Understand Storage Technologies and Trends

Identify Locality in a Program

Understand Caching in the Memory Hierarchy

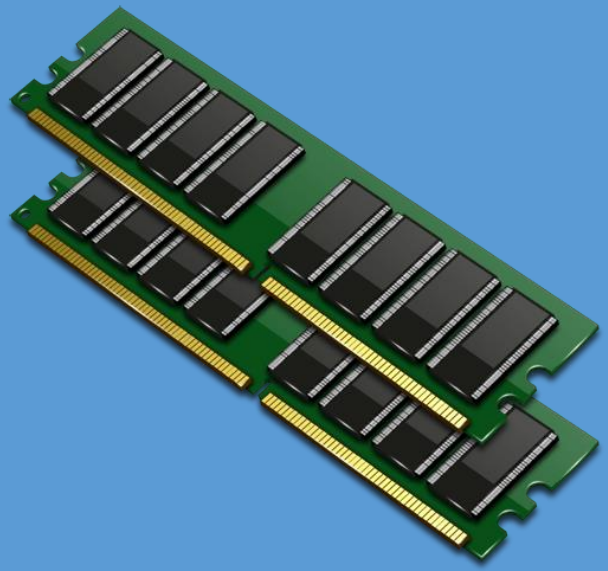
The Memory Hierarchy

① Storage Technologies and Trends

② Locality of Reference

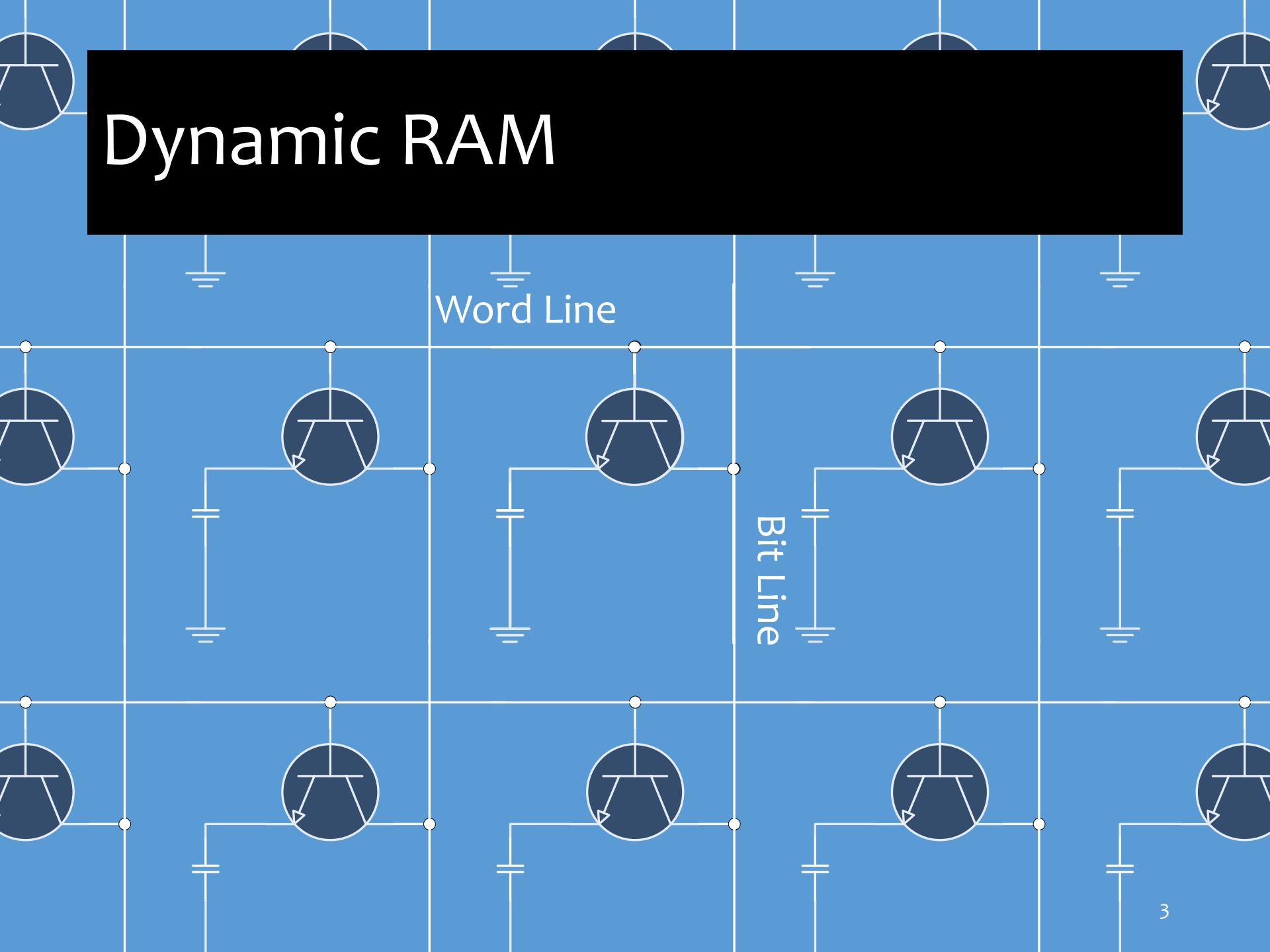
③ Caching in the Memory Hierarchy

Random-access Memory (RAM)

