第 3 章

同步電路之設計

3-0 緒　論

在第二章中，我們介紹了一些基本的同步電路，但對於這些電路的基本設計原理則並未提及，本章中，就要詳細介紹同步電路一般化的設計原理及步驟，這些原理及步驟不僅適用於第二章中所提及的各種基本同步電路，亦且適用於一般化之同步序向系統之設計。

3-1 由系統規定至狀態遷移表

在設計一個同步序向系統之前，通常我們必須先知道此序向系統的系統規定（System specification）。而一般所謂的系統規定，均是由該序向系統的輸入輸出間的關係來加以描述的。

對於一個序向系統的輸入輸出關係的描述，則又必須定義其輸入序列（Input sequence）及輸出序列（Output sequence），也就是說給予此系統一連串輸入符號 x_{i_1}，x_{i_2}，x_{i_3}，……，規定出其對應的一連串輸出符號 z_{i_1}，z_{i_2}，z_{i_3}，……。

同步序向系統設計的第一步，便是根據此系統的輸入輸出規定，寫出其狀態表或狀態圖，或進而寫出其狀態遷移表（state transition table），所謂的狀態遷移表，卽是以系統的輸出函數 λ 及次狀態函數 δ 所列出的函數表，其中函數 λ 將 $\{Y;X\}$ 映至 $\{Z\}$，函數 δ 則將 $\{Y;X\}$ 映至 $\{Y\}$。X，Y，Z分別代表輸入變數，狀態變數及輸出變數所成的集合。

玆舉二例以說明此一步驟。

【例 3－1】 欲設計一同步序向系統，規定每當輸入 3K（K＝0，1，2，……）個 1 的時候，輸出爲 1，否則輸出爲 0，試寫出其狀態表。

【解】 由此系統規定，我們可以很自然地定出三個不同的狀態 a，b 以及 c 如下：

a：當系統輸入了 3K 個 1 時，此時的輸出爲 1。

b：當系統輸入了 3K＋1 個 1 時，此時的輸出爲 0。

c：當系統輸入了 3K＋2 個 1 時，此時的輸出爲 0。

根據這三個狀態，我們就可寫出其狀態表如下：

現在狀態	次一狀態		現在輸出
	x＝0	x＝1	
a	a	b	1
b	b	c	0
c	c	a	0

其中輸出一欄僅與現在狀態有關，因此由第一章中所述，可知這是一個摩爾式序向系統。

在上面這個例題中，很顯然地可以看出，這個同步序向系統必須從狀態 a 爲開始，因此我們稱狀態 a 爲起始狀態（Initial state），起始狀態的意義通常是指一個序向系統剛打開電源時所處的狀態，如無特殊規定，通常均假設此起始狀態爲 0，但對某些雙穩態（bipolar）的記憶元件而言，此假設並不正確。

讀者可以驗證如果上例中的序向系統之起始狀態爲 b 的話，則當輸入 2 個，5 個或 8 個…… 1 時，系統的輸出卽爲 1，此與原先的系統規定不符。同理狀態 c 亦然。

【例 3－2】　欲設計一同步序向系統，規定每當輸入序列之長度在三個以上，並且最後三個是 101 時，輸出為 1，否則輸出為 0，試寫出其狀態遷移表。

【解】　為簡便計，我們將忽略輸入長度在三個以下的情況，而將此系統定出下列八種可能的狀態：

a：當系統的輸入序列為 000 時，此時輸出為 0。

b：當系統的輸入序列為 001 時，此時輸出為 0。

c：當系統的輸入序列為 010 時，此時輸出為 0。

d：當系統的輸入序列為 011 時，此時輸出為 0。

e：當系統的輸入序列為 100 時，此時輸出為 0。

f：當系統的輸入序列為 101 時，此時輸出為 1。

g：當系統的輸入序列為 110 時，此時輸出為 0。

h：當系統的輸入序列為 111 時，此時輸出為 0。

根據以上八種狀態，我們就可以將其狀態表寫出如表 3-1。又因為其中有八種狀態，故由第一章中所述可知共須 $\log_2 8 = 3$ 個狀態變數來代表它們，將它們分別定為 $000_2 \cdots\cdots 111_2$ 等八種，就可將表 3-1改變成表 3-2的狀態遷移表。又由此狀態表亦可看出這也是一個摩爾式的序向系統。

