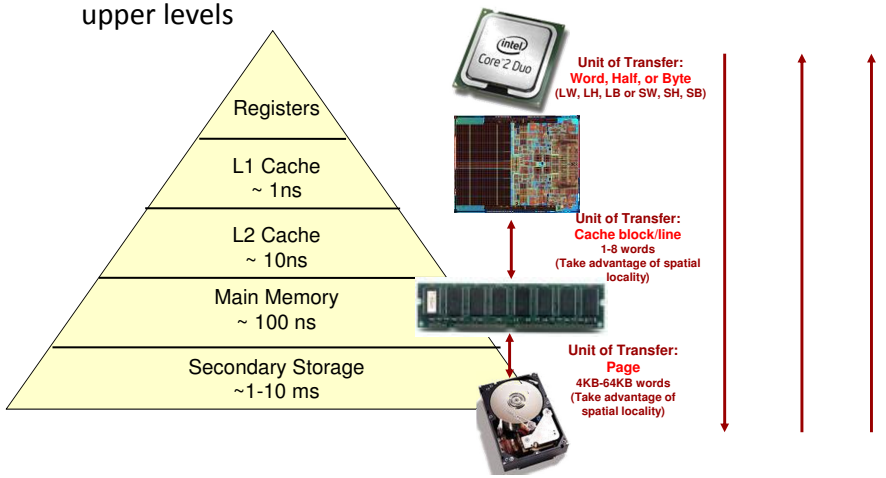
EE 457 Unit 7a

Cache and Memory Hierarchy

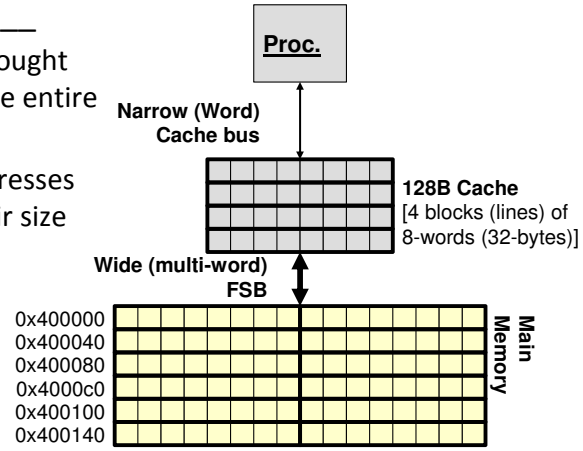
Memory Hierarchy & Caching

- Use several levels of faster and faster memory to hide delay of upper levels



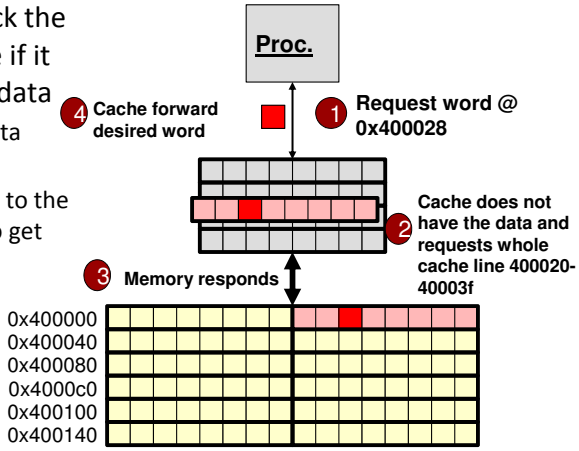
Cache Blocks/Lines

- Cache is broken into _____ or _____
  - Any time data is brought in, it will bring in the entire block of data
  - Blocks start on addresses _____ of their size



Cache Blocks/Lines

- Whenever the processor generates a read or a write, it will first check the cache memory to see if it contains the desired data
  - If so, it can get the data _____ from cache
  - Otherwise, it must go to the slow main memory to get the data



