



正確學會

改訂新版 デジタル回路と Verilog HDL

Verilog

的16堂課

第 11 章

致能控制的電路設計

本投影片（下稱教用資源）僅授權給採用教用資源相關之旗標書籍為教科書之授課老師（下稱老師）專用，老師為教學使用之目的，得摘錄、編輯、重製教用資源（但使用量不得超過各該教用資源內容之80%）以製作為輔助教學之教學投影片，並於授課時搭配旗標書籍公開播放，但不得為網際網路公開傳輸之遠距教學、網路教學等之使用；除此之外，老師不得再授權予任何第三人使用，並不得將依此授權所製作之教學投影片之相關著作物移作他用。

本章重點

先前我們利用時脈輸入來控制電路運作，但由於系統時脈是固定的，我們無法於任意時間點操控資料的變化。本章我們將導入具備致能 (Enable) 控制的數位電路設計，讓電路可以配合系統時脈做同步，也能依照我們的需求來控制電路功能的變化。

- 11.1 附有致能的暫存器
- 11.2 附有致能的移位暫存器
- 11.3 附有致能的計數器
- 11.4 附有致能的上下數計數器
- 11.5 附有讀取、致能的上下數計數器

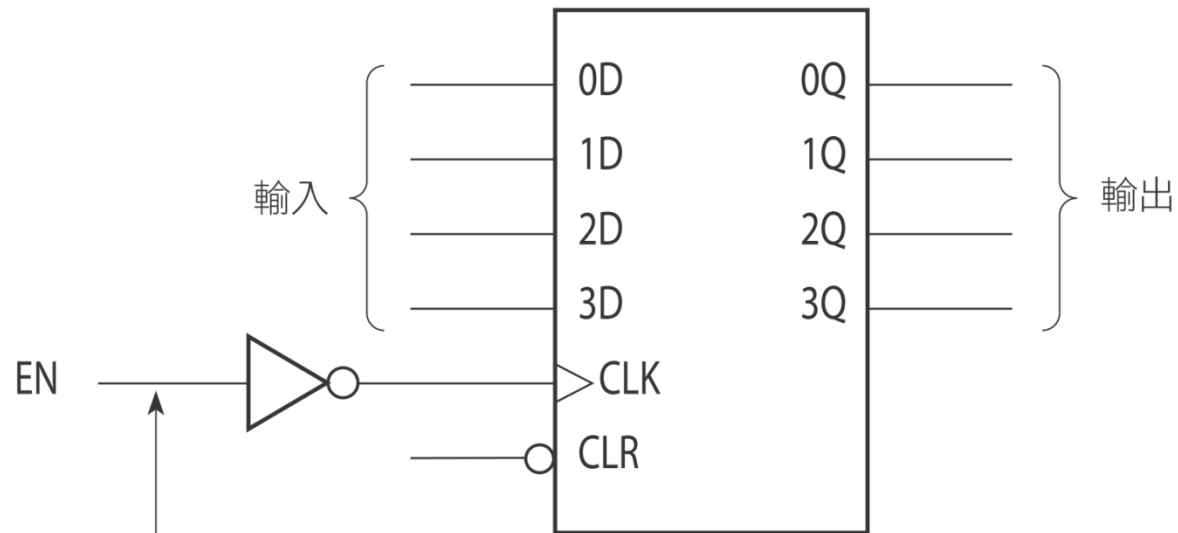
同步電路的特性

- (a) 所有正反器的「CLK」都跟「時脈」接在一起
- (b) 所有正反器的 'S' 都不使用
- (c) 所有正反器的 'R', 不是不使用就是接到重置

11.1 附有致能的暫存器

- 將 7.1 節介紹的暫存器電路改成將 EN 做為訊號輸入，「EN」為 ON 的時候才會記憶 D 的樣式

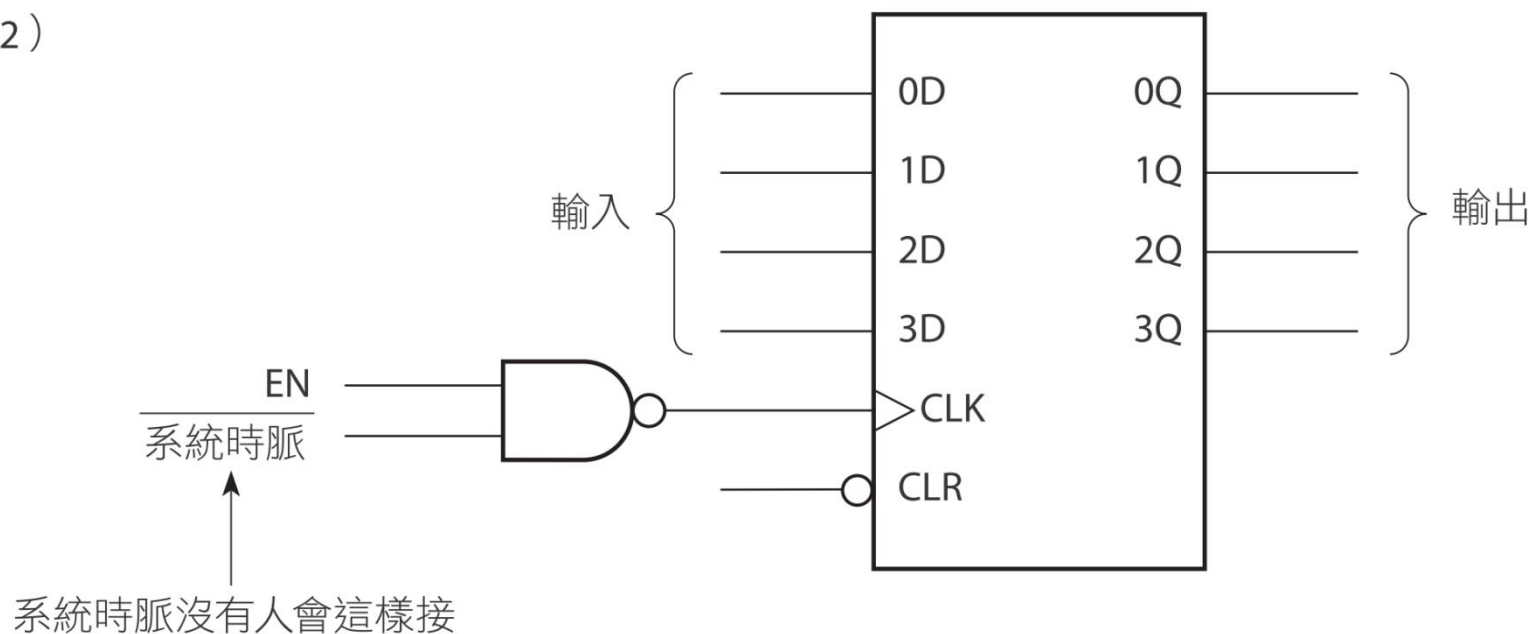
(1)



