

## Lab 4: Counters and Shift Registers II

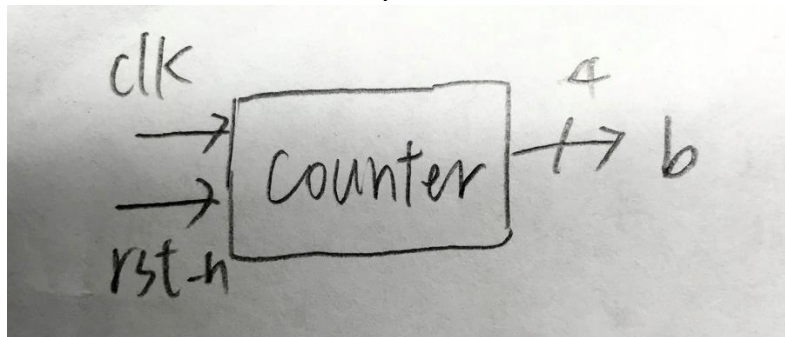
1. Construct a 4-bit synchronous binary up counter (b<sub>3</sub>b<sub>2</sub>b<sub>1</sub>b<sub>0</sub>) with the 1-Hz clock frequency from exp2 and use 4 LEDs for display.

I/O	$f_{crystal}$	b <sub>3</sub>	b <sub>2</sub>	b <sub>1</sub>	b <sub>0</sub>
Site	W5	V19	U19	E19	U16

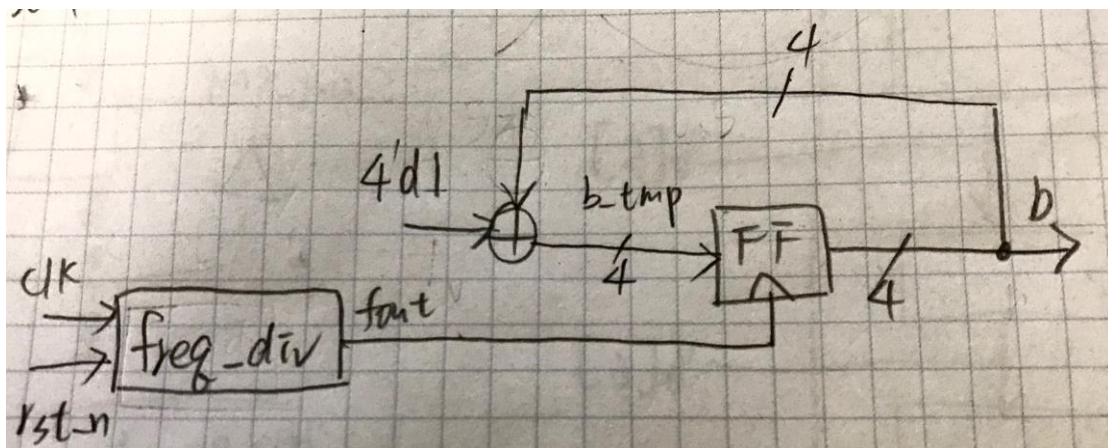
### Specification:

Input : clk, rst\_n

Output : b



### Implementation:



要做一個binary up counter with the 1-Hz clock，就先把之前做的  
freq\_div 1Hz clock的output fout接到一個flip-flops的clk，將FF的output  
設為4個bits b，將b+1設為b\_tmp，rst\_n是一個reset的功能，一開始b  
的值為0，當1Hz的clock開始數後，每1秒都會將b\_tmp的值送給b，b

就會是一個每秒都加1的output，這樣就完成binary up counter with the 1-Hz clock了。

將FPGA的W5接到input clk，就是把100Hz的clock接進來，再把4-bit b接到相對應的V19,U19,E19,U16這4個LED燈，1為亮，0為不亮，LED燈就會隨著output b的值而閃爍。

### 討論:

這個實驗沒什麼問題，將以前做過的實驗接起來就行了。

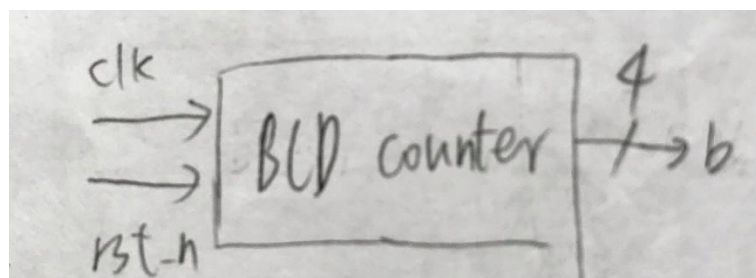
**2. Construct a single digit BCD up counter with the divided clock as the clock frequency and display on the seven-segment display.**

