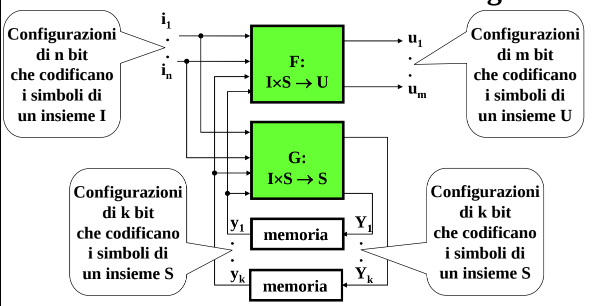
Capitolo 4

Reti logiche

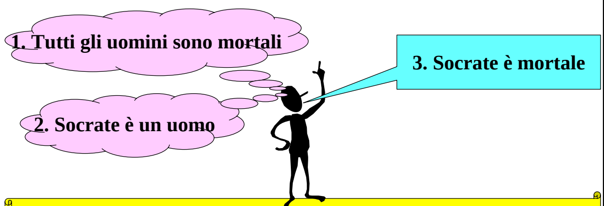
4.1 Funzioni, espressioni e schemi logici

Il modello strutturale delle reti logiche



Rete logica combinatoria	*nessuna retroazione*
Rete logica sequenziale asincrona	*retroazioni dirette*
Rete logica sequenziale sincrona	*retroazioni con flip-flop*

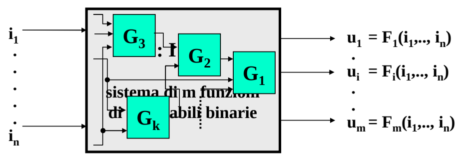
Logica e Reti logiche



Rete logica -Modello matematico che assume come *primitive* alcune semplici modalità di elaborazione di segnali binari e deduce da queste in modo rigoroso
- quale struttura soddisfa un dato comportamento,
- quale comportamento ha una data struttura.

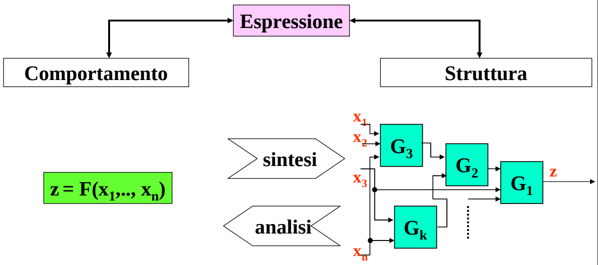
Reti combinatorie

Rete combinatoria: comportamento e struttura



Rete logica combinatoria - I valori dei segnali d'uscita dipendono solo dai valori contemporanei dei segnali d'ingresso.

comportamento-espressione-struttura

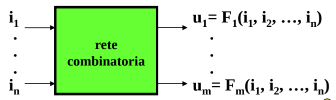


Descrizione matematica delle reti combinatorie

- **Segnali** — **Variabili binarie**
- **Blocchi** — **Funzioni booleane**
- **Gate** — **Operazioni logiche**
- **Schemi** — **Espressioni logiche**

Funzioni booleane

Funzioni di variabili binarie



Funzione completa di n variabili binarie $z = F(x_1, x_2, \ldots, x_n)$
Insieme di 2^n coppie ordinate $\{x, z \mid x \in B^n, z \in B\}$ formate da una configurazione di valori delle variabili indipendenti x_i e dal corrispondente valore della variabile dipendente z.

Il numero di distinte funzioni di n variabili binarie è finito.

$$\Phi(n) = 2^{2^n}$$

4 funzioni di 1 variabile,
16 funzioni di 2 variabili,
256 funzioni di 3 variabili,
65.536 funzioni di 4 variabili, ecc.

