Habib University

EE-371/CS-330/CE-321 Computer Architecture – Spring 2023
Instructors: Dr. Tariq Kamal, Dr. Farhan Khan, Dr. Munzir Zafar, Dr. Sorath

Time = 40 minutes **Quiz 04** **Max Points: 20**

Instructions:

i. ***Smart watches, laptops, and similar electronics are strictly NOT allowed.***
ii. ***Answer sheets should contain all steps, working, explanations, and assumptions.***
iii. Attempt the quiz on clean papers with black/blue ink.
iv. Print your name and HU ID on all sheets.
v. Turn in your question paper along with your answer sheets.
vi. You are not allowed to ask/share your method or answer with your peers. The work submitted by you is solely your own work. Any violation of this will be the violation of HU Honor code and proper action will be taken as per university policy if found to be involved in such an activity.

CLO Assessment:

This assignment assesses students for the following course learning outcomes.

Course Learning Outcomes		CLO Assessed
CLO 1	***Explain*** the role of ISA in modern processors and instruction encodings and assembly language programming	
CLO 2	***Explain*** the architecture and working of a single cycle processor	
CLO 3	***Design*** the architecture to mitigate issues of a pipelined processor	
CLO 4	***Analyze the*** performance of cache operations	✓

Question 1 [6 points]:

Increasing associativity requires more comparators and more tag bits per cache block. Assuming a cache of 4096 blocks, a four-word block size, and a 64-bit address, find the total number of sets and the total number of tag bits for caches that are direct-mapped, two-way and four-way set associative, and fully associative.

No. of blocks = 4096 blocks = 2^{12} blocks -> Index = 12 bits
4 words per block = 4 * 4 = 16 Bytes = 2^4 Bytes -> Offset = 4 bits
Address: 64 bits

Type of Cache	Number of Sets	Tag bits	Index	Offset
Direct Mapped	4096	48	12	4
Two-Way Set Associative	2048	49	11	4
Four-Way Set Associative	1024	50	10	4
Fully Associative	1	60	-	4

Question 2 [6 points]:

Suppose we have a processor with a base CPI of 1.0, assuming all references hit in the primary cache, and a clock rate of 2 GHz. Assume a main memory access time of 100 ns, including all the miss handling.
Suppose the miss rate per instruction at the primary cache is 4%.
How much faster will the processor be if we add a secondary cache that has a 10-ns access time for either a hit or a miss and is large enough to reduce the miss rate to main memory to 0.4%?

Base CPI = 1.0
Clock rate = 2 GHz
Main memory access time = 100ns
Miss rate per instruction = 4%

With Primary Cache:
The miss penalty to main memory is:

$$\frac{100\ ns}{2\ GHz} = \frac{100\ ns}{2 * \ 10^9\ cycles\ per\ second} = \frac{100\ ns}{0.5\ ns\ per\ clock\ cycle} = 200\ clock\ cycles$$

The effective CPI with one level of caching is given by

Total CPI = Base CPI + Memory-stall cycles per instruction

For the processor with one level of caching,

Total CPI = 1 + 4% x 200 = 9

With two levels of caching, a miss in the primary (or first-level) cache can be satisfied either by the secondary cache or by main memory. The miss penalty for an access to the second-level cache is:

$$\frac{10\ ns}{2\ GHz} = \frac{10\ ns}{2 * \ 10^9\ cycles\ per\ second} = \frac{10\ ns}{0.5\ ns\ per\ clock\ cycle} = 20\ clock\ cycles$$

Thus, for a two-level cache, total CPI is the sum of the stall cycles from both levels of cache and the base CPI:

Total CPI = Base CPI + Primary-stall per instruction + Secondary stalls per instruction
Total CPI = 1 + 4% x 20 + 0.4% x 200 = 1+0.8+0.8 = 2.6

Thus, the processor with the secondary cache is faster by 9.0/2.6 = **3.4**

Question 3 [8 points]:

Assume a two-way set associative cache with four one-word blocks. Consider the following address sequence: 0, 2, 4, 8, 10, 2, 6, 4, 0, 11, 3, 10. Assuming an LRU replacement policy, which accesses are hits?

Word Address	Cache Index	Hit/Miss	Cache Index			
			Set 0		Set 1	
			Block 0	Block 1	Block 0	Block 1
0	0	M	MEM[0]			
2	0	M	MEM[0]	MEM[2]		
4	0	M	MEM[4]	MEM[2]		
8	0	M	MEM[4]	MEM[8]		
10	0	M	MEM[10]	MEM[8]		
2	0	M	MEM[10]	MEM[2]		
6	0	M	MEM[6]	MEM[2]		
4	0	M	MEM[6]	MEM[4]		
0	0	M	MEM[0]	MEM[4]		
11	1	M	MEM[0]	MEM[4]	MEM[11]	
3	1	M	MEM[0]	MEM[4]	MEM[11]	MEM[3]
10	0	M	MEM[0]	MEM[10]	MEM[11]	MEM[3]

There are no hits for the given sequence.