

# CIRCUITI COMBINATORI MSI



Nelle tabelle della verità che descrivono il funzionamento degli integrati, i valori delle variabili vengono indicati con i livelli H o L delle tensioni e non mediante i valori logici 1 o 0. In questo modo la tabella risulta valida sia nel caso di dati in **logica positiva** ( $H \rightarrow 1, L \rightarrow 0$ ) che in quello di dati in **logica negativa** ( $H \rightarrow 0, L \rightarrow 1$ ).

Negli integrati presentati in questo e nei prossimi capitoli, alcuni pin hanno o funzione di **controllo**; i due livelli di tensione H e L questa volta non rappresenteranno valori logici 1 e 0, ma gli stati "attivo" e "non attivo" dei pin. In logica positiva vale l'associazione:  $H \rightarrow$  attivo,  $L \rightarrow$  non attivo; in logica negativa vale:  $H \rightarrow$  non attivo,  $L \rightarrow$  attivo.

La tab. 5.1 sintetizza i concetti appena esposti.

Tab. 5.1 - Logica positiva e logica negativa.

Livello	Logica Positiva		Logica Negativa	
	Dato	Controllo	Dato	Controllo
L	0	Non attivo	1	Attivo
H	1	Attivo	0	Non attivo

Per segnalare ingressi o uscite in logica negativa la maggior parte dei costruttori pone un trattino sopra la sigla che identifica il pin e un pallino di negazione in prossimità del blocco funzionale.

Nell'uso degli integrati, è necessario analizzare attentamente la funzione di tutti i pin, evitando di lasciare scollegati gli ingressi inutilizzati; la logica TTL, ad esempio, interpreta come ALTO lo stato degli ingressi scollegati.

## 5.1 Multiplexer

**Def.** Il **multiplexer** (mux) è un circuito che trasferisce verso l'uscita  $Y$  il valore logico presente su uno dei  $2^n$  ingressi dati, individuato dalla combinazione di bit posta sugli  $n$  ingressi di selezione.

In fig. 5.1 sono rappresentati un multiplexer con quattro ingressi dati ( $I$ ) più due ingressi di selezione ( $S$ ), e la relativa tabella della verità. L'ingresso  $\bar{E}$ , se posto a livello BASSO, abilita l'integrato; in caso contrario l'uscita rimane a un livello fisso.

Sviluppando il progetto del multiplexer con le tecniche studiate nel cap. 3, si ottiene la rete logica di fig. 5.2, dove si nota che le porte AND svolgono la funzione di interruttori, controllati dalle combinazioni di valori presenti su  $S_0$  e  $S_1$ . L'unica porta abilitata consentirà al corrispondente segnale dati  $I$  di procedere verso l'uscita  $Y$ .

### 5.1.1 Multiplexer integrati

In tutte le famiglie logiche è possibile reperire vari multiplexer integrati, differenti per numero d'ingressi ed organizzazione interna. Esempi di multiplexer TTL sono gli integrati 74151 e 74150, di cui si riportano le principali caratteristiche.

I pin del multiplexer 74151 (fig. 5.3) hanno le seguenti funzioni:

- data inputs  $D_0 \div D_7$ : ingressi dati;
- data select  $A, B, C$ : ingressi di selezione;

## 5.2.1 Codificatori integrati

Nella fig. 5.8 sono riportati il pin-out e la tabella della verità del priority encoder integrato a otto ingressi 74148.

Descrizione dei pin del priority encoder 74148:

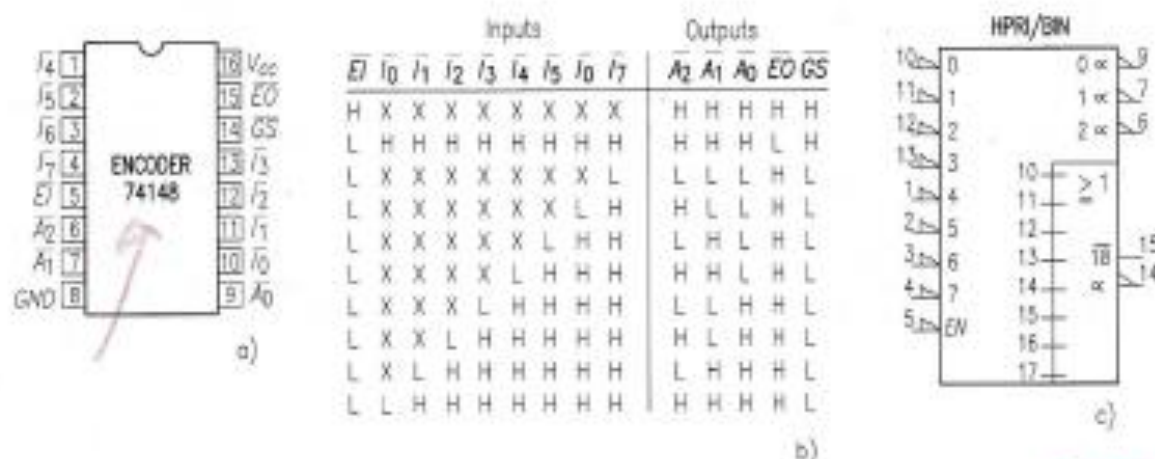
- $\bar{I}_0 + \bar{I}_7$ : terminali d'ingresso a priorità crescente;
- $\bar{A}_0 + \bar{A}_2$ : terminali d'uscita.

