**Lab4.1**Construct a 4-bit synchronous binary up counter (b3b2b1b0) with the 1-Hz clock frequency.

1. System specification

功能: LED表示的二進位數字會每過一秒會往上數一。

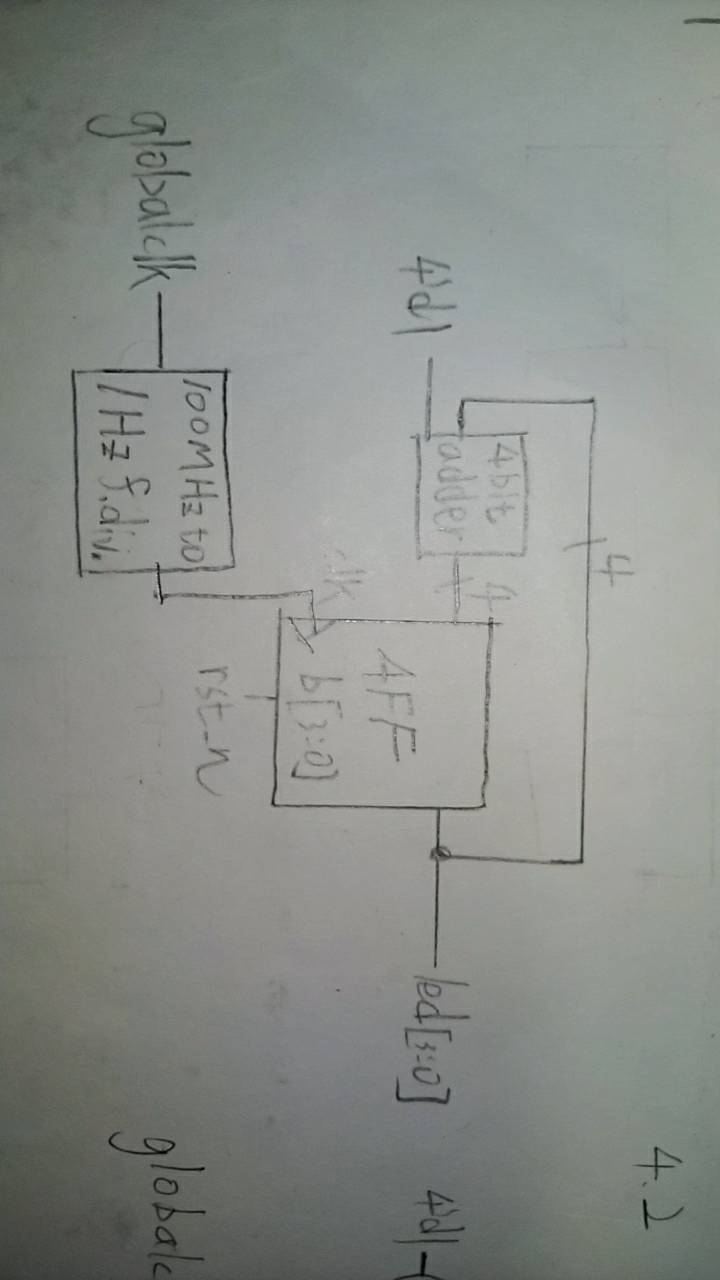
輸入: globalclk (基本的石英震盪器)

rst\_n (將計數器歸零)

輸出: [3:0]b (輸出LED訊號)

2. System implementation

這個向上計數器的原理很簡單，跟上次的除頻器的原理差不多。首先是4個FFs用來存值，然後其輸出的訊號，會同時傳給LED燈及四位元加法器。然後四位元加法器讓訊號加一後的值會於下一個時脈(1hz)正緣觸發暫存器，使得暫存器於每一個時脈正緣後其儲存及輸出的值會比原先的值再加一。然後當值為15時，計數器會由於overflow而又重新變回零。



3. I/O pin assignment

|  |  |  |  |
| --- | --- | --- | --- |
|  | 變數接收處 | 晶片I/O點 | 描述 |
| 輸入 | globalclk | W5 | 石英震盪器時脈輸入 |
| rst\_n | V17 | 重置訊號 |
| 輸出 | b[3] | V19 | 控制晶片LED燈 |
| b[2] | U19 |  |
| b[1] | E19 |  |
| b[0] | U16 |  |

**Lab4.2** Construct a single digit BCD up counter with the divided clock as the clock frequency and display on the seven-segment display.

