



# 序向邏輯電路之設計

## —計數器

# 概 述

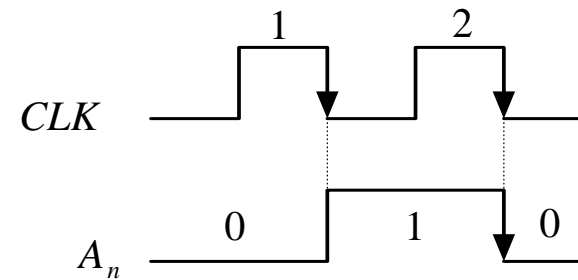
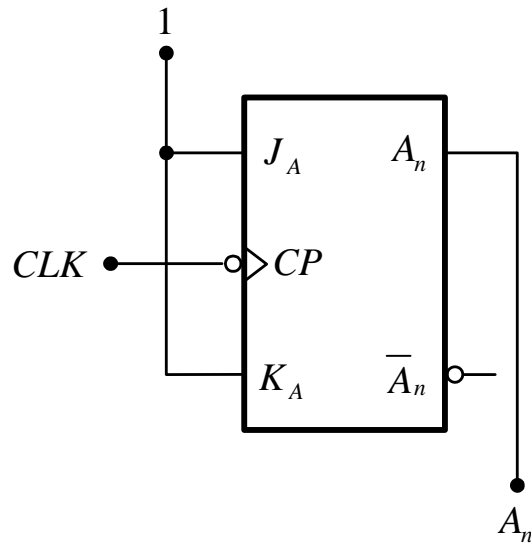
- ◆ **計數器** (Counter) 在數位電路中是一種應用相當廣範之**序向邏輯電路**，它是由**正反器**加上**組合邏輯**所構成，當將**週期性的時序脈波**送到計數器之輸入端時，它便可以由各個正反器之**輸出**來計算出在一定時間內，電路所**接收之脈波數目**，故計數器可同時用來作**脈波計數**與**除頻**之用途。
- ◆ 依所有正反器之**脈衝控制端** ( $CP$ ) 之連接情況，計數器可分為**非同步計數器** (Asynchronous Counter) 與**同步計數器** (Synchronous Counter) 兩大類。
- ◆ 當時脈訊號**未同時連接至正反器之時脈控制端** ( $CP$ ) 時，稱為**非同步計數器**，即此種計數器之所有正反器，**不會同時受到時脈訊號觸發** (Trigger)。
- ◆ 當時脈訊號**同時連接至所有正反器之時脈控制端** ( $CP$ ) 時，稱為**同步計數器**，即此種計數器之所有正反器，**皆同時接受時脈訊號之觸發**。
- ◆ 對具  $n$  個正反器之計數器而言，因**每一個正反器**可用來表示**一個變數** (Variable)，若有  $n$  個變數可供使用，且每個變數皆可以**補數** (Complement) 或**非補數** (Noncomplement) 之型態出現，故計數器 (不論同步或非同步計數器) 之**最大計數模式**  $N_{\max} = 2^n$  個 (即  $n = \lceil \log_2 N \rceil$ )。

## 計數模數為 $N = 2^n$ 之非同步上數(漣波)計數器

- ◆ 最典型之非同步計數器是將計數脈波(CP)加在第一個正反器之 CP 端，接著以每一級正反器的輸出，作為下一級正反器 CP 之輸入，因所有正反器不會同時受到計數脈波之觸發，即此種計數器僅第一級正反器，隨時脈訊號轉態之同時，才會受到觸發，而第二級以後之正反器，隨前一級輸出轉態之同時，才會受到觸發，這種類似波浪式的觸發方式，故此種計數器亦可稱為漣波計數器 (Ripple Counter)。
- ◆ 因計數器之計數模式  $N$  與正反器數目  $n$  成指數函數 (Exponential Function) 之正比關係 (即  $N \leq 2^n$ )，因此本節將分 4 個部分討論計數模式  $N = 2^n$  之上數非同步計數器的設計方法。

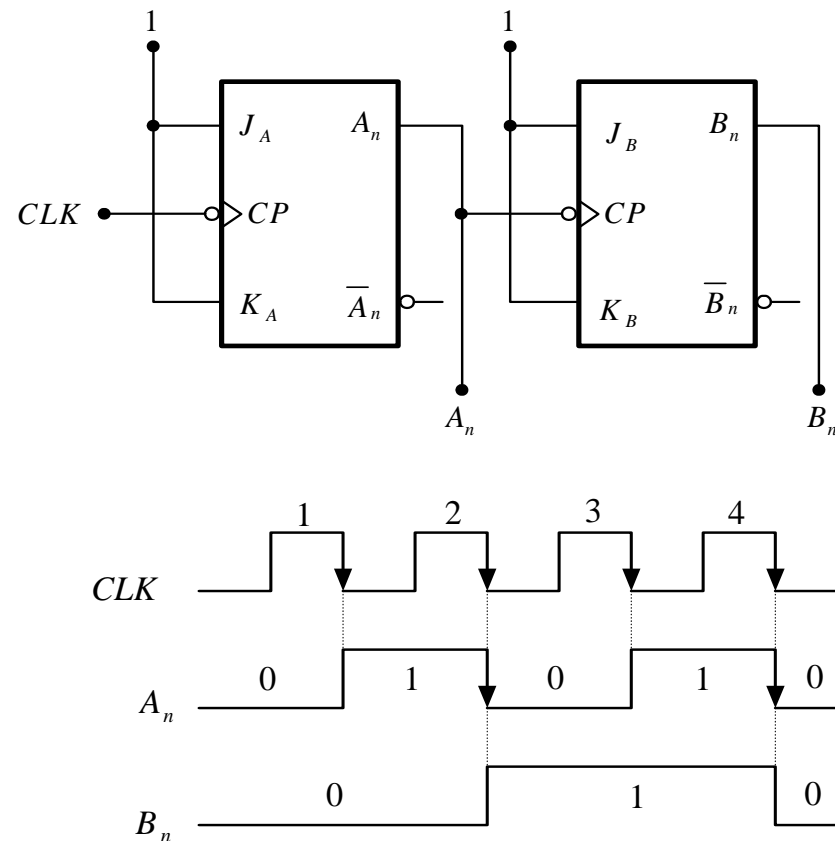
## 二進上數非同步計數器( $N = 2^1 = 2$ )

- ◆ 若將  $JK$  正反器之  $J$  與  $K$  輸入端，皆設定為邏輯 1，而將計數脈波 ( $CLK$ ) 連接至  $JK$  正反器之  $CP$  端，即可在正反器之輸出  $A_n$  得到一個 2 進上數連波計數器( $N = 2^1 = 2$ )，如下圖所示。



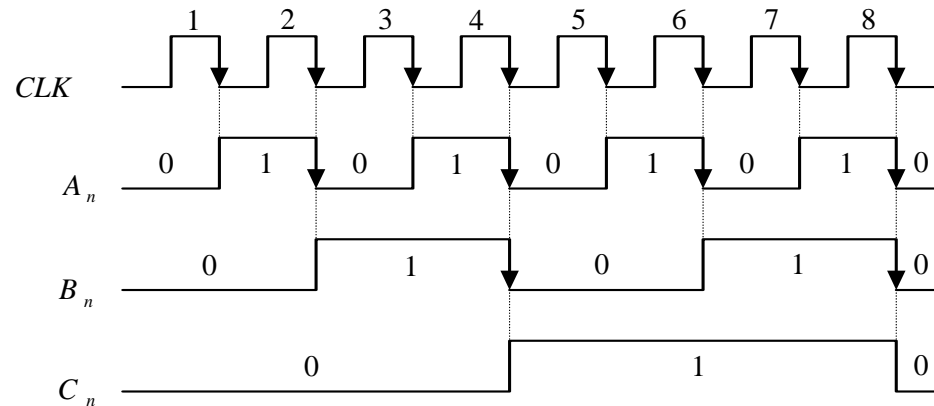
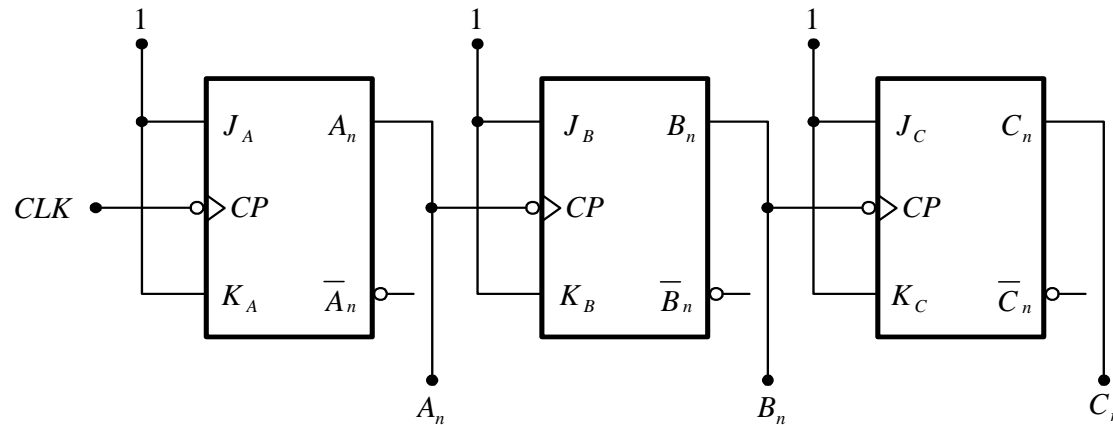
## 四進上數非同步計數器( $N = 2^2 = 4$ )

- ◆ 若將所有  $JK$  正反器之  $J$  與  $K$  輸入端，皆設定為邏輯 1，而將計數脈波 ( $CLK$ ) 連接至第 1 個正反器之  $CP$  端，接著將輸出  $A_n$  連接至第 2 個正反器之  $CP$  端，即可在正反器之輸出  $B_n A_n$  得到一個 4 進上數連波計數器( $N = 2^2 = 4$ )，如下圖所示。



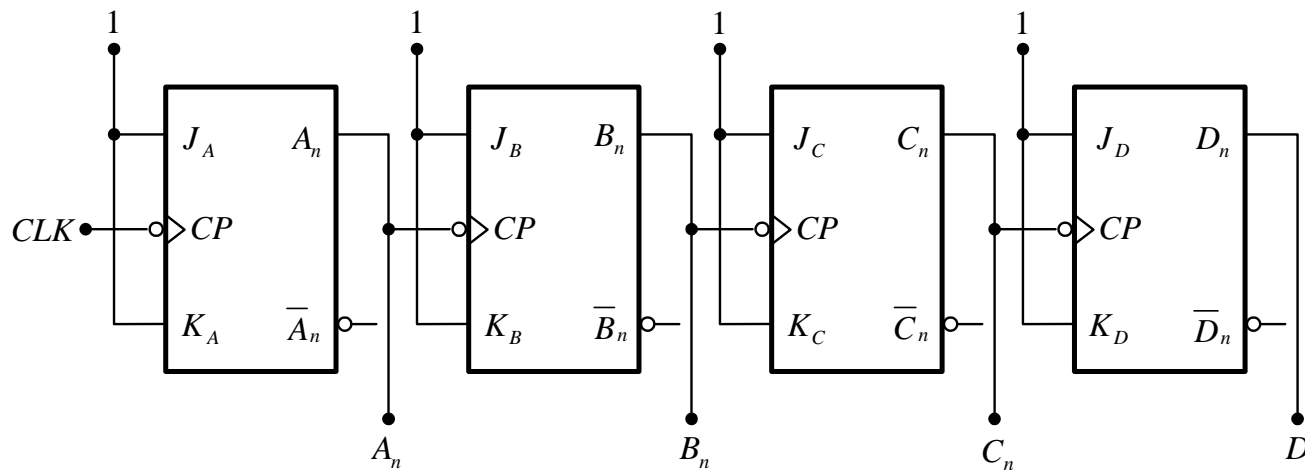
## 八進上數非同步計數器( $N = 2^3 = 8$ )

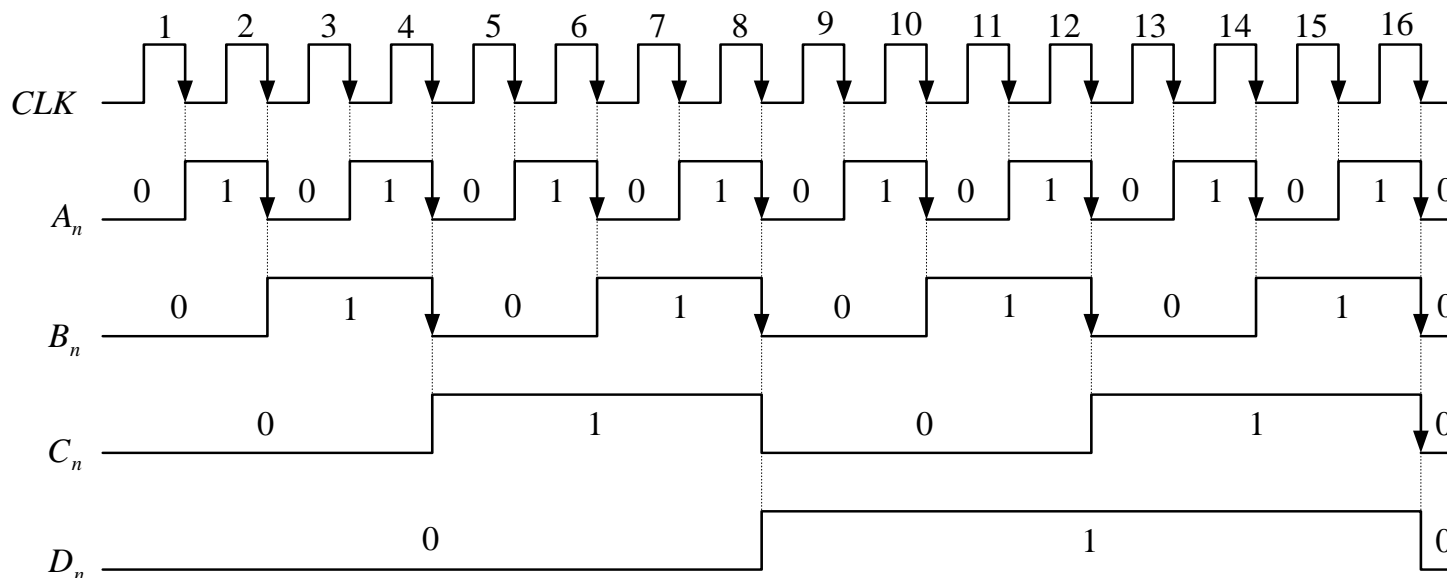
- ◆ 若將所有  $JK$  正反器之  $J$  與  $K$  輸入端，皆設定為邏輯 1，而將計數脈波 ( $CLK$ ) 連接至第 1 個正反器之  $CP$  端，而將輸出  $A_n$  連接至第 2 個正反器之  $CP$  端，接著將輸出  $B_n$  連接至第 3 個正反器之  $CP$  端，即可在正反器之輸出  $C_n B_n A_n$  得到一個 8 進上數連波計數器，如下圖所示。



## 十六進上數非同步計數器( $N = 2^4 = 16$ )

- ◆ 若將所有  $JK$  正反器之  $J$  與  $K$  輸入端，皆設定為邏輯 1，而將計數脈波 ( $CLK$ ) 連接至第 1 個  $JK$  正反器之  $CP$  端，而將輸出  $A_n$  連接至第 2 個  $JK$  正反器之  $CP$  端，接著將輸出  $B_n$  連接至第 3 個  $JK$  正反器之  $CP$  端，最後再將輸出  $C_n$  連接至第 4 個  $JK$  正反器之  $CP$  端，即可在正反器之輸出  $D_n C_n B_n A_n$  得到一個 16 進上數連波計數器( $N = 2^4 = 16$ )，如下圖所示。



