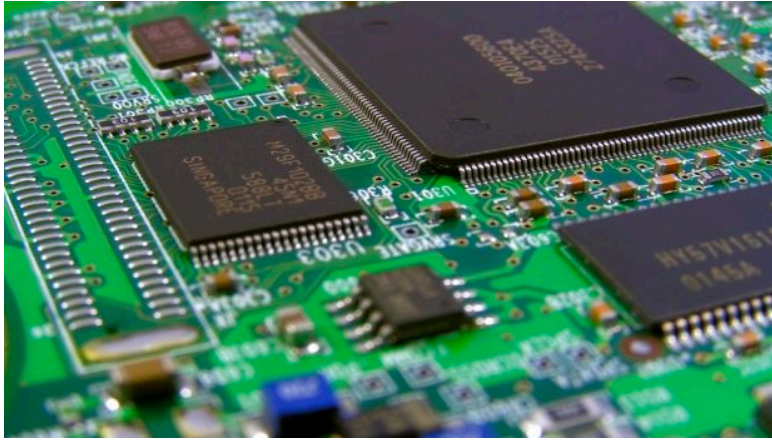


Caching



Data Storage

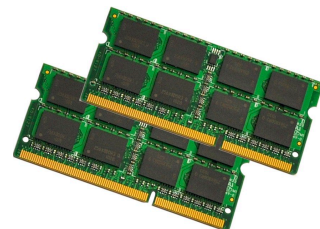
Disks

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



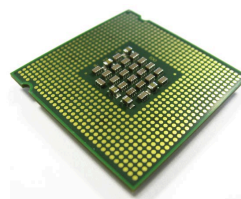
