CSE2431 – Lecture Topic 6
Memory Hierachy (part 1)

The Ohio State University
COLLEGE OF ENGINEERING

Memory Hierarchy (Part 1)

Instructor: Luan Duong, Ph.D.

CSE 2431: Introduction to Operating Systems

Reading: **Chap. 6** from Computer Systems: A Programmer's Perspective by Randal E. Bryant and David R. O'Halloran, 3rd edition, Pearson/Prentice Hall, 2016

Previous lecture

- Memory overview (part 1)
- Virtual memory procedures (part 2-3-4)
  - Dynamic relocation
  - Paging
  - Swaping

Motivation

- Briefly seen previously, an actual computer system contains a ***hierarchy*** of storage devices with different costs, capacities, and access times.
- With a memory hierarchy, a faster storage device at one level of the hierarchy acts as a staging area for a slower storage device at the next lower level.
- Well-written software exploits the hierarchy, accessing the faster storage device at a particular level more frequently than the storage at the next level.
- As a programmer, understanding the memory hierarchy will result in better application performance.

Outline: Memory Hierarchy

- **Storage Technologies**
- Locality
- Memory Hierarchy
- Cache Memories
- Writing Cache-friendly Code
- Impact of Caches on Program Performance

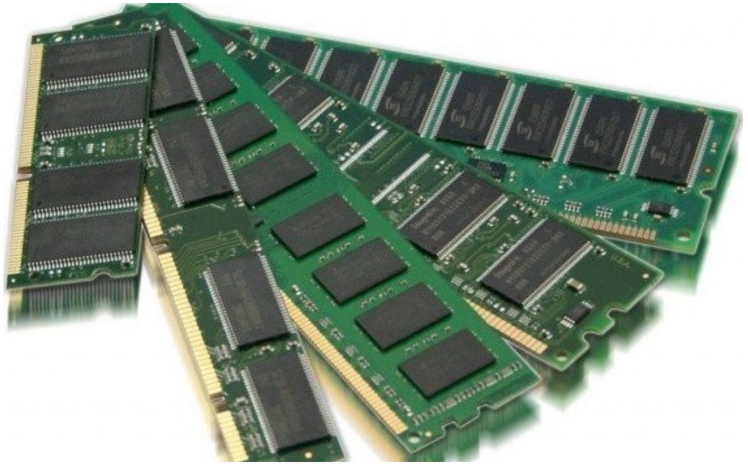
Storage Technologies

- Random-Access Memory (**RAM**)
- Disk Storage (**Disk**)
- Solid State Disks (**SSD**)
- Storage Technology Trends

Random-Access Memory **(RAM)**

- Features:
  - Basic storage unit is usually a cell (one pit per cell)
  - RAM is traditionally packaged as a chip
  - Multiple chips form memory
- Different RAM:
  - Static RAM (sRAM)
  - Dynamic RAM (DRAM)

A SRAM from Nintendo Entertainment System (wiki)



Typical DRAM

Random-Access Memory **(RAM)**

- Static RAM (SRAM) and Dynamic RAM (DRAM)

Static RAM (SRAM)	**Dynamic RAM (DRAM)**
+ Each cell implemented with a six-transistor circuit	+ Each bit stored as charge on a capacitor
+ Holds value while power is maintained: volatile	+ Value must be refreshed every 10 msec–100 msec: volatile
+ Insensitive to disturbances such as electrical noise, radiation, etc.	+ Sensitive to disturbances
+ Faster and more expensive than DRAM	+ Slower and cheaper than SRAM

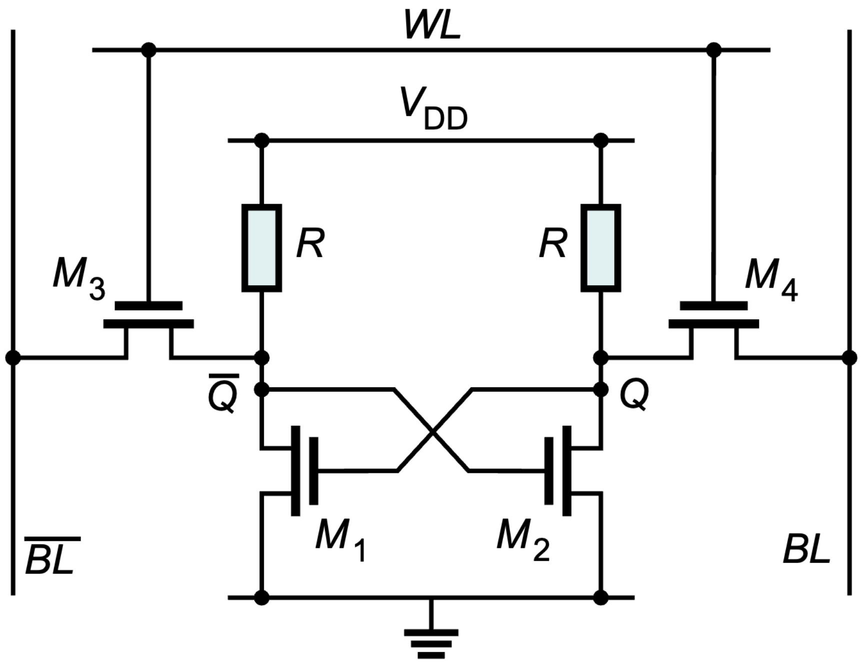
Random-Access Memory **(RAM)**

- Static RAM (SRAM) and Dynamic RAM (DRAM)

RAM Type	Transistors / Bit	Access Time	Needs Refresh?	Sensitive?	Cost	Applications
SRAM	4 or 6	1×	No	No	100×	**Cache memories**
DRAM	1	10×	Yes	Yes	1×	**Main memories, cache buffers**

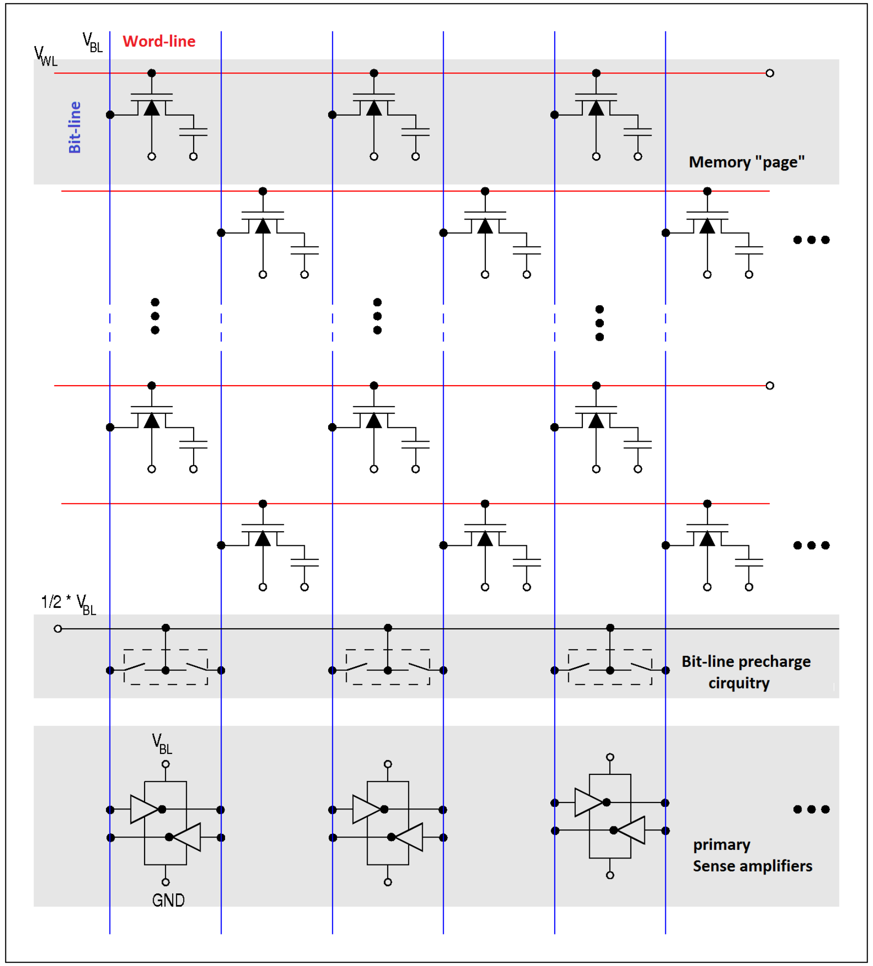
Random-Access Memory **(RAM)**

- Static RAM (SRAM) and Dynamic RAM (DRAM)



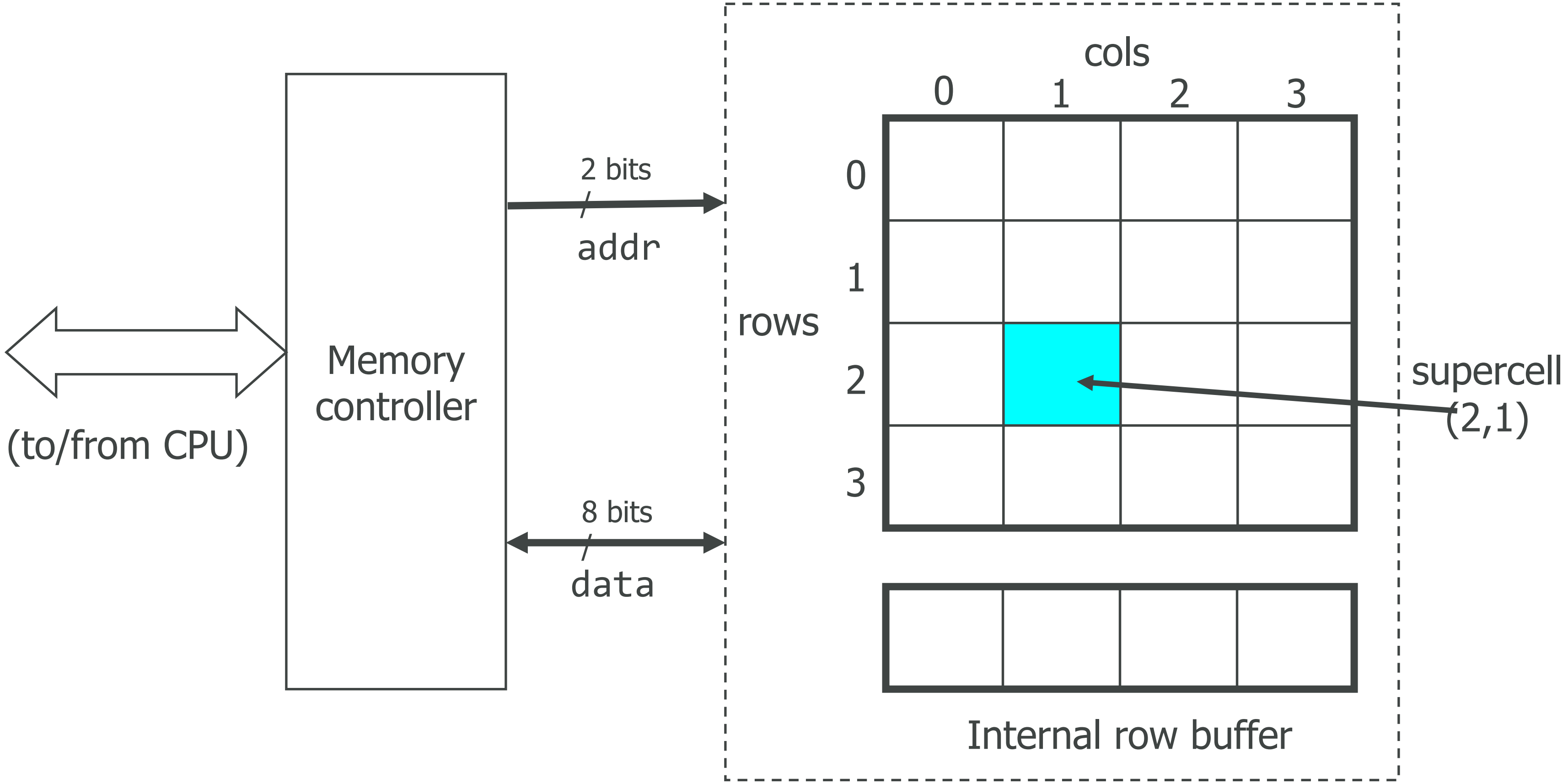
1 cell of SRAM. It is also called “6T cell, or 6 Transistors per bit

A cell of DRAM only consists of 1 transistor/ capacitor (1T cell)



