**2.19** Trovare la forma minima SP per la funzione

$$\Sigma_6(8, 15, 22, 23, 30, 31, 56, 57, 58, 59, 60, 61, 62, 63).$$

**2.20** Costruire un confrontatore (vedi esercizio 2.16) per numeri di tre bit.

## 2.7 Funzioni incompletamente specificate

Nella descrizione del comportamento di una rete combinatoria non sempre si specifica il valore delle uscite per tutte le configurazioni di ingresso. Ciò può dipendere da diversi motivi; anzitutto gli $n$ segnali di ingresso codificano una informazione che non necessariamente è significativa in $2^n$ casi diversi, tanti quante sono le configurazioni *distinte* di ingresso: vi saranno dunque alcune configurazioni che non vengono mai presentate agli ingressi (un esempio in tal senso è esaminato qui sotto). Vi sono poi situazioni in cui limitazioni tecnologiche vietano l'applicazione di alcuni gruppi di segnali. Vi sono infine situazioni in cui il valore delle uscite non è significativo per il problema trattato, e non è quindi necessario specificarlo.

Tutti questi casi conducono a funzioni di uscita *incomplete*, o *incompletamente specificate*: le configurazioni di ingresso per cui la funzione non è specificata si dicono *condizioni di indifferenza*.

Poniamo per esempio che si debba realizzare una rete combinatoria con 4 morsetti di ingresso $x_3, x_2, x_1, x_0$ e un morsetto di uscita $z$: i quattro segnali di ingresso rappresentano, in codice binario naturale, una cifra decimale $N$ ($x_3$ è il bit più significativo); in uscita deve essere $z=1$ se $N$ è uguale a 0, 2, 4, 8, $z=0$ altrimenti. La funzione $z$ è evidentemente incompleta, poiché non è specificata per le configurazioni di ingresso corrispondenti ai numeri decimali tra 10 e 15: la $z$ è dunque rappresentata come indicato nella figura 2.21, ove le crocette corrispondono alle condizioni di indifferenza.

Si tratta ora di costruire una rete che realizzi la $z$. A questo proposito notiamo subito che esistono funzioni incomplete, ma *non esistono reti incomplete*, poiché una rete produce un'uscita per qualunque configurazione di ingresso le si presenti. Il problema della costruzione di una rete che realizzi una funzione incompleta $f$ si pone dunque come: costruire una rete che realizzi una funzione (completa) $f'$, tale che $f'=f$ per tutte le configurazioni di ingresso ove $f$ è specificata. Ne segue che due reti che realizzino due funzioni $f'$ e $f''$ diverse, ma coinci-

| $x_3x_2 \backslash x_1x_0$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 1 | 0 | 0 | 1 |
| 01 | 1 | 0 | 0 | 0 |
| 11 | X | X | X | X |
| 10 | 1 | 0 | X | X |

**Figura 2.21**
Una funzione incompleta.

denti con $f$ nelle combinazioni in cui $f$ è specificata, sono due soluzioni al problema.

Se le condizioni di indifferenza della $f$ sono $h$, vi sono $2^h$ modi per assegnare 0 o 1 a tali condizioni: esistono quindi $2^h$ funzioni complete distinte che coincidono con la $f$ ovunque questa è specificata, e altrettante reti che risolvono il problema della realizzazione di $f$. Fra tutte queste reti cercheremo la minima (a due livelli), assegnando alle condizioni di indifferenza i valori, 0 o 1, più opportuni. Nell'esempio di figura 2.21 la rete SP minima si ottiene specificando le condizioni di indifferenza come indicato nella figura 2.22.

L'esempio visto è stato risolto in modo intuitivo; indichiamo ora

| $x_3x_2 \backslash x_1x_0$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 1 | 0 | 0 | 1 |
| 01 | 1 | 0 | 0 | 0 |
| 11 | 1 | 0 | 0 | 0 |
| 10 | 1 | 0 | 0 | 1 |

$$z = \overline{x}_1\overline{x}_0 + \overline{x}_2\overline{x}_0$$

**Figura 2.22**
La forma SP minima per la funzione della figura 2.21.

un'estensione sistematica dei procedimenti descritti nei paragrafi precedenti a funzioni incomplete. Una forma SP (o NAND) minima di una funzione incompleta $f$ si ottiene nel modo seguente:

*a*) si attribuisce il valore 1 a tutte le condizioni di indifferenza di $f$, e si determinano (sulla mappa, o con metodo tabellare) gli implicanti primi della funzione completa $f'$ così ottenuta;

*b*) si scartano gli implicanti primi che coprono solo 1 corrispondenti a condizioni di indifferenza di $f$;

*c*) si effettua la scelta ottima di implicanti primi di $f'$ (sulla mappa, o in una tabella di copertura), notando però che i soli 1 della funzione $f$ devono necessariamente essere coperti: in particolare un implicante primo è essenziale se unico a coprire un 1 della funzione $f$.

