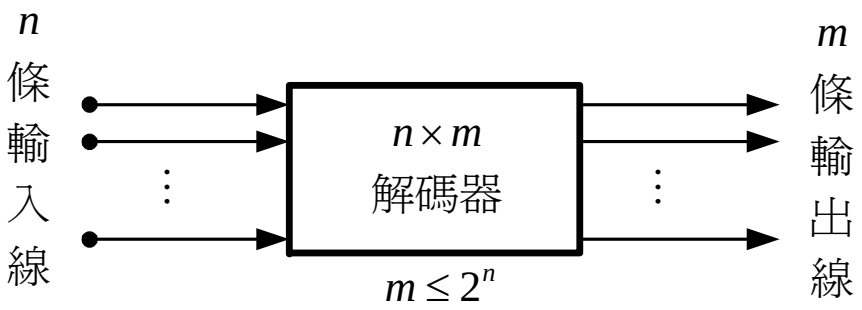
組合邏輯電路設計

— 資料處理電路

概　　述

- 本章將著重於討論各種二進位數碼之資料處理電路，包括解碼器 (Decoder)、編碼器 (Encoder)、多工器 (Multiplexer) 與解多工器 (Demultiplexer) 等單元電路之設計方法與實際之應用。

- 解碼 (Decoding) 是將某一數碼轉換成其所對應值至單一輸出的過程，即可將 n 個位元之數碼轉換成最多 m 個輸出中的一個之組合邏輯電路，稱為 $n \times m$ 解碼器 (Decoder)，而一個 $n \times m$ 解碼器之方塊圖，如下圖所示。



- 所謂數碼 (Code) 為一種按某種規則排序之數字，因此數碼種類會相當多，本節首先將討論二進位數碼 (Binary Code) 轉換為十進位數碼 (Decimal Code) 之解碼電路設計，即將 n 條輸入所組成之 2^n 條可能的輸入組合，僅對應 m 條輸出線中的一條，使此對應之輸出線成為高電位 (或低電位，依電路之設計方式而定)，其它 $m-1$ 條輸出線成為低電位 (或高電位)。
- 對具 n 個輸入變數的二進位解碼電路而言，若這些輸入變數以補數和非補數之組合形式來表示，則應有 2^n 種輸入變數之組合 (2^n 個最小項之布林函數式) 出現，而輸出變數 m 應為小於或等於 2^n ($m \le 2^n$)。

解碼電路之設計(1×2 解碼器)

◆ **1 對 2 線解碼器**：1×2 解碼器僅有 1 個 $(n=1)$ 輸入變數 x，而有 2 個 $(m=2^1=2)$ 輸出變數 Q_1 與 Q_0 之組合邏輯電路，根據解碼器之原理，可得高態致動之 1×2 解碼器的**方塊圖**與**真值表**，如下圖所示。

(a)方塊圖

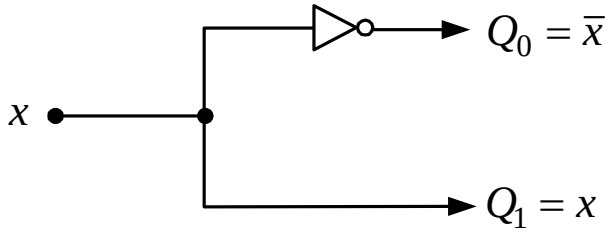
輸 入	布林函數	輸 出	
x	(最小項)	Q_0	Q_1
0	$\overline{x}$	1	0
1	x	0	1

(b) 真值表

◆ 利用以上之真值表，可得**高態致動**之 1×2 解碼器輸出 Q_1 與 Q_0 的布林函數式，即輸出變數所對應之最小項來表示如下：

$$Q_0=\overline{x} \quad 、 \quad Q_1=x$$

◆ 使用邏輯閘來實現Q_1與Q_0之布林函數式，即可得1×2解碼器之邏輯電路，如下圖所示。



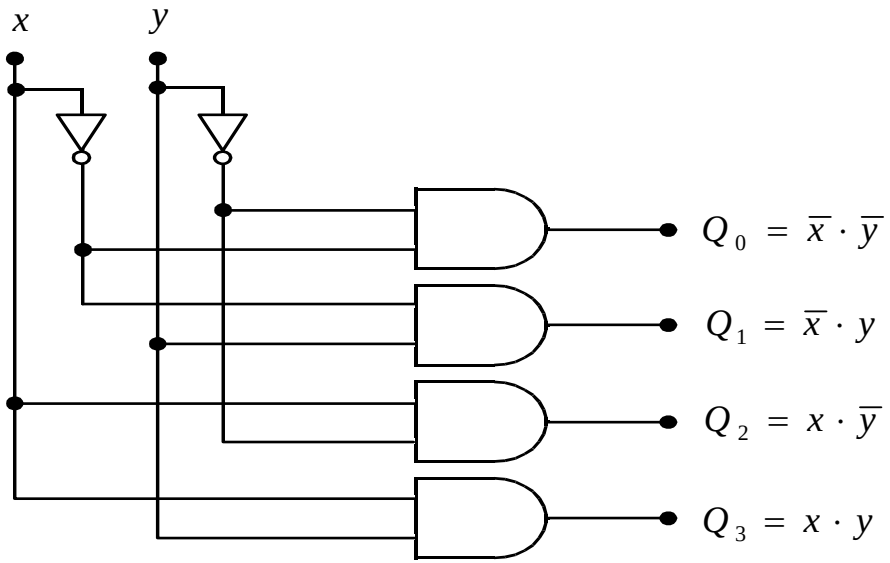
解碼電路之設計(2×4解碼器)

- **2對4線解碼器**（2×4解碼器）有2個$(n=2)$輸入變數x與y，而有4個$(m=2^2=4)$輸出變數Q_3、Q_2、Q_1與Q_0之組合邏輯電路，根據解碼器之原理，可得高態致動之2×4解碼器真值表，如下表所示。

輸 入		布林函數（最小項）	輸 出			
x	y		Q_0	Q_1	Q_2	Q_3
0	0	$\bar{x}\cdot\bar{y}$	1	0	0	0
0	1	$\bar{x}\cdot y$	0	1	0	0
1	0	$x\cdot\bar{y}$	0	0	1	0
1	1	$x\cdot y$	0	0	0	1

- 利用上面之**真值表**，使用**邏輯閘**來實現Q_3、Q_2、Q_1與Q_0所對應之最小項，即可得**高態致動**之2×4解碼器之方塊圖與邏輯電路，如下圖所示。

(a) 方塊圖



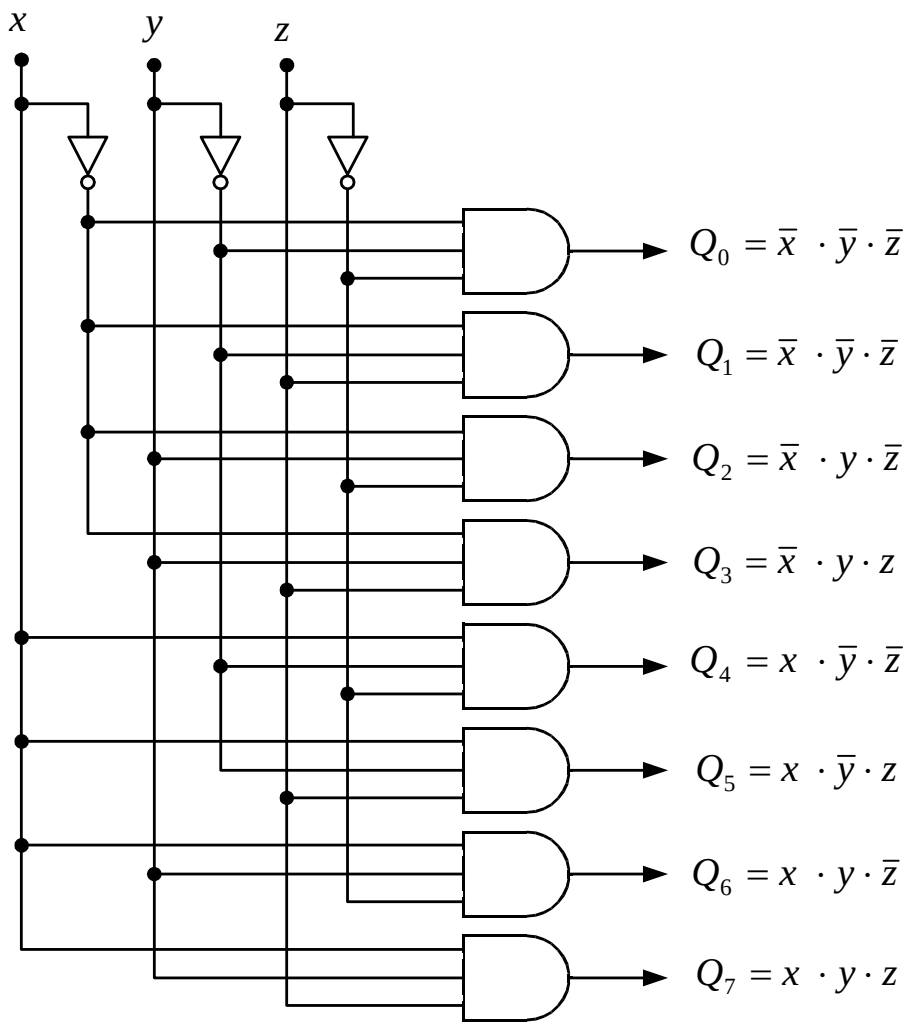
(b) 邏輯電路圖

解碼電路之設計(3×8解碼器)

◆ **3 對 8 線解碼器** (3×8解碼器) 有 3 個 ($n=3$) 輸入變數 x、y 與 z，而有 8 個 ($m=2^3=8$) 輸出變數Q_7、Q_6、Q_5、Q_4、Q_3、Q_2、Q_1與Q_0之組合邏輯電路，根據解碼器原理，即可得3×8解碼器之真值表，如下表所示。

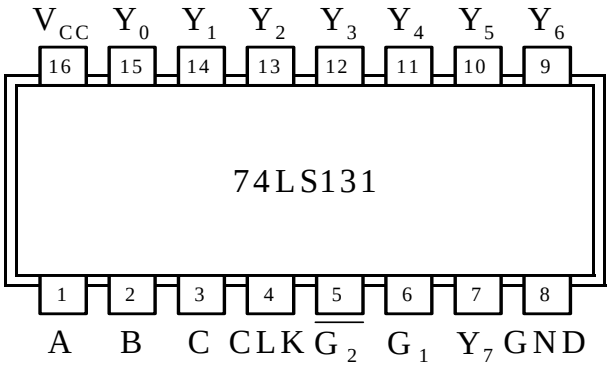
輸入			布林函數	輸出							
x	y	z	(最小項)	Q_0	Q_1	Q_2	Q_3	Q_4	Q_5	Q_6	Q_7
0	0	0	$\bar{x}\cdot\bar{y}\cdot\bar{z}$	1	0	0	0	0	0	0	0
0	0	1	$\bar{x}\cdot\bar{y}\cdot z$	0	1	0	0	0	0	0	0
0	1	0	$\bar{x}\cdot y\cdot\bar{z}$	0	0	1	0	0	0	0	0
0	1	1	$\bar{x}\cdot y\cdot z$	0	0	0	1	0	0	0	0
1	0	0	$x\cdot\bar{y}\cdot\bar{z}$	0	0	0	0	1	0	0	0
1	0	1	$x\cdot\bar{y}\cdot z$	0	0	0	0	0	1	0	0
1	1	0	$x\cdot y\cdot\bar{z}$	0	0	0	0	0	0	1	0
1	1	1	$x\cdot y\cdot z$	0	0	0	0	0	0	0	1

◆ 利用以上之真值表，使用邏輯閘來實現Q_7、Q_6、Q_5、Q_4、Q_3、Q_2、Q_1與Q_0所對應之最小項，即可得3×8解碼器之邏輯電路，如下圖所示。

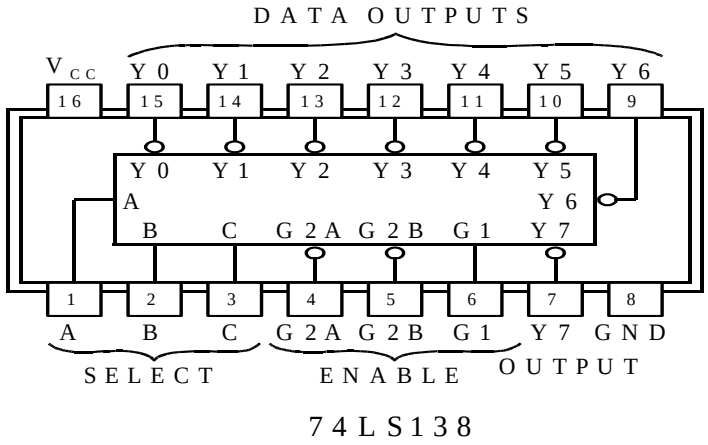

x
y
z
$Q_0 = \overline{x} \cdot \overline{y} \cdot \overline{z}$
$Q_1 = \overline{x} \cdot \overline{y} \cdot \overline{z}$
$Q_2 = \overline{x} \cdot y \cdot \overline{z}$
$Q_3 = \overline{x} \cdot y \cdot z$
$Q_4 = x \cdot \overline{y} \cdot \overline{z}$
$Q_5 = x \cdot \overline{y} \cdot z$
$Q_6 = x \cdot y \cdot \overline{z}$
$Q_7 = x \cdot y \cdot z$

3×8 解碼器(低態致動)

◆ 目前市售有兩種 3 對 8 線解碼器之套裝 IC，分別為高態致動(各個輸入組合所對應之輸出線為邏輯 1，剩下 7 條輸出線皆為邏輯 0)之 74LS131 和低態致動(各個輸入組合所對應之輸出線為邏輯 0，剩下 7 條輸出線皆為邏輯 1)74LS138 等兩種，分別如下圖所示。

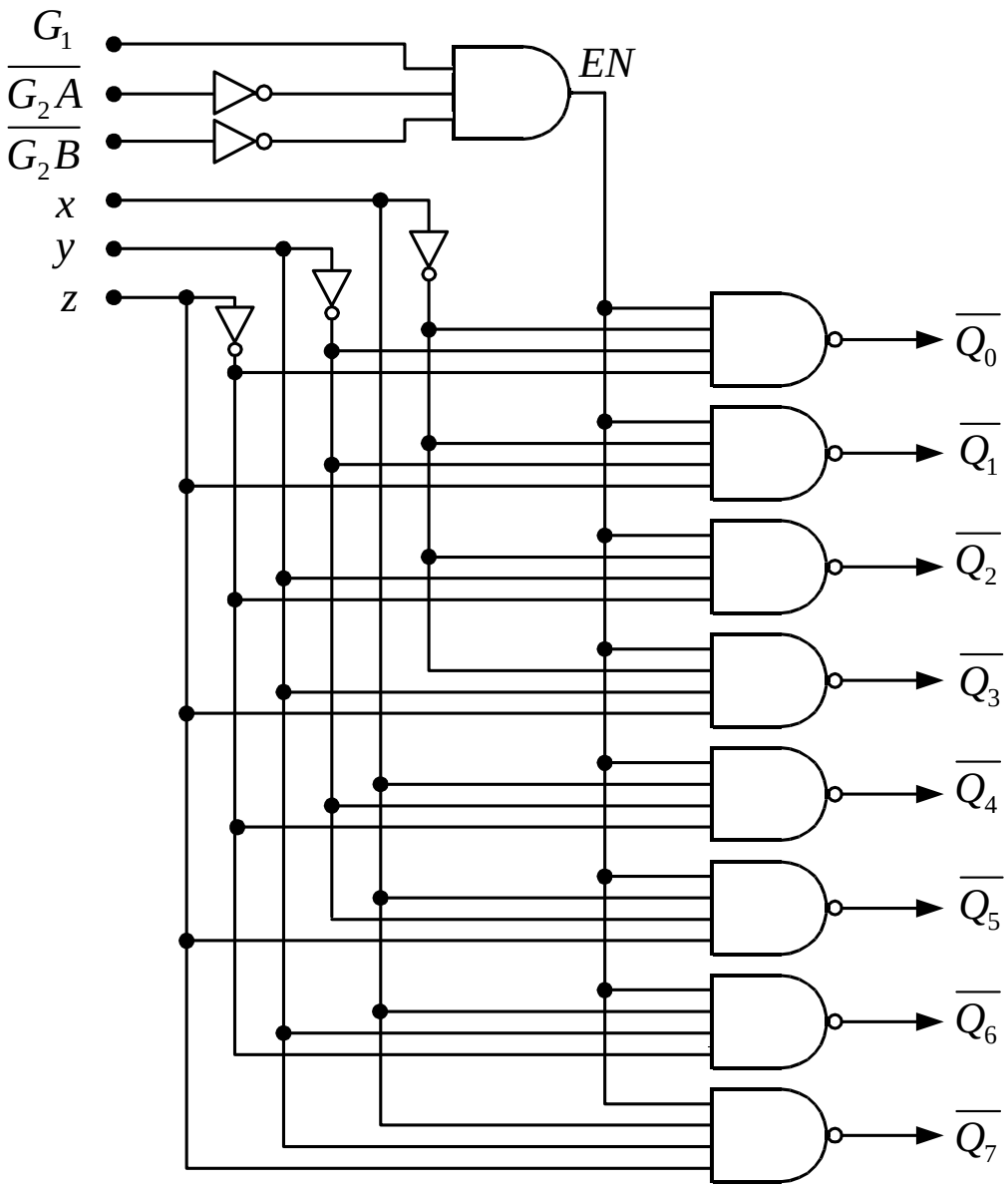


(a) 74LS131 之接腳圖



(b) 74LS138 接腳圖

◆ 74LS138 多了 G_1、$\overline{G_2A}$ 與 $\overline{G_2B}$ 等 3 條額外輸入線，這 3 條稱為致能(Enable; $EN = G_1 \cdot \overline{G_2A} \cdot \overline{G_2B}$)輸入線，而 EN 可用來決定解碼器使用與否之控制線，即當 $EN = 0$ 時，不論解碼器輸入線之邏輯值為何，輸出線全部為邏輯 1，則表示此時解碼器不使用；而當 $EN = 1$，則表示此解碼器正常使用，因此多了這 3 條致能線，可使解碼器在設計上更具彈性。

