Week 8

Secondary Memory, I/O And Bus

Hard disk

I/O: device, controller, data transfer, polling, interrupt

Bus: bus arbitrator, ISA and PCI bus

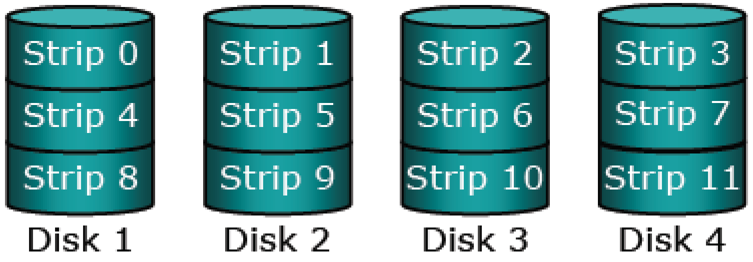
RAID

- Disk performance has not improved much over past few decades
  - – 1970’s, seek time = 50-100 msec
  - – 1990’s, seek time = 10 msec
  - – 2006, seek time = 8.5 msec
  - – 2014, seek time average 4.16 msec, some newer drives down to 2msec
  - • Large gap between CPU and disk performance
- ***RAID***: ***R***edundant ***A***rray of ***I***nexpensive ***D***isks
  - – Use disk layout to improve performance
  - – Distributed data (robust), parallel access
  - – Appear to OS as single disk

RAID con,t

- Inexpensive or Independent
- Parallel I/O
- Six different disk organizations called levels although there is no hierarchy
- Better than ‘SLED’ Single Large Expensive Disk
- The idea is to install a box of disks next to the computer that appear to the OS as a single disk
- No software changes needed
- But you need to replace disk controller with RAID controller
- Often used SCSI disks (or now more commonly SATA)

RAID 0

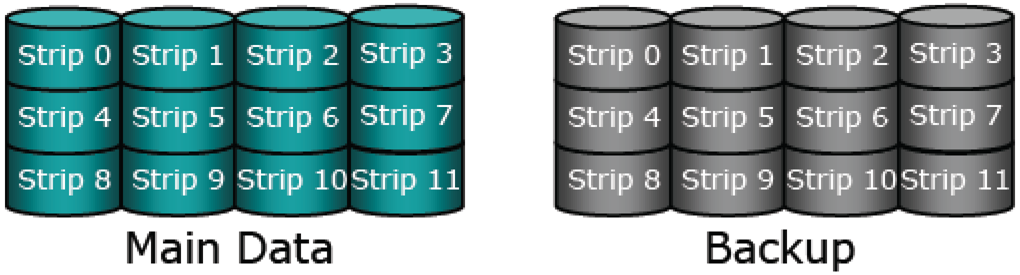


- Each strip is a number of contiguous sectors
- High data-rate
- Large requests can be dealt with in parallel
- Can also handle simultaneous requests
- No redundancy or robustness to disk failure
- So not really a true RAID design
- Poor response for many small sector requests

RAID 0 con’t

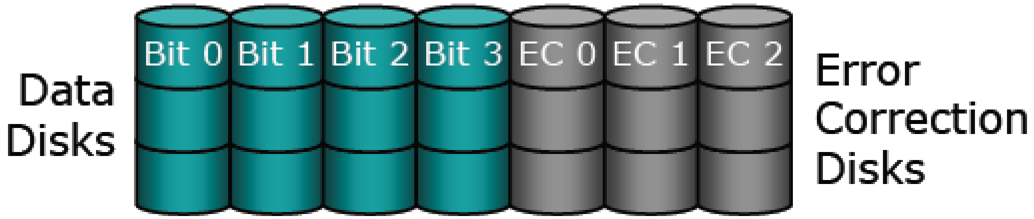
- Data on multiple drives like this is called striping
- If software issues a command to read a block of four consecutive strips the RAID controller will split this into 4 parallel commands, one for each disk.
- Large requests are better then lots of small ones.
- Reliability is worse, if disk failure occurs once in 20,000 hours then a RAID 0 setup will fail once in 5,000 hours with 4 disks

RAID 1



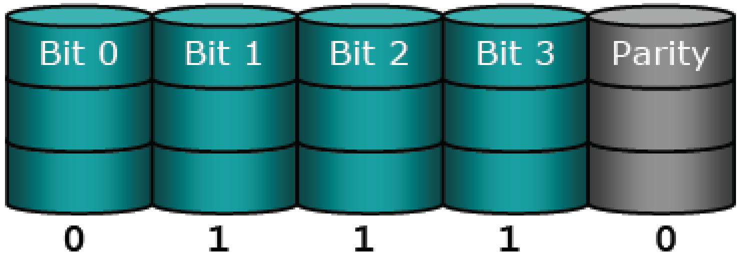
- Duplicate all the disks in the RAID 0 design
- If a drive crashes, can simply replace and copy the backup
- Read performance can be twice as good
- Main disadvantage is the number of disks needed

RAID 2



- Each byte split into two 4-bit segments then add 3 error correction bits to each
- Synchronise the arm and rotational position of the 7 drives
- Fast, as can read/write 1/2 byte each cycle
- Robust, as error-correction allows any drive to fail
- With 32 drives and 6 parity drives, overhead is 19%
- Disadvantage: hard to synchronise drives

RAID 3



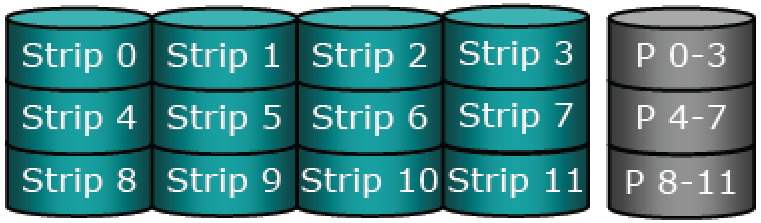
- Simpler version of RAID 2
- Still requires drive synchronisation
- Parity drive provides robustness
- Know which drive went wrong, so we can correct any single error
- Both RAID 2 and 3 have a high data rate
- But cannot handle parallel IO

RAID 3 con't

- Single parity bit is calculated for each data word and written to parity drive
- Might seem that we can only detect errors and not correct
- If a drive crashes we know the position of the bad bit, controller zeros all bits and if the word has a parity error then we know the value of

  the bad bit must be 1
- Raid 2 & 3 have very high data rates the number of I/O requests per second is no better than a single drive.

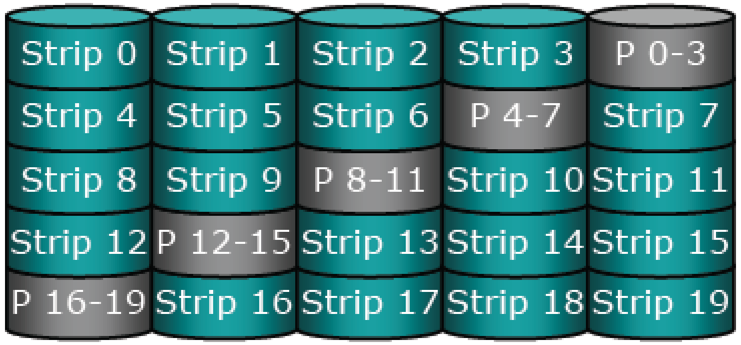
RAID 4



Strip 0	Strip 1	Strip 2	Strip 3	P 0-3
Strip 4	Strip 5	Strip 6	Strip 7	P 4-7
Strip 8	Strip 9	Strip 10	Strip 11	P 8-11

- Like RAID 0, but with strip parity kept on a separate drive
- Does not require drive synchronisation
- Poor performance for small updates
- All drives must be ready to re-compute parity
- Heavy load on parity drive may become a bottle-neck

RAID 5



Strip 0	Strip 1	Strip 2	Strip 3	P 0-3
Strip 4	Strip 5	Strip 6	P 4-7	Strip 7
Strip 8	Strip 9	P 8-11	Strip 10	Strip 11
Strip 12	P 12-15	Strip 13	Strip 14	Strip 15
P 16-19	Strip 16	Strip 17	Strip 18	Strip 19

- Like RAID 4, but distribute parity blocks over all drives
- Can be difficult to reconstruct the contents of a drive if one fails though

SOLID STATE DRIVES (SSD) and USB

- From the users perspective. SSD’s and USB devices have exactly the same structure as mechanical Hard Drives. However, internally they are completely different.
- These devices are usually made from NAND Flash memory chips.
- So internally the memory is arranged in pages.
- NAND is nowhere near as fast as main memory, but it’s multiple orders of magnitude faster than a hard drive.
- While write latencies are significantly slower for NAND flash than read latencies, they still outstrip traditional spinning media.

How SSD memory works.

- Electrons are stored in the floating gate, which then reads as charged “0” or not-charged “1.”
- **Yes**! in NAND flash, a 0 means that data is stored in a cell — it’s the opposite of how we typically think of a zero or one.
- NAND flash is organized in a grid. The entire grid layout is referred to as a block, while the individual rows that make up the grid are called a page.
- Common page sizes are 2K, 4K, 8K, or 16K, with 128 to 256 pages per block.
- Block size therefore typically varies between 256KB and 4MB.

How SSD memory works.

- SSD storage retains data by trapping electrons inside nanoscale memory cells.
- A process called tunneling is used to move electrons in and out of the cells, but the back-and-forth traffic erodes the physical structure of the cell, leading to breaches that can render it useless.
- Electrons also get stuck in the cell wall, where their associated negative charges complicate the process of reading and writing data.
- This accumulation of stray electrons eventually compromises the cell's ability to retain data reliably—and to access it quickly.
- Most commercially available flash products are guaranteed to withstand around 100,000 P/E (Program/Erase) cycles before the wear begins to deteriorate the integrity of the storage.

READ/WRITE and ERASE

- One of the functional limitations of SSD's is that while they can read and write data very quickly *to an empty drive*, overwriting data is much slower.
- This is because while SSDs read data at the page level (meaning from individual rows within the NAND memory grid) and can write at the page level, assuming that surrounding cells are empty, they can only erase data at the block level.
- This is because the act of erasing NAND flash requires a high amount of voltage.
- While you can theoretically erase NAND at the page level, the amount of voltage required stresses the individual cells around the cells that are being re-written. Erasing data at the block level helps mitigate this problem.

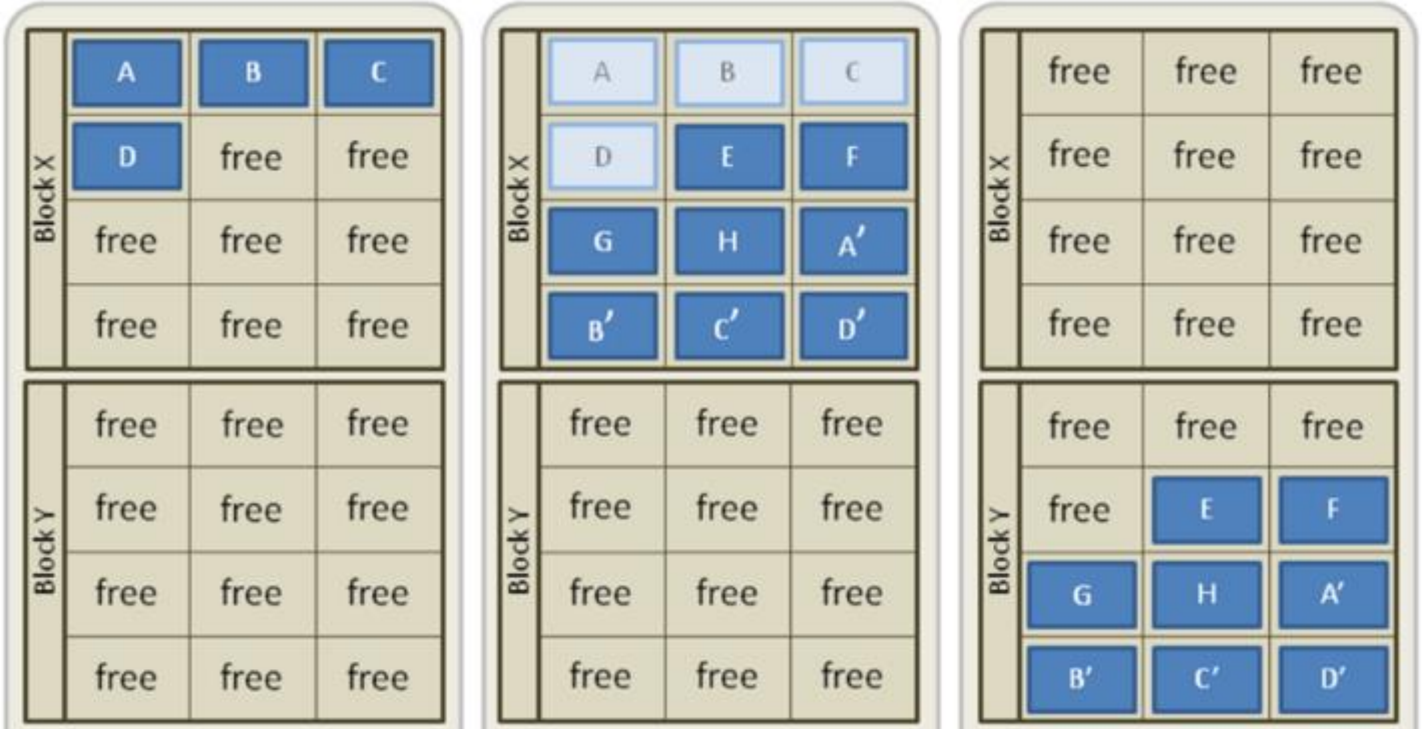
READ/WRITE and ERASE con’t

- The only way for an SSD to update an existing page is to copy the contents of the entire block into memory, erase the block, and then write the contents of the old block + the updated page.
- If the drive is full and there are no empty pages available, the SSD must first scan for blocks that are marked for deletion but that haven’t been deleted yet, erase them, and then write the data to the now-erased page.
- This is why SSDs can become slower as they age — a mostly-empty drive is full of blocks that can be written immediately, a mostly-full drive is more likely to be forced through the entire program/erase sequence.

