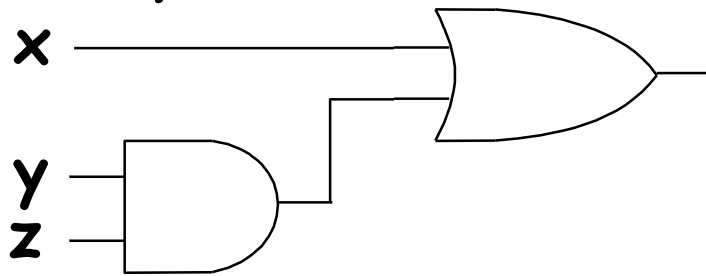


## 3.3 Reti di costo minimo

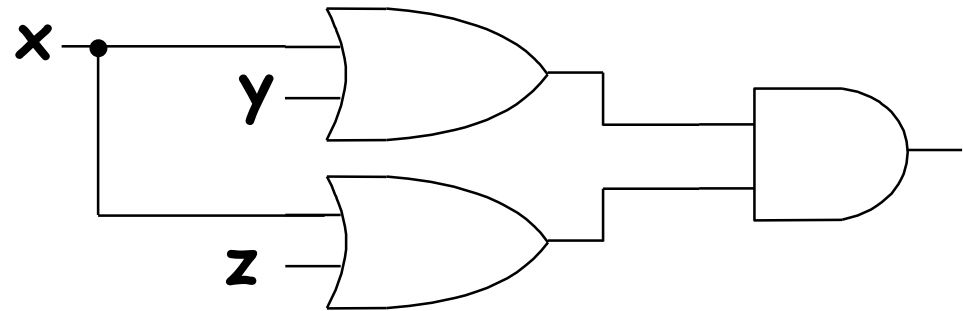
# Complessità e velocità

- Per valutare la complessità di una rete logica in termini di complessità e velocità si utilizzano 3 indicatori :
  - $N_{gate}$  = numero di gate,
  - $N_{conn}$  = numero di connessioni
  - $N_{casc}$  = max numero di gate disposti in cascata
- **Complessità**  $\Rightarrow$  funzione crescente di  $N_{gate}$  ,  $N_{conn}$
- **Velocità di elaborazione**  $\Rightarrow$  funzione decrescente di  $N_{casc}$

## Esempio:



$x + yz$



$(x+y)(x+z)$

- Le due reti sono equivalenti (prop. distrib., E3).
- Hanno la stessa velocità di elaborazione.
- La rete di sinistra è meno complessa.

# Schemi logici di "costo minimo"

## IPOSTESI:

- ingressi disponibili in forma vera e complementata
- fan-in grande quanto serve

Rete combinatoria di costo minimo (tipo SP e tipo PS) -  
Schema logico che realizza una funzione qualsiasi con

1. non più di 2 gate in cascata tra ingressi e uscita
2. minimo numero di gate
3. minimo numero di ingressi per gate.

N.B. - Il numero di gate e/o di connessioni della rete di costo minimo di tipo SP è in generale diverso da quello della rete di costo minimo di tipo PS che realizza la stessa funzione.

# Espressioni normali e minime

Espressione normale - Qualsiasi espressione del tipo SP o PS

Espressione minima - Espressione normale che descrive algebricamente una rete di costo minimo, ovvero formata dal minimo numero possibile di "termini" (prodotti/somme) aventi ciascuno il minimo numero possibile di "letterali" (variabili in forma vera o complementata).

- Il numero di termini dell'espressione determinerà il numero di gate utilizzati ( $N_{gate}$ ) mentre il numero di letterali coinvolti in tali termini determinerà il numero di connessioni ( $N_{conn}$ )
- E' possibile che **più espressioni normali dello stesso tipo siano minime** (abbiano cioè eguali valori di  $N_{gate}$  e  $N_{conn}$ ).

# Implicanti e implicati primi

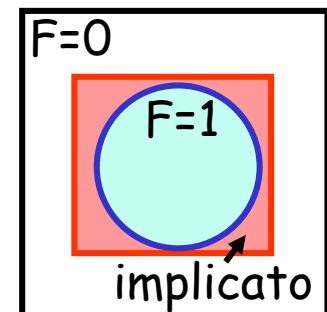
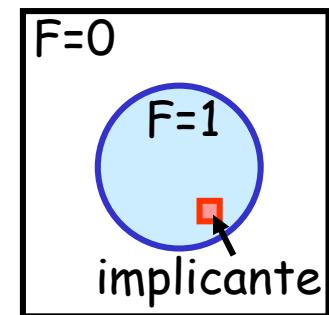
Implicante di una funzione - Termine **prodotto** di **n o meno variabili** che assume valore 1 per configurazioni per cui anche la funzione vale 1.

Implicante primo - Implicante da cui non è possibile eliminare alcun letterale senza perdere la sua proprietà.

Implicato di una funzione - Termine **somma** di **n o meno variabili** che assume valore 0 per configurazioni per cui anche la funzione vale 0.

Implicato primo - Implicato da cui non è possibile eliminare alcun letterale senza perdere la sua proprietà.

- Un implicante rappresenta dunque un certo numero di mintermini, ovvero un certo numero di configurazioni delle  $n$  variabili d'ingresso per le quali la funzione assume valore 1
- Similmente, un implicato rappresenta un certo numero di maxtermini, ovvero un certo numero di configurazioni delle  $n$  variabili d'ingresso per le quali la funzione assume valore 0
- Gli implicanti (implicati) primi di una funzione si individuano a partire dalla prima (seconda) forma canonica applicando in ogni maniera possibile e finché è possibile la proprietà di **combinazione (E9)**.



# Es. - Encoder a 3 ingressi (3:2)

- Avevamo precedente determinato due espressioni equivalenti per l'encoder a 3 bit di ingresso:

$x_3$	$x_2$	$x_1$	$z_1$	$z_0$
0	0	0	0	0
0	0	1	0	1
0	1	0	1	0
1	0	0	1	1

$$z_0 = \underbrace{x_1' x_2' x_3}_{\text{mintermine}} + \underbrace{x_1 x_2' x_3'}_{\text{mintermine}}$$

Sia implicanti, sia mintermini  
(espressione canonica)

$$z_1 = \underbrace{x_1' x_2' x_3}_{\text{implicant}} + \underbrace{x_1' x_2 x_3}_{\text{implicant}}$$

$$z_0 = \underbrace{x_1}_{\text{implicant}} + \underbrace{x_3}_{\text{implicant}}$$

Implicanti ma NON mintermini  
(espressione normale NON canonica)

$$z_1 = \underbrace{x_2}_{\text{implicant}} + \underbrace{x_3}_{\text{implicant}}$$

# Condizioni necessarie

Espressione irridondante - Espressione normale SP o PS da cui non può essere eliminato alcun termine senza invalidare l'equivalenza con l'espressione stessa.

V) L'espressione minima SP è una somma irridondante di implicanti primi.

VI) L'espressione minima PS è un prodotto irridondante di di implicati primi.

N.B - E9 e E11 (combinazione e consenso) sono gli "strumenti" che consentono di passare per manipolazione algebrica dall'espressione canonica a quella minima.

# Metodi per la determinazione dell'espressione minima

## Metodi algoritmici

(Quine-Mc Cluskey, Petrick)

*consentono di trattare funzioni con un numero qualsiasi di variabili e vengono tipicamente eseguiti da un calcolatore.*

## Metodo grafico

(Mappe di Karnaugh)

*consente di trattare agevolmente funzioni fino a 6 variabili.  
e viene eseguito manualmente.*



## 3.4 Mappe di Karnaugh

# Mappe

Mappa di Karnaugh - Rappresentazione bidimensionale della tabella della verità di una funzione di 2,3,4 variabili, i cui valori sono elencati sui bordi in maniera tale che due configurazioni consecutive differiscano per il valore di un solo bit.

## Esempi:

	<b>b</b>	0	1
<b>a</b>	0	0	1
	1	1	1

*Somma di  
2 variabili*

	<b>br</b>	00	01	11	10
<b>a</b>	0	0	0	1	0
	1	0	1	1	1

*Riporto del  
Full Adder*

	<b>cd</b>	00	01	11	10
<b>ab</b>	00	0	1	0	1
	01	1	0	1	0
	11	0	1	0	1
	10	1	0	1	0

*Parità su  
4 variabili*

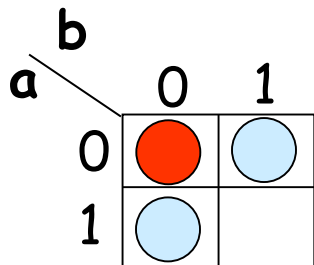
# Adiacenza tra celle

Coppia di celle adiacenti su mappe di Karnaugh : due celle le cui coordinate differiscono per un solo bit.  
In una mappa che descrive una funzione di  $n$  variabili ogni cella ha  $n$  celle adiacenti.

Regola grafica per l'adiacenza - Sono adiacenti celle aventi un lato in comune o poste all'estremità di una stessa riga o colonna.

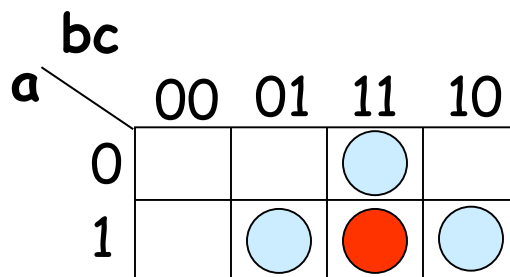
● cella scelta come esempio

○ celle adiacenti



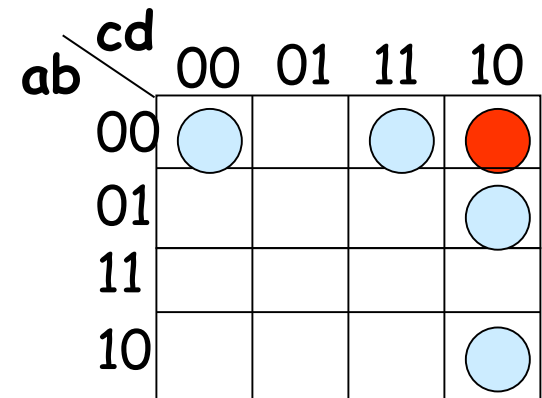
2 variabili

$ab=00$  adiac. a:  
 $01, 10$



3 variabili

$abc=111$  adiac. a:  
 $101, 011, 110$

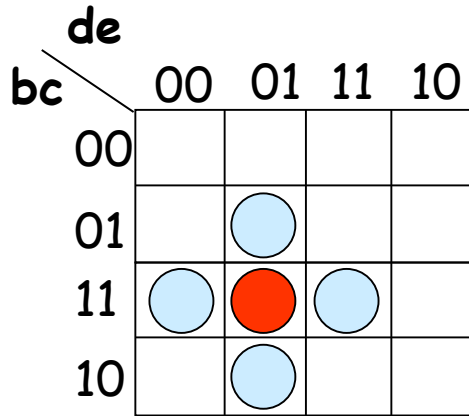


4 variabili

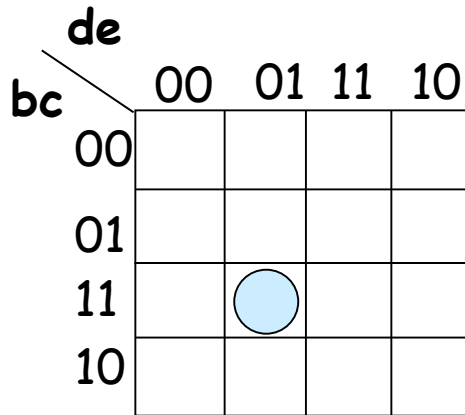
$abcd=0010$  adiac. a:  $0011, 0110$   
ma anche:  $0000, 1010$

Le mappe permettono di identificare graficamente (dunque in maniera agevole) configurazioni adiacenti, in particolare quelle aventi il medesimo valore di uscita (il che sarà particolarmente utile ai fini della semplificazione algebrica)

# Estensione delle mappe a 5 e a 6 variabili



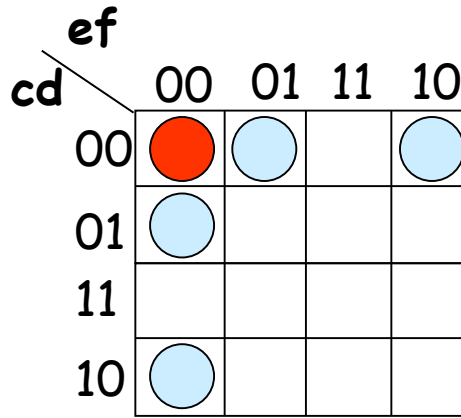
**a=0**



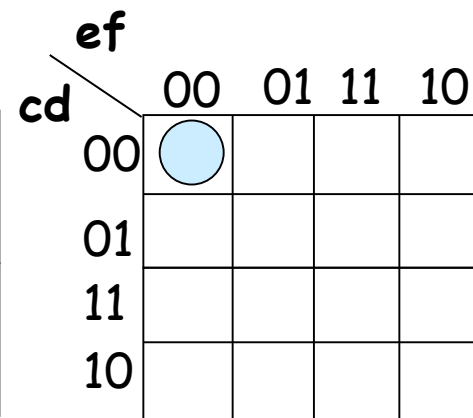
**a=1**

**5 variabili**

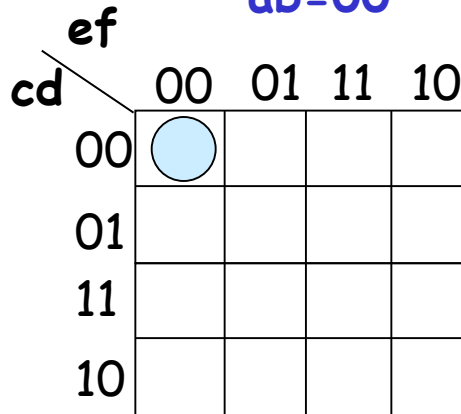
*abcde=0 1101* adiac. *a*:  
*0 1100*, *0 0101*, *0 1001*, *0 1111*  
 ma anche: *1 1101*



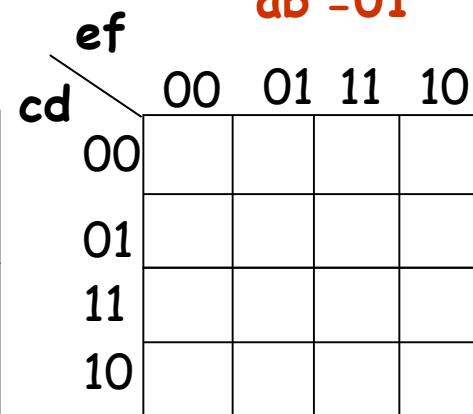
**ab=00**



**ab=01**



**ab=10**



**ab=11**

**6 variabili**

*abcdef=00 0000* adiac. *a*:  
*00 0100*, *00 0001*, *00 1000*, *00 0010*  
 ma anche: *10 0000*, *01 0000*

Ulteriore regola di adiacenza - Sono adiacenti celle che occupano la stessa posizione in sotto-mappe adiacenti.



**Ricerca  
degli implicanti  
e degli implicati**

# Manipolazione algebrica per via grafica (1)

		cd			
ab		00	01	11	10
00	X	X	X	X	
01	X	X	X	X	
11	X	X	X	1	
10	X	X	X	1	

canonica SP:  
 $ab'cd' + abcd' + \dots$

**E9**

normale SP:  $acd' + \dots$

		cd			
ab		00	01	11	10
00	X	X	X	X	
01	X	X	X	X	
11	X	0	0	X	
10	X	X	X	X	

canonica PS:  $(a'+b'+c+d')(a'+b'+c'+d')$

...

