Capitolo 3

Alcuni moduli combinatori

Il progetto di una rete logica richiede un'analisi preliminare che stabilisca se il problema da risolvere è per sua natura combinatorio o sequenziale. La domanda deve essere posta sul modo con cui la rete scambia informazione con l'esterno oltre che sulla elaborazione che essa deve eseguire; questa infatti può in linea di principio avere sempre natura combinatoria, se la dimensione dei dati non eccede un limite fissato a priori.

Per comprendere quest'affermazione si consideri che l'intera elaborazione eseguita da un calcolatore, che interpreta un programma su un insieme di dati, potrebbe al limite essere eseguita "in parallelo" da una rete combinatoria a n ingressi e m uscite, nell'ipotesi che tale elaborazione termini in tempo finito e siano fissate le lunghezze massime n, m delle stringhe binarie che codificano rispettivamente il programma più i dati (n), e i risultati (m).

Dall'ovvia irragionevolezza di realizzare reti così grandi deriva la necessità di presentare sequenzialmente l'informazione di ingresso, il che in genere richiede che la rete abbia facoltà di memorizzazione (sia cioè una rete sequenziale). Le reti combinatorie sono quindi destinate a problemi di modeste dimensioni, e sono in genere parti di sistemi più grandi.

I metodi illustrati nel capitolo precedente consentono di realizzare reti per qualsiasi problema combinatorio, in particolare in forma minima a due livelli. Tali problemi sono però a volte ancora così vasti che la rete a due livelli risulta troppo costosa, mentre un approccio differente al progetto può condurre a soluzioni assai più economiche. Si agisce principalmente secondo due criteri: il primo è di decomporre la

rete in più reti in cascata rinunciando così ai due livelli, se è possibile in tal modo ridurre drasticamente il numero di blocchi N_b e di connessioni ai blocchi N_m. Il secondo criterio è dare alla rete una struttura "modulare", cioè realizzarla mediante l'interconnessione di reti più piccole tutte uguali: a questo punto può non aver più molto interesse la riduzione di N_b e N_m, ma la riduzione del numero di moduli distinti, poiché ogni modulo, se sufficientemente piccolo o standard, potrebbe essere disponibile sotto forma di singolo componente integrato (tra l'altro se una rete deve essere prodotta in un grandissimo numero di esemplari, si può pensare di decomporla in moduli non standard, richiedendo ai costruttori di circuiti integrati la fornitura a basso costo di moduli realizzati *ad hoc*; vedi il cap. 8).

Naturalmente non esiste metodologia sistematica per affrontare questo tipo di progetti, e ci si deve affidare di volta in volta all'esperienza e all'intuito. Noi ci limiteremo a discutere qui tre tipi di reti, i codificatori, i selettori e le unità aritmetiche, non solo per l'importanza che esse rivestono nei sistemi per l'elaborazione dell'informazione, ma anche perché dalla loro metodologia di progetto il lettore potrà ricavare alcune idee per affrontare la realizzazione di altre reti complesse.

3.1 Codificatori

I codificatori costituiscono in senso lato una famiglia di reti combinatorie che trasformano, tra ingresso e uscita, parole rappresentate in un dato codice in parole rappresentate in un codice diverso. Ne studieremo qui due tipi, che prendono in genere i nomi di *codificatori* (in senso stretto) e *decodificatori*.

$x_7x_6x_5x_4x_3x_2x_1x_0$	$z_2z_1z_0$
0 0 0 0 0 0 0 1	0 0 0
0 0 0 0 0 0 1 0	0 0 1
0 0 0 0 0 1 0 0	0 1 0
0 0 0 0 1 0 0 0	0 1 1
0 0 0 1 0 0 0 0	1 0 0
0 0 1 0 0 0 0 0	1 0 1
0 1 0 0 0 0 0 0	1 1 0
1 0 0 0 0 0 0 0	1 1 1

Figura 3.1
Comportamento ai morsetti di un codificatore, per $n = 3$: tutte le configurazioni di x non indicate dànno luogo a non specificazioni per le z.

Un codificatore ha 2^n ingressi $x_{2^n-1}, \ldots, x_0$ e n uscite $z_{n-1}, \ldots, z_0$. Le uniche configurazioni ammesse all'ingresso contengono esattamente un 1: si ha cioè $x_i = 1$, $x_j = 0$ per ogni $j \neq i$; le corrispondenti configurazioni di uscita sono date da $(z_{n-1} \ldots z_0) = i$, ove la stringa di valori di z si interpreta come la rappresentazione dell'intero i in codice binario. Per esempio, per $n = 3$, il comportamento ai morsetti del codificatore è indicato nella figura 3.1.

Ovviamente le configurazioni di ingresso sono moltissime, ma solo quelle per cui una x vale 1 sono significative e vengono quindi indicate. Ci si può immediatamente rendere conto che le funzioni di uscita per il codificatore di figura 3.1 sono espresse da:

$$z_2 = x_7 + x_6 + x_5 + x_4,$$

$$z_1 = x_7 + x_6 + x_3 + x_2,$$

$$z_0 = x_7 + x_5 + x_3 + x_1.$$

Similmente si ottengono le espressioni delle z per ogni altro valore di n.

