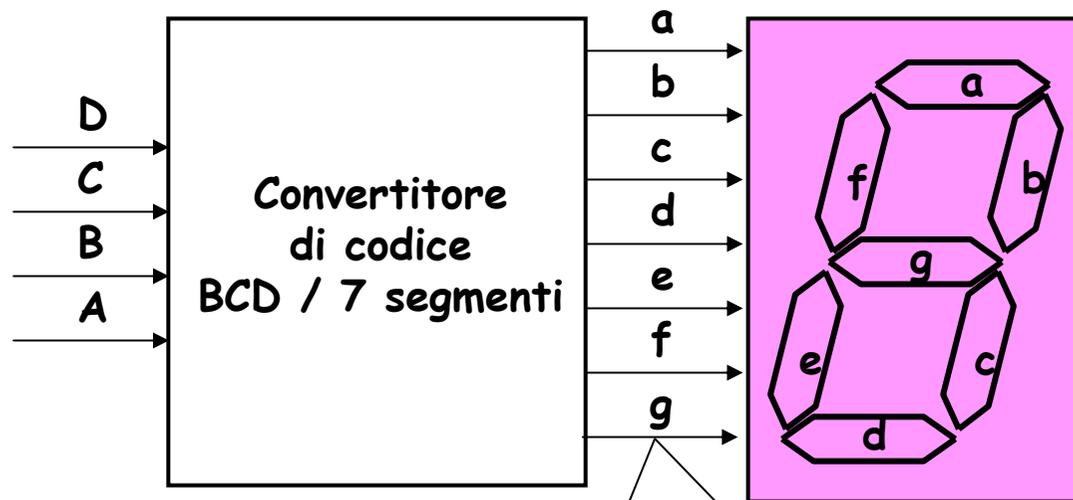


A bright yellow starburst shape with a black outline, centered on the page. The starburst has multiple sharp points of varying lengths, creating a jagged, sunburst-like appearance.

**Reti
combinatorie**

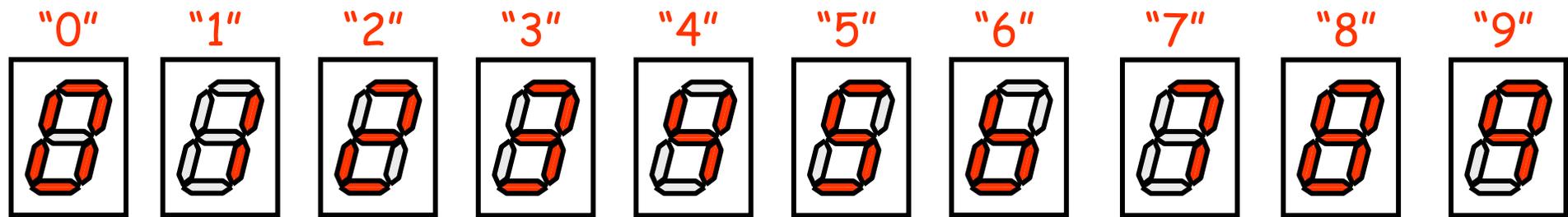
Esercizio 1

Si esegua il progetto di un convertitore di codice BCD / 7 segmenti.



0: segmento acceso
1: segmento spento

D	C	B	A	a	b	c	d	e	f	g
0	0	0	0	0	0	0	0	0	0	1
0	0	0	1	1	0	0	1	1	1	1
0	0	1	0	0	0	1	0	0	1	0
0	0	1	1	0	0	0	0	1	1	0
0	1	0	0	1	0	0	1	1	0	0
0	1	0	1	0	1	0	0	1	0	0
0	1	1	0	1	1	0	0	0	0	0
0	1	1	1	0	0	0	1	1	1	1
1	0	0	0	0	0	0	0	0	0	0
1	0	0	1	0	0	0	1	1	0	0



Il metodo di sintesi della rete di costo minimo basato su mappe di Karnaugh ...

		BA			
		00	01	11	10
DC	00	0	1	0	0
	01	1	0	0	1
	11	-	-	-	-
	10	0	0	-	-

a

$$a = D'C'B'A + CA'$$

		BA			
		00	01	11	10
DC	00	0	0	0	0
	01	0	1	0	1
	11	-	-	-	-
	10	0	0	-	-

b

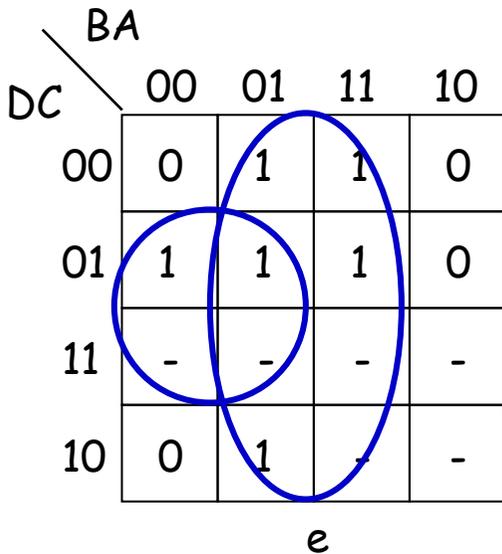
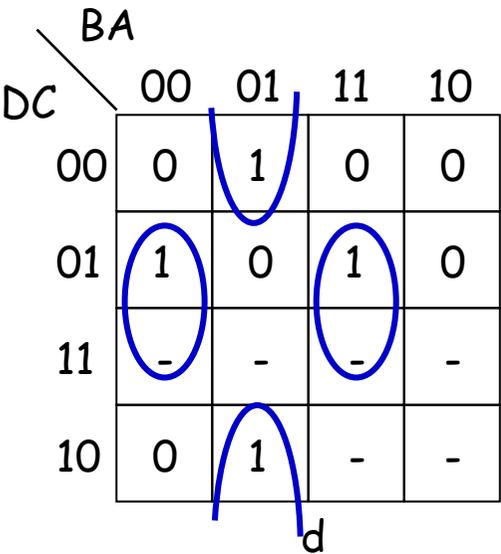
$$b = CB'A + CBA'$$

		BA			
		00	01	11	10
DC	00	0	0	0	1
	01	0	0	0	0
	11	-	-	-	-
	10	0	0	-	-

c

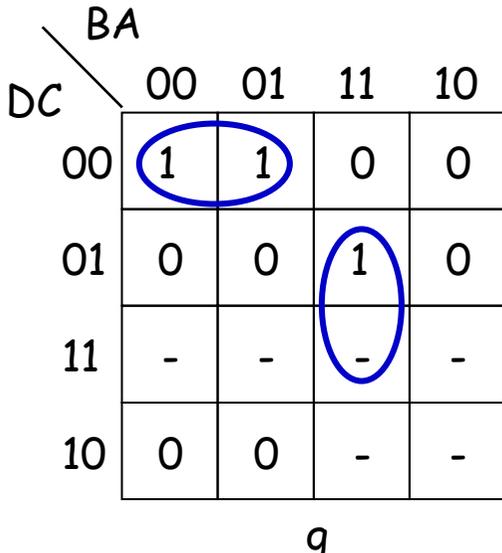
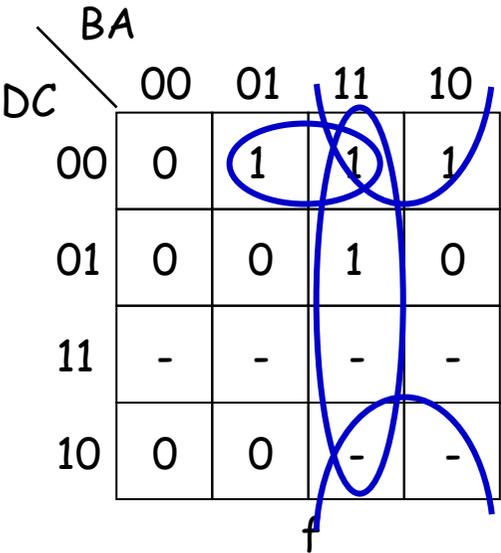
$$c = C'BA'$$

... Il metodo di sintesi della rete di costo minimo basato su mappe di Karnaugh



$$d = CB'A' + C'B'A + CBA$$

$$e = A + CB'$$

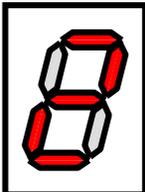
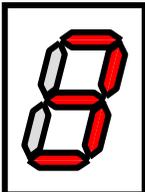
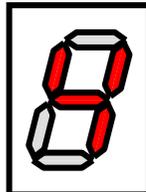
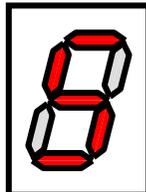
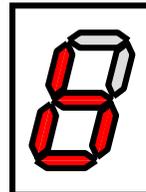
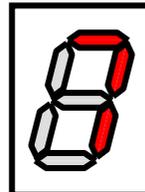


$$f = D'C'A + BA + C'B$$

$$g = D'C'B' + CBA$$

Risposta della rete di costo minimo a configurazioni non previste dal codice BCD

	DCBA					
	1010	1011	1100	1101	1110	1111
a	0	0	1	0	1	0
b	0	0	0	1	1	0
c	1	0	0	0	0	0
d	0	0	1	0	0	1
e	0	1	1	1	0	1
f	1	1	0	0	0	1
g	0	0	0	0	0	1

					
---	---	--	---	---	---



la rete di costo minimo non consente la rilevazione
di alcuna configurazione di ingresso "illecita"

Progetto della rete in grado di rilevare le configurazioni di ingresso illecite ...

Alle configurazioni illecite devono corrispondere sul display simboli diversi da quelli previsti per le configurazioni lecite; in particolare il display deve essere spento per la configurazione DCBA = 1111.

Quest'ultima specifica richiede di risintetizzare soltanto le funzioni a, b, c.

	BA			
DC	00	01	11	10
00	0	1	0	0
01	1	0	0	1
11	-	-	1	-
10	0	0	-	-

a

$$a = D'C'B'A + CA'$$

	BA			
DC	00	01	11	10
00	0	0	0	0
01	0	1	0	1
11	-	-	1	-
10	0	0	-	-

b

$$b = CB'A + CBA'$$

	BA			
DC	00	01	11	10
00	0	0	0	1
01	0	0	0	0
11	-	-	1	-
10	0	0	-	-

c

$$c = C'BA'$$

$$a_1 = a + DC$$

$$a_2 = a + DB$$

$$b_1 = b + DC$$

$$b_2 = b + DB$$

$$c_1 = c + DC$$

$$c_2 = c + DB$$

... Progetto della rete in grado di rilevare le configurazioni di ingresso illecite

In dipendenza delle espressioni selezionate per le funzioni a, b, c si ottengono così otto reti (R_1, R_2, \dots, R_8), tutte della medesima complessità, caratterizzate dallo stesso comportamento sia per le configurazioni di ingresso previste dal codice BCD, sia per la configurazione DCBA = 1111 (display spento).

$$R_1 : a_1 b_1 c_1 d e f g$$

$$R_5 : a_2 b_1 c_1 d e f g$$

$$R_2 : a_1 b_1 c_2 d e f g$$

$$R_6 : a_2 b_1 c_2 d e f g$$

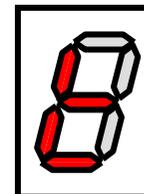
$$R_3 : a_1 b_2 c_1 d e f g$$

$$R_7 : a_2 b_2 c_1 d e f g$$

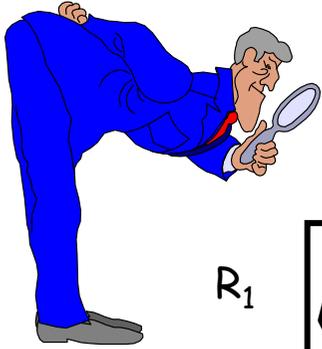
$$R_4 : a_1 b_2 c_2 d e f g$$

$$R_8 : a_2 b_2 c_2 d e f g$$

Il comportamento è lo stesso anche per la configurazione DCBA = 1110, poiché i segmenti a, b, c sono comunque spenti, indipendentemente dalle espressioni selezionate. Il simbolo corrispondentemente visualizzato è:



Scelta della rete "ottimale" ...



DCBA

	1010	1011	1100	1101	1110	1111	
R ₁		☹️	☹️				
R ₂		☹️					
R ₃							
R ₄				☹️			
R ₅							
R ₆					☹️		
R ₇							
R ₈				☹️	☹️		

... Scelta della rete "ottimale"



Le reti R_1 , R_2 , R_4 , R_6 e R_8 non consentono la rilevazione di tutte le configurazioni di ingresso illecite.



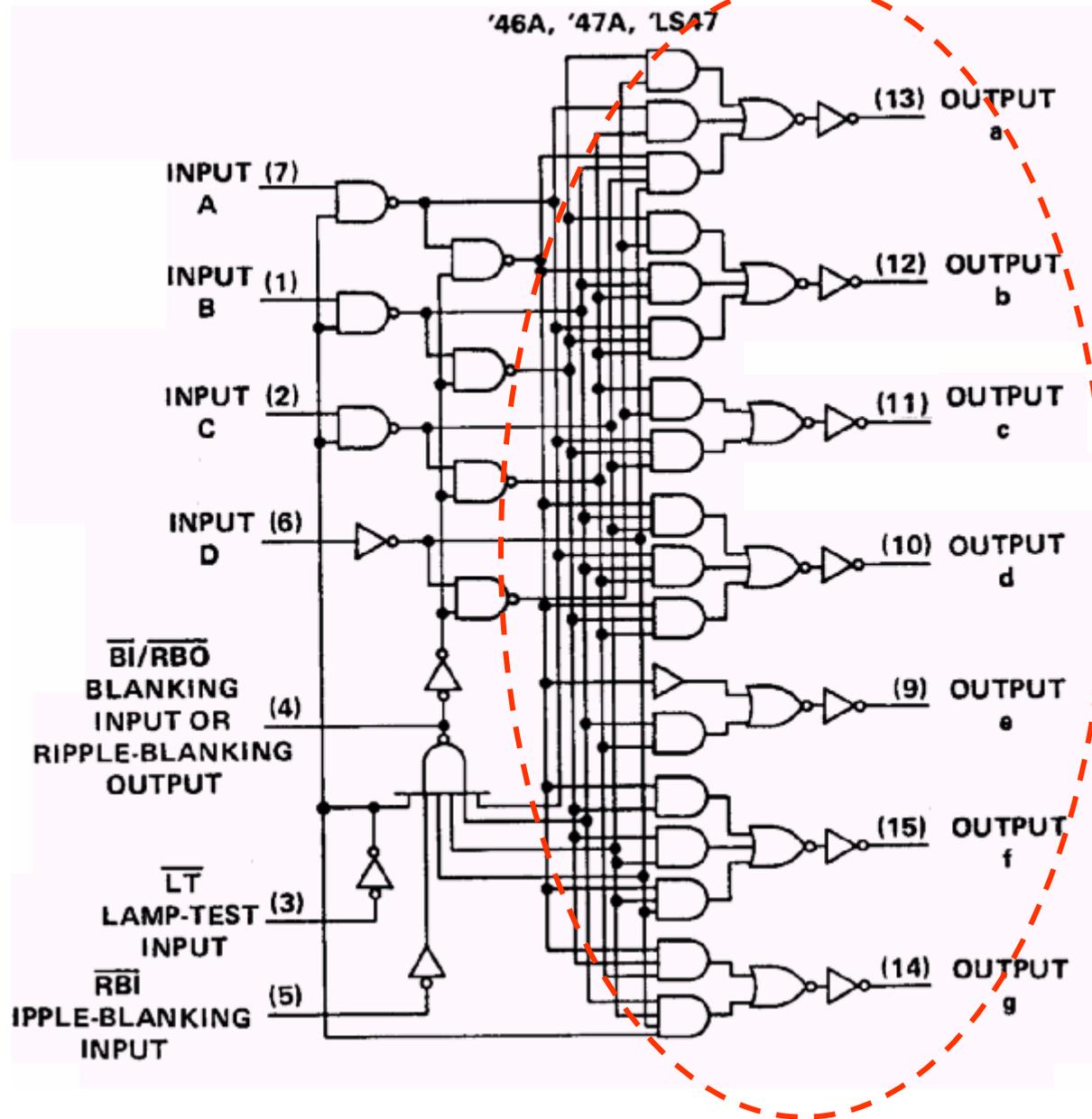
Le reti R_3 , R_5 e R_7 consentono la rilevazione di tutte le configurazioni di ingresso illecite, peraltro con simboli tutti diversi fra loro.



