Memory Architecture II

CS2053 Computer Architecture

Computer Science & Engineering

University of Moratuwa

Sulochana Sooriyaarachchi

Chathuranga Hettiarachchi

Outline

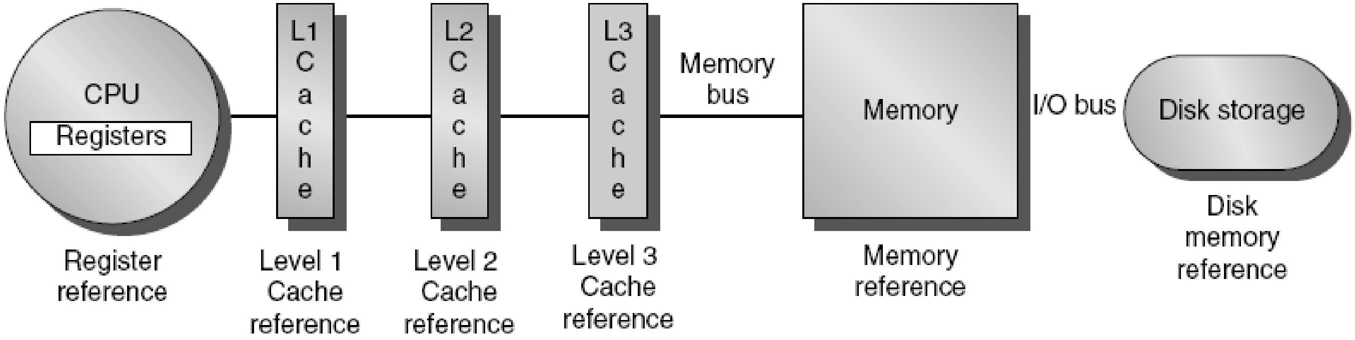
- Memory types
- Memory access
- Memory hierarchy
  - Main memory
  - Cache
  - Permanent storage
- Example architecture RV32I
  - Superscalar architecture
  - Pipelining

Cache

Cache Memory

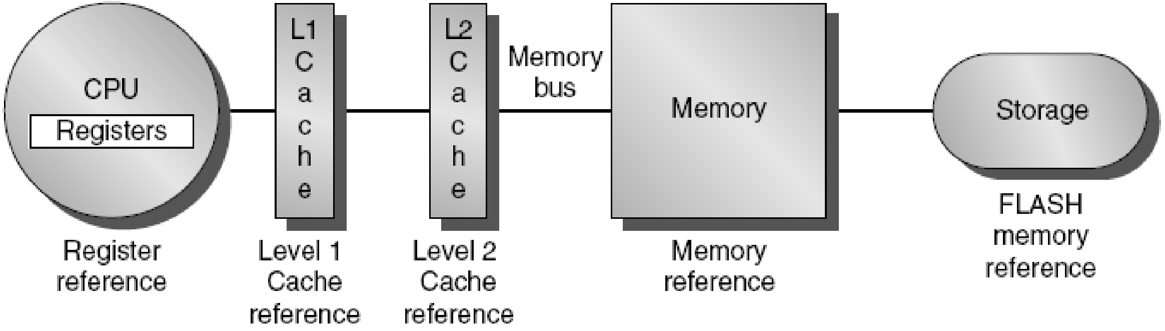
- Small amount of memory that is faster than DRAM
  - Slower than registers
  - Built using SRAM
  - Range from few KB to few MB
- Used by CPU to store frequently used instructions & data
  - Spatial & temporal locality
- Use multiple levels of cache
  - L1 Cache – Very fast, usually within CPU itself
  - L2 Cache – Slower than L1, but faster than DRAM
  - Today there’s even L3 Cache

Multiple Levels of Caching



Size:	1000 bytes	64 KB	256 KB	2–4 MB	4–16 GB	4–16 TB
Speed:	300 ps	1 ns	3–10 ns	10–20 ns	50–100 ns	5–10 ms

(a) Memory hierarchy for server

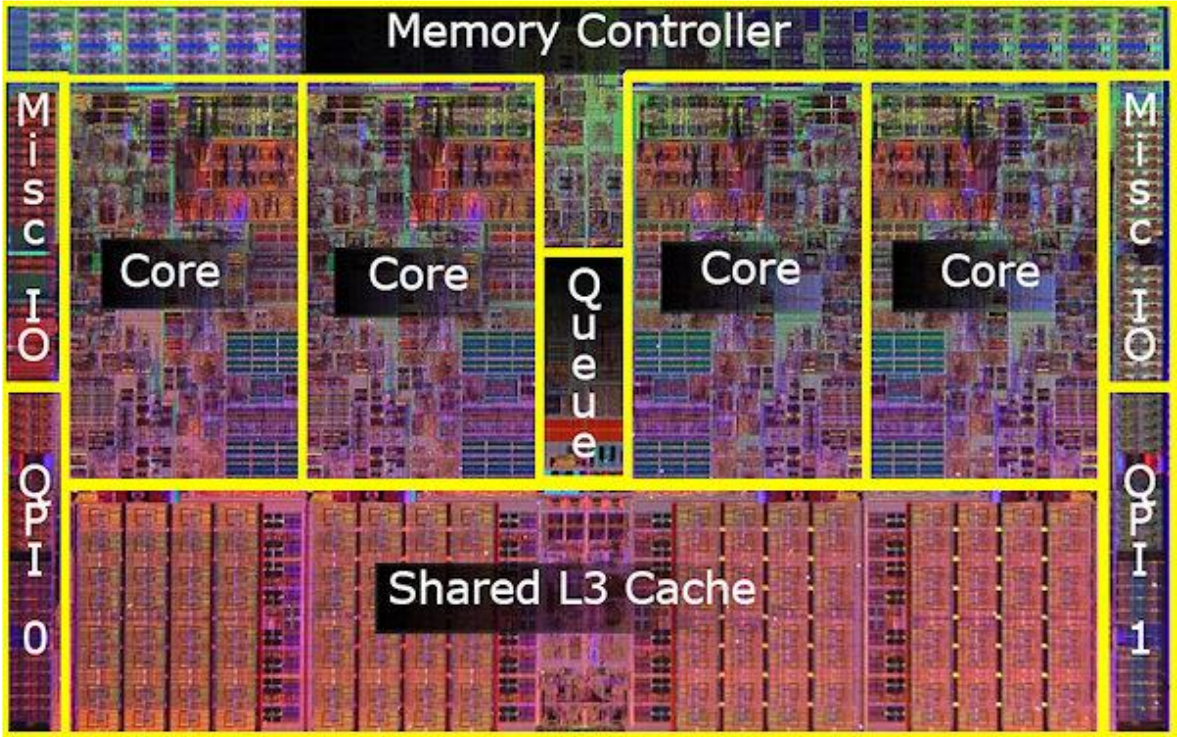


Size:	500 bytes	64 KB	256 KB	256–512 MB	4–8 GB
Speed:	500 ps	2 ns	10–20 ns	50–100 ns	25–50 us

(b) Memory hierarchy for a personal mobile device

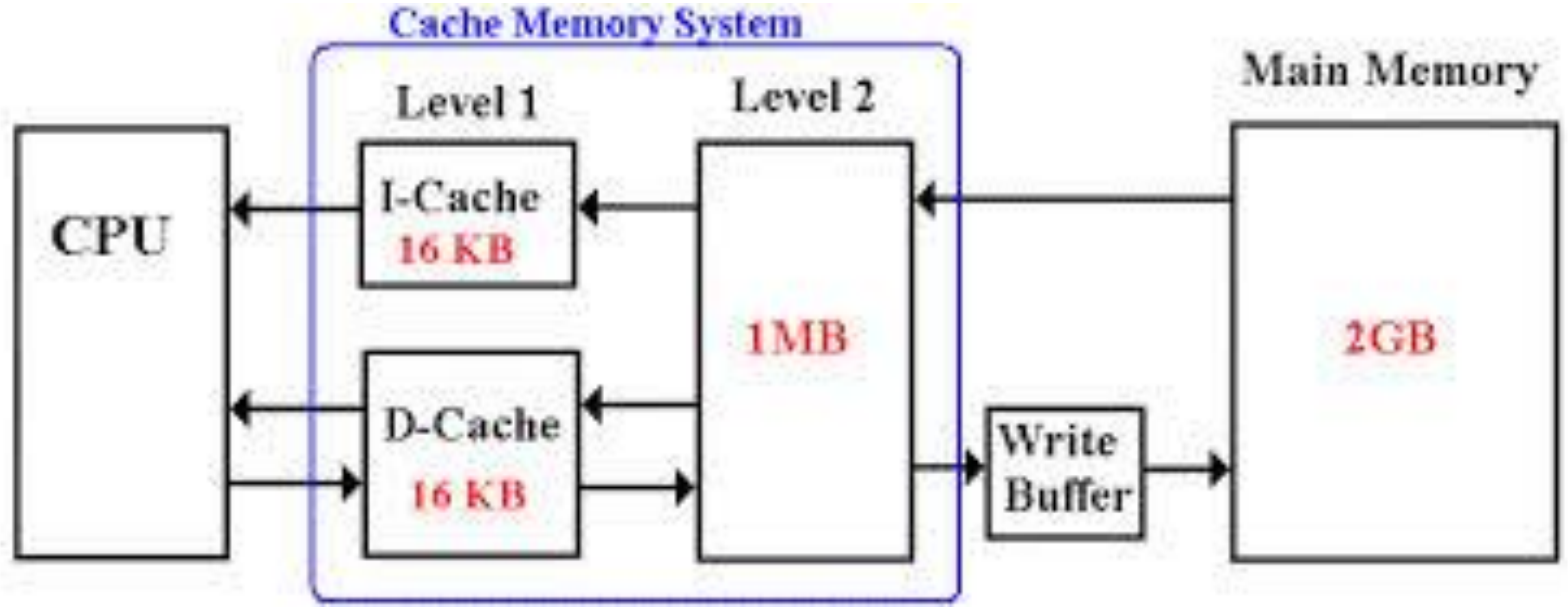
Source: Computer Architecture, A Quantitative Approach by John L. Hennessy and David A. Patterson

