ECE 2300
Digital Logic & Computer Organization

Spring 2025

More Caches

Cornell University

Announcements

- **Prelim 2 stats**
  - **Mean: 50.8 (out of 64); Median: 51.5; High: 64**
  - **We will go over the final letter grading scheme in the next lecture**
- **HW 7 will be released today**
  - **(part of) last question ties to next lecture**
- **Lab 3 report due tomorrow**
- **Instructor OH today (4/17) is cancelled**

Cache Basics (True or False)

- Cache is usually implemented using SRAM
- Memory block address is *not longer* than the memory address
- In a direct mapped (DM) cache, a memory block can be mapped to different cache blocks

Exercise: DM Cache Organization

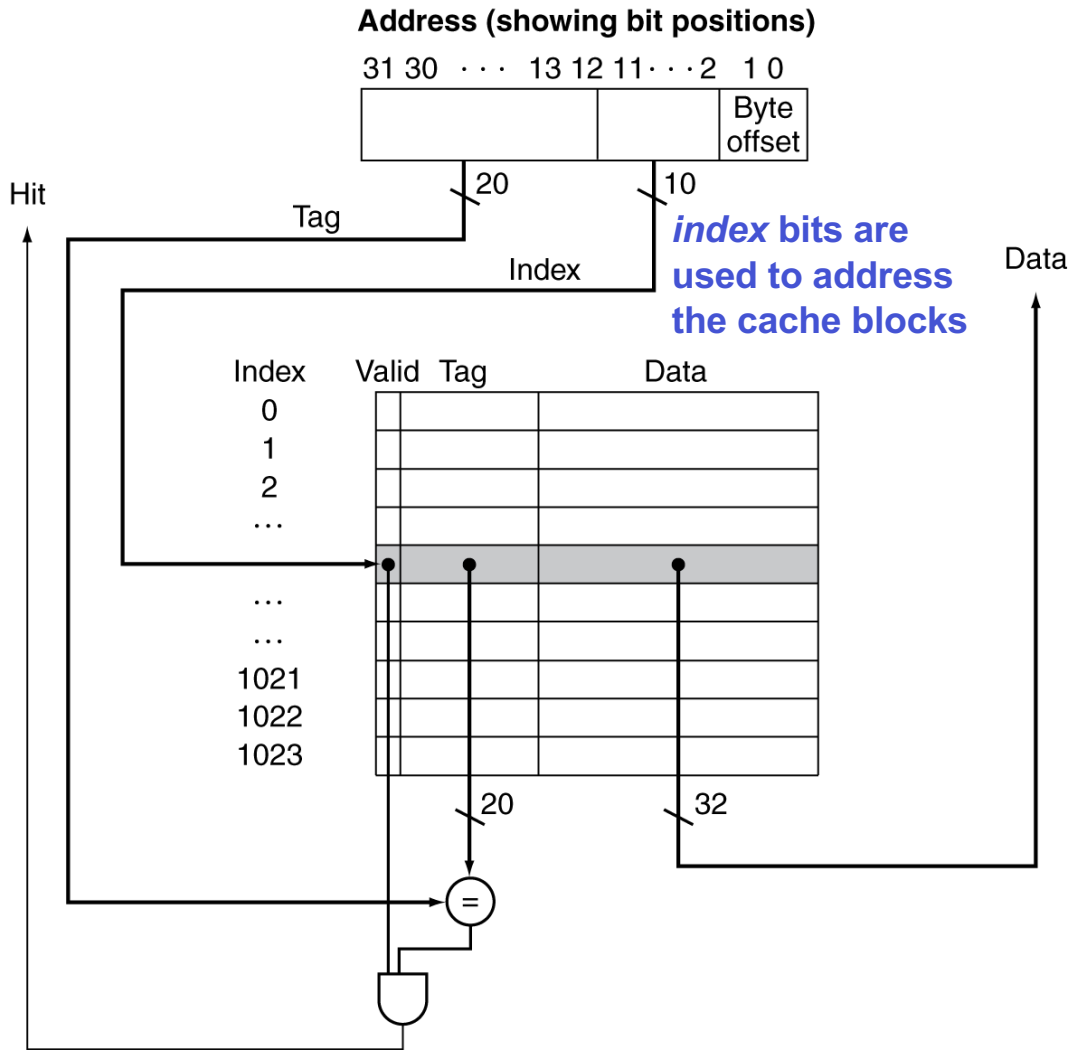
Example

32-bit memory address

Cache holds 1024 blocks

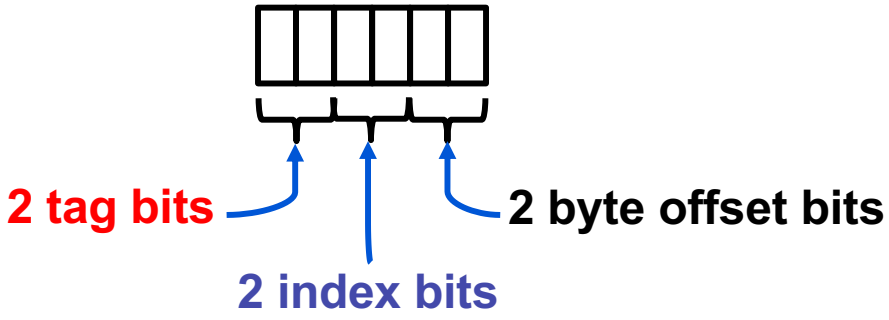
- **Each block holds 4 bytes**
- **Each cache block is associated with a tag and valid bit**

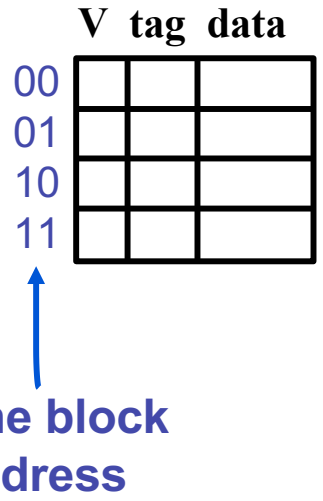
How many different memory blocks can map to the same cache block?



Review: DM Cache Example

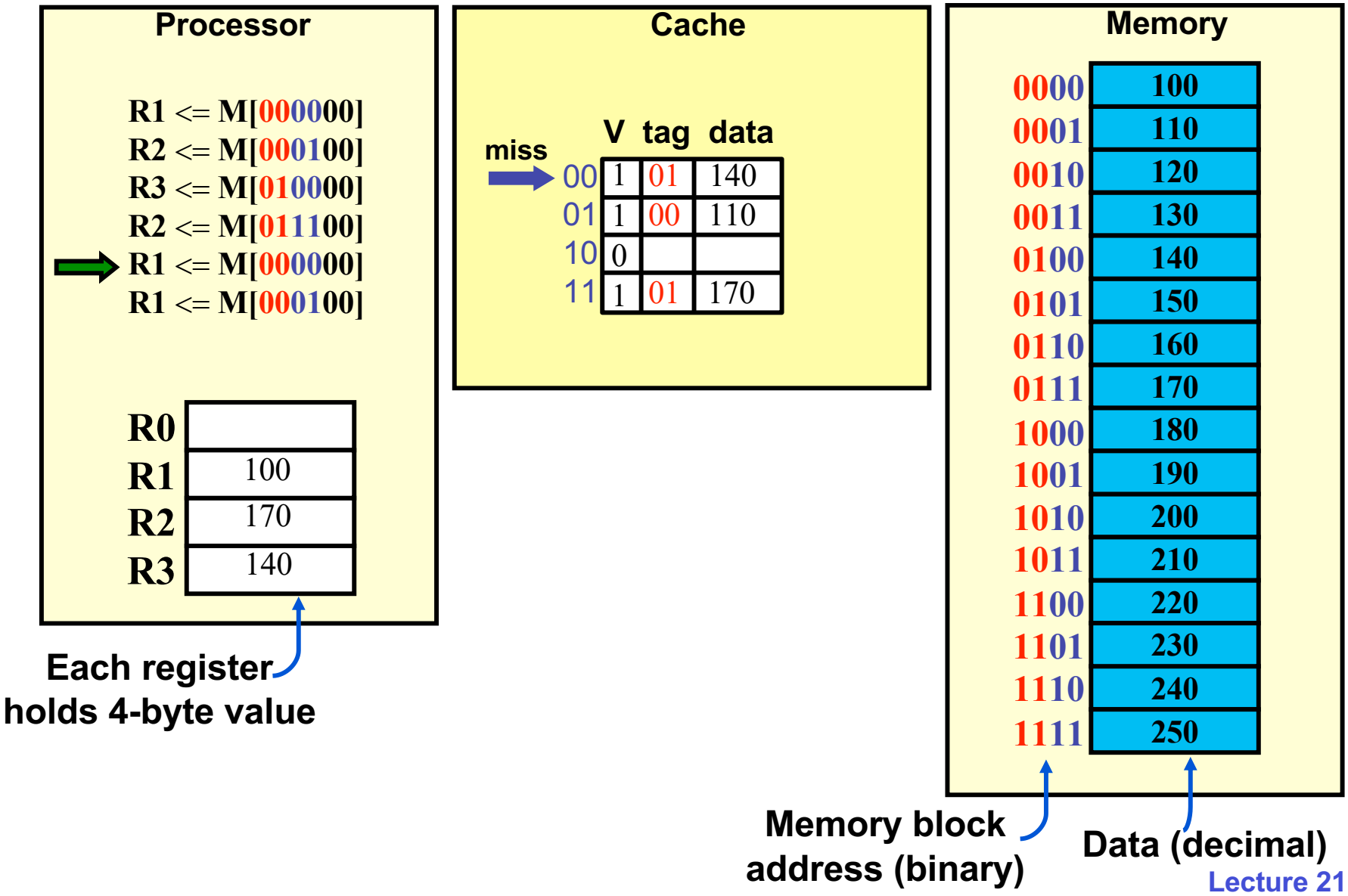
- Size of each block is 4 bytes
- Cache holds 4 blocks
- Memory holds 16 blocks
- Memory address has 6 bits