Quale scegliere allora, visto che hanno tutte la medesima complessità e velocità di elaborazione ???

La soluzione R_7 , che richiede un minore consumo di energia (17 segmenti complessivamente accesi, anziché 18) per visualizzare le configurazioni illecite. Questa è la soluzione adottata nei circuiti integrati SN 7446, 7447, 7448, 7449 !!!

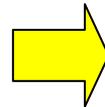
La soluzione integrata ...



... La soluzione integrata ...

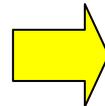
I gate aggiuntivi previsti nella soluzione integrata servono per conseguire ulteriori funzionalità, derivabili da specifici segnali di ingresso-uscita (tutti attivi a livello logico 0), ed elencate in ordine di priorità decrescente:

BI' (Blanking Input)



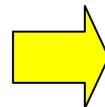
display spento

LT' (Lamp Test)



display acceso

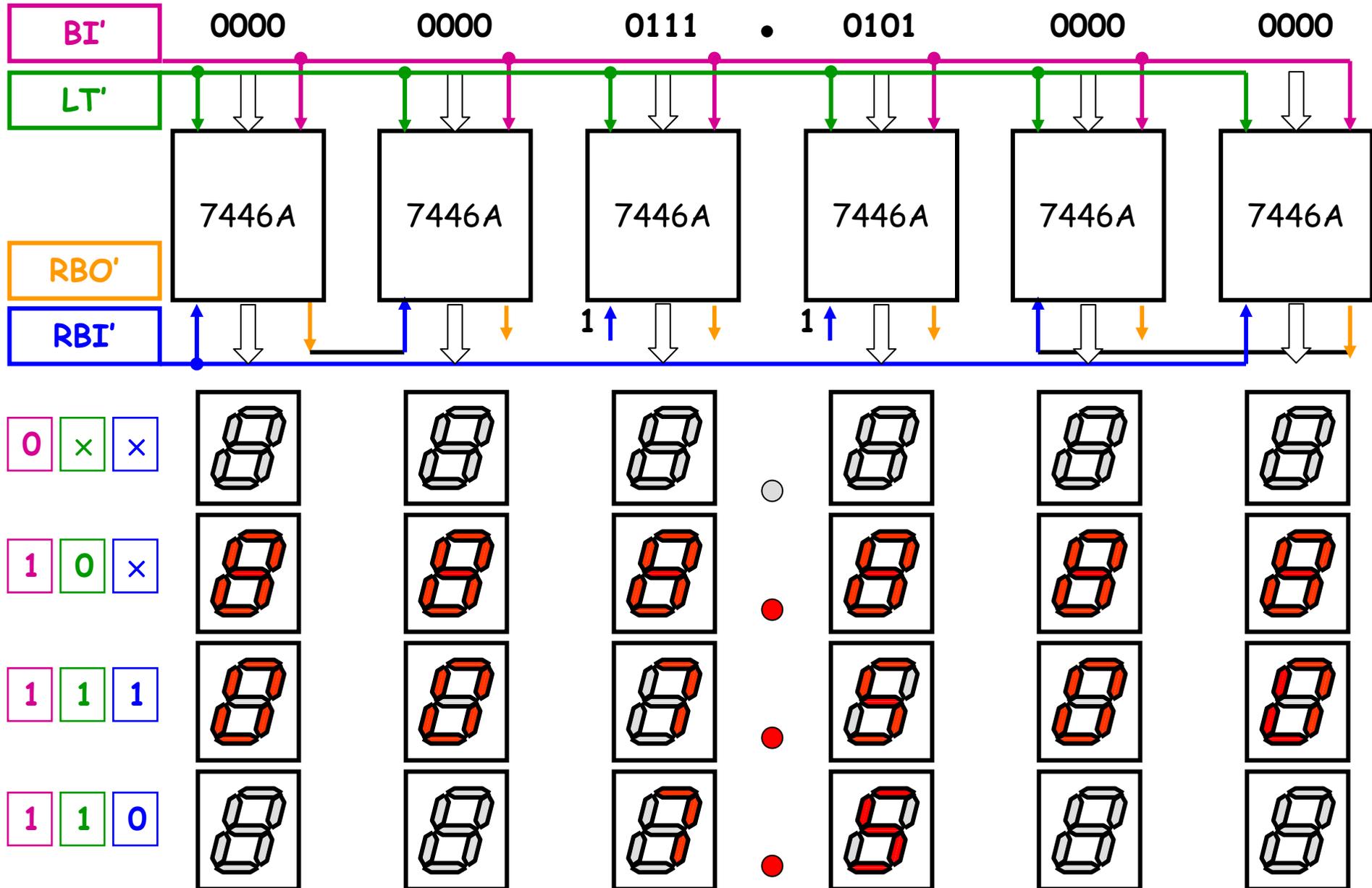
RBI' (Ripple Blanking Input)



display spento
e attivazione del segnale di uscita
RBO' (Ripple Blanking Output)
se il dato in ingresso è zero
(DCBA = 0000)

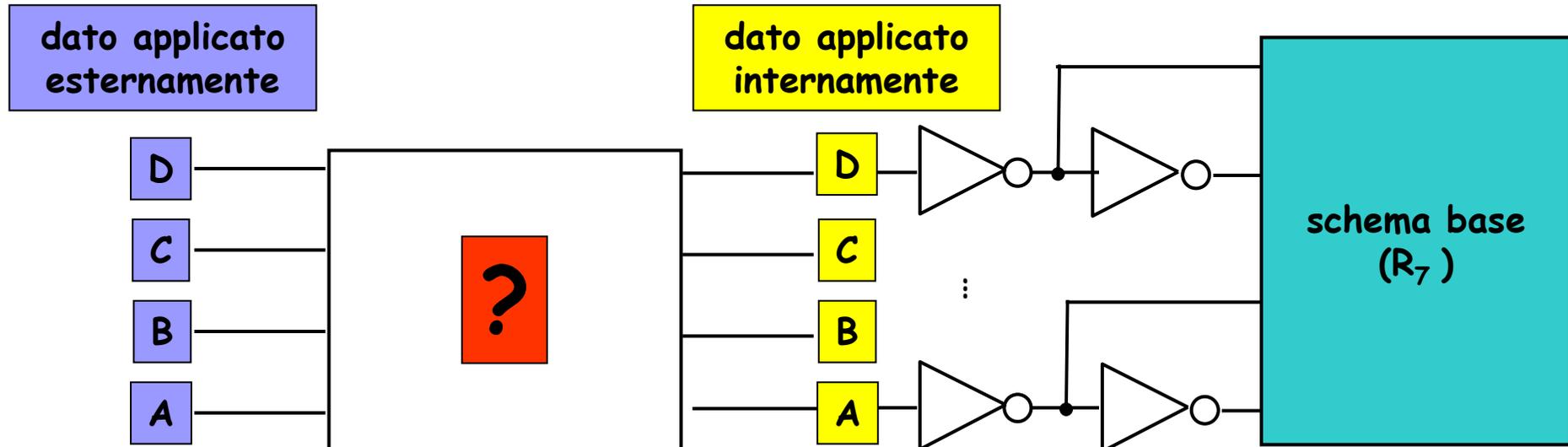
... La soluzione integrata ...

Estensione delle funzionalità e modalità di visualizzazione [es. $(007.500)_{dieci}$]



... La soluzione integrata ...

Estensione dello schema base per la gestione dei segnali ausiliari



BI'	LT'	RBI'	RBO'	D	C	B	A
-----	-----	------	------	---	---	---	---

1	1	1	1	D	C	B	A
---	---	---	---	---	---	---	---

0	×	×	0	1	1	1	1
---	---	---	---	---	---	---	---

display spento

1	0	×	1	1	0	0	0
---	---	---	---	---	---	---	---

display acceso

1	1	0	0	1	1	1	1
---	---	---	---	---	---	---	---

se DCBA = 0000

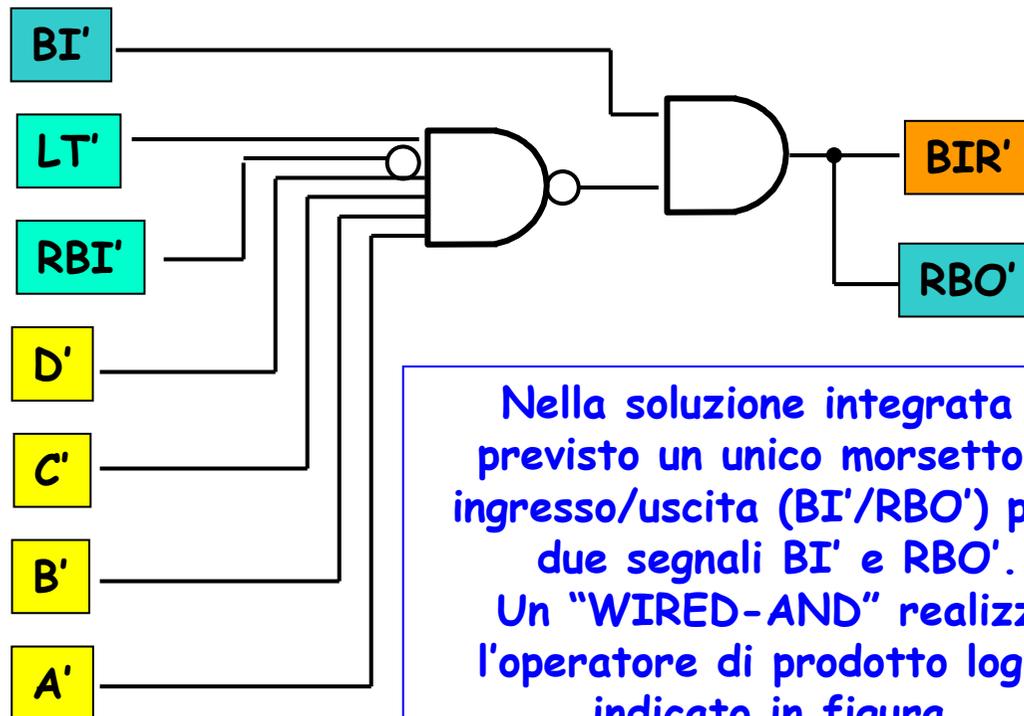
1	1	0	1	D	C	B	A
---	---	---	---	---	---	---	---

se DCBA ≠ 0000

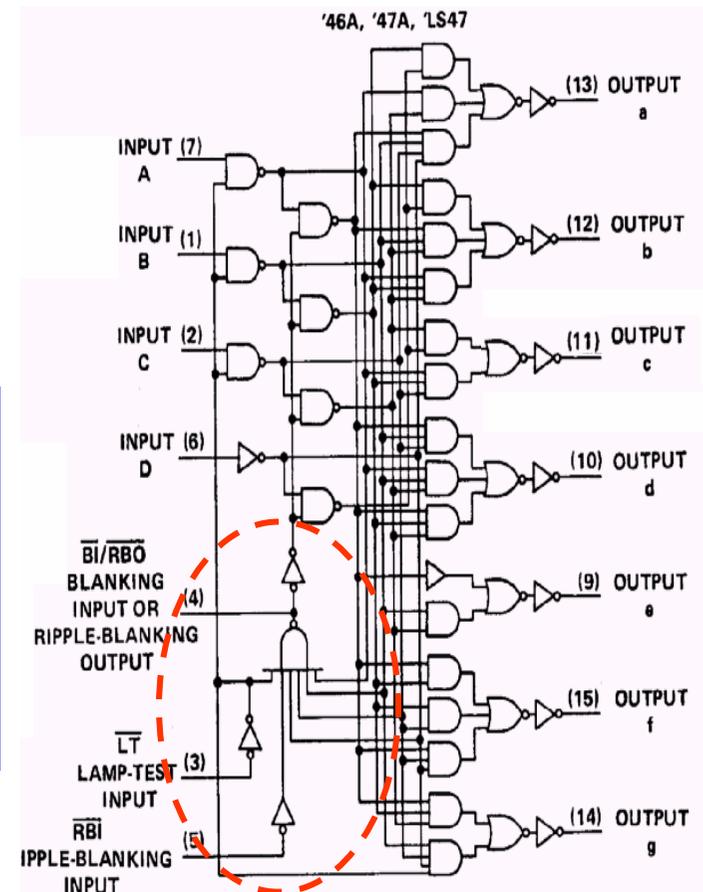
... La soluzione integrata ...

Occorre generare internamente un segnale di richiesta di spegnimento del display (Blanking Input Request (BIR')), attivo (valore logico 0) solo se $BI' = 0$, oppure se $RBI' = 0$, $DCBA = 0000$ e $LT' = 1$ (LT' è prioritario rispetto a RBI').

Corrispondentemente si deve avere $RBO' = 0$.

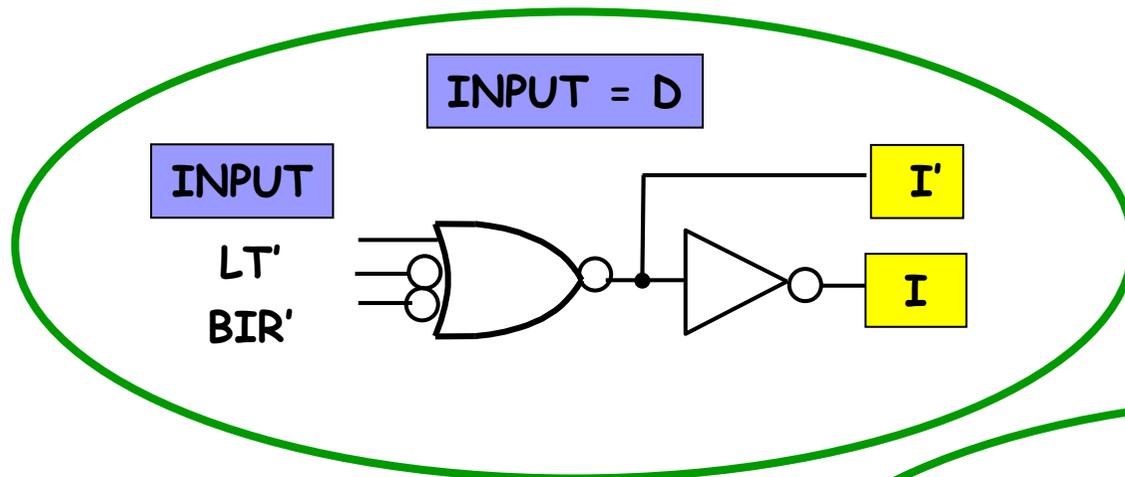


Nella soluzione integrata è previsto un unico morsetto di ingresso/uscita (BI'/RBO') per i due segnali BI' e RBO' . Un "WIRED-AND" realizza l'operatore di prodotto logico indicato in figura.

