CS429: Computer Organization and Architecture

Storage Technologies

Warren Hunt, Jr. and Bill Young
Department of Computer Sciences
University of Texas at Austin

Last updated: November 13, 2014 at 14:24

Random-Access Memory (RAM)

- Key Features
  - RAM is packaged as a chip
  - The basic storage unit is a cell (one bit per cell)
  - Multiple RAM chips form a memory.
- Static RAM (SRAM)
  - Each cell stores a bit with a 6-transistor circuit.
  - Retains value indefinitely, as long as kept powered.
  - Relatively insensitive to disturbances such as electrical noise.
  - Faster but more expensive than DRAM.
- Dynamic RAM (DRAM)
  - Each cell stores a bit with a capacitor and transistor.
  - Value must be refreshed every 10–100 ms.
  - Sensitive to disturbances, slower and cheaper than SRAM

Random-Access Memory (RAM) (2)

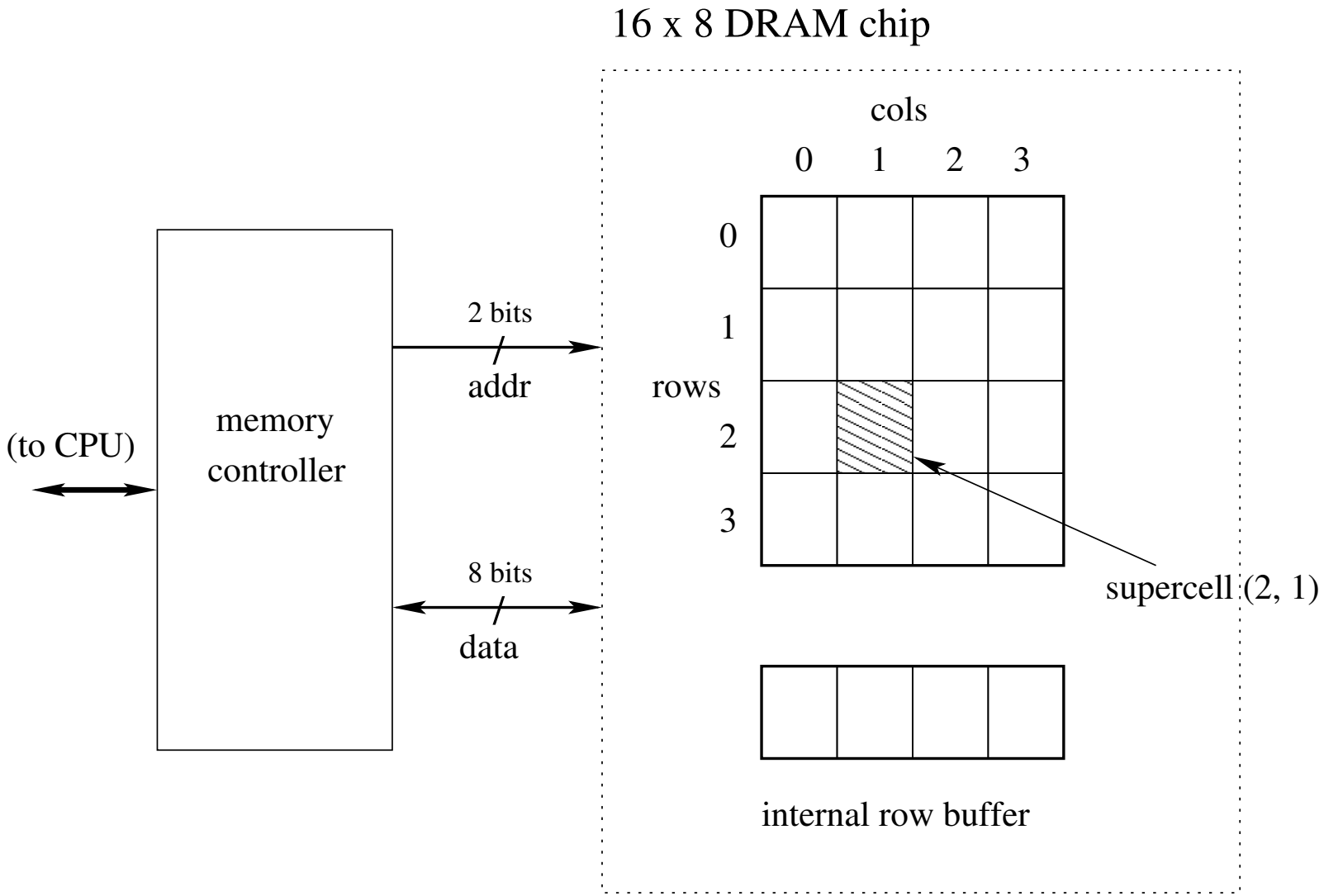
- Flash RAM (what's in your ipod and cell phone)
  - Each cell stores 1 or more bits on a "floating-gate" capacitor
  - Keeps state even when power is off
  - As cheap as DRAM, but much slower

RAM Summary

Type	Trans. per bit	Access time	Persist?	Sensitive	Cost	Applications
SRAM	6	1X	No	No	100X	cache memory
DRAM	1	10X	No	Yes	1X	main memory
Flash	1/2–1	10000X	Yes	No	1X	disk substitute

