

## Il problema della sintesi

Assegnata una qualsiasi funzione di variabili binarie, è possibile descriverla con una espressione contenente solo le operazioni eseguite dai gate?

## Algebre binarie

**Algebra binaria** - Sistema matematico formato da un insieme di operatori definiti assiomaticamente ed atti a descrivere con una espressione ogni funzione di variabili binarie

Calcolo delle proposizioni Crisippo (250 a.c.)  
{vero, falso} {e, o, non} G. Boole (1854)  
tre operatori

**Algebra di commutazione**  
{0, 1} {+, ., '}  
tre operatori

C. Shannon (1938)

Algebra del nand  
{0, 1} {↑}  
un operatore

Algebra del nor  
{0, 1} {↓}  
un operatore

Algebra lineare  
{0, 1} {⊕, ⊙}  
due operatori

## 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:

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

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

## Teoremi di equivalenza

## 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

N.B. le altre configurazioni sono per ipotesi impossibili

$x_2$	$x_1$	$x_0$	$z_1$	$z_0$
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	-	-

## 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

## Espressioni canoniche

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

**1ª forma canonica** - Ogni funzione di  $n$  variabili è descritta da una **somma di tanti prodotti logici** quante sono le configurazioni per cui vale. In ciascun prodotto, **minitermine**, appare ogni variabile, in forma vera se nella configurazione corrispondente vale 1, in forma complementata se vale 0.

Completezza  
dell'algebra  
di commutazione

Completezza  
del Calcolo  
delle proposizioni

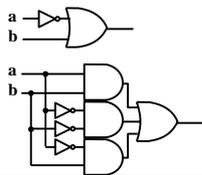
T7) **2ª forma canonica** - Ogni funzione di  $n$  variabili è descritta da un **prodotto di tante somme logiche** quante sono le configurazioni per cui vale. In ciascuna somma, o **maxitermine**, appare ogni variabile, in forma vera se nella configurazione corrispondente vale 1, in forma complementata se vale 0.

### 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:

$$\begin{aligned}
 F(a,b) &= a' \cdot b' + a' \cdot b + a \cdot b \\
 &= a' \cdot (b' + b) + a \cdot b \\
 &= a' \cdot 1 + a \cdot b \\
 &= a' + a \cdot b \\
 &= a' + a \cdot b + a' \cdot b \\
 &= a' + b
 \end{aligned}$$

**E3**

**E8**

**E5**

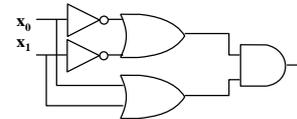
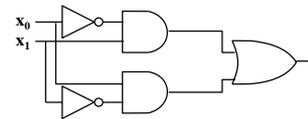
**una parte è inclusa nel tutto**  
**E3, E8, E5**

### EX-OR

$$x_0 \oplus x_1 = x_0'x_1 + x_0x_1'$$

x <sub>0</sub>	x <sub>1</sub>	x <sub>0</sub> ⊕x <sub>1</sub>
0	0	0
0	1	1
1	0	1
1	1	0

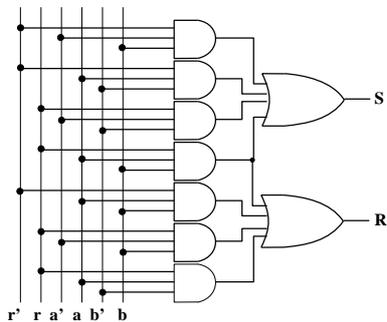
$$x_0 \oplus x_1 = (x_0' + x_1') \cdot (x_0 + x_1)$$



### Full Adder

$$\begin{aligned}
 S &= r' \cdot a' \cdot b + r' \cdot a \cdot b' + r \cdot a' \cdot b' + r \cdot a \cdot b \\
 R &= r' \cdot a \cdot b + r \cdot a' \cdot b + r \cdot a \cdot b' + r \cdot a \cdot b
 \end{aligned}$$

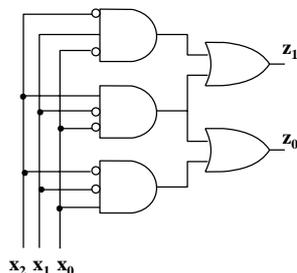
r	a	b	R	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1



### Espressioni canoniche di funzioni incomplete

x <sub>2</sub>	x <sub>1</sub>	x <sub>0</sub>	z <sub>1</sub>	z <sub>0</sub>
0	0	0	0	0
1	0	0	1	1
0	1	0	1	0
0	0	1	0	1

N.B. le altre configurazioni sono per ipotesi impossibili



$$\begin{aligned}
 z_1 &= x_2 x_1' x_0' + x_2' x_1 x_0' + x_2 x_1 x_0 + \dots \\
 z_0 &= x_2 x_1' x_0' + x_2' x_1' x_0 + \dots
 \end{aligned}$$

