

## 4.2 Algebra di commutazione

### Algebra di commutazione

1) **Costanti:** 0, 1

2) **Operazioni:**  
 somma logica (+)    prodotto logico (.)    complementazione (')

3) **Postulati:**

$0 + 0 = 0$	$0 \cdot 0 = 0$	$0' = 1$
$1 + 0 = 1$	$1 \cdot 0 = 0$	$1' = 0$
$0 + 1 = 1$	$0 \cdot 1 = 0$	
$1 + 1 = 1$	$1 \cdot 1 = 1$	

4) **Variabili:** simboli sostituibili o con 0 o con 1

### Espressioni

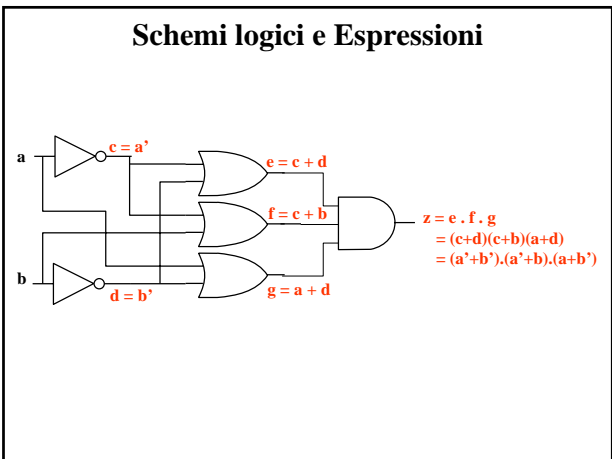
5) **Espressione** - Stringa finita di costanti, variabili, operatori e parentesi, formata in accordo con le seguenti regole:

- 1) 0 e 1 sono espressioni
- 2) una variabile è una espressione
- 3) se A è un'espressione, lo è anche (A')
- 4) se A, B sono espressioni, lo sono anche (A+B), (A.B)

*Esempi:*

$a' \cdot b$	$a + (b \cdot c)$	$(a + b)'$	$a + bc$	$a'b + 0 + ab'$
--------------	-------------------	------------	----------	-----------------

N.B - L'operazione di prodotto è prioritaria rispetto alla somma e non è obbligatorio racchiuderla tra parentesi. La notazione AB indica A.B





### Equivalenze notevoli

Proprietà della somma e del prodotto logico:

E1) <i>commutativa</i>	$x + y$	=	$y + x$
	$x \cdot y$	=	$y \cdot x$
E2) <i>associativa</i>	$(x + y) + z$	=	$x + y + z$
	$(x \cdot y) \cdot z$	=	$x \cdot y \cdot z$
E3) <i>distributiva</i>	$(x \cdot y) + (x \cdot z)$	=	$x \cdot (y + z)$
	$(x + y) \cdot (x + z)$	=	$x + (y \cdot z)$
E4) <i>idempotenza</i>	$x + x$	=	$x$
	$x \cdot x$	=	$x$
E5) <i>identità</i>	$x + 0$	=	$x$
	$x \cdot 1$	=	$x$
E6) <i>limite</i>	$x + 1$	=	$1$
	$x \cdot 0$	=	$0$

### Equivalenze notevoli

Proprietà della complementazione:

E7) <i>involuzione</i>	$(x')' = x$		
E8) <i>limitazione</i>	$x + x' = 1$		
	$x \cdot x' = 0$		
E9) <i>combinazione</i>	$xy + xy' = x$		
	$(x+y) \cdot (x+y)' = x$		
E10) <i>I<sup>a</sup> legge di De Morgan</i>	$(x + y)' = x' \cdot y'$		
<i>II<sup>a</sup> legge di De Morgan</i>	$(x \cdot y)' = x' + y'$		
E11) <i>consenso</i>	$xy + x'z + yz = xy + x'z$		
	$(x+y) \cdot (x'+z) \cdot (y+z) = (x+y) \cdot (x'+z)$		

### Espressioni di funzioni incomplete

ENCODER a 3 ingressi

$x_2$	$x_1$	$x_0$	$z_1$	$z_0$
0	0	0	0	0
1	0	0	1	1
0	1	0	1	0
0	0	1	0	1

$x_2$	$x_1$	$x_0$	$z_1$	$z_0$
0	0	0	0	0
1	0	0	1	1
0	1	0	1	0
0	0	1	0	1
1	1	0	-	-
1	0	1	-	-
0	1	1	-	-
1	1	1	-	-

N.B. le altre configurazioni sono per ipotesi impossibili

## Espressioni di funzioni incomplete

**Espressioni equivalenti di funzioni incomplete** - Espressioni che forniscono eguale valutazione limitatamente al dominio di una funzione incompleta sono dette **equivalenti**.