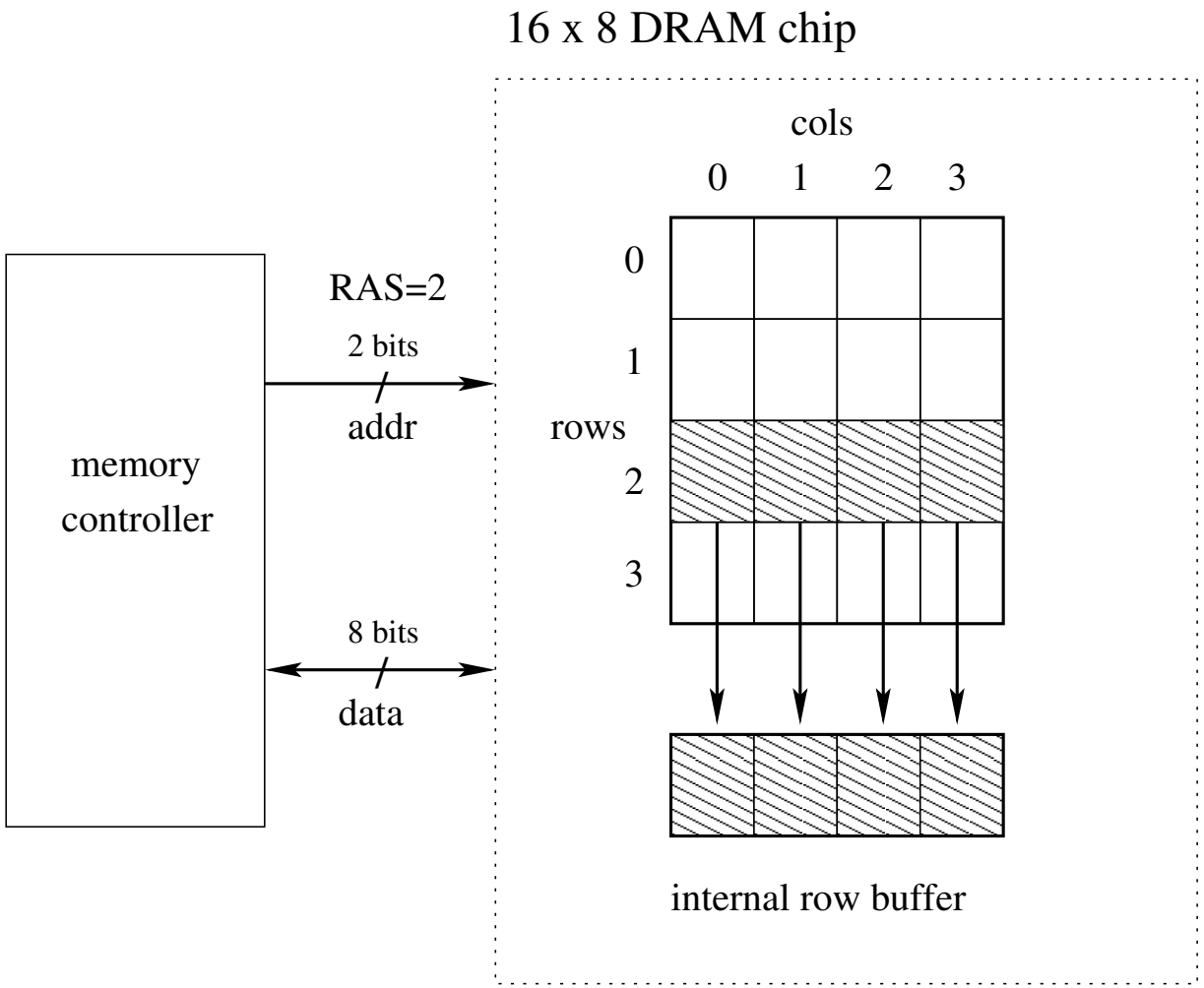
Conventional DRAM Organization

DRAM is typically organized as a $d \times w$ array of d supercells of size w bits.



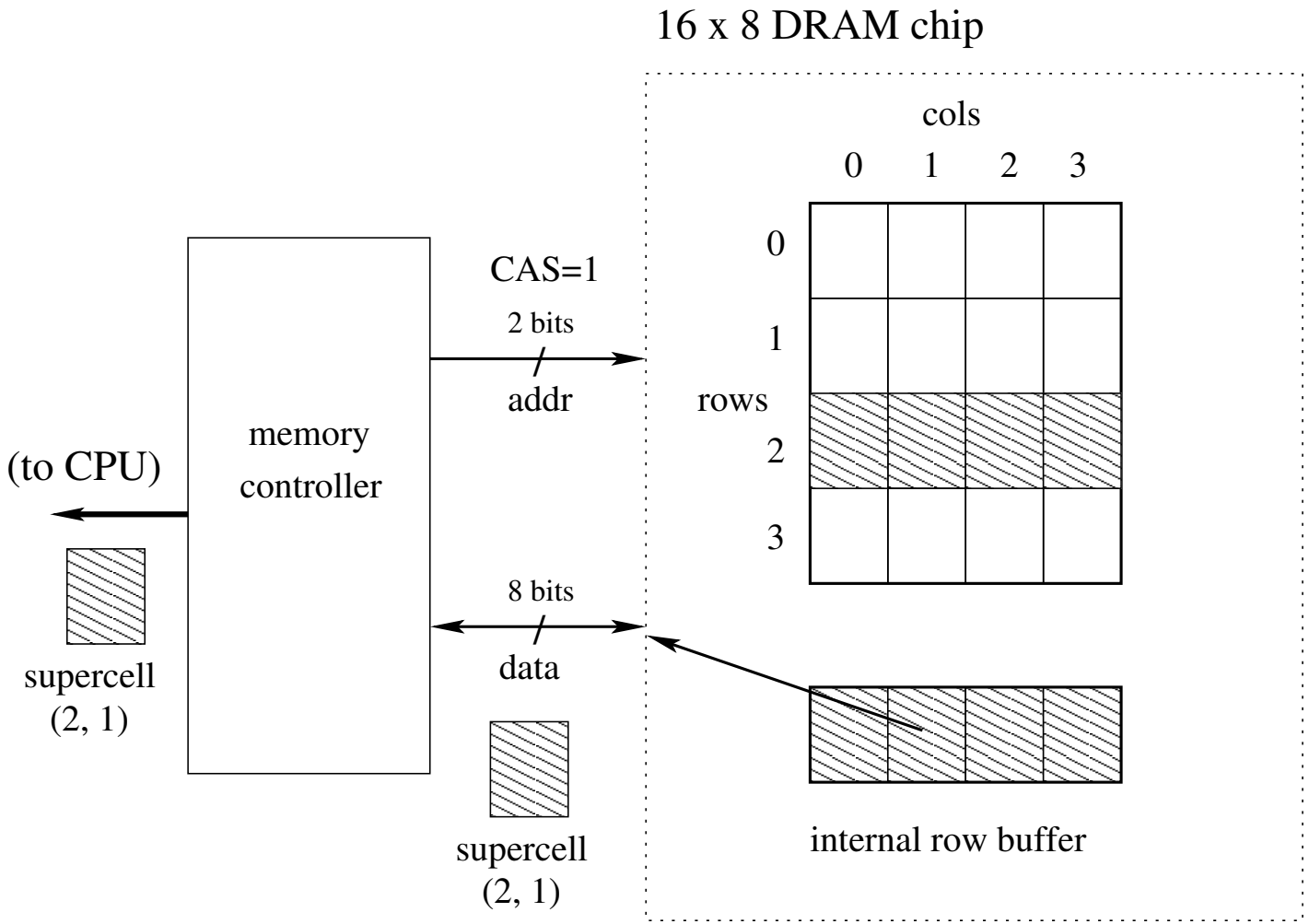
Reading DRAM Supercell (2, 1)

