

Operatori logici e porte logiche



Operatori unari	1665
Connettivo AND	1667
Connettivo OR	1667
Connettivo XOR	1668
Ordine di precedenza	1669
Valori irrilevanti nelle tabelle di verità	1669
Equivalenze	1670
Somma dei prodotti	1674
Mappe di Karnaugh	1676
Mappe di Karnaugh con la funzione XOR	1684

La logica formale studia le proposizioni dichiarative, dove per proposizione si intende l'insieme di soggetto e verbo. È una **proposizione dichiarativa** quella proposizione nei confronti della quale è possibile stabilire se è vera o se è falsa. *Vero* o *Falso* sono gli unici valori che può assumere una proposizione dichiarativa. La proposizione che non si può suddividere in altre proposizioni, si dice essere elementare.

Il valore di una proposizione dichiarativa (*Vero* o *Falso*) può essere espresso in vari modi, a seconda del contesto. Generalmente, si attribuisce alla cifra numerica uno il significato di *Vero*, mentre a zero

si attribuisce il valore *Falso*, così come nell'applicazione elettronica si utilizza un livello di tensione vicino a quello di alimentazione per rappresentare *Vero* e un valore di tensione vicino a zero per rappresentare *Falso*.

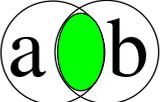
<i>Falso</i>	0	0 V	
<i>Vero</i>	1	tensione di alimentazione	

La variabile che può assumere solo il valore risultante da una proposizione dichiarativa, è una **variabile logica**. Un'**espressione logica** o una **funzione logica** è quella che produce un risultato *Vero* o *Falso*. L'espressione o funzione logica può essere costituita da proposizioni dichiarative, da valori costanti (espressi secondo la forma prevista per rappresentare *Vero* o *Falso*) e da variabili logiche. Per connettere o comunque per intervenire nei valori delle varie componenti dell'espressione, si utilizzano degli operatori che rappresentano delle funzioni elementari.

Si distinguono generalmente gli operatori logici in «unari» e in «connettivi logici», per distinguere se intervengono in un solo valore logico, oppure su due o più valori logici. Gli operatori logici si possono vedere come delle scatoline, aventi uno o più ingressi, ma con una sola uscita: in tal caso, si chiamano **porte logiche**.

Gli operatori logici hanno forme differenti di rappresentazione, in base al contesto e in base alle limitazioni tipografiche a cui si deve sottostare. Pertanto, ogni volta che si legge un documento che tratta questo genere di argomenti, è necessario inizialmente comprendere qual è la simbologia adottata, tenendo conto che anche all'interno dello stesso testo si possono alternare simbologie differenti.

Tabella u97.2. Alcune notazioni alternative degli operatori logici comuni, associate alla simbologia delle porte logiche corrispondenti.

	a	a	a	a  a
	NOT a	a'	\bar{a}	a  \bar{a}
	a AND b	$a \cdot b$	$a \wedge b$	a b  $a \wedge b$
	a OR b	$a + b$	$a \vee b$	a b  $a \vee b$
	a XOR b		$a \oplus b$	a b  $a \oplus b$

Per definire il significato degli operatori logici si utilizzano delle tabelle di verità, mettendo a confronto le variabili in ingresso con l'esito finale.

Operatori unari

Gli operatori logici unari ottengono in ingresso un solo valore logico; sono disponibili l'invertitore logico (NOT) e il non-invertitore logico. «

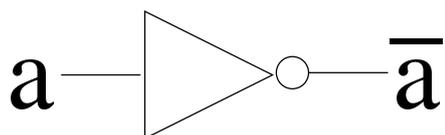
L'invertitore logico è l'operatore unario che inverte il valore logico ricevuto in ingresso: se in ingresso riceve il valore *Vero* (1), in uscita genera il valore *Falso* (0); se in ingresso riceve il valore *Falso* (0), in uscita genera il valore *Vero* (1).

A titolo di esempio, se la variabile logica «A» contiene il risultato della proposizione dichiarativa «Antonio mangia», l'espressione lo-

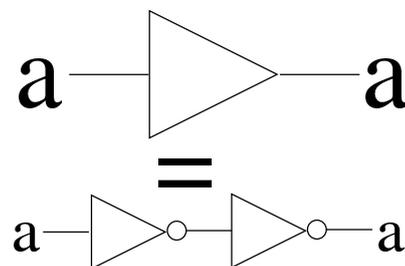
gica «NOT A» è equivalente alla proposizione dichiarativa «Antonio non mangia».

Il non-invertitore logico è l'operatore unario che presenta in uscita lo stesso valore ricevuto in ingresso. Il nome che viene dato a questo tipo di operatore allude alla presenza ipotetica di due negazioni consecutive che si eliminano a vicenda. Per esempio, se la variabile logica «A» contiene il risultato della proposizione dichiarativa «Antonio mangia», l'espressione logica «NOT A» è equivalente alla proposizione dichiarativa «Antonio non mangia», ma nello stesso modo, «NOT (NOT A)» è equivalente alla proposizione originale: «Antonio mangia».

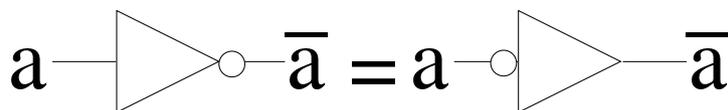
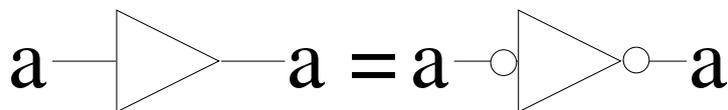
a	\bar{a}
0	1
1	0



a	a
0	0
1	1



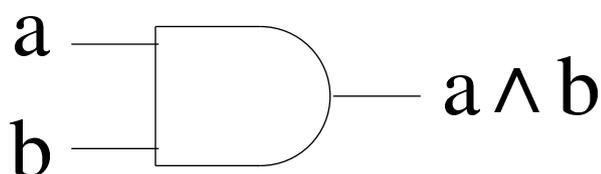
Dagli esempi mostrati si dovrebbe intendere il fatto che un piccolo cerchio rappresenta un'inversione logica, e si può collocare all'ingresso o all'uscita di una funzione logica rappresentata graficamente. Pertanto, valgono le equivalenze seguenti:



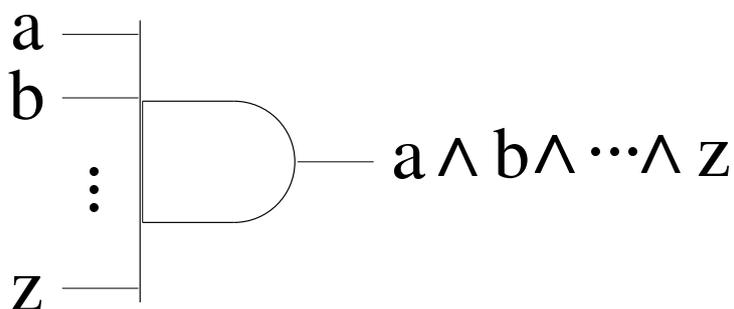
Connettivo AND

I connettivi logici sono gli operatori che utilizzano due ingressi. Il connettivo AND restituisce il valore *Vero* solo se entrambi i valori in ingresso sono pari a *Vero*. Per esempio, se la variabile logica «A» contiene il risultato della proposizione dichiarativa «Antonio mangia» e la variabile «B» contiene il risultato di «Piero legge», l'espressione «A AND B» equivale alla proposizione «Antonio mangia e Piero legge».

a	b	$a \wedge b$
0	0	0
0	1	0
1	0	0
1	1	1



La porta logica AND, quando ha più di due ingressi, esprime l'equivalente di un'espressione in cui tutte le variabili in ingresso sono collegate dall'operatore AND:

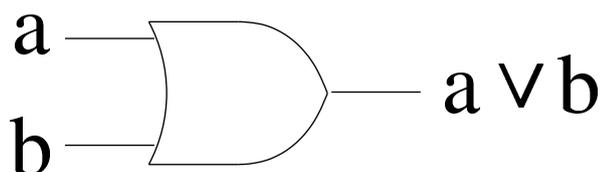


Connettivo OR

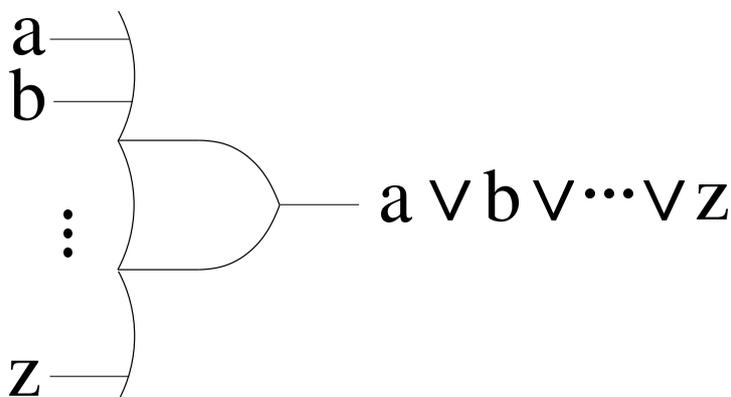
Il connettivo OR restituisce il valore *Vero* se almeno uno dei due ingressi dispone di un valore pari a *Vero*. Per esempio, se la variabile logica «A» contiene il risultato della proposizione dichiarativa «Antonio mangia» e la variabile «B» contiene il risultato di «Piero leg-

ge», l'espressione «A OR B» equivale alla proposizione «Antonio mangia e/o Piero legge».

a	b	$a \vee b$
0	0	0
0	1	1
1	0	1
1	1	1



La porta logica AND, quando ha più di due ingressi, esprime l'equivalente di un'espressione in cui tutte le variabili in ingresso sono collegate dall'operatore OR:

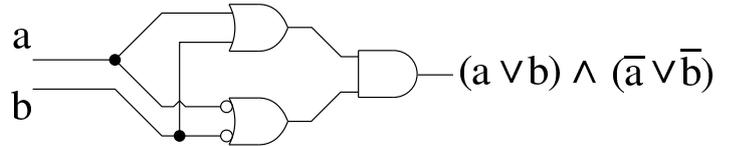
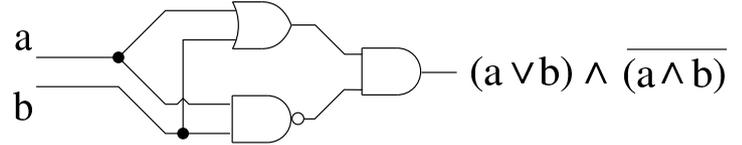
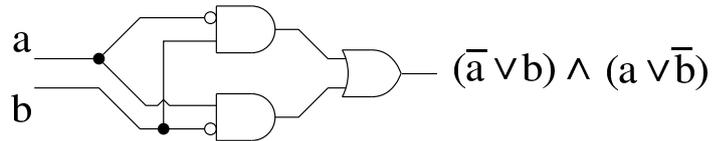
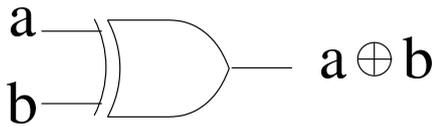


Connettivo XOR

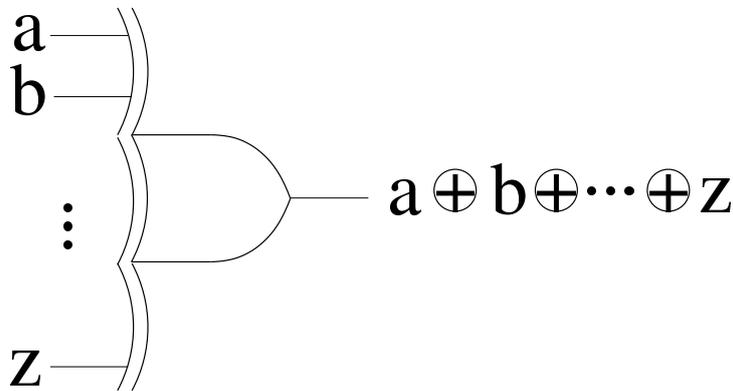
«

Il connettivo XOR restituisce il valore *Vero* se solo uno dei due ingressi dispone di un valore pari a *Vero*. Per esempio, se la variabile logica «A» contiene il risultato della proposizione dichiarativa «Antonio mangia» e la variabile «B» contiene il risultato di «Piero legge», l'espressione «A XOR B» equivale alla proposizione «Antonio mangia oppure Piero legge». Va comunque osservato che il connettivo XOR si considera derivato dagli altri e può essere tradotto in forme diverse, ma equivalenti.

a	b	$a \oplus b$
0	0	0
0	1	1
1	0	1
1	1	0



Anche la porta logica XOR può avere più ingressi, come avviene per AND e OR:



Ordine di precedenza

Le espressioni logiche vanno risolte tenendo conto che gli operatori hanno un ordine di precedenza: NOT, AND, OR. Per esempio, $a+b \cdot c'$ va inteso come $a+(b \cdot (c'))$. Naturalmente, l'ordine di precedenza può essere modificato utilizzando opportunamente le parentesi.

Valori irrilevanti nelle tabelle di verità

A volte si possono semplificare le tabelle di verità di una funzione logica quanto alcune variabili in ingresso, in certe circostanze, possono assumere qualsiasi valore senza interferire nel risultato; in quei

casi, al posto del valore si mette convenzionalmente una lettera «x». A titolo di esempio, si consideri la funzione logica che si ottiene dall'espressione $A \cdot (B + C)$. Nella figura successiva si vede la tabella di verità completa e a fianco una versione semplificata della stessa.

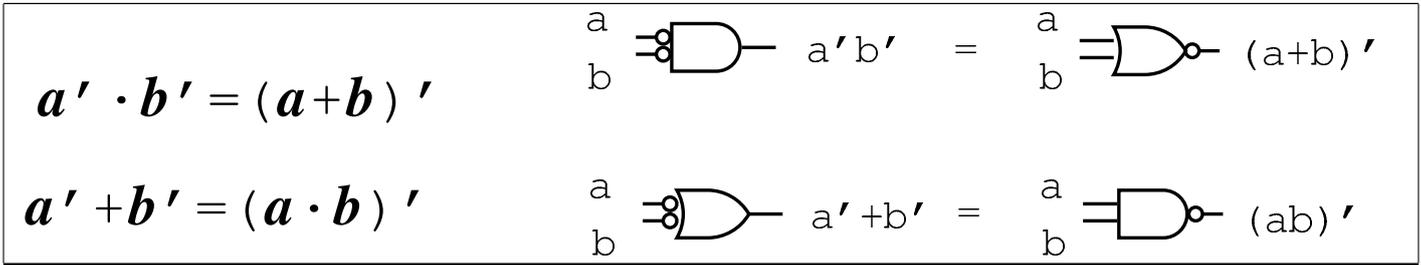
Figura u97.11. Semplificazione delle tabelle di verità in presenza di condizioni in cui i valori di certe variabili diventano irrilevanti.

A	B	C	A(B+C)	=	A	B	C	A(B+C)
0	0	0	0		0	x	x	0
0	0	1	0		1	0	0	0
0	1	0	0		1	0	1	1
0	1	1	0		1	1	0	1
1	0	0	0		1	1	1	1
1	0	1	1		1	1	1	1
1	1	0	1		1	1	1	1
1	1	1	1					

Equivalenze

«

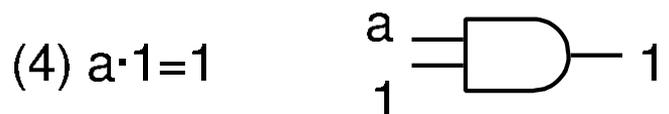
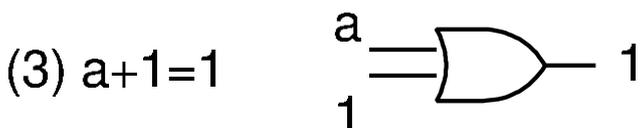
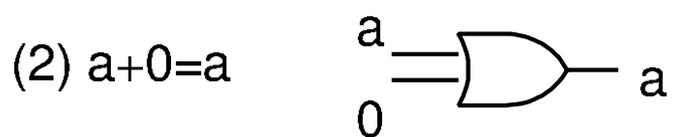
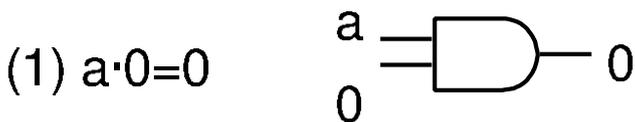
Attraverso gli operatori logici si possono costruire delle espressioni complesse, nelle quali esiste la facoltà di applicare delle trasformazioni, di solito con lo scopo di cercare di semplificarle, mantenendo però lo stesso risultato. Per questo sono molto importanti i teoremi di De Morgan, secondo i quali:

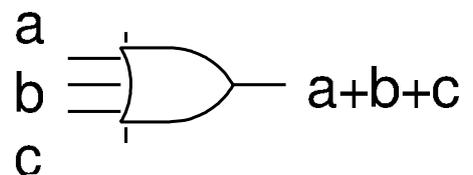
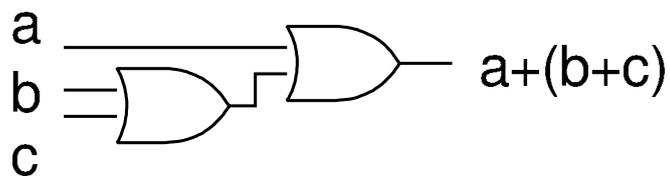
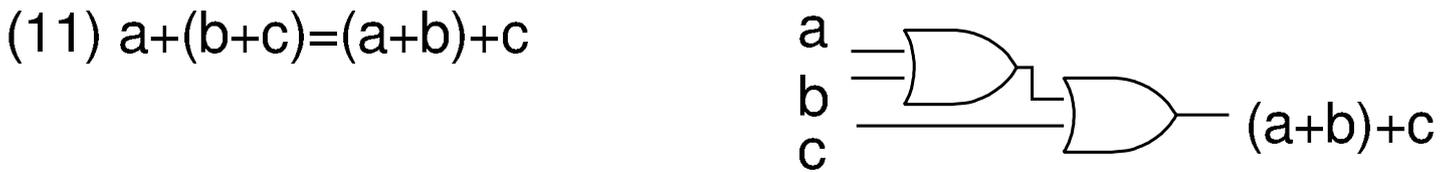
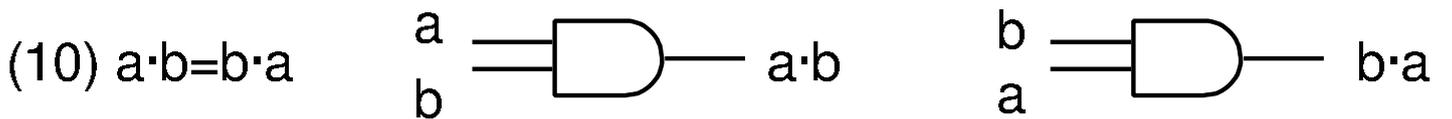
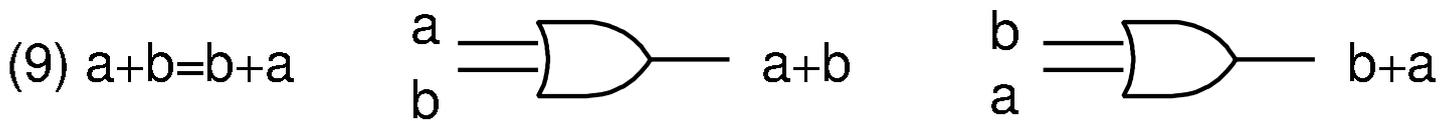
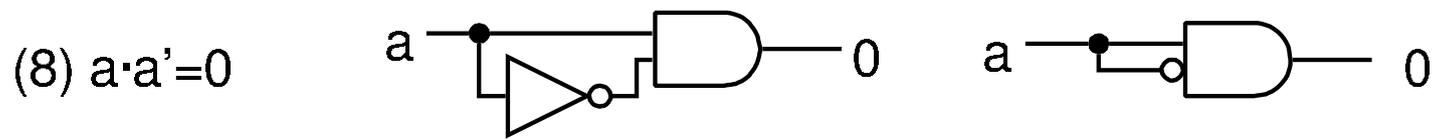
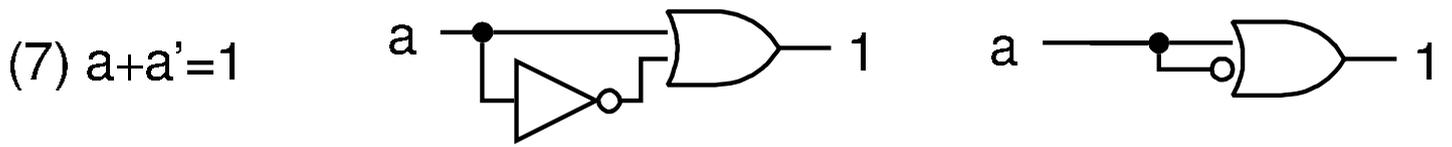
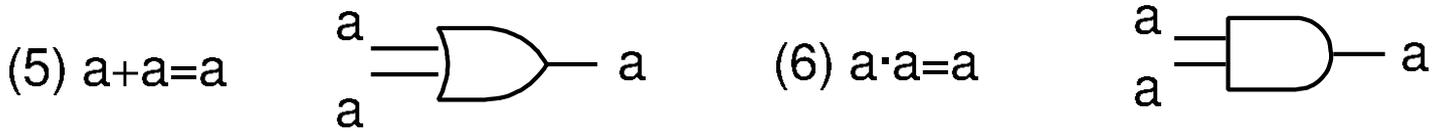


