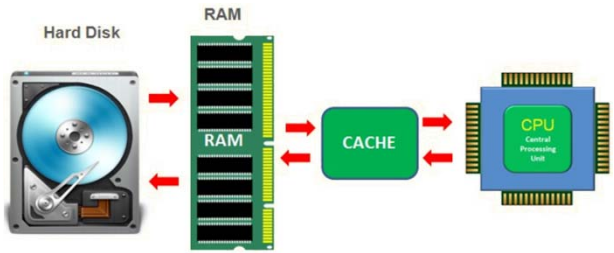
CSE320 System Fundamentals II
The Memory Hierarchy

YoungMin Kwon

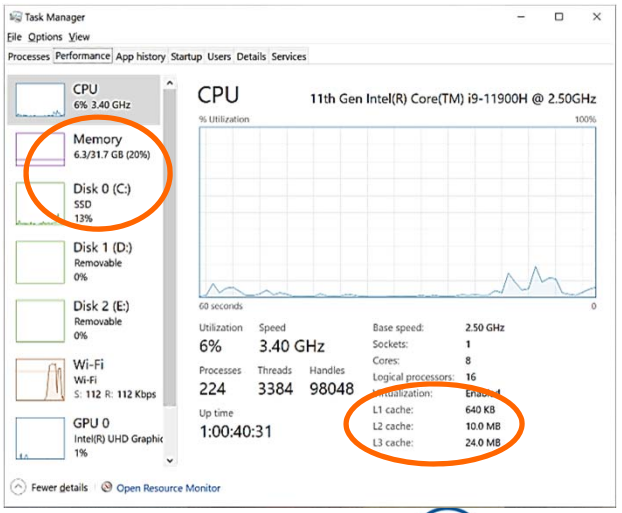
SUNY Korea

Memory System

- A memory system
  - A hierarchy of storage devices with different capacities, costs, and access times



  - E.g.
    - Registers (0 cycles)
    - Cache memories (4 ~ 75 cycles)
    - Main memory (hundreds of cycles)
    - Disks (millions of cycles)



SUNY Korea

Memory System

- Locality
  - Well-written programs tend to access the same set of data items over and over again

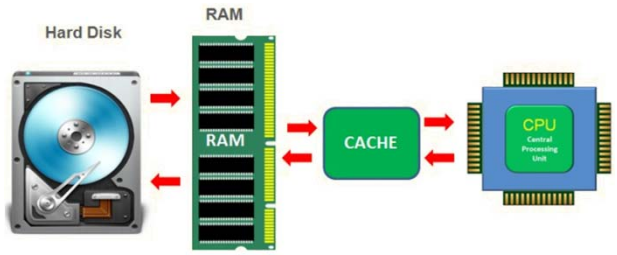
```
int sum(int *items, int nr_item) {
    int i, s = 0;

    for(i = 0; i < nr_item; i++)
        s += items[i];

    return s;
}
```

SUNY Korea

Memory System

- Locality
  - Memory hierarchy works because such programs tend to access a particular level more frequently than the next lower level

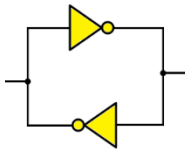
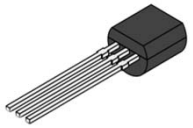
```
int sum(int *items, int nr_item) {
    int i, s = 0;

    for(i = 0; i < nr_item; i++)
        s += items[i];

    return s;
}
```



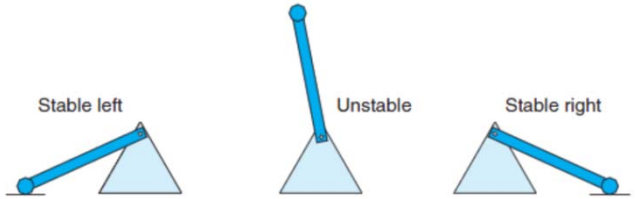
SUNY Korea

SRAM: Static Random Access Memory

- Each memory cell is a flip-flop implemented with a 6-transistor circuit

  - Flip-flops make bistable memory cells



  - Memory cells retain their values as long as they are powered.
  - Robust against disturbances

SUNY Korea

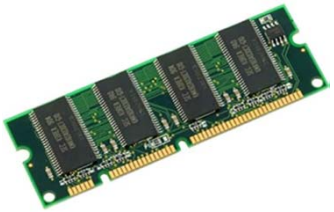
DRAM: Dynamic RAM

- Each bit is stored as charge on a capacitor

- Sensitive to any disturbances
  - Exposure to light will cause the capacitor voltage change
  - Image sensors in digital cameras are essentially arrays of DRAM cells

SUNY Korea

DRAM: Dynamic RAM



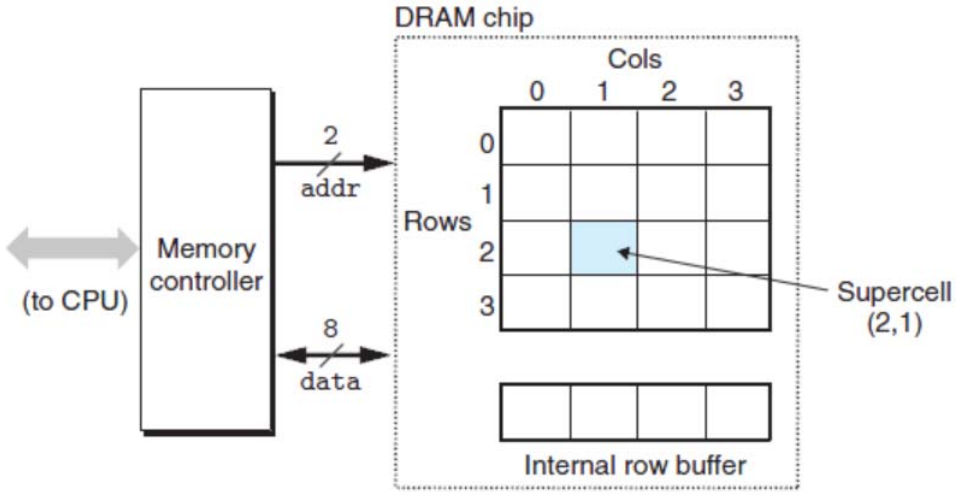
- Leakage current causes a DRAM cell to lose its charge within 10 to 100 milliseconds
  - Memory system must periodically refresh every bit of memory (read and rewrite it)

	Transistors per bit	Relative access time	Persistent?	Sensitive?	Relative cost	Applications
SRAM	6	1×	Yes	No	100×	Cache memory
DRAM	1	10×	No	Yes	1×	Main mem, frame buffers