Core i7 Die & Major Components



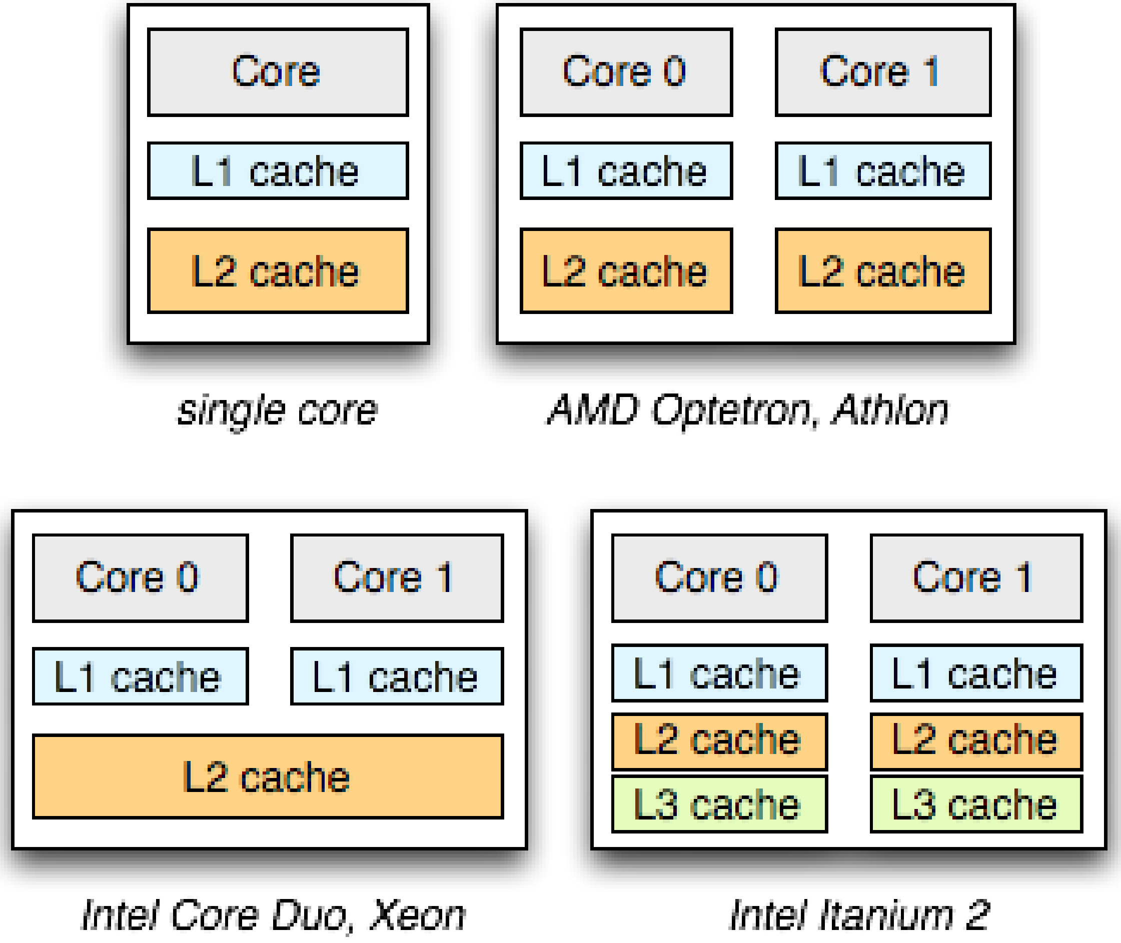
Source: Intel Inc.

L1 & L2 Cache



Source: www.edwardbosworth.com/CPSC2105/Lectures/Slides_06/Chapter_07/Pentium_Architecture.htm

Caching in Multi-Core Systems

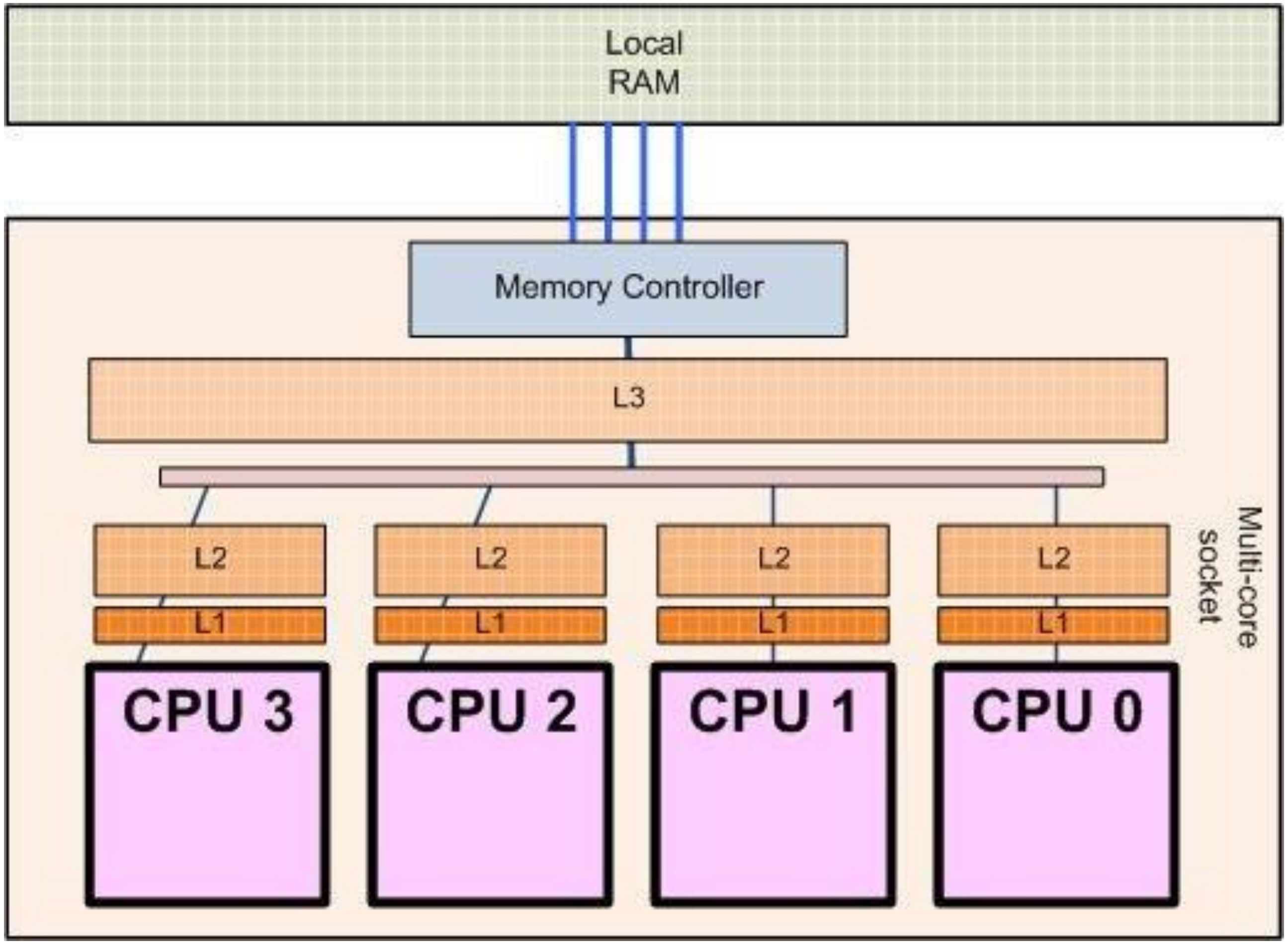


single core *AMD Optetron, Athlon*

Intel Core Duo, Xeon *Intel Itanium 2*

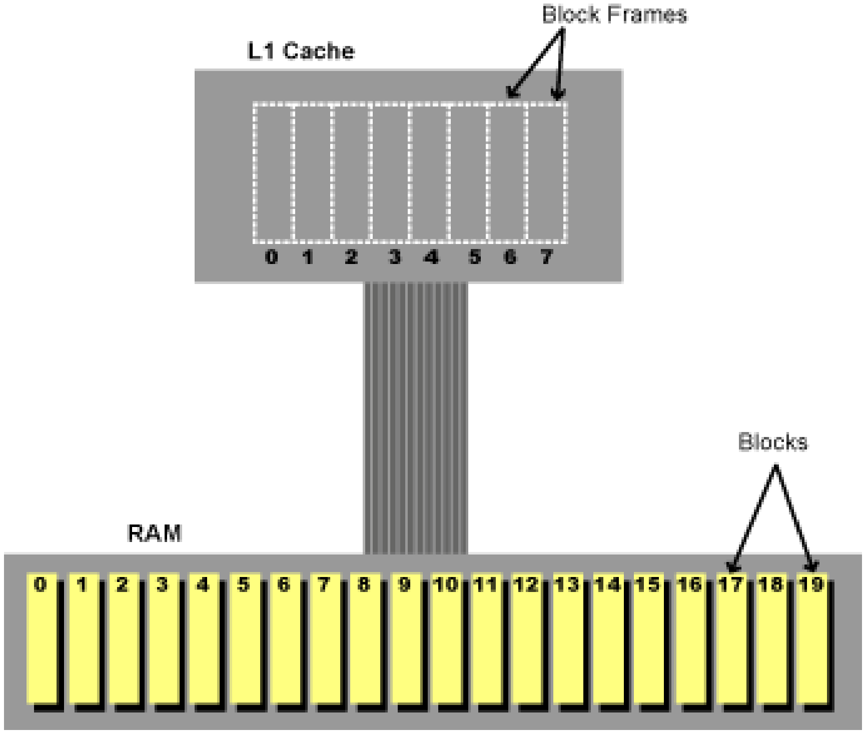
Source: www.igvita.com/2010/08/18/multi-core-threads-message-passing/

Caching in Multi-Core Systems (Cont.)



