CSE 315: Computer Organization

Sheet 8

1. Show the address decomposition of a 128kB direct-mapped cache that uses a 32-bit address and 16 bytes per block.

16byte = 4word

→ 2bit (offset)

Answer:

#bits in offset field = $\log_2 (16)$ = 4 bits

#blocks per cache = cache size/block size = 128 KB/ 16 = 2^{17} / 2^4 = 2^{13} blocks

#bits in index field = 13 bits

#bits in tag field = total address bits - #of bits in index field - #bits in offset field

=32 – 13 – 4 = 15 bits

block off byte off

31 17	16 4	3 0
TAG	Index	Offset
15 bits	13 bits	4 bits

2. Show the address decomposition of a 8-way associative cache that has 16 blocks and 32 bytes per block. Assume a 32-bit address.

32 byte → 8 word

16block / 8 way = 2set

Answer:

#bits in offset field = $\log_2 (32)$ = 5 bits

#sets = number of blocks / associativity (#ways) = 16/8 = 2 sets

1bit

#bits in set field = 1 bits

#bits in tag field = total address bits - #of bits in set field - #bits in offset field

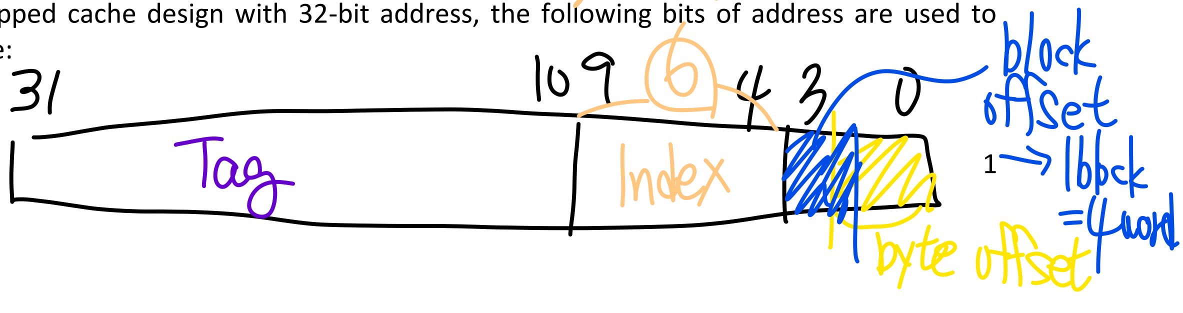
=32 – 1 – 5 = 26 bits

3 2

31 6	5	4 0
TAG	Set	Offset
26 bits	1 bit	5 bits

2^6 = 64 blocks

3. For a direct-mapped cache design with 32-bit address, the following bits of address are used to access the cache:
TAG: bits 31-10



Index: bits 9-4
Offset: bits 3:0

a. What is the cache line size in words?
b. How many entries does cache have?
c. If starting from power on, the following byte addressed cached references are recorded:
0, 4, 16, 132, 232, 160, 1024, 30, 140, 3100, 180, 2180
   i. How many blocks are replaced?
   ii. What is the hit ratio?
   iii. List the final state of the cache, with each valid entry represented as a record of <index, tag, data>.

Answer:

a. Since Offset field is 4 bits, then cache line/block size is 16 bytes; i.e. 4 words
b. Since Index field is 6 bits, then number of entries/blocks is 64 blocks
c. Block address = referenced address / block size in bytes
Line/block ID = Number of block in cache = block address mod (#blocks in cache)

Address	0	4	16	132	232	160	1024	30	140	3100	180	2180
Line ID	0	0	1	8	14	10	0	1	8	1	11	8
Hit/miss	M	H	M	M	M	M	M	H	H	M	M	M
Replace	N	N	N	N	N	N	Y	N	N	Y	N	Y

i. Number of blocks replaced = 3
ii. Hit ratio = number of hits/total number of accesses = 3/ 12 = 0.25
iii. Final state: <index, tag, data>. (The index and tag are shown in binary)
<000000, 0001, mem[1024]>
<000001, 0011, mem[3088]>
<001011, 0000, mem[176]>
<001000, 0010, mem[2176]>
<001110, 0000, mem[224]>
<001010, 0000, mem[160]>

4. Assume the following 10-bit address sequence generated by the processor:

Access #	0	1	2	3	4	5	6	7
Address in binary	10001101	10110010	10111111	10001100	10011100	11101001	11111110	11101001
TAG								
SET								
OFFSET								

The cache uses 4 bytes per block. Assume a 2-way set associative cache design that uses the LRU algorithm (with a cache that can hold a total of 4 blocks). Assume that the cache is initially empty. First determine the TAG, SET, BYTE OFFSET fields and fill in the table above. For each access, show TAG stored in cache, determine the LRU cache block, and HIT/MISS information for each access. Calculate Hit ratio for this access sequence.

Answer:

Since there are 4 bytes per block, then offset is 2-bit field.
Number of set = Number of blocks/associativity = 4/2 = 2 sets that requires 1-bit SET field
TAG field = total address bits - #of bits in SET field - #bits in OFFSET field
= 10 - 1 – 2 = 7 bits

Access #	0	1	2	3	4	5	6	7
Address in binary	10001101	10110010	10111111	10001100	10011100	11101001	11111110	11101001
TAG	10001	10110	10111	10001	10011	11101	11111	11101
SET	1	0	1	1	1	0	1	0
OFFSET	01	10	11	00	00	01	10	01

TAG content of cache after each access:

Access 0		
	Block 0	Block 1
Set 0		
Set 1	10001	

Access 1		
	Block 0	Block 1
Set 0	10110	
Set 1	10001	

Access 2		
	Block 0	Block 1
Set 0	10110	
Set 1	10001	10111

After access 2, block 0 of set 1 is LRU

Access 3		
	Block 0	Block 1
Set 0	10110	
Set 1	10001	10111

At access 3, a Hit occurs on block 0 of set 1, hence block 1 of set 1 is LRU

Access 4		
	Block 0	Block 1
Set 0	10110	
Set 1	10001	10011

Access 5		
	Block 0	Block 1
Set 0	10110	11101
Set 1	10001	10011

At access 4, block 1 of set 1 is replaced and block 0 of set 1 is LRU
After access 5, block 0 of set 0 is LRU

Access 6		
	Block 0	Block 1
Set 0	10110	11101
Set 1	11111	10011

Access 7		
	Block 0	Block 1
Set 0	10110	11101
Set 1	11111	10011

At access 6, block 0 of set 1 is replaced and block 1 of set 1 is LRU
At access 7, hit occurs on block 1 of set 0

Total number of hits during the above 8 accesses sequence = 2 hits
Hit ratio = number of hits/total number of accesses = 2/8 = 0.25

5. If 13% of instructions executed are stores, and cache write miss penalty is 10 cycles, what is the effective CPI if the processor handles write hits with a write-through scheme? Consider that the CPI without cache misses is 1.2. Account on effect of adopting write-through scheme on performance.

Answer:

- CPI = 1.2 + 10*0.13 = 2.5

- This reduces performance by more than a factor of two.