Course No.: CSE 306

Course Name: COMPUTER ARCHITECTURE SESSIONAL

Topic: Assignment on 8-bit MIPS Pipelined Execution

Group No: 05

Department: CSE

Section: B2

Level/Term: 3/1

Group Member's ID: 1705095

1705096

1705097

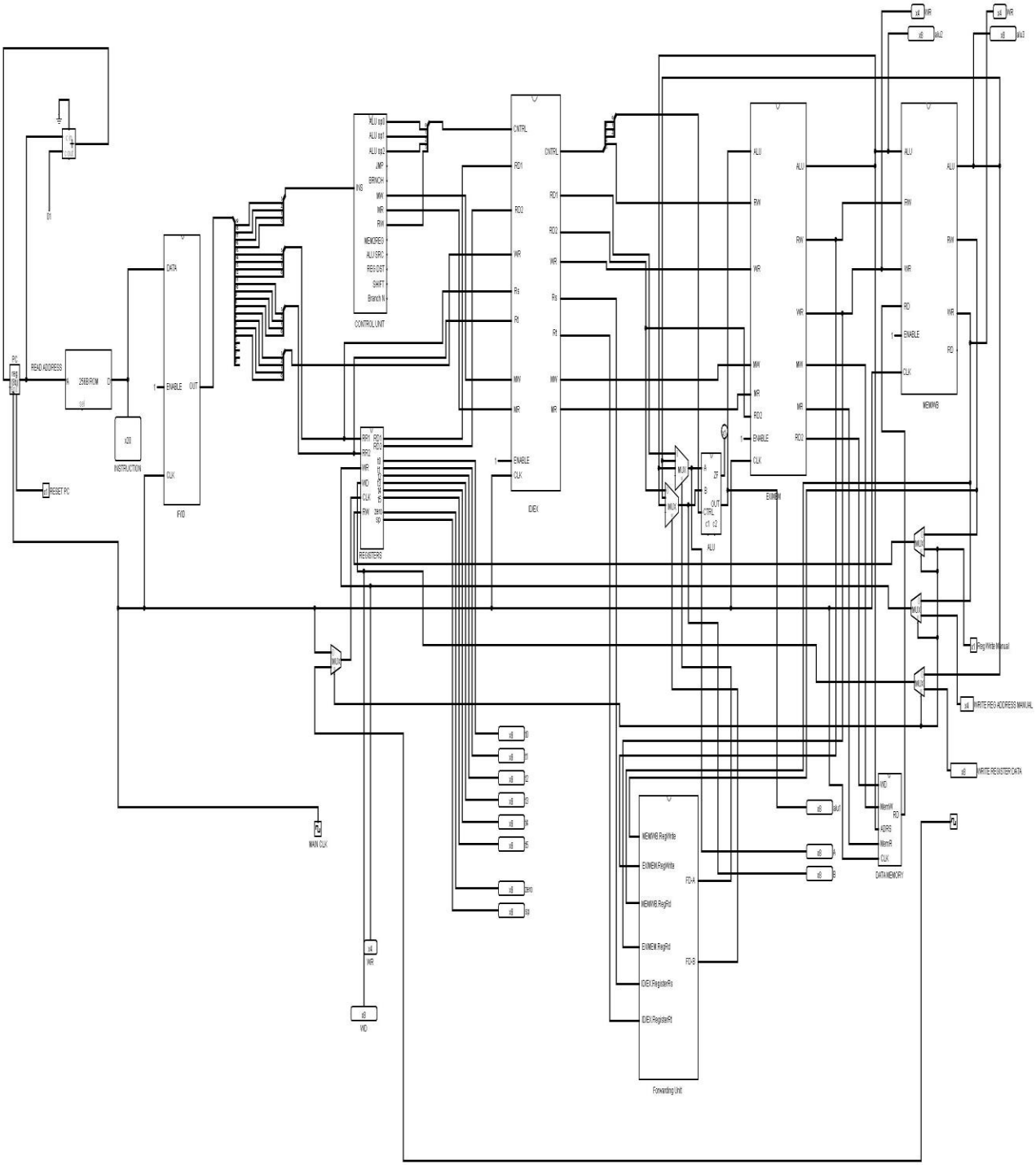
1705102

1705104

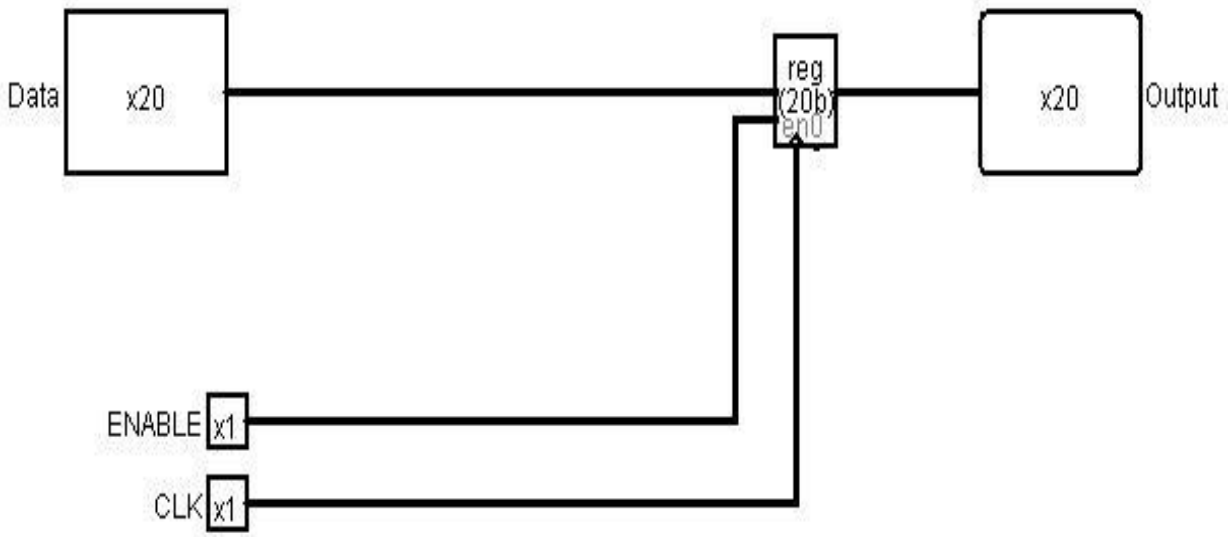
Introduction:

Pipelining is a technique that exploits parallelism among the instructions in a sequential instruction stream. It increases the number of simultaneously executing instructions and the rate at which instructions are started and completed. In this assignment, we designed an 8-bit processor that supports pipelined datapath for a subset of MIPS instruction set. In this design, each instruction is divided into five stages: Instruction Fetch(IF), Instruction Decode (ID), Execution and address calculation (EX), Data Memory access (MEM), and Write Back(WB). Therefore, each instruction takes up to five clock cycles to be executed. The length of the clock cycle equals the maximum time to execute any single stage.

Complete Block Diagram Of Pipelined Datapath :