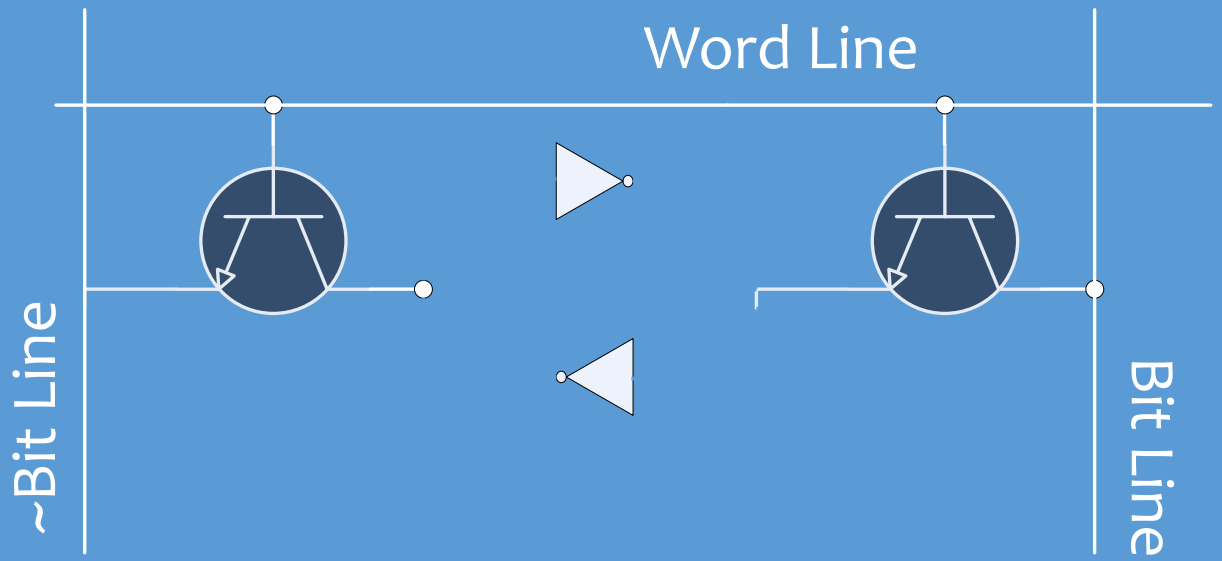


	Trans /bit	Access time	Needs refresh	Cost	Applications
SRAM	4 or 6	1x	No	100x	Cache memories
DRAM	1	10x	Yes	1x	Main memories

