7 Tomasulo's Algorithm [36 points]

In this problem, we consider an in-order fetch, out-of-order dispatch, and out-of-order retirement execution engine that employs Tomasulo's algorithm. This engine behaves as follows:

- The engine has four main pipeline stages: Fetch (F), Decode (D), Execute (E), and Write-back (W).
- The engine can fetch **FW** instructions per cycle, decode **DW** instructions per cycle, and write back the result of **RW** instructions per cycle.
- The engine has two execution units: 1) an *integer ALU* for executing integer instructions (i.e., addition and multiplication) and 2) a *memory unit* for executing load/store instructions.
- Each execution unit has an **R**-entry reservation station.
- An instruction always allocates the first available entry of the reservation station (in top-to-bottom order) of the corresponding execution unit.

The reservation stations are all initally empty. The processor fetches and executes *six* instructions. Table 2 shows the six instructions and their execution diagram.

Using the information provided above and in Table 2 (see the next page), fill in the blanks below with the configuration of the out-of-order microarchitecture. Write "Unknown" if the corresponding configuration cannot be determined using the information provided in the question.

The latency of the ALU and memory unit instructions:	ALU - 2 cycles, MU - 10 cycles
In which pipeline stage is an instruction dispatched?	Decode (D) stage
Number of entries of each reservation station (R):	Two entries each
Fetch width (FW):	2
Decode width (DW):	2
Retire width (RW):	Unknown
Is the integer ALU pipelined?	Unknown
Is the memory unit pipelined?	Yes
If applicable, between which stages is data forwarding implemented?	No data forwarding

Instruction/Cycle:	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33
1: ADD R1 ← R0, R1	F	D	E1	E2	W																												
2: LD R2 ← [R1]	F	D	-	-	-	E1	E2	E3	E4	E5	E6	E7	E8	E9	E10	W																	
3: ADDI R1 ← R1, #4		F	D	-	-	E1	E2	W																									
4: LD R3 ← [R1]		F	D	-	-	-	-	-	E1	E2	E3	E4	E5	E6	E7	E8	E9	E10	W														
5: MUL R4 ← R2, R3			F	-	-	D	-	-	-	-	-	-	-	-	-	-	-	-	-	E1	E2	W											
6: ST [R0] ← R4			F	-	-	-	-	-	-	-	-	-	-	-	-	-	D	-	-	-	-	-	E1	E2	E3	E4	E5	E6	E7	E8	E9	E10	W

Table 2: Execution diagram of the six instructions.