REGISTER :-

MAIN PROGRAM :-

module register\_file (

input clk,

input [4:0] rs1, rs2, rd,

input [18:0] write\_data,

input reg\_write,

output [18:0] reg1\_data,

output [18:0] reg2\_data);

reg [18:0] registers[31:0];

assign reg1\_data = registers[rs1];

assign reg2\_data = registers[rs2];

always @(posedge clk) begin

if (reg\_write) begin

registers[rd] <= write\_data;

end

end

endmodule

TESTBENCH :-

`timescale 1ns / 1ps

module register\_file\_tb();

reg clk;

reg [4:0] rs1, rs2, rd;

reg [18:0] write\_data;

reg reg\_write;

wire [18:0] reg1\_data, reg2\_data;

register\_file uut (

.clk(clk),

.rs1(rs1),

.rs2(rs2),

.rd(rd),

.write\_data(write\_data),

.reg\_write(reg\_write),

.reg1\_data(reg1\_data),

.reg2\_data(reg2\_data));

initial clk = 0;

always #5 clk = ~clk;

initial begin

reg\_write = 1; rd = 5'd1; write\_data = 19'd10;

#10;

reg\_write = 0; rs1 = 5'd1;

#10;

$display("Read register 1: %d", reg1\_data);

reg\_write = 1; rd = 5'd2; write\_data = 19'd20;

#10;

reg\_write = 0; rs2 = 5'd2;

#10;

$display("Read register 2: %d", reg2\_data);

#10 $finish;

end

endmodule

OUTPUT :-

