Capitolo 2

PROGETTO LOGICO DELLE RETI COMBINATORIE

I Parte - Reti a due livelli di logica

2.1 - Introduzione

Avendo definito nel primo capitolo le porte logiche elementari AND, OR, NOT e le porte NAND, NOR, XOR da esse derivate, siamo ora in grado di precisare più dettagliatamente il significato del termine rete logica.

Una **rete logica** è un circuito che risulta dalla connessione dell'uscita di una o più porte logiche con l'ingresso di una o più altre porte logiche; a sua volta l'interconnessione di più reti logiche dà luogo ancora ad una rete logica nella quale ingressi e uscite esterni appartengono all'insieme degli ingressi e delle uscite delle reti componenti.

Come esempio, in Fig. 2.1 è schematizzata una rete logica ottenuta collegando quattro reti logiche più semplici.

Esistono due famiglie di reti logiche: le **reti combinatorie** e le **reti sequenziali**. Di entrambi i tipi può essere data sia una definizione funzionale che una definizione strutturale.

Funzionalmente una rete logica è **combinatoria** se le sue uscite dipendono in ogni istante solo dal valore dei segnali applicati agli ingressi in quell'istante; strutturalmente una rete è combinatoria se non contiene **cicli** nè **elementi di memoria**: si dice anche che una rete combinatoria è **aciclica** o **priva di memoria**. Nell'ambito delle reti logiche, un ciclo è un cammino che riporta un segnale dall'uscita di una porta (o da una delle uscite di una sottorete) al suo ingresso, direttamente o attraverso un numero qualsiasi di altre porte logiche (o di sottoreti). Per esempio nella Fig. 2.1 esistono due cicli marcati con 1 e 2 su tutti i rami che li compongono.

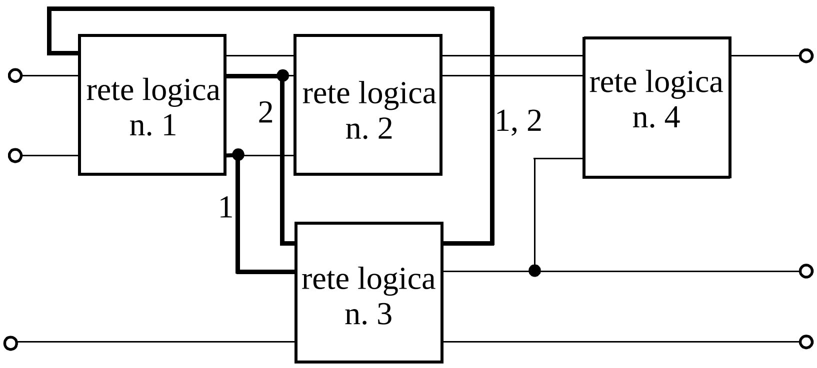


Fig. 2.1

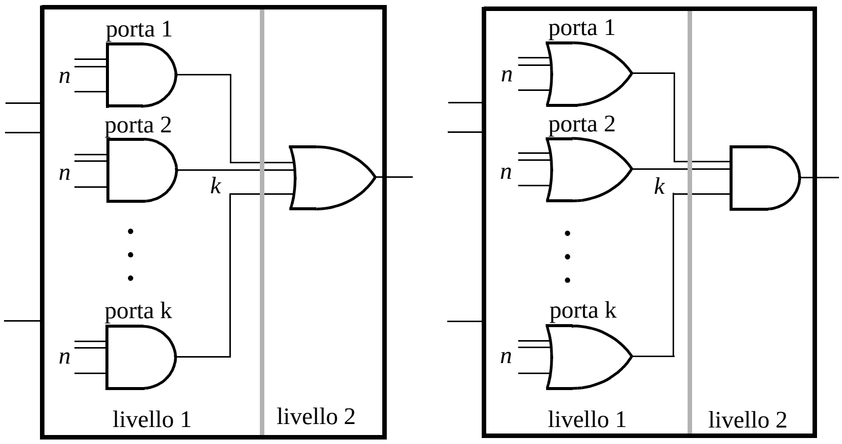
Viceversa una **rete sequenziale** è una rete per la quale non è possibile predire il valore delle uscite in un certo istante solo dall'esame dei valori degli ingressi in quell'istante, ma occorre tenere conto anche dei valori che si sono succeduti agli ingressi in tutti gli istanti passati. Strutturalmente una rete sequenziale contiene cicli ed è dotata di memoria.

Una rete combinatoria realizza una funzione logica e la sua struttura, intesa come topologia e numero di porte e di segnali interni tra le porte, dipende dall'espressione algebrica della funzione. Pertanto una stessa funzione può essere realizzata mediante più reti che differiscono tra di loro solo per numero di porte e di segnali (e quindi per complessità strutturale), ma non per la legge di trasferimento ingresso-uscita.

2.2 - Reti combinatorie a due livelli di logica

Tra tutte le reti che possono realizzare una data funzione logica, consideriamo prima di tutto quelle che derivano dalla forme canoniche SP e PS della funzione ed il cui schema generale è illustrato nella Fig. 2.2. Si tratta di reti con struttura regolare, appartenenti al gruppo delle **reti a due livelli di logica**: il primo livello è costituto solo da porte AND o OR in numero pari ai mintermini o ai maxtermini della forma canonica ed ogni porta ha tanti ingressi quante sono le variabili della funzione. Il secondo livello comprende una sola porta OR o AND rispettivamente per la rete SP e PS, con un numero di ingressi pari al numero dei mintermini o dei maxtermini, collegati alle uscite delle porte del primo livello.

Osserviamo che nel calcolo del numero dei livelli di logica normalmente non si considerano le porte NOT, poichè si suppone di disporre anche delle variabili complementate direttamente all’ingresso della rete; ciò è realistico dal momento che molto spesso il complemento di una variabile è disponibile senza aumento di costo all’uscita della rete che genera la variabile stessa.



Rete canonica SP   Rete canonica PS

Fig. 2.2

Reti a due livelli non si ottengono solo da espressioni canoniche, ma da qualsiasi espressione in somma di prodotti o in prodotto di somme. Di particolare interesse sono le reti derivanti da espressioni NAND o NOR, normali o no, poichè sono caratterizzate da una completa omogeneità strutturale che ne rende facile la realizzazione pratica. Per esempio se consideriamo l’espressione $x\bar{y}+z$, la sua forma NOR, ottenuta dalla forma PS, è $(x{\downarrow}y){\downarrow}(({\downarrow}y){\downarrow}z)$, a cui corrisponde la rete di Fig. 2.3; si noti che, come è stato detto sopra, la porta NOR utilizzata per generare la variabile $\bar{y}$ non rientra nel calcolo dei livelli di logica della rete.

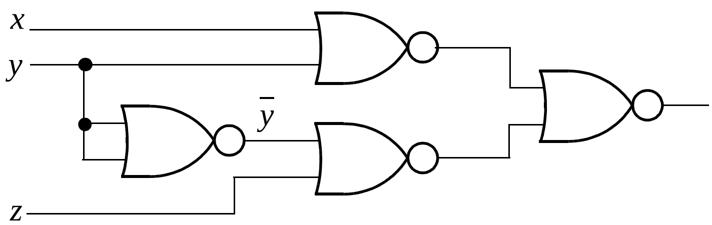


Fig. 2.3

L’interesse verso le reti a due livelli di logica deriva dal fatto che sono le **più veloci** che è possibile progettare per una data funzione. Infatti il ritardo, che dipende dal percorso interno più lungo e quindi dal numero massimo di porte che vengono attraversate su tutti i cammini, in queste reti è pari soltanto al doppio del ritardo di una singola porta.

2.3 - Il problema della sintesi delle reti combinatorie

La sintesi di una rete combinatoria consiste nel ricavarne la struttura, conoscendone la funzione ingresso-uscita sotto forma di tavola di verità o di espressione algebrica.

In generale il processo non porta ad una soluzione unica, poichè la stessa tavola di verità può dare luogo ad una molteplicità di reti, differenti per numero e tipo di porte e per collegamenti interni, essendo possibile operare riduzioni algebriche sull'espressione data o ricavata (in forma canonica) dalla tavola di verità; tutte queste reti, tuttavia, sono tra loro equivalenti agli effetti esterni.

La scelta di una rete piuttosto che un'altra viene operata sulla base di **criteri di ottimo**, che tendono a minimizzare parametri di costo ritenuti significativi. Per esempio si può ricorrere a metodi che portano a realizzare, per una data tavola di verità, la rete con il minimo numero di porte e di segnali e quindi più economica: per questo approccio, limitatamente alle reti a due livelli di logica, si conoscono metodi sistematici per la sintesi ottima.

Un altro approccio è determinato dalla tecnologia di fabbricazione dei circuiti integrati, in particolare di quelli a larga e larghissima scala di integrazione (LSI e VLSI). Infatti in tale tecnologia si assume come parametro di costo non tanto il numero delle porte, ma piuttosto l'estensione della superficie di silicio richiesta per realizzare su un singolo micrologico o "chip" sia gli elementi attivi, sia le piste di comunicazione, le quali hanno dimensione di complessità paragonabile a quella dei primi.

Prima di occuparci del problema della sintesi ottima, verifichiamo come dalla descrizione tabellare di una funzione sia possibile effettivamente sintetizzare più reti funzionalmente equivalenti. Consideriamo a questo scopo la funzione definita dalla seguente tavola di verità:

a	b	c	z
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

da questa possiamo derivare l'espressione canonica SP, la quale può essere ridotta, producendo una nuova espressione equivalente; inoltre dalla forma canonica si può ricavare la forma normale NAND, oppure si può trasformare l'espressione ridotta in una forma NOR, generando così altre rappresentazioni ancora della medesima funzione. A queste espressioni corrispondono altrettante reti topologicamente differenti, ma equivalenti dal punto di vista funzionale. Avremo così:

a) forma canonica SP $z = \bar{a}b\bar{c} + \bar{a}bc + abc$

b) espressione ridotta $z = b(\bar{a} + c)$

c) forma normale NAND $z = ((|a)|b(|c))|((|a)|b|c)|(a|b|c)$

d) forma NOR $z = (\downarrow b)\downarrow((\downarrow a)\downarrow c)$

2.4 - Criteri di minimizzazione delle funzioni combinatorie SP e PS

Come è stato detto nel paragrafo precedente, oltre alle reti che derivano dalle espressioni canoniche SP e PS, è possibile realizzare altre reti a due livelli di logica. La loro struttura deriva sempre da espressioni algebriche che sono nella forma di somma di prodotti o prodotto di somme e quindi è sempre del tipo schematizzato in Fig. 3.2, con le seguenti, peraltro ovvie, osservazioni:

a) non tutte le porte AND (OR) del primo livello hanno lo stesso numero di ingressi, ovvero non tutte generano mintermini (maxtermini) della funzione;

b) il loro numero non è direttamente in relazione con il numero di 1 (di 0) presenti nella tavola di verità della funzione.