BINGR.  
3 VSC.

Tutti i terminali dell'integrato funzionano in logica negativa, quindi considerano come livello attivo quello basso (L). Ad esempio per codificare l'ingresso 5 ( $\bar{I}_5$ ), si porrà  $\bar{I}_5 = L$ , mentre gli altri ingressi saranno H; si avranno così sulle uscite i livelli LHL, che in logica negativa rappresentano il numero binario  $101_2$ .

Sono presenti inoltre alcuni terminali ausiliari, con il seguente significato:

- $\bar{EI}$  (Enable Input): ingresso che abilita l'integrato a livello L; se è posto a livello H (prima riga della tabella) tutte le uscite rimangono alte indipendentemente dal valore degli ingressi;
- $\bar{EO}$  (Enable Output): uscita che si porta a livello L quando l'integrato è abilitato ma non è presente nessun ingresso attivo (L);
- $\bar{GS}$  (Group Signal): uscita che si porta a livello L quando l'integrato è abilitato ed è attivo almeno un ingresso da codificare.



Priority encoder integrato a otto ingressi 74148: a) pin-out; b) tabella della verità; c) simbolo IEEE-IEC.

Fig. 5.8

Dalla terza riga della tabella si nota come l'ingresso  $\bar{I}_7$  sia quello a priorità maggiore; infatti, se attivato (L), le uscite lo codificano (LLL) qualunque siano i valori degli altri ingressi.

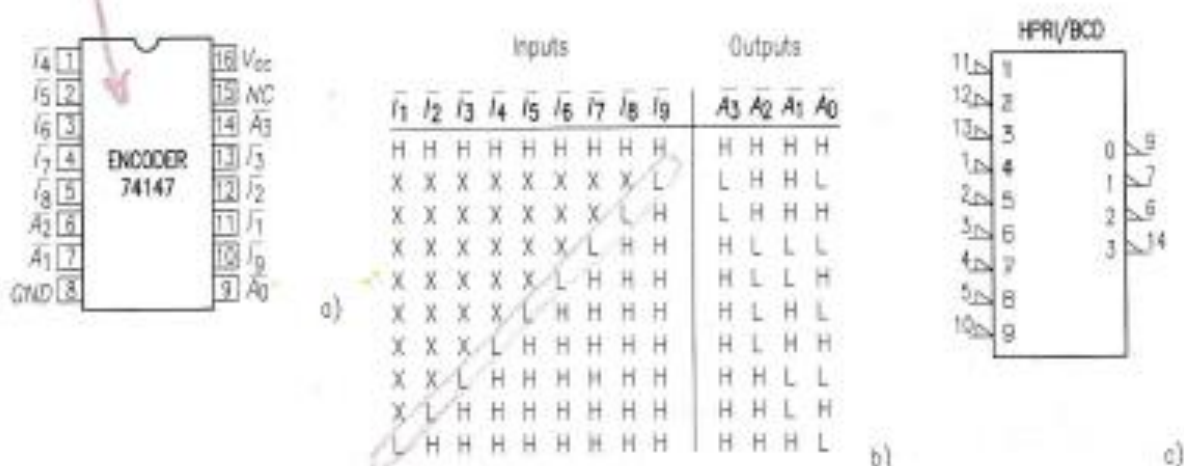
Collegando opportunamente i terminali  $\bar{EI}$ ,  $\bar{EO}$  e  $\bar{GS}$  è possibile utilizzare più integrati per espandere il numero degli ingressi.

Nella fig. 5.9 è riportato il pin-out del priority encoder integrato a nove ingressi 74147.

Descrizione dei pin del priority encoder 74147:

- come il 74148 ha tutti i terminali attivi bassi.
- È un codificatore decimale-BCD ma possiede solo nove ingressi (1 + 9) perché la configurazione di uscita HHHH, che in logica negativa corrisponde al valore 0, viene prodotta quando nessun ingresso è attivo; in pratica l'ingresso 0 è stato sostituito dalla combinazione degli ingressi 1 + 9 a livello H.