SUNY Korea

DRAM

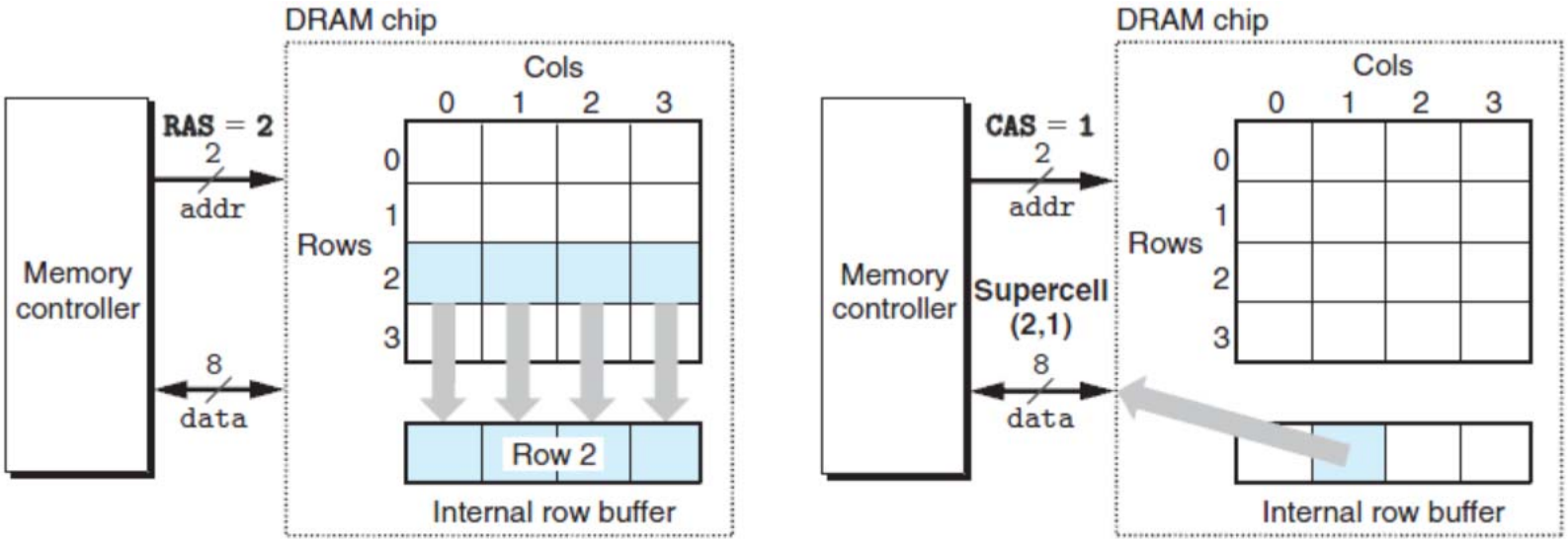
High-level view of a 128-bit 16 × 8 DRAM chip.



- Supercells
  - The cells in a *d* x *w* DRAM chip are partitioned into *d* supercells, each consisting of *w* DRAM cells (*d* · *w* bits of information)
  - Supercells are organized as a rectangular array with *r* rows and *c* columns, where *r* · *c* = *d*

SUNY Korea

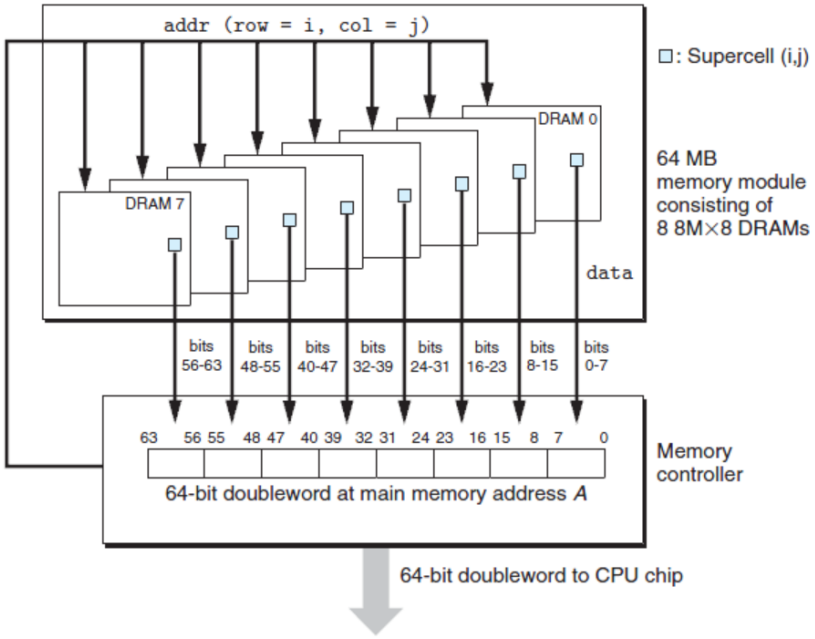
DRAM



- To read the contents of supercell (i, j)
  - The memory controller sends the row address i (RAS: row access strobe), followed by the column address j (CAS: column access strobe).

SUNY Korea

DRAM (Memory Modules)

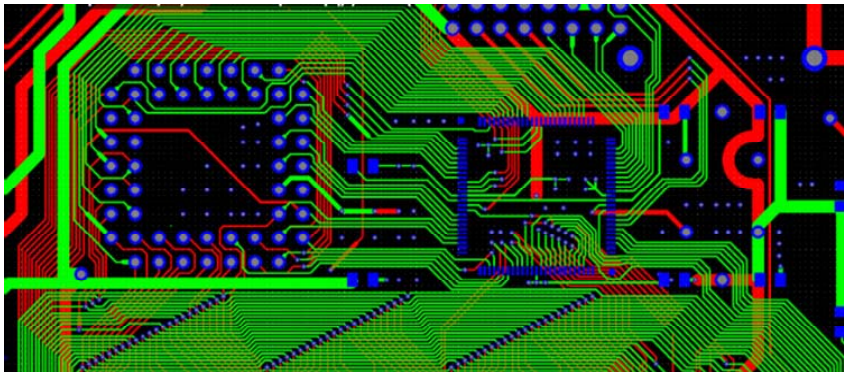


- DRAM chips are packaged in memory modules

- 64 MB using eight 8M x 8 DRAM chips (64-Mbit)
  - Each supercell stores 1 byte
  - Each 64bit word is represented by 8 supercells whose address is (i, j)

