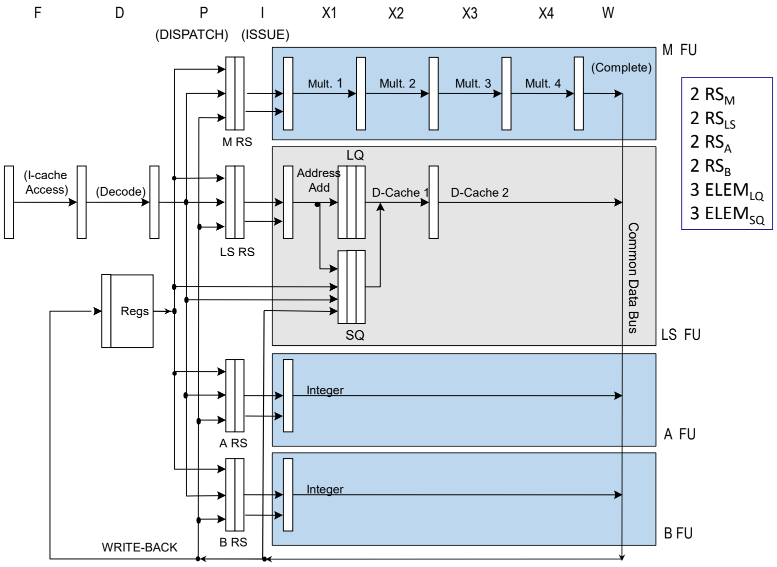
HIGH PERFORMANCE COMPUTER ARCHITECTURE midterm exam 28-10-2022 **MATR.NO.**___________________

SURNAME___________________

FIRST NAME___________________

1) (POINTS 27/30) Consider a **dual-dispatch (2 instructions per cycle)** 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 issue stage (I) calculates the effective address of the loads and stores.
- dispatch stage (D) and commit stage (C) require one clock cycle.
- only one instruction is issued per cycle.
- loads require five clock cycles, stores require one clock cycle.
- instructions are executed speculatively, and branches are predicted “always taken.”
- dispatch will break the instruction flow when a branch instruction is encountered.
- **there two CDBs**.
- integer units are separated, as illustrated in this table:

Type of Functional Unit	No. of Functional Units	Cycles for stage I	No. of reservation stations
LS: Integer (effective addr.)	1	1	2
A: Integer (op. A-L)	1	1	2
B: Integer (branch cond. calc.)	1	1	2
M: Integer Multiplication	1	4	2

- the functional units TAKE advantage of pipelining techniques internally.
- the load and store queues have three slots each.
- stores have priority over the loads when there is a conflict at the same cycle to access cache.
- ASSUME that the reservation stations could be freed right BEFORE the issue phase.

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:

Instr. No..	Instruction name	ALU RS1	ALU RS2	EAU RS1	EAU RS2	BU RS1	BU RS2	MU RS1	MU RS2	P: Dispatch (start)	I+X: Issue (start-stop)	MEM. ACCESS (start-stop)	W: CDB-write (clock)	C: Commit (clock)	Comments
I01	LW R2,0(R1)			I01 1-1						1	2-2	3-7	8	9	
...	...														
...	...														

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):

- Change permission for the owner to read and execute all files which have a name starting with “finance” and ending with “sh” and remove permission of reading to the group and other users.

HIGH PERFORMANCE COMPUTER ARCHITECTURE midterm exam 28-10-2022 – SOLUTION

EXERCIZE 1

Instr. No..	Instruction name	ALU RS1 (start-stop)	ALU RS2 (start-stop)	EAU RS1 (start-stop)	EAU RS2 (start-stop)	BU RS1 (start-stop)	BU RS2 (start-stop)	MU RS (start-stop)1	MU RS2 (start-stop)	P: Dispatch (start)	I+X: Issue (start-stop)	MEM. ACC. (start-stop)	W: CDB-write (clock)	C: Commit (clock)	Comments
I01	LW R2,0(R1)			I01 1-1						1	2-2	3-7	8	9	
I02	MULI R2,R2,3							I02 1-8		1	9-12	--	13	14	I waits R2 from 1/LW
I03	SW R2,0(R1)				I03 2-2					2	3-3	18	--	19	I waits issue logic; M waits R2 M waits mem
I04	ADDI R1,R1,4	I04 2-3								2	4-4	--	5	20	I waits issue logic;
I05	BNE R2,R0,etic					I05 3-13				3	14-14	--	--	21	I waits R2 from 1/MULI
I06	LW R2,0(R1)			I06 4-5						4	6-6	8-12	13	22	I waits R1; M waits mem
I07	MULI R2,R2,3								I07 4-15	4	15-18	--	19	23	I waits R2 from 2/LW; I waits issue logic;
I08	SW R2,0(R1)			I08 5-6						5	7-7	24	--	25	I waits R1; I waits issue logic; M waits R2; M waits mem
I09	ADDI R1,R1,4	I09 5-7								5	8-8	--	9	26	I waits R1 from 1/ADDI; I waits issue logic;
I10	BNE R2,R0,etic						I10 6-19			6	20-20	--	--	27	I waits R2 from 2/MULI;
I11	LW R2,0(R1)			I11 7-9						7	10-10	13-17	18	28	P waits EA-RSs I waits issue logic; I waits R1; M waits mem
I12	MULI R2,R2,3							I12 9-18		9	19-22	--	23	29	P waits M-RSs; I waits R2 from 3/LW
I13	SW R2,0(R1)				I13 9-10					9	11-11	25	--	30	I waits R1; I waits issue logic; M waits R2; M waits mem
I14	ADDI R1,R1,4		I14 10-11							10	12-12	--	14	31	I waits issue logic; W waits CDB
I15	BNE R2,R0,etic					I15 14-23				14	24-24	--	--	32	P waits B-RSs; I waits R2 from 3/MULI
I16	LW R2,0(R1)			I16 15-15						15	16-16	19-23	24	33	I waits R1; I waits issue logic; M waits mem;
I17	MULI R2,R2,3								I17 15-24	15	25-28	--	29	34	I waits R2 from 4/LW; I waits issue logic;
I18	SW R2,0(R1)				I18 16-					16	26-26	30	--	35	I waits R1; I waits issue logic; M waits R2; M waits mem*;
I19	ADDI R1,R1,4	I19 16-16								16	17-17	--	18	36	
I20	BNE R2,R0,etic						I20 20-29			20	30-30		--	37	P waits B-RSs; I waits R2 from 4/MULI

EXERCIZE 2

The requested command line is:

```
chmod u+rx,og-r finance*sh
```