Per esempio consideriamo la seguente espressione canonica

$$z = \bar{a}\bar{b}c + a\bar{b}c + abc$$

mediante l'algebra essa può essere semplificata, come è noto, nel modo seguente:

$$\bar{a}\bar{b}c + a\bar{b}c + abc = \bar{b}c(\bar{a} + a) + abc = \bar{b}c + abc = c(\bar{b} + ab) = \bar{b}c + ac$$

ottenendo ancora una forma SP, quindi a due livelli di logica ma non canonica, più semplice, per numero di porte e di segnali, di quella canonica da cui è derivata.

Questo esempio mette in evidenza il problema di come progettare reti **minime** a due livelli di logica. Tra le varie scelte che possono essere fatte per individuare un criterio di minimo, alcune delle quali sono state accennate nel paragrafo precedente, esamineremo quella (tradizionale) che mira alla minimizzazione di due parametri che influenzano la complessità, e quindi il costo, di una rete: 1) il numero dei porte logiche o gate $N_{g;}$ 2) il numero dei segnali N_d presenti nella rete.

Nel caso di reti derivate da espressioni SP, questi parametri dipendono dal numero dei termini prodotto N_p e dal numero di letterali N_v presenti nelle espressioni stesse secondo le relazioni:

$$N_g = N_p + 1$$
$$N_d = N_v + N_p$$

Analogamente per reti combinatorie costruite su espressioni PS valgono le seguenti relazioni:

$$N_g = N_s + 1$$
$$N_d = N_v + N_s$$

nelle quali N_s è il numero totale di termini somma ed N_v il numero di tutti i letterali presenti in esse.

In generale esiste un metodo algebrico per ottenere una somma minima, basato sull'uso ripetuto della proprietà associativa, del teorema della complementazione e del teorema del consenso. Per esempio data la funzione:

$$f = \bar{x}\bar{y}\bar{z} + \bar{x}\bar{y}z + x\bar{y}z + xyz$$

e considerato che valgono le uguaglianze:

$$\bar{x}\bar{y}\bar{z} + \bar{x}\bar{y}z = \bar{x}\bar{y}$$
$$x\bar{y}z + xyz = xy$$

si ottiene la somma minima:

$$f = \bar{x}\bar{y} + xz$$

Ragionamenti duali possono essere fatti relativamente a realizzazioni in prodotto di somme.

Tuttavia il procedimento algebrico è poco adatto per la riduzione a forma minima di espressioni con molti termini e con molte variabili, perchè tedioso, non sistematico e incline a errori. Una notevole semplificazione si ha con i metodi che saranno illustrati di seguito.

Osserviamo prima di tutto che una n-upla di variabili logiche, x_0, x_1,..., x_{n-1}, può essere interpretata come l'insieme delle coordinate dei vertici di un cubo di spigolo unitario nello spazio ad n dimensioni (ipercubo di grado n): infatti le coordinate dei vertici dell'ipercubo assumono in essi tutte le 2^n possibili combinazioni di 0 e 1, esattamente come possono fare n variabili logiche.

Inoltre ricordiamo che un prodotto logico di n variabili affermate o complementate (**prodotto completo** o **mintermine**) è una funzione che vale 1 solo per una tra le 2^n possibili combinazioni dei valori 0 e 1 delle variabili. Poichè questa combinazione costituisce il valore delle coordinate di un vertice dell'ipercubo di grado n, possiamo dire che un vertice è rappresentato da un prodotto completo.

Un prodotto logico che non comprenda tutte le n variabili rappresenta più vertici dell'ipercubo associato, e precisamente un suo sottocubo. Infatti sia P un prodotto di n-1 variabili e x_i la variabile mancante; poiché possiamo scrivere:

$$P = P \cdot (x_i + \bar{x}_i) = P \cdot x_i + P \cdot \bar{x}_i$$

P può essere espresso come somma di due prodotti completi, che differiscono solo per la variabile x_i, che compare affermata in uno e complementata nell'altro e conseguentemente può assumere due volte valore 1, una volta quando il prodotto $P \cdot x_i = 1$ ed una quando il prodotto $P \cdot \overline{x}_i = 1$. Esso rappresenta quindi due vertici adiacenti dell'ipercubo, ovvero un sottocubo di grado 1; si dice che P **copre** due vertici, di coordinate XX...1...X e XX...0...X rispettivamente, con X $\in$ {0,1}, o che rappresenta un sottocubo di 1 di grado 1.

Per esempio nel caso di tre variabili a, b, c il termine $\overline{a}c$ copre i punti 001 e 011. In generale un prodotto di k variabili ($k \leq n$) copre 2^{n-k} vertici che formano un ipercubo di 1 di grado n-k.

Con ragionamenti analoghi, ricordando che una somma di tutte le variabili (**somma completa** o **maxtermine**) è una funzione che vale 0 solo per una delle 2^n possibili combinazioni binarie delle variabili e poiché tale combinazione rappresenta il valore delle coordinate di un vertice dell'ipercubo associato, si può affermare che un vertice di un ipercubo è rappresentabile da una somma completa.

Una somma di n-1 variabili rappresenta un sottocubo di grado 1, ossia nell'ipercubo copre due vertici di coordinate XX...0...X e XX...1...X rispettivamente, con X $\in$ {0,1}, potendosi esprimere nella forma

$$S = (S + x_i) \cdot (S + \bar{x}_i)$$

che vale 0 due volte, una quando $S + x_i = 0$ ed una quando $S + \overline{x}_i = 0$. In generale una somma di k variabili ($k \leq n$) copre 2^{n-k} vertici che formano un sottocubo di 0 di grado n-k.

2.4.1 - Implicanti

Sia $f(x_0, x_1, ..., x_{n-1})$ una funzione di n variabili; diremo che un termine prodotto P è

un ***implicante*** *di f e scriveremo P → f se P copre almeno un vertice dell'ipercubo di grado n, nel quale f vale* 1, *e non copre alcun vertice in cui f vale* 0.

Poichè in generale f vale 1 anche in altri vertici non coperti da P, si può affermare che essa **include** o **copre** il suo implicante P. Dalla definizione discende che una funzione è esprimibile come somma dei suoi implicanti:

$$f = \sum_{i=1}^{...} P_i$$

e che ogni forma SP è una somma di implicanti di f. In particolare lo è la prima forma canonica, nella quale ogni mintermine è implicante di f.

Tra tutti gli implicanti di una funzione hanno particolare importanza gli **implicanti primi**, definiti nel modo seguente:

Un implicante P si dice primo se non esiste alcun altro implicante $P' \rightarrow f$ *tale che* $P \rightarrow P'$.

Questa definizione significa che P è un implicante primo se, per qualunque altro implicante P', esiste almeno un vertice, in cui la funzione vale 1, coperto da P ma non da P', ossia se accade che un qualunque altro implicante P' non copre P. In pratica un implicante primo è l'implicante che copre il maggior numero di vertici adiacenti in cui la funzione vale 1 e che non è coperto da alcun altro implicante.

Come esempio consideriamo la funzione definita dalla seguente tavola di verità:

a	b	c	z
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	0

I termini

$$\bar{a}\bar{b}\bar{c},\ \bar{a}\bar{b}c,\ \bar{a}bc,\ a\bar{b}\bar{c},\ a\bar{b}c$$

sono implicanti della funzione, coincidendo con i mintermini dell'espressione canonica SP. Analogamente sono implicanti i termini:

$$\bar{a}\bar{b},\ a\bar{b},\ \bar{a}c,\ \bar{b}\bar{c},\ \bar{b}c$$

ciascuno dei quali copre due vertici dell'ipercubo delle tre variabili, in cui la funzione vale 1, come pure il termine $\bar{b}$ che copre quattro vertici in cui f vale 1. Questo termine è anche implicante primo in quanto è l'unico implicante che copre quattro vertici e non è coperto da alcun altro implicante. L'implicante $\bar{a}c$ è a sua volta primo poichè è l'unico che copre il vertice di coordinate 011 in cui la funzione vale 1.

Si può dimostrare il seguente teorema:

Per ogni funzione f esiste almeno un sottoinsieme di implicanti primi $\mathcal{P} = \{P_1, P_2, ..., P_p\}$ *tale che la funzione può essere espressa nella forma* $f = \sum_{P_i \in \mathcal{P}} P_i$.

il quale asserisce che una funzione può essere espressa mediante la somma di tutti i suoi implicanti primi. Tuttavia non tutti gli implicanti primi sono sempre indispensabili per l'intera copertura, ma alcuni possono essere sostituiti dall'unione di altri implicanti primi; in particolare possono essere sostituiti da quelli che sono assolutamente necessari per coprire anche altri vertici in cui la funzione vale 1.

Per esempio se ab, $\overline{b}c$, ac sono tre implicanti primi di una funzione di tre variabili, l'implicante ac può essere sostituito dall'unione degli altri due i quali, oltre ad altri vertici, coprono tutti quelli coperti da ac. Ciò risulta evidente anche algebricamente, ricordando il teorema del consenso.

Da queste considerazioni sull'indispensabilità o meno degli implicanti primi, possiamo introdurre il concetto di **insieme non ridondante** di implicanti primi:

Un insieme $\boldsymbol{\mathcal{R}} = \{P_1, P_2, ..., P_r\}$ *è un insieme non ridondante di implicanti primi se una funzione f può essere espressa come somma di suoi implicanti primi, mentre non può essere espressa come somma di implicanti primi di un qualunque sottoinsieme di* $\boldsymbol{\mathcal{R}}$.

In modo informale possiamo dire che un insieme non ridondante è un insieme di implicanti primi avente la minima cardinalità sufficiente per coprire tutti i vertici dell'ipercubo in cui la funzione è 1, relativamente ad una particolare scelta di implicanti primi.

Una funzione espressa attraverso un insieme non ridondante dei suoi implicanti primi si dice in **forma SP prima non ridondante**. Tale forma può essere sempre trovata con una procedura che consiste nell'eliminare dall'insieme $\boldsymbol{\mathcal{R}}$ degli implicanti primi ogni implicante che possa essere sostituito dall'insieme di altri due e ripetendo tale operazione finchè è possibile.

