



CS110 Discussion Week 10: Cache

Guanghui Hu

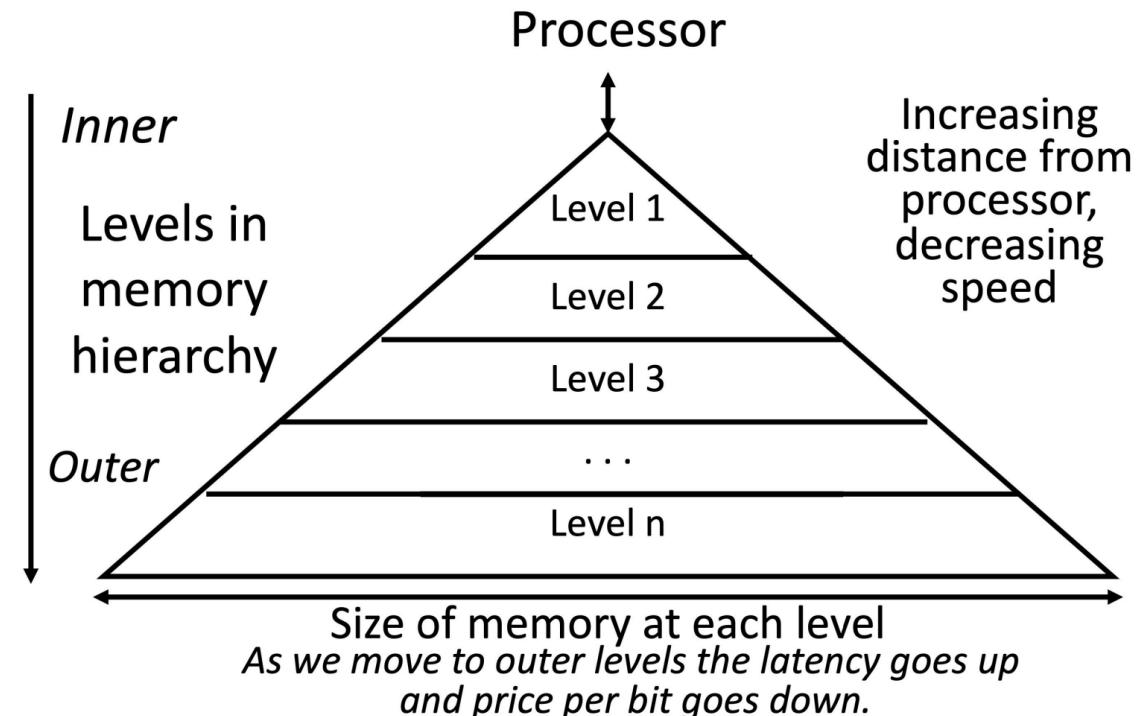
2025.4.21





Why Memory Hierarchy?

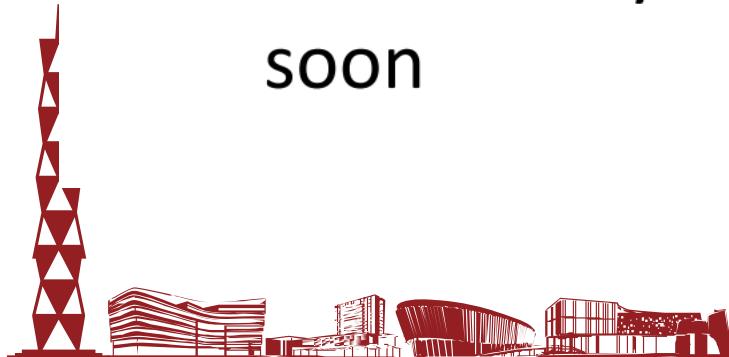
- Huge memory works slow
- Small memory works fast
- Use memory hierarchy can make CPU get most data faster