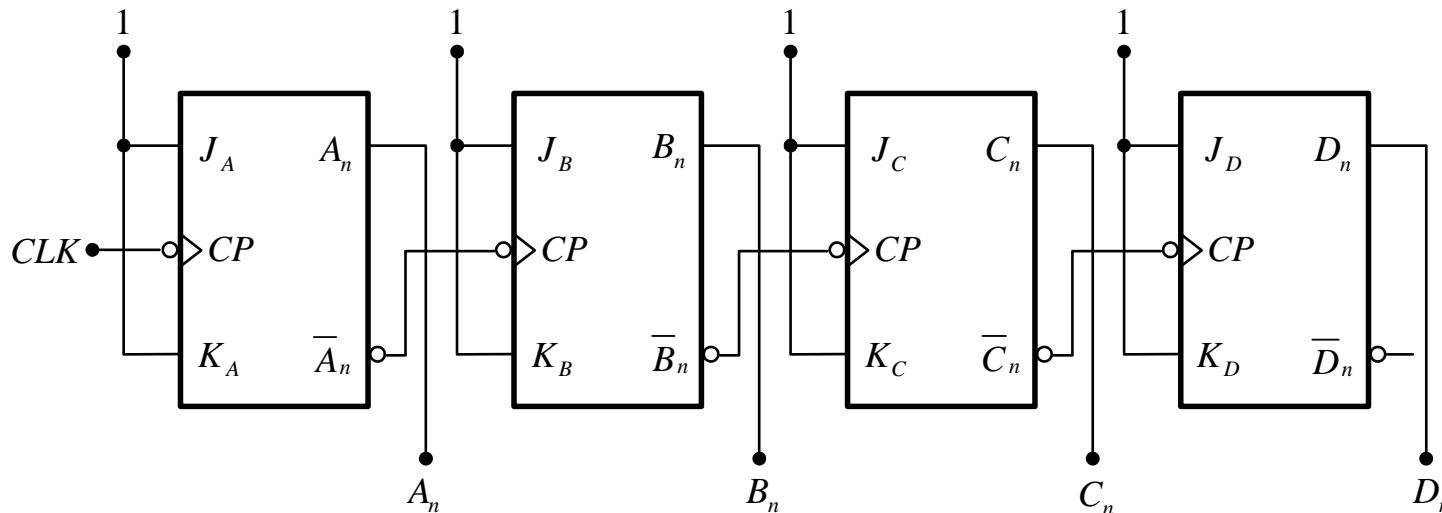


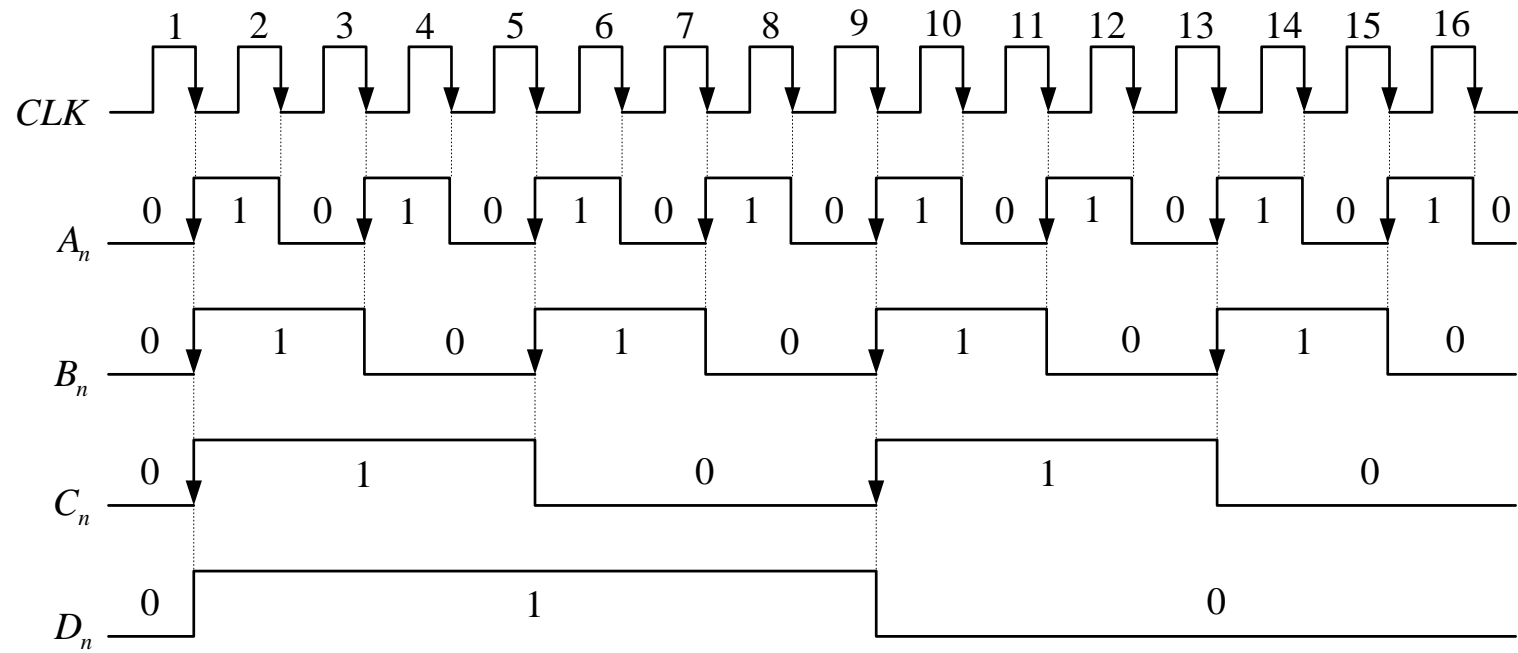
- ◆ 十六進上數漣波計數器，不僅可用來作為計算在一定時間內所接收脈波數目之用途，亦可作為除頻之用。即若從  $A_n$  取出訊號，可得除 2 之除頻器，即  $A_n$  之頻率為輸入脈波 ( $CLK$ ) 之  $\frac{1}{2}$ ；若從  $B_n$  取出訊號，可得除 4 之除頻器，即  $B_n$  之頻率為輸入脈波 ( $CLK$ ) 之  $\frac{1}{4}$ ；依此類推，若從  $D_n$  取出訊號，可得除 16 之除頻器，即  $D_n$  之頻率為輸入脈波 ( $CLK$ ) 之  $\frac{1}{16}$ 。
- ◆ 計數模式  $N = 2^n$  之漣波計數器的構造相當簡單(不需外加任何組合邏輯電路)，且計數模式  $N$  與正反器成指數函數關係，欲實現更多計數模式之漣波計數器，僅需增加正反器之數目。



## 計數模式為 $2^n$ 之下數連波計數器

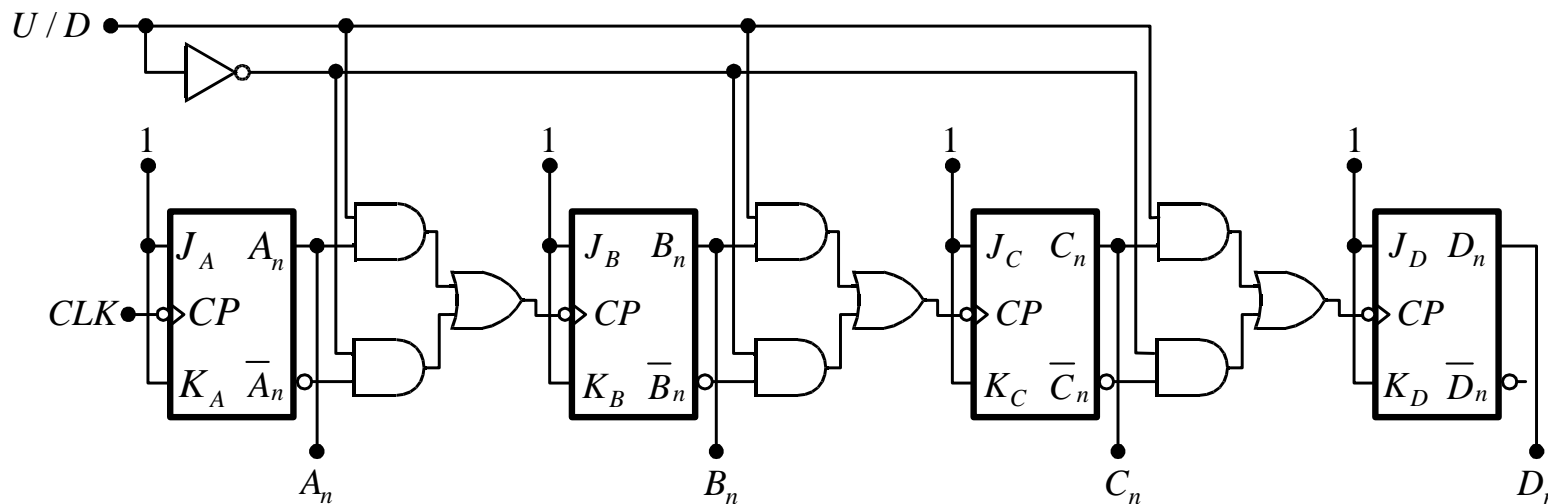
- ◆ 若將前級之反相輸出 $\bar{Q}_n$ 連接至下一級之脈衝輸入端 (CP)，則可得到計數模式為 $2^n$ 之下數連波計數器。
- ◆ 因計數模式為 $2^n$ 的下數連波計數器之電路結構與上數連波計數器相似，為避免重複起見，本節僅討論利用 4 個 JK 正反器所組成之十六進下數連波計數器( $N = 2^4 = 16$ )，如下圖所示。





## 計數模式為 $2^n$ 之上 / 下數連波計數器

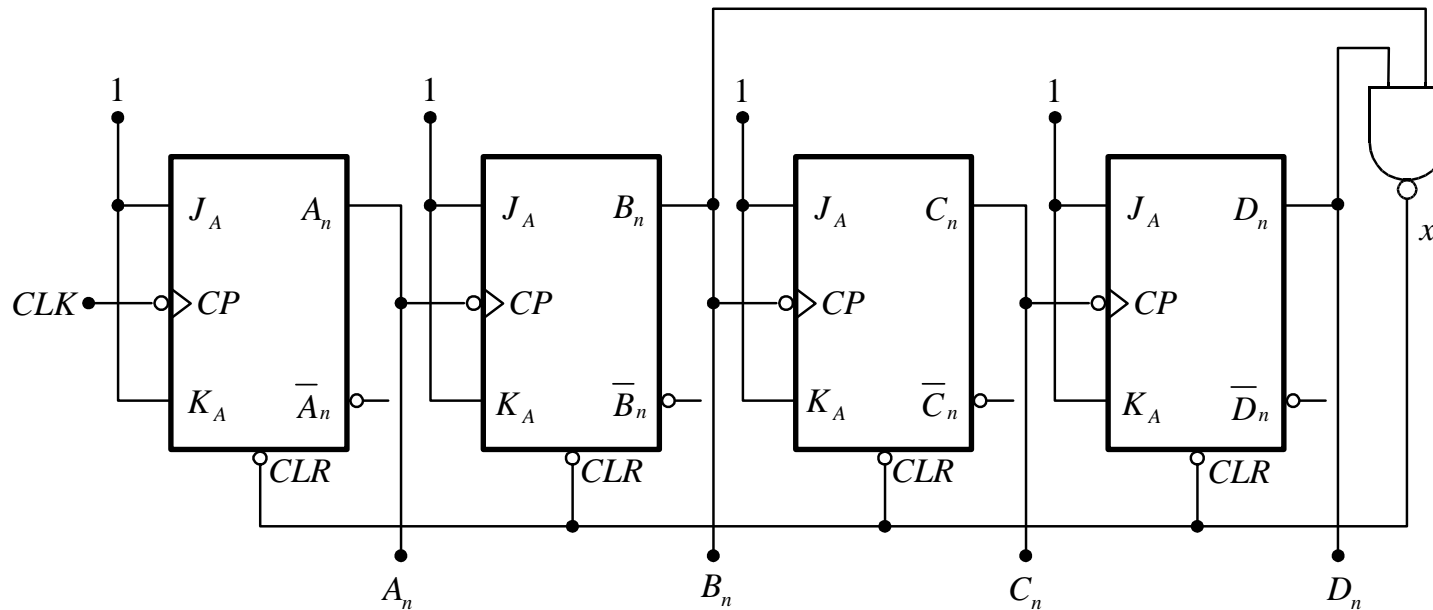
- ◆ 若加上 1 個控制  $U/D$  ( 上 / 下數之控制 ) 輸入端，即可結合十六進上數與下數連波計數器，以構成一個可同時實現十六進上 / 下數連波計數器，如下圖所示。

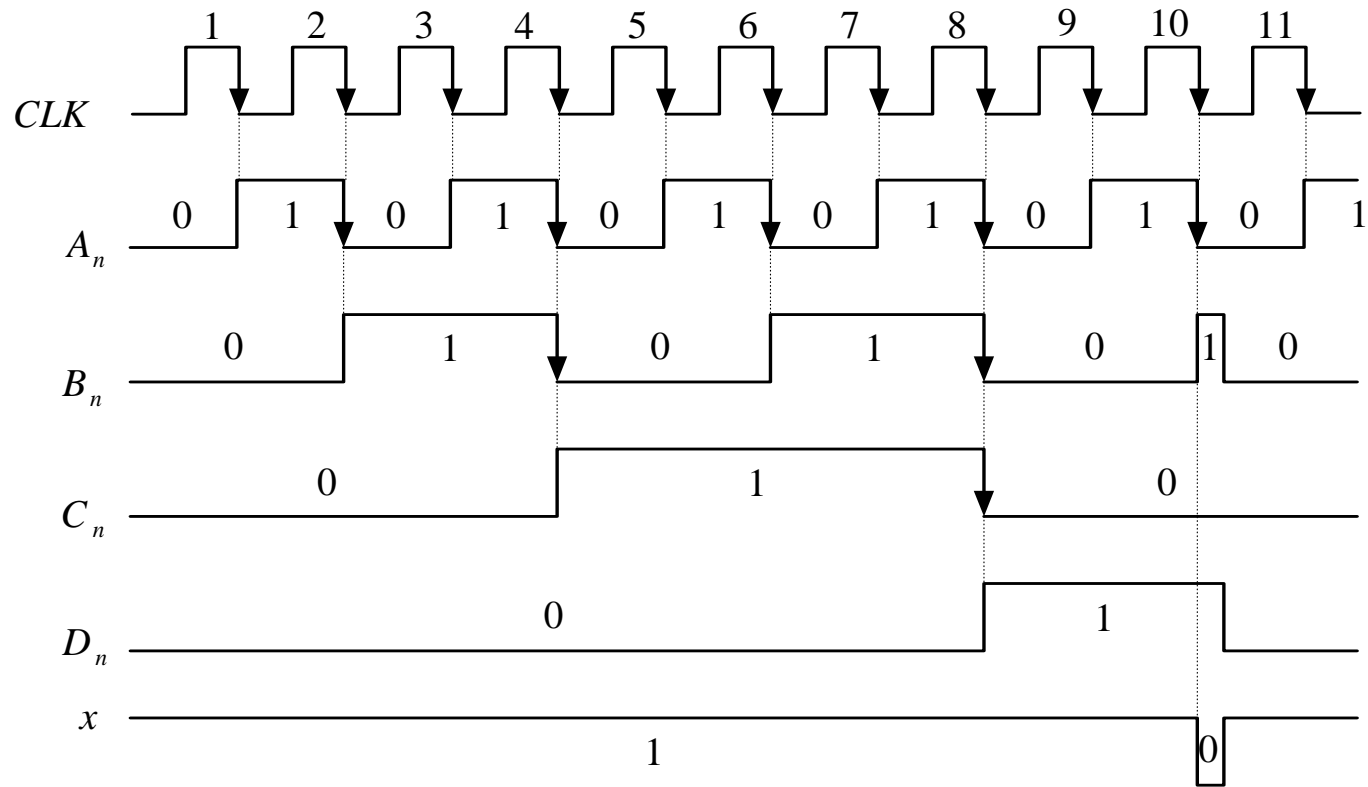


- ◆ 當  $U/D = 1$  時，除第一級  $JK$  正反器外，前一級之非反相輸出  $Q_n$  皆連接至下一級之脈衝輸入端 ( $CP$ )，故上圖將可執行 16 進上數之計數功能。
- ◆ 當  $U/D = 0$  時，除第一級  $JK$  正反器外，前一級之反相輸出  $\bar{Q}_n$  皆連接至下一級之脈衝輸入端 ( $CP$ )，故上圖亦可執行 16 進下數之計數功能。