Accessing Main Memory: BUS

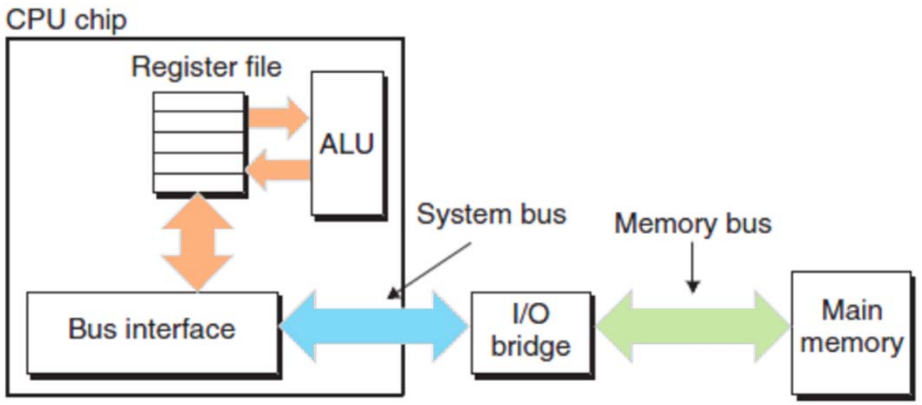
- BUS
  - A shorthand for the Latin omnibus, also called data highway
  - A collection of parallel wires that carry address, data, and control
  - Data flows back and forth between the processor and the main memory over shared electrical conduits called buses



SUNY Korea

Accessing Main Memory: BUS

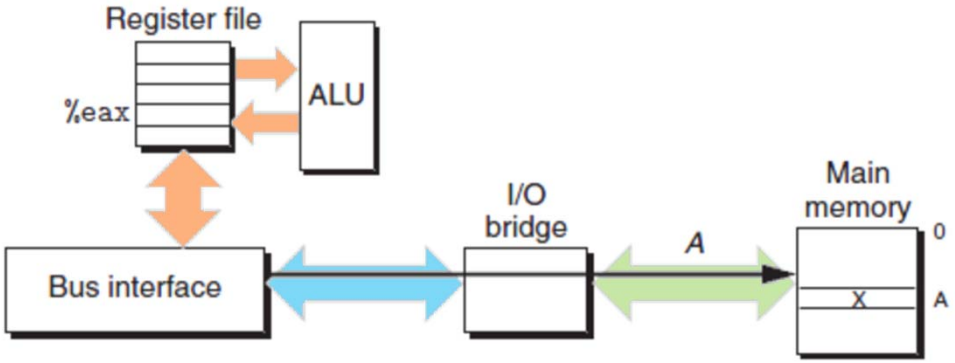
- BUS
  - Address and data can share the same set of wires
  - Control wires
    - Main memory or I/O devices
    - Address or data
    - Read or write



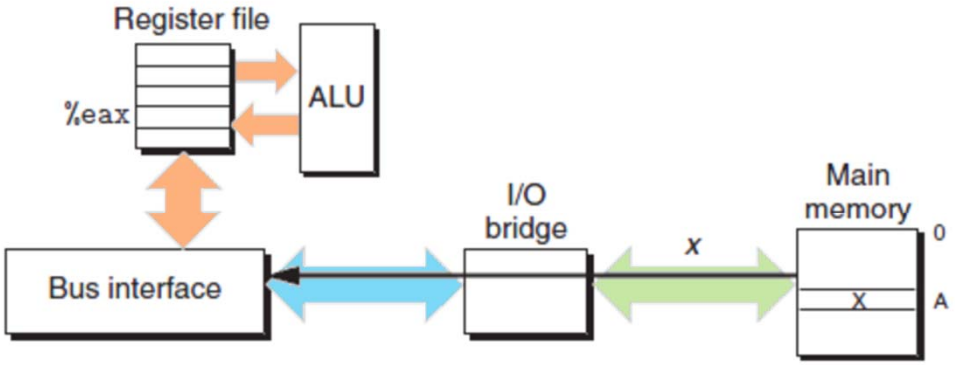
SUNY Korea

Memory read

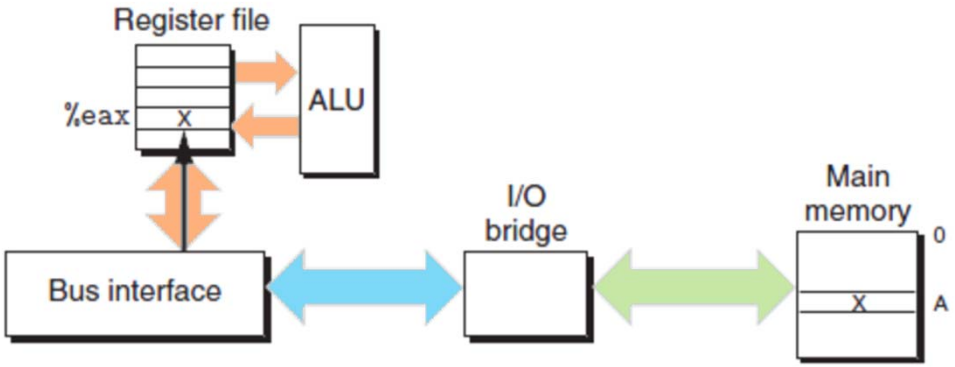
movl A, %eax



(a) CPU places address *A* on the memory bus.



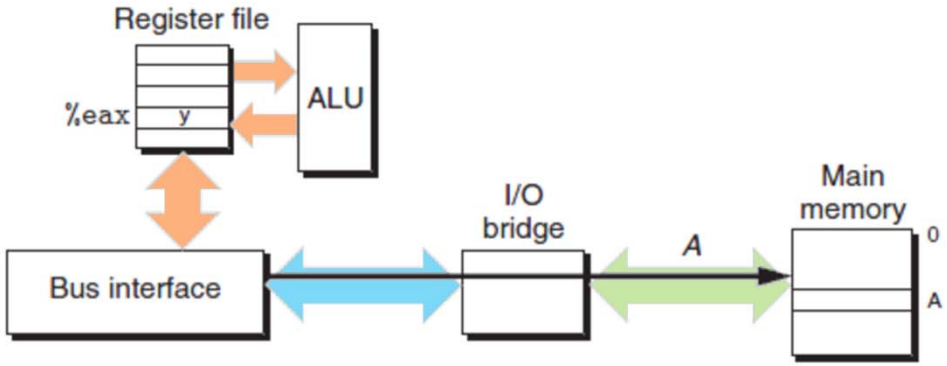
(b) Main memory reads *A* from the bus, retrieves word *x*, and places it on the bus.



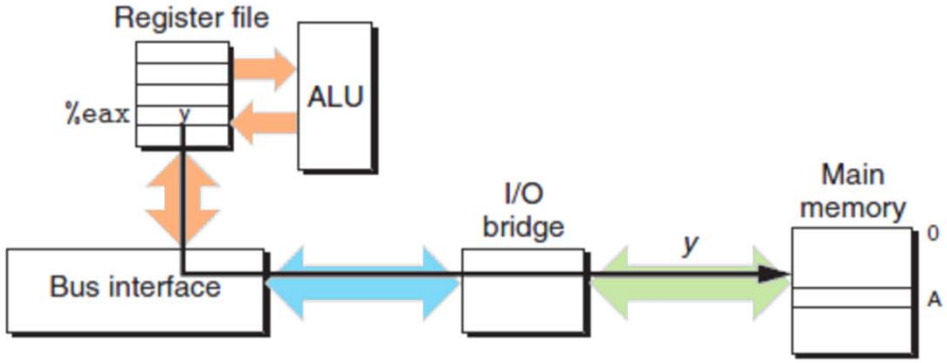
(c) CPU reads word *x* from the bus, and copies it into register %eax.