Per la funzione di figura 2.21 il passo *a*) conduce alla situazione di figura 2.23*a*; il passo *b*) causa l'eliminazione dei due implicanti indicati nella stessa figura; il passo *c*) consiste nella selezione dei due implicanti essenziali $\bar{x}_1\bar{x}_0$ e $\bar{x}_2\bar{x}_0$ (unici a coprire rispettivamente le celle 0100 e 0010 corrispondenti in origine a 1 della funzione). Il procedimento termina a questo punto, poiché tutti gli 1 originali della funzione sono stati coperti, e si ottiene il risultato: $z=\bar{x}_1\bar{x}_0+\bar{x}_2\bar{x}_0$, già ottenuto euristicamente nella figura 2.22.

Un procedimento perfettamente duale vale per le forme PS (o NOR): si attribuisce il valore 0 a tutte le condizioni di indifferenza; si

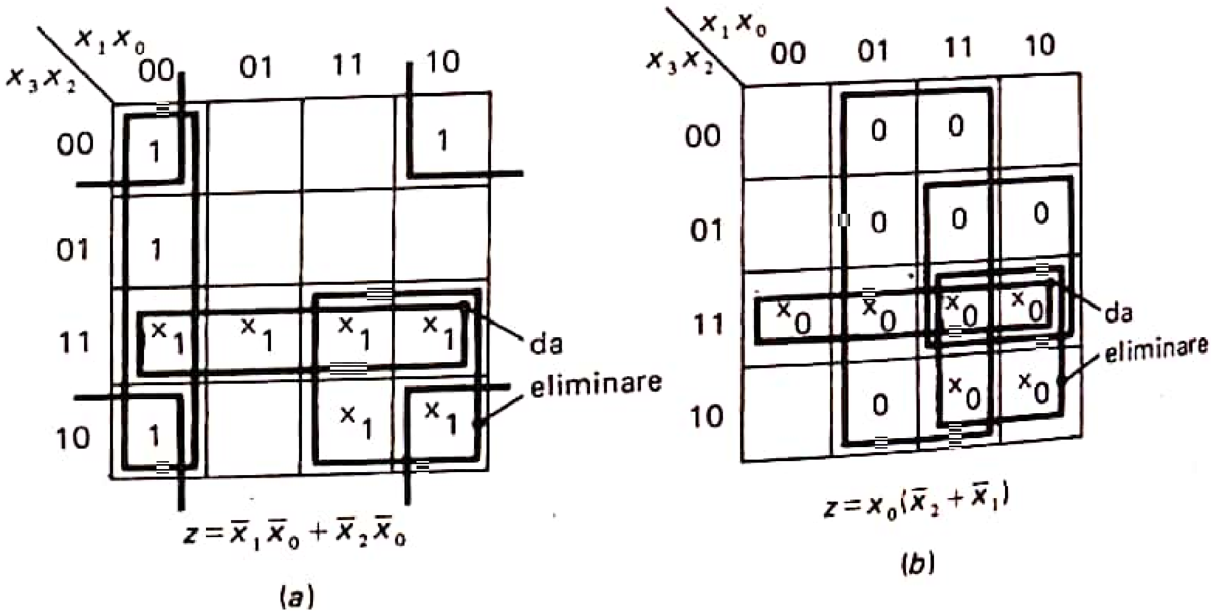



**Figura 2.23**
Determinazione delle forme minime SP e PS per la funzione della figura 2.21.

un'estensione sistematica dei procedimenti descritti nei paragrafi precedenti a funzioni incomplete. Una forma SP (o NAND) minima di una funzione incompleta $f$ si ottiene nel modo seguente:

*a*) si attribuisce il valore 1 a tutte le condizioni di indifferenza di $f$, e si determinano (sulla mappa, o con metodo tabellare) gli implicanti primi della funzione completa $f'$ così ottenuta;

*b*) si scartano gli implicanti primi che coprono solo 1 corrispondenti a condizioni di indifferenza di $f$;

*c*) si effettua la scelta ottima di implicanti primi di $f'$ (sulla mappa, o in una tabella di copertura), notando però che i soli 1 della funzione $f$ devono necessariamente essere coperti: in particolare un implicante primo è essenziale se unico a coprire un 1 della funzione $f$.