# 利用清除輸入端設計之十進上數連波計數器

- ◆ 若分別利用  $JK$  正反器之清除 (Clear) 與預置 (Preset) 等兩個輸入端，採用強迫中斷計數狀態的方法，以設計十進上數連波計數器之邏輯電路。
- ◆ 若將 16 進上數連波計數器之正反器的  $B_n$  與  $D_n$  端，當作反及閘 (NAND 閘) 之輸入端，且將反及閘之輸出  $x$  連接至所有正反器之清除 (CLR) 輸入端，即可得到十進上數連波計數器 (10 個計數模式之計數器，必須使用 4 個 ( $n = \lceil \log_2 10 \rceil = 4$ ) 正反器才可實現，如下圖所示。

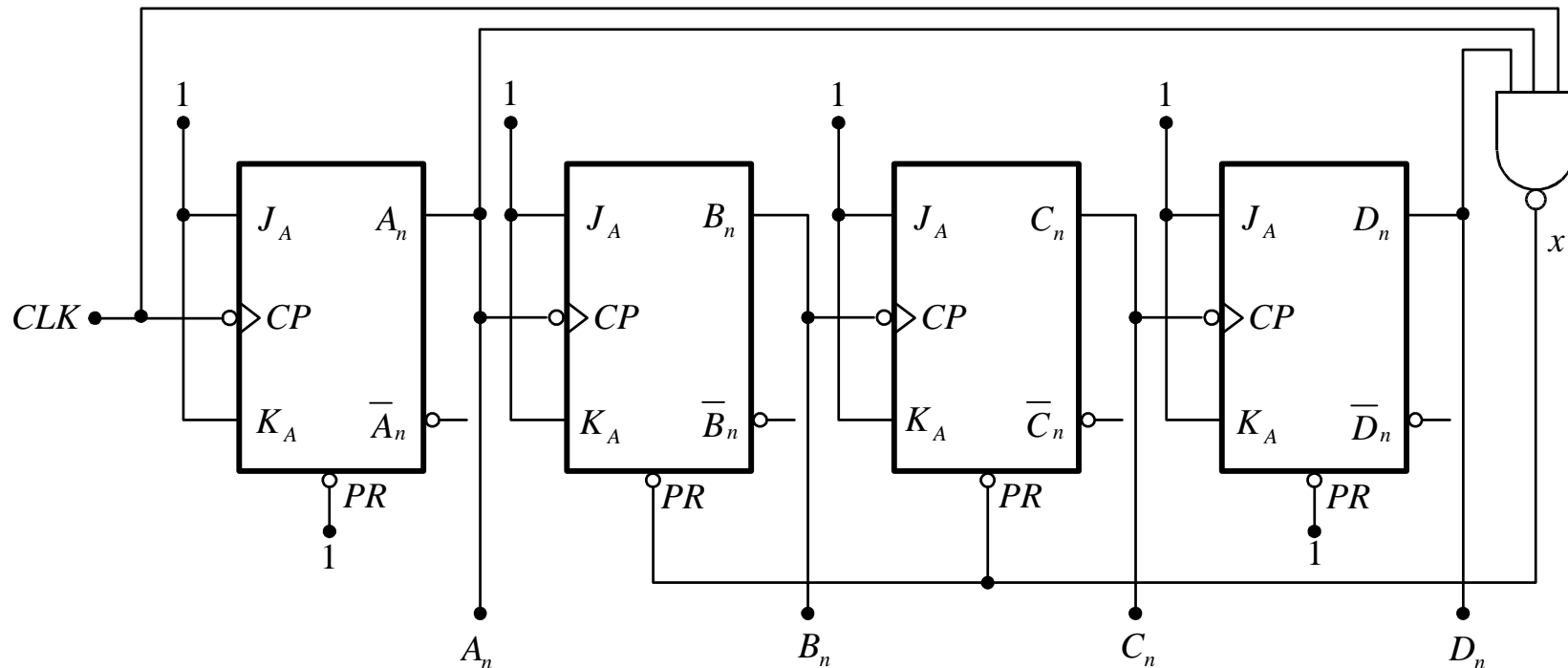


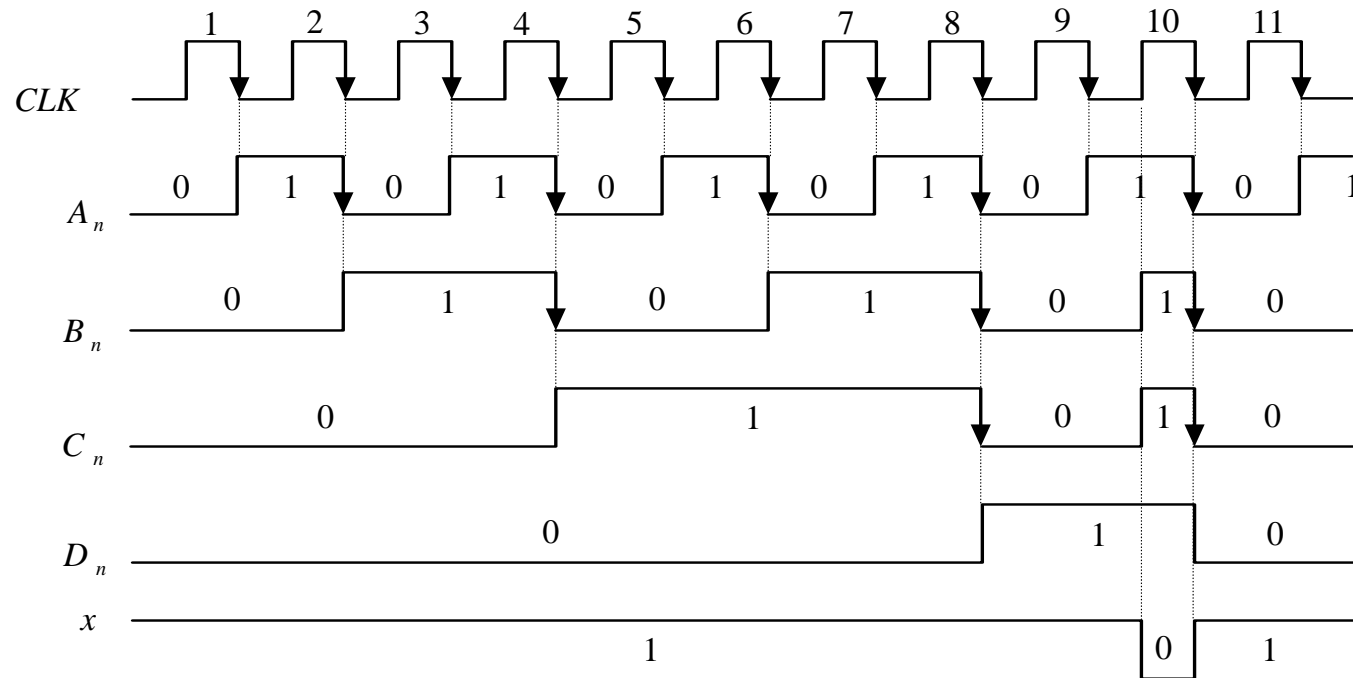


- ◆ 利用  $JK$  正反器之清除 (CLR) 來強制中斷計數狀態的方法，可以設計任意計數模數 ( $N < 2^n$ ) 之連波計數器。

# 利用預置輸入端設計之十進上數連波計數器

- ◆ 若將 16 進上數連波計數器之正反器的  $A_n$  與  $D_n$  端，當作反及閘 (NAND 閘) 之輸入端，且將反及閘之輸出  $x$  連接至  $B$  與  $C$  正反器之預置(PRE) 輸入端，亦可得到十進上數連波計數器，如下圖所示。





◆ 比較上面之十進上數連波計數器的輸出波形可知，此兩個計數電路皆會有下列之**誤差**出現，使用者必須**注意此誤差**情況，以免所得之**計數狀態與實際不符合**的結果。

1. 用**清除輸入端**來中斷計數狀態，以實現**十進上數連波計數器**，當計數狀態到 **1001** 時，會有**短暫時間**（此時間的長短是以正反器之**傳遞延遲時間**來決定）進入 **1010** 之狀態後，再回到 **0000** 之狀態。
2. 使用**預置輸入端**來中斷計數狀態，以實現**十進上數連波計數器**，當計數狀態到 **1001** 時，會有**短暫時間**進入 **1111** 之狀態後，再回到 **0000** 之狀態。

# 漣波計數器之最高操作頻率

- ◆ 漣波計數器之**每一級正反器的觸發**皆會受到**前一級之影響**，因此計數模式愈多，則須**串接愈多級**之正反器，故**時間延遲**會累積愈長。若每個正反器之**時間延遲**皆為 $t_d$ ，則串接 $n$ 個正反器之**累積時間延遲**為 $t_T = n \times t_d$ 。
- ◆ 對**週期性之脈波訊號**而言，當時間延遲 $t_T$ 大於計數器之輸入脈波訊號的週期 $\left(T = \frac{1}{f}\right)$ 時，便可能會造成計數器之**誤動作**，故漣波計數器無法滿足**高速操作**之要求，而漣波計數器之**最高操作頻率** $f_{\max} = \frac{1}{n \times t_d} = \frac{1}{t_T}$ 。
- ◆ 漣波計數器（非同步計數器）之**構造相當簡單**（僅使用正反器，而幾乎**不需任何的組合邏輯電路**）為此種計數器之**最大優點**，但**操作頻率受到相當大之限制**為此種計數器之**最大缺點**。
- ◆ 下一節將討論**同步計數器**（計數脈波同時加在每一個正反器之脈波輸入端）之設計方法，此種計數器之**時間延遲相對短了相當多**，故可**操作於比較高頻之計數脈波**，但需額外**增加一些組合邏輯**來控制正反器之輸出狀態，故需**較複雜構造**為同步計數器之缺點。



# 同步計數器之設計(概述)

- ◆ 同步計數器之計數脈波 ( $CLK$ ) 同時加在每一個正反器之脈波輸入端 ( $CP$ )，因此所有正反器之觸發是在同一時間完成(任何型式之正反器，皆可用來實現同步計數)，故此種計數器之優點為時間延遲較短 (僅有一個正反器加上控制邏輯閘之時間延遲，而與計數器之位元數目無關)。
- ◆ 因同步計數器之所有正反器是同時被觸發，因此需使用額外組合邏輯電路，以控制每個正反器之輸出狀態，即控制正反器之輸出是否隨計數脈波作轉態，以得到實際所需之輸出狀態，故往往需較複雜之電路結構為此種計數器之缺點。
- ◆ 本節討論計數模數  $N \leq 2^n$  (即  $N = 2^n$  與  $N < 2^n$  之任意模式) 的同步計數器的設計方法。
- ◆ 接著說明使用一個控制輸入端，結合上數與下數同步計數器，以組成一個可同時執行上數與下數之同步計數器。
- ◆ 最後使用不同型式之正反器，以設計計數狀態為亂數(即計數狀態不依二進位順序變化)之同步計數器的設計方法。

## 計數模式 $N = 2^n$ 之上數同步計數器

- ◆ 因同步計數器中的所有正反器之觸發皆在同一時間完成，故必須依第八章所討論之序向邏輯電路的設計方法來完成。
- ◆ 同步計數器之設計須依實際所需之計數模式，以得知計數器所須之正反器數目後，再列出計數器之狀態表 (State Table)，接著依計數模式來進行狀態指定 (State Assignment) 後，再依所使用正反器之激勵表 (Excitation Table)，配合狀態指定後之狀態表，以求取計數器之轉態表 (Transition Table) 後，接著再利用卡諾圖化簡每一個正反器輸入之最簡布林代數，最後再依據所選用正反器型式與正反器輸入之最簡布林代數，以繪出所求之同步計數器的邏輯電路圖。
- ◆ 使用  $JK$  正反器來設計序向邏輯電路，所需外加之組合邏輯較為簡單，故除非特別聲明，本節皆採用  $JK$  正反器來設計同步計數器為主。
- ◆ 本節將分 3 個部分討論計數模式  $N = 2^n$  上數同步計數器之設計方法，即說明分別使用 2、3 與 4 個  $JK$  正反器，以設計計數模式為 4、8 與 16 之上數同步計數器的設計方法。

## 四進上數同步計數器

- ◆ 若使用 2 個 JK 正反器，即可用來實現計模式  $N = 2^2 = 4$  之同步計數器，即計數模式為  $00 \rightarrow 01 \rightarrow 10 \rightarrow 11 \rightarrow 00$  之計數器。首先將兩個 JK 正反器之輸出標示為  $A_n$  與  $B_n$  之符號，並依據計數模數列出 4 進上數同步計數器之狀態表(State Table)與利用 JK 正反器之激勵表(Excitation Table)，以列出所求計數器之轉態表 (Transition Table)，如下表所示。

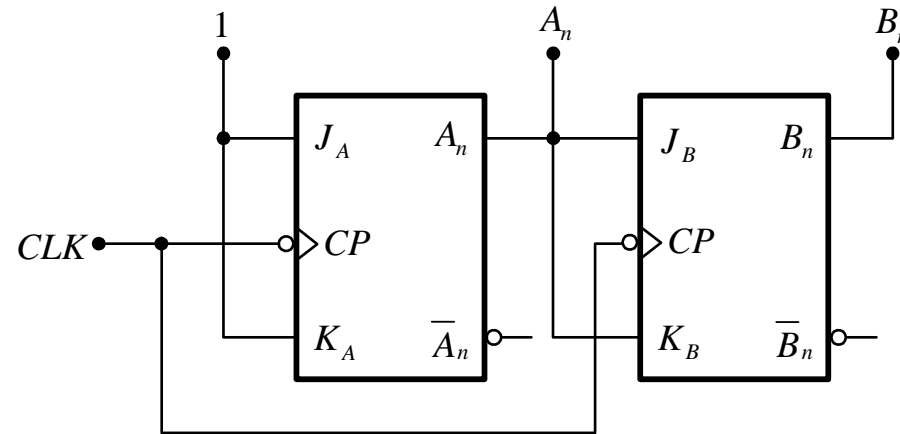
目前狀態			下一狀態			正反器之輸入			
狀態	$B_n$	$A_n$	狀態	$B_{n+1}$	$A_{n+1}$	$J_B$	$K_B$	$J_A$	$K_A$
0	0	0	1	0	1	0	×	1	×
1	0	1	2	1	0	1	×	×	1
2	1	0	3	1	1	×	0	1	×
3	1	1	0	0	0	×	1	×	1

- ◆ 利用卡諾圖分別對  $J_A$ 、 $K_A$ 、 $J_B$  與  $K_B$  等 JK 正反器輸入端進行邏輯化簡，即可得到這些數之最簡布林代數式如下：

$$J_A = K_A = 1$$

$$J_B = K_B = A_n$$

- ◆ 使用 2 個  $JK$  正反器與化簡所得之  $J_A$ 、 $K_A$ 、 $J_B$  與  $K_B$  的最簡布林代數，即可得到 4 進上數同步計數器之邏輯電路，如下圖所示。



