

正確學會

改訂新版 ディジタル回路と Verilog HDL

Verilog 的 16 堂課

第 8 章

序向電路 ||

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

用上正反器的序向電路需要理解特性方程式才能進行設計；最基礎的電路設計就是「運算電路」

- 8.1 同步電路的設計
- 8.2 序向電路的重置
- 8.3 最大時脈週期
- 8.4 延伸學習

8.1 同步電路的設計

- 同步電路的設計主要是根據邏輯運算來求出邏輯式，只要理解特性表、卡諾圖、特性方程式就不難了

8.1.1 mode-4 上數計數器的設計

- 下表紀錄電路現在狀態跟下一個狀態的特性表
- mode-4 上數計數器的正反器輸出 $[Q_1, Q_0]$, 下一個狀態就寫成 $[Q_1', Q_0']$

| Q_1 | Q_0 | Q_1' | Q_0' |
|-------|-------|--------|--------|
| 0 | 0 | 0 | 1 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 |

$\brace{ \text{現在的狀態} }$ $\brace{ \text{下一個狀態} }$

▲ 表 8.1 mode-4 上數計數器的特性表

- 根據特性表即可求出 Q_1' , Q_0' 的邏輯式 (不需透過卡諾圖)

$$Q_1' = Q_1 \oplus Q_0$$

$$Q_0' = \overline{Q_0}$$

- 邏輯式

$$Q' = D$$

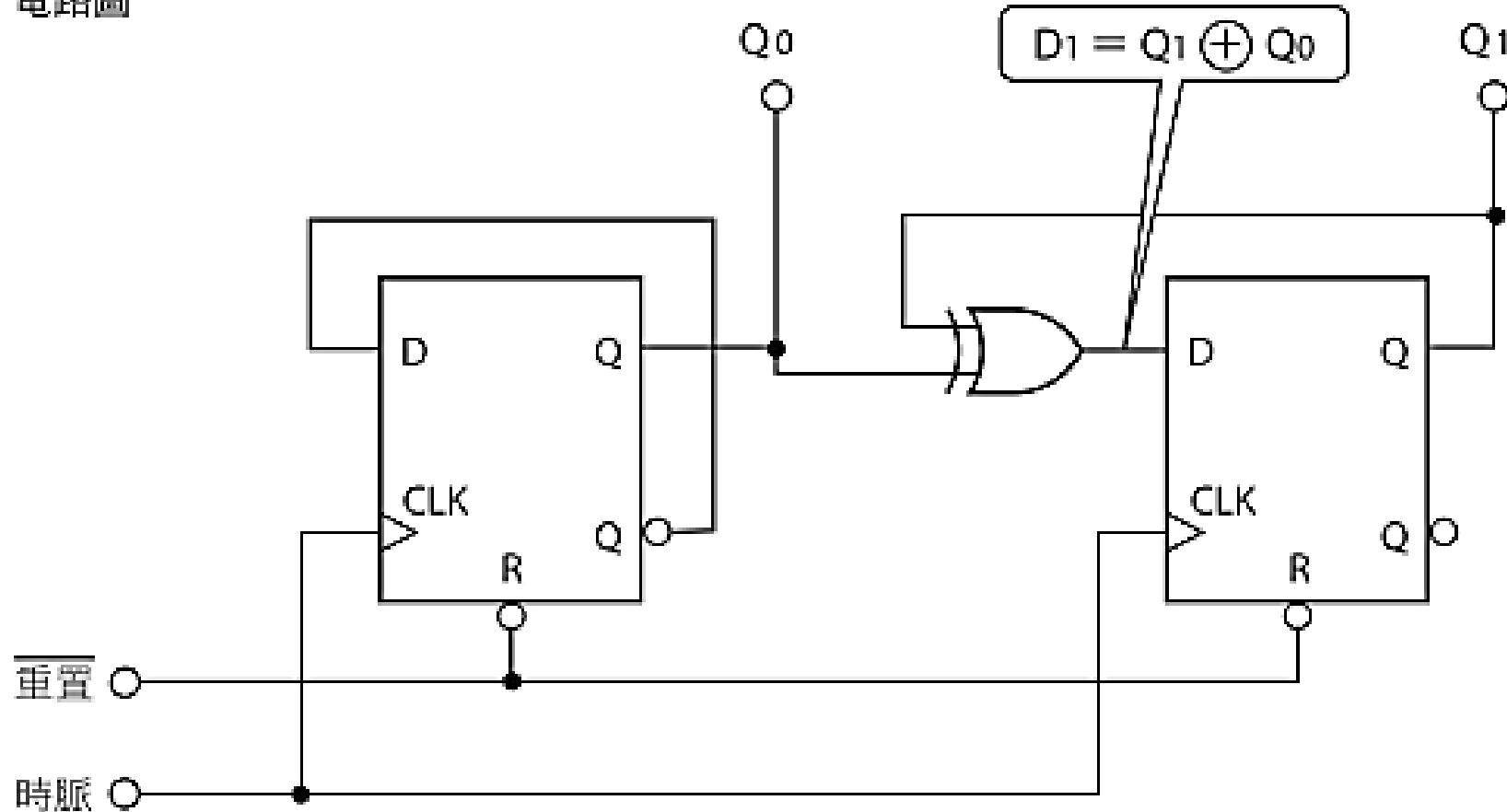


$$D_1 = Q_1 \oplus Q_0$$

$$D_0 = \overline{Q_0}$$

● 電路圖

(a) 電路圖



8.1.2 用 JK 正反器來設計 mode-4 上數計數器

- 邏輯式

$$Q_1' = Q_1 \oplus Q_0$$

$$Q_0' = \overline{Q_0}$$

- JK 正反器的特性方程式為,

$$Q' = J \cdot \overline{Q} + \overline{K} \cdot Q$$

- 可改成：

$$\begin{aligned} Q_0' &= \overline{Q_0} \\ &= 1 \cdot \overline{Q_0} + 0 \cdot Q_0 \end{aligned}$$

或者

$$\begin{aligned} Q_0' &= \overline{Q_0} \\ &= \overline{Q_0} \cdot \overline{Q_0} + \overline{Q_0} \cdot Q_0 \end{aligned}$$

