ALGEBRA BOOLEANA:

VARIABILI E FUNZIONI LOGICHE

Andrea Bobbio
Anno Accademico 2000-2001

Calcolatore come rete logica

Il calcolatore può essere visto come una *rete logica* cioè come un insieme di dispositivi chiamati *porte logiche* opportunamente connessi.

Le *porte logiche* sono dispositivi capaci di eseguire operazioni logiche su *segnali binari*.

I segnali binari sono livelli di tensione.

Il valore esatto della tensione del segnale non è significativo: conta l'appartenenza ad un livello contrassegnato **alto** e ad un livello contrassegnato **basso**.

Questi livelli sono identificati tramite una coppia di simboli:

0	1
Low	High
False	True
Open	Close

Algebra Booleana

Le tecniche di composizione delle porte logiche in una rete sono derivate da una particolare algebra operante su variabili binarie e chiamata *Algebra Booleana* (o *Switching Algebra*).

L'algebra Booleana prende il nome dal matematico inglese George Boole (1815-1864) autore del testo *The mathematical analysis of logic*.

A lui è legato lo sviluppo della logica simbolica e degli operatori binari. Nel 1938 Shannon ha dimostrato come l'algebra booleana potesse essere presa a fondamento per la progettazione di circuiti logici digitali.

Elementi dell'Algebra Booleana

Vengono definiti i seguenti concetti:

- ◇ *variabili booleane*
- ◇ *operatori booleani*
- ◇ *funzioni booleane*
- ◇ *porte logiche*
- ◇ *circuiti logici*
  - *combinatori*
  - *sequenziali*

Variabili Booleane

Una variabile booleana è una variabile binaria che può assumere esclusivamente due valori logici che saranno denotati con 0 e 1.

Se x è una variabile booleana, vale quindi la seguente definizione formale:

$$\begin{array}{lcl} x = 0 & \text{se} & x \neq 1 \\ x = 1 & \text{se} & x \neq 0 \end{array}$$

Operatori Booleani

Si definiscono gli operatori booleani o logici fondamentali:

NOT	Negazione Logica
AND	Prodotto Logico
OR	Somma Logica

Negazione o Complementazione

Definizione informale

Trattasi di una operazione unaria che restituisce il valore logico opposto a quello della variabile di ingresso.

Rappresentazione come operatore

Per rappresentare il complemento di una variabile x vengono usate varie notazioni. Fra le più comunemente usate ricordiamo:

$$not(x)$$
$$\overline{x}$$
$$x'$$
$$-\,x$$

Negazione o Complementazione / 2

Rappresentazione dell'operazione $not(x)$ con la tavola della verità:

x	$not(x)$
0	1
1	0

Proprietà

$$not(not(x)) = x$$

$$\overline{\overline{1}} = 1$$

$$\overline{\overline{\overline{0}}} = \overline{\overline{1}} = \overline{0} = 1$$

Prodotto Logico (AND)

Definizione informale

L'operazione di prodotto logico fra due (o più) variabili fornisce il valore logico 1 se e solo se *tutte* le variabili assumono valore logico 1.

Rappresentazione come operatore

Per rappresentare il prodotto logico di due variabili x e y si usa la notazione:

$$x \, and \, y$$
$$x \cdot y$$
$$x \, y$$

Prodotto Logico (AND) / 2

Rappresentazione dell'operazione $x \cdot y$ con la tavola della verità:

x	y	$x \cdot y$
0	0	0
0	1	0
1	0	0
1	1	1

Proprietà

$$x \cdot 0 = 0$$

$$x \cdot 1 = x$$

$$x \cdot x = x$$

$$x \cdot \overline{x} = 0$$

Somma Logica (OR)

Definizione informale

L'operazione di somma logica fra due (o più) variabili fornisce il valore logico 1 se e solo se *almeno una* delle variabili assume valore logico 1.

Rappresentazione come operatore

Per rappresentare la somma logica di due variabili x e y si usa la notazione:

$$x \mathit{\ or\ } y$$
$$x + y$$

Somma Logica (OR) / 2

Rappresentazione dell'operazione $x + y$ con la tavola della verità:

