LED blinker: solution

Introduction to digital Low-Level Radio Frequency Controls in Accelerators

Lab 3

Qiang Du

US Particle Accelerator School
Janurary 23 – 27, 2023

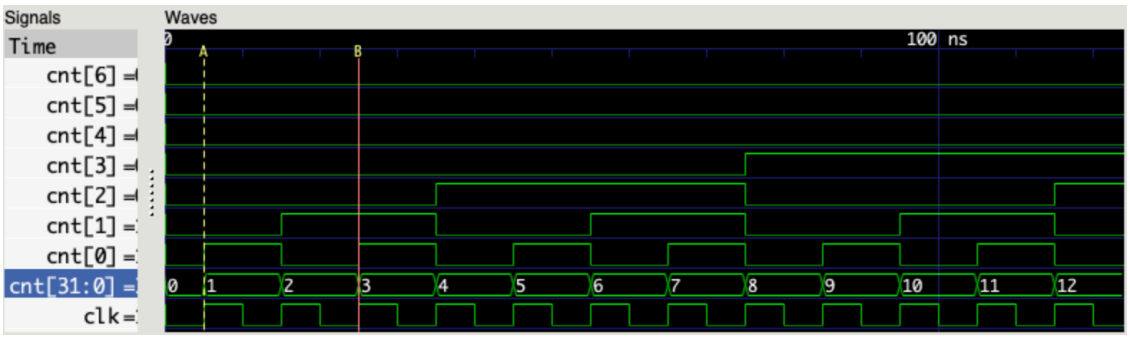
Contents

1 Solutions

1.1 RTL logic

> If the frequency of `clk` is 100 MHz, what's the expected LED blinking rate for each bit?

The following gtkwave simulation shows that the least significant bit cnt[0] is blinking at frequency of 50 MHz, or $100^6/2^1$ Hz.



Given the `MSB` is 27, the 4 LEDs are blinking at the rate of:

LED3: bit 27, $100^6/2^{28} = 0.37$ Hz

LED2: bit 26, $100^6/2^{27} = 0.76$ Hz

LED1: bit 25, $100^6/2^{26} = 1.49$ Hz

LED0: bit 24, $100^6/2^{25} = 2.98$ Hz

1.2 Your turn: make the LED dimmable

Any implementation of PWM is acceptable. The idea is to stop the counter at a given setpoint, and then reset and repeat.

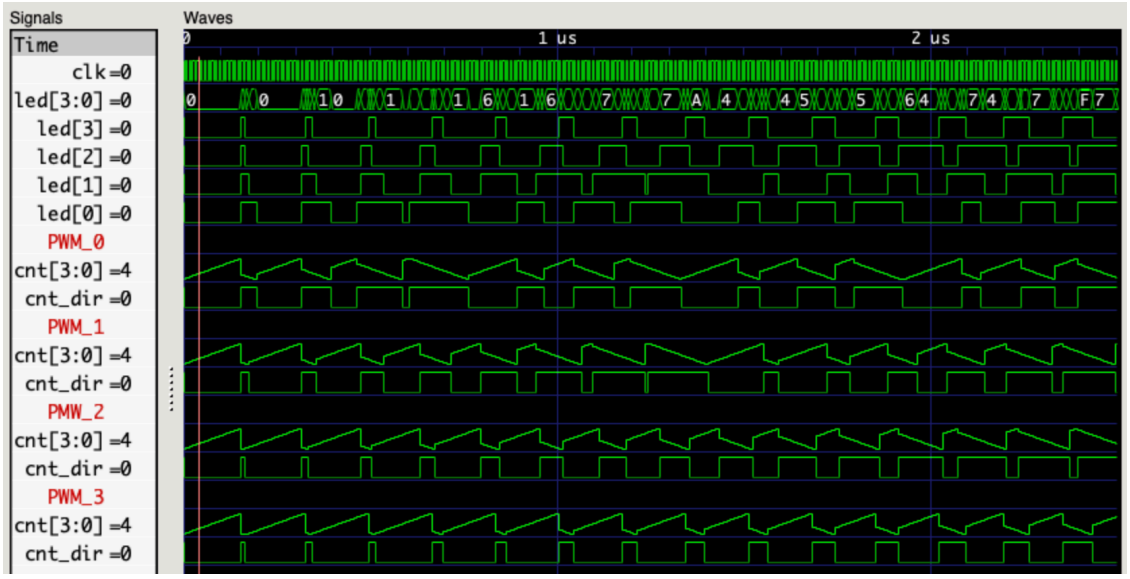
The following example is from one of the top search results of "FPGA PWM" fpga4fun.com, and just wire the 4 LEDs to the PWM module, and set the `PWM_in` port to a ramping setpoint using different bits of the same counter, so the 4 LEDs will change the dimming brightness over time.

New `led_test.v`:

```
module led_test #(
    parameter MSB = 27
) (
    input clk,
    input reset,
    output [3:0] led
);
```

```
reg [31:0] cnt=0;
always @(posedge clk) begin
    cnt <= reset ? 32'h0 : cnt + 1'b1;
end

PWM PWM_3 (.clk(clk), .PWM_in(cnt[MSB-0:MSB-3]), .PWM_out(led[3]));
PWM PWM_2 (.clk(clk), .PWM_in(cnt[MSB-1:MSB-4]), .PWM_out(led[2]));
PWM PWM_1 (.clk(clk), .PWM_in(cnt[MSB-2:MSB-5]), .PWM_out(led[1]));
PWM PWM_0 (.clk(clk), .PWM_in(cnt[MSB-3:MSB-6]), .PWM_out(led[0]));

endmodule

// https://www.fpga4fun.com/PWM_DAC_1.html
module PWM(
    input clk,
    input [3:0] PWM_in,
    output PWM_out
);

reg [3:0] cnt=0;
reg cnt_dir=0;  // 0 to count up, 1 to count down
wire [3:0] cnt_next = cnt_dir ? cnt-1'b1 : cnt+1'b1;
wire cnt_end = cnt_dir ? cnt==4'b0000 : cnt==4'b1111;

always @(posedge clk) cnt <= cnt_end ? PWM_in : cnt_next;
always @(posedge clk) cnt_dir <= cnt_dir ^ cnt_end;
assign PWM_out = cnt_dir;
endmodule
```

The following simulation shows details when setting the `MSB` to 8.



The synthesized file now results in a changing brightness with a period. For LED3, this period is defined by the frequency of `cnt[27:24]`.