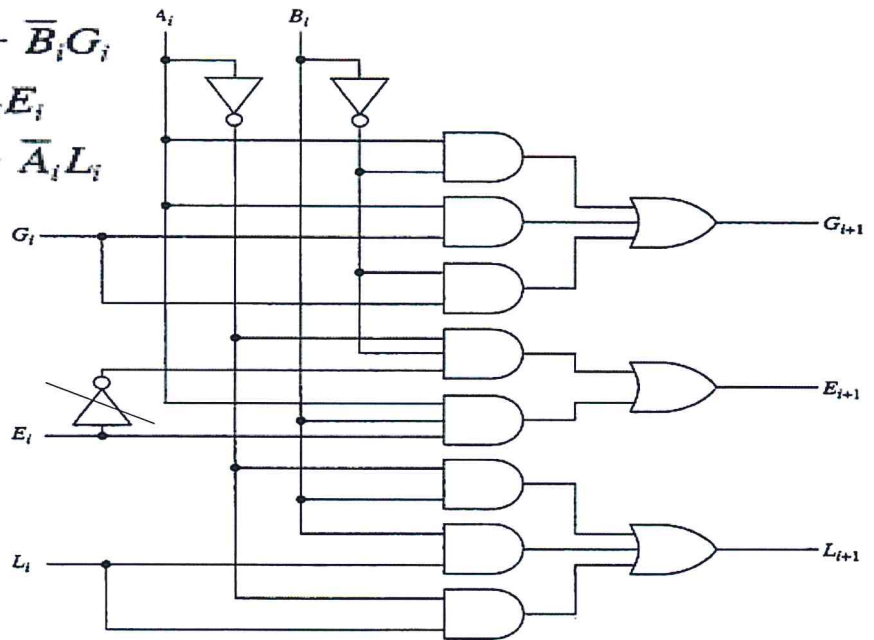


L'elemento di confronto (2)

$$G_{i+1} = A_i \bar{B}_i + A_i G_i + \bar{B}_i G_i$$

$$E_{i+1} = \bar{A}_i \bar{B}_i E_i + A_i B_i E_i$$

$$L_{i+1} = \bar{A}_i B_i + B_i L_i + \bar{A}_i L_i$$



17

Altre RETI COMBINATORIE
Decoder, encoder, ecc

14.1

Decoder

- › Esplicita l'informazione contenuta in un codice
 - › Numero di uscite **maggiore** del numero di ingressi
 - › Con l'eccezione dei codici con ridondanza
 - › In cui sono tolti i bit aggiunti per rendere più sicura la trasmissione
- › Esempi significativi
 - › Decoder $n \rightarrow 2^n$
 - › Decodificatore "generale"
 - › Rete che seleziona una linea tra quelle che possono indirizzare gli ingressi di selezione
 - › Esiste la versione con abilitazione
 - › Decoder specifici: decoder BCD - 7 segmenti
 - › Pilota i segmenti di un display numerico

Un esempio: decoder 3 a 8 (1)

Inputs			Outputs							
x_2	x_1	x_0	z_0	z_1	z_2	z_3	z_4	z_5	z_6	z_7
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