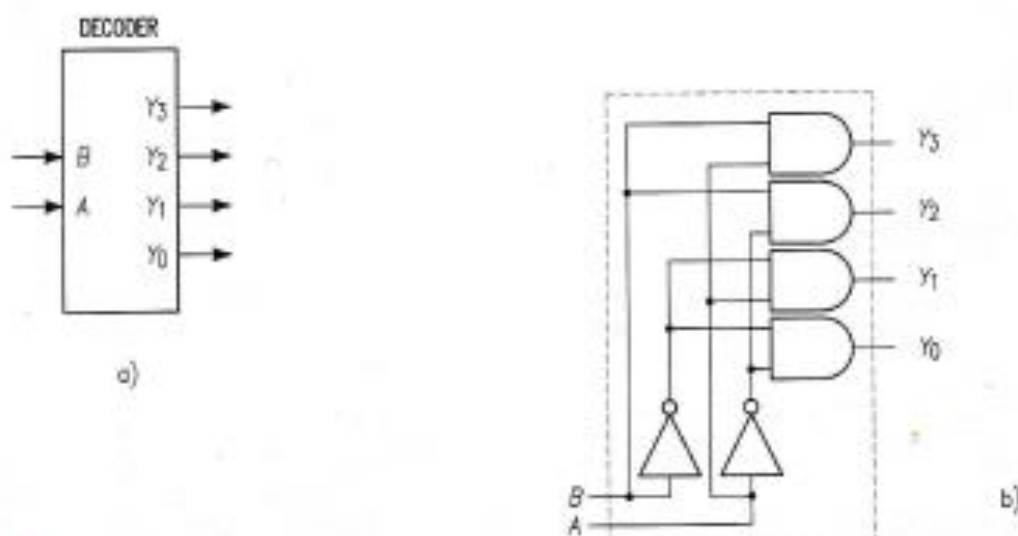
**Fig. 5.9** Priority encoder integrato decimale/BCD 74147: a) pin-out; b) tabella della verità; c) simbolo IEEE-IEC.

### 5.3 Decodificatore (decoder) / demultiplexer

**Def.** Il **decodificatore** (decoder) svolge la funzione inversa del codificatore, pertanto, ricevendo in ingresso un numero binario di  $n$  bit, attiva la corrispondente uscita tra le  $2^n$  disponibili.

Si riportano in fig. 5.10 il simbolo funzionale e lo schema a porte logiche di un decoder a due ingressi e quattro uscite. Dallo schema si nota che ogni diversa combinazione sugli ingressi  $B$  e  $A$ , porta a 1 entrambi gli ingressi di una sola porta AND e di conseguenza rende attiva l'uscita corrispondente.

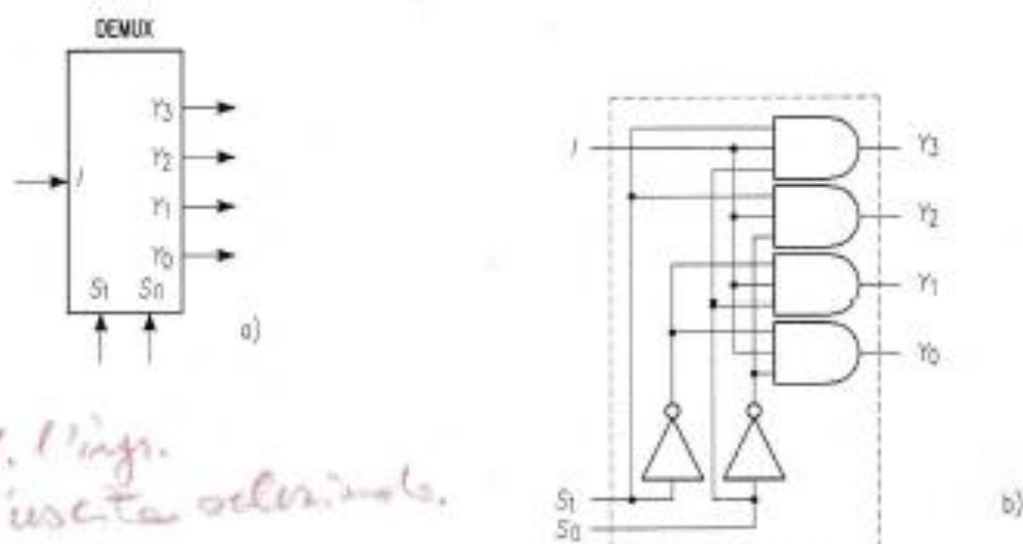
**Def.** Il **demultiplexer** (demux) svolge la funzione inversa del multiplexer, cioè riproduce il valore logico presente nell'ingresso dati / sull'uscita individuata, tra le  $2^n$  disponibili, dal numero binario posto sugli  $n$  ingressi di selezione.



**Fig. 5.10** Decoder a due ingressi: a) simbolo funzionale; b) schema a porte logiche.

Si riportano in fig. 5.11 il simbolo funzionale e lo schema a porte logiche di un demultiplexer con un ingresso dati, due ingressi di selezione e quattro uscite.

Si osservi che lo schema del demux è molto simile a quello del decoder; l'unica differenza è costituita dall'aggiunta dell'ingresso dati  $I$ , che viene riportato all'ingresso di ogni AND. In questo modo la porta abilitata dai valori di  $S_0$  ed  $S_1$  si comporta come un interruttore chiuso che consente il transito dei dati dall'ingresso  $I$  all'uscita selezionata.



Demultiplexer a quattro uscite: a) simbolo funzionale; b) schema a porte logiche. Fig. 5.11

Per utilizzare un demux come decoder sarà quindi sufficiente porre sugli ingressi di selezione il numero binario da decodificare e collegare l'ingresso  $I$  a 1; quest'ingresso assume così il significato di abilitazione che, se posto a 0, rende inattive tutte le uscite.

