Cache Example

- 8-blocks, direct mapped

Cache Memory

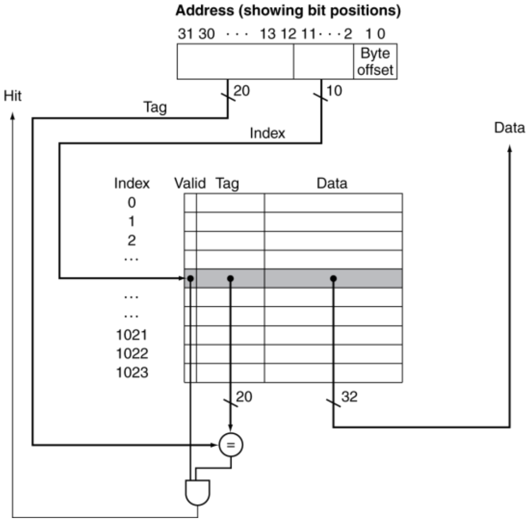
Index	V	Tag	Data
000	~~N~~Y	10	Mem[10 000]
001	N		
010	~~N~~Y	~~11~~10	10 Mem[~~11~~ 010]
011	~~N~~Y	00	Mem[00 011]
100	N		
101	N		
110	~~N~~Y	10	Mem[10 110]
111	N		

Memory Access

Order	Word addr	Binary addr	Hit/miss
1	22	10 110	M
2	26	11 010	M
3	22	10 110	H
4	26	11 010	H
5	16	10 000	M
6	3	00 011	M
7	16	10 000	H
8	18	10 010	M

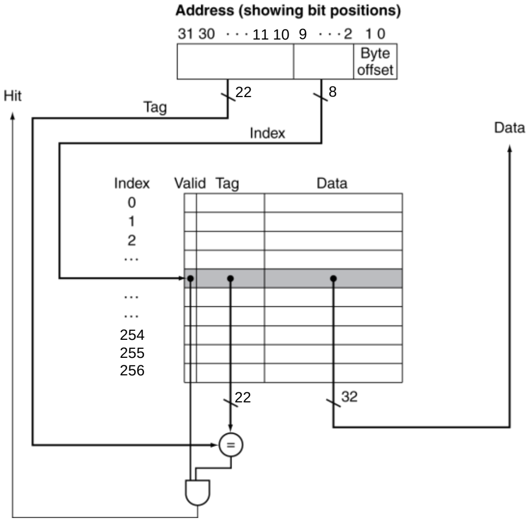
Address Subdivision

- 1024 blocks, 4 bytes/block



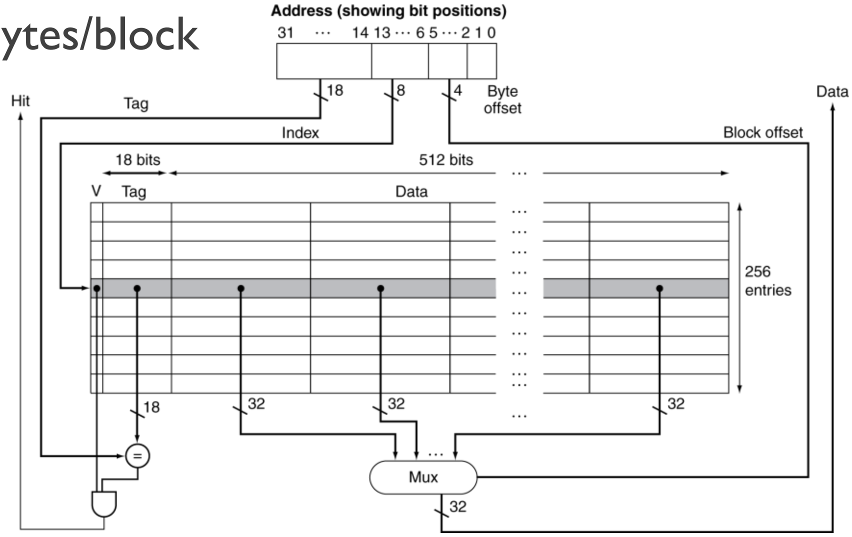
Address Subdivision

- 256 blocks, 4 bytes/block
- Spatial locality considered?
  - No

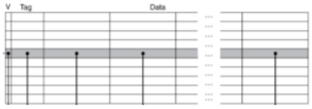


Intrinsity FastMATH

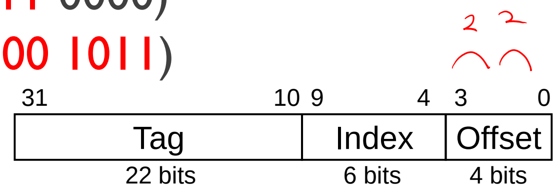
- 256 blocks, 64 bytes/block



Example: Larger Block Size



- 64 blocks, 16 bytes/block
  - To what block number does address 1200 map?
- Block index
  - 1200/16 = 75 (0~0 0100 1011 0000)
  - 75 modulo 64 = 11 (0~0 0100 1011)
- Spatial locality considered?
  - Yes