**E9**

normale PS:  $(a'+b'+d')\dots$

Due termini di una espressione canonica (SP o PS) corrispondenti a configurazioni che individuano celle adiacenti sono equivalenti ad un unico termine con un letterale in meno.

# Funzioni incomplete

		cd			
	ab	00	01	11	10
00	X	X	X	X	
01	X	X	X	X	
11	X	X	X	1	
10	X	X	X	1	

canonica SP:

$$ab'cd' + abcd' + \dots$$

**E9**

normale SP:  $acd' + \dots$

		cd			
	ab	00	01	11	10
00	X	X	X	X	
01	X	X	X	X	
11	X	0	-	X	
10	X	X	X	X	

canonica PS:  $(a'+b'+c+d')(a'+b'+c'+d')$

...

**E9**

normale PS:  $(a'+b'+d')\dots$

# Manipolazione algebrica per via grafica (2)

		cd			
		00	01	11	10
ab	00	X	X	X	X
	01	X	1	1	X
	11	X	1	1	X
	10	X	X	X	X

canonica SP:

$$a'bc'd + a'bcd + abc'd + abcd + \dots$$

↓ E9

↓ E9

normale SP:

$$a'bd + abd + \dots$$

↓ E9

normale SP:

$$bd + \dots$$

Quattro mintermini corrispondenti a configurazioni che individuano un "raggruppamento" di 4 celle a 2 a 2 adiacenti sono equivalenti ad un unico termine con due letterali in meno.



# Manipolazione algebrica per via grafica (3)

La proprietà è vera anche per quattro maxtermini

		cd			
	ab	00	01	11	10
00		X	X	X	X
01		X	X	X	X
11		X	X	0	0
10		X	X	0	0

canonica PS:

$$(a'+b'+c'+d') \cdot (a'+b'+c'+d) \cdot (a'+b+c'+d') \cdot (a'+b+c'+d) \cdot (...)$$


E9

E9

normale PS:  $(a'+b'+c') \cdot (a'+b+c') \cdot (...)$

E9

normale PS:  $(a' + c') \cdot (...)$



**Individuazione dei  
termini primi**

# Raggruppamenti rettangolari

Raggruppamento Rettangolare (RR) di ordine  $p$  - Insieme di  $2^p$  celle di una mappa all'interno del quale ogni cella ha esattamente  $p$  celle adiacenti.

RR ed implicanti - Un RR di ordine  $p$  costituito da celle contenenti valore 1, ed eventualmente condizioni di indifferenza, individua un implicante della funzione. Nel prodotto compaiono le sole  $(n-p)$  variabili che rimangono **costanti** nel RR, in forma vera se valgono 1, in forma complementata se valgono 0.

RR ed implicati - Un RR di ordine  $p$  costituito da celle contenenti valore 0, ed eventualmente condizioni di indifferenza, individua un implicato della funzione. Nella somma compaiono le sole  $(n-p)$  variabili che rimangono **costanti** nel RR, in forma vera se valgono 0, in forma complementata se valgono 1.

# Raggruppamenti, Implicanti e Implicati

RR di dimensione massima ed implicanti primi - Un RR formato da celle contenenti valore "1" o "-" e non interamente incluso in un RR di ordine superiore individua un implicante primo.

RR di dimensione massima ed implicati primi - Un RR formato da celle contenenti valore "0" o "-" e non interamente incluso in un RR di ordine superiore individua un implicato primo (RR).

Esempio (caso SP):

cd \ ab	00	01	11	10
00	X	1	1	X
01	X	1	1	X
11	X	1	1	X
10	0	1	1	X

**bd** non è un implicante primo!

**d** è un implicante primo!

# Esempio (caso PS)

ab \ cd	00	01	11	10
00	0	x	x	0
01	0	x	x	0
11	0	x	x	0
10	0	x	1	0

non è un  
implicato  
primo !

$c + d$

$c' + d$  non è un  
implicato  
primo !

$d$  è un implicato primo !

# Individuazione grafica dei termini ridondanti

Un RR le cui celle sono tutte incluse in altri RR può non essere preso in considerazione per E11.

		cd			
ab		00	01	11	10
00	X	X	X	X	X
01	X	X	X	X	X
11	X	1	1	1	1
10	X	X	X	1	1

normale SP:

$$acd' + abc + abd + \dots$$

$$a(cd' + bc + bd) + \dots \text{ distr. (E3)}$$

↓ **consenso (E11)**

normale SP:  $acd' + abd + \dots$

		cd			
ab		00	01	11	10
00	X	X	X	X	X
01	X	X	X	X	X
11	X	0	0	0	X
10	X	X	0	0	0


normale PS:

$$(a'+b'+d')(a'+c'+d')(a'+b+c') \dots$$

$$a'+(b'+d')(c'+d')(b+c') \dots \text{ distr. (E3)}$$

↓ **consenso (E11)**

normale PS:  $(a'+b'+d')(a'+b+c') \dots$



**Individuazione della  
copertura minima**

# Copertura minima

Copertura di una funzione su una mappa - Insieme di RR la cui unione racchiude tutte le celle contenenti o valore 1 (copertura degli uni) o valore 0 (copertura degli zeri), ed eventualmente celle con valore indifferente.

Coperture ed espressioni normali - Una copertura degli uni (zeri) individua una espressione normale SP (PS) che descrive, nel suo dominio, la funzione assegnata tramite la mappa. Gli implicanti (implicati) che appaiono nell'espressione sono individuati dai raggruppamenti componenti la copertura.

Copertura minima - Copertura costituita dal minimo numero possibile di RR di dimensione massima e corrispondente alla espressione minima.



# Coperture ed espressioni (1)

		cd				
ab		00	01	11	10	
	00	1	1	0	0	$c'$ + $acd'$
	01	1	-	0	-	
	11	1	1	0	1	
	10	1	1	0	1	

Uno dei due RR non è di dimensione massima ( $acd'$  non è un implicante primo): l'espressione non è minima.

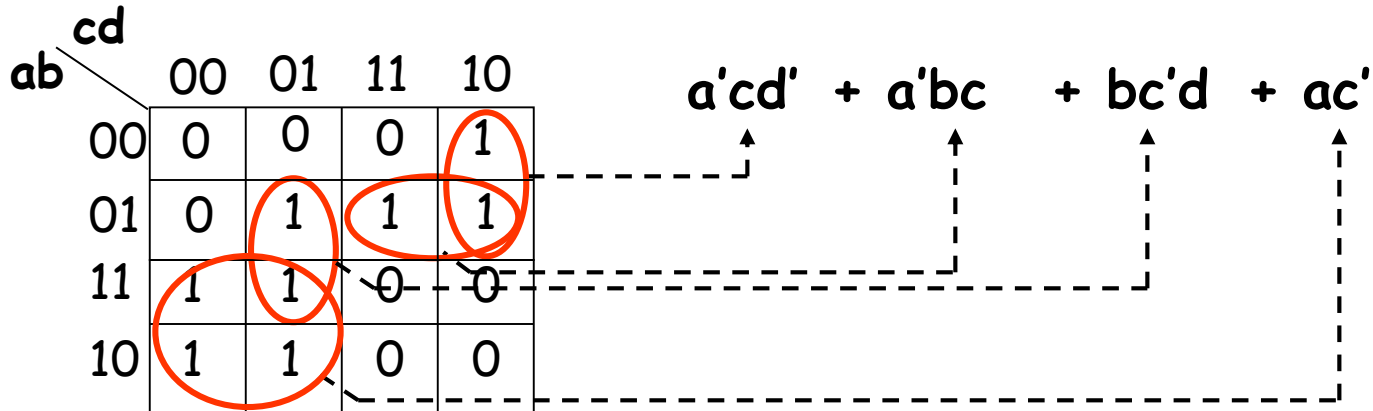


		cd				
ab		00	01	11	10	
	00	1	1	0	0	$c'$ + $ad'$
	01	1	-	0	-	
	11	1	1	0	1	
	10	1	1	0	1	

L'espressione è minima!

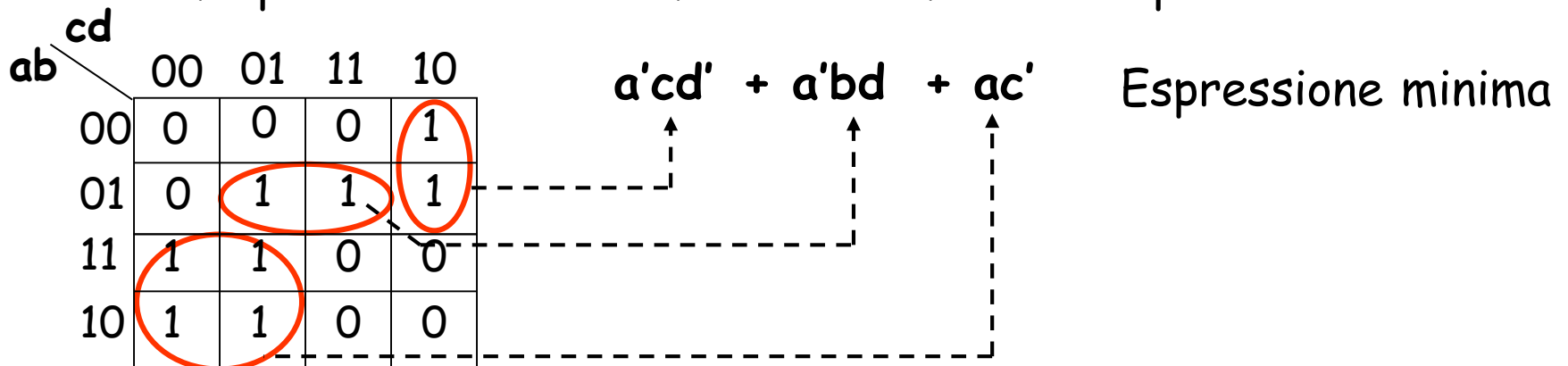
Celle con valore 1 possono essere racchiuse in più raggruppamenti

# Coperture ed espressioni (2)

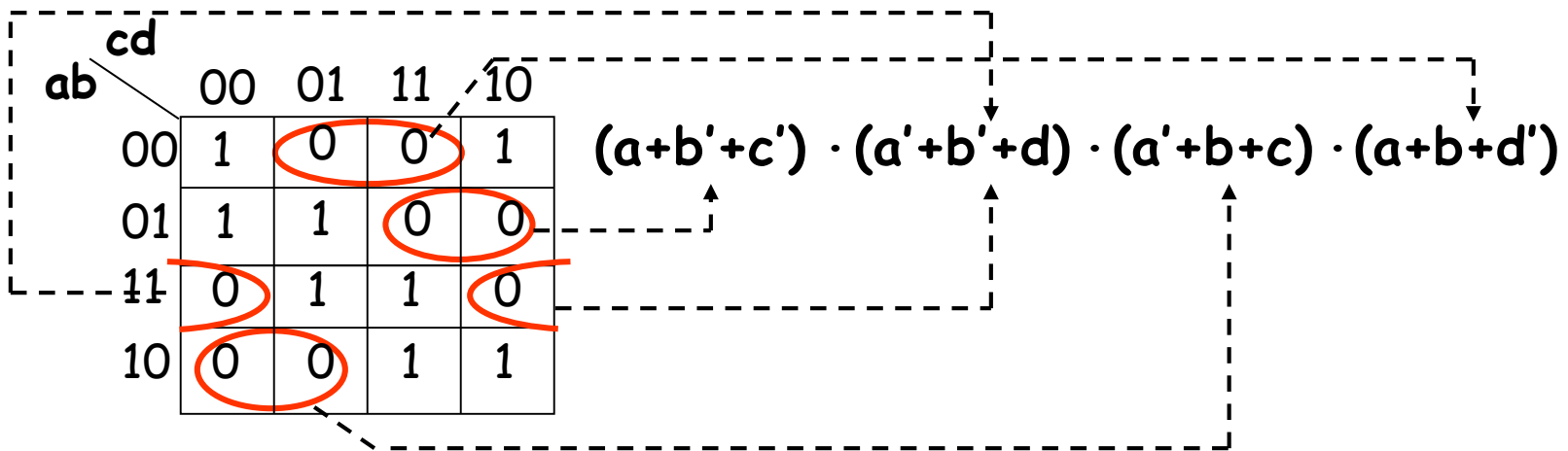
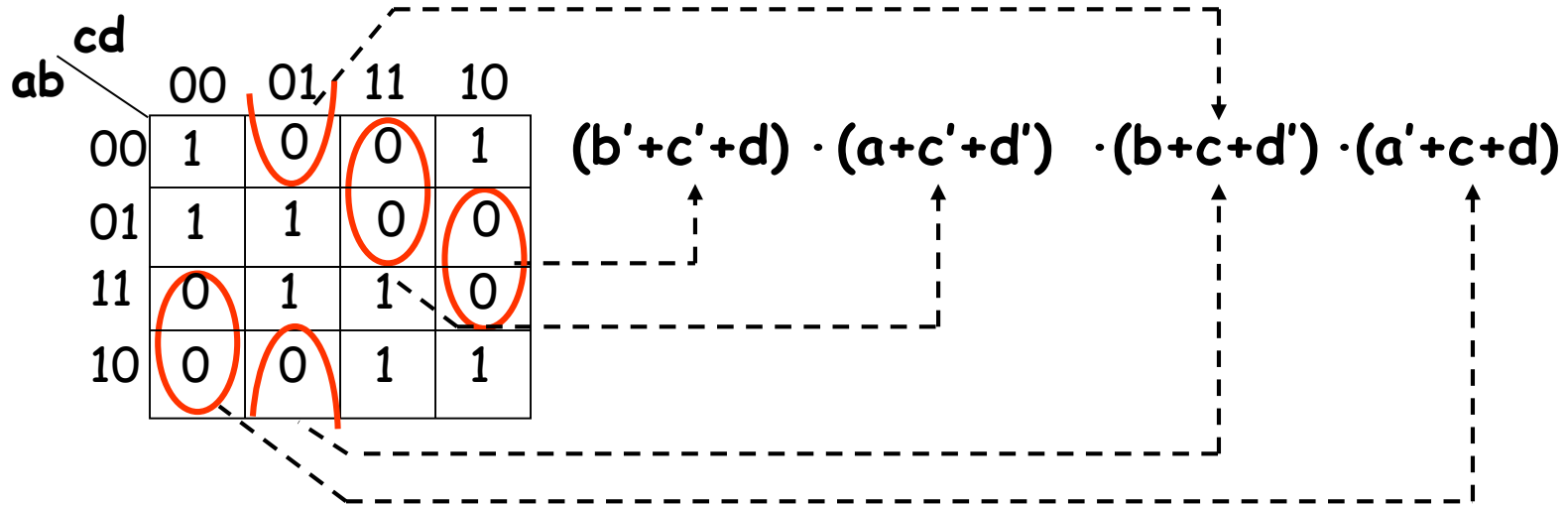


Somma irridondante di implicanti primi, ma non espressione minima  
 (copertura minima: copertura costituita dal **minimo numero possibile** di RR di dimensione massima)

-> non posso ridurre il numero di letterali componenti i singoli termini,  
 ma posso ridurre il numero di termini dell'espressione



# Coperture ed espressioni (3)



Due espressioni minime di tipo PS

# Coperture ed espressioni (4)

		cd			
	ab	00	01	11	10
00	1	1	1	1	
01	1	1	1	1	
11	-	1	-	1	
10	1	1	-	1	

La funzione  $f(a, b, c, d)$  è identicamente uguale a 1

		bc			
	a	00	01	11	10
0	0	1	0	0	
1	1	0	1	0	

$$a' b'c + ab'c' + abc$$

L'espressione minima SP è l'espressione canonica

		cd			
	ab	00	01	11	10
00	0	1	1	0	
01	1	1	1	-	
11	-	1	1	1	
10	0	1	1	0	

$$PS: b + d$$

$$SP: b + d$$

Le coperture minime PS ed SP portano alla stessa espressione

# Individuazione grafica dell'espressione minima (1)

A partire dalla mappa che descrive la funzione occorre determinare la copertura minima e da questa la corrispondente espressione minima. Il procedimento è per sua natura non sistematico e presuppone l'abilità di chi lo esegue.