La rete logica del codificatore per $n = 3$ è rappresentata nella figura 3.2 (si noti il simbolo grafico per l'OR a più ingressi, già indicato in fig. 2.1).

La struttura della rete ricalca formalmente quella della tabella dei valori di z (ponendo "$\oplus$" al posto di 1). La variabile x_0 non è mai usata, e potrebbe essere omessa.

I codificatori sono tipicamente impiegati quando diverse unità sono connesse a un sistema: l'unità i-esima domanda attenzione "attivando" una linea ($x_i = 1$), e il sistema rappresenta al suo interno, in modo

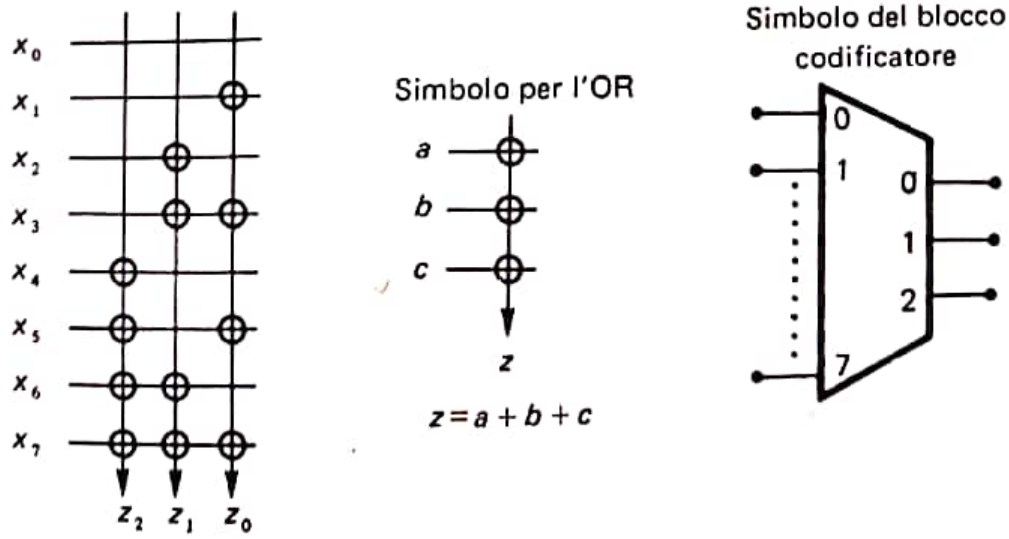


Figura 3.2
Codificatore per $n = 3$.

compatto $(z_{n-1} \ldots z_0)$ il valore di i, per ogni riferimento all'unità. Le linee di ingresso possono essere meno di 2^n, e la rete può essere priva delle relative connessioni: nulla cambia, comunque, nel metodo di costruzione delle funzioni indicato nella figura 3.2.

I codificatori sono spesso costruiti come singoli circuiti integrati. Tuttavia, al crescere delle loro dimensioni, diviene necessario realizzarli per composizione di pezzi più piccoli: ciò anche perché le operazioni di OR tra molti ingressi possono dover essere decomposte in alberi di OR più piccoli (§ 2.8). Ci aiuta nella decomposizione la natura ricorsiva della rete, come è mostrato nell'esempio di figura 3.3, immediatamente generalizzabile a qualunque valore di n, e suscettibile di numerose variazioni. Il passaggio dalla struttura della figura 3.3*a* a quella della figura 3.3*b* è banale per quanto riguarda le variabili z_2, z_1, z_0, mentre richiede un attimo di riflessione per la z_3. La nuova struttura (fig. 3.3*b*) impiega OR a quattro ingressi anziché otto, e può essere

