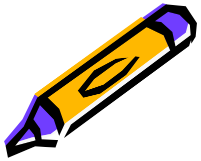
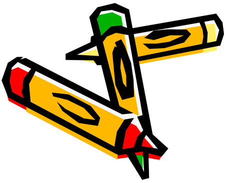
Computer Architecture

Hossein Asadi
Department of Computer Engineering
Sharif University of Technology
asadi@sharif.edu



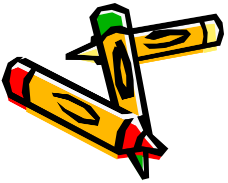
Today's Topics

- I/O & Mass Storage
  - Busses
  - I/O Handshaking
  - Disks (HDDs vs. SSDs)
  - Introduction to Non-Volatile Memories

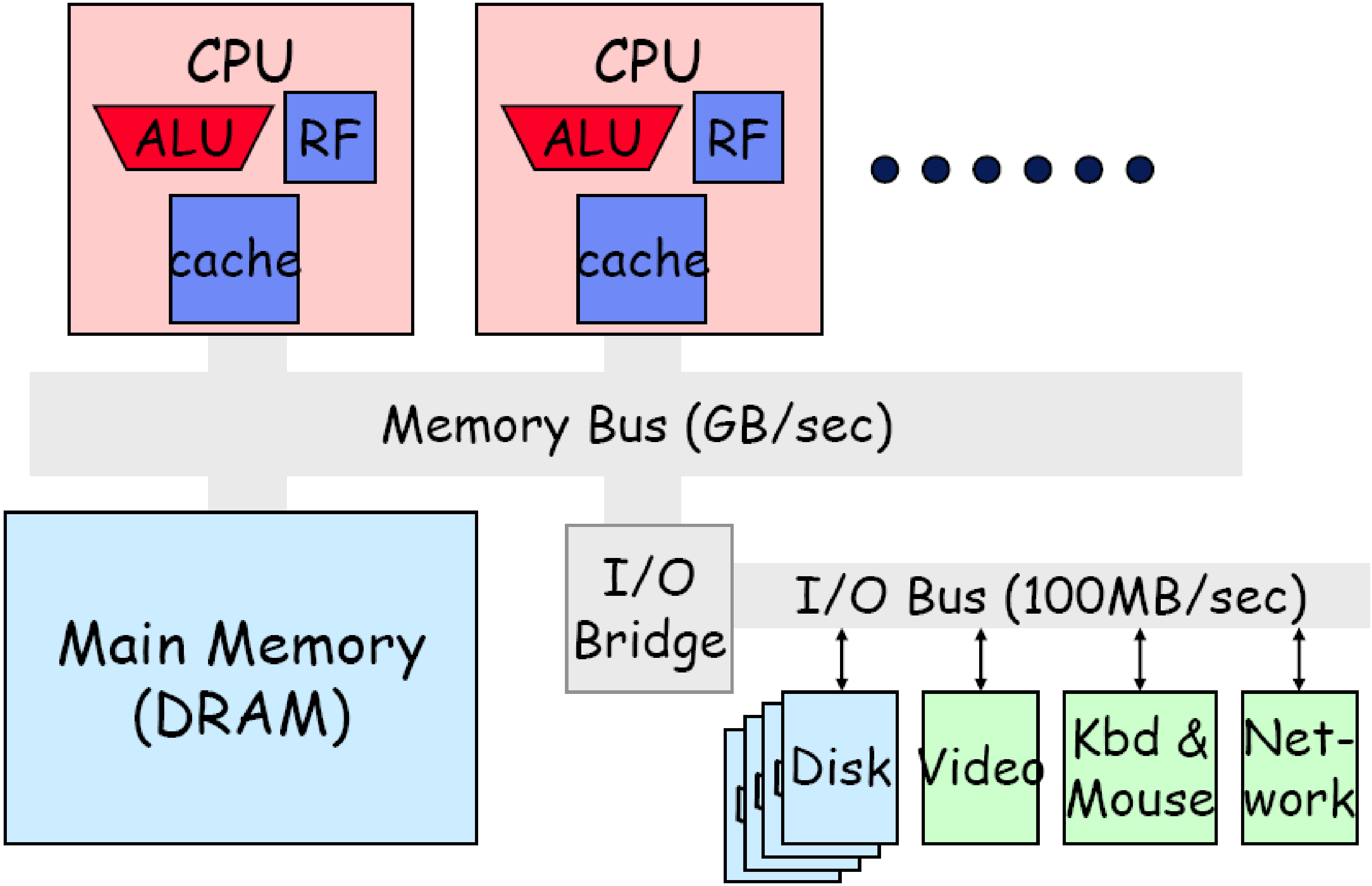


Copyright Notice

- Parts (text & figures) of this lecture adopted from:
  - Computer Organization & Design, The Hardware/Software Interface, 3rd Edition, by D. Patterson and J. Hennessey, MK publishing, 2005.
  - "Intro to Computer Architecture" handouts, by Prof. Hoe, CMU, Spring 2009.
  - "Computer Architecture & Engineering" handouts, by Prof. Kubiatowicz, UC Berkeley, Spring 2004.
  - "Intro to Computer Architecture" handouts, by Prof. Hoe, UWisc, Spring 2019.
  - "Computer Arch I" handouts, by Prof. Garzarán, UIUC, Spring 2009.
  - "Intro to Computer Organization" handouts, by Prof. Mahlke & Prof. Narayanasamy, Winter 2008.



Computer Organization



Input/Output

- Fact
  - Input/outputs very slow devices
    - Average response time: few mili-seceonds
  - CPUs very fast devices
    - Average running time: nano-seconds
- Question:
  - How Input/Output devices are connected to CPUs?
    - I/O controller (also called I/O bridge)

Input/Output (cont.)

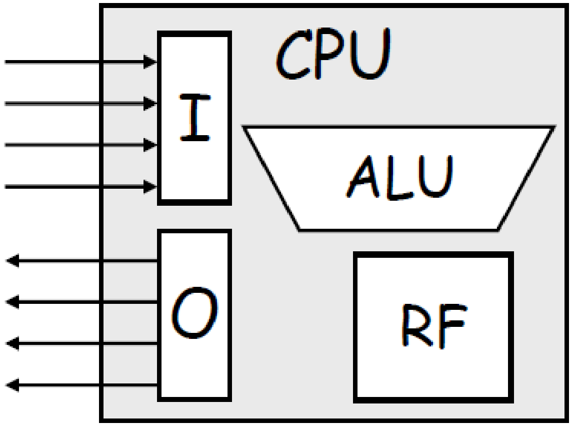
- I/O Controller
  - Input/Output handshaking
    - Interrupt
    - Polling

Input/Output (cont.)

- I/O Mapped I/O
- Memory-Mapped I/O

Input/Output (cont.)

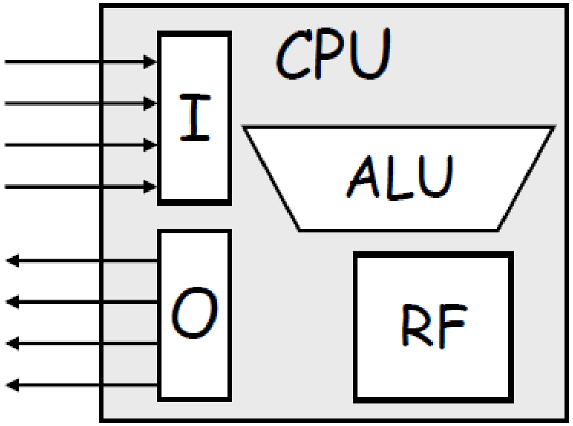
- I/O Mapped I/O
  - Dedicated I/O instructions
    - Part of ISA
  - Output
    - Values written to an output register
    - On output pins of an external port
  - Input
    - Values read from an input register
    - At input pins of an external port



- Example: Intel microprocessors: IN / OUT

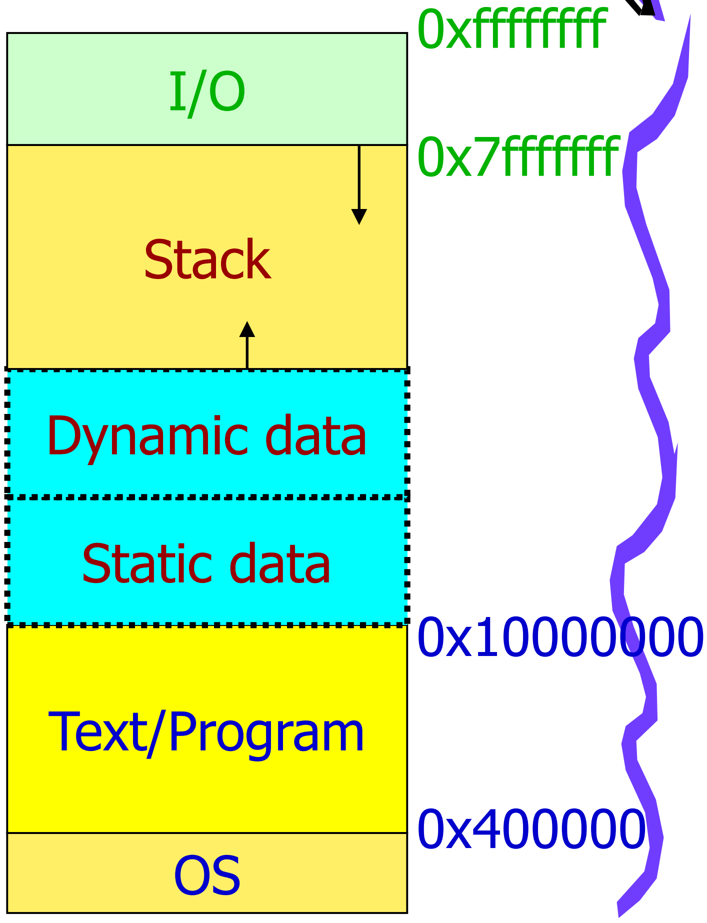
I/O Mapped I/O

- Pros
  - Low latency
    - Simpler decoding
- Cons
  - Not implemented by all CPUs
  - Limited addressing
    - Predetermined # of I/O signals



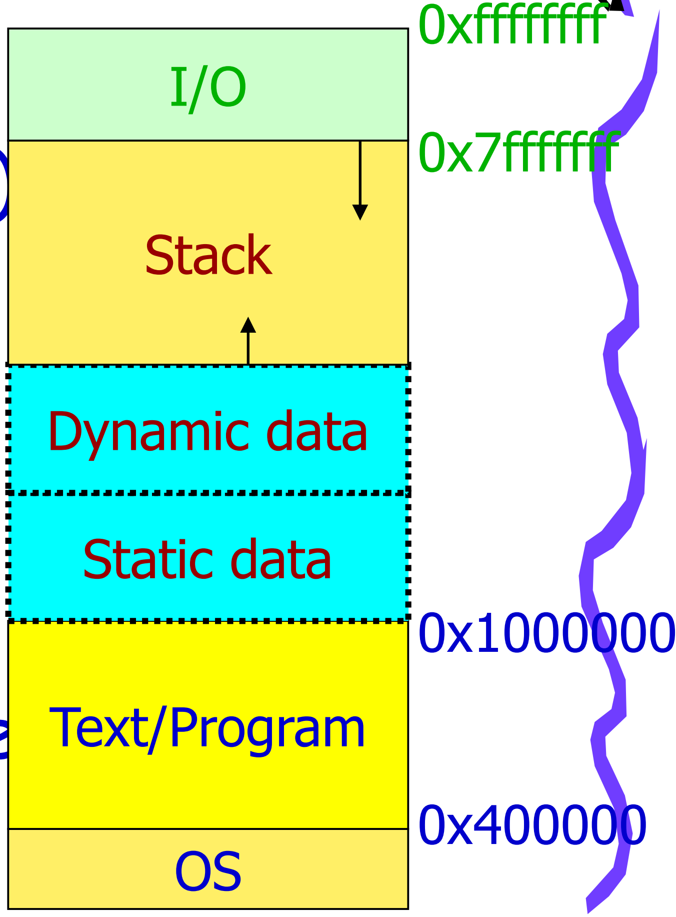
Input/Output (cont.)

- Memory-Mapped I/O
  - Performed by simple load/store instructions
  - A subset of unused memory addresses mapped to registers of external devices
  - Memory & devices on bus programmed to respond only to their own address ranges



Memory-Mapped I/O

- Pros
  - Easy to handle memory locations (SW perspective)
  - Wide range of addresses
- Cons
  - Caching may cause to memory incoherency issue
  - Slow & consumes CPU cycle
    - → Use DMA



0xffffffff

I/O

0x7fffffff

Stack

Dynamic data

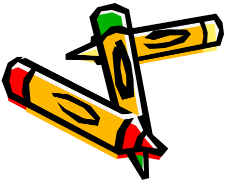
Static data

0x1000000

Text/Program

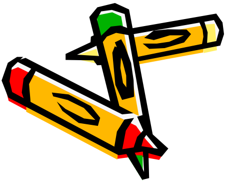
0x400000

OS



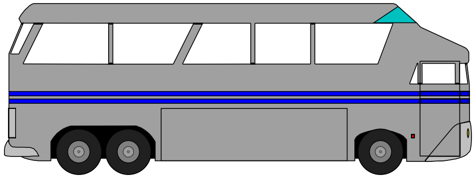
Input/Output (cont.)