Per la funzione di figura 2.21 il passo *a*) conduce alla situazione di figura 2.23*a*; il passo *b*) causa l'eliminazione dei due implicanti indicati nella stessa figura; il passo *c*) consiste nella selezione dei due implicanti essenziali $\bar{x}_1\bar{x}_0$ e $\bar{x}_2\bar{x}_0$ (unici a coprire rispettivamente le celle 0100 e 0010 corrispondenti in origine a 1 della funzione). Il procedimento termina a questo punto, poiché tutti gli 1 originali della funzione sono stati coperti, e si ottiene il risultato: $z=\bar{x}_1\bar{x}_0+\bar{x}_2\bar{x}_0$, già ottenuto euristicamente nella figura 2.22.

Un procedimento perfettamente duale vale per le forme PS (o NOR): si attribuisce il valore 0 a tutte le condizioni di indifferenza; si

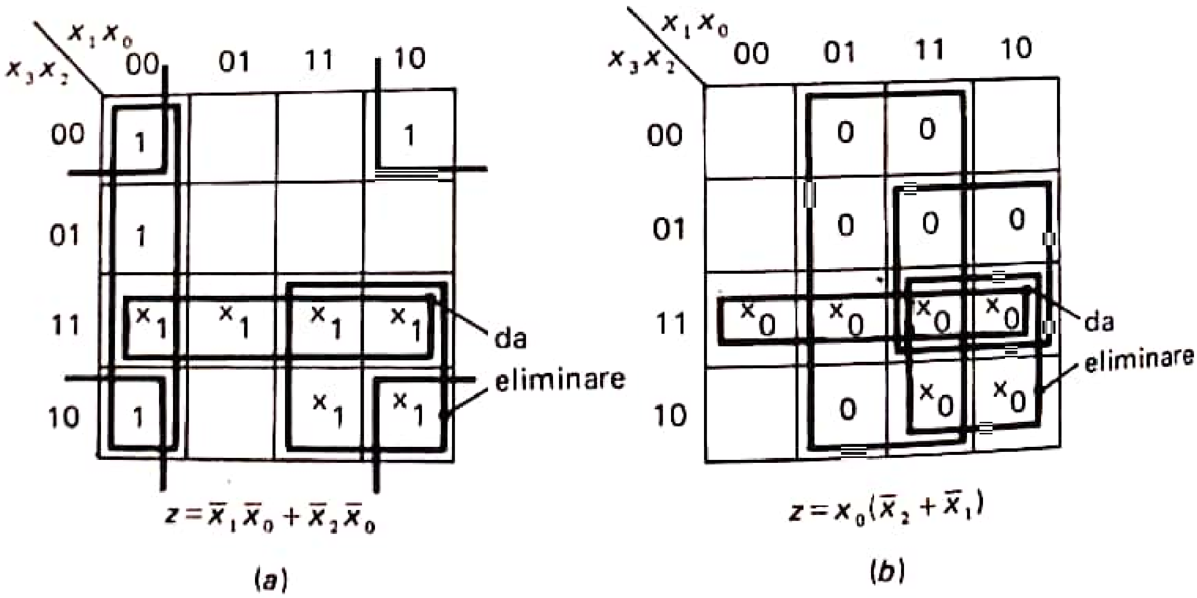



**Figura 2.23**
Determinazione delle forme minime SP e PS per la funzione della figura 2.21.

determinano tutti gli implicati primi, a eccezione di quelli che coprono solo condizioni di indifferenza; si scelgono gli implicati che coprono gli 0 originari della funzione e dànno luogo a un prodotto minimo.

Per la funzione di figura 2.21 si ha la situazione illustrata nella figura 2.23*b*, che il lettore studierà per proprio conto.

Per indicare una funzione incompleta possiamo utilizzare un'ovvia estensione della notazione fin qui adottata, che riportiamo per la funzione della figura 2.21:

$$z = \Sigma_4(0, 2, 4, 8), \quad X(10, 11, 12, 13, 14, 15)$$

oppure:

$$z = \Pi_4(1, 3, 5, 6, 7, 9), \quad X(10, 11, 12, 13, 14, 15).$$

Nella pratica le funzioni sono spesso assegnate incompletamente: abbiamo qui preferito descrivere prima le funzioni complete, e passare poi a quelle incomplete, per semplicità di trattazione.

