Lab4: Counters

107061112 王昊文

Experiments

1. Construct a 4-bit synchronous binary up counter (b3b2b1b0) with the 1-Hz clock frequency from exp2 and use 4 LEDs for display.

I/O	$f_{crystal}$	b3	b2	b1	b0
Site	W5	V19	U19	E19	U16

Design Specification

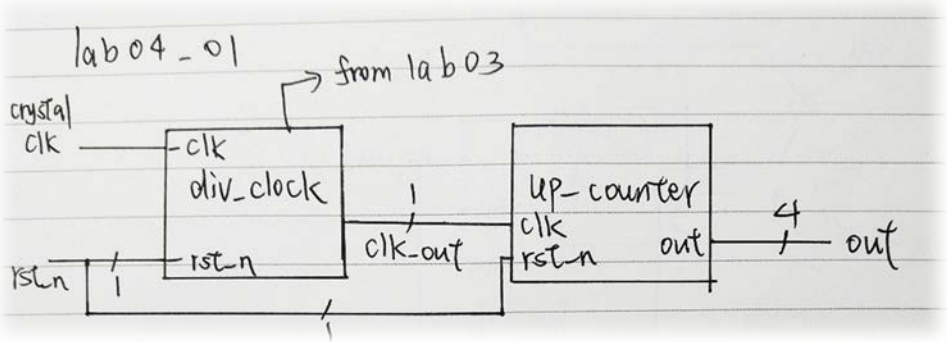
✓ I/O

Input clk // global clock input

Input rst_n // active low reset

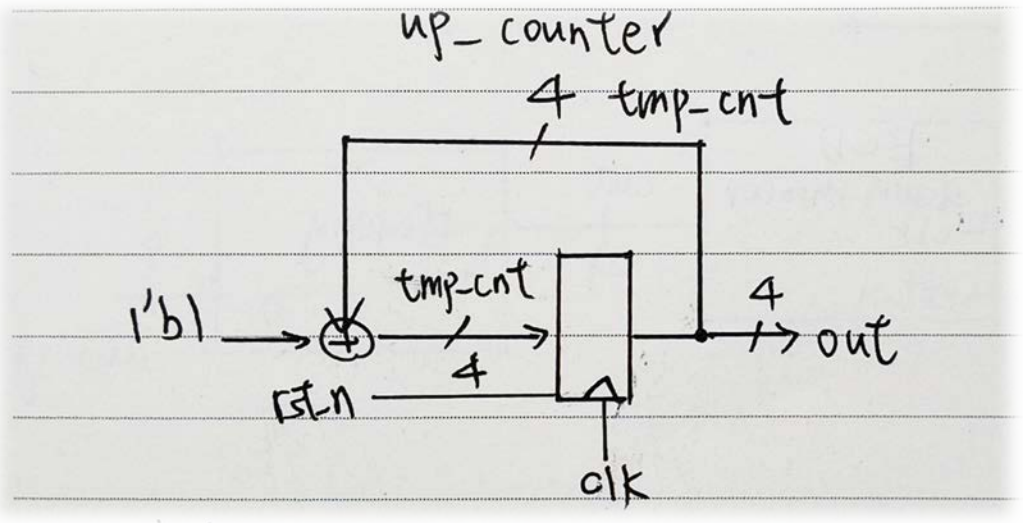
Output out // counter output

✓ Logic Block:



Div_clock 使用的是lab3除頻過的 clock。使用除頻後的 clock，將訊號送進counter中，每經過一個clock cycle往上加一。後面counter的out直接接到led的input，即可達到題目的條件。

Up_counter



以上為binary_up_counter的邏輯圖。每經過一個clock cycle會往上加一。

Design Implementation

✓ IO Pin

Input	
clk	*rst_n*
W5	R2

Output			
out[0]	out[1]	out[2]	out[3]
U16	E19	U19	V19

Discussion

這一題的難度不高，只要將之前寫過的模組拼起來就可以了。

2. Construct a single digit BCD up counter with the divided clock as the clock frequency and display on the seven-segment display.