Funzione incompleta o non completamente specificata
Il dominio è un sottoinsieme di B^n

Esempio: BCD →7 segmenti

Tabelle della verità

Tabella della verità - Descrizione tabellare di una funzione di variabili binarie.

n+1 colonne

2^n righe

x_1 x_2 … x_n	$F(x_1, x_2, \ldots, x_n)$
0 0 0 ……..0	0 *oppure* 1 *oppure* -
1 0 0 ……..0	0 *oppure* 1 *oppure* -
0 1 0 ……..0	0 *oppure* 1 *oppure* -
1 1 0 ……..0	0 *oppure* 1 *oppure* -
0 0 1 ……..0	0 *oppure* 1 *oppure* -
⋮	⋮
0 1 1 ……..1	0 *oppure* 1 *oppure* -
1 1 1 ……..1	0 *oppure* 1 *oppure* -

Funzioni incomplete

Il Calcolo delle proposizioni

- **Proposizioni**: significato “**vero**”/ “**falso**”
- **Connettivi**: “**e**”/“**o**”/“**non**”

P, Q : proposizioni

Assunzioni:
“non P” è “vero” se e solo se P è “falso”
“P e Q” è “vero” se e solo se P è “vero” e Q è “vero”
“P o Q” è “vero” se e solo o P è “vero”, o Q è “vero”, o lo sono entrambe

Funzioni di una variabile

x	f_0	f_3	f_1	f_2
0	0	1	0	1
1	0	1	1	0

4 funzioni di una variabile

f_0 falso
f_3 vero
f_1 x
f_2 non x

Funzioni di due variabili

x_0	x_1	f_0	f_{15}	f_3	f_5	f_{12}	f_{10}	f_1	f_{14}	f_7	f_8	f_9	f_6	f_{13}	f_2	f_{11}	f_4
0	0	0	1	0	0	1	1	0	1	0	1	1	0	1	0	1	0
0	1	0	1	0	1	1	0	0	1	1	0	0	1	1	0	0	1
1	0	0	1	1	0	0	1	0	1	1	0	0	1	0	1	1	0
1	1	0	1	1	1	0	0	1	0	1	0	1	0	1	0	1	0

16 funzioni di due variabili

f_0 falso
f_{15} vero

f_3 x_0
f_5 non x_0
f_{12} x_1
f_{10} non x_1

f_1 x_0 e x_1
f_{14} non (x_0 e x_1)
f_7 x_0 o x_1
f_8 non (x_0 o x_1)
f_9 (x_0 e x_1) o (non x_0 e non x_1)
f_6 (non x_0 e x_1) o (x_0 e non x_1)

f_{13} non x_0 o x_1
x_0 *implica* x_1
***se* x_0 *allora* x_1**

f_2 ...
f_{11} ...
f_4 ...

Porte logiche

Strutture e comportamenti elementari (3)

realizza f_1: x_0 e x_1

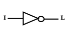
funzione and

Strutture e comportamenti elementari (4)

realizza f_7: x_0 o x_1

funzione or

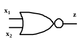
Il "not" elettronico



realizza f_5: non x_0

funzione not

Il gate "nor"

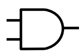


realizza f_8: non (x_0 o x_1)

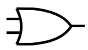
funzione nor

Dualità tra "and" e "or"(1)

Logica positiva



I1	I2	AB
0	0	0
1	0	0
0	1	0
1	1	1

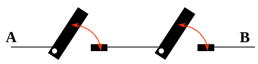


I1	I2	AB
0	0	0
1	0	1
0	1	1
1	1	1

Due differenti astrazioni!

{aperto = 0, chiuso = 1} {aperto = 1, chiuso = 0}

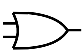
Contatti in serie



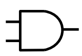
I1	I2	AB
aperto	aperto	aperto
aperto	chiuso	aperto
chiuso	aperto	aperto
chiuso	chiuso	chiuso

Dualità tra "and" e "or"(2)

Logica positiva



I1	I2	AB
0	0	0
1	0	1
0	1	1
1	1	1

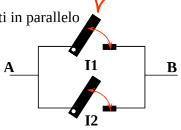


I1	I2	AB
0	0	0
1	0	0
0	1	0
1	1	1

Due differenti astrazioni!

