**CS 2104-01 Hardware Lab**

**Lab 6 April 7, 2015**

***Catch the Stars***

**Lab report** (upload to iLMS before 4/14 3:30p.m.)

Your report should include the following:

* description and explanation of your work (including answers to all questions in the pre-lab)

這次的Lab我在top下，使用助教提供的 clk\_div module 取出我需要的 clock ，再用 Pre\_btn module 做 buttons 訊號的處理，其中包含助教提供的 debounced 和onepulse， Rand module 負責亂數決定星星位置，搭配 Check module 決定遊戲是否結束亦或要重新開始，並且輸出需要block的星星位置，最後修改之前的14 segment 的display module 成display\_star module，依照順序判斷：是否over、是否有星星、是否有block。

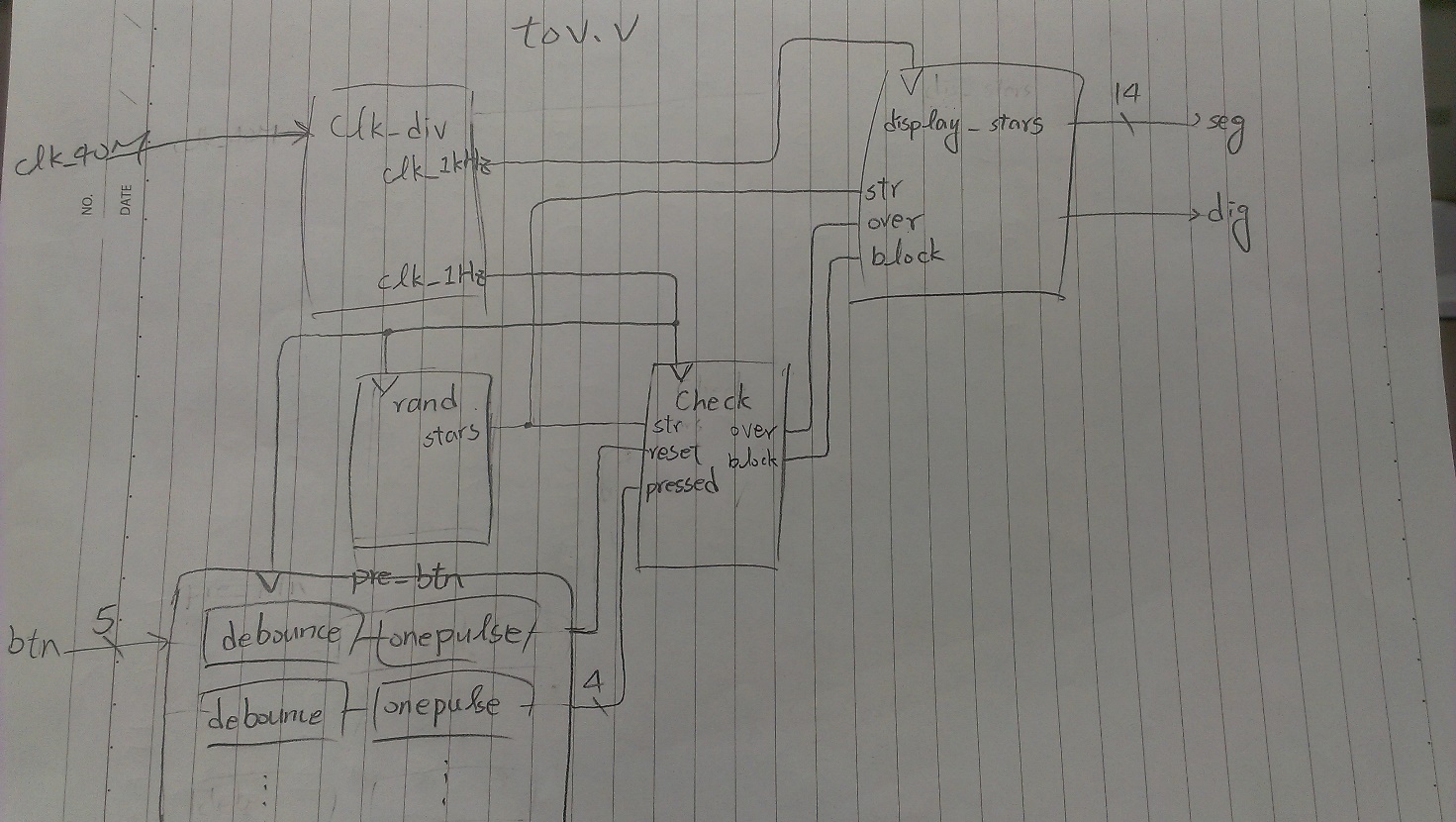
1. To make the game really interesting, we want to introduce some randomness into the game, therefore the stars should appear in any order at random. Search the internet and find out how to introduce randomness into your design. (You may first implement and test a simplified prototype assuming the stars always appear from right to left repeatedly, but if you just submit that simplified prototype you will get at most 80 points for this lab.)

網路資料參考：

<http://en.wikipedia.org/wiki/Linear_feedback_shift_register>

<http://stackoverflow.com/questions/14497877/how-to-implement-a-pseudo-hardware-random-number-generator>

1. Draw a block diagram for the complete design by re-using existing modules introduced in the class previously and defining any new module.



* discussion of any issue or problem worthy of note (yes, even mistakes that you made)

之前把Check的動作放在display的module裏頭，之後和同學討論才釐清，他們需要的clock不同，應該把它放到另一個module比較好。

computer所謂的亂數都還是有pattern存在，這次的lab亦是如此，只是循環的週期如何而已，雖然我們只用到一層LSFR，所以大概二十次左右為一個週期，但有看到有同學再多放一個counter增加pattern的複雜度，例如每一百個裡面只取第23、29、51次做星星位置的亂數。

* optional: any extra feature you added or any suggestion

沒有抓到星星時，會變成顯示”OVER”。

另外，跑random的長度越長，能做出的random pattern複雜度越高，因此將reg 增加為16 bits。