FPGA Lab-07

Red and green light control circuit

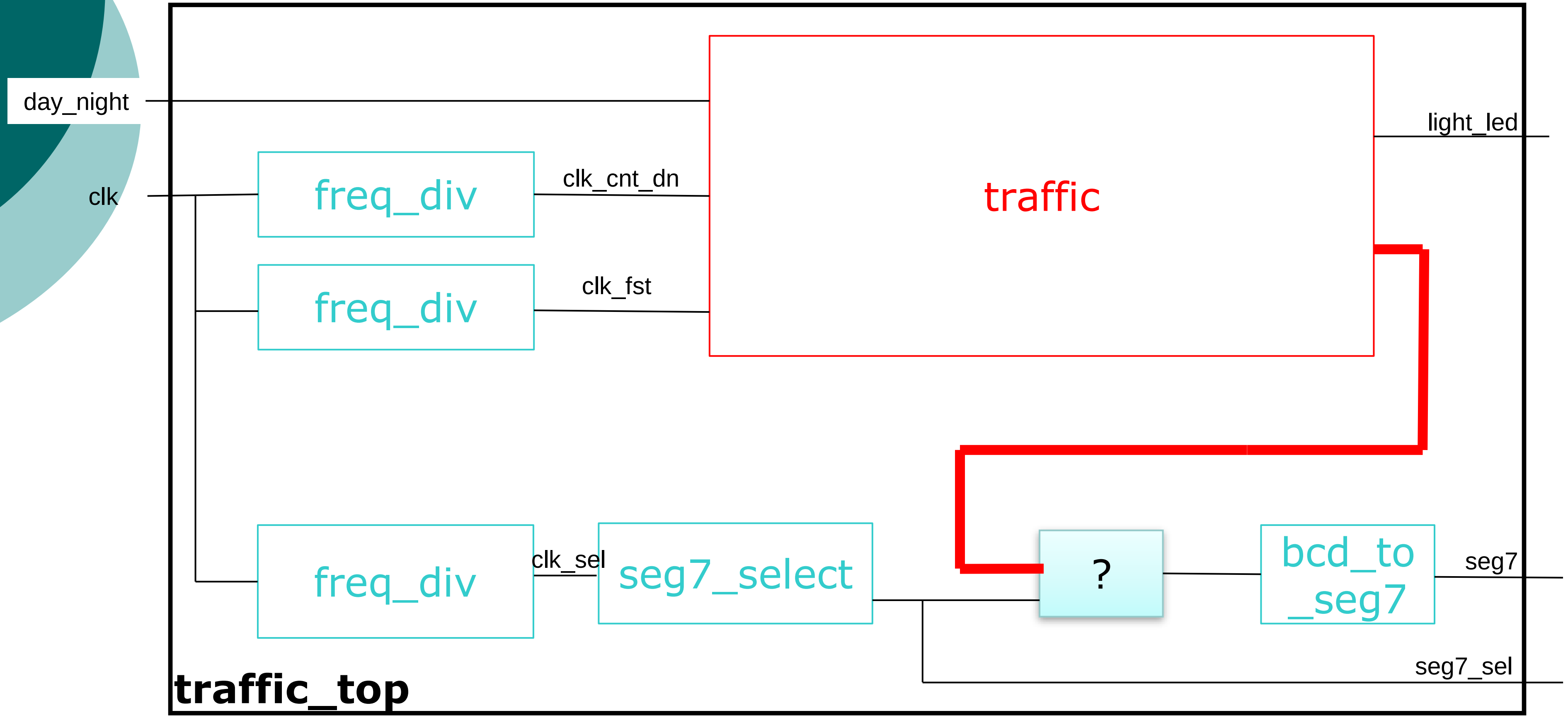
實驗內容

- Design a daytime and nighttime traffic light control circuit.
- Daytime (Mode 0~5 repeat cycle):
- Mode = 0, Green 1 is on, Red 2 is on (20 sec.)
- Mode = 1, Green 1 flashing, Red 2 on (5 seconds)
- Mode = 2, Yellow 1 is on, Red 2 is on (4 seconds)
- Mode = 3, Red 1 on, Green 2 on (after 20 seconds)
- Mode = 4, Red 1 is on, Green 2 is flashing (after 5 seconds)
- Night：
- Yellow light 1 blinking, yellow light 2 blinking
- 7-segment display counts down the red light at the same time as the green light sign.
- 7-segment display 0 at night