### 5.3.1 Decoder/demultiplexer integrati

L'affinità circuitale dei decoder e dei demux ha spinto i costruttori a realizzare integrati che svolgono entrambe le funzioni. Si riportano le caratteristiche di alcuni decoder/demux della famiglia logica TTL.

Descrizione dei pin del decoder/demux 74138 (fig. 5.12):

- 3 ingressi di selezione in logica positiva ( $A, B, C$ );
- 8 uscite in logica negativa ( $\overline{Y_0} + \overline{Y_7}$ );
- utilizzo come decoder: gli ingressi  $G1$ ,  $\overline{G2A}$ ,  $\overline{G2B}$  fungono da abilitazione ( $G1 = 1$ ,  $\overline{G2A} = \overline{G2B} = 0$ ), su  $A, B, C$  si pone il numero binario che rende attiva (L) l'uscita ( $\overline{Y_0} + \overline{Y_7}$ ) corrispondente;
- utilizzo come demux: posti  $\overline{G2A} = \overline{G2B} = 0$ , nell'ingresso  $G1$  s'inseriscono i dati che saranno indirizzati sull'uscita ( $\overline{Y_0} + \overline{Y_7}$ ) individuata dal valore binario presente sugli ingressi di selezione ( $A, B, C$ ).

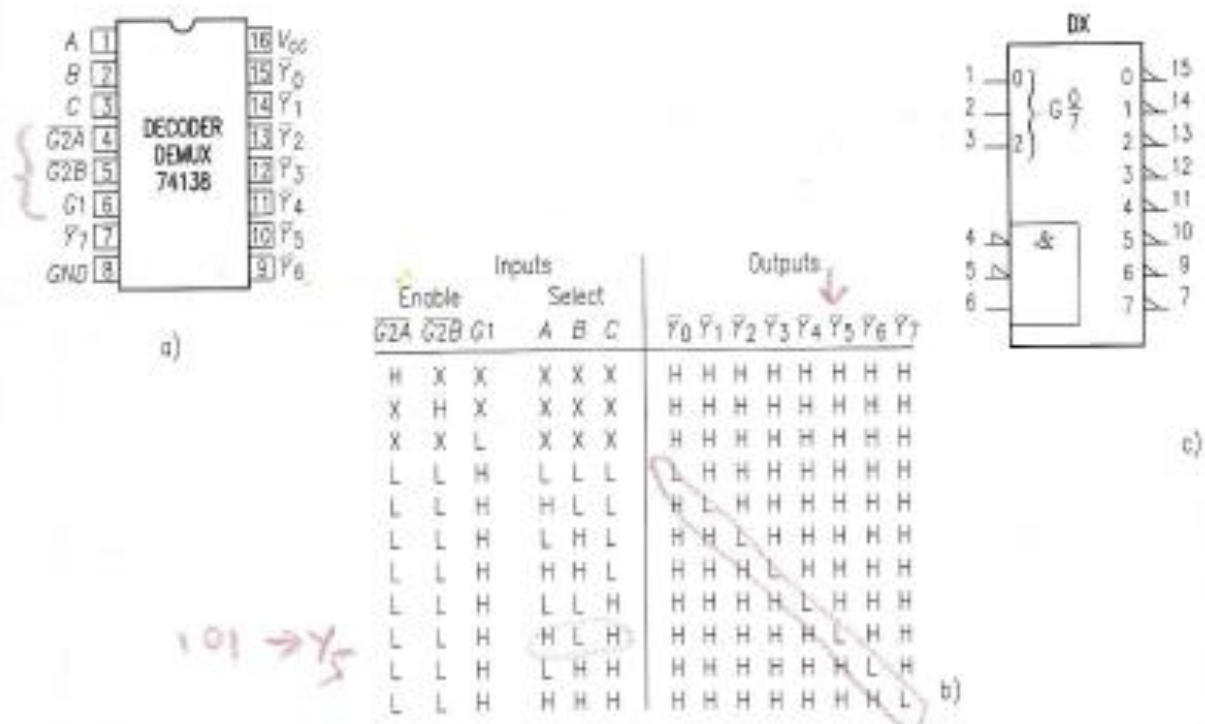


Fig. 5.12 Decoder /demultiplexer integrato 74138: a) pin-out; b) tabella della verità; c) simbolo IEEE-IEC.

Descrizione dei pin del decoder/demux 74154 (fig. 5.13):

- 4 ingressi di selezione in logica positiva (A,B,C,D);
- 16 uscite in logica negativa ( $\bar{0}$ -15);
- 2 ingressi (G1, G2) di abilitazione/ingresso dati.

