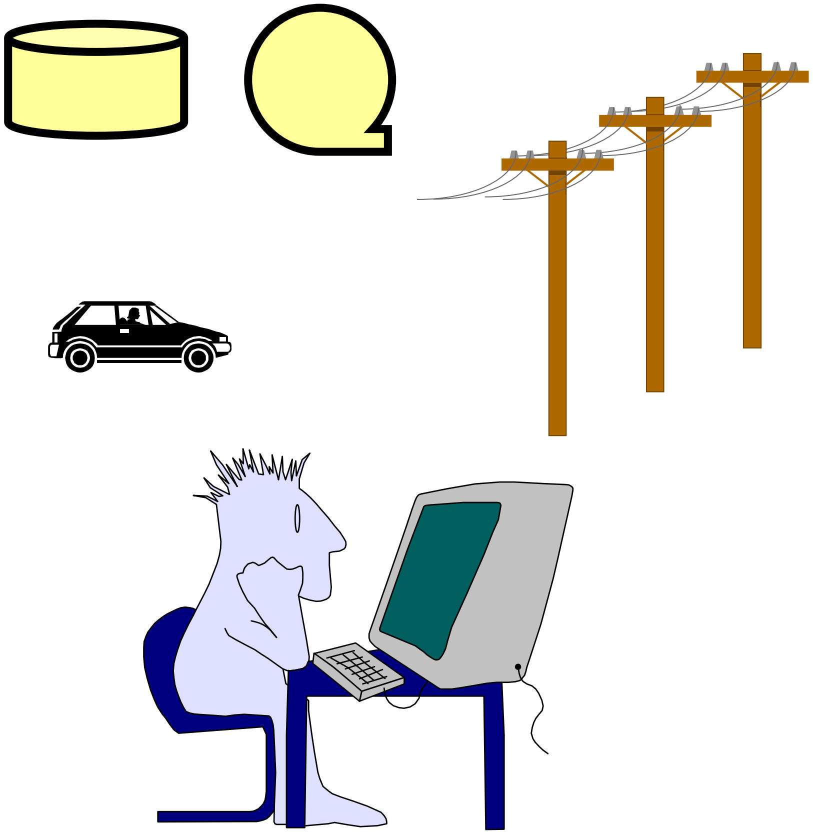
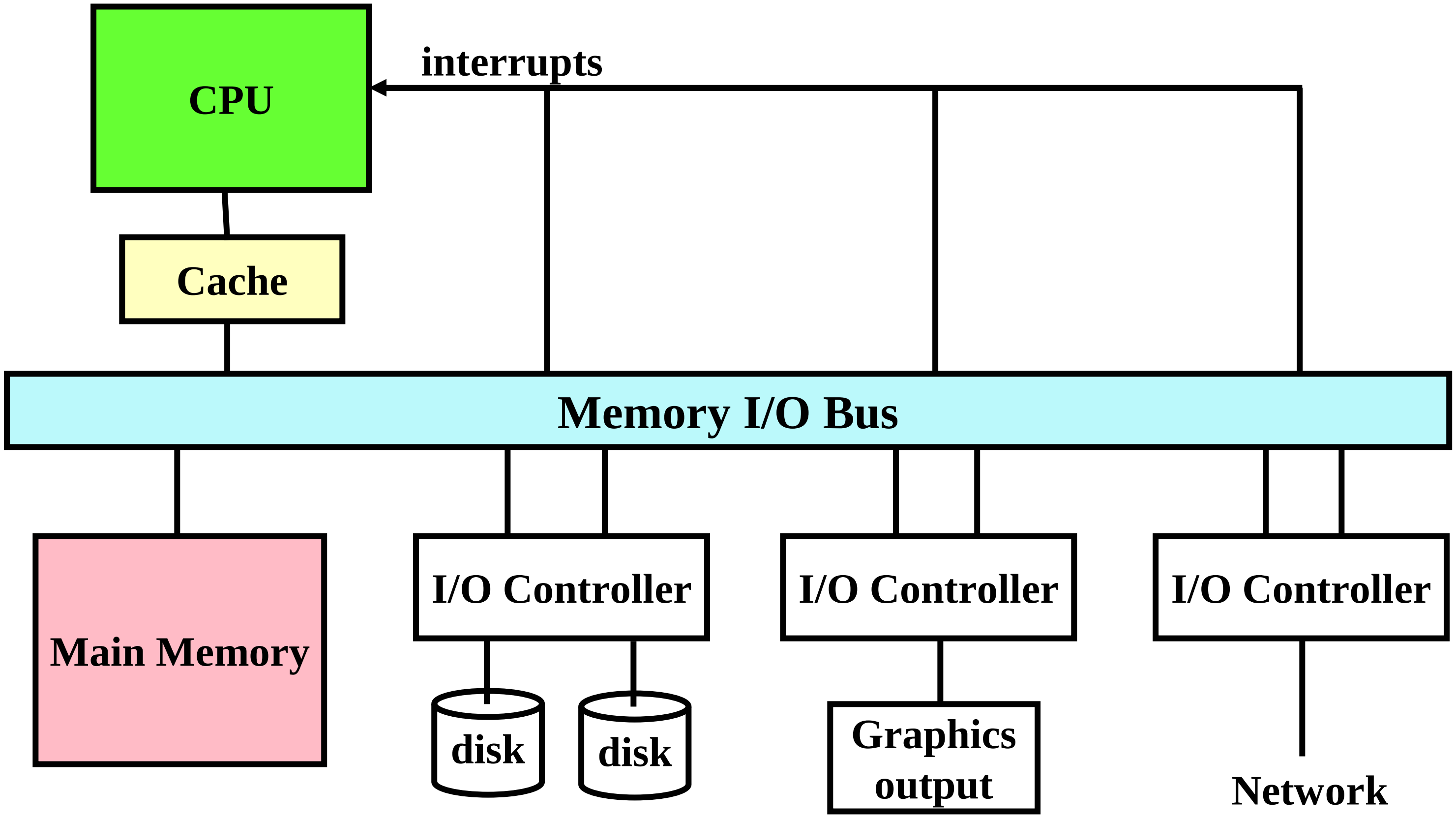
Input/Output (I/O)

- Secondary storage
  - disks, tapes, CD ROMs
- Communication
  - networks
- ‘Real world’ interface
  - temperature, pressure, position, velocity, voltage, current,…
- Human interface
  - video, audio, keyboards, ...



