JOINT INSTITUTE
交大密西根学院

Ve370 Introduction to Computer Organization

Homework 7

Assigned: November 23, 2021

Due: 2:00pm on November 30, 2021

Submit a PDF file on Canvas

1. (20 points) For a 2-way set associative cache with a 32-bit address and write back mechanism, the partition of the 32 bits are as follows:
   - Offset: bit 6 to 0
   - Index: bit 11-7

   (1) What is the size of the cache? (5 points)

   Answer:

   Indexes are for sets: $2^5 = 32$ sets

   2-way associative: 2 * 32 = 64 blocks

   Offset includes 5 bits of word offset and 2 bits of byte offset, so

   Block size = 2^5 words * 2^2 bytes = 32 words * 4 bytes = 128 bytes

   Space for data in the cache = 64 blocks * 128 bytes = 8192 bytes * 8 bits = 65536 bits

   Tag size = 32-5-7 = 20 bits

   Space for tag in the cache = 64 blocks * 20 bits = 1280 bits

   Assuming a valid bit and a dirty bit for each block, size for status bits=64 blocks*2=128 bits

   Total size = 65536 + 1280 + 128 = 66944 bits

   Starting from power on, the following byte addresses were used to access the cache memory: 0, 4, 20, 136, 232, 164, 1024, 30, 140, 3100, 176, 2180

   (2) What is the hit ratio? (5 points)

   Answer:

   Hit ratio = 7/12 = 58.3%

   (3) Show the final state of the cache, with each valid line represented as <index, tag, data>. (10 points)

Hit/Miss	Byte Address	Binary	Tag (20 bits)	Set Index	Word Offset	Byte Offset
M	0	0x0000_0000	0	00000	00000	00
H	4	0x0000_0004	0	00000	00001	00
H	20	0x0000_0014	0	00000	00101	00
M	136	0x0000_0088	0	00001	00010	00
H	232	0x0000_00E8	0	00001	11010	00
H	164	0x0000_00A4	0	00001	01001	00
M	1024	0x0000_0400	0	01000	00000	00
H	30	0x0000_001E	0	00000	00111	10
H	140	0x0000_008C	0	00001	00011	00
M	3100	0x0000_0C1C	0	11000	00111	00
H	176	0x0000_00B0	0	00001	01100	00
M	2180	0x0000_0884	0	10001	00001	00

Set Index	Valid	Dirty	Tag	Data (128 bytes or 32 words)
00000	1		0	Mem[0] ~ mem[127]
00001	1		0	Mem[128] ~ mem[256]
00010				
…				
01000	1		0	Mem[1024] ~ mem[1151]
…				
10001	1		0	Mem[2176] ~ mem[2303]
…				
11000	1		0	Mem[3072] ~ mem[3199]
…				

2. (20 points) In general, cache access time is proportional to its capacity. Assume that main memory accesses take 70 ns and that memory accesses are 36% of all instructions. The following table shows data for caches attached to each of two processors, P1 and P2.

	Size	Miss Rate	Hit Time
P1	16KB	4.3%	1.18 ns
P2	32KB	2.7%	2.22 ns

(1) Assuming that the cache hit time determines the cycle times for P1 and P2, what are their respective clock rates? (5 points)

Answer:

For P1: clock rate = 1/1.18 = 847.5 MHz

For P2: clock rate = 1/2.22 = 450.5 MHz

(2) What is the AMAT for P1 and P2? AMAT (Average Memory Access Time) is defined as follows: AMAT = Hit time + Miss rate × Miss penalty (5 points).

Answer:

For P1: AMAT = 1.18 + 4.3% * 70 = 4.19 ns

For P2: AMAT = 2.22 + 2.7% * 70 = 4.11 ns

(3) Assuming a base CPI of 1.0 without any memory stalls, what is the actual CPI for P1 and P2? Which processor is faster? (10 points)

Answer: assuming the same data for i-cache and d-cache

For P1:

Miss penalty = 70/1.18 = 60 cycles

Actual CPI = 1 + 4.3% * 60 + 36% * 4.3% * 60 = 5.77

Time to execute 1 instruction = 5.77 * 1.18 = 6.81 ns

For P2:

Miss penalty = 70/2.22 = 32 cycles

Actual CPI = 1 + 2.7% * 32 + 36% * 2.7% * 32 = 2.60

Time to execute 1 instruction = 2.60 * 2.22 = 5.77 ns

P2 is faster.

