Algebra di Boole

I circuiti logici sono componenti hardware che manipolano informazione binaria. I circuiti di base sono detti PORTE LOGICHE (*logical gate).*

Allo scopo di descrivere i comportamenti dei circuiti digitali si può usare una *algebra* (notazione matematica) che specifica l'operazione di ogni *gate* e permette di analizzare e sintetizzare (disegnare) il circuito.

Algebra di Boole

L'algebra che useremo è dovuta a Boole ed è detta ALGEBRA BOOLEANA. Le variabili di questa algebra sono binarie, possono assumere solo due valori (0,1).

Le *variabili* si indicano con le lettere A,B,C,X,Y,W,Z.

Le *operazioni base* sono AND (•), OR (+) ,NOT ($\bar{}$)

Possiamo definire questi operatori tramite la *tabella di verità*

Algebra di Boole

TABLE 2-1
Truth Tables for the Three Basic Logic Operations

AND			OR			NOT	
X	Y	$Z = X \cdot Y$	X	Y	$Z = X + Y$	X	$Z = \overline{X}$
0	0	0	0	0	0	0	1
0	1	0	0	1	1	1	0
1	0	0	1	0	1		
1	1	1	1	1	1		

Table 2-1 Truth Tables for the Three Basic Logical Operations

La Tabella di Verità permette di definire gli operatori AND,OR NOT.

Algebra di Boole

Altra interpretazione che possiamo dare è quella dei *CIRCUITI ELETTRONICI.*

Questi sistemi sono caratterizzati da grandezze fisiche (segnali) che assumono due gamme distinte di *livelli logici* H (alto) L (basso) ai quali è spontaneo far corrispondere i valori 0 e 1 detti *stati logici* la corrispondenza può essere secondo la *logica positiva* o secondo quella *negativa*.

Algebra di Boole

Sono stati individuati e costruiti circuiti elettronici c
realizzano le operazioni elementari, questi sono detti
PORTE LOGICHE

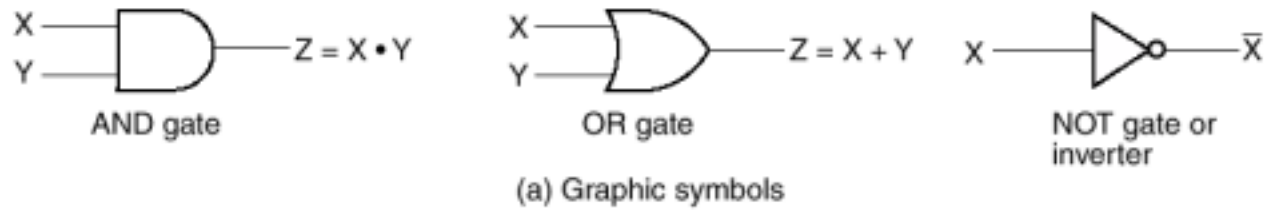
Le *porte logiche* sono circuiti che operano su più segnali di ingresso e producono un segnale di uscita.

Rispondono a due valori di range di tensioni che associamo ai valori logico 0 e 1.

Le porte logiche che rappresentano le operazioni AND OR e NOT sono di seguito riportate.

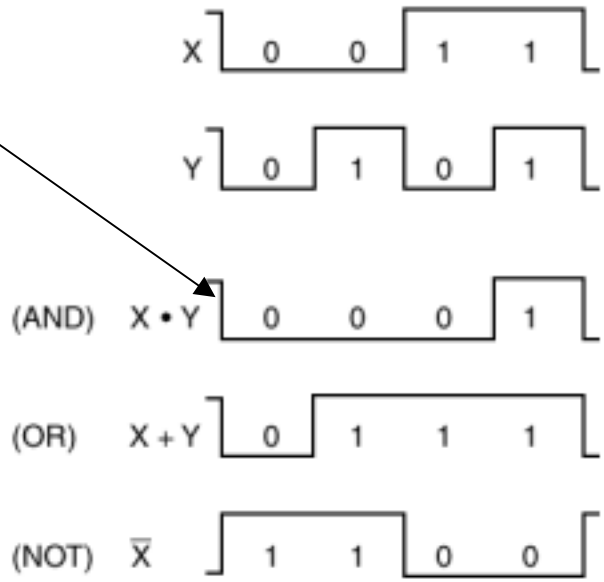
Algebra di Boole

Porte Logiche



(a) Graphic symbols

Transizione



(b) Timing diagram

Fig. 2-1 Digital Logic Gates

Le *funzioni booleane* sono quelle funzioni di variabile booleana che possono assumere soltanto i valori vero e falso (1,0).

Esempio:

$$F = X + \overline{Y}Z$$

Possiamo rappresentare la funzione usando la tabella di verità.

TABLE 2-2
Truth Table
for the Function F = X + $\overline{Y}$Z

X	Y	Z	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

Table 2-2 Truth Table for the Function F = X + $\overline{Y}$Z

Il circuito logico corrispondente è

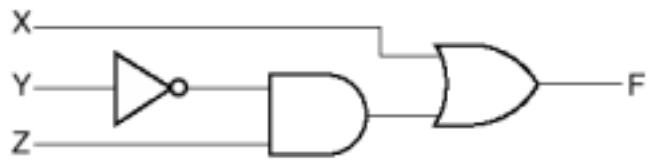


Fig. 2-3 Logic Circuit Diagram for $F = X + \overline{Y}Z$

TEOREMI

TABLE 2-3
Basic Identities of Boolean Algebra

1.	$X+0=X$	2.	$X\cdot 1=X$	
3.	$X+1=1$	4.	$X\cdot 0=0$	
5.	$X+X=X$	6.	$X\cdot X=X$	
7.	$X+\overline{X}=1$	8.	$X\cdot\overline{X}=0$	
9.	$\overline{\overline{X}}=X$			
10.	$X+Y=Y+X$	11.	$XY=YX$	Commutative
12.	$X+(Y+Z)=(X+Y)+Z$	13.	$X(YZ)=(XY)Z$	Associative
14.	$X(Y+Z)=XY+XZ$	15.	$X+YZ=(X+Y)(X+Z)$	Distributive
16.	$\overline{X+Y}=\overline{X}\cdot\overline{Y}$	17.	$\overline{X\cdot Y}=\overline{X}+\overline{Y}$	DeMorgan's

Table 2-3 Basic Identities of Boolean Algebra

Duali

Algebra di Boole

Identità di base

Il *principio di dualità* afferma che data una eguaglianza se ne ottiene un'altra sostituendo l'operatore AND con l'operatore OR, 1 con 0 e viceversa.

