2 FUNZIONI COMBINATORIE

2.1 Tavola della verità

Consideriamo una funzione booleana di due variabili A, B, molto semplice:

$$f(A,B) = \overline{A} + \overline{B} \tag{2.1.1}$$

Per *descrivere* la funzione il metodo più efficace è quello di costruire la tabella dei possibili valori **0**, **1** delle variabili e il valore corrispondente della funzione, come nella tabella 2.1.1, chiamata usualmente *tavola della verità.*

Tabella 2.1.1

A	B	$\overline{A} + \overline{B}$
0	0	1
1	0	1
0	1	1
1	1	0

Questa tabella ci descrive una f il cui valore è funzione predeterminata degli ingressi. Le funzioni di questo tipo sono dette *combinatorie*. C'è un'altra classe di funzioni, che vedremo nel capitolo 5, che non possono essere descritte dalla sola funzione combinatoria degli ingressi e che sono dette *sequenziali* in quanto richiedono la conoscenza degli stati degli ingressi nell'intervallo di tempo precedente.

La stessa tabella 2.1.1 potrebbe essere usata anche per *definire* la funzione, una volta stabiliti i suoi valori in corrispondenza delle possibili combinazioni delle variabili d'ingresso.

È facile verificare dalla tavola della verità 2.1.1 che la stessa funzione potrebbe essere scritta nella altre forme:

$$
\begin{aligned}
f(A,B) &= \overline{A \bullet B} \\
&= \overline{A}\,\overline{B} + \overline{A}B + A\overline{B} \\
&= \overline{A} + A\overline{B} \\
&= \overline{B} + \overline{A}B \qquad \text{etc.}
\end{aligned}
$$

Ciascuna delle espressioni precedenti differisce per il numero di operatori circuitali necessari per la sua realizzazione, ma in ogni caso otterremmo lo stesso valore di f per una data coppia di valori A e B.

Nella ricerca del metodo per sintetizzare in modo sistematico le funzioni booleane, troveremo che esse potranno essere espresse indifferentemente come somma di un opportuno numero di prodotti (*termini minimi*) delle variabili, ovvero come prodotto di un opportuno numero di somme (*termini massimi*) delle variabili.

2.2 Termini minimi e termini massimi

Definiamo come *termine minimo* di n variabili un prodotto booleano di *tutte* le variabili presenti come tali o come complemento. Date n variabili i termini minimi sono quindi 2^n. Nel caso di due variabili essi sono:

$$\overline{A}\overline{B}, A\overline{B}, \overline{A}B, AB.$$

Sono definiti analogamente i *termini massimi* come somme di variabili booleane. Anche i termini massimi sono naturalmente 2^n e nel caso di due variabili sono:

$$(\overline{A}+\overline{B}), (A+\overline{B}), (\overline{A}+B), (A+B).$$

Nella tavola 2.2.1 sono dati i termini minimi e i termini massimi, nel caso di tre variabili, insieme con i loro simboli convenzionali m_n e M_n .

Gli indici corrispondono, convenzionalmente, al peso binario (1, 2, 4, 8 etc.) dato alle variabili seguendo l'ordine alfabetico.

Il termine minimo m_i risulta uguale ad **1** *solo per la combinazione delle variabili della i-esima riga* mentre il termine

Tabella 2.2.1

A	B	C		
0	0	0	$m_0 = \overline{A}\overline{B}\overline{C}$	$M_0 = \overline{A}+\overline{B}+\overline{C}$
1	0	0	$m_1 = A\overline{B}\overline{C}$	$M_1 = A+\overline{B}+\overline{C}$
0	1	0	$m_2 = \overline{A}B\overline{C}$	$M_2 = \overline{A}+B+\overline{C}$
1	1	0	$m_3 = AB\overline{C}$	$M_3 = A+B+\overline{C}$
0	0	1	$m_4 = \overline{A}\overline{B}C$	$M_4 = \overline{A}+\overline{B}+C$
1	0	1	$m_5 = A\overline{B}C$	$M_5 = A+\overline{B}+C$
0	1	1	$m_6 = \overline{A}BC$	$M_6 = \overline{A}+B+C$
1	1	1	$m_7 = ABC$	$M_7 = A+B+C$

massimo M_i risulta sempre uguale ad **1**, salvo che per *la combinazione della riga* (2^n-1-i)*-esima per la quale risulta uguale a* ***0***.

