
可程式邏輯陣列設計實習

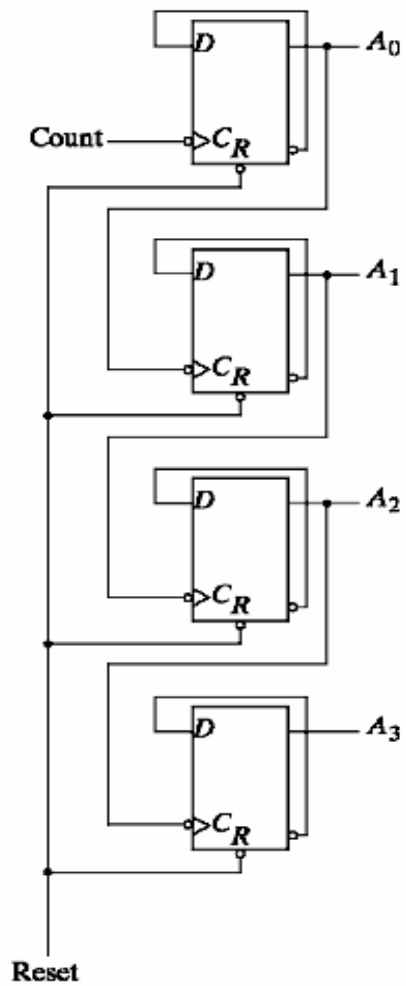
Lecture 7

課程代碼：EE3811301
指導老師：王乃堅 老師
實習課助教：周 霖

漣波計數器

- 計數器由正反器構成，可以記錄狀態的變遷，或可說是正反器隨時脈的變化次數做固定狀態的循環。
- 計數器以正反器狀態改變是否與計數時脈同步來區分，可分為同步計數器與非同步兩種。
- 漣波計數器是一種**非同步計數器**，由 D 型正反器推動下一只 D型正反器產生計數狀態，每一只正反器對於脈波的反應都有延遲現象，而且越到後面的正反器延遲越大，最大延遲時間與正反器的個數成正比。

漣波計數器



漣波計數器

漣波計數器

▶ 漣波計數器(I)：

```
module ripple_counter (Count, Reset, A0, A1, A2, A3);  
input Count, Reset;  
output A0, A1, A2, A3;  
wire A0, A1, A2, A3;  
D_FF D1(~A0, A0, ~Count, Reset);  
D_FF D2(~A1, A1, ~A0, Reset);  
D_FF D3(~A2, A2, ~A1, Reset);  
D_FF D4(~A3, A3, ~A2, Reset);  
endmodule
```

