

數位系統設計實習

Lecture 5

Counter

指導老師: 陳勇志 教授

實習課助教: 鍾兆鋁

Outline

2

- Verilog 複習 : Behavior-Level
- 同步 / 非同步計數器
Asynchronous / Synchronous counter
- 漣波計數器 Ripple carry counter
- 四位元同步計數器 4_bit Synchronous counter
- 上下計數器 Bidirectional counter
- LAB 5-1 and 5-2

Chapter

3

- Verilog 複習 : Behavior-Level
- 同步 / 非同步計數器
Asynchronous / Synchronous counter
- 漣波計數器 Ripple carry counter
- 四位元同步計數器 4_bit Synchronous counter
- 上下計數器 Bidirectional counter
- LAB 5-1 and 5-2

Recap Behavior Modeling

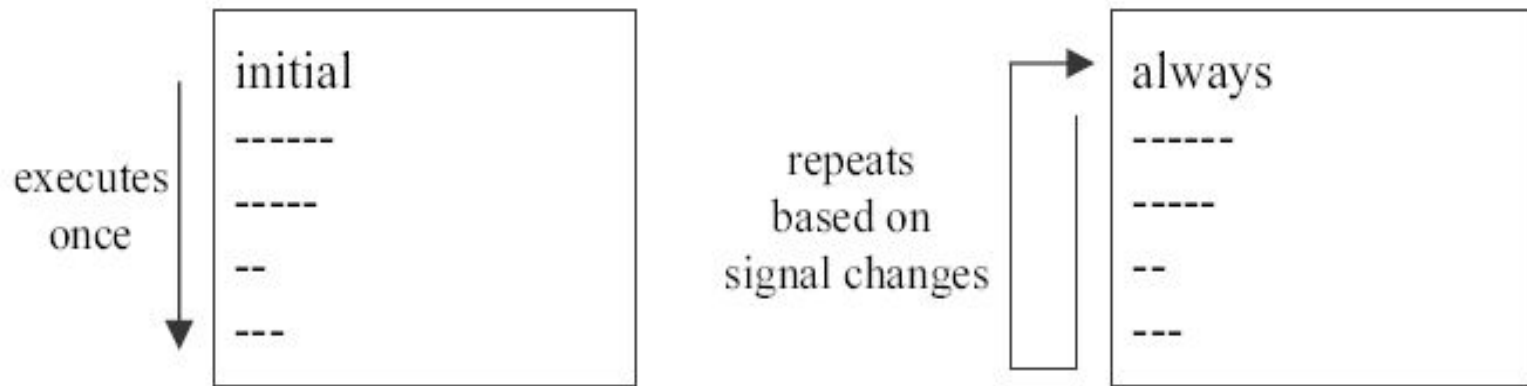
- 驅動某值至reg (等號的左式必為reg, 右式可為net 或 reg)
- 事件與事件之間需用” or ”或”, ”分開。
- 當”事件”有變化時($0 \rightarrow 1$ 、 $1 \rightarrow 0$), 則會執行”敘述”
- 事件可使用正緣觸發posedge($0 \rightarrow 1$)或負緣觸發negedge($1 \rightarrow 0$)。
- 觸發為所有 ”事件” 時, 可使用 * 號。

```
always @( 事件1, 事件2, ... )  
    begin  
        敘述1;  
        敘述2;  
        ... ..  
    end
```

Always撰寫範例碼

Recap Behavior Modeling

- Always 運作原理模式如圖，當事件發生變動就會執行。
- Initial 則只會執行一次。



運作原理圖

Recap Behavior Modeling

- Blocking 敘述由上至下執行，資料會覆蓋。
- Nonblocking 敘述同步執行，資料不會覆蓋。

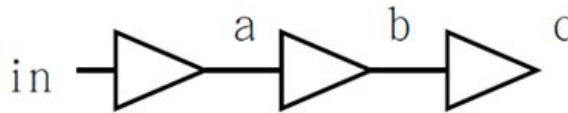
Blocking :

```
a = in;
```

```
b = a;
```

```
c = b;
```

```
//in=a=b=c
```