圖表 3－2　狀態遷移表

現在狀態	次一狀態 x＝0	次一狀態 x＝1	現在輸出
$y_1\ y_2\ y_3$	$Y_1\ Y_2\ Y_3$	$Y_1\ Y_2\ Y_3$	z
0 0 0	0 0 0	0 0 1	0
0 0 1	0 1 0	0 1 1	0
0 1 0	1 0 0	1 0 1	0
0 1 1	1 1 0	1 1 1	0
1 0 0	0 0 0	0 0 1	0
1 0 1	0 1 0	0 1 1	1
1 1 0	1 0 0	1 0 1	0
1 1 1	1 1 0	1 1 1	0

由以上兩個例子我們可以看出，由系統規定至寫出狀態表的步驟，最重要的就是要根據系統規定之輸入輸出關係而找出其中可能的各種狀態，然後再根據這些狀態去寫出狀態表或狀態遷移表，就容易多了。當然在找出可能的各種狀態時，每個人均可以由不同的觀點而找出不同數目的各種不同狀態，但只要這些狀態足以代表了系統所有可能的情況時，即使狀態數目不同，也可以經由3-3節中所述的等效狀態的方法加以化簡，最後仍然可以得到相同的結果，我們將在該節中再予介紹。

3-2 無起始狀態之序向系統

在上節的兩個例子中，例3-1必須以a為起始狀態，而例3-2也必須以a或e為起始狀態，否則均會產生與系統規定不符的輸出結果，因此這種系統是屬於有起始狀態的序向系統。

另有某些序向系統，其系統規定之輸入輸出關係對起始時並無限制，且亦無週期性可言，以致於此類的序向系統沒有明顯的起始狀態，因此在設計此序向系統時可以忽略此限制。例如前章中所介紹的移位暫存器即屬於此類。

【例3－3】 試寫出一個三位元移位暫存器的狀態遷移表。

圖表3－3 例3－3之狀態遷移表

現在狀態	次一狀態		次一輸出	
	x＝0	x＝1	x＝0	x＝1
$y_1\ y_2\ y_3$	$Y_1\ Y_2\ Y_3$	$Y_1\ Y_2\ Y_3$	$z_1\ z_2\ z_3$	$z_1\ z_2\ z_3$
0 0 0	0 0 0	0 0 1	0 0 0	0 0 1
0 0 1	0 1 0	0 1 1	0 1 0	0 1 1
0 1 0	1 0 0	1 0 1	1 0 0	1 0 1
0 1 1	1 1 0	1 1 1	1 1 0	1 1 1
1 0 0	0 0 0	0 0 1	0 0 0	0 0 1
1 0 1	0 1 0	0 1 1	0 1 0	0 1 1
1 1 0	1 0 0	1 0 1	1 0 0	1 0 1
1 1 1	1 1 0	1 1 1	1 1 0	1 1 1

【解】 三位元的移位暫存器有三個記憶元件，故有 $2^3 = 8$ 個狀態，其狀態遷移表如表 3-3所示。其輸出也有三個，恰巧即爲其三個狀態變數。由表中我們可看出這是一個米黎式的序向系統，當然我們也可以很容易地將它改寫爲摩爾式的序向系統。

在上例中，很顯然地無論系統的起始狀態爲何，均不影響以後系統中移位暫存的特性，因此我們可以不必介意其起始狀態爲何，故它是一個無起始狀態的序向系統。

3-3 等效狀態

雖然在設計一個序向系統的時候，必須先確定系統中的各種狀態的遷移，但事實上，對於外界的使用者而言，系統內部狀態的變化是並不重要的，他所眞正關心的是系統的輸入序列與輸出序列而已！因此一個（或兩個不同的）序向系統中的兩個狀態，如果對它們施以任何相同的輸入序列，都能夠得到相同的輸出序列的話，對於使用者來說，這兩個狀態是沒有什麼區別的。基於這個道理，我們就定義等效狀態如下：