**2.1 Construct a BCD up counter.**

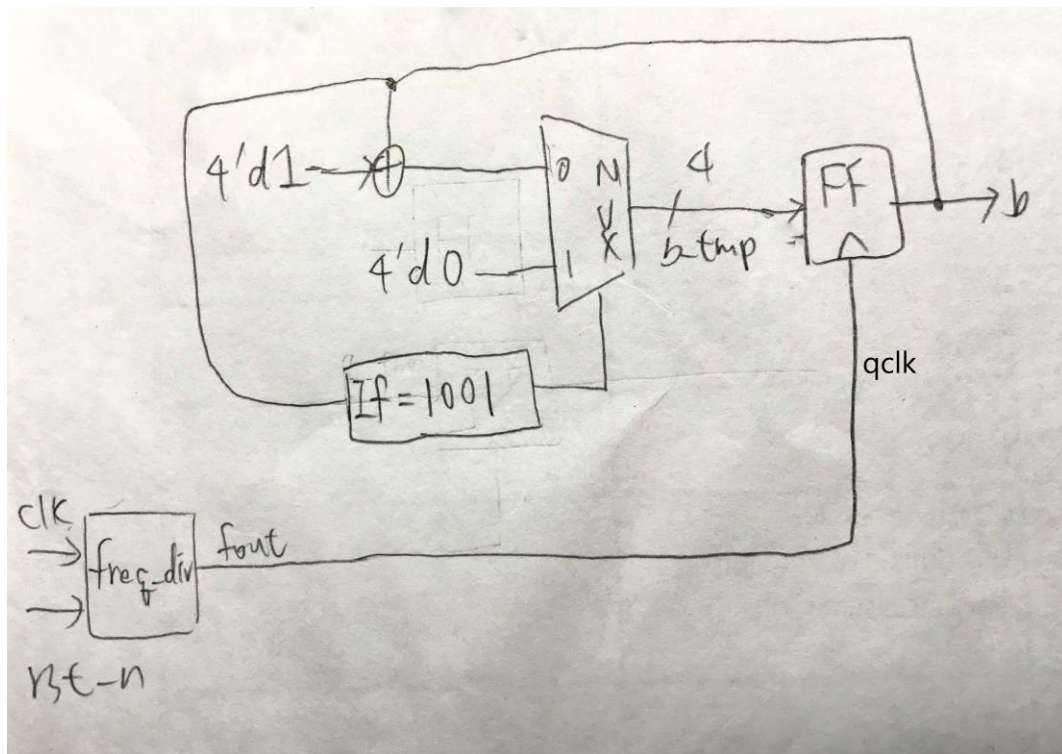
### Specification:

Input : clk,rst\_n

Output : b



### Implementation:



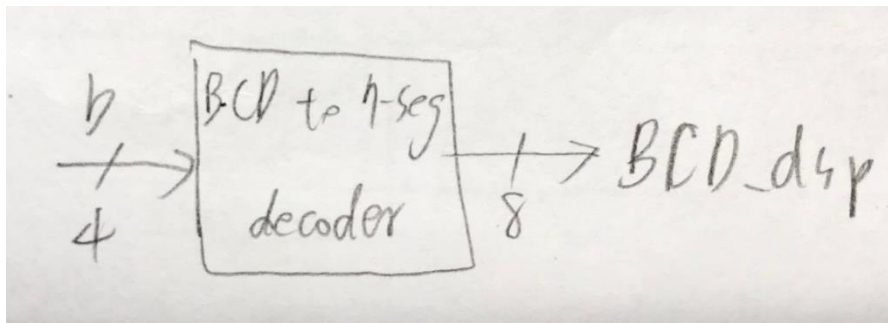
與上一題很像，把freq\_div 1Hz clock的output fout接到wire qclk給FF做為clock，將FF的output設為4個bits b，rst\_n是一個reset的功能，一開始b的值為0，當1Hz的clock開始數後，會有一個MUX判斷，當b不是4'b1001，會將b+1設給b\_tmp，過1秒再將b\_tmp的值送給b，當b一直向上數到4'b1001=9時，b\_tmp就會變成0，再將值給b，這樣b就會從0-1-2-....數到9後，變回0開始數了。