Mettendo assieme le tabelle della verità dei vari operatori logici con i teoremi di De Morgan, si ottengono le equivalenze della tabella successiva su cui si basa l'algebra «logica», ovvero l'algebra di Boole.

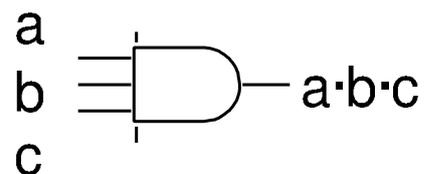
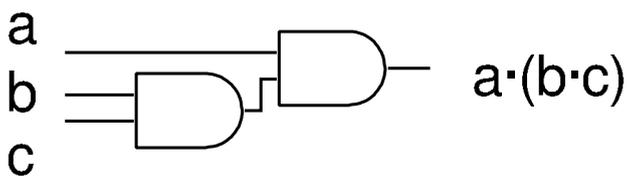
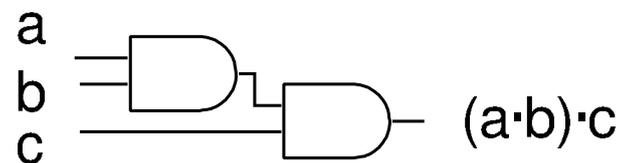
Tabella u97.13. Equivalenze dell'algebra di Boole. Si intende che ab rappresenti semplicemente $a \cdot b$ in modo compatto.

(1) $a+0 = a$	(2) $a \cdot 0 = 0$
(3) $a+1 = 1$	(4) $a \cdot 1 = a$
(5) $a+a = a$	(6) $a \cdot a = a$
(7) $a+a' = 1$	(8) $a \cdot a' = 0$
(9) $a+b = b+a$	(10) $ab = ba$
(11) $a+(b+c) = (a+b)+c$	(12) $a(bc) = (ab)c$
(13) $a(b+c) = ab+ac$	(14) $a+bc = (a+b)(a+c)$
(15) $(a+b)' = a'b'$	(16) $(ab)' = a'+b'$
(17) $(a')' = a$	

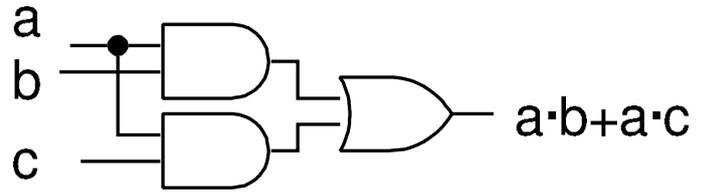
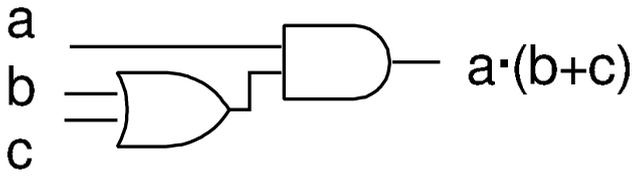




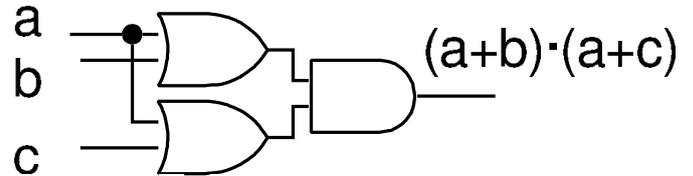
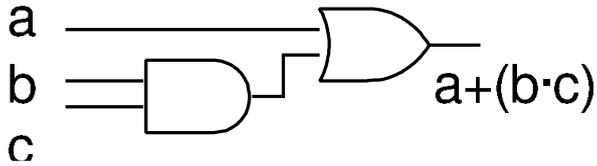
(12) $a \cdot (b \cdot c)=(a \cdot b) \cdot c$



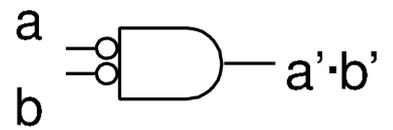
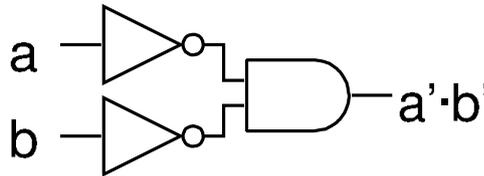
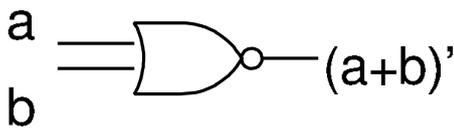
(13) $a \cdot (b+c) = a \cdot b + a \cdot c$



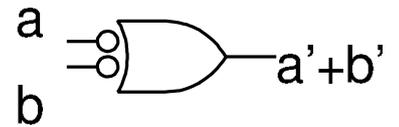
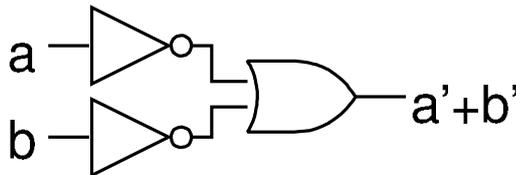
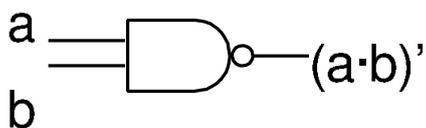
(14) $a + (b \cdot c) = (a+b) \cdot (a+c)$



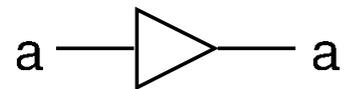
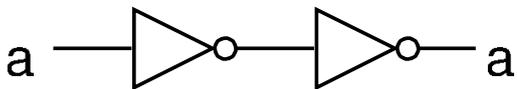
(15) $(a+b)' = a' \cdot b'$



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



(17) $(a')' = a$



Quando nelle espressioni si utilizzano solo gli operatori logici tradizionali (AND, OR, NOT), i teoremi di De Morgan consentono di trasformare facilmente una funzione logica nel suo complemento. In pratica, data la funzione f , si ottiene la negazione logica f' seguendo la procedura seguente:

- si scambiano gli operatori AND originali con gli operatori OR, mettendo delle parentesi per non modificare l'ordine di valutazione;
- si scambiano gli operatori OR originali con gli operatori AND, facendo in modo di evitare che questo cambiamento cambi l'ordine di esecuzione della valutazione;
- si complementano tutte le variabili (operatore NOT).

Si veda l'esempio seguente:

$$f = a + b'c + de' + g'h'$$

$$f' = a'(b+c')(d'+e)(g+h)$$

Somma dei prodotti

«

Una funzione logica è un'espressione composta da variabili, costanti logiche e operatori logici, la quale produce un risultato logico (*Vero* o *Falso*). Tuttavia, una funzione logica può essere descritta semplicemente attraverso la tabella di verità che abbina i valori delle variabili al risultato atteso per ogni combinazione delle stesse.

È possibile sintetizzare il comportamento di una funzione logica attraverso ciò che si definisce come *somma dei prodotti*: si parte dalla tabella di verità che si vuole ottenere e si elencano i *prodotti fondamentali* logici che descrivono ogni condizione degli ingressi.

Tabella u97.14. Prodotti fondamentali delle funzioni aventi due, tre e quattro variabili.

