邏輯設計實驗 Lab08 結報

104060012 邱怡庭

1 Implement a stopwatch function with the FPGA board.

1.1 Use the four 14SD as the display. The left two digits represent the minute and the right two digits represent the second.

1.2 Use two push buttons to control the function. Use one button to control start/stop and the other to control the lap and reset. When the stopwatch counts, press the 'lap' button will freeze the 14SD display but the stopwatch continues counting, and when press the 'lap' button again, the 14SD display will start to show current time.

Design Specification

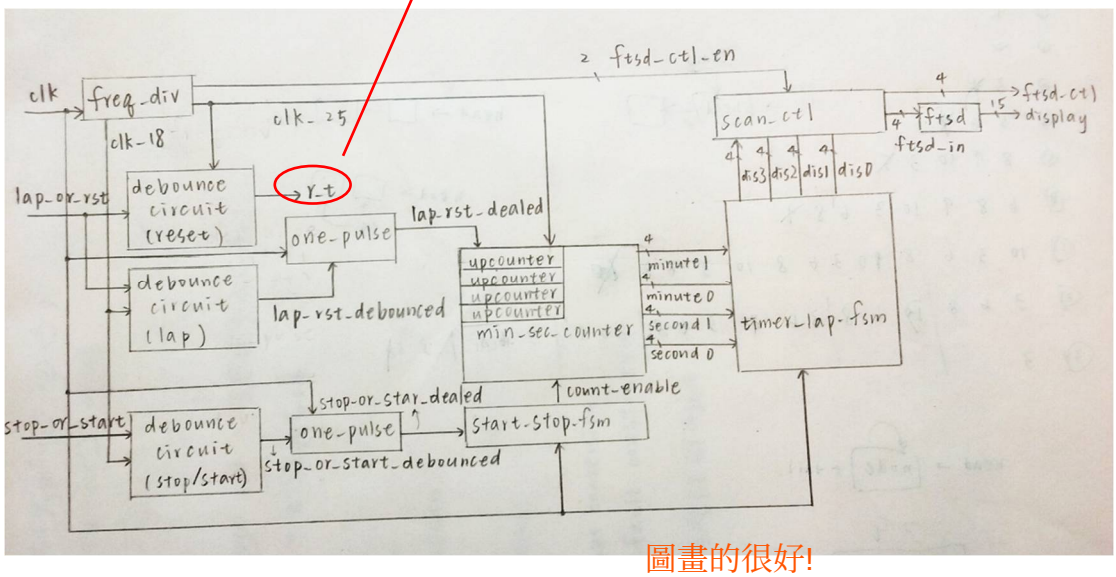
output : [3:0] ftsd_ctl, [14:0] display

input : clk, lap_or_rst, stop_or_start

wire : [1:0]ftsd_ctl_en, r_t, clk_25, clk_18, lap_rst_debounced, stop_or_start_debounced, lap_rst_dealed, stop_or_start_dealed, [3:0] minute1, [3:0]minute0, [3:0]second1, [3:0]second0, [3:0] ftsd_in, [3:0]dis3, [3:0]dis2, [3:0]dis1, [3:0]dis0, count_enable

reset接給誰? 那些線會用到?

block diagram:

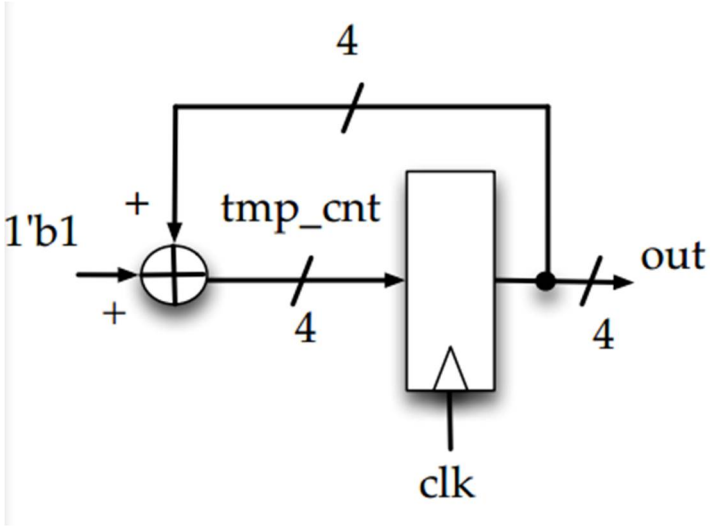


圖畫的很好!