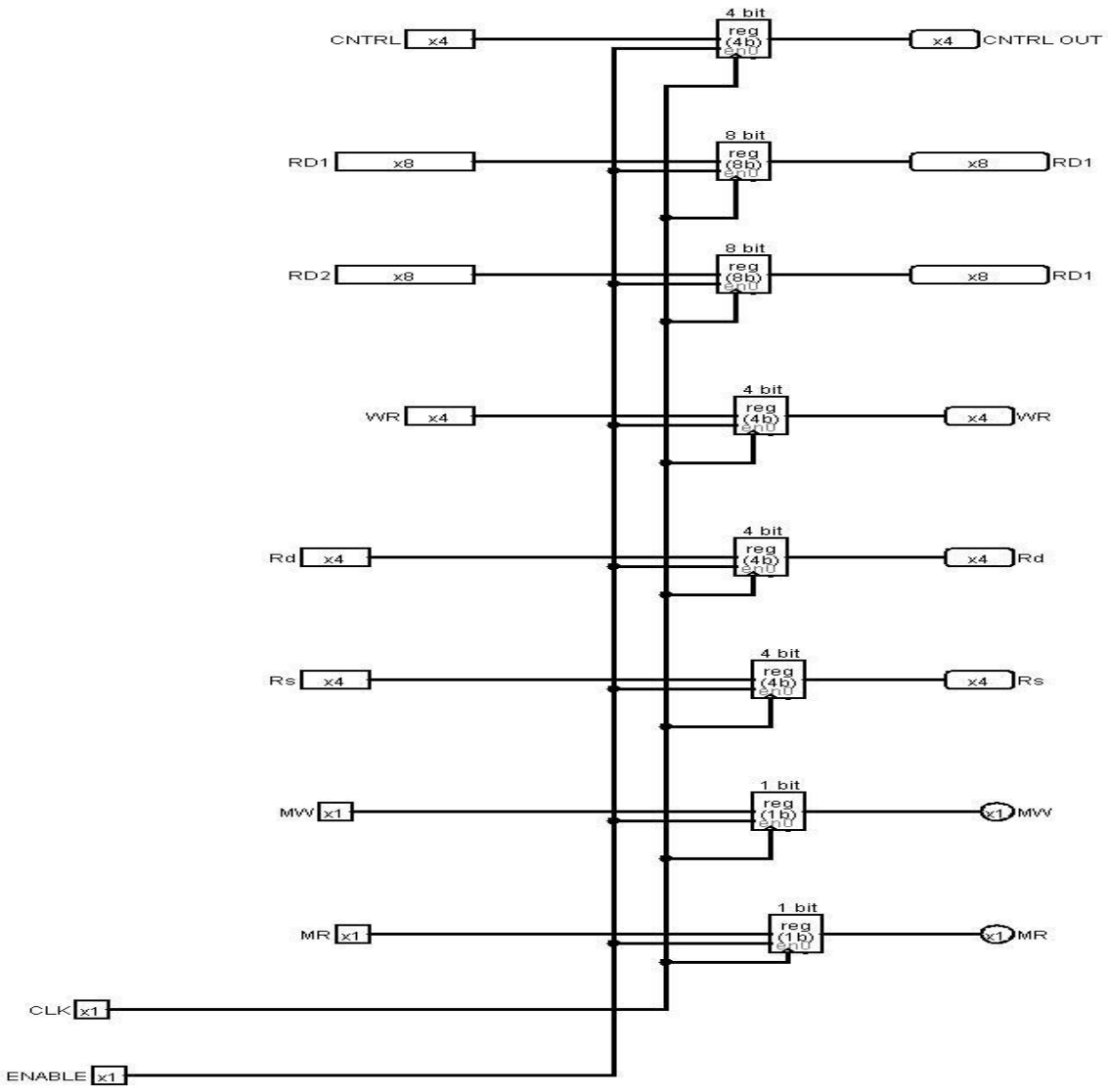


Block Diagram Of IF/ID Pipelined Register :