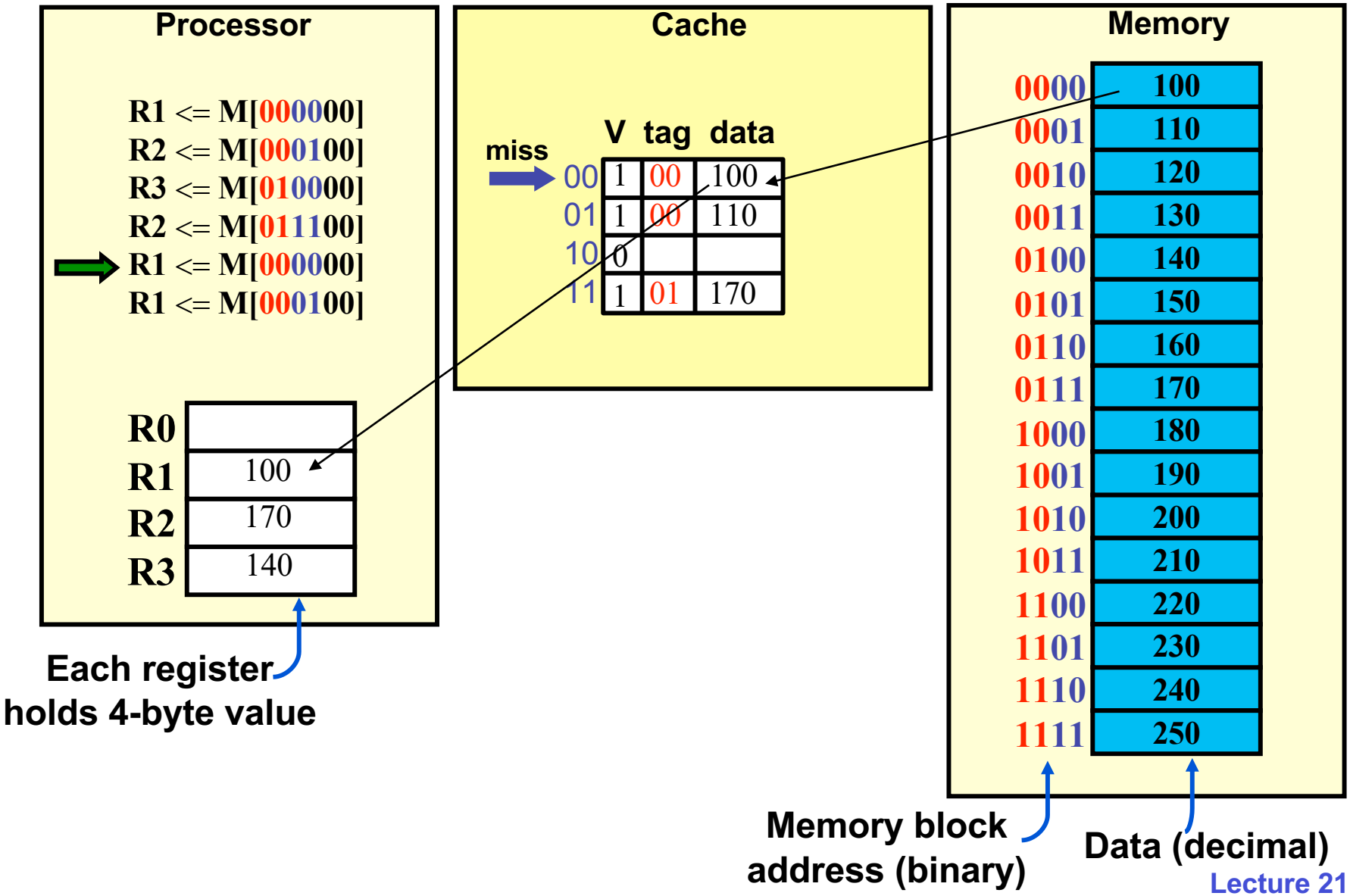


{ tag, index } = memory block address

Review: DM Cache Example

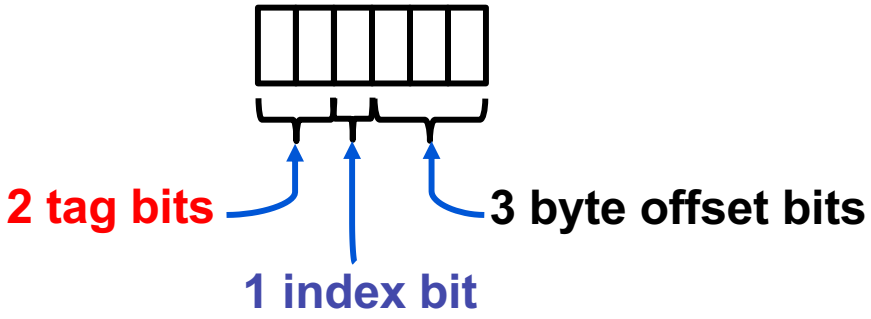


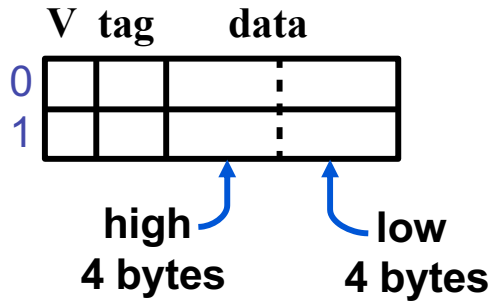
Review: DM Cache Example



Doubling the Block Size

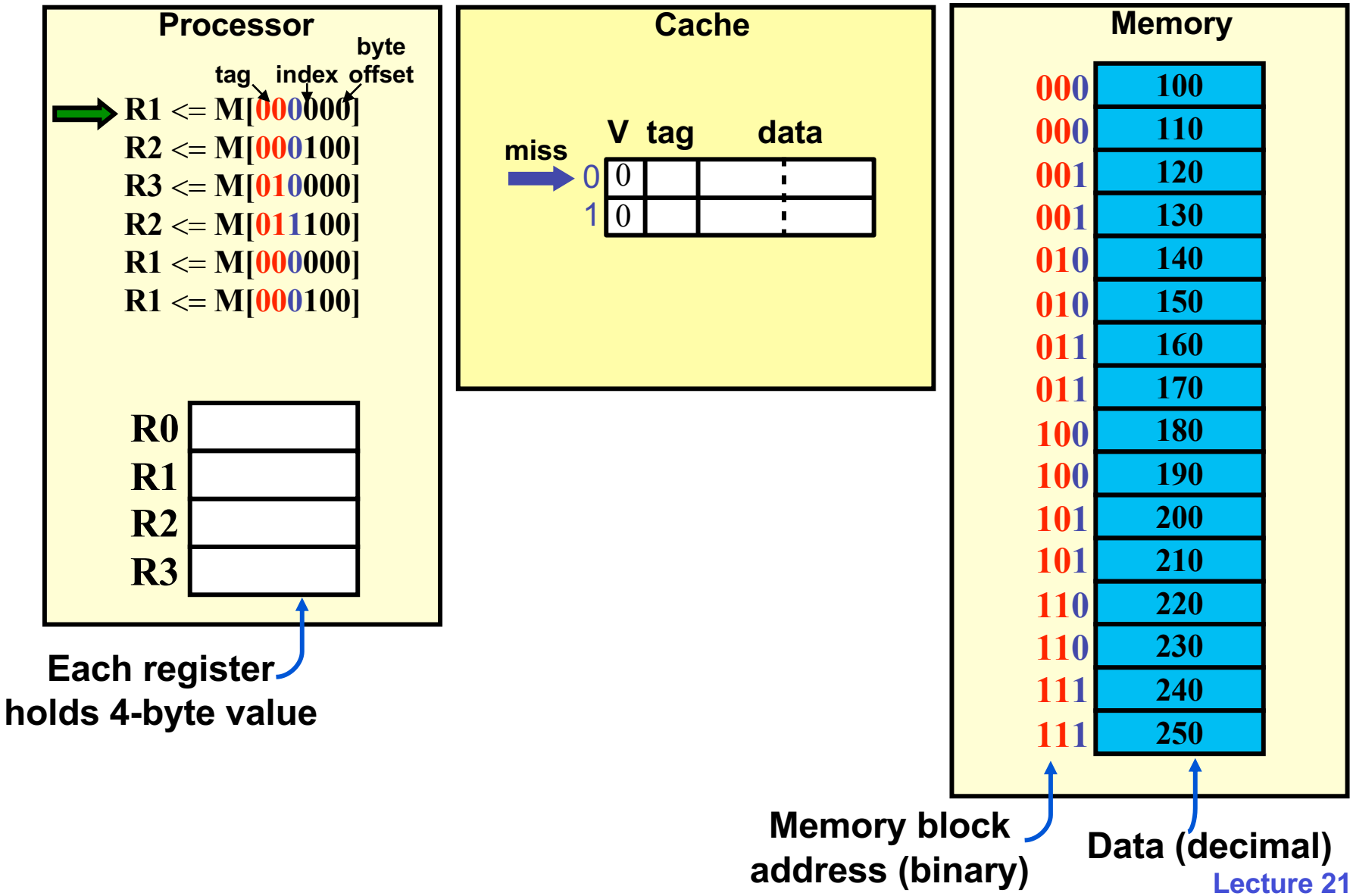
- Size of each block is 8 bytes
- Cache holds 2 blocks
- Memory holds 8 blocks
- Memory address has 6 bits