# 八進上數同步計數器

- ◆ 使用 3 個  $JK$  正反器，即可用來實現計模式  $N = 2^3 = 8$  之同步計數器，即計數模式為  $000 \rightarrow 001 \rightarrow 010 \rightarrow 011 \rightarrow 100 \rightarrow 101 \rightarrow 110 \rightarrow 111 \rightarrow 000$  之計數器。首先將 3 個  $JK$  正反器之輸出標示為  $A_n$ 、 $B_n$  與  $C_n$  之符號，並依據計數模數列出所求計數器之狀態表與利用  $JK$  正反器之激勵表，以列出所求計數器之轉態表，如下表所示。

目前狀態				下一狀態				正反器之輸入					
狀態	$C_n$	$B_n$	$A_n$	狀態	$C_{n+1}$	$B_{n+1}$	$A_{n+1}$	$J_C$	$K_C$	$J_B$	$K_B$	$J_A$	$K_A$
0	0	0	0	1	0	0	1	0	×	0	×	1	×
1	0	0	1	2	0	1	0	0	×	1	×	×	1
2	0	1	0	3	0	1	1	0	×	×	0	1	×
3	0	1	1	4	1	0	0	1	×	×	1	×	1
4	1	0	0	5	1	0	1	×	0	0	×	1	×
5	1	0	1	6	1	1	0	×	0	1	×	×	1
6	1	1	0	7	1	1	1	×	0	×	0	1	×
7	1	1	1	0	0	0	0	×	1	×	1	×	1

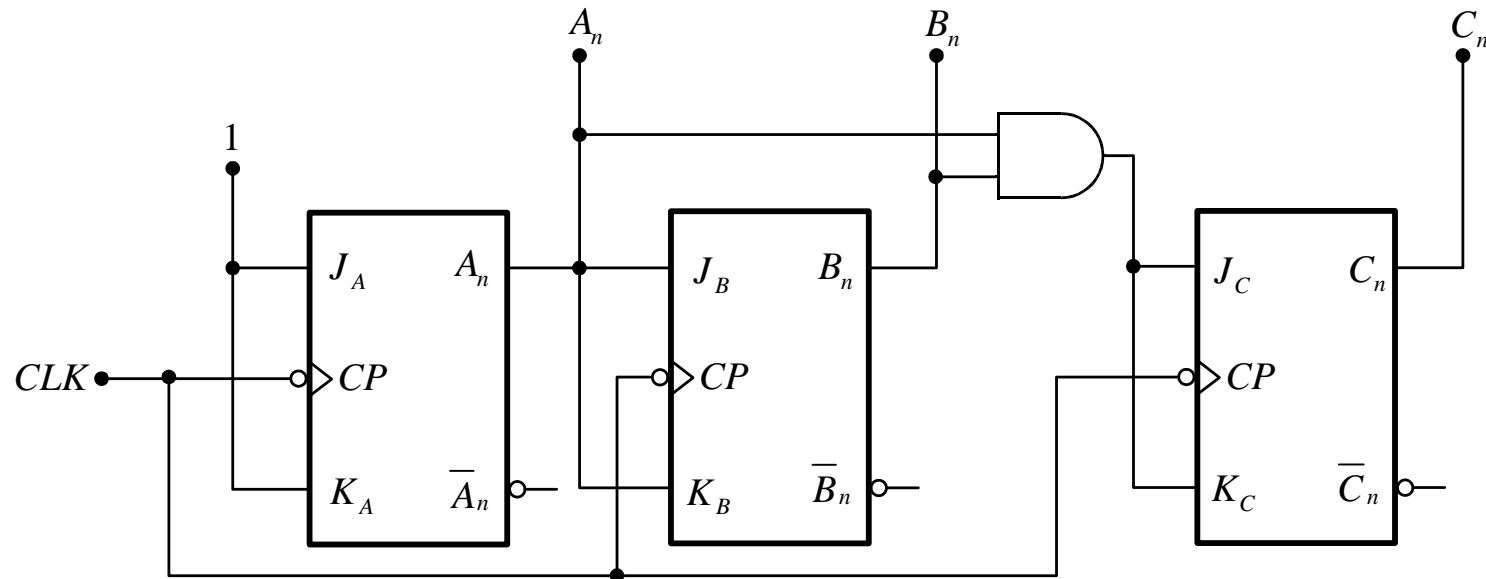
- ◆ 利用卡諾圖分別對  $J_A$ 、 $K_A$ 、 $J_B$ 、 $K_B$ 、 $J_C$  與  $K_C$  等  $JK$  正反器之輸入端進行邏輯化簡，即可得到這些輸入之最簡布林代數式如下：

$$J_A = K_A = 1$$

$$J_B = K_B = A_n$$

$$J_C = K_C = B_n \cdot A_n$$

- ◆ 最後使用 3 個  $JK$  正反器與化簡所得之每一個正反器之輸入  $J_A$ 、 $K_A$ 、 $J_B$ 、 $K_B$ 、 $J_C$  與  $K_C$  的最簡布林代數，即可得到 8 進上數同步計數器之邏輯電路，如下圖所示。



# 十六進上數同步計數器

- ◆ 使用 4 個 *JK* 正反器，即可實現計數模式  $N = 2^4 = 16$  之同步計數器，即計數模式為  $0000 \rightarrow 0001 \rightarrow 0010 \rightarrow 0011 \rightarrow 0100 \rightarrow 0101 \rightarrow 0110 \rightarrow 0111 \rightarrow 1000 \rightarrow 1001 \rightarrow 1010 \rightarrow 1011 \rightarrow 1100 \rightarrow 1101 \rightarrow 1110 \rightarrow 1111 \rightarrow 0000$  之計數器。首先假設將 4 個 *JK* 正反器之輸出標示為  $A_n$ 、 $B_n$ 、 $C_n$  與  $D_n$  之符號，並依據計數模數列出所求計數器之狀態表與利用 *JK* 正反器之激勵表，以列出所求計數器之轉態表。

目前狀態					下一狀態					正反器之輸入							
狀態	$D_n$	$C_n$	$B_n$	$A_n$	狀態	$D_{n+1}$	$C_{n+1}$	$B_{n+1}$	$A_{n+1}$	$J_D$	$K_D$	$J_C$	$K_C$	$J_B$	$K_B$	$J_A$	$K_A$
0	0	0	0	0	1	0	0	0	1	0	×	0	×	0	×	1	×
1	0	0	0	1	2	0	0	1	0	0	×	0	×	1	×	×	1
2	0	0	1	0	3	0	0	1	1	0	×	0	×	×	0	1	×
3	0	0	1	1	4	0	1	0	0	0	×	1	×	×	1	×	1
4	0	1	0	0	5	0	1	0	1	0	×	×	0	0	×	1	×
5	0	1	0	1	6	0	1	1	0	0	×	×	0	1	×	×	1
6	0	1	1	0	7	0	1	1	1	0	×	×	0	×	0	1	×
7	0	1	1	1	8	1	0	0	0	1	×	×	1	×	1	×	1
8	1	0	0	0	9	1	0	0	1	×	0	0	×	0	×	1	×
9	1	0	0	1	10	1	0	1	0	×	0	0	×	1	×	×	1
10	1	0	1	0	11	1	0	1	1	×	0	0	×	×	0	1	×
11	1	0	1	1	12	1	1	0	0	×	0	1	×	×	1	×	1
12	1	1	0	0	13	1	1	0	1	×	0	×	0	0	×	1	×
13	1	1	0	1	14	1	1	1	0	×	0	×	0	1	×	×	1
14	1	1	1	0	15	1	1	1	1	×	0	×	0	×	0	1	×
15	1	1	1	1	0	0	0	0	0	×	1	×	1	×	1	×	1

- ◆ 利用卡諾圖分別對上表之  $J_A$ 、 $K_A$ 、 $J_B$ 、 $K_B$ 、 $J_C$ 、 $K_C$ 、 $J_D$  與  $K_D$  等  $JK$  正反器之輸入端進行邏輯化簡，即可得到這些輸入之最簡布林代數式如下：

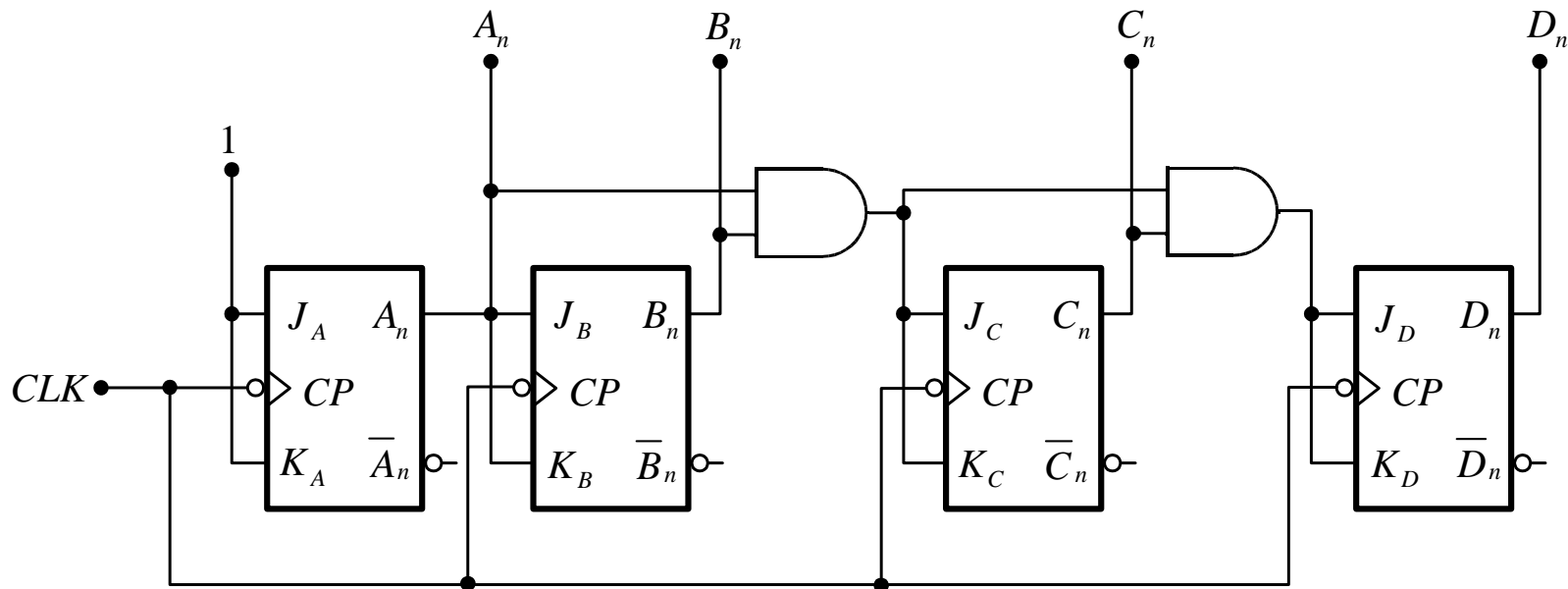
$$J_A = K_A = 1$$

$$J_B = K_B = A_n$$

$$J_C = K_C = B_n \cdot A_n$$

$$J_D = K_D = C_n \cdot B_n \cdot A_n$$

- ◆ 使用 4 個  $JK$  正反器與化簡所得之每一個正反器之輸入  $J_A$ 、 $K_A$ 、 $J_B$ 、 $K_B$ 、 $J_C$ 、 $K_C$ 、 $J_D$  與  $K_D$  的最簡布林代數，即可得到 16 進上數同步計數器之邏輯電路，如下圖所示。





## 計數模式 $N > 16$ 之 $2^n$ 之上數同步計數器

- ◆ 由以上之討論可知，利用 **JK 正反器** 所設計之計數模式為  $N = 2^n$  的**上數同步計數器**，控制每一個正反器之輸入端之**組合邏輯**有一定之規則可循。
- ◆ 若使用  $z$  個 **JK 正反器** 來實現計數模式  $N = 2^z$  之**上數同步計數器**，控制每一個正反器輸入端（ 假設  $A_n$  為最低有效位元， $Y_n$  為次最高有效位元， $Z_n$  為最高有效位元 ）之**組合邏輯**為

$$J_A = K_A = 1$$

$$J_B = K_B = A_n$$

$$J_C = K_C = B_n \cdot A_n$$

·

·

·

$$J_Z = K_Z = Y_n \cdot \dots \cdot C_n \cdot B_n \cdot A_n$$

# 八進下數同步計數器

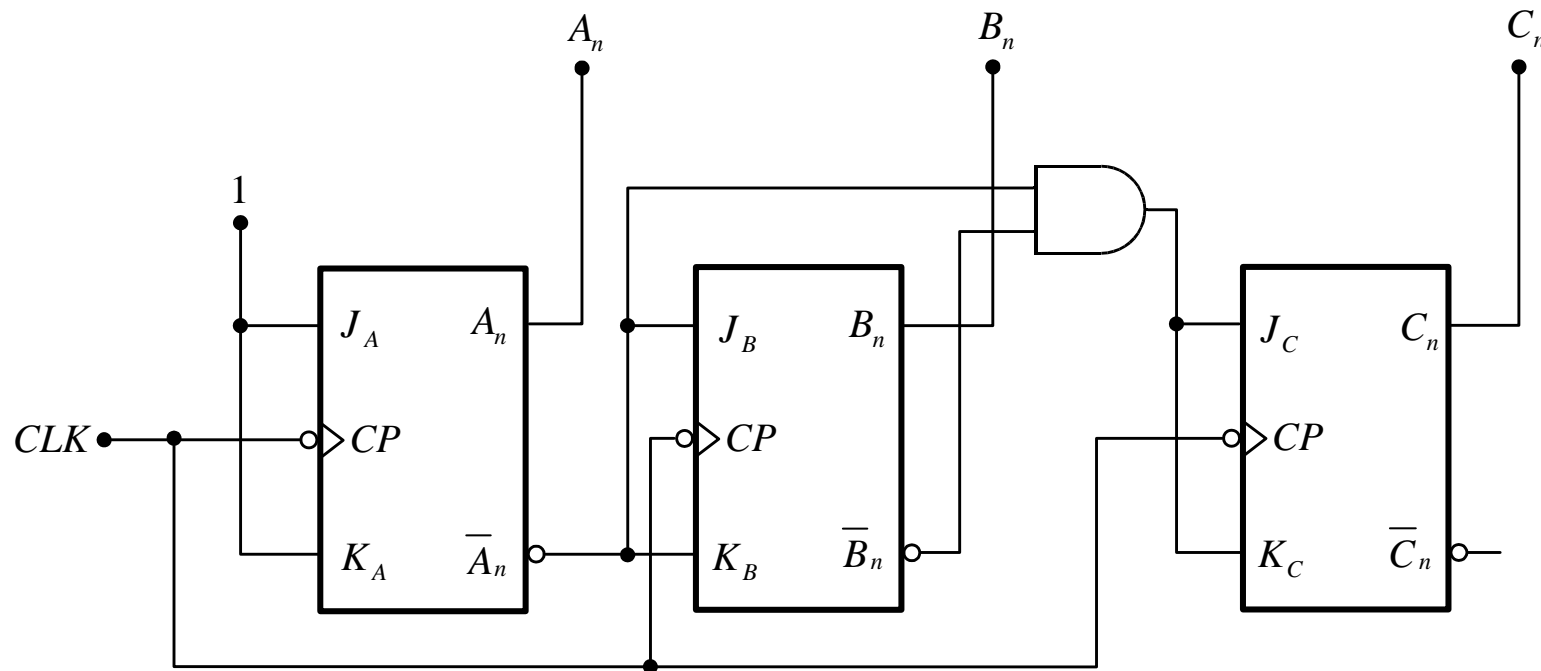
- ◆ 計數模式為  $2^n$  之下數同步計數器的設計方法與上數同步計數器相同，為避免重複起見，本節僅討論計數模式  $N = 2^3 = 8$  之下數同步計數器與計數模式  $N > 8$  之  $2^n$  下數同步計數器。
- ◆ 使用 3 個  $JK$  正反器亦可用來實現計數模式為 8 ( $n = \log_2 8 = 3$ ) 之下數同步計數器 ( 假設起始狀態為 000)，即計數模式為  $000 \rightarrow 111 \rightarrow 110 \rightarrow 101 \rightarrow 100 \rightarrow 011 \rightarrow 010 \rightarrow 001 \rightarrow 000$  之計數器。首先將 3 個  $JK$  正反器之輸出標示為  $A_n$ 、 $B_n$  與  $C_n$  之符號，並依據計數模數列出所求計數器之狀態表與利用  $JK$  正反器之激勵表，以列出所求計數器之轉態表，如下表所示。