## 2.2 Construct a BCD-to-seven-segment display decoder.

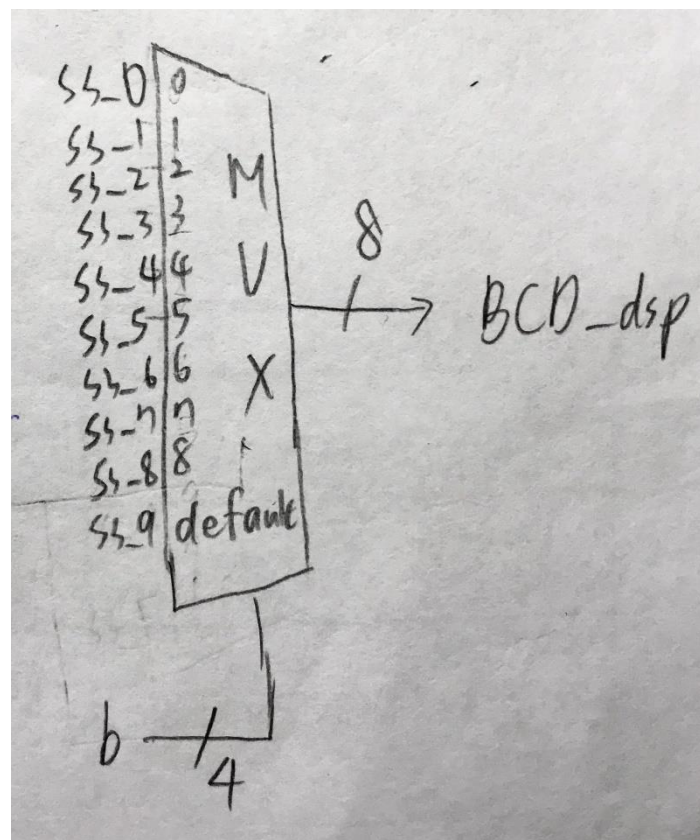
### Specification:

Input : b

Output : BCD\_dsp



### Implementation:



先將控制7-seg LED要顯示的數字的8個bits define給對應的SS\_0~9

```

`define SS_0 8'b00000011
`define SS_1 8'b10011111
`define SS_2 8'b00100101
`define SS_3 8'b00001101
`define SS_4 8'b10011001
`define SS_5 8'b01001001
`define SS_6 8'b01000001
`define SS_7 8'b00011111
`define SS_8 8'b00000001

```

```
`define SS_9 8'b00001001
```

將4-bit input b送入MUX裡判斷，當b為0，output BCD\_dsp為SS\_0，其

他以此類推，最後一個default是當b不是1~8，也就是b是9時，

BCD\_dsp設為SS\_9，BCD to 7seg display decoder就完成了。

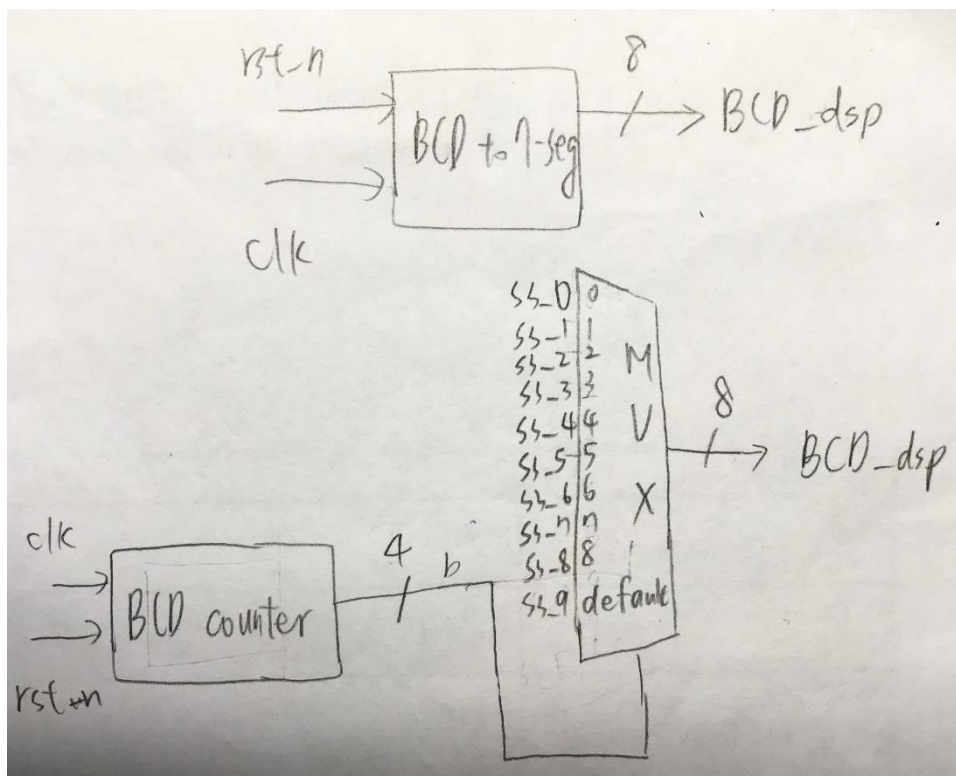
### 2.3 Combine the above two together.

