Corso di Architettura dei Sistemi a Microprocessore

Introduzione Alle Reti Logiche

Luigi Coppolino

Contact info

Prof. Luigi Coppolino
luigi.coppolino@uniparthenope.it

Università degli Studi di Napoli "Parthenope"
Dipartimento di Ingegneria

Centro Direzionale di Napoli, Isola C4
V Piano lato SUD - Stanza n. 512

Tel: +39-081-5476702
Fax: +39-081-5476777

L'algebra di Boole - richiami

Operazioni fondamentali sui bit:

x	y	x AND y
0	0	0
0	1	0
1	0	0
1	1	1

Congiunzione
x AND y si indica anche con: $x \cdot y$

x	y	x OR y
0	0	0
0	1	1
1	0	1
1	1	1

Disgiunzione
x OR y si indica anche con $x + y$

x	NOT x
0	1
1	0

Negazione
NOT x si indica anche con $\overline{x}$

L'algebra di Boole - alcune proprietà (1)

- Proprietà commutativa:

  x AND y = y AND x   x OR y = y OR x

- Proprietà associativa:

  (x AND y) AND z = x AND (y AND z)

  (x OR y) OR z = x OR (y OR z)

  per la propr. associativa si posso definire AND e OR a più di due operandi (es. x AND y AND z)

- Proprietà di idempotenza e assorbimento:

  x AND x = x   x OR x = x

  x AND (x OR y) = x   x OR (x AND y) = x

L'algebra di Boole - alcune proprietà (2)

- Proprietà distributiva
  x AND (y OR z) = (x AND y) OR (x AND z)
  x OR (y AND z) = (x OR y) AND (x OR z)

- Proprietà di convoluzione
  NOT (NOT x) = x

- Proprietà del minimo e del massimo:
  x AND 1 = x   x AND 0 = 0
  x OR 0 = x   x OR 1 = 1

- Leggi di De Morgan:
  NOT (x AND y) = (NOT x) OR (NOT y)
  NOT (x OR y) = (NOT x) AND (NOT y)

Funzioni booleane (o logiche)

$y = f(x_1, x_2, \ldots, x_n)$ è una funzione booleana se ad ogni ennupla di valori booleani $x_1,\ldots,x_n$ associa un valore booleano y

Due esempi:

x_1	x_2	$f(x_1,x_2)$
0	0	0
0	1	1
1	0	1
1	1	0

Questa funzione è detta OR esclusivo, o XOR

x_1	x_2	$f(x_1,x_2)$
0	0	1
0	1	0
1	0	0
1	1	1

Questa funzione è detta equivalenza, o EQU

Anche AND, OR e NOT sono funzioni booleane. Esse vengono dette *funzioni fondamentali* dell'algebra

Insiemi funzionalmente completi

Si può dimostrare che qualsiasi funzione booleana può essere calcolata applicando le funzioni AND, OR, e NOT. Ad esempio:

x XOR y = (x AND NOT y) OR (y AND NOT x)

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 anche {AND, NOT} e {OR, NOT} sono insiemi funzionalmente completi.

Reti logiche

I valori booleani possono essere rappresentati da grandezze elettriche.
Ad esempio:

0 <=> tensione di 0 Volt

1 <=> tensione di +5 Volt

In tal caso le funzioni booleane possono essere realizzate mediante circuiti elettronici detti *reti logiche*.

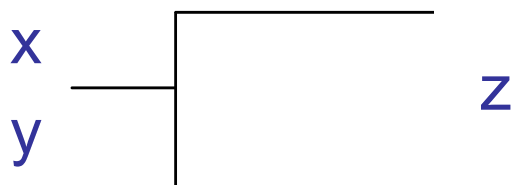
Nelle reti logiche *unilaterali*, le uscite della rete corrispondono a valori di grandezze elettriche misurate in opportuni punti del circuito; il flusso dell'elaborazione procede fisicamente in un'unica direzione, dai segnali di ingresso verso i segnali di uscita.