Espressioni per l'ENCODER:

$$z_1 = x_2 x_1 x_0' + x_2' x_1 x_0'$$

$$z_0 = x_2 x_1' x_0' + x_2' x_1' x_0$$

$$u_1 = x_2 + x_1$$

$$u_0 = x_2 + x_0$$

$x_2$	$x_1$	$x_0$	$z_1$	$u_1$	$z_0$	$u_0$
0	0	0	0	0	0	0
1	0	0	1	1	1	1
0	1	0	1	1	0	0
0	0	1	0	0	1	1
1	1	0	0	1	0	1
1	0	1	0	1	0	1
0	1	1	0	1	0	1
1	1	1	0	1	0	1



## Espressioni canoniche

### T6) Espressione canonica SP (Somma di Prodotti)

**I<sup>a</sup> forma canonica** - Ogni funzione di  $n$  variabili è descritta da una **somma di tanti prodotti** logicamente veri come le configurazioni per cui vale 1. In ciascuna **somma**, **maxtermine**, appare ogni variabile, in forma vera se la configurazione corrispondente vale 1, in forma complementata se vale 0.

### T7) Espressione canonica PS (Prodotto di Somme)

**II<sup>a</sup> forma canonica** - Ogni funzione di  $n$  variabili è descritta da un **prodotto di tante somme** logicamente vere come le configurazioni per cui vale 0. In ciascuna **somma**, **maxtermine**, appare ogni variabile, in forma vera se nella configurazione corrispondente vale 0, in forma complementata se vale 1.

Completezza dell'algebra di commutazione

## Espressioni canoniche della funzione "a implica b"

a	b	a→b
0	0	1
0	1	1
1	0	0
1	1	1

**II<sup>a</sup> forma canonica:**  
 $F(a,b) = a' + b$

**I<sup>a</sup> forma canonica:**  
 $F(a,b) = a' \cdot b' + a' \cdot b + a \cdot b$

Verifica della equivalenza per manipolazione algebrica:

$$F(a,b) = a' \cdot b' + a' \cdot b + a \cdot b$$

$$= a' \cdot (b' + b) + a \cdot b$$

$$= a' \cdot 1 + a \cdot b$$

$$= a' + a \cdot b$$

$$= a' + a \cdot b + a' \cdot b$$

$$= a' + b$$

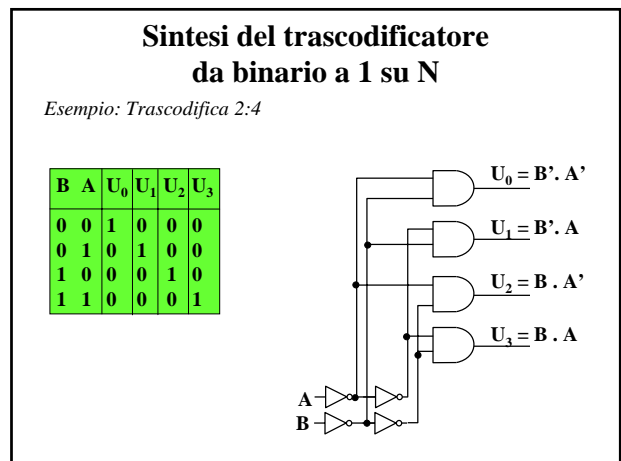
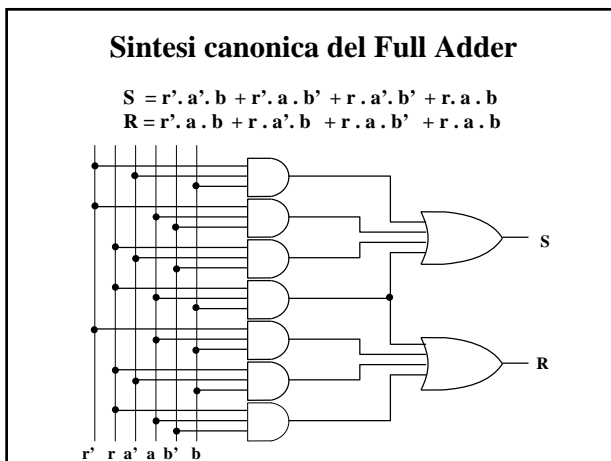
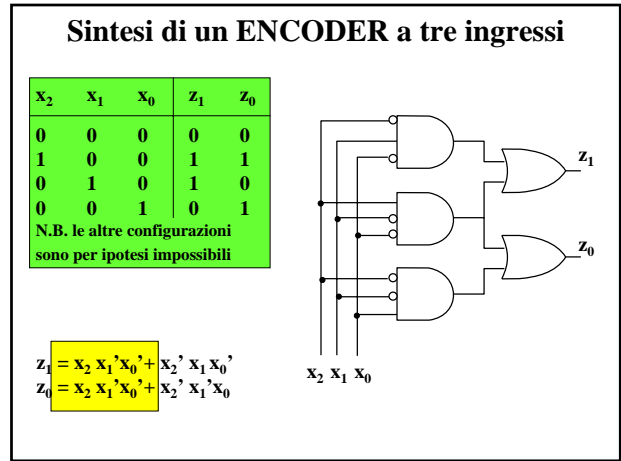
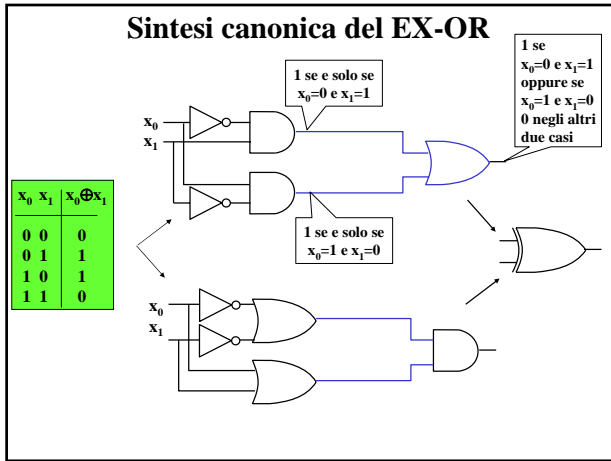
E3