Le relazioni 1 e 2 sono duali.

I teoremi di DeMorgan sono molto importanti per ottenere il complemento di una espressione.

Dimostrazione del Teorema di DeMorgan tramite tabella di Verità.

TABLE 2-4
Truth Tables to Verify DeMorgan's Theorem

A) X	Y	X + Y	$\overline{X+Y}$	B) X	Y	$\overline{X}$	$\overline{Y}$	$\overline{X} \cdot \overline{Y}$
0	0	0	1	0	0	1	1	1
0	1	1	0	0	1	1	0	0
1	0	1	0	1	0	0	1	0
1	1	1	0	1	1	0	0	0

Table 2-4 Truth Tables to Verify DeMorgan's Theorem

Algebra di Boole

Il vantaggio dell'algebra di Boole sta nel fatto di permettere la semplificazione dei circuiti digitali.

$$F = \overline{X}YZ + \overline{X}Y\overline{Z} + XZ$$

POSSIAMO SEMPLIFICARE LA FUNZIONE

$$F = \overline{X}YZ + \overline{X}Y\overline{Z} + XZ$$

$$= \overline{X}Y(Z + \overline{Z}) + XZ \quad (\text{IDENTITA}'\ 14)$$

$$= \overline{X}Y \bullet 1 + XZ \quad (\ \ " \quad 7)$$

$$= \overline{X}Y + XZ \quad (\ \ " \quad 2)$$

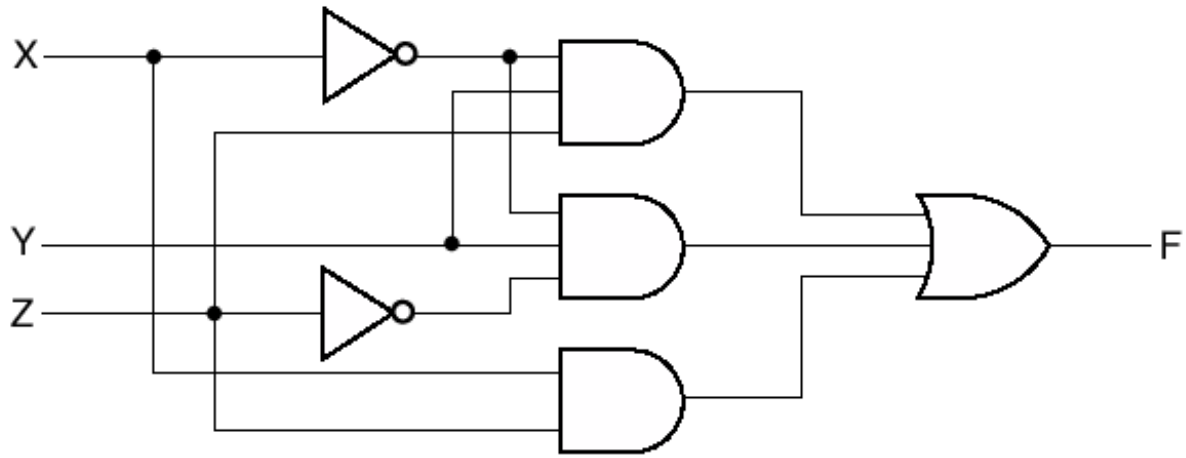
Le due funzioni sono equivalenti.

Hanno la stessa tabella di verità ma la seconda funzione è realizzabile con un circuito più semplice.

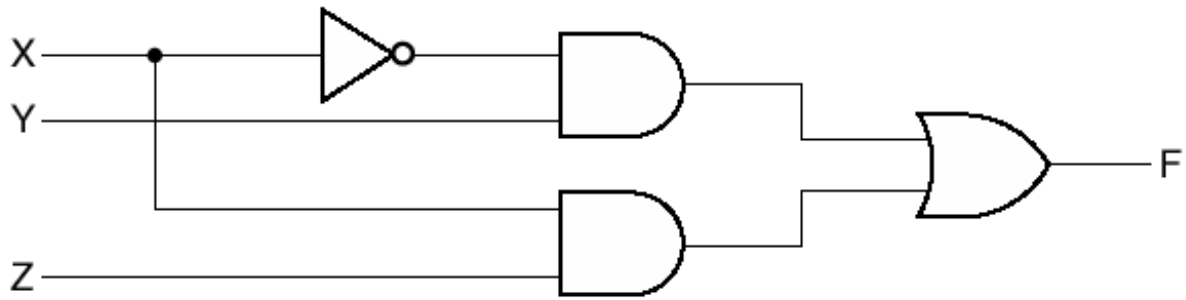
TABLE 2-5
Truth Table for Boolean Function

X	Y	Z	F
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

Table 2-5 Truth Table for Boolean Function



(a) $F = \bar{X}YZ + \bar{X}Y\bar{Z} + XZ$



(b) $F = \bar{X}Y + XZ$

Fig. 2-4 Implementation of Boolean Function with Gates

Il complemento di una funzione si ottiene applicando la seguente formula:

$$\overline{F}(a, b, c, \ldots, +, \bullet) = F(\overline{a}, \overline{b}, \overline{c}, \ldots, \bullet, +)$$

Algebra di Boole

consensus theorem

Il teorema del consensus permette di semplificare una espressione Booleana:

$$XY + \overline{X}Z + YZ = XY + \overline{X}Z$$

Come si vede salta il terzo termine,YZ, questo è ridondante e può essere eliminato. Si noti che Y e Z sono associati a X e $\overline{X}$.

Nei primi due termini e appaiono insieme nel termine che è eliminato.

Algebra di Boole

consensus theorem

Dimostrazione:

$$XY + \overline{X}Z + YZ = XY + \overline{X}Z = XY + \overline{X}Z + YZ(X + \overline{X})$$

$$= XY + \overline{X}Z + XYZ + \overline{X}YZ$$

$$= XY + XYZ + \overline{X}Z + \overline{X}YZ$$

$$= XY(1 + Z) + \overline{X}Z(1 + Y)$$

$$= XY + \overline{X}Z.$$

DUALE:

$$(X + Y)(\overline{X} + Z)(Y + Z) = (X + Y)(\overline{X} + Z).$$

Analisi di reti Combinatorie

Una rete combinatoria è una rete logica con *n ingressi m uscite.*

Le uscite sono funzione degli ingressi, ma non del tempo: cambiano gli ingressi ed immediatamente cambiano le uscite *(ovviamente è un modello).*

In un circuito logico le porte logiche viste sono combinate tra loro formando un *circuito combinatorio.*

Le variabili sono combinate tramite le operazioni logiche.