x	y	$x + y$
0	0	0
0	1	1
1	0	1
1	1	1

Proprietà

$$x + 0 = x$$

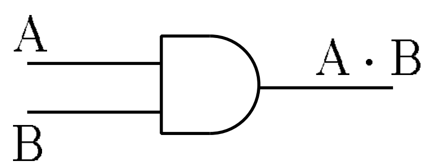
$$x + 1 = 1$$

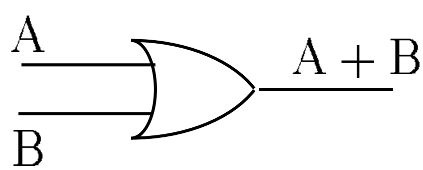
$$x + x = x$$

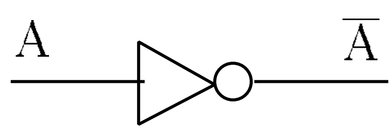
$$x + \overline{x} = 1$$

Porte Logiche

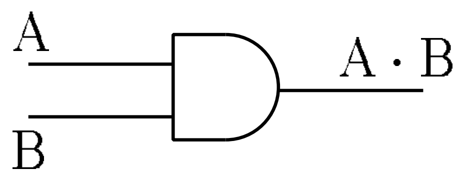
Le *porte logiche* sono dispositivi elettronici capaci di eseguire operazioni logiche su *variabili booleane*.





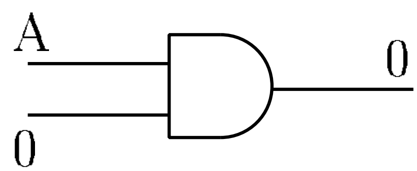


Porta AND

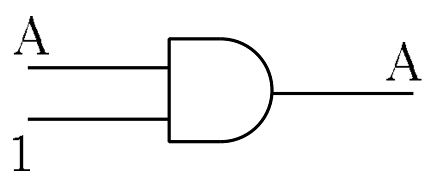


Alcune proprietà della porta *AND*.

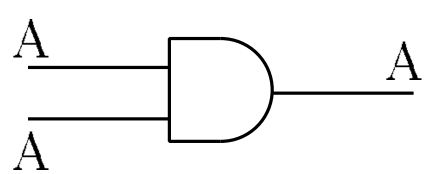
$$A \cdot 0 = 0$$



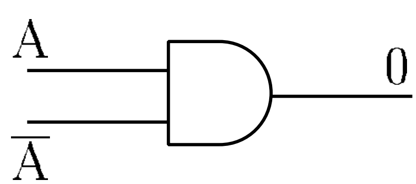
$$A \cdot 1 = A$$



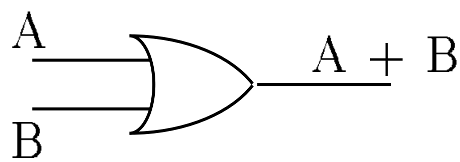
$$A \cdot A = A$$



$$A \cdot \overline{A} = 0$$

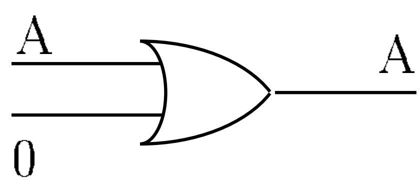


Porta OR

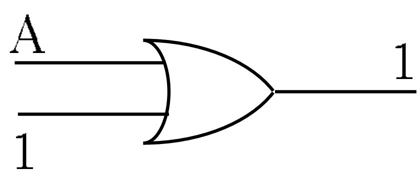


Alcune proprietà della porta *OR*.

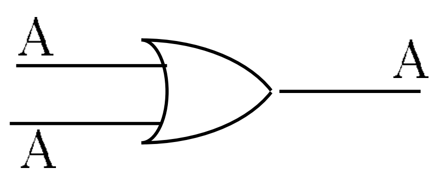
$$A + 0 = A$$



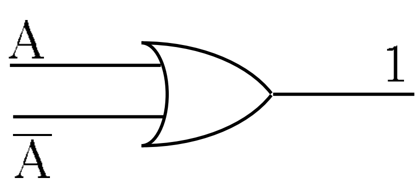
$$A + 1 = 1$$



$$A + A = A$$



$$A + \overline{A} = 1$$



Proprietà dell'algebra Booleana

Le proprietà degli operatori logici NOT, AND e OR, permettono di stabilire le seguenti proprietà.

- *Idempotenza*

  $x + x = x$

  $x \cdot x = x$

- *Elemento nullo (forcing function)*

  $x + 1 = 1$

  $x \cdot 0 = 0$

- *Proprietà Commutativa*

  $x + y = y + x$

  $x \cdot y = y \cdot x$

- *Proprietà Associativa*

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

  $x \cdot (y \cdot z) = (x \cdot y) \cdot z = x \cdot y \cdot z$

Reciprocità dei Teoremi dell'algebra Booleana

Le proprietà che valgono per l'operatore $+$ valgono anche per l'operatore $\cdot$ purchè si scambino gli 1 con gli 0 (e viceversa).