E8

E5

una parte è inclusa nel tutto

E3, E8, E5



### Manipolazione algebrica

A	I <sub>0</sub>	I <sub>1</sub>	U
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

1ª forma canonica:

$$U = A' \cdot I_0 \cdot I_1' + A' \cdot I_0 \cdot I_1 + A \cdot I_0' \cdot I_1 + A \cdot I_0 \cdot I_1$$

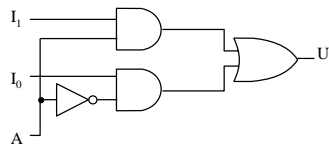
4 AND a 3 ingressi e 1 OR a 4 ingressi

forme equivalenti ottenute per "manipolazione":

$$= A' \cdot I_0 \cdot (I_1' + I_1) + A \cdot (I_0' + I_0) \cdot I_1$$

$$= A' \cdot I_0 \cdot 1 + A \cdot 1 \cdot I_1$$

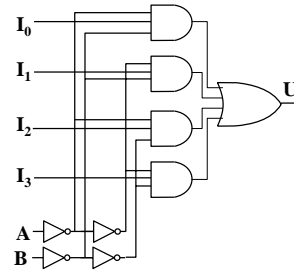
$$= A' \cdot I_0 + A \cdot I_1$$



### Il Selettore a quattro vie

MUX a 4 vie (espressione SP)

$$U = A_1' \cdot A_0' \cdot I_0 + A_1' \cdot A_0 \cdot I_1 + A_1 \cdot A_0' \cdot I_2 + A_1 \cdot A_0 \cdot I_3$$



Notazioni simboliche

### Notazioni simboliche per le espressioni canoniche

i	r	a	b	R	S
0	0	0	0	0	0
1	0	0	1	0	1
2	0	1	0	0	1
3	0	1	1	1	0
4	1	0	0	0	1
5	1	0	1	1	0
6	1	1	0	1	0
7	1	1	1	1	1

$$S(r,a,b) = \Sigma_3 m(1,2,4,7)$$

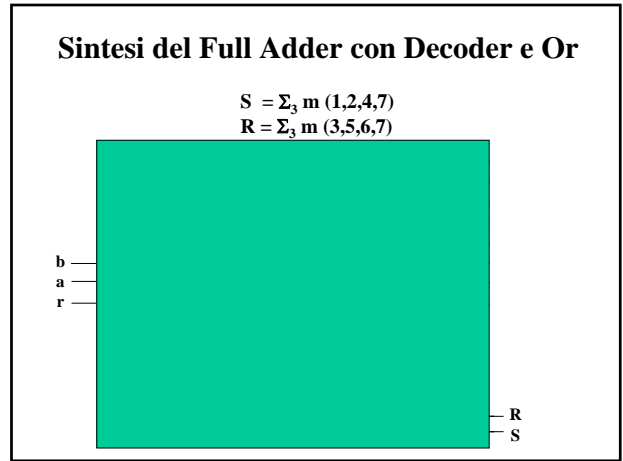
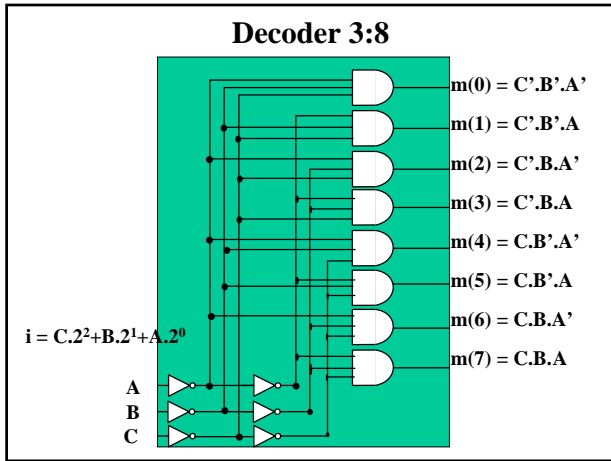
$$S(r,a,b) = \Pi_3 M(0,3,5,6)$$

$$R(r,a,b) = \Sigma_3 m(3,5,6,7)$$

$$R(r,a,b) = \Pi_3 M(0,1,2,4)$$

• **m(i)** : mintermine di n bit che assume il valore 1 solo per la n-pla di valori delle variabili corrispondente all'indice i

