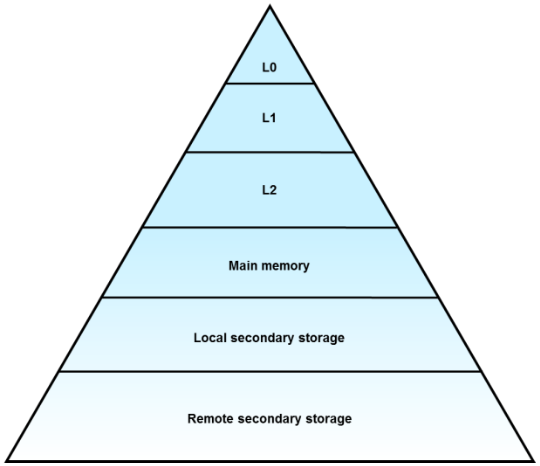
Accessing The Outside World

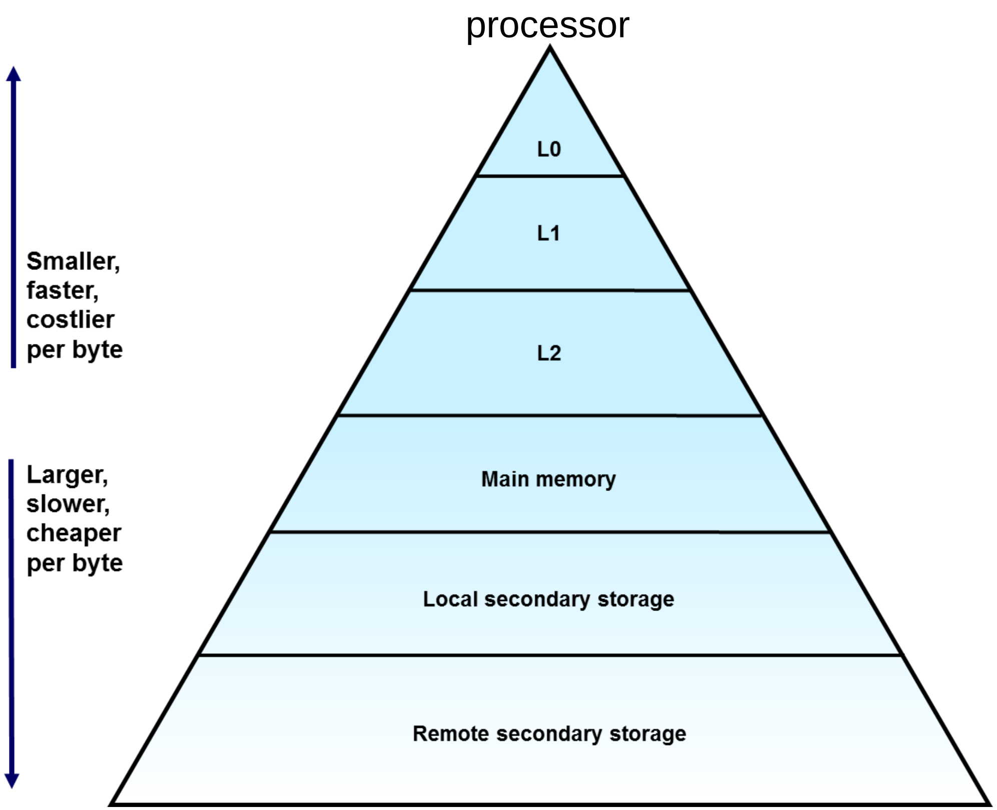
The Memory Hierarchy



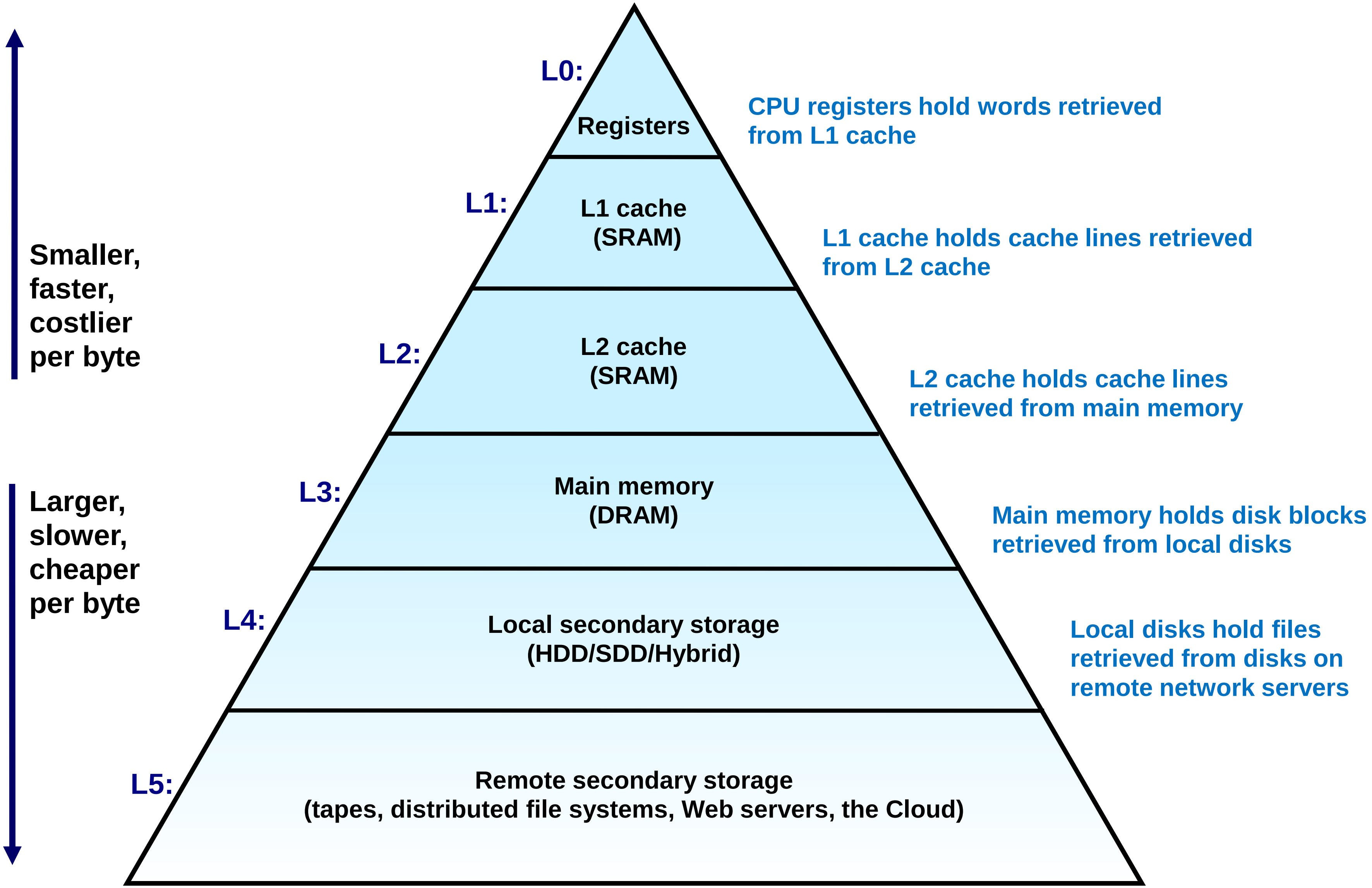
4190.308 Computer Architecture, Fall 2014

Memory Hierarchy

> Ideally one would desire an indefinitely large memory capacity such that any particular … word would be immediately available … We are … forced to recognize the possibility of constructing a hierarchy of memories, each of which has greater capacity than the preceding but which is less quickly accessible.
>
> Burks, Goldstine, and von Neumann,1946



An Example Memory Hierarchy


L0:
Registers
CPU registers hold words retrieved from L1 cache
L1:
L1 cache (SRAM)
L1 cache holds cache lines retrieved from L2 cache
L2:
L2 cache (SRAM)
L2 cache holds cache lines retrieved from main memory
L3:
Main memory (DRAM)
Main memory holds disk blocks retrieved from local disks
L4:
Local secondary storage (HDD/SDD/Hybrid)
Local disks hold files retrieved from disks on remote network servers
L5:
Remote secondary storage (tapes, distributed file systems, Web servers, the Cloud)
Smaller, faster, costlier per byte
Larger, slower, cheaper per byte

The Memory Hierarchy

- **Storage technologies and trends**
- Locality of reference
- Caching in the memory hierarchy

Acknowledgement: slides based on the cs:app2e material

Random-Access Memory (RAM)

- Key features
  - RAM is traditionally packaged as a chip.
  - Basic storage unit is normally a cell (one bit per cell).
  - Multiple RAM chips form a memory.

- Static RAM (SRAM)
  - Each cell stores a bit with a four or six-transistor circuit.
  - Retains value indefinitely, as long as it is kept powered.
  - Relatively insensitive to electrical noise (EMI), radiation, etc.
  - Faster and more expensive than DRAM.

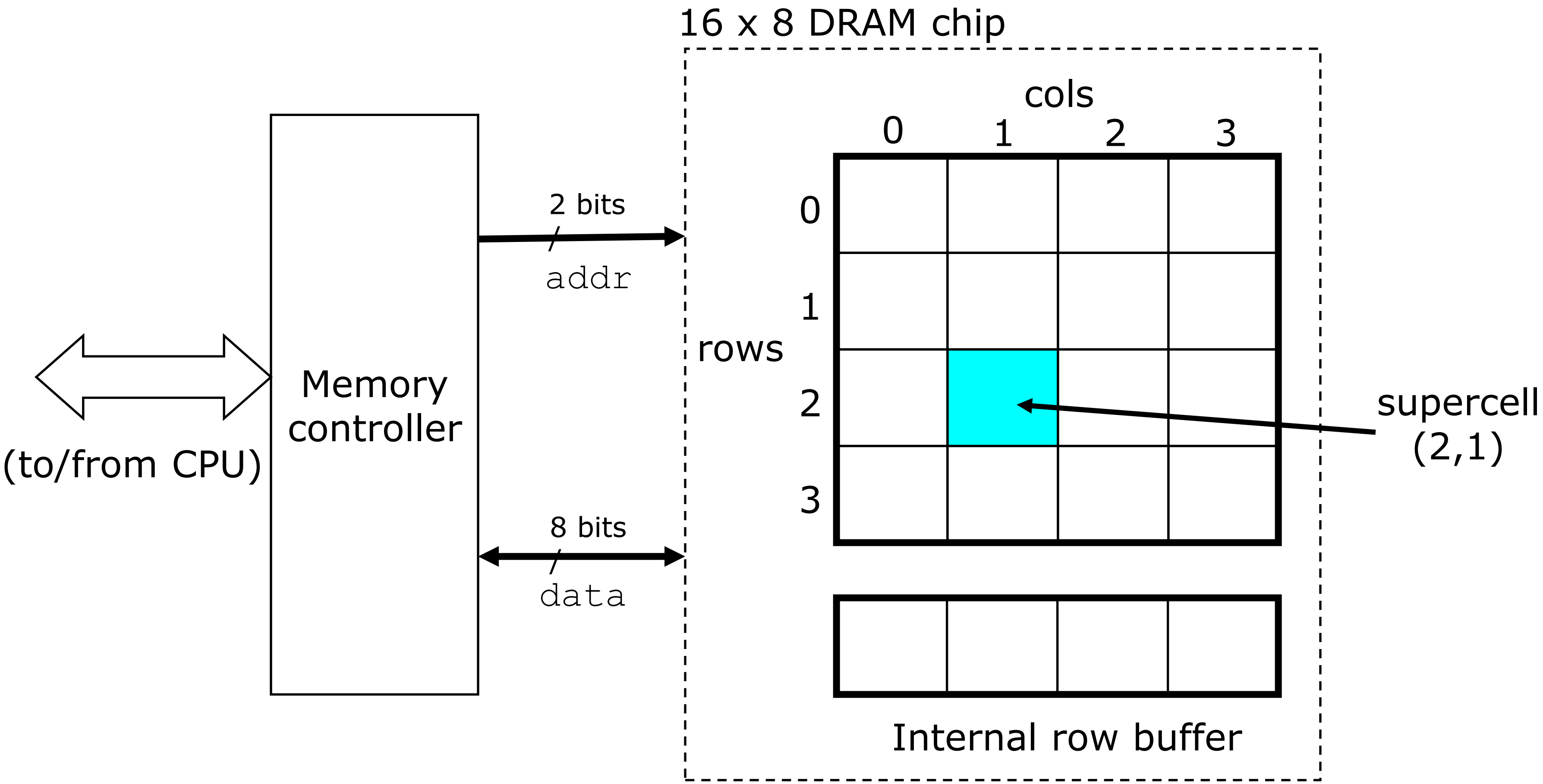
- Dynamic RAM (DRAM)
  - Each cell stores bit with a capacitor. One transistor is used for access
  - Value must be refreshed every 10-100 ms.
  - More sensitive to disturbances (EMI, radiation,…) than SRAM.
  - Slower and cheaper than SRAM.

SRAM vs DRAM Summary

	Trans. per bit	Access time	Needs refresh?	Needs EDC?	Cost	Applications
SRAM	4 or 6	1X	No	Maybe	100x	Cache memories
DRAM	1	10X	Yes	Yes	1X	Main memories, frame buffers

