

Contents

1

chapter 1-1

數位邏輯

第一節 布林代數與邏輯閘 • 1-1

第二節 邏輯簡化 • 1-8

第三節 組合電路 • 1-18

第四節 循序電路 • 1-27

2

chapter 2-1

計算機組織 與結構

第一節 計算機硬體簡介 • 2-1

第二節 微處理器 • 2-19

第三節 記憶體實作 • 2-41

第四節 週邊設備與安寶定律 • 2-61

3

chapter 3-1

資料表示法

第一節 數字系統 • 3-1

第二節 數字資料表示法 • 3-9

第三節 文字與多媒體資料 • 3-25

第四節 錯誤的偵測與修正 • 3-48

4

chapter 4-1

程式語言

第一節 軟體類型與智慧財產權 • 4-1

第二節 程式語言簡介 • 4-10

第三節 結構化程式設計 • 4-30

第四節 常見資料型態與運算子 • 4-47

第五節 巨集置換與函式呼叫 • 4-68

第六節 物件導向與程序導向 • 4-90

Contents

5

chapter 5-1

作業系統

第一節 作業系統簡介 • 5-1

第二節 行程與執行緒 • 5-24

第三節 同步與死結 • 5-44

第四節 記憶體管理 • 5-52

第五節 檔案組織與檔案配置 • 5-68

6

chapter 6-1

資料庫應用

第一節 資訊系統開發概論 • 6-1

第二節 資料庫簡介 • 6-12

第三節 實體關聯模型與資料庫綱要 • 6-23

第四節 資料庫正規化 • 6-30

第五節 關聯代數與 SQL 語法 • 6-38

7

chapter 7-1

資料結構與 演算法

第一節 演算法的分析與設計 • 7-1

第二節 基本資料結構 • 7-18

第三節 樹狀結構 • 7-33

第四節 圖形理論 • 7-52

第五節 搜尋演算法 • 7-74

第六節 排序演算法 • 7-91

8

chapter 8-1

電腦網路

第一節 網路概論 • 8-1

第二節 應用層與傳輸層 • 8-18

第三節 網路層與連結層 • 8-36

第四節 區域網路與上網方式 • 8-58

Contents

9

chapter 9-1

資訊安全與 網路應用	第一節 資訊安全概論 • 9-1
	第二節 基礎密碼學 • 9-19
	第三節 加密法應用 • 9-30
	第四節 網路應用與資訊化社會 • 9-40

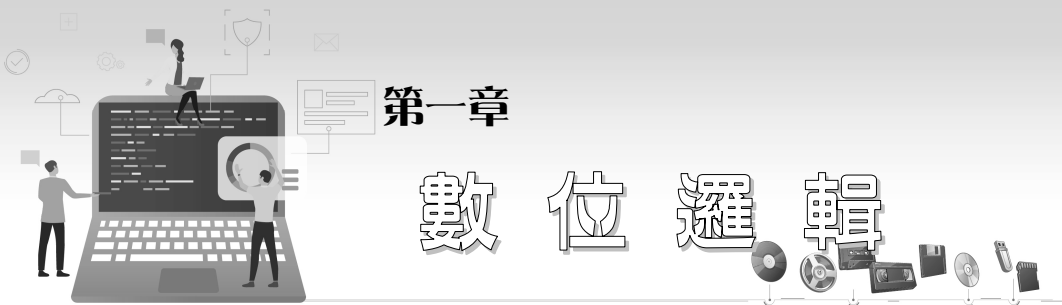
更多~

精選試題

99~106年試題

貼心為您準備





第一章

數位邏輯



布林代數與邏輯閘



布林代數 (Boolean Algebra)