漣波計數器

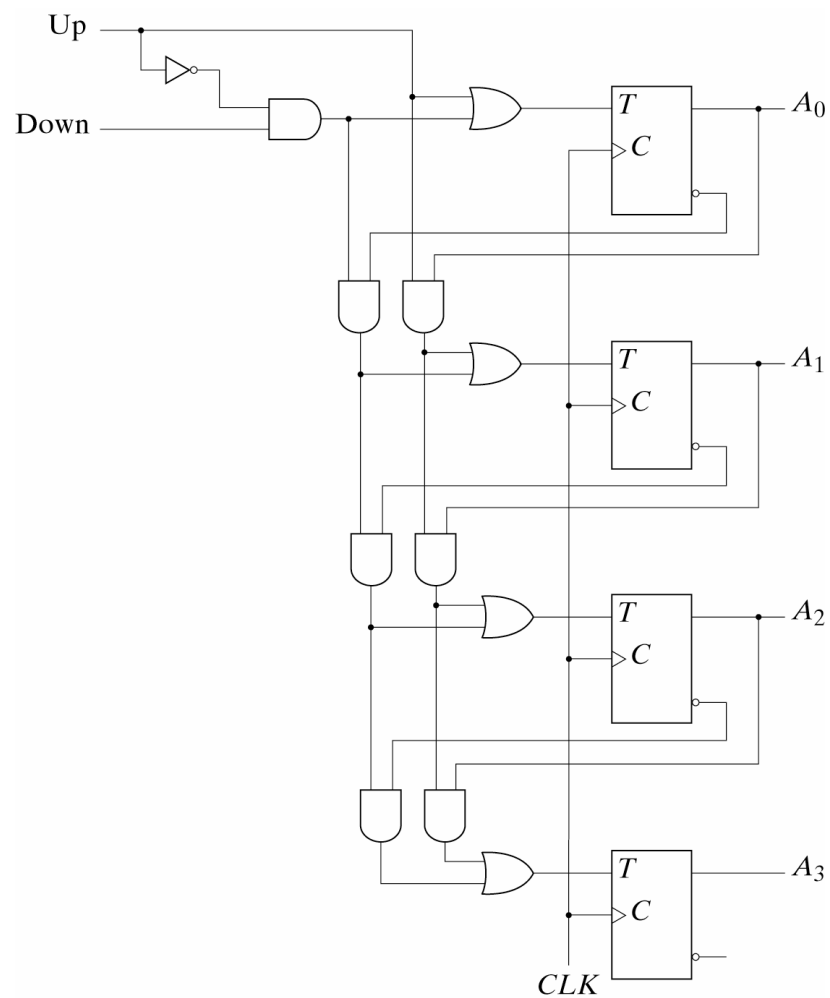
▶ 漣波計數器(II)：

```
//D flip-flop  
module D_FF(D, Q, CLK, RST);  
input D, CLK, RST;  
output Q;  
reg Q;  
always @(posedge CLK or negedge RST)  
    if(~RST)  
        Q = 1'b0;  
    else  
        Q = D;  
endmodule
```

上、下數計數器

- 上、下數計數器為一同步計數器，結構上與漣波計數器最大的不同，就是同步計數器中所有正反器的時脈輸入端（CLK）都接在一起，這意味著當計數脈波發生時，所有正反器將同步反應輸出狀態。
- 當上數時，輸入 Up 為 1，Down 為 0，由 0000 數至 1111 後再返回 0000繼續向上計數；下數時，輸入 Up 為 0，Down 為 1，由 1111 數至 0000再返回 1111 繼續向下計數。