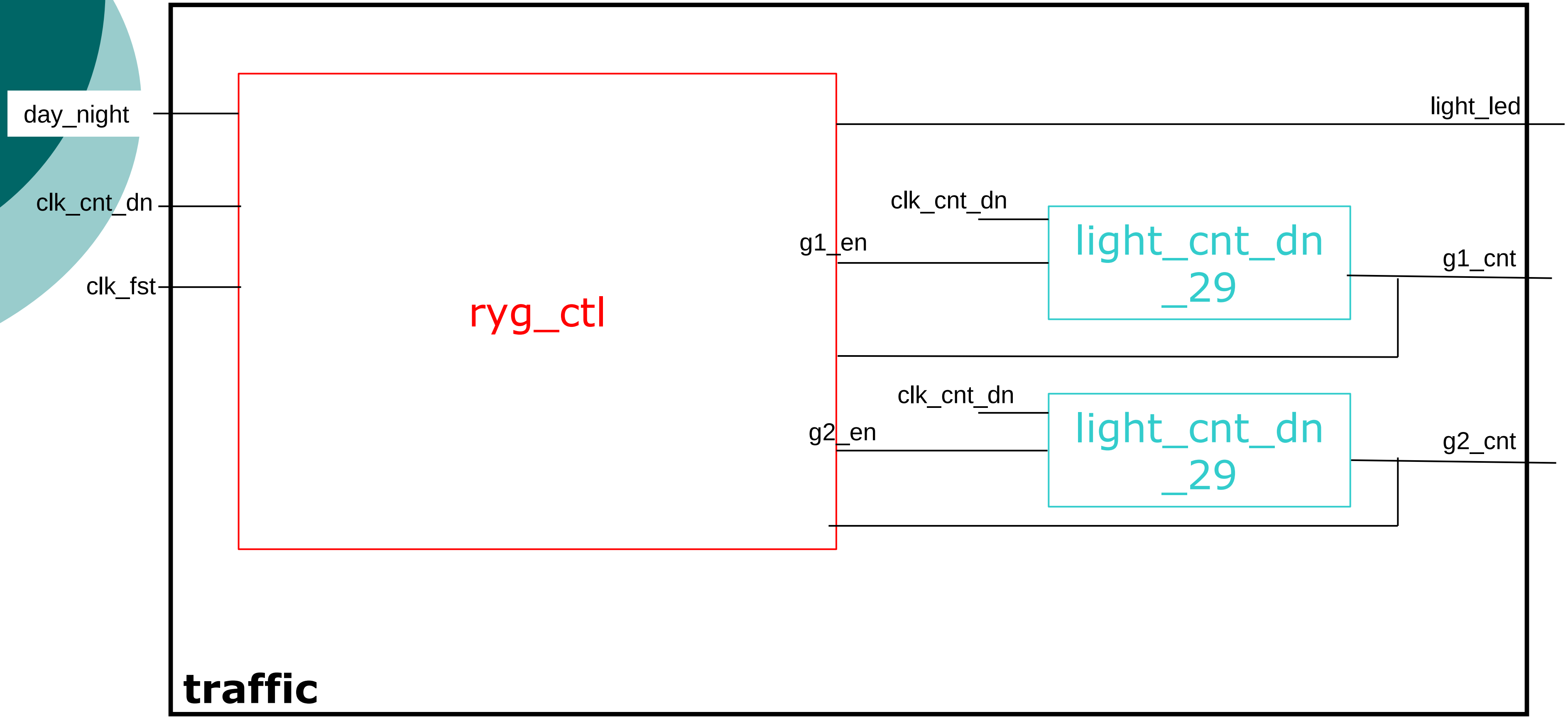
Verilog Codes

- light_cnt_dn_29.v (countdown 29~0)
- ryg_ctl.v (Control Mode 0~5 repeatedly)
- traffic.v (Combine light_cnt_dn_29. and ryg_ctl.v)
- traffic_top.v (Combine all the module)
- freq_div.v (straight use from previous code)
- seg7_select.v (straight use from previous code)
- bcd_to_seg7.v (straight use from previous code)

