

數位系統設計

當一個大型的數位電路使用IC方式來實現時，一般會採用所謂的半客戶式(semi-custom)或稱標準單元式(cell-based)的設計方式來進行。設計時，通常會將系統分為兩個部分：控制單元(control unit)與資料運算單元(datapath)，如圖6.35所示。控制單元產生控制訊號(control signals)用以控制資料運算單元；而資料運算單元產生的狀態訊號(status signals)則會送至控制單元當成部分的輸入訊號。

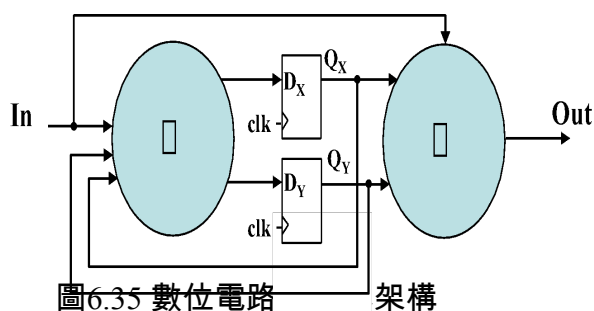


圖6.35 數位電路架構

控制單元與資料運算單元的詳細內部電路，可使用圖6.36來說明。控制單元主要由：輸出邏輯(output logic)、次狀態邏輯(next state logic)與目前狀態記憶暫存器(current state registers)，三者組合而成。其中的目前狀態暫存器是記憶單元(由正反器/暫存器組成)，用以紀錄目前的電路狀態，需使用時脈訊號clock來做同步驅動；輸出邏輯模組是組合電路，根據目前的電路狀態與輸入訊號，產生所需的控制訊號來控制datapath；次狀態邏輯也是組合電路，根據目前的電路狀態與輸入訊號，來產生下一時刻的電路狀態。

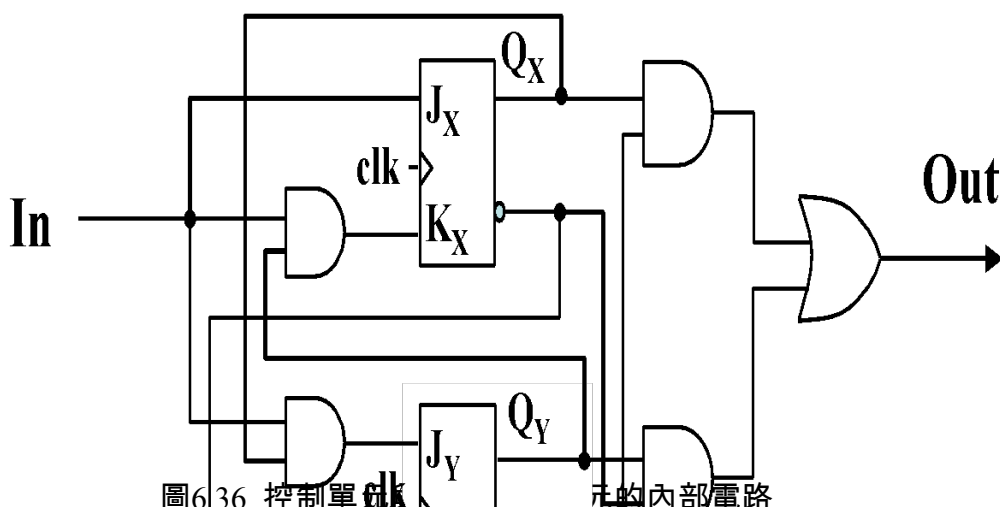


圖6.36 控制單元的內部電路

資料運算單元(datapath)通常包含加減法器(adder)、乘法器(multiplier)、除法器(divider)、比較器(comparator)、移位器(shift register)、邏輯運算元(arithmetic logic unit, 簡寫為ALU)、加減法器、乘法器、比較器等。如圖6.36所示, datapath主要執行設計者所規劃電路的運算行為。資料運算單元根據控制單元傳來的控制訊號, 依序進行(一個狀態接著另一個狀態)所規劃好的運算動作, 譬如: 根據傳來的控制訊號, 進行乘法或除法運算、從記憶體或暫存器讀出一個值、選擇兩個輸入中的一個當成輸出等等。