Source: http://sips.inesc-id.pt/~nfvr/msc_theses/msc09e/

Cache Blocks

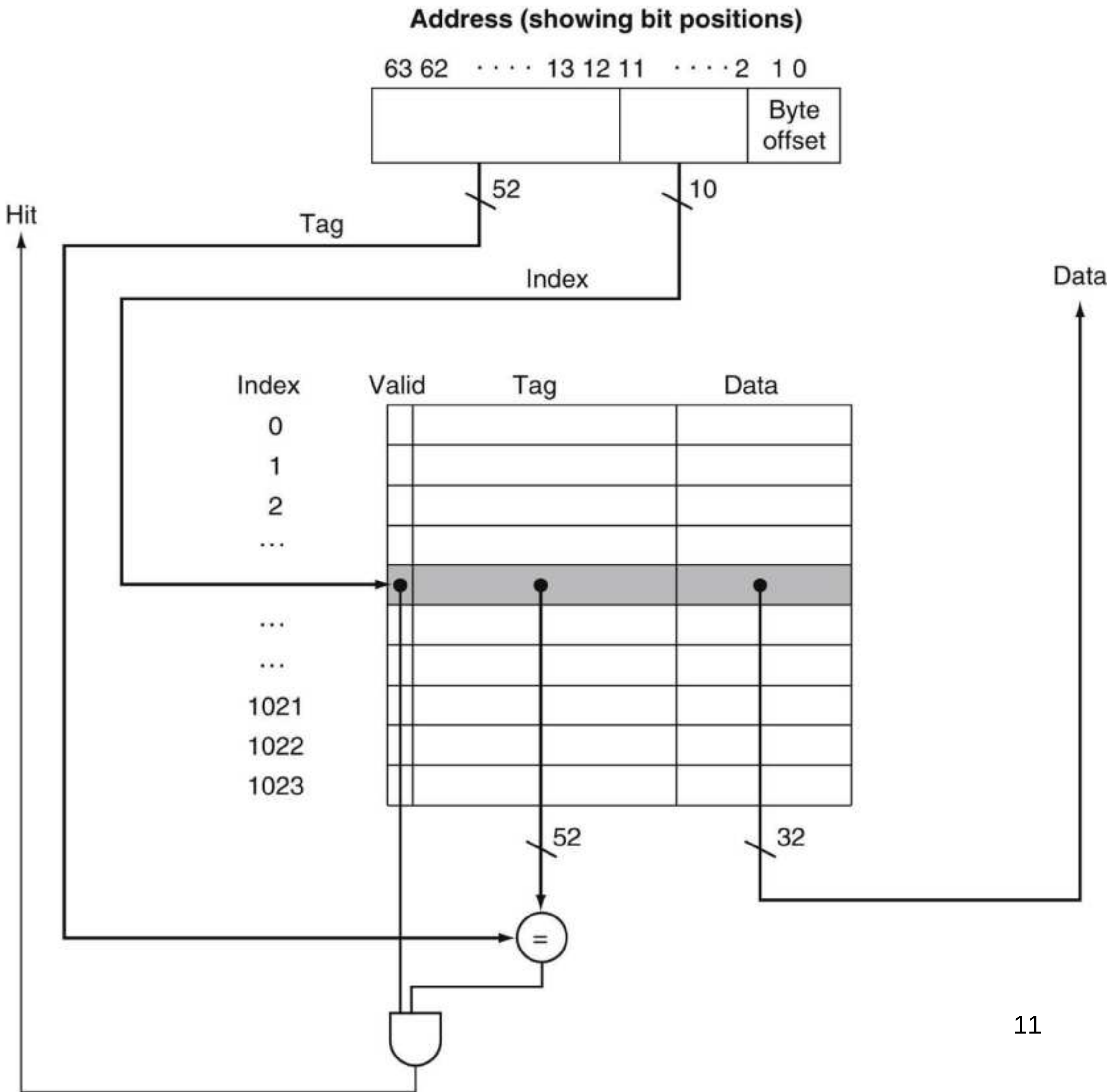


Source: http://archive.arstechnica.com/paedia/c/caching/m-caching-5.html

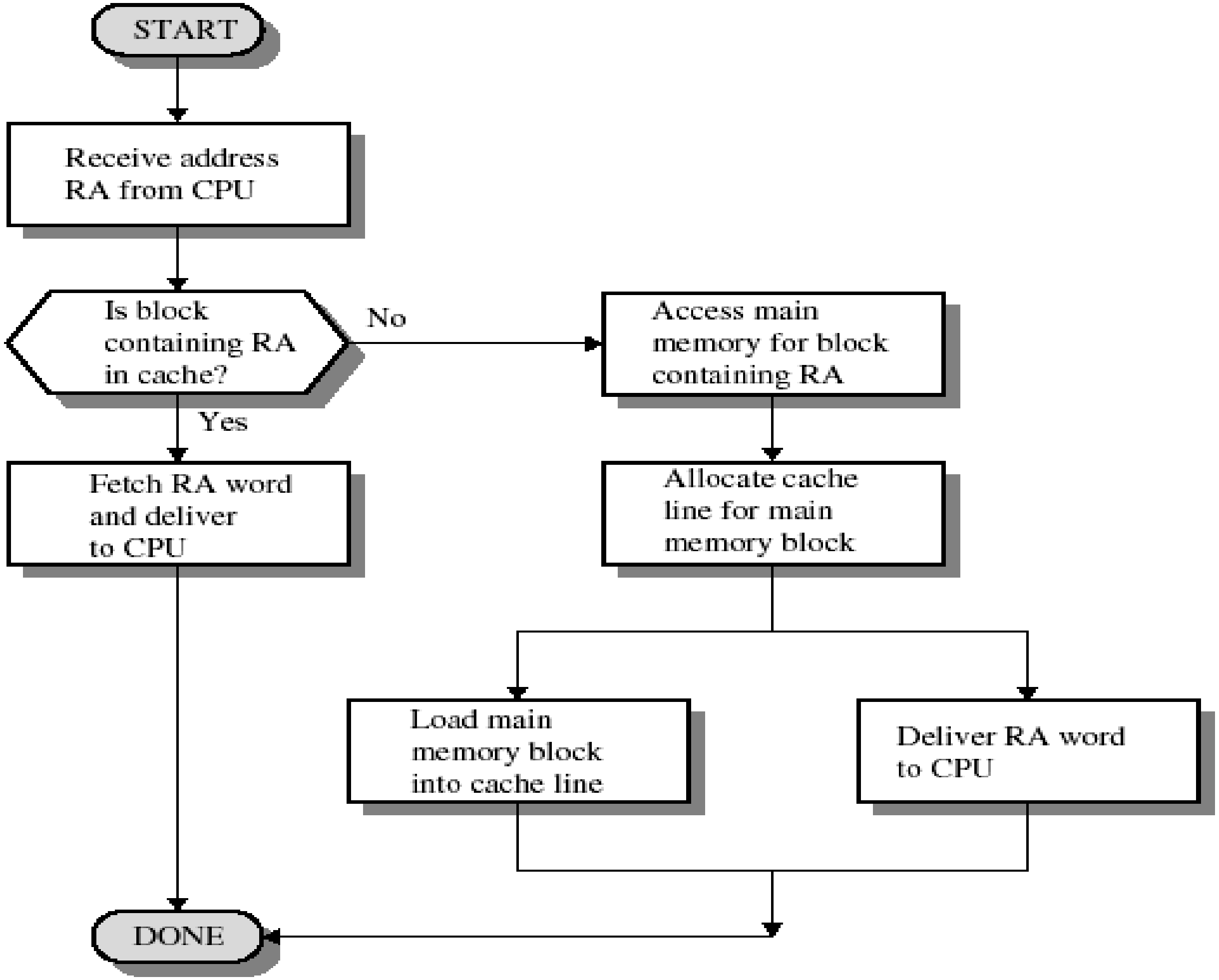
- A collection of *words* are called a *block*
- Multiple blocks are moved between levels in cache hierarchy
- Blocks are tagged with memory address
  - Tags are searched parallel

Accessing Cache

- Tag = upper portion of address
- Index = (Memory block address) modulo #Cache blocks
- Valid bit



Cache Read Operations


START
Receive address RA from CPU
Is block containing RA in cache?
No
Access main memory for block containing RA
Yes
Fetch RA word and deliver to CPU
Allocate cache line for main memory block
Load main memory block into cache line
Deliver RA word to CPU
DONE

Cache Misses

- When required item is not found in cache
- Miss rate – fraction of cache accesses that result in a failure
- Types of misses
  - Compulsory – 1st access to a block
  - Capacity – limited cache capacity force blocks to be removed from a cache & later retrieved
  - Conflict (collision)– multiple blocks compete for the same set/block
- Average memory access time

  = *Hit time* + *Miss rate* x *Miss penalty*

Cache Misses – Example

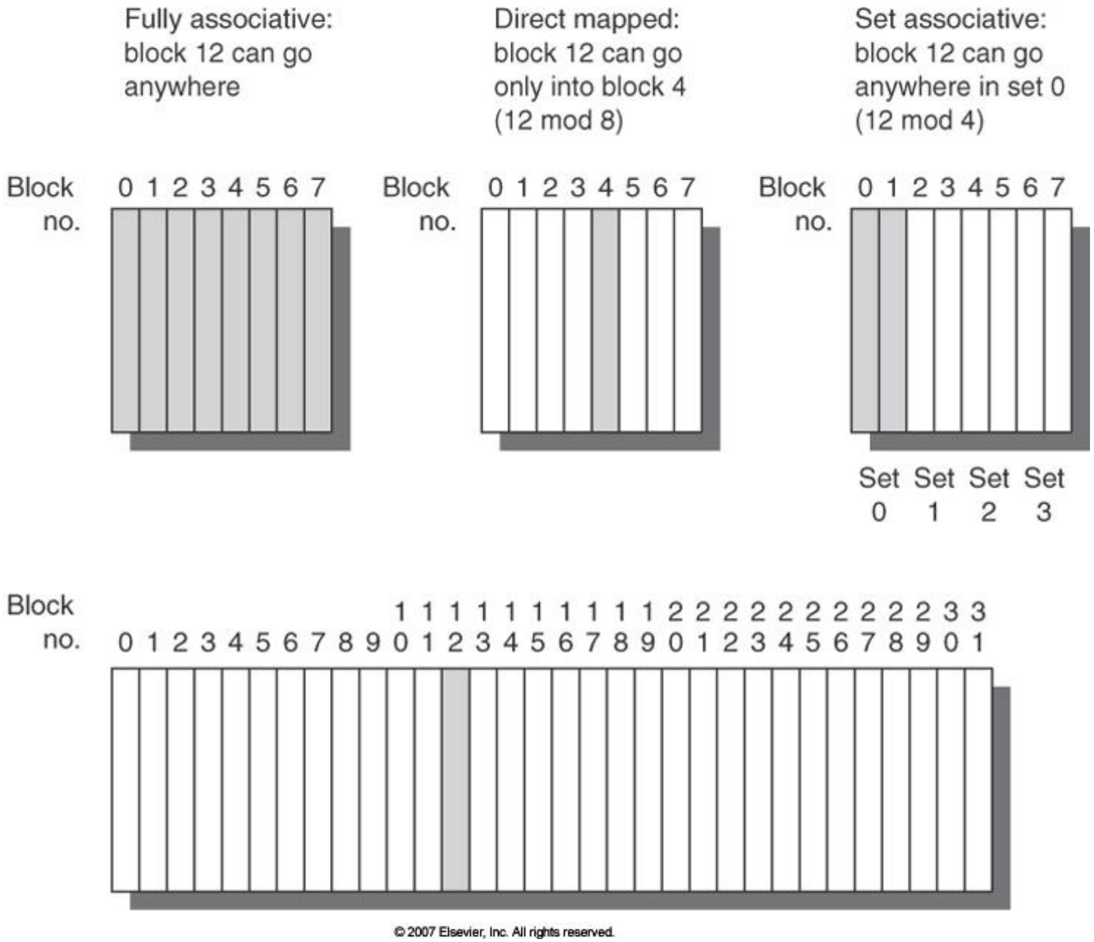
- Consider a cache with a block size of 4 words
- It takes 1 clock cycle to access a word in cache
- It takes 15 clock cycles to access a word from main memory
  - How much time will it take to access a word that's not in cache?
  - What is the average memory access time if miss rate is 0.4?
  - What is the average memory access time if miss rate is 0.1?

Cache Misses – Example

- Assume 40% of the instructions are data accessing instructions
- A hit takes 1 clock cycle & miss penalty is 100 clock cycles
- Assume instruction miss rate is 4% & data access miss rate is 12%, what is the average memory access time?

