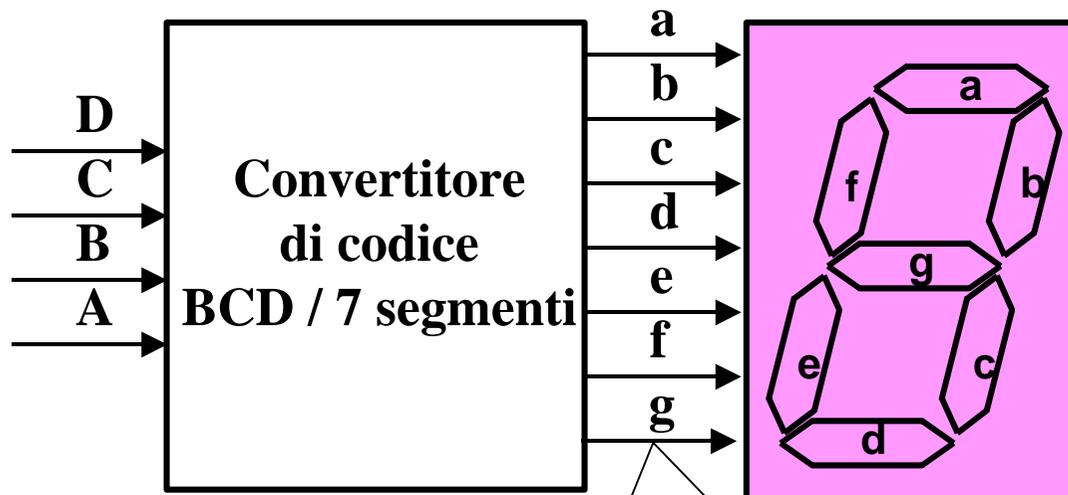


A bright yellow starburst shape with a black outline, centered on the page. The starburst has multiple sharp points of varying lengths, creating a jagged, sunburst-like appearance.

**Reti
combinatorie**

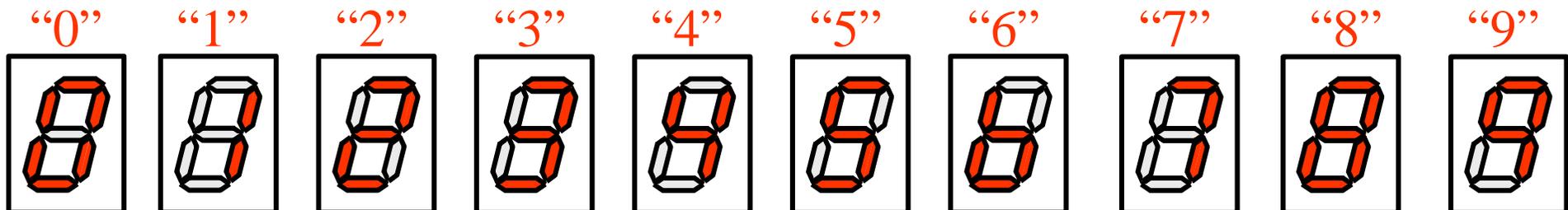
Esercizio 1

Si esegua il progetto di un convertitore di codice BCD / 7 segmenti, secondo le seguenti specifiche:



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

$$\mathbf{a = D'C'B'A + CA'}$$

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

b

$$\mathbf{b = CB'A + CBA'}$$

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

c

$$\mathbf{c = C'BA'}$$

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

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

d

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

e

$$\mathbf{d = CB'A' + C'B'A + CBA}$$

$$\mathbf{e = A + CB'}$$

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

f

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

g

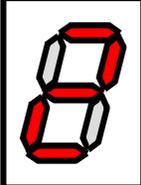
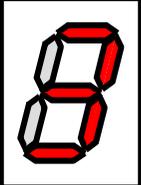
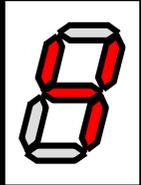
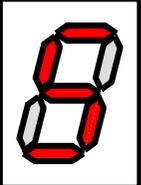
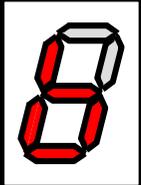
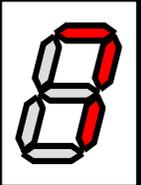
$$\mathbf{f = D'C'A + BA + C'B}$$

$$\mathbf{g = D'C'B' + CBA}$$

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

a

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

$$a_1 = a + DC$$

$$a_2 = a + DB$$

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

b

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

$$b_1 = b + DC$$

$$b_2 = b + DB$$

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

c

$$c = C'BA'$$

$$c_1 = c + DC$$

$$c_2 = c + DB$$

... 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).

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

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

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

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

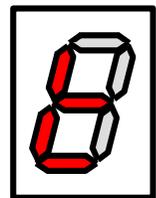
$$R_3 : a_1 b_2 c_1 d e f g$$

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

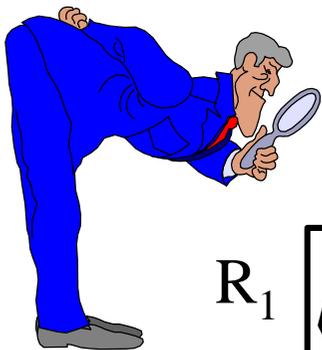
$$R_4 : a_1 b_2 c_2 d e f g$$

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

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 soluzioni R_1 , R_2 , R_4 , R_6 e R_8 vanno scartate, in quanto non consentono la rilevazione di tutte le configurazioni di ingresso illecite.