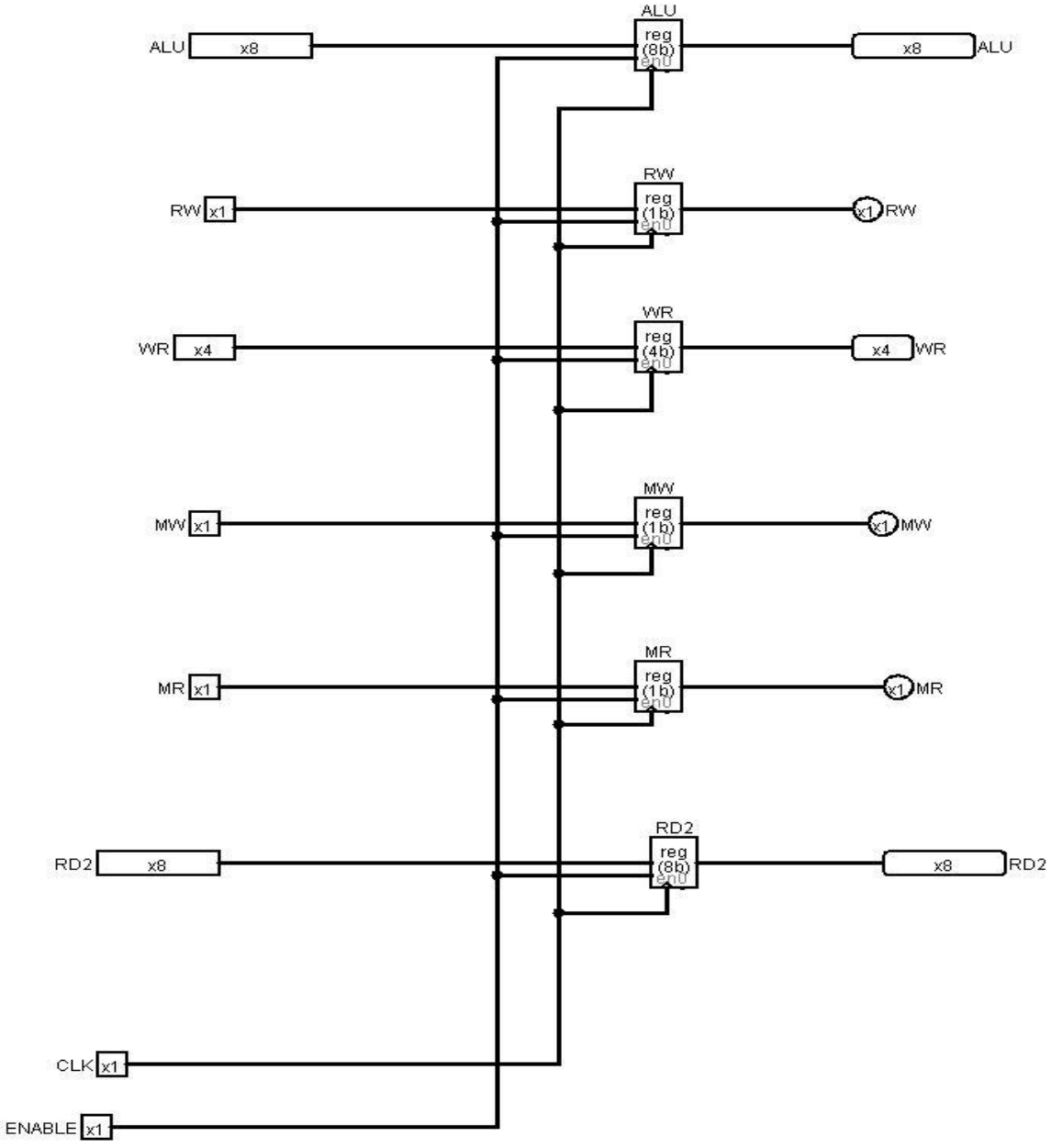
Size of IF/ID Pipeline Register: one 20bit register(Total 20 bit)

