ALGEBRA BOOLEANA

VARIABILI E PORTE LOGICHE

Andrea Bobbio
Anno Accademico 1998-1999

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.

Definizione di Algebra Booleana

Un insieme $B = \{a, b, c, \ldots\}$ e due operazioni binarie ad esso associate $+$ e $\cdot$, formano un'*Algebra Booleana* se e solo se sono soddisfatti i seguenti postulati:

- ◇ Le operazioni sono commutative:
  $a + b = b + a$
  $a \cdot b = b \cdot a$
- ◇ Ciascuna delle due operazioni gode della proprietà distributiva rispetto all'altra:
  $a \cdot (b + c) = a \cdot b + a \cdot c$
  $a + b \cdot c = (a + b) \cdot (a + c)$
- ◇ Esistono due elementi identità: 0 e 1 (o elementi *neutri*) rispettivamente per le operazioni $+$ e $\cdot$, tali che $\forall a \in B$:
  $0 + a = a$
  $1 \cdot a = a$
- ◇ Per ogni $a \in B$ esiste un elemento $\overline{a}$ (detto complemento di a) tale che:
  $a + \overline{a} = 1$
  $a \cdot \overline{a} = 0$

Teoremi dell'Algebra Booleana

Dall'insieme di postulati è possibile dimostrare i seguenti teoremi:

- *Idempotenza:*
  $a + a = a$
  $a \cdot a = a$
- *Elemento nullo (forcing function):*
  $a + 1 = 1$
  $a \cdot 0 = 0$
- *Involuzione:*
  $\overline{(\overline{a})} = a$
- *Assorbimento*:
  $a + a \cdot b = a$
  $a \cdot (a + b) = a$
- *Associatività:*
  $a + (b + c) = (a + b) + c = a + b + c$
  $a \cdot (b \cdot c) = (a \cdot b) \cdot c = a \cdot b \cdot c$
- *De Morgan:*
  $\overline{(a + b)} = \overline{a} \cdot \overline{b}$
  $\overline{(a \cdot b)} = \overline{a} + \overline{b}$

Algebra degli Insiemi

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

- ⇒ Dato un insieme universo U, gli insiemi B siano costituiti da tutti i possibili sottoinsiemi dell'insieme universo.
- ⇒ Il prodotto logico $\cdot$ sia interpretato come l'operazione di intersezione fra insiemi.
- ⇒ La somma logica $+$ sia interpretata come l'operazione di unione fra insiemi.
- ⇒ Il valore 1 (elemento neutro rispetto a $\cdot$) sia sostituito dall'insieme universo U (elemento neutro rispetto all'intersezione).
- ⇒ Il valore 0 (elemento neutro rispetto a $+$) sia sostituito dall'insieme vuoto Φ (elemento neutro rispetto all'unione).

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

Elementi dell'Algebra Booleana

Vengono definiti i seguenti concetti:

- ◇ *variabili booleane*
- ◇ *operatori booleani*
- ◇ *funzioni booleane*
- ◇ *porte logiche*

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:

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

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

$$x + 0 = x$$

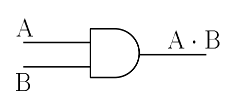
$$x + 1 = 1$$

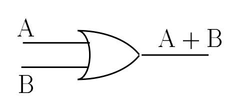
$$x + x = x$$

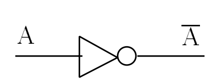
$$x + \overline{x} = 1$$

Porte Logiche

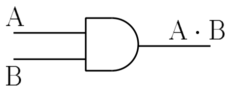
Le *porte logiche* sono dispositivi elettronici capaci di eseguire operazioni logiche su *variabili booleane*.





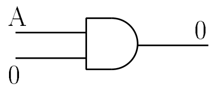


Porta AND

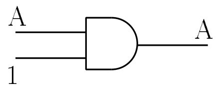


Alcune proprietà della porta *AND*.

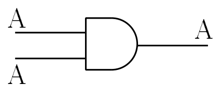
$A \cdot 0 = 0$



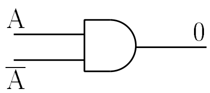
$A \cdot 1 = A$



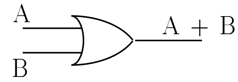
$A \cdot A = A$



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

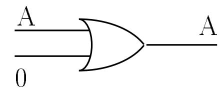


Porta OR

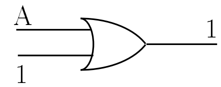


Alcune proprietà della porta *OR*.

