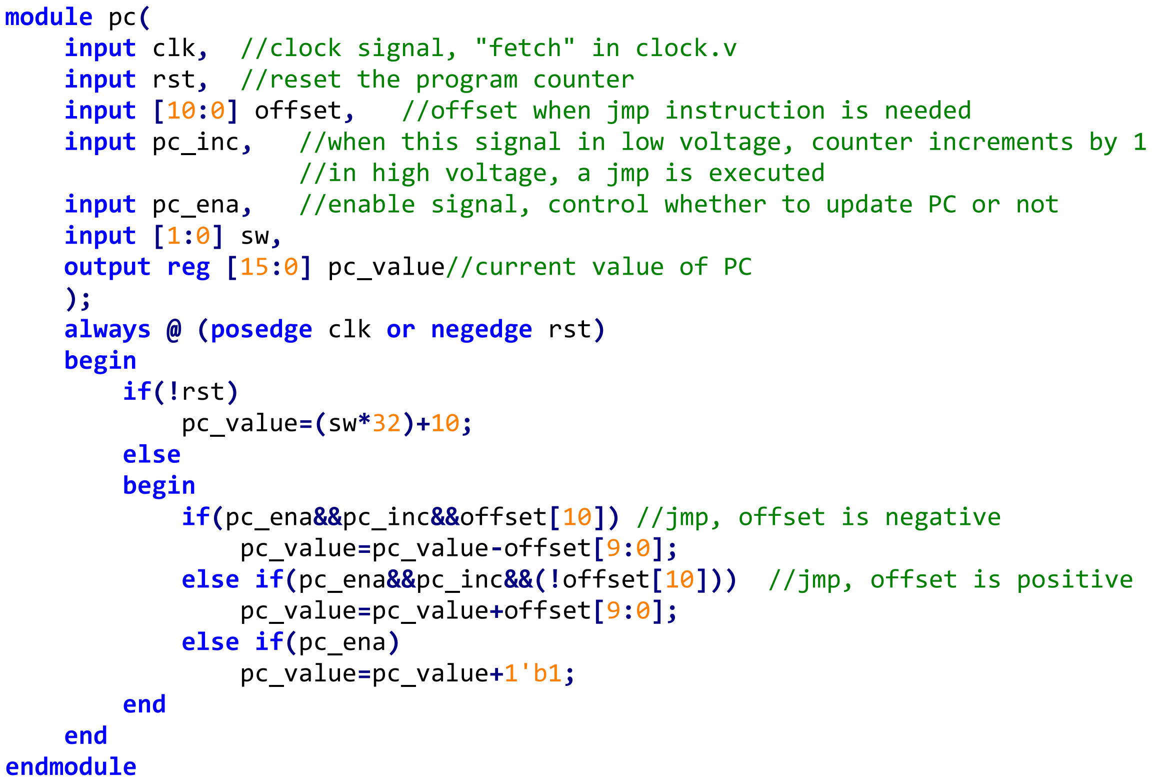

```
module pc(
    input clk,  //clock signal, "fetch" in clock.v
    input rst,  //reset the program counter
    input [10:0] offset,   //offset when jmp instruction is needed
    input pc_inc,   //when this signal in low voltage, counter increments by 1
                    //in high voltage, a jmp is executed
    input pc_ena,   //enable signal, control whether to update PC or not
    input [1:0] sw,
    output reg [15:0] pc_value//current value of PC
    );
    always @ (posedge clk or negedge rst)
    begin
        if(!rst)
            pc_value=(sw*32)+10;
        else
        begin
            if(pc_ena&&pc_inc&&offset[10]) //jmp, offset is negative
                pc_value=pc_value-offset[9:0];
            else if(pc_ena&&pc_inc&&(!offset[10]))  //jmp, offset is positive
                pc_value=pc_value+offset[9:0];
            else if(pc_ena)
                pc_value=pc_value+1'b1;
        end
    end
endmodule
```