a) fsm_6s:

D:/ModelsimProjects/fsm_6s.v

```
Ln#
 1  module fsm_6s (
 2    input msg_bgn, brdr_idle, pkt_rdy, lst_pkt, clk, rst,
 3    input [4:0] ldx,
 4    output reg clr, idle, inc_mgln, c_up, st_pkt, blk_rdy, hi_mgln, msg_end, lo_mgln, zero_pkt, pad_pkt
 5    );
 6
 7    // stari
 8    localparam INIT = 3'd0;
 9    localparam BGN = 3'd1;
10    localparam RX_PKT = 3'd2;
11    localparam PAD = 3'd3;
12    localparam ZERO = 3'd4;
13    localparam TX_MSG = 3'd5;
14
15    // semnale de tip registru
16    reg [2:0] st;
17    reg [2:0] st_next;
18
19    // starea urmatoare
20
21    always @ (*) begin
22      st_next = INIT;
23      case (st)
24        INIT: if(!msg_bgn)
25                  st_next = INIT;
26              else st_next = BGN;
27        BGN: if(!pkt_rdy)
28                  st_next = BGN;
29                else if(pkt_rdy && !lst_pkt)
30                        st_next = RX_PKT;
31                else st_next = PAD;
32        RX_PKT: if(ldx!=0 && pkt_rdy && lst_pkt) st_next = PAD;
33                else if(ldx == 0 && brdr_idle && pkt_rdy && lst_pkt) st_next = PAD;
34                else st_next = RX_PKT;
35        PAD:    if(ldx==0 && !brdr_idle) st_next = PAD;
36              else if(ldx!=0) st_next = ZERO;
37              else st_next = ZERO;
38        ZERO: if(ldx==14) st_next = TX_MSG;
39            else st_next = ZERO;
40        TX_MSG:
41              if(ldx==0 && !brdr_idle) st_next = TX_MSG;
42              else if(ldx!=0) st_next = TX_MSG;
43              else if(ldx==0 && brdr_idle) st_next = INIT;
44          endcase
45        end
```

```
46
47          always @(*) begin
48            blk_rdy = 1'b0;
49            c_up = 1'b0;
50            idle = 1'b0;
51            inc_mgln = 1'b0;
52            clr = 1'b0;
53            st_pkt = 1'b0;
54            zero_pkt = 1'b0;
55            pad_pkt = 1'b0;
56            lo_mgln = 1'b0;
57            hi_mgln = 1'b0;
58            msg_end = 1'b0;
59
60            case(st)
61
62              INIT: if(msg_bgn) clr = 1'b1;
63              BGN: if (!pkt_rdy) idle = 1'b1;
64                   else begin
65                     idle = 1'b1;
66                     inc_mgln = 1'b1;
67                     c_up = 1'b1;
68                     st_pkt = 1'b1;
69                   end
70              RX_PKT: if(ldx==0 && brdr_idle && pkt_rdy && !lst_pkt) begin
71                           idle = 1'b1;
72                           blk_rdy = 1'b1;
73                           inc_mgln = 1'b1;
74                           c_up = 1'b1;
75                           st_pkt = 1'b1;
76                         end
```

```
    end
else if (ldx!=0 && pkt_rdy &&!lst_pkt) begin
    idle=1'b1;
    inc_mgln=1'b1;
    c_up=1'b1;
    st_pkt=1'b1;
    end
else if(ldx!=0 && !pkt_rdy) idle=1'b1;
else if(ldx!=0 && pkt_rdy && lst_pkt) begin
    idle=1'b1;
    inc_mgln=1'b1;
    c_up=1'b1;
    st_pkt=1'b1;
    end
else if(ldx==0 && brdr_idle && pkt_rdy && lst_pkt) begin
    blk_rdy=1'b1;
    idle=1'b1;
    inc_mgln=1'b1;
    c_up=1'b1;
    st_pkt=1'b1;
    end
```

```
        PAD: if (ldx!=0) begin
                c_up=1'b1;
                pad_pkt=1'b1;
                st_pkt=1'b1;
             end
             else if(ldx==0 && brdr_idle) begin
                blk_rdy=1'b1;
                c_up=1'b1;
                pad_pkt=1'b1;
                st_pkt=1'b1;
             end
        ZERO: if (ldx==14) begin
                c_up=1'b1;
                hi_mgln=1'b1;
                st_pkt=1'b1;
             end
           else if (ldx==0 && brdr_idle) begin
                blk_rdy=1'b1;
                c_up=1'b1;
                zero_pkt=1'b1;
                st_pkt=1'b1;
             end
           else if (ldx!=14 && ldx!=0) begin
                c_up=1'b1;
                zero_pkt=1'b1;
                st_pkt=1'b1;
            end
        TX_MSG: if (ldx!=0) begin
                  c_up=1'b1;
                  lo_mgln=1'b1;
                  st_pkt=1'b1;
                end
             else if (ldx==0 && brdr_idle) begin
                  blk_rdy=1'b1;
                  msg_end=1'b1;
             end
          endcase
    end

    always @(posedge clk, negedge rst) begin
      if(!rst)st<=INIT;
      else
        st<=st_nxt;
     end

    endmodule
```

b) Testbench - fsm_6s_tb:

```
:/ModelsimProjects/fsm_6s_tb.v *
#
module fsm_6s_tb(
  output reg msg_bgn, lst_pkt, pkt_rdy, brdr_idle, clk, rst,
  output reg[3:0] ldx,
  output blk_rdy, idle, clr, c_up, st_pkt, zero_pkt, pad_pkt, inc_mgln, lo_mgln, hi_mgln, msg_end
  );
```

```
fsm_6s DUT(
  .msg_bgn(msg_bgn),
  .pkt_rdy(pkt_rdy),
  .lst_pkt(lst_pkt),
  .brdr_idle(brdr_idle),
  .clk(clk),
  .rst(rst),
  .ldx(ldx),
  .blk_rdy(blk_rdy),
  .idle(idle),
  .inc_mgln(inc_mgln),
  .c_up(c_up),
  .st_pkt(st_pkt),
  .clr(clr),
  .zero_pkt(zero_pkt),
  .pad_pkt(pad_pkt),
  .lo_mgln(lo_mgln),
  .hi_mgln(hi_mgln),
  .msg_end(msg_end)
);
initial begin
  clk=0;
  forever #5 clk=~clk;
end

initial begin
  rst=1;
end

initial begin
  msg_bgn=0;
  pkt_rdy=0;
  lst_pkt=0;
  ldx=4'd0;

  #50 msg_bgn=~msg_bgn;
  #50 pkt_rdy=~pkt_rdy;
      lst_pkt=~lst_pkt;
  #50 ldx=4'd1;
  #50 ldx=4'd14;
  #50 ldx=4'd0;
      brdr_idle=~brdr_idle;
end
endmodule
```