Design Specification

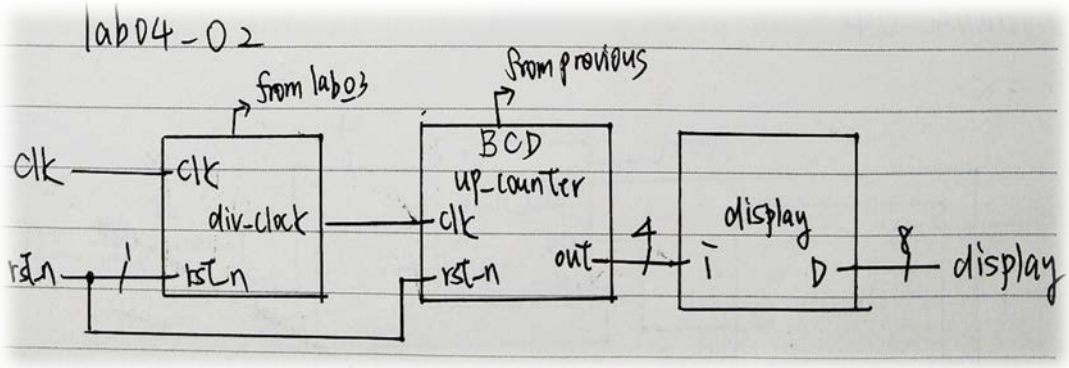
✓ I/O

Input clk // global clock input

Input rst_n // active low reset

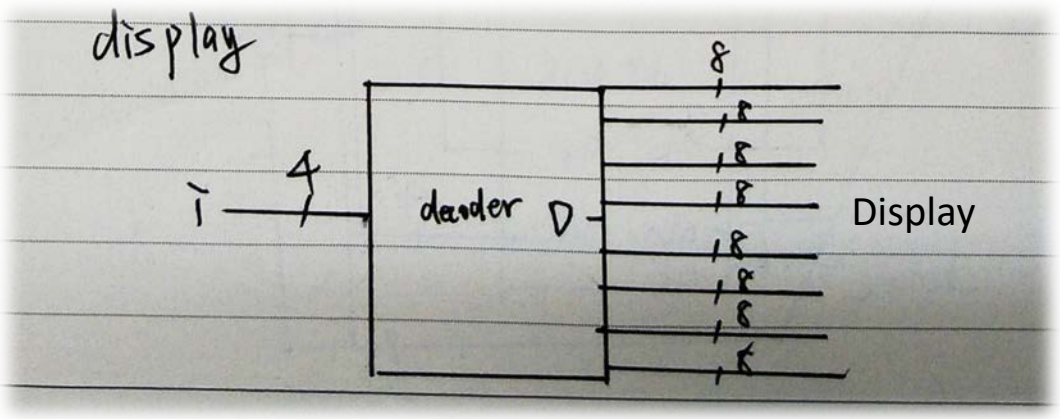
Output [7:0]display // seven segment display

✓ Logic Block:



作法與上一題類似，基本上除頻的clock引用上一個lab，而counter部分，與上一題類似，只是將上限設定為9。將counter的輸出連接到display裡面做decoding輸出成seven segment的格式。

Display 模組



基本上沿用lab02的顯示模組。

Design Implementation

Input	
clk	*rst_n*
W5	R2

Output							
display[0]	***display[1]***	***display[2]***	***display[3]***	***display[4]***	***display[5]***	***display[6]***	***display[7]***
V7	U7	V5	U5	V8	U8	W6	W7

Discussion

將之前lab的模組接起來就可以輕鬆完成這一題。從這裡開始發現模組越接越多，要是Block Diagram沒有畫好就很容易接錯。

3. Construct a single digit BCD down counter with the divided clock as the clock frequency and display on the seven-segment display.