Conventional DRAM Organization

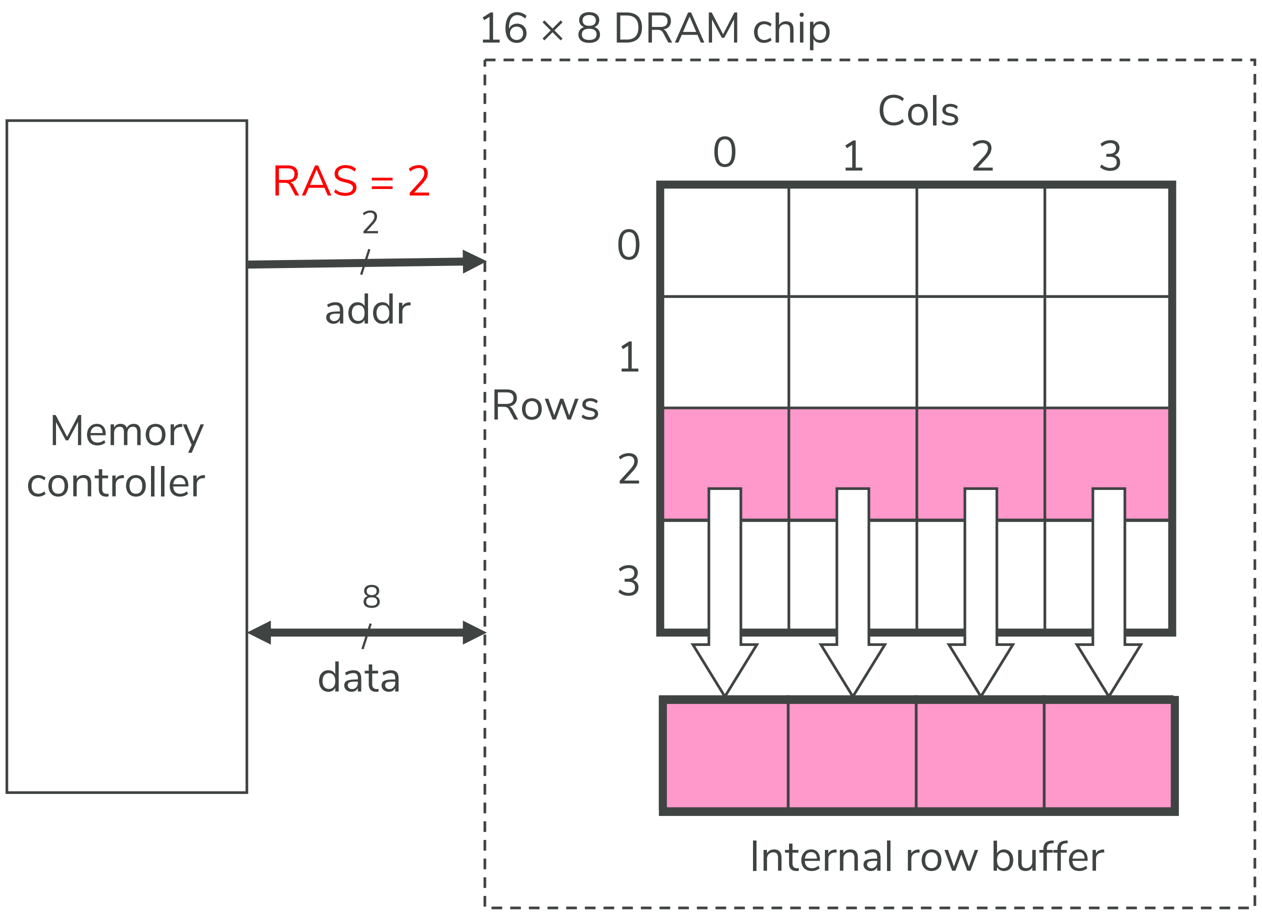
- d × w DRAM: dw total bits organized as d supercells of size w bits



Reading DRAM Supercell (2,1) (1)

- Step 1(a): Row access strobe (RAS) selects row 2.

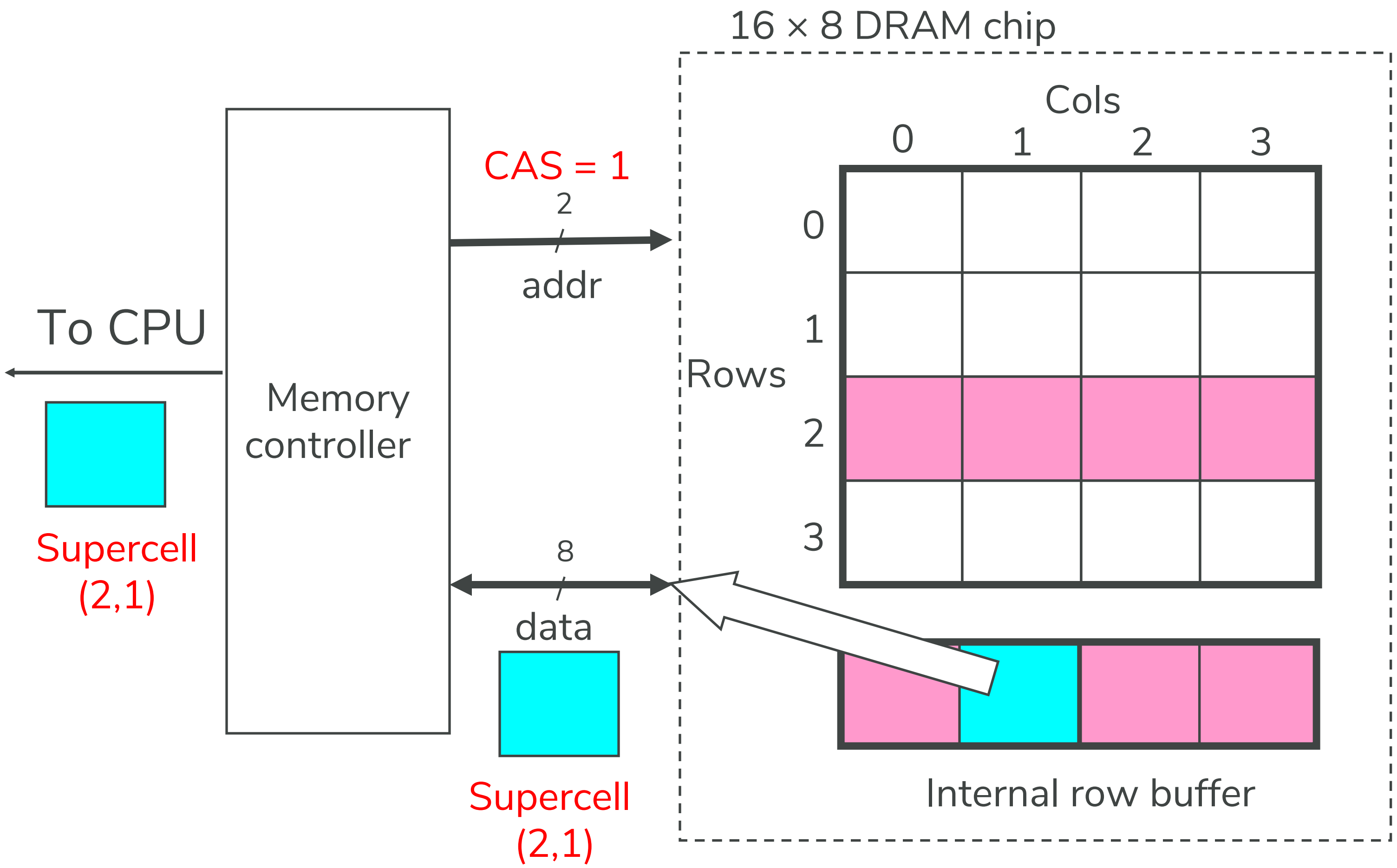
- Step 1(b): Row 2 copied from DRAM array to row buffer.



Reading DRAM Supercell (2,1) (1)

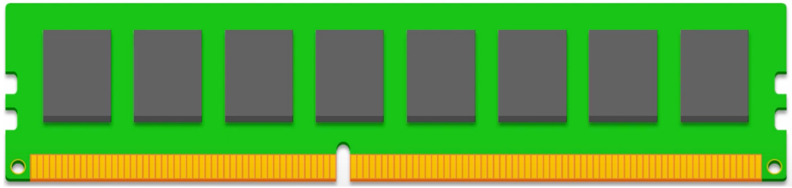
- Step 2(a): Column access strobe (CAS) selects column 1.

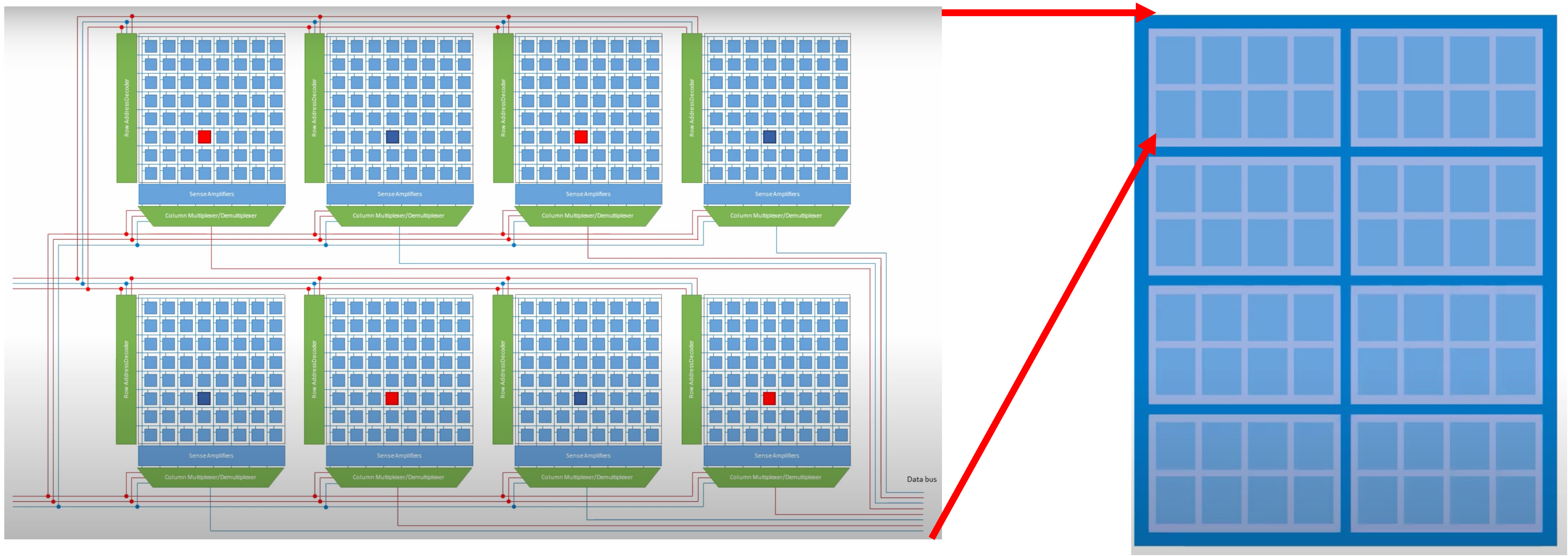
- Step 2(b): Supercell (2,1) copied from buffer to data lines, and eventually back to the CPU.



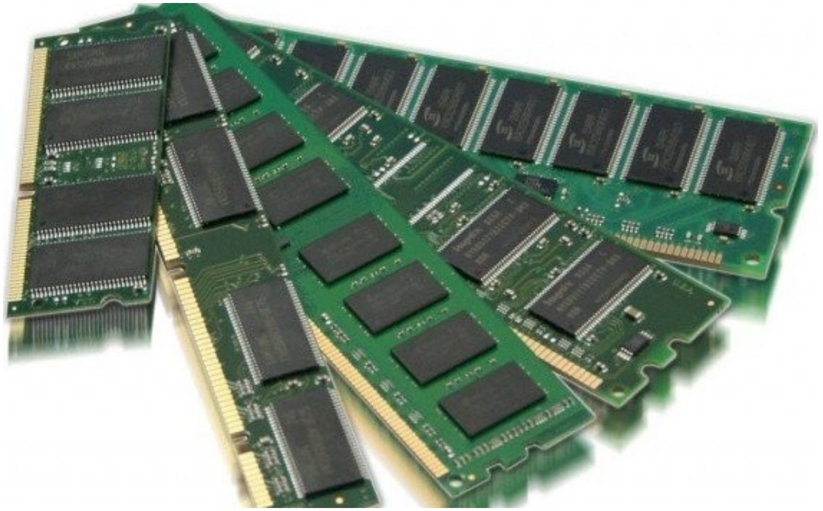
DRAM Memory Modules

- From 1 bit → a byte read → A bank.
- 8 banks → included in a single micro-chip (typically 4, 8, 16/chip) (bank address)
- 8 chips will be fit into a single circuit (DIMM – Dual Inline Memory Module) Ref: https://www.youtube.com/watch?v=Mhqi70OPW0o

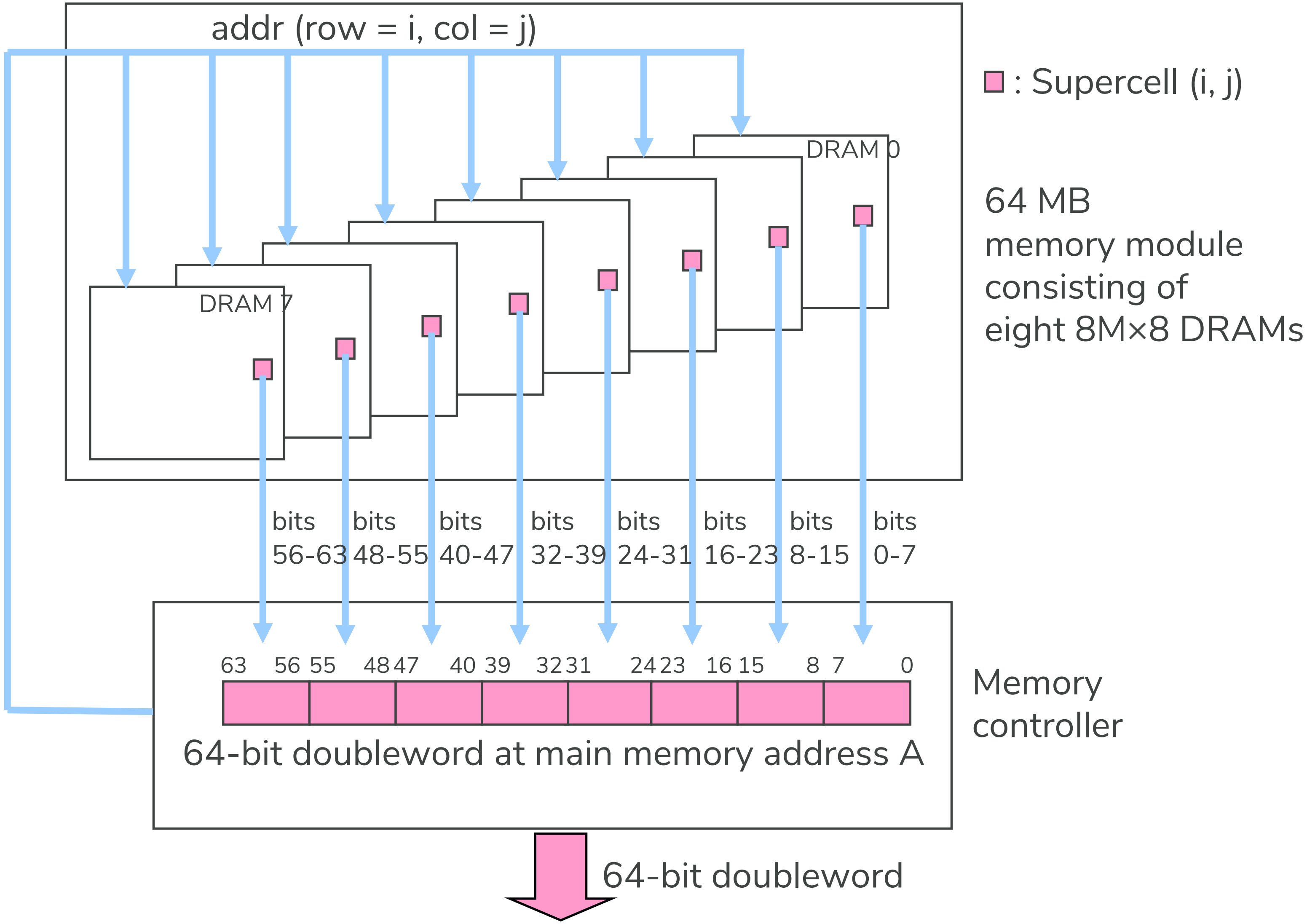




Memory Modules



Did you ever notice why there are different ‘black cells’ on your DRAMs?