Cache Associativity

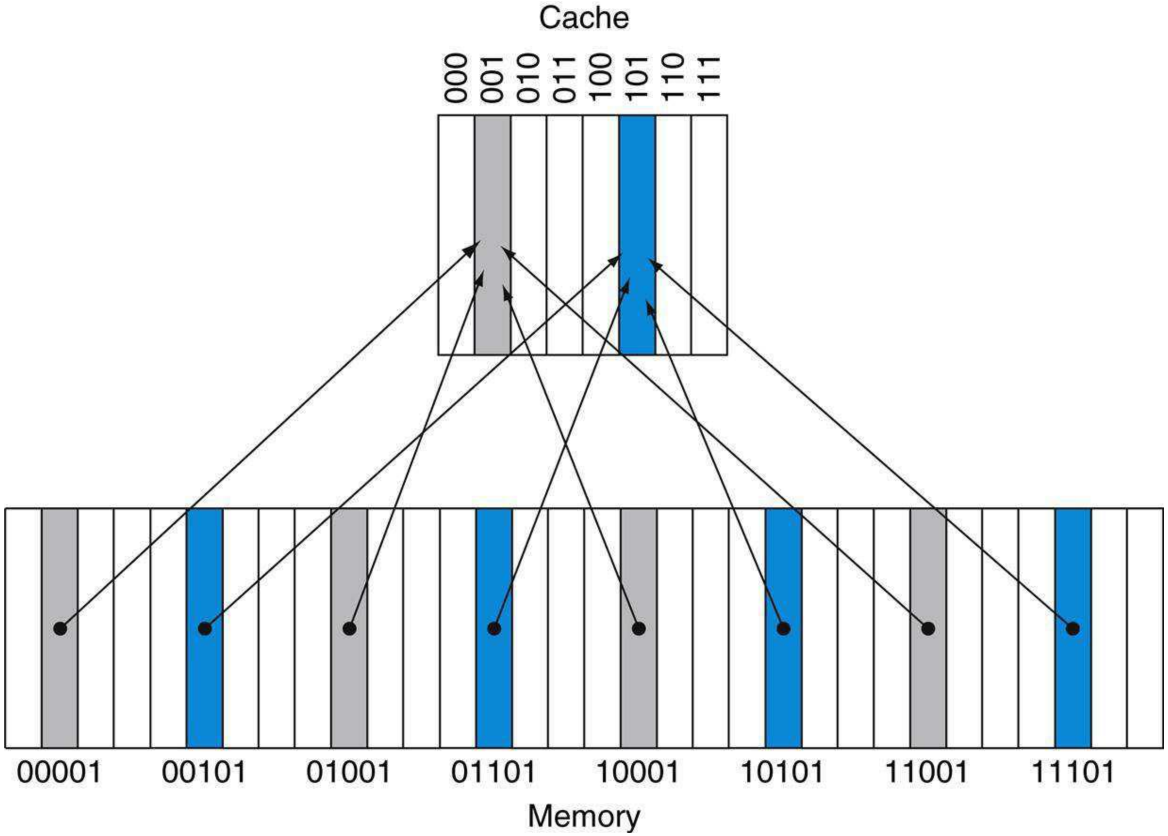
- Defines where blocks can be placed in a cache



Source: Computer Architecture, A Quantitative Approach by John L. Hennessy and David A. Patterson

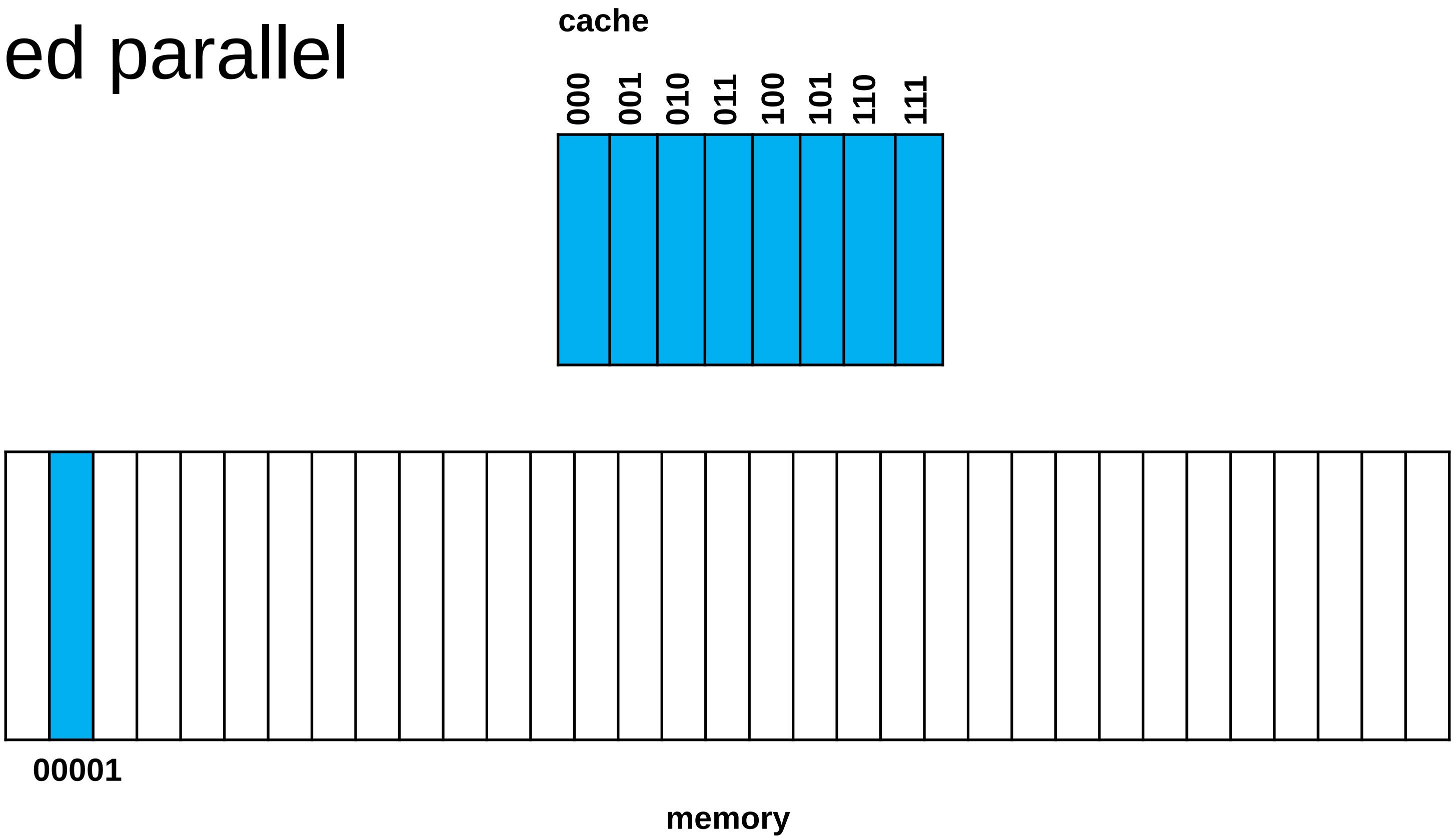
Direct mapping

- A memory block can go exactly to one place in cache



Fully associative

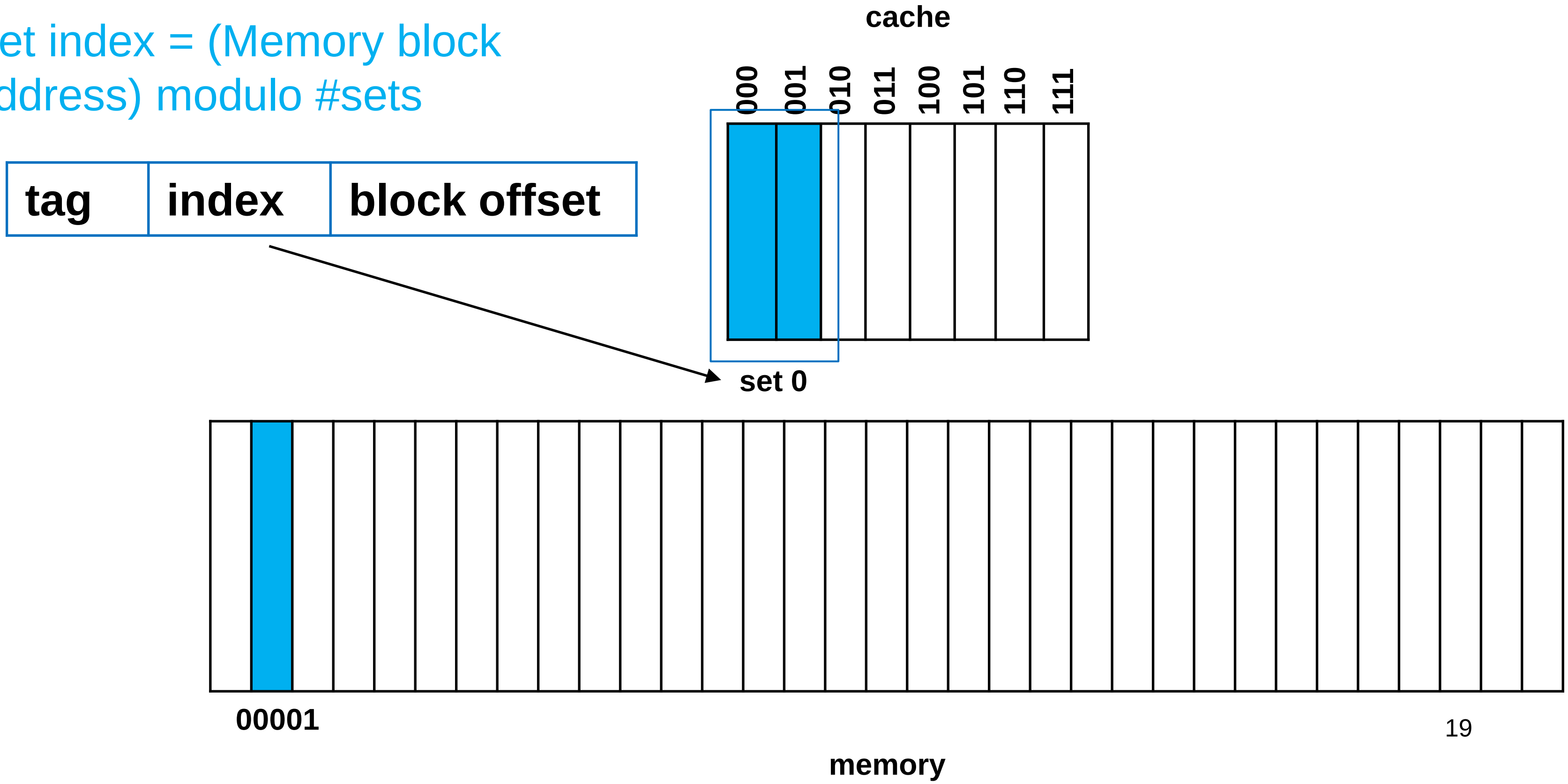
- A block in main memory can go to any block in cache
- Searched parallel