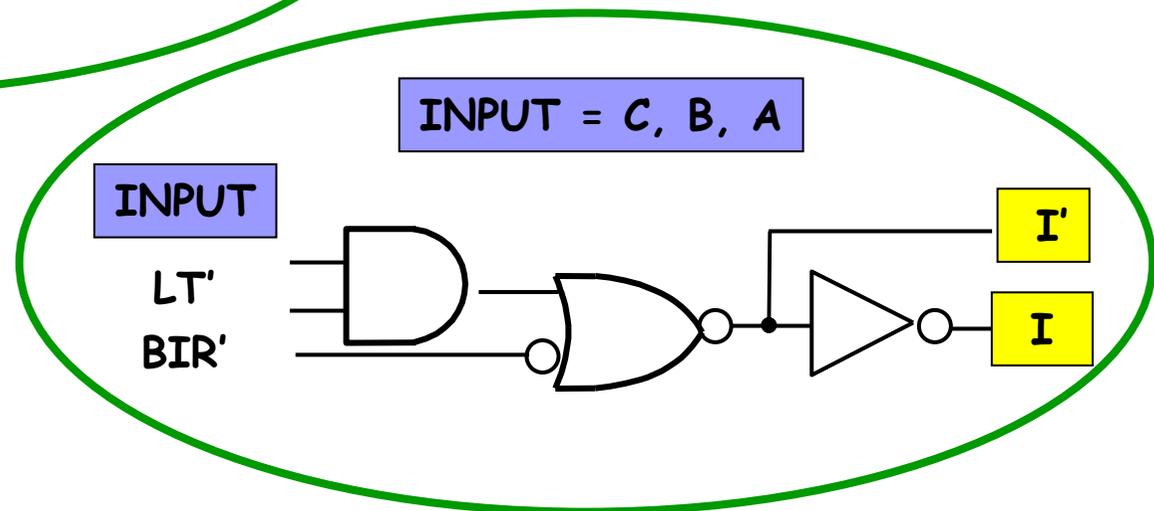


... La soluzione integrata ...

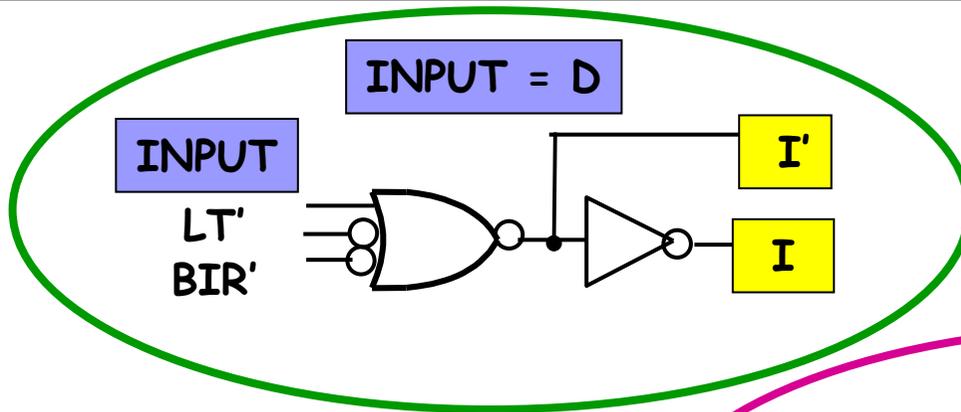
la gestione dei segnali BIR' e LT'



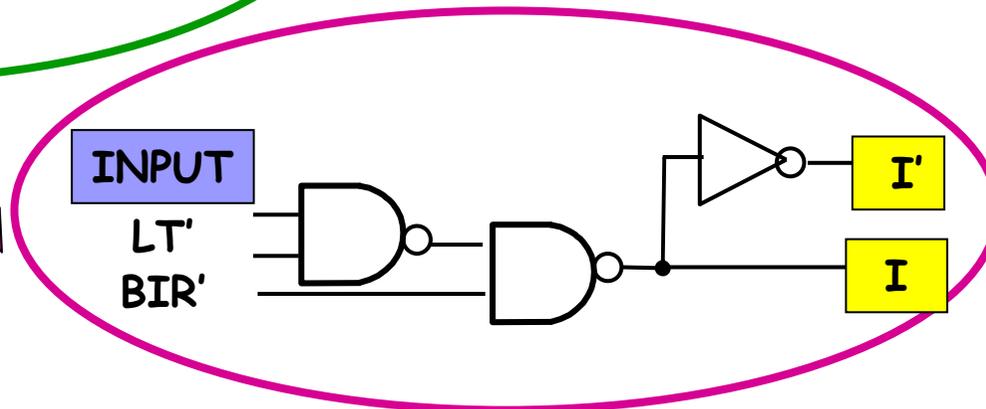
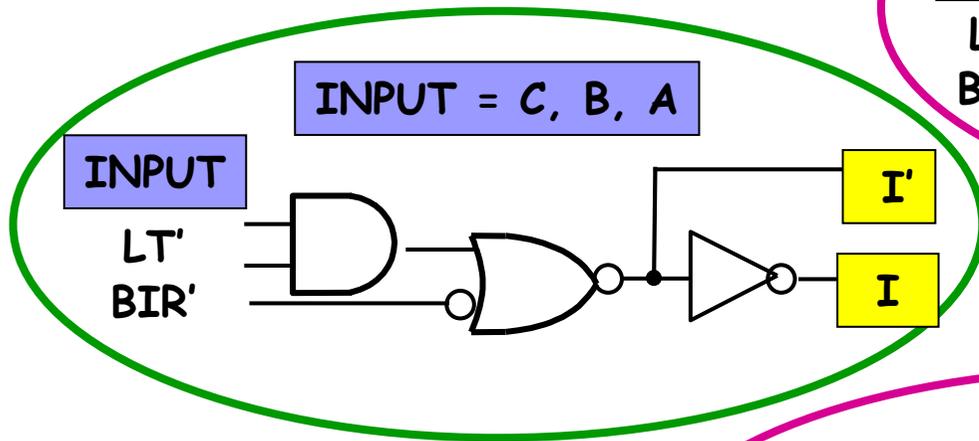
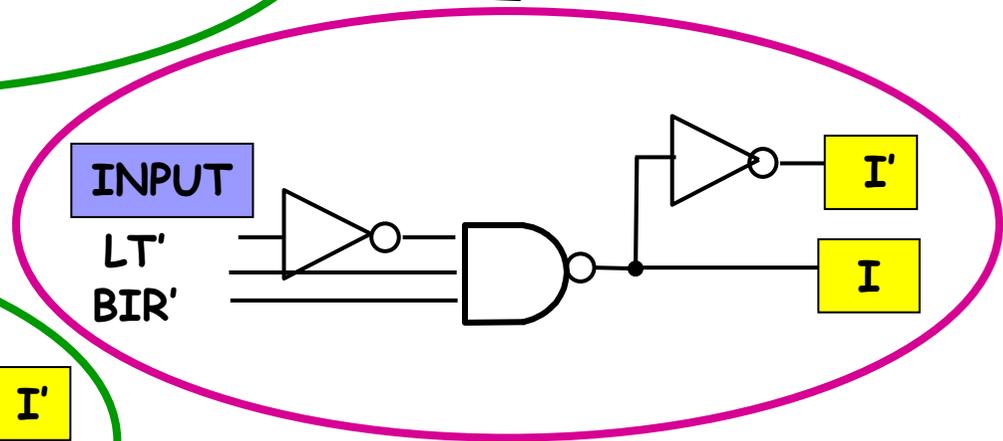
BIR'	LT'	D	C	B	A
1	1	D	C	B	A
0	×	1	1	1	1
1	0	1	0	0	0



... La soluzione integrata ...

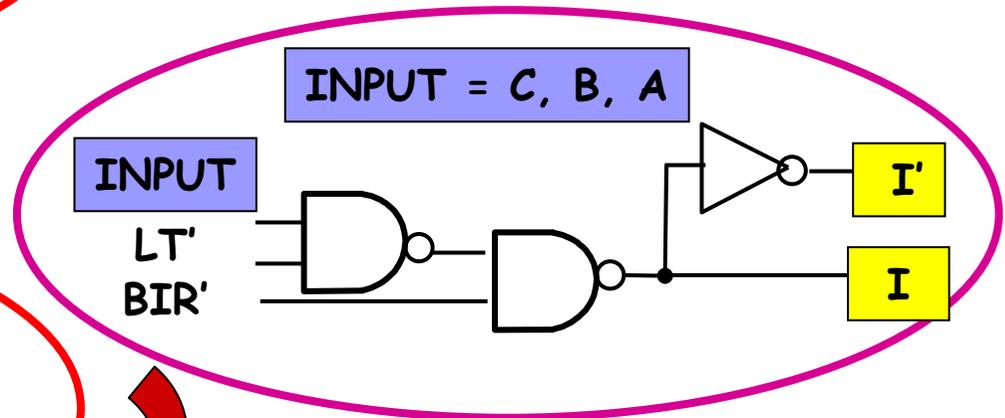
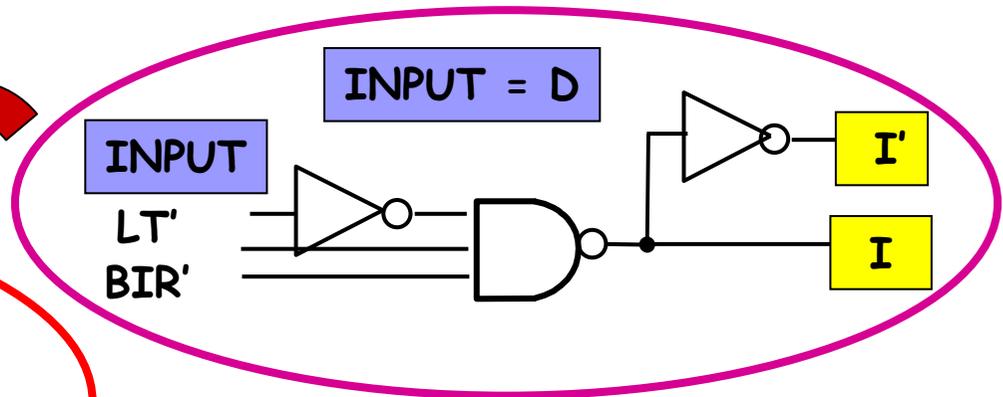
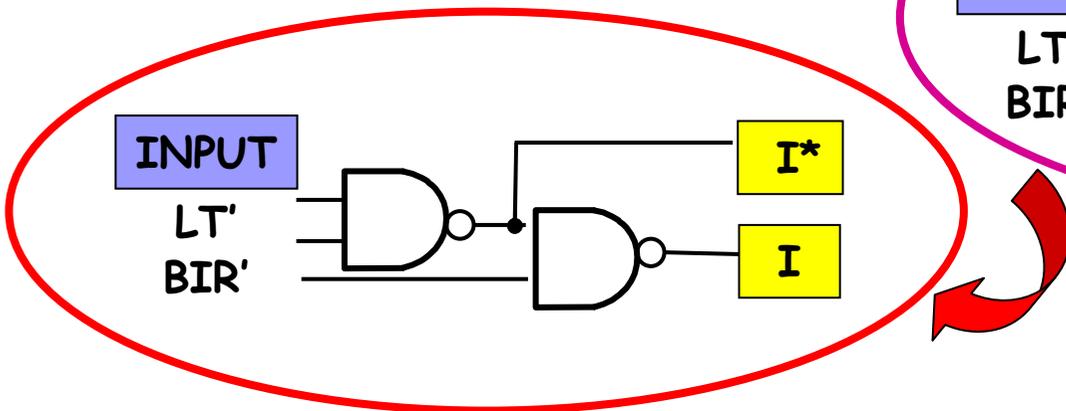
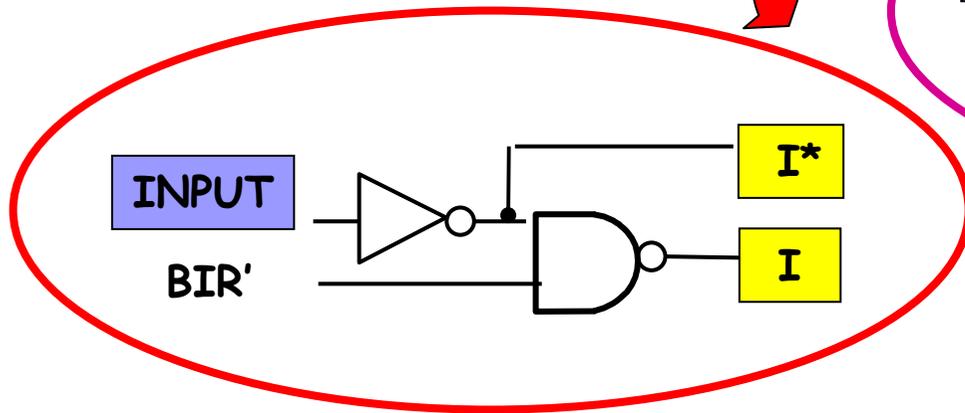


legge di De Morgan



... La soluzione integrata ...

la soluzione integrata



... La soluzione integrata

Solo se $BIR' = 1$

$I^* = I'$



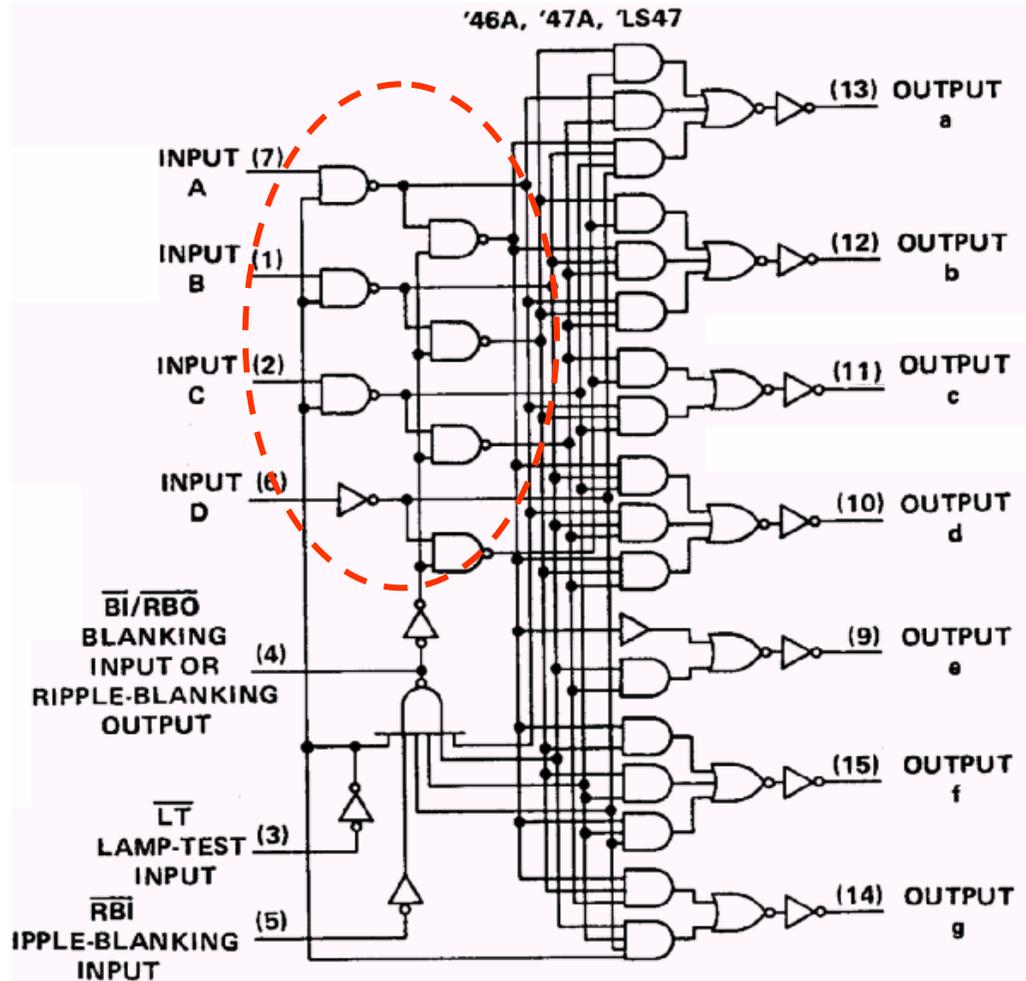
In effetti se $BIR' = 0$

$I^* = -$

poiché

$I = 1$

e nell'espressione di ogni variabile di uscita esiste un implicante in cui tutte le variabili di ingresso intervengono in forma vera
→ display spento

