Università di Napoli Federico II – Scuola Politecnica e delle Scienze di Base
Corso di Laurea in Ingegneria Informatica

Corso di Calcolatori Elettronici I

Funzioni booleane

Variabili e funzioni booleane

- Valori booleani:
  - Elementi del sostegno dell'algebra K (0 e 1)
- Variabili booleane:
  - variabili che possono assumere valori booleani
- Funzioni booleane:
  - funzioni che associano ad una n-upla di valori booleani $x_1,x_2...x_n$ un determinato valore booleano y
    - $y=f(x_1,x_2...x_n)$
- Tabelle di verità:
  - elencano i valori della funzione per tutte le possibili combinazioni degli ingressi (esempio in figura)
  - rappresentano il modo più generale per definire una funzione booleana

x_1	x_2	x_3	y
0	0	0	*1*
0	0	1	*1*
0	1	0	*1*
0	1	1	*0*
1	0	0	*0*
1	0	1	*1*
1	1	0	*0*
1	1	1	*1*

Tabelle di verità

- Funzione algebrica
  - Funzione definita in maniera tabellare per cui alla variabile dipendente sono associate tutte le possibili combinazioni delle n variabili indipendenti
- $N = k^n$
  - numero delle ripetizioni di k valori su n posti
- $M = k^N$
  - numero delle ripetizioni di k valori su N posti
- ove:
  - n=numero delle variabili indipendenti
  - k=numero dei valori dell’algebra (k=2)
  - N=numero totale di punti della funzione
  - M=numero totale delle funzioni di n variabili

Funzioni di due variabili

- Esistono 16 diverse funzioni booleane di due variabili

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

Funzioni di due variabili

f	algebrica	nome	simb	f	algebrica	nome	simb
f_0	0	contraddizione		f_8	$\overline{x}\cdot\overline{y}=\overline{x+y}$	nor	$x\downarrow y$
f_1	$x\cdot y$	and	$x\cdot y$	f_9	$\overline{x}\cdot\overline{y}+x\cdot y$	equivalenza	$x\equiv y$
f_2	$x\cdot\overline{y}$	and-not-y		f_{10}	$\overline{y}$	not-y	$\overline{y}$
f_3	x	x		f_{11}	$x+\overline{y}$	implicazione	$y\rightarrow x$
f_4	$\overline{x}\cdot y$	and-not-x		f_{12}	$\overline{x}$	not-x	$\overline{x}$
f_5	y	y		f_{13}	$\overline{x}+y$	implicazione	$x\rightarrow y$
f_6	$\overline{x}\cdot y+x\cdot\overline{y}$	or esclusivo	$x\oplus y$	f_{14}	$\overline{x}+\overline{y}=\overline{x\cdot y}$	nand	$x\uparrow y$
f_7	$x+y$	or	$x+y$	f_{15}	1	tautologia	

Insiemi funzionalmente completi

- Un insieme *F* di funzioni si dice funzionalmente completo se qualsiasi funzione dell'algebra può essere ottenuta come composizione di funzioni appartenenti ad *F*
- Si può dimostrare che qualsiasi funzione booleana può essere espressa in forma algebrica, ed in particolare come composizione delle funzioni AND, OR, e NOT.

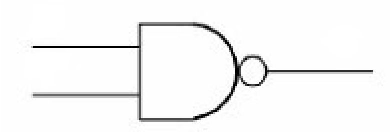
x_1	x_2	y
0	0	*0*
0	1	*1*
1	0	*1*
1	1	*0*

- Per questo, l'insieme {***AND**, **OR**, **NOT***} si dice *funzionalmente completo*
  - » esistono altri insiemi funzionalmente completi
- Si noti che grazie alle leggi di De Morgan si può costruire la AND da {OR, NOT}, oppure la OR da {AND, NOT}.
  - » quindi {**AND**, **NOT**} e {**OR**, **NOT**} sono insiemi funzionalmente completi.