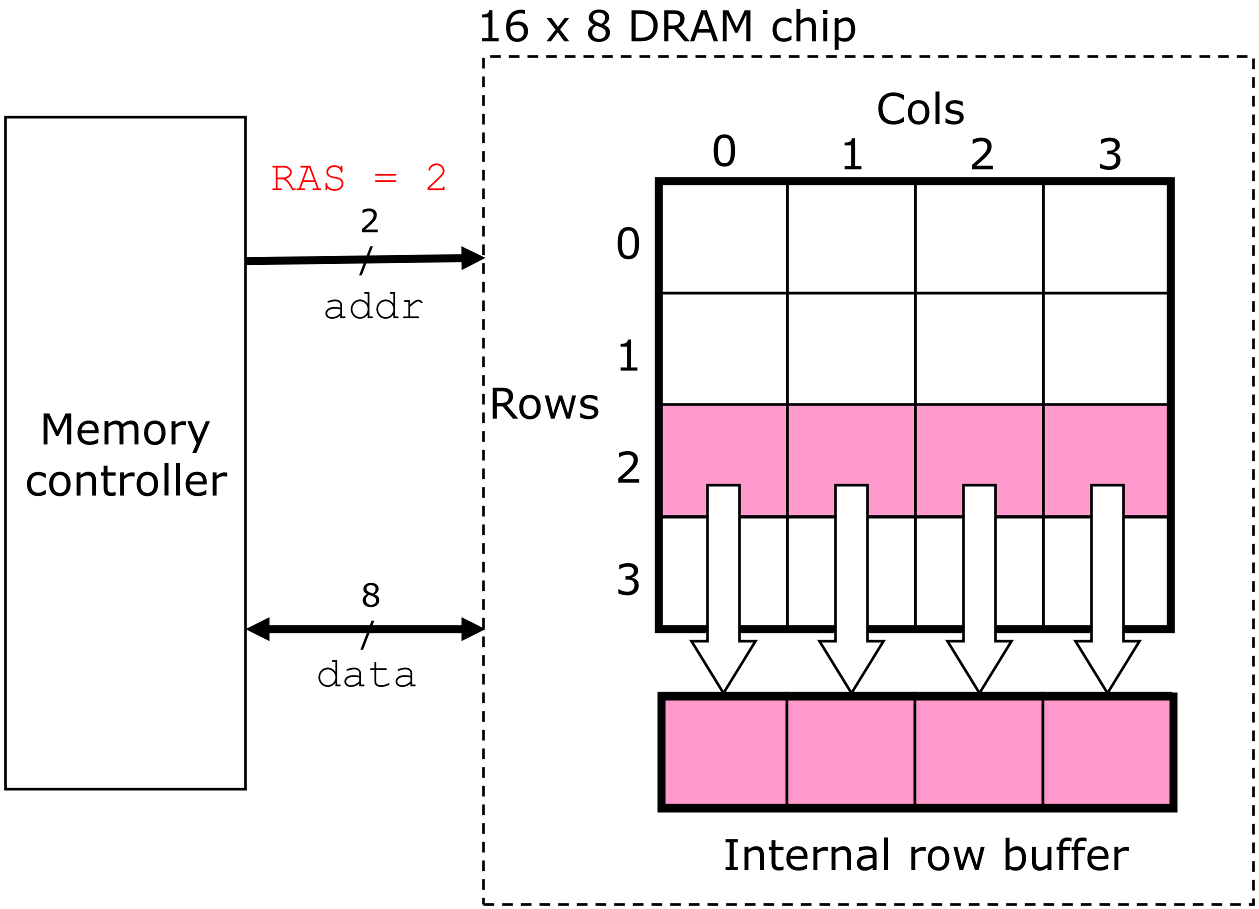
Conventional DRAM Organization

- d x w DRAM:
  - dw total bits organized as d supercells of size w bits



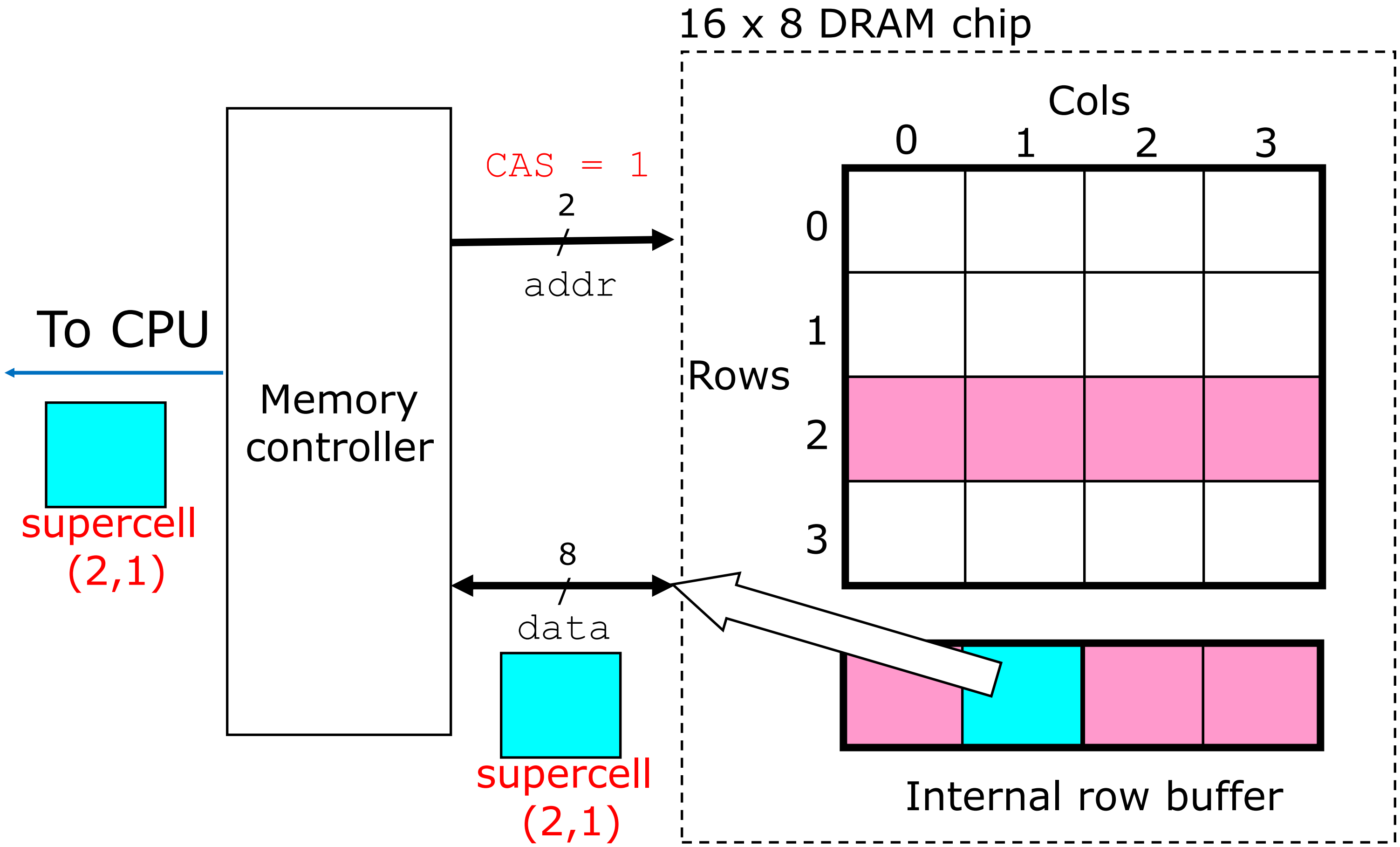
Reading DRAM Supercell (2,1)

- Step 1(a): Row access strobe (RAS) selects row 2.
- Step 1(b): Row 2 copied from DRAM array to row buffer.

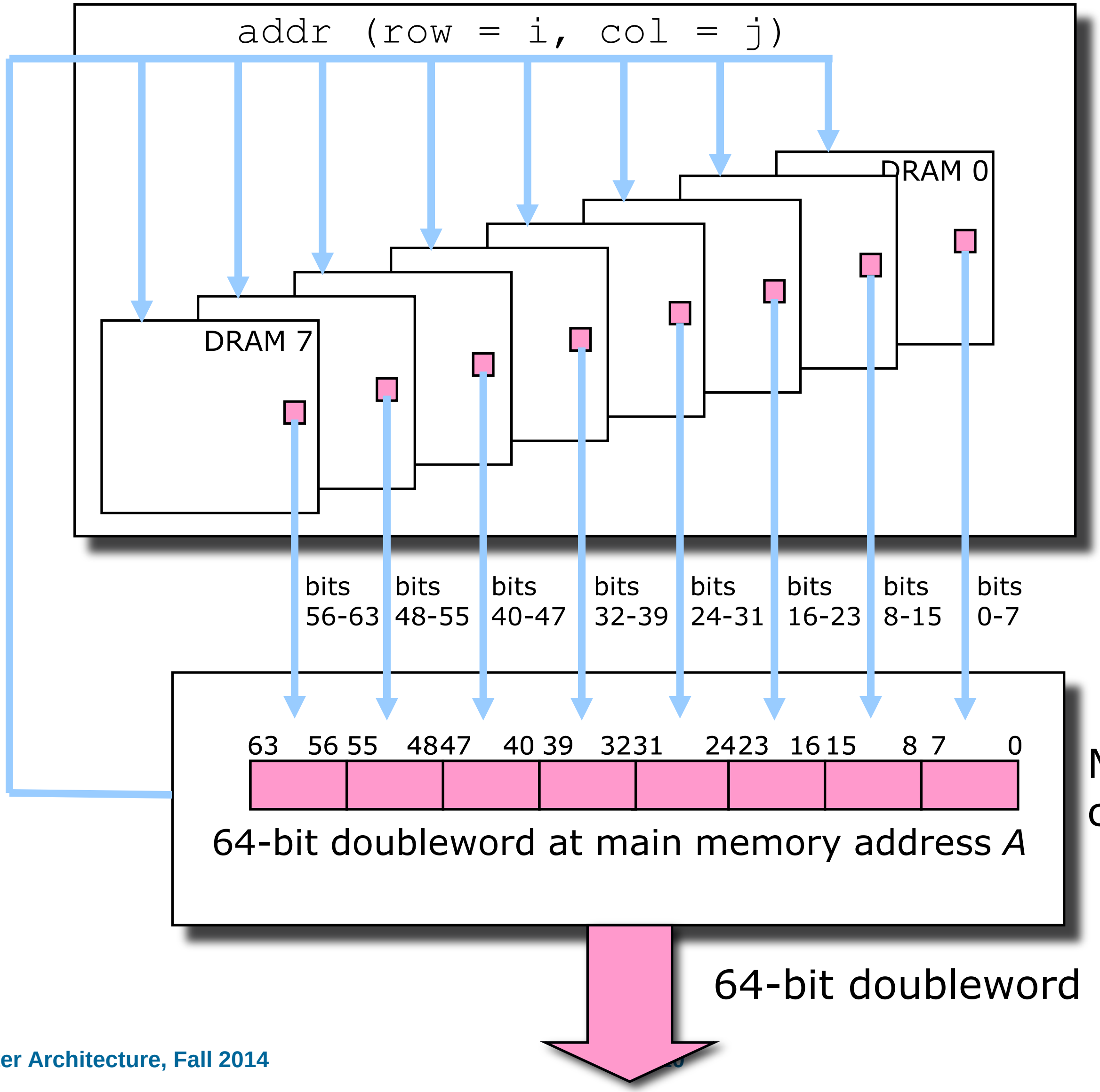


Reading DRAM Supercell (2,1)

- Step 2(a): Column access strobe (CAS) selects column 1.
- Step 2(b): Supercell (2,1) copied from buffer to data lines, and eventually back to the CPU.



Memory Modules



64-bit doubleword at main memory address *A*

: supercell (i,j)

64 MB memory module consisting of eight 8Mx8 DRAMs

Memory controller

64-bit doubleword

Enhanced DRAMs

- Basic DRAM cell has not changed since its invention in 1966.
  - Commercialized by Intel in 1970.

- DRAM cores with better interface logic and faster I/O :
  - Synchronous DRAM (SDRAM)
    - Uses a conventional clock signal instead of asynchronous control
    - Allows reuse of the row addresses (e.g., RAS, CAS, CAS, CAS)

  - Double data-rate synchronous DRAM (DDR SDRAM)
    - Double edge clocking sends two bits per cycle per pin
    - Different types distinguished by size of small prefetch buffer:
      - DDR (2 bits), DDR2 (4 bits), DDR4 (8 bits)
    - By 2010, standard for most server and desktop systems
    - Intel Core i7 supports only DDR3 SDRAM

Nonvolatile Memories

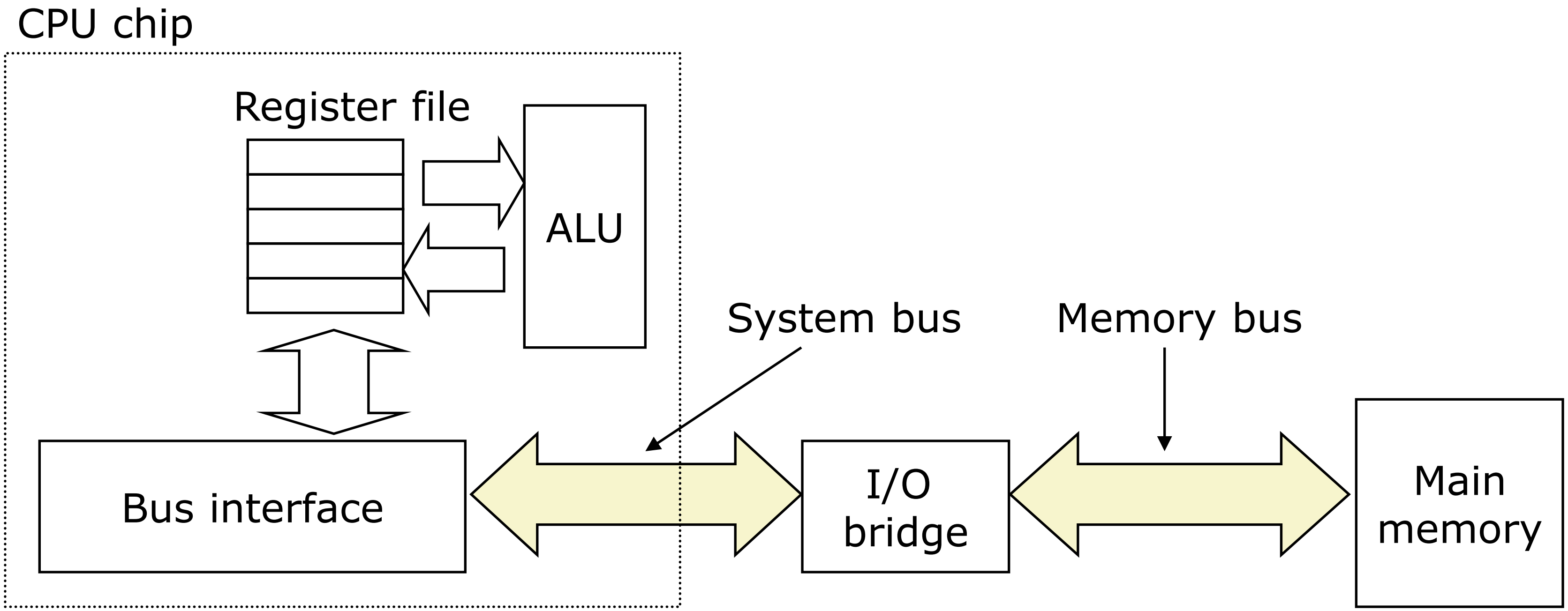
- DRAM and SRAM are volatile memories
  - Lose information if powered off.

- Nonvolatile memories retain value even if powered off
  - Read-only memory (ROM): programmed during production
  - Programmable ROM (PROM): can be programmed once
  - Eraseable PROM (EPROM): can be bulk erased (UV, X-Ray)
  - Electrically eraseable PROM (EEPROM): electronic erase capability
  - Flash memory: EEPROMs with partial (sector) erase capability
    - Wears out after about 100,000 erasings.

- Uses for Nonvolatile Memories
  - Firmware programs stored in a ROM (BIOS, controllers for disks, network cards, graphics accelerators, security subsystems,…)
  - Solid state disks (replace rotating disks in thumb drives, smart phones, mp3 players, tablets, laptops,…)
  - Disk caches

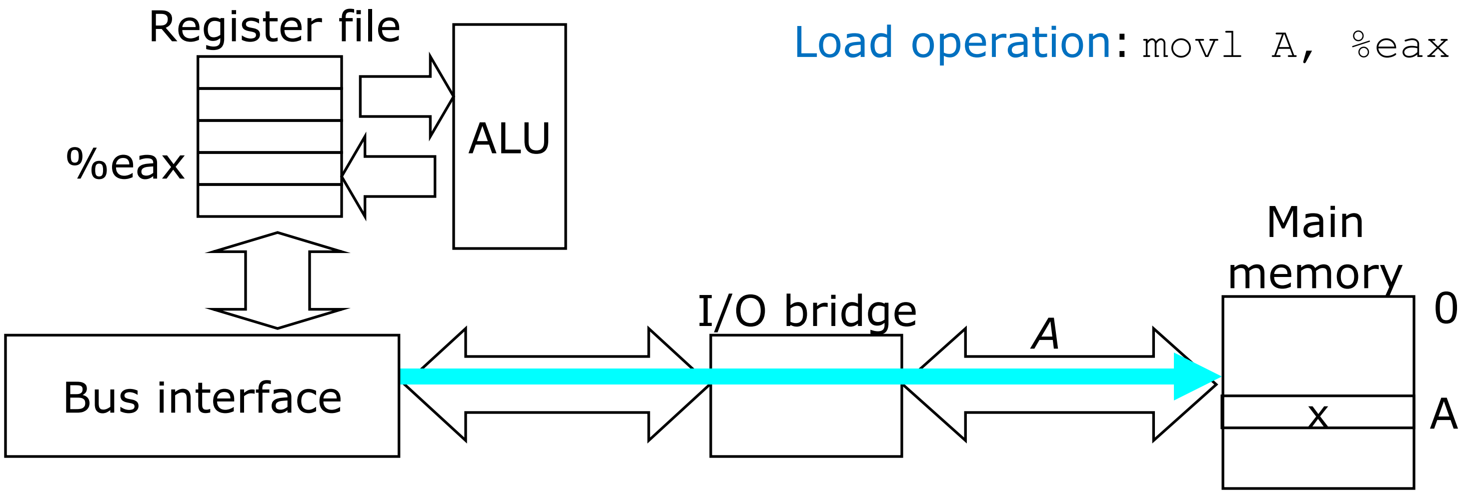
Traditional Bus Structure Connecting CPU and Memory

- A bus is a collection of parallel wires that carry address, data, and control signals.
- Buses are typically shared by multiple devices.