EN 為負緣的時候會記憶輸入的資料

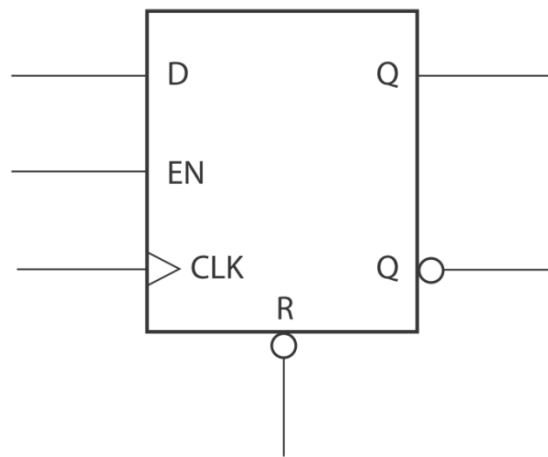
- 但是這樣的電路不能稱作同步的。因為暫存器的CLK 沒有跟系統時脈連接在一起的關係。

(2)

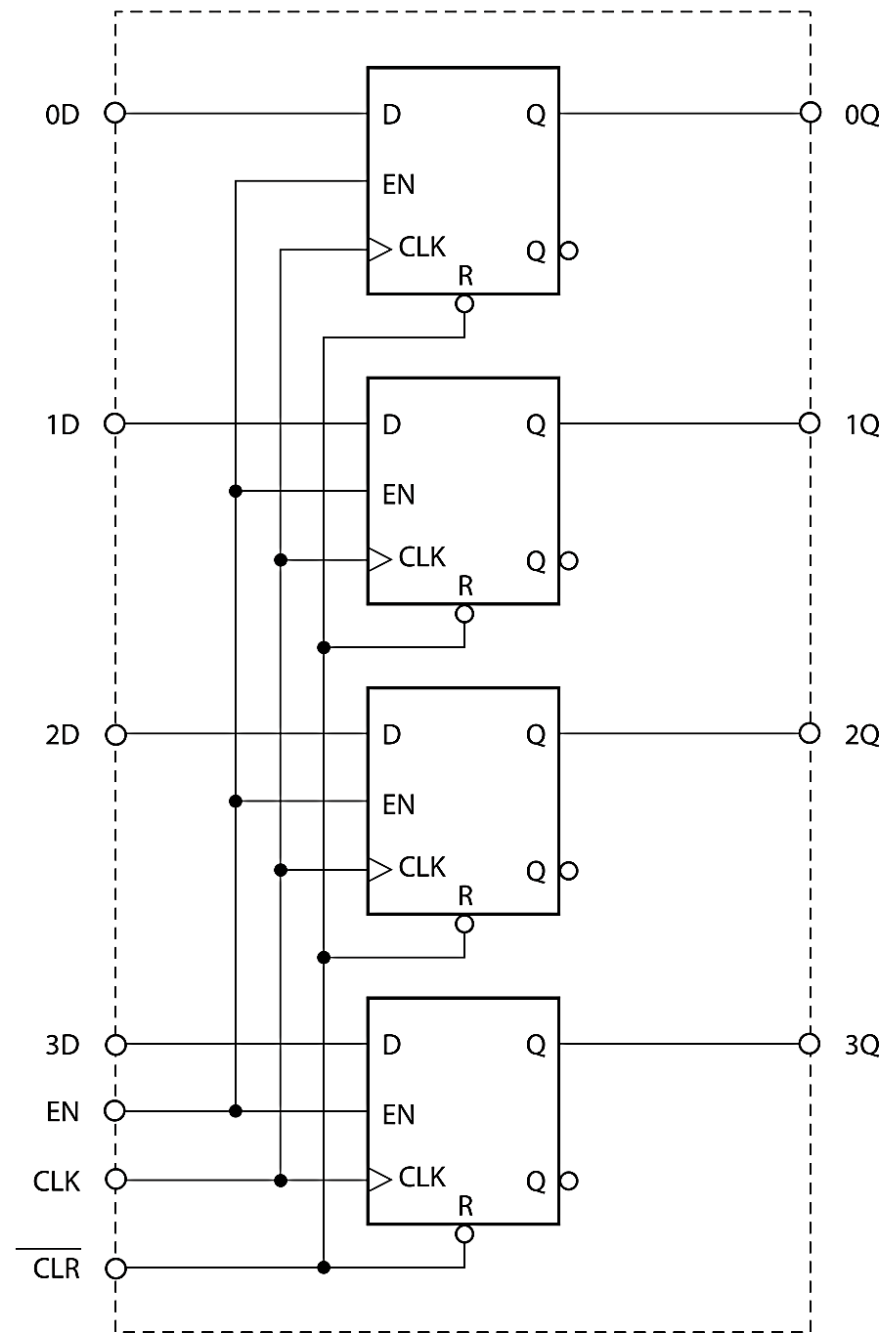


11.1.1 附有致能的正反器

- 要解決前述問題，就需要「附有致能的正反器」
- 當時脈正緣且「EN」為 '1' 的時候 $D = \text{「輸入」}$ 所以會記憶新的值，「EN」='0' 的話 $D = Q$ 所以輸出沒有變化
- 「EN」為致能 (enable: 變可能) 的簡稱，當它為 ON 的時候就會記憶新的資料



▲ 圖 11.4 附有致能的正反器的電路符號



▲ 圖 11.5 附有致能的暫存器

11.1.2 附有致能的暫存器的 Verilog HDL 描述

程式 11.2 附有致能的暫存器的 Verilog HDL 描述

LST11_2.V

```
/*          EN _ REG4 */
module      EN _ REG4 ( CLR _ B, D, CLK, EN, Q );
input       CLR _ B, CLK, EN;
input       [3:0] D;
output      [3:0] Q;

reg         [3:0] Q;
always      @( posedge CLK or negedge CLR _ B )
    if ( !CLR _ B )
        Q <= 0;
    else if ( EN )
        Q <= D;
endmodule
```