BCD - decod.  
74141

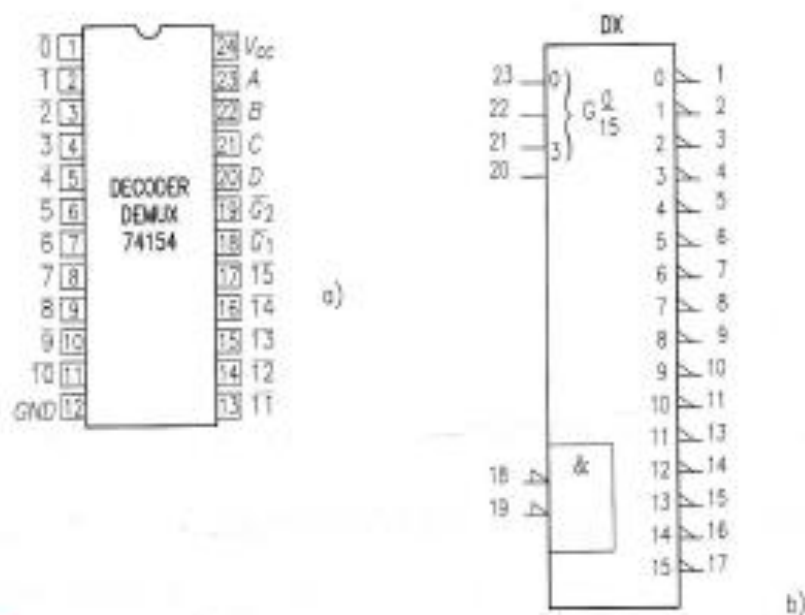


Fig. 5.13 Decoder /demultiplexer integrato 74154: a) pin-out; b) simbolo IEEE-IEC.



Altri integrati dello stesso tipo sono il 74137 (come il 74138 ma fornito di latch per memorizzare i dati in ingresso) e il 74139 (doppio decoder/demux a due ingressi e quattro uscite con un ingresso dati/abilitazione).

Esistono alcuni integrati che svolgono solo la funzione di decoder decimale: possiedono dieci uscite e ricevono in ingresso quattro bit in codice BCD (7442) o in codice eccesso-3 (7443).

L'espansione del numero delle uscite, che può essere ottenuta collegando opportunamente più integrati, viene lasciata al lettore come esercizio.

## ESEMPIO 5.2

Per controllare il funzionamento di una centrale remota, si vogliono trasmettere alla centrale principale quindici segnali d'allarme, caratterizzati da un certo ordine d'importanza, che saranno visualizzati mediante quindici LED rossi.

Essendo disponibile un cavo con soli cinque fili, s'individuì la soluzione circuitale che consenta l'accensione, nella centrale principale, del LED rosso che segnala l'allarme di maggiore gravità presente nella centrale remota.

Nel caso di assenza di allarmi dovrà accendersi un LED verde. (Si trascurino i problemi elettrici dovuti alla lunghezza del collegamento e si suppongano tutti i segnali attivi a livello ALTO).

### Soluzione

Tramite un priority encoder si codifica, con un numero binario di quattro bit, l'allarme a maggior priorità presente in ingresso. L'ingresso a più bassa priorità dell'encoder, è collegato al livello attivo (H) e viene codificato in assenza di allarmi. Le quattro uscite del codificatore, più il riferimento di massa per le tensioni, sono collegate alla centrale principale tramite un cavo a cinque fili. In ricezione il decoder attiva l'uscita corrispondente alla codifica binaria rilevata; all'uscita 0 è collegato il LED verde, che segnala la condizione di assenza di allarmi, mentre le restanti uscite sono collegate ai LED rossi.

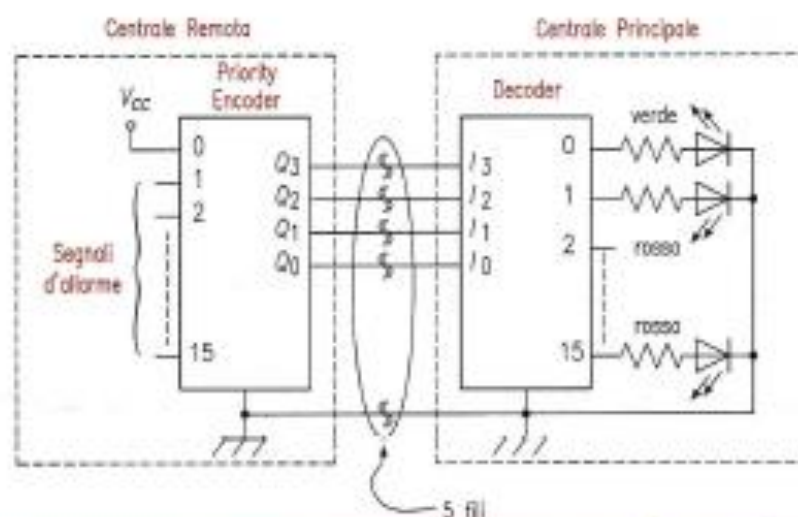


Fig. 5.14

## 5.4 Generazione di funzioni logiche con decoder e multiplexer

Nel cap. 3 si è affrontato il progetto delle reti combinatorie studiando le tecniche che consentono di passare da una tabella della verità ad una rete logica che la soddisfi. Si è rilevata anche la necessità di ottimizzare la rete per utilizzare il minimo numero di circuiti

integrati. Un modo per ridurre il numero degli integrati, consiste nel generare la funzione logica sfruttando decoder o multiplexer.