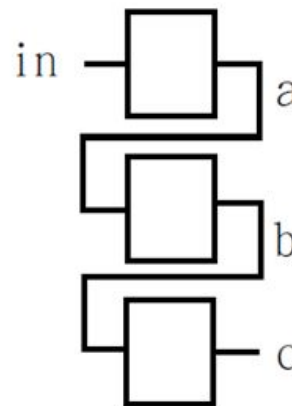


Nonblocking :

```
a <= in;
```

```
b <= a;
```

```
c <= b;
```



運作原理圖

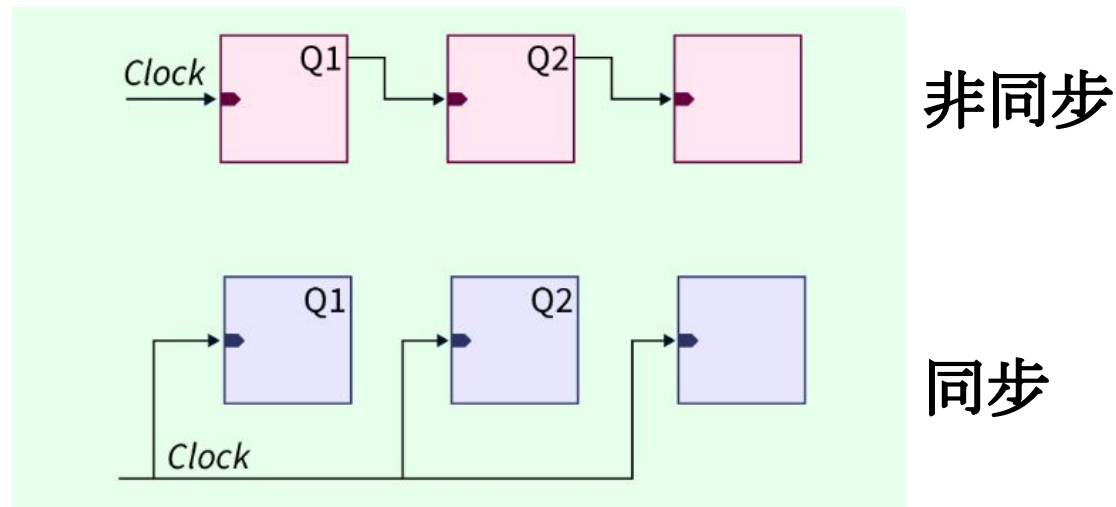
Chapter

7

- Verilog 複習 : Behavior-Level
- 同步/非同步計數器
Asynchronous / Synchronous counter
- 漣波計數器 Ripple carry counter
- 四位元同步計數器 4_bit Synchronous counter
- 上下計數器 Bidirectional counter
- LAB 5-1 and 5-2

Asyn / Synchronous counter

- 計數器由正反器(Flip-Flop)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與否根據時脈同步來區分。
- 非同步：本級Flip-Flop時脈與前級有關，不同時觸發。
- 同步：所有Flip-Flop時脈同步觸發。



