CS 152
Computer Systems Architecture
Winter 2015

Homework 4

Due Tuesday, March 10th, 2015, EEE 11:59PM PDF

Name	*Student ID*
Ford Tang	46564602

IMPORTANT NOTES:

No late submissions.
Please show your work. Remember that bottom line answers without proper explanations are worth ZERO points.
Remember that you are solely responsible for the answers to the questions, therefore, please refrain from copying from your class peers.

For Grading Purposes Only:

Q1	Q2	Q3	Q4	Q5	Q6
10	15	10	20	10	10

Q7	Q8
5	20

Total Score
100

Problem 1 (10 points)

Consider a RISC microprocessor, like the MIPS presented in the textbook, for which we want to implement the full addressable space. Assume we have a 30GB hard disk, a 512MB main memory, a 1MB L2 Cache and a 256KB internal Cache.

Explain how programmers are given the possibility of writing programs that assume an implementation of the full addressable space. (Hint: explain the function of the management of the memory hierarchy).

The hard disk is divided into pages of 4 kB. As a memory location is needed from the hard disk, the sequential pages of data is loaded from the hard disk into the main memory to the maximum amount. Which in turn will fill the L2 cache to its maximum amount, and then into the internal cache to the maximum page amount. The idea is that as the memory is used, the next data needed will be in sequence or at least in close proximity. That way, if there is a need for another memory location not in the internal memory, the microprocessor can load the page of data into the internal memory quickly by first checking for the page in the L2 cache, or the main memory, then the hard disk. The further the page is from the microprocessor the longer it takes to access. Since most data are kept in pages in close proximity, keeping data in large chunks of pages will speed up access time.

Microprocessor → internal cache → L2 cache → main memory → hard disk
Fast ----------→ Slow

Problem 2 (15 points)

Define direct mapped, fully associative and set associative caches. Explain how they relate to one another. Describe the advantages and disadvantages for each type of cache. Clearly explain your answer.

For fully associative caches, the address is compared against a directory of caches and if it is made, the cache is return to the CPU.

For direct mapped caches, the directory of caches hold multiple caches in the same index. To determine which cache should be returned, the requested cache index is accompanied by some tag bits to be compared. If a request hits the index and tag bits, that cache is returned.

For set associative caches, the directory of caches now have sub caches. Set associative caches work similarly to direct mapped caches. The requested cache index is given along with tag caches which determine which set or sub cache. If a request is found, that set or sub cache is returned.

Problem 3 (10 points)

Consider two processors with different cache configurations:
Cache 1: Direct-mapped with one-word blocks
Cache 2: Two-way set associative with four-word blocks
The following miss rate measurements have been made:

Cache 1: Instruction miss rate is 3%; data miss rate is 6%
Cache 2: Instruction miss rate is 2%; data miss rate is 3%

For these processors, one-half of the instructions contain a data reference. Assume that the cache miss penalty is 6 + Block size in words. Determine which processor spends more cycles on cache misses.