GARBAGE collection

- SSDs, have a process called “garbage collection.”
- Garbage collection is a background process that allows a drive to mitigate the performance impact of the program/erase cycle by performing certain tasks in the background. The following image on the next slide steps through the garbage collection process.

Garbage collection



1. Four pages (A-D) are written to a block (X). Individual pages can be written at any time if they are currently free (erased).

2. Four new pages (E-H) and four replacement pages (A′-D′) are written to the block (X). The original A-D pages are now invalid (stale) data, but cannot be overwritten until the whole block is erased.

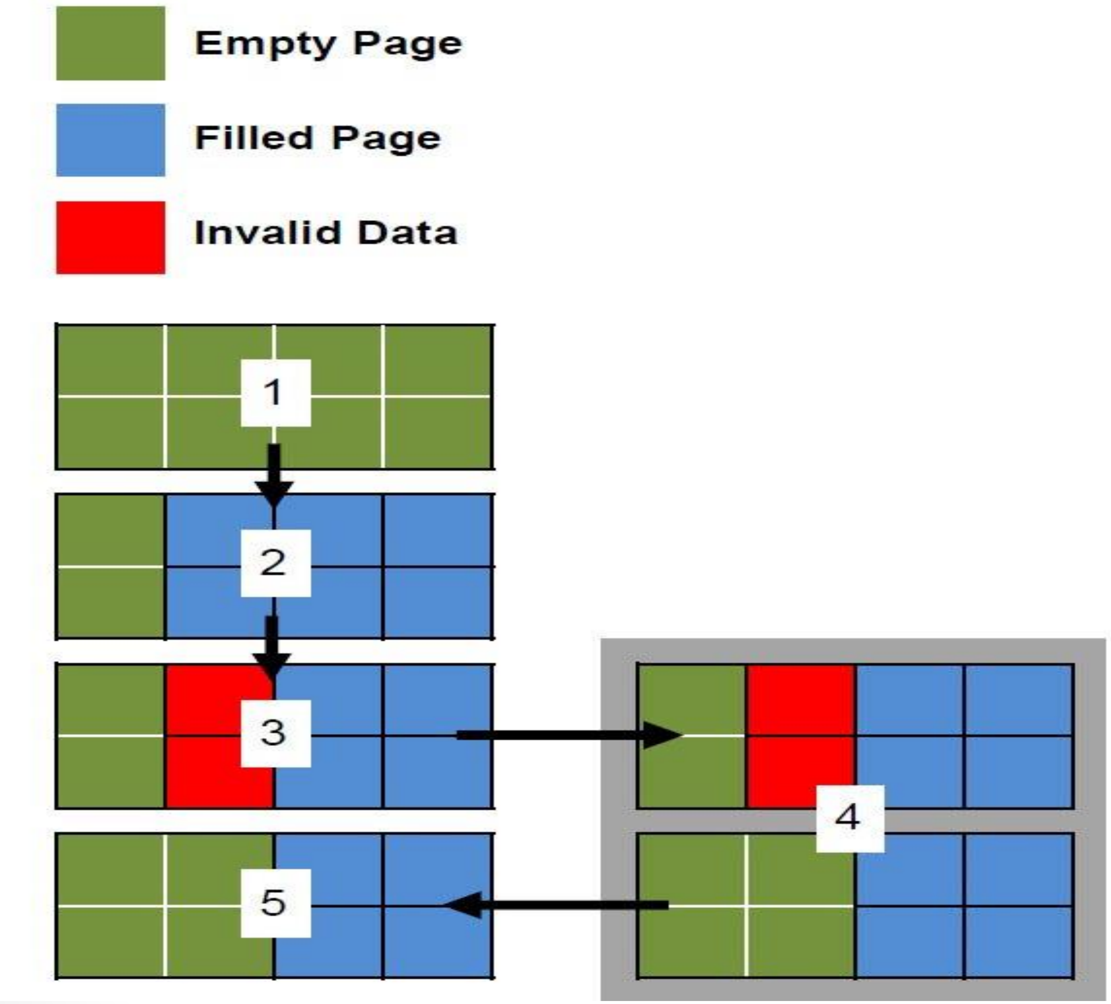
3. In order to write to the pages with stale data (A-D) all good pages (E-H & A′-D′) are read and written to a new block (Y) then the old block (X) is erased. This last step is *garbage collection*.

Garbage Collection con’t

- Note that in the example, the drive has taken advantage of the fact that it can write very quickly to empty pages by writing new values for the first four blocks (A’-D’).
- It’s also written two new blocks, E and H.
- Blocks A-D are now marked as stale, meaning they contain information that the drive has marked as out-of-date.
- During an idle period, the SSD will move the fresh pages over to a new block, erase the old block, and mark it as free space.
- This means that the next time the SSD needs to perform a write, it can write directly to the now-empty Block X, rather than performing the program/erase cycle.

TRIM

- When you delete a file from Windows on a typical hard drive, the file isn’t deleted immediately. Instead, the operating system tells the hard drive that it can overwrite the physical area of the disk where that data was stored the next time it needs to perform a write. This is why it’s possible to undelete files (and why deleting files in Windows doesn’t typically clear much physical disk space until you empty the recycling bin). With a traditional HDD, the OS doesn’t need to pay attention to where data is being written or what the relative state of the blocks or pages is. With an SSD, this matters.
- The TRIM command allows the operating system to tell the SSD that it can skip rewriting certain data the next time it performs a block erase. This lowers the total amount of data that the drive writes and increases SSD longevity.
- Both reads and writes damage NAND flash, but writes do far more damage than reads.



1.) SSD pages contain no data

2.) User writes data to SSD pages

3.) User deletes some data. Pages are marked as 'not in use' by the host OS, but data remains on SSD.

4.) TRIM command tells SSD controller that pages contain invalid data. Pages with invalid data are cleaned.

5.) Data is written back to SSD memory cells. The invalid data has been cleaned and data is able to be written to the pages at full speed.

WRITE AMPLIFICATION

- Because SSDs write data to pages but erase data in blocks, the amount of data being written to the drive is always larger than the actual update.
- If you make a change to a 4KB file, for example, the entire block that 4K file sits within must be updated and rewritten.
- Depending on the number of pages per block and the size of the pages, you might end up writing 4MB worth of data to update a 4KB file.
- Garbage collection reduces the impact of write amplification, as does the TRIM command. Keeping a significant chunk of the drive free and/or manufacturer overprovisioning can also reduce the impact of write amplification.

Wear Levelling

- Wear leveling refers to the practice of ensuring that certain NAND blocks aren’t written and erased more often than others.
- While wear leveling increases a drive’s life expectancy and endurance by writing to the NAND equally, it can actually increase write amplification.
- In other to distribute writes evenly across the disk, it’s sometimes necessary to program and erase blocks even though their contents haven’t actually changed.
- A good wear leveling algorithm seeks to balance these impacts.

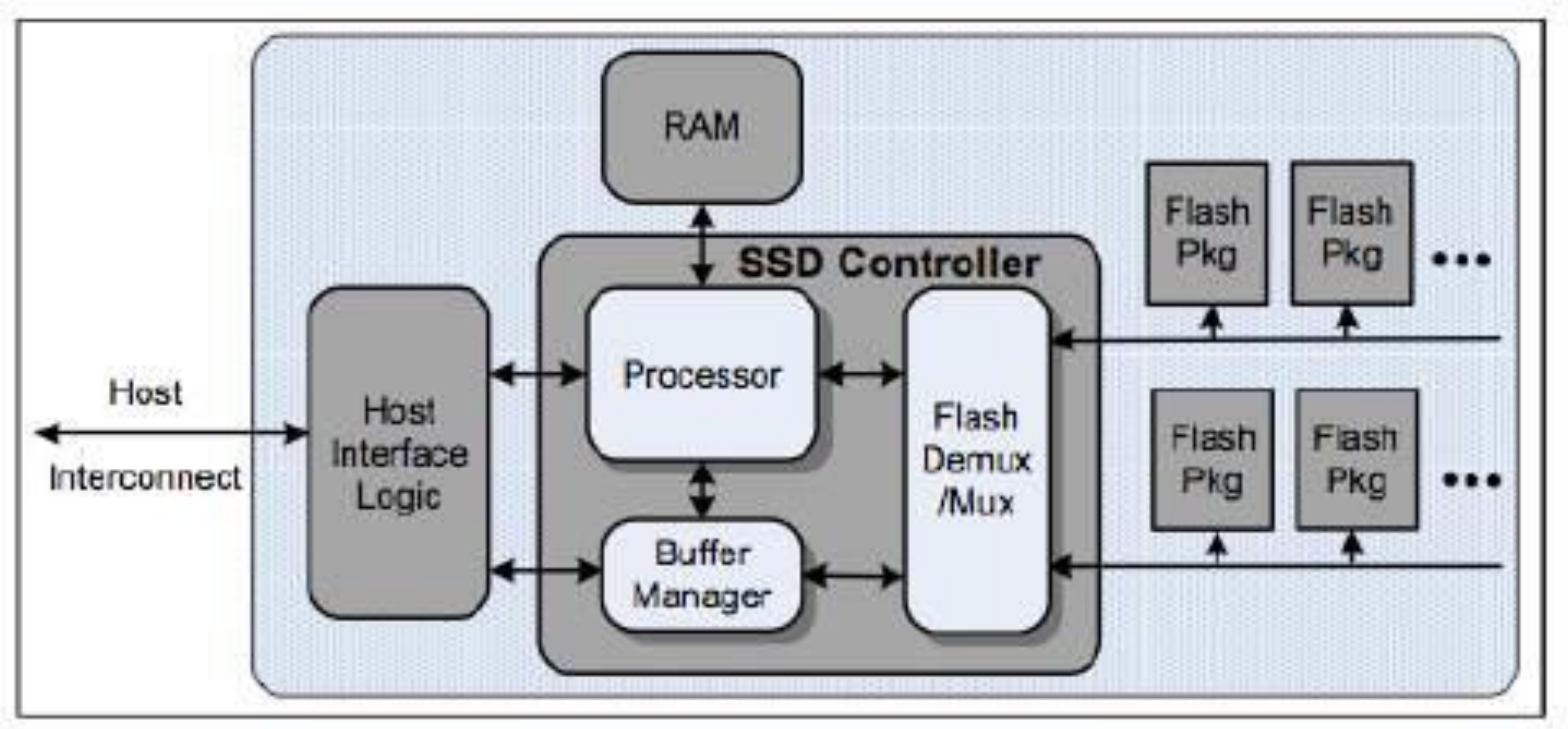
Wear Levelling con’t

- With Flash memory the FTL (File Translation Layer) keeps track of the writes to the drive and designates the next area that should be written to based on the amount of writes that have already occurred in each section.
- This is important to know since before any information can be written to an available section it must first be cleansed.