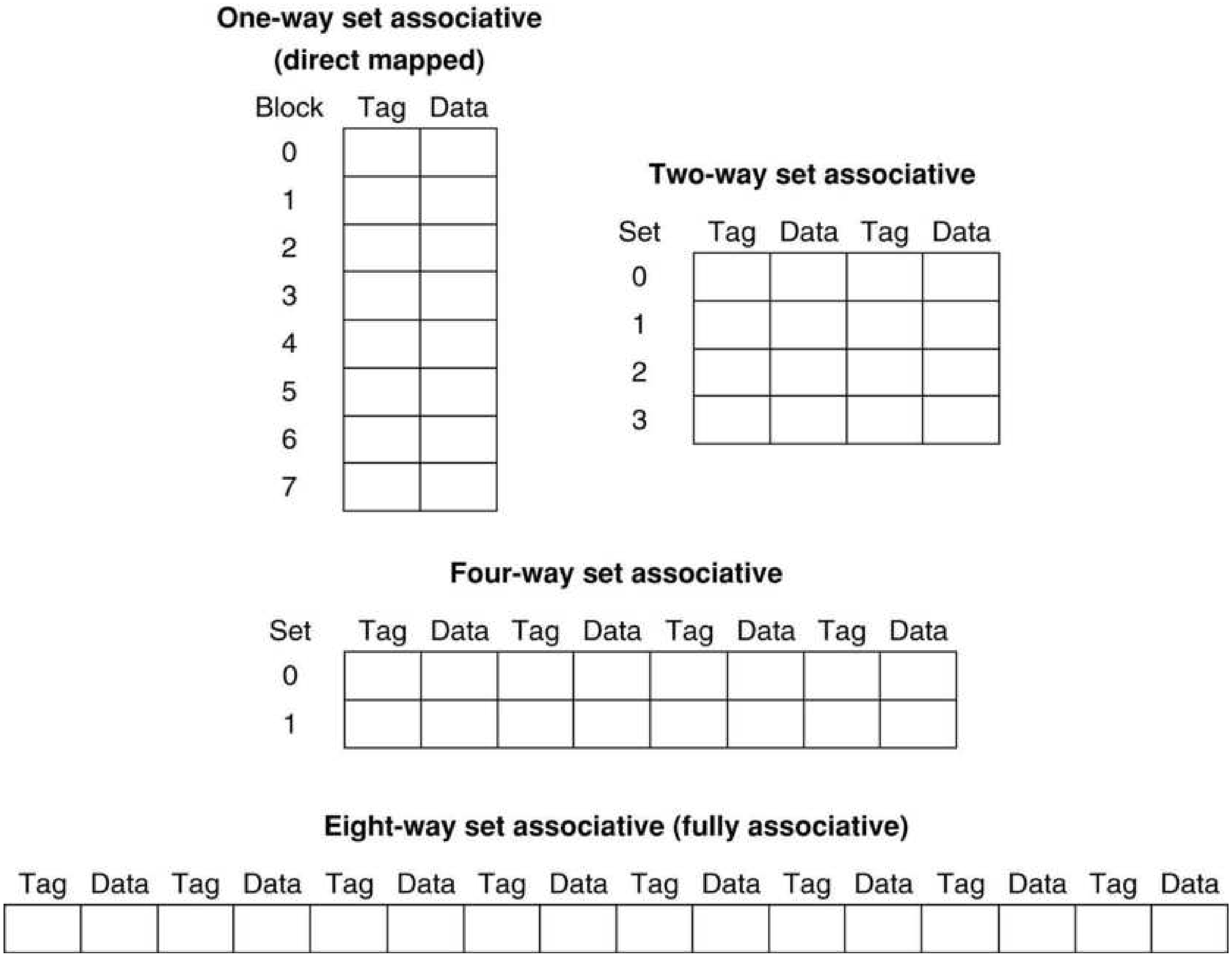
Set associative

- A block in main memory can go to any block in a set of cache blocks
- *n-way set associative*: *n* blocks for a set

Set index = (Memory block address) modulo #sets



Cache configuration examples



One-way set associative
(direct mapped)

Block	Tag	Data
0		
1		
2		
3		
4		
5		
6		
7		

Two-way set associative

Set	Tag	Data	Tag	Data
0				
1				
2				
3				

Four-way set associative

Set	Tag	Data	Tag	Data	Tag	Data	Tag	Data
0								
1								

Eight-way set associative (fully associative)

Tag	Data	Tag	Data	Tag	Data	Tag	Data	Tag	Data	Tag	Data	Tag	Data	Tag	Data

Cache Replacement Policies

- When cache is full some of the cached blocks need to be removed before bringing new ones in
  - If cached blocks are dirty (written/updated), then they need to be written to RAM
- Cache replacement policies
  - Random
  - Least Recently Used (LRU)
    - Need to track last access time
  - Least Frequently Used (LFU)
    - Need to track no of accesses
  - First In First Out (FIFO)

Increasing Cache Performance

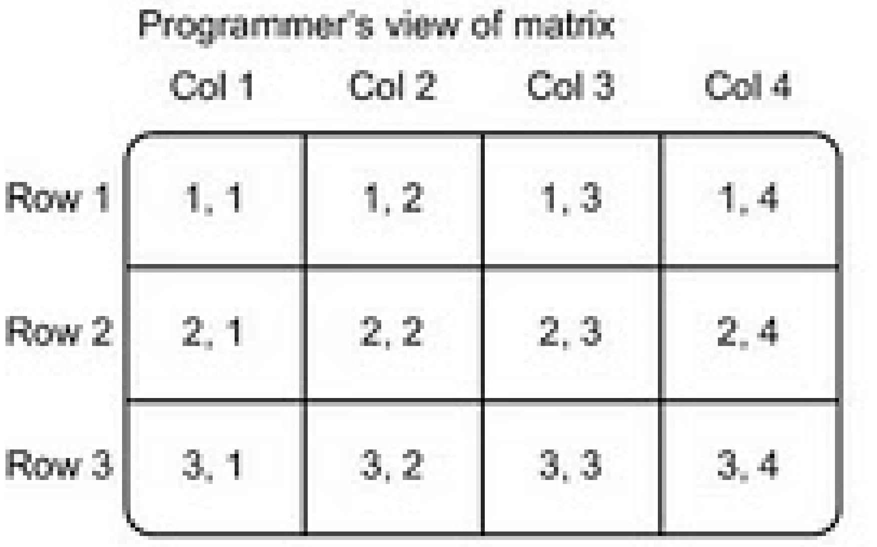
- Large cache capacity
- Multiple-levels of cache
- Prefetching
  - a block of data is brought into the cache before it is actually referenced
- Fully associative cache

Prefetching – Example

- Which of the following code is faster?

```
sum = 0;
for (i = 0; i < n; i++)
  for (j = 0; j < m; j++)
     sum += a[i][j];
return sum;

sum = 0;
for (j = 0; j < m; j++)
  for (i = 0; i < n; i++)
     sum += a[i][j];
return sum;
```



Programmer's view of matrix

	Col 1	Col 2	Col 3	Col 4
Row 1	1, 1	1, 2	1, 3	1, 4
Row 2	2, 1	2, 2	2, 3	2, 4
Row 3	3, 1	3, 2	3, 3	3, 4

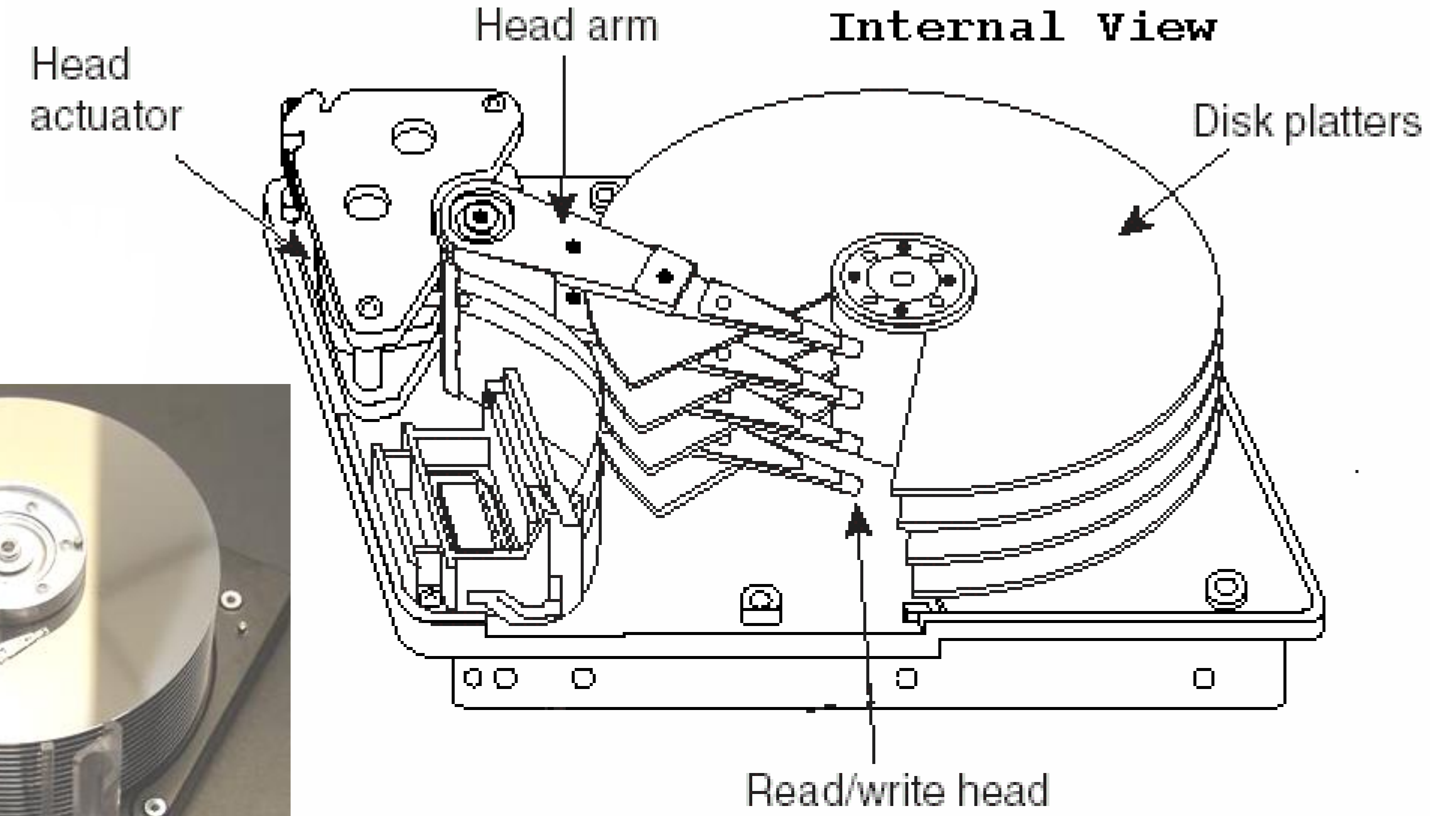
1,1	1,2	1,3	1,4	2,1	2,2	2,3	2,4	3,1	…		

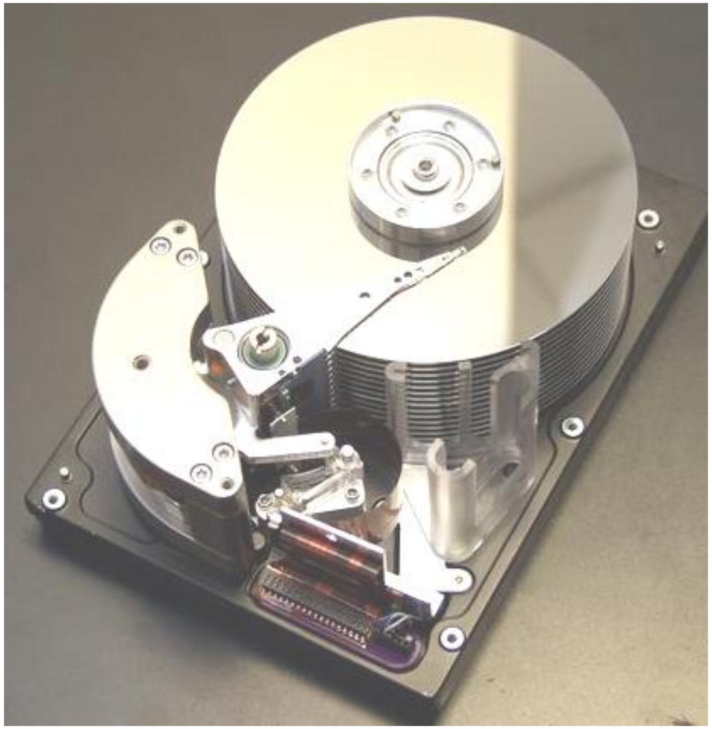
Intel Pentium 4 vs. AMD Opteron Memory Hierarchy