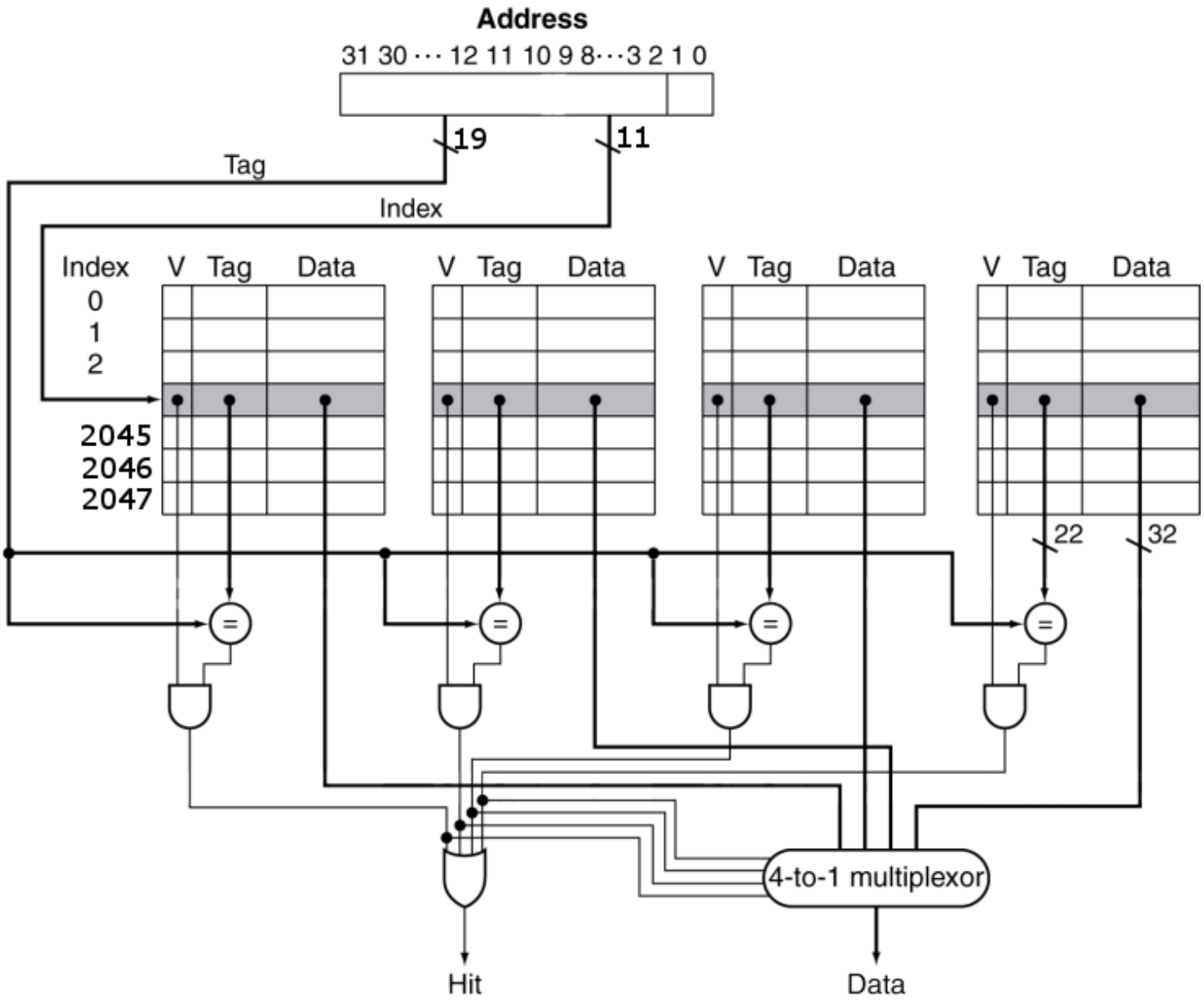
Cache 1:
Misses = 6 cycles + 1 word = 7 cycles
Stalls = 7 x 0.03 + 0.5 x 7 x 0.06 = 0.42

Cache 2:
Misses = 6 cycles + 4 words = 10 cycles
Stalls = 10 x 0.02 + 0.5 x 10 x 0.03 = 0.35

Cache 1 waste more times on misses.

Problem 4 (20 points)

Design a 32 KB, 4-way set-associative cache that has one-word blocks and each word is 32 bits. The computer's address size is 32 bits and uses byte addressing. Assume a valid bit is used in the cache. Make sure you calculate the number of bits you will use for the index and tag. Include all necessary hardware.



Problem 5 (10 points)

The following is a sequence of address references given as word addresses.
1, 5, 8, 4, 17, 19, 20, 6, 9, 8, 43, 5, 6, 21, 9, 17
Assuming a 2-way set associative 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. Assume LRU replacement policy.

Reference	Hit or Miss
1	Miss
5	Miss
8	Miss
4	Miss
17	Miss
19	Miss
20	Miss
6	Miss
9	Miss
8	Hit
43	Miss
5	Hit
6	Hit
21	Miss
9	Hit
17	Hit

Set #	Address
0	8
1	9
	17
2	
3	19
	43
4	4
	20
5	5
	21
6	6
7	

Problem 6 (10 points)

Assuming a direct-mapped cache with 4 four-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.

binary	Reference	Hit or Miss
00000001	1	Miss
00000101	5	Miss
00001000	8	Miss
00000100	4	Miss
00010001	17	Miss
00010011	19	Miss
00010100	20	Miss
00000110	6	Miss
00001001	9	Miss
00001000	8	Hit
00101011	43	Miss
00000101	5	Hit
00000110	6	Hit
00010101	21	Miss
00001001	9	Hit
00010001	17	Hit

Block #	Word0 -- 00	Word1 -- 01	Word2 -- 10	Word3 -- 11
0 – 00		~~1~~ → 17		19
1 – 01	4 → 20	~~5~~ →21	6	
2 – 10	8	9		43
3 – 11				

Problem 7 (5 points)

Multiple Choice Questions:
What is the reason for using a TLB?

To reduce address translation time.

The speed of the memory system affects the designer's decision on the size of the cache block. Which of the following cache designer guidelines are generally valid? Choose two.