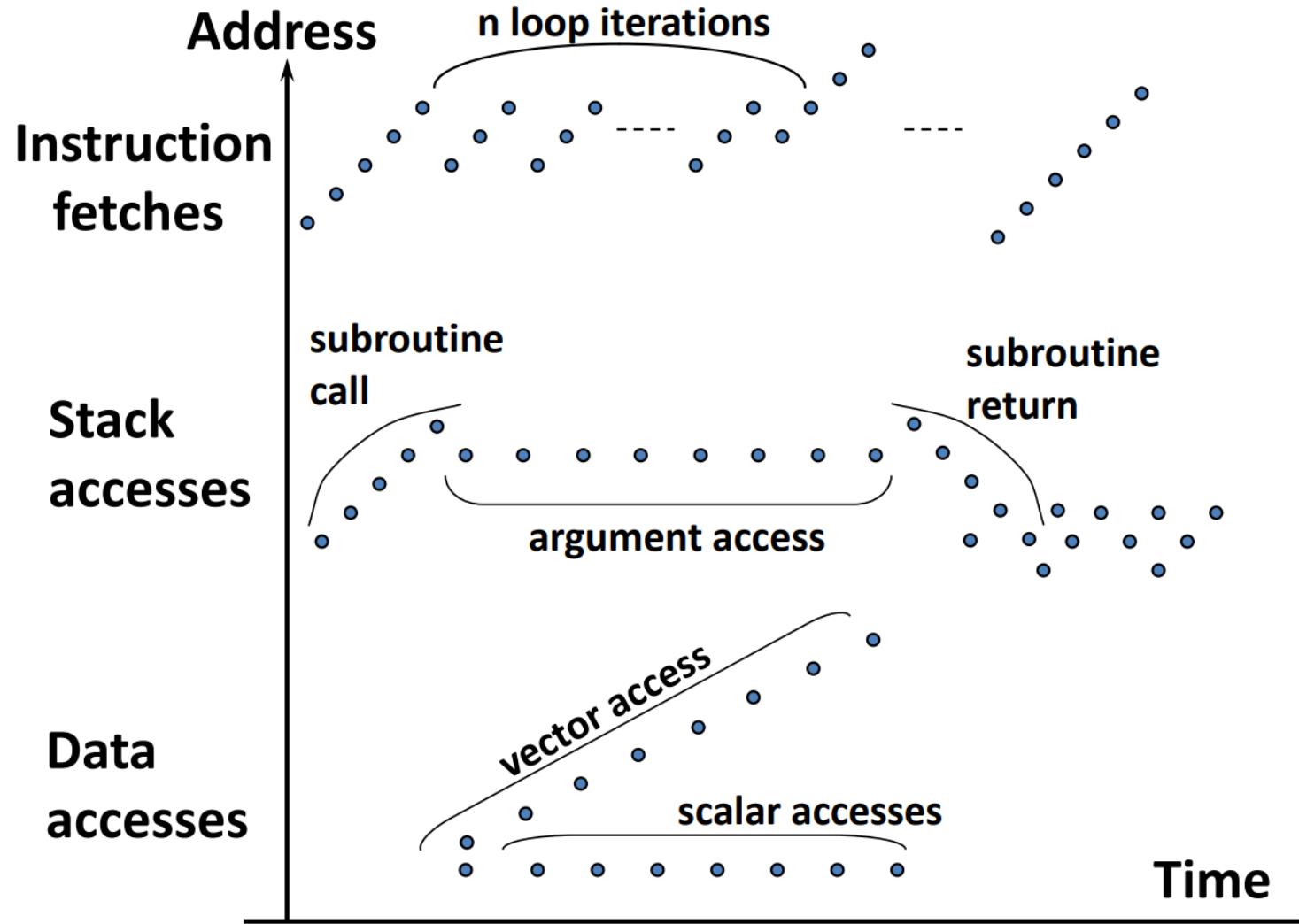
立志成才报国裕民



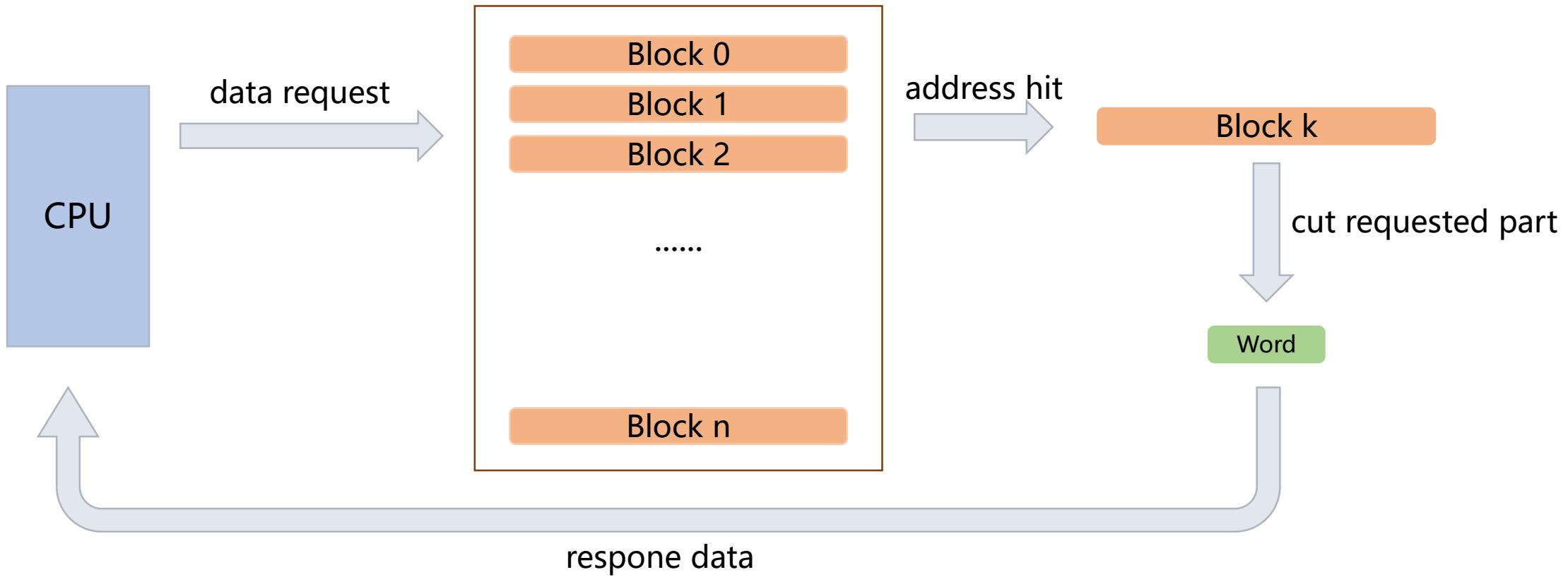
- *Temporal Locality* (locality in time)
 - If a memory location is referenced, then it will tend to be referenced again soon
- *Spatial Locality* (locality in space)
 - If a memory location is referenced, the locations with nearby addresses will tend to be referenced soon

