布氏代數的基本定理與化簡法

2-1 簡介

- 布林代數(Boolean algebra)- 數位系統的邏輯設計所需的基本工具。(成為所有數位系統電路的數學基礎)
- A的反相(inverse)元素或補數的兩個常用符號是 $\overline{A}$ 和A’，A’適合於打字機、列表機和計算機使用
- 大多數工程師以+代表OR，以·（或者不使用任何符號）代表AND，數學家常用的表示法是以V代表OR，以Λ代表AND

- 本單元探討的是布林代數的一個特殊狀況−所有的變數只能是兩種數值中的一種，這種雙值式的布氏代數通常稱作交換代數(Switching Algebra)
- 0或1不具有數值，而只是兩個符號，用來表示交換變數的兩種值。 EX: 0→低電壓 1→高電壓
- 一般來說，若電壓值有兩種，則正和負邏輯的定義是：

  正邏輯 − 兩種電壓值中的較高者，代表邏輯值1，較低者代表邏輯值0。

  負邏輯 − 反之。

- 運算子的優先順序：()，。＋
- 運算式中的變數或其補數稱作文字(Literal)

EX: $ab'c + a'b + a'bc' + b'c'$

有3個變數，10個數字

2-2 布氏代數

- 三個重要定理:
  1. 第摩根定理(DeMorgan's Theorem);
     布氏運算式內積的補數 = 補數的和
     和的補數 = 補數的積

     Ex: $(A \cdot B \cdot C)' = A'+B'+C'$
     $(A+B+C)' = A' \cdot B' \cdot C'$

2. 對偶定理(Dual Theorem)
在布林代數中，若將運算子＋與・互換，並將單位元素0與1互換，則所得之結論亦成立，

如：

$$X + XY \leftrightarrow X(X + Y)$$

$$XY + XY \leftrightarrow (X + Y)(X + Y)$$

- 求對偶式 ：

1.利用對偶定理

ex: $F = ab' + c + 0 \cdot d'(1 + e)$

則對偶式是 $F^D=(a+b') \cdot C \cdot (1+d'+0 \cdot e)$

2.利用第摩根定理

先求運算式的補數，然後將各變數改成其補數

ex: $F = ab'+ c + 0 \cdot d'(1+e)$

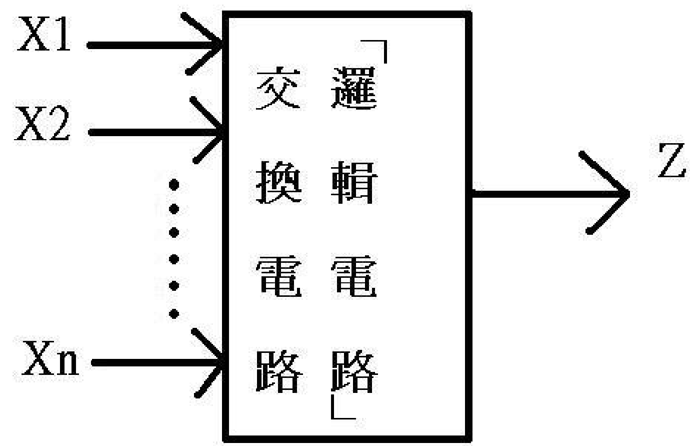
$F'= (a'+ b)(c')(1 + d' + 0 \cdot e')$

$=> F^D= (a+b') \cdot c \cdot (1+d'+0 \cdot e)$

- 對偶式應用：

如：原本變數使用正邏輯欲改成使用負邏輯

Sol：求其對偶式



註1: XOR運算式的對偶式

$$(X \oplus Y)^D = (X \equiv Y) \text{ or } (X \odot Y)$$

註2：交換函數f的補數函數求法有三

(1)真值表

(2) DeMorgan’s Low

(3) $f \rightarrow f^D$ 再將各交換變數取其補數。

3.分配律(Distributive Law)

A・(B + C) = (A・B) + (A・C) (又稱乘對加分配律)

A + (B・C) = (A + B)・(A + C) (又稱加對乘分配律)

例子: proof xy’+ y = x + y

由加對乘分配律

原式(y + x)・(y + y’) = x + y

pb.

$$A+\overline{A}B=A+B$$

$$\overline{A}+A\overline{B}=\overline{A}+\overline{B}$$

$$B+A\overline{B}=A+B$$

$$\overline{B}+\overline{A}B=\overline{A}+\overline{B}$$

$$A+\overline{A}BC=A+BC$$

2-3 布氏函數的標準形式

- 求一個真值表所對應的交換函數的代數表示有兩種做法

  (1)為真值表中，函數值為1的所有最小項的和項

  (2)為真值表中，函數值為0的所有最大項的乘積項

  (最小項或稱全及向，最大項或稱全或項)

例：

十進制	x y z	f	最小項	最大項
0	0 0 0	1	$X'Y'Z' = m_0$	$X+Y+Z = M_0$
1	0 0 1	0	$X'Y'Z = m_1$	$X+Y+Z' = M_1$
2	0 1 0	0	$X'YZ' = m_2$	$X+Y'+Z = M_2$
3	0 1 1	1	$X'YZ = m_3$	$X+Y'+Z' = M_3$
4	1 0 0	0	$XY'Z' = m_4$	$X'+Y+Z = M_4$
5	1 0 1	1	$XY'Z = m_5$	$X'+Y+Z' = M_5$
6	1 1 0	1	$XYZ' = m_6$	$X'+Y'+Z = M_6$
7	1 1 1	0	$XYZ = m_7$	$X'+Y'+Z' = M_7$