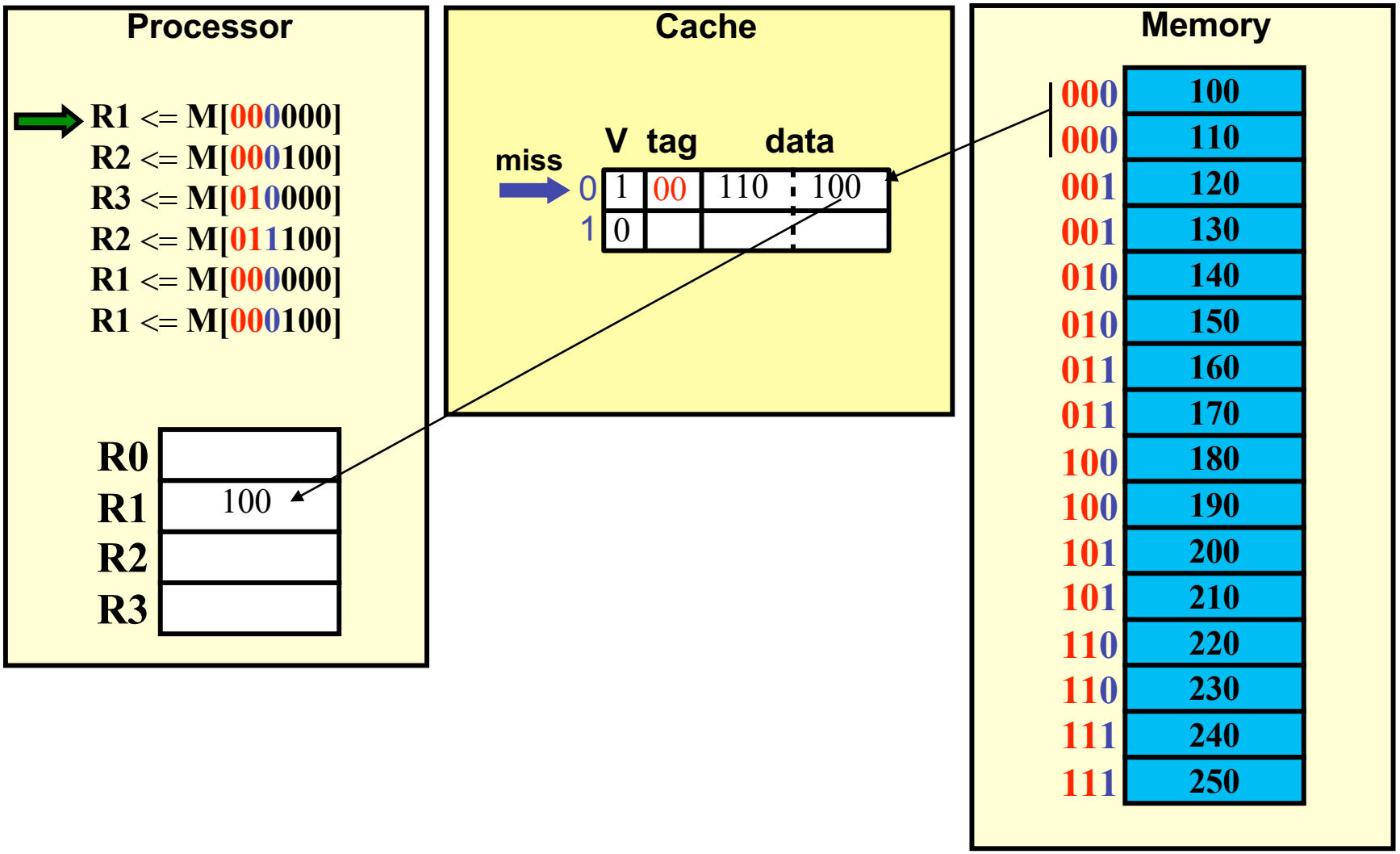


{ tag, index } = memory block address

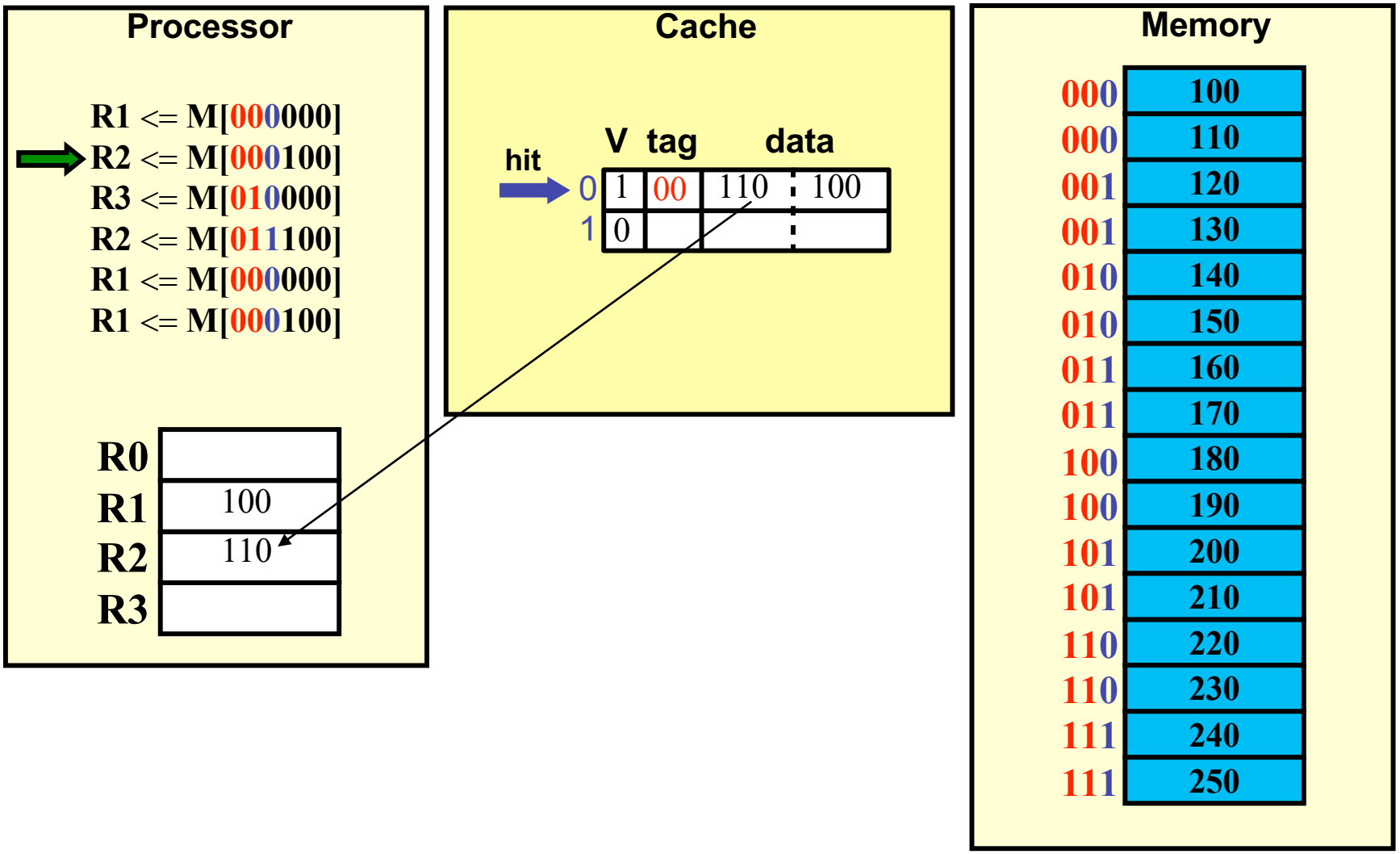
Doubling the Block Size



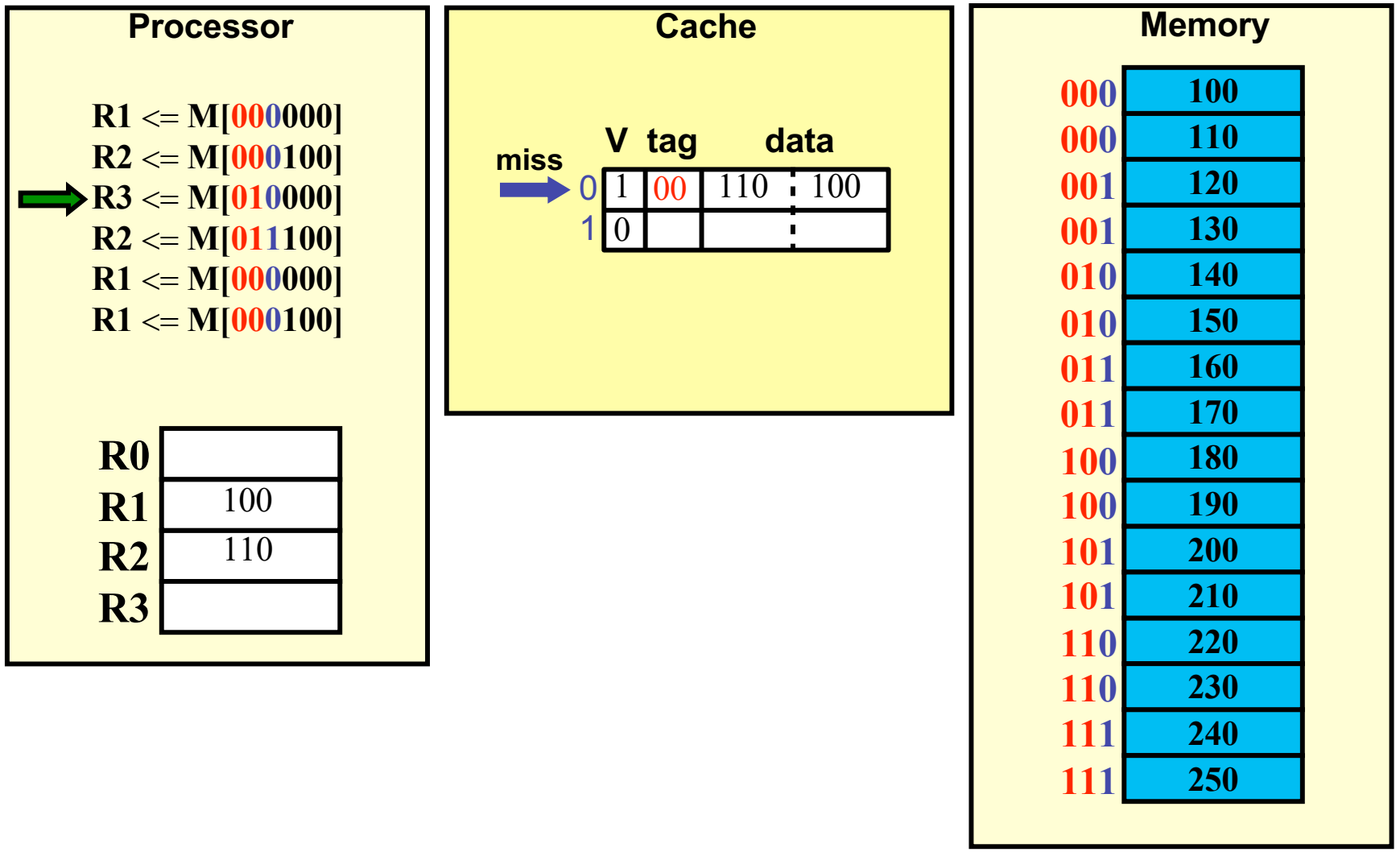
Doubling the Block Size



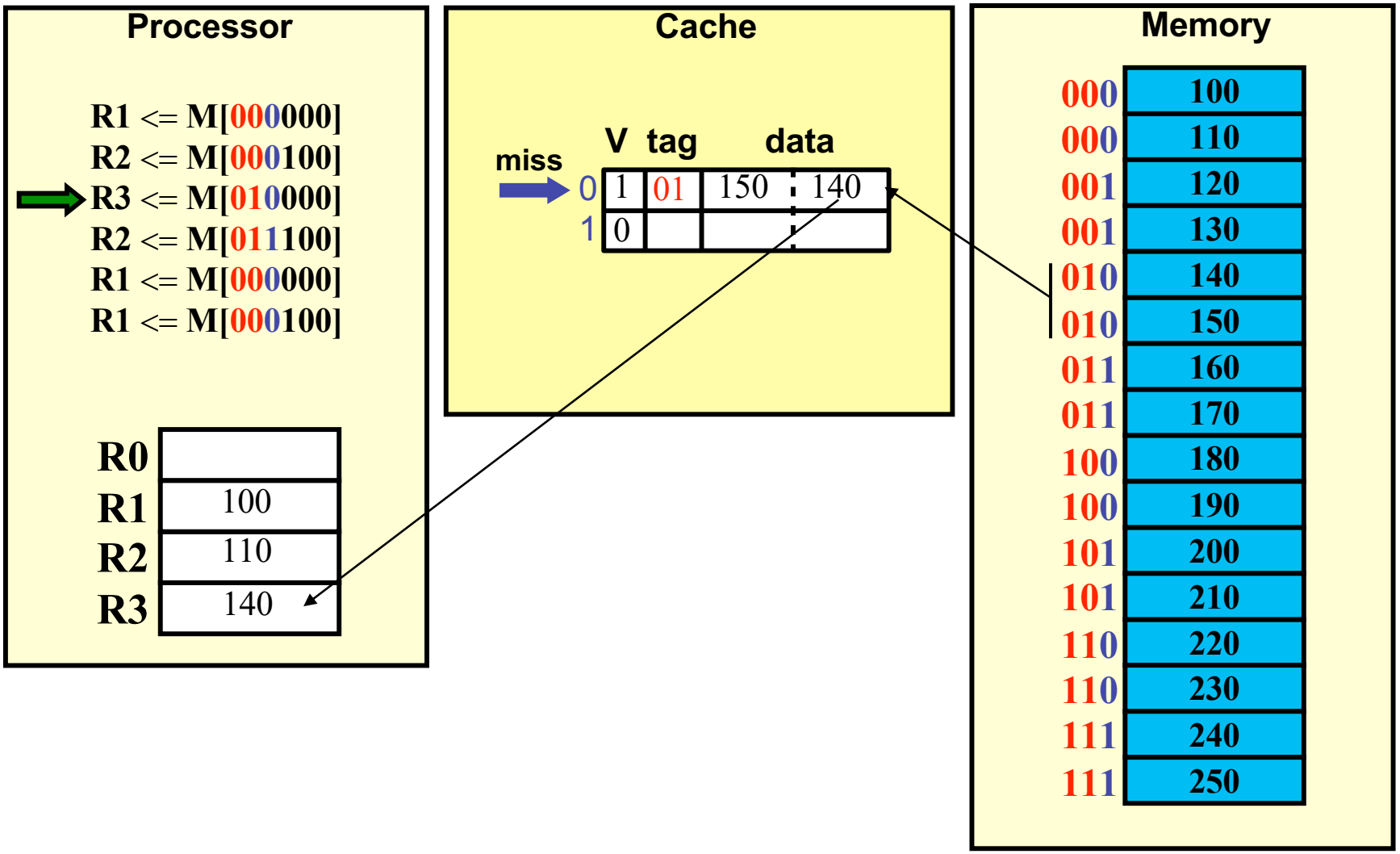
Doubling the Block Size



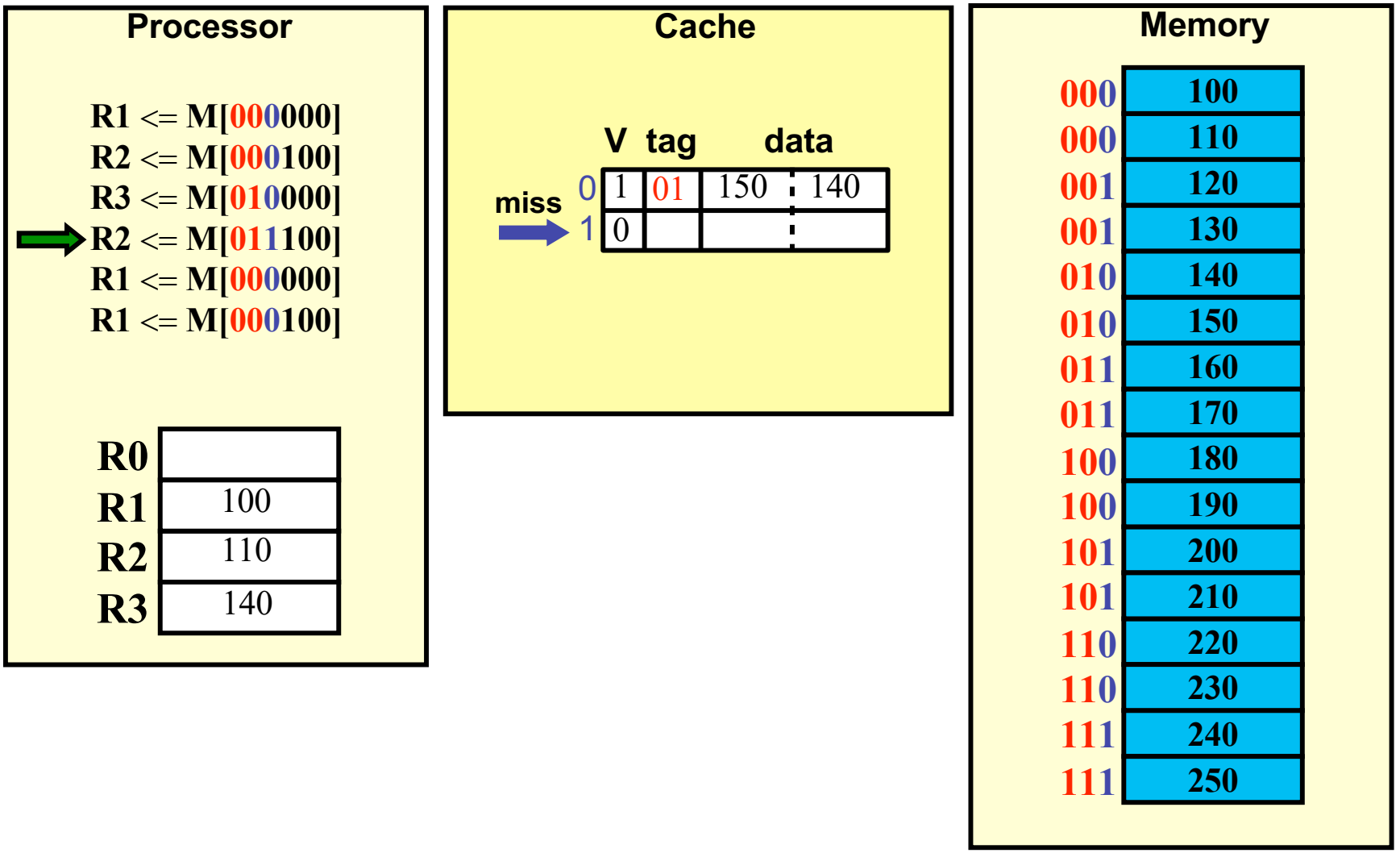
Doubling the Block Size



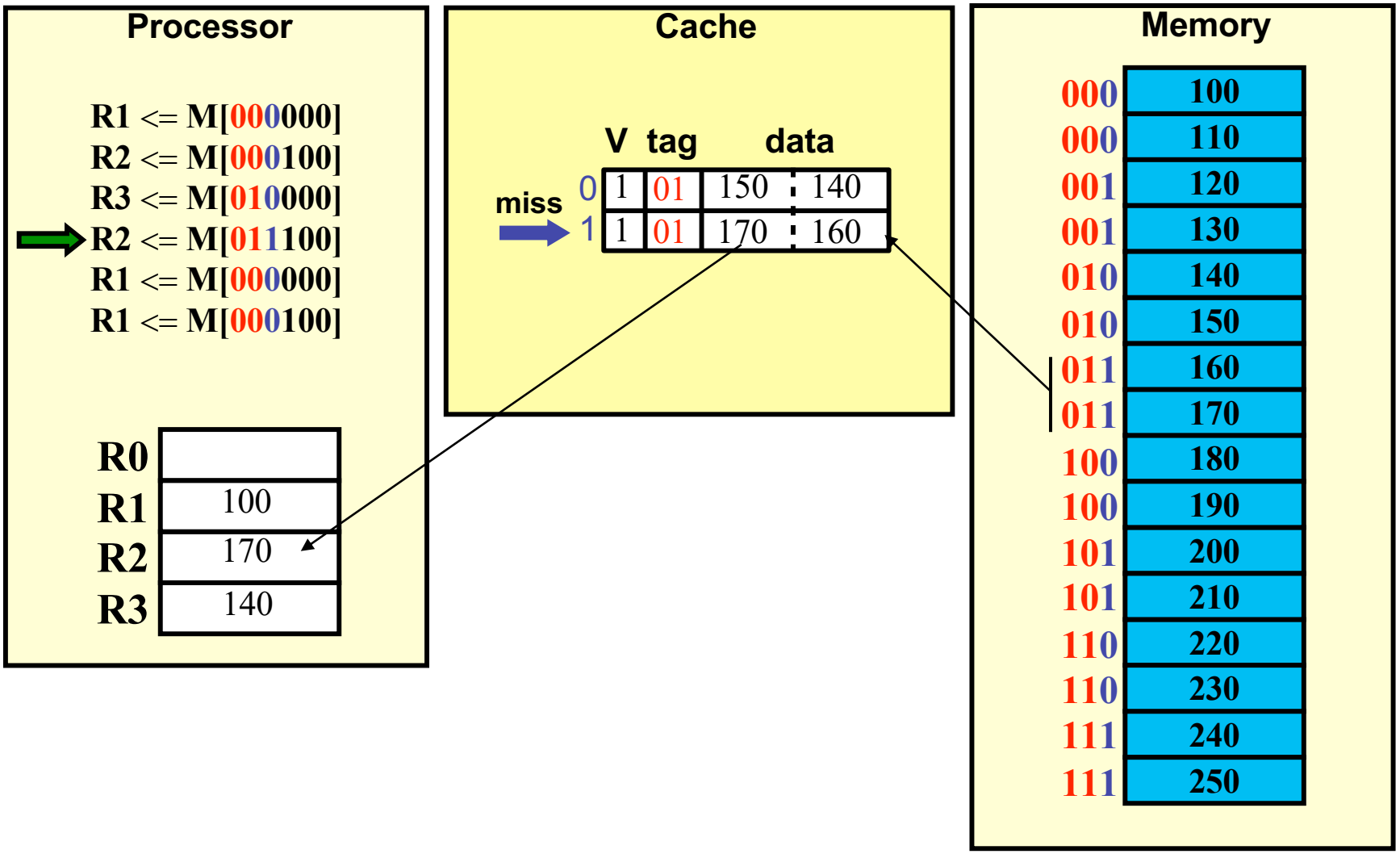
Doubling the Block Size



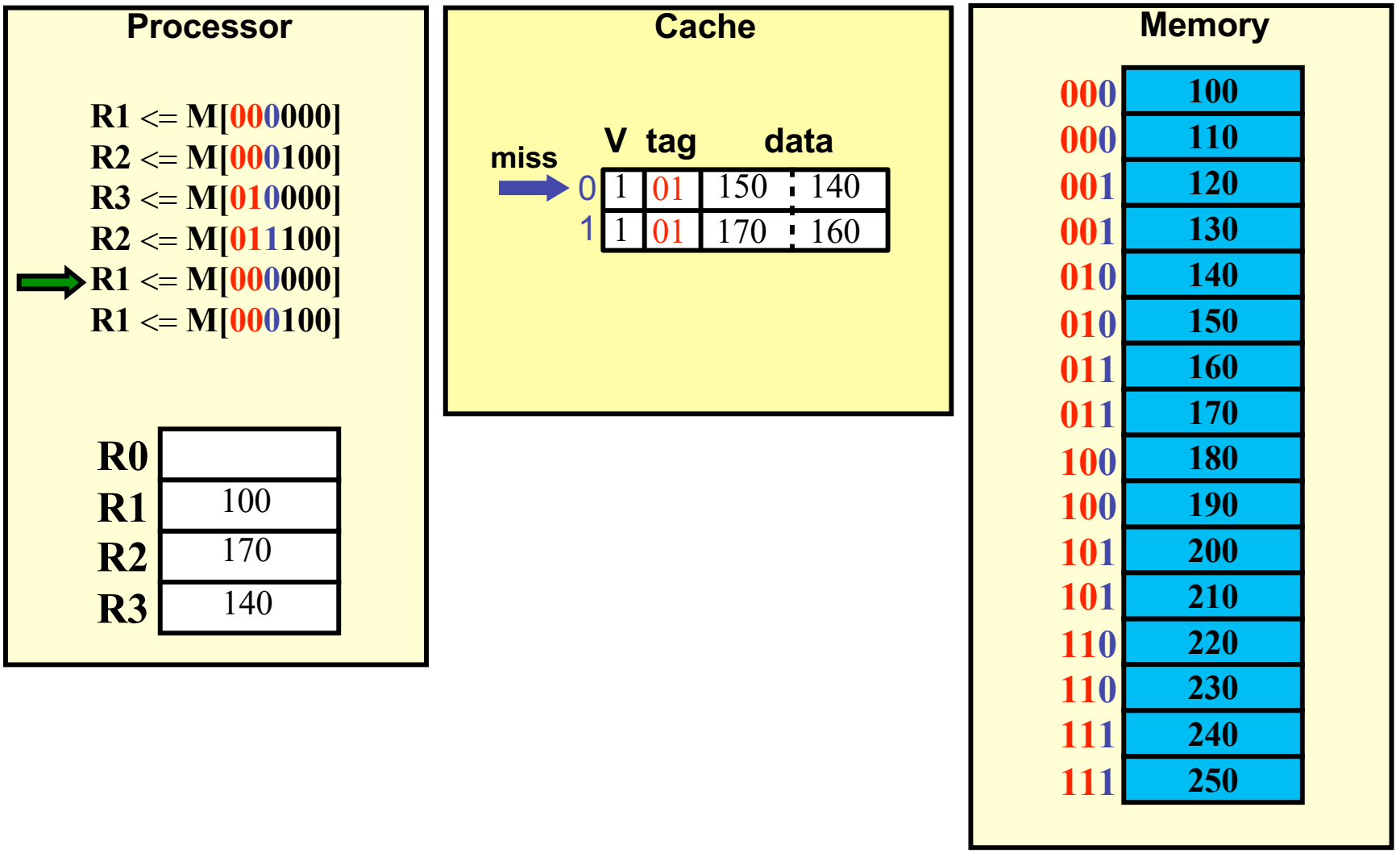
Doubling the Block Size



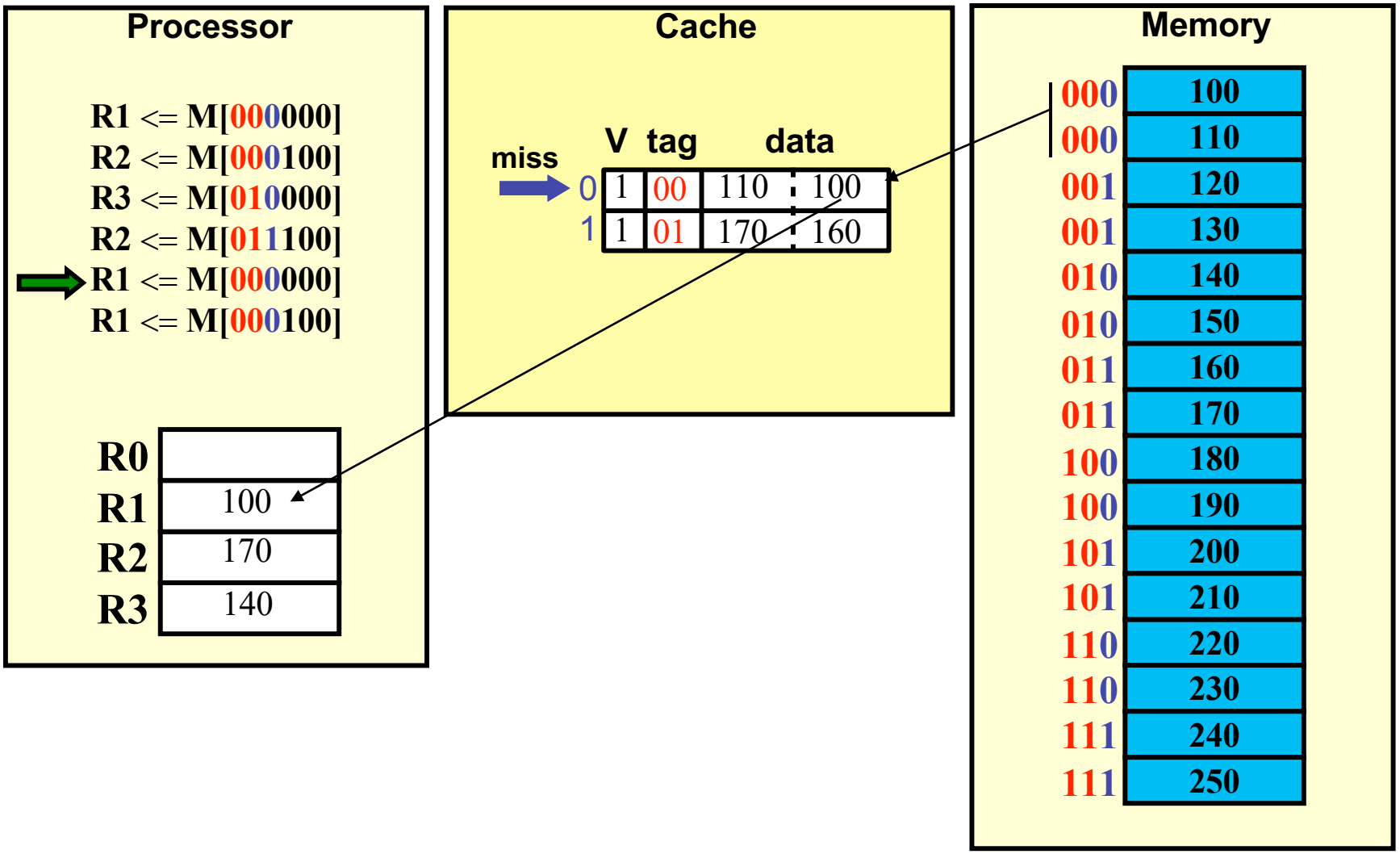
Doubling the Block Size



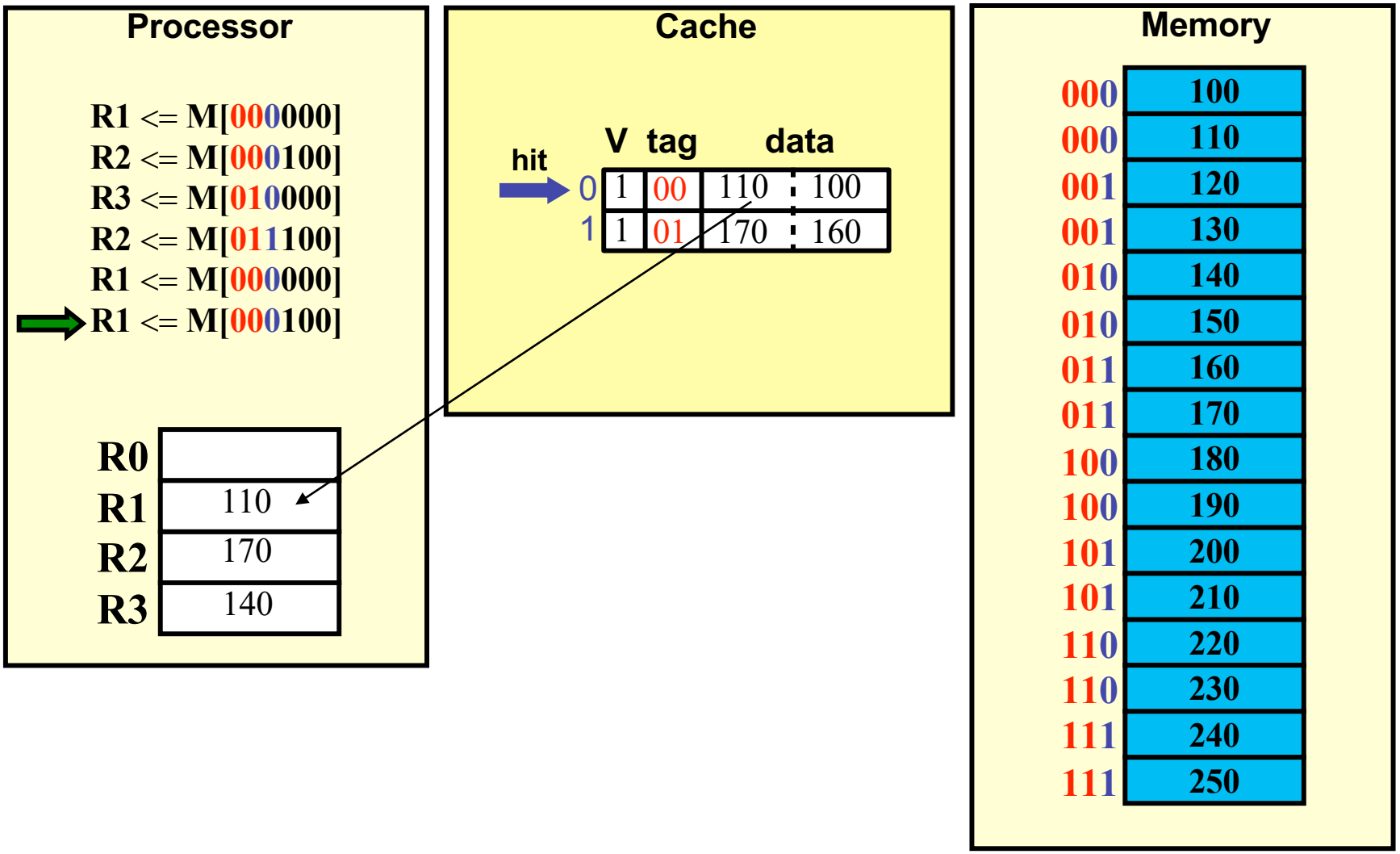
Doubling the Block Size



Doubling the Block Size



Doubling the Block Size



Block Size Considerations

- Larger blocks may reduce miss rate due to spatial locality

- But in a fixed-sized cache
  - Larger blocks => fewer of them => increased miss rate due to conflicts
  - Larger blocks => data fetched along with the requested data may not be used

- Larger blocks increase the miss penalty
  - Takes longer to transfer a larger block from memory

Exercise: DM Cache Address Breakdown

- Assuming 16-bit memory addresses, how many bits are associated with the tag, index, and offset of the following configuration for a direct mapped cache?

- 16 blocks, 4 bytes per block

  Byte offset: ? bits

  Index: ? bits

  Tag: ? bits

Exercise: DM Cache Address Breakdown

- Assuming 16-bit memory addresses, how many bits are associated with the tag, index, and offset of the following configuration for a direct mapped cache?