SSD Controllers

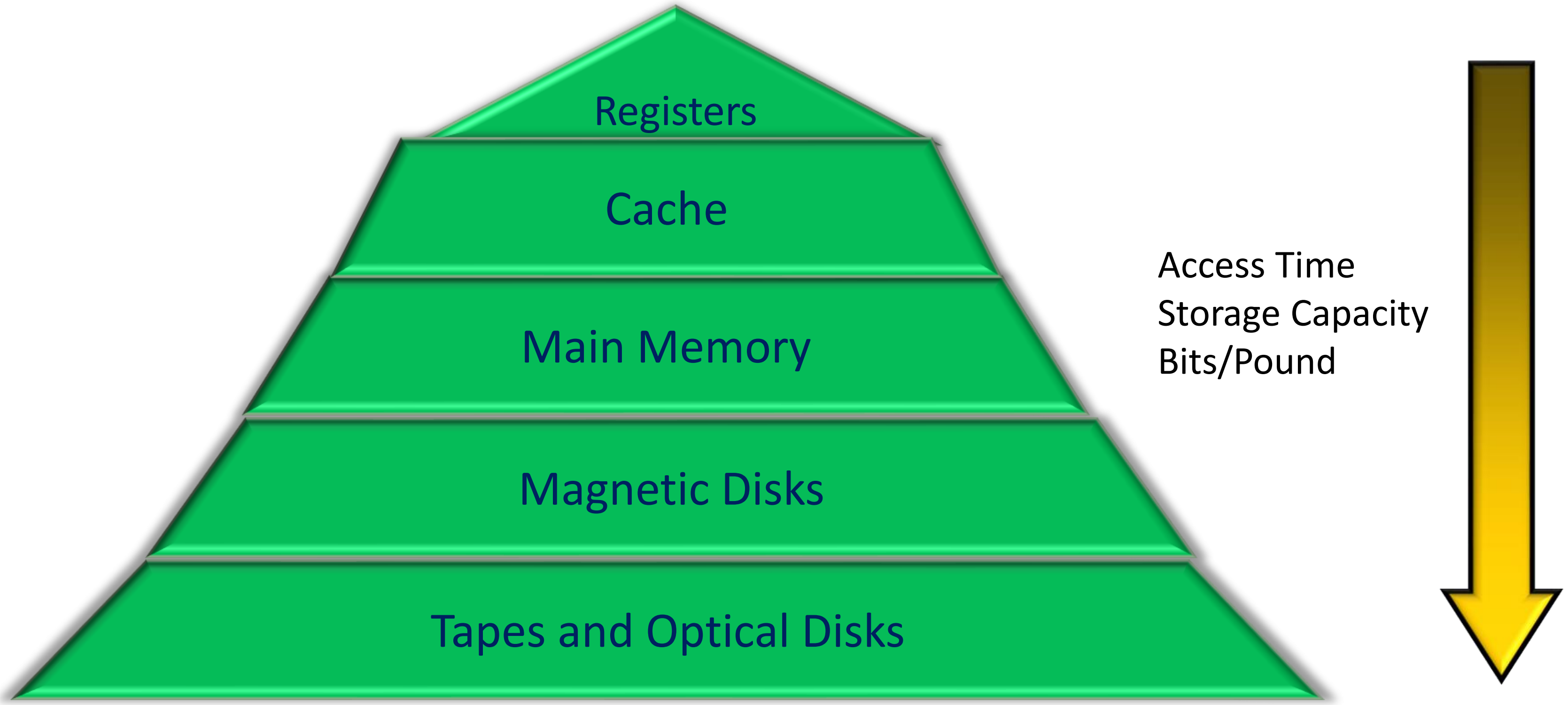
- SSDs require much more sophisticated control mechanisms than hard drives do. That's not to diss magnetic media — HDDs deserve respect. The mechanical challenges involved in balancing multiple read-write heads nanometers above platters that spin at 5,400 to 10,000 RPM are amazing.
- The fact that HDDs perform this challenge while pioneering new methods of recording to magnetic media and eventually wind up selling drives at 3-5 cents per gigabyte is simply incredible.
- SSD *controllers*, however, are in a class by themselves. They often have a DDR3 memory pool to help with managing the NAND itself.
- Many drives also incorporate single-level cell caches that act as buffers, increasing drive performance by dedicating fast NAND to read/write cycles.
- Because the NAND flash in an SSD is typically connected to the controller through a series of parallel memory channels, you can think of the drive controller as performing some of the same load balancing work as a high-end storage array — SSDs don't deploy RAID internally, but wear leveling, garbage collection, and SLC cache management all have parallels in the big iron world.

Typical SSD Structure



SSD Logic Components

Different types of SSD



The Memory Hierarchy

CD ROM and DVD

- Data encoded as changes in reflectivity Pits and Land
- Much slower than magnetic disks
  - Seek times usually 100-200 msec
- Uses constant *linear* velocity, rather than constant angular velocity
- Speed of player changes depending on position: 530 rpm at inside, 200 rpm at outside
- CD-ROM can hold around 650 – 700 Mb
- DVD has smaller pits (0.4 v.s 0.8 microns) and a tighter spiral (0.74 v.s 1.6 microns)
- Can hold 4.7 to 17 Gb, depending on type
- Only 29% of a CD is data – the rest is error-correcting code and the file-system

History

- Compact Disc - Digital Audio *(CD-DA),* the original CD specification developed by Philips and Sony in 1980
- Specifications were published in *Red Book,* continued to be updated (lastest version in 1999)
- In 1985 a standard for the storage of computer data by Sony and Philips, CD-ROM (Compact Disc Read Only Memory)
- developments in the technology have been ongoing and rapid
  - Compact Disc Interactive (CD-I)
  - Compact Disc Television (CD-TV)
  - Compact Disc Recording (CD-R)
  - Digital Video Disc (DVD)

Structure

- A CDROM Drive uses a small plastic-encapsulated disk that can store data
- This information is retrieved using a Laser Beam
- A CD can store vast amounts of information because it uses light to record data in a tightly packed form

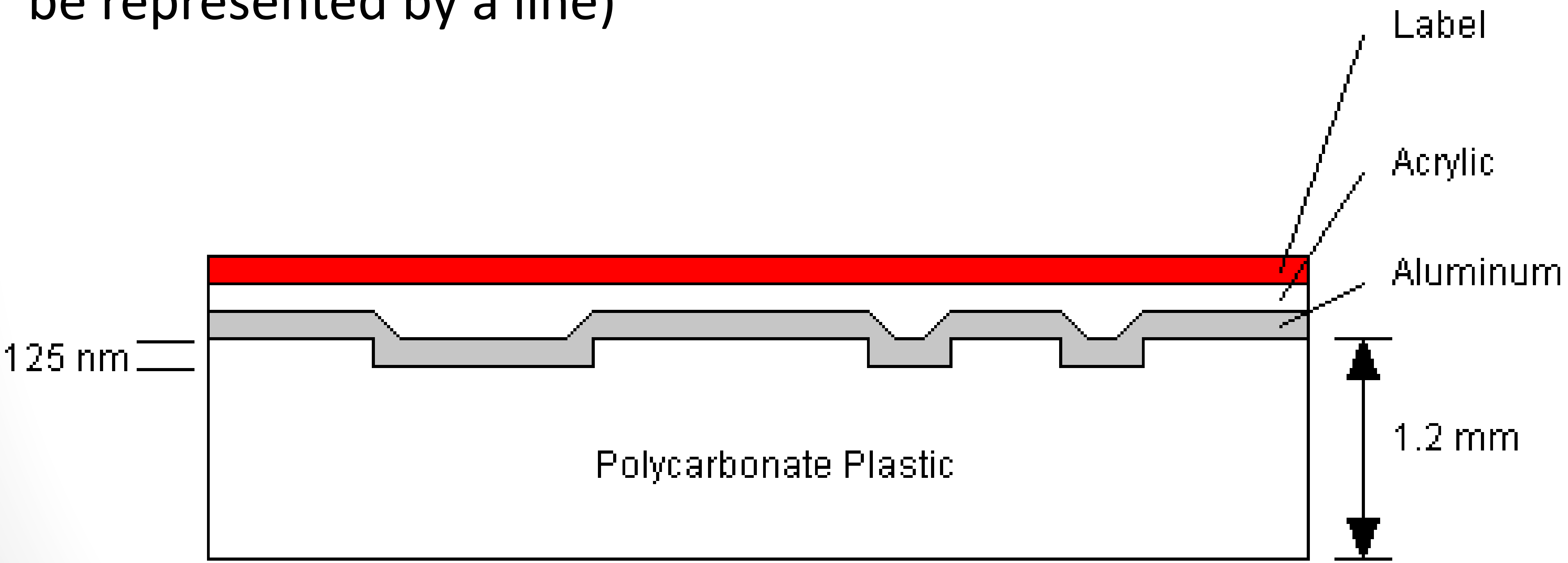
CD Layers

- The *thickness* of a CD can vary between 1.1 and 1.5mm
- A CD consists of four layers
  - The biggest part is clear polycarbonate (nominally 1.2mm)
  - There is a very thin layer of reflective metal (usually aluminum) on top of the polycarbonate
  - Then a thin layer of some protective material covering the reflective metal
  - A label or some screened lettering on top of protective material

CD Layers (cont'd)

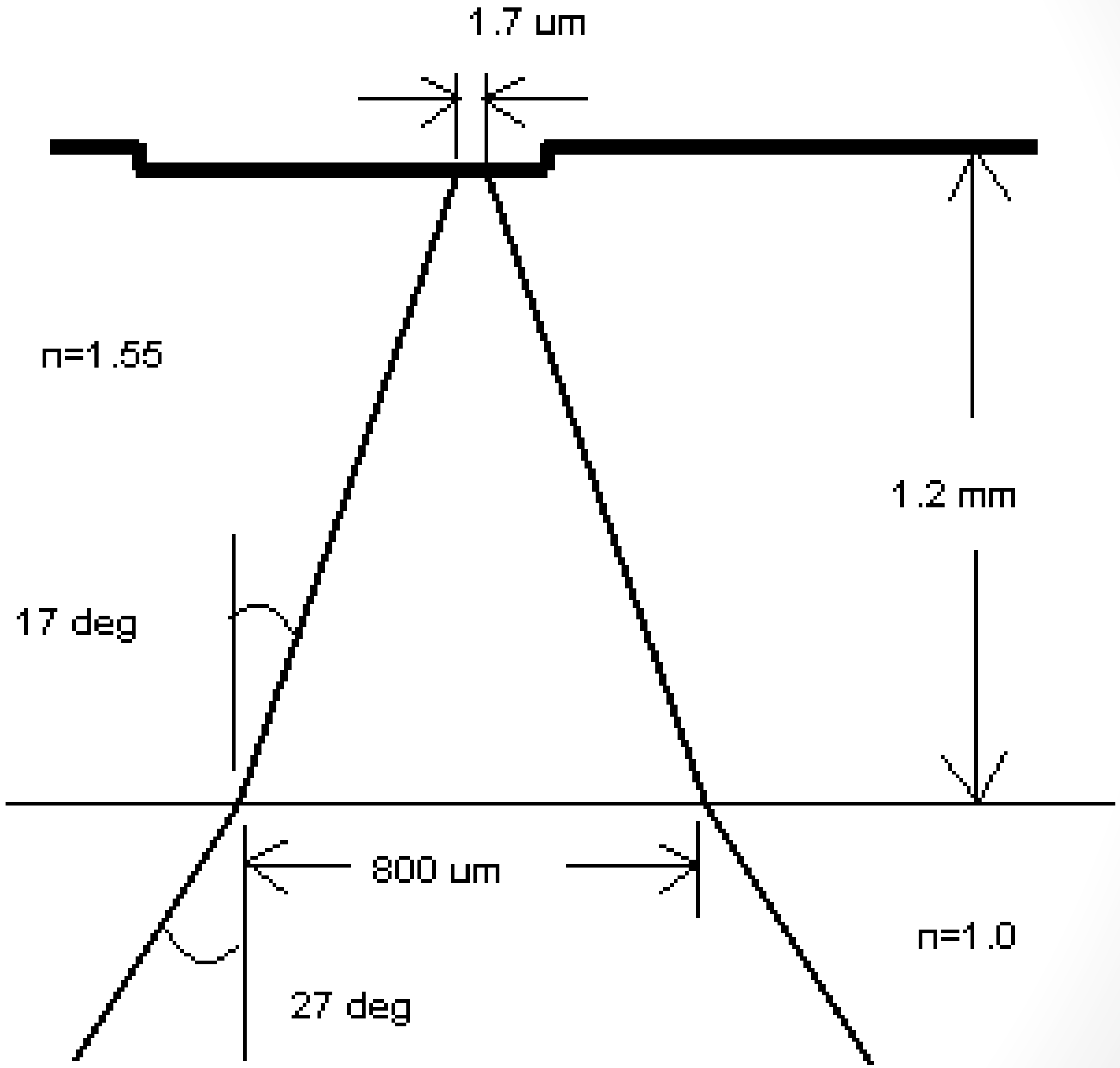
Different layers of a CD :

(though the reflective metal layer is really so thin that it should just be represented by a line)



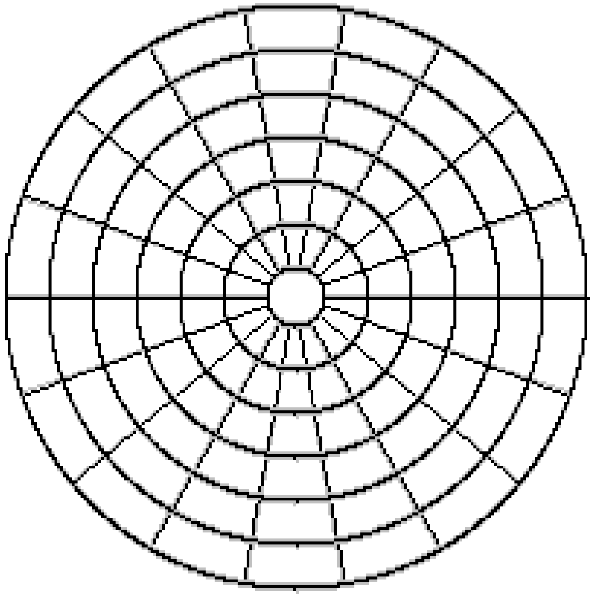
CD Safety

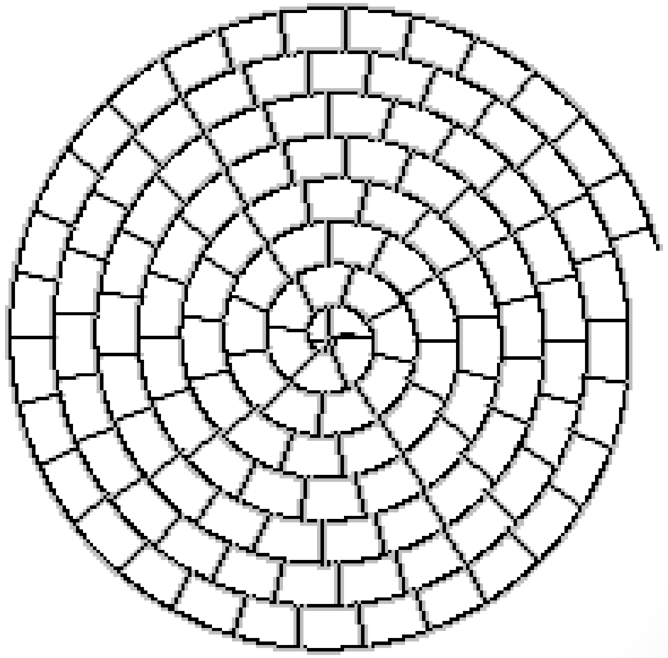
- The label side of a CD is the most vulnerable part of the disk.
- the other side is protected by the thick (1.2mm) and hard polycarbonate.
- It is possible to carefully clean and even to polish this surface to remove fingerprints and even scratches.
- Many flaws on the polycarbonate surface will simply go unnoticed.