Cache & Virtual Memory

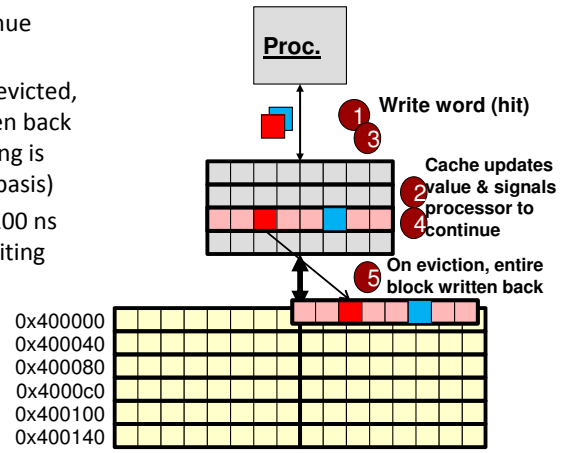
- Exploits the Principle of Locality
  - Allows us to implement a hierarchy of memories: cache, MM, second storage
  - Temporal Locality: If an item is reference it will tend to be _____
    - Examples: _____, _____, setting a variable and then reusing it many times
  - Spatial Locality: If an item is referenced items whose _____ will tend to be referenced soon
    - Examples: _____ and _____

Cache Definitions

- Cache _____ = Desired data is in cache
- Cache _____ = Desired data is not present in cache
- When a cache _____ occurs, a new block is brought from MM into cache
  - _____: First load the word requested by the CPU and forward it to the CPU, while continuing to bring in the remainder of the block
  - _____: First load entire block into cache, then forward requested word to CPU
- On a _____ we may choose to not bring in the MM block since writes exhibit less locality of reference compared to reads
- When CPU writes to cache, we may use one of two policies:
  - _____: Every write updates both cache and MM copies to keep them in sync. (i.e. coherent)
  - _____: Let the CPU keep writing to cache at fast rate, not updating MM. Only copy the block back to MM when it needs to be replaced or flushed

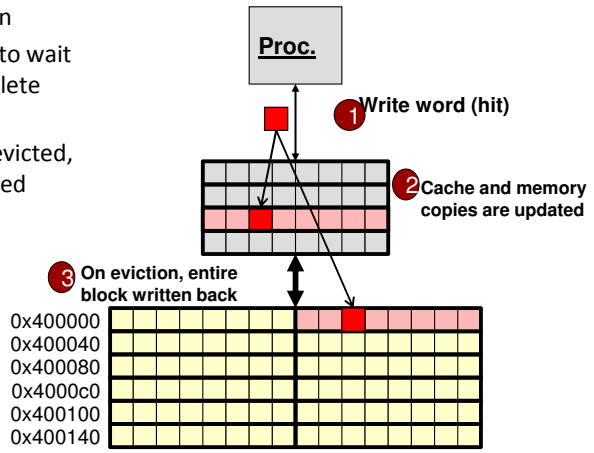
Write Back Cache

- On write-hit
  - Update only cached copy
  - Processor can continue quickly (e.g. 10 ns)
  - Later when block is evicted, entire block is written back (because bookkeeping is kept on a per block basis)
    - Ex: 8 words @ 100 ns per word for writing mem.
      = 800 ns



Write Through Cache

- On write-hit
  - Update both cached and main memory version
  - Processor may have to wait for memory to complete (e.g. 100 ns)
  - Later when block is evicted, no writeback is needed

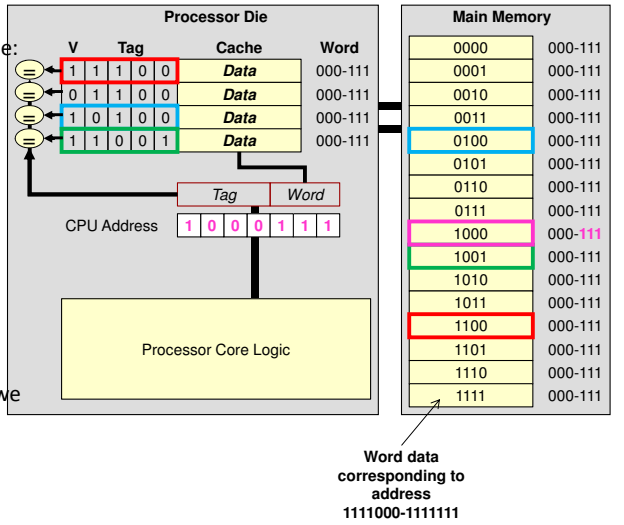


Cache Definitions

- Mapping Function: The correspondence between MM blocks and cache block frames is specified by means of a mapping function
  - _____
  - _____
  - _____
- Replacement Algorithm: How do we decide which of the current cache blocks is removed to create space for a new block
  - _____
  - _____