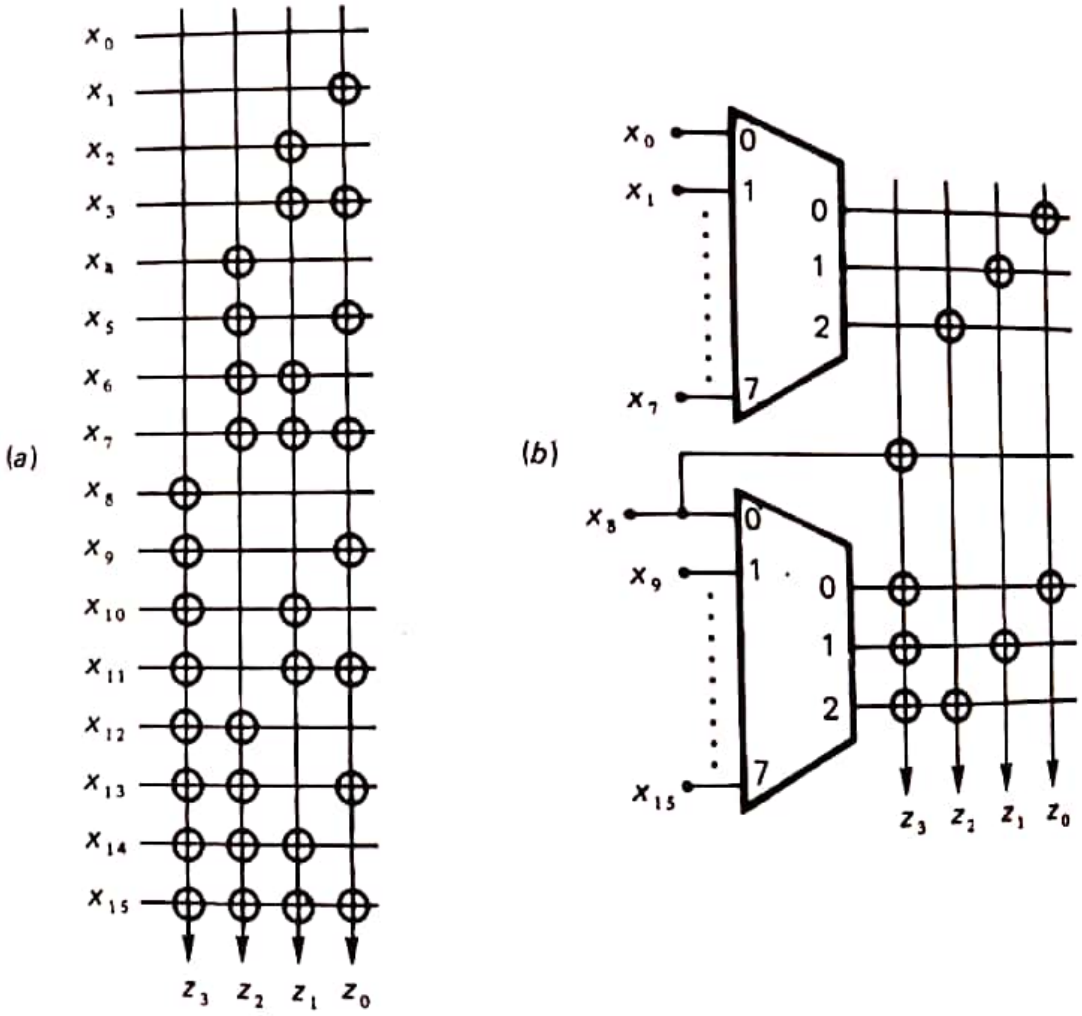


Figura 3.3
(*a*) Codificatore per $n=4$; (*b*) sua scomposizione ricorsiva in due codificatori per $n=3$.

facilmente costruita con due blocchi codificatori per $n=3$, se questi sono disponibili come singoli circuiti integrati.

Inversamente al codificatore, il decodificatore ha n ingressi $x_{n-1}, \ldots, x_0$ e 2^n uscite $z_{2^n-1}, \ldots, z_0$; esso interpreta la configurazione di ingresso come numero binario $(x_{n-1}, \ldots, x_0)=i$, e produce in uscita $z_i=1$, $z_j=0$ per ogni $j \neq i$. Anche questa rete trasforma quindi la rappresentazione dei numeri, passando dal codice binario a un'esplicita indicazione della posizione del numero in un ordinamento progressivo.

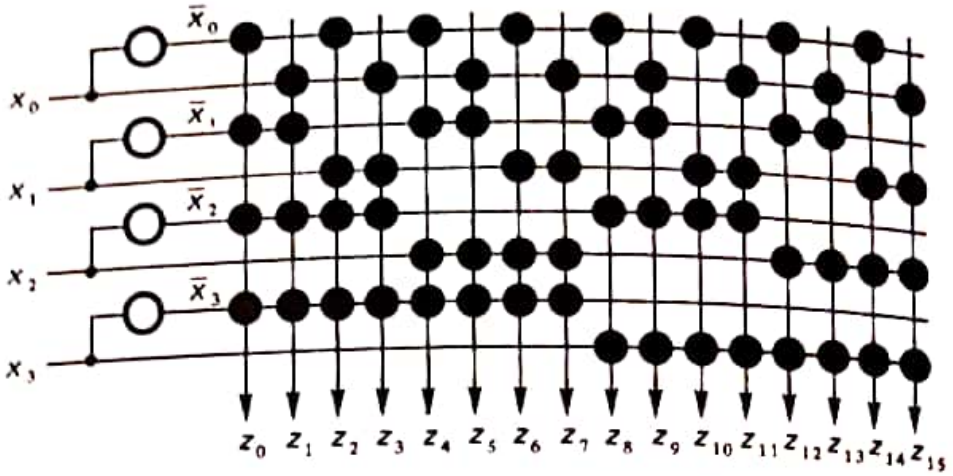
Le funzioni di uscita del decodificatore sono semplicemente

$$z_i = p_i,$$

dove p_i è il mintermine i-esimo in n variabili; cioè z_i è un prodotto delle n variabili di ingresso, e come tale può essere realizzato a uno o più livelli, in funzione delle dimensioni degli AND disponibili. Nella figura 3.4*a* è indicato un decodificatore a un livello per $n=4$, composto di 16 blocchi AND a quattro ingressi che realizzano i 16 mintermini di quattro variabili. Nella figura 3.4*b* lo stesso decodificatore è spezzato ricorsivamente in due decodificatori per $n=2$, dando luogo a una rete a due livelli composta di 24 blocchi AND a due ingressi; le uscite (mintermini di quattro variabili) sono costruite combinando in tutti i modi possibili le uscite dei due decodificatori al primo livello (mintermini di due variabili).