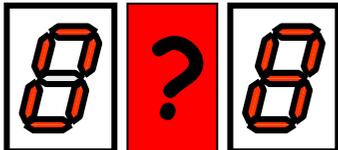


Se $BIR' = 1$ e $LT' = 0$

$C, B, A = 0$

ma non necessariamente

$D = 1$



L'accensione del segmento g è ottenuta intervenendo sullo schema base

$$g = LT' D'C'B' + CBA$$

Esercizio 2

La relazione di ingresso-uscita di una rete combinatoria caratterizzata da cinque segnali di ingresso (a, b, c, d, e) e da un segnale di uscita (z) è definita dalla seguente espressione:

$$z(a,b,c,d,e) = \sum_5 m(2,7,12,13,18,20,24,25,27; 0,3,8,10,16,22,26,28,29,30)$$

o, equivalentemente, dalla seguente mappa di Karnaugh:

Corrispondenza tra le configurazioni delle variabili indipendenti a, b, c, d, e (in base 2 e in base 10) e le celle della mappa:

		cde							
		000	001	011	010	100	101	111	110
ab	00	0	1	3	2	4	5	7	6
	01	8	9	11	10	12	13	15	14
	11	24	25	27	26	28	29	31	30
	10	16	17	19	18	20	21	23	22

		cde							
		000	001	011	010	100	101	111	110
ab	00	-	0	-	1	0	0	1	0
	01	-	0	0	-	1	1	0	0
	11	1	1	1	-	-	-	0	-
	10	-	0	0	1	1	0	0	-

z (a,b,c,d,e)

Si individuino le espressioni minime SP e PS di z applicando i metodi di Petrick e di Quine-McCluskey.

I metodi di Petrick e Quine-McCluskey (espressione SP)

	RR ₀	celle			RR ₁	celle			RR ₂	celle			RR ₃	celle	
g ₀	00000	0	✓	g ₀	000-0	0 2	✓	g ₀	0-0-0	0 2 8 10	✓	g ₀	--0-0	0 2 8 10 16 18 24 26	G ₃
g ₁	00010	2	✓		0-000	0 8	✓		-00-0	0 2 16 18	✓	g ₁	1---0	16 18 20 22 24 26 28 30	H ₃
	01000	8	✓		-0000	0 16	✓		--000	0 8 16 24	✓				
g ₂	10000	16	✓	g ₁	0001-	2 3	A ₁	g ₁	--010	2 10 18 26	✓				
	00011	3	✓		0-010	2 10	✓		-10-0	8 10 24 26	✓				
	01010	10	✓		-0010	2 18	✓		-1-00	8 12 24 28	C ₂				
	01100	12	✓		010-0	8 10	✓		10--0	16 18 20 22	✓				
	10010	18	✓		01-00	8 12	✓		1-0-0	16 18 24 26	✓				
	10100	20	✓		-1000	8 24	✓		1--00	16 20 24 28	✓				
g ₃	11000	24	✓		100-0	16 18	✓	g ₂	-110-	12 13 28 29	D ₂				
	00111	7	✓		10-00	16 20	✓		1--10	18 22 26 30	✓				
	01101	13	✓	g ₂	00-11	3 7	B ₁		1-1-0	20 22 28 30	✓				
	10110	22	✓		-1010	10 26	✓		110--	24 25 26 27	E ₂				
	11001	25	✓		0110-	12 13	✓		11-0-	24 25 28 29	F ₂				
	11010	26	✓		-1100	12 28	✓		11--0	24 26 28 30	✓				
g ₄	11100	28	✓		10-10	18 22	✓								
	11011	27	✓	g ₃	1-010	18 26	✓								
	11101	29	✓		101-0	20 22	✓								
	11110	30	✓		1-100	20 28	✓								
					1100-	24 25	✓								
					110-0	24 26	✓								
					11-00	24 28	✓								
					-1101	13 29	✓								
					1-110	22 30	✓								
					110-1	25 27	✓								
					11-01	25 29	✓								
					1101-	26 27	✓								
					11-10	26 30	✓								
					1110-	28 29	✓								
					111-0	28 30	✓								

STEP 1: Individuazione di tutti i raggruppamenti rettangolari primi

Colonna RR_k (k=0,1,2): insieme dei raggruppamenti rettangolari di ordine k costituiti da celle contenenti il valore 1 o -.

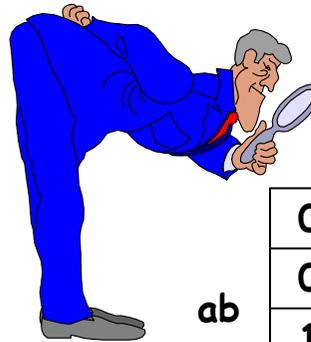
In ogni insieme i raggruppamenti sono partizionati in gruppi g_n (n=0,1,2,...) in base al numero n di variabili che assumono il valore 1 nell'ambito di tutte le celle di un raggruppamento.

I simboli ✓ e ∇_k (∇=A,B,C,...) differenziano i raggruppamenti non primi da quelli primi, rispettivamente.

I metodi di Petrick e Quine-McCluskey (espressione SP)

STEP 2: Costruzione della tabella di copertura

Evidenziazione sulla mappa di Karnaugh (istanziata due volte per motivi di leggibilità) dei raggruppamenti rettangolari primi:
 A_1 B_1 C_2 D_2 E_2 F_2 G_3 H_3



celle da coprire	
numero	sì
	no
-	sì/no

configurazioni (in base 10) delle celle per cui $z = 1$

	2	7	12	13	18	20	24	25	27
A_1	x								
B_1		x							
C_2			x				x		
D_2			x	x					
E_2							x	x	x
F_2							x	x	
G_3	x				x		x		
H_3					x	x	x		

RR primi di celle per cui $z = 1$ o -

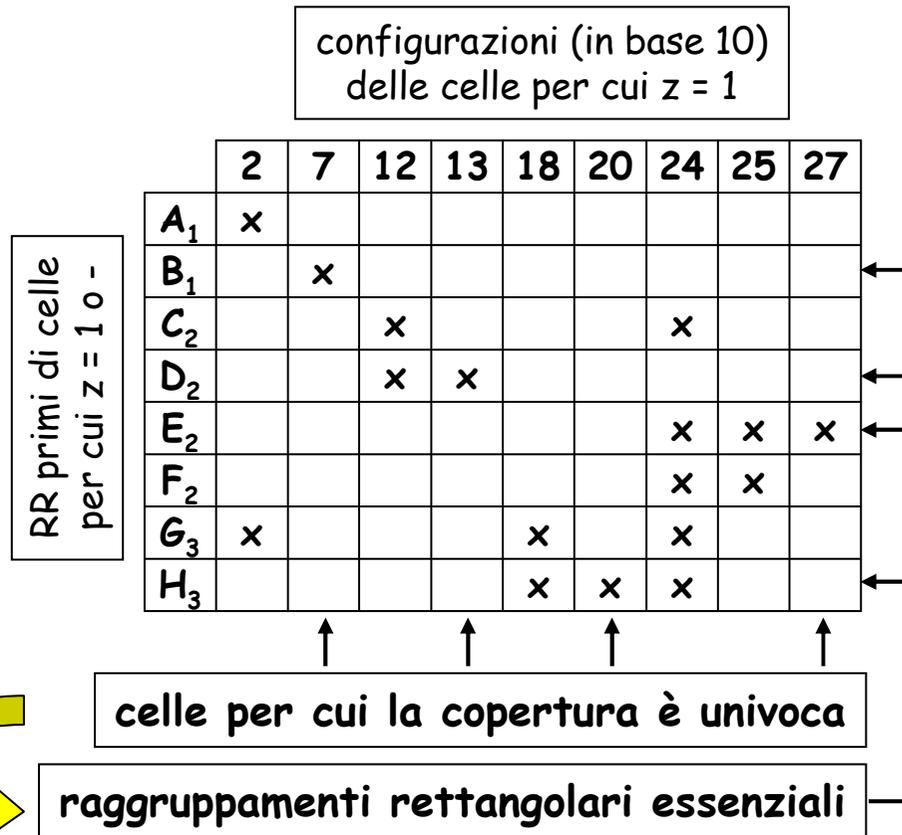
		cde							
		000	001	011	010	100	101	111	110
ab	00	-		-	2			7	
	01	-			-	12	13		
	11	24	25	27	-	-	-		-
	10	-			18	20			-

		cde							
		000	001	011	010	100	101	111	110
ab	00	-		-	2			7	
	01	-			-	12	13		
	11	24	25	27	-	-	-		-
	10	-			18	20			-

Il metodo di Quine-McCluskey (espressione SP)

STEP 3: Soluzione per via grafica del problema della copertura

Selezione ai fini della copertura delle **righe essenziali** $B_1 D_2 E_2 H_3$.
Rimozione dalla tabella di copertura delle righe essenziali e delle colonne 7 12 13 18 20 24 25 27 da esse coperte.



	cde							
	000	001	011	010	100	101	111	110
00	-		-	2			7*	
01	-			-	12	13*		
11	24	25	27*	-	-	-		-
10	-			18	20*			-

Il metodo di Quine-McCluskey (espressione SP)

Tabella di copertura ridotta



		cde							
		000	001	011	010	100	101	111	110
ab	00	-		-	2			7*	
	01	-				12	13*		
	11	24	25	27*	-	-	-		-
	10	-			18	20*			-

celle ancora da coprire

RR primi non essenziali		2
	A ₁	x
	C ₂	
	F ₂	
	G ₃	x

numero/-

celle già coperte

Rimozione delle righe C₂, F₂ (raggruppamenti che coprono celle già coperte dall'unione dei raggruppamenti essenziali).

		cde							
		000	001	011	010	100	101	111	110
ab	00	-		-	2			7*	
	01	-				12	13*		
	11	24	25	27*	-	-	-		-
	10	-			18	20*			-

Selezione del raggruppamento G₃ in quanto di ordine superiore.
Copertura minima: B₁ D₂ E₂ G₃ H₃.

Espressione minima: $z = a'b'de + bcd' + abc' + c'e' + ae'$

Il metodo di Petrick (espressione SP)

STEP 3: Soluzione per via algebrica del problema della copertura

configurazioni (in base 10)
delle celle per cui $z = 1$

	2	7	12	13	18	20	24	25	27
A_1	x								
B_1		x							
C_2			x				x		
D_2			x	x					
E_2							x	x	x
F_2							x	x	
G_3	x				x		x		
H_3					x	x	x		

RR primi di celle
per cui $z = 1$

Coperture irridondanti:

$$(A_1 + G_3) B_1 (C_2 + D_2) D_2 (G_3 + H_3) H_3 (C_2 + E_2 + F_2 + G_3 + H_3) (E_2 + F_2) E_2 =$$

$$(A_1 + G_3) B_1 D_2 H_3 E_2 = A_1 B_1 D_2 E_2 H_3 + B_1 D_2 E_2 G_3 H_3$$

Copertura minima: $B_1 D_2 E_2 G_3 H_3$

Espressione minima: $z = a'b'de + bcd' + abc' + c'e' + ae'$

I metodi di Petrick e Quine-McCluskey (espressione PS)

STEP 2: Costruzione della tabella di copertura

Evidenziazione sulla mappa di Karnaugh (istanziata 3 volte) dei raggruppamenti rettangolari primi:
 A_1 B_2 C_2 D_2 E_2 F_2 G_2 H_2
 I_2 L_2 M_2 N_2 O_2 P_2 Q_2 R_2

configurazioni (in base 10) delle celle per cui $z = 0$

RR primi di cui $z = 0$ -

	1	4	5	6	9	11	14	15	17	19	21	23	31
A_1		x		x									
B_2	x	x	x										
C_2	x								x				
D_2	x				x								
E_2	x				x	x							
F_2	x								x	x			
G_2	x		x						x		x		
H_2					x	x							
I_2				x			x						
L_2						x	x	x					
M_2							x						
N_2									x	x	x	x	
O_2							x	x					x
P_2											x	x	x
Q_2													x
R_2												x	x

		cde							
		000	001	011	010	100	101	111	110
ab	00	-	1	-	-	4	5	-	6
	01	-	9	11	-	-	-	15	14
	11	-	-	-	-	-	-	31	-
	10	-	17	19	-	-	21	23	-

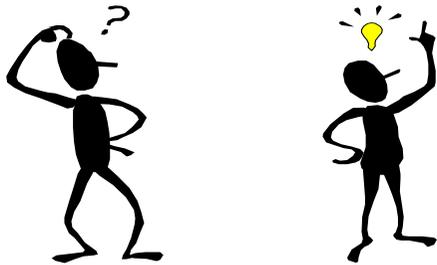
		1	4	5	6
		9	11	15	14
		17	19	21	23
		1	4	5	6
		9	11	15	14
		17	19	21	23

celle da coprire:	numero	sì
		no
	-	sì/no

Il metodo di Quine-McCluskey (espressione PS)

STEP 3: Soluzione per via grafica del problema della copertura

Nessuna riga essenziale



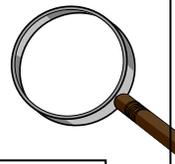
Rimozione delle
colonne dominanti
 $1 \supseteq 5, 14 \supseteq 15, 17 \supseteq 19$

La copertura della cella 15 impone un vincolo più restrittivo (L_2 o O_2) di quello derivante dalla copertura della cella 14 (L_2 o O_2 o I_2 o M_2). La cella 14 può essere trascurata in quanto sarà comunque coperta dal raggruppamento selezionato per la cella 15.

configurazioni (in base 10) delle celle per cui $z = 0$

	1	4	5	6	9	11	14	15	17	19	21	23	31
A_1		x		x									
B_2	x	x	x										
C_2	x								x				
D_2	x				x								
E_2	x				x	x							
F_2	x								x	x			
G_2	x		x						x		x		
H_2					x	x							
I_2				x			x						
L_2						x	x	x					
M_2							x						
N_2									x	x	x	x	
O_2								x	x				x
P_2											x	x	x
Q_2													x
R_2												x	x

RR primi di celle per cui $z = 0$



cde

	000	001	011	010	100	101	111	110
00	-	1	-		4	5		6
01	-	9	11	-			15	14
11				-	-	-	31	-
10	-	17	19			21	23	-

Il metodo di Quine-McCluskey (espressione PS)

