SAPIENZA
Università di Roma
Dipartimento di Informatica

Espressioni ed operatori booleani

Prof. Daniele Gorla

Espressioni Booleane

Un'*espressione booleana* è una sequenza composta da operatori booleani, parentesi, costanti e variabili booleane, induttivamente definita come segue:

Sia V un insieme numerabile di variabili; allora

- $0, 1 \in EB$;
- se $x \in V$, allora $x \in EB$;
- se $E \in EB$, allora $\overline{E}$, $(E) \in EB$;
- se $E_1, E_2 \in EB$, allora E_1+E_2 , $E_1 \cdot E_2 \in EB$.

Espressione Duale e Complementare

Espressione duale: si ottiene scambiando 0 e 1 , + e ·

Espressione complementare: come la duale, ma in più complementa le variabili (si ottiene applicando De Morgan)

Esempio: $E = (x+0) \cdot y + 1 \cdot z$

Duale:

$(x \cdot 1 + y) \cdot (0 + z)$

Complementare:

$$\overline{(x+0)\cdot y + 1\cdot z} = \overline{(x+0)\cdot y}\cdot\overline{1\cdot z} = (\overline{x+0}+\overline{y})\cdot(\overline{1}+\overline{z})$$

$$= (\overline{x}\cdot\overline{0}+\overline{y})\cdot(0+\overline{z}) = (\overline{x}\cdot 1+\overline{y})\cdot(0+\overline{z})$$

Equivalenza di Espressioni Booleane

Def.: E_1 ed E_2 sono *equivalenti* se hanno lo stesso valore a fronte dello stesso assegnamento di valori booleani alle loro variabili.

Verifica: 1. tramite dimostrazioni formali
2. tramite induzione perfetta
- Considera tutti i possibili assegnamenti alle variabili
- Calcola incrementalmente il valore dell'espressione per ogni assegnamento

Esempio: $x + xy = x + xz$

- $x + xy = x\,(1 + y) = x = x\,(1 + z) = x + xz$
-

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

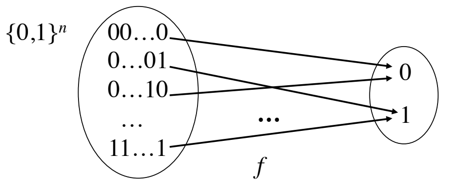
Funzioni Booleane

Quindi, una EB identifica una *funzione booleana*, cioè una legge che, in base ai valori delle variabili, restituisce in maniera univoca un valore booleano:

$$f: \{0,1\}^n \rightarrow \{0,1\}$$

numero di variabili

Graficamente:



N.B.: due EB sono equivalenti se identificano la stessa FB

Tavole di verità

Una *funzione booleana* può essere rappresentata mediante una **tavola di verità** che descrive completamente l'associazione tra gli elementi del dominio e quelli del codominio.

Date n variabili, una tavola di verità è composta da 2 parti:

- nella parte sinistra elenca ordinatamente tutte le 2^n combinazioni possibili di valori binari assegnabili alle variabili
- nella parte destra, contiene una colonna di 0 e 1 tale che il valore nella riga i sia il valore assunto dalla funzione in corrispondenza dell'i-esima n-pla di valori booleani assegnati alle variabili.

Esempio (funzione associata all'EB $x \cdot y$):

x y	f
0 0	0
0 1	0
1 0	0
1 1	1

FB costanti e unarie

$$f: \{0,1\}^n \rightarrow \{0,1\}$$

Se $n = 0$, f è una costante, che può essere o 0 o 1

Se $n = 1$, sia hanno quattro possibili funzioni booleane:

x	f_0	f_1	f_2	f_3
0	0	0	1	1
1	0	1	0	1
	costante0	identità	complemento (NOT)	costante1

FB binarie

Se $n = 2$, abbiamo 16 possibili funzioni:

x y	f_0	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}	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
	costante0	• (AND)		x		y	XOR $\oplus$	+ (OR)	NOR	XNOR	$\overline{y}$		$\overline{x}$		NAND	costante1