- 16 blocks, 4 bytes per block

  Byte offset: 2 bits

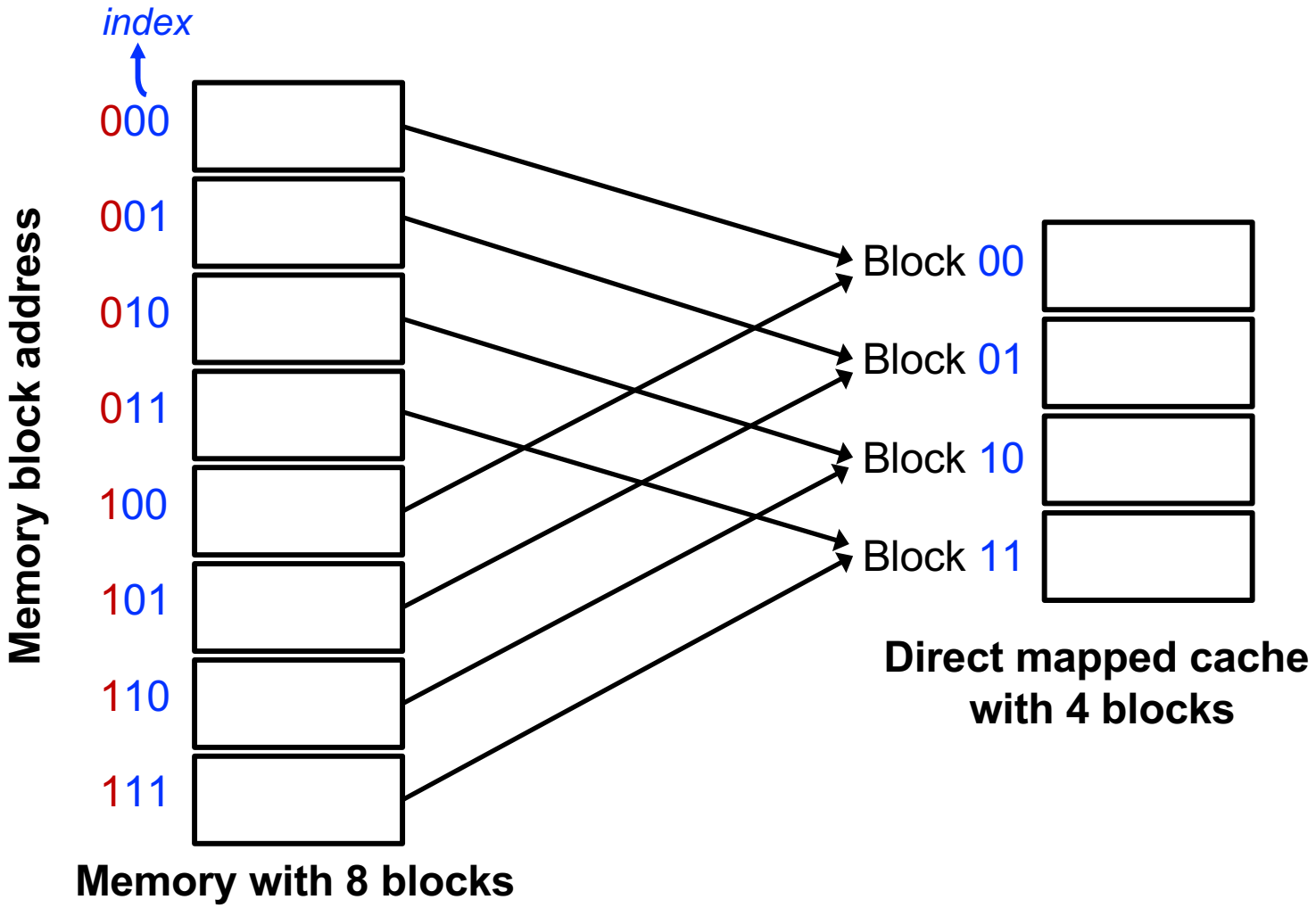
  Index: 4 bits

  Tag: 10 bits

Cache Intuition Revisited

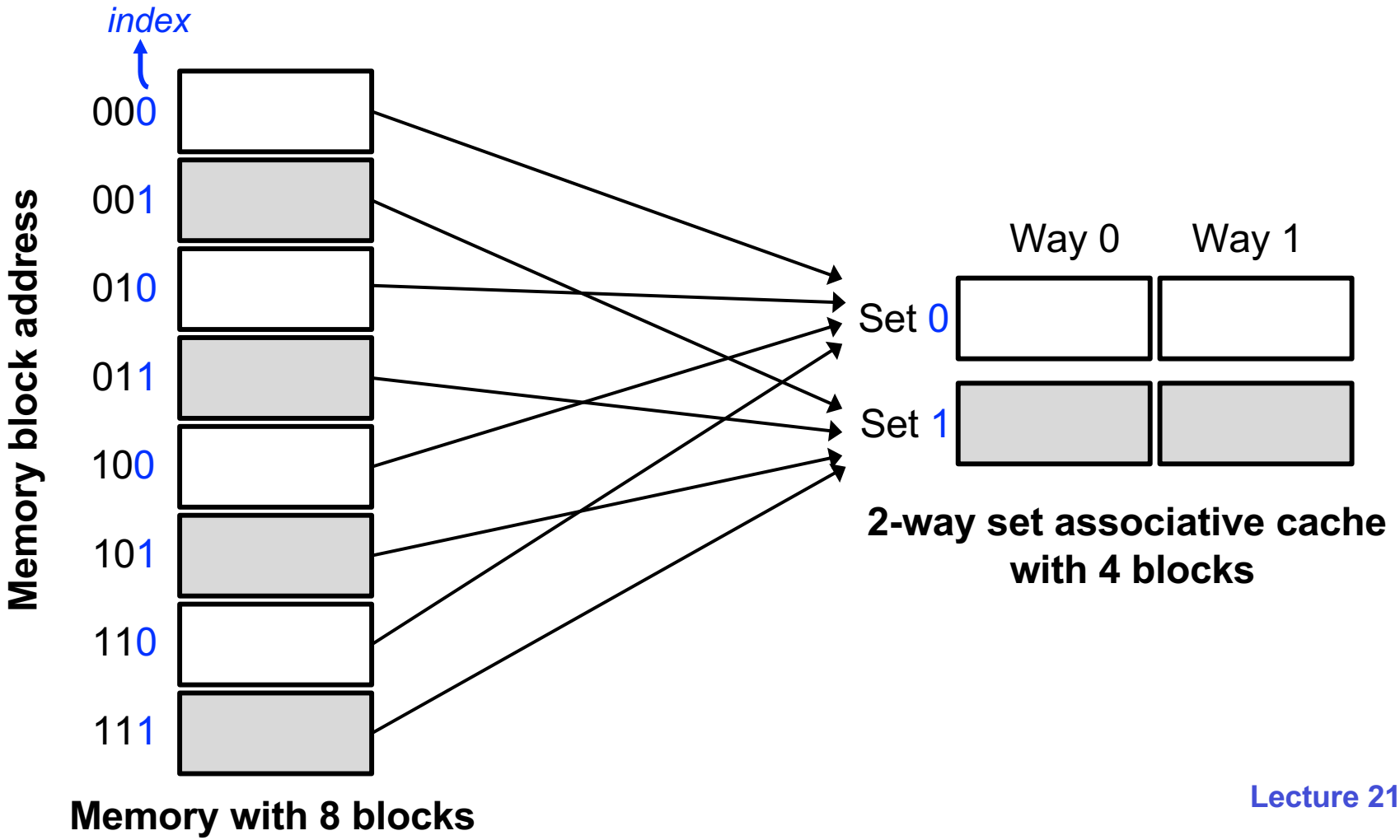
Block Placement in DM Cache

- **Direct mapped cache: Each memory block maps to one cache block**



More Flexible Block Placement

- **K-way Set Associate Cache: each memory block maps to one set, which contains *K* blocks (ways)**
  - **A memory block can be stored anywhere in the cache set**



Associative Caches

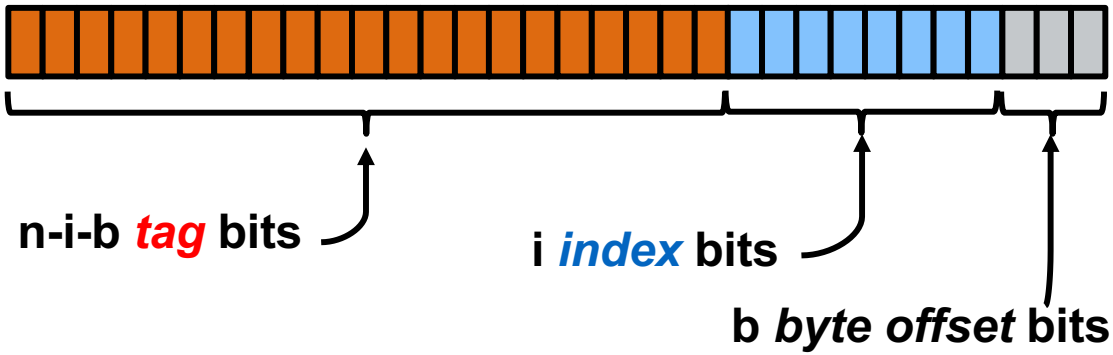
- Provide more flexible placement of blocks

- *K*-way set associative
  - Index bits determine which set to address
  - Each set contains *K* entries (ways)
  - All ways in the selected set are searched in parallel
    - *K* comparators (more expensive than direct mapped)

- An extreme case: Fully associative
  - Block can go in any cache location
    - Only one set => No need for index bits
  - All entries are searched in parallel
    - Comparator per entry (most expensive)

Address Translation for Associative Caches

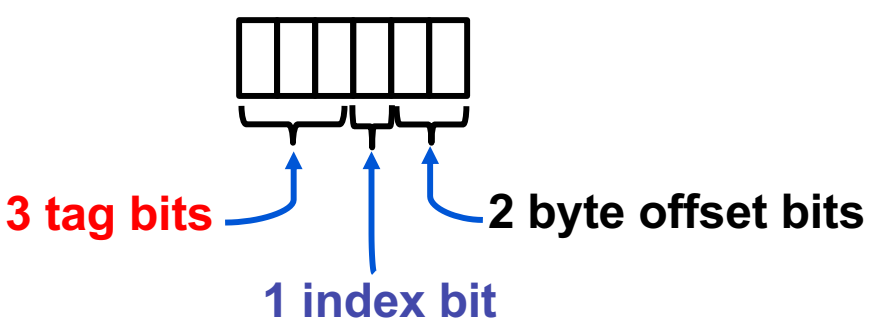
- **Breakdown of memory address for cache use**

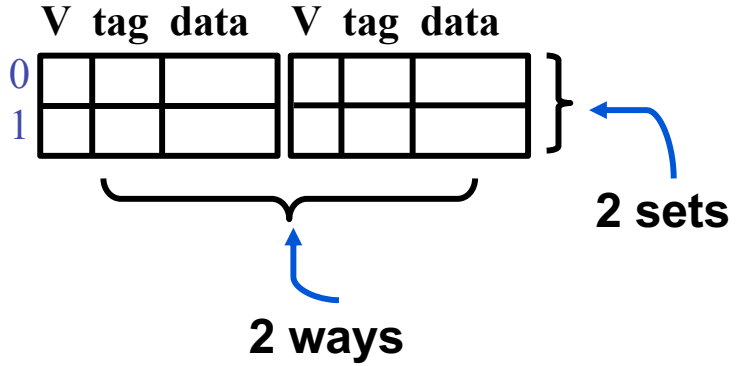


- **Parameters for a K-way set associative cache**
  - Number of sets is 2^i
  - Number of blocks is $K \times 2^i$
  - Size of each cache block is 2^b bytes
  - Total cache size is $(K \times 2^{i+b})$ bytes

2-way Set Associative Example

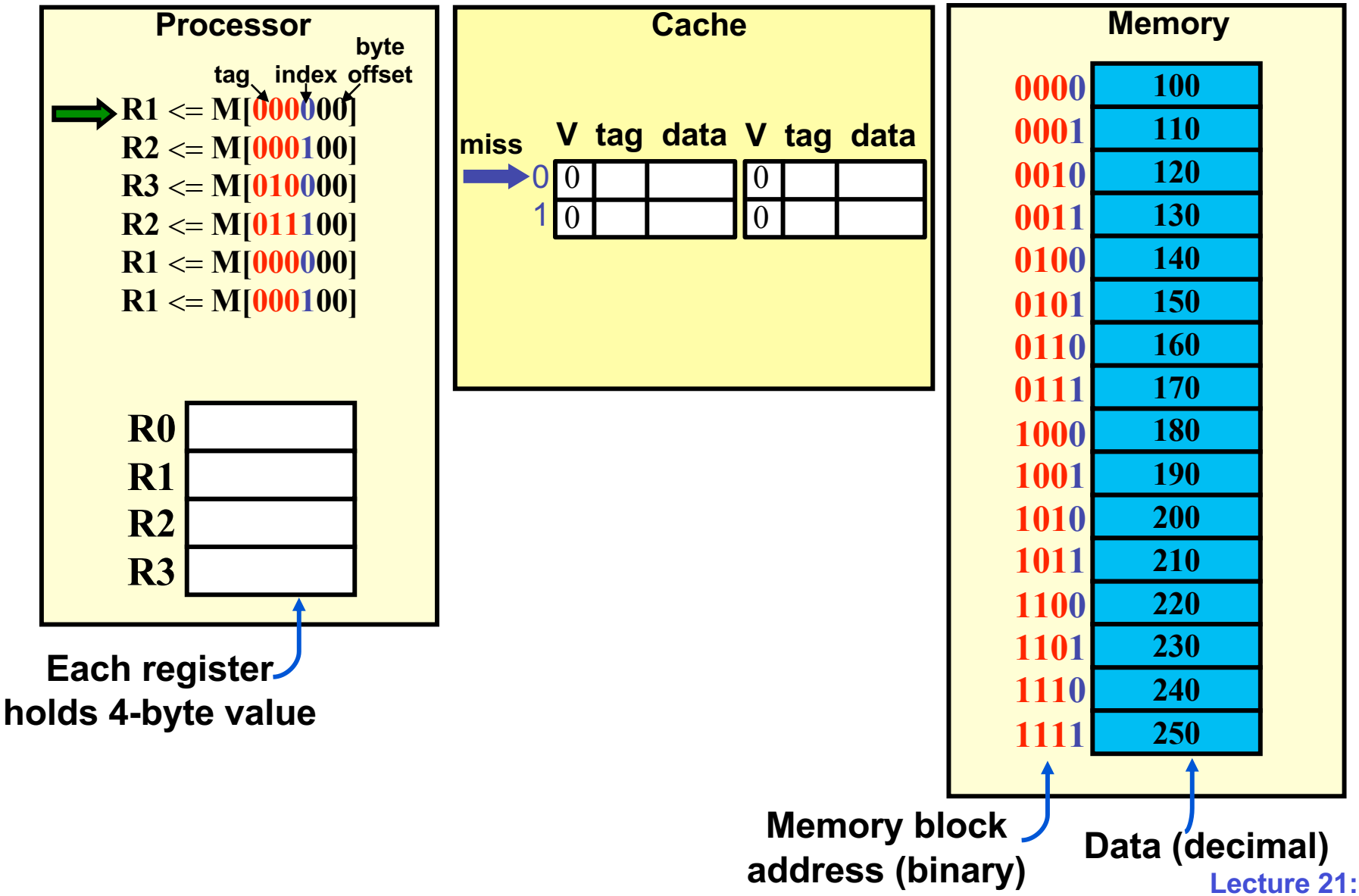
- Size of each block is 4 bytes
- Cache holds 4 blocks, 2-way set associative
- Memory holds 16 blocks
- Memory address