目前狀態				下一狀態				正反器之輸入					
狀態	$C_n$	$B_n$	$A_n$	狀態	$C_{n+1}$	$B_{n+1}$	$A_{n+1}$	$J_C$	$K_C$	$J_B$	$K_B$	$J_A$	$K_A$
0	0	0	0	7	1	1	1	1	×	1	×	1	×
7	1	1	1	6	1	1	0	×	0	×	0	×	1
6	1	1	0	5	1	0	1	×	0	×	1	1	×
5	1	0	1	4	1	0	0	×	0	0	×	×	1
4	1	0	0	3	0	1	1	×	1	1	×	1	×
3	0	1	1	2	0	1	0	0	×	×	0	×	1
2	0	1	0	1	0	0	1	0	×	×	1	1	×
1	0	0	1	0	0	0	0	0	×	0	×	×	1

- ◆ 利用卡諾圖分別對上表之  $J_A$ 、 $K_A$ 、 $J_B$ 、 $K_B$ 、 $J_C$  與  $K_C$  等  $JK$  正反器之輸入端進行邏輯化簡，即可得到這些輸入之最簡布林代數式如下：

$$J_A = K_A = 1 \qquad J_B = K_B = \bar{A}_n \qquad J_C = K_C = \bar{B}_n \cdot \bar{A}_n$$

- ◆ 使用 3 個  $JK$  正反器與化簡所得之每一個正反器之輸入  $J_A$ 、 $K_A$ 、 $J_B$ 、 $K_B$ 、 $J_C$  與  $K_C$  的最簡布林代數，即可得到 8 進下數同步計數器之邏輯電路，如下圖所示。



## 計數模式 $N > 8$ 之 $2^n$ 之下數同步計數器

- ◆ 由以上之討論可知，利用  $JK$  正反器所設計之計數模式為  $N = 2^n$  的下數同步計數器，控制每一個正反器之輸入端之組合邏輯亦有一定之規則可循。
- ◆ 若使用  $z$  個  $JK$  正反器來實現計數模式  $N = 2^z$  之下數同步計數器，控制每一個正反器輸入端（假設  $A_n$  為最低有效位元， $Y_n$  為次最高有效位元， $Z_n$  為最高有效位元）之組合邏輯為

$$J_A = K_A = 1$$

$$J_B = K_B = \overline{A_n}$$

$$J_C = K_C = \overline{B_n} \cdot \overline{A_n}$$

.

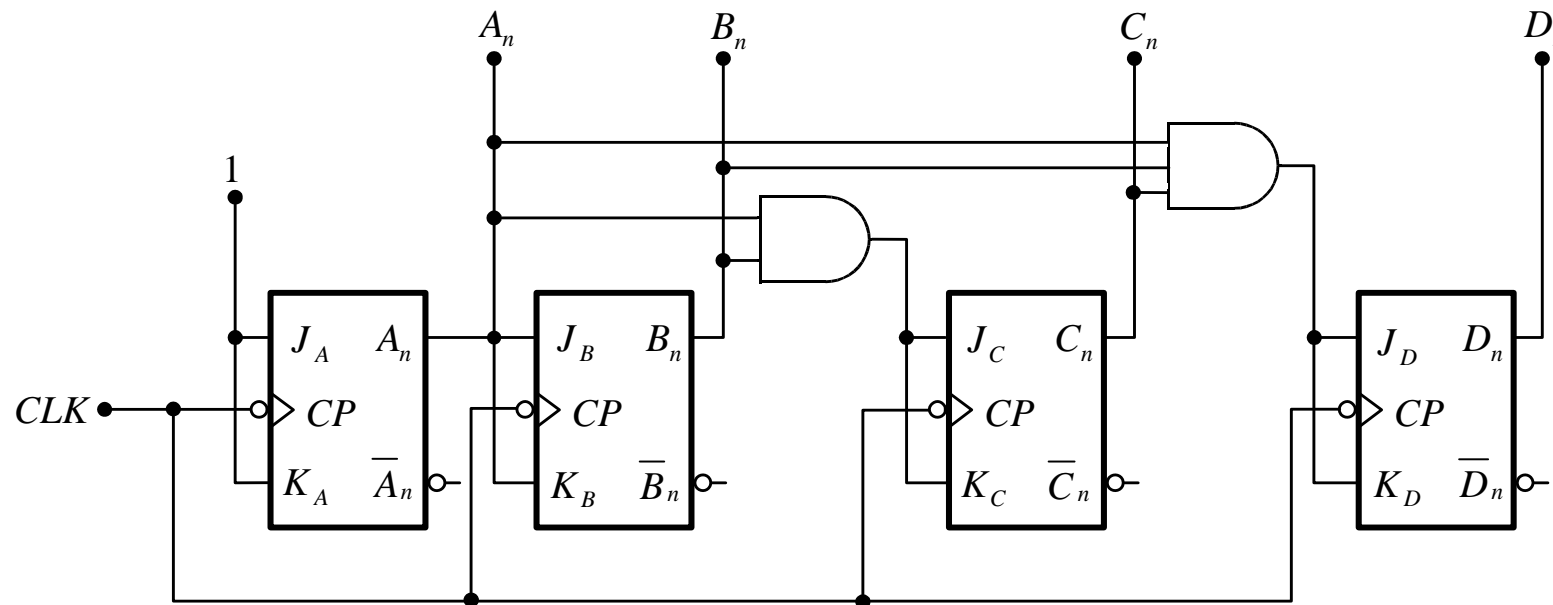
.

.

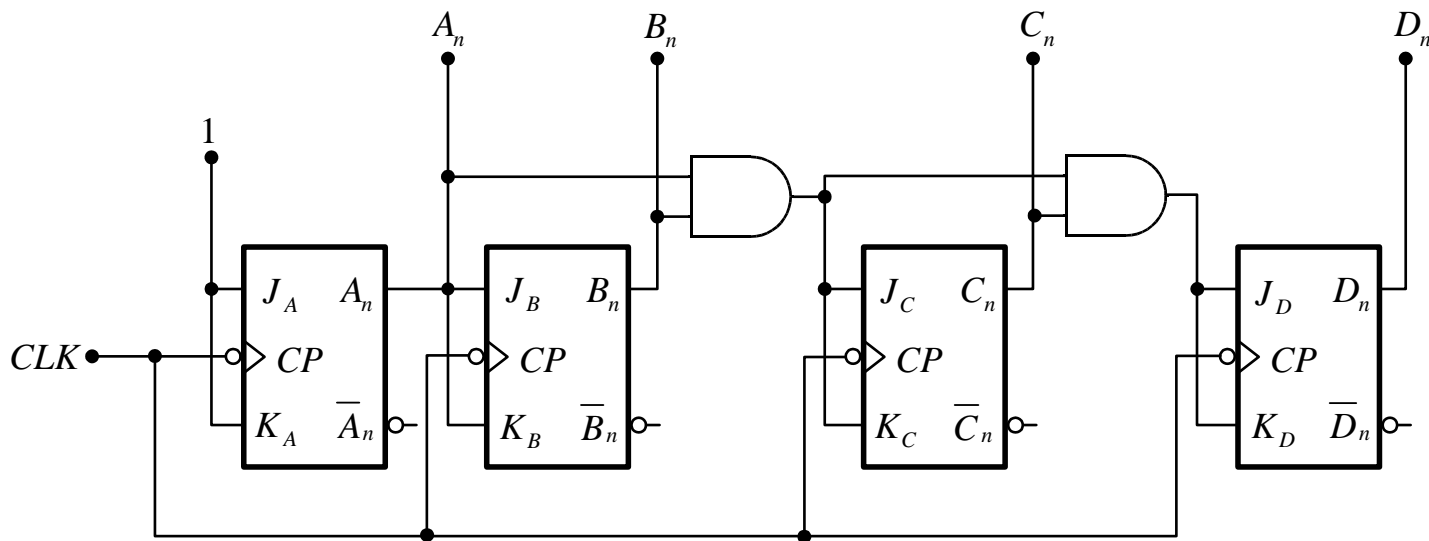
$$J_Z = K_Z = \overline{Y_n} \cdot \dots \cdot \overline{C_n} \cdot \overline{B_n} \cdot \overline{A_n}$$

## 同步計數器之最高操作頻率

- ◆ 同步計數器之外加組合邏輯的連接方式，可分為**漣波進位模式** (Ripple Carry Mode) 與**並行進位模式** (Parallel Carry Mode) 兩種。
- ◆ **並行進位模式**之時間延遲較**漣波進位模式**之同步計數器**短**，即並行進位模式之時間延遲，僅為**正反器加上一個邏輯閘**，如下圖所示。



## 並行進位模式之 16 進上數同步計數器



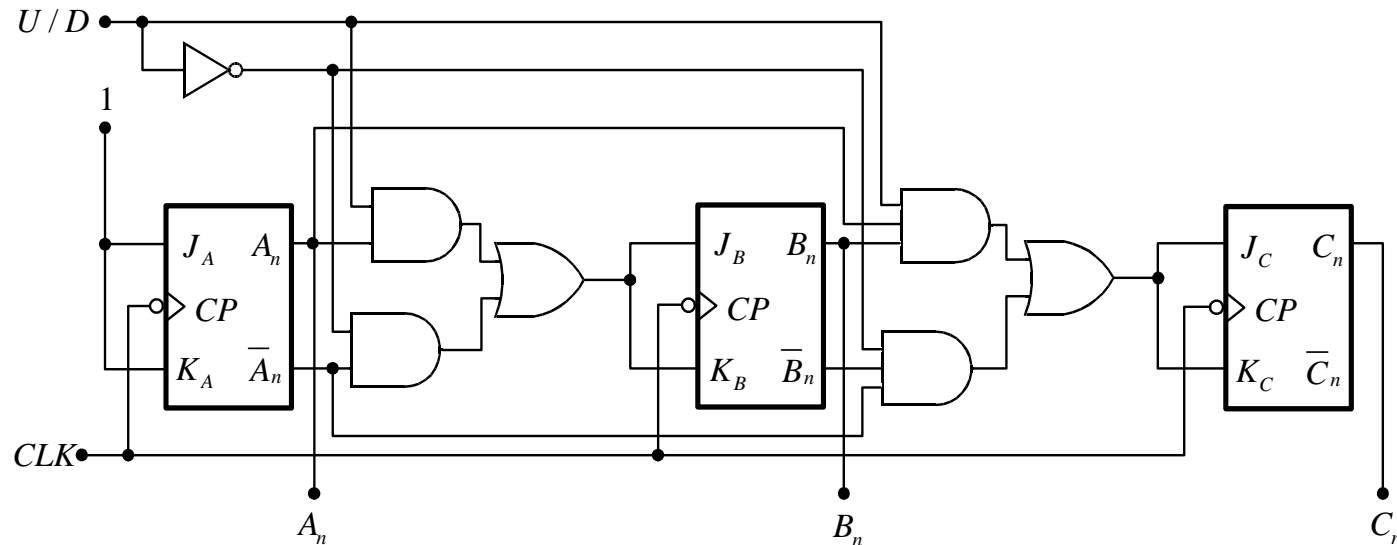
連波進位模式之 16 進上數同步計數器

- ◆ 連波進位模式之同步計數器，除了正反器之延遲外(如前面所討論之 16 進上數同步計數器)，另外再增加串接級數之邏輯閘的時間延遲。
- ◆ 根據以上之討論，可得到連波進位模式與並行進位模式等兩種同步計數器之操作頻率上限

$f_{\max}$  分別為  $f_{\max} = \frac{1}{t_d + (n-2) \cdot t_G}$  與  $f_{\max} = \frac{1}{t_d + t_G}$ ，其中  $t_d$  為一個正反器之延遲時間， $t_G$  為一個邏輯閘之延遲時間， $n$  為正反器之數量，且兩者之區別僅考慮正反器數量大於 3 之情況。

## 計數模式為 $2^n$ 之上 / 下數同步計數器

- ◆ 若加上 1 個控制  $U/D$  ( 上 / 下數之控制 ) 輸入端，以設計一個計數模式為 $2^n$ 之上 / 下數同步計數器；本節僅討論結合 8 進 ( $n = 3$ ) 上數與下數同步計數器，以構成一個可同時實現 8 進上 / 下數之同步計數器，如下圖所示。



1. 當  $U/D = 1$  時，則 3 個正反器之輸入分別為  $J_A = K_A = 1$ 、 $J_B = K_B = A_n$  與  $J_C = K_C = B_n \cdot A_n$ ，故上圖可執行 8 進上數運算。
2. 當  $U/D = 0$  時，則 3 個正反器之輸入分別為  $J_A = K_A = 1$ 、 $J_B = K_B = \bar{A}_n$  與  $J_C = K_C = \bar{B}_n \cdot \bar{A}_n$ ，故上圖可執行 8 進下數運算。

# 十進位上數同步計數器

- ◆ 因人類習慣使用十進位數，故十進位計數器在實際上有其實用之價值，故首先討論使用  $JK$  正反器來設計十進位同步計數器（亦可稱為  $BCD$  同步計數器）之方法。
- ◆ 接著分別討論使用  $SR$ 、 $JK$ 、 $D$  型與  $T$  型等 4 種正反器，以設計不規則計數順序（即不依二進位之計數順序）之同步計數器的方法。
- ◆ 十進位計數器之計數順序為  $0000 \rightarrow 0001 \rightarrow 0010 \rightarrow 0011 \rightarrow 0100 \rightarrow 0101 \rightarrow 0110 \rightarrow 0111 \rightarrow 1000 \rightarrow 1001 \rightarrow 0000$ （即為從 0 至 9 後又回到 0）之循環，故此種計數器共有 10 個狀態，故必須使用 4 個（ $n = \lceil \log_2 10 \rceil = 4$ ）正反器才可實現。
- ◆ 假設將 4 個  $JK$  正反器之輸出標示為  $A_n$ 、 $B_n$ 、 $C_n$  與  $D_n$  之符號，並依據計數模數列出所求計數器之狀態表與利用  $JK$  正反器之激勵表 (Excitation Table)，以列出所求計數器之轉態表 (Transition Table)，如下表所示。



目 前 狀 態					下 一 狀 態					正 反 器 之 輸 入							
狀態	$D_n$	$C_n$	$B_n$	$A_n$	狀態	$D_{n+1}$	$C_{n+1}$	$B_{n+1}$	$A_{n+1}$	$J_D$	$K_D$	$J_C$	$K_C$	$J_B$	$K_B$	$J_A$	$K_A$
0	0	0	0	0	1	0	0	0	1	0	×	0	×	0	×	1	×
1	0	0	0	1	2	0	0	1	0	0	×	0	×	1	×	×	1
2	0	0	1	0	3	0	0	1	1	0	×	0	×	×	0	1	×
3	0	0	1	1	4	0	1	0	0	0	×	1	×	×	1	×	1
4	0	1	0	0	5	0	1	0	1	0	×	×	0	0	×	1	×
5	0	1	0	1	6	0	1	1	0	0	×	×	0	1	×	×	1
6	0	1	1	0	7	0	1	1	1	0	×	×	0	×	0	1	×
7	0	1	1	1	8	1	0	0	0	1	×	×	1	×	1	×	1
8	1	0	0	0	9	1	0	0	1	×	0	0	×	0	×	1	×
9	1	0	0	1	0	0	0	0	0	×	1	0	×	0	×	×	1