控制單元的設計也稱之為有限狀態機(finite state machine, 簡寫為FSM)的設計。如圖6.36所示, 控制單元分成三個模組:

- (a) 次狀態邏輯：組合電路，根據現在輸入(control inputs)與目前狀態(current state)，產生下一個電路狀態(next state)。

- (b) 狀態記憶暫存器：記憶單元，儲存目前的電路狀態(由正反器/暫存器組成，需使用脈波clock來同步)。
- (c) 輸出邏輯：組合電路，根據現在輸入與目前狀態，產生輸出控制訊號來控制datapath單元進行所需運算。

因為FSM的設計，是數位系統中極重要的一部分，所以接下來我們將針對它做更詳細的介紹。

有限狀態機電路設計

前一小節已經介紹有限狀態機的分析流程，本小節將進入最重要的部分 - 介紹設計一個有限狀態機FSM的詳細流程。如前所述，一個FSM主要由：輸出邏輯模組、次狀態邏輯模組與目前狀態記憶暫存器，三者組合而成。FSM的設計主要就是：先決定所需要的目前狀態記憶暫存器，然後再根據電路的狀態表與控制輸出，設計得出輸出邏輯模組與次狀態邏輯模組這兩個組合電路模組的內部邏輯閘接線圖。

設計一個有限狀態機電路的詳細流程，主要包含七個步驟：

1. 電路規格定義(specification definition)
2. 狀態圖的建構(state diagram construction)與狀態指派(state assignment)
3. 正反器的決定(flip-flop determination)
4. 狀態表的建構(state table construction)
5. 畫出卡諾圖並化簡
6. 分別導出次狀態邏輯模組與輸出邏輯模組之最簡輸出布林表示式
7. 畫出其邏輯電路圖

接下來我們將使用一個例子來詳細介紹每一步驟。

步驟一：電路規格定義

要設計一個電路，首先需要有電路功能規格的詳細定義與說明。這個功能規格可能是負責電路設計的公司與其客戶討論得出的，或者是公司針對未來市場需求而規劃定義得出的。當然電路規格可能需要一陣子的來回修正才能最後定案，但是總而言之，電路規格一定要定義的清楚且完整。

一個簡略的電路規格描述如下：

設計一個紅、綠、黃燈控制的電路，紅燈亮40秒後，換綠燈亮；綠燈亮30秒後，換黃燈亮；黃燈亮5秒後，換紅燈亮。電路會持續循環不停止。

步驟二：狀態圖建構與狀態指派

根據電路的規格需求，規劃電路所需要的狀態個數並建構出其狀態圖，圖6.46是一個可能的電路狀態圖，它描述四個狀態與輸入輸出之間的變化關係。

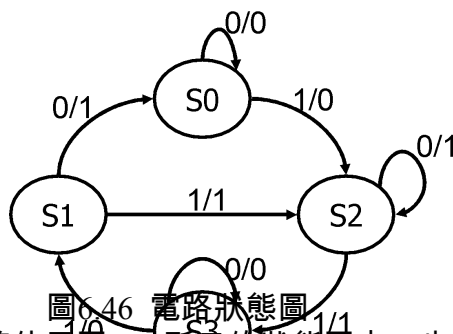


圖6.46 電路狀態圖

為了方便介紹，我們將使用圖6.46所示的狀態圖來一步一步執行剩下來的步驟，以設計出所需的電路。因為電路有四個狀態，我們首先需要指派每一個狀態一個二進制編碼，如下所示：

狀態	編碼一	編碼二	編碼三
S0	0 0	1 1	0 0
S1	0 1	1 0	0 1
S2	1 0	0 1	1 1
S3	1 1	0 0	1 0

上表提供了三種編碼表示，編碼一是最常見的二進制編碼方式，從小到大，以00、01、10、11順序來編碼。編碼二是從大到小，以11、10、01、00順序來編碼。編碼三則是使用格雷碼來編碼，以00、01、11、10順序來編碼。哪一種編碼方式較好？優缺點為何？本書不針對此做討論也不特別建議，有興趣的讀者可參考相關書籍。我們先假設使用第一種方式來編碼。

