

# Computer Organization and Design

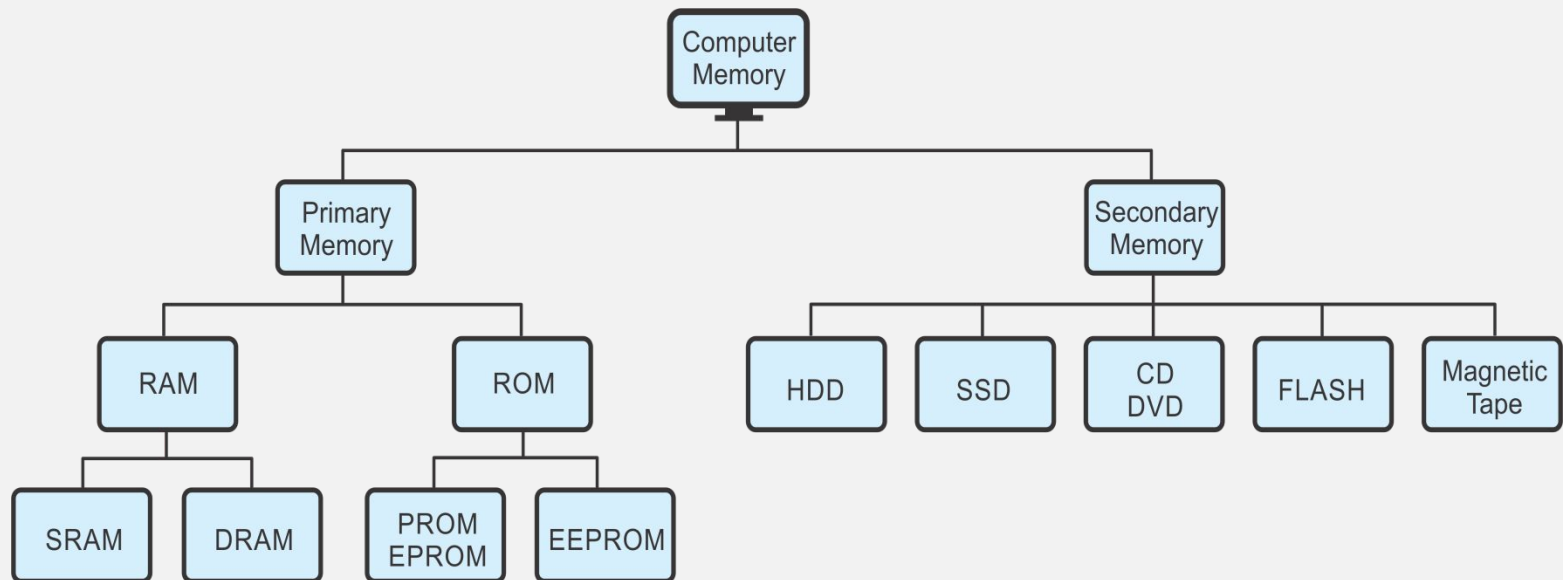
## Chapter 5

### Memory Hierarchy and I/O

#### Section 5.1 – 5.4, 5.8

# Computer Memory

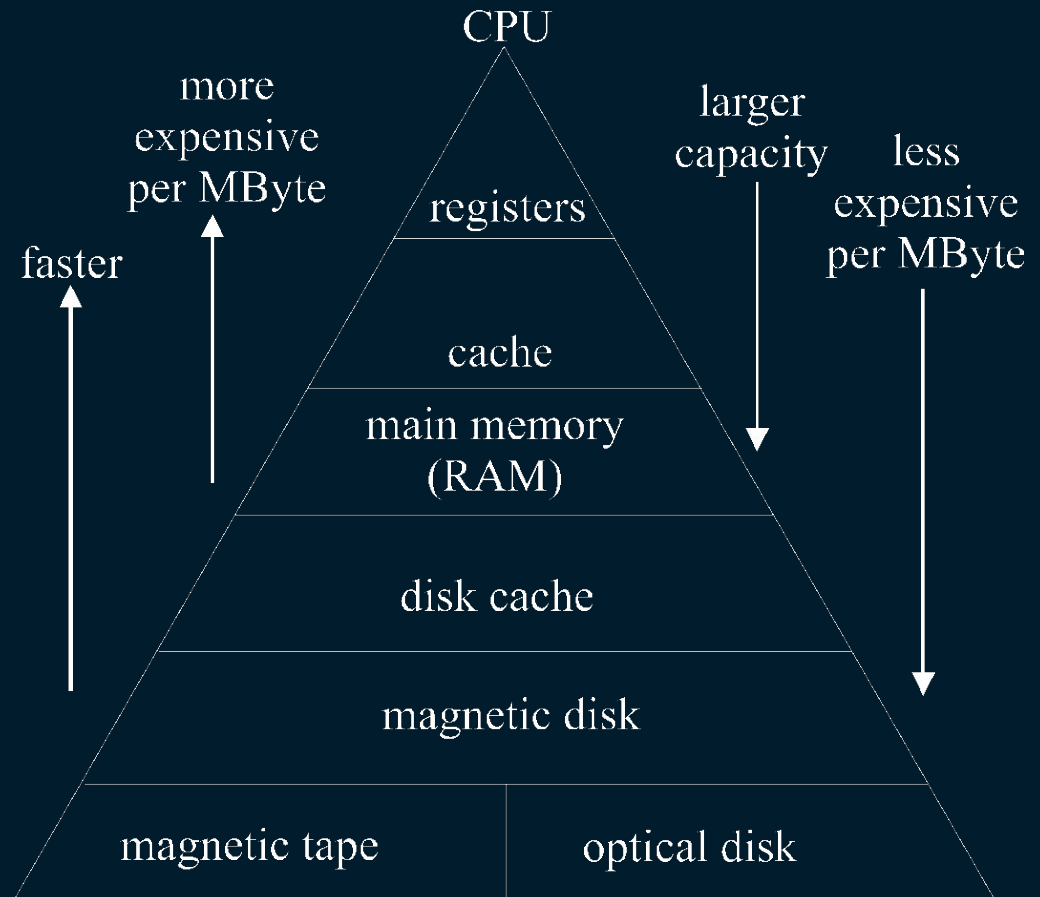
- Many types of memory used in computers
  - Primary: memory used for fast access
  - Secondary: memory used for long-term storage



# Memory Hierarchy

- Memory in general includes all storage in a system
- Memory is structured in a hierarchy to facilitate the transfer of data between the different levels

A structure that uses multiple levels of memories; as the distance from the processor increases, the size of the memories and the access time both increase.



# Principle of Locality

- Programs access a relatively small portion of their address space at any instant of time (instructions and data)
- Two types of locality
  - Temporal locality (locality in time) – if an item is referenced, it will tend to be referenced again soon
  - Spatial locality (locality in space) – if an item is referenced, items whose addresses are nearby are likely to be referenced

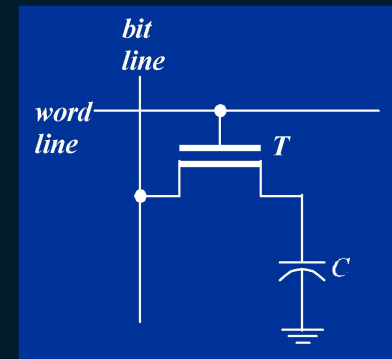
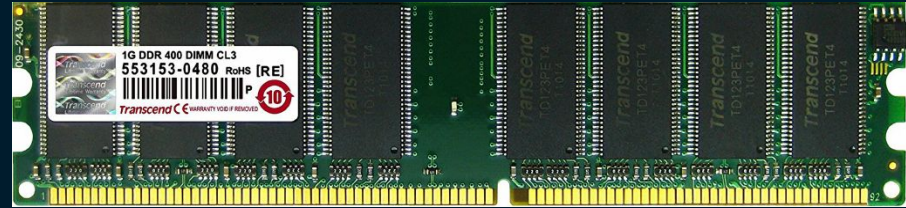
# Memory Technologies

- Four primary technologies for memory
  - DRAM (dynamic random access memory)
  - SRAM (static random access memory)
  - Magnetic
  - Optical
- Magnetic and optical are “disk” technologies and are not integrated circuits.
- Flash memories are a special form of IC memory
  - Used primarily as secondary storage and are discussed in I/O
- New technologies are being introduced

# DRAM

## Dynamic Random Access Memory

- Less area per bit, thus higher density (larger capacity)
- Slower than SRAM
- Less costly than SRAM
- Storage cell based on capacitor (stored charge)
  - Charge dissipates over time (microseconds)
  - Must periodically refresh contents (read & write)
  - Refresh takes 1-2% of active cycles
- Destructive reads (most DRAM)
  - Reads follow with write to restore
- Volatile memory
  - Contents lost when power off



# DRAM Evolution

- SDRAM (Synchronous DRAM)
  - Faster than DRAM due to its synchronization with the CPU's clock
  - Allows for quicker data processing
    - Data Integrity: SDRAM can handle more complex transactions more efficiently, which enhances overall data integrity and performance
- DDR (Double Data Rate)
  - Transfers data to the processor on both the risign and falling edge of the clock signal
  - Two transfers initiated per clock cycle.
  - DDR is identified by the generation
    - DDR5 is the fifth and current generation

# DDR Comparison

	DDR	DDR2	DDR3	DDR4	DDR5
Data Rate (MT/s)	266 - 400	533 - 800	1066 - 1600	2133 - 5100	3200 - 6400
Transfer Rate (GB/s)	2.1 - 3.2	4.2 - 6.4	8.5 - 14.9	17 - 25.6	38.4 - 51.2
Voltage (V)	2.5 - 2.6	1.8	1.35 - 1.5	1.2	1.1

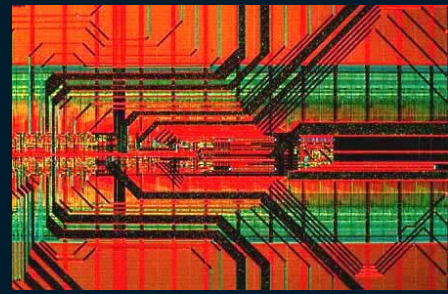
- **DDR Terminology**

- Data rate (MT/s): stands for mega transfers (million transfers) per second and is an accurate measurement of data rate transfer speeds.
- Transfer rate (GB/s): stands for Gigabits per second, and is a unit of data transfer rate equal to 1,000,000,000 bytes per second. (1 Megatransfers per Second = 0.008 Gigabyte per Second)

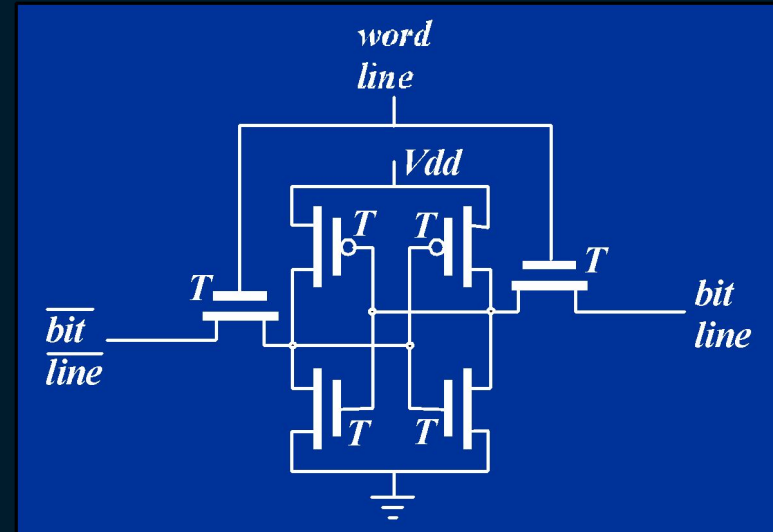


# SRAM

## Static Random Access Memory



- Built using gates (flip-flops)
- Takes more area per bit (less capacity than DRAM)
- Faster than DRAM
- Costs more than DRAM
- Storage cell based on transistor – no charge dissipation
  - No refresh cycle required
- Used for registers and cache memories
- Volatile (most types)
  - Contents lost when power off
  - There are some non-volatile SRAM
    - NV/SRAM
    - BBSRAM



