

# 數位電路實驗 HW9: Traffic Light

第十四組：A1055547 王公志、A1073136 黃柏盛

## Abstract

在 FPGA 上，以 VHDL 程式碼模擬一個每 20 cycle 兩個方向紅綠切換的交通號誌。綠燈在剩下 2 cycle 時變為黃燈，倒數到 0 時轉為紅燈，另一方向則轉為綠燈，方向和秒數則分別用兩個七段顯示器來顯示

## Content

### *1. Principles:*

由於 FPGA 上的七段顯示器一次只能顯示一個，為了讓顯示器實際上看起來是真的時鐘，因此只要讓在每個 cycle 切換顯示不同的顯示器，就能達到目的。

### *2. Method:*

為了讓顯示器不會因為切得太快而漏掉某些數字，因而採用除頻器的作法。且考慮到有八個七段顯示器，因此每隔 150 個 cycle 將秒數個位數(sec\_1)減一，並依照規則從十位數(sec\_10)倒數。當倒數至 0 分 0 秒時，將號誌方向和紅綠燈切換，如此一來號誌的秒數和方向由於重複顯示多次，因此在高頻率下都能顯示明確。

在每個 cycle 將兩個秒數和方向轉換為 8-bit 長的七段顯示器相對應 output，並在不同 cycle 輪流顯示不同的七段顯示器。而 LED 模擬顯示紅綠燈的規則只要照個投影片的指示設定就好。