I/O Organization Today



Features of I/O Devices

- **Behavior**
  - **Input, Output, Storage**
- **“Partner”**
  - **What’s on the other end - human or machine**
- **Data Rate**
  - **Peak transfer rate between machine and partner**
  - **Examples:**
    - Keyboard: 0.01KB/sec
    - Laser Printer: 200KB/sec
    - Modem: 8KB/sec
    - Floppy disk: 100KB/sec

I/O Devices
Characterize by Latency, Bandwidth, Block Size

Device	Latency	Peak BW	Avg BW	Xfer Size
Tape	300s	1Mb/s		
Disk	10ms	32Mb/s	32Kb/s	100B
			16Mb/s	100KB
Network	0	100Mb/s	10Kb/s	1KB
CRT	16ms	1.5Gb/s	1.5Gb/s	3MB
Audio	200μs	100KB/s	100KB/s	

peak BW >> avg BW ⇒ path sharing

high BW (CRT) requires specialized hardware

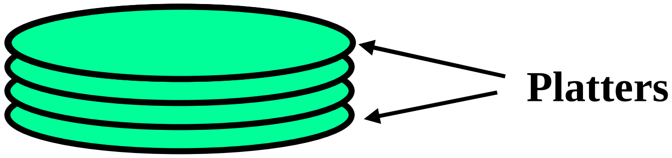
high latency devices can be handled by software

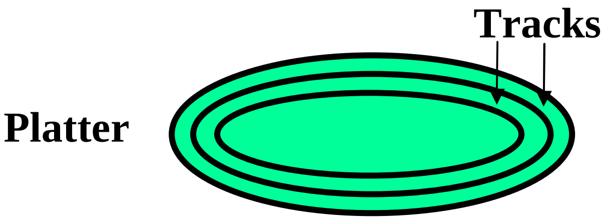
streaming devices have constant BW (CRT and audio)

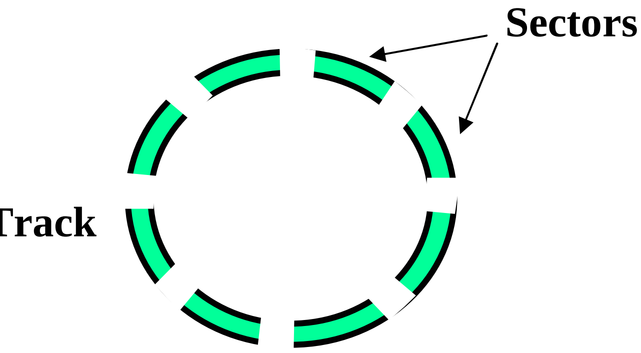
disk can be I/O or bandwidth limited

devices may be block access or streaming

Magnetic Disk Drives







- <1” to 8” diameter
- up to 10000 RPM
- 1000s of tracks per surface
- 100s of sectors per track
- Today: 512 bytes per sector

Gigabytes per surface

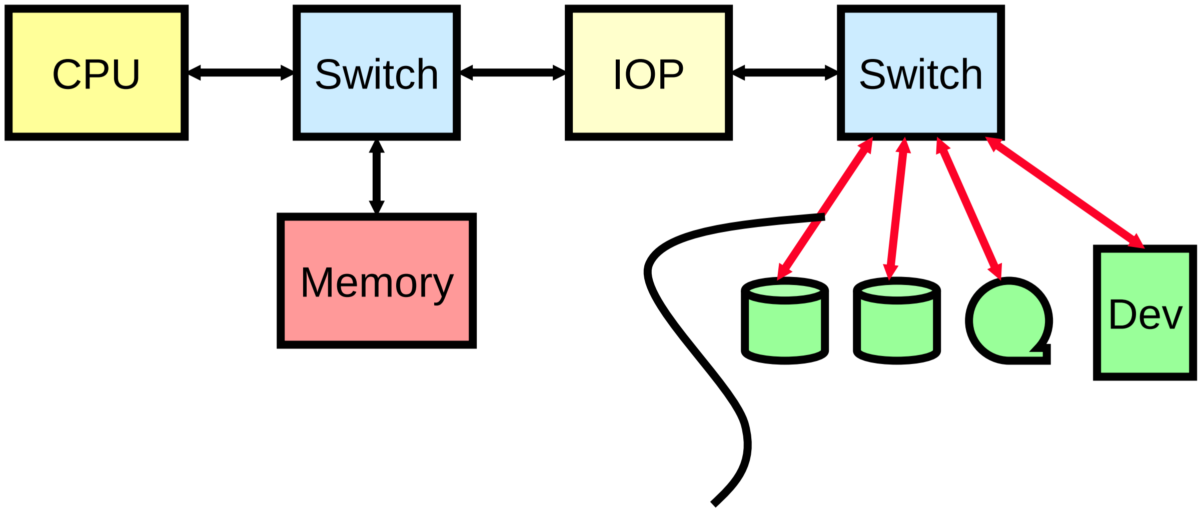
Disk Access

- Seek
  - Move head over proper track
  - Depends on previous head placement
  - Typically 8-10ms
- Rotational Delay
  - Rotate sector under head
  - On average - must rotate 1/2 way
- Controller overhead
- Transfer rate

Magnetic Storage Technology

- **Floppy disks - 1.44MB**
- **Zip disks - 100MB**
- **Removable “Hard” Disks - 1 GB capacity**
- **Hard disks - 200GB**
  - **Bigger, better density, higher data rate, more platters**
- **Tape**
  - **Archival, slower, high density (30+ GB)**

