

Lab 3

110060027 朱豐蔚

I. prelab_1(binary up counter)

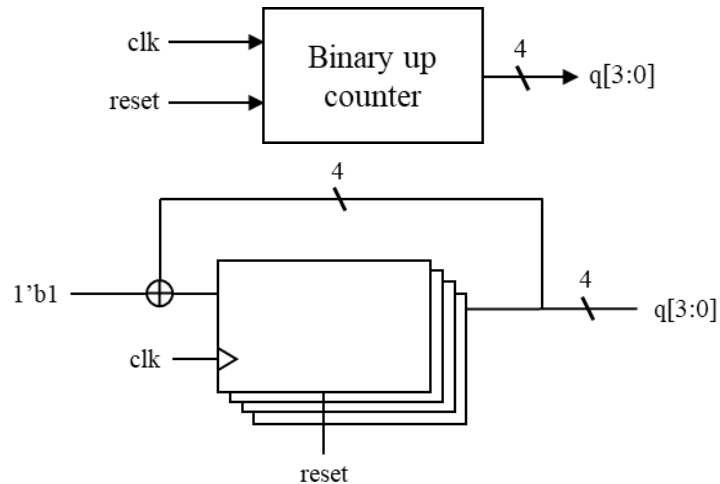
Design Specification

Input: clk, reset.

Output: q[3:0].

Design Implementation

Logic diagram:



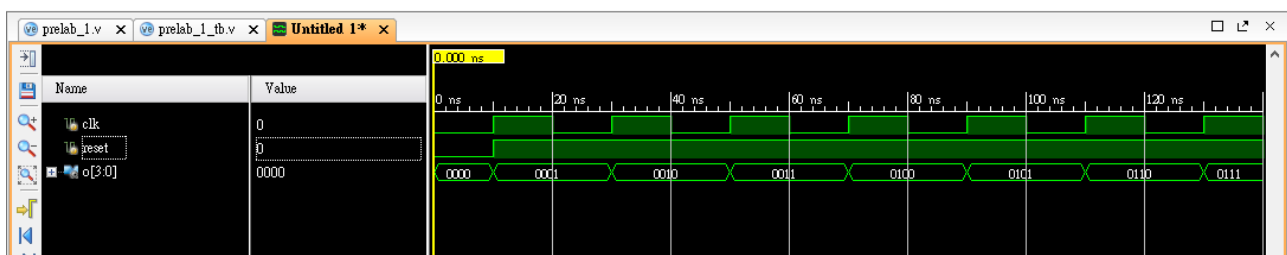
Code:

```

19 //
20 //
21 //
22 //
23 module prelab_1(
24     input clk,
25     output [3:0] q,
26     input reset
27 );
28 //
29 reg [3:0] q;
30 reg [3:0] next;
31 //
32 always @(q)
33 begin
34     next = q + 4'd1;
35 end
36 //
37 always @(posedge clk or negedge reset)
38 begin
39     if (~reset)
40     begin
41         q = 4'd0;
42     end
43     else
44         q = next;
45 end
46 //
47 endmodule
48 //

```

Testbench result:



II. prelab_2 (shift register)

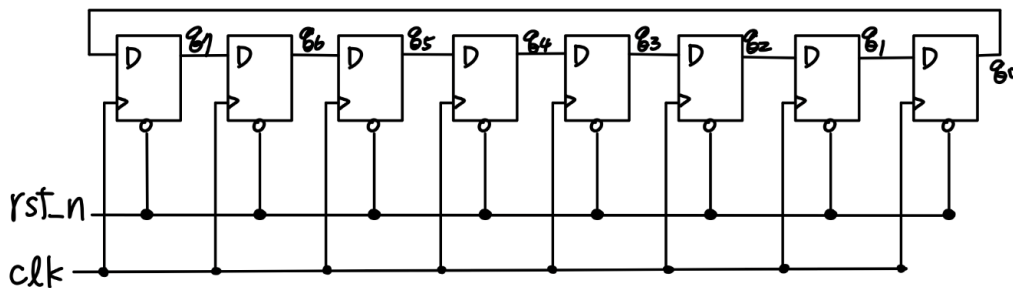
Design Specification

Input: clk, reset.

Output: q[7:0].