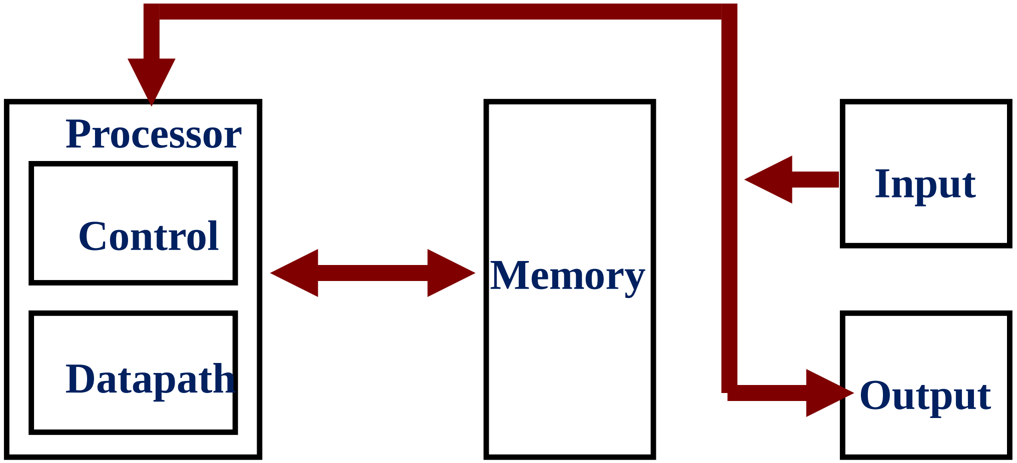
- Standard I/O Protocols
  - PCIe
  - Serial ATA (SATA)
  - Serial Attached SCSI (SAS)
  - USB 2.0
  - Parallel ATA (PATA)



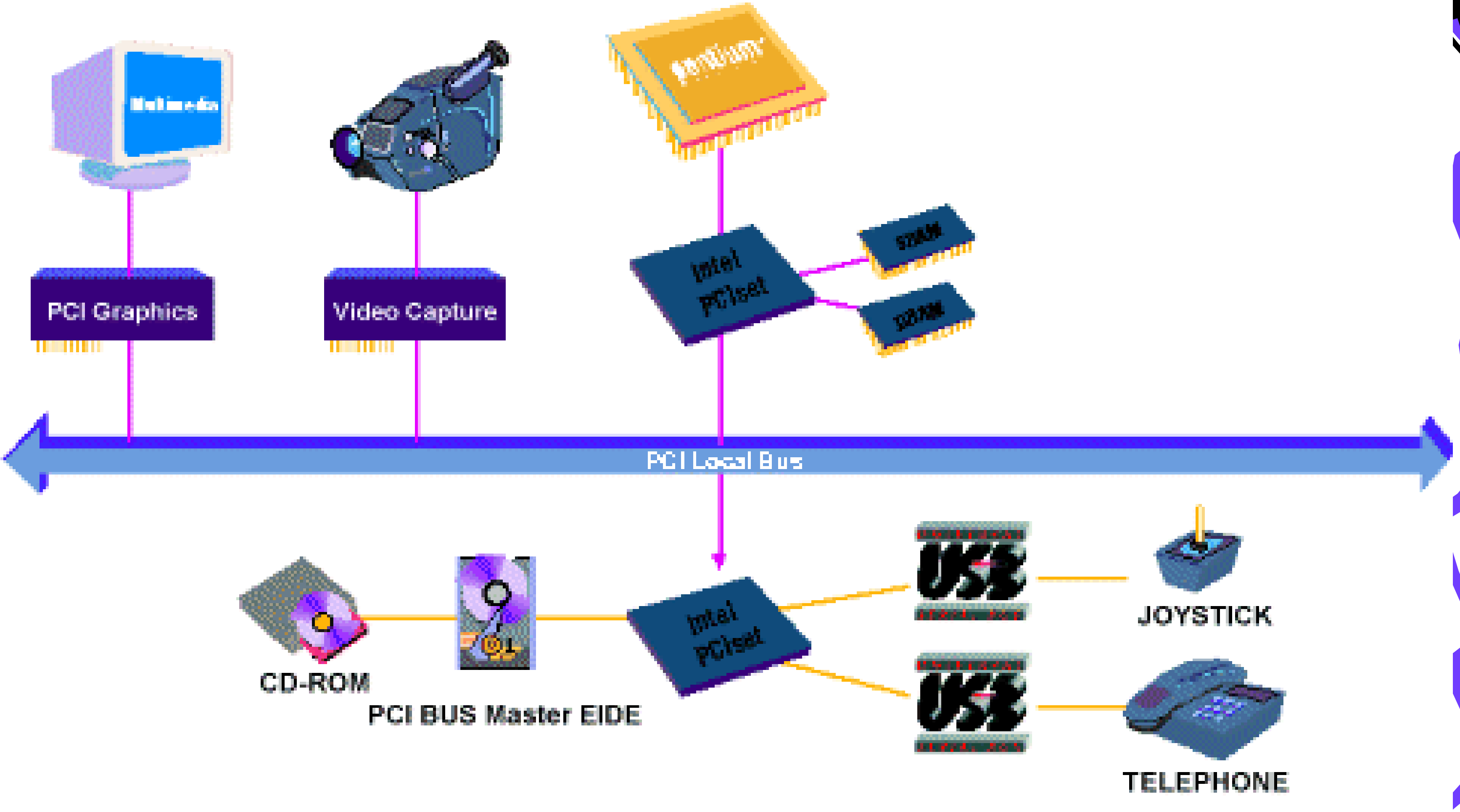
What is a Bus?

- Bus
  - Shared communication link
  - Single set of wires used to connect multiple subsystems

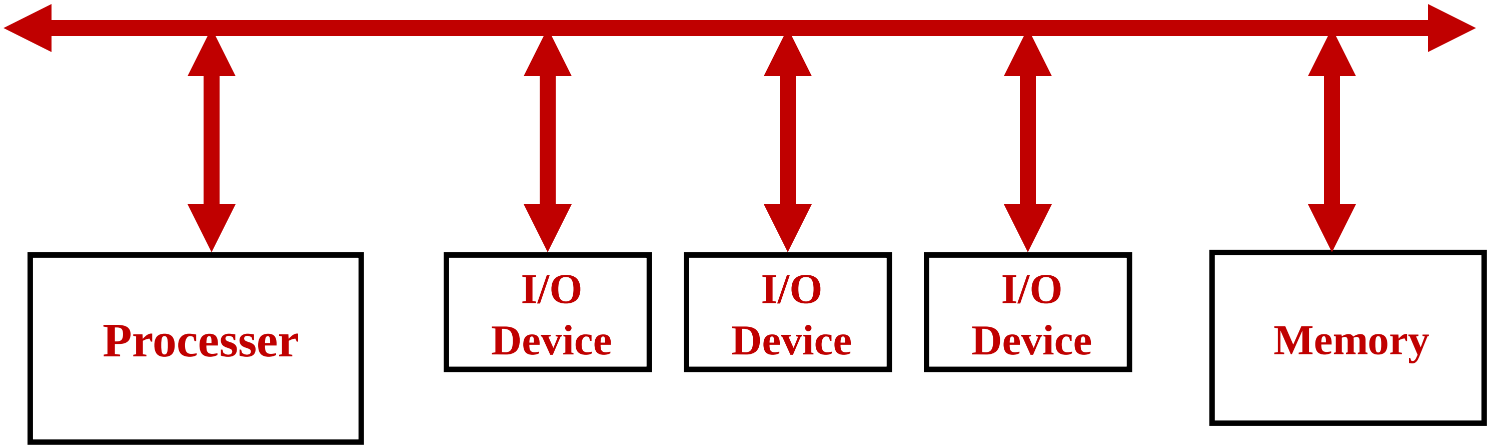




Buses


Multimedia
Pentium
intel PCIset
SRAM
DRAM
PCI Graphics
Video Capture
PCI Local Bus
intel PCIset
USB
USB
JOYSTICK
TELEPHONE
CD-ROM
PCI BUS Master EIDE

Advantages of Buses



Processer

I/O Device

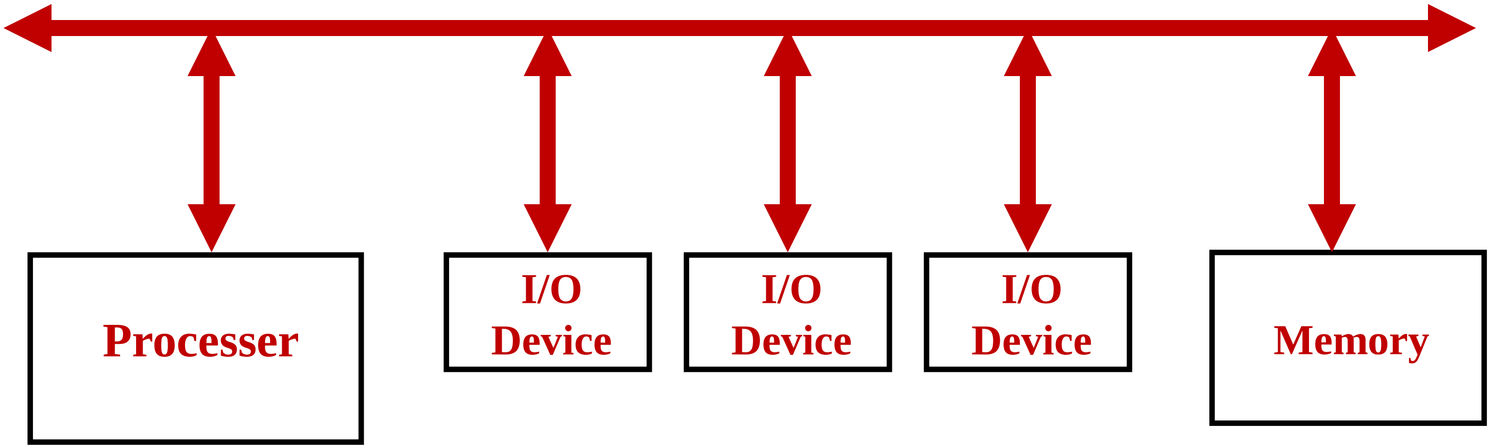
I/O Device

I/O Device

Memory

- Versatility
  - New devices can be added easily
  - Peripherals can be moved between computer systems that use same bus standard
- Low Cost
  - A single set of wires shared in multiple ways

Disadvantage of Buses



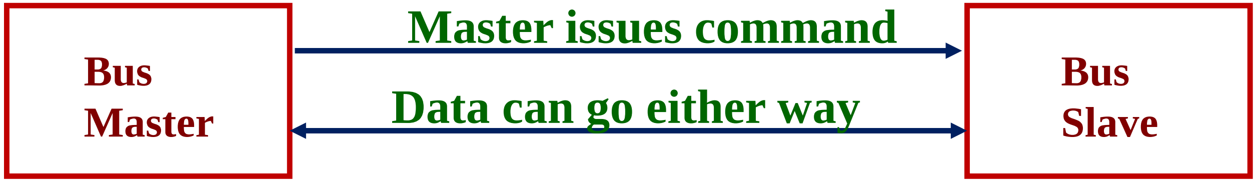
- Creates a communication bottleneck
  - Bus bandwidth limits maximum I/O throughput
- Maximum bus speed largely limited by:
  - Length of bus
  - Number of devices on bus
  - Slowest device on bus

General Organization of Buses

Control Lines
Data Lines

- Control Lines
  - Signal requests and acknowledgments
  - Indicate what type of information is on data lines
- Data Lines
  - Carry information between source and destination
  - Data and Addresses
  - Complex commands

Master versus Slave



Bus Master

Master issues command

Data can go either way

Bus Slave

- A Bus Transaction Includes Two Parts:
  - Issuing command (and address) – request
  - Transferring data – action
- Master: one who starts bus transaction by:
  - Issuing command (and address)
- Slave: one who responds to address by:
  - Sending data to master (M) if M asks for data
  - Receiving data from M if M wants to send data

Types of Buses

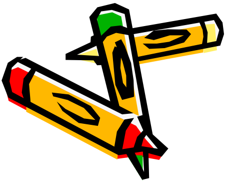
- Processor-Memory Bus
- I/O Bus
- Backplane Bus

Types of Buses

- Processor-Memory Bus
  - Design specific
  - Short and high speed
  - Only need to match memory system
    - Maximize memory-to-processor bandwidth
  - Connects directly to processor
  - Optimized for cache block transfers

Types of Buses

- I/O Bus
  - Industry standard
  - Usually is lengthy and slower
  - Need to match a wide range of I/O devices
  - Connects to processor-memory bus or backplane bus