#### Specification:

Input : clk, rst\_n

Output : BCD\_dsp

#### Implementation:



將上面兩個block接起來，BCD counter的output b接到BCD to 7seg

display decoder的input b，將FPGA的W5接到input clk，再把output 8-

bit BDC\_dsp接到相對應的7-seg ports，當clock開始跑，LED就會顯示  
0-1-2....-9-0-1...一直循環。

### 討論:

這個實驗只要照著題目，照著步驟，一部分的blocks設計好，再做另一部分的blocks，就能完成整個實驗了。

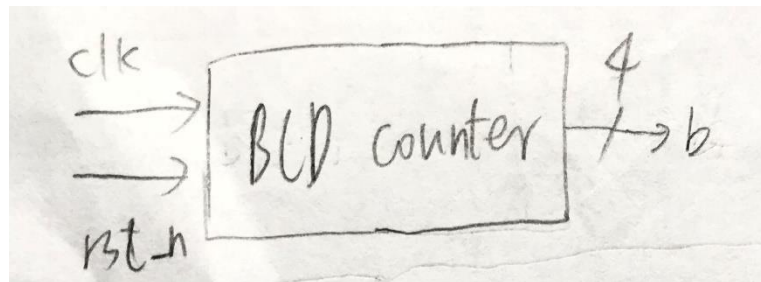
**3. Construct a single digit BCD down counter with the divided clock as the clock frequency and display on the seven-segment display.**