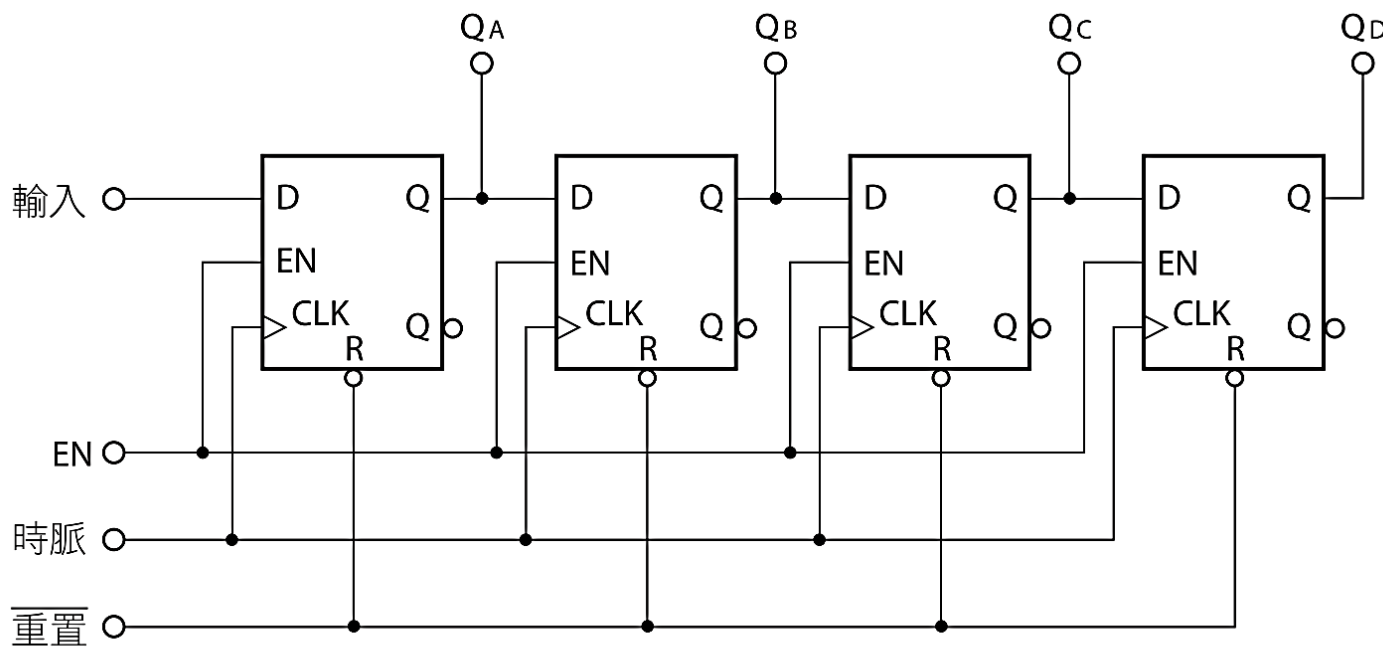
接下頁

11.2 附有致能的移位暫存器

- 移位暫存器是每當時脈輸入的時候就會做移位的電路
- 附上致能的移位暫存器, 就可以在任意時機點對資料做移位

11.2.1 串列輸入並列輸出移位暫存器

- 當時脈正緣的時候且「EN」為 ON 的話移位 1 個位元, OFF 的話則沒有變化
- 各正反器的「EN」都一起連接到移位暫存器的「EN」上, D 跟前一級的 Q 連接