Forme canoniche

Abbiamo visto che è possibile esprimere le funzioni booleane tramite la sua espressione analitica oppure tramite la tabella di verità.

Le funzioni booleane possono essere scritte in vari modi ma vi sono delle espressioni che vengono considerate standard.

Per far ciò definiamo i mintermini e i maxtermini

Algebra di Boole

Considerando una riga della tabella di verità si definisce *mintermine* il prodotto delle variabili booleane relative a tal riga prese in forma diretta o complementata a seconda se assumono valore 1 o 0.

Si definisca *maxtermine* la somma delle variabili booleane prese in forma diretta o negata a seconda se assumono valore 0 o 1.

Con n variabili abbiamo 2^n mintermini e maxtermini

Forme Standard

TABLE 2-6
Minterms for Three Variables

X	Y	Z	Product Term	Symbol
0	0	0	$\overline{X}\,\overline{Y}\,\overline{Z}$	m_0
0	0	1	$\overline{X}\,\overline{Y}Z$	m_1
0	1	0	$\overline{X}Y\overline{Z}$	m_2
0	1	1	$\overline{X}YZ$	m_3
1	0	0	$X\overline{Y}\,\overline{Z}$	m_4
1	0	1	$X\overline{Y}Z$	m_5
1	1	0	$XY\overline{Z}$	m_6
1	1	1	XYZ	m_7

Table 2-6 Minterr

TABLE 2-7
Maxterms for Three Variables

X	Y	Z	Sum Term	Symbol
0	0	0	$X+Y+Z$	M_0
0	0	1	$X+Y+\overline{Z}$	M_1
0	1	0	$X+\overline{Y}+Z$	M_2
0	1	1	$X+\overline{Y}+\overline{Z}$	M_3
1	0	0	$\overline{X}+Y+Z$	M_4
1	0	1	$\overline{X}+Y+\overline{Z}$	M_5
1	1	0	$\overline{X}+\overline{Y}+Z$	M_6
1	1	1	$\overline{X}+\overline{Y}+\overline{Z}$	M_7

Table 2-7 Maxte

Il pedice j del mintermine è dato dal'equivalente decimale del numero binario che si ottiene dando valore 1 alla variabile in forma diretta, 0 per quella complementata. Per il maxtermine 1 per complemetata, 0 per diretta.

SOMMA DI PRODOTTI

Possiamo ottenere la forma analitica di una funzione a partire dalla tabella di verità nel seguente modo:

1. Si individuano le righe per cui F ha valore 1;
2. Si scrivono tanti prodotti quante sono le righe individuate
3. Ogni prodotto è il mintermine relativo alla riga
4. Si sommano i prodotti.

SOMMA DI PRODOTTI

a	b	F	
0	0	0	
0	1	1	←
1	0	1	←
1	1	0	

$$F = a\bar{b} + \bar{a}b$$

PRODOTTO DI SOMME

Possiamo ottenere la forma analitica di una funzione a partire dalla tabella di verità nel seguente modo:

1. Si individuano le righe per cui F ha valore 0;
2. Si scrivono tante somme quante sono le righe individuate
3. Ogni somma è il maxtermine relativo alla riga
4. Si effettua il prodotto delle somme.

Forme Standard

Il vantaggio delle forme standar è quello di permettere la realizzazione delle funzioni con circuiti a due livelli: AND-OR oppure OR-AND

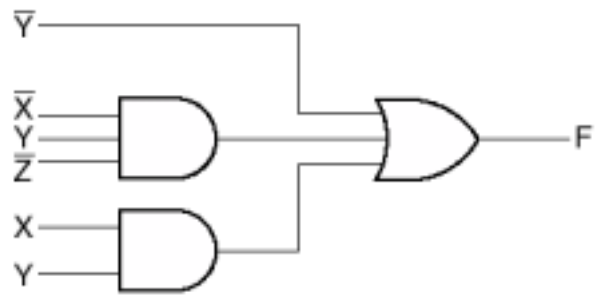


Fig. 2-5 Sum-of-Products Implementation

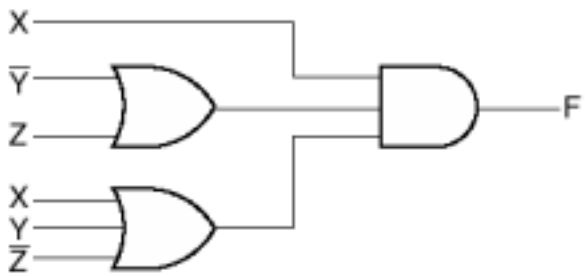


Fig. 2-7 Product-of-Sums Implementation

Mappe di KARNAUGH

Una funzione booleana può essere rappresentata, oltre che con la tabella di verità, con le mappe di KARNOUGH.

Questa mappa è costituita da quadrati chiamati *celle.*

Due lati del quadrato sono contrassegnati dai valori delle variabili. Le colonne e le righe presentano un ordine ciclico in modo che due celle adiacenti differiscano tra loro solo per il valore di un letterale.

K-Mappe

Le celle corrispondono ai *mintermini* di una funzione ad *n variabili*.

Tutta la mappa meno la cella *i* corrisponde al maxtermine *i*.

Queste mappe sono utili per rappresentare le funzioni in forma standard, se si usano i mintermini si devono considerare le celle contenenti 1, se si usano i maxtermini quelle contenenti 0.

Queste mappe sono molto utili per la semplificazione di una funzione.

