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.

```
    //led[0] delay counter
    always @(posedge clk or negedge rst_n) begin
        if (rst_n == 1'b0)
            cnt <= 24'b0  ;

        else if (add_cnt) begin
            if (end_cnt)
```

```
                cnt <= 24'b0 ;
            else
                cnt <= cnt + 1'b1;
        end
    end

    assign add_cnt = 1 ;
    assign end_cnt = add_cnt && cnt == total-1;

    //led[1] delay counter
    always @(posedge clk or negedge rst_n) begin
        if (rst_n == 1'b0)
            cnt2 <= 24'b0  ;

        else if (add_cnt2) begin
            if (end_cnt2)
                cnt2 <= 24'b0 ;
            else
                cnt2 <= cnt2 + 1'b1;
        end
    end

    assign add_cnt2 = 1 ;
    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.

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

```
        cnt_model <= 4'b0  ;
    else if (add_cnt_model) begin
        if (end_cnt_model)
            cnt_model <= 4'b0 ;
        else
            cnt_model <= cnt_model + 1'b1;
        end
  end

    assign add_cnt_model = end_cnt ;        //  led[0] Blink once plus one
    assign end_cnt_model = add_cnt_model && cnt_model == 9;

    always @(posedge clk or negedge rst_n) begin
     if (rst_n == 1'b0)
         wave_mode <= 2'b0  ;
     else if (add_wave_mode) begin
         if (end_wave_mode)
             wave_mode <= 2'b0 ;
         else
             wave_mode <= wave_mode + 1'b1;
         end
  end

    assign add_wave_mode = end_cnt_model ;   // The previous level counter full plus one
    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:

```
```

```
    // Set the state of led[0] in half and full cycles
    always @(posedge clk or negedge rst_n) begin
        if(!rst_n) begin
            led[0] <= 1'b0;
        end
        else if(cnt == total/2-1) begin
            led[0] <= wave_1;
        end
        else if(cnt == total-1)begin
             led[0] <= wave_2 ;
         end
    end

    // Set the state of led[1] in half and full cycles
    always @(posedge clk or negedge rst_n) begin
        if(!rst_n) begin
            led[1] <= 1'b0;
        end
        else if(cnt2 == total_2 / 2 - 1) begin
            led[1] <= wave_1_2 ;
        end
        else if(cnt2 == total_2 - 1)begin
            led[1] <= wave_2_2 ;
        end
    end

    // led blinking mode
    always @(*) begin
        if(!rst_n)begin
            total = 0;
            total_2 = 0;
            wave_1 = 1'b0;
            wave_2 = 1'b0;
            wave_1_2 = 1'b0;
            wave_2_2 = 1'b0;
        end
```

```
        // in this mode, led[0] and led[1] cycle, a second, alternately blinking
        else if (wave_mode == 0) begin
            total = 6250000;
            total_2 = 6250000;
            wave_1 = 1'b0;              // led[0] off in half cycle
            wave_2 = 1'b1;              // led[0] on in full cycle
            wave_1_2 = 1'b1;            // led[1] on in half cycle
            wave_2_2 = 1'b0;            // led[1] off in full cycle
        end
        else if (wave_mode == 1)begin
            total = 12500000;
            total_2 = 6250000;
            wave_1 = 1'b0;
            wave_2 = 1'b1;
            wave_1_2 = 1'b1;
            wave_2_2 = 1'b0;
        end
        else if (wave_mode == 2) begin
            total = 6250000;
            total_2 = 6250000;
            wave_1 = 1'b1;
            wave_2 = 1'b0;
            wave_1_2 = 1'b1;
            wave_2_2 = 1'b0;
        end
        else if (wave_mode == 3) begin
            total = 12500000;
            total_2 = 12500000;
            wave_1 = 1'b1;
            wave_2 = 1'b0;
            wave_1_2 = 1'b1;
            wave_2_2 = 1'b0;
        end
        else  begin
            total = 0;
            total_2 = 0;
```

```
            wave_1 = 1'b0;
            wave_2 = 1'b0;
            wave_1_2 = 1'b0;
            wave_2_2 = 1'b0;
        end
    end
```