È tuttavia possibile delineare una sequenza di passi che consentono di individuare con facilità la copertura minima:

1) Si decide se cercare l'espressione di tipo SP o PS e ci si predisponde di conseguenza a coprire gli uni o gli zeri.

ab \ cd	00	01	11	10
00	0	0	0	1
01	0	1	-	-
11	1	1	0	0
10	1	1	0	0

1) scegliamo SP

# Individuazione grafica dell'espressione minima (2)

2) Si cerca di individuare tra le celle da coprire una cella che possa essere racchiusa in un solo RR e lo si traccia di dimensione massima, annotando il termine corrispondente. Se la funzione è incompleta il RR può contenere anche condizioni di indifferenza.

ab \ cd	00	01	11	10
00	0	0	0	1
01	0	1	-	-
11	1	1	0	0
10	1	1	0	0

1) scegliamo  $SP$

2)  $a'cd'$

# Individuazione grafica dell'espressione minima (3)

3) Si ripete fino a quando è possibile il passo 2, tenendo conto della possibilità di coprire anche celle incluse in RR già tracciati.

ab \ cd	00	01	11	10
00	0	0	0	1
01	0	1	-	-
11	1	1	0	0
10	1	1	0	0

1) scegliamo  $SP$

2)  $a'cd'$

3)  $ac'$

# Individuazione grafica dell'espressione minima (4)

4) Si prendono in considerazione le celle ancora da coprire e se ne sceglie a colpo d'occhio la copertura migliore, tenendo conto come al solito della possibilità di coprire celle già coperte e condizioni di indifferenza.

ab \ cd	00	01	11	10
00	0	0	0	1
01	0	1	-	-
11	1	1	0	0
10	1	1	0	0

1) scegliamo SP

2)  $a'cd'$

3)  $ac'$

4)  $a'bd$  oppure  $bc'd$

5) Si ripete il passo 4 fino a soddisfare la condizione di copertura. Si scrive infine l'espressione minima.

$$5) a'cd' + ac' + \begin{cases} a'bd \\ bc'd \end{cases}$$



# Individuazione grafica della espressione minima (5)

		cd			
		00	01	11	10
ab	00	0	0	0	0
	01	0	1	1	0
	11	-	1	-	0
	10	1	1	-	0

1) scegliamo PS

2)  $a+b$

3)  $b'+d$  oppure  $a + d$

4)  $a'+c'$  oppure  $c'+d$

5)  $(a+b) \cdot \left\{ \begin{array}{l} b'+d \\ a+d \end{array} \right\} \cdot \left\{ \begin{array}{l} a'+c' \\ c'+d \end{array} \right\}$

$$(a+b) \cdot (a+d) \cdot (c'+d)$$

# Sintesi minima di un encoder

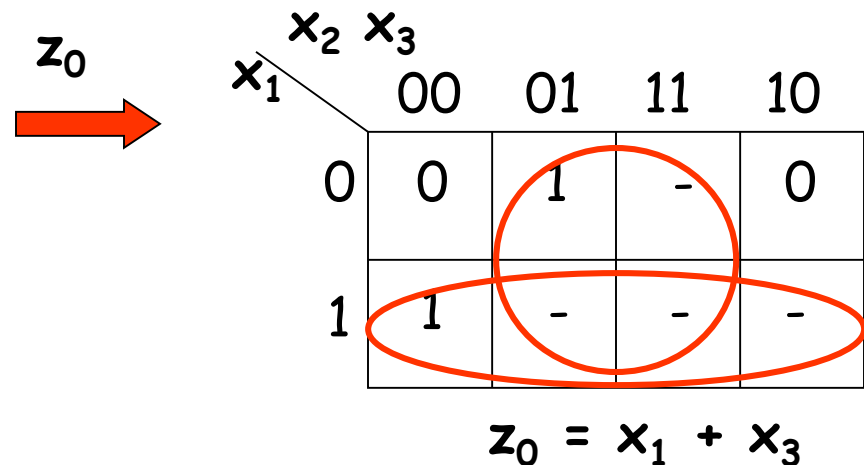
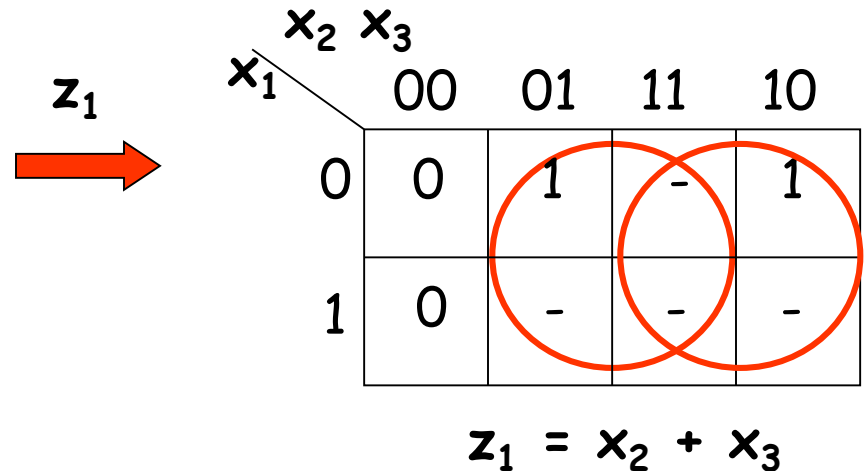
- L'encoder converte il codice «1 su N» in rappresentazione binaria («duale» del decoder)
- Esempio: da codice «1 su 3» a numero binario a due cifre

$x_3$	$x_2$	$x_1$	$z_1$	$z_0$
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 canoniche SP:

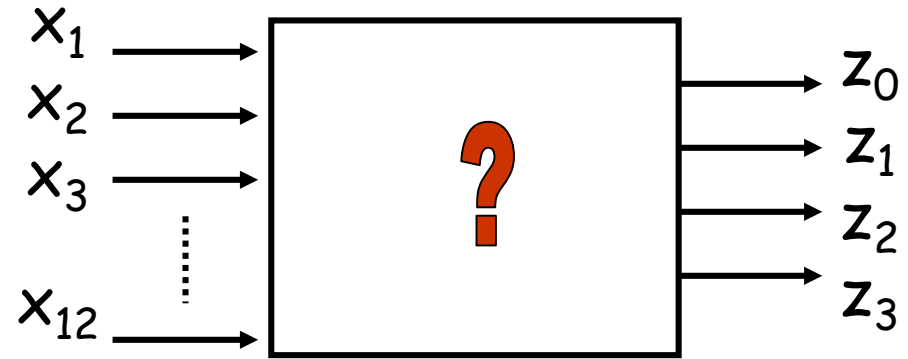
$$z_0 = x_1' x_2' x_3 + x_1 x_2' x_3'$$

$$z_1 = x_1' x_2' x_3 + x_1' x_2 x_3'$$



# Esercizio: l'encoder 12:4

- Per sintetizzare un Encoder a 3 bit d'ingresso è agevole usare le mappe di Karnaugh
- Non è però agevole per sintetizzare un encoder da codice «1 su 12» a numero binario a 4 cifre



- Generalizzando dal risultato precedente:

- la risposta a ogni «1» in ingresso è il numero binario corrispondente al pedice del segnale che lo presenta:

- $x_1(0001) = 1 \rightarrow z_0 = 1, z_1 = 0$ ;  $x_2(0010) = 1 \rightarrow z_0 = 0, z_1 = 1$ ;  $x_3(0011) = 1 \rightarrow z_0 = z_1 = 1$

- L'uscita  $z_i$  sarà dunque generata da un OR degli ingressi il cui pedice (espresso in binario) ha quel bit uguale a 1

- es:  $x_1(0001)$  comparirà solo in  $z_0$ ;  $x_3(0011)$  comparirà in  $z_0$  e  $z_1$

$$Z_0 = x_1(0001) + x_3(0011) + x_5 + x_7 + x_9 + x_{11}$$

$$Z_1 = x_2(0010) + x_3(0011) + x_6 + x_7 + x_{10} + x_{11}$$

$$Z_2 = x_4(0100) + x_5 + x_6 + x_7 + x_{12}$$

$$Z_3 = x_8(1000) + x_9 + x_{10} + x_{11} + x_{12}$$

# Esempio: Full Adder : da espressioni canoniche a espressioni normali

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

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

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

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

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

		ab			
		00	01	11	10
r	0	0	0	1	0
	1	0	1	1	1

$$R = ab + br + ar$$

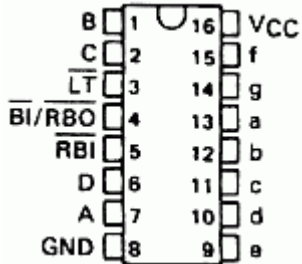
è l'espressione ottenuta in precedenza mediante manipolazione algebrica