Mappe di KARNAUGH

Mappa per Due Variabili

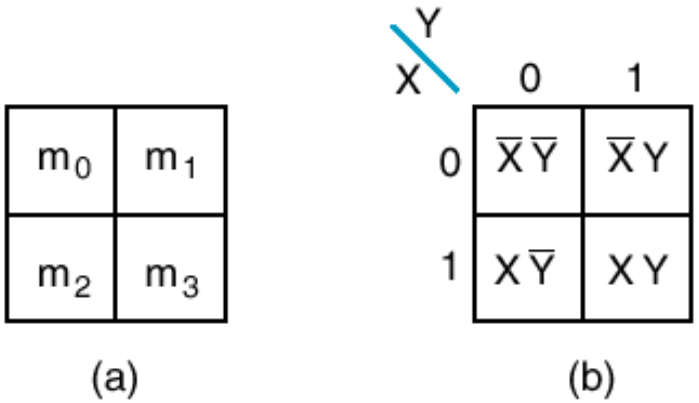


Fig. 2-8 Two-Variable Map

Rappresentazione di funzioni nella MAPPA

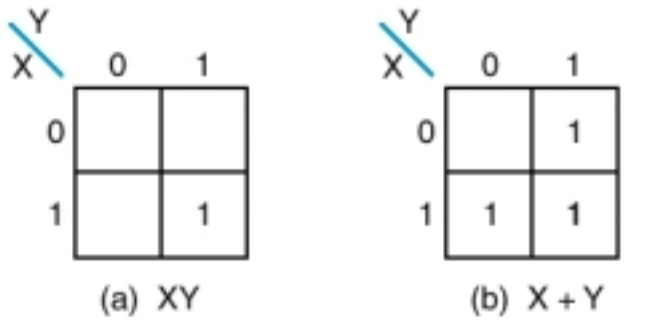


Fig. 2-9 Representation of Functions in the Map

Mappe di KARNAUGH

Mappa per Tre Variabili

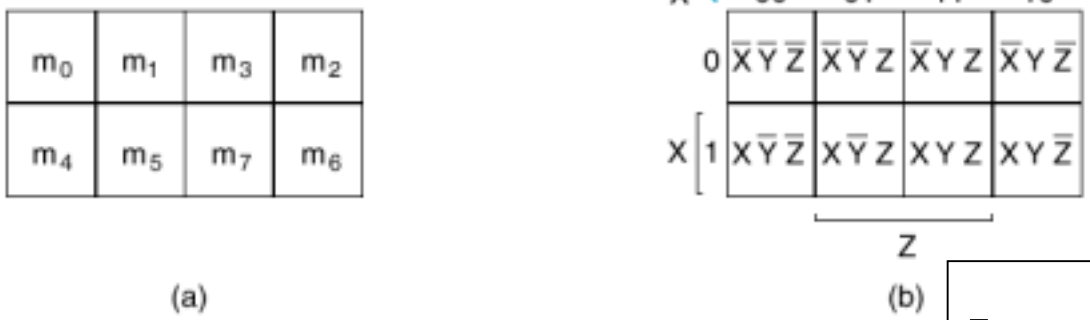


Fig. 2-10 Three-Variable Map

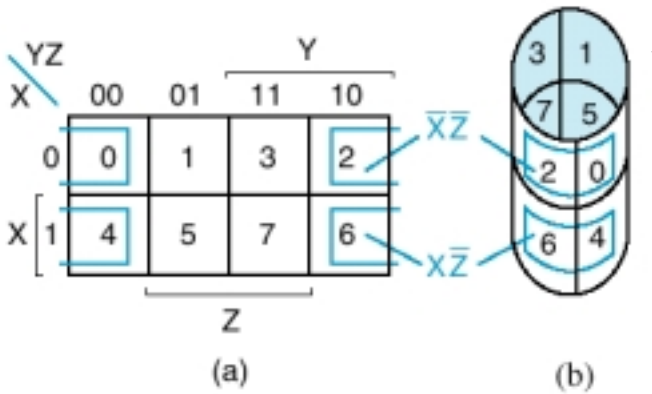


Fig. 2-12 Three-Variable Map: Flat and on a Cylinder to Show Adjacent Squares

La mappa a tre variabili è lo sviluppo nel piano di un cilindro.

Le celle 0-2, 4-6 sono adiacenti.

Mappe di KARNAUGH

Mappa per Quattro Variabili

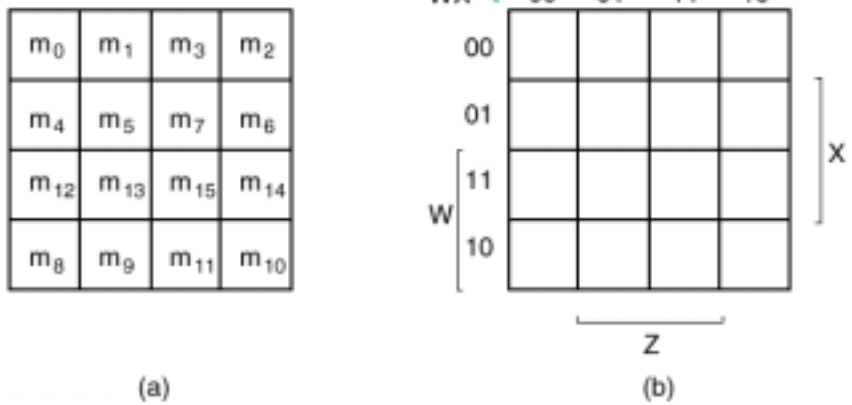


Fig. 2-17 Four-Variable Map

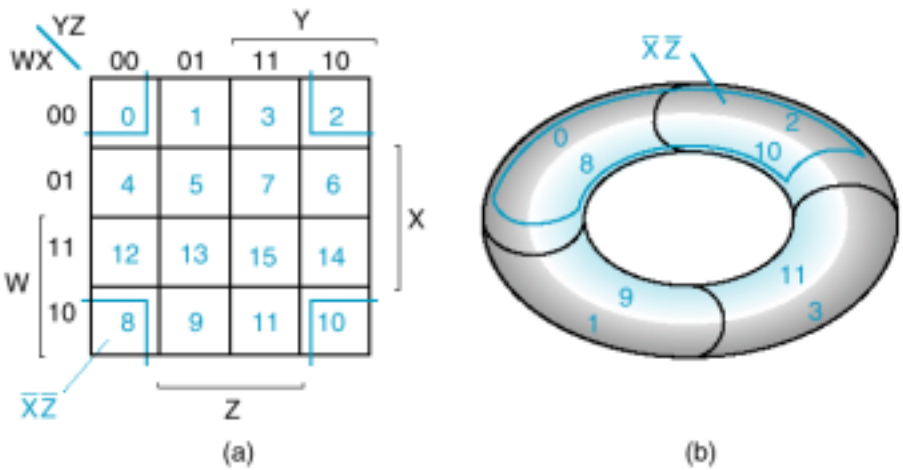


Fig. 2-18 Four-Variable Map: Flat and on a Torus to Show Adjacencies

La mappa a quattro variabili è lo sviluppo nel piano di un toroide.