• **M(i)** : maxtermine di n bit che assume il valore 0 solo per la n-pla di valori delle variabili corrispondente all'indice i



**Espressioni generali**

### Teoremi di espansione (o di Shannon)

T8)  $E(x_1, x_2, \dots, x_{n-1}, x_n) = x_n' \cdot E(x_1, x_2, \dots, x_{n-1}, 0) + x_n \cdot E(x_1, x_2, \dots, x_{n-1}, 1)$

T9)  $E(x_1, x_2, \dots, x_{n-1}, x_n) = (x_n + E(x_1, x_2, \dots, x_{n-1}, 0)) \cdot (x_n' + E(x_1, x_2, \dots, x_{n-1}, 1))$

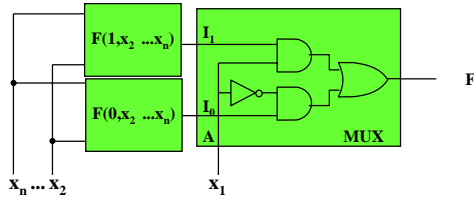
*Esempio:*

$$E = x_1 + x_2 x_3'$$

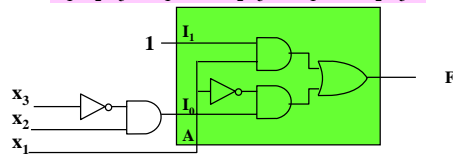
$$= x_1' \cdot (0 + x_2 x_3') + x_1 \cdot (1 + x_2 x_3')$$

$$= (x_1' + (0 + x_2 x_3')) \cdot (x_1' + (1 + x_2 x_3'))$$

### Mux e teoremi di espansione



Esempio :  $x_1 + x_2 x_3' = x_1' (0 + x_2 x_3') + x_1 (1 + x_2 x_3')$



### Applicazione iterata dei teoremi di espansione

$$\begin{aligned}
 E(x_1 x_2 x_3) &= x_1 + x_2 x_3' \\
 &= x_1' (0 + x_2 x_3') + x_1 (1 + x_2 x_3') \\
 &= x_1' x_2' x_3' (0 + 0.0) + x_1' x_2' (0 + 1.x_3') + x_1 x_2' (1 + 0.x_3') + x_1 x_2 (1 + 1.x_3') \\
 &= x_1' x_2' x_3' (0+0.0) + m(0).E(0) + x_1' x_2' x_3 (0+0.1) + m(1).E(1) + x_1' x_2 x_3' (0+1.0) + m(2).E(2) + x_1' x_2 x_3 (0+1.1) + m(3).E(3) + x_1 x_2' x_3' (1+0.0) + m(4).E(4) + x_1 x_2' x_3 (1+0.1) + m(5).E(5) + x_1 x_2 x_3' (1+1.0) + m(6).E(6) + x_1 x_2 x_3 (1+1.1) + m(7).E(7)
 \end{aligned}$$

### Espressioni generali

T10 e T11)- Ogni funzione è descritta da una espressione in cui compaiono o tutti i mintermini o tutti i maxtermini:

$$F(x_1, x_2, \dots, x_{n-1}, x_n) = \sum_{i=0}^{2^n-1} m(i) \cdot F(i) \quad \text{(SP)}$$

$$F(x_1, x_2, \dots, x_{n-1}, x_n) = \prod_{i=0}^{2^n-1} (M(i) + F(i)) \quad \text{(PS)}$$

#### Caso SP

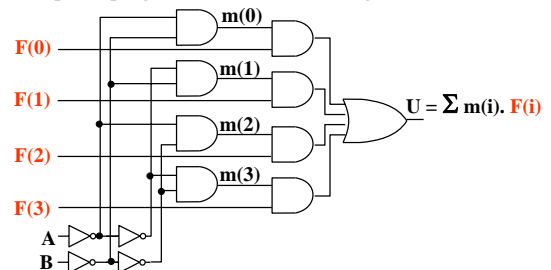
m(i) : mintermine di n bit  
F(i): **valore** dalla funzione per la n-pla di valori delle variabili per cui m(i)=1