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

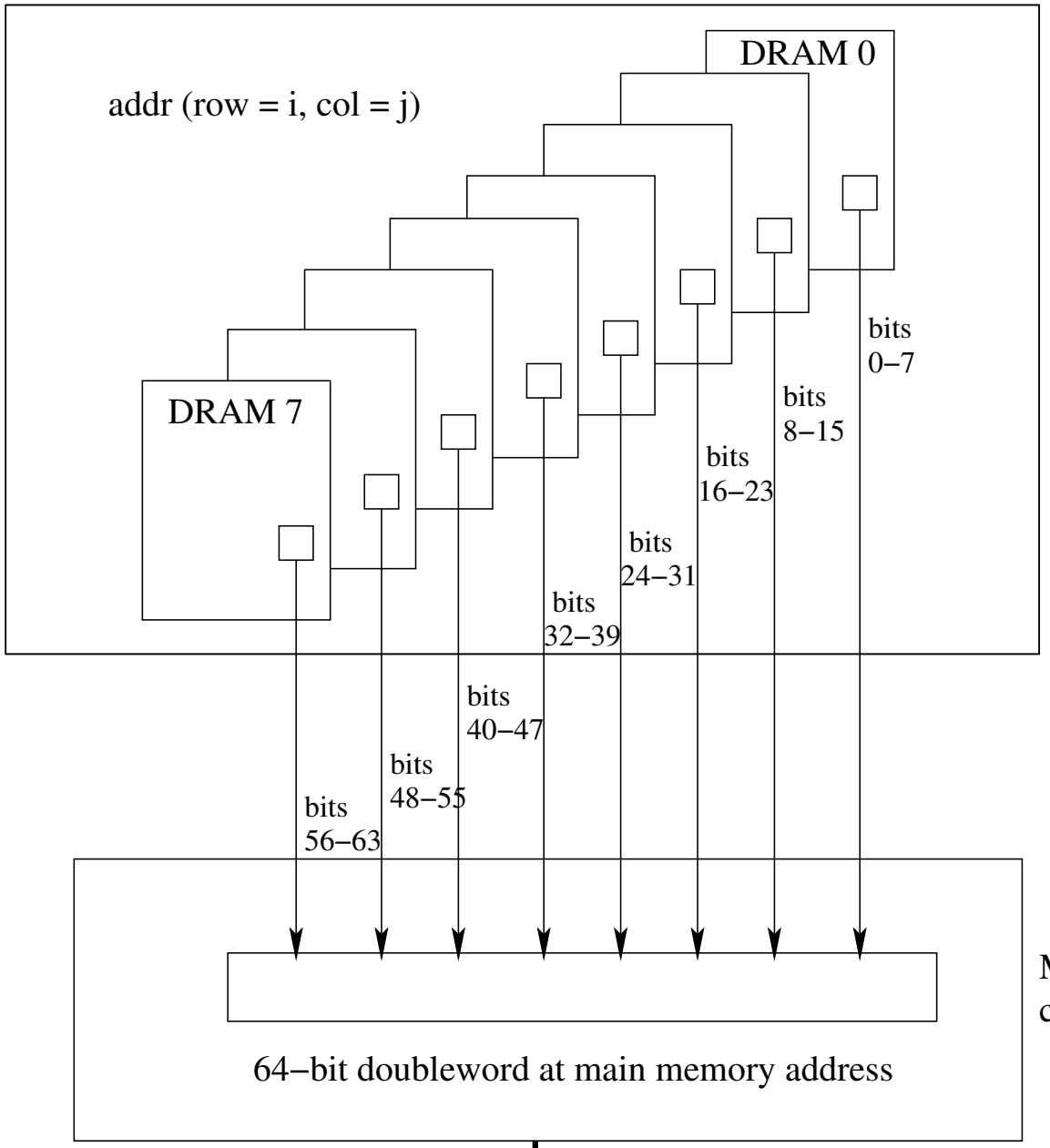


Reading DRAM Supercell (2, 1)

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



Memory Modules



□ : supercell (i, j)

64 MB
memory module
consisting of
eight 8Mx8 DRAMs

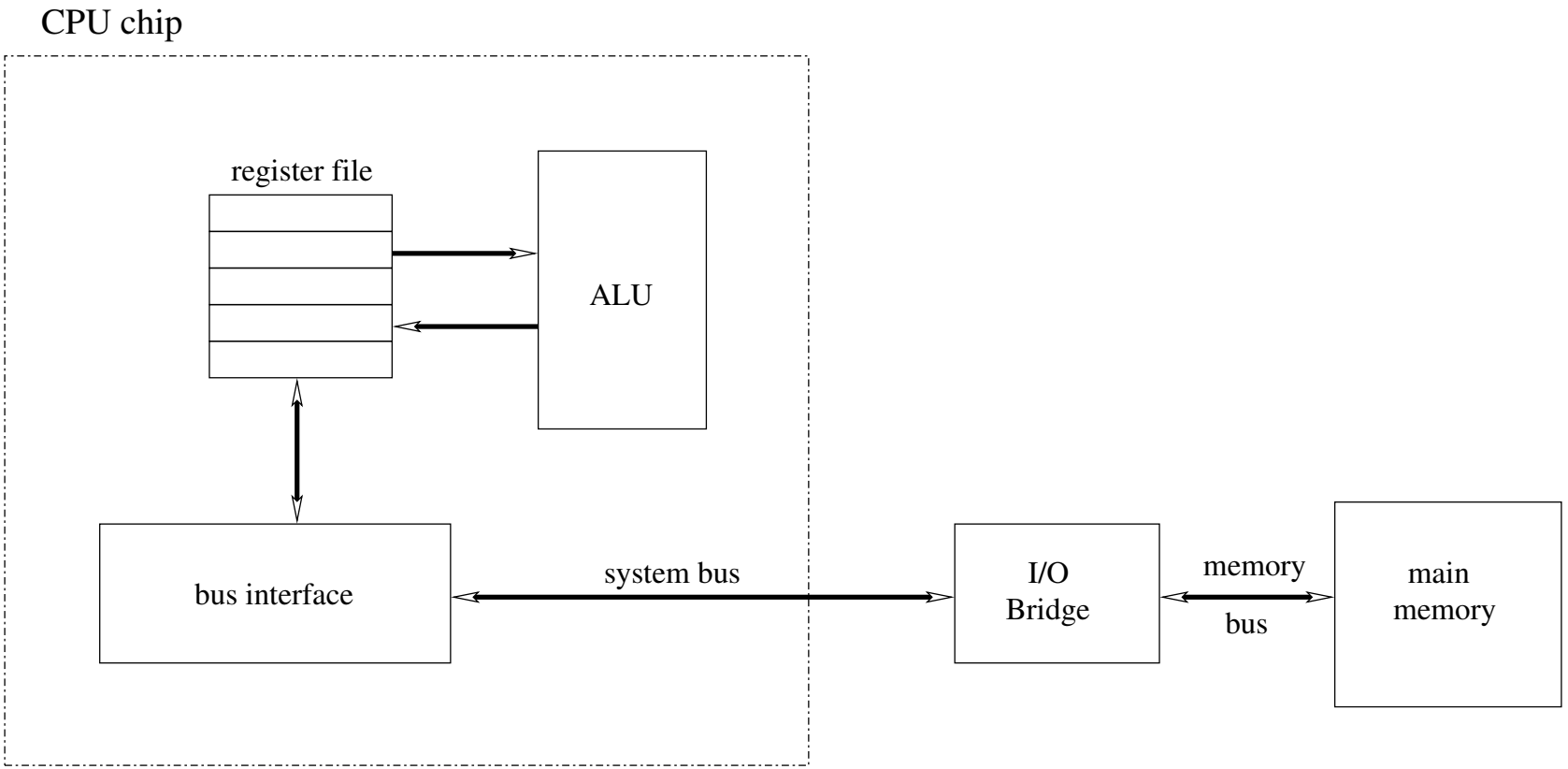
Memory
controller

Nonvolatile Memories

- DRAM and SRAM are volatile memories; they lose information if powered off.
- Nonvolatile memories retain their value even if powered off.
  - The generic name is read-only memory (ROM).
  - This is misleading because some ROMs can be read and modified.
- Types of ROMs
  - Programmable ROM (PROM)
  - Eraseable programmable ROM (EPROM)
  - Electrically eraseable PROM (EEPROM)
  - Flash memory
- Firmware: Program stored in a ROM
  - Boot time code, BIOS (basic input/output system)
  - Graphics cards, disk controllers