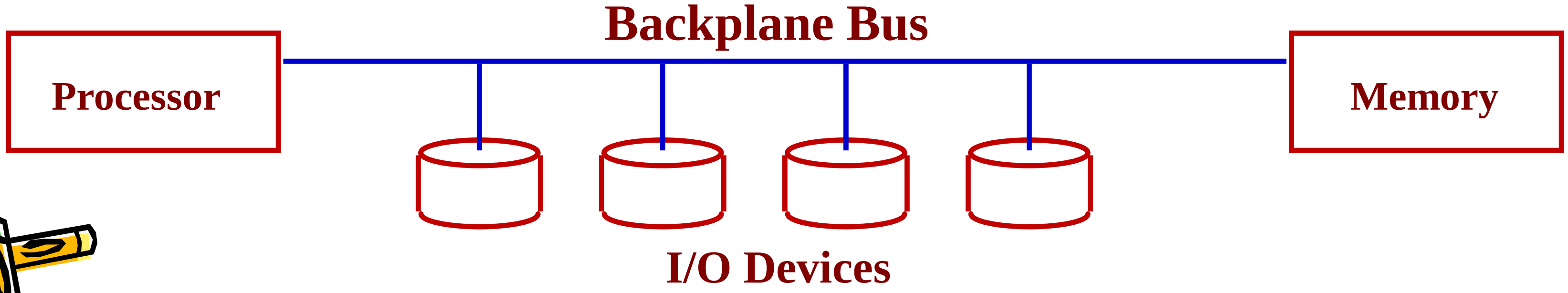


Types of Buses

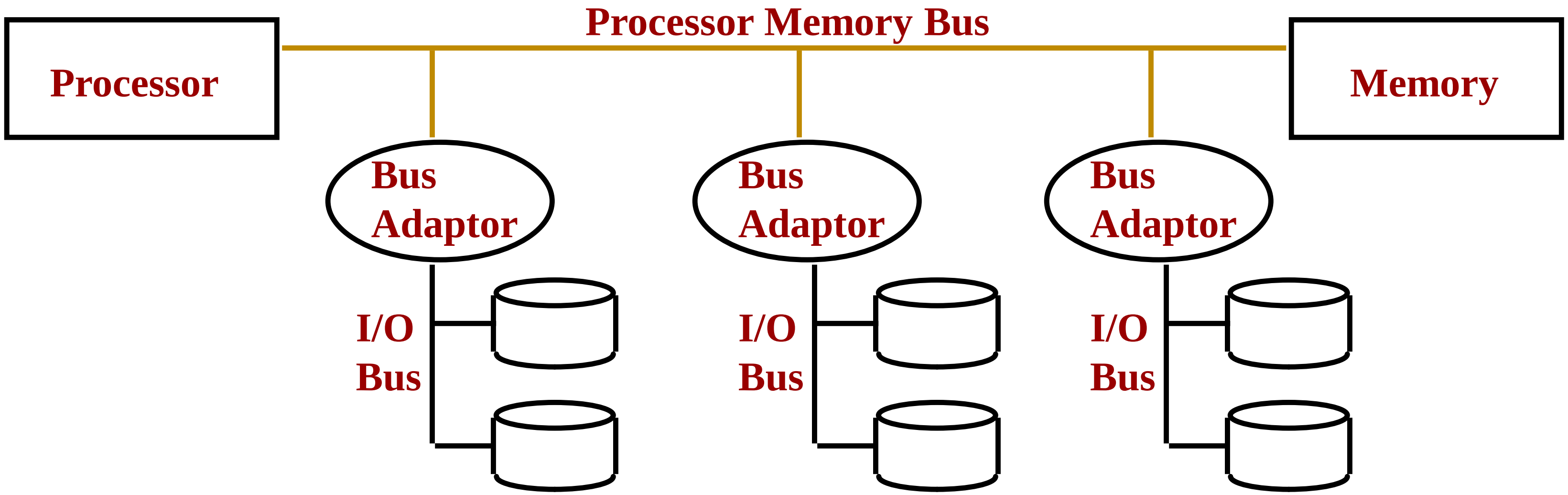
- Backplane Bus
  - Standard or proprietary
  - A single bus used for:
    - Processor to memory communication
    - Communication between I/O devices and memory
  - Backplane is an interconnection structure within chassis
  - Allow processors, memory, and I/O devices to coexist

Types of Buses

- Backplane Bus
  - Advantages: Simple and low cost
  - Disadvantages: slow and bus can become a major bottleneck
  - Example: IBM PC - AT



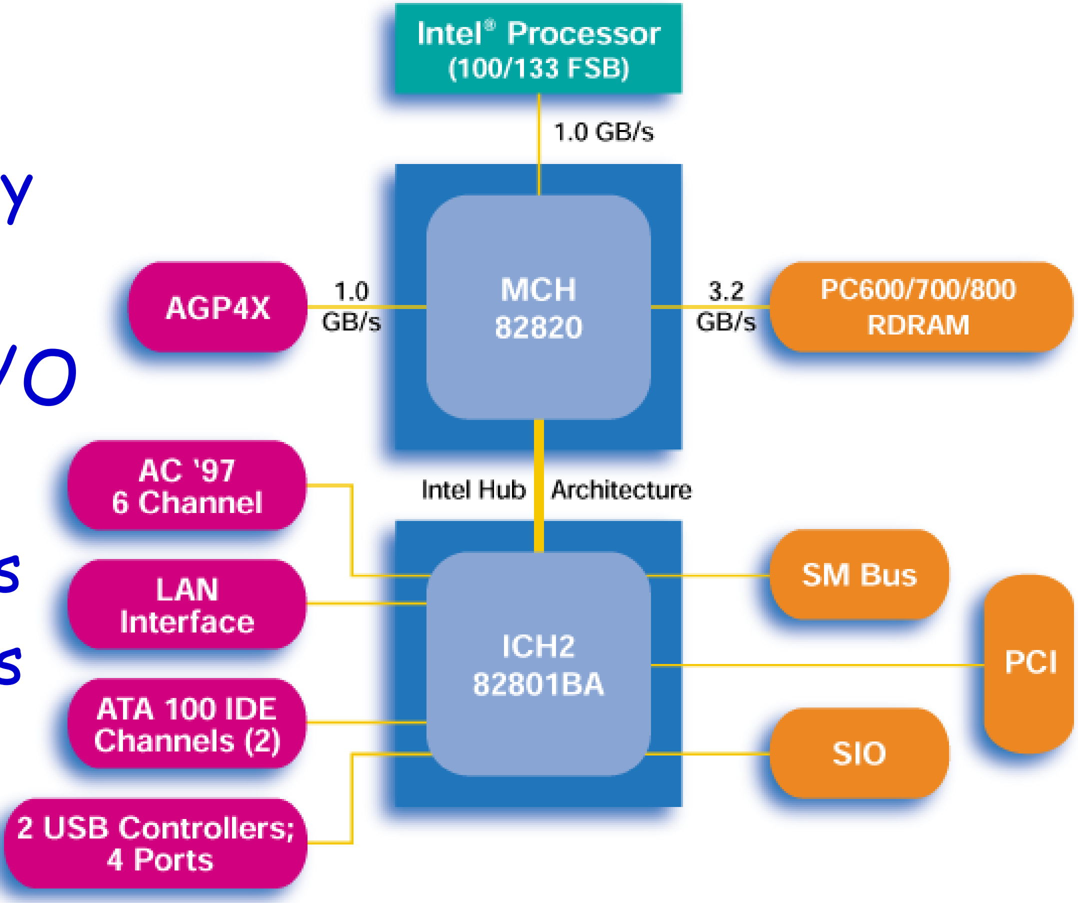
Two-Bus System



- I/O buses tap into processor-memory bus via bus adaptors:
- I/O buses provide expansion slots for I/O devices
- Example: Apple Macintosh-II

Main components of Intel Chipset: Pentium II/III

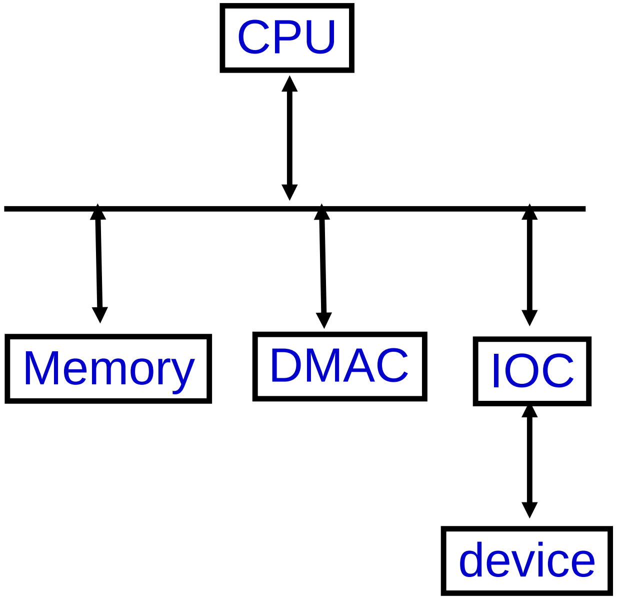
- Northbridge:
  - Handles memory
  - Graphics
- Southbridge: I/O
  - PCI bus
  - Disk controllers
  - USB controllers
  - Audio
  - Serial I/O
  - Interrupt controller
  - Timers



Delegating I/O Responsibility from CPU: DMA

- Direct Memory Access (DMA):
  - External to CPU
  - Act as a maser on bus
  - Transfer blocks of data to or from memory without CPU intervention

CPU sends a starting address, direction, and length count to DMAC. Then issues "start".



CPU

Memory

DMAC

IOC

device

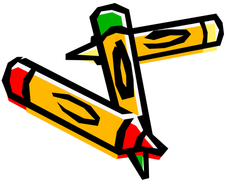
DMAC provides handshake signals for Peripheral Controller, and Memory Addresses and handshake signals for Memory.

Bus Transactions

- Transaction Phases
  - Master requests ownership from arbiter
  - Arbiter grants ownership to master
  - Master drives address for all to see
  - A slave claims transaction
  - Master (or slave) drives data (depending on read or write) for all to see
  - Master terminates transaction and bus ownership

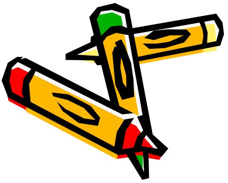
Basic Bus Signals

- CLK
  - All devices synchronized by a clock signal
- Private Signals to/from Arbiter per Master
  - **REQ** (output): assert to request ownership; de-assert to signal end of transaction
  - **GNT** (input): ownership is granted



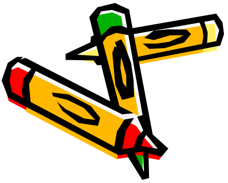
Basic Bus Signals

- “Broadcast” signals shared by all devices
  - **AD** (address/data bus, bi-directional): master drives address during the address phase, master/slave drives data during the data phase
  - Why not have separate address bus and data bus?
- R/W (bi-directional):
  - Bus commands, e.g. read vs. write



Synchronous Bus

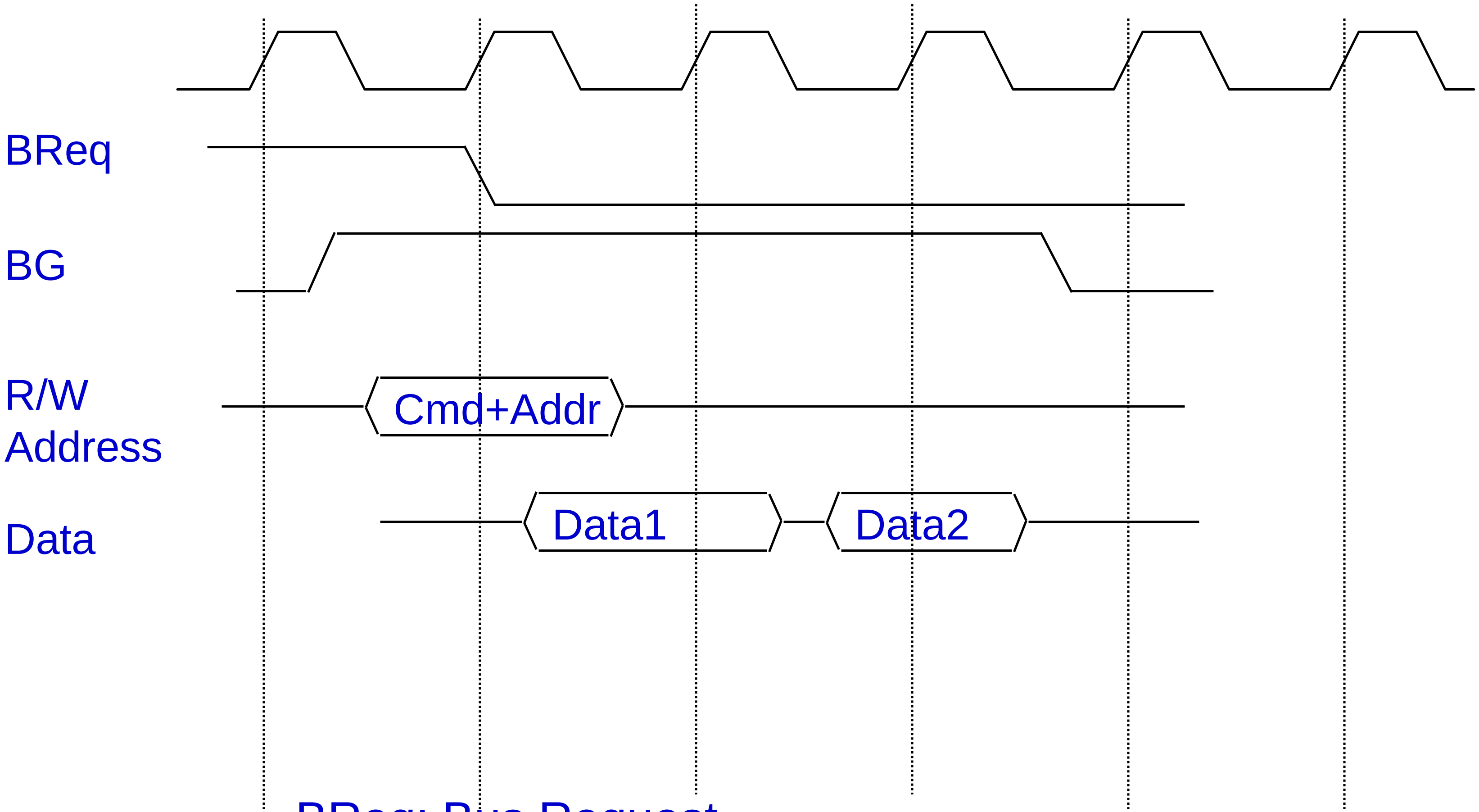
- Synchronous Bus:
  - Includes a clock in control lines
  - A fixed protocol relative to the clock
  - Advantage: little logic and very fast
  - Disadvantages:
    - Every device on the bus must run at the same clock rate
    - To avoid clock skew, they cannot be long if they are fast