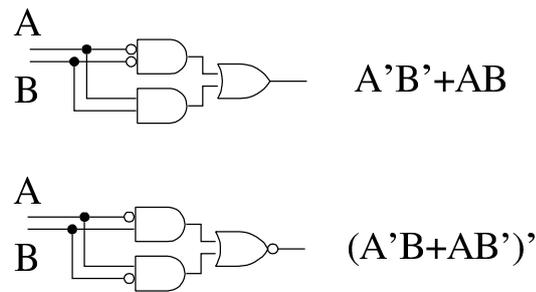
A	B	<i>prodotto fondamentale</i>	A	B	C	D	<i>prodotto fondamentale</i>
0	0	$A'B'$	0	0	0	0	$A'B'C'D'$
0	1	$A'B$	0	0	0	1	$A'B'C'D$
1	0	AB'	0	0	1	0	$A'B'CD'$
1	1	AB	0	0	1	1	$A'B'CD$
			0	1	0	0	$A'BC'D'$
			0	1	0	1	$A'BC'D$
			0	1	1	0	$A'BCD'$
			0	1	1	1	$A'BCD$
			1	0	0	0	$AB'C'D'$
			1	0	0	1	$AB'C'D$
			1	0	1	0	$AB'CD'$
			1	0	1	1	$AB'CD$
			1	1	0	0	$ABC'D'$
			1	1	0	1	$ABC'D$
			1	1	1	0	$ABCD'$
			1	1	1	1	$ABCD$

Per esempio, la funzione nota come NXOR (funzione di due variabili che si avvera solo quando le due variabili hanno lo stesso valore) si sintetizza secondo il procedimento che si può comprendere intuitivamente dalla figura successiva.

Figura u97.15. Sintesi a partire dalla tabella di verità della funzione NXOR, attraverso gli operatori logici fondamentali.

A	B	prodotto fondamentale	risultato atteso	sintesi normale	sintesi inversa
0	0	$A'B'$	1	$A'B'$	
0	1	$A'B$	0		$A'B$
1	0	AB'	0		AB'
1	1	AB	1	AB	

\downarrow (sintesi normale) \rightarrow $A'B' + AB$
 \downarrow (sintesi inversa) \rightarrow $(A'B + AB')'$
 \downarrow (sintesi normale) \rightarrow $A'B' + AB$



L'esempio della figura permette di ottenere due alternative, secondo quello che è noto come «somma dei prodotti», ma ciò non toglie che ci possano essere altre possibilità di sintesi.

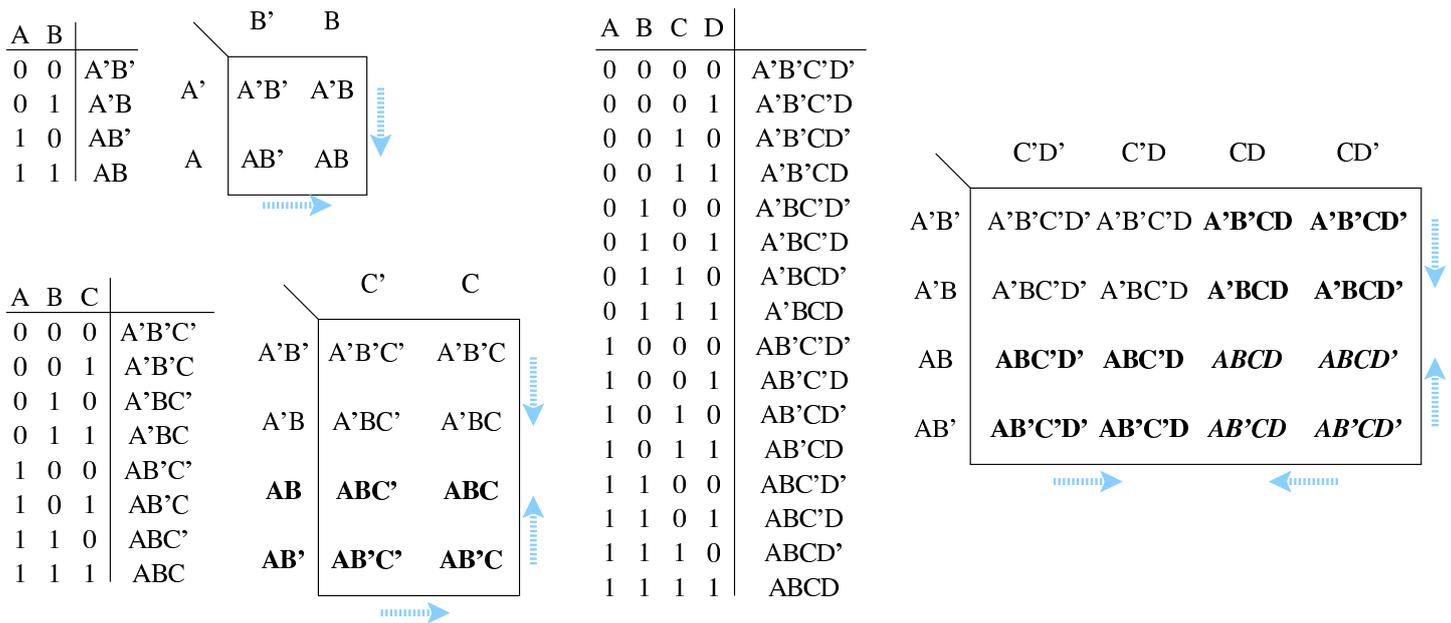
Mappe di Karnaugh

«

Una funzione logica può essere rappresentata in una mappa di Karnaugh, a partire dalla sua tabella di verità; attraverso tale mappa si può poi cercare un modo per semplificare l'espressione che sintetizza la funzione e di conseguenza il circuito che la realizza.

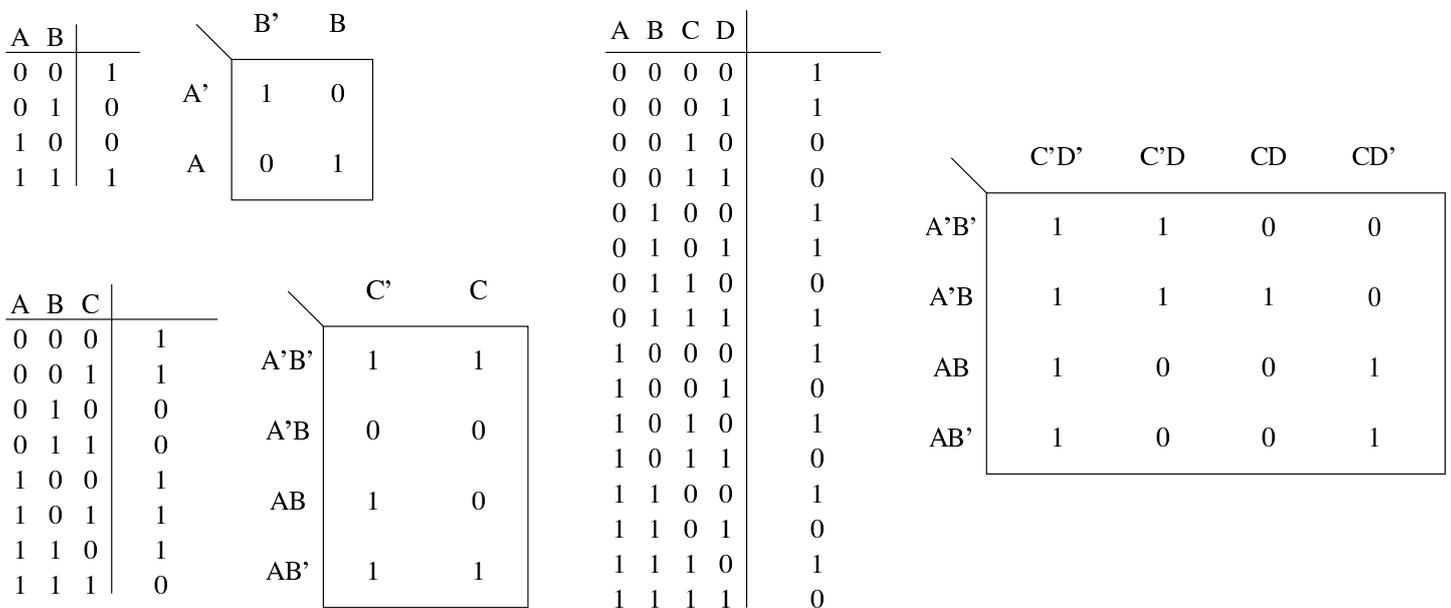
La mappa di Karnaugh è una rappresentazione bidimensionale dei prodotti fondamentali di una tabella di verità da analizzare: in pratica si dividono a metà i prodotti fondamentali e si rappresentano in un rettangolo. Nelle immagini successive si mostrano tre mappe, vuote, relative al caso di due, tre e quattro variabili: bisogna fare molta attenzione alla sequenza dei prodotti fondamentali, perché questa non corrisponde a quella che si usa normalmente nelle tabelle della verità.

Figura u97.16. Mappe di Karnaugh contenenti la definizione dei prodotti fondamentali a cui ogni cella fa riferimento.