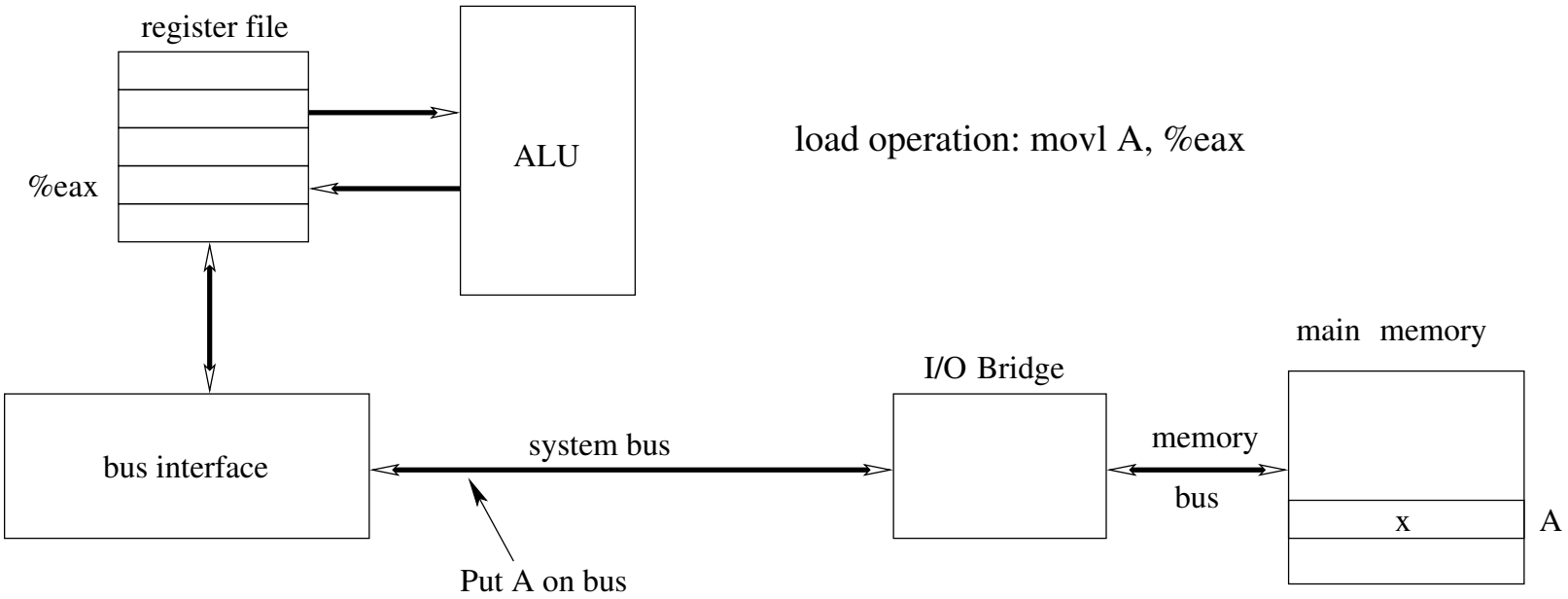
Connecting CPU and Memory

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



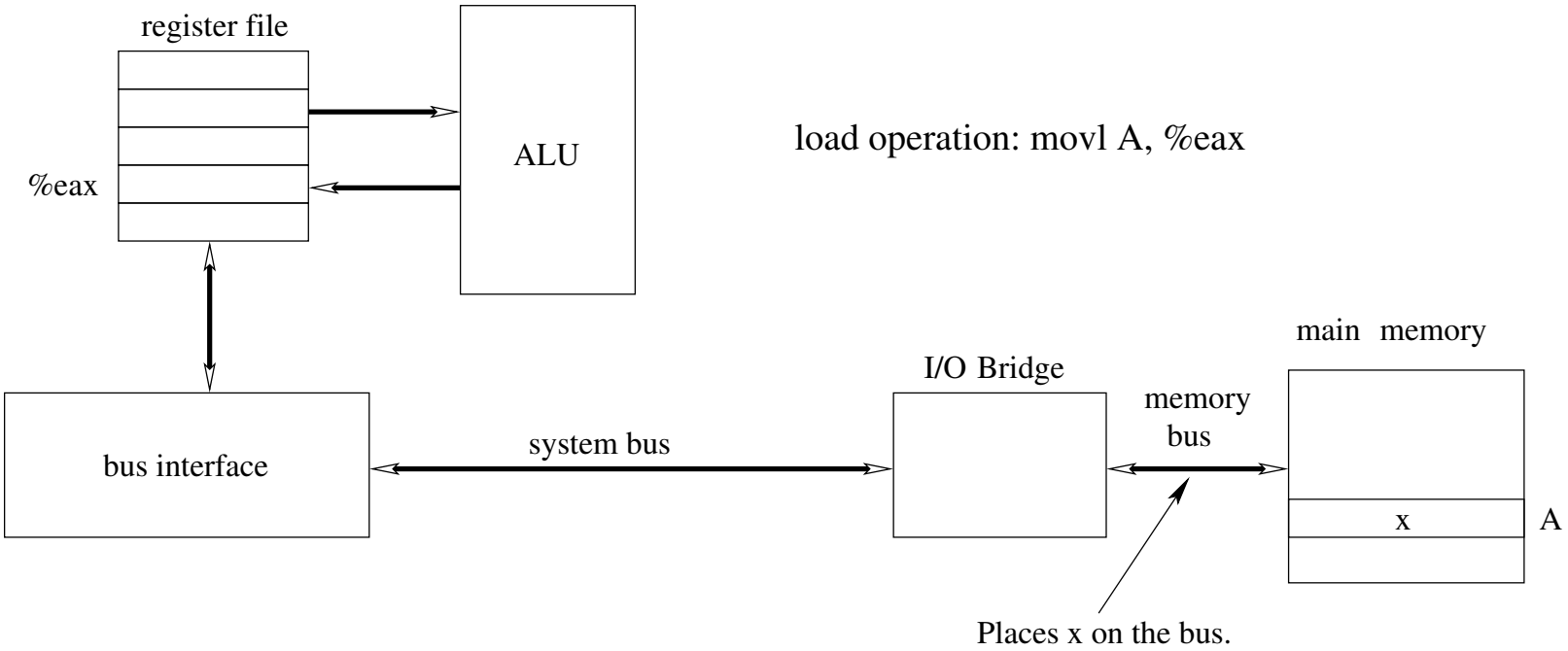
Memory Read Transaction (1)

CPU places address A on the memory bus.



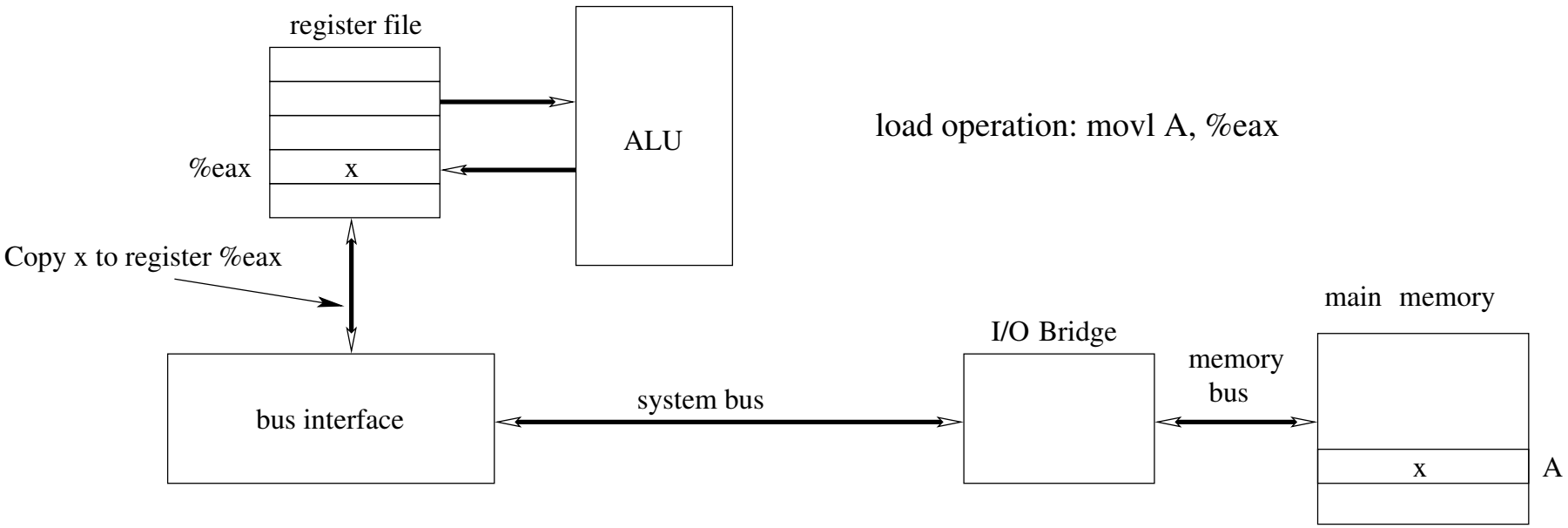
Memory Read Transaction (2)