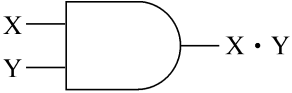
(一)英國數學家 George Boole 提出：

以代數符號處理邏輯運算的一組結構與規則，又稱符號邏輯 (Symbolic Logic)。

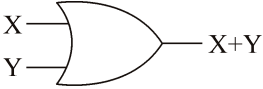
(二)常見運算子：

AND、OR、NOT 為基本運算子，可實作所有布林函數。

1. AND：及運算，又稱乘法運算，符號 $X \cdot Y$ 或 $X * Y$ 。

真值表			補充說明
X	Y	$X \cdot Y$	※二輸入同為 1，輸出才為 1 ※邏輯閘符號，如下圖： <div style="text-align: center;">  <p>AND Gate</p> </div>
0	0	0	
0	1	0	
1	0	0	
1	1	1	
AND，有 0 則 0			

2. OR：或運算，又稱加法運算，符號 $X + Y$ 。

真值表			補充說明
X	Y	$X + Y$	※一輸入為 1，輸出即為 1 ※邏輯閘符號，如下圖： <div style="text-align: center;">  <p>OR Gate</p> </div>
0	0	0	
0	1	1	
1	0	1	
1	1	1	
OR，有 1 則 1			



3. NOT：反運算，符號 X' 或 \bar{X} 。

真值表		補充說明						
<table border="1"> <tr> <th>X</th> <th>X'</th> </tr> <tr> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> </tr> </table>		X	X'	0	1	1	0	※1 變 0；0 變 1 ※邏輯閘符號，如下圖：
X	X'							
0	1							
1	0							
		<p>NOT Gate</p>						

4. NAND：反及運算，符號 $(X \cdot Y)'$ 或 $(X' + Y)'$ 。

真值表				補充說明
X	Y	$X \cdot Y$	$(X \cdot Y)'$	※先求 AND，再求 NOT ※邏輯閘符號，如下圖：
0	0	0	1	<p>NAND Gate</p>
0	1	0	1	
1	0	0	1	
1	1	1	0	
NAND，有 0 則 1				

5. NOR：反或運算，符號 $(X + Y)'$ 或 $(X' \cdot Y)'$ 。

真值表				補充說明
X	Y	$X + Y$	$(X + Y)'$	※先求 OR，再求 NOT ※邏輯閘符號，如下圖：
0	0	0	1	<p>NOR Gate</p>
0	1	1	0	
1	0	1	0	
1	1	1	0	
NOR，有 1 則 0				

6. XOR (eXclusive OR)：互斥運算，符號 $X \oplus Y$ 或 $X'Y + XY'$ 。

真值表			補充說明
X	Y	$X \oplus Y$	※二輸入不同 (互斥)，輸出為 1 ※邏輯閘符號，如下圖：
0	0	0	<p>XOR Gate</p>
0	1	1	
1	0	1	
1	1	0	
兩個不同，為互斥			



7. XNOR (eXclusive NOR)：反互斥（相等）運算，符號 $X \odot Y$ 或 $X'Y' + XY$ 。

真值表			補充說明
X	Y	$X \odot Y$	※（互斥後求反運算） ※邏輯閘符號，如下圖：
0	0	1	
0	1	0	
1	0	0	
1	1	1	
兩個相同，反互斥			<p>XNOR Gate</p>

(三) 衍生應用：

1. 近代電晶體元件符合二元變數特性，可實作布林代數基本運算子。
2. 以布林代數堆積各種算術邏輯運算，為電腦系統的重要基礎。

重點 2

萬用邏輯閘 (Universal Logic Gate)

