HIGH PERFORMANCE COMPUTER ARCHITECTURE 20-11-2025 MATRICULATION NO.__________________

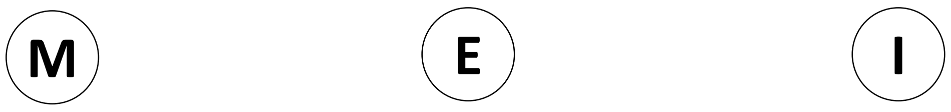
PLEASE REUTURN THIS SHEET ALONG WITH ALL THE SHEETS YOU WERE GIVEN

SURNAME______________________

FIRST NAME______________________

Consider a bus-based multicore that supports a cache-coherence protocol called MEI, what was used in the PowerPC-755. Compared to the well-known MESI protocol, the MEI protocol does not have the S state. There is no need for a BusUpgr transaction, only Flush*, BusRd and BusRdX may happen.

1a) [Points 4/30] Draw the diagram of the MEI protocol according to the above description.



1b) [Points 18/30] Assuming a cost of 1cc (1 clock-cycle) for read/write operations, 90cc for BusRd or BusRdx transactions, and 20cc for Flush*. Evaluate the total cost (in clock-cycles) for the following streams:

stream-1 MEI

Core Operation	C1	C2	C3	Bus Transaction	Data from	Cycles
PrRd1						
PrWr1						
PrRd1						
PrWr1						
PrRd2						
PrWr2						
PrRd2						
PrWr2						
PrRd3						
PrWr3						
PrRd3						
PrWr3						
TOTAL						

stream-2 MEI

Core Operation	C1	C2	C3	Bus Transaction	Data from	Cycles
PrRd1						
PrRd2						
PrRd3						
PrWr1						
PrWr2						
PrWr3						
PrRd1						
PrRd2						
PrRd3						
PrWr3						
PrWr1						
TOTAL						

stream-3 MEI

Core Operation	C1	C2	C3	Bus Transaction	Data from	Cycles
PrRd1						
PrRd2						
PrRd3						
PrRd3						
PrWr1						
PrWr1						
PrWr1						
PrWr1						
PrWr2						
PrWr3						
TOTAL						

2) Let's consider a cache-coherent multiprocessor system, in which each processor executes its code in program order (no reordering in source), but the hardware may reorder memory operations according to the consistency model. The variable x, y, z, are initialized to 0. We run the following program **once**:

P0:	**P1:**

`x = 1;`
`r1 = y;`
	`y = 1;`
	`r2 = x;`

2a) [4/30] Under **Sequential Consistency (SC)**: i) list all possible combinations of final values for `(r1, r2)` and ii) for each combination, say whether it is **allowed or forbidden** under SC.

2b) [4/30] Under a **TSO model** (store buffering but loads cannot be reordered with older loads, and stores become visible to others later): i) is the outcome `(r1=0, r2=0)` possible? Explain informally how store buffers make this outcome possible even if each core "respects program order" locally; ii) explain why coherence is **not** violated even if `(r1=0, r2=0)` occurs.