時脈正緣的時候且 EN= '1' 的話做移位動作, EN= '0' 的話則沒有變化

11.2.2 串列輸入並列輸出移位暫存器的 Verilog HDL 描述

程式 11.3 附有致能的串列輸入並列輸出移位暫存器的 Verilog HDL 描述

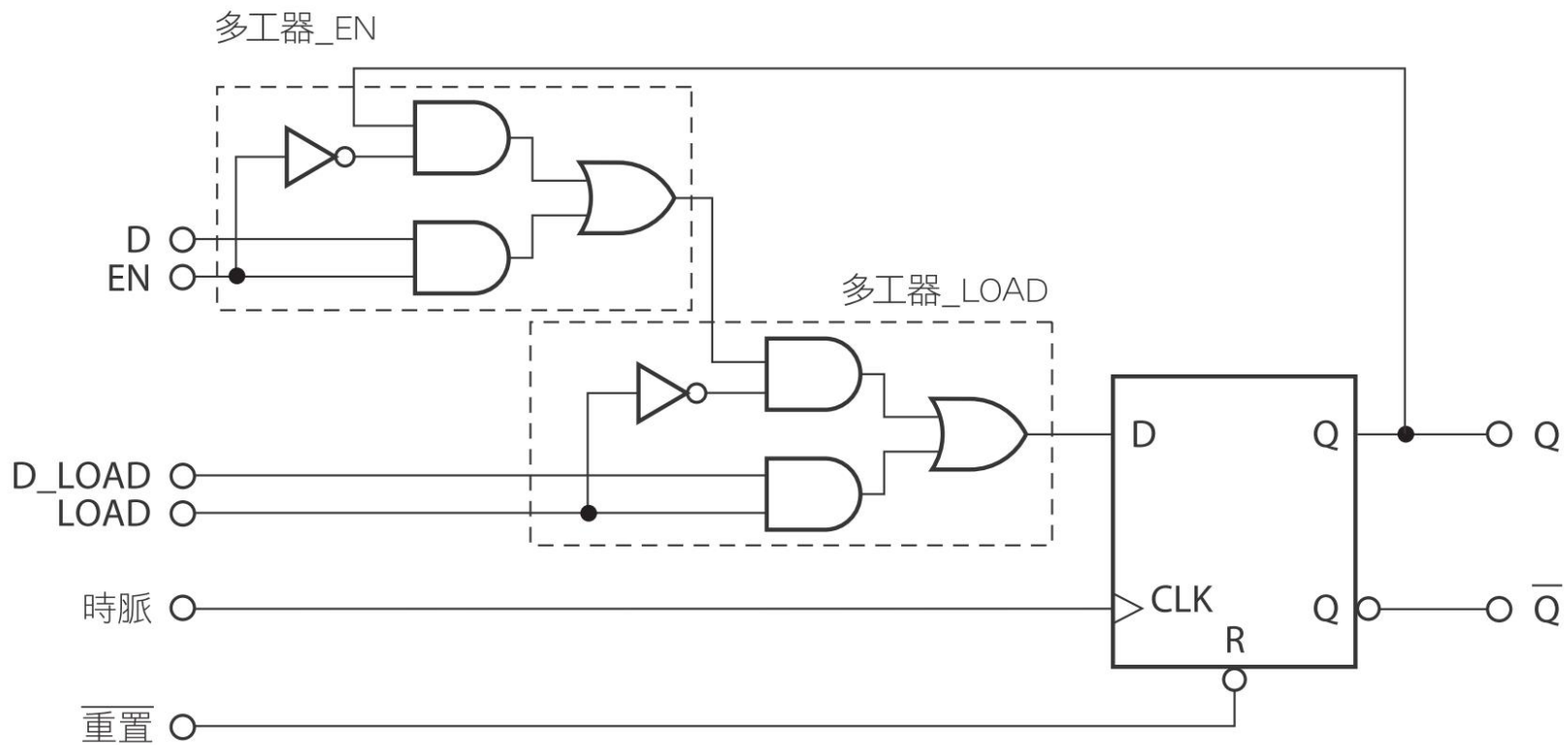
LST11_3.V

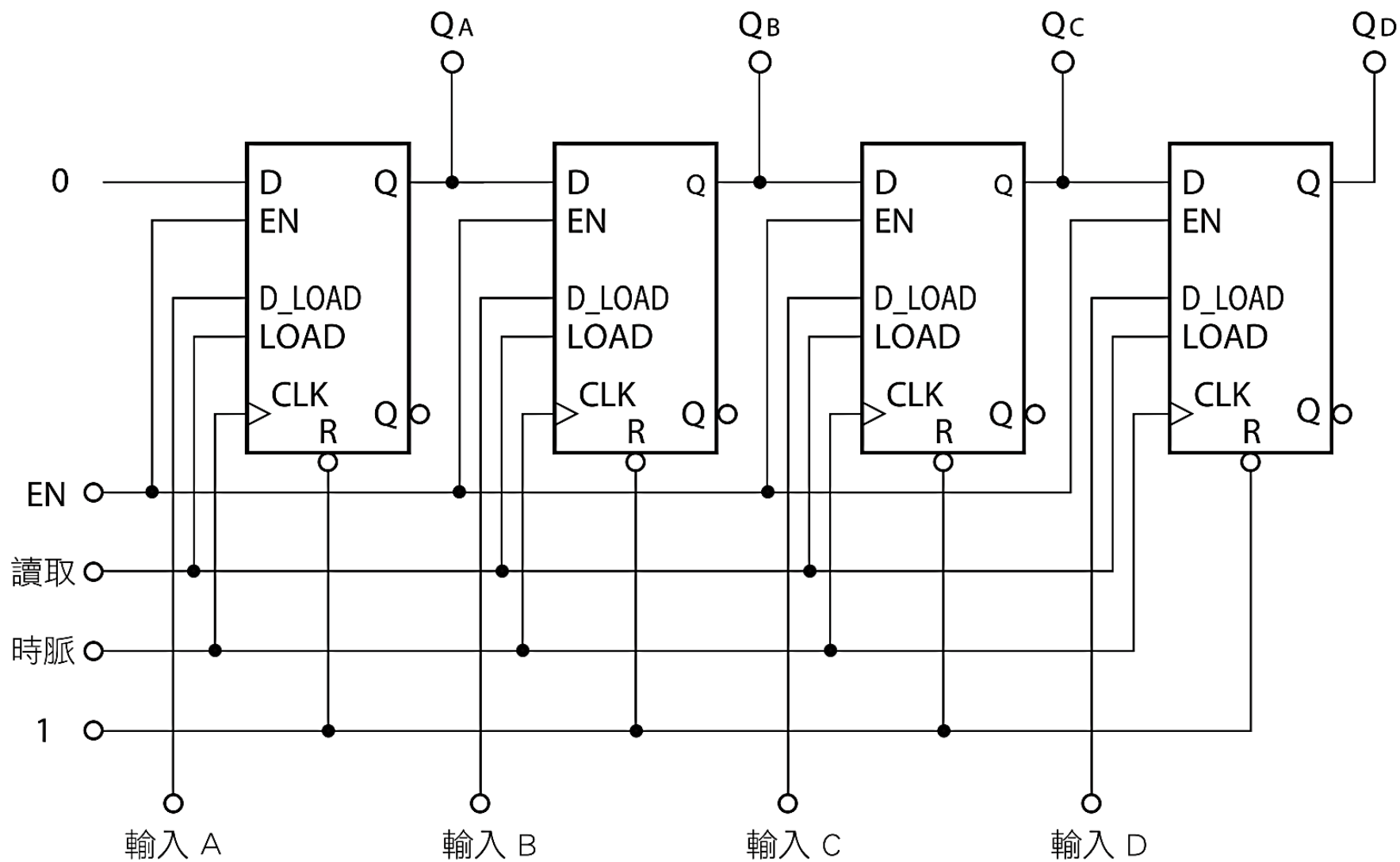
```
/*      EN _ SIN _ POUT _ SHIFT      */
module  EN _ SIN _ POUT _ SHIFT      ( RESET _ B, IN, CLK, EN, Q );
input   RESET _ B, CLK, IN, EN;
output  [3:0] Q;
reg     [3:0] Q;
      always @( posedge CLK or negedge RESET _ B )
            if ( !RESET _ B )
                  Q <= 0;
            else if ( EN )
                  Q <= {Q,IN};
endmodule
```

11.2.3 並列輸入串列輸出移位暫存器

- 時脈正緣的時候且「讀取」為 ON 的話
→ 以輸入來設定正反器
- 時脈正緣的時候且「EN」為 ON 的話
→ 以前一級的值來設定正反器
- 電路設計要在輸入 D 的前面設置了 2 級的多工器，根據「LOAD」與「EN」來控制正反器的下一個狀態