1. System specification

功能: 七段顯示器上的十進位數字會每過一秒會往上數一，然後至九後又 歸零。

輸入: globalclk (基本的石英震盪器)

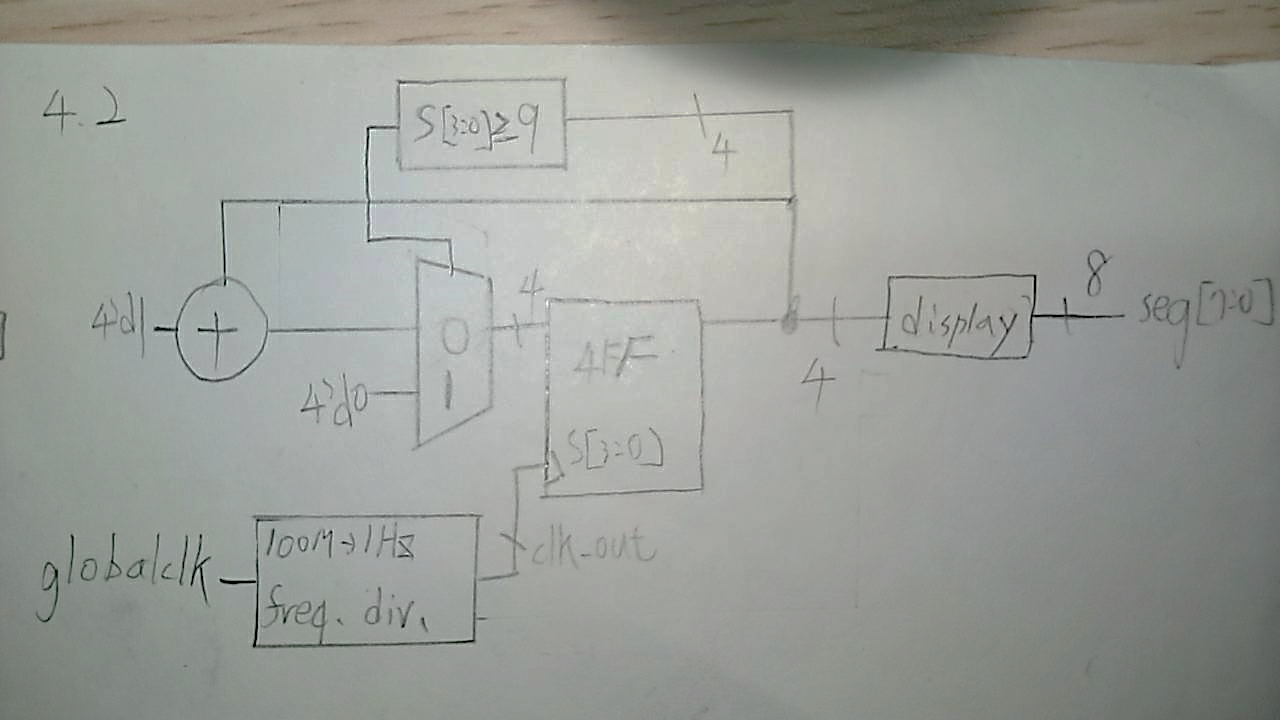
rst\_n (將計數器歸零)

輸出: [7:0]seg (輸出顯示器訊號)

2. System implementation

主要的計數原理和上一題差不多，只是由於這題要的是BCD而不是binary數字，所以不能使用overflow方法歸零。我們要的是讓暫存器在九的時候要歸零，所以我們可以使用一個當訊號為九時會輸出一的比較器，然後其輸出訊號會接到多工器，比較器輸入零時會選擇經過fulladder處理過的加一的訊號輸入到暫存器，反之則是輸入零的訊號。

暫存器的訊號則會被輸入到display decoder去，在那裏會有一個4 to 16的多工器會輸出由輸入的BCD數字所對應的七段顯示器腳位訊號。



3. I/O pin assignment

|  |  |  |  |
| --- | --- | --- | --- |
|  | 變數接收處 | 晶片I/O點 | 描述 |
| 輸入 | globalclk | W5 | 石英震盪器時脈輸入 |
| rst\_n | V17 | 重置訊號 |
| 輸出 | seg[7] | W7 | 控制七段顯示器腳位 |
| seg[6] | W6 |  |
| seg[5] | U8 |  |
| seg[4] | V8 |  |
| seg[3] | U5 |  |
| seg[2] | V5 |  |
| seg[1] | U7 |  |
| seg[0] | V7 |  |