Alternative I/O Architecture

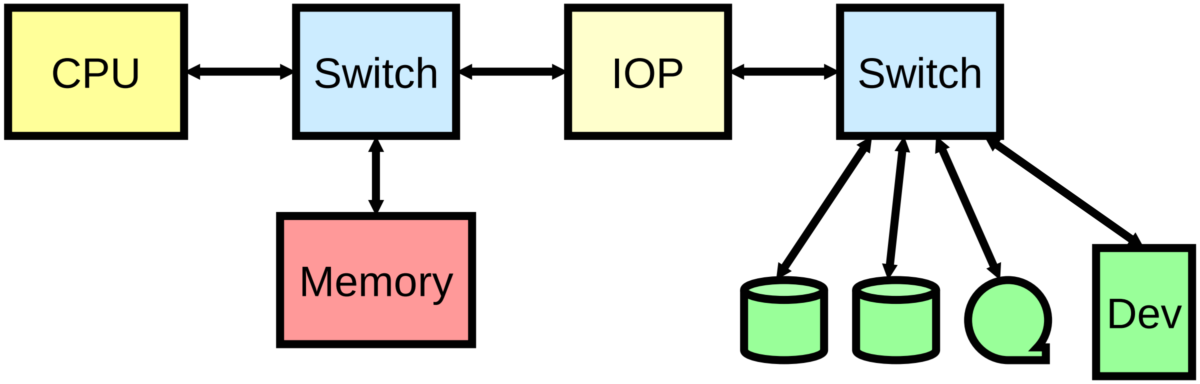


Standard I/O Interface

Variables:

- Transfers by CPU or IOP (DMA)
- Notification by *polling* or *interrupt*
- Bandwidth of data paths
- Locations of buffers
- Division of function (moving to peripherals)
- Interface standard
- Protocol

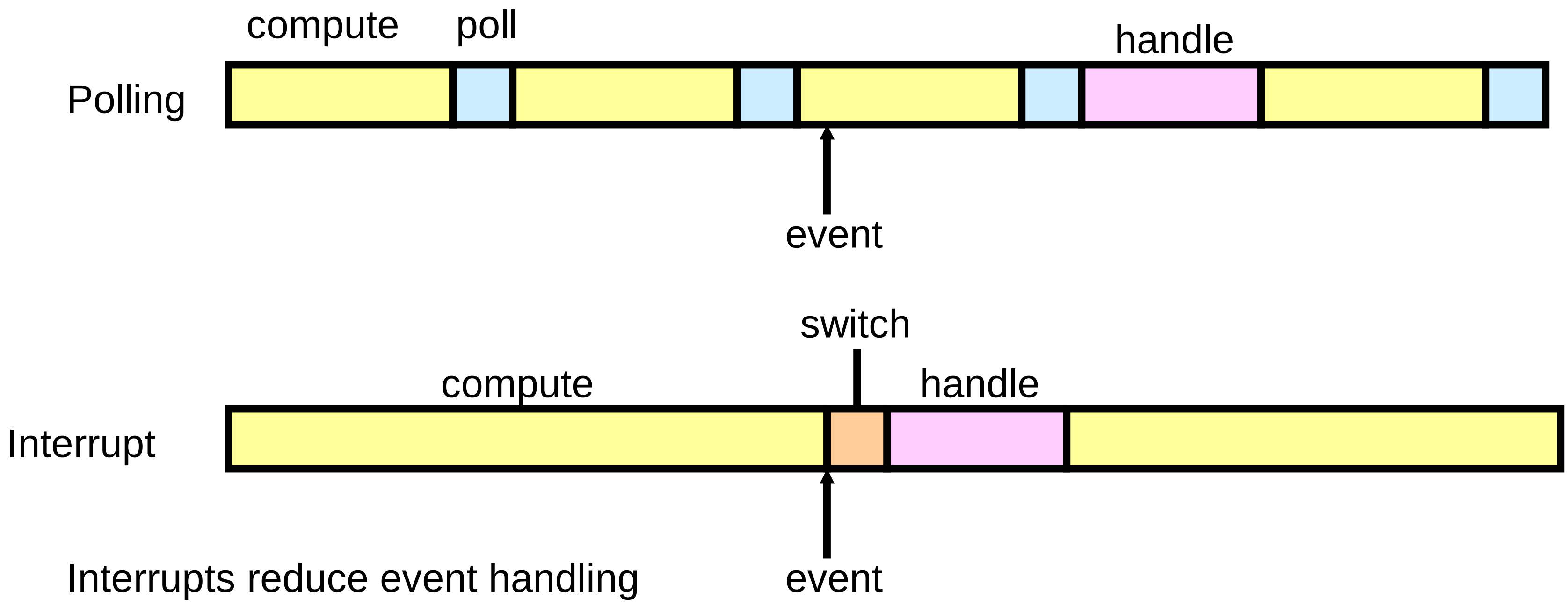
A Typical Operation



A typical *operation*

- CPU asks IOP to perform disk sector read
- IOP schedules request
- IOP passes request to disk
- Disk reads data
- Disk transfers data to memory under IOP control *(DMA)*
- IOP notifies CPU that transfer is complete *(interrupt)*

Interrupts vs Polling



Interrupts reduce event handling latency and eliminate overhead of polling.

Are there cases where polling does better?