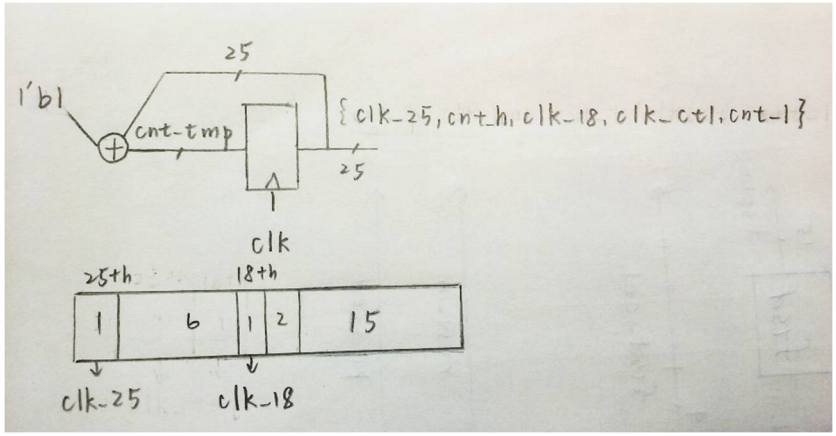
Design Implementation

logic function / logic diagram:

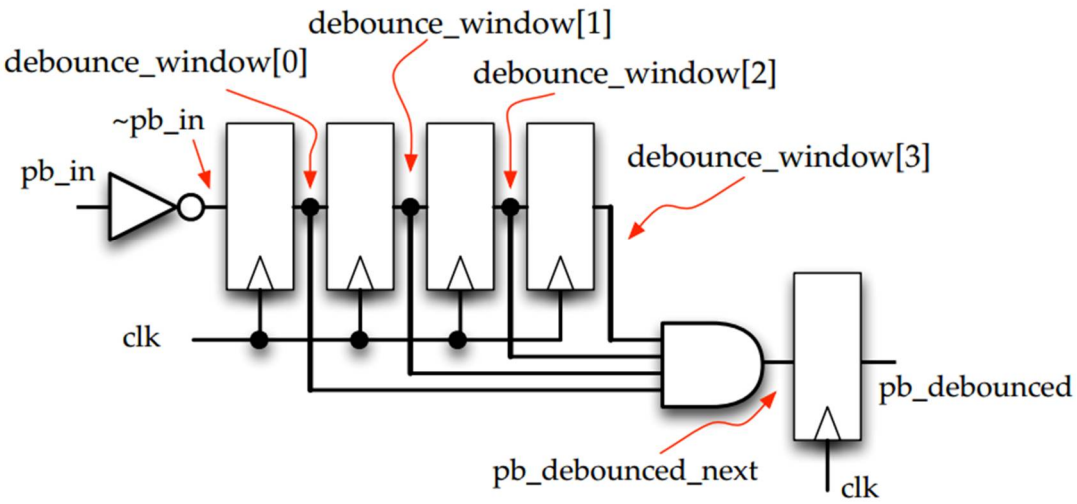
upcounter



freq_div:

