CS202 HW4

Problem 1 20points

For a 256 Byte data cache, 4 word per block, direct mapped design with a 32-bit address, beginning from power on, the following byte-addressed cache references are recorded.

Hex	00	04	10	84	E8	A0	400	1E	8C	C1C	B4	884

(a) Calculate the range of bits used for Tag, index and offset (for example, offset: 1-0, index: 3-2, etc)

(b) For each reference, list (1) its tag, index, and offset, (2) whether it is a hit or a miss, and (3) which bytes were replaced (if any). Using the table format shown below.

Address	Binary	Tag	Index	Offset	Hit/Miss	Replaced
0x00	0000_0000	0x0	0x0	0x0	Miss	NA
0x04	...	...	...	...	Hit	NA
...	...	...	...	...	...	...

(c) What is the hit ratio?

(d) List the final state of the cache, with each valid entry represented as a record of . For example, <0, 3, Mem[0x00]-Mem[0x1F]>.

Problem 2 25 points

This exercise examines the effect of different cache designs, specifically comparing associative caches to the direct-mapped caches. For a three-way set associative cache with two-word blocks and a total size of 48 words. Refer to the sequence of word address shown below. 0x03, 0xB4, 0x2B, 0x02, 0xBE, 0x58, 0xBF, 0x0E, 0x1F, 0xB5, 0xBF, 0xBA, 0x2E, 0xCE .

(a) Calculate the range of bits used for Tag, index and offset (for example, offset: 1-0, index: 3-2, etc)

(b) 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. For invalid blocks use NA as content. Then show the Final cache contents. Using the table format shown below.

Address	Binary	Tag	Index	Offset	Hit/Miss
0x00	0000_0000	0x0	0x0	0x0	Miss
0x04	...	...	...	...	Hit
...	...	...	...	...	...

Index	Way0	Way1	Way2
0	Mem[0x58-0x59]	NA	...
1	...	...	...

(c) For a fully associative cache with one word blocks and a total size of 8 words, using the given sequence, 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. Using the table format shown below.

Address	Tag	Index	Offset	Hit/Miss	Way
0x03	...	...	...	Miss	0
...	...	...	...	...	...

Problem 3 20points

Multilevel caching is an important technique to overcome the limited amount of space that a first-level cache can provide while still maintaining its speed. Consider a processorwith the following parameters and the miss rate are global miss:

Base CPI,No Memeory Stalls	Processor Speed	Main Memory Access Time	First-level Cache Rate Per Instruction**	Second-Level Cache, Direct-Mapped Speed	Miss Rate with Second-Level Cache, Direct-Mapped	Second-Level Cache, Eight-Way set Associative Speed	Miss Rate with Second-Level Cache, Eight-Way set Associative
1.5	4GHz	100ns	6%	15 cycels	4%	30 cycles	1.5%

Calculate the CPI for the processor in the table using:

a) only a first-level cache

b) a second-level direct-mapped cache, and

c) a second-level eight-way set associative cache.

d) How do the CPI value calculated in b) change if all miss rates in table become local miss rate?

Problem 4 15 points

This exercise examines the single error correcting, double error detecting (SEC/DED) Hamming code.

a) What is the minimum number of parity bits required to protect a 128-bit word using the SEC/DED code?

b) Consider a SEC code that protects 8 bit words with 4 parity bits. If we read the value 0x375, is there an error? If so, correct the error.

Problem 5 20 points

Virtual memory uses a page table to track the mapping of virtual addresses to physical addresses (in unit of byte). This exercise shows how this table must be updated as addresses are accessed. The following data constitute a stream of virtual byte addresses as seen on a system. Assume 4 KiB pages, a four entry fully associative TLB, and true LRU replacement. If pages must be brought in from disk, increment the next largest page number. i.e. If the current maximum physical page number is 12, then the next physical page brought in from disk has a page number of 13.

Initial TLB State:

Valid	Tag	Physical Page Number	Time Since Last Access
1	0xB	12	4
1	0x7	4	1
1	0x3	6	3
0	0x4	9	7

Initial Page Table State:

Valid	Physical Page or in Disk
1	5
0	Disk
0	Disk
1	6
1	9
1	11
0	Disk
1	4
0	Disk
0	Disk
1	3
1	12

Sequence of Virtual Address Accesses:

Hex	0x123D	0x08B3	0x365C	0x871B	0xBEE6	0x3140	0x2330

When a TLB entry is replaced, the new entry's "Time Since Last Access" is initialized to 1. For example, if the first entry in the table below is replaced:

Valid	Tag	Physical Page Number	Time Since Last Access
1	0xB	12	4
1	0x7	4	1

Valid	Tag	Physical Page Number	Time Since Last Access
1	0x1	13	1
1	0x7	4	2

a) For each access shown above, fill up the following table to show the status of each access, and then show the final state of the TLB and Page table. Using the table format shown below.

Address	Virtual Page	TLB Hit/Miss	Page Table Hit/Miss	Page Fault
0x123D	...	...	Miss	Yes
0x08B3	...	...	Hit	No
...	...	...	...	...

Valid	Tag	Physical Page Number	Time Since Last Access
1	...	...	1
...	...	...	...

Valid	Physical Page or in Disk
1	5
...	...

b) This time use 4 KiB pages and a two-way set associative TLB(Initial State shown below). Fill up the following table to show the status of each access, and then show the final state of the TLB. Using the table format shown below.

Initial TLB State:

Valid	Set	Tag	Physical Page Number	Time Since Last Access
1	0	0x2	9	7
1	0	0x5	3	3
1	1	0x5	12	4
1	1	0x3	4	1

Address	Virtual Page	Index	Tag	TLB Hit/Miss	Page Table Hit/Miss	Page Fault
0x123D	...	...	...	Miss	Miss	Yes
...	...	...	...	...	...	...

Valid	Set	Tag	Physical Page Number	Time Since Last Access
1	0	...	...	...
1	0	...	...	...
1	1	...	...	...
...	...	...	...	...