Teoremi dell'algebra Booleana

◇ *Distributività*

La proprietà distributiva vale sia rispetto alla somma di prodotti (come nell'algebra ordinaria) che rispetto al prodotto di somme.

$$x \cdot y + x \cdot z = x \cdot (y + z)$$

$$(x + y) \cdot (x + z) = x + y \cdot z$$

Teoremi dell'algebra Booleana

Verifica dei teoremi di *distributività* mediante la tavola della verità.

x	y	z	xy	xz	$xy+xz$	$y+z$	$x(y+z)$
0	0	0	0	0	0	0	0
0	0	1	0	0	0	1	0
0	1	0	0	0	0	1	0
0	1	1	0	0	0	1	0
1	0	0	0	0	0	0	0
1	0	1	0	1	1	1	1
1	1	0	1	0	1	1	1
1	1	1	1	1	1	1	1

x	y	z	yz	$x+yz$	$x+y$	$x+z$	$(x+y)(x+z)$
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

Teoremi dell'algebra Booleana

◇ *Assorbimento*

Il teorema dell'assorbimento può assumere varie forme.

$$x + (x \cdot y) = x$$

$$x \cdot (x + y) = x$$

$$(x + \overline{y}) \cdot y = x \cdot y$$

$$x \cdot \overline{y} + y = x + y$$

Teoremi dell'algebra Booleana

◇ *Teoremi di De Morgan*

$$\overline{x \cdot y} = \overline{x} + \overline{y}$$

$$\overline{x + y} = \overline{x} \cdot \overline{y}$$

Teoremi dell'algebra Booleana

◇ *Legge di Cancellazione*

Nell'algebra booleana non vale la legge di cancellazione

Dall'espressione:

$$x + y = x + z$$

non è possibile dedurre:

$$y = z$$

Teoremi dell'algebra Booleana

◇ *Legge di Cancellazione*

Si dimostra la non validità della legge di cancellazione mediante la tavola della verità.

x	y	z	$x+y$	$x+z$	$x+y=x+z$	$y=z$
0	0	0	0	0	1	1
0	0	1	0	1	0	0
0	1	0	1	0	0	0
0	1	1	1	1	1	1
1	0	0	1	1	1	1
1	0	1	1	1	1	0
1	1	0	1	1	1	0
1	1	1	1	1	1	1

Algebra degli Insiemi

L'*Algebra degli insiemi* (o *Teoria degli insiemi*) è formalmente identica all'algebra booleana a condizione che:

- $\Rightarrow$ Le variabili logiche siano interpretate come possibili sottoinsiemi di un insieme universo U.
- $\Rightarrow$ Il prodotto logico $\cdot$ (AND) sia interpretato come l'operazione di intersezione fra insiemi.
- $\Rightarrow$ La somma logica $+$ (OR) sia interpretata come l'operazione di unione fra insiemi.
- $\Rightarrow$ Il valore 1 (elemento neutro rispetto a $\cdot$) sia sostituito dall'insieme universo U (elemento neutro rispetto all'intersezione).
- $\Rightarrow$ Il valore 0 (elemento neutro rispetto a $+$) sia sostituito dall'insieme vuoto Φ (elemento neutro rispetto all'unione).

È quindi possibile dimostrare le proprietà e i teoremi dell'algebra booleana ricorrendo alla rappresentazione degli insiemi mediante *diagrammi di Venn*.

Operazione di nand logico

Definizione informale

L'operazione di *nand* logico è l'operazione negata dell'operazione and.

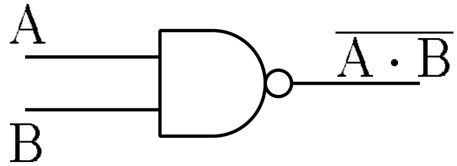
Il simbolo *nand* è una contrazione di *not and*.

Quindi l'operazione di nand logico fra due (o più) variabili fornisce il valore logico 1 se *almeno una* delle variabili assume il valore logico 0.