Fully Associative Cache Example

- Cache Mapping Example:
  - Fully Associative
  - MM = 128 words
  - Cache Size = 32 words
  - Block Size = 8 words
- Fully Associative mapping allows a MM block to be placed (associate with) _____ _____ cache block
- To determine hit/miss we have to search _____



Implementation Info

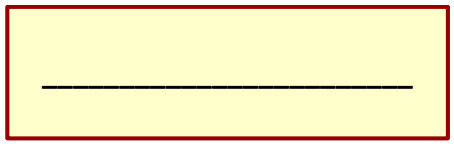
- Tags: Associated with each cache block frame, we have a TAG to identify its _____
- Valid bit: An additional bit is maintained to indicate that whether the TAG is valid (meaning it contains the TAG of an actual block)
  - Initially when you turn power on the cache is empty and all valid bits are turned to '0' (invalid)
- _____: This bit associated with the TAG indicates when the block was modified (got dirtied) during its stay in the cache and thus needs to written back to MM (used only with the write-back cache policy)

Fully Associative Hit Logic

- Cache Mapping Example:
  - Fully Associative, MM = 128 words (2^7), Cache Size = 32 (2^5) words, Block Size = (2^3) words
- Number of blocks in MM = _____
- Block ID = _____
- Number of Cache Block Frames = _____
  - Store ____ Tags of 4-bits + 1 valid bit
  - Need 4 _____ each of _____
- CAM (Content Addressable Memory) is a special memory structure to store the tag+valid bits that takes the place of these comparators but is too expensive

Fully Associative Does Not Scale

- If 80386 used Fully Associative Cache Mapping :
  - Fully Associative, MM = 4GB (2^{32}), Cache Size = 64KB (2^{16}), Block Size = ($16=2^4$) bytes = 4 words
- Number of blocks in MM = _____
- Block ID = _____
- Number of Cache Block Frames = _____
  - Store _____ Tags of 28-bits + 1 valid bit
  - Need _____ Comparators each of 29 bits

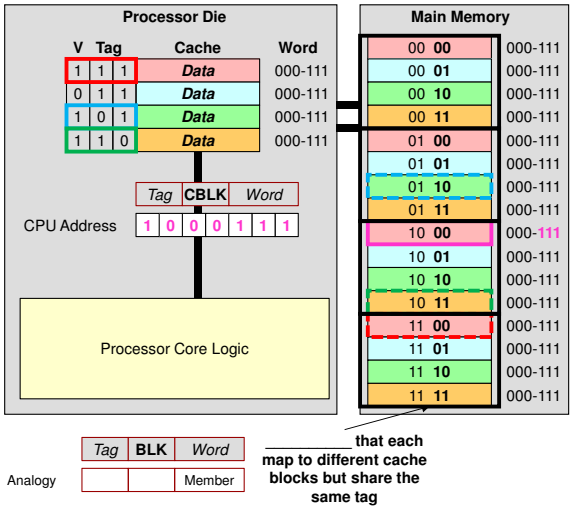


Fully Associative Address Scheme

- A[1:0] unused => /BE3…/BE0
- Word bits = _____
- Tag = Remaining bits

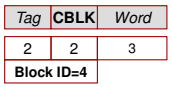
Direct Mapping Cache Example

- Limit each MM block to _____ location in cache
- Cache Mapping Example:
  - Direct Mapping
  - MM = 128 words
  - Cache Size = 32 words
  - Block Size = 8 words
- Each MM block i maps to Cache frame _____
  - N = # of cache frames
  - Tag identifies which group that colored block belongs