(一) NAND/NOR Gate：

可模擬三種基本邏輯閘，故稱萬用邏輯閘。

(二) 等效電路 321：

跨類 3；同類 2；NOT 都是 1，自己接自己。

<p>NAND 模擬 AND：同類 2 二個 NAND 模擬一個 AND</p> <p>$[(X \cdot Y)' \cdot (X \cdot Y)']'$ $= [(X \cdot Y)']' = X \cdot Y$</p>	<p>NOR 模擬 OR：同類 2 二個 NOR 模擬一個 OR</p> <p>$[(X + Y)' + (X + Y)']'$ $= [(X + Y)']' = X + Y$</p>
<p>NAND 模擬 OR：跨類 3 三個 NAND 模擬一個 OR</p> <p>$(X' \cdot Y)'$ $= (X')' + (Y)' = X + Y$</p>	<p>NOR 模擬 AND：跨類 3 三個 NOR 模擬一個 AND</p> <p>$(X' + Y)'$ $= (X')' \cdot (Y)' = X \cdot Y$</p>
<p>NOT 都是 1，自己接自己 一個 NAND 模擬一個 NOT</p> <p>$(X \cdot X)' = X'$</p>	<p>NOT 都是 1，自己接自己 一個 NOR 模擬一個 NOT</p> <p>$(X + X)' = X'$</p>



(三) 等效電路：

1. 笛摩根定律 (DeMorgan's Law)：先→後，AND→OR。

(1) $(X \cdot Y)' = X' + Y'$ (先·後 Prime ≡ 先 Prime 後+)。

(2) $(X + Y)' = X' \cdot Y'$ (先+後 Prime ≡ 先 Prime 後·)。

2. 等效電路：對偶性 (AND 換 NOR | OR 換 NAND)，3 皇 (反) 3 家 (加)，加三個反運算，以「。」符號表示。

(1) AND/NOR 等效電路：

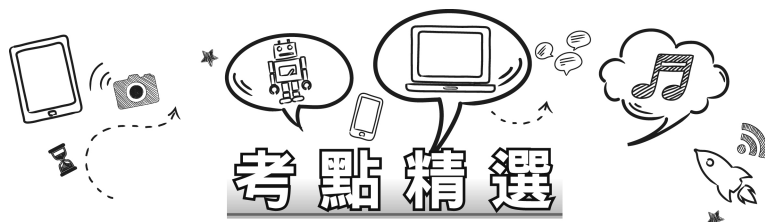
$\therefore X \cdot Y = (X + Y)'$ (DeMorgan's Law)



(2) OR/NAND 等效電路：

$\therefore (X + Y) = (X' \cdot Y)'$ (DeMorgan's Law)





問答題

●某一作業系統有一特徵，一臺中毒的電腦會回傳錯誤答案給任何從網路上收到的問題，如它（中毒的電腦）被問到「你被感染到病毒了嗎」，它會回傳「沒有」。而一臺沒有中毒的電腦總是會正確回答，若被問到「你被感染到病毒了嗎」，它的答案是「沒有」。請問以下那一個句子，是只有中毒的伺服器才會回傳的訊息，請同時說明理由。

- (A)我是一臺中毒的伺服器。
- (B)我不是一臺中毒的伺服器。
- (C)我是一臺中毒的筆記型電腦。
- (D)我不是一臺中毒的筆記型電腦。

【104 關務三】

擬答：中毒電腦回傳錯誤答案，無毒電腦回傳正確答案；依電腦類型（伺服器、筆記型）與中毒與否，可能回傳如下表。

	中毒	無毒
伺服器	(B), (C), (D)	(B)
筆記型	(A), (B), (D)	(D)

- (1)中毒伺服器回傳錯誤答案，可能訊息(B)(C)(D)；但無毒伺服器回傳(B)（正確答案），中毒筆記型可能回傳(D)（錯誤答案），故只有(C)（我是一臺中毒的筆記型電腦）是中毒伺服器才會回傳的訊息。
- (2)中毒筆記型回傳錯誤答案，可能訊息(A)(B)(D)；但無毒筆記型回傳(D)（正確答案），中毒伺服器可能回傳(B)（錯誤答案），故只有(A)（我是一臺中毒的伺服器電腦）是中毒筆記型才會回傳的訊息。

選擇題

國考類

1. 下列何種邏輯閘，可以用來建構出其他所有的邏輯閘？

(C)