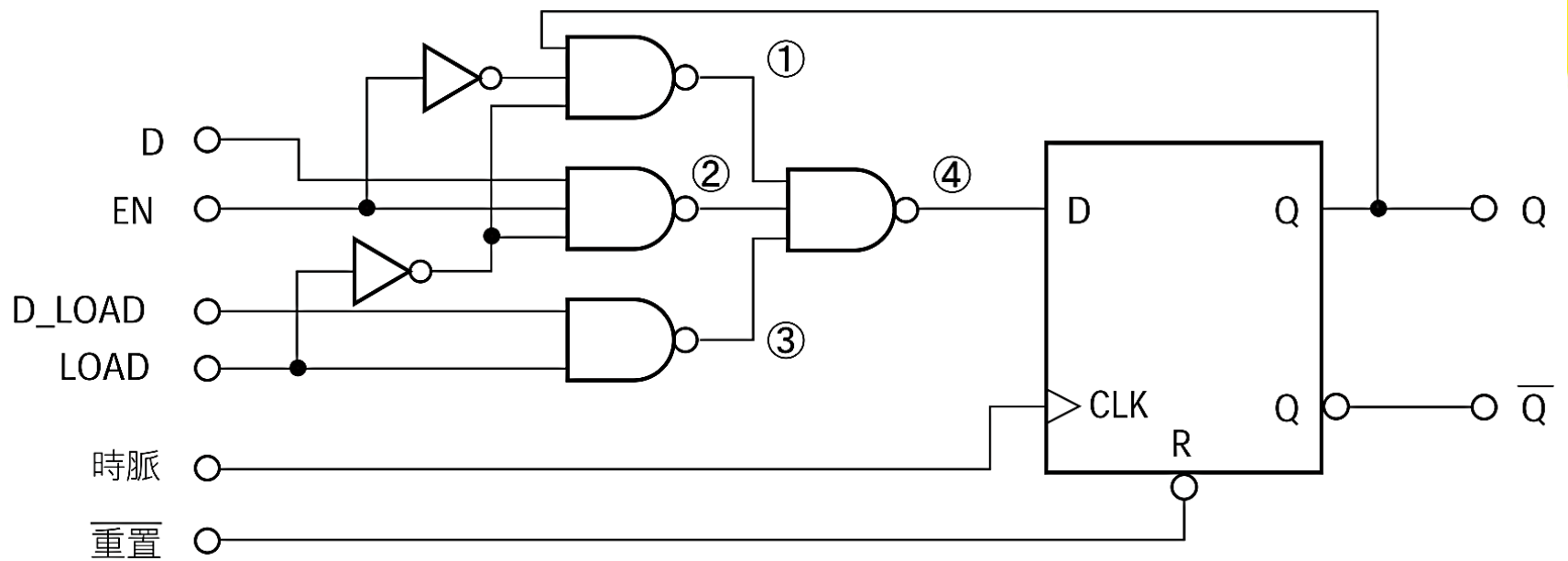
(a) 電路圖





11.2.4 用 2 級邏輯做成的並列輸入串列輸出移位暫存器

- 前面的電路是在正反器輸入加上兩級多工器構成，不算組合電路 2 級邏輯，會有傳遞延遲的問題
- 前述的移位暫存器改成 NAND 電路如下頁圖：



$$\textcircled{1} = \overline{\overline{\text{LOAD}} \cdot \overline{\text{EN}} \cdot \overline{Q}}$$

$$\textcircled{2} = \overline{\overline{\text{LOAD}} \cdot \text{EN} \cdot D}$$

$$\textcircled{3} = \overline{\text{LOAD} \cdot D_LOAD}$$

$$\textcircled{4} = \overline{\overline{\text{LOAD}} \cdot \overline{D_LOAD} \cdot \overline{\overline{\text{LOAD}} \cdot \text{EN} \cdot D} \cdot \overline{\overline{\text{LOAD}} \cdot \overline{\text{EN}} \cdot Q}}$$

$$= \text{LOAD} \cdot D_LOAD + \overline{\overline{\text{LOAD}}} \cdot \text{EN} \cdot D + \overline{\overline{\text{LOAD}}} \cdot \overline{\text{EN}} \cdot Q$$

11.2.5 並列輸入串列輸出移位暫存器的 Verilog HDL 描述

- 要注意 if 裡面的條件式「LOAD」與「EN」有先後順序
- 「LOAD」為 OFF 的時候不代表「EN」就是有效

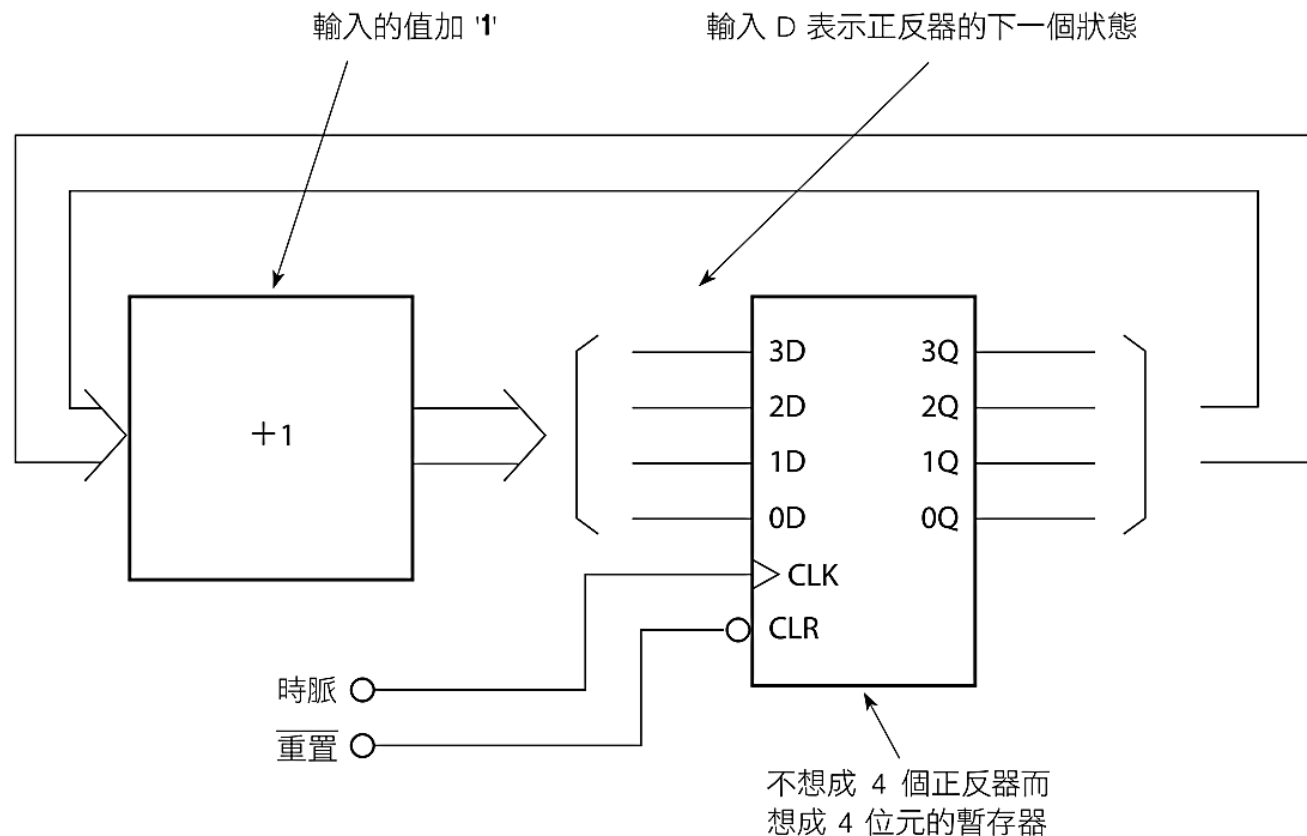
程式 11.4 附有致能的並列輸入串列輸出移位暫存器的 Verilog HDL 描述