【定義】 設有二個序向系統 M_a 及 M_b（或一個序向系統 M_a）中的兩個狀態 q_a 及 q_b，並設（x_1，x_2 ……… x_n）爲任意長度的輸入序列，若且唯若

$$\lambda_a(q_a ; x_1, x_2, \cdots\cdots x_n) = \lambda_b(q_b ; x_1, x_2 \cdots\cdots x_n) \ \forall n$$

則稱 q_a 與 q_b 爲等效狀態（Equivalent states）或不可區別的狀態（Indistinguish states），而記爲 $q_a \equiv q_b$。其中 λ_a 及 λ_b 分別代表 M_a 及 M_b 的輸出序列函數。

在定義中，等效狀態之所以記爲 $q_a \equiv q_b$，是由於此一等效關係事實上是符合代數中的相等關係（Equivalence relation）的，也就是說，等效狀態具有以下三個性質：

1. 反身性： $q_a \equiv q_a$。
2. 對稱性： 若 $q_a \equiv q_b$，則 $q_b \equiv q_a$。
3. 遞移性： 若 $q_a \equiv q_b$，$q_b \equiv q_c$，則 $q_a \equiv q_c$。

讀者可以從定義中推證出這三個性質。

茲再定義等效序向機如下：

【定義】 若序向系統 M_a 中的每一個狀態，在另一序向系統 M_b 中均有一等效狀態存在，則稱 M_a 與 M_b 為等效序向機 (Equivalent machine) ，而記為：

$$M_a \equiv M_b$$

等效序向機的等效關係同樣具有前面所說的三個性質。

茲舉一例以說明以上兩個定義。

【例 3 － 4】 設有二序向系統 M_a 及 M_b ，其狀態表如表 3-4所示，試說明其間的等效關係。

圖表 3 － 4 例 3 － 4 之狀態表

M_a	Y		Z	
y	x = 0	x = 1	x = 0	x = 1
q_1	q_3	q_2	0	1
q_2	q_1	q_2	1	0
q_3	q_1	q_2	0	1

M_b	Y		z	
y	x = 0	x = 1	x = 0	x = 1
p_1	p_1	p_2	0	1
p_2	p_1	p_2	1	0

【解】 (1)在 M_a 中，$q_1 \equiv q_3$ ：這可以由分析 M_a 的狀態表而得知，當輸入 x 為 0 時，q_1 會變為 q_3，而 q_3 則會變為 q_1 ，當輸入 x 為 1 時，則二者均會變為 q_2 ，且輸出完全相同，因此可知任意加以一輸入序列，則無論起始狀態在 q_1 或 q_3 ，均會得到相同的輸出

序列，此可由圖 3-1驗證得之。因此由定義可知，q_1 與 q_3 爲等效狀態。

(2)在 M_a 與 M_b 之間，$q_1 \equiv p_1 \equiv q_3$ ，$q_2 \equiv p_2$ ：這也可以由分析 M_a 與 M_b 之狀態表而得知，讀者可以用類似(1)中的方法驗證之。

(3) $M_a \equiv M_b$ ：由於 $q_1 \equiv q_3 \equiv p_1$ ，$q_2 \equiv p_2$ ，故由定義知 $M_a \equiv M_b$ 且 $M_b \equiv M_a$（反身性），因此 M_a 與 M_b 爲等效序向機。

	起始狀態		起始狀態	
狀態變化	q_1	q_3 q_1 q_2 q_2 q_1 ……	q_3	q_1 q_3 q_2 q_2 q_1 ……
輸入序列	0	0 1 1 0 ……	0	0 1 1 0 ……
輸出序列	0	0 1 0 1 ……	0	0 1 0 1 ……
	起始狀態		起始狀態	
狀態變化	q_1	q_2 q_2 q_2 q_1 ……	q_3	q_2 q_2 q_2 q_1 ……
輸入序列	1	1 1 0 ……	1	1 1 0 ……
輸出序列	1	0 0 1 ……	1	0 0 1 ……

圖 3-1　等效狀態

3-4 完全規定序向機之狀態表簡化法

在設計一序向系統時，由系統規定所直接寫出的狀態表中，往往其中包含了許多等效狀態在內，以致於使設計出來的系統似乎變得很繁雜，因此欲簡化整個系統的設計，就必須將各組等效狀態各自歸併成一項，以減少所需的狀態變數之個數。

