```
`timescale 1ns / 1ps
module led(
    clk,
    reset,
    led
);
input clk;
input reset;
output [7:0] led;

reg [7:0] led;
reg [31:0] counter;
reg [1:0] state;


always @(posedge clk or negedge reset)
    begin
        if(!reset)
            begin
                counter <= 32'd0;
            end
        else if ( counter == 32'd10000000) //change T
            begin
                counter <= 32'd0;
            end
        else
            begin
                counter <= counter + 32'd1;
            end
        end
```

```
always @(posedge clk or negedge reset)
    begin
        if(!reset)
            begin
                state <= 1'd0;
            end
        else if( counter == 32'd0 )
            begin
            state <= 1'd1;
            end
        else
            begin
            state <= 1'd0;
            end

    end


always @(posedge clk or negedge reset)
    begin
        if(!reset)
            begin
                led <= 8'b1111_1111;
            end
        else
            begin
                if(state == 1'd1)
                begin
                    if(led == 8'b1111_1111)
                    begin
                        led <= 8'b0000_0001;
                    end
                    else if(led == 8'b1000_0000)
                    begin
                        led <= 8'b0000_0001;
                    end
                    else
                    begin
                        led <= led << 1'b1;
                    end
                end
            end
    end

endmodule
```

```
set_property    -dict {PACKAGE_PIN P17 IOSTANDARD LVCMOS33} [get_ports clk]
set_property    -dict {PACKAGE_PIN P15 IOSTANDARD LVCMOS33} [get_ports reset]

set_property    -dict {PACKAGE_PIN F6 IOSTANDARD LVCMOS33} [get_ports {led[0]}]
set_property    -dict {PACKAGE_PIN G4 IOSTANDARD LVCMOS33} [get_ports {led[1]}]
set_property    -dict {PACKAGE_PIN G3 IOSTANDARD LVCMOS33} [get_ports {led[2]}]
set_property    -dict {PACKAGE_PIN J4 IOSTANDARD LVCMOS33} [get_ports {led[3]}]
set_property    -dict {PACKAGE_PIN H4 IOSTANDARD LVCMOS33} [get_ports {led[4]}]
set_property    -dict {PACKAGE_PIN J3 IOSTANDARD LVCMOS33} [get_ports {led[5]}]
set_property    -dict {PACKAGE_PIN J2 IOSTANDARD LVCMOS33} [get_ports {led[6]}]
set_property    -dict {PACKAGE_PIN K2 IOSTANDARD LVCMOS33} [get_ports {led[7]}]
```

```
`timescale 1ns/1ns
module sim_dev1();
parameter T=1000;
reg clk;
reg reset;
wire [7:0] led8;

initial
begin
clk=1'b0;
reset=1'b0;
#(T+1) reset=1'b1;
end

always #(T/2) clk=~clk;

led u1(.clk(clk), .reset(reset), .led(led8));

endmodule
```

```
`timescale 1ns / 1ps
module JK(clk,Reset,Set,J,K,Q);
input clk;
input Reset;
input Set;
input J;
input K;

output Q;

reg q;

always@(posedge clk)
    if(!Reset)
        begin
            q <= 1'b0;
        end
    else if(!Set)
        begin
            q <= 1'b1;
        end
    else
        begin
            case({J,K})
                2'b00 : q <= q;
                2'b01 : q <= 0;
                2'b10 : q <= 1;
                default : q <= ~q;
            endcase
         end
     assign Q=q;
endmodule
```

```
`timescale 1ns / 1ps
module JK_sim();
reg Reset, Set, clk;
wire Q;
reg J, K;

JK u(clk, Reset, Set, J, K, Q);
initial
    begin
    Reset = 0;
    Set = 0;
    clk = 1;
    J = 0;
    K = 0;
    #80 Reset = 1;
    #40 Set = 1;
    end
always #20{J, K} = {J, K} + 1;
always #10 clk = ~clk;
endmodule
```

```
`timescale 1ns / 1ps
module counter_12(reset,set,clk,Y);
input reset;
input set;
input clk;
output [3:0]Y;

wire j1,j2,j3,j4,k1,k2,k3,k4;

assign j1 = 1;
assign k1 = 1;
assign j2 = Y[0];
assign k2 = Y[0];
assign j3 = (~Y[3])&Y[1]&Y[0];
assign k3 = Y[0]&Y[1];
assign j4 = Y[0]&Y[1]&Y[2];
assign k4 = Y[0]&Y[1];
JK_0 u1 (clk,reset,set,j1,k1,Y[0]);
JK_0 u2 (clk,reset,set,j2,k2,Y[1]);
JK_0 u3 (clk,reset,set,j3,k3,Y[2]);
JK_0 u4 (clk,reset,set,j4,k4,Y[3]);

endmodule
```

```
`timescale 1ns / 1ps
module sim_dev1();
reg reset,set,clk;
wire [3:0]Y;

initial
    begin
        reset = 0;
        set = 0;
        clk = 1;
        #40 set = 1;
        #40 reset = 1;
    end
always#20 clk = ~clk;
counter_12 u(reset,set,clk,Y);
endmodule
```