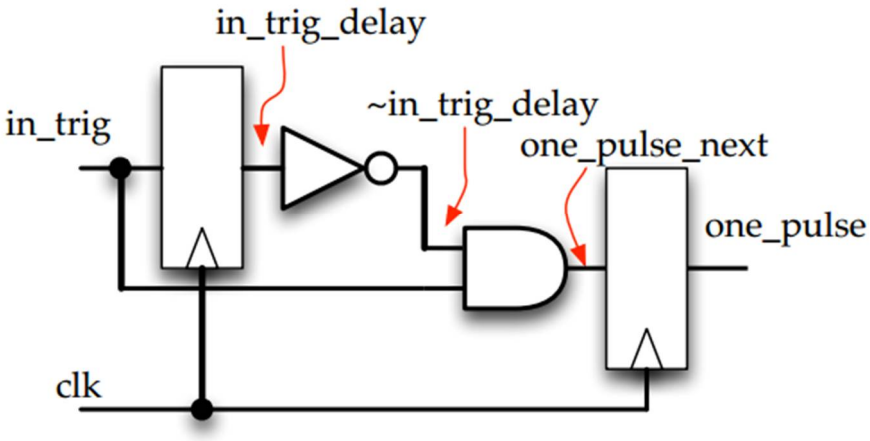


debounce_circuit



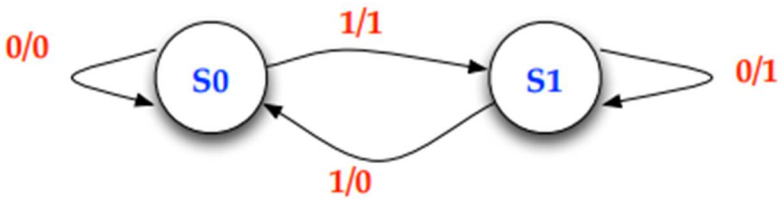
1. When all 4 bits of the registers are high the output of the debounce circuit changes to high
2. 此實驗有三個 debounce_circuit 分別控制 reset,lap,stop/start

one_pulse



1. When one presses the push button for a short moment, the time that the switch is closed (ms range) is usually much longer than one clock period (μs or ns range). The one-pulse circuit generates only a one-clockperiod-long pulse every time the push button is hit, independent of the time one keeps the button pressed
2. 此實驗有兩個 one_pulse 分別控制 lap/reset,stop/start

fsm



此實驗有兩個 fsm 分別控制 lap/reset,stop/start

scan_ctl

if ftsd_ctl_en=00

->控制第一個 14 段顯示器

if ftsd_ctl_en=01

->控制第二個 14 段顯示器

if ftsd_ctl_en=10

->控制第三個 14 段顯示器

if ftsd_ctl_en=11

->控制第四個 14 段顯示器

ftsd

當 in=4'd0: display = 15'b0000_0011_1111_111; //0

當 in=4'd1: display = 15'b1111_1111_1011_011; //1

當 in=4'd2: display = 15'b0010_0100_1111_111; //2

當 in=4'd3: display = 15'b0000_1100_1111_111; //3

當 in=4'd4: display = 15'b1001_1000_1111_111; //4

當 in=4'd5: display = 15'b0100_1000_1111_111; //5

當 in=4'd6: display = 15'b0100_0000_1111_111; //6

當 in=4'd7: display = 15'b0001_1111_1111_111; //7

當 in=4'd8: display = 15'b0000_0000_1111_111; //8

當 in=4'd9: display = 15'b0000_1000_1111_111; //9

default: display = 15'b1111_1111_1111_111; //DEF

I/O pin assignment:

```
NET "display[0]" LOC = U5;
NET "display[1]" LOC = T7;
NET "display[2]" LOC = R7;
NET "display[3]" LOC = V7;
NET "display[4]" LOC = V4;
NET "display[5]" LOC = T4;
NET "display[6]" LOC = T3;
NET "display[7]" LOC = R5;
NET "display[8]" LOC = N5;
NET "display[9]" LOC = R3;
NET "display[10]" LOC = U7;
NET "display[11]" LOC = T5;
NET "display[12]" LOC = V5;
NET "display[13]" LOC = N4;
NET "display[14]" LOC = P6;
NET "ftsd_ctl[0]" LOC = V8;
NET "ftsd_ctl[1]" LOC = U8;
```

```
NET "ftsd_ctl[2]" LOC = V6;
NET "ftsd_ctl[3]" LOC = T6;
NET "lap_or_rst" LOC = T2;
NET "stop_or_start" LOC = U1;
NET "clk" LOC = R10;
```

Discussion:

此題結合前幾個實驗，包含:

1. 不同長度的 clock 同樣的按鈕可以表現不同功能
2. 組合 upcounter
3. fsm 中不同 state 的設置

※在 timer_lap_fsm 中，當按下 lap 時，顯示當前的時間，此時的 output 為上一個 clock 時存取之來自 counter 的 input，但 timer 仍繼續執行沒有停止，當再次按下 lap 時，回歸計時的畫面，output 則為當前 clock 下存取之來自 counter 的 input。

這段只是解釋何為lap,
但沒解釋到怎麼實現lap的功能。

2 Implement a timer (can support as long as 23:59) with the following functions.

2.1 Use one DIP switch as the 'setting' control. When the 'setting' is ON, you can usetwo buttons to set the hour and minute.

2.2 Use other two buttons to control the timer operation. One button for start/stop and the other button for pause/resume.