本節中我們將導出一種方法以找出狀態表中所有的等效狀態，予以區分並合併之。在介紹之前，首先要定義一些名詞。

【定義】　一序向系統，若其狀態遷移表中之輸出欄及次狀態欄，對於每一項輸入變數及狀態變數可能的組合均已規定（ specified ）其值，則稱此系統爲一完全規定之序向機（ complete specified machine ）。

在本節中我們所介紹的方法僅適用於完全規定序向機，至於不完全規定序向

機之狀態表簡化法，則將於下節中再予介紹。

【定義】 一序向系統M中的二個狀態 q_a 及 q_b ，若對 q_a 及 q_b 分別施以長度為k的相同輸入序列，而所得的輸出序列也相同，則稱 q_a 及 q_b 為k次等效（k－equivalent）

【定義】 將狀態表中之各狀態按k次等效之關係予以歸類所得之區分稱為k次區分（k－partition）。

【定義】 經由k次區分所得之各狀態組稱之為k次等效組（k－equivalence classes）。

有了以上幾個定義以後，下面幾個定理似乎已是顯而易見的了。

【定理3－1】 任一序向系統，只要它包含兩種（或兩種以上）可能的輸出，則必可經由1次區分將它分為至少兩個1次等效組。

【證明】 由於1次區分是僅施以一個單一輸入，因此只須將狀態表中輸出不同的狀態予以分類即可，又由於該序向系統至少有兩種不同的輸出，，因此至少可分為兩個1次等效組，故得證。

【定理3－2】 對於一個n狀態的序向系統，若能連續施以不同的1次，，2次，……（n－1）次區分的話，則所得之各（n－1）次等效組必然僅包含一個狀態。

【證明】 由定理3－1可知1次區分必然包含至少2個1次等效組，又由於2次區分與1次區分結果不同，因此2次區分至少包含3個2次等效組，…………如此類推下去可知，（n－1）次區分至少包含了n個（n－1）次等效組，但是因為系統中僅有n個狀態，故每一個（n－1）次等效組必然僅能包含一個狀態而已！得證。

【定理3－3】 對於n狀態的序向系統至多僅能做n－1次連續且不同的區分。

【證明】 由定理3－2可以很容易地推論得本定理，故從略。

【定理3－4】 若一序向系統中各狀態之k次區分與（k＋1）次區分完全相同，則所得之各k次等效組中的每一狀態在該組中均互為等效狀態。

【證明】 假設 q_a 與 q_b 並非等效狀態，但卻在同一k次等效組中，則由定義知，至少有一個輸入序列有使 q_a 與 q_b 產生不同的輸出序列，因此也

就至少有一個 k 次等效組中的某兩個狀態，其次狀態不在同一個（ k ＋ 1 ）次等效組中，也就是說（ k ＋ 1 ）次區分與 k 次區分之結果必然不同，但此與已知條件矛盾，因此 q_a 與 q_b 必須爲等效狀態，故得證。

【推論 1】　若一 n 狀態序向系統中的兩個狀態 q_a 與 q_b 互爲（ n － 1 ）次等效，則 q_a 與 q_b 必爲等效狀態。

【推論 2】　任一 n 狀態之序向系統中的所有等效狀態均可經由（ n － 1 ）次區分而予以確定。

以上二個推論均可經由定理 3 － 2 及定理 3 － 4 推得，極爲顯而易見，故證明從略。

基於以上諸定義及定理，我們可以導出完全規定序向系統之狀態表簡化法，此法通稱爲 k 次區分法（ k － partition method ），步驟如下：

【簡化步驟】　設序向系統有 n 個狀態

1. 根據狀態表中輸出之不同而找出其 1 次等效組。
2. 執行 k 次區分（ k ＝ 2 ， 3 ，…… n － 1 ）找出 k 次等效組。
3. 若 k 次區分與（ k － 1 ）次區分結果完全相同則執行步驟 A ，否則回到步驟 2 繼續執行。
4. 將所得的 k 次等效組之各組分別合併成一個狀態，假設有 m 個等效組，則最後將有 m 個狀態（ m ≤ n ）。