K-Mappe

Le K-mappe permettono la semplificazione delle funzioni booleane. Supponiamo di avere la funzione espressa come somma di mintermini : $F=\sum(4,5,6,12,13)$

Nella K-mappa la rappresentazione si ottiene mettendo 1 nella cella corrispondente al mintermine.

a b \ c d	00	01	11	10
00				
01	1	1		1
11	1	1		
10				

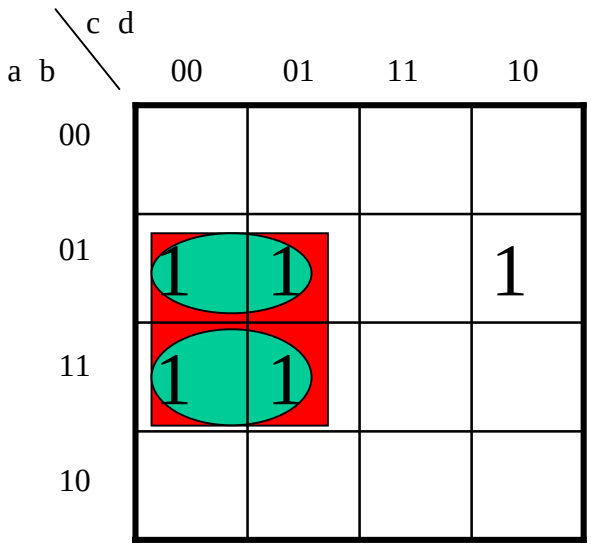
K-Mappe

I mintermini 4 e 5 sono adiacenti, risulta che: $\bar{a}b\bar{c}\bar{d} + \bar{a}b\bar{c}d = \bar{a}b\bar{c}$

Lo stesso per 12 e 13: $ab\bar{c}\bar{d} + ab\bar{c}d = ab\bar{c}$

Abbiamo trasformato la somma di due prodotti di 4 variabili in un prodotto di di 3 variabili, è saltato un letterale.

Queste coppie di caselle adiacenti in cui si trova 1 costituiscono un *accoppiamento a due*

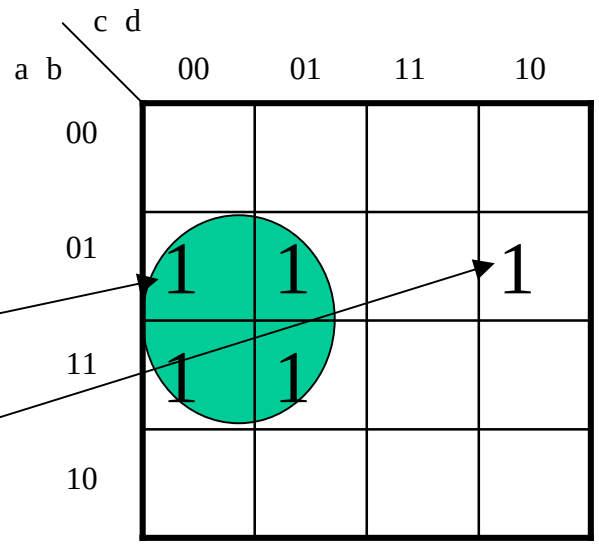


Questi due accoppiamenti sono contigui sommandoli salta un altro letterale:

accoppiamento a quattro

$$\bar{a}b\bar{c} + ab\bar{c} = b\bar{c}$$

$$F = b\bar{c} + \bar{a}bc\bar{d}$$



K-Mappe

E’ possibile generare accoppiamenti a 2,4,….,2^i

In un accoppiamento 2^i si perdono *i* letterali.

La ricerca deve partire dagli accoppiamenti più grandi si perde un maggior numero di letterali.

In una tabella a 4 variabili gli accoppiamenti che si possono trovare sono a 2,4,8,16.

ESERCIZIO:

Semplificare le seguenti funzioni:

F= $\sum$(0,1,2,4,5,6,8,9,12,13,14);

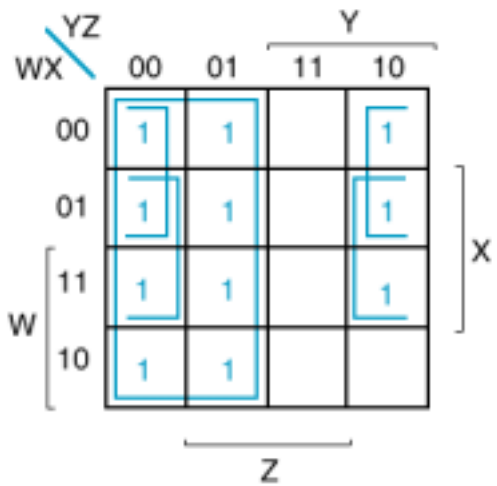


Fig. 2-19 Map for Example 2-5: $F = \overline{Y} + \overline{W}\overline{Z} + X\overline{Z}$

Somma di prodotti

Se vogliamo scrivere la funzione come prodotto di somme , usando i maxtermini.

Dobbiamo considerare le caselle conteneti 0, quindi accoppiare queste caselle, ottenendo la $\overline{F}$

Complementando otteniamo la F come prodotto di somme.