- ◆ 利用卡諾圖分別對下表之 $J_A$ 、 $K_A$ 、 $J_B$ 、 $K_B$ 、 $J_C$ 、 $K_C$ 、 $J_D$ 與 $K_D$ 等 JK 正反器之輸入端進行邏輯化簡，即可得到最簡布林代數式如下：

$$J_A = K_A = 1$$

$$J_B = \overline{D}_n \cdot A_n$$

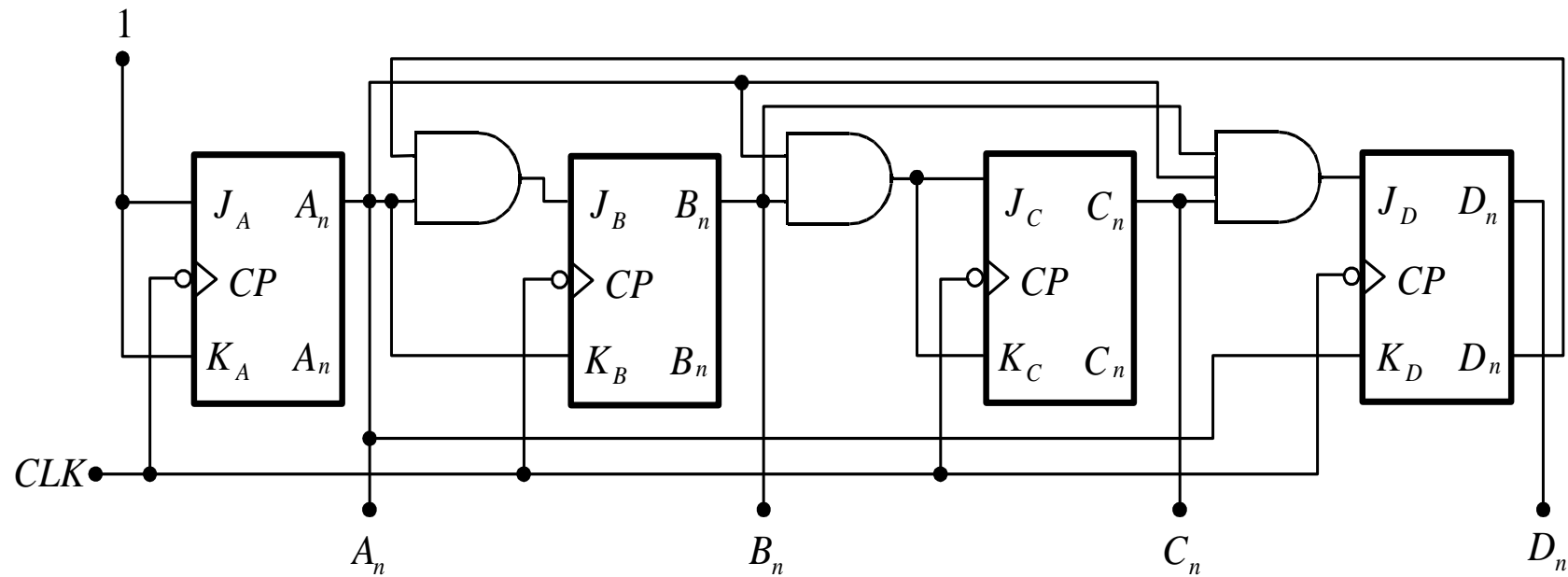
$$K_B = A_n$$

$$J_C = K_C = B_n \cdot A_n$$

$$J_D = C_n \cdot B_n \cdot A_n$$

$$K_D = A_n$$

◆ 使用 4 個  $JK$  正反器，即可繪製十進上數同步計數器之邏輯電路，如下圖所示。



# 不規則計數順序（亂數）之同步計數器

- ◆ 若依照同步計數器之設計步驟，亦可設計出不規則計數順序之同步計數器，亦可稱為亂數同步計數器 (Random Synchronous Counter)。
- ◆ 任一種型式之正反器，皆可用來實現同步計數器，而僅是控制正反器轉態之組合邏輯複雜度不同之別，故本節將討論利用不同型式之正反器，以實現不規則計數順序之同步計數器，以提供讀者參考比較。

## 例題 9-1

試分別利用  $JK$  與  $D$  型正反器，設計一個計數序列為  $100 \rightarrow 001 \rightarrow 101 \rightarrow 111 \rightarrow 000 \rightarrow 100$  之同步計數器。

 解

欲實現所求之同步計數器必需使用 3 個正反器，首先分別將 3 個正反器之輸出標示為  $A_n$ 、 $B_n$  與  $C_n$  之符號，並依據所求同步計數器之計數順序，分別利用  $JK$  與  $D$  型正反器之激勵表 (Excitation Table)，以列出所求同步計數器之轉態表 (Transition Table)，如下表所示。

目前狀態			下一狀態			正 反 器 之 輸 入								
$C_n$	$B_n$	$A_n$	$C_{n+1}$	$B_{n+1}$	$A_{n+1}$	$J_C$	$K_C$	$J_B$	$K_B$	$J_A$	$K_A$	$D_C$	$D_B$	$D_A$
1	0	0	0	0	1	×	1	0	×	1	×	0	0	1
0	0	1	1	0	1	1	×	0	×	×	0	1	0	1
1	0	1	1	1	1	×	0	1	×	×	0	1	1	1
1	1	1	0	0	0	×	1	×	1	×	1	0	0	0
0	0	0	1	0	0	1	×	0	×	0	×	1	0	0

接著利用卡諾圖分別對上表之  $J_A$ 、 $K_A$ 、 $J_B$ 、 $K_B$ 、 $J_C$ 、 $K_C$  與  $D_A$ 、 $D_B$ 、 $D_C$  的等 JK 與 D 型正反器之輸入端進行邏輯化簡，即可得到最簡布林代數式如下：

$$J_A = C_n$$

$$K_A = B_n$$

$$J_B = C_n \cdot A_n$$

$$K_B = 1$$

$$J_C = 1$$

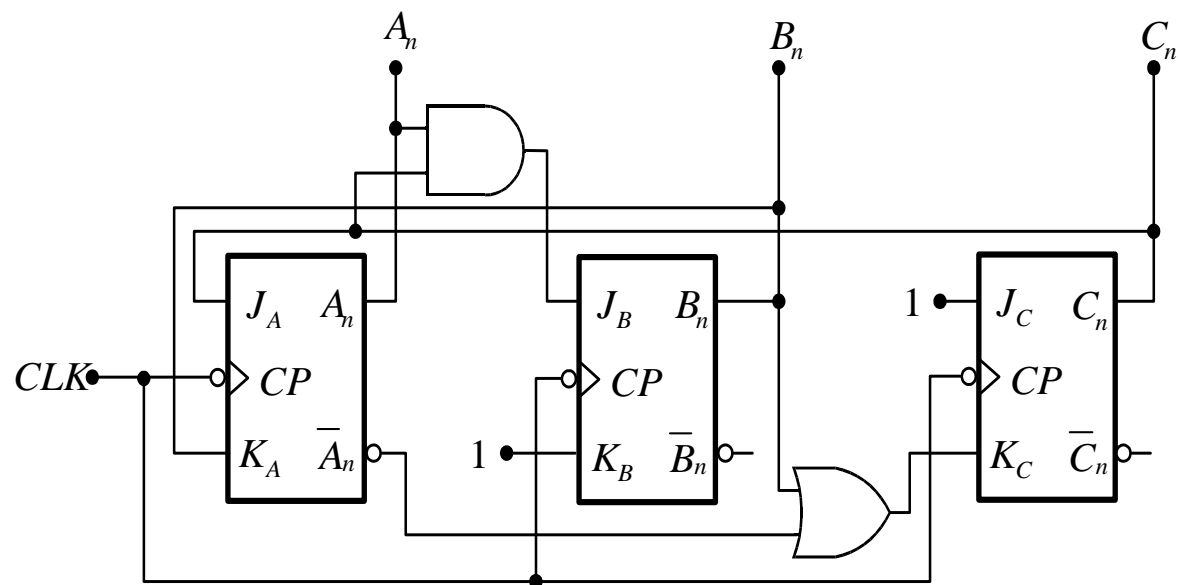
$$K_C = \overline{A_n} + B_n$$

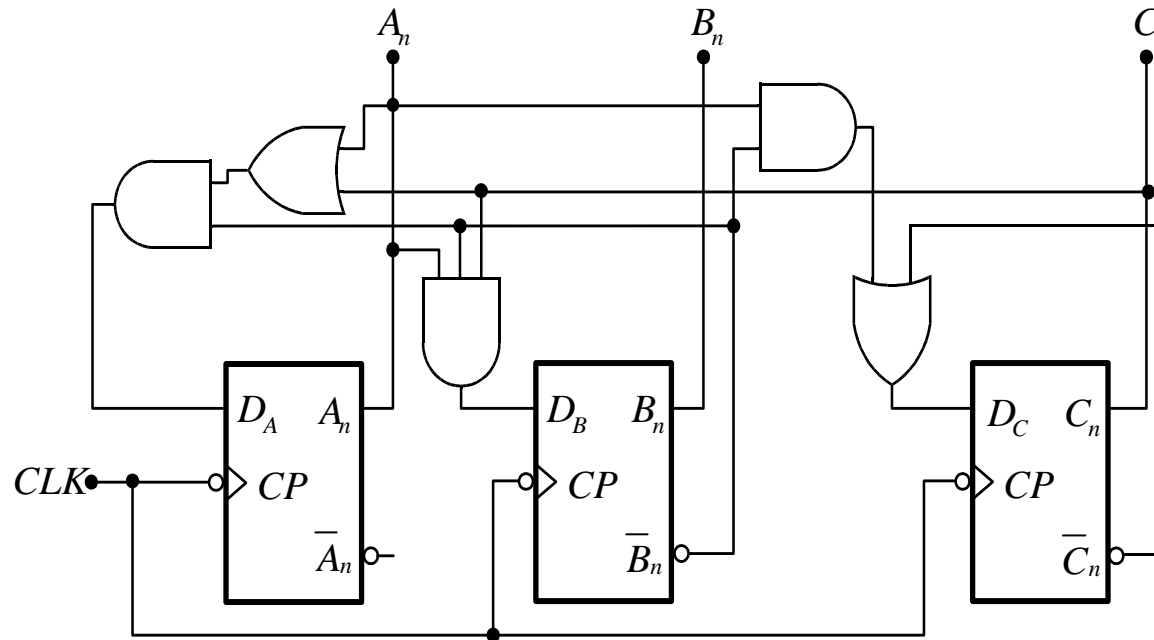
$$D_A = C_n \cdot \overline{B_n} + \overline{B_n} \cdot A_n = \overline{B_n} \cdot (A_n + C_n)$$

$$D_B = C_n \cdot \overline{B_n} \cdot A_n$$

$$D_C = \overline{C_n} + \overline{B_n} \cdot A_n$$

最後分別使用 3 個  $JK$  正反器與  $D$  型正反器，繪出所求不規則計數順序之同步計數器的邏輯電路，分別如下 2 圖所示。





註：觀察上兩圖可知，當加入週期性計數脈波  $CLK$  於正反器之  $CP$  端後，藉由控制正反器轉態之組合邏輯，皆可滿足所求計數順序之同步計數器。而控制  $JK$  正反器轉態所需之組合邏輯複雜度，明顯的較使用  $D$  型正反器相同計數順序之同步計數器簡單了許多。

## 例題 9-2

試分別利用  $SR$  與  $T$  型正反器，設計一個計數序列為  $100 \rightarrow 001 \rightarrow 101 \rightarrow 111 \rightarrow 000 \rightarrow 100$  之同步計數器。

解

欲實現所求之同步計數器必需使用 3 個正反器，首先分別將 3 個  $SR$  與  $T$  型正反器之輸出標示為  $A_n$ 、 $B_n$  與  $C_n$  之符號，並依據所求同步計數器之計數順序，分別利用  $SR$  與  $T$  型正反器之激勵表，以列出所求同步計數器之轉態表，如下表所示。

目前狀態			下一狀態			正反器之輸入								
$C_n$	$B_n$	$A_n$	$C_{n+1}$	$B_{n+1}$	$A_{n+1}$	$S_C$	$R_C$	$S_B$	$R_B$	$S_A$	$R_A$	$T_C$	$T_B$	$T_A$
1	0	0	0	0	1	0	1	0	×	1	0	1	0	1
0	0	1	1	0	1	1	0	0	×	×	0	1	0	0
1	0	1	1	1	1	×	0	1	0	×	0	0	1	0
1	1	1	0	0	0	0	1	0	1	0	1	1	1	1
0	0	0	1	0	0	1	0	0	×	0	×	1	0	0

利用卡諾圖分別對上表之  $S_A$ 、 $R_A$ 、 $S_B$ 、 $R_B$ 、 $S_C$ 、 $R_C$  與  $T_A$ 、 $T_B$ 、 $T_C$  等 SR 與 T 型正反器之輸入端進行邏輯化簡，即可得到最簡布林代數式如下：

$$S_A = C_n \cdot \overline{A_n}$$

$$R_A = B_n$$

$$S_B = C_n \cdot \overline{B_n} \cdot A_n$$

$$R_B = B_n$$

$$S_C = \overline{C_n}$$

$$R_C = B_n + C_n \cdot \overline{A_n}$$

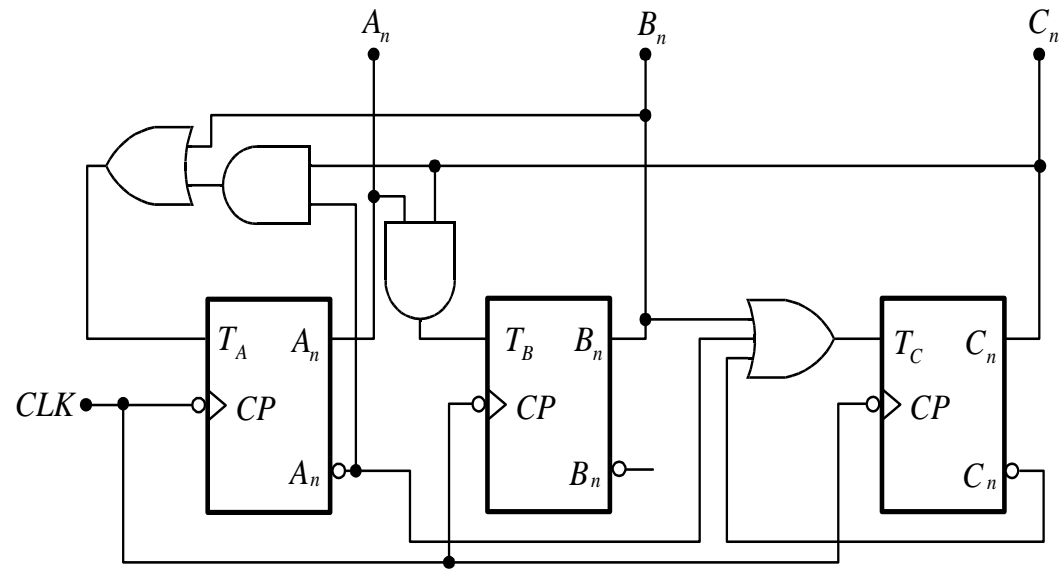
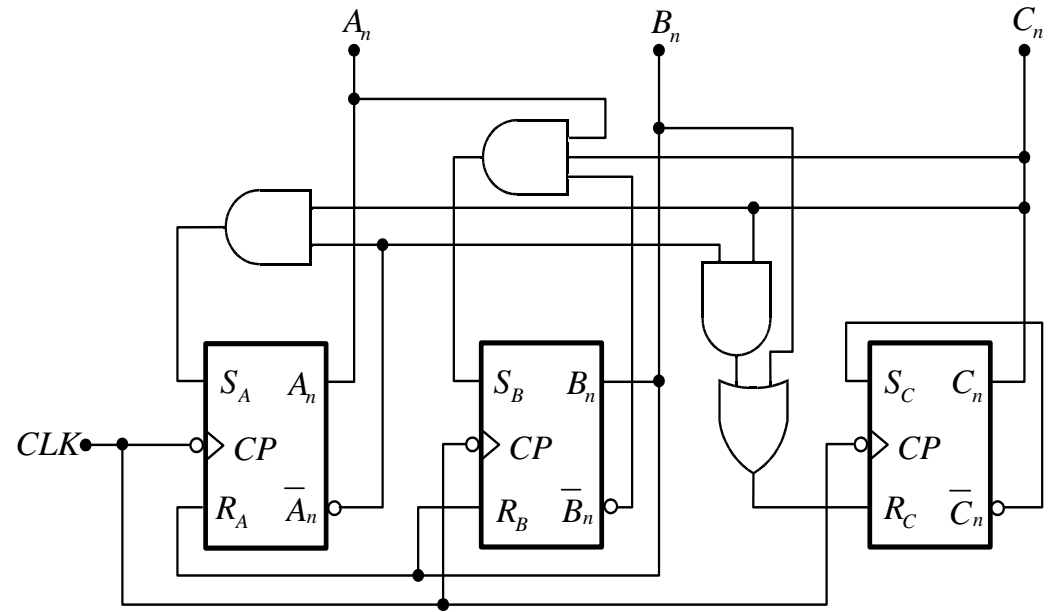
$$T_A = B_n + C_n \cdot \overline{A_n}$$

