CAPITOLO 3

ALGEBRA BOOLEANA

3 INTRODUZIONE

Nei capitoli 1 e 2 abbiamo visto l'uso di variabili binarie come elementi di informazione e come è possibile attraverso di esse, codificare informazioni più complesse. Abbiamo anche introdotto, procedendo empiricamente, degli operatori che operando su due variabili binarie ci forniscono il valore di una terza. Abbiamo altresì accennato che prima di addentrarci oltre e cioè: operare su più di due variabili, era meglio rivolgersi ad una derivazione teorica e formale per trovare come operare con espressioni che contengono più di uno degli operatori individuati. Abbiamo anche accennato che George Boole nel 1854 introdusse una trattazione sistematica dei problemi di logica e ha sviluppato, per questo scopo, un sistema algebrico ora chiamato *Algebra Booleana* in suo onore.
Noi procederemo prima ad una sua definizione assiomatica e, poiché può essere fatta in diversi modi, seguiremo quella data da E. V. Huntington nel 1904. Quindi procederemo a definire l'algebra Booleana su l'insieme di due soli elementi distinti 0 ed 1 e deriveremo i teoremi che ci saranno utili e necessari nel prosieguo.

3.1 Definizione assiomatica dell'algebra Booleana

L'algebra Booleana è una struttura algebrica definita da un insieme di elementi *B* su cui operano due operatori + o somma e • o prodotto che soddisfano i seguenti postulati:

1a) Chiusura rispetto all'operatore +, cioè se $x, y \in B \Rightarrow z = x + y \in B$;	1b) Chiusura rispetto all'operatore •, cioè se $x, y \in B \Rightarrow z = x \cdot y \in B$;
2a) Esistenza dell'elemento identità rispetto a +, indicato con 0: $x + 0 = 0 + x = x, \forall x \in B$;	2b) Esistenza dell'elemento identità rispetto al prodotto •, indicato con 1: $x \cdot 1 = 1 \cdot x = x, \forall x \in B$;
3a) Vale la legge commutativa per l'oper. +: $x + y = y + x, \forall x, y \in B$;	3b) Vale la legge commutativa per l'op. • : $x \cdot y = y \cdot x, \forall x, y \in B$;
4a) Vale la legge distributiva di · su +: $x \cdot (y + z) = (x \cdot y) + (x \cdot z), \forall x, y, z \in B$;	4b) Vale la legge distributiva di + su • : $x + (y \cdot z) = (x + y) \cdot (x + z), \forall x, y, z \in B$;
5a) $\forall x \in B$ esiste un $x' \in B$ (chiamato complemento di x) tale che: $x + x' = 1$;	5b) $\forall x \in B$ esiste un $x' \in B$ (chiamato complemento di x) tale che: $x \cdot x' = 0$;
6a) Esistono almeno due elementi $x, y \in B$ con $x \neq y$	6b) Esistono almeno due elementi $x, y \in B$ con $x \neq y$

Se paragoniamo L'algebra Booleana con quella ordinaria (nel campo dei numeri reali) notiamo le seguenti differenze:

(a) I postulati non includono la legge associativa. Questa sarà derivata dagli altri postulati.

(b) La legge distributiva di + su · non è valida per l'algebra ordinaria.

(c) L'algebra Booleana non ha inversi degli operatori somma e prodotto e quindi non esistono né la sottrazione né la divisione

(d) Il postulato 5 definisce un operatore chiamato *complemento* che non esiste nell'algebra ordinaria

(e) L'algebra ordinaria tratta con in numeri reali, che costituiscono un insieme infinito di elementi. L'algebra Booleana tratta con un insieme di elementi non definito, ma nell'algebra a due valori definita più avanti (e di nostro interesse nel futuro uso di questa algebra), B è definito come un insieme che contiene soltanto due elementi distinti 0 ed 1.

Per alcuni aspetti l'algebra Booleana assomiglia a quella ordinaria. La scelta dei simbolo + e · è fatta intenzionalmente per facilitare le manipolazioni algebriche alle persone già familiari con quell'algebra. Sebbene spesso vengono usate le conoscenze dell'algebra ordinaria il principiante stia attento a non sostituire le regole dell'algebra ordinaria a quella Booleana dove quelle non possono essere applicate.

É importante distinguere tra elementi di un insieme di una struttura algebrica e le variabili del sistema algebrico. Per es., gli elementi del campo dei numeri reali sono numeri, mentre le variabili come a, b, c, etc., usati nell'algebra ordinaria, sono simboli che stanno al posto dei numeri reali. Allo stesso modo, nell'algebra Booleana, si definiscono gli elementi dell'insieme B e le variabili x, y, z sono solo simboli che rappresentano gli elementi. A questo punto è importante realizzare che per avere un'algebra Booleana uno deve dimostrare che sia gli

elementi dell'insieme B che le regole di operazione per i due operatori binari (somma e prodotto) soddisfano i 6 postulati:
Si possono formulare molte algebre Booleane, in funzione della scelta sia degli elementi di B che delle regole delle operazioni. Noi ci limiteremo a trattare soltanto l'algebra Booleana a due valori che ha grandi applicazioni sia nella teoria degli insiemi (l'algebra delle classi) che nella logica proposizionale. Il nostro principale interesse in quest'algebra è ovviamente le applicazione all'algebra nei circuiti logici.

3.2 Algebra Booleana a due valori

x	y	x • y (AND)
0	0	0
0	1	0
1	0	0
1	1	1

x	y	x + y (OR)
0	0	0
0	1	1
1	0	1
1	1	1

x	x' (NOT)
0	1
1	0

Tab. 3.1 Tabella di verità per gli operatori dell'algebra Booleana a 2 valori

L'algebra Booleana a due valori è definita scegliendo l'insieme *B* costituto da due soli elementi, B = {0,1}, la regola di complementazione: se x = 0 => x'= 1 e se x = 1 => x'= 0, e due operatori binari: + e · definiti dalle relative tabelle della verità, che coincidono con gli operatori OR e AND in logica positiva già visti e che qui riportiamo per convenienza.
Dobbiamo ora verificare che questa scelta soddisfa i postulati dell'algebra Booleana.

