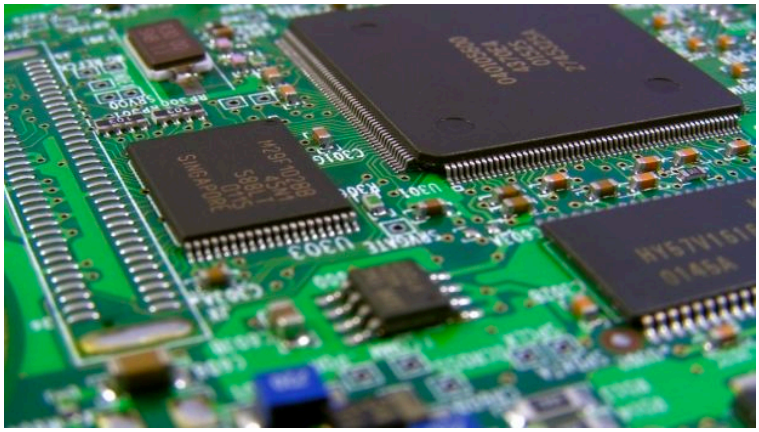
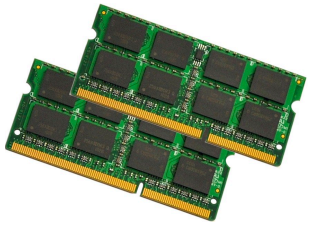
High-Performance Data Storage



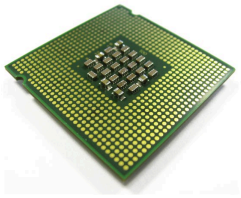
Data Storage

- Disks
  - Hard disk (HDD)
  - Solid state drive (SSD)

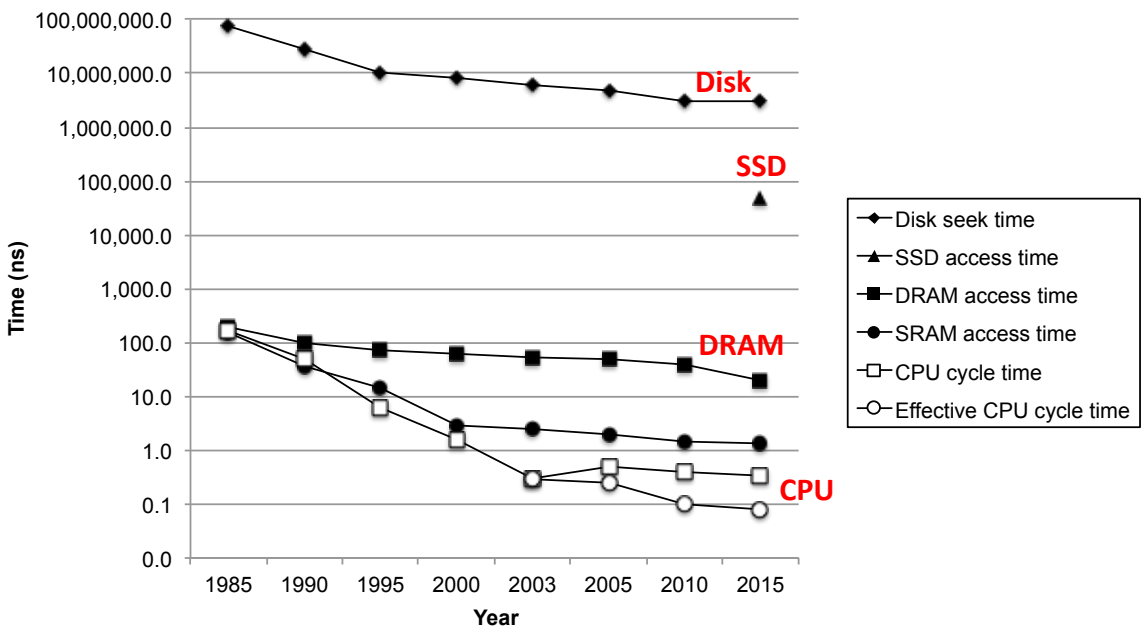
- Random Access Memory
  - Dynamic RAM (DRAM)
  - Static RAM (SRAM)