Come già accennato può accadere che alcuni implicanti primi non possano essere assolutamente eliminati, perchè sono gli unici che coprono vertici in cui la funzione vale 1. Tali implicanti sono detti **primi essenziali**; più precisamente si può enunciare la seguente definizione:

Un implicante P è primo essenziale se esiste almeno un punto dove $P = 1$ *e* $f = 1$, *e non esiste alcun altro implicante primo P' tale che sia* $P' = 1$ *in quel punto.*

Tutti gli implicanti primi essenziali devono figurare nella forma SP prima non ridondante di una funzione.

Ad esempio per la funzione definita dalla tavola di verità della pagina precedente, l'implicante primo $\overline{a}c$ è essenziale poichè è l'unico a garantire la copertura del vertice 011.

2.4.2 - Implicati

Data una funzione di n variabili $f(x_0, x_1,..., x_{n-1})$ ed un termine S, somma di un sottoinsieme di quelle variabili, diremo che

*S è **implicato** di f, e scriveremo S ← f, se S copre almeno un vertice dell'ipercubo di grado n in cui f vale* 0 *e non copre alcun vertice in cui f vale* 1.

Analogamente a quanto visto per gli implicanti, la definizione precedente equivale a dire che la funzione f copre l'implicato S; discende inoltre da essa che una funzione è esprimibile come prodotto dei suoi implicati $f = \prod_{j=1}^{r} S_j$. Questa espressione è di tipo PS e si può perciò affermare che **ogni forma PS è un prodotto di implicati** di *f*. In particolare lo è la seconda forma canonica, nella quale ogni maxtermine è implicato di *f*.

In modo del tutto analogo a quanto detto nel paragrafo precedente si definiscono i concetti di **implicato primo** e di **insieme non ridondante** di implicati primi, come pure quello di **forma PS prima non ridondante** e infine il concetto di **implicato primo essenziale**.

Riprendiamo la funzione definita nell'esempio del paragrafo precedente; essa ammette come maxtermini le somme

$$a + \bar{b} + c, \;\; \bar{a} + \bar{b} + c, \;\; \bar{a} + \bar{b} + \bar{c}$$

le quali sono perciò implicati. Sono inoltre implicati i termini $\bar{b}+c$ e $\bar{a}+\bar{b}$ che coprono rispettivamente i vertici (010, 110) e (110, 111). Tali implicati sono anche primi ed essenziali perchè ciascuno copre il massimo numero di vertici in cui la funzione vale 0 e non è coperto da alcun altro implicato, ed inoltre ciascuno copre un vertice (010 e 111 rispettivamente) che non può essere coperto in alcun altro modo. Pertanto questi due implicati fanno parte della forma PS prima non ridondante che risulta:

$$f = (\bar{b} + c)(\bar{a} + \bar{b})$$

2.4.3 - Sintesi ottima di reti combinatorie a due livelli di logica

Avendo introdotto i concetti di forma SP e di forma PS prima non ridondante, possiamo dimostrare il seguente teorema:

Una forma SP o PS minima di una funzione è una forma SP o PS prima non ridondante

Consideriamo una funzione *f* di *n* variabili, espressa in una forma SP o PS a cui corrisponde una rete per la quale siano minimi i valori dei parametri N_g e N_d. Questa espressione è certamente una somma di implicanti o di implicati primi. Infatti se un implicante P_j (o un implicato S_j) in *f* non fosse primo, esisterebbe un implicante primo $P_j' \rightarrow f$ (o un implicato primo $S_j' \leftarrow f$) tale che $P_j \rightarrow P_j'$ (oppure $S_j \leftarrow S_j'$), il quale potrebbe essere sostituito a P_j o a S_j nell'espressione minima di *f*. Poichè P_j' o S_j' comprenderebbe un minore numero di variabili rispetto a P_j o S_j, ne deriverebbe una riduzione del valore di N_d, contro l'ipotesi che la forma data sia minima. Inoltre se la forma non fosse non ridondante, si potrebbe eliminare da essa almeno un termine, riducendo così il valore di N_g, nuovamente contro l'ipotesi che la forma sia minima.

Come conseguenza di questo teorema, la sintesi ottima di reti a due livelli di logica si riduce alla ricerca delle forme SP o PS prime non ridondanti delle funzioni logiche. La ricerca procede attraverso i seguenti passi:

a) determinazione dell'insieme $\mathcal{P}$ degli implicanti o dell'insieme $\mathcal{S}$ degli implicati primi della funzione;

b) selezione da $\mathcal{P}$ o da $\mathcal{S}$ di un insieme non ridondante di implicanti o di implicati primi, la cui somma o il cui prodotto copra la funzione ed il cui costo sia minimo.

È possibile definire metodi sistematici per trovare la forma minime di una funzione. Questi metodi sono il metodo delle **mappe di Karnaugh** e il metodo di **Quine-McCluskey** o metodo **tabellare**.

2.5 - Mappe di Karnaugh

Un ipercubo di grado n può essere rappresentato sotto forma di una tabella di 2^n caselle, ciascuna delle quali corrisponde ad un vertice dell'ipercubo ed è identificata di conseguenza dalle sue n coordinate binarie. Tabelle di questo genere sono chiamate **mappe di Karnaugh** e sono utilizzate per descrivere funzioni logiche, secondo il procedimento di seguito illustrato.

Date n variabili x_0, x_1, ..., x_{n-1}, la mappa di Karnaugh dell'ipercubo associato ad esse si costruisce facendo in modo che l'adiacenza geometrica tra due caselle, ovvero la comunanza di un lato, corrisponda alla loro adiacenza logica e viceversa, intendendo che due caselle sono *logicamente adiacenti* quando i loro identificatori differiscono per il valore di una sola coordinata. Le mappe per n = 2, 3, 4 sono illustrate nella Fig. 2.4, nella quale la numerazione decimale delle caselle è ottenuta dalle rispettive coordinate binarie:

x_1 \ x_2	0	1
0	0	1
1	2	3

n = 2

x_1 \ x_2x_3	00	01	11	10
0	0	1	3	2
1	4	5	7	6

n = 3

x_1x_2 \ x_3x_4	00	01	11	10
00	0	1	3	2
01	4	5	7	6
11	12	13	15	14
10	8	9	11	10

n = 4

Fig. 2.4

Affinchè l'adiacenza logica sia completa, la mappa deve essere pensata non distesa su un piano, come per convenienza viene disegnata e come è in effetti per n = 2, bensì su una superficie cilindrica per n = 3, toroidale per n = 4, su superfici più complesse per $n \geq 5$, in modo che i lati opposti coincidano.

Consideriamo ora una funzione f delle n variabili x_0, x_1, ..., $x_{n\text{-}1}$; essa può essere rappresentata sulla mappa di Karnaugh delle variabili, contrassegnando con 1 tutte le caselle le cui coordinate costituiscono un numero binario che corrisponde ad uno dei prodotti fondamentali (mintermini) inclusi nella funzione; con 0 (o lasciandole vuote) le rimanenti caselle. In questo modo si stabilisce una corrispondenza biunivoca tra le caselle della mappa contrassegnate con 1 e i mintermini dell'espressione canonica SP di f.

Per esempio la seguente funzione di tre variabili in forma canonica SP:

$$f = \bar{a}\bar{b}c + \bar{a}bc + abc$$

è descritta dalla mappa della Fig. 2.5:

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

Fig. 2.5

Anche funzioni in prodotto di somme possono essere rappresentate mediante mappe di Karnaugh, salvo contrassegnare con 0 le caselle di coordinate corrispondenti ad una delle somme complete (maxtermini) incluse nella funzione e con 1 (o lasciandole vuote) le altre. Così la stessa funzione dell'esempio precedente, espressa in forma canonica PS:

$$(a + b + c)(a + \bar{b} + c)(\bar{a} + b + c)(\bar{a} + b + \bar{c})(\bar{a} + \bar{b} + c)$$

è rappresentata dalla mappa:

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

Fig. 2.6

nella quale le cinque caselle contrassegnate con 0 sono in corrispondenza biunivoca con i maxtermini dell'espressione canonica.

Le mappe di Karnaugh trovano pratica applicazione fino ad un massimo di 5 o 6 variabili; tuttavia per questi valori conviene disegnare due e quattro mappe da quattro variabili rispettivamente, al fine di facilitare la ricerca delle adiacenze. Nel caso di cinque variabili una delle due mappe corrisponde a tutte le righe della tavola di verità per le quali la quinta variabile vale 0, l'altra a tutte le righe della tavola per le quali la quinta variabile vale 1; pertanto risultano adiacenti anche le caselle sull'una e l'altra mappa che si trovano nelle posizioni corrispondenti, ossia hanno identiche le altre quattro coordinate; per esempio sono adiacenti le caselle {1, 17}, {13, 29} e così via tutte le coppie i cui equivalenti decimali differiscono di 16:

$x_1 = 0$

x_2x_3 \ x_4x_5	00	01	11	10
00	0	1	3	2
01	4	5	7	6
11	12	13	15	14
10	8	9	11	10

$x_1 = 1$

x_2x_3 \ x_4x_5	00	01	11	10
00	16	17	19	18
01	20	21	23	22
11	28	29	31	30
10	24	25	27	26

Fig. 2.7

Ritorniamo all'esempio precedente e alle rappresentazioni sulle mappe delle forme canoniche della funzione data; per quanto detto a proposito di implicanti ed implicati, gli 1 e gli 0 delle due mappe corrispondono ad altrettanti mintermini e maxtermini della funzione. Consideriamo allora le seguenti forme ridotte SP e PS della funzione:

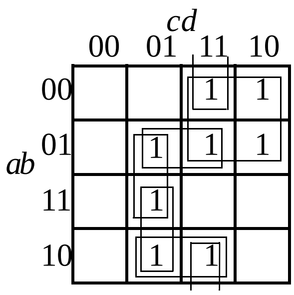
$$f_{SP} = \bar{a}c + bc \qquad f_{PS} = (\bar{a} + b)c$$

Nella prima il termine $\bar{a}c$ copre i punti 001 e 011 che corrispondono sulla mappa all'insieme delle due caselle della prima riga contrassegnate con 1; poichè $\bar{a}c$ è un implicante di f, si conclude che tale implicante è rappresentato da quelle due caselle. Analogamente il termine bc è un altro implicante della funzione e sulla mappa corrisponde al gruppo di due caselle di coordinate 011 e 111. Questi implicanti sono primi ed essenziali.

Analogamente nell'espressione PS il termine somma $\bar{a}+b$ è un implicato ed è rappresentato dall'insieme delle caselle di coordinate 100 e 101. Infine il termine c è pure un implicato e corrisponde alle caselle di coordinate 000, 100, 010, 110. Anche questi implicati sono primi ed essenziali, per cui le due espressioni f_{SP} e f_{PS} sono forme prime non ridondanti.

