**Implementation of the Control section of the single-cycle data path for MicroMIPS**

**Code:**

module controlunit(op,fn,clk,reg\_write,reg\_dst,reg\_insrc,ALU\_src,add\_sub,logic\_fn,fn\_class,data\_read,data\_write,brtype,PCsrc);

input [5:0]op,fn;

input clk;

output reg [1:0]reg\_dst,reg\_insrc,logic\_fn,fn\_class,brtype,PCsrc;

output reg reg\_write,ALU\_src,add\_sub,data\_read,data\_write;

always @(posedge clk)

begin

case({op,fn})

12'b001111xxxxxx:

begin

reg\_write <= 1'b1;

reg\_dst <= 2'b00;

reg\_insrc <= 2'b01;

ALU\_src <= 1'b1;

add\_sub <= 1'bx;

logic\_fn <= 2'bx;

fn\_class <= 2'b00;

data\_read <= 1'b0;

data\_write <= 1'b0;

brtype <= 2'b00;

PCsrc <=2'b00;

end

12'b000000100000:

begin

reg\_write <= 1'b1;

reg\_dst <= 2'b01;

reg\_insrc <= 2'b01;

ALU\_src <= 1'b0;

add\_sub <= 1'b0;

logic\_fn <= 2'bx;

fn\_class <= 2'b10;

data\_read <= 1'b0;

data\_write <= 1'b0;

brtype <= 2'b00;

PCsrc <=2'b00;

end

12'b000000100010:

begin

reg\_write <= 1'b1;

reg\_dst <= 2'b01;

reg\_insrc <= 2'b01;

ALU\_src <= 1'b0;

add\_sub <= 1'b1;

logic\_fn <= 2'bx;

fn\_class <= 2'b10;

data\_read <= 1'b0;

data\_write <= 1'b0;

brtype <= 2'b00;

PCsrc <=2'b00;

end

12'b000000101010:

begin

reg\_write <= 1'b1;

reg\_dst <= 2'b01;

reg\_insrc <= 2'b01;

ALU\_src <= 1'b0;

add\_sub <= 1'b1;

logic\_fn <= 2'bx;

fn\_class <= 2'b01;

data\_read <= 1'b0;

data\_write <= 1'b0;

brtype <= 2'b00;

PCsrc <=2'b00;

end

12'b001000xxxxxx:

begin

reg\_write <= 1'b1;

reg\_dst <= 2'b00;

reg\_insrc <= 2'b01;

ALU\_src <= 1'b1;

add\_sub <= 1'b0;

logic\_fn <= 2'bx;

fn\_class <= 2'b10;

data\_read <= 1'b0;

data\_write <= 1'b0;

brtype <= 2'b00;

PCsrc <=2'b00;

end

12'b001010xxxxxx:

begin

reg\_write <= 1'b1;

reg\_dst <= 2'b00;

reg\_insrc <= 2'b01;

ALU\_src <= 1'b1;

add\_sub <= 1'b1;

logic\_fn <= 2'bx;

fn\_class <= 2'b01;

data\_read <= 1'b0;

data\_write <= 1'b0;

brtype <= 2'b00;

PCsrc <=2'b00;

end

12'b000000100100:

begin

reg\_write <= 1'b1;

reg\_dst <= 2'b01;

reg\_insrc <= 2'b01;

ALU\_src <= 1'b0;

add\_sub <= 1'bx;

logic\_fn <= 2'b00;

fn\_class <= 2'b11;

data\_read <= 1'b0;

data\_write <= 1'b0;

brtype <= 2'b00;

PCsrc <=2'b00;

end

12'b000000100101:

begin

reg\_write <= 1'b1;

reg\_dst <= 2'b01;

reg\_insrc <= 2'b01;

ALU\_src <= 1'b0;

add\_sub <= 1'bx;

logic\_fn <= 2'b01;

fn\_class <= 2'b11;

data\_read <= 1'b0;

data\_write <= 1'b0;

brtype <= 2'b00;

PCsrc <=2'b00;

end

12'b000000100110:

begin

reg\_write <= 1'b1;

reg\_dst <= 2'b01;

reg\_insrc <= 2'b01;