步驟三：決定正反器

步驟二決定出四個狀態與其編碼方式之後，接下來我們要決定使用何種正反器來儲存資料？有D型、JK型與T型三種，設計者可依需要選擇其中的一種來實現。若使用D型正反器來當作狀態暫存單元，則我們擬建構的FSM概念示意圖如圖6.47所示，其中輸入與輸出變數分別標示為In與Out，要區別四個狀態則需要兩個D型正反器，分別標示為X與Y。

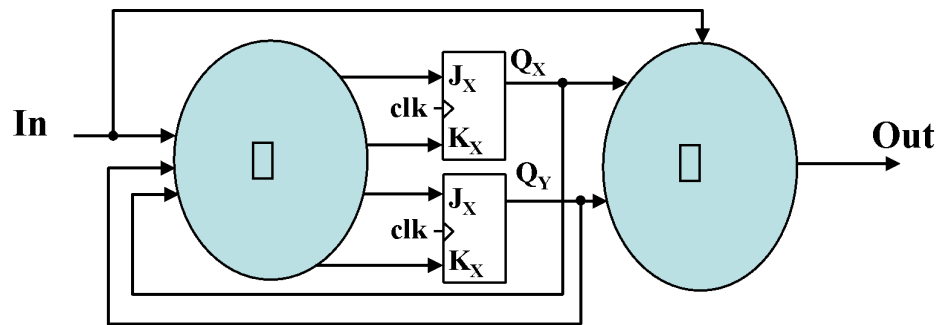


圖6.47 以D型正反器為基礎的FSM概念圖

接下來的步驟就是根據電路的需求，分別設計出左邊那個次狀態邏輯組合模組所需的詳細電路圖與右邊那個輸出邏輯組合模組所需的詳細電路圖。

使用D型正反器來當作狀態暫存單元

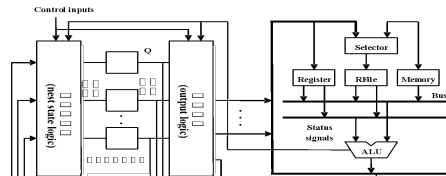
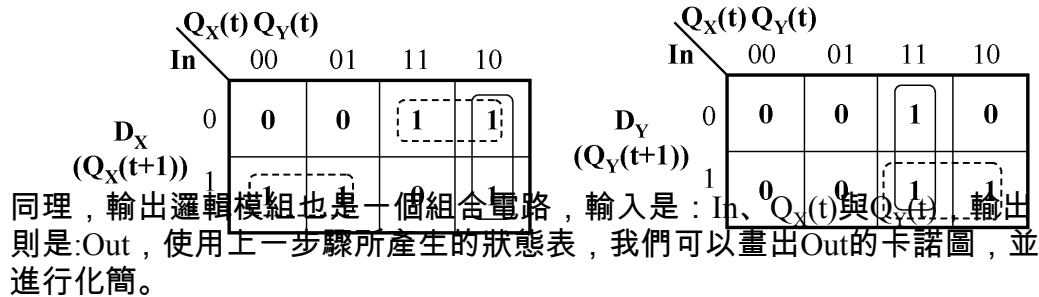
步驟四：狀態表建構

根據狀態圖與D型正反器的特性，我們可建構如下的狀態表

輸入	輸出				
外部輸入	目前狀態	下一個狀態	外部輸出		
In	$Q_x(t)$	$Q_y(t)$	$Q_x(t+1)$	$Q_y(t+1)$	Out
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	1	0	1
0	1	1	1	1	0
1	0	0	1	0	0
1	0	1	1	0	1
1	1	0	1	1	1
1	1	1	0	1	0

步驟五：畫出卡諾圖並化簡

次狀態邏輯模組是一個組合電路，輸入變數是：In、 $Q_X(t)$ 與 $Q_Y(t)$ ，輸出變數則是： D_X 與 D_Y （即是 $Q_X(t+1)$ 與 $Q_Y(t+1)$ ）。針對每一個輸出變數我們需要建構一個卡諾圖，使用上一步驟所產生的狀態表，我們可以分別畫出 D_X 與 D_Y 的卡諾圖，並進行化簡。