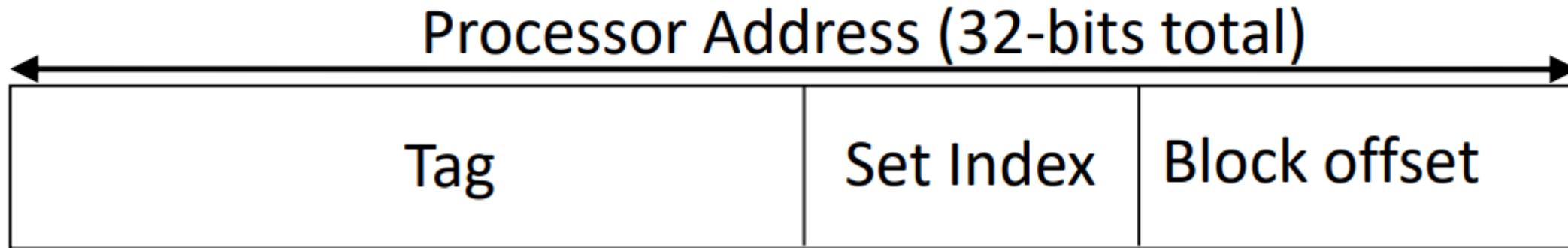


Memory Reference Patterns



Cache Design





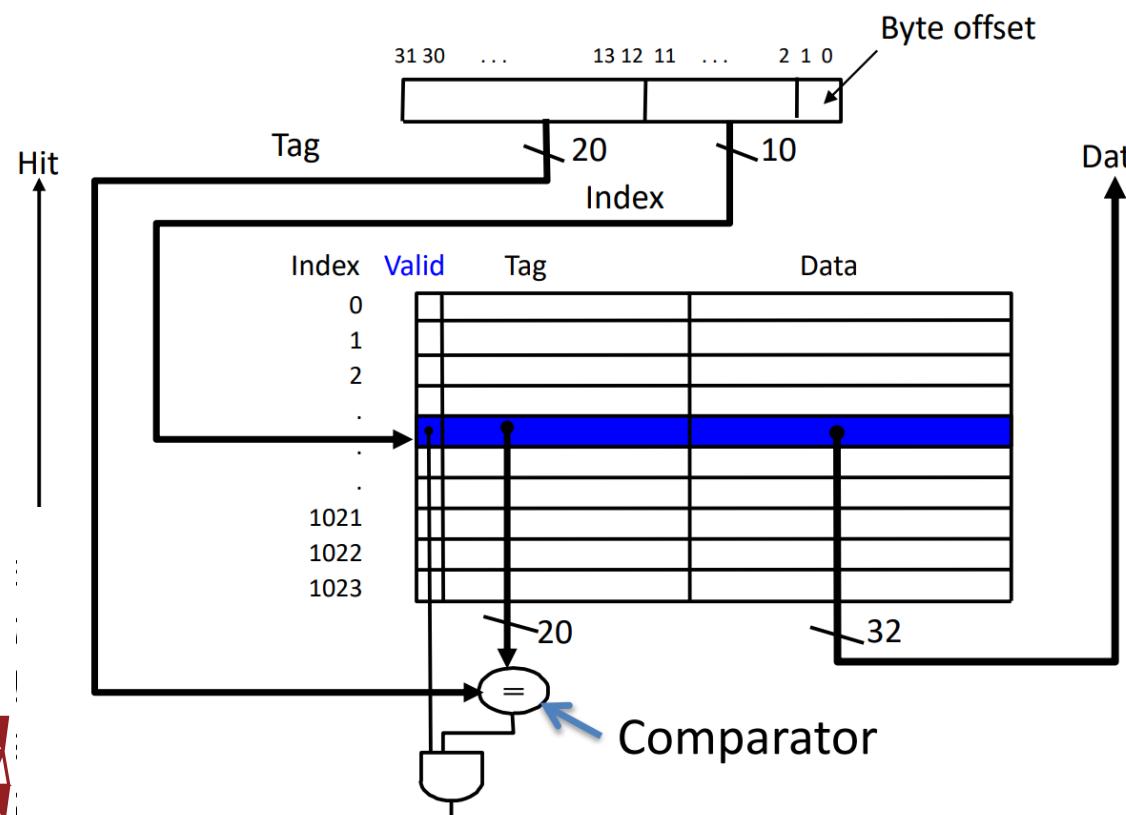
- Size of Cache Block(or Cache Line) → Block offset
- Number of sets → Set index
- Remain bits → Tag
- Associativity(or Way) → Number of cache blocks within a set