- 各正反器的輸入「J, K」會變成

Q_1 的 J

$$J_1 = Q_0$$

Q_1 的 K

$$\bar{K}_1 = \bar{Q}_0$$

因此, $K_1 = Q_0$

Q_0 的 J

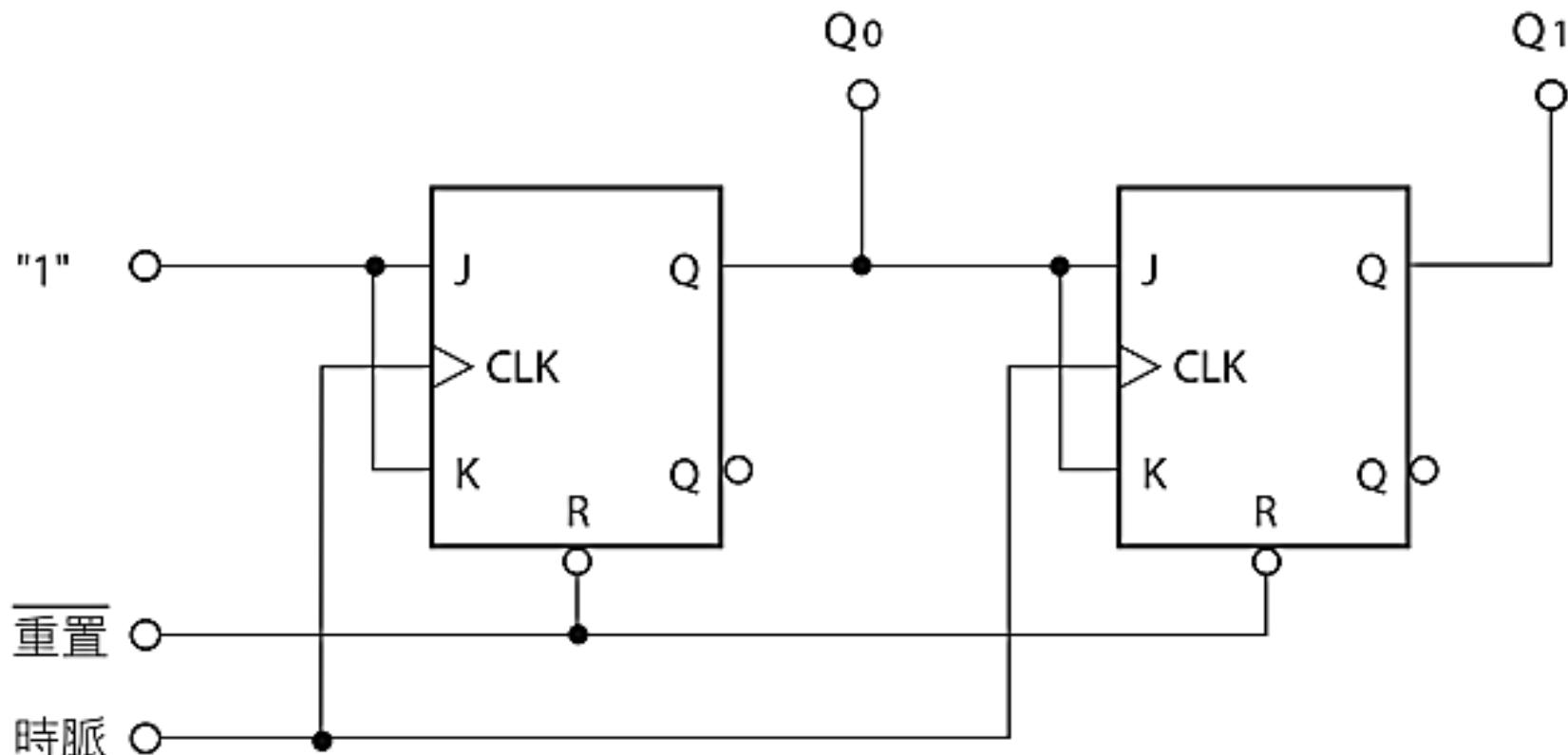
$$J_0 = 1$$

或者 \bar{Q}_0

Q_0 的 K

$$\bar{K}_0 = 0$$

或者 \bar{Q}_0 因此, $K_0 = 1$ 或者 Q_0



▲ 圖 8.1 JK 正反器做成的 mode-4 上數計數器

8.1.3 用 D 型正反器來設計 mode-10 上數計數器

- mode-10 上數計數器的特性表為計數值依序遞增，加到 '9' 的後面接 '0'，計數值不會超過 'A'，

| 計數值 | Q_3 | Q_2 | Q_1 | Q_0 | Q_3' | Q_2' | Q_1' | Q_0' | 計數值 |
|-----|-------|-------|-------|-------|--------|--------|--------|--------|-----|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 2 |
| 2 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 3 |
| 3 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 4 |
| 4 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 5 |
| 5 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 6 |
| 6 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 7 |
| 7 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 8 |
| 8 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 9 |
| 9 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| A | 1 | 0 | 1 | 0 | — | — | — | — | — |
| B | 1 | 0 | 1 | 1 | — | — | — | — | — |
| C | 1 | 1 | 0 | 0 | — | — | — | — | — |