$$T_B = C_n A_n$$

$$T_C = \overline{C_n} + B_n + \overline{A_n}$$

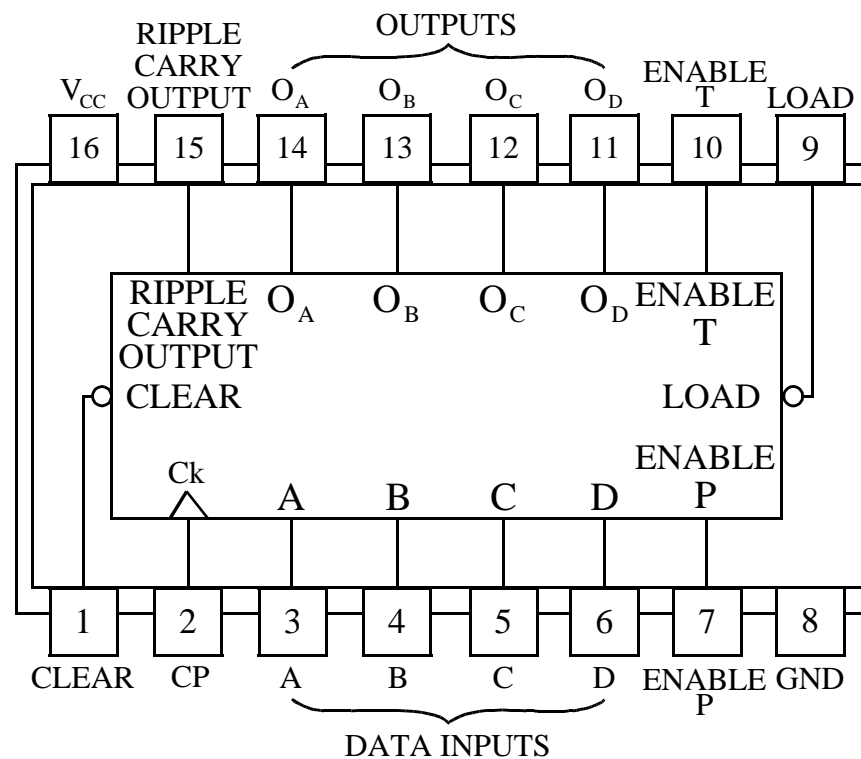
分別使用 3 個 SR 正反器與 T 型正反器，將化簡所得之每一個正反器之輸入  $S_A$ 、 $R_A$ 、 $S_B$ 、 $R_B$ 、 $S_C$ 、 $R_C$  與  $T_A$ 、 $T_B$ 、 $T_C$  等的最簡布林代數，即可繪出求不規則計數順序之同步計數器的邏輯電路，分別如下 2 圖所示。





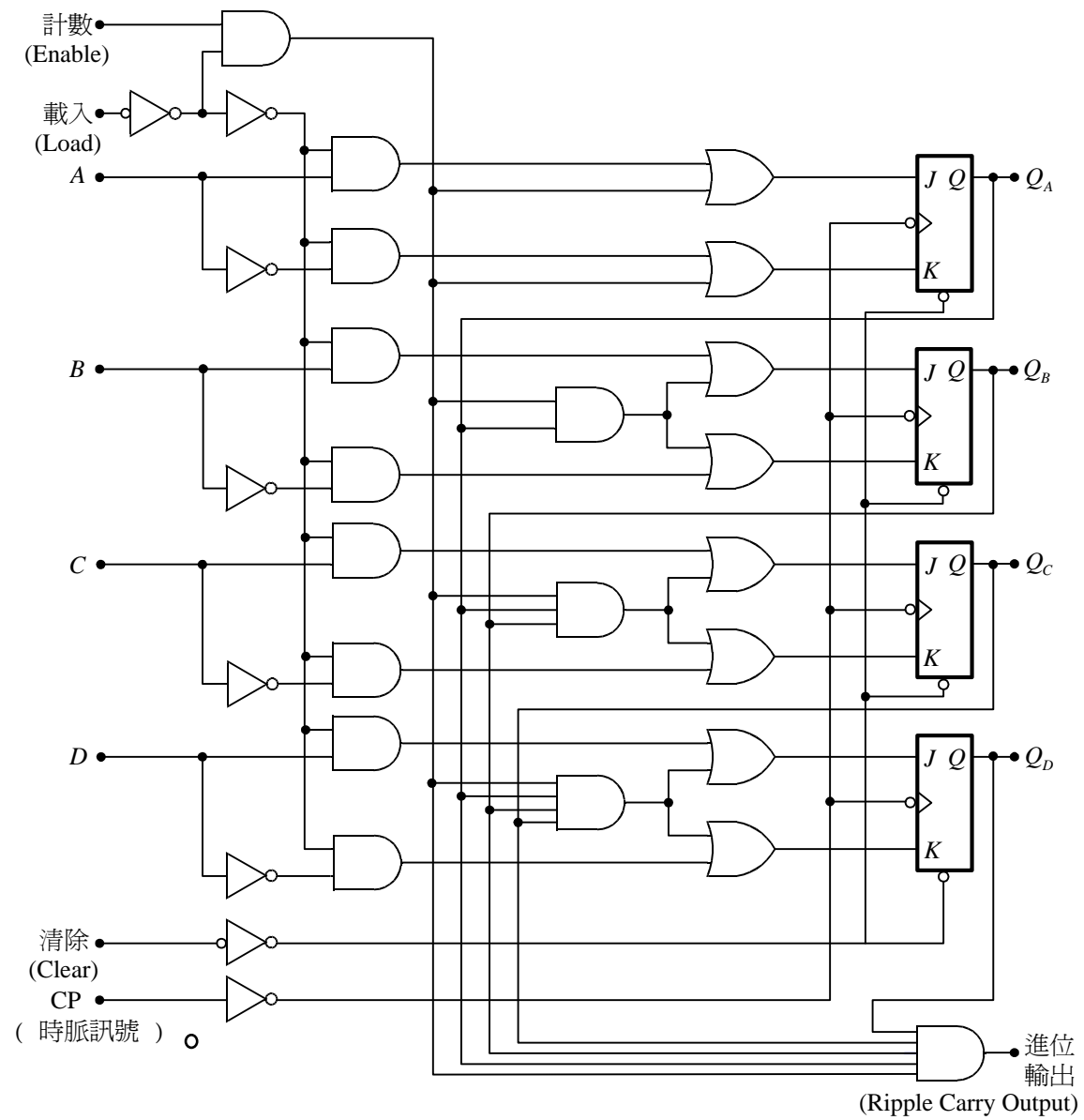
# 可同步預置與清除計數狀態之積體化同步計數器

- ◆ 本節將以 IC 編號為 **74LS161** 為例，以說明具並聯載入的 16 進上數同步計數器，以任意設定開始與結束計數狀態之方法。首先列出此 IC 之接腳圖與邏輯電路圖，分別如下 2 圖所示。



- ◆ 此計數器可藉由**計數**、**載入**和**清除**等控制訊號，以提供使用者任意設定計數模式小於 16 的**開始與結束**之計數狀態。而此 IC 之**功能表**，如下表所示。

清 除	<i>CP</i>	載 入	計 數	功 能
0	×	×	×	清除全部正反器為 0
1	0	×	×	輸出維持不變
1	↑	0	×	轉移輸入至輸出
1	↑	1	1	二進位計數狀態

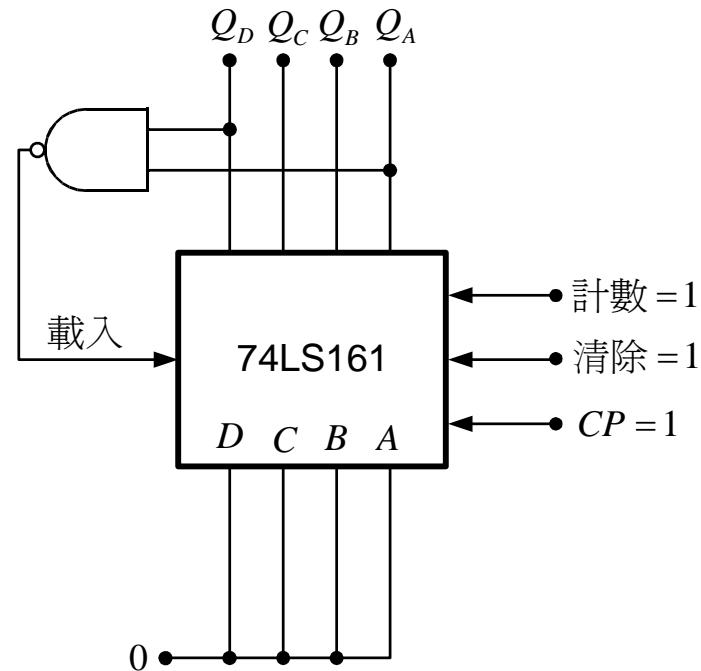
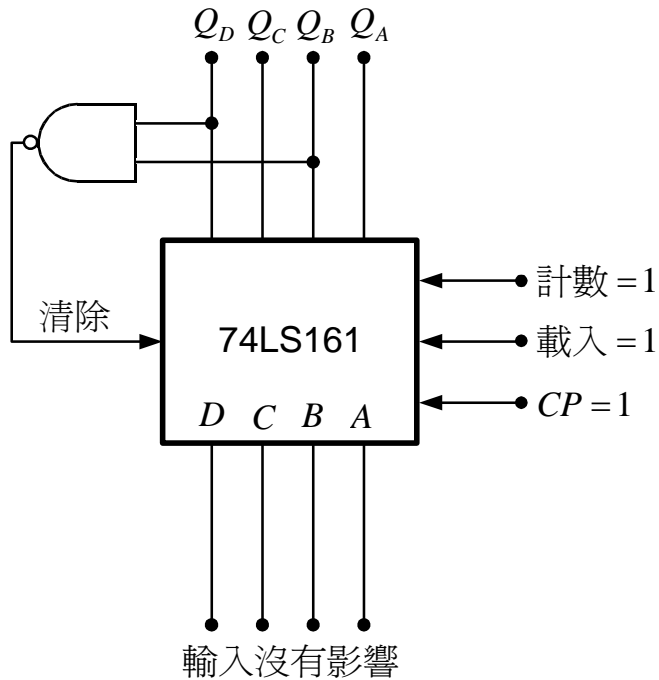


### 例題 9-3

試利用 74LS161 來實現計數模式為 0 (0000) 至 9 (1001) 之十進上數同步計數器。

解

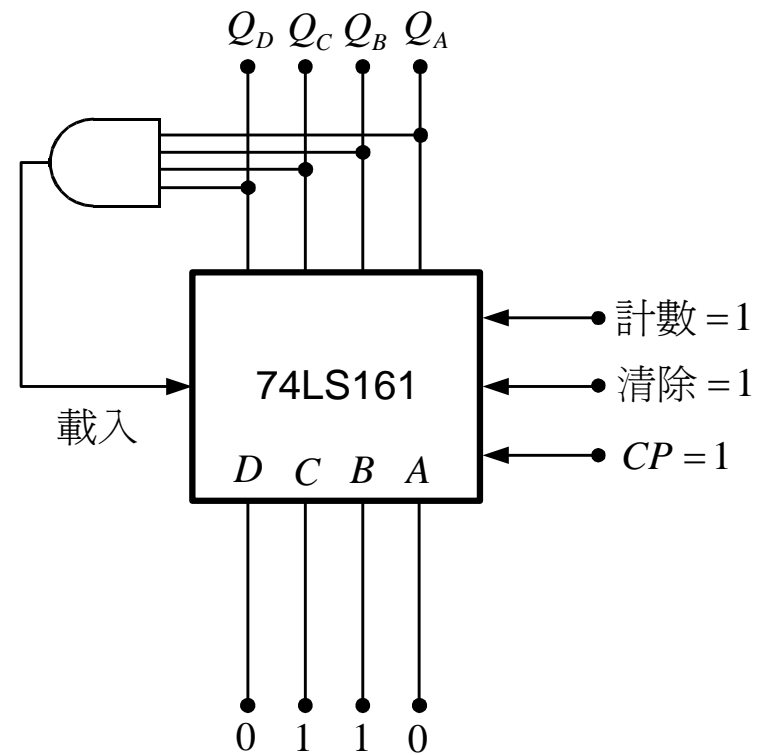
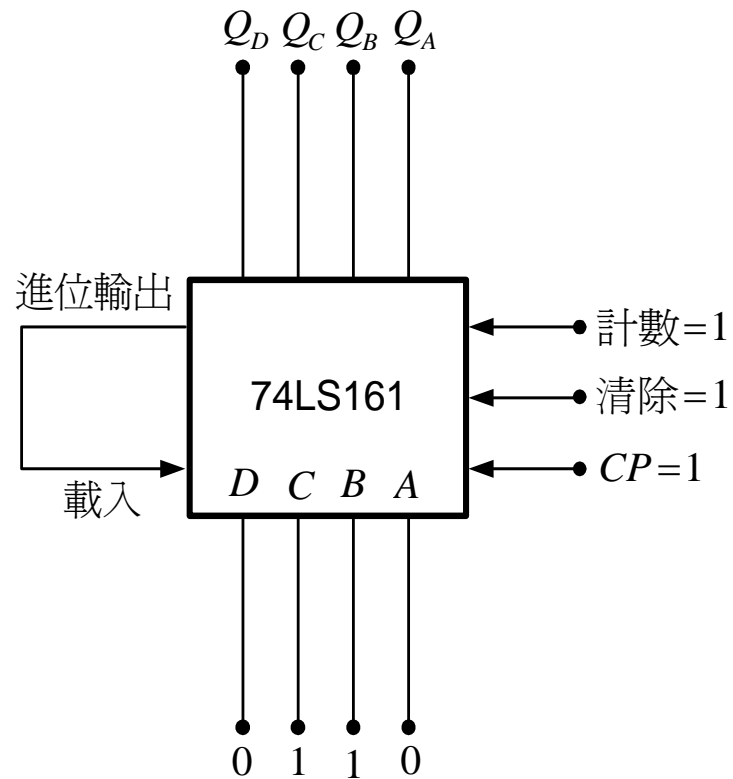
本例題將提供兩種實現計數模式為 0 (0000) 至 9 (1001) 之十進上數同步計數器，分別如下 2 圖所示。