1a) Chiusura rispetto all'operatore + :
Dalla tabella si vede che il risultato è 0 o 1;
2a) Elemento identità rispetto a +, designato con 0: 0 + 0 = 0 ; 0 + 1 = 1 + 0 = 1;
3a) Legge commutativa per l'oper. + è ovvia per la simmetria della tabella dell'operatore
5a) Esistenza del complemento: x + x' = 1
infatti 0 + 0' = 0 + 1 = 1 e 1 + 1'= 1 + 0 =1;
6a) Esistono almeno due elementi distinti :
è evidente dalla definizione dell'algebra

1b) Chiusura rispetto all'operatore · :
Dalla tabella si vede che il risultato è 0 o 1;
2b) Elemento identità rispetto a ·, designato con 1: 1 • 1 = 1; 0 • 1 = 1 • 0 = 0;
3b) Legge commutativa per l'oper. • è ovvia per la simmetria della tabella dell'operatore
5b) Esistenza del complemento: x • x' = 0
infatti 0 • 0' = 0 • 1 = 0 e 1 • 1' = 1 • 0 = 0 ;
6b) Esistono almeno due elementi distinti:
è evidente dalla definizione dell'algebra

4a) Legge distributiva del prodotto rispetto alla somma: x • (y + z) = (x • y) + (x • z), può essere verificata formando la tabella di tutte le combinazioni possibile di x, y e z. Per ognuna di queste, usando la tabella degli operatori, derivare i valori per le funzioni: x · (y + z) e (x · y) + (x · z) e quindi fare rilevare la loro uguaglianza.

x	y	z	y + z	x • (y+z)	x • y	x • z	(x • y)+(x • z)
0	0	0	0	0	0	0	0
0	0	1	1	0	0	0	0
0	1	0	1	0	0	0	0
0	1	1	1	0	0	0	0
1	0	0	0	0	0	0	0
1	0	1	1	1	0	1	1
1	1	0	1	1	1	0	1
1	1	1	1	1	1	1	1

4b) Legge distributiva della somma rispetto al prodotto: x + (y • z) = (x + y) • (x + z) può essere verificata in modo simile della 4a).

x	y	z	y • z	x + (y • z)	x + y	x + z	(x + y) • (x + z)
0	0	0	0	0	0	0	0
0	0	1	0	0	0	1	0
0	1	0	0	0	1	0	0
0	1	1	1	1	1	1	1
1	0	0	0	1	1	1	1
1	0	1	0	1	1	1	1
1	1	0	0	1	1	1	1
1	1	1	1	1	1	1	1

Con quest'ultima dimostrazione abbiamo stabilito non solo che la scelta dei due elementi 0 e 1, la regola di complementazione e gli operatori somma e prodotto di Tab. 3.1 definiscono una algebra Booleana a due valori ma anche che, poiché questi sono identici agli elementi ed operatori già presi in considerazione nel cap. 2, tutti i teoremi, che possono essere dedotti dall'algebra Booleana a due valori, sono applicabili alle funzioni che possono essere scritte con le variabile binarie o logiche introdotte in quel capitolo.

3.2.1 Proprietà fondamentali e Teoremi dell'Algebra Booleana

Principio Dualità

Abbiamo elencato i postulati che definiscono l'algebra Booleana in coppia: parte (a) e parte (b). Il lettore verifichi che gli enunciati della parte (a) possono essere ottenuti dalla parte (b) e viceversa con lo scambio dell'operatore binario e dell'elemento identità. Questa importante proprietà che avevamo già messo in evidenza in modo euristico, si chiama *principio di dualità*. Esso stabilisce che *qualsiasi espressione algebrica, deducibile dai postulati dell'algebra Booleana, resta valida se si scambiano tra di loro gli operatori somma- prodotto e gli elementi identità.*
Per l'algebra a due valori l'enunciato equivale allo scambio degli operatori AND ed OR e lo 0 con 1 e viceversa; cosa già messa in evidenza.

Teoremi Fondamentali

Ricordiamo che mentre i postulati sono assiomi fondamentali della struttura algebrica e non necessitano di dimostrazione, i teoremi, invece, debbono essere provati utilizzando i postulati. Ricordiamo ancora che, essendo valido il principio di dualità, se l'enunciato di un teorema può essere ricavato da quello duale basta dimostrarne solo uno perché l'altro resta provato per dualità.
Nel Cap. 2 abbiamo mostrato che gli operatori AND ed OR sono equivalenti rispettivamente alle operazioni di intersezione e di unione quando si lavora con le mappe. Bisogna qui precisare quali insiemi o sotto insiemi corrispondono ai due elementi identità: *all'elemento identità della somma, lo 0 , corrisponde un insieme che non contiene punti dello spazio cioè l'insieme vuoto, mentre a quello del prodotto 1 corrisponde un insieme che contiene tutto lo spazio*. Per la dimostrazione dei teoremi potremmo quindi utilizzare sia la derivazione algebrica, sia le tabelle degli operatori che le mappe. Noi faremo uso soltanto di quello che risulterà più facile da memorizzare ed usare, ma il lettore provi a trovare le dimostrazioni anche con gli altri due metodi.

Teorema 1 o dell'idem potenza:
a) $\mathbf{x + x = x}$; L'unione di un sotto insieme con se stesso produce il sotto insieme stesso.
b) $\mathbf{x \cdot x = x}$; L'intersezione di un sotto insieme con se stesso produce il sotto insieme stesso.

Teorema 2:
a) $\mathbf{x + 1 = 1}$; L'unione di un sotto insieme con tutto lo spazio produce l'intero spazio.
b) $\mathbf{x \cdot 0 = 0}$; L'intersezione di un sotto insieme con l'insieme vuoto produce l'insieme vuoto.

Teorema 3 o dell'involuzione:

a) **(x')' = x**; Il complemento del complemento di un insieme è l'insieme stesso.

b) ; non esiste poiché non sono coinvolti operatori binari.

