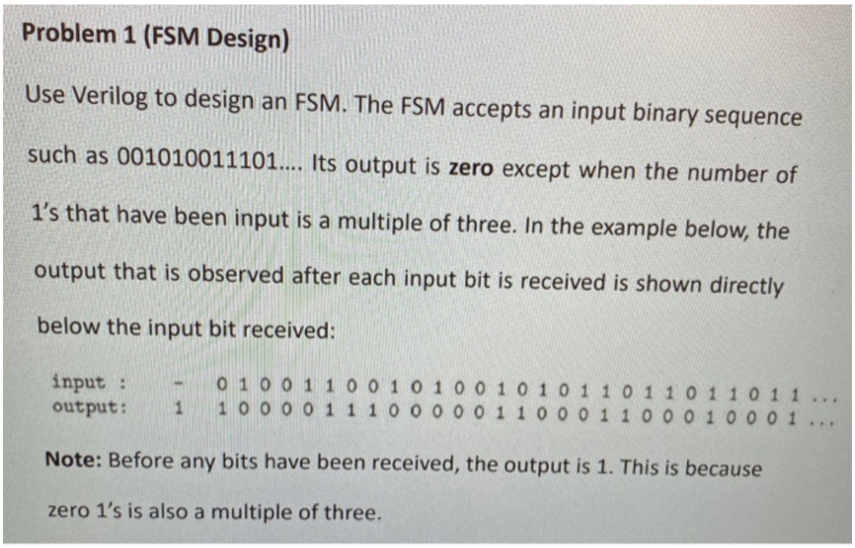
Question: Problem 1 (FSM Design) Use Verilog to design an FSM. The F·



Problem 1 (FSM Design)

Use Verilog to design an FSM. The FSM accepts an input binary sequence such as 001010011101.... Its output is **zero** except when the number of 1's that have been input is a multiple of three. In the example below, the output that is observed after each input bit is received is shown directly below the input bit received:

```
input :   -  0 1 0 0 1 1 0 0 1 0 1 0 0 1 0 1 0 1 1 0 1 1 0 1 1 0 1 1 ...
output:   1  1 0 0 0 0 1 1 1 0 0 0 0 0 1 1 0 0 0 1 1 0 0 0 1 0 0 0 1 ...
```

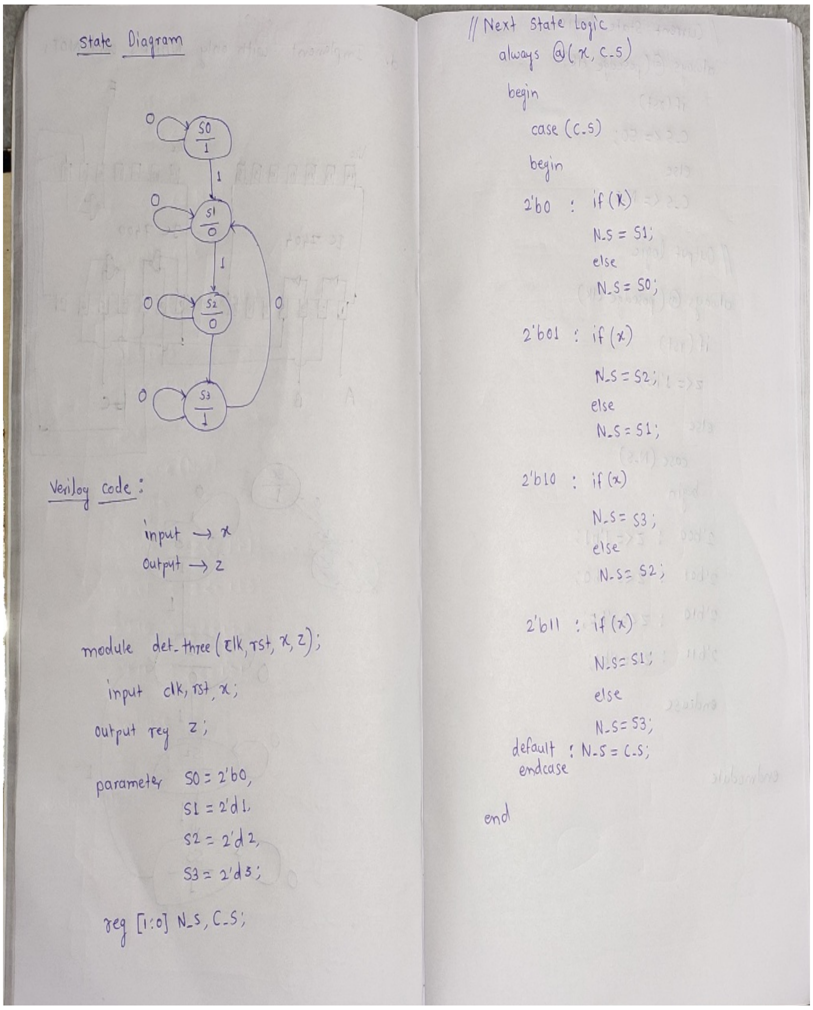
Note: Before any bits have been received, the output is 1. This is because zero 1's is also a multiple of three.