Asynchronous Bus

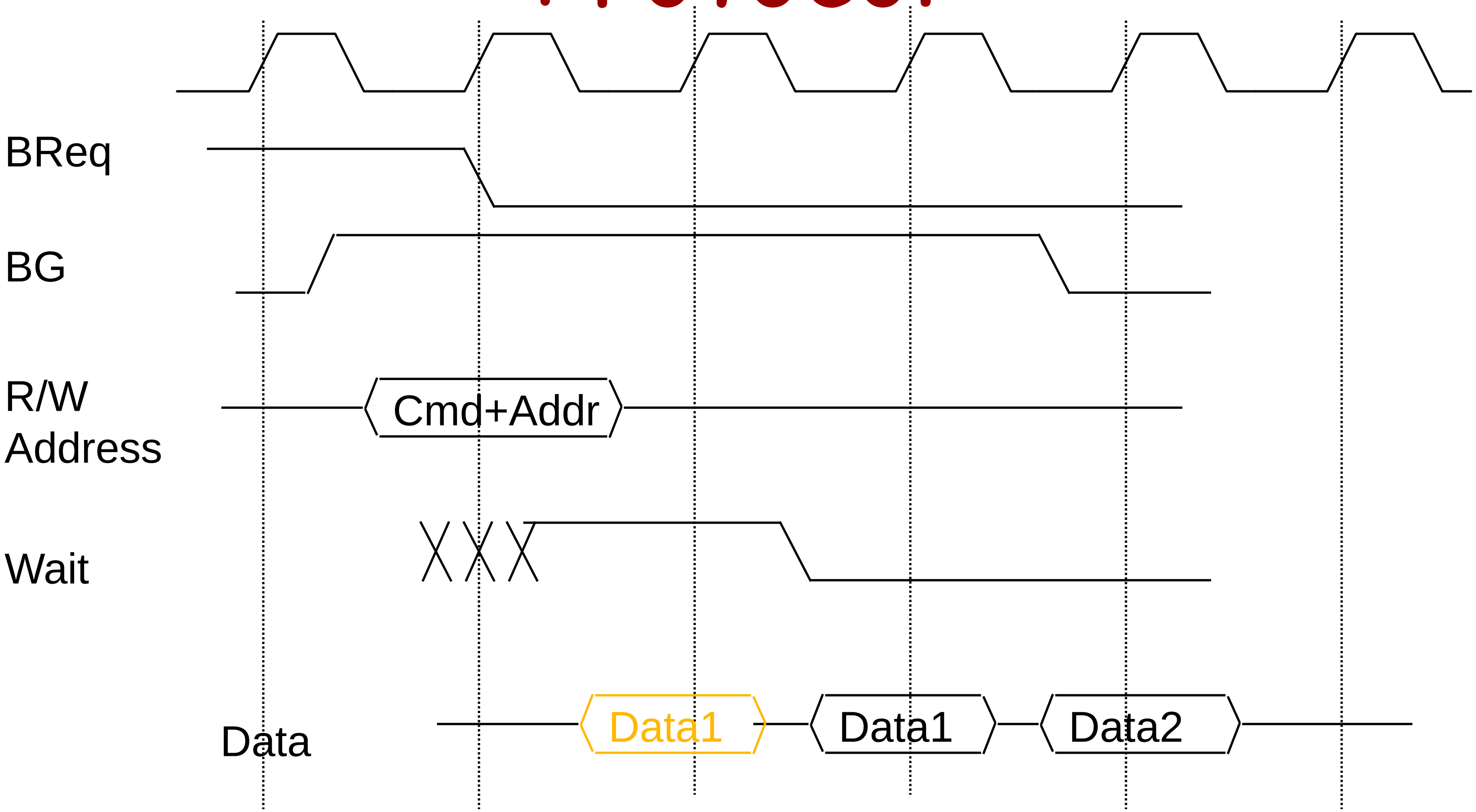
- Asynchronous Bus:
  - It is not clocked
  - It can accommodate a wide range of devices
  - It can be lengthened without worrying about clock skew
  - It requires a handshaking protocol

Simple Synchronous Protocol



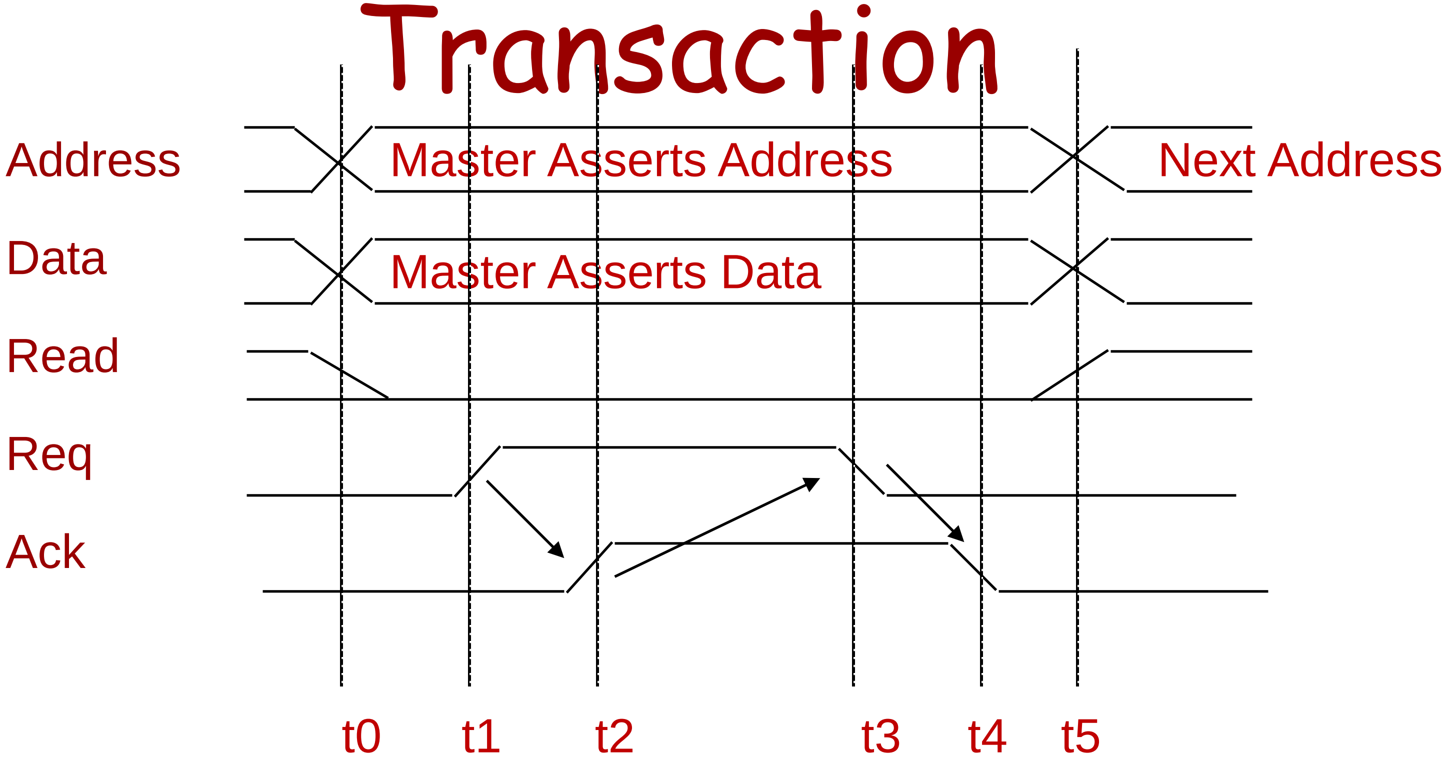
BReq: Bus Request
BG: Bus Grant

Typical Synchronous Protocol



- Slave indicates when it is prepared for data xfer
- Actual transfer goes at bus rate

Asynchronous Write Transaction



t0 : Master has obtained control and asserts address, direction, data
Waits a specified amount of time for slaves to decode target.

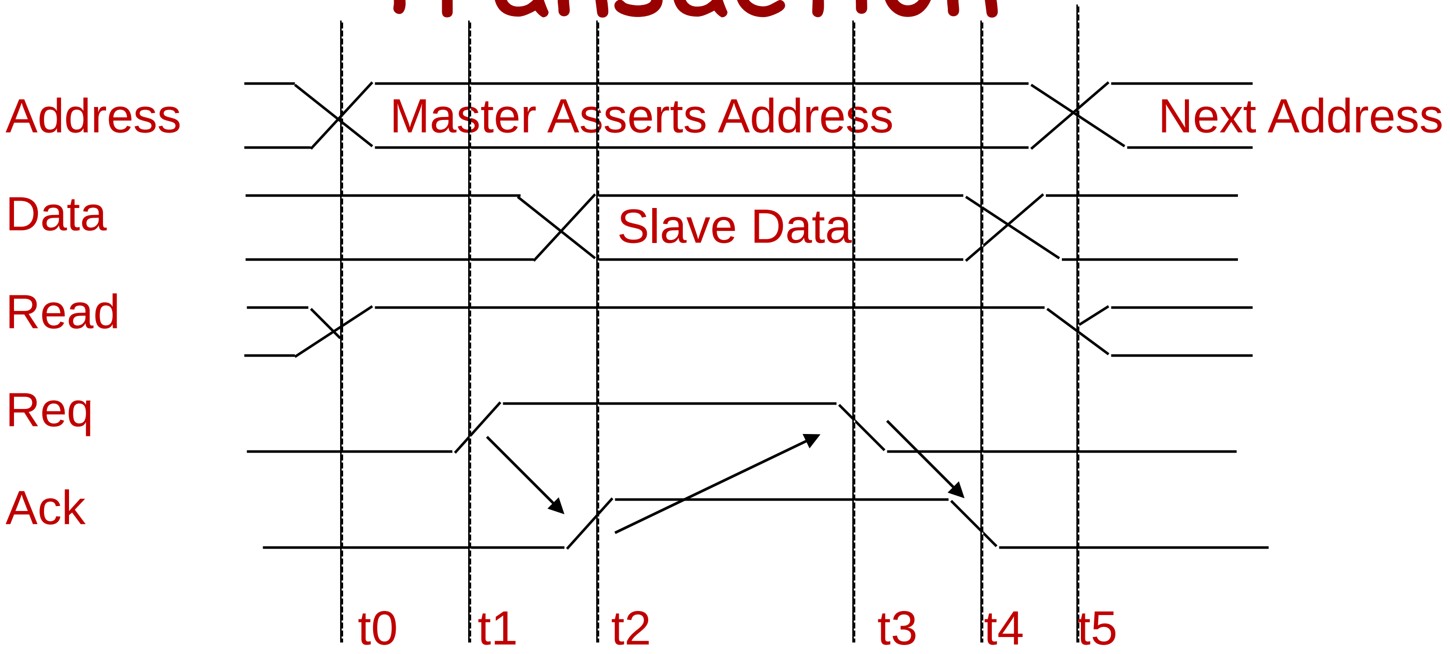
t1: Master asserts request line

t2: Slave asserts ack, indicating data received

t3: Master releases req

t4: Slave releases ack

Asynchronous Read Transaction



t0 : Master has obtained control and asserts address, direction, data

Waits a specified amount of time for slaves to decode target.