Enchanced DRAMs

- Enhanced DRAMs have optimizations that improve the speed with which basic DRAM cells are accessed.
- **Examples:**
  - Fast page mode DRAM (**FPM DRAM**)
  - Extended data out DRAM (**EDO DRAM**)
  - Synchronous DRAM (**SDRAM**)
  - Double Data-Rate Synchronous DRAM (**DDR SDRAM**)
  - Rambus DRAM (**RDRAM**)
  - Video RAM (**VRAM**)

But…

- DRAMs and SRAMs are **volatile** memory!
- It means “information will **NOT** be retained if supply voltage is turned off”.
- Is that what you want from your computers? Isn’t a computer supposed to **process the data AND still retain the data in case of power outage**?

Non-volatile Memory

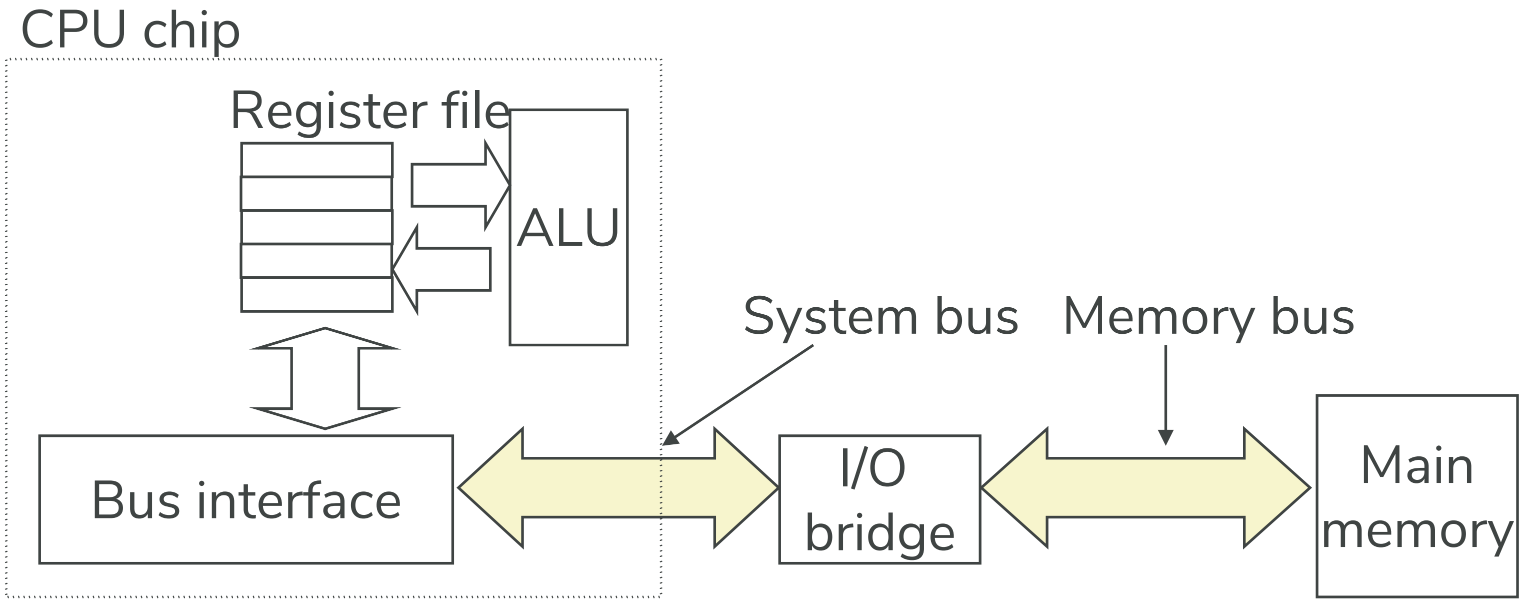
- Information **retained** if supply voltage is **turned off**
- Collectively referred to as **read-only memories (ROM)** although some **may be written to as well as read**
- **Distinguishable** by the **number of times** they can be reprogrammed (written to) and by the mechanism for reprogramming them
- Used for **firmware programs** (BIOS, controllers for disks, network cards, graphics accelerators, security subsystems…), solid state disks, disk caches

Non-volatile Memory

Term	Abbreviation	Definition	# of Times Programmed
Read-only memory	**ROM**	Programmed during production	0
Programmable ROM	**PROM**	Fuse associated with NAND, NOR cell that's blown once (zapping with electrical current)	1
Erasable ROM	**EPROM**	Cells cleared by shining ultraviolet light, special device used to write 1s	1,000
Electrically erasable PROM	**EEPROM**	Like EPROM, but doesn't require separate programming device; can be programmed on printed circuit boards	100,000
Flash memory	–	Based on EEPROM; wears out after ~100,000 repeated writes	100,000

Traditional Bus Structure Connect Bus, Memory

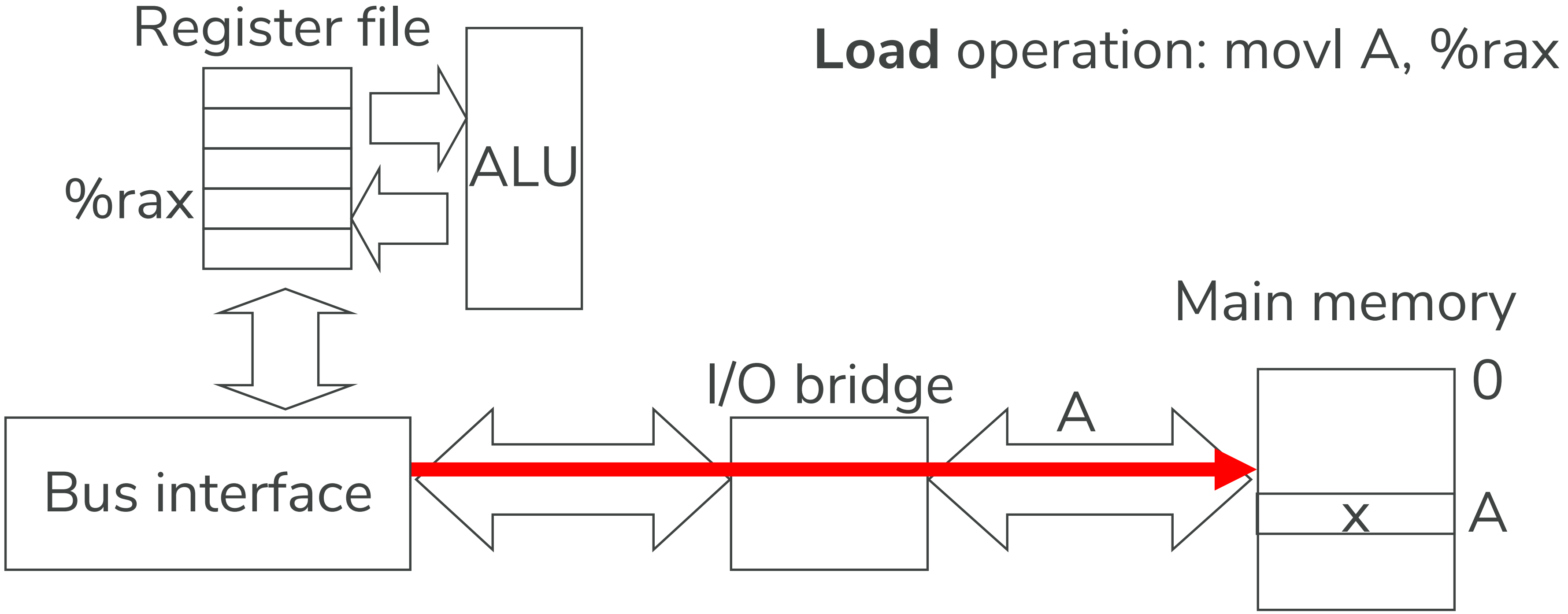
- If you recall… from the first lesson…



- 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)

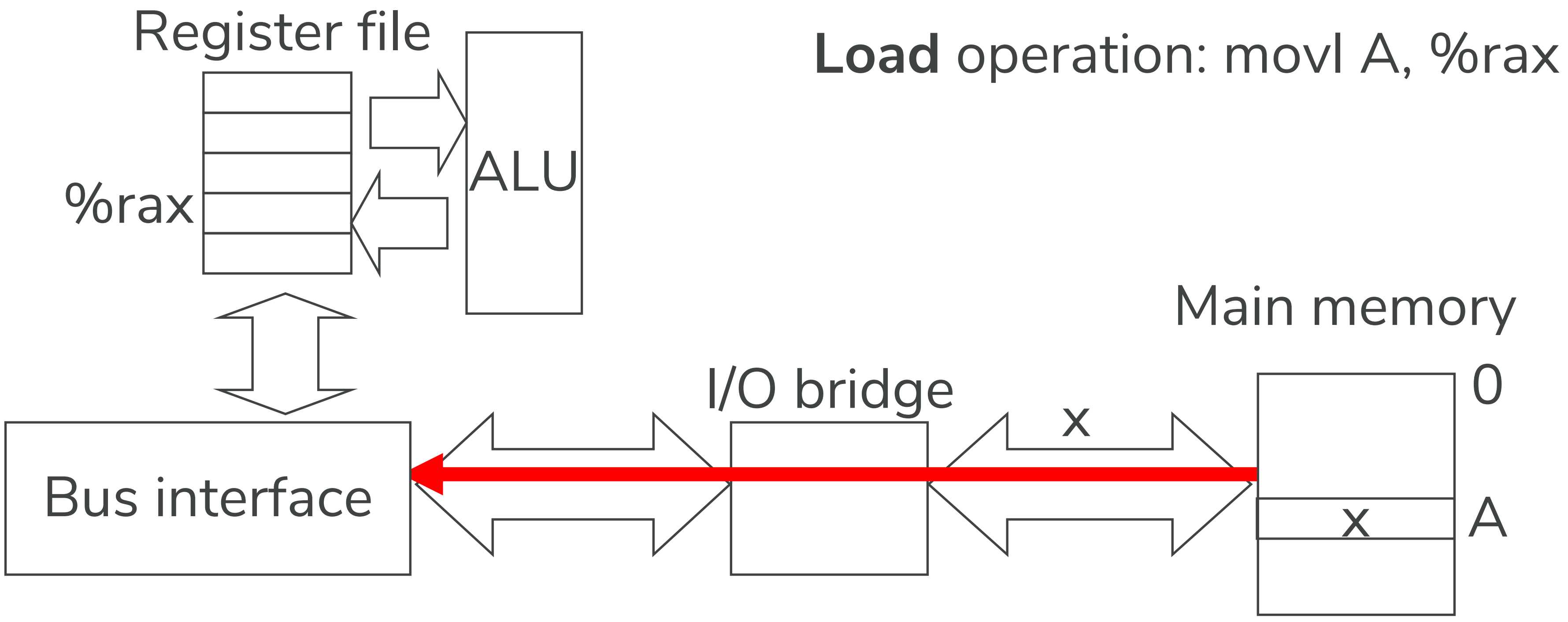
- CPU places address A on the memory bus.



21

Memory **READ** Transaction (2)

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

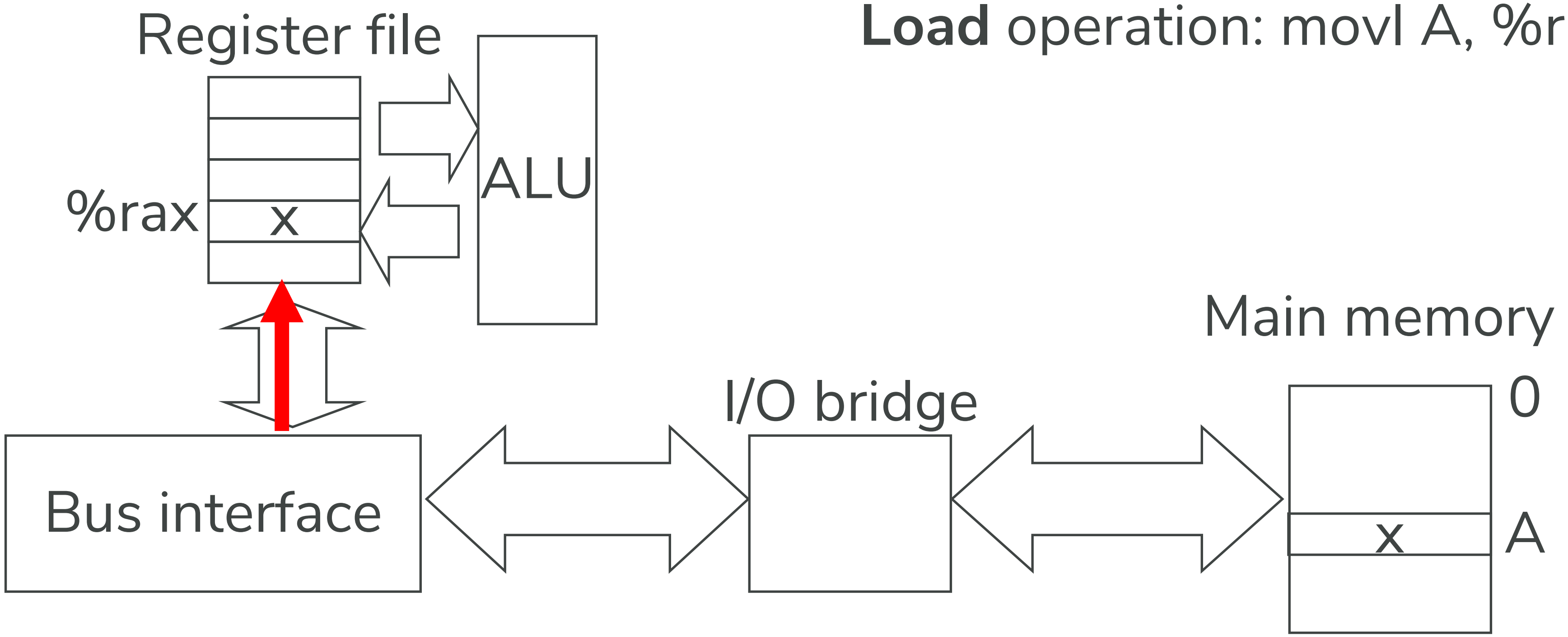


22

Memory **READ** Transaction (3)