### 3. VHDL 程式碼：

```
1  module traffic_light(seg, SEL, mul_out, sec_10, sec_1, dir, led, clk, rst);
2
3      input clk, rst;
4      // sec_10: 十位數; sec_1: 個位數
5      output SEL, mul_out, sec_10, sec_1, dir, led;
6      output [7:0] seg;
7      reg [2:0] SEL; // 七段顯示器的COM選擇腳位
8      reg [3:0] mul_out;
9      reg [1:0] sec_10;
10     reg [3:0] sec_1, dir;
11     reg [11:0] led;
12
13     reg [7:0] cnt; // counter for frequency divider
14     reg count; // 當除頻器觸發後, counter++
15
16     reg switch; // 用來在兩個紅綠燈之間切換
17
18     always @ (posedge clk or negedge rst) begin // 除頻器
19         if (!rst) begin
20             cnt <= 0;
21             SEL <= 0;
22         end
23         else begin
24             if (cnt == 150) begin // 每過150個cycle, 秒數(count)加一
25                 cnt <= 0;
26                 count <= 1;
27             end
28             else begin
29                 cnt <= cnt + 1;
30                 count <= 0;
31             end
32
33             if (SEL == 7) // 每個cycle切換到不同七段顯示器
34                 SEL <= 0;
35             else
36                 SEL <= SEL + 1;
37
38             if (SEL == 1 || SEL == 5) // 顯示十位數
39                 mul_out <= sec_10;
40             else if (SEL == 0 || SEL == 4) // 顯示個位數
41                 mul_out <= sec_1;
42             else
43                 mul_out <= dir; // 顯示方向
44
45             if ((sec_10 == 4'b0000) && (sec_1 < 4'b0011)) // 少於0分2秒時, 顯示黃燈
46                 if (switch == 0) // 兩個紅綠燈互相切換
47                     led <= 12'b010111111100;
48                 else // 兩個紅綠燈互相切換
49                     led <= 12'b1001111111010;
50             else
51                 if (switch == 0) // 兩個紅綠燈互相切換
52                     led <= 12'b001111111100;
53                 else // 兩個紅綠燈互相切換
54                     led <= 12'b1001111111001;
55             end
56         end
57     end
```

```

58 always @ (posedge count) begin
59     if (sec_1 == 4'b0000) begin
60         sec_1 <= 9;
61         if (sec_10 == 4'b0000) begin
62             sec_10 <= 1;
63
64             // 當紅綠燈倒數完，互相交換
65             switch <= switch + 1;
66             if (dir == 10) // 10和11用來顯示方向
67                 dir <= 11;
68             else
69                 dir <= 10;
70         end
71     else
72         sec_10 <= 0;
73 end
74 else begin
75     sec_1 <= sec_1 - 1;
76 end
77 end

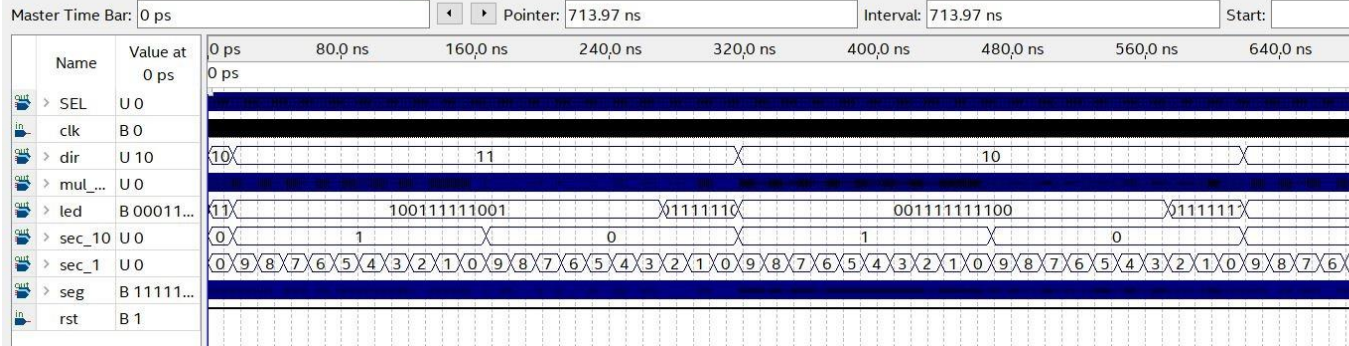
79 assign seg = (mul_out == 4'b0000) ? 8'b11111100:
80             (mul_out == 4'b0001) ? 8'b01100000:
81             (mul_out == 4'b0010) ? 8'b11011010:
82             (mul_out == 4'b0011) ? 8'b11110010:
83             (mul_out == 4'b0100) ? 8'b01100110:
84             (mul_out == 4'b0101) ? 8'b10110110:
85             (mul_out == 4'b0110) ? 8'b10111110:
86             (mul_out == 4'b0111) ? 8'b11100000:
87             (mul_out == 4'b1000) ? 8'b11111110:
88             (mul_out == 4'b1001) ? 8'b11110110:
89             (mul_out == 4'b1010) ? 8'b00010010:
90             (mul_out == 4'b1011) ? 8'b00101000:
91             8'b10011110;
92 endmodule
93