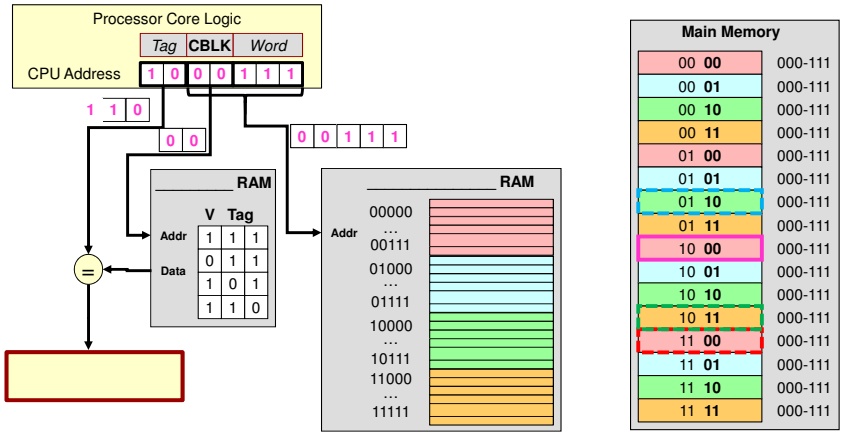
Direct Mapping Address Usage

- Cache Mapping Example:
  - Direct Mapping, MM = 128 words (2^7), Cache Size = 32 (2^5) words, Block Size = (2^3) words
- Number of blocks in MM = 2^7 / 2^3 = 2^4
- Block ID = 4 bits
- Number of Cache Block Frames = 2^5 / 2^3 = 2^2 = 4
  - Number of "colors“ => ____ Number of Block field Bits
- _____ = 4 Groups of blocks
  - 2 Tag Bits



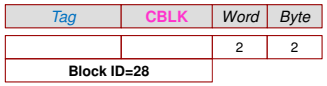
Direct Mapping Hit Logic

- Direct Mapping Example:
  - MM = 128 words, Cache Size = 32 words, Block Size = 8 words
- Block field addresses tag RAM and compares stored tag with tag of desired address



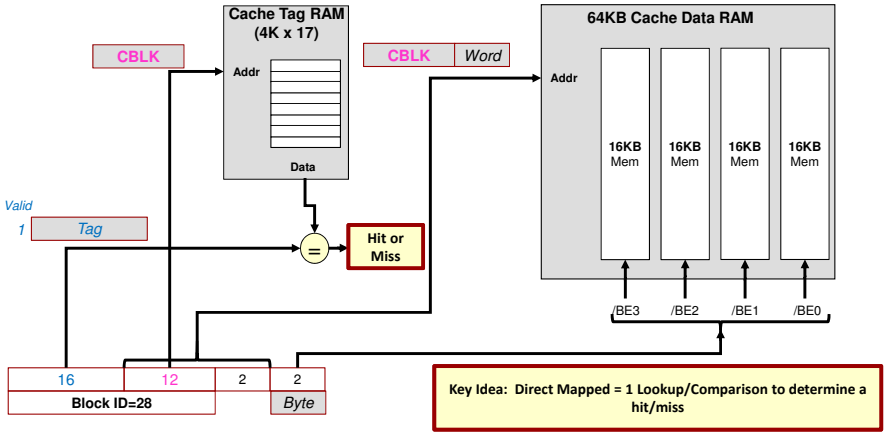
Direct Mapping Address Usage

- If 80386 used Direct Cache Mapping :
  - MM = 4GB (2^{32}), Cache Size = 64KB (2^{16}), Block Size = ($16=2^4$) bytes = 4 words
- Number of blocks in MM = 2^{32} / 2^4 = 2^{28}
- Number of Cache Block Frames = 2^{16} / 2^4 = 2^{12} = 4096
  - Number of "colors“ => _____ Block field bits
- _____ Groups of blocks
  - 16 Tag Field Bits



Tag and Data RAM

- 80386 Direct Mapped Cache Organization

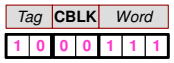


Direct Mapping Address Usage

- Divide MM and Cache into equal size blocks of __ words
  - M main memory blocks, N cache blocks
  - Log_2(B) word field bits
- A block in caches is often called a cache block/line frame since it can hold many possible MM blocks over time
- For direct mapping, if you have N cache frames, then define N “colors/patterns”
  - _____ block field bits
- Repeatedly paint MM blocks with those N colors in round-robin fashion
- _____ groups will form
  - Log_2(_____) tag field bits