SUNY Korea

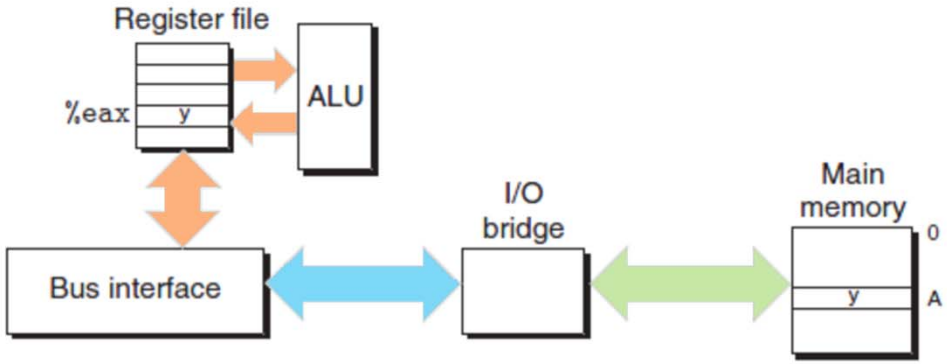
Memory write
movl %eax, A



(a) CPU places address *A* on the memory bus. Main memory reads it and waits for the data word.



(b) CPU places data word *y* on the bus.



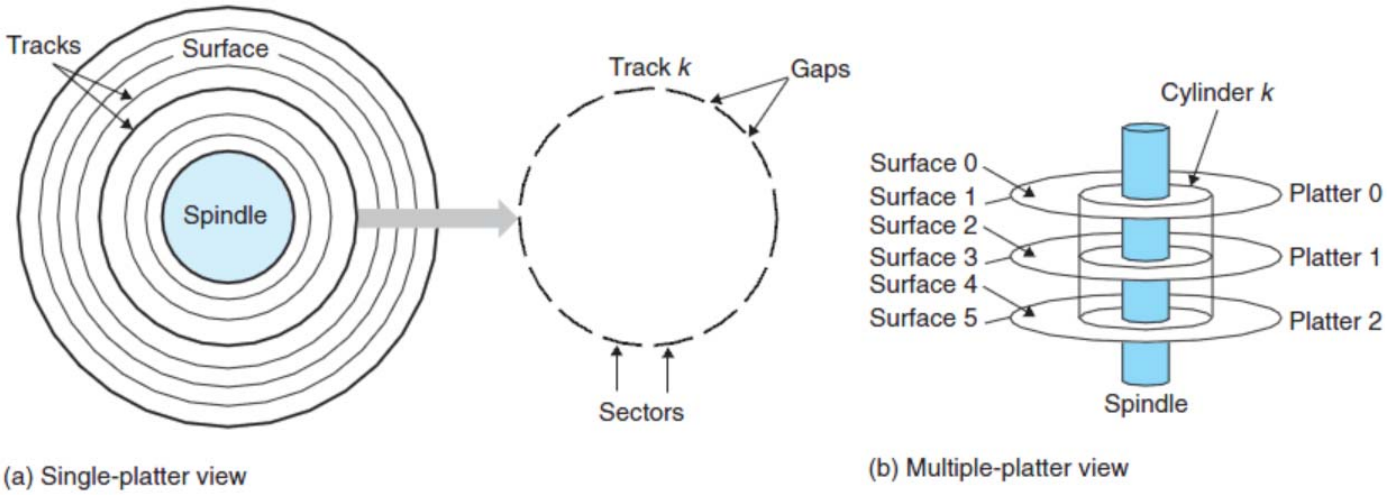
(c) Main memory reads data word *y* from the bus and stores it at address *A*.

SUNY Korea

Disk Storage

SUNY Korea

Disk Storage



- Disks have platters
- Each plater consists of two surfaces
- Each surface consists of a collection of rings called tracks
- Each track is partitioned into sectors
- Each sector contains equal number of data bits (512 bytes)
- Sectors are separated by gaps

SUNY Korea

Disk Capacity

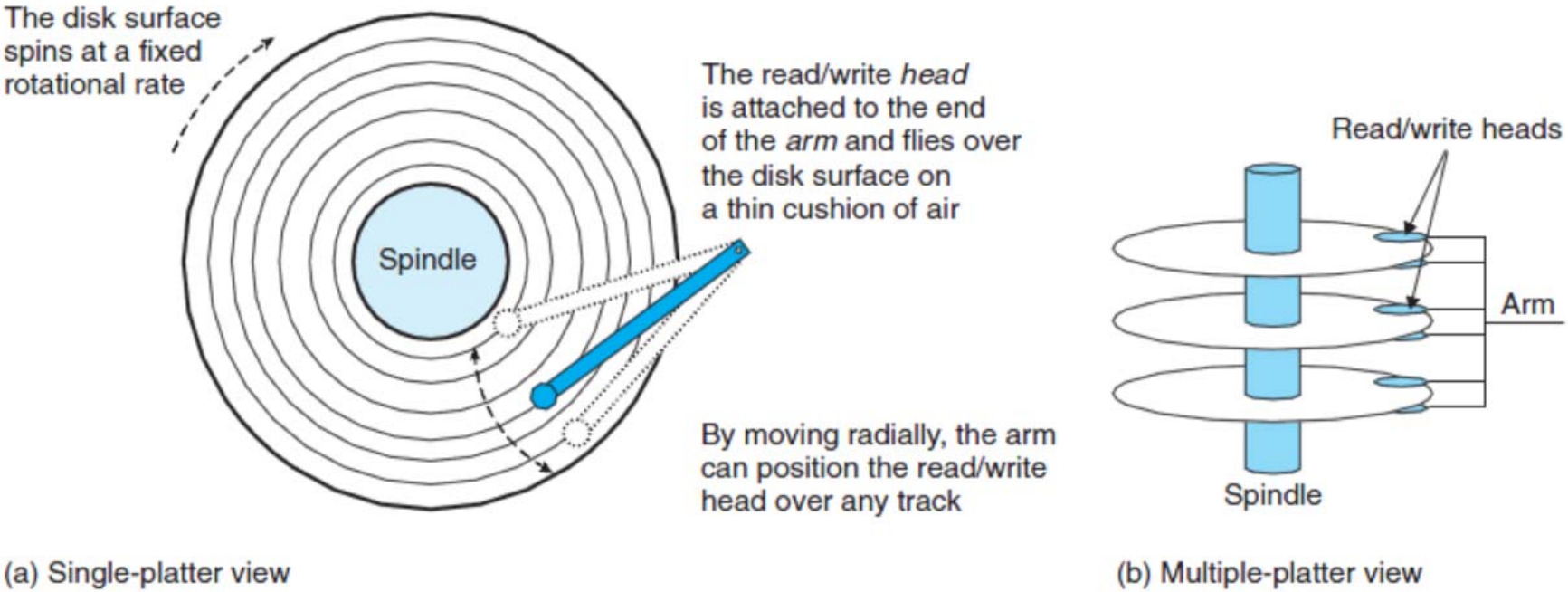
$$\text{Disk capacity} = \frac{\text{\# bytes}}{\text{sector}} \times \frac{\text{average \# sectors}}{\text{track}} \times \frac{\text{\# tracks}}{\text{surface}} \times \frac{\text{\# surfaces}}{\text{platter}} \times \frac{\text{\# platters}}{\text{disk}}$$