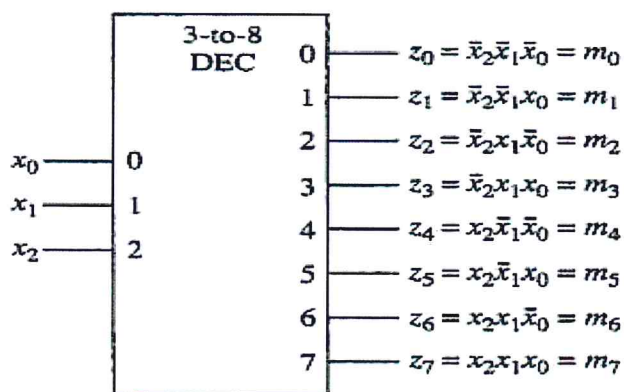
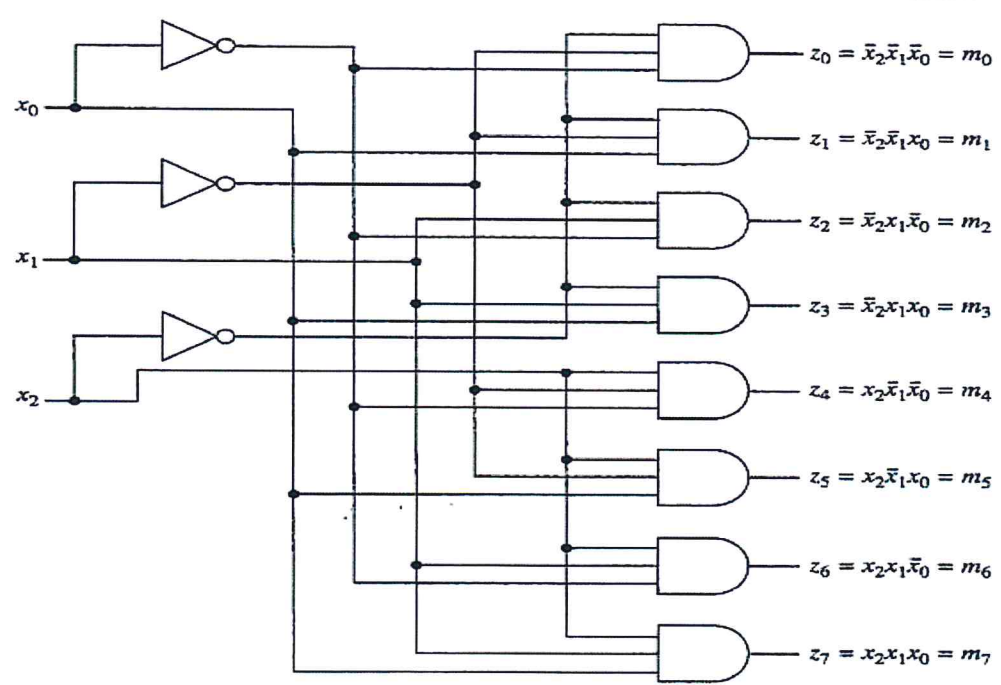


Tabella di verità

Simbolo grafico a blocco

Un esempio: decoder 3 a 8 (2)



Un esempio: BCD-7segmenti

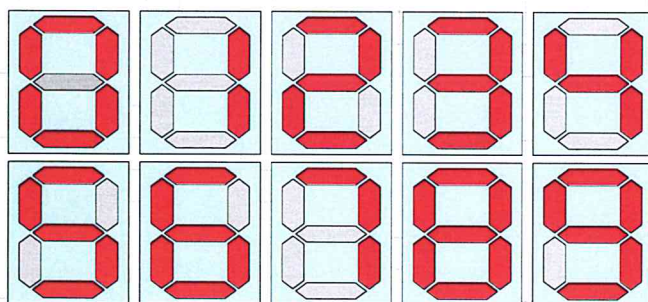
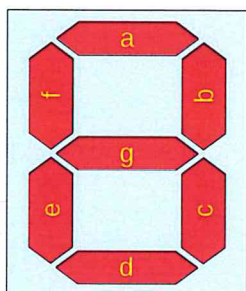


Tabella di verità

X_3	X_2	X_1	X_0	a	b	c	d	e	f	g
0	0	0	0	1	1	1	1	1	1	0
0	0	0	1	0	1	1	0	0	0	0
0	0	1	0	1	1	0	1	1	0	1
0	0	1	1	1	1	1	1	0	0	1
0	1	0	0	0	1	1	0	0	1	1
0	1	0	1	1	0	1	1	0	1	1
0	1	1	0	1	0	1	1	1	1	1
0	1	1	1	1	1	1	0	0	0	0
1	0	0	0	1	1	1	1	1	1	1
1	0	0	1	1	1	1	1	0	1	1