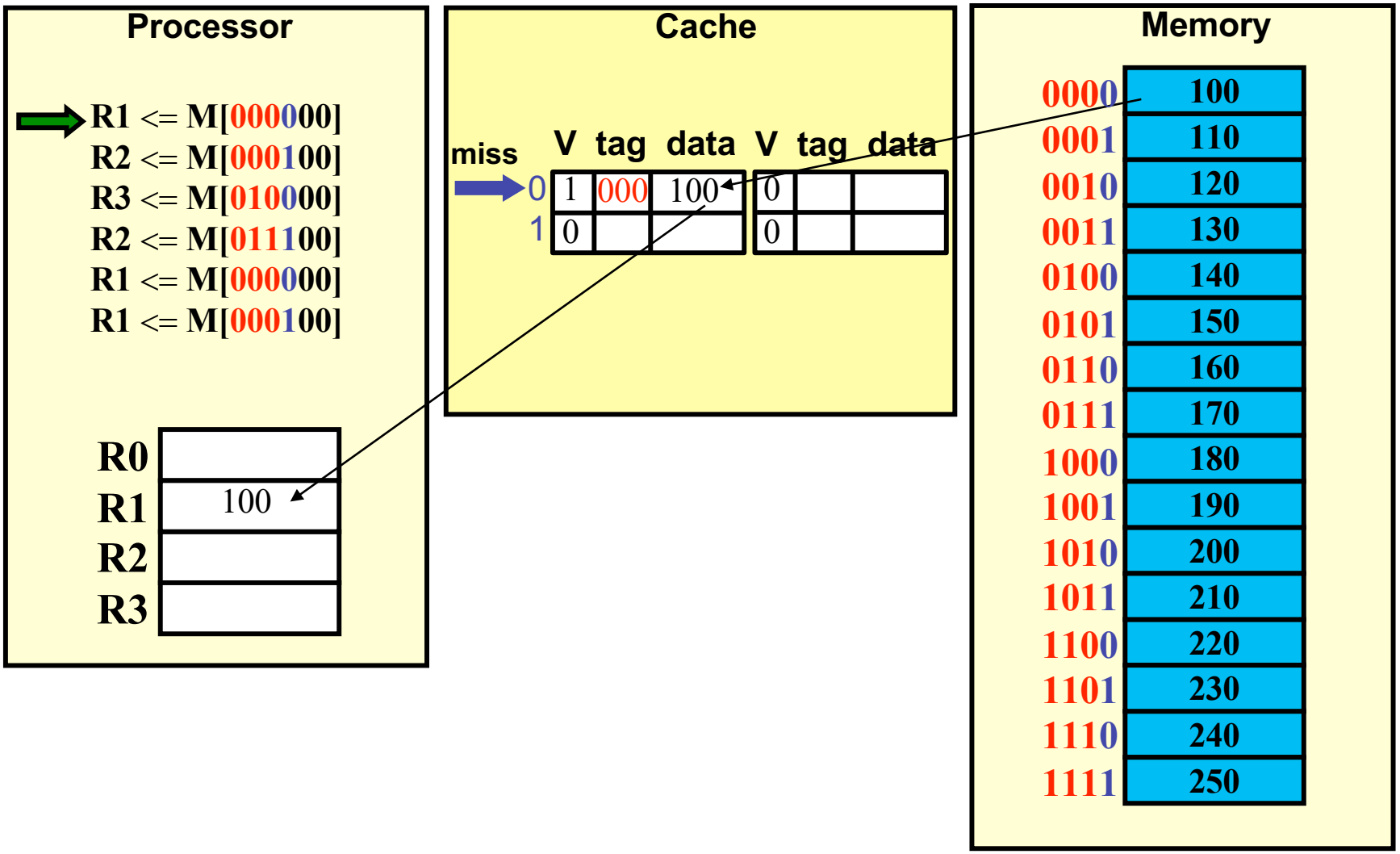


{ tag, index } = memory block address

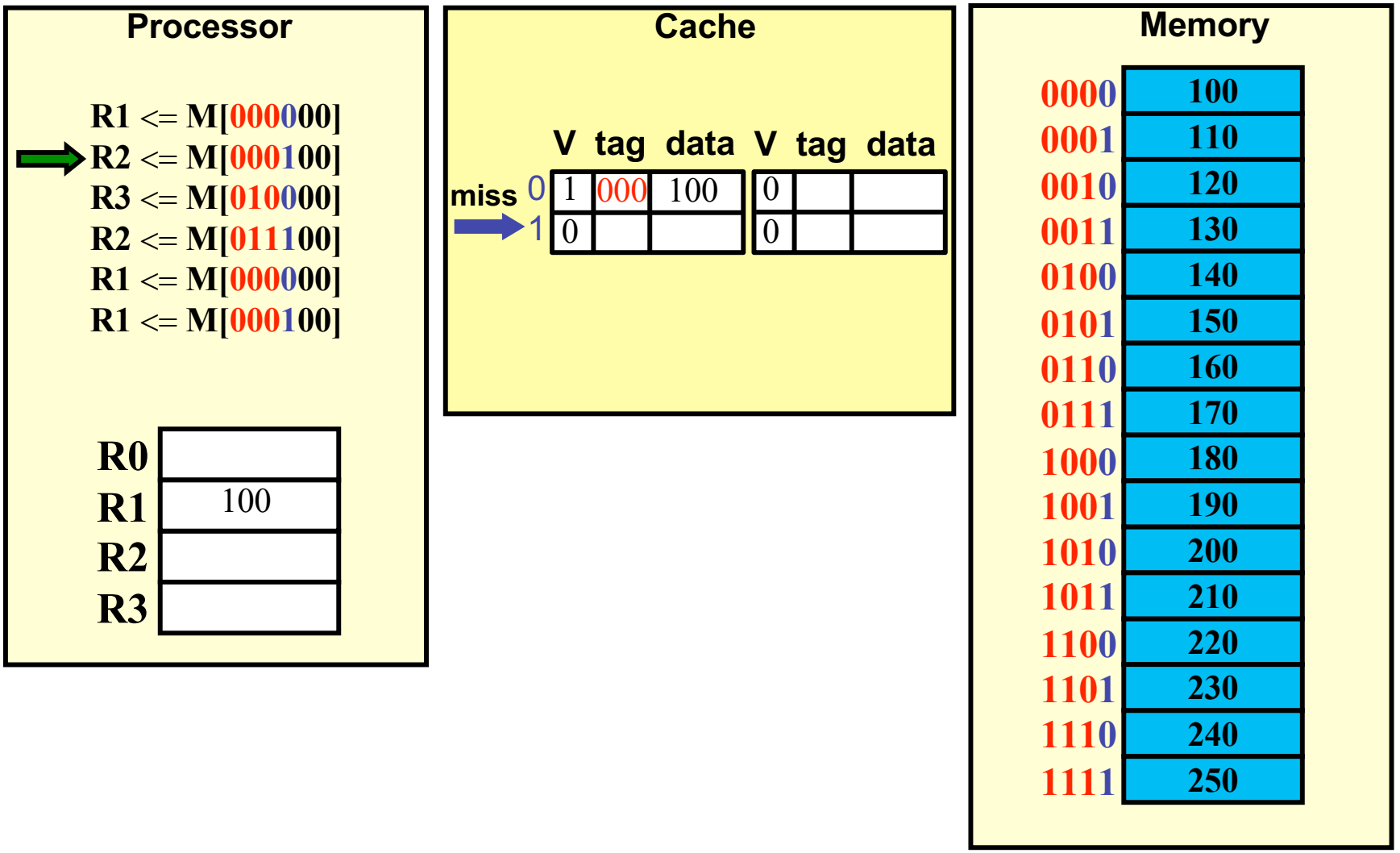
2-way Set Associative Example



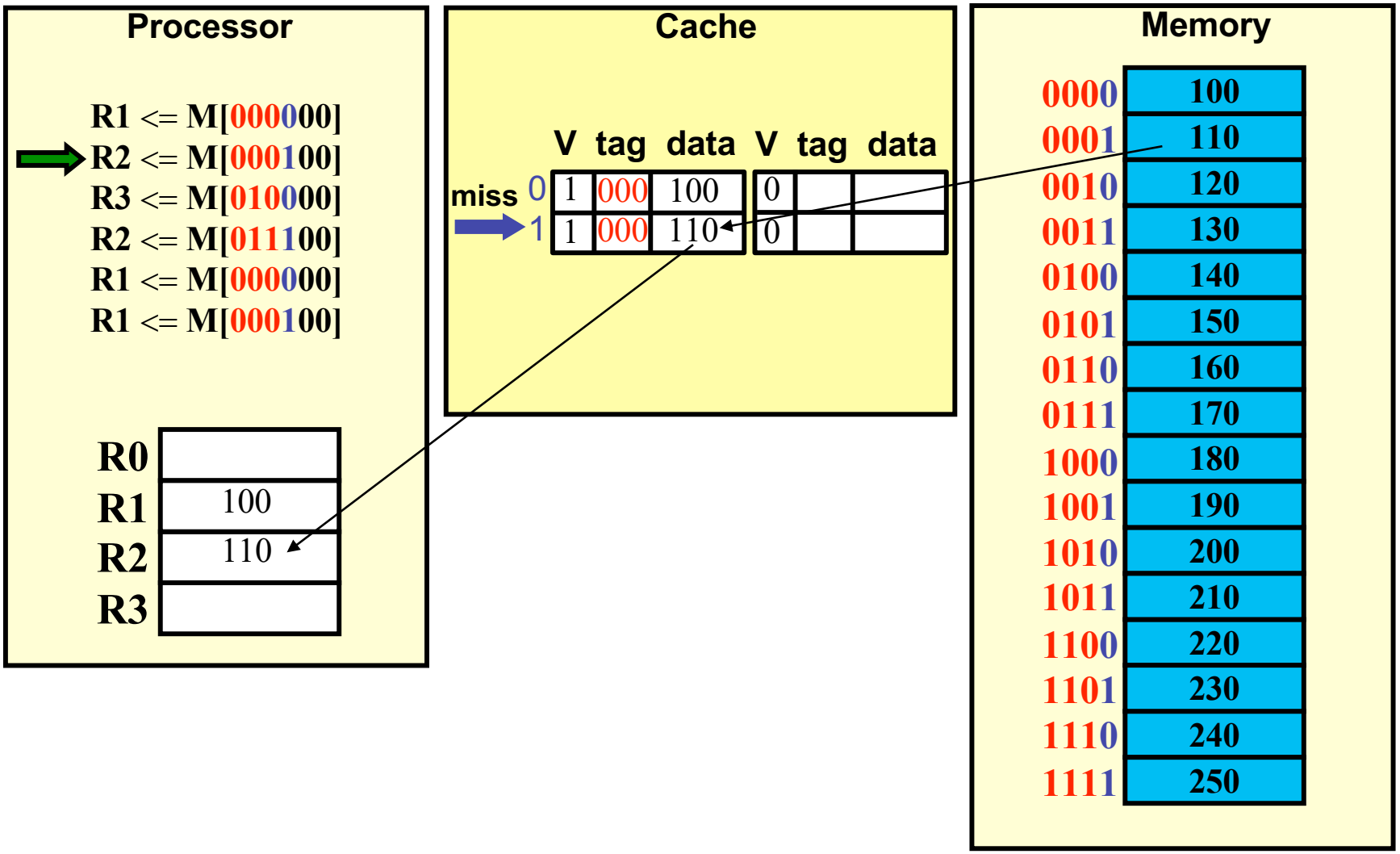
2-way Set Associative Example



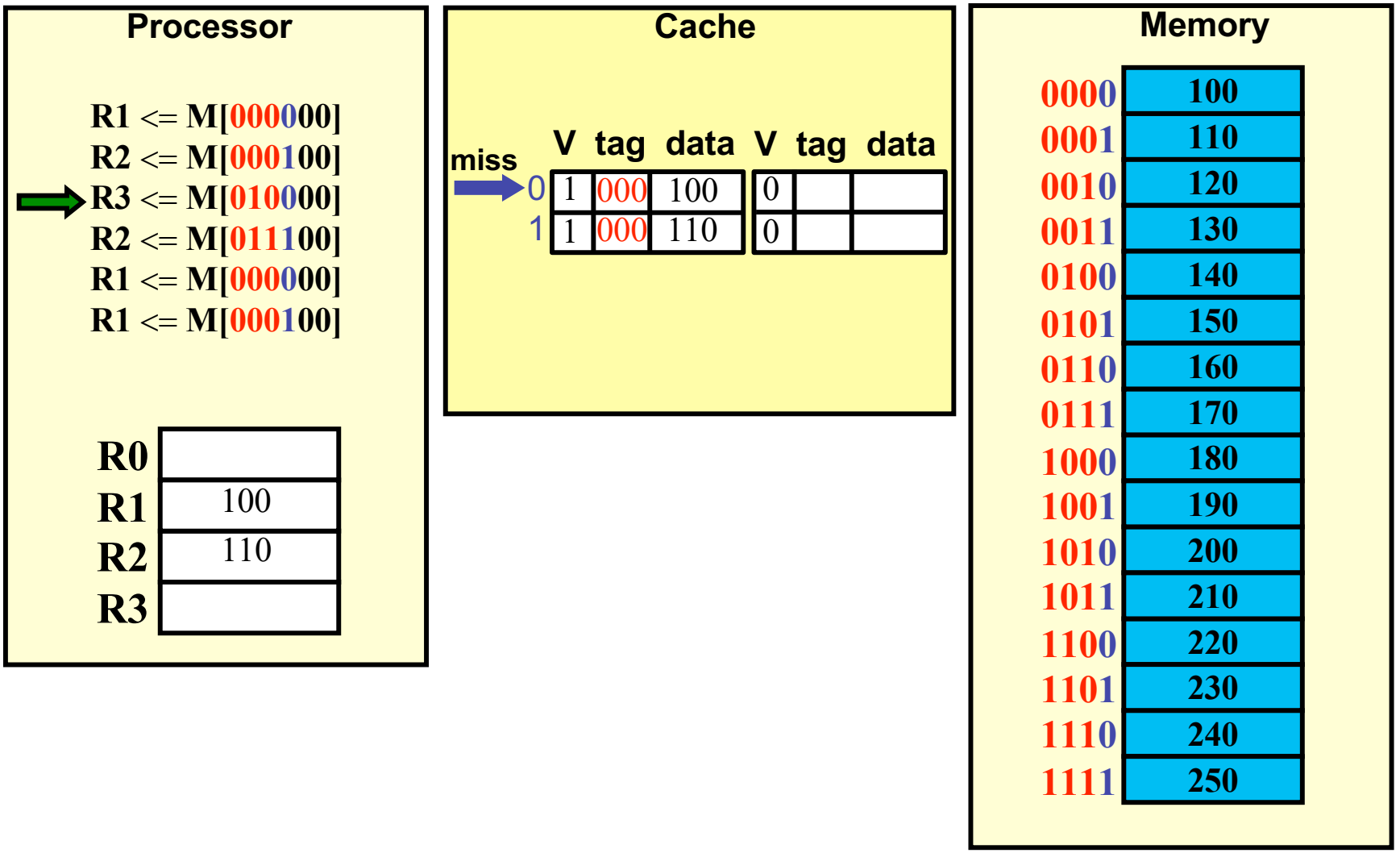
2-way Set Associative Example



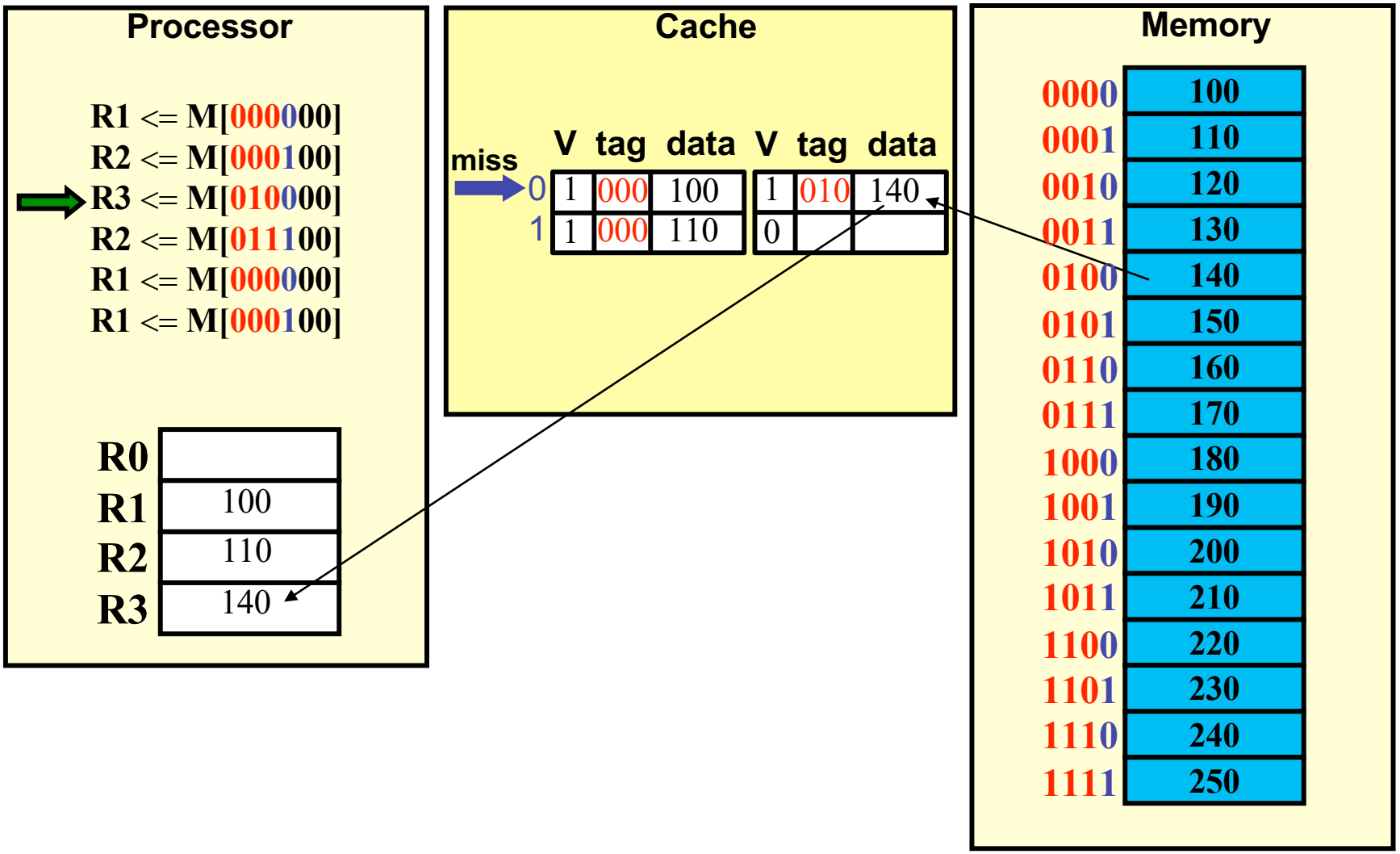
2-way Set Associative Example



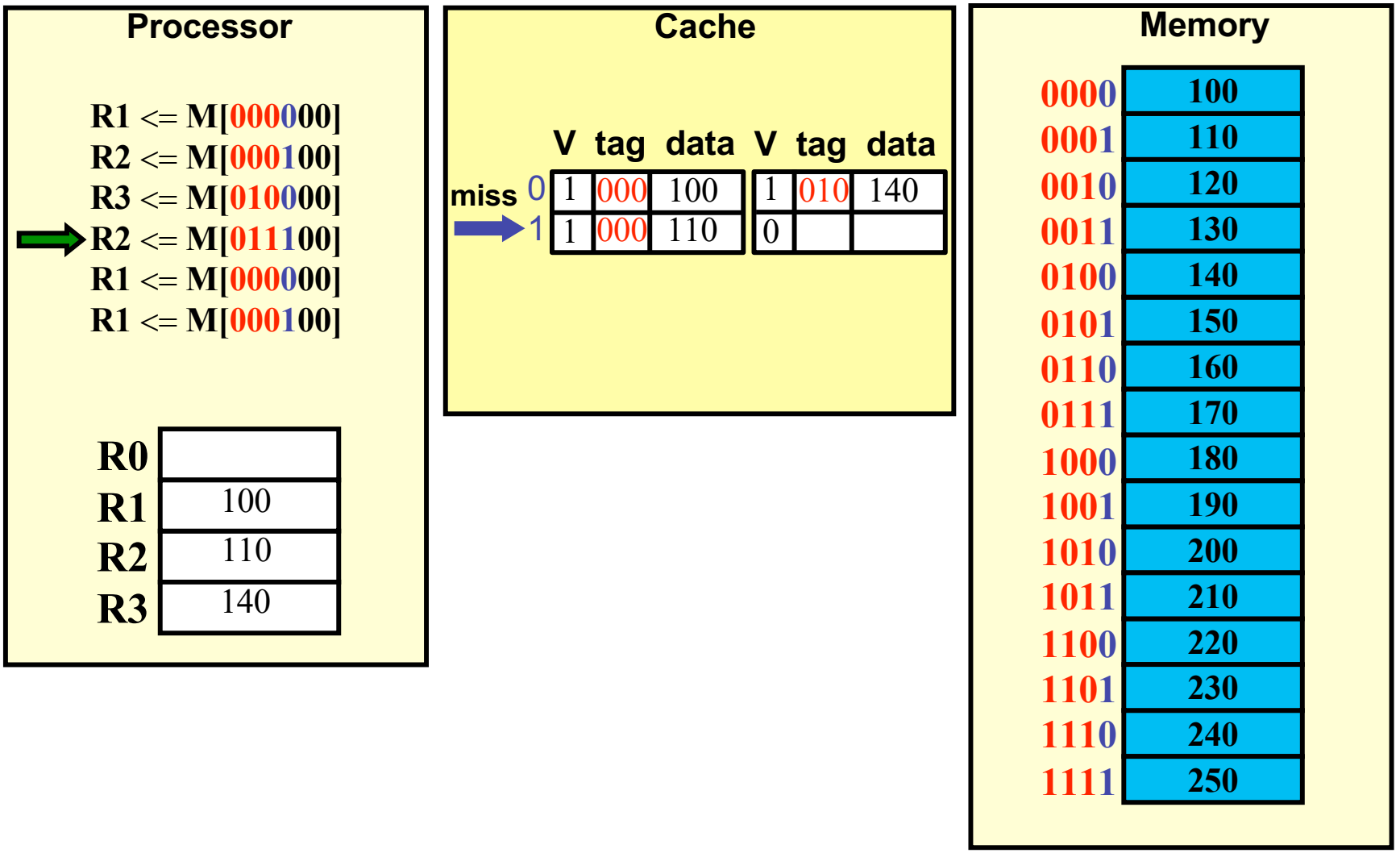
2-way Set Associative Example



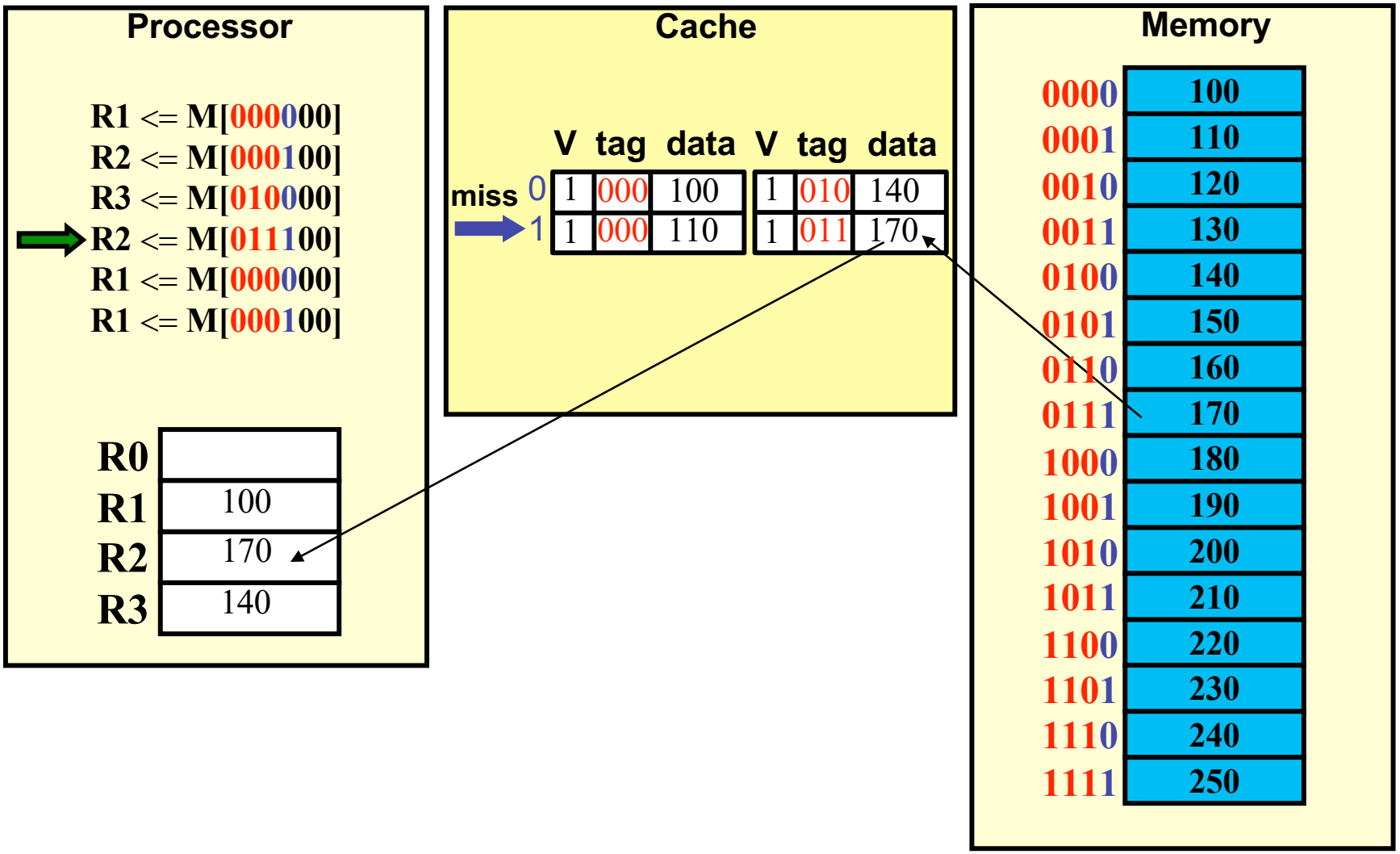
2-way Set Associative Example



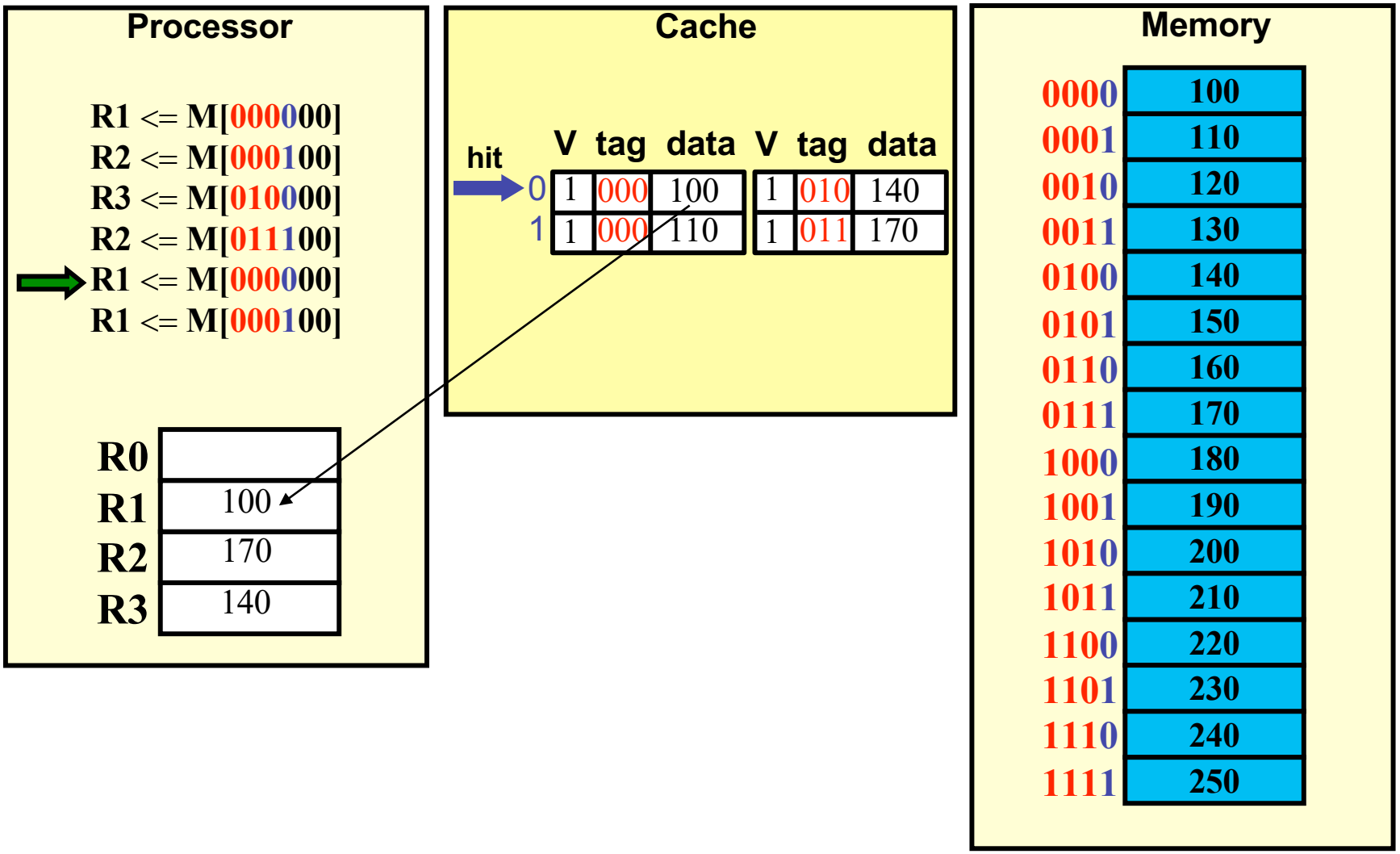
2-way Set Associative Example



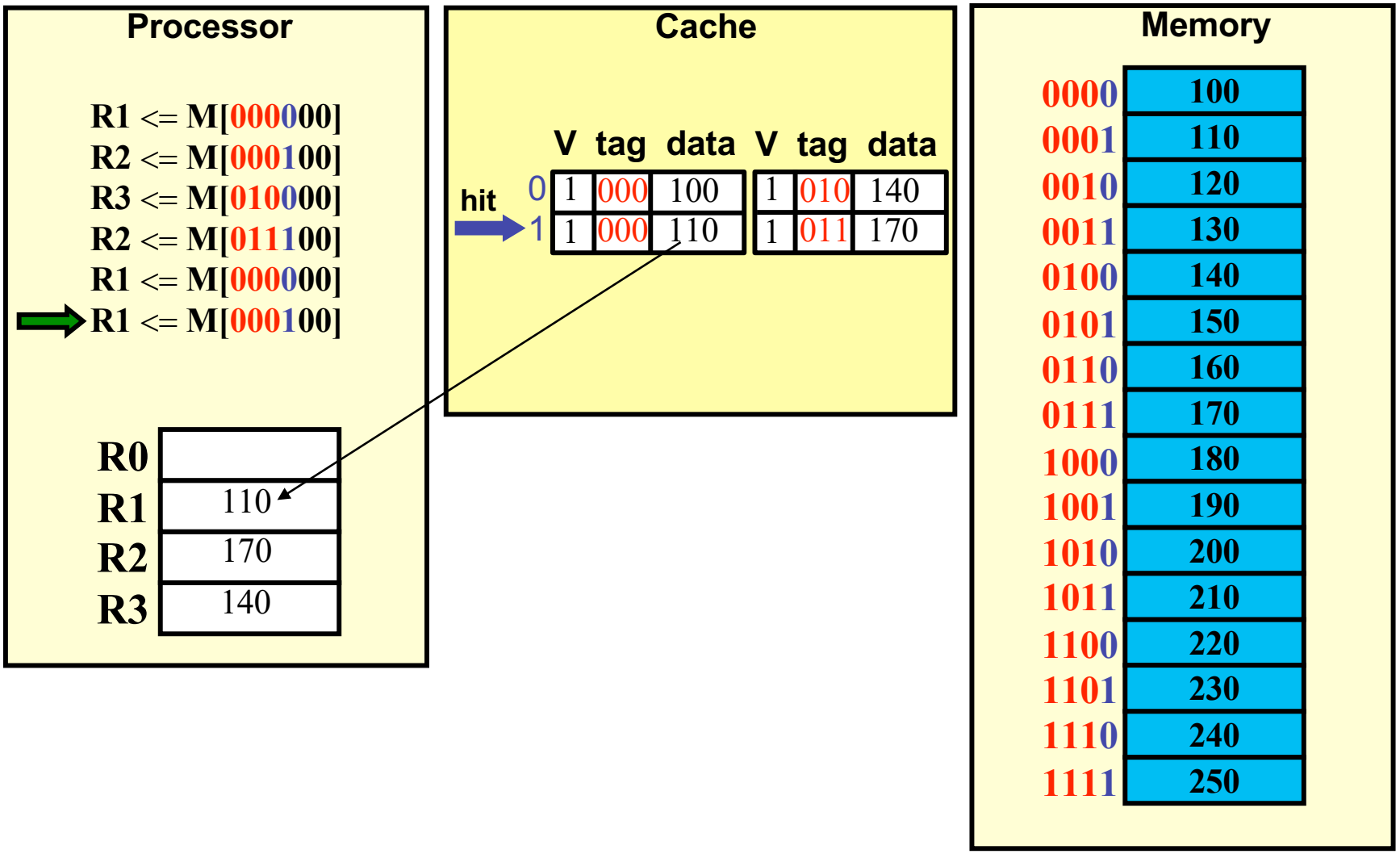
2-way Set Associative Example



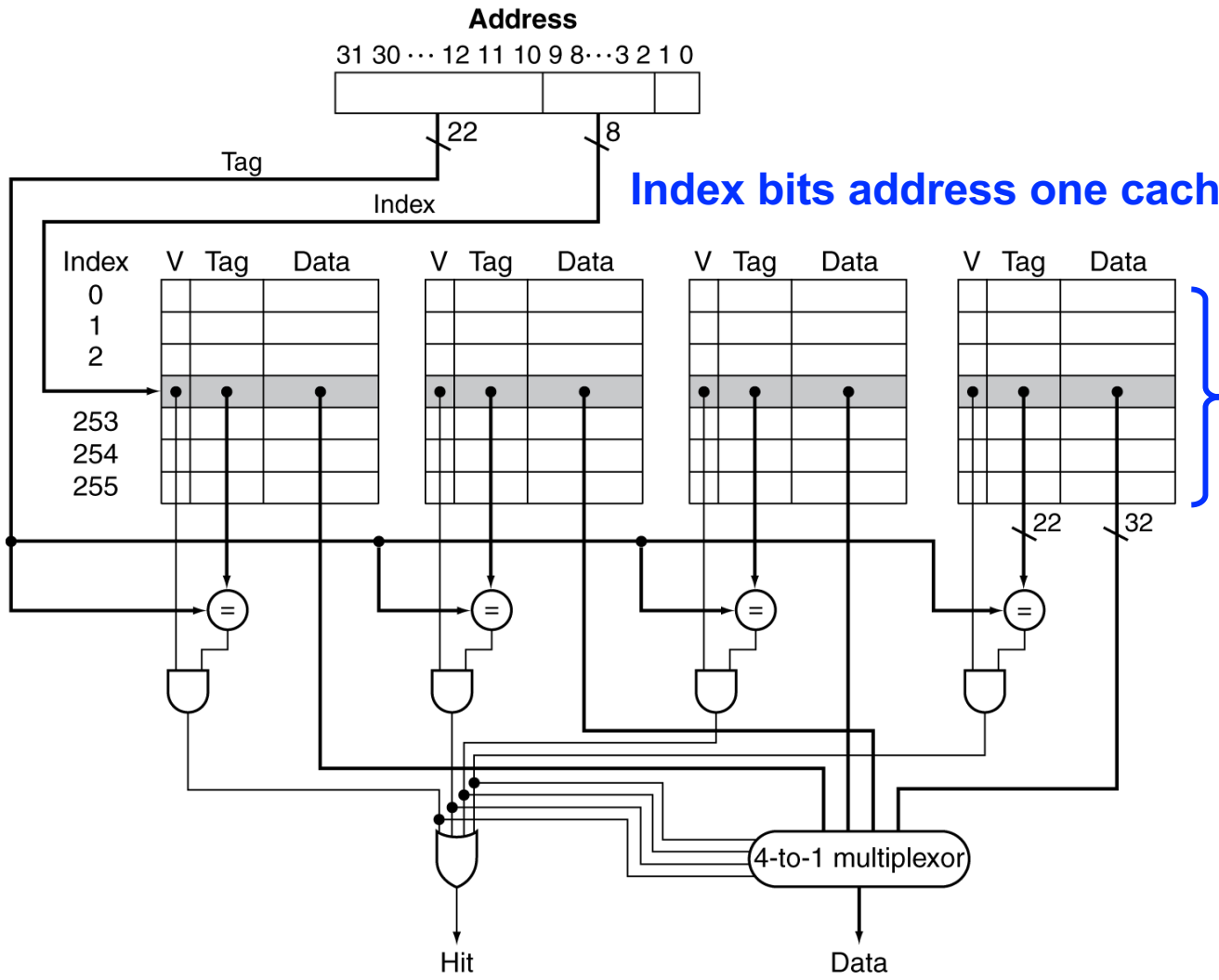
2-way Set Associative Example



2-way Set Associative Example



Example: 4-way Set Associative Cache



All 4 ways within the selected cache set are searched in parallel