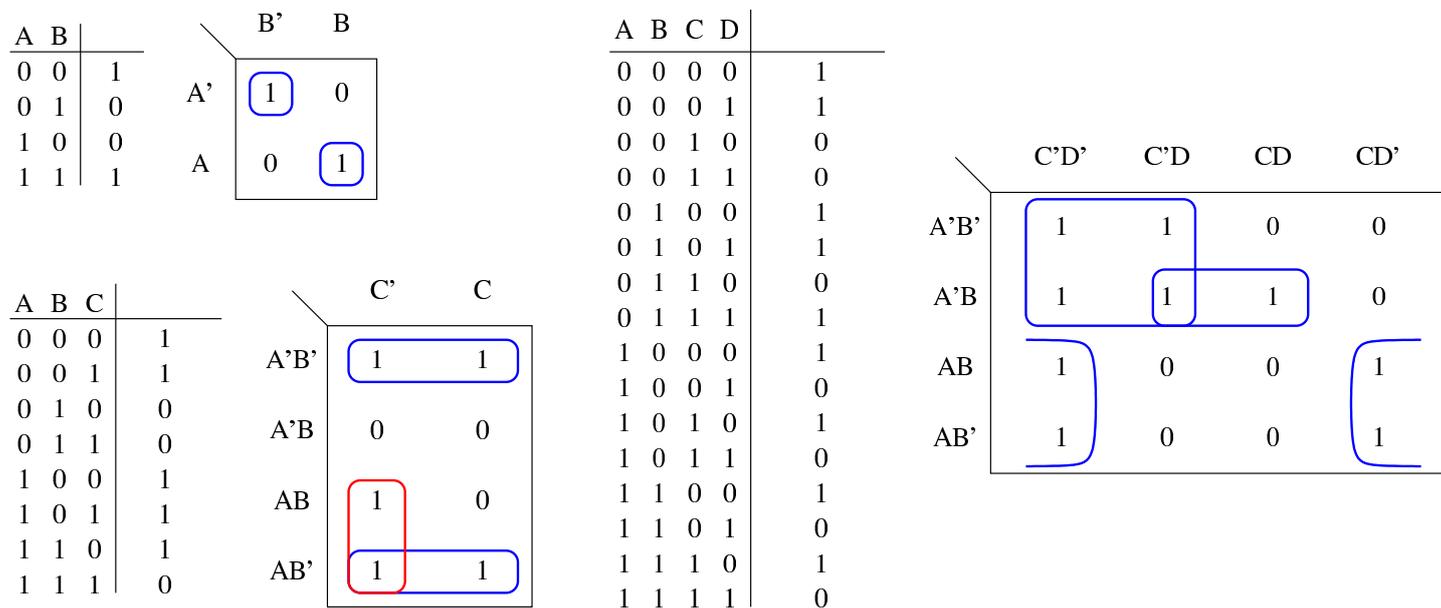
Le mappe vanno compilate mettendo nelle celle la cifra 1 in corrispondenza dei prodotti fondamentali validi. Le immagini successive propongono degli esempi, confrontando la tabella di verità con la mappa compilata corrispondente.

Figura u97.17. Mappe di Karnaugh compilate in base alle tabelle di verità che appaiono al loro fianco.



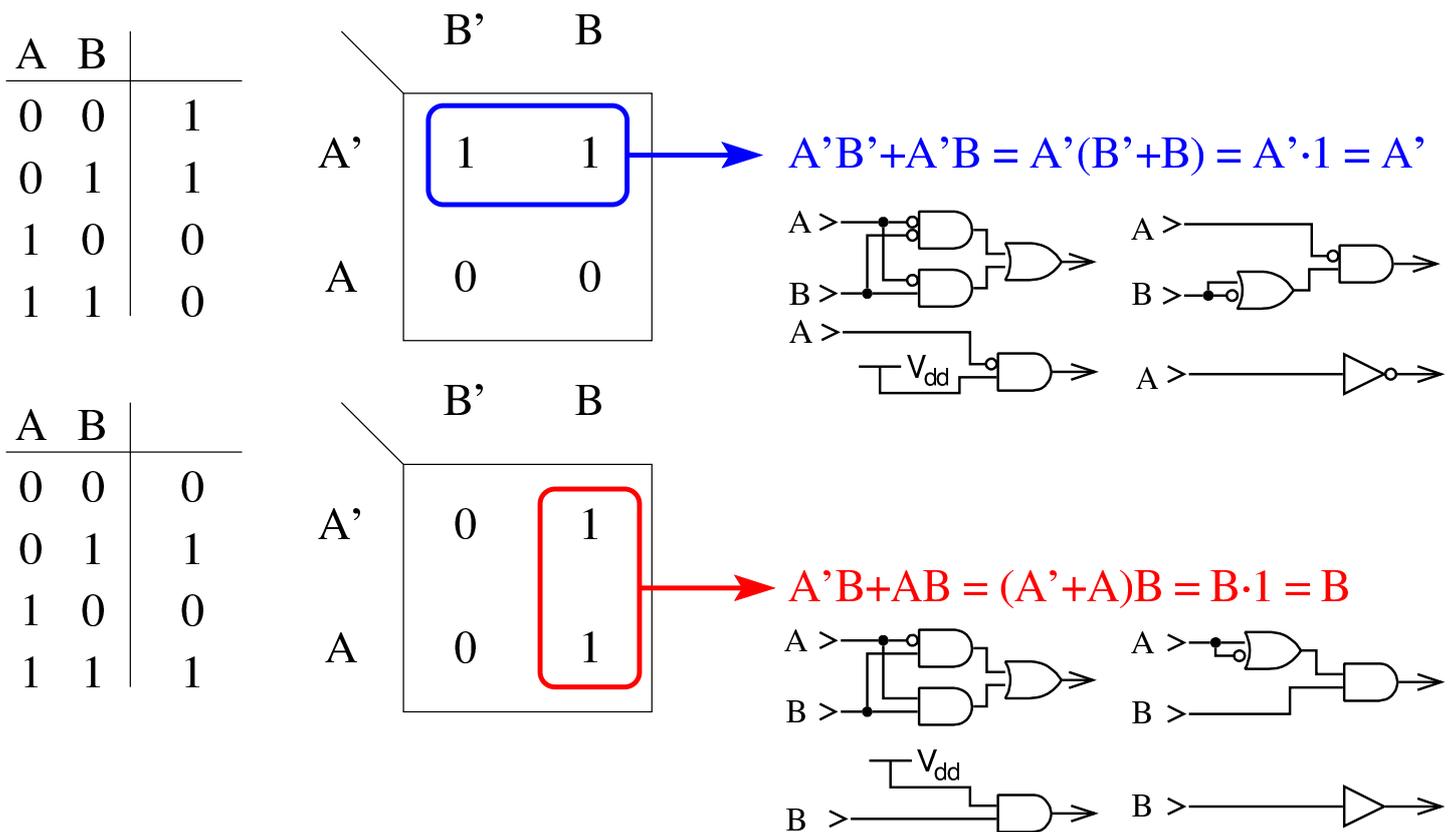
Una volta compilata la mappa, si vanno a raggruppare le celle che contengono la cifra 1 e che si trovano adiacenti in senso orizzontale o verticale.

Figura u97.18. Mappe di Karnaugh con le celle raggruppate verticalmente e orizzontalmente.



Quando due celle adiacenti, in verticale o in orizzontale, contengono il valore 1, una variabile che riguarda le due celle diventa inutile. La figura successiva dimostra intuitivamente il procedimento.

Figura u97.19. Semplificazione in presenza di due celle adiacenti con valore 1.



La semplificazione deriva dal fatto che $x \text{ OR } x'$ è sempre vero; pertanto, quando un raggruppamento rettangolare dimostra che una variabile appare sia nella sua forma normale, sia negata, significa che si può semplicemente ignorare.

Figura u97.20. Esempi di semplificazioni con mappe a due e tre variabili.

