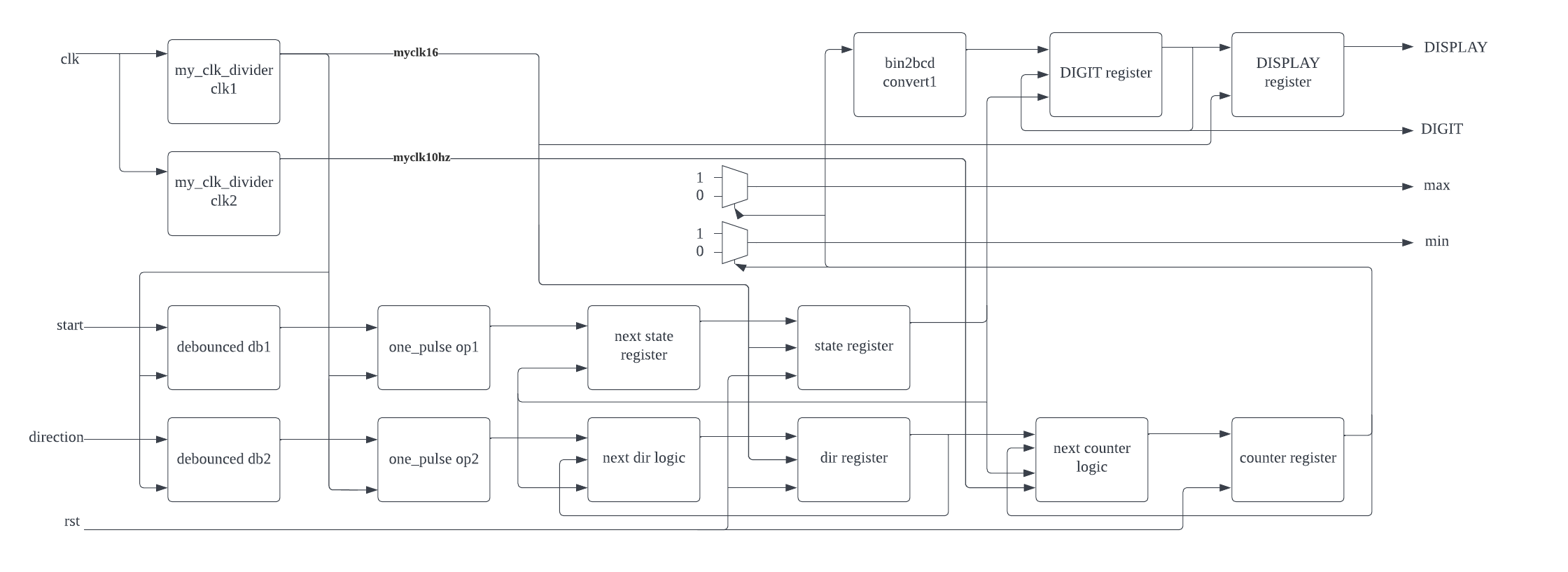
|  |  |
| --- | --- |
| **Lab 4** | |
| 學號: 110062131 | 姓名: 馬毓昇 |

1. **Lab Implementation**

Lab4\_1: 支援加減變換的7段顯示計數器。



Lab4\_1 segment:

1. binary counter: 一般的counter寫法，但有用dir判斷加減，最大加到999，最小減到0。為了符合規格要求，以slow clk（10hz）更新。

一張含有 文字 的圖片

自動產生的描述



2. 10 bits binary to 4\*4 bits BCD converter: 將binary的counter轉成4\*4 bits 的BCD。運用 ”double dabble” 演算法，實作上會跑過每一個binary bit，先判斷各BCD區，如果大於等於5的話就加3，最後把binary的bits左移一位進BCD區，直到binary bits全部被移進BCD區，不再判斷。







一張含有 文字 的圖片

自動產生的描述



3. 7 segment display: BCD0、BCD1、BCD2直接用剛剛轉好的BCD來接；BCD3則是用state與dir還有direction按下與否來判斷要呈現上、下、或一橫。4碼7段顯示的部分用跑馬燈的方式分別更新，同時更新value至對應的BCD值。後面依據各BCD來編碼右邊三碼7段顯示與左邊一碼的加減狀態顯示，而依眼睛視覺暫留選擇合適clk rate，所以用fast clk（100Mhz/2^16）來更新。