Supponiamo di avere la seguente funzione:

$$F(A,B,C,D) = \sum(0,1,2,5,8,9,10)$$

Rappresentiamola tramite la mappa di Karnaugh:

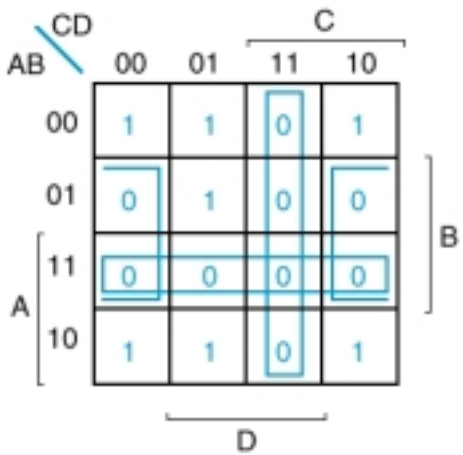


Fig. 2-24 Map for Example 2-10: $F = (\bar{A} + \bar{B})(\bar{C} + \bar{D})(\bar{B} + D)$

Somma di prodotti

Se si desidera esprimere la funzione come somma di prodotti, si devono considerare le *celle* contenenti 0 trovare gli accoppiamenti, ottenendo così la $\overline{F}$

$$\overline{F} = AB + CD + B\overline{D}$$

Utilizzando la formula

$$\overline{F}(a, b, c, ..., +, \bullet) = F(\overline{a}, \overline{b}, \overline{c}, ..., \bullet, +)$$

Si ottiene

$$F = (\overline{A} + \overline{B})(\overline{C} + \overline{D})(\overline{B} + D)$$

Primi Implicati

Un prodotto di termini *implica* una funzione quando per ogni combinazione delle variabili del termine per cui esso assume valore 1 anche la funzione f assume valore 1.

Si dice *Primo Implicante* (P) della funzione f un prodotto di termini che implica f e tale che eliminando un qualunque letterale il prodotto rimanente non implica più f.

Se un mintermine di una funzione è incluso solo in un primo implicante quest'ultimo si dice primo *implicante essenziale*.

Primi Implicati

Si noti che gli *accoppiamenti* costituiscono *primi implicanti* purché non siano *completamente interni* ad accoppiamenti di ordine superiore.

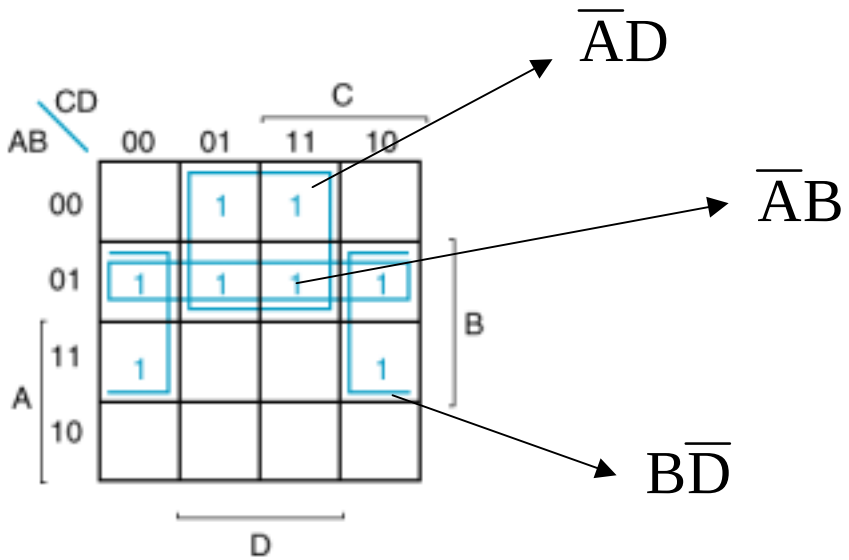


Fig. 2-21 Prime Implicants for Example 2-7: $\overline{A}D$, $B\overline{D}$, and $\overline{A}B$

Primi Implicati

PRIMI IMPLICANTI ESSENZIALI

SONO: $\overline{A}D$ $B\overline{D}$

LA FUNZIONE F PUO' ESSERE SCRITTA COME SOMMA DEI PRIMI IMPLICANTI ESSENZIALI:

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

$\overline{A}B$ NON E' UN PRIMO IMPLICANTE ESSENZIALE PERCHE' NON CONTIENE MINTERMINI COPERTI ESCLUSIVAMENTE DA ESSO.

Primi Implicati

ESERCIZIO

Determinare i primi implicanti della funzione $F=\sum(0,1,2,4,5,10,11,13,15)$.

Dire quali sono Essenziali.

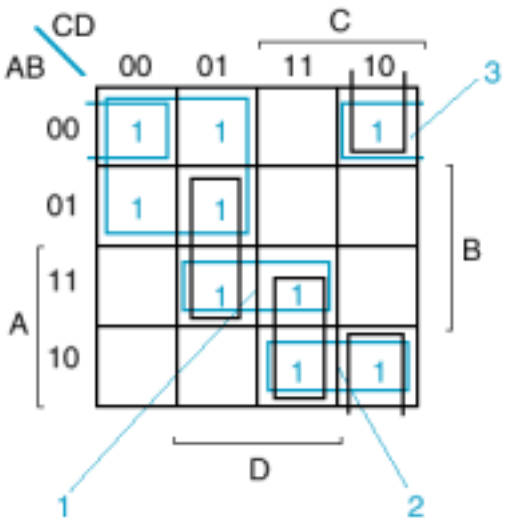


Fig. 2-23 Map for Example 2-9

Don't Care Condictions

FUNZIONI NON COMPLETAMENTE SPECIFICATE

Vi possono essere delle funzioni non *completamente specificate*

Funzioni in cui , in corrispondenza a certi valori di ingresso, non si vuole un fissato valore di uscita, ovvero qualunque valore dell'uscita è accettabile.

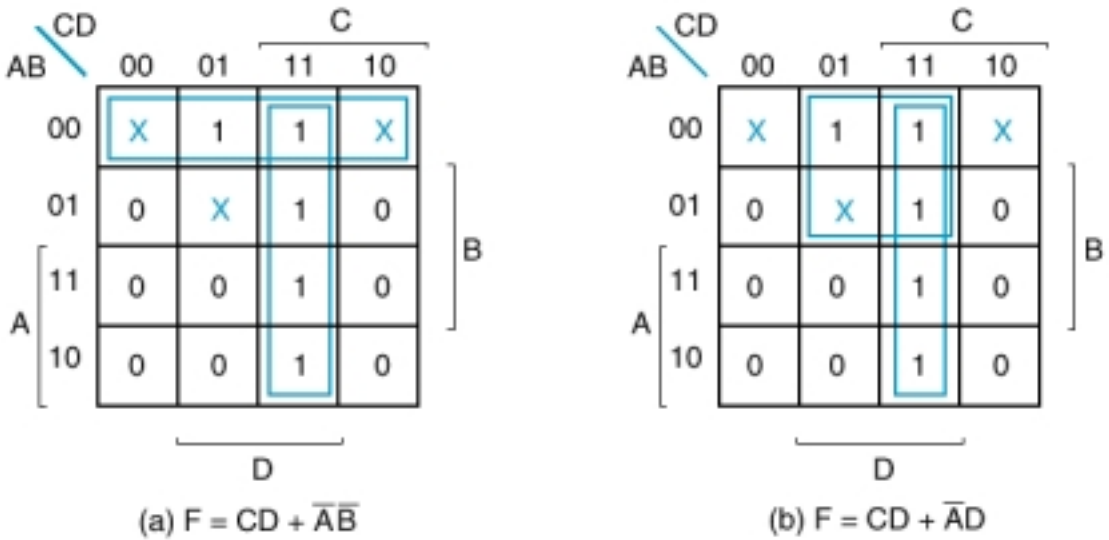
Nella mappa si indicano con *d* (*don't care*) oppure X oppure – .