Teorema 4: Legge associativa:

Questo teorema permette di estendere i due operatori + e · a più di due operandi.

a) **(x + y)+ z = x +(y+z)**; La dimostrazione di questo teorema risulta immediato utilizzando il metodo della tabella della verità come fatto per la legge distributiva.

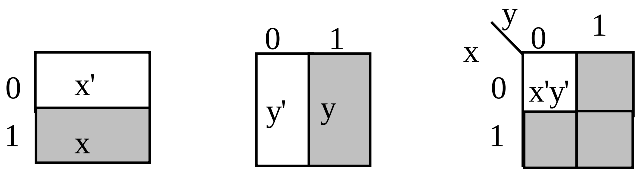
b) **(x · y) · z = x · (y · z)** ; per dualità o come per la (a).

Notare che poiché è valido anche la commutatività è possibile scrive le precedenti come:

(x + y)+ z = x +(y+z) = x + y + z **(x · y) · z = x · (y · z) = x · y · z**

Teorema 4 o di De Morgan:

a) **(x + y)' = x' · y'**; La dimostrazione può essere fatta con l'aiuto delle tre seguenti figure in cui lo spazio associato al valore 1 della variabile compare ombreggiato. La terza figura rappresenta, ombreggiata, l'unione di x ed y cioè x + y per cui la parte non ombreggiata è l'insieme complementare che è proprio individuato da x' • y'.



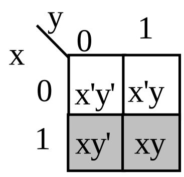
b) (**x · y)' = x' + y'**; Vera per dualità.

Il teorema di *De Morgan* può essere *generalizzato* a più di due variabili. Per es. per l'espressione con 3 variabili sia ha (x + y + z)' = [x + (y + z)]' = x' • (y + z)' = x' • (y' • z') = x' · y' · z' . Questa espressione può essere generalizza come segue:

(x + y + ...+ z + t)' = x' · y' · ...z' · t' e la duale **(x · y · ... · z · t)'= x' + y' + ...+ z' + t'**

Teorema 5 o dell'assorbimento:

a) **x + x · y = x**; L'unione di un sotto insieme con un altro sotto insieme che è una sua parte deve coincidere con il sotto insieme di partenza.



b) **x · (x + y)= x**; Vera per dualità.

3.2.2 Precedenza degli operatori

La precedenza degli operatori per il calcolo delle espressioni Booleane è identica a quella dell'algebra normale se si fa corrispondere all'operatore AND il prodotto e all'operatore OR la somma Booleana. L'operatore NOT (che non esiste nell'algebra normale), si pone in ordine tra la parentesi e il prodotto. Pertanto la precedenza è: 1) Operazioni indicate dentro la parentesi, 2) Not, 3) AND, 4) OR. Per es.: Provare, usando la tabella di verità, il teorema di De Morgan: $(x + y)' = x'y'$. A sinistra abbiamo $(x + y)'$; prima calcoliamo l'espressione dentro la parentesi e poi complementiamo il risultato. A destra abbiamo l'espressione $x'y'$, quindi, prima complementiamo sia x che y e poi eseguiamo l'AND dei risultati. Infine fare rilevare che le due espressioni prendono gli stessi valori per la stessa combinazione di ingressi.

3.3 Funzioni Booleane

Una variabile Booleana può prendere o il valore 0 o 1. *Una **funzione Booleana** è una espressione formata con variabili binarie, i due operatori binari OR e AND, l'operatore NOT, parentesi e segno di uguaglianza*. Per un dato valore delle variabili la funzione può valere solo 0 o 1 (chiusura). Consideriamo le seguenti funzioni Booleane in cui l'operatore prodotto è stato omesso perché implicito:

$F_1 = xyz'$ $\quad$ $F_2 = x + y'z;$ $\quad$ $F_3 = x'y'z + x'yz + xy';$ $\quad$ $F_4 = xy' + x'z;$

Questi sono esempi di funzioni Booleane rappresentate da una espressione algebrica.
Ma le funzioni Booleane possono anche essere rappresentata da una tabella della verità. Per rappresentare una funzione Booleana con una tabella della verità, noi dobbiamo elencare tutte le 2^n combinazioni di 1 e 0 delle n variabili binarie e scrivere, in una colonna a fianco, il valore 1 o 0 della funzione per ogni combinazione di variabili. Per le funzioni sopra elencate contenenti 3 variabili binarie ciascuna, la tabella di verità relativa avrà 8 entrate ed 7 colonne (3 per le variabili e 4 per le funzioni, vedi Tab. 3.2).

x	y	z	F_1	F_2	F_3	F_4
0	0	0	0	0	0	0
0	0	1	0	1	1	1
0	1	0	0	0	0	0
0	1	1	0	0	1	1
1	0	0	0	1	1	1
1	0	1	0	1	1	1
1	1	0	1	1	0	0
1	1	1	0	1	0	0

Tab. 3.2

Un altro modo di rappresentare le funzione Booleane è con l'uso delle mappe. Queste hanno il

x\y z	00	01	11	10
0				
1				1

F_1

x\y z	00	01	11	10
0		1		
1	1	1	1	1

F_2

x\yz	00	01	11	10
0		1	1	
1	1	1		

F_3

x\yz	00	01	11	10
0		1	1	
1	1	1		

F_4

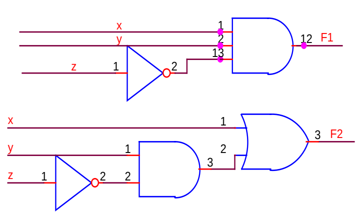
Fig. 3.1

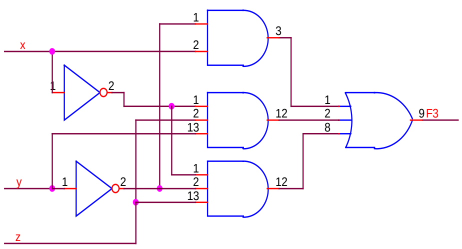
vantaggio, rispetto alla tabella della verità, di una rappresentazione pittorica immediatamente chiara. Ovviamente il numero di quadratini, per quanto ricordato anche precedentemente, è 2^n e se noi rinunciamo a fare mappe che caratterizzano i quadratini vicini, le variabili possono essere più di 4.