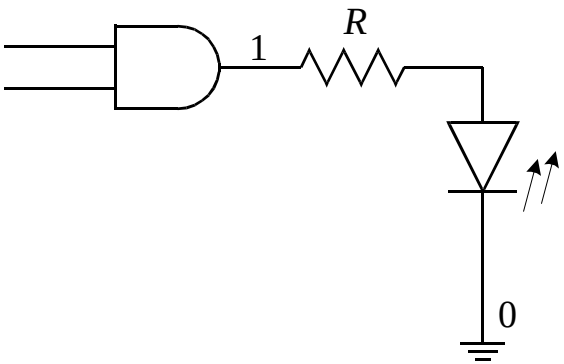

G_1
$\overline{G_2 A}$
$\overline{G_2 B}$
x
y
z
EN
$\overline{Q_0}$
$\overline{Q_1}$
$\overline{Q_2}$
$\overline{Q_3}$
$\overline{Q_4}$
$\overline{Q_5}$
$\overline{Q_6}$
$\overline{Q_7}$

低態致動解碼器之優點

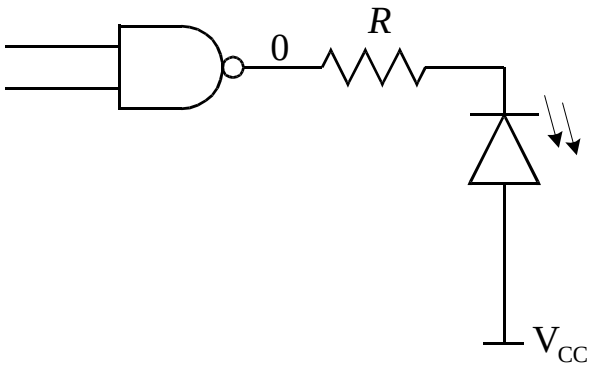
◆ 目前在設計數位系統時，大部分皆採用「低態致動」(低電位輸出) 解碼器，為提供讀者深一層的了解，特別說明使用低態致動解碼器之優點如下：

1. 組成「低態致動」解碼器的邏輯閘為 NAND 閘，而「高態致動」為 AND 閘。因使用積體電路來製造 NAND 閘比 AND 閘，所需之電晶體 (Transistor) 數目較少，故採用「低態致動」解碼器可減少積體電路 (IC) 所需之電晶體數目，相對的可降低 IC 之成本。

2. 因「低態致動」解碼器之各個輸入組合所對應之輸出線為邏輯 0 (低電位)，即相對邏輯閘之輸出僅提供一個接地點，故驅動負載所需之電流應由外接電源 V_{CC} 供給，而不是由解碼器之輸出端提供，故此種解碼器可用來驅動較大之負載；但「高態致動」解碼器之各個輸入組合所對應之輸出線邏輯 1 (高電位)，即由相對邏輯閘提供一個 V_{CC} 的電位，故驅動負載所需之電流，需由解碼器之輸出端提供，因 IC 之輸出端所能提供電流相當有限，故此種解碼器所能驅動之負載相對較小。



(a) 高態致動

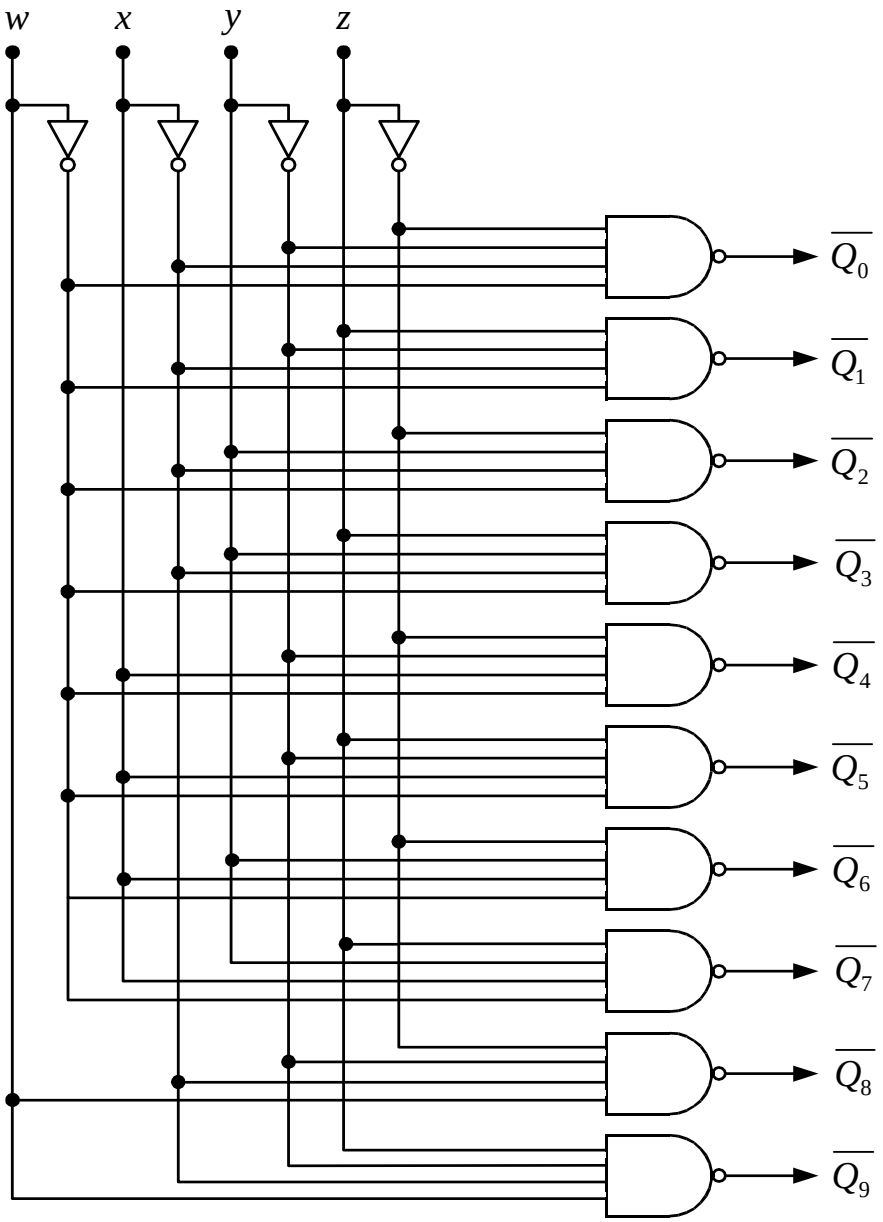


(b) 低態致動

解碼電路之設計(4×10解碼器)

- 對具 4 條 ($n=4$) 輸入線之解碼器而言，其輸出線最多可以有 16 條 ($m=2^4=16$)，因人類習慣使用十進位數，本節將討論具 4 個輸入線，但卻只有 10 條輸出線之解碼電路，稱為 4 對 10 線解碼器 (4×10 解碼器)，亦可稱為 BCD 解碼器。

- 使用邏輯閘來實現$\overline{Q_9}$、$\overline{Q_8}$、$\overline{Q_7}$、$\overline{Q_6}$、$\overline{Q_5}$、$\overline{Q_4}$、$\overline{Q_3}$、$\overline{Q_2}$、$\overline{Q_1}$與$\overline{Q_0}$所對應之最小項，即可得低態致動之4×10解碼器(BCD 解碼器)的邏輯電路，如下圖所示。


w
x
y
z
$\overline{Q_0}$
$\overline{Q_1}$
$\overline{Q_2}$
$\overline{Q_3}$
$\overline{Q_4}$
$\overline{Q_5}$
$\overline{Q_6}$
$\overline{Q_7}$
$\overline{Q_8}$
$\overline{Q_9}$

解碼器之擴充

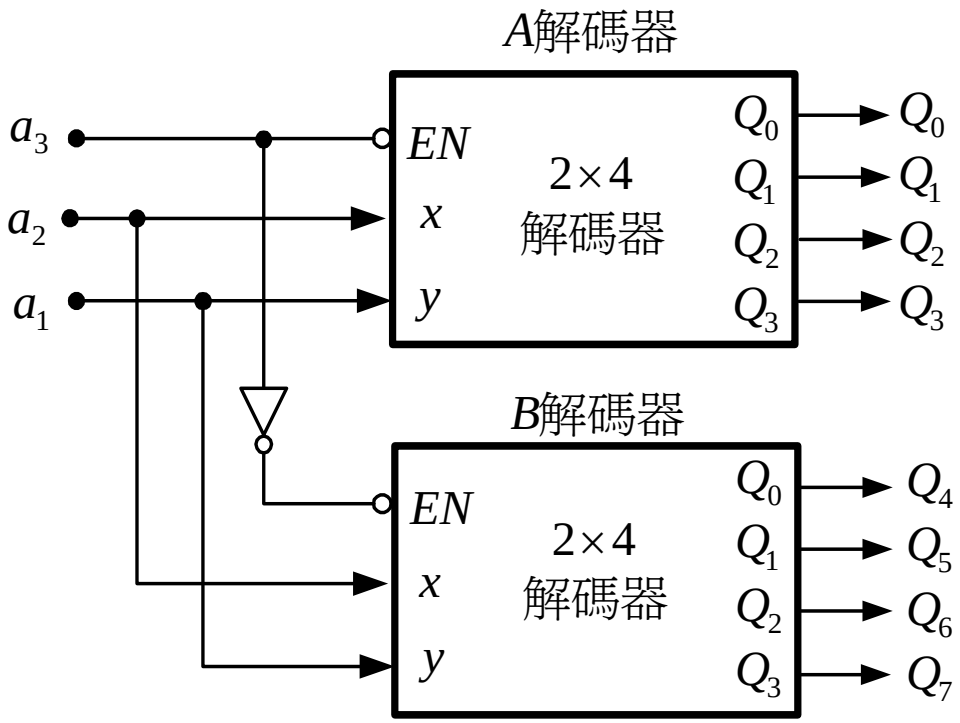
- 欲使用基本邏輯閘來繪製規格較大之解碼器，則會導致邏輯電路間連接太過於繁雜，故實用上相當不合適，若利用具致能 (Enable) 輸入之解碼器來進行串接，便可將多個規格較小之解碼器，擴充成一個規格較大之解碼器。
- 對可用來擴充之解碼器而言，必需使用具備完全解碼功能之解碼電路，即對輸入為 n 條之解碼電路而言，必須具輸出 $m = 2^n$ 條。

例題 6-1

試設計將兩個具致能輸入之 2×4 解碼器擴充為一個 3×8 解碼器的邏輯電路圖。

解

若將致能(EN) 當作 2×4 解碼器之一個輸入變數 (通常為輸入變數之最高有效位元)，且以補數與非補數之形式，分別連接至兩個 2×4 解碼器之致能輸入端，如下圖所示，即可將 2×4 解碼器擴充為 3×8 解碼器。


A解碼器
a_3
a_2
a_1
EN
x
y
2×4
解碼器
Q_0
Q_1
Q_2
Q_3
Q_0
Q_1
Q_2
Q_3
B解碼器
EN
x
y
2×4
解碼器
Q_0
Q_1
Q_2
Q_3
Q_4
Q_5
Q_6
Q_7

使用解碼器實現布林函數

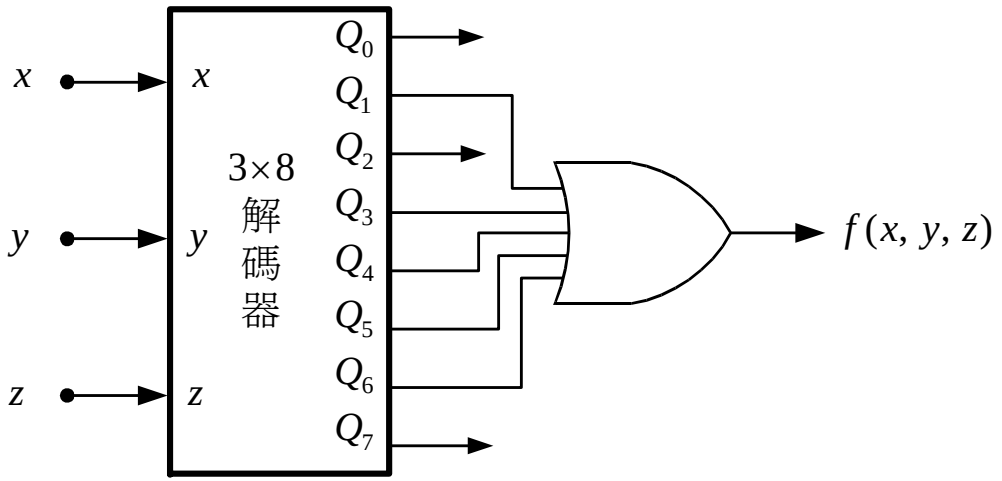
◆ 對具備完全解碼功能之解碼器而言，每一個輸出線均可表示對應至輸入變數之最小項，因 $m = 2^n$ 條輸出，故輸出可提供 n 個輸入變數所有可能組合之最小項，因此只要使用具備完全解碼功能之解碼器，並適當的將輸出(選擇所求最小項之輸出線)連接至 OR 閘之輸入端，即可實現 n 個變數之積項之和(SOP)的布林函數。

例題 6-2

試利用具備完全解碼功能之解碼器與一個 OR 閘來實現 $f(x, y, z) = x \cdot \bar{y} + \bar{x} \cdot z + x \cdot y \cdot \bar{z}$ 之布林函數式。

解

首先將已知布林函數式展開為標準積項之和，即 $f(x, y, z) = x \cdot \bar{y} + \bar{x} \cdot z + x \cdot y \cdot \bar{z} = \sum(1, 3, 4, 5, 6)$。因 $f(x, y, z)$ 有 3 條輸入線，故需使用 3×8 解碼器，且 $f(x, y, z)$ 在輸入組合為 1、3、4、5 與 6 時為邏輯 1，故需將 Q_1、Q_3、Q_4、Q_5 與 Q_6 等解碼器之輸出線連接至 OR 閘之輸入端，即可實現 $f(x, y, z)$ 布林函數，如下圖所示。