2.3 When the time goes to 0, light up all the LEDs.

Design Specification

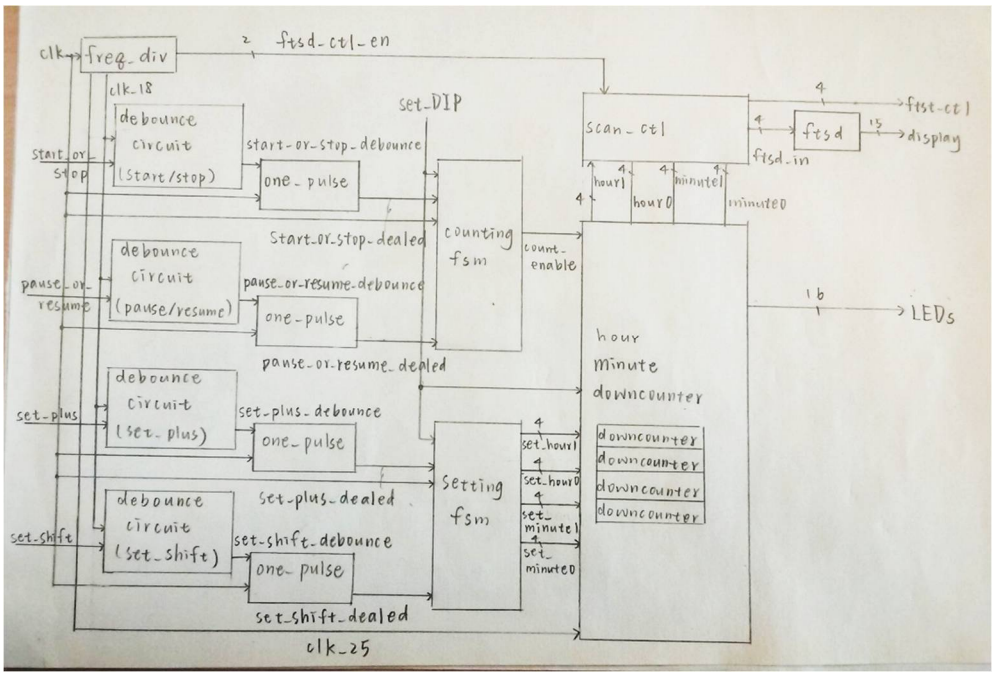
output : [3:0] ftsd_ctl, [14:0] display, [15:0] LEDs

input : clk, rst_n, set_DIP, set_plus, set_shift, start_or_stop, pause_or_resume

wire : [1:0]ftsd_ctl_en, r_t, clk_25, clk_18, set_plus_debounce,
set_shift_debounce, set_plus_dealed, set_shift_dealed,
start_or_stop_debounce, pause_or_resume_debounce,
start_or_stop_dealed, pause_or_resume_dealed, [3:0]
minute1, [3:0]minute0, [3:0] hour1, [3:0] hour0, [3:0] ftsd_in, [3:0] ,
set_minute1, [3:0] set_minute0, [3:0] set_hour1, [3:0] set_hour0,
count_enable

block diagram:

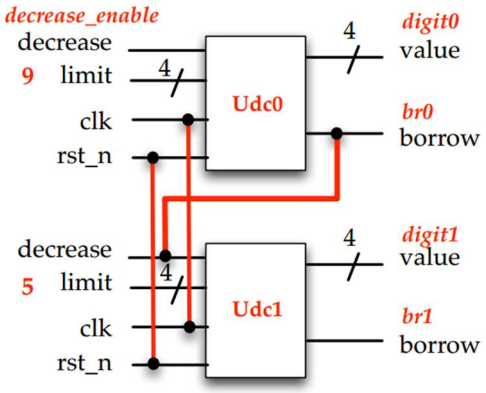
good!!!!!!!