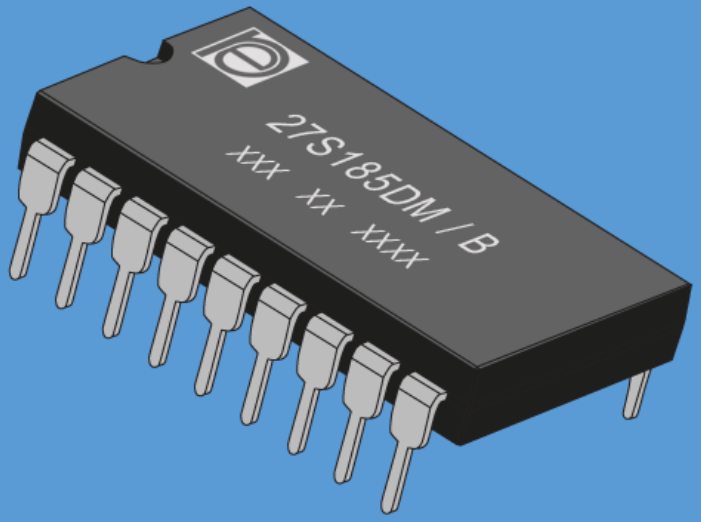
Dynamic RAM



Static RAM

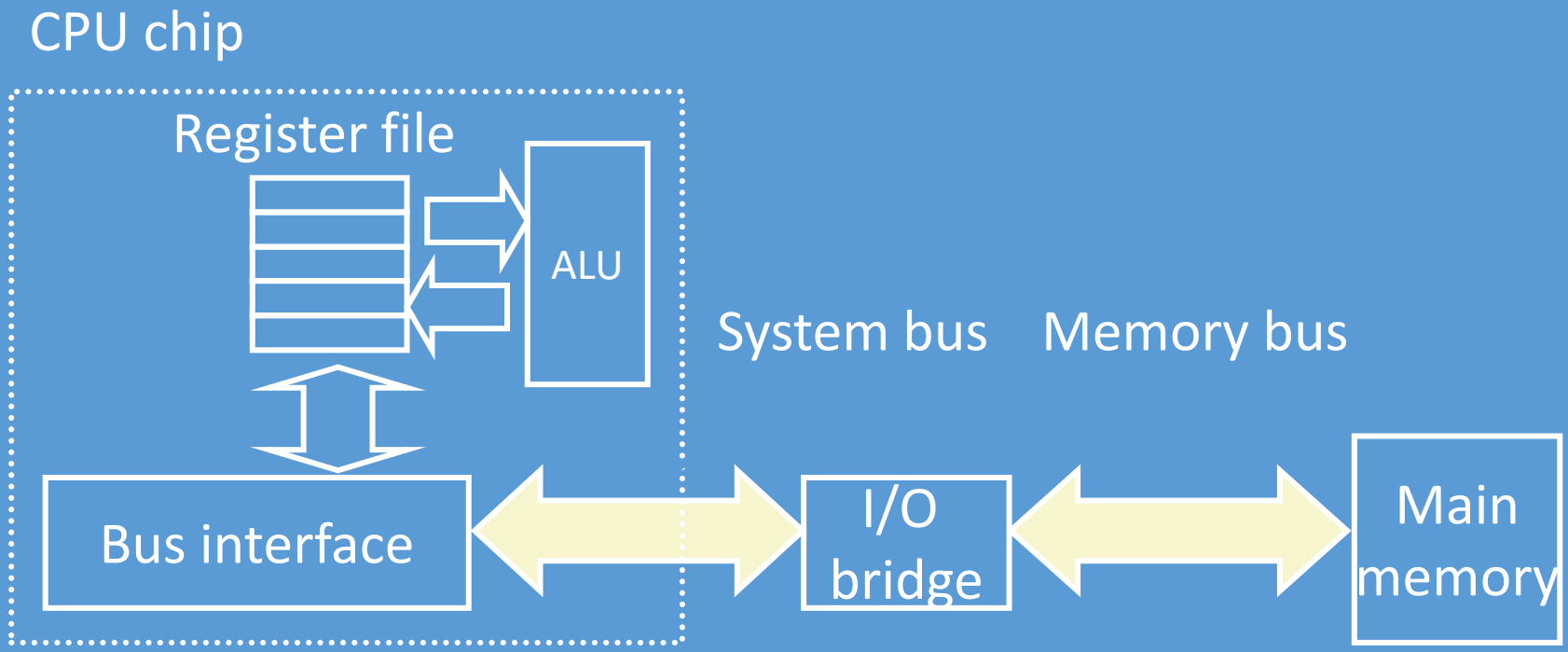


Nonvolatile Memories



Read-only memory (ROM)
Programmable ROM (PROM)
Erasable PROM (EPROM)
Electrically erasable PROM (EEPROM)
Flash memory

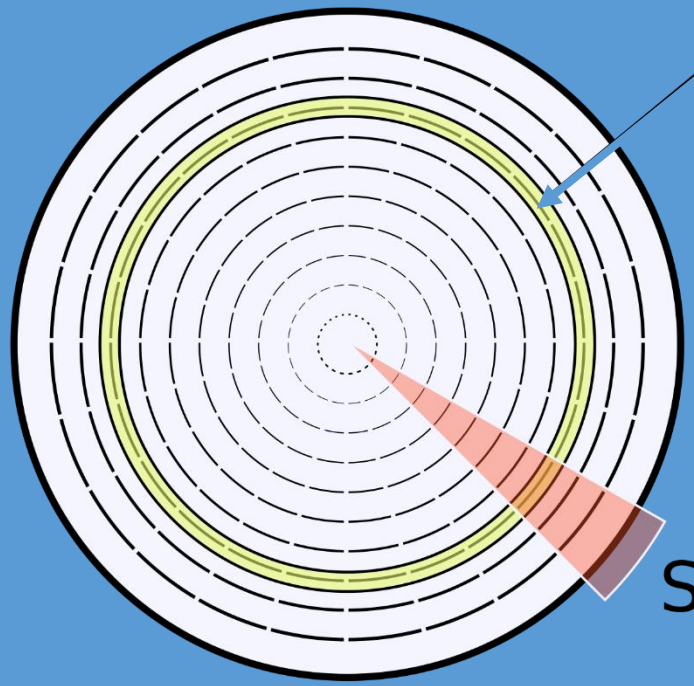
Bus

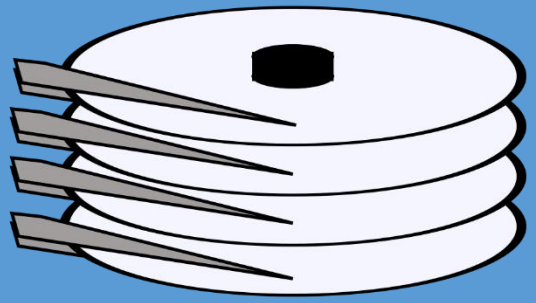

CPU chip
Register file
ALU
System bus
Memory bus
Bus interface
I/O bridge
Main memory