例題 6-3

試利用具完全解碼功能之解碼器與兩個 OR 閘來實現全加法器 (Full Adder)。

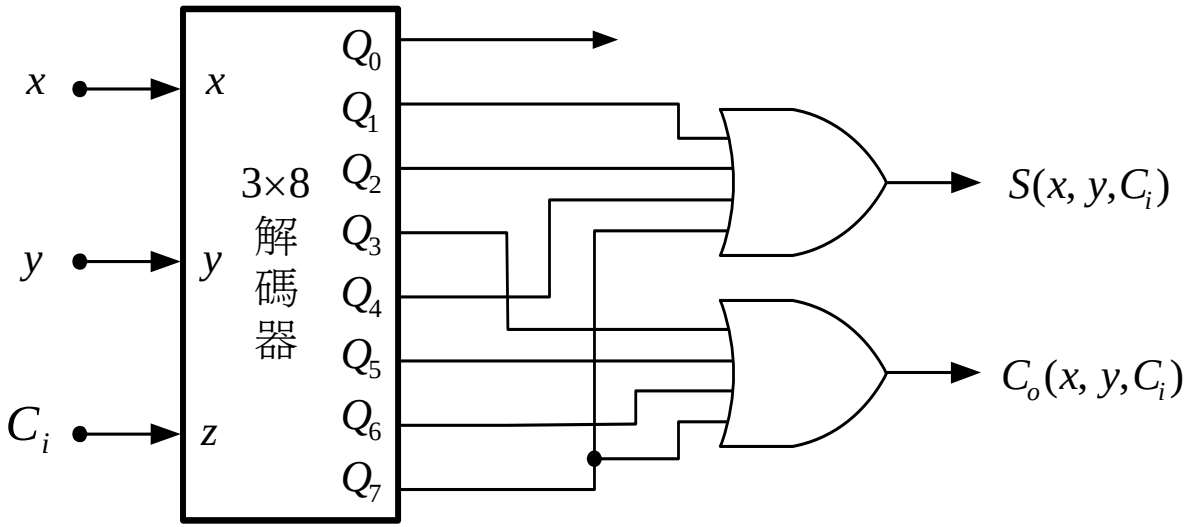
解

全加法器有 x、y 與 C_i 等 3 個輸入變數，故需用 3×8 解碼器，而有 S 與 C_o 兩個輸出變數，接著列出可執行全加器運算功能之布林函數式如下：

$$S(x,y,C_i)=\bar{x}\cdot y\cdot C_i+\bar{x}\cdot y\cdot\bar{C_i}+x\cdot\bar{y}\cdot\bar{C_i}+x\cdot y\cdot C_i=\sum(1,2,4,7)$$

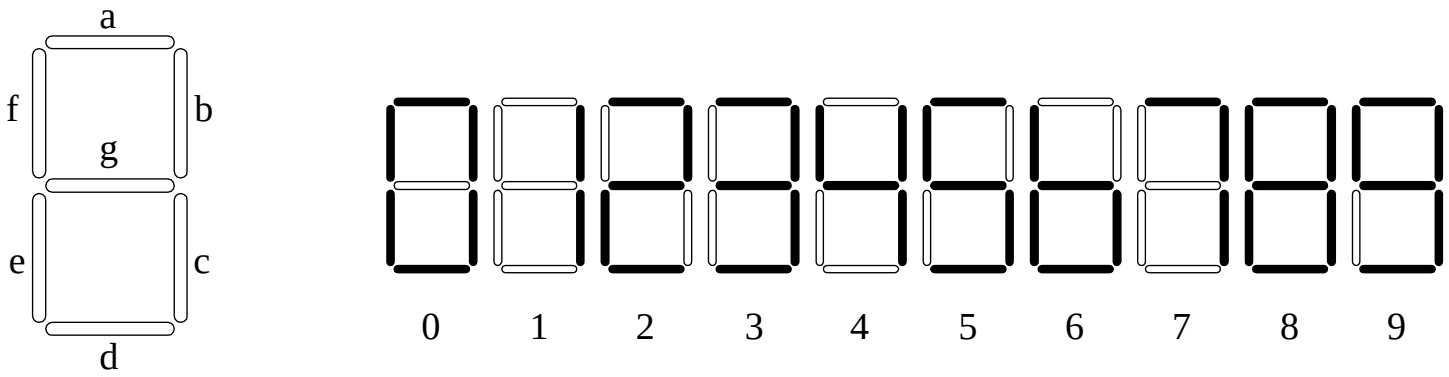
$$C_o(x,y,C_i)=x\cdot y+x\cdot C_i+y\cdot\bar{C_i}=\sum(3,5,6,7)$$

因$S(x, y, C_i)$在輸入組合為 1、2、4 與 7 時為邏輯 1，而$C_o(x, y, C_i)$在輸入組合為 3、5、6 與 7 時為邏輯 1，故首先將Q_1、Q_2、Q_4與Q_7等解碼器之輸出線連接至一個 OR 閘之輸入端，接著再將Q_3、Q_5、Q_6與Q_7等解碼器之輸出線連接至另外一個 OR 閘之輸入端，即可實現全加法器，如下圖所示。



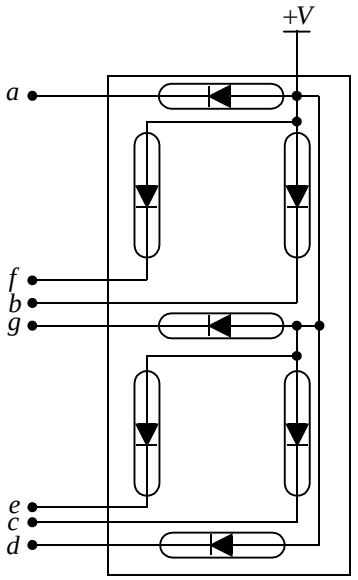
十進位顯示電路之設計

- 在數位系統中，常常需要顯示出一些數目字，以方便一般大眾的閱讀數位電路運算的結果，而能實現此種功能之電路稱為顯示電路 (Display Circuits)。
- 因人類習慣使用十進位數，故需使用十進位數字顯示器來作人與機器間之溝通。在數位系統設計上，通常以七段式顯示器 (Seven-Segment Display Devices) 來顯示十進位數字的 0 至 9 為最為普遍。
- 七段式顯示器是數位電路上最常用之十進位數顯示器材，它是由 7 支可發亮之元件所組成，而較常用來製作顯示器發亮之材料為發光二極體 (Light Emitting Diode; LED) 和液晶顯示器 (Liquid Crystal Display; LCD) 等兩種元件，因這兩種顯示元件之發光原理不同，故所需之驅動介面電路亦不相同。
- 七段式 LED 顯示器是由 a、b、c、d、e、f 與 g 等七支可發亮之 LED，排列成一個 8 字型所組成的，而以獨立控制每支 LED 發光之排列，以顯示十進位數之 0～9，如下圖所示。



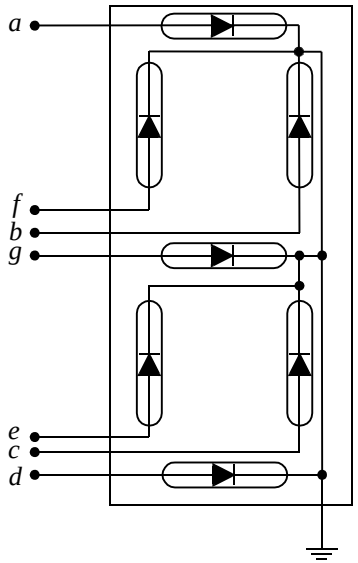
◆ 為配合不同輸出狀態之數位電路，七段 LED 顯示器，依 LED 連接方式之不同，可分為共陽極和共陰極兩種，接著將此兩種不同連接方式之特性與應用範圍說明於下：

1. 若將所有 LED 之陽極(正端)均連接在一起，並將此點連接至邏輯 1 (V_{CC})，而以每個 LED 之陰極(負端)當作輸入端，此種連接方式稱為共陽極七段 LED 顯示器，如右圖(a)所示。根據 LED 之電氣特性，必須施加順向偏壓 (Forward Bias) 於 LED 兩端 (即 LED 之陽極電壓比陰極電壓高)，才會使 LED 發光。因此解碼器之輸出為邏輯 0 (低電位) 時，才可以驅動 LED 發光，故此種連接方式需配合低態致動解碼器。



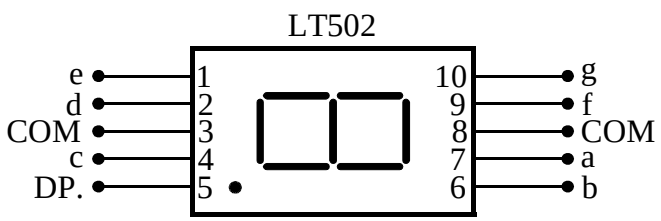
(a) 共陽極

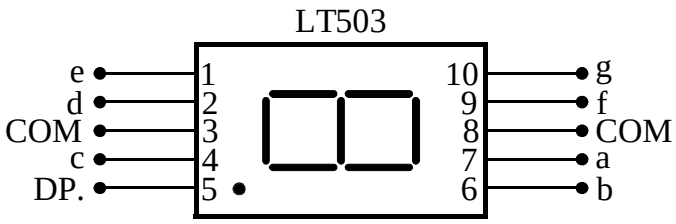
2. 若將所有 LED 之陰極（負端）均連接在一起，並將此點連接至邏輯 0（接地點），而以每個 LED 之陽極當作輸入端，此種連接方式稱為共陰極七段 LED 顯示器，如右圖 (b) 所示。因此解碼器之輸出為邏輯 1（高電位）時，才可以驅動 LED 發光，故此種連接方式需配合高態致動解碼器。



(b) 共陰極

◆ 在數位系統設計上，因低態致動解碼器較高態致動解碼器受歡迎，故共陽極之七段式 LED 顯示器較為普遍，目前市售之共陽極與共陰極七段 LED 顯示器的編號，分別為 LT502 與 LT503，而此兩種顯示裝置之外觀與接腳圖，如下圖所示。





BCD 碼對七段 LED 顯示器之解碼電路

◆ 數位電路運算所得之 BCD 碼轉成七段 LED 顯示器可接受之數碼，以顯示出相對應之十進位數字的解碼電路，稱為 BCD 碼對七段 LED 顯示器之解碼電路。因此種解碼器亦為組合邏輯電路，故可依下列步驟來設計：

1. 因電路之輸入為 BCD 碼，故應有 4 個輸入變數，分別標示為 w、x、y 與 z 之符號，而輸出為驅動 7 段 LED，故輸出變數應為 7 個，分別以 a、b、c、d、e、f 與 g 等 7 個符號來標示。
2. 接著利用上圖之十進位數字排列方式，以定義出輸出與輸入之關係，若採用共陽極七段 LED 顯示器（當輸出為邏輯 0 時，才會使 LED 發光），則欲顯示十進位數之「1」，則 b 與 c 兩段 LED 必須亮，故僅輸出 b 與 c 為邏輯 0，而其它輸出皆為邏輯 1；而欲顯示十進位之「6」，則 c、d、e、f 與 g 等 5 段 LED 必須亮，故輸出 c、d、e、f 與 g 等為邏輯 0，而其它輸出皆為邏輯 1，依此類推，則可得輸出與輸入關係之真值表，如下表所示。

十進位數	輸　入				輸　出						
	w	*x*	*y*	*z*	*a*	*b*	*c*	*d*	*e*	*f*	*g*
0	0	0	0	0	0	0	0	0	0	0	1
1	0	0	0	1	1	0	0	1	1	1	1
2	0	0	1	0	0	0	1	0	0	1	0
3	0	0	1	1	0	0	0	0	1	1	0
4	0	1	0	0	1	0	0	1	1	0	0
5	0	1	0	1	0	1	0	0	1	0	0
6	0	1	1	0	1	1	0	0	0	0	0
7	0	1	1	1	0	0	0	1	1	1	1
8	1	0	0	0	0	0	0	0	0	0	0
9	1	0	0	1	0	0	0	0	1	0	0

3. 利用卡諾圖化簡右表可得 *a*、*b*、*c*、*d*、*e*、*f* 與 *g* 等 7 個輸出之布林函數式如下：

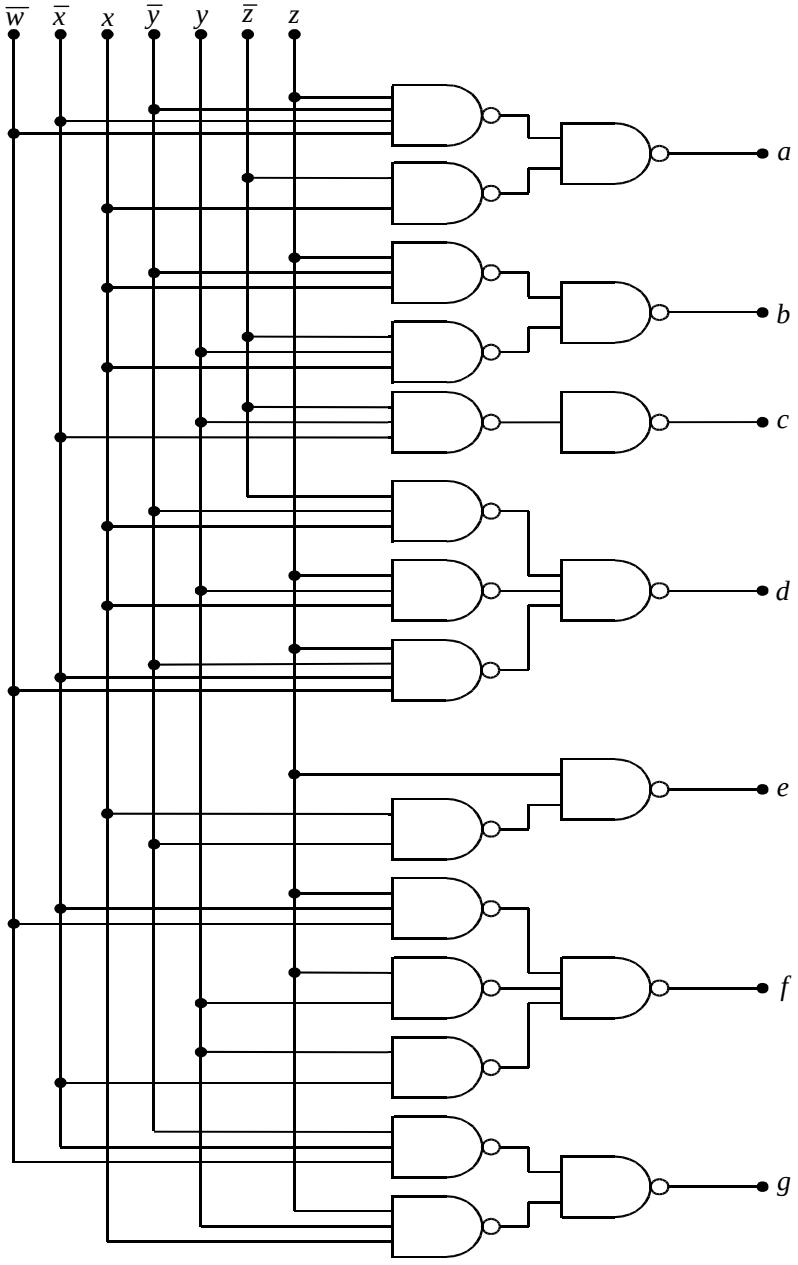
$a = \overline{w} \cdot \overline{x} \cdot \overline{y} \cdot z + x \cdot \overline{z}$ 、 $b = x \cdot \overline{y} \cdot z + x \cdot y \cdot \overline{z}$

$c = \overline{x} \cdot y \cdot \overline{z}$ 、 $d = x \cdot \overline{y} \cdot \overline{z} + x \cdot y \cdot z + \overline{w} \cdot \overline{x} \cdot \overline{y} \cdot z$

$e = z + x \cdot \overline{y}$ 、 $f = \overline{w} \cdot \overline{x} \cdot z + y \cdot z + \overline{x} \cdot y$

$g = \overline{w} \cdot \overline{x} \cdot \overline{y} + x \cdot y \cdot z$ 、

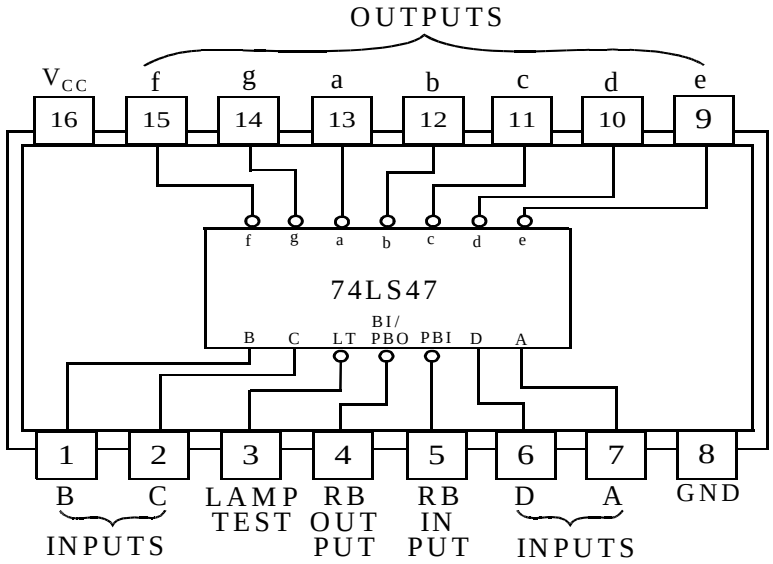
4. 最後使用 NAND 閘來實現所得之布林函數式，即可得 BCD 對七段 LED 顯示器解碼器之邏輯電路，如下圖所示。