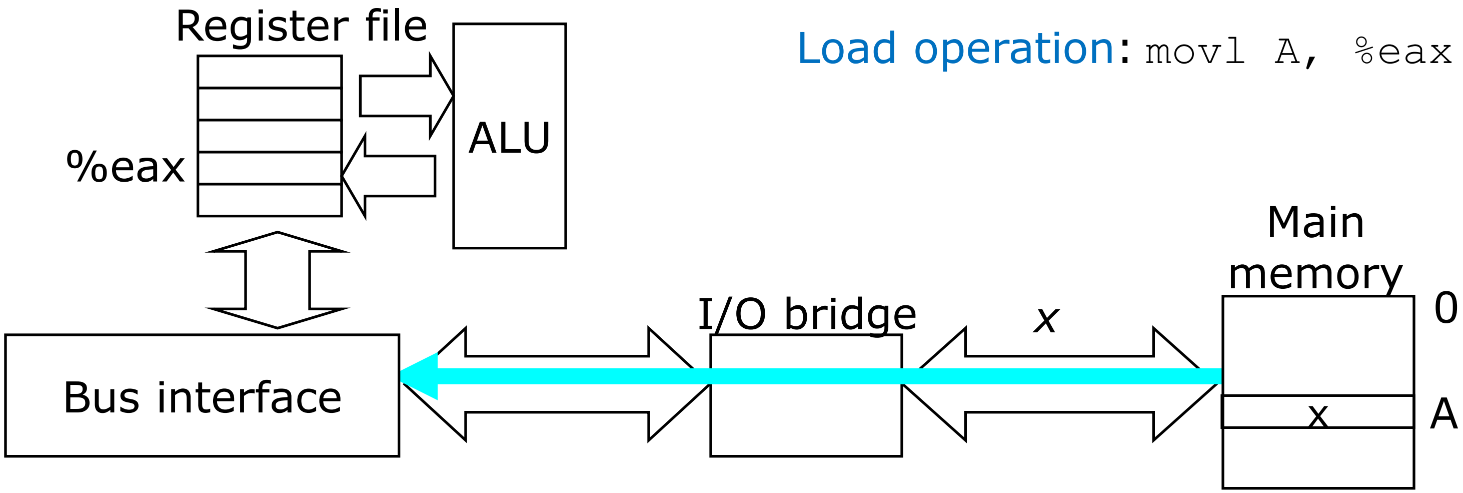
Memory Read Transaction (1)

- CPU places address A on the memory bus.



Memory Read Transaction (2)

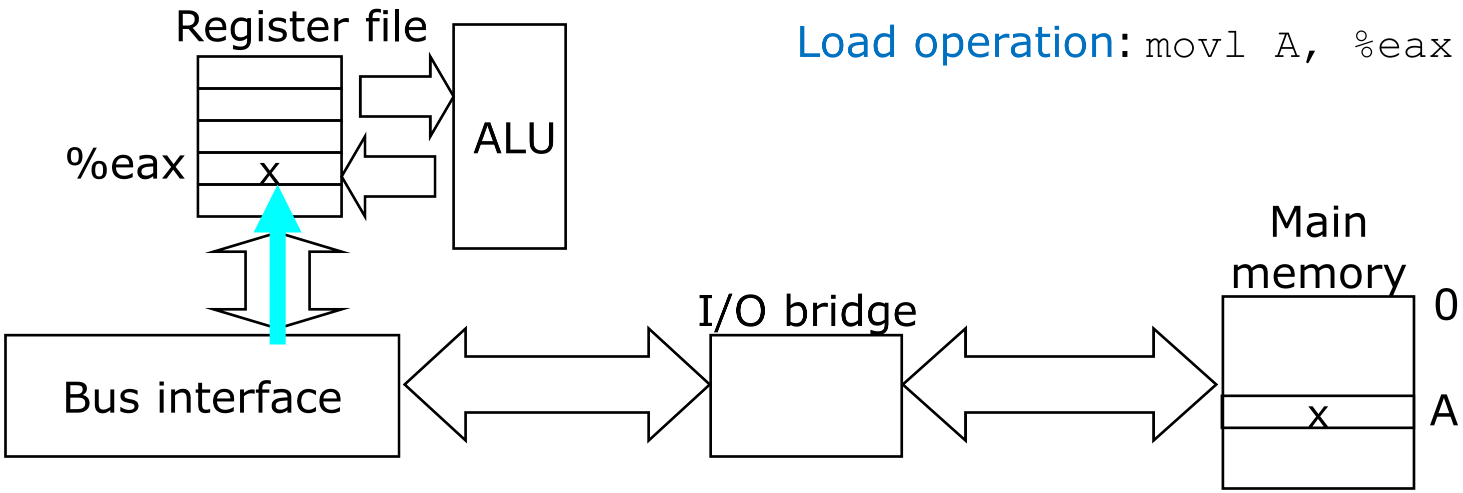
- Main memory reads A from the memory bus, retrieves word x, and places it on the bus.



Load operation: `movl A, %eax`

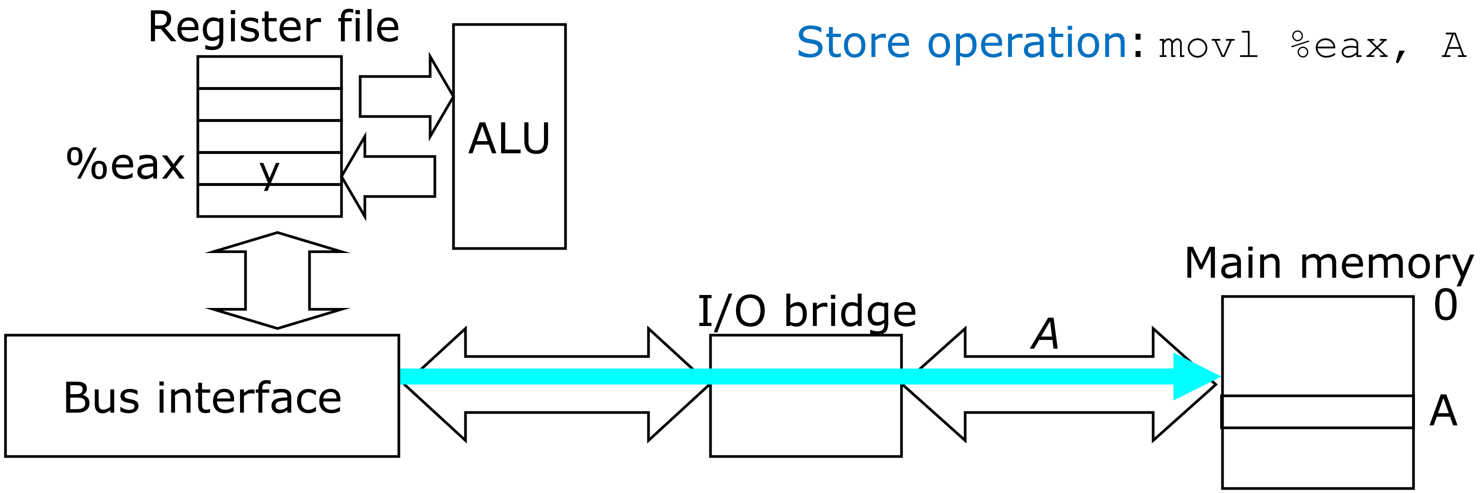
Memory Read Transaction (3)

- CPU read word x from the bus and copies it into register %eax.



Memory Write Transaction (1)

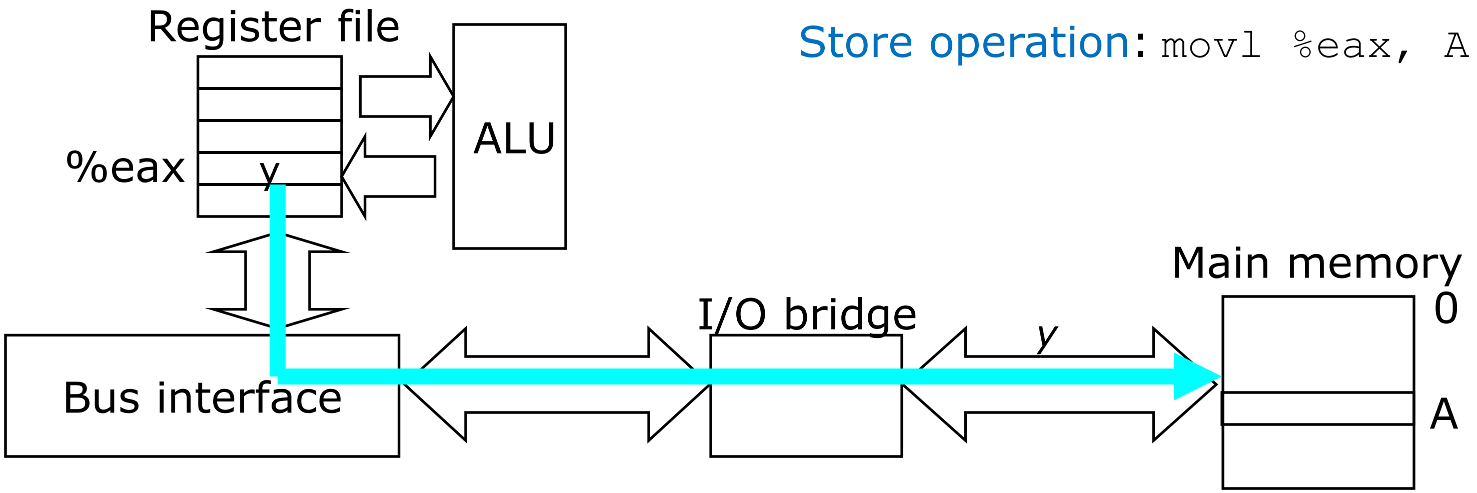
- CPU places address A on bus. Main memory reads it and waits for the corresponding data word to arrive.



Store operation: `movl %eax, A`

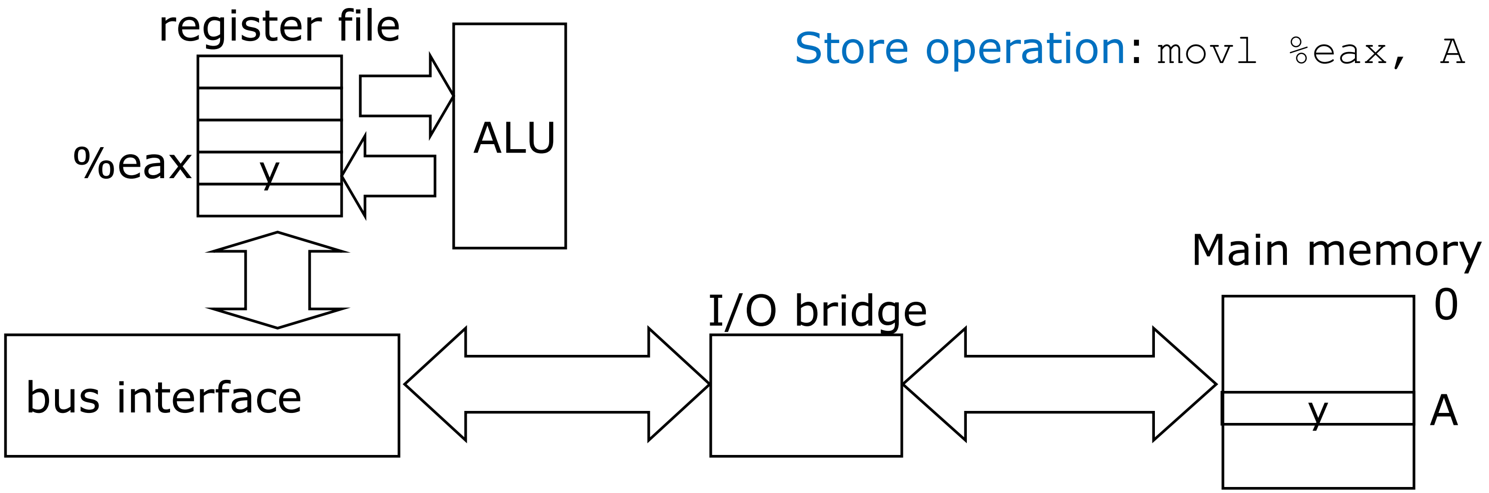
Memory Write Transaction (2)

- CPU places data word y on the bus.

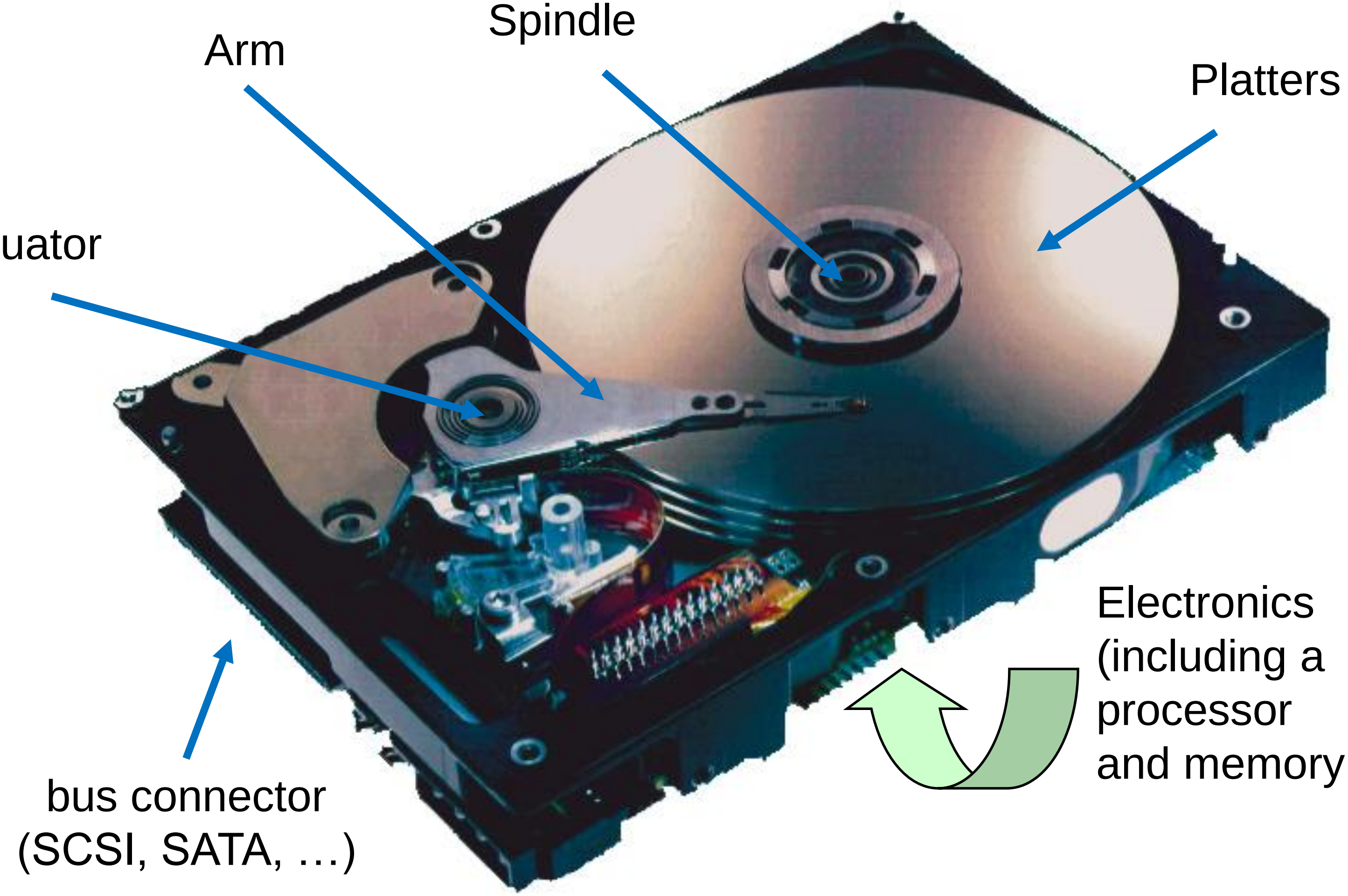


Memory Write Transaction (3)

- Main memory reads data word y from the bus and stores it at address A.



