

## CAPITOLO III CIRCUITI COMBINATORI

### 3.1) Introduzione.

Si chiamano combinatori quei circuiti il cui funzionamento, per quanto riguarda la relazione ingresso uscita, e' descritto da una funzione logica; quelli cioe' per i quali gli ingressi e le uscite possono assumere solo uno di due valori nettamente distinti ed in cui l'uscita, istante per istante, e' funzione unicamente degli ingressi.

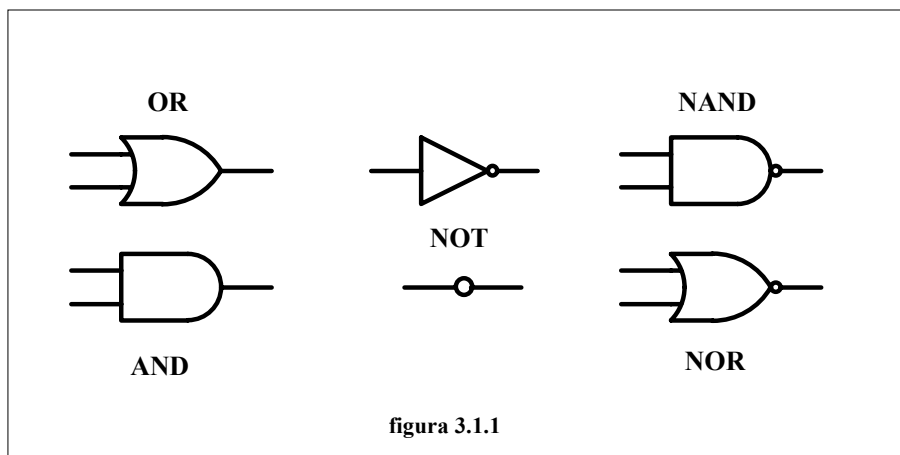
Circuitualmente una funzione logica si realizza usando componenti capaci di assumere l'uno o l'altro di due stati diversi. Nei circuiti elettronici questi due stati sono realizzati mediante due livelli caratteristici di tensione, detti livello alto (**h**) e livello basso (**b**).

L'effettiva corrispondenza tra **h** e **b** e le costanti logiche 0 e 1 e' convenzionale e va precisata di volta in volta.

E' detta **logica positiva** la convenzione secondo la quale il valore 1 viene associato al livello alto **h**; **logica negativa** quello in cui il valore 1 e' associato al livello basso **b**.

Si chiama circuito logico elementare o circuito porta ciascun circuito a  $n$  ingressi e un'uscita il cui valore e' 1 in corrispondenza alle configurazioni degli ingressi descritte dalle funzioni logiche OR, AND, NOT, NAND, NOR.

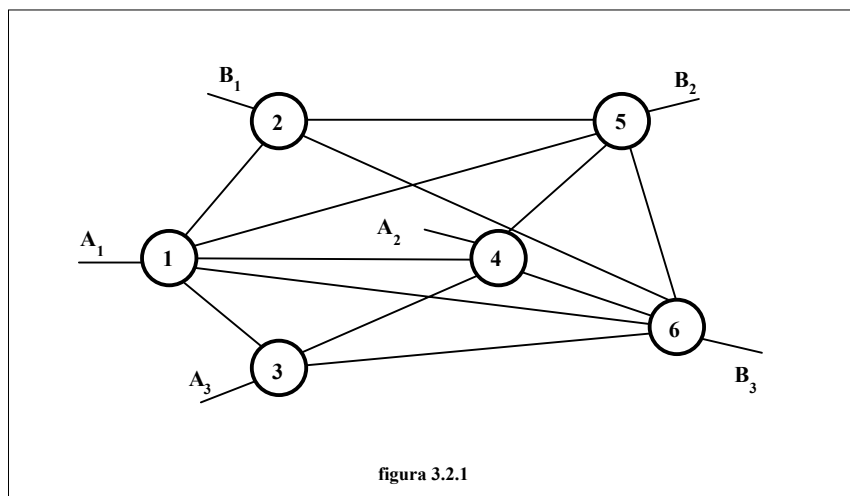
Indipendentemente dalla loro realizzazione circuitale e dal tipo di logica, i circuiti porta vengono indicati graficamente con i simboli di fig. 3.1.1.



Per i circuiti combinatori, come per ogni altro tipo di circuito, si pongono due problemi opposti: da un lato quello dell'analisi, cioe' quello della descrizione del funzionamento del circuito, una volta che sia nota la sua configurazione; dall'altro quello della sintesi, cioe' del progetto di un circuito che realizzi una certa funzione logica, comunque descritta.

### 3.2) Itinerari e livelli.

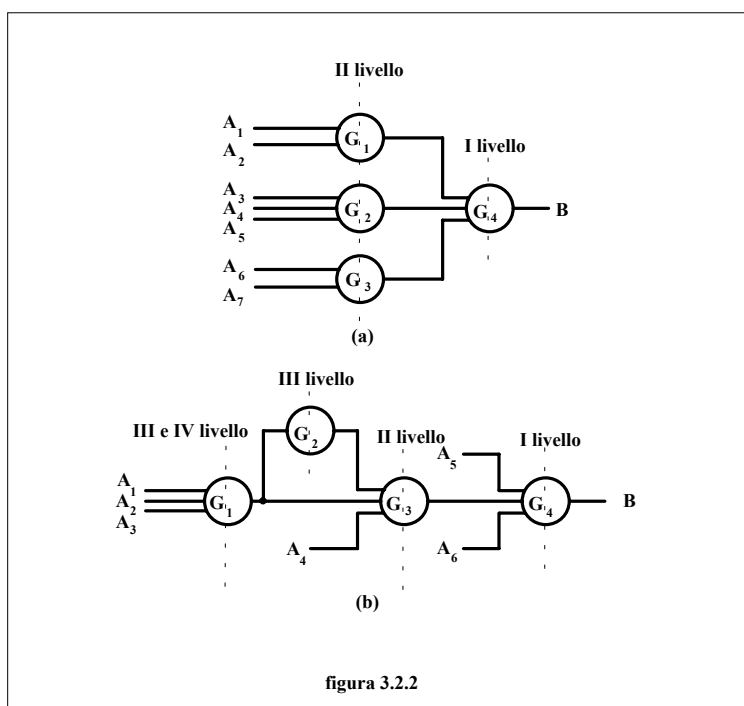
Ogni circuito di commutazione e' formato da un certo numero di elementi (NAND, NOR, ecc.) tra loro variamente interconnessi, da un certo numero di ingressi, contraddistinti in fig. 3.2.1 con i simboli  $A_i$ , e da un certo numero di uscite, contraddistinte nella medesima figura con i simboli  $B_k$ .



Si dice itinerario tra due elementi X e Y qualsiasi percorso che colleghi X e Y. Si dice invece livello di un elemento X rispetto all'uscita  $B_j$  e a un determinato itinerario I il numero di elementi, X compreso, disposti lungo l'itinerario I a partire dall'uscita  $B_j$ .

Livello di una variabile rispetto all'uscita  $B_j$  e all'itinerario I e' il numero di elementi compresi tra il rispettivo ingresso e l'uscita  $B_j$  lungo l'itinerario I.

In fig. 3.2.2 sono riportati due esempi.



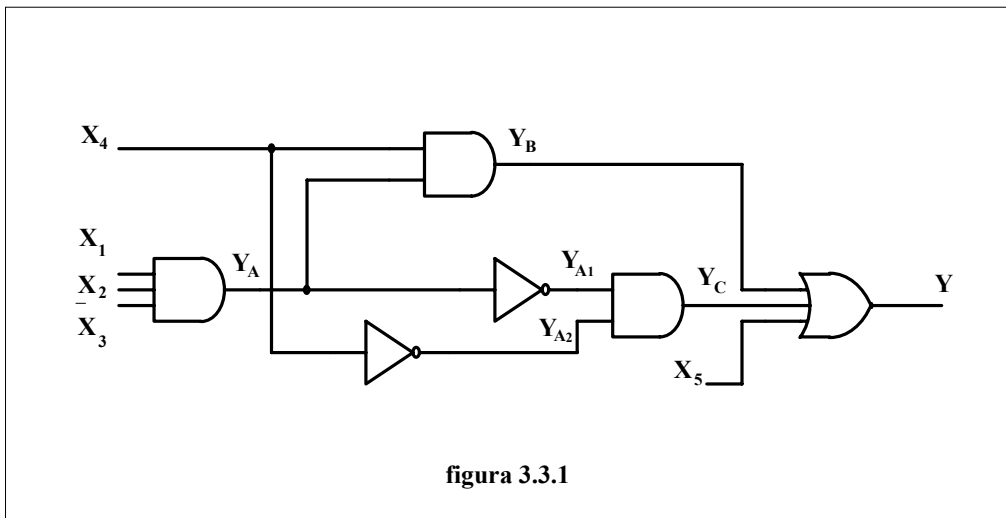
Si noti che uno stesso elemento puo' avere piu' livelli secondo l'itinerario. Ad esempio in fig. 3.2.2 (b) il gate  $G_1$  e' di III livello secondo l'itinerario  $G_4, G_3, G_1$  e di IV livello secondo l'itinerario  $G_4, G_3, G_2, G_1$ .

### 3.3) Analisi dei circuiti AND-OR-NOT.

L'analisi di un circuito combinatorio tende ad ottenere una rappresentazione della funzione d'uscita  $y$ , analitica o sotto la forma di tavola di verità. Poiché la rappresentazione circuitale è simbolica, l'analisi non è legata a considerazioni di logica positiva o negativa.

Per effettuare l'analisi di circuiti AND-OR-NOT è sufficiente partire dagli elementi su cui entrano le variabili e procedere verso il terminale di uscita secondo tutti i possibili itinerari, usando la funzione di uscita di ciascun elemento come variabile di ingresso dell'elemento successivo.

Si abbia da esempio il seguente circuito:

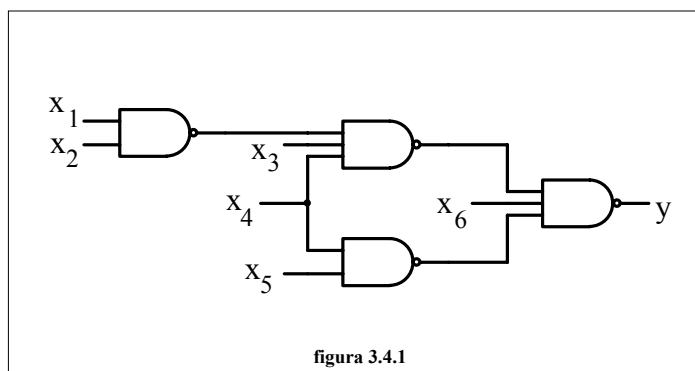


Si ha:

$$\begin{aligned}
 Y_A &= \overline{x_1 \cdot x_2 \cdot x_3} \\
 Y_{A1} &= \overline{x_1 \cdot x_2 \cdot x_3} = \overline{x_1} + \overline{x_2} + \overline{x_3} \\
 Y_{A2} &= \overline{x_4} \\
 Y_B &= x_4 \cdot Y_A = x_4 \cdot \overline{x_1 \cdot x_2 \cdot x_3} \\
 Y_C &= Y_{A1} \cdot Y_{A2} = \overline{x_4} \cdot (\overline{x_1} + \overline{x_2} + \overline{x_3}) \\
 y &= Y_B + Y_C + x_5 = x_4 \cdot \overline{x_1 \cdot x_2 \cdot x_3} + \overline{x_4} \cdot (\overline{x_1} + \overline{x_2} + \overline{x_3}) + x_5
 \end{aligned}$$

### 3.4) Analisi dei circuiti NAND.

L'analisi dei circuiti realizzati con porte NAND può essere condotta nella stessa maniera in cui si analizza un circuito AND-OR-NOT. Le espressioni che se ne ricavano sono tuttavia notevolmente complesse a causa della non associatività dell'operatore NAND. Ad esempio per il circuito di fig. 3.4.1 si ottiene l'espressione analitica:

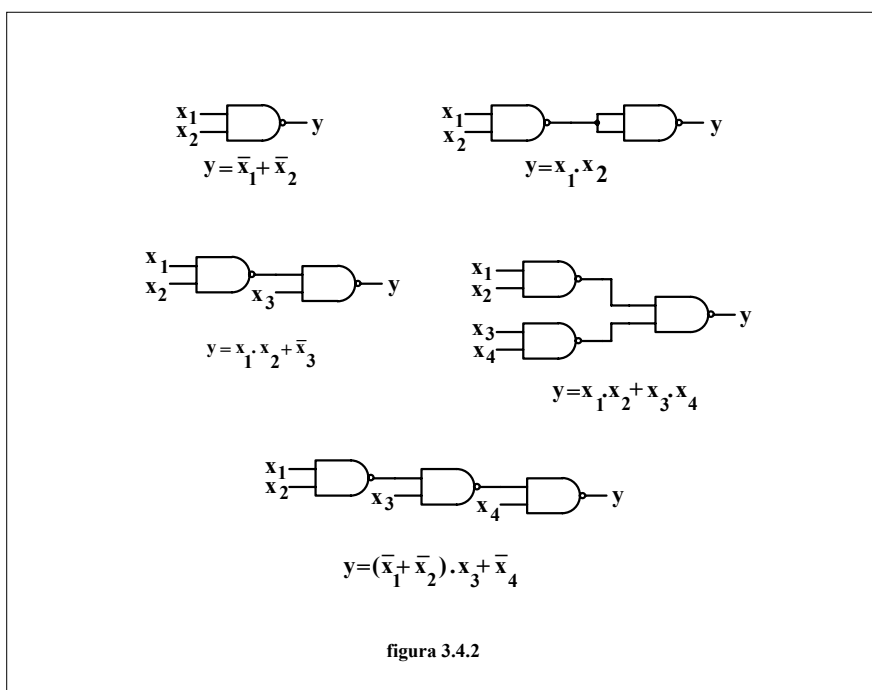


$$y = [(x_1/x_2)/x_3/x_4]/(x_4/x_5)/x_6$$

Applicando ripetutamente il teorema di De Morgan si puo' passare alla forma AND-OR-NOT.

$$\begin{aligned} y &= \overline{\overline{x_1 \cdot x_2 \cdot x_3 \cdot x_4 \cdot x_4 \cdot x_5 \cdot x_6}} = \overline{(\overline{x_1 + x_2})x_3 \cdot x_4 \cdot (\overline{x_4 + x_5})x_6} = \\ &= (\overline{x_1 + x_2})x_3 \cdot x_4 + x_4 \cdot x_5 + \overline{x_6} \end{aligned}$$

E' possibile pero' ricavare la y, nella forma di somma di prodotti, direttamente dall'esame del circuito, quando si tengano presenti le proprieta' dell'operatore NAND. Si esaminino i seguenti semplici circuiti (figura 3.4.2):



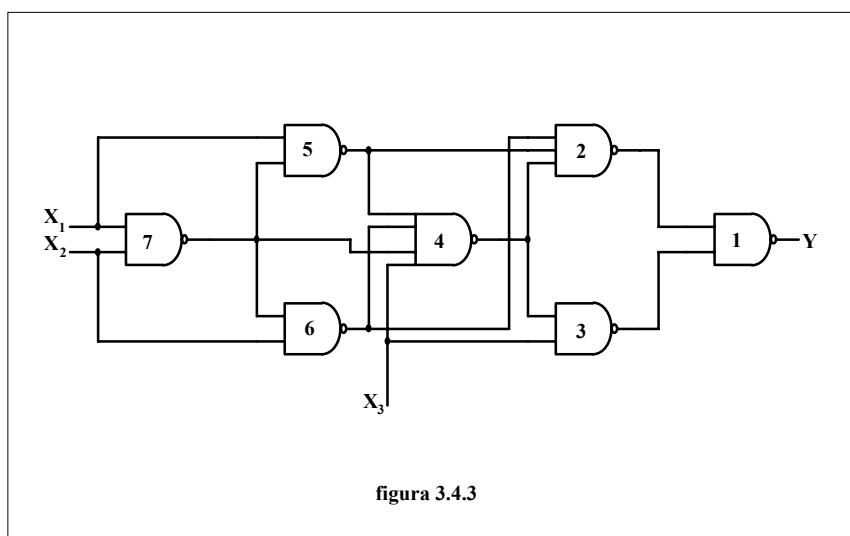
Dall'esame di questi circuiti e della loro funzione di uscita si deduce che in un circuito NAND, comunque complesso, la funzione di uscita, nella forma di somma di prodotti, e' la somma di tutte le variabili che entrano al primo livello, negate, piu' tanti termini quanti sono

gli itinerari  $I_{12}$  verso gli elementi del secondo livello; ogni itinerario  $I_{12}$  da' origine al prodotto di tutte le variabili che entrano al secondo livello e di tanti fattori quanti sono gli itinerari  $I_{23}$  verso il terzo livello; ogni itinerario  $I_{23}$  origina la somma di tutte le variabili che entrano al terzo livello, negate, piu' tanti termini quanti sono gli itinerari verso il quarto livello, e cosi' via.