$\overline{w}$
$\overline{x}$
x
$\overline{y}$
y
$\overline{z}$
z
a
b
c
d
e
f
g

◆ 若直接使用基本邏輯閘來實現 BCD 對七段顯示器解碼器，則會導致電路之連接相當複雜，為減少線路連接之複雜度，一般市售 TTL-IC 已將此邏輯電路積體化，以提供數位系統設計者選用。而相關解碼器之 IC 編號與類型如下：

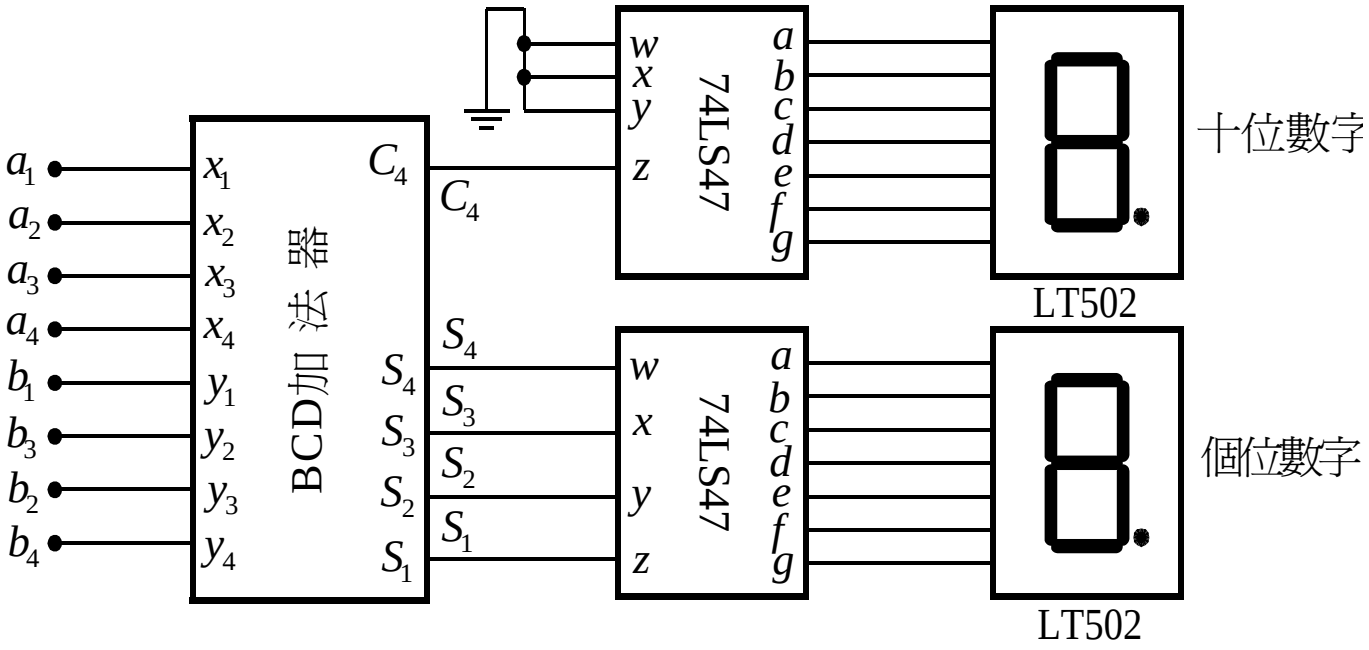
1. 低態致動之 BCD 對七段 LED 顯示器解碼器之 IC 編號為 74LS47，此 IC 應配合共陽極七段 LED 顯示器使用，而其外觀與接腳，如下圖所示。

2. 高態致動之 BCD 對七段顯示器解碼器之 IC 編號為 74LS48，此 IC 應配合共陰極七段 LED 顯示器使用。



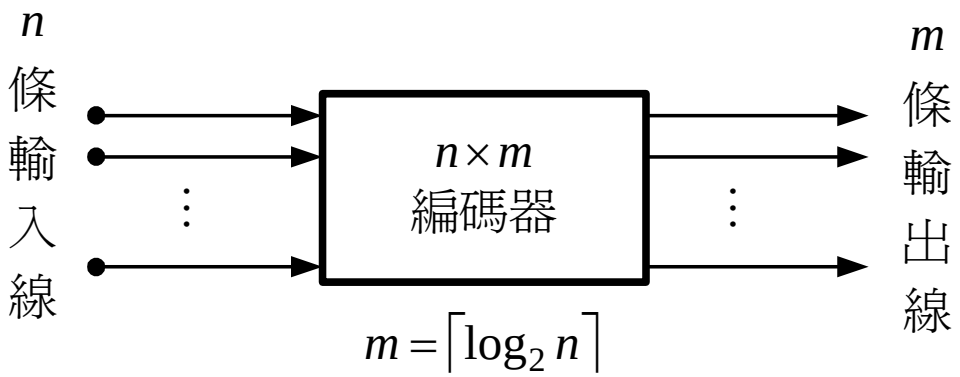
十進位加法器之顯示電路

◆ 將 BCD 加法器之輸出連接 BCD 碼至七段 LED 顯示解碼電路 (74LS47) 的輸入端，再配合七段 LED 顯示器 (LT502)，如下圖所示之邏輯電路，即可將兩個 BCD 碼相加後之結果，使用十進位數顯示出來。



編碼器

- **編碼 (Encoding) 的功用恰與解碼 (Decoding) 相反，因此編碼器 (Encoder) 之作用正好與解碼器 (Decoder) 相反，因解碼器可用 AND 閘來實現，而編碼器是用 OR 閘來實現。**
- **編碼是由單一個輸入對應至一組被編碼的輸出組合，即將 n 個輸入線轉變成為 m 個 ($m=\lceil \log_2 n \rceil$) 輸出組合之二進位碼的組合邏輯電路，稱為 $n \times m$ 編碼器 (Encoder)。**



- **本節首先討論十進位數碼 (Decimal Code) 轉換為二進位數碼 (Binary Code) 之編碼電路，即 m 條輸出線對應至 n 條輸入線所組成之 2^n 個可能的輸入組合(電路之輸出產生對應於輸入之二進位組合)。接著討論使用優先編碼器 (Priority Encoder) 來減少這些可能之錯誤，最後再討論編碼器之擴充方法，即利用多個較小規格之編碼器，組成一個較大規格之編碼器。**

編碼電路之設計(4 對 2 線編碼器)

- 2 對 1 線編碼器(2×1編碼器)有兩個輸入變數I_1與I_0，而僅有一個輸出變數 x 之組合邏輯電路，根據編碼器之原理，可得2×1編碼器為 $x = I_1$，邏輯電路圖僅需為一直線，故不需詳加討論。

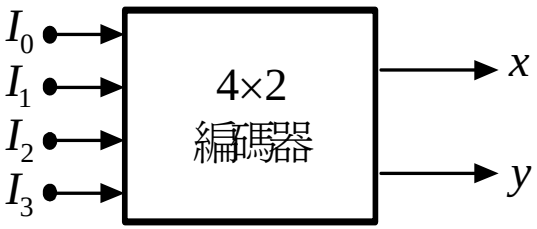
- 4 對 2 線編碼器 (4×2編碼器)有I_3、I_2、I_1與I_0等 4 個輸入變數，而有 x 與 y 等 2 個輸出變數之組合邏輯電路，根據編碼器之原理，可得4×2編碼器真值表，如下表所示。

十進位數	輸入				輸出	
	I_3	I_2	I_1	I_0	x	y
0	0	0	0	1	0	0
1	0	0	1	0	0	1
2	0	1	0	0	1	0
3	1	0	0	0	1	1

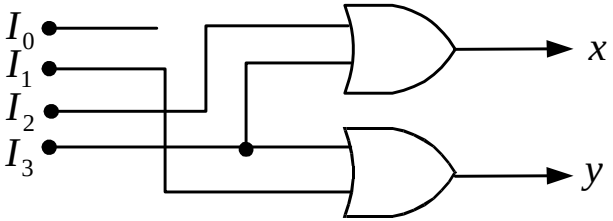
- 利用上面之真值表，可得4×2編碼器之 x 與 y 等兩個輸出之布林函數：

$$x = I_3 + I_2 \quad \text{與} \quad y = I_1 + I_3 \text{。}$$

◆ **最後使用 OR 閘來實現 x 與 y 之布林函數，即可得 4×2 編碼器之方塊圖與邏輯電路，如下圖所示。**



(a) 方塊圖



(b) 邏輯電路圖

8 對 3 線編碼器

◆ **8 對 3 線編碼器** (8×3編碼器) 有I_7、I_6、I_5、I_4、I_3、I_2、I_1與I_0等 8 個輸入變數，而有 x、y 與 z 等 3 個輸出變數之組合邏輯電路，根據編碼器原理，即可得8×3編碼器真值表，如下表所示。

十進位數	輸入								輸出		
	I_0	I_1	I_2	I_3	I_4	I_5	I_6	I_7	x	y	z
0	1	0	0	0	0	0	0	0	0	0	0
1	0	1	0	0	0	0	0	0	0	0	1
2	0	0	1	0	0	0	0	0	0	1	0
3	0	0	0	1	0	0	0	0	0	1	1
4	0	0	0	0	1	0	0	0	1	0	0
5	0	0	0	0	0	1	0	0	1	0	1
6	0	0	0	0	0	0	1	0	1	1	0
7	0	0	0	0	0	0	0	1	1	1	1

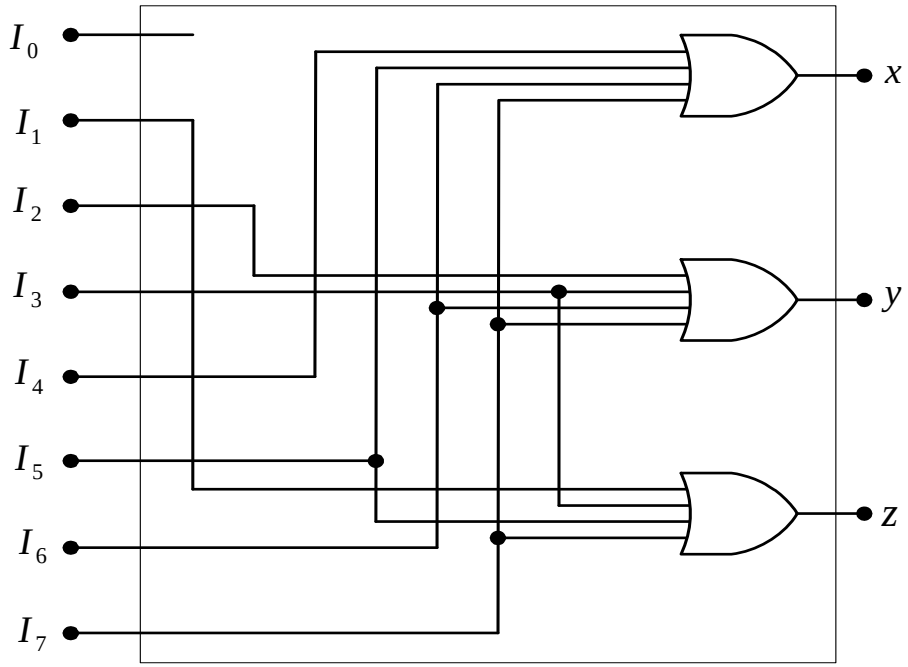
◆ 利用上面之真值表，可得8×3編碼器之 x、y 與 z 等 3 個輸出的布林函數式如下：

$$x = I_4 + I_5 + I_6 + I_7$$

$$y = I_2 + I_3 + I_6 + I_7$$

$$z = I_1 + I_3 + I_5 + I_7$$

◆ 使用 OR 閘來實現 x、y 與 z 等 3 個輸出之布林函數，即可得8×3編碼器的邏輯電路，如下圖所示。



10 對 4 線優先編碼電路之設計

- 因編碼器每一次僅允許一條輸入線啓動，為了減少錯誤發生之機率，必須利用抑制閘(Inhibit Gate)來排列編碼之優先順序 (Priority) 的編碼電路稱為優先編碼器 (Priority Encoder)。
- 優先編碼器可分高階與低階優先編碼兩種，而高階與低階之分別，相對十進位數值之優先編碼順序的不同，即若有兩條輸入線同時被啓動，則高階優先編碼器是抑制數值較低之輸入線，而使數值較高之輸入線優先編碼。
- 本節將討論具 4 個輸出變數，但卻只有 10 個輸入變數之高階10×4優先編碼器 (74LS147)，亦可稱為 BCD 高階優先編碼器，即若兩輸入線$\bar{I}_8$與$\bar{I}_6$同時被啓動 ($\bar{I}_8$與$\bar{I}_6$同時為邏輯 0)，則此電路會抑制$\bar{I}_6$之編碼，而僅對$\bar{I}_8$進行編碼。而10×4高階優先編碼器之設計步驟如下：

  1. 此電路應有$\overline{w}$、$\overline{x}$、$\overline{y}$與$\overline{z}$等 4 個輸出變數，而有$\bar{I}_9$、$\bar{I}_8$、$\bar{I}_7$、$\bar{I}_6$、$\bar{I}_5$、$\bar{I}_4$、$\bar{I}_3$、$\bar{I}_2$、$\bar{I}_1$與$\bar{I}_0$等 10 個輸入變數之組合邏輯電路。
  2. 根據編碼器定義可得，高態致動之10×4優先編碼器的真值表，如下表所示。

十進位數	輸入										輸出			
	$\bar{I}_0$	$\bar{I}_1$	$\bar{I}_2$	$\bar{I}_3$	$\bar{I}_4$	$\bar{I}_5$	$\bar{I}_6$	$\bar{I}_7$	$\bar{I}_8$	$\bar{I}_9$	$\overline{w}$	$\overline{x}$	$\overline{y}$	$\overline{z}$
0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	×	0	1	1	1	1	1	1	1	1	1	1	1	0
2	×	×	0	1	1	1	1	1	1	1	1	1	0	1
3	×	×	×	0	1	1	1	1	1	1	1	1	0	0
4	×	×	×	×	0	1	1	1	1	1	1	0	1	1
5	×	×	×	×	×	0	1	1	1	1	1	0	1	0
6	×	×	×	×	×	×	0	1	1	1	1	0	0	1
7	×	×	×	×	×	×	×	0	1	1	1	0	0	0
8	×	×	×	×	×	×	×	×	0	1	0	1	1	1
9	×	×	×	×	×	×	×	×	×	0	0	1	1	0

3. 利用真值表可知，此電路為低態致動之高階優先編碼器，即當 $\bar{I}_9 = 0$ (因編碼器為低態致動，故輸入使用邏輯 0 來啓動)，則 $\bar{I}_8$ 以下之輸入線接會被抑制 (即下表最後一行，$\bar{I}_8$ 以下之方格用「□」來標示)，故編碼器之輸出 $\overline{w}\ \overline{x}\ \overline{y}\ \overline{z} = 0110$ (1001 之 1 的補數)；而當 $\bar{I}_8 = 0$，則 $\bar{I}_7$ 以下之輸入線亦會被抑制 (真值表之最後第二行)，故編碼器之輸出 $\overline{w}\ \overline{x}\ \overline{y}\ \overline{z} = 0111$ (1000 之 1 的補數)，依此類推，即可得低態致動之 10×4 優先編碼器之 $\overline{w}$、$\overline{x}$、$\overline{y}$ 與 $\overline{z}$ 等 4 個輸出的布林函數式如下：

$$\overline{w} = \bar{I}_9 \cdot \bar{I}_8$$

$$\bar{x} = (\bar{I}_7 + I_8 + I_9) \cdot (\bar{I}_6 + I_8 + I_9) \cdot (\bar{I}_5 + I_8 + I_9) \cdot (\bar{I}_4 + I_8 + I_9)$$

$$\bar{y} = (\bar{I}_7 + I_8 + I_9) \cdot (\bar{I}_6 + I_8 + I_9) \cdot (\bar{I}_3 + I_4 + I_5 + I_8 + I_9) \cdot (\bar{I}_2 + I_4 + I_5 + I_8 + I_9)$$

$$\bar{z} = \bar{I}_9 \cdot (\bar{I}_7 + I_8 + I_9) \cdot (\bar{I}_5 + I_6 + I_8 + I_9) \cdot (\bar{I}_3 + I_4 + I_6 + I_8 + I_9) \cdot (\bar{I}_1 + I_2 + I_4 + I_6 + I_8 + I_9)$$

◆ 觀察之上述之布林函數式可知，若使用基本邏輯閘來實現此優先編碼電路，則會得到過於繁雜邏輯線路，因此請讀者自行參閱 TTL-IC 資料手冊中之詳細邏輯電路圖。

編碼器之擴充

- 利用具致能 (Enable) 輸入線之編碼器，市售套裝 IC-74LS148 為 3 對 8 線之優先編碼器，即可用致能輸入線來進行串接，以擴充編碼器之規格。
- 對可用來擴充之編碼器而言，必需使用具備完全編碼功能之編碼器，即對輸入為 n 條之編碼電路而言，其輸出必須等於 $m = \log_2 n$ 條。
- IC-74LS148 多加入 $\overline{EI}$ 、$\overline{GS}$ 與 $\overline{EO}$ 等 3 個接腳來取代致能 (Enable) 之用，以提供設計上更多的彈性。當加入這 3 個接腳後，則可將編碼器之主要邏輯功能，分成 3 個部份來說明如下：

