EE 457 Homework 6

Redekopp

Name: _____ Score: _____/ 100_

Cache Memory

1.) (38 pts.) Complete the table below. All processors are byte-addressable..

Processor	Mapping Technique	Addr. Space Size (B)	Cache Size	Block Size	Tag Field	Block or Set Field (as appropriate)	Word Field	Byte Field
16-bit Data 20-bit Address	Direct	1MB	8KB	2 words (4 bytes)	7-bit wide A19-13	11-bit wide A12-A2	1-bit wide A1	A0 (/BE1,/BE0)
	Fully Associative				18-bit wide A19-A2		1-bit wide A1	A0 (/BE1,/BE0)
	2-way Set Associative				8-bit wide A19-A12	10-bit wide A11-A2	1-bit wide A1	A0 (/BE1,/BE0)
16-bit Data 20-bit Address	Direct		16KB	4 words (8 bytes)				
	Fully Associative							
	4-way Set Associative							
32-bit Data 20-bit Address	Direct		32KB	2 words (8 bytes)				
	Fully Associative							
	2-way Set Associative							
32-bit Data 32-bit Address	Direct		128 KB	4 words (16 bytes)				
	Fully Associative							
	4-way Set Associative							
64-bit Data 32-bit Address	Direct		512 KB	2 words (16 bytes)				
	Fully Associative							
	4-way Set Associative							
$(2^w x 8)$-bit Data m-bit Address	Direct	$2^m B$	$2^c B$	2^b words (2^{b+w} Bytes)				
	Fully Associative							
	2^k-way Set Associative							

These three problems come from 7.7(7.1), 7.8(7.2), 7.20(7.3) from Hennessy and Patterson, CO&D 2nd (1st) editions.

2.) (10 pts.) Consider a series of address references given as **word** addresses: 1, 4, 8, 5, 20, 17, 19, 56, 9, 11, 4, 43, 5, 6, 9, 17. Assuming a direct mapped cache with 16 one-word blocks that is initially empty, label each access in the list below as a hit or miss and show what addresses are present in cache at the end of the trace.

Use the diagram on the next page to show what addresses are present in the cache at the end of the trace.

3.) (10 pts.) Use the series of references given in the previous exercise to show the hits and misses and what addresses are present in cache at the end of the trace assuming a direct-mapped cache with four-word blocks and a *total size* of 16 words.

4.) (10 pts.) Use the series of references given in the previous exercise to show the hits and misses and what addresses are present in cache at the end of the trace assuming a 2-way set-associative cache with one-word blocks and a *total size* of 16 words. Assume LRU replacement.

Dec. Address	**Address in binary**	**Problem 2 (Direct-Mapped, 1-word blocks) [Hit / Miss]**	**Problem 3 (Direct-Mapped, 4-word blocks) [Hit / Miss]**	**Problem 4 (2-way Set Associative, 1-word blocks) [Hit / Miss]**
1	0000 0001			
4	0000 0100			
8	0000 1000			
5	0000 0101			
20	0001 0100			
17	0001 0001			
19	0001 0011			
56	0011 1000			
9	0000 1001			
11	0000 1011			
4	0000 0100			
43	0010 1011			
5	0000 0101			
6	0000 0110			
9	0000 1001			
17	0001 0001			

Worksheet for problems 2, 3, and 4 Cache Total Size = 16 words

Address sequence: 1, 4, 8, 5, 20, 17, 19, 56, 9, 11, 4, 43, 5, 6, 9, 17

Direct Mapped Cache
Block size = 1 word

Block 0	
Block 1	
Block 2	
Block 3	
Block 4	
Block 5	
Block 6	
Block 7	
Block 8	
Block 9	
Block 10	
Block 11	
Block 12	
Block 13	
Block 14	
Block 15	

Direct Mapped Cache
Block size = 4 words

Block 0				
Block 1				
Block 2				
Block 3				

2-way Set Associative Cache
Block size = 1 word

	Set 0	
	Set 1	
	Set 2	
	Set 3	
	Set 4	
	Set 5	
	Set 6	
	Set 7	

5.) (6 pts.) Consider the 64-bit Intel i860 which has a 32-bit logical address and has 32 physical pins labeled A31-A3, /BE[7:0]. Assume a little-endian system.

Assume that the 64-bit word at the address (location) 0x73468108 in the memory is 0x123456789ABCDEF0

a. When the processor tries to access (read) the 16-bit word at the address 0x7346810a, what 16-bit word (value) will it retrieve?

b. During the 16-bit word access above, what signal values (in binary) would you see on the address line A3 and /BE[7:0] (note /BE[0] pertains to LS-data byte].

A3 = _____ /BE[7:0] (show all 8-bits): _____

6.) (6 pts.) Consider the cache design of the 8-bit processor Intel 8088 with 20-bit address and 8-bit data) utilizing a 4-way set-associative mapping with 512 sets and 2-byte blocks.

a. Total size of the cache in bytes: _____

b. How many comparators and of what size (width) are required to determine HIT/MISS (include the valid bit in your answer)?

c. How many TAG RAMs do you need and of what size? (e.g. 3 TAG RAMs of 1024x6)

d. How many DATA RAMs do you need and of what size?

7.) (6 pts.) Consider a cache memory system with 128 bytes of main memory. Assume 1 word = 1 byte (8-bit data bus). The CPU has a 7-bit address, a block size of 2 bytes and a set size of 4 blocks. The cache is 32-bytes total broken into 16 blocks.

a. How many comparators and of what size (width) are required to determine HIT/MISS (include the valid bit in your answer)?

b. How many TAG RAMs do you need and of what size? (e.g. 3 TAG RAMs of 1024x6)

c. What address bits (i.e. subset of A6-A0) would you use as the address of the TAG RAMs?

d. What address bits (i.e. subset of A6-A0) would you feed into the comparators to compare with the output of the TAG RAM?

e. How many DATA RAMs do you need and of what size?

f. What address bits (i.e. subset of A6-A0) would you feed into the address inputs of the data RAMs?

8.) (14 pts.) Exercises 5.3.5 in CO&D, 4th Ed. page 551. Consider a direct-mapped cache of 64 KB and a block size of 2 words (1 word = 4 bytes). Generate a series of read requests (address sequence) that will have a lower miss rate on a 2 KB 2-way set associative cache with the same block size than this 64 KB direct mapped cache? Then generate a series of read requests (address sequence) that will have a lower miss rate in the 64KB direct-mapped cache. To make answers more standard, use address that are as small as possible to make your case (i.e. make unneeded tag bits of your address 0 [you should just need to change a few of the LS tag bits] and make the word portion [i.e. LSB] 0 wherever possible).

First perform the address breakdown for each cache:

Cache	Tag	Block or Set	Word	Byte
64 KB, direct, 2 w/block	A19-A____	A____-A3	A2	A1-A0
2 KB, 2-way SA, 2 w/block	A19-A____	A____-A3	A2	A1-A0

For the first question, find a 3 address sequence. Write out your address to **20-bits**. (e.g. 0x00800)

	20-bit address in hex.
1	
2	
3	

For the second question, find a 4 address sequence. Write out your address to **20-bits**. (e.g. 0x00800)

	20-bit address in hex.
1	
2	
3	
4	