		ab			
		00	01	11	10
r	0	0	1	0	1
	1	1	0	1	0

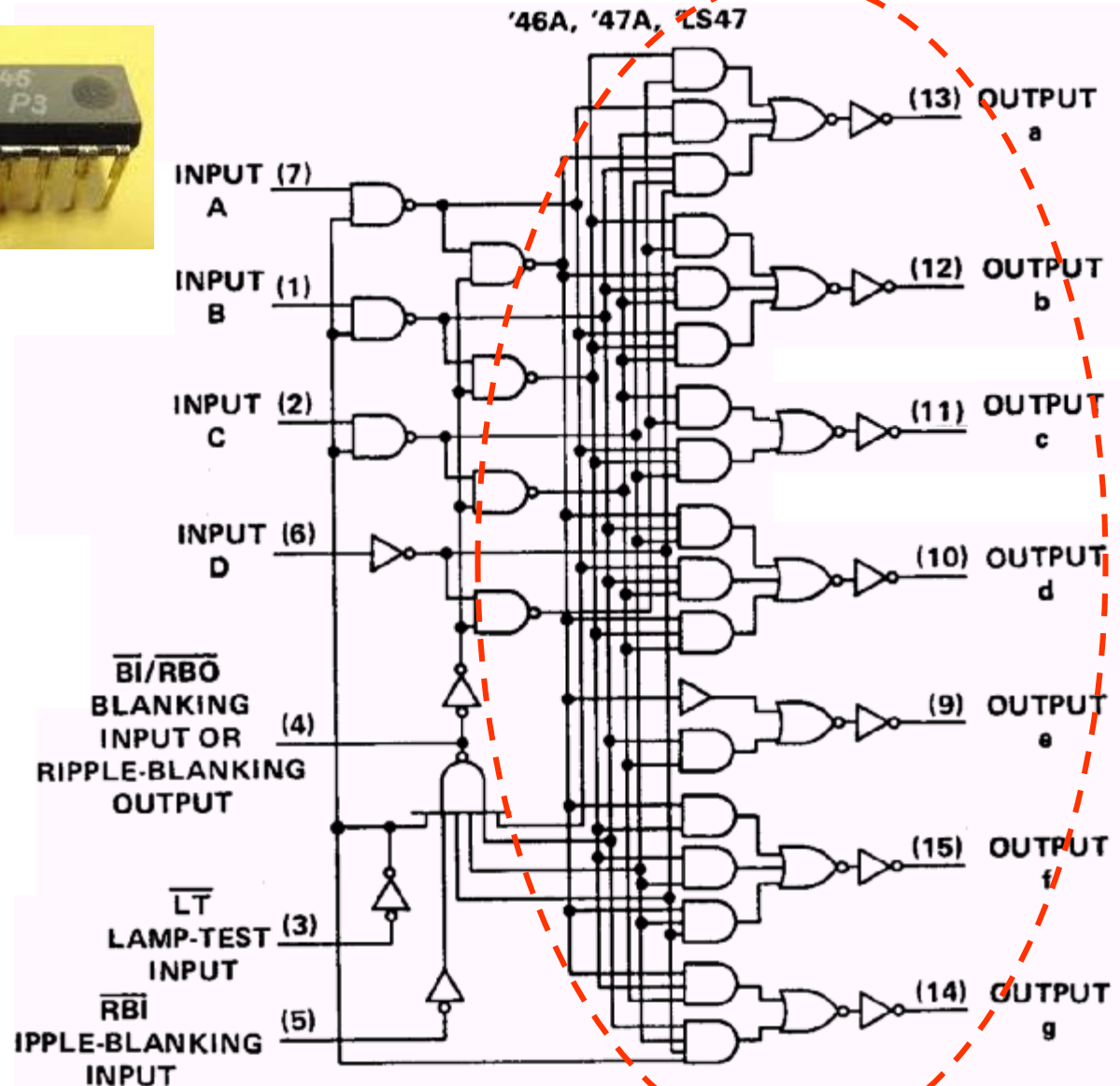
$$S = a'b'r + a'br' + abr + ab'r'$$

L'espressione minima SP per S è l'espressione canonica

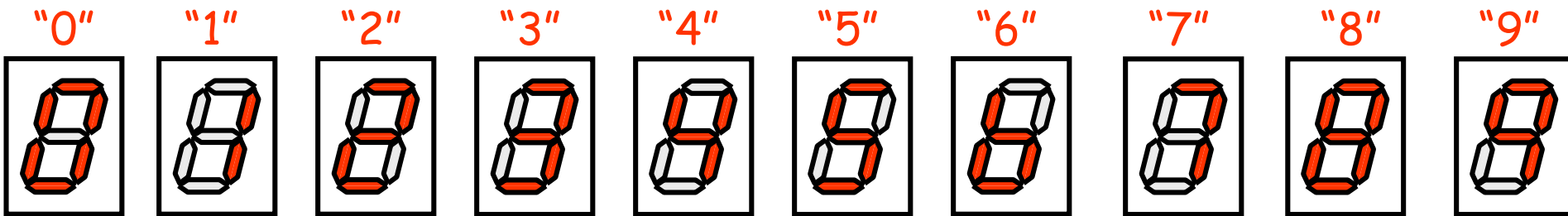
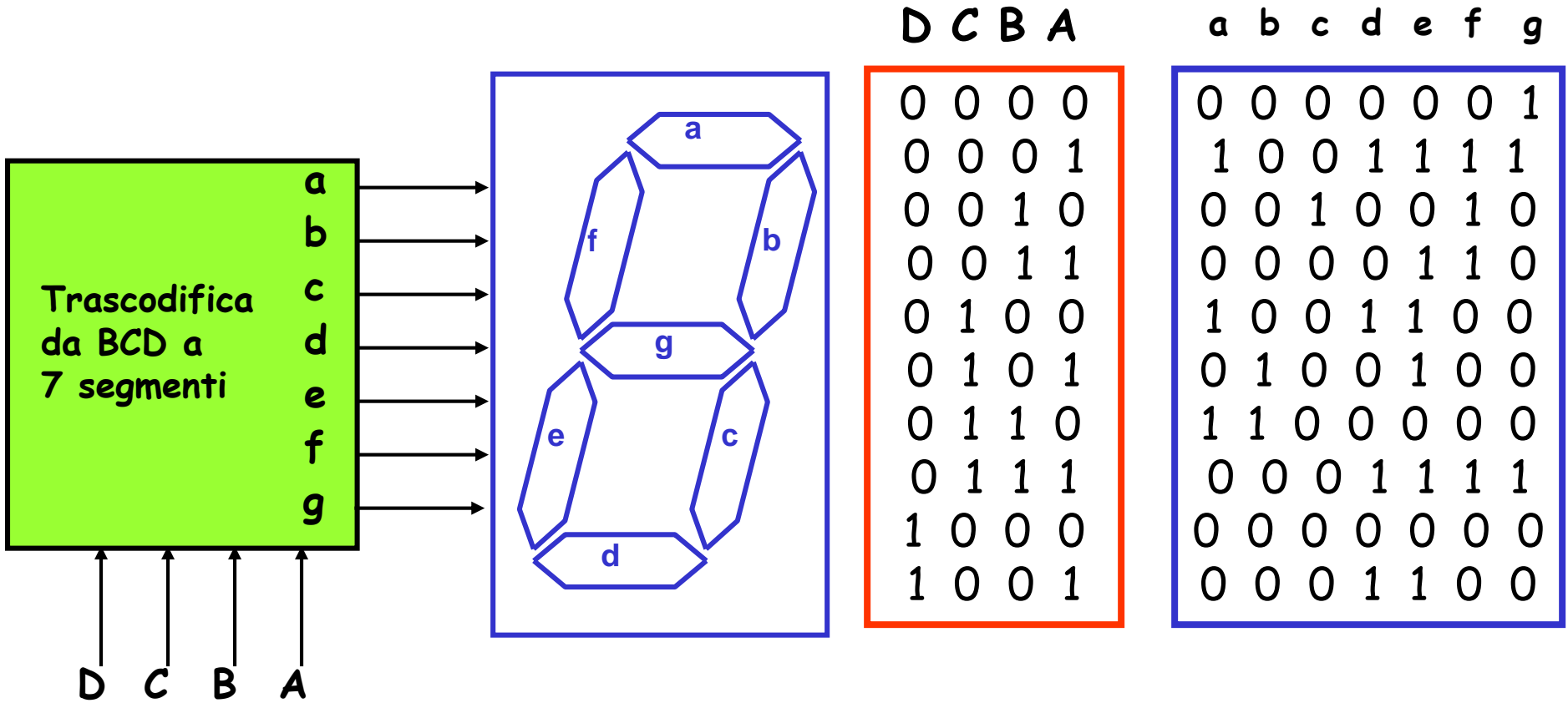
# Il circuito di trascodifica BCD-7 segmenti



SN 7446



# Sintesi



N.B.: 0 visibile, 1 non visibile  
(47 segmenti «accessi» su  $10 \cdot 7 = 70$  totali)

# Progetto della rete di costo minimo (1)

		BA			
		00	01	11	10
DC	00	0	1	0	0
	01	1	0	0	1
	11	-	-	-	-
	10	0	0	-	-

a

$$a = D'C'B'A + CA'$$

		BA			
		00	01	11	10
DC	00	0	0	0	0
	01	0	1	0	1
	11	-	-	-	-
	10	0	0	-	-

b

$$b = CB'A + CBA'$$

		BA			
		00	01	11	10
DC	00	0	0	0	1
	01	0	0	0	0
	11	-	-	-	-
	10	0	0	-	-

c

$$c = C'BA'$$

# Progetto della rete di costo minimo (2)

		BA			
		00	01	11	10
DC	00	0	1	0	0
	01	1	0	1	0
	11	-	-	-	-
	10	0	1	-	-

**d**

		BA			
		00	01	11	10
DC	00	0	1	1	0
	01	1	1	1	0
	11	-	-	-	-
	10	0	1	-	-

**e**

$$d = CB'A' + C'B'A + CBA$$

$$e = A + CB'$$

		BA			
		00	01	11	10
DC	00	0	1	1	1
	01	0	0	1	0
	11	-	-	-	-
	10	0	0	-	-

**f**

		BA			
		00	01	11	10
DC	00	1	1	0	0
	01	0	0	1	0
	11	-	-	-	-
	10	0	0	-	-

**g**

$$f = D'C'A + BA + C'B$$

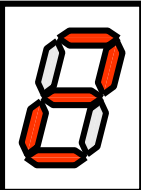
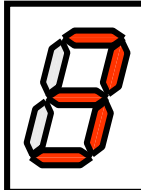
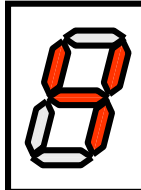
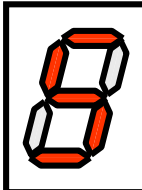
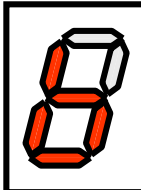
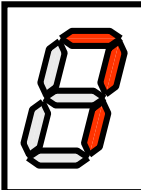
$$g = D'C'B' + CBA$$



# Risposta della rete di costo minimo a configurazioni non previste dal codice BCD

	DCBA					
	1010	1011	1100	1101	1110	1111
a	0	0	1	0	1	0
b	0	0	0	1	1	0
c	1	0	0	0	0	0
d	0	0	1	0	0	1
e	0	1	1	1	0	1
f	1	1	0	0	0	1
g	0	0	0	0	0	1


  

					
-------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------



la rete di costo minimo non consente la rilevazione di alcuna configurazione di ingresso "illecita"

# Progetto della rete in grado di rilevare le configurazioni di ingresso illecite (1)

- Alle configurazioni illecite devono corrispondere sul display simboli diversi da quelli previsti per le configurazioni lecite;
- in particolare il display deve essere spento per la configurazione DCBA = 1111.
- Poichè per tale configurazione la rete risponde accendendo solo i segmenti a,b,c (  ), quest'ultima specifica richiede di ri-sintetizzare solo queste 3 funzioni.

DC \ BA		BA			
		00	01	11	10
DC	00	0	1	0	0
	01	1	0	0	1
	11	-	-	1	-
	10	0	0	-	-

a

$$a = D'C'B'A + CA'$$

$$a_1 = a + DC$$

$$a_2 = a + DB$$

DC \ BA		BA			
		00	01	11	10
DC	00	0	0	0	0
	01	0	1	0	1
	11	-	-	1	-
	10	0	0	-	-

b

$$b = CB'A + CBA'$$

$$b_1 = b + DC$$

$$b_2 = b + DB$$

DC \ BA		BA			
		00	01	11	10
DC	00	0	0	0	1
	01	0	0	0	0
	11	-	-	1	-
	10	0	0	-	-

c

$$c = C'BA'$$

$$c_1 = c + DC$$

$$c_2 = c + DB$$

# Progetto della rete in grado di rilevare le configurazioni di ingresso illecite (2)

In dipendenza delle espressioni selezionate per le funzioni a, b, c, si ottengono così 8 reti ( $R_1, R_2, \dots, R_8$ ), tutte della medesima complessità, caratterizzate dallo stesso comportamento sia per le configurazioni di ingresso previste dal codice BCD, sia per la configurazione DCBA = 1111 (display spento).

$$R_1 : a_1 b_1 c_1 d e f g$$

$$R_5 : a_2 b_1 c_1 d e f g$$

$$R_2 : a_1 b_1 c_2 d e f g$$

$$R_6 : a_2 b_1 c_2 d e f g$$

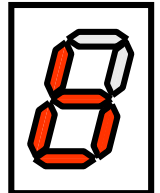
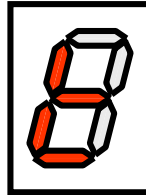
$$R_3 : a_1 b_2 c_1 d e f g$$

$$R_7 : a_2 b_2 c_1 d e f g$$

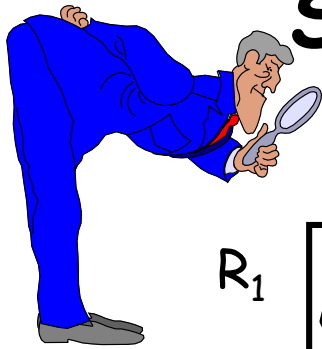
$$R_4 : a_1 b_2 c_2 d e f g$$

$$R_8 : a_2 b_2 c_2 d e f g$$

- Il comportamento è lo stesso anche per la configurazione DCBA = 1110, dal momento che i segmenti a, b, c sono comunque spenti, qualunque siano le espressioni selezionate (tale configurazione è sempre coperta da tutte le reti  $R_1..R_8$ ).
- Il simbolo corrispondentemente visualizzato, che originariamente era:
- ora (spegnendo a,b,c) diventa:



# Scelta della rete "ottimale" (1)



DCBA

	1010		1011		1100		1101		1110		1111	
R <sub>1</sub>												
R <sub>2</sub>												
R <sub>3</sub>												OK
R <sub>4</sub>												
R <sub>5</sub>												OK
R <sub>6</sub>												
R <sub>7</sub>												OK
R <sub>8</sub>												

# Scelta della rete "ottimale" (2)



Le soluzioni  $R_1$ ,  $R_2$ ,  $R_4$ ,  $R_6$  e  $R_8$  vanno scartate, in quanto non consentono la rilevazione di tutte le configurazioni di ingresso illecite.



Le soluzioni  $R_3$ ,  $R_5$  e  $R_7$  vanno bene, in quanto consentono la rilevazione di tutte le configurazioni di ingresso illecite, peraltro con simboli tutti diversi fra loro.



Quale scegliere allora, visto che hanno tutte la medesima complessità e velocità di elaborazione ???



La soluzione  $R_7$ , che richiede un minore consumo di energia (17 segmenti globalmente accesi, anziché 18) per visualizzare le configurazioni illecite. Questa è la soluzione adottata nei circuiti integrati SN 7446A, 7447A !

# Calcolo delle proposizioni

Una certa polizza assicurativa P può essere emessa solo se il richiedente soddisfa almeno una delle seguenti condizioni:

- è un uomo che non ha ancora 25 anni; • U G —
- è sposato ed ha 25 anni o più; • S G' —
- è un uomo sposato che già possiede la polizza Q; • U S Q —
- è una donna sposata che non possiede la polizza Q; • U' S Q' —
- è sposato, non ha ancora 25 anni e possiede già la polizza Q. • S G Q —

SQ

	00	01	11	10
UG				
00				
01				
11				
10				

SQ

	00	01	11	10
UG				
00	0	0	1	1
01	0	0	1	1
11	1	1	1	1
10	0	0	1	1

P

$$P = S + U G$$

La polizza P può essere emessa se il richiedente è sposato oppure se è un uomo che non ha ancora 25 anni.

La selezione dell'equipaggio per una missione spaziale, indicati con A, B, C, D, E i potenziali candidati, deve essere operata in modo tale da soddisfare tutte le seguenti condizioni:

- l'equipaggio deve comprendere o A o B, ma non entrambi;
- l'equipaggio deve comprendere o C o E o entrambi;
- l'equipaggio deve comprendere sia A che C, o nessuno dei due;
- se D fa parte dell'equipaggio, anche B deve farne parte;
- se E fa parte dell'equipaggio, anche C e D devono farne parte.

AB \ DE	00	01	11	10
00	0	0	0	0
01				
11	0	0	0	0
10				

$C=0$

AB \ DE	00	01	11	10
00	0	0	0	0
01				
11	0	0	0	0
10				

$C=1$

- $A B' + A' B$  —
- $C + E$  —
- $A C + A' C'$  —
- $D B + D'$  —
- $E C D + E'$  —

AB \ DE	00	01	11	10
00	0	0	0	0
01	0			0
11	0	0	0	0
10	0			0

$C=0$

AB \ DE	00	01	11	10
00	0	0	0	0
01				
11	0	0	0	0
10				

$C=1$

AB \ DE	00	01	11	10
00	0	0	0	0
01	0			0
11	0	0	0	0
10	0	0	0	0

C=0

AB \ DE	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	0	0	0	0
10	0	0	0	0

C=1

- $A B' + A' B$  —
- $C + E$  —
- $A C + A' C'$  —
- $D B + D'$  —
- $E C D + E'$  —

AB \ DE	00	01	11	10
00	0	0	0	0
01	0			0
11	0	0	0	0
10	0	0	0	0

C=0

AB \ DE	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	0	0	0	0
10	0	0	0	0

C=1

AB \ DE	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	0	0	0	0
10	0	0	0	0

C=0

AB \ DE	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	0	0	0	0
10	1	0	0	0

C=1

**A B' C D' E'**

Un solo equipaggio soddisfa tutte le condizioni: quello costituito dagli astronauti A e C.

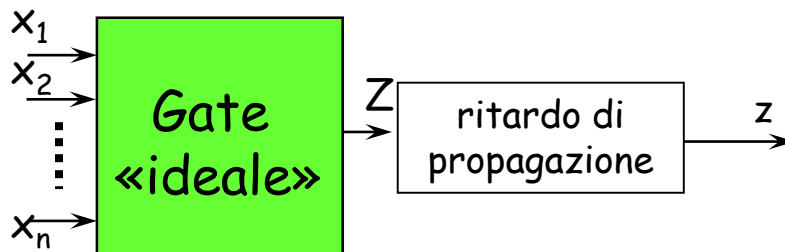




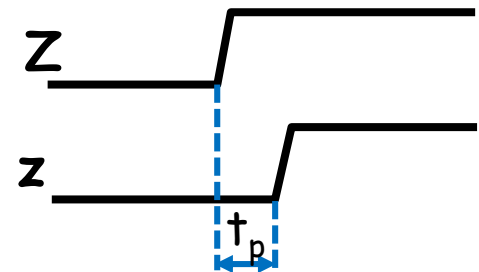
**Alee**

# Il ritardo sui fronti

- Il ritardo sui fronti di salita ( $t_{LH}$ ) e di discesa ( $t_{HL}$ ) 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.
- Un modello più realistico prevede l'utilizzo di un blocco a valle del gate che introduce un ritardo pari al **tempo di propagazione**  $t_p = \max(\tau_{LH}, \tau_{HL})$
- I costruttori di famiglie logiche forniscono i valori *minimo, nominale e massimo* di  $t_p$



