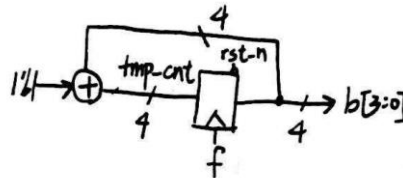


Lab4_1

Design Specification

- ✓ For a 4-bits binary up counter:
 - Input: rst_n, f
 - Output: b[3:0]
- ✓ Draw the block diagram of the design.



Design Implementation

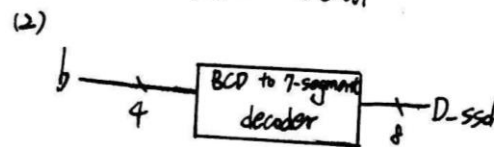
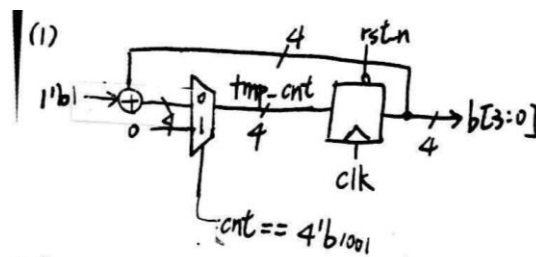
- ✓ Input f 為 Lab3 的 26 frequency divider 除頻後的 1 Hz。
- ✓ Logic function
 - 當 rst_n 為下緣或 f 為上緣時，判斷若 rst_n = 0，則 b 歸零；否則 b 加 1 直到 b = 16
- ✓ I/O pin

I/O	F	rst_n	clk_out
VOC	W5	V17	U16

Lab4_2

Design Specification

- ✓ For a BCD up counter:
 - Input: rst_n, clk
 - Output: D_ss[7:0], b[3:0], B[3:0]
- ✓ Draw the block diagram of the design.



Input: clk, rst_n.

Output: b[3:0], D_ss[7:0], B

Design Implementation

- ✓ 當 rst_n 為下緣或 clk 為上緣時，判斷若 rst_n = 0，則 b 歸零；否則 b 加 1
- ✓ 由於此為 BCD counter，且為 4-bit，因此最大值為 9。因此當 b = 9 時，BCD counter 歸零重

新 count。

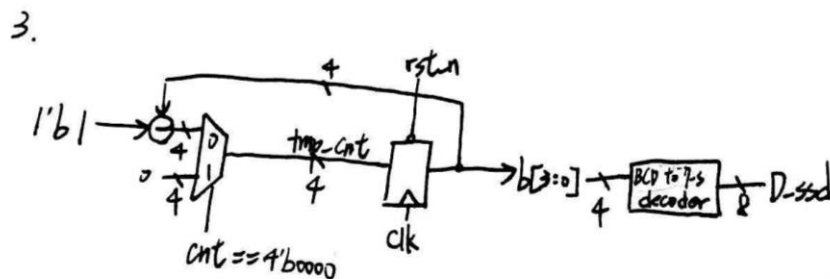
- ✓ 因為需要用 7-segment display 顯示，因此需要用到 Lab2 的 3 to 8 decoder 將其轉為 7-segment。
- ✓ I/O pin

I/O	rst_n	D_ssd7	D_ssd6	D_ssd5	D_ssd4	D_ssd3	D_ssd2	D_ssd1	D_ssd0
Site	V17	W7	W6	U8	V8	U5	V5	U7	V7
I/O	clk	b3	b2	b1	b0	B3	B2	B1	B0
Site	W5	V19	U19	E19	U16	W4	V4	U4	U2

Lab4_3

Design Specification

- ✓ For a BCD down counter:
Input: rst_n, clk
Output: D_ssd[7:0], b[3:0], B[3:0]
- ✓ Block diagram