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



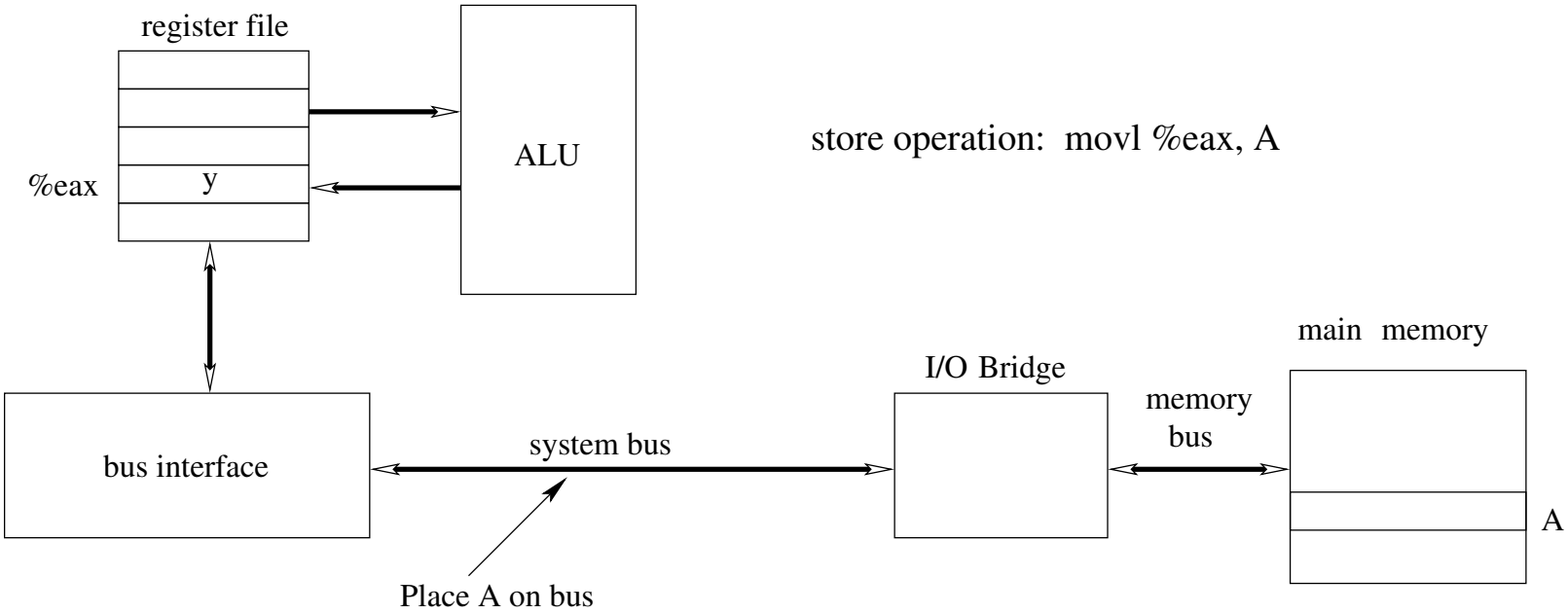
Memory Read Transaction (3)

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



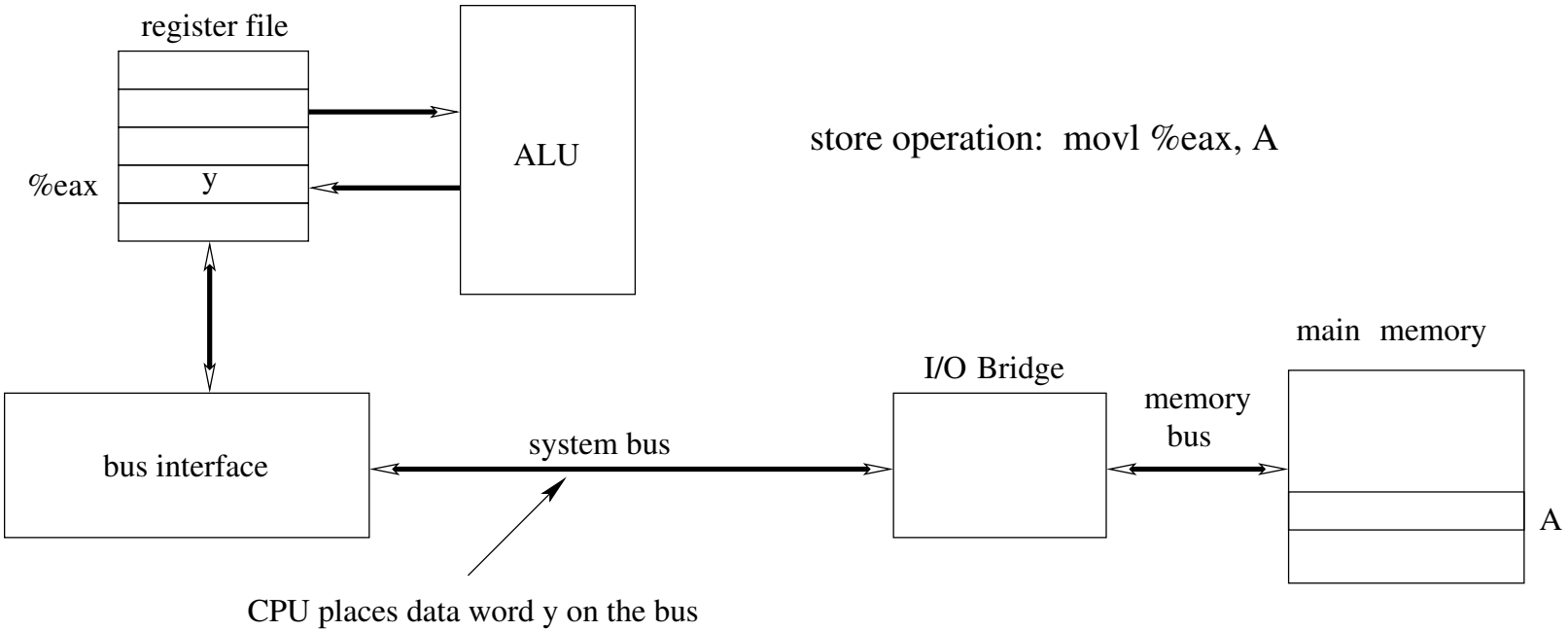
Memory Write Transaction (1)

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



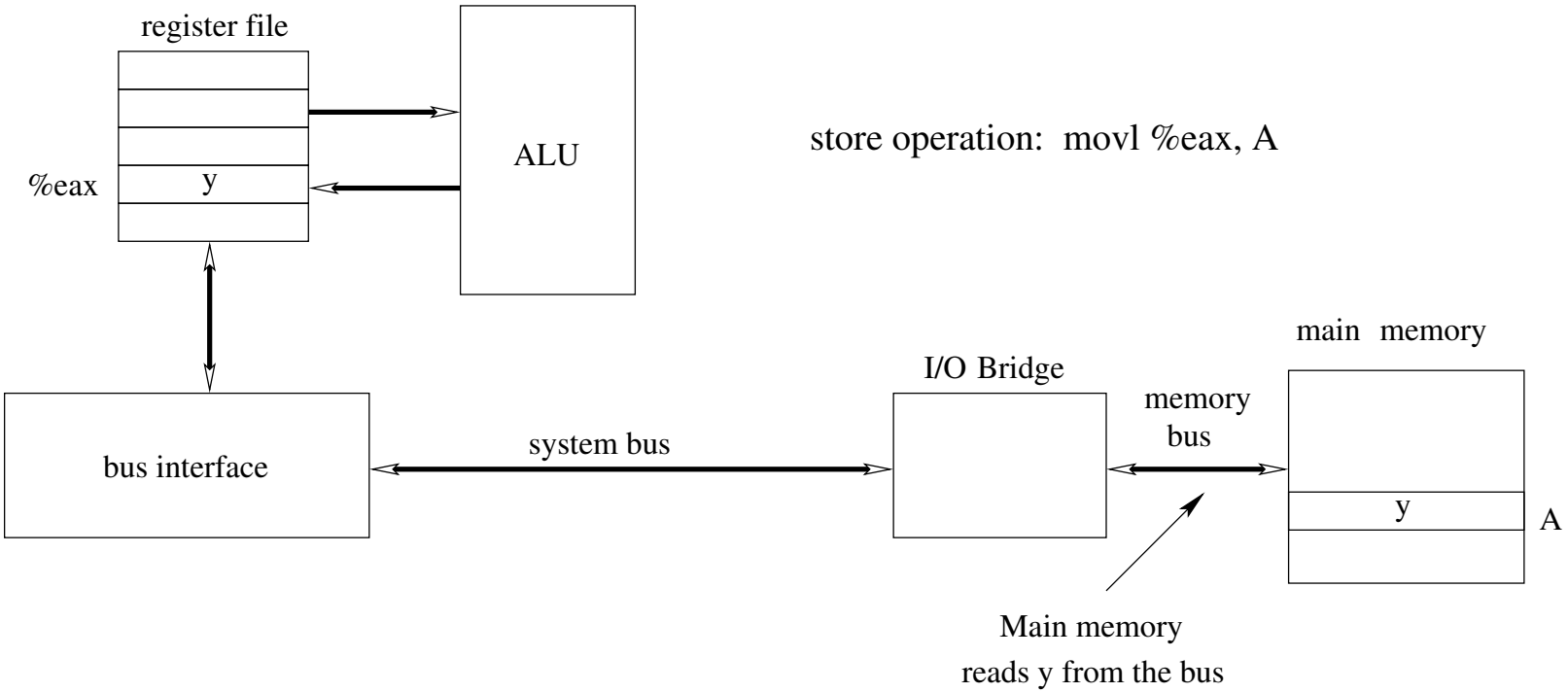
Memory Write Transaction (2)