La Fig. 3.1 rappresenta le mappe delle funzioni F_1, F_2, F_3, F_4 in cui è stato posto un 1 soltanto nelle caselle che corrispondono alle combinazione di valori delle variabili binarie per cui la relativa funzione assume il valore 1.

Analizziamo più da vicino le funzioni proposte. F_1 è uguale a 1 se x = 1, y = 1 e z = 0, altrimenti F_1 è uguale a 0. Ciò lo ricaviamo dalla stessa espressione, ed è rispecchiato sia nella tabella della verità che, in modo più immediato, nella mappa per F_1. Una elencazione simile può essere fatta per tutte le altre funzioni e la lasciamo al lettore per esercizio. Noi facciamo soltanto l'osservazione che le funzioni F_3 ed F_4 mostrano, in modo immediatamente evidente nella mappa, che non solo hanno lo stesso numero di 1 ma anche nella stessa posizione. A questo punto dobbiamo chiederci se l'espressione algebrica di una funzione Booleana è unica o no. In altre parole è possibile scrivere la stessa funzione con due espressioni algebriche diverse? La risposta a questa domanda è evidentemente affermativa in quanto che, se non conoscessimo l'espressione di provenienza, guardando le mappe noi diremmo senz'altro che F_3 = F_4 cioè rappresentano la stessa funzione. Allora concludiamo che: in generale, *due funzioni di n variabili binarie si dicono uguali se esse assumono gli stessi valori per tutte le* 2^n *combinazioni delle* n *variabili*.

3.4 Funzioni Booleane e Circuiti Logici





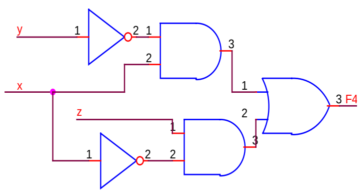


Fig. 3.2: Fig 3.3 Fig.3.4 Fig. 3.5

Una Funzione Booleana può essere trasformata da espressione algebrica a circuito logico composto da porte AND, OR e NOT. Nelle Figg. 3.2 - 3.5 sono mostrati i circuiti logici relative alle quattro funzioni presentate nel paragrafo precedente. Il circuito logico include un circuito inverter per ogni variabile presente in forma complementata. (L'inverter non è necessario se sono disponibili le variabili complementate).

Notare che abbiamo usato porte AND ed OR con più di 2 ingressi, questo è permesso dal fatto che, per questi operatori, vale sia la legge associativa che quella commutativa. Paragonando il circuiti logici che realizzano le funzioni F_3 e F_4 è evidente che il numero di porte usato è

diverso. In particolare la realizzazione di F_3 presenta un numero maggiore di porte e quindi di collegamenti.
Questa osservazione ci fa concludere che l'espressione algebrica influisce sulla complessità (e quindi sul costo), sia dei componenti che dei collegamenti e sulla affidabilità delle realizzazione. Più collegamenti ci sono più alta è la probabilità che uno di essi sia errato o faccia un falso contatto. Concludiamo che *il costo delle realizzazione di una funzione Booleana dipende dalla sua espressione algebrica* per cui sarebbe opportuno, usare una espressione che, realizzando la stessa funzione, ne minimizza il costo di realizzazione. Trovare espressioni algebriche che minimizzano il costo di realizzazione elettronica è fondamentale ed è oggetto dei prossimi paragrafi e capitoli.
Come primo passo verso la minimizzazione cerchiamo di individuare i responsabili dei costi di realizzazione. Definiamo *letterale* una variabile Booleana complementata o no. Quando una funzione Booleana è realizzata con porte logiche, ciascun letterale rappresenta un ingresso di una porta, ed una porta rappresenta un termine dell'espressione della funzione. La minimizzazione del numero di letterali e del numero dei termini produrrà un circuito con minor componenti. Non è sempre possibile minimizzare entrambi contemporaneamente; per far ciò bisogna ricorrere ad altri criteri. Per il momento noi ci limiteremo a considerare che *la minimizzazione coincide con la minimizzazione dei letterali*. Il numero dei letterali di una funzione Booleana può essere minimizzato per manipolazione algebrica. Sfortunatamente non esistono regole specifiche che ci assicurino la migliore risposta finale. Il solo metodo possibile è quello di prova e riprova applicando i postulati ed i teoremi dell'algebra Booleana.
Pertanto utilizzeremo dei metodi alternativi che saranno spiegati nel prossimo capitolo. In quel che segue definiremo il substrato necessario ad una loro migliore comprensione.

3.5 Complemento di Funzioni Booleane

Il complemento di una Funzione Booleana F è la funzione F' ottenuta intercambiando lo 0 con 1 e 1 con 0 nella tabella della verità che rappresenta F. Il complemento di una funzione può essere anche derivato algebricamente applicando il *teorema di De Morgan*. Questo nella sua generalità può essere enunciato come: *Il complemento di una funzione può essere ottenuto intercambiando gli operatori AND e OR e complementando ciascun letterale*.

Es. 3.1 Trovare il complemento della funzione $F_1 = x'yz' + x'y'z$ ed $F_2 = x(y'z' + yz)$.
Applicando il teorema di De Morgan tante volte quanto è necessario, i complementi sono ottenuti nel seguente modo:

$$F'_1 = (x'yz' + x'y'z)' = (x'yz')' \cdot (x'y'z)' = (x + y' + z)(x + y + z')$$

$$F'_2 = [x(y'z' + yz)]' = x' + (y'z' + yz)' = x' + (y + z)(y' + z')$$

Un procedimento più semplice, per derivare il complemento di una funzione, è quello di prendere la funzione duale e complementare tutti i letterali. Questo metodo deriva dallo stesso teorema di De Morgan. Ricordiamo che il duale di una funzione si trova intercambiando OR ed AND ed 1 con 0.

