**Lab 3 110060027朱豐蔚**

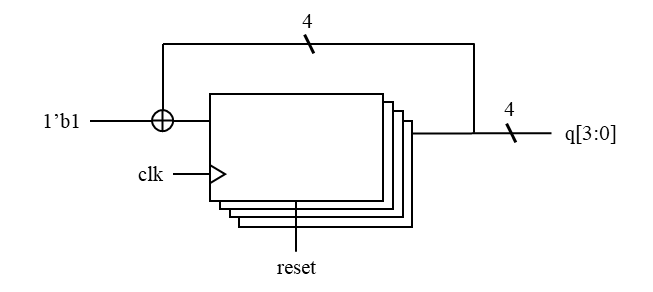
1. **prelab\_1(binary up counter)**

**Design Specification**



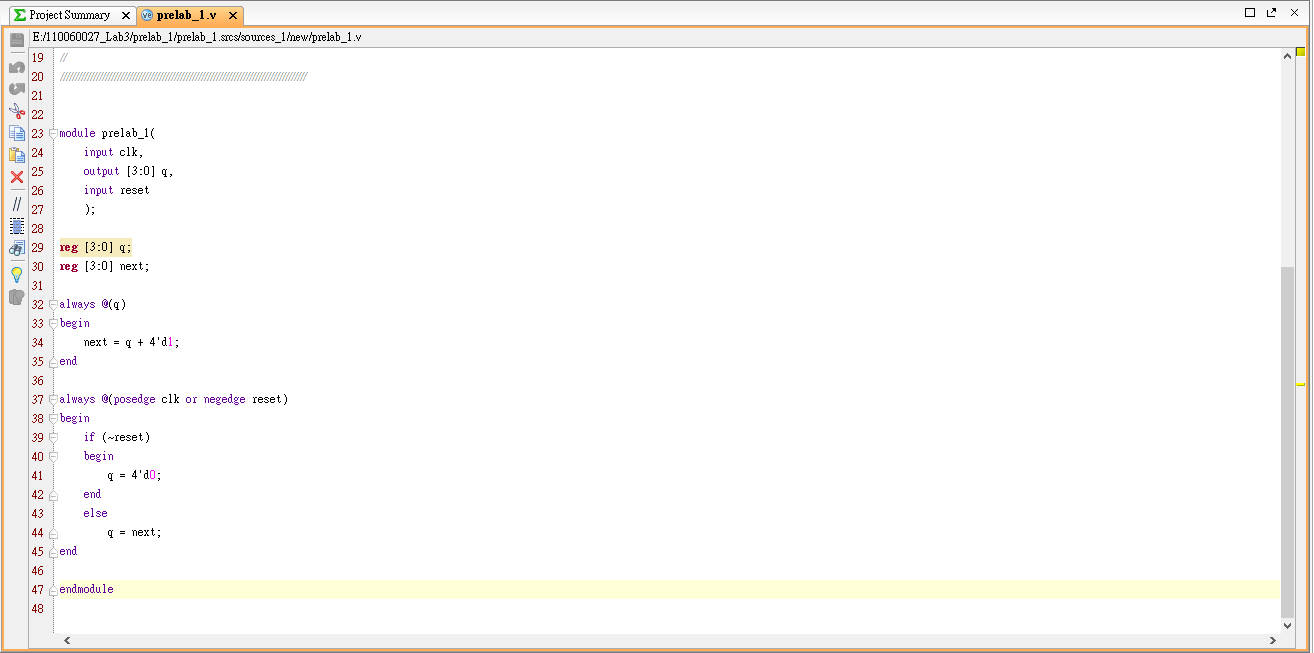
Input: clk, reset.

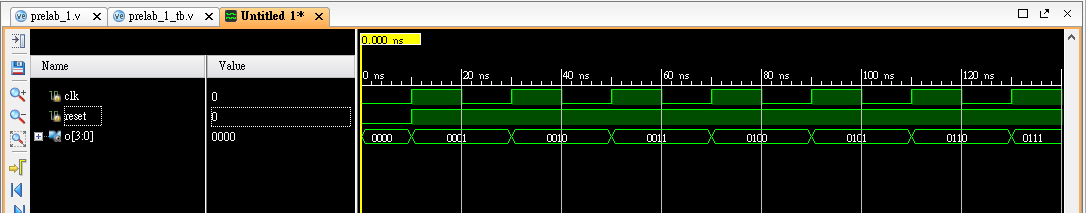
Output: q[3:0].