LST11_4.V

```
/*          EN_PIN_SOUT_SHIFT      */
module      EN_PIN_SOUT_SHIFT      ( LOAD, IN, CLK, EN, RESET_B, Q );
input       LOAD, CLK, EN, RESET_B;
input       [3:0] IN;
output      [3:0] Q;
reg         [3:0] Q;
always      @( posedge CLK or negedge RESET_B )
    if      ( !RESET_B )
        Q <= 0;
    else if ( LOAD )
        Q <= IN;
    else if ( EN )
        Q <= Q << 1;
endmodule
```

11.3 附有致能的計數器

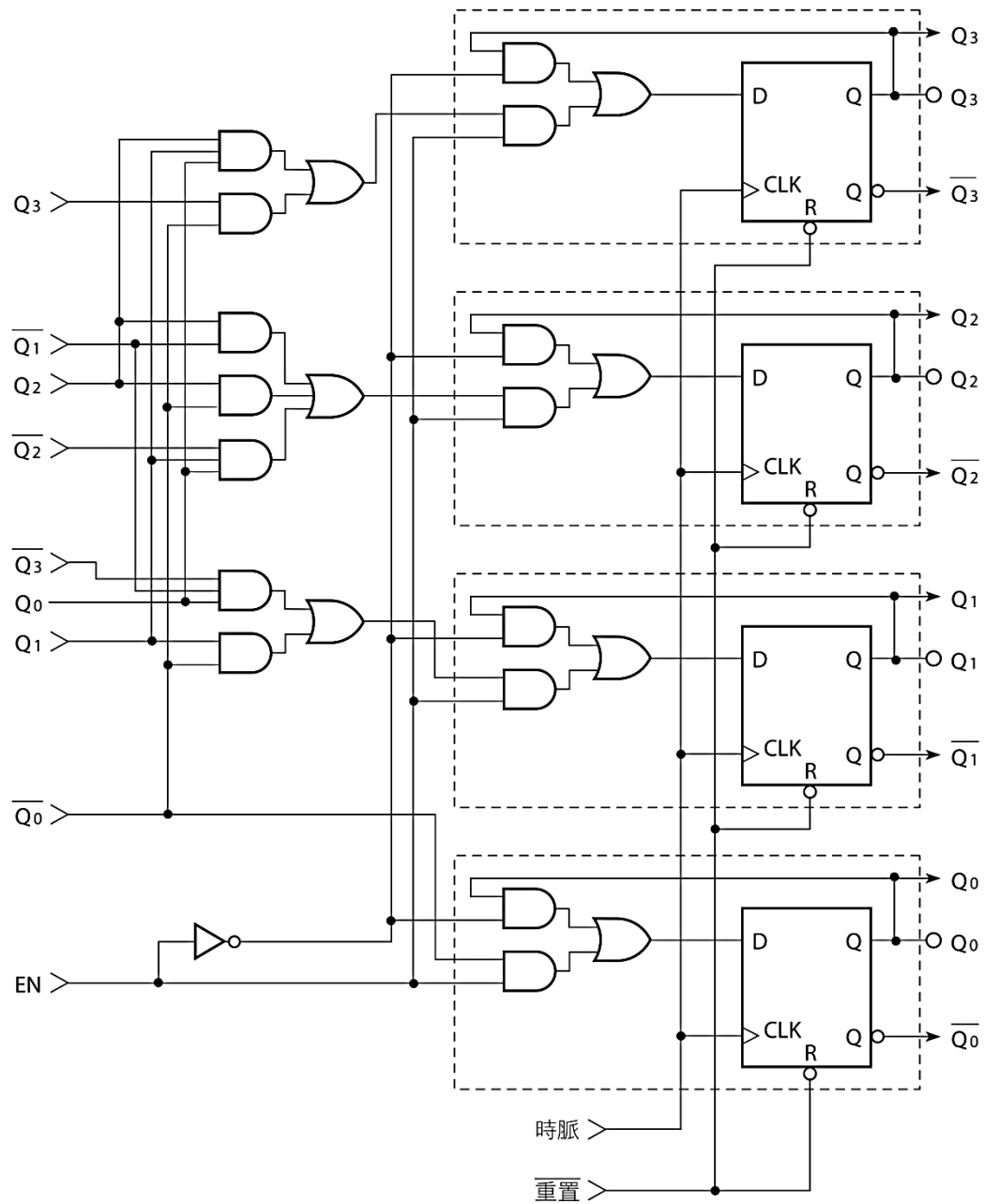
11.3.1 「+1 電路」與暫存器



▲ 圖 11.12 mode-10 上數計數器

- 根據輸入 D 的值來決定 D 型正反器的下個狀態 Q' 的結果

- 上述電路動作, 可以在「+1 電路」的後面設置一個多工器控制 D 即可
 - ▣ 「EN」='0', D=「正反器」的輸出所以正反器沒有變化
 - ▣ 「EN」='1', D=「+1 電路」的輸出所以會遞增



▲圖 11.15 附有致能的 mode-10 上數計數器

11.3.2 附有致能的計數器的 Verilog HDL 描述

- 「時脈」正緣且「EN」為 ON，計數值不為 '9' 做遞增，為 '9' 的時候計數值變為 '0'