In fase di semplificazione di una funzione ad essi si può assegnare valore 1 oppure 0 a seconda se permettono di ottenere accoppiamenti più grandi e quindi maggiori semplificazioni.

Don't Care Condictions

FUNZIONI NON COMPLETAMENTE SPECIFICATE

Esempio:



(a) $F = CD + \overline{A}\,\overline{B}$ (b) $F = CD + \overline{A}D$

Fig. 2-25 Example with Don't-Care Conditions

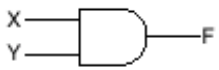
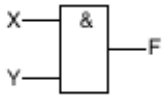
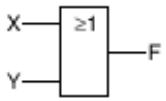
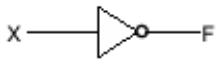
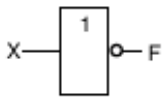
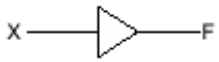
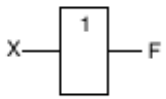
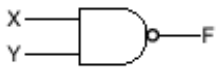
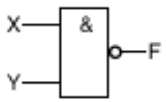
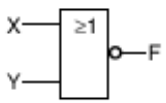
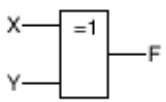
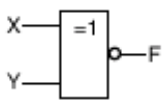
NAND e NOR Gates

Possiamo esprimere una funzione Booleana tramite i *gate* AND, OR e NOT.

Vi sono però altri *gate* di particolare interesse e che da soli permettono di rappresentare la funzione come *somma di prodotti* (*NAND*) oppure come *prodotto di somme* (*NOR*).

Gates

Graphics Symbols

Name	Distinctive shape	Rectangular shape	Algebraic equation	Truth table
AND			$F = XY$	X Y | F 0 0 | 0 0 1 | 0 1 0 | 0 1 1 | 1
OR			$F = X + Y$	X Y | F 0 0 | 0 0 1 | 1 1 0 | 1 1 1 | 1
NOT (inverter)			$F = \overline{X}$	X | F 0 | 1 1 | 0
Buffer			$F = X$	X | F 0 | 0 1 | 1
NAND			$F = \overline{X \cdot Y}$	X Y | F 0 0 | 1 0 1 | 1 1 0 | 1 1 1 | 0
NOR			$F = \overline{X + Y}$	X Y | F 0 0 | 1 0 1 | 0 1 0 | 0 1 1 | 0
Exclusive–OR (XOR)			$F = X\overline{Y} + \overline{X}Y$ $= X \oplus Y$	X Y | F 0 0 | 0 0 1 | 1 1 0 | 1 1 1 | 0
Exclusive–NOR (XNOR)			$F = XY + \overline{X}\overline{Y}$ $= \overline{X \oplus Y}$	X Y | F 0 0 | 1 0 1 | 0 1 0 | 0 1 1 | 1

NAND

Il gate *NAND (/)* è detto universale perché perché ogni circuito digitale può essere implementato con i soli NAND.

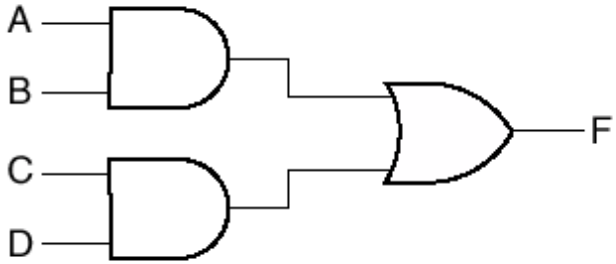
Una funzione espressa come somma di prodotti può essere rappresentata mediante dei NAND sostituendo a titti gli AND e OR dei NAND, e negando le variabili che compaiono da sole come addendi.

Supponiamo di avere la seguente funzione *somma di prodotti*:

$$F=AB+CD$$

Il circuito logico che la realizza usando AND e OR:

Gate NAND

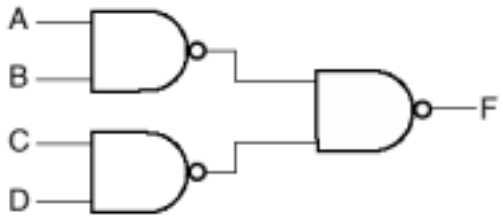


Applicando DeMorgan alla funzione F si ottiene: $F = \overline{\overline{AB} \cdot \overline{CD}}$

Il NAND (/) è il NOT di un AND quindi possiamo scrivere

$$F = (A/B)/(C/D)$$

Il circuito con i soli gate NAND è:



NOR GATE

Il gate *NOR* ($\downarrow$) è detto universale perché perché ogni circuito digitale può essere implementato con i soli NOR.

Una funzione espressa come prodotto di somme può essere rappresentata mediante dei NOR sostituendo a tutti gli AND e OR dei NOR, e negando le variabili che compaiono da sole come fattori.

Exclusive-OR GATE (XOR, ⊕)

L'Or Esclusivo il cui simbolo è ⊕ è l'operazione logica che implementa la funzione

$$X \oplus Y = X\overline{Y} + \overline{X}Y$$

La funzione è vera se le variabili non sono contemporaneamente vere(1) o contemporaneamente False (0).

L'exclusive-NOR è il complemento del XOR:

$$\overline{X \oplus Y} = XY + \overline{X}\,\overline{Y}$$

Funzione dispari

L'OXR di tre o più variabili ha valore 1 solo se le variabili che assumono valore 1 sono in numero dispari, per tal motivo è detta *funzione dispari*

Considerando i mintermini che hanno un numero dispari di variabili in forma diretta, si vede che nelle K-mappe essi non sono adiacenti e differiscono al più per due letterali.