Inoltre il prodotto di due termini minimi diversi è sempre ***0*** *mentre la somma di due termini massimi distinti è sempre uguale ad* ***1***.

Per quanto detto, visto che per qualsiasi combinazione delle variabili c'è sempre un termine minimo uguale ad **1,** avremo per la soma di tutti i termini minimi:

$$\sum_{i=0}^{2^n-1} m_i = 1 \tag{2.2.1}$$

Applicando i teoremi di De Morgan è immediato verificare che valgono le relazioni:

$$\overline{m_i} = M_{(2^n-1-i)} \tag{2.2.2}$$
$$\overline{M_i} = m_{(2^n-1-i)} \tag{2.2.3}$$

Applicando De Morgan alla (2.2.1), otterremo la relazione duale:

$$\prod_{i=0}^{2^n-1} M_i = 0\,. \tag{2.2.4}$$

2.3 Teorema fondamentale

Dalla tavola della verità 2.1.1 osserviamo che la (2.1.1) vale **1** per i valori di *A* e *B* per i quali sono uguali ad **1** i termini minimi m_0, m_1, m_2 pertanto possiamo scrivere:

$$f(A,B) = m_0 + m_1 + m_2 \qquad (2.3.1)$$

Se denotiamo con f_i il valore che assume una generica funzione in corrispondenza alla combinazione di variabili per cui m_i vale **1**, possiamo riscrivere la tabella 2.1.1 come tabella 2.3.1, dalla quale, generalizzando la (2.3.1), si deriva la rappresentazione generale, detta *forma canonica*, di una funzione booleana, F_k, di n variabili:

$$F_k = \sum_{i=0}^{2^n-1} f_i m_i \qquad (2.3.2)$$

Tabella 2.3.1

A	B	F_k
0	0	f_0
1	0	f_1
0	1	f_2
1	1	f_3

Si vede come il numero delle possibili funzioni booleane di n variabili sia $(2)^{2^n}$ tanti essendo i valori che può assumere il numero binario $(f_{2^n-1},\dots f_1, f_0)$, identificativo della specifica funzione, quindi nella (2.3.2) l'indice $k = 0,..,(2^{2^n} - 1)$ vale:

$$k = \sum_{i=0}^{2^n-1} f_i \bullet 2^i \qquad (2.3.3)$$

Nel caso di due variabili, *A* e *B*, si otterranno quindi 16 possibili funzioni, alcune delle quali banali, secondo la tabella 2.3.2.

Le funzioni XOR, NAND e NOR saranno illustrate in dettaglio nei paragrafi che seguono.

Si nota nella tabella una dualità fra le funzioni della prima metà con quelle della seconda, con simmetria speculare rispetto al centro della tabella.

Tabella 2.3.2

f_0	f_1	f_2	f_3	F_k
0	0	0	0	**0**
1	0	0	0	NOR
0	1	0	0	$A\overline{B}$
1	1	0	0	$\overline{B}$
0	0	1	0	$\overline{A}B$
1	0	1	0	$\overline{A}$
0	1	1	0	XOR
1	1	1	0	NAND
0	0	0	1	AND
1	0	0	1	$\overline{\text{XOR}}$
0	1	0	1	A
1	1	0	1	$A+\overline{B}$
0	0	1	1	B
1	0	1	1	$\overline{A}+B$
0	1	1	1	OR
1	1	1	1	**1**

Avendo trovato la (2.3.2) che ci da la forma canonica di una generica funzione combinatoria come somma di prodotti, *termini minimi*, vogliamo verificare se sia possible trovare una forma analoga che si basi invece sul prodotto di somme, *termini massimi*.

Considerando che la funzione complemento è vera dove la F_k è falsa, dalla tabella 2.3.1 risulta:

$$\overline{F}_k = \sum_{i=0}^{2^n-1} \overline{f}_i m_i \tag{2.3.4}$$

Applicando alla (2.3.4) i teoremi di De Morgan, ricordando la (2.2.2) si ottiene:

$$F_k = \prod_{i=0}^{2^n-1} (f_i + M_{2^n-1-i}) \tag{2.3.5}$$

Dalla lettura delle (2.3.2) e (2.3.5) si conclude (teorema fondamentale) che *la forma più generale di scrittura di una funzione booleana è ottenuta come somma di un opportuno numero di termini minimi oppure come prodotto di un opportuno numero di termini massimi.*