Random Access Memory

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

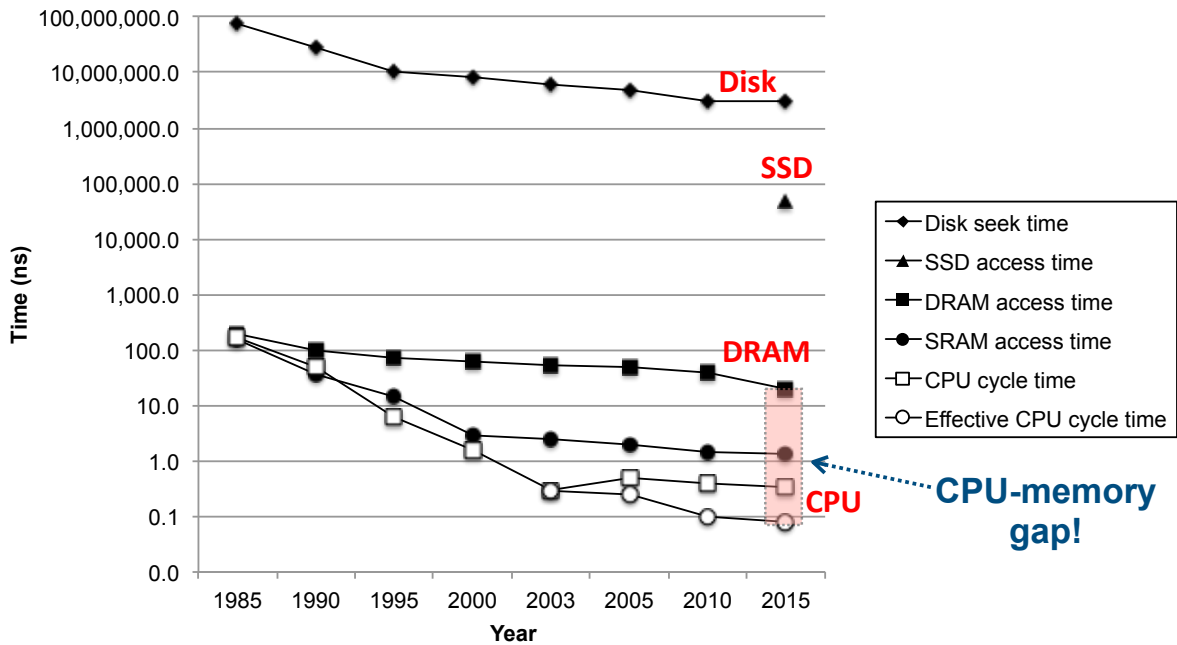


Registers

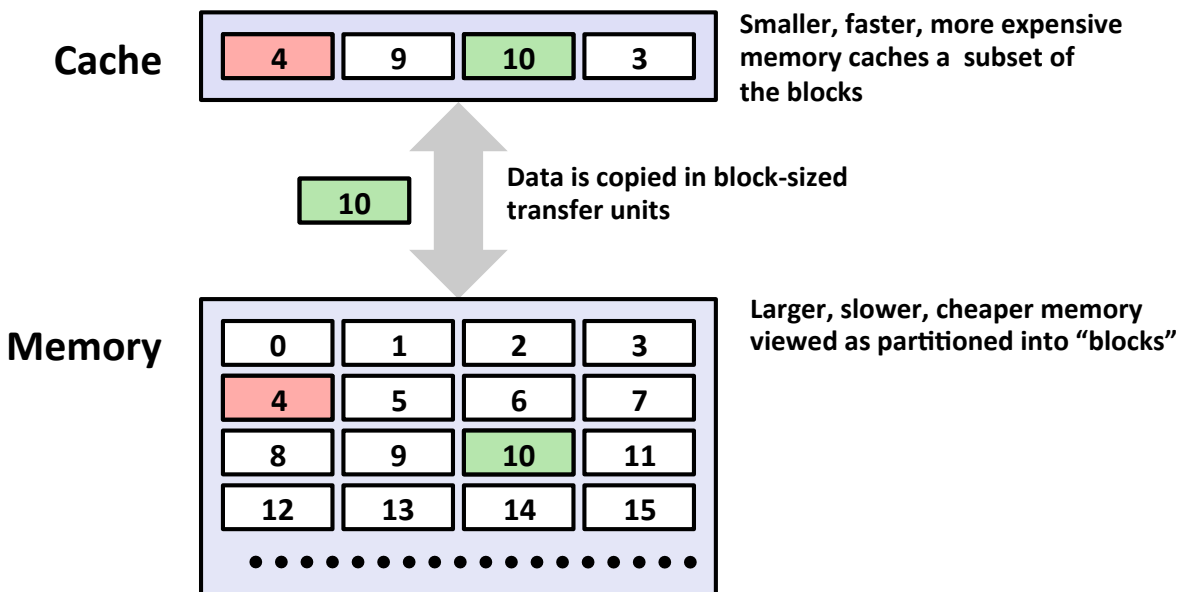
- `%rax`, `%rbx`, ...