CPU places data word y on the bus.



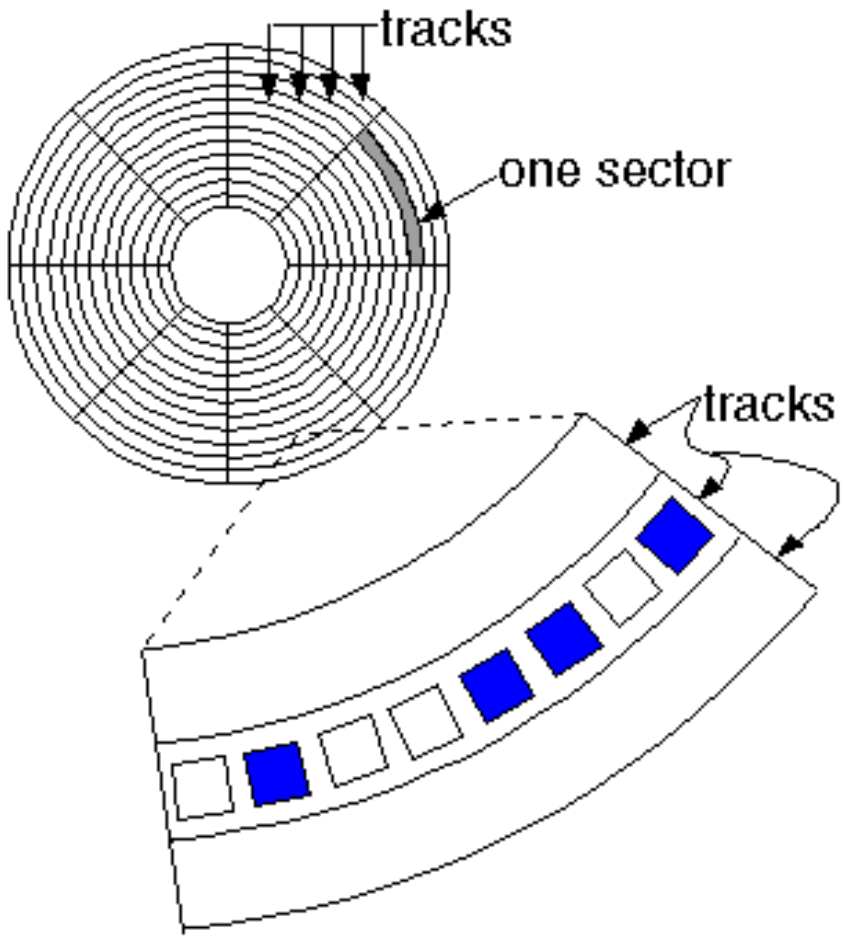
Memory Write Transaction (3)

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



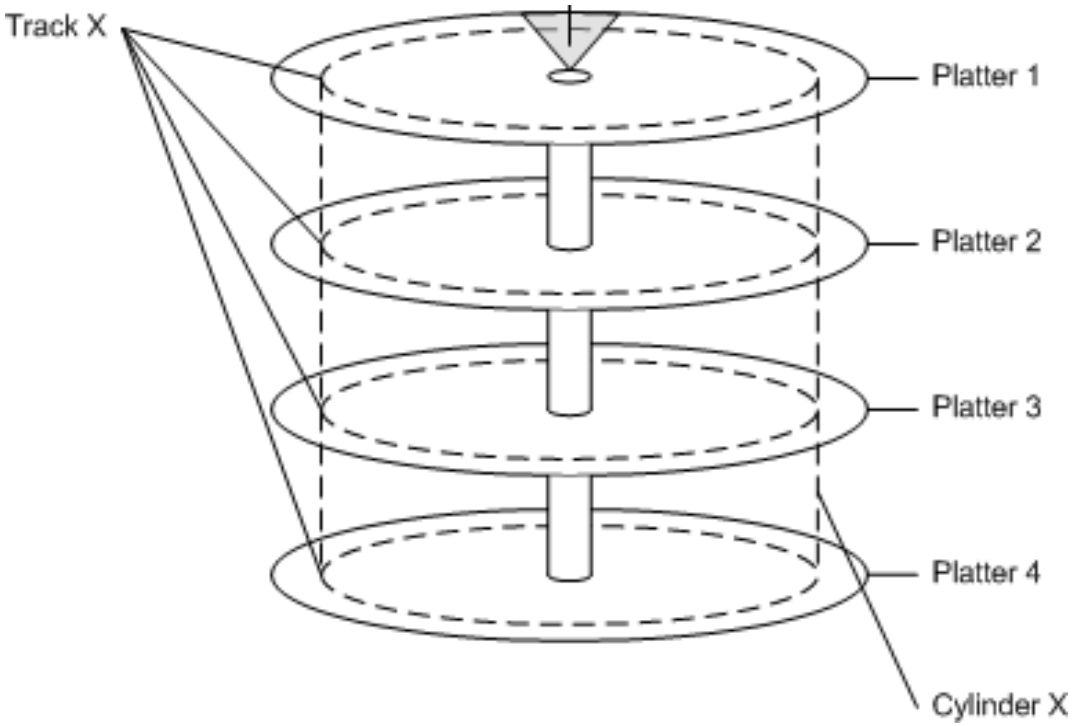
Disk Geometry

- 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 (Multiple-Platter View)

Aligned tracks form a cylinder. Read/write heads move in unison so are all on the same cylinder at any one time.



Disk Capacity

- **Capacity:** maximum number of bits that can be stored. Vendors express this in terms of gigabytes (GB), where 1GB $= 10^9$ 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/in^2): 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 the innermost track.
  - Each zone has a different number of sectors/track.