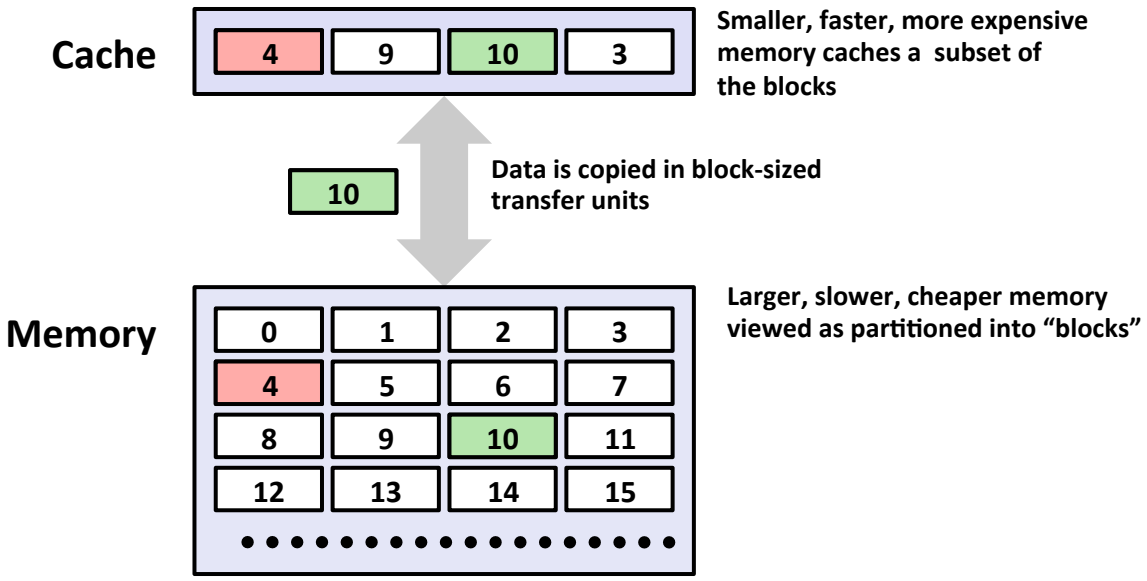
- Registers
  - %rax, %rbx, ...



