module inst\_decoder(

input [15:0] instruction,

output [3:0] opcode,

output [1:0] rs\_addr,

output [1:0] rt\_addr,

output [1:0] rd\_addr,

output [7:0] immediate,

output reg RegDst,

output reg RegWrite,

output reg ALUSrc1,

output reg ALUSrc2,

output reg [2:0] ALUOp,

output reg MemWrite,

output reg MemToReg

);

assign opcode = instruction[15:12];

assign rs\_addr = instruction[11:10];

assign rt\_addr = instruction[9:8];

assign rd\_addr = instruction[7:6];

assign immediate = instruction[7:0];

always@(\*)

begin

case(opcode)

4'b0000:

begin

RegDst = 1'b0;

RegWrite = 1'b1;

ALUSrc1 = 1'b0;

ALUSrc2 = 1'b1;

ALUOp = 3'b000;

MemWrite = 1'b0;

MemToReg = 1'b1;

end

4'b0001:

begin

RegDst = 1'bx;

RegWrite = 1'b0;

ALUSrc1 = 1'b0;

ALUSrc2 = 1'b1;

ALUOp = 3'b000;

MemWrite = 1'b1;

MemToReg = 1'bx;

end

4'b0010:

begin

RegDst = 1'b1;

RegWrite = 1'b1;

ALUSrc1 = 1'b0;

ALUSrc2 = 1'b0;

ALUOp = 3'b000;

MemWrite = 1'b0;

MemToReg = 1'b0;

end

4'b0011:

begin

RegDst = 1'b0;

RegWrite = 1'b1;

ALUSrc1 = 1'b0;

ALUSrc2 = 1'b1;

ALUOp = 3'b000;

MemWrite = 1'b0;

MemToReg = 1'b0;

end

4'b0100:

begin

RegDst = 1'b1;

RegWrite = 1'b1;

ALUSrc1 = 1'b0;

ALUSrc2 = 1'b0;

ALUOp = 3'b001;

MemWrite = 1'b0;

MemToReg = 1'b0;

end

4'b0101:

begin

RegDst = 1'b1;

RegWrite = 1'b1;

ALUSrc1 = 1'b0;

ALUSrc2 = 1'b0;

ALUOp = 3'b010;

MemWrite = 1'b0;

MemToReg = 1'b0;

end

4'b0110:

begin

RegDst = 1'b0;

RegWrite = 1'b1;

ALUSrc1 = 1'b0;

ALUSrc2 = 1'b1;

ALUOp = 3'b010;

MemWrite = 1'b0;

MemToReg = 1'b0;

end

4'b0111:

begin

RegDst = 1'b1;

RegWrite = 1'b1;

ALUSrc1 = 1'b0;

ALUSrc2 = 1'b0;

ALUOp = 3'b011;

MemWrite = 1'b0;

MemToReg = 1'b0;

end

4'b1000:

begin

RegDst = 1'b0;

RegWrite = 1'b1;

ALUSrc1 = 1'b0;

ALUSrc2 = 1'b1;

ALUOp = 3'b011;

MemWrite = 1'b0;

MemToReg = 1'b0;

end

4'b1001:

begin

RegDst = 1'b1;

RegWrite = 1'b1;

ALUSrc1 = 1'b0;

ALUSrc2 = 1'b0;

ALUOp = 3'b100;

MemWrite = 1'b0;

MemToReg = 1'b0;

end

4'b1010:

begin

RegDst = 1'b1;

RegWrite = 1'b1;

ALUSrc1 = 1'b0;

ALUSrc2 = 1'b0;

ALUOp = 3'b101;

MemWrite = 1'b0;

MemToReg = 1'b0;

end

4'b1011:

begin

RegDst = 1'bx;

RegWrite = 1'b0;

ALUSrc1 = 1'b0;

ALUSrc2 = 1'b0;

ALUOp = 3'b110;

MemWrite = 1'b0;

MemToReg = 1'bx;

end

4'b1100:

begin

RegDst = 1'bx;

RegWrite = 1'b0;

ALUSrc1 = 1'b0;

ALUSrc2 = 1'b0;

ALUOp = 3'b111;

MemWrite = 1'b0;

MemToReg = 1'bx;

end

4'b1101:

begin

RegDst = 1'bx;

RegWrite = 1'b1;

ALUSrc1 = 1'b0;

ALUSrc2 = 1'b0;

ALUOp = 3'b010;

MemWrite = 1'b0;

MemToReg = 1'b0;

end

endcase

end

endmodule