Applicando ad esempio la (2.3.2) e la (2.3.5) rispettivamente, alla tabella 2.1.1 si ottengono le due *rappresentazioni canoniche*:

$$f(A,B) = m_0 + m_1 + m_2 = \overline{A}\overline{B} + A\overline{B} + \overline{A}B$$

$$f(A,B) = M_0 = \overline{A} + \overline{B}$$

Si può scrivere per una qualsiasi funzione che:

$$F_k = \overline{F}_{{}^1k} \tag{2.3.6}$$

se si definisce:

$${}^1k = \sum_{i=0}^{2^n-1} \overline{f_i} \bullet 2^i \tag{2.3.7}$$

come il valore del complemento del numero binario dato dalla (2.3.3).

Essendo inoltre (vedi anche §8.1 per una trattazione più dettagliata):

$$k + {}^1k = 2^{2^n} - 1 \tag{2.3.8}$$

si può riscrivere la (2.3.6) come:

$$F_k = \overline{F}_{2^{2^n}-1-k} \tag{2.3.9}$$

che spiega la dualità, o simmetria speculare delle funzioni della tabella 2.3.2.

Naturalmente le rappresentazioni canoniche del tipo (2.3.2) e (2.3.5) non sono generalmente le rappresentazioni più semplici, in termini di numero di operatori necessari per la loro realizzazione circuitale. Nella pratica è molto importante trovare espressioni semplificate delle funzioni sia del tipo somma di prodotti, non necessariamente termini minimi, sia del tipo prodotto di somme, non necessariamente termini massimi.

2.4 Realizzazioni circuitali

Nel capitolo 1 sono stati introdotti i semplici circuiti logici a diodi del tipo ($A \bullet B$), detto AND, e ($A+B$), detto OR (fig.1.1.1), descritti dalle tabelle 1.1.1 e 1.1.2. In esse i diodi sono stati considerati elementi ideali che si comportano o come corti circuiti, con resistenza nulla, o come circuiti aperti, con resistenza infinita, trascurando la caduta di tensione sulla giunzione direttamente polarizzata.

Nei sistemi digitali è impensabile utilizzare componenti di tal tipo che farebbero degenerare il segnale dopo pochi livelli di elaborazione. In questo testo non tratteremo in dettaglio delle varie famiglie di circuiti digitali (DTL, TTL, ECL, CMOS, etc.), ma è importante mettere in evidenza che i componenti che si usano, anche se basati su tecnologie diverse, sono tutti caratterizzati dal fatto che rigenerano il segnale, dopo ogni operazione logica, utilizzando componenti attivi, almeno nello stadio d'uscita.

Poiché il più semplice amplificatore è realizzato con un transistore a emettitore comune, che inverta il segnale, le operazioni fondamentali che le porte digitali più comunemente utilizzate compiono, sono del tipo $\overline{A \bullet B}$, circuito NAND, e $\overline{A+B}$, circuito NOR.

Nella fig. 2.4.1 sono schematizzate due porte NAND di tipo DTL (Diode Transistor Logic) e di tipo TTL (Transistor Transistor Logic). Le porte DTL oggi non sono praticamente più utilizzate, sono state tuttavia "storicamente" molto importanti perché fino alla fine degli anni 60 sono state largamente impiegate, prima nella versione a componenti discreti e, successivamente, in versione integrata. La famiglia TTL per lungo tempo è stata la più diffusa, mentre oggi è largamente affermata la CMOS della quale si parlerà in dettaglio nel capitolo 4.