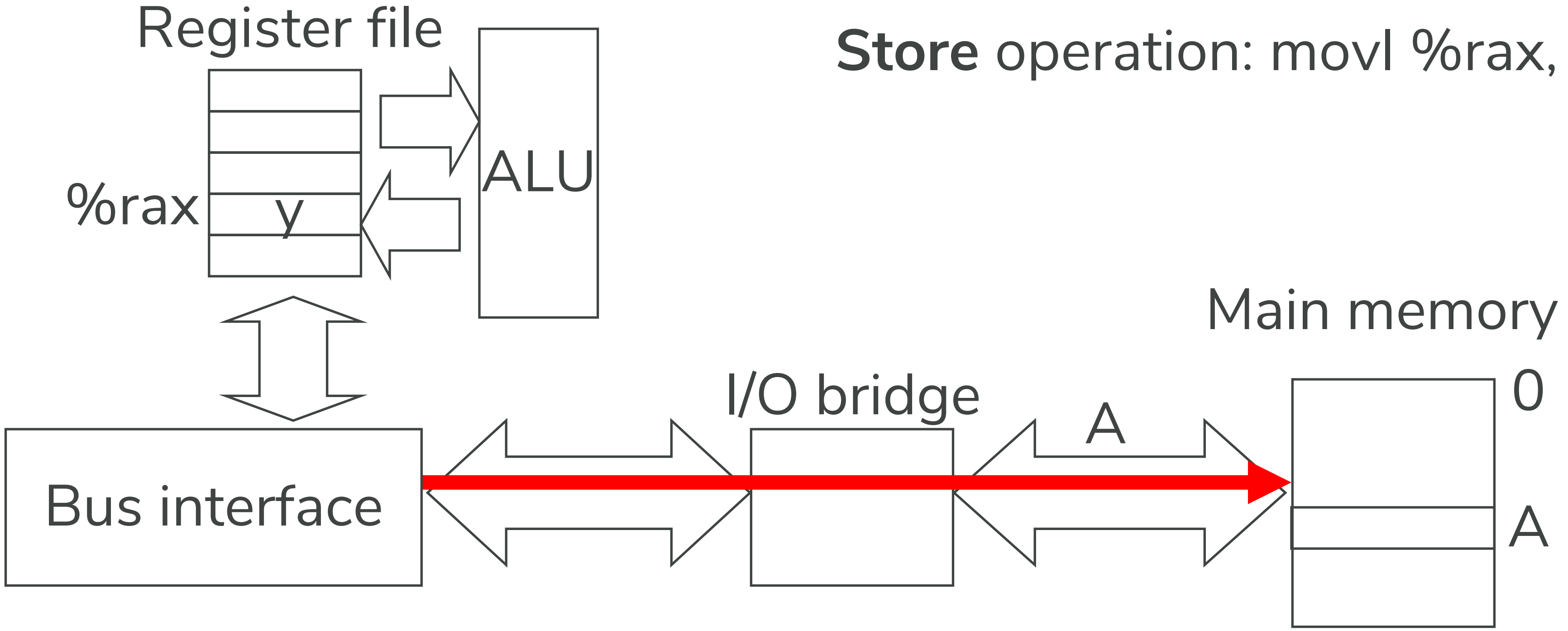
- CPU reads word x from the bus and copies it into register %rax.

Load operation: movl A, %rax



Memory **WRITE** Transaction (1)

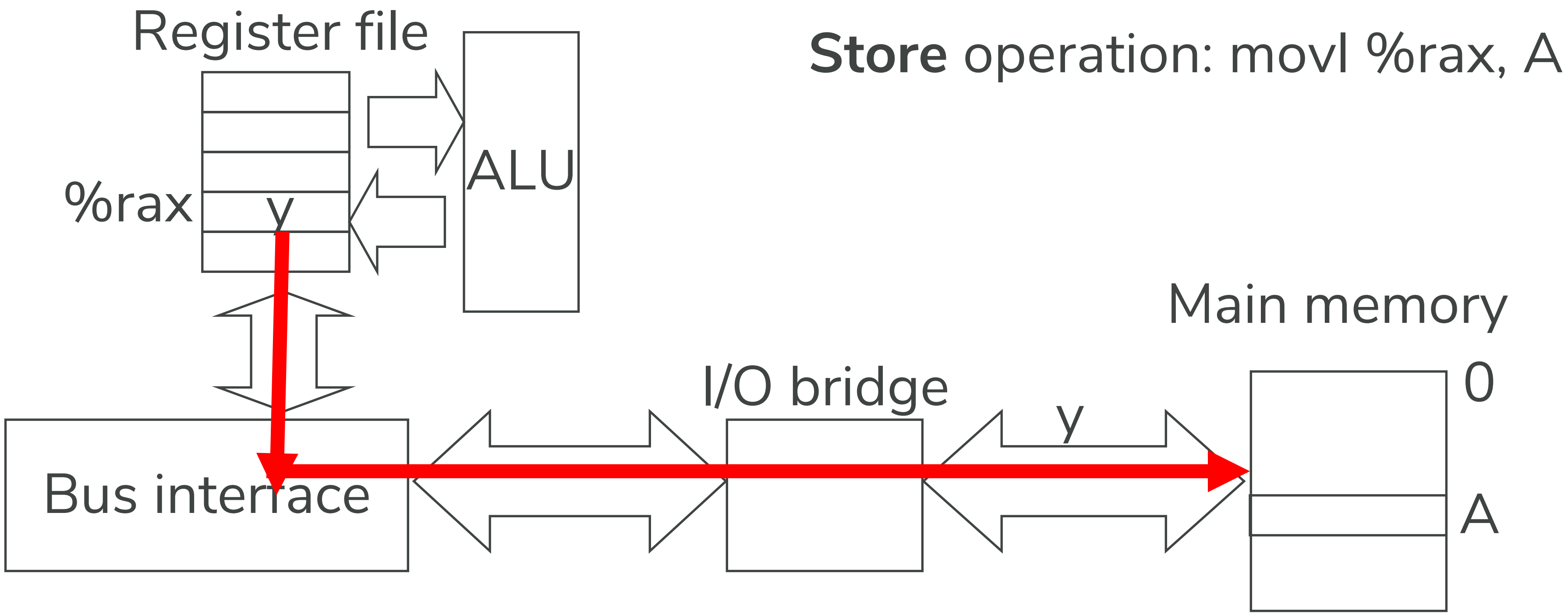
- CPU places address **A** on bus. Main memory reads it and waits for the corresponding data word to arrive.



Store operation: movl %rax, A

Memory **WRITE** Transaction (2)

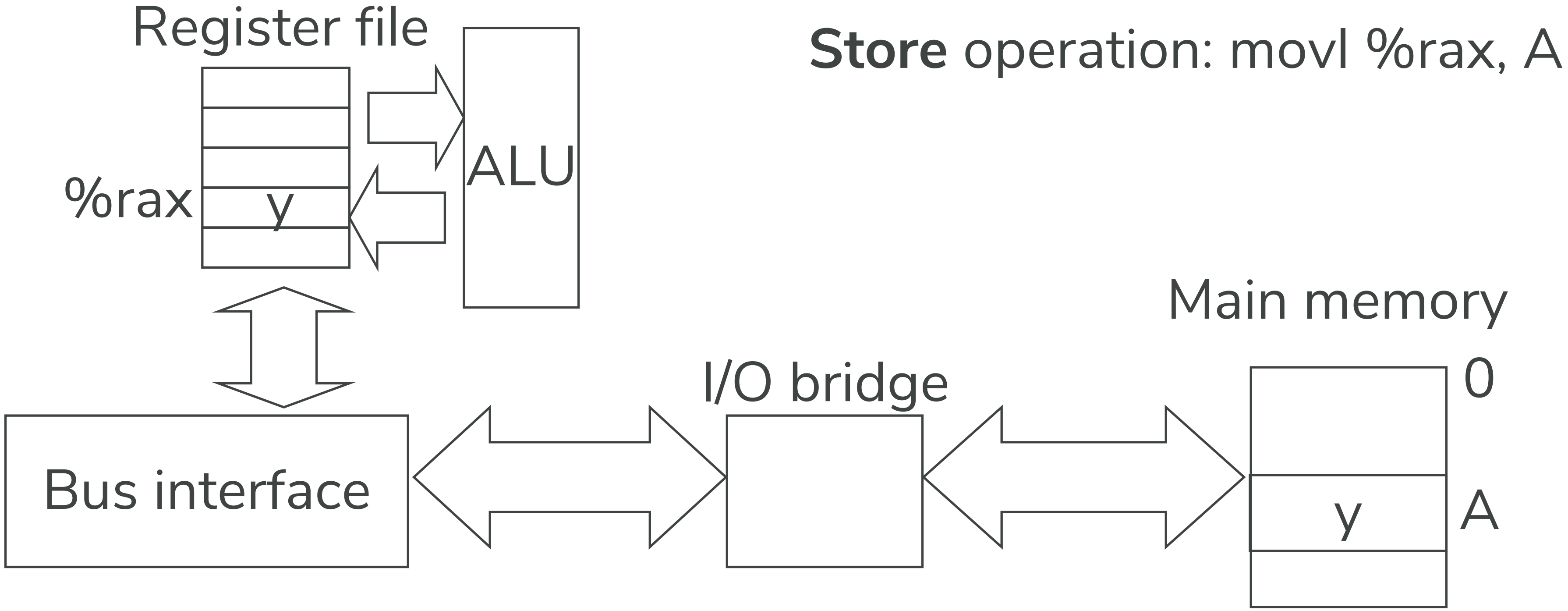
- CPU places data word **y** on the bus.



Store operation: movl %rax, A

Memory **WRITE** Transaction (3)

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



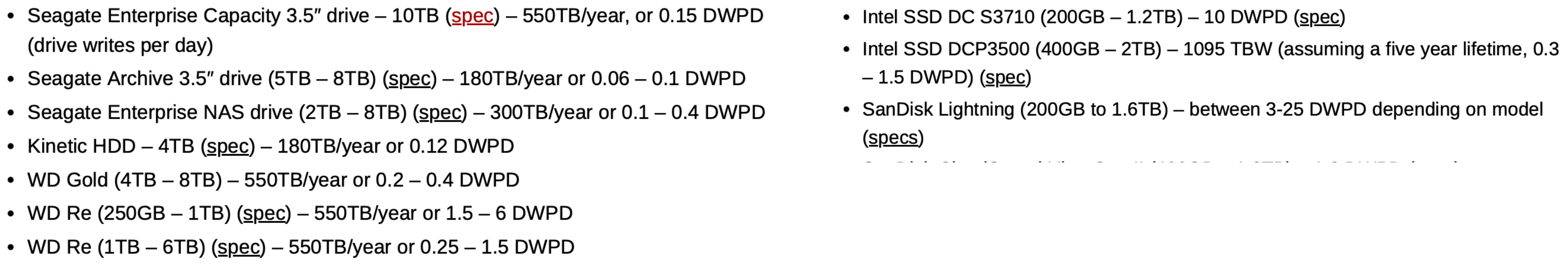
Store operation: movl %rax, A

Another type of Non-volatile Memory: Disk

- **Disks hold enormous amount of data** – on the order of hundreds to thousands of gigabytes compared to hundreds to thousands of megabytes in memory.
- **Disks are slower than RAM-based memory** – on the order of milliseconds to read info on (hard) disk, 100,000 times longer than from DRAM and 1,000,000 times longer than SRAM.
- **Solid-state disks (SSDs)** are faster than hard disk drives (HDD), but **slower** than DRAM and SRAM
- SSDs also have one disadvantage: limited number of writes! (Typical SSDs: ~100,000 write cycles). How about HDDs?

Another type of Non-volatile Memory: Disk

- **Write cycles' comparison:**
  - SSDs: typically **100,000** write cycles
  - NAND: **10,000** write cycles
  - 3D NAND: **35,000** write cycles.
  - How about HDDs? Typically unlimited write cycles. But we can still measure by DWPD = Drive Writes Per Day.