**Lab4.3** Construct a single digit BCD down counter with the divided clock as the clock frequency and display on the seven-segment display

1. System specification

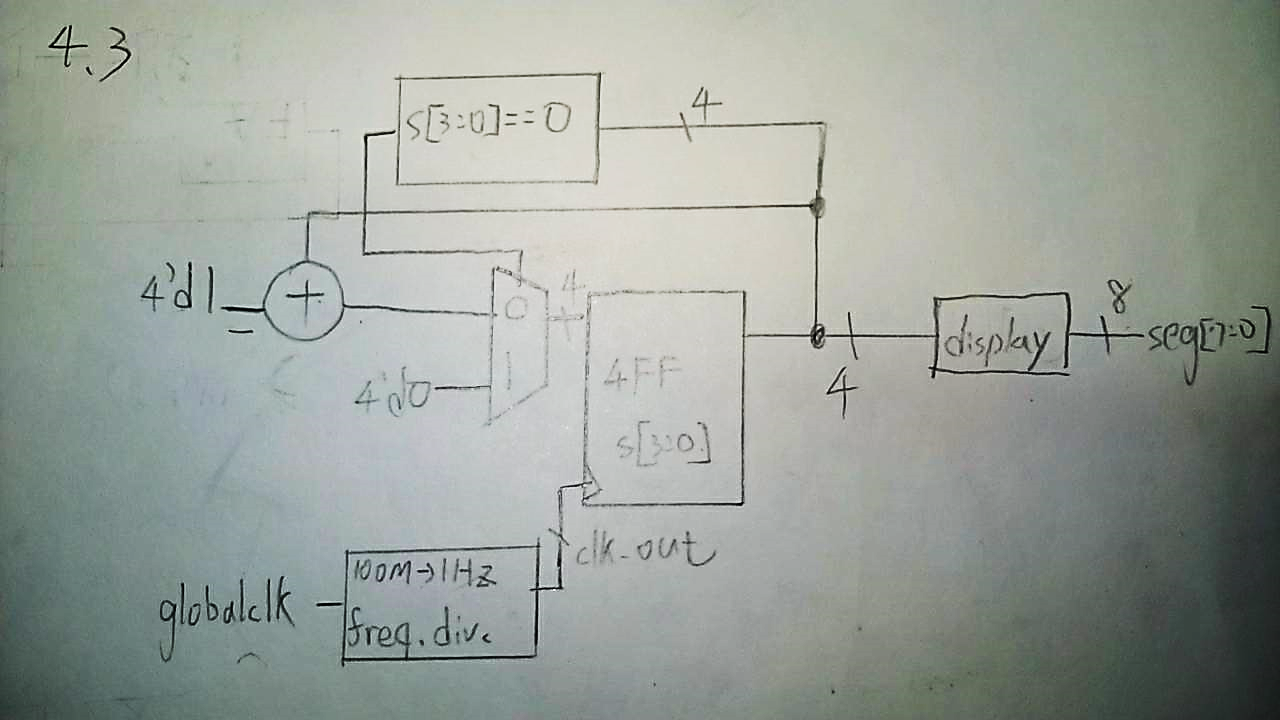
功能: 七段顯示器上的十進位數字會每過一秒會往下數一，然後至零後又 會跑到九。

輸入: globalclk (基本的石英震盪器)

rst\_n (將計數器歸九)

輸出: [7:0]seg (輸出顯示器訊號)

2. System implementation

這題的計數原理跟第二題很相像，只是要輸入fulladder的訊號為暫存器訊號及-1，讓counter改成向下數。然後比較器要改成等於零時會使得暫存器輸入九的訊號而已。

3. I/O pin assignment

|  |  |  |  |
| --- | --- | --- | --- |
|  | 變數接收處 | 晶片I/O點 | 描述 |
| 輸入 | globalclk | W5 | 石英震盪器時脈輸入 |
| rst\_n | V17 | 重置訊號 |
| 輸出 | seg[7] | W7 | 控制七段顯示器腳位 |
| seg[6] | W6 |  |
| seg[5] | U8 |  |
| seg[4] | V8 |  |
| seg[3] | U5 |  |
| seg[2] | V5 |  |
| seg[1] | U7 |  |
| seg[0] | V7 |  |