In definitiva:

- 1) I livelli dispari originano somme.
- 2) I livelli pari originano prodotti.
- 3) Le variabili che entrano ai livelli dispari vanno negate.

A titolo di esempio si analizzi il circuito di fig. 3.4.3 secondo le regole teste' enunciate.



Poiche' esistono due itinerari  $I_{12}, I_{13}$  la funzione di uscita sara' la somma di due termini. Di questi il primo, proveniente dal gate 2, sara' il prodotto di  $y_6, y_5$  e  $y_4$ . Il termine proveniente da  $y_6$  sara' la somma di  $y_7$  e  $x_2$  (in quanto  $x_2$  entra su un livello dispari) e poiche' il gate 7 e' a livello pari esso dara' origine al prodotto delle variabili su esso entranti. Procedendo in modo analogo per tutti gli itinerari si ottiene:

$$\begin{aligned}
 y &= (\overline{x_2} + x_1 \cdot x_2)(\overline{x_1} + x_1 \cdot x_2)(\overline{x_3} + x_1 \cdot x_2 + x_1 \cdot (\overline{x_1} + \overline{x_2}) + x_2 \cdot (\overline{x_1} + \overline{x_2})) + \\
 &\quad + x_3 \cdot (\overline{x_3} + x_1 \cdot x_2 + \overline{x_1} \cdot x_2 + x_1 \cdot \overline{x_2}) = \\
 &= (\overline{x_2} \cdot x_1)(\overline{x_1} \cdot x_2)(\overline{x_3} + x_1 \cdot x_2 + \overline{x_1} \cdot x_2 + x_1 \cdot \overline{x_2}) + \\
 &\quad + x_3 \cdot (\overline{x_3} + x_1 \cdot x_2 + \overline{x_1} \cdot x_2 + x_1 \cdot \overline{x_2}) = \\
 &= [(\overline{x_2} + x_1)(\overline{x_1} + x_2) + x_3](\overline{x_3} + x_1 \cdot x_2 + \overline{x_1} \cdot x_2 + x_1 \cdot \overline{x_2}) = \\
 &= (\overline{x_3} + x_1 \cdot x_2 + \overline{x_1} \cdot x_2)(\overline{x_3} + x_1 + x_2) = \\
 &= \overline{x_1} \cdot \overline{x_2} \cdot \overline{x_3} + x_1 \cdot \overline{x_2} \cdot \overline{x_3} + x_1 \cdot x_3 + x_1 \cdot x_2 + x_1 \cdot x_3 = \\
 &= \overline{x_1} \cdot \overline{x_2} \cdot \overline{x_3} + x_1 \cdot x_2 + x_1 \cdot x_3 + x_2 \cdot x_3
 \end{aligned}$$

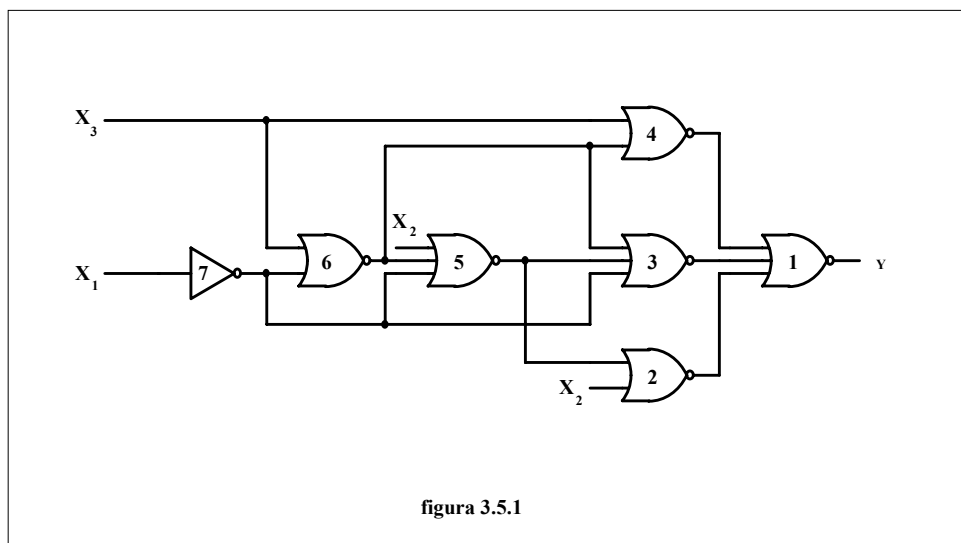
### 3.5) Analisi dei circuiti NOR.

Anche per i circuiti NOR l'analisi puo' essere condotta in analogia a quanto si fa per i circuiti AND-OR-NOT, oppure la funzione di uscita puo' essere ricavata direttamente dal circuito nella forma di prodotto di somme, sfruttando le proprieta' dell'operatore NOR.

Le regole da seguire per l'analisi diretta sono del tutto analoghe a quelle ottenute per l'operatore NAND, e' cioe':

- 1) I livelli dispari danno origine a prodotti.
- 2) I livelli pari danno origine a somme.
- 3) Le variabili che entrano su livelli dispari vanno negate.

A titolo di esempio si esamini il circuito di fig. 3.5.1. Si ottiene:



$$\begin{aligned}
 y &= (x_2 + \overline{x_2} \cdot x_1 \cdot (\overline{x_1} + x_3)) (\overline{x_1} + x_1 \cdot \overline{x_3} + \overline{x_2} \cdot x_1 \cdot (\overline{x_1} + x_3)) (x_3 + x_1 \cdot \overline{x_3}) = \\
 &= (x_2 + x_1 \cdot x_3) (\overline{x_1} + \overline{x_2} + \overline{x_3}) (x_1 + x_3) = \\
 &= (x_1 \cdot x_2 + x_1 \cdot x_3 + x_2 \cdot x_3) (\overline{x_1} + \overline{x_2} + \overline{x_3}) = \\
 &= x_1 \cdot x_2 \cdot \overline{x_3} + x_1 \cdot \overline{x_2} \cdot x_3 + \overline{x_1} \cdot x_2 \cdot x_3
 \end{aligned}$$

### 3.6) La sintesi dei circuiti combinatori.

Eseguire la sintesi di un circuito combinatorio consiste, come gia' accennato, nel progettare un circuito a n ingressi che soddisfi una determinata funzione di uscita y.

La funzione y che il circuito deve soddisfare puo' essere assegnata in diverse forme. Piu' precisamente:

- 1) Con la descrizione a parole del funzionamento del circuito. E' questa la forma di assegnazione piu' comune, ma anche la piu' imprecisa. E' necessario porre un'estrema attenzione per interpretare correttamente eventuali condizioni implicite e l'esistenza di vincoli di qualsiasi natura. Dalla descrizione verbale e' necessario passare alla tavola di verita' assegnando il valore della funzione per ognuna delle 2n configurazioni degli ingressi.

- 2) Con una vera e propria tavola di verita', che e' in definitiva l'effettivo punto di partenza della sintesi e a cui tutti gli altri tipi di assegnazione devono essere ricondotti.
- 3) Con un'espressione analitica, che e' il modo piu' conciso, anche se non univoco, di descrivere il funzionamento di un circuito.
- 4) Con uno schema logico, procedura generalmente usata quando un determinato circuito logico debba venire riprogettato con componenti diversi. In tal caso, con le regole dell'analisi, si ricava un'espressione analitica della funzione  $y$ .

Qualunque sia il metodo di assegnazione la sintesi procede partendo dalla tavola di verita' o da un'espressione analitica e applicando i metodi di semplificazione delle funzioni logiche fino a giungere alla forma piu' conveniente per gli scopi che ci si propone.

Si noti che non sempre la forma piu' conveniente corrisponde alla forma minima della funzione. Ad esempio la forma minima algebrica non sempre si puo' realizzare circuitalmente in quanto vi possono essere dei vincoli sul numero massimo di livelli. Infatti quanto esposto precedentemente e cioe' che nei circuiti combinatori l'uscita e', istante per istante, funzione unicamente degli ingressi, non significa che la variazione degli ingressi sia avvertita immediatamente in uscita; sta piuttosto a significare che ogni configurazione di ingresso da' luogo a una determinata uscita e che eventuali transitori di commutazione possono ritardare, ma non modificare quest'uscita.

Ora il tempo di commutazione di qualsiasi elemento fisico, per quanto piccolo, non e' mai nullo; il tempo di risposta di un circuito a  $n$  livelli al variare della configurazione di ingresso e'  $n \cdot \Delta$ , avendo indicato con  $\Delta$  il tempo di commutazione del singolo componente.

In definitiva il ritardo totale tra ingresso e uscita e' proporzionale al numero di livelli e potendo la forma minima di una funzione contenere un numero di livelli molto elevato, la sua diretta realizzazione circuitale potrebbe dar luogo a ritardi intollerabili.

La forma in cui si ha il minimo ritardo e' quella a due livelli, che d'altra parte e' quella che si ottiene con i metodi di semplificazione che sono stati esposti al capitolo II. La convenienza di eventuali fattorizzazioni va valutata caso per caso.

Si puo' concludere percio' che la sintesi di un circuito combinatorio procede attraverso i seguenti passi, di cui quello (5) non puo' venir condotto secondo un procedimento sistematico, ma andra' verificato di volta in volta.

- 1) Descrizione del funzionamento del circuito.**
- 2) Determinazione della tavola di verita'.**
- 3) Sintesi della funzione di commutazione.**
- 4) Semplificazione della funzione logica relativa.**
- 5) Determinazione della forma minima piu' conveniente.**
- 6) Progetto del circuito.**

### 3.7) Sintesi di circuiti AND-OR-NOT.

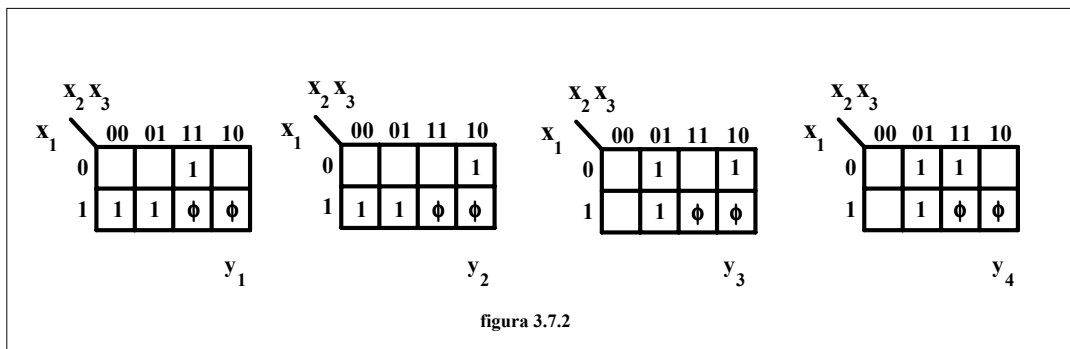
Il procedimento e' banale; consiste nel ricavare, a partire dalla tavola di verita', la forma minima a due livelli; passare poi da questa alla forma minima piu' conveniente, eventualmente con tecniche basate sul concetto di decomponibilita', che saranno illustrate piu' avanti. Infine si disegna il circuito.

Si voglia ad esempio realizzare un circuito a tre ingressi, sui quali possa presentarsi un numero binario compreso tra 0 e 5. All'uscita di tale circuito debba essere realizzato il prodotto per 3 del numero di ingresso.

$x_1$	$x_2$	$x_3$	$y_1$	$y_2$	$y_3$	$y_4$
0	0	0	0	0	0	0
0	0	1	0	0	1	1
0	1	0	0	1	1	0
0	1	1	1	0	0	1
1	0	0	1	1	0	0
1	0	1	1	1	1	1
1	1	0	-	-	-	-
1	1	1	-	-	-	-

figura 3.7.1

Poiche' il massimo numero di uscita e' 15, rappresentabile con 4 bit, sara' necessario sintetizzare quattro funzioni logiche. Le relative tavole di verita' sono riportate in fig. 3.7.1 mentre in fig. 3.7.2 si hanno le corrispondenti mappe di Karnaugh.

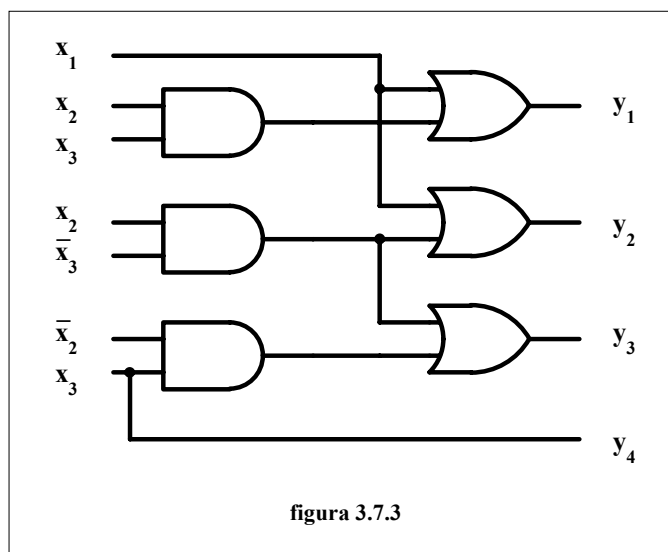


Utilizzando opportunamente le condizioni non specificate si ottiene:

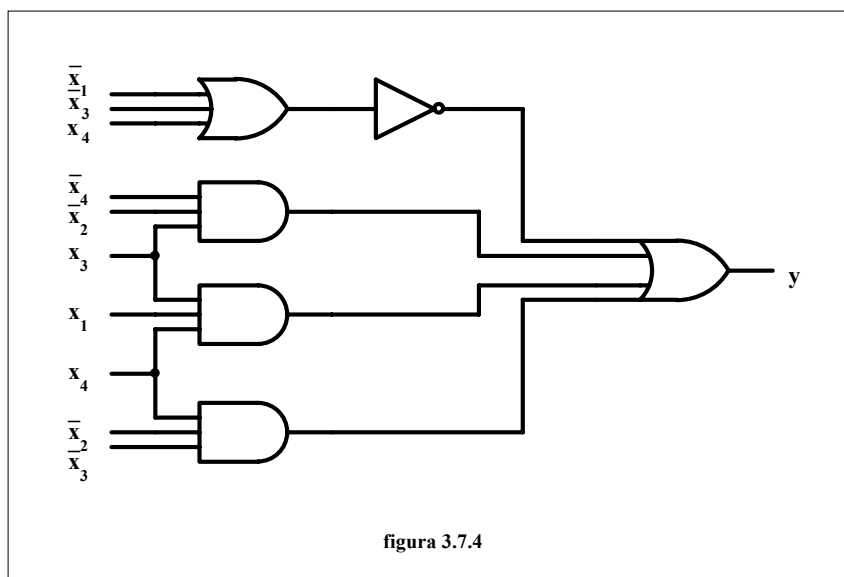
$$\begin{aligned}
 y_1 &= x_1 + x_2 \cdot \overline{x_3} \\
 y_2 &= \overline{x_1} + x_2 \cdot \overline{x_3} \\
 y_3 &= \overline{x_2} \cdot x_3 + x_2 \cdot \overline{x_3} \\
 y_4 &= x_3
 \end{aligned}$$

cui corrisponde il circuito di fig. 3.7.3, ottenuto mettendo in comune tra le funzioni  $y_2$  e  $y_3$  il termine  $\overline{x_2} \cdot x_3$ .





Quale altro esempio si voglia sintetizzare un circuito con la stessa funzione di trasmissione di quello illustrato in fig. 3.7.4, ma possibilmente più economico.