- A disk with 5 platters, 512 bytes per sector, 20,000 tracks per surface, an average of 300 sectors per track

$$\begin{aligned}\text{Disk capacity} &= \frac{512 \text{ bytes}}{\text{sector}} \times \frac{300 \text{ sectors}}{\text{track}} \times \frac{20{,}000 \text{ tracks}}{\text{surface}} \times \frac{2 \text{ surfaces}}{\text{platter}} \times \frac{5 \text{ platters}}{\text{disk}} \\ &= 30{,}720{,}000{,}000 \text{ bytes} \\ &= 30.72 \text{ GB}.\end{aligned}$$

SUNY Korea

Disk Operation



(a) Single-platter view

(b) Multiple-platter view

SUNY Korea

Disk Operation (Access Time)

- Seek time
  - The time required to move the arm to the track
  - Average: 3 ~ 9 msec, max: ~20 msec
- Rotational latency
  - The time for the sector to move under the head
  - **1/2** x 1/RPM x 60sec/1min
- Transfer time
  - The time a sector to be read
  - 1/RPM x 1/(avg # of sectors /track) x 60sec/1min

SUNY Korea

Disk Operation (Access Time)

Parameter	Value
Rotational rate	7200 RPM
$T_{avg\ seek}$	9 ms
Average # sectors/track	400

$$T_{avg\ rotation} = 1/2 \times T_{max\ rotation}$$
$$= 1/2 \times (60\ \text{secs} / 7200\ \text{RPM}) \times 1000\ \text{ms/sec}$$
$$\approx 4\ \text{ms}$$

$$T_{avg\ transfer} = 60 / 7200\ \text{RPM} \times 1 / 400\ \text{sectors/track} \times 1000\ \text{ms/sec}$$
$$\approx 0.02\ \text{ms}$$

$$T_{access} = T_{avg\ seek} + T_{avg\ rotation} + T_{avg\ transfer}$$
$$= 9\ \text{ms} + 4\ \text{ms} + 0.02\ \text{ms}$$
$$= 13.02\ \text{ms}$$

SUNY Korea

Access Time

- To read 512 bytes
  - SRAM: 256 ns
  - DRAM: 4000 ns
  - Disk: 10 msec (40,000 times larger than SRAM, 2,500 times larger than DRAM)

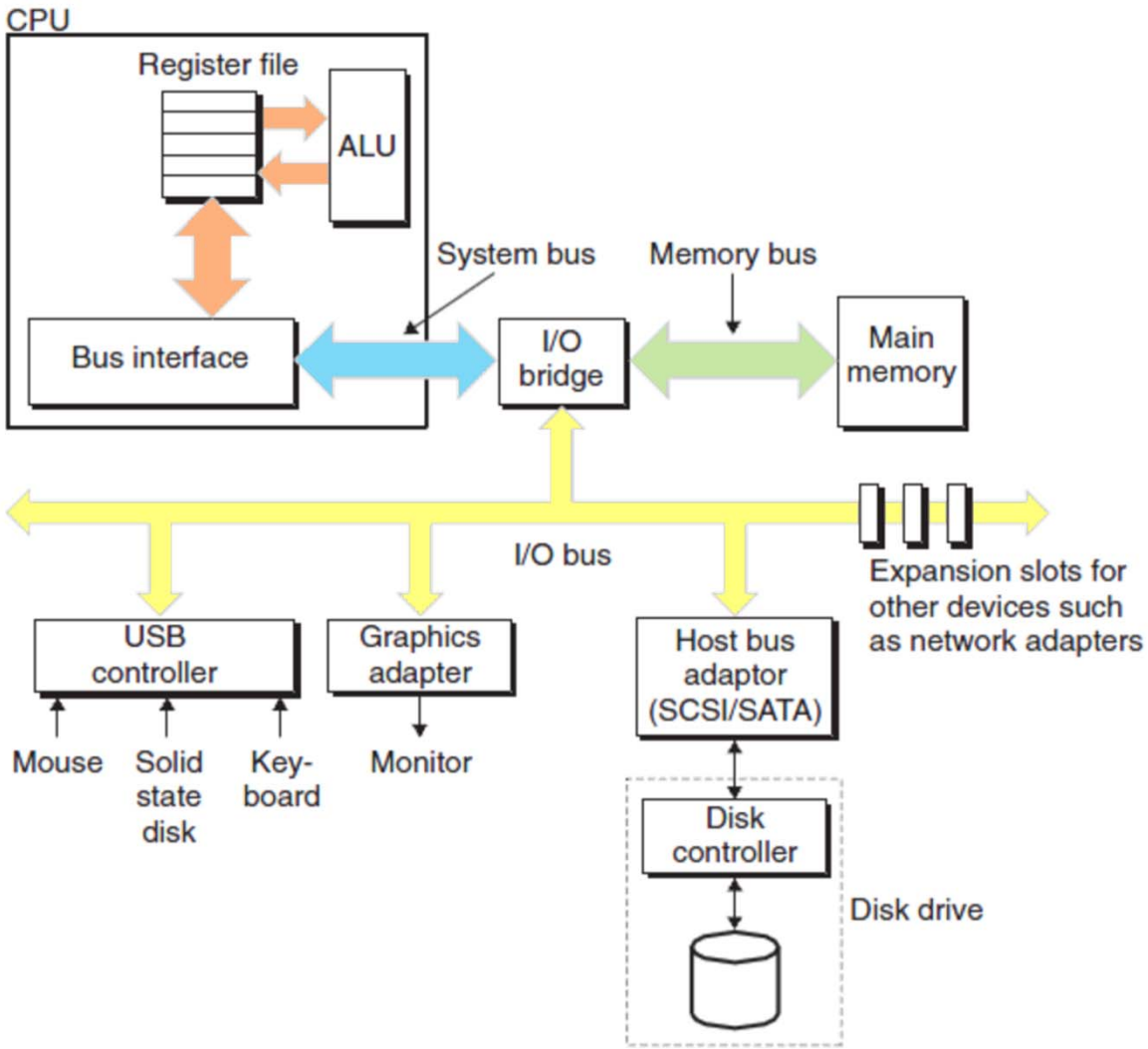
SUNY Korea

Logical Disk Blocks

- To hide the complexity (specifying surface#, track#, sector#) from the OS, modern disks provide a simpler view
  - B sector-size logical blocks are numbered: 0, 1, … , B-1
  - Disk controller maintains the mapping between logical block numbers and actual disk sectors