# ROM

(Read Only Memory)

- A type of non-volatile memory used in computers
- Data stored in ROM cannot be electronically modified after the manufacture of the memory device
  - Except EPROM or EEPROM under special circumstances or with special equipment
- Read-only memory is useful for storing software that is rarely changed during the life of the system, also known as **firmware**.
  - The BIOS (Basic Input/Output System) is stored on a ROM chip and contains the boot loader used to initialize a computer system for loading OS
  - UEFI (Unified Extensible Firmware Interface) is a newer technology that will eventually replace BIOS

# Memory Terminology

- Except for CPU registers, minimum unit of data transfer in the memory hierarchy is a **block**
  - A block represents some predetermined number of words or bytes
  - Data is transferred from one level of memory to another level in the hierarchy in blocks of data
- Block data transfer is more efficient than moving bytes or words even though the CPU only needs a word or byte at a time
  - Longer bus distances are slower so bandwidth is increased by transferring larger amounts of data

# Hits and Misses

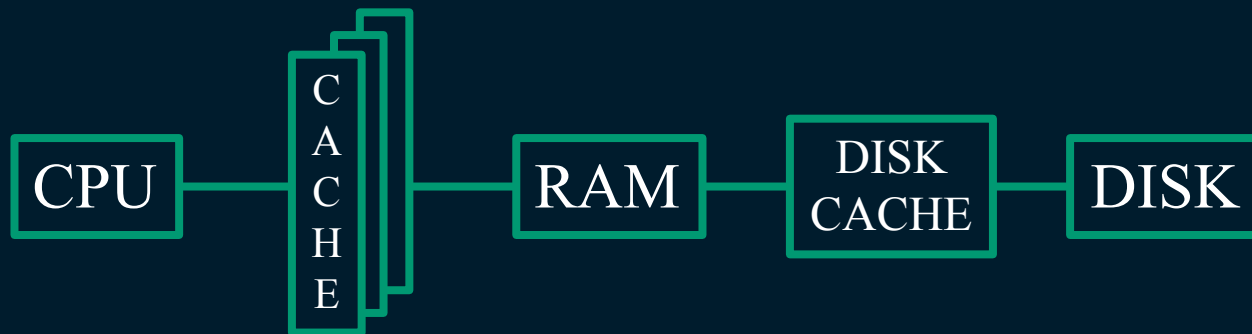
- A **hit** is when the processor finds the data it needs in an upper level of memory
- A **miss** is when the processor doesn't find the data it needs in an upper level of memory and has to access a lower level memory
- The **hit rate (hit ratio)** is the fraction of memory accesses where the data is found in the upper level memory
  - A hit or miss can occur at any level of the hierarchy

## Hits and Misses (con't)

- The **hit time** is the time to access the upper level memory plus the time to determine if the data is there (hit/miss)
- The **miss rate** is the fraction of memory accesses not found in the upper level memory;  $1 - \text{hit rate}$
- The **miss penalty** is the time to replace a block in the upper level memory plus the time required to deliver the data to the CPU

# Cache Memory

- **Cache** memory is any memory that is designed to take advantage of locality of reference
- In most contemporary computer systems, cache memory can be found:
  - Between the CPU registers and main memory (RAM)
  - Between main memory and disk storage (specifically called **disk cache** to differentiate it from the CPU/main memory cache)



# CPU Cache Memory

- CPU cache memory is much faster than main memory (RAM)
  - On CPU chip cache operates at the same clock rate as CPU
  - Most caches use SRAM
- Cache memory is much smaller than main memory
- Cache memory holds copies of data that are stored in main memory
- This size difference means that cache cannot hold all the contents of main memory
  - The key is to make sure the cache has the right contents from main memory when the CPU needs it

# Cache Mapping

- Multiple addresses in main memory are mapped to the same cache location
  - Because cache is smaller than main memory
  - only one memory block can be in a cache location at a time
- **Block identification** refers to how a block is found in the cache and how you recognize that it is the correct block you need



# Cache Mapping

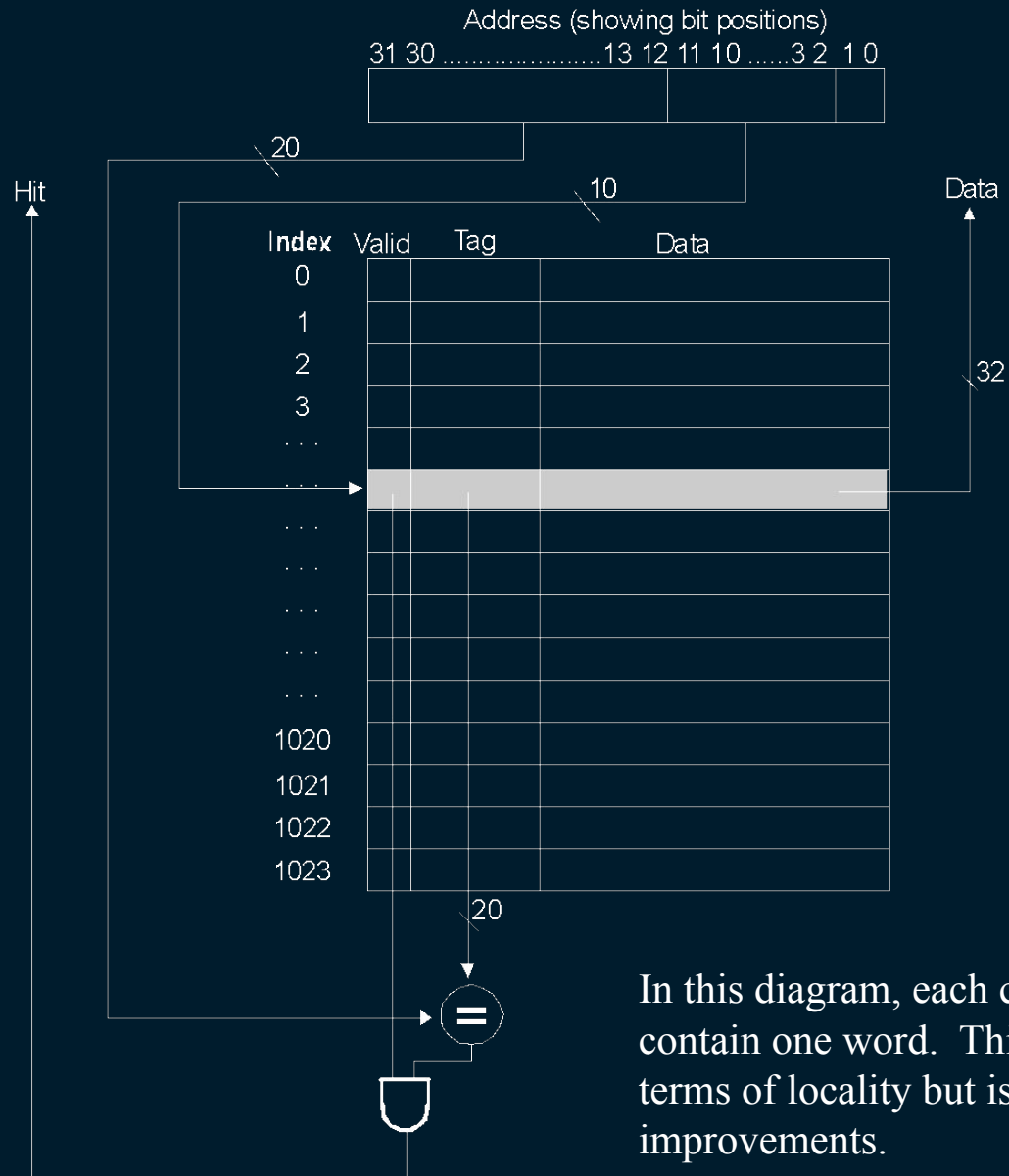
- **Cache mapping** is the process of using the address issued by the CPU to identify a cache location
- Simplest mapping scheme is called **direct mapping**
- Each main memory block is mapped to a specific cache block
- **Cache address = block address modulo number of blocks in the cache**

# Cache Structure

- Definitions

- Each block in the cache has an **address tag** that contains a value used to identify if the correct word is in the cache or not
  - **Tag bits** are some number of upper bits of CPU memory address
- An **index** points to the appropriate location in the cache
  - Index bits are some number of lower bits of CPU address
- A **valid bit** indicates whether the data in the cache is valid or not
  - Valid bit is set by the cache controller when cache values are loaded

# Diagram of Cache Mapping



# Cache Control

- Ideally, when the CPU accesses memory, you want the item, whether instruction or data, to be in the cache for retrieval.
- When a miss occurs, extra work must be done to access the lower level memory to copy the required item to cache and deliver it to the CPU so it can continue processing
- The cache controller generates the miss signal sent to the main memory control to refill the cache.

# Processor Stall

- The process of refilling the cache on a miss causes the processor to stall during the lower level memory access.
- The processor control unit initiates the stall and triggers the cache controller to initiate the memory access.
- Once the memory access has completed, the processor control unit restarts at the point where the miss occurred.

# Writing to Cache

- If on a store instruction, an item were written only to the cache, the main memory would not have the same value at the address represented by the cache.
- This inconsistency between cache and main memory is called the **cache coherency** problem.
- This problem is even more severe on multiprocessor systems with multiple caches accessing a single common RAM.

# Cache Writes

- Two primary approaches for writes
  - Write-through: the information is written to both the cache block and lower level memory block
    - Not good for performance due to long waits for memory writes
    - Write buffer could be used to hold data until it is written and free the cache for continued execution but buffer can fill up and slow memory access

## Cache Writes (con't)

- Write-back: the information is written only to the cache block; the write to memory occurs only when the cache block is replaced, typically by a read miss
  - Better for performance
  - More complex to implement than write-through
  - Need to add dirty bit to cache blocks to identify modified data that must be written before being replaced
    - No need to write unchanged data to memory



# Associative Cache

- A block of data can be placed anywhere in the cache
- No restriction on the placement of blocks
- Any combination of blocks can be present in the cache
- To find a given block, the entire cache must be searched
- Cache entries are searched in parallel using a comparator associated with each entry
- Costly to implement
- Found in very high performance, special purpose systems

