ECE 30
Introduction to Computer Engineering

Study Problems, Set #9
Spring 2014

1. Given the following series of address references given as word addresses: 2, 3, 11, 16, 21, 13, 64, 48, 19, 11, 3, 22, 4, 27, 6, and 11. Assuming a direct-mapped cache with 16 one-word blocks that is initially empty, label each reference in the list as a hit or a miss and show the final contents of the cache (make sure to represent the cache structure correctly; show the index for every line of the cache and include valid and tag bits as well besides the data).

Address	Index	Tag	Hit/Miss
2			
3			
11			
16			
21			
13			
64			
48			
19			
11			
3			
22			
4			
27			
6			
11			

Solution:

- index = address % 16
- tag = $\lfloor$address / 16$\rfloor$

Address	Index	Tag	Hit/Miss
2	2	0	M
3	3	0	M
11	11	0	M
16	0	1	M
21	5	1	M
13	13	0	M
64	0	4	M
48	0	3	M
19	3	1	M
11	11	0	H
3	3	0	M
22	6	1	M
4	4	0	M
27	11	1	M
6	6	0	M
11	11	0	M

Index	Valid	Tag	Data
0	1	3	48
1	0		
2	1	0	2
3	1	0	3
4	1	0	4
5	1	1	21
6	1	0	6
7	0		
8	0		
9	0		
10	0		
11	1	0	11
12	0		
13	1	0	13
14	0		
15	0		

The “Data” column contains the word address, in memory, of the content that is stored, as data, in the cache.

2. Using the series of references given in Problem 1, show the hits and misses and final cache contents for a direct-mapped cache with four-word blocks and a *total size* of 16 words (make sure to represent the cache structure correctly; again, show the index for every line of the cache and include valid and tag bits).

Address	Block offset	Index	Tag	Hit/Miss
2				
3				
11				
16				
21				
13				
64				
48				
19				
11				
3				
22				
4				
27				
6				
11				

Solution:

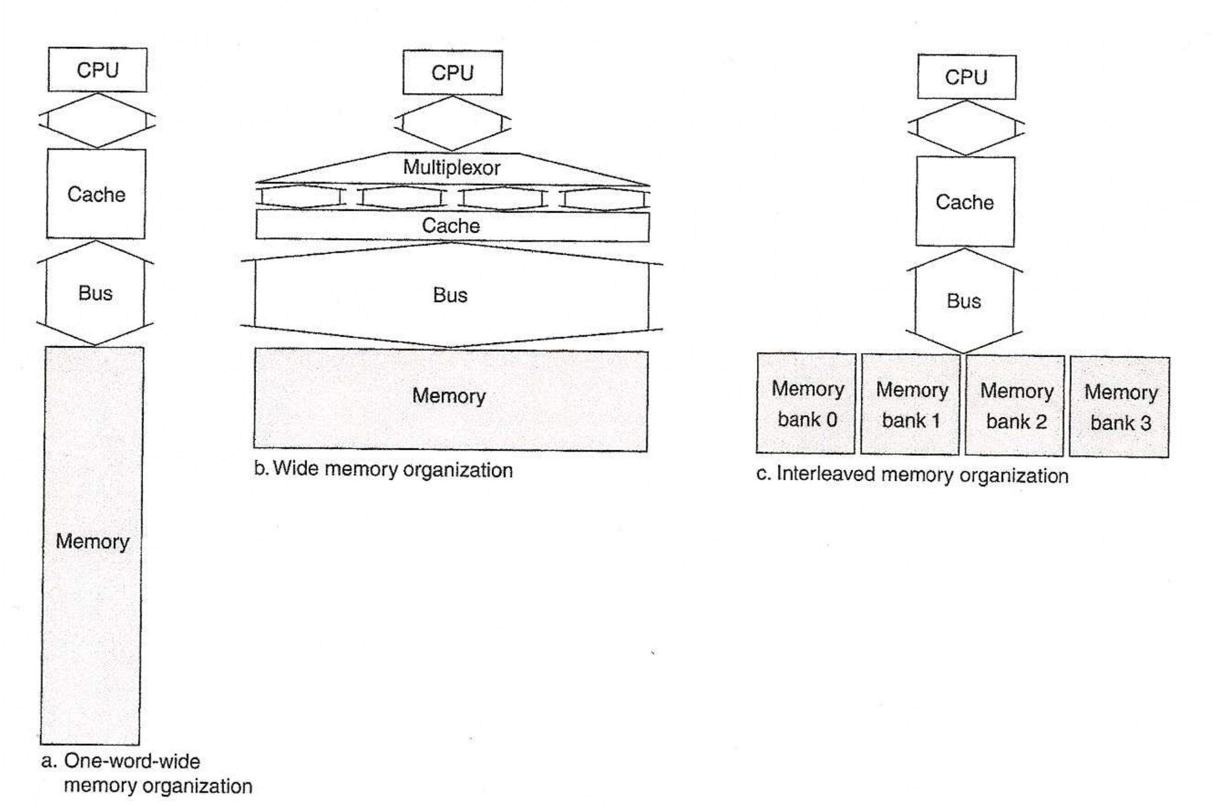
- block offset = address % 4
- index = $\lfloor(\text{address} \% 16) / 4\rfloor$
- tag = $\lfloor\text{address} / 16\rfloor$