La porzione di rete che nella figura 3.4*b* combina i segnali al secondo livello è detta *matrice di* AND 4×4. Utilizzando blocchi AND a k ingressi si possono costruire matrici di AND $h \times h \times \ldots \times h$ (h compare k volte) con $k \cdot h$ ingressi (k gruppi di h ingressi ciascuno) e h^k uscite, ove ogni AND riceve un ingresso da ognuno dei k gruppi di ingresso della matrice. Per esempio la matrice 4×4 di figura 3.4*b* ha due gruppi di quattro ingressi ciascuno (provenienti dai due decodificatori per $n=2$).

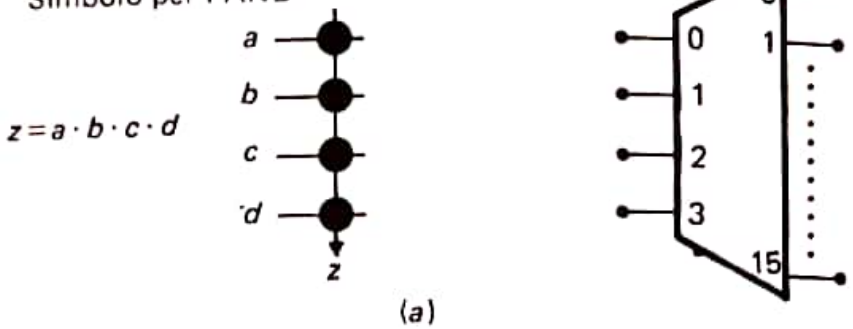
La disponibilità di matrici $h \times h \times \ldots \times h$ consente grande flessibilità nel progetto di decodificatori. Per esempio mediante blocchi AND fino a quattro ingressi, un decodificatore a otto ingressi può essere costruito come rete a due livelli, con 16 blocchi AND a due ingressi e 256 blocchi AND a quattro ingressi (fig. 3.5*a*); oppure con 256 blocchi AND a due ingressi e 32 blocchi AND a quattro ingressi (fig. 3.5*b*). La scelta del progetto dipende come sempre da considerazioni tecnologiche: se l'unico criterio è la minimizzazione di N_b, prevarrà lo schema di figura 3.5*a*; ma più probabilmente la scelta sarà legata alla disponibilità di intere parti, come decodificatori per pochi ingressi, e matrici.