1^a tabella di copertura ridotta

Rimozione delle righe ora inutili C_2 e M_2



		cde							
		000	001	011	010	100	101	111	110
ab	00	-	1	-		4	5		6
	01	-	9	11	-			15	14
	11				-	-	-	31	-
	10	-	17	19			21	23	-

celle residue

	4	5	6	9	11	15	19	21	23	31
A_1	x		x							
B_2	x	x								
C_2										
D_2				x						
E_2				x	x					
F_2							x			
G_2		x						x		
H_2				x	x					
I_2			x							
L_2					x	x				
M_2										
N_2							x	x	x	
O_2						x				x
P_2								x	x	x
Q_2										x
R_2									x	x

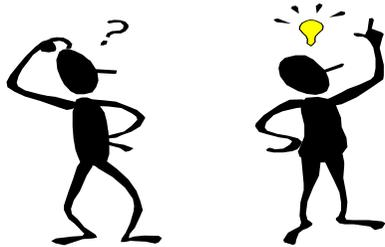
RR primi residui

numero

celle la cui copertura discende automaticamente dalla copertura di altre celle

Il metodo di Quine-McCluskey (espressione PS)

2ª tabella di copertura ridotta

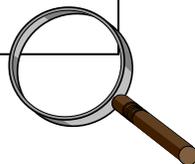


Rimozione delle **righe dominate**

$$D_2 \subseteq E_2, F_2 \subseteq N_2,$$

$$Q_2 \subseteq P_2, R_2 \subseteq P_2.$$

La riga **I₂**, pur essendo dominata dalla riga **A₁**, non deve essere rimossa perché l'ordine (2) del relativo raggruppamento è maggiore dell'ordine (1) del raggruppamento associato alla **riga dominante**.



celle residue

	4	5	6	9	11	15	19	21	23	31
A ₁	x		x							
B ₂	x	x								
D ₂				x						
E ₂				x	x					
F ₂							x			
G ₂		x						x		
H ₂				x	x					
I ₂			x							
L ₂					x	x				
N ₂							x	x	x	
O ₂						x				x
P ₂								x	x	x
Q ₂										x
R ₂									x	x

RR primi residui

cde

	000	001	011	010	100	101	111	110
00	-	1	-		4	5		6
01	-	9	11	-			15	14
11				-	-	-	31	-
10	-	17	19			21	23	-

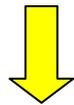
ab

Il metodo di Quine-McCluskey (espressione PS)

3^a tabella di copertura ridotta

Selezione della **riga ora essenziale** N_2 .
Rimozione di tale riga e delle colonne da essa coperte 19 21 23.

Rimozione della **colonna dominante**
 $11 \supseteq 9$.



4^a tabella di copertura ridotta

Rimozione delle **righe dominate**
 $G_2 \subseteq B_2$, $L_2 \subseteq O_2$, $P_2 \subseteq O_2$
($I_2 \subseteq A_1$, ma di ordine superiore).

celle residue

	4	5	6	9	11	15	19	21	23	31
A_1	x		x							
B_2	x	x								
E_2				x	x					
G_2		x						x		
H_2				x	x					
I_2			x							
L_2					x	x				
N_2							x	x	x	
O_2						x				x
P_2								x	x	x

RR primi residui

celle residue

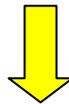
	4	5	6	9	15	31
A_1	x		x			
B_2	x	x				
E_2				x		
G_2		x				
H_2				x		
I_2			x			
L_2					x	
O_2					x	x
P_2						x

RR primi residui

Il metodo di Quine-McCluskey (espressione PS)

5^a tabella di copertura ridotta

Selezione delle **righe ora essenziali**
 B_2 (colonna 5) e O_2 (colonne 15 31).
 Rimozione di tali righe e delle colonne
 da esse coperte 4 5 15 31.



6^a tabella di copertura ridotta

Selezione della riga I_2
 per la copertura della colonna 6
 (A_1 è di ordine inferiore) e di E_2 o H_2
 per la copertura della colonna 9.

celle residue

	4	5	6	9	15	31
A_1	x		x			
B_2	x	x				
E_2				x		
H_2				x		
I_2			x			
O_2					x	x

RR primi residui

celle residue

	6	9
A_1	x	
E_2		x
H_2		x
I_2	x	

RR primi residui

Il metodo di Quine-McCluskey (espressione PS)

Coperture minime: $\{N_2 B_2 O_2 I_2 E_2\}$, $\{N_2 B_2 O_2 I_2 H_2\}$

		cde							
		000	001	011	010	100	101	111	110
ab	00	-	1	-	-	4	5	-	6
	01	-	9	11	-	-	-	15	14
	11	-	-	-	-	-	-	31	-
	10	-	17	19	-	-	21	23	-

		cde							
		000	001	011	010	100	101	111	110
ab	00	-	1	-	-	4	5	-	6
	01	-	9	11	-	-	-	15	14
	11	-	-	-	-	-	-	31	-
	10	-	17	19	-	-	21	23	-

Espressioni minime:

$$z = (a' + b + e') (a + b + d) (b' + c' + d') (c' + d' + e) (a + c + e')$$

$$z = (a' + b + e') (a + b + d) (b' + c' + d') (c' + d' + e) (a + b' + c)$$

Il metodo di Petrick (espressione PS)

STEP 3: Soluzione per via algebrica del problema della copertura

configurazioni (in base 10) delle celle per cui $z = 0$

	1	4	5	6	9	11	14	15	17	19	21	23	31
A₁		x		x									
B₂	x	x	x										
C₂	x								x				
D₂	x				x								
E₂	x				x	x							
F₂	x								x	x			
G₂	x		x						x		x		
H₂					x	x							
I₂				x			x						
L₂						x	x	x					
M₂							x						
N₂									x	x	x	x	
O₂							x	x					x
P₂											x	x	x
Q₂													x
R₂												x	x

RR primi di celle per cui $z = 0$ -

Coperture irridondanti:

$$\begin{aligned}
 & (B_2 + C_2 + D_2 + E_2 + F_2 + G_2) \quad (A_1 + B_2) \quad (B_2 + G_2) \quad (A_1 + I_2) \quad (D_2 + E_2 + H_2) \\
 & (E_2 + H_2 + L_2) \quad (I_2 + L_2 + M_2 + O_2) \quad (L_2 + O_2) \quad (C_2 + F_2 + G_2 + N_2) \quad (F_2 + N_2) \\
 & (G_2 + N_2 + P_2) \quad (N_2 + P_2 + R_2) \quad (O_2 + P_2 + Q_2 + R_2) =
 \end{aligned}$$

Il metodo di Petrick (espressione PS)

$$\begin{aligned}
 & (B_2+C_2+D_2+E_2+F_2+G_2) (A_1+B_2) (B_2+G_2) (A_1+I_2) (D_2+E_2+H_2) (E_2+H_2+L_2) (I_2+L_2+M_2+O_2) \\
 & (L_2+O_2) (C_2+F_2+G_2+N_2) (F_2+N_2) (G_2+N_2+P_2) (N_2+P_2+R_2) (O_2+P_2+Q_2+R_2) = \\
 & (A_1+B_2) (B_2+G_2) (A_1+I_2) (D_2+E_2+H_2) (E_2+H_2+L_2) (L_2+O_2) \\
 & (F_2+N_2) (G_2+N_2+P_2) (N_2+P_2+R_2) (O_2+P_2+Q_2+R_2) = \\
 & (A_1+B_2I_2) (B_2+G_2) (E_2+H_2+D_2L_2) (L_2+O_2) (F_2+N_2) (N_2+P_2+G_2R_2) (O_2+P_2+Q_2+R_2) =
 \end{aligned}$$

/* Coperture irridondanti che coinvolgono 7 raggruppamenti rettangolari primi */

$$\begin{aligned}
 & A_1B_2E_2L_2F_2G_2R_2 + A_1B_2E_2O_2F_2G_2R_2 + A_1B_2H_2L_2F_2G_2R_2 + A_1B_2H_2O_2F_2G_2R_2 + \\
 & A_1B_2D_2L_2F_2G_2R_2 + B_2I_2E_2L_2F_2G_2R_2 + B_2I_2E_2O_2F_2G_2R_2 + B_2I_2H_2L_2F_2G_2R_2 + \\
 & B_2I_2H_2O_2F_2G_2R_2 + B_2I_2D_2L_2F_2G_2R_2 +
 \end{aligned}$$

/* Coperture irridondanti che coinvolgono 6 raggruppamenti rettangolari primi */

$$\begin{aligned}
 & A_1B_2E_2L_2N_2P_2 + A_1B_2E_2L_2N_2R_2 + A_1B_2E_2L_2N_2Q_2 + A_1B_2E_2L_2F_2P_2 + A_1B_2E_2O_2F_2P_2 + \\
 & A_1B_2H_2L_2N_2P_2 + A_1B_2H_2L_2N_2R_2 + A_1B_2H_2L_2N_2Q_2 + A_1B_2H_2L_2F_2P_2 + A_1B_2H_2O_2F_2P_2 + \\
 & A_1B_2D_2L_2N_2P_2 + A_1B_2D_2L_2N_2R_2 + A_1B_2D_2L_2N_2O_2 + A_1B_2D_2L_2N_2Q_2 + A_1B_2D_2L_2F_2P_2 + \\
 & A_1G_2E_2L_2N_2P_2 + A_1G_2E_2L_2N_2R_2 + A_1G_2E_2L_2N_2Q_2 + A_1G_2E_2L_2F_2P_2 + A_1G_2E_2L_2F_2R_2 + \\
 & A_1G_2E_2O_2F_2P_2 + A_1G_2E_2O_2F_2R_2 + A_1G_2H_2L_2N_2P_2 + A_1G_2H_2L_2N_2R_2 + A_1G_2H_2L_2N_2Q_2 + \\
 & A_1G_2H_2L_2F_2P_2 + A_1G_2H_2L_2F_2R_2 + A_1G_2H_2O_2F_2P_2 + A_1G_2H_2O_2F_2R_2 + A_1G_2D_2L_2N_2P_2 + \\
 & A_1G_2D_2L_2N_2R_2 + A_1G_2D_2L_2N_2O_2 + A_1G_2D_2L_2N_2Q_2 + A_1G_2D_2L_2F_2P_2 + A_1G_2D_2L_2F_2R_2 + \\
 & B_2I_2E_2L_2N_2P_2 + B_2I_2E_2L_2N_2R_2 + B_2I_2E_2L_2N_2Q_2 + B_2I_2E_2L_2F_2P_2 + B_2I_2E_2O_2F_2P_2 + \\
 & B_2I_2H_2L_2N_2P_2 + B_2I_2H_2L_2N_2R_2 + B_2I_2H_2L_2N_2Q_2 + B_2I_2H_2L_2F_2P_2 + B_2I_2H_2O_2F_2P_2 + \\
 & B_2I_2D_2L_2N_2P_2 + B_2I_2D_2L_2N_2R_2 + B_2I_2D_2L_2N_2O_2 + B_2I_2D_2L_2N_2Q_2 + B_2I_2D_2L_2F_2P_2 +
 \end{aligned}$$

Il metodo di Petrick (espressione PS)

/* Coperture irridondanti che coinvolgono 5 raggruppamenti rettangolari primi */
 $A_1B_2E_2O_2N_2 + A_1B_2H_2O_2N_2 + A_1G_2E_2O_2N_2 + A_1G_2H_2O_2N_2 + B_2I_2E_2O_2N_2 + B_2I_2H_2O_2N_2$

La copertura minima va individuata nell'ambito di queste ultime 6 coperture irridondanti in quanto coinvolgono il numero minimo (5) di raggruppamenti rettangolari.
Le prime 4 coperture non sono minime in quanto prevedono, a differenza delle ultime 2, un raggruppamento rettangolare (A_1) di ordine inferiore (1).

Coperture minime: $\{N_2 B_2 O_2 I_2 E_2\}$, $\{N_2 B_2 O_2 I_2 H_2\}$

Espressioni minime:

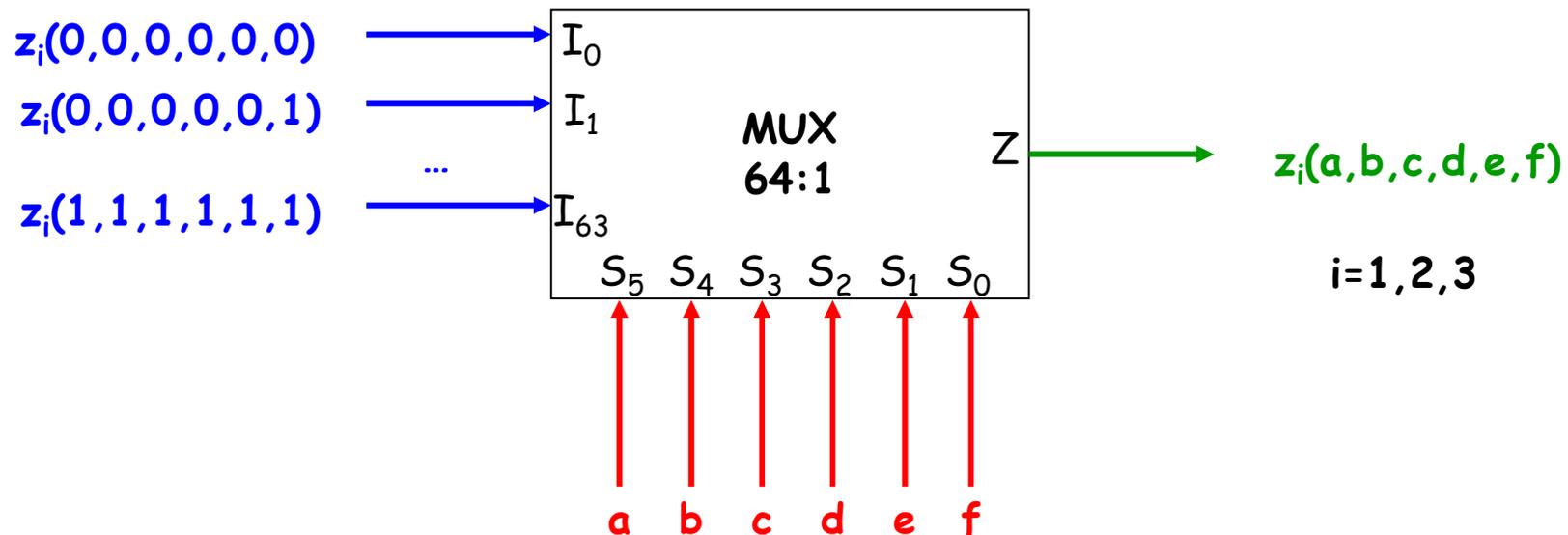
$$z = (a' + b + e') (a + b + d) (b' + c' + d') (c' + d' + e) (a + c + e')$$
$$z = (a' + b + e') (a + b + d) (b' + c' + d') (c' + d' + e) (a + b' + c)$$