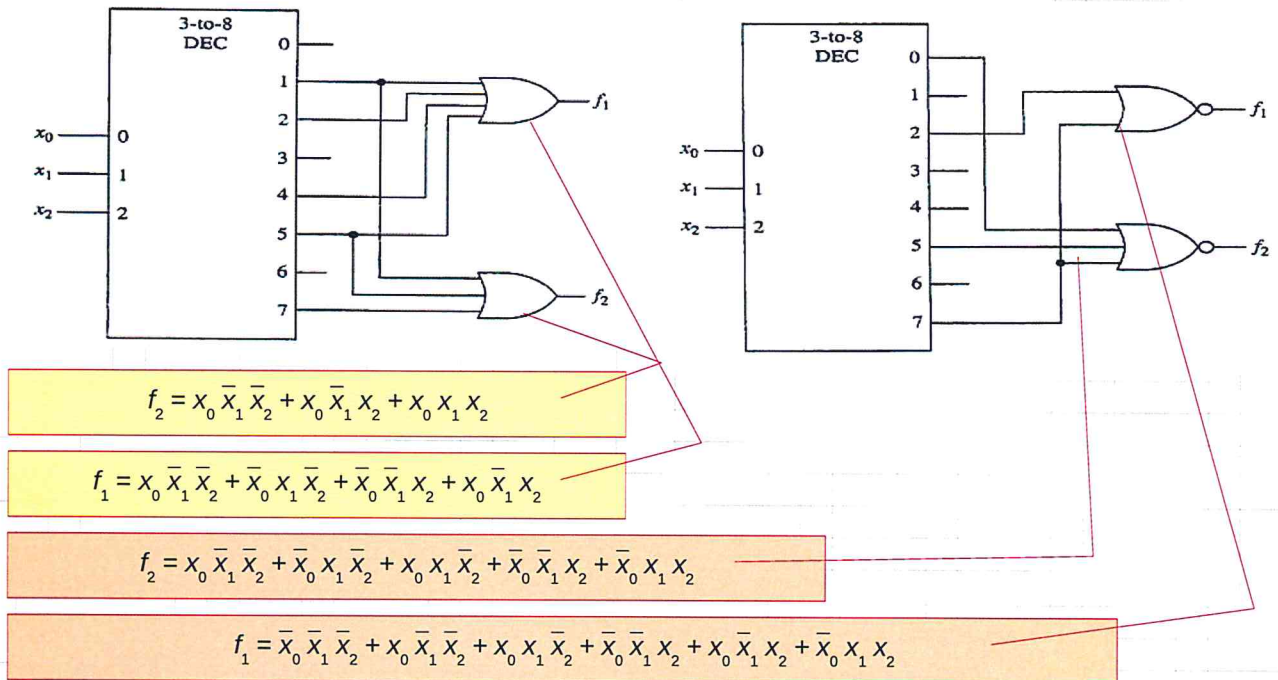
hex - 7 segment -

21

Logiche con decoder (1)

- › I decoder $n \rightarrow 2^n$ possono essere usati per realizzare logiche generiche
 - › Le uscite del decoder corrispondono ai **mintermini**
 - › Con una OR si realizza la forma canonica SP
 - › Se i mintermini sono **più della metà** delle uscite del decoder conviene usare una NOR
 - › I cui ingressi sono i mintermini non presenti nella forma SP
 - › Quindi per gli ingressi che li verificano, la funzione deve essere 0
- › Può essere un modo conveniente di ottenere funzioni a molte uscite
 - › Con **un solo decoder** e diverse porte OR/NOR

Logiche con decoder (2)