Altre funzioni booleane notevoli

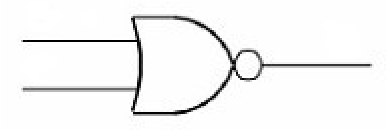
a	*b*	*y*
0	0	*1*
0	1	*1*
1	0	*1*
1	1	*0*

NAND
uguale a NOT(AND)
pari a '0' solo se entrambi gli ingressi sono '1'



a	*b*	*y*
0	0	*1*
0	1	*0*
1	0	*0*
1	1	*0*

NOR
uguale a NOT(OR)
pari a '1' solo se entrambi gli ingressi sono '0'

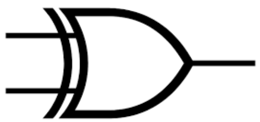


- {**NAND**} e {**NOR**} sono due insiemi funzionalmente completi
- ad esempio, con la sola NAND è possibile ottenere
  - **NOT**: si può ottenere come **NOT**(*a*) = **NOT** (a **AND** a) = *a* **NAND** *a*
  - **AND**: si può ottenere come *a* **AND** *b* = **NOT**(*a* **NAND** *b*)
  - **OR**: si può ottenere usando la seconda Legge di De Morgan:
    NOT(*a* **OR** *b*) = **NOT**(*a*) **AND NOT**(*b*) e quindi, negando a destra e a sinistra:
    a **OR** *b* = **NOT**[**NOT**(*a*) **AND NOT**(*b*)] = **NOT**(*a*) **NAND NOT**(*b*)
- Con la sola porta NAND si può quindi realizzare qualsiasi funzione booleana
- Similmente per la porta NOR

Altre funzioni booleane notevoli

a	*b*	*y*
0	0	*0*
0	1	*1*
1	0	*1*
1	1	*0*

XOR
funzione **OR-esclusivo**
o funzione **disparità**
pari a '1' quando i due ingressi sono diversi



a	*b*	*y*
0	0	*1*
0	1	*0*
1	0	*0*
1	1	*1*

EQ
funzione **eguaglianza**
pari a '1' quando i due ingressi sono uguali

- In forma algebrica, la funzione XOR si può scrivere come

  a **XOR** *b* = [*a* **AND NOT**(*b*)] **OR** [**NOT**(*a*) **AND** *b*]

- In forma algebrica, la funzione EQ può essere espressa come

  a **EQ** *b* = [*a* **AND** *b*] **OR** [**NOT**(*a*) **AND NOT**(*b*)]

Definizioni

$$y = a\bar{b} + b\bar{d}x$$

SOMMA DI PRODOTTI

b,d,¬x letterali

$$y = a\,(b+\bar{c})\,(\bar{a}+x)$$

PRODOTTO DI SOMME

termine elementare (clausola)

$$f(a,b,c) = \ldots + a\bar{b}\bar{c} + \ldots$$

fattore elementare

$$f(a,b,c) = \ldots(\ldots)\cdot(a+\bar{b}+\bar{c})\cdot(\ldots)\ldots$$

mintermine (P_i)

maxtermine (S_i)

Mintermini e Maxtermini

$$P_0 = \bar{a}\bar{b}\bar{c} \quad P_5 = a\bar{b}c$$

$$S_0 = a + b + c \quad S_5 = \bar{a} + b + \bar{c}$$

$$\overline{P_i} = S_i$$ (da de Morgan)

$$\forall i \neq j \; P_i \cdot P_j = 0, \; S_i + S_j = 1$$

$$\sum P_i = 1, \prod S_i = 0$$

Forma normale di tipo P

$$f(x_1,\ldots\ldots,x_n)=\overline{x_1}f(0,x_2,\ldots\ldots,x_n)+x_1f(1,x_2,\ldots\ldots,x_n)=$$

$$=\overline{x_1}[\overline{x_2}f(0,0,x_3,\ldots x_n)+x_2f(0,1,x_3,\ldots x_n)]+$$

$$x_1[\overline{x_2}f(1,0,x_3,\ldots x_n)+x_2f(1,1,x_3,\ldots x_n)]$$

$\ldots\ldots$

$$=f(0,0,\ldots 0)\overline{x_1}\,\overline{x_2}\cdots\overline{x_n}+\ldots+f(1,1,\ldots 1)x_1x_2\cdots x_n=\sum_{i=0}^{2^n-1}\alpha_iP_i$$

dove

$$\alpha_0=f(0,0,\ldots 0),\ \alpha_1=f(0,0,\ldots 1),\ldots\ldots\alpha_{2^n-1}=f(1,1,\ldots 1),$$