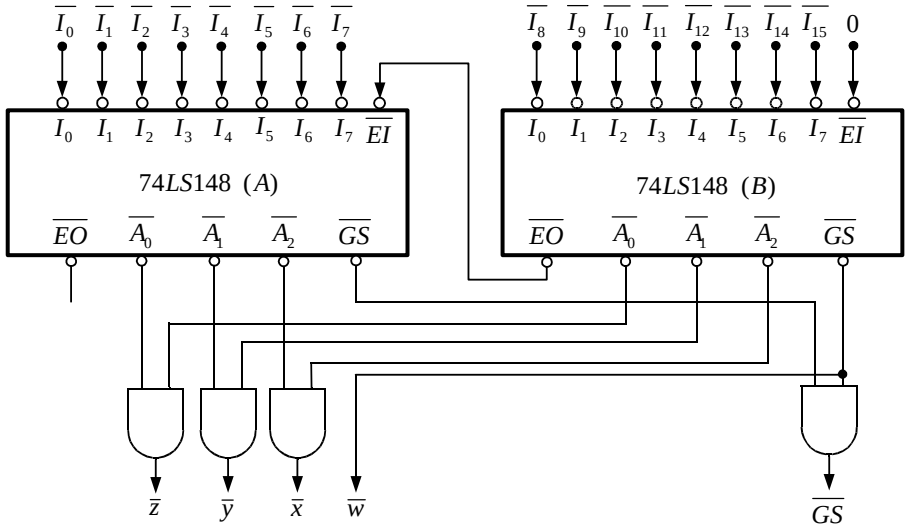
1. 當 $\overline{EI} = 1$ 時，不管編碼器輸入線之邏輯值為何，其輸出皆保持邏輯 1，即此情況 IC-74LS148 沒有編碼之功能，且在此情況下，會使 $\overline{EO} = 1$ 、$\overline{GS} = 1$ 。
2. 當 $\overline{EI} = 0$ ，且編碼器沒有輸入啓動 (所有輸入皆保持邏輯 1) 時，則 IC-74LS148 之輸出亦皆保持邏輯 1，且在此情況下，會使 $\overline{EO}$ 改變為 $\overline{EO} = 0$ ，而 $\overline{GS}$ 維持 $\overline{GS} = 1$ 。
3. 當 $\overline{EI} = 0$ ，且編碼器有一個輸入啓動 (至少有一個輸入線為邏輯 0) 時，則 IC-74LS148 執行高階優先編碼之功能，且在此情況下，會使 $\overline{EO}$ 改變為 $\overline{EO} = 1$ ，而 $\overline{GS}$ 亦改變為 $\overline{GS} = 0$ 。

例題 6-4

試繪出串接兩個 74LS148 (低態致動之8×3線優先編碼器)，以擴充為一個低態致動之16×4優先編碼器的邏輯電路圖。

解

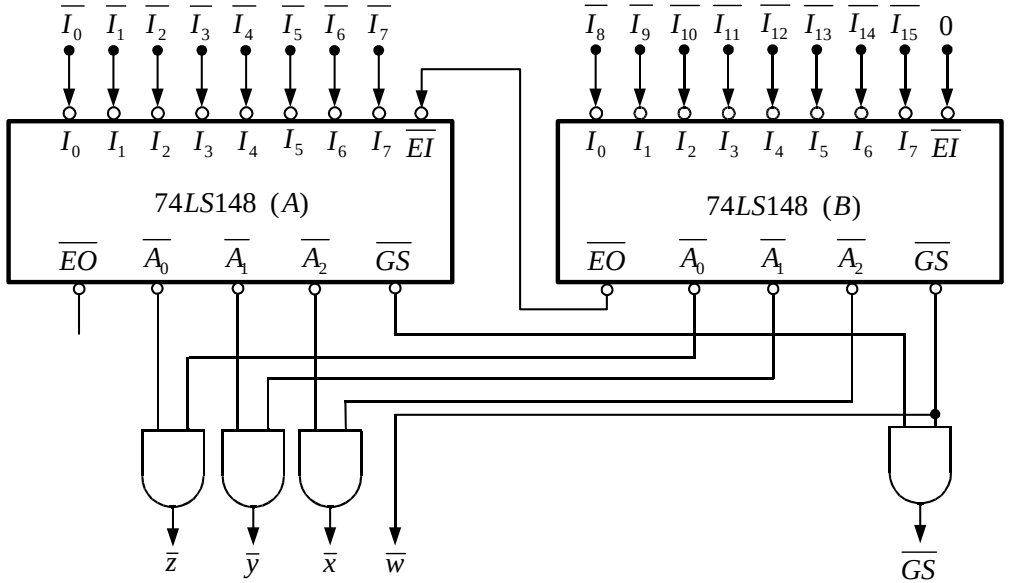
首先令 74LS148 (B) 之致能輸入$\overline{EI}=0$，接著將 74LS148 (A) 之$\overline{EI}$連接至 74LS148 (B) 之$\overline{EO}$，並將 74LS148 (B) 之$\overline{GS}$當作所求16×4優先編碼器之一個輸出線，最後將兩個 74LS148 之$\overline{GS}$連接至 AND 閘，以作為繼續擴充之用，即可串接兩個 74LS148 擴充為一個低態致動之16×4優先編碼器，如下圖所示。



接著討論串接兩個 74LS148 擴充為一個低態致動之16×4優先編碼器之原理如下：

1. 當 74LS148 (B) 之輸入線皆未啓動時，則其$\overline{EO}=0$，且此接腳連接至 74LS148 (A) 之$\overline{EI}$，則此時 74LS148 (A) 執行優先編碼之功能，故可對$\bar{I}_0\sim\bar{I}_7$進行編碼 $(\overline{GS}=\overline{w}=1)$。

2. 若 74LS148 (A) 之輸入線皆未啓動時，則此時輪到 74LS148 (B) 執行優先編碼之功能，故可對$\bar{I}_8\sim\bar{I}_{15}$進行編碼 $(\overline{GS}=\overline{w}=0)$。

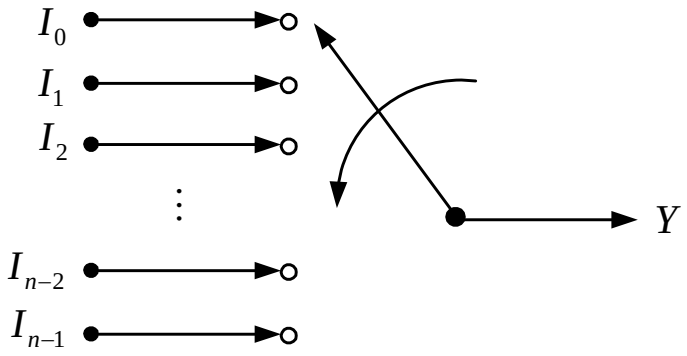


由以上之討論可知，若合併兩個 74LS148 便可得到能與16×4優先編碼器相同邏輯運算功能之編碼電路。

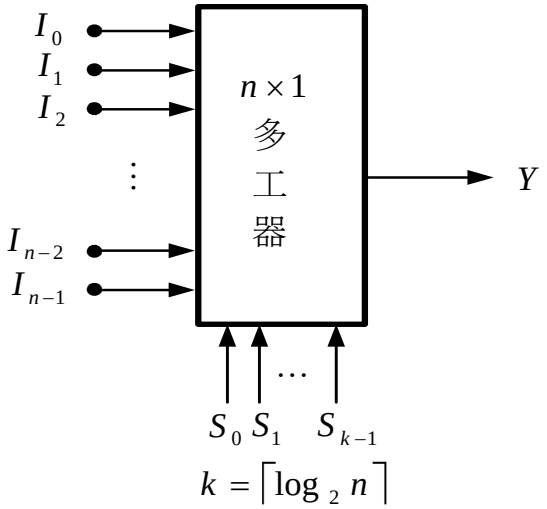
多工器

- **多工器** (Multiplexer; MUX) 或稱**資料選擇器** (Data Selector)，它可以接受**數個資料輸入** I_i，而僅**選擇其中一個輸入資料送至單一輸出線** Y，而要指定**輸入資料** I_i **傳送至輸出端** Y，可由 k **條** ($k = \lceil \log_2 n \rceil$) **選擇線之二進位組合的等效十進位數** i **來決定**。

- **數位多工器** (MUX) 通常有 n **條輸入線**、僅有 1 **條輸出線**與 k **條選擇線** (亦可稱為**位址線**) 之組合邏輯電路，而一個 $n \times 1$ **多工器之示意圖與方塊圖**，如下圖所示。

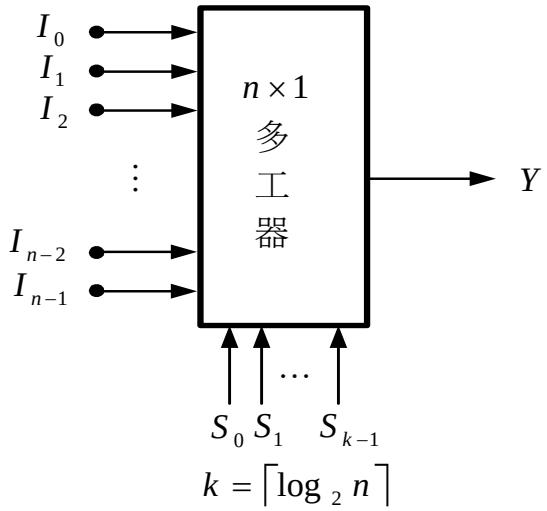


(a) 示意圖



(b) 方塊圖

◆ 使用 k 條選擇線 $(S_0,\ S_1,\ \ldots,\ S_{k-1})$，以從 n 條輸入線 $(I_0, I_1,\ \ldots,\ I_{n-1})$ 中，指定其中一條予輸出端 Y 之組合邏輯電路。因被選取之輸入線 I_i 為選擇線之二進位組合（亦可視為選擇線所組成之最小項）的等效十進位數，因此只要適當安排輸入線 I_i 與選擇線 S_k 之關係，多工器 (MUX) 亦可用來實現任何布林函數。



多工器之設計(2×1 MUX)

- 對具 n 條輸入線之數位多工器而言，若這些輸入線以 n 個十進位數 I_i 來表示，且需以 $k=\lceil \log_2 n \rceil$ 選擇線，以二進位組合形式來指定其中一輸入 I_i 之資料傳送至單一個輸出端 Y，即具有 n 條輸入之多工器，應有 k 條選擇線與一條輸出線。

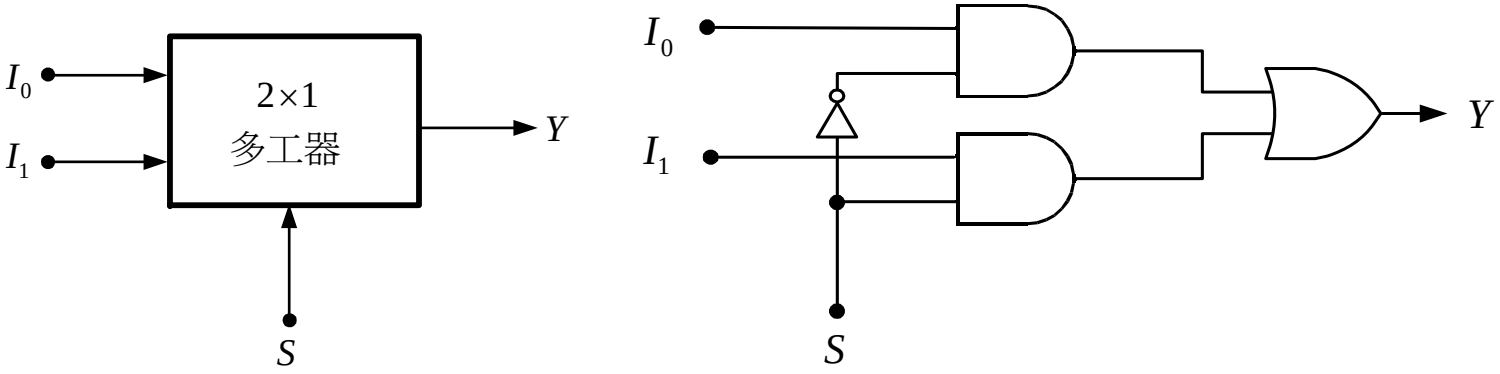
- 2 對 1 線多工器 (2×1 MUX) 有 I_1 與 I_0 等 2 個輸入變數，且需有 1 條 ($k=\lceil \log_2 2 \rceil=1$) 選擇線 S 與一個輸出變數 Y 之組合邏輯電路。根據多工器之定義可得 2×1 多工器之真值表，如下表所示。

選　擇	輸　出
S	Y
0	I_0
1	I_1

- 利用真值表，可得 2×1 多工器之輸出 Y 的布林函數式如下：

$$Y=\overline{S}\cdot I_0+S\cdot I_1$$

◆ 使用**邏輯閘**來實現輸出 Y 之布林函數，即可得2×1多工器之**方塊圖與邏輯電路**，如下圖所示。



4 對 1 線多工器

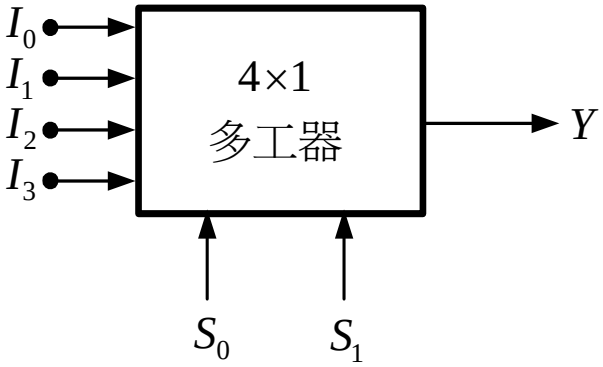
◆ 4 對 1 線多工器 (4×1MUX) 有I_3、I_2、I_1與I_0等 4 個輸入變數，且需有S_1、S_0等 2 條 ($k=\lceil\log_2 4\rceil=2$) 選擇線與一個輸出變數 Y 之組合邏輯電路。根據多工器之定義，可得4×1多工器之真值表，如下表所示。

選　擇		輸　出
S_1	S_0	Y
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

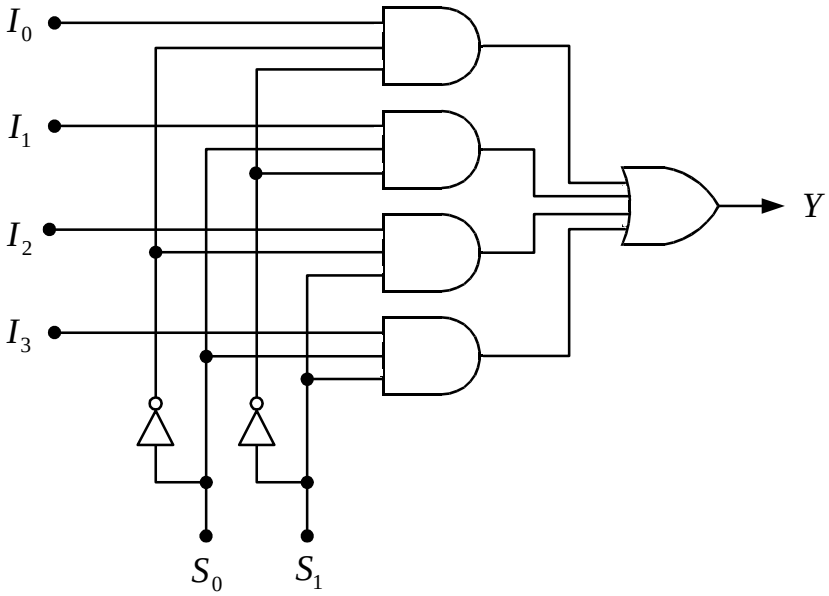
◆ 利用真值表，可得4×1多工器之輸出 Y 的布林函數式如下：

$$Y=\overline{S}_1\cdot\overline{S}_0\cdot I_0+\overline{S}_1\cdot S_0\cdot I_1+S_1\cdot\overline{S}_0\cdot I_2+S_1\cdot S_0\cdot I_3$$