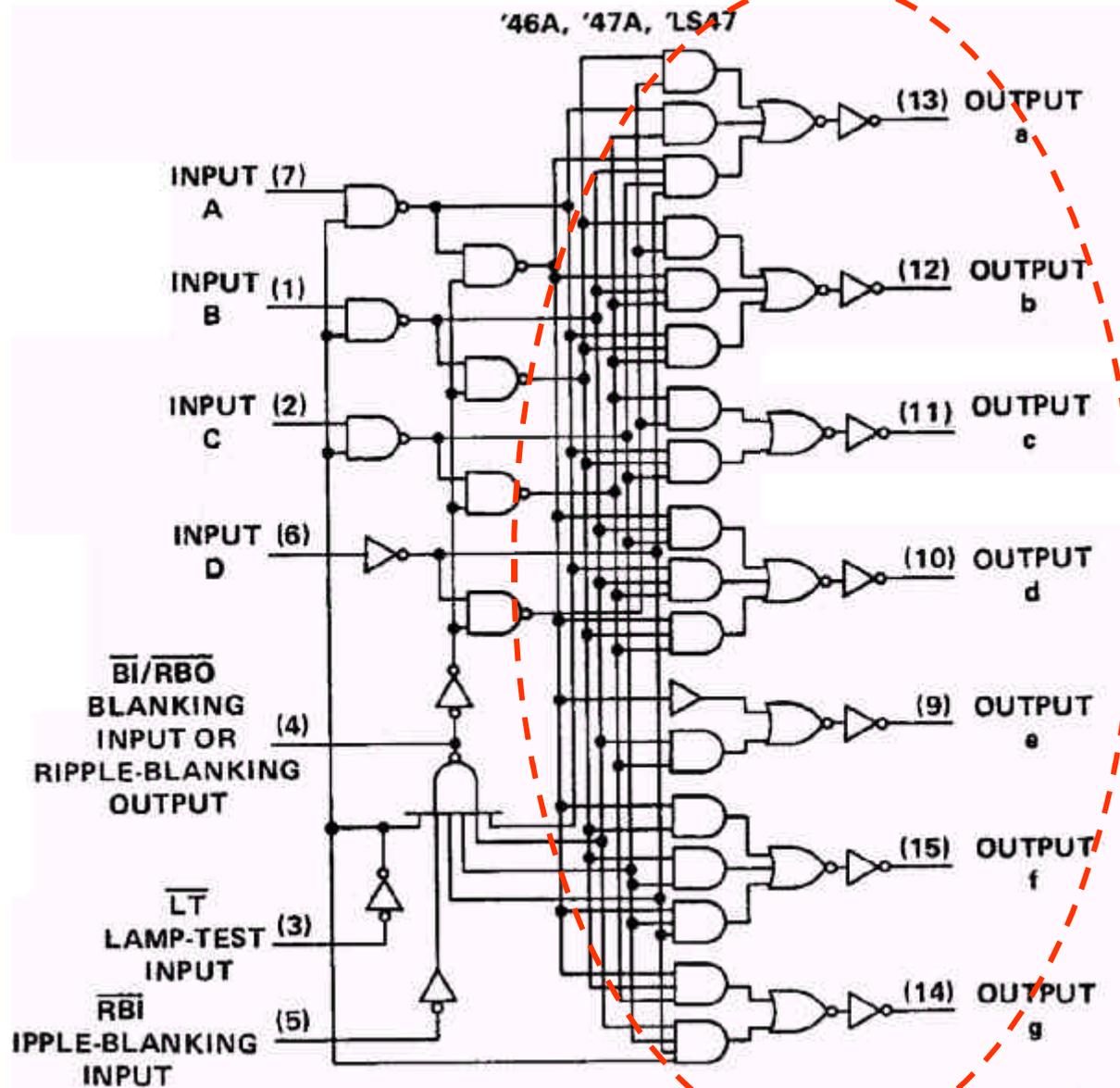
Le soluzioni R_3 , R_5 e R_7 vanno bene, in quanto consentono la rilevazione di tutte le configurazioni di ingresso illecite, peraltro con simboli tutti diversi fra loro.



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 globalmente accesi, anziché 18) per visualizzare le configurazioni illecite. Questa è la soluzione adottata nei circuiti integrati SN 7446, 7447, 7448, 7449 !!!

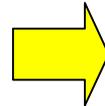
La soluzione integrata ...



... La soluzione integrata ...

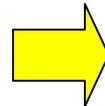
I gate aggiuntivi previsti nella soluzione integrata servono per conseguire ulteriori funzionalità, derivabili da specifici segnali di ingresso-uscita (tutti attivi a livello logico 0), ed elencate in ordine di priorità decrescente:

BI' (Blanking Input)



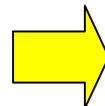
display spento

LT' (Lamp Test)



display acceso

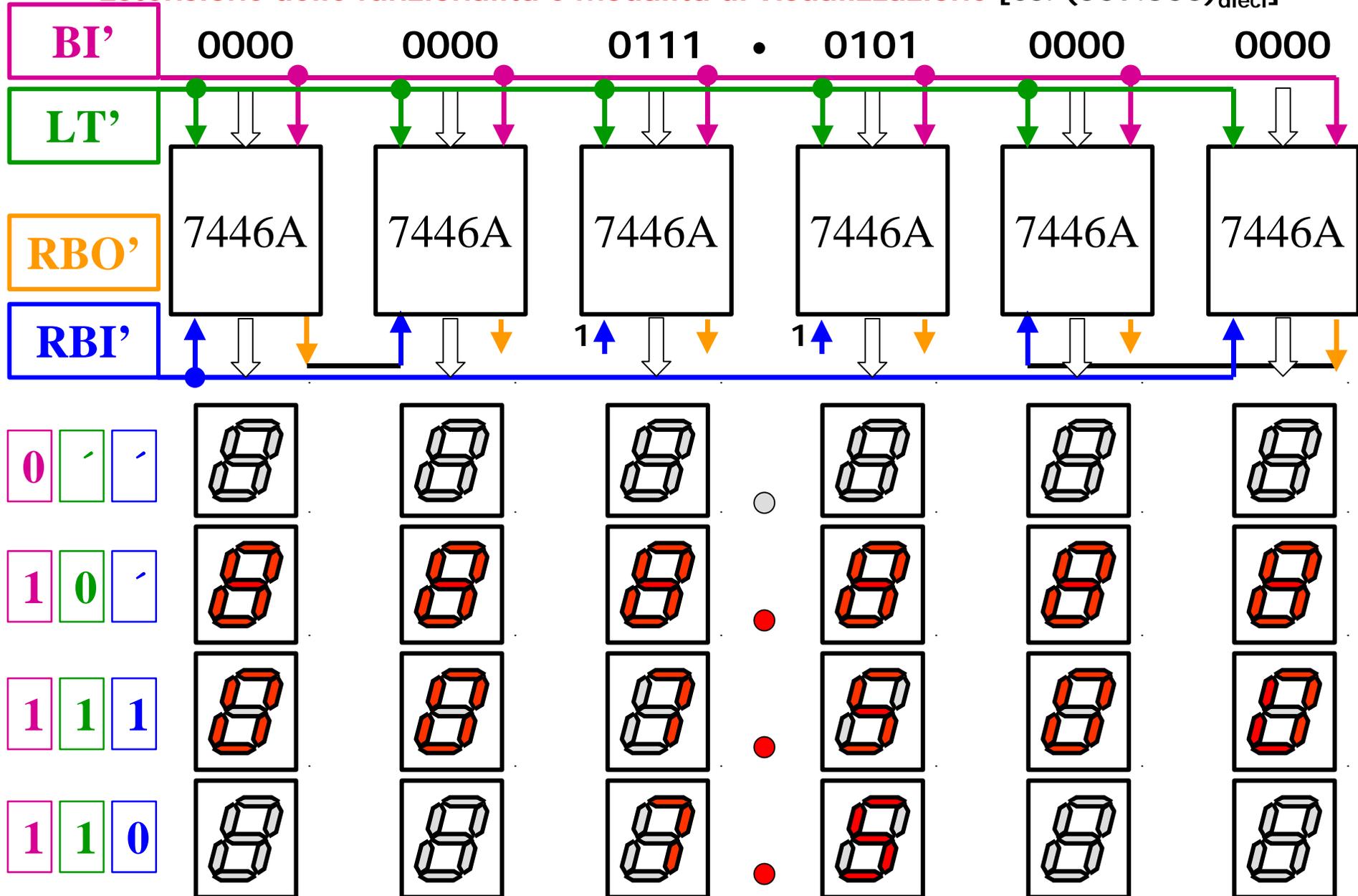
RBI' (Ripple Blanking Input)