Cache size = *sets number* \times *size of block* \times *associativity*



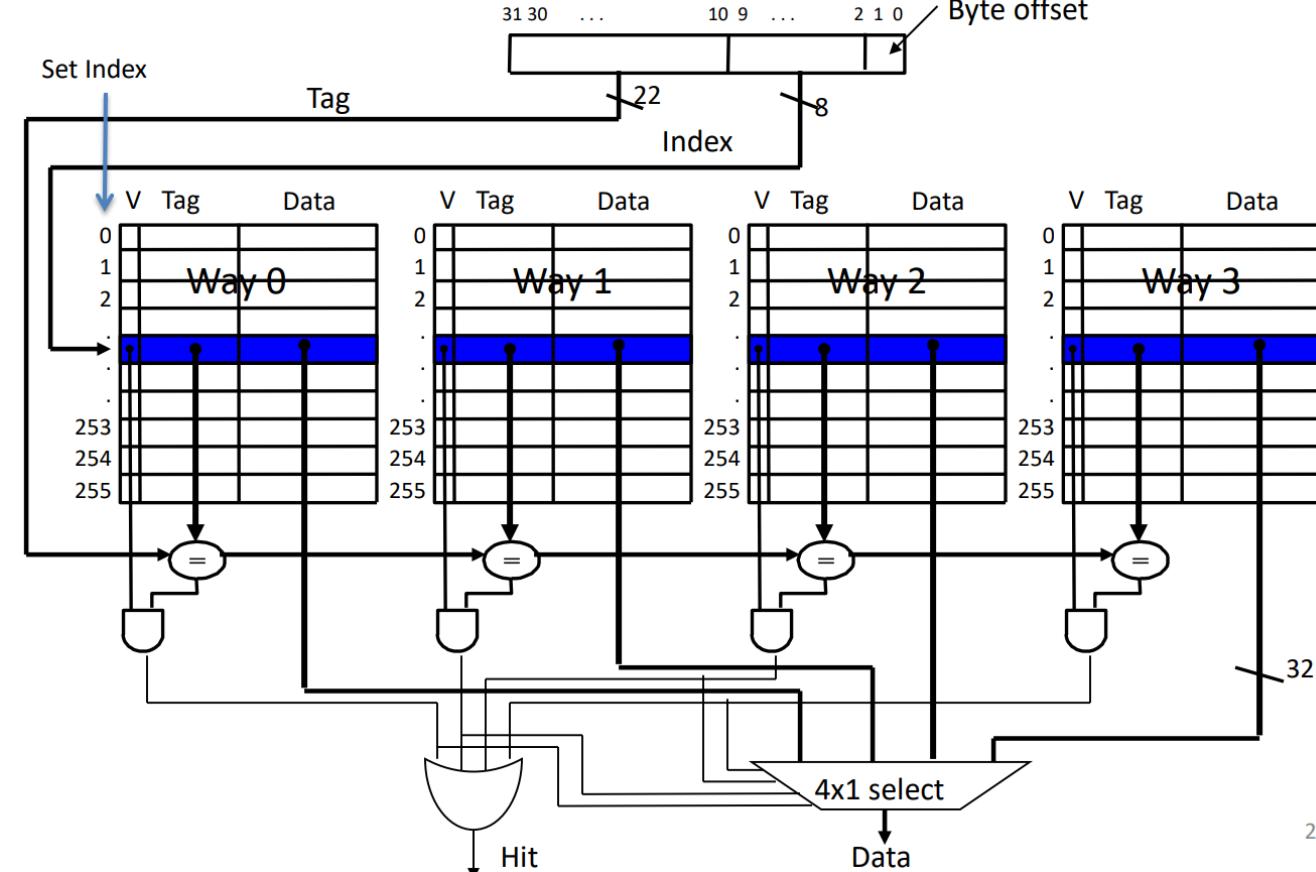
立志成才报国裕民

Direct Mapped Cache

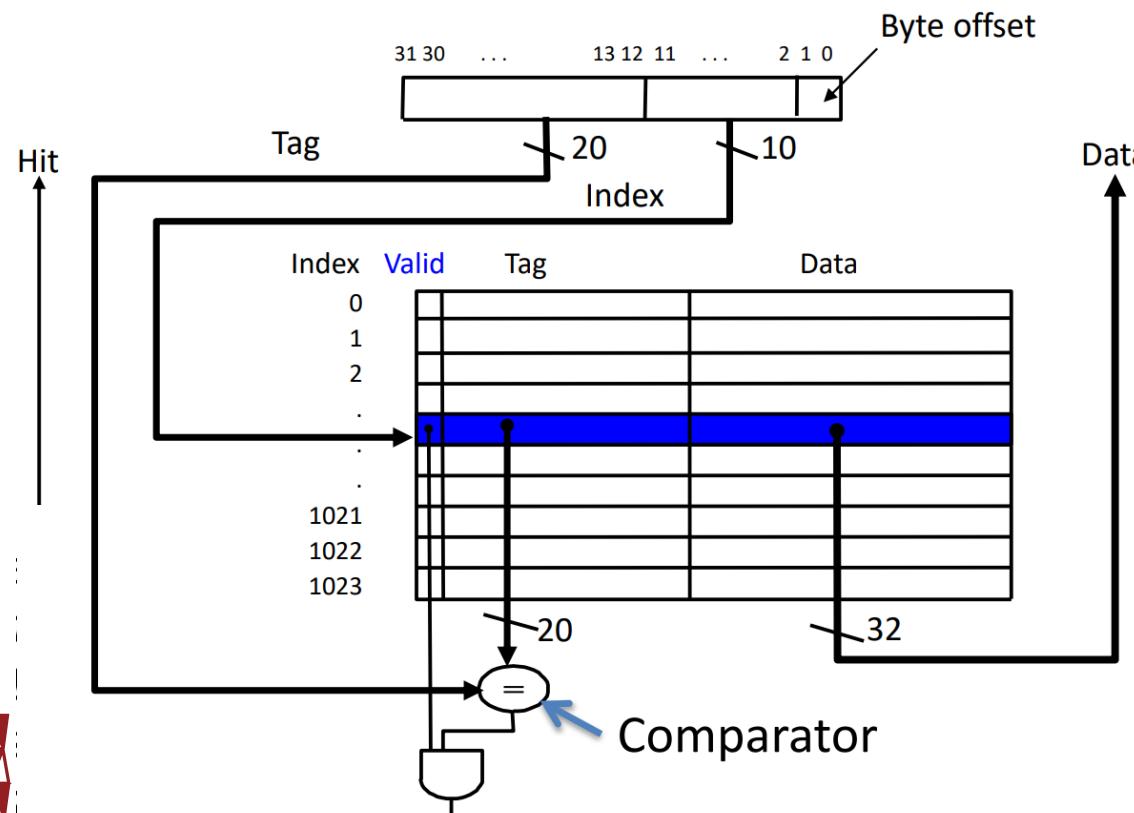


Four-Way Set-Associative Cache

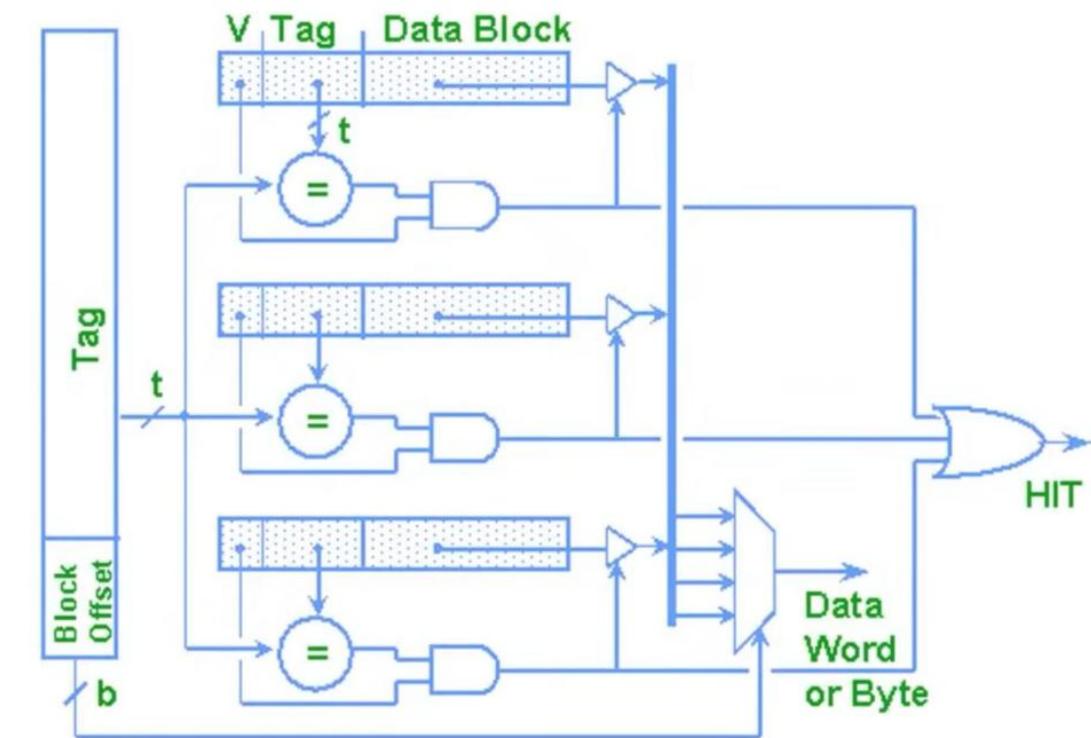
- $2^8 = 256$ sets each with four ways (each with one block)



Direct Mapped Cache



Fully Associative Cache



Real exam problem(1)



(a) We have an 10-bit address space and a n-way set associative cache. After a while, the entire cache has the following state:

Index	Tag 1	Valid 1	Tag 2	Valid 2
0b00	0b1011	1	0b1101	1
0b01	0b0011	1	0b0010	1
0b10	0b1110	1	0b0111	0
0b11	0b1111	0	0b0001	0

Solution: 4, 2, 4.

Calculate the bit width of tag, index, and block offset.

Tag	Index	Offset

Solution:

(b) Calculate the following parameters of the cache in (b):