**Design Implementation**

Logic diagram:

 Code:

Testbench result:

1. **prelab\_2 (shift register)**

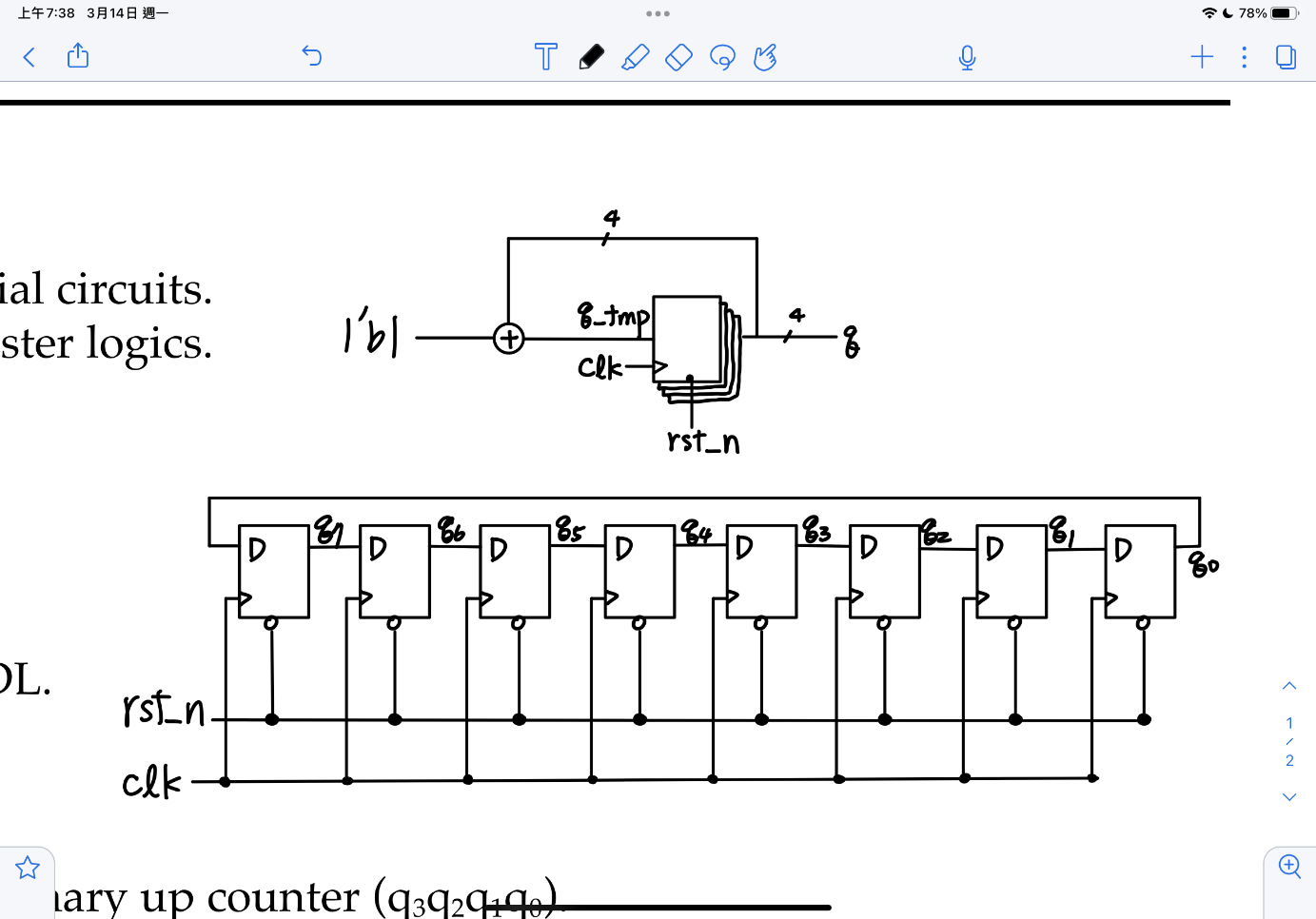
**Design Specification**

Input: clk, reset.