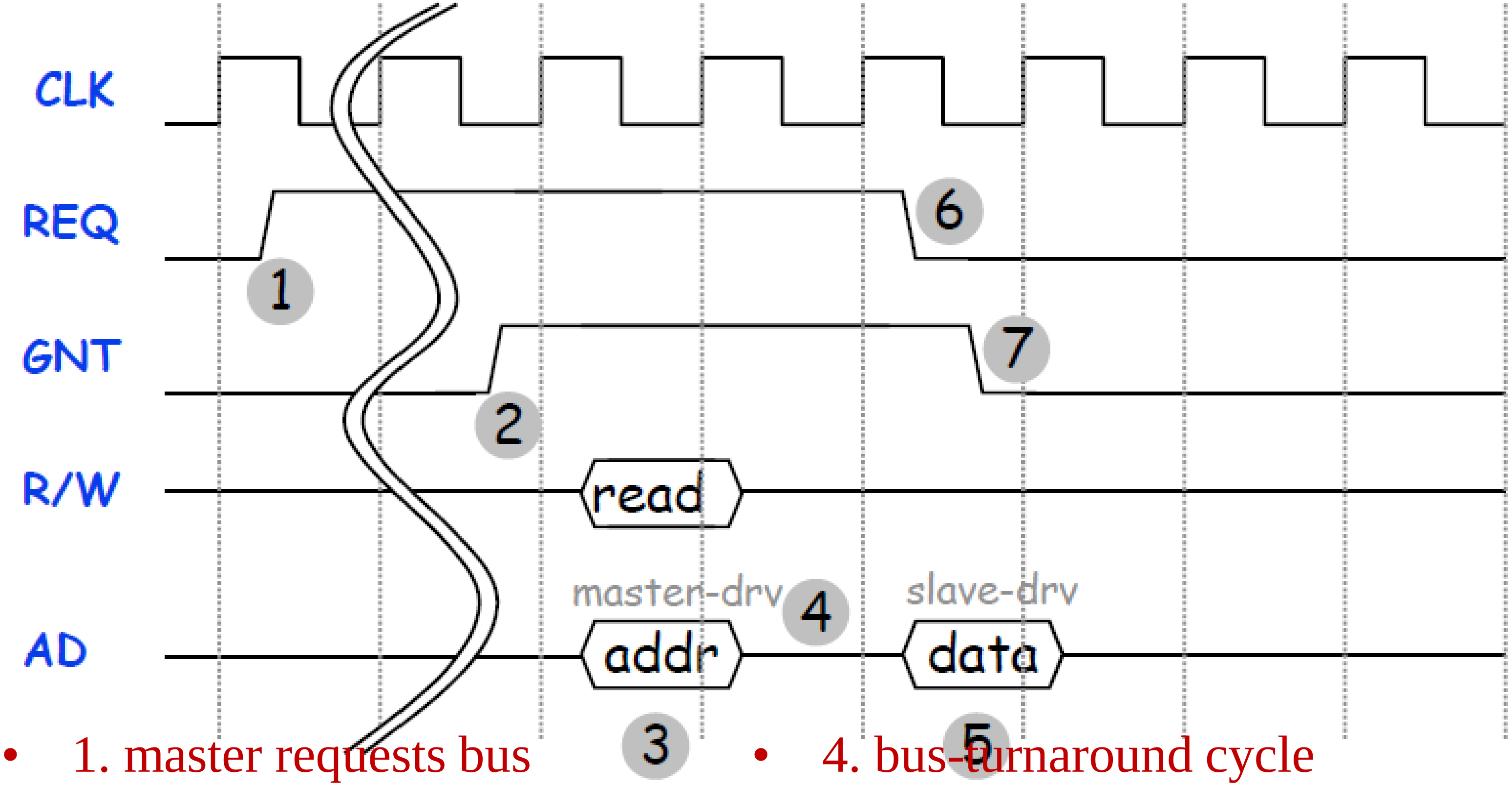
t1: Master asserts request line

t2: Slave asserts ack, indicating ready to transmit data

t3: Master releases req, data received

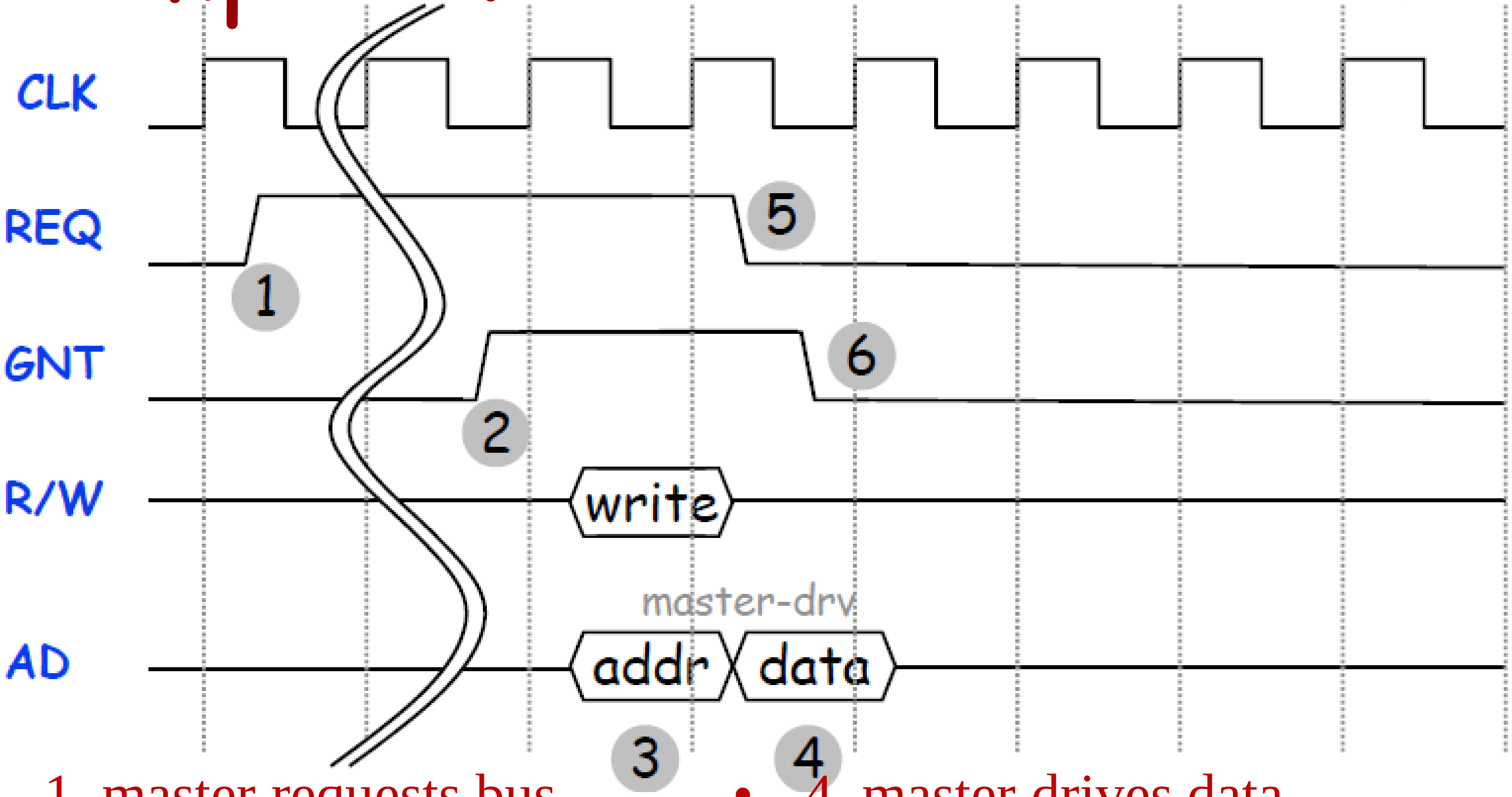
t4: Slave releases ack

Simple Read Transaction



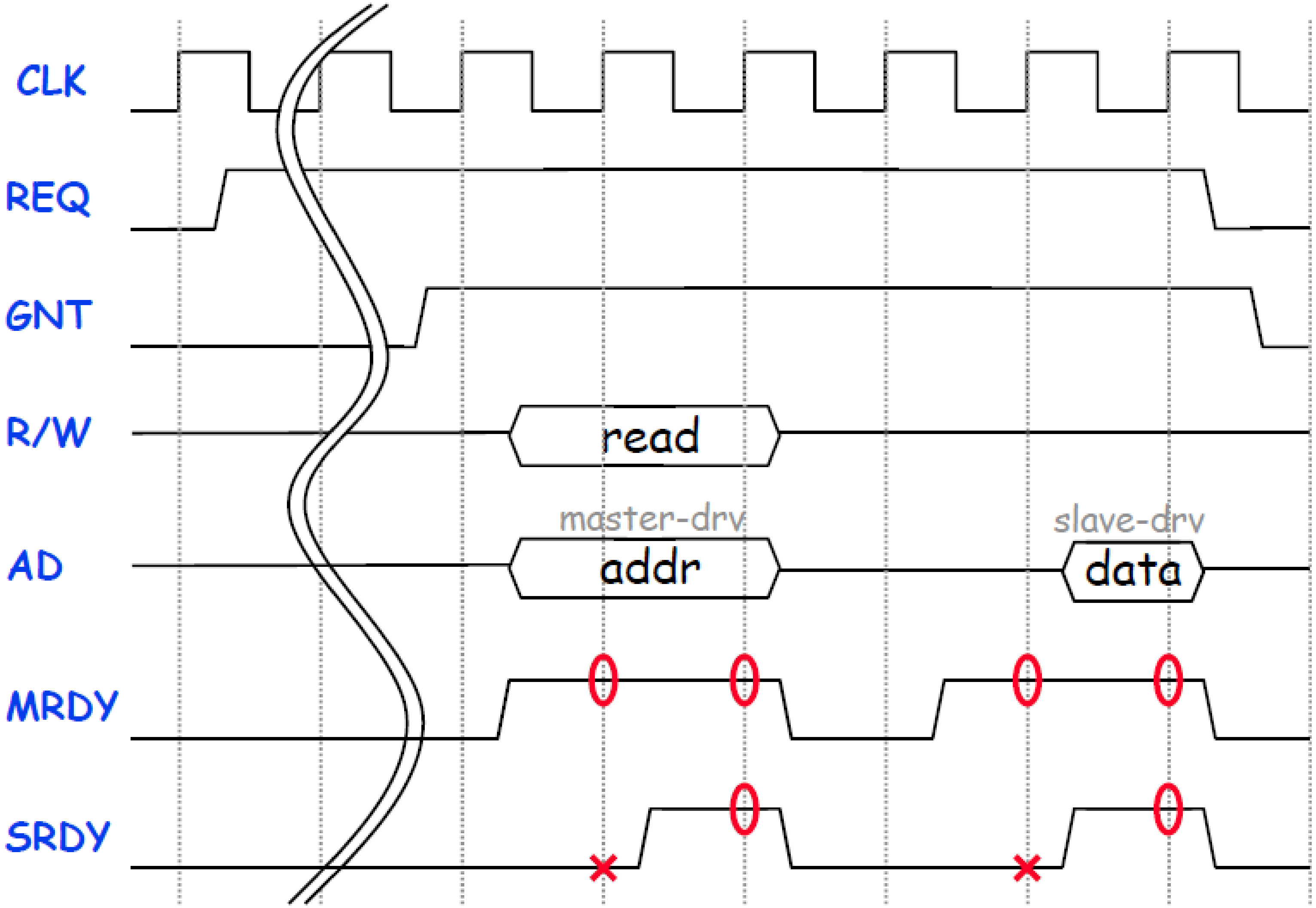
- 1. master requests bus
- 2. arbiter grants bus
- 3. master drives address /command
- 4. bus-turnaround cycle
- 5. slave drives data
- 6. master signals final cycle
- 7. arbiter acknowledges

Simple Write Transaction

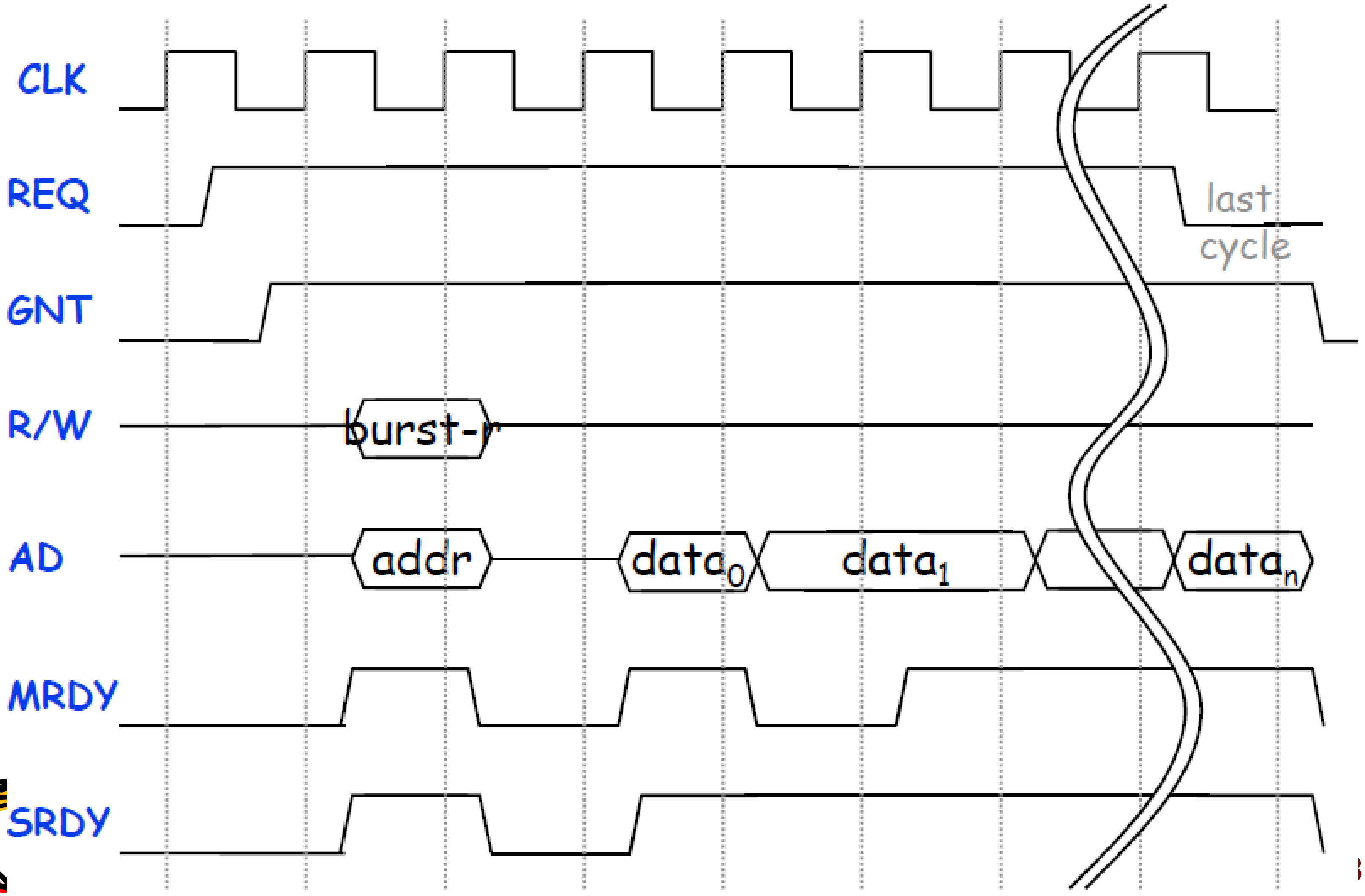


- 1. master requests bus
- 2. arbiter grants bus
- 3. master drives address /command
- 4. master drives data
- 5. master signals final cycle
- 6. arbiter acknowledges