◆ 使用**邏輯閘**來實現輸出 Y 之布林函數，即可得 4×1 多工器之**方塊圖與邏輯電路**，如下圖所示。



(a) 方塊圖



(b) 邏輯電路圖

具致能輸入之 8 對 1 線多工器

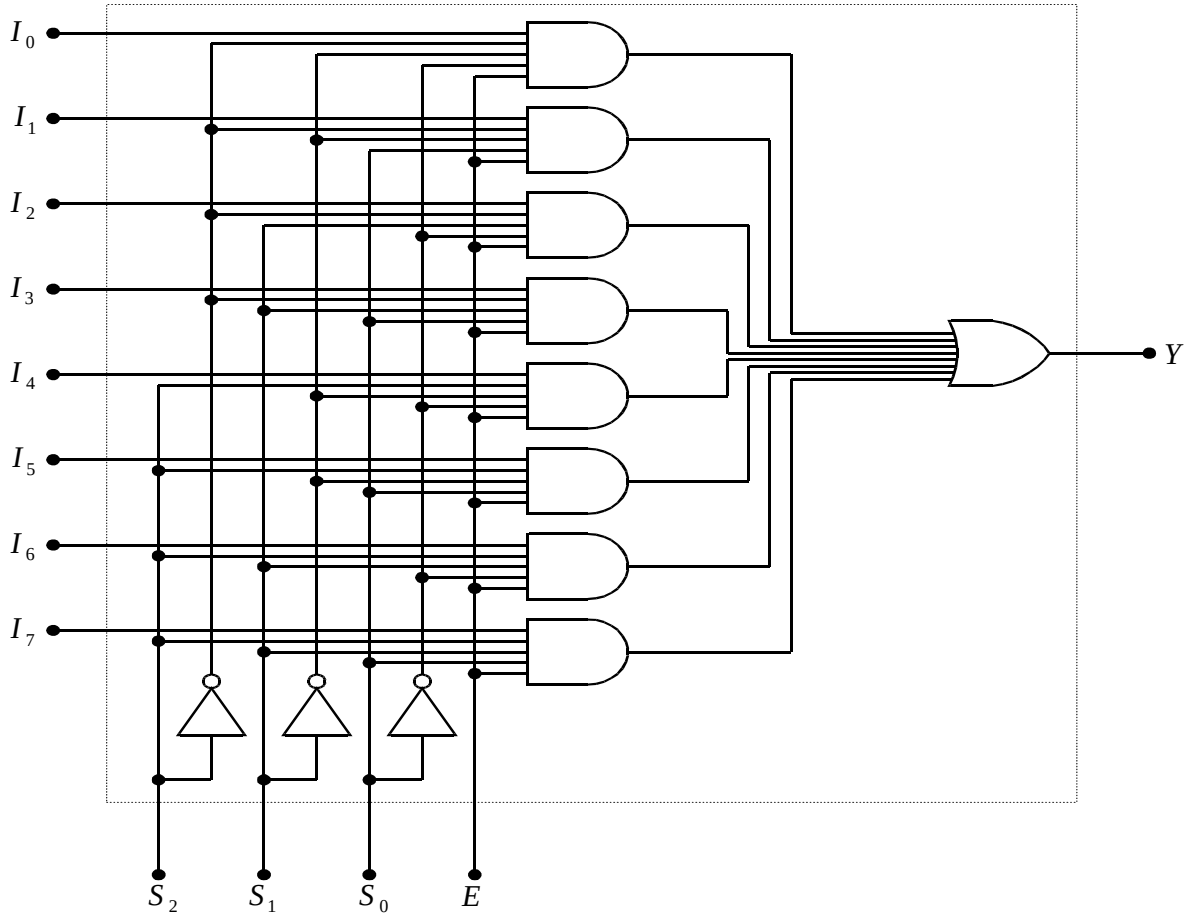
◆ 具致能輸入之 8 對 1 線多工器 (8×1多工器) 有 E (致能)、I_7、I_6、I_5、I_4、I_3、I_2、I_1 與 I_0 等 9 個輸入變數，且需 S_2、S_1 與 S_0 等 3 條 ($k=\lceil \log_2 8 \rceil = 3$) 選擇線與一個輸出變數 Y 之組合邏輯電路，而致能(E)用來決定多工器是否使用之控制線。根據具致能輸入之多工器之定義，可得8×1多工器之真值表，如下表所示。

致能	選　擇　線			輸　　出
E	S_2	S_1	S_0	Y
0	×	×	×	0
1	0	0	0	I_0
1	0	0	1	I_1
1	0	1	0	I_2
1	0	1	1	I_3
1	1	0	0	I_4
1	1	0	1	I_5
1	1	1	0	I_6
1	1	1	1	I_7

◆ 利用真值表，可得具致能輸入8×1多工器之輸出 Y 的布林函數式如下：

$$Y = E\cdot(I_0\cdot\overline{S}_2\cdot\overline{S}_1\cdot\overline{S}_0 + I_1\cdot\overline{S}_2\cdot\overline{S}_1\cdot S_0 + I_2\cdot\overline{S}_2\cdot S_1\cdot\overline{S}_0 + I_3\cdot\overline{S}_2\cdot S_1\cdot S_0 +$$
$$I_4\cdot S_2\cdot\overline{S}_1\cdot\overline{S}_0 + I_5\cdot S_2\cdot\overline{S}_1\cdot S_0 + I_6\cdot S_2\cdot S_1\cdot\overline{S}_0 + I_7\cdot S_2\cdot S_1\cdot S_0\ )$$

◆ 使用邏輯閘來實現輸出 Y 之布林函數，即可得具致能輸入之8×1多工器之邏輯電路，如下圖所示。



使用多工器實現布林函數

- 具有 n 條輸入線與選擇線等於 $k(k=\log_2 n)$ 條之多工器而言，因 k 條選擇線可提供 n 條輸入線的完全解碼，故可視為一個通用邏輯模組 (University Logic Module)，因此利用具完全解碼功能之多工器，若分別將輸入線與選擇線給予的適當連接與定義，便可用來實現任何布林函數。

- 對使用 $n\times 1$ 多工器來實現布林函數而言，則輸入線 I_i 與選擇線 $(S_{k-1},...,S_0)$ 所組成的最小項 m_i 之關係為

$$f(S_{k-1},...,S_0)=\sum_{i=0}^{n-1} I_i\cdot m_i$$

- 欲實現 $k(k=\log_2 n)$ 個變數之布林函數時，可使用選擇線數目小於或等於 k 的多工器，即可使用小於 $2^k\times 1$ 多工器來實現 k 個變數之布林函數，即

  1. 若使用 $2^k\times 1$ 多工器來實現 k 個變數之布林函數時，則可使用 k 條選擇線來代表 k 個變數。

  2. 若使用 $2^{k-\alpha}\times 1$ 多工器來實現 k 個變數之布林函數式時，除使用 $(k-\alpha)$ 條選擇線來代表 $(k-\alpha)$ 個變數外，還需用輸入線來代表剩餘之 α 個變數。

◆ 選用的α值愈大 (α值必須為整數)，則所使用多工器的規格會愈小，而當所選用之α值為 2 以上時，將會導致加於多工器輸入之變數大於或等於 2 個，故必須對這些變數加以編碼(即需這些變數進行邏輯化簡，以簡化編碼所需之邏輯閘，以節省所需之硬體成本)。

例題 6-5

試分別使用來8×1與4×1多工器實現 $f(x, y, z) = x \cdot \bar{y} + \bar{x} \cdot z + x \cdot y \cdot \bar{z}$ 布林函數式。

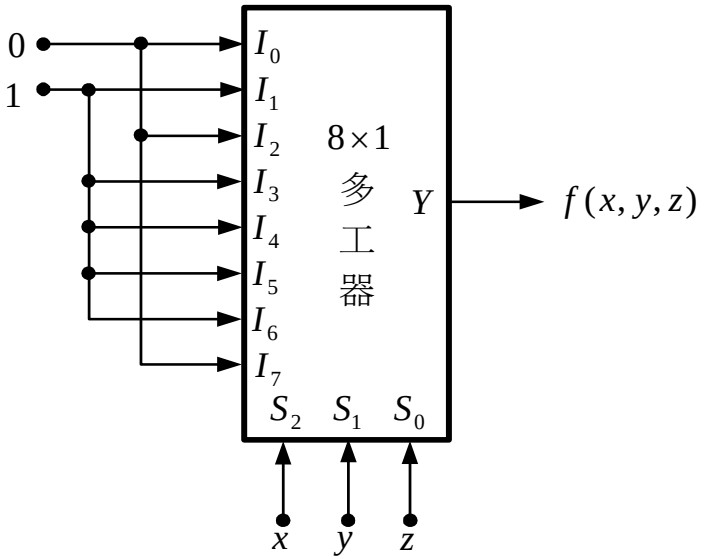
解

首先展開已知布林函數式為標準積項之和，即 $f(x, y, z) = x \cdot \bar{y} + \bar{x} \cdot z + x \cdot y \cdot \bar{z} = \sum(1, 3, 4, 5, 6)$。

1. $f(x, y, z)$ 之布林函數包含 3 個變數，若使用8×1多工器實現時，因$k = \log_2 8 = 3$，故可使用 3 條選擇線S_2、S_1與S_0來取代變數 x、y 與 z。接著製作一個選擇線S_2、S_1與S_0所組成最小項之等效十進位數 i 與變數 x、y 與 z 之對應關係表，如下表所示。

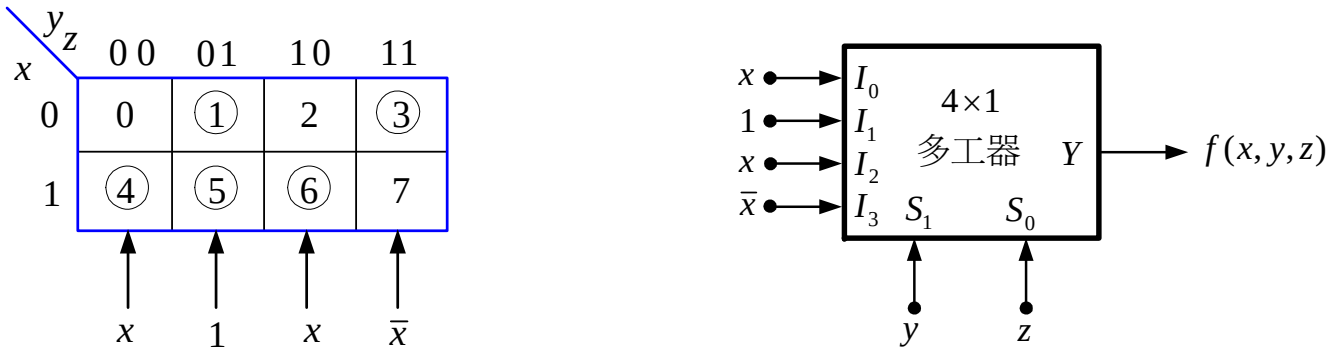
x y z	0	1	2	3	4	5	6	7

利用上表對應至所求布林函數之最小項的等效十進位數 i 圈選起來，再令所有圈選之部份所對應的輸入線 $I_i = 1$，而令無圈選之部份的輸入線 $I_i = 0$，便可得到使用 8×1 多工器來實現 $f(x, y, z)$ 之邏輯電路圖，如右圖所示。

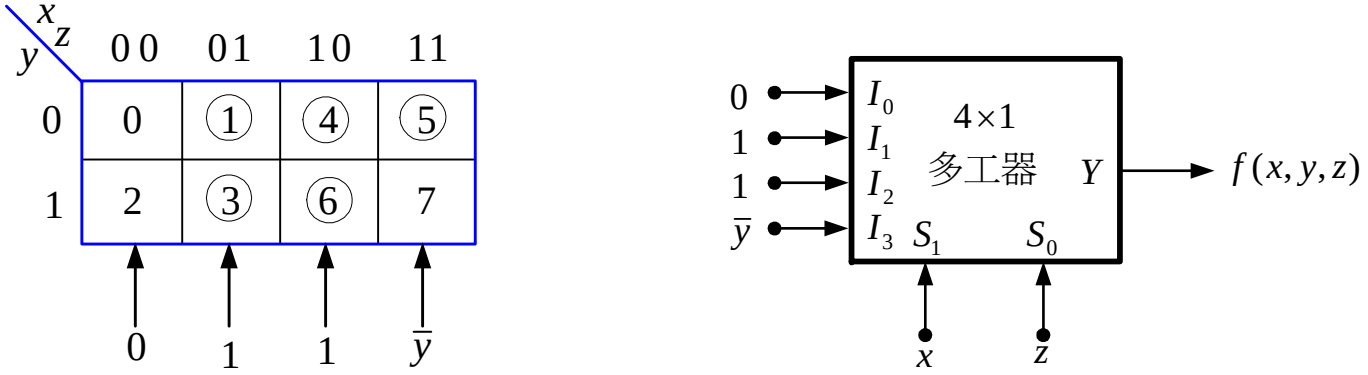


2. 因 $f(x, y, z)$ 之布林函數包含 3 個變數，若使用 4×1 多工器實現時，因 $k = \log_2 4 = 2$，故僅可使用 2 條選擇線 S_1 與 S_0 來取代 x、y 與 z 之其中兩變數，而剩下 1 個變數需用輸入線來取代。

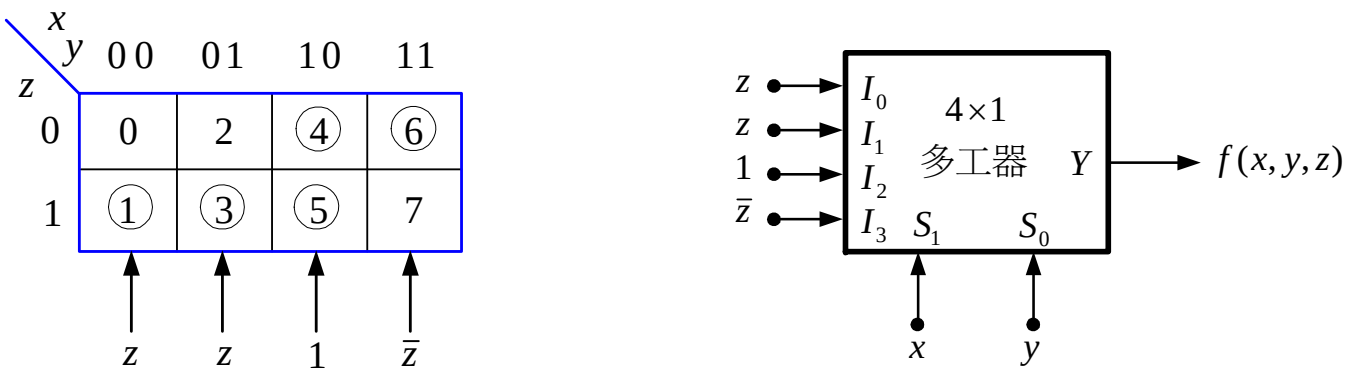
因選擇線與變數之對應關係可用來決定輸入線 I_i 之值，故欲使用 4×1 多工器實現 3 變數之布林函數，則輸出 Y 等於其中 2 條選擇線所組成最小項 2 次。接著利用右圖對應至所求布林函數之最小項的等效十進位數 i 圈選起來(列出 3 種輸入 I_i 與選擇線 S_1、S_0 所組成最小項之等效十進位數 i 與輸入變數 x、y 與 z 的對應關係表)，最後再整理圈選 i 之輸入線 I_i 所對應變數關係，便可得到使用 4×1 多工器來實現 $f(x, y, z)$ 之邏輯電路圖，如下圖所示。