步驟六：分別導出次狀態邏輯模組與輸出邏輯模組之最簡輸出布林表示式

卡諾圖化簡後，次狀態邏輯模組所需的輸出布林表示式可寫成：

$$D_X = \text{In}Q_X'(t) + \text{In}'Q_X(t) + Q_X(t)Q_Y'(t)$$

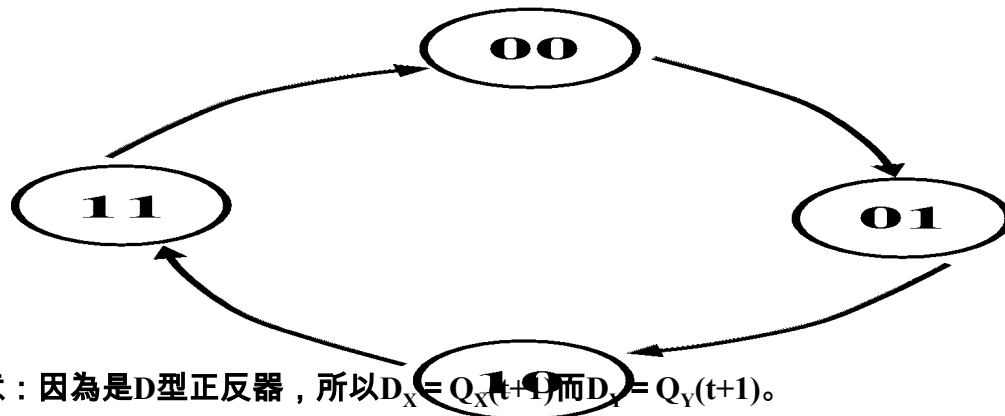
$$D_Y = Q_X(t)Q_Y(t) + \text{In}Q_X(t)$$

輸出邏輯模組所需的輸出布林表示式可寫成：

$$\text{Out} = Q_X(t)Q_Y'(t) + Q_X'(t)Q_Y(t)$$

步驟七：畫出其邏輯電路圖

使用步驟六所推導出的輸出布林表示式並結合圖6.47，可以建構出次狀態邏輯模組與輸出邏輯模組兩者的詳細邏輯閘接線圖。



使用JK型正反器來當作狀態暫存單元

針對圖6.46的狀態圖，若使用JK型正反器來當作狀態暫存單元，則我們擬建構設計的FSM概念示意圖如圖6.48所示，其中輸入與輸出變數分別標示為In與Out，要區別四個狀態則需要兩個JK型正反器，分別標示為X與Y。