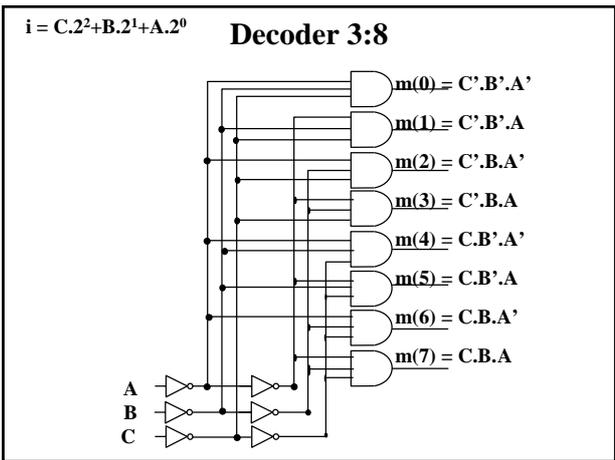
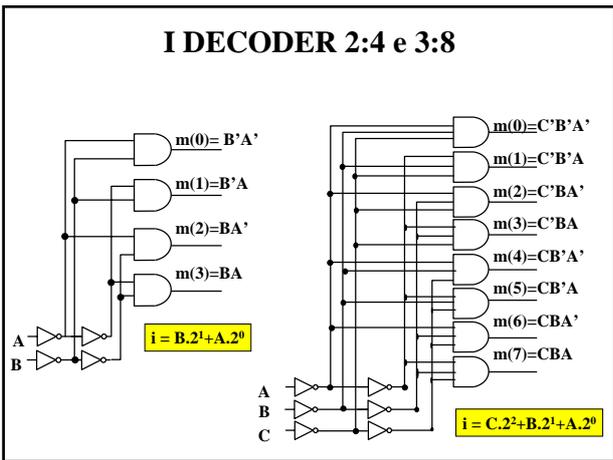
**Notazioni simboliche**

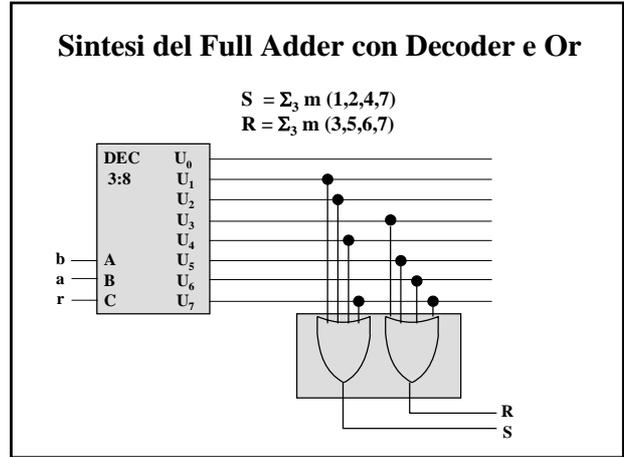
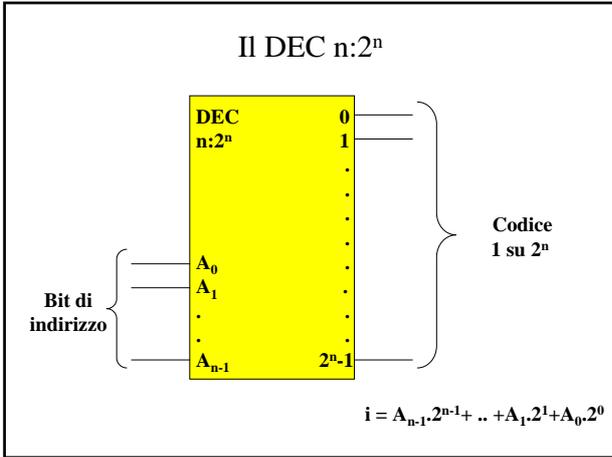
### Notazioni simboliche per le espressioni canoniche

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





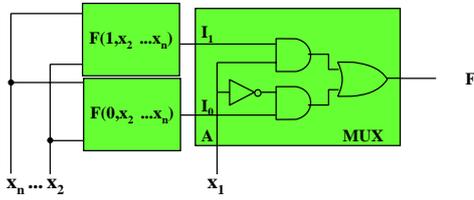
**Teoremi di espansione (o di Shannon)**

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

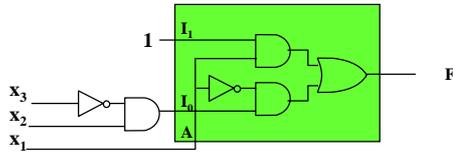
**T9)  $E(x_1, x_2, \dots, x_{n-1}, x_n) = (x_n + E(x_1, x_2, \dots, x_{n-1}, 0)) \cdot (x_n' + E(x_1, x_2, \dots, x_{n-1}, 1))$**

*Esempio:*  
 $E = x_1 + x_2 x_3'$   
 $= x_1' \cdot (0 + x_2 x_3') + x_1 \cdot (1 + x_2 x_3')$   
 $= (x_1' + (0 + x_2 x_3')) \cdot (x_1' + (1 + x_2 x_3'))$

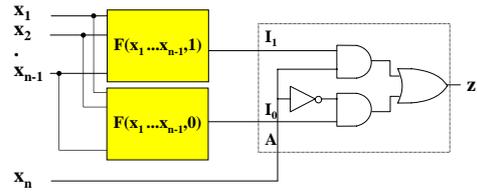
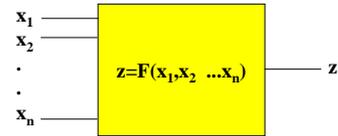
### Mux e teoremi di espansione



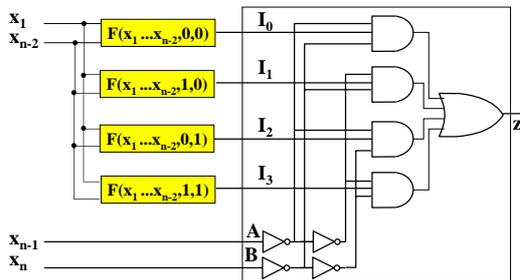
Esempio :  $x_1 + x_2 x_3' = x_1' (0 + x_2 x_3') + x_1 (1 + x_2 x_3')$