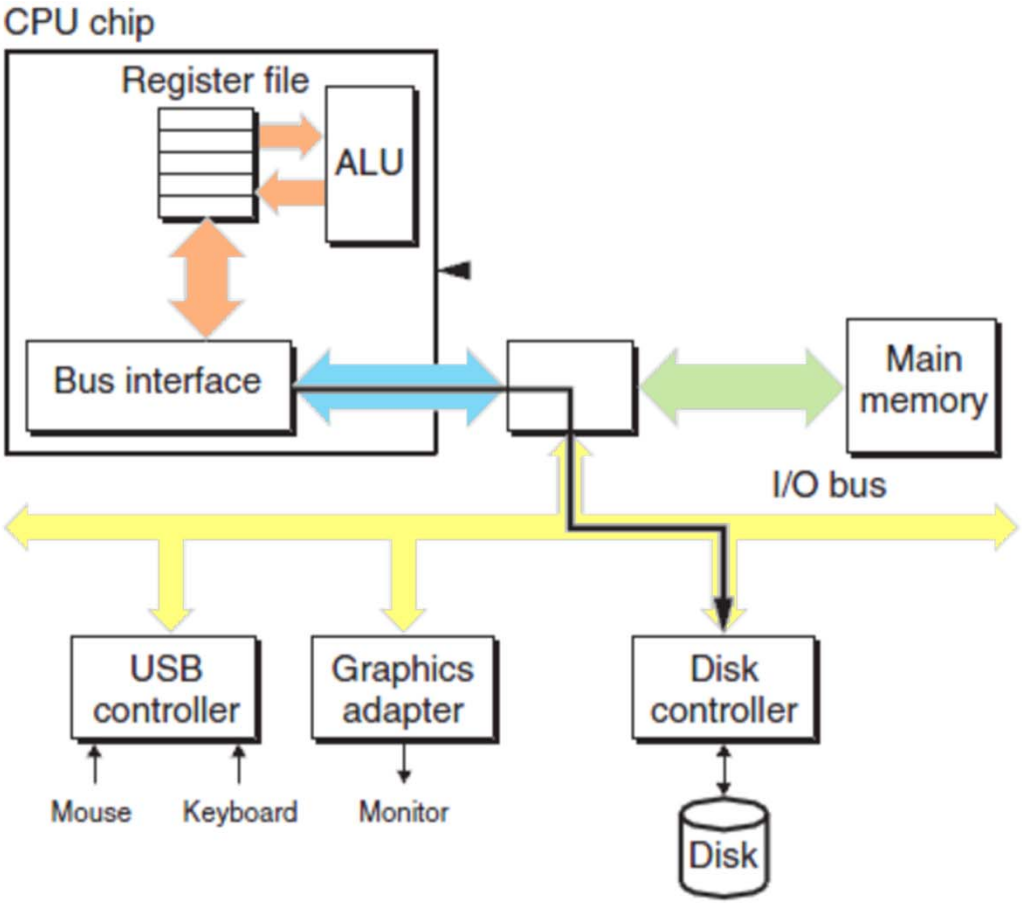
SUNY Korea

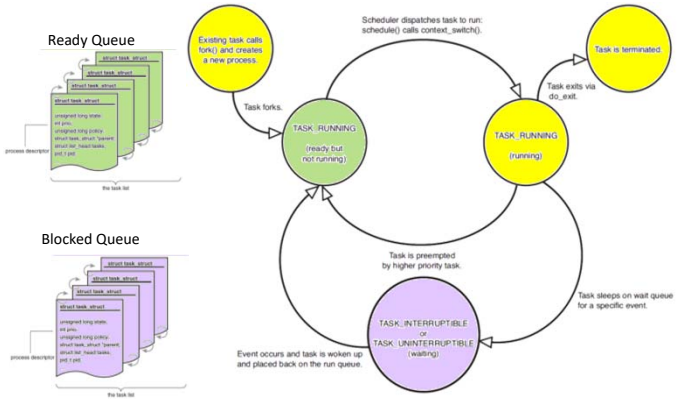
Connecting I/O Devices



SUNY Korea

Reading a Disk Sector (1)