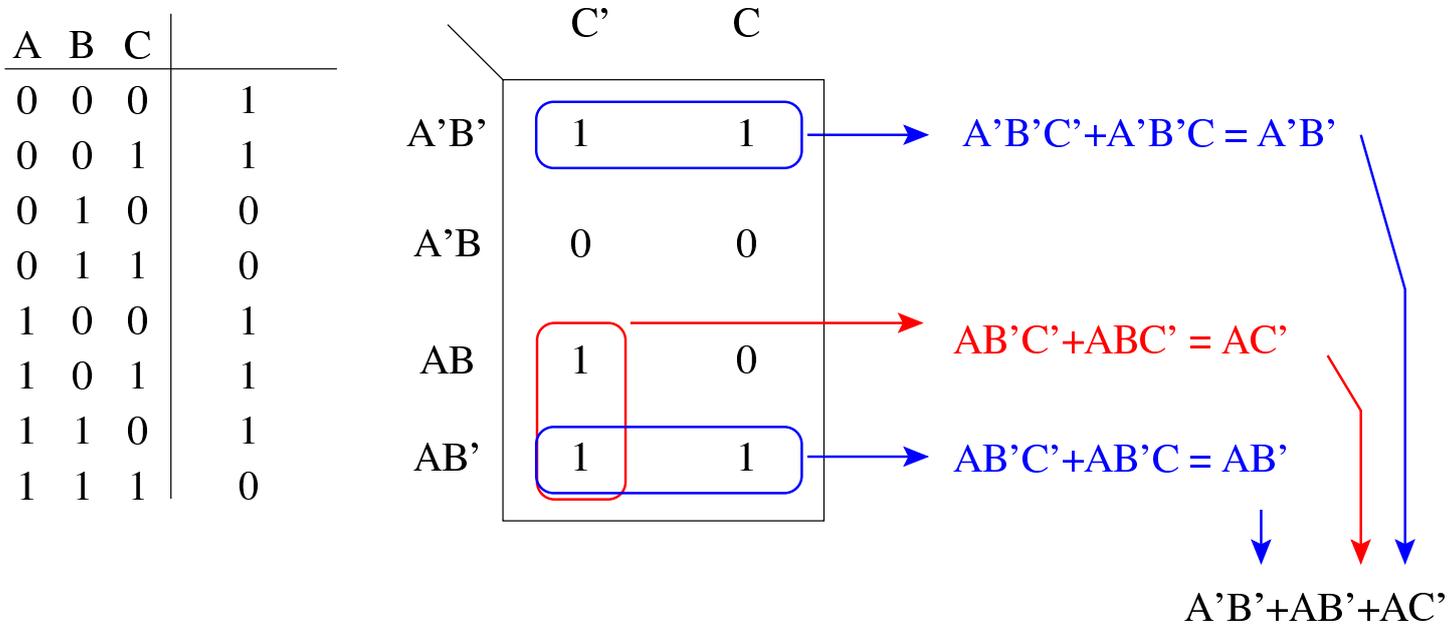
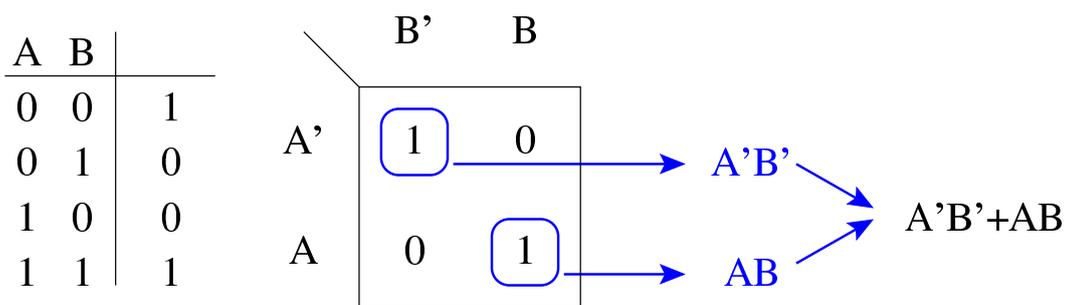
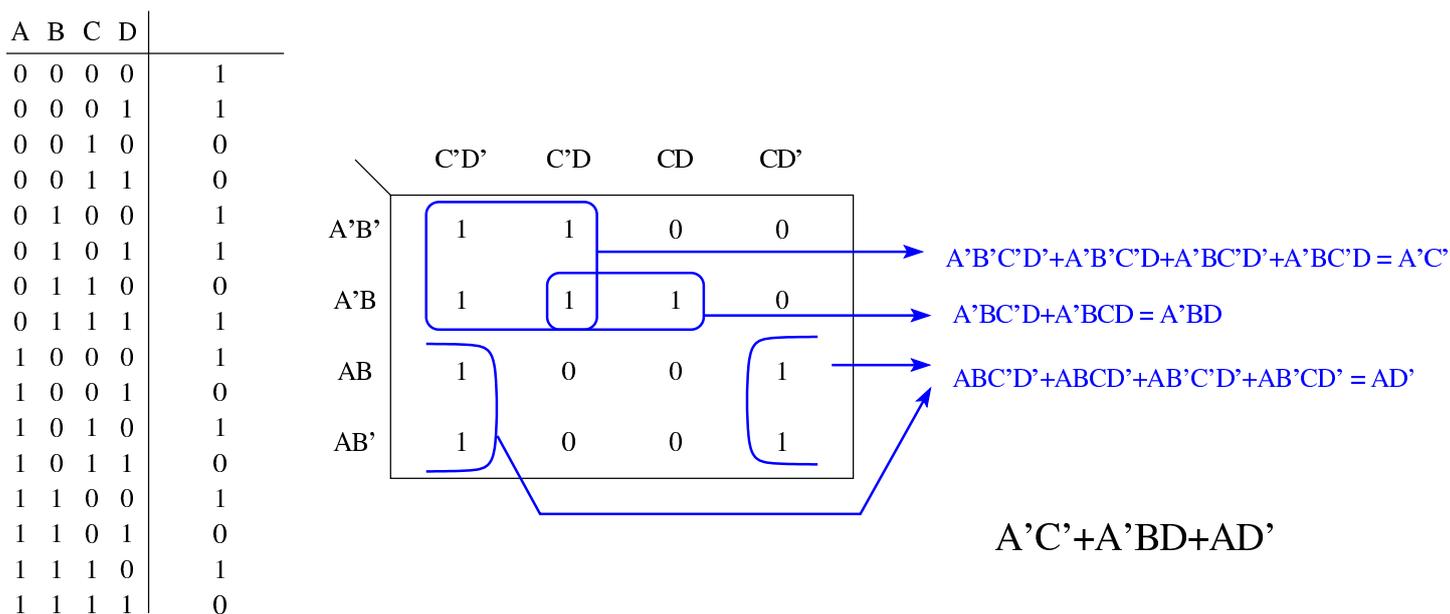
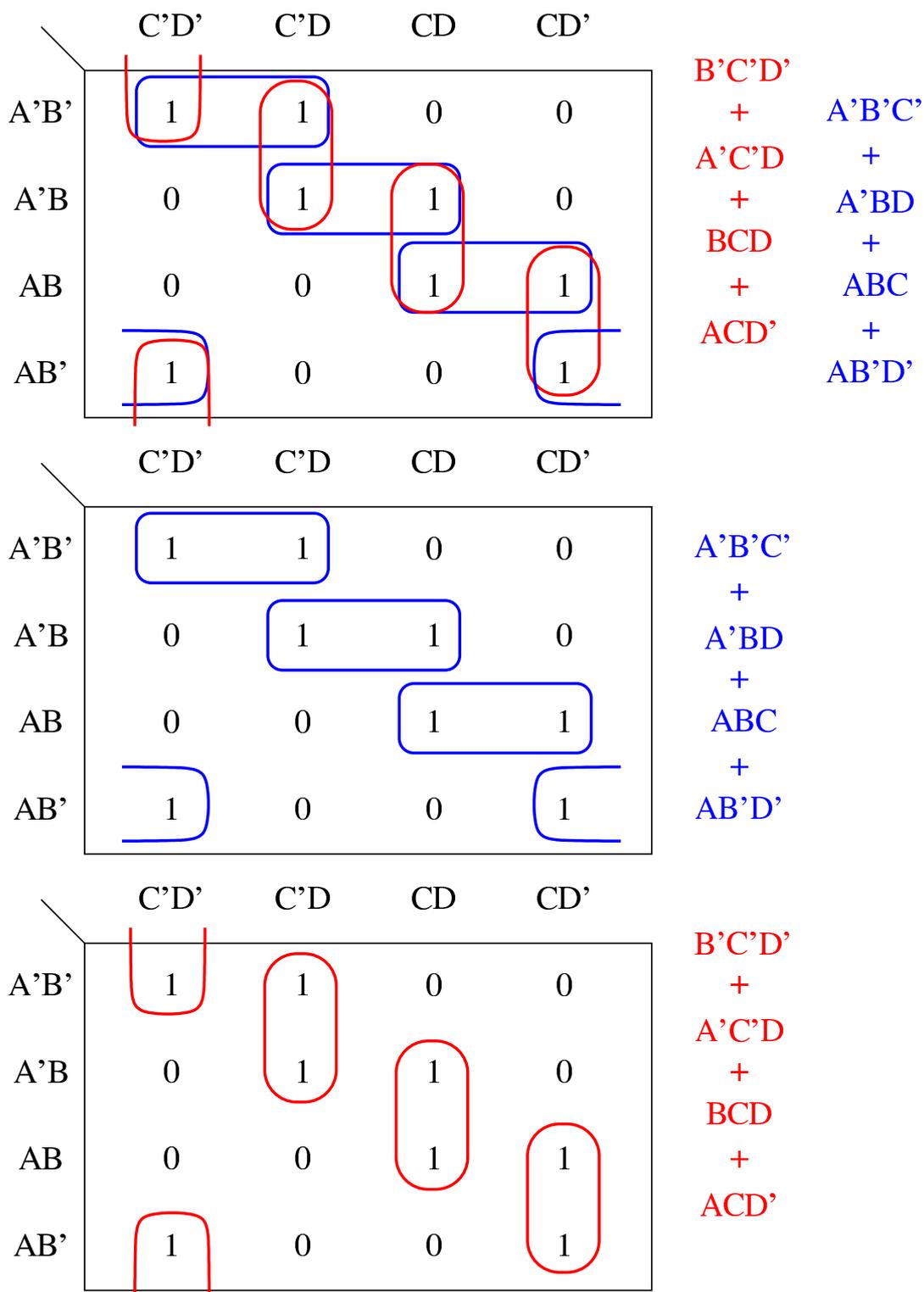


Figura u97.21. Esempio di semplificazione con mappa a quattro variabili.



Nella delimitazione delle zone che consentono di ottenere una semplificazione logica, si possono ignorare le zone che si sovrappongono completamente con altre, come si vede nella figura successiva, dove si possono produrre due risultati alternativi equivalenti.

Figura u97.22. Esempi di semplificazioni in presenza di aree sovrapposte e ridondanti.