CD vs. Magnetic Media

- In Magnetic Media (like floppy/hard disk) the surface is arranged into concentric circles called “tracks”.
- Number of sectors per track is constant for all tracks.
- the CD has one single track, starts at the center of the disk and spirals out to the circumference of the disk.
- This track is divided into sectors of equal size.





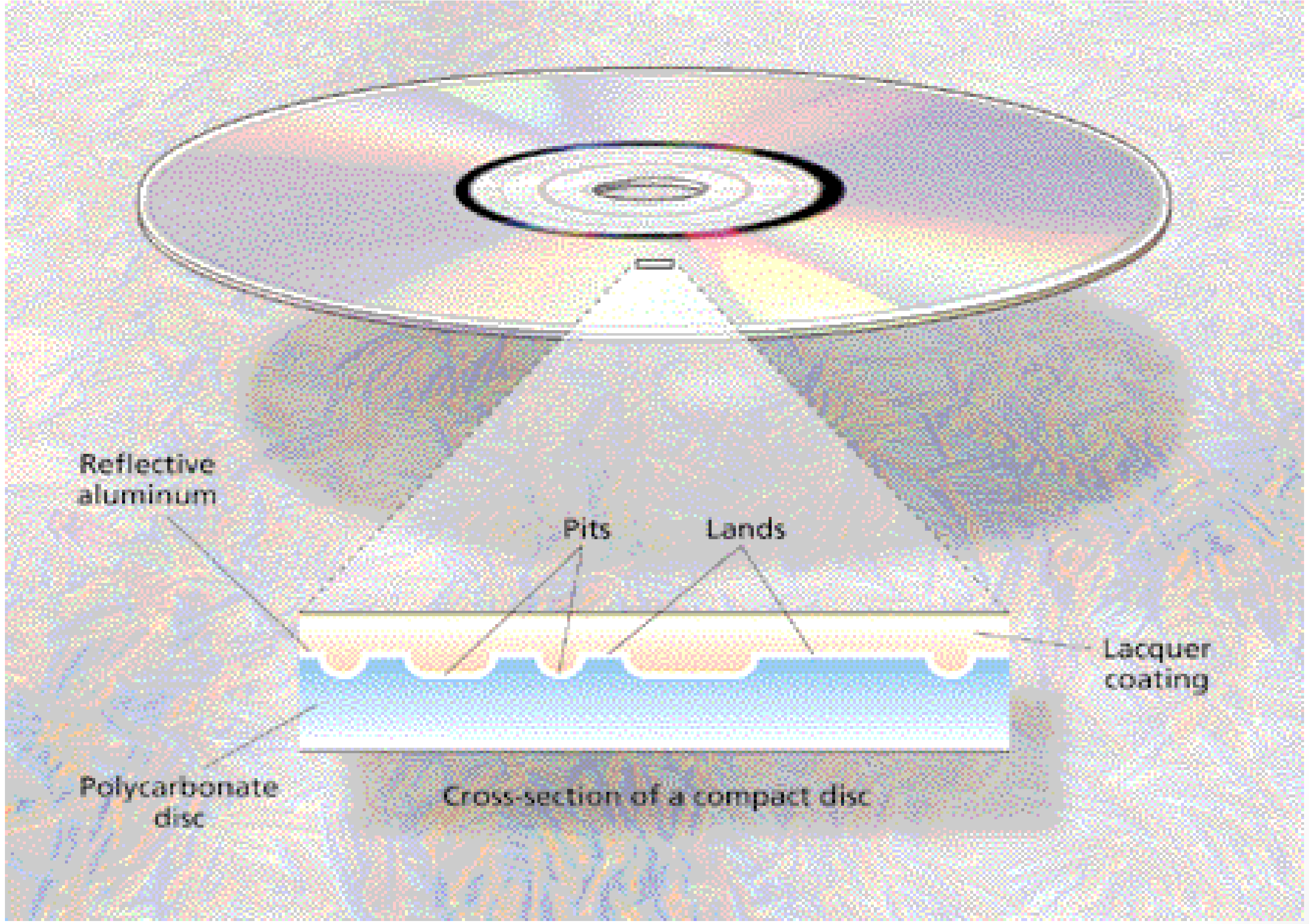
CD Data Recording

- Information is recorded on a CD using a series of bumps
- These bumps are called “pits” because they are looking like pits in the polycarbonate layer
- The disk is read from the bottom, through the transparent polycarbonate (the pits appear as bumps to the scanning laser)

Data Recording (cont'd)

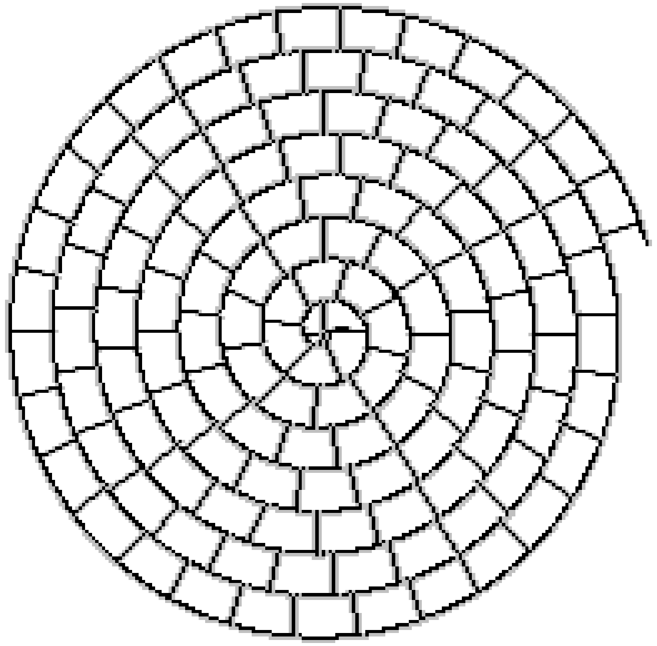
- The unmarked areas between pits are called "lands"
- Lands are flat surface areas

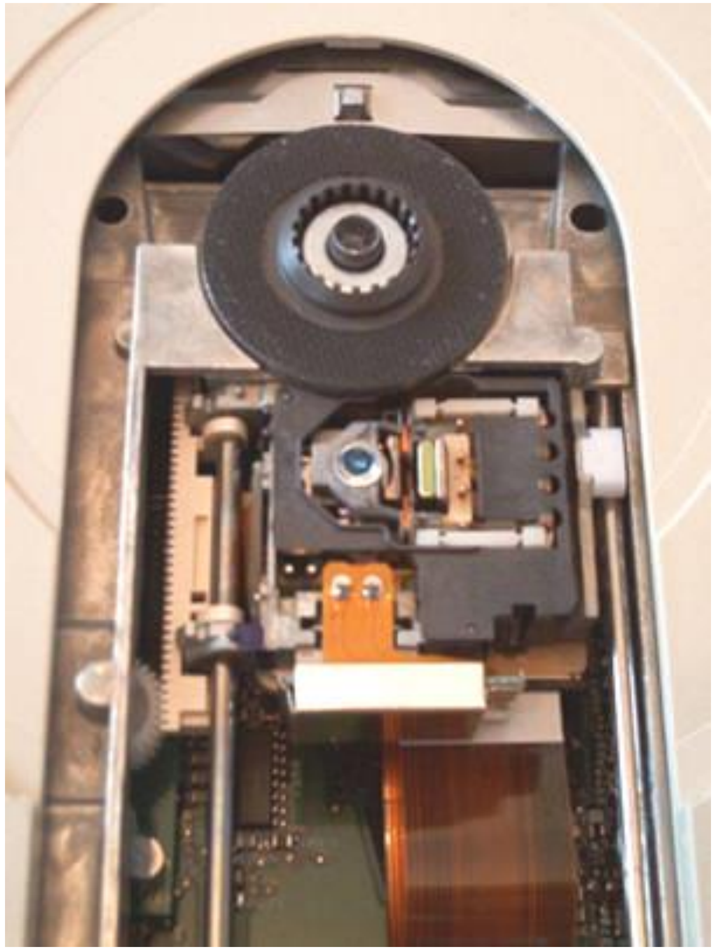
- The information is stored permanently as pits and lands on the CD-ROM. It cannot be changed once the CD-ROM is mastered, this is why its called CD-ROM



How The CD Drive Works

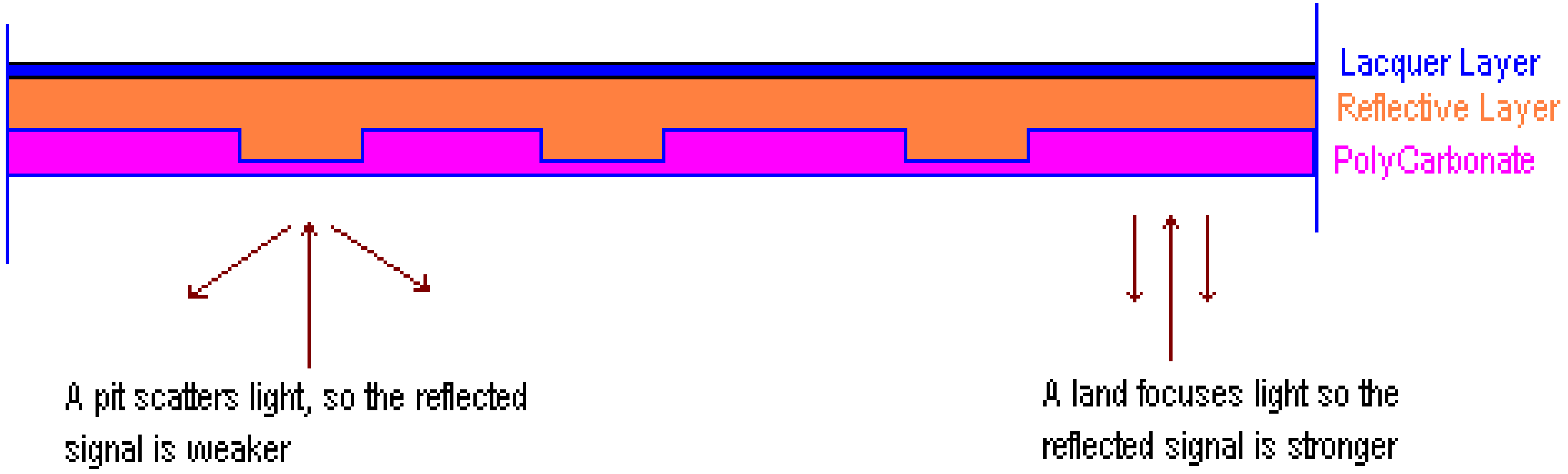
- A motor rotates the CD
- the rotational speed varies so as to maintain a Constant Linear Velocity (the disk is rotated faster when its inner "SPIRALS" are being read)



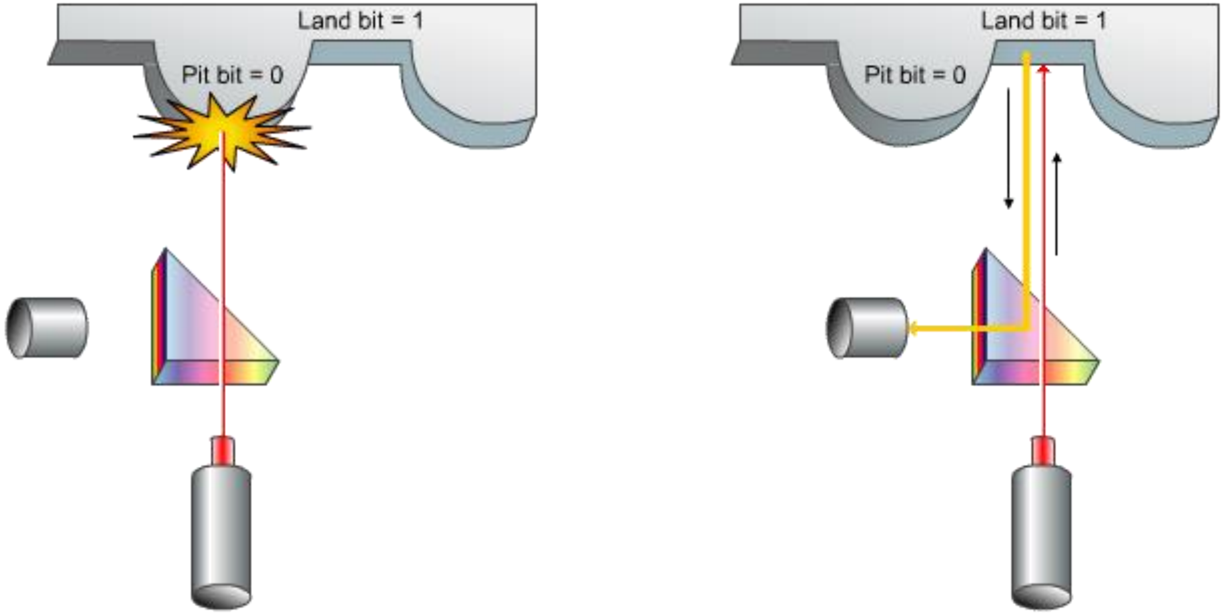


How The CD Drive Works (cont'd)