| D | 1 | 1 | 0 | 1 | — | — | — | — | — |
| E | 1 | 1 | 1 | 0 | — | — | — | — | — |
| F | 1 | 1 | 1 | 1 | — | — | — | — | — |

因為現在的狀態不會有這種值，
所以下一個狀態就寫成 don't care

- $Q_3' \sim Q_0'$ 邏輯式

$$Q_3' = Q_2 \cdot Q_1 \cdot Q_0 + Q_3 \cdot \overline{Q}_0$$

$$Q_2' = Q_2 \cdot \overline{Q}_1 + Q_2 \cdot \overline{Q}_0 + \overline{Q}_2 \cdot Q_1 \cdot Q_0$$

$$Q_1' = Q_1 \cdot \overline{Q}_0 + \overline{Q}_3 \cdot \overline{Q}_1 \cdot Q_0$$

$$Q_0' = \overline{Q}_0$$

- D型正反器的右邊表示輸入，代入上式即可求得電路圖

8.1.4 用 JK 正反器來設計 mode-10

上數計數器

- $Q_3' \sim Q_0'$ 的邏輯式

$$Q_3' = Q_2 \cdot Q_1 \cdot Q_0 + Q_3 \cdot \overline{Q}_0$$

$$Q_1' = Q_1 \cdot \overline{Q}_0 + \overline{Q}_3 \cdot \overline{Q}_1 \cdot Q_0$$

$$Q_2' = Q_2 \cdot \overline{Q}_1 + Q_2 \cdot \overline{Q}_0 + \overline{Q}_2 \cdot Q_1 \cdot Q_0$$

$$Q_0' = \overline{Q}_0$$

- 將 J,K 代入邏輯式

□ $J_0 = 1$ 或者 Q_0

□ $K_0 = 1$ 或者 Q_0

□ $J_1 = Q_3 \cdot Q_0$

□ $K_1 = Q_0$

□ $J_2 = Q_1 \cdot Q_0$

□ $K_2 = Q_1 \cdot Q_0$

□ $J_3 = Q_2 \cdot Q_1 \cdot Q_0$

□ $K_3 = Q_0$

- 同步電路的設計從特性表之後都是機械化作業。
因此「真正的設計在於特性表的作成」

8.2 序向電路的重置

- 為了要設定序向電路剛開啟電源的初始狀態，所以會用重置信號來把正反器重置為 '0'。因為這種目的而加進電路的重置稱為啟動重置 (power on reset) 或者系統重置 (system reset)，動作分為非同步重置與同步重置兩種。