(a) 以 $I_i \rightarrow x$ 、$S_1 \rightarrow y$ 與 $S_0 \rightarrow z$



(b) 以 $S_1 \rightarrow x$ 、$I_i \rightarrow y$ 與 $S_0 \rightarrow z$



(c) 以 $S_1 \rightarrow x$ 、 $S_0 \rightarrow y$ 與 $I_i \rightarrow z$

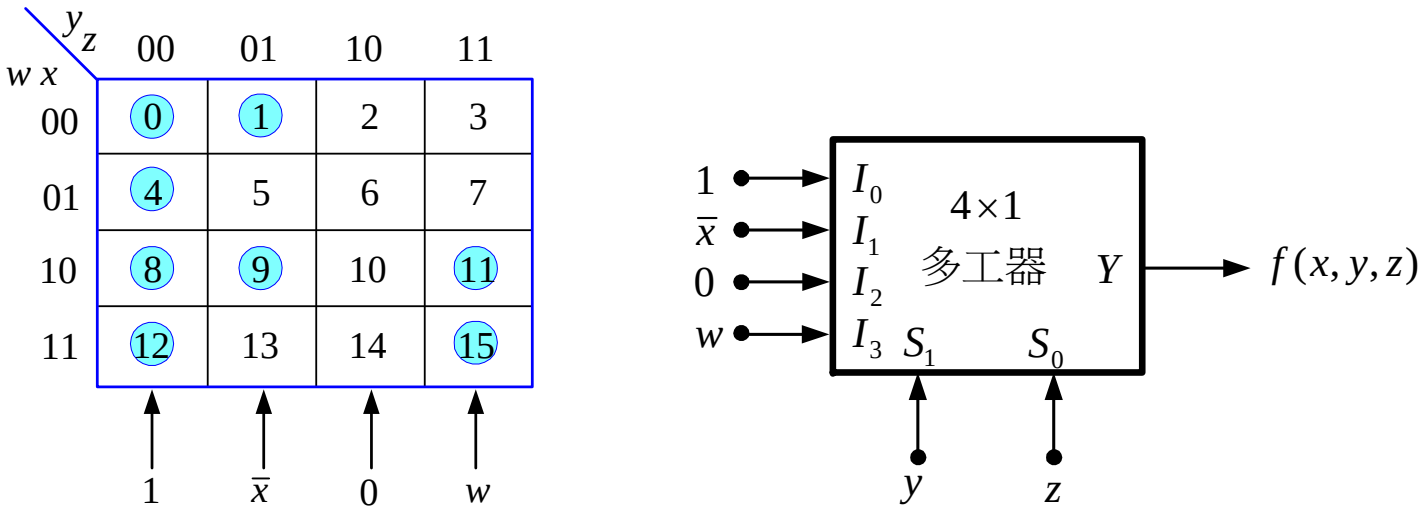
例題 6-6

試使用 4×1 多工器實現 $f(w, x, y, z)=\sum(0, 1, 4, 8, 9, 11, 12, 15)$ 布林函數。

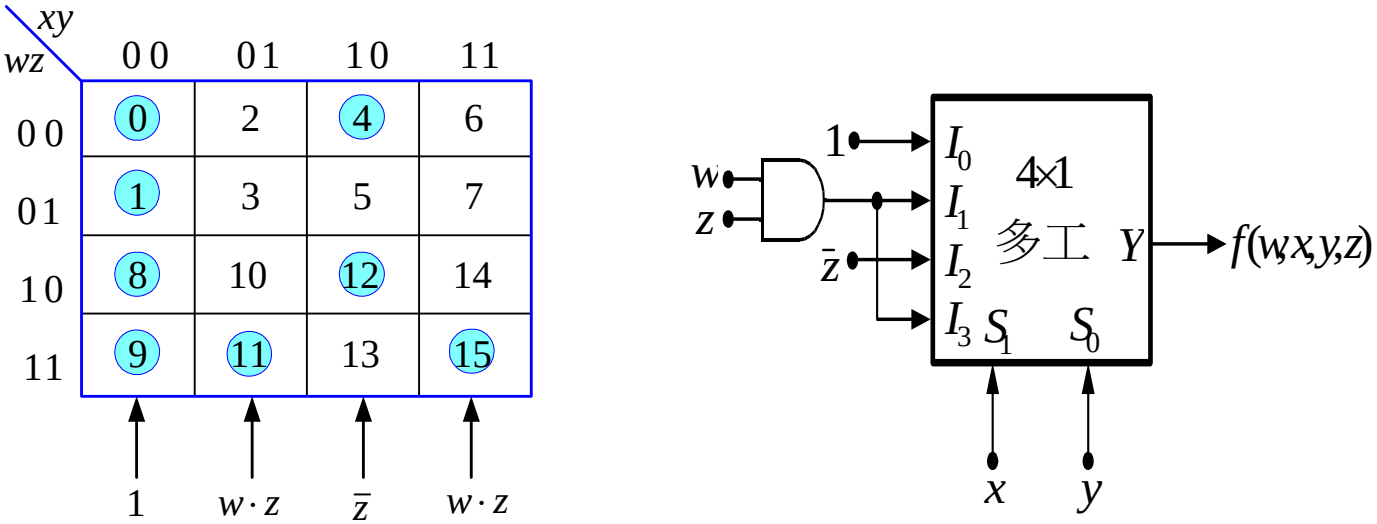
解

已知布林函數 $f(w, x, y, z)$ 包含 4 個變數，當使用 4×1 多工器實現時，因 $k=\log_2 4=2$，故僅可使用 2 條選擇線 S_1 與 S_0 來取代 w、x、y 與 z 之其中 2 個變數，而剩下 2 個變數需用輸入線來取代。

因選擇線與變數之對應關係，可用來決定輸入線 I_i 之值，故欲使用 4×1 多工器實現 4 變數之布林函數，輸出 Y 等於 2 條選擇線所組成最小項 4 次。因此應有 6 種方法 $\left(C_2^4=\frac{4\times3}{2}=6\right)$ 可指定輸入 I_i 與選擇線 S_1、S_0 所組成最小項之十進位數 i 與變數 w、x、y 與 z 之對應關係。未免佔用太多篇幅，本例題僅以兩種做法提供讀者參考，如下圖所示。



(a) 以 $I_i \rightarrow (w$ 與 $x)$、$S_1 \rightarrow y$ 與 $S_0 \rightarrow z$



(b) 以 $I_i \rightarrow (w$ 與 $z)$、$S_1 \rightarrow x$ 與 $S_0 \rightarrow y$

多工器之擴充

◆ 若欲使用基本邏輯閘來繪製規格較大之多工器，則會導致邏輯電路連接太過於繁雜，故於實際應用上相當不合適，若串接幾個具致能 (Enable) 輸入之多工器，便可將規格較小之多工器，擴充成一個規格較大之多工器。

例題 6-7

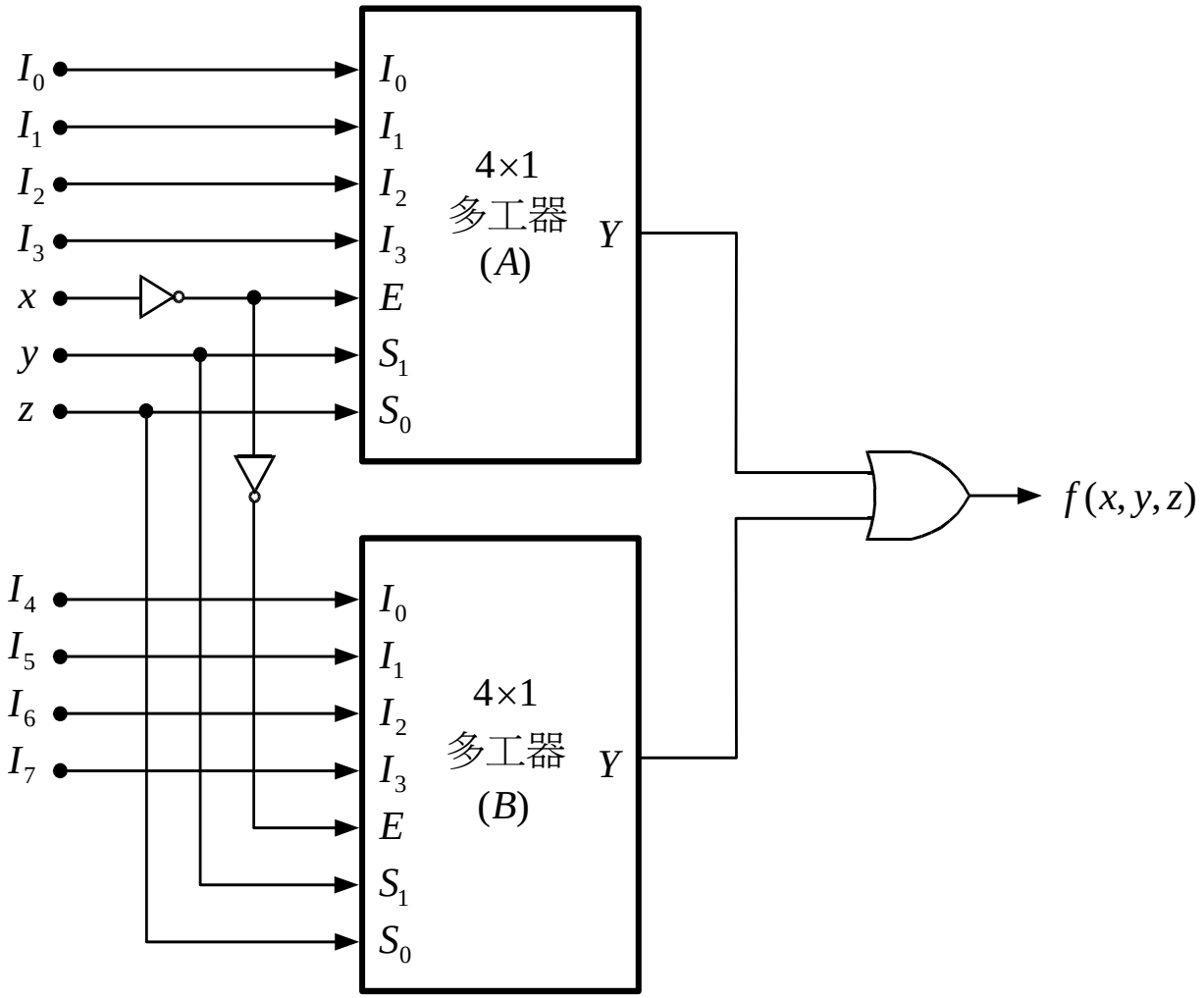
試設計將兩個具致能輸入之4×1多工器擴充為一個8×1多工器的邏輯電路圖。

解

將致能輸入端 (E) 當作4×1多工器之一個輸入變數，且以補數與非補數之形式，分別連接至兩個4×1多工器之致能輸入端，即可串接兩個4×1多工器擴充為8×1多工器，如右圖所示。

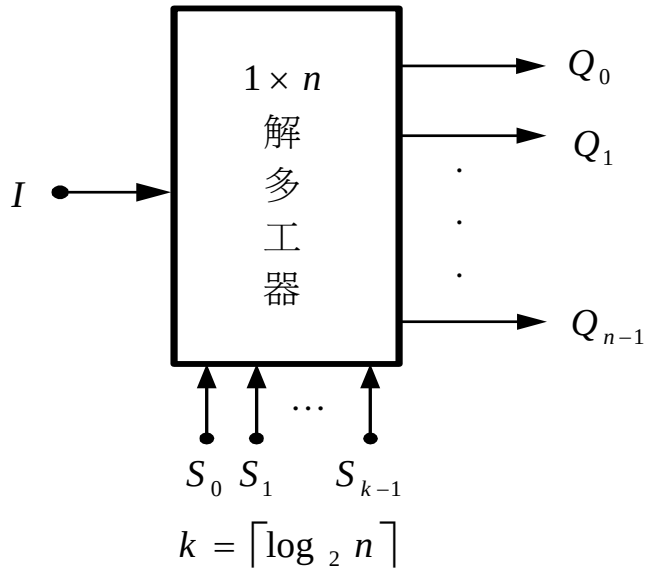
1. 當$x=0$時，會使多工器 (A) 致能輸入$E=1$，則僅多工器 (A) 有作用，即僅對輸入I_0至I_3與輸出 Y 有功能。
2. 當$x=1$時，會使多工器 (B) 致能輸入$E=1$，則僅多工器 (B) 有作用，即僅對輸入I_4至I_7與輸出 Y 有功能。

由以上之討論可知，串接兩個4×1多工器可得到與8×1多工器相同邏輯運算功能之邏輯電路，如下圖所示。



解多工器

- 解多工器(Demultiplexer; DeMUX)可將單一個輸入 I 傳送到 n 個輸出線 Q_i 中之任一條，而指定輸入 I 傳送至輸出線 Q_i，可由 k 條 $(k = \lceil \log_2 n \rceil)$ 選擇線之二進位組合的等效十進位數 i 來決定。
- 數位解多工器通常僅有一條輸入線 I、n 條輸出線與 k 條選擇線之組合邏輯電路，而一個 $1 \times n$ 解多工器之方塊圖，如下圖所示。



- 解多工器之 k 條選擇線 $(S_0, S_1, \ldots, S_{k-1})$，可指定輸入 I 傳送 n 條輸出線 $(Q_0, Q_1, \ldots, Q_{n-1})$ 中之一條的組合邏輯電路。因被指定等於輸入 I 之輸出線 Q_i 為選擇線之二進位組合的等效十進位數 i，故只要經過適當指定與安排，解多工器 (DeMUX) 亦可用來實現任何布林函數。

解多工器之設計(1×8解多工器)

- 若將解多工器之輸入線當作解碼器之致能 (Enable) 輸入線，選擇線當作解碼器之輸入線，而輸出線亦當作解碼器之輸出線，則解多工器與解碼器之電路結構完全相同，故解多工器與解碼器之設計方法完全相同。
- 1 對 8 線解多工器僅有一個輸入線 I，而有Q_7、Q_6、Q_5、Q_4、Q_3、Q_2、Q_1與Q_0等 8 個輸出變數與S_2、S_1與S_0等 3 條 ($k=\lceil \log_2 8 \rceil = 3$) 選擇線之組合邏輯電路。根據解多工器之定義，可得1×8解多工器之真值表，如下表所示。

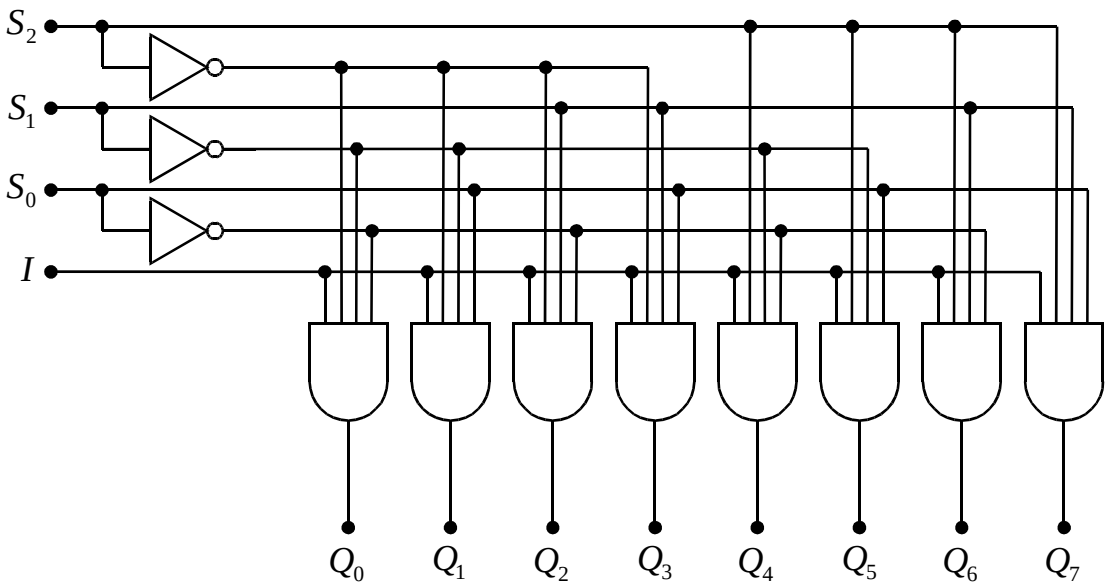
選 擇 線			輸 出							
S_2	S_1	S_0	Q_0	Q_1	Q_2	Q_3	Q_4	Q_5	Q_6	Q_7
0	0	0	I	0	0	0	0	0	0	0
0	0	1	0	I	0	0	0	0	0	0
0	1	0	0	0	I	0	0	0	0	0
0	1	1	0	0	0	I	0	0	0	0
1	0	0	0	0	0	0	I	0	0	0
1	0	1	0	0	0	0	0	I	0	0
1	1	0	0	0	0	0	0	0	I	0
1	1	1	0	0	0	0	0	0	0	I

◆ 利用上面之真值表，可得1×8解多工器之輸出Q_i的布林函數式如下：

$$Q_0 = I \cdot \overline{S}_2 \cdot \overline{S}_1 \cdot \overline{S}_0 \quad 、 \quad Q_1 = I \cdot \overline{S}_2 \cdot \overline{S}_1 \cdot S_0 \quad 、 \quad Q_2 = I \cdot \overline{S}_2 \cdot S_1 \cdot \overline{S}_0 \quad 、 \quad Q_3 = I \cdot \overline{S}_2 \cdot S_1 \cdot S_0$$

$$Q_4 = I \cdot S_2 \cdot \overline{S}_1 \cdot \overline{S}_0 \quad 、 \quad Q_5 = I \cdot S_2 \cdot \overline{S}_1 \cdot S_0 \quad 、 \quad Q_6 = I \cdot S_2 \cdot S_1 \cdot \overline{S}_0 \quad 、 \quad Q_7 = I \cdot S_2 \cdot S_1 \cdot S_0$$