L'espressione analitica della funzione di trasmissione è

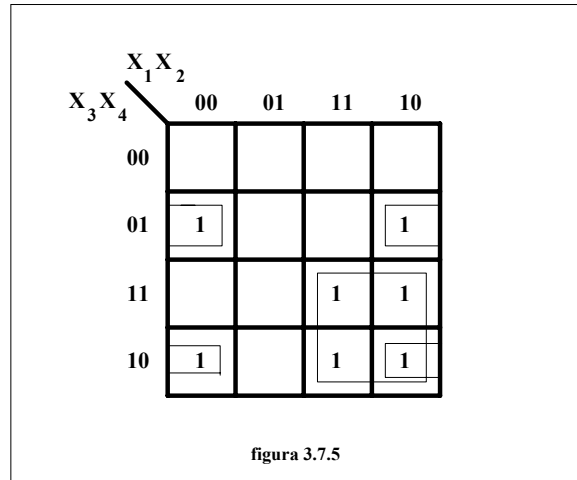
$$\begin{aligned}
 y &= \overline{x_1} + \overline{x_3} + \overline{x_4} + \overline{x_2} \cdot \overline{x_3} \cdot \overline{x_4} + x_1 \cdot \overline{x_3} \cdot \overline{x_4} + \overline{x_2} \cdot \overline{x_3} \cdot \overline{x_4} = \\
 &= x_1 \cdot \overline{x_3} \cdot \overline{x_4} + \overline{x_2} \cdot \overline{x_3} \cdot \overline{x_4} + x_1 \cdot \overline{x_3} \cdot \overline{x_4} + \overline{x_2} \cdot \overline{x_3} \cdot \overline{x_4}
 \end{aligned}$$

Dalla mappa di Karnaugh di fig. 3.7.5 si ricava l'espressione minima a due livelli

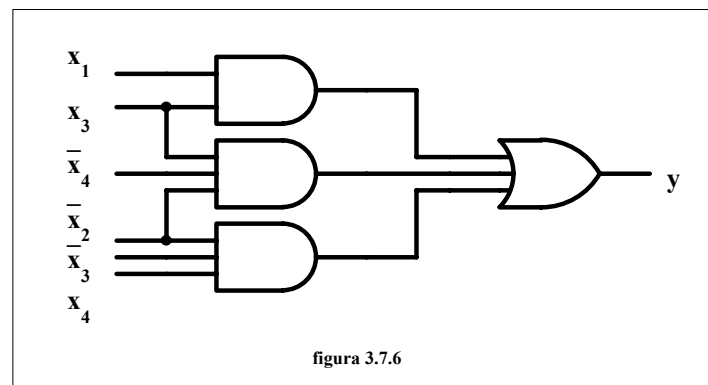
$$y = x_1 \cdot \overline{x_3} + \overline{x_2} \cdot \overline{x_3} \cdot \overline{x_4} + \overline{x_2} \cdot \overline{x_3} \cdot \overline{x_4} \quad (3.7.1)$$

e fattorizzando si ottiene:

$$y = x_1 \cdot x_3 + \overline{x_2} \cdot (x_3 \cdot \overline{x_4} + \overline{x_3} \cdot x_4) \quad (3.7.2)$$



Per la realizzazione della (3.7.1) sono sufficienti 4 gate, mentre per la (3.7.2) ne sono necessari 6. Il circuito realizzato e' allora quello relativo all'espressione (3.7.1) ed e' riportato in fig. 3.7.6



Si noti che negli esempi fatti si e' supposto di avere a disposizione all'ingresso del circuito sia le variabili dirette che la loro negazione. Quando questa situazione non si verifica anche il numero di invertitori va minimizzato. Ad esempio la funzione

$$y = \overline{x_1} + \overline{x_2} + \overline{x_3} + \overline{x_4}$$

realizzabile con quattro invertitori e un gate OR, puo' essere realizzata molto piu' convenientemente tenendo presente che:

$$y = \overline{x_1} + \overline{x_2} + \overline{x_3} + \overline{x_4} = \overline{x_1 \cdot x_2 \cdot x_3 \cdot x_4}$$

richiedendo in tal caso solamente un AND e un NOT.

A causa della proprieta' associativa di somma e prodotto, quando qualcuno dei gate cui si perviene nel corso del progetto ha un numero di ingressi eccessivamente elevato, lo si puo' spezzare in due o piu' elementi dello stesso tipo, aumentando pero' il numero di livelli.

### 3.8) La decomposizione in sconessione semplice.

La semplificazione dei circuiti mediante la fattorizzazione e' una tecnica soddisfacente nella maggior parte dei casi, ma e' un processo di tipo empirico largamente dipendente dall'esperienza e dal colpo d'occhio di chi opera. Sarebbe conveniente avere a disposizione qualche metodo piu' sistematico; cio' si puo' ottenere mediante la decomposizione funzionale.

Il metodo di semplificazione in sostanza consiste nello spezzare una singola funzione complessa in un certo numero di funzioni semplici ed e' stato proposto da Ashenhurst e Curtis.

Sia  $x_1, x_2, \dots, x_n$  un insieme di variabili di commutazione e siano A e B due sottoinsiemi disgiunti di X, tali che:

$$A \cup B = X$$

$$A \cap B = \Phi$$

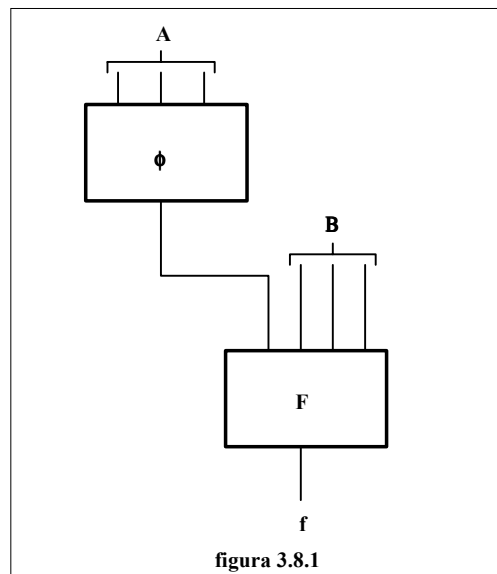
Sia assegnata una funzione booleana:

$$f(x_1, x_2, \dots, x_n)$$

Se e' possibile individuare due funzioni F e  $\phi$ , tali che:

$$f(X) = F[\phi(A), B]$$

allora si dice che F e  $\phi$  formano una decomposizione in sconessione semplice della funzione f. L'insieme delle variabili A e' detto **insieme delle variabili al contorno**, mentre l'insieme B e' quello delle **variabili indipendenti**. Il significato circuitale della decomposizione in sconessione semplice e' illustrato in fig. 3.8.1.



Circuitalmente una funzione logica si realizza usando componenti capaci di assumere l'uno o l'altro di due stati diversi. Nei circuiti elettronici questi due stati sono realizzati mediante due livelli caratteristici di tensione, detti livello alto (**h**) e livello basso (**b**).

Si consideri ad esempio la seguente funzione:

$$f(x_1, x_2, x_3, x_4) = \overline{x_1} \cdot \overline{x_2} \cdot x_3 \cdot x_4 + x_1 \cdot \overline{x_2} \cdot x_3 \cdot \overline{x_4} + \overline{x_1} \cdot \overline{x_2} \cdot \overline{x_3} \cdot \overline{x_4} + \overline{x_1} \cdot x_2 \cdot \overline{x_3} \cdot x_4$$

Manipolando opportunamente tale funzione si ottiene:

$$\begin{aligned} f(x_1, x_2, x_3, x_4) &= x_1 \cdot x_3 \cdot (\overline{x_2} \cdot x_4 + x_2 \cdot \overline{x_4}) + \overline{x_1} \cdot \overline{x_3} \cdot (\overline{x_2} \cdot \overline{x_4} + x_2 \cdot x_4) = \\ &= x_1 \cdot x_3 \cdot \varphi(x_2, x_4) + \overline{x_1} \cdot \overline{x_3} \cdot \overline{\varphi}(x_2, x_4) = \\ &= F[\varphi(x_2, x_4), x_1, x_3] \end{aligned}$$

con:

$$\varphi(x_2, x_4) = \overline{x_2} \cdot x_4 + x_2 \cdot \overline{x_4}$$

che corrisponde alla partizione  $x_1, x_3 / x_2, x_4$ .

Si noti che nell'esempio fatto si e' semplicemente verificato che e' possibile ottenere una sconnessione semplice nella forma  $x_1, x_3 / x_2, x_4$ . L'obiettivo che ci si prefigge e' invece individuare un metodo generale che permetta di determinare se esiste o meno una particolare decomposizione.

Si consideri ad esempio la seguente funzione

$$f(x_1, x_2, x_3, x_4) = \sum (4, 5, 6, 7, 8, 13, 14, 15)_m$$

che e' decomponibile nella forma  $F[\varphi(x_1, x_4), x_2, x_3]$ .

Per prima cosa si raccolgano i termini minimi secondo le quattro combinazioni possibili di  $x_2$  e  $x_3$  in modo da ottenere la forma;

$$\begin{aligned} f(x_1, x_2, x_3, x_4) &= \overline{x_2} \cdot \overline{x_3} \cdot \alpha(x_1, x_4) + \overline{x_2} \cdot x_3 \cdot \beta(x_1, x_4) + x_2 \cdot \overline{x_3} \cdot \gamma(x_1, x_4) + \\ &+ x_2 \cdot x_3 \cdot \delta(x_1, x_4) \end{aligned} \quad (3.8.1)$$

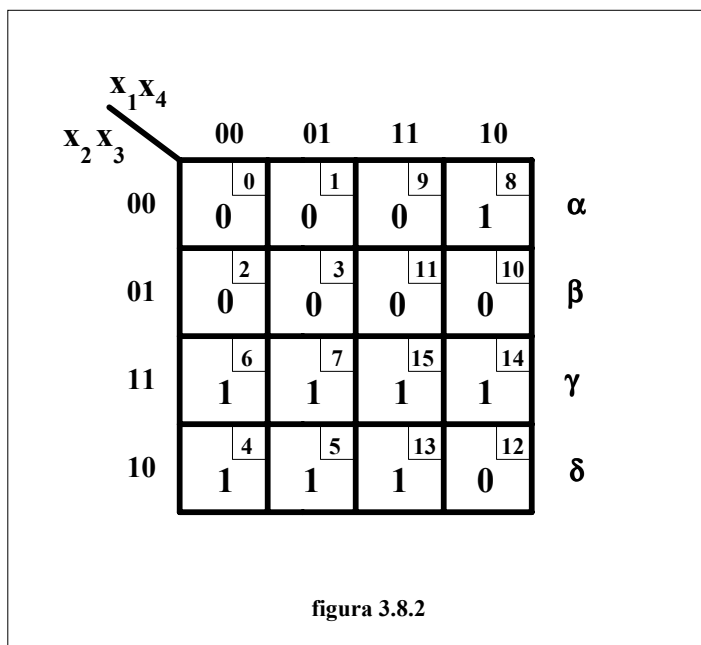
La cosa e' evidentemente possibile per qualsiasi funzione, sia essa decomponibile o meno.

Si costruisca poi una speciale mappa della funzione, detta mappa di partizione (figura 3.8.2); essa e' semplicemente una mappa di Karnaugh in cui le variabili indipendenti sono sistemate sulla verticale, mentre le variabili al contorno tarano l'asse orizzontale.

Confrontando la mappa di partizione con l'equazione (3.8.1) si vede che la prima riga corrisponde alla funzione  $\alpha(x_1, x_4)$ , la seconda a  $\beta(x_1, x_4)$ , la terza a  $\gamma(x_1, x_4)$  e infine la quarta a  $\delta(x_1, x_4)$ . Anche questa e' evidentemente un'osservazione del tutto generale, che si applica sia alle funzioni decomponibili che a quelle non decomponibili.

Si osservi ora che, come affermato in precedenza, per avere la decomposizione in sconnessione semplice la funzione  $F$  deve essere funzione di  $x_2, x_3$  e di una sola funzione

$\varphi(x_1, x_4)$ . Si supponga che la  $\varphi(x_1, x_4)$  sia la  $\alpha$ . Allora, affinché la funzione  $f$  sia decomponibile le rimanenti righe devono rappresentare o  $\varphi$  o  $\bar{\varphi}$  o la costante logica 1 o la costante logica 0. In altre parole le righe successive alla prima devono essere o identiche a  $\varphi$  o al suo complemento oppure devono essere riempite totalmente con 1 o 0.



Questa condizione nell'esempio fatto è rispettata, in quanto:

$$\begin{aligned}
 f(x_1, x_2, x_3, x_4) &= \overline{x_2} \cdot \overline{x_3} \cdot \varphi(x_1, x_4) + x_2 \cdot \overline{x_3} \cdot \overline{\varphi}(x_1, x_4) + x_2 \cdot x_3 = \\
 &= F(\varphi(x_1, x_4), x_2, x_3)
 \end{aligned}$$

con:

$$\varphi(x_1, x_4) = x_1 \cdot \overline{x_4}$$

Il criterio descritto permette di discriminare se una funzione è decomponibile, ma si applica con difficoltà notevolissime in mappe di partizione di dimensioni notevoli.

Si definisca allora molteplicità di colonna il numero di differenti insiemi di 1 e 0 che si possono individuare esaminando le colonne della mappa di partizione. Ad esempio, nella mappa di fig. 3.8.2 si ha una molteplicità di colonna 2, in quanto si possono individuare unicamente i due insiemi:

$$\begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix} \quad \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

Il seguente teorema offre allora un semplice metodo per determinare se una mappa di partizione corrisponde ad una decomposizione in sconnessione semplice.

**TEOREMA:** Una mappa di partizione corrisponde ad una decomposizione in sconnessione semplice se e solo se la molteplicita' di colonna e' minore o uguale a 2.

Infatti, come gia' e' stato detto, se una funzione e' scomponibile, i valori funzionali di ciascuna riga della mappa di partizione devono essere 0 o 1 o  $\varphi$  o  $\bar{\varphi}$ . Le righe che corrispondono alle costanti 0 e 1 contengono gli stessi valori in ciascuna colonna e quindi non contribuiscono alla molteplicita'. D'altra parte se una colonna ha un 1 in una riga  $\varphi$  allora deve contenere uno 0 nelle righe  $\bar{\varphi}$  e un 1 nelle altre righe  $\varphi$ . Sono quindi possibili solo due distinti tipi di colonna.

In modo simile si puo' dedurre che, se si assume una molteplicita' di colonna 2, si e' in presenza di una decomposizione in sconnessione semplice.

Comunque sia, il lavoro da eseguire rimane notevole, poiche' e' necessario investigare su tutte le possibili partizioni delle variabili in variabili al contorno e variabili indipendenti.

Si ricorre allora all'uso della carta di decomposizione, che altro non e' se non l'insieme delle mappe di Karnaugh su cui sono riportate le posizioni dei termini minimi per ciascuna possibile combinazione di variabili indipendenti e al contorno.

In fig. 3.8.3 e' riportata la carta di decomposizione per funzioni di quattro variabili. Per usare la carta si cerchi i termini minimi delle funzione su tutte le mappe e si individuano quelle in cui la molteplicita' di colonna e' 2 o 1. Si noti che le carte di decomposizione di dimensione 4x4 vanno utilizzate sia direttamente che trasposte. Ad esempio, la prima delle tre mappe di fig. 3.8.3 e' rappresentativa sia della decomposizione  $F(\varphi(x_1, x_2), x_3, x_4)$ , quando usata in senso diretto, che della decomposizione  $F(\varphi(x_3, x_4), x_1, x_2)$ , quando usata trasposta.

Le carte di dimensione 2x8 vanno invece esaminate solo direttamente poiche' la decomposizione secondo una sola variabile al contorno e' una decomposizione banale.

Si esamini ad esempio la funzione:

$$f(x_1, x_2, x_3, x_4) = \sum (4, 5, 6, 7, 8, 13, 14, 15)_m$$

Dall'esame della carta di decomposizione si ottengono le seguenti possibili partizioni:

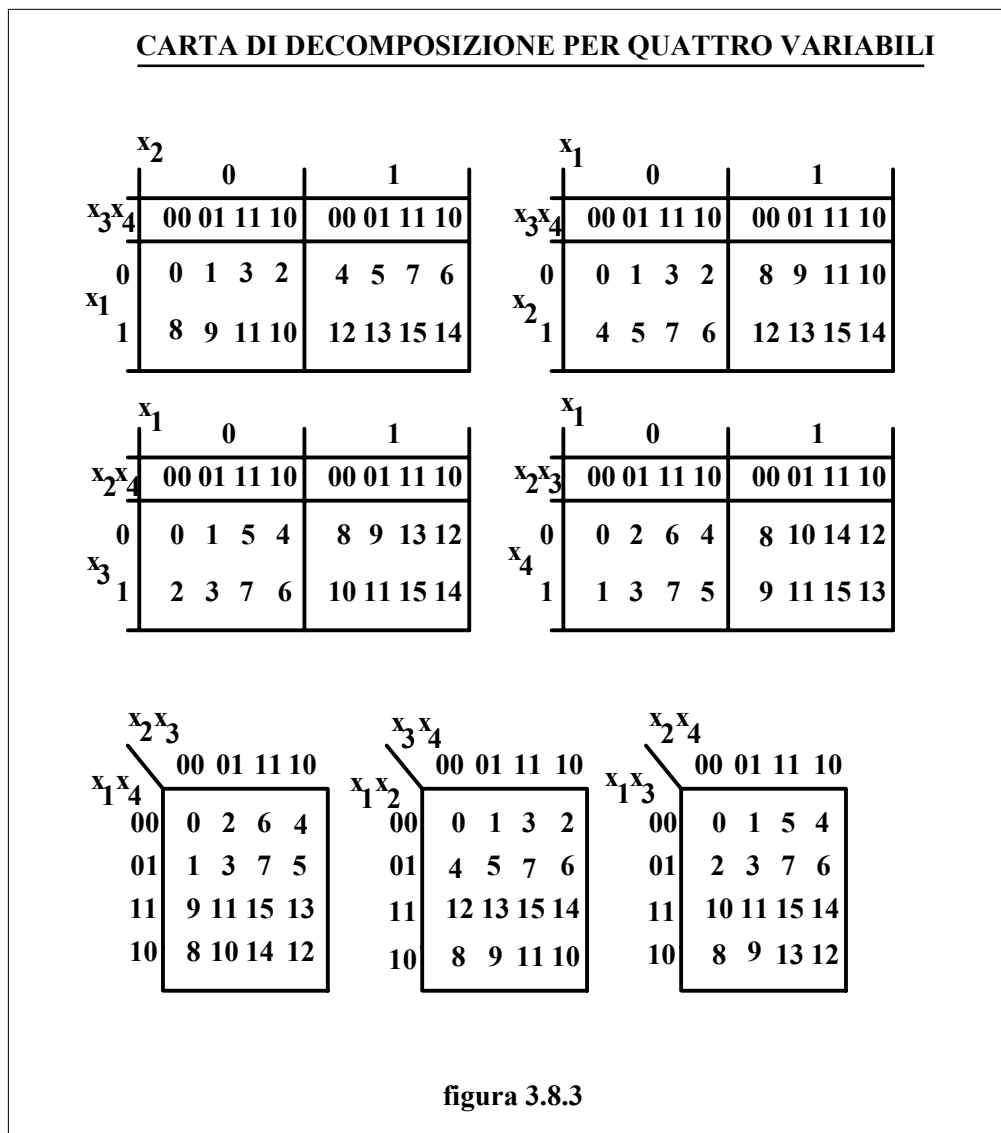
$$\begin{aligned} f(x_2/x_1, x_3, x_4) &= \bar{x}_2 \cdot (\bar{x}_1 \cdot \bar{x}_3 \cdot \bar{x}_4) + x_2 \cdot (\bar{x}_1 \cdot \bar{x}_3 \cdot \bar{x}_4) = \\ &= \bar{x}_2 \cdot \varphi(x_1, x_3, x_4) + x_2 \cdot \bar{\varphi}(x_1, x_3, x_4) \end{aligned}$$