上、下數計數器



上、下數計數器

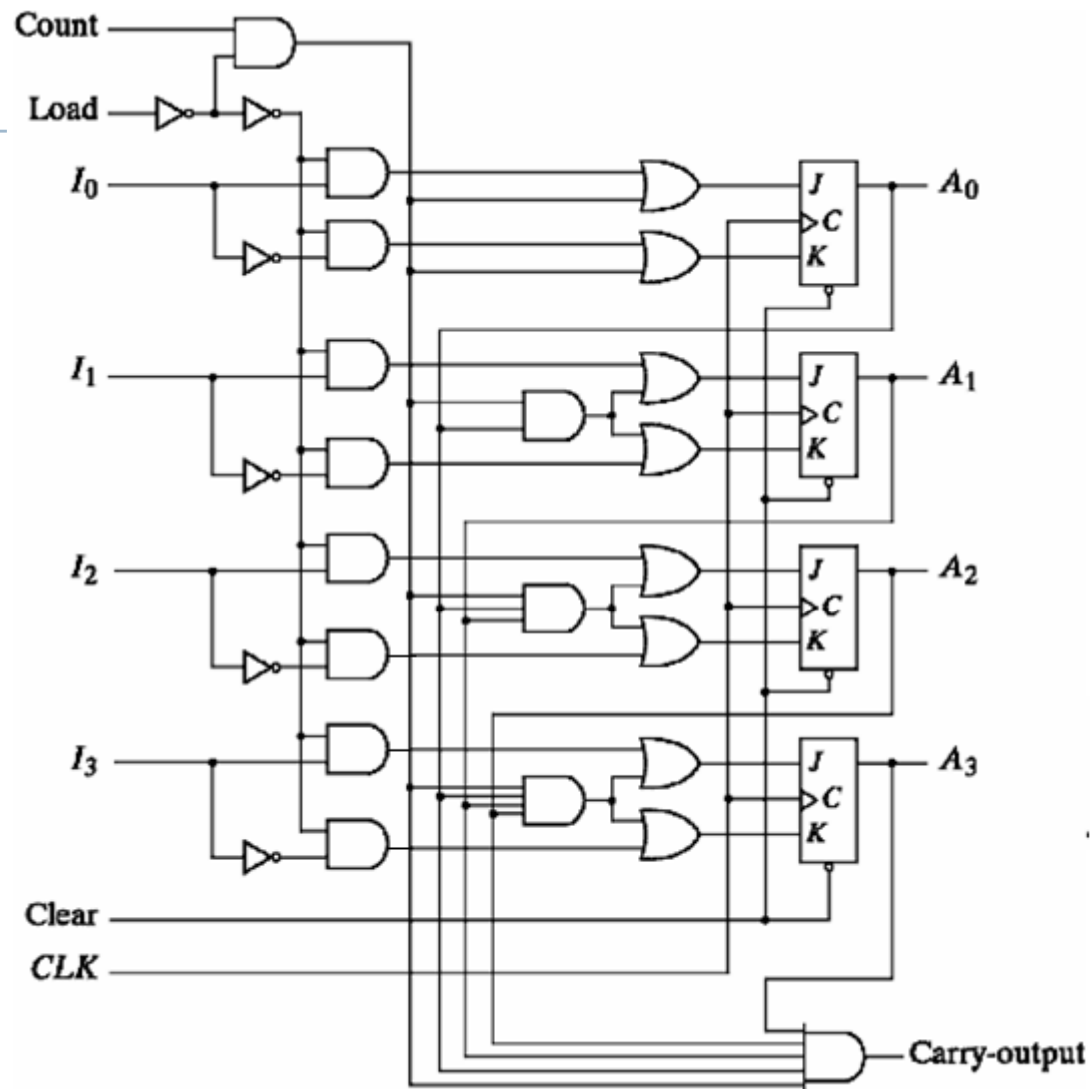
上、下數計數器

```
// Behavioral description of up-down counter
module up_down_counter (up, down, CLK, A0, A1, A2, A3);
input up, down, CLK;
output A0, A1, A2, A3;
reg A0, A1, A2, A3;
always @(posedge CLK)
    if (up == 1 && {A3, A2, A1, A0} == 4'b1111)
        {A3, A2, A1, A0} = 4'b0000;
    else if (down == 1 && {A3, A2, A1, A0} == 4'b0000)
        {A3, A2, A1, A0} = 4'b1111;
    else if (up == 1)
        {A3, A2, A1, A0} = {A3, A2, A1, A0} + 4'b0001;
    else if (down == 1)
        {A3, A2, A1, A0} = {A3, A2, A1, A0} - 4'b0001;

endmodule
```

四位元同步計數器

- 時脈CLK為正緣觸發。
- 當Clear為0且負緣時，即將計數器之輸出清除為0000。
- Load為同步載入，當 Load 訊號為1時，在 CLK 之正緣會將4個輸入資料 I_0 、 I_1 、 I_2 、 I_3 載入到4個JK正反器。
- 計數器之 Count 為1時，計數器才能正常計數。
- 當輸出狀態為1111時，Carry-output 輸出一個進位訊號，當計數器多級串接時，作為下一級計數器的致能信號，使計數器做同步計數。



四位元同步計數器

四位元同步計數器

- ▶ 四位元同步計數器所有正反器同時使用一時脈觸發，同步反應輸出狀態。
- ▶ 令同步計數器輸入 I_0 I_1 I_2 I_3 為 0000，當 CLK 正緣觸發，Count 訊號為1時，計數器向上計數，當數至 1111 時，JK正反器之輸入皆為1，下個 CLK 正緣觸發，正反器裡的值變為 Complement， A_0 A_1 A_2 A_3 輸出為 0000 再繼續向上計數，每16 個 CLK 為1週期。