【注意】　若 m ＝ n ，則表示該序向系統中沒有等效狀態存在，故無法將狀態表簡化。

茲舉二例以說明以上步驟。

【例 3 － 5】　例 3 － 2 爲一完全規定之序向系統，試簡化其狀態表（見表 3-1 ），並計算狀態變數之減少個數。

【解】　參考表 3-5

1. 根據輸出之不同而做 1 次區分如下：

　　Ⅰ： f （輸出爲 1 ）

　　Ⅱ： a ， b ， c ， d ， e ， g ， h （輸出爲 0 ）

2. 根據次狀態之不同而做 2 次區分如下：

　　Ⅰ： f

　　Ⅱ： c ， g （次狀態爲 Ⅱ ， Ⅰ ）

Ⅲ：a，b，d，e，h（次狀態爲Ⅱ，Ⅱ）

3. 根據次狀態之不同而做 3 次區分如下：

Ⅰ：f

Ⅱ：c，g（次狀態爲Ⅲ，Ⅰ）

圖表 3－5　*k* 次區分法

<table>
<tr><td>現在狀態</td><td colspan="2">次　一　狀　態</td><td>輸出</td><td>1 次</td><td>2 次</td><td>3 次</td><td>4 次</td></tr>
<tr><td></td><td>x = 0</td><td>x = 1</td><td></td><td>區分</td><td>區分</td><td>區分</td><td>區分</td></tr>
<tr><td>f</td><td>c</td><td>d</td><td>1</td><td>Ⅰ</td><td>Ⅰ</td><td>Ⅰ</td><td>Ⅰ</td></tr>
<tr><td>c</td><td>e</td><td>f</td><td>0</td><td rowspan="7">Ⅱ</td><td rowspan="2">Ⅱ</td><td rowspan="2">Ⅱ</td><td rowspan="2">Ⅱ</td></tr>
<tr><td>g</td><td>e</td><td>f</td><td>0</td></tr>
<tr><td>a</td><td>a</td><td>b</td><td>0</td><td rowspan="5">Ⅲ</td><td rowspan="2">Ⅲ</td><td rowspan="2">Ⅲ</td></tr>
<tr><td>e</td><td>a</td><td>b</td><td>0</td></tr>
<tr><td>d</td><td>g</td><td>h</td><td>0</td><td rowspan="3">Ⅳ</td><td rowspan="3">Ⅳ</td></tr>
<tr><td>h</td><td>g</td><td>h</td><td>0</td></tr>
<tr><td>b</td><td>c</td><td>d</td><td>0</td></tr>
</table>

Ⅲ：a，e（次狀態爲Ⅲ，Ⅲ）

Ⅳ：d，h，b（次狀態爲Ⅱ，Ⅲ）

4. 根據次狀態之不同而做 4 次區分如下：

Ⅰ：f

Ⅱ：c，g（次狀態爲Ⅲ，Ⅰ）

Ⅲ：a，e（次狀態爲Ⅲ，Ⅳ）

Ⅳ：d，h，b（次狀態爲Ⅱ，Ⅳ）

5. 由於 4 次區分與 3 次區分結果完全相同，因此所得之 4 個等效狀態組爲：（f），（c，g），（a，e），（d，h，b）。

6. 將這 4 個組Ⅰ，Ⅱ，Ⅲ，Ⅳ，分別用 p_1，p_2，p_3，p_4來代表，重寫該狀態表即得表 3-6 。

7. 簡化後之狀態表僅有 4 個狀態，因此僅需要 $\log_2 4 = 2$ 個狀態變數，故減少了 1 個狀態變數。

表 3－6　簡化後之結果

現在狀態	次一狀態		現在輸出	等效狀態
	x = 0	x = 1		
p_1	p_2	p_4	1	f
p_2	p_3	p_1	0	c,g
p_3	p_3	p_4	0	a,e
p_4	p_2	p_4	0	d,h,b