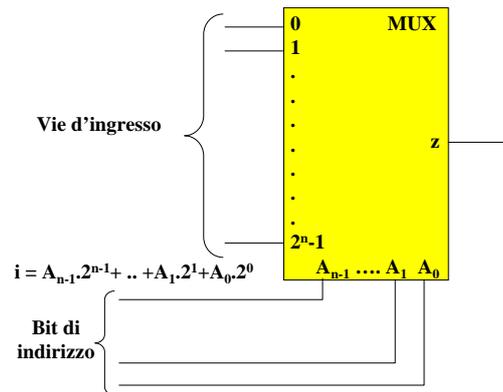
### La decomposizione indotta da T8



### Estrazione di due variabili con T8



### Il MUX con n bit d'indirizzo e 2^n vie d'ingresso



### Applicazione iterata dei teoremi di espansione

$$\begin{aligned}
 E(x_1, x_2, x_3) &= x_1 + x_2 + x_3 \\
 &= x_1'(0 + x_2 x_3') + x_1(1 + x_2 x_3') \\
 &= x_1' x_2'(0 + 0 \cdot x_3') + x_1' x_2(0 + 1 \cdot x_3') + x_1 x_2'(1 + 0 \cdot x_3') + x_1 x_2(1 + 1 \cdot x_3') \\
 &= x_1' x_2' x_3'(0 + 0 \cdot 0) + x_1' x_2' x_3(0 + 0 \cdot 1) + x_1' x_2 x_3'(0 + 1 \cdot 0) + x_1' x_2 x_3(0 + 1 \cdot 1) + x_1 x_2' x_3'(1 + 0 \cdot 0) + x_1 x_2' x_3(1 + 0 \cdot 1) + x_1 x_2 x_3'(1 + 1 \cdot 0) + x_1 x_2 x_3(1 + 1 \cdot 1) \\
 &= m(0) \cdot E(0) + m(1) \cdot E(1) + m(2) \cdot E(2) + m(3) \cdot E(3) + m(4) \cdot E(4) + m(5) \cdot E(5) + m(6) \cdot E(6) + m(7) \cdot E(7)
 \end{aligned}$$

### Espressioni generali

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

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

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

#### Caso SP

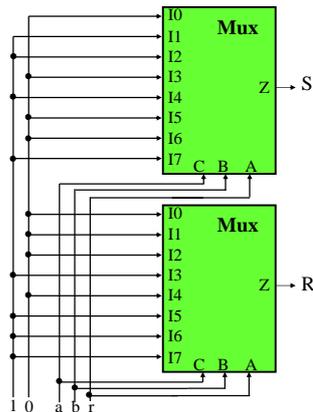
$m(i)$  : mintermine di n bit  
 $F(i)$  : valore dalla funzione per la n-pla di valori delle variabili per cui  $m(i)=1$

#### Caso PS

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

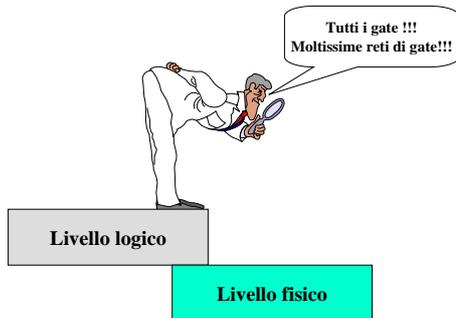
### Sintesi di un full-adder con MUX

a	b	r	S	R
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1



### 4.3 Famiglie logiche

## 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:

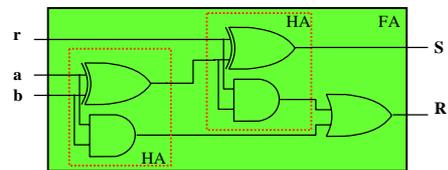
$$S = r'. (a'. b + a . b') + r . (a'. b' + a . b)$$

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

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

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

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



## 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

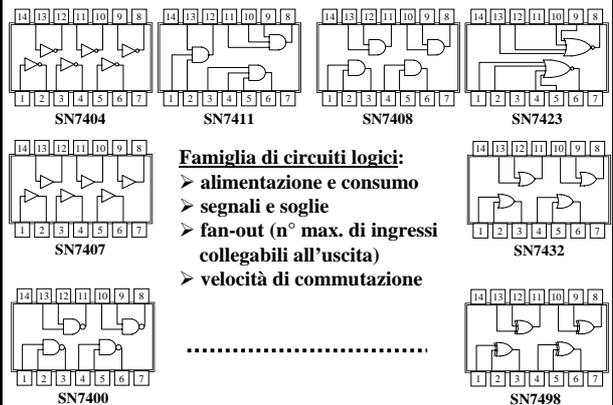
1 OR a due ingressi

2 NOT

CELLA:

1 EX-NOR a due ingressi

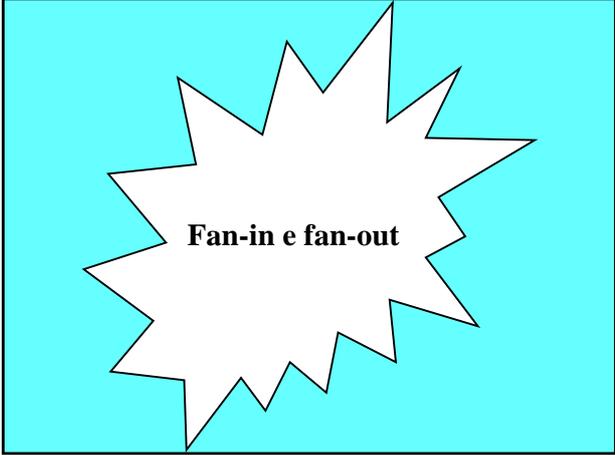
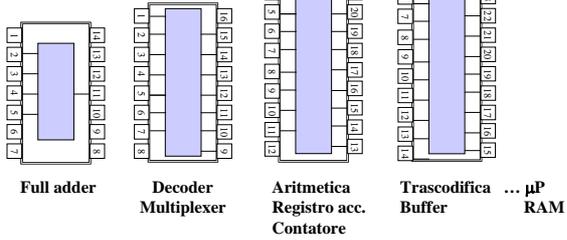
## Famiglie di gate (TTL SSI -1968/74)



### Circuiti combinatori MSI e LSI

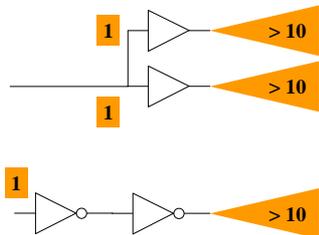
Sono disponibili come parti elementari anche reti di gate particolarmente utili per il progettista logico:

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

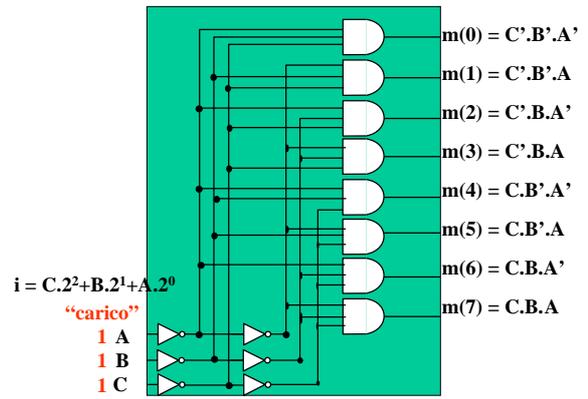


### Effetto di carico: uso di Buffer e Not

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



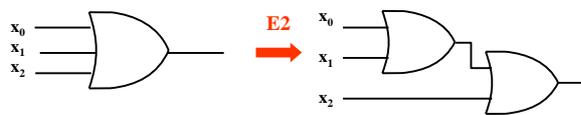
### Decoder 3:8



### And e Or: proprietà associativa

Fan-in

Gate con un massimo di otto ingressi



### Parità con EX-OR (1)

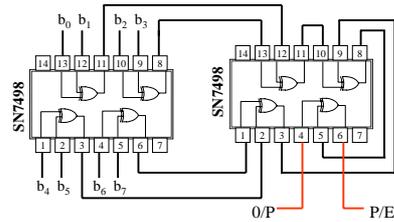
Fan-in

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

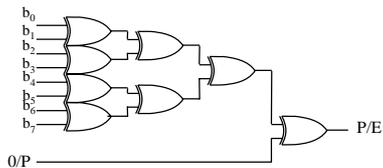
N.B. L'operazione di somma modulo due è associativa

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

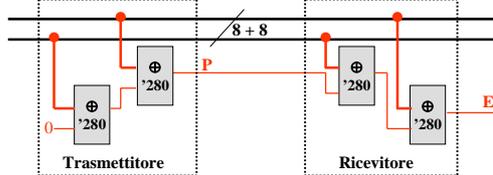
$$E = P \oplus (((b_0 \oplus b_1) \oplus (b_2 \oplus b_3)) \oplus (\dots \oplus b_7))$$

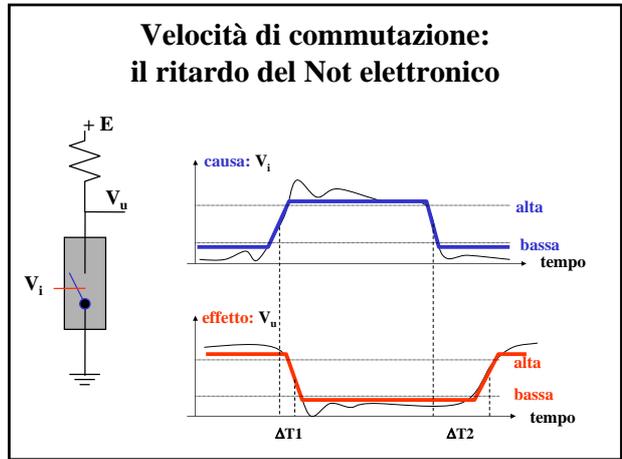
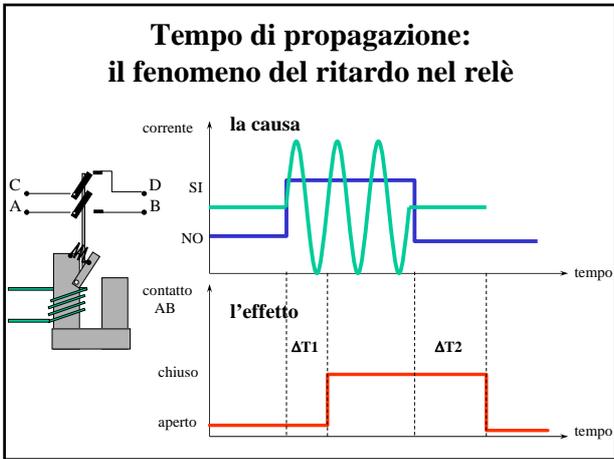


