Date:10/09/2024

DRN: COURSE.FDI0003

Author: Sadeque Reza Khan

LED Blink with Clock Divider Verilog Code:

```
module blinking_LED( clk, divided_clk );
   input clk;
   output divided_clk;
   wire clk;
   reg divided_clk = 0;

   localparam div_value = 24999999;
   //localparam div_value = 49999999;

   //<statements>
   // division_value = 50MHz/(2*desired_value) - 1

   integer counter_value = 0;

   // counter
   always@ (posedge clk)
   begin
      if (counter_value == div_value)
         counter_value <= 0;
      else
         counter_value <= counter_value+1;
      end

    // clock divider
    always@ (posedge clk)
```

```
    begin
      if(counter_value == div_value)
        divided_clk <= ~divided_clk;    // Suppose to occur after 0.5sec
      else
        divided_clk <= divided_clk;     // If the counter is not at its limit, do nothing
      end
    endmodule
```

Test Bench:

```
`timescale 1ns/100ps

module blinking_LED_tb;
reg clk ;                 // input
wire divided_clk;         // output

initial
begin
  clk = 0;
end

    ////////////////////////////////////////////////////////////////
    // Clock Driver
    ////////////////////////////////////////////////////////////////

always
  #10 clk = ~clk;

    ////////////////////////////////////////////////////////////////

    //Instantiate Unit Under Test:  blinking_LED
```

```
//////////////////////////////////////////////////////////////////////

 blinking_LED blinking_LED_0 (

// Inputs
.clk(clk),

// Outputs
.divided_clk(divided_clk)

// Inouts

 );

endmodule
```