Funzioni a Codominio $\{0,1\}^n$

SAPIENZA Università di Roma Dipartimento di Informatica

In generale, una funzione booleana può restituire una *m*-pla di bit:

$$f : \{0,1\}^n \rightarrow \{0,1\}^m$$

ES.:

x y	f
0 0	0 0 0
0 1	1 0 0
1 0	0 1 1
1 1	1 0 0

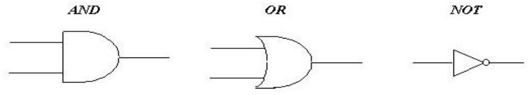
D'ora in poi, vedremo una tale funzione come una *m*-pla di funzioni a codominio {0,1}.

ES.:

x y	f_1 f_2 f_3
0 0	0 0 0
0 1	1 0 0
1 0	0 1 1
1 1	1 0 0

Porte logiche elementari

SAPIENZA Università di Roma Dipartimento di Informatica



AND

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

OR

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

NOT

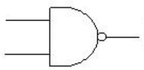
x	y
0	1
1	0

Altre Porte Logiche

SAPIENZA Università di Roma Dipartimento di Informatica

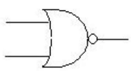
NAND

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



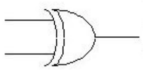
NOR

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



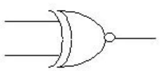
XOR

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



XNOR

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



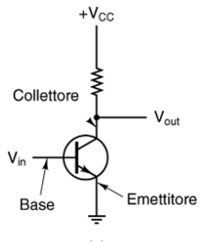
$OSS.: x \oplus y = \overline{x} \cdot y + x \cdot \overline{y}$

$OSS.: \overline{x \oplus y} = \overline{x} \cdot \overline{y} + x \cdot y$

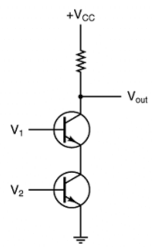
Livello fisico

SAPIENZA Università di Roma Dipartimento di Informatica

Porta NOT



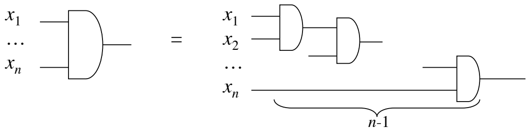
Porta NAND



Porte a più ingressi

SAPIENZA Università di Roma Dipartimento di Informatica

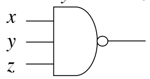
$x_1 \cdot \ldots \cdot x_n = (\ldots(x_1 \cdot x_2) \cdot \ldots \cdot x_n)$ (propr. Associativa)



Simile per la porta OR, che modella un operatore associativo (+).

Cosa succede per NAND, NOR, XOR e XNOR?

- per XOR e XNOR la situazione è simile (sono operatori associativi)
- per NAND e NOR la situazione è diversa: non essendo associativi, la scrittura x NAND y NAND z non ha senso. Pertanto, quando scriveremo



intenderemo una porta specifica a 3 ingressi, non realizzabile con due porte NAND a 2 ingressi messe in cascata.

Porte a più Ingressi (cont.)

SAPIENZA Università di Roma Dipartimento di Informatica

Lo XOR è associativo:

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

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

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

Similmente si dimostra che lo XNOR è associativo.

Invece, NOR e NAND non lo sono!

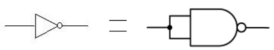
Es. (NAND): $\overline{x \cdot (\overline{y \cdot z})} = \overline{x} + y \cdot z$ $\qquad$ $\overline{(\overline{x \cdot y}) \cdot z} = x \cdot y + \overline{z}$

Queste due EB non sono equivalenti:
basta considerare l'assegnamento $x = y = 0$ e $z = 1$, che rende 1 la prima EB e 0 la seconda.