$$\begin{aligned} f(x_2, x_3/x_1, x_4) &= x_2 \cdot x_3 + \bar{x}_2 \cdot \bar{x}_3 \cdot (\bar{x}_1 + x_4) + x_2 \cdot \bar{x}_3 \cdot (\bar{x}_1 + x_4) = \\ &= x_2 \cdot x_3 + \bar{x}_2 \cdot \bar{x}_3 \cdot \bar{\lambda}(x_1, x_4) + x_2 \cdot \bar{x}_3 \cdot \lambda(x_1, x_4) \end{aligned}$$

$$\begin{aligned} f(x_1, x_2/x_3, x_4) &= \bar{x}_1 \cdot x_2 + x_1 \cdot \bar{x}_2 \cdot (\bar{x}_3 + x_4) + x_1 \cdot x_2 \cdot (\bar{x}_3 + x_4) = \\ &= \bar{x}_1 \cdot x_2 + x_1 \cdot \bar{x}_2 \cdot \bar{\psi}(x_3, x_4) + x_1 \cdot x_2 \cdot \psi(x_3, x_4) \end{aligned}$$

$$f(x_2, x_4/x_1, x_3) = x_2 \cdot x_4 + x_2 \cdot \overline{x_4} \cdot (\overline{x_1} + x_3) + \overline{x_2} \cdot x_4 \cdot (\overline{x_1} + x_3) =$$

$$= x_2 \cdot x_4 + x_2 \cdot \overline{x_4} \cdot \rho(x_1, x_3) + \overline{x_2} \cdot x_4 \cdot \rho(x_1, x_3)$$



La procedura descritta puo' essere applicata a funzioni di un maggior numero di variabili ed in teoria essere estesa a funzioni di qualsiasi numero di variabili. In pratica per un numero di variabili superiore a 5 la dimensione della carta di decomposizione diventa eccessiva. Tuttavia la generalita' del metodo permette di pensare ad una computerizzazione del processo quando il numero delle variabili fosse superiore a 5.

Qualora esistano condizioni non specificate nella funzione assegnata e' necessario contrassegnare opportunamente sulla carta di decomposizione (ad esempio barrandoli) i relativi termini minimi. Si utilizzano poi le condizioni non specificate in modo da ottenere, se possibile, una molteplicita' di colonna non superiore a 2.

### 3.9) Altre decomposizioni disgiuntive.

E' evidente che esistono altre forme di decomposizione diverse che non quella in sconnessione semplice. In questo paragrafo verranno prese in esame le decomposizioni che si ottengono con la partizione dell'insieme delle variabili indipendenti in tre o piu' sottoinsiemi disgiunti.

Sia  $X = x_1, x_2, \dots, x_n$  un insieme di  $n$  variabili di commutazione e siano  $A_1, A_2, \dots, A_m$  dei sottoinsiemi disgiunti di  $X$ , tali che:

$$A_1 \cup A_2 \cup \dots \cup A_m = X$$

La decomposizione, ove possibile, in una delle due forme:

$$f(X) = F[\psi_1(A_1), \psi_2(A_2), \dots, \psi_m(A_m)]$$

oppure

$$f(X) = F[\psi_1(A_1), \psi_2(A_2), \dots, \psi_{m-1}(A_{m-1}), A_m]$$

prende il nome di decomposizione in sconnessione multipla.

La decomposizione nella forma:

$$f(X) = F[\lambda(\psi[A], B), C]$$

o, piu' in generale:

$$f(X) = F[\lambda_{m-1}(\lambda_{m-2}[\dots \lambda_2(\lambda_1[A_1], A_2), \dots, A_{m-2}], A_{m-1}), A_m]$$

e' conosciuta come decomposizione in sconnessione iterativa.

Infine la combinazione di queste due forme, cioe':

$$f(X) = F[\gamma(\psi[A], B), \lambda(C), D]$$

e' detta decomposizione in sconnessione complessa.

In ogni caso la decomposizione sara' non banale se gli insiemi delle variabili al contorno ne conterranno almeno due.

Esiste un certo numero di teoremi che mettono in relazione le decomposizioni appena illustrate con quella in sconnessione semplice e permettono di estendere l'uso della carta di decomposizione alla loro determinazione.

Di questi teoremi, di dimostrazione spesso lunga e noiosa, ci si limitera' a dare l'enunciato ed un esempio di applicazione.

#### 3.9.1) Decomposizione iterativa.

Sia  $f(X)$  una funzione per la quale esistano due diverse decomposizioni in sconnessione semplice:



$$f(X) = F[\lambda(A, B), C] = G[\psi(A), B, C]$$

Esiste allora la decomposizione in sconessione iterativa

$$f(X) = F[\rho(\psi[A], B), C]$$

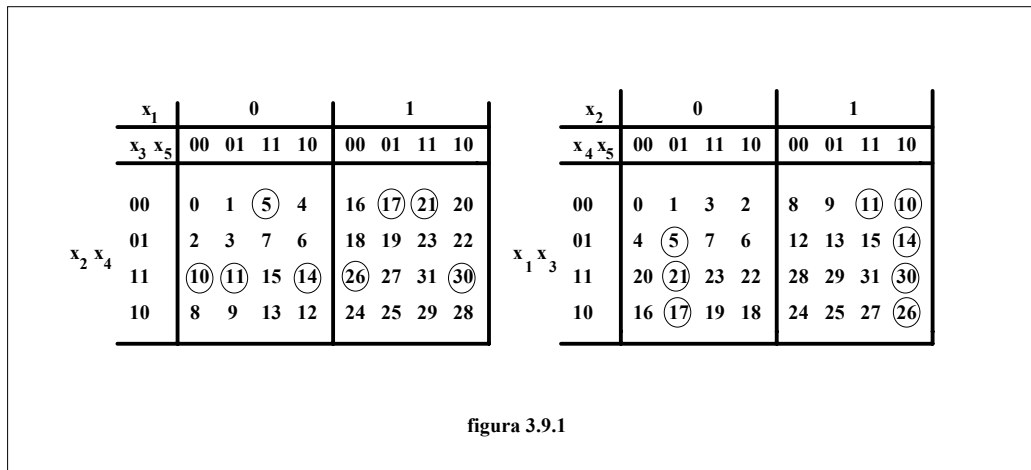
con:

$$\rho(\psi[A], B) = \lambda(A, B)$$

Quale esempio di applicazione di questo teorema si consideri la funzione:

$$f(x_1, x_2, x_3, x_4, x_5) = \sum (5, 10, 11, 14, 17, 21, 26, 30)_m$$

In fig. 3.9.1 sono riportate due mappe di decomposizione della funzione; da esse si deduce che esistono le due seguenti decomposizioni in sconessione semplice:



$$f(X) = F[\psi(x_1, x_3, x_5), x_2, x_4]$$

$$f(X) = G[\lambda(x_1, x_3), x_2, x_4, x_5]$$

Le condizioni enunciate sono soddisfatte ponendo:

$$A = (x_1, x_3) \quad B = (x_5) \quad C = (x_2, x_4)$$

Dalla prima mappa di partizione della fig. 3.9.1 si ottiene:

$$f(X) = \overline{x_2} \cdot \overline{x_4} \cdot \psi(x_1, x_3, x_5) + x_2 \cdot x_4 \cdot \overline{\psi}(x_1, x_3, x_5)$$

con:

$$\psi(x_1, x_3, x_5) = x_1 \cdot x_5 + x_3 \cdot x_5$$

Ora, dalle condizioni enunciate all'inizio di questo paragrafo, la funzione  $\psi(x_1, x_3, x_5)$  puo' essere decomposta in accordo con la partizione  $x_1, x_3/x_5$ . Per ispezione diretta della  $\psi(x_1, x_3, x_5)$  si vede che la decomposizione e' data da:

$$\psi(x_1, x_3, x_5) = x_5 \cdot (x_1 + x_3)$$

Questo esempio e' ovviamente banale, nel senso che non ci sarebbe bisogno di ricorrere alle condizioni enunciate all'inizio per individuare la fattorizzazione eseguita sulla  $\psi(x_1, x_3, x_5)$ . Tuttavia il teorema stesso puo' essere usato, indipendentemente dal numero di variabili, per individuare, tramite la carta di decomposizione, quelle situazioni in cui le sotto-funzioni possono essere di volta in volta ulteriormente decomposte.

### 3.9.2) Decomposizione multipla.

Sia  $f(X)$  una funzione per la quale esistano due diverse decomposizioni in sconessione semplice.

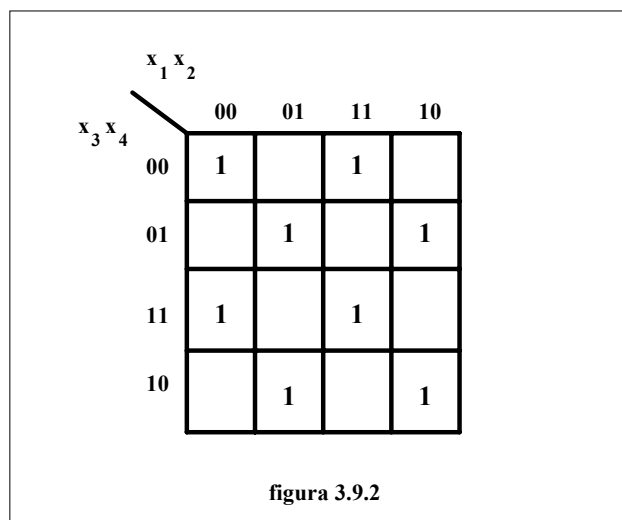
$$f(X) = F[\lambda(A), B] = G[\psi(B), A]$$

Esiste allora la decomposizione in sconessione multipla:

$$f(X) = H[\lambda(A), \psi(B)]$$

La condizione teste' espressa sara' soddisfatta quando sulla carta di partizione della  $f(X)$  la molteplicita' di colonna sara' non superiore a 2 sia per la carta diretta che per quella trasposta.

Si consideri, ad esempio, la funzione di parita' di quattro variabili, la cui mappa di Karnaugh e' riportata in fig. 3.9.2.



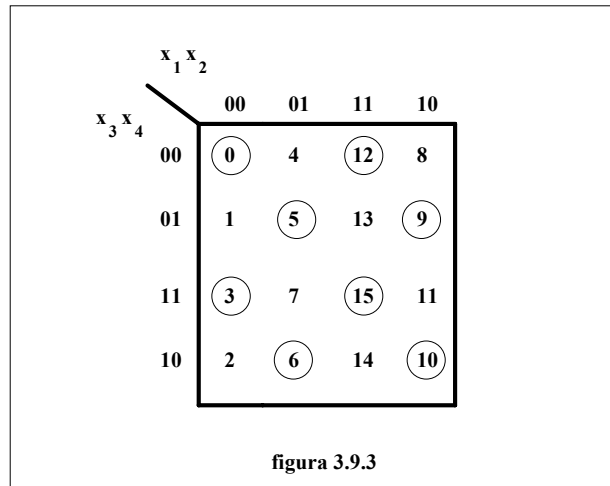
La mappa di decomposizione della stessa funzione e' illustrata in fig. 3.9.3. Si puo' notare immediatamente che la molteplicita' e' 2 sia per le colonne che per le righe.

Dalla carta trasposta si ottiene:

$$f(x_1, x_2/x_3, x_4) = (\overline{x_1} \cdot \overline{x_2} + x_1 \cdot x_2) \overline{\psi}(x_3, x_4) + (\overline{x_1} \cdot x_2 + x_1 \cdot \overline{x_2}) \psi(x_3, x_4)$$

con:

$$\psi(x_3, x_4) = \overline{x_3} \cdot x_4 + x_3 \cdot \overline{x_4}$$



Dall'esame della carta diretta si ottiene:

$$f(x_3, x_4/x_1, x_2) = (\overline{x_3} \cdot x_4 + x_3 \cdot \overline{x_4}) \lambda(x_1, x_2) + (\overline{x_3} \cdot \overline{x_4} + x_3 \cdot x_4) \overline{\lambda}(x_1, x_2)$$

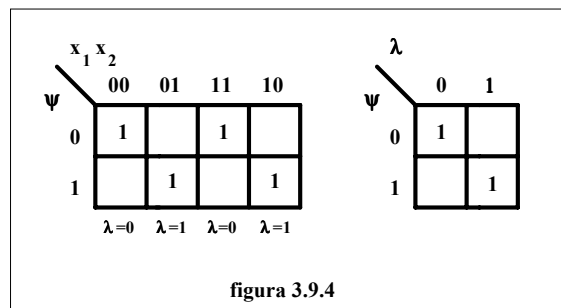
con:

$$\lambda(x_1, x_2) = \overline{x_1} \cdot x_2 + x_1 \cdot \overline{x_2}$$

Il teorema enunciato indica l'esistenza di una decomposizione multipla:

$$f(X) = H[\psi(x_3, x_4), \lambda(x_1, x_2)]$$

Per determinare H si costruisca (figura 3.9.4) la mappa di Karnaugh della  $F(\psi, x_1, x_2)$ . I valori di  $\lambda(x_1, x_2)$  sono indicati in tale mappa, permettendo quindi l'immediata costruzione della mappa della funzione  $H(\psi, \lambda)$ .



Si ottiene pertanto:

$$H(\psi, \lambda) = \overline{\psi} \cdot \overline{\lambda} + \psi \cdot \lambda$$

dove:

$$\psi = x_3 \oplus x_4$$

$$\lambda = x_1 \oplus x_2$$

Anche in questo caso l'esempio e' banale, in quanto lo stesso risultato poteva essere ottenuto per ispezione diretta. Tuttavia esso e' importante in quanto il metodo descritto si estende facilmente a funzioni con qualsiasi numero di variabili.

Una decomposizione in sconnessione multipla puo' poi essere individuata anche nel caso seguente.

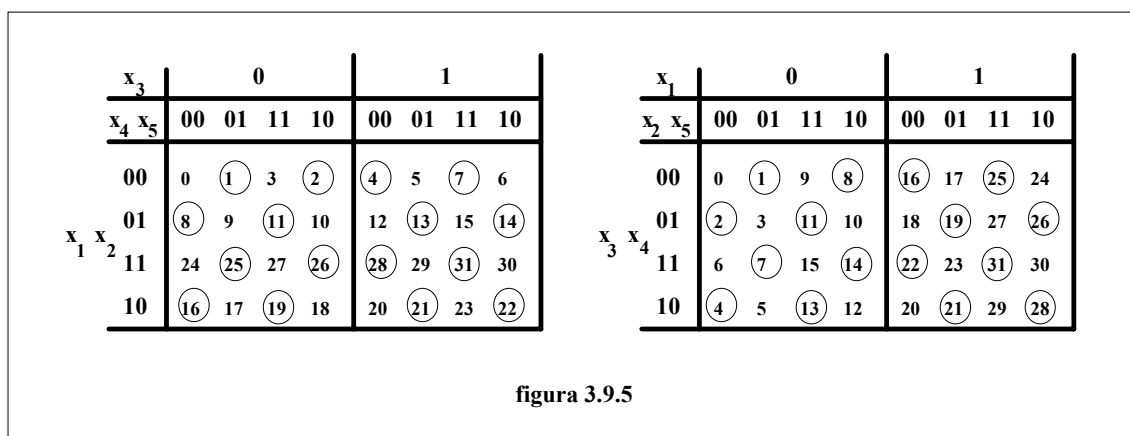
Sia  $f(X)$  una funzione per la quale esistano le due seguenti decomposizioni in sconnessione semplice:

$$f(X) = F[\lambda(A), B, C] = G[\varphi(B), A, C]$$

Esiste allora la decomposizione in sconnessione multipla

$$f(X) = H[\lambda(A), \varphi(B), C]$$

Quale esempio si consideri la funzione di disparita' di 5 variabili, di cui due mappe di partizione sono riportate in fig. 3.9.5.



