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. else if (add\_cnt) begin
6. if (end\_cnt)
7. cnt <= 24'b0 ;
8. else
9. cnt <= cnt + 1'b1;
10. end
11. end
12. assign add\_cnt = 1 ;
13. assign end\_cnt = add\_cnt && cnt == total-1;
14. //led[1] delay counter
15. always @(posedge clk or negedge rst\_n) begin
16. if (rst\_n == 1'b0)
17. cnt2 <= 24'b0  ;
18. else if (add\_cnt2) begin
19. if (end\_cnt2)
20. cnt2 <= 24'b0 ;
21. else
22. cnt2 <= cnt2 + 1'b1;
23. end
24. end
25. assign add\_cnt2 = 1 ;
26. 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. assign add\_cnt\_model = end\_cnt ;        //  led[0] Blink once plus one
12. assign end\_cnt\_model = add\_cnt\_model && cnt\_model == 9;
13. always @(posedge clk or negedge rst\_n) begin
14. if (rst\_n == 1'b0)
15. wave\_mode <= 2'b0  ;
16. else if (add\_wave\_mode) begin
17. if (end\_wave\_mode)
18. wave\_mode <= 2'b0 ;
19. else
20. wave\_mode <= wave\_mode + 1'b1;
21. end
22. end
23. assign add\_wave\_mode = end\_cnt\_model ;   // The previous level counter full plus one
24. 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, 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. // Set the state of led[1] in half and full cycles
14. always @(posedge clk or negedge rst\_n) begin
15. if(!rst\_n) begin
16. led[1] <= 1'b0;
17. end
18. else if(cnt2 == total\_2 / 2 - 1) begin
19. led[1] <= wave\_1\_2 ;
20. end
21. else if(cnt2 == total\_2 - 1)begin
22. led[1] <= wave\_2\_2 ;
23. end
24. end
26. // led blinking mode
27. always @(\*) begin
28. if(!rst\_n)begin
29. total = 0;
30. total\_2 = 0;
31. wave\_1 = 1'b0;
32. wave\_2 = 1'b0;
33. wave\_1\_2 = 1'b0;
34. wave\_2\_2 = 1'b0;
35. end
36. // in this mode, led[0] and led[1] cycle, a second, alternately blinking
37. else if (wave\_mode == 0) begin
38. total = 6250000;
39. total\_2 = 6250000;
40. wave\_1 = 1'b0;              // led[0] off in half cycle
41. wave\_2 = 1'b1;              // led[0] on in full cycle
42. wave\_1\_2 = 1'b1;            // led[1] on in half cycle
43. wave\_2\_2 = 1'b0;            // led[1] off in full cycle
44. end
45. else if (wave\_mode == 1)begin
46. total = 12500000;
47. total\_2 = 6250000;
48. wave\_1 = 1'b0;
49. wave\_2 = 1'b1;
50. wave\_1\_2 = 1'b1;
51. wave\_2\_2 = 1'b0;
52. end
53. else if (wave\_mode == 2) begin
54. total = 6250000;
55. total\_2 = 6250000;
56. wave\_1 = 1'b1;
57. wave\_2 = 1'b0;
58. wave\_1\_2 = 1'b1;
59. wave\_2\_2 = 1'b0;
60. end
61. else if (wave\_mode == 3) begin
62. total = 12500000;
63. total\_2 = 12500000;
64. wave\_1 = 1'b1;
65. wave\_2 = 1'b0;
66. wave\_1\_2 = 1'b1;
67. wave\_2\_2 = 1'b0;
68. end
69. else  begin
70. total = 0;
71. total\_2 = 0;
72. wave\_1 = 1'b0;
73. wave\_2 = 1'b0;
74. wave\_1\_2 = 1'b0;
75. wave\_2\_2 = 1'b0;
76. end
77. end