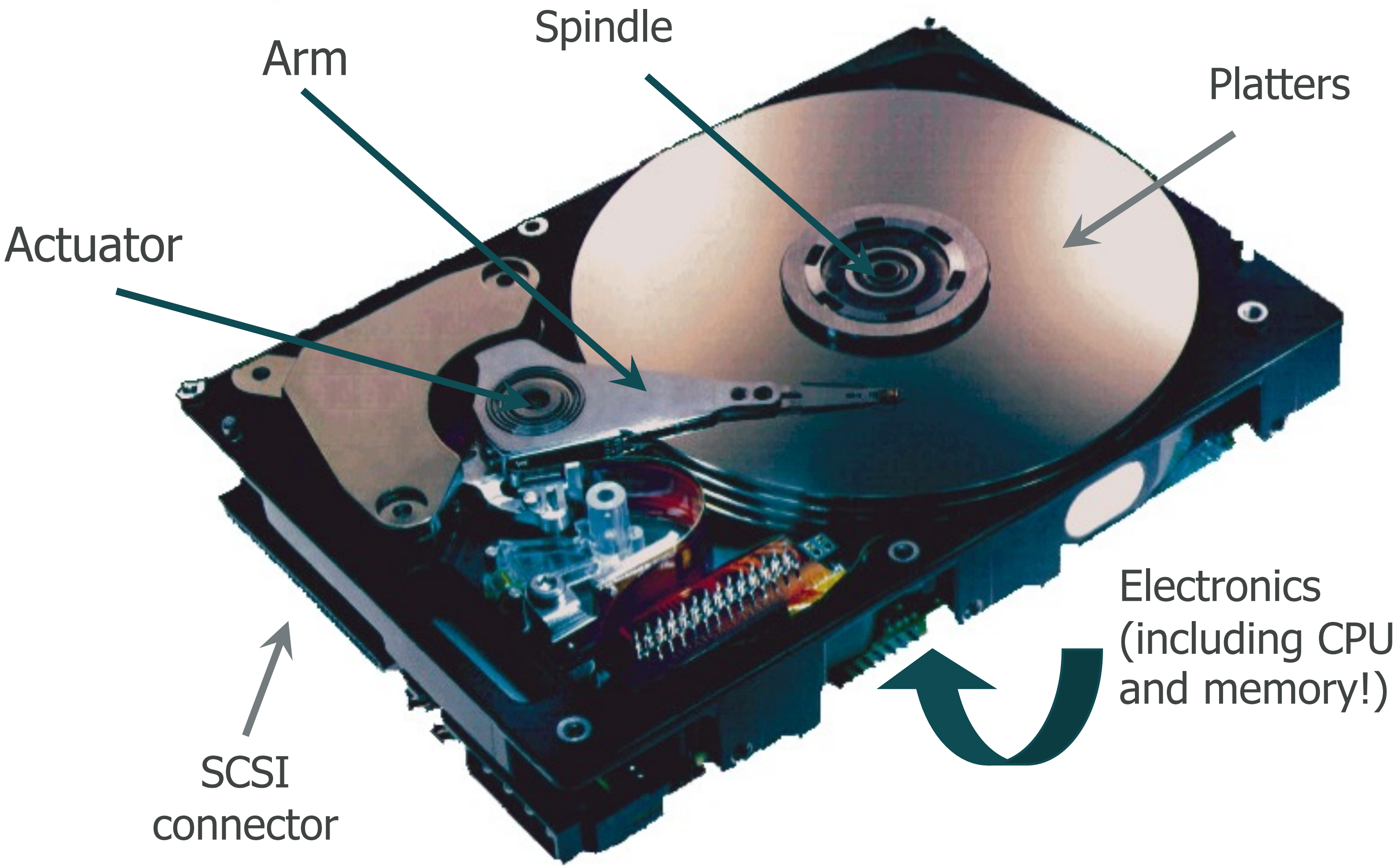
- Seagate Enterprise Capacity 3.5″ drive – 10TB (spec) – 550TB/year, or 0.15 DWPD (drive writes per day)
- Seagate Archive 3.5″ drive (5TB – 8TB) (spec) – 180TB/year or 0.06 – 0.1 DWPD
- Seagate Enterprise NAS drive (2TB – 8TB) (spec) – 300TB/year or 0.1 – 0.4 DWPD
- Kinetic HDD – 4TB (spec) – 180TB/year or 0.12 DWPD
- WD Gold (4TB – 8TB) – 550TB/year or 0.2 – 0.4 DWPD
- WD Re (250GB – 1TB) (spec) – 550TB/year or 1.5 – 6 DWPD
- WD Re (1TB – 6TB) (spec) – 550TB/year or 0.25 – 1.5 DWPD

- Intel SSD DC S3710 (200GB – 1.2TB) – 10 DWPD (spec)
- Intel SSD DCP3500 (400GB – 2TB) – 1095 TBW (assuming a five year lifetime, 0.3 – 1.5 DWPD) (spec)
- SanDisk Lightning (200GB to 1.6TB) – between 3-25 DWPD depending on model (specs)

- Source: https://www.techtarget.com/searchstorage/definition/write-cycle#:~:text=The%20number%20of%20write%20cycles,up%20to%20100%2C000%20write%20cycles.
- Source: https://www.theregister.com/2016/05/03/when_did_hard_drives_get_workload_rate_limits/

28

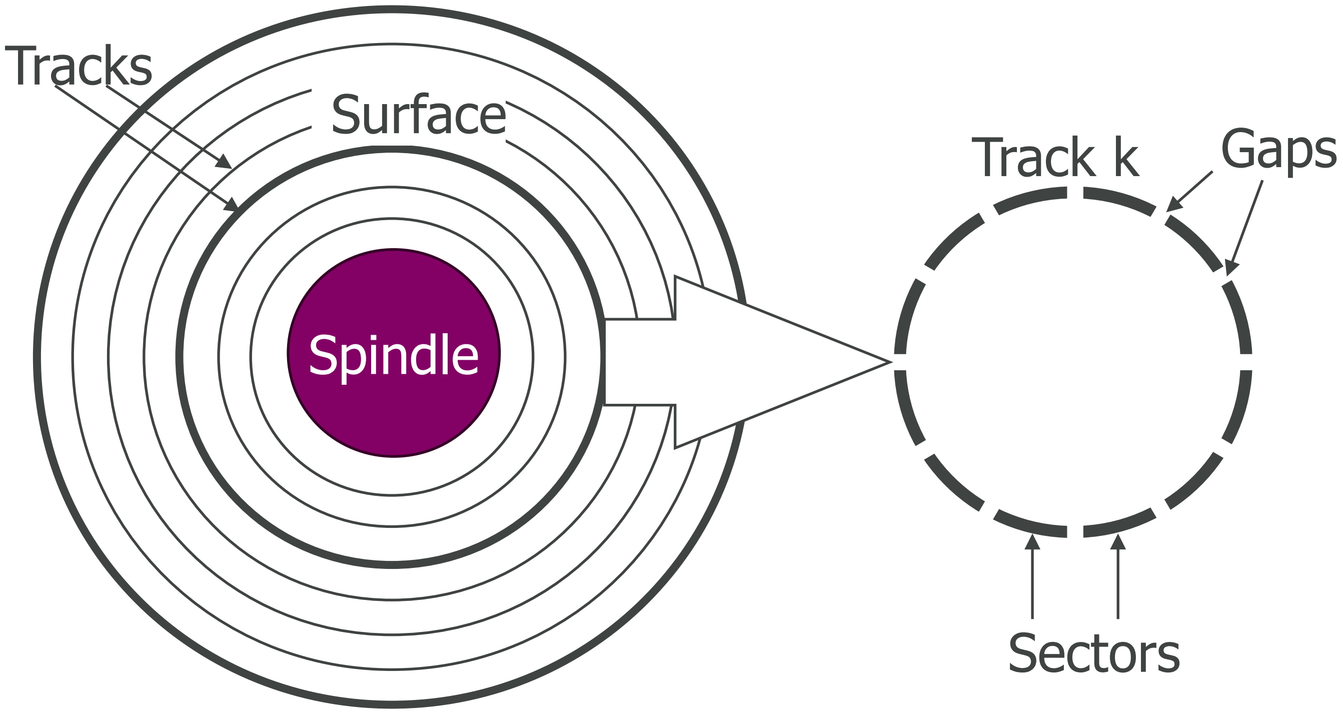
Anatomy of A Disk Drive



Source: Seagate Technology

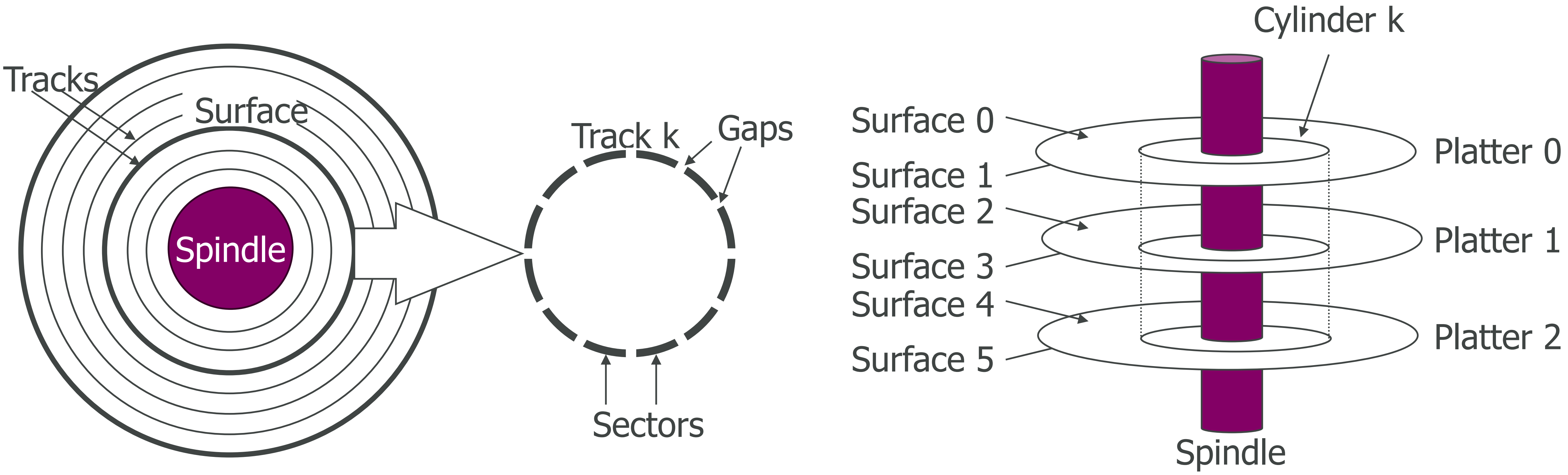
Disk Geometry

- Disks consist of **platters**, each with two **surfaces**.
- Each surface consists of concentric rings called **tracks**.
- Each track consists of **sectors** separated by **gaps**.



Disk Geometry (Multi-Platter View)

- Aligned tracks form a **cylinder**.



Disk Capacity (1)

Capacity defined to be the maximum number of bits that can be recorded on a disk. Determined by the following factors:

- **Recording density (bits/in):** The number of bits on a 1-inch segment of a track.
- **Track density (tracks/in):** The number of tracks on a 1-inch segment of radius extending from the center of the platter.
- **Areal density (bits/in^2):** product of recording density and track density

Disk Capacity (2)

Determining areal density:

- Original disks partitioned every track into the same number of sectors, which was determined by the innermost track. Resulted in sectors being spaced further apart on outer tracks.
- Modern disks partition into disjoint subsets called **recording zones**.
  - Each track within zone same number of sectors, determined by the innermost track.
  - Each zone has a different number of sectors/track.

How to compute Disk Capacity?

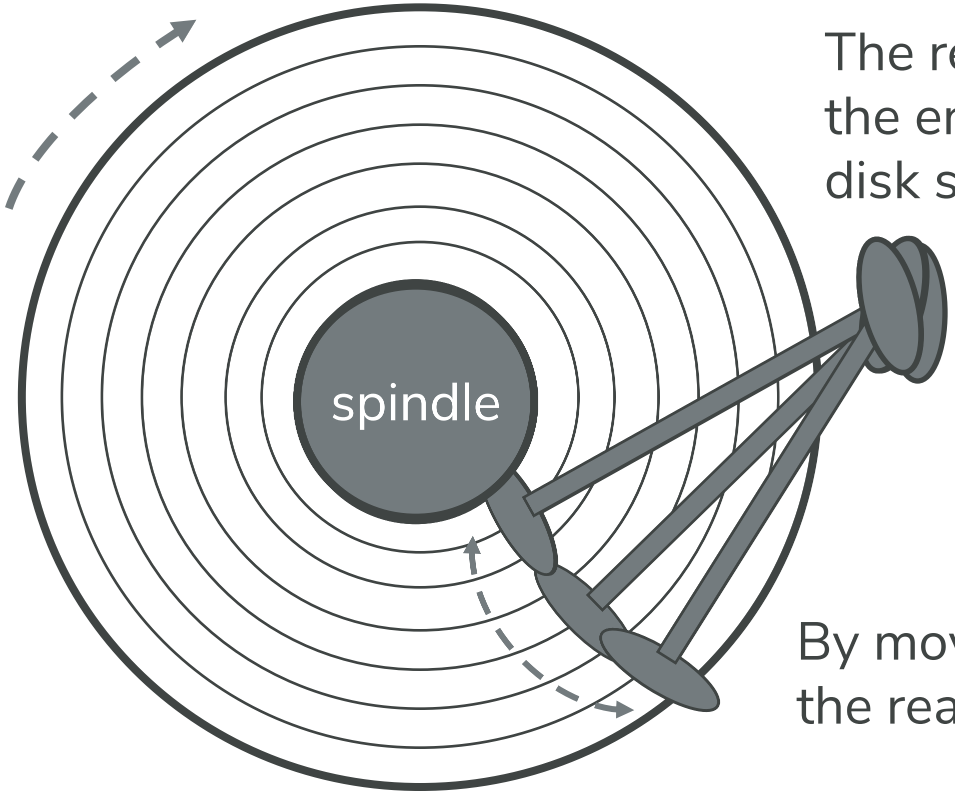
Capacity = (#bytes/sector) × (avg #sectors/track) × (#tracks/surface) × (#surfaces/platter) × (#platters/disk)

Example:

- 512 bytes/sector
- Average of 300 sectors/track
- 20,000 tracks/surface
- 2 surfaces/platter
- 5 platters/disk

Capacity = 512 × 300 × 20,000 × 2 × 5 = 30,720,000,000 = 30.72 GB.