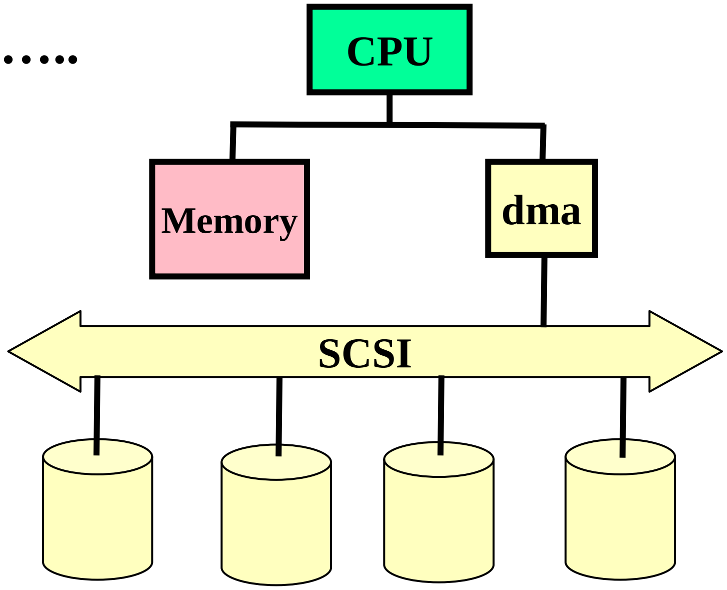
Programmed Transfer, DMA, and IOPs

- Consider loading a 1K disk block from a 4MB/s disk
  - 1µs between 32-bit words
  - 256 transfers
- CPU could perform all 256 transfers
  - poll to wait for each word ready
  - interrupt on each word
  - little time for anything else
- Direct-Memory Access (DMA)
  - CPU sets up start and length registers
  - DMA controller sequences the individual words
  - CPU notified (poll or interrupt) upon completion
- Input/Output Processor (IOP)
  - Queue multiple requests
  - IOP schedules the requests and sets up the individual DMA operations
  - CPU notified per request or when all are finished

Advanced Storage Systems

- Processors getting faster (performance doubling every 18 months)
- Disks also getting faster (seek time improves 7%/year)

- Instead of building one large disk…..
  - Build a disk array!



Reliability

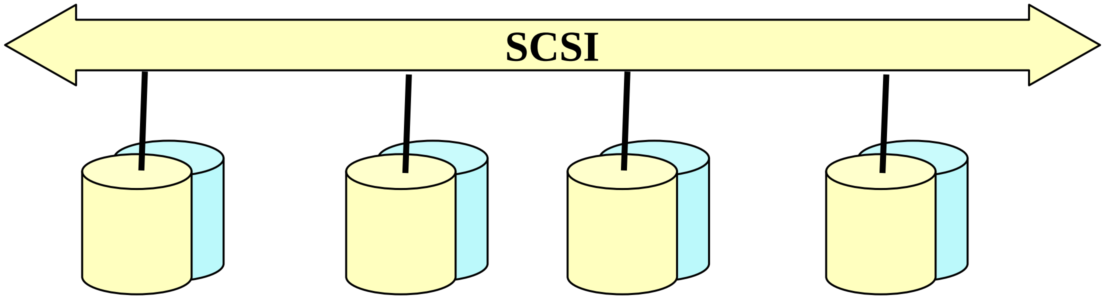
- MTTF = Mean time to failure
  - On the order of 50,000 hours for a disk drive (= 5.7 years)

- MTTF for disk array = $\text{MTTF}_{\text{disk}}$ / # disks
  - 500 hours for 100 disks (= 21 days!)

- Need to make data available at all times
  - Use redundancy!

RAID

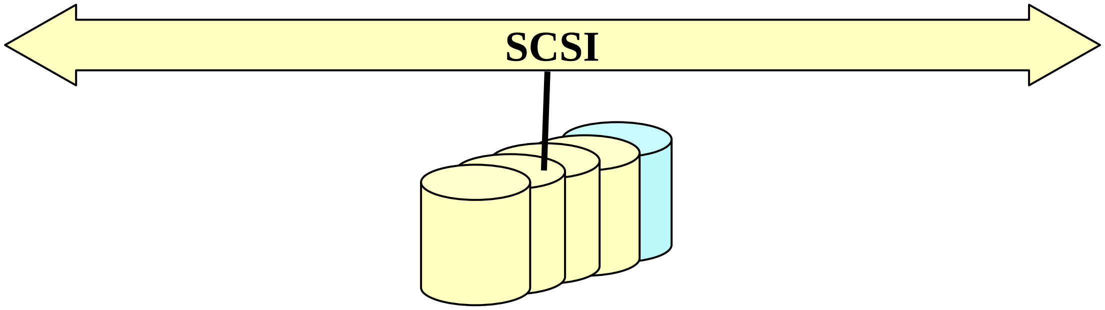
- **Redundant Array of Inexpensive Disks**
  - **Encode data redundantly**
  - **If a disk fails, the data is still available**



- **Mirrored Disks**
- **MTTF > 500 years**
- **Disk overhead = 100%**

RAID-IV

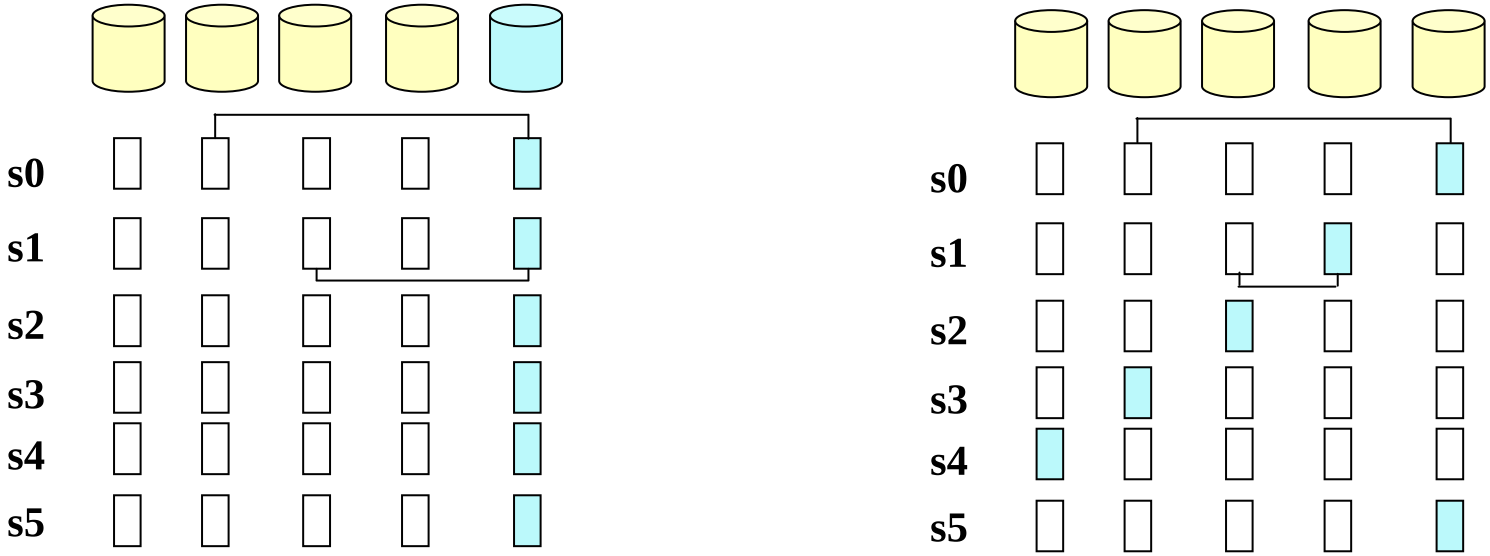
- Share check disk across multiple data disks
  - Encode using parity
  - Writes require read from check disk