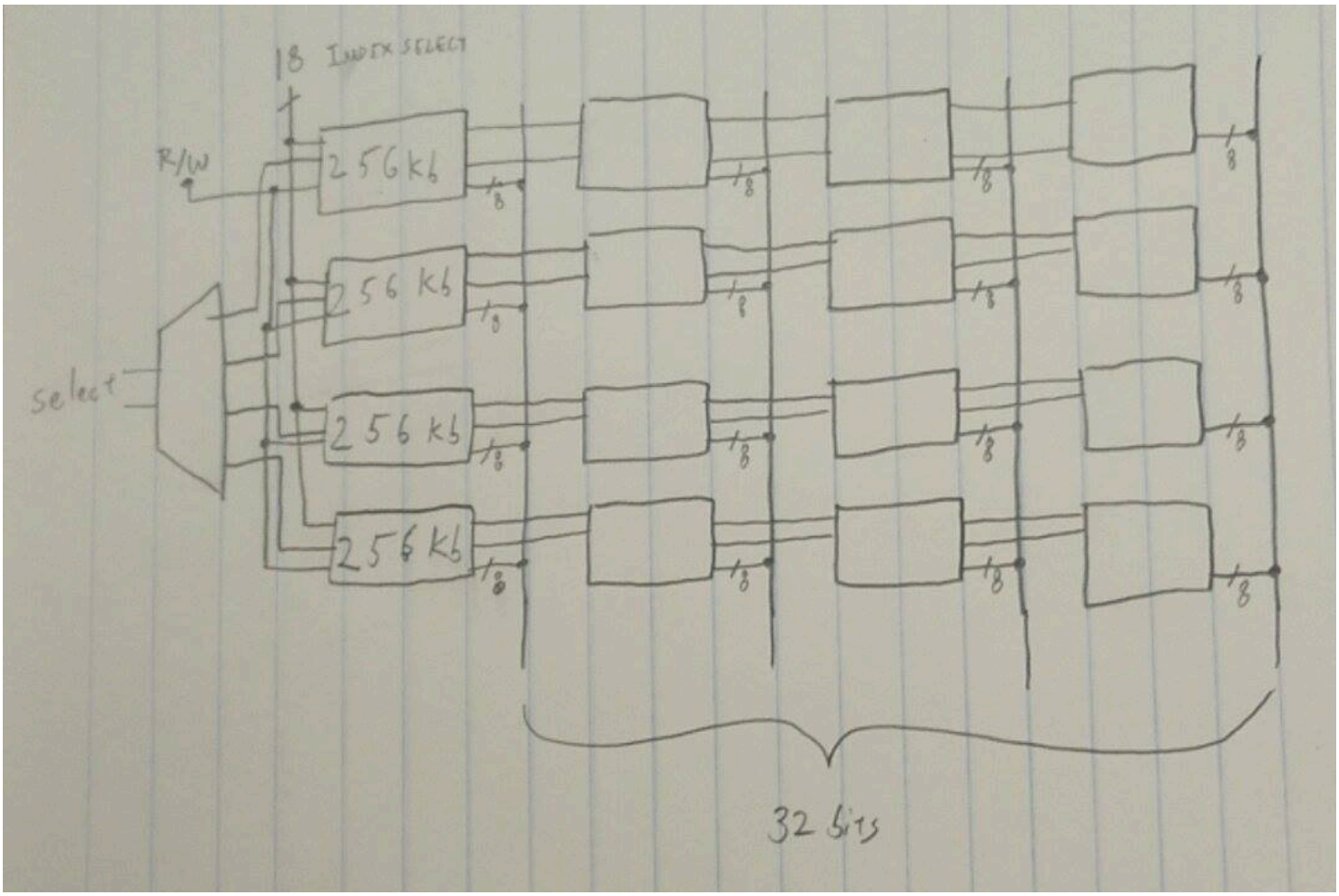
The higher the memory bandwidth, the larger the cache block.
The shorter the memory latency, the smaller the cache block.

Which of the following combinations of events in the TLB, virtual memory system, and cache is impossible?

TLB hit – page table miss – cache hit

Problem 8 (20 points)

Consider 256Kx8bits dynamic RAM chips where the access time is 1/2 (0.5) of the cycle time. Suggest a memory organization that will contain 4 megabytes, will have a 32-bit data bus and that will yield one word (32-bits) every access time if words are read from consecutive memory locations. Please clearly explain your answer.



4 MB = 16 x 256 Kb DRAM
1 word per access = 4 x 256 Kb DRAM
2 bits to select which word
18 bits to select which index in DRAM set
The 4 selected 256 Kb DRAM chip will out the necessary word (32 bits; 4 x 8 bits)