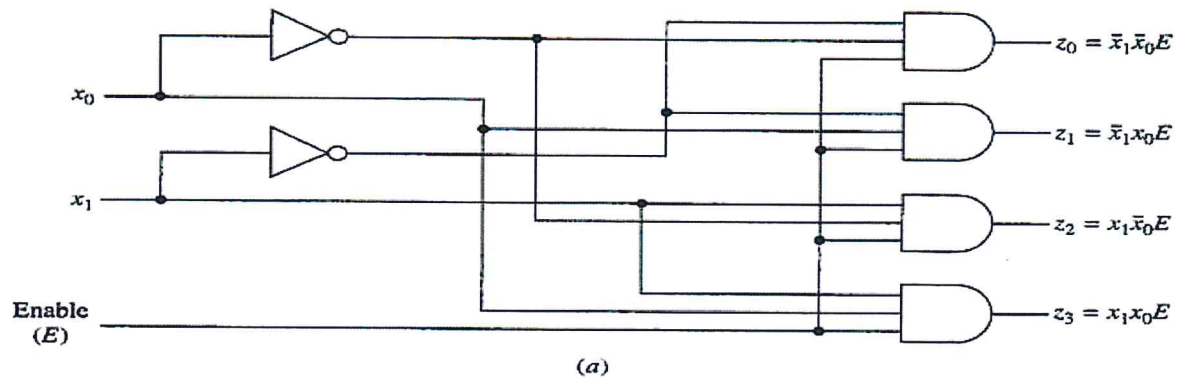
23

Decoder con abilitazione (1)

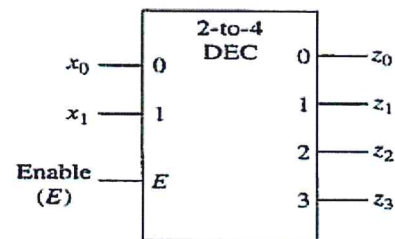
- Oltre agli ingressi di selezione, dispone di un ingresso generale E che abilita la selezione stessa
 - Se $E = 0$ tutte le linee di uscita sono nulle
 - Aumenta la versatilità del blocco; in qualche caso gli ingressi di abilitazione sono più di uno
 - Permette la modularità, aumentando il numero di linee selezionabili

24

Decoder con abilitazione (2)

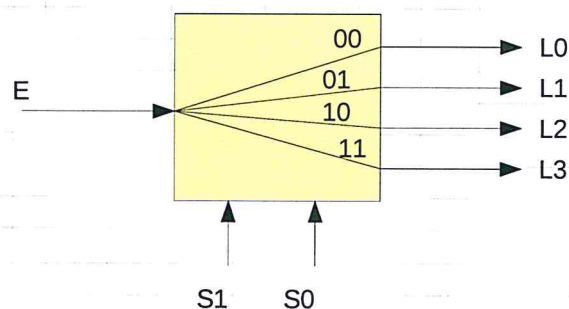


Inputs			Outputs			
<i>E</i>	x_1	x_0	z_0	z_1	z_2	z_3
0	×	×	0	0	0	0
1	0	0	1	0	0	0
1	0	1	0	1	0	0
1	1	0	0	0	1	0
1	1	1	0	0	0	1

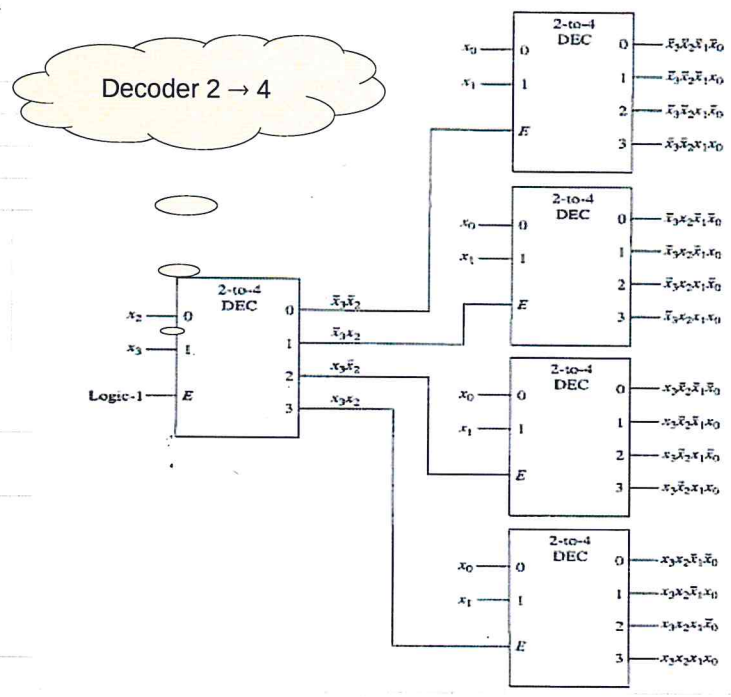


Decoder con abilitazione (3)