Program Architecture (1/3) **traffic_top.v**



Program Architecture (2/3) **traffic.v**



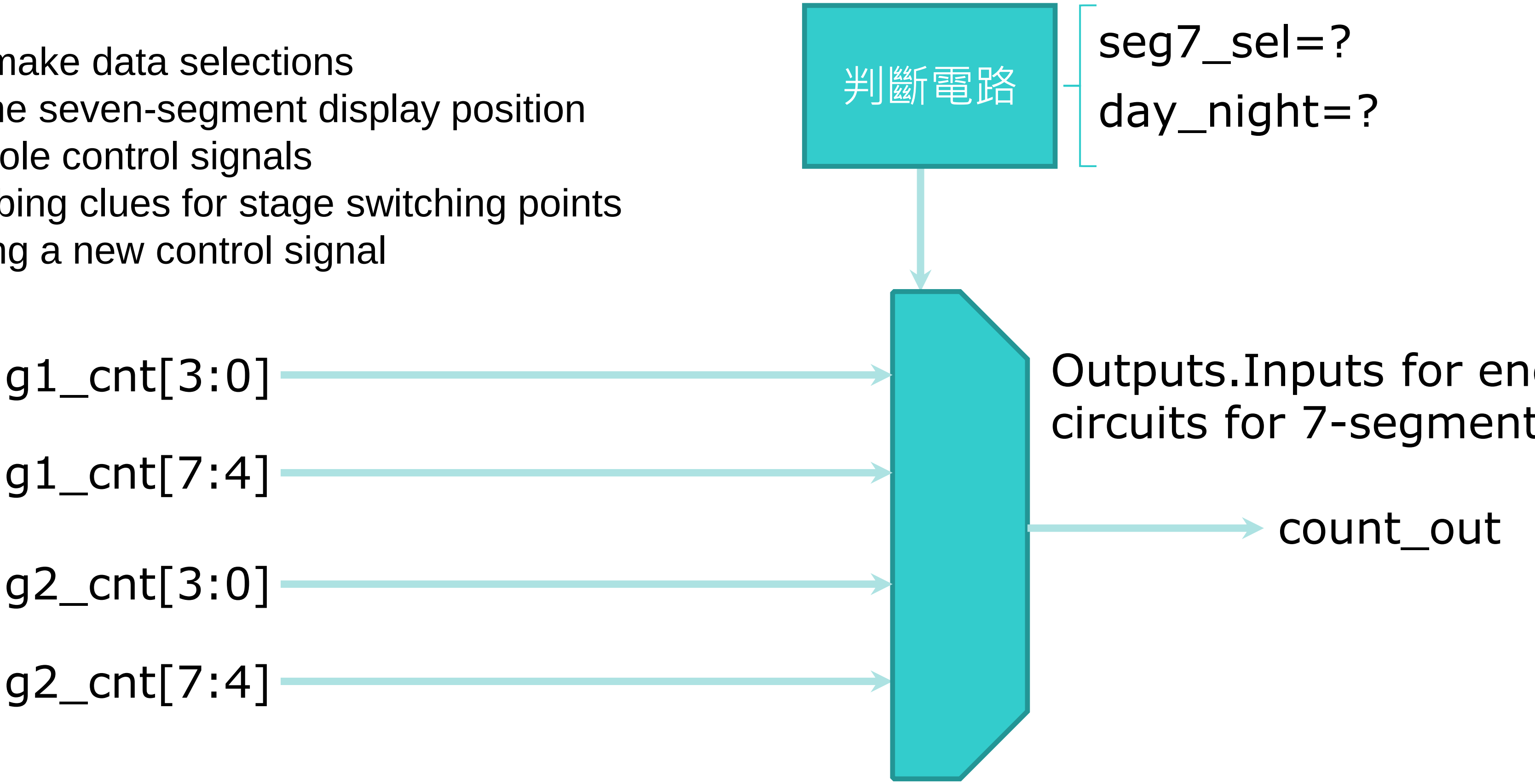
?

Tip: Generate control signals within your stage.

Program Architecture (3/3)

How to make data selections

- : of the seven-segment display position
- Console control signals
- Grabbing clues for stage switching points
- Adding a new control signal



light_cnt_dn_29.v

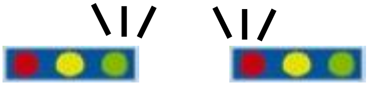
```
module light_cnt_dn_29 (clk, rst, enable, cnt);
input            clk, rst, enable;
output[7:0]      cnt;
reg[7:0]         cnt;//MSB[7:4] for tens digits,LSB [3:0] for ones digits
always@(posedge clk or posedge rst) begin
if(rst)
        cnt= 8'b0; // initial state
else if(enable)  // 0 -> 29 -> 24 -> ... -> 1 -> 0 -> 29
        if(cnt== 8'b0)
                      your code;  // 29
        else if(cnt[3:0] == 4'd0) begin  // 20 -> 19, 10 -> 09
                      your code;
                      your code;
        end
        else
                      your code;  // 19 -> 18, 18 -> 17, 17 -> 16, …
else    cnt=8’b0;
end
endmodule
```

ryg_ctl.v(1/5)

