Lab 4

110060027 朱豐蔚

I. exp_1(1 Hz binary down counter)

Design Specification

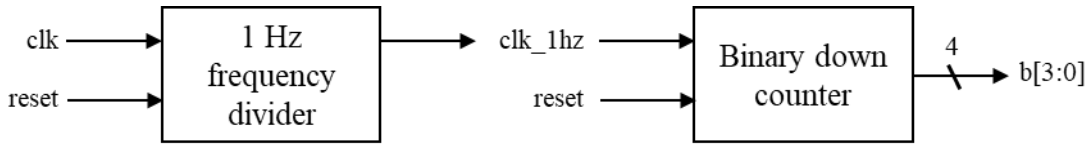
Input: clk, reset.
Output: b[3:0].
Wire: clk_1hz.

Design Implementation

Specification of the frequency divider:

使用一數 50M 次的 counter 來對原本 100M Hz 的 clk 進行除頻，使輸出為 1 Hz 的訊號 clk_1hz，即達成題目所求。

Block diagram:



Design flow:

考慮 down counter，我使用 always 的寫法，初始值設為 b = 0，並使 b_next 永遠都為 b – 1，而當 clk 的正緣訊號產生時，就讓 b 的值等於 b_next，即可達成 down counter 的設計。

I/O pins assignment

Input:

I/O	clk	reset
LOC	W5	V17

Output:

I/O	b[3]	b[2]	b[1]	b[0]
LOC	V19	U19	E19	U16

Discussion:

本實驗是考驗我們對 block 互相串接的熟悉程度，使用在 Lab3 製作的 1Hz 除頻器，並將在 Lab3 中大量使用的 up counter 改造為 down counter，透過兩者的串接，就可以達到題目所要求的效果。

II. exp_2 (1 Hz binary down counter with SSD)

Design Specification

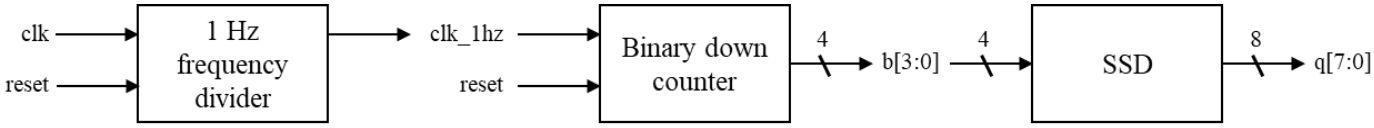
Input: clk, reset.
Output: q[7:0].
Wire: clk_1hz, b[3:0].

Design Implementation

Specification of the frequency divider:

使用一數 50M 次的 counter 來對原本 100M Hz 的 clk 進行除頻，使輸出為 1 Hz 的訊號 clk_1hz，即達成題目所求。

Block diagram:



Design flow:

同 exp_1 中的除頻器與 down counter，可以直接拿來做使用，但為了要使結果能夠呈現於 FPGA 板上，我們還需要引用 Lab2-2 中的 SSD 模組，來讓 4-bit 的 binary 訊號轉為 7 段顯示器的 8-bit 顯示輸入。

I/O pins assignment

Input:

I/O	clk	reset
LOC	W5	V17

Output:

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

Discussion:

在 exp_1 中，我們使用 4 個燈去做輸出，雖然已經很習慣 binary 訊號的讀取，但一加上了 exp_2 的 SSD，更能讓我們一眼就看出數字的變化有沒有符合我們的預期，而這樣的顯示也讓我們在隨著 clk 一秒一秒數的過程更有成就感。

III. exp_3 (0.5Hz BCD down counter with SSD)

Design Specification

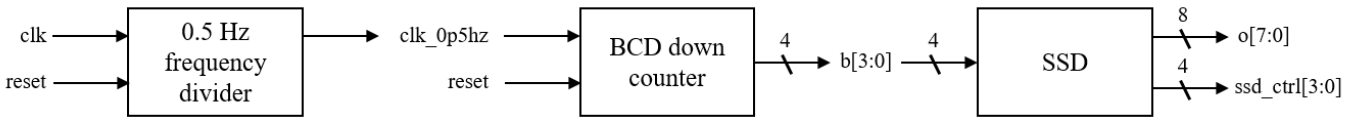
Input: clk, reset.
Output: o[7:0], ssd_ctrl[3:0].
Wire: clk_0p5hz, b[3:0].