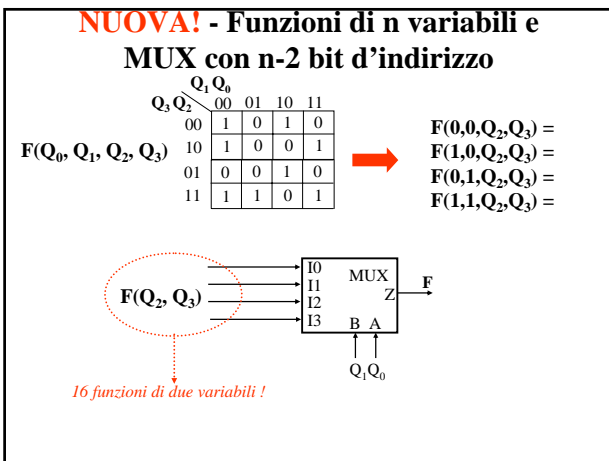
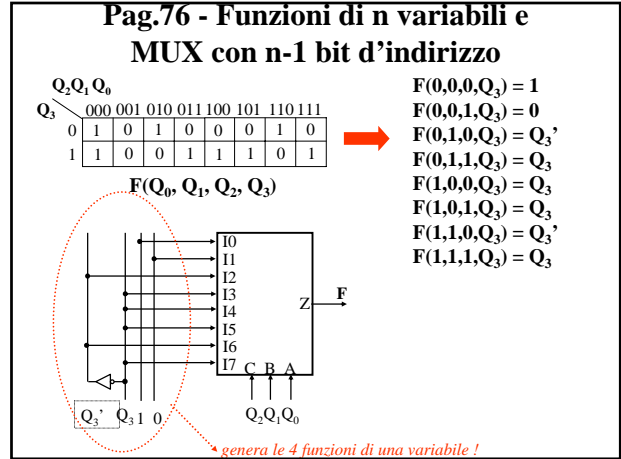
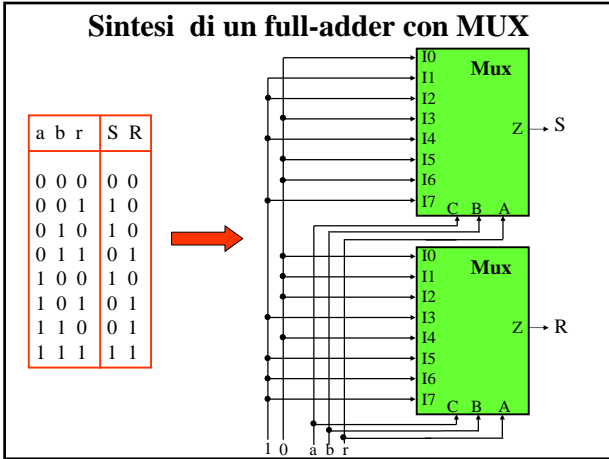
#### Caso PS

M(i) : maxtermini di n bit  
F(i): **valore** dalla funzione per la n-pla di valori delle variabili per cui M(i)=0

### La rete combinatoria programmabile

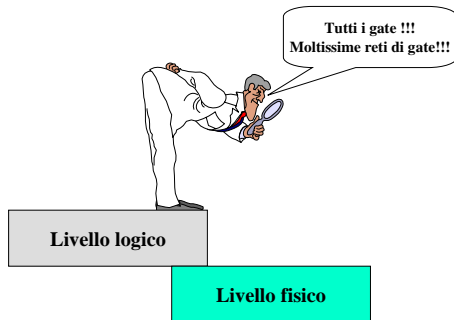
Non conviene realizzare separatamente tutte le funzioni di n variabili e scegliere di volta in volta quella che serve. E' molto più utile fare un'unica rete che le realizza tutte con una semplice "programmazione" dei suoi ingressi.







## Famiglie di circuiti logici integrati



## Full Adder con AND, OR e EX-OR

$$S = r'. a'. b + r'. a. b' + r. a'. b' + r. a. b$$

$$R = r'. a. b + r. a'. b + r. a. b' + r. a. b$$

manipolazione algebrica:

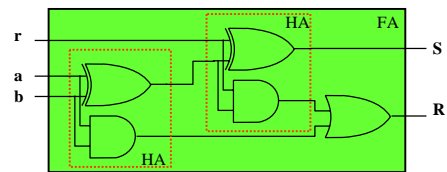
$$S = r'. (a'. b + a. b') + r. (a'. b' + a. b)$$

$$S = r'. (a \oplus b) + r. (a \oplus b)'$$

$$S = r \oplus (a \oplus b)$$

$$R = (r' + r) . a . b + r . (a' . b + a . b')$$

$$R = a . b + r . (a \oplus b)$$



## Confronto tra due numeri di n bit

$$CFR = (a_0'. b_0' + a_0. b_0) . (a_1'. b_1' + a_1. b_1) . \dots$$

CELLA:

2 AND a due ingressi

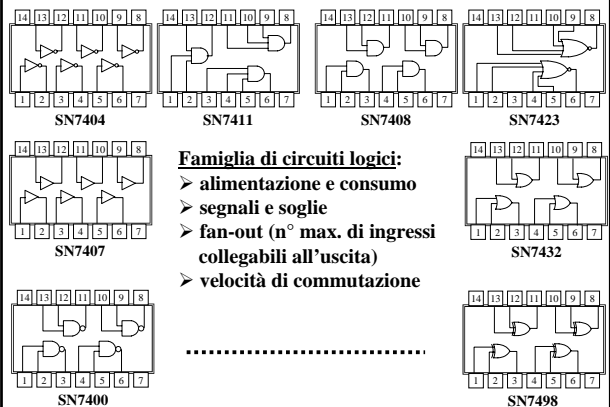
1 OR a due ingressi

2 NOT

CELLA:

1 EX-NOR a due ingressi

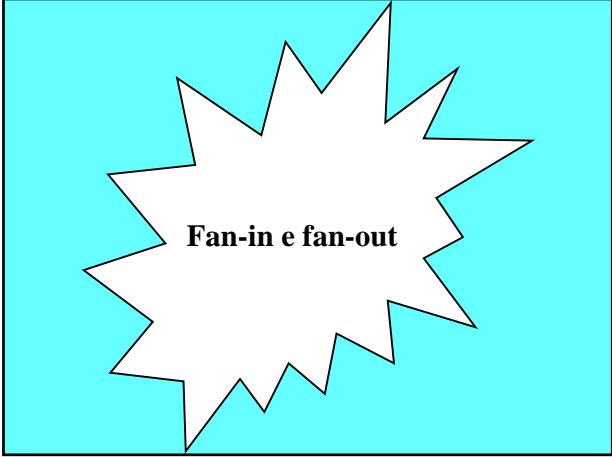
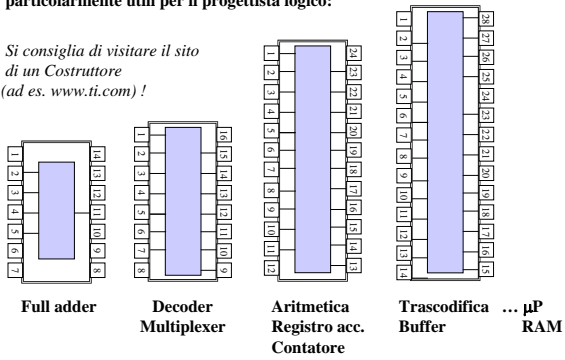
## Famiglie di gate (TTL SSI -1968/74)



### Circuiti combinatori MSI e LSI

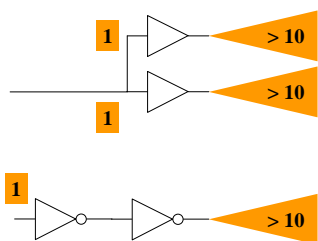
Sono disponibili come parti elementari anche reti di gate particolarmente utili per il progettista logico:

Si consiglia di visitare il sito di un Costruttore (ad es. [www.ti.com](http://www.ti.com))!

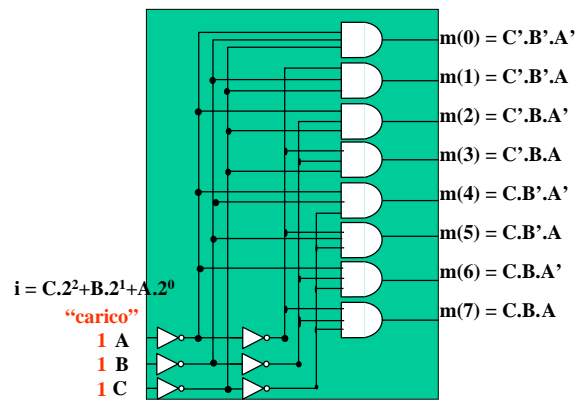


### Effetto di carico: uso di Buffer e Not

**Fan-out**  
 "l'uscita di un gate ha un numero massimo di ingressi di altri gate a cui può essere collegata"



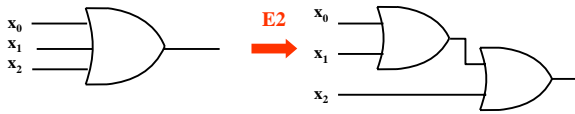
### Decoder 3:8



## And e Or: proprietà associativa

Fan-in

Gate con un massimo di otto ingressi



## Parità con EX-OR (1)

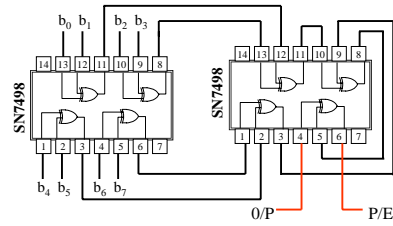
Fan-in

$$P = b_0 \oplus b_1 \oplus b_2 \oplus b_3 \oplus \dots \oplus b_7$$

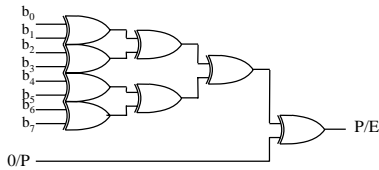
N.B. L'operazione di somma modulo due è associativa