Si dice che hanno distanza due l'un dall'altro.

L'*funzione pari* è il complemento della dispari.

Odd Function

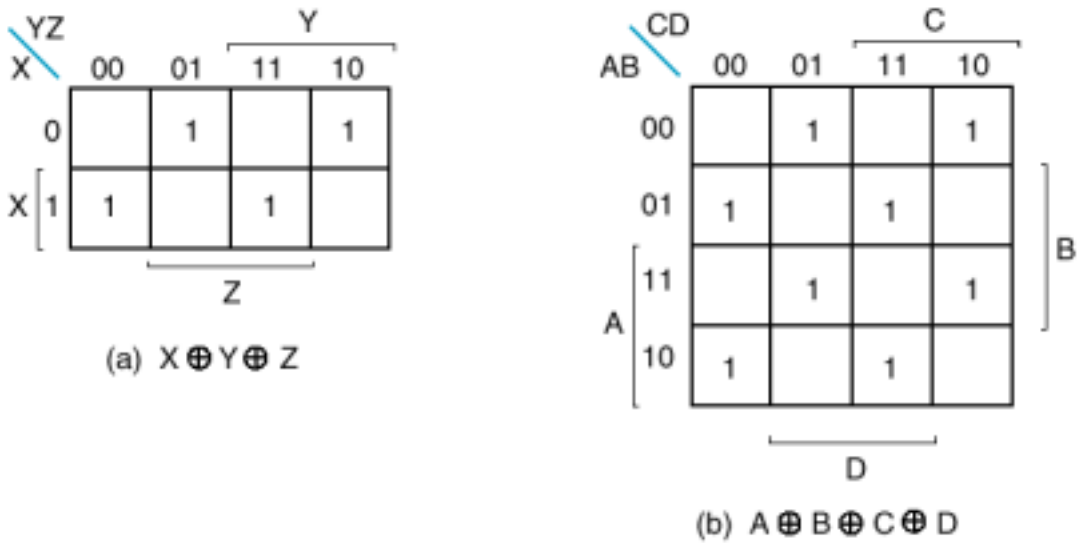


Fig. 2-38 Maps for Multiple-Variable Odd Functions

Odd Function

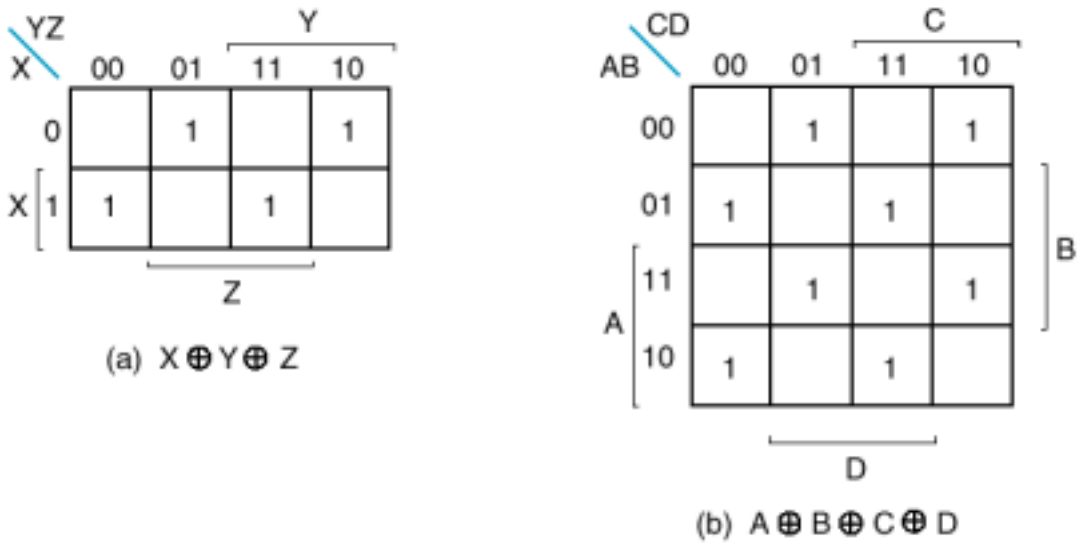


Fig. 2-38 Maps for Multiple-Variable Odd Functions

Parity generation e Checking

Il circuito che permette l'aggiunta del *bit di parità* prende il nome di *parity genereto*r. Il circuito che ne permette il controllo e il *parity checher*.Usando la parità pari, avremo un errore se il parity checker da in uscita uno (perché un XOR di più variabili è 1 se il numero di bit 1 è dispari).

I circuiti sono:

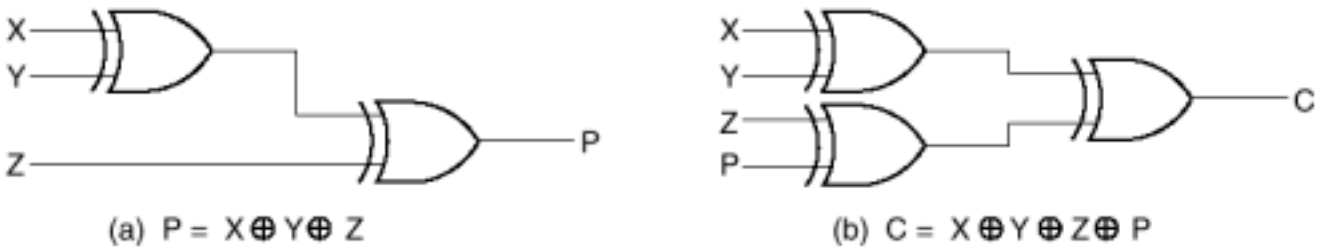


Fig. 2-39 Multiple-Input Odd Functions

Circuiti Integrati

I circuiti digitali sono costruiti come circuiti integrati (IC) scristalli di semiconduttori al silicio, detti informalmente chip, contenenti i componenti elettronici per i gate digitali.

I gate sono interconnessi sul chip per formare il circuito integrato.

A seconda del numero di gate che possono essere messi su un chip variano da pochi a migliaia di milioni.

Circuiti Integrati

A seconda del numero parleremo di:

Small-scale integrated (SSI): diversi gate;

Medium-scale integrated (MSI): 10-100;

Large-scale integrated (LSI) : da 100 a poche migliaia;

Very large-scale integrated (VLSI): da parecchie migliaia a 100 milioni.