- Il decoder con abilitazione è chiamato anche **demultiplexer** (smistatore)
 - Le linee di selezione indicano su quale linea viene **instradato** il segnale di abilitazione
 - Ha un simbolo specifico



Modularità dei decoder



Decoder 2 → 4

Decoder 4 → 16

Per ottenere un decoder 6 → 64 si possono aggiungere altri 16 decoder 2 → 4.

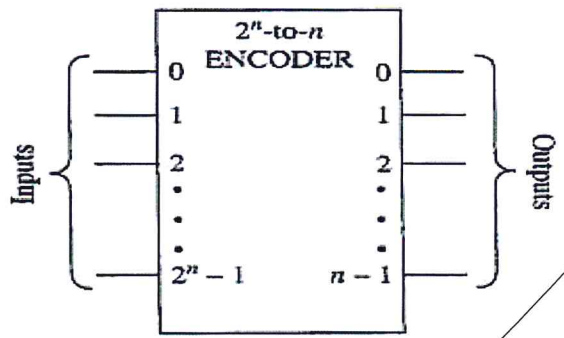
Decoder necessari:
 1 per 4 uscite
 5 per 16 uscite
 21 per 64, 85...

In generale, per ottenere un decoder $2n \rightarrow 2^{(2n)}$ occorrono $[2^{(2n)} - 1] / 3$ decoder 2 → 4.

Encoder

- Codifica l'informazione insita in forma espansa nel mondo reale
 - Numero di uscite **minore** numero di ingressi
 - Con l'eccezione dei codici con ridondanza
 - In cui sono aggiunti bit al dato di partenza
 - Per la correzione degli errori di trasmissione
- Esempi significativi
 - Encoder $2^n \rightarrow n$ con priorità
 - Rete che identifica una linea attiva tra n possibili
 - In caso di più linee attive attribuisce una priorità
 - Esiste la versione con abilitazione
 - Encoder specifici
 - Applicazioni di compressione audio/video
 - Compressione file

Encoder $2^n \rightarrow n$



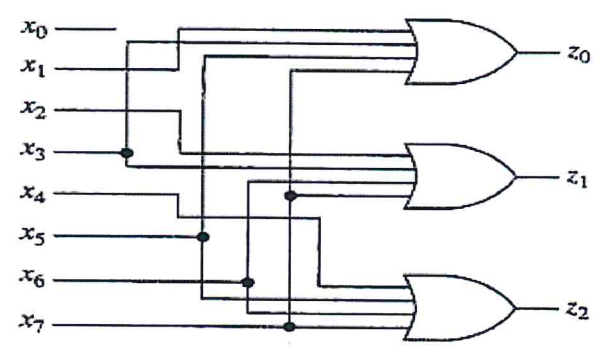
Porta a 1 le cifre corrispondenti al suo numero d'ordine. È attiva una sola linea di ingresso alla volta.

Se più linee sono attive, il risultato non è significativo

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

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

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



Encoder con priorità (8 a 3)

