11th Day

4. Write an RTL code to design the following pattern generator

0000_0001

0001_0000

0000_0001

0010_0000

0000_0001

0100_0000

0000_0001

1000_0000

and repeat. Under reset condition the initial output should be 0000_0001

Design

```
module pattern_gen( input clk,rst, output [7:0]out);
reg count;
reg [7:0]a,c;
always@(posedge clk)
  begin
    if(rst)
      begin
        c<={4'd0,4'd1};
        count<=0;
        a<={4'd1,4'd0};
      end
```

```
        else
          begin
             count<=count+1;
              if(count)
                 a<= {a[6:4],a[7],a[3:0]};
           end
     end
  assign out = (count) ? a : c ;
endmodule
```

Testbench

```
module pattern_gen_tb;
   reg clk, rst;
   wire [7:0] out;

   pattern_gen dut (.clk(clk),.rst(rst),.out(out) );

   // Clock generation
   always #5 clk = ~clk;

   initial
   begin

      clk = 0;
      rst = 1;
      #10;

      rst = 0;
      #100;
```

```
    $finish;
  end

  initial
  begin
    $monitor("Time = %0t | out = %b", $time, out);
  end
endmodule
```

Simulation

```
Time = 0 | out = xxxxxxxx
Time = 5000 | out = 00000001
Time = 15000 | out = 00010000
Time = 25000 | out = 00000001
Time = 35000 | out = 00100000
Time = 45000 | out = 00000001
Time = 55000 | out = 01000000
Time = 65000 | out = 00000001
Time = 75000 | out = 10000000
Time = 85000 | out = 00000001
Time = 95000 | out = 00010000
Time = 105000 | out = 00000001
$finish called at time : 110 ns : File "C:/Users/manojmsd/100_days_of_RTL_2/100_days_of_RTL_2.
```