

Lab 03

105060016 謝承儒

一. Lab03-1 : Frequency Divider(原本的 $\frac{1}{2^{27}}$ 倍)

1. Design Specification :

- A. Input : f_{crystal} //輸入的頻率(100MHz)
Rst //當=1 時，使板子暫停運作
- B. Output : f_{out} //頻率為 f_{crystal} 的 $1/2^{27}$ 倍(接近 1Hz)
- C. Wire : [26:0] cnt
[26:0] cnt_tmp //作為一個 27-bits 的 counter

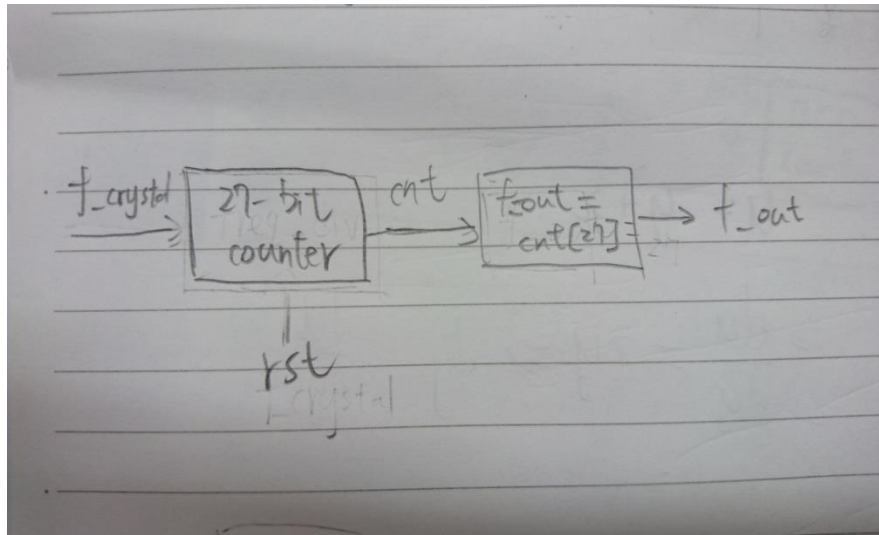


圖 1 Lab03-1 的區塊圖

2. Design Implementation :

- A. Logic function :
cnt_tmp = cnt + 1
f_out = cnt [27]

B. Logic diagram :

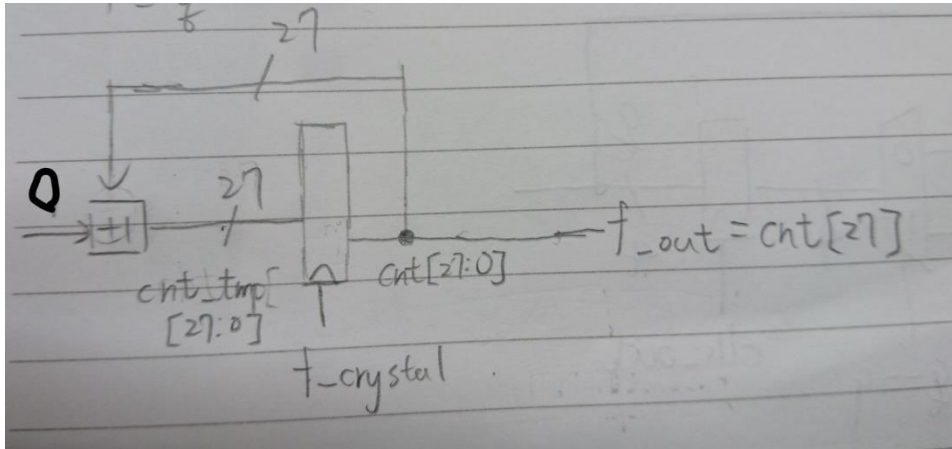


圖 2 Lab03-1 的邏輯圖

C. Pin assignment :

a. Input :

1) $f_crystal = W5$

2) $rst = U16$

b. Output :