Design Specification

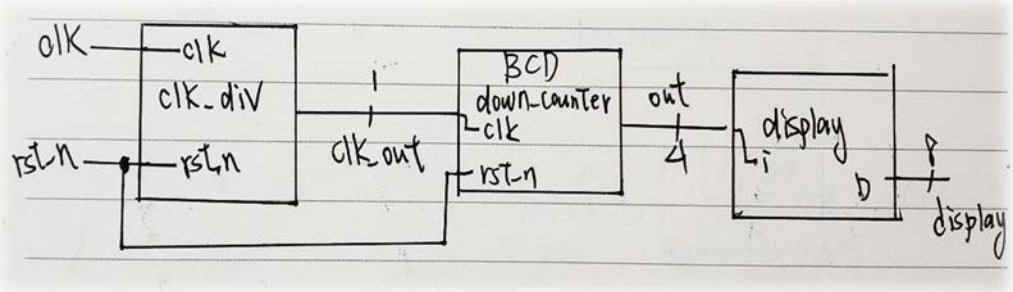
✓ I/O

Input clk // global clock input

Input rst_n // active low reset

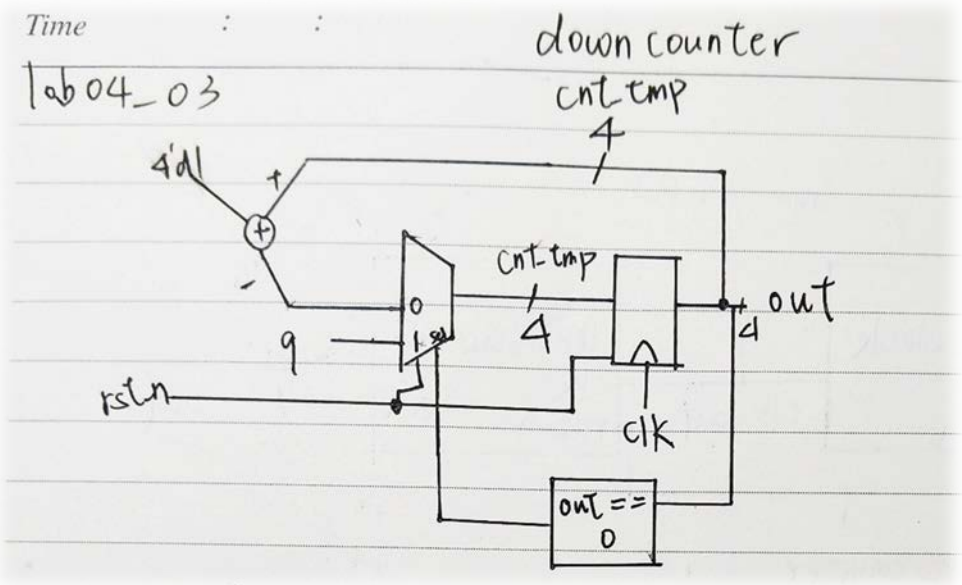
Output [7:0]display // seven segment display

✓ Logic Block:



作法與上一題類似，基本上除頻的clock引用上一個lab，Counter的部分將上一題的逐次加一變成減一，然後因為是BCD down counter，每數到0的時候又重新從9開始。

Down counter 模組



設計方式與up counter類似，但因為是往下數，必須有一個判斷out是否為0的條件，然後通過MUX來選擇進入cnt_tmp的值。

Design Implementation

Input	
clk	*rst_n*
W5	R2

Output							
display[0]	***display[1]***	***display[2]***	***display[3]***	***display[4]***	***display[5]***	***display[6]***	***display[7]***
V7	U7	V5	U5	V8	U8	W6	W7

Discussion

基本上也是一樣，將up counter改成逐次減一，初始值從9開始倒數即可。

4 (Bonus) Construct a 30 seconds count down timer (stop at 00).