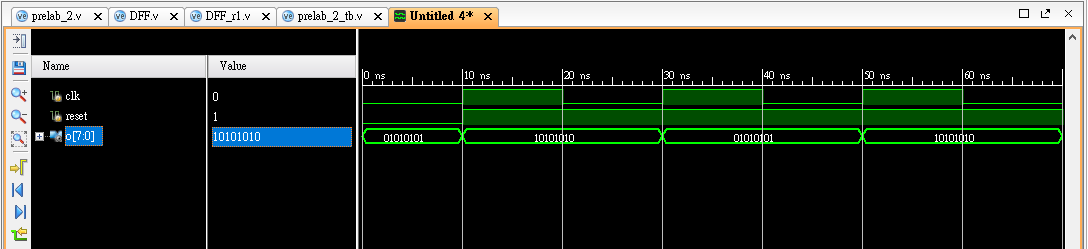
Output: q[7:0].

**Design Implementation**

Logic diagram

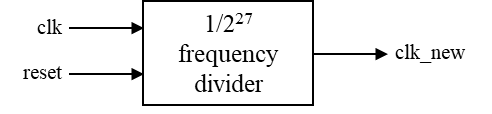


考慮到reset後的初始值問題，我將DFF分成兩類，一類是reset後存1的DFF，另一類則是reset後存0的DFF，使用兩種DFF交互設立，即可達成題目所要求的初始化後為01010101的8-bit ring shift register。

Testbench resul

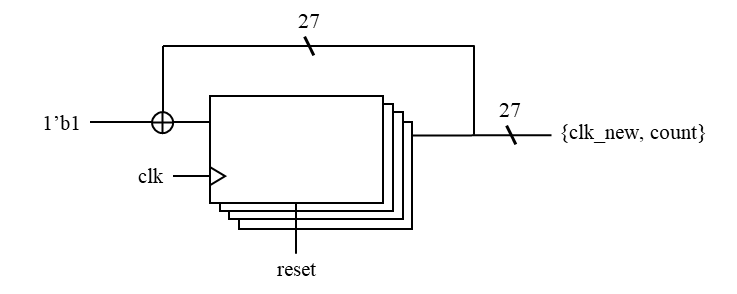
1. **exp\_1 (1/227 frequency divider)**

**Design Specification**



Input: clk, reset.

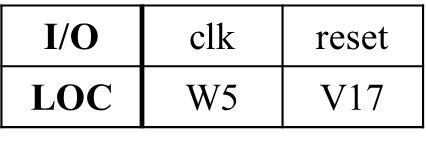
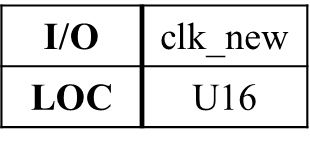
Output: clk\_new**.**

**Design Implementation**

Logic diagram:

I/O pins assignment

Input: Output:

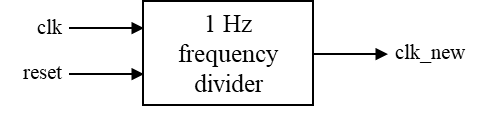


**Discussion:**

本實驗是設計一個除頻器，使得將初始時脈頻率(clk =100M Hz)除以227後，除到接近1Hz的頻率。我們設計一個27-bit的binary up counter，並將此27-bit的最高位數當作輸出，便可除到接近1Hz的頻率。

1. **exp\_2 (1 Hz frequency divider)**

**Design Specification**



Input: clk, reset.

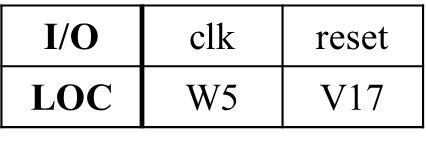
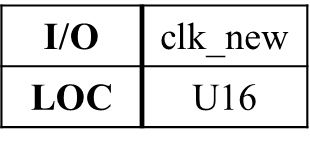
Output: clk\_new**.**

**Design Implementation**

本實驗我們要將FPGA板上100M Hz的頻率轉換成1 Hz頻率輸出的除頻器。由於題目要求製作一個50M的binary up counter，因此設計理念是每當counter數到50M時讓輸出值反轉一次，這樣的話，輸出在等於0和等於1的時間會一樣；又因為每0.5秒反轉一次，因此輸出值的頻率會等於1 Hz。

I/O pins assignment

Input: Output:



**Discussion:**

在最開始，我自己設想的1 Hz除頻器是使用100M的counter，但後來經過更深入的思考後，才考慮到反轉0跟1會讓時間延長為兩倍，因此在一開始就只需要50M的counter，我認為這個實驗對我熟悉除頻器有非常大的幫助。

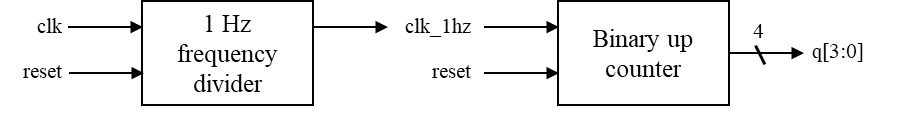
1. **exp\_3 (1 Hz binary up counter)**

**Design Specification**

Input: clk, reset.

Output: q[3:0]**.**

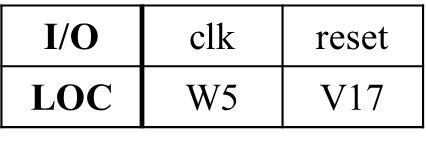
Wire: clk\_1hz.



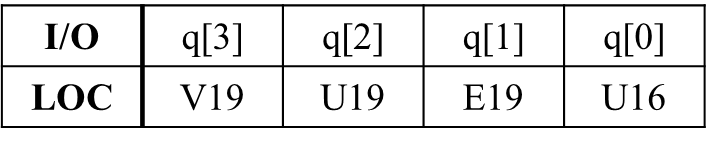
**Design Implementation**

I/O pins assignment

Input:



Output:

****

**Discussion:**

本實驗要設計一個clock為1 Hz的binary up counter。做法其實只要將Lab3-2做好的除頻器之輸出值(clk\_new)，接上Prelab3-1第一題counter的輸入(clk)，就完成實驗了。

這個小實驗應該是我們第一次真的需要引入多種module來進行完整的線路設計，難度也恰到好處，能讓我們馬上上手。

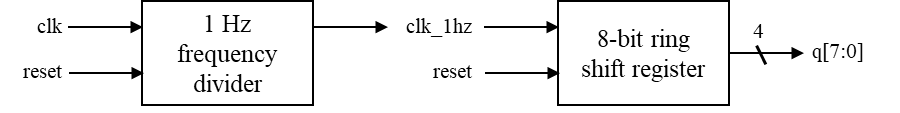
1. **exp\_4 (1 Hz 8-bit ring shift register)**

**Design Specification**

Input: clk, reset.

Output: q[7:0]**.**

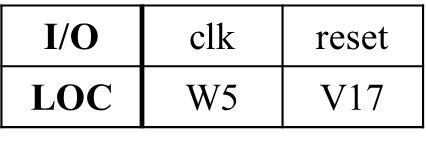
Wire: clk\_1hz.



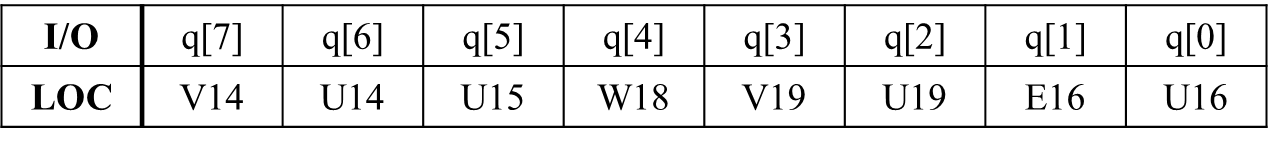
**Design Implementation**

I/O pins assignment

Input:



Output:



**Discussion:**

本實驗要設計一個clock為1Hz的ring shift register，做法與上題一樣將Lab3-2除頻器1Hz 的輸出(clk\_new)接上prelab-2的輸入(clk)之後,便可達成題目要求的結果。

在reset值為01010101的設計上我苦惱了很久，想出兩種方案，第一種就是我使用的設計，透過設立兩種不同reset值的DFF來交替排放，這也是我將在Lab3-5中所使用的。第二種是使用相同的DFF，而在初始化時填入各值，這種方案我也有私下嘗試過，但會比第一種方式更家複雜，會需要加入select來讓register擁有load的功能，於是我在此題中選擇了第一種設計思路。

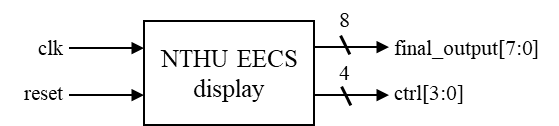
1. **exp\_5 (NTHU EECS 跑馬燈顯示器)**

**Design Specification**

Input: clk, reset.

