Name: _____.

Computer Science and Engineering 431 (CMPEN 431)

Exam 3

Fall 2022

12/6/2022

Name: _____.

Q1	15	
Q2	20	
Q3	15	
Q4	20	
Q5	10	
Q6	10	
Total	90	

Note:

You have **1 hour and 15 minutes** to complete the exam.

This exam is **closed notes and book**.

Please **write down solution steps** for partial credits to be considered.

Please also **write you name on top of each page**.

I strongly recommend **solving easy problems first**, as problems are not in the order of complexity.

Name: _____.

Q1. Select True or False (15 pt, 1pt each).

- (True / False) Virtual memory can give an illusion to each program having the entire private address space.
- (True / False) Virtual memory space cannot be larger than the physical memory space.
- (True / False) Page table is a dedicated special hardware for page translation.
- (True / False) Page placement is usually fully associative inside the page table, unlike cache.
- (True / False) If an access causes a page fault, it will always cause a TLB miss.
- (True / False) It is possible to have a TLB hit but a cache miss.
- (True / False) Invalidation-based cache coherence protocol can result in a higher bus traffic than an update-based cache coherence protocol.
- (True / False) In an MSI protocol, two cores can hold the same block in S state.
- (True / False) In an MSI protocol, one core can hold the same block in S state while another core is holding the block in M state.
- (True / False) In processor consistency, loads and stores to the same memory address can be reordered.
- (True / False) A critical section is a section of code that only one thread can enter at each time.
- (True / False) Lock can eliminate data races.
- (True / False) SIMD is good for regular structure of computations, such as image processing.
- (True / False) Completely connected network topology can scale well.
- (True / False) How you write your code can also impact the cache hit rate.

Name: _____.

Q2. Consider the page table below. Assume all entries are valid and only the first 5 rows of the page table are shown (20pt).

Page table

0x1234
0x5678
0x9abc
0xdef2
0x426a
…

- Q2-1. Assume a 32-bit virtual address 0x12345678. If the page is 4KB, what is the **virtual page number (VPN)** and the **page offset**, respectively? (5pt)

VPN : 0x12345
PO : 0x678

- Q2-2. Assume a 32-bit virtual address 0x12345678. If the page is 1MB, what is the **virtual page number (VPN)** and the **page offset**, respectively? (5pt)

VPN : 0x123
PO : 0x45678

- Q2-3. Assume a 32-bit virtual address 0x00001234 is translated into a 28-bit physical address. If the page is 4KB, what will the translated address be? Write in hexadecimal. Page table index starts from zero at the top. (5pt)

0x5678234

- Q2-4. If the page table indicates that the required page is not in the memory, briefly explain what happens. **Your explanation must include:** (1) the name of the exception, (2) which part of the system handles the exception, and (3) from where the missing page is read. (5pt)

① page fault
② OS
③ Secondary memory (storage)

Name: _____.

Q3. Consider the TLB shown below. Assume the TLB is fully associative. (15pt)

Tag (VPN)	PPN
0x001	0x78
0x012	0x56
0x123	0x34
0x678	0x12

- Q3-1. Assume a 32-bit virtual address 0x12345678. If the page is 1MB, what will the translated 28-bit physical address be? If the TLB does not contain the necessary entry to answer the question, write "TLB miss". (5pt)

0x3445678

- Q3-2. Explain why TLB can improve memory access latency. Your **answer must include why TLB reduces the number of memory accesses**. (5pt)

If TLB hits, page table lookup is not needed. Because the page table resides in the memory, it reduces the number of memory read.

- Q3-3. TLB can be placed before or after the cache (it can also be partially parallel, but let's not think about that here). Compared to placing TLB before the cache, what are the **advantage** and **disadvantage** of placing the TLB **after the cache**? (5pt)

Advantage: TLB lookup is not needed on cache hit.

Disadvantage: When two processes have two different VA for the same memory location or the same VA for different memory location, program is incorrect. (we need a way to handle such a case)

Name: _____.

Q4. Consider a SMP with three processors with local cache and shared memory. Assume they implement MSI writeback write-allocate cache coherence protocol with bus snooping. Below table shows the state of a single cache block of the three processors. (20pt total)

- Q4-1. Assume that initially, all blocks are in I state. Fill the table appropriately. (15pt)
  - Write M, S, or I under "P1—3 state" columns.
  - Write BusRd, BusRdX, or – (no bus signal generated) under the "Bus signal" column to indicate the signal that is generated by each processor action.
  - Write Memory, P1, P2, or P3 under the "Data supplier" column to indicate where the data is coming from. If the data is flushed from another processor, write P1, P2, or P3, instead of Memory (even if it is possible to first flush to the memory and read from the memory). If the data comes from the local cache of the processor, again write P1, P2, or P3 properly.

Action	P1 state	P2 state	P3 state	Bus signal	Data supplier
P1 reads X	S	I	I	BusRd	Memory
P2 reads X	S	S	I	BusRd	Memory
P2 writes X	I	M	I	BusRdX	P2
P3 reads X	I	S	S	BusRd	P2
P1 writes X	M	I	I	BusRdX	Memory
P2 writes X	I	M	I	BusRdX	P1

- Q4-2. **What is** a directory-based cache coherence protocol, and **what is the advantage** over snooping-based cache coherence protocol? (5pt)

Explanation: (skip)

Advantage: less bus traffic, more scalable

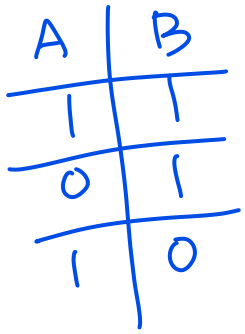
partial answer: faster (this is not always true)

Name: _____.

Q5. Consider a multithreaded code below. Initially, assume that A=0, B=0. (10pt total)

Thread 1	Thread 2
A = 1 print(B)	B=1 print(A)

- Q5-1. Write all the possible outputs assuming sequential consistency (SC). (5pt)



- Q5-2. Explain how having a write buffer can violate SC and allow an output that was not possible in Q5-1 to become possible. (5pt)

See the slide for answer

Name: _____.

Q6. Assume $s0 holds the address to the lock variable. Consider the below code for acquiring a lock. Assume that the lock variable holds 0 if the lock is available and holds 1 otherwise. The code is incomplete. Put **one** proper line of code somewhere to make it work.

Hint: **Load-linked** (ll $r1, offset($r2)) loads the value at memory address $r2 + offset into $r1. **Store-conditional** (sc $r1, offset($r2)) stores the value in $r1 into memory address $r2 + offset only if the memory location is not updated since the last load-linked. If the store succeeds, $r1=1, otherwise, $r1=0. (10pt total)

```
Lock: ll $t0, 0($s0)

      bne $t0, $zero, Lock

      addi $t0, $t0, 1

      sc $t0, 0($s0)

      beq $t0, $zero, Lock
```

(end of the exam)