Homework12

Tuesday, May 6, 2014 4:01 PM

yxu66

5.2 Caches are important to providing a high-performance memory hierarchy to processors. Below is a list of 32-bit memory address references, given as word addresses.3, 180, 43, 2, 191, 88, 190, 14, 181, 44, 186, 253.

5.2.3. For each of the references, find the binary address, the tag, and index, if the cache is direct-mapped with 4 4-word blocks. Which of the caches in 5.2.1-3 has the best miss rate?

5.2.1

Word Address	Binary Address	Tag	Index	Hit/Miss
3	0000 0011	0	3	M
180	1011 0100	11	4	M
43	0010 1011	2	11	M
2	0000 0010	0	2	M
191	1011 1111	11	15	M
88	0101 1000	5	8	M
190	1011 1110	11	14	M
14	0000 1110	0	14	M
181	1011 0101	11	5	M
44	0010 1100	2	12	M
186	1011 1010	11	10	M
253	1111 1101	15	13	M

Miss rate = 12/12 = 1

5.2.2

Word Address	Binary Address	Tag	Index	Hit/Miss
3	0000 0011	0	1	M
180	1011 0100	11	2	M
43	0010 1011	2	5	M
2	0000 0010	0	1	H
191	1011 1111	11	7	M
88	0101 1000	5	4	M
190	1011 1110	11	7	H
14	0000 1110	0	7	M
181	1011 0101	11	2	H
44	0010 1100	2	6	M
186	1011 1010	11	5	M
253	1111 1101	15	6	M

Miss rate = 9/12 = 0.75

5.2.3
Word offset = 2 bit
Index = 2 bit
Tag = 32-2-2-2=26 bit

Word address	Binary address	Tag	index	Hit/Miss
3	0000 0011	0000	00	M
180	1011 0100	1011	01	M
43	0010 1011	0010	10	M
2	0000 0010	0000	00	H
191	1011 1111	1011	11	M
88	0101 1000	0101	10	M
190	1011 1110	1011	11	H
14	0000 1110	0000	11	M
181	1011 0101	1011	01	H
44	0010 1100	0010	11	M
186	1011 1010	1011	10	M
253	1111 1101	1111	11	M

Miss rate = 9/12 = 0.75
So 5.2.2 and 5.2.3 has the same miss rate, but 5.2.3 has small block size, so it will have lower miss rate with other reference.

5.2.6 [15] <§5.3> The formula shown in Section 5.3 shows the typical method to index a direct-mapped cache, specifically (Block address) modulo (Number of blocks in the cache). Assuming a 32-bit address and 1024 blocks in the cache, consider a different indexing function, specifically (Block address[31:27] XOR Block address[26:22]). Is it possible to use this to index a direct-mapped cache? If so, explain why and discuss any changes that might need to be made to the cache. If it is not possible, explain why.

Yes, it's possible. But since use Block address[31:27] XOR Block address[26:22], it will lose information, so it will need to add more bits to tag to distinguish the address in the cache.

5.5.4-6. The "miss latency" is the time it takes to transfer a line (or block) from memory to cache.
5.5 Media applications that play audio or video files are part of a class of workloads called "streaming" workloads; i.e., they bring in large amounts of data but do not reuse much of it. Consider a video streaming workload that accesses a 512 KiB working set sequentially with the following address stream:
0, 2, 4, 6, 8, 10, 12, 14, 16, ...
Cache block size (B) can affect both miss rate and miss latency. Assuming a 1-CPI machine with an average of 1.35 references (both instruction and data) per instruction, help find the optimal block size given the following miss rates for various block sizes.

8：4%	**16：3%**	**32：2%**	**64：1.5%**	**128：1%**

5.5.4 [10] <§5.3> What is the optimal block size for a miss latency of 20×B cycles?
Miss cycles per reference = cycles per miss * misses per reference

Block size	Cycles per miss	Misses per reference	Miss cycles per reference
8	**20*8 = 160**	**4%**	**160*4% = 6.4**
16	20*16 = 320	3%	320*3% = 9.6
32	20*32 = 640	2%	640*2% = 12.8
64	20*64 = 1,280	1.5%	1280*1.5% = 19.2
128	20*128 = 2,560	1%	2560*1% = 25.6

So 16 blocks is the best.

5.5.5 [10] <§5.3> What is the optimal block size for a miss latency of 24+B cycles?

Miss cycles per reference = cycles per miss * misses per reference

Block size	Cycles per miss	Misses per reference	Miss cycles per reference
8	24+8 = 32	4%	32*4% = 1.28
16	24+16 = 40	3%	40*3% = 1.2
32	**24+32 = 56**	**2%**	**56*2% = 1.12**
64	24+64 = 88	1.5%	88*1.5% = 1.32
128	24+128 = 152	1%	152*1% = 1.52

So 32 blocks is the best.

5.5.6 [10] <§5.3> For constant miss latency, what is the optimal block size?

Assume the miss latency = n

Block size	Cycles per miss	Misses per reference	Miss cycles per reference
8	n	4%	0.04n
16	n	3%	0.03n
32	n	2%	0.02n
64	n	1.5%	0.015n
128	**n**	**1%**	**0.01n**

So 128 blocks is the best.

Describe what happens after the following events occur.

1. **The processor attempts to read the word at address 0x1000 D000 and the valid bit is 0.**
   Since the valid bit is 0, it means it contains invalid data. So the tag should be ignored for such entries and there cannot be a match for this block. So cache signal is miss. Then:
   Report "miss" to processor.
   Processor request word from memory.
   (In-order processor will stall, out-of-order process can allow execution of instructions while waiting for a cache miss)
   Memory sends line to cache.
   Cache appends valid bit.

2. **The processor attempts to read the word at address 0x1000 D000 and the tags don't match.**
   If the tags don't match, which means the cache signal is a miss. Then:
   Report "miss" to processor.
   Processor request word from memory.
   (In-order processor will stall, out-of-order process can allow execution of instructions while waiting for a cache miss)
   Memory sends line to cache.
   Cache appends tag bit.

3. **The processor attempts to write the word at address 0x1000 D000 and the tags don't match. The cache is write-through and write-allocate.**
   The tags don't match, then the cache signal is miss. Then:
   Report "miss" to processor.
   Processor request word from memory.
   Load line into cache and proceed as in a hit
   Write the data into both the memory and the cache

4. **The processor attempts to write the word at address 0x1000 D000 and the tags don't match The cache is write-through and no-write-allocate.**
   The tags don't match, then the cache signal is miss. Then:
   Report "miss" to processor.
   Processor request word from memory.
   Write the data into memory.

5. **The processor attempts to write the word at address 0x1000 D000 and the valid bit is 0. The cache is write-back and write-allocate.**
   The valid bit is 0, then the cache signal is miss. Then:
   Report "miss" to processor.
   Processor request word from memory.
   Check current line in the computed cache index to see if it needs to be evicted.
   If not, load required line and proceed as with a hit, write the data into cache.
   Else, write the data into memory.