```
1   `timescale 1ns / 1ps
2   //////////////////////////////////////////////////////////////////////////////////
3   //                                                                              //
4   //  Class: CECS 360                                                             //
5   //  Project name: Project1_CECS360                                              //
6   //  File name: led_clk.v                                                  //
7   //                                                                              //
8   //  Created by Umar Khan 09/19/2017       .                                     //
9   //  Abstract:                                                                   //
10  //          Frequency of clk_out is determined by integer count,                //
11  //          count = (Incoming Freq/(Outgoing Freq))/2.                          //
12  //          clk_out will alternate once every time clk_in                       //
13  //          alternates 'count' number of times. To achieve a refresh            //
14  //          rate of 60Hz with 8 pixels, the output frequency must be            //
15  //          480hz.[Outgoing Freq = (Refresh Rate * Number of Pixels)]           //
16  //          To produce an output clock of 480hz, we divide the clock            //
17  //          every 104167 cycles. We assume that clk_in from the Nexys 4         //
18  //          FPGA is 100mhz.                                                     //
19  //                                                                              //
20  //////////////////////////////////////////////////////////////////////////////////
21  ////////////////////////////////////////////////////////////////////////////////////
22  module led_clk(clk, reset, led_clk);
23     input  clk, reset;
24     output led_clk;
25     reg    led_clk;
26     integer clk_ticks;
27
28     always @(posedge clk, posedge reset) begin
29        if(reset == 1'b1) begin
30           clk_ticks = 0;
31           led_clk   = 0;
32        end
33        //got a clock, so increment the counter and
34        //test to see if half a period has elapsed
35        else begin
36           clk_ticks = clk_ticks + 1;
37           //104166 is used to create 480Hz
38           if (clk_ticks >= 104166) begin
39              led_clk = ~led_clk;
40              clk_ticks = 0;
41           end
42        end
43     end
44
45
46  endmodule
```