**解**

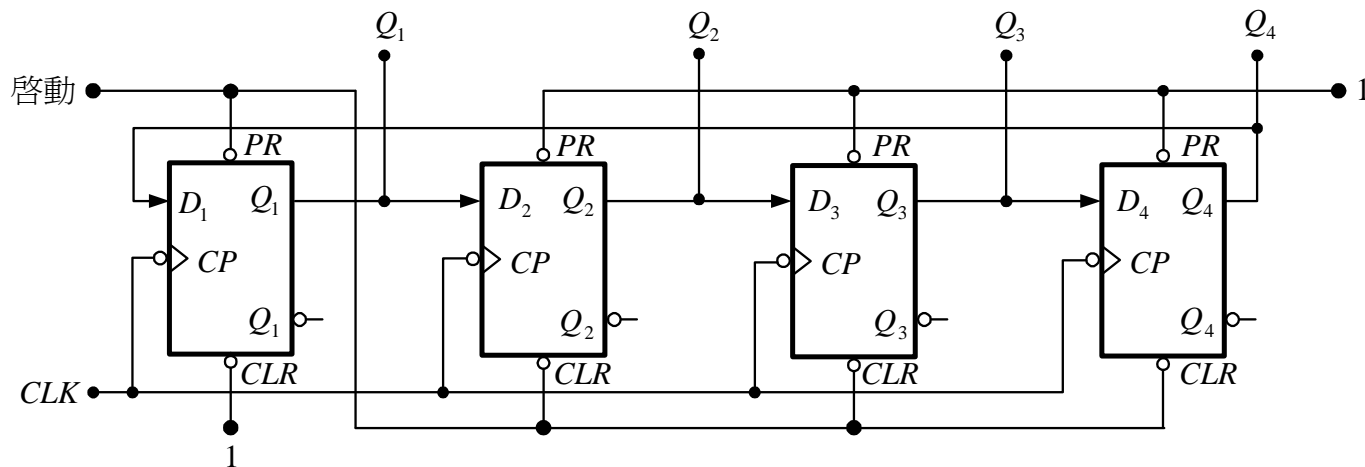
試利用 74LS161 來實現計數模式為 6 (0110) 至 15 (1111) 之上數同步計數器。

本例題亦將提供兩種實現計數模式為 6(0110)15 (1111)之同步計數器，分別如下 2 圖所示。



## 特殊型式之同步計數器(環型計數器)

- ◆ 前面所討論之二進位計數器(包括同步與非同步計數器)的最大計數模式  $N$  與正反器數目成指數 (Exponential) 函數 ( $N = 2^n$ ，其中  $n$  為計數器所使用之正反器數量) 之正比關係。
- ◆ 本節將討論一種移位暫存計數器 (Shift-Register Counter)，這種計數器之最大計數模式與正反器之數量成線性 (Linear) 正比關係，因此將其歸類為特殊型式計數器。
- ◆ 環型計數器 (Ring Counter) 可用來實現任意模數之計數器，且欲實現  $N$  個計數模數之環型計數器，必須使用  $N$  個正反器，因此設計相同模數之計數器而言，環型計數器所需之正反器數目，比前面所討論之二進位計數器多。
- ◆ 環型計數器使用  $D$  型正反器來設計較為簡單，而其電路結構是將正反器串接起來，使二進位資料可由左向右移位，且將最後一級正反器之非反相輸出端  $Q$ ，回饋至第一級正反器之輸入端，而一個計數模數為 4 之環型計數器，如下圖所示。



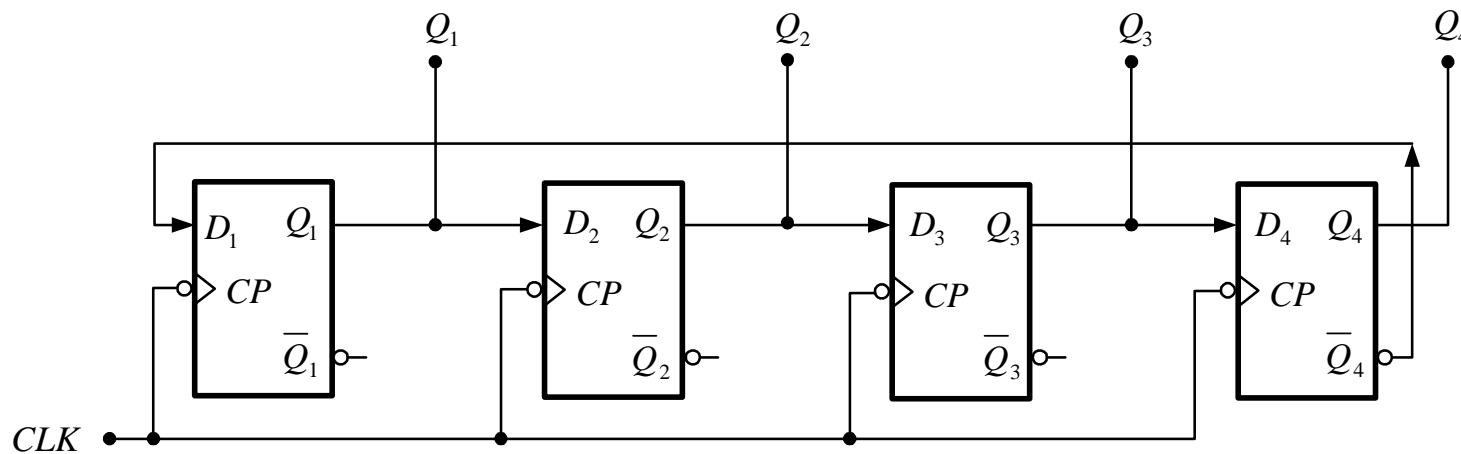
當對上圖加上計時脈波後，則**邏輯 1** 會隨著計數脈波之變動，在移位計數器內循環，如下表所示，因此稱為**環型計數器**(使用環型計數器須先設定其中一個正反器之輸出狀態為邏輯 1)。

計數脈波 (CP)	$Q_1$	$Q_2$	$Q_3$	$Q_4$
0 ( 起始狀態 )	1	0	0	0
1	0	1	0	0
2	0	0	1	0
3	0	0	0	1
4	1	0	0	0
5	0	1	0	0
6	0	0	1	0
⋮	⋮			



# 強生計數器

- ◆ **強生計數器** (Johnson Counter) 亦可用來實現**任意模數**之計數器，而欲實現  $N$  個計數模數之強生計數器，僅須使用  $\frac{N}{2}$  個正反器即可，因此設計**相同模數**之計數器而言，強生計數器所需之**正反器數目**為環型計數器之**一半**。
- ◆ 強生計數器亦使用 **D 型正反器** 來設計較為簡單，而其電路結構亦是將正反器**串接**起來，使二進位資料可**由左向右**移位，再將**最後一級**正反器之**反相輸出端**  $\bar{Q}$ ，回饋至**第一級**正反器之輸入端，而一個計數模數 8 之強生計數器，如下圖所示。



- ◆ 當對上圖加上計時脈波後，則邏輯 1 與邏輯 0 會隨著計數脈波之變動，在移位計數器內移進與移出，如下表所示。觀察下表可知，強生計數器之計數模數永遠是環型計數器的兩倍。

計數脈波 (CP)	$Q_1$	$Q_2$	$Q_3$	$Q_4$
0 ( 起始狀態 )	0	0	0	0
1	1	0	0	0
2	1	1	0	0
3	1	1	1	0
4	1	1	1	1
5	0	1	1	1
6	0	0	1	1
7	0	0	0	1
8	0	0	0	0
9	1	0	0	0
10	1	1	0	0
11	1	1	1	0
⋮	⋮			

註：強生計數器不須設定其中一個正反器之輸出狀態為邏輯 1，即可設定為任一種起始狀態。

## 環型計數器與強生計數器之比較

- ◆ 使用  $N$  個  $D$  型正反器之環型計數器，僅可實現  $n$  個計數模式之同步計數器，此為環型計數器之缺點；但環型計數器之輸出，永遠僅有一個邏輯 1，故不須經由解碼電路，即各個不同之計數狀態，可由相對正反器輸出得到，此為環型計數器之優點。
- ◆ 使用  $N$  個  $D$  型正反器之強生計數器，卻可實現  $2n$  個計數模式之同步計數器，因此在相同數目正反器的條件下，可提供較多計數模式，此為強生計數器之優點；然而需加上解碼電路，才可分辨出計數器之輸出模式，此為強生計數器之缺點。

# 問題討論

1、試比較**非同步計數器**(連波計數器)與**同步計數器**之優缺點。

解：

- (1) 連波計數器（非同步計數器）之**構造相當簡單**（僅使用正反器，而幾乎不需任何的）組合邏輯電路，為此種計數器之最大優點，但**操作頻率受到相當大之限制**，為此種計數器之最大缺點。
- (2) 同步計數器之時間延遲相對非同步計數器短了相當多，可**操作於比較高頻之計數脈波**，但需額外增加一些組合邏輯來控制正反器之輸出狀態，故需**較複雜構造**為同步計數器之缺點。

## 2、試討論漣波計數器之最高操作頻率的限制。

解：

漣波計數器之所有正反器輸入  $J$  與  $K$  均接邏輯 1，且上一級之輸出  $Q_n$  連接至下一級之脈衝輸入端，即每一級正反器的觸發皆會受到前一級之影響，依此類推，即可構成漣波計數器。因此計數模式愈多，則須串接愈多級之正反器，導致時間延遲會累積愈長。若每個正反器之時間延遲皆為  $t_d$ ，則串接  $n$  個正反器（即最大計數模式  $N = 2^n$  之計數器）之累積時間延遲為  $t_T = n \times t_d$ 。對週期性之脈波訊號而言，當時間延遲  $t_T$  大於計數器之輸入脈波訊號的週期  $\left(T = \frac{1}{f}\right)$  時，便可能會造成計數器之誤動作，故漣波計數器無法滿足高速操作之要求，而漣波計數器之最

高操作頻率  $f_{\max} = \frac{1}{n \times t_d} = \frac{1}{t_T}$ 。

3、若以 20MHz 之時序脈波加至一個 10 位元之計數器，試討論第 2 級、第 5 級與第 10 級正反器之輸出頻率。

解：

1、第2級正反器之輸出頻率為  $\frac{20M}{2^2} = 5MHz$ 。

2、第5級正反器之輸出頻率為  $\frac{20M}{2^5} = 625kHz$ 。

3、第9級正反器之輸出頻率為  $\frac{20M}{2^{10}} = 20kHz$ 。

#### 4、試討論同步計數器之最高操作頻率的限制。

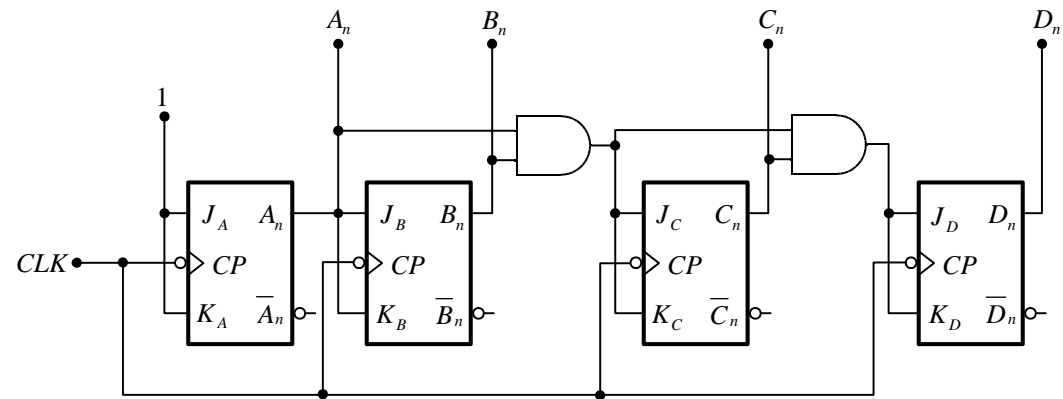
解：

同步計數器之外加組合邏輯的连接方式，可分為**漣波進位模式** (Ripple Carry Mode) 與**並行進位模式** (Parallel Carry Mode) 兩種。**漣波進位模式**與**並行進位模式**等兩種同步計數器之**操作頻率上限**分別為

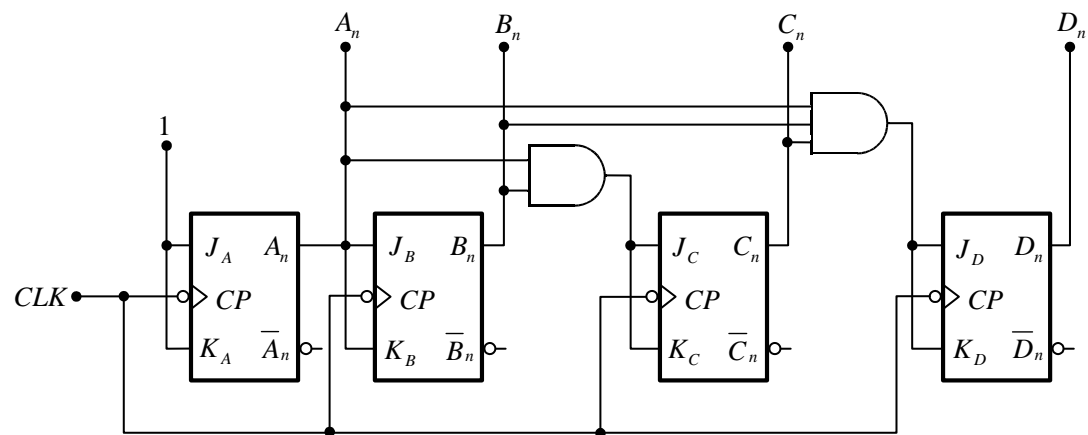
$$f_{\max} = \frac{1}{t_d + (n-2) \cdot t_G}$$

$$f_{\max} = \frac{1}{t_d + t_G}$$

其中 $t_d$ 為一個**正反器**之延遲時間， $t_G$ 為一個**邏輯閘**之延遲時間， $n$ 為正反器之數量，且兩者之**區別**僅考慮正反器數量**大於3**之情況。



十六進上數同步計數器（**漣波進位模式**）之邏輯電路圖



十六進上數同步計數器（**並行進位模式**）之邏輯電路圖

5、試說明環型計數器與強生計數器之電路結構有何差異，並進一步說明此兩種計數器之優、缺點為何？

解：

(1) 環型計數器與強生計數器之電路結構之差異如下：

- (a) 環型計數器使用  $D$  型正反器來設計較為簡單，而其電路結構是將正反器串接起來，使二進位資料可由左向右移位，且將最後一級正反器之非反相輸出端  $Q$ ，回饋至第一級正反器之輸入端。
- (b) 強生計數器亦使用  $D$  型正反器來設計較為簡單，而其電路結構亦是將正反器串接起來，使二進位資料可由左向右移位，再將最後一級正反器之反相輸出端  $\bar{Q}$ （環式計數器是由非反相端  $Q$ ），回饋至第一級正反器之輸入端，故亦可稱為扭合環式計數器 (Twist-Ring Counter)。

(2) 環型計數器與強生計數器之優、缺點如下：

- (a) 使用  $n$  個  $D$  型正反器之環型計數器，僅可實現  $n$  個計數模式之同步計數器，此為環式計數器之缺點；而環式計數器之輸出，永遠僅有一個邏輯 1，故不須經由解碼電路，即各個不同之計數狀態，可由相對之正反器輸出得到，此為環式計數器之優點。
- (b) 使用  $n$  個  $D$  型正反器之強生計數器，卻可實現  $2n$  個計數模式之同步計數器，因此在相同數目正反器的條件下，可提供較多計數模式，此為強生計數器之優點；然而需加上解碼電路，才可分辨出計數器之不同輸出模式，此為強生計數器之缺點。