

## Lab8

### 1. Specification

Speaker

Input : C, D, E, up, down, clk, rst\_n\_init

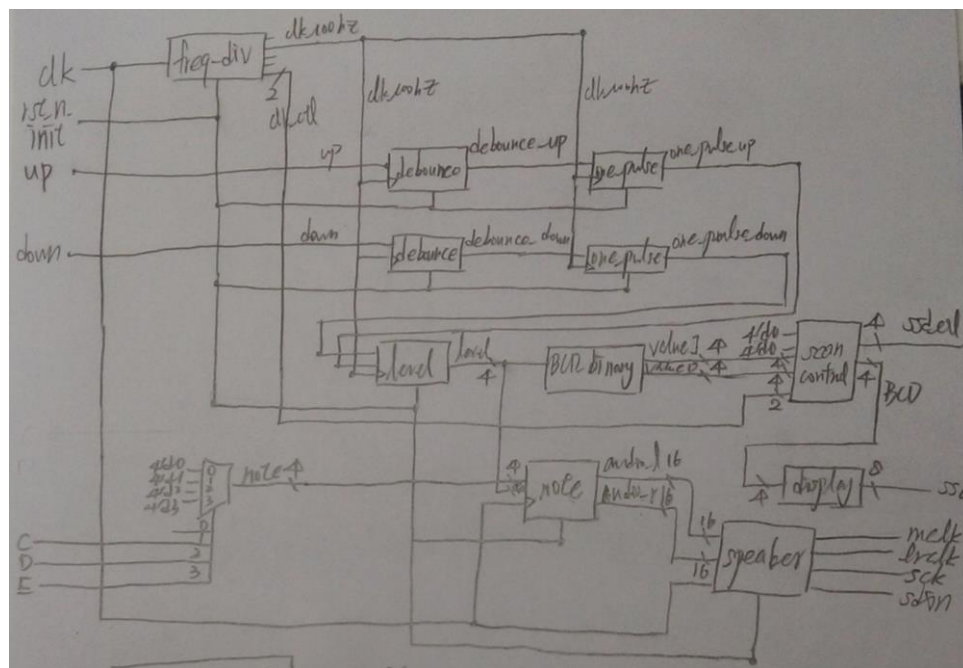
Output : mclk, lrclk, sck, sdin, [7:0] ssd, [3:0] ssd\_ctl

Mclk, lrclk, sck, sdin 為給 speaker 的 output，按下 C 時，發出 Do；按下 D 時，發出 Re；按下 E，發出 Mi

### 2. Implementation

這次 lab 因為有用到按鈕和七段顯示器，所以有關前兩者的 module 都會使用到。除此之外，我還有另外寫了 note, speaker, BCD binary converter, adder & subtracter

以下為各 module 間 wire 的連結



I/O	Ssd[7]	Ssd[6]	Ssd[5]	Ssd[4]	Ssd[3]	Ssd[2]	Ssd[1]	Ssd[0]
Pin	W7	W6	U8	V8	U5	V5	U7	V7

I/O	Ssd_ctl[3]	Ssd_ctl[2]	Ssd_ctl[1]	Ssd_ctl[0]
Pin	W4	V4	U4	U2

I/O	MCLK	LRCLK	SCK	SDIN
-----	------	-------	-----	------

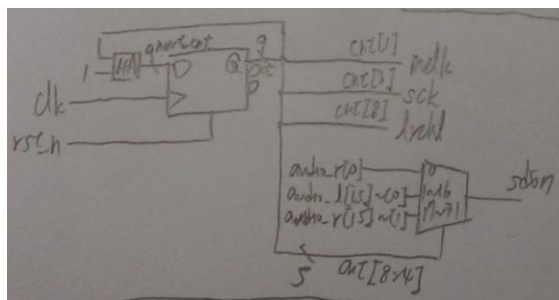
Pin	A14	A16	B15	B16
-----	-----	-----	-----	-----

I/O	C	D	E	Up	down	clk	Rst_n_init
Pin	W19	U18	T17	T18	U17	W5	V17

## Speaker

需要 generate 25MHz for MCLK, LRCLK 25MHZ/128 for lrclk, 25MHz/4 for SCK。SDIN 則是依序 output audio\_r[0], audio\_l[15] ~ audio\_l[0], audio\_r[15] ~ audio\_r[1]，並且會 delay 一個 SCK cycle