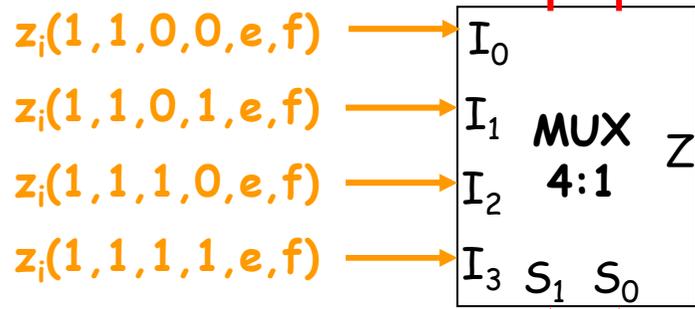
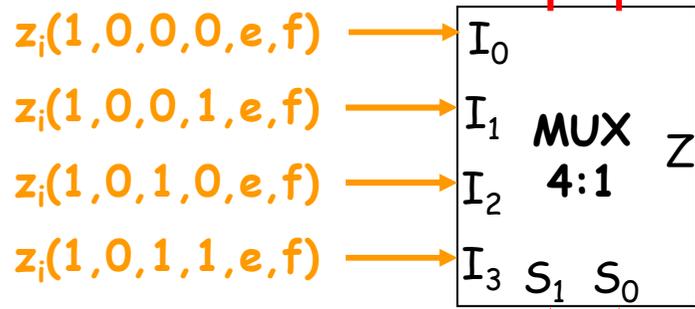
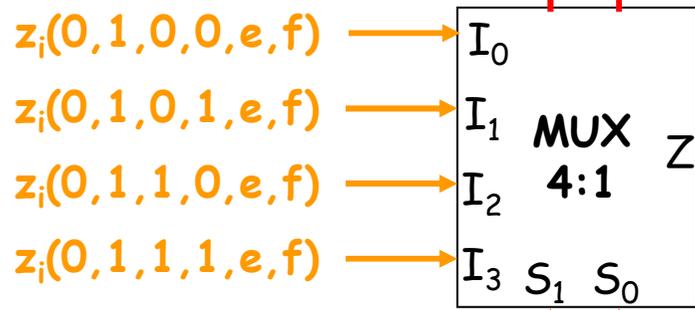
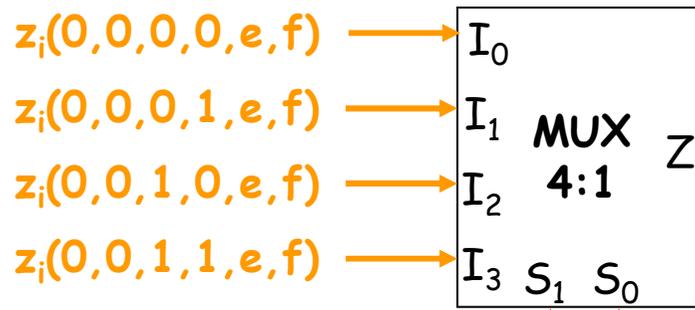
Esercizio 3

Si definisca la struttura di una rete programmabile, costituita da MUX con $k = 2$ bit di selezione e $2^k = 4$ bit di programmazione, che consenta di sintetizzare $m = 3$ funzioni combinatorie *qualsiasi* (z_1, z_2, z_3) di $n = 6$ variabili indipendenti (a, b, c, d, e, f).

L'impiego di MUX comporta la realizzazione indipendente di ciascuna funzione combinatoria.

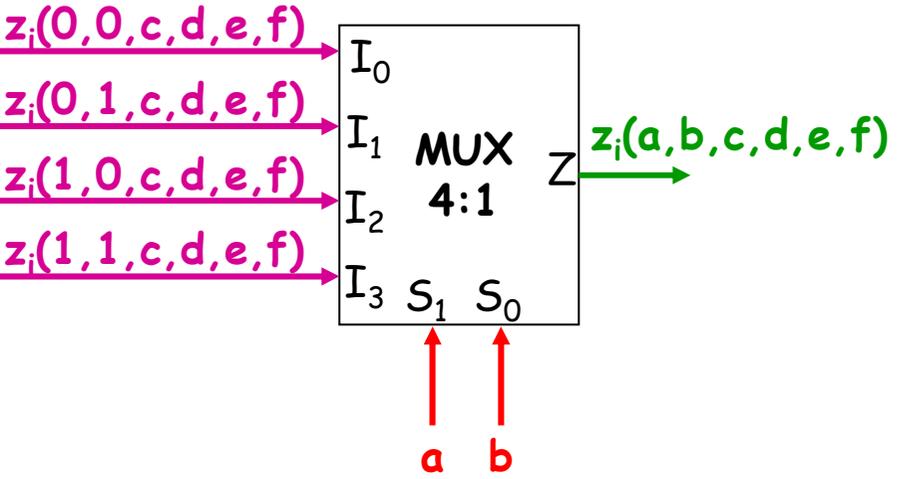
Ciascuna sottorete deve essere strutturata in maniera tale da dare luogo ad un comportamento equivalente a quello di un MUX con 6 bit di selezione e $2^6 = 64$ bit di programmazione.



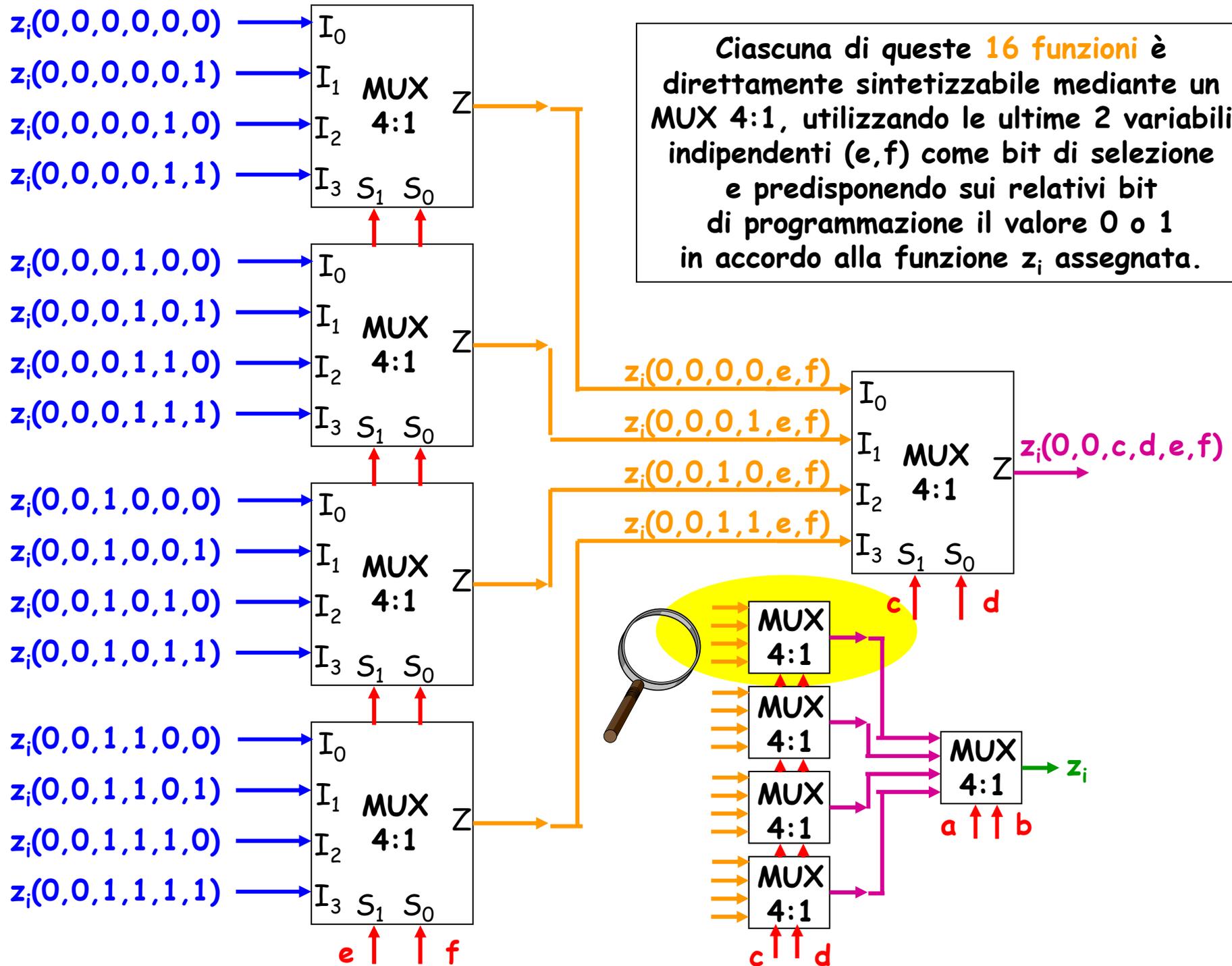


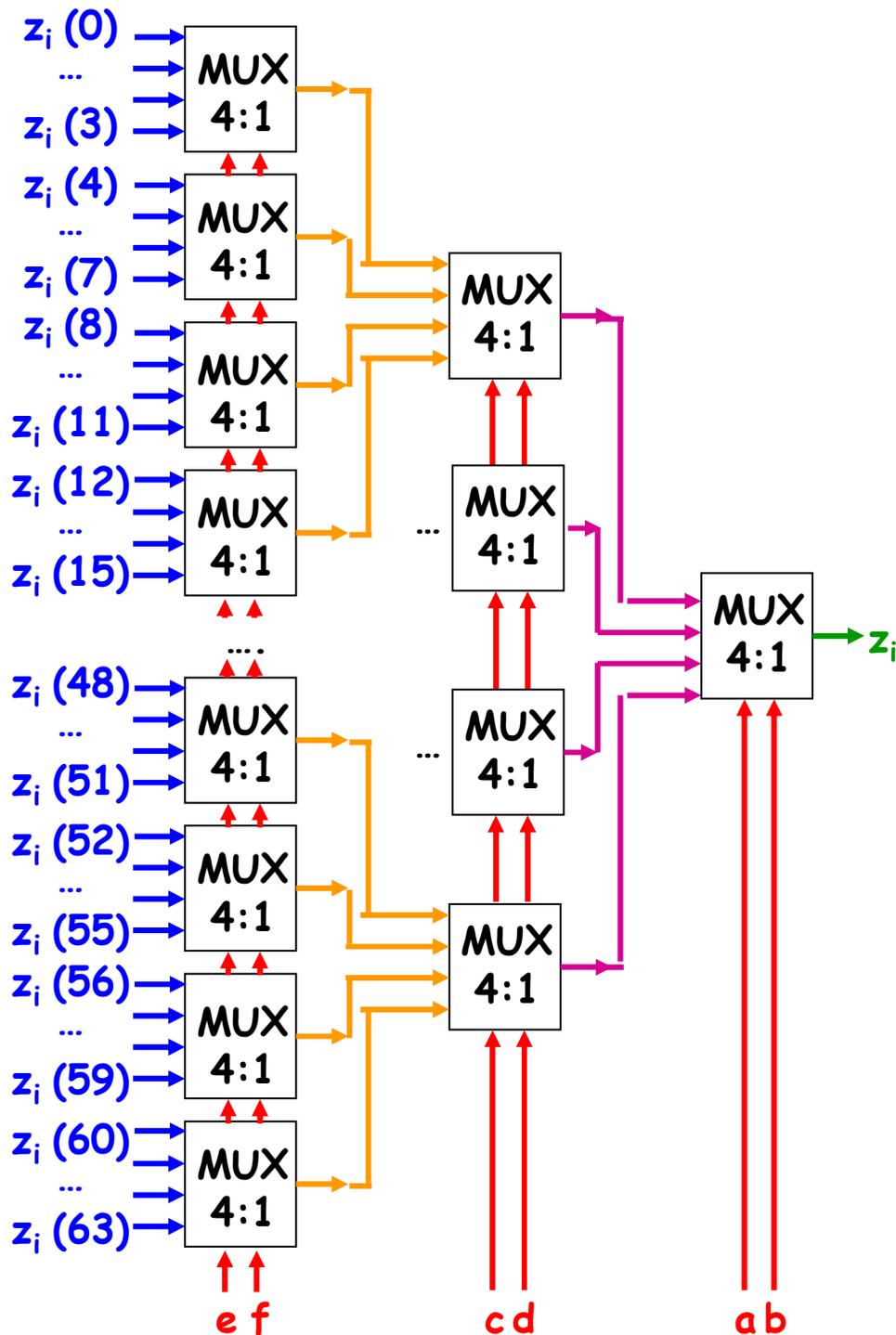
$c \uparrow \uparrow d$

Mediante un primo MUX 4:1, in cui 2 variabili indipendenti (ad es. a,b) intervengono come bit di selezione, è possibile ricondurre il problema della sintesi di **1 funzione combinatoria di 6 variabili** a quello della sintesi di **4 funzioni combinatorie di 4 variabili** (c,d,e,f).



Mediante quattro ulteriori MUX 4:1, in cui altre 2 variabili indipendenti (c,d) intervengono come bit di selezione, ci si riconduce al problema della sintesi di **16 funzioni combinatorie delle residue 2 variabili** (e,f).





Nel caso in esame ($n=6$, $m=3$, $k=2$)
occorrono quindi complessivamente
63 MUX, 21 per ogni funzione
combinatoria.

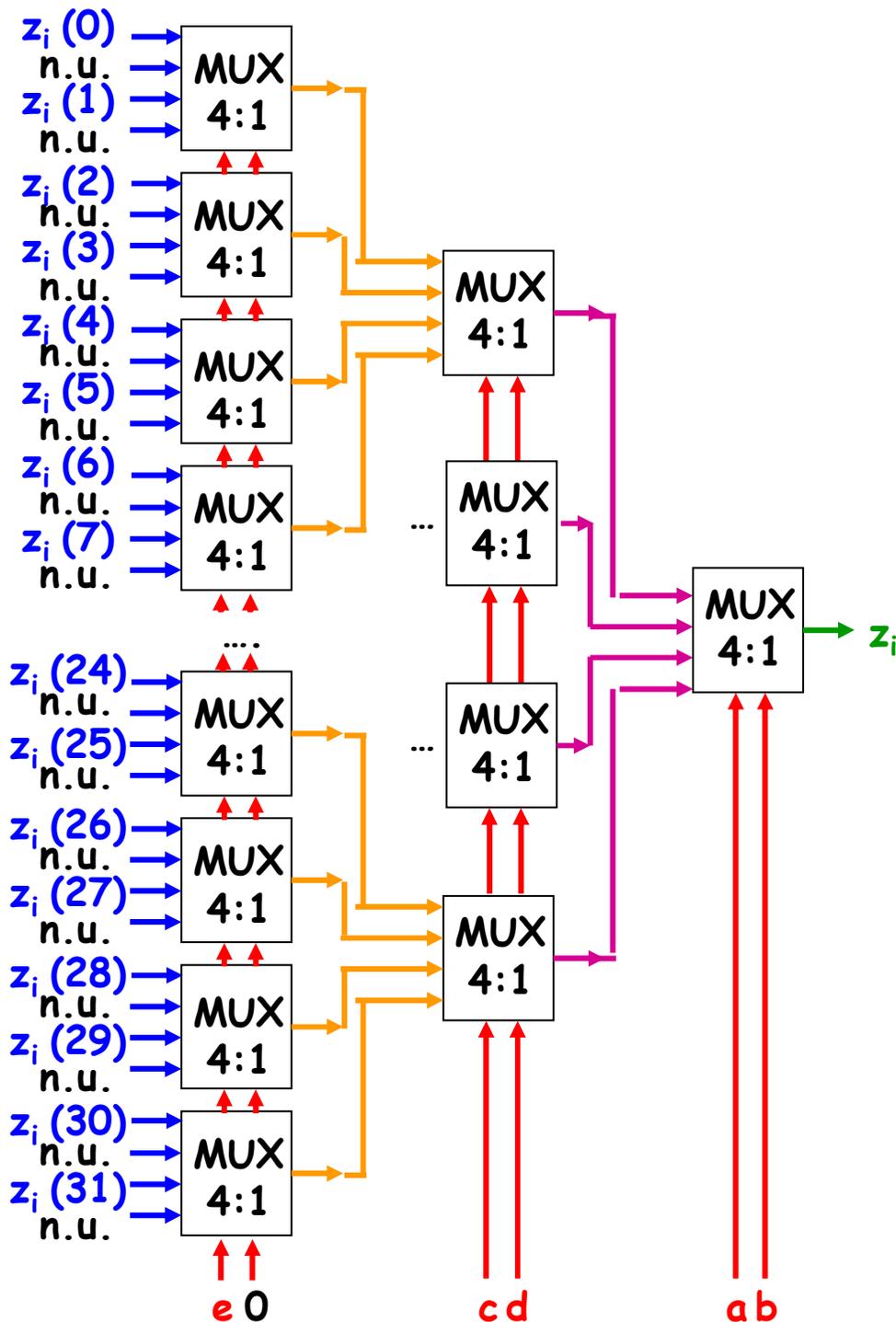
In generale, per sintetizzare
mediante MUX a k bit di selezione
una qualunque funzione combinatoria
di n variabili indipendenti,
con $n \bmod k = 0$, occorrono

