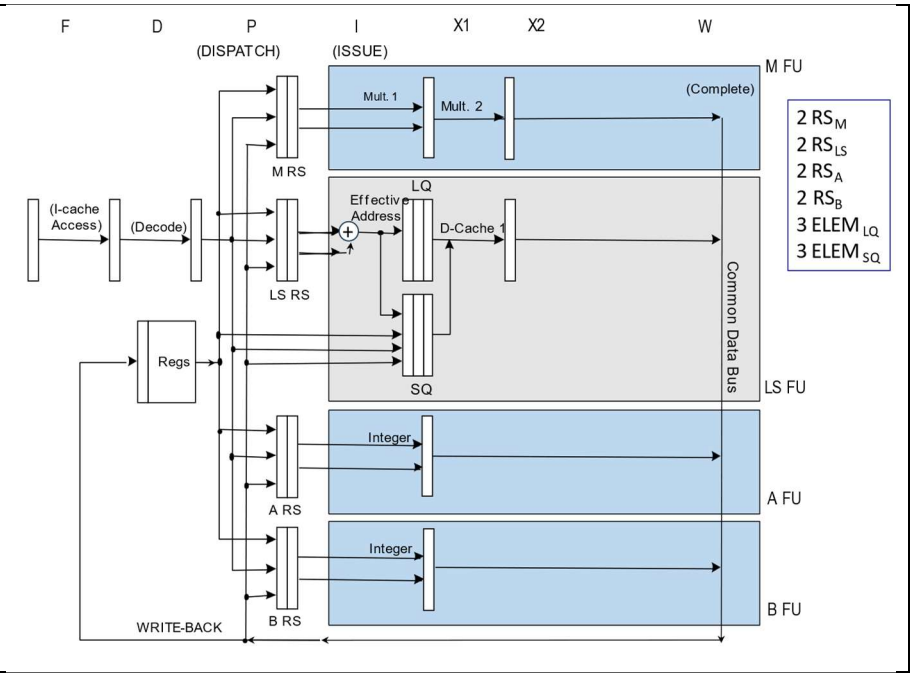
**HIGH PERFORMANCE COMPUTER ARCHITECTURE midterm exam 29-10-2024** **MATR.NO.**_____

**SURNAME**_____

**FIRST NAME**_____

1) (POINTS 27/30) Consider a **dual-dispatch (2 instructions per cycle)** processor using Tomasulo's algorithm to perform the dynamic scheduling of instructions on the pipeline shown in the following figure. This pipeline is executing the following program, which performs a search within a vector (initially, R1=0).

```
etic: LW   R2, 0(R1)     ; read Xi
      MULI R2, R2, 3     ; multiplies Xi by 3
      SW   R2, 0(R1)     ; write Xi
      ADDI R1, R1, 4     ; update  R1
      BNE  R2, R0, etic  ; continue to loop if
false
```




Working hypothesis:

- the loop executes speculatively in terms of direction (always taken) and regarding the branch condition; high-performance fetch breaks after fetching a branch
- the issue stage (I) calculates the address of the actual read/write and push it into load/store queues; only 1 instruction is issued per cycle
- **reads require 2 clock cycles**; writes take 1 cycle (this means that stage M is lasting just 1 cycle)
- when accessing memory (M), **writes** have precedence over reads and must be executed in-order
- **there is a single CDB**
- dispatch stage (P) and complete stage (W) require 1 clock cycle
- **ASSUME** that the reservation stations could be freed right before the start of issue phase (therefore extending the duration of P stage)
- only 1 instruction is committed (C stage) per cycle
- there are separated integer units: one for the calculation of the actual address, one for arithmetic and logical operations, one of the integer multiplication and one for the evaluation of the branch condition, as illustrated in this table:

| Type of Functional Unit | No. of Functional Units | Cycles for stage I+X | No. of reservation stations |
|---|---|---|---|
| LS: Integer (effective addr.) | 1 | 1 | 2 |
| A: Integer (op. A-L) | 1 | 1 | 2 |
| B: Integer (branch calc.) | 1 | 1 | 2 |
| M: Integer Multiplication | 1 | **2** | 2 |

- the functional units TAKE advantage of pipelining techniques internally
- the load queue has 3 slots; the store queue has 3 slots (writes wait for the operand in the store queue, i.e., in the execution stage)

Complete the following chart until the end of the FOURTH iteration of the above code fragment in the case of dynamic scheduling with speculation. Also add the instruction that occupies a certain reservation station (one of the 8) as indicated below:

| Instr. No.. | Instruction name | ALU RS1 | ALU RS2 | LS RS1 | LS RS2 | BU RS1 | BU RS2 | MU RS1 | MU RS2 | P: disPatch (clock) | I+X:Issue+Exec (start-stop) | M: MEM.ACCESS (start-stop) | W: CDB-write (clock) | C: Commit (clock) | Comments |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| I01 | **LW R2,0(R1)** | | | I01 1-1 | | | | | | 1 | 2-2 | 3-4 | 5 | 6 | |
| ... | ... | | | | | | | | | | | | | | |
| ... | ... | | | | | | | | | | | | | | |

2) (POINTS 3/30) On a Linux system, write the SINGLE command line to perform at the BASH shell prompt the following operation (please note that no intermediate files should be used):

- Print on the standard output the name and size of the files (including "." and "..") of the directory "/home/mario".