圖6.48 以JK型正反器為基礎的FSM概念圖

接下來就是根據電路的需求重做步驟四到七，分別設計出左邊那個次狀態邏輯組合模組所需的詳細電路圖與右邊那個輸出邏輯組合模組所需的詳細電路圖。

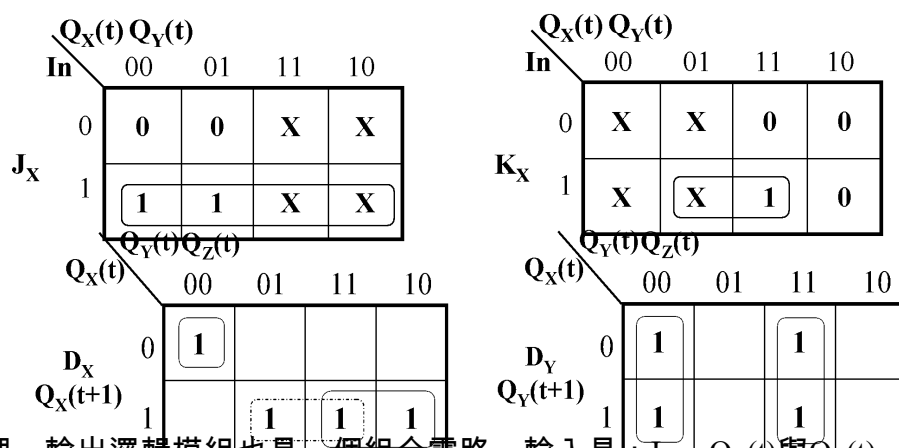
步驟四：狀態表建構

根據狀態圖與JK型正反器的特性，我們可建構如下的狀態表

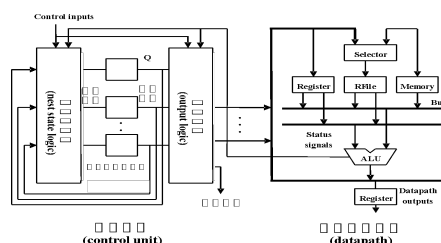
輸入	輸出								
外部輸入	目前狀態	下一個狀態	X的設定	Y的設定	外部輸出				
In	$Q_X(t)$	$Q_Y(t)$	$Q_X(t+1)$	$Q_Y(t+1)$	J_X	K_X	J_Y	K_Y	Out
0	0	0	0	0	0	X	0	X	0
0	0	1	0	0	0	X	X	1	1
0	1	0	1	0	X	0	0	X	1
0	1	1	1	1	X	0	X	0	0
1	0	0	1	0	1	X	0	X	0
1	0	1	1	0	1	X	X	1	1
1	1	0	1	1	X	0	1	X	1
1	1	1	0	1	X	1	X	0	0

步驟五：畫出卡諾圖並化簡

次狀態邏輯模組是一個組合電路，輸入變數是： In 、 $Q_X(t)$ 與 $Q_Y(t)$ ，輸出變數則是： J_X 、 K_X 、 J_Y 、 K_Y 。針對每一個輸出變數我們需要建構一個卡諾圖，使用上一步驟所產生的狀態表，我們可以分別畫出 J_X 、 K_X 、 J_Y 、 K_Y 的卡諾圖，並進行化簡。



同理，輸出邏輯模組也是一個組合電路，輸入是： In 、 $Q_X(t)$ 與 $Q_Y(t)$ ，輸出則是： Out ，使用上一步驟所產生的狀態表，我們可以畫出 Out 的卡諾圖，並進行化簡。



步驟六：分別導出次狀態邏輯模組與輸出邏輯模組之最簡輸出布林表示式

卡諾圖化簡後，次狀態邏輯模組所需的輸出布林表示式可寫成：

$$J_X = \text{In} \quad K_X = \text{In}Q_Y(t)$$

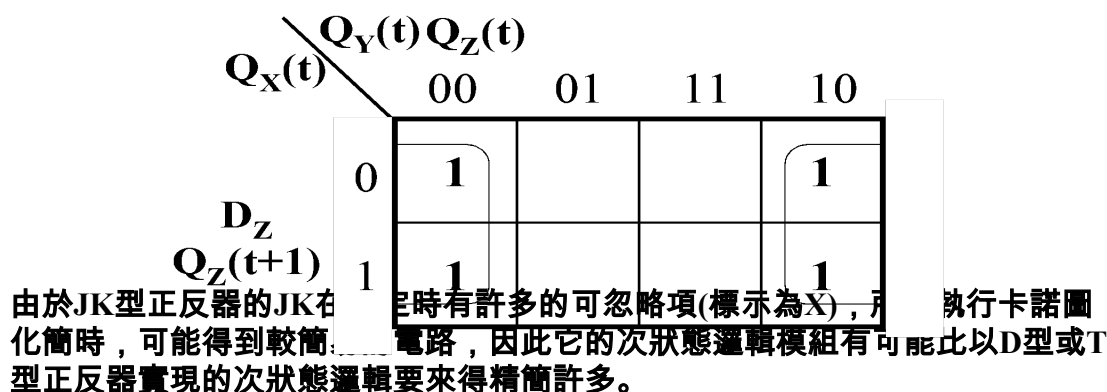
$$J_Y = \text{In}Q_X(t) \quad K_Y = Q_X'(t)$$

輸出邏輯模組所需的輸出布林表示式可寫成：

$$\text{Out} = Q_X(t)Q_Y'(t) + Q_X'(t)Q_Y(t)$$

步驟七：畫出其邏輯電路圖

使用步驟六所推導出的輸出布林表示式並結合圖6.48，可以建構出次狀態邏輯模組與輸出邏輯模組兩者的詳細邏輯閘接線圖。



同步計數器(synchronous counter)

同步計數器中的所有正反器都使用同一個時鐘脈衝來觸發，所以其輸出狀態會在同一時刻一起改變，在控制或應用上都比非同步計數器來的簡單。同步計數器的應用範圍相當廣泛，本小節將詳細介紹各種不同的同步計數器，包含：上數計數器(countup counter)、下數(countdown counter)計數器與任意數值計數器等。