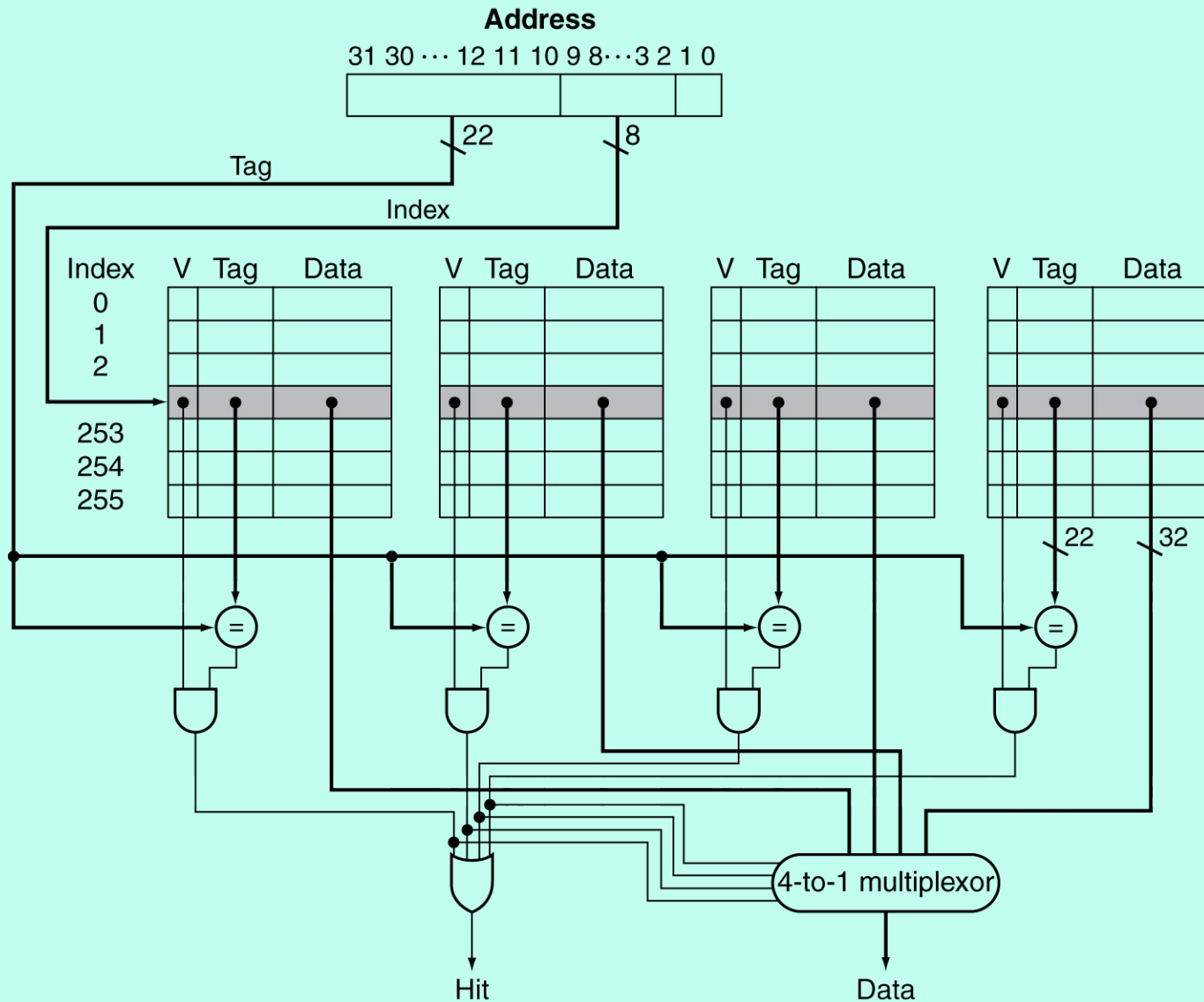
# Set Associative Cache

- A block of data can be placed in a limited number of places in the cache
- A set associative cache with  $n$  locations for a block is called an  $n$ -way set associative cache
- A block in memory maps to a specific set identified by the index field and a block can be placed in any element of that set
- Very good compromise between direct mapped and fully associative cache

# Set Associative Cache

- Only the elements of a set need to be searched to find the needed block
- Microprocessors typically use 2-, 4-, and 8-way set associative caches
- The mapping formula for set associative cache is  
 $(\text{block number}) \bmod (\text{number of sets in cache})$

# Set Associative Cache Design



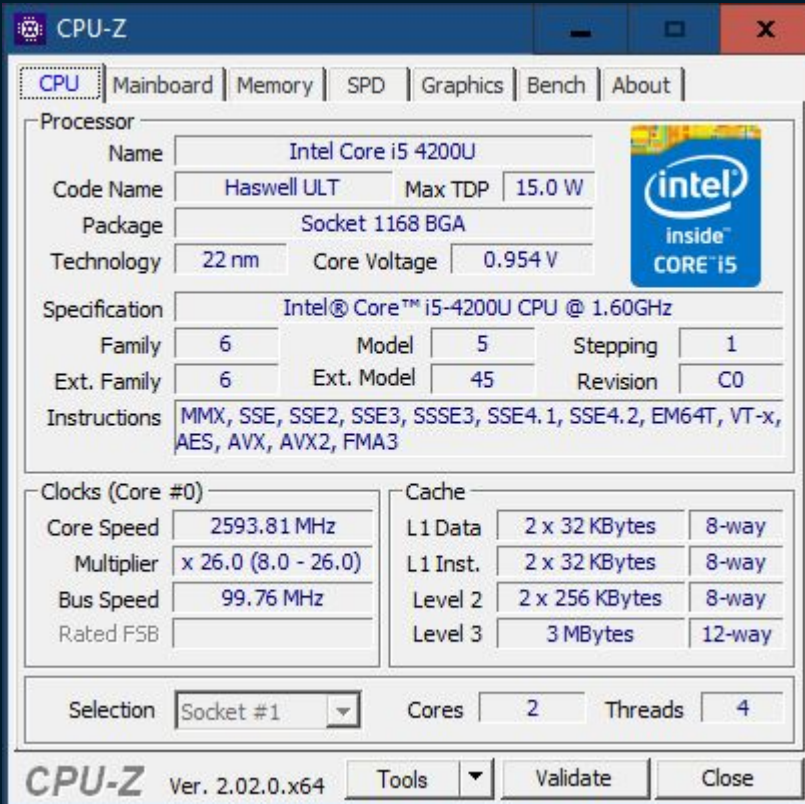
# Block Replacement Strategies

- If cache is full, which block is selected to be replaced by a new block of data?
  - Only in set-associative or fully associative caches
- Random
  - Candidate blocks are selected at random which spreads the allocation uniformly
- Least-recently used
  - The block replaced is the one that has gone the longest without being referenced (most common)
- Least-frequently used
  - The block replaced is the one that has been referenced less than other blocks

# Multi-Level Caches

- Optimize the trade-off between memory access speed and storage capacity
- Level 1
  - Smallest but fastest access (per core in multicore)
  - Has largest bandwidth due to fast access
  - Physically small
- Level 2
  - Larger than level 1 but slightly slower access
  - May be per core (typical) or shared with core pairs
- Level 3
  - Also included on-chip in contemporary CPUs
  - Larger than level 2 and shared among all cores

# Cache Configuration




CPU-Z Ver. 2.02.0.x64

**CPU** | Mainboard | Memory | SPD | Graphics | Bench | About

**Processor**

Name	Intel Core i5 4200U		
Code Name	Haswell ULT	Max TDP	15.0 W
Package	Socket 1168 BGA		
Technology	22 nm	Core Voltage	0.954 V



Specification: Intel® Core™ i5-4200U CPU @ 1.60GHz

Family	6	Model	5	Stepping	1
Ext. Family	6	Ext. Model	45	Revision	C0

Instructions: MMX, SSE, SSE2, SSE3, SSSE3, SSE4.1, SSE4.2, EM64T, VT-x, AES, AVX, AVX2, FMA3

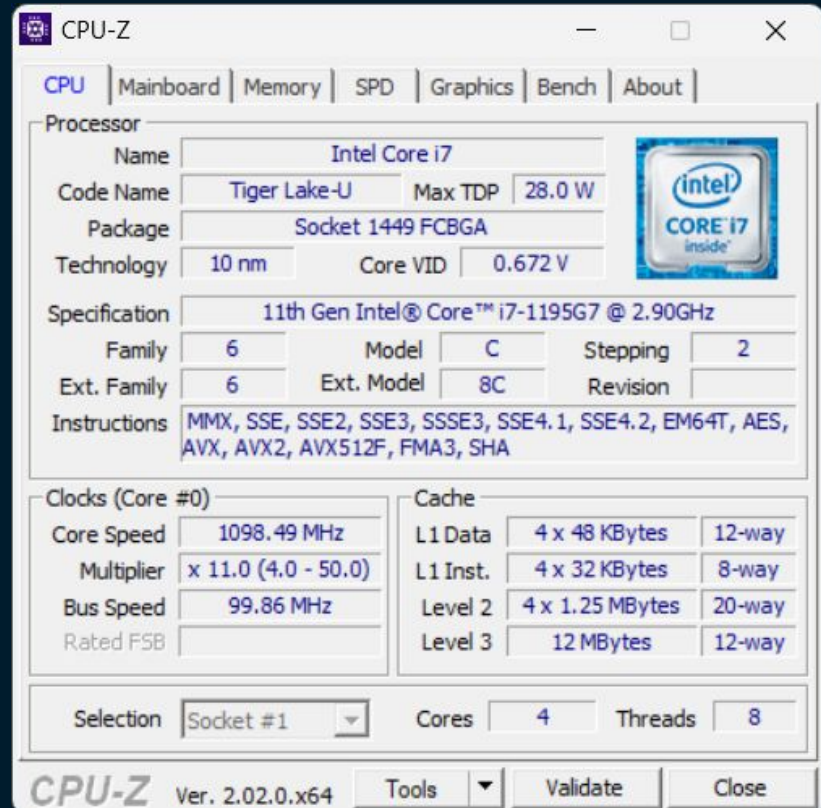
**Clocks (Core #0)**

Core Speed	2593.81 MHz
Multiplier	x 26.0 (8.0 - 26.0)
Bus Speed	99.76 MHz
Rated FSB	

**Cache**

L1 Data	2 x 32 KBytes	8-way
L1 Inst.	2 x 32 KBytes	8-way
Level 2	2 x 256 KBytes	8-way
Level 3	3 MBytes	12-way

Selection: Socket #1 | Cores: 2 | Threads: 4




CPU-Z Ver. 2.02.0.x64

**CPU** | Mainboard | Memory | SPD | Graphics | Bench | About

**Processor**

Name	Intel Core i7		
Code Name	Tiger Lake-U	Max TDP	28.0 W
Package	Socket 1449 FCBGA		
Technology	10 nm	Core VID	0.672 V



Specification: 11th Gen Intel® Core™ i7-1195G7 @ 2.90GHz

Family	6	Model	C	Stepping	2
Ext. Family	6	Ext. Model	8C	Revision	

Instructions: MMX, SSE, SSE2, SSE3, SSSE3, SSE4.1, SSE4.2, EM64T, AES, AVX, AVX2, AVX512F, FMA3, SHA

**Clocks (Core #0)**

Core Speed	1098.49 MHz
Multiplier	x 11.0 (4.0 - 50.0)
Bus Speed	99.86 MHz
Rated FSB	

**Cache**

L1 Data	4 x 48 KBytes	12-way
L1 Inst.	4 x 32 KBytes	8-way
Level 2	4 x 1.25 MBytes	20-way
Level 3	12 MBytes	12-way

Selection: Socket #1 | Cores: 4 | Threads: 8

Images from CPU-Z program

# Harvard Architecture (Split Caches)

- Physically separate cache storage and signal pathways for instructions and data
- Term originated from the Harvard Mark I relay-based computer, which stored instructions on punched tape (24 bits wide) and data in electro-mechanical counters (23 digits wide)
  - Actually two different physical memories
  - Contrast with von Neumann architecture with single main memory
- Access to instructions presents a different locality profile than access to data
  - More sequential versus random
  - Level 1 caches are usually split (instructions vs. data)



# The Three Cs

(Categories of Misses)

- **Compulsory misses** – cold start misses – caused by first access to a block that has never been in the cache
  - Referenced item has never been in the cache
- **Capacity misses** – caused when cache cannot contain all the blocks needed during program execution
  - Referenced item has been in the cache, but space was tight and it was forced out
- **Conflict misses** – collision misses – occur in set associative or direct mapped caches when multiple blocks compete for the same set
  - Referenced item was in the cache, but the cache was not associative enough, so it was forced out

# Cache Block Size

- Also known as the cache line
- Cache block size represents the unit of data exchanged between the cache and main memory.
  - If data is not found in the cache, a block of data containing the required information gets loaded.
- In common systems, cache block sizes range from 16 bytes to 256 bytes.
- When determining the ideal block size, you must seek a balance:
  - larger blocks can take better advantage of spatial locality,
  - but too large blocks can waste cache space and increase miss penalty.

# Hit/Miss Rate Effect

- As we increase the block size, we would expect the miss rate to decrease; the desired effect.
- Example: cache block size = 4 words
  - a reference to address 16 will result in a block containing address data for 16-28.
- Subsequent references to 20, 24, and 28 will result in hits.
- Increasing the block size to some optimal number will result in a decrease in miss rate.