**3.1 Construct a BCD down counter.**

### Specification:

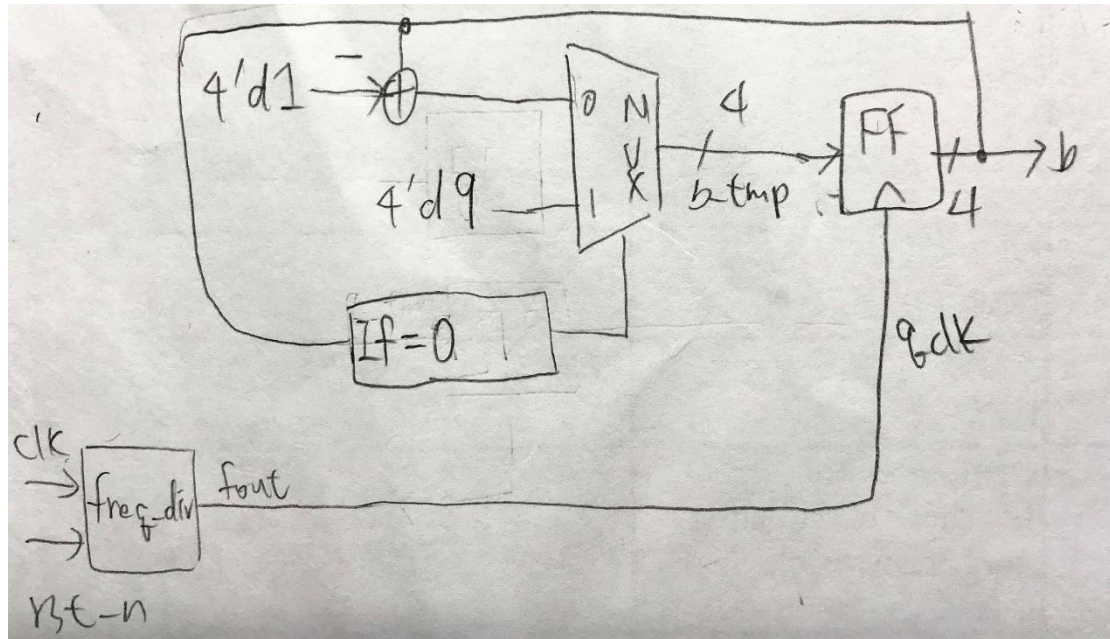
Input : clk,rst\_n

Output : b



### Implementation:





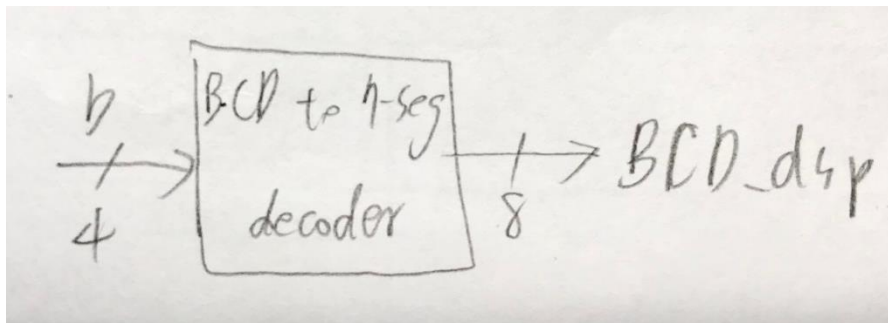
這與第二題觀念一樣，把freq\_div 1Hz clock的output fout接到wire qclk給FF做為clock，將FF的output設為4個bits b，rst\_n是一個reset的功能，一開始b的值變成設為9，當1Hz的clock開始數後，會有一個MUX判斷，當b不是0，會將b-1設給b\_tmp，過1秒再將b\_tmp的值送給b，當b一直向下數減到0時，b\_tmp就會變成9，再將值給b，這樣b就會從9-8-7-....數到0後，變回9開始數了。

### 3.2 Construct a BCD-to-seven-segment display decoder.

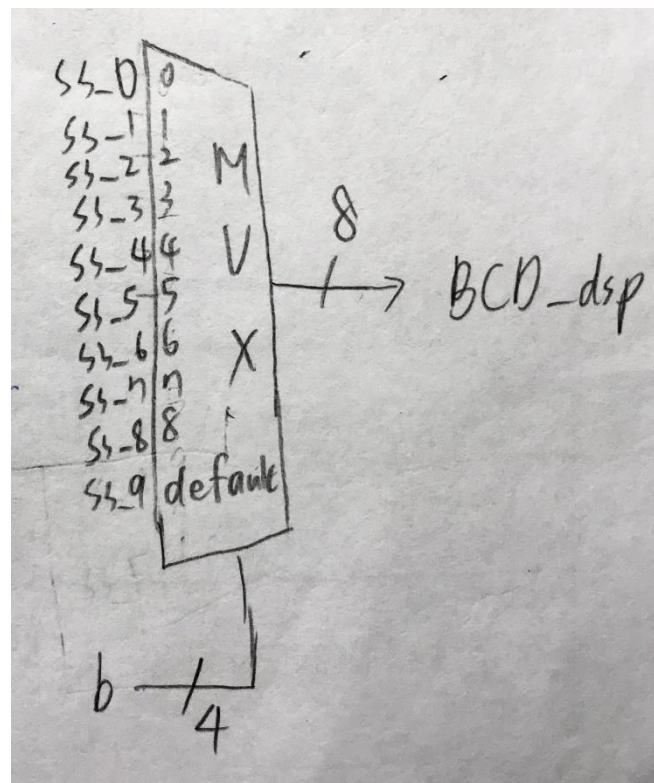
#### Specification:

Input : b

Output : BCD\_dsp



### Implementation:



這跟2.2的部分一模一樣，先將控制7-seg LED要顯示的數字的8個bits define給對應的SS\_0~9，將4-bit input b送入MUX裡判斷，當b為0，output BCD\_dsp為SS\_0，其他以此類推，最後一個default是當b不是1~8，也就是b是9時，BCD\_dsp設為SS\_9，BCD to 7seg display decoder就完成了。



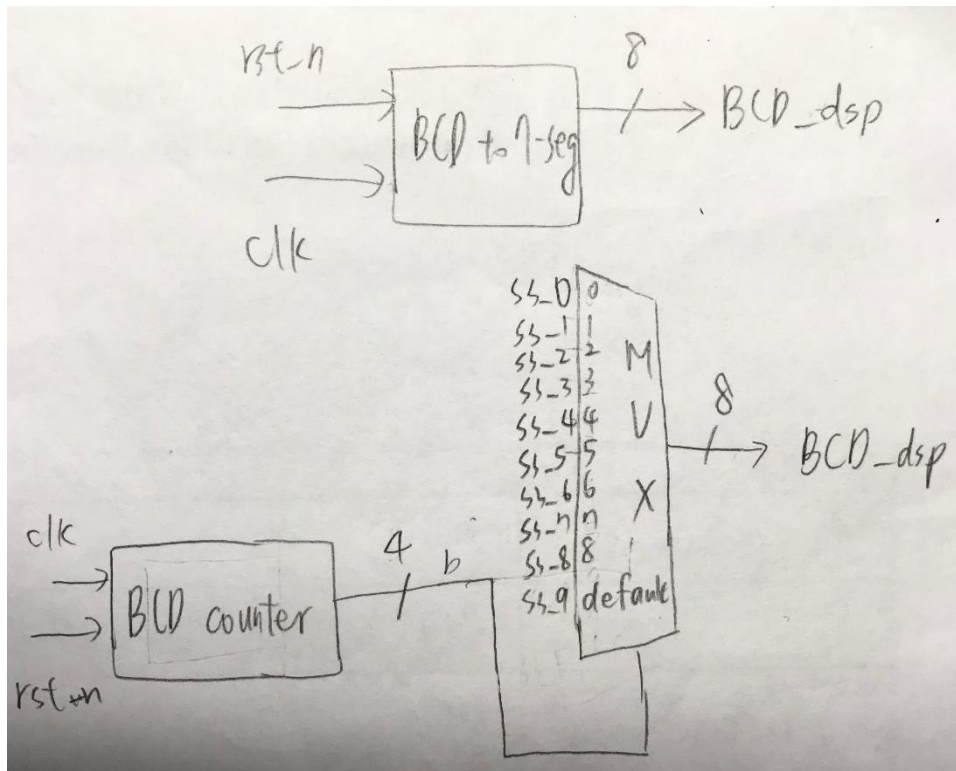
### 3.3 Combine the above two together.