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

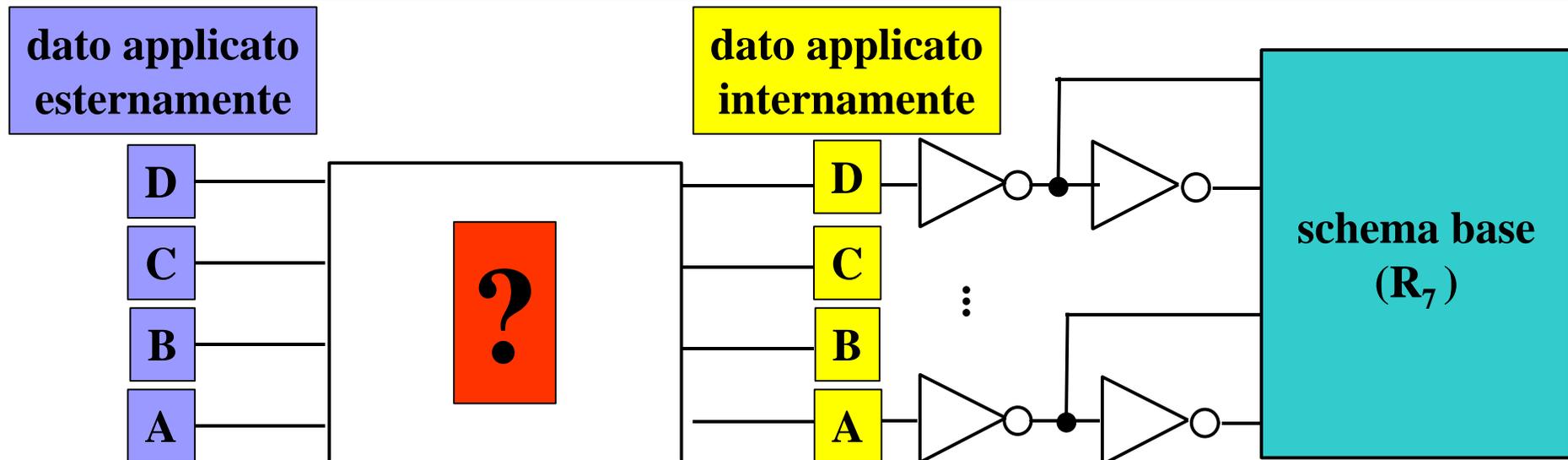
... La soluzione integrata ...

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



... La soluzione integrata ...

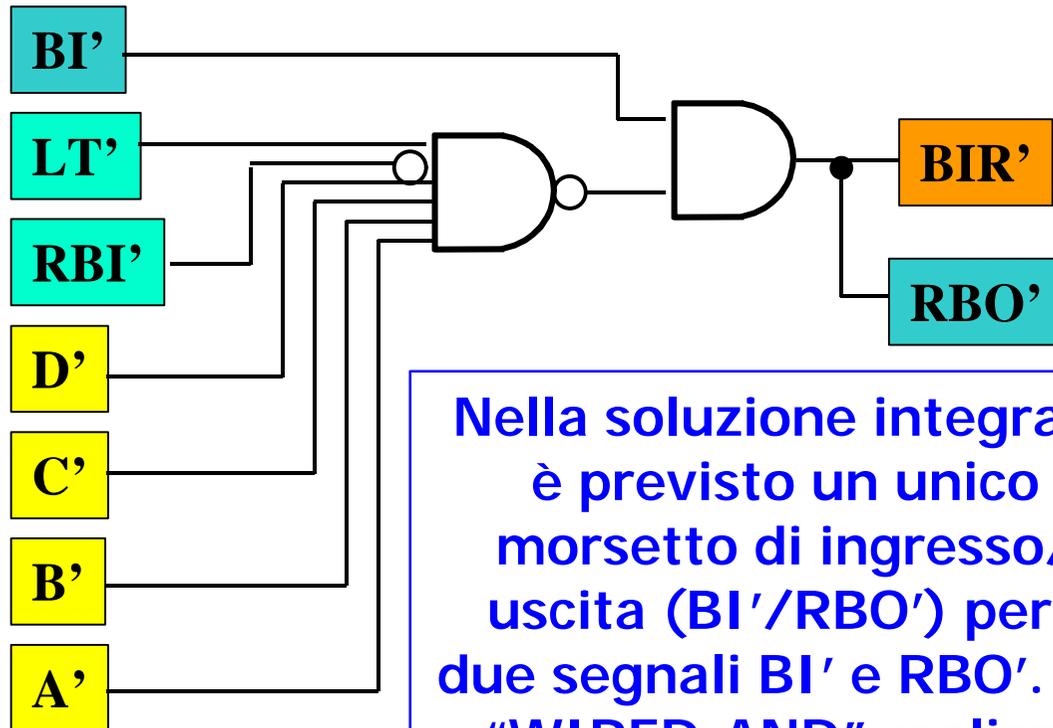
Estensione dello schema base per la gestione dei segnali ausiliari