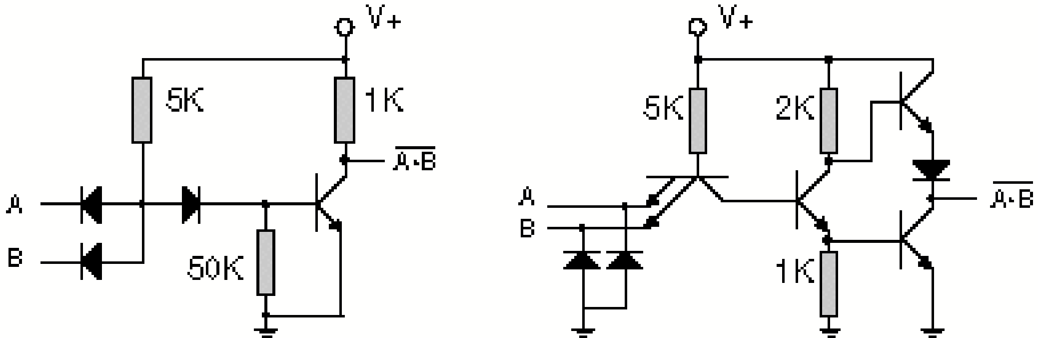

Figura 2.4.1

Le famiglie DTL e TTL sono fra loro compatibili essendo entrambe caratterizzate dal fatto che gli ingressi caricano i circuiti di pilotaggio solo al livello basso e che le uscite, al livello basso appunto, hanno la maggior possibilità di condurre corrente attraverso il transistor ad emettitore comune che lavora in saturazione. Lo stadio d'uscita DTL ha tempi di commutazione asimmetrici, essendo la costante di tempo del circuito d'uscita diversa per le transizioni positive, transistor che si spegne, e transizioni negative, transistor che satura. Lo stadio d'uscita dei TTL, detto *totem pole*, ha tempi di commutazione più simmetrici essendo entrambe le transizioni a bassa impedenza d'uscita.

Alcuni esercizi in classe sulla struttura delle porte TTL richiameranno alcuni concetti fondamentali sul funzionamento dei transitori bipolari in *interdizione* e *saturazione*.

È facile verificare che è possibile realizzare qualsiasi funzione booleana combinatoria disponendo o di operatori di tipo NAND o di tipo NOR.

Osserviamo anzitutto che entrambi i circuiti generano il complemento di una variabile quando questa sia presente in entrambi gli ingressi oppure, se si utilizza un solo ingresso, ponendo l'altro ad **1** per il NAND e a **0** per il NOR.

Ricordiamo inoltre che per il teorema di De Morgan si può dire che il NAND è l'OR delle variabili negate e che il NOR è l'AND delle variabili negate.

Consideriamo una semplice funzione, f, di due variabili e rappresentiamola utilizzando il teorema fondamentale sia come somma di termini minimi che prodotto di termini massimi:

$$f = AB + \overline{A}\,\overline{B} \tag{2.4.1}$$
$$f = (A + \overline{B})(\overline{A} + B) \tag{2.4.2}$$

Nella fig. 2.4.2 sono date le realizzazioni circuitali utilizzando sia NAND che NOR. Nei simboli utilizzati il circoletto indica inversione. Si vede che è immediata la realizzazione di funzioni booleane scritte con somma di prodotti utilizzando NAND mentre il NOR è adatto alla realizzazione del prodotto di somme.

Per quanto scritto sopra sul significato del circoletto, è possibile introdurre i simboli duali (De Morgan) degli operatori di NAND, NOR, AND e OR, (fig. 2.4.3), che evidenziano il tipo di operazione che si vuol realizzare rendendo più facile la lettura dello schema logico: lo schema di fig. 2.4.2 diviene allora quello di fig. 2.4.4.

Notiamo che la funzione (2.4.1) è la negazione dell'OR esclusivo, XOR, tra le variabili A e B, definito nella tabella 2.4.1 e che viene indicato anche col simbolo "$\oplus$".

Tabella 2.4.1

A	B	$A \oplus B$
0	0	0
1	0	1
0	1	1
1	1	0

La funzione XOR è vera se o l'una o l'altra, ma non entrambe, le variabili sono vere (circuito di disegualianza). La sua negazione è vera se le due variabili hanno lo stesso valore (circuito di uguaglianza).

Lo XOR e la sua negazione godono di alcune proprietà particolari che sono riassunte dalle seguenti relazioni:

$$A \oplus B = \overline{A} \oplus \overline{B} = A\overline{B} + \overline{A}B \qquad (2.4.3)$$
$$\overline{A \oplus B} = \overline{A} \oplus B = \overline{A}\,\overline{B} + AB \qquad (2.4.4)$$

I simboli circuitali di XOR e $\overline{\text{XOR}}$ sono dati in fig. 2.4.3.