- A laser beam is shone onto the surface of the disk
- The light is scattered by the pits and reflected by the lands, these two variations encode the binary 0's and 1's
- A light sensitive diode picks up the reflected laser light and converts the light to digital data

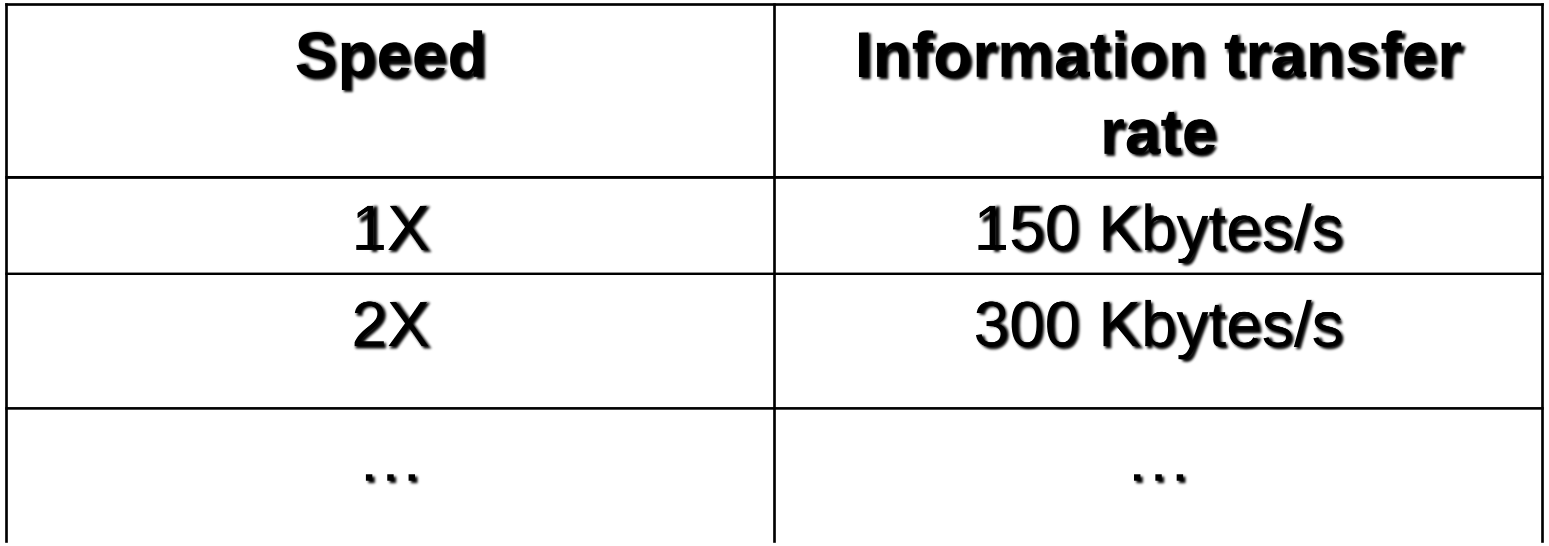

Lacquer Layer
Reflective Layer
PolyCarbonate
A pit scatters light, so the reflected signal is weaker
A land focuses light so the reflected signal is stronger

How The CD Drive Works (cont’d)



CD-ROM Drive Speed

- The CD-ROM drives are classified by their **rotational speed**
- Based on the original speed of a CD-Audio (e.g. A "2X" CD-ROM drive will run at twice the speed of a CD- Audio)



Speed	**Information transfer rate**
1X	150 Kbytes/s
2X	300 Kbytes/s
…	…

CD Physical Specifications

Diameter	120mm ±0.3mm
Transparent Layer Thickness	1.2mm ±0.1mm
Total Thickness	1.1mm - 1.5mm
Transparent Layer Index of Refraction	1.55 ±0.10
Reflectance of Metal Layer through Transparent Layer	70% minimum
Laser Wavelength	780nm ±10nm
Track Pitch	1.6 micron ±0.1 micron
Scanning Linear Velocity	1.20m/s - 1.40m/s (±0.01m/s)

CD File Systems

1. **ISO-9660**

   The base standard defines three levels of compliance

- Level 1 limits file names to 8+3 format. Many special characters (space, hyphen, equals, and plus) are forbidden
- Level 2 and 3 allow longer filenames (up to 31) and deeper directory structures (32 levels instead of 8)
- Level 2 and 3 are not usable on some systems, notably MS-DOS

CD File Systems (cont’d)

2. **Rock Ridge**

- Extensions to ISO-9660 file system
- Favored in the Unix world
- Lifts file name restrictions, but also allows Unix-style permissions and special files to be stored on the CD
- Machines that don't support Rock Ridge can still read the files because it's still an ISO-9660 file system (they won't see the long forms of the names)
- UNIX systems and the Mac support Rock Ridge
- DOS and Windows currently don't support it

CD File Systems (cont’d)

3. **Joliet**

- Favored in the MS Windows world
- Allows Unicode characters to be used for all text fields (including file names and the volume name)
- Disk is readable as ISO-9660, but shows the long filenames under MS Windows

4. **HFS (Hierarchical File System)**

   Used by the Macintosh in place of the ISO-9660, making the disk unusable on systems that don't support HFS

Multiple Sessions

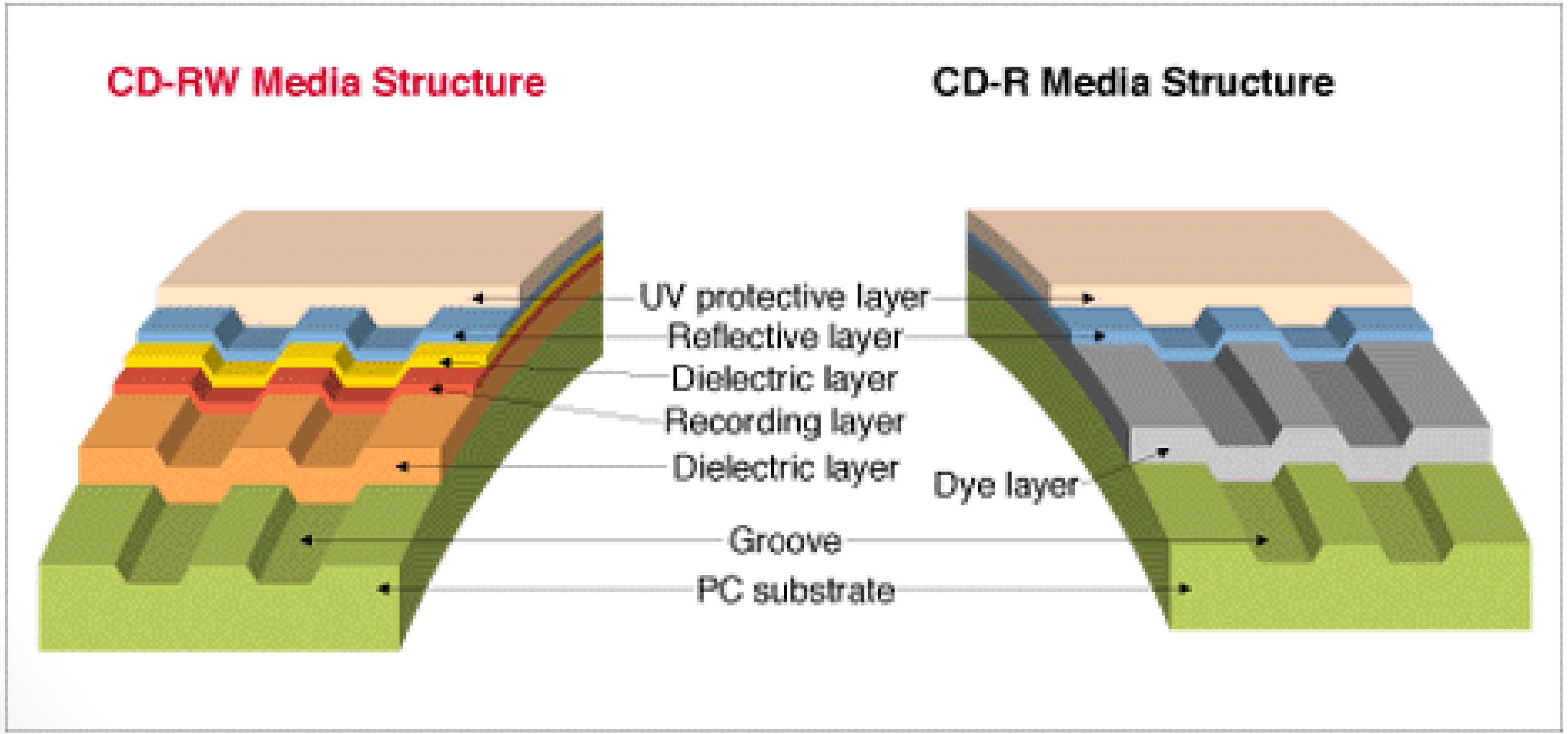
- Allows CDs to be written more than once (not re-written)
- Some CD writers support this feature
- About 640MB of data can be written to the CD, as some space is reserved for timing and other information
- Each session written has an overhead of approximately 20MB per session

CD-ReWritable (CD-RW)

- It is essentially CD-R
- Allows discs to be written and re-written up to 1000 times
- The storage capacity is the same as that for CD-R
- Based on phase-change technology
- The recording layer is a mixture of silver, indium, antimony and tellurium

CD-RW Recording Process

- The recording layer is polycrystalline
- The laser heats selected areas of the recording track to the recording layer's melting point of 500 to 700 degrees Celsius



CD-RW Recording (cont'd)

- The laser beam melts the crystals and makes them non-crystalline (amorphous phase)
- The medium quickly cools, locking in the properties of the heated areas
- The amorphous areas have a lower reflectivity than the crystalline areas
- This creates a pattern which can be read as pits and lands of the traditional CD
- To erase a CD-RW disc, the recording laser turns the amorphous areas back into crystalline areas

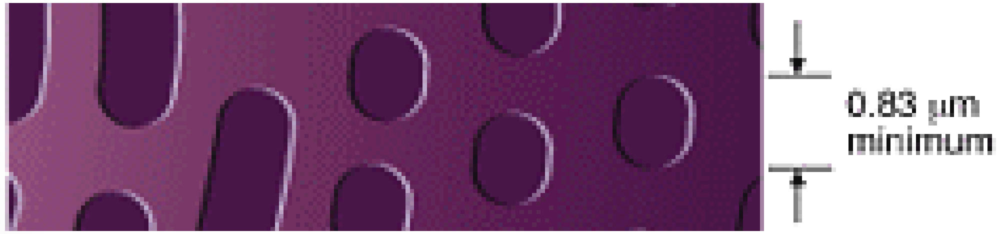
DVD

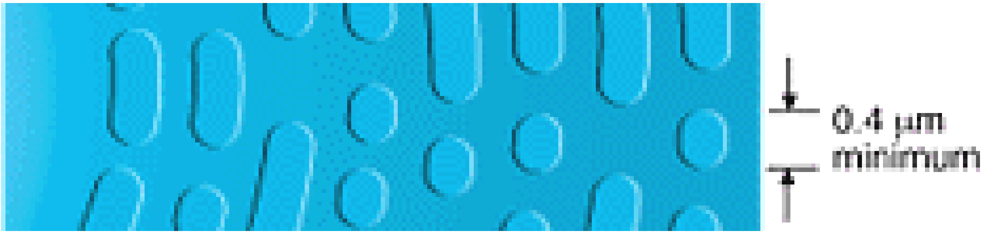
- **Digital Versatile Disk** (Formerly **Digital Video Disk**)

- same size (120mm) and thickness (1.2mm) as CD
- Improvements in the logarithms used for error correction
- Much greater data accuracy using smaller Error Correction Codes (ECC)
- More effective use of the track space

DVD vs. CD

- DVD uses a tighter spiral (track or helix) with only 0.74 microns between the tracks (1.6 microns on CDs)
- DVD recorders use a laser with a smaller wavelength, 635nm or 650 nm (visible red light) vs. 780nm (infrared) for CDs
- DVD has smaller "burns" (pits) in the translucent dye layer (0.4 microns minimum vs. 0.83 microns minimum on CDs)





- These technologies allow DVDs to store large amounts of data.