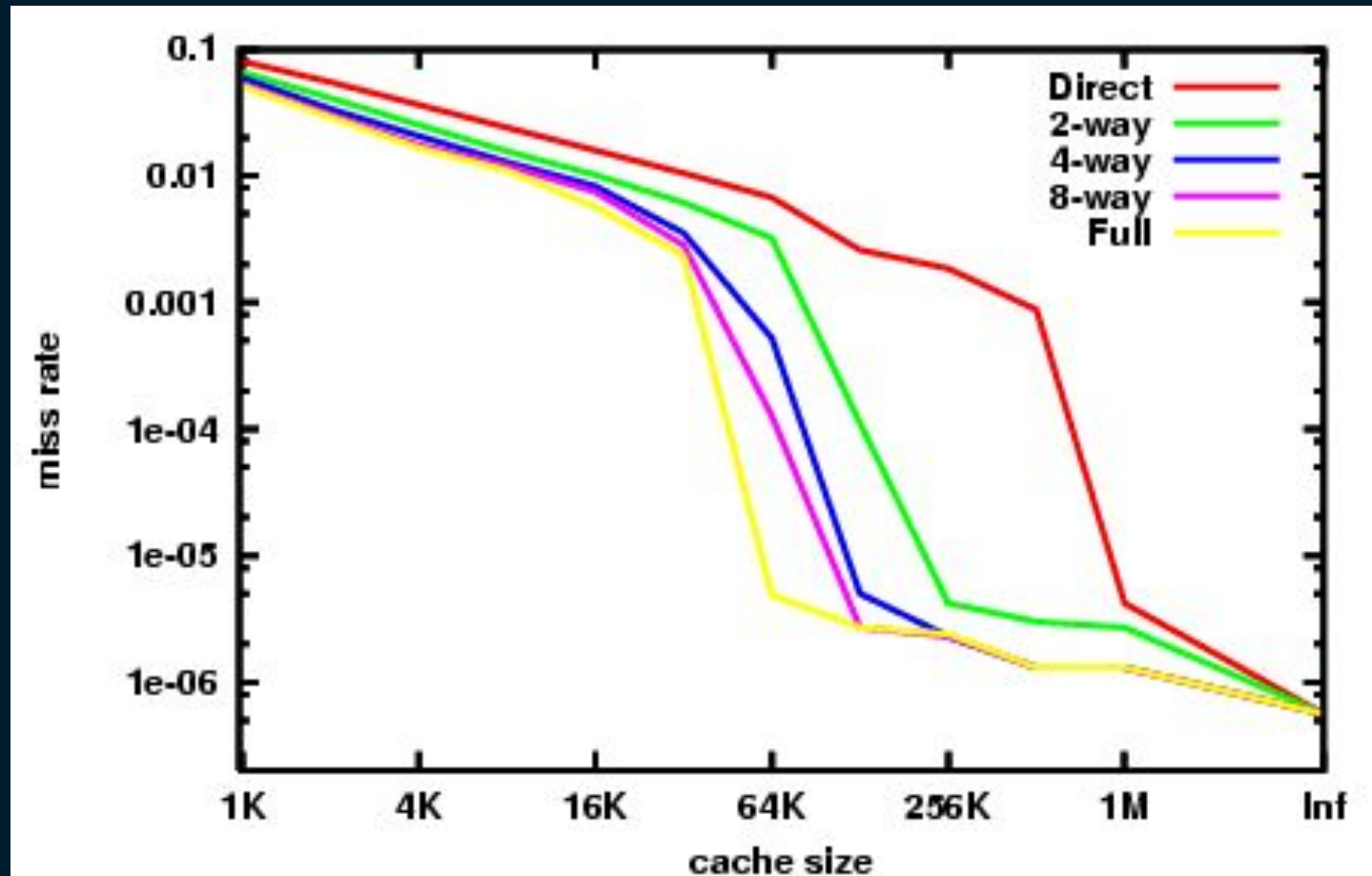
## Too Much...

- Increasing the block size to too great a value can actually cause the miss rate to increase because you exceed the nominal boundaries of spatial locality.
- You end up replacing the block in cache before you have accessed all the words within the block.
- Example, a block of instructions containing several branch instructions with target labels residing outside the branch block.

# Block Size vs Miss Penalty

- Making the block size too big compounds the miss penalty.
- The time to transfer a block of data has two components:
  - The latency (access time) to the first word
  - The transfer time for the remainder of the block
- The bigger the block, the more data that has to be transferred thus the more potential time the CPU must wait for cache to be loaded.
- Most all contemporary processors have cache block sizes of 64 bytes.

- Miss rate vs. cache size & associativity



# Memory Performance

- AMAT (Average Memory Access Time)
  - Common metric to analyze and measure memory system performance
  - Uses hit time, miss rate and miss penalty to quantify memory performance
  - Calculated as an average based on the overall memory architecture
    - Includes main memory and the number of levels of cache

# Example

- Memory hierarchy design:
  - L1 cache requires 1 cycle to access and has a miss rate of 10%
  - L2 cache requires 10 cycles to access and has a miss rate of 2%
  - DRAM requires 80 cycles to access and must have a miss rate of 0%
- What is the AMAT for this memory system?
- General AMAT formula:

$$AMAT = Hit\ time + Miss\ ratio \times Miss\ penalty$$

- Expansion required for multilevel caches



# AMAT Example Calculation

$1 +$   $\leftarrow$  Always access L1 cache

$0.10 \times 10 +$   $\leftarrow$  Probability miss in L1 cache  $\times$   
time to access L2

$0.10 \times 0.02 \times 80$   $\leftarrow$  Probability miss in L1 cache  $\times$   
probability miss in L2 cache  $\times$   
time to access DRAM

$= 2.16$  cycles

# Alternative AMAT Example Calculation

$0.90 \times 1 +$        $\leftarrow$  Probability hit in L1 cache  $\times$  time to access L1

$0.10 \times 0.98 \times$        $\leftarrow$  Probability miss in L1 cache  $\times$   
 $(10 + 1) +$       probability hit in L2  $\times$  time to access  
L1 then L2

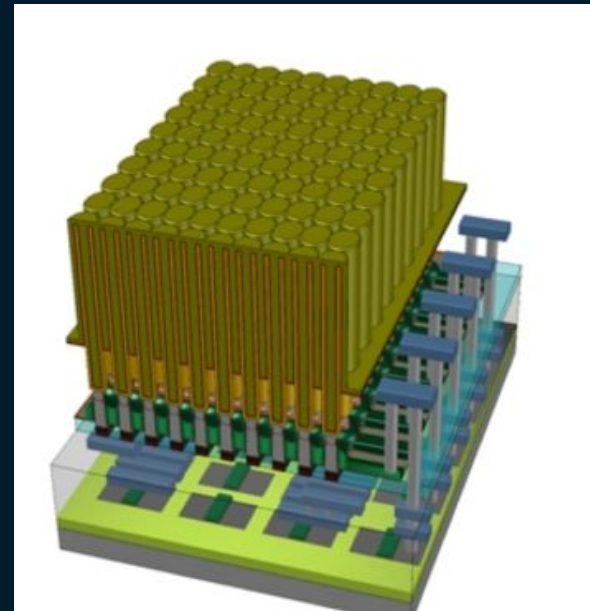
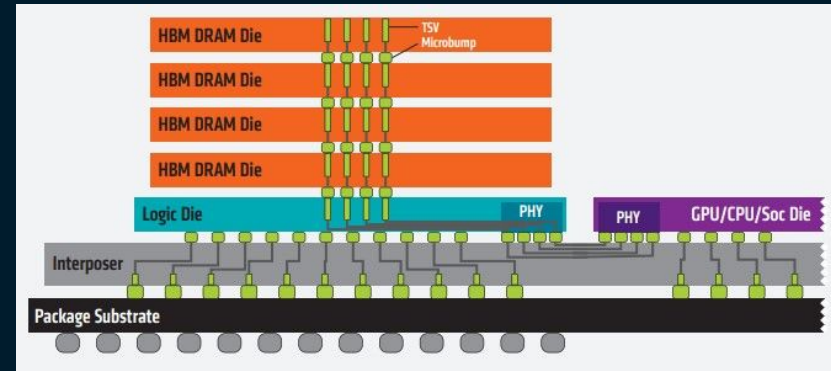
$0.10 \times 0.02 \times$        $\leftarrow$  Probability miss in L1 cache  $\times$   
 $(1 + 10 + 80)$       probability miss in L2 cache  $\times$  time to  
access L1 then L2 then DRAM

$= 2.16$  cycles

# Advanced Memory Design

- 3-D Stacking
  - Wide-I/O
  - Either package or chip layering to increase data width
- 3-D Integration
  - Same concept as 3-D stacking only at the integration level

These new designs called high-bandwidth memory (HBM) are being implemented in some new high-performance CPUs.



# Memory Terminology / Vocabulary

- Memory hierarchy
- Principle of locality
  - - temporal
  - - spatial
- DRAM
  - - SDRAM
  - - DDR
- SRAM
- Block
- Hits/misses
- Hit ratio (hit rate)
- Hit time
- Miss rate
- Miss penalty
- Cache memory
- Cache mapping (general)
- Direct mapping
- Cache structure
  - - index
  - - tag bits
  - - valid bit
- Cache writes
  - - write-through
  - - write-back
- Associative cache
- Set associative cache
- Block replacement
  - - random
  - - LRU (least recently used)
  - - LRU (least frequently used)
- Three C's
  - - compulsory
  - - capacity
  - - conflict
- Harvard memory architecture
- Block size vs. miss penalty
- AMAT & calculation
- HBM (high-bandwidth memory)

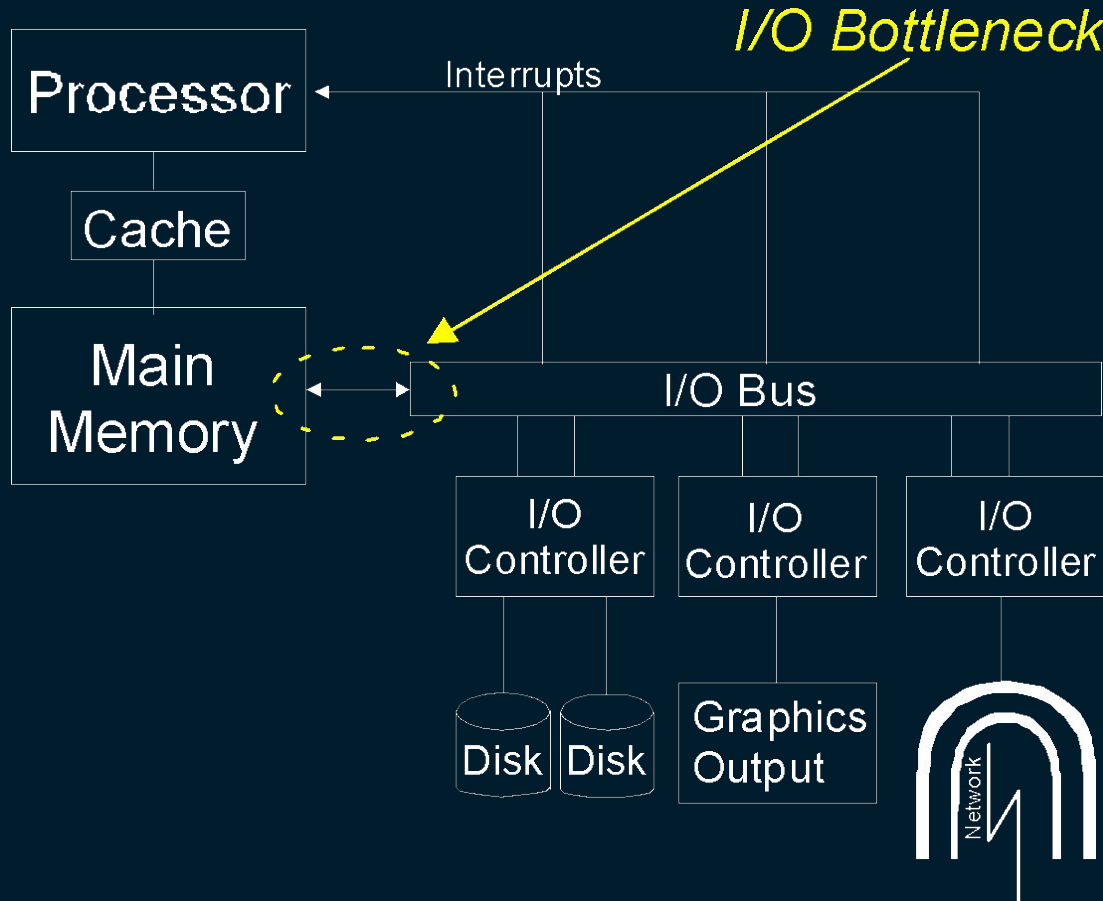
# Input/Output

- What is important about I/O?
  - Users get frustrated if computer hangs and must be rebooted but they are furious if their storage system crashes and they lose information.
  - Dependability is a much higher priority for storage systems than for computation.
  - I/O systems emphasize dependability and cost.
    - The CPU emphasis is on performance and cost.
  - I/O systems have to plan for expandability and diversity of devices
    - Not a concern of the CPU
    - Expandability is related to storage capacity and device support
  - I/O system performance is more complex than CPU performance
    - Depends on device characteristics, connections and interfaces with the rest of the system, memory hierarchy and OS