Es. 3.2 Trovare il complemento della funzione F_1 = x'yz' + x'y'z ed F_2 = x(y'z' + yz) prendendo i loro duali e complementando i letterali.

F_1 duale = (x' + y + z') (x' + y' + z) => F'_1 = (x + y' + z) (x + y + z')

F_2 duale = x + (y' + z') (y + z) => F'_2 = x' + (y + z) (y' + z')

3.6 Forme canoniche delle Funzioni Booleane

Una variabile binaria può apparire sia in forma normale *x* che in forma complementata, *x'*. Consideriamo ora due variabili binarie x ed y e formiamo l'AND delle due. Poiché ciascuna variabile può comparire in entrambe le forme, ci saranno 4 possibili combinazioni: xy, x'y, xy' e x'y'. Ciascuno di queste combinazioni o termini, che rappresenta un quadratino in una mappa, prende il nome di *prodotto standard* o (in inglese) *minterm*. Similmente possiamo combinare *n* variabili per formare 2^n prodotti standard. I 2^n differenti minterm possono essere determinati con un metodo simile a quello mostrato in Tab. 3.3 per 3 variabili. Nelle colonne intestate con le *n* variabili sono elencati i numeri binari da 0 a 2^n -1. Ciascun prodotto standard è ottenuto dall'AND delle *n* variabili, con ciascuna variabile complementata se il corrispondente bit nel numero binario è 0 e non complementata se è 1.

X	y	z	Minterm termine	Minterm Designazione	Maxterm termine	Maxterm Designazione
0	0	0	x'y'z'	m_0	x+y+z	M_0
0	0	1	x'y'z	m_1	x+y+z'	M_1
0	1	0	x'yz'	m_2	x+y'+z	M_2
0	1	1	x'yz	m_3	x+y'+z'	M_3
1	0	0	xy'z'	m_4	x'+y+z	M_4
1	0	1	xy'z	m_5	x'+y+z'	M_5
1	1	0	xyz'	m_6	x'+y'+z	M_6
1	1	1	xyz	m_7	x'+y'+z'	M_7

Tab. 3.3 Minterm e Maxterm per tre Variabili Binarie

Inoltre è stato utilizzato un simbolo della forma m_J dove j denota l'equivalente decimale del numero binario che determina il minterm.

In modo simile, si formano i termini OR di *n* variabili. Scegliendo ciascuna variabile in modo accentato e non otteniamo 2^n possibili combinazioni, chiamate *somme standard* o (in inglese) *maxterm*. Gli 8 termini che si ottengono con 3 variabili sono elencati in Tab. 3.2. Ogni uno dei 2^n maxterm che si possono ottenere con *n* variabili possono essere determinati in modo simile. *Ciascun maxterm è ottenuto formando il termine OR (o somma logica, o unione) delle n variabili, ciascuna presa in modo accentato se il corrispondente bit è ad 1 o normale se è a 0.* Notare che ciascun maxterm è il complemento del corrispondente minterm e viceversa cioè: $\mathbf{M_J = m'_J}$.

Una funzione Booleana può essere espressa algebricamente partendo da una data tabella della verità, scrivendo un minterm per ciascuna combinazione delle variabili che corrispondono ad un 1 nella funzione e facendone la somma logica cioè l'OR.

Per es. la funzione f_1, nella Tab. 3.4, è determinata esprimendo le combinazioni 001, 100, e 111 come x'y'z, xy'z' e xyz, rispettivamente. Siccome ciascun dei minterm rende $f_1 = 1$ noi possiamo scrivere

$$f_1 = x'y'z + xy'z' + xyz = m_1 + m_4 + m_7$$

Allo stesso modo può essere verificato che

$$f_2 = x'yz + xy'z + xyz' + xyz = m_3 + m_5 + m_6 + m_7$$

x	y	z	f_1	f_2
0	0	0	0	0
0	0	1	1	0
0	1	0	0	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Tab. 3.4 Funzioni di 3 Variabili

Questi esempi dimostrano una importante proprietà dell'algebra Booleana: *ogni funzione Booleana può essere espressa come somma (OR) di minterm.*

Consideriamo ora il complemento di una funzione Booleana. Questa può essere tratta dalla tabella di verità formando i minterm di ogni combinazione che corrisponde a 0 della funzione e poi facendone la somma logica. Così il complemento di f_1 si scrive come:

$$f'_1 = x'y'z' + x'yz' + x'yz + xy'z + xyz' = m_o + m_2 + m_3 + m_5 + m_6$$

Se prendiamo il complemento di f'_1, otteniamo ancora f_1:

$$f_1 = (x + y + z)(x + y' + z)(x + y' + z')(x' + y + z')(x' + y' + z) = M_o \cdot M_2 \cdot M_3 \cdot M_5 \cdot M_6$$

In modo simile per f_2 troviamo:

$$f_2 = (x + y + z)(x + y + z')(x + y' + z)(x' + y + z) = M_o \cdot M_1 \cdot M_2 \cdot M_4$$

Questi esempi dimostrano una seconda importante proprietà dell'algebra Booleana: *Ogni funzione può essere espressa come prodotto (AND) di maxterm.*

Il procedimento per ottenere, direttamente dalla tabella di verità, il prodotto di maxterm è il seguente: formare i maxterm di ciascuna combinazione di variabili che corrisponde ad uno 0 della funzione e quindi formare l'AND di questi maxterm.

Le funzioni Booleane espresse come somma di minterm o come prodotto di maxterm sono dette essere scritte in *forma canonica*.

3.6.1 Trasformazione di funzioni Booleane in forma canonica

Una caratteristica peculiare delle funzioni Booleane, espresse in forma canonica, è che ogni termine, sia esso maxterm o minterm, contiene tutte le variabili da cui la funzione dipende (in forma normale o complementata). Nelle forme non canoniche ciò non accade per cui potrebbe essere opportuno trasformarle in una delle due forma canoniche (essendo queste la forma da cui si parte per la minimizzazione dei letterali).

