Assignment – 1. 30 marks. Due: March 6, 2021, 11:59pm. Individual assignment. Assignment will need to be demo'd. You can alternately create a short video demo and upload it. A report should briefly contain the results (clock cycle in which instructions complete each stage). Implementation can be done in any language of your choice

Consider an out of order processor implementing a Tomasulo algorithm with an 8-entry re-order buffer. The number of reservation stations is shown below. Contents of the architectural register file (ARF) are shown. We have one execution unit for Add, one for MUL/Div (They are pipelined)

1. Issue and Write take one cycle each. So, if data is written to CDB in clock cycle 'a', the data is read from the reservation station or register in the next clock cycle 'a+1'
2. Similarly, if execution completes in cycle 'a', it is written to CDB in cycle 'a+1'
3. Latency: Add : 1 cycle. Mult: 10 cycles. Divide: 40 cycles, Load: 5 cycles

Implement a single-issue RISC-V processor which reads in the instructions shown in the table and runs the instructions out of order. Follow the RISC-V instruction encodings as in the RISC-V Specification. The MUL and DIV are part of the 'M' extensions of RISC-V. You can choose the number of stages in the pipeline as per the Tomasulo algorithm (4 stages) or you can implement a 7-stage pipeline.
The Tomasulo structure with RoB is shown below (CDB is an important component of the design). RAT also needs to be a part of the design.

Three Reservation station (Add/Sub)

Instruction	Busy	Destination tag	Source tag1	Source tag2	Value of source 1	Value of source 2

Two Reservation stations (Multiplier/Divide)

Instruction	Busy	Destination tag	Source tag1	Source tag2	Value of source 1	Value of source 2

Three Load/Store buffers

Instruction	Busy	Destination tag	Address offset	Source register

Architectural Register file (ARF)

R1	R2	R3	R4	R5	R6	R7	R8	R9	R10
12	16	45	5	3	4	1	2	2	3

Instructions

	Issue to RS	Execute start/Address calculation for LD	Execute/ Memory access complete	Write to CDB	Commit
LW R3, 0(R2)					
DIV R2, R3, R4					
MUL R1, R5, R6					
ADD R3, R7, R8					
MUL R1, R1, R3					
SUB R4, R1, R5					
ADD R1, R4, R2					

Register allocation table (RAT)

R1	R2	R3	R4	R5	R6	R7	R8	R9	R10

ROB

	Instruction type	Destination	Value
RoB1			
RoB2			
RoB3			
RoB4			
RoB5			
RoB6			
RoB7			
RoB8