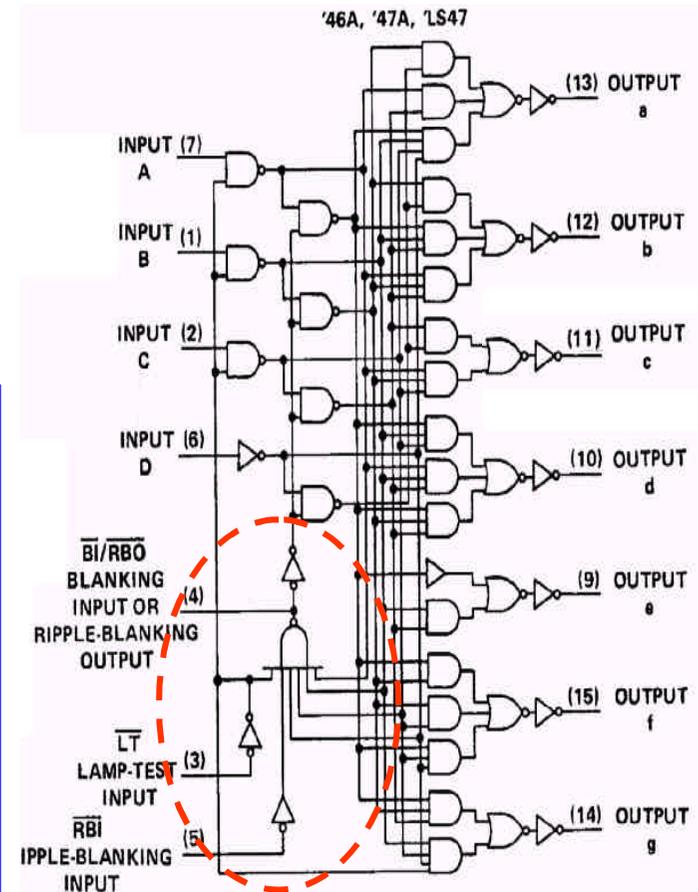
BI'	LT'	RBI'	RBO'	D	C	B	A	
1	1	1	1	D	C	B	A	
0	'	'	0	1	1	1	1	display spento
1	0	'	1	1	0	0	0	display acceso
1	1	0	0	1	1	1	1	se DCBA = 0000
1	1	0	1	D	C	B	A	se DCBA ¹ 0000

... 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 $RBO' = 0$.

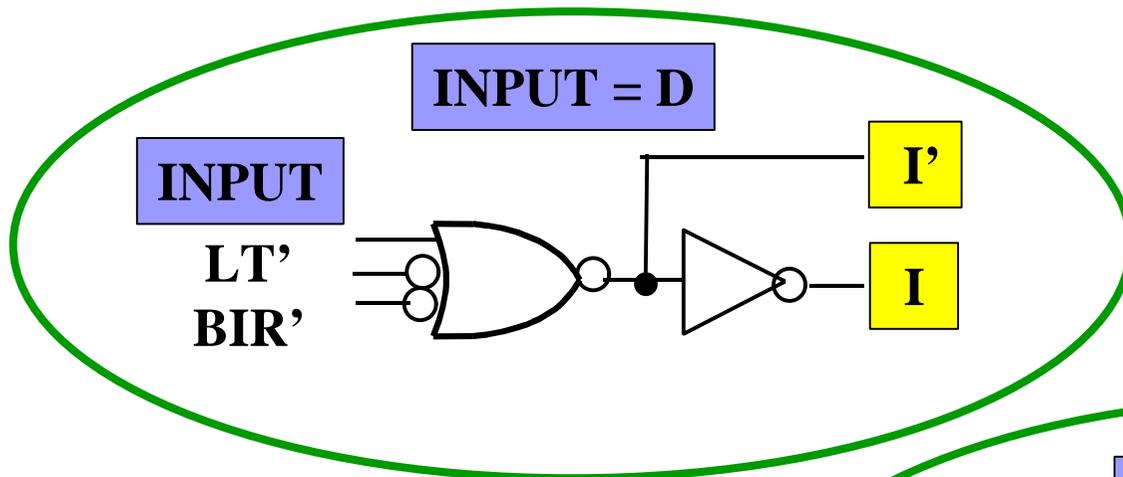


Nella soluzione integrata è previsto un unico morsetto di ingresso/uscita (BI'/RBO') per i due segnali BI' e RBO' . Un "WIRED-AND" realizza l'operatore di prodotto logico indicato in figura.

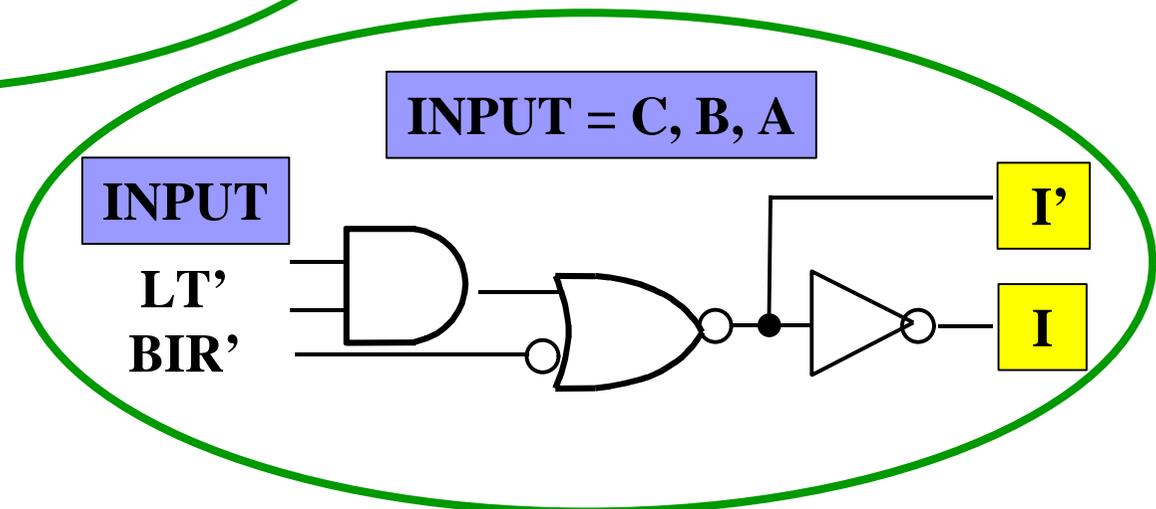


... La soluzione integrata ...

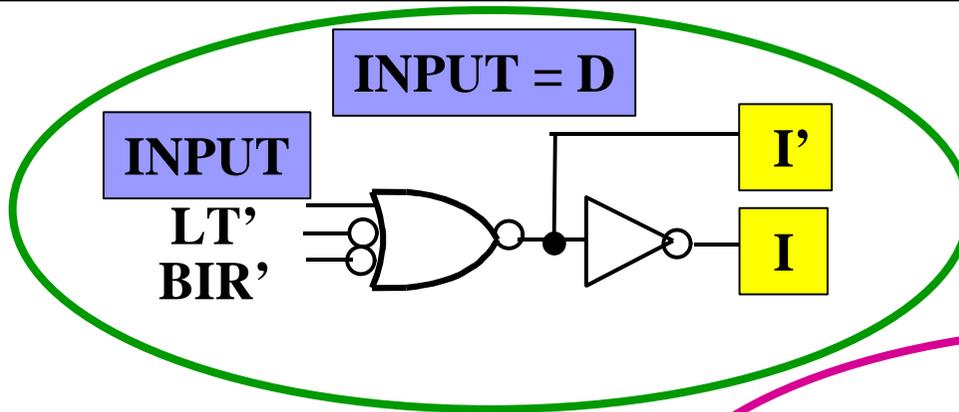
la gestione dei segnali BIR' e LT'