- $f_1 = x'y'z' + x'yz + xy'z + xyz'$

  此全及項和稱全及項展開式或稱標準的積項和
  (standard sum of products) or
  (canonical sum of products) or
  (disjunctive normal form) or
  (SOP)

  尚可寫成 $f_1 = m_0 + m_3 + m_5 + m_6$ (簡潔方式連記法)
  $= \sum_m (0, 3, 5, 6)$

- $f_2=(x + y + z')\ (x + y' + z)\ (x' + y + z)\ (x' + y' + z')$

  此全或項乘積稱為全或項展開式，也稱作標準的和項積
  (standard product of sums) or
  (canonical products of sums) or
  (conjunctive normal forms) or
  (POS)

亦可寫成 $f_2 = M_1 M_2 M_4 M_7$
$= \prod_m(1, 2, 4, 7)$

其中 $f_1 = f_2$

註1: 最大項(全或項)是其對應(全及項)最小項的補數，即$M_i=m'_i$

註2: 任何最小項只有在它所對應的二進制組合之下，它的值才為1，(在其他的組合下，它的值均為0);而任何最大項只有在它所對應的二進制組合下，它的值才為0（在其他的組合下，它的值均為1)。

2-4 交換函數性質

- 列出兩個交換變數的所有交換函數

  以標準SOP型式為例

  則 $f(x, y) = X_0 X'Y' + X_1 X'Y + X_2 XY' + X_3 XY$

  由於係數 $X_3 X_2 X_1 X_0$ 共有16種不同的組合 因此共有16種不同的交換函數。

  亦即 2^{2^n} 個組合 (對 n 個交換變數而言)

兩個交換變數的所有交換函數

Boolean functions　　Name　　Operator symbol

$X_3X_2X_1X_0$	交換函式f (x, y)	函數名稱	運算子符號
0 0 0 0	0	Null(零函數常數0)	
0 0 0 1	$x'y' = (x + y)'$	NOR(反或)	$X \downarrow Y$
0 0 1 0	$x'y$	Inhibition(禁止)	Y/X
0 0 1 1	x'	Complement	X'
0 1 0 0	xy'	Inhibition	X/Y
0 1 0 1	y'	Complement	Y'
0 1 1 0	$x'y + xy'$	Exclusive-OR (XOR，互斥或)	$X \oplus Y$
0 1 1 1	$x' + y' = (xy)'$	NAND	$X \uparrow Y$

$X_3X_2X_1X_0$	交換函式f (x ,y)	函數名稱	運算子符號
1 0 0 0	xy	AND(及)	x ‧ y
1 0 0 1	x'y+x'y'	Equivalent(相當) (XNOR)	x⊙y or x≡y
1 0 1 0	y	Transfer(轉移)	
1 0 1 1	x'+y	Implication(意含)	x⊃y or x→y
1 1 0 0	x	Transfer	x
1 1 0 1	x+y'	Implication	x⊂y or y→x
1 1 1 0	x+y	OR(或)	x+y
1 1 1 1	1	Identity(單位函數)	

2-5 布氏函數的化簡

- 交換函數化簡的基本概念:
  即交換函數最好能以最少數目的變數與乘積項(或和項)表示，以減少使用邏輯閘數目和邏輯閘的扇入數目。
  pb.布氏函數化簡法有:(並做簡單比較)
  1.代數化簡法
  有兩個困難的地方
  ①代數技巧無法系統化地應用
  ②很難判別出是否已經化簡到最簡的形式

2.卡諾圖化簡法

優點 ①克服代數化簡法之困難，即快速而且容易

缺點 ②只應用在6變數以下的場合

3.列表法(Tabulation method)

最早用於Quine所發表，而後由McCluskey修正，固此又稱Quine - McCluskey。(昆麥克萊斯)法

優點：

①能夠有系統的化簡多變數函數(含6個以上)

②容易寫成數位計算機程式

缺點:

演算過程冗長而且單調，令人厭倦，因此反而容易造成錯誤。

其他還有許多化簡方法可以在文獻中找到，但是其中大部分都是卡諾圖法或者昆麥克勞斯法的變化法或改良法。EX:變數引入圖法

pb. 試利用布林代數(Boolean Algebra)規則 化簡下列FORTRAN邏輯指述(Logical Statement):

LV=.NOT.((.NOT.L.OR.(B.LT.C)).AND.(I.EQ.J))

又設L=.FALSE. , B=10.5 , C=8.3 , I=5 , J=5，則該邏輯指述LV值為何?

sol: LV = ((L’+ (B<C))．(I = J))’

= ((TRUE + FALSE)．(TRUE))’

= (TRUE．FRUE)’

= (TRUE)’ = FALSE

<u>∴LV之值為FALSE</u>

pb.試化簡 $x_1 + x_1' x_2 + x_1' x_2' x_3 + x_4$ 布林代數 (Boolen Algebra)

(提示:利用加對乘分配率)

Ans: $x_1 + x_2 + x_3 + x_4$