Dall'esame della prima carta trasposta si ottiene la decomposizione:

$$f(x_3, x_4, x_5/x_1, x_2) = (\overline{x_3} \cdot \overline{x_4} \cdot \overline{x_5} + \overline{x_3} \cdot x_4 \cdot x_5 + x_3 \cdot \overline{x_4} \cdot x_5 + x_3 \cdot x_4 \cdot \overline{x_5}) \lambda(x_1, x_2) + (\overline{x_3} \cdot x_4 \cdot x_5 + \overline{x_3} \cdot x_4 \cdot \overline{x_5} + x_3 \cdot \overline{x_4} \cdot \overline{x_5} + x_3 \cdot x_4 \cdot x_5) \overline{\lambda}(x_1, x_2)$$

con:

$$\lambda(x_1, x_2) = \overline{x_1} \cdot x_2 + x_1 \cdot \overline{x_2}$$

Dalla seconda carta di decomposizione, sempre trasposta, si puo' individuare la decomposizione:

$$f(x_1, x_2, x_5/x_3, x_4) = (\overline{x_1} \cdot \overline{x_2} \cdot \overline{x_5} + \overline{x_1} \cdot x_2 \cdot x_5 + x_1 \cdot \overline{x_2} \cdot x_5 + x_1 \cdot x_2 \cdot \overline{x_5})\varphi(x_3, x_4) + (\overline{x_1} \cdot x_2 \cdot \overline{x_5} + \overline{x_1} \cdot \overline{x_2} \cdot x_5 + x_1 \cdot \overline{x_2} \cdot \overline{x_5} + x_1 \cdot x_2 \cdot x_5)\overline{\varphi}(x_3, x_4)$$

con:

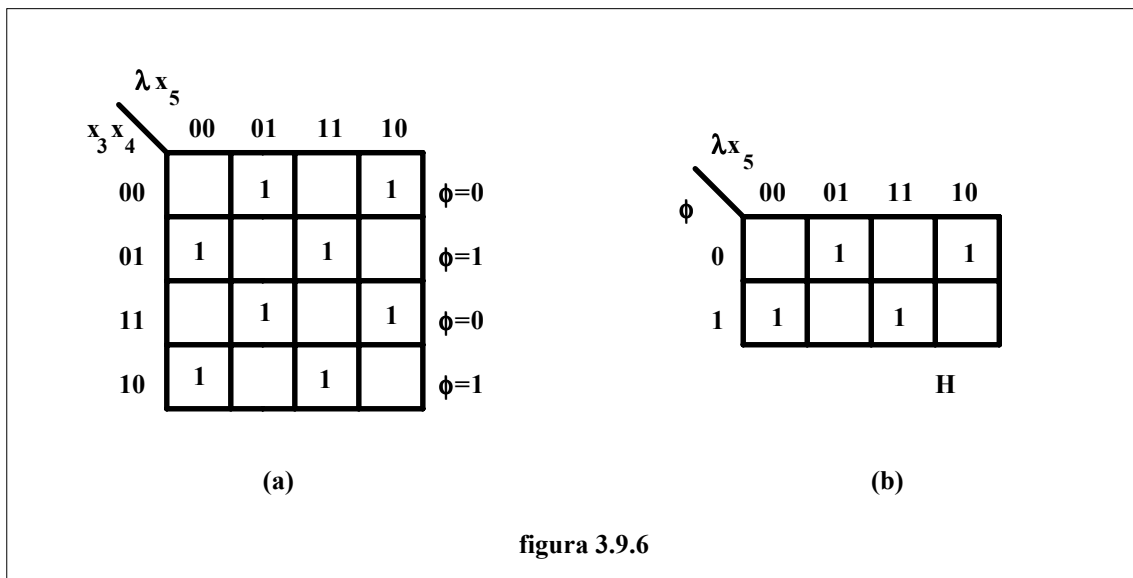
$$\varphi(x_3, x_4) = \overline{x_3} \cdot x_4 + x_3 \cdot \overline{x_4}$$

Le condizioni enunciate in precedenza sono soddisfatte con :

$$A = (x_1, x_2) \quad B = (x_3, x_4) \quad C = (x_5)$$

Per determinare la funzione composta H, si costruisca la mappa di Karnaugh della funzione  $f(x_3, x_4, x_5/x_1, x_2)$  indicando per ciascuna delle sue righe i valori assunti dalla funzione  $\varphi$ , come illustrato in fig. 3.9.6 (a).

Si puo' allora, a partire da tale mappa, ricostruire la mappa della funzione H, come viene illustrato in fig. 3.9.6 (b).



Da tale mappa si ottiene:

$$H(\lambda, \varphi, x_5) = (\overline{\lambda} \cdot \overline{x_5} + \lambda \cdot x_5)\varphi + (\overline{\lambda} \cdot x_5 + \lambda \cdot \overline{x_5})\overline{\varphi}$$

Questa operazione completa la decomposizione. Si puo' tuttavia notare dalla mappa finale di fig. 3.9.6 (b) che la stessa H e' decomponibile.

Definendo:

$$\rho(\lambda, x_5) = \bar{\lambda} \cdot x_5 + \lambda \cdot \bar{x}_5$$

si ottiene in definitiva:

$$f(X) = \bar{\rho} \cdot \varphi + \rho \cdot \bar{\varphi}$$

E' evidente che le funzioni possono essere decomposte in maniera non disgiuntiva; sono stati sviluppati a tale proposito (ad esempio da Curtis "Design of Switching Circuits" Van Nostrand, Princetown, N.J. 1962) dei metodi completamente generali. Tuttavia tali metodi sono talmente complessi da risultare di dubbia utilita' pratica.

C'e' da rilevare che non e' conveniente tentare di decomporre una funzione a meno che non ci sia la possibilita' di realizzare rilevanti economie. Molto spesso pero' funzioni, la cui realizzazione a due livelli e' eccessivamente costosa, vengono decomposte in modo disgiuntivo.

Dall'ultimo esempio fatto si puo' facilmente osservare che le due carte di decomposizione di fig. 3.9.5 sono strutturalmente identiche. Infatti la funzione di disparita' presa in esame e' una funzione simmetrica il cui valore dipende unicamente dal numero delle variabili poste a 1.

La stessa proprieta', cioe' il fatto di avere carte di decomposizione strutturalmente identiche, e' valida per tutte le funzioni simmetriche e quindi quanto visto in quest' ultimo paragrafo si estende alla maggior parte, se non a tutte le funzioni simmetriche. Questa osservazione suggerisce a sua volta che i metodi di decomposizione disgiuntiva sono in genere adatti alla semplificazione delle funzioni simmetriche.

Quale ultima osservazione e' bene far notare che la decomposizione di funzioni con piu' di sei variabili diviene di solito talmente complessa da rendere di difficile e pesante applicazione i metodi proposti.

### **3.10) Sintesi dei circuiti NAND**

La sintesi di un circuito NAND va fatta a partire dall'espressione minima a due livelli, intendendo come espressione minima quella che minimizza il numero di gate, e, a parita' di gate, il circuito che usa il minor numero di ingressi. E' evidente che la minimizzazione deve operare anche sugli invertitori, potendo essi venir concepiti come NAND a un ingresso.

La sintesi quindi appare come un problema nettamente diverso a seconda che si abbiano o meno a disposizione le variabili negate.

#### **3.10.1) Sintesi a partire da variabili dirette e negate.**

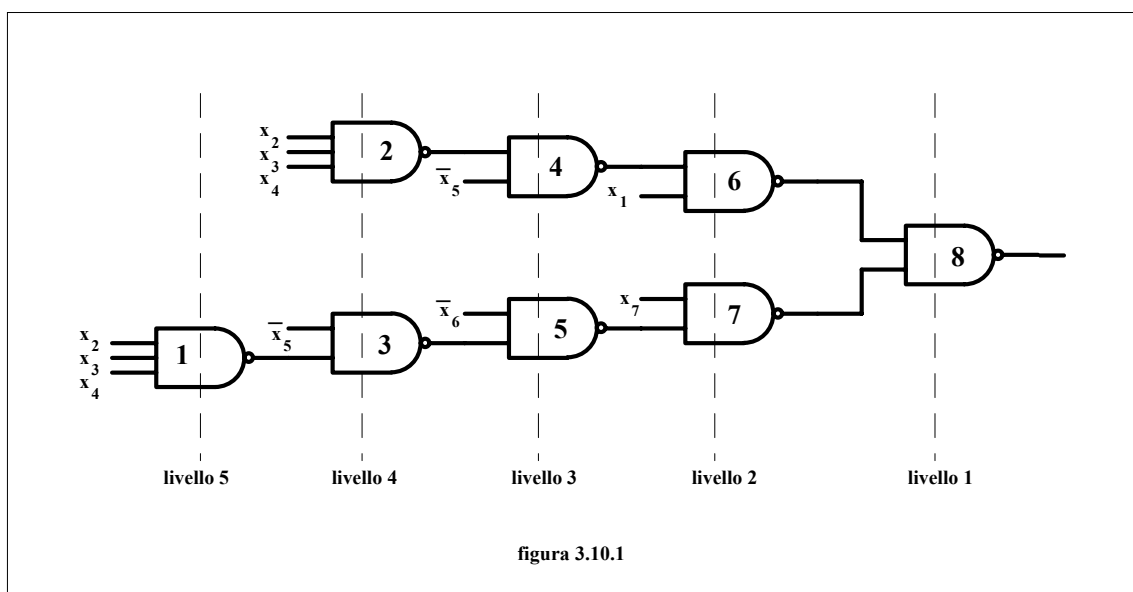
Nel caso dell'analisi si e' visto che un circuito NAND origina una somma di prodotti, poiche' ogni NAND, a parte la negazione o meno delle variabili che entrano su di esso, si comporta come un AND o un OR a seconda del livello in cui si trova inserito lungo ciascun itinerario.

Si puo' allora pensare di realizzare un circuito NAND nello stesso modo in cui si realizza un circuito AND-OR-NOT, tenendo pero' presente che e' necessario negare le variabili che entrano ai livelli dispari.

Ad esempio si voglia realizzare il circuito NAND per la funzione:

$$y = x_1 \cdot (x_2 \cdot x_3 \cdot x_4 + x_5) + [(\overline{x_2} + \overline{x_3} + \overline{x_4})\overline{x_5} + x_6]x_7$$

Il circuito che se ne ricava sarà evidentemente a 5 livelli, uno per la somma dei due addendi, ed i successivi per la realizzazione dei due addendi stessi. Si ottiene il circuito di fig. 3.10.1.



Il circuito ottenuto può venir semplificato osservando che il blocco formato dai NAND 1 e 3 si ritrova all'ingresso del secondo ramo (NAND 2 e 4). Il blocco stesso può quindi essere messo in comune tra i due rami, con un risparmio di due gates.

Ricorrendo ad una semplice osservazione si può usare la stessa procedura di sintesi circuitale anche per le funzioni espresse in termini di prodotto di somme.

Si ha infatti:

$$f(X) = \prod \varphi_i(x_1, \dots, x_n) = \left[ \prod \varphi_i(x_1, \dots, x_n) \right] + 0$$

L'espressione che si ricava in tal modo è nella forma di somma di prodotti e si può allora realizzare il corrispondente circuito NAND.

Le variabili che entrano ai livelli dispari vanno ovviamente complementate. Si può infine eliminare la costante logica 1 che risulta presente al primo livello in quanto:

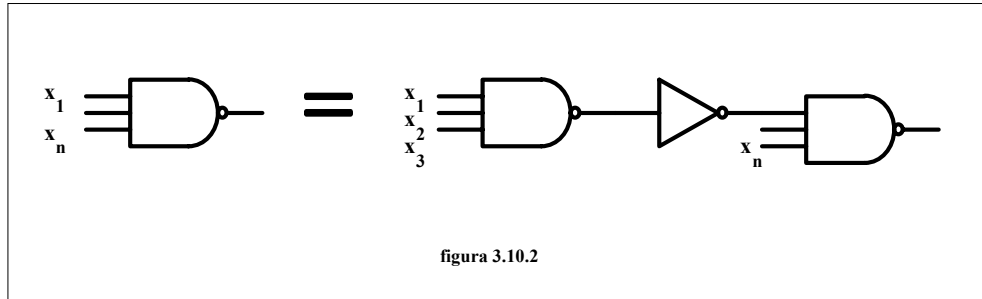
$$x/1 = \overline{x \cdot 1} = \overline{x}$$

Quale ultima osservazione si può notare che, se nel corso della sintesi qualcuno dei NAND, cui si perviene, avesse un numero di ingressi superiore a quello ammissibile, esso non è semplicemente separabile in due elementi in cascata a causa della non associatività dell'operatore NAND.

Ricordando che:

$$x_1/x_2/x_3/..../x_k = (\overline{x_1/x_2/x_3})/..../x_k$$

il circuito cui si perviene e' quello di fig.3.10.2



### 3.10.2) Sintesi a partire dalle sole variabili dirette.

Quando le variabili di ingresso non sono disponibili nella loro forma negata, non e' conveniente usare dei NAND semplicemente per invertirle, in quanto il costo di ciascuna porta e' approssimativamente uguale, qualunque sia il numero dei suoi ingressi.

E' piu' opportuno usare degli artifici, quando possibile, per far entrare le variabili dirette sui livelli pari e quelle negate sui livelli dispari, in modo da eliminare gli invertitori sfruttando le proprieta' delle reti NAND.

Questo modo di operare modifica spesso in modo profondo la forma della funzione, aumentando di solito il numero di livelli e di NAND. La scelta del circuito minimo va fatta confrontando tra di loro un certo numero di circuiti ottenuti per via diversa, derivanti dai due procedimenti di:

- a) Separazione delle variabili dirette da quelle negate su livelli diversi.
- b) Aggiunta di termini ridondanti, per creare elementi comuni o stabilire itinerari differenti.

Si voglia ad esempio realizzare il circuito la cui funzione di trasmissione e':

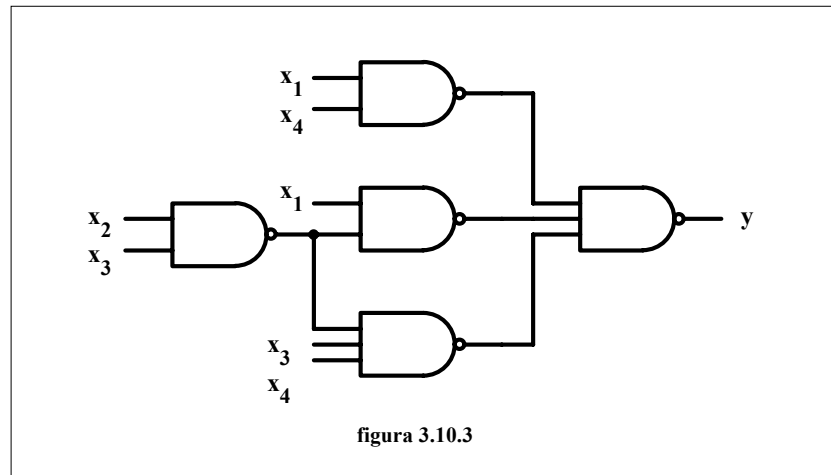
$$y = x_1 \cdot (\overline{x_2} + \overline{x_3} + x_4) + \overline{x_2} \cdot x_3 \cdot x_4$$

Utilizzando la procedura di sintesi illustrata al paragrafo precedente si ricava un circuito con quattro gates e due invertitori. E' possibile viceversa manipolare opportunamente la funzione, ottenendo:

$$y = x_1 \cdot (\overline{x_2} + \overline{x_3}) + x_1 \cdot x_4 + (\overline{x_2} + \overline{x_3}) \cdot x_3 \cdot x_4$$

ottenendo il circuito di fig. 3.10.3 con cinque NAND e nessun invertitore.





Quale ulteriore esempio si voglia eseguire la sintesi del circuito per la funzione:

$$y = \sum (0,1,4,6,7,9,12,15)_m$$

Dalla minimizzazione tramite mappa di Karnaugh si puo' ottenere la forma a due livelli:

$$y = \overline{x_1} \cdot \overline{x_3} \cdot \overline{x_4} + x_2 \cdot \overline{x_3} \cdot \overline{x_4} + \overline{x_2} \cdot \overline{x_3} \cdot x_4 + x_2 \cdot x_3 \cdot \overline{x_4} + \overline{x_1} \cdot x_2 \cdot x_3$$

e fattorizzando

$$y = \overline{x_2} \cdot \overline{x_3} \cdot \overline{x_4} + x_2 \cdot x_3 \cdot (\overline{x_1} + x_4) + \overline{x_3} \cdot \overline{x_4} \cdot (\overline{x_1} + x_2)$$

Tale espressione analitica da' luogo a una realizzazione circuitale che richiede sei NAND e tre invertitori.