In generale un implicante che contiene k variabili, $1 \leq k \leq n$ è rappresentato su una mappa di Karnaugh di 2^n caselle dall'insieme di 2^{n-k} caselle adiacenti contenenti 1 della funzione: esso è dunque un ipercubo di 1 di grado n-k e la sua espressione è data dal prodotto delle variabili il cui valore rimane costante nel passare da una casella a quella adiacente, prese affermate o complementate a seconda che tale valore costante sia 1 oppure 0.

Un implicato di k variabili è rappresentato dall'insieme di 2^{n-k} caselle adiacenti contenenti 0 della funzione ed è un ipercubo di 0 di grado n-k, avente come espressione la somma delle variabili i cui valori rimangono costanti da una casella a quella adiacente, prese affermate o complementate a seconda che tale valore costante sia 0 oppure 1. Per esempio la funzione di quattro variabili rappresentata dalla mappa di Fig. 2.8 possiede sei implicanti primi, indicati sulla mappa con un rettangolo ed elencati a lato, dei quali $\bar{a}c$ è essenziale, mentre degli altri un sottoinsieme non ridondante è formato da $\{b\bar{c}d, a\bar{b}d\}$, che dà luogo all'espressione minima:



$\bar{a}c$, $\bar{a}bd$, $b\bar{c}d$, $a\bar{c}d$, $a\bar{b}d$, $\bar{b}cd$

Fig. 2.8

$$f = \bar{a}c + b\bar{c}d + a\bar{b}d$$

Nella Fig. 2.9 è mostrata la mappa di una funzione che possiede tre implicanti primi, di cui due sono essenziali ($\bar{a}\bar{b}$ e ac), mentre il terzo $\bar{b}c$ è coperto dall'unione degli altri due e può essere eliminato nella ricerca dell'espressione non ridondante; ciò è verificabile anche algebricamente, per il teorema del consenso; in altri termini si dice che $\bar{b}c$ *implica* la somma di $\bar{a}\bar{b}$ e *di ac*:

$$\bar{b}c \rightarrow \bar{a}\bar{b} + ac$$

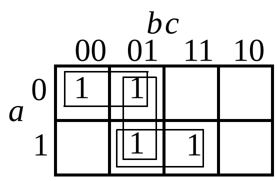


Fig. 2.9

Pertanto l'espressione $\bar{a}\bar{b} + ac$ costituisce la forma SP prima non ridondante, formata solo da implicanti essenziali.

Un altro esempio di funzione la cui mappa è coperta da implicanti essenziali è il seguente:

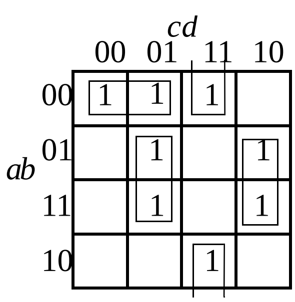


Fig. 2.10

Una stessa funzione può avere più forme prime non ridondanti, nelle quali figurano lo stesso numero di implicanti (o implicati) dello stesso ordine: pertanto la scelta di quale forma realizzare è una questione che dipende da considerazioni diverse da quelle finora esaminate. Un esempio mette in luce questo fatto: la mappa della Fig. 2.11 può essere coperta con le due espressioni:

$$f_1 = \bar{a}b + \bar{b}\bar{c} + bc \qquad f_2 = \bar{a}\bar{c} + \bar{b}\bar{c} + bc$$

Si tratta di due forme prime non ridondanti, in ciascuna delle quali compaiono gli implicanti essenziali $\bar{b}\bar{c}$ e bc, mentre la casella di 1 di coordinate 010 può essere coperta dagli implicanti primi $\bar{a}b$ o $\bar{a}\bar{c}$ indifferentemente.

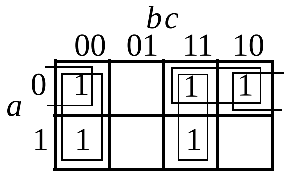


Fig. 2.11

Infine le due mappe seguenti rappresentano funzioni prive di implicanti e di implicati essenziali:

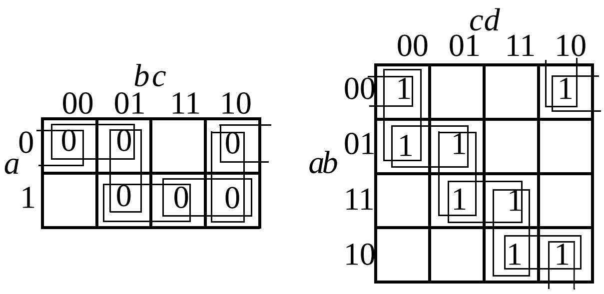


Fig. 2.12

Riassumendo, con le mappe di Karnaugh la sintesi ottima delle reti a due livelli avviene secondo il seguente procedimento:

a) si individuano dapprima gli implicanti (o gli implicati) primi essenziali che, come è stato detto, devono figurare nella forma minima.
b) Si effettua una ulteriore scelta in modo da coprire tutti gli 1 (o gli 0) rimasti scoperti dalla prima selezione. In questa seconda fase sono possibili più scelte, tutte plausibili almeno in base al criterio di ottimizzazione che è stato suggerito.

2.6 - Metodo tabellare di Quine-McCluskey

In alternativa al metodo delle mappe, che tra l'altro perde molta della sua utilità quando il numero di variabili è superiore a 6, ne è stato sviluppato un altro il quale, pur complesso, ha il vantaggio di poter trattare un numero qualunque di variabili; esso è noto come **metodo tabellare** o di Quine-McCluskey ed è articolato in due fasi:

a) determinazione di tutti gli implicanti (o implicati) primi mediante l'uso della **tabella degli implicanti** (o **degli implicati**);
b) scelta della forma minima utilizzando la **tabella di copertura**.

2.6.1 - Fase a)

Consideriamo per semplicità solo il caso della forma minima SP per una funzione logica di n variabili. Questa funzione vale 1 per un certo numero di combinazioni binarie dei valori delle sue variabili, siano h_1, h_2, ..., h_α. A ciascuna di esse attribuiamo un **peso** espresso dal numero di 1 presenti nella combinazione; per esempio la combinazione 01011 ha peso tre in quanto contiene tre 1.

Le combinazioni h_i, $1 \leq i \leq \alpha$, vengono disposte in una tabella, raggruppate in sezioni ordinate secondo il peso crescente; all'interno di una sezione le combinazioni sono in ordine numerico crescente. In tal modo coppie di combinazioni che si trovano in sezioni adiacenti differiscono esattamente per il valore di un bit nella stessa posizione e corrispondono a espressioni del tipo:

$$P \cdot x_i,\ P \cdot \overline{x}_i$$

nelle quali P è un prodotto di n-1 variabili esclusa la variabile x_i che vale 0 nella combinazione di una sezione e 1 in quella della sezione adiacente. Fra queste combinazioni viene eseguita un'operazione, detta **prodotto di fusione**, che consiste nel sostituire ad esse una nuova combinazione avente gli stessi bit in tutte le posizioni, tranne in quella relativa ai bit che differiscono nelle combinazioni di partenza, dove non viene specificato alcun valore. Questo processo equivale ad applicare la proprietà associativa e quella della complementazione:

$$P \cdot x_i + P \cdot \overline{x}_i = P$$

ovvero a sostituire ai due termini prodotto un unico prodotto P che possiede la x_i variabile in meno.

Il procedimento è applicato sistematicamente tra ogni combinazione di una sezione e tutte quelle della sezione successiva, arrivando a costruire una seconda tabella che possiede una sezione in meno rispetto alla tabella di partenza e nella quale, sezione per sezione, il risultato delle fusioni è costituito da combinazioni con un bit in meno rispetto alle combinazioni che hanno fuso, nella posizione in cui è avvenuta la fusione. Tale assenza è contrassegnata con “–”. Vengono inoltre contrassegnate nella tabella originaria tutte le coppie di combinazioni che hanno fuso.

Ripetendo il procedimento sulla seconda tabella, se ne genera una terza e così via fino ad arrivare ad una situazione nella quale non sono più possibili prodotti di fusione. I prodotti che corrispondono a tutte le combinazioni che nelle varie tabelle non hanno fuso, ovvero quelle che non sono state contrassegnate, più il prodotto relativo all’ultima fusione effettuata costituiscono l’insieme degli implicanti primi.

Sia data per esempio la funzione di quattro variabili $f = \sum_4 (0, 1, 5, 6, 7, 8, 10, 12, 13, 15)$.

In base ad essa costruiamo la tabella degli implicanti, che ha un numero di righe pari al numero di 1 della funzione, ovvero otto; ogni riga è contrassegnata con l’equivalente decimale della combinazione binaria delle variabili per la quale la funzione vale 1 e le dieci combinazioni sono ripartite in sezioni secondo il peso e all’interno della sezione sono in ordine crescente (tabella a):

	a	*b*	*c*	*d*	
0	1	0	0	0	√
1	0	0	0	1	√
8	1	0	0	0	√
5	0	1	0	1	√
6	0	1	1	0	√
10	1	0	1	0	√
12	1	1	0	0	√
7	0	1	1	1	√
13	1	1	0	1	√
15	1	1	1	1	√

a)

	a	*b*	*c*	*d*	
0,1	0	0	0	–	
0,8	–	0	0	0	
1,5	0	–	0	1	
8,10	1	0	–	0	
8,12	1	–	0	0	
5,7	0	1	–	1	√
5,13	–	1	0	1	√
6,7	0	1	1	–	
12,13	1	1	0	–	
7,15	–	1	1	1	√
13,15	1	1	–	1	√

b)

	a	*b*	*c*	*d*	
5,7,13,15	–	1	–	1	

c)

Eseguiti tutti i possibili prodotti di fusione, la tabella si trasforma in quella indicata in b) e da questa con un ulteriore passo si ottiene la tabella finale c).