3. (60 points) Given the following byte addresses for memory access:

3, 180, 43, 3, 191, 89, 190, 14, 181, 44, 186, 252

(1) Show the final cache contents for a 3-way set associative cache with two-word blocks and a total size of 24 words. Use LRU replacement. For each reference identify the index bits, the tag bits, the offset bits, and if it is a hit or a miss. (20 points)

Byte Address	Binary	Tag	Set Index	Word Offset	Byte Offset	Hit/Miss
3	0x03	000	00	0	11	M
180	0xB4	101	10	1	00	M
43	0x2B	001	01	0	11	M
3	0x03	000	00	0	11	H

191	0xBF	101	11	1	11	M
89	0x59	010	11	0	01	M
190	0xBE	101	11	1	10	H
14	0x0E	000	01	1	10	M
181	0xB5	101	10	1	01	H
44	0x2C	001	01	1	00	H
186	0xBA	101	11	0	10	H
252	0xFC	111	11	1	00	M

Set Index	Valid	Tag	Word0 (byte address in decimal)	Word1 (word address)
00	1	000	Mem[0~3]	Mem[4~7]
01	1	001	Mem[40~43]	Mem[44~47]
	1	000	Mem[8~11]	Mem[12~15]
10	1	101	Mem[176~179]	Mem[180~183]
11	1	101	Mem[184~187]	Mem[188~191]
	1	010	Mem[88~91]	Mem[92~95]
	1	111	Mem[248~251]	Mem[252~255]

(2) Show the final cache contents for a fully associative cache with one-word blocks and a total size of 8 words. Use LRU replacement. For each reference identify the index bits, the tag bits, and if it is a hit or a miss. (20 points)

Byte Address	Binary	Tag	Byte Offset	Hit/Miss
3	0x03	000000	11	M
180	0xB4	101101	00	M
43	0x2B	001010	11	M
3	0x03	000000	11	H
191	0xBF	101111	11	M
89	0x59	010110	01	M
190	0xBE	101111	10	H
14	0x0E	000011	10	M
181	0xB5	101101	01	H
44	0x2C	001011	00	M
186	0xBA	101110	10	M
252	0xFC	111111	00	M

Valid	Tag	Data (byte address in decimal)
1	000000	Mem[0-3]
1	101101	Mem[180-183]

1	111111	Mem[252-255]
1	101111	Mem[188-191]
1	010110	Mem[88-91]
1	000011	Mem[12-15]
1	001011	Mem[44-47]
1	101110	Mem[184-187]

(3) What is the miss rate for a fully associative cache with two-word blocks and a total size of 8 words, using LRU replacement? What is the miss rate using MRU (most recently used) replacement? Finally what is the best possible miss rate for this cache, given any replacement policy? (20 points)

Byte Address	Binary	Tag	Word Offset	Byte Offset	H/M (LRU)	H/M (MRU)	H/M (any)
3	0x03	00000	0	11	M	M	M
180	0xB4	10110	1	00	M	M	M
43	0x2B	00101	0	11	M	M	M
3	0x03	00000	0	11	H	H	H
191	0xBF	10111	1	11	M	M	M
89	0x59	01011	0	01	M	M	M (replace 0-7)
190	0xBE	10111	1	10	H	M	H
14	0x0E	00001	1	10	M	M	M (replace 88-95)
181	0xB5	10110	1	01	M	H	H
44	0x2C	00101	1	00	M	H	H
186	0xBA	10111	0	10	H	M	H
252	0xFC	11111	1	00	M	M	M

LRU

Valid	Tag	Word0 (byte address in decimal)	Word0 (byte address in decimal)
1	10110	Mem[176-179]	Mem[180-183]
1	00101	Mem[40-43]	Mem[44-47]
1	11111	Mem[248-251]	Mem[252-255]
1	10111	Mem[184-187]	Mem[188-191]

MRU

Valid	Tag	Word0 (byte address in decimal)	Word0 (byte address in decimal)
1	00000	Mem[0-3]	Mem[4-7]
1	10110	Mem[176-179]	Mem[180-183]
1	11111	Mem[248-251]	Mem[252-255]
1	00001	Mem[8-11]	Mem[12-15]