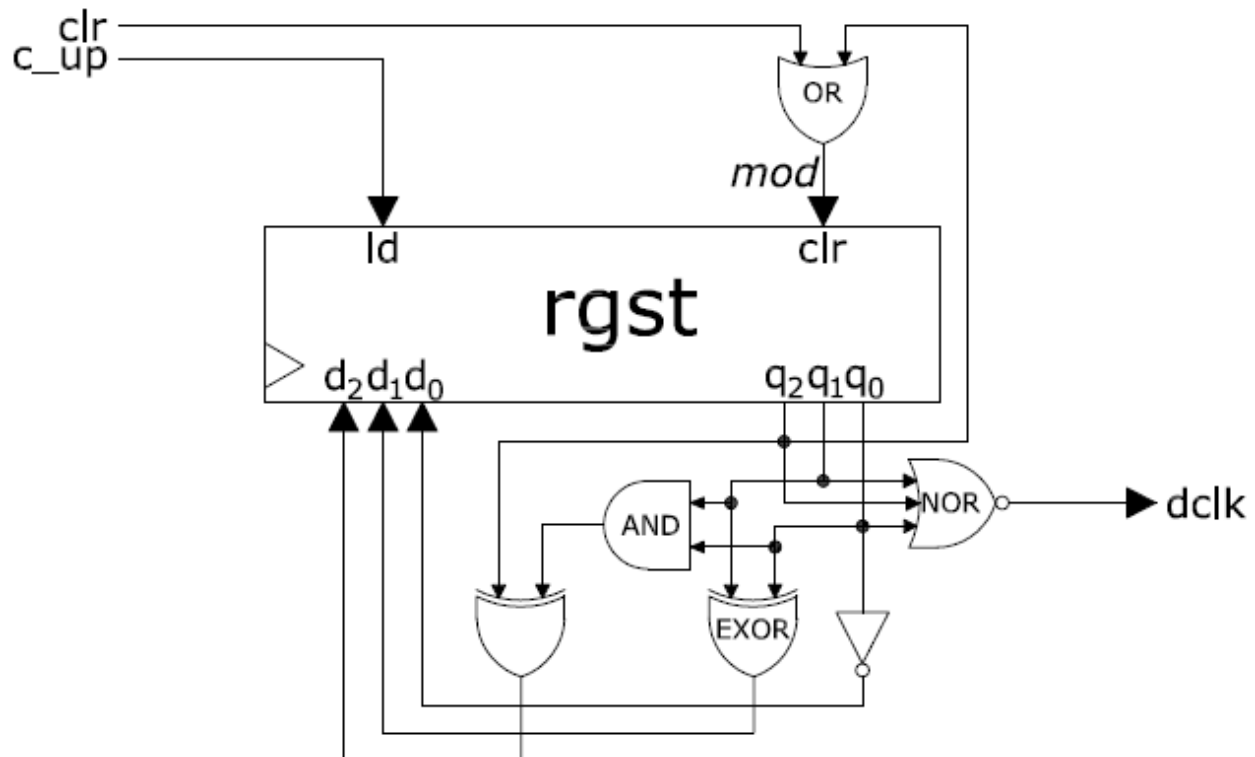


Laborator S9 AC

Codificarea One Hot pentru implementarea FSM-urilor

P.9.1 Implementați, utilizând limbajul Verilog, un numărător divide-by-5 descris în arhitectura de mai jos:



// Soluție:

```

module register # (
  parameter w=3,
  parameter iv= { w { 1'b0} } )
  ( input [w-1:0] d,
    input rst_b, clk, ld, clr,
    output reg [w-1:0]q
  );
  
```

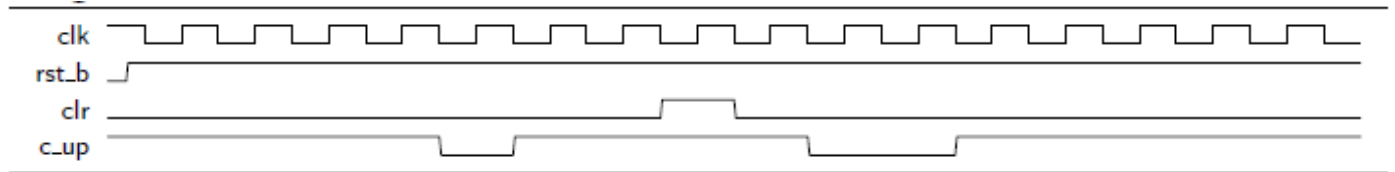
```

always @ (posedge clk, negedge rst_b)
begin
    if ((clr==1) || (!rst_b) )
        q<=iv;
    else if (ld==1)
        q<=d;
    end
endmodule

// Instantierea
module d5cntr(
    input clk, rst_b, c_up, clr,
    output dclk
);
    wire mod;
    wire [2:0]q;
    assign mod= clr | q[2];
    register #(
        .w(3),
        .iv(3'd0))
    register1 (
        .d( { q[2]^ (q[1]&q[0]) , (q[1] ^q[0]) , ~q[0] } ),
        .clk(clk), .ld(c_up), .rst_b(rst_b), .clr(mod),
        .q(q));
    assign dclk= ~(q[2] | q[1] | q[0]);
endmodule

```

P.9.2 Verificați modulul *d5cntr* cu un testbench care generează intrările precum este ilustrat în cronograma de mai jos:



```
// Soluție:

module d5cntr_tb (
    output reg clk,
    output reg rst_b, //asynch
    output reg clr,
    output reg c_up,
    output dclk
);
    d5cntr d5( .clk(clk),
    .rst_b(rst_b),
    .clr(clr),
    .c_up(c_up),
    .dclk(dclk));
    initial begin
        clk=1'd1;
        repeat(200) #20 clk=~clk;
    end
    initial begin
        rst_b=1'd0;
        #10 rst_b=1'd1;
    end
end
```

```

initial begin

    clr=1'd0;

    #300 clr=1'd1;

    #40 clr=1'd0;

end

initial begin

    c_up=1'd1;

    #180 c_up=1'd0;

    #40 c_up=1'd1;

    #160 c_up=1'd0;

    #80 c_up=1'd1;

end

endmodule