Asynch Read Transaction

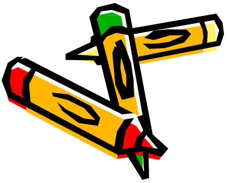


Burst Read Transaction



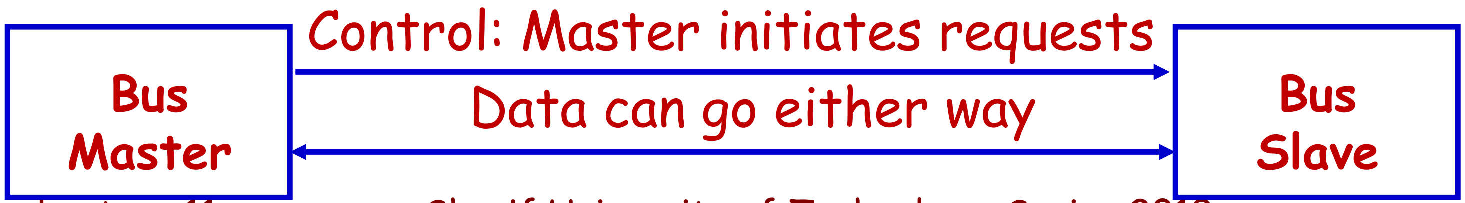
Bus Arbitration

- Arbitration scheme:
  - A bus master wanting to use the bus asserts the bus request
  - A bus master cannot use the bus until its request is granted
  - A bus master must signal to the arbiter after finish using the bus
- Arbitration schemes balance two factors:
  - Bus priority: the highest priority device should be serviced first
  - Fairness: Even the lowest priority device should never be completely locked out from bus

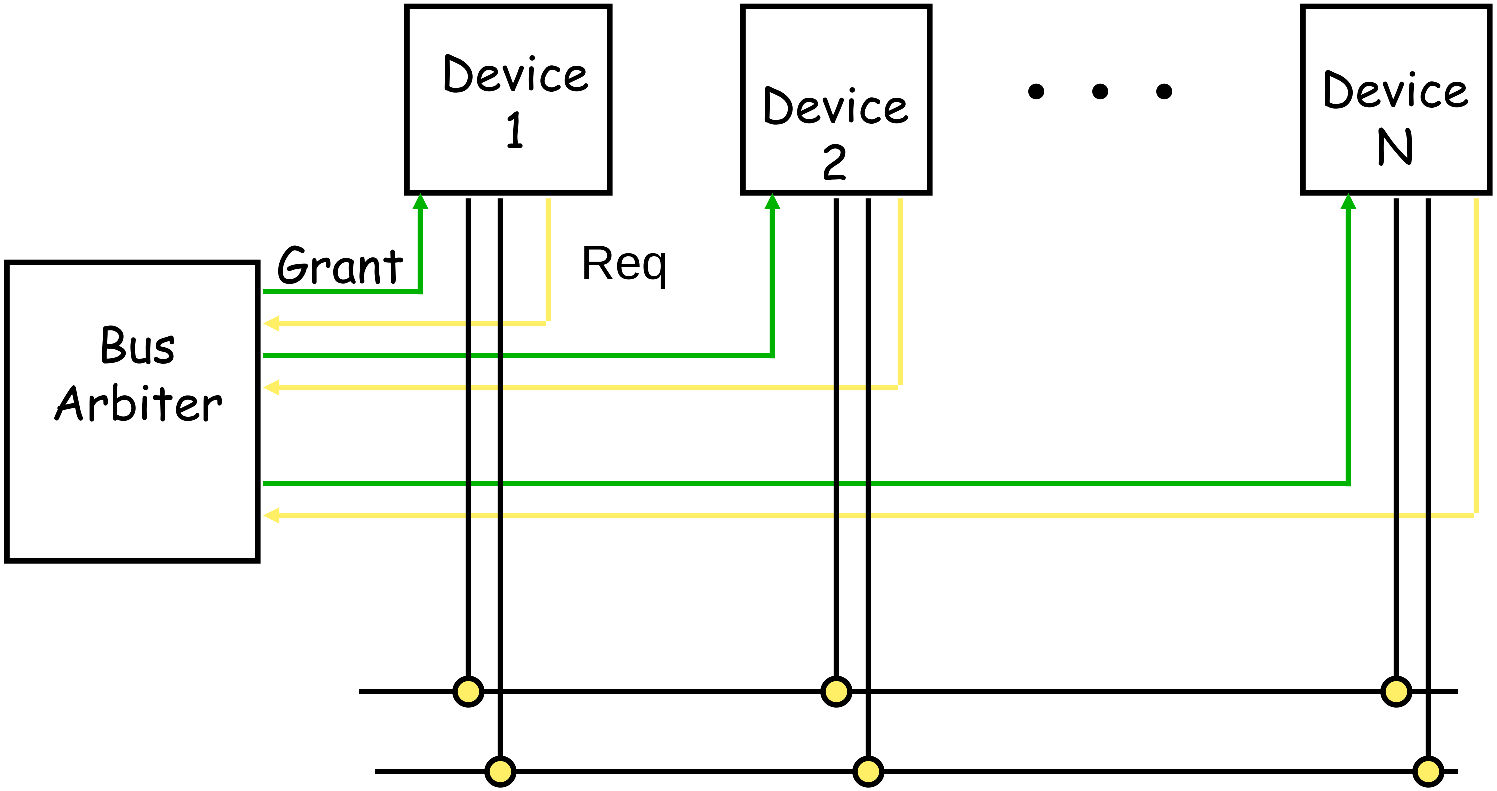


Simple Bus Arbitration

- Master-slave arrangement:
  - Only bus master can control access to bus
    - It initiates and controls all bus requests
  - Slave responds to read/write requests
  - Example:
    - Processor is only bus master
    - All bus requests controlled by processor
    - Major drawback ?



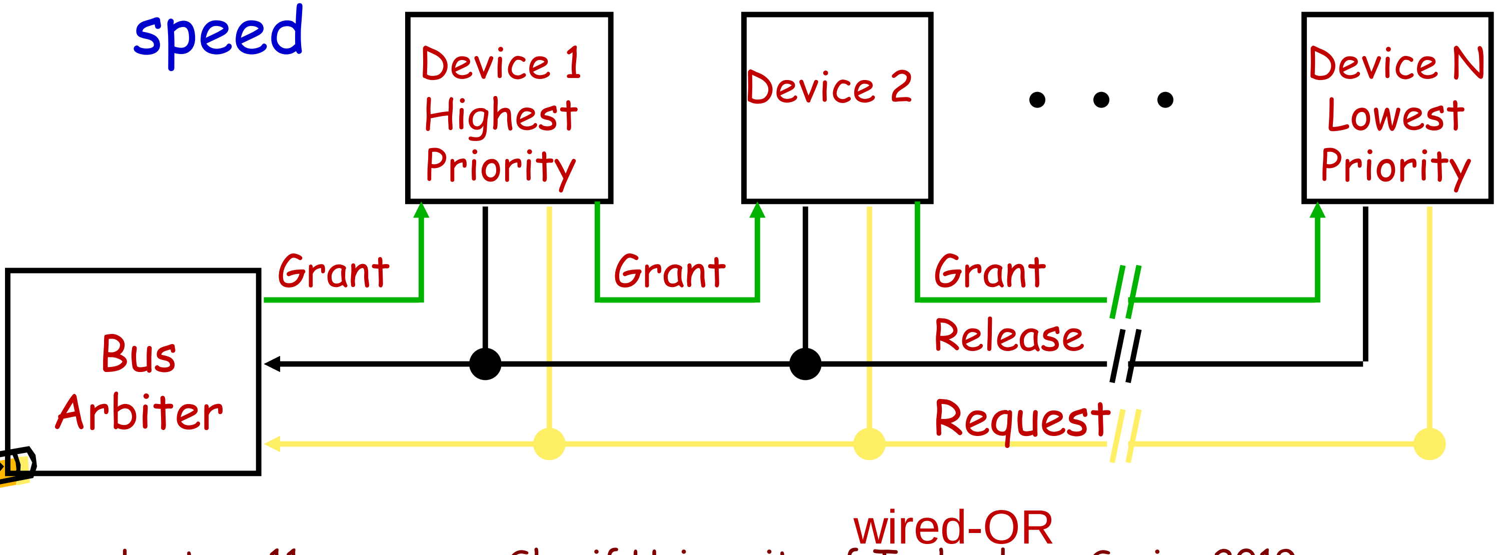
Centralized Parallel Arbitration



- Used in essentially all processor-memory busses and in high-speed I/O busses

Daisy Chain Bus Arbitrations Scheme

- Advantage: simple
- Disadvantages:
  - Cannot assure fairness:
    - A low-priority device may be locked out indefinitely
  - Use of daisy chain grant signal also limits bus speed



Increasing Bus Bandwidth

- Separate Address and Data Lines
- Data Bus Width
- Block Transfers

Increasing Bus Bandwidth

- Separate Address and Data Lines
  - vs. multiplexed address/data lines
  - Address and data can be transmitted in one bus cycle if separate address and data lines are available
  - Cost:
    - More bus lines
    - Increased complexity

Increasing Bus Bandwidth

- Data Bus Width
  - By increasing width of data bus, transfers of multiple words require fewer bus cycles
  - Example: SPARCstation 20's memory bus is 128 bit wide
  - Cost: more bus lines

Increasing Bus Bandwidth

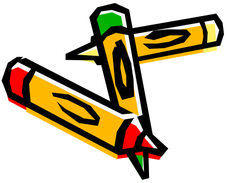
- Block Transfers (Bursts Transfer)
  - Allow bus to transfer multiple words in back-to-back bus cycles
  - Only one address sent at beginning
  - Bus is not released until the last word is transferred
  - Cost:
    - Increased complexity
    - Decreased response time for request

I/O System Characteristics

- Dependability
  - Particularly for storage devices
- Performance Measures
  - Latency (response time)
  - Throughput (bandwidth)
  - Desktops & embedded systems
    - Mainly interested in response time & diversity of devices
  - Servers
    - Mainly interested in throughput & expandability of devices

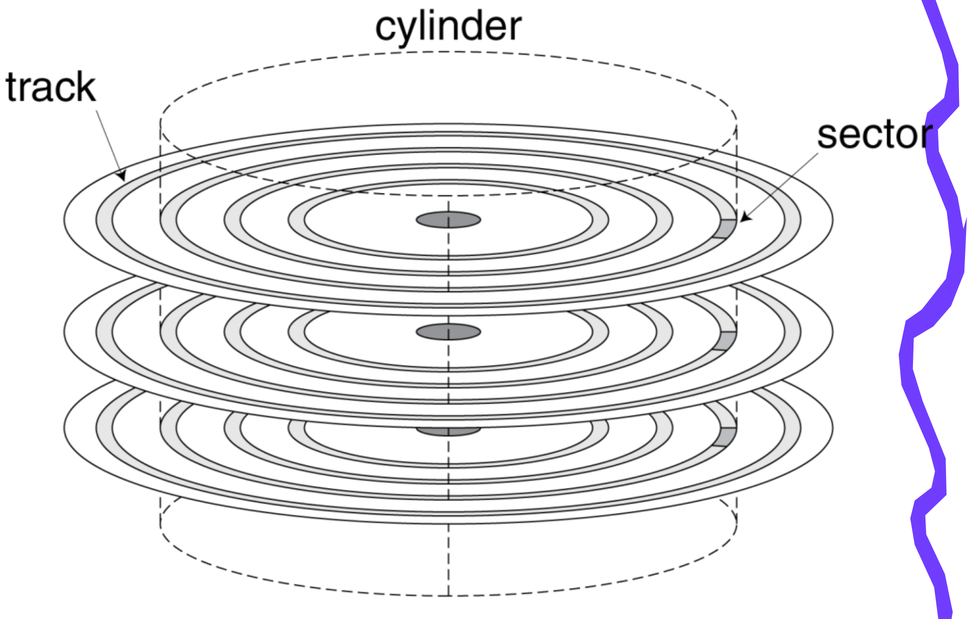
I/O Device Examples

Device	Behavior	Partner	Data Rate: KB/sec
Keyboard	Input	Human	0.01
Mouse	Input	Human	0.02
Line Printer	Output	Human	1.00
Floppy disk	Storage	Machine	50.00
Laser Printer	Output	Human	100.00
Optical Disk	Storage	Machine	500.00
Magnetic Disk	Storage	Machine	5,000.00
Network-LAN	Input/Output	Machine	20 – 1,000.00
Graphics Display	Output	Human	30,000.00