ALU\_src <= 1'b0;

add\_sub <= 1'bx;

logic\_fn <= 2'b10;

fn\_class <= 2'b11;

data\_read <= 1'b0;

data\_write <= 1'b0;

brtype <= 2'b00;

PCsrc <=2'b00;

end

12'b000000100111:

begin

reg\_write <= 1'b1;

reg\_dst <= 2'b01;

reg\_insrc <= 2'b01;

ALU\_src <= 1'b0;

add\_sub <= 1'bx;

logic\_fn <= 2'b11;

fn\_class <= 2'b11;

data\_read <= 1'b0;

data\_write <= 1'b0;

brtype <= 2'b00;

PCsrc <=2'b00;

end

12'b001100xxxxxx:

begin

reg\_write <= 1'b1;

reg\_dst <= 2'b00;

reg\_insrc <= 2'b01;

ALU\_src <= 1'b1;

add\_sub <= 1'bx;

logic\_fn <= 2'b00;

fn\_class <= 2'b11;

data\_read <= 1'b0;

data\_write <= 1'b0;

brtype <= 2'b00;

PCsrc <=2'b00;

end

12'b001101xxxxxx:

begin

reg\_write <= 1'b1;

reg\_dst <= 2'b00;

reg\_insrc <= 2'b01;

ALU\_src <= 1'b1;

add\_sub <= 1'b1;

logic\_fn <= 2'b01;

fn\_class <= 2'b11;

data\_read <= 1'b0;

data\_write <= 1'b0;

brtype <= 2'b00;

PCsrc <=2'b00;

end

12'b001110xxxxxx:

begin

reg\_write <= 1'b1;

reg\_dst <= 2'b00;

reg\_insrc <= 2'b01;

ALU\_src <= 1'b1;

add\_sub <= 1'b1;

logic\_fn <= 2'b10;

fn\_class <= 2'b11;

data\_read <= 1'b0;

data\_write <= 1'b0;

brtype <= 2'b00;

PCsrc <=2'b00;

end

12'b100011xxxxxx:

begin

reg\_write <= 1'b1;

reg\_dst <= 2'b00;

reg\_insrc <= 2'b00;

ALU\_src <= 1'b1;

add\_sub <= 1'b0;

logic\_fn <= 2'bx;

fn\_class <= 2'b10;

data\_read <= 1'b1;

data\_write <= 1'b0;

brtype <= 2'b00;

PCsrc <=2'b00;

end

12'b101011xxxxxx:

begin

reg\_write <= 1'b0;

reg\_dst <= 2'bx;

reg\_insrc <= 2'bx;

ALU\_src <= 1'b1;

add\_sub <= 1'b0;

logic\_fn <= 2'bx;

fn\_class <= 2'b10;

data\_read <= 1'b0;

data\_write <= 1'b1;

brtype <= 2'b00;

PCsrc <=2'b00;

end

12'b000010xxxxxx:

begin

reg\_write <= 1'b0;

reg\_dst <= 2'bx;

reg\_insrc <= 2'bx;

ALU\_src <= 1'bx;

add\_sub <= 1'bx;

logic\_fn <= 2'bx;

fn\_class <= 2'bx;

data\_read <= 1'b0;

data\_write <= 1'b0;

brtype <= 2'bx;

PCsrc <=2'b01;

end

12'b000000001000:

begin

reg\_write <= 1'b0;

reg\_dst <= 2'bx;

reg\_insrc <= 2'bx;

ALU\_src <= 1'bx;

add\_sub <= 1'bx;

logic\_fn <= 2'bx;

fn\_class <= 2'bx;

data\_read <= 1'b0;

data\_write <= 1'b0;

brtype <= 2'bx;

PCsrc <=2'b10;

end

12'b000001xxxxxx:

begin

reg\_write <= 1'b0;

reg\_dst <= 2'bx;

reg\_insrc <= 2'bx;

ALU\_src <= 1'bx;

add\_sub <= 1'bx;

logic\_fn <= 2'bx;

fn\_class <= 2'bx;

data\_read <= 1'b0;

data\_write <= 1'b0;

brtype <= 2'b11;