Partendo da una diversa forma minima a due livelli, ottenuta leggendo sulla mappa di Karnaugh il termino minimo 0 con quello 1 anziche' con quello 4, si ottiene:

$$y = \overline{x_1} \cdot \overline{x_2} \cdot \overline{x_3} + \overline{x_1} \cdot \overline{x_2} \cdot x_3 + \overline{x_2} \cdot \overline{x_3} \cdot \overline{x_4} + x_2 \cdot x_3 \cdot \overline{x_4} + \overline{x_2} \cdot \overline{x_3} \cdot x_4$$

e fattorizzando nella forma:

$$y = \overline{x_3} \cdot (\overline{x_1} \cdot \overline{x_2} + \overline{x_2} \cdot x_4) + x_2 \cdot (\overline{x_1} \cdot x_3 + x_3 \cdot \overline{x_4}) + \overline{x_2} \cdot \overline{x_3} \cdot \overline{x_4}$$

si puo' trasformare l'espressione nel modo che segue:

$$\begin{aligned} y &= (x_2 + \overline{x_3}) (\overline{x_1} \cdot \overline{x_2} + \overline{x_2} \cdot x_4 + \overline{x_1} \cdot x_3 + x_3 \cdot \overline{x_4}) + x_2 \cdot \overline{x_3} \cdot \overline{x_4} = \\ &= (x_2 \cdot x_3 + \overline{x_3}) (\overline{x_1} \cdot \overline{x_2} + \overline{x_2} \cdot x_4 + \overline{x_1} \cdot x_3 + x_3 \cdot \overline{x_4}) + x_2 \cdot \overline{x_4} \cdot (\overline{x_2} + \overline{x_3}) = \\ &= (x_2 \cdot x_3 + \overline{x_3}) (\overline{x_1} \cdot \overline{x_2} + \overline{x_2} \cdot x_4 + \overline{x_1} \cdot x_2 \cdot x_3 + x_2 \cdot x_3 \cdot \overline{x_4}) + x_2 \cdot \overline{x_4} \cdot (\overline{x_2} + \overline{x_3}) = \\ &= (x_2 \cdot x_3 + \overline{x_3}) (\overline{x_1} + x_4) (\overline{x_2} + x_2 \cdot x_3) + x_2 \cdot \overline{x_4} \cdot (\overline{x_2} + \overline{x_3}) \end{aligned}$$

ottenendo una forma che permette di realizzare il circuito con 7 NAND e un invertitore.

C'e' da osservare che gli artifici da usare non sono intuitivi e che non sempre si riesce a diminuire il numero di gates. Ad esempio, per la funzione:

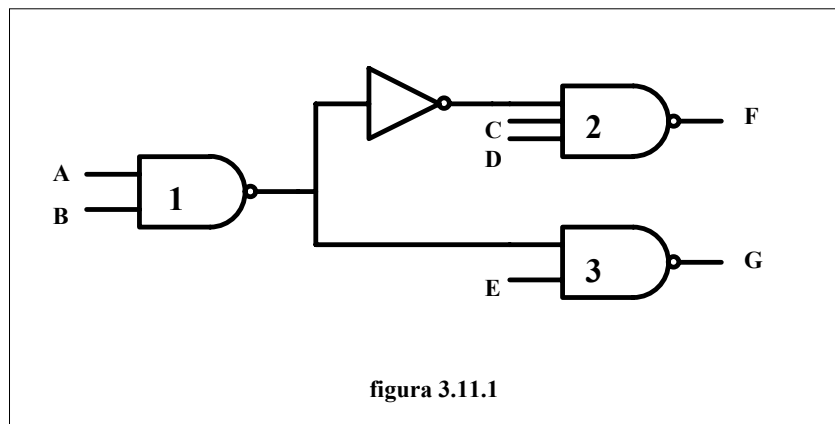
$$y = \overline{x_1} \cdot \overline{x_2} \cdot x_3 + \overline{x_1} \cdot x_2 \cdot \overline{x_3}$$

non c'e' alcun artificio che sia in grado di eliminare gli invertitori di ingresso.

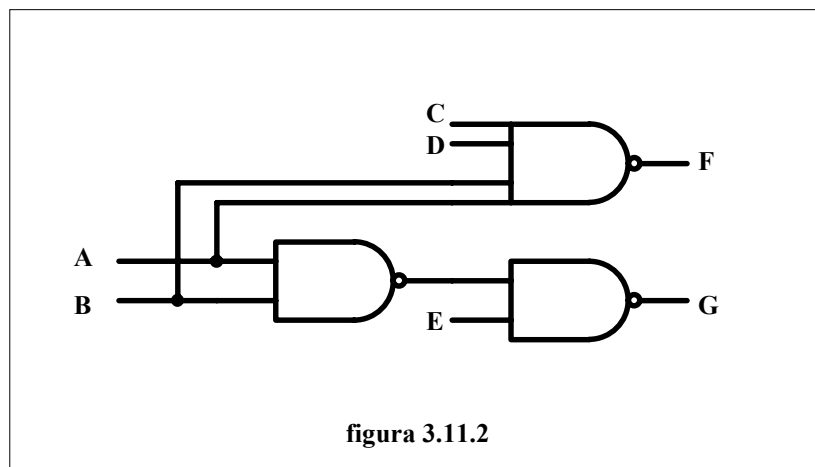
### 3.11) La tecnica del bundling.

Nei circuiti realizzati in logica NAND e' possibile eliminare gli invertitori che compaiono nel circuito, ad eccezione di quelli di ingresso, per mezzo di una tecnica chiamata "bundling" (pagando pero' il prezzo di un maggior numero di ingressi per i rimanenti gates).

La tecnica qui descritta e' applicabile anche ai circuiti realizzati in logica NOR e si puo' usare quando il segnale in uscita di un gate viene invertito per essere applicato all'ingresso di un altro gate, come illustrato in fig. 3.11.1.



Le due inversioni in serie possono dunque essere cancellate, raggruppando gli ingressi del gate 1 e riportandoli all'ingresso del gate 2 e contemporaneamente eliminando l'invertitore. Per l'esempio in esame si ottiene il circuito di fig. 3.11.2.



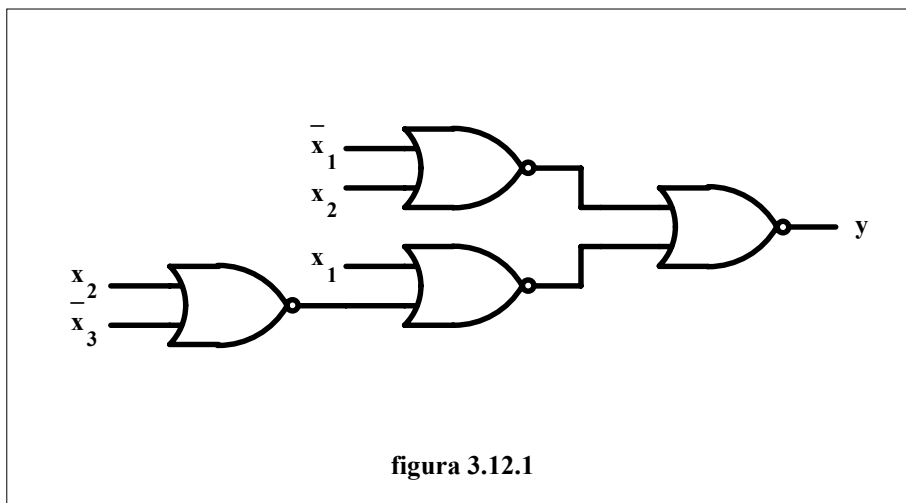
### 3.12) Sintesi dei circuiti NOR

Si e' gia' visto nel caso dell'analisi che un circuito NOR da' origine ad una funzione di trasmissione in termini di prodotto di somme, e che ogni NOR, a parte la negazione o meno delle variabili, si comporta come un AND o un OR secondo il livello occupato dal gate lungo i diversi itinerari. Si puo' quindi eseguire la sintesi di un circuito NOR con gli stessi criteri utilizzati per i circuiti AND-OR-NOT negando poi le variabili che entrano ai livelli dispari, purché la funzione sia espressa in termini di prodotto di somme.

Si voglia da esempio realizzare un circuito NOR, la cui funzione di trasmissione sia:

$$y = (\overline{x_1} + x_2)(x_1 + \overline{x_2} \cdot x_3)$$

Si ottiene la configurazione circuitale di fig. 3.12.1



Con elementi NOR si possono realizzare anche circuiti la cui funzione di trasmissione sia espressa in termini di somma di prodotti. Infatti, tenendo presente che:

$$y = \sum \varphi_i(x_1, \dots, x_n) = \left[ \sum \varphi_i(x_1, \dots, x_n) \right] 1$$

dove le  $\varphi_i$  sono funzioni realizzate tramite prodotti logici, e tenendo altresì presente che le variabili che entrano ai livelli dispari vanno negate, si puo' realizzare un circuito nel quale al primo livello si avra' un NOR con funzioni unicamente di invertitore. Infatti:

$$x \downarrow 0 = \overline{x+0} = \overline{x}$$

### 3.13) Esempio di sintesi.

Si voglia realizzare un circuito capace di sommare due bit secondo le regole dell'addizione binaria. Esso dovra' avere due uscite, S e R, di cui la prima rappresenta la somma dei due bit, mentre la seconda e' il riporto al rango immediatamente successivo. La relativa tavola di verita' e' la seguente:

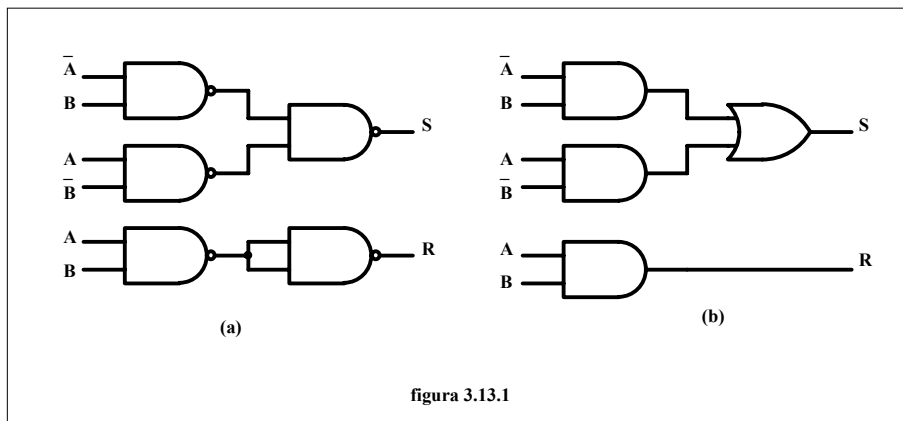
A	B	S	R
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Le equazioni del circuito saranno quindi:

$$S = \bar{A}.B + A.\bar{B}$$

$$R = A.B$$

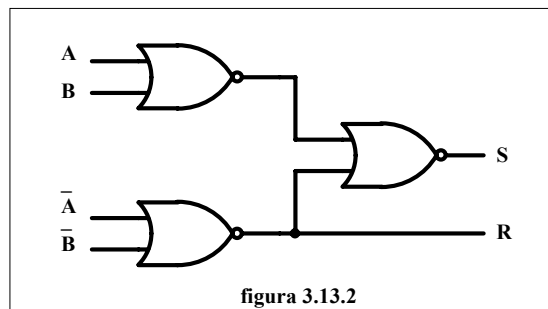
In fig. 3.13.1 (a) e (b) sono riportati i circuiti realizzati rispettivamente in logica NAND e in logica AND-OR-NOT.



Per la realizzazione mediante elementi NOR e' invece conveniente scrivere la funzione nella forma:

$$S = (\bar{A} + \bar{B})(A + B)$$

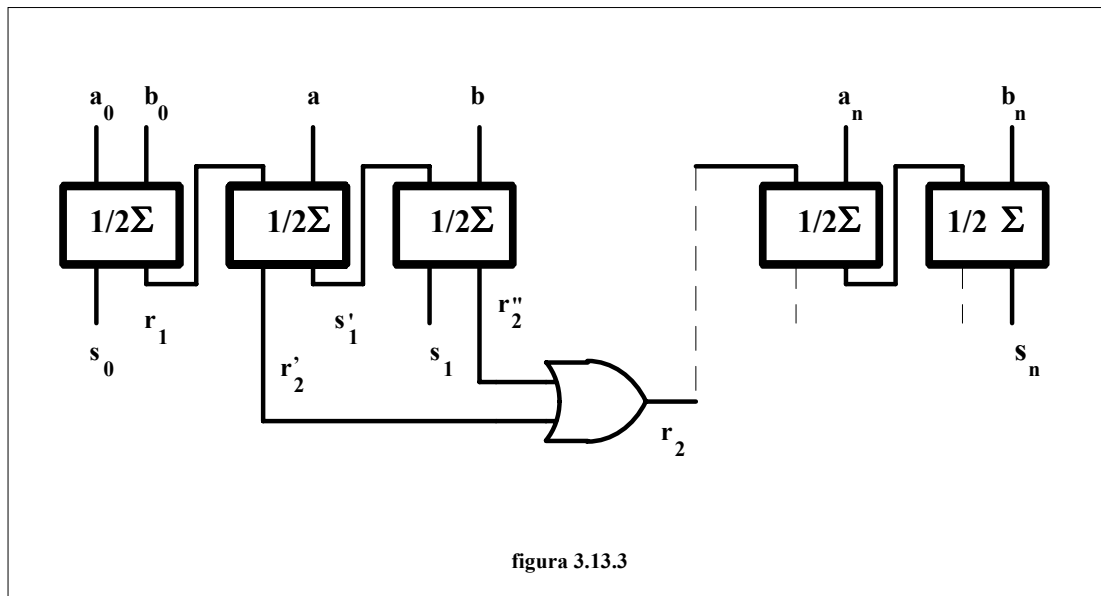
A partire da tale espressione si ottiene finalmente il circuito di fig. 3.13.2.



Si sono in tal modo ottenute tre diverse realizzazioni dello stesso circuito, che è conosciuto sotto il nome di semisommatore binario, in quanto può essere usato solamente per i bit meno significativi dei due numeri da sommare.

Se ne deduce che il circuito realizzato non è direttamente utilizzabile per la somma degli altri ranghi, per i quali è necessario un circuito sommatore a tre ingressi, i due bit, cifre dei numeri da sommare, ed il riporto dal rango precedente.

Il problema può essere risolto secondo due vie diverse. La prima prevede la sintesi di un circuito sommatore completo (o full-adder), la seconda un'opportuna interconnessione di due semisommatori come è illustrato in fig. 3.13.3.



### 3.14) I circuiti multiterminali.

È detto circuito multiterminale (MT) un circuito combinatorio a  $n$  ingressi e  $m$  uscite, su ciascuna delle quali è realizzata una diversa funzione  $y$  delle variabili  $x_1, x_2, \dots, x_n$ . Circuito MT è ad esempio il semisommatore binario esaminato al paragrafo precedente.

Il problema che i circuiti MT pongono è quello dell'individuazione di termini comuni alle diverse funzioni d'uscita, in modo da realizzare circuiti, che, mettendo in comune tra più uscite le parti circuitali comuni, risultino più semplici e più economici. È evidente che, operando le semplificazioni separatamente sulle singole funzioni, le parti comuni alle diverse uscite potrebbero andare perse.

Per funzioni fino a cinque variabili il metodo di semplificazione più rapido, anche se non del tutto rigoroso, è quello delle mappe di Karnaugh, mentre se il numero di variabili è maggiore o si volesse usare un procedimento assolutamente sistematico è più conveniente rifarsi al metodo analitico di Quine- McCluskey.

#### 3.14.1) Semplificazione mediante le mappe di Karnaugh.

Il metodo di semplificazione per i circuiti MT mediante le mappe di Karnaugh viene condotto secondo i seguenti passi:

- 1) Si riporta ognuna delle  $y_i$  sulle mappe.

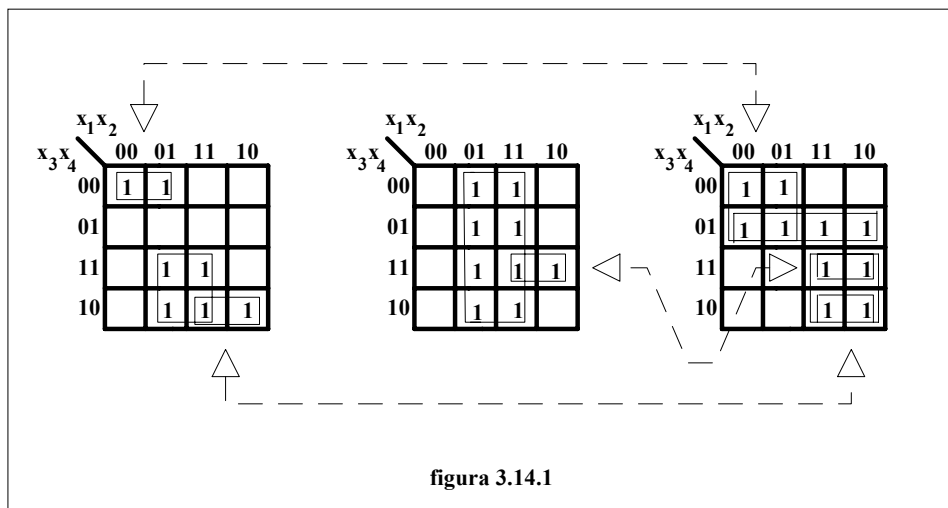
- 2) Si individuano su ciascuna mappa i sottoinsiemi che danno luogo alla forma minima.
- 3) Si confrontano tutti gli insiemi così individuati mettendo in comune quelli che compaiono in almeno due mappe e spezzando quelli che possono essere ottenuti come somma di più insiemi di altre mappe.
- 4) Per ogni uscita si scrivono le forme fattorizzate ricavate dai sottoinsiemi di cui al passo 2 e 3 e si sceglie la forma più economica.