一張含有 文字 的圖片

自動產生的描述



一張含有 桌 的圖片

自動產生的描述



一張含有 桌 的圖片

自動產生的描述



4. dir: 用direction\_1pulse切換加減狀態，next\_dir = !dir可以實現0、1狀態切換，0是UP、1是DOWN。以fast clk更新使操作反應順暢。



一張含有 文字 的圖片

自動產生的描述



5. clock\_divider: 除數不局限於2的冪次的clock divider。以counter邏輯實作，當counter < DIVISOR/2輸出high，不然就輸出low。

一張含有 文字 的圖片

自動產生的描述

一張含有 文字 的圖片

自動產生的描述



6. debounce、one\_pulse: 都是用上課的模板。以fast clk更新使操作反應順暢。debounce需要連吃4個high才會輸出high，不然輸出low，以此消弭那些只有幾個clk的雜訊。one\_pulse偵測前一輸入是0現在是1的那個clk輸出脈衝，去除長按按鈕的誤差。

一張含有 文字 的圖片

自動產生的描述



一張含有 文字 的圖片

自動產生的描述



7. max、min: MUX判斷counter值來決定high or low。



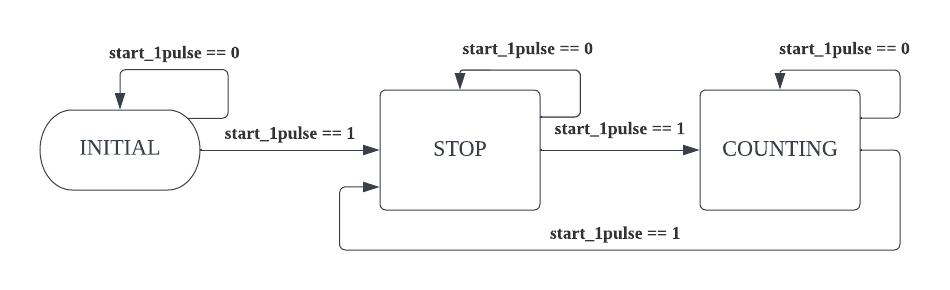
8. state: 吃到start\_1pulse就更新狀態，以fast clk更新使操作反應順暢。

一張含有 文字 的圖片

自動產生的描述



Lab4\_1 FSM:



1. INIT -> INIT: 沒按開始就不變。

2. INIT -> STOP: 按開始就進STOP。

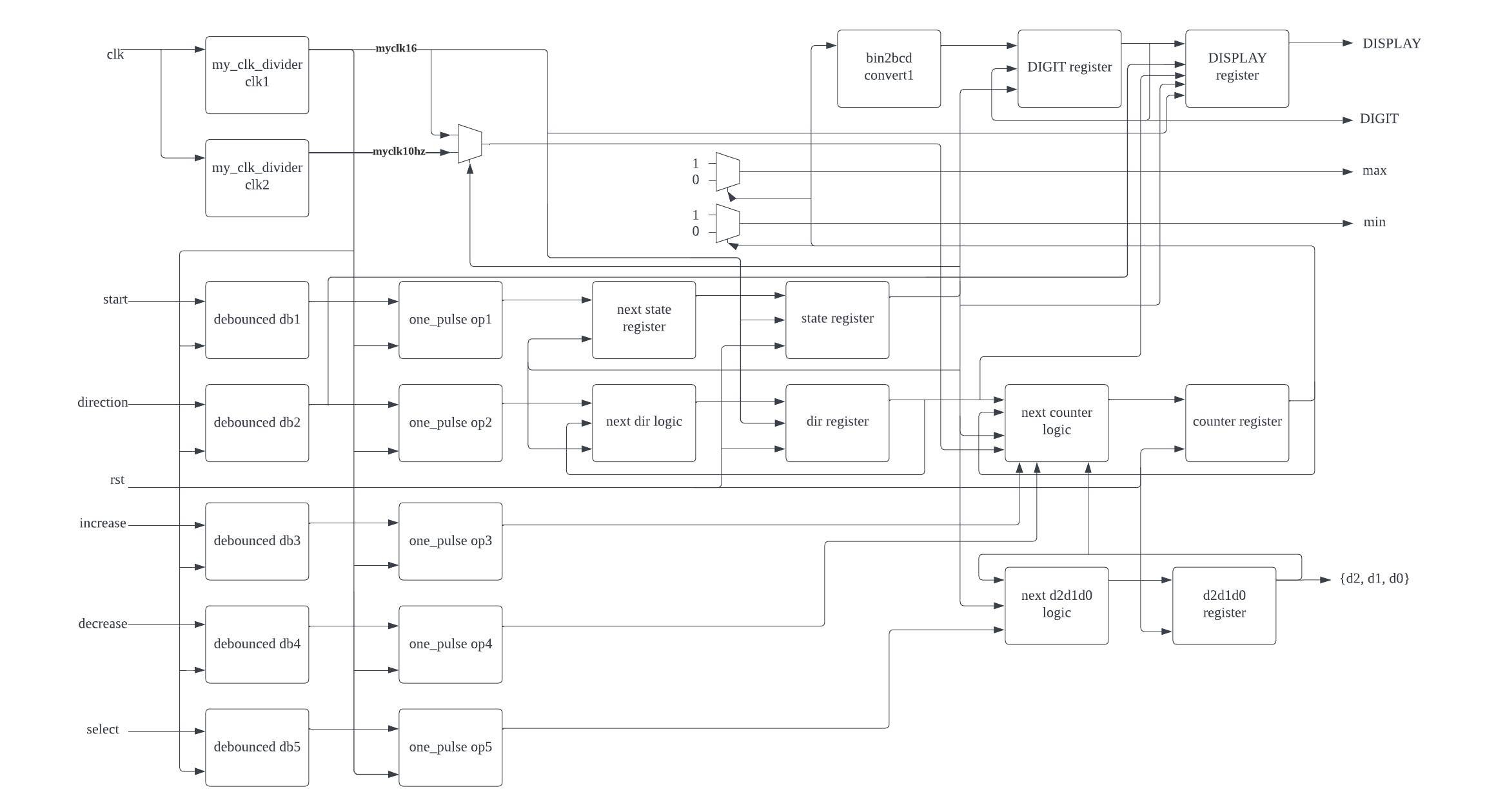
3. STOP -> STOP: 沒按開始就不變。

4. STOP -> CNT: 按開始就進CNT。

5. CNT -> CNT: 沒按開始就不變。

6. CNT -> STOP: 按開始就進STOP。

Lab4\_2: 支援選擇位數加減調整與加減變換的7段顯示計數器。



Lab4\_2 segment:

1. clkmux: 因為在STOP state繼續用10hz的slow clk的話，會跟不上fast clk的操作頻率而吃不到變化，所以用一個MUX來在兩個clk中切換。換成fast clk既可以吃到操作，反應上也會順暢。

一張含有 文字 的圖片

自動產生的描述



一張含有 文字 的圖片

自動產生的描述



2. increase、decrease: 先決定要調整哪位數，再用除10的冪次+取餘的方式判斷base10下的那位數值做變化。個位數9變0可以看做是counter + (-9)，其他位數同理。

一張含有 文字 的圖片

自動產生的描述



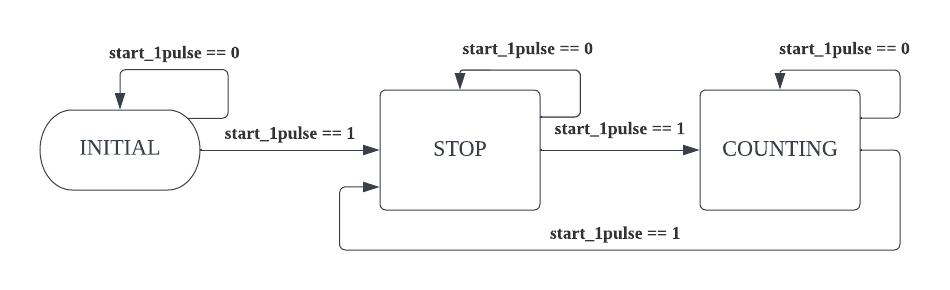
3. {d2, d1, d0}: 像跑馬燈一樣，每吃到一次select\_1pulse就換。fast clk更新使操作順暢。