DVD (cont’d)

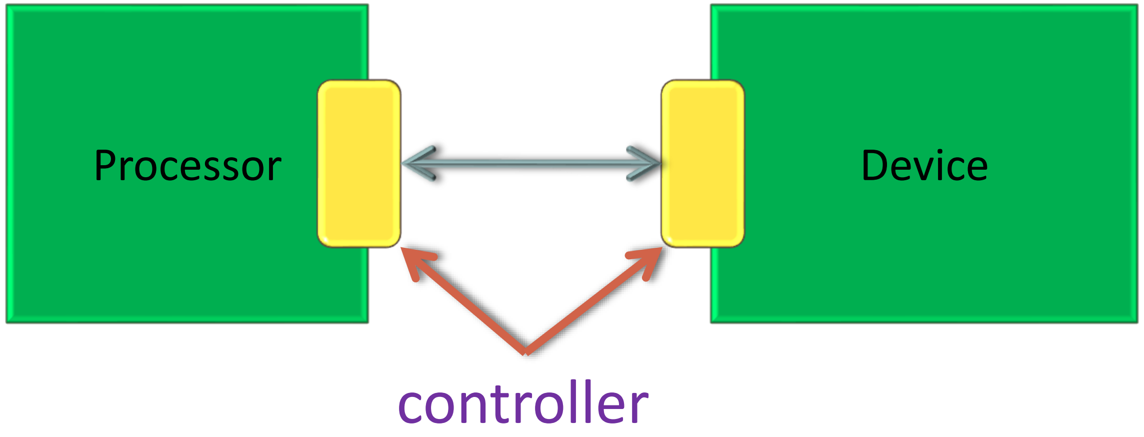
- Standard single-sided DVDs store up to 4.7GB of data
- Dual-sided discs hold about 8.5GB of data (9.4GB for back-to-back layers dual-sided discs)
- In back-to-back layers discs, it must be turned over to access the data on the reverse side
- DVD uses MPEG2 compression for high quality pictures
- DVD drives have a much faster transfer rate than CD drives
- DVD-ROM drives will read and play existing CD-ROM and CD-A disks

BLUE RAY

- The future
- Blue laser rather than red, shorter wavelength.
- Allows more accurate focus and smaller pits.
- Single sided 25Gb 4.5 Mb/sec
- Still not fast compared to ATA100 Mb/sec

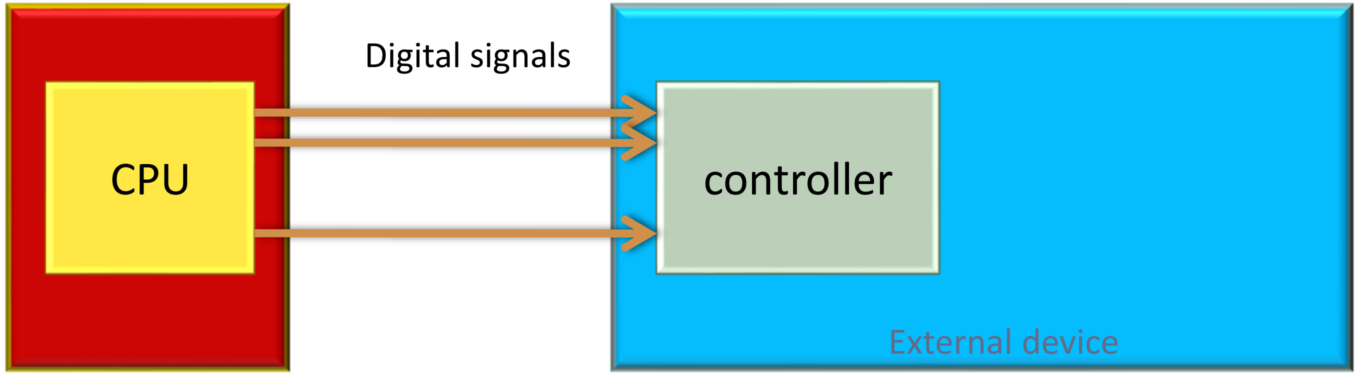
Input and Output Devices

- External devices
  - Storage devices: disk, tape, DVD.
  - Transmission devices: network cards, modems.
  - Human-interface devices: screen, keyboard, mouse.
- A device **controller** connects the device to the computer's bus and controls the operation of the device.



I/O Data Transfer

- A device communicates with a system by sending signal through a cable or through the air.
  - **Port**: a connection point between a device and the machine
    - Parallel port, serial port, USB port
  - **Bus**: a set of wires shared by one or more devices.
    - Use patterns of electrical voltages to represent messages/signals



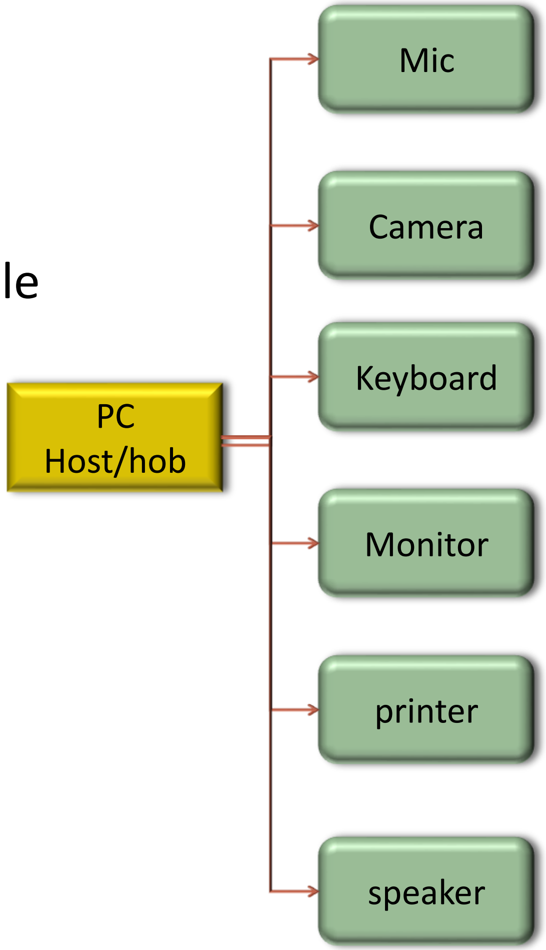
Serial and Parallel Data Transfers

- **Serial interface** transfers one bit at any time.
  - Only two wires
- **Parallel interface** transfers multiple bits of data simultaneously.
  - Interface width refer to the number of parallel wires an interface uses
- Faster I/O interface
  - **USB (Universal Serial Bus)**

Universal Serial Bus

- USB was designed to
  - Allow up to 127 of low- to medium-speed peripherals to be attached to a PC.
  - Be plug and play
- USB uses bus architecture
  - Data from different devices travel across the same cable
  - Use multilevel star topology
- Speed

Port	Speed
Serial Port	230 Kp/s
Parallel Port	6 Mb/s
USB 2.0 (Intel)	480 Mb/s
USB 3.0	5Gb/s—16Gb/s



I/O Controller

- How CPU communicates with a controller
  - A controller has a set of registers for data and control signals.
  - CPU executes I/O requests by reading and writing device-control registers

I/O Controller

- A controller typically consists of four registers:
  - **Status register** — the status of the device.
    - Command completed, data ready to be read, error
  - **Command register** — command from the host.
    - Certain task, change the mode of the device
  - **Data-in register** — input data for the host .
  - **Data-out register** — output data from the host.

I/O Polling

- A simple protocol for interaction between the host and controller
  - The controller indicates its state through the **busy** bit in the **status register**.
  - The host actively monitoring the status of a device by reading the **busy** bit in the **status register**.

An I/O Polling Example

- The host writes output to a controller.

For each byte, the following loop is performed.

1. The CPU repeatedly reads the **busy** bit in the **status register** until that bit becomes clear
2. The host sets the **write** bit in the **command register** and write a byte into the **data-out register**
3. The host sets the **command-ready** bit in the **command register**
4. When the controller notices that the **command-ready** bit is set, it sets the **busy** bit
5. The controller reads the **command register** and sees the **write** command. It reads the **data-out register** to get the byte and does the I/O to the device.
6. The controller clears the **command-ready** bit, clears **error** bit in the **status register** to indicate that the I/O succeeded, and clears the **busy** bit to indicate that the I/O finished and it is free now.
7. Go back to 1

I/O Polling

- It is efficient if the controller and device are fast and always ready for service.
  - Normally It takes only 3 CPU cycles to poll a service.
  - It needs to poll the device only once if the device is ready.
- It is inefficient if a device is not ready.
  - The polling has to be repeated until the device is ready.
  - It may be more efficient to have the device controller notify the CPU that a device is ready for service – **Interrupt**.

Interrupts

- It enables the CPU to response to an asynchronous event
  - An efficient way to dispatch to the proper interrupt handler for a device, without first polling all the devices to see which one raised the interrupt

Interrupts

- Two categories
  - **Hardware interrupts**
    - Such as system timers, disks I/O, power-off signals, etc
  - **Software interrupts** (also called **traps**)
    - Such as page fault, division by zero, etc

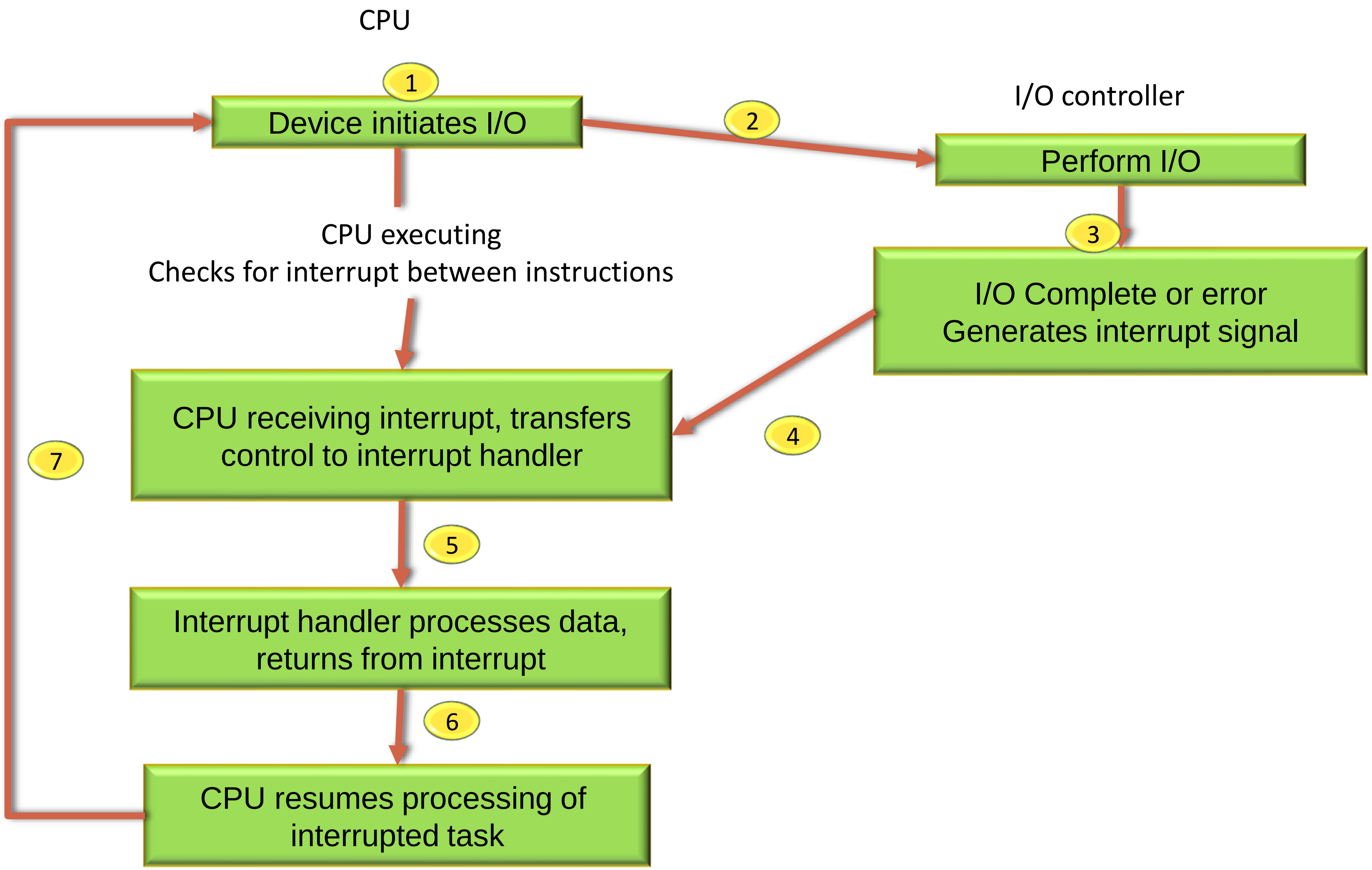
Interrupts

- A wire called **Interrupt request line** is used to signal interrupt.
- A device wishing to signal an interrupt asserts a signal to the **interrupt request line.**
- The CPU senses the **interrupt request line** after executing every instruction

Interrupt-driven I/O Cycle

- The I/O interrupt cycle:
  - The device controller raises an interrupt
  - The CPU detects the interrupt, saves its state, and then jumps to the interrupt-handler routine.
  - The interrupt handler performs the necessary processing.
  - The interrupt handler returns the CPU to the execution state prior to the interrupt.