Possono essere seguiti diversi metodi:

a) *Uso della tabella di verità*:

1a) Individuare il numero e le variabili Booleane da cui la funzione dipende;

2a) Scrivere una tabella della verità in cui compaiono tutte le combinazione delle variabili individuate al punto 1) e scrivere nella colonna dei valori della funzione il valore che la stessa assume, per ogni combinazione di valori delle variabili, calcolato a partire dalla espressione di partenza;

3a) Avendo costruito la tabella di verità della funzione procedere, come spiegato in precedenza, a scrivere la funzione in una delle due forme canoniche

b) *Uso delle mappe*: questo metodo non è diverso da quello della tabella di verità. L'individuazione della combinazione di variabili normale o complementate di ogni quadratino corrisponde ad una entrata della tabella di verità ed in esso si scrive il valore che assume la funzione per quella combinazione. Quindi si procede come al punto 3a) utilizzando le coordinate di ogni quadratino marcato 1 per la forma canonica in somma di prodotti e le coordinate di ogni quadratino marcato con 0 per la forma canonica in prodotto di somme.

c) *Manipolazione matematica*: questo metodo differisce dai metodi precedenti e si distingue rispetto alla forma canonica che si vuole scrivere:

1c) somma di Minterm

I) Se la funzione Booleana non è in forma di somma di prodotti (*forma standard*) si eseguono le manipolazioni algebriche necessarie per ottenerla in tale forma;

II) Si ispeziona ciascun termine per vedere se contiene tutte le variabili. Se qualcuna delle variabili manca si moltiplica per una espressione del tipo x + x', dove x è una delle variabili che manca e si esegue il prodotto.

III) Per ogni altra variabile che manca, in ogni termine risultante, si ripete l'operazione II;

IV) Si ripetono le operazioni II) e III) per tutti gli altri termini prodotto della funzione.

1c) Prodotto di Maxterm

I) Se la funzione Booleana non è in forma di prodotto di somme (*forma standard*) si eseguono le manipolazioni algebriche necessarie per ottenerla in tale forma;

II) Si ispeziona ciascun termine per vedere se contiene tutte le variabili. Se qualcuna delle variabili manca si somma una espressione del tipo x · x', dove x è una delle variabili che manca

III) Per ogni altra variabile che manca, in ogni termine risultante, si ripete l'operazione II;

IV) Si ripetono le operazioni II) e III) per tutti gli altri termini somma della funzione.

Es. 1 Esprimere la funzione Booleana F = A + B'C in somma di minterm.

Questa espressione è già in forma standard, somma di prodotti. Le variabili da cui dipende sono 3 e cioè A, B, C.

a) *Manipolazione algebrica*:

Il primo termine manca di 2 variabili: B e C, quindi:

$$A = A(B + B') = AB + AB'$$

Ogni termine trovato manca ancora della variabile C, quindi:

$$A = A(B + B') = AB + AB' = AB(C + C') + AB'(C + C') = ABC + ABC' + AB'C + AB'C'$$

Il secondo termine, dell'espressione originaria, manca di 1 variabile: A, quindi:

$$B'C = B'C(A + A') = AB'C + A'B'C$$

e combinando le espressioni trovate abbiamo:

$$F = ABC + ABC' + AB'C + AB'C' + AB'C + A'B'C =$$
$$ABC + ABC' + AB'C + AB'C' + A'B'C$$

Nell'ultima espressione abbiamo eliminato una delle due espressioni identiche AB'C per il teorema 1 (x + x = x). Disponendo i termini in ordine ascendente otteniamo:

$$F = A'B'C + AB'C' + AB'C + ABC' + ABC = m_1 + m_4 + m_5 + m_6 + m_7$$

b) *Tabella di verità*:

Formando la tabella di verità delle 3 variabili e ponendo nella colonna della funzione il valore che la funzione assume in corrispondenza ad ogni combinazione delle variabili si ottiene la tabella a fianco. Da cui si ha $F = m_1 + m_4 + m_5 + m_6 + m_7$

A	B	C	F
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

c) *Mappe*:

Formando la mappa delle 3 variabili e ponendo in ogni casella il valore della funzione si ottiene il diagramma a fianco. Da cui si ha $F = m_1 + m_4 + m_5 + m_6 + m_7$

A/BC	00	01	11	10
0	0	1	0	0
1	1	1	1	1

Es. 2 Esprimere la funzione Booleana F = xy + x'z in prodotto di maxterm.
Questa espressione non è in forma di prodotto di maxterm per cui bisogna prima trasformala. Applicando la legge distributiva della somma rispetto al prodotto abbiamo:

$$F = xy + x'z = (xy + x')(xy + z) =(x' + x)(x' + y)(z + x)(z + y) = (x' + y)(z + x)(z + y)$$

La funzione ha 3 variabili cioè x, y, z, quindi:

a) *Manipolazione algebrica*:
Il primo termine manca di una variabile: z, quindi:

$$x' + y + zz' = (x' + y + z)(x' + y + z')$$

Il secondo termine manca di una variabile: y, quindi:

$$z + x + yy' = (x + y + z)(x + y' + z)$$

Il terzo termine manca di una variabile: x, quindi:

$$z + y + xx' = (x + y + z)(x' + y + z)$$

Combinando tutti i termini e rimuovendo quelli ripetuti otteniamo:

$$F = (x + y + z)(x + y' + z)(x' + y + z)(x' + y + z') = M_0 \cdot M_2 \cdot M_4 \cdot M_5$$

b) *Tabella di verità*:
Formando la tabella di verità delle 3 variabili e ponendo nella colonna della funzione il valore che la funzione assume in corrispondenza ad ogni combinazione delle variabili si ottiene la tabella a fianco. Da cui si ha $F = M_0 \cdot M_2 \cdot M_4 \cdot M_5$

x	y	z	F
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

c) *Mappe*:
Formando la mappa delle 3 variabili e ponendo in ogni casella il valore della funzione si ottiene il diagramma a fianco. Da cui si ha $F = M_0 \cdot M_2 \cdot M_4 \cdot M_5$