A questo punto non sono più possibili fusioni e le combinazioni che nelle tre tabelle non hanno fuso corrispondono ad altrettanti implicanti primi, che pertanto sono:

$$\bar{a}\bar{b}\bar{c},\ \bar{b}\bar{c}\bar{d},\ \bar{a}\bar{c}d,\ a\bar{b}\bar{d},\ a\bar{c}\bar{d},\ \bar{a}bc,\ ab\bar{c},\ bd$$

Prima di proseguire osserviamo un fenomeno che si presenta molto frequentemente nel procedimento descritto. Esso consiste nel fatto che ogni combinazione contenente due o più posizioni vuote può essere generata dalla fusione di più coppie di righe. Questo è il caso della combinazione –1–1 ottenuta al secondo passo dalla fusione della riga (5,7) con (13,15) e della riga (5,13) con (7,15). Un modo di evitare questa ripetizione è quello di considerare solo quelle coppie di combinazioni in sezioni adiacenti i cui equivalenti decimali formano una successione crescente: così la coppia ((5,7)–(13,15)) è da considerare, mentre non lo è la coppia ((5,13)–(7,15)). Tuttavia bisogna ricordarsi di contrassegnare come partecipanti a fusione tutte quelle righe i cui equivalenti decimali figurano in qualche riga prodotta dalla fusione di altre coppie; così le righe (5,13) e (7,15), pur non essendo prese in considerazione, devono essere contrassegnate perchè i loro equivalenti figurano nella riga (5,7,13,15) della terza tabella. L'altra possibilità consiste nell'effettuare comunque la fusione, ma non scriverla se già presente in qualche riga di una tabella.

2.6.2 - Fase b)

In questa fase viene costruita un'espressione SP prima non ridondante per la forma minima della funzione, esaminando le relazioni tra gli implicanti primi ed i mintermini della funzione stessa. In questo processo si fa uso di una tabella, detta **tabella di copertura**, in cui ogni riga corrisponde ad un implicante ed ogni colonna ad un mintermine ossia ad un 1. L'incrocio della riga i con la colonna j viene contrassegnato con ×, se l'implicante relativo a quella riga copre l'1 relativo alla colonna.

Una forma SP minima viene ricavata dalla tabella di copertura come somma degli implicanti corrispondenti ad un insieme $\mathcal{R}$ di righe della tabella tale che:

- $\mathcal{R}$ copre la tabella, ovvero per ogni colonna della tabella esiste almeno una riga che presenta un contrassegno in quella colonna;
- fra tutti i possibili insiemi che possono essere scelti in accordo al punto precedente, $\mathcal{R}$ è quello di minima cardinalità.

Per poter determinare l'insieme $\mathcal{R}$ si applicano due criteri detti **essenzialità** e **dominanza**.

Una riga è detta **essenziale** se esiste almeno una colonna che ha un contrassegno solo in corrispospondenza di essa; l'implicante corrispondente è essenziale e deve figurare nella forma minima SP della funzione.

Una riga i **domina** una riga j se oltre agli stessi contrassegni della riga j presenta almeno un contrassegno in più.

Illustriamo il procedimento attraverso l'esempio fatto per la fase a). La tabella di copertura si presenta come mostrato di seguito, avendo indicato, per comodità, con le lettere maiuscole dalla A alla H nell'ordine gli implicanti elencati al termine della fase a).

Il primo passo consiste nel determinare, se ve ne sono, tutte le colonne che hanno una sola ×: per esse esiste un solo implicante che copre l'1 corrispondente; tale implicante, come detto, è essenziale e la riga associata prende il nome di riga **essenziale**, mentre la colonna è detta **distinta**.

Nella tabella dell'esempio le colonne 6, 10 e 15 sono distinte e le righe D, F ed H sono essenziali; i rispettivi implicanti devono figurare perciò in qualunque forma SP minima.

	0	1	5	6	7	8	10	12	13	15
A	×	×								
B	×					×				
C		×	×							
D						×	×			
E						×		×		
F				×	×					
G								×	×	
H			×		×				×	×

Tutte le colonne distinte e le righe essenziali vengono cancellate dalla tabella. La cancellazione di una riga essenziale comporta la cancellazione anche delle colonne che, pur non essendo distinte, hanno una × in quella riga: infatti l'1 di ciascuna di quelle colonne è coperto almeno dall'implicante della riga essenziale e non occorre ricercare per esso altre coperture.

Nella tabella data la cancellazione della riga D comporta la cancellazione anche delle colonne 8 e 10 (distinta); la cancellazione della riga F determina la cancellazione anche delle colonne 6 (distinta) e 7; infine la cancellazione della riga H fa sparire le colonne 5, 7, 13 e 15 (distinta).

Al termine di questo passo, per la forma minima possiamo scrivere:

$$f \;=\; D + F + H + \ldots \;=\; a\bar{b}\bar{d} + \bar{a}bc + bd + \ldots$$

mentre la tabella di copertura si riduce alla seguente:

	0	1	12
A	×	×	
B	×		
C		×	
E			×
G			×

In essa si osserva che le righe B e C sono dominate dalla riga A e possono essere eliminate, lasciando inalterate le colonne 0 ed 1 dove hanno rispettivamente un contrassegno; la tabella si riduce ulteriormente alla seguente:

	0	1	12
A	×	×	
E			×
G			×

In questo modo la riga *A* diventa essenziale (secondaria) per le colonne distinte 0 ed 1, pertanto l'implicante associato dovrà fare parte della forma minima; la cancellazione di *A* comporta la cancellazione anche delle colonne 0 ed 1 e la tabella si riduce alla colonna 12 e alle righe *E* ed *H*. Queste sono uguali, avendo gli stessi contrassegni, per cui una delle due è superflua e può essere rimossa; quella rimasta diventa allora essenziale e l'implicante associato dovrà entrare a far parte della forma minima: se si cancella la riga *E*, rimane la *H* come essenziale secondaria, viceversa se si cancella la riga *H*, rimane come essenziale la *E* e in definitiva il procedimento porta ad individuare due forme minime:

$$f = D + F + H + A + \begin{cases} E \\ G \end{cases} = a\bar{b}\bar{d} + \bar{a}bc + bd + \bar{a}\bar{b}\bar{c} + \begin{cases} a\bar{c}\bar{d} \\ ab\bar{c} \end{cases}$$

2.6.3 - Tabelle cicliche

Può accadere che nella tabella di copertura esistano o rimangano righe e colonne non vuote che non possono essere cancellate nè invocando l'essenzialità nè la dominanza: una tabella di questo genere è detta **ciclica** e può presentarsi quando esistono più forme prime non ridondanti.

Il procedimento più comunemente seguito per risolvere una tabella ciclica è noto come **metodo dell'albero binario** ed è articolato in più passi, in corrispondenza di ciascuno dei quali sono eseguite due fasi applicate alla riga della tabella di copertura incompleta che ha più contrassegni, fino a completare una di due forme non ridondanti possibili.

fase a) Si considera la riga selezionata come **essenziale** e si semplifica la tabella cancellando quella riga e le colonne coperte da essa; quindi si procede alla ulteriore riduzione applicando iterativamente i passi illustrati nel paragrafo precedente.

fase b) Si considera la riga selezionata come **eliminabile**, quindi la si rimuove lasciando inalterate le colonne coperte da essa; dopo la sua rimozione si semplifica la tabella secondo il consueto procedimento.

E` ovvio che durante l'esecuzione di ogni fase può risultare un'ulteriore tabella ciclica, per cui la procedura è di fatto ricorsiva e può portare ad una molteplicità di forma minime, tra le quali verrà scelta quella di costo minimo, per minimo numero di termini prodotto e/o di letterali distinti.

2.7 - Funzioni non completamente specificate

Accade di frequente che il valore di una funzione logica non debba essere specificato per tutte le combinazioni di valori delle sue variabili ma solo per alcune, e ciò per vari motivi:

a) in relazione ad un dato problema, certe combinazioni di valori non si presentano mai, in quanto codificano informazione **non richiesta**;

b) il valore dell'uscita non ha **interesse** per certe combinazioni, in rapporto allo specifico problema, quindi è inutile generarle.

c) alcune combinazioni sono **proibite** perchè corrispondono a valori dei segnali di ingresso alla rete che realizza la funzione i quali, per ragioni tecnologiche, non devono essere mai applicati.

Supponiamo per esempio che una funzione di quattro variabili debba riconoscere le cifre decimali pari e quelle dispari, assumendo valore 0 per le prime e valore 1 per le seconde. È chiaro che in questo problema non è significativo il valore assunto dalla funzione per le combinazioni di valori delle sue variabili corrispondenti agli interi da 10 a 15, che pure con quattro variabili binarie sono rappresentabili.

In tutti questi casi si dice, sia pure impropriamente, che siamo in presenza di **funzioni incomplete** o **non completamente specificate**; le combinazioni di valori delle variabili per le quali il valore della funzione non è specificato si dicono **condizioni di indifferenza** (**don't care**) e ad esse corrispondono nel metodo delle mappe di Karnaugh caselle contrassegnate in qualche modo, ad esempio con $\times$.

Ai fini della sintesi ottima di una rete che realizzi una funzione incompleta *f*, siano *h* le condizioni di indifferenza della funzione; esistono allora 2^h modi distinti di specificare tali valori come 1 o come 0 e quindi 2^h distinte funzioni complete f' che coprono la funzione *f*. Tra esse si sceglie, per la sintesi, quella che consente di realizzare la rete di costo minimo.

In pratica sulla mappa di Karnaugh della funzione *f* tutti i valori $\times$ vengono specificati come 1 o come 0, a seconda che ci si riferisca ad una mappa di 1 o di 0. Quindi si cerca una forma prima non ridondante, tenendo conto che in essa **non** devono essere presenti implicanti o implicati primi che non coprono 1 o 0 appartenenti anche alla funzione *f*.

Per esempio per il problema del rilevatore della parità di una cifra decimale avremo la seguente mappa:

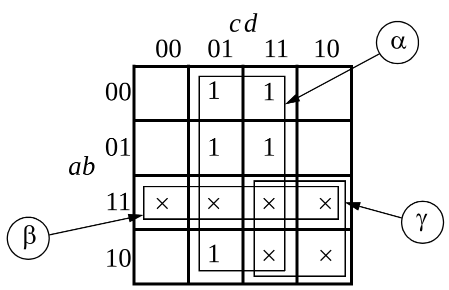


Fig. 2.13

Specificando le $\times$ come 1 si ottiene una nuova mappa, relativa ad una delle funzioni complete f', che ammette la forma prima non ridondante $\{\alpha, \beta, \gamma\}$ e che copre la funzione f.

Poichè gli implicanti β e γ coprono punti di f' ma non di f, la forma prima non ridondante di questa sarà costituita solo dall'implicante α, per cui avremo $f = d$, come peraltro era da aspettarsi.

La presenza di condizioni di indifferenza viene simbolicamente indicata, nella forma SP, con la notazione $\Delta_n(l_1, l_2, ..., l_\gamma)$, dove gli interi l_i per $i = 1, ..., \gamma\text{-}1$, $\gamma \in \{1, 2, ..., 2^n\}$ indicano le caselle della mappa di Karnaugh o equivalentemente le righe della tabella di verità per le quali non è significativo il valore della funzione ed n è il numero delle variabili.