Block Diagram Of ID/EX Pipelined Register :



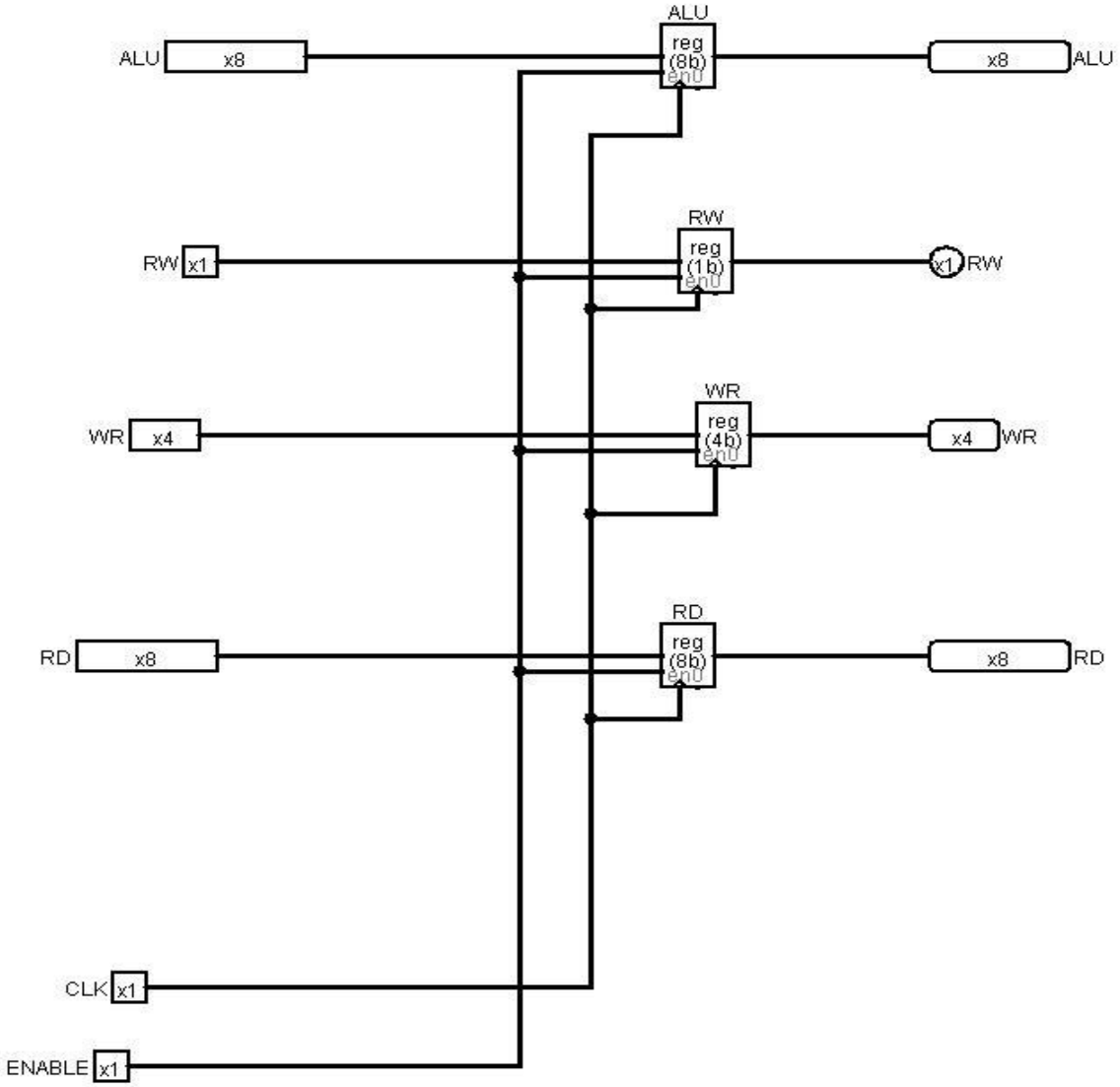
Size of ID/EX Pipeline Register : four 4bit, two 8bit, two 1bit registers(Total 34 bit).