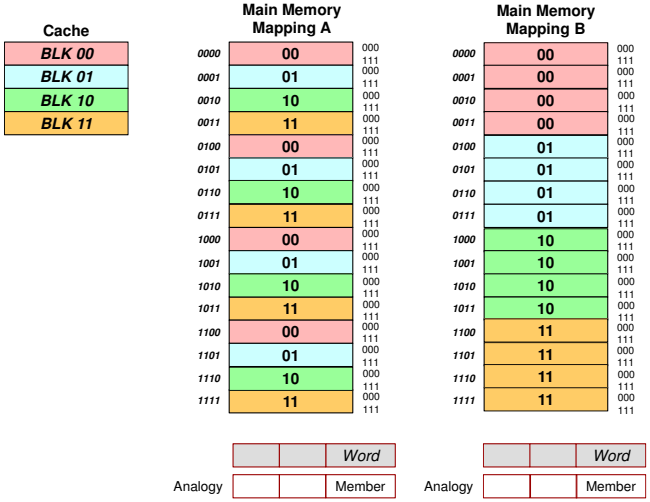
Direct Mapping Datapath

- How many TAG RAM's?
  - Is that answer dependent on address field sizes?
- How many entries in the TAG RAM?
- How many bits wide is each entry in the TAG RAM?
- How many DATA RAM's?
  - What size is the address field?



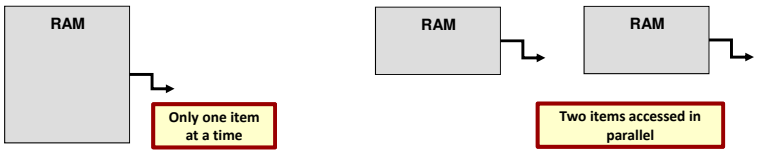
Alternate Direct Mapping Scheme

- Can you "color" (i.e. map) the blocks of main memory in a different order?
- Use _____ as BLK field or _____ bits
- Which is more desirable or does it not really matter?



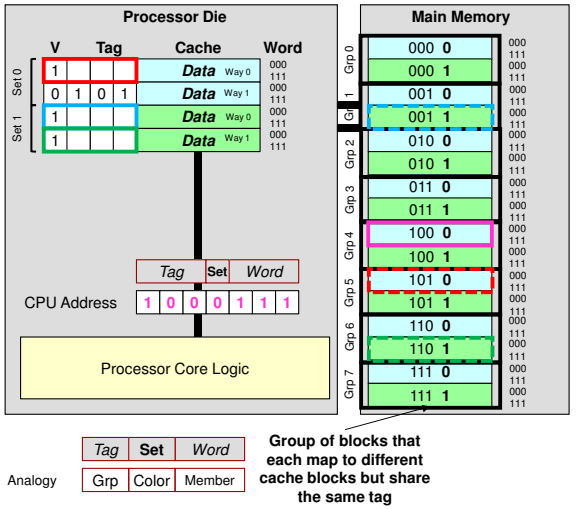
Single or Parallel RAM's

- Is it cheaper to have
  - (1) 2KB RAM
  - (2) 1KB RAM's
- Area wise a 2KB RAM _____
- For tag and data RAMs it would be more economical to use fewer, big RAM's
- However, consider need for parallel access

