```
1   // Pattern counter display module
2
3   module patternCounterDisplay (pattern_count, HEX4, HEX5);
4       input logic [3:0] pattern_count;
5       output logic [6:0] HEX4, HEX5;
6
7       // Pattern Counter Display Logic
8       always_comb begin
9           // Display pattern_count on HEX4 and HEX5
10          HEX4 = getHexDigit(pattern_count % 10);
11          HEX5 = getHexDigit(pattern_count / 10);
12      end
13
14      function logic [6:0] getHexDigit(input logic [3:0] digit);
15          case (digit)
16              4'd0: getHexDigit = 7'b1000000;
17              4'd1: getHexDigit = 7'b1111001;
18              4'd2: getHexDigit = 7'b0100100;
19              4'd3: getHexDigit = 7'b0110000;
20              4'd4: getHexDigit = 7'b0011001;
21              4'd5: getHexDigit = 7'b0010010;
22              4'd6: getHexDigit = 7'b0000010;
23              4'd7: getHexDigit = 7'b1111000;
24              4'd8: getHexDigit = 7'b0000000;
25              4'd9: getHexDigit = 7'b0010000;
26              default: getHexDigit = 7'b1111111; // Error state
27          endcase
28      endfunction
29  endmodule
30
31
32
33  module patternCounterDisplay_testbench();
34      logic clk;
35      logic [3:0] pattern_count;
36      logic [6:0] HEX4, HEX5;
37
38      patternCounterDisplay dut (pattern_count, HEX4, HEX5);
39
40      // Set up a simulated clock.
41      parameter CLOCK_PERIOD = 100;
42      initial begin
43          clk <= 0;
44          forever #(CLOCK_PERIOD/2) clk <= ~clk; // Forever toggle the clock
45      end
46
47      // Test the design.
48      initial begin
49          // Test case 1: pattern_count = 0
50          pattern_count = 4'd0; @(posedge clk); @(posedge clk);
51
52          // Test case 2: pattern_count = 5
53          pattern_count = 4'd5; @(posedge clk); @(posedge clk);
54
55          // Test case 3: pattern_count = 9
56          pattern_count = 4'd9; @(posedge clk); @(posedge clk);
57
58          // Test case 4: pattern_count = 10
59          pattern_count = 4'd10; @(posedge clk); @(posedge clk);
60
61          // Test case 5: pattern_count = 15
62          pattern_count = 4'd15; @(posedge clk); @(posedge clk);
63
64          // Test case 6: pattern_count = 20
65          pattern_count = 4'd20; @(posedge clk); @(posedge clk);
66
67          $stop;
68      end
69  endmodule
70
```