{aperto = 0, chiuso = 1} {aperto = 1, chiuso = 0}

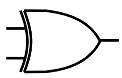
Contatti in parallelo



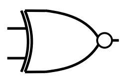
I1	I2	AB
aperto	aperto	aperto
aperto	chiuso	chiuso
chiuso	aperto	chiuso
chiuso	chiuso	chiuso

Dualità tra "ex-or" e "ex-nor"(3)

Logica positiva

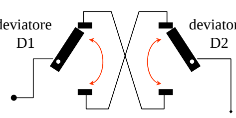


I1	I2	AB
0	0	0
1	0	1
0	1	1
1	1	0



I1	I2	AB
0	0	1
1	0	0
0	1	0
1	1	1

{alto = 0, basso = 1} {aperto = 0, chiuso = 1}
{alto = 1, basso = 0} {aperto = 1, chiuso = 0}



D1	D2	AB
alto	alto	aperto
basso	alto	chiuso
alto	basso	chiuso
basso	basso	aperto

Operazioni logiche

Funzioni e operazioni

Un'operazione è detta **logica** se è la descrizione matematica di una funzione booleana di una o di due variabili.

NOTAZIONI

f(x) = *(x)
f(x) = (x)*

f(x,y) = *(x,y)
f(x,y) = x * y

SIMBOLI

= **"f è descritta da .."**

***** **operatore**

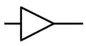
Identità : z = x

Regole:
0 = 0
1 = 1

Funzione:

x	z
0	0
1	1

Realizzazione:



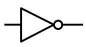
Complementazione : x' , $\overline{x}$, $\neg$x

Regole:
0' = 1
1' = 0

Funzione:

x	z
0	1
1	0

Realizzazione:



= **:** il complemento di 0 **vale** 1

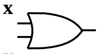
Somma logica: x + y , x ∨ y

Regole:
0 + 0 = 0
0 + 1 = 1
1 + 0 = 1
1 + 1 = 1

Funzione:

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

Realizzazione:



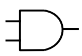
Prodotto logico: x . y , xy , x ∧ y

Regole:
0 . 0 = 0
0 . 1 = 0
1 . 0 = 0
1 . 1 = 1

Funzione:

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

Realizzazione:



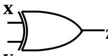
Somma modulo due: x ⊕ y

Regole:
0 ⊕ 0 = 0
0 ⊕ 1 = 1
1 ⊕ 0 = 1
1 ⊕ 1 = 0

Funzione:

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

Realizzazione:



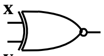
Equivalenza: x ≡ y

Regole:
0 ≡ 0 = 1
0 ≡ 1 = 0
1 ≡ 0 = 0
1 ≡ 1 = 1

Funzione:

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

Realizzazione:



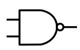
Nand (operazione di Shaffer): z = x ↑ y

Regole:
0 ↑ 0 = 1
0 ↑ 1 = 1
1 ↑ 0 = 1
1 ↑ 1 = 0

Funzione:

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

Realizzazione:



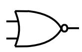
Nor (operazione di Pierce): z = x ↓ y

Regole:
0 ↓ 0 = 1
0 ↓ 1 = 0
1 ↓ 0 = 0
1 ↓ 1 = 0

Funzione:

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

Realizzazione:



Operazioni e Espressioni

$f_1(x) = x$ | $f_7(x,y) = x + y$ | $f_1(x,y) = x . y$ | $f_6(x,y) = x \oplus y$

$f_2(x) = x'$ | $f_8(x,y) = x \downarrow y$ | $f_{14}(x,y) = x \uparrow y$ | $f_9(x,y) = x \equiv y$