CPU	Pentium 4 (3.2 GHz)	Opteron (2.8 GHz)
Instruction Cache	Trace Cache 8K micro-ops	2-way associative, 64 KB, 64B block
Data Cache	8-way associative, 16 KB, 64B block, inclusive in L2	2-way associative, 64 KB, 64B block, exclusive to L2
L2 Cache	8-way associative, 2 MB, 128B block	16-way associative, 1 MB, 64B block
Prefetch	8 streams to L2	1 stream to L2
Memory	200 MHz x 64 bits	200 MHz x 128 bits

Permanent Storage

Permanent Storage – Hard Disks

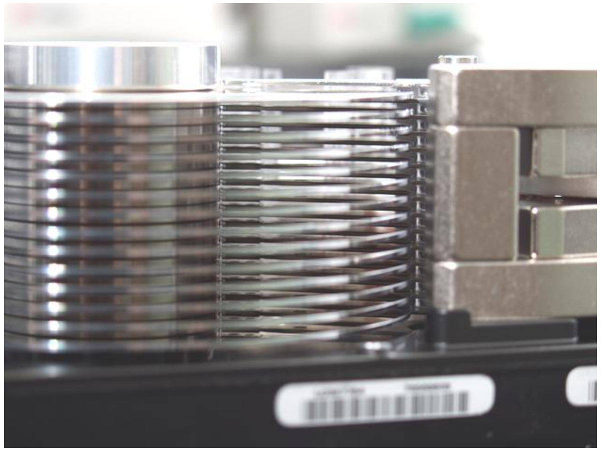


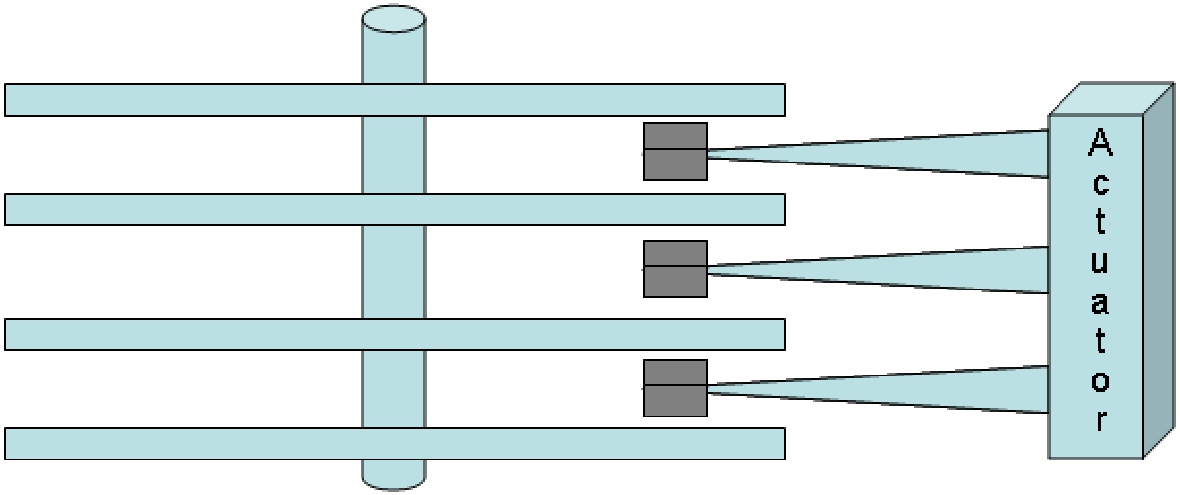


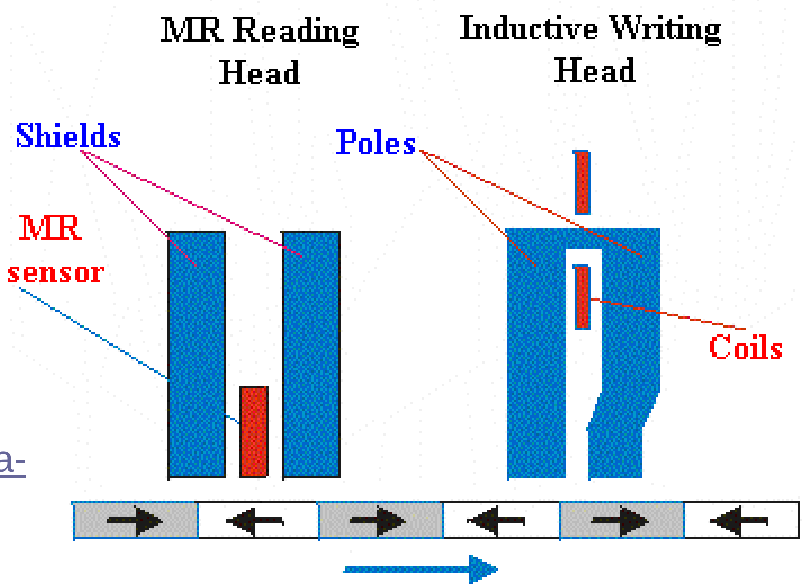
- Rigid disk (aluminum or glass) with a magnetic coating

Source: Upgrading & Repairing a PC by Scott Mueller

Hard Disks (Cont.)

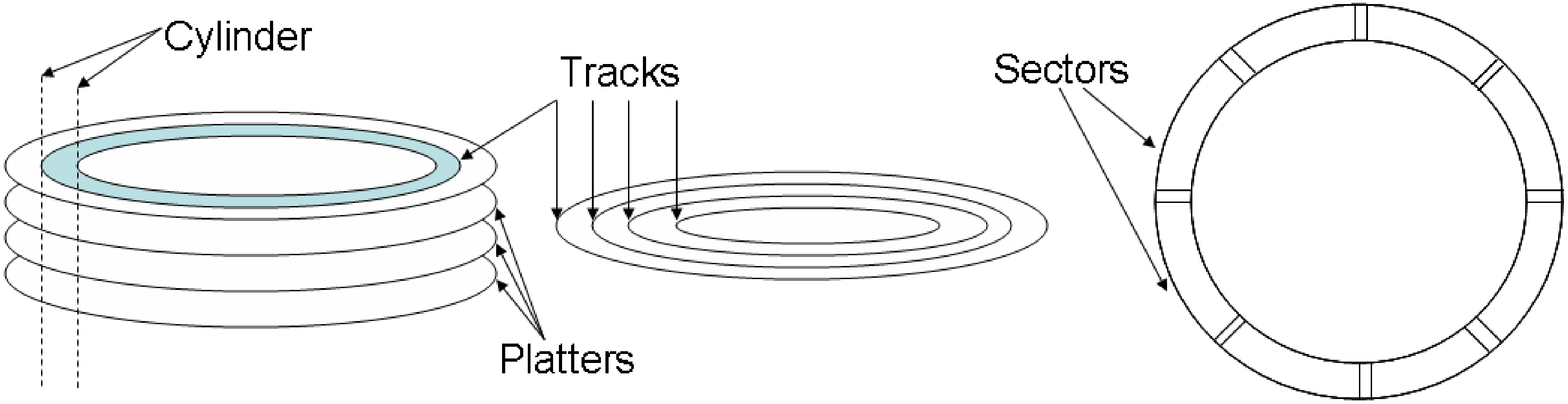






Source: www.dataclinic.it/data-recovery/hard-disk-functionality.htm

Cylinders, Tracks, & Sectors



- Data access time depends on
  - Seek time – time to position head over desired track
  - Rotational latency – time till desire sector is under the head
  - Read time – actual time to read data

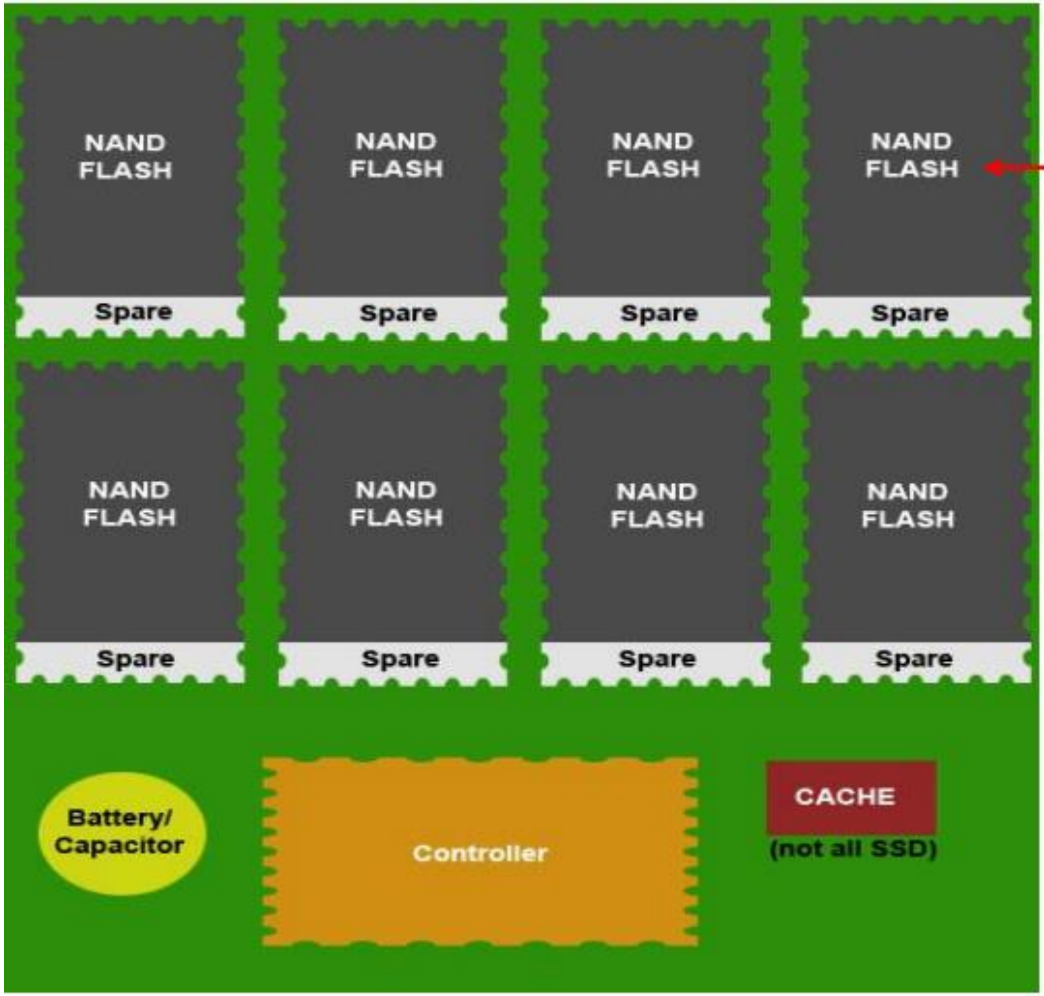
Classification of Hard Disks

- Disk capacity
  - 250 MB, 1GB, 60GB, 500GB, 1TB, 2TB
- Type of controller
  - IDE, SCSI, SATA, eSATA, SAS
- Speed (rpm)
  - 3600, 5,400, 7,200, 10,000, 15,000