Interrupt-driven I/O Cycle


CPU
1
Device initiates I/O
2
I/O controller
Perform I/O
3
CPU executing
Checks for interrupt between instructions
I/O Complete or error
Generates interrupt signal
4
CPU receiving interrupt, transfers control to interrupt handler
7
5
Interrupt handler processes data, returns from interrupt
6
CPU resumes processing of interrupted task

CPU Chips

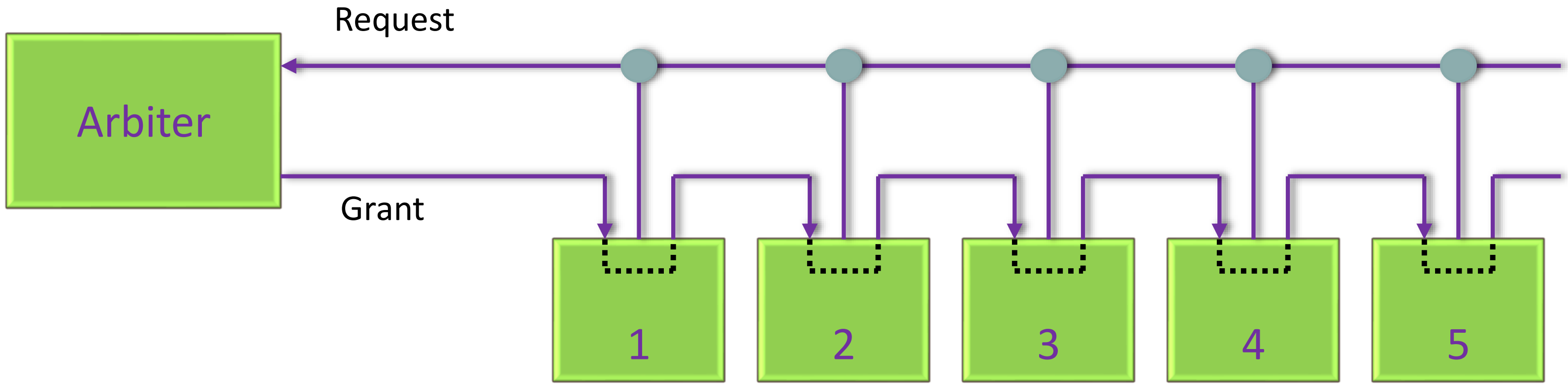
- All modern CPU’s are built into a single chip
- The CPU interacts with the outside world solely through physical wires (*lines*):
  - Address lines
  - Data lines
  - Control lines
- E.g. to fetch data from the memory:
  - Place address of data on address lines
  - Signal that the address is ready using one or more control lines
  - Memory responds by placing required data on the data lines, then signaling that it has finished
  - The CPU reads the data from the bus

CPU Performance Parameters

- The number of address lines (*m*)
  - Can address up to 2^m addresses
  - Commonly m=16, 20, 32 or 64
- The number of data lines (*n*)
  - Can read/write n-bit word in a single operation
  - Commonly, n=8, 16, 32, 36 or 64
- A chip with 8 data pins will take 4 cycles to read a 32-bit word

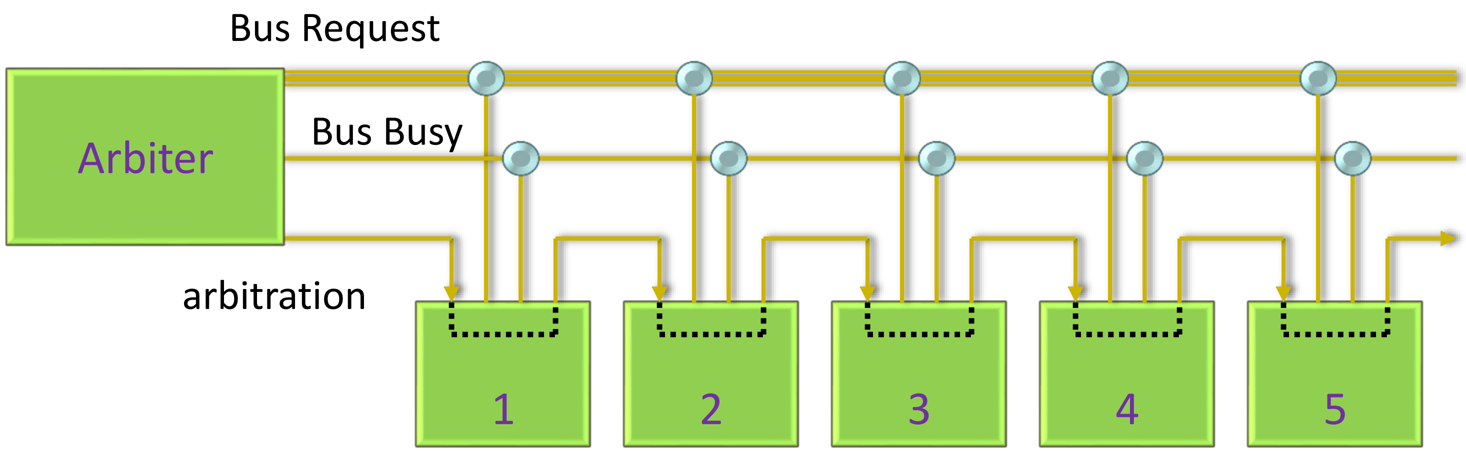
- What happens if two devices want to use the bus at the same time?
- **The *Bus Arbitrator*** decides who gets the bus in case of a conflict
- Two main types of arbitrator
  - Centralised
  - Decentralised

Bus Arbitration



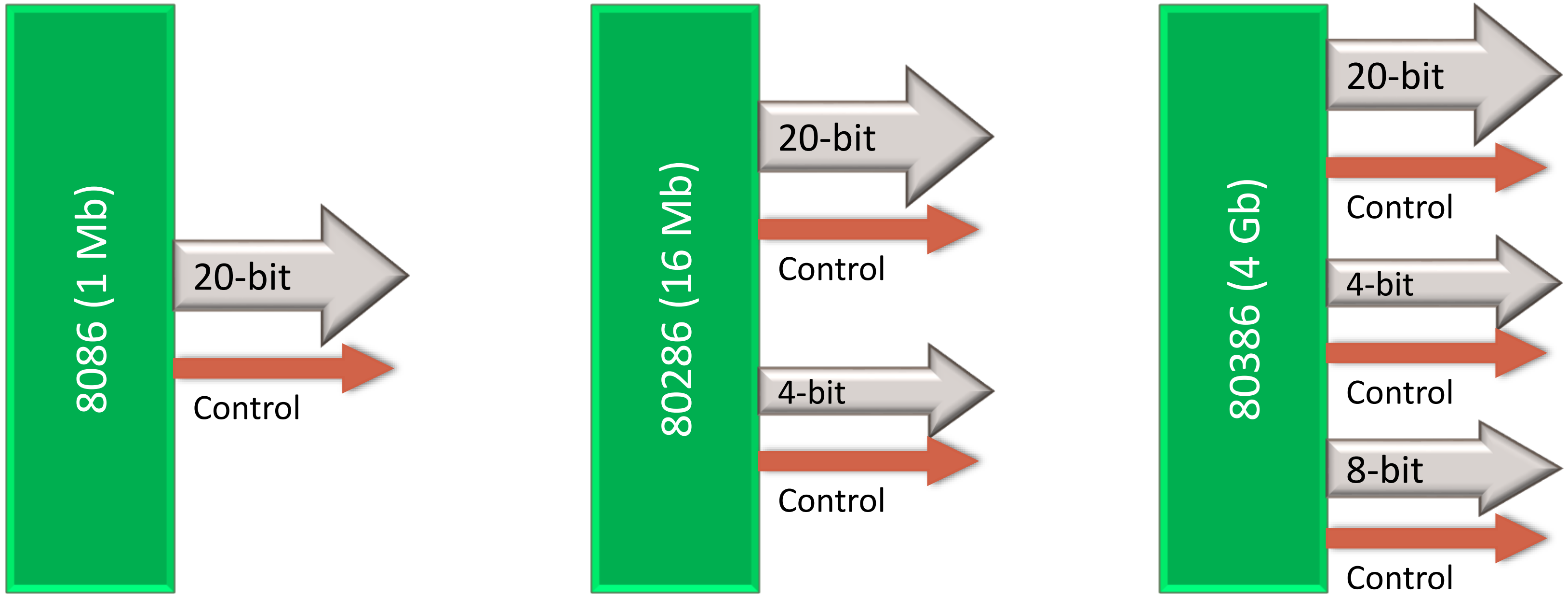
- This type of arbitration is used with the PCI bus (more on this latter)
- Each device must ask to use the bus
- When the arbiter receives a request, it issues a grant
  - The arbiter doesn’t know who asked for the bus
- If a device doesn’t need the bus, it passes the grant along to the next (*daisy-chaining*)
  - Sets up implicit priorities between devices

Centralised Arbitration



- There are several request lines. All devices know their own priority level and monitor all request lines
- To acquire the bus, the device must first check to see if a higher priority device wants the bus. It sets the request line if there isn’t any, then waits until the bus is not busy and the inward arbitration signal is set
  - It then unsets the outward arbitration and request lines, sets the busy line. It can then use the bus
  - When done, it unsets the busy line and resets the outward arbitration line

Decentralized Arbitration

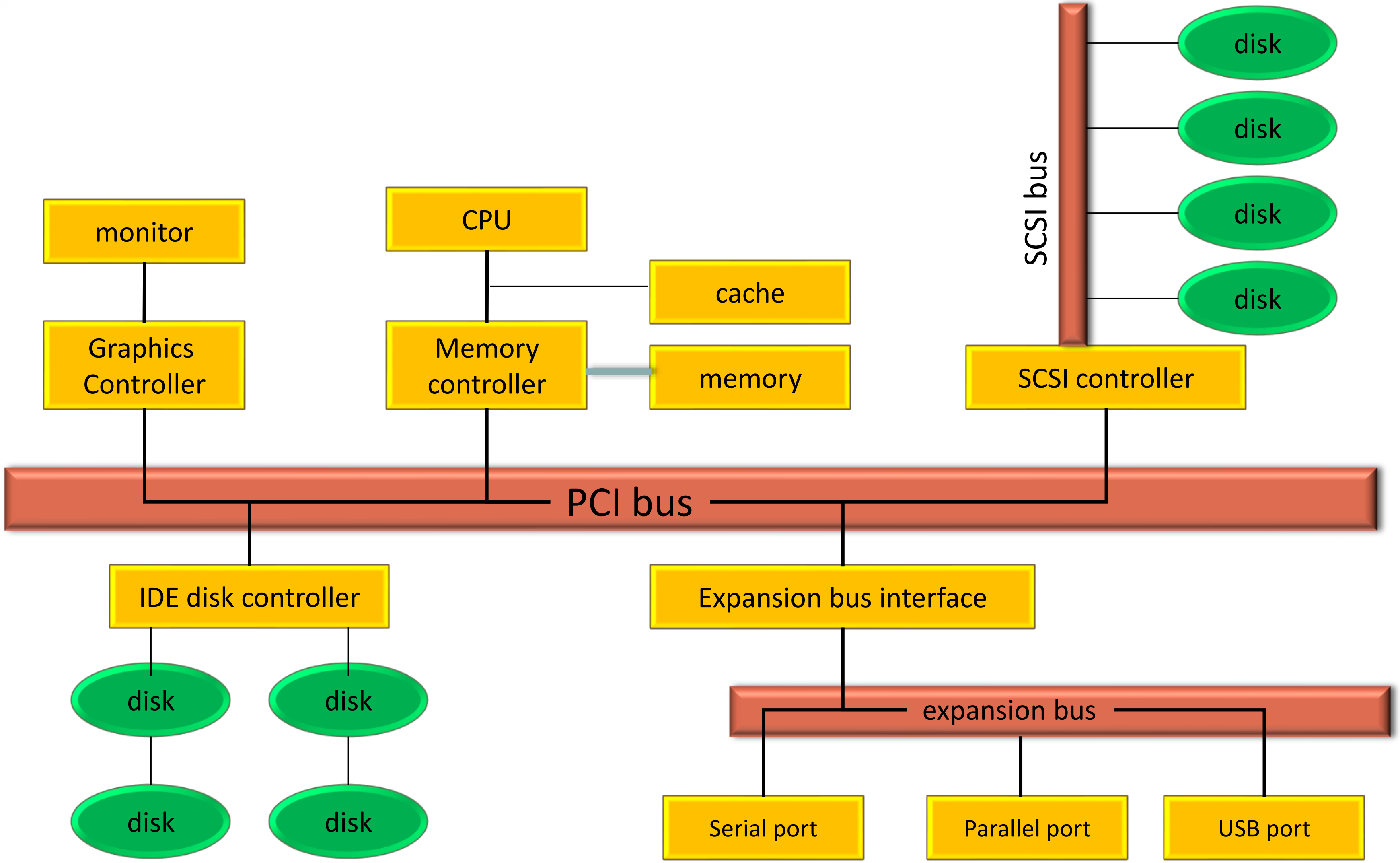


- Industry-Standard Architecture Bus (8.33 Mhz)
- 8086 has a 20 bit bus
- 80286 has a 20 bit + 4 bit bus
  - (Backward compatibility)
- 80386 has a 20 bit + 4 bit + 8 bit bus
- Extra control lines are also needed

The ISA Bus

The PCI (Peripheral Component Interconnect) Bus

- ISA is too slow for multimedia applications
  - Maximum rate is 16.7 Mb/s
  - Need at least 100 Mb/s for smooth full-screen, full-colour video
- Intel created the PCI bus to deal with this
  - 32 bits/cycle, 33 Mhz, 133 Mb/s
  - PCI-2: 66 Mhz, 64 bits/cycle, 528 Mb/s
  - Multiplexed data and address lines
- But PCI still not fast enough for a memory bus



A Typical Bus Structure

PCI-X (Extended)

- **PCI-X**, short for Peripheral Component Interconnect eXtended, is a computer bus and expansion card standard that enhances the 32-bit PCI Local Bus for higher bandwidth demanded by servers. It is a double-wide version of PCI, running at up to four times the clock speed, but is otherwise similar in electrical implementation and uses the same protocol.
- It has been replaced in modern designs by the similar-sounding PCI Express (officially abbreviated as PCIe), with a completely different connector and a very different logical design, being a single narrow but fast serial connection instead of a number of slower connections in parallel.

PCI Express (PCIe)

- **PCI Express** (**Peripheral Component Interconnect Express**), officially abbreviated as **PCIe**, is a high-speed serial computer expansion bus standard designed to replace the older PCI, PCI-X, and AGP bus standards. PCIe has numerous improvements over the aforementioned bus standards, including higher maximum system bus throughput, lower I/O pin count and smaller physical footprint, better performance-scaling for bus devices, a more detailed error detection and reporting mechanism (Advanced Error Reporting (AER), and native hot-plug functionality. More recent revisions of the PCIe standard support hardware I/O virtualization.
- The PCIe electrical interface is also used in a variety of other standards, most notably ExpressCard, a laptop expansion card interface.