### Esercizi

2.21 Ricavare una forma NAND minima a due livelli, e una forma NOR minima a due livelli, per le seguenti funzioni:

$$f = \Sigma_5(3, 5, 7, 11, 12, 29, 31), \quad X(1, 2, 6, 10, 28);$$

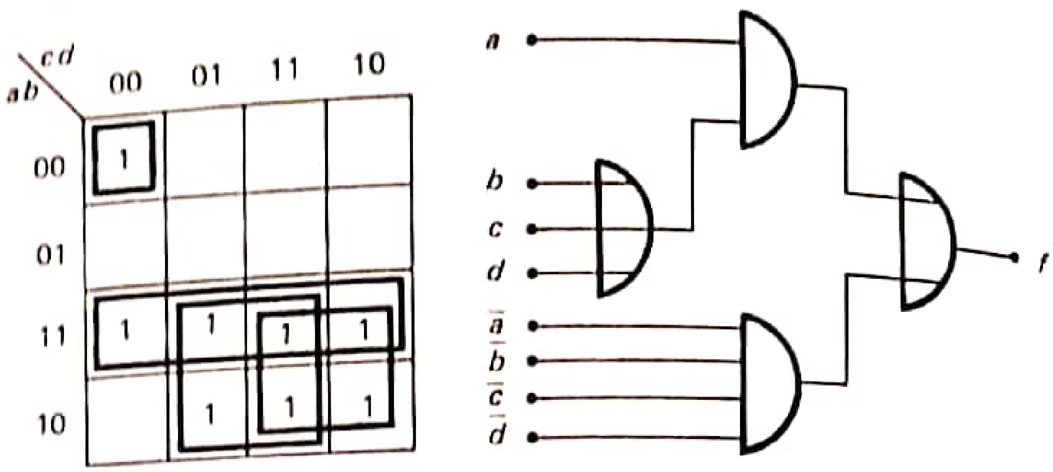
$$f = \Pi_4(1, 4, 5), \quad X(2, 3, 6, 7, 8, 9, 12, 13).$$

Si vedano anche i problemi riassuntivi riportati alla fine del capitolo.

## 2.8 Reti a più di due livelli

Nei paragrafi precedenti abbiamo studiato la sintesi ottima a due livelli, prendendo come obiettivo la minimizzazione dei parametri $N_b$ e $N_m$. Anche se, come già osservato, tale minimizzazione non conduce necessariamente a reti di costo minimo per ogni realizzazione tecnologica, lo studio condotto è ragionevolmente significativo, e serve come pietra di paragone per realizzazioni diverse.

Notiamo però che la sintesi in forma SP o PS minima può comunque non condurre a reti di costo minimo poiché, a parte ogni considerazione costruttiva, la rete che impiega il minimo numero di blocchi può non essere a due livelli. E' questo il caso della funzione di figura 2.24 per cui può essere realizzata una rete a tre livelli con $N_b = 4$ e $N_m = 11$, contro $N_b = 5$ e $N_m = 14$ delle reti minime SP e PS.




$$f=ab+ac+ad+\bar{a}\bar{b}\bar{c}\bar{d}=a\cdot(b+c+d)+\bar{a}\bar{b}\bar{c}\bar{d}$$