2.8 - Reti a più uscite

2.8.1 - Definizione del problema

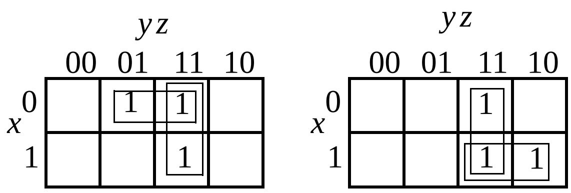
Spesso le reti combinatorie hanno molteplici uscite ed esistono perciò più funzioni, ciascuna delle quali controlla una delle uscite della rete, correlandone i valori a quelli presenti in ogni istante sugli ingressi ed espressi dai valori di n variabili $x_0, x_1, ..., x_{n\text{-}1}$.

In linea di principio la sintesi di una rete a più uscite potrebbe consistere nella sintesi separata delle reti per le singole funzioni, con gli ingressi a comune: sarebbe una soluzione molto semplice ma certamente non sempre ottima. Per rendercene conto basta esaminare due semplici casi.

Sia data una rete a due uscite, definite dalle funzioni:

$$f_1 = \sum_3 (1, 3, 7) \qquad f_2 = \sum_3 (3, 6, 7)$$

la sintesi separata di esse porta alle seguenti espressioni minime derivabili dalle mappe di Fig. 2.14:



$f_1 = \bar{x}z+yz$ $\qquad$ $f_2 = yz+xy$

Fig. 2.14

cui corrisponde la seguente rete (Fig. 2.15):

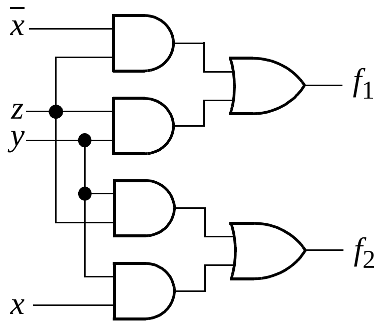


Fig. 2.15

È ovvio che questa non è la soluzione ottimale, dal momento che l'implicante primo *yz* è comune alle forme minime di entrambe le funzioni e può quindi essere sintetizzato una sola volta, mediante una sola porta AND la cui uscita è collegata all'ingresso delle porte OR di uscita di f_1 ed f_2. Si passa in questo modo alla rete di Fig. 2.16, che è ottima:

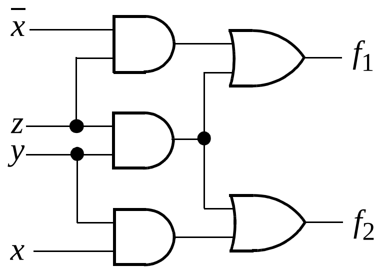


Fig. 2.16

Tuttavia il procedimento che è stato seguito in questo esempio non è sempre così chiaramente intuibile, come risulta da un altro caso. Le due funzioni siano infatti:

$$f_1 = \sum_3 (1, 3, 7) \qquad f_2 = \sum_3 (2, 6, 7)$$

alle quali corrispondono due reti minime ottenute dalle mappe della figura seguente:

x \ yz	00	01	11	10
0		1	1	
1			1	

$f_1 = \bar{x}z + yz$

x \ yz	00	01	11	10
0				1
1			1	1

$f_2 = xy + y\bar{z}$

Fig. 2.17

Esse non hanno implicanti primi in comune per cui sembrerebbe che l'unica realizzazione possibile fosse la rete di Fig. 2.18:

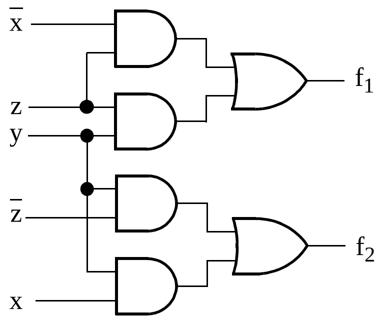


Fig. 2.18

Invece si può osservare che in entrambe le funzioni esiste un 1 nella stessa posizione 111 ≡ 7, per cui se esso viene coperto con il mintermine *xyz*, le due funzioni acquistano l'espressione:

$$f_1 = \bar{x}z + xyz \qquad f_2 = y\bar{z} + xyz$$

alle quali corrisponde la rete di Fig. 2.19 che, pur non essendo minima, è più economica della precedente avendo meno porte:

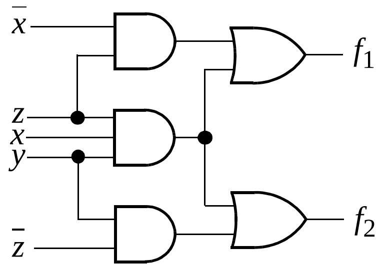


Fig. 2.19

Questi esempi mettono in luce che nella sintesi di reti a più uscite occorre tenere conto di due fatti importanti, sui quali deve essere basata una tecnica di progetto specifica:

1) non è pensabile di sintetizzare le singole funzioni separatamente;
2) non sempre la definizione di un insieme non ridondante di implicanti (o di implicati) primi porta ad una soluzione globale migliore, anzi non è sufficiente considerare solo implicanti (o implicati) primi.

Per risolvere il problema osserviamo che la forma più generale di una rete combinatoria in forma SP ad *m* uscite è quella nella quale vi sono porte AND collegate ad una sola porta OR di uscita, porte AND collegate a due porte OR di uscita, ecc., ed infine porte AND collegate ad *m* porte OR di uscita. Inoltre possono esservi ingressi della rete collegati direttamente alle porte OR di uscita: essi possono essere considerati equivalenti a porte AND con gli ingressi uguali (Fig. 2.20).

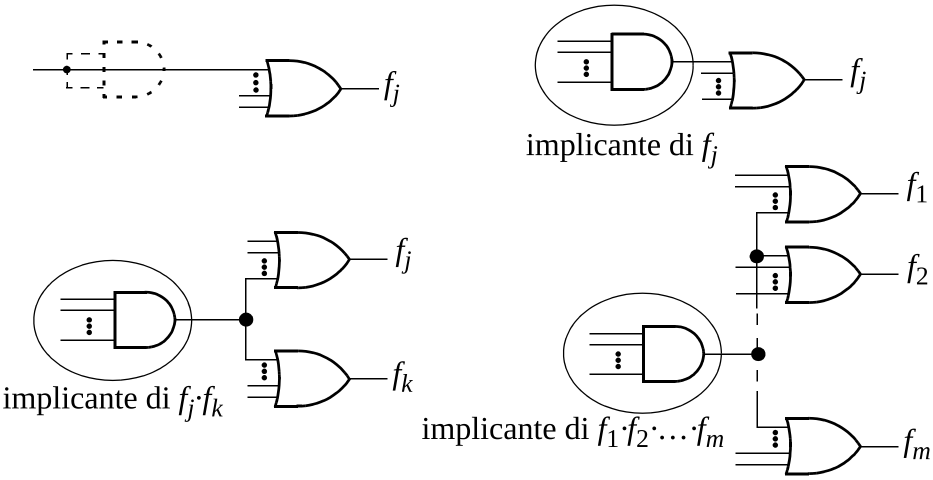


Fig. 2.20

Se una porta AND è collegata ad una sola porta di uscita *j* della rete, esiste una ed una sola combinazione di valori dei suoi ingressi per la quale la sua uscita vale 1 e di conseguenza vale 1 anche l'uscita della rete e la funzione f_j associata. Questa porta realizza pertanto un implicante della funzione f_j e se la rete è minima si tratta certamente di un implicante primo.

Consideriamo ora una porta AND che alimenta due porte OR di uscita *j* e *k*. Per una ed una sola combinazione di valori degli ingressi la sua uscita è 1 e di conseguenza sono 1 le uscite delle porte OR ad essa collegate e in definitiva le funzioni f_j ed f_k associate. Viceversa se l'uscita della porta AND è 0, può darsi che f_j ed f_k continuino ad essere 1, oppure può darsi che una o l'altra (o eventualmente entrambe) siano 0: se ciò accade sicuramente è 0 la funzione prodotto $f_j{\cdot}f_k$, pur non essendo necessario che siano 0 tanto f_j quanto f_k. Pertanto il prodotto delle variabili che sono di ingresso a quella porta AND implica la funzione prodotto $f_j{\cdot}f_k$, che vale 1 se e solo se f_j ed f_k valgono 1. Se la rete è globalmente minima, la rimozione di un ingresso dalla porta AND che alimenta f_j ed f_k può alterare il valore di f_j o di f_k (o di entrambe), ma sicuramente altera la funzione $f_j{\cdot}f_k$: il prodotto generato dalla porta AND è un implicante primo di $f_j{\cdot}f_k$, mentre non è detto che lo sia anche delle funzioni componenti.

Questa situazione si presenta nel secondo esempio, nel quale il prodotto *xyz* è implicante primo di $f_1{\cdot}f_2$, in quanto si ha:

$$f_1 \cdot f_2 \;=\; \sum_3(7) \;=\; (\bar{x}z + yz)(yz + xy) \;=\; xyz$$

ma non di f_1 nè di f_2, dal momento che nelle loro espressioni è possibile eliminare rispettivamente *x* o *z* dal prodotto *xyz* senza alterare il loro valore:

$$f_1 = xyz + \bar{x}z = z(xy + \bar{x}) = z(y + \bar{x}) = yz + \bar{x}z$$
$$f_2 = xyz + y\bar{z} = y(xz + \bar{z}) = y(x + \bar{z}) = xy + y\bar{z}$$

Estendendo queste considerazioni ad una porta AND collegata ad r uscite, $f_1, f_2, ..., f_r$, si può affermare che essa corrisponde ad un implicante primo della funzione $f_1 \cdot f_2 \cdot ... \cdot f_r$ e, in generale, possiamo concludere che l'insieme di tutte le porte di ingresso della rete corrisponde all'insieme degli implicanti primi di $f_1, f_2, ..., f_m, f_1 \cdot f_2, \dots, f_1 \cdot f_m, \dots, f_{m-1} \cdot f_m, \dots, f_1 \cdot f_2 \cdot ... \cdot f_m$.

Per il progetto ottimo di una rete a più uscite è dunque necessario generare gli implicanti primi delle singole funzioni di uscita e di tutte le funzioni prodotto a 2 a 2, a 3 a 3, ..., ad m ad m delle funzioni date. Questi implicanti si dicono **implicanti primi multipli**; più precisamente un implicante multiplo risponde alla seguente definizione:

Un implicante primo multiplo di un insieme di m funzioni $f_1, f_2, ..., f_m$ *è un prodotto di variabili di ingresso che costituisce* 1) *un implicante primo di una delle funzioni date* f_j*, oppure* 2) *un implicante primo di una delle funzioni prodotto* $f_{j_1} \cdot f_{j_2} \cdot \dots \cdot f_{j_k}$, $j_k = 1, \dots, m$, $k \le m$.