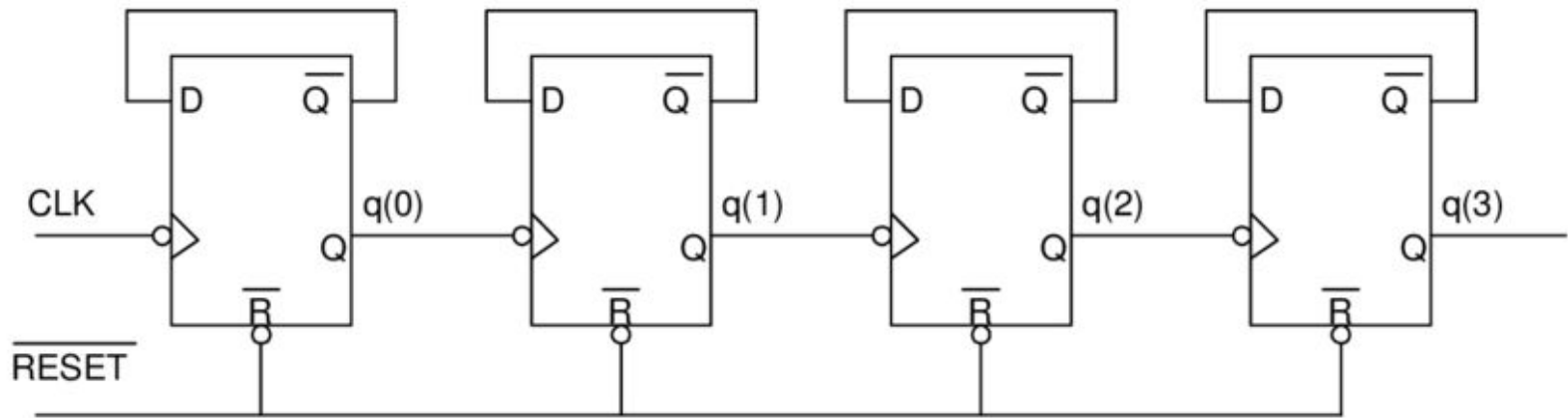
Chapter

9

- Verilog 複習 : Behavior-Level
- 同步/非同步計數器
Asynchronous / Synchronous counter
- 漣波計數器 **Ripple carry counter**
- 四位元同步計數器 4_bit Synchronous counter
- 上下計數器 Bidirectional counter
- LAB 5-1 and 5-2

Ripple Carry Counter

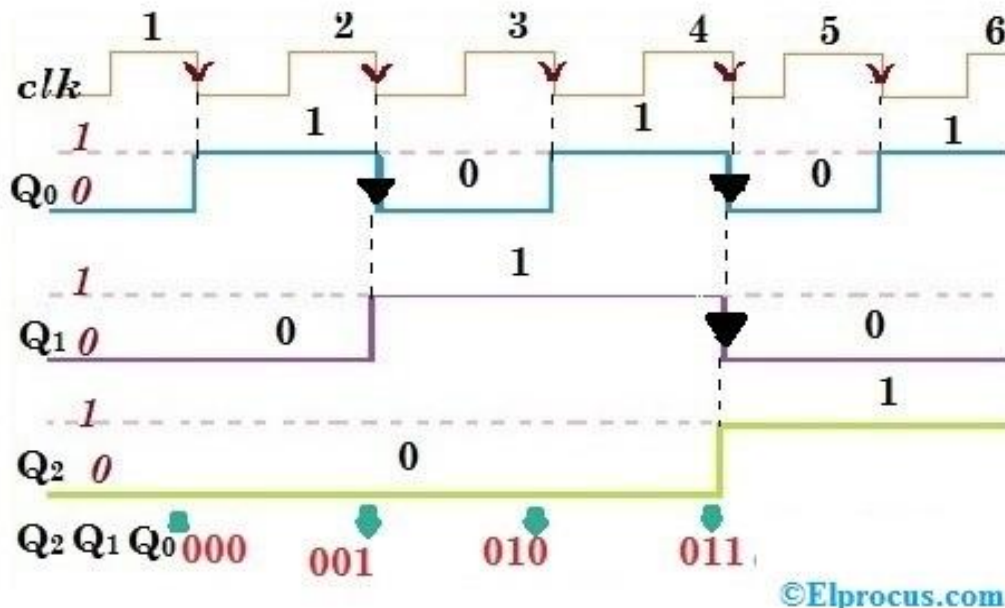
- 鏈波計數器是一種**非同步計數器**，由 D 型正反器推動下一級 D 型正反器產生計數狀態。
- 每級正反器對於脈波的反應都有延遲現象，而且越到後面的正反器延遲越大，最大延遲時間與正反器的個數成正比。