Disk Operation (Single-Platter View)

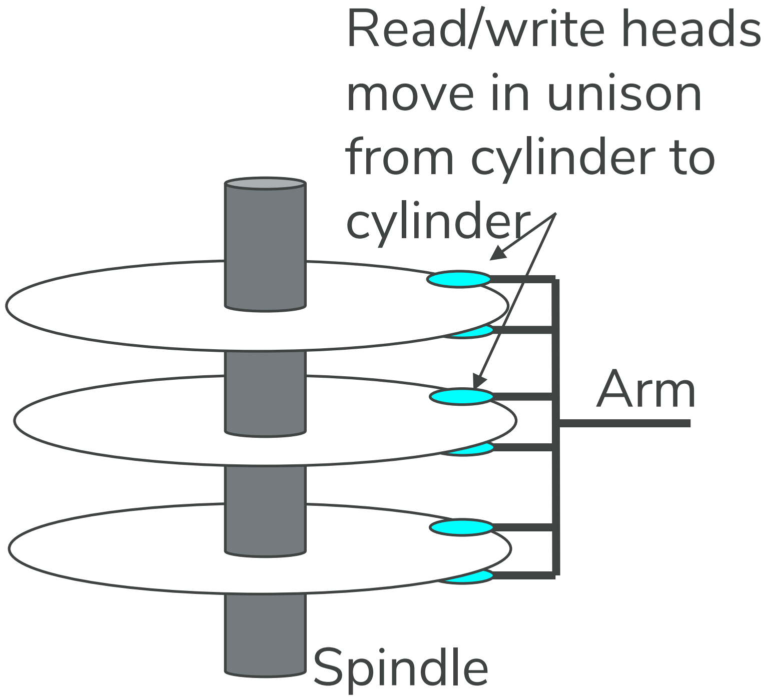


The disk surface spins at a fixed rotational rate

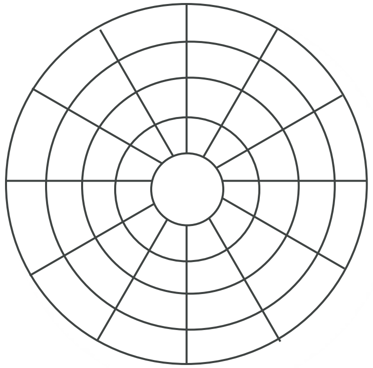
The read/write head is attached to the end of the arm and flies over the disk surface on a thin cushion of air.

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

Disk Operation (Multi-Platter View)



Disk Structure: Top view of a Single Platter



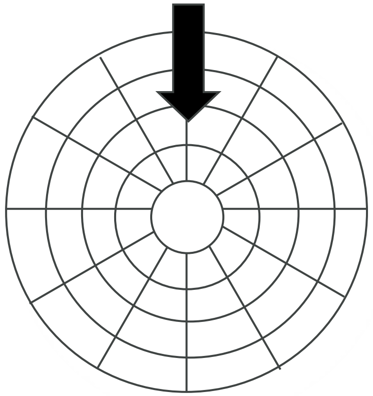
Surface organized into tracks

Tracks divided into sectors

7200RPM Hard Drive running without the cover:
https://www.youtube.com/watch?v=ZBx4H908_xI

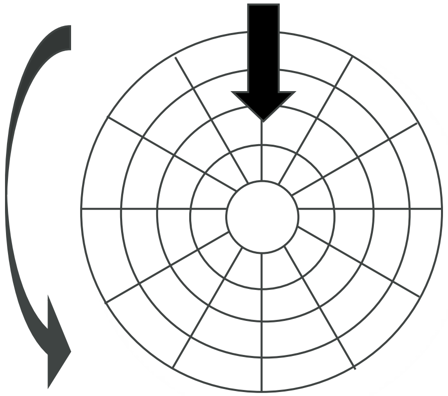
If you do not see the multiple platters in this video, then check this out:
https://www.youtube.com/watch?v=w3bt5yNt3xY

Disk Access (step 1)



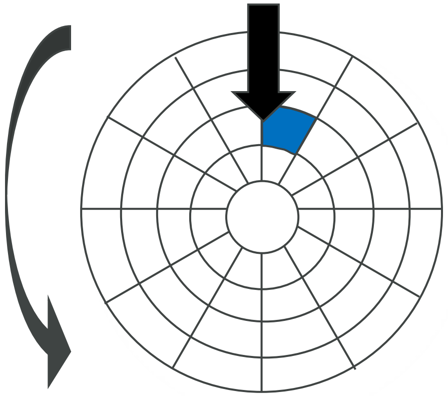
Head in position above a track

Disk Access (step 2)



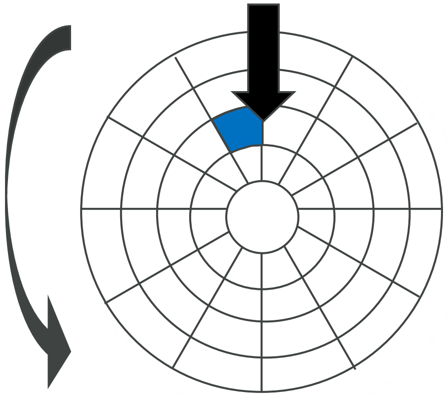
Rotation is counter-clockwise

Disk Access: Read (1-A)



About to read **BLUE** sector

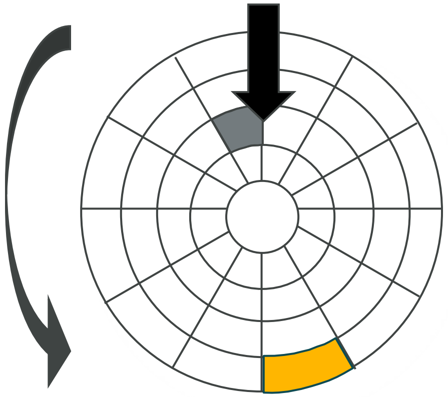
Disk Access: Read (1-B)



After **BLUE** read

After reading **BLUE** sector

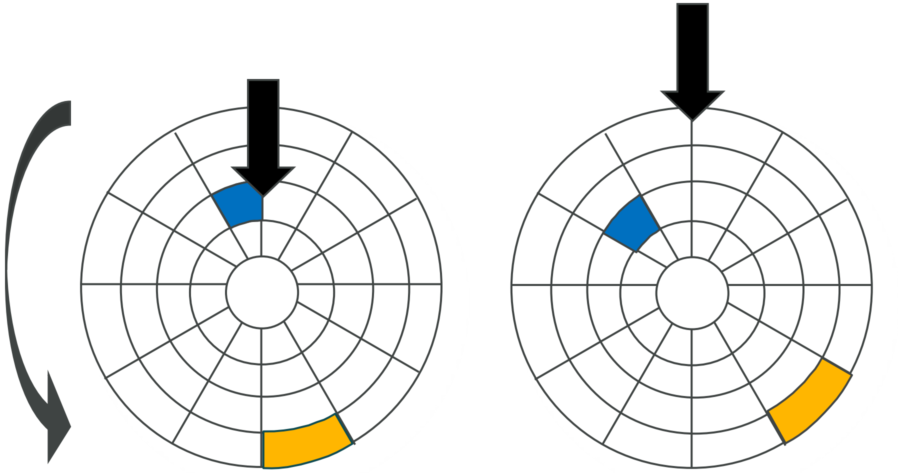
Disk Access: Read (1-C)



After **BLUE** read

ORANGE request scheduled next

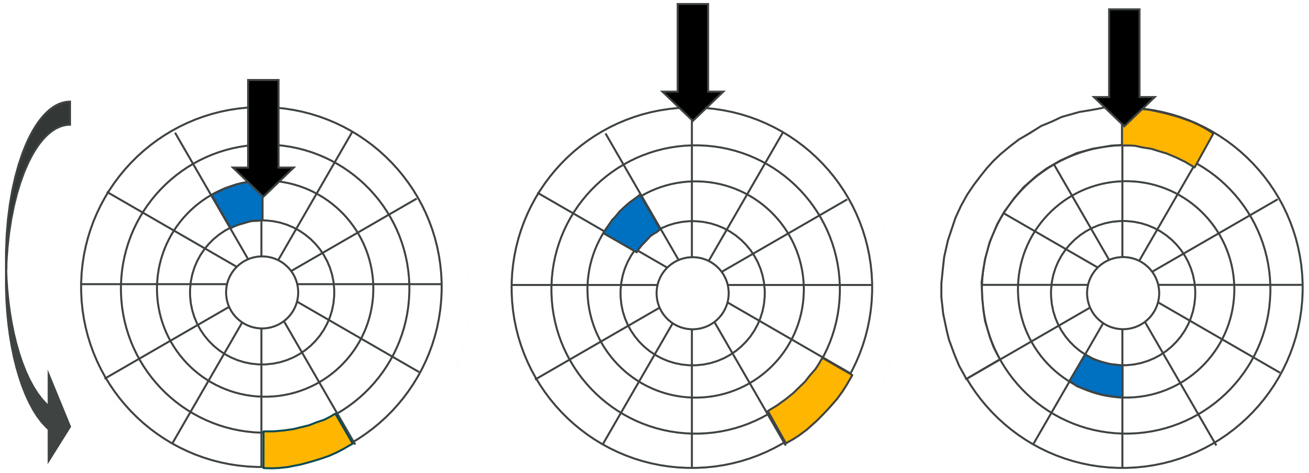
Disk Access: Seek



After **BLUE** read Seek for **ORANGE**

Seek to **ORANGE's** track

Disk Access: Rotational Latency



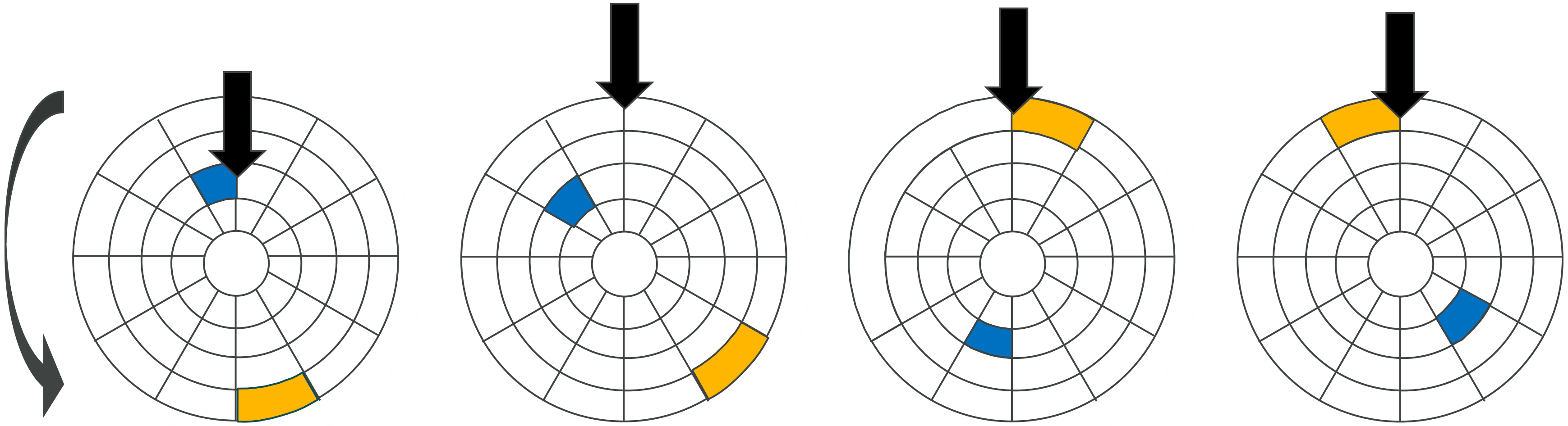
After **BLUE** read

Seek for **ORANGE**

Rotational latency

Wait for **Orange** sector to rotate around

Disk Access: Read (2)



After **BLUE** read

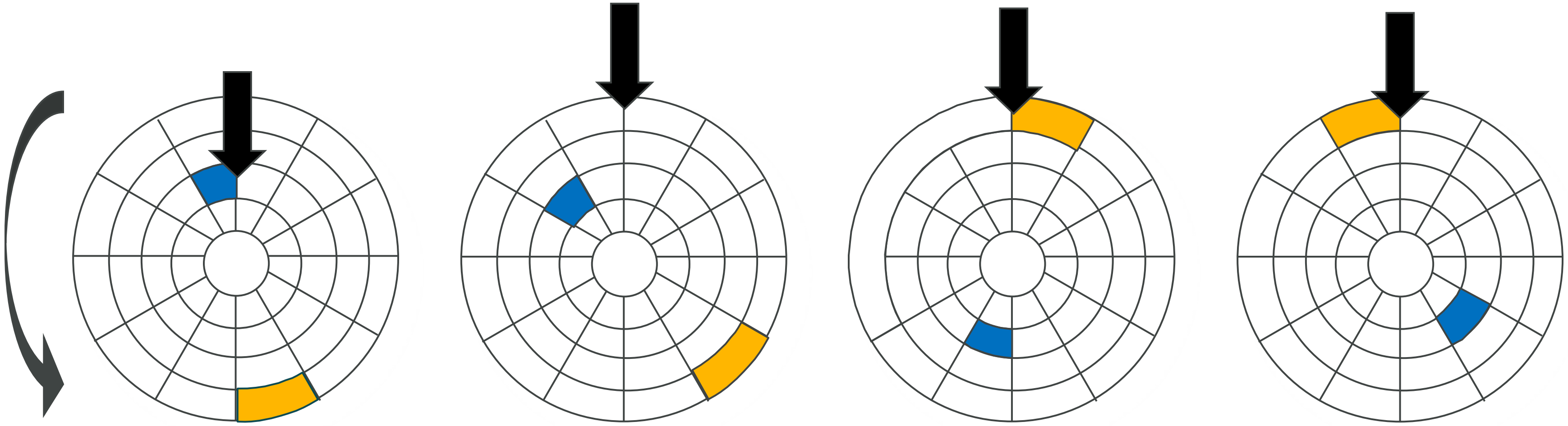
Seek for **ORANGE**

Rotational latency

After **ORANGE** read

Complete read of **ORANGE** sector

Disk Access: Service Time Components



After **BLUE** read

Seek for **ORANGE**

Rotational latency

After **ORANGE** read

① Data transfer

② Seek

③ Rotational latency

④ Data transfer

Calculating Access Time (1)

Average access time for a sector:

$$T_{access} = T_{avg\_seek} + T_{avg\_rotation} + T_{avg\_transfer}$$

Seek time (T_{avg_seek}):

- Time to position heads over cylinder
- Typical T_{avg_seek} is 3–9 msec, 20-msec maximum

Rotational latency ($T_{avg\_rotation}$):

- After head is positioned over track, the time it takes for the first bit of the sector to pass under the head
- Worst case: head just misses the sector and waits for the disk to rotate 360

$$T_{max\_rotation} = (1/\text{RPM}) \times (60 \text{ secs}/1 \text{ min})$$

- Average case is **half** of worst case:

$$T_{avg\_rotation} = (1/2) \times (1/\text{RPM}) \times (60 \text{ secs}/1 \text{ min})$$

- Typical $T_{avg\_rotation}$ = 5400–7200 RPM

Calculating Access Time (2)

Transfer time ($T_{avg\_transfer}$):

- Time to read bits in the sector
- Time depends on rotational speed, number of sectors per track.
- Estimate of the average transfer time:
  - $T_{avg\_transfer}$ = (1/RPM) x (1/(avg #sectors/track)) × (60 secs/1 min) × (1000 msec/1 sec)

Example:

- Rotational rate = 7200 RPM
- Average seek time = 9 msec
- Avg #sectors/track = 400

$T_{avg\_rotation}$ = 1/2 × (60 secs/7200 RPM) × (1000 msec/sec) = **4 msec**

$T_{avg\_transfer}$ = (60/7200 RPM) × (1/400 secs/track) × (1000 msec/sec) = **0.02 msec**

T_{access} = 9 msec + 4 msec + 0.02 msec = **13.02 msec**

Access Time

Time to access the **512 bytes** in a disk sector is dominated by the seek time (**9 msec**) and rotational latency (**4 msec**).