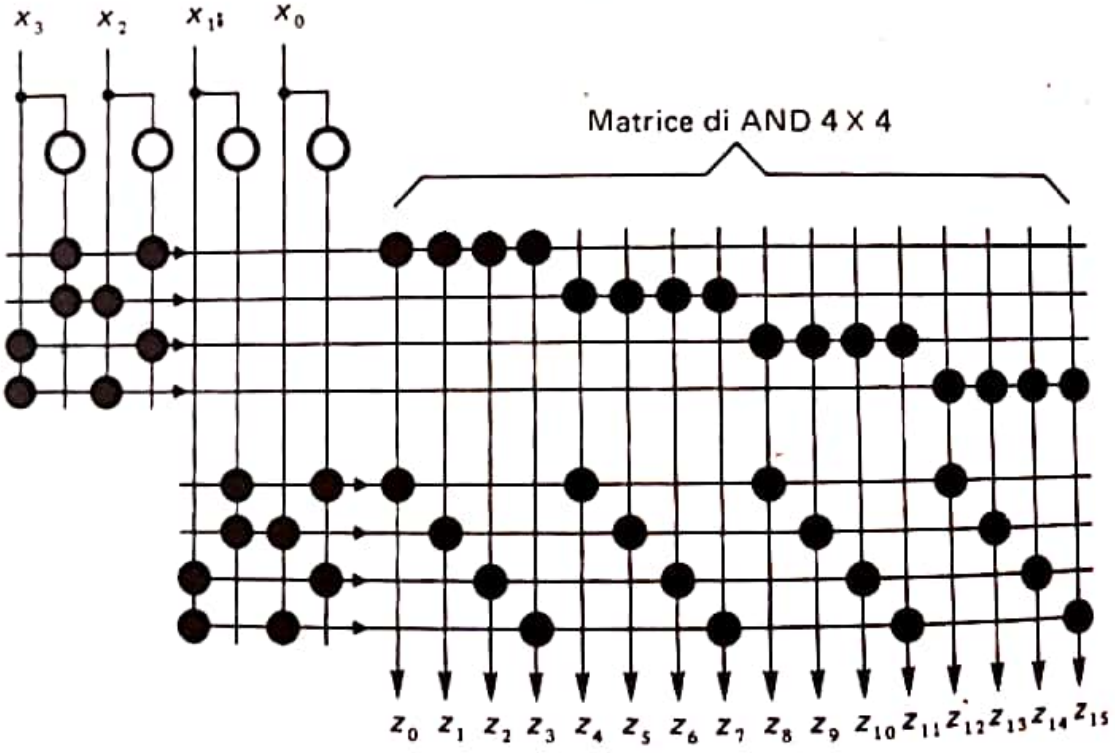
Simbolo del blocco decodificatore

Simbolo per l'AND

$z = a \cdot b \cdot c \cdot d$

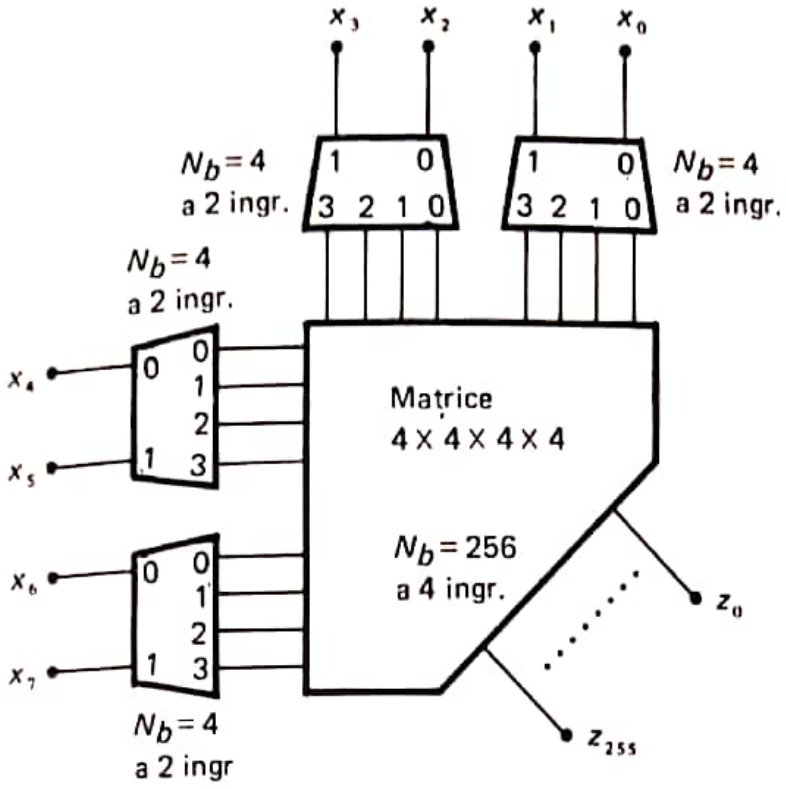


(*a*)

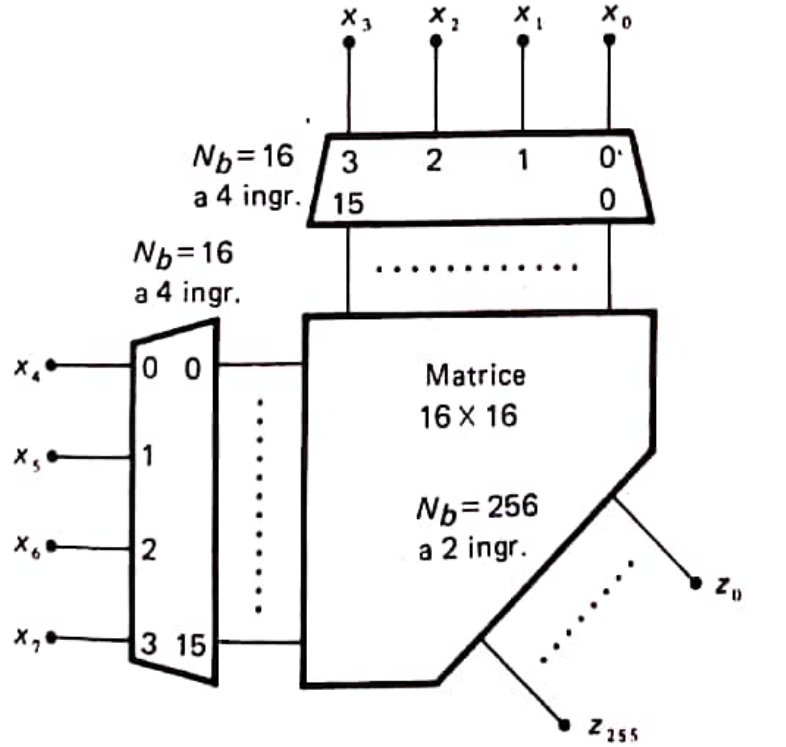


(*b*)

Figura 3.4
Decodificatore per $n = 4$. (*a*) Realizzazione a un livello; (*b*) scomposizione ricorsiva in due livelli, con due decodificatori per $n = 2$ e una matrice di AND 4 X 4.



(*a*)



(*b*)

Figura 3.5
Realizzazioni a due livelli per un decodificatore a otto ingressi.

Il decodificatore è tipicamente usato per accedere a parti di una rete (per esempio parole di memoria) identificate da un numero $(x_{n-1} \ldots x_0) = i$, e raggiunte con "l'attivazione" di una linea ($z_i = 1$). Le uscite possono essere meno di 2^n senza che nulla cambi nei criteri di progetto del decodificatore.

