****

**module fsm\_counter(count, clock);**

**input clock;**

**output [2:0] count;**

**reg [2:0] count;**

**//use parameter statement to define states**

**parameter [2:0] A = 3'b000, B = 3'b100, C = 3'b111, D = 3'b010, E = 3'b011;**

**always @ (posedge clock)**

**//use case statement to implement state transitions**

**case(count)**

**A: count <= B;**

**B: count <= C;**

**C: count <= D;**

**D: count <= E;**

**E: count <= A;**

**default: count <= C;**

**endcase**

**endmodule**

**module test1\_fsm\_counter;**

**reg clock;**

**wire [2:0] count;**

**fsm\_counter dut (count, clock);**

**initial begin**

**clock = 0;**

**end**

**always begin**

**#30 clock = ~clock;**

**end**

**endmodule**

**module test2\_fsm\_counter;**

**reg clock;**

**wire [2:0] count;**

**fsm\_counter dut (count, clock);**

**initial begin**

**clock = 0;**

**end**

**always begin**

**#5 clock = ~clock;**

**end**

**initial begin**

**$display ("\*\*\*\*\*\*\*\*\*\*\*Time-----Output");**

**$monitor ($time, " Count = %d", count);**

**end**

**endmodule**