LAB 7-1

- ▶ 請使用Verilog HDL描寫出四位元同步計數器，於Quartus II 模擬訊號波型加以驗證結果。

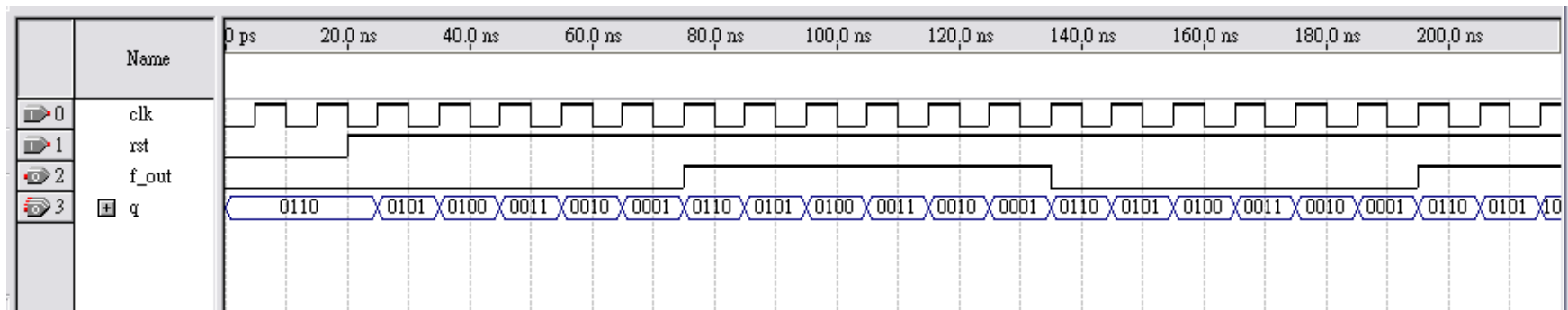
LAB 7-2

- 除頻電路由計數電路衍生而來。
- 此除頻器輸入端有 CLK 及 RST，輸出為 f_out，當 RST 為 0 時 f_out 重設為 0。
- 當 RST 為 1 時，內部計數器開始計數，q[3:0] 由 0110 數至 0001 時， $f_out = \sim f_out$ ，使輸出 f_out 之輸出週期為 CLK 週期之 12 倍。
- q[3:0] 再由 0110 繼續下數，此即為除以 12 之除頻器。

LAB 7-2

State table

Present State				Next State			
q[3]	q[2]	q[1]	q[0]	q[3]	q[2]	q[1]	q[0]
0	1	1	0	0	1	0	1
0	1	0	1	0	1	0	0
0	1	0	0	0	0	1	1
0	0	1	1	0	0	1	0
0	0	1	0	0	0	0	1
0	0	0	1	0	1	1	0



