

EECS 2070 02 Digital Design Labs 2019	
Lab 4	
學號：107062314	姓名：陳柏均

0. 前言

這次的 lab4 主要要我們實作 counter, 而此次也多了七段顯示器的部分, 跟以往打的方式也有一定的差距, 一不小心 clk 設錯就會發生一些不可預期的顯示結果, 而除了第一題外, 其它題的條件都頗多, 也花了我很多時間在思考跟除錯上面; 而 lab4 讓我學到許多容易犯的錯誤以及新觀念, 相信以後一定會用的到。

1. 實作過程

(i)

第一題的題目要求我們做一個可以在 FPGA 板上顯示數字的功能, 而這題在給值的部分我便直接將 SW(switch)的值給它應屬於的那一個 bit, 然後再用 digit 去控制它的輪流顯示, 最後再使用 display 將其的數字顯示在七段顯示器上。

而我相信助教出這題的主要原因大概是能夠用在後面兩三題吧, 如果沒有第一題的話, 我將無法輕易理解這裡的道理。我認為第一題是一個附屬在二三題之中的基本卻又是極度重要的一個環節。

(ii)

看到第二題的第一眼, 我覺得第二題突然變得很難而且條件十分多, 很怕一步沒做好便會造成一個很難更正的錯誤; 第二題是我花最久時間去除錯的一題, 大概有其它題加起來的時間, 主要是因為 clk 值的除頻不知要用多少比較佳以及 debounce 跟 one_pulse 的新觀念, 這兩個我理解了一段時間, 最後看了講義並經過自己的思考過後終於知道這兩個東西存在的意義以及必要性。

這題在顯示數字的部分我跟上一題的方法一樣, 不過由於這題需要讓數字往上或往下數, 我用了類似 DFF 的概念在 combinational block 中給值並且在 sequential block 的地方去更新, 而條件的判斷(en dir reset 等)我主要是在 combinational block 中寫, 讓數字在裡面進行更新和給下一個狀態值。

Lab4_25 中我認為最具有挑戰性的部分就是 record 這一功能, 我後來是在當 record 的 onepulse 為一時, 則右邊的兩位指定給左邊的兩個 bits, 而若非如此, 則將它將無法維持原本的狀態。

```
begin
    if(record_pul==1'b1) begin BCD2<=BCD0; BCD3<=BCD1; end
    else begin BCD2<=BCD2; BCD3<=BCD3; end
end
```

(註:record 經過 onepulse 之後再去判斷現在 4 digits 中左邊兩位要給多少)

(iii)

第三題是我認為全部題目之中最有趣的一題，因為它的主要功能是去做設定值以及進行倒數的功能，我認為他有趣的主要原因是它感覺跟生活比較相近，而我今天竟然可以用 verilog 去呈現一個倒數計時器，我認為這十分激起我的興趣。

做倒數計時器的方式我跟上一題差不多，不過這一題並沒有往上加，只有往下數，一開始當我看到題目要求的時候，我第一個想到的是，我該如何做出兩個 mode 的狀態，後來經過了思考，才發現其實兩個 mode 也只是在 combinational block 中多一個判斷去決定目前要顯示哪個數字而已。

與上題不同，此題主要是想模擬真正的倒數計時器吧，我認為主要的差別是這題是 60 進位的概念，畢竟我們現在大部分人所使用的時間單位都是以 60 進位為主。不像前一題是十進位，這題需要多許多判斷式來判斷各種狀況。

起初我本打算用%(取餘數)去解決這一題，用這樣達到 60 進位的功能，但後來卻發現其實非我原本想像中那麼簡單，因此我最後還是決定使用比較直覺的方法去做，在特殊狀況的時候去給特定值，如此一來雖說好像要思考的比較多，但我認為至少在進行除錯的時候比較容易改也比較好看是出是哪種狀況出了問題。

這題如果單看左邊或右邊的兩個 bits 主要兩大種特殊狀態，一種是在 setting mode 的時候，而另一種便是在 counting mode 的時候，而之中主要的狀況又可以分為很多種，我下面主要針對秒數的部分進行分析，而分鐘的部分則是大同小異的。

Setting mode(sec_plus)

59:59 → 00:00

(註:分與秒都為 59 時,下一個狀態兩個都歸零)

09:59 → 10:00

(註:分不為 59 但個位為 9 且秒的部分為 59,則秒和分的個位歸零後分的十位數加一)

08:59 → 09:00

(註:分不為 59 個位也非 9, 則秒數地方歸零, 分個位加一, 其十位不變)

08:58 → 08:59

(註:秒的部分非 59, 則秒的個位加一, 秒的十位數以及分的地方皆不變)

Counting mode

00:00 → 00:00

(註:秒與分皆為 0, 則不動)

10:00 → 09:59

(註:秒為 0 且分的個位數為 0 但十位數不為 0, 則秒變為 59, 分的十位減一, 個位變為 9)

05:00 → 04:59

(註:秒為 0 且但分的個位不為 0,則秒變為 59, 分的個位數減一,十位數不變)

05:01 → 05:00

(註:秒不為 0, 則分不變秒減一)

(iv)