**Lab4.4** Construct a 30 seconds count down timer (stop at 00)

1. System specification

功能:由三十秒向下計數到零後停止

輸入: globalclk (基本的石英震盪器)

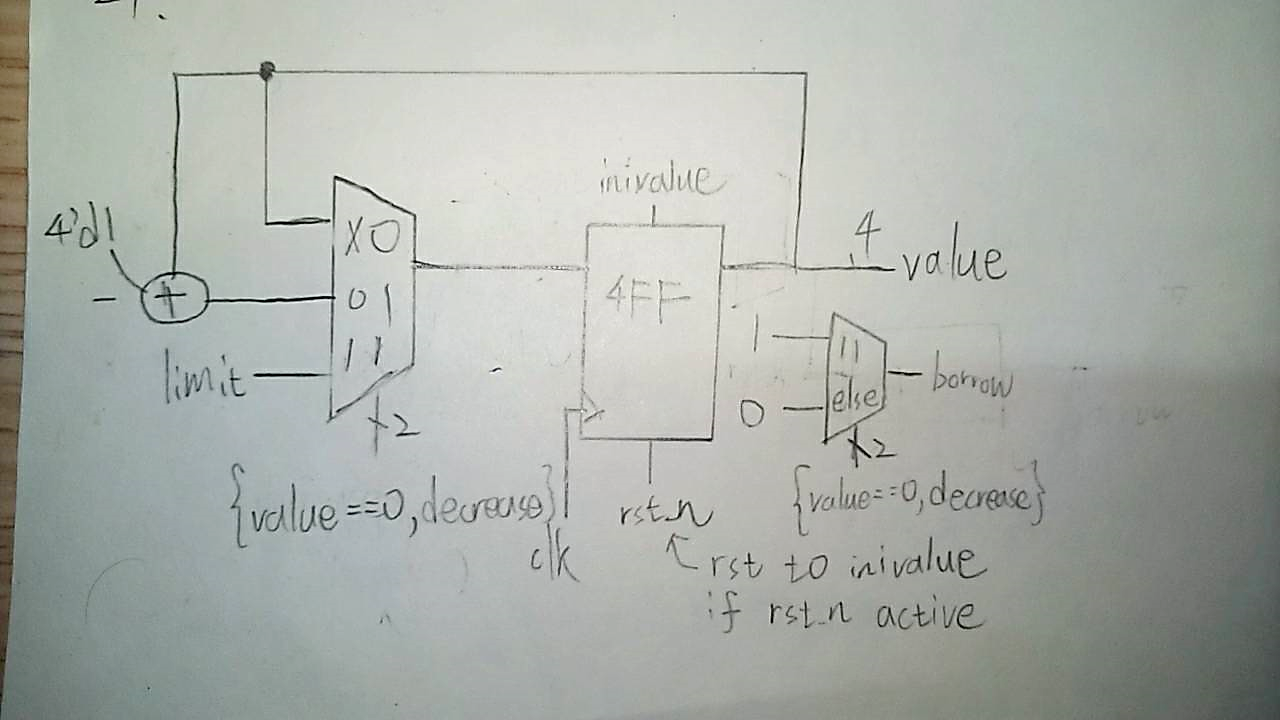
rst\_n (將計數器歸九)

輸出: [7:0]seg (輸出顯示器訊號)

[3:0]dis (控制每個七段顯示器開關)