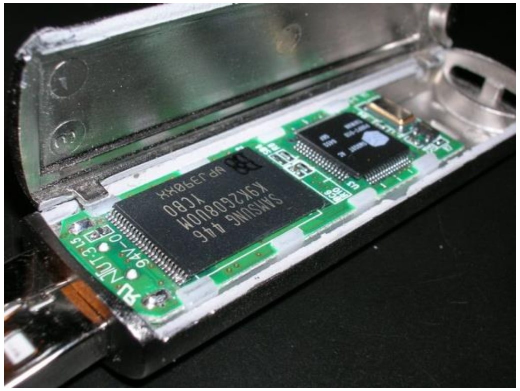
Disk Storage

- Nonvolatile, Rotating Magnetic Storage



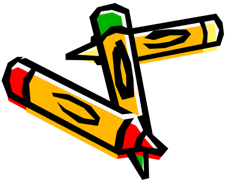
Solid-State Disks (SSDs)

- Non-Volatile Semiconductor Storage
  - 100× – 1000× faster than disk (??)
  - Smaller, lower power, more robust
  - But more $/GB (between disk and DRAM)

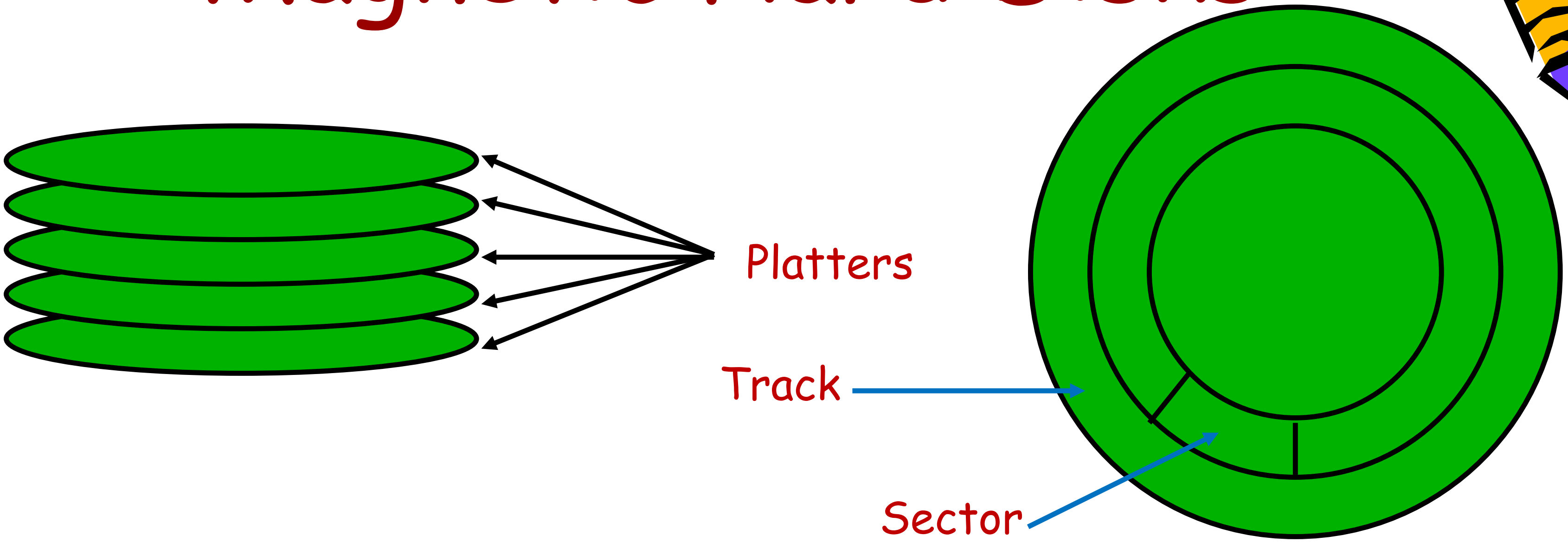


Flash Types

- NOR flash: bit cell like a NOR gate
  - Random read/write access
  - Used for instruction memory in embedded systems
- NAND flash: bit cell like a NAND gate
  - Denser (bits/area), but block-at-a-time access
  - Cheaper per GB
  - Used for USB keys, media storage, …
- Flash bits wears out after 1000's of accesses
  - Not suitable for direct RAM or disk replacement
  - Wear leveling: remap data to less used blocks

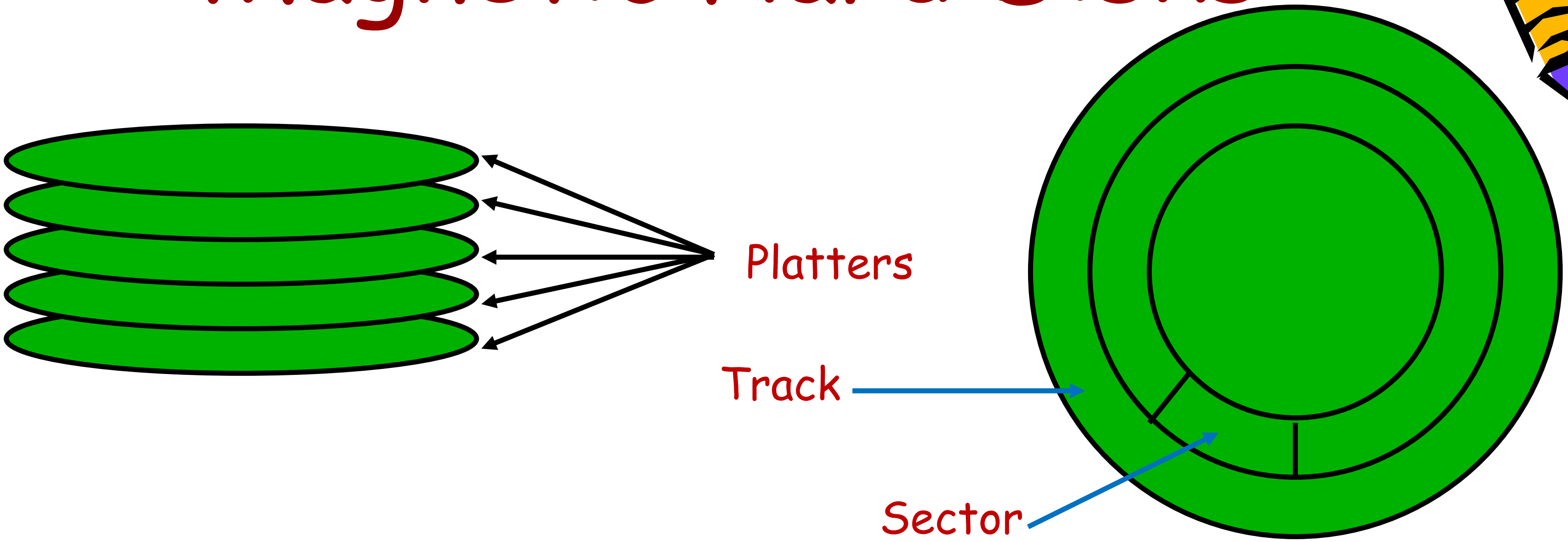


Magnetic Hard Disks



- Typical numbers
  - 500 to 2,000 tracks per surface
  - 32 to 128 sectors per track
    - *A sector is the smallest unit that can be read or written*

Magnetic Hard Disks



Platters

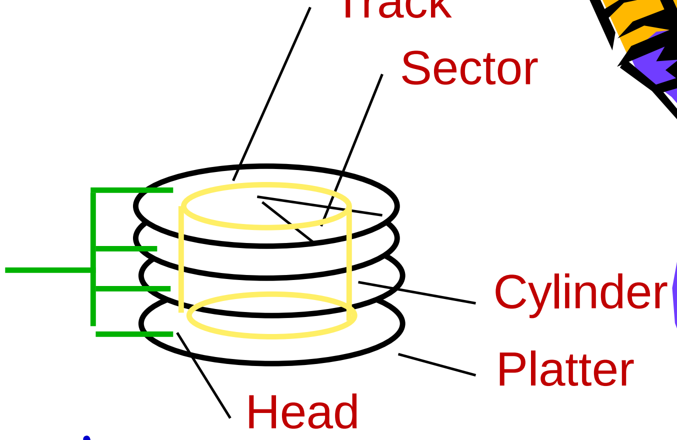
Track

Sector

- Tracks have same number of sectors
  - Constant bit density:
    - Record more sectors on outer tracks
  - Recently relaxed:
    - Constant bit size, speed varies with track location

Magnetic Hard Disks

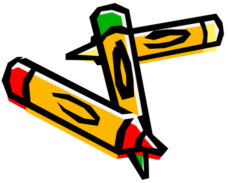
- Cylinder:
  - All tacks under head at a given point on all surface



- Read/write data is a three-stage process:
  - Seek time: position arm over proper track
  - Rotational latency: wait for desired sector to rotate under read/write head
  - Transfer time: transfer a block of bits (sector) under the read-write head
- Average seek time: 8 ms ~ 12 ms

Why I/O is Important?

- Processing Power
  - – doubles every 18 months
- Memory Size
  - – Doubles every 18 months (4X/3yr)
- Disk Capacity
  - – Doubles every 18 months
- Disk Positioning Rate (Seek + Rotate)
  - – Doubles every Ten years!



HDDs vs. SSDs

WD Black SATA HDD (4TB)

Slow (180 MB/s, IOPS: 140) ☹

Power Hungry (Active: 9w, Idle: 8w) ☹

Heavy (780g) ☹

Noisy ☹

MTTF < 1 Mh ☹

Inexpensive (0.05 $/G) ☺

$/IOPS (1.5 $) ☹

Unlimited Writes ☺

Samsung 863a SATA SSD (2TB)