```

## 4. 模擬結果：

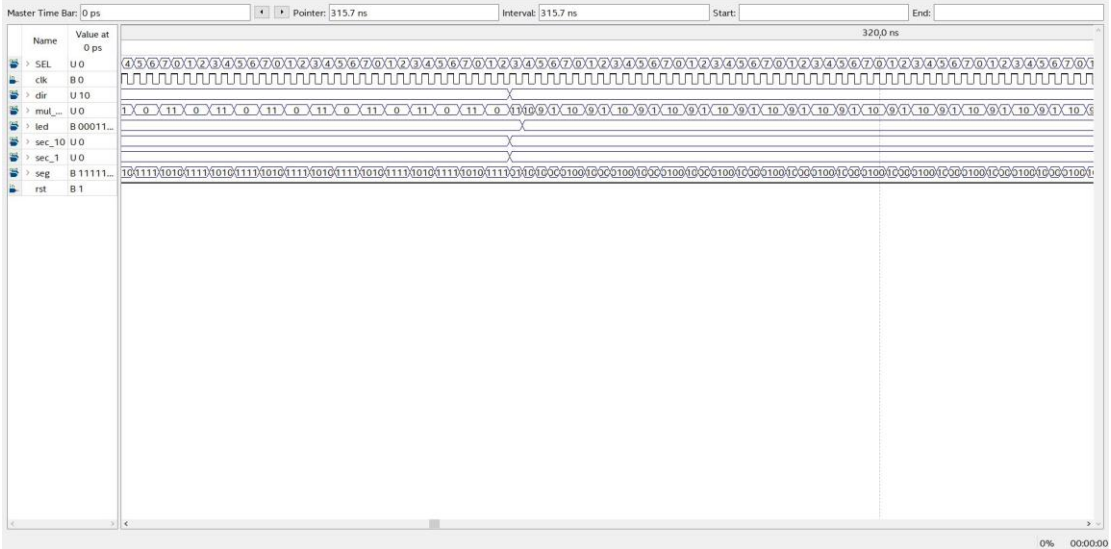
Simulation Waveform Editor - D:\intelFPGA\_lite\18.1\traffic\_light\traffic\_light - traffic\_light - [traffic\_light\_20210608111113.sim.vwf (Read-Only)]

File Edit View Simulation Help



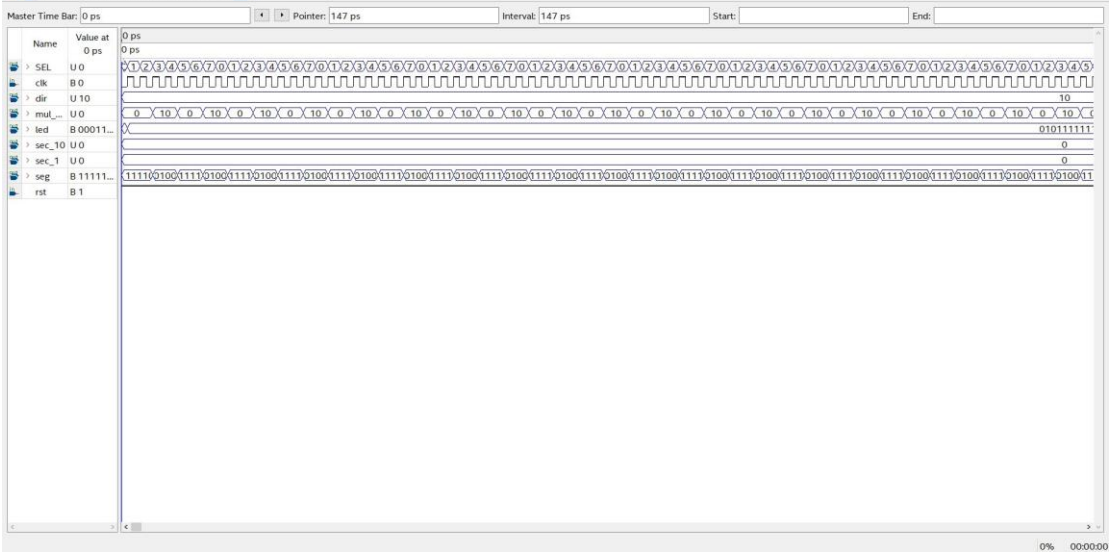
Simulation Waveform Editor - D:\intelFPGA\_lite\18.1\traffic\_light\traffic\_light - traffic\_light - [traffic\_light\_20210608111113.sim.vwf (Read-Only)]

File Edit View Simulation Help



Simulation Waveform Editor - D:\intelFPGA\_lite\18.1\traffic\_light\traffic\_light - traffic\_light - [traffic\_light\_20210608111113.sim.vwf (Read-Only)]

File Edit View Simulation Help



Simulation Waveform Editor - D:/intelFPGA\_lite/18.1/traffic\_light/traffic\_light - traffic\_light - [traffic\_light\_20210608111113.sim.vwf (Read-Only)]