1) $f_out = V17$

3. Discussion :

A. 思考過程 :

這題是要將輸入的頻率($f_crystal$)降為原先的 $\frac{1}{2^{27}}$ 倍再將其輸出

(f_out)。

我選擇使用一個 27-bit 的 counter(cnt)，只要每次 $f_crystal$ 從 0 變為 1

就使 cnt 加 1，而 cnt 的最高位(cnt[27])在 $f_crystal$ 變化 2^{26} 次便會變化，在下一個 2^{26} 次後又回變回來，如此 cnt[27]的變化頻率即

$f_crystal$ 的 $\frac{1}{2^{27}}$ 倍。因此，cnt[27]便是我們所求的 f_out 。

B. 過程中的 Bug :

起初並不曉得板子上 clk 的接腳是 W5，而隨便接上一個 Switch，導致程式無法編譯。之後回頭看了講義就解決了。

二.Lab03-2 : Frequency Divider(輸出為 1Hz)

1. Design Specification :

- A. Input : f_{crystal} //輸入的頻率(100MHz)
rst //當=1 時，使板子暫停運作
- B. Output : f_{out} //輸出的頻率(1Hz)
- C. Wire : [26:0] cnt //27-bit 的 counter 裡的值

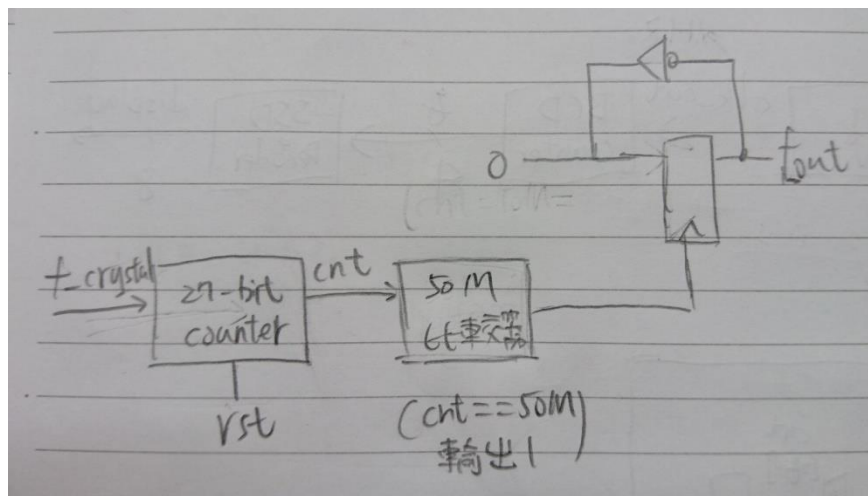


圖 3 Lab03-2 的區塊圖

2. Design Implementation :

- A. Logic diagram :

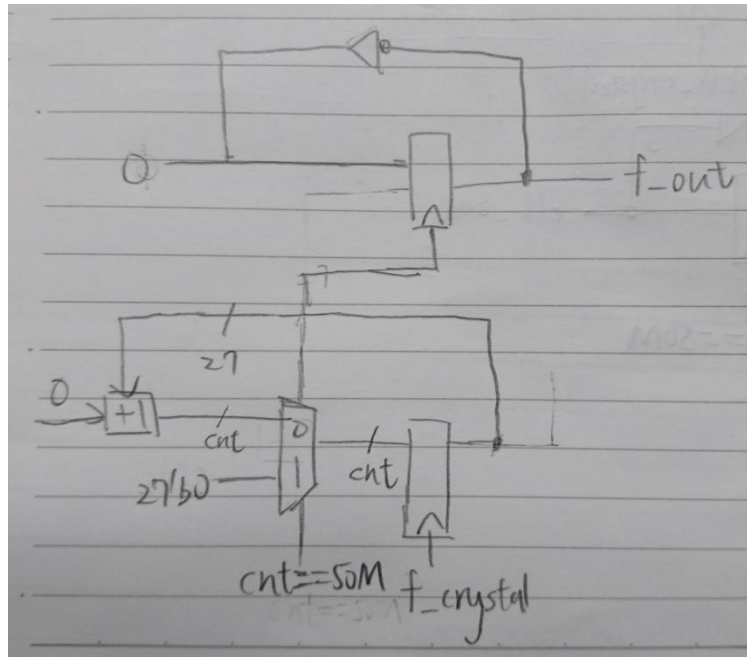


圖 4 Lab03-2 的邏輯圖

B. Pin assignment :

a. Input :