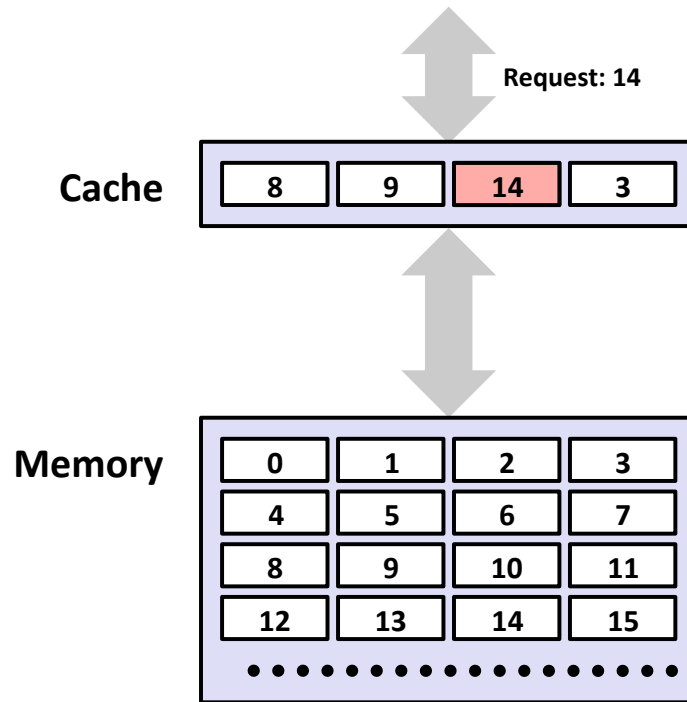
Storage Access Times



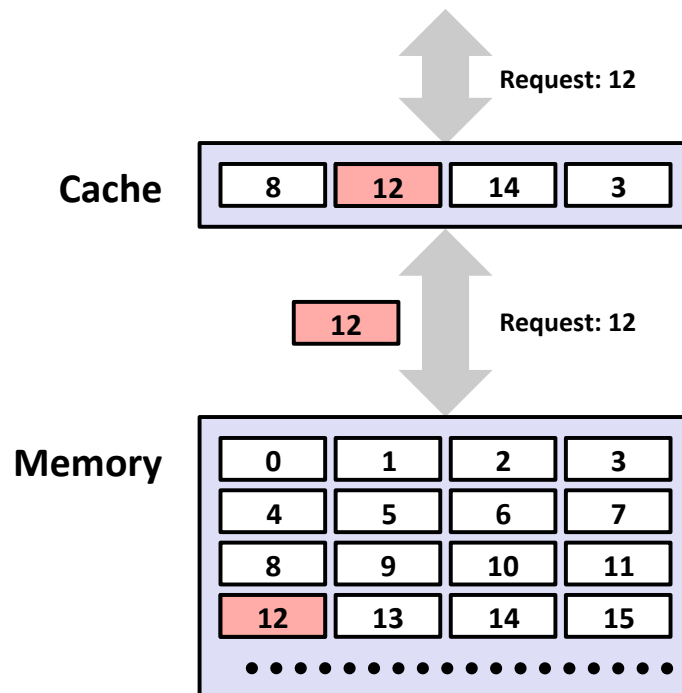
Caching Overview



Cache Hit



Cache Miss



Direct-Mapped Cache

valid bit dirty bit (assumes write-back)



1024 cache lines
 × 8-byte blocks
 = 8 KB cache size

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

Direct-Mapped Address Components

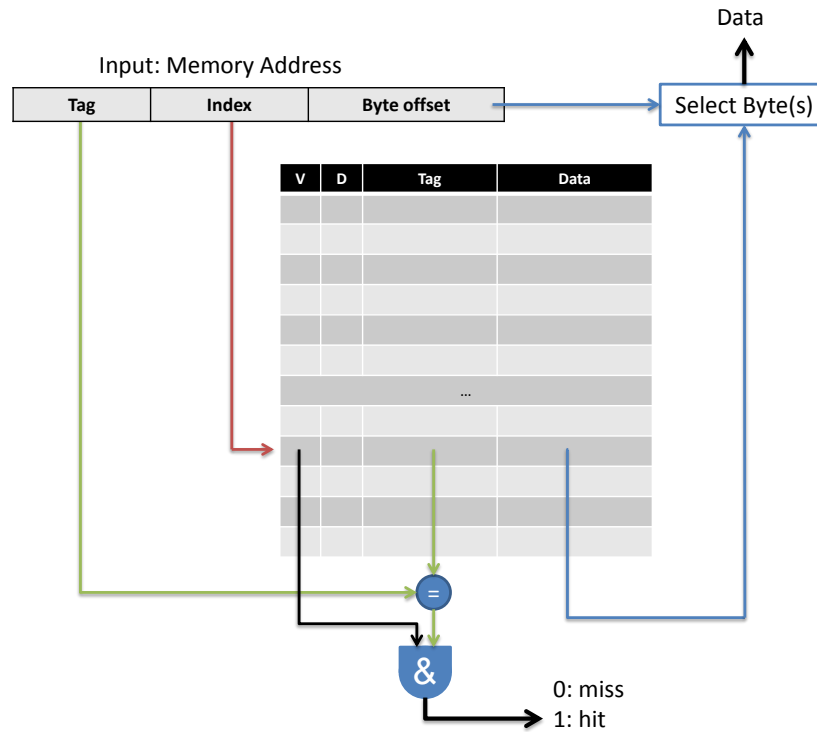
(assumes 32-bit addresses)