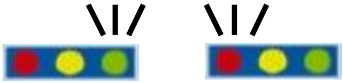
```
module ryg_ctl (clk_fst, clk_cnt_dn, rst,
day_night, g1_cnt, g2_cnt, g1_en, g2_en,
light_led);
input         clk_fst, clk_cnt_dn, rst, day_night;
input[7:0]    g1_cnt, g2_cnt;
output        g1_en, g2_en;
output[5:0]   light_led;
reg           g1_en, g2_en;
reg[5:0]      light_led;
reg[2:0]      mode;
```

ryg_ctl.v(2/5)

```
always@(posedge clk_fst or posedge rst) begin
if (rst)begin
        light_led <= 6'b001_100; // g1 : r2
```



```
        mode <= 3'b0;
        g1_en <= 1'b0;
        g2_en <= 1’b0;
        end
else if(day_night == 1'b1) // day time
case(mode)
3'd0: begin
        light_led <= 6'b001_100; // g1 : r2
```



```
        g1_en <= 1'b1;     // g1 count down
        if(g1_cnt == 8'b0000_1001)          // after 20 seconds
        mode <= mode + 3'b1;
        end
```

ryg_ctl.v(3/5)

```
3'd1: begin       // g1 flashes : r2
          if (g1_cnt == 8'b0000_0100) //after 5 seconds
          mode <= mode + 3'b1;
          else
          light_led[3] <= clk_cnt_dn; // g1 flashes
          end
3'd2: begin
          light_led = 6'b010_100;        // y1 : r2
          if (g1_cnt == 8'b0000_0000) begin   // after 4 seconds
          g1_en <= 1'b0;
          mode <= mode + 3'b1;
          end
          end
```

ryg_ctl.v(4/5)

```
3'd3: begin
        light_led <= 6'b100_001;    // r1 : g2
        g2_en <= 1'b1;
        if(g2_cnt == 8'b0000_1001)            // after 20 seconds
        mode <= mode + 3'b1;
        end
```

```
3'd4: begin      // r1 : g2 flashes
        if(g2_cnt == 8'b0000_0100)            // after 5 seconds
        mode <= mode + 3'b1;
        else
        light_led[0] <= clk_cnt_dn; // g2 flashes
        end
```

ryg_ctl.v(5/5)

```
3'd5:   begin
        light_led <= 6'b100_010;            // r1 : y2
```

```
        if (g2_cnt == 8'b0000_0000) begin // after 4 seconds
        g2_en <= 1'b0;
        mode <= 3'b0;
        end
        end
default: begin       // back to mode0
```

```
        light_led <= 6'b001_100;            // g1 : r2
        g1_en <= 1'b1; // g1 count down
        if(g1_cnt == 8'b0000_1001)          // after 20 seconds
        mode <= mode + 3'b1;
        end
endcase
else if(day_night == 1'b0)begin  // night time
row_en <= 2'b11;
light_led <= {{1'b0, clk_cnt_dn, 1'b0}, {1'b0, clk_cnt_dn, 1'b0}};
// y1 flashes : y2 flashes
g1_en <= 1'b0;
g2_en <= 1'b0;
end
end
endmodule
```

traffic.v

```
module traffic (clk_fst, clk_cnt_dn, rst, day_night, g1_cnt,
g2_cnt, light_led);
input          clk_fst, clk_cnt_dn, rst, day_night;
output[5:0]    light_led;
output[7:0]    g1_cnt;
output[7:0]    g2_cnt;
wire           g1_en, g2_en;
wire[7:0]      g1_cnt;
wire[7:0]      g2_cnt;
ryg_ctl M0(your code);
light_cnt_dn_29 M1(your code); // for light 1
light_cnt_dn_29 M2(your code); // for light 2
endmodule
```

traffic_top.v(1/2)

```
module        traffic_top(clk, rst, day_night, light_led,
led_com, seg7_out, seg7_sel);
Input         clk;
Input         rst;
input         day_night;
output[5:0]   light_led;//pin E2 ,D3 ,C2 ,N1 ,AA1 ,AA1\2
output        led_com;//pin N20
output[2:0]   seg7_sel;//pin AB10 ,AB11, AA12
output[6:0]   seg7_out;
wire          led_com;
wire          clk_cnt_dn;
wire[7:0]     g1_cnt;
wire[7:0]     g2_cnt;
wire[3:0]     count_out;
```

traffic_top.v(2/2)

```
○ assign       led_com= 1'b1;
○ assign       count_out = your code;
○ freq_div#(23) M0(clk, rst, clk_cnt_dn);
○ freq_div#(21) M1(clk, rst, clk_fst);
○ freq_div#(15) M2(clk, rst, clk_sel);
○ traffic M3(your code);
○ bcd_to_seg7 M4(your code);
○ seg7_select#(?) M5(your code);
○ endmodule
```

Extra Question

- Creating pedestrian signals in the LED matrix:
- 20 seconds before the green light countdown: Green Man walking
- 5 seconds left of green countdown: Green man running
- Yellow light: Little Red Man
- Red light: Little Red Man