# Characteristics of I/O Devices

- Behavior
  - Input (read), output (write), storage (read/write)
- Partner
  - What's at the other end: human or machine?
- Data rate
  - How fast can data be transferred between devices, memory, or the CPU?

# von Neumann I/O Bottleneck

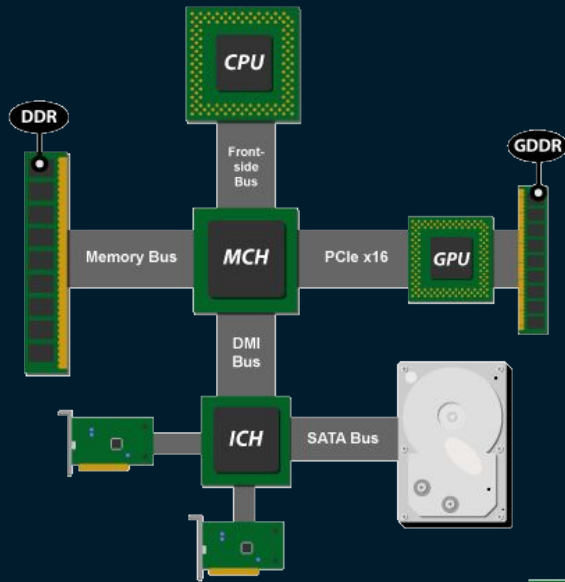


A typical collection of I/O devices and connections.

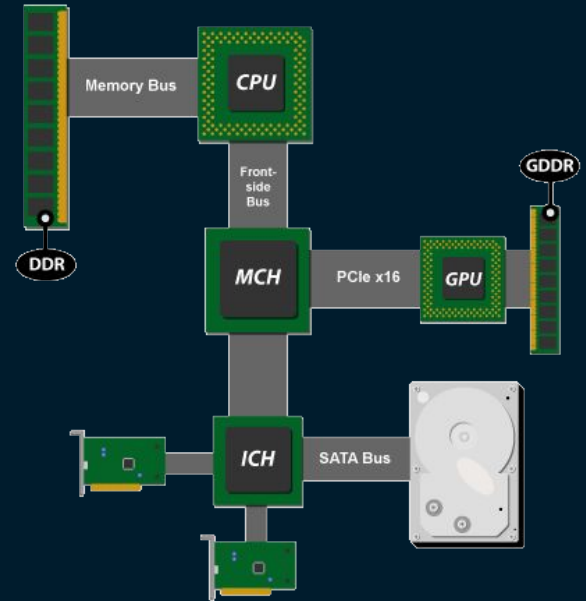
Connections between devices are usually by buses although some systems have more elaborate switching networks.

I/O bottleneck has traditionally been a problem due to single I/O bus structures. Multiple bus architectures have improved I/O operability.

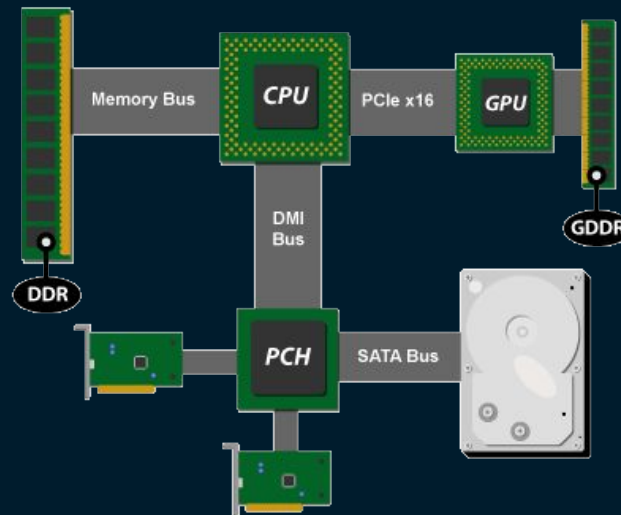
# Newer System Configurations



Intel Pre-Nehalem



AMD Single-Socket



Intel i5/i7 (P55)

MCH = Memory Controller Hub  
ICH = I/O Controller Hub  
PCH = Peripheral Controller Hub



# Magnetic Disk Drives

- Primary permanent storage for computers
- Nonvolatile storage
- Two kinds of magnetic disks
  - Floppy disks – magnetic storage material is coated on a flexible material (plastic or acetate)
    - low capacity, slow access, mostly obsolete
  - Hard disks – magnetic storage material is coated on glass or ceramic
    - High capacity and growing, faster access and improving
    - Not going away anytime soon

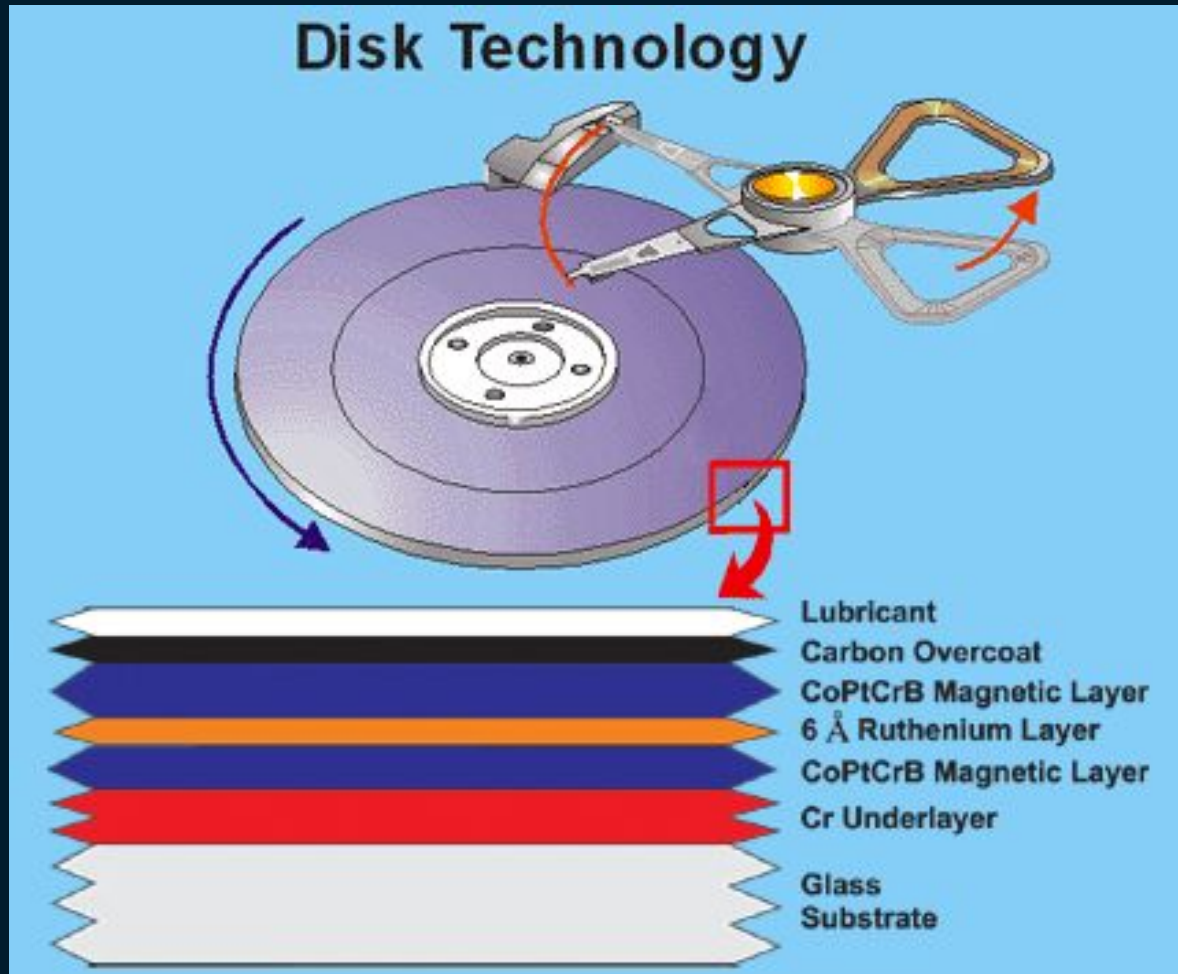


# Hard Disks

- Higher density (more storage)
  - Storage sizes are in the Gigabytes and Terabytes
  - Largest today is 30 Terabytes (Seagate Exos Mozaic 3+)
- Higher data transfer rates because disks spin faster
  - Rotational speeds of 5400, 7200, 10500 and 15000 RPM are common today
- Usually consist of multiple disks on a spindle
- Disks can range in size from <1 in. to 14 in.
  - 3.5" for desktops, 2.5" for laptops (common usage)
  - Other sizes from 0.84" to 14" are obsolete

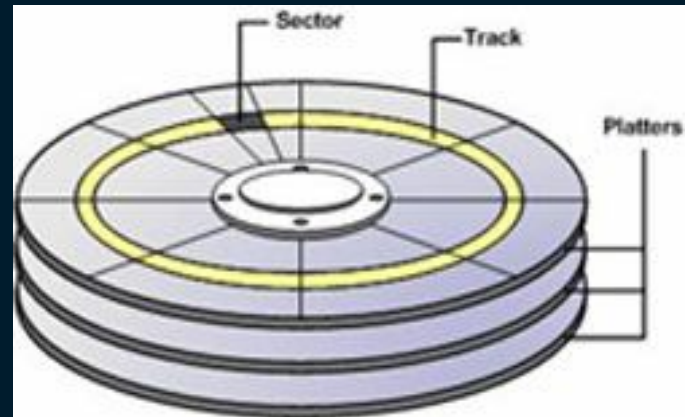


# Disk Surfaces Are Layered



# Disk Organization

- Each disk is divided into concentric circles called **tracks**.
- Both sides of a disk are used for storage.
- Tracks are subdivided into **sectors** where the data is written.
- A sector is the smallest unit of data that can be written.
  - The standard size of a sector was typically 512 bytes.
  - Advanced Format (March 2010) defines sectors of 4096 bytes.
  - All sectors in all tracks are the same size.
- All the tracks in the same vertical plane across all the disks is called a **cylinder**.