鏈波計數器

Ripple Carry Counter

- 鏈波計數器時脈，每級正反器輸出Q都會除2，組合起來即成為 2^n 計數器。
- 原理：Clock 訊號觸發， $Q = \sim D$ (上一級)



X	Y	Z
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

Ripple Carry Counter

□ 漣波計數器(1:D_Filp Flop):

```
//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
```

Ripple Carry Counter

□ 漣波計數器(2: Circuit):

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

Chapter

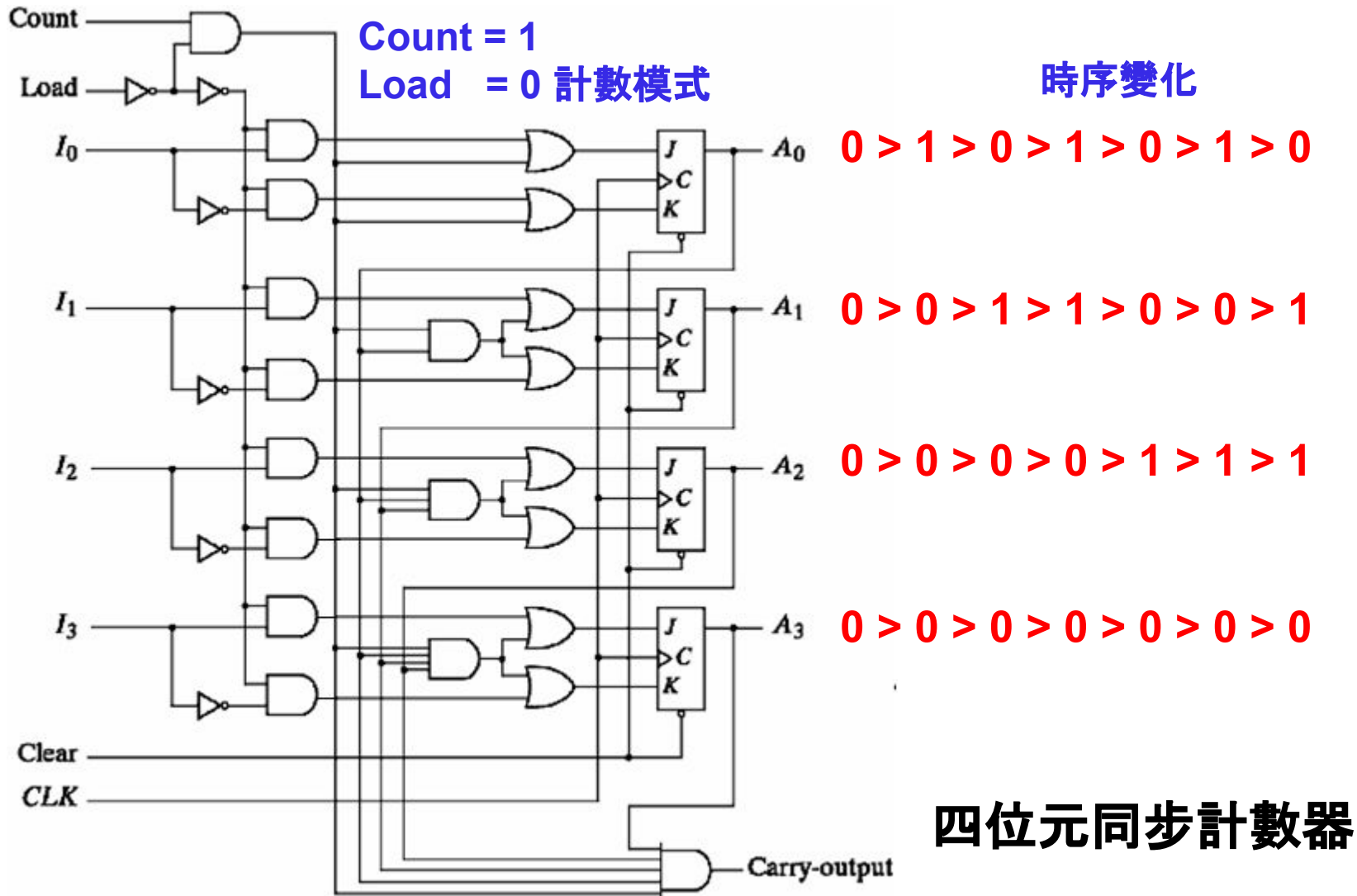
14

- Verilog 複習 : Behavior-Level
- 同步/非同步計數器
Asynchronous / Synchronous counter
- 漣波計數器 Ripple carry counter
- 四位元同步計數器 4_bit Synchronous counter
- 上下計數器 Bidirectional counter
- LAB 5-1 and 5-2

4bit Synchronous counter

- 時脈 CLK 為正緣觸發。
- 當Clear為1且CLK為正緣時，即將計數器之輸出清除為0000。
- Load為同步載入，可設定計數器初始值。當訊號為1時，在CLK正緣觸發時將四個輸入資料 I_0 、 I_1 、 I_2 、 I_3 載入進四個JK正反器， $A_0 \sim A_3$ 為 $I_0 \sim I_3$ 。
- 計數器之Count為1、Load為0，計數器才能正常計數(上數)。
- 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(等於下一級的 JK 控制訊號)。