Forma normale di tipo P

- E' sempre possibile ottenere l'espressione algebrica di una funzione booleana data a partire dalla sua tabella di verità
- Un modo per farlo consiste nel vedere la funzione come somma (OR) di prodotti (AND)
  - ogni prodotto corrisponde ad un '1' nella tabella di verità e si può ottenere come AND delle variabili di ingresso, (x_1,x_2,x_3) nell'esempio, negate se nella riga corrispondente all''1' figurano come '0'

x_1	x_2	x_3	**y**
0	0	0	***1***
0	0	1	*0*
0	1	0	*0*
0	1	1	*0*
1	0	0	*0*
1	0	1	***1***
1	1	0	*0*
1	1	1	***1***

- Nell'esempio in figura la funzione può essere scritta come:

$$y = \overline{x}_1\overline{x}_2\overline{x}_3 + x_1\overline{x}_2x_3 + x_1x_2x_3$$

Forma normale di tipo P

- E' sempre possibile ottenere l'espressione algebrica di una funzione booleana data a partire dalla sua tabella di verità
- Un modo per farlo consiste nel vedere la funzione come somma (OR) di prodotti (AND)
  - ogni prodotto corrisponde ad un '1' nella tabella di verità e si può ottenere come AND delle variabili di ingresso, (x_1,x_2,x_3) nell'esempio, negate se nella riga corrispondente all''1' figurano come '0'

x_1	x_2	x_3	y
0	0	0	***1***
0	0	1	*0*
0	1	0	*0*
0	1	1	*0*
1	0	0	*0*
1	0	1	***1***
1	1	0	*0*
1	1	1	***1***

- Nell'esempio in figura la funzione può essere scritta come:

$$y = \boxed{\bar{x}_1\bar{x}_2\bar{x}_3} + x_1\bar{x}_2x_3 + x_1x_2x_3$$

Forma normale di tipo P

- E' sempre possibile ottenere l'espressione algebrica di una funzione booleana data a partire dalla sua tabella di verità
- Un modo per farlo consiste nel vedere la funzione come somma (OR) di prodotti (AND)
  - ogni prodotto corrisponde ad un '1' nella tabella di verità e si può ottenere come AND delle variabili di ingresso, (x_1,x_2,x_3) nell'esempio, negate se nella riga corrispondente all''1' figurano come '0'

x_1	x_2	x_3	y
0	0	0	***1***
0	0	1	*0*
0	1	0	*0*
0	1	1	*0*
1	0	0	*0*
1	0	1	***1***
1	1	0	*0*
1	1	1	***1***

- Nell'esempio in figura la funzione può essere scritta come:

$$y = \bar{x}_1\bar{x}_2\bar{x}_3 + \boxed{x_1\bar{x}_2x_3} + x_1x_2x_3$$

Forma normale di tipo P

- E' sempre possibile ottenere l'espressione algebrica di una funzione booleana data a partire dalla sua tabella di verità
- Un modo per farlo consiste nel vedere la funzione come somma (OR) di prodotti (AND)
  - ogni prodotto corrisponde ad un '1' nella tabella di verità e si può ottenere come AND delle variabili di ingresso, (x_1,x_2,x_3) nell'esempio, negate se nella riga corrispondente all''1' figurano come '0'

x_1	x_2	x_3	y
0	0	0	***1***
0	0	1	*0*
0	1	0	*0*
0	1	1	*0*
1	0	0	*0*
1	0	1	***1***
1	1	0	*0*
1	1	1	***1***