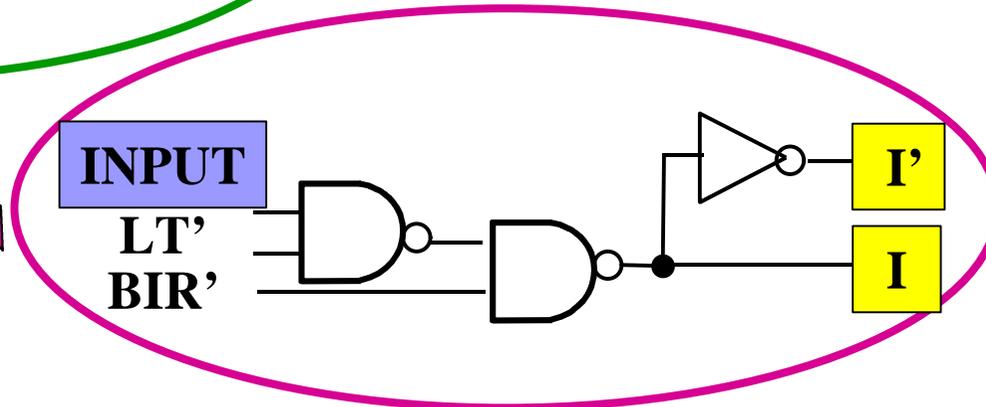
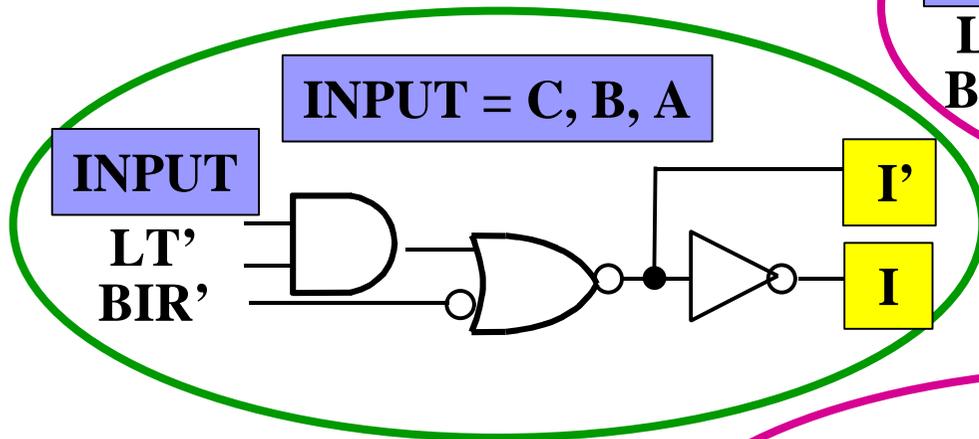
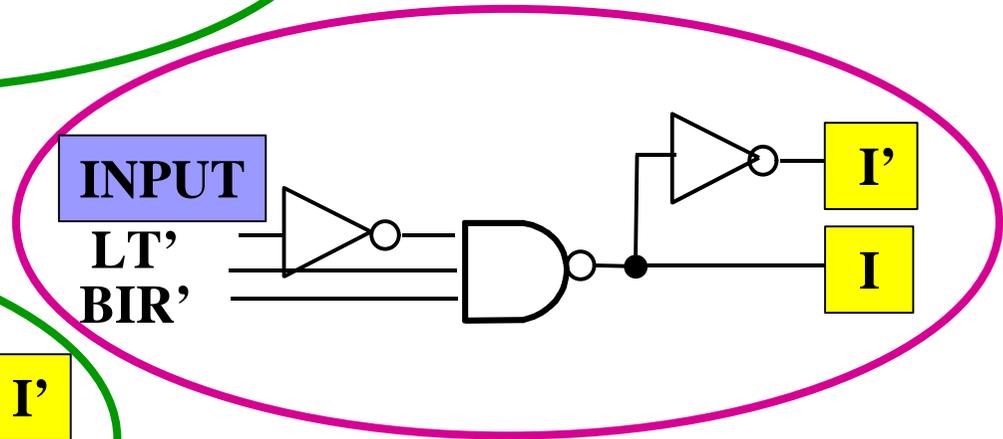
BIR'	LT'	D	C	B	A
1	1	D	C	B	A
0	1	1	1	1	1
1	0	1	0	0	0



... La soluzione integrata ...