### 5.4.1 Utilizzo del decoder come generatore di funzioni

Dallo schema a porte logiche di un decoder (fig. 5.9b) si nota come ad ogni uscita corrisponda un mintermine delle variabili d'ingresso. È quindi evidente che, per realizzare la funzione in forma canonica SP, sarà sufficiente porre in OR le uscite del decoder corrispondenti alle righe della tabella che presentano un 1 nella variabile d'uscita.

Se il decoder utilizzato possiede le uscite in logica negativa si dimostra, attraverso il teorema di De Morgan, che alla porta OR è necessario sostituire una porta NAND.

#### ESEMPIO 5.3

Generare la funzione logica specificata dalla tabella della verità di fig. 5.15.a, utilizzando un decoder.

*Soluzione*

Si pongono in OR le uscite del decoder corrispondenti alle righe della tabella che presentano il valore 1 della variabile d'uscita.

C	B	A	Z
0	0	0	1 → $Y_0$
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1 → $Y_4$
1	0	1	0
1	1	0	1 → $Y_6$
1	1	1	0

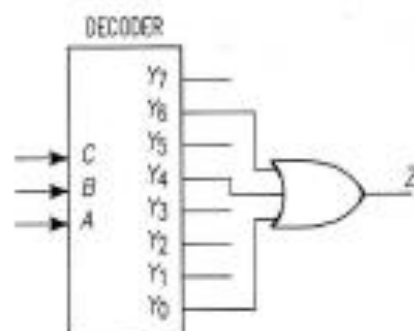


Fig. 5.15

a)

b)

Impiegando un decoder si può ottenere un risparmio sul numero di integrati utilizzati, quando si hanno più variabili d'uscita funzioni delle stesse variabili d'ingresso; è così sufficiente aggiungere all'integrato una porta OR per ogni variabile che s'intende sintetizzare, come mostrato nell'esempio 5.4.

#### ESEMPIO 5.4

Utilizzando il decoder integrato 74138, si realizzi un circuito che segnali tramite un numero binario di due cifre, la quantità di 1 presenti sui tre ingressi. Ingressi e uscite vanno considerati in logica positiva.

### Soluzione

La tabella della verità corrispondente al problema posto è rappresentata in fig. 5.16.a.

C	B	A	Z <sub>1</sub>	Z <sub>0</sub>
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

a)

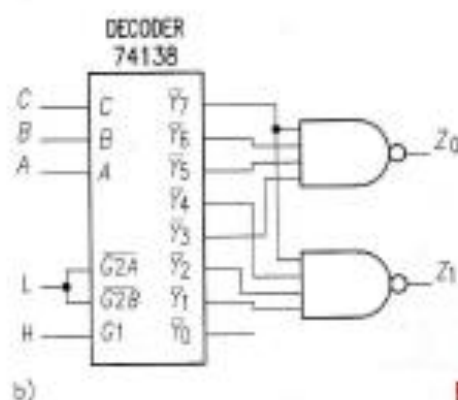


Fig. 5.16

Si procede per ognuna delle variabili d'uscita come nell'esempio 5.3, tenendo conto che, essendo le uscite del decoder 74138 in logica negativa, bisogna utilizzare due porte NAND.

## 5.4.2 Utilizzo del multiplexer come generatore di funzioni

Analizzando lo schema a porte logiche di un multiplexer (fig. 5.2), si nota come esso contenga una rete AND-OR. Ciò consente di realizzare la funzione logica corrispondente ad una data tabella della verità, utilizzando uno dei tre metodi seguenti:

### a) Mux con tanti ingressi di selezione quante sono le variabili d'ingresso della funzione

- 1) Si collegano le variabili d'ingresso agli ingressi di selezione del mux.
- 2) Si collegano al valore logico 1 gli ingressi dati del mux corrispondenti agli 1 della tabella della verità.
- 3) Si collegano al valore logico 0 gli ingressi dati del mux corrispondenti agli 0 della tabella della verità.

In questo modo la porta AND abilitata dalla combinazione delle variabili d'ingresso poste sui terminali di selezione, consentirà al valore logico presente sull'ingresso dati corrispondente, di passare verso l'uscita come specificato dalla tabella della verità.



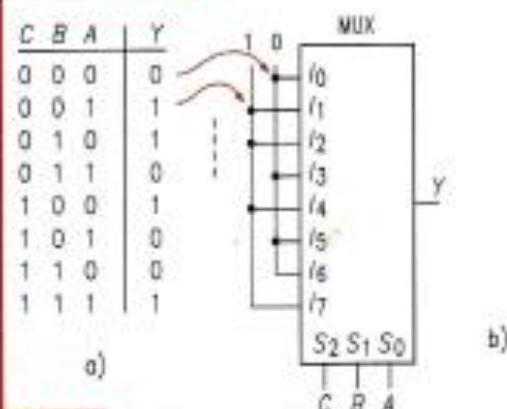
**ESEMPIO 5.5**

Fig. 5.17

Generare la funzione logica specificata dalla tabella in fig. 5.17a, utilizzando un multiplexer.

*Soluzione*

La soluzione è rappresentata nella fig. 5.17.