# Read/Write

- The mechanism that actually reads and writes data is called the read/write head which is positioned above the disk.
- The heads are electro-magnetic like the record/playback heads on a tape recorder only much smaller ( $\sim 8 \mu\text{m}$  wide).
- Electrical pulses representing binary data are converted to magnetic pulses to activate the storage material on the disk.
- During reading, the magnetic data is read and converted back to electrical information.

# Read/Write

- The read/write head is movable.
- It is mounted on an arm that can move it across the surface of the disk to read different tracks. The heads don't actually touch the surface.
- If the drive has multiple platters, each usable surface will have a read/write head that all move in parallel so that at any instant the heads will be positioned over all the tracks that comprise a cylinder.
- In some older disk drives, the heads were fixed. The arm holding the heads did not move and there was one head per track per disk surface.

# Disk Data Access

- Data access is a three stage process:
  - The head is positioned over the proper track – called a **seek**. The time to accomplish this is referred to as seek time.
  - Wait for the desired sector to rotate under the head. The time for this is called **rotational latency** (rotational delay).
  - Transfer the block of bits (sector). The term for this is **transfer time**.
- A disk controller handles the details of the read/write process and adds some amount of controller time to the process.

## Example: Disk Read Time

- What is the average time to read a 512-byte sector for a disk rotating at 7200 RPM if the seek time is 6 ms, the transfer rate is 50 MB/sec, and the controller overhead is .2 ms?

$$\begin{array}{ccccccc} \text{Average} & & \text{Average} & & \text{Transfer} & & \text{Controller} \\ \text{Seek} & + & \text{Rotational} & + & \text{Time} & + & \text{Overhead} \\ \text{Time} & & \text{Latency} & & & & \\ \\ 6.0 \text{ ms} & + & \boxed{\frac{.5^*}{7200 \text{ RPM}}} & + & \frac{.5 \text{ kB}}{50 \text{ MB/sec}} & + & 0.2 \text{ ms} \\ \\ & & 6.0 + 4.17 + 0.01 + 0.2 = & & 10.38 \text{ ms} \end{array}$$

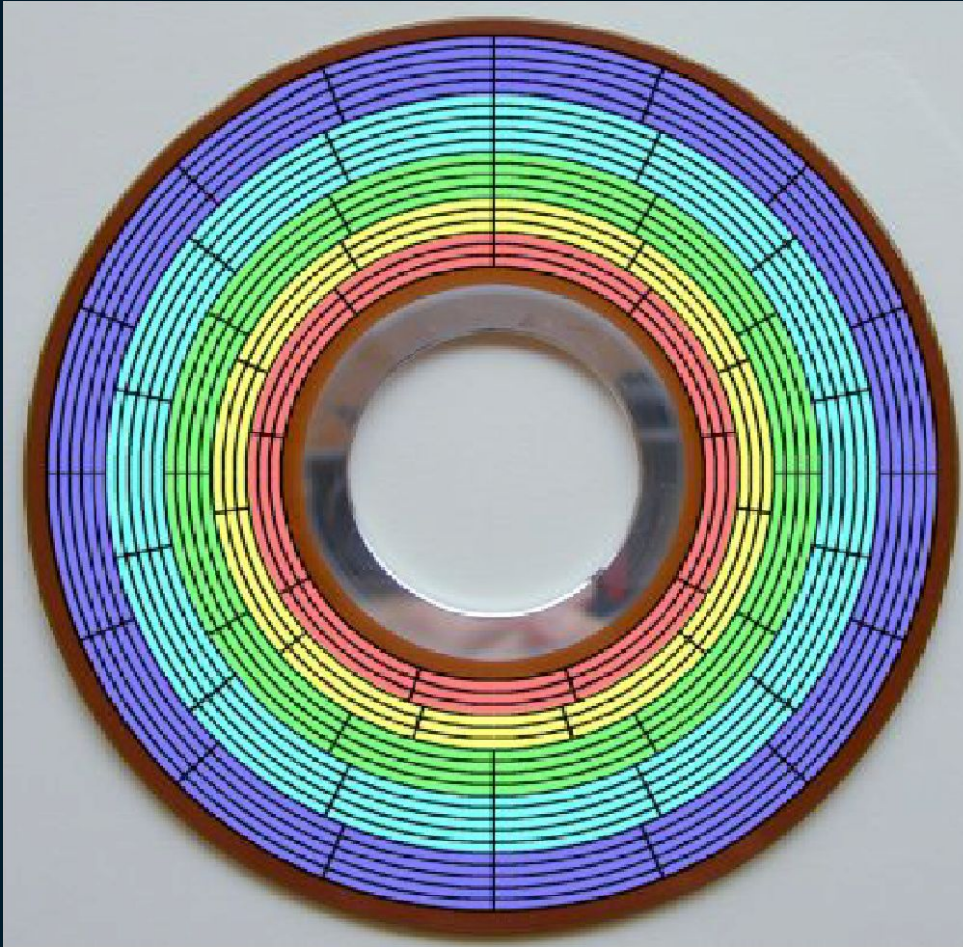
Actual rotational latency depends on the difference between the current position and the target position which could be in the range 0 revolutions to 1 full revolution. Therefore, the average rotational latency is used calculated as  $(0+1)/2$ .  $[(0.5/7200)*60 = 0.00417\text{s} = 4.17\text{ms}]$



# Improved Storage Utilization

- Newer disk drives utilize the difference in linear track differences by storing more data on the outside tracks.
- Technique is called **zoned bit recording**.
- Tracks are grouped into zones depending on their distance from the center of the disk.
- Each zone is assigned a number of sectors per track.
- The outer zones have more sectors and store more data than the inner zones.
- Results in higher storage capacities for disks.

# Zoned Bit Recording



A graphical illustration of zoned bit recording. This model hard disk has 20 tracks. They have been divided into 5 zones, each of which is shown as a different color. The size of each sector is fairly constant and as you go from the inner zones to the outer zones, the number of sectors per track increases.

The total number of sectors on this disk using zoned bit recording is 268.

Blue:  $5 \times 16 = 80$

Cyan:  $5 \times 14 = 70$

Green:  $4 \times 12 = 48$

Yellow:  $3 \times 11 = 33$

Red:  $3 \times 9 = 27$

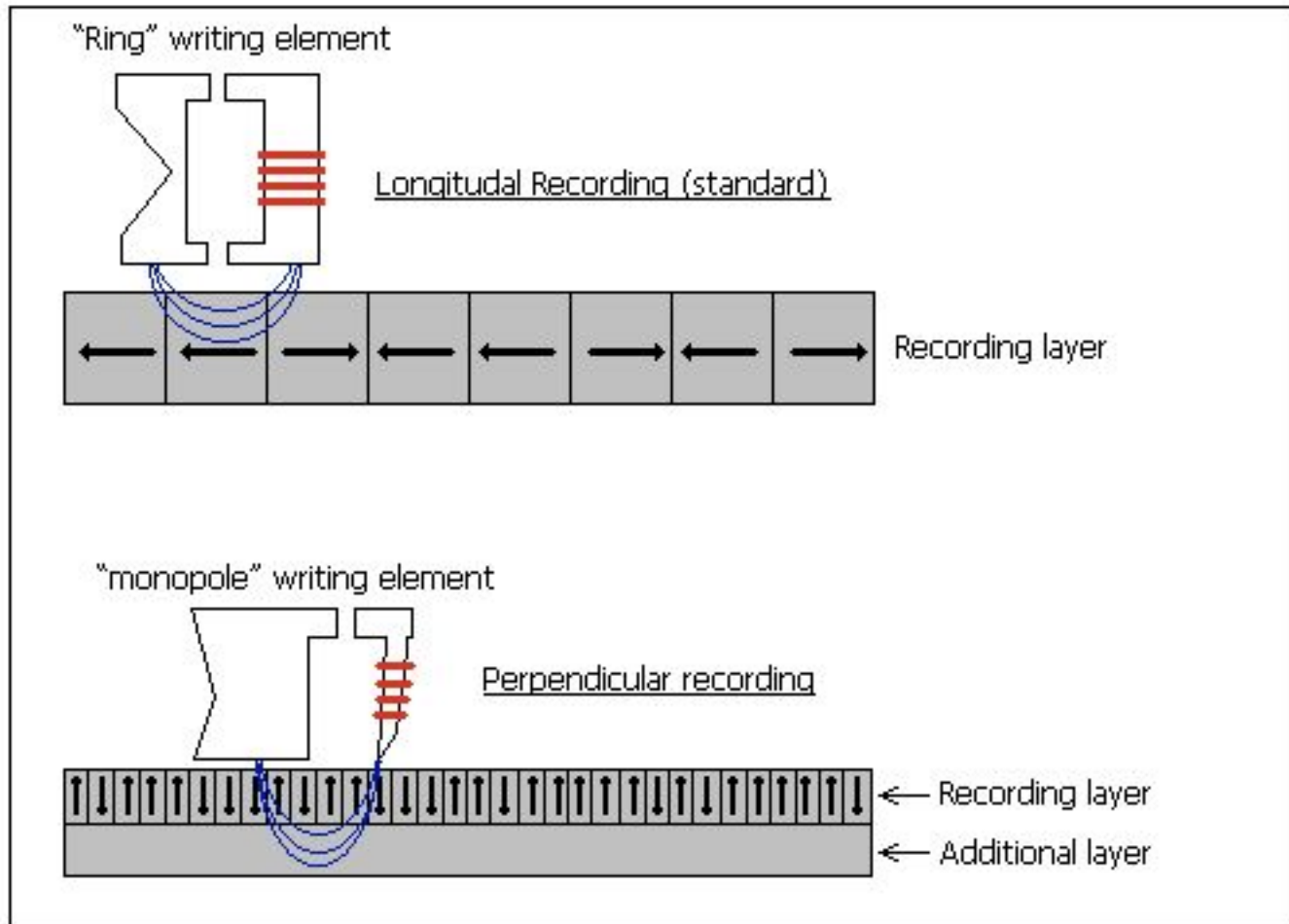
Total: ----- 268

Older disks would be limited to 9 sectors per track (the maximum possible for the inner tracks) resulting in only 180 sectors.

# Data Encoding

- Additional gains in storage capacity have been realized by new data encoding techniques
- Older disk drives used **longitudinal encoding**
  - estimated limit of 100 to 200 gigabits per square inch
- New disk drives implement **perpendicular encoding** which takes less physical space
  - predicted to allow information densities of up to around 1 Tbit/sq. inch (1000 Gbit/sq. inch).

# Longitudinal vs Perpendicular

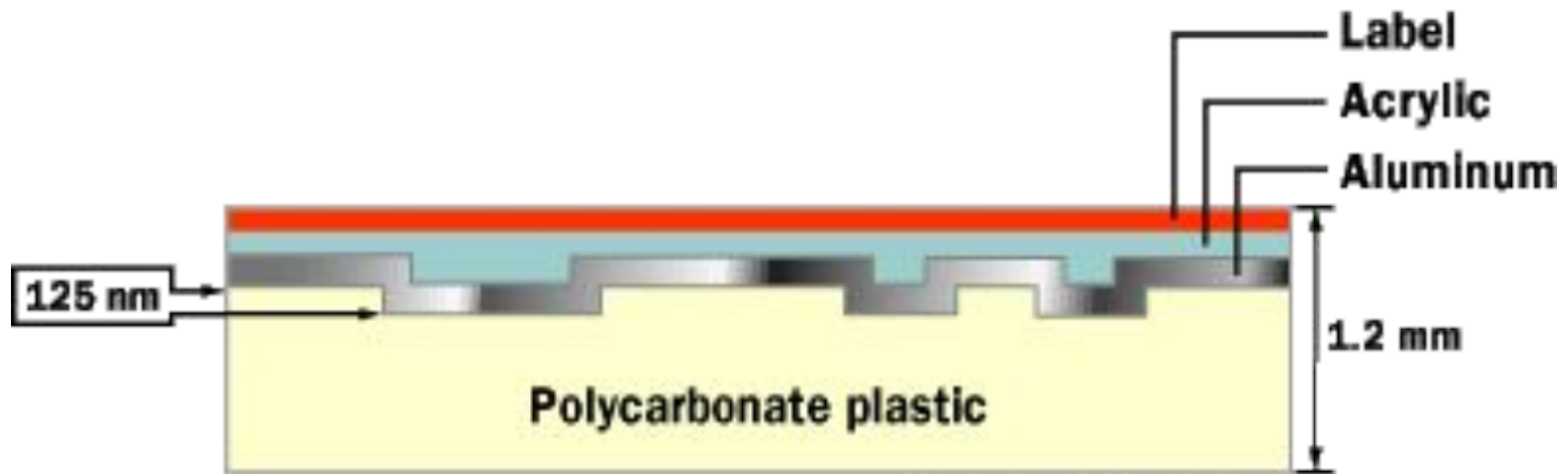


# CD & DVD

- CD and DVD are optical storage technology
- Use a laser to read from and write data on a special optical surface
  - Red or IR lasers used; Blu-ray technology uses blue laser
- Data is written in a spiral and is contained in a single track (from center to outside)
- Unlike magnetic disks, optical disks spin at a **constant linear velocity** (CLV) so that the data at every point along the track is read at the same rate.
- The drive motor will speed up or slow down depending where the laser is positioned along the track.