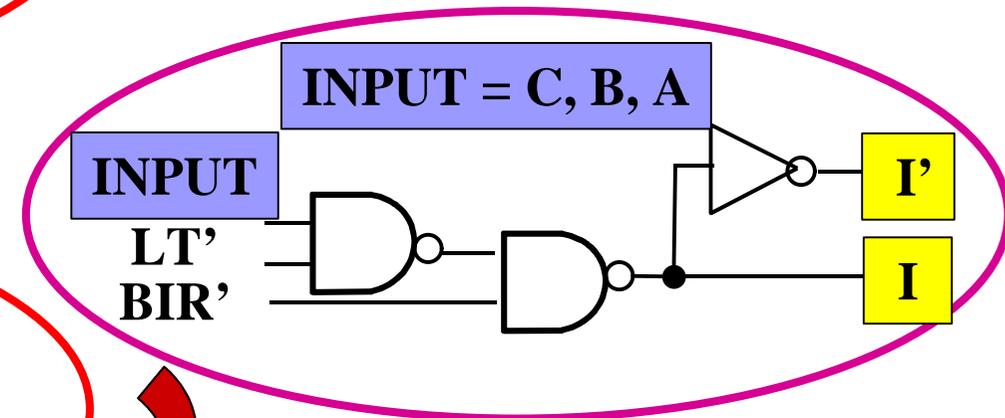
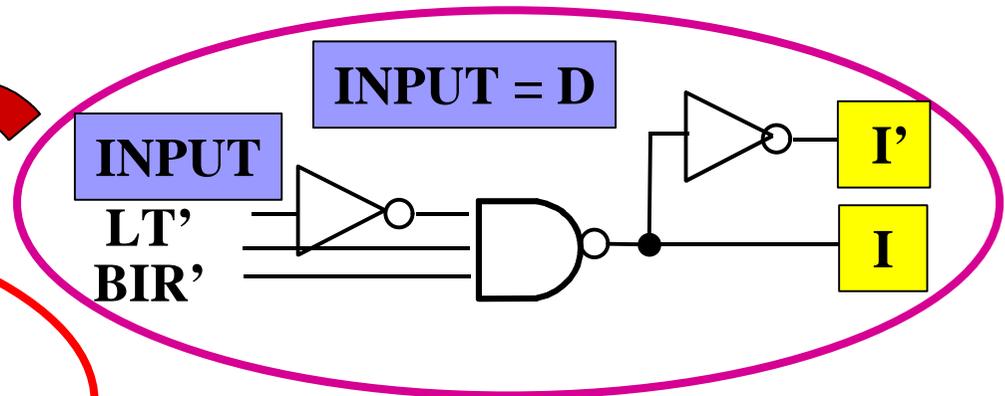
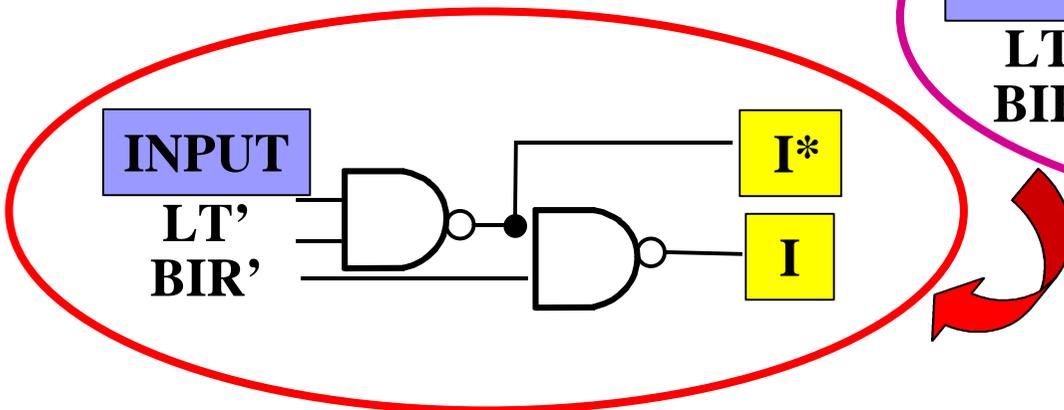
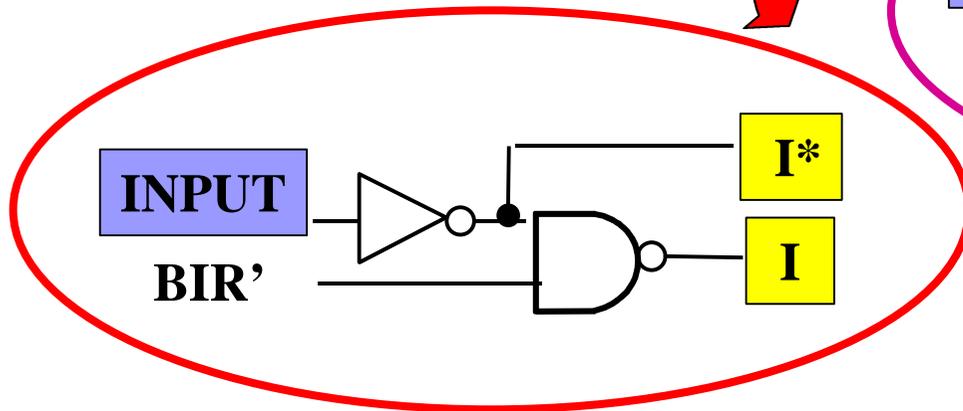


legge di De Morgan



... La soluzione integrata ...

la soluzione integrata



... La soluzione integrata

Solo se $BIR' = 1$

$I^* = I'$

?

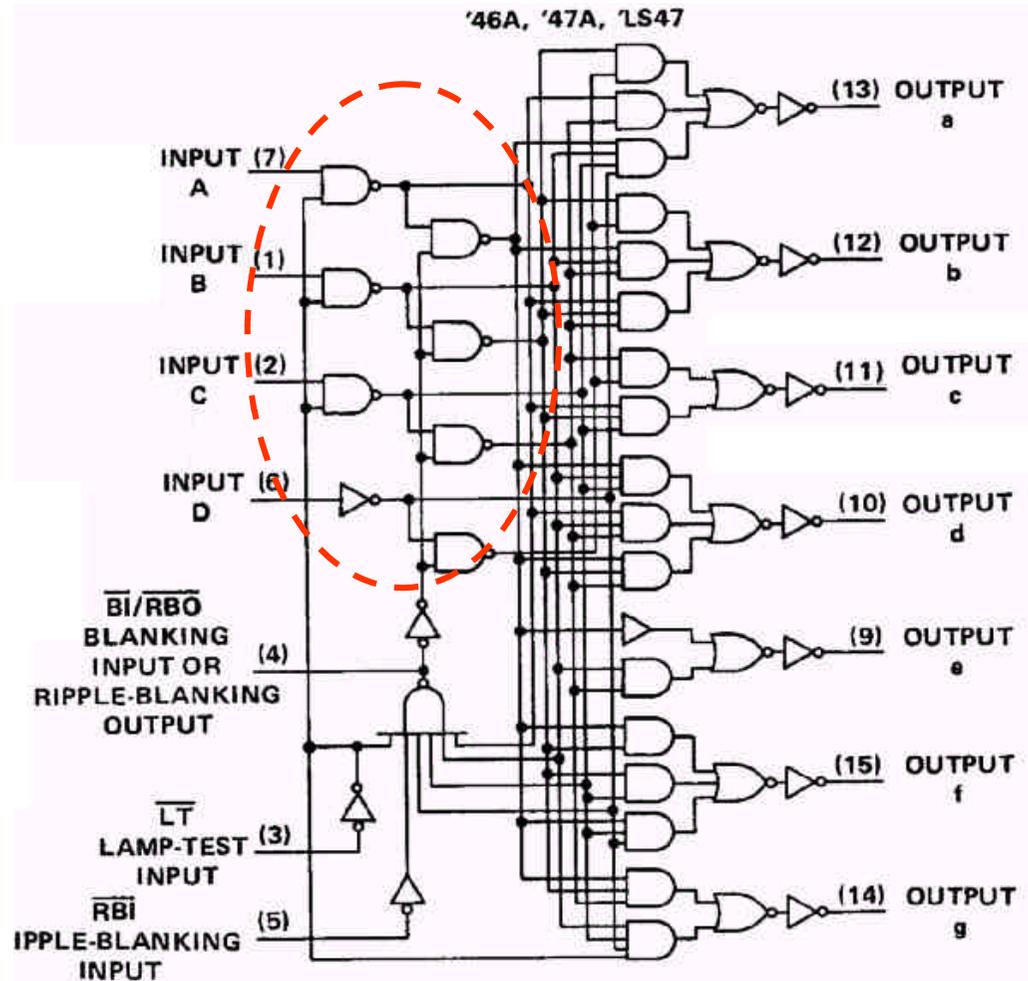
In effetti se $BIR' = 0$

$I^* = -$

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

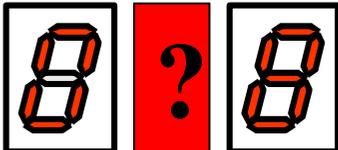


Se $BIR' = 1$ e $LT' = 0$

$C, B, A = 0$

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 5 segnali di ingresso (a, b, c, d, e) e da 1 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:

ab/cde	000	001	011	010	100	101	111	110
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

ab/cde	000	001	011	010	100	101	111	110
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 applicando i metodi di Petrick e di Quine-McCluskey.