最後一題是 bonus, 我幾乎到最後一刻才將它打出來, 以前我們為了讓 FPGA 板上的 clk 做更動, 通常都是使用 clock_divider 這個方便的用具, 但以往卻都幾乎只是做 2 的 n 次方的除頻, 但這題卻要做一秒這樣的一個 clock cycle。

很明顯的因為 FPGA 的 clock 是 100MHz, 我們無法用以前 clock_divider 的方式去做, 但經過深度的思考過後, 其實不難發現, 只要 clk 數到 499999999 後將依訊號拉起來或拉下去(變為 1 或 0, 只要與原來訊號的相反), 這樣就可以呈現一個剛好 1 秒的新 clk。

2. 學到的東西與遇到的困難

(i) 7 segments 以及 4 digits 的顯示方式

我這次 lab 主要學到的一個東西便是七段顯示器的表示方法, 一開始我看了一段時間不懂它的表示方法, 而後來我仔細地再去看了一下上課關於 seven segments 的說明並經過自己深入想後才了解其中的道理。在還沒學這個以前, 一直以為像這種顯示數字的都是獨立且不相干的; 後來才發現在這個板子上它的 7 segments 是一起用的。

而四個 digits 則是用來選目前要顯示哪一位數字然後分別顯示, 這是我原本怎麼想都想不到的方式。

```
begin
  case(digit)
    4'b1110: begin val=BCD1; digit=4'b1101; end
    4'b1101: begin val=BCD2; digit=4'b1011; end
    4'b1011: begin val=BCD3; digit=4'b0111; end
    4'b0111: begin val=BCD0; digit=4'b1110; end
    default: begin val=BCD0; digit=4'b1110; end
  endcase
end
```

(註解:看 digit 去決定現在要顯示哪一位數字)



(註:純顯示示意圖 ,代表一次其實只顯示一位, 然後不斷輪流顯示)

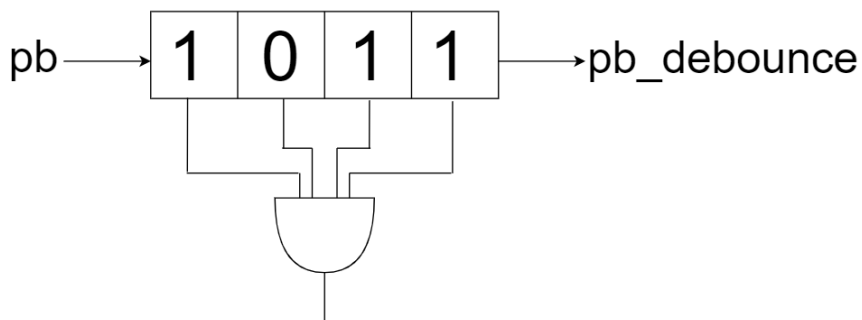
我認為使用這一個方法來進行顯示有它一定的優點, 那就是四個顯示的規則可以共用, 也就是說像這題每一個 digit 都可以顯示 0~9 並且是共用的。另外這裡還讓我學到一個新東西, 就是在 seven segments 及 four digits 的表示這邊的表示方式, 跟我最初想的也不一樣。

直覺會覺得亮就是 1, 暗就是 0, 但這裡卻是相反的, 我認為萬分特別, 因為打了這幾題, 我學到了這一個部分。不然原本的我照直覺走會是相異的結果。

(ii) Debounce 與 One-pulse

Lab4 中要用運到許多 debounce 跟 one-pulse 來改變原本真正的訊號, 以往都沒學過這方面的知識, 也沒想到會有這類的功能以達到自己所要。以我的理解, debounce 大概的功能便是進去的訊號可以更加穩定且清楚, 因為是接上 FPGA 板上的按鍵部分還有一些感應上的問題, debounce 其實就顯得很重要, 可以幫助了解進來的訊號究竟想表達哪一個數字, 是 1 或 0。

沒經過 debounce 前的資料可能會稍顯雜亂, 有階段性訊號不明的狀況, 而此時只要將訊號接過 debounce 便可以解決這個問題, 十分方便且具一定的必要性。



(註:必須至少一小段時間是 1, 其值才是 1, 避免因雜訊而誤判)

而另外 one-pulse 的主要目的是讓進來的訊號只顯示 1 個 clk(傳進來的 clock), 以方便在處理訊號的時候能達到自己想要的結果。 題目中按鍵有許多要求但其中最重要的大概就是按鍵按下去後, 就算繼續下壓, 顯示之數字仍然是原本的, 也就是說它只在手壓下去的那時去判斷並改變狀態, 但之後的訊號經過 one-pulse 之後便都是 0 所以不會有反應。

我自己在寫 lab4_2 遇到最嚴重同時也除錯最久的就是這一個部分, 在做 record 這一個功能時, 剛開始我下壓按鈕, 若一直持續壓著不放, 則它完全會跟著右邊的數字跑, 也就是說它有點像成為了另一個 counter 的概念, 題目有特別去講這一個部分, 不然我大概會以為我做的並沒有錯。