What’s Inside A Disk Drive?



Arm

Spindle

Platters

Actuator

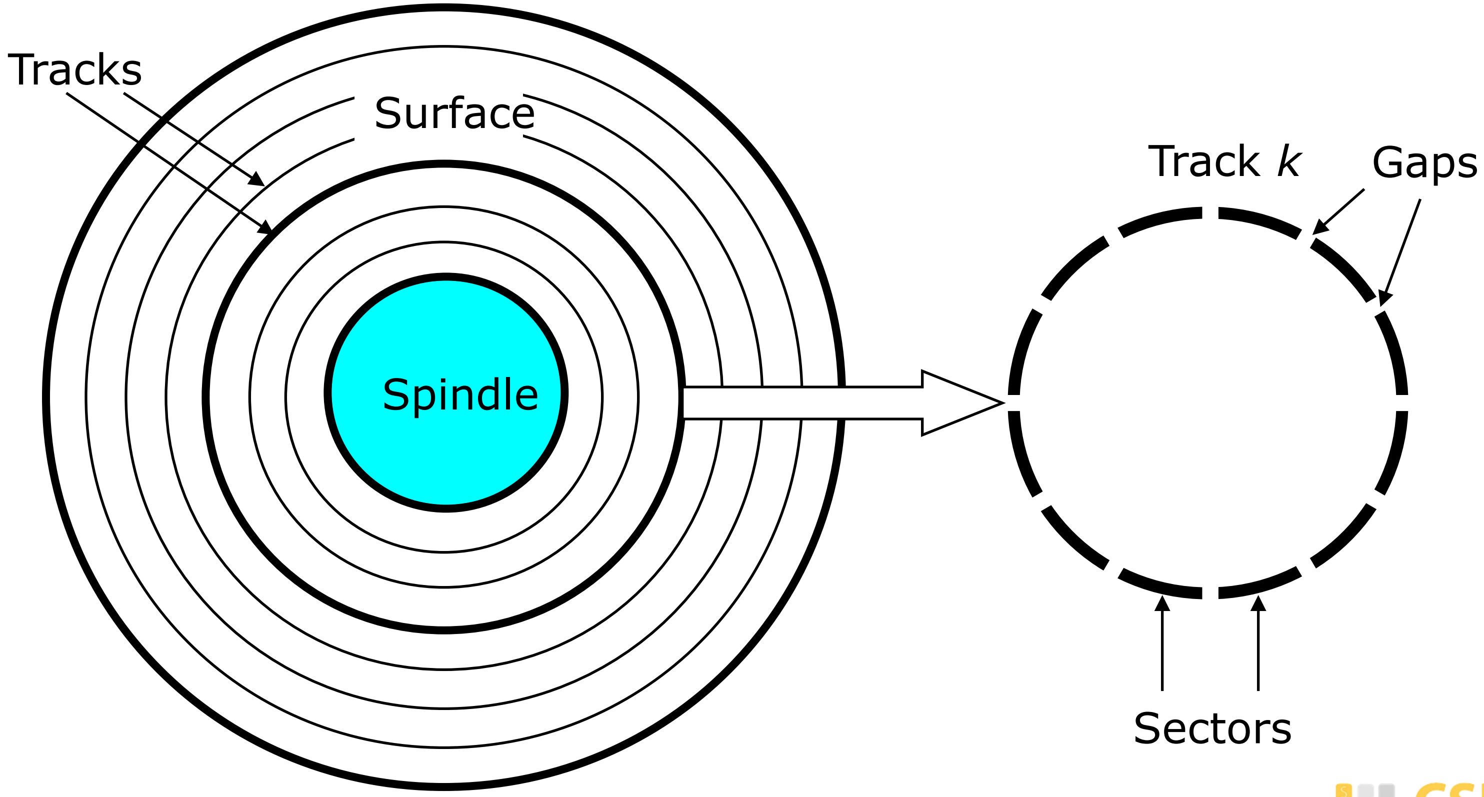
Electronics (including a processor and memory!)

bus connector (SCSI, SATA, …)

Image courtesy of Seagate Technology

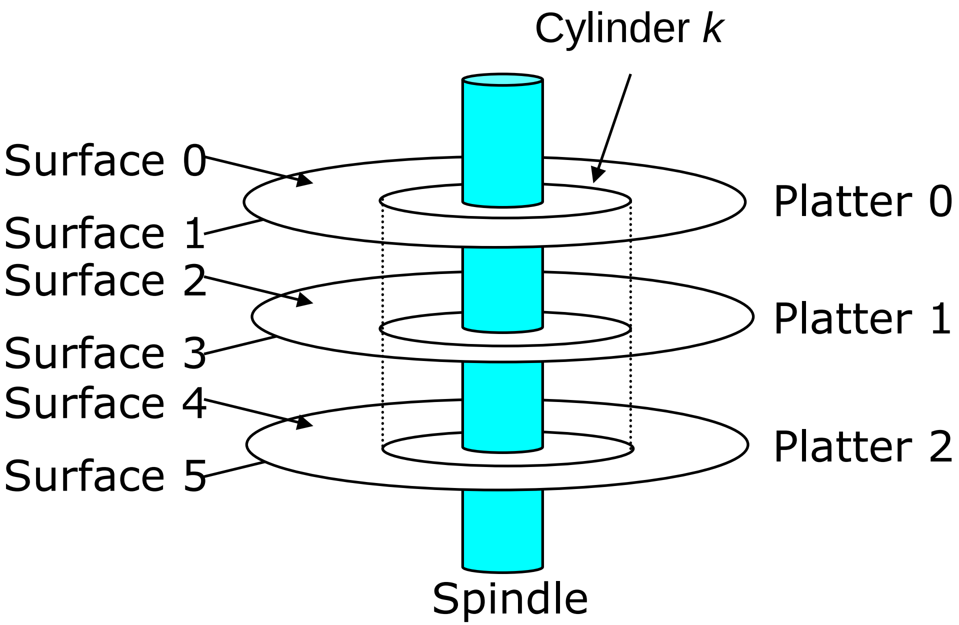
Disk Geometry

- Disks consist of platters, each with two surfaces.
- Each surface consists of concentric rings called tracks.
- Each track consists of sectors separated by gaps.
- Each sector contains an equal number of data bits (typically 512 bytes)



Disk Geometry (Muliple-Platter View)

- Aligned tracks form a cylinder.



Disk Capacity

- Capacity: maximum number of bits that can be stored.
  - Vendors express capacity in units of gigabytes (GB), where 1 GB = 10^9 Bytes

- Capacity is determined by these technology factors:
  - Recording density (bits/in): number of bits that can be squeezed into a 1 inch segment of a track.
  - Track density (tracks/in): number of tracks that can be squeezed into a 1 inch radial segment.
  - Areal density (bits/in2): product of recording and track density.

- Modern disks partition tracks into disjoint subsets called recording zones
  - Each track in a zone has the same number of sectors, determined by the circumference of innermost track.
  - Each zone has a different number of sectors/track

Computing Disk Capacity

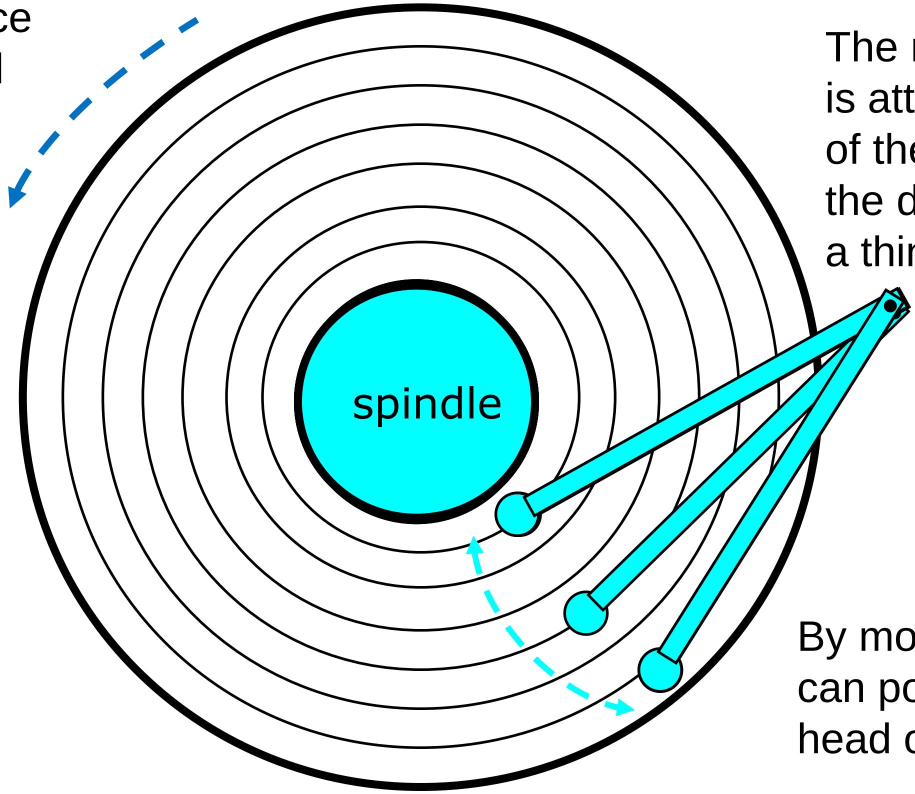
- Capacity = (# bytes/sector) x (avg. # sectors/track) x
  (# tracks/surface) x (# surfaces/platter) x
  (# platters/disk)

- Example:
  - 512 bytes/sector
  - 300 sectors/track (on average)
  - 20,000 tracks/surface
  - 2 surfaces/platter
  - 5 platters/disk

- Capacity = 512 x 300 x 20000 x 2 x 5
  = 30,720,000,000
  = 30.72 GB

Disk Operation (Single-Platter View)

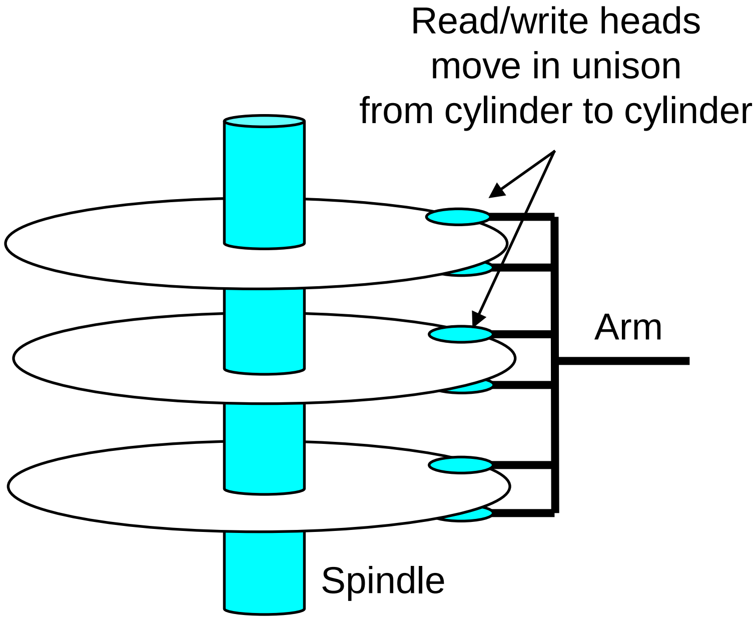
The disk surface
spins at a fixed
rotational rate



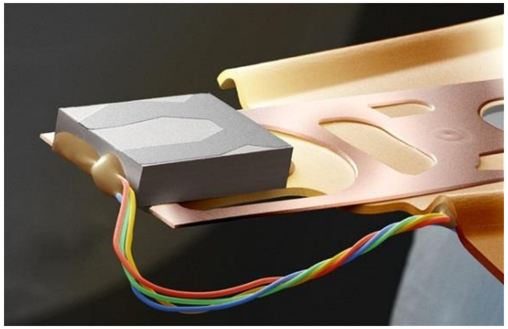
The read/write *head*
is attached to the end
of the *arm* and flies over
the disk surface on
a thin cushion of air.

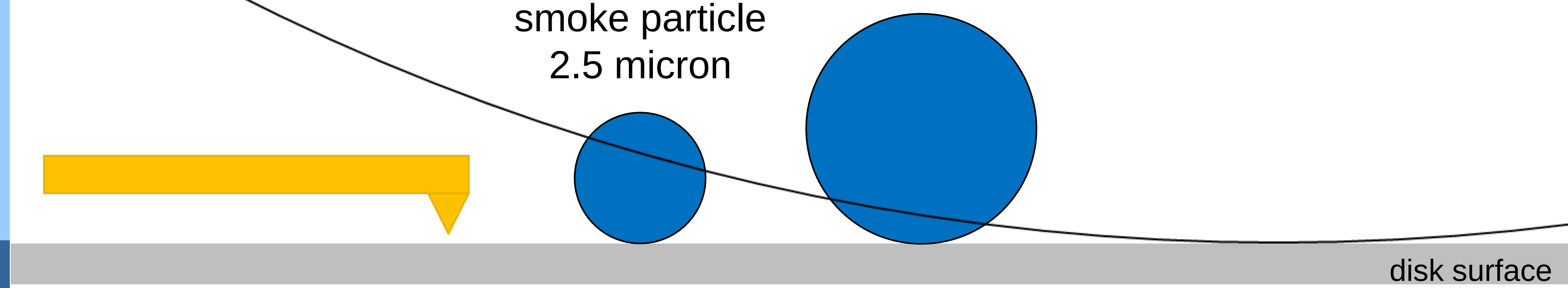
By moving radially, the arm
can position the read/write
head over any track (*seek*).

Disk Operation (Multi-Platter View)



Read/Write Head





human hair
75 micron

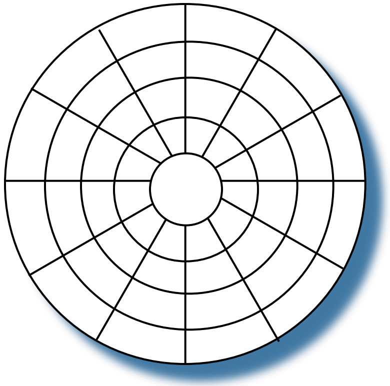
smoke particle
2.5 micron

disk surface

surface - disk head
0.1 micron
80km/h

dust particle
4 micron

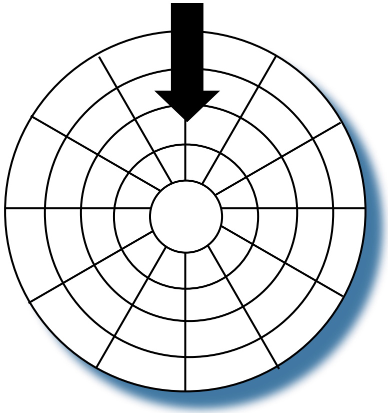
Disk Structure - top view of single platter