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

STEP 1: Individuazione di tutti i raggruppamenti rettangolari primi

Colonna RR_k ($k=0,1,2,3$): 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 ($n=0,1,2,\dots$) in base al numero n di variabili che assumono il valore 1 nell'ambito di tutte le celle di un raggruppamento.

	RR_0			RR_1			RR_2			RR_3	
g_0	00000	Ö	g_0	000-0	Ö	g_0	0-0-0	Ö	g_0	--0-0	G_3
g_1	00010	Ö		0-000	Ö		-00-0	Ö	g_1	1---0	H_3
	01000	Ö		-0000	Ö		--000	Ö			
	10000	Ö	g_1	0001-	A_1	g_1	--010	Ö			
g_2	00011	Ö		0-010	Ö		-10-0	Ö			
	01010	Ö		-0010	Ö		-1-00	C_2			
	01100	Ö		010-0	Ö		10--0	Ö			
	10010	Ö		01-00	Ö		1-0-0	Ö			
	10100	Ö		-1000	Ö		1--00	Ö			
	11000	Ö		100-0	Ö	g_2	-110-	D_2			
g_3	00111	Ö		10-00	Ö		1--10	Ö			
	01101	Ö		1-000	Ö		1-1-0	Ö			
	10110	Ö	g_2	00-11	B_1		110--	E_2			
	11001	Ö		-1010	Ö		11-0-	F_2			
	11010	Ö		0110-	Ö		11--0	Ö			
	11100	Ö		-1100	Ö						
g_4	11011	Ö		10-10	Ö						
	11101	Ö		1-010	Ö						
	11110	Ö		101-0	Ö						
				1-100	Ö						
				1100-	Ö						
				110-0	Ö						
				11-00	Ö						
			g_3	-1101	Ö						
				1-110	Ö						
				110-1	Ö						
				11-01	Ö						
				1101-	Ö						
				11-10	Ö						
				1110-	Ö						
				111-0	Ö						

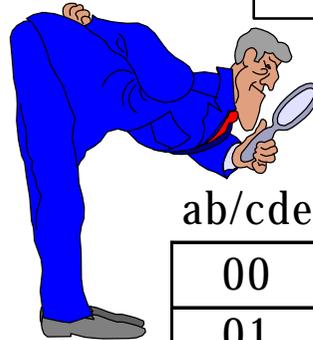
Ö:
raggruppamento rettangolare non primo

"_k (" = A, B, C, ...):
raggruppamento rettangolare primo di ordine k

I metodi di Petrick e di 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:
A₁ B₁ C₂ D₂ E₂ F₂ G₃ H₃



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 $z = 1$

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

ab/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)

STEP 3: Soluzione per via grafica del problema della copertura

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

RR primi di celle per
cui $z = 1$ 0 -

	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		

celle per cui la copertura è univoca

raggruppamenti rettangolari essenziali



ab/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)

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.

celle ancora da coprire

	2
A₁	x
C₂	
F₂	
G₃	x

tabella di copertura ridotta



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

numero/- celle già coperte

Rimozione delle righe C_2, F_2 (raggruppamenti che coprono celle già coperte dall'unione dei raggruppamenti essenziali).

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

Selezione del raggruppamento G_3 in quanto di ordine superiore. **Copertura minima:**
 $B_1 D_2 E_2 G_3 H_3$.

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 $z = 1$

RR primi di celle
per cui $z = 1$ 0 -

	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		

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 di Quine-McCluskey (espressione PS)

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 0 o -.

In ogni insieme i raggruppamenti sono partizionati in gruppi g_n ($n=0,1,2,\dots$) in base al numero n di variabili che assumono il valore 1 nell'ambito di tutte le celle di un raggruppamento. I simboli $\ddot{0}$ e $"_k$ (" $=A,B,C,\dots$) differenziano i raggruppamenti non primi da quelli primi, rispettivamente.

	RR_0			RR_1			RR_2	
g_0	00000	$\ddot{0}$	g_0	0000-	$\ddot{0}$	g_0	00-0-	B_2
g_1	00001	$\ddot{0}$		00-00	$\ddot{0}$		-000-	C_2
	00100	$\ddot{0}$		0-000	$\ddot{0}$		0-00-	D_2
	01000	$\ddot{0}$		-0000	$\ddot{0}$	g_1	0-0-1	E_2
	10000	$\ddot{0}$	g_1	000-1	$\ddot{0}$		-00-1	F_2
g_2	00011	$\ddot{0}$		00-01	$\ddot{0}$		-0-01	G_2
	00101	$\ddot{0}$		0-001	$\ddot{0}$		010--	H_2
	00110	$\ddot{0}$		-0001	$\ddot{0}$	g_2	--110	I_2
	01001	$\ddot{0}$		0010-	$\ddot{0}$		01-1-	L_2
	01010	$\ddot{0}$		001-0	A_1		-1-10	M_2
	10001	$\ddot{0}$		010-0	$\ddot{0}$		10--1	N_2
g_3	01011	$\ddot{0}$		0100-	$\ddot{0}$	g_3	-111-	O_2
	01110	$\ddot{0}$		1000-	$\ddot{0}$		1-1-1	P_2
	10011	$\ddot{0}$	g_2	0-011	$\ddot{0}$		111--	Q_2
	10101	$\ddot{0}$		-0011	$\ddot{0}$		1-11-	R_2
	10110	$\ddot{0}$		-0101	$\ddot{0}$			
	11010	$\ddot{0}$		0-110	$\ddot{0}$			
	11100	$\ddot{0}$		-0110	$\ddot{0}$			
g_4	01111	$\ddot{0}$		0101-	$\ddot{0}$			
	10111	$\ddot{0}$		010-1	$\ddot{0}$			
	11101	$\ddot{0}$		01-10	$\ddot{0}$			
	11110	$\ddot{0}$		-1010	$\ddot{0}$			
g_5	11111	$\ddot{0}$		100-1	$\ddot{0}$			
				10-01	$\ddot{0}$			
			g_3	01-11	$\ddot{0}$			
				0111-	$\ddot{0}$			
				-1110	$\ddot{0}$			
				10-11	$\ddot{0}$			
				101-1	$\ddot{0}$			
				1011-	$\ddot{0}$			
				11-10	$\ddot{0}$			
				1-110	$\ddot{0}$			
				1110-	$\ddot{0}$			
				111-0	$\ddot{0}$			
			g_4	-1111	$\ddot{0}$			
				1-111	$\ddot{0}$			
				111-1	$\ddot{0}$			
				1111-	$\ddot{0}$			