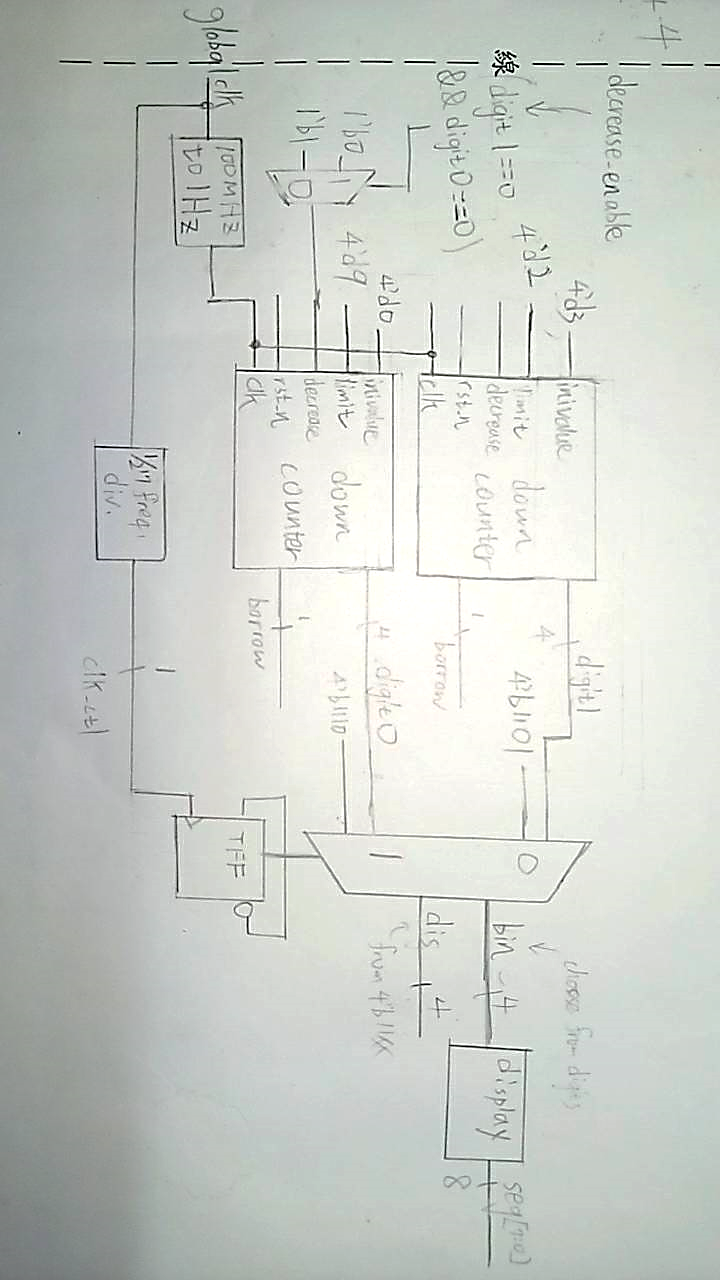
2. System implementation

首先，先由30秒倒數器的核心，一位元向下計數器開始講起。由於本題會用到二位元向下計數器，所以向下計數器應該要具有本位元倒數和判斷借位的功能。所以會有一個decrease是用來偵測要不要使本位元減一，當前位元為零時或是功能上有需要，就會輸入高電位到decrease，啟動連減功能，而當decrease為低電位時，則會保持原本的值。而判斷是不是該借位的時候，就由decrease及值為零時判斷，假設兩項條件皆成立時，borrow(借位)為一輸出，同時本位元會輸入limit值，也就是該位元的最大值。而當值不為零但decrease為一時，功能則和第三題的累減器一樣。decrease為零時則不減值。



好，有了倒數計時器後，剩下就比較好辦了。先將倒數計時器接上除頻器(1hz)，再讓個位數的倒數計時器decrease接到decrease\_enable上，decrease\_enable在兩位數皆為零時會變成低電位，使得計時器停止。

然後，輸出的兩個位數會被送到多工器去。多工器由一個TFF控制，而TFF所接上的時脈為大概一毫秒。當多工器為零時會輸出第二的七段顯示器的訊號及數字，數字之後會轉成七段顯示器腳位訊號輸出。而多工器為一時則輸出個位數訊號。



3. I/O pin assignment

|  |  |  |  |
| --- | --- | --- | --- |
|  | 變數接收處 | 晶片I/O點 | 描述 |
| 輸入 | globalclk | W5 | 石英震盪器時脈輸入 |
|  | rst\_n | V17 | 重置訊號 |
| 輸出 | seg[7] | W7 | 控制七段顯示器腳位 |
|  | seg[6] | W6 |  |
|  | seg[5] | U8 |  |
|  | seg[4] | V8 |  |
|  | seg[3] | U5 |  |
|  | seg[2] | V5 |  |
|  | seg[1] | U7 |  |
|  | seg[0] | V7 | 控制七段顯示器開關 |
|  | dis[3] | W4 |  |
|  | dis[2] | V4 |  |
|  | dis[1] | U4 |  |
|  | dis[0] | U2 |  |

Conclusion & discussion:

這一次的lab都是上學期所教過的東西，只是把它利用晶片實體化而已。而第四題我覺得是有挑戰性的一題，不僅要考慮到借位的問題，還要將輸出的值顯示到顯示器上，但藉由畫出block diagram，把每一個block的功能及輸入輸出都跑過一輪後，這個難題就迎刃而解了。