$L=n/k$ livelli di elaborazione;

in ogni livello, a partire dal 1°,
intervengono rispettivamente
 $1, 2^k, 2^{2k}, \dots, 2^{(L-1)k}$ MUX;

il numero complessivo di MUX è

$$\sum_{l=1}^L 2^{(l-1)k} = \frac{2^n - 1}{2^k - 1}$$



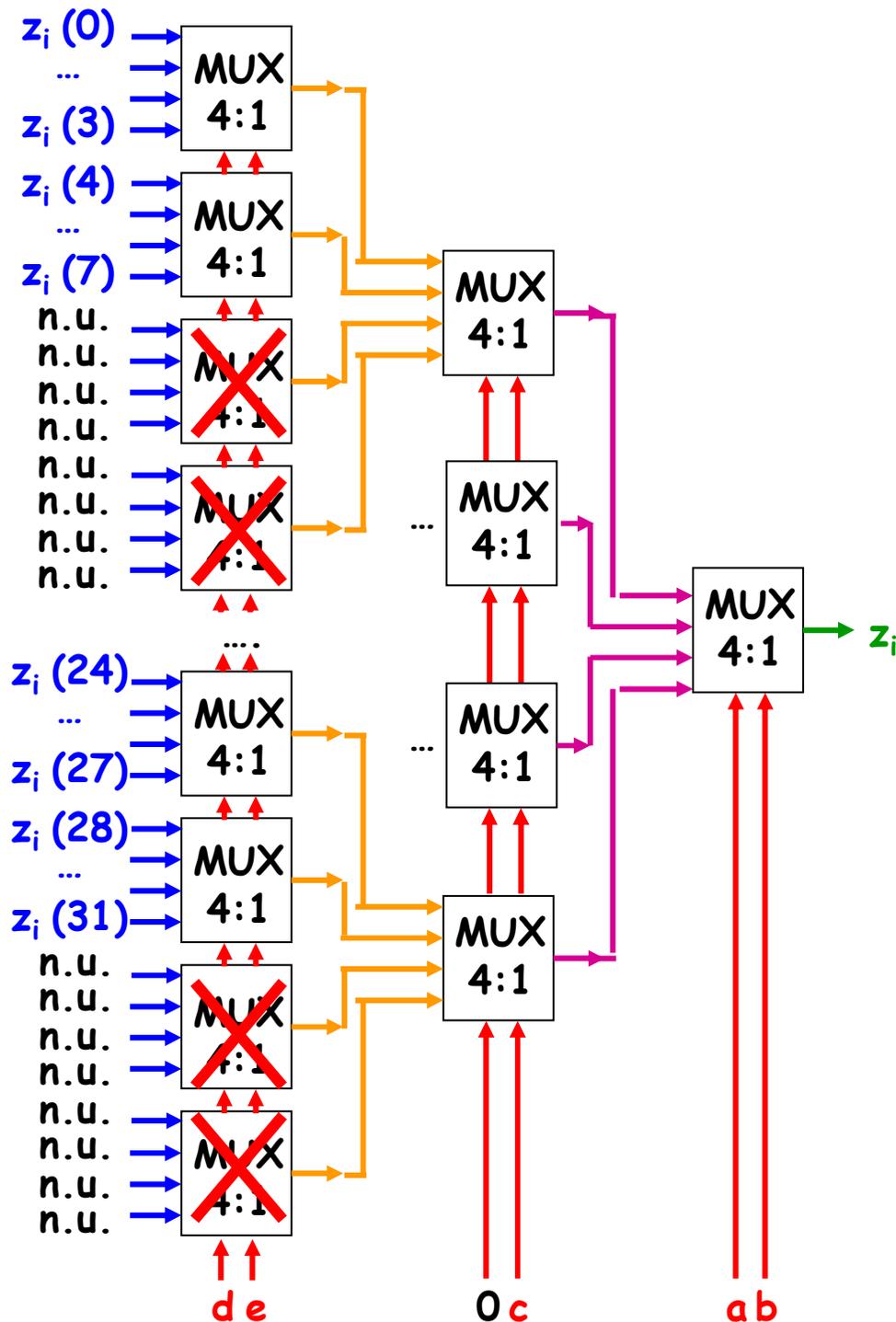
e se $(n \bmod k) > 0$???

Esempio: $n=5, k=2$
 $z_i(a,b,c,d,e)$

Una prima soluzione:
 2 variabili indipendenti (a,b) intervengono come bit di selezione al 1° livello, altre 2 (c,d) al 2° livello, l'ultima (e) al 3° livello.

Metà dei bit di programmazione dei MUX che operano al 3° livello non sono utilizzati (vanno collegati indifferentemente al livello logico 0 o 1). Più convenientemente si possono utilizzare al 3° livello 16 MUX 2:1.

Occorrono 21 (5 + 16) MUX.



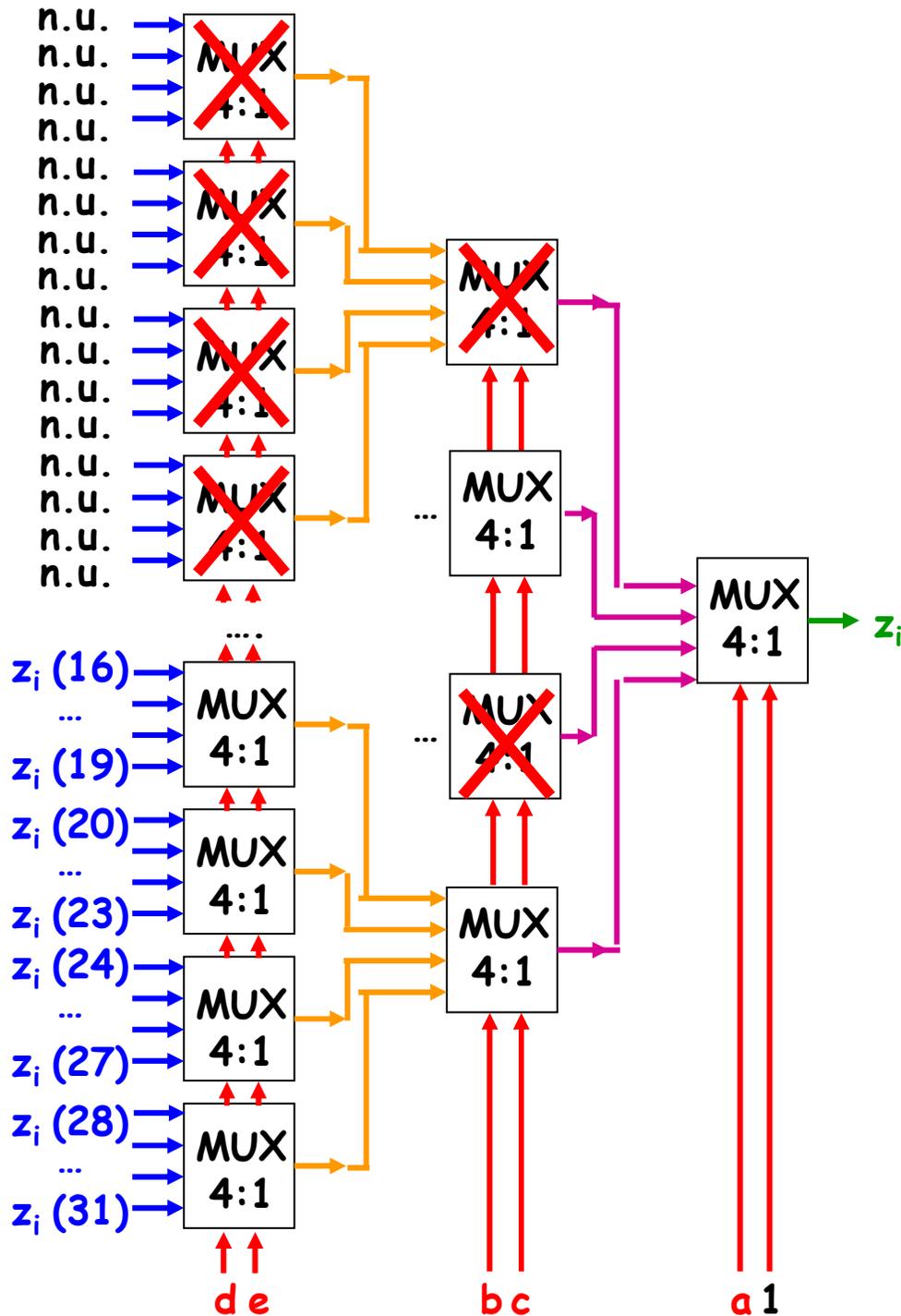
e se $(n \bmod k) > 0$???

Esempio: $n=5, k=2$
 $z_i(a, b, c, d, e)$

Una seconda soluzione:
 due variabili indipendenti (a,b) intervengono come bit di selezione al 1° livello, un'altra (c) al 2° livello, le ultime due (d,e) al 3° livello.

Metà dei MUX che operano al 3° livello non sono più necessari.
 Metà dei bit di programmazione dei MUX che operano al 2° livello non sono utilizzati (vanno collegati al livello logico 0 o 1).
 Più convenientemente si possono utilizzare al 2° livello 4 MUX 2:1.

Occorrono 13 (9 + 4) MUX.



e se $(n \bmod k) > 0$???

Esempio: $n=5, k=2$
 $z_i(a, b, c, d, e)$

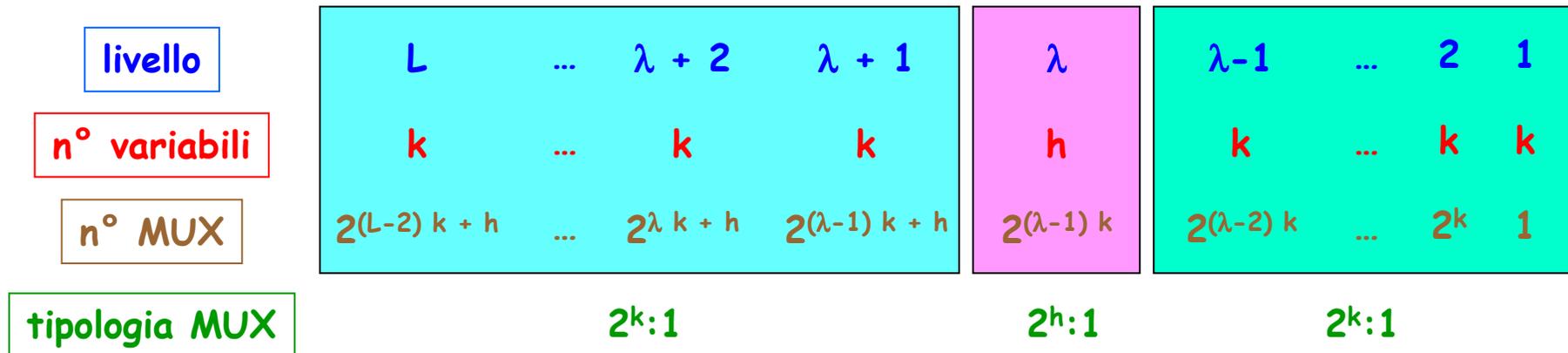
Una terza soluzione:
 1 sola variabile indipendente (a) interviene come bit di selezione al 1° livello, altre 2 (b,c) al 2° livello, le ultime 2 (d,e) al 3° livello.

Metà dei MUX che operano sia al 2° che, di conseguenza, al 3° livello non sono più necessari. Metà dei bit di programmazione del MUX che opera al 1° livello non sono utilizzati (vanno collegati al livello logico 0 o 1). Più convenientemente si può utilizzare 1 MUX 2:1.

Occorrono 11 (10 + 1) MUX.

In generale, per \forall coppia (n, k) , $n = i k + h$, $i = \lfloor n/k \rfloor > 0$, $h = n \bmod k > 0$,
 si hanno $L = i + 1$ livelli di elaborazione.

In ogni livello intervengono come bit di selezione k variabili indipendenti,
 ad eccezione di un livello λ ($\lambda = 1, 2, \dots, L$) in cui le variabili sono solo h .