A/BC	00	01	11	10
0	0	1	1	0
1	0	0	1	1

3.6.2 Conversione tra Forme Canoniche di Funzioni Booleane

Il complemento di una funzione espressa come somma di prodotti è uguale alla somma dei minterm mancanti nell'espressione originale. Ciò perché la funzione originale è espressa per mezzo dei minterm che rendono la funzione uguale ad 1, mentre il complemento della funzione è 1 per quei minterm che rendevano la funzione originale uguale a 0. Per es. la funzione:

$$F(A, B, C, D) = m_1 + m_4 + m_5 + m_6 + m_7$$

ha per complemento la funzione

$$F'(A, B, C, D) = m_0 + m_2 + m_3$$

Se ora prendiamo il complemento di F' ed utilizziamo il teorema di De Morgan otteniamo ancora F in forma differente:

$$F(A, B, C, D) = (m_0 + m_2 + m_3)' = M_0 M_2 M_3$$

essendo il maxterm $M_j = m'_j$

Cioè il numero e l'indice dei minterm, che definiscono il complemento di una funzione e quello dei maxterm che definiscono la funzione resta invariato.

L'ultimo esempio ha mostrato la conversione della forma canonica da somma di prodotto a prodotto di somme. Un procedimento analogo mostrerà la conversione tra le forme canoniche prodotto di somme e somma di prodotti. Pertanto possiamo stabilire un procedimento generale. *Per convertire una forma canonica nell'altra si scambia la somma (prodotto) con il prodotto (somma) e invece dei minterm (maxterm) originali si pongono i maxterm (minterm) che mancano nella forma originale*. Un esempio renderà più chiaro l'enunciato. La funzione:

$$F(x, y, z) = m_0 + m_2 + m_4 + m_5$$

è espressa sotto forma di somma di prodotti. La sua conversione in prodotti di somme si esprime:

$$F(x, y, z) = M_1 M_3 M_6 M_7$$

Notare che nel ritrovare i termini mancanti bisogna ricordarsi che il numero totale di termini che si hanno con *n* variabile è 2^n.

3.7 Lista degli operatori Binari.

Ricordiamo che per o*peratore binario si intende un operatore che agisce su due variabili appartenenti all'insieme degli elementi di un algebra e fornisce come risultato un elemento appartenente ancora allo stesso insieme*. Noi abbiamo usato essenzialmente soltanto due operatori binari AND e OR. Ma in realtà poiché, per *n* variabili noi abbiamo 2^n minterm o maxterm e per ogni uno di essi la funzione può assume o il valore 0 o il valore 1, è possibile definire 2^{2^n} funzioni. Se n = 2 si ottengono 16 funzioni diverse che, per definizione, possono essere chiamati operatori binari. Le loro tabelle di verità sono in Tab. 3.4; l'indice che compare nei nomi delle funzioni "$\mathbf{f_j}$ " va da 0 a 15 e corrisponde alla lettura delle colonne come un

x	y	$\mathbf{f_0}$	$\mathbf{f_1}$	$\mathbf{f_2}$	$\mathbf{f_3}$	$\mathbf{f_4}$	$\mathbf{f_5}$	$\mathbf{f_6}$	$\mathbf{f_7}$	$\mathbf{f_8}$	$\mathbf{f_9}$	$\mathbf{f_{10}}$	$\mathbf{f_{11}}$	$\mathbf{f_{12}}$	$\mathbf{f_{13}}$	$\mathbf{f_{14}}$	$\mathbf{f_{15}}$
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
Operat.			$\cdot$	$>$		$<$		$\oplus$	$+$	$\downarrow$	$\odot$		$\geq$		$\leq$	$\uparrow$	

Tab. 3.4 Tabella della verità per le 16 funzioni Booleane di due variabili

numero binario (avente il bit più significativo in alto). Nell'ultimo rigo sono posti anche i simboli di identificazione degli operatori quando ne esiste uno.
La Tab. 3.5 contiene la lista di tutti gli operatori binari che ne derivano espressi algebricamente. Paragonando le funzioni per cui la somma degli indici =15 cioè f_i e f_{15-i} con i = 0,..,7 si nota che esse sono uno il complementare dell'altro e quindi lo sono anche i corrispondenti operatori binari.

Funzione	Nome	Commento
$f_0 = 0$		Zero o funzione nulla
$f_1 = x \cdot y$	AND	x and y
$f_2 = xy' = x > y$	INHIBITION	x ma non y o x > y
$f_3 = x$		la funzione è uguale ad x o x ≥ y
$f_4 = x'y = x < y$	INHIBITION	y ma non x o x < y
$f_5 = y$		la funzione è uguale ad y
$f_6 = x'y + xy' = x \oplus y$	EXCLUSIVE OR	x o y ma non entrambi o x ≠ y
$f_7 = x + y$	OR	x o y o entrambi
$f_8 = (x + y)' = x \downarrow y$	NOR	Not OR
$f_9 = x'y' + xy = x \odot y$	EQUIVALENCE	x uguale ad y o x = y
$f_{10} = y'$	COMPLEMENT	Not y
$f_{11} = x + y' = x \geq y$	IMPLICATION	se y allora x o x ≥ y
$f_{12} = x'$	COMPLEMENT	Not x
$f_{13} = x' + y = x \leq y$	IMPLICATION	se x allora y o x ≤ y
$f_{14} = (x \cdot y)' = x \uparrow y$	NAND	Not and
$f_{15} = 1$		Uno o funzione identità

Tab. 3.5 Tabella degli operatori binari

In particolare si noti che l'EQ è il complementare dell'XOR come è evidente dalla tabella di verità o dalla seguente manipolazione algebrica:

$$f'_9 = (x'y' + xy)' = (x + y)(x' + y') = xy' + x'y = f_6$$

Nell'algebra Booleana definita nel paragrafo 3.1 abbiamo usato i due operatori binari: AND e OR ed un operatore "unario" (che cioè agisce su una sola variabile) NOT. Da quella definizione abbiamo ricavato un certo numero di proprietà di questi operatori ed ora abbiamo definito altri operatori in termini di essi. Questa procedura come avevamo anticipato, non è unica. Per esempio saremmo potuti partire dalla definizione dell'operatore NOR, e poi definire gli operatori AND, OR e NOT in funzione di questo. Ci sono, tuttavia delle ottime ragioni per introdurre l'algebra Booleana nel modo in cui l'abbiamo fatto. I concetto di "and" "or" e "not" sono sufficientemente familiari e sono usati normalmente da tutti nell'esprimere le idee logiche nella vita di tutti i giorni. Inoltre i postulati di Huntington riflettono la natura duale dell'algebra, enfatizzando la simmetria reciproca dei due operatori AND e OR.