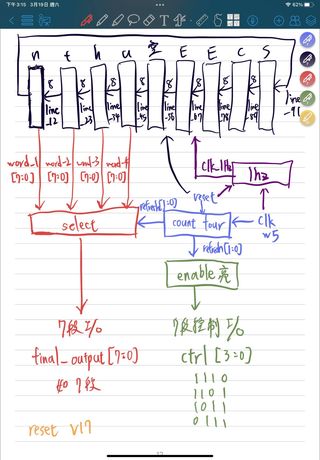
Output: final\_output[7:0], ctrl[3:0]**.**

Wire: clk\_1hz, clk\_10Khz, line\_12[7:0] , line\_23[7:0] , line\_34[7:0] , line\_45[7:0] , line\_56[7:0] , line\_67[7:0] , line\_78[7:0] , line\_89[7:0] , line\_91[7:0], refresh[1:0].



**Design Implementation**

Logic diagram:



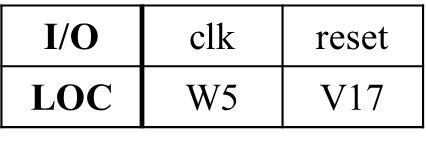
先使用不同的兩種降頻器，製造出1 Hz與10K Hz的訊號，前者給8-bit ring shift register使用，後者給計數器count 4使用。

在8-bit ring shift register中，我設立了九個8-bit DFFs，並針對不同初始值的DFF做調整，使整體reset後能維持NTHU EECS的順序。

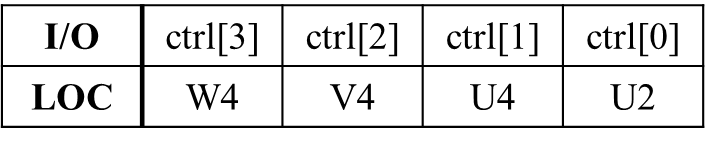
count 4為以10K Hz訊號持續數4次，將其轉成refresh[1:0]使用，並透過後續的select與enable模組，以case的寫法將final\_output與ctrl選擇出來，以進行後續SSD的顯示用途。

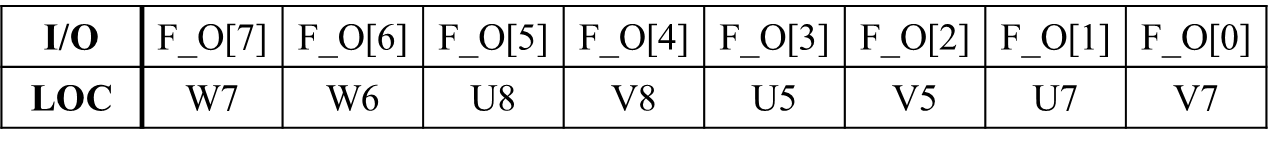
I/O pins assignment

Input:



Output:





**Discussion:**

由於此題我是在馬老師的時段進行Demo的，當時被告知EECS後不用有空白，所以在此題的設計中，我只使用了9個DFFs，若需加上空白，只需要再加設wire line\_910與DFF\_10­即可。

**======================================================================**

**Conclusion**

這次的Lab對我來說很富有挑戰性，特別是在Lab3-5的設計中，有許多知識都是要到第四週才會明瞭的，而我由於提前Demo，所以在看到題目的剎那，由於對SSD的更新不熟悉與未知，我腦袋是一片空白的，為此，我特地上網看影片、查詢Basys3 Reference Manual，並進行了好幾次的測試，最終才在還未上到禮拜四的課前提早完成Demo。

因此，現在回過頭去看當初的設計，才發現count 4可以使用降頻器中的[17:16]訊號來代替，select模組與enable模組也可以合併精簡化，甚至在shift register的撰寫中，也有更進一步提升的空間。

但很高興能夠靠自己這份願意去試、去學的心，完成了這次的Lab3，或許未來可能實驗會越來越難，但我永遠不會忘記這次富有成就感與開心的實驗經驗。

**References**

7段顯示器的輸入接口查詢:(from Basys3 Reference Manual)

7段顯示器的更新方法:

<https://www.youtube.com/watch?v=v2CM8RaEeQU&t=428s>