【例 3－6】　表 3-7 爲一個錯誤偵測序向系統的狀態表，試簡化之。

【解】　參考表 3-7。

表 3－7　例 3－6 之狀態表

現在狀態	次一狀態		輸出		1 次區分	2 次區分	3 次區分
	x＝0	x＝1	x＝0	x＝1			
q_0	q_0	q_4	1	0	Ⅰ	Ⅰ	Ⅰ
q_1	q_0	q_4	0	0	Ⅱ	Ⅱ	Ⅱ
q_2	q_1	q_5	0	0		Ⅲ	Ⅲ
q_3	q_1	q_5	0	0			
q_4	q_2	q_6	0	1	Ⅲ	Ⅳ	Ⅳ
q_5	q_2	q_6	0	1			
q_6	q_3	q_7	0	1			
q_7	q_3	q_7	0	1			

1. 首先根據輸出之不同而做 1 次區分如下：

　　Ⅰ：q_0　（輸出爲 1，0 ）

　　Ⅱ：q_1，q_2，q_3　（輸出爲 0，0 ）

　　Ⅲ：q_4，q_5，q_6，q_7　（輸出爲 0，1 ）

2. 根據次狀態之不同而做 2 次區分如下：

　　Ⅰ：q_0

Ⅱ：q_1 （次狀態爲Ⅰ，Ⅲ）

Ⅲ：q_2，q_3（次狀態爲Ⅱ，Ⅲ）

Ⅳ：q_4，q_5，q_6，q_7

3. 根據次狀態之不同再做3次區分如下：

Ⅰ：q_0

Ⅱ：q_1（次狀態爲Ⅰ，Ⅳ）

Ⅲ：q_2，q_3（次狀態爲Ⅱ，Ⅳ）

Ⅳ：q_4，q_5，q_6，q_7（次狀態爲Ⅲ，Ⅳ）

4. 由於3次區分與2次區分之結果完全相同，因此所得之4個等效狀態組爲：（q_0），（q_1），（q_2，q_3），（q_4，q_5，q_6，q_7）。

5. 將這4個組Ⅰ，Ⅱ，Ⅲ，Ⅳ，分別用p_1，p_2，p_3，p_4等四個狀態來代表，重寫狀態表卽得表3-8之結果。

表3－8　簡化後之結果

現在狀態	次一狀態		輸　出		等效狀態
	x＝0	x＝1	x＝0	x＝1	
p_1	p_1	p_4	1	0	q_0
p_2	p_1	p_4	0	0	q_1
p_3	p_2	p_4	0	0	q_2，q_3
p_4	p_3	p_4	0	1	q_4，q_5，q_6，q_7

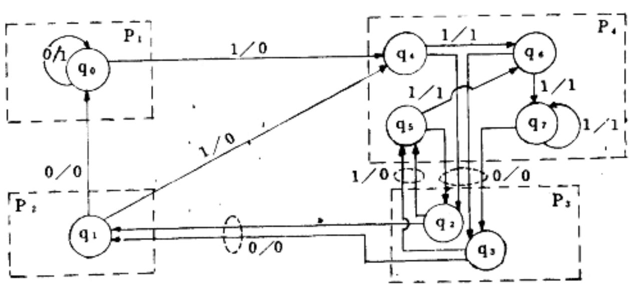

圖3-2　等效狀態圖

6. 本題之狀態在簡化前與簡化後之關係也可以用圖 3-2之狀態圖來代表，由圖中不難看出所謂等效狀態之意義。圖中的虛線方塊即代表各等效狀態組簡化後所成之一狀態。

3-5 不完全規定序向機之狀態表簡化法

首先定義不完全規定序向機如下：

【定義】 一序向系統，若其輸出函數及（或）次狀態函數並未完全規定（specified），而有隨意項（don't care）存在，則稱此系統為一不完全規定之序向機（Incomplete specified machine）。

造成不完全規定序向機之主要原因，是由於在設計一般的序向系統時，往往有許多輸入及狀態的組合，實際上是不可能發生的，既然不可能發生，因此在狀態表中我們就可以賦予隨意值或根本忽略它，均不會影響系統的正常工作。另一個原因，則是有些序向系統在某些狀況下，容許任意的輸出值，因此在狀態表中可以賦予隨意值，這些都是造成不完全規定序向機的原因。茲舉一例說明之：