**Figura 2.24**
Una rete a 3 livelli.

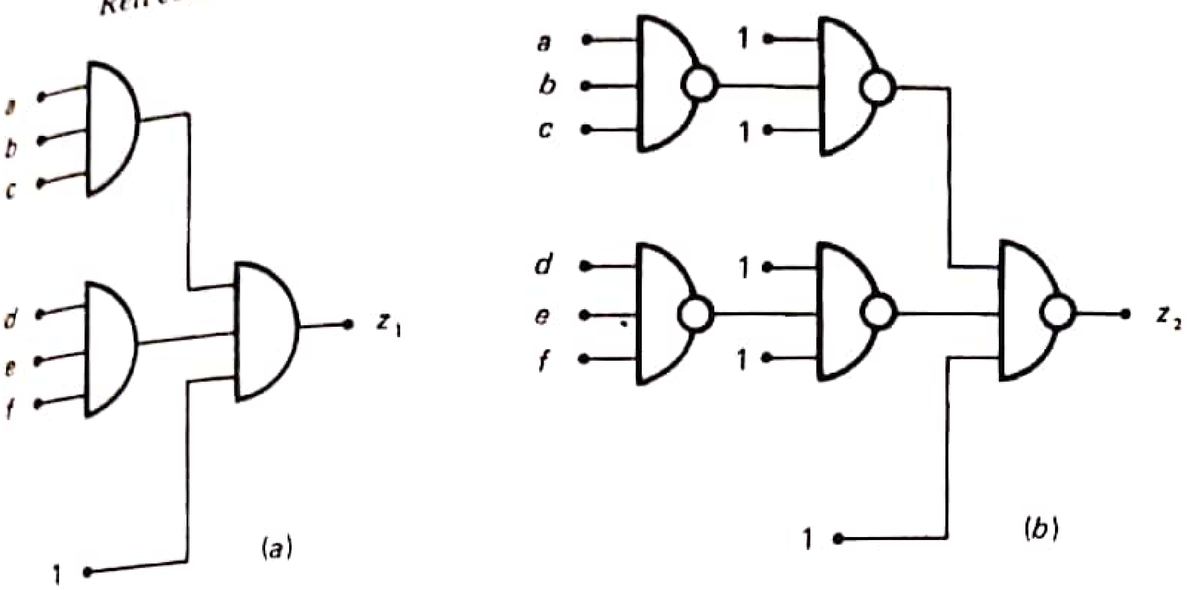
Le tecniche per ottenere reti minime a più di due livelli sono assai complesse e non possono essere discusse in questo testo. Facciamo in proposito solo alcune considerazioni.

La rete della figura 2.24 è stata ottenuta per "fattorizzazione" di un termine comune (la variabile $a$) tra diversi prodotti di variabili, legata alla proprietà distributiva dell'operatore AND sull'OR. Ciò ha permesso di ridurre il numero complessivo di blocchi, accrescendo di uno il numero di livelli. E' questo un metodo molto comune per ridurre la complessità di forme SP, e si applica in modo duale alle forme PS. Il metodo non si estende però immediatamente alle forme NAND e NOR, che non ammettono la proprietà distributiva. In questo caso, per evitare errori, è consigliabile trasformare preventivamente le espressioni SP e PS in forme a più livelli, e trasformare poi queste in NAND o NOR: in quest'ultimo passaggio il numero dei livelli in genere aumenta ancora.

Seguiamo per esempio le trasformazioni algebriche della seguente funzione $f$, che da forma SP o NAND a due livelli passa, per fattorizzazione, a forma AND/OR a tre livelli, o NAND a quattro livelli:

$$f=ab+ac+ade=(a|b)|(a|c)|(a|d|e)$$
$$\downarrow$$
$$a(b+c+de)=a(\bar{b}|\bar{c}|(d|e))=\overline{a|(\bar{b}|\bar{c}|(d|e))}.$$

Un altro motivo per trasformare a più livelli forme SP e PS è che può non essere possibile, o conveniente, reperire blocchi logici a molti ingressi, per realizzare operatori tra molte variabili. Tali blocchi sono allora spezzati in piccoli alberi di blocchi a pochi ingressi: per gli ope-




$$z_1 = abcdef = (abc)\cdot(def)\cdot 1$$
$$z_2 = a|b|c|d|e|f = \overline{(abc)\cdot(def)\cdot 1} = \overline{(a|b|c)}\,|\,\overline{(d|e|f)}\,|\,1$$

**Figura 2.25**
Realizzazione di operazioni tra molte variabili mediante blocchi a tre ingressi: (*a*) AND tra sei variabili; (*b*) NAND tra sei variabili.

ratori AND e OR, che godono della proprietà associativa, la trasformazione è immediata (fig. 2.25*a*); gli operatori NAND e NOR, che non godono di tale proprietà, richiedono un maggior numero di livelli nella trasformazione (fig. 2.25*b*). Ritorneremo su questo argomento nel prossimo capitolo studiando moduli combinatori complessi.

Nonostante queste considerazioni, le reti a due livelli rimangono molto importanti. Esse *sono le più veloci*, poiché il ritardo che subisce il segnale tra ingresso e uscita è approssimativamente proporzionale al numero di livelli della rete. Inoltre alcune considerazioni sul regime transitorio contribuiranno, in un prossimo paragrafo, a illustrare i vantaggi delle reti a due livelli.

## 2.9 Reti a più uscite

Abbiamo fin qui preso in considerazione solo reti con un morsetto di uscita, ammettendo implicitamente che la sintesi ottima di una rete a *m* uscite si ottenga realizzando separatamente una forma minima per ogni funzione di uscita: un approfondimento del problema mostra però che questa ipotesi non è vera in genere, e che una trattazione completa è tutt'altro che semplice.