

Name: _____

Roll No: _____

(COL 216) Computer Architecture

May 3, 2025

Major Exam

Duration: 120 minutes

(70 marks)

Beware: Be concise in your writing. You can use rough sheets for calculations. But you cannot submit any additional sheet for grading on Gradescope. So make sure you are certain when you write something (after rough work, or use a dark pencil). If you cheat, you will surely get an F in this course.

1. Excited after building the biometric attendance system, Samyak has now built an 8-bit navigation system. He takes this device to spend his summer vacation in Antarctica. However, as happens with prototypes, two circuit elements within his 8-bit navigation computer break. Miraculously, the computer still seems to work, except that when it reads or writes any value, the two least significant bits are always 0. For instance, this computer would erroneously compute: $00000001 + 00000011 = 00000000$. Notice that the computer also failed to carry from the two LSBs, since it read them as 0.
 - (a) What two's complement integers will this computer read correctly? [2 marks]
 - (b) What is the smallest magnitude normalized value that Samyak can represent with this broken system? [3 marks]
 - (c) Samyak is trying to navigate to the south pole with this broken computer. Fortunately, the navigation computer represents latitude with double precision, with 1 bit for the sign, 7 bits for the exponent, and 8 bits for the significand. This means that the double precision value will occupy two bytes (and each of the two bytes will have 0s in the two LSBs). The south pole has a latitude of -90. What is the closest latitude to the south pole that the computer can represent? Recall that with a 7 bit exponent, the exponent bias will be $2^{7-1} - 1 = 63$. [5 marks]

Hints:

- What positive exponents can this computer represent correctly?
- What should be the value of the sign bit for south pole?
- What is the biggest significand this computer can represent correctly?
- What is the number closest to -90 that Samyak can get on his device?
- What will be the two byte encoding of this number?

Name: _____

Roll No: _____

Name: _____

Roll No: _____

2. Consider a processor using a 4-block LRU-based L1 data cache. Starting with an empty cache, an application accesses three L1 cache blocks in the following order: $n \rightarrow n+2 \rightarrow n+4$. Consecutive numbers (e.g., $n, n+1, n+2, \dots$) represent the starting addresses of consecutive cache blocks in memory.

Vanshika wants to reverse engineer the number of sets and ways in the L1 data cache by issuing just two more accesses and observing only the cache hit rate across these two accesses. Assume that she can insert the malicious accesses only after the above three accesses of the program.

- (a) What are the next two cache blocks that she should access (e.g. $n+?$, $n+?$)? Is this choice of the next two cache blocks unique, or are there alternatives? [**3 marks**].
- (b) Explain the above choice of cache blocks with reasons [**5 marks**].
- (c) What should the next two accesses be if the replacement policy had been Most Recently Used (MRU)? [**2 marks**].

Name: _____

Roll No: _____

Name: _____

Roll No: _____

3. Nipun wants to parallelize the following code for his dual-core laptop. Each processor core has a 256KB on-chip cache with cache line size of 16 bytes, and uses the MESI protocol to ensure coherence between the two cores. Integer arrays A, B, and C do not overlap in memory and integers are four bytes.

```
1 for (int i=0; i<N; i++) {
2   A[i] = B[i] + C[i];
3 }
```

To equally distribute work between the two processors P0 and P1, the first parallelization Nipun tries is

```
1 //Loop for P0
2 for (int i=0; i<N; i+=2) {
3   A[i] = B[i] + C[i];
4 }
```

```
1 //Loop for P1
2 for (int i=1; i<N; i+=2) {
3   A[i] = B[i] + C[i];
4 }
```

- (a) Assuming that address a starts at the beginning of a cache line, fill in the state (M,E,S,I) of the cache line containing words a through $a+3$. The cache line state is the final state after the processor action finishes. Briefly explain the entries. [4 marks].

Loop iteration	Action	P0 cache line state	P1 cache line state
-	-	I	I
P0:1	P0 writes a		
P1:1	P1 writes $a + 1$		
P1:2	P1 writes $a + 3$		
P0:2	P0 writes $a + 2$		

Table 1: Cache line state

Name: _____

Roll No: _____

(b) Nikunj comments that the work may have been equally divided between the two processor cores, but the two cores will take more time to finish all loop iterations, compared to a single core. Nikunj rewrites the loops for the two processors, so that time to finish all loop iterations is now almost half, compared to a single core running all loop iterations. Write Nikunj's code for the two processor cores, and explain why the performance is better than Nipun's implementation. [**6 marks**].

Name: _____

Roll No: _____

4. Amber's processor has 128 Byte page size and an 8-entry direct-mapped TLB. The TLB is indexed by the lowest order bits of the virtual page number. Amber writes the program below, which operates on a matrix A. A has 4 rows and 256 columns. Each element of A is a 32-bit integer, and elements are laid out in row-major order (i.e., consecutive elements of the same row are in contiguous memory locations). The program sums the entries of a top-left submatrix B of matrix A. Assume that A starts at virtual address 0x0000, and the variable sum is stored in a register. Ignore instruction fetches.

```
1 int sum = 0;
2 for (int i = 0; i < H; i++){
3     for (int j = 0; j < W; j++){
4         sum += A[i][j];
5     }
6 }
```

