Homework 3

Help Center

Cache Organization and Indexing

Assigned	Wednesday, July 30, 2014
Suggested Completion Date	*Wednesday, August 6, 2014*

Introduction

The purpose of written homework assignments is to get you thinking about the topics being covered in lecture and in readings in the textbook which are not represented in the hands-on, programming lab assignments. These written assignments also better prepare you for course examinations. It is worth noting that the book contains many practice problems similar to the problems we ask on these written assignments. The solutions for those practice problems are located at the end of each chapter and should give you a feel for the kind of answers we expect you to turn in for these kind of assignments.

Logistics

These written homeworks will not turned in for credit like the programming lab assignments. There also won't be solutions provided, but the forums are open to all discussion about the homework. We encourage you to complete the homework to the best of your ability and then discuss your findings/questions with your peers on the forums.

Questions

Answer the following problems:

1. Practice Problem 6.10. (pg 598)
   The following table gives the parameters for a number of different caches. For each cache, determine the number of cache sets (*S*), tag bits (*t*), set index bits (*s*), and block offset bits (*b*).

Cache	***m***	***C***	***B***	***E***	***S***	***t***	***s***	***b***
1.	32	1024	4	1	______	______	______	______
2.	32	1024	8	4	______	______	______	______
3.	32	1024	32	32	______	______	______	______
4.*	38	32768	64	8	______	______	______	______
5.**	48	524288	64	16	______	______	______	______

* Intel Xeon L1 data cache

** AMD Opteron 6168 L2 unified cache

2. Homework Problem 6.27 (pg 632)

The following table gives the parameters for a number of different caches. Your task is to fill in the missing fields in the table. Recall that *m* is the number of physical address bits, *C* is the cache size (number of data bytes), *B* is the block size in bytes, *E* is the associativity, *S* is the number of cache sets, *t* is the number of tag bits, *s* is the number of set index bits, and *b* is the number of block offset bits.

Cache	*m*	*C*	*B*	*E*	*S*	*t*	*s*	*b*
1.	32	______	8	1	______	21	8	3
2.	32	2048	______	______	128	23	7	2
3.	32	1024	2	8	64	______	______	1
4.	32	1024	______	2	16	23	4	______

3. Practice Problem 6.13 (pg 609)

The problems that follow will help reinforce your understanding of how caches work. Assume the following:

- The memory is byte addressable.
- Memory accesses are to **1-byte words** (not to 4-byte words).
- Addresses are 13 bits wide.
- The cache is two-way set associative (*E*=2), with a 4-byte block size (*B*=4), and eight sets (*S*=8).

The contents of the cache are as follows, with all numbers given in hexadecimal notation.

2-way set associative cache

	Line 0						Line 1					
Set index	**Tag**	**Valid**	**Byte 0**	**Byte 1**	**Byte 2**	**Byte 3**	**Tag**	**Valid**	**Byte 0**	**Byte 1**	**Byte 2**	**Byte 3**
0	09	1	86	30	3F	10	00	0	__	__	__	__
1	45	1	60	4F	E0	23	38	1	00	BC	0B	37
2	EB	0	__	__	__	__	0B	0	__	__	__	__
3	06	0	__	__	__	__	32	1	12	08	7B	AD
4	C7	1	06	78	07	C5	05	1	40	67	C2	3B
5	71	1	0B	DE	18	4B	6E	0	__	__	__	__
6	91	1	A0	B7	26	2D	F0	0	__	__	__	__
7	46	0	__	__	__	__	DE	1	12	C0	88	37

The following figure shows the format of an address (one bit per box). Indicate (by labeling the diagram) the fields that would be used to determine the following:

CO The cache block offset
CI The cache set index
CT The cache tag

12	11	10	9	8	7	6	5	4	3	2	1	0

4. Practice Problem 6.14 (pg 610)

   Suppose a program running on the machine in Problem 6.13 references the 1-byte word at address `0x0E34`. Indicate the cache entry accessed and the cache byte value returned **in hex**. Indicate whether a cache miss occurs. If there is a cache miss, enter "-" for "Cache byte returned."

   A. Address format (one bit per box):

12	11	10	9	8	7	6	5	4	3	2	1	0

   B. Memory reference:

Parameter	Value
Cache block offset (CO)	`0x_____`
Cache set index (SI)	`0x_____`
Cache tag (CT)	`0x_____`
Cache hit? (Y/N)	`0x_____`
Cache byte returned	`0x_____`

5. Practice Problem 6.15 (pg 611)

   Repeat Problem 6.14 for memory address `0x0DD5`.

6. Practice Problem 6.16 (pg 611)

   Repeat Problem 6.14 for memory address `0x1FE4`.

7. Practice Problem 6.17 (pg 611)

   For the cache in Problem 6.13, list all of the hex memory addresses that will hit in set 3.

8. Homework Problem 6.28 (pg 632)

   This problem concerns the cache in Problem 6.13

   A. List all of the hex memory addresses that will hit in set 1.
   B. List all of the hex memory addresses that will hit in set 6.

Created Wed 15 May 2013 8:36 PM PDT

Last Modified Wed 16 Jul 2014 9:58 AM PDT