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

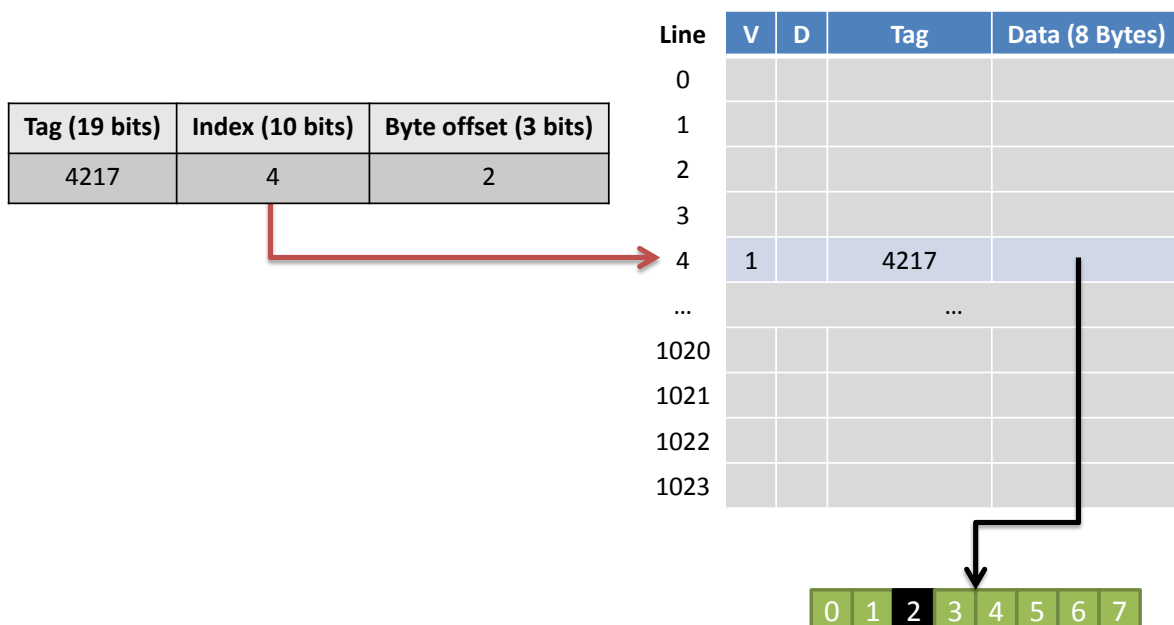


Line	V	D	Tag	Data (8 Bytes)
0				
1				
2				
3				
4				
...			...	
1020				
1021				
1022				
1023				

Direct-Mapped Cache Lookup



Direct-Mapped Address Components



Caching Exercises

tag | index | offset

Line	V	D	Tag	Data (4 Bytes)
0	1	0	111	17
1	1	0	011	9
2	0	0	101	15
3	1	1	001	8
4	1	0	011	4
5	0	0	111	6
6	0	0	101	32
7	1	0	110	3

Direct-Mapped Cache (redux)

(assumes 32-bit addresses)

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



1024 cache lines
 × 8-byte blocks
 = 8 KB cache size

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

2-Way Set Associative Cache

512 **cache sets** × 2 lines per set
 × 8-byte blocks
 = same 8 KB cache size

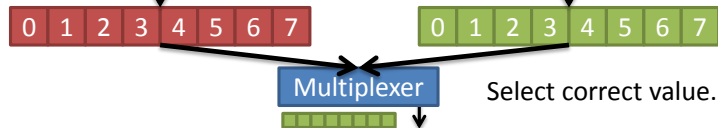
Tag (20 bits)	Set (9 bits)	Byte offset (3 bits)
3941	4	