Show transcribed image text

Expert Answer

Anonymous answered this
294 answers

Was this answer helpful?



State Diagram

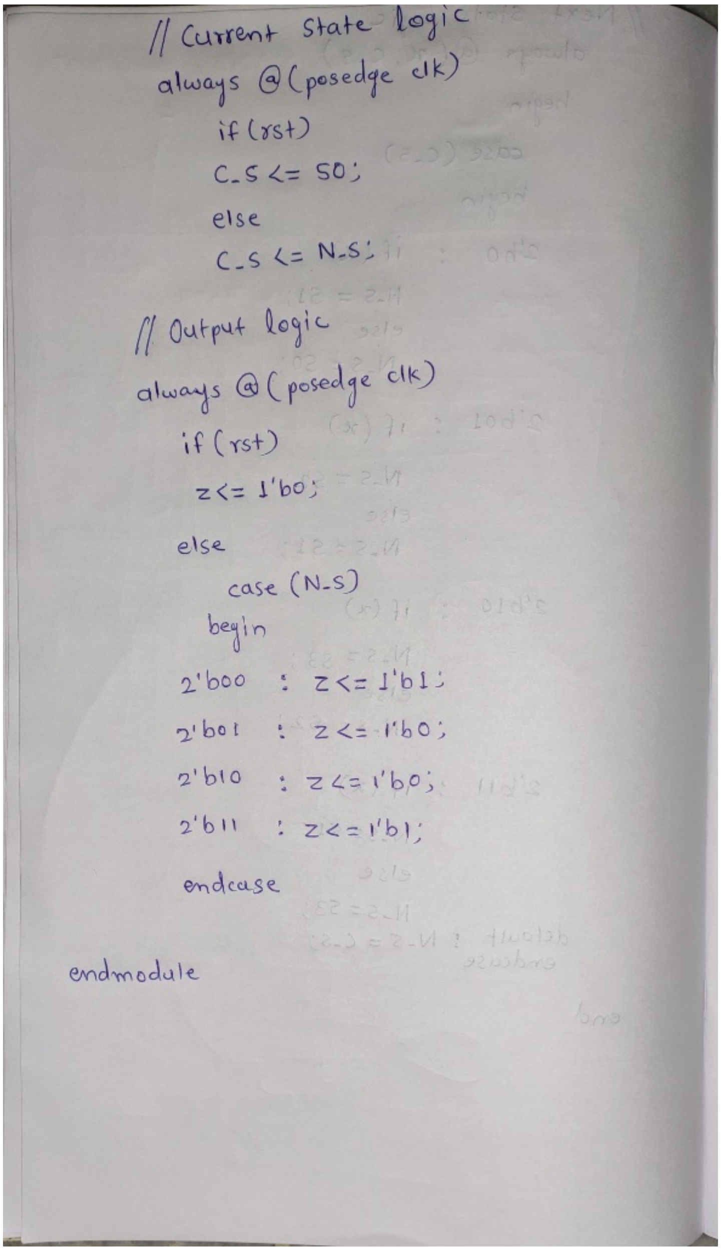
S0
1
S1
0
S2
0
S3
1
0
0
0
0
1
1
0

Verilog code :

input → x
output → z

```
module det_three (clk, rst, x, z);
  input clk, rst, x;
  output reg z;
  parameter S0 = 2'b0,
            S1 = 2'd1,
            S2 = 2'd2,
            S3 = 2'd3;
  reg [1:0] N_S, C_S;
```

```
// Next State Logic
 always @(x, C_S)
  begin
    case (C_S)
    begin
    2'b0  : if (x)
              N_S = S1;
              else
               N_S = S0;
    2'b01 : if (x)
              N_S = S2;
              else
               N_S = S1;
    2'b10 : if (x)
              N_S = S3;
              else
               N_S = S2;
    2'b11 : if (x)
              N_S = S1;
              else
              N_S = S3;
   default : N_S = C_S;
   endcase
 end
```



```
// Current State logic
always @(posedge clk)
    if (rst)
    C_S <= S0;
    else
    C_S <= N_S;

// Output logic
always @(posedge clk)
   if (rst)
    z<= 1'b0;
   else
       case (N_S)
     begin
   2'b00  :  z<= 1'b1;
   2'b01  :  z<= 1'b0;
   2'b10  :  z<= 1'b0;
   2'b11  :  z<= 1'b1;

   endcase

endmodule
```

Comment ›

Your answer

Add your answer