Surface organized into tracks

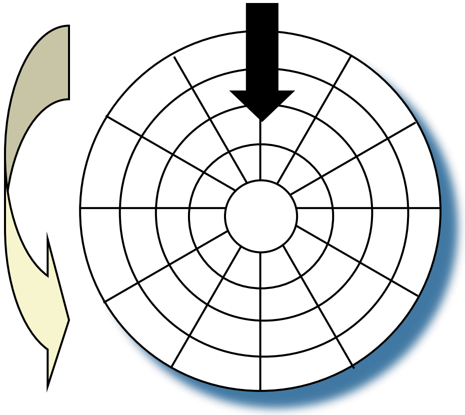
Tracks divided into sectors

Disk Access



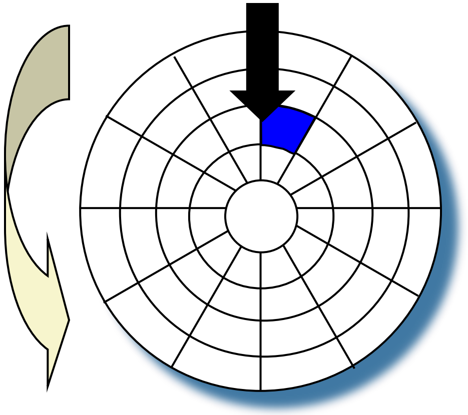
Head in position above a track

Disk Access



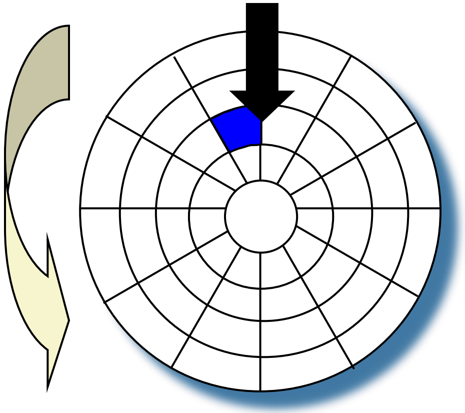
Rotation is counter-clockwise

Disk Access – Read



About to read blue sector

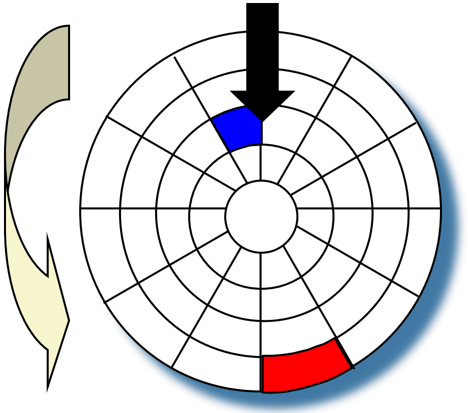
Disk Access – Read



After BLUE read

After reading blue sector

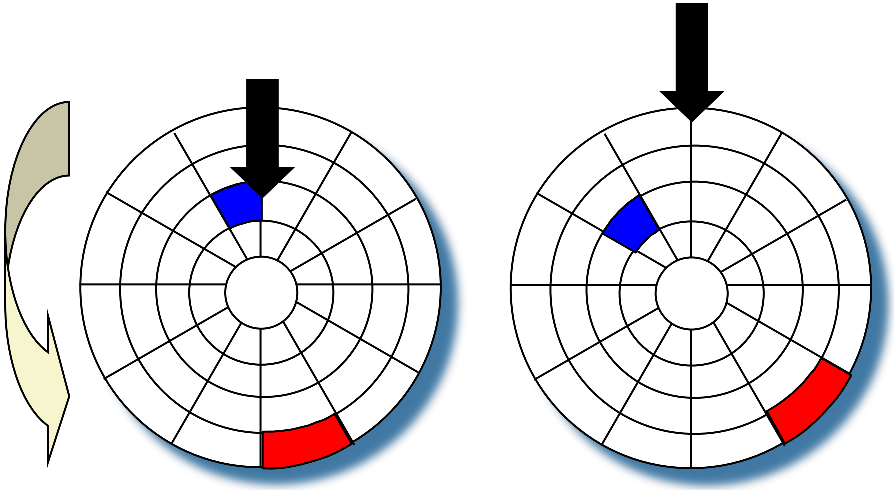
Disk Access – Read



After BLUE read

Red request scheduled next

Disk Access – Seek

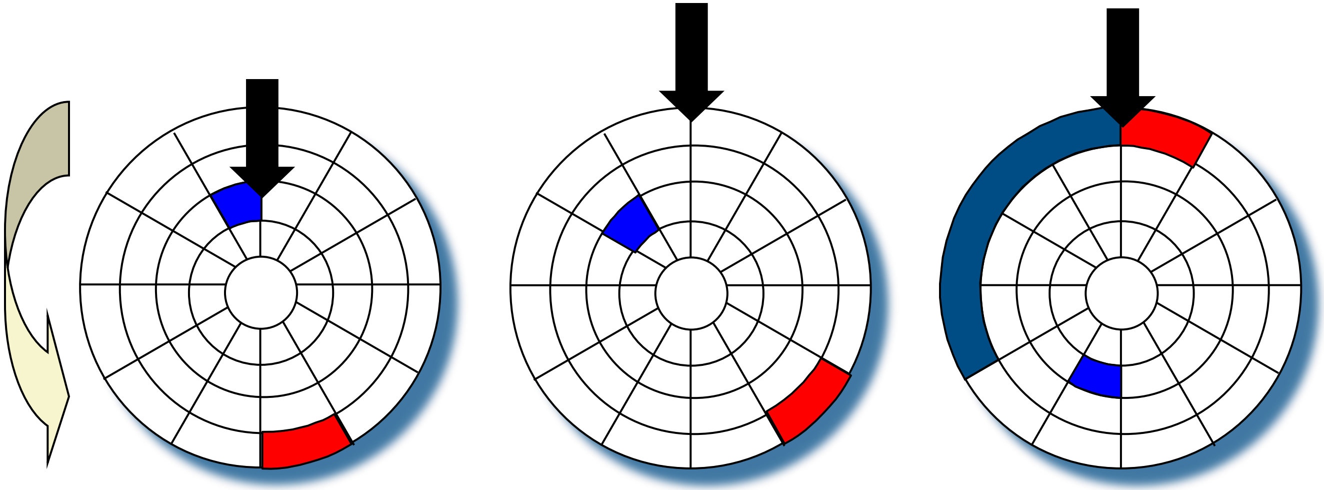


After BLUE read

Seek for RED

Seek to red’s track

Disk Access – Rotational Latency



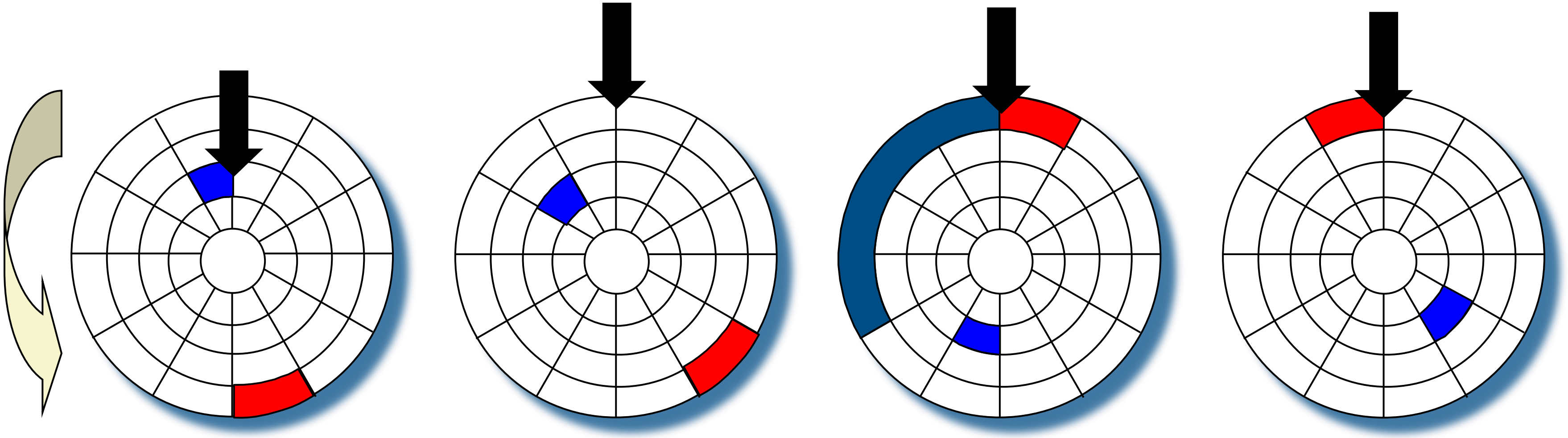
After BLUE read

Seek for RED

Rotational latency

Wait for red sector to rotate around

Disk Access – Read



After BLUE read

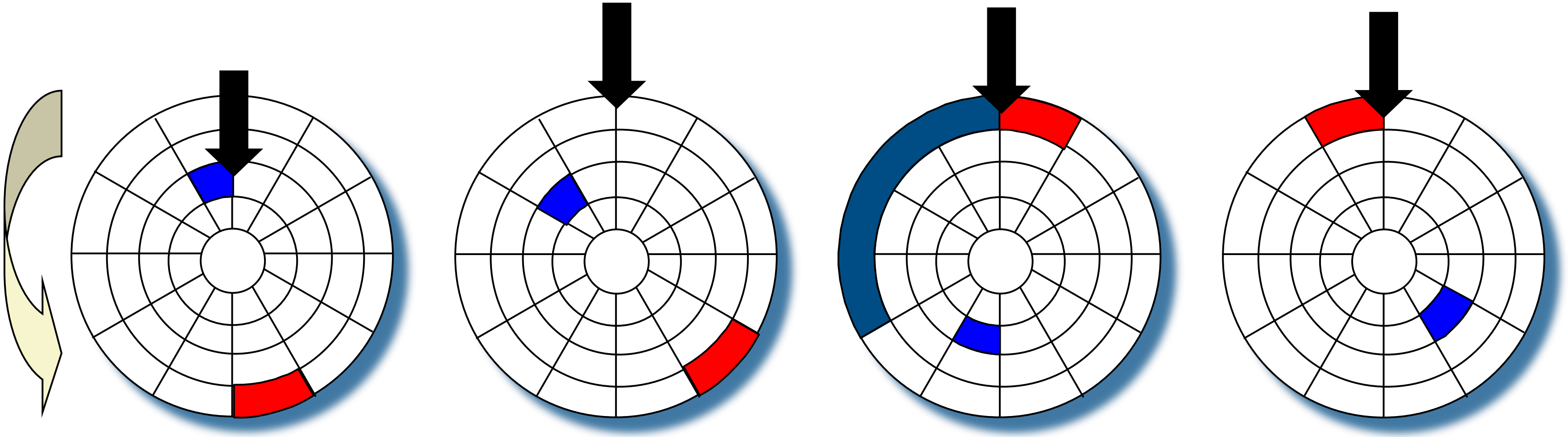
Seek for RED

Rotational latency

After RED read

Complete read of red

Disk Access – Service Time Components



After BLUE read

Seek for RED

Rotational latency

After RED read

Data transfer

Seek

Rotational latency

Data transfer

Disk Access Time

- Average time to access some target sector approximated by :
  - Taccess = Tavg seek + Tavg rotation + Tavg transfer

- Seek time (Tavg seek)
  - Time to position heads over cylinder containing target sector.
  - Typical Tavg seek is 3—9 ms

- Rotational latency (Tavg rotation)
  - Time waiting for first bit of target sector to pass under r/w head.
  - Tavg rotation = 1/2 x 1/RPMs x 60 sec/1 min
  - Typical Tavg rotation = 7200 RPMs

- Transfer time (Tavg transfer)
  - Time to read the bits in the target sector.
  - Tavg transfer = 1/RPM x 1/(avg # sectors/track) x 60 secs/1 min.

Disk Access Time Example

- Given:
  - Rotational rate = 7,200 RPM
  - Average seek time = 9 ms.
  - Avg # sectors/track = 400.

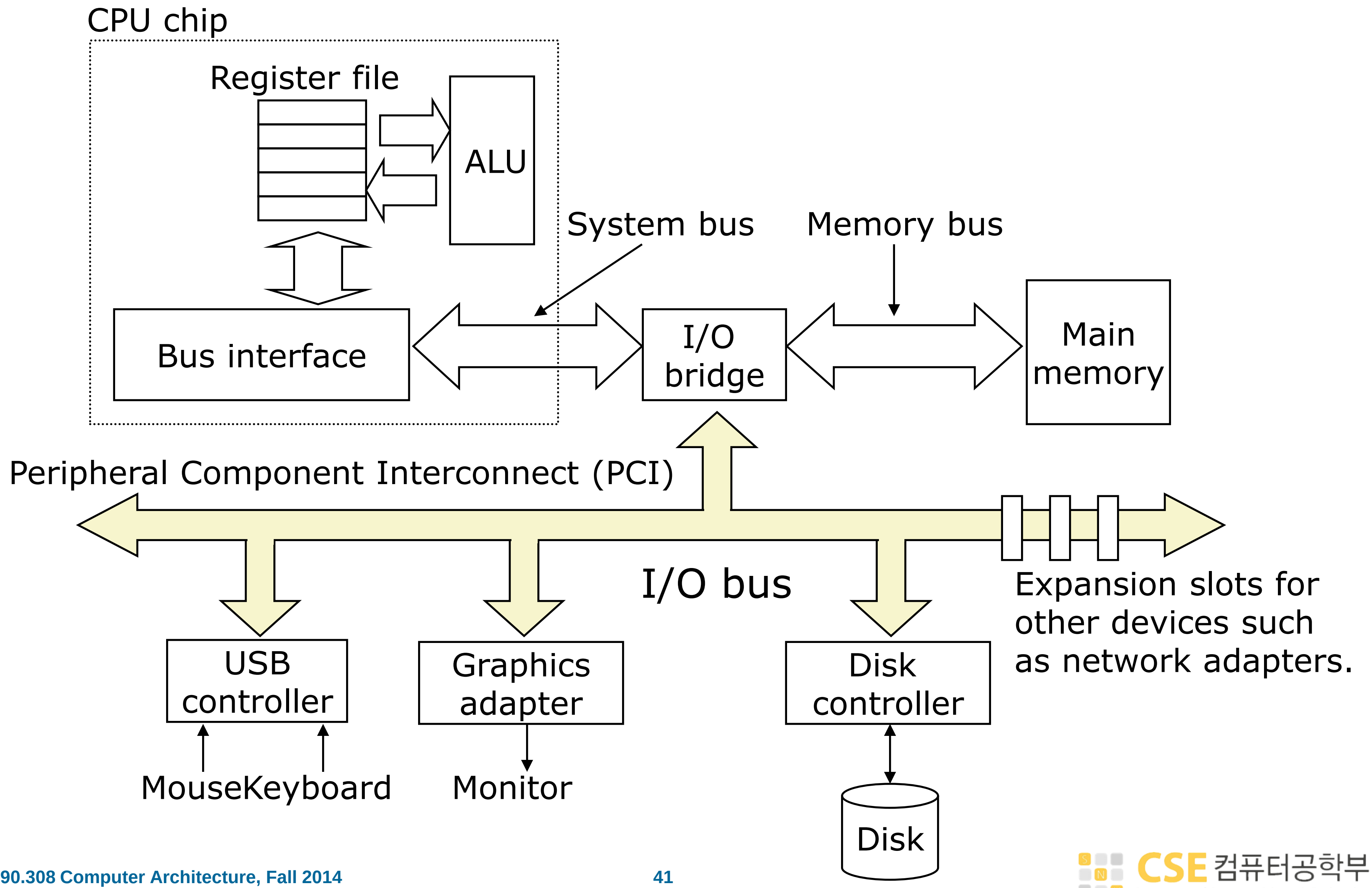
- Derived:
  - Tavg rotation = 1/2 x (60 secs/7200 RPM) x 1000 ms/sec = 4 ms.
  - Tavg transfer = 60/7200 RPM x 1/400 secs/track x 1000 ms/sec = 0.02 ms
  - Taccess = 9 ms + 4 ms + 0.02 ms

- Important points:
  - Access time dominated by seek time and rotational latency.
  - First bit in a sector is the most expensive, the rest are free.
  - SRAM access time is about 4 ns/doubleword, DRAM about 60 ns
    - Disk is about 40,000 times slower than SRAM,
    - 2,500 times slower than DRAM.