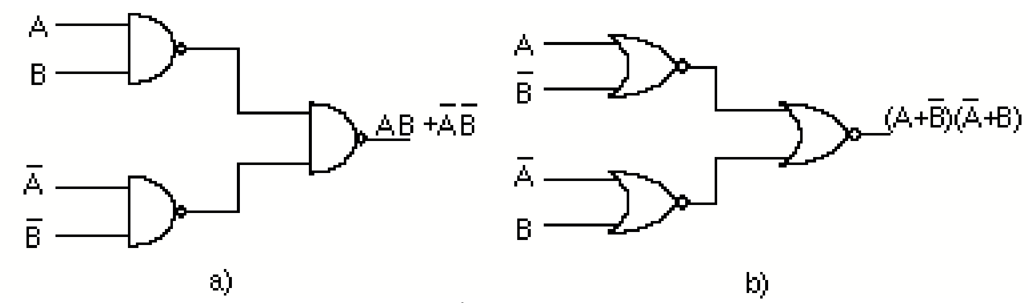

Figura 2.4.2

Possiamo verificare che è possibile realizzare le funzioni booleane non solo con NAND o NOR, ma anche con XOR , o $\overline{\text{XOR}}$, purché, insieme a questo operatore, sia definito anche quello di AND o di OR. Infatti le operazioni fondamentali si realizzano come segue:

$$\overline{A} = 1 \oplus A$$
$$A + B = \overline{\overline{A}\,\overline{B}} = 1 \oplus [(1 \oplus A)(1 \oplus B)]$$
$$AB = \overline{\overline{A} + \overline{B}} = 1 \oplus [(1 \oplus A) + (1 \oplus B)].$$

Nel caso dello $\overline{\text{XOR}}$, il complemento di A si ottiene facendo lo $\overline{\text{XOR}}$ con **0**.

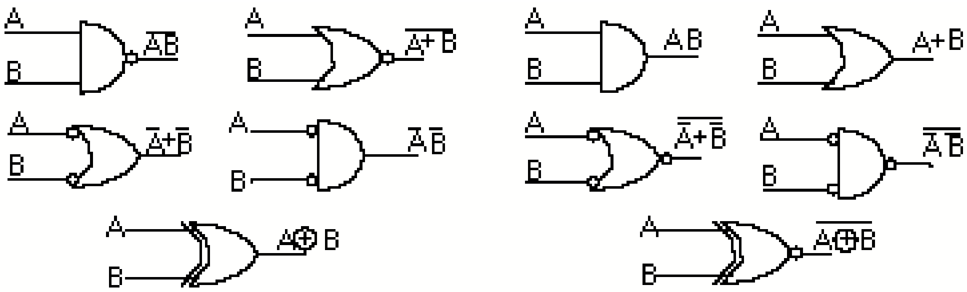

Figura 2.4.3

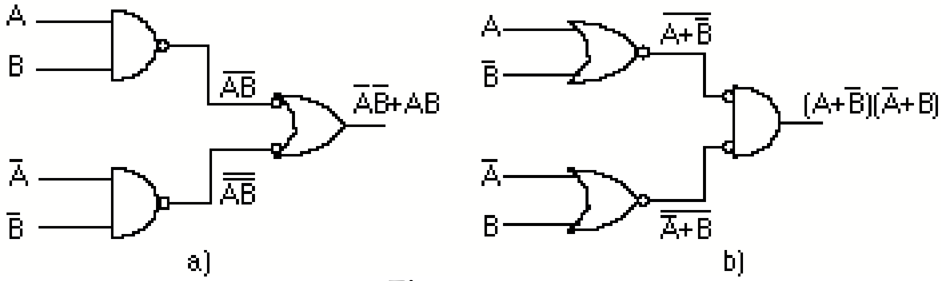

Figura 2.4.4

2.5 Il metodo grafico di semplificazione

La possibilità di realizzare logiche integrate con alta densità di operatori e grande numero di variabili, ha reso necessario sviluppare algoritmi di semplificazione che potessero essere facilmente utilizzati in programmi di calcolo. Il loro uso tuttavia non è giustificato quando si usino elementi discreti che permettono la realizzazione di circuiti con un numero limitato di operatori e variabili. Più in generale, operando con elementi discreti, è sempre conveniente ripartire il progetto in sottoinsiemi che risultino più facilmente sintetizzabili. In questa trattazione presentiamo un metodo grafico di semplificazione molto efficiente e semplice da usare fino ad un numero ragionevole (5 o 6) variabili.