Accessing the sector takes a long time but transferring bits is basically ***free***.

Since seek time and rotational latency are roughly the same, at least same order of magnitude, doubling the seek time is a reasonable estimate for access time.

Comparison of access times of various storage devices when reading a comparable 512-byte sector sized block:

- SRAM: 256 **nanoseconds** (nsec)
- DRAM: 5000 **nsec**
- Disk: 10 **msec**
- Disk is ~40,000 times slower than SRAM, 2,500 times slower than DRAM.

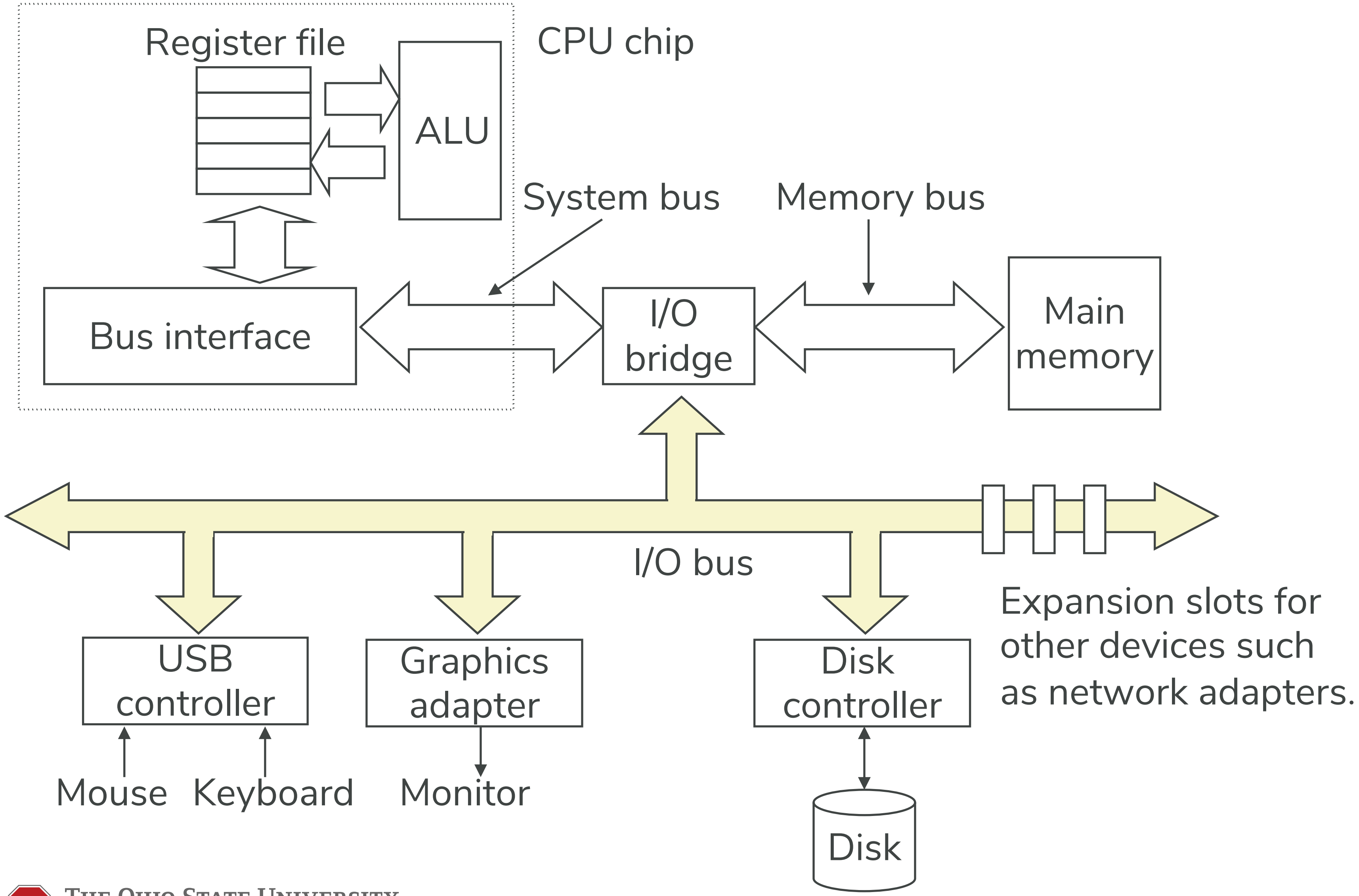
Logical Disk Blocks and Formatted Disk Capacity

- **Logical Disk Blocks**
  - Although modern disks have complex geometries, they present a simpler abstract view as a sequence of *B* sector-sized logical blocks, numbered 0, 1, 2, ... *B* – 1.
  - Disk controller maintains the mapping between the logical and actual (physical) disk sectors and converts requests for block into a surface, track, and sector by performing a fast table lookup.
- **Formatted Disk Capacity**
  - Before disks can be used for the first time, they must be formatted by the disk controller.
  - Gaps between sectors filled in with info to identify sectors.
  - Finds surface defects and sets aside cylinders to be used for spares.
  - Formatted capacity is less than the maximum capacity.

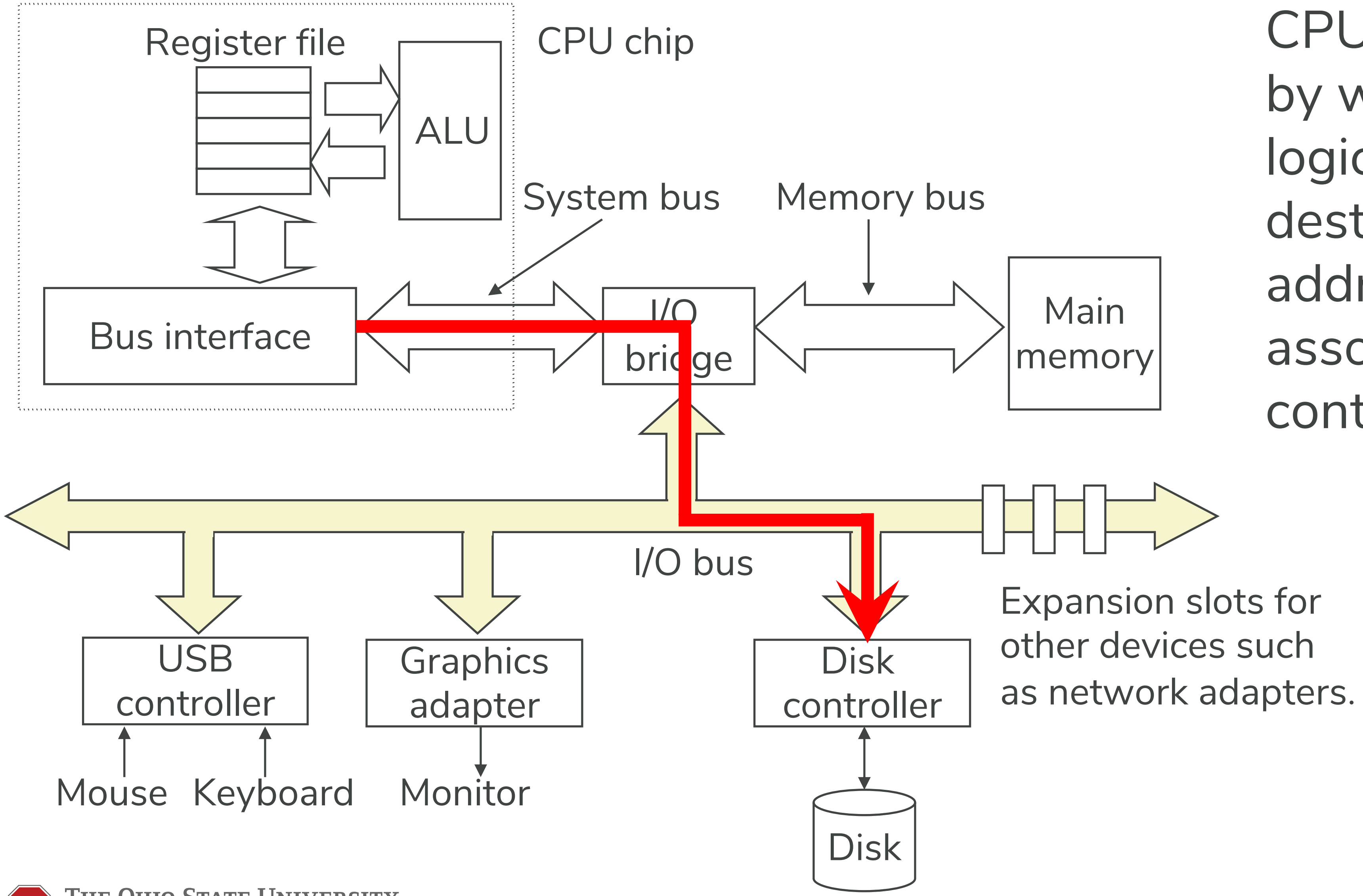
Connecting I/O Devices

- I/O devices such as disks, graphics cards, monitors, mice, and keyboards connect to the CPU and main memory via an **I/O bus.**
- Unlike the system bus and memory bus which are CPU specific, the I/O bus is independent of the underlying CPU.
- The I/O bus is slower than the system and memory buses but can accommodate a wide variety of third-party I/O devices. For instance, USB, graphics card or adapter, host bus adapter (SCSI/SATA).
- Network adapters can be connected to the I/O bus by plugging the adapter into an empty expansion slot on the motherboard.

I/O Bus

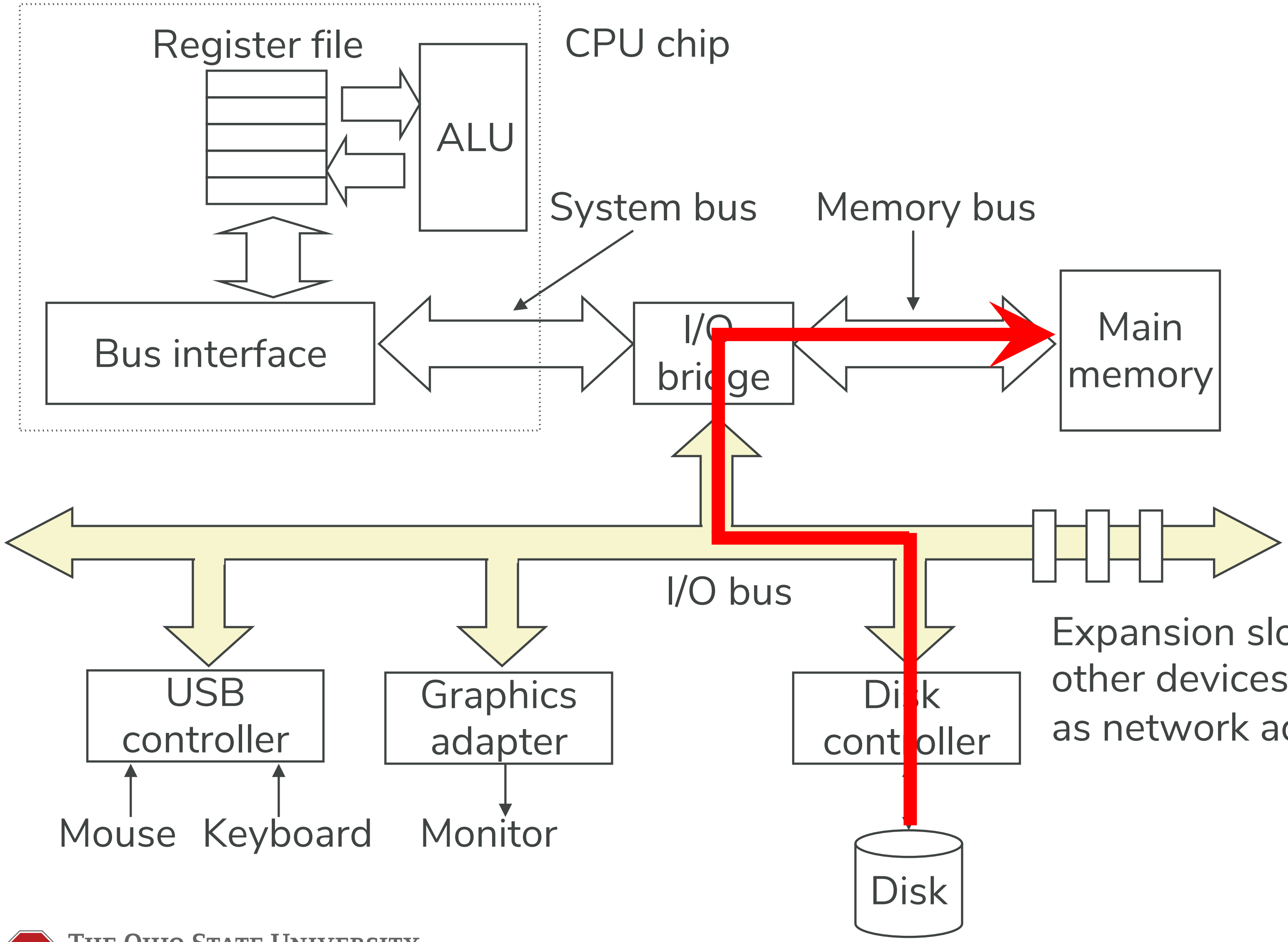

CPU chip
Register file
ALU
Bus interface
System bus
Memory bus
I/O bridge
Main memory
I/O bus
USB controller
Graphics adapter
Disk controller
Mouse
Keyboard
Monitor
Disk
Expansion slots for other devices such as network adapters.