4bit Synchronous counter



4bit Synchronous counter

- 四位元同步計數器之邏輯總結：

- ※ A_0 轉態: Count

- ※ A_1 轉態: $\text{Count} * A_0$

- ※ A_2 轉態: $\text{Count} * A_0 * A_1$

- ※ A_3 轉態: $\text{Count} * A_0 * A_1 * A_2$

- Count訊號為1、Load為0，計數器上數，數至1111時，所有輸出皆為1，下個CLK全部轉態，輸出回到0000重複向上數，週期16個CLK。

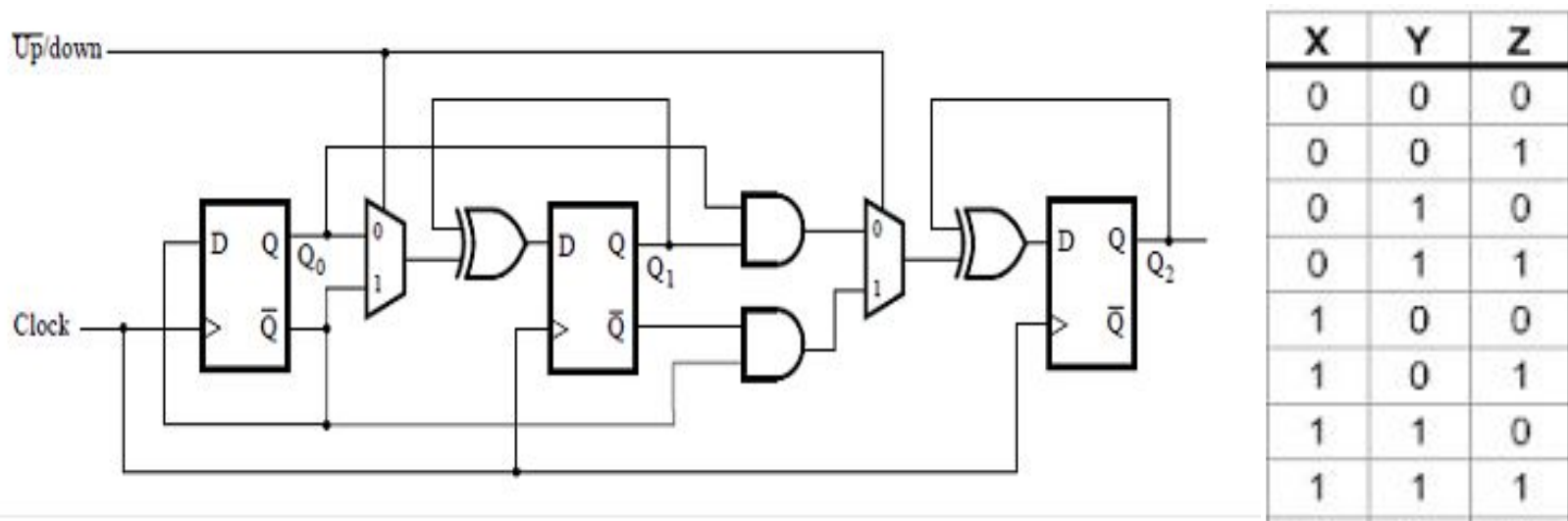
Chapter

18

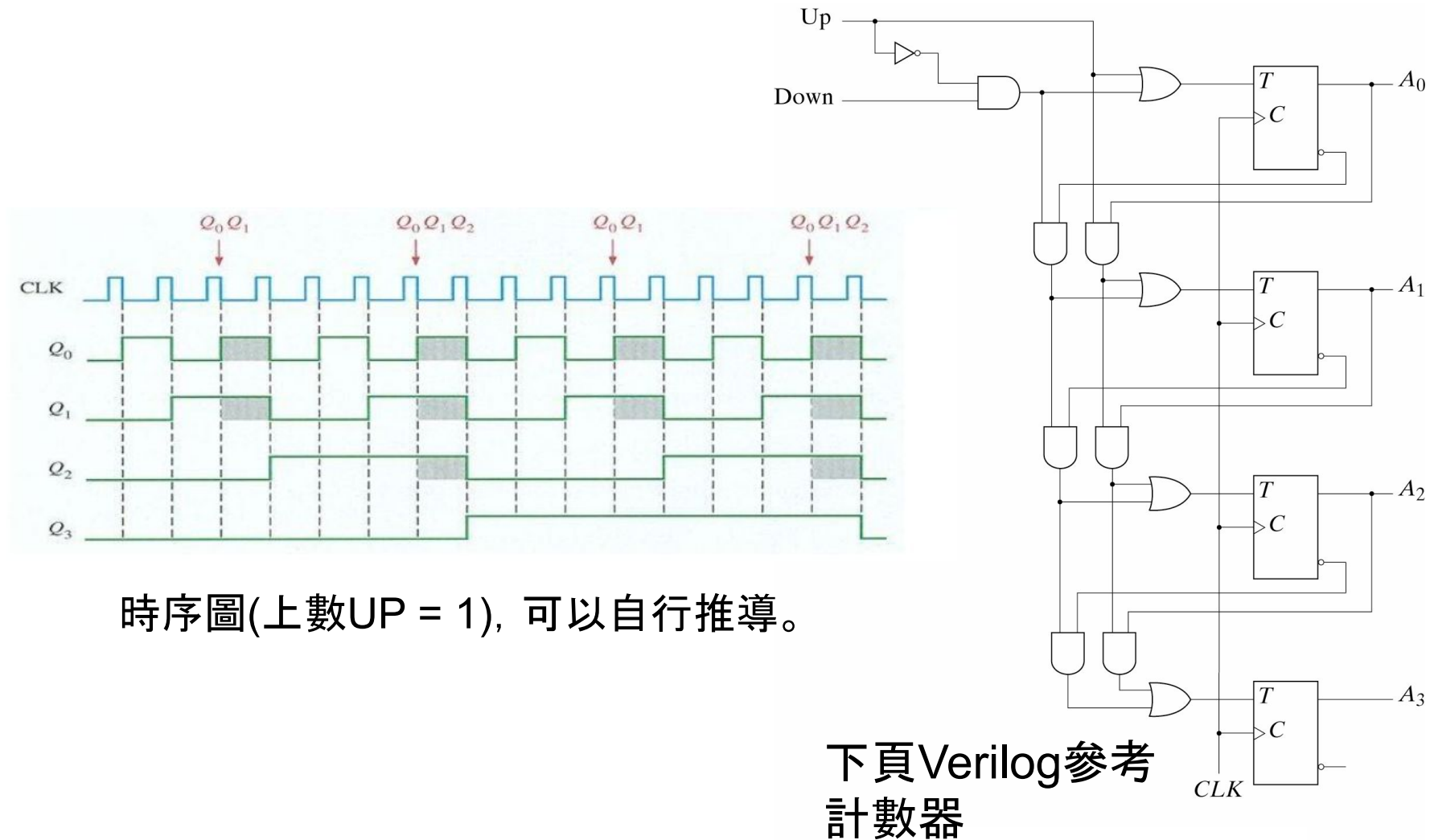
- Verilog 複習 : Behavior-Level
- 同步/非同步計數器
Asynchronous / Synchronous counter
- 漣波計數器 Ripple carry counter
- 四位元同步計數器 4_bit Synchronous counter
- **上下計數器** **Bidirectional counter**
- LAB 5-1 and 5-2