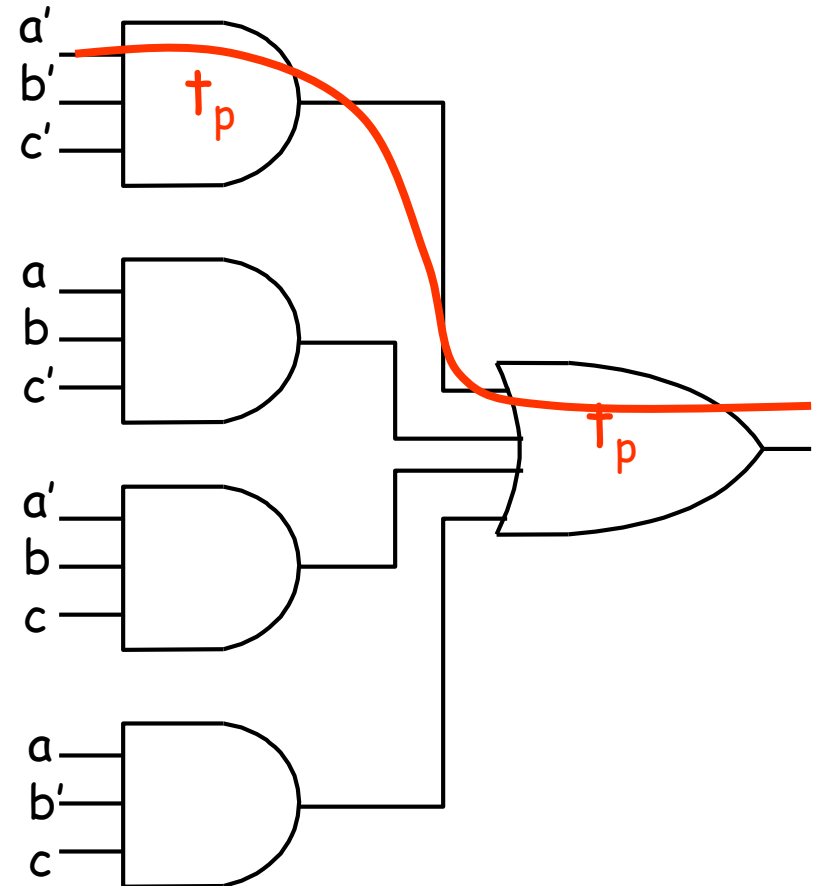
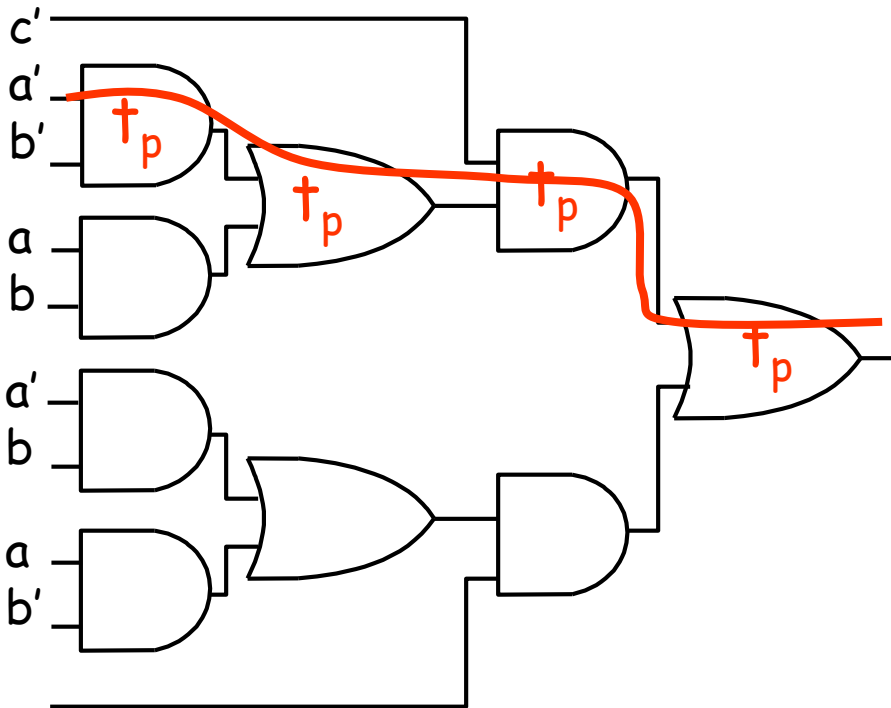
$$Z = F(x_1, x_2, \dots, x_n)$$
$$z(t) = Z(t - t_p)$$



# Velocità e lunghezza dei percorsi

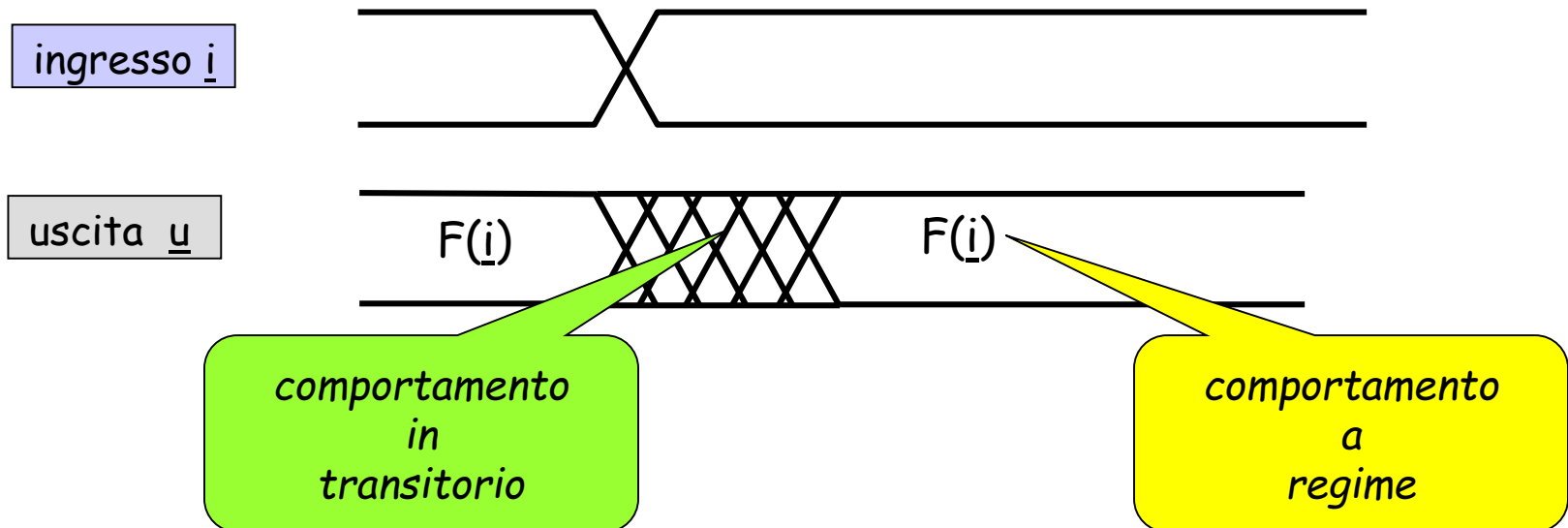
$$(a' \cdot b' + a \cdot b) \cdot c' + (a' \cdot b + a \cdot b') \cdot c = a' \cdot b' \cdot c' + a \cdot b \cdot c' + a' \cdot b \cdot c + a \cdot b' \cdot c$$

- Secondo il modello introdotto in precedenza, la rete di destra è più veloce di quella di sinistra, nonostante realizzino due espressioni tra loro equivalenti



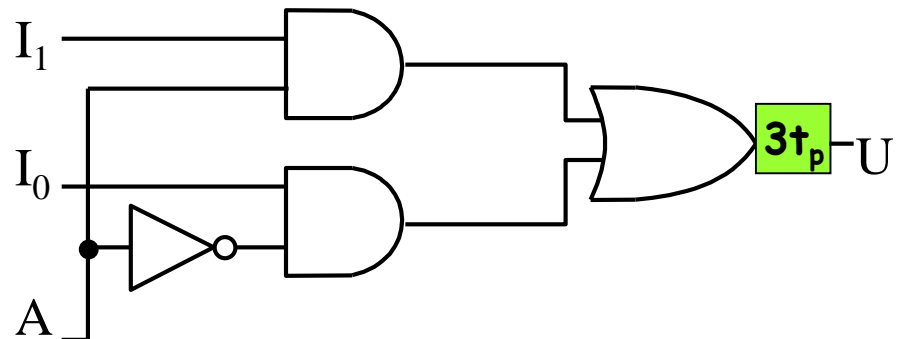
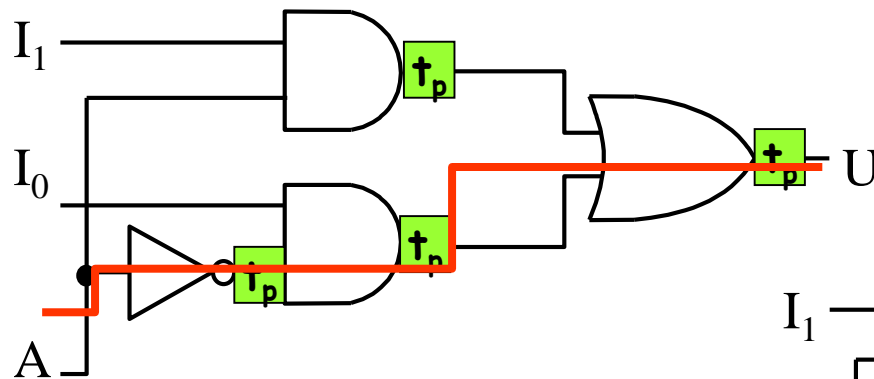
# Comportamento a regime e in transitorio

- Dopo ogni cambiamento dei segnali d'ingresso di una rete combinatoria c'è una fase, detta **comportamento in transitorio**, caratterizzata dal fatto che l'uscita non ha ancora presentato il valore previsto per la nuova configurazione degli ingressi
- Ciò è dovuto al fatto che i nuovi valori d'ingresso devono propagarsi all'interno della struttura prima di riuscire ad imporre al segnale d'uscita il valore che ad essi deve corrispondere
- Al termine del transitorio vi è il **comportamento a regime**



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

- Il transitorio è stimato sul percorso più lento che ogni segnale di ingresso deve attraversare, equivalente a  $n \cdot t_p$ , ove  $n$  è il numero di gate in cascata sul percorso più lungo
- La rete nel suo complesso può essere modellata con un unico ritardo a valle del gate d'uscita
- Es: il **selettore a 2 vie**

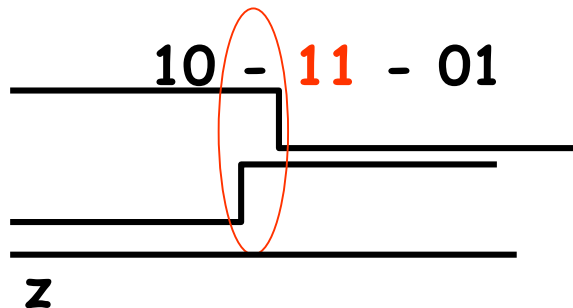
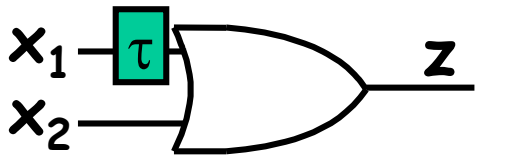
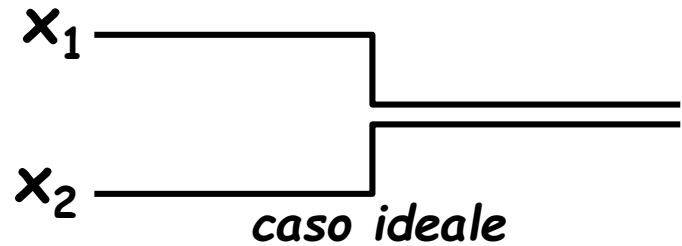
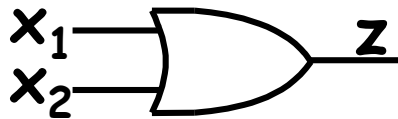


# Alea statica

- **Alea statica:** l'uscita, che dovrebbe rimanere costante, subisce in regime di transitorio una temporanea variazione

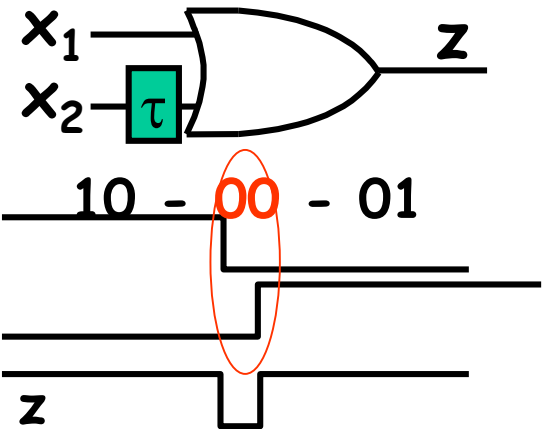


- Esempio: i due ingressi di un gate OR cambiano valore contemporaneamente:



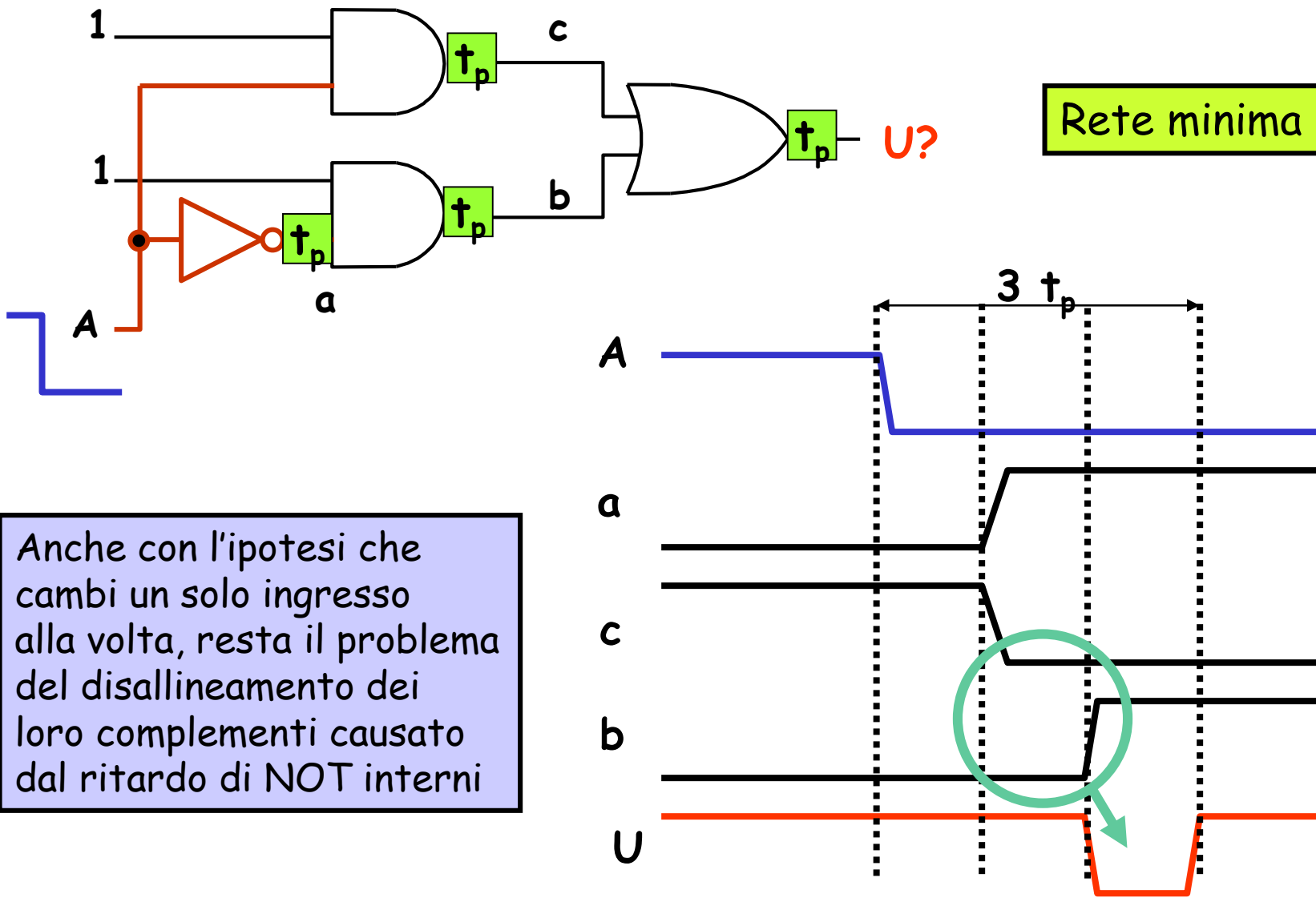
caso reale 1: cambia prima  $x_2$

Occorre far cambiare un solo ingresso alla volta!