Reading a Disk Sector (1)



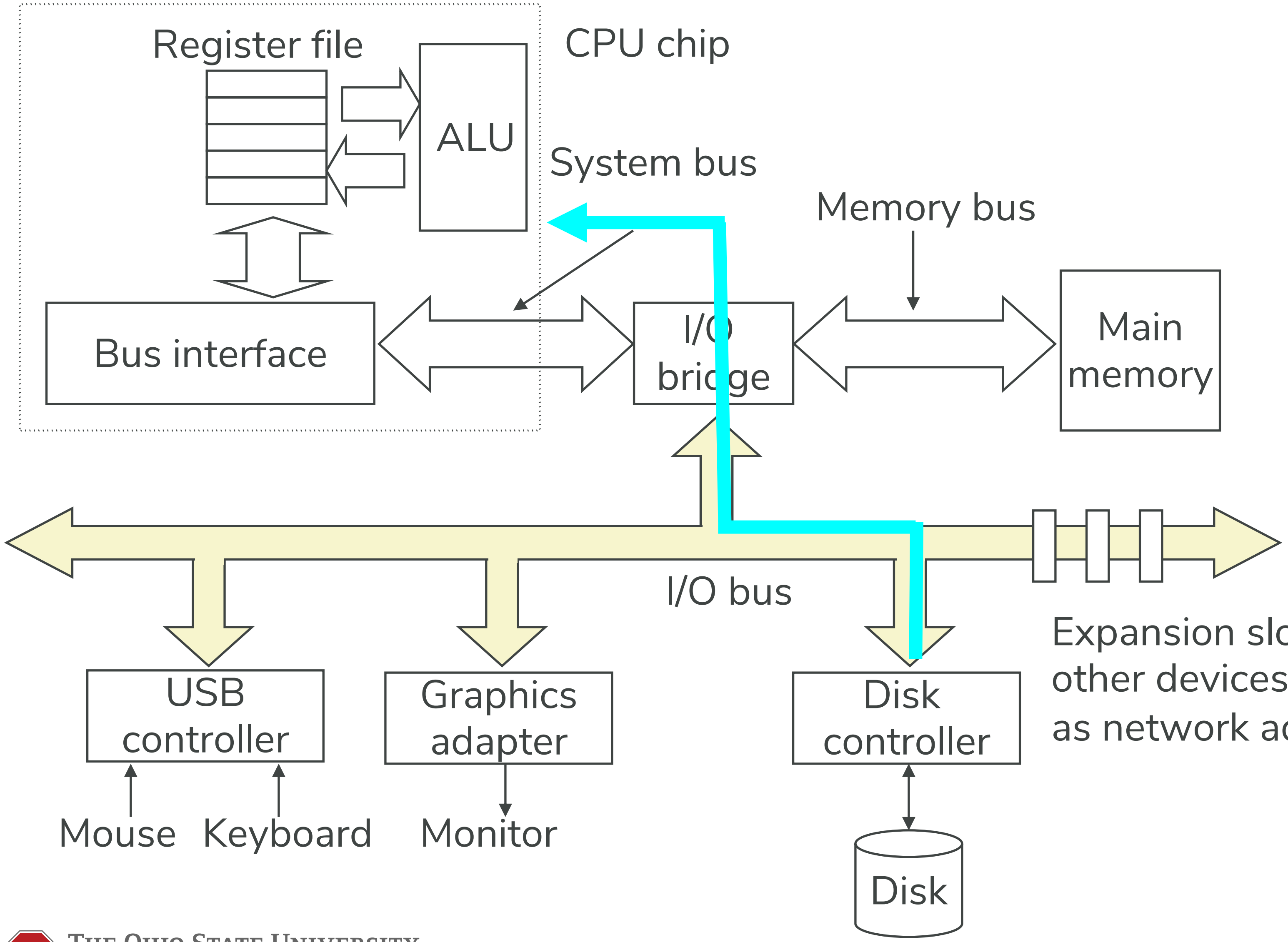
CPU initiates a disk read by writing a command, logical block number, and destination memory address to a port (address) associated with disk controller.

Reading a Disk Sector (2)



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

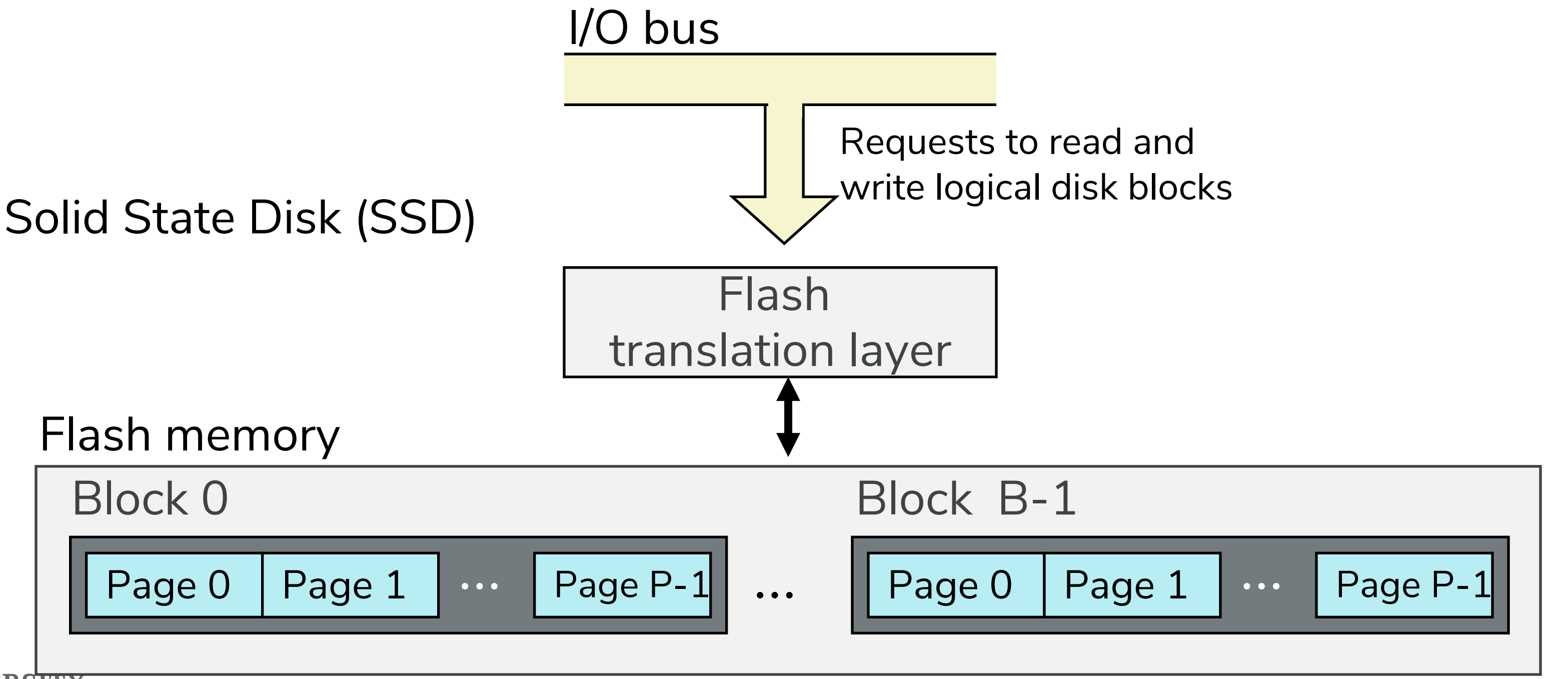
Reading a Disk Sector (3)



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 Drive/Disks (SSDs)

- Pages: 512 KB to 4 KB, 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

Description of Statistic	Speed, Time	Description of Statistic	Speed, Time
Sequential read throughput	550 MB/sec	Sequential write throughput	470 MB/sec
Random read throughput	365 MB/sec	Random write throughput	303 MB/sec
Random read access	50 μsec	Random write access	60 μsec

Source: Intel SSD 730 product specification

- Why are random writes so slow?
  - Erasing a block is slow (around 1 msec)
  - Write to a page triggers a copy of all useful pages in the block
    - Find a used block (new block) and erase it
    - Write the page into the new block
    - Copy other pages from old block to the new block

SSDs vs Rotating (Hard) Disks

- **Advantages:**
  - No moving parts (semiconductor memory); more rugged
  - Much faster random-access times
  - Use less power
- **Disadvantages:**
  - SSDs wear out with usage
  - More expensive than hard disks

Storage Technology Trends

SRAM

Metric	1985	1990	1995	2000	2005	2010	2015	2015:1985
$/MB	2,900	320	256	100	75	60.0	32	116
Access (nsec)	150	35	15	3	2	1.5	2	115

DRAM

Metric	1985	1990	1995	2000	2005	2010	2015	2015:1985
$/MB	880.00	100	30	1	0.1	0.06	0.02	44,000
Access (nsec)	200.00	100	70	60	50.0	40.00	20.00	10
Typical Size (MB)	0.25	4	16	64	2,000.0	8,000.00	16,000.00	62,500

Disk

Metric	1985	1990	1995	2000	2005	2010	2015	2015:1985
$/GB	100,000.00	8,000.00	300	10	5	0.3	0.03	3,333,333
Access (msec)	75.00	28.00	10	8	5	3.0	3.00	25
Typical Size (GB)	0.01	0.16	1	20	160	1,500.0	3,000.00	300,000

CPU Trends

Inflection point in computer history
when designers hit the "Power Wall"

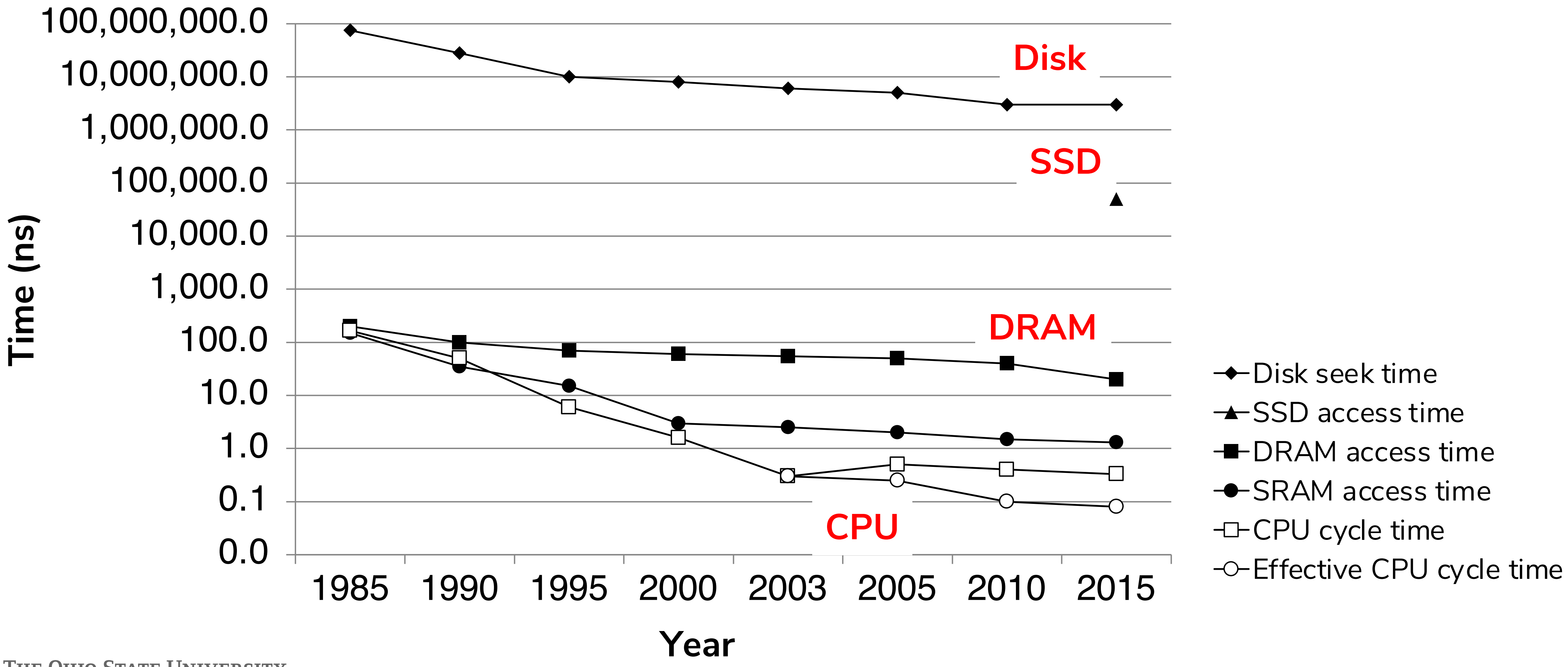
	1985	1990	1995	2000	2003	2005	2010	2015	2015:1985
CPU	80286	80386	Pentium	P-III	P-4	Core 2	Core i7 (N)	Core i7 (H)	—
Clock rate (MHz)	6	20	150	600.0	3,300.0	2,000.00	2,500.0	3,000.00	500
Cycle time (nsec)	166	50	6	1.6	0.3	0.50	0.4	0.33	500
Cores	1	1	1	1.0	1.0	2.00	4.0	4.00	4
Effective cycle time (nsec)	166	50	6	1.6	0.3	0.25	0.1	0.08	2,075

* (N) indicates Intel's Nehalem architecture; (H) indicates Intel's Haswell architecture.

- Around 2003, system designers reached a limit regarding the exploitation of **instruction-level parallelism (ILP)** in sequential programs.
- Since 2000, processor speed has not greatly increased; instead, **multicore CPUs**.

The CPU-Memory Gap

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



Summary of Storage Technologies

- Storage Technologies in Computer have a lot of diversities. Two main types:
  - Volatile memory (DRAM, SRAM)
  - Non-Volatile Memory (ROM, Flash-Drive)
  - Non-Volatile Memories for storage: HDDs, SSDs
- No matter what storage technologies are chosen, one thing we have to care is the CPU Performance. Without an improvement for CPU, even when we have the fastest SSDs in the world it's still useless.