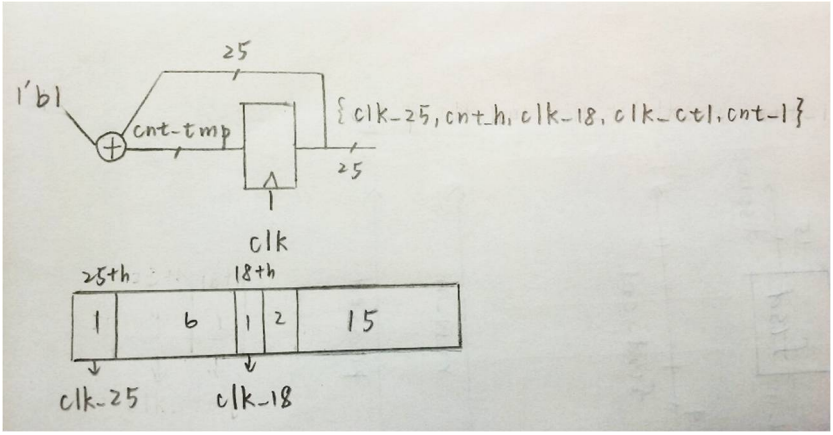
Design Implementation

logic function / logic diagram:

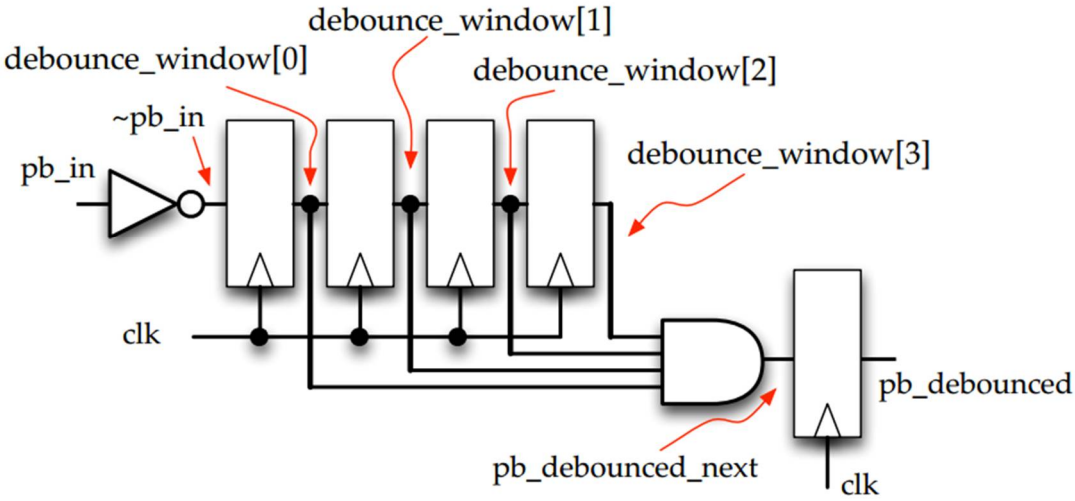
downcounter



freq_div:

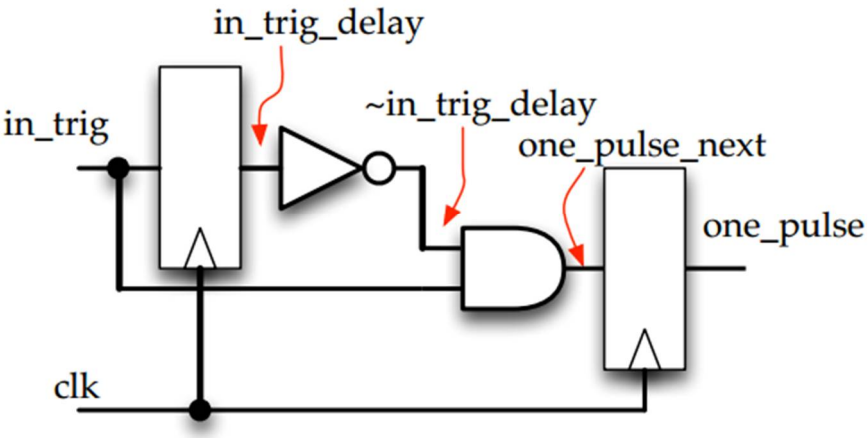


debounce_circuit



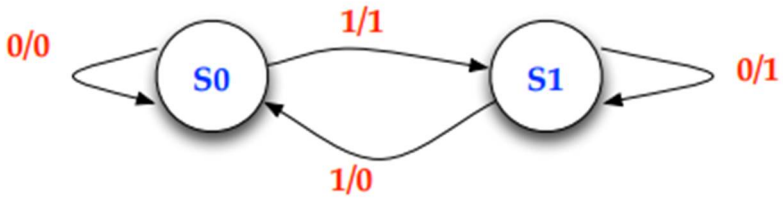
1. When all 4 bits of the registers are high the output of the debounce circuit changes to high
2. 此實驗有四個 debounce_circuit 分別控制 stop/start,pause/resume,plus,shift