Rappresentazione come operatore

Per rappresentare il nand logico non esiste un simbolo specifico.

$$x \ nand \ y$$



Operazione nand logico / 2

Rappresentazione dell'operazione x *nand* y con la tavola della verità:

x	y	x *nand* y
0	0	1
0	1	1
1	0	1
1	1	0

Proprietà

$$x \; nand \; 0 = 1$$

$$x \; nand \; 1 = \overline{x}$$

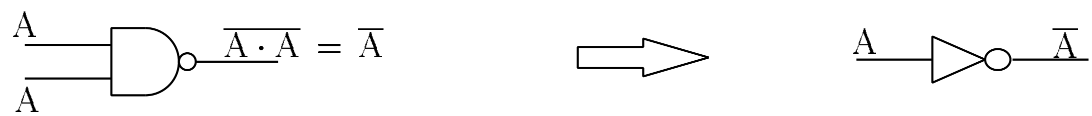
$$x \; nand \; x = \overline{x}$$

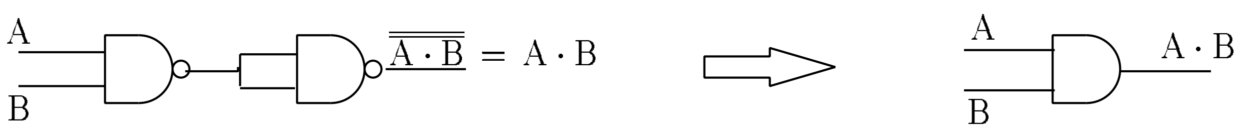
$$x \; nand \; \overline{x} = 1$$

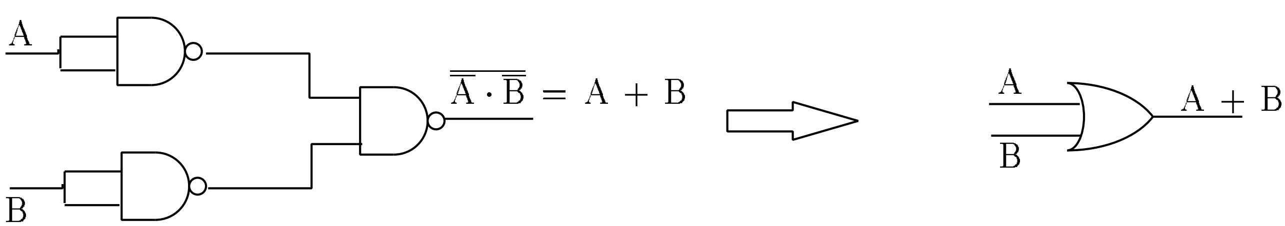
$$x \; nand \; y = \overline{x \cdot y}$$

Porta NAND

Con il solo operatore *NAND*, si possono rappresentare gli operatori *NOT*, *AND* e *OR*.







Operazione di nor logico

Definizione informale

L'operazione di *nor* logico è l'operazione negata dell'operazione or.

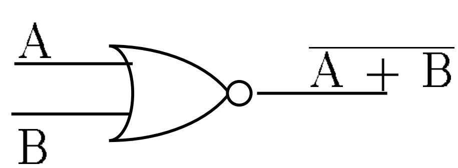
Il simbolo *nor* è una contrazione di *not or*.

Quindi l'operazione di nor logico fra due (o più) variabili fornisce il valore logico 1 se *nessuna* delle variabili assume il valore logico 1.

Rappresentazione come operatore

Per rappresentare il nor logico non esiste un simbolo specifico.

$$x \;\; nor \;\; y$$



Operazione nor logico / 2

Rappresentazione dell'operazione $x \; nor \; y$ con la tavola della verità:

x	y	$x \; nor \; y$
0	0	1
0	1	0
1	0	0
1	1	0

Proprietà

$$x \; nor \; 0 = \overline{x}$$

$$x \; nor \; 1 = 0$$

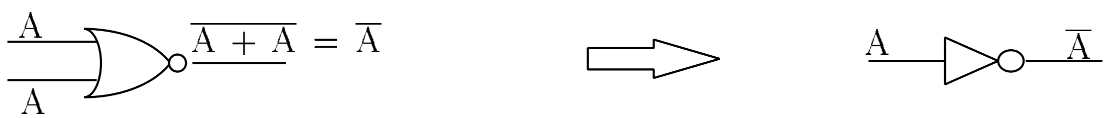
$$x \; nor \; x = \overline{x}$$

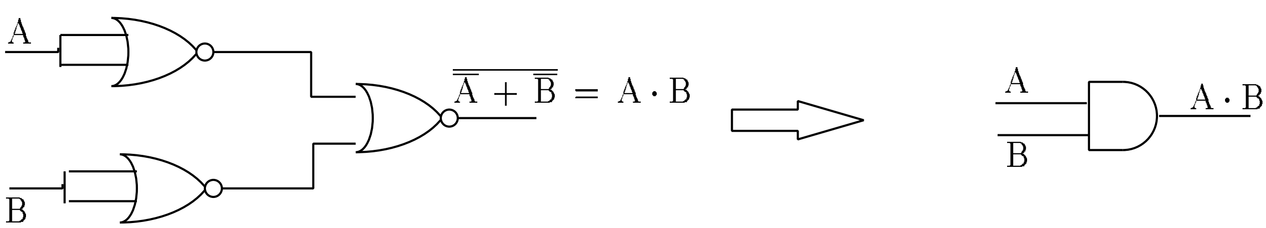
$$x \; nor \; \overline{x} = 0$$

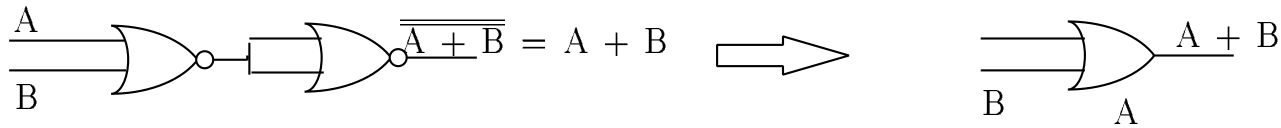
$$x \; nor \; y = \overline{x + y}$$

Porta NOR

Con il solo operatore *NOR*, si possono rappresentare gli operatori *NOT*, *AND* e *OR*.







Operazione di or esclusivo (exor) / 1

Definizione informale

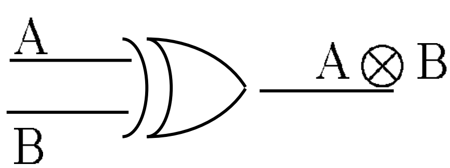
L'operazione di or esclusivo *(exor)* fra due (o più) variabili fornisce il valore logico 1 se il numero delle variabili che assumono valore logico 1 è *dispari*.

Rappresentazione come operatore

Per rappresentare l'operatore *exor* si usa comunemente la seguente notazione:

$$x \oplus y$$

$$x \ \textit{exor} \ y$$



Operazione or esclusivo (exor) / 2

Rappresentazione dell'operazione x *exor* y con la tavola della verità:

x	y	$x \oplus y$
0	0	0
0	1	1
1	0	1
1	1	0

Proprietà

$$x \oplus 0 = x$$

$$x \oplus 1 = \overline{x}$$

$$x \oplus x = 0$$

$$x \oplus \overline{x} = 1$$

$$x \oplus y = \overline{x} \oplus \overline{y}$$

Operazione or esclusivo (exor) / 3

Proprietà

$$x \oplus y = x \cdot \overline{y} + \overline{x} \cdot y$$

◇ l'operatore exor può essere visto come un comparatore di uguaglianza:

$$\text{if } x = y \text{ then } x \bigoplus y = 0$$

$$\text{else } x \bigoplus y = 1$$

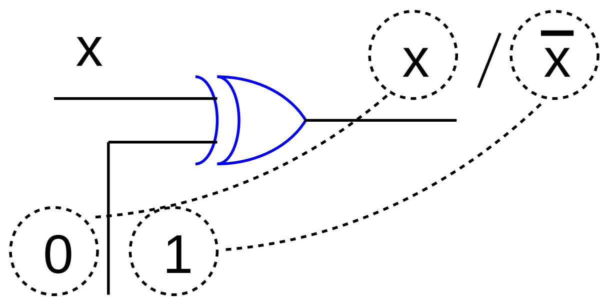
◇ l'operatore exor può essere visto come un invertitore controllato:

$$\text{if } x = 0 \text{ then } x \bigoplus y = y$$

$$\text{else } x \bigoplus y = \overline{y}$$

exor come invertitore controllato

Mettendo una variabile x come ingresso di una porta *exor*, si può ottenere in uscita il valore x stesso forzando il secondo ingresso a 0, o il valore complementato $\bar{x}$ forzando il secondo ingresso a 1.