**b) Mux con numero d'ingressi di selezione pari al numero delle variabili d'ingresso della funzione meno uno**

- 1) Si inviano le  $n-1$  variabili d'ingresso più significative ( $C$  e  $B$  nell'esempio 5.6) agli ingressi di selezione del mux.
- 2) Si suddivide la tabella in tante parti quante sono le possibili combinazioni delle  $n-1$  variabili d'ingresso più significative; in ogni porzione queste variabili hanno valore costante, mentre quella meno significativa ( $A$ ) commuta.
- 3) Si confronta poi, in ogni porzione, il valore della variabile meno significativa ( $A$ ) con quello dell'uscita  $Y$ : a seconda della relazione tra  $A$  e  $Y$  ( $Y = A$ ,  $Y = \bar{A}$ ,  $Y = 0$  o  $Y = 1$ ) si collega l'ingresso dati del mux, corrispondente alla porzione di tabella esaminata, a  $A$ ,  $\bar{A}$ ,  $0$  o  $1$ .

Questa tecnica, rispetto alla precedente, consente, a parità di funzione da sintetizzare, l'utilizzo di un mux con un ingresso di selezione in meno e metà ingressi dati, richiedendo solo l'aggiunta di qualche porta NOT.

**ESEMPIO 5.5**

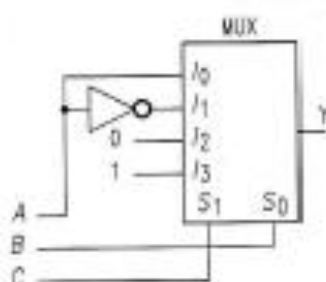
Generare la funzione logica con tre variabili d'ingresso, specificata dalla tabella della verità in figura, utilizzando un mux con due ingressi di selezione.

*Soluzione*

- 1) Si collegano le variabili d'ingresso  $B$  e  $C$  agli ingressi di selezione  $S_0$  ed  $S_1$ .
- 2) Si suddivide la tabella nelle quattro parti individuate da combinazioni identiche dei valori delle variabili  $B$  e  $C$ .
- 3) Per ogni porzione s'individua la funzione che lega  $Y$  con  $A$ .
- 4) Si collega l'ingresso corrispondente ad ogni porzione della tabella, in base alla funzione individuata nel punto precedente.

	C	B	A	Y
$I_0$	0	0	0	0
	0	0	1	1
$I_1$	0	1	0	1
	0	1	1	0
$I_2$	1	0	0	0
	1	0	1	0
$I_3$	1	1	0	1
	1	1	1	1

a)



b)

Fig. 5.18

### c) Mux con numero d'ingressi di selezione pari al numero delle variabili d'ingresso della funzione meno due

- 1) S'invisano le  $n-2$  variabili d'ingresso più significative ( $D$  e  $C$  nell'esempio 5.7) agli ingressi di selezione del mux.
- 2) Si suddivide la tabella in tante parti quante sono le possibili combinazioni delle  $n-2$  variabili d'ingresso più significative; in ogni porzione queste variabili hanno valore costante mentre quelle meno significative ( $B$  e  $A$ ) commutano.
- 3) Per ogni porzione si ricava, utilizzando le mappe di Karnaugh, la funzione di commutazione che lega le due variabili meno significative con l'uscita; nella funzione si sostituisce alla variabile d'uscita quella d'ingresso dati individuata dagli ingressi di selezione.
- 4) Si collega ogni ingresso dati del mux alla corrispondente rete logica, ricavata al punto 3.

La tecnica c) consente l'utilizzo di un mux con due ingressi di selezione in meno rispetto al numero delle variabili d'ingresso, però necessita di una rete logica esterna che richiede un certo numero di circuiti integrati.

### ESEMPIO 5.7

Generare la funzione logica con quattro variabili d'ingresso, specificata dalla tabella della verità in figura, utilizzando un mux con due ingressi di selezione.

#### Soluzione

- 1) Si collegano le variabili d'ingresso  $D$  e  $C$  agli ingressi di selezione  $S_1$  e  $S_0$ .
- 2) Si suddivide la tabella nelle quattro parti individuate da combinazioni identiche dei valori delle variabili  $D$  e  $C$ .
- 3) Per ogni porzione s'individua la funzione di commutazione che lega  $Y$  con  $A$  e  $B$ .
- 4) Si collegano i quattro ingressi dati ( $I_0 + I_3$ ) con le uscite delle reti corrispondenti alle funzioni ricavate al punto precedente.