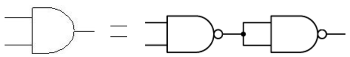
Universalità delle porte NAND

SAPIENZA Università di Roma Dipartimento di Informatica

Per idempotenza del prodotto, $x = x \cdot x \Rightarrow \overline{x} = \overline{x \cdot x} = NAND(x,x)$

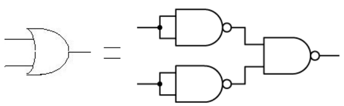


Per involuzione, $x \cdot y = \overline{\overline{x \cdot y}} = \overline{NAND(x,y)} = NAND(NAND(x,y), NAND(x,y))$



Per involuzione e De Morgan,

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



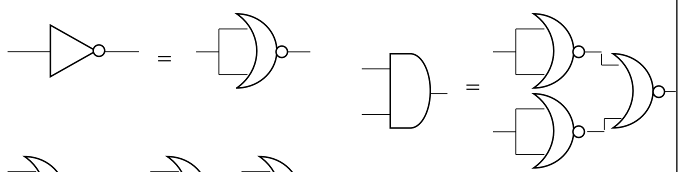
Universalità delle porte NOR

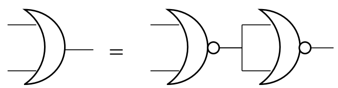
SAPIENZA Università di Roma Dipartimento di Informatica

Per dualità, si ha che

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

da cui



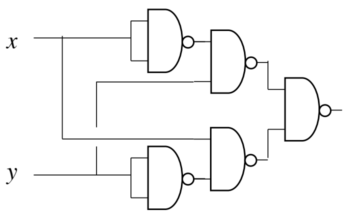


Esempio di realizzazione di un circuito con un solo tipo di porta

SAPIENZA Università di Roma – Dipartimento di Informatica

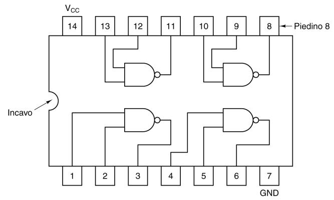
Vogliamo realizzare lo XOR usando solo porte NAND:

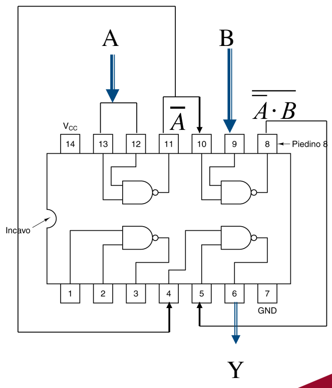
$$x \oplus y = x\cdot\overline{y} + \overline{x}\cdot y = \overline{\overline{(x\cdot\overline{y})}\cdot\overline{(\overline{x}\cdot y)}} = \overline{\overline{(x\cdot\overline{y\cdot y})}\cdot\overline{(\overline{x\cdot x}\cdot y)}}$$



Uso: circuiti integrati

SAPIENZA Università di Roma – Dipartimento di Informatica





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

$$= \overline{\overline{A}\cdot\overline{(\overline{A}\cdot B)}} = \overline{\overline{A\cdot A}\cdot\overline{(\overline{A\cdot A}\cdot B)}}$$

Dalle forme SOP a espressioni ALL-NAND

SAPIENZA Università di Roma – Dipartimento di Informatica

Data una forma SOP (somma di prodotti di variabili e variabili negate), è molto facile costruire l'espressione ALL-NAND equivalente (assumendo di poter usare porte NAND e NOR a più ingressi):

1. Applicare De Morgan alla disgiunzione (operatore più esterno)
   - Ciò trasforma l'OR in un AND negato
   - Ciò trasforma anche le congiunzioni tra le variabili in un NAND
2. Quindi, resta solo da sostituire le negazioni sulle variabili con NAND

ES. (di prima):

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

Dalle forme POS a espressioni ALL-NOR

SAPIENZA Università di Roma – Dipartimento di Informatica

Dualmente, data una forma POS (prodotto di somme di variabili e variabili negate), è molto facile costruire l'espressione ALL-NOR equivalente (assumendo di poter usare porte NAND e NOR a più ingressi):

1. Applicare De Morgan alla congiunzione (operatore più esterno)
   - Ciò trasforma l'AND in un OR negato
   - Ciò trasforma anche le disgiunzioni tra le variabili in un NOR
2. Quindi, resta solo da sostituire le negazioni sulle variabili con NOR

ES.:

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

N.B.: qui sto usando un NOR a 3 ingressi!!