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

STEP 2: Costruzione della tabella di copertura

Evidenziazione sulla mappa di Karnaugh (istanziata 4 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



RR primi di 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	

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

ab/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/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	-

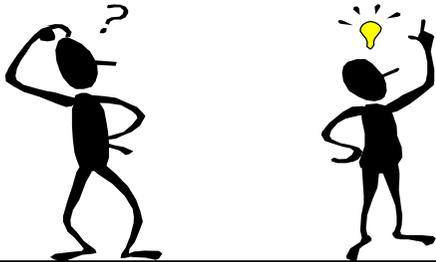
ab/cde	000	001	011	010	100	101	111	110
00	-	1	-	-	4	5	-	6
01	-	9	11	-	-	-	15	14
11	-	-	-	-	-	-	31	-
10	-	17	19	-	-	21	23	-

ab/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)

STEP 3: Soluzione per via grafica del problema della copertura

Nessuna riga essenziale



Rimozione delle colonne dominanti
 $1 \hat{=} 5, 14 \hat{=} 15, 17 \hat{=} 19$

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

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

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

ab/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



RR primi residui

ab/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	-

celle residue

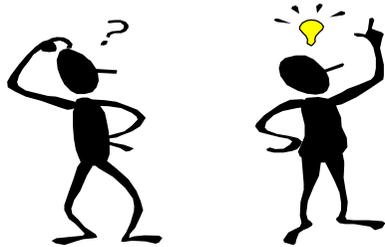
	4	5	6	9	11	15	19	21	23	31
A_1	x		x							
B_2	x	x								
C_2										
D_2				x						
E_2				x	x					
F_2							x			
G_2		x						x		
H_2				x	x					
I_2			x							
L_2					x	x				
M_2										
N_2							x	x	x	
O_2						x				x
P_2								x	x	x
Q_2										x
R_2									x	x

numero

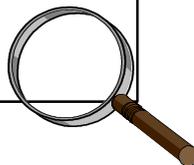
celle la cui copertura discende automaticamente dalla copertura di altre celle

Il metodo di Quine-McCluskey (espressione PS)

2^a tabella di copertura ridotta



Rimozione delle **righe dominate** D_2 , E_2 , F_2 , N_2 , Q_2 , R_2 . La riga I_2 , pur essendo dominata dalla riga A_1 , 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

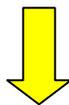
ab/cde	000	001	011	010	100	101	111	110
00	-	1	-		4	5		6
01	-	9	11	-			15	14
11				-	-	-	31	-
10	-	17	19			21	23	-

Il metodo di Quine-McCluskey (espressione PS)

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** $11 \hat{=} 9$.



4^a tabella di copertura ridotta

Rimozione delle **righe dominate** $G_2 \hat{=} B_2, L_2 \hat{=} O_2, P_2 \hat{=} O_2$ ($I_2 \hat{=} A_1$, 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

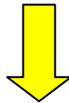
	4	5	6	9	15	31
A ₁	x		x			
B ₂	x	x				
E ₂				x		
G ₂		x				
H ₂				x		
I ₂			x			
L ₂					x	
O ₂					x	x
P ₂						x

RR primi residui

Il metodo di Quine-McCluskey (espressione PS)

5^a tabella di copertura ridotta

Selezione delle righe ora essenziali B_2 e O_2 . Rimozione di tali righe e delle colonne da essa 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
RR primi residui	A_1	x		x			
	B_2	x	x				
	E_2				x		
	H_2				x		
	I_2			x			
	O_2					x	x

		celle residue	
		6	9
RR primi residui	A_1	x	
	E_2		x
	H_2		x
	I_2	x	

Il metodo di Quine-McCluskey (espressione PS)

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

Espressioni minime:

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

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

ab/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/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 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
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 $z = 0$

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)

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

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

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

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

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

Il metodo di Petrick

(espressione PS)

/* 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:

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

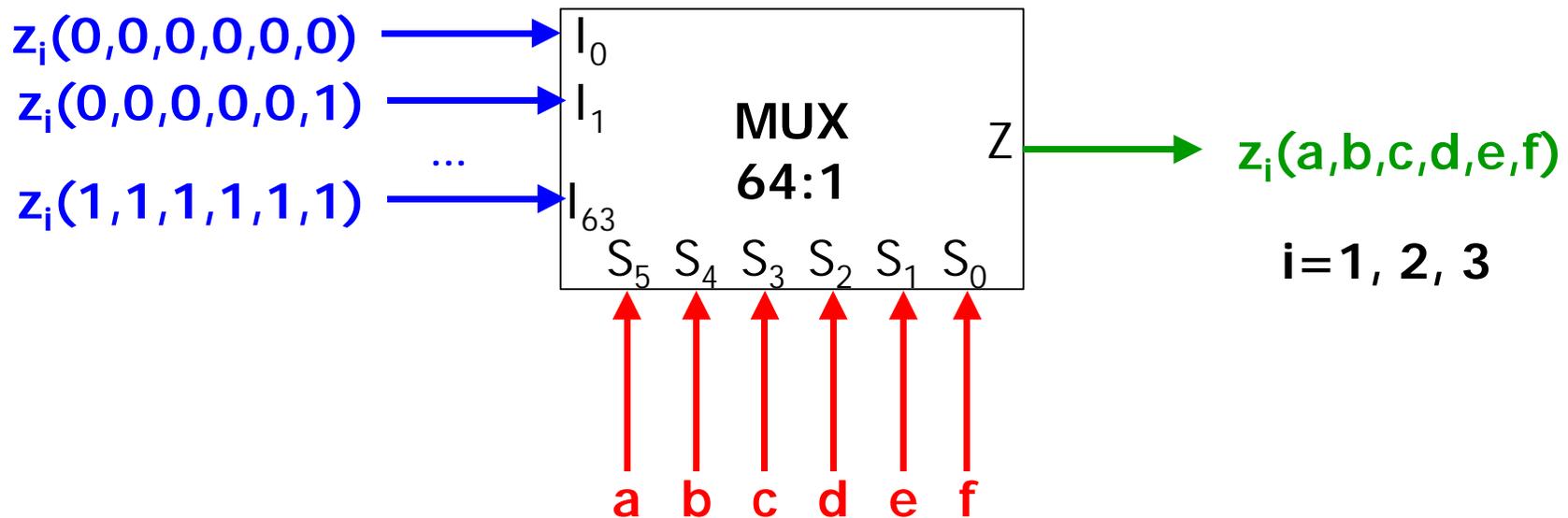
$$z = (a' + b + e') (a + b + d) (b' + c' + d') (c' + d' + e) (a + b' + c)$$