Esercizi

3.1 Costruire un codificatore a 12 ingressi (4 uscite) mediante blocchi OR a due o tre ingressi.

3.2 Costruire un decodificatore a sette ingressi con blocchi AND a tre ingressi.

3.3 Realizzare una rete che generi il bit di parità per configurazioni di quattro bit, impiegando solo blocchi AND a due ingressi. (La rete si ottiene con una semplice trasformazione del decodificatore di fig. 3.4*b*.)

3.2 Selettori

Studiamo ora una nuova famiglia di reti combinatorie di uso assai comune, dette selettori.

I selettori sono impiegati per dirottare su una linea di trasmissione messaggi provenienti da sorgenti diverse (*selettori di ingresso* o *multiplexer*), o per dirottare su linee diverse il messaggio proveniente da una sorgente (*selettori di uscita* o *demultiplexer*), o per combinare entrambi questi effetti (*selettori di ingresso-uscita*). Ci troviamo così di fronte a una rete combinatoria in cui alcuni segnali di ingresso rappresentano i messaggi, e altri, detti *di controllo*, selezionano le vie attraverso cui istradare i primi.

La figura 3.6 mostra i simboli funzionali dei selettori. Interpretati i gruppi di segnali di controllo $x_{k-1}, \ldots, x_0$ e $y_{h-1}, \ldots, y_0$ come numeri binari x, y, il selettore di ingresso dirotta sulla linea B il messaggio A_x; il selettore di uscita dirotta sulla linea B_y il messaggio A; il selettore di ingresso-uscita dirotta sulla linea B_y il segnale A_x. Le linee sono indicate con doppia freccia contrassegnata da n, a indicare che i messaggi sono in genere costituiti da n bit trasmessi in parallelo.

La figura 3.7 mostra due possibili realizzazioni per un selettore a quattro ingressi $A_0 = a_0^1 a_0^2 \ldots a_0^n, \ldots, A_3 = a_3^1 a_3^2 \ldots a_3^n$, e una uscita $B = b^1 b^2 \ldots b^n$: in entrambi i casi è indicata solo una cella relativa al bit

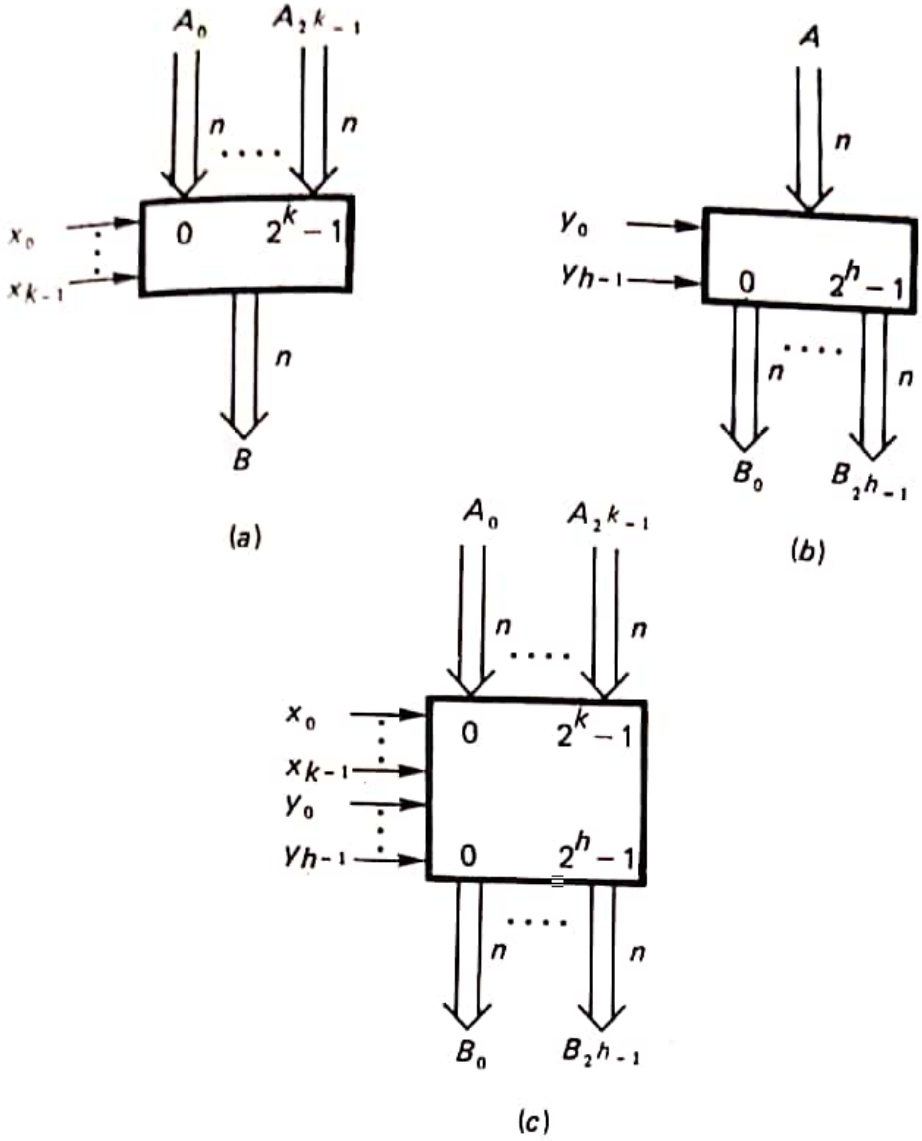
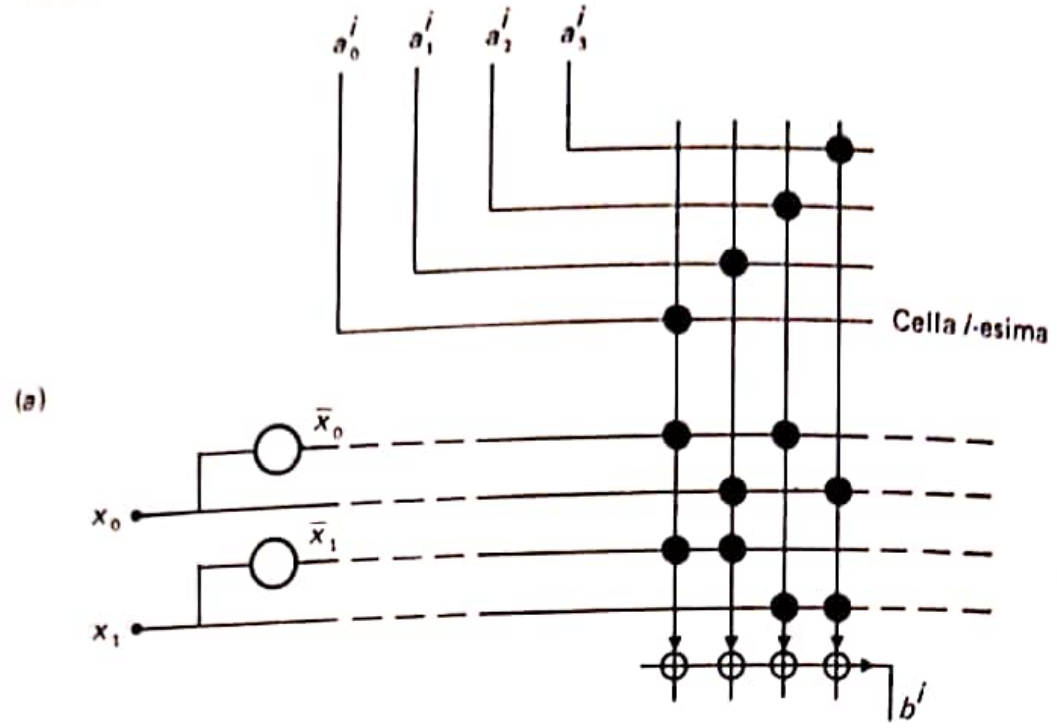