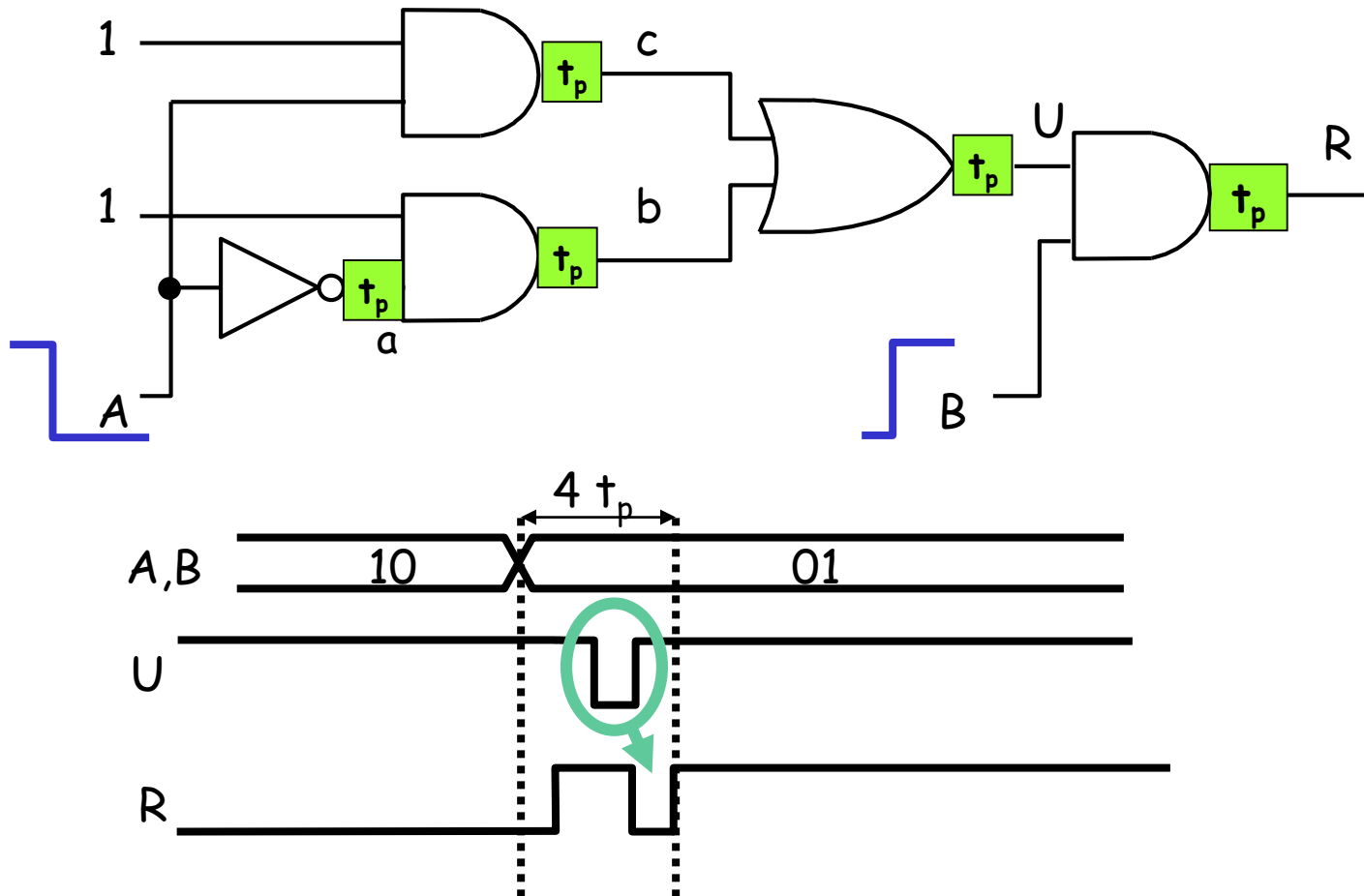
caso reale 2: cambia prima  $x_1$

# Alea statica nel SELETTORE



# Alea dinamica

- **Alea dinamica:** l'uscita, in regime di transitorio, varia più volte prima di assestarsi sul nuovo valore.
- Esempio del selettore con un AND in cascata (PSP) in cui cambiano contemporaneamente i due ingressi A,B:

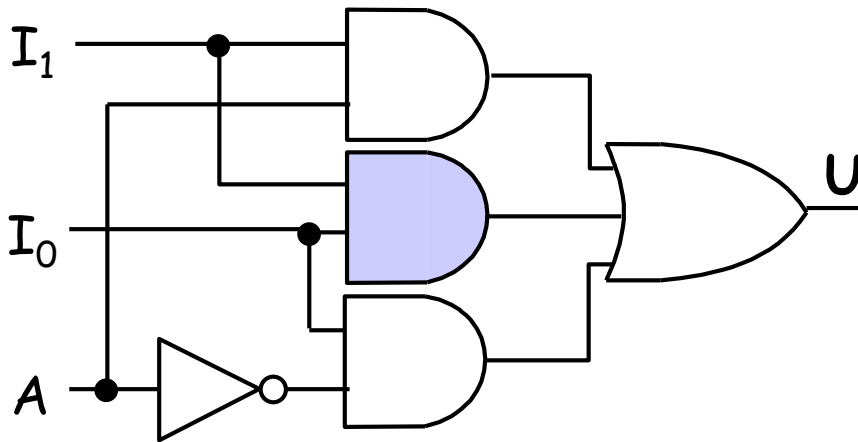




# La copertura "ridondante"

- una rete combinatoria descritta da espressioni SP o PS non presenta mai alee dinamiche

- Per evitare le alee statiche, esistono procedure specifiche in espressioni SP/PS che impiegano una copertura «ridondante» della mappa delle adiacenze
- Esempio del selettore:**



SP ridondante:

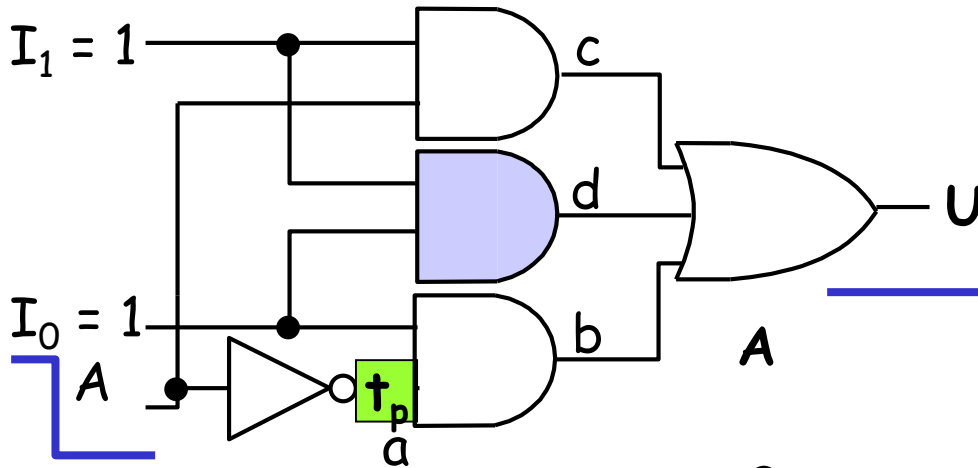
$$U = A'I_0 + I_1 I_0 + AI_1$$

	I <sub>1</sub> I <sub>0</sub>			
A	00	01	11	10
0	0	1	1	0
1	0	0	1	1

SP minima:

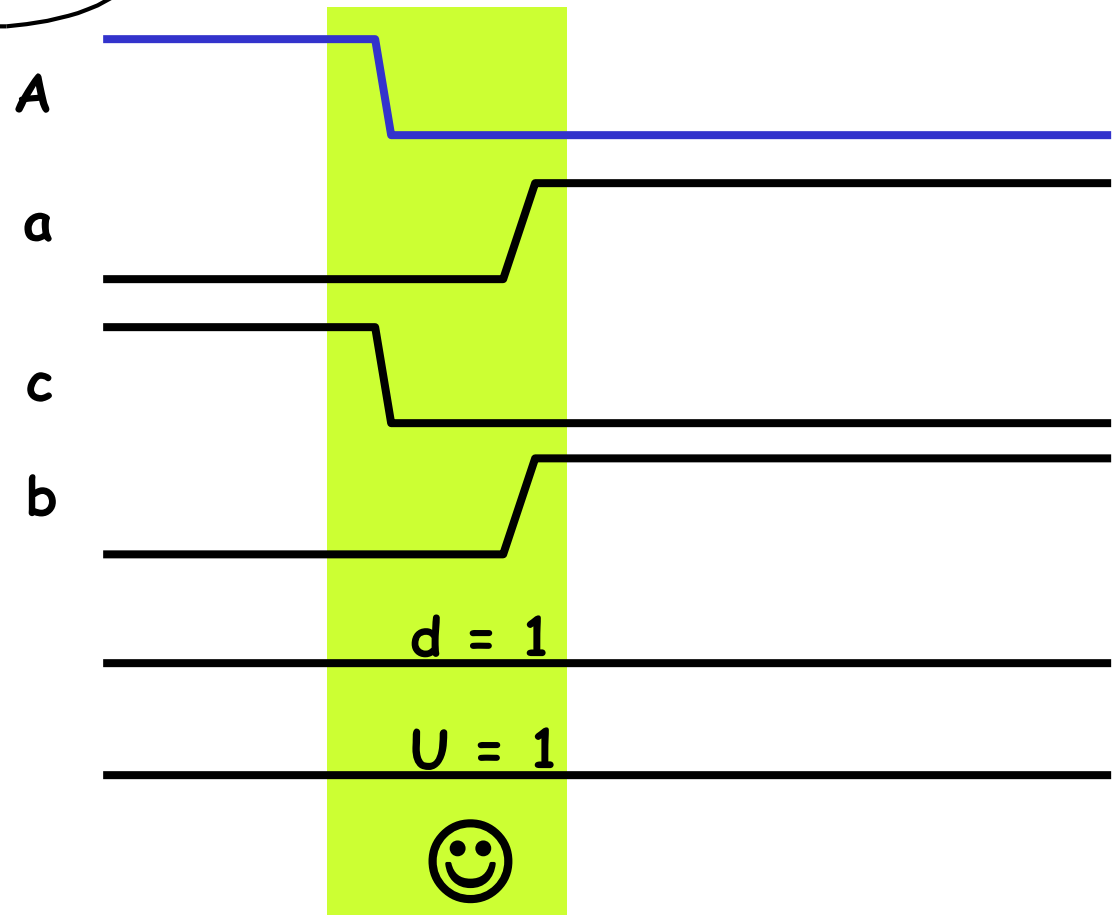
$$U = A'I_0 + AI_1$$

# SELETTORE: rete non minima



SP ridondante:  
 $U = A'I_0 + I_1 I_0 + AI_1$

L'AND in più mantiene il corretto valore di  $U$  per tutto il transitorio.



# Eliminazione a priori delle alee statiche

- Una r. c. completamente specificata ed i cui segnali d'ingresso cambiano di valore uno solo alla volta **non presenta alea statica** se è descritta da un'espressione normale formata da **tutti i termini primi**
- Per evitare la presenza di alee statiche posso dunque utilizzare l'espressione formata da tutti i termini primi.
- In alternativa, posso applicare la procedura della sintesi minima mediante mappe di Karnaugh assicurandomi che valga la seguente condizione:

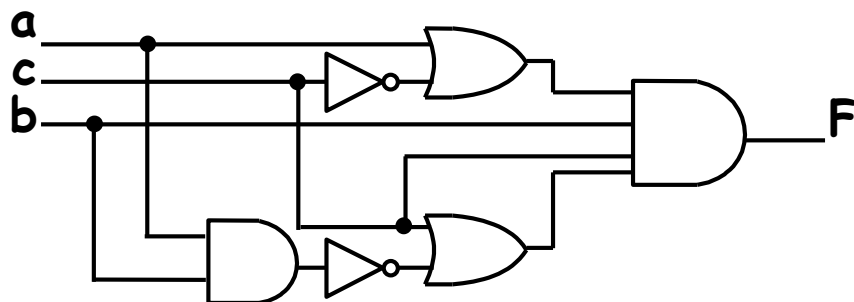
Per eliminare a priori le alee statiche in una rete combinatoria in cui i segnali d'ingresso cambiano di valore uno solo alla volta è necessario e sufficiente scegliere una copertura in cui **ogni coppia di 1 (o di 0) contenuta in celle adiacenti sia racchiusa in almeno un RR.**



**Analisi con  
le mappe**

# Uso delle mappe in sede di analisi (1)

1) Si scrive l'espressione associata allo schema e la si manipola fino ad ottenere una espressione normale:



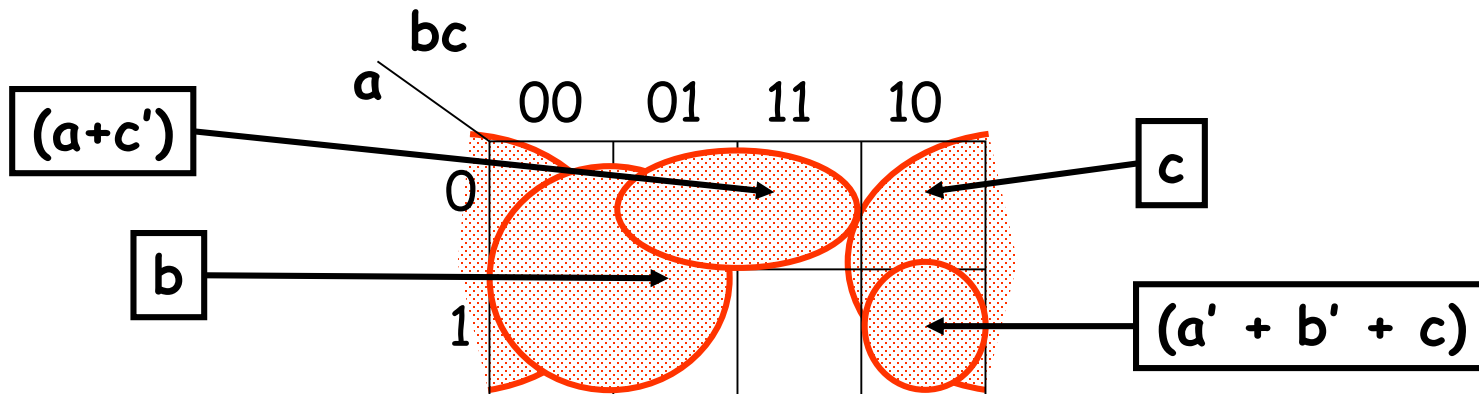
$$F = (a + c') \cdot b \cdot c \cdot ((a \cdot b)' + c)$$