Associativity: _____

Block size (in Bytes): _____

Cache size (in Bytes): _____

1. Associativity is 2
2. Block size is 16 Bytes;
3. Cache size is 128 Bytes;



立志成才报国裕民

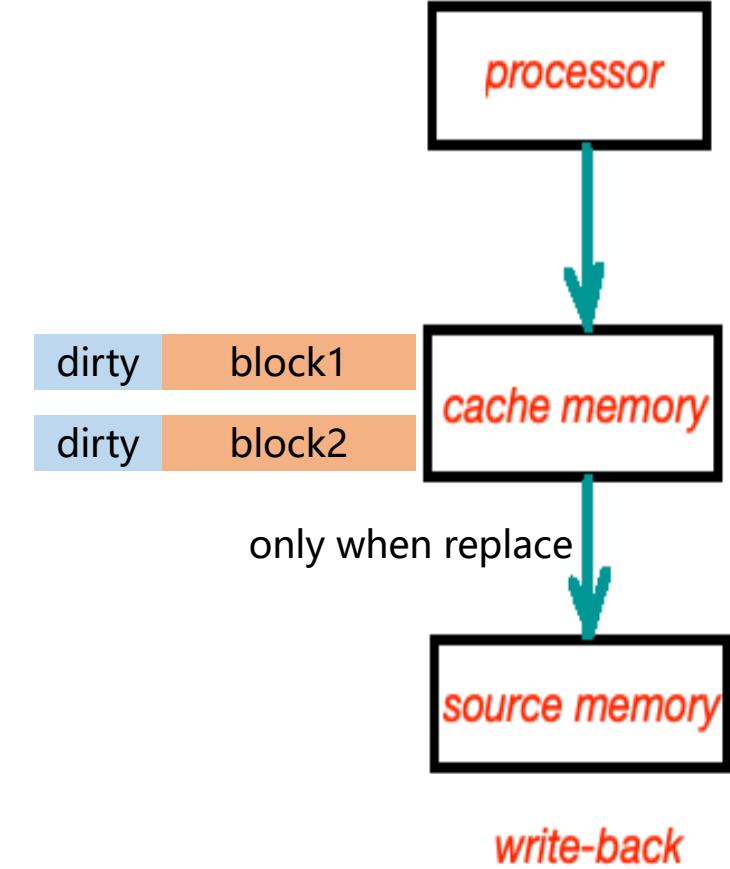
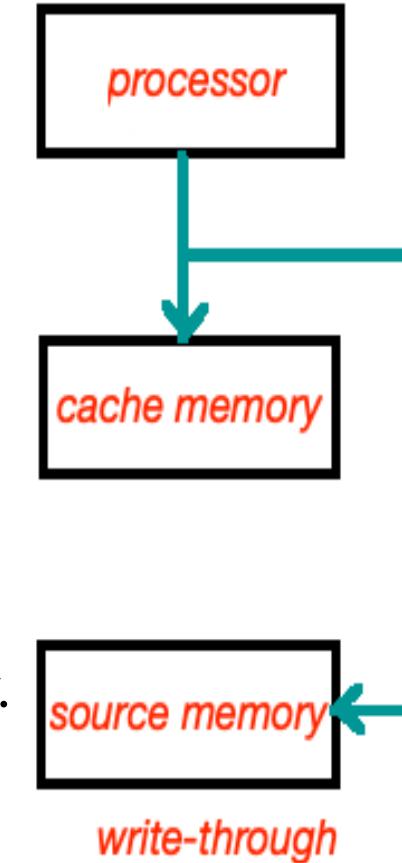


Write Through

- Write data to cache and next level.
- Cache data and memory is consistent.

Write Back

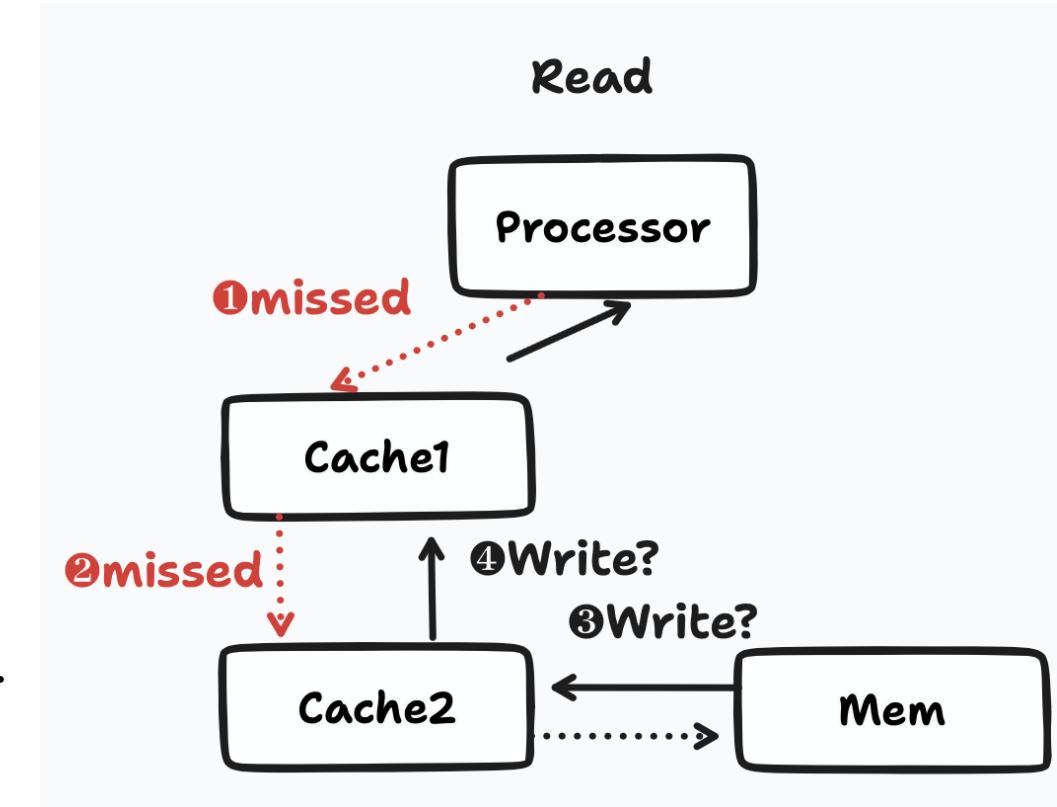
- Only write cache when CPU writes.
- Write memory when dirty block is replaced.
- Combining multiple writes to one for memory.