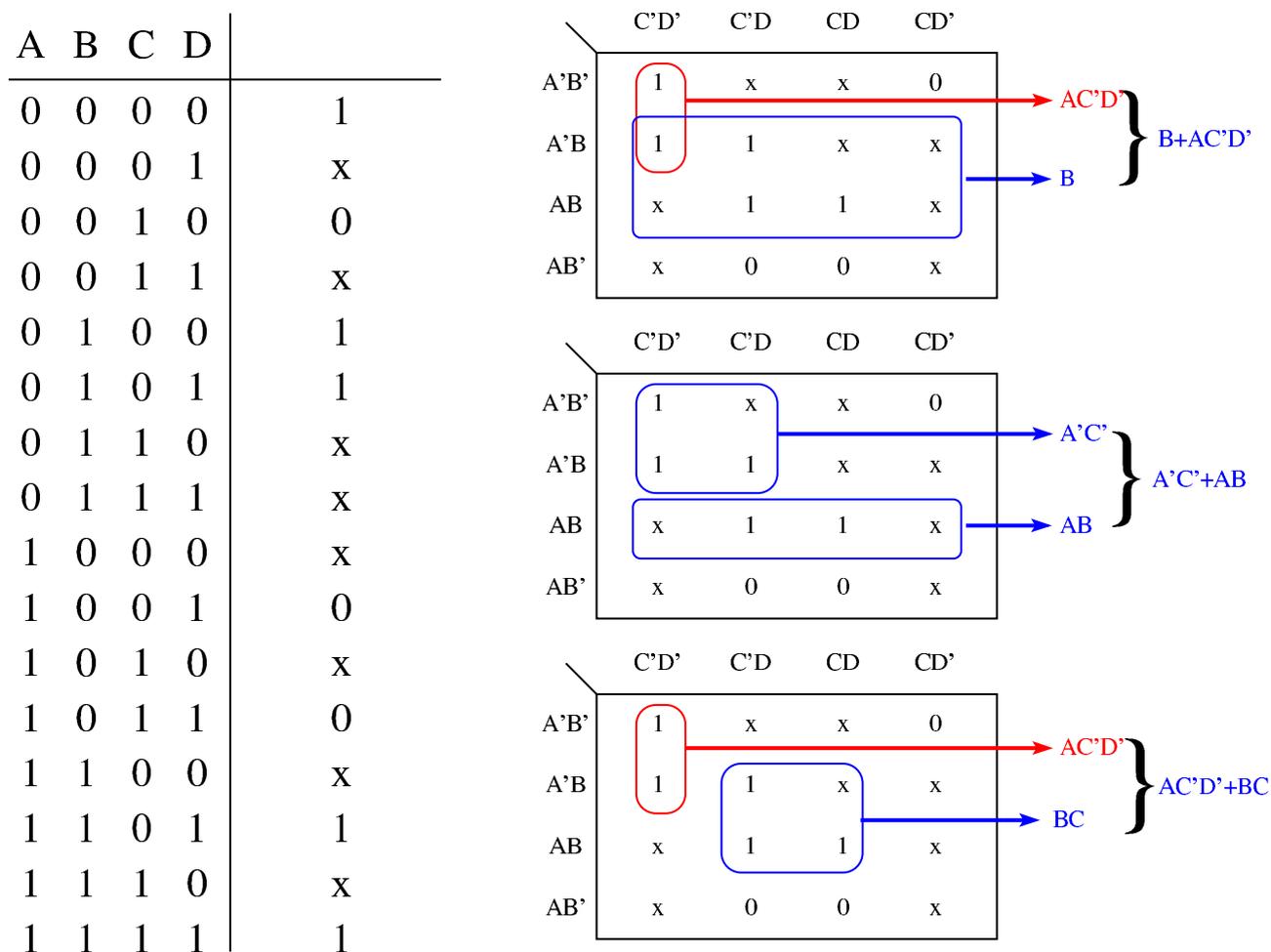


Le mappe di Karnaugh permettono di semplificare facilmente una funzione logica, purché non si superino le quattro variabili: di-

versamente sarebbe necessaria una rappresentazione a tre o più dimensioni.

In situazioni particolari, può accadere che sia indifferente il valore prodotto da alcune combinazioni delle variabili di ingresso. In questi casi, per sintetizzare la rete combinatoria si può scegliere liberamente il valore di uscita che risulta più conveniente nell'ottica della semplificazione.

Figura u97.23. Esempio di semplificazione in presenza di esiti indifferenti; si mostrano tre ipotesi di soluzione.



Le mappe di Karnaugh si possono usare anche in presenza di più funzioni che condividono le stesse variabili di ingresso. In tal caso si valutano tante mappe quante sono le funzioni per cercare una sintesi

per ognuna; successivamente, se si realizzano queste funzioni attraverso un circuito logico, si cercano elementi comuni per condividerli senza duplicarli.

Figura u97.24. Due funzioni delle stesse variabili, sintetizzate attraverso due mappe.

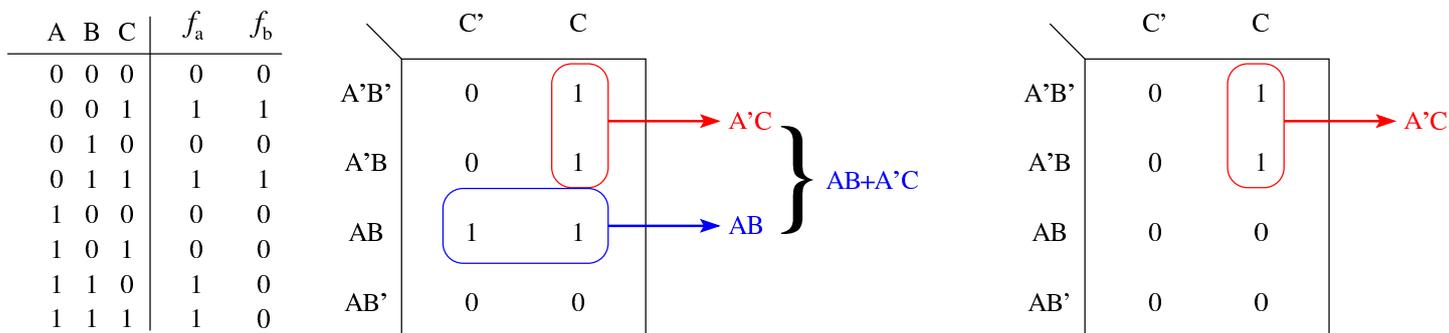
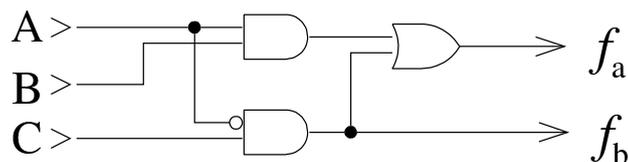
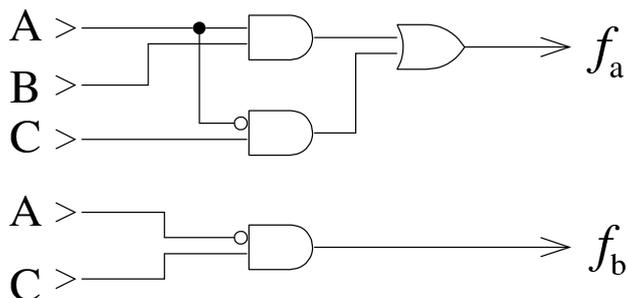


Figura u97.25. Soluzione in forma di circuito logico, prima separata, poi unita risparmiando una porta logica.



Mappe di Karnaugh con la funzione XOR

«

Le mappe di Karnaugh danno soluzioni formate attraverso gli operatori logici fondamentali (AND, OR, NOT); tuttavia, spesso è comodo avvalersi dell'operatore XOR, al pari di quelli fondamentali. Nelle mappe di Karnaugh l'operatore XOR si manifesta in modo particolare, come si vede nella figura successiva.

Figura u97.26. Mappe di Karnaugh relative a funzioni XOR a due, tre e quattro variabili.

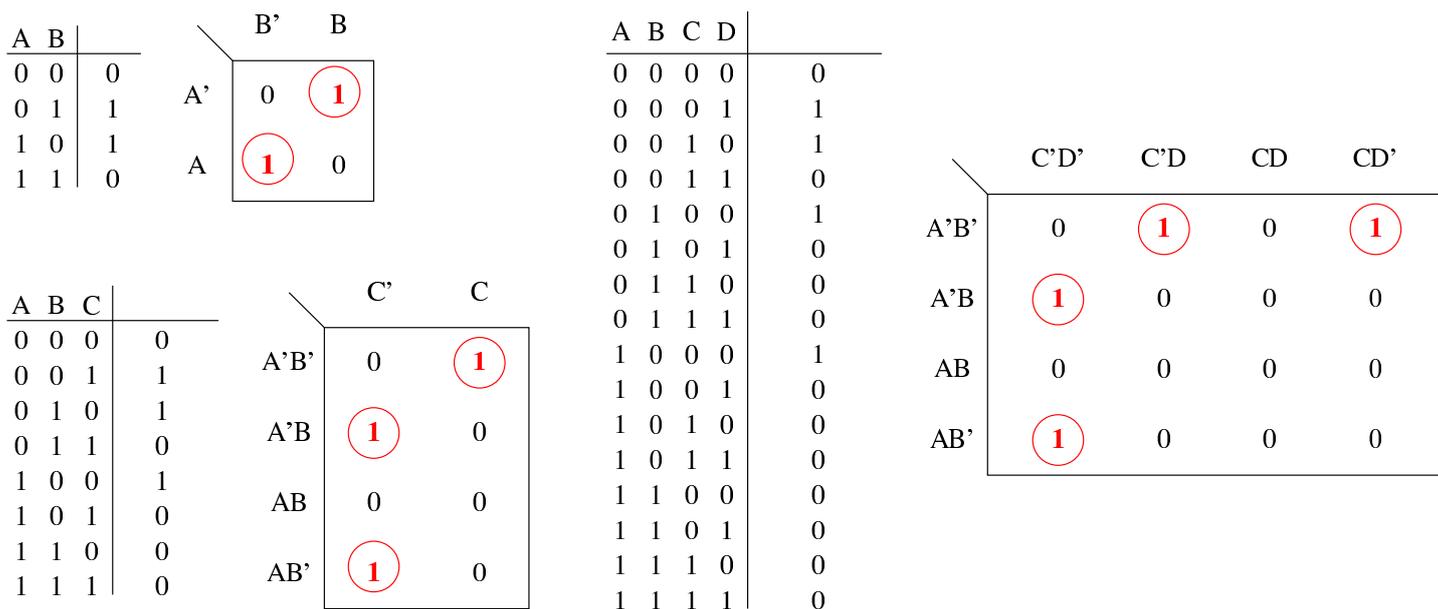
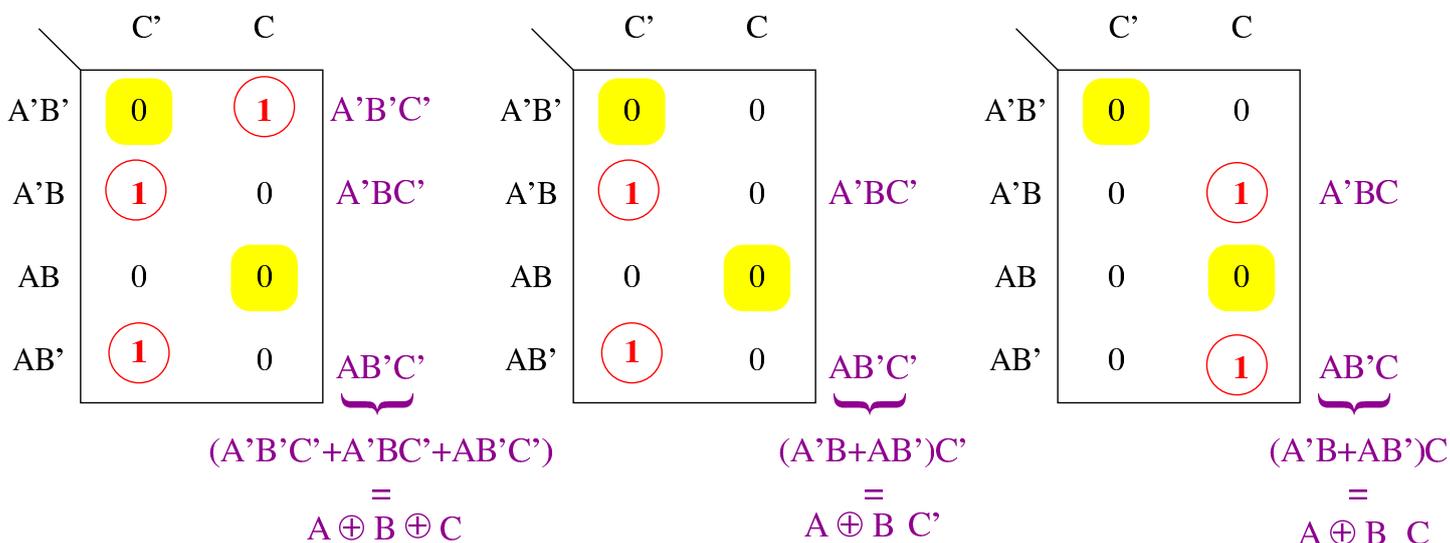