```

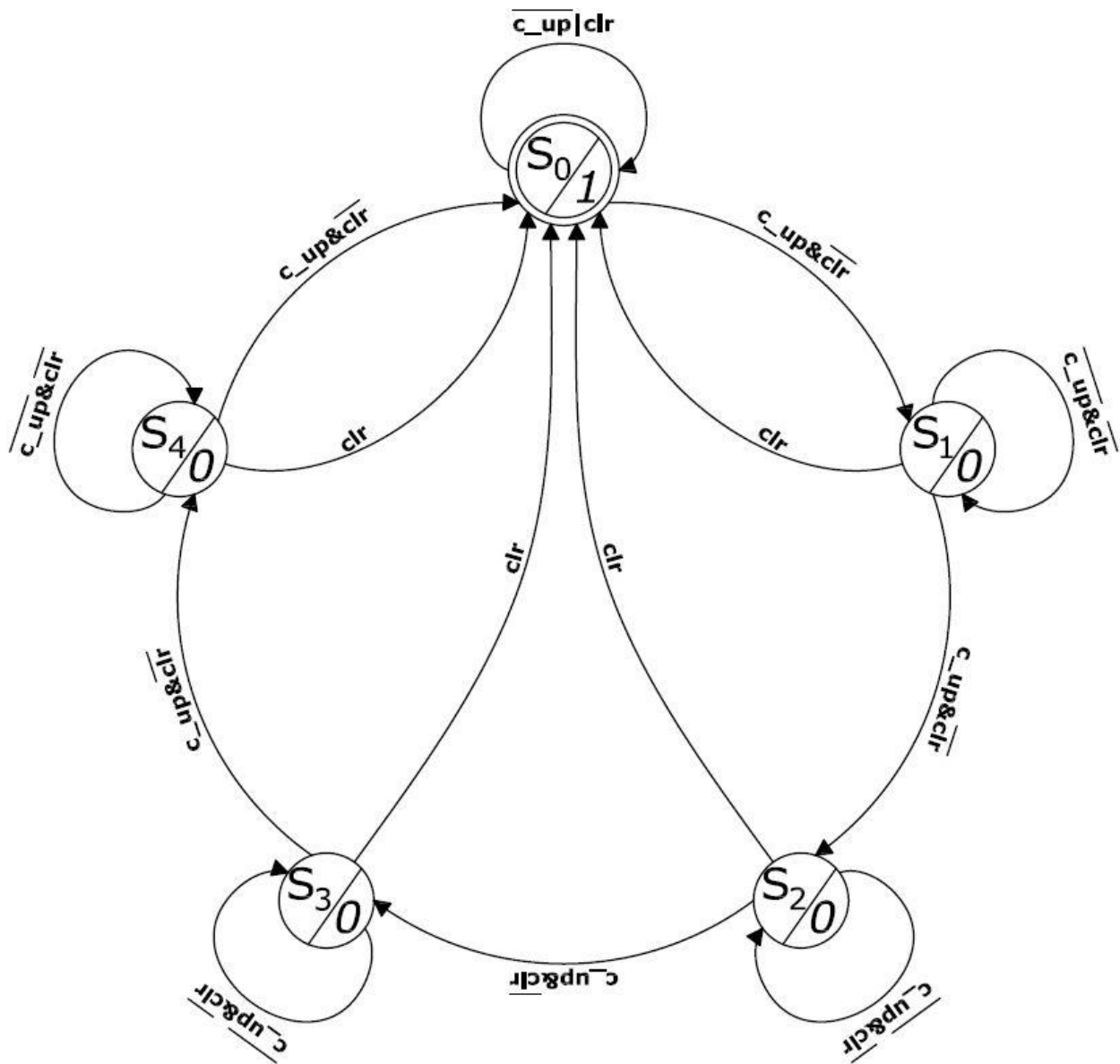
P.9.3 Construiți, folosind limbajul Verilog, diagrama de tranziții corespunzătoare numărătorului divide-by-5 descrisă în figura de mai jos:

```

// Soluție:

module fsm_d5cntr(
    input clk,
    input rst_b, //asynch
    input clr,
    input c_up,
    output dclk
);

```



```
// Declararea registrelor pentru pastrarea starii curente si starii urmatoare
reg [4:0] din;
wire [4:0] din_nxt;

// Scrierea ecuatiilor booleene pentru fiecare bistabil de tip D
assign din_nxt[1] = (din[0] & c_up & (~clr)) | (din[1] & (~c_up) & (~clr)) ;
assign din_nxt[2] = (din[1] & c_up & (~clr)) | (din[2] & (~c_up) & (~clr)) ;
assign din_nxt[3] = (din[2] & c_up & (~clr)) | (din[3] & (~c_up) & (~clr)) ;
assign din_nxt[4] = (din[3] & c_up & (~clr)) | (din[4] & (~c_up) & (~clr)) ;
assign din_nxt[0] = (din[4] & c_up & (~clr)) | (din[0] & (~c_up) | clr) | (din[1] & clr) | (din[2]
& clr) | (din[3] & clr) | (din[4] & clr) ;
assign dclk = din[0];
always @ (posedge clk, negedge rst_b)
    if (! rst_b) din<=5'd1; // starea implicita va fi S0
    else din <= din_nxt;
endmodule
```