【例3－7】 有一序向系統，包含四種可能的輸入：A，B，C，K 以及兩個輸出端 z_1，z_2。此序向系統在平時均處於“非檢查狀況”，其輸入為不定值。A，B，C中任何之輸入均不改變此狀況，但此時若輸入為K，則將使系統變為“檢查狀況”，且輸出 $z_1z_2 = 00$。在“檢查狀況”時，系統之輸入可以是A，B，C三者任意排列所得之序列（但前三個必須為不同符號），在此期間輸出為不定值，如此輸入3個以上的符號以後，系統即會在適當時機輸入一個K以回到非檢查狀況。在回到非檢查狀況時，若先前之輸入序列為ABCC……CK，則輸出為 $z_1z_2 = 01$，若輸入序列為CBAA……AK，則輸出將為 $z_1z_2 = 10$，任何其他的輸入序列，其輸出均將為 $z_1z_2 = 00$，試根據以上條件繪出此系統之狀態圖及狀態表。

【解】 茲以狀態1代表“非檢查狀況”，而以狀態 3-9分別表“檢查狀況”之各種可能狀態，所繪出之狀態圖如圖 3-3所示。根據此狀態圖，即可寫出其狀態表如表 3-9。在表中，可見有許多隨意項（don't care X）包含其中，每一項都代表了一個不可能的組合情況或任意的輸出值

。因此，這是一個標準的不完全規定序向機。

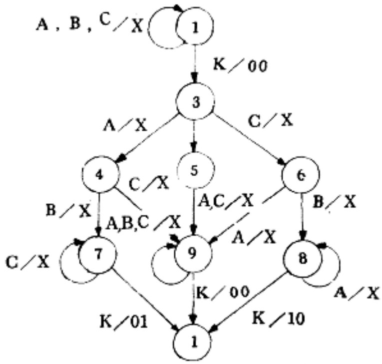

圖 3-3　不完全規定序向機

表 3-9　不完全規定狀態表

現在狀態	次一狀態				次一輸出			
	A	B	C	K	A	B	C	K
1	1	1	1	3	×	×	×	00
3	4	5	6	×	×	×	×	×
4	×	7	9	×	×	×	×	×
5	9	×	9	×	×	×	×	×
6	9	8	×	×	×	×	×	×
7	×	×	7	1	×	×	×	01
8	8	×	×	1	×	×	×	10
9	9	9	9	1	×	×	×	00

不完全規定序向機既然包含了許多隨意項，那麼我們當然可以依照設計者自己的意思賦予它們一些值，使其變為完全規定序向機後，再依照上節中的方法予

以簡化。這當然不失爲一個辦法，但它往往較爲麻煩，而且必然不會得到最佳的設計結果（即無法得到最少之狀態個數）。因此我們必須用另外一種方法來簡化它，這種方法通常稱之爲“調和法”（ Compatibility method ），在介紹調和法之前，首先要定義一些名詞。

【定義】 對某序向系統M施予一個輸入序列 x_{i_1} ， x_{i_2} …… x_{ik} ，若序列中每一輸入均能使M之次狀態爲規定之值，而不會導致隨意項之狀態，則此輸入序列稱爲“適用於”M的序列。

例： CKABCCKA 爲“適用於”例3 － 7 序向系統的輸入序列。

CKABCAKA 爲“不適用於”例3 － 7 序向系統的輸入序列。

【定義】 二個輸出序列 z_i 和 z_j ，若除了其中的隨意項以外，所有的對應值均相等，則稱此二輸出序列是“調和的”（ compatible ）。

例： $z_1 = 1010 \times 110 \times 0$

$z_2 = 10 \times 0 \times 11 \times \times 0$

$z_3 = 1 \times 0 \times 011110$

則 z_1 與 z_2 ， z_2 與 z_3 均爲“調和的”，但 z_1 與 z_3 却是“不調和的”。

【定義】 一序向系統M中的兩個狀態 q_i 和 q_j ，若且唯若施予任意“適用於” M的輸入序列，對此二狀態所產生的輸出序列 z_i 及 z_j 均爲調和的，則稱此二狀態 q_i 及 q_j 是“調和的”（ Compatible ），否則稱之爲“不調和的”（ incompatible ）。