2 2

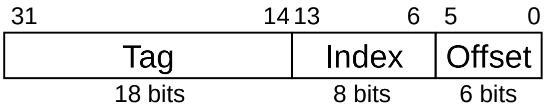
31 10 9 4 3 0

Tag | Index | Offset

22 bits | 6 bits | 4 bits

Example: Larger Block Size

- 256 blocks, 16 words/block
  - To what block number does address 18000 map?
- Block index
  - 18000/(16*4) = 281.25 → 281(0~0 0100 0110 0101 0000)
  - 281 modulo 256 = 25 (0~0 0100 0110 01)


31 14 13 6 5 0

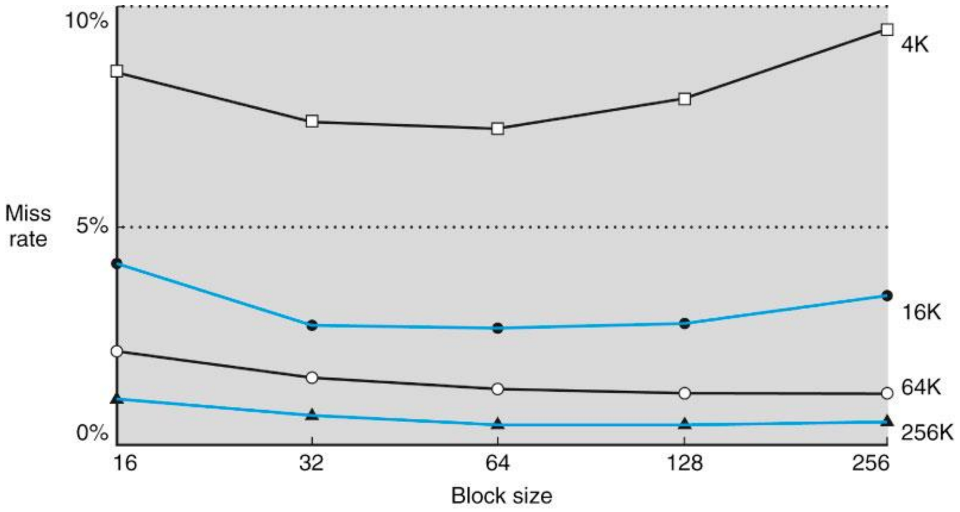
Tag | Index | Offset

18 bits | 8 bits | 6 bits

Block Size Considerations

- Larger blocks are expected to reduce miss rate
  - Due to spatial locality
- But in a fixed-sized cache
  - Larger blocks
  - Fewer number of blocks
  - More competition
  - Increased miss rate
- Larger miss penalty
  - Time required to fetch the block from the lower level
  - Can override benefit of reduced miss rate

Miss rate vs. Block size vs. Total cache size



Write-Through

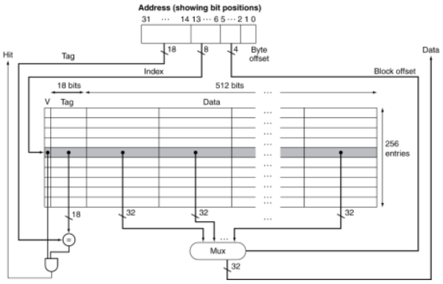
- Write through: also update memory (consistent)
- But makes writes take longer
  - e.g., if base CPI = 1, 10% of instructions are stores, write to memory takes 100 cycles
    - Effective CPI = 1 + 0.1×100 = 11
- Solution
  - write buffer
  - Holds data waiting to be written to memory
  - CPU continues immediately
    - Only stalls on write if write buffer is already full

Write-Back

- Alternative: On data-write hit, just update the block in cache (inconsistent)
  - Keep track of whether each block is dirty
  - Dirty bit tells if the block needs to be written back to memory when it is evicted
- Can use a write buffer to allow replacing block to be read first
- Synchronous process required for multi-cache/shared-memory system.

Example: Intrinsity FastMATH

- Embedded MIPS processor
  - 12-stage pipeline
  - Instruction and data access on each cycle
- Split cache: separate I-cache and D-cache
  - Each 16KB: 256 blocks × 16 words/block
  - D-cache: write-through or write-back
- SPEC2000 miss rates
  - I-cache: 0.4%
  - D-cache: 11.4%
  - Weighted average: 3.2%



11/30