Figura 3.6
Simboli di selettori per messaggi di n bit. (*a*) Selettore di ingresso; (*b*) di uscita; (*c*) di ingresso-uscita.

i-esimo dei messaggi, ed è inteso che la rete conterrà altre $n-1$ celle identiche. La prima realizzazione (fig. 3.7*a*) è una rete SP. La seconda (fig. 3.7*b*) contiene un decodificatore a due ingressi x_1, x_0 comune a tutte le celle, e ogni cella consiste in una rete di istradamento a segnali di controllo decodificati: questa realizzazione è a tre livelli, e impiega blocchi a minor numero di ingressi, perché le operazioni di AND tra i segnali di controllo x_1, x_0 sono eseguite una volta per tutte nel decodificatore.

La scelta tra le due realizzazioni sarà guidata da considerazioni tecnologiche: è però evidente, per quanto detto a proposito dei codificatori, che all'aumentare del numero di messaggi da controllare potrà imporsi la decomposizione della rete in due, o più, reti in cascata. Ovvia



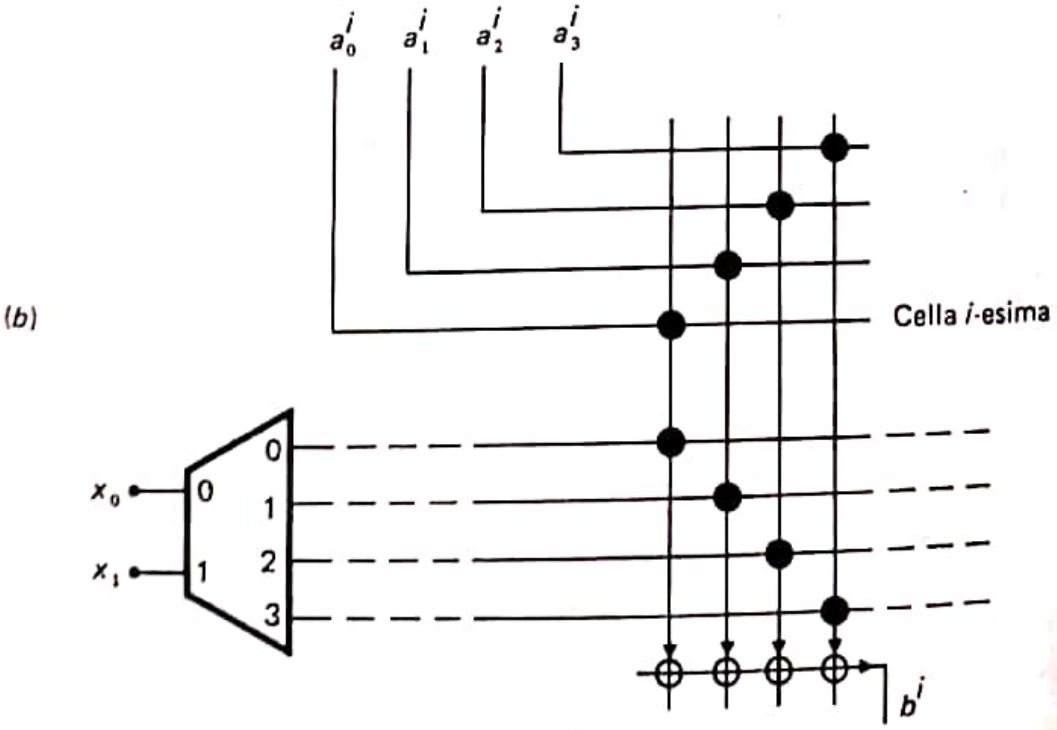


Figura 3.7
Selettore di ingresso per quattro messaggi (i-esimo bit). (a) Realizzazione a due livelli; (b) realizzazione a tre livelli mediante decodificatore.

è l'estensione del progetto a selettori con un numero qualsiasi di ingressi.

Realizzazioni analoghe a quelle della figura 3.7 valgono per i selettori di uscita, in cui il messaggio A è inviato in parallelo all'ingresso di blocchi AND controllati dalle y (non sono necessari blocchi OR, perciò la rete ha un livello in meno). Per esempio la figura 3.8 mostra la cella i-esima di un selettore di uscita per otto messaggi: la rete, a due livelli, è scomposta in un decodificatore comune, e una successiva rete di istradamento per ogni cella.

Combinando le reti descritte sopra, si ottengono varie strutture per i selettori di ingresso-uscita.

Oltre che nella trasmissione di dati tra sistemi diversi, i selettori hanno assunto grande importanza *all'interno* dei sistemi di elaborazione (per esempio all'interno di un semplice elaboratore) da quando il costo dei collegamenti è divenuto prevalente rispetto a quello dei circuiti integrati. Come vedremo, la struttura di un sistema si sviluppa attorno a poche linee di trasmissione (dette "bus") su cui, e da cui, i messaggi provenienti da varie unità sono dirottati mediante selettori. In queste configurazioni, tuttavia, ha importanza preminente il selettore di ingresso, che sceglie il messaggio da trasmettere, mentre spesso il selettore di uscita non è esplicitamente presente perché le unità connesse alla linea stabiliscono, con controllo locale, se acquisire o meno il messaggio in arrivo.

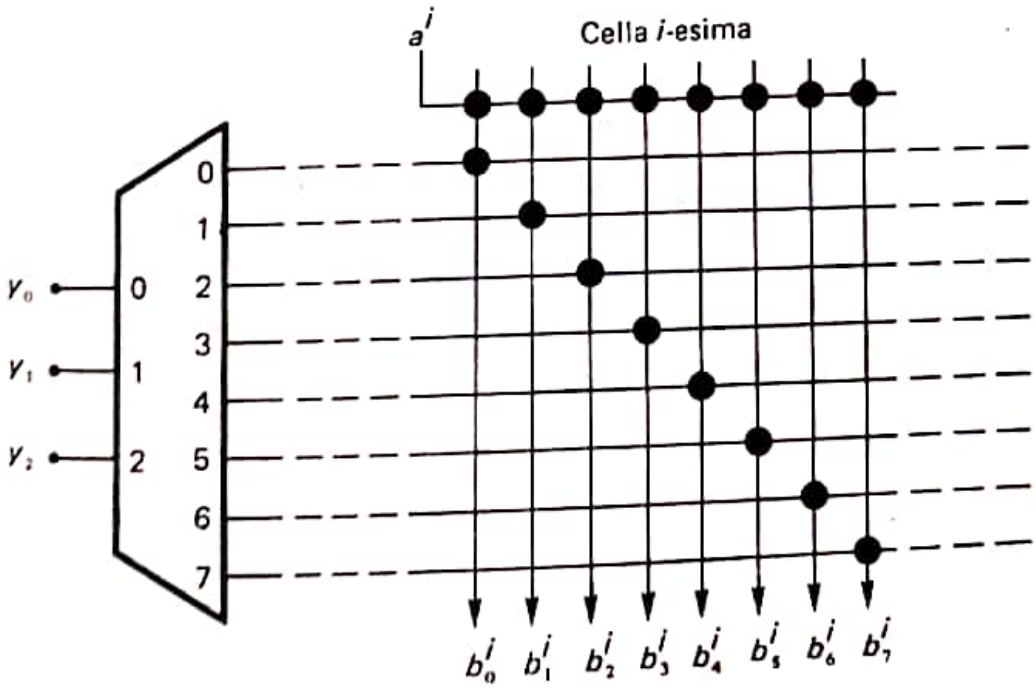


Figura 3.8
Selettore di uscita per otto messaggi (i-esimo bit).