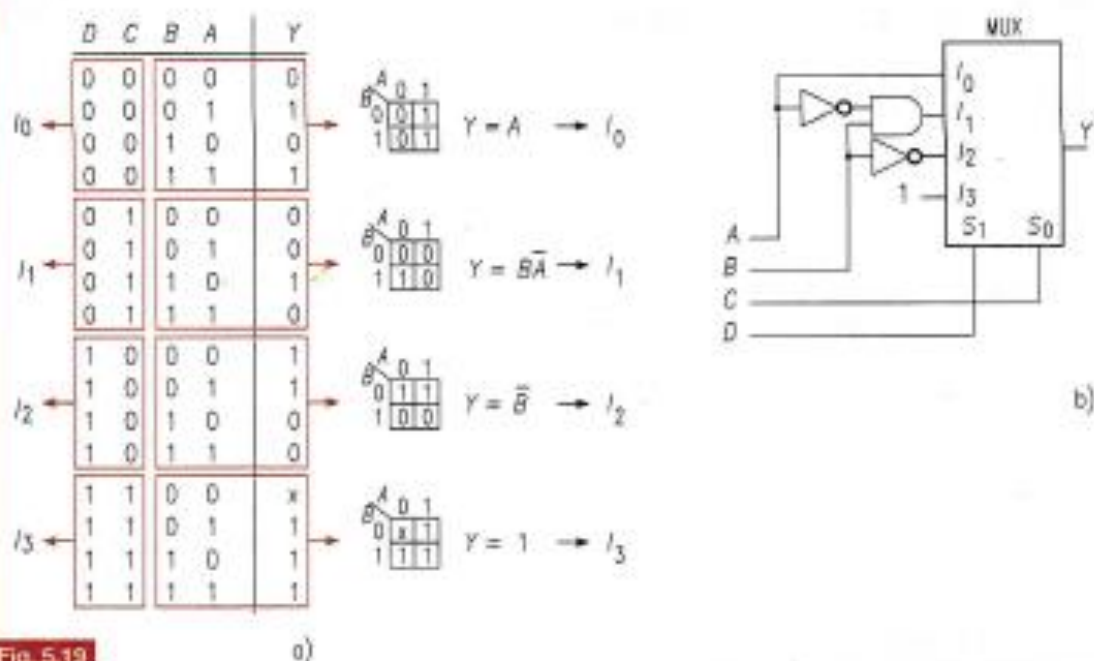


Fig. 5.19

Procedendo in maniera simile è possibile sintetizzare funzioni con tre o quattro variabili in più rispetto agli ingressi di selezione; questo comporta una rete esterna complessa, ma consente di affrontare il progetto di circuiti con un numero di variabili d'ingresso elevato, non risolvibili con il semplice uso delle mappe di Karnaugh.

Ad esempio è possibile realizzare funzioni con sette variabili d'ingresso, utilizzando un mux con quattro ingressi di selezione e portando ai sedici ingressi dati le combinazioni delle tre variabili d'ingresso meno significative, ricavate con la tecnica descritta.

Il quadro globale delle tecniche per il progetto di reti logiche combinatorie, descritte nel presente paragrafo e nel cap. 3, è riportato in fig. 5.20.

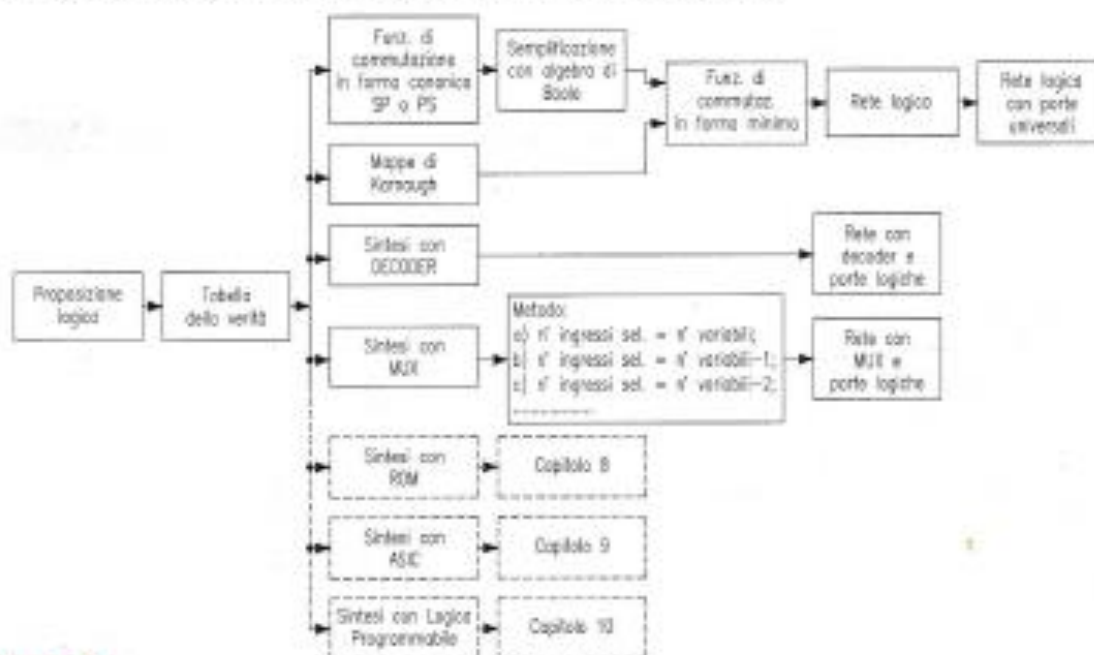


Fig. 5.20

Schema complessivo delle tecniche di progetto di reti combinatorie.