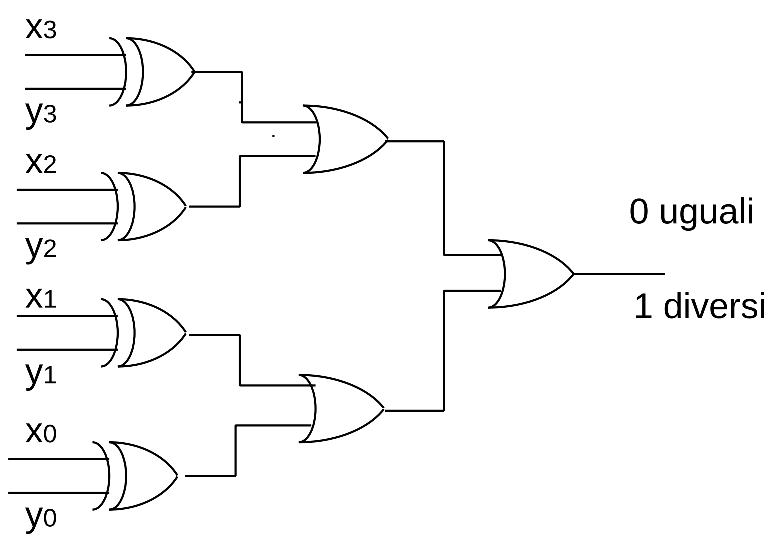
x	0	$x \oplus 0 = x$
0	0	0
1	0	1

x	1	$x \oplus 1 = \bar{x}$
0	1	1
1	1	0

exor come comparatore di uguaglianza

Si abbiano due parole di 4 bit $(x_3\, x_2\, x_1\, x_0)$ e $(y_3\, y_2\, y_1\, y_0)$.

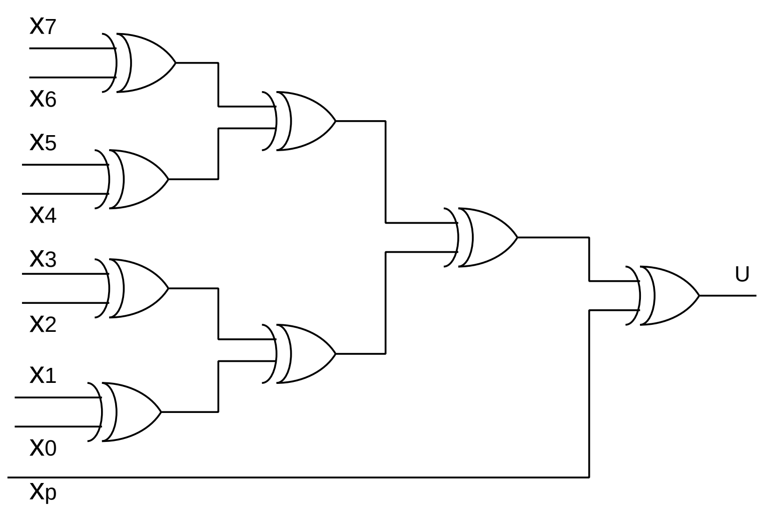
La rete logica in figura verifica se le due parole sono uguali e diverse: in particolare se l'uscita è 0 le parole sono uguali, se l'uscita è 1 le parole sono diverse.



exor come verificatore di parità

Si abbia una parola di 8 bit con bit di parità $(x_7\, x_6\, x_5\, x_4\, x_3\, x_2\, x_1\, x_0\, x_p)$.

La rete logica in figura verifica se il numero totale di 1 della parola è pari o dispari: in particolare se l'uscita è 0 il numero di 1 è pari, se l'uscita è 1 il numero di 1 è dispari.



Funzioni Logiche

Le funzioni logiche sono ottenute dalla composizione di *operazioni logiche*.

Una funzione di variabili logiche X, Y, Z, ...

$$\Phi = F(X, Y, Z, \ldots)$$

ha come dominio il prodotto cartesiano delle sue variabili, e come immagine una variabile logica.

Φ è una variabile logica $\{0, 1\}$.

Tavola della verità di funzioni logiche

La tavola della verità di una funzione logica si ottiene valutando il valore di verità della funzione in corrispondenza di tutte le possibili combinazioni delle sue variabili.

Se la funzione Φ dipende da n variabili logiche, la tavola della verità avrà 2^n righe.

X	Y	Z	$\cdots$	$F(X, Y, Z, \ldots)$
0	0	0	$\cdots$	$\{0, 1\}$
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\{0, 1\}$
1	1	1	$\cdots$	$\{0, 1\}$

Minterm di funzioni logiche

Il *minterm* di ordine i di una funzione di n variabili è una funzione prodotto delle n variabili in forma diretta o in forma negata che vale 1 in corrispondenza alla sola combinazione i delle variabili.