#### Specification:

Input : clk, rst\_n

Output : BCD\_dsp

#### Implementation:



將上面兩個block接起來，BCD counter的output b接到BCD to 7seg display decoder的input b，將FPGA的W5接到input clk，再把output 8-bit BCD\_dsp接到相對應的7-seg ports，當clock開始跑，LED就會顯示9-8-7....-0-9-8...一直循環。

#### 討論:

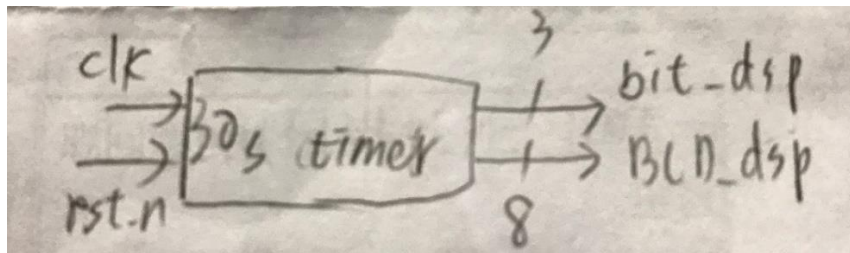
這個實驗跟第二題很像，只要觀念了解，修改一下，這題就很快的

4. (Bonus) Construct a 30 seconds count down timer (stop at 00).

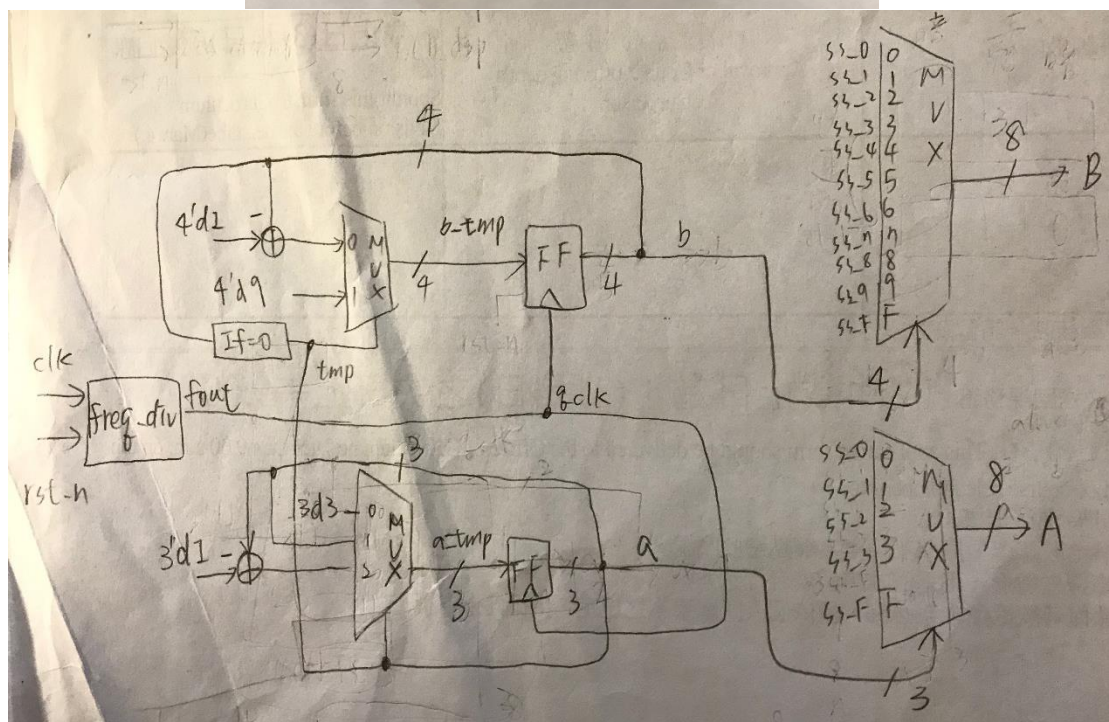
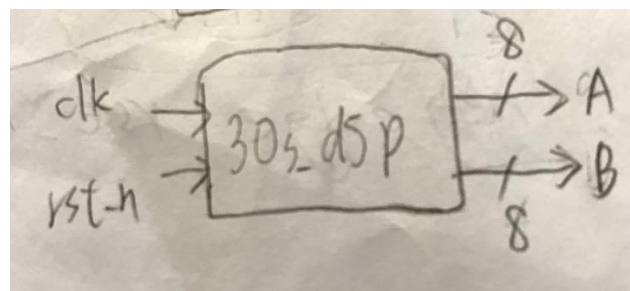
**Specification:**