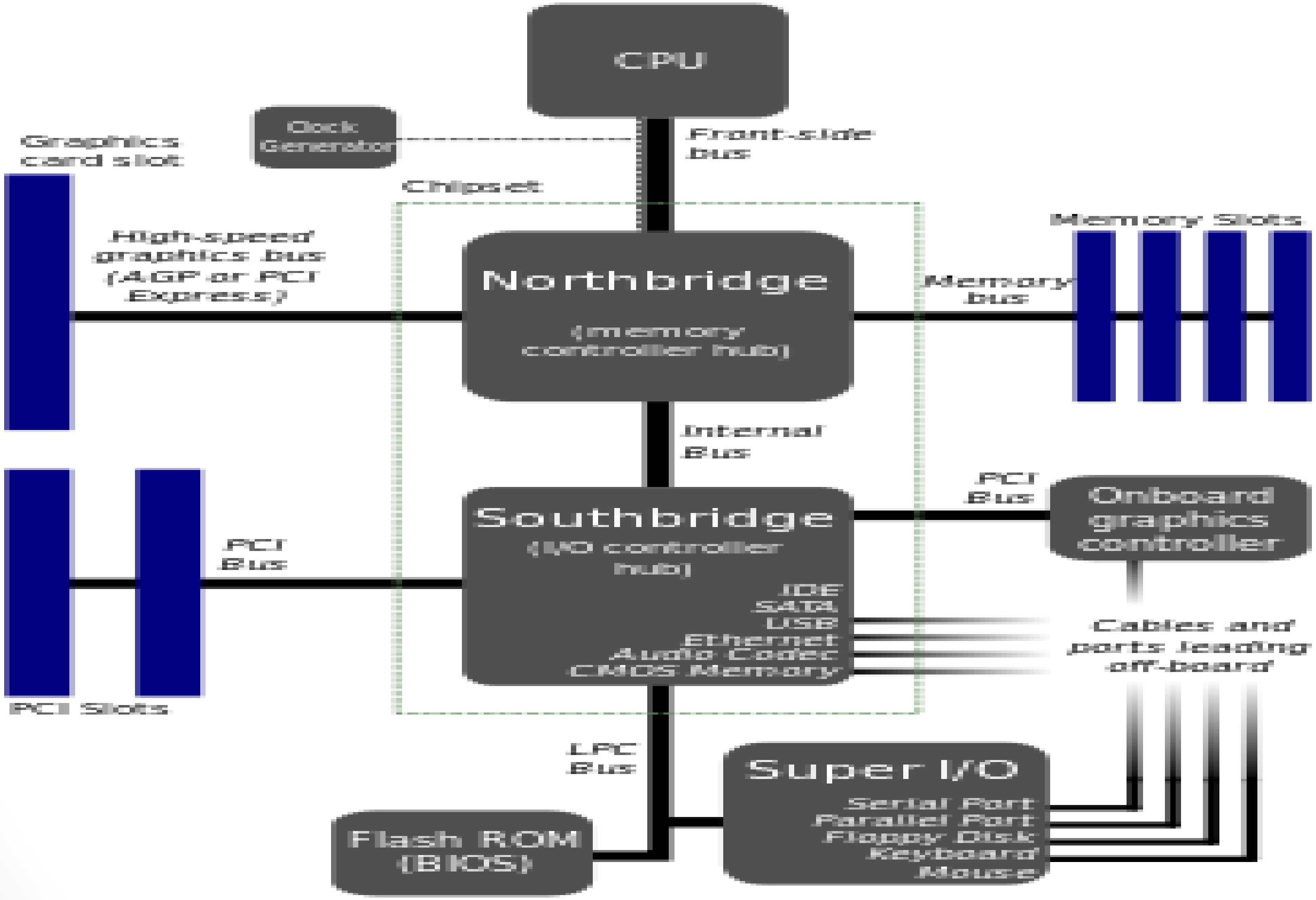
SOUTHBRIDGE controller

- The **southbridge** was one of the two chips in the core logic chipset on a personal computer (PC) motherboard, the other being the northbridge.
- The southbridge typically implements the slower capabilities of the motherboard in a northbridge/southbridge chipset computer architecture. In Intel chipset systems, the southbridge is named *Input/Output Controller Hub (ICH)*. AMD, beginning with its Fusion APUs, has given the label *FCH*, or *Fusion Controller Hub*, to its southbridge.
- The southbridge can usually be distinguished from the northbridge by not being directly connected to the CPU. Rather, the northbridge ties the southbridge to the CPU. Through the use of controller integrated channel circuitry, the northbridge can directly link signals from the I/O units to the CPU for data control and access.

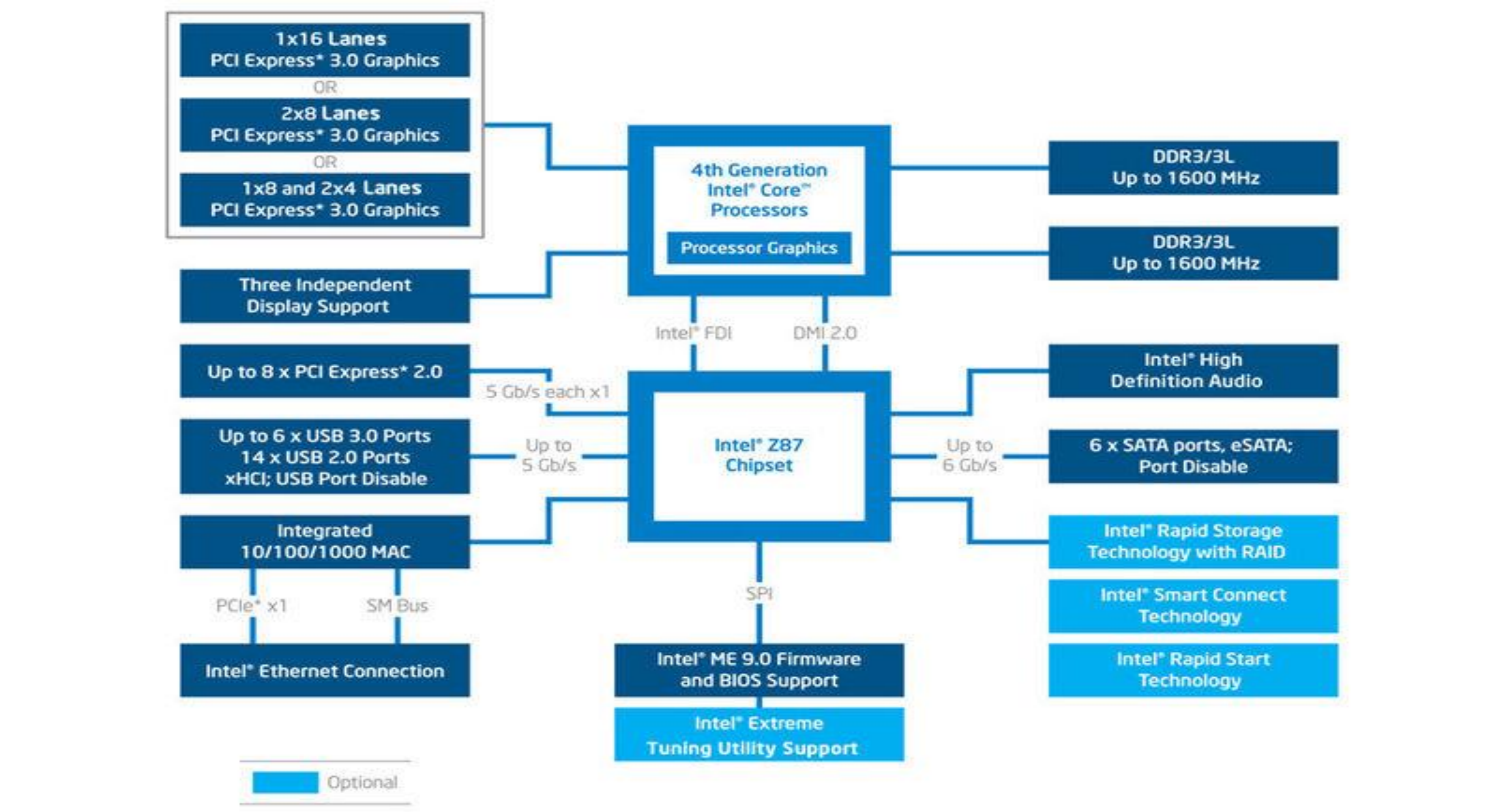
Southbridge con’t

- A southbridge chipset handles all of a computer's I/O functions, such as USB, audio, serial, the system BIOS, the ISA bus, the interrupt controller and the IDE channels. Different combinations of Southbridge and Northbridge chips are possible, but these two kinds of chips must be designed to work together.
- There is no industry-wide standard for interoperability between different core logic chipset designs. Traditionally, the interface between a northbridge and southbridge was the PCI bus. The main bridging interfaces used now are DMI (Intel) and UMI (AMD).

OLD Southbridge/Northbridge



Newer South/Northbridge PC’s



1x16 Lanes
PCI Express* 3.0 Graphics
OR
2x8 Lanes
PCI Express* 3.0 Graphics
OR
1x8 and 2x4 Lanes
PCI Express* 3.0 Graphics

Three Independent
Display Support

4th Generation
Intel® Core™
Processors

Processor Graphics

DDR3/3L
Up to 1600 MHz

DDR3/3L
Up to 1600 MHz

Intel® FDI

DMI 2.0

Up to 8 x PCI Express* 2.0

5 Gb/s each x1

Up to 6 x USB 3.0 Ports
14 x USB 2.0 Ports
xHCI; USB Port Disable

Up to
5 Gb/s

Intel® Z87
Chipset

Up to
6 Gb/s

Intel® High
Definition Audio

6 x SATA ports, eSATA;
Port Disable

Integrated
10/100/1000 MAC

Intel® Rapid Storage
Technology with RAID

Intel® Smart Connect
Technology

Intel® Rapid Start
Technology

PCIe* x1

SM Bus

SPI

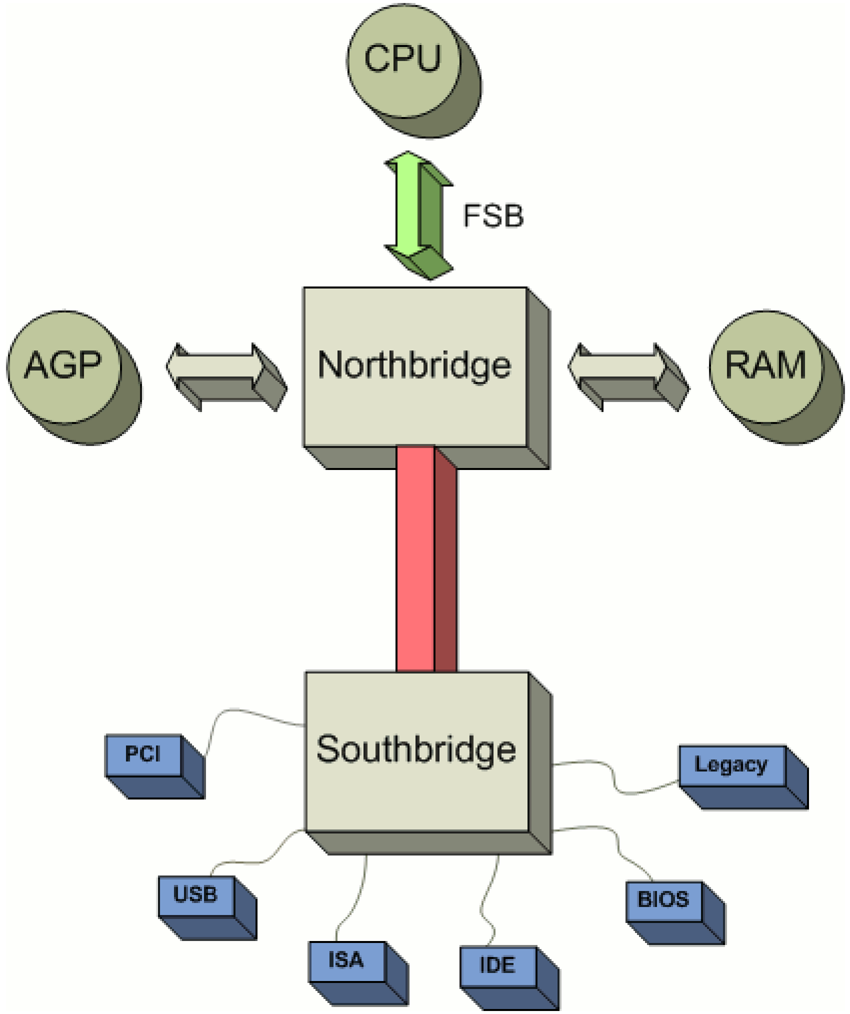
Intel® Ethernet Connection

Intel® ME 9.0 Firmware
and BIOS Support

Intel® Extreme
Tuning Utility Support

🟦 Optional

BUS ARBITRATORS



Summary

- Secondary Memory
  - HDD structure, capacity, formatting and speed
- I/O device
  - I/O controller and its registers
  - Communication to CPU
    - Port Interfaces: Serial, parallel and USB ports
    - Polling, interrupts
- Bus
  - The width of the bus
  - The arbitration scheme used
  - ISA bus and PCI bus