



```
1 `timescale 1ns/1ns
2
3 `define idle 3'b000
4 `define init 3'b001
5 `define read 3'b010
6 `define reg_write 3'b011
7 `define cal 3'b100
8 `define write 3'b101
9 `define done 3'b110
10
11 module controller(clk, rst, line_index, start, read_file, write_reg, write_file, finish);
12
13     input clk, start;
14
15     output reg read_file, write_reg, write_file, finish;
16     output reg rst;
17     output [5:0] line_index;
18
19     reg [2:0] pstate = `idle, nstate;
20     reg cnt_inc;
21
22     reg [5:0] counter;
```

```

always @(pstate, start, counter) begin
    {nstate, rst, read_file, write_reg, cnt_inc, write_file, finish} = 9'b0;
    case (pstate)
        `idle: begin nstate = (start) ? `init: `idle; end
        `init: begin nstate = `read; rst = 1'b1; read_file = 1'b1; end
        `read: begin nstate = `reg_write; end
        `reg_write: begin nstate = `cal; write_reg = 1'b1; cnt_inc = 1'b1; end
        `cal: begin nstate = `write; end
        `write: begin nstate = (counter == 0) ? `done : `reg_write; write_file = 1'b1; end
        `done: begin nstate = `idle; finish = 1'b1; end
        default: nstate = `idle; // pstate
    endcase
end

always @(posedge clk) begin
    pstate <= nstate;
end

assign line_index = counter;

always @(posedge clk) begin
    if (rst) begin
        counter <= 0;
    end
    else if (cnt_inc) begin
        counter <= counter + 1;
    end
end

endmodule

```