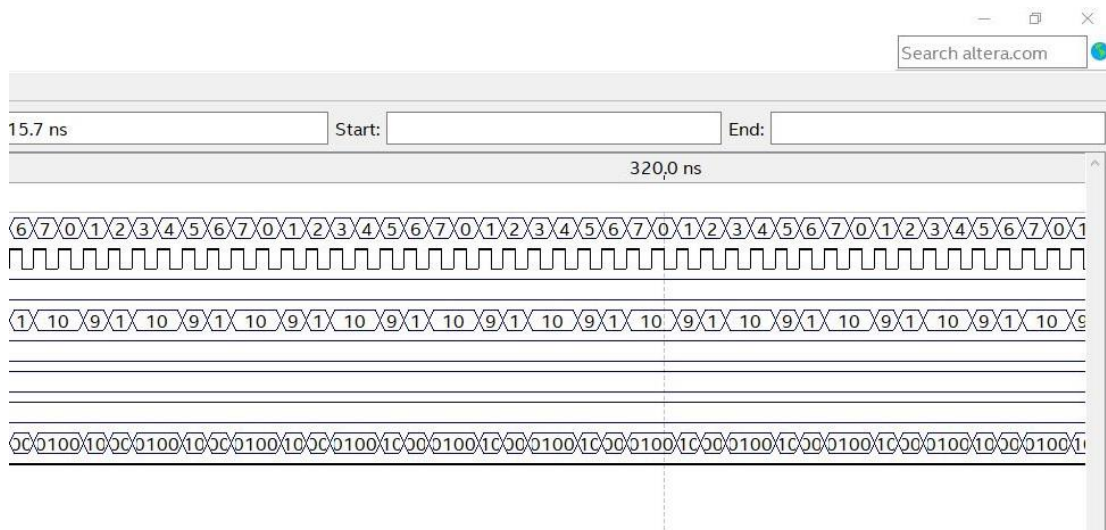
Timing diagram showing signals over time. The signals are: SEL, clk, dir, mul\_..., led, sec\_10, sec\_1, seg, and rst. The clock (clk) is a periodic square wave. The select signal (SEL) is high for the first 10 clock cycles and low for the next 10. The direction signal (dir) is high for the first 10 cycles and low for the next 10. The multiplexer select signal (mul\_...) is a 10-bit binary sequence: 0, 10, 0, 10, 0, 10, 0, 10, 0, 10. The LED enable signal (led) is high for the first 10 cycles and low for the next 10. The segment data signals (seg) are 7-bit binary sequences: 1111100, 0100111, 0100111, 0100111, 0100111, 0100111, 0100111, 0100111, 0100111, 0100111. The reset signal (rst) is high for the first 10 cycles and low for the next 10.

Interval: 147 ps      Start:      End:

Timing diagram showing the clock (C) and data input (D) signals. The clock is a square wave with a period of 147 ps. The data input is a sequence of 16 bits: 1001111101001111. The output (Q) is shown as a square wave that follows the data input.

Simulation Waveform Editor - D:/intelfPGA\_lite/18.1/traffic\_light/traffic\_light - traffic\_light - [traffic\_light\_20210608111113.sim.vwf (Read-Only)]

Timing diagram for the 74161 counter circuit. The diagram shows the waveforms for SEL, clk, dir, mul\_10, led, sec\_10, sec\_1, seg, and rst over a 315 ns interval. The clk signal is a periodic square wave. The rst signal is a single pulse at the beginning. The other signals show the state of the counter and its outputs over time.



## Discussions

考慮到七段顯示器不能同時，因此顯示器的切換方法可以遵照 HW6 的時鐘作法即可。

由於該作業和先前 24 小時時鐘的作法大同小異，因此只需要將時鐘的各位數改成方向和秒數，並設定在不同秒數下 LED 燈的顯示狀況即可。