```verilog
module pattern (input clk, input rst, input sig, output out);

localparam  S0 = 3'd0;
localparam  S1 = 3'd1;
localparam  S2 = 3'd2;
localparam  S4 = 3'd3;
localparam  S9 = 3'd4;
localparam  S18 = 3'd5;

reg [2:0] CurrentState;
reg [2:0] NextState;

assign out = (CurrentState == S18);

always @(posedge clk) begin
    if (rst) begin
        CurrentState <= S0;
    end else begin
        CurrentState <= NextState;
    end
end

always @(*) begin
    NextState = CurrentState;
    case (CurrentState)
        S0: begin
            if (sig == 1'b1) NextState = S1;
        end
        S1: begin
            if (sig == 1'b0) NextState = S2;
        end
        S2: begin
            if (sig == 1'b0) NextState = S4;
            else NextState = S1;
        end
        S4: begin
            if (sig == 1'b1) NextState = S9;
            else NextState = S0;
        end
        S9: begin
            if (sig == 1'b0) NextState = S18;
            else NextState = S1;
        end
        S18: begin
            if (sig == 1'b1) NextState = S1;
            else NextState = S0;
        end
    endcase
end
endmodule
```