程式 11.6 附有致能的 mode-10 上數計數器的 Verilog HDL 描述

LST11_6.V

```
/*          EN_CNT10          */
module      EN_CNT10      ( RESET_B, CLK, EN, Q );
input       RESET_B, CLK, EN;
output      [3:0] Q;
reg         [3:0] Q;

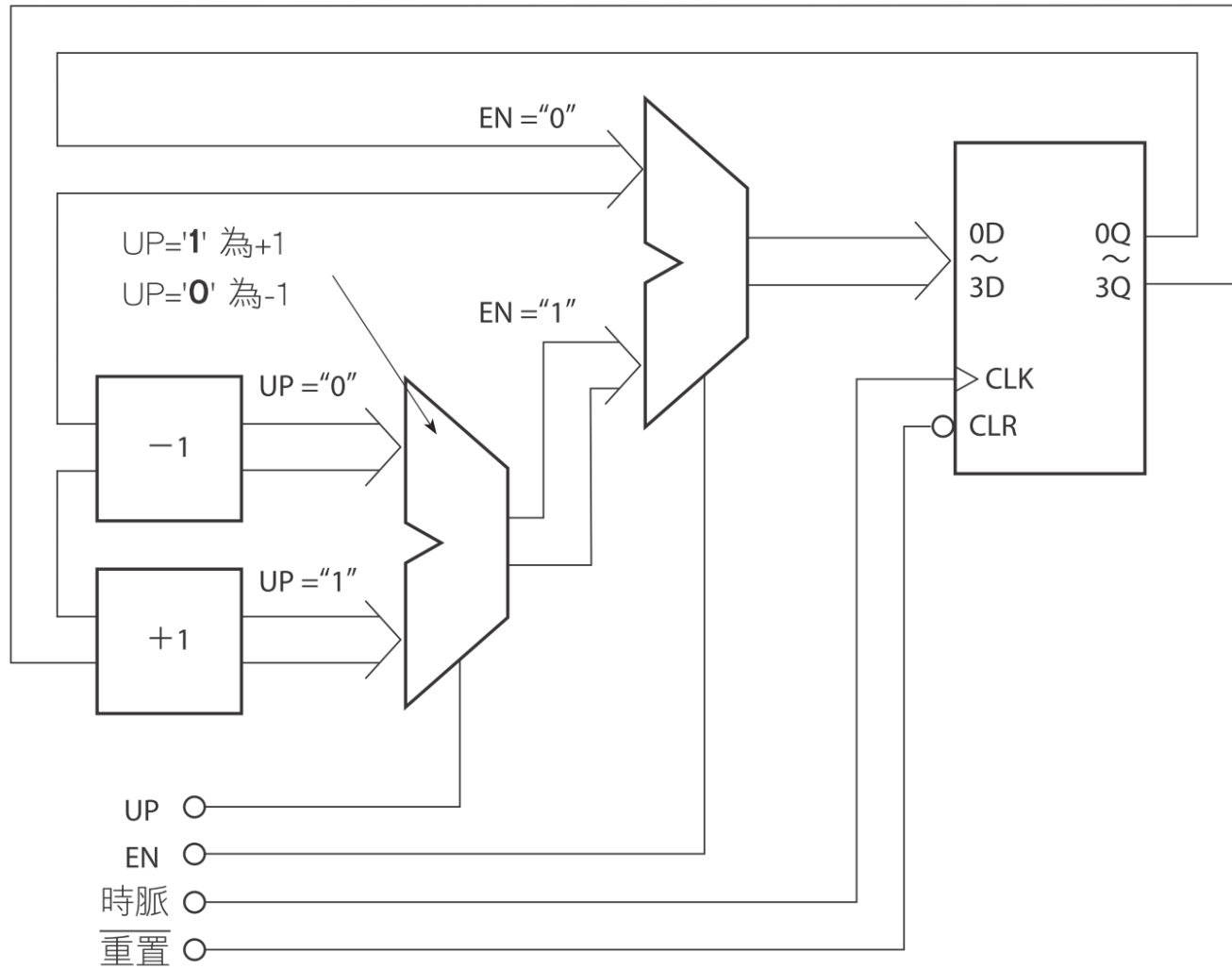
    always @( posedge CLK or negedge RESET_B )
        if      ( !RESET_B )
            Q <= 0;
        else if      ( EN )
            if ( Q == 9 )
                Q <= 0;
            else
                Q <= Q + 1;

endmodule
```