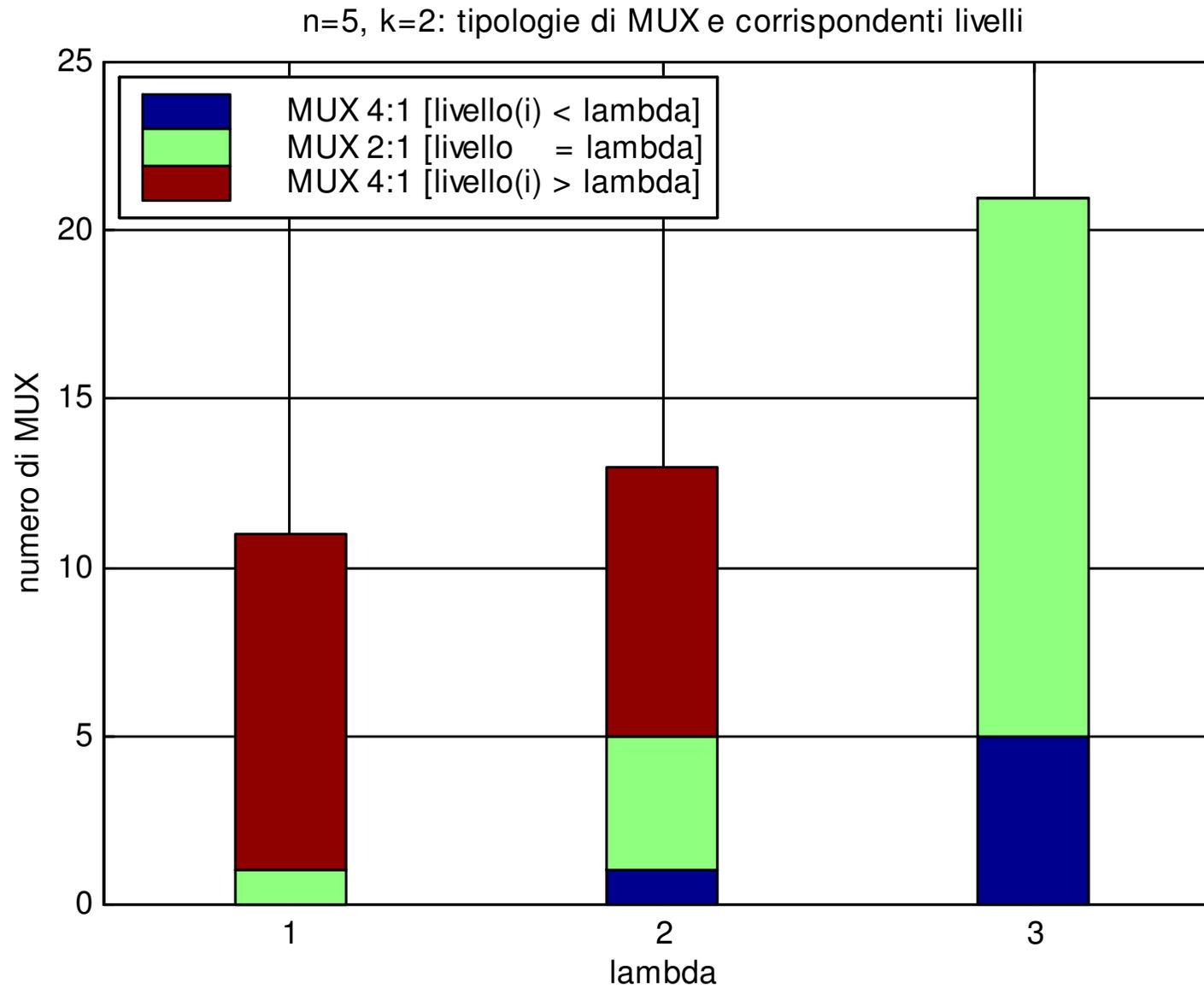


n° MUX $2^h:1$ $2^{(\lambda-1)k}$

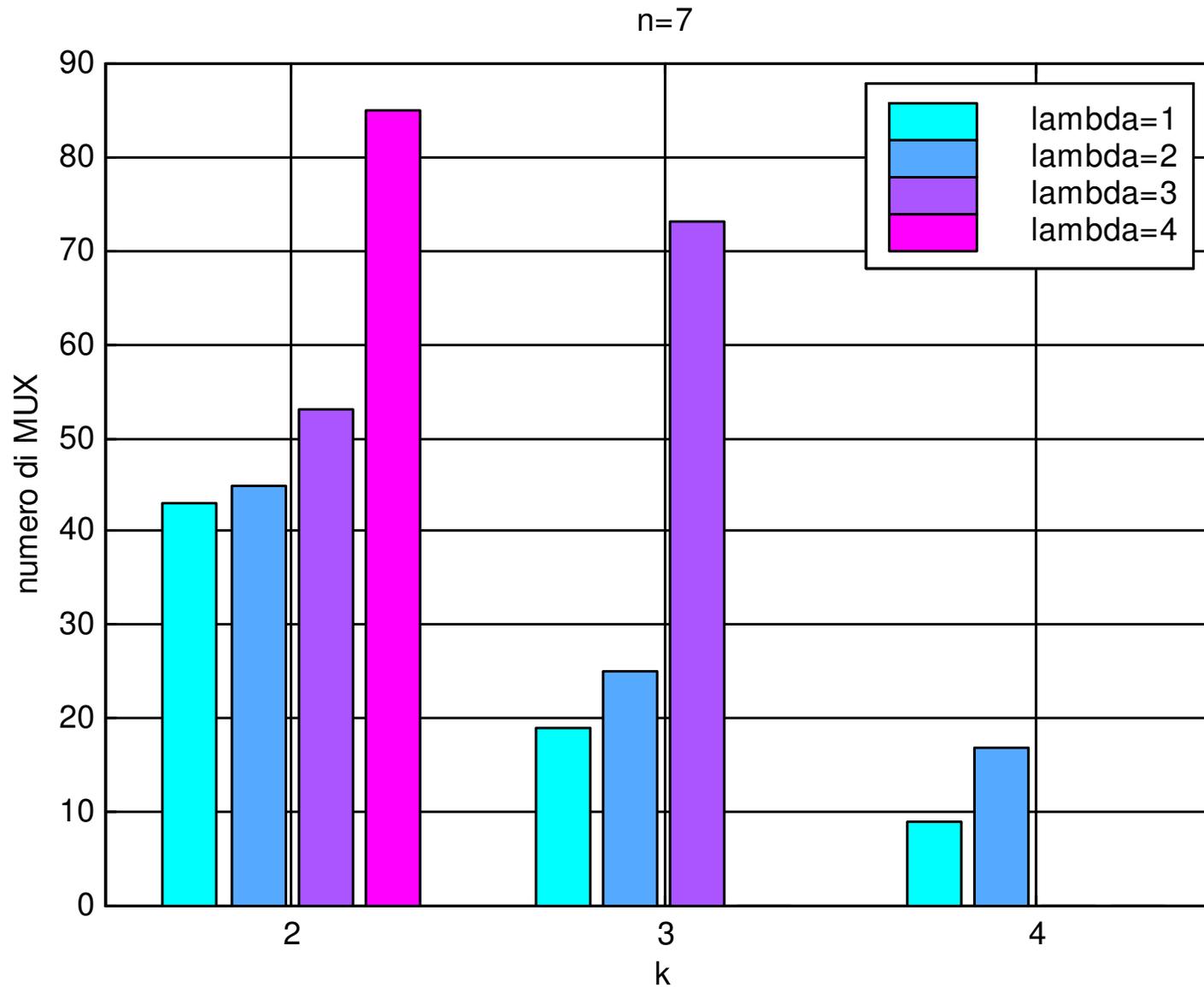
n° MUX $2^k:1$ $\frac{2^n - 2^{(\lambda-1)k+h}}{2^k - 1} + \frac{2^{(\lambda-1)k} - 1}{2^k - 1} = \frac{2^n - 1 - 2^{(\lambda-1)k}(2^h - 1)}{2^k - 1}$

n° totale MUX $N_{MUX}(n, k, \lambda) = \frac{2^n - 1 + 2^{(\lambda-1)k}(2^k - 2^h)}{2^k - 1}$

Al decrescere di λ aumenta il numero di MUX $2^k:1$, ma contemporaneamente diminuisce, ed in maniera più rilevante, il numero di MUX $2^h:1$.



Al fine di ottenere la soluzione che coinvolge il minor numero di MUX conviene pertanto scegliere $\lambda=1$, ovvero utilizzare un MUX del tipo $2^h:1$ al 1° livello e MUX del tipo $2^k:1$ nei successivi livelli:



Il numero complessivo di MUX che deriva dalla scelta $\lambda=1$ è:

$$n_{\text{MUX}}(n,k) = N_{\text{MUX}}(n,k,1) = 1 + \frac{2^n - 2^h}{2^k - 1} \quad (h = n \bmod k > 0)$$

tipologia MUX

$2^h:1$

$2^k:1$

livello

1

2 ... $\lceil n/k \rceil$

In particolare per $h=0$ il numero di MUX è (come già visto):

$$n_{\text{MUX}}(n,k) = \frac{2^n - 1}{2^k - 1} \quad (h = n \bmod k = 0)$$

tipologia MUX

$2^k:1$

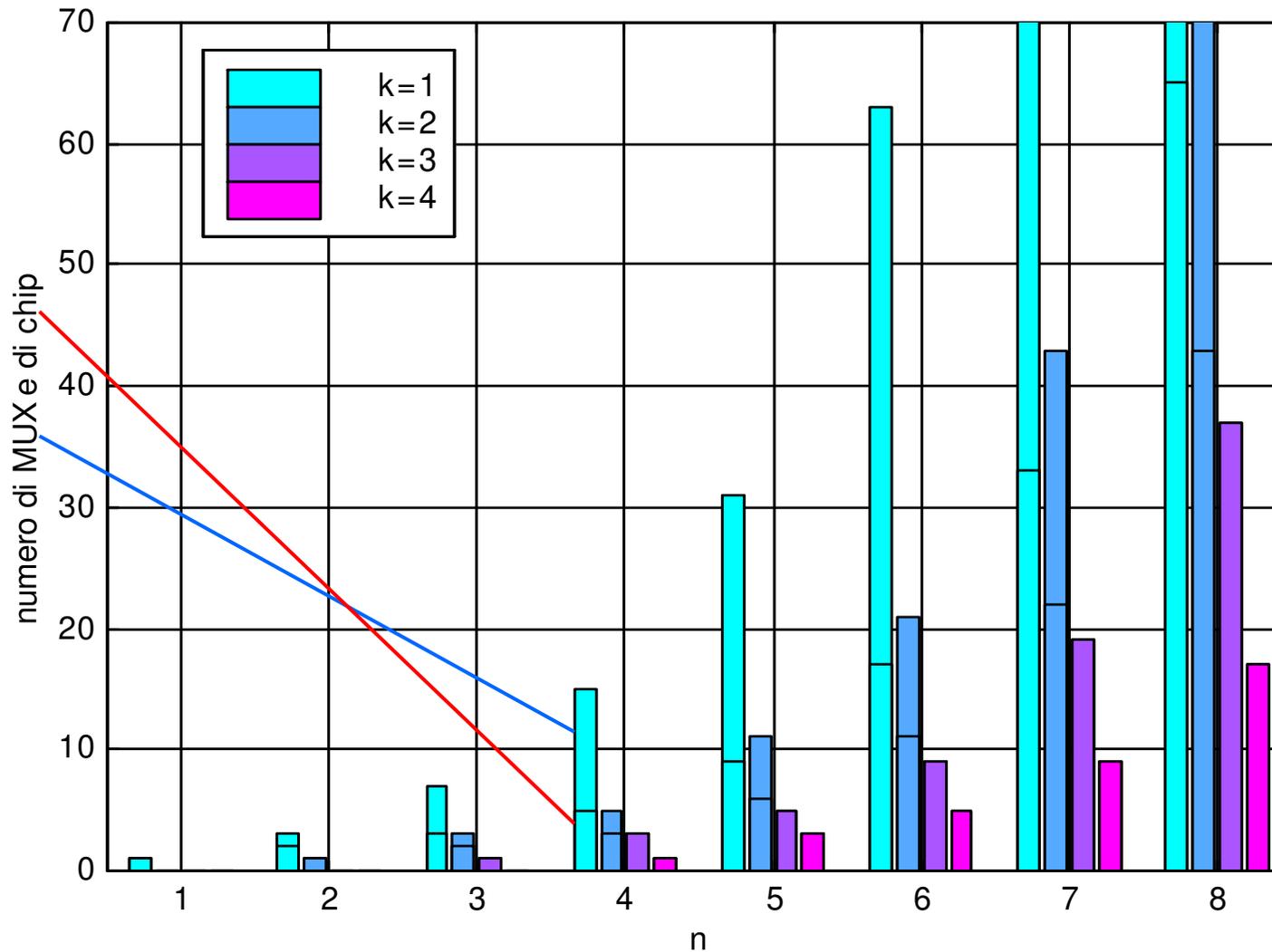
livello

1 2 ... n/k

Dal momento che $n_{\text{MUX}}(n, k)$ diminuisce al crescere di k , conviene utilizzare MUX caratterizzati dal valore massimo di k :
 $\min \{n_{\text{MUX}}(n, k)\} = n_{\text{MUX}}(n, k_{\text{max}})$.

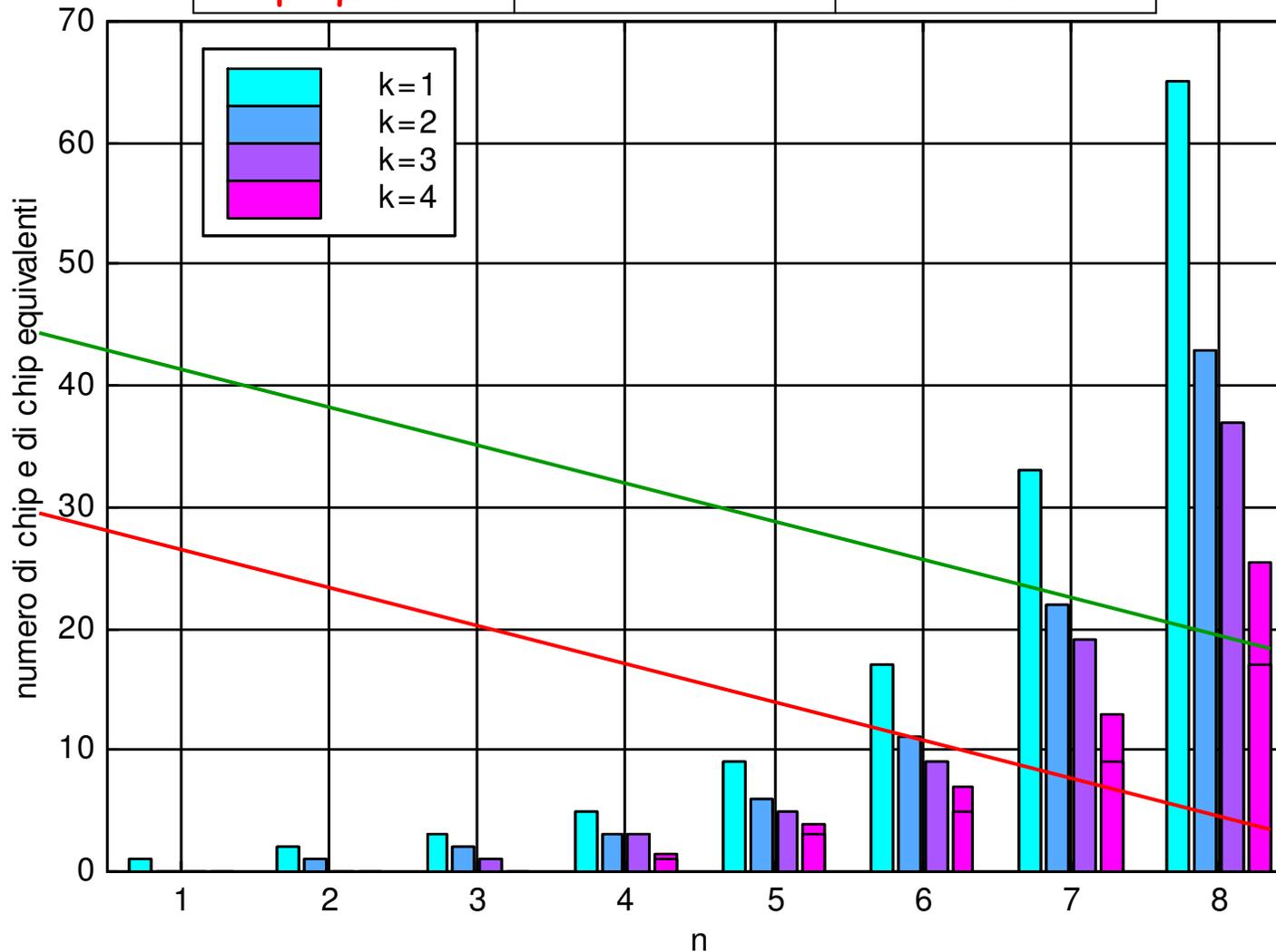
La scelta $k=k_{\max}$ (ovvero $k=4$) risulta preferibile dal punto di vista del numero di chip necessari per la realizzazione della rete, anche se, all'aumentare di k , diminuisce il n° di MUX/chip ...

chip	SN74157	SN74153	SN74151	SN74150
n° e tipologia di MUX	4 x 2:1	2 x 4:1	1 x 8:1	1 x 16:1



... ed aumentano le dimensioni, il n° di pin ed il consumo dei chip.

chip	SN74157/3/1	SN74150
dimensioni [pollici]	0.785 × 0.31	1.29 × 0.615
n° di pin	16	24
consumo [mW]	145	200
chip equivalenti	1	1.5



La scelta $k=k_{\max}$ consente infine di minimizzare anche il numero di livelli di elaborazione e, conseguentemente, il tempo di risposta della rete complessiva.