### Parità con EX-OR (2)



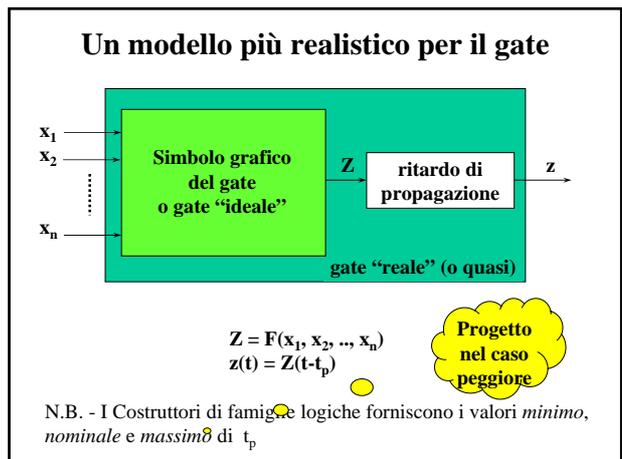
Generazione parità e rilevazione errori singoli su dati da due byte:





### Il ritardo sui fronti

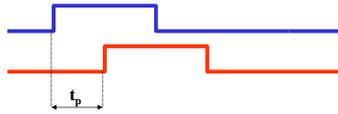
- Il ritardo sui fronti di salita ( $\tau_{LH}$ ) e di discesa ( $\tau_{HL}$ ) è presente in ogni tipo di gate e varia in modo notevole da dispositivo a dispositivo.
- A causa della marcata differenza dei due valori, la durata di una situazione H o L in ingresso ad un gate è diversa dalla corrispondente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



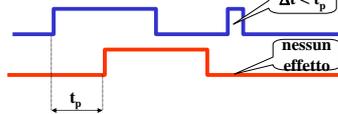
### I modelli del ritardo di propagazione

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

- Ritardo puro



- Ritardo inerziale

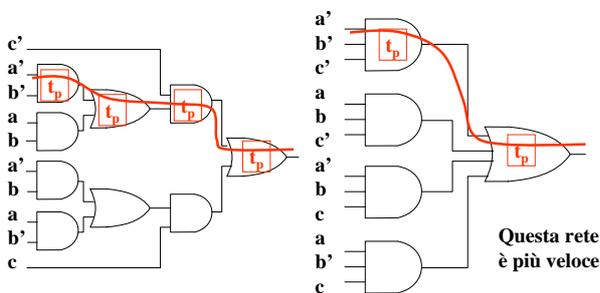


**Durata minima di un valore H o L:**  $3-4 t_p$

Comportamento  
in transitorio

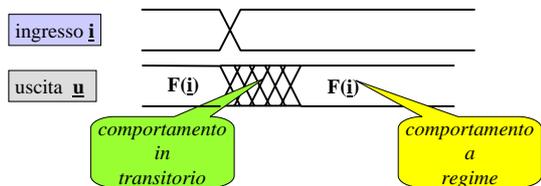
### Velocità e lunghezza dei percorsi

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

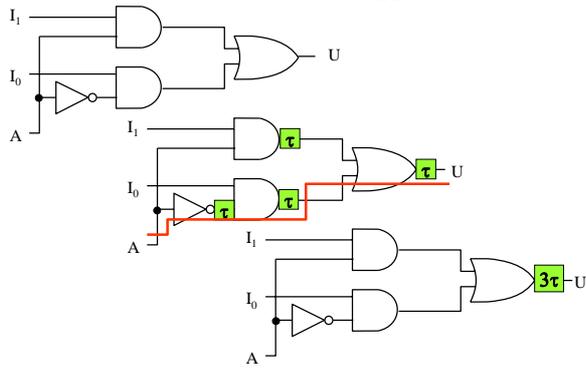


### Comportamento a regime e in transitorio dei circuiti combinatori

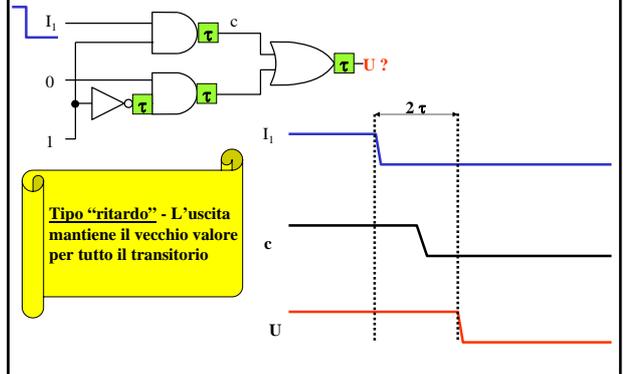
I nuovi valori dei segnali di ingresso di una rete combinatoria devono propagarsi all'interno della struttura prima di riuscire ad imporre al segnale d'uscita il valore che ad essi deve corrispondere. Ciò determina un comportamento in transitorio, che in generale sarà diverso da quello a regime.