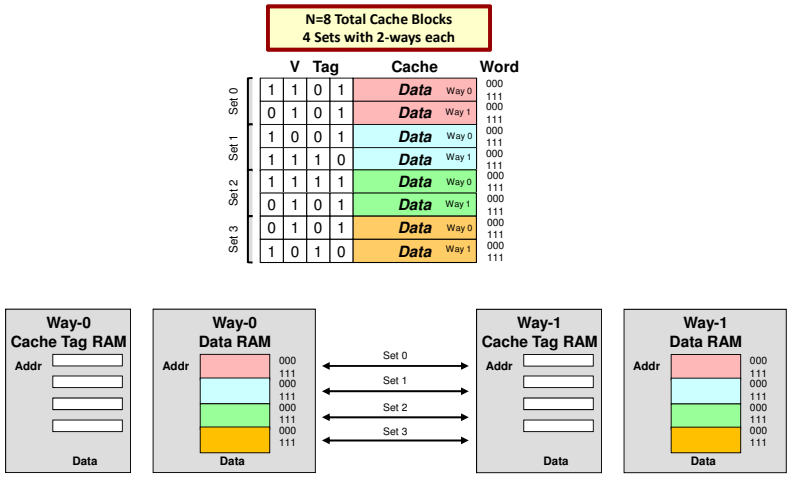


Set-Associative Mapping Example

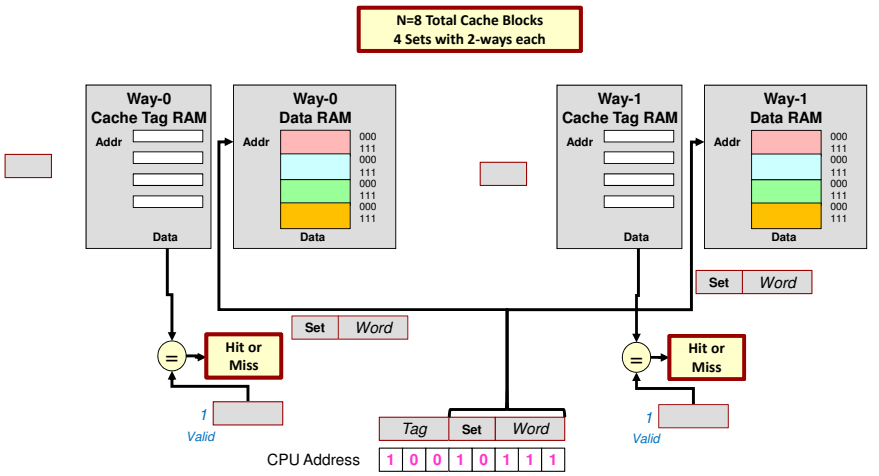
- Cache Mapping Example:
  - Direct Mapping
  - MM = 128 words
  - Cache Size = 32 words
  - Block Size = 8 words
- Each MM block i maps to Cache frame _____
  - S = # of sets (_____ of cache frames)
  - Tag identifies which group that colored block belongs to



Set-Associative Datapath



Set-Associative Datapath



Set-Associative Mapping Address Usage

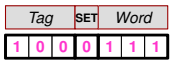
- Define K = _____
- If you have N total cache frames, then define number of sets, S, = _____
- Define S colors/patterns
  - $Log_2(S) = Log_2(\_\_\_\_\_)$ set field bits
- Repeatedly paint MM blocks with those S colors in round-robin fashion
- _____ groups will form
  - $Log_2(\_\_\_\_)$ tag field bits

Set-Associative Mapping Datapath

- How many TAG RAM's?
- How many entries in the TAG RAM?
- Place tags from different sets that belong to 'Way 0' in one tag ram, 'Way 1' in another, etc.
- How many DATA RAM's?
  - What size is the address field?

Key Idea: K-Ways => K comparators

(What is a 1-way Set Associative Mapping)

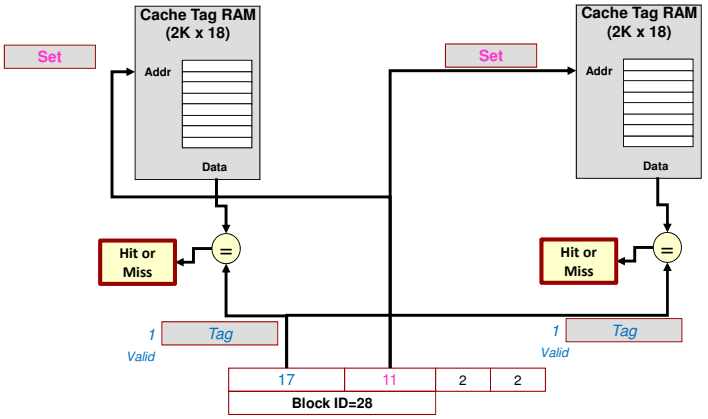


K-Way Set Associative Mapping

- If 80386 used K-Way Set-Associative Mapping:
  - MM = 4GB (2^{32}), Cache Size = 64KB (2^{16}), Block Size = ($16=2^4$) bytes = 4 words
- Number of blocks in MM = 2^{32} / 2^4 = 2^{28}
- Number of Cache Block Frames = 2^{16} / 2^4 = 2^{12} = 4096
- Set Associativity/Ways (K) = 2 Blocks/Set
  - Number of "colors" => $2^{12}/2$ = 2^{11} Sets => 11 Set field bits
- 2^{28} / 2^{11} = 2^{17} = 128K Groups of blocks
  - 17 Tag Field Bits