Nelle reti logiche *bilaterali*, invece, l'uscita della rete è determinata dalla presenza o dall'assenza di "contatto" tra due punti della rete.

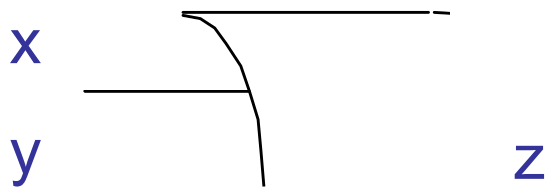
Porte logiche (gates)

Circuiti logici elementari che realizzano le operazioni fondamentali. Le reti logiche si costruiscono connettendo più porte logiche.

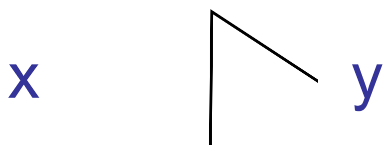
Simboli delle principali porte logiche:



z = x AND y



z = x OR y



y = NOT x

(se connesso a un’altra porta, il NOT si indica talora con un semplice pallino)

Operatori logici generalizzati (1)

Dato un vettore di variabili booleane $X = (x_1, x_2, \ldots, x_n)$, e una variabile booleana α, indicheremo con la notazione:

$Y = \alpha\ \mathit{OP}\ X$ (dove *OP* è un operatore booleano)

l'operazione che produce il vettore booleano Y così definito:

$Y = (y_1, y_2, \ldots, y_n)$ con

$y_1 = \alpha\ \mathit{OP}\ x_1$

. . . .

$y_n = \alpha\ \mathit{OP}\ x_n$

Un vettore di bit viene indicato anche come **parola** (**word**)

Esempio:

α AND X ha come risultato il vettore formato da:

$(\alpha \text{ AND } x_1, \alpha \text{ AND } x_2, \ldots, \alpha \text{ AND } x_n)$

Operatori logici generalizzati (2)

Dati due vettori di variabili booleane $X = (x_1, x_2, \ldots, x_n)$ e $Y= (y_1, y_2, \ldots, y_n)$ indicheremo con la notazione:

$Z = X$ *OP* Y (dove *OP* è un operatore booleano)

l'operazione che produce il vettore booleano Z così definito:

$Z = (z_1, z_2, \ldots, z_n)$ con

$z_1 = x_1$ *OP* y_1

. . . .

$z_n = x_n$ *OP* y_n

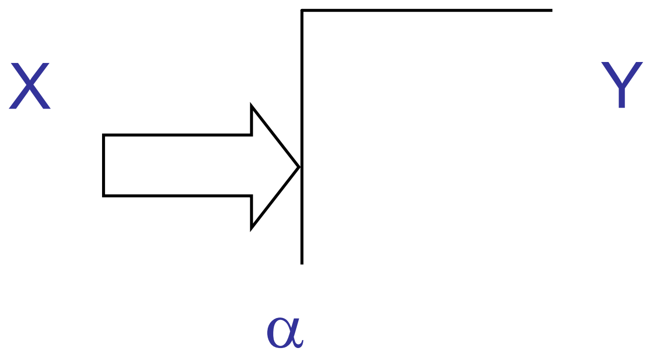
Esempio:

X OR Y ha come risultato il vettore formato da:

$(x_1 \text{ OR } y_1, x_2 \text{ OR } y_2, \ldots, x_n \text{ OR } y_n)$

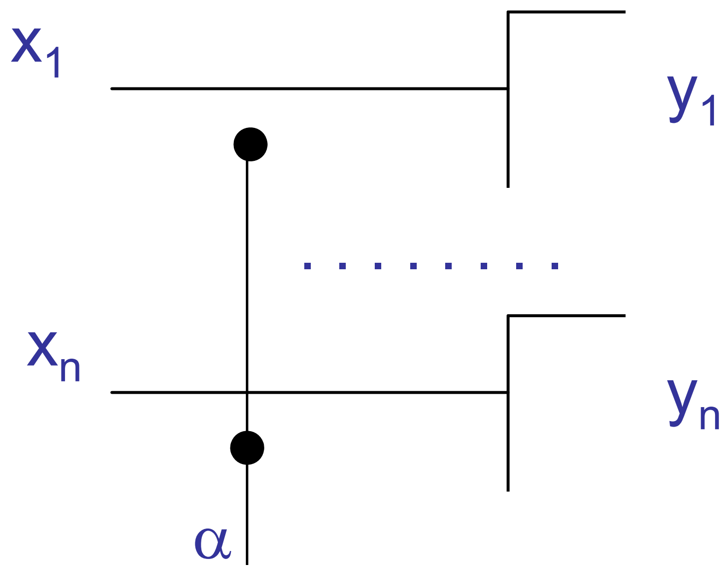
Porte logiche generalizzate (1)

Rappresentazione simbolica:



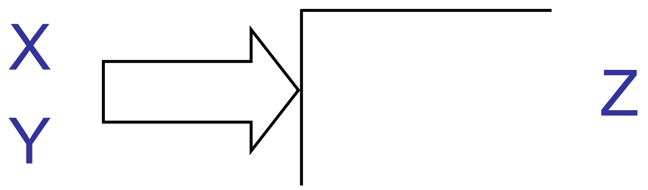
$Y = \alpha$ AND X

Circuito equivalente:



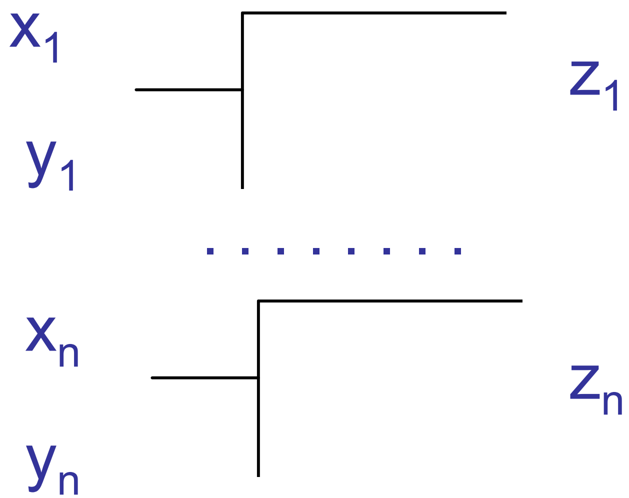
Porte logiche generalizzate (2)

Rappresentazione simbolica:



Z = X AND Y

Circuito equivalente:



Tassonomia dei circuiti digitali

- I circuiti digitali possono essere classificati in due categorie
- Circuiti combinatori
  - Il valore delle uscite ad un determinato istante dipende unicamente dal valore degli ingressi in quello stesso istante.
- Circuiti sequenziali
  - Il valore delle uscite in un determinato istante dipende sia dal valore degli ingressi in quell'istante sia dal valore degli ingressi in istanti precedenti
  - Per definire il comportamento di un circuito sequenziale è necessario tenere conto della storia passata degli ingressi del circuito
- La definizione di circuito sequenziale implica due concetti:
  - Il concetto di tempo
  - Il concetto di stato

Analisi e Sintesi di un sistema

Per *analisi* di un sistema si intende l'individuazione delle relazioni di causa/effetto tra i segnali di ingresso e uscita, attraverso l'esame di una rappresentazione schematica dei suoi componenti elementari e dei collegamenti che li interconnettono, ovvero:

- *data la rappresentazione schematica del sistema, individuarne il comportamento*.