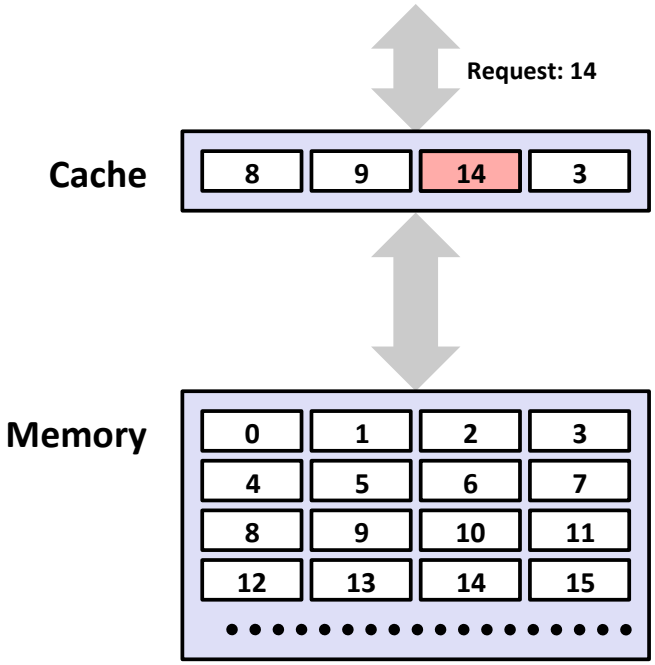
The CPU-Memory Gap



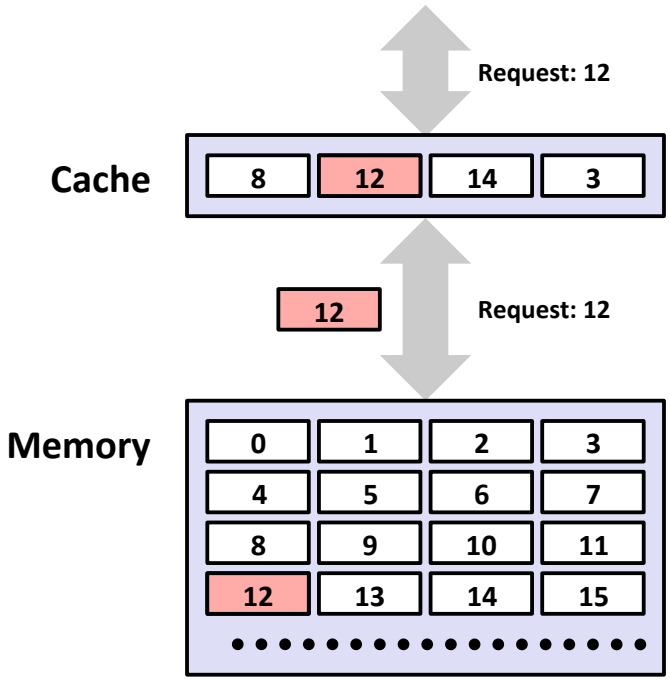
Caching



Cache Hit



Cache Miss



Direct-Mapped Cache

Line	V	D	Tag	Data (8 Bytes)
0				
1				
2				
3				
4				
…	…			
1020				
1021				
1022				
1023				

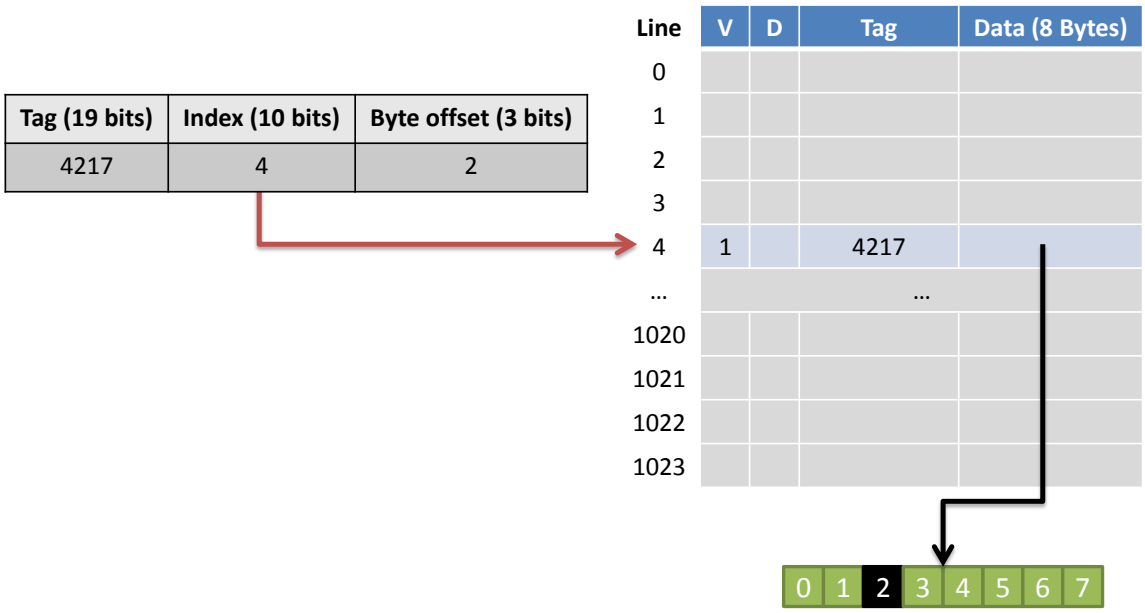
Direct-Mapped Address Components

(assumes 32-bit addresses)