Rotating Disk

Disk Access

Track/Cylinder

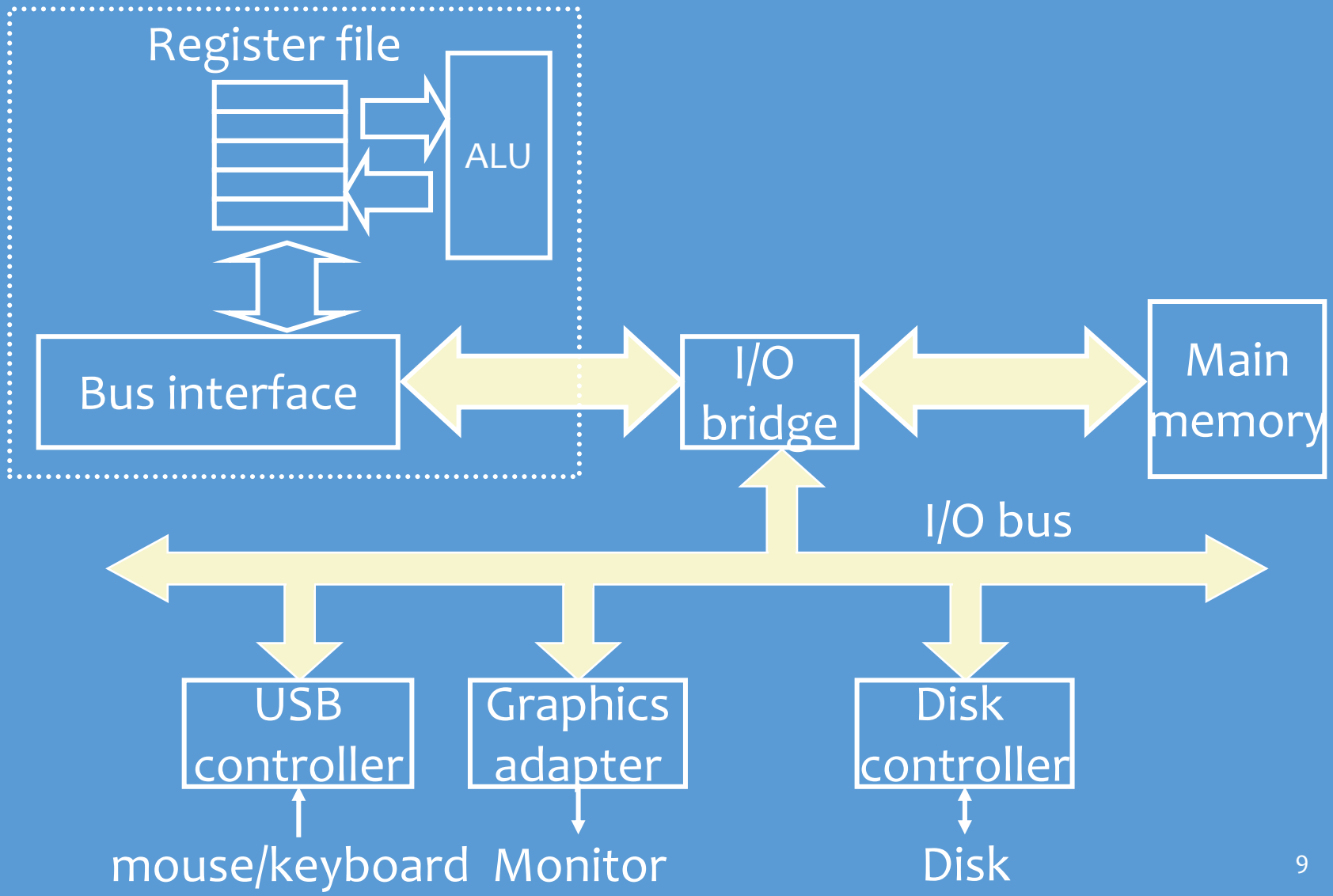




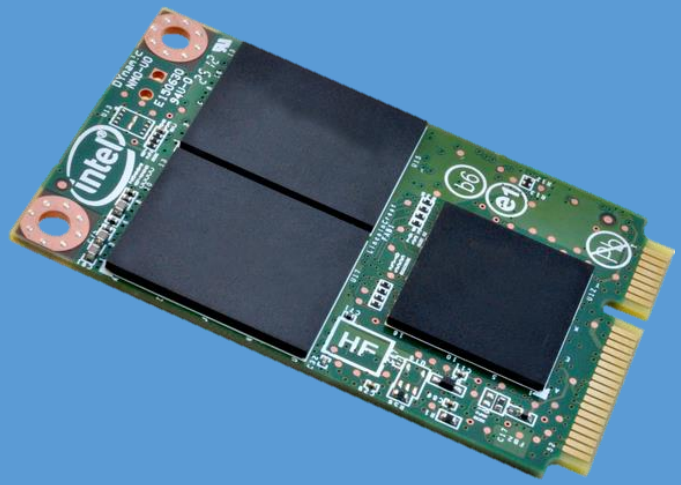
Heads

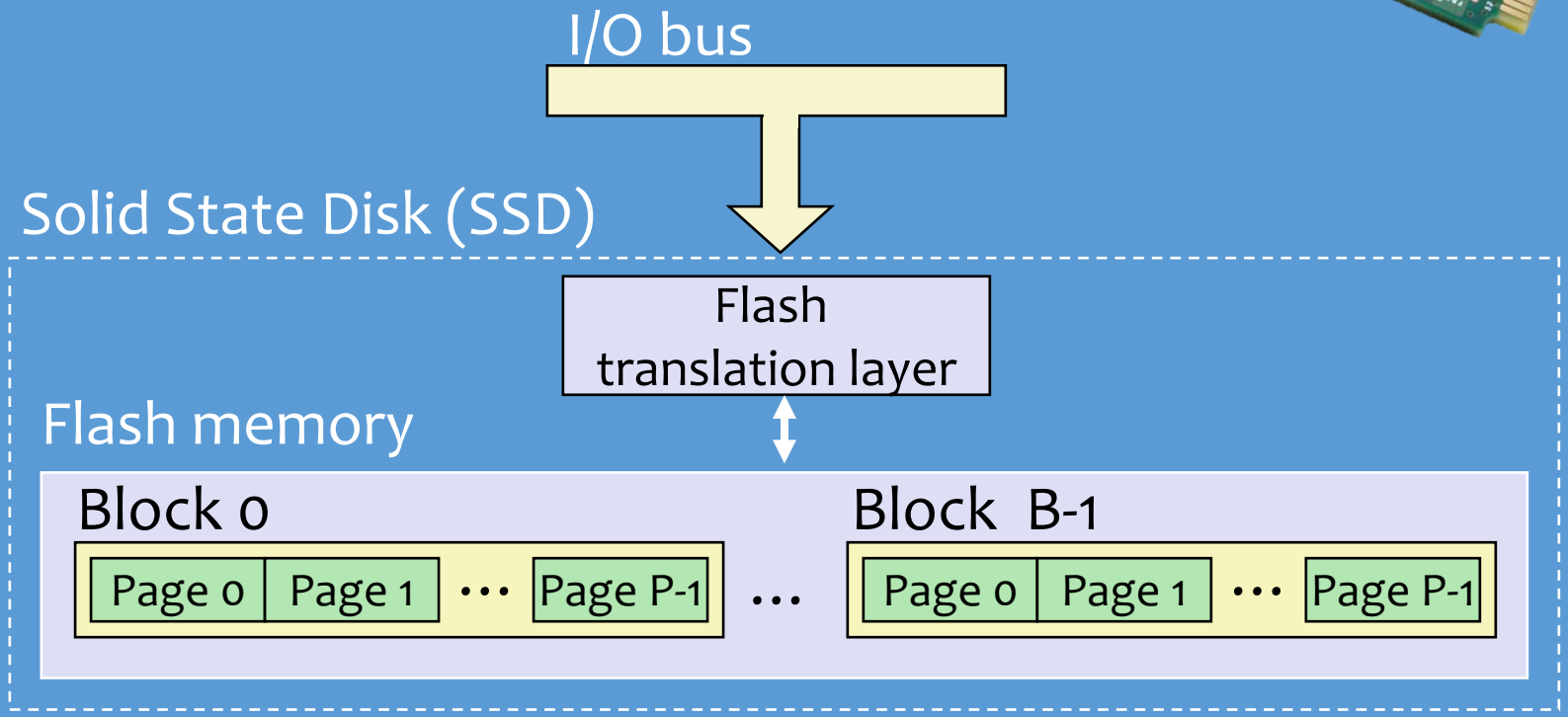
8 Heads,
4 Platters

Reading a Disk Sector

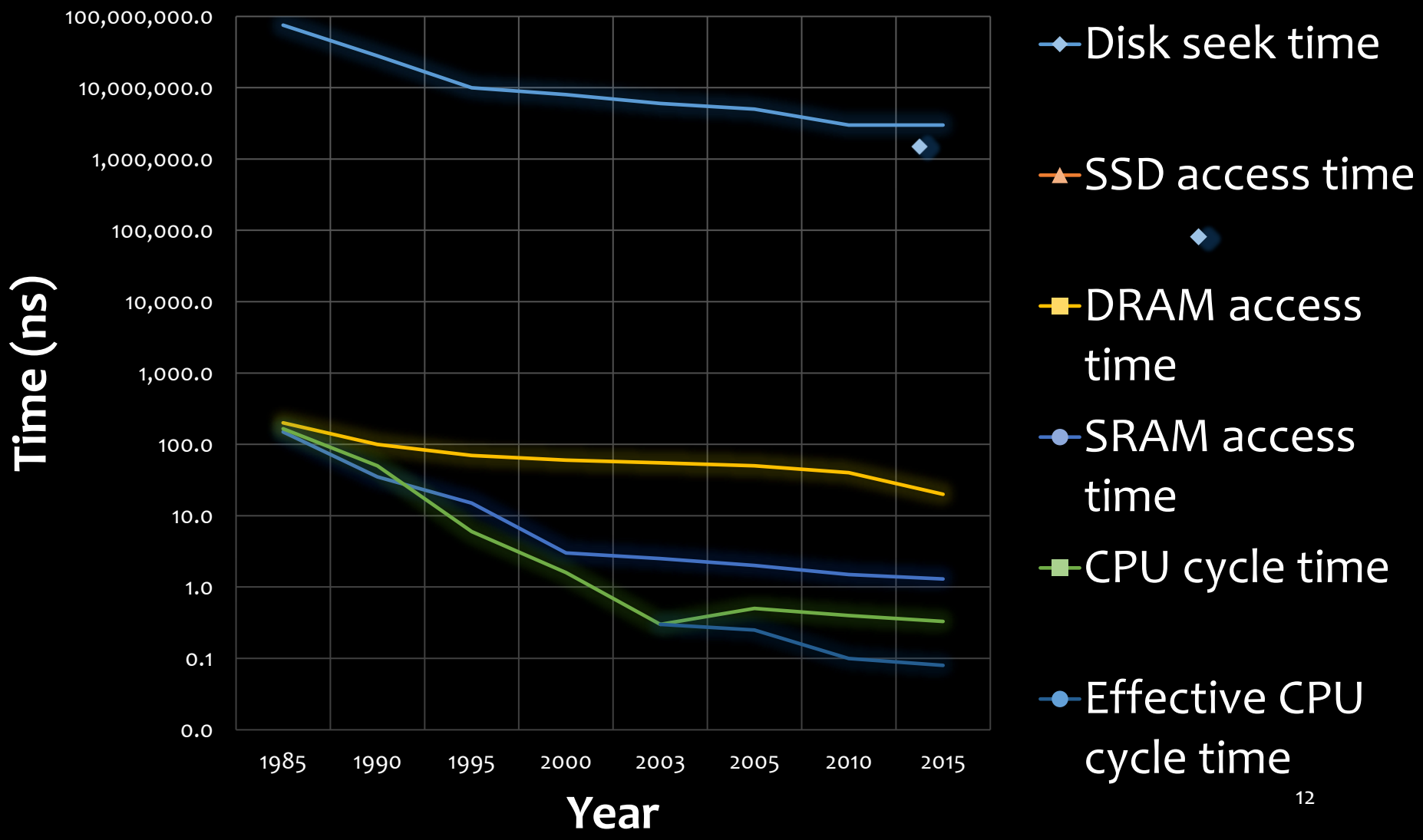


Solid State Disks





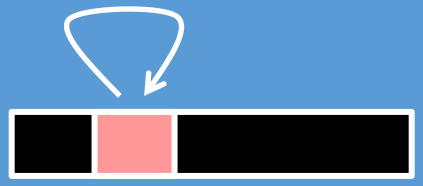
The CPU-Memory Gap



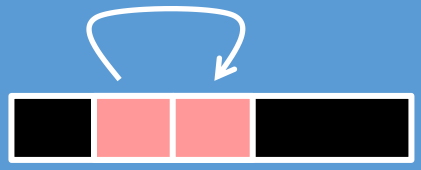
Locality