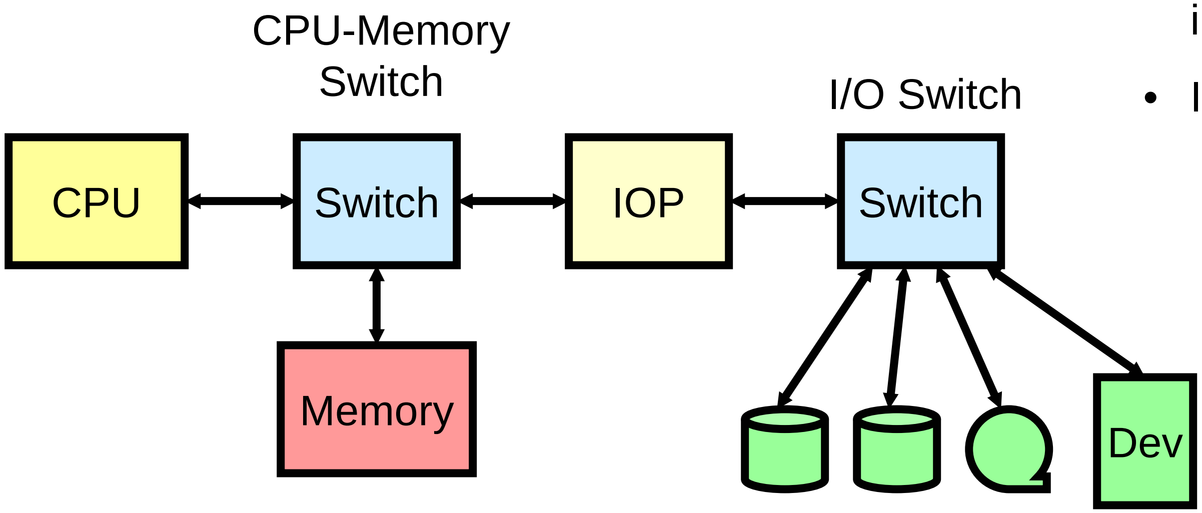
- Parallel reads
- MTTF > 40 years
- Disk overhead = 25%

RAID-V

- Rotating parity
  - No single check disk
  - Multiple writes proceed simultaneously



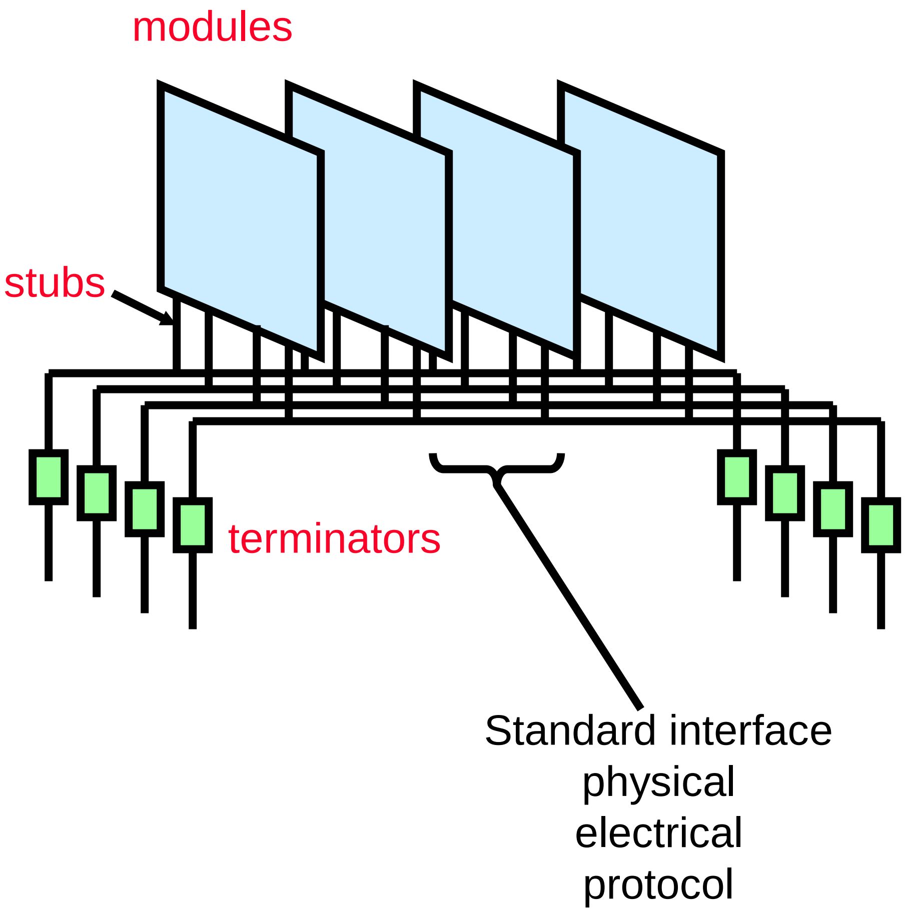
Switches - Buses and Networks



- Switches used in several places in a typical computer
- I/O switch provides
  - standard interface to peripheral devices
  - data bandwidth to/from memory
  - must provide adequate bandwidth for fastest peripherals
  - must not burden slow peripherals
- Historically switches were implemented with buses. Point-to-point networks now more attractive