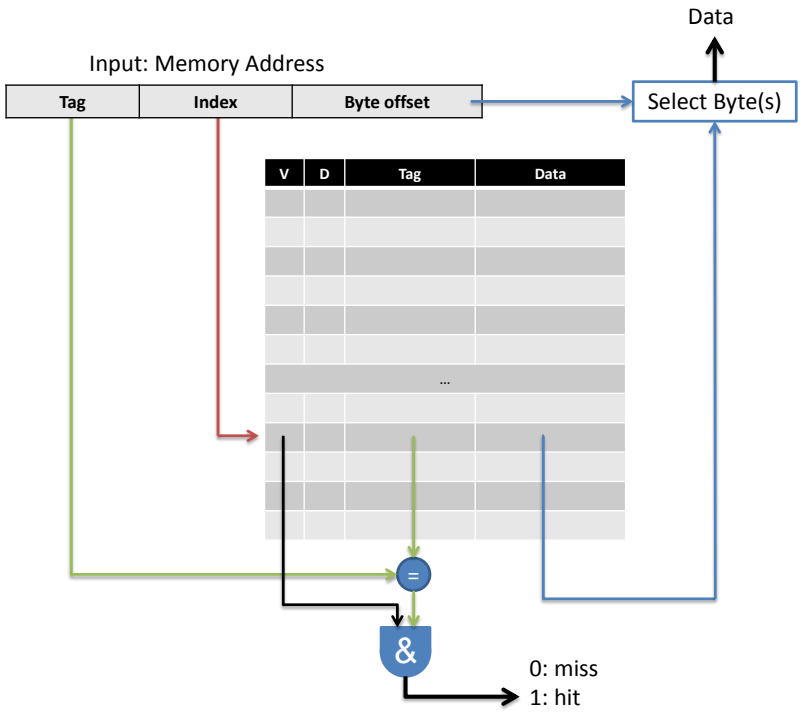
Tag (19 bits)	Index (10 bits)	Byte offset (3 bits)

Line	V	D	Tag	Data (8 Bytes)
0				
1				
2				
3				
4				
…	…			
1020				
1021				
1022				
1023				

Direct-Mapped Address Components



Direct-Mapped Cache Lookup



2-Way Set Associative (1024 lines)

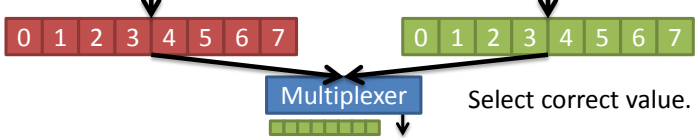
Tag (20 bits)	Set (9 bits)	Byte offset (3 bits)
3941	4	

Set #	V	D	Tag	Data (8 Bytes)	V	D	Tag	Data (8 Bytes)
0								
1								
2								
3								
4	1	1	4063		1	0	3941	
…			…				…	
508								
509								
510								
511								