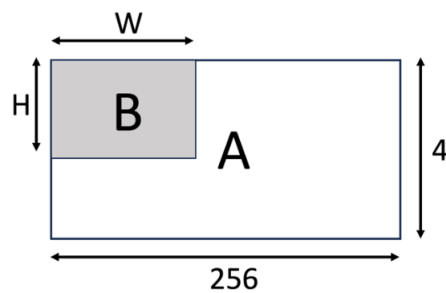


Figure 1: Matrix A and top-left sub-matrix B

- How many TLB misses will this program incur for $H = 4$, $W = 256$ (i.e., when $B = A$)? **[2 marks]**
- How many pages does a row of A occupy? **[1 mark]**
- What is the minimum TLB hit rate (i.e., TLB hits divided by number of accesses) that this program can have? What values of H and W cause this minimum? Specify all values of H and W that cause this. **[3 marks]**
- What is the maximum TLB hit rate this program can achieve? What values of H and W achieve this maximum? Specify all values of H and W that achieve this. **[3 marks]**
- What is the minimum page size that will cause Amber's program to incur at least one TLB hit for all values of H and W such that $H*W > 1$? Assume page sizes must be a power of 2. **[3 marks]**
- Amber rewrites the program as given below, swapping the order of the two for loops. Does this program have a better TLB hit than the previous one? Explain. **[3 marks]**

```
1 int sum = 0;
2 for (int j = 0; j < W; j++){
3     for (int i = 0; i < H; i++){
4         sum += A[i][j];
5     }
6 }
```

Name: _____

Roll No: _____

Name: _____

Roll No: _____

Name: _____

Roll No: _____

5. An out-of-order processor executes its instructions based on Tomasulo's original algorithm (i.e. no in-order retirement). The ISA specifies 8 registers R0 to R7. The execute stage of the pipeline contains one pipelined adder and one pipelined multiplier. This allows one ADD/MUL operation to start executing every clock cycle.

- Fetch and Decode take one cycle each.
- ADD takes 4 cycles to execute.
- MUL takes 5 cycles to execute.
- Both functional units have three-entry reservation stations and are initially empty.
- Reservation station entries are allocated at the end of Decode, choosing the first free entry from the top. Entries in the reservation station are freed in the cycle following their result broadcast.
- Data forwarding is implemented. Functional units broadcast results to the bus at the end of the last cycle of execution. This allows dependent instructions to start executing immediately after the value of the dependent source is known.
- When allocating a reservation station entry for ADD DR, SR1, SR2, the order of the operands SR1 and SR2 is maintained.
- The writeback bus supports only one result being stored at a time.

The Table below contains a six instruction program segment.

	Opcode	DR	SR1	SR2
I1				R4
I2				
I3				
I4	ADD	R5		
I5				
I6		R0		

Table 2: Instructions

Name: _____

Roll No: _____

Three snapshots of the register alias table (RAT) are provided: one before the first instruction is fetched, one at the end of cycle 7, and one after the 6th instruction stores its result.

	V	Tag	Value
R0	1	-	0
R1	1	-	1
R2	1	-	2
R3	1	-	3
R4	1	-	4
R5	1	-	5
R6	1	-	6
R7	1	-	7

Table 3: Before Cycle 1

	V	Tag	Value
R0	0	ρ	-
R1	1	-	8
R2	0		-
R3	1	-	3
R4	1	-	4
R5	0		-
R6	1	-	6
R7	1	-	7

Table 4: End of cycle 7

	V	Tag	Value
R0	1	-	225
R1	1	-	8
R2	1	-	12
R3	1	-	3
R4	1	-	4
R5	1	-	11
R6	1	-	6
R7	1	-	7

Table 5: Instructions complete

Here is a snapshot of the two reservation stations after cycle 7.

	V	Tag	Value	V	Tag	Value
α	0	β	-	1	-	3
β	1	-	8	1	-	4
γ	1	-	5	1	-	6

Table 6: Reservation station for Adder

	V	Tag	Value	V	Tag	Value
π	1	-	8	0	β	-
ρ	0	α	-	0	α	-
σ						

Table 7: Reservation station for multiplier

- Determine the 6 instructions that were executed and fill Table 2 [5 marks].
- Write in brief, how you derive each instruction [5 marks].
- Fill the missing tags in the boxed entries in Table 4 [2 marks].
- Complete the timing diagram for the execution of the six instructions. Write the stage each instruction occupies during each cycle. If an instruction is storing a result, write the name of the register written in that cycle. If an instruction is stalled in a stage, write an asterick (*) in that cycle. [3 marks].

Instruction	t1	t2	t3	t4	t5	t6	t7	t8	t9	t10	t11	t12	t13	t14	t15	t16	t17	t18	t19	t20
I1	F	D																		
I2		F	D																	
I3			F	D																
I4				F	D															
I5					F	D														
I6						F	D													

Table 8: Timing diagram

Name: _____

Roll No: _____

Name: _____

Roll No: _____

6. The following assembly code attempts to execute, but the first MUL instruction fails during the 5th clock cycle of its 6 cycles of execution. Each ADD instruction takes 4 cycles to execute. Both ADD and MUL instructions take one cycle each to Fetch, Decode and Store result. Data forwarding is supported.

```
1 ADD R1, R2, R3
2 MUL R4, R1, R2
3 ADD R2, R5, R6
4 MUL R4, R5, R6
```

- (a) What issue will a programmer face on a Tomasulo's machine to debug the MUL failure? **[2 marks]**
- (b) What needs to be added to Tomasulo's machine, to aid the programmer in debugging the MUL failure? **[2 marks]**
- (c) Other than debugging support for the programmer, lower latency, higher throughput and ensuring program correctness have been important goals handled in this Computer Architecture course. Among the concepts covered after minor, namely (a) caching, (b) cache coherence, (c) virtual memory, (d) out of order execution – name one concept needed to ensure program correctness and briefly explain why. Name one concept needed to reduce instruction latency and briefly explain why. Name one concept needed to increase throughput and briefly explain why. **[6 marks]**

Name: _____

Roll No: _____