【107.鐵員】

9. 將兩個二進位數字 11110000 與 00000111 經邏輯運算子 AND 運算，其運算後之結果為下列何者？ (C)
- (A) 11110000 (B) 00000111 (C) 00000000 (D) 11110111

【107.鐵員】

國營事業類

1. 0100 和 1100 邏輯運算後的結果是 1011，請問運算子為下列何者？ (D)
- (A) AND (B) NOR (C) XOR (D) NAND

【107.臺電】

2. 下列那一個運算式的執行結果與其他三個不同？ (A)
- (A) NOT(18 > 17) (B) (12 <= 0) OR (220 > 120)
- (C) (18 > 14) AND (220 > 120) (D) (12 <= 0) XOR (220 > 120)

【107.漢翔】

3. 假設晶圓製作的良率為 50%，封裝的良率為 30%，則合格的處理器不超過幾%？ (C)
- (A) 80% (B) 20% (C) 15% (D) 10% 【107.中鋼】



1-2 邏輯簡化 (Logic Minimization)



代數式推導

(一)恆等式：

二類型，符合對偶性 (Duality, AND/OR 互換, 0/1 互調)。

名稱	AND 類型	OR 類型
同一律	$1 \cdot X = X$	$0 + X = X$
無效律	$0 \cdot X = 0$	$1 + X = 1$
冪等律	$X \cdot X = X$	$X + X = X$
逆定律	$X \cdot X' = 0$	$X + X' = 1$
交換律	$X \cdot Y = Y \cdot X$	$X + Y = Y + X$
結合律	$(X \cdot Y) \cdot Z = X \cdot (Y \cdot Z)$	$(X + Y) + Z = X + (Y + Z)$
分配律	$X + (Y \cdot Z)$ $= (X + Y) \cdot (X + Z)$	$X \cdot (Y + Z)$ $= (X \cdot Y) + (X \cdot Z)$
吸收律	$X \cdot (X + Y) = X$	$X + (X \cdot Y) = X$
笛摩根	$(X \cdot Y)' = X' + Y'$	$(X + Y)' = X' \cdot Y'$
雙重互補	$(X')' = X$	

(二)布林代數化簡重點：

1. 笛摩根定律 (DeMorgan's Law)：先→後，AND→OR。

2. XOR 互斥運算： $X \oplus 0 = X$, $X \oplus 1 = X'$ 。

3. 常考： $X + (X' \cdot Y) = X + Y$ ，需熟記。

(1) $X + (X' \cdot Y) = (X + X') \cdot (X + Y) = 1 \cdot (X + Y) = X + Y$ 。

(2) 想法：函數何時輸出 1？ $X = 1$ 或 $X = 0$ 時 $Y = 1$ ，得 $X + Y$ 。

4. 證明兩布林代數式相等：能推則推，不能（會）推，列真值表，所有狀況輸出相等，兩式相等，稱邏輯等價 (Logical Equivalent)。

(三)遮罩運算 (Masking)：

實例如下，利用 AND 運算的同一律 ($1 \cdot X = X$) 與無效律 ($0 \cdot X = 0$)，可將二進值與特定的位元樣式 (Bit Pattern) 做 AND 運算，以求得所需運算結果。



1. 求 MOD 8 : 「8 bits 二進值 · 00000111」可得計算結果；如
 $(35)_{10} \% (8)_{10} = (00100 - 011)_2 \cdot (00000 - 111)_2 = (00000 - 011)_2$
 $= (3)_{10}$ 。
2. 將 ASCII 碼小寫字母轉大寫字母：ASCII 碼（大寫字母） $_{10} =$ （小寫字母） $_{10} - 32$ ，故「小寫字母二進值 · 11011111」可得轉換結果；如
 $ASC(a) \cdot 11011111 = (97)_{10} \cdot 11011111 = (01-1-00001) \cdot (11-0-11111) = (01-0-00001)_2 = (65)_{10} = ASC(A)$ 。