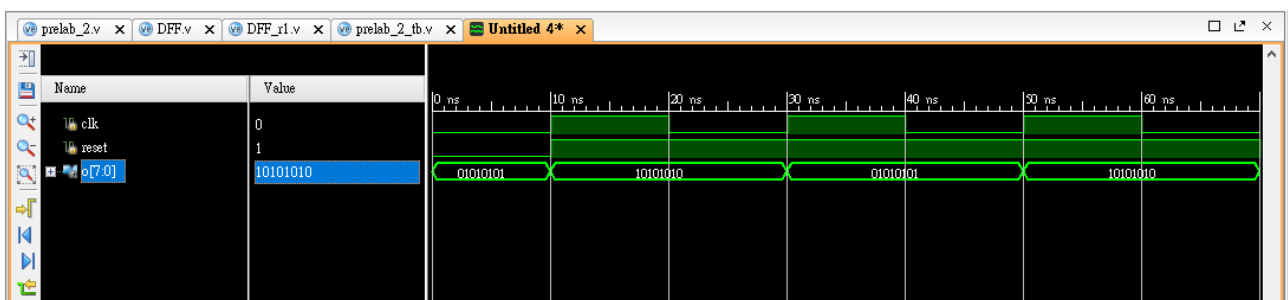
Design Implementation

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 result

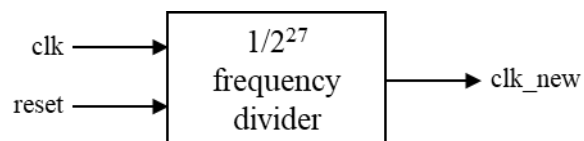


III. exp_1 ($1/2^{27}$ frequency divider)

Design Specification

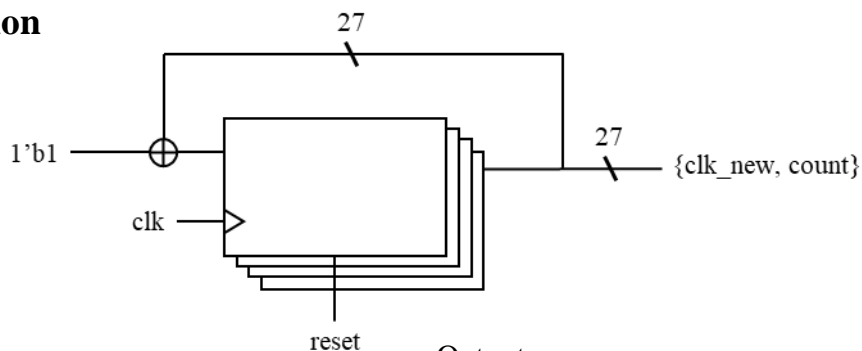
Input: clk, reset.

Output: clk_new.



Design Implementation

Logic diagram:



I/O pins assignment

Input:

I/O	clk	reset
LOC	W5	V17

Output:

I/O	clk_new
LOC	U16

Discussion:

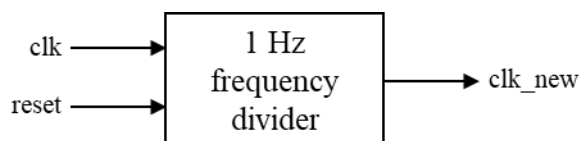
本實驗是設計一個除頻器，使得將初始時脈頻率($\text{clk} = 100\text{M Hz}$)除以 2^{27} 後，除到接近 1Hz 的頻率。我們設計一個 27-bit 的 binary up counter，並將此 27-bit 的最高位數當作輸出，便可除到接近 1Hz 的頻率。

IV. 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:

I/O	clk	reset
LOC	W5	V17

Output:

I/O	clk_new
LOC	U16

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。

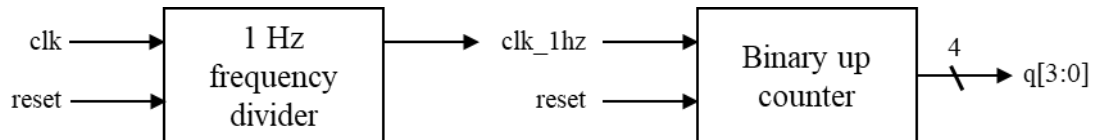
V. exp_3 (1 Hz binary up counter)

Design Specification

Input: clk, reset.

Output: q[3:0].

Wire: clk_1hz.



Design Implementation

I/O pins assignment

Input:

I/O	clk	reset
LOC	W5	V17

Output:

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

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。

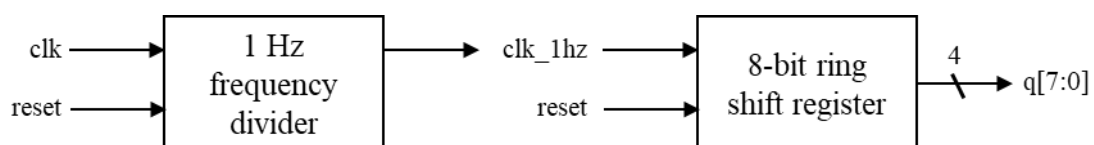
VI. exp_4 (1 Hz 8-bit ring shift register)

Design Specification

Input: clk, reset.

Output: q[7:0].

Wire: clk_1hz.