Si voglia, ad esempio, realizzare un circuito MT a 4 ingressi e tre uscite:

$$y_1 = \sum (0,4,6,7,10,14,15)_m \qquad y_2 = \sum (4,5,6,7,11,12,13,14,15)_m$$

$$y_3 = \sum (0,1,4,5,9,10,11,13,14,15)_m$$

In fig.3.14.1 sono riportate le relative mappe di Karnaugh.



Le espressioni minime a due livelli, ricavabili dalle mappe considerate separatamente, sono:

$$y_1 = \overline{x_1} \cdot \overline{x_3} \cdot \overline{x_4} + x_2 \cdot x_3 + x_1 \cdot x_3 \cdot \overline{x_4}$$

$$y_2 = x_2 + x_1 \cdot x_3 \cdot x_4$$

$$y_3 = \overline{x_1} \cdot \overline{x_3} + \overline{x_3} \cdot x_4 + x_1 \cdot x_3$$

e la relativa realizzazione circuitale in termini di AND-OR-NOT richiede 10 gates.

Spezzando gli insiemi  $\overline{x_1} \cdot \overline{x_3}$  e  $x_1 \cdot x_3$  di  $y_3$  come illustrato in fig.3.14.1, si ha:

$$y_1 = \overline{x_1} \cdot \overline{x_3} \cdot \overline{x_4} + x_2 \cdot x_3 + x_1 \cdot x_3 \cdot \overline{x_4}$$

$$y_2 = x_2 + x_1 \cdot x_3 \cdot x_4$$

$$y_3 = \overline{x_3} \cdot x_4 + x_1 \cdot x_3 \cdot x_4 + x_1 \cdot x_3 \cdot \overline{x_4} + \overline{x_1} \cdot \overline{x_3} \cdot \overline{x_4}$$

che, sempre in termini di AND-OR-NOT, richiedono solamente otto gates per la loro realizzazione.

### 3.14.2) Semplificazione dei circuiti multiterminali con il metodo tabellare.

Il metodo e' molto simile a quello illustrato in precedenza per i circuiti ad uscita singola. La sua illustrazione, per esigenze di semplicita' di trattazione, verra' fatta con riferimento ad un particolare esempio. Si voglia realizzare un circuito MT a quattro ingressi e tre uscite, secondo la seguente tavola di verita':

$x_1 x_2 x_3 x_4$	$y_1 y_2 y_3$
0 0 0 0	0 0 0
0 0 0 1	1 0 1
0 0 1 0	0 0 0
0 0 1 1	0 0 0
0 1 0 0	0 1 1
0 1 0 1	0 1 1
0 1 1 0	1 1 0
0 1 1 1	1 1 0
1 0 0 0	0 0 0
1 0 0 1	1 0 1
1 0 1 0	0 1 1
1 0 1 1	1 0 1
1 1 0 0	0 1 1
1 1 0 1	0 1 1
1 1 1 0	1 1 1
1 1 1 1	1 1 1

a) Costruzione della prima tabella da cui partire con la semplificazione.

Vengono prese in considerazione tutte le configurazioni degli ingressi per cui almeno una delle funzioni vale 1. Ad esse vengono aggiunte, come parte letterale, le configurazioni delle uscite che valgono 1. Ad esempio, essendo per  $x_1 x_2 x_3 x_4 = 0001$   $y_1 y_2 y_3 = 101$ , si avra' un termine  $0 0 0 1 y_1 - y_3$ . I termini cosi' ricavati sono ordinati in livelli allo stesso modo gia' illustrato per le funzioni a singola uscita. Si ottiene:

1	0	0	0	1	$y_1$	-	$y_3$
4	0	1	0	0	-	$y_2$	$y_3$
6	0	1	1	0	$y_1$	$y_2$	-
5	0	1	0	1	-	$y_2$	$y_3$
12	1	1	0	0	-	$y_2$	$y_3$
10	1	0	1	0	-	$y_2$	$y_3$
9	1	0	0	1	$y_1$	-	$y_3$
7	0	1	1	1	$y_1$	$y_2$	-
13	1	1	0	1	-	$y_2$	$y_3$
14	1	1	1	0	$y_1$	$y_2$	$y_3$
11	1	0	1	1	$y_1$	-	$y_3$
15	1	1	1	1	$y_1$	$y_2$	$y_3$

b) Confronti tra i termini e costruzione delle tabelle successive.

I confronti vanno ancora eseguiti tra tutti i termini di un livello e tutti i termini del livello successivo. Sono semplificabili tutti i termini che differiscono per un solo bit, purché compaiano ambedue in almeno una funzione. Sono ad esempio semplificabili i termini:

$$\begin{array}{l} 0011 \quad y_1 - - \\ 0001 \quad y_1 y_2 - \end{array}$$

dando luogo al termine:

$$00-1 \quad y_1 - -$$

mentre non sono semplificabili i termini:

$$\begin{array}{l} 0011 \quad y_1 - - \\ 0001 \quad - y_2 y_3 \end{array}$$

Si ottengono in tal modo le seguenti tabelle:

<hr/>					
<b>A ...</b>	1,5	0 - 0 1	- -	$y_3$	
	1,9	- 0 0 1	$y_1$ -	$y_3$	
	4,6	0 1 - 0	$- y_1$	$y_2$ -	
	4,5	0 1 0 -	- -	$y_2 y_3$	
	4,12	- 1 0 0	- -	$y_2 y_3$	
<hr/>					
<b>B ...</b>	6,7	0 1 1 -	$y_1 y_2$ -		
	6,14	- 1 1 0	$y_1 y_2$ -		
	5,7	0 1 - 1	$- y_1$	$y_2$ -	
	5,13	- 1 0 1	- -	$y_2 y_3$	
	12,13	1 1 0 -	- -	$y_2 y_3$	
	12,14	1 1 - 0	- -	$y_2 y_3$	
	10,14	1 - 1 0	- -	$y_2 y_3$	
	10,11	1 0 1 -	- -	$y_3$	
	9,13	1 - 0 1	- -	$y_3$	
	9,11	1 0 - 1	$y_1$ -	$y_3$	
<hr/>					
<b>D ...</b>	13,15	1 1 - 1	- -	$y_2 y_3$	
	14,15	1 1 1 -	$y_1 y_2$	$y_3$	
	11,15	1 - 1 1	$y_1$ -	$y_3$	
	7,15	- 1 1 1	$y_1 y_2$ -		
<hr/>					



E ...	1,5,9,13	-	-	0	1	-	-	$y_3$
	4,5,6,7	0	1	-	-	-	-	$y_2$
F ...	4,5,12,13	-	1	0	-	-	-	$y_2 y_3$
	4,6,12,14	-	1	-	0	-	-	$y_2$
G ...	6,7,14,15	-	1	1	-	$y_1$	$y_2$	-
	5,7,13,15	-	1	-	1	-	$y_2$	-
H ...	12,13,14,15	1	1	-	-	-	$y_2 y_3$	-
I ...	10,11,14,15	1	-	1	-	-	$y_3$	-
L ...	9,11,13,15	1	-	-	1	-	$y_3$	-

---

**M ... 4,5,6,7,12,13,14,15    - 1 - - -  $y_2$  -**

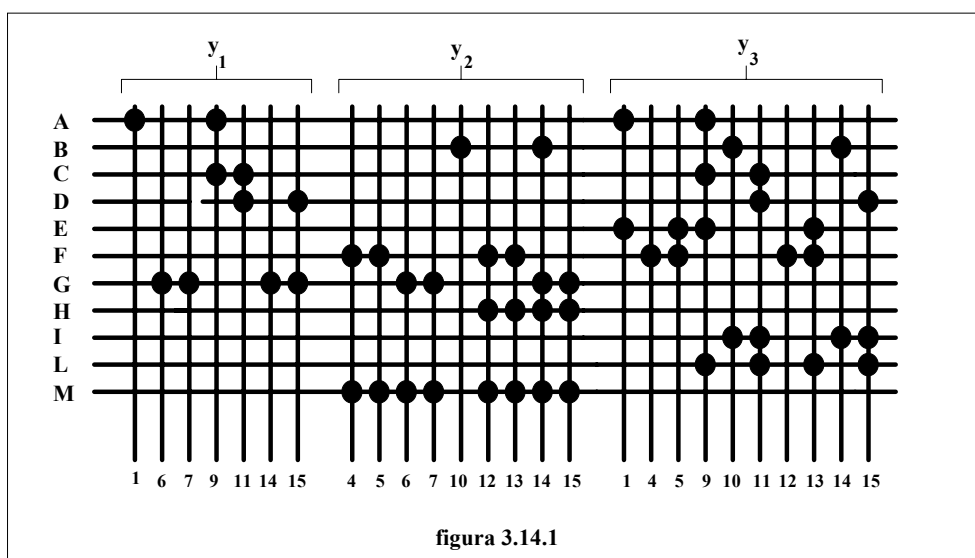
---

Si noti che nel passaggio dalla seconda alla terza tabella il termine 1,9, pur dando luogo assieme al termine 5,13 all'implicante 1,5,9,13 della terza tabella, deve purtuttavia essere considerato un implicante primo per la  $y_1$ , in quanto la semplificazione citata ha luogo per la sola  $y_3$ .

c) Determinazione della copertura minima.

La determinazione della copertura minima si fa ancora con l'aiuto di un reticolo, realizzato come se le funzioni di uscita dovessero essere tradotte in circuito in modo indipendente. Il reticolo ha cioè sulle ascisse i termini minimi delle singole funzioni da realizzare e sulle ordinate gli implicanti primi individuati. Dall'esame del reticolo di fig. 3.14.1 si riconoscono i seguenti implicanti essenziali:

- A che copre i termini minimi 1 e 9 in  $y_1$  e  $y_3$ .
- B che copre i termini minimi 10 e 14 in  $y_2$  e  $y_3$ .
- F che copre i termini minimi 4, 5, 12 e 13 in  $y_2$  e  $y_3$ .
- G che copre i termini minimi 6, 7, 14 e 15 in  $y_1$  e  $y_2$ .



Rimangono scoperti il termine minimo 11 in  $y_1$  e  $y_3$  e quello 15 in  $y_3$ ; l'implicante primo D li copre entrambi.

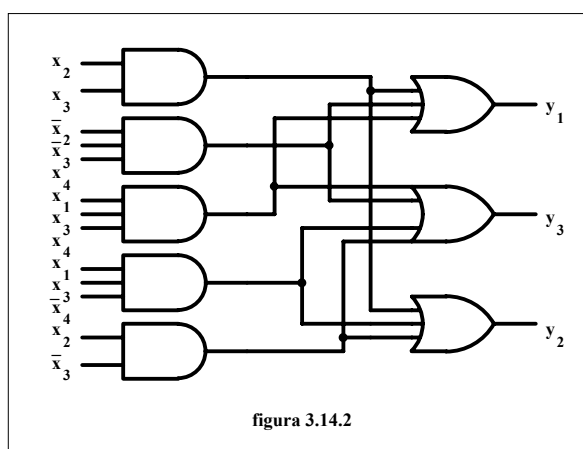
Si ha in tal modo:

$$y_1 = G + A + D = x_2 \cdot x_3 + \overline{x_2} \cdot \overline{x_3} \cdot x_4 + x_1 \cdot x_3 \cdot x_4$$

$$y_2 = G + F + B = x_2 \cdot x_3 + x_2 \cdot \overline{x_3} + x_1 \cdot x_3 \cdot \overline{x_4}$$

$$y_3 = D + F + B + A = x_1 \cdot x_3 \cdot x_4 + x_2 \cdot \overline{x_3} + x_1 \cdot x_3 \cdot \overline{x_4} + \overline{x_2} \cdot \overline{x_3} \cdot x_4$$

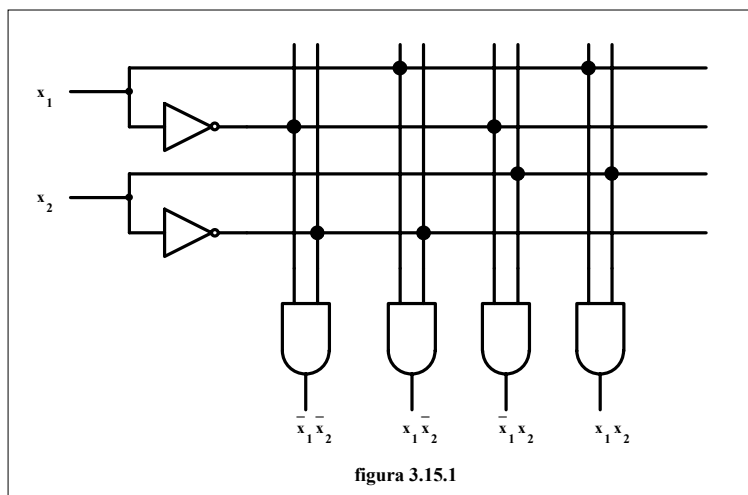
La realizzazione circuitale in termini di AND-OR-NOT richiede solamente 8 gates, come e' illustrato in fig. 3.14.2.



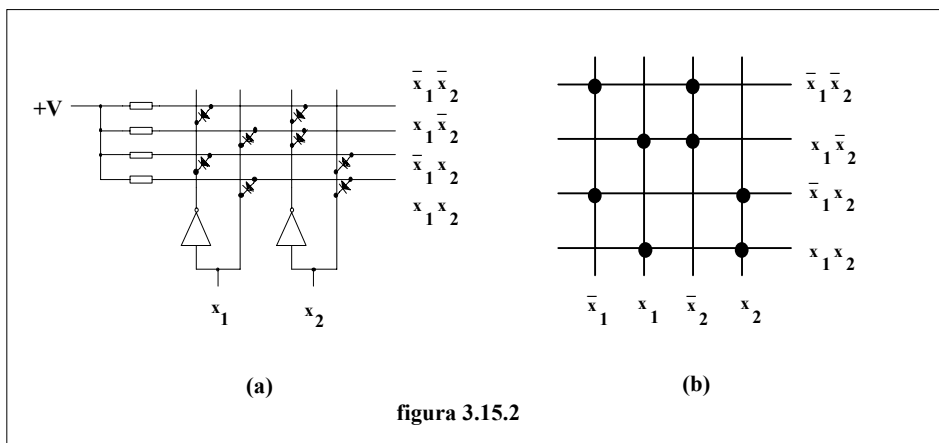
### 3.15) Selettori e matrici.

I selettori sono particolari circuiti multiterminali; circuiti cioe' a  $n$  ingressi e  $2^n$  uscite, ognuna delle quali assume valore 1 in corrispondenza ad una e una sola delle  $2^n$  configurazioni possibili dell'ingresso.

Un selettore puo' ovviamente essere realizzato con  $2^n$  circuiti AND a  $n$  ingressi, ciascuno dei quali sintetizza uno dei  $2^n$  termini minimi delle  $n$  variabili  $x_1, x_2, \dots, x_n$ , come illustrato in fig. 3.15.1.



I selettori prendono il nome di matrici quando sono realizzati secondo la particolare tecnica circuitale illustrata in fig. 3.15.2 (a). In fig. 3.15.2 (b) e' riportata la relativa rappresentazione simbolica.

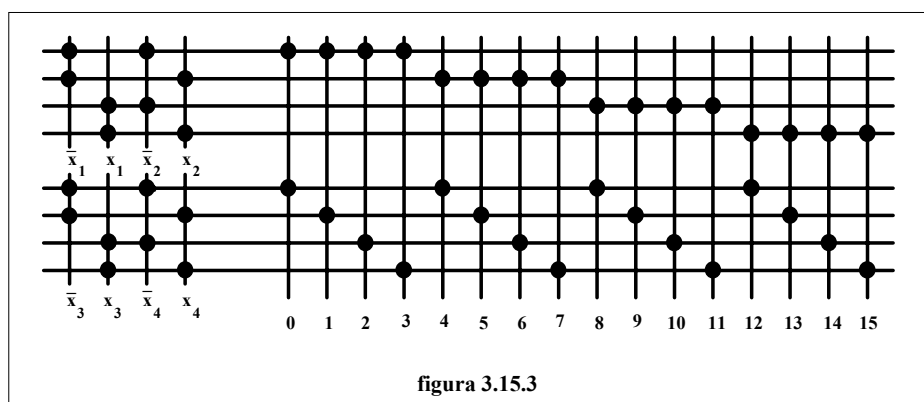


Le matrici sono circuiti particolarmente semplici e, malgrado che il loro uso sia sempre meno diffuso, sopravvivono al giorno d'oggi in particolare nei circuiti di indirizzamento, presentando quale inconveniente il fatto che all'aumentare del numero degli ingressi il numero dei diodi necessari, pari a  $n \cdot 2^n$ , diviene eccessivo. Montaggi piu' economici possono essere ottenuti utilizzando la tecnica delle matrici multiple.