一張含有 文字 的圖片

自動產生的描述



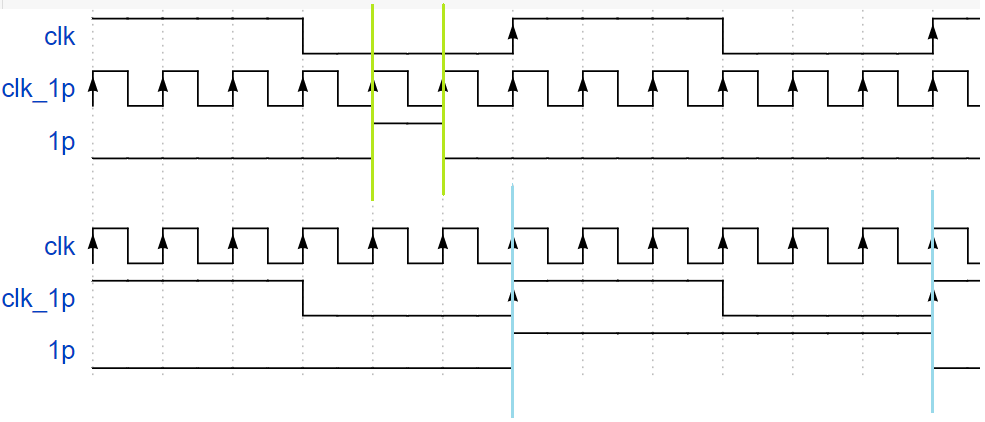
Lab4\_2 FSM:



1. **Questions and Discussions**

A: 按鈕內部是像彈簧一樣，壓放的時候總會有隨意的µs尺度彈跳訊號產生，又因為FPGA對脈衝的靈敏度有達到ns尺度，所以需要debounce來消除這些不希望出現的脈衝訊號。不需要，因為debounce負責的是穩定訊號，one-pulse是依據訊號產生脈衝。debounce的clk影響的是穩定訊號時對於原始訊號的靈敏度（clk越快，越接受細碎的訊號），one-pulse的clk影響的則是產生的脈衝長度（clk愈慢，一次脈衝的時間越久），與穩定訊號的輸入的無關，兩者間並不一定要使用同個clk，只是在合理的範圍內（不合理：用100Mhz跑debounce，這樣根本沒有debounce到；或是clk慢到one-pulse吃不到輸入訊號）都能選用各種clk rate。

B: 如果比較快的話，有可能會讓FSM沒反應到，於是就忽略了這次變化（如下圖綠框，沒有變化產生）；而比較慢的話，就有可能讓FSM因為一次脈衝而變化不只一次（如下圖藍框，其中每個posedge clk都會變化）。這都不是理想中的結果，應該要一次脈衝產生一次變化。



C: 因為有視覺暫留，當燈號的4bits依序變動得夠快，我們就會把他們當成是在同一時間一起變動的。

1. **Problem Encountered**

1. 想破頭都想不出來Question A第二題要怎麼證明我的答案，最後只能盡力用文字敘述表達我的想法。

2. lab中小卡的一點就是4\_2要上increase/decrease變化那邊，一開始沒注意到clk的問題還另外接了led測試是不是button接觸不良，測試時看到led瞬間閃爍就馬上給了我一個想法，會不會是button的操作（fast clk）相對於counter的變化（10 hz slow clk）太快，最後發現還真的是，一下子就改成功了。

1. **Suggestions**

我發現用Lucidchart畫圖很方便，說不定老師上課可以推薦給同學知道。(gapp, office365信箱可以有免費premium)