# Optical Technology

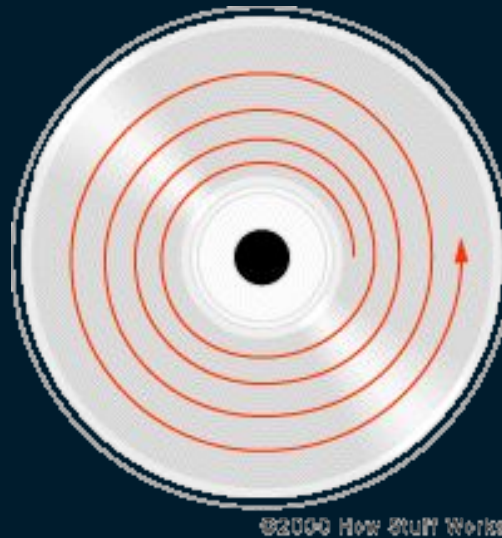
- Optical laser is used as the read/write tool
- Disk surface is aluminum encased in a plastic polymer.



This technology is used for mass-produced commercial media.

# CD

- Unlike magnetic disks, CD tracks are spiral



The spiral track beginning is at the center of the disk. This makes it possible to have varying sizes of CDs.

# Pits and Bumps

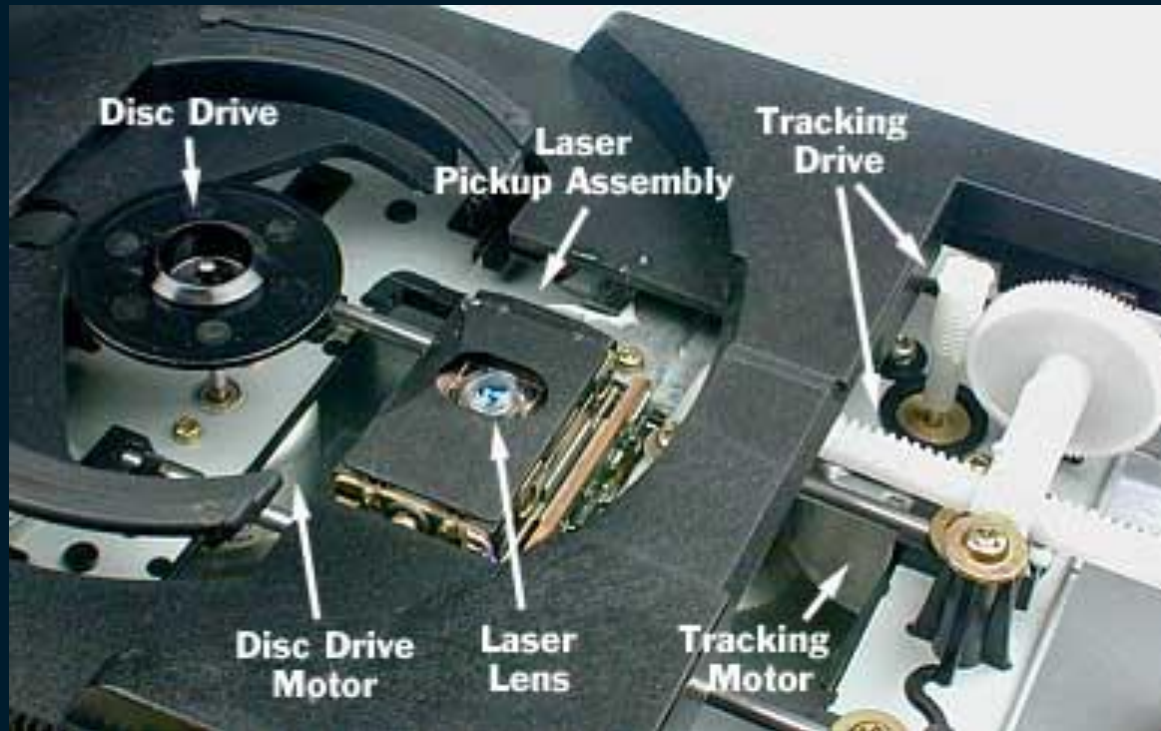
- Digital information is encoded in the aluminum layer in the form of pits and bumps.



- CD-R technology uses a photosensitive dye whose reflective properties are changed by the writing laser.
- CD-RW technology uses a metallic alloy instead of a dye.
  - The writing laser is used to heat and alter the properties (amorphous vs. crystalline) of the alloy, and hence change its reflectivity.

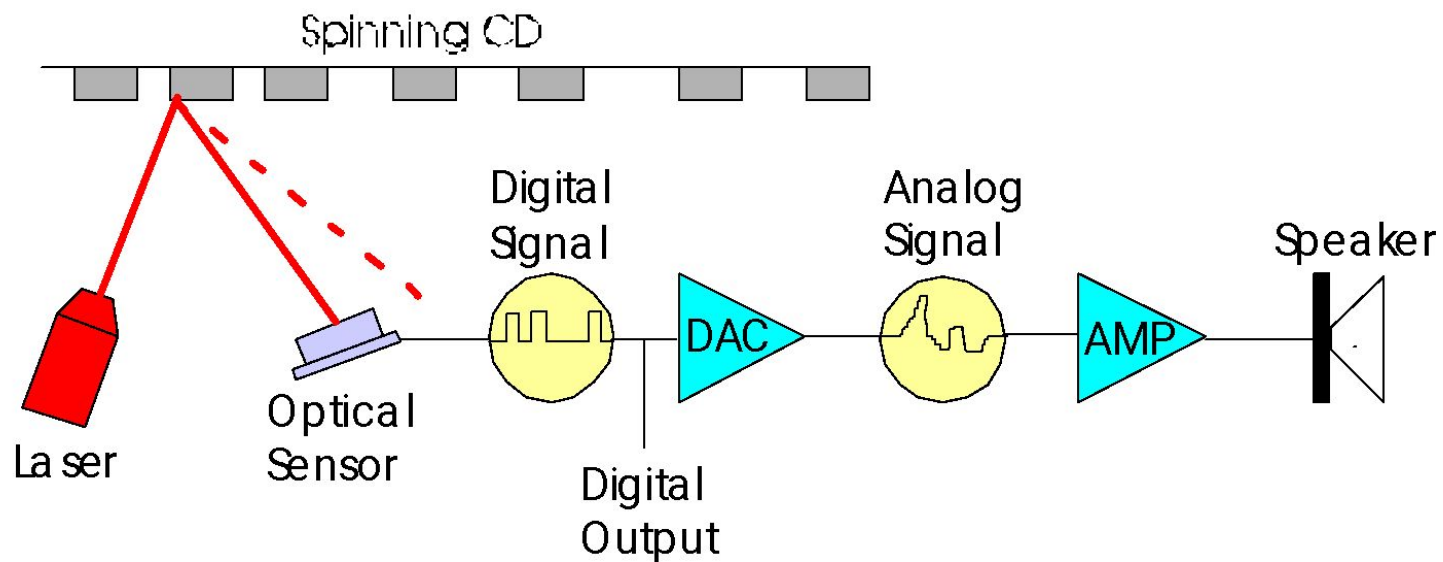


# Inside a Drive



The disk spins faster when the laser is closer to the center of the disk and slows down when the laser travels toward the outer edge.

# What's Going On At The Bit Level



# DVD

- Called Digital Versatile Disk originally
- Today commonly called digital video disk
  - Primary use is for video media
- Like CDs, comes in DVD-R and DVD-R/W formats
  - And also comes in single and dual layer formats
- Same size as CD but can store more data due to narrower and shorter pits and a smaller pitch to pack data into the media more tightly

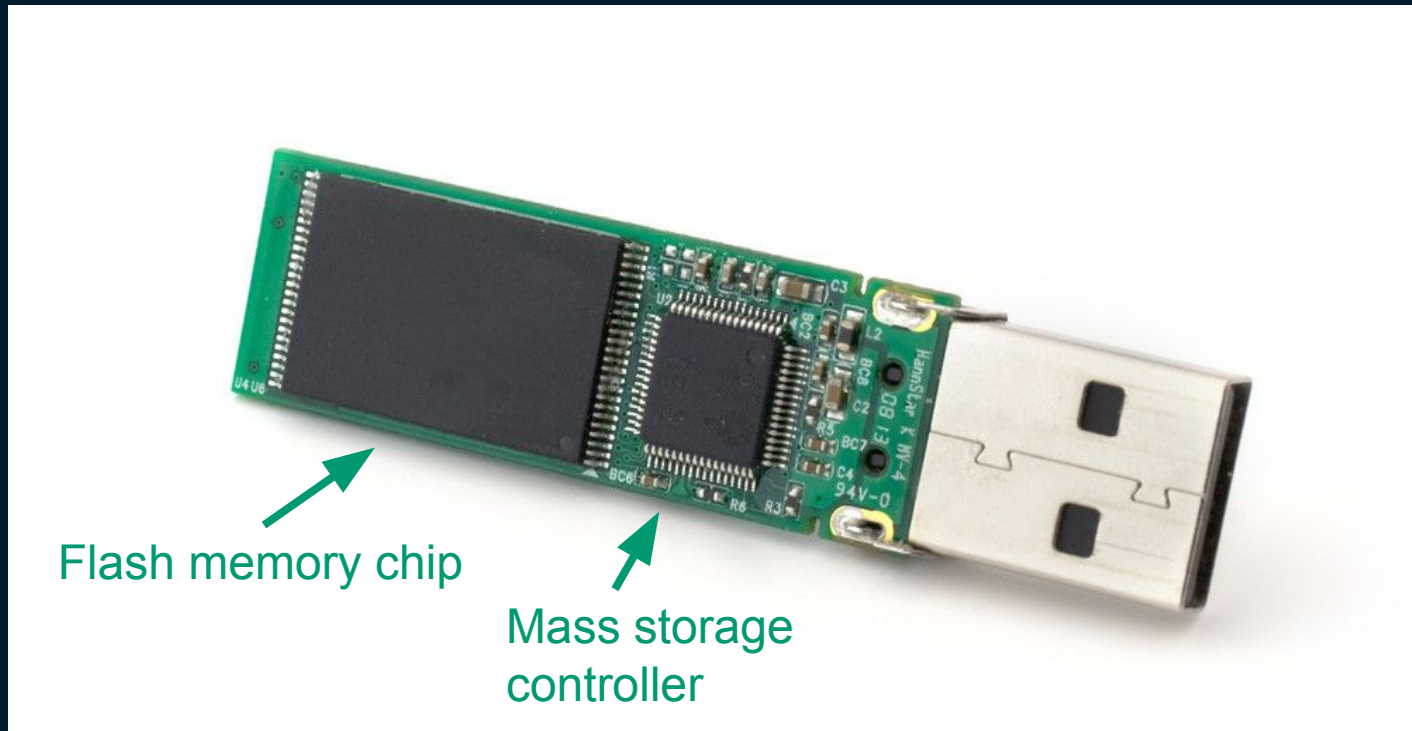
# DVD

- Blu-ray Disc uses a 405 nm "blue" laser diode.
  - Although the laser is called "blue", its color is actually in the violet range.
- The shorter wavelength can be focused to a smaller area, thus enabling it to read information recorded in pits that are less than half the size of those on a DVD
- Blu-ray Discs hold about five times the amount of information that can be stored on a DVD.