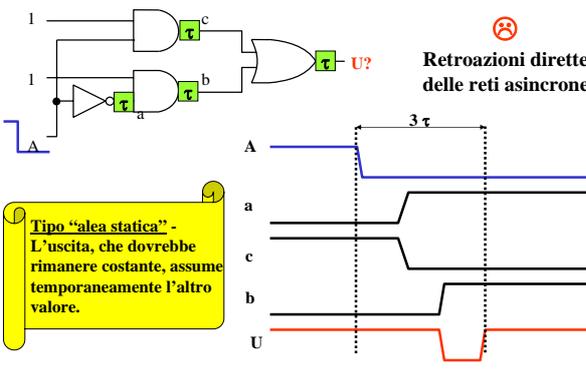
### Stima della durata del transitorio (metodo del caso peggiore)



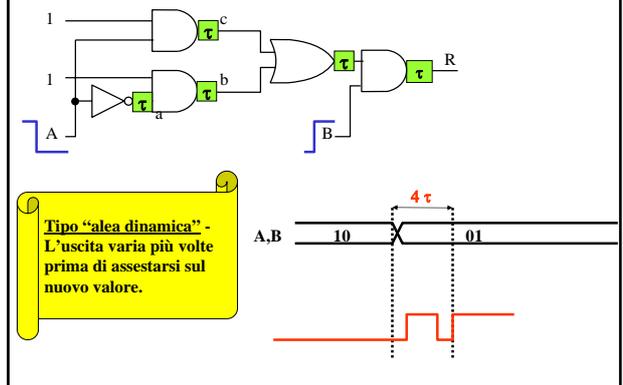
### Tipi di transitorio: il ritardo

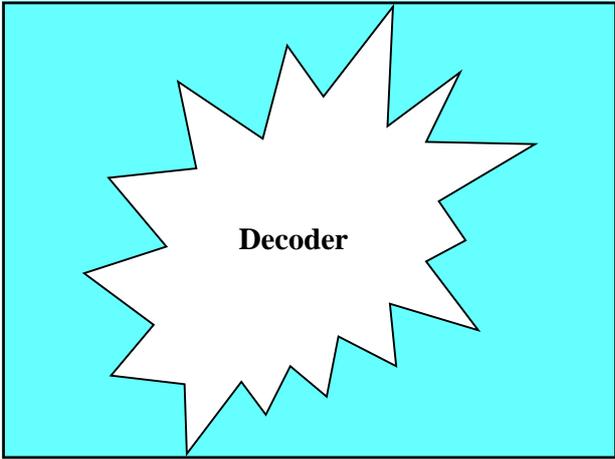


### Tipi di transitorio: l'alea statica



### Tipi di transitorio: l'alea dinamica





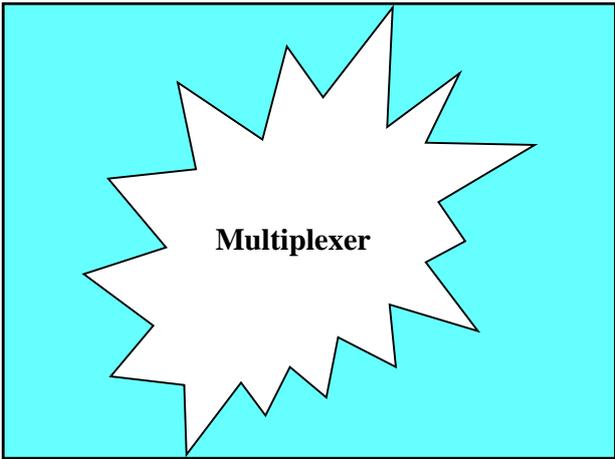
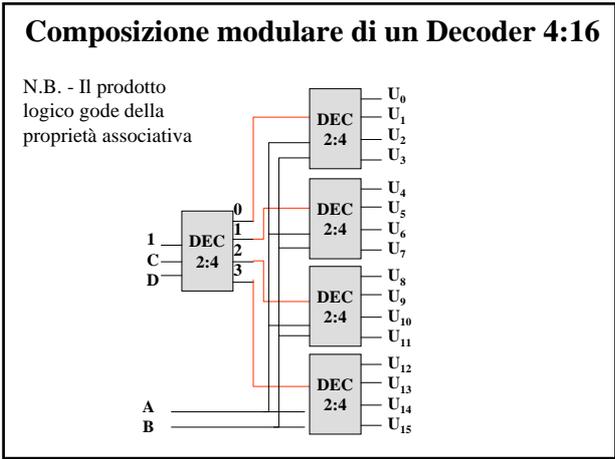
### Il circuito integrato DECODER