Design Implementation

I/O pins assignment

Input:

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	V14	U14	U15	W18	V19	U19	E16	U16

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，於是我在這題中選擇了第一種設計思路。

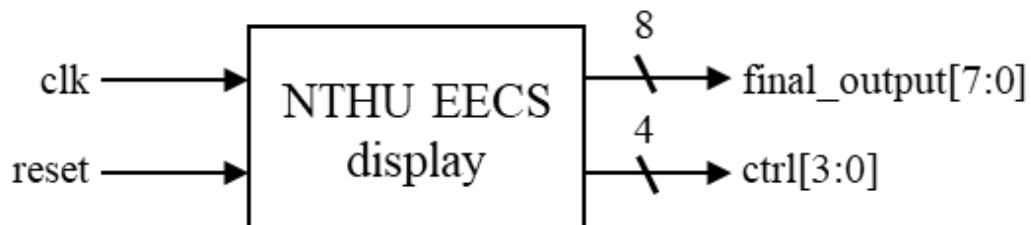
VII. exp_5 (NTHU EECS 跑馬燈顯示器)

Design Specification

Input: clk, reset.

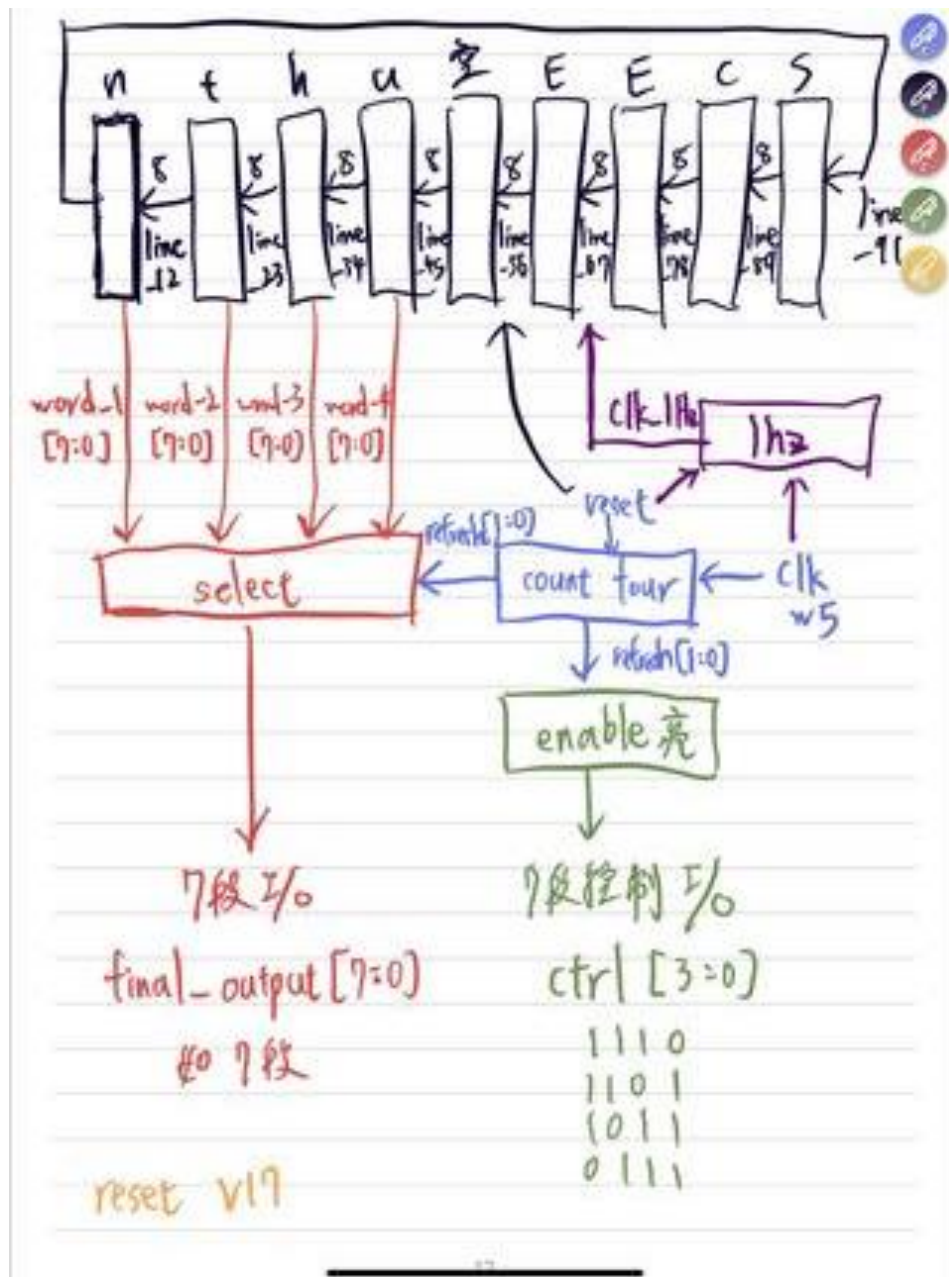
Output: final_output[7:0], ctrl[3:0].

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:

I/O	clk	reset
LOC	W5	V17

Output:

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

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

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>