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

Temporal locality



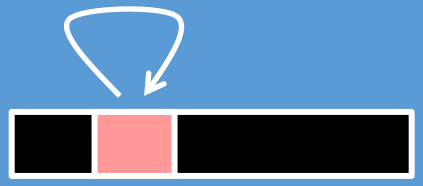
Spatial locality



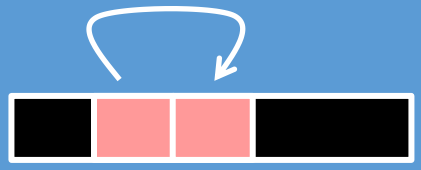
Locality Example

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

Temporal locality



Spatial locality



Q: 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;
}
```

Q: 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;
}
```

Q: 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[M][N][N])
{
    int i, j, k, sum = 0;

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

Fundamental properties

- 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.

Smaller,
Faster,
costlier

Regs

L1 cache
(SRAM)

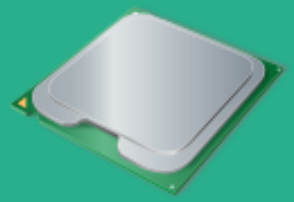
L2 cache
(SRAM)

L3 cache
(SRAM)

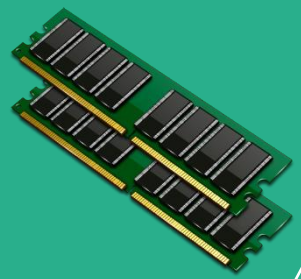
Main memory
(DRAM)

Local secondary storage
(local disks)