Exercise: Associative Cache Address Breakdown

- Assuming 16-bit addresses, how many bits are associated with the tag, index, and offset of the following cache configuration?

- 8 blocks, 4 bytes per block, 4-way set associative

Exercise: Associative Cache Address Breakdown

- Assuming 16-bit addresses, how many bits are associated with the tag, index, and offset of the following cache configuration?

- 8 blocks, 4 bytes per block, 4-way set associative

  Byte offset: 2 bits

  Index: 1 bit

  Tag: 13 bits

Spectrum of Associativity

- A *K*-way set associative cache with *N* blocks
  - Number of cache sets $S = N / K$
    - Number of index bits = $\log_2(S)$
  - When $K = N$, fully associative cache
    - ONE cache set → zero index bits
  - When $K = 1$ (one-way), direct mapped cache
    - *N* cache sets

- Increasing the associativity
  - Typically improves the hit rate (fewer conflicts)
  - But increases the hit time (takes longer to search)

Spectrum of Associativity

For a cache with 8 blocks, 4 bytes per block, 16-bit memory address

One-way set associative (direct mapped)

Block	Tag	Data
0		
1		
2		
3		
4		
5		
6		
7		

Two-way set associative

Set	Tag	Data	Tag	Data
0				
1				
2				
3				

Four-way set associative

Set	Tag	Data	Tag	Data	Tag	Data	Tag	Data
0								
1								

Eight-way set associative (fully associative)

Tag	Data	Tag	Data	Tag	Data	Tag	Data	Tag	Data	Tag	Data	Tag	Data	Tag	Data

How Much Associativity?

- Higher associativity decreases miss rate
  - But with diminishing returns

- Miss rates for 64KB data cache, 64 byte block size, SPEC2000 benchmarks
  - 1-way: 10.3%
  - 2-way: 8.6%
  - 4-way: 8.3%
  - 8-way: 8.1%

Next Class

More Caches