La scelta di quali implicanti primi multipli siano da selezionare per la forma minima viene effettuata con i metodi già visti per le reti ad una uscita, adattati al caso specifico; considerazioni analoghe valgono per le reti a più uscite in forma PS, cioè aventi porte AND come porte di uscita.

2.8.2 - Sintesi delle reti a più uscite con le mappe di Karnaugh

La sintesi viene effettuata in due passi. Il primo passo consiste nel determinare gli implicanti primi multipli, utilizzando le mappe di Karnaugh per le funzioni date e per le funzioni prodotto di esse a 2 a 2, a 3 a 3, ..., ad m ad m.

Il secondo passo consiste nel ricavare una scelta opportuna di implicanti primi multipli, ricordando che in essa devono figurare necessariamente gli implicanti essenziali, la cui definizione è estesa rispetto a quella data a suo tempo nel seguente modo:

Siano $f_1, f_2, ..., f_m$ *m funzioni delle medesime variabili* $x_0, x_1, \dots, x_{n-1}$ *e* $\{P_1, P_2, ..., P_r\}$ *il corrispondente insieme di implicanti primi multipli. Un implicante primo multiplo della funzione* f_i *o di qualunque prodotto di funzioni comprendente* f_i*, è detto* ***essenziale*** *per la funzione* f_i *se e solo se è l'unico che include un prodotto fondamentale di* f_i*; quel prodotto fondamentale è detto* ***distinto*** *e quell'implicante deve figurare nell'espressione di* f_i.

Per esempio, date le seguenti funzioni ed il loro prodotto :

$$f_1 = \sum_4 (0, 1, 5, 8, 11) \qquad f_2 = \sum_4 (1, 5, 9, 10, 11, 13, 15) \qquad f_1 f_2 = \sum_4 (1, 5, 11)$$

è immediato determinarne sulle relative mappe (Fig. 2.21) gli implicanti primi multipli, elencati di seguito: f_1: $\{A \equiv \overline{a}\ \overline{b}\ \overline{c}$, $B \equiv \overline{a}\ c\ \overline{d}$, $C \equiv \overline{a}\ \overline{c}\ \overline{d}$, $D \equiv a\ \overline{b}\ \overline{d}\}$; f_2: $\{E \equiv\ \overline{c}\ d$, $F \equiv a\ d$, $F \equiv a\ \overline{b}\ c\}$; $f_1 f_2$: $\{B \equiv \overline{a}\ c\ \overline{d}$, $H \equiv a\ \overline{b}\ c\ \overline{d}\}$.

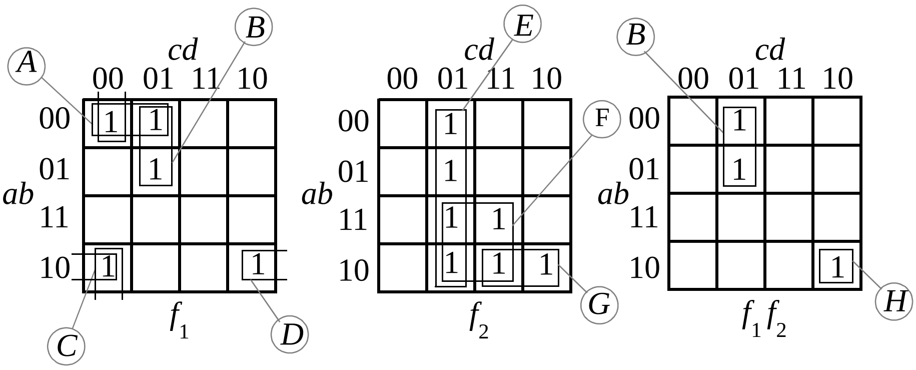


Fig. 2.21

Per la funzione f_1, l'implicante primo multiplo B è essenziale, in quanto copre l'1 in posizione 5 e come tale deve essere selezionato per la forma minima. B è implicante primo multiplo anche della funzione $f_1 f_2$, ma non della funzione f_2, per la quale gli 1 in posizione 1 e 5 sono coperti dall'implicante E; quindi per f_2 si dovrebbe selezionare E e non B. D'altra parte però l'implicante F è essenziale per f_2, mentre non lo è per f_1 e tanto meno per $f_1 f_2$; la selezione di F per f_2 garantisce la copertura degli 1 di questa funzione in posizione 9, 11, 13 e pertanto conviene rinunciare ad E e coprire gli altri due 1 in posizione 1 e 5 con B, con un risparmio complessivo di un implicante. L'1 in posizione 10 dovrebbe essere coperto dagli implicanti primi multipli essenziali D e G per f_1 ed f_2 rispettivamente; tuttavia al loro posto conviene selezionare l'implicante primo multiplo essenziale H della funzione $f_1 f_2$, essendo già stato coperto per f_2 l'1 in posizione 11con la scelta di F e potendo coprire l'1 in posizione 8 per f_1 con l'implicante primo multiplo C in alternativa ad A. La scelta di C rende superfluo l'implicante A di f_1. Pertanto le espressioni di f_1 ed f_2 sono:

$$f_1 = B+C+H \equiv \overline{a}\ \overline{c}\ d+\overline{b}\ \overline{c}\ \overline{d}+a\ \overline{b}\ c\ \overline{d}$$
$$f_2 = B+F+H \equiv \overline{a}\ \overline{c}\ d+ad+a\ \overline{b}\ c\ \overline{d}$$

ESERCIZI

1) Delle seguenti funzioni trovare:
   a) gli implicanti o gli implicati primi;
   b) gli implicanti o gli implicati primi essenziali;
   c) tutte le espressioni minime SP o PS;
   d) le forme NAND e NOR:

$$f = \sum_{3}(0, 1, 2, 5, 6, 7)$$
$$f = \sum_{4}(1, 3, 4, 6, 7, 9, 12, 13)$$
$$f = \sum_{5}(0, 1, 2, 3, 5, 7, 8, 9, 10, 16, 19, 21, 23, 24, 25, 26, 27, 28, 29, 30, 31)$$
$$f = \prod_{5}(0, 1, 2, 5, 7, 10, 11, 14, 15, 16, 17, 18, 21, 23, 26, 27, 30, 31)$$
$$f = \prod_{4}(0, 1, 2, 4, 5, 7, 9, 10, 11, 13, 15)$$

2) Minimizzare le seguenti funzioni con il metodo tabellare di Quine-McCluskey:

$$f = \Sigma_4(1, 2, 4, 5, 7, 8, 10, 13, 14, 15)$$
$$f = \Sigma_4(1, 2, 7, 8, 9, 10, 13, 15)$$

3) Minimizzare le seguenti funzioni, specificando in modo appropriato le condizioni di indifferenza:

$$f = \Sigma_4(1, 2, 7, 8, 10, 13, 15) + \Delta(0, 5, 11, 12)$$
$$f = \Pi_4(1, 9, 14, 15) \cdot \Delta(4, 5, 6, 7)$$
$$f = \Sigma_4(1, 4, 9, 14, 15) + \Delta(2, 5, 6, 11, 12)$$

4) Sintetizzare in forma minima le seguenti reti a più uscite:

$$f_1 = \Sigma_3(0, 1, 3, 5, 7)$$
$$f_2 = \Sigma_3(1, 2, 3, 4, 6)$$
$$f_3 = \Sigma_3(5, 6, 7)$$

$$f_1 = \Sigma_4(0, 2, 4, 5, 7, 10, 15)$$
$$f_2 = \Sigma_4(0, 1, 3, 4, 5, 6, 14)$$
$$f_3 = \Sigma_4(1, 3, 5, 7, 8, 12, 15)$$

$$f_1 = \Sigma_4(0, 2, 3, 4, 8, 12, 13)$$
$$f_2 = \Sigma_4(2, 3, 5, 6, 7, 8, 13, 14, 15)$$

$$f_1 = \Pi_4(2, 3, 8, 12, 13)$$
$$f_2 = \Pi_4(2, 3, 5, 6, 7, 8, 12, 13, 14, 15)$$

$f_1 = \Sigma_4\,(1, 3, 5, 8, 9, 11, 13)$
$f_2 = \Sigma_4\,(3, 5, 7, 8, 13, 14, 15)$

$$f_1 = \sum_4 (0, 4, 5, 6, 7, 8, 11, 15)$$
$$f_2 = \sum_4 (4, 5, 7, 14, 15)$$
$$f_3 = \sum (1, 6, 7, 9, 13, 14, 15)$$

5) Trovare la rete NAND ottima per le funzioni:
$f_1 = \Pi_4(2, 3, 8, 12, 13)$
$f_2 = \Pi_4(2, 3, 5, 6, 7, 8, 12, 13, 14, 15)$

6) Realizzare un circuito combinatorio che riceve in ingresso un numero intero di cinque bit e riconosce se si tratta di un numero primo; il circuito deve essere in forma minima SP e in forma NOR.

7) Data la funzione descritta dalla seguente mappa, si realizzi la relativa rete logica:

		a = 0, *d,e*				*a* = 1, *d,e*			
		00	01	11	10	00	01	11	10
b,c	00	1	0	1	1	1	1	0	1
	01	×	1	1	1	1	0	×	1
	11	×	1	×	0	×	0	×	×
	10	×	×	×	1	1	×	0	×

8) Una rete combinatoria ha quattro ingressi *a*, *b*, *c*, *d* ed una uscita *z*, la quale dipende dagli ingressi secondo la funzione:

$$f_1 = \bar{a}\bar{c} + a\bar{b} + bcd + ac$$

Si vuole trasformare la rete in un'altra con gli stessi ingressi e funzione di uscita $f_2 = b\bar{c} + bd + \bar{b}c\bar{d}$, mediante una seconda rete avente funzione di uscita $F(a, b, c, d)$. Trovare l'espressione minima di F e specificare quale operazione logica deve essere realizzata tra f_1 ed F per ottenere il risultato voluto.

9) Data la funzione $f_1 = \Sigma_4(0, 1, 3, 4, 5, 7, 8, 9, 11, 15)$, realizzare la funzione $f = \Sigma(0, 1, 2, 3, 4, 5, 7, 10, 11, 14, 15)$, combinandola con un'opportuna funzione f_2.