Solid State Drive (SSD)

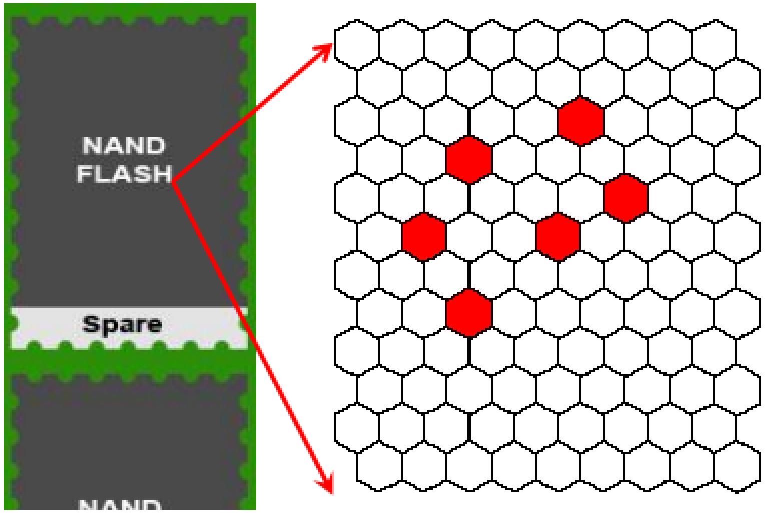
- No moving parts
- Permanent storage
  - Based on flash memory technologies
- High speed, large capacity, & robust
- More expensive
- Used in tablets, thin laptops, laptops

SSD Layout



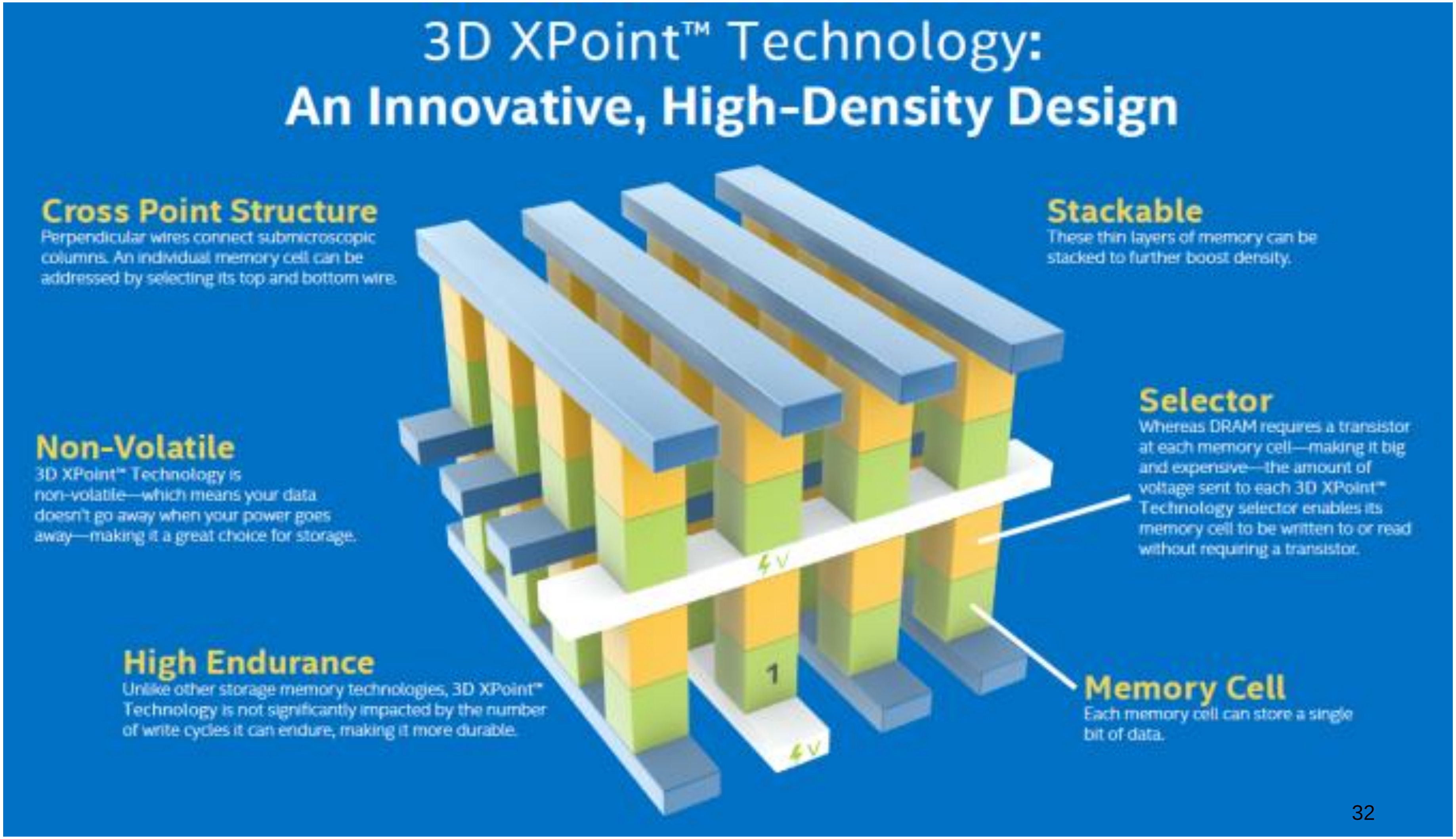
Each chip is usually 2GB to 128GB

Non volatile memory



Source: www.blog.solidstatediskshop.com/2012/how-does-an-ssd-write-part-2/

Intel Optane (Cont.)



3D XPoint™ Technology:

An Innovative, High-Density Design

Cross Point Structure
Perpendicular wires connect submicroscopic columns. An individual memory cell can be addressed by selecting its top and bottom wire.

Stackable
These thin layers of memory can be stacked to further boost density.

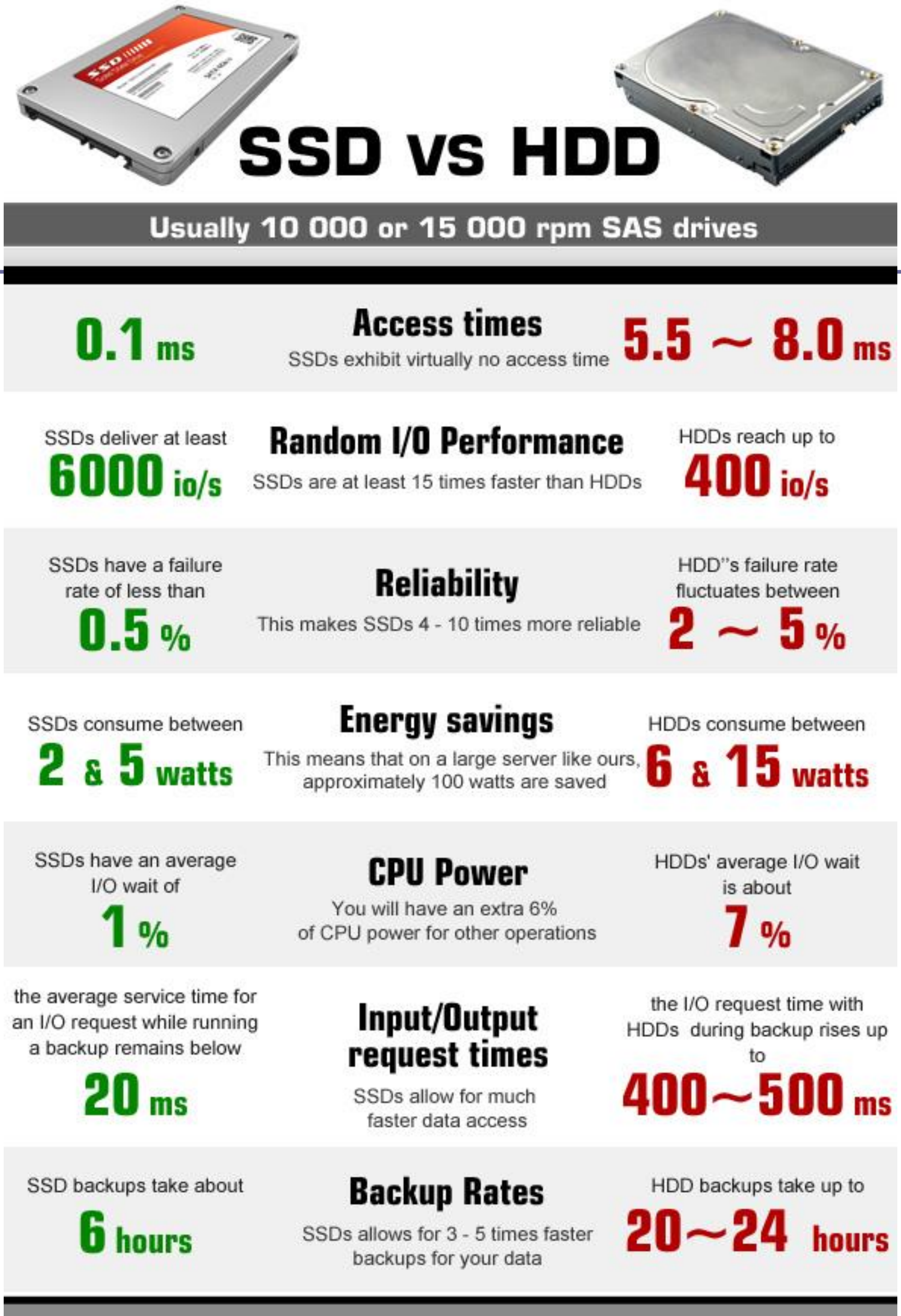
Non-Volatile
3D XPoint™ Technology is non-volatile—which means your data doesn't go away when your power goes away—making it a great choice for storage.