Linee di ingresso

Uscite

Ingresso valido

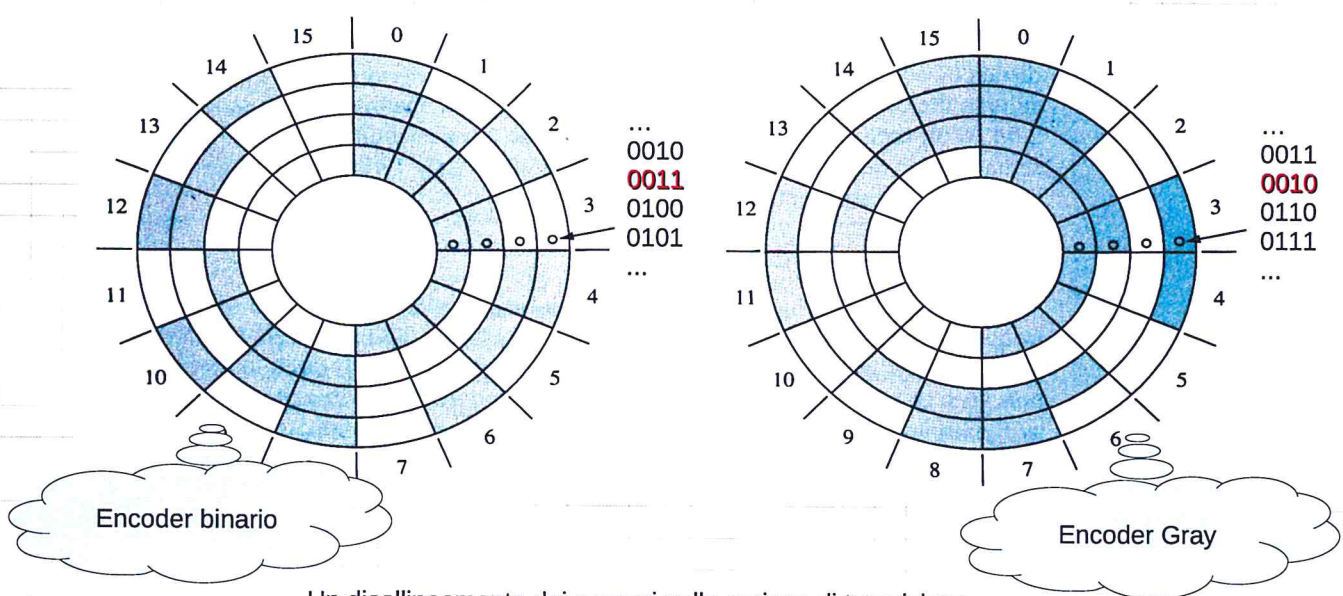
X7	X6	X5	X4	X3	X2	X1	X0	Z2	Z1	Z0	V
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	0	0	0	1
0	0	0	0	0	0	1	x	0	0	1	1
0	0	0	0	0	1	x	x	0	1	0	1
0	0	0	0	1	x	x	x	0	1	1	1
0	0	0	1	x	x	x	x	1	0	0	1
0	0	1	x	x	x	x	x	1	0	1	1
0	1	x	x	x	x	x	x	1	1	0	1
1	x	x	x	x	x	x	x	1	1	1	1

Tabella di verità compressa

Un particolare codice

- Codice Gray
 - Codice ordinato di valori interi binari
 - Ogni valore differisce dal precedente (e quindi dal successivo) per una sola cifra binaria
 - Il numero di cifre differenti tra due binari è la distanza di Hamming
 - In un codice Gray la distanza di Hamming tra numeri vicini è 1
 - Esistono 2^n simboli diversi in un codice Gray
 - L'insieme dei codici Gray può essere messo in **relazione biunivoca** con i numeri binari con lo stesso numero di cifre
- Applicazioni
 - Codifica di posizione lineare o angolare
 - Fotolettura della posizione di un rettangolo o di un cerchio
 - In cui sono disegnati pattern opportuni

En(De)coder Binario-Gray



Un disallineamento dei sensori nella regione di transizione provoca nell'encoder binario errori di grandezza arbitraria nell'encoder Gray solo errori di ± 1

Costruire il codice Gray (1)

- › Costruzione per **induzione**
 - › Un codice Gray a 1 bit G_1 è banale
 - › Dato G_n a n bit, si costruisce G_{n+1} a $(n + 1)$ bit
 - › La prima metà del codice G_{n+1} coincide con G_n
 - › A cui è aggiunto un bit g_n (per esempio come MSB) di valore 0
 - › La seconda metà si ottiene da G_n in **ordine inverso**
 - › E aggiungendo un bit g_n di valore 1
- › Il codice Gray non è unico
 - › Esistono (almeno) $2^n n!$ codici diversi che godono della proprietà che definisce una sequenza di numeri come codice Gray