Computing Disk Capacity

Capacity = (bytes/sector) $\times$ (avg. sectors/track) $\times$
(tracks/surface) $\times$ surfaces/platter) $\times$ platters/disk)

Example:

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

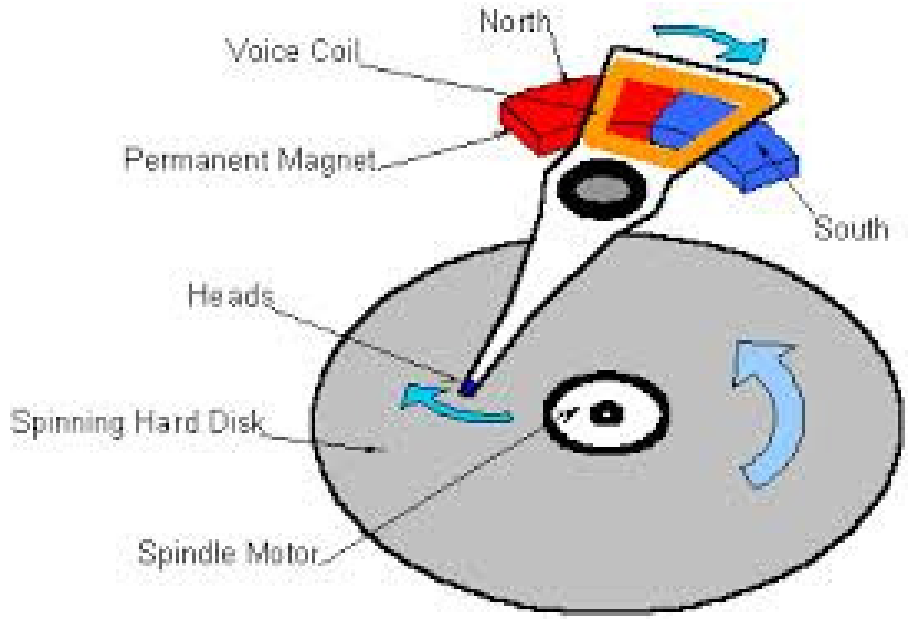
$$\text{Capacity} = 512 \times 300 \times 20000 \times 2 \times 5 = 30,720,000,000 = 30.72\text{GB}$$

Disk Operation (Single-Platter View)

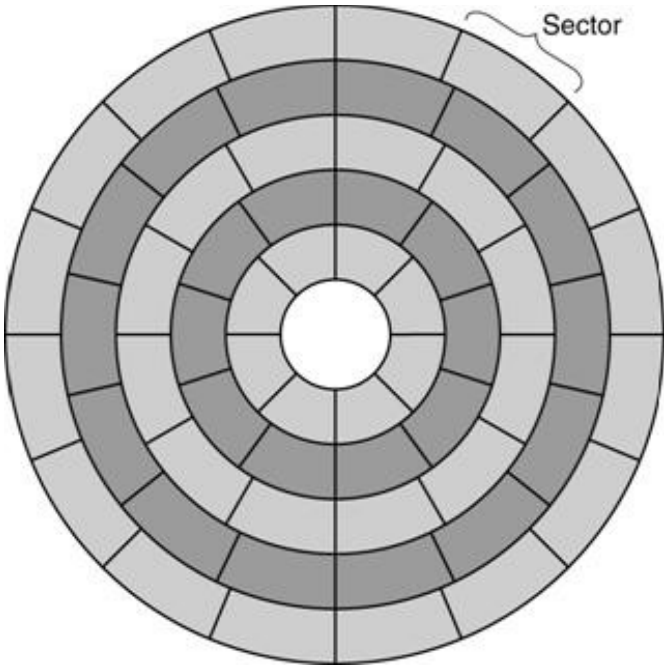
The disk surface spins at a fixed rotational rage.

The read/write head is attached to the end of the arm and flies over the disk surface on a a very thin cushion of air (around 0.1 microns).

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



Reading a Sector



To read a sector on a disk requires:

- **Seek:** the read head is moved to the proper track.
- **Rotational latency:** the desired sector must rotate to the read head.
- **Data transfer:** the sector is read as it rotates under the read head.

Writing is the same.

Disk Access Time

The average time to access a target sector is approximately:

$$T_{access} = T_{seek} + T_{rotation} + T_{transfer}$$

- **Seek time** (T_{seek})
  - Time to position heads over cylinder containing the target sector.
  - Average $T_{seek} = 9\text{ms}$
- **Rotational latency** ($T_{rotation}$)
  - Time waiting for first bit of target sector to pass under read/write head.
  - Average $T_{rotation} = 1/2 \times 1/\text{RPMs} \times 60\text{sec}/1\text{min}$
- **Transfer time** ($T_{transfer}$)
  - Time to read the bits in the target sector.
  - Average $T_{transfer} = 1/\text{RPM} \times 1/((\text{average sectors/track}) \times 60\text{sec}/1\text{min}$

Disk Access Time Example

Given:

- Rotational rate: 7,200 RPM
- Average seek time: 9 ms
- Average sectors/track: 400

Derived:

- Average T_{seek}: $1/2 \times (60\text{sec}/7200\text{RPM} \times 1000\text{ms/sec} = 4\text{ms}$
- Average $T_{transfer}$: $60/7200\text{RPM} \times 1/(400 \text{ sectors/track}) \times 1000\text{ms/sec} = 0.02\text{ms}$
- T_{access}: 9 ms + 4 ms + 0.02 ms

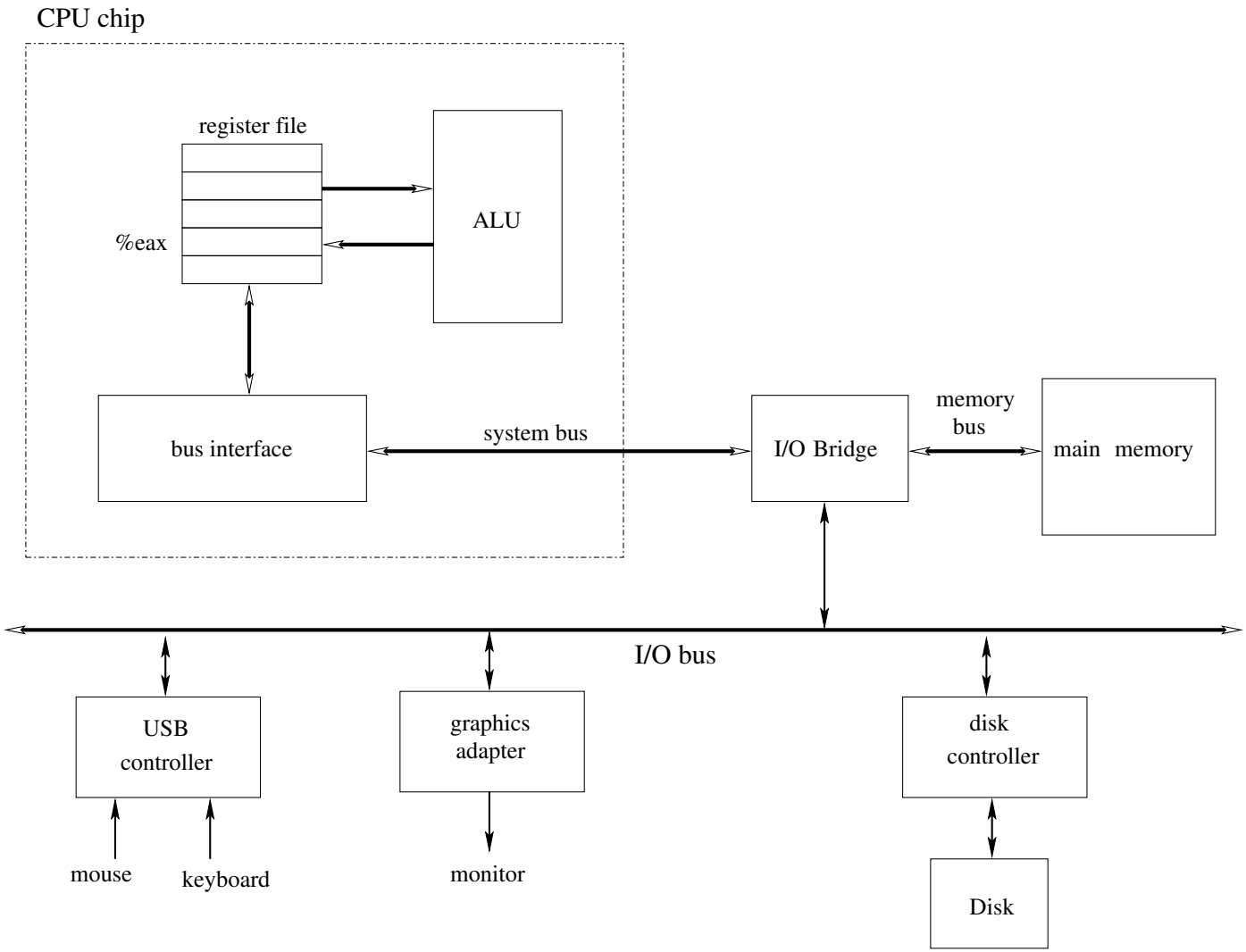
Important points:

- Access time is dominated by seek time and rotational latency.
- The first bit in a sector is the most expensive; the rest are free.
- SRAM access time is about 4ns / doubleword; DRAM about 60ns.
- Disk is about 40,000 times slower than SRAM, and 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** $(0, 1, 2, \ldots)$.
- Mapping between logical blocks and actual (physical) sectors:
  - Is maintained by a hardware/firmware device called a disk controller.
  - Converts requests for logical blocks into (surface, track, sector) triples.
- Allows the controller to set aside spare cylinders for each zone.
  - This accounts for the difference between "formatted capacity" and "maximum capacity."

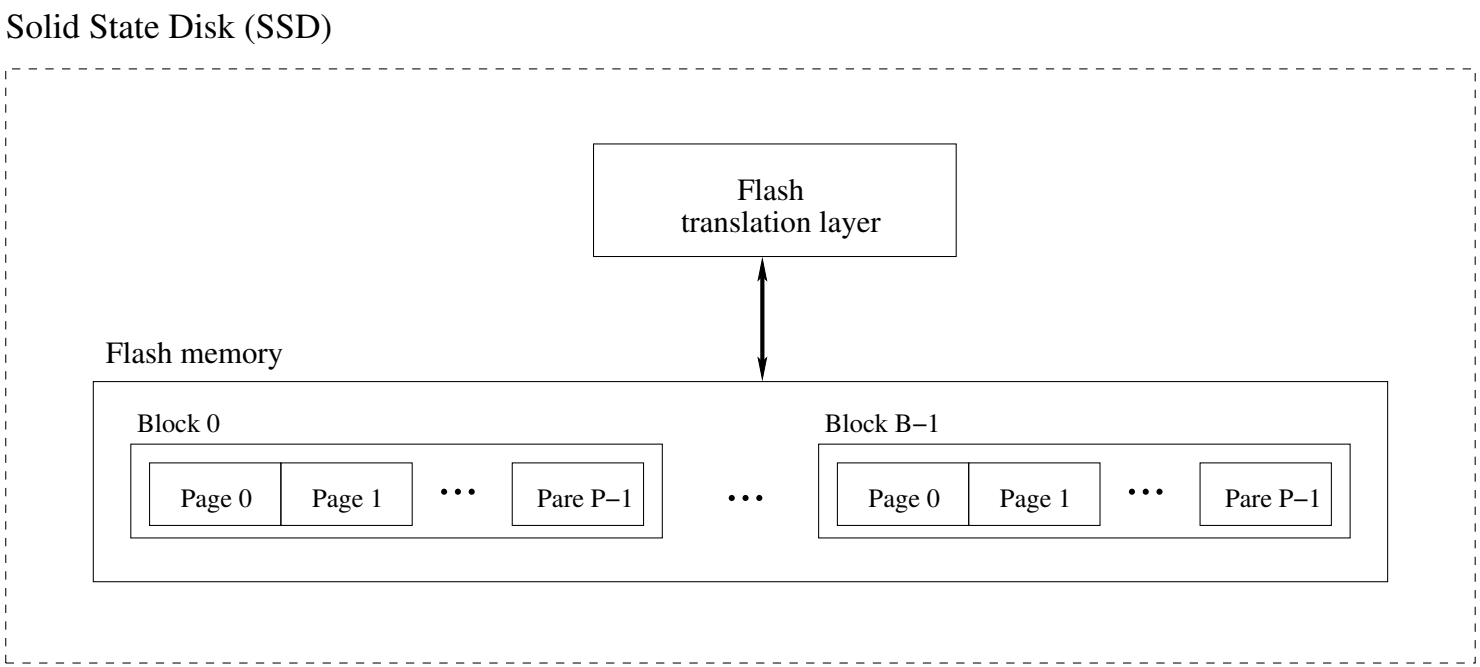
I/O Bus


CPU chip
register file
%eax
ALU
bus interface
system bus
I/O Bridge
memory bus
main memory
I/O bus
USB controller
graphics adapter
disk controller
mouse
keyboard
monitor
Disk

Reading a Disk Sector

1. The CPU initiates a disk read by writing a command, logical block number, and destination memory address to a *port* (address) associated with the disk controller.
2. The disk controller reads the associated sector and performs a direct memory access (DMA) transfer into main memory.
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)



- Requests to read and write logical disk blocks come across the I/O bus to the Flash translation layer.
- Pages are 512KB to 4KB; blocks are 32 to 128 pages.
- Data is read/written in units of pages.
- A page can only be written after its block has been erased.
- A block wears out after around 100,000 repeated writes.

SSDs Performance Characteristics

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

Why are random writes so slow?

- Erasing a block is slow (around 1 ms).
- Write to a page triggers a copy of all useful pages in the block.
- Must find a used block (new block) and erase it.
- Write the page into the new block.
- Copy other pages from the old block to the new block.

SSD vs. Rotating Disks

Advantages:

- No moving parts; faster, less power, more rugged.

Disadvantages:

- Have the potential to wear out. This is mitigated by "wear leveling logic" in the flash translation layer.
- E.g., Intel X25 guarantees 1 petabyte (10^{15} bytes) of random writes before they wear out.
- In 2010, they were about 100X more expensive. But by November, 2013 this has fallen to 10X.

Applications:

- MP3 players, smart phones, laptops.
- They are beginning to appear in desktops and servers.

Storage Trends

Year:	1980	1985	1990	1995	2000	2005	2010	1980:2010
SRAM								
$/MB	19.2K	2.9K	320	256	100	75	60	320
access (ns)	300	150	35	15	3	2	1.5	200
DRAM								
$/MB	8K	880	100	30	1	0.1	0.06	130K
access (ns)	375	200	100	70	60	50	40	9
typical size	0.064	0.256	4	16	64	2K	8K	125K
Disk								
$/MB	500	100	8	0.30	0.001	0.005	0.0003	1.6M
access (ns)	87	75	28	10	8	4	3	29
typical size	1	10	160	1K	20K	160K	1.5M	1.5M

CPU Clock Rates

Year:	1980	1985	1990	1995	2000	2005	2010	1980:2010
CPU	8080	386	Pentium	P-III	P-4	Core 2	Core i7	
Clock MHz	1	20	150	600	3300	2000	2500	2500
Cycle (ns)	1000	50	6	1.6	0.3	0.5	0.4	2500
Cores	1		1	1	1	1 2	4	4
Effective Cycle time	1000	50	6	1.6	0.3	0.25	0.1	10K

Around 2003, was the inflection point in computer history when designers hit the "Power Wall." Cores increased, but the clock rate actually decreased.

CPU-Memory Gap

CPU speed increases *faster* than memory speed, meaning that:

- memory is more and more a limiting factor on performance;
- increased importance for caching and similar techniques.