Bus Basics

- A special case of a communication network comprising
  - A single physical channel (bunch of wires) with multiple senders and receivers
  - A protocol obeyed by the senders and receivers
- Advantages
  - simplicity, broadcast, serialization
- Disadvantages
  - Multi-drop wires slow and electrically difficult
  - serialization



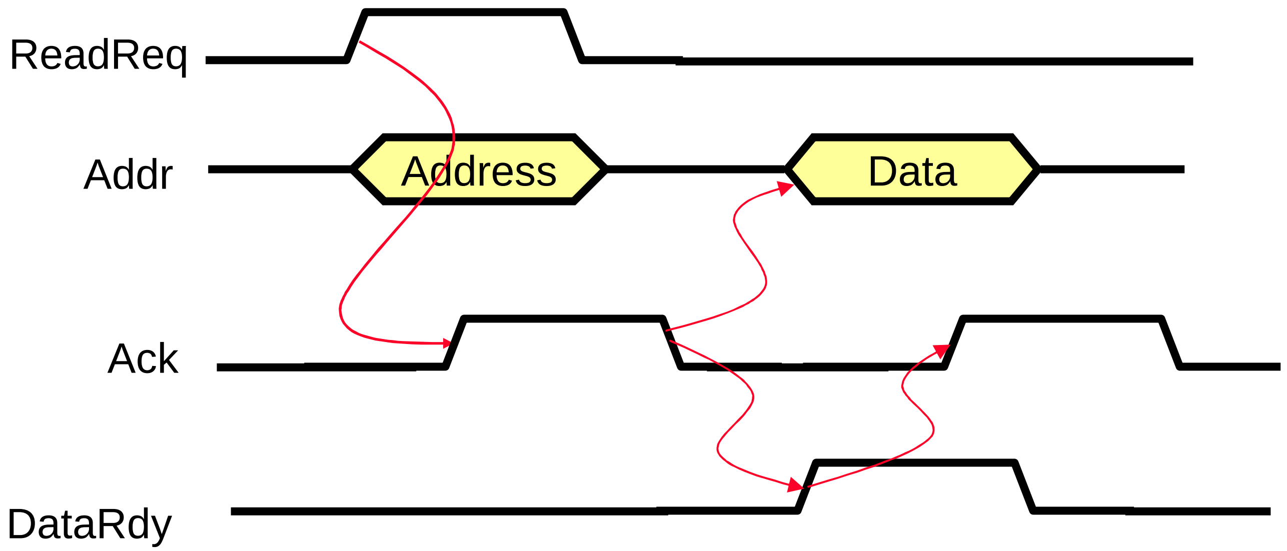
modules

stubs

terminators

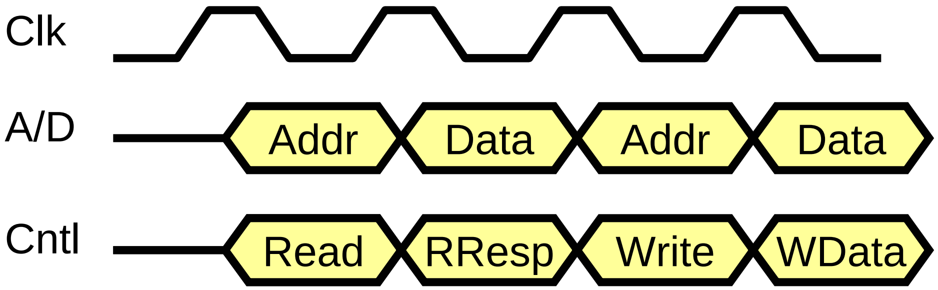
Standard interface
physical
electrical
protocol

Bus Cycles and Transactions



- A bus *cycle* transfers one datum (address or data)
- A bus *transaction* completes an operation (read or write)
- An *asynchronous* bus signals each cycle when ready
  - can slow down to accommodate slow peripherals
- A synchronous bus performs a cycle every clock
- Lines on a bus can be dedicated or the lines can be multiplexed

Synchronous Transactions



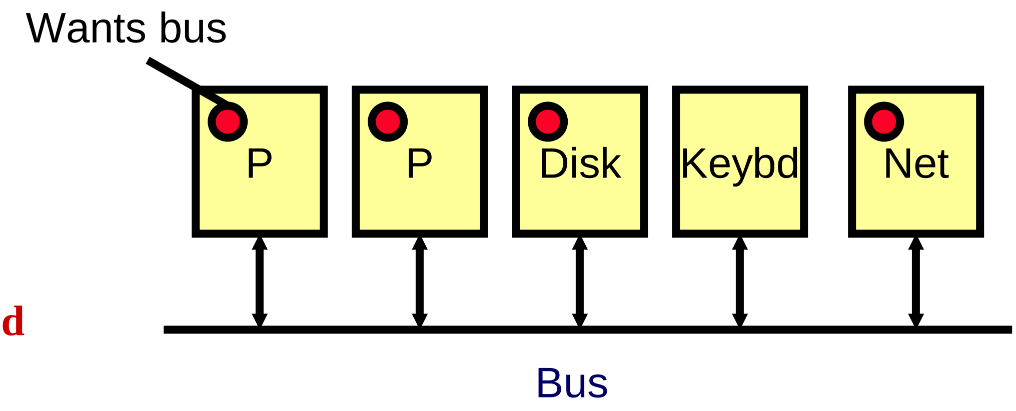
- Shared synchronous clock
- Advantages
  - Easier to define protocols
- Disadvantages
  - Must have shared synchronous clock
  - Difficult to run devices at different speeds
- Synchronous typically used between processor and memory