- Nell'esempio in figura la funzione può essere scritta come:

$$y = \overline{x}_1\overline{x}_2\overline{x}_3 + x_1\overline{x}_2x_3 + \boxed{x_1x_2x_3}$$

Forma normale di tipo P

- Viceversa, qualsiasi funzione algebrica può essere posta in forma normale P "aggiungendo" i letterali mancanti
- Basta sviluppare tutte le operazioni fino ad ottenere una somma di prodotti
- Le clausole che non siano mintermini (ovvero che non contengano tutte le variabili della funzione) possono essere moltiplicate per la somma di tutte le possibili clausole ottenibili con le variabili assenti
  - es. $f(a,b,c) = b \cdot c \cdot (a+\overline{b}+c) + c \cdot (\overline{a}+b) =$

$a \cdot b \cdot c + \cancel{b \cdot c \cdot \overline{b}} + b \cdot c \cdot c + \overline{a} \cdot c + b \cdot c =$

$a \cdot b \cdot c + b \cdot c + \overline{a} \cdot c + b \cdot c = a \cdot b \cdot c + b \cdot c + \overline{a} \cdot c =$ ← forma di tipo P

$a \cdot b \cdot c + (a+\overline{a}) \cdot b \cdot c + \overline{a} \cdot (b+\overline{b}) \cdot c = \underline{a \cdot b \cdot c} + a \cdot b \cdot c + \underline{\overline{a} \cdot b \cdot c} + \overline{a} \cdot b \cdot c + \overline{a} \cdot \overline{b} \cdot c =$

$a \cdot b \cdot c + \overline{a} \cdot b \cdot c + \overline{a} \cdot \overline{b} \cdot c$ ← forma normale di tipo P

Forma normale di tipo S

$$f(x_1 \dots x_n) = \prod_{i=0}^{2^n-1} (\alpha_i + S_i)$$

- Si può ottenere con il procedimento duale di quello usato per la forma di tipo P
- In alternativa, si può negare la forma di tipo P e poi applicare de Morgan

$$f(x_{1,} \dots x_n) = \sum_{i=0}^{2^n-1} \alpha_i P_i \;\Rightarrow\; \overline{f(x_{1,} \dots x_n)} = \sum_{i=0}^{2^n-1} \overline{\alpha_i} P_i$$

$$f(x_{1,} \dots x_n) = \overline{\sum_{i=0}^{2^n-1} \overline{\alpha_i} P_i} = \prod_{i=0}^{2^n-1} \overline{\overline{\alpha_i} P_i} = \prod_{i=0}^{2^n-1} \alpha_i + S_i$$

Forma normale di tipo S

- E’ sempre possibile ottenere l’espressione algebrica di una funzione booleana data a partire dalla sua tabella di verità
  - ogni fattore del prodotto è associato ad uno 0 presente nella colonna della tabella ed è una somma delle n variabili, ciascuna delle quali nella forma negata o non a seconda che nelle colonne corrispondenti sia presente un 1 o uno 0

x_1	x_2	x_3	y
0	0	0	*1*
0	0	1	*0*
0	1	0	*0*
0	1	1	*0*
1	0	0	*0*
1	0	1	*1*
1	1	0	*0*
1	1	1	*1*

- Nell’esempio in figura la funzione può essere scritta come:

$$y = (x_1+x_2+\overline{x}_3)\cdot(x_1+\overline{x}_2+x_3)\cdot(x_1+\overline{x}_2+\overline{x}_3)\cdot(\overline{x}_1+x_2+x_3)\cdot(\overline{x}_1+\overline{x}_2+x_3)$$

Numero caratteristico

- E’ la stringa ordinata di valori, tipica di ciascuna funzione, di lunghezza 2^n per funzioni di n variabili, coincidente con la colonna di "0" e "1" nella tabella di verità

$f = a + bc + \overline{a}b$

$\#a = 00001111$

$\#b = 00110011$

$\#c = 01010101$

$\#bc = 00010001$

$\#a + bc = 00011111$

$\#\overline{a}b = 00110000$

$\#f = \underline{00111111}$