Homework

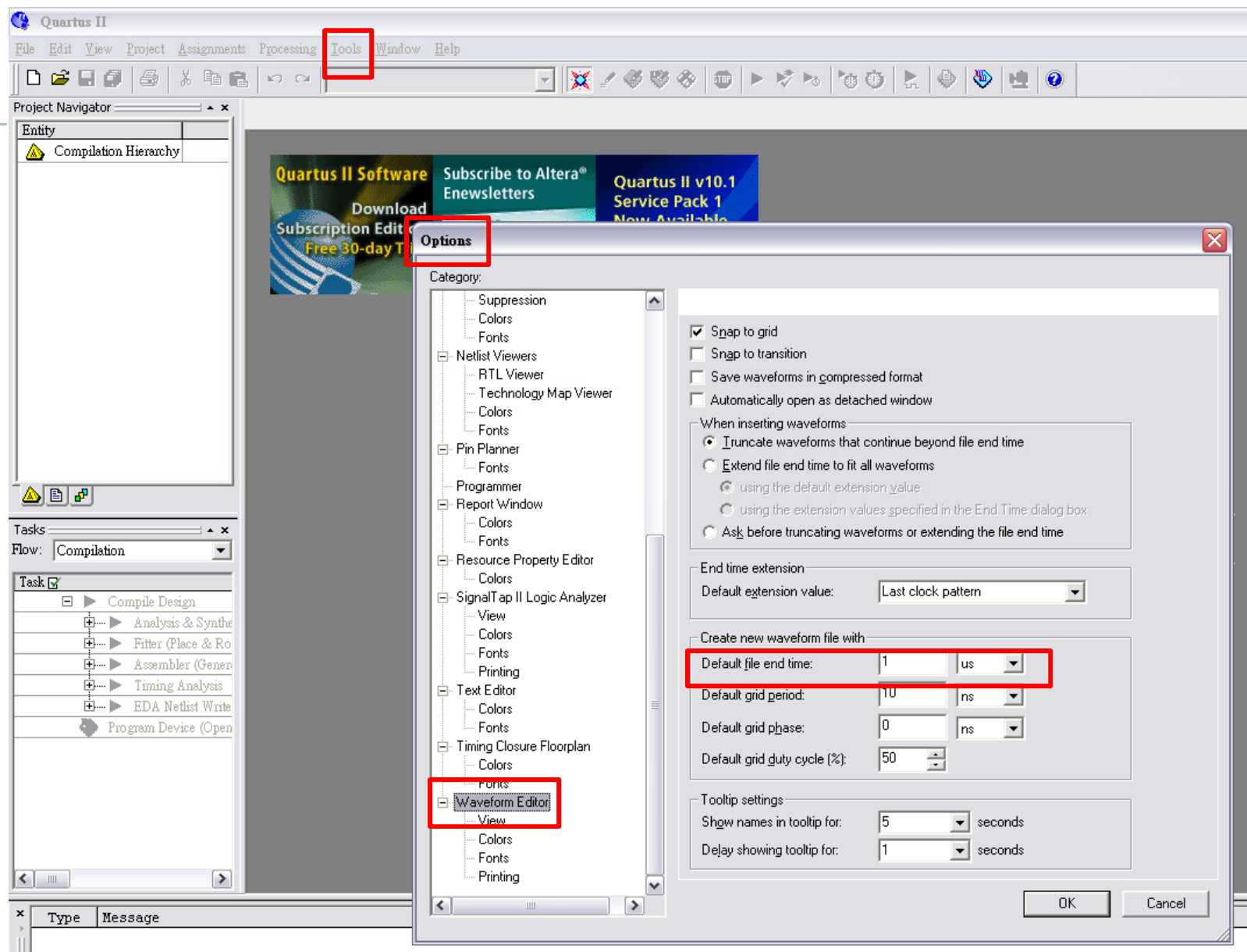
1. 利用兩個四位元同步計數器(或一個八位元)，與1個 8 bit的 4 對 1多工器，設計一個多功能除頻器規格如下：

- ▶ 當控制線 S1，S2 輸入為 00 時，表除頻器沒有動作，輸出為 0
 - ▶ 當控制線 S1，S2 輸入為 01 時，表除頻器除以8
 - ▶ 當控制線 S1，S2 輸入為 10 時，表除頻器除以60
 - ▶ 當控制線 S1，S2 輸入為 11 時，表除頻器除以128
- ▶ P. S. 兩個四位元比較難，所以分數會較高。

Homework

2. 假定系統clock為200MHz，基於此系統clock，設計一電路產生 clock cycle time 為 50 ns，除頻 50 之電路，並其正電位及負電位時間比為1：1（即 duty cycle 為 50 %）。

※若波形檔時間不夠長，可以修改波形時間的最大值，可從Tools > Options > Waveform Editor這裡修改，下頁有設定的教學。



Homework

□ 繳交作業報告格式：

1. 郵件檔名 - EX：學號_系級_姓名_homework1
2. 寄件檔名同上，請壓縮成.rar或.zip檔
3. 作業內容包括：報告word檔、程式.v檔、波形檔
4. 報告內容包括：作業題目、程式說明、結果波形圖及說明、心得
5. 作業上傳：Blackboard System

★繳交時間：04/14 下午3:30以前