Essenzialmente questo metodo si basa sul fatto che l'algebra di Boole si applica anche agli insiemi di punti: *si rappresenta quindi la funzione come unione di intersezioni di sottoinsiemi e poi si cerca di dare dell'insieme cosi individuato la rappresentazione più semplice possibile.*

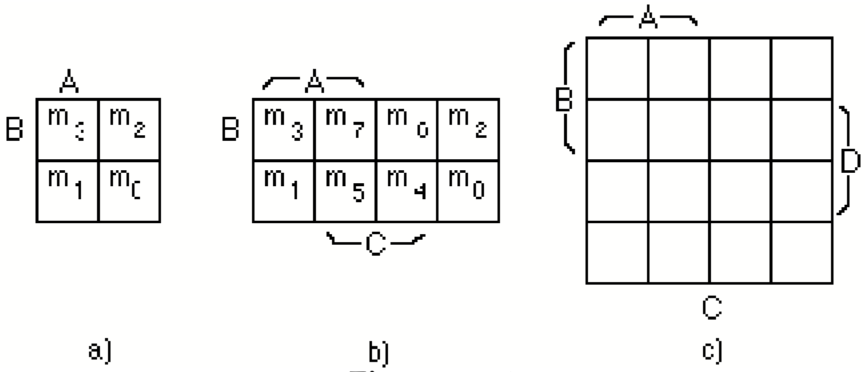

Figura 2.5.1

Nella fig. 1.1.4 abbiamo dato il modello per rappresentare funzioni di due variabili, e nella fig. 1.1.5 quello per tre variabili. Questi grafici sono stati introdotti da J. Venn, ma per la nostra applicazione è più conveniente utilizzare i grafici modificati secondo E. W. Veitch e M. Karnaugh. Nella fig. 2.5.1 sono date le mappe di Veitch per 2, 3, 4 variabili mentre in fig. 2.5.2 quelle corrispondenti di Karnaugh.

È evidente che la più piccola area distinguibile rappresenta un *termine minimo*, mentre il *termine massimo* è la più grande area che non coincide con l'intera mappa, ovvero l'intera mappa ove si escluda un sottoinsieme che rappresenti un termine minimo.

Gruppi di 2, 4, 8, etc. termini minimi connessi per un lato possono essere rappresentati in forma semplificata: più grande è il blocco descritto più semplice è la sua rappresentazione booleana.

I lati della mappa sono connessi come è evidente dalla fig. 2.5.3.

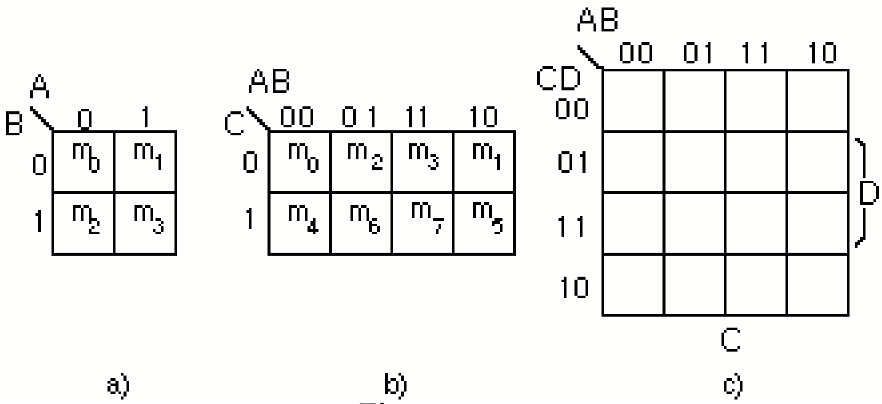

Figura 2.5.2

La semplificazione della funzione, rappresentata inizialmente da una scrittura del tipo (2.3.2), darà una somma di prodotti adatta ad essere realizzata da NAND.

Qualora si voglia una espressione semplificata, adatta ad essere realizzata con NOR (prodotto di somme), sarà opportuno rappresentare graficamente la funzione complemento, semplificarla e poi complementarla di nuovo. In fig. 2.5.4 sono illustrati alcuni esempi di semplificazioni.

