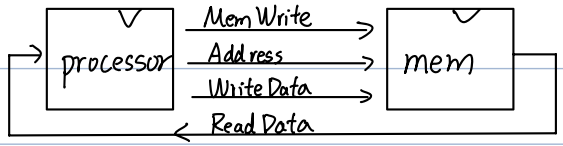
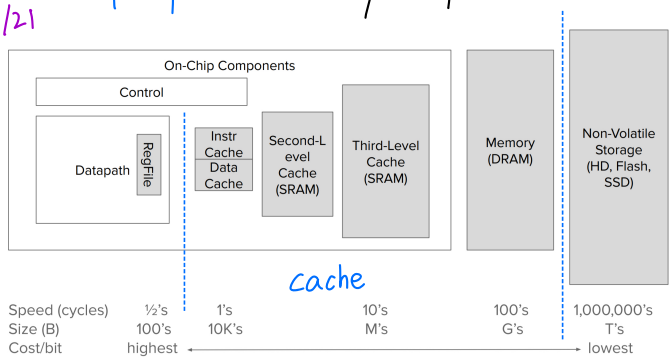
Mem physically separate from processor. Slow :(



To keep up: hierarchy (speed vs size)

11/21



Temporal locality if data recently used → likely used in future

Spacial ~ nearby data used in future

Performance Hit — data found in that level of mem.

Miss — X → next level

$AMAT_i$ (avg mem access time) $= \text{AccessTime}_i + \text{MissRate}_i \cdot AMAT_{i+1}$ (recursive hierarchy)

E. 2000 s/l, 1250 in cache, 750 not, $\text{AccTime}_{cache} = 1$ cycle; $\text{AccTime}_{mem} = 100$ cycle

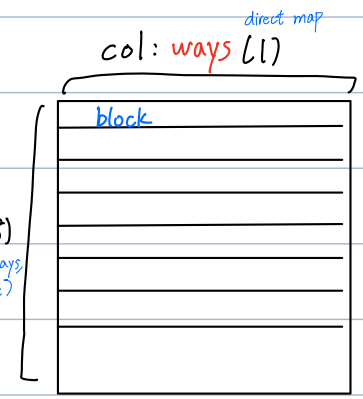
$1 + \frac{750}{2000} \cdot 100 = 38.5$ cycles

Cache: recent (or neighboring) data (too old → replaced)

addressed w/ simple address hash

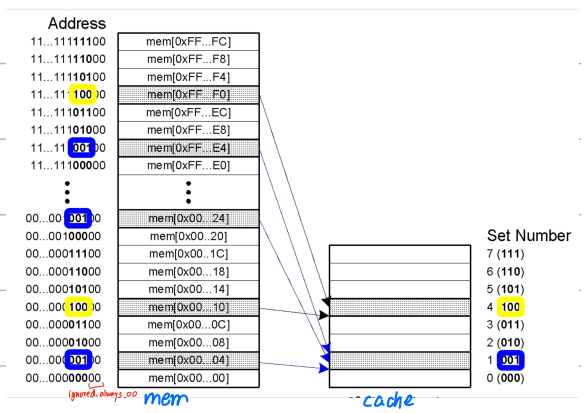
Def. Capacity (C): # bytes = Bb | 8 words (B) (meta not included)

# blocks	B = SN	8
Block size	b (byte/block)	1 word/block
# sets	S	8
Deg. of associativity	N	1



addr $\xrightarrow[\text{hash}]{\text{map}}$ one set, index set, check all ways

Hash: pull a few (4:2) bits from addr.



V: valid bit, 1 ✓ ; Tag: remainder of addr.

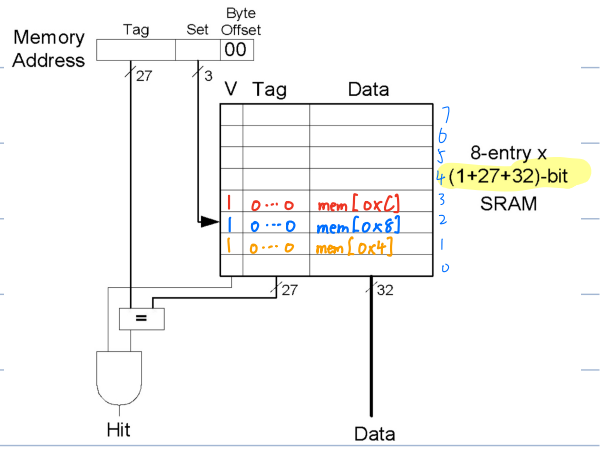
Hit = V and (Tags =)

···00000|00

E. load 0x4, 0xC, 0x8 5 times

```
        addi $t0, $0, 5
loop:   beq  $t0, $0, done
        lw   $t1, 0x4($0)    00000000 00000000 00000000 00000100
        lw   $t2, 0xC($0)    00000000 00000000 00000000 00001100
        lw   $t3, 0x8($0)    00000000 00000000 00000000 00001000
        addi $t0, $t0, -1
        j    loop
done:
```

set index



First time miss. Once in cache → saved → hit

3·1 Xs 3·(5-1) √s → 20% miss rate

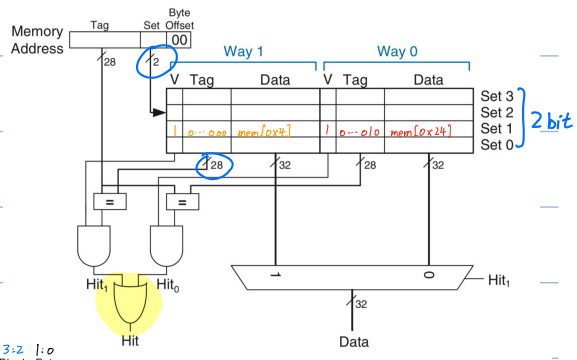
If addr. are 0x4, 0x24 → conflict misses overwrite same hash → 100% miss :(

2-way associative cache (8x1 → 4x2) idx: 3 → 2

···00000|00

E. load 0x4, 0x24 5 times

Fully-asso. → min. conflict miss, slow, $$ (more read)



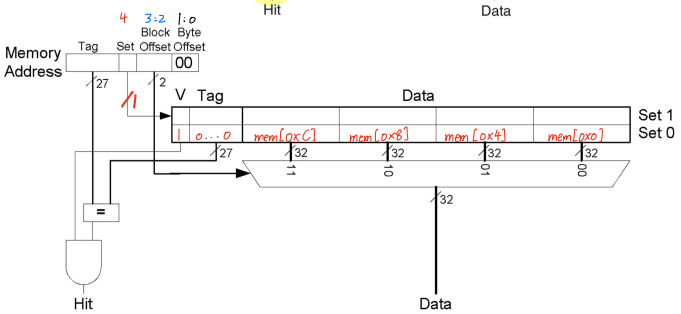
Spatial block size ↑

E. b = 4 words → B = C/b = 2 blocks (log2 4)

3b → 1 set + 2 block offset

···0000|00

E. 0x4, 0x8, 0xC



only 0x4 1 miss → loads 0x0 ~ 0xC → all hit! → miss rate = 1/15

Miss 1. Compulsory: 1st time access → 0

2. Capacity: too much data to cache ← C

3. Conflict: map to same location ← C, N

but as C↑, AMAT↓