Espressione logica - Stringa formata da costanti, bit, operatori logici e parentesi, in accordo con le seguenti regole:
- **le costanti 0 e 1 sono espressioni**
- **le variabili binarie sono espressioni**
- **se x è un'espressione, allora anche (x)' è un'espressione**
- **se x e y sono espressioni, allora lo sono anche** **(x+y), (x.y), (x⊕y), (x≡y), (x↓y), (x↑y)**

Esempi: $(x \oplus y) \oplus (z \oplus w)$ $\quad (x \downarrow y) \downarrow 0$ $\quad a + (b.c)$

Valutazione di una espressione

Valutazione di una espressione di n variabili per una n-pla di valori
1. **Si sostituisce ad ogni variabile il valore che le compete.**
2. **Partendo dalle parentesi più interne si sostituisce ogni operazione con il suo risultato fino ad ottenere o la costante 0 o la costante 1.**

Esempio: E(a,b,c) = a+(b.c) per a=0, b=1, c=0
= 0+(1.0)
= 0+0
= 0

N° di valutazioni - Una espressione di n variabili può essere valutata in 2^n modi diversi.

Espressioni e Funzioni

Le 2^n valutazioni di una espressione $E(x_1, x_2, \dots, x_n)$ creano 2^n coppie x, z $\{x, z \mid x \in B^n, z \in B\}$

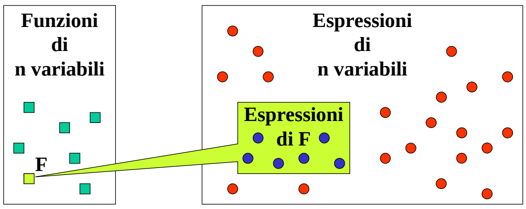
Esempio: E(a,b,c) = a+(b.c)

	a b c	E
E(0,0,0) = 0+(0.0) = 0	0 0 0	0
E(0,0,1) = 0+(0.1) = 0	0 0 1	0
E(0,1,0) = 0+(1.0) = 0	0 1 0	0
E(0,1,1) = 0+(1.1) = 1	0 1 1	1
E(1,0,0) = 1+(0.0) = 1	1 0 0	1
E(1,0,1) = 1+(0.1) = 1	1 0 1	1
E(1,1,0) = 1+(1.0) = 1	1 1 0	1
E(1,1,1) = 1+(1.1) = 1	1 1 1	1

T1) Ogni espressione descrive una e una sola funzione completa.

Equivalenza tra espressioni

Espressioni equivalenti - Due espressioni E_1, E_2 sono equivalenti, e si scrive $E_1 = E_2$, se e solo se descrivono la stessa funzione.



Metodi per dimostrare l'equivalenza: induzione perfetta, manipolazione algebrica

Proprietà

T2) proprietà commutativa (+, ., ↓, ↑, ⊕, ≡)

$a * b = b * a$

T3) proprietà associativa (+, ., ⊕)

$(a * b) * c = a * (b * c)$ $= a * b * c$

T4) complementi:

$(x + y)' = x \downarrow y$

$(x . y)' = x \uparrow y$

$(x \equiv y)' = x \oplus y$

N.B. il pallino!

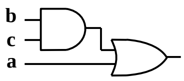
Espressioni e Schemi logici

T5) Ogni espressione descrive una struttura formata da gate connessi in serie e/o in parallelo.

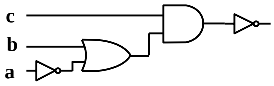
Per individuare lo schema descritto da una espressione:
1 - si parte dalle parentesi più interne e si traccia il simbolo del gate corrispondente all'operazione, collegandone gli ingressi ai segnali esterni;
2 - si procede in modo analogo con le altre coppie di parentesi, considerando via via come ingressi dei nuovi gate anche le uscite di quelli già tracciati.

Esempi

a+(b.c)



(((a)' + b) . c)'



N.B. - Lo schema logico di una espressione non può avere segnali in <u>retroazione</u> (l'uscita di ogni gate dipende da segnali d'ingresso e/o da uscite di gate disposti "a monte").

Schemi logici e Espressioni