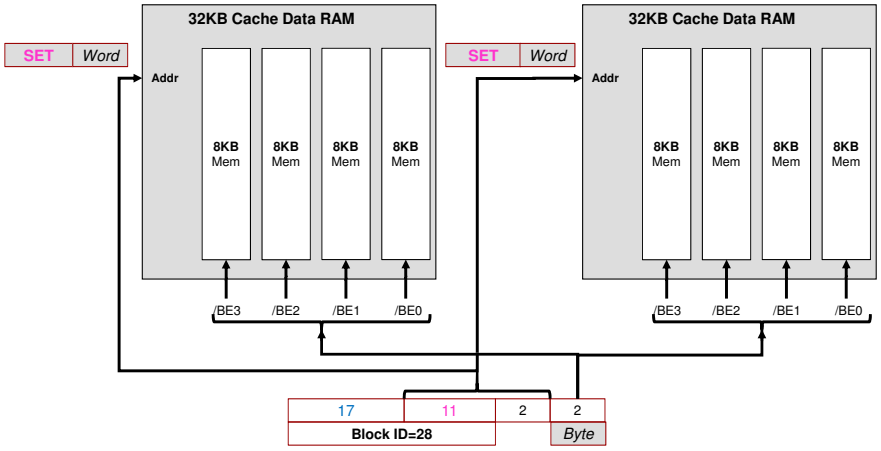
Tag RAM Organizations

- 80386 2-Way Set-Associative Cache Organization



Data RAM Organizations

- 80386 2-Way Set-Associative Cache Organization



Set Associative Example

Tag	Set	Word	Byte
18	10	2	2

- Suppose the cache size is 2^{12} blocks
- What is the set size?
- If the set associativity can be changed,
  - What is the smallest set size?
    - Maximum # of sets =
    - Largest Set Field=_____ Smallest Tag=
    - _____ Mapping
  - What is the largest set size?
    - Minimum # of sets =
    - Smallest Set Field=_____ , Largest Tag=
    - _____ Mapping

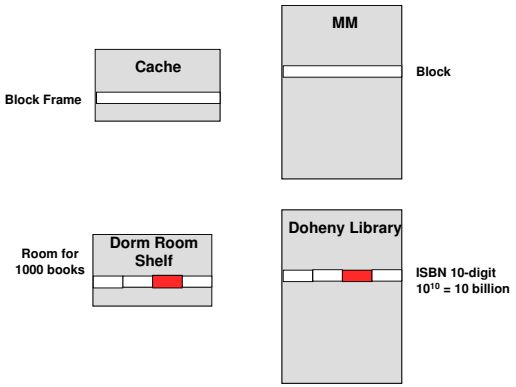
LIBRARY ANALOGY

Mapping Functions

- A mapping function determines the correspondence between MM blocks and cache block frames
- 3 Schemes
  - Fully Associative
  - Direct Mapping
  - Set-Associative
- Really just 1 scheme
  - Fully Associative = N-way Set Associative
  - Direct Mapping = 1-way Set Associative

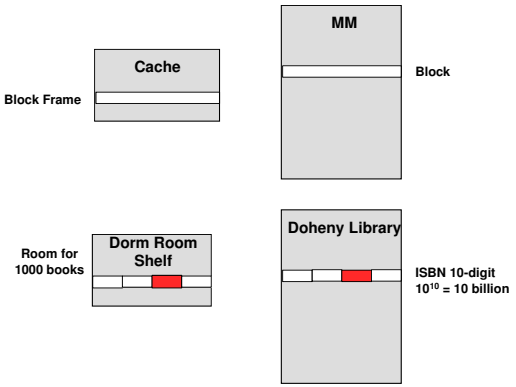
Library ⇔ Memory

- Compare MM to a large library
- Compare cache to your dorm room book shelf
- "Address" of a book = 10-digit ISBN number
- Assume library has a location on the shelf for all 10^{10} possible books