要設計各式各樣不同需求的同步計數器，最簡單且直接的方式就是 - 應用第六章所介紹的同步有限狀態機設計步驟，按部就班進行，就能設計出不同的「同步計數器」。為什麼呢？在6.5.2節曾提及，一個同步的有限狀態機主要由：輸出邏輯模組、次狀態邏輯模組與目前狀態記憶暫存器，三者組合而成。FSM的設計主要就是：先決定所需要的目前狀態記憶暫存器，然後再根據電路的狀態表與控制輸出，設計得出「輸出邏輯模組」與「次狀態邏輯模組」這兩個組合電路模組的內部邏輯閘接線圖。

而事實上，一個同步計數器應該算是一個精簡過的有限狀態機，它包含：目前狀態暫存器與次狀態邏輯模組而已，不需要具備輸出邏輯模組。為了方便說明，我們修改圖6.47成下圖8.5。在這個圖中，我們拿掉輸出邏輯模組與輸入值In。如果我們能適當地設計次狀態邏輯模組，讓目前狀態暫存器(圖中兩個D型正反器)，能在每次時脈正邊緣到達時，依序輸出00、01、10、11、00、01、10、11...週而復始、持續循環，這就是一個標準的兩位元同步上數計數器。整個電路使用同一個時脈clk觸發，每次邊緣觸發會引起一個狀態的改變，從00、01、10到11然後再回到00，重新開始開始上數。同理，如果需要四位元計數器，

則使用四個正反器搭配合適的「次狀態邏輯模組」即可完成。另外，和設計一般的有限狀態機相同，目前狀態暫存器可使用D型、JK型或是T型正反器來實現。

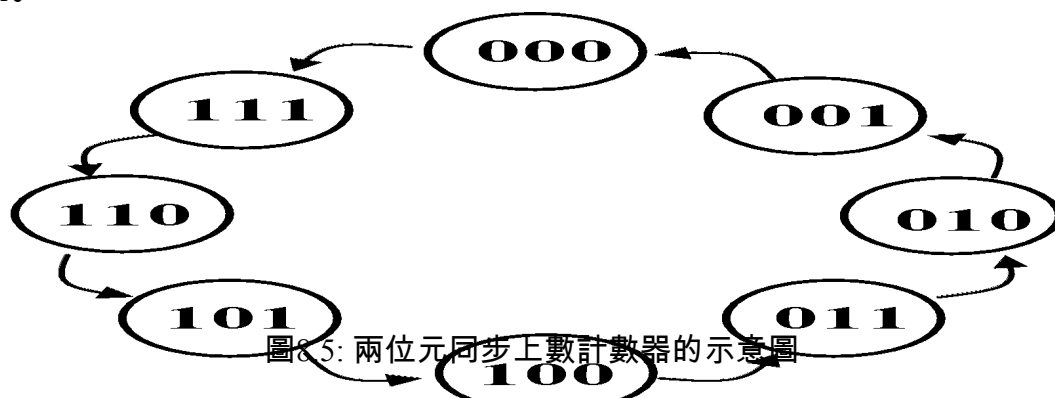


圖8-5: 兩位元同步上數計數器的示意圖

以JK型正反器建構的兩位元同步上數計數器

要設計一個用JK型正反器建構的兩位元同步上數計數器，其步驟如下。首先，我們畫出所需的狀態圖與狀態表，如下所示，其中假設使用兩個正緣觸發JK型正反器，分別為X與Y（X代表高位元）：

輸入	輸出						
目前狀態	下一個狀態	X的設定	Y的設定				
$Q_X(t)$	$Q_Y(t)$	$Q_X(t+1)$	$Q_Y(t+1)$	J_X	K_X	J_Y	K_Y
0	0	0	1	0	X	1	X
0	1	1	0	1	X	X	1
1	0	1	1	X	0	1	X
1	1	0	0	X	1	X	1

每個正反器JK的設定值，可由JK型正反器的激勵表來得出。很明顯地，此電路無外部輸入值。我們只關心，每次時脈邊緣到達時，電路能否順利進入所規劃的下一狀態，所以電路沒有輸入值也不需要輸出邏輯模組。因為是使用JK型正反器，所以需要針對兩個正反器的JK設定值，畫出所需的卡諾圖，並進行化簡。