【定義】 若一個狀態集合中的每一對狀態均是調和的，則稱此狀態集合爲一“調和組”（ compatibility class ）。

【定義】 若在一調和組中加上任一不屬於此調和組中的狀態，均能使其喪失調和組之性質，則稱此調和組爲一個最大調和組（ Maximal compatible class ）。

【定義】 若序向系統M之二個狀態 q_i 及 q_j 合於下列二個條件，則稱 q_i 與 q_j 爲“簡單調和”的（ simply compatible ）：

1. 在狀態表中除隨意項外，二者的輸出欄各行完全相同。
2. 在狀態表中除隨意項外，二者的次狀態欄各行完全相同或者其次狀態爲 q_i ， q_j 或 q_j ， q_i 。

【定義】 若序向系統M之二個狀態q_i及q_j，在狀態表中除隨意項外，其輸出欄至少有一行不一樣，則稱此二狀態爲“簡單不調和”的（simply incompatidle ）。

例： 在表3-9中，狀態4與5，6與7爲簡單調和的，而狀態7與8，8與9則爲簡單不調和的。

【定理3－5】 若序向系統中的二狀態q_i及q_j爲調和的，則對任何單一輸入x，必有如下二個性質：

1. $\lambda(q_i;x)=\lambda(q_j;x)$，若二者均不爲隨意項。
2. $\delta(q_i;x)$與$\delta(q_j;x)$爲調和的，若二者均不爲隨意項。

【證明】 第一個性質可由定義中直接推得，第二個性質之證明則頗爲繁瑣，本書將從略。

由定理3－5可知，簡單不調和的二狀態必然不可能是調和的，而簡單調和的二狀態則極可能是調和的，但調和的二狀態却不一定必然是簡單調和的。這二者之間的差別與關係，值得讀者仔細玩味。

瞭解了以上諸定義定理後，就可以介紹所謂的調和法了，其簡化步驟如下：

【簡化步驟】 設不完全規定序向機M有n個狀態：

1. 首先繪出“調和性檢查表”，此表是一個階梯狀的表格，表中的每一列分別對應於狀態q_2，q_3……q_n，每一行則分別對應於狀態q_1，q_2，……q_{n-1}，如圖3-4所示。
2. 檢查每一對狀態q_i及q_j，在表中對應的空格中填入下列符號之一：

   (a)若q_i與q_j爲簡單不調和的，則填入“×”號。

   (b)若q_i與q_j爲簡單調和的，則填入“✓”號。

   (c)若q_i與q_j不屬於以上二種，則填入其不同的次狀態組。
3. 自上至下，自左至右，反覆檢查表中所填入的各次狀態組，如果表中的第q_i-q_j格已被填入“×”號，則在所有包含q_i-q_j爲次狀態組的格中，均填入“×”號。
4. 重覆步驟3直至無法再填入“×”號爲止。
5. 此時表中未被填入“×”號的各格，每一格即代表一個q_i-q_j狀態的調和組，列出所有此類調和組。

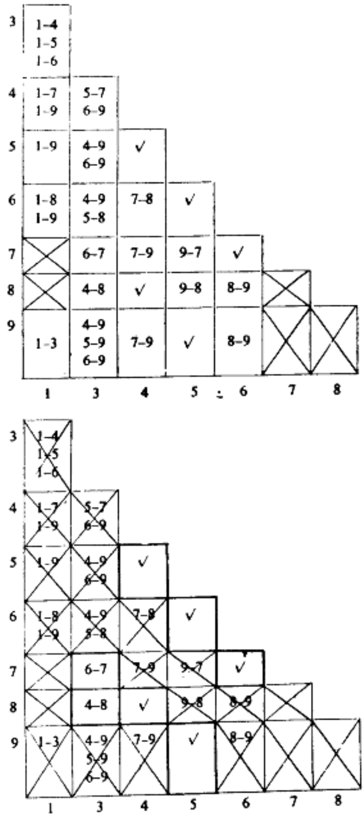

圖 3-4　不完全規定狀態表之簡化

6. 以這些調和組爲基礎，根據定義，找出包含 3 個，4 個……乃至（ m－1 ）個狀態的調和組，每找出一個較大的調和組，即將它所包含的所有較小的調和組删去。