# Flash Memory

- Used mostly for I/O
  - Has replaced floppy diskette on most systems
  - Usually employs USB or Firewire interface
  - MicroSD is preferred for A/V
- It is a non-volatile integrated circuit memory
  - Special type of EEPROM (Electrically Erasable Programmable Read-Only Memory)
  - Erased and programmed using large data blocks
- Slower than DRAM
  - Storage cells are floating-gate transistors
  - Cells are initialized to 1
  - Data written by forcing cell to binary 0
  - Limited life (about 10,000 – 100,000 writes)

# Inside Flash Memory Drive



Storage and controller chip on circuit board

# Solid State Drives

- Data storage device that uses solid-state memory to store persistent data
- SSD terminology has been adopted to distinguish solid-state electronics from electromechanical devices (traditional hard drives)
- Have hard disk form factor and SATA interface
- No moving parts
  - Solid-state drives are less fragile than hard disks
  - Silent (unless a cooling fan is used)
  - As there are no mechanical delays, they usually have low access time and latency
- Improving capacity and cost
  - But still 3 – 5 times more expensive than HDD
  - Largest SSD today is 8 TB

# Solid State Drives (cont)

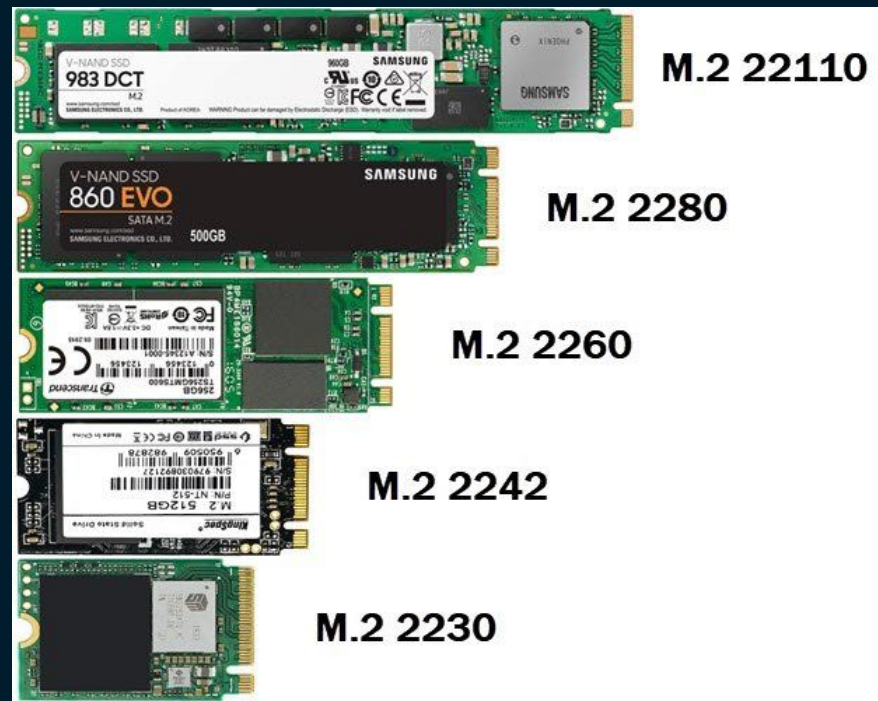
## Internal



3.5" and 2.5"  
Form Factors



## Various M.2 HD Configurations





# SSD Controller

- Electronics that interface the SSD memory components to the host computer
  - Controller is an embedded processor that executes firmware-level code
  - is one of the most important factors of SSD performance
- Functions performed
  - Error-correcting code (ECC)
  - Wear leveling
  - Bad block mapping
  - Read scrubbing and read disturb management
    - Correcting bit errors and relocating adjacent memory after read limit
  - Read and write caching
  - Garbage collection
  - Encryption

# Buses

- Buses have traditionally been the means of connecting various components of a system
  - CPU to memory
  - Memory to I/O
- Advantage has always been versatility and low cost
- Disadvantage has been the bottleneck which limits throughput
  - Maximum speed is determined by length and number of devices
  - Supporting devices with various latencies and transfer rates

# Bus Characteristics

- Bus lines (set of wires or connectors)
  - Control lines to handle inter-device communication
  - Data lines to pass data, addresses between devices
- Types of buses
  - Processor-memory: short, high-speed, matched to the memory system to maximize data transfer between memory and CPU (aka front-side bus)
  - I/O: parallel buses that interconnect I/O devices to system through interface circuitry; capable of supporting various devices with different I/O characteristics

# Bus Types

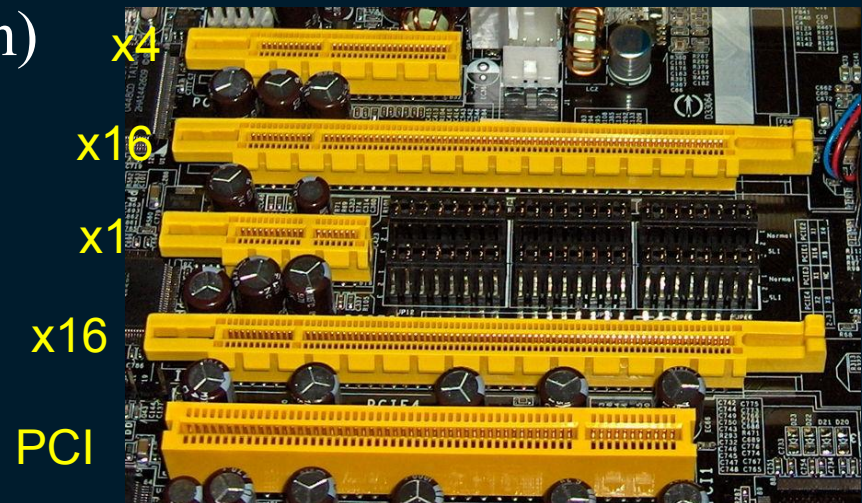
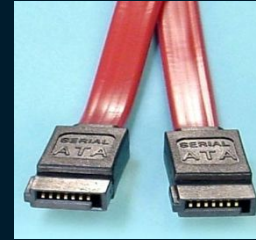
- Parallel
  - One “wire” for each bit of a data item
  - Bus is multiple conductors – transmission is multiple bits at a time
  - Usually controlled by a clock for synchronization
  - Skewing is a problem
  - Usually uses a synchronous communication protocol
- Serial
  - One “wire” for all bits
  - Bus is single conductor – transmission is one bit at a time
  - Usually uses asynchronous communication
  - No skewing issues
  - Can be synchronous or asynchronous

# Bus Communication

- Synchronous
  - Uses clock for timing
  - Uses a fixed communication protocol based on clock cycles
    - Strict timing for data transfers
  - Run very fast and have small interface logic
  - Disadvantages:
    - Bus length can result in clock skew
    - All devices must operate at the same clock speed
- Asynchronous
  - No clock (usually); less stringent timing constraints
  - Can support devices of varying speeds
  - Communication and coordination is accomplished through a handshaking protocol
    - Handshaking is a series of steps where the sender and receiver proceed to the next step only if both agree

# Bus Standards

- SATA (Serial ATA)
- PATA (Parallel ATA)
- PCI (Peripheral Component Interconnect)
- PCI Express
- SCSI (Small Computer System Interface)
- Firewire (IEEE-1394) – a.k.a. iLink
- USB (Universal Serial Bus)
- IrDA (Infrared Data Association)
- I<sup>2</sup>C
- eSATA
- FutureBus
- InfiniBand



# Methods for I/O Communication

- Memory-mapped I/O
  - A portion of address space is assigned to I/O devices (usually upper high address range)
  - Referencing any of these addresses are interpreted as commands to a specific I/O device (not used to access RAM)
    - 32-bit systems can only utilize up to 3 GB RAM
  - Memory controller ignores the operation for I/O device
  - Device controller recognizes the command and uses the info to activate the I/O device and perform the operation

# Methods for I/O Communication

- Dedicated I/O instructions
  - Special I/O instructions executed by dedicated I/O processors
  - I/O instructions are offloaded from the CPU to the I/O subsystem
  - I/O instructions consist of a command word and device ID number
  - I/O controller reads its ID and accesses the bus to get address and data for the operation
  - Used in older mainframe systems
  - I/O processors also referred to as channels



# Direct-Memory Access (DMA)

- An I/O feature that allows certain hardware subsystems to access main memory independently of the CPU
  - Without DMA, during I/O, the CPU is using programmed input/output and is fully involved for the entire duration of the I/O, and is unavailable to perform other work.
  - With DMA, the CPU first initiates the transfer, then it does other operations while the transfer is in progress
    - It receives an interrupt from the DMA controller (DMAC) when the operation is done.
  - Many hardware systems use DMA, including disk drive controllers, graphics cards, network cards and sound cards.
    - Is often implemented with both memory-mapped and dedicated I/O communication techniques

# Interrupts

- Interrupt-driven I/O is a technique that allows I/O devices to communicate with the processor
- Very similar to exceptions except
  - I/O interrupts are asynchronous with respect to instruction execution – they don't prevent an instruction from completing execution
  - Interrupts identify the device needing attention and the priority of the device
- Interrupt information is conveyed through vectored interrupts or through a cause register in the CPU status area

# I/O Performance

- Important and complex
  - Bottlenecks abound regarding bandwidth
  - Contributes significantly to overall system performance
- Measurements
  - Performance depends on data transfer rate (GB/sec) which depends on the clock rate given in gigahertz ( $10^9$ ) – base ten numbers are used
  - Data blocks are typically expressed in GiBytes ( $2^{30}$ )
  - The time to transfer 1GiB of data on a 1GB/sec bus is
$$2^{30} / 10^9 = 1.0737 \text{ seconds}$$
  - If you don't convert, you will get some tiny error

# I/O Benchmarks

- Designed to test I/O system based on different access characteristics
- Transaction processing (a type of application that handles small short operations that require both I/O and computation) – emphasis is on response time (I/O rate: I/Os per unit time)
  - TPC benchmarks (Transaction Processing Committee)
- File server/Web I/O may involve larger data transfers where the emphasis is how much data is transferred (data rate: bytes per unit time)
  - SPECFS and SPECWeb benchmarks

# Tape Storage

- Important storage technology for large computing centers
  - Use high-capacity tape cartridges for data back-up
    - One cartridge = 5 Tbytes data (normal); 15 Tbytes (compressed)
  - Often will have an automated storage library
    - Thousands of tapes totaling in the exabytes with hundreds of drives
    - Tapes automatically handled by robotic units
  - HSM (Hierarchical Storage Management)
    - Combines tape and disk in data management (online, nearline, offline storage)

IBM 3592 Tape Cartridge



IBM TS3500 Tape Shuttle Library Complex



# I/O Terminology / Vocabulary

- I/O bottleneck
- Hard disk
  - Density
  - Data transfer rate
  - Platter
  - Sector
  - Track
  - Cylinder
  - Components of disk access
    - Seek
    - Rotational latency
    - Transfer time
    - Controller overhead
- Disk read time calculation
- CD/DVD technology
- Flash memory
- SSD
- Buses
  - Synchronous
  - Asynchronous
- I/O communication
  - Memory mapped
  - Dedicated I/O
  - DMA
  - Interrupt Driven I/O
- Tape storage
  - HSM