化簡後，可得到次狀態模組的輸出布林表示式為：

$$J_X = Q_Y(t) \quad K_X = Q_Y(t)$$

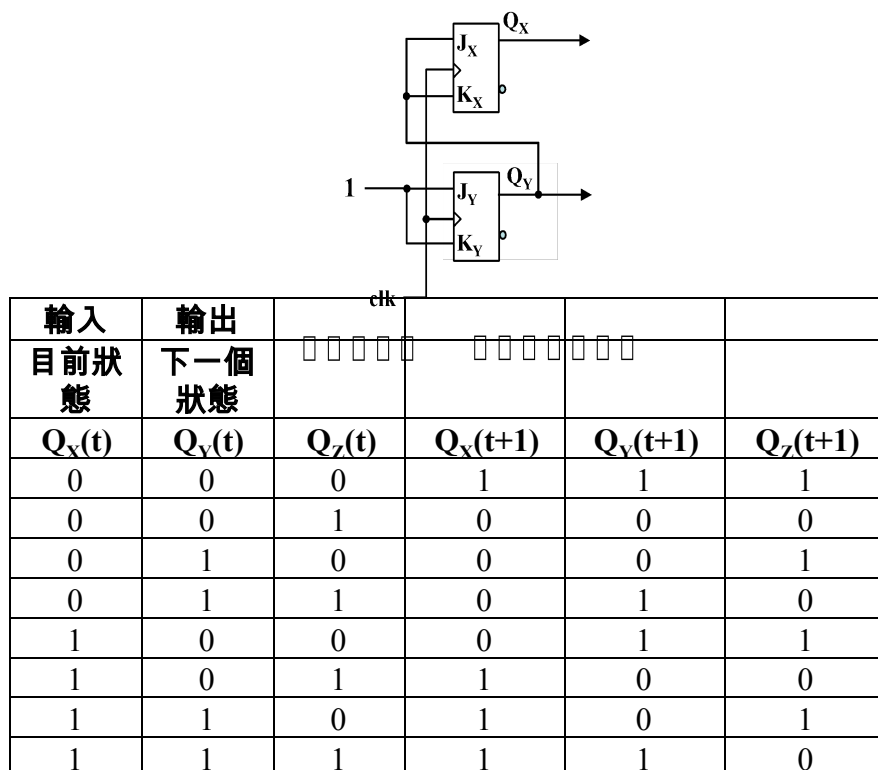
$$J_Y = 1 \quad K_Y = 1$$

故其電路圖如下所示：

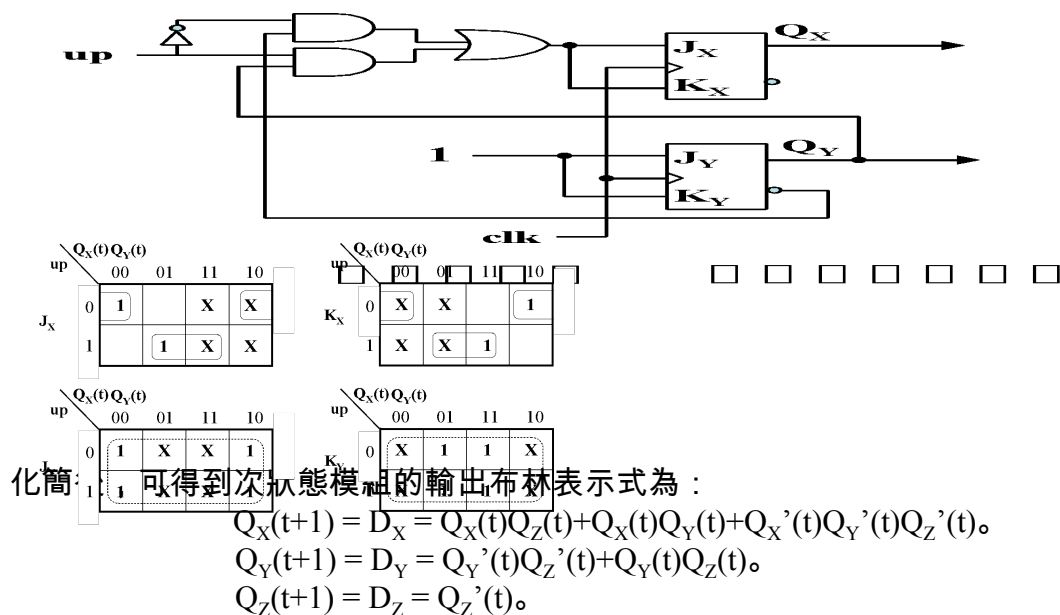
和6.5.2節的實現方式一樣，我們也可以嘗試使用D型或T型正反器來設計實現此電路。

