

Direct Mapping Address Structure

Tag	Line or Slot	Word
t	s	w

- Cache line size determines how many bits in word field (ex: 32 bytes => $w = 5$)
- Number of lines in cache determines number of bits in slot field (ex: 4096 lines => $s = 12$)
- Address bits – ($s+w$) is tag size (ex: 32 bit addr – 17 bits => $t = 15$)
- So for memory address 4C0180F7:

4C0180F7 = 0100 1100 0000 0001 1000 0000 1111 0111		
0100110000000000	110000000111	10111
tag (15 bits)	Slot(12 bits)	word (5 bits)
- Each address maps to exactly ONE slot (determined by s field)
- To determine cache hit, compute slot number, check tag at that slot against tag of memory address

Set Associative Mapping Address Structure

Tag t bits	Set s bits	Word w bits
------------	------------	-------------

- Cache line size determines how many bits in word field (ex: 32 bytes => w = 5)
- **Not shown in the mapping structure are the “ways:”** how many cache lines are in one set. Typical are 2, 4, 8 way caches
- So a 2-way set associative cache with 4096 lines has 2048 sets, requiring 11 bits for the set field
- So for memory address 4C0180F7:
 4C0180F7 = 0100 1100 0000 0001 1000 0000 1111 0111
 0100110000000001 10000000111 10111
 tag (16 bits) set (11 bits) word (5 bits)
- Tags are much smaller than fully associative memories and comparators for simultaneous lookup are much less expensive
- To determine cache hit
 1. Compute set number from address
 2. examine tag field of ALL cache lines IN THAT SET simultaneously against tag field of memory address
 3. Note that for a k-way set associative cache, a memory address can be in one of only k lines