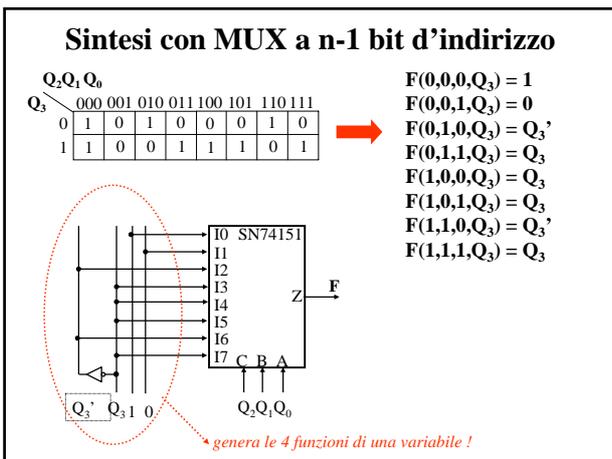
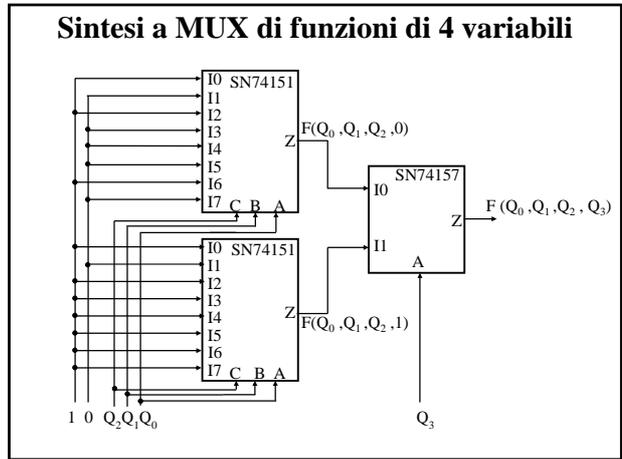
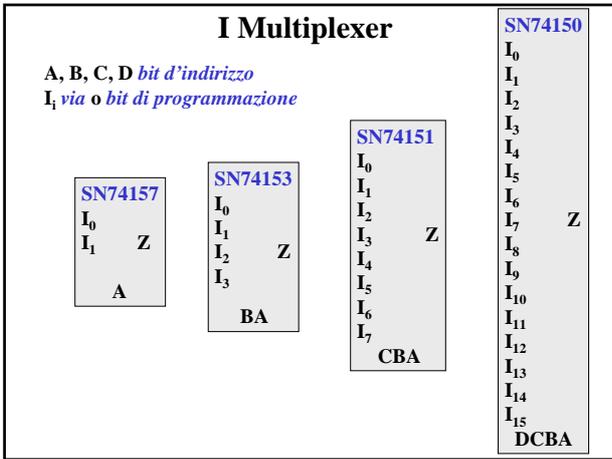
**Decoder** o Rete di decodifica - Rete logica combinatoria che realizza i  $2^n$  distinti mintermini di  $n$  variabili ( $n = 2,3,4$ )

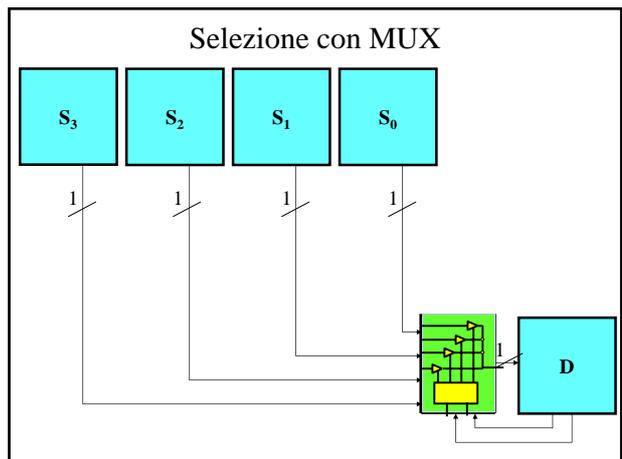
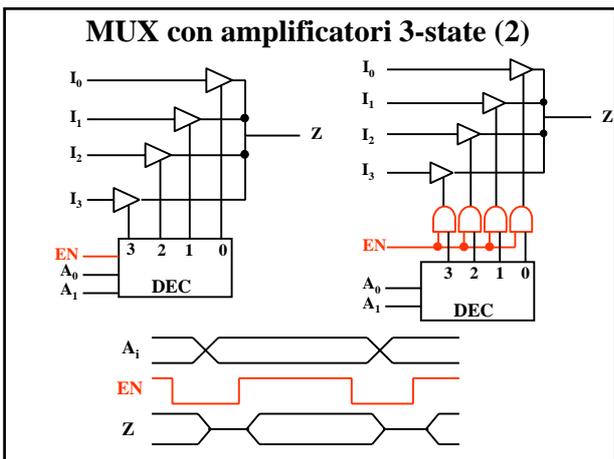
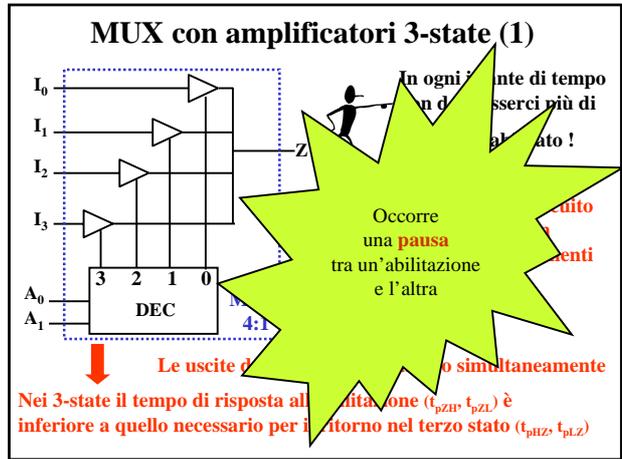
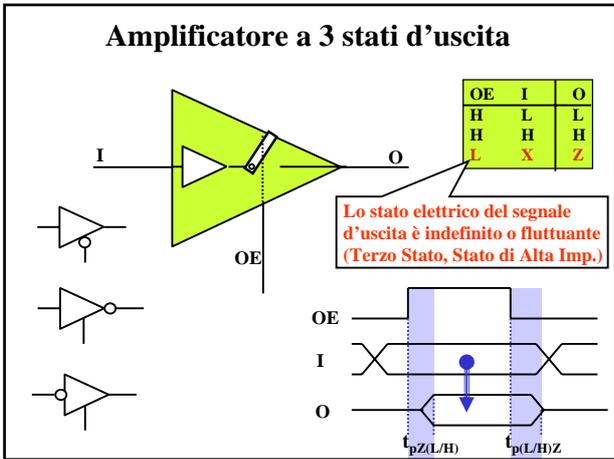
SN74139 (MSI)	U <sub>0</sub>	U <sub>1</sub>	U <sub>2</sub>	U <sub>3</sub>
EN	U <sub>4</sub>	U <sub>5</sub>	U <sub>6</sub>	U <sub>7</sub>
A	U <sub>8</sub>	U <sub>9</sub>	U <sub>10</sub>	U <sub>11</sub>
B	U <sub>12</sub>	U <sub>13</sub>	U <sub>14</sub>	U <sub>15</sub>