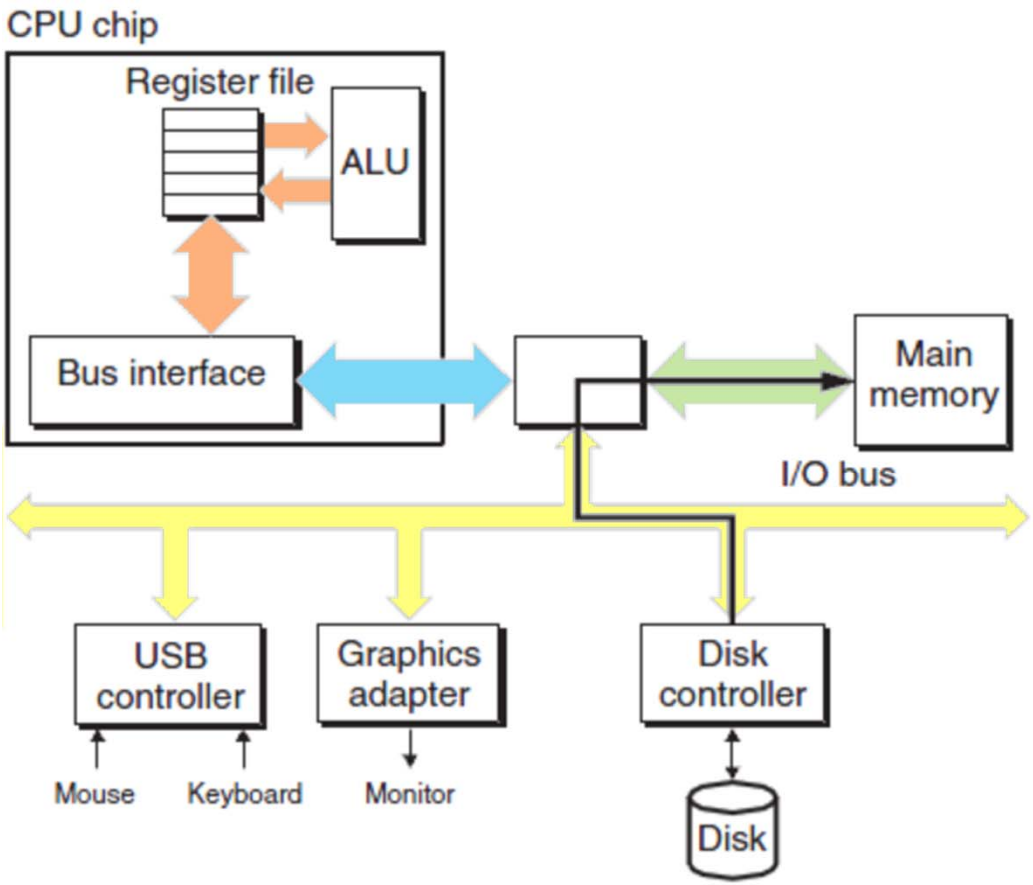


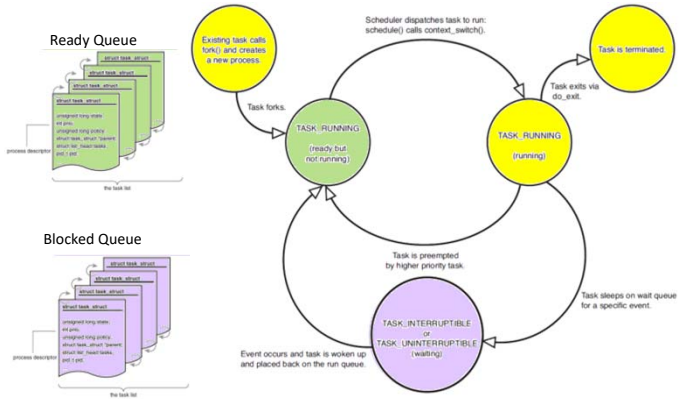


(a) The CPU initiates a disk read by writing a command, logical block number, and destination memory address to the memory-mapped address associated with the disk.

SUNY Korea

Reading a Disk Sector (2)

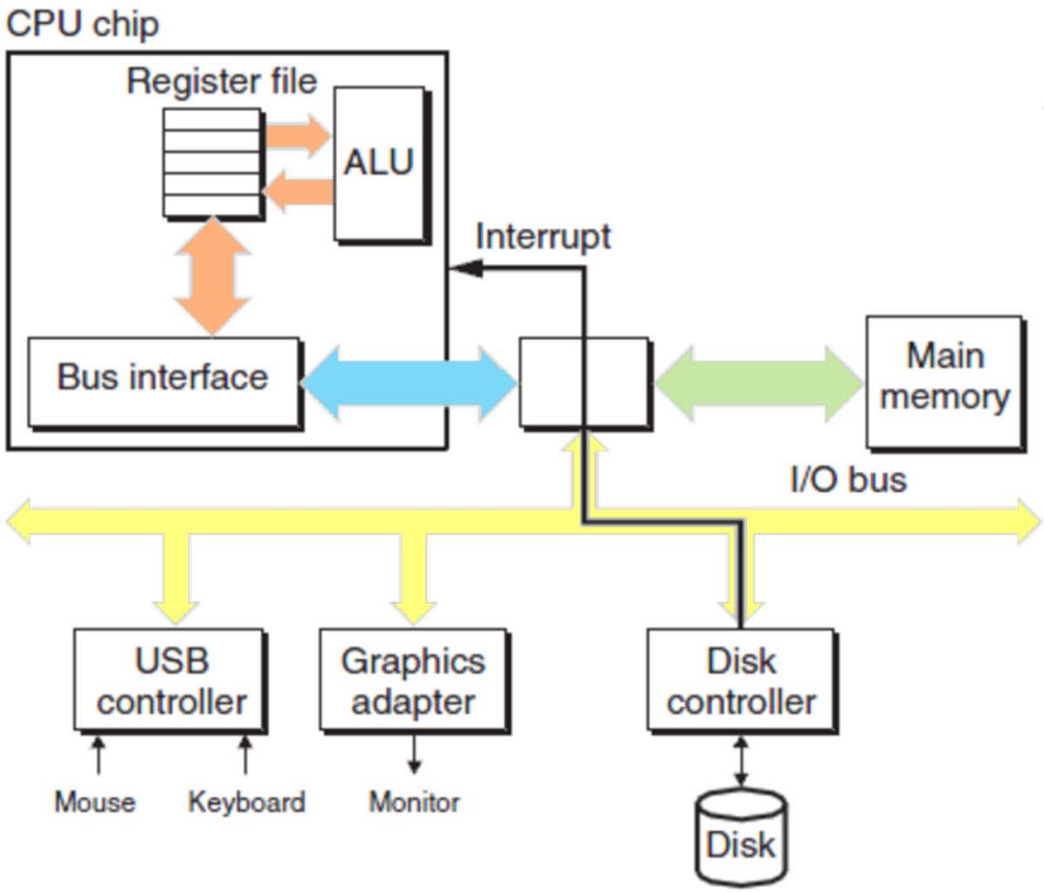


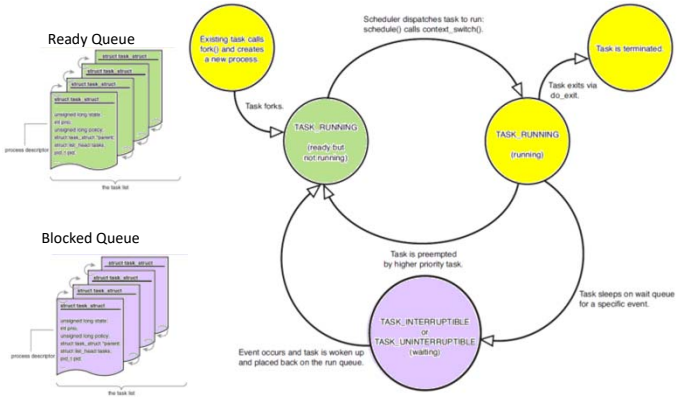


(b) The disk controller reads the sector and performs a DMA transfer into main memory.

SUNY Korea

Reading a Disk Sector (3)





(c) When the DMA transfer is complete, the disk controller notifies the CPU with an interrupt.