Input : clk,rst\_n

Output : BCD\_dsp,bit\_dsp



### Implementation:



這題是運用上一題的觀念，只是要做成2位數的count down，我有問過助教，題目說要數到00後停止，但是助教說都可以，所以我讓它從30-29-....數到00之後回到29-28-...這樣一直循環倒數。我分成2個source檔。一個是設計出30s的count down timer，一個是將結果顯示再LED上。

個位數的部分是由4-bit b控制，把freq\_div 1Hz clock的output fout接到wire qclk給FF做為clock，將FF的output設為4個bits b，rst\_n是一個reset的功能，一開始b的值變成設為0，因為是從30開始倒數，當1Hz的clock開始數後，會有一個MUX判斷，當b不是0，會將b-1設給b\_tmp，過1秒再將b\_tmp的值送給b，當b一直向下數，減到0時，b\_tmp就會變成9，再將值給b，這樣b就會從9-8-7-....數到0後，變回9開始數了。

十位數的部分由3-bit a控制，把freq\_div 1Hz clock的output fout接到wire qclk給FF做為clock，將FF的output設為3個bits a，rst\_n是一個reset的功能，一開始a的值變成設為3，從個位數那邊拉出一個值tmp，if b=0，tmp=0，當b為其他數字，則tmp=1，當1Hz的clock開始數後，會有一個MUX判斷，把a和tmp接到MUX中，當tmp為0且a為0，代表個位和十位數都是0，要回到29開始數，就將a\_tmp設為2，若tmp為0但a不是0，代表個位數數到0了，十位數要借位，就將a\_tmp設為a-1，若tmp為1，則a\_tmp為a，就是十位數不需要改變的