8.2.1 非同步重置

- 目前的序向電路的重置，都是使用非同步的重置
- 只要重置為 ON 就會瞬間把正反器給重置

8.2.2 同步重置

- 重置的時間點是「時脈正緣」且重置信號為 ON

D 型正反器的同步重置

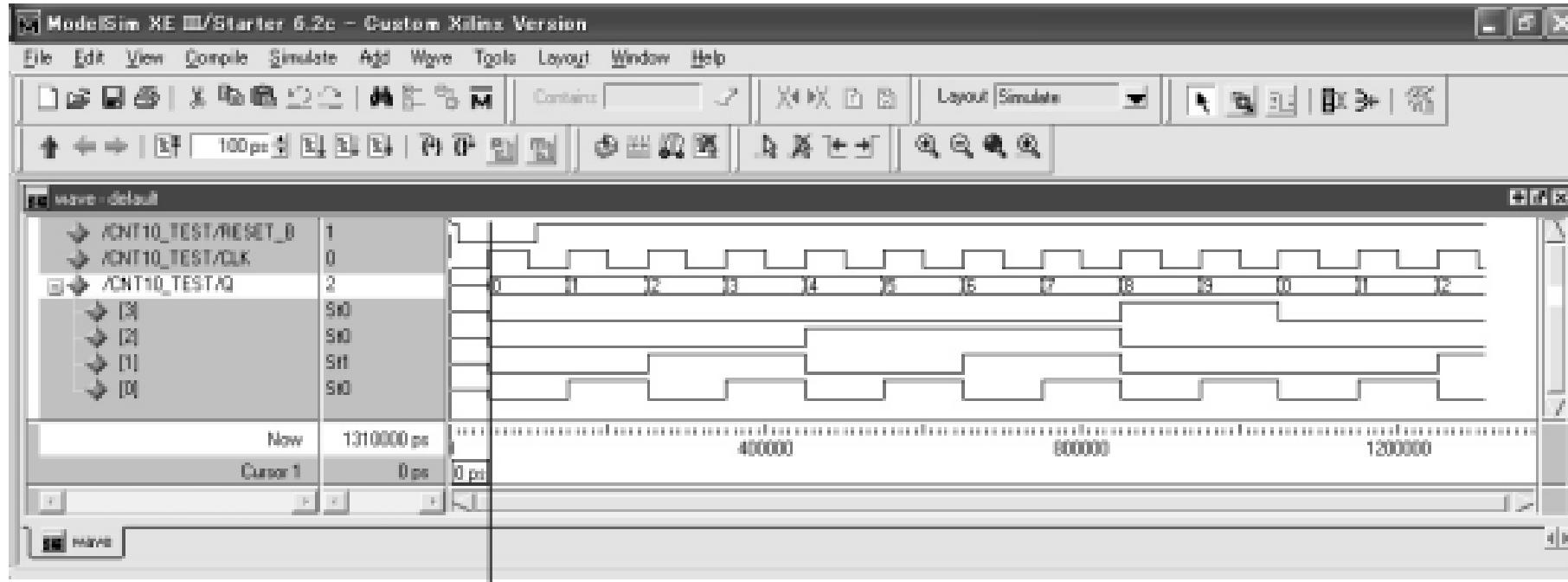
- D 型正反器想讓接下來的狀態為 '0', 現在輸入 D 就要為 '0'
- 在各個正反器的 D 輸入跟「重置」作 AND 運算

JK 正反器的同步重置

- 若要 JK 正反器的下一個狀態為 '0', 要讓 $J='0'$, $K='1'$,
- 各正反器的 J 跟「重置」作 AND, 'K' 則是跟「重置」作 OR

同步重置的 Verilog HDL 描述

- 這個電路的所有動作都是當時脈是正緣的時候才執行，要特別留意 always 裡面「posedge CLK」的寫法



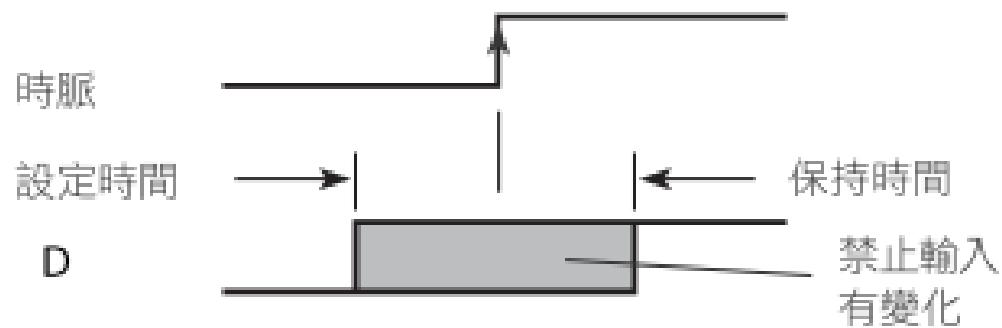
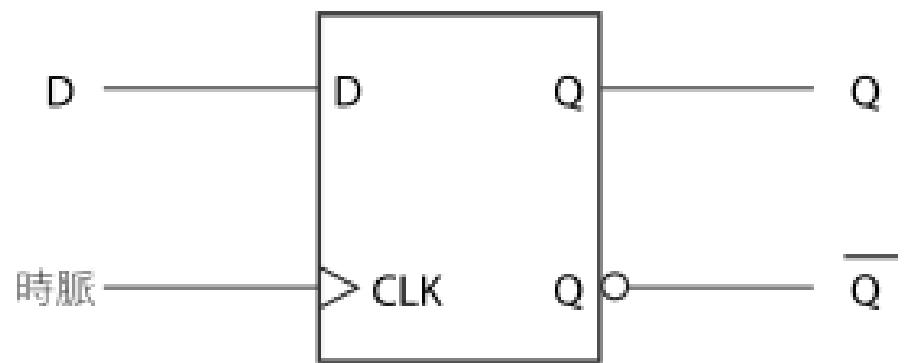
CLK 正緣的時候如果 $\text{RESET_B} = '0'$ 的話執行重置動作