Bus Bandwidth vs. Latency

- Increasing Bus Bandwidth
  - Make data bus wider
  - Separate address and data wires
  - Block Transfers
  - More buffering (pipelining)

- BUT - these can have adverse affect on bus latency

Bus Protocols



- Protocols determine
  - the transactions that are supported
  - the timing of their cycles
  - how modules are addressed
  - allocation of resources
- Arbitration
  - determines what module gets to use the bus
    - modules make requests
    - arbiter grants the bus to one requester
      - fixed priority
      - round robin
      - random
- Preemption
  - interrupt long transaction to run more critical operation
- Arbitration is often pipelined with bus use

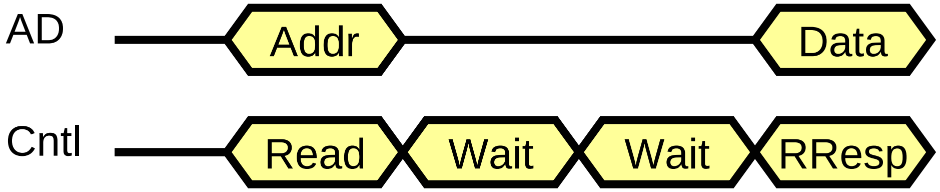
Arbitration Policies

- Daisy Chain
  - High priority devices see grant lines first
- Centralized
  - One bus master sees all requests
- Distributed Self Selection
  - Each device sees all requestors, decide in parallel
- Collision detection
  - Ethernet

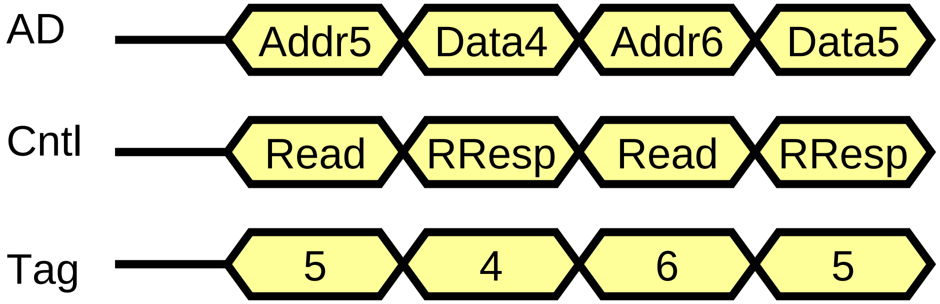
- Issues
  - Priority
  - Fairness (non-starvation)

Split Transaction Buses

- **Separate request and reply transactions**
- **Decouples throughput of bus from latency of module**
- **Accommodates slow modules without tying up bus**
- **Requires method to identify responses**
  - **tags - allows responses in any order**
  - **in order response - can delay transactions behind slow responder**



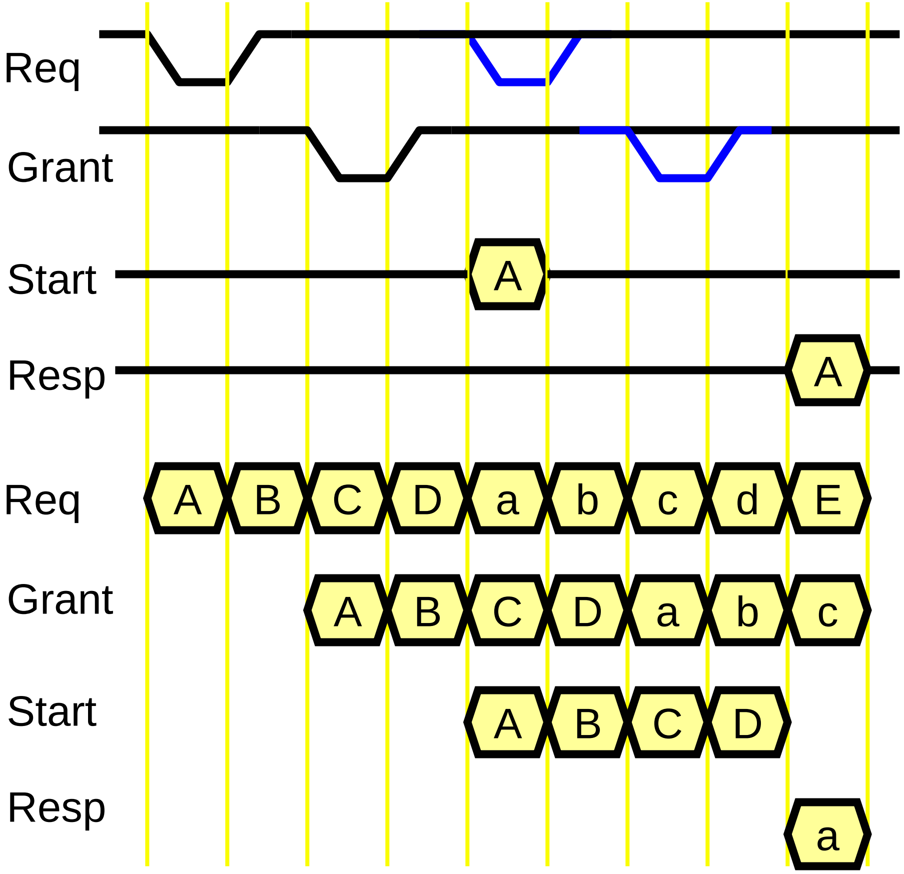
Bus idle during transaction latency



Bus handles other transactions while waiting

Bus Pipelining

- **Typical latency from request to bus cycle may be 4 cycles or more**
- **Make request in advance**
  - **request bus 4 cycles before you will need it**
    - if you know then
  - **reduces latency due to bus arbitration**
  - **allows back-to-back bus cycles**