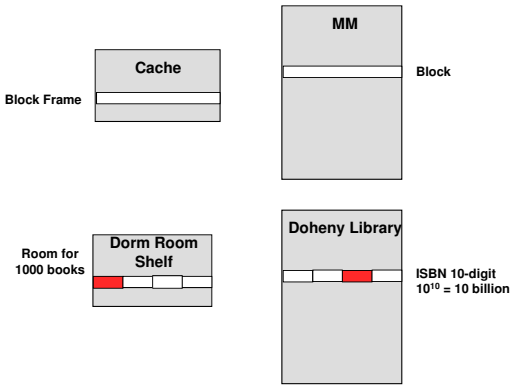
Book Addressing

- Addresses are not stored in memory (only data)
- Assume library has a location on the shelf for all 10^{10} possible books
- No need to print ISBN on the book if each book has a location (find a book by going to its slot using ISBN as index)



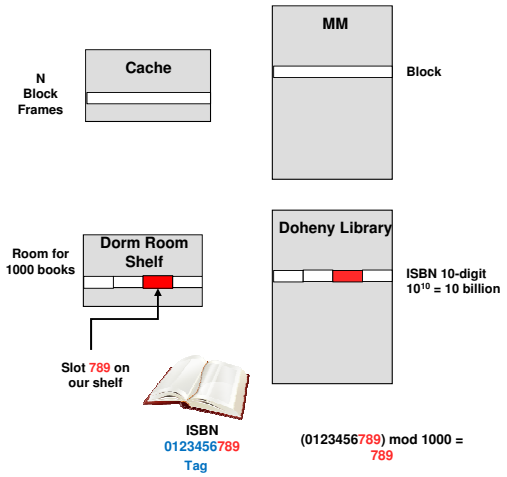
Fully Associative Analogy

- Cache stores full Block-ID as a TAG to identify that block
- When we check a book out and take it to our dorm room shelf…
  - Let's allow it to be put in any free slot on the shelf
  - We need to keep the entire ISBN number as a TAG
- To find a book with a given ISBN on our shelf, we must look through them all



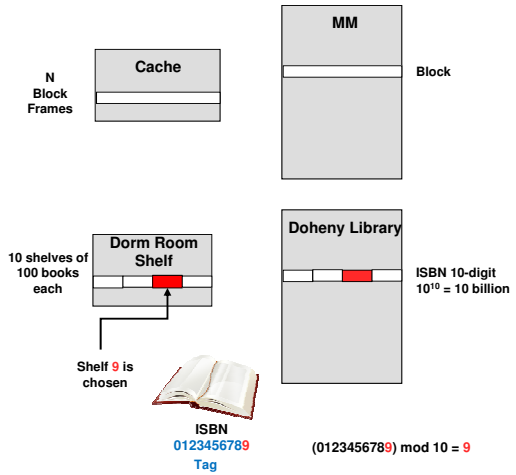
Direct Mapping Analogy

- Cache uses block field to identify the slot in the cache and then stores remainder as TAG to identify that block from others that also map to that slot
- Assume we number the slots on our book shelf from 0 to 999
- When we check a book out and take it to our dorm room shelf we can…
  - Use last 3-digits of ISBN to pick the slot to store it
  - If another book is their, take it back to Doheny library (evict it)
  - Store upper 7 digits to identify this book from others that end with the same 3-digits
- To find a book with a given ISBN on our shelf, we use the last 3-digits to choose which slot to look in and then compare the upper 7-digits



Set Associative Mapping Analogy

- Cache blocks are divided into groups known as sets. Each MM block is mapped to a particular set but can be anywhere in the set (i.e. all TAGS in the set must be compared)
- Assume our bookshelf is 10 shelves with room for 100 books each
- When we check a book out and take it to our dorm room shelf we can…
  - Use last 1-digit of ISBN to pick the shelf but store the book anywhere on the shelf where there is an empty slot
  - Only if the shelf is full do we have to pick a book to take back to Doheny library (evict it)
  - Store upper 9 digits to identify this book from others that end with the same 1-digit
- To find a book with a given ISBN on our shelf, we use the last 1-digits to choose which shelf to look in and then compare upper 9-digits with those of all the books on the shelf



Set Associative Mapping Analogy

- Can we confidently say,
  - We can bring in any (10/100/other) book(s)
  - We can bring in (10/100/other) consecutive book(s)
- Library analogy:
  - 10 sets each with 100 slots = 100-way set associative cache