Si tenga presente che quanto verra' esposto nei paragrafi successivi con riferimento al numero di diodi necessari a realizzare un selettore con la tecnica delle matrici rimane pienamente valido anche se la realizzazione non e' a matrice, purché ci si riferisca al numero di ingressi anziché al numero di diodi.

### 3.15.1) Matrici multiple.

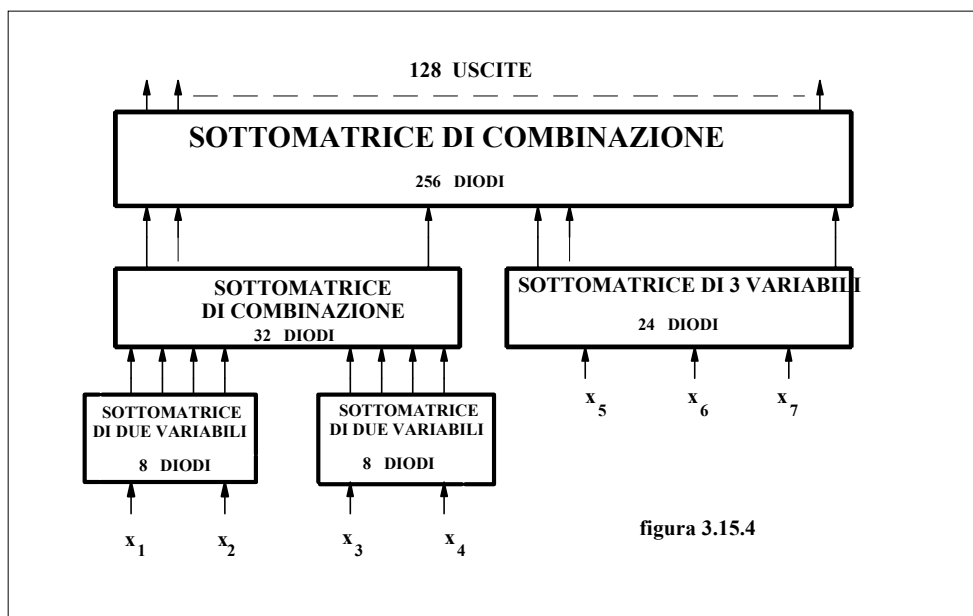
Prendono il nome di matrici multiple i selettori in cui la scelta di una delle  $2^n$  uscite viene fatta attraverso un certo numero di matrici parziali di 2 o 3 variabili, le cui uscite vengono combinate in tutti i modi possibili ad un successivo livello. In fig. 3.15.3 e' rappresentata una matrice multipla di 4 variabili realizzata con due sottomatrici da due variabili ciascuna.



La ripartizione ottimale delle  $n$  variabili ( $n \geq 5$ ) in un selettore a matrici multiple si ottiene ripartendo le variabili in due gruppi  $G1$  e  $G2$  il piu' possibile uguali tra di loro ed

ognuno di tali gruppi in ulteriori altri due il piu' possibile uguali fino ad ottenere gruppi di non piu' di tre variabili.

In fig. 3.15.4 e' riportata la ripartizione ottima per una matrice di 7 variabili. Sono in tal caso necessari 328 diodi in luogo degli 896 ( $7 \cdot 2^7$ ) necessari per la realizzazione a matrice semplice.

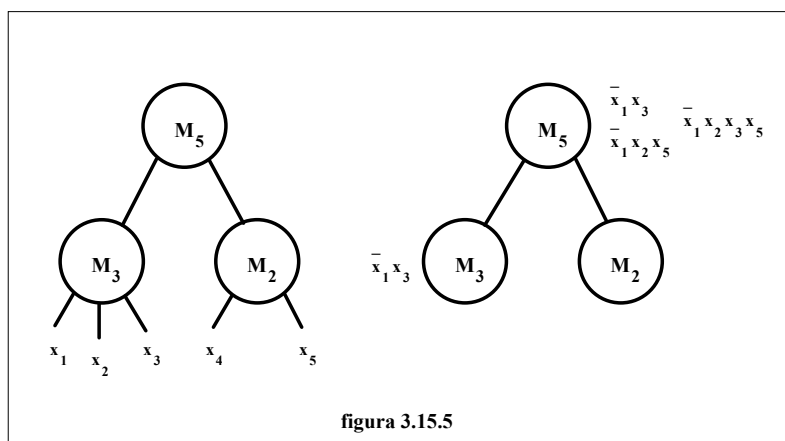


### 3.15.2) Matrici incomplete.

In una realizzazione a matrice multipla, se qualcuna delle  $2^n$  uscite non e' necessaria, il numero di diodi da utilizzare diminuisce non solo nella matrice di combinazione dell'ultimo stadio, ma anche in quelle degli stadi precedenti. La diminuzione dipende dalla ripartizione delle variabili effettuata sulle varie matrici di ingresso. A titolo di esempio si voglia realizzare il selettore di 5 variabili fra le cui uscite non compaiano i termini minimi della funzione:

$$y = \overline{x_1} \cdot (x_2 \cdot x_5 + x_3)$$

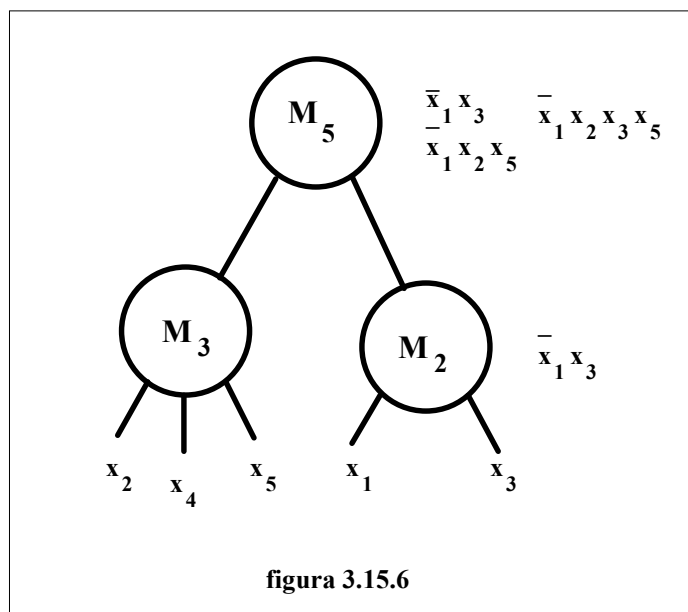
Si supponga di aver ripartito le variabili come illustrato in fig. 3.15.5.



Il termine  $\overline{x_1} \cdot x_3$  apporta una riduzione in  $M_3$  e  $M_5$  e per tenerne conto lo scriveremo accanto a  $M_3$  e  $M_5$ ; il termine  $\overline{x_1} \cdot x_2 \cdot x_5$  da' invece luogo a una riduzione solo su  $M_5$  poiche'  $x_1$  e  $x_2$  appartengono a  $M_3$  e  $x_5$  a  $M_2$ . Lo scriveremo allora solo in corrispondenza a  $M_5$ . A parte verra' poi preso in considerazione il prodotto delle variabili comuni  $\overline{x_1} \cdot x_2 \cdot x_3 \cdot x_5$ .

Ora in  $M_3$   $2^1$  uscite contengono il termine  $\overline{x_2} \cdot x_3$  e quindi la loro eliminazione fa risparmiare 6 diodi. In  $M_5$   $2^3=8$  uscite contengono il termine  $\overline{x_1} \cdot x_3$ , mentre  $2^2=4$  uscite contengono il termine  $\overline{x_1} \cdot x_2 \cdot x_5$  e  $2^1=2$  uscite il termine  $\overline{x_1} \cdot x_2 \cdot x_3 \cdot x_5$ . Vengono quindi eliminate  $8 + 4 - 2=10$  uscite con un risparmio di 20 diodi. In totale pertanto si risparmiano 26 diodi.

Con una diversa ripartizione delle variabili, quale ad esempio quella di fig. 3.15.6 si risparmiano ancora 20 diodi in  $M_5$ , ma solamente 2 in  $M_2$ .



Si vede immediatamente che e' conveniente ripartire le variabili in modo che i termini da eliminare facciano sentire la loro influenza il piu' possibile in prossimita' degli ingressi e compaiano nelle sottomatrici di ingresso a massimo numero di variabili.

### 3.16) Implementazione delle funzioni simmetriche.

Le funzioni simmetriche, in aggiunta alle proprieta' logiche messe in luce al capitolo II, possiedono anche interessanti proprieta' aritmetiche.

Quando le variabili di simmetria di una funzione simmetrica vengono usate come ingressi di un sommatore binario, le uscite di quest'ultimo realizzano le funzioni simmetriche corrispondenti a quell'insieme di variabili.

Ad esempio, se le variabili di simmetria sono A, B, C, D il relativo sommatore binario corrispondera' alla seguente tavola di verita':

A	B	C	D	$S_1$	$S_2$	$S_3$
0	0	0	0	0	0	0
0	0	0	1	0	0	1
0	0	1	0	0	0	1
0	0	1	1	0	1	0
0	1	0	0	0	0	1
0	1	0	1	0	1	0
0	1	1	0	0	1	0
0	1	1	1	0	1	1
1	0	0	0	0	0	1
1	0	0	1	0	1	0
1	0	1	0	0	1	0
1	0	1	1	0	1	1
1	1	0	0	0	1	0
1	1	0	1	0	1	1
1	1	1	0	0	1	1
1	1	1	1	1	0	0

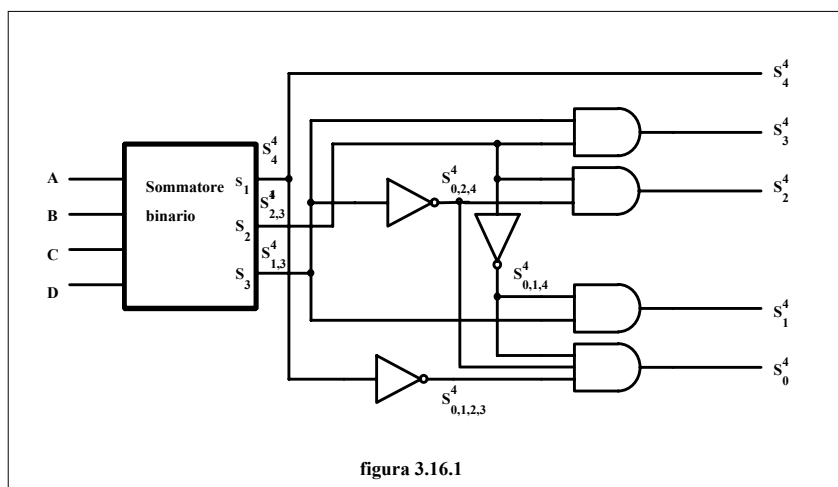
Si ricava pertanto:

$$S_1 = A \cdot B \cdot C \cdot D = S_4^4(A, B, C, D)$$

$$S_2 = \bar{A} \cdot \bar{B} \cdot C \cdot D + \bar{A} \cdot B \cdot \bar{C} \cdot D + \bar{A} \cdot B \cdot C \cdot \bar{D} + \bar{A} \cdot B \cdot C \cdot D + A \cdot \bar{B} \cdot \bar{C} \cdot D + A \cdot \bar{B} \cdot C \cdot \bar{D} + A \cdot \bar{B} \cdot C \cdot D + A \cdot B \cdot \bar{C} \cdot \bar{D} + A \cdot B \cdot \bar{C} \cdot D + A \cdot B \cdot C \cdot \bar{D} = S_{(2,3)}^4(A, B, C, D)$$

$$S_3 = \bar{A} \cdot \bar{B} \cdot \bar{C} \cdot D + \bar{A} \cdot \bar{B} \cdot C \cdot \bar{D} + \bar{A} \cdot B \cdot \bar{C} \cdot \bar{D} + \bar{A} \cdot B \cdot C \cdot D + A \cdot \bar{B} \cdot \bar{C} \cdot \bar{D} + A \cdot \bar{B} \cdot C \cdot D + A \cdot B \cdot \bar{C} \cdot \bar{D} + A \cdot B \cdot C \cdot D = S_{(1,3)}^4(A, B, C, D)$$

Per ottenere le funzioni simmetriche elementari delle 4 variabili A, B, C, D le uscite del sommatore binario possono essere combinate come illustrato in fig. 3.16.1.



La disponibilita' di circuiti integrati che realizzano la funzione di sommatore binario fa si' che l'implementazione di funzioni simmetriche diventi particolarmente semplice, operando nel campo dell'aritmetica binaria piuttosto che in quello dell'algebra booleana.

### ESEMPIO 1

Si voglia progettare il circuito che realizza la seguente funzione simmetrica:

$$f(A, B, C, D, E, F) = S_5^6(A, B, C, D, E, F)$$

La funzione assegnata puo' essere scritta come:

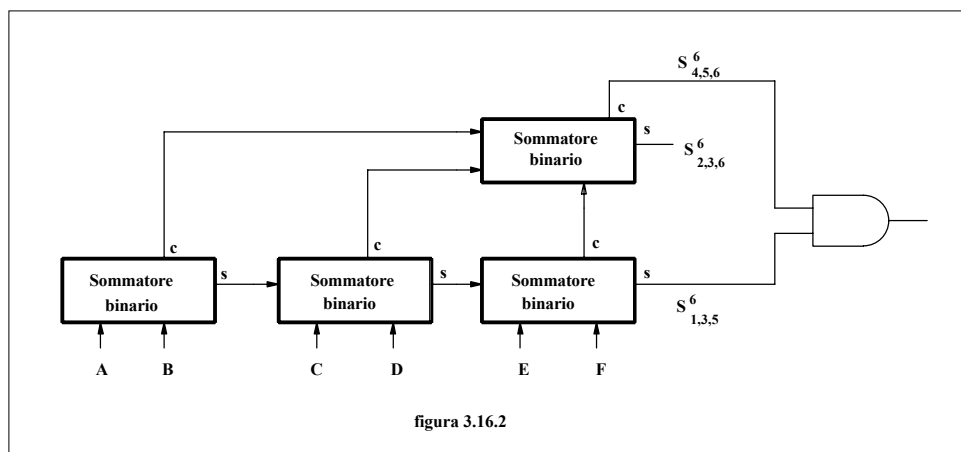
$$\begin{aligned} f(A, B, C, D, E, F) &= A \cdot B \cdot C \cdot D \cdot E \cdot \bar{F} + A \cdot B \cdot C \cdot D \cdot \bar{E} \cdot F + A \cdot B \cdot C \cdot \bar{D} \cdot E \cdot F + \\ &+ A \cdot B \cdot \bar{C} \cdot D \cdot E \cdot F + A \cdot \bar{B} \cdot C \cdot D \cdot E \cdot F + \bar{A} \cdot B \cdot C \cdot D \cdot E \cdot F = \\ &= (A \cdot B \cdot C \cdot D \cdot E + A \cdot B \cdot C \cdot D \cdot F + A \cdot B \cdot C \cdot E \cdot F + A \cdot B \cdot D \cdot E \cdot F + \\ &+ A \cdot C \cdot D \cdot E \cdot F + B \cdot C \cdot D \cdot E \cdot F) \cdot (\bar{A} + \bar{B} + \bar{C} + \bar{D} + \bar{E} + \bar{F}) = \\ &= [B \cdot C \cdot (A \cdot D \cdot E + A \cdot D \cdot F) + B \cdot E \cdot (A \cdot C \cdot F + A \cdot D \cdot F) + \\ &+ E \cdot F \cdot (A \cdot C \cdot D + B \cdot C \cdot D)] \cdot \overline{A \cdot B \cdot C \cdot D \cdot E \cdot F} \end{aligned}$$

L'implementazione di questa funzione, ottenuta con i metodi di progetto tradizionali, richiede, se realizzata con elementi NOR, 19 gates, senza contare gli invertitori necessari

La stessa funzione puo' tuttavia essere scritta come:

$$f(A, B, C, D, E, F) = S_{(4,5,6)}^6(A, B, C, D, E, F) \cdot S_{(1,3,5)}^6(A, B, C, D, E, F)$$

La realizzazione ottenuta per mezzo di sommatore binari e' riportata in fig. 3.16.2.



### ESEMPIO 2

Si progetti il circuito che realizza la seguente funzione simmetrica:

$$f(A,B,C,D,E) = S_3^5(\bar{A}, B, \bar{C}, D, \bar{E})$$

Le tecniche di progetto tradizionali conducono, nella migliore delle ipotesi, ad un circuito che impiega 13 gates.

La funzione data puo' tuttavia essere anche scritta come :

$$S_3^5(\bar{A}, B, \bar{C}, D, \bar{E}) = S_{2,3}^5(\bar{A}, B, \bar{C}, D, \bar{E}) S_{1,3,5}^6(\bar{A}, B, \bar{C}, D, \bar{E})$$

e la relativa realizzazione ottenuta con i sommatore e' riportata in fig. 3.16.3.