Block Diagram Of EX/MEM Pipelined Register :



Size of EX/MEM Pipeline Register : two 8bit, one 4bit, three 1bit registers(Total 23 bit).

Block Diagram Of MEM/WB Pipelined Register :



Size of EX/MEM Pipeline Register : two 8bit, one 4bit, one 1bit registers(Total 21 bit).

Mechanism Of Forwarding Unit :

Forwarding unit is a hardware solution to deal with data hazards. The idea is to pass proper values early from the pipeline registers to ALU rather than waiting for the WB stage to write the register file. In this design, the only instructions we need to forward are four R-format instructions: add, sub, and , or. Forwarding unit uses the following conditions to detect two types of hazards:

1.EX hazard:

if (EX/MEM.RegWrite
and (EX/MEM.RegisterRd ≠ 0)
and (EX/MEM.RegisterRd = ID/EX.RegisterRs)) ForwardA = 10

if (EX/MEM.RegWrite
and (EX/MEM.RegisterRd ≠ 0)
and (EX/MEM.RegisterRd = ID/EX.RegisterRt)) ForwardB = 10

2.MEM hazard:

if (MEM/WB.RegWrite
and (MEM/WB.RegisterRd ≠ 0)
and not (EX/MEM.RegWrite and (EX/MEM.RegisterRd ≠ 0)
and (EX/MEM.RegisterRd ≠ ID/EX.RegisterRs))
and (MEM/WB.RegisterRd = ID/EX.RegisterRs)) ForwardA = 01