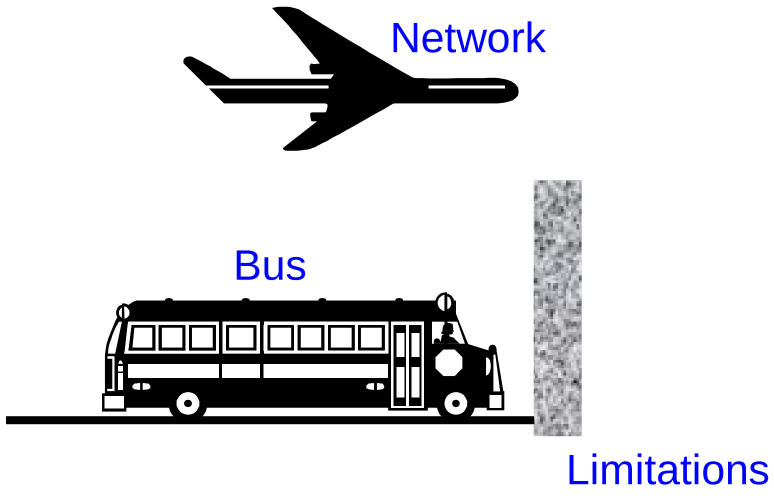
Examples of Busses

	MicroChannel	PCI	SCSI 2	Sun P/M
Data Width	32 bits	32-64 bits	8-16 bits	245 bits
Clock rate	Asynchronous	33MHz	10MHz or Asynch.	48MHz
Bus Masters	Multiple	Multiple	Multiple	Multiple
Peak BW	75MB/s	132MB/s	20MB/s	1200Mb/s

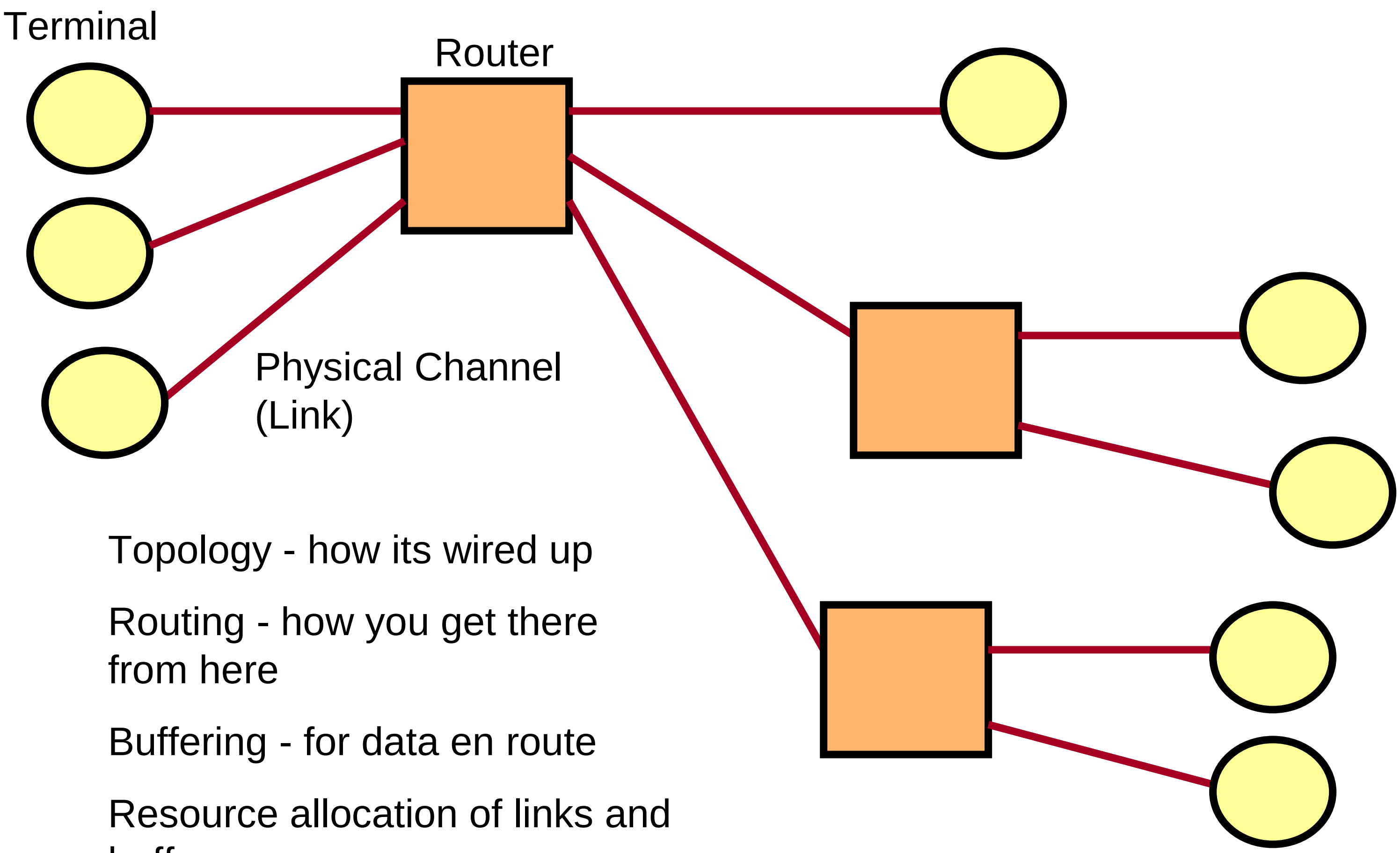
- **I/O Busses tend to have standards**
  - **Multiple vendors supply devices**
- **P/M busses tend to be proprietary**
  - **Faster speeds**

Why Buses are a Bad Idea

- Electrically difficult
  - stubs and spaces, speed limited by size
- Limited scaling
  - load and delay grow with N
- Inherently sequential
  - throughput 1/N
- Arbitration slow - global
- Can’t exploit locality



Point-to-Point Network Basics



Topology - how its wired up

Routing - how you get there from here

Buffering - for data en route

Resource allocation of links and buffers