而後來發現原來是我的 one-pulse 的 code 一個小地方不小心寫錯而導致了我整體的嚴重錯誤。由這裡便可知道 one-pulse 的重要性以及它所帶給我們寫程式者一個方便進行我們要之操作的方法。



(註:如圖我們可以清楚看出 one-pulse 讓訊號變為只有一個 clock cycle)

(iii)clk 訊號的同步

在打這一次的 lab 時, clk 的選擇我認為也是我碰到的一個大問題, 這樣會使我的按鍵在按的時候好像不是很靈敏, 實際上是因為 clk 的問題, 而我也是過很多種 clk, 有的會有反應但有的幾乎沒反應, 而這一個部分我感到非常挫折剛開始也不知該如何是好。

後來我決定讓 clk 有點像是強制同步, 如第三題有兩個不同的 mode, 我便哪裡應用哪個 clk 就給哪一個。也就是說, 如果 mode 是 1 的話, 我便跟 counting mode 做同步, 而當 mode 是 0 的時候, 則是與 FSM 同步, 經過這樣的修改之後我終於解決了我的問題, 按鍵在按的時候也可以正常運作並發揮功用。

```
assign stop=stp;  
assign clk_=(mode == 1) ? clk25 : clk13 ;
```

```
always@(posedge clk_, posedge reset)  
begin
```

(註:使用一個 clk_ 去強制判斷現在 clk 應該要是多少以達到同步)

(iv)歸零的問題

在第三題中有要求當為 setting mode 時應該歸零，一開始不知道為什麼需要強調這個，後來才明瞭它其實並沒有我原本頭腦思考的那麼輕易實作，也難怪題目需要在這個地方做提醒。因為在 mode 為零時，不能單單考慮給它零值，由於還需要做設定的動作，它不能永遠是 0。

對此我的解決方法是若 mode 此刻由 1 變為 0，我就將 counter 歸零，同時我在這一次的 clk cycle 去記下這次為第一次；而當下一次再次判斷到 mode 為 0 時，我就去看上一次的 mode 是不是 1，若是 1 則代表上次在 mode 1 的狀態，則歸零，反之則讓他維持原來應該要給的數字。

```
BCD0<=(mode==1'b0&&premode==1'b1)?0:nextBCD0;  
BCD1<=(mode==1'b0&&premode==1'b1)?0:nextBCD1;  
BCD2<=(mode==1'b0&&premode==1'b1)?0:nextBCD2;  
BCD3<=(mode==1'b0&&premode==1'b1)?0:nextBCD3;
```

(註:每次給值前再多判斷上一次的 mode 狀態)

(v)接在板子上其它地方

這次我還學到了一個 debug 的小技巧，當我去助教時間請助教幫我解決問題時，有次助教就叫我將訊號去接上燈看看，看它的反應如何。我之前都沒想過有這樣的一個方法。

我覺得這樣的主要目的除了檢查是否是板子出了問題，還有能從簡單的顯示上(led 燈)先去進行思考，進而在回到原本數字上的問題。這次我學到這一個小小的方法覺得十分有趣，同時也讓我在以後多少可以用到。

3. 想對老師或助教說的話

Lab4 的完成給了我時分大的成就感，但是不得不說這次我真的覺得很難，當各種問題夾雜在一起時，有時會讓我不知從何開始著手進行除錯。老實說，這次的挫折感真的是不小的，但也無法反對這幫助我學習新知識，使自己未來不再犯同樣的過錯。

一次次 lab 不就是如此：雖說是辛苦，但學到的仍是很多。這次的有趣程度我認為又增加了。尤其是七段顯示器的部分我感到十分新鮮，而且在生活中處處可以看到它們的蹤影。

打完 lab4，我了解了這類顯示器的運作方式，也讓我每次看到時鐘、紅綠燈倒數器和店家的一些號碼燈，都可以想到這個。以前都覺得這沒什麼好稀奇的，但學了之後才發現裡面有很多我從未發掘的事物。除了七段顯示器，還有各種類似的顯示器；像我現在打完 lab4 之後，深深體會到表面看起來簡單的東西，實際上可能非如此。

在這裡我就聯想到中文的這類顯示器真的做的很好，能顯示那麼多種中文字，更何況中文字的變化真的多到不計其數，我很佩服發明這個的人。人類為了生活發明了新事物帶給自己方便，很多發明其實去細想都是很偉大的。這次的 lab4 我收穫良多，同時也做出一個我自己認為十分好玩且感覺很實用的東西，經過無數困難與挑戰，我終於得以解決重重難題並完成這次 lab。

今天來講一個雖然聽過很多次還是覺得很好笑的笑話吧：

在一個鄉下的酒吧裡，有兩個牧師在一起喝酒，不久因喝多了而十分興奮，其中一個牧師便建議騎車去兜風。

騎車的牧師就來個放開雙手騎車的特技，好死不死被巡警撞見，便被攔了下來：「酒後駕車，特技行駛，再加上兩人沒戴安全帽，準罰死你們！」

牧師見狀忙道：「警察大人，剛剛我沒有特技行駛，而是上帝在操縱著機車呢！」

警察一聽，又拿出了罰單邊寫邊說：「三貼再加上一人沒戴安全帽，簽名吧！」