```
`timescale 1ns / 1ps
module
controlunit
(ALU_OP,
read_reg1,
read_reg2,
write_reg,
branch,
jump,
ALU_IMM,
sel_imm,
write_data,
read_data,
opcode);

output reg ALU_OP,read_reg1,read_reg2,write_reg,branch,jump,
write_data,read_data,ALU_IMM;
output reg [1:0] sel_imm;
input [6:0] opcode;


always@(opcode)
begin
case(opcode)
7'b0110011:          // R -Type
begin
ALU_OP = 1'b1;
read_reg1 = 1'b1;
read_reg2 = 1'b1;
write_reg = 1'b0;
jump = 1'b0;
branch = 1'b0;
ALU_IMM = 1'b0;
write_data = 1'b0;
read_data = 1'b0;
sel_imm = 2'b00;
end
7'b0010011: // I type
begin
ALU_OP = 1'b1;
read_reg1 = 1'b1;
read_reg2 = 1'b0;
write_reg = 1'b1;
jump = 1'b0;
branch = 1'b0;
ALU_IMM = 1'b1;
```

```
write_data = 1'b0;
read_data = 1'b1;
sel_imm = 2'b01;
end
7'b1100011: // B type
begin
ALU_OP = 1'b0;
read_reg1 = 1'b0;
read_reg2 = 1'b0;
write_reg = 1'b0;
jump = 1'b0;
branch = 1'b0;
ALU_IMM = 1'b0;
write_data = 1'b0;
read_data = 1'b0;
sel_imm = 2'b10;
end
7'b0100011:  // S-TYPE
begin
ALU_OP = 1'b1;
read_reg1 = 1'b1;
read_reg2 = 1'b0;
write_reg = 1'b0;
jump = 1'b0;
branch = 1'b0;
ALU_IMM = 1'b1;
write_data = 1'b1;
read_data = 1'b0;
sel_imm = 2'b11;
end
7'b0110111:    // lui
begin
ALU_OP = 1'b0;
read_reg1 = 1'b0;
read_reg2 = 1'b0;
write_reg = 1'b0;
jump = 1'b0;
branch = 1'b0;
ALU_IMM = 1'b0;
write_data = 1'b0;
read_data = 1'b0;
sel_imm = 2'b00;
end
7'b0010111:    // auipc
begin
ALU_OP = 1'b0;
```

```
read_reg1 = 1'b0;
read_reg2 = 1'b0;
write_reg = 1'b0;
jump = 1'b0;
branch = 1'b0;
ALU_IMM = 1'b0;
write_data = 1'b0;
read_data = 1'b0;
sel_imm = 2'b00;
end
default:
begin
ALU_OP = 1'b0;
read_reg1 = 1'b0;
read_reg2 = 1'b0;
write_reg = 1'b0;
jump = 1'b0;
branch = 1'b0;
ALU_IMM = 1'b0;
write_data = 1'b0;
read_data = 1'b0;
sel_imm = 2'b00;
end
endcase
end
endmodule
```