

---

**SI232 Slide Set #17:  
More More Memory (Hierarchy)  
(Chapter 7)**

1

---

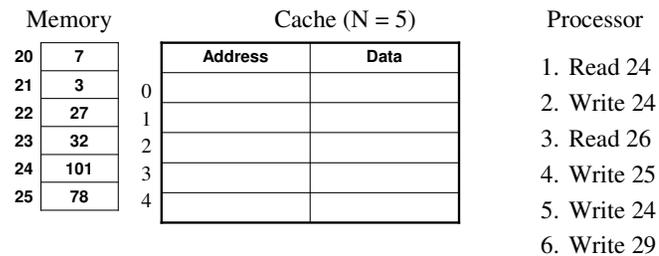
**Improving our Simple Cache**

1. How to handle a write?
2. Efficient Bit Manipulation
3. How to handle a miss?
4. How to eliminate even more conflicts?
5. Can hierarchy help?

2

---

**Issue #1: What to do on a write?**



3

---

**Comparing Write Strategies**

- Write-through:
- Write-back
- How to improve write-through?

4

## Issue #2: Efficient Bit Manipulation

**OLD:** 
$$\text{Index} = \left\lfloor \frac{\text{ByteAddress}}{\text{BytesPerBlock}} \right\rfloor \bmod N$$

Example:

BytesPerBlock = 8  
N = 16

How to

New: ByteOffset =

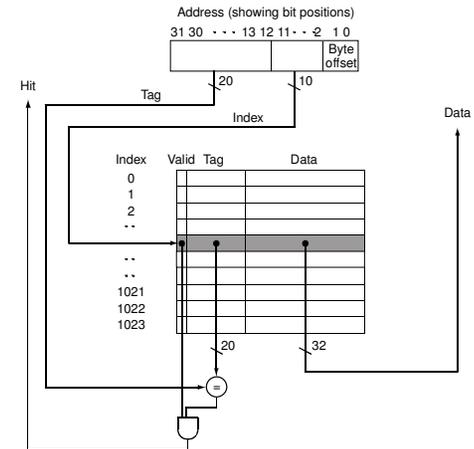
Index =

Example:

Address 0000 1000 0101 1100 0001 0001 0111 1001

5

## Real Cache with Efficient Bit Manipulation



6

## Example #1: Bit Manipulation

1. Suppose cache has:

- 8 byte blocks
- 256 blocks

Show how to break the following address into the tag, index, & byte offset.

0000 1000 0101 1100 0001 0001 0111 1001

2. Same cache, but now 4-way associative. How does this change things?

0000 1000 0101 1100 0001 0001 0111 1001

7

## Example #2: Bit Manipulation

Suppose a direct-mapped cache divides addresses as follows:



What is the block size?

The number of blocks?

Total size of the cache?

(usually refers to size of data only)

8

## Key Rules

---

- How the # sets and # blocks relate?
  
- Calculate # index bits from # sets
  
- One hex 'digit' = 4 bits
  - 0x1234 = 0001 0010 0011 0100

9

## Exercise #2

---

1. Suppose cache has:
  - 4 byte blocks
  - 128 blocks
 Show how to break the following address into the tag, index, & byte offset.  
 0000 1000 0101 1100 0001 0001 0111 1001
  
2. Same cache, but now 8-way associative. How does this change things?  
 0000 1000 0101 1100 0001 0001 0111 1001

11

## Exercise #1

---

Suppose a cache divides addresses as follows:



Fill in the values for a direct-mapped or 4-way associative cache:

	Direct-mapped	4-way associative
Block size		
Number of blocks		
Total size of cache (e.g. 32 * 128 – don't have to multiply out)		
Tag size (# bits)		

10

## Exercise #3

---

- Given a cache that is:
  - 4-way associative
  - 32 blocks
  - 16 byte block size
 What is the cache index and byte offset for the following address:  
 0x3ab12395

Cache index =  
 Byte offset =

And this one:  
 0x70ff1213

Cache index =  
 Byte offset =

Do these addresses conflict in the cache?

12

## Exercise #4

- Cache parameters are often a power of two. Given what you know so far, explain why each of the following should be a power of two (or need not be).
  - Block size
  
  - Number of cache blocks
  
  - Number of cache sets
  
  - Associativity

13

## Exercise #5

- What is the total number of bits needed to implement the storage for the direct mapped cache given in Exercise #2. What do you need these bits for besides the data?

14

## Issue #3: How to handle a miss?

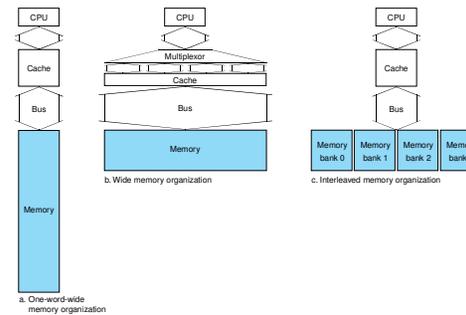
- Things we need to do:
  1. \_\_\_\_\_ the CPU until miss completes
  2. \_\_\_\_\_ old data from the cache  
Which data?
  
  3. \_\_\_\_\_ the needed data from memory  
Pay the \_\_\_\_\_  
How long does this take?
  4. \_\_\_\_\_ the CPU

What about a write miss?

15

## Decreasing the Miss Penalty

- Time to fetch data from memory (with sample times) =  
SendAddress (1 bus cycle) +  
Initiate DRAM Access (15 bus cycles per word read) +  
Bus transfer time (1 bus cycle per word)
- How can we decrease this?



16

## Issue #4: How to eliminate even more conflicts?

---

- Fully associative cache – cache block can go \_\_\_\_\_ in cache
- Pros
- Cons
- Can view all caches as n-way associative:
  - Direct-mapped, n =
  - 4-way associative, n =
  - Fully associative, n =

17

## Issue #5: More hierarchy – L2 cache?

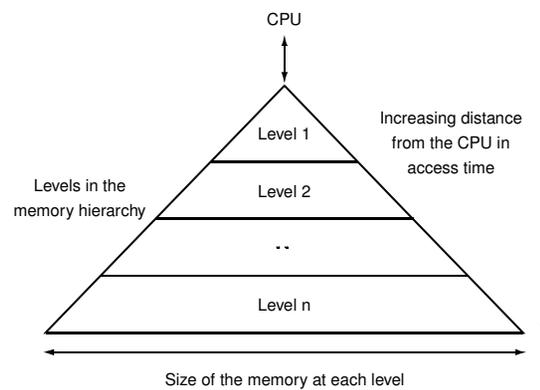
---

- Add a second level cache:
  - often primary cache is on the same chip as the processor
  - use SRAMs to add another cache above primary memory (DRAM)
  - miss penalty goes down if data is in 2nd level cache
- Performance smarts:
  - try and optimize the \_\_\_\_\_ on the 1st level cache
  - try and optimize the \_\_\_\_\_ on the 2nd level cache

18

## Memory Hierarchy

---



19

## Questions

---

- Will the miss rate of a L2 cache be higher or lower than for the L1 cache?
- Claim: "The register file is really the highest level cache" What are reasons in favor and against this statement?

20

## More Questions

---

- How else might you might improve the performance of a cache.  
Think about either:
  - Improving hit time
  - Improving the hit rate
  - Decreasing miss penalty