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								

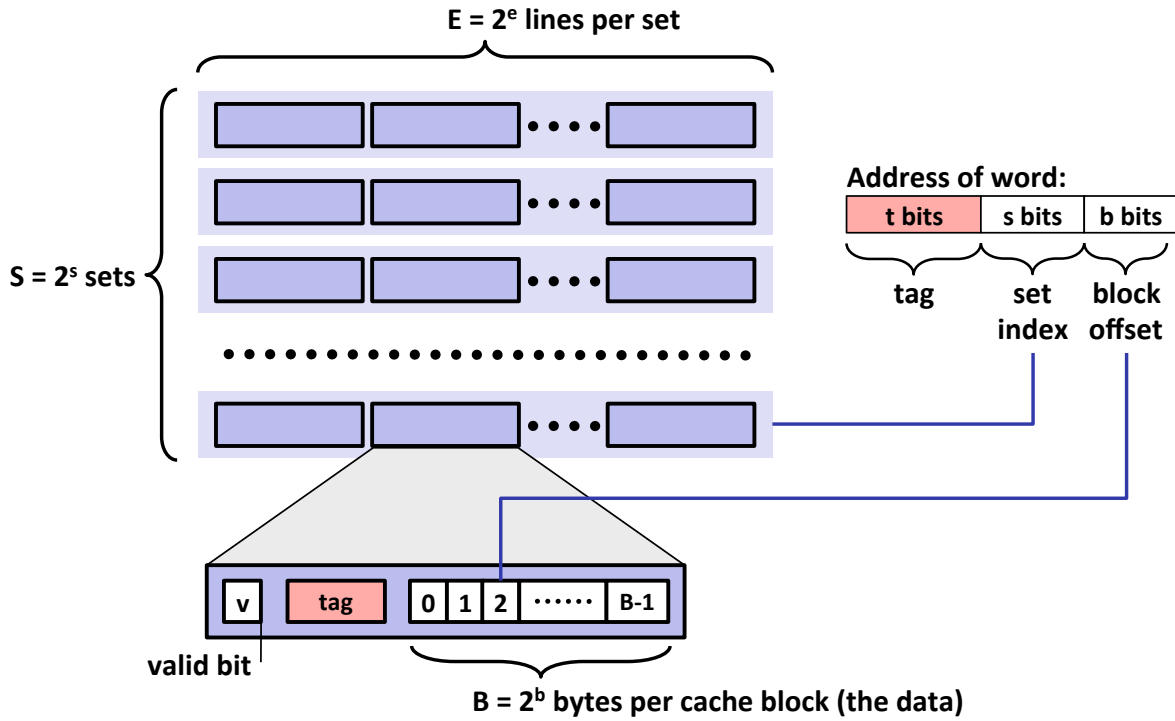
2-Way Set Associative Line Matching

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								

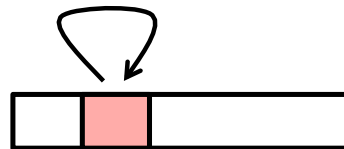


General Cache Model (S, E, B)

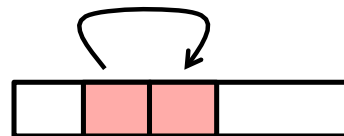


Locality

■ **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;

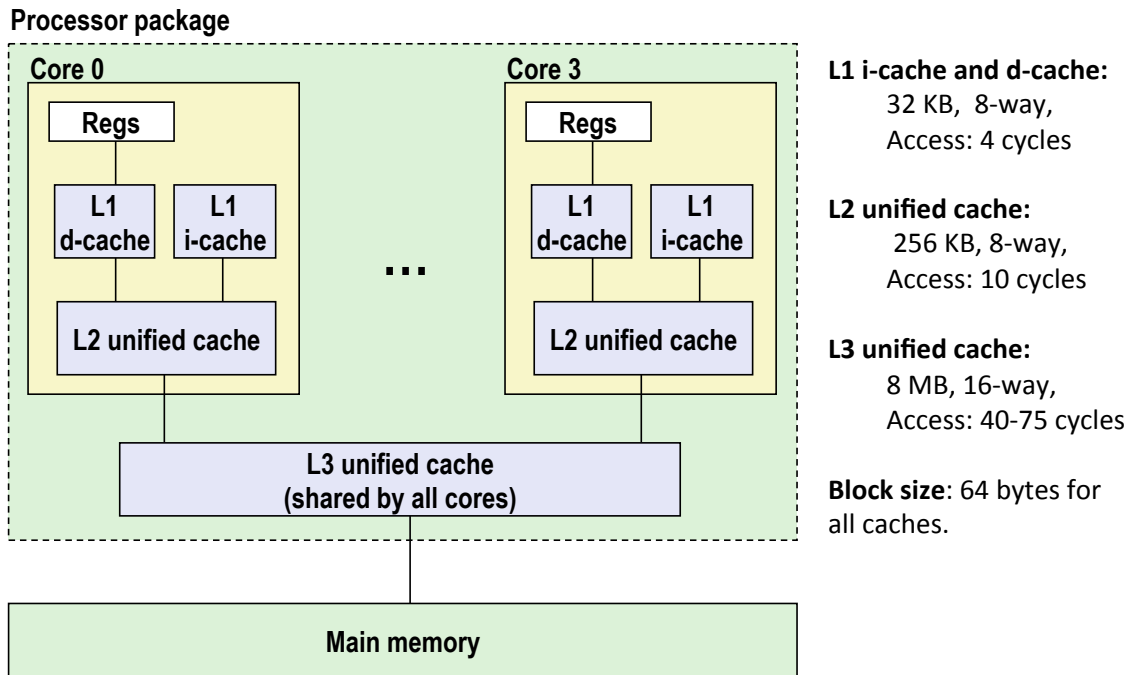
    for (i = 0; i < M; i++)
        for (j = 0; j < N; j++)
            sum += a[i][j];
    return sum;
}
```

(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 (2015)



The Memory Hierarchy