Output 那些 clk 的方法跟 frequency divider 的寫法很像，我設一個 counter，每次 clock 被 trigger 就+1，不過差別在於，因為這次 clock 都是除以二的次方，所以只要取不同的 bit 就可以取得不同頻率的 clock，不用每個頻率都設一個 counter。SDIN 則是在設一個 counter 並是以 SCK 為單位，counter 為不同數字時，用 case 顯示不同的 serial output。



```
assign mclk = cnt[0];
assign sck = cnt[1];
assign lrclk = cnt[2];

assign next_cnt = cnt + 9'd1;
```

## note

這個 module 負責產生不同頻率音量的聲音，不同 level 的時候，振幅會不同，而 b\_clk 為 1 時，代表波峰，為 0 時，代表波谷。至於不同頻率，則是用類似 frequency divider 的方式，數到特定數字後，將 b\_clk 變號。而不同的頻率要數到不同的數字。

```
always @*
begin
    if(cnt == limit && note != 4'b0)
```

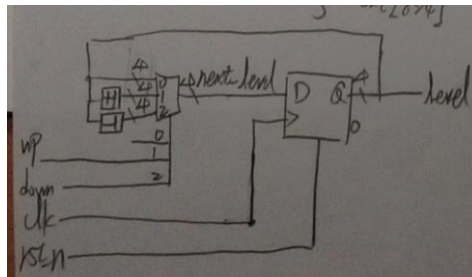
```

begin
    next_cnt = 18'd0;
    next_b_clk = ~b_clk;
end
else
begin
    next_cnt = cnt + 18'd1;
    next_b_clk = b_clk;
end
end

```

### level

此為可以加或減的 module，因為之前 counter 和 down counter 都分開寫，所以要再寫一個可以加也可以減的 module。



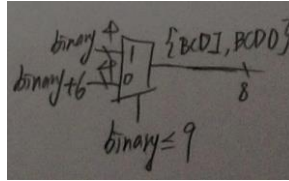
```

always @*
begin
    if(up == 1'b1)
        next_level = level + 4'd1;
    else if(down == 1'b1)
        next_level = level - 4'd1;
    else
        next_level = level;
end

```

### BCD\_binary

BCD binary converter。之前一直沒有寫過這個 module 因為我大都直接用 BCD 來處理需要顯示在七段顯示器的數據。但這次 level 需要以 binary 的形式 input 進去 note module，所以要寫一個 BCD binary converter，來轉換 level 成為 BCD 的形式。

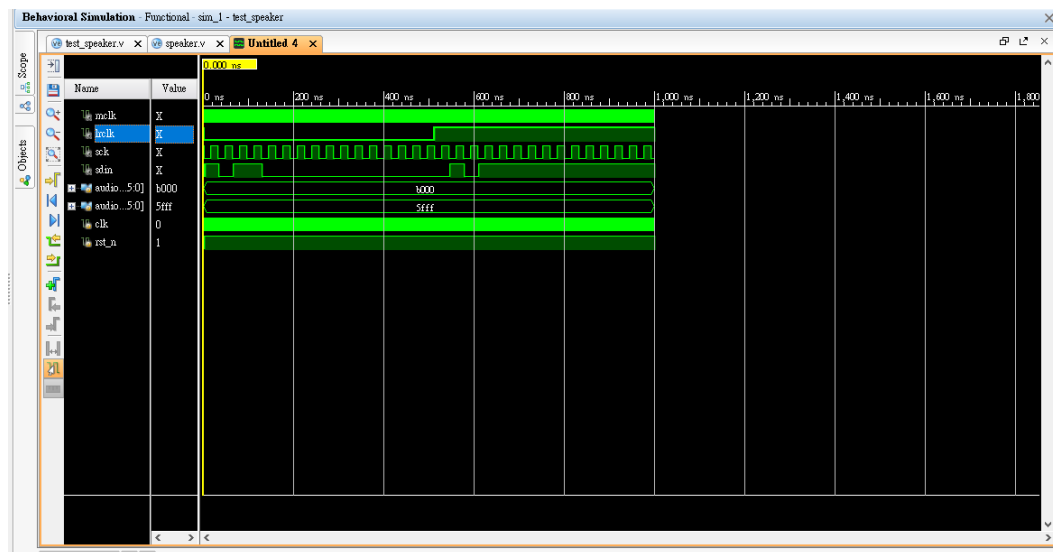


```

always @*
begin
    if(binary <= 4'd9)
    begin
        BCD1 = 4'd0;
        BCD0 = binary;
    end
    else
        {BCD1, BCD0} = binary + 4'd6;
    End
end