意思，過1秒後a\_tmp的值會送給a，a再回到判斷的地方，藉由這些

blocks，3-bit a和4-bit b就能從30開始倒數到00後回到29開始倒數

了。

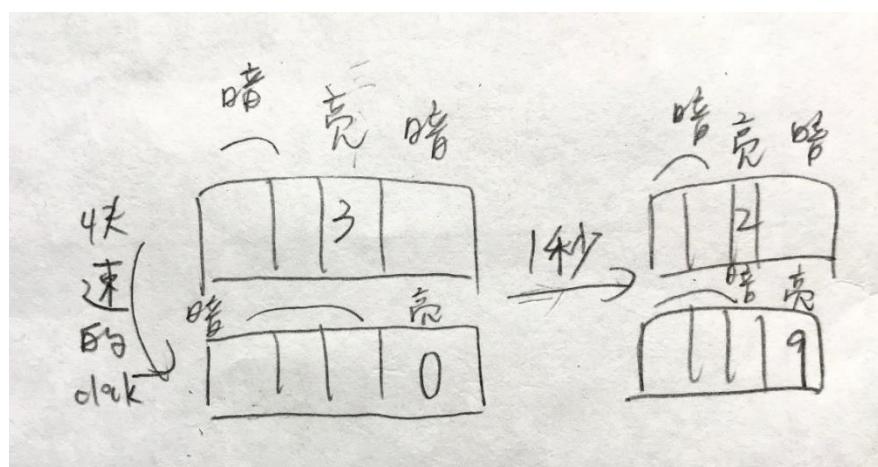
為了要把數字顯示再7-seg LED燈上，先將控制7-seg LED要顯示的數

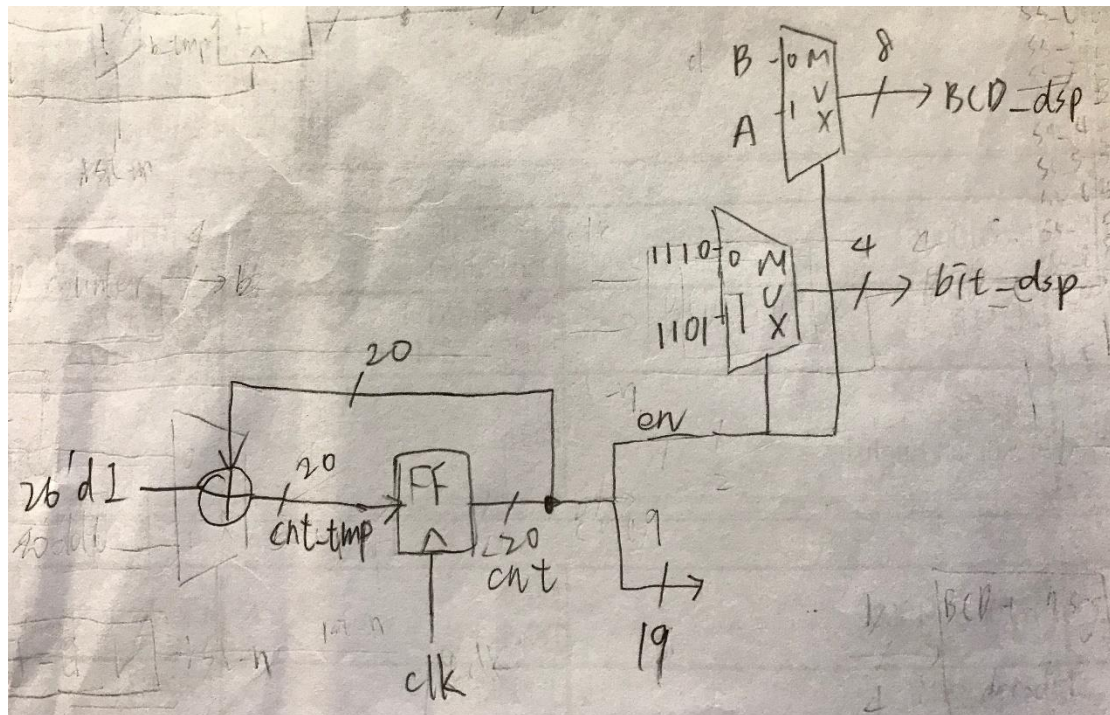
字的8個bits define給對應的SS\_0~9

```
`define SS_0 8'b00000011
`define SS_1 8'b10011111
`define SS_2 8'b00100101
`define SS_3 8'b00001101
`define SS_4 8'b10011001
`define SS_5 8'b01001001
`define SS_6 8'b01000001
`define SS_7 8'b00011111
`define SS_8 8'b00000001
`define SS_9 8'b00001001
```

再把3-bit a和4-bit b分別接到MUX上，output是8個bits A和B，這樣就

能選擇相對應的8個bits去控制7-seg LED。





因為7-seg一次只能顯示相同的數字，所以要利用LED燈快速閃爍與視覺停留呈現一次顯示2個不同數字的感覺，先做出一個frequency divider，將原本100MHz的clock頻率變小，FF出來是20-bit cnt，會一直加1從0數到 $2^{20}-1$ 然後又歸零，將最左邊的bit取出來設為en，就是頻率比100MHz小的clock了，這個en會接到兩個MUX，一個是選擇亮7-seg的最右邊兩個燈的哪一個，output是4-bit bit\_dsp，會接到FPGA板的U2,U4,V4,W4，可以控制7-seg亮的燈，1110代表亮最右邊的燈，其他暗，1101讓右邊第二個燈亮其他暗。另一個MUX是選擇亮的那個燈要顯示的數字，output是8-bit BCD\_dsp，把上面設計好的30s count down timer的output A,B接到這個MUX中，當en為0，顯示B，en為1顯示另一個數字A，當en快速變化，就能看到2個不同的數字。



當這兩個source檔接起來後，7-seg就能隨著30s count down timer的變化而顯示出來。

### 討論:

這個實驗比較困難的地方是想出十位數的部分要怎麼隨著個位數做變化，其他部分就是利用以前做過的實驗，將全部結合起來後就能完成了。

### 結論:

這次demo的時候，助教很認真看我的code，跟我講了哪些地方需要修改，雖然結果跑得出來，但是程式碼很容易產生錯誤，之後比較困難的實驗就會容易有bug，還好助教有提醒，讓我早一點把錯誤糾正。