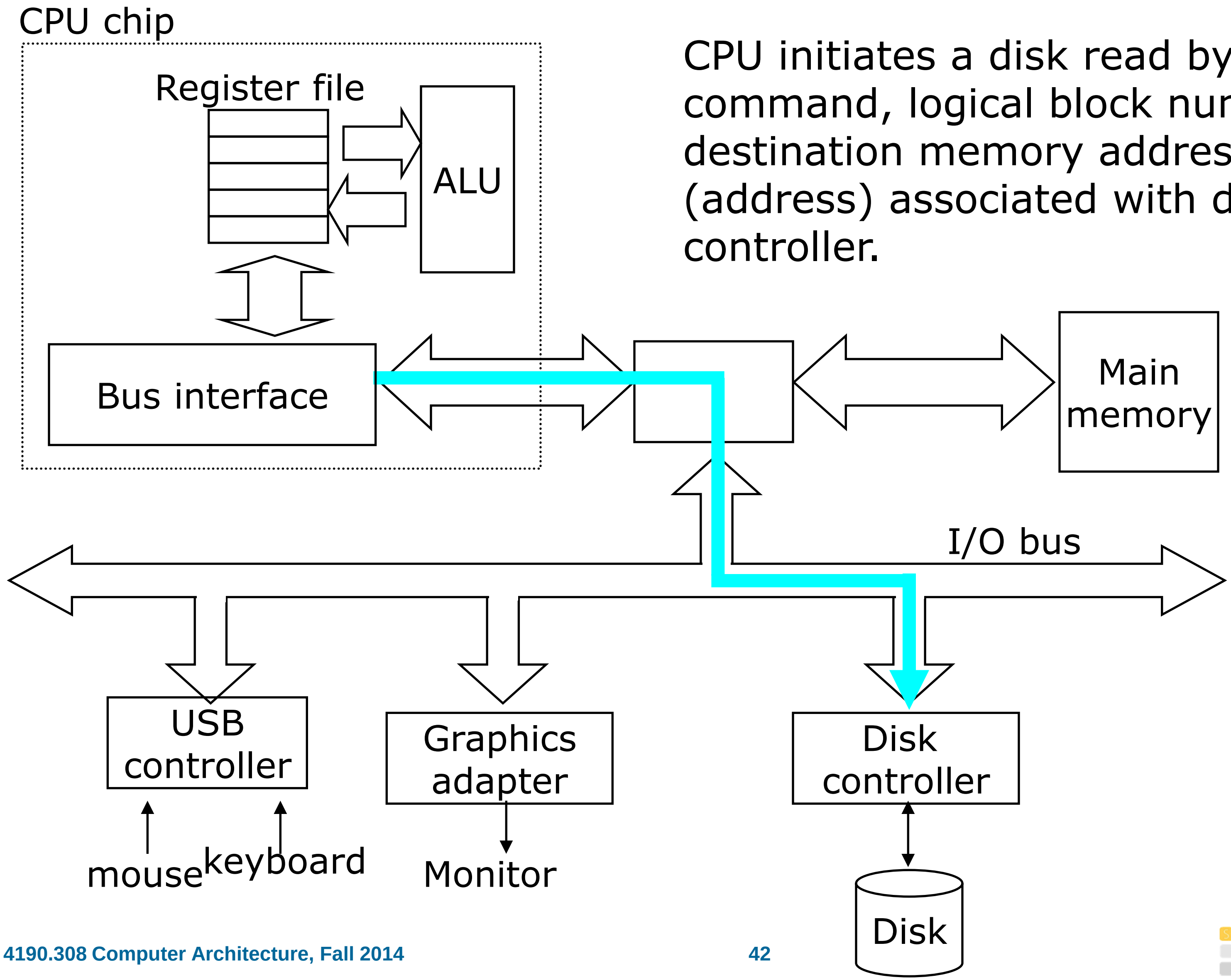
Logical Disk Blocks

- Modern disks present a simpler abstract view of the complex sector geometry:
    - The set of available sectors is modeled as a sequence of b-sized logical blocks (0, 1, 2, ...)
- Mapping between logical blocks and actual (physical) sectors
    - Maintained by hardware/firmware device called disk controller.
    - Converts requests for logical blocks into (surface,track,sector) triples.
- Allows controller to set aside spare cylinders for each zone.
    - Accounts for the difference in “formatted capacity” and “maximum capacity”.

I/O Bus


CPU chip
Register file
ALU
System bus
Memory bus
Bus interface
I/O bridge
Main memory
Peripheral Component Interconnect (PCI)
I/O bus
Expansion slots for other devices such as network adapters.
USB controller
Graphics adapter
Disk controller
MouseKeyboard
Monitor
Disk

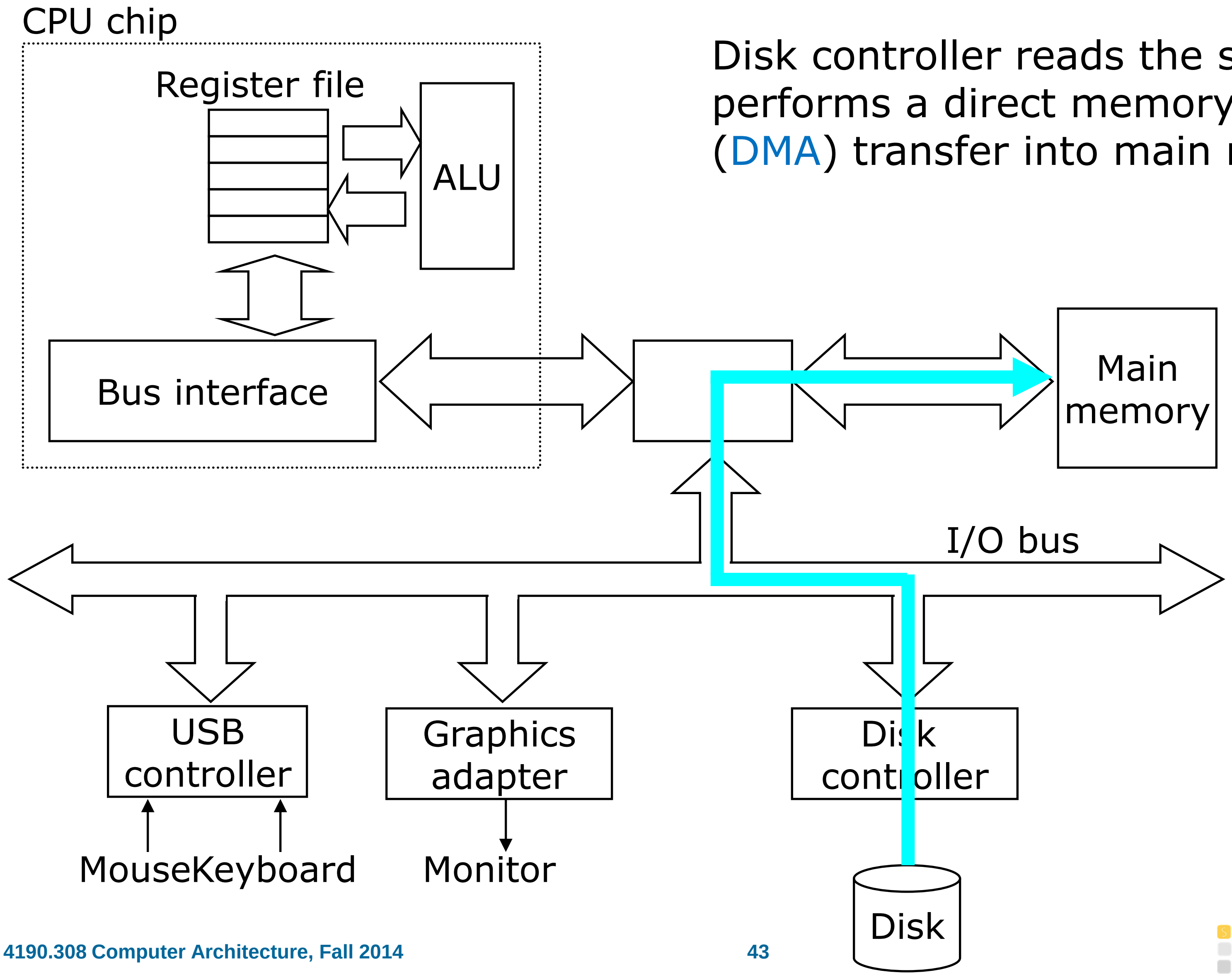
Reading a Disk Sector (1)



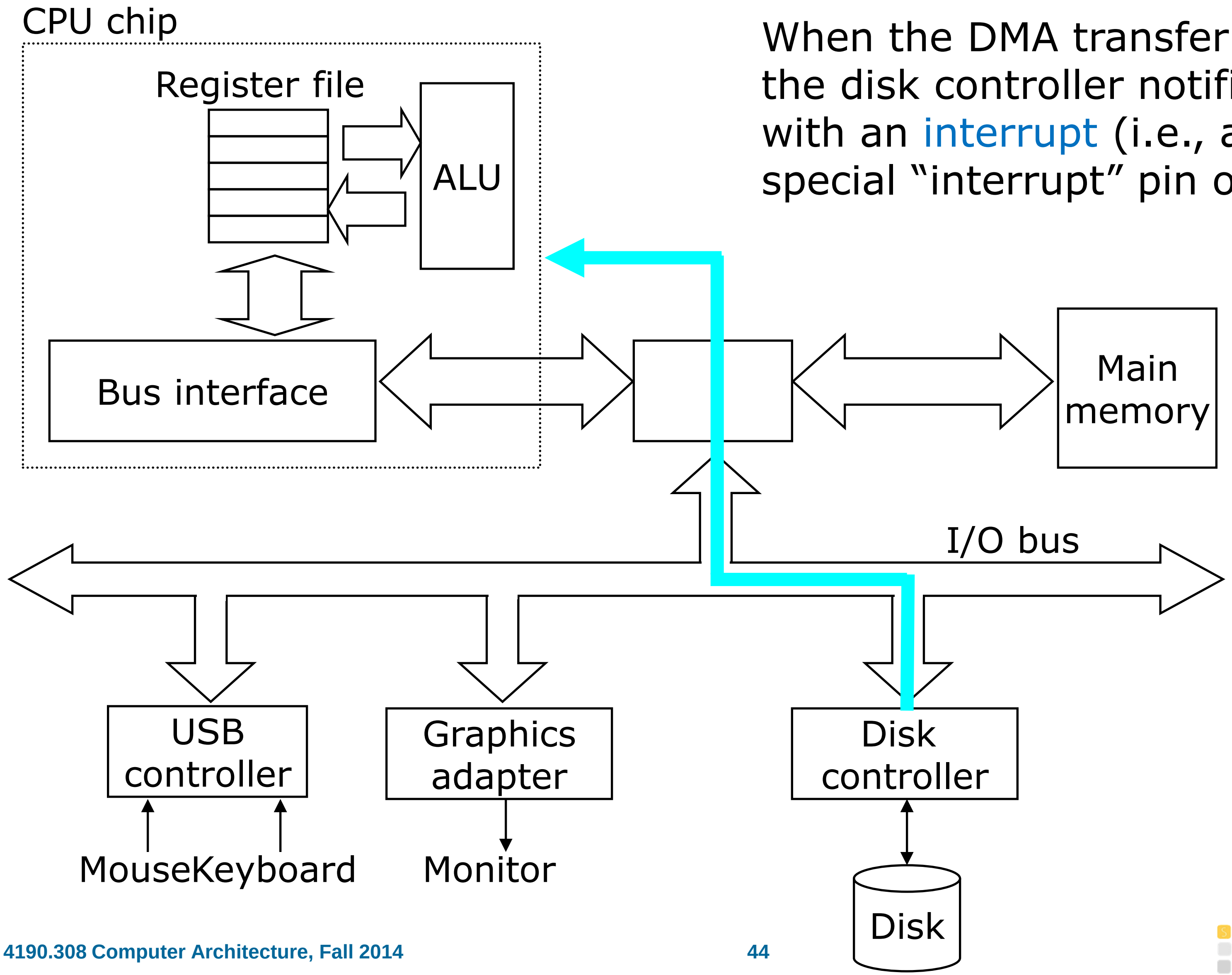
CPU initiates a disk read by writing a command, logical block number, and destination memory address to a port (address) associated with disk controller.

Reading a Disk Sector (2)

Disk controller reads the sector and performs a direct memory access (DMA) transfer into main memory.

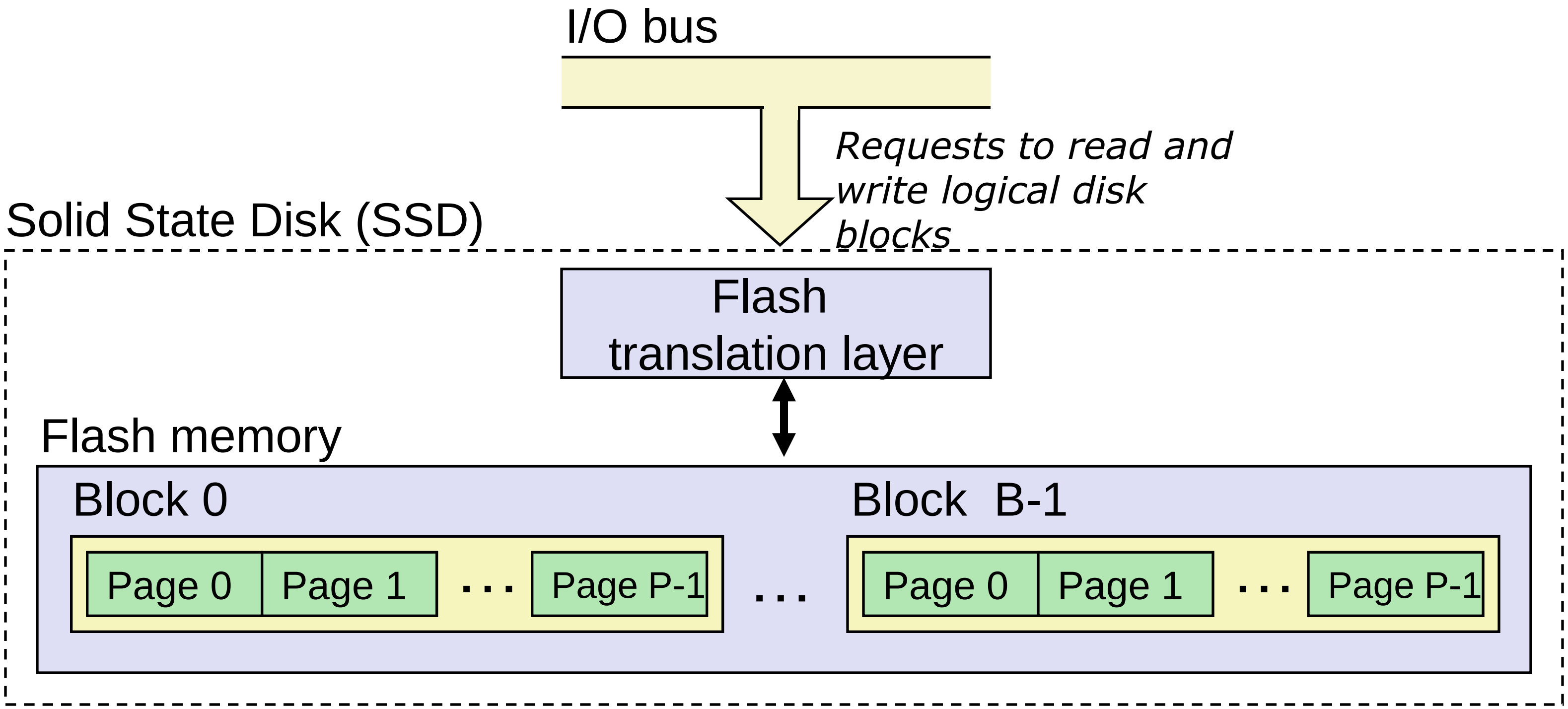


Reading a Disk Sector (3)



When the DMA transfer completes, the disk controller notifies the CPU with an interrupt (i.e., asserts a special "interrupt" pin on the CPU)

Solid State Disks (SSDs)



- Pages: 512KB to 4KB, Blocks: 32 to 128 pages
- Data read/written in units of pages.
- Page can be written only after its block has been erased
- A block wears out after 100,000 repeated writes.