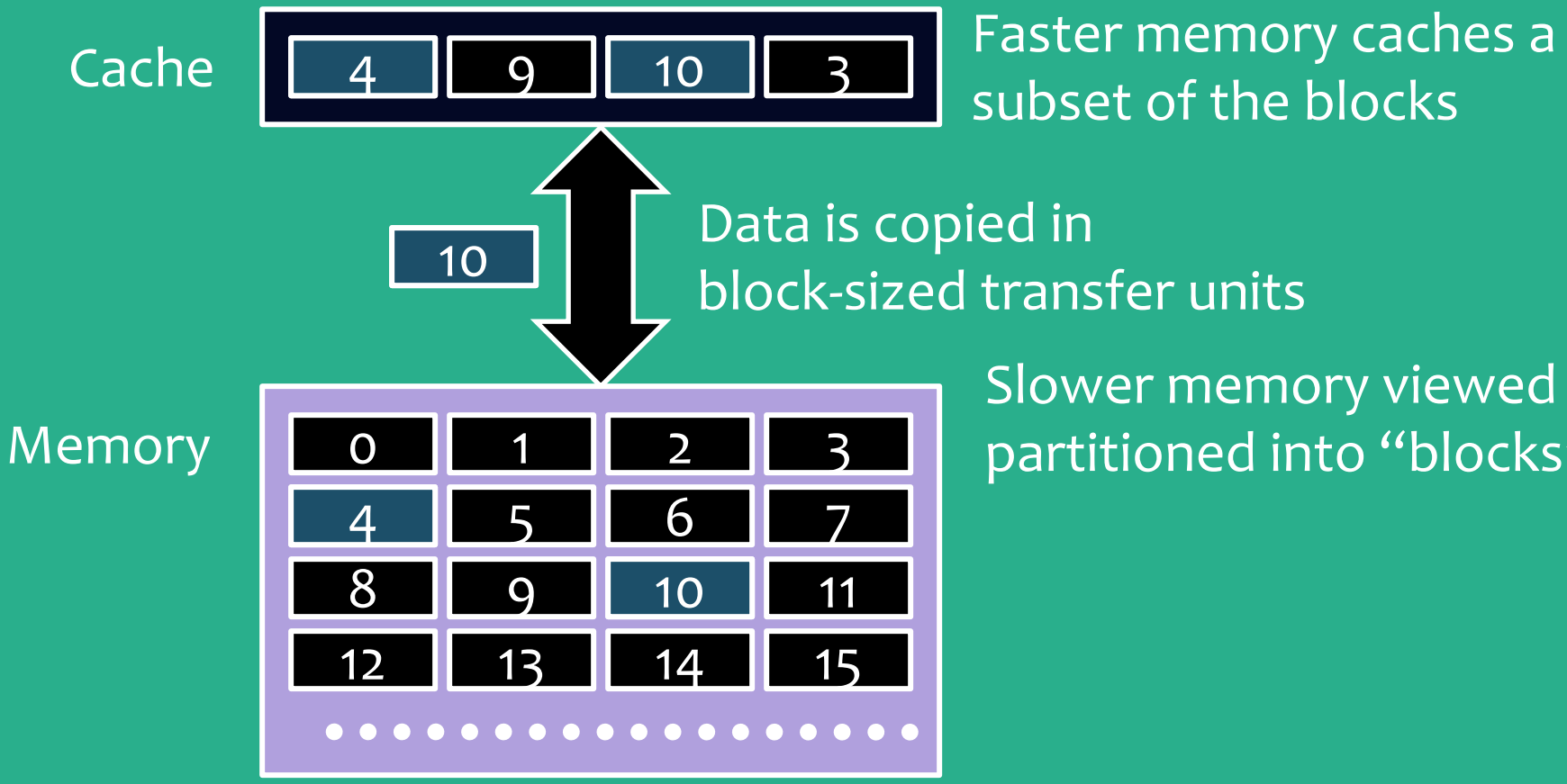
Remote secondary storage
(distributed file systems, Web servers)