Selector
Whereas DRAM requires a transistor at each memory cell—making it big and expensive—the amount of voltage sent to each 3D XPoint™ Technology selector enables its memory cell to be written to or read without requiring a transistor.

High Endurance
Unlike other storage memory technologies, 3D XPoint™ Technology is not significantly impacted by the number of write cycles it can endure, making it more durable.

Memory Cell
Each memory cell can store a single bit of data.

Source: www.anandtech.com/show/9541/intel-announces-optane-storage-brand-for-3d-xpoint-products



SSD vs HDD

Usually 10 000 or 15 000 rpm SAS drives

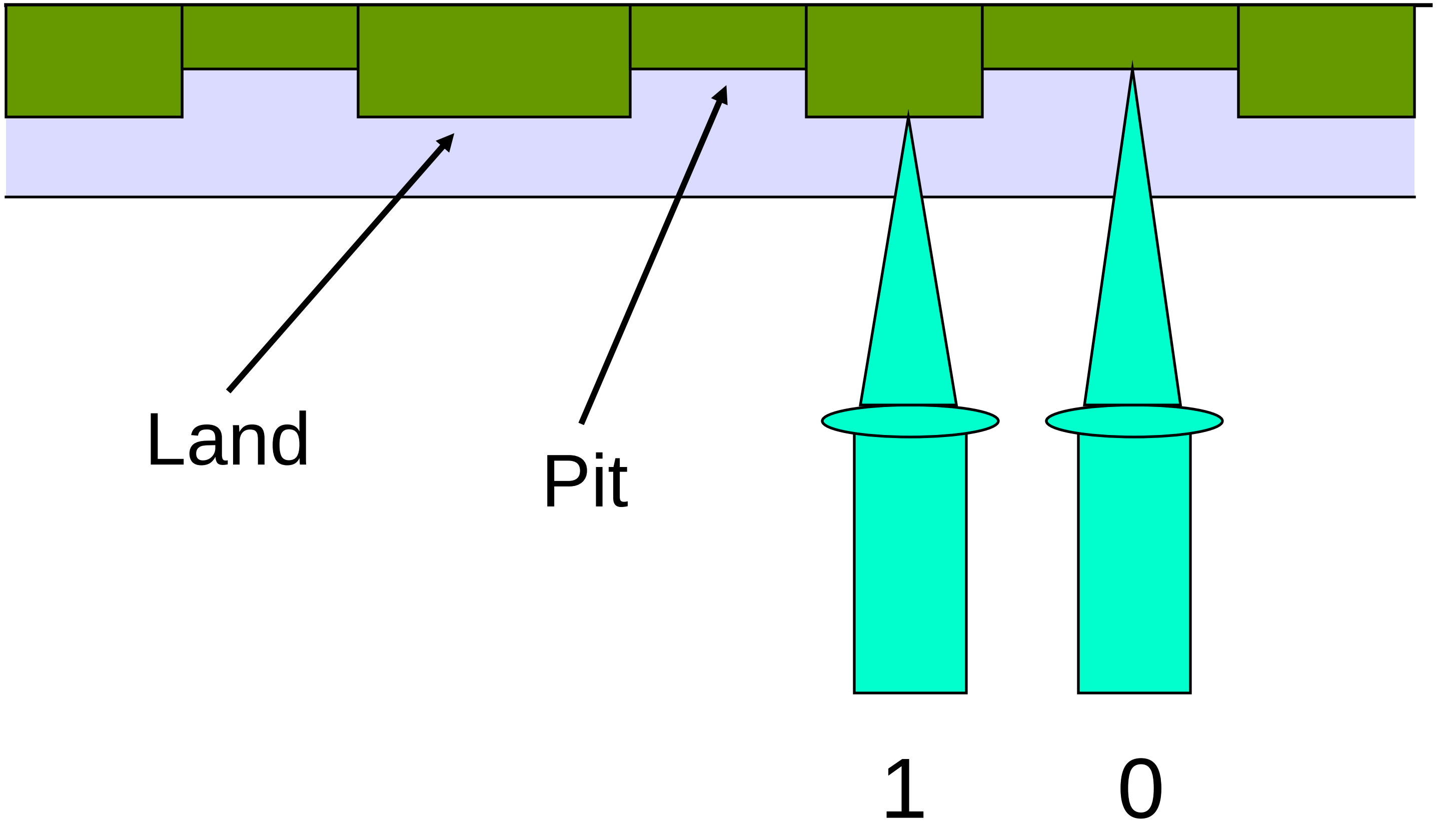
0.1 ms	**Access times** SSDs exhibit virtually no access time	5.5 ~ 8.0 ms
SSDs deliver at least 6000 io/s	**Random I/O Performance** SSDs are at least 15 times faster than HDDs	HDDs reach up to 400 io/s
SSDs have a failure rate of less than 0.5 %	**Reliability** This makes SSDs 4 - 10 times more reliable	HDD''s failure rate fluctuates between 2 ~ 5 %
SSDs consume between 2 & 5 watts	**Energy savings** This means that on a large server like ours, approximately 100 watts are saved	HDDs consume between 6 & 15 watts
SSDs have an average I/O wait of 1 %	**CPU Power** You will have an extra 6% of CPU power for other operations	HDDs' average I/O wait is about 7 %
the average service time for an I/O request while running a backup remains below 20 ms	**Input/Output request times** SSDs allow for much faster data access	the I/O request time with HDDs during backup rises up to 400~500 ms
SSD backups take about 6 hours	**Backup Rates** SSDs allows for 3 - 5 times faster backups for your data	HDD backups take up to 20~24 hours

Source:
www.resellerspanel.com

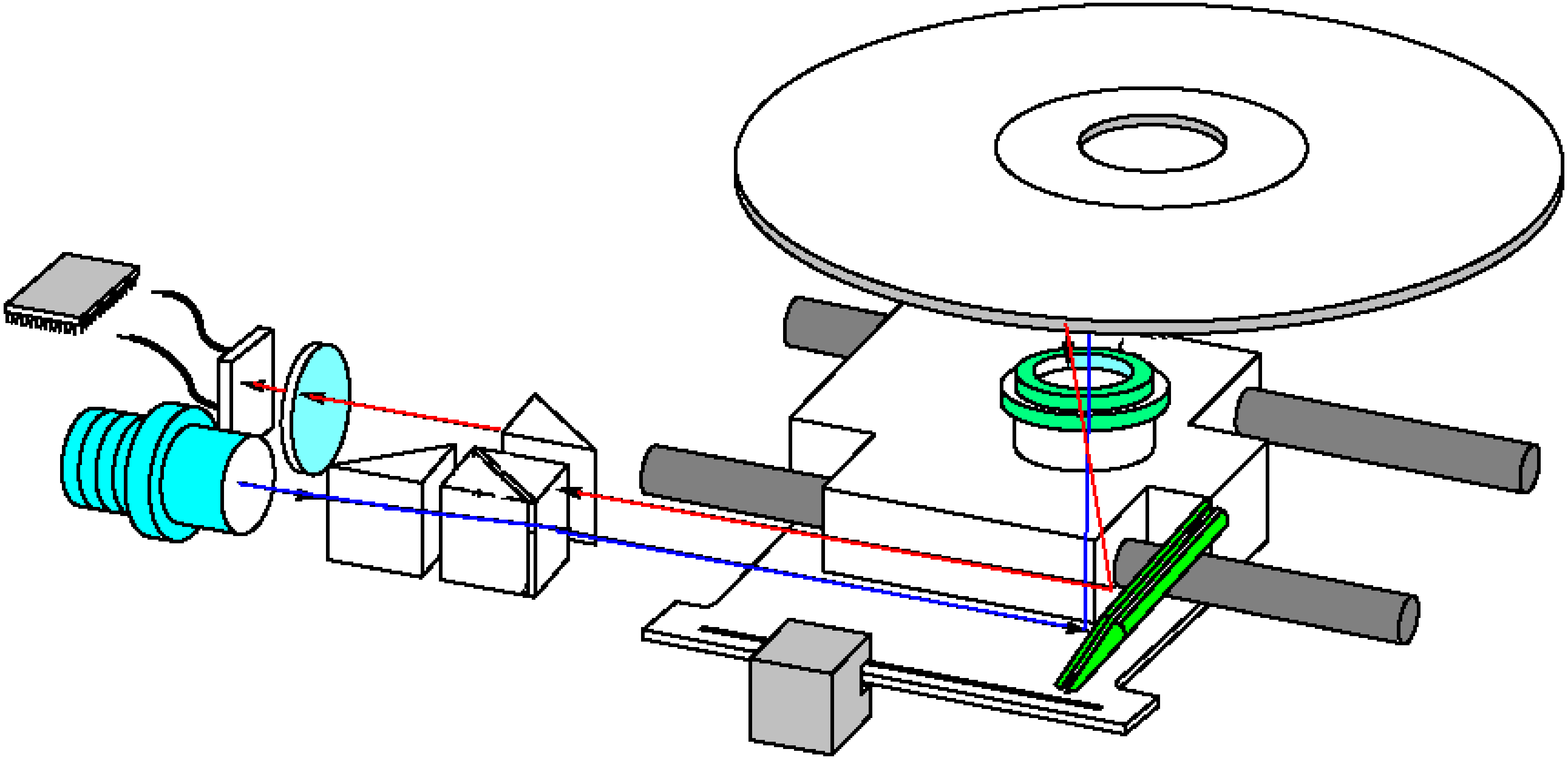
Optical Storage

- Make use of light instead of magnetism
- Different forms of optical storage
  - CD-ROM
  - CD-R – *Recordable*
  - CD-RW –*Rewritable*
  - DVD – Digital Versatile/Video Disk
  - DVD-R/DVD-RW
  - Blu-ray

Geometry of a CD



Components of a CD-ROM drive



Source: Upgrading & Repairing a PC by Scott Mueller

References

- Waterman, A., Lee, Y., Patterson, D. A., & Asanovi, K. (2014). *The risc-v instruction set manual. volume 1: User-level isa, version 2.0*. California Univ Berkeley Dept of Electrical Engineering and Computer Sciences.
- Harris, S. L., Chaver, D., Piñuel, L., Gomez-Perez, J. I., Liaqat, M. H., Kakakhel, Z. L., ... & Owen, R. (2021, August). RVfpga: Using a RISC-V Core Targeted to an FPGA in Computer Architecture Education. In *2021 31st International Conference on Field-Programmable Logic and Applications (FPL)* (pp. 145-150). IEEE.
- Hennessy, J. L., & Patterson, D. A. (2011). *Computer architecture: a quantitative approach*. Elsevier.
- Patterson, D. A., & Hennessy, J. L. (2016). *Computer organization and design ARM edition: the hardware software interface*. Morgan kaufmann.

THANK YOU