one_pulse



1. When one presses the push button for a short moment, the time that the switch is closed (ms range) is usually much longer than one clock period (μs or ns range). The one-pulse circuit generates only a one-clockperiod-long pulse every time the push button is hit, independent of the time one keeps the button pressed
2. 此實驗有四個 one_pulse 分別控制 stop/start,pause/resume,plus,shift

fsm



此實驗有兩個 fsm 分別控制 counting, setting

scan_ctl

if ftsd_ctl_en=00
->控制第一個 14 段顯示器
if ftsd_ctl_en=01
->控制第二個 14 段顯示器
if ftsd_ctl_en=10
->控制第三個 14 段顯示器
if ftsd_ctl_en=11
->控制第四個 14 段顯示器

ftsd

當 in=4'd0: display = 15'b0000_0011_1111_111; //0
當 in=4'd1: display = 15'b1111_1111_1011_011; //1
當 in=4'd2: display = 15'b0010_0100_1111_111; //2
當 in=4'd3: display = 15'b0000_1100_1111_111; //3
當 in=4'd4: display = 15'b1001_1000_1111_111; //4
當 in=4'd5: display = 15'b0100_1000_1111_111; //5
當 in=4'd6: display = 15'b0100_0000_1111_111; //6
當 in=4'd7: display = 15'b0001_1111_1111_111; //7
當 in=4'd8: display = 15'b0000_0000_1111_111; //8
當 in=4'd9: display = 15'b0000_1000_1111_111; //9
default: display = 15'b1111_1111_1111_111; //DEF

I/O pin assignment:

```
NET "display[0]" LOC = U5;
NET "display[1]" LOC = T7;
NET "display[2]" LOC = R7;
NET "display[3]" LOC = V7;
```

```
NET "display[4]" LOC = V4;
NET "display[5]" LOC = T4;
NET "display[6]" LOC = T3;
NET "display[7]" LOC = R5;
NET "display[8]" LOC = N5;
NET "display[9]" LOC = R3;
NET "display[10]" LOC = U7;
NET "display[11]" LOC = T5;
NET "display[12]" LOC = V5;
NET "display[13]" LOC = N4;
NET "display[14]" LOC = P6;
NET "ftsd_ctl[0]" LOC = V8;
NET "ftsd_ctl[1]" LOC = U8;
NET "ftsd_ctl[3]" LOC = T6;
NET "ftsd_ctl[2]" LOC = V6;
NET "clk" LOC = R10;
NET "rst_n" LOC = T2;
NET "set_plus" LOC = M5;
NET "set_shift" LOC = L6;
NET "set_DIP" LOC = T1;

NET "pause_or_resume" LOC = U2;
NET "start_or_stop" LOC = U1;

NET "LEDs[15]" LOC = H5;
NET "LEDs[14]" LOC = H6;
NET "LEDs[13]" LOC = F1;
NET "LEDs[12]" LOC = F2;
NET "LEDs[11]" LOC = J6;
NET "LEDs[10]" LOC = J7;
NET "LEDs[9]" LOC = G1;
NET "LEDs[8]" LOC = G3;
NET "LEDs[7]" LOC = K6;
NET "LEDs[6]" LOC = L7;
NET "LEDs[5]" LOC = H3;
NET "LEDs[4]" LOC = H4;
NET "LEDs[3]" LOC = K5;
NET "LEDs[2]" LOC = L5;
```

```
NET "LEDs[1]" LOC = K3;
NET "LEDs[0]" LOC = K4;
```

Discussion:

設計類似電子手錶的功能，當 DIPswitch=1 時，則可進行設定，使用 set_plus 鈕加一，set_shift 鈕來選擇控制的位置。當歸零的時候讓 LED 全亮，也就是讓 LEDs=1。

Conclusion:

lab08 只是結合前幾個 lab 使用過的功能，雖然有點複雜，但其實概念並不困難!

8-1
Design Specification (2/2)
block diagram of the design or Logic Diagram (3/4)
I/O pin assignment (1/1)
Discussion +Conclusion (3/3)
Function explanation (3/5)
8-2
Design Specification (3/3)
block diagram of the design or Logic Diagram (5/5)
I/O pin assignment (2/2)
Discussion +Conclusion (5/5)
Function explanation (5/5)
Bonus (0/2)