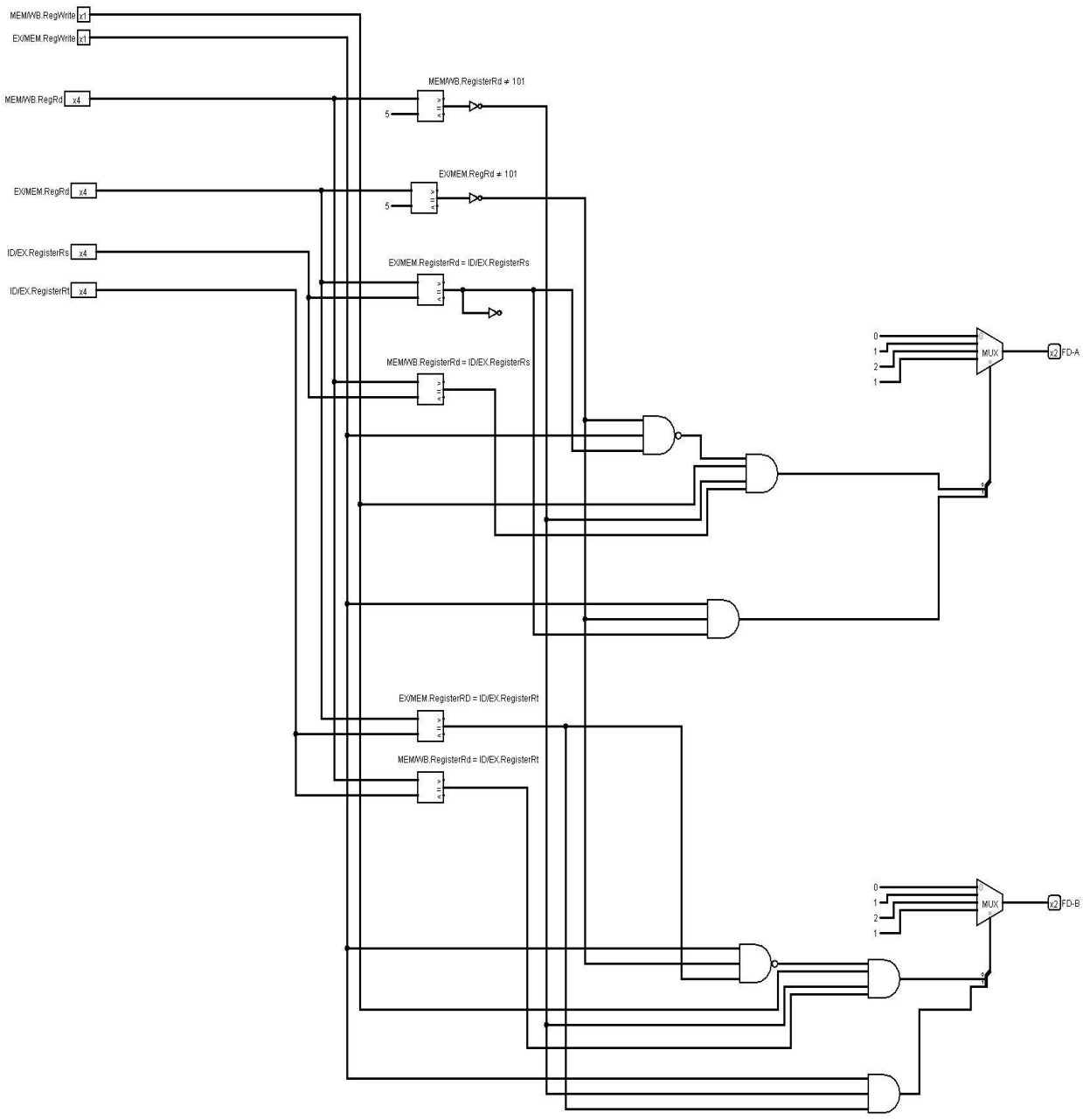
if (MEM/WB.RegWrite
and (MEM/WB.RegisterRd ≠ 0)
and not (EX/MEM.RegWrite and (EX/MEM.RegisterRd ≠ 0)
and (EX/MEM.RegisterRd ≠ ID/EX.RegisterRt))
and (MEM/WB.RegisterRd = ID/EX.RegisterRt)) ForwardB = 01

Forwarding control will be in the EX stage because the ALU forwarding multiplexors are found in that stage. The control values for the forwarding multiplexors are given in the table below :

Mux control	**Source**	**Explanation**
ForwardA=00	ID/EX	The first ALU operand comes from the register file.
ForwardA=10	EX/MEM	The first ALU operand is forwarded from the prior ALU result.
ForwardA=01	MEM/WB	The first ALU operand is forwarded from data memory or an earlier ALU result.
ForwardB=00	ID/EX	The second ALU operand comes from the register file.
ForwardB=10	EX/MEM	The second ALU operand is forwarded from the prior ALU result.
ForwardB=01	MEM/WB	The second ALU operand is forwarded from data memory or an earlier ALU result.

Table: The control values for the forwarding multiplexors.

Block Diagram Of Forwarding Unit :



Discussion:

We used components from the previous offline. We added four pipeline registers to save states of the instruction to use them on appropriate time. Because if not saved by the pipeline register, they will be overwritten by future instructions. All the registers are triggered through rising edge of clock except for the ones the registers file. Those are triggered through the falling edge to save the pipelined datapath from a potential hazard. We did not use forwarding or any other method to resolve this hazard. But we improved the hardware in the simulation by making the register triggered in the falling edge.