CONTROL UNIT :-

MAIN PROGRAM :-

module control\_unit (

input [18:0] instruction,

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

output reg [3:0] alu\_op,

output reg reg\_write,

output reg mem\_read, mem\_write,

output reg [18:0] imm);

always @(\*) begin

rs1 = instruction[13:9];

rs2 = instruction[8:4];

rd = instruction[18:14];

imm = instruction[8:0];

alu\_op = 4'b0000;

reg\_write = 0;

mem\_read = 0;

mem\_write = 0;

case (instruction[18:14])

5'b00000: alu\_op = 4'b0000;

5'b00001: alu\_op = 4'b0001;

5'b00010: alu\_op = 4'b0010;

5'b00011: alu\_op = 4'b0011;

5'b00100: alu\_op = 4'b1000;

5'b00101: alu\_op = 4'b1001;

5'b01000: alu\_op = 4'b0100;

5'b01001: alu\_op = 4'b0101;

5'b01010: alu\_op = 4'b0110;

5'b01011: alu\_op = 4'b0111;

5'b10000: begin

mem\_read = 1;

reg\_write = 1;

end

5'b10001: begin

mem\_write = 1;

end

endcase

end

endmodule

TESTBENCH :-

`timescale 1ns / 1ps

module control\_unit\_tb();

reg [18:0] instruction;

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

wire [3:0] alu\_op;

wire reg\_write, mem\_read, mem\_write;

control\_unit uut (

.instruction(instruction),

.rs1(rs1),

.rs2(rs2),

.rd(rd),

.alu\_op(alu\_op),

.reg\_write(reg\_write),

.mem\_read(mem\_read),

.mem\_write(mem\_write));

initial begin

instruction = 19'b00000\_00001\_00010\_00011;

#10;

$display("ADD: rs1=%d, rs2=%d, rd=%d, alu\_op=%b, reg\_write=%b", rs1, rs2, rd, alu\_op, reg\_write);

instruction = 19'b00001\_00001\_00010\_00011;

#10;

$display("SUB: rs1=%d, rs2=%d, rd=%d, alu\_op=%b, reg\_write=%b", rs1, rs2, rd, alu\_op, reg\_write);

#10 $finish;

end

endmodule

OUTPUT :-