▲ 圖 8.9 附有同步重置的同步 mode-10 上數計數器的模擬結果

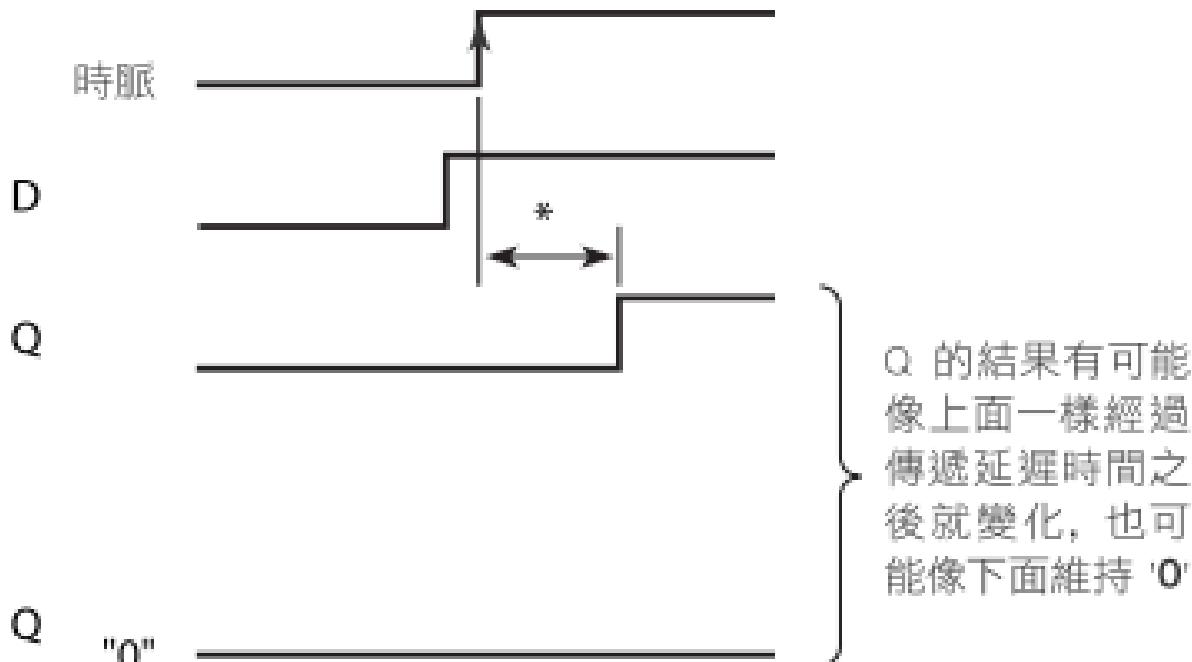
8.3 最大時脈週期

8.3.1 保持時間與設定時間

- 對於同步正反器來說，以正緣觸發為例，CLK 正緣之後的某一段時間內，D 或者 J, K 不能有任何變化
- 時脈正緣附近輸入 D 有變化，時脈正緣之後就不能保證的 Q 與 \bar{Q} 的值為何
- 這個在時脈正緣前不能有變化的時間稱為設定，後面的時間稱為保持時間