- 1) f_crystal : W5
- 2) rst : U16

b. Output :

- 1) f_out : V17

3. Discussion :

A. 思考過程 :

這次要輸出的頻率(f_{out})為 1Hz，因為 100M 沒辦用 2 的次方達成，沒辦法使用上一個實驗的方法。

因此，改為利用 27-bit 的 counter(cnt)，每個 $f_{crystal}$ 的週期都使 cnt 加 1。當 cnt 數到 50M 時，便讓 $f_{out} = \sim f_{out}$ 且 cnt 歸零，如此 $f_{crystal}$ 跑完 100M 個週期後， f_{out} 也跑出 1 個週期，達到 1Hz 的效果。

B. 過程中的 Bug :

本來將除頻器和 counter 兩者的 always 區塊分開來寫，不過卻編譯錯誤，同學說 code 裡的變數盡量不要在兩個不同的 always 裡存值，之後就把兩個 always 合成一個來寫。雖然之後可以編譯，不過 code 整體變得十分難看懂。

三.Lab03-3 : 1Hz 的 BCD Counter

1. Design Specification :

- | | |
|---------------------------|------------------------|
| A. Input : clk_crystal | //板子輸入的頻率(100MHz) |
| rst | //讓板子暫停運作 |
| B. Output : [7:0] display | //七段顯示器的顯示碼 |
| C. Wire : clk_out | //用除頻器得到的 1Hz clk |
| [26:0] cnt | //27-bit 的 counter 裡的值 |
| [3:0] q | //BCD counter 裡的值 |

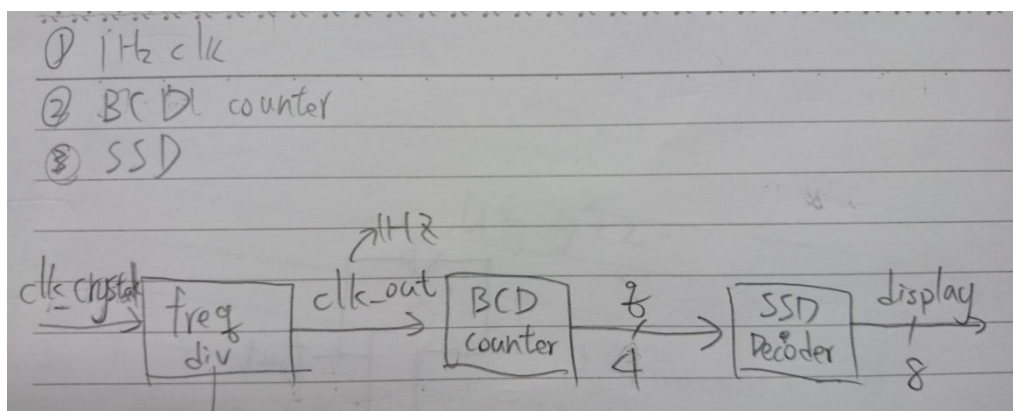


圖 5 Lab03-3 的區塊圖

2. Design Implementation :

A. Logic function :

a. Output :