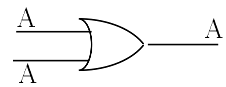
$A + 0 = A$



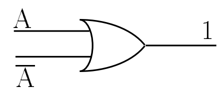
$A + 1 = 1$



$A + A = A$



$A + \overline{A} = 1$



Teoremi dell'algebra Booleana

Le proprietà degli operatori logici *NOT*, *AND* e *OR*, permettono di stabilire i seguenti teoremi.

◇ *Teorema di Idempotenza*

$x + x = x$

$x \cdot x = x$

◇ *Elemento nullo (forcing function)*

$x + 1 = 1$

$x \cdot 0 = 0$

◇ *Associatività*

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

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

Reciprocità dei Teoremi dell'algebra Booleana

Le proprietà che valgono per l'operatore + valgono anche per l'operatore · 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

◇ *Teoremi di De Morgan*

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

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

Teoremi dell'algebra Booleana

◇ *Assorbimento*

Il teorema dell'assorbimento può assumere varie forme.

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

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

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

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

◇ *Legge di Cancellazione*

Nell'algebra booleana non vale la legge di cancellazione

Dall'espressione:

$$x + y = x + z$$

non è possibile dedurre:

$$y = z$$

◇ *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

Operazione di nand logico

Definizione informale

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

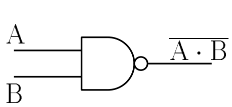
Il simbolo *nand* è una contrazione di *not and*.

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

$$x \ nand \ 0 = 1$$

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

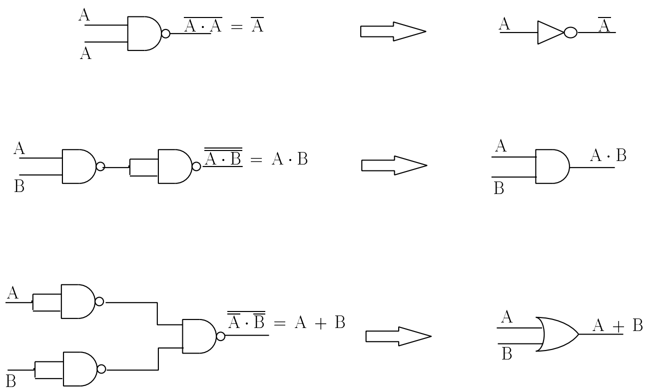
$$x \ nand \ x = \overline{x}$$

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

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

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.

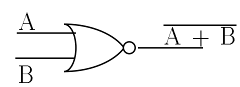
Il simbolo *nor* è una contrazione di *not or*.

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

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

$$x \ nor \ 1 = 0$$

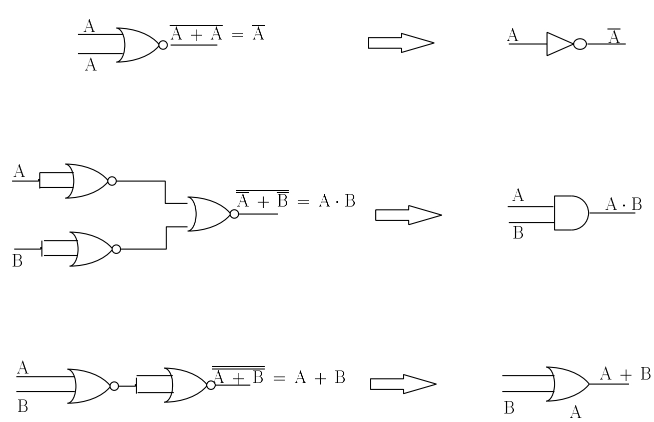
$$x \ nor \ x = \overline{x}$$

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

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

Porta NOR

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



Operazione di or esclusivo (exor) / 1

Definizione informale

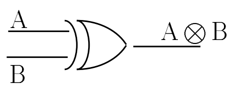
L'operazione di or esclusivo *(exor)* fra due (o più) variabili fornisce il valore logico 1 se il numero delle variabili che assumono valore logico 1 è *dispari*.

Rappresentazione come operatore

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

$$x \oplus y$$

$$x \text{ } exor \text{ } y$$



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:

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

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

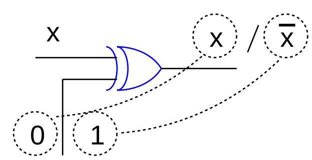
◇ l'operatore exor può essere visto come un invertitore controllato:

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

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

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.



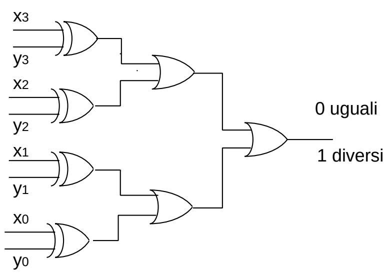
x	0	$x \oplus 0 = x$
0	0	0
1	0	1

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.