Write Allocation vs Write No Allocation

- Allocation will load data from next level first
- Then allocate a new Cache Block for missing date
- Allocation always combine with Write Back
- No allocation will bypass cache to write memory.
- No allocation always combine with Write Through.





Cache Replacement Strategies

LRU : Replace the cache block which is used least recently

FIFO : Replace the cache block which is loaded into cache earliest

Random : random select a cache block to replace

Cache Misses Type

- **Compulsory** : First access to block impossible to avoid
- **Capacity** : Cache cannot contain all blocks accessed by the program
 - Capacity misses are misses that will still occur even if the cache were fully associative with LRU replacement
- **Conflict** : Multiple memory locations mapped to the same cache location
 - Conflict misses may not occur if the cache were fully associative with LRU replacement





- **Hit rate** : fraction of accesses that hit in the cache
- **Miss rate** : 1- Hit rate
- **Hit time** : time to access cache.
- **Miss penalty** : time to replace a cache block from lower level in memory.
- **AMAT(Average Memory Access Time)**: the average time to access memory considering both hits and misses in the cache.
- $AMAT = Hit\ time + Miss\ rate \times Miss\ penalty$



Real exam problem(2)



(a) We have an 8-bit address space and a 2-way set associative cache with properties as follows:

1. Cache size is 32 Bytes;
2. Block size is 8 Bytes;

Calculate the bit width of tag, index, and offset bits.

TAG	Set Index	Block Offset

Solution: 4, 1, 3.

(b) We will access the data of addresses as follows. Fill in the blanks. It is about T/I/O (tag/index/offset, write down the value in decimal), classify the access as a Hit, Miss or Replace. (each line worth 1 pt.)

Address	T/I/O	Hit, Miss or Replace
0b00000100		
0b00000101		
0b01101000		
0b11001000		
0b01101000		
0b11011101		

Solution:

Address	T/I/O	Hit, Miss or Replace
0b00000100	0/0/4	Miss
0b00000101	0/0/5	Hit
0b01101000	6/1/0	Miss
0b11001000	12/1/0	Miss
0b01101000	6/1/0	Hit
0b11011101	13/1/5	Replace



Real exam problem(2)



(c) Assume we have a single-level, 1 KiB direct-mapped L1 cache, whose bit width of tag, index, and offset bits are 22, 6, 4 separately. An integer is 4 bytes. The array is block-aligned. Given the following C source code, what is the hit rate?

```
1 #define LEN 512
2
3 int array[LEN];
4 int main() {
5     for (int i = 0; i < LEN; i += 128) {
6         array[i] = 0;
7     }
8     for (int i = LEN - 128; i >= 0; i -= 128) {
9         array[i] = 0;
10    }
11    return 0;
12 }
```

Solution1: 1/4



Real exam problem(3)



(b) Suppose the cache has the following settings:

Cache levels	1
Block size	16 bytes
Number of sets	4
Cache size	128 bytes
Block replacement policy	LRU

1. Is this cache direct-mapped, set-associative, or fully-associative? If it is associative, also write down its associativity.

2. Suppose the memory addresses are 32-bit long, what is the length of the tag field?

Solution: set-associative, 2 way; 26.



Real exam problem(3)



(c) Suppose the following code is running on a system with the above cache, where `sizeof(int) == 4`.

```
1 #define array_size 64
2 #define repeat_times 1
3 #define step_size 2
4
5 int main() {
6     int array[array_size] = { };
7
8     for (int r = 0; r < repeat_times; r++) {
9         for (int i = 0; i < array_size; i += step_size) {
10             array[i] = array[i] + 2333;
11         }
12     }
13
14     return 0;
15 }
```

1. What is the total number of accesses to the cache?

2. What is the hit rate?

3. Which type(s) of miss occur(s)?

4. Suppose `repeat_times` goes to infinity (only for this question), what number will the hit rate converge to?

5. If `repeat_times` is changed to 2 (only for this question), try to swap two lines of the above code to maximize the hit rate without disturbing the results. Which two lines will you choose and what is the maximized hit rate?

6. For the modified code in the previous question, suppose again `repeat_times` goes to infinity (only for this question), what number will the hit rate converge to?

Solution1: 64

Solution2: 3/4

Solution3: Compulsory

Solution4: 3/4

Solution5: 8, 9

Solution 6: 1