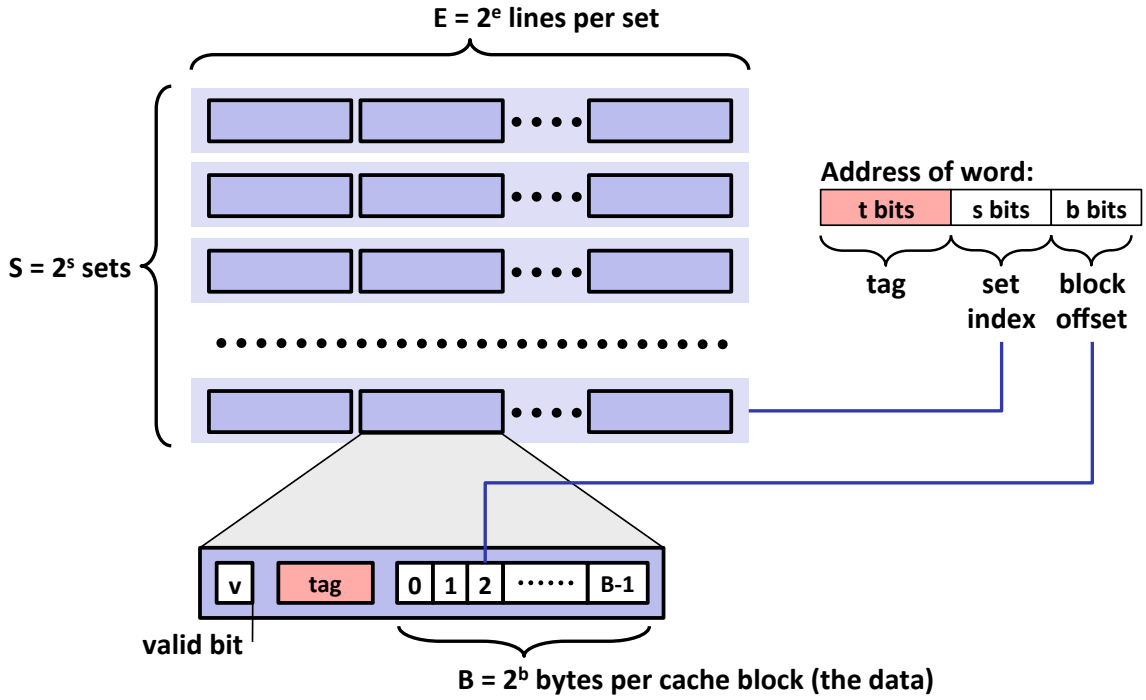
2-Way Set Associative Line Matching

Tag (20 bits)	Set (9 bits)	Byte offset (3 bits)
3941	4	

Set #	V	D	Tag	Data (8 Bytes)	V	D	Tag	Data (8 Bytes)
0								
1								
2								
3								
4	1	1	4063		1	0	3941	
…			…				…	
508								
509								
510								
511								

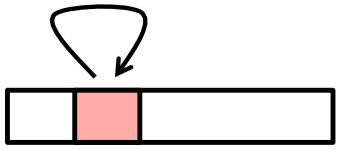


General Cache Model (S, E, B)

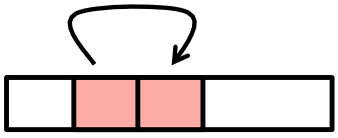


Locality

- **Temporal locality:**



- **Spatial locality:**



Locality Example

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


Locality Design

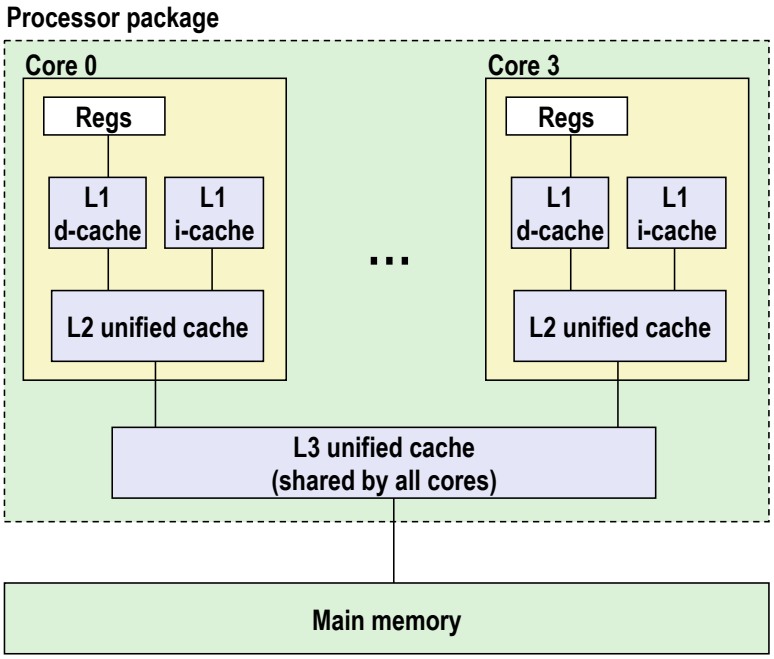
(v1)

```
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;
}
```

(v2)

```
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;
}
```

Intel Core i7 Cache Hierarchy



L1 i-cache and d-cache:
32 KB, 8-way,
Access: 4 cycles

L2 unified cache:
256 KB, 8-way,
Access: 10 cycles

L3 unified cache:
8 MB, 16-way,
Access: 40-75 cycles

Block size: 64 bytes for all caches.

The Memory Hierarchy