CPU :-

MAIN PROGRAM :-

module cpu\_19bit (

input clk, reset,

input [18:0] instruction,

output [18:0] result);

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

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

wire [3:0] alu\_op;

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

wire zero;

register\_file reg\_file (

.clk(clk),

.rs1(rs1),

.rs2(rs2),

.rd(rd),

.write\_data(alu\_result),

.reg\_write(reg\_write),

.reg1\_data(reg1\_data),

.reg2\_data(reg2\_data));

alu\_19bit alu (

.a(reg1\_data),

.b(reg2\_data),

.alu\_op(alu\_op),

.result(alu\_result),

.zero(zero));

control\_unit ctrl (

.instruction(instruction),

.rs1(rs1),

.rs2(rs2),

.rd(rd),

.alu\_op(alu\_op),

.reg\_write(reg\_write),

.mem\_read(mem\_read),

.mem\_write(mem\_write));

assign result = alu\_result;

endmodule

TESTBENCH :-

`timescale 1ns / 1ps

module cpu\_tb();

reg clk, reset;

reg [18:0] instruction;

wire [18:0] result;

cpu\_19bit uut (

.clk(clk),

.reset(reset),

.instruction(instruction),

.result(result));

initial clk = 0;

always #5 clk = ~clk;

initial begin

reset = 1;

#10 reset = 0;

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

#10;

$display("Result of ADD: %d", result);

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

#10;

$display("Result of SUB: %d", result);

instruction = 19'b00100\_00001\_00000\_00000;

#10;

$display("Result of INC: %d", result);

#10 $finish;

end

endmodule

OUTPUT :-