↓ De Morgan (E10)

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

*normale PS*

2) Si predispone una mappa di dimensioni adeguate e si tracciano sulla mappa i RR che corrispondono ai termini :



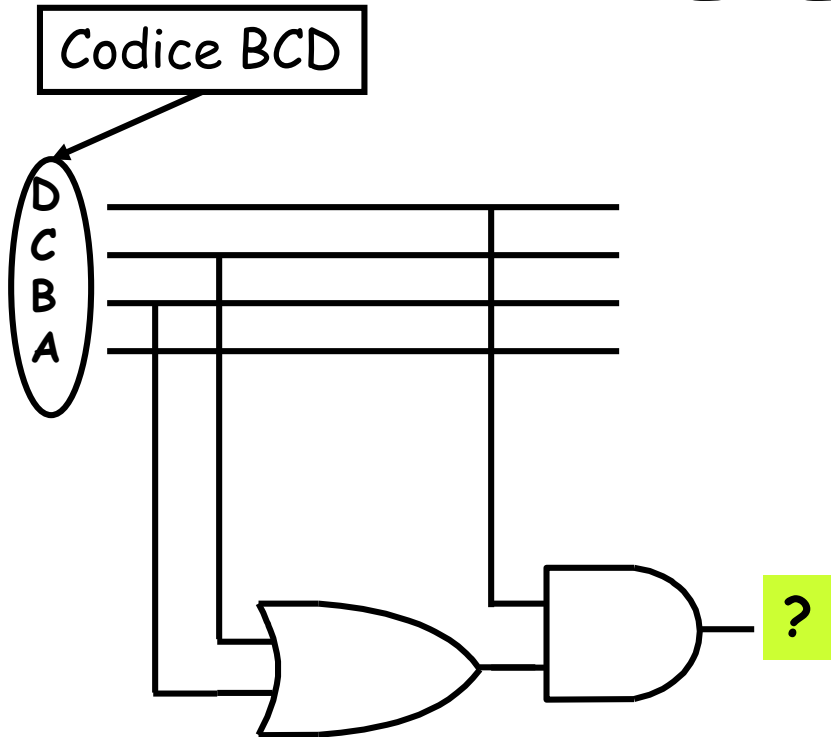
# Uso delle mappe in sede di analisi (2)

3) Nelle celle coperte da un RR si indica il valore 1 se l'espressione normale è SP, 0 se è PS; nelle celle non coperte da RR si inserisce 0 nel caso SP, 1 nel caso PS:

	bc			
a	00	01	11	10
0	0	0	0	0
1	0	0	1	0

N.B. - La valutazione di una espressione individua sempre una funzione completa !

# ESERCIZIO



$$D \cdot (C+B)$$

DC \ BA		BA			
		00	01	11	10
00	00	0	0	0	0
	01	0	0	0	0
11	11	1	1	1	1
	10	0	0	1	1

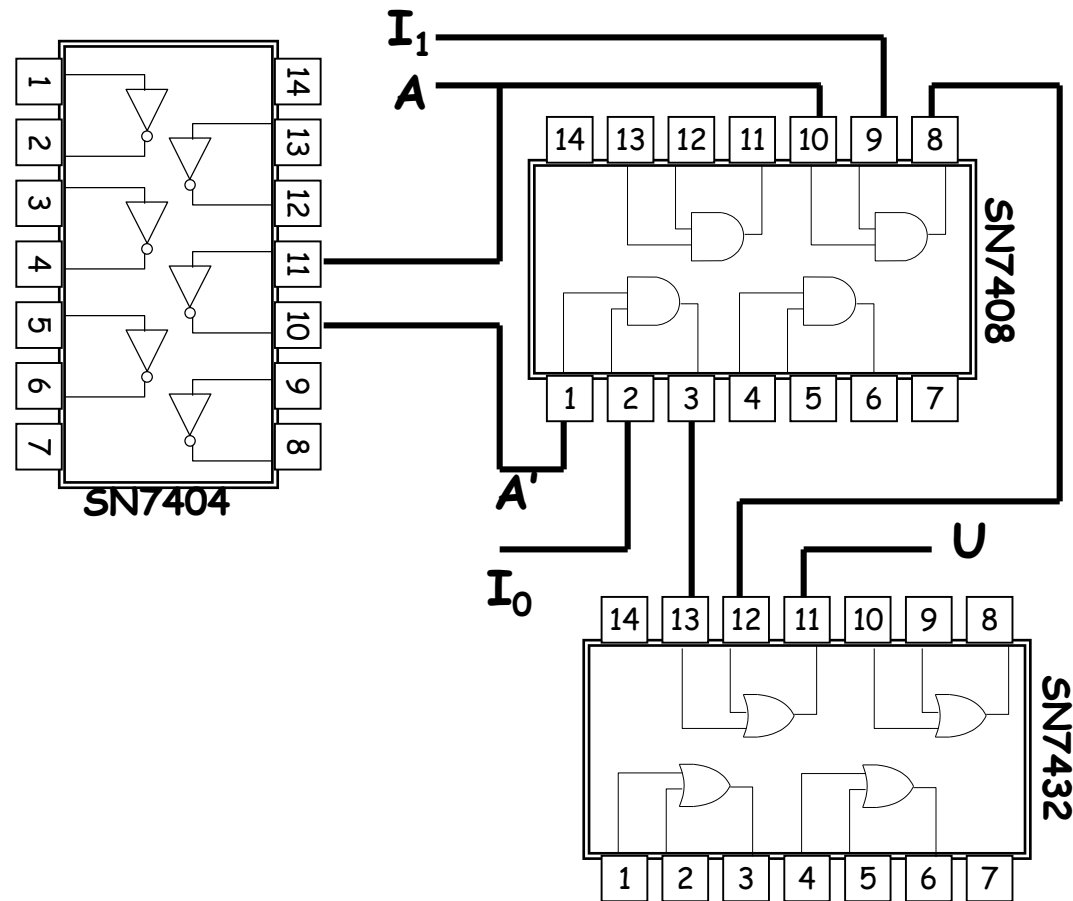
Configurazioni  
«illecite» per  
il codice BCD



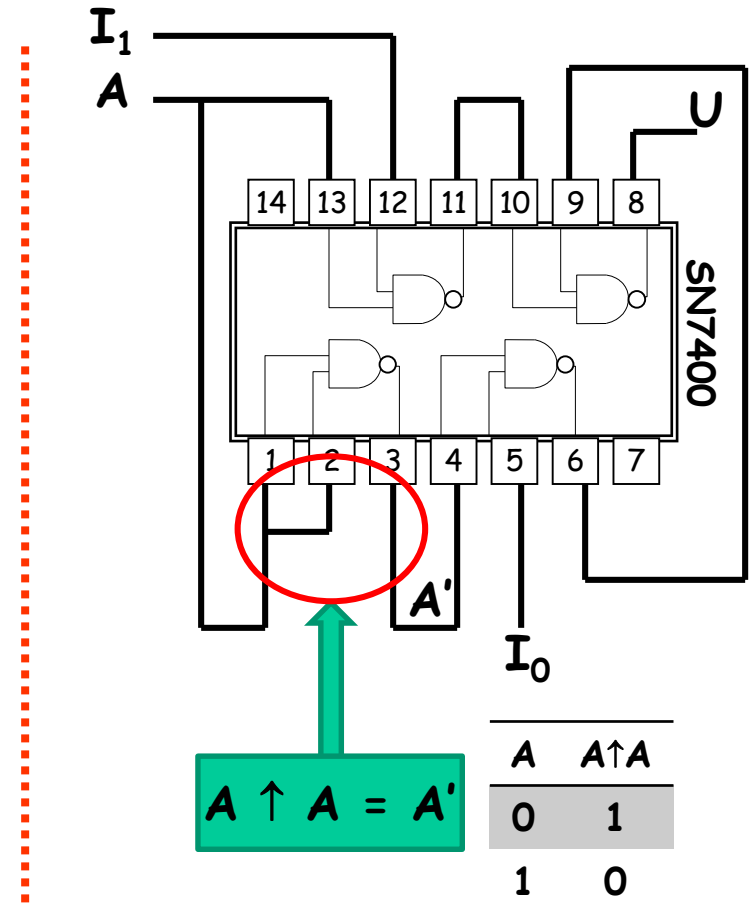
**NAND e NOR**



# Realizzazione con NAND SSI di un selettore a due vie



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



$$A \uparrow A = A'$$

A	$A \uparrow A$
0	1
1	0

$$U = ((A \uparrow A) \uparrow I_0) \uparrow (A \uparrow I_1)$$

Reti a NAND/NOR: utilizzano un unico tipo di gate, riducendo spesso il numero di circuiti SSI necessari

# Sintesi con NAND

La sintesi "a NAND" può essere effettuata trasformando un'espressione normale SP che descrive la funzione assegnata in una nuova espressione contenente esclusivamente operatori "↑":

$$F = a \cdot b + c' \cdot d + e \cdot f' + g$$



definizione dell'operatore ↑

$$F = (a \uparrow b)' + (c' \uparrow d)' + (e \uparrow f')' + g$$



E10 (II<sup>a</sup> legge di De Morgan)

$$F = ((a \uparrow b) \cdot (c' \uparrow d) \cdot (e \uparrow f') \cdot g)'$$



definizione dell'operatore ↑

$$F = (a \uparrow b) \uparrow (c' \uparrow d) \uparrow (e \uparrow f') \uparrow g$$

N.B. : stesso numero di operatori!

# Algoritmo per la sintesi a NAND

1) Si parte da un'espressione SP, SPS, SPSP... e si introducono gli operatori "·" e le parentesi non indicati esplicitamente.

2) Si sostituisce il simbolo "↑" ad ogni simbolo "·"

3) Si sostituisce il simbolo "↑" ad ogni simbolo "+" e si **complementano** le variabili e le costanti affiancate a tale simbolo senza l'interposizione di una parentesi.

4) Si disegna lo schema logico corrispondente all'espressione trovata. Se l'espressione di partenza è a più di due livelli si cerca l'eventuale presenza di NAND con ingressi identici e li si sostituisce con uno solo (sfruttando il fan-out >1 del gate corrispondente).

N.B. - La trasformazione dell'espressione minima SP individua l'espressione minima a NAND.

# Esempio: sintesi a NAND di un EX-OR

$$U = a b' + a'b$$

Espressione canonica SP dell' EX-OR

↓

*passo 1:*

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

↓

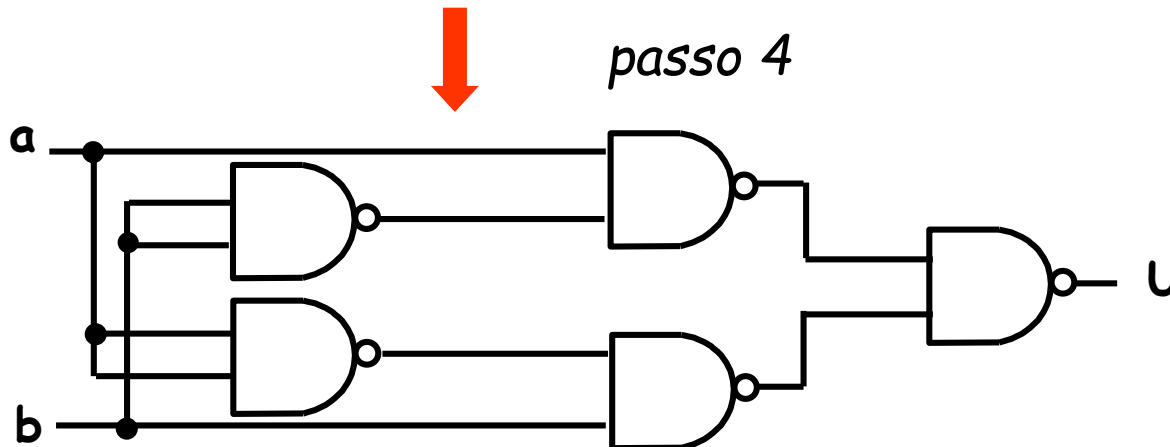
*passi 2 e 3*

$$U = (a \uparrow b') \uparrow (a' \uparrow b)$$

Compaiono delle variabili d'ingresso in forma negata! Se non ne dispongo, le sostituisco aggiungendo ulteriori gate, sfruttando le proprietà del NAND

↓

$$U = (a \uparrow (b \uparrow b)) \uparrow ((a \uparrow a) \uparrow b)$$



5 NAND

# Esempio: sintesi a NAND di un EX-OR

$$U = a b' + a'b$$

Manipolo algebricamente l'espressione di partenza per avere variabili esterne sugli OR in forma negata

$$U = a b' + a'b + a'a + b'b$$

Limitazione e identità

$$U = a (a' + b') + b (a' + b') \quad \text{SPS !}$$



passo 1:

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

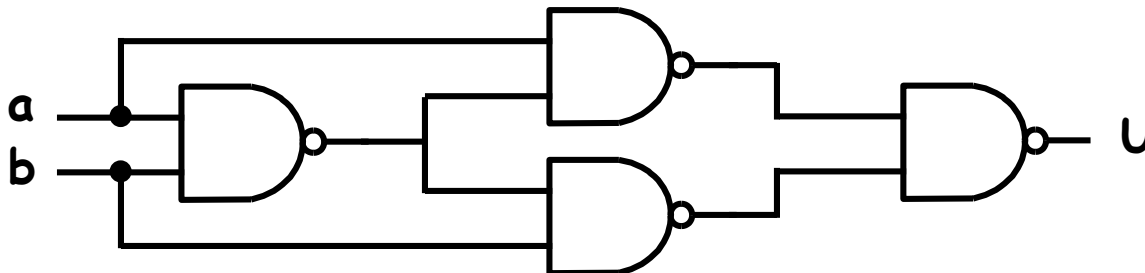


passi 2 e 3

$$U = ( a \uparrow (a \uparrow b) ) \uparrow ( b \uparrow (a \uparrow b) )$$



passo 4



4 NAND

# Esercizio: sintesi a NAND di un Selettore a 2 vie

- Dimostriamo il risultato visto in precedenza relativo alla sintesi a NAND di un selettore a 2 vie

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



passo 1:

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



passi 2 e 3

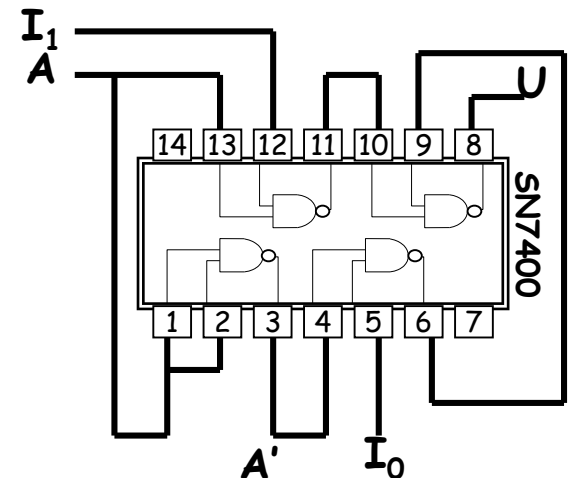
$$U = (A' \uparrow I_0) \uparrow (A \uparrow I_1)$$

Non disponendo del segnale  $A'$ , lo sostituisco mediante un NAND aggiuntivo



$$U = ((A \uparrow A) \uparrow I_0) \uparrow (A \uparrow I_1)$$

4 NAND



# Sintesi con NOR

La sintesi "a NOR" può essere effettuata trasformando un'espressione normale PS che descrive la funzione assegnata in una nuova espressione contenente esclusivamente operatori "↓":

$$F = (a' + b' + c) \cdot (d' + e) \cdot f' \cdot g$$



definizione dell'operatore ↓

$$F = (a' \downarrow b' \downarrow c)' \cdot (d' \downarrow e)' \cdot f' \cdot g$$



E10 (I<sup>a</sup> legge di De Morgan)

$$F = ((a' \downarrow b' \downarrow c) + (d' \downarrow e) + f + g)'$$



definizione dell'operatore ↓

$$F = (a' \downarrow b' \downarrow c) \downarrow (d' \downarrow e) \downarrow f \downarrow g$$

# Algoritmo per la sintesi a NOR

1) Si parte da un'espressione PS, PSP, PSPS... e si introducono gli operatori "·" e le parentesi non indicati esplicitamente.