$$P = ((b_0 \oplus b_1) \oplus (b_2 \oplus b_3)) \oplus (\dots \oplus b_7)$$

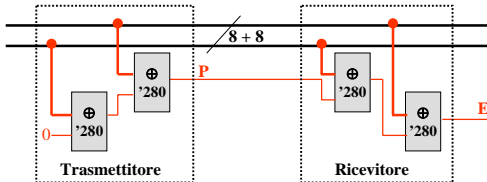
$$E = P \oplus (((b_0 \oplus b_1) \oplus (b_2 \oplus b_3)) \oplus (\dots \oplus b_7))$$



## Parità con EX-OR (2)

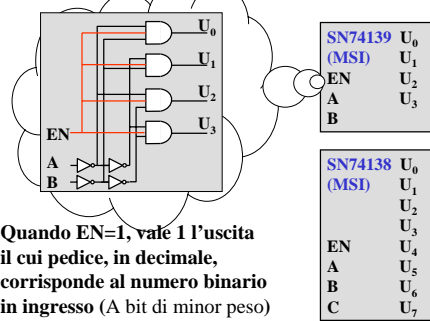


Generazione parità e rilevazione errori singoli su dati da due byte:



## Il circuito integrato DECODER

Decoder o Rete di decodifica - Rete logica combinatoria che realizza i  $2^n$  distinti mintermini di  $n$  variabili ( $n = 2, 3, 4$ )

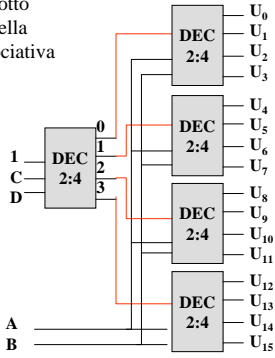


Quando  $EN=1$ , vale 1 l'uscita il cui pedice, in decimale, corrisponde al numero binario in ingresso (A bit di minor peso)

SN74154 (MSI)	$U_0$
	$U_1$
	$U_2$
	$U_3$
	$U_4$
	$U_5$
	$U_6$
	$U_7$
	$U_8$
	$U_9$
	$U_{10}$
EN	$U_{11}$
A	$U_{12}$
B	$U_{13}$
C	$U_{14}$
D	$U_{15}$

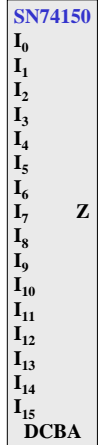
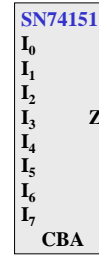
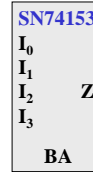
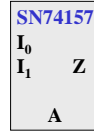
### Composizione modulare di un Decoder 4:16