Per *sintesi* di un sistema si intende l'individuazione dei componenti e delle interconnessioni necessarie per realizzarlo seguendo la preassegnata specifica funzionale:

- *data la specifica funzionale individuarne la struttura.*

Analisi e sintesi

Analisi

Data la descrizione della STRUTTURA (come è fatta*)*

Determinarne il COMPORTAMENTO *(cosa fa)*

Sintesi

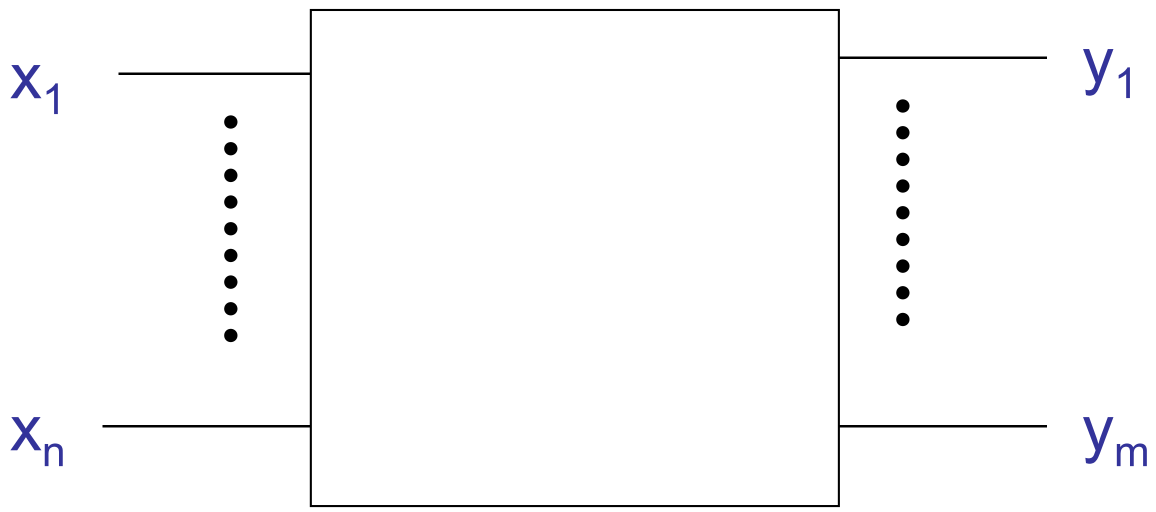
Data la descrizione del COMPORTAMENTO *(cosa deve fare)*

Determinarne la STRUTTURA (come è fatta)

Macchine combinatorie (1)

Reti logiche con *n* ingressi $x_1, x_2, \ldots, x_n$ e *m* uscite $y_1, y_2, \ldots, y_m$ che realizzano la corrispondenza:

$$y_1 = f_1(x_1, x_2, \ldots, x_n)$$
$$\cdots\cdots$$
$$y_m = f_m(x_1, x_2, \ldots, x_n)$$



Macchine combinatorie (2)

In una macchina combinatoria i valori delle uscite dipendono esclusivamente dai valori degli ingressi

In una macchina combinatoria ideale tale dipendenza è istantanea

In una macchina reale c'è sempre un ritardo tra l'istante in cui c'è una variazione in uno degli ingressi e l'istante in cui l'effetto di questa variazione si manifesta sulle uscite

Sintesi di funzioni logiche

- Possiamo sintetizzare una funzione logica a partire dalla sua tabella di verità nella forma *somma di prodotti* o *prodotti di somme.*

x_1	x_2	x_3	f_1	f_2
0	0	0	1	1
0	0	1	1	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	0	1
1	1	0	0	0
1	1	1	1	0

- Somma di prodotti:
  - Individuare le righe della tabella di verità a valore 1
  - Per ognuna di esse, rappresentare un termine come il prodotto delle variabili in input, prese come x_i se $x_i = 1$, altrimenti (NOT x_i) se $x_i = 0$.
  - Sommare tutti i termini così ottenuti