2) Si sostituisce il simbolo "↓" ad ogni simbolo "+"

3) Si sostituisce il simbolo "↓" ad ogni simbolo "·" e si **complementano** le variabili e le costanti affiancate a tale simbolo senza l'interposizione di una parentesi.

4) Si disegna lo schema logico corrispondente all'espressione trovata. Se l'espressione di partenza è a più di due livelli si cerca l'eventuale presenza di NOR con ingressi identici e li si sostituisce con uno solo (sfruttando il fan-out >1 del gate corrispondente).

N.B. - La trasformazione dell'espressione minima PS individua l'espressione minima a NOR.



# Esempio: sintesi a NOR di un "equivalence"

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

Espressione canonica PS dell'equivalence

↓ passo 1,2,3

$$U = (a \downarrow b') \downarrow (a' \downarrow b)$$

$$A \downarrow A = A'$$

A	A ↓ A
0	1
1	0

Sfrutto tale proprietà per ottenere un'espressione in cui le variabili compaiono solo in forma vera

$$U = (a \downarrow (b \downarrow b)) \downarrow ((a \downarrow a) \downarrow b)$$

**5 NOR**

Manipolo algebricamente l'espressione di partenza per avere sugli AND variabili esterne in forma negata

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

Limitazione e identità

$$U = (a + a'b') \cdot (b + a'b') \quad PSP!$$

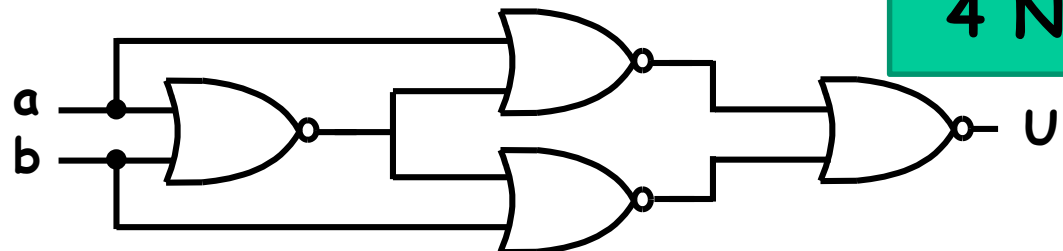
↓ passo 1

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

↓ passi 2 e 3

$$U = (a \downarrow (a \downarrow b)) \downarrow (b \downarrow (a \downarrow b))$$

↓ passo 4



**4 NOR**

# Determinazione dell'espressione minima con NAND

**NAND:**

- 1 - Copertura degli "uni" con il minimo numero di RR "massimi"
- 2 - Espressione minima **SP**
- 3 - Sostituzione di + e · con  $\uparrow$
- 4 - Complementazione delle variabili direttamente operate dal «+»

		M A			
i		00	01	11	10
	0		0	0	-
1		1	0	-	1

$$I = M + A' \cdot i$$

$$\begin{aligned} I &= M + (A' \cdot i) \\ &= M \uparrow (A' \uparrow i) \\ &= M' \uparrow (A' \uparrow i) \end{aligned}$$

# Determinazione dell'espressione minima con NOR

**NOR:**

- 1 - Copertura degli "zeri" con il minimo numero di RR "massimi"
- 2 - Espressione minima PS
- 3 - Sostituzione di + e · con ↓
- 4 - Complementazione delle variabili direttamente operate dal «·»

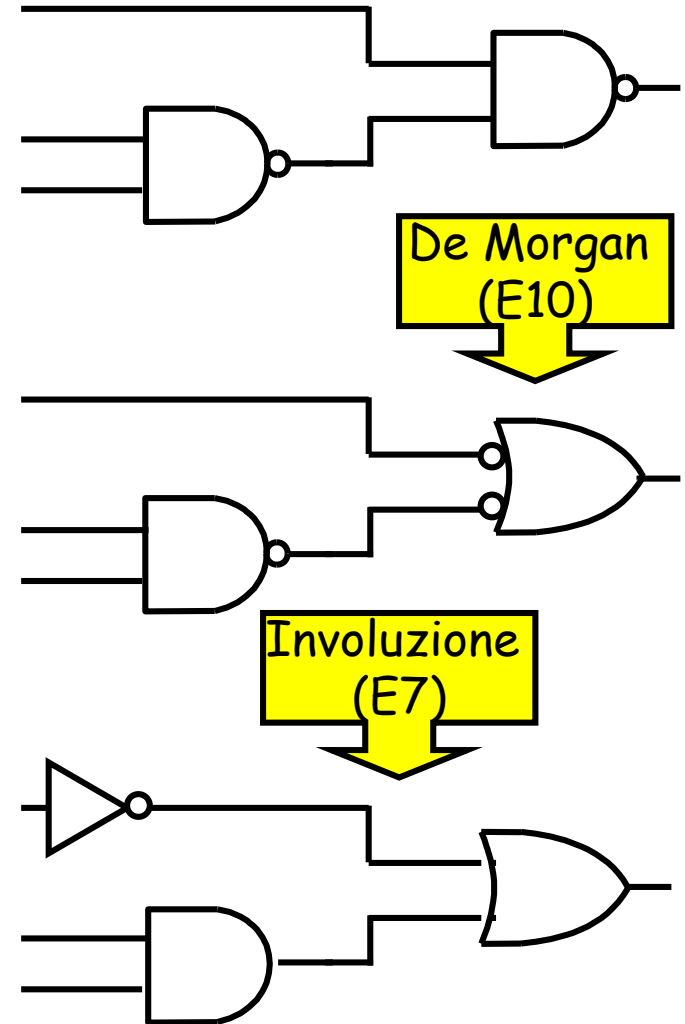
	M		A	
i	00	01	11	10
0	0	0	-	1
1	1	0	-	1

$$I = A' \cdot (M + i)$$

$$\begin{aligned} I &= A' \cdot (M + i) \\ &= A' \downarrow (M \downarrow i) \\ &= A \downarrow (M \downarrow i) \end{aligned}$$

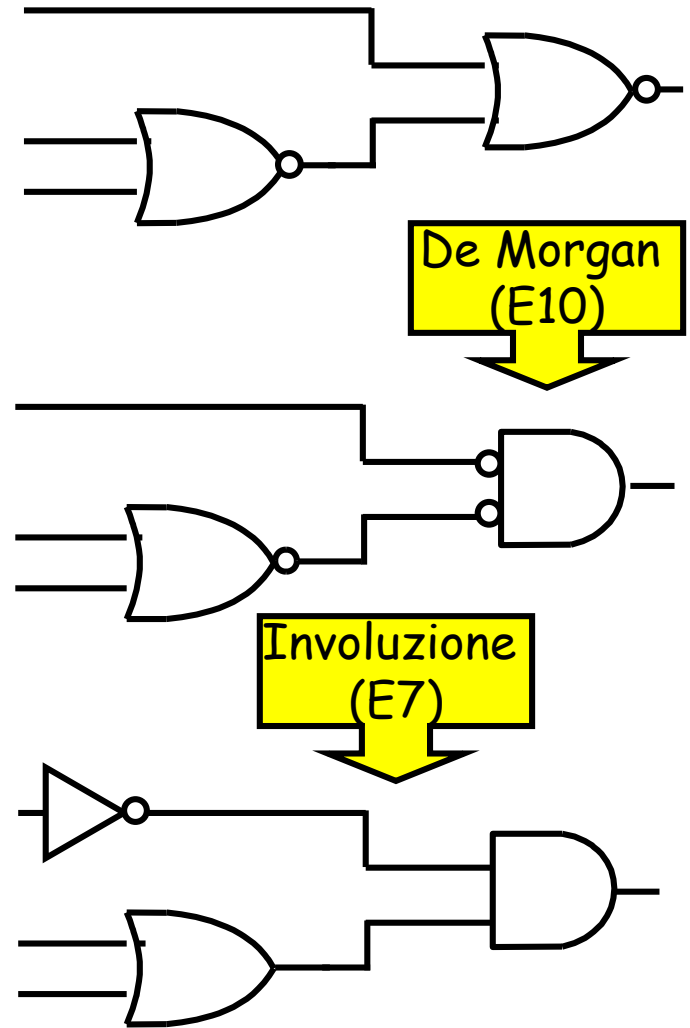
# ANALISI per via grafica (reti a NAND)

- Da rete a NAND (2 livelli) a rete a AND/OR/NOT (2 livelli):
  1. Applicare il teorema di De Morgan, sostituendo il NAND d'uscita con un OR
  2. Eliminare le coppie di negazione in serie (proprietà di involuzione), trasformando così i rimanenti gate NAND in AND
- Per reti a più di due livelli: iterare questo procedimento per gli eventuali stadi a monte



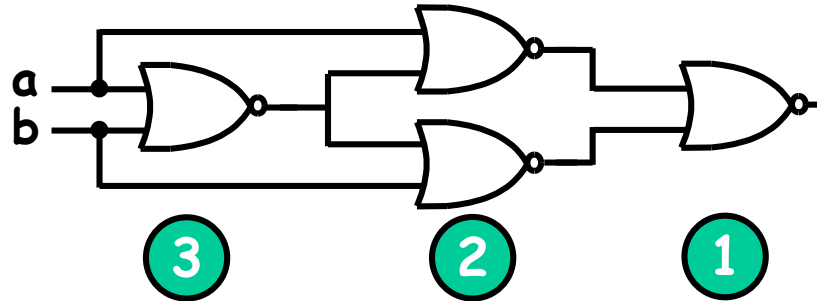
# ANALISI per via grafica (reti a NOR)

- Da rete a NOR (2 livelli) a rete a AND/OR/NOT (2 livelli):
  1. Applicare il teorema di De Morgan, sostituendo il NOR d'uscita con un AND
  2. Eliminare le coppie di negazione in serie (proprietà di involuzione), trasformando così i rimanenti gate NOR in OR
- Per reti a più di due livelli: iterare questo procedimento per gli eventuali stadi a monte



# Regole per analisi di reti a NAND&NOR

**Livello di un gate:** n° di gate interposti tra i suoi ingressi e l'uscita (procedendo lungo l'espressione da sx. verso dx., il livello cresce dopo una parentesi aperta, decresce dopo una parentesi chiusa)



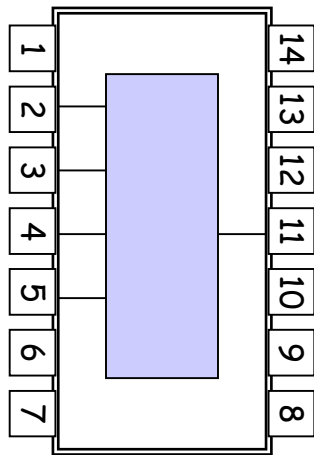
$$\begin{array}{c}
 ( a \downarrow (a \downarrow b) ) \downarrow ( b \downarrow (a \downarrow b) ) \\
 \begin{array}{cccc}
 & 2 & 3 & & 1 & & 2 & 3 \\
 & \downarrow & \downarrow & & \downarrow & & \downarrow & \downarrow \\
 ( a + (a' \cdot b') ) \cdot ( b + (a' \cdot b') )
 \end{array}
 \end{array}$$

	Se livello pari sostituire con:	Se livello dispari sostituire con:
NAND	AND	OR con ingressi <i>esterni negati</i>
NOR	OR	AND con ingressi <i>esterni negati</i>

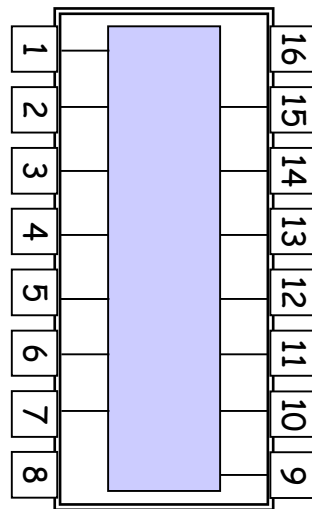
# Circuiti combinatori MSI e LSI

- Sono disponibili nella forma di componenti elementari anche reti più complesse delle semplici «famiglie di gate» viste in precedenza
- Tali reti risultano particolarmente utili per il progettista logico, in quanto molto utilizzate per la realizzazione di componenti via via più complessi
- vedremo ora l'utilizzo di alcune di tali reti al fine della sintesi combinatoria

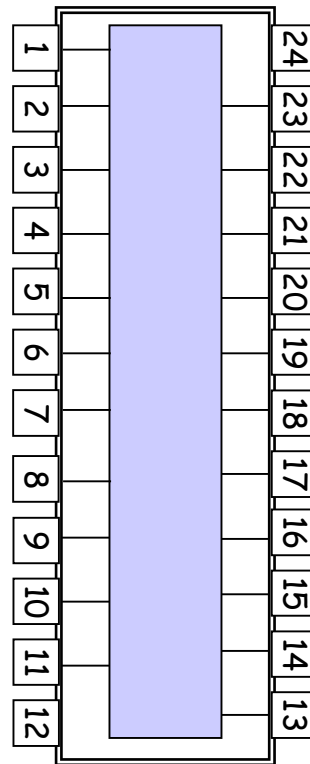
*Si consiglia di visitare il sito di un Costruttore (ad es. [www.ti.com](http://www.ti.com))!*



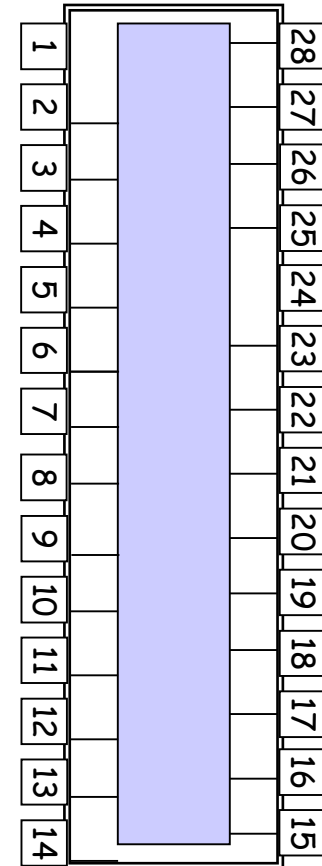
**Full adder**



**Decoder Multiplexer**



**Aritmetica Registro acc.**



**Trascodifica Buffer Contatore**

...  $\mu$ P  
RAM