1) $\text{display}[0] = 1$

2) $\text{display}[1] = q_3'q_2'q_1' + q_3'q_2q_1q_0'$

3) $\text{display}[2] = q_3'q_2'q_0 + q_3'q_2'q_1 + q_3'q_1q_0$

4) $\text{display}[3] = q_3q_2q_1' + q_3'q_0$

5) $\text{display}[4] = q_3q_2 + q_3q_1 + q_2'q_1'q_0 + q_2q_1'q_0' + q_2q_1q_0$

6) $\text{display}[5] = q_3q_2 + q_3q_0 + q_2'q_1q_0'$

7) $\text{display}[6] = q_3q_2 + q_3q_1 + q_3q_0 + q_2q_1'q_0 + q_2q_1q_0'$

8) $\text{display}[7] = q_3'q_2'q_1'q_0 + q_3'q_2q_1'q_0'$

B. Logic diagram :

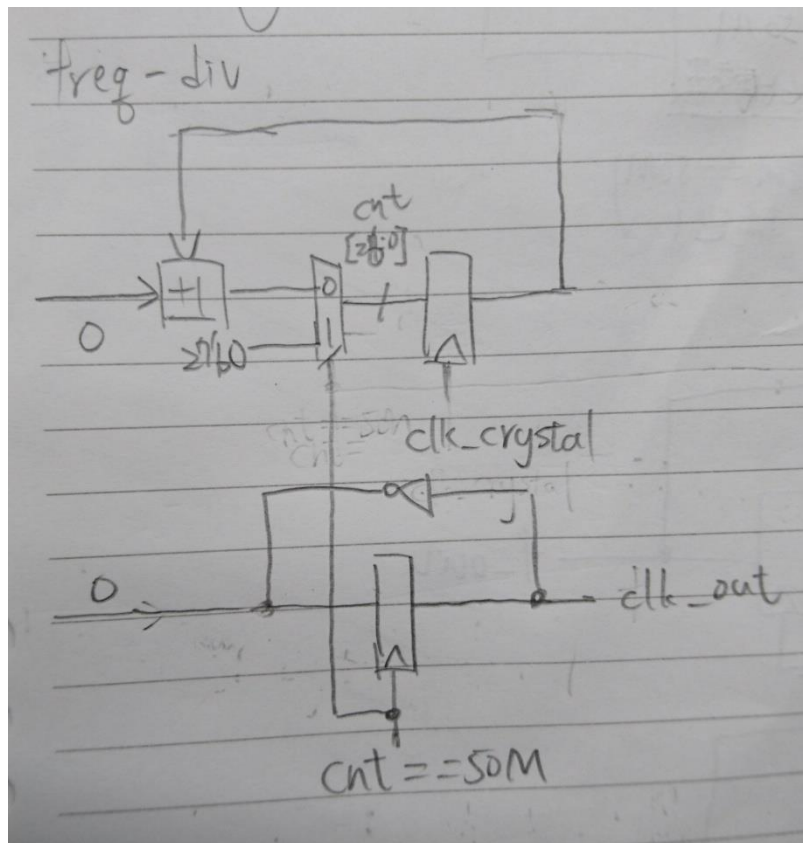


圖 6 除頻器的邏輯圖

把 `q` 輸入進來，經過 `Decoder` 解碼成 8-bit 的顯示碼 (`[7:0]display`)，可以利用畫 `truth table`、`K-MAP` 來取得其 `function`，在 `code` 裡可以使用 `case` 來達到同樣的效果。

B. 遇到的 Bug：

起初打開 `rst` 時，產生的效果並不是歸零，而是暫停。檢查 `code` 後發現當 `rst` 打開時，只有停止除頻器運作，也就是暫停 `clk_out`。除此之外應該還要將 `q` 也設為 0，在 `BCD Counter` 那裏的 `code` 改完後便能正常運作了。

四.Conclusion：

這次實驗結合前兩次實驗的結果，不過還是遇到很多小 `bug`，耗了很多時間。但看到自己做出來的結果越來越複雜也很有成就感。

五.Reference：

1. 老師給的實驗講義
讓我知道 `code` 怎麼打和辨別不同元件的接腳。
2. 上學期的邏輯設計講義
讓我知道 `Counter` 和除頻器的原理。