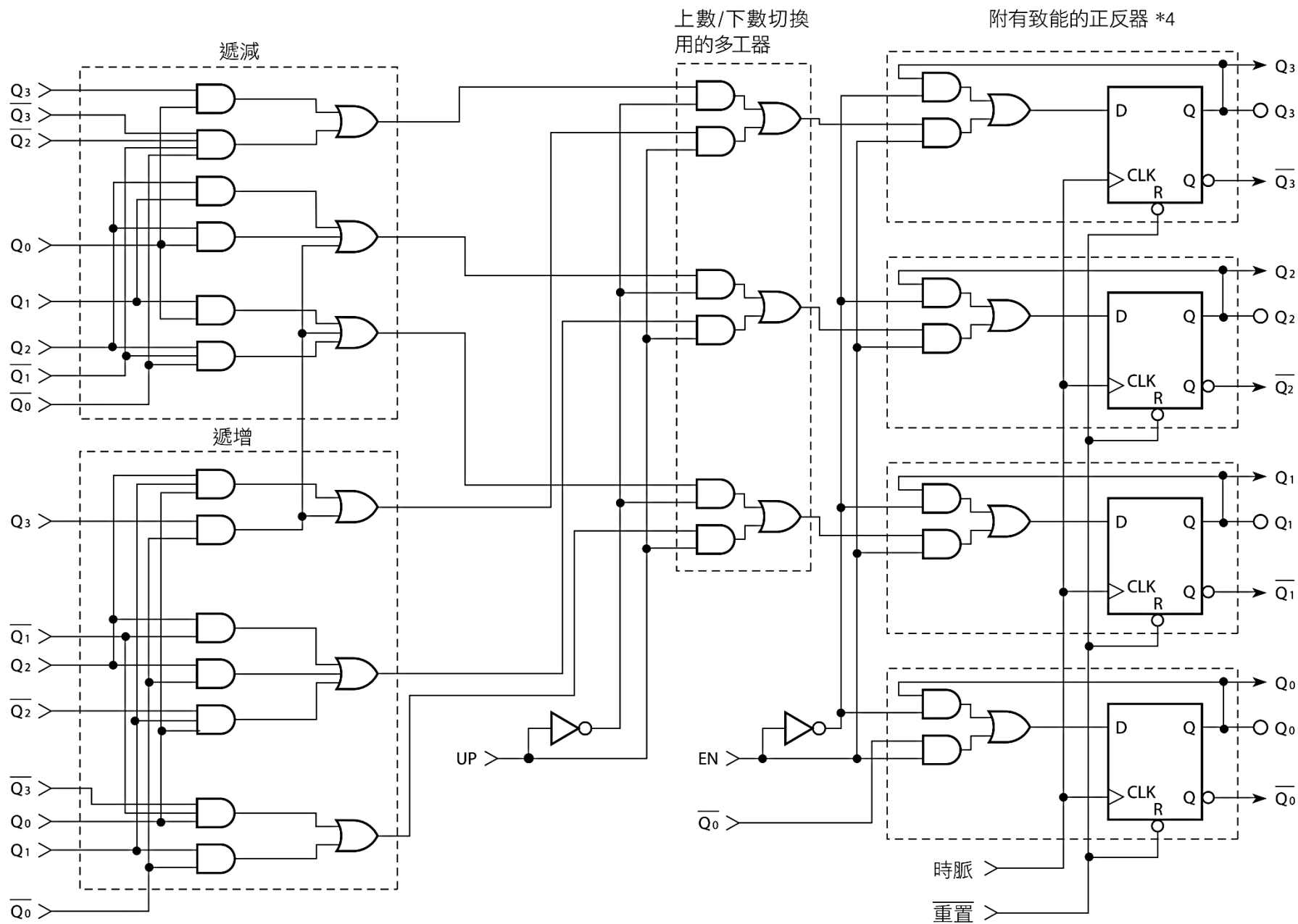
接下頁

11.4 附有致能的上下數計數器

- 根據「UP」來做 +1/-1, 附有致能的 mode-10 上下數計數器
 - ▣ $EN='0'$, $D=$ 「正反器的輸出」所以沒有變化
 - ▣ $EN='1'$ 且「UP」='1', $D=$ 「+1 電路」輸出會遞增
 - ▣ $EN='1'$ 且「UP」='0', $D=$ 「-1 電路」輸出會遞減



▲ 圖 11.17 附有致能的 mode-10 上下數計數器



▲ 圖 11.18 附有致能的 mode-10 上下數計數器

11.4.2 附有致能的上下數計數器的 Verilog HDL 描述

- EN為 ON 且UP為 ON 的時候遞增
- EN為 ON 且UP為 OFF 的時候遞減
- 遞減時計數值 '0' 的時候會變為 '9'

程式 11.8 附有致能的 mode-10 上下數計數器的 Verilog HDL 描述

LST11_8.V

```

/*          EN_UDCNT10    */
module      EN_UDCNT10    ( RESET_B, CLK, EN, UP, Q );
input       RESET_B, CLK, EN, UP;
output      [3:0] Q;
reg         [3:0] Q;

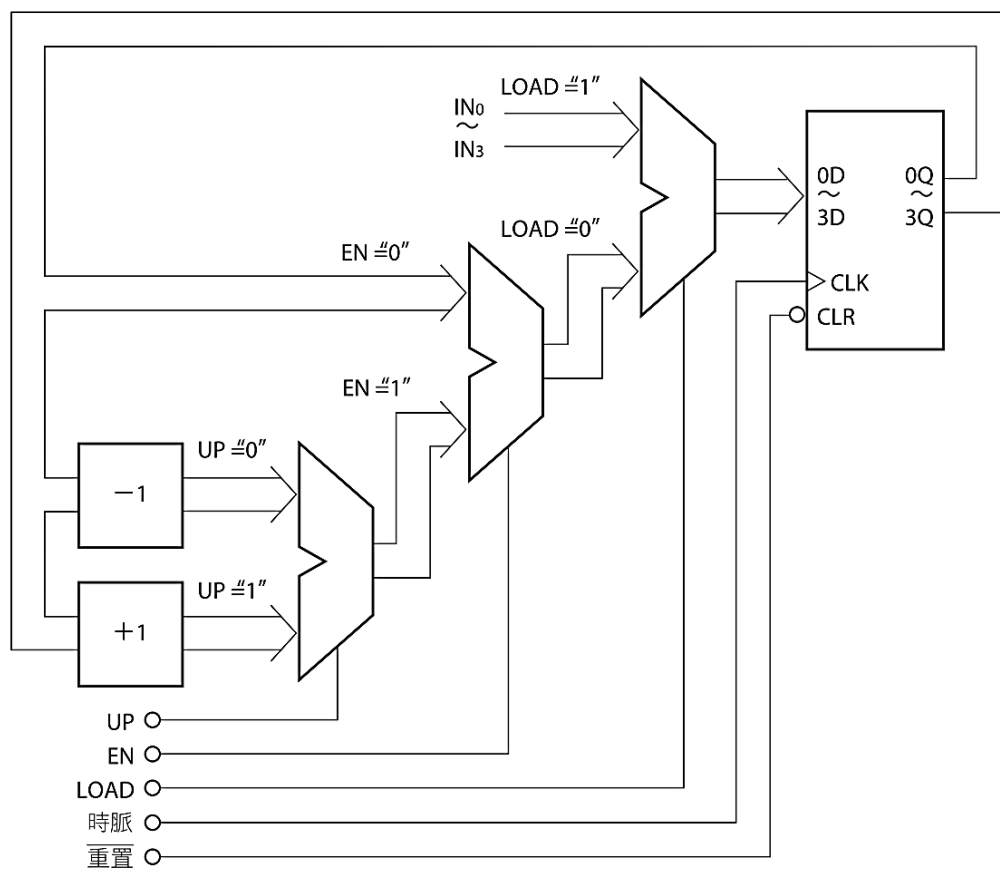
    always  @( posedge CLK or negedge RESET_B )
        if ( !RESET_B )
            Q <= 0;
        else if ( EN )
            if ( UP )                // COUNT UP
                if ( Q == 9 )
                    Q <= 0;
                else
                    Q <= Q + 1;
            else                      // COUNT DOWN
                if ( Q == 0 )
                    Q <= 9;
                else
                    Q <= Q - 1;

endmodule

```

11.5 附有讀取、致能的上下數計數器

- 時脈為正緣且「LOAD」為 ON，
各正反器透過「IN 0 ~ IN 3」來設定



▲ 圖 11.20 附有讀取, 致能的 mode-10 上下數計數器

11.5.2 附有讀取、致能的上下數計數器的 Verilog HDL 描述

- 上述電路因為 XOR 的輸出是 1 的補數, 所以有可能會無法正確判斷「符號的異同」

程式 11.10 附有讀取, 致能的 mode-10 上下數計數器的 Verilog HDL 描述 LST11_10.V

```
/*      LD_EN_UDCNT10      */
module LD_EN_UDCNT10      ( RESET_B, CLK, LOAD, EN, UP, IN, Q );
input  RESET_B, CLK, LOAD, EN, UP;
input  [3:0] IN;

output [3:0] Q;
reg       [3:0] Q;
always @( posedge CLK or negedge RESET_B )
    if      ( !RESET_B )
        Q <= 0;
    else if ( LOAD )
        Q <= IN;
    else if ( EN )
        if ( UP )                // COUNT UP
            if ( Q == 9 )
                Q <= 0;
            else
                Q <= Q + 1;
        else                    // COUNT DOWN
            if ( Q == 0 )
                Q <= 9;
            else
                Q <= Q - 1;
endmodule
```

接下頁