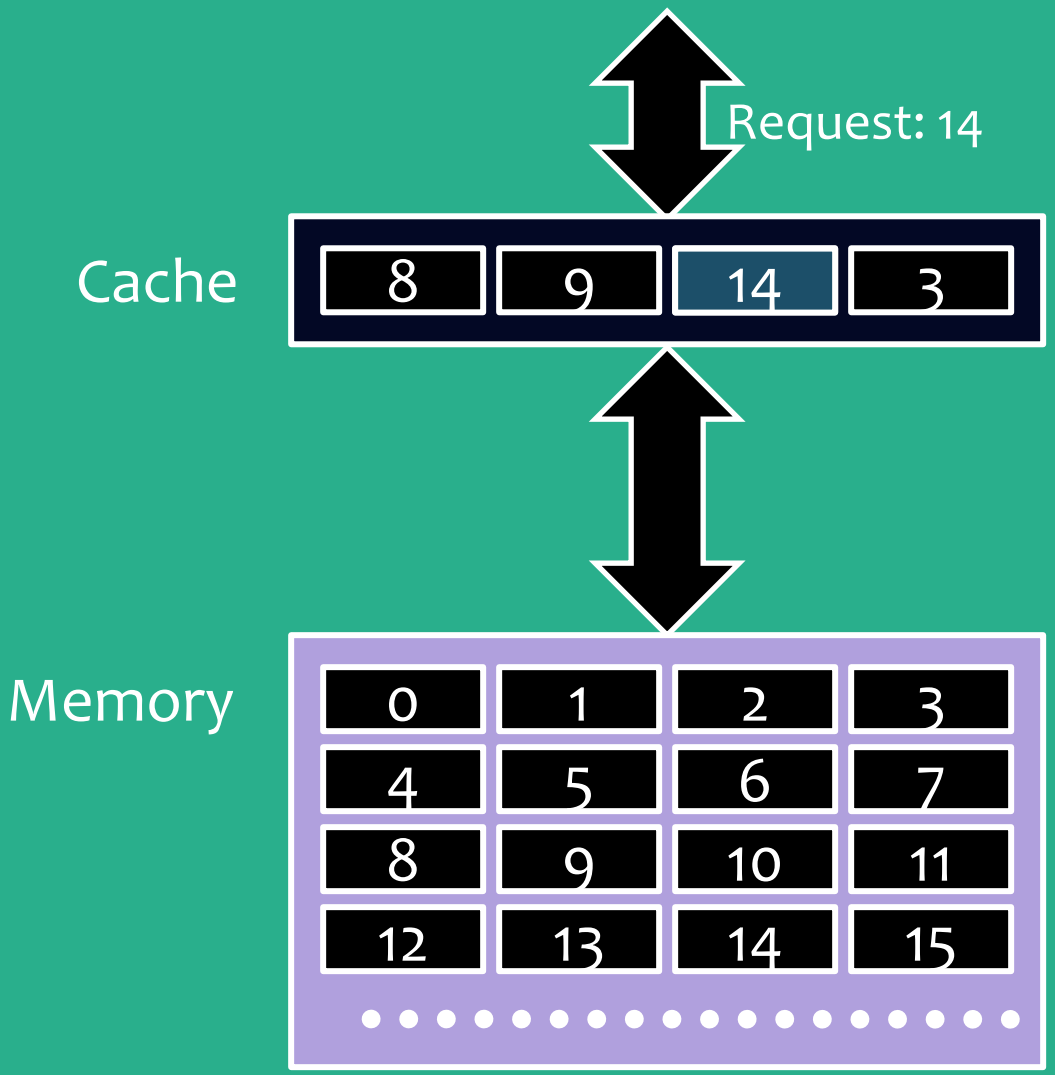
Larger,
slower,
cheaper



General Cache Concepts



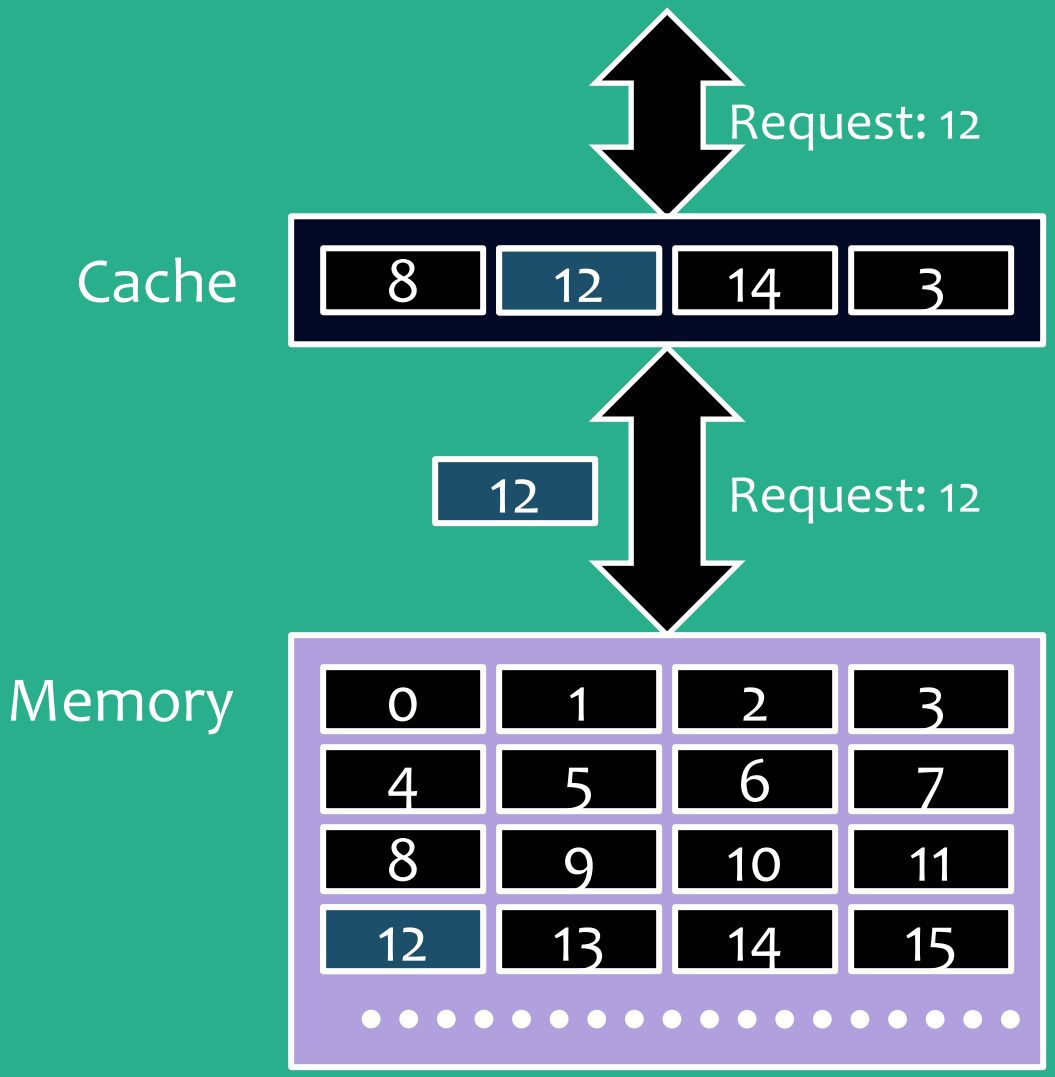
Cache Hit



Data in block b is needed

Block b is in cache:
Hit!

Cache Miss



Data in block b is needed

Block b is not in cache: Miss!

Block b is fetched from memory

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 MMU
L1 cache	64-byte blocks	On-Chip L1	4	Hardware
L2 cache	64-byte blocks	On-Chip L2	10	Hardware
Virtual Memory	4-KB pages	Main memory	100	Hardware + OS
Buffer cache	Parts of files	Main memory	100	OS
Disk cache	Disk sectors	Disk controller	100K	Disk firmware
Network buffer cache	Parts of files	Local disk	10M	NFS client
Browser cache	Web pages	Local disk	10M	Web browser
Web cache	Web pages	Remote server disks	1B	Web proxy server

Summary

- Storage Technologies
  - RAM, ROM
  - Rotating disk, Solid state disk
- Locality
  - Temporal
  - Spatial
- Memory Hierarchy
- Cache
  - Hit
  - Miss

William H. Gates III

Bill Gates

Co-founder of Microsoft

> “No one will need more than 637Kb of memory for a personal computer.”