PCsrc <=2'b00;

end

12'b000100xxxxxx:

begin

reg\_write <= 1'b0;

reg\_dst <= 2'bx;

reg\_insrc <= 2'bx;

ALU\_src <= 1'bx;

add\_sub <= 1'bx;

logic\_fn <= 2'bx;

fn\_class <= 2'bx;

data\_read <= 1'b0;

data\_write <= 1'b0;

brtype <= 2'b01;

PCsrc <=2'b00;

end

12'b000101xxxxxx:

begin

reg\_write <= 1'b0;

reg\_dst <= 2'bx;

reg\_insrc <= 2'bx;

ALU\_src <= 1'bx;

add\_sub <= 1'bx;

logic\_fn <= 2'bx;

fn\_class <= 2'bx;

data\_read <= 1'b0;

data\_write <= 1'b0;

brtype <= 2'b10;

PCsrc <=2'b00;

end

12'b000011xxxxxx:

begin

reg\_write <= 1'b1;

reg\_dst <= 2'b10;

reg\_insrc <= 2'b10;

ALU\_src <= 1'bx;

add\_sub <= 1'bx;

logic\_fn <= 2'bx;

fn\_class <= 2'bx;

data\_read <= 1'b0;

data\_write <= 1'b0;

brtype <= 2'b00;

PCsrc <=2'b01;

end

12'b000000001100:

begin

reg\_write <= 1'b0;

reg\_dst <= 2'bx;

reg\_insrc <= 2'bx;

ALU\_src <= 1'bx;

add\_sub <= 1'bx;

logic\_fn <= 2'bx;

fn\_class <= 2'bx;

data\_read <= 1'b0;

data\_write <= 1'b0;

brtype <= 2'bx;

PCsrc <=2'b11;

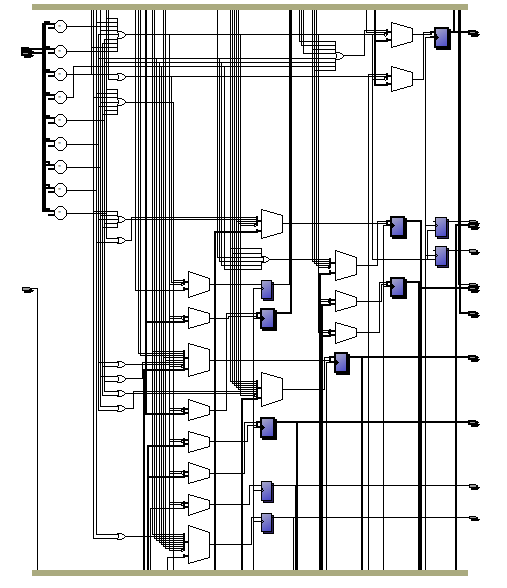
end

endcase

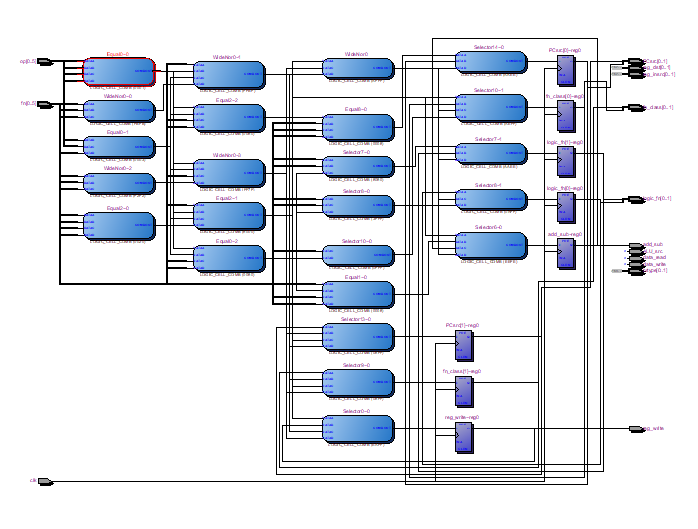
end

endmodule

**RTL view:**



**Technology Mapping View:**



**RTL Simulation:**