Address	Block Offset	Index	Tag	Hit/Miss
2	2	0	0	M
3	3	0	0	H
11	3	2	0	M
16	0	0	1	M
21	1	1	1	M
13	1	3	0	M
64	0	0	4	M
48	0	0	3	M
19	3	0	1	M
11	3	2	0	H
3	3	0	0	M
22	2	1	1	H
4	0	1	0	M
27	3	2	1	M
6	2	1	0	H
11	3	2	0	M

Index	Valid	Tag	Data			
			0	1	2	3
0	1	0	0	1	2	3
1	1	0	4	5	6	7
2	1	0	8	9	10	11
3	1	0	12	13	14	15

The “Data” column contains the word address, in memory, of the content that is stored, as data, in the cache.

3. Consider a memory hierarchy using one of the three organizations for main memory shown in the following figure.



a. One-word-wide memory organization

b. Wide memory organization

c. Interleaved memory organization

Assume that the cache block size is 16 words, that the width of organization (b) in the figure is four words, and that the number of banks in organization (c) is four. If the main memory latency for a new access is 10 memory bus clock cycles and the transfer time is 1 memory bus clock cycle, what are the miss penalties for each of these organizations?

Solution:

Cache block size is 16 words.

(a) One-word wide memory.
Miss penalty = 16 × Memory access time + 1 × Transfer time (for sending the address) + 16 × Transfer time (for sending 16 words) = 177 memory bus clock cycles.

(b) Four-word wide memory.
Miss penalty = 4 × Memory access time + 1 × Transfer time (for sending the address) + 4 × Transfer time (for sending 4 sets of 4 words) = 45 memory bus clock cycles.

(c) 4 banks of one-word wide interleaved memory.
Miss penalty = 4 × Memory access time + 1 × Transfer time (for sending the address) + 16 × Transfer time (for sending 16 words) = 57 memory bus clock cycles.

4. Using the series of references given in Problem 1 and assuming a two-way set-associative cache (replacement scheme: least-recently-used) with two-word blocks and a *total size* of 16 words that is initially empty, label each reference in the table below as a hit or a miss and show the final contents of the cache (make sure to represent the cache structure correctly; show the index for every line of the cache and include all special bits as well besides the data).

Address	Block offset	Set index	Tag	Set entry	Hit/Miss
2					
3					
11					
16					
21					
13					
64					
48					
19					
11					
3					
22					
4					
27					
6					
11					

Solution:

- Two-way set associative cache with a total of 16 words.
- Two-word blocks.
- Use LRU for replacement policy.
- block offset = address % 2
- index = $\lfloor (\text{address} \% 8) / 2 \rfloor$
- tag = $\lfloor \text{address} / 8 \rfloor$

Address	Block offset	Set index	Tag	Set entry	Hit/Miss
2	0	1	0	0	M
3	1	1	0	0	H
11	1	1	1	1	M
16	0	0	2	0	M
21	1	2	2	0	M
13	1	2	1	1	M
64	0	0	8	1	M
48	0	0	6	0	M
19	1	1	2	0	M
11	1	1	1	1	H
3	1	1	0	0	M
22	0	3	2	0	M
4	0	2	0	0	M
27	1	1	3	1	M
6	0	3	0	1	M
11	1	1	1	0	M

Set Index	Entry 0				Entry 1			
	Valid	Tag	Data		Valid	Tag	Data	
			0	1			0	1
0	1	6	48	49	1	8	64	65
1	1	1	10	11	1	3	26	27
2	1	0	4	5	1	1	12	13
3	1	2	22	23	1	0	6	7

The “Data” column contains the word address, in memory, of the content that is stored, as data, in the cache.

5. Using the series of references given in Problem 1 and assuming a fully associative cache (replacement scheme: least-recently-used) with four-word blocks and a *total size* of 16 words that is initially empty, label each reference in the table below as a hit or a miss and show the final contents of the cache (make sure to represent the cache structure correctly; show the index for every line of the cache and include all special bits as well besides the data).

Address	Block offset	Tag	Set Entry	Hit/Miss
2				
3				
11				
16				
21				
13				
64				
48				
19				
11				
3				
22				
4				
27				
6				
11				

Solution:

- Fully associative cache (1 set) with a total of 16 words.
- Four words per block ⇒ 4 blocks.
- Number of tag bits = number of block-address bits.
- No index ⇒ must search every block for match.
- Each block must contain both the tag and the data bits.
- Use LRU for replacement policy.

Address	Block offset	Tag	Set Entry	Hit/Miss
2	2	0	0	M
3	3	0	0	H
11	3	2	1	M
16	0	4	2	M
21	1	5	3	M
13	1	3	0	M
64	0	16	1	M
48	0	12	2	M
19	3	4	3	M
11	3	2	0	M
3	3	0	1	M
22	2	5	2	M
4	0	1	3	M
27	3	6	0	M
6	2	1	3	H
11	3	2	1	M

Set entry	Valid	Tag	Data			
			0	1	2	3
0	1	6	24	25	26	27
1	1	2	8	9	10	11
2	1	5	20	21	22	23
3	1	1	4	5	6	7

The “Data” column contains the word address, in memory, of the content that is stored, as data, in the cache.