SUNY Korea

Storage Technology Trend (2015 vs 1985)

- SRAM
  - $/MB: 116 times cheaper, Access(ns): 115 times faster
- DRAM
  - $/MB: 44,000, Access (ns): 10, Typical size (MB): 62,500
- Rotating Disk
  - $/GB: 3,333,333, Seek time (ms): 25, Typical Size (GB): 300,000
- CPU
  - Effective Cycle time (ns) 2,075

SUNY Korea

Locality

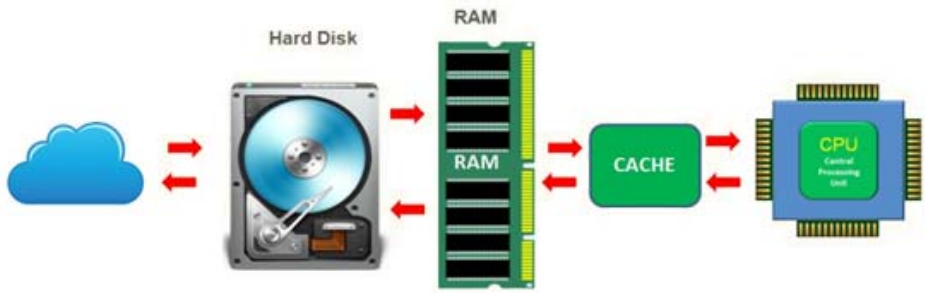
- Temporal locality
  - A memory location referenced once is likely to be referenced again in the near future
- Spatial locality
  - If a memory location is referenced, its nearby locations are likely to be referenced in the near future

```
int sum(int *items, int nr_item) {
    int i, s = 0;

    for(i = 0; i < nr_item; i++)
        s += items[i];

    return s;
}
```

SUNY Korea

Locality

- Cache
  - Main memory as a cache for the virtual memory
  - L1, L2, L3 caches as a cache for main memory
  - Local files as a cache for network contents



SUNY Korea

Locality

```
int sumarrayrows(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;
}
```

Address	0	4	8	12	16	20
Contents	a_{00}	a_{01}	a_{02}	a_{10}	a_{11}	a_{12}
Access order	1	2	3	4	5	6

Program with a **good** spatial locality

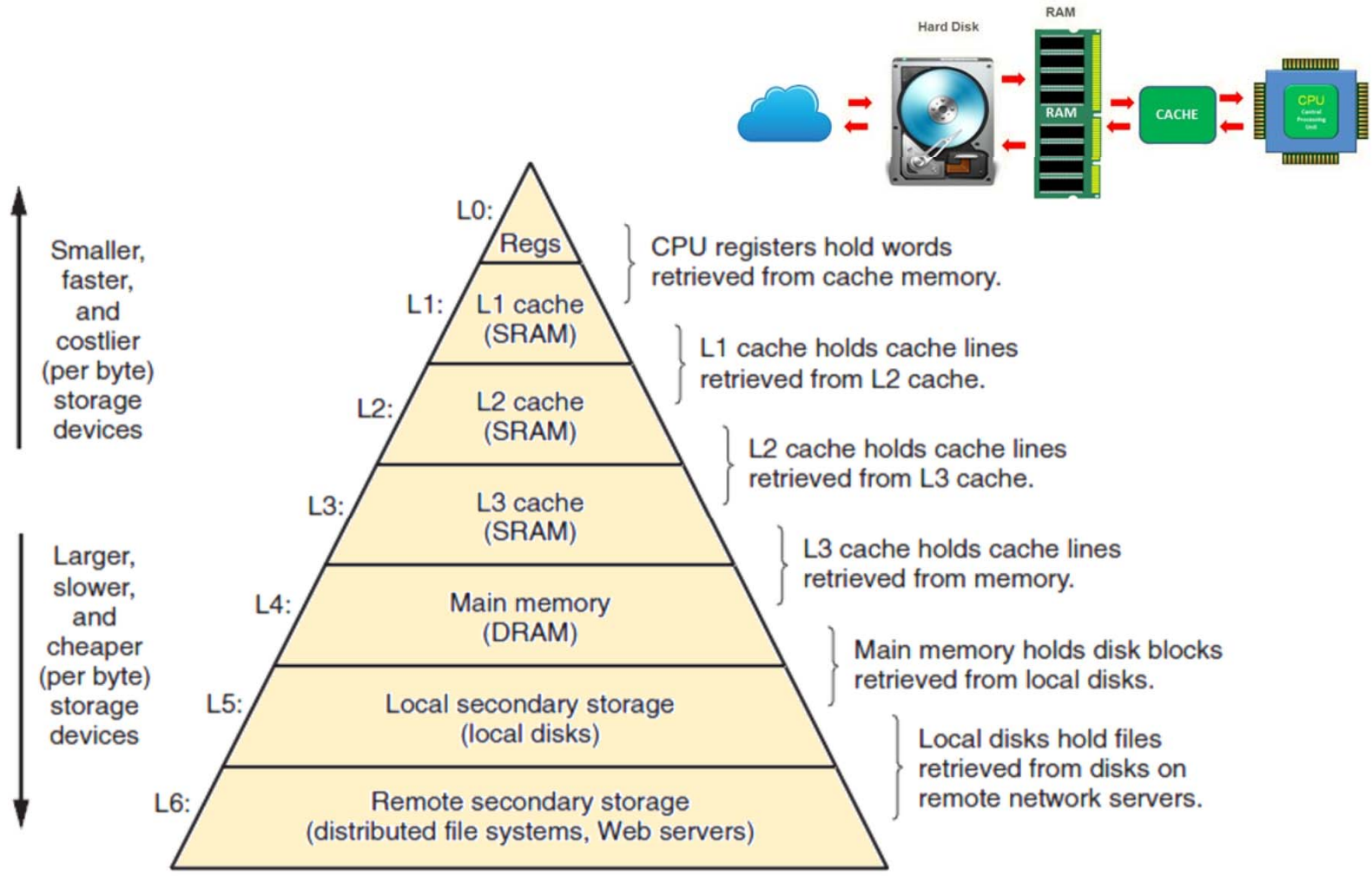
```
int sumarraycols(int a[M][N])
{
    int i, j, sum = 0;

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

Address	0	4	8	12	16	20
Contents	a_{00}	a_{01}	a_{02}	a_{10}	a_{11}	a_{12}
Access order	1	3	5	2	4	6

Program with a **poor** spatial locality

SUNY Korea

Memory Hierarchy



SUNY Korea

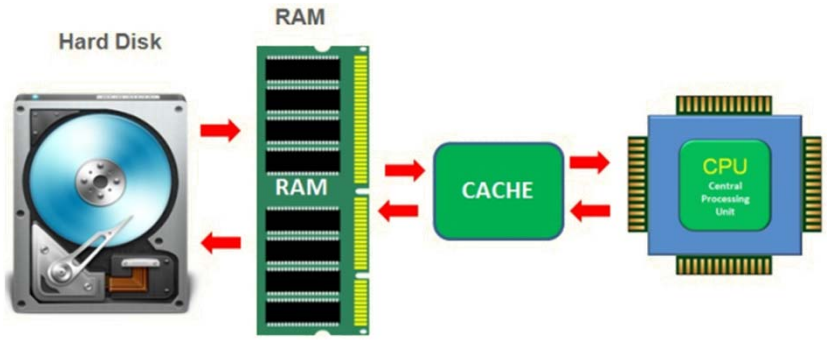
Cache

- Kinds of cache misses
  - Cold miss: initially empty cache
  - Conflict miss: cache is large enough to hold the referenced data objects, but they are mapped to the same cache block

SUNY Korea

Cache

- Cache management
  - Registers: allocated by compilers
  - L1, L2, L3 caches: managed by hardware logic
  - DRAM (Virtual memory): Operating System and hardware



SUNY Korea