以D型正反器建構的三位元同步下數計數器

要設計一個用D型正反器建構的三位元同步下數計數器(111□110□101□100□011□010□001□000□111□110...持續循環下去)，其步驟如下。首先，我們畫出所需的狀態圖與狀態表，如下圖，其中假設三個D型正反器分別為X、Y與Z（X代表最高位元）：



因為是使用D型正反器，所以可根據每個狀態的輸出狀態值，畫出所需的卡諾圖，並進行化簡。



故其電路圖如下所示：

當然，這個電路也可以用JK型或T型正反器來實現。

以JK型正反器建構的兩位元上下數計數器

要設計一個用JK型正反器建構的兩位元上下數計數器，其步驟如下。假設有一個外部輸入訊號線up，當up=0時，電路會下數(11□10□01□00□11 □10...持續循環下去)，當up=1時，電路會上數(00□01□10□11□00□01...持續循環下去)。首先，我們畫出所需要的狀態圖與狀態表，如下圖，其中假設兩個JK型正反器為X與Y (X代表高位元)，因為是使用JK型正反器，所以需要針對兩個正反器的JK設定值，畫出所需的卡諾圖，並進行化簡。

輸入	輸出							
外部輸入	目前狀態	下一個狀態	X的設定	Y的設定				
up	$Q_X(t)$	$Q_Y(t)$	$Q_X(t+1)$	$Q_Y(t+1)$	J_X	K_X	J_Y	K_Y
0	0	0	1	1	1	X	1	X
0	0	1	0	0	0	X	X	1
0	1	0	0	1	X	1	1	X

0	1	1	1	0	X	0	X	1
1	0	0	0	1	0	X	1	X
1	0	1	1	0	1	X	X	1
1	1	0	1	1	X	0	1	X
1	1	1	0	0	X	1	X	1

化簡後，可得到次狀態模組的輸出布林表示式為：

$$J_X = up'Q_Y'(t) + upQ_Y(t) \quad K_X = up'Q_Y'(t) + upQ_Y(t)$$

$$J_Y = 1 \quad K_Y = 1$$

故其電路圖如下所示：

當然，這個電路也可以用D型或T型正反器來實現。

以D型正反器建構的特定數值同步計數器

假設有一個特定數值同步計數器，其計數順序如下： 000□010□100□101□110□111□000□010□100...持續循環下去。若要使用D型正反器來建構此電路，其步驟如下：首先，我們畫出所需的狀態圖與狀態表，如下圖，其中假設三個D型正反器為X、Y與Z(X為最高位元)。

輸入	輸出				
目前狀態	下一個狀態				
$Q_X(t)$	$Q_Y(t)$	$Q_Z(t)$	$Q_X(t+1)$	$Q_Y(t+1)$	$Q_Z(t+1)$
0	0	0	0	1	0
0	0	1	X	X	X
0	1	0	1	0	0
0	1	1	X	X	X
1	0	0	1	0	1
1	0	1	1	1	0
1	1	0	1	1	1
1	1	1	0	0	0

狀態001與011不應該發生，所以其下一狀態為可忽略項X，以利卡諾圖化簡。
因為是使用D型正反器，所以可根據每個狀態的輸出狀態值，畫出所需的卡諾圖，並進行化簡。

化簡後，可得到次狀態模組的輸出布林表示式為：

$$Q_X(t+1) = D_X = Q_X(t)Q_Y'(t) + Q_Y(t)Q_Z'(t)。$$

$$Q_Y(t+1) = D_Y = Q_X'(t)Q_Y'(t) + Q_Y'(t)Q_Z(t) + Q_X(t)Q_Y(t)Q_Z'(t)。$$

$$Q_Z(t+1) = D_Z = Q_X(t)Q_Z'(t)。$$

最後，可依表示式畫出所需的電路圖。