33

Costruire il codice Gray (2)

Tabella di verità

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
b_3	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
b_2	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
b_1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
b_0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
g_3	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
g_2	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0
g_1	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0
g_0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0

Sintesi **ad hoc**

$$g_3 = b_3$$

$$g_2 = b_3 \oplus b_2$$

$$g_1 = b_2 \oplus b_1$$

$$g_0 = b_1 \oplus b_0$$

$$b_3 = g_3$$

$$b_2 = g_3 \oplus g_2$$

$$b_1 = g_3 \oplus g_2 \oplus g_1$$

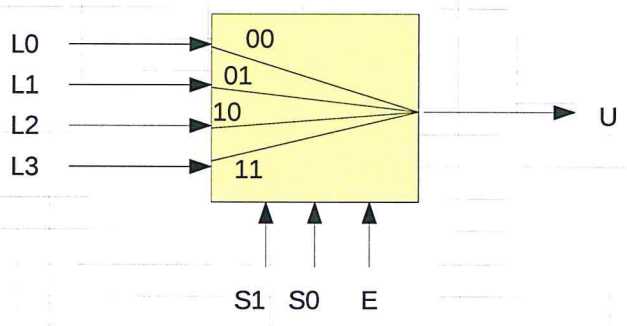
$$b_0 = g_3 \oplus g_2 \oplus g_1 \oplus g_0$$

34

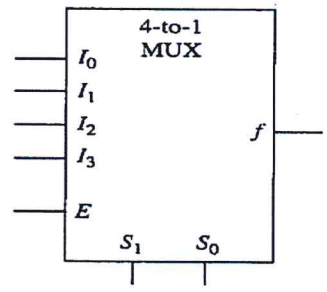
14,0

Multiplexer

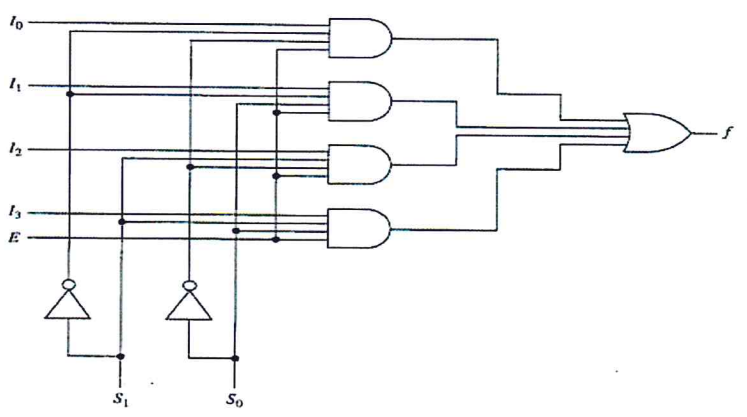
- › Circuito di instradamento
 - › Esegue la funzione inversa del demultiplexer
 - › **Instrada** verso l'uscita il valore della linea selezionata
 - › Ha quindi n ingressi di selezione per 2^n linee di ingresso
 - › Può avere una linea di abilitazione
 - › Che fissa a 0 il valore dell'uscita, se non attiva



Multiplexer



E	S_1	S_0	I_0	I_1	I_2	I_3	f
0	x	x	x	x	x	x	0
1	0	0	0	x	x	x	0
1	0	0	1	x	x	x	1
1	0	1	x	0	x	x	0
1	0	1	x	1	x	x	1
1	1	0	x	x	0	x	0
1	1	0	x	x	1	x	1
1	1	1	x	x	x	0	0
1	1	1	x	x	x	1	1