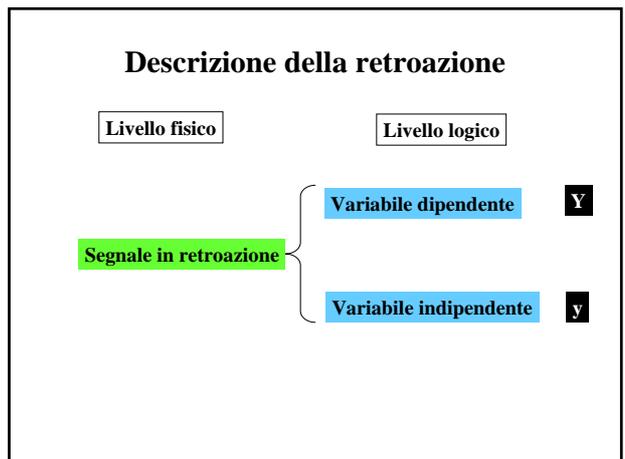
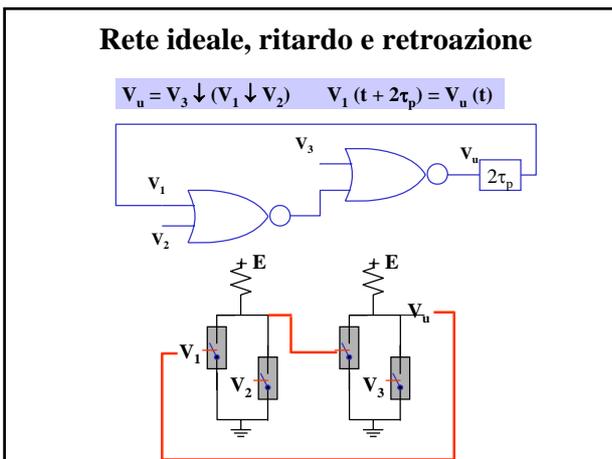
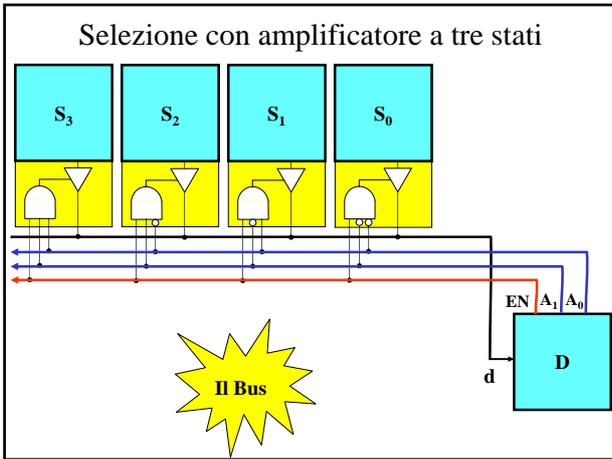
SN74138 (MSI)	U <sub>0</sub>	U <sub>1</sub>	U <sub>2</sub>	U <sub>3</sub>	U <sub>4</sub>	U <sub>5</sub>	U <sub>6</sub>	U <sub>7</sub>
EN	U <sub>8</sub>	U <sub>9</sub>	U <sub>10</sub>	U <sub>11</sub>	U <sub>12</sub>	U <sub>13</sub>	U <sub>14</sub>	U <sub>15</sub>
A	U <sub>16</sub>	U <sub>17</sub>	U <sub>18</sub>	U <sub>19</sub>	U <sub>20</sub>	U <sub>21</sub>	U <sub>22</sub>	U <sub>23</sub>
B	U <sub>24</sub>	U <sub>25</sub>	U <sub>26</sub>	U <sub>27</sub>	U <sub>28</sub>	U <sub>29</sub>	U <sub>30</sub>	U <sub>31</sub>
C	U <sub>32</sub>	U <sub>33</sub>	U <sub>34</sub>	U <sub>35</sub>	U <sub>36</sub>	U <sub>37</sub>	U <sub>38</sub>	U <sub>39</sub>

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









### Latch SR a NOR

$Q = R \downarrow (q \downarrow S)$   
 $Q = (R + (q + S))'$   
 $Q = R' \cdot (q + S)$

$V_2=V_3=1$  vietato!  
 Pongo  $S = V_2$   
 $R = V_3$   
 $Q = V_u$

$V_u = V_3 \downarrow (V_1 \downarrow V_2)$

q	S, R		
	00	01	11 10
0	0	0	- 1
1	1	0	- 1

**Q**

### Cosa occorre per scrivere un 1 (o uno 0)?

$\Delta t = 2\tau_p$

S: 0 → 1

Q: 0 → 1

q: 0 → 1

Dopo  $\Delta t$  dal fronte di salita di S, q passa da 0 a 1, condizione che si mantiene anche se S torna a 0

La durata minima di un comando di set/reset è spesso indicata con la denominazione di *tempo di set-up* del latch

### 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

**I<sup>a</sup> 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":**

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

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

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