```

### 3. Simulation Speaker

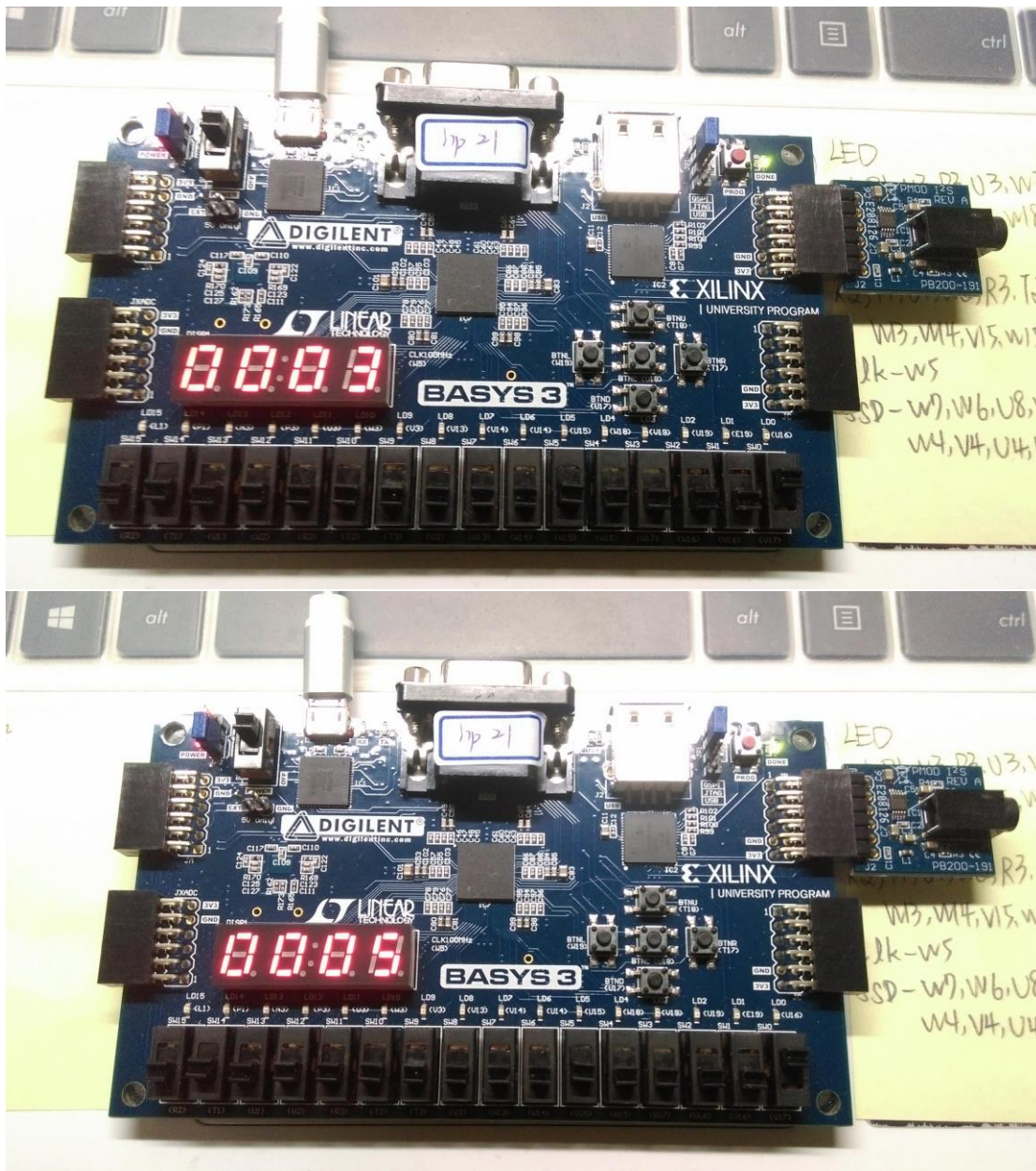


如圖所示，32 個 SCK 為一個 LRCLK，而 SDIN 要在這段期間恰傳出一組雙聲道數據

**Level**



這次 lab 讓我學會如何處理 parallel to serial 的 data，還有如何使用 speaker，以及有關聲音如何產生的概念



## 6. Reference

### 08\_speaker

P. 2 PMod 的腳位

P. 4, 6, 7 有關 speaker clock

P. 5 speaker module 的連結

P. 8 buzzer 的 frequency

P. 9 the verilog of buzzer