Figura u97.27. Esempi di individuazione di funzioni XOR a due variabili nelle mappe di Karnaugh a ingressi variabili. Le zone colorate di giallo sono impossibili, ammesso che le variabili usate per le funzioni XOR non siano negate.



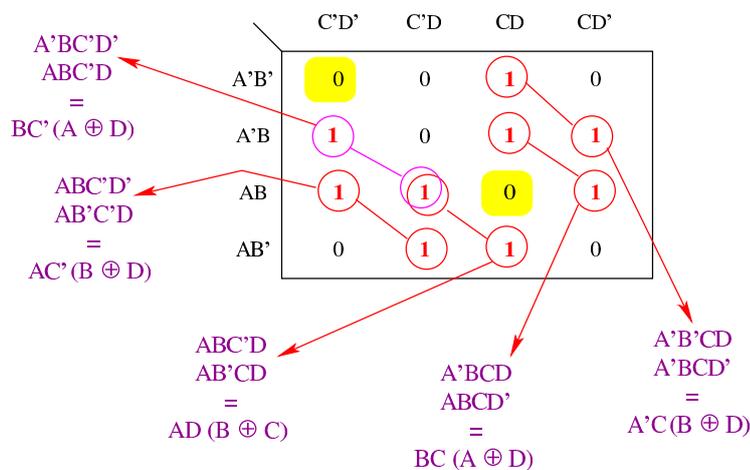
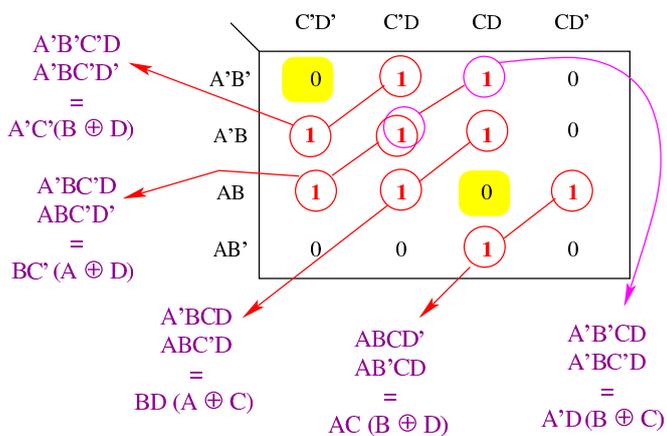
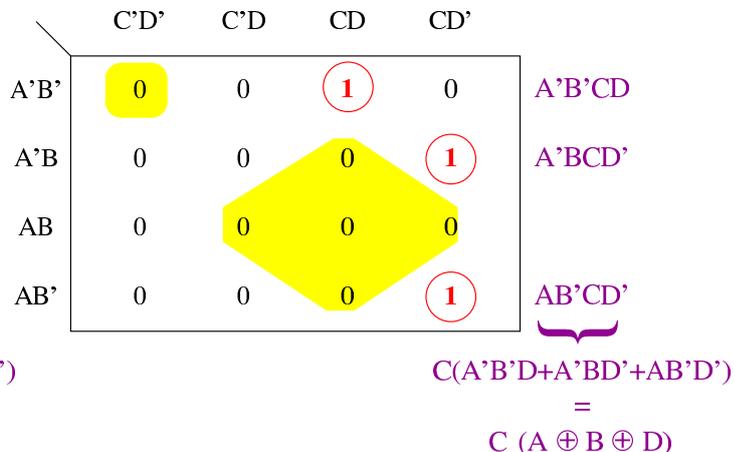
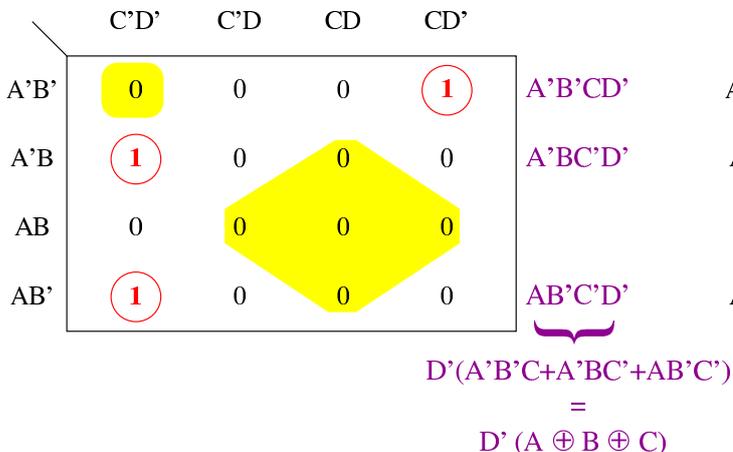
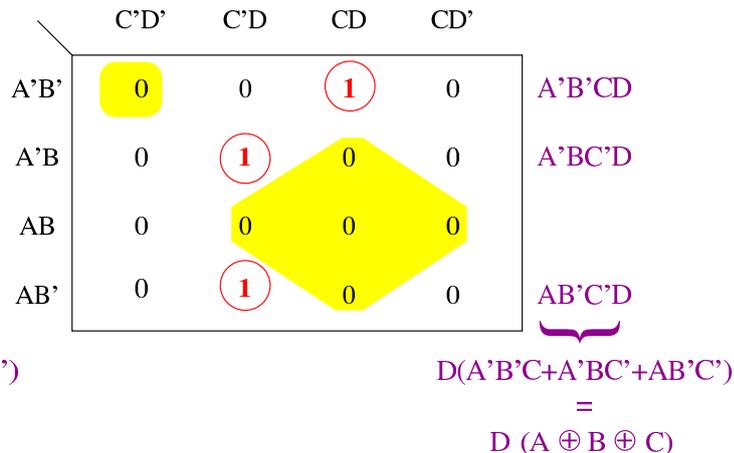
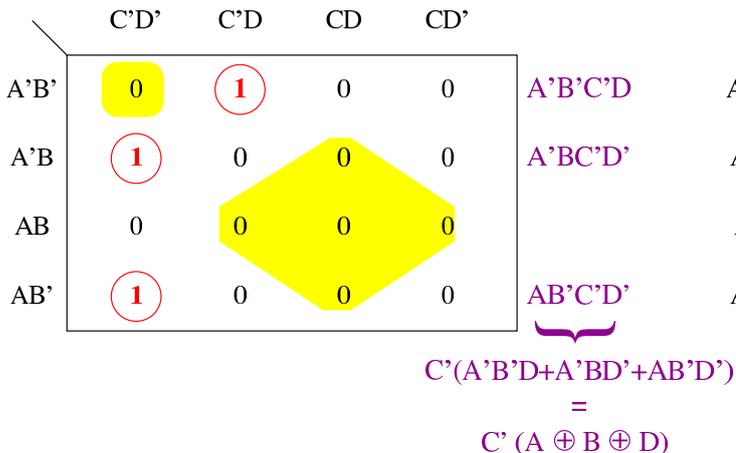


Figura u97.29. Esempi di individuazione di funzioni XOR a tre variabili nelle mappe di Karnaugh a quattro ingressi: le zone in giallo sono impossibili, ammesso che le variabili usate per le funzioni XOR non siano negate.



	C'D'	C'D	CD	CD'
A'B'	0	1	0	1
A'B	1	0	0	0
AB	0	0	0	0
AB'	0	0	0	0

A'B'C'D
A'B'CD'

A'BC'D'

$$A'(B'C'D+B'CD'+BC'D')$$

$$=$$

$$A'(B \oplus C \oplus D)$$

	C'D'	C'D	CD	CD'
A'B'	0	0	0	0
A'B	0	1	0	1
AB	1	0	0	0
AB'	0	0	0	0

A'BC'D
A'BCD'

ABC'D'

$$B(A'C'D+A'CD'+AC'D')$$

$$=$$

$$B(A \oplus B \oplus C)$$

	C'D'	C'D	CD	CD'
A'B'	0	1	0	1
A'B	0	0	0	0
AB	0	0	0	0
AB'	1	0	0	0

A'B'C'D
A'B'CD'

AB'C'D'

$$B'(A'C'D+A'CD'+AC'D')$$

$$=$$

$$B'(A \oplus B \oplus C)$$

	C'D'	C'D	CD	CD'
A'B'	0	0	0	0
A'B	0	0	0	0
AB	1	0	0	0
AB'	0	1	0	1

ABC'D'
AB'C'D
AB'CD'

$$A(BC'D'+B'C'D+B'CD')$$

$$=$$

$$A(B \oplus C \oplus D)$$