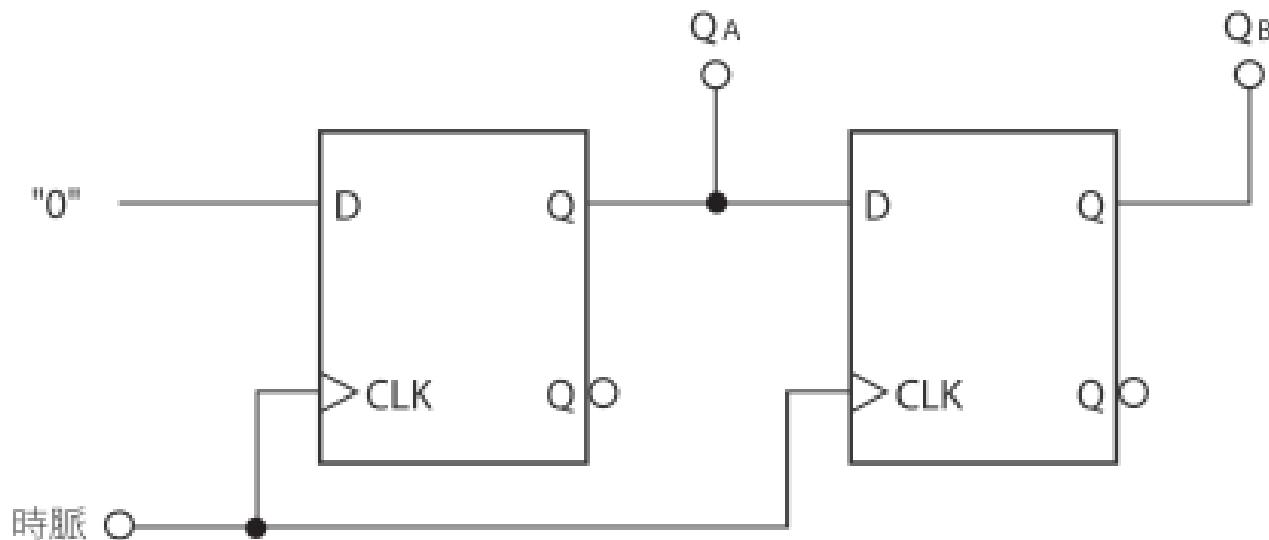
譬如輸入 D 在設定時間中有如下變化的話, Q 的值就不一定了



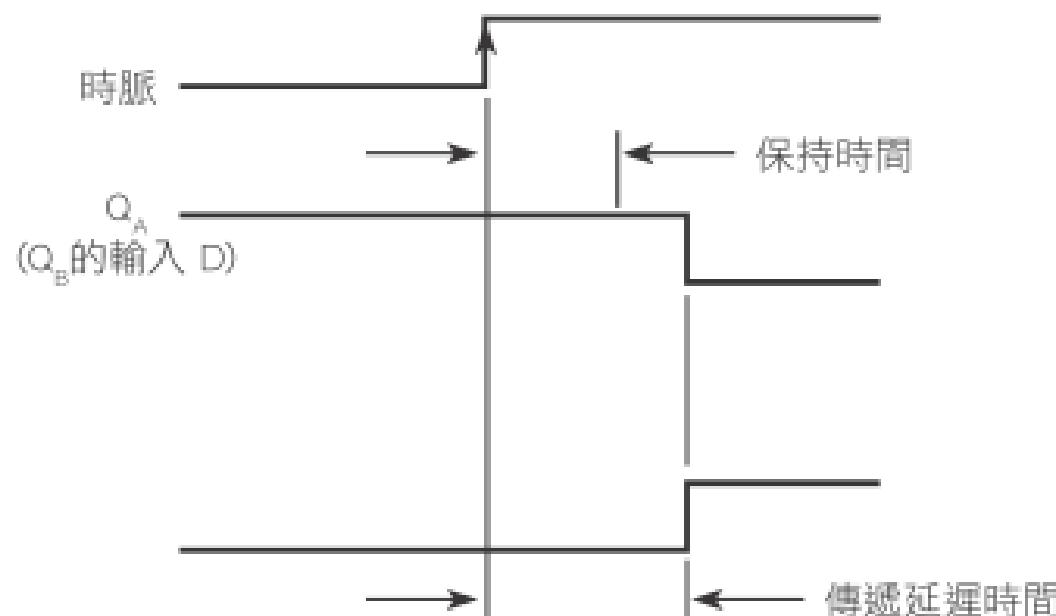
▲ 圖 8.10 設定時間與保持時間

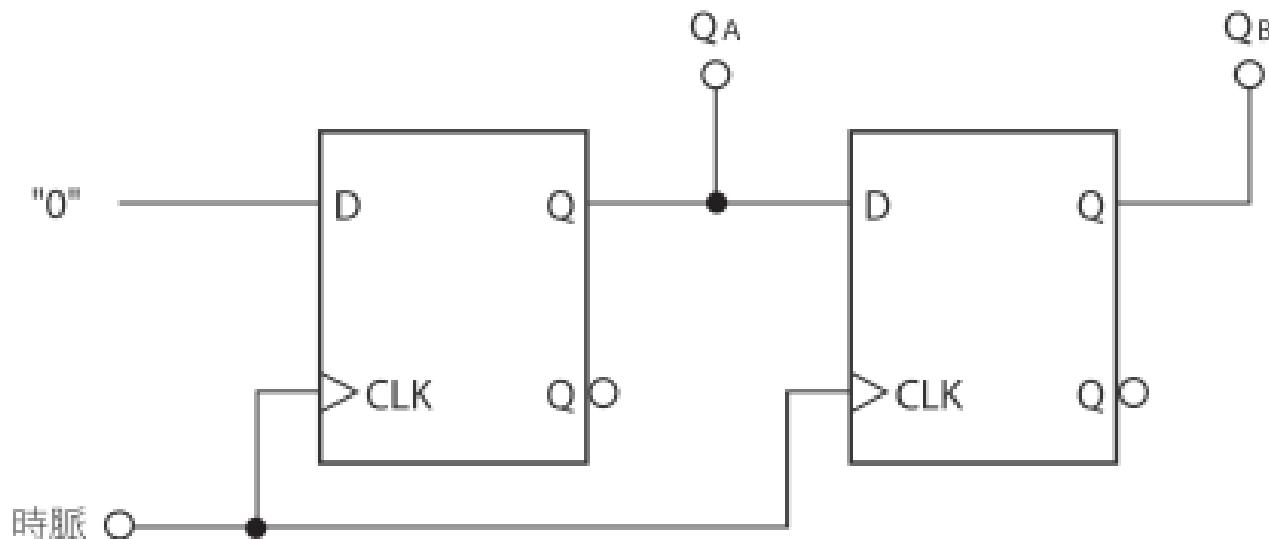
傳遞延遲時間與保持時間的關係

- 傳遞延遲時間 > 保持時間 否則無法正常動作
- 傳遞延遲時間 > 保持時間
 - 這時候是在時脈正緣之後, Q_B 的輸入 D 在保持時間中保持 '0' 不變, 所以 Q_B 可以輸出正確的值 '1'
- 傳遞延遲時間 < 保持時間
 - 因為輸入 D 在保持時間內有變化, 所以 Q_B 的值變成不確定了