SSD Performance Characteristics

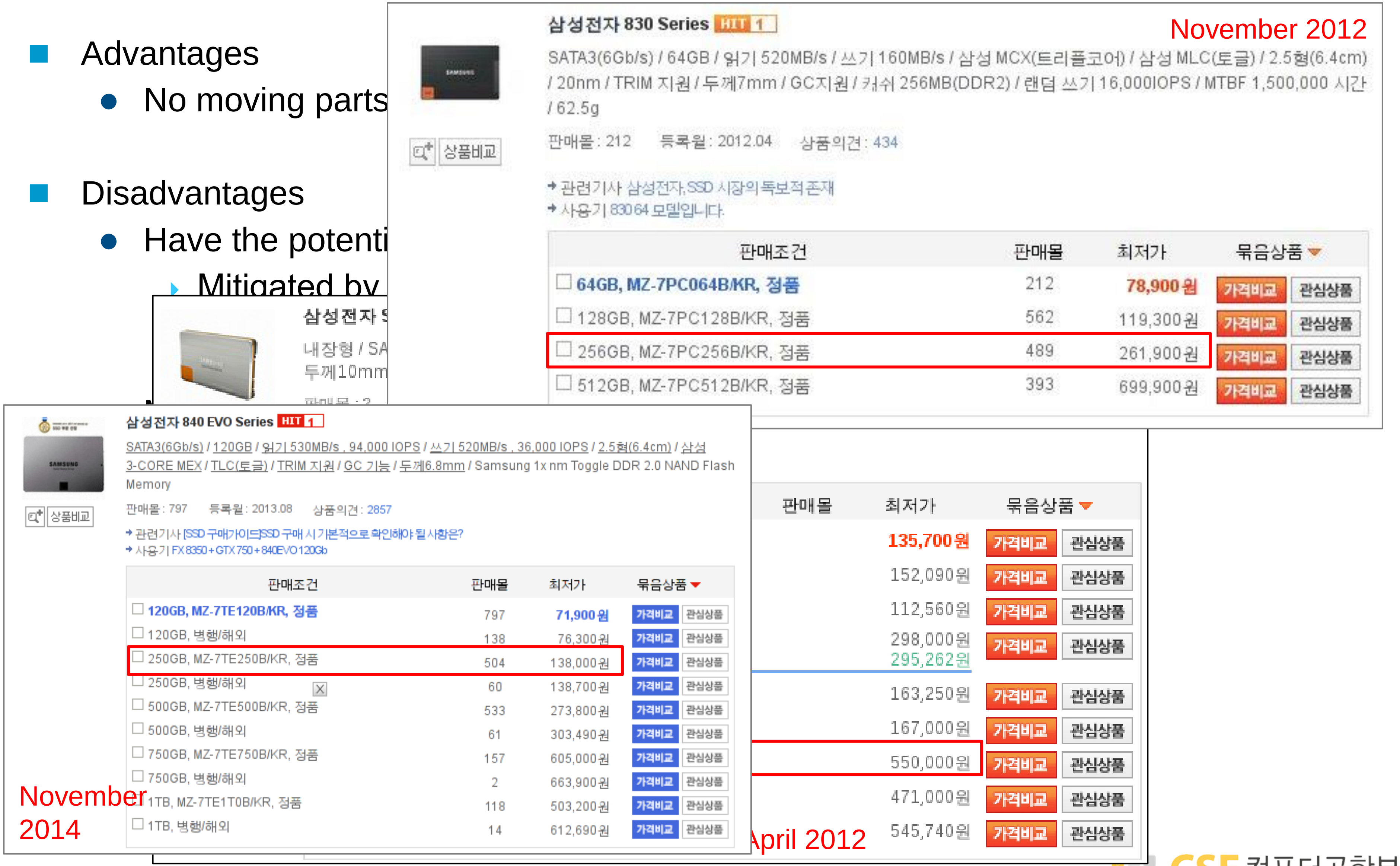
- Why are random writes so slow?

Sequential read tput	250 MB/s	Sequential write tput	170 MB/s
Random read tput	140 MB/s	Random write tput	14 MB/s
Rand read access	30 us	Random write access	300 us

  - Write the page into the new block
  - Copy other pages from old block to the new block

SSD Tradeoffs vs Rotating Disks

- Advantages
  - No moving parts
- Disadvantages
  - Have the potenti
    - Mitigated by



November 2012

삼성전자 830 Series HIT 1

SATA3(6Gb/s) / 64GB / 읽기 520MB/s / 쓰기 160MB/s / 삼성 MCX(트리플코어) / 삼성 MLC(토글) / 2.5형(6.4cm) / 20nm / TRIM 지원 / 두께7mm / GC지원 / 캐쉬 256MB(DDR2) / 랜덤 쓰기 16,000IOPS / MTBF 1,500,000 시간 / 62.5g

판매몰 : 212 등록일 : 2012.04 상품의견 : 434

➜ 관련기사 삼성전자,SSD 시장의 독보적 존재
➜ 사용기 830 64 모델입니다.

판매조건	판매몰	최저가	묶음상품
□ **64GB, MZ-7PC064B/KR, 정품**	212	**78,900원**	가격비교 관심상품
□ 128GB, MZ-7PC128B/KR, 정품	562	119,300원	가격비교 관심상품
□ 256GB, MZ-7PC256B/KR, 정품	489	261,900원	가격비교 관심상품
□ 512GB, MZ-7PC512B/KR, 정품	393	699,900원	가격비교 관심상품

삼성전자 S

내장형 / SA
두께10mm

판매몰 : 2

판매몰	최저가	묶음상품
	135,700원	가격비교 관심상품
	152,090원	가격비교 관심상품
	112,560원	가격비교 관심상품
	298,000원 295,262원	가격비교 관심상품
	163,250원	가격비교 관심상품
	167,000원	가격비교 관심상품
	550,000원	가격비교 관심상품
	471,000원	가격비교 관심상품
	545,740원	가격비교 관심상품

April 2012

삼성전자 840 EVO Series HIT 1

SATA3(6Gb/s) / 120GB / 읽기 530MB/s , 94,000 IOPS / 쓰기 520MB/s , 36,000 IOPS / 2.5형(6.4cm) / 삼성 3-CORE MEX / TLC(토글) / TRIM 지원 / GC 기능 / 두께6.8mm / Samsung 1x nm Toggle DDR 2.0 NAND Flash Memory

판매몰 : 797 등록일 : 2013.08 상품의견 : 2857

➜ 관련기사 [SSD 구매가이드]SSD 구매 시 기본적으로 확인해야 될 사항은?
➜ 사용기 FX 8350 + GTX 750 + 840EVO 120Gb

판매조건	판매몰	최저가	묶음상품
□ **120GB, MZ-7TE120B/KR, 정품**	797	**71,900원**	가격비교 관심상품
□ 120GB, 병행/해외	138	76,300원	가격비교 관심상품
□ 250GB, MZ-7TE250B/KR, 정품	504	138,000원	가격비교 관심상품
□ 250GB, 병행/해외	60	138,700원	가격비교 관심상품
□ 500GB, MZ-7TE500B/KR, 정품	533	273,800원	가격비교 관심상품
□ 500GB, 병행/해외	61	303,490원	가격비교 관심상품
□ 750GB, MZ-7TE750B/KR, 정품	157	605,000원	가격비교 관심상품
□ 750GB, 병행/해외	2	663,900원	가격비교 관심상품
□ 1TB, MZ-7TE1T0B/KR, 정품	118	503,200원	가격비교 관심상품
□ 1TB, 병행/해외	14	612,690원	가격비교 관심상품

November 2014

Storage Trends

SRAM

Metric	1980	1985	1990	1995	2000	2005	2010	*2010:1980*
$/MB	19,200	2,900	320	256	100	75	60	*320*
access (ns)	300	150	35	15	3	2	1.5	*200*

DRAM

Metric	1980	1985	1990	1995	2000	2005	2010	*2010:1980*
$/MB	8,000	880	100	30	1	0.1	0.06	*130,000*
access (ns)	375	200	100	70	60	50	40	*9*
typical size (MB)	0.064	0.256	4	16	64	2,000	8,000	*125,000*

Disk

Metric	1980	1985	1990	1995	2000	2005	2010	*2010:1980*
$/MB	500	100	8	0.30	0.01	0.005	0.0003	*1,600,000*
access (ms)	87	75	28	10	8	*4*	*3*	*29*
typical size (MB)	1	10	160	1,000	20,000	160,000	1,500,000	*1,500,000*

→ it is easier to increase density than to decrease access time

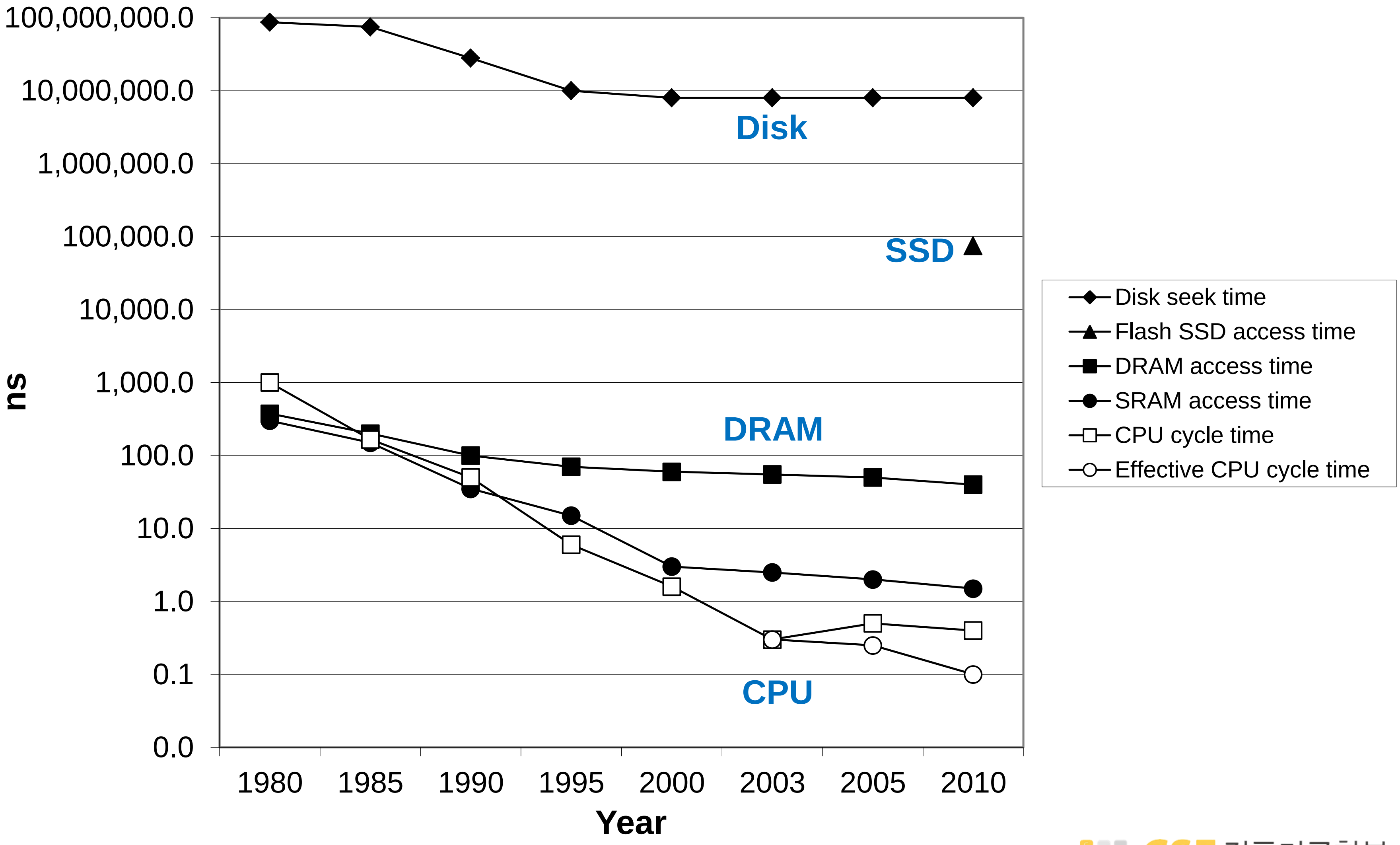
CPU Clock Rates

Inflection point in computer history
when designers hit the “Power Wall”

	1980	1990	1995	2000	2003	2005	2010	*2010:1980*
CPU	8080	386	Pentium	P-III	P-4	Core 2	Core i7	---
Clock rate (MHz)	1	20	150	600	3300	2000	2500	2500
Cycle time (ns)	1000	50	6	1.6	0.3	0.50	0.4	2500
Cores	1	1	1	1	1	2	4	4
Effective cycle time (ns)	1000	50	6	1.6	0.3	0.25	0.1	10,000

The CPU-Memory Gap

The gap widens between DRAM, disk, and CPU speeds.



Locality to the Rescue!

- The key to bridging this CPU-Memory gap is a fundamental property of computer programs known as locality

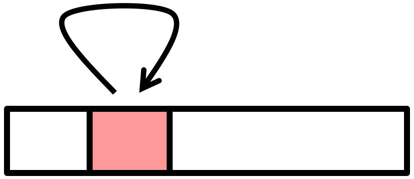
The Memory Hierarchy

- Storage technologies and trends
- **Locality of reference**
- Caching in the memory hierarchy

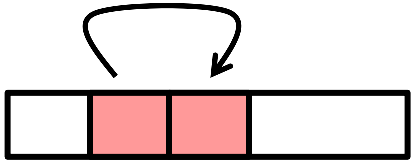
Locality

- Principle of Locality: Programs tend to use data and instructions with addresses near or equal to those they have used recently

- Temporal locality:
  - Recently referenced items are likely to be referenced again in the near future



- Spatial locality:
  - Items with nearby addresses tend to be referenced close together in time



Locality Example

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

- Data references
  - Reference array elements in succession (stride-1 reference pattern). Spatial locality
  - Reference variable sum each iteration. Temporal locality
- Instruction references
  - Reference instructions in sequence. Spatial locality
  - Cycle through loop repeatedly. Temporal locality

Qualitative Estimates of Locality

- Claim: Being able to look at code and get a qualitative sense of its locality is a key skill for a professional programmer.

- Question: Does this function have good locality with respect to array a?

```
int sum_array_rows(int a[M][N])
{
    int i, j, sum = 0;

    for (i = 0; i < M; i++)
        for (j = 0; j < N; j++)
            sum += a[i][j];
    return sum;
}
```

Locality Example

- Question: Does this function have good locality with respect to array a?

```
int sum_array_cols(int a[M][N])
{
    int i, j, sum = 0;

    for (j = 0; j < N; j++)
        for (i = 0; i < M; i++)
            sum += a[i][j];
    return sum;
}
```

Locality Example

- Question: Can you permute the loops so that the function scans the 3-d array ***a*** with a stride-1 reference pattern (and thus has good spatial locality)?

```
int sum_array_3d(int a[N][N][N])
{
    int i, j, k, sum = 0;

    for (i = 0; i < N; i++)
        for (j = 0; j < N; j++)
            for (k = 0; k < N; k++)
                sum += a[k][i][j];
    return sum;
}
```

Memory Hierarchies

- Some fundamental and enduring properties of hardware and software:
  - Fast storage technologies cost more per byte, have less capacity, and require more power (heat!).
  - The gap between CPU and main memory speed is widening.
  - Well-written programs tend to exhibit good locality.