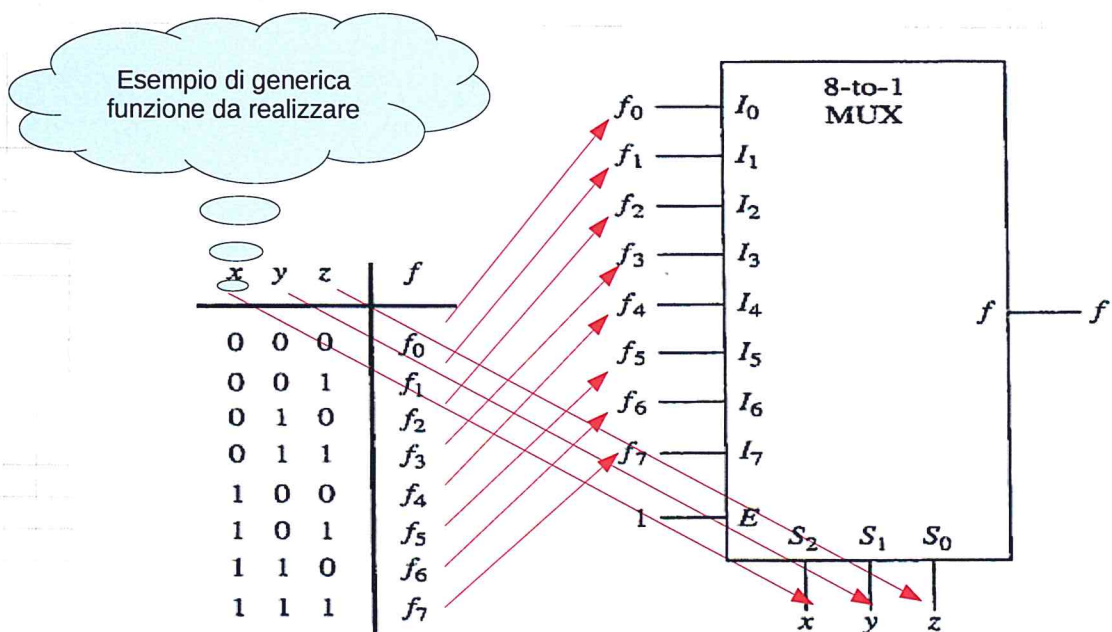


Logiche con multiplexer (1)

- › Come il decoder, il **mux** puà essere usato per realizzare logiche generiche
 - › In modo diretto
 - › Applicando gli ingressi della rete da realizzare alle linee di selezione e mettendo delle costanti in ingresso
 - › In modo composto
 - › Applicando gli ingressi della rete sia alle linee di selezione sia, parzialmente, alle linee dei dati
 - › In questo caso è necessario disporre del dato affermato e negato

37

Logiche con multiplexer (2)

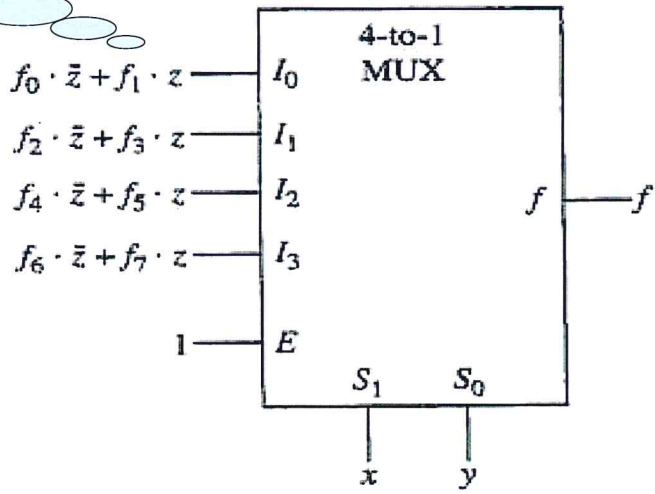


38

Logiche con multiplexer (3)

A avendo una variabile anche in forma negata si può risparmiare un ingresso nel mux

Ogni ingresso può valere 0, 1, z oppure \bar{z}



Logiche con multiplexer (4)

Stesso esempio precedentemente realizzato con mux 8 a 1

x	y	z	f
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	0