Design Specification

✓ I/O

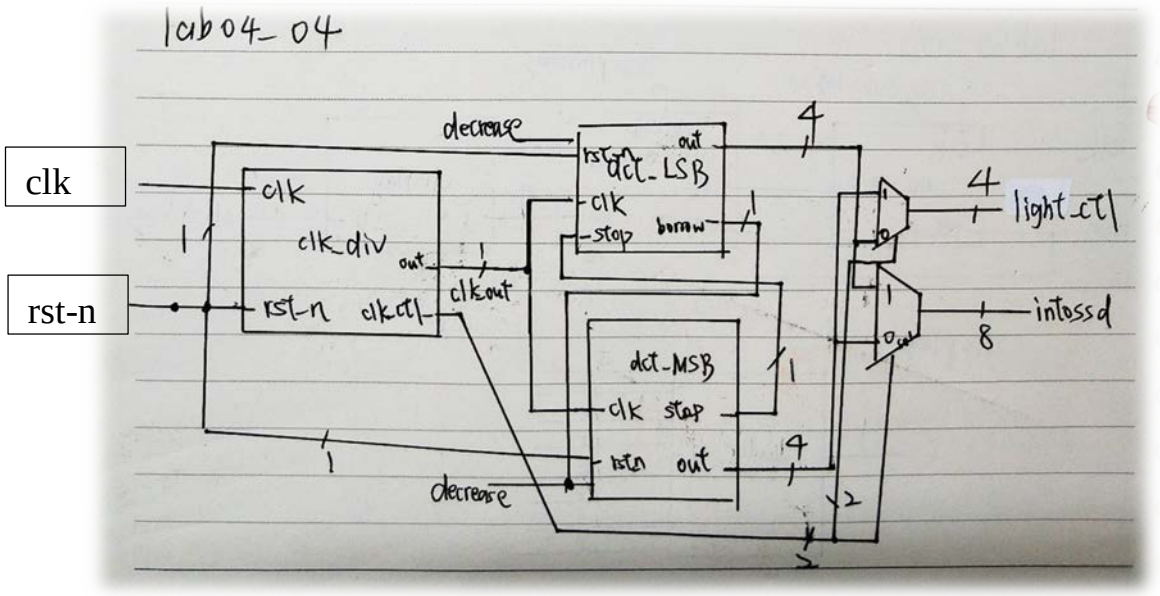
Input clk // global clock input

Input rst_n // active low reset

Output [7:0]num // seven segment display

Output [3:0]lightctl // control which light on

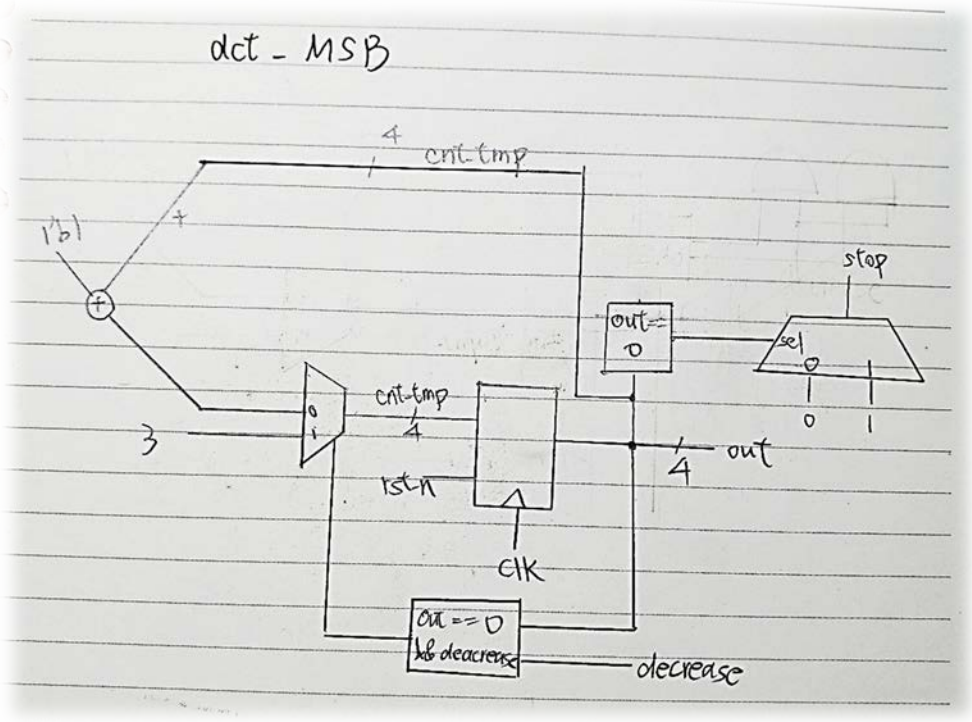
✓ Logic Block:



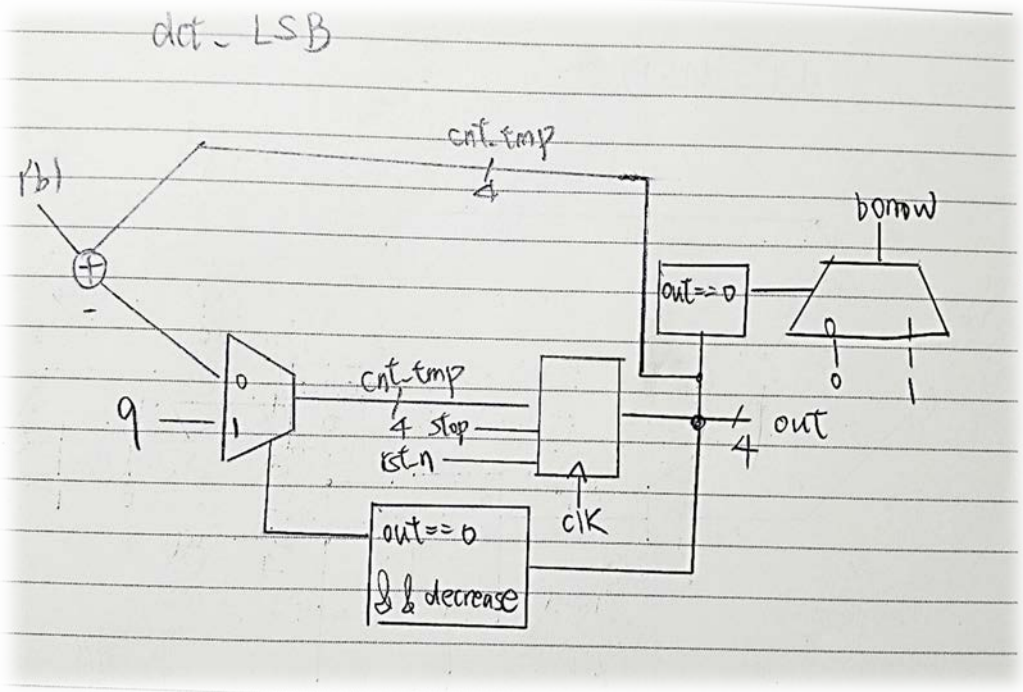
這一題題目要求顯示兩位數30秒倒數，倒數的部分我使用兩個down counter來建置，分為lsb 跟 msb 。一樣先使用一個除頻過的clock當作兩個counter的clock input。再來在lsb上面裝一個borrow的訊號。每當lsb_counter數到零的時候，將訊號傳給msb的decrease，告訴msb要往下減一。當msb down counter數到0的時候會發出一個stop的訊號，告訴lsb已經沒有位可以借了。這時只要

lsb本身數到0就會停在00了。然後剩下就是顯示的部分，與lab03的作法一樣，製作一個類似掃描器，在一個clock cycle內掃描，即可完成顯示不同數字的功能。

Down counter 模組



基本上是down counter的模型，Msb的部分我多設計了一個stop的訊號，當out為零的時候將stop輸出給lsb，告知已經沒有位可以借了。



Lsb也是down counter的模型，但是會產生一個borrow信號給msb，同時在flip flop的部分我用了一個stop的訊號，會將近入counter的值全部assign成0，有點像reset的功能。

Design Implementation

Input	
clk	*rst_n*
W5	R2

Output							
num[0]	***num[1]***	***num[2]***	***num[3]***	***num[4]***	***num[5]***	***num[6]***	***num[7]***
V7	U7	V5	U5	V8	U8	W6	W7

Output			
lightctl[0]	*lightctl[1]*	*lightctl[2]*	*lightctl[3]*
U2	U4	V4	W4

Discussion

做完之後我檢討了一下我這一題的設計，其實應該有更好的作法。譬如那些counter模組，如果今天是要設計多bit以上，每一個模組都長的不一樣會很麻煩。這一題的部分我應該將每一個down counter的模組都長一樣，然後用一個類似top module的概念將這些down counter串起來，直接指定這個top module數到零的時候不要繼續數即可。有了這個觀念會在之後的lab做改進。

Conclusion

這次的lab讓我更看見了邏輯圖的重要性。隨著邏輯越來越複雜，更不可能直接在腦中響好直接打。Bonus的部分算是小有挑戰性，花了不少時間構思，最後debug也花了不少時間，我發現自己邏輯有很多的錯誤導致code一直出現錯誤，這次的bonus也讓我繪製邏輯圖的技術大幅進步。