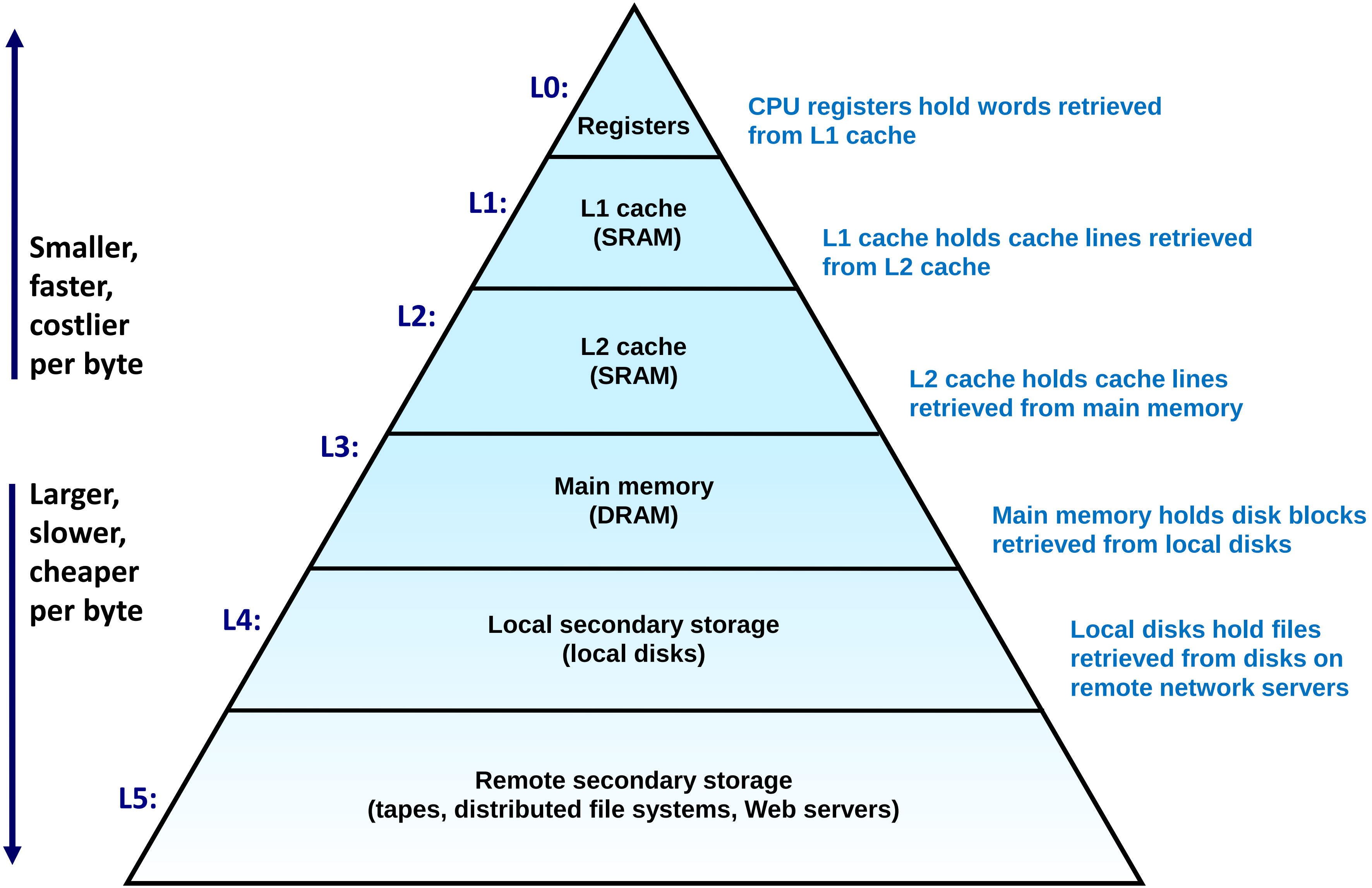
- These fundamental properties complement each other beautifully.

- They suggest an approach for organizing memory and storage systems known as a memory hierarchy.

The Memory Hierarchy

- Storage technologies and trends
- Locality of reference
- **Caching in the memory hierarchy**

An Example Memory Hierarchy


Smaller, faster, costlier per byte
Larger, slower, cheaper per byte
L0:
Registers
L1:
L1 cache (SRAM)
L2:
L2 cache (SRAM)
L3:
Main memory (DRAM)
L4:
Local secondary storage (local disks)
L5:
Remote secondary storage (tapes, distributed file systems, Web servers)
CPU registers hold words retrieved from L1 cache
L1 cache holds cache lines retrieved from L2 cache
L2 cache holds cache lines retrieved from main memory
Main memory holds disk blocks retrieved from local disks
Local disks hold files retrieved from disks on remote network servers

Caches

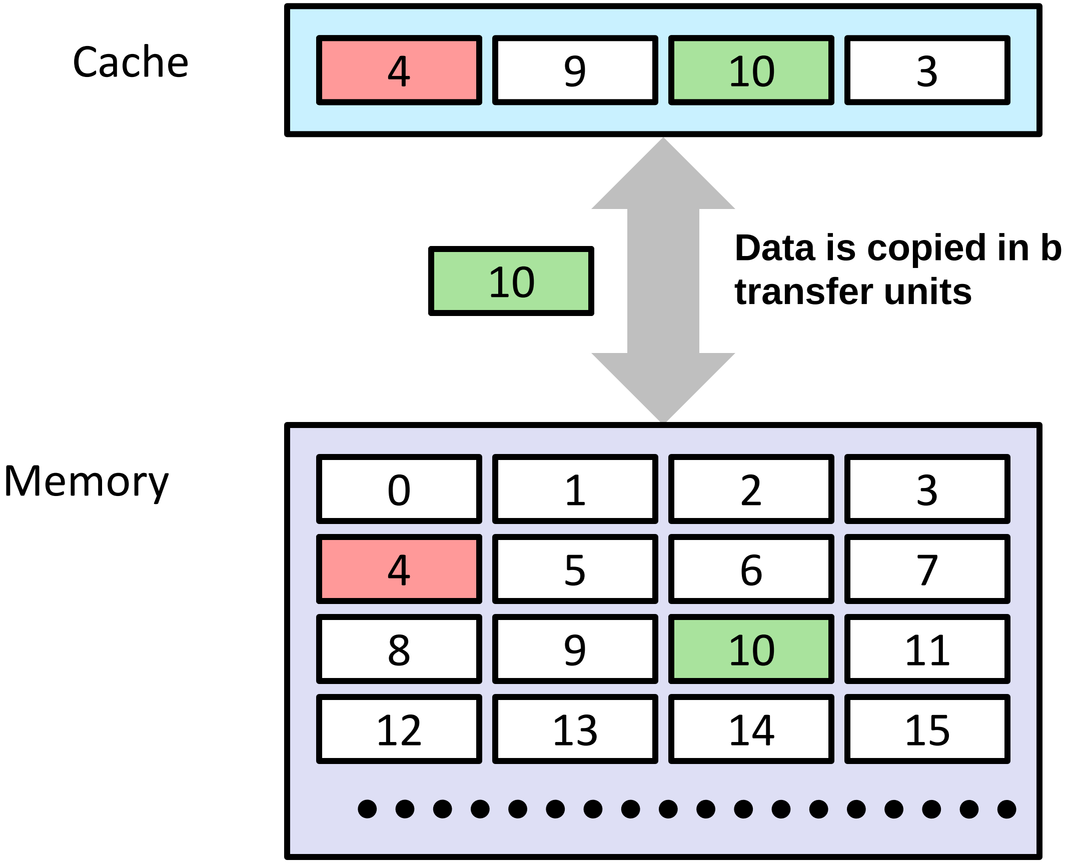
- Cache: A smaller, faster storage device that acts as a staging area for a subset of the data in a larger, slower device.

- Fundamental idea of a memory hierarchy:
  - For each k, the faster, smaller device at level k serves as a cache for the larger, slower device at level k+1.

- Why do memory hierarchies work?
  - Because of locality, programs tend to access the data at level k more often than they access the data at level k+1.
  - Thus, the storage at level k+1 can be slower, and thus larger and cheaper per bit.

- Big Idea: The memory hierarchy creates a large pool of storage that costs as much as the cheap storage near the bottom, but that serves data to programs at the rate of the fast storage near the top.

General Cache Concepts

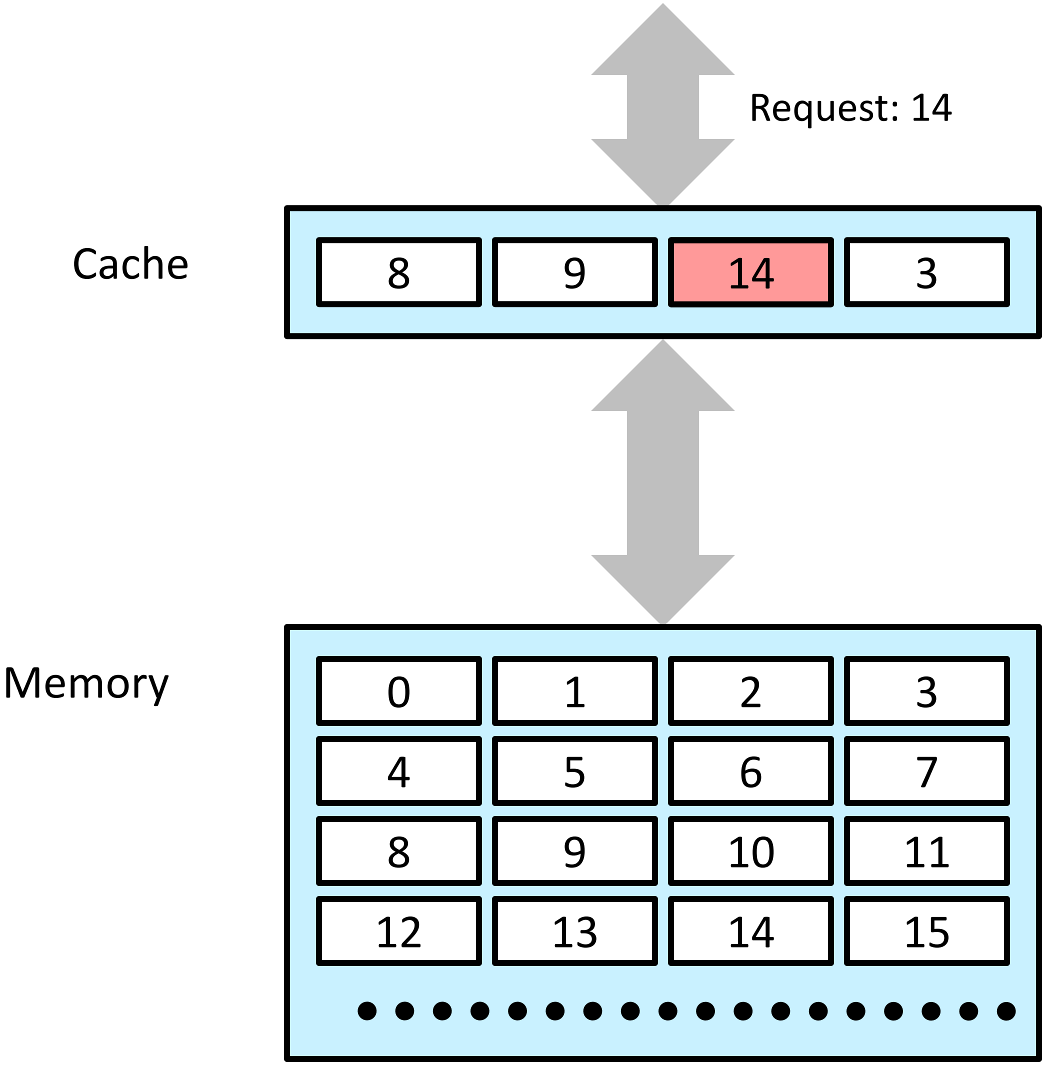


Smaller, faster, more expensive memory caches a subset of the blocks

Data is copied in block-sized transfer units

Larger, slower, cheaper memory viewed as partitioned into "blocks"

General Cache Concepts: Hit



Request: 14

Cache

8	9	14	3

Memory

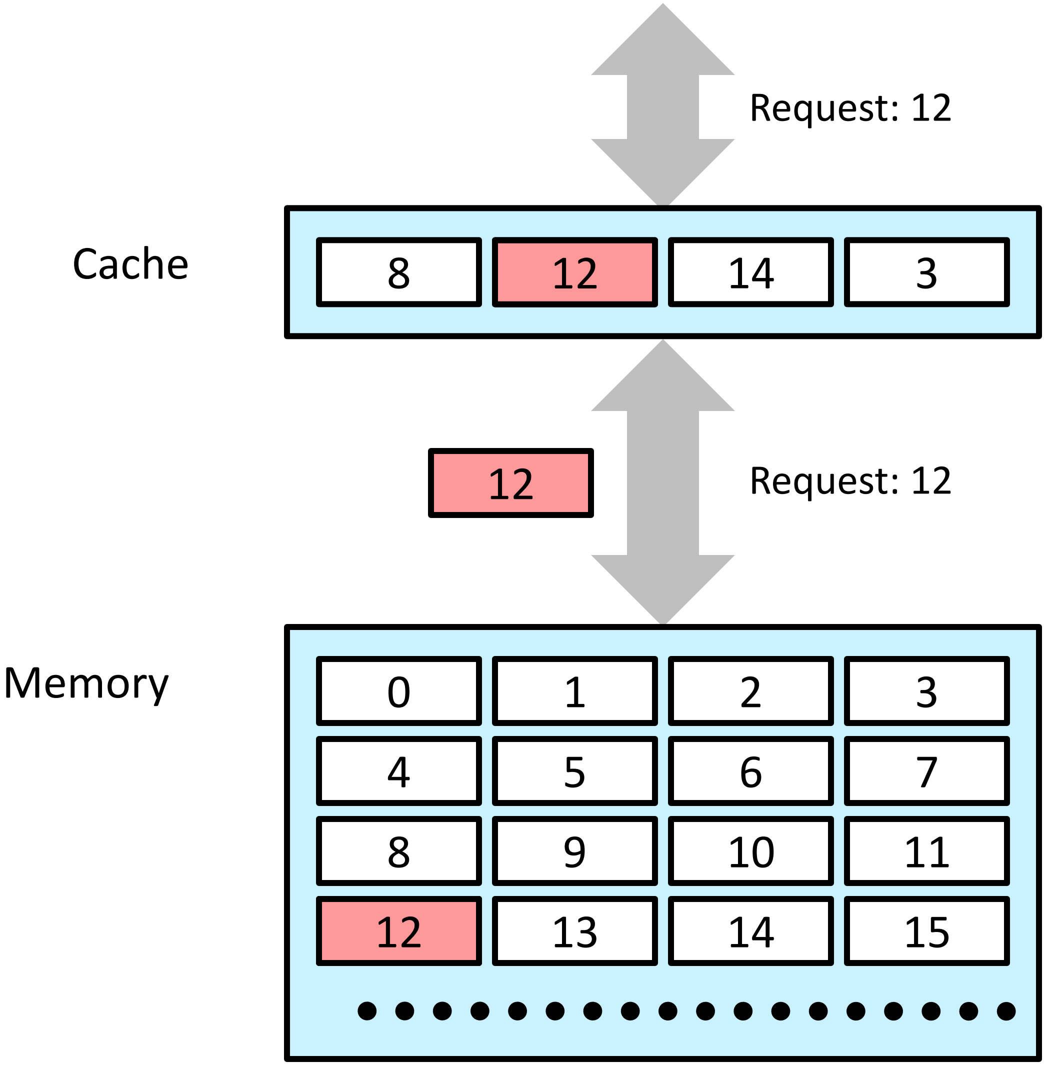
0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15

● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ●

Data in block b is needed

Block b is in cache:
Hit!

General Cache Concepts: Miss



Data in block b is needed

Block b is not in cache:
Miss!

***Block b is fetched from** memory*

Block b is stored in cache

- Placement policy: determines where b goes
- Replacement policy: determines which block gets evicted (victim)

General Caching Concepts: Types of Cache Misses

- Cold (compulsory) miss
  - Cold misses occur because the cache is empty.

- Conflict miss
  - Most caches limit blocks at level k+1 to a small subset (sometimes a singleton) of the block positions at level k.
    - e.g. Block i at level k+1 must be placed in block (i mod 4) at level k.
  - Conflict misses occur when the level k cache is large enough, but multiple data objects all map to the same level k block.
    - e.g. Referencing blocks 0, 8, 0, 8, 0, 8, ... would miss every time.

- Capacity miss
  - Occurs when the set of active cache blocks (working set) is larger than the cache.

Examples of Caching in the Hierarchy

Cache Type	What is Cached?	Where is it Cached?	Latency (cycles)	Managed By
Registers	4-8 bytes words	CPU core	0	Compiler
TLB	Address translations	On-Chip TLB	0	Hardware
L1 cache	64-bytes block	On-Chip L1	1	Hardware
L2 cache	64-bytes block	On/Off-Chip L2	10	Hardware
Virtual Memory	4-KB page	Main memory	100	Hardware + OS
Buffer cache	Parts of files	Main memory	100	OS
Disk cache	Disk sectors	Disk controller	100,000	Disk firmware
Network buffer cache	Parts of files	Local disk	10,000,000	AFS/NFS client
Browser cache	Web pages	Local disk	10,000,000	Web browser
Web cache	Web pages	Remote server disks	1,000,000,000	Web proxy server

Summary

- The speed gap between CPU, memory and mass storage continues to widen.
- Well-written programs exhibit a property called locality.
- Memory hierarchies based on caching close the gap by exploiting locality.