Bidirectional Counter

- 雙向計數器(UP/DOWN)為一種同步計數器，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。
- 下圖為D型上下計數器，UP/DOWN控制線決定上下數，



Bidirectional Counter



Bidirectional Counter

```
// 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
```

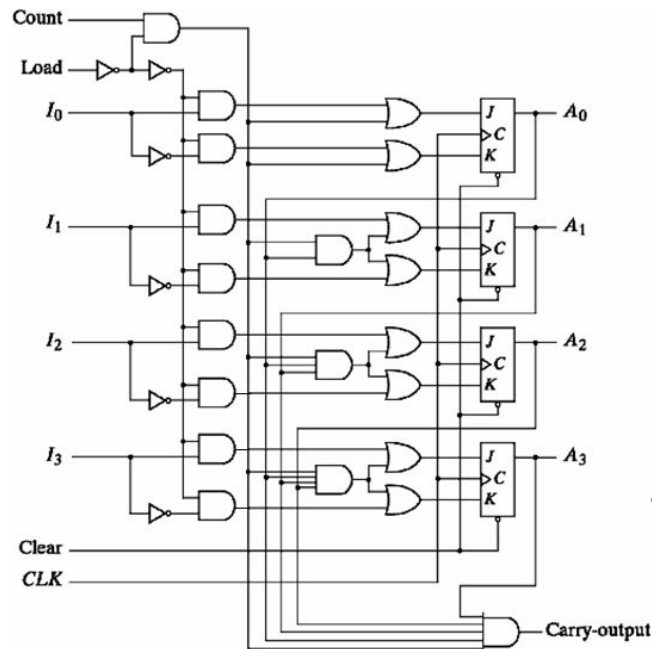
Chapter

22

- Verilog 複習 : Behavior-Level
- 同步/非同步計數器
Asynchronous / Synchronous counter
- 漣波計數器 Ripple carry counter
- 四位元同步計數器 4_bit Synchronous counter
- 上下計數器 Bidirectional counter
- **LAB 5-1 and 5-2**