Nel *minterm* di ordine i compaiono in forma diretta le variabili il cui valore è 1 nella tavola della verità e compaiono in forma negata le variabili il cui valore è 0 nella tavola della verità.

x	y	*minterm*
0	0	$\overline{x}\,\overline{y}$
0	1	$\overline{x}\,y$
1	0	$x\,\overline{y}$
1	1	$x\,y$

Minterm di funzioni logiche

La seguente tavola della verità riporta i *minterm* per una funzione di 3 variabili.

x	y	z	*minterm*
0	0	0	$\overline{x}\,\overline{y}\,\overline{z}$
0	0	1	$\overline{x}\,\overline{y}\,z$
0	1	0	$\overline{x}\,y\,\overline{z}$
0	1	1	$\overline{x}\,y\,z$
1	0	0	$x\,\overline{y}\,\overline{z}$
1	0	1	$x\,\overline{y}\,z$
1	1	0	$x\,y\,\overline{z}$
1	1	1	$x\,y\,z$

Forma canonica SP

La forma canonica *Somma di Prodotti (SP)* di una funzione logica si ottiene sommando i minterm in corrispondenza dei quali la funzione vale 1.

ESEMPIO:

Data la funzione F espressa dalla seguente tavola della verità:

x	y	z	F	*minterm*
0	0	0	1	$\overline{x}\,\overline{y}\,\overline{z}$
0	0	1	0	
0	1	0	0	
0	1	1	1	$\overline{x}\,y\,z$
1	0	0	1	$x\,\overline{y}\,\overline{z}$
1	0	1	0	
1	1	0	1	$x\,y\,\overline{z}$
1	1	1	0	

$$F(x,\, y,\, z) \,=\, \overline{x}\,\overline{y}\,\overline{z} \,+\, \overline{x}\,y\,z \,+ x\,\overline{y}\,\overline{z} \,+\, x\,y\,\overline{z}$$

Notazione compatta forme canoniche SP

Per la forma SP si usa la seguente notazione compatta:

$$F = \sum_n \{ i_1, i_2, \ldots, i_k \}$$

dove n è il numero di variabili, e $i_1, i_2, \ldots, i_k$ il valore (espresso come numero decimale), dell'indice delle righe nella tavola della verità della funzione in cui la funzione stessa vale 1.

ESEMPIO - La funzione di pg 43 ha la seguente espressione compatta:

$$F = \sum_3 \{ 0, 3, 4, 6 \}$$

ESEMPIO - Sia data la seguente funzione, espressa in forma compatta:

$$F = \sum_4 \{ 1, 5, 9, 10, 11, 14, 15 \}$$

Indicando con $(a, b, c, d,)$ le 4 variabili (nell'ordine), l'espressione canonica PS della funzione è data da:

$$F = \overline{a}\overline{b}\overline{c}d + \overline{a}b\overline{c}d + a\overline{b}\overline{c}d + a\overline{b}c\overline{d} + a\overline{b}cd + abc\overline{d} + abcd$$

Insiemi completi di operatori

Poichè qualsiasi funzione logica può essere espressa in forma canonica *SP* o *PS*, l'insieme degli operatori *not*, *and* e *or* costituisce un insieme completo di operatori.

Ricorrendo ai teoremi di De Morgan si può vedere che anche *not* e *and*, oppure *not* e *or* costituiscono un insieme completo di operatori.

Ricorrendo alle proprietà degli operatori *nand* e *nor*, si può vedere che ciascuno di essi, singolarmente, costituisce un insieme completo di operatori.

Si può quindi realizzare una qualunque funzione logica utilizzando circuiti che abbiano solo porte *nand* (oppure *nor*).

Logica a due livelli

Se si trascura l'operatore *not* (cioè si suppone che ogni variabile sia disponibile in forma diretta e in forma negata), qualunque funzione logica può essere rappresentata con una rete logica a due livelli.

Infatti, qualunque funzione logica può essere espressa in forma canonica *SP* o *PS*, da cui:

Forma Canonica SP - Ad un primo livello le variabili sono poste all'ingresso di porte *AND*, realizzando il prodotto, e le uscite delle porte *AND* sono poste all'ingresso di un porta *OR* realizzando la somma.

Forma Canonica PS - Ad un primo livello le variabili sono poste all'ingresso di porte *OR*, realizzando la somma, e le uscite delle porte *OR* sono poste all'ingresso di un porta *AND* realizzando il prodotto.

Logica a due livelli - Esempio

Date tre variabili a, b e c, si definisce funzione di maggioranza $F(a, b, c)$ la funzione che vale 1 quando la maggioranza delle variabili vale 1.