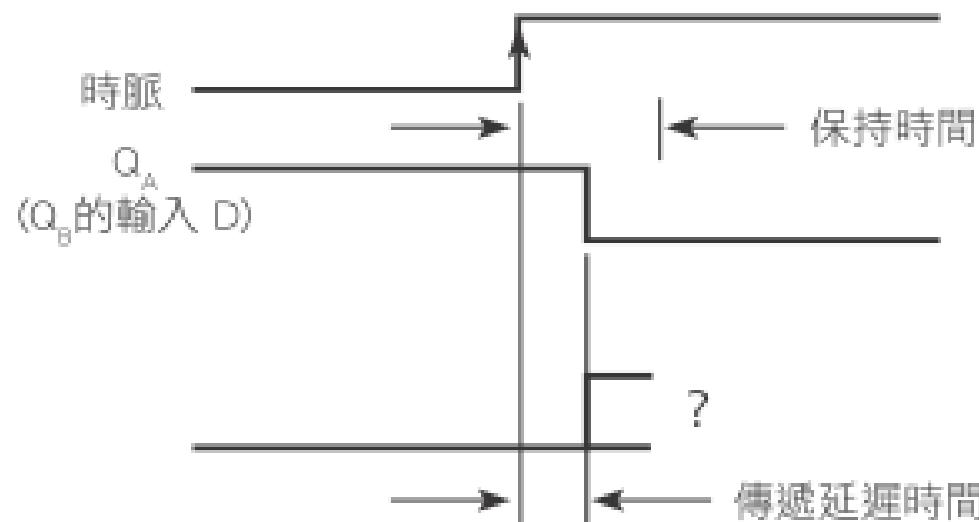


(1) 傳遞延遲時間 > 保持時間的話



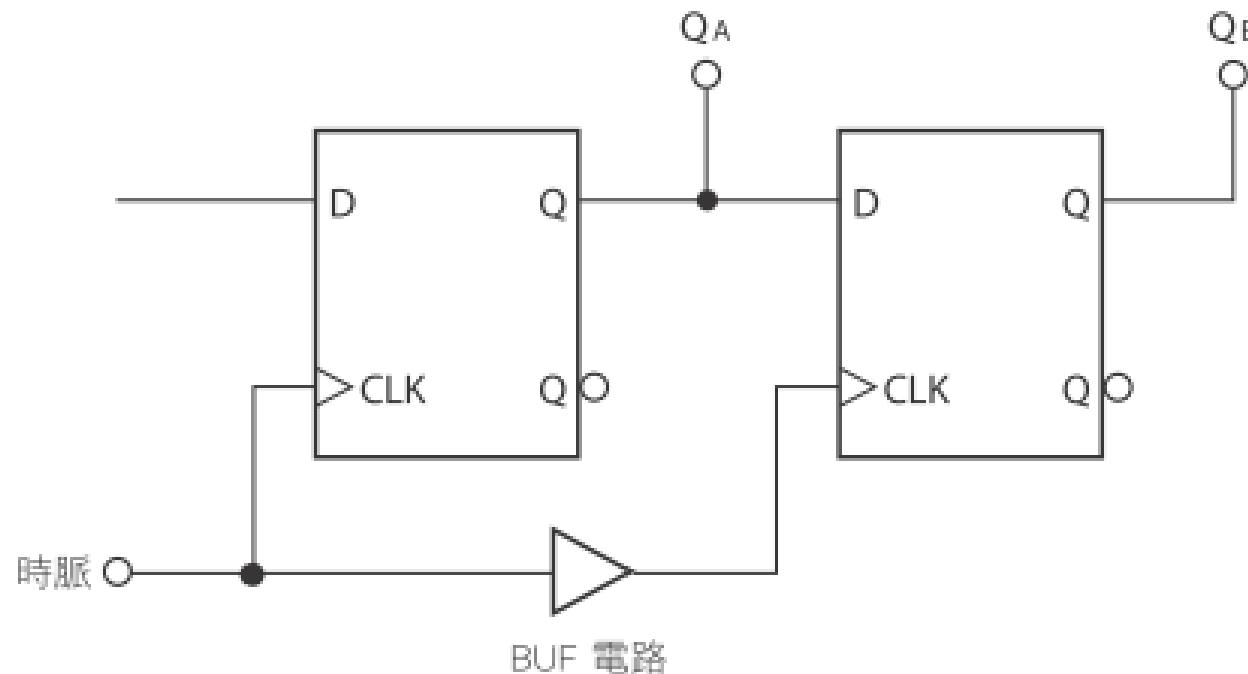


(2) 傳遞延遲時間 < 保持時間的話



保持時間中輸入 D 有變化,
所以 Q_B 的值變成不確定了

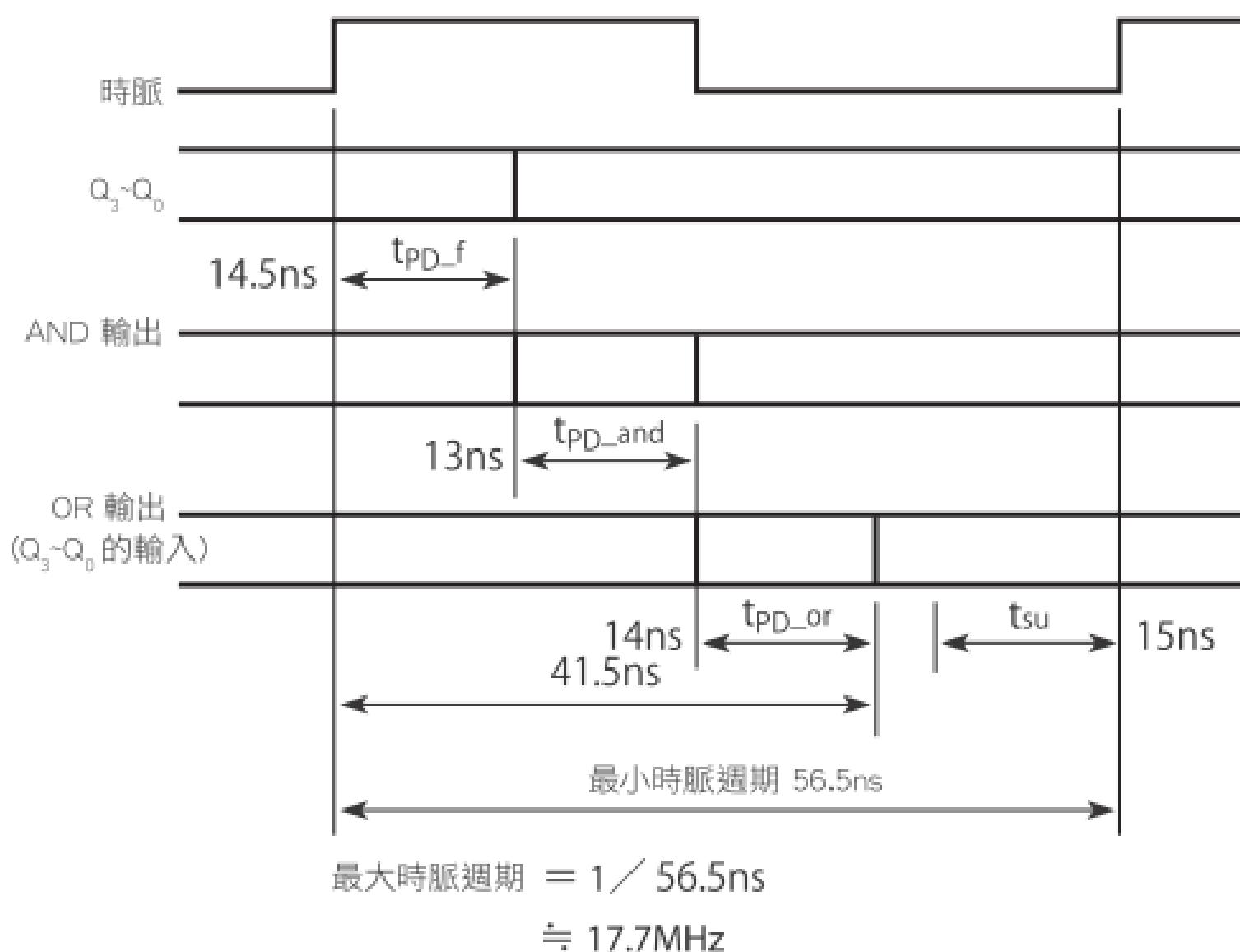
- 數位 IC 都會滿足傳遞延遲時間 > 保持時間
- 不過若像下圖的接法，因為還要加上 Q_B 的 CLK 前面的 BUF 電路的傳遞延遲時間，有可能會造成保持時間中輸入有變化的誤動作。



8.3.2 最大時脈週期

- 最小時脈週期
 - = 正反器的傳遞延遲時間(t_{PD_f})
 - + 正反器之間邏輯閘電路的傳遞延遲時間(t_{PD_g})
 - + 正反器的設定時間(t_{su})
- 最大時脈週期 = $\frac{1}{\text{最小時脈週期}}$

- 舉例：
 - 正反器的傳遞延遲時間：14.5[ns]
 - 正反器之間邏輯閘電路的傳遞延遲
 - AND 電路： 13[ns]
 - OR 電路： 14[ns]
 - 正反器的設定時間： 15[ns]



▲ 圖 8.13 時脈正緣時正反器輸入變化到達輸入 D 的所需時間