(REVISED 11/11/2025)

HIGH PERFORMANCE COMPUTER ARCHITECTURE 10-02-2016 **MATRICULATION NO._____**

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

SURNAME_____

FIRST NAME_____

1) (POINTS 24/40) Consider the following snippet of code running on 4-ways out-of-order superscalar processor. Initially, all registers contain zero.

```
lab1:   LW   R2,0(R1)
        ADDI R2,R2,1
        MUL  R4,R2,R2
        SW   R4,0(R1)
        ADDI R1,R1,4
        BNE  R2,R0,lab1
```

Working hypothesis:
* the fetch, decode and commit stages are 4 instructions wide
* the instruction window has 18 slots
* we have 8 physical registers in the free pool
* the reorder buffer has unlimited size
* the integer multiplier has 4 stages
* the load/store queues have 3 slots each and a common effective-address calculation unit
* there are 4 ALUs for arithmetic and logic operations and for branching
* an ALU performs its operation in the same cycle when the operation is issued
* reads require 1 clock cycle (after the addressing phase)
* the register file has 4 input- and 4 output-ports
* there are 9 logical registers (including R0 which is hardwired to 0)
* the store operation leaves the issue stage as it is inserted in the store queue

In order to calculate the total cycles needed to execute 3 iterations of the above loop on such machine, complete the following chart until the end of the third iteration of the code fragment above, including the renamed stream the precise evolution of the free pool of the physical registers (the register map), the Instruction Window, the Reorder Buffer (ROB) and the Load Queue (LQ) and Store Queue (SQ). Calculate the total cycles needed to execute three iterations of the above loop on such machine.

2) (POINTS 8/40) Consider a bus-based multicore that support a) MSI or b) MESI cache coherence protocol. The cost of a read/write operation is 1 cycle, the cost of a BusRd (or BusRdX) transaction is 90 cycles; all caches are write-back, write-allocate and initially empty. For the MSI protocol the BusUpgr is not used. The cost of a BusUpgr is 60 cycles. The cost of a Flush or Flush* is assumed 0. Evaluate the total cost of executing the following streams in the cases a and b:
Stream1: R1, W1, R1, W1, R2, W2, R2, W2, R3, W3, R3, W3
Stream2: R1, R2, R3, W1, W2, W3, R1, R2, R3, W3, W1
Stream3: R1, R2, R3, R3, W1, W1, W1, W1, W2, W3

3) (POINTS 8/40) The following code runs on a multicore that does not impose any ordering of the memory operations. However, the machine provides a FENCE instruction that, if inserted in a code, prevents issuing memory operations that come after the fence before the memory operations that are before the fence are globally performed. For simplicity, assume that lock and unlock instructions behave like acquire and release respectively by performing a single memory operation.
   a) Insert FENCE instructions as appropriate to ensure sequential consistency;
   b) Insert FENCE instructions as appropriate to ensure processor consistency;
   c) Insert FENCE instructions as appropriate to ensure weak ordering;

Lock1
Load A
Store B
Unlock1
Load C
Store D
Lock2
Load E
Store F
Unlock2