Fast (550 MB/s, IOPS > 90K) ☺

Low Power (Active: 5w, Idle: 60mw) ☺

Light (80g) ☺

Very Low Noise ☺

MTTF: 2 Mh ☺

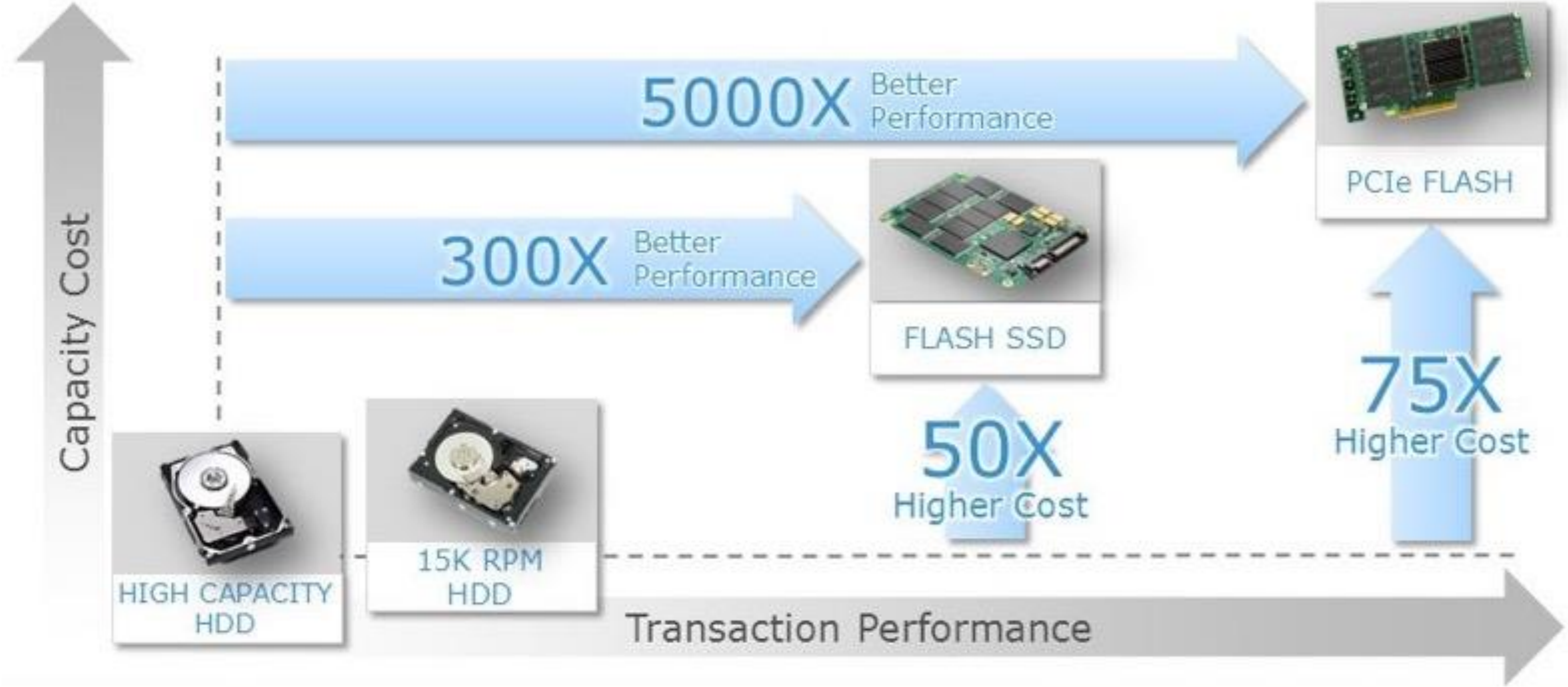
Expensive (0.5 $/G) ☹

$/IOPS (0.1) ☺

Limited Writes ☹

http://www.storagereview.com/, Accessed: Oct. 2015

Hybrid Storage



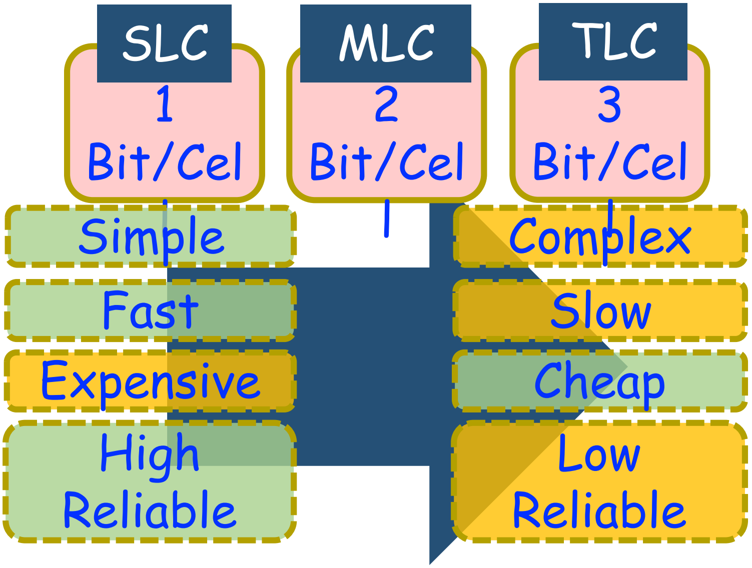
EMC Corp., 2013

SSD → High Performance
HDD → Low Cost + Large Capacity

SSD + HDD → Feasible Cost and Performance

SSDs: Quick Introduction

- Flash-Based Solid State Drives
  - Non-volatile
  - Consist of NAND flashes



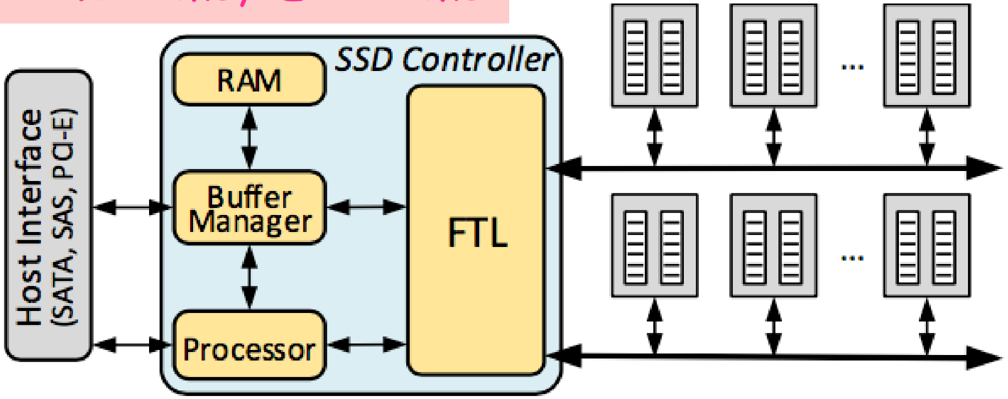
SLC
1 Bit/Cell

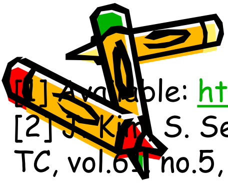
MLC
2 Bit/Cell

TLC
3 Bit/Cell

Simple | Complex
Fast | Slow
Expensive | Cheap
High Reliable | Low Reliable

✓SLC
P: 200us, E: 0.4~1ms
✓MLC
P: 0.3~2ms, E: 2~4ms

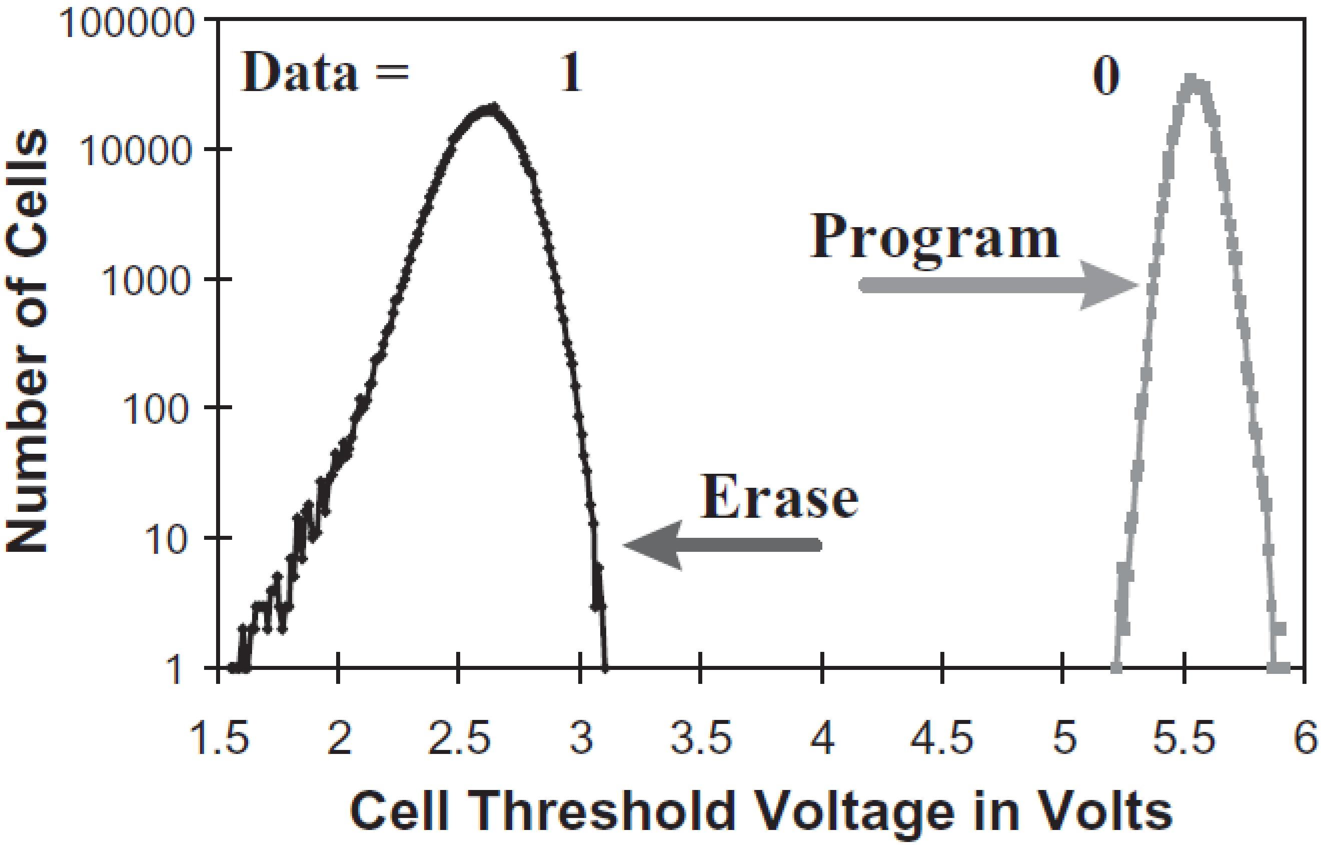




[1] Available: http://codecapsule.com/wp-content/uploads/2014/01/samsungssd840pro-02.jpg, Accessed: Sep. 2017.
[2] J. Kim, S. Seo, D. Jung, J.-S. Kim, and J. Huh, “Parameter-Aware I/O Management for Solid State Disks (SSDs),” IEEE TC, vol.61, no.5, pp.636–649, 2012.

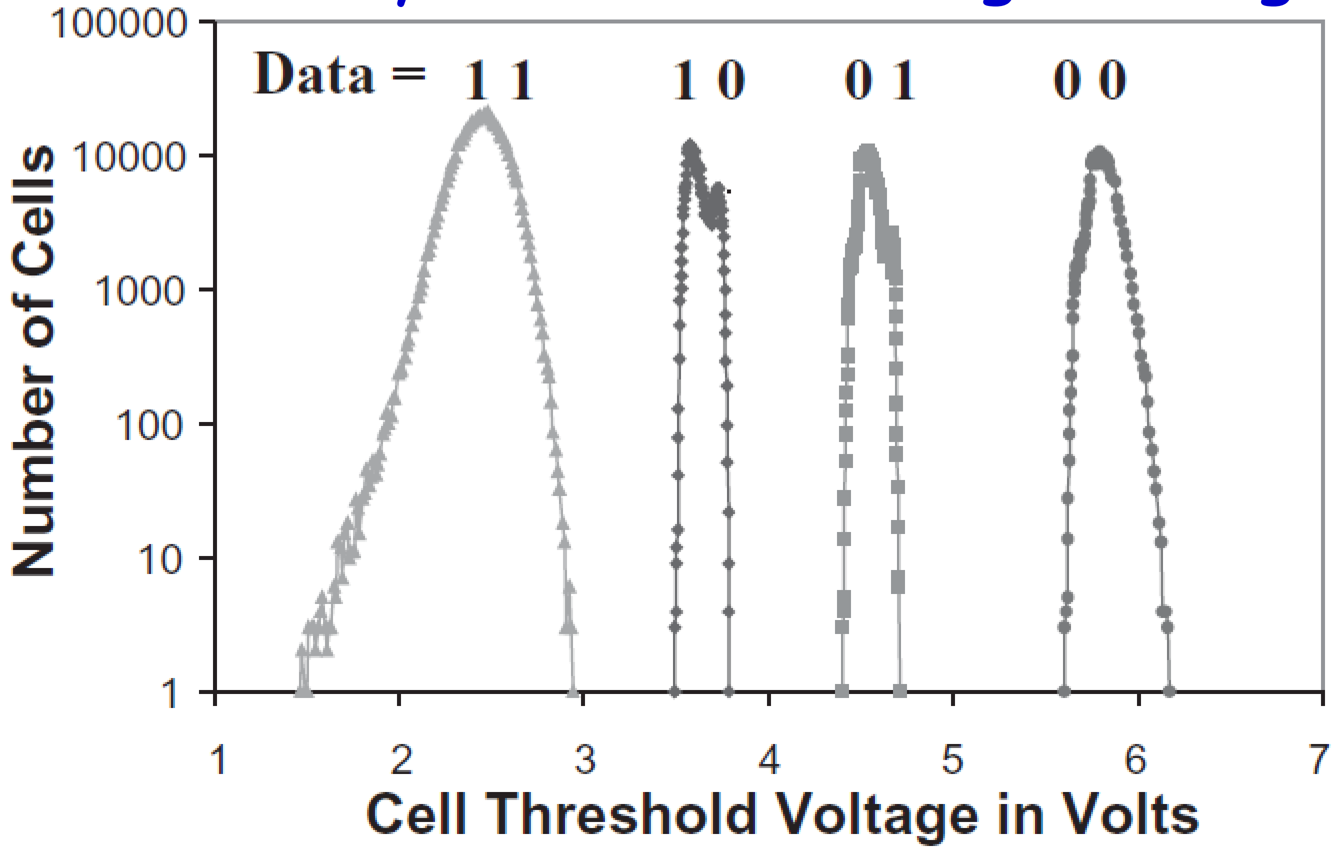
SLC Flash Technology

Single-bit/cell array threshold voltage histogram



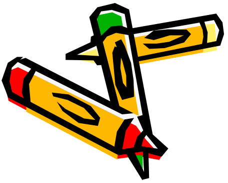
MLC Flash Technology

Two-bit/cell array threshold voltage histogram



Why NVMs?

- Computer Architecture Perspective
  - Limitations of DRAM
    - Power wall
    - Cost wall
    - Scaling wall
- Storage Perspective
  - Limitations of HDDs
  - Limitations of Flash-based SSDs



Emerging NVM Technologies

- Ferroelectric RAM (FRAM)
- Magnetic RAM (MRAM)
- Phase Change Memory (PCM)
- Spin Torque Transfer RAM (STT-RAM)
- Resistive Random Access Memory (ReRAM/RRAM)
- …

Backup