PROBLEMI

P3.1) Disegnare lo schema logico, utilizzando soltanto porte NAND la seguente Funzione F=AB

P3.2) Disegnare lo schema logico, utilizzando soltanto porte NOR la seguente Funzione F=AB

P3.3) Disegnare lo schema logico, utilizzando soltanto porte NAND la seguente Funzione F=A+B

P3.4) Disegnare lo schema logico, utilizzando soltanto porte NOR la seguente Funzione F=A+B

P3.5) Disegnare lo schema logico, utilizzando soltanto porte NAND la seguente Funzione F=A(B+CD)+BC’

P3.6) Disegnare lo schema logico, utilizzando soltanto porte NAND la seguente Funzione F=(A+B’)(CD+E)

P3.7) Disegnare lo schema logico, utilizzando soltanto porte NAND la seguente Funzione F=AB+CD+E

P3.8) Disegnare lo schema logico, utilizzando soltanto porte NAND la seguente Funzione F=(A+B)(C+D)E

P3.9) Disegnare lo schema logico, utilizzando soltanto porte NAND la seguente Funzione F=(A+B)(C+D)E

P3.10) Disegnare lo schema logico, utilizzando soltanto porte NOR la seguente Funzione F=A(B+CD)+BC’

P3.11) Disegnare lo schema logico, utilizzando soltanto porte NOR la seguente Funzione F=A(B+CD)+BC’

P3.12) Disegnare lo schema logico, utilizzando soltanto porte NOR la seguente Funzione F=(B’+C)(A+C’+D)(A’+B+C’+D’)

P3.13) Disegnare lo schema logico, utilizzando soltanto porte NOR e OR la seguente Funzione F=A’B’+C’D’+E

P3.14) Disegnare lo schema logico, utilizzando soltanto porte NAND e AND la seguente Funzione F=(A’+B’)+(C’+D’)E

P3.15) Disegnare lo schema logico, utilizzando soltanto porte OR e NAND la seguente Funzione F=A’B’+C’D’+E’

P3.16) Disegnare lo schema logico, utilizzando soltanto porte AND e NOR la seguente Funzione F=(A’+B’)+(C’+D’)E’

P3.17) Disegnare lo schema logico, utilizzando soltanto porte NAND la seguente Funzione F=XOR(A,B)

P3.18) Ridurre le seguenti espressioni Booleane al numero di letterali indicato

a) ABC+A’B’C+A’BC+ABC’+A’B’C’ a 5 letterali

b) BC+AC’+AB+BCD a 4 letterali

c) [(CD)’+A]’+A+CD+AB a 3 letterali

d) (A+C+D)(A+C+D’)(A+C’+D)(A+B’) a 4 letterali

P3.19) Trovare il complemento delle seguenti funzioni Booleane e ridurle al minimo numero di letterali

a) (BC’+A’D)(AB’+CD’) b) B’D+A’BC’+ACD+A’BC c) [(AB)’A][(AB)’B]

d) AB’+C’D’+A’CD’+DC’(AB+A’B’) +BD(AC’+A’C)

P3.20) Date le funzione qui di seguito indicate determinare le funzioni F_1+G_1 e F_2+G_2 e ridurle al minimo numero di letterali

a) F_1 =D+ABC’+A’C G_1 =D’(A’+B’+C)(A+C’)

b) F_2 =AB’+A’B’+C’D+A’CD’ G_2 =C+A’B’C’+AB’C’+BC’D’

P3.21) Ricavare le tabelle di verità dei problemi P3.18 e P3.20

P3.22) Esprimere le seguenti funzioni in somma di minterm e prodotto di maxterm.

a) $F(A,B,C,D)=D(A'+B)+B'D$ b) $F(w,x,y,z)=y'z+wxy'+wxz'+w'x'z$

c) $F(A,B,C,D)=(A+B'+C)(A+B')(A+C'+D')(A'+B+C+D')(B+C'+D')$

d) $F(A,B,C)=(A'+B)(B'+C)$ e) $F(x,y,z)=1$ f) $F(x,y,z)=(xy+z)(y+xz)$

P3.23) Quale è la differenza tra forma canonica e forma standard? Quale delle due forme è preferibile quando le funzioni debbono essere realizzate con porte? Quale delle due forme si ottiene quando si traduce una funzione booleana da una tabella di verità?

BIBLIOGRAFIA

1. Boole, G., “An Investigation of the Laws of Thought”. New York: Dover Pub., 1954
2. Shannon, C. E., “ A Symbolic Analisys of Relay and Switching Circuits”, Trans. Of the AIEE, Vol 57 (1938), 713-23
3. Huntington, E. V., “Set of Independent Postulates for the Algebra of Logic”. Trans Am. Math. Soc., Vol. 5 (1904), 288-309
4. Birkhoff, G., and T. C. Bartee, “Modern Applied Algebra”, New York: MacCraw-Hill Book Co., 1970
5. Birkhoff, G., and S. Maclane, “A Survey of Modern Algebra”, 3rd ed. New York: The Macmillan Co., 1965
6. Hohn, F. E., Applied Boolean Algebra, 2nd ed. New York: The Macmillan Co., 1966
7. Whitesitt, J. E., “Boolean Algebra and its Applications”. Reading, Mass.: Addison-wesley pub. Co., 1961