SOP/POS 標準形式

(一) 最小項 (Miniterms) 與最大項 (Maxiterms) :

1. 最小項：取二元變數相乘為 1，變數值 0 時取 Prime，小 m 表示。
2. 最大項：取二元變數相加為 0，變數值 1 時取 Prime，大 M 表示。
3. 以三個二元變數 X、Y、Z 為例，最小項與最大項表列如下：

X	Y	Z	最小項	符號	最大項	符號
0	0	0	$\bar{X} \bar{Y} \bar{Z}$	m0	$X + Y + Z$	M0
0	0	1	$\bar{X} \bar{Y} Z$	m1	$X + Y + \bar{Z}$	M1
0	1	0	$\bar{X} Y \bar{Z}$	m2	$X + \bar{Y} + Z$	M2
0	1	1	$\bar{X} Y Z$	m3	$X + \bar{Y} + \bar{Z}$	M3
1	0	0	$X \bar{Y} \bar{Z}$	m4	$\bar{X} + Y + Z$	M4
1	0	1	$X \bar{Y} Z$	m5	$\bar{X} + Y + \bar{Z}$	M5
1	1	0	$XY \bar{Z}$	m6	$\bar{X} + \bar{Y} + Z$	M6
1	1	1	XYZ	m7	$\bar{X} + \bar{Y} + \bar{Z}$	M7

(二) 函數表示法：

已知布林函數真值表如下，二種表示法。

X	Y	Z	F(X, Y, Z)	最小項(m)	最大項(M)
0	0	0	0	m0	M0
0	0	1	1	m1	M1
0	1	0	1	m2	M2
0	1	1	0	m3	M3



1-10

計算機概論 (概要)



1	0	0	0	m4	M4
1	0	1	1	m5	M5
1	1	0	1	m6	M6
1	1	1	0	m7	M7

1. 最小項之和 (Sum of Miniterms) : 從 F 值為 1 出發, 找最小項求和 (+ 運算, 一個為 1, 輸出定為 1) ; $F(X, Y, Z) = m1 + m2 + m5 + m6 = \sum m(1, 2, 5, 6) = X'Y'Z + X'YZ' + XY'Z + XYZ'$ 。
2. 最大項之積 (Product of Maxiterms) : 從 F 值為 0 出發, 找最大項求積 (\cdot 運算, 一個為 0, 輸出定為 0) ; $F(X, Y, Z) = M0 \cdot M3 \cdot M4 \cdot M7 = \prod M(0, 3, 4, 7) = (X + Y + Z) \cdot (X + Y' + Z') \cdot (X' + Y + Z) \cdot (X' + Y' + Z')$ 。



函數化簡—卡諾圖 (Karnaugh Map, Kmap)

(一) 最小項之和, 化積之和 (Sum Of Product, SOP) :

化簡步驟, 四大項。

1. 由 F (Prime 為 0, 其餘 1), 得 $\sum m$, 轉 2 進值。
2. 畫卡諾圖, 填 0、1 (函數值), 注意順序 (00、01、11、10)。
3. 圈選相鄰 1: 先四後兩; 上下左右、邊界循環 (可四邊無對角)。
4. 看圖說故事 (0 取 Prime; 變數相乘「 \cdot 」得項; 各項相加「+」得函數式), 列最簡函數 (去除主要項, 只留必要項, 得 SOP)。

(二) 兩變數卡諾圖:

化簡 $F(A, B) = A'B + A'B'$ 。

	B	0	1	※由 F, 得 $\sum m(01, 00)$ ※畫卡諾圖, 填 0、1 (函數值) ※圈選相鄰 1: 相鄰 1 表可化簡, 稱項 (Implicant) ※由圖可知, A 為 0, F 值定為 1, 與 B 無關, $F(A, B) = A'$
A		0	1	
	0	1	1	
	1	0	0	

(三) 三變數:

化簡 $F(A, B, C) = A'B'C + A'BC + AB'C + ABC + ABC'$ 。