N.B. - Il prodotto logico gode della proprietà associativa

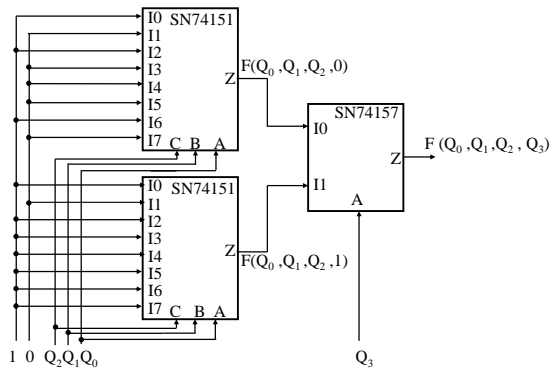


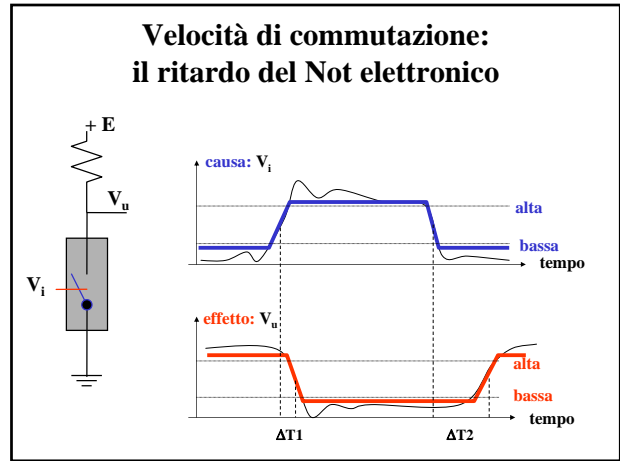
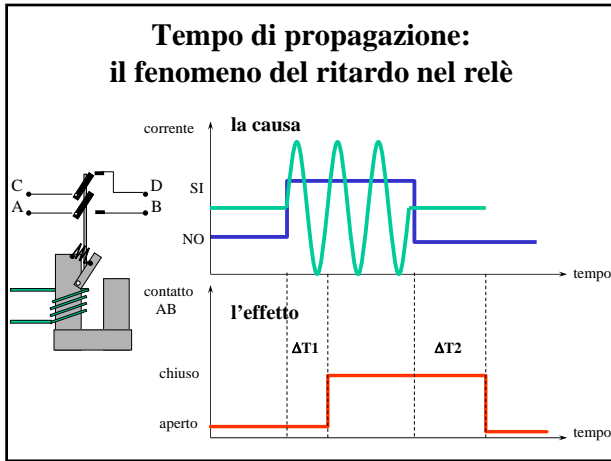
### I Multiplexer

A, B, C, D bit d'indirizzo  
I, via o bit di programmazione



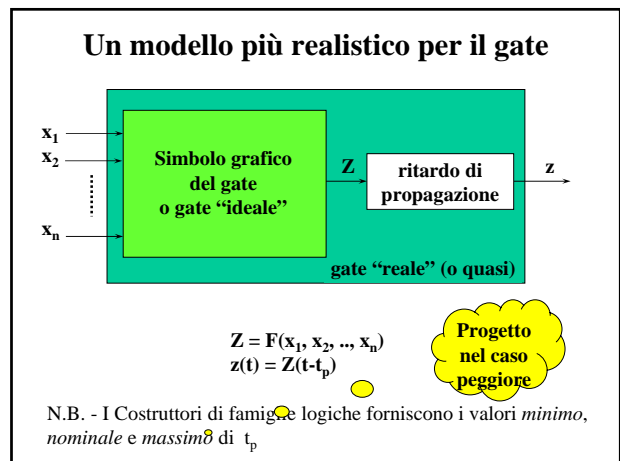
### Sintesi a MUX di funzioni di 4 variabili





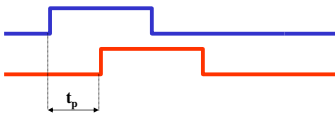
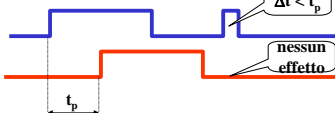
### Il ritardo sui fronti

- Il ritardo sui fronti di salita ( $\tau_{LH}$ ) e di discesa ( $\tau_{HL}$ ) è presente in ogni tipo di gate e varia in modo notevole da dispositivo a dispositivo.
- A causa della marcata differenza dei due valori, la durata di una situazione H o L in ingresso ad un gate è diversa dalla corrispondente situazione in uscita.
- A causa della “inerzia” del gate, un segnale di ingresso “impulsivo” e “troppo stretto” può non essere avvertito in uscita.



### I modelli del ritardo di propagazione

**ritardo di propagazione:  $t_p = \max(\tau_{LH}, \tau_{HL})$**

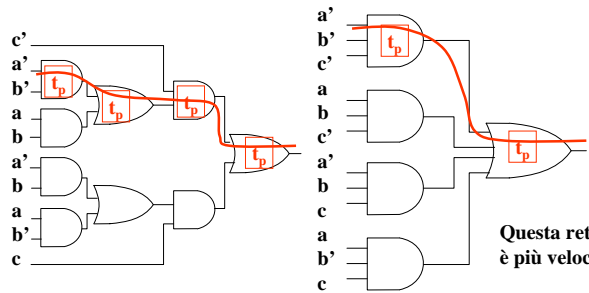
- Ritardo puro
 
- Ritardo inerziale
 

**Durata minima di un valore H o L: 3-4  $t_p$**



### Velocità e lunghezza dei percorsi

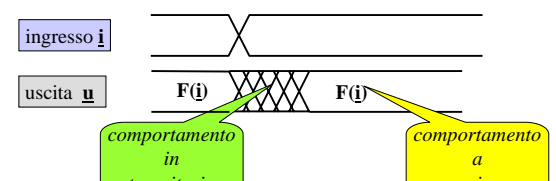
$(a'.b'+a.b).c'+(a'.b+a.b').c = a'.b'.c' + a.b.c' + a'.b.c + a.b'.c$



Questa rete è più veloce

### Comportamento a regime e in transitorio dei circuiti combinatori

I nuovi valori dei segnali di ingresso di una rete combinatoria devono propagarsi all'interno della struttura prima di riuscire ad imporre al segnale d'uscita il valore che ad essi deve corrispondere. Ciò determina un comportamento in transitorio, che in generale sarà diverso da quello a regime.



comportamento in transitorio

comportamento a regime