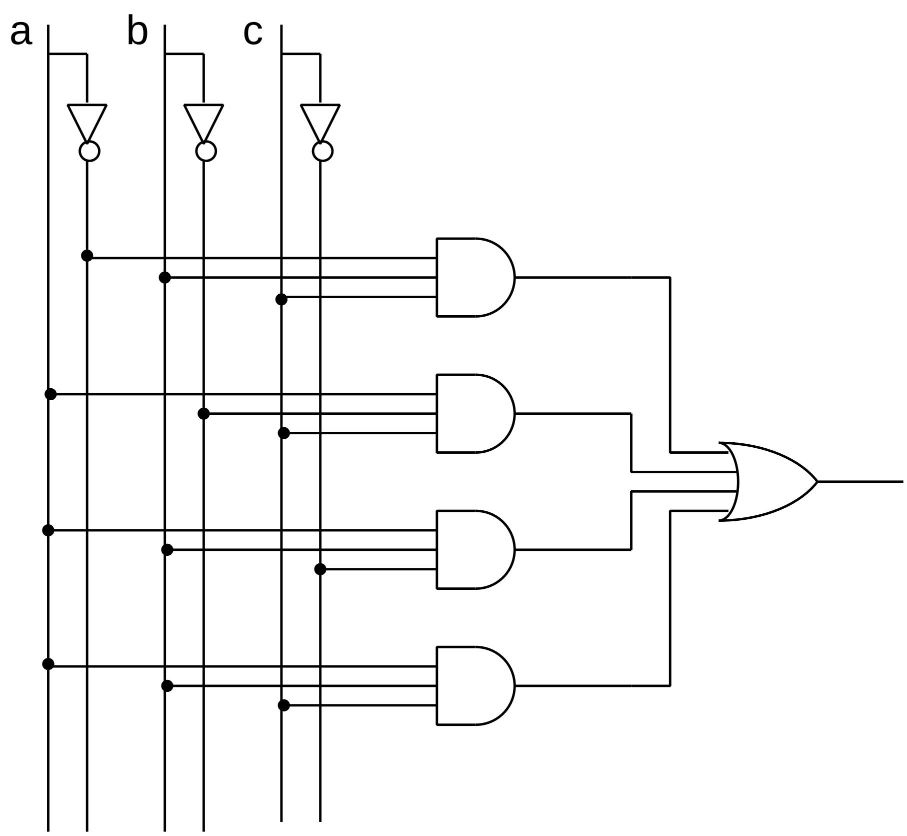
La funzione di maggioranza ha la seguente forma canonica *SP*:

a	b	c	F	*minterm*
0	0	0	0	
0	0	1	0	
0	1	0	0	
0	1	1	1	$\overline{a}\,b\,c$
1	0	0	0	
1	0	1	1	$a\,\overline{b}\,c$
1	1	0	1	$a\,b\,\overline{c}$
1	1	1	1	$a\,b\,c$

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

Logica a due livelli - Rete Logica

La rete logica corrispondente alla funzione di maggioranza in forma canonica *SP* ha la seguente struttura:



Enumerazione di funzioni logiche

Date n variabili, esistono soltanto $2^{(2^n)}$ funzioni logiche distinte. Infatti, le configurazioni delle n variabili sono 2^n, e le possibili combinazioni di 0 e 1 che si possono costruire su 2^n configurazioni sono appunto $2^{(2^n)}$.

Tutte le possibili funzioni di due variabili sono $2^{(2^2)} = 16$ e sono elencate nel seguito:

0	0	1	1	a
0	1	0	1	b
0	0	0	0	$F_0 = 0$ (costante 0)
0	0	0	1	$F_1 = a \cdot b$ (*AND*)
0	0	1	0	$F_2 = a \cdot \overline{b}$ (*a AND NOT b*)
0	0	1	1	$F_3 = a$
0	1	0	0	$F_4 = \overline{a} \cdot b$ (*NOT a AND b*)
0	1	0	1	$F_5 = b$
0	1	1	0	$F_6 = a \oplus b$ (disparità *XOR*)
0	1	1	1	$F_7 = a + b$ (*OR*)
1	0	0	0	$F_8 = \overline{a} \cdot \overline{b}$ (*NOR*)
1	0	0	1	$F_9 = \overline{a}\overline{b} + ab$ (equivalenza *NOT XOR*)
1	0	1	0	$F_{10} = \overline{b}$
1	0	1	1	$F_{11} = a + \overline{b}$ (b implica a)
1	1	0	0	$F_{12} = \overline{a}$ (*NOT a*)
1	1	0	1	$F_{13} = \overline{a} + b$ (a implica b)
1	1	1	0	$F_{14} = \overline{a} + \overline{b}$ (*NAND*)
1	1	1	1	$F_{15} = 1$ (costante 1)