10) Da un gruppo di quattro soldati devono essere selezionati 2 o 3 volontari per una missione pericolosa. Interrogati dal comandante circa le condizioni per la loro disponibilità, i quattro danno le seguenti risposte:

soldato A: è disposto ad andare con B e D, con B e C, con C e D oppure con B da solo, ma non con C né con D da soli;

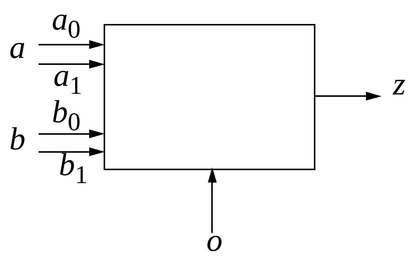
soldato B: può andare con C o con D, ma non con entrambi, a meno che non sia presente anche A;

soldato C: non va d'accordo con D, tranne quando è presente anche A, mentre va sempre d'accordo con B;

soldato D: non vuole né A né C da soli, ma è disposto a fare gruppo con entrambi; con B va d'accordo e accetta anche di far parte di un gruppo di tre soldati in cui oltre a B ci sia A.

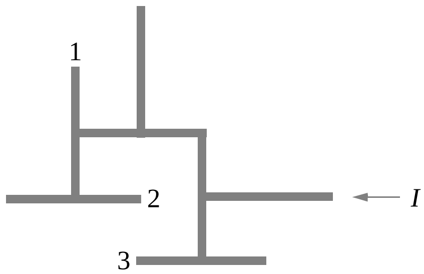
Studiare una rete logica NOR di minimo costo per aiutare il povero comandante a capire quali gruppi potrebbe formare.

11) Un circuito combinatorio ha due ingressi di due bit ciascuno $a(a_0, a_1)$ e $b(b_0, b_1)$, un ingresso o di un bit ed una uscita z, pure di un bit.



Se o vale 1, l'uscita assume valore 1 se $a > b$, altrimenti vale 0. Se o vale 0, l'uscita vale 1 se $a < b$. Si definisca il circuito attraverso la tavola di verità della funzione $z = f(o, a_0, a_1, b_0, b_1)$ e se ne dia una realizzazione in forma minima PS.

12) Progettare una rete logica che simula il comportamento di un visitatore in un labirinto come quello illustrato in figura, nel quale I indica il punto di ingresso ed i numeri 1, 2 e 3 indicano le uscite possibili:



Nota: indicare con variabili logiche le alternative di fronte alle quali si trova il visitatore in corrispondenza dei bivi.

13) Realizzare un circuito “maggioritario”, ossia un circuito che rileva se tra n ingressi, con n dispari, almeno $(n+1)/2$ sono uguali; fare il caso di $n = 5$.

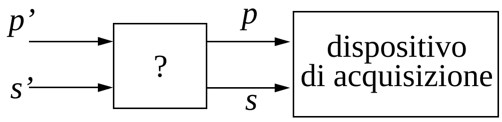
14) Realizzare la funzione $f = \bar{a}\bar{c}\bar{d} + b\bar{c}\bar{d} + a\bar{b}d + a\bar{b}c$ come combinanzione di tre funzioni f_1, f_2, f_3 secondo l’espressione $f = f_1 f_2 + f_3$, sapendo che $f_1 = a \oplus b$.

15) Data la funzione $f = \Sigma_4(0, 1 ,2, 3, 9, 11, 13, 15)$, si realizzi una rete per la funzione $F = \Pi_4(1, 3, 4, 6, 8, 10, 11, 12)$ secondo la forma $F = f \bullet f_1$, essendo f_1 una funzione da definirsi ed il simbolo • un opportuno operatore logico da individuarsi come una delle 16 funzioni di 2 variabili.

16) Realizzare un comparatore per coppie di numeri di due bit in complemento a due, il quale dà in uscita l’informazione se i due numeri sono uguali, oppure qual è il più grande.

17) Un dispositivo di acquisizione dati possiede due ingressi di controllo p ed s per la selezione del modo di funzionamento; purtroppo il costruttore non ha tenuto conto del fatto che quando entrambi i controlli sono 1, il dispositivo non può funzionare correttamente, mentre se entrambi sono 0 nessuna operazione ha luogo. Si realizzi una rete impedisce il verificarsi della situazione di errore, per qualunque coppia di valori

assegnata ai due segnali di controllo (v. figura). Proporre più di una soluzione.



18) Realizzare un circuito combinatorio che converte una cifra decimale dal codice 8, 4, 2, 1 al codice 8, 4, -2, -1, sapendo che le cifre con le quali si caratterizza un codice posizionale sono i pesi da attribuire ai singoli bit della rappresentazione in tale codice, a seconda della loro posizione.
Esempio: il numero 6 è rappresentato dalla consueta stringa 0110 nel codice 8, 4, 2, 1 e dalla stringa 1010 nel codice 8, 4, -2, -1

19) Realizzare un circuito che converte una cifra decimale dal codice BCD (Binary Coded Decimal, ovvero decimale codificato in binario) al codice Gray; questo può essere costruito dal corrispondente codice BCD secondo le relazioni:

$$g_i = b_i \oplus b_{i+1}, \quad i = 0, \ldots, 3 \quad \text{e} \quad b_4 = 0$$

20) Realizzare un circuito incrementatore modulo 10, ossia un circuito con quattro ingressi e quattro uscite che funziona nel modo seguente. Quando all'ingresso si presenta una delle configurazioni binarie corrispondenti ai numeri decimali tra 0 e 8 compresi, le uscite assumono la configurazione binaria che corrisponde all'intero successivo di quello in ingresso, mentre se all'ingresso si presenta una delle configurazioni che rappresentano uno degli interi tra 9 e 15, le uscite assumono la configurazione degli interi tra 0 e 6.

21) Ripetere l'esercizio 20), considerando come condizioni di indifferenza tutti gli ingressi tra 9 e 15.

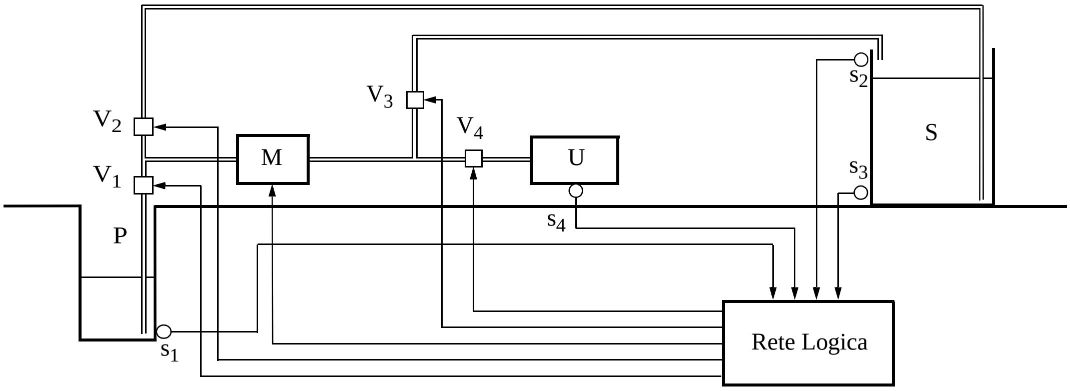
22) Realizzare un circuito che calcola il numero X mod m, per interi X di 5 bit e per una scelta arbitraria di $m \geq 2$.

23) Un circuito combinatorio implementa la funzione $f = a + bc + \bar{a}\bar{c}\bar{d}$, ma durante il collaudo si scopre che per gli stati di ingresso 0011, 0111 e 0010 si verificano anomalie di funzionamento; realizzare un circuito da applicare agli ingressi del circuito difettoso, in modo da evitarne il malfunzionamento.

24) Una stringa binaria di 4 bit può contenere una cifra decimale codificata secondo la seguente tabella;Nessun'altra configurazione è significativa. Si progetti un circuito in forma minima NOR che riceve in ingresso la stringa binaria e dà in uscita 01 se essa contiene 0, 2, 6, 7, oppure 8; 10 se contiene un'altra configurazione legittima; 11 altrimenti.

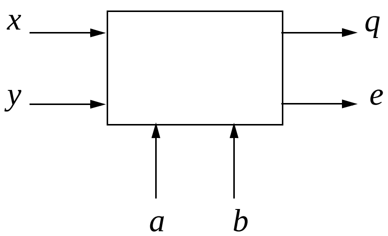
0	0011
1	0100
2	0101
3	0111
4	1000
5	1001
6	1011
7	1100
8	1101
9	1111

25) Un impianto idraulico è costituito da un pozzo P, una pompa M, un serbatoio S ed un utilizzatore U. La pompa è usata sia per prelevare l'acqua dal pozzo e mandarla nel serbatoio o all'utilizzatore, sia per mandare l'acqua dal serbatoio all'utilizzatore. Per questo nell'impianto si trovano le elettrovalvole V_1 e V_2, che scelgono l'afflusso all'entrata della pompa (V_1 aperta, V_2 chiusa: acqua dal pozzo; V_1 chiusa, V_2 aperta: acqua dal serbatoio), e le elettrovalvole V_3 e V_4 all'uscita della pompa (V_3 aperta, V_4 chiusa: acqua al serbatoio; V_3 chiusa, V_4 aperta: acqua all'utilizzatore). Inoltre quattro sensori rilevano la condizione dell'impianto e precisamente s_1 avverte se nel pozzo c'è acqua, s_2 avverte se il serbatoio è pieno, s_3 se è vuoto ed infine s_4 rileva se l'utilizzatore richiede acqua.



Se il serbatoio non è pieno, nel pozzo c'è acqua e l'utilizzatore non la richiede, la pompa è in funzione e le valvole sono orientate in modo da fare affluire l'acqua al serbatoio. Invece se l'utilizzatore richiede acqua ed il serbatoio ne contiene, la pompa è in funzione con le valvole disposte in modo da alimentare U dal serbatoio; se questo è vuoto, l'utilizzatore riceve l'acqua dal pozzo. In ogni altro caso la pompa è ferma e le valvole si trovano in qualsiasi posizione che eviti il pericolo di riversare acqua dal serbatoio al pozzo. Si progetti una rete logica in forma minima che ha come ingressi i segnali binari generati dai sensori e come uscite cinque segnali binari per il controllo delle valvole e del motore della pompa.

26) Sintetizzare mediante porte NOR la rete combinatoria di figura in modo che l'uscita *q* abbia il valore indicato in tabella per le varie combinazioni degli ingressi di controllo *a* e *b*, e l'uscita *e* assuma valore 1 in caso di errore per traboccamento, operazione impossibile, ecc.



a	b	q
0	0	(x+y) mod 2
0	1	(x-y) mod 2
1	0	xy mod 2
1	1	x/y mod 2