

1. Introduction

This LED demo is based on MINI_STAR_4K development board, and use Gowin Software. The demo verifies the feasibility of the development board and EDA tool.

2. Environment

- Windows 10 64 bits
- MINI_STAR_4K development board
- Gowin Software: Win V1.9.7.03 Beta

3. Program Design

There are two LEDs (HIGH, ON) on the development board, and you can set four blinking modes.

- Mode one: led[0] and led[1] on/off cycle is one second, alternately blinking.
- Mode two: led[0] and led[1] on/off cycle is half a second, alternately blinking.
- Mode 3: led[0] and led[1] on/off cycle is half a second, on/off at the same time.
- Mode 4: one led on/off cycle is one second, and the other on/off cycle is half a second, one fast and one slow blinking.

LED Counter: There are two counters; one counter controls one led. When the clock frequency is 25 MHz (period, 40ns), led delay counter counts 1250000 cycles for a second, and 6250000 cycles counting is one second. The crystal oscillator on the development board is 27 MHz, and here 1250000 cycles delay is actually not enough for a second, but this test requirements are not so strict, it is available here.

```
1. //led[0] delay counter
2. always @(posedge clk or negedge rst_n) begin
3.     if (rst_n == 1'b0)
4.         cnt <= 24'b0 ;
5.
6.     else if (add_cnt) begin
7.         if (end_cnt)
```

```

8.      cnt <= 24'b0 ;
9.      else
10.     cnt <= cnt + 1'b1;
11.     end
12.     end
13.
14.     assign add_cnt = 1 ;
15.     assign end_cnt = add_cnt && cnt == total-1;
16.
17.     //led[1] delay counter
18.     always @(posedge clk or negedge rst_n) begin
19.         if (rst_n == 1'b0)
20.             cnt2 <= 24'b0 ;
21.
22.         else if (add_cnt2) begin
23.             if (end_cnt2)
24.                 cnt2 <= 24'b0 ;
25.             else
26.                 cnt2 <= cnt2 + 1'b1;
27.             end
28.         end
29.
30.         assign add_cnt2 = 1 ;
31.         assign end_cnt2 = add_cnt2 && cnt2 == total_2-1;

```

Mode Counter: Mode counter 1 adds one after led[0] on/off once and counts ten times. Mode counter 2 adds one when mode counter 1 is full and counts four times.

```

1. always @(posedge clk or negedge rst_n) begin
2.     if (rst_n == 1'b0)

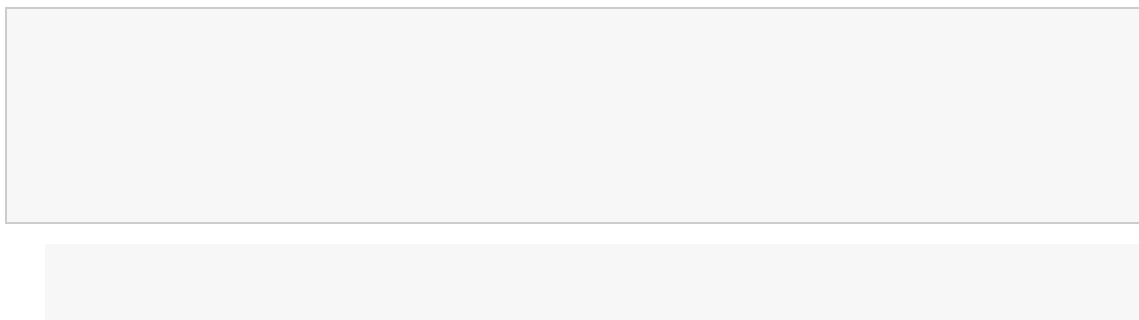
```

```

3.     cnt_model <= 4'b0 ;
4.     else if (add_cnt_model) begin
5.         if (end_cnt_model)
6.             cnt_model <= 4'b0 ;
7.         else
8.             cnt_model <= cnt_model + 1'b1;
9.         end
10.    end
11.
12.    assign add_cnt_model = end_cnt ;    // led[0] Blink once plus one
13.    assign end_cnt_model = add_cnt_model && cnt_model == 9;
14.
15.    always @(posedge clk or negedge rst_n) begin
16.        if (rst_n == 1'b0)
17.            wave_mode <= 2'b0 ;
18.        else if (add_wave_mode) begin
19.            if (end_wave_mode)
20.                wave_mode <= 2'b0 ;
21.            else
22.                wave_mode <= wave_mode + 1'b1;
23.            end
24.        end
25.
26.        assign add_wave_mode = end_cnt_model ;    // The previous level counter full
            plus one
27.        assign end_wave_mode = add_wave_mode && wave_mode == 3;

```

LED Control: Using combination logic, you can modify the maximum value of the two led counters according to the value of mode counter 2,Q and control the two leds on/off. The code is as follows:



```

1. // Set the state of led[0] in half and full cycles
2. always @(posedge clk or negedge rst_n) begin
3.     if(!rst_n) begin
4.         led[0] <= 1'b0;
5.     end
6.     else if(cnt == total/2-1) begin
7.         led[0] <= wave_1;
8.     end
9.     else if(cnt == total-1)begin
10.        led[0] <= wave_2 ;
11.    end
12. end
13.
14. // Set the state of led[1] in half and full cycles
15. always @(posedge clk or negedge rst_n) begin
16.     if(!rst_n) begin
17.         led[1] <= 1'b0;
18.     end
19.     else if(cnt2 == total_2 / 2 - 1) begin
20.         led[1] <= wave_1_2 ;
21.     end
22.     else if(cnt2 == total_2 - 1)begin
23.         led[1] <= wave_2_2 ;
24.     end
25. end
26.
27. // led blinking mode
28. always @(*) begin
29.     if(!rst_n)begin
30.         total = 0;
31.         total_2 = 0;
32.         wave_1 = 1'b0;
33.         wave_2 = 1'b0;
34.         wave_1_2 = 1'b0;
35.         wave_2_2 = 1'b0;
36.     end

```

```
37. // in this mode, led[0] and led[1] cycle, a second, alternately blinking
38. else if (wave_mode == 0) begin
39.     total = 6250000;
40.     total_2 = 6250000;
41.     wave_1 = 1'b0; // led[0] off in half cycle
42.     wave_2 = 1'b1; // led[0] on in full cycle
43.     wave_1_2 = 1'b1; // led[1] on in half cycle
44.     wave_2_2 = 1'b0; // led[1] off in full cycle
45. end
46. else if (wave_mode == 1)begin
47.     total = 12500000;
48.     total_2 = 6250000;
49.     wave_1 = 1'b0;
50.     wave_2 = 1'b1;
51.     wave_1_2 = 1'b1;
52.     wave_2_2 = 1'b0;
53. end
54. else if (wave_mode == 2) begin
55.     total = 6250000;
56.     total_2 = 6250000;
57.     wave_1 = 1'b1;
58.     wave_2 = 1'b0;
59.     wave_1_2 = 1'b1;
60.     wave_2_2 = 1'b0;
61. end
62. else if (wave_mode == 3) begin
63.     total = 12500000;
64.     total_2 = 12500000;
65.     wave_1 = 1'b1;
66.     wave_2 = 1'b0;
67.     wave_1_2 = 1'b1;
68.     wave_2_2 = 1'b0;
69. end
70. else begin
71.     total = 0;
72.     total_2 = 0;
```

```
73.     wave_1 = 1'b0;
74.     wave_2 = 1'b0;
75.     wave_1_2 = 1'b0;
76.     wave_2_2 = 1'b0;
77.     end
78.     end
```