Design Implementation

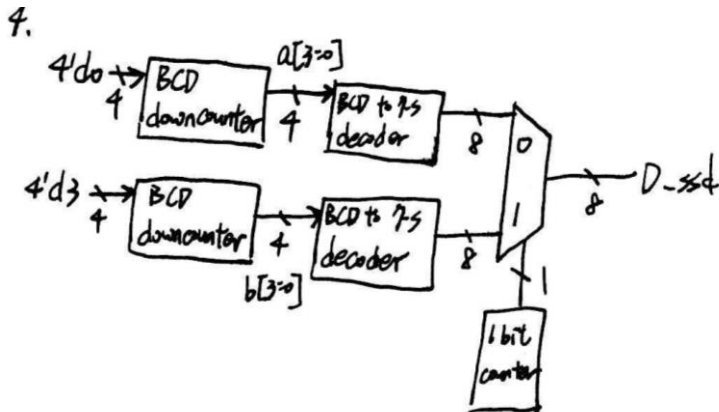
- ✓ 當 rst_n 為下緣或 clk 為上緣時，判斷若 rst_n = 0，則 b 歸零；否則減 1
- ✓ 由於此為 BCD counter，且為 4-bit，因此最小值為 0。因此當 b = 0 時，b 回到 9 重新 count。
- ✓ 因為需要用 7-segment display 顯示，因此需要用到 Lab2 的 3 to 8 decoder 將其轉為 7-segment。
- ✓ I/O pin

I/O	rst_n	D_ssd7	D_ssd6	D_ssd5	D_ssd4	D_ssd3	D_ssd2	D_ssd1	D_ssd0
Site	V17	W7	W6	U8	V8	U5	V5	U7	V7
I/O	clk	b3	b2	b1	b0	B3	B2	B1	B0
Site	W5	V19	U19	E19	U16	W4	V4	U4	U2

Lab4_4

Design Specification

- ✓ For a 30 second down counter:
Input: rst_n, clk
Output: D_ssd[7:0], b[3:0], B[3:0]
- ✓ Block diagram



Design Implementation

- ✓ 將個位數與十位數分別用 down counter 計算，當 rst_n 為下緣或 clk 為上緣時，判斷若 rst_n = 0，則 a 歸零，b 減 1；否則 a 減 1，b 不變。
- ✓ 因為需要用 7-segment display 顯示，因此需要用到 Lab2 的 3 to 8 decoder 將其轉為 7-segment。
- ✓ 由於要在 7-segment 上顯示不同數字，因此須利用 $1/2^{27}$ frequency divider 中擷取中間頻率 ($16^{\text{th}} \sim 17^{\text{th}}$)，造成視覺暫留。設定 c，若 rst_n = 0，c = 0；否則 c = cnt_ctl(cnt_ctl = c + 1'b1)。藉由 c 的值來決定顯示個位數還是十位數的數值。
- ✓ I/O pin

I/O	rst_n	D_ssd7	D_ssd6	D_ssd5	D_ssd4	D_ssd3	D_ssd2	D_ssd1	D_ssd0
Site	V17	W7	W6	U8	V8	U5	V5	U7	V7
I/O	clk	b3	b2	b1	b0	B3	B2	B1	B0
Site	W5	V19	U19	E19	U16	W4	V4	U4	U2

Discussion

前三題皆利用了上一個 lab 的 frequency divider 的頻率作為 clk，並結合加法器讓他變成 counter，其中第一題與二、三題的差別在於其為 binary，因此二、三題需要一個 mux 判斷要不要重新計算。

第四題則是結合了兩個 down counter，這裡我利用了 counter 判斷其值是否為 0 來決定個位數與十位數的 next state。另外，為了讓 7-segment 顯示不同數字，須利用 divider 擷取不同頻率讓她快速顯示個位數與十位數造成視覺暫留，讓我們看到不一樣的數字。

Conclusion

這次實驗用到了很多 lab3 的 divider，尤其是第四題，其實可以直接用 top module 去做，但我還是花了一些時間做成一個 module。之後可以練習用 top module 做做看。