Design Implementation

Specification of the frequency divider:

使用一數 100M 次的 counter 來對原本 100M Hz 的 clk 進行除頻，使輸出為 0.5 Hz 的訊號 clk_0p5hz，即達成題目所求。

Block diagram:



Design flow:

在此題中，相較於 exp_2，頻率變為 0.5 Hz、counter 變為 BCD counter、SSD 顯示的數字也變為只能輸出 single digit。

針對 0.5 Hz，我們只需要將原本數 50M 次的 counter 改為 100M 即可。

而針對 BCD counter，我考慮 b_next 的值會隨著 b 值有所不同，若 b 現在為 0，則 b_next 就應該為 9 而不是 b - 1 = 15，在其他情況下，b_next 都為 b – 1。而一樣，初始值 b = 0，當正緣的 clk 訊號輸入時，就使 b = b_next。

最後考慮 single digit 的輸出問題，我讓 ssd_ctrl 永遠為 1110，即 SSD 上的數字永遠只有第四位會顯示，即可達成輸出 single digit 的要求。

I/O pins assignment

Input:

I/O	clk	reset
LOC	W5	V17

Output:

I/O	ssd_ ctrl[3]	ssd_ ctrl[2]	ssd_ ctrl[1]	ssd_ ctrl[0]
LOC	W4	V4	U4	U2

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

Discussion:

我認為 exp_3 是 exp_4 的前哨站，不管是在 single digit 的寫法上，還是 BCD counter 的要求，都為我們進行 exp_4 打下很好的基礎，也方便我們後續在進行 exp_4 時能夠更快上手。

IV. exp_4 (1 Hz BCD 2-digit up counter with SSD)

Design Specification

Input: clk, reset.
Output: q[7:0], ssd_ctrl[3:0].
Wire: clk_1hz, dig_0[3:0], dig_1[3:0], dig_choose[3:0], refresh, borrow.

Design Implementation

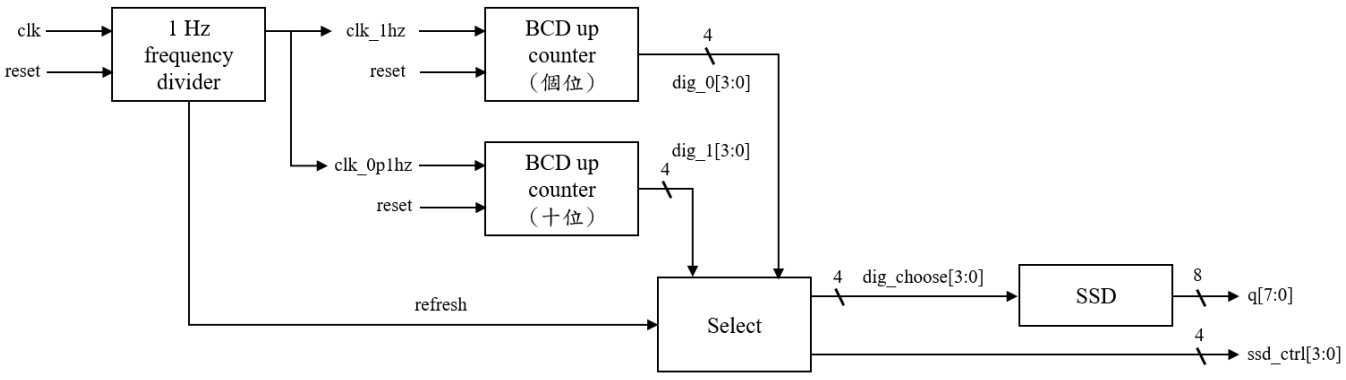
Specification of the frequency divider:

使用一數 50M 次的 counter 來對原本 100M Hz 的 clk 進行除頻，使輸出為 1 Hz 的訊號 clk_1hz，即達成題目所求。

此外，assign counter 的第 19 位元作為 refresh，以提供後續 select 的運作。

(若取太後面的位元，會導致模糊閃爍；若取太前面的位元，會導致視覺暫留不明顯)

Block diagram:



Design flow:

在此題中，相較於 exp_3，要解決 3 項問題，第一是 BCD down counter 改為 BCD up counter，第二是雙位元彼此間的進位問題，第三則是 select 的編寫。

針對第一項問題，我考慮 b_next 的值會隨著 b 值有所不同，若 b 現在為 9，則 b_next 就應該為 0 而不是 b + 1 = 10，在其他情況下，b_next 都為 b + 1。而一樣，初始值 b = 0，當正緣的 clk 訊號輸入時，就使 b = b_next。

而針對第二項問題，我使用了一個數 5 次的 counter，來將 clk_1hz 轉變成 clk_0p1hz 的訊號，而這個新產生的訊號就是給處理十位數的 BCD counter 的 clk 作為使用，使每數 10 秒就會讓十位數加一，達到進位的效果。

最後則是 select 的問題，這次的 select 相較於 Lab3-5 較為簡單，因為這次只需要選擇兩項即可。而同樣的，我使用 case 的寫法，當 refresh 為 0 時，引入 dig_choose = dig_0，且 ssd_ctrl = 1110，此時最右側的燈會更新為個位數字顯示；同樣的，當 refresh 為 1 時，引入 dig_choose = dig_1，且 ssd_ctrl = 1101，此時右側第二個燈會更新為十位數字顯示。而由於 refresh 的訊號變動夠快，視覺暫留的影響下，就會讓 SSD 看起來為兩位數字同時獨立顯現，達成題目的要求。

I/O pins assignment

Input:

I/O	clk	reset
LOC	W5	V17

Output:

I/O	ssd_ ctrl[3]	ssd_ ctrl[2]	ssd_ ctrl[1]	ssd_ ctrl[0]
LOC	W4	V4	U4	U2

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

Discussion:

我認為 exp_4 的題目是前面四次 Lab 題目的綜合體，用到了 Lab2 的 SSD 顯示、用到了 Lab3 的除頻器與 SSD 獨立顯示方式、更用到了這次 Lab 就已經有了的 counter 寫法。

而在過程中，困擾我最久的就是全新出現的進位問題，由於是第一次出現，讓我在思考如何進位時遇到了很大的挑戰，我首先設想了兩種進位方式，第一種方式是當個位數為 9 時去 enable 處理十位數的 BCD counter。而第二種正如上述，使用 0.1hz 的 clk 作為十位數的進位驅動，但後來經過考慮，我發現不管是在 reset 的設定，還是 clk 本身正緣驅動的影響，都會影響到最後 0.1 Hz 可能會與 1 Hz 處在不相同的 clk 上，而導致可能會出現延遲顯示的問題，故在往後的 Lab 中，我應該都會選擇以第一種作為進位的設定，來確保進位的獨立與穩定性。

Conclusion

在這次 Lab，最讓我印象深刻的題目就是 exp_4，如果我上面所說，exp_4 有點像是前面 4 週所有所學的綜合體題目，雖然在 counter 的編寫上沒有變得更難、SSD 的更新顯示也比 Lab3-5 簡單，但當同時要處理那麼大量不同的 block 去做功能上的串聯與設計，還是會感到複雜與困難。

而我最大的心得就是，未來在處理這樣會使用到許多 block 的邏輯電路設計時，一定要在一開始就設想好可能會發生的問題與解決方案，最重要的是，一定要有設計圖之類的去輔助思考，或者是在真正撰寫 module 時有個參考，這樣才能知道每個 module 需要多少 input、output，甚至在一開始的宣告就能把那些線路是 wire 是 reg 都區分開來，雖然這樣的前處理會花上不少時間，但卻能大大省去之後 debug 的迷茫與未知。

References

7 段顯示器的輸入接口查詢: (from Basys3 Reference Manual)
counter 編撰方式: (from 第四週上課講義)
7 段顯示器的更新方法: (from 第四週上課講義)