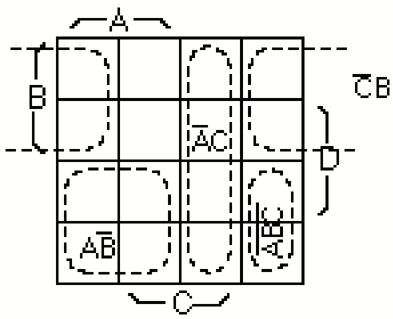

Figura 2.5.3

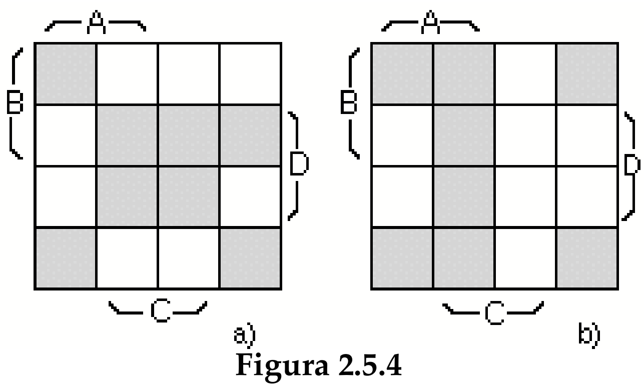

Figura 2.5.4

La funzione :

$$f = \overline{A}\,\overline{B}\,\overline{C}\,\overline{D} + A\overline{B}\,\overline{C}\,\overline{D} + AB\overline{C}\,\overline{D} + \overline{A}\,\overline{B}CD + \\ + A\overline{B}CD + \overline{A}BCD + ABCD + \overline{A}B\overline{C}D \tag{2.5.1}$$

aree grigie, insieme alla sua funzione $\overline{f}$, area bianche, sono riportate nel grafico di fig.2.5.4a che permette di ottenere la forma semplificata come somma di prodotti:

$$f = A\overline{C}\,\overline{D} + \overline{A}BD + \overline{B}\,\overline{C}\,\overline{D} + CD \tag{2.5.2}$$

oppure come prodotto di somme:

$$f = (\overline{A} + C + \overline{D})(A + \overline{B} + D)(B + C + \overline{D})(\overline{C} + D) \tag{2.5.3}$$

ottenuta dal complemento di:

$$\overline{f} = A\overline{C}D + \overline{A}B\overline{D} + \overline{B}\,\overline{C}D + C\overline{D} \tag{2.5.4}$$

ottenuta come rappresentazione più semplice delle aree bianche della fig.2.5.4a.

In modo analogo nella fig.2.5.4b è semplificata la:

$$f = \overline{A}\,\overline{B}\,\overline{C}\,\overline{D} + A\overline{B}\,\overline{C}\,\overline{D} + \overline{A}B\overline{C}\,\overline{D} + AB\overline{C}\,\overline{D} \\ + A\overline{B}C\overline{D} + ABC\overline{D} + A\overline{B}CD + ABCD \tag{2.5.5}$$

in modo da ottenere sia la rappresentazione:

$$f = \overline{C}\,\overline{D} + AC \tag{2.5.6}$$

che quella:

$$f = (A + \overline{C})(C + \overline{D}) \tag{2.5.7}$$

ottenuta dal complemento di:

$$\overline{f} = \overline{A}C + \overline{C}D \tag{2.5.8}$$

Si può verificare che una generica funzione di n variabili può essere sempre rappresentata con un numero di addendi non maggiore di 2^{n-1}. Basti pensare al grafico della funzione non semplificabile col massimo numero di addendi (termini minimi), che è del tipo a scacchiera.

2.6 Esercizi

a) Rappresentare, in forma semplificata, come prodotto di somme e somma di prodotti le seguenti funzioni:

$$f_1 = AB\overline{C} + ABC + \overline{A}BC + \overline{A}B\overline{C} + A\overline{B}C + \overline{A}\,\overline{B}C$$
$$f_2 = AB\overline{C} + A\overline{B}\,\overline{C} + ABC + \overline{A}BC + \overline{A}B\overline{C} + \overline{A}\,\overline{B}\,\overline{C}$$

b) Disegnare un circuito che realizzi lo XOR, in funzione di A e B entrambe vere, utilizzando solo 4 NAND.

c) Verificare che funzione realizza il circuito precedente se si sostituiscono i NAND con NOR.

d) Dimostrare che:
$$(A \oplus B) \oplus C = A \oplus (B \oplus C) = B \oplus (A \oplus C)$$

e) Dimostrare che :
$$(A \oplus \overline{B}) \oplus C = (A \oplus B) \oplus \overline{C}$$