LAB 5-1

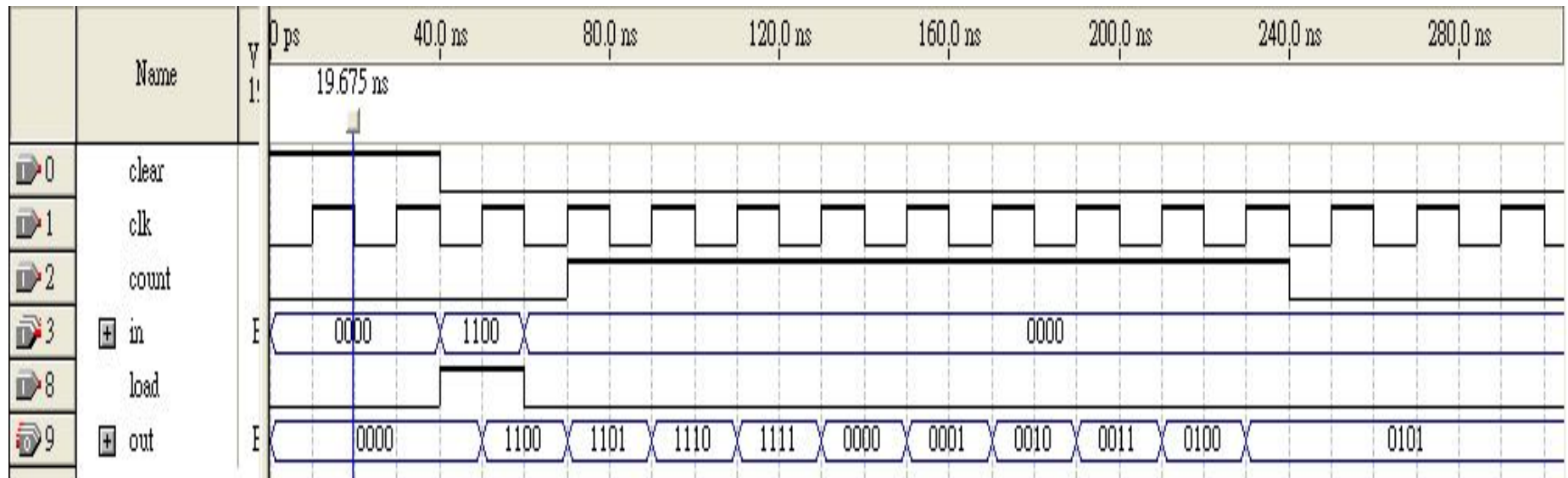
- 使用Verilog HDL描述出四位元同步計數器，於Quartus II 模擬訊號波型加以驗證結果！
- 需要包含 Count、Load、[3:0]I 等。
- 可使用任意Model撰寫(DataFlow & Behavior)



LAB 5-1

24

波行模擬結果

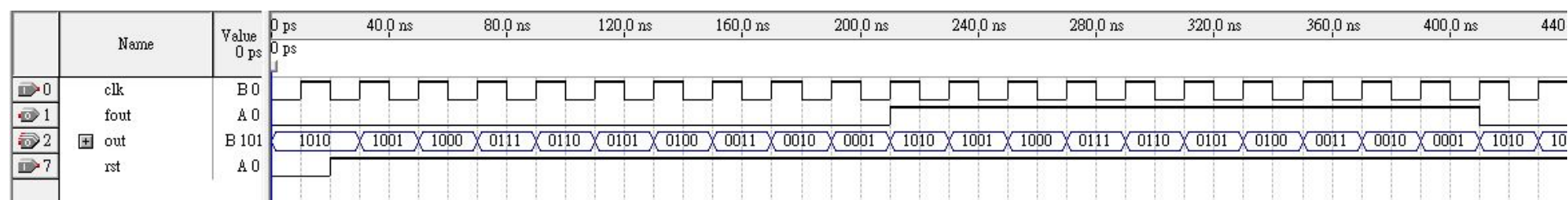


LAB 5-2

- 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 1010 數至 0001 時, $f_out = \sim f_out$, 使輸出 f_out 之輸出週期為 CLK 週期之 20 倍。
- $q[3:0]$ 再由 1010 繼續下數, 此即為除以 20 之除頻器。

LAB 5-2

26



LAB 5-2

State table

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

LAB 5

28

下課前繳交至moodle：

上傳verilog.v

波形截圖