◆ 使用邏輯閘來實現輸出Q_i之布林函數式，即可得1×8解多工器之邏輯電路，如下圖所示。



(b) 邏輯電路圖

◆ 觀察上圖可知，只要將解碼器之致能(EN)視為解多工器的輸入(I)，且將輸入線作為選擇線，則具致能輸入之解碼器與解多工器的電路完全相同。

使用解多工器實現布林函數

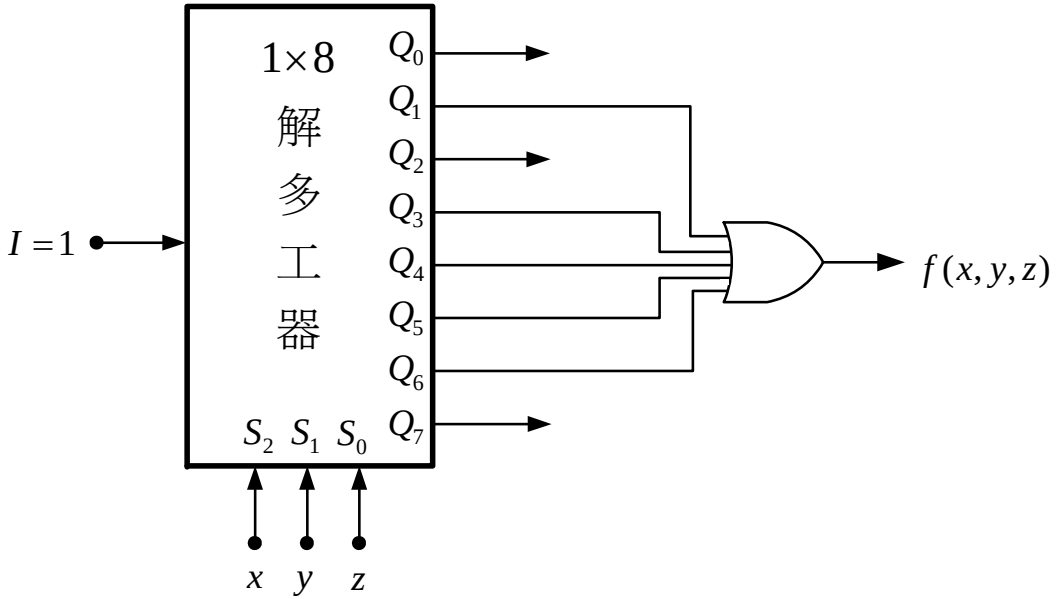
◆ 具有 n 條輸出線與選擇線等於 $k(k = \log_2 n)$ 條之解多工器而言，因 k 條選擇線可提供 n 個輸出線的完全解碼，故亦可視為一個通用邏輯模組 (University Logic Module)，因此只要將具完全解碼功能之解多工器，若將輸出線給予的適當連接，便可用來實現任何布林函數式。

例題 6-8

試利用具完全解碼功能之解多工器與一個 OR 閘來實現 $f(x, y, z) = x \cdot \bar{y} + \bar{x} \cdot z + x \cdot y \cdot \bar{z}$ 之布林函數式。

解

首先展開已知布林函數式為標準積項之和，即 $f(x, y, z) = x \cdot \bar{y} + \bar{x} \cdot z + x \cdot y \cdot \bar{z} = \sum(1, 3, 4, 5, 6)$。因 $f(x, y, z)$ 有 3 個輸入變數，故需使用 1×8 解多工器，首先令輸入 $I = 1$，因 $f(x, y, z)$ 在輸入組合為 1、3、4、5 與 6 時為邏輯 1，故可將布林函數 x、y 與 z 等 3 個輸入變數分別指定給解多工器之 S_2、S_1 與 S_0 等 3 條選擇線，且將 Q_1、Q_3、Q_4、Q_5 與 Q_6 等輸出線連接至 OR 閘之輸入端，即可實現 $f(x, y, z)$ 布林函數，如下圖所示。



例題 6-9

試利用具完全解碼功能之解多工器與兩個 OR 閘來實現全加法器 (Full Adder)。

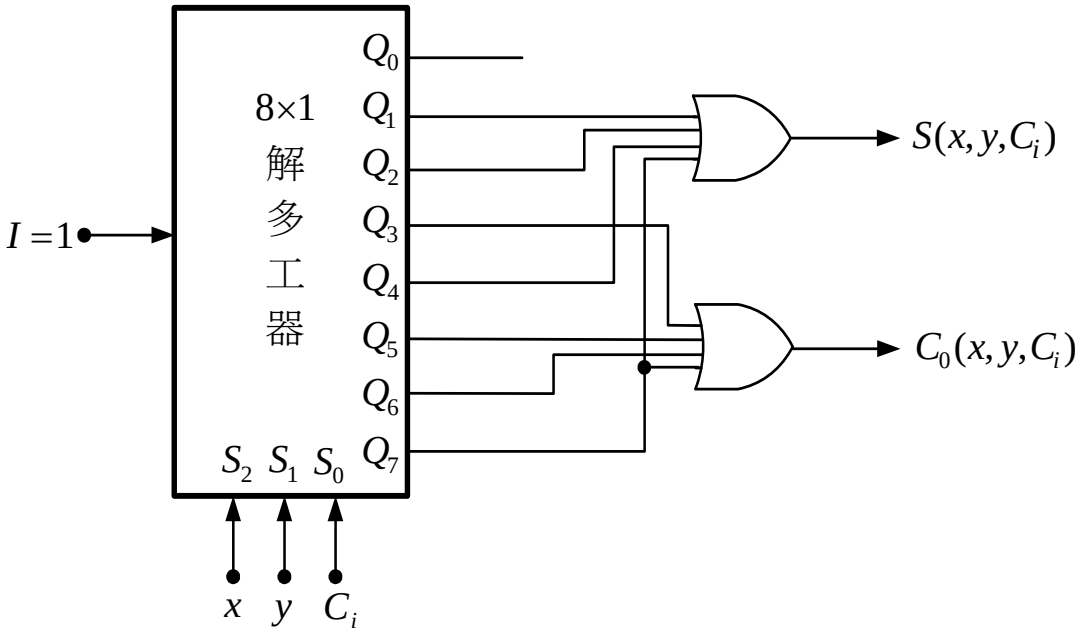
解

全加法器有 x、y 與 C_i 等 3 個輸入變數，故需用 1×8 解多工器，而有 S 與 C_o 兩個輸出變數，而可執行全加法器運算功能之布林函數式如下：

$$S(x, y, C_i) = \overline{x}\cdot\overline{y}\cdot C_i + \overline{x}\cdot y\cdot\overline{C_i} + x\cdot\overline{y}\cdot\overline{C_i} + x\cdot y\cdot C_i = \sum(1, 2, 4, 7)$$

$$C_o(x, y, C_i) = x\cdot y + x\cdot C_i + y\cdot C_i = \sum(3, 5, 6, 7)$$

首先令輸入$I=1$，且因$S(x, y, C_i)$在輸入組合為 1、2、4 與 7 時為邏輯 1，而$C_o(x, y, C_i)$在輸入組合為 3、5、6 與 7 時為邏輯 1，故可分別將解多工器之Q_1、Q_2、Q_4與Q_7等輸出線連接至一個 OR 閘之輸入端，接著再將Q_3、Q_5、Q_6與Q_7等輸出線連接至另外一個 OR 閘之輸入端，即可實現全加法器，如右圖所示。



問題討論

1、試簡述本章所討論之解碼器、編碼器、多工器與解多工器等 4 種資料處理電路之主要邏輯運算功能。

解：

(a)、解碼器：將某一數碼轉換成其所對應值至單一輸出端的過程，而可將*n*個位元之數碼轉換成最多*m*個輸出中一個之組合邏輯電路稱為 $n \times m$ 解碼器(Decoder)。

(b)、編碼器：將單一個輸入對應至一組被編碼的輸出組合，即將*n*個輸入線轉變成為*m* 個 $(m = \lceil \log_2 n \rceil)$ 輸出組合之二進位碼的組合邏輯電路，稱為 $n \times m$ 編碼器(Encoder)。

(c)、多工器：將數個資料輸入 I_i，而僅選擇其中一個輸入資料送至單一輸出線*Y*，而要指定輸入資料 I_i 傳送至輸出端*Y*，可由*k*條($k = \lceil \log_2 n \rceil$)選擇線之二進位組合的等效十進位數*i*來決定。

(d)、解多工器：將*n*個輸出線以十進位 Q_i 來表示，而需用 $k = \lceil \log_2 n \rceil$ 選擇線，以二進位組合形式，以指定唯一輸入*I*之資料至其中一輸出線 Q_i。

2、試詳細說明在設計數位系統時，大部分皆採用低態致動(低電位輸出)解碼器之優點。

解：

(1)、組成『低態致動』輸出之解碼器的邏輯閘為 NAND 閘，而『高態致動』輸出為 AND 閘。因使用積體電路來製造 NAND 閘比 AND 閘，所需之電晶體(Transistor)數目較少，故採用「低態致動」輸出之解碼器可減少積體電路所需之成本。

(2)、因『低態致動』輸出之解碼器，各個輸入組合所對應之輸出線的電壓為邏輯 0(低電位)，即輸出僅提供一個接地點(邏輯 0)，如圖(b)所示，故驅動負載所需之電流是由電源供給，而不是由解碼器輸出端提供，因此可用驅動較大之負載；但『高態致動』輸出之解碼器，各個輸入組合所對應之輸出線之電壓為邏輯 1(高電位)，即輸出僅提供一個 V_{CC} (邏輯 1)，如圖(a)所示，故驅動負載所需之電流需由解碼器之輸出端提供，因 IC 所能提供之輸出電流相當有限，故此種解碼器能驅動之負載相對較小。

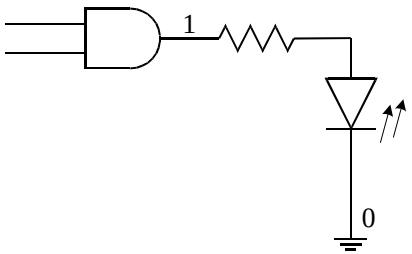


圖 (a)

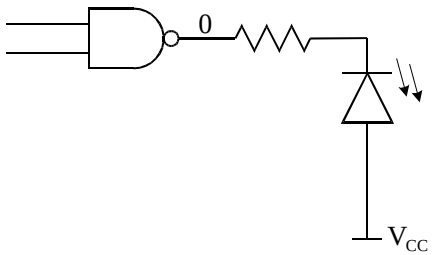
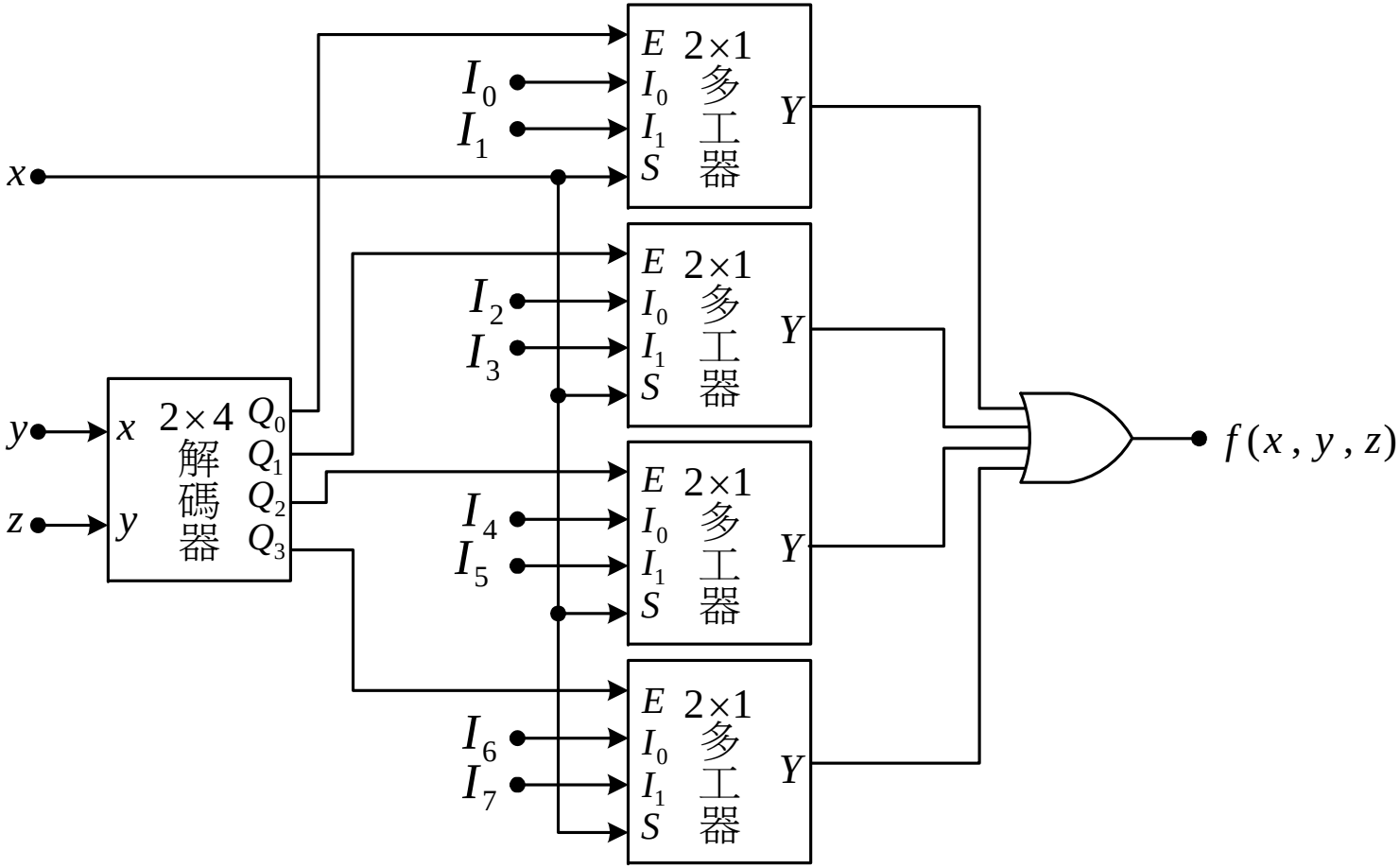


圖 (b)

3、試利用 4 個具致能輸入之 2 對 1 線多工器與 1 個 2 對 4 線解碼器來組成一個 8×1 多工器。

解：



4、試設計一個 1×4 解多工器，並說明所設計之電路，如何轉換為可執行 2×4 解碼器之邏輯運算。

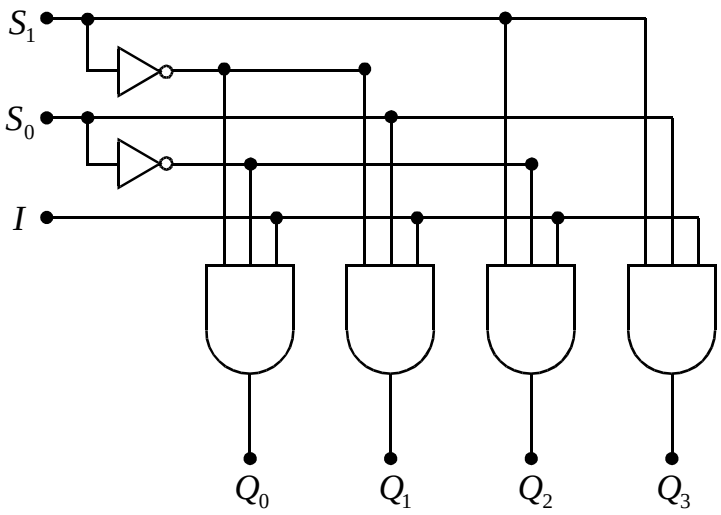
解：

1 對 4 線解多工器僅有一個輸入線 I，Q_3, Q_2, Q_1 與 Q_0 等 4 個輸出變數與 S_1 與 S_0 等 3 條 $k=\lceil \log_2 4 \rceil = 2$ 選擇線之組合邏輯電路。根據解多工器之定義（即利用選擇線所組成二進位之等效十進位數 i，以指定輸入 I 傳送至輸出 Q_i），可得 1×4 解多工器之 Q_i 的布林函數式如下：

$$Q_0 = I\cdot\overline{S}_1\cdot\overline{S}_0 \quad 、、 \quad Q_2 = I\cdot S_1\cdot\overline{S}_0 \quad 、 Q_3 = I\cdot S_1\cdot S_0$$

使用邏輯閘來實現輸出 Q_i 之布林函數式，即可得 1×4 解多工器之邏輯電路，如下圖所示。

觀察右圖可知，只要將解碼器之致能 (E) 視為解多工器的輸入 (I)，且將輸入線 $(x$ 與 $y)$ 作為選擇線 $(S_1$ 與 $S_0)$，則具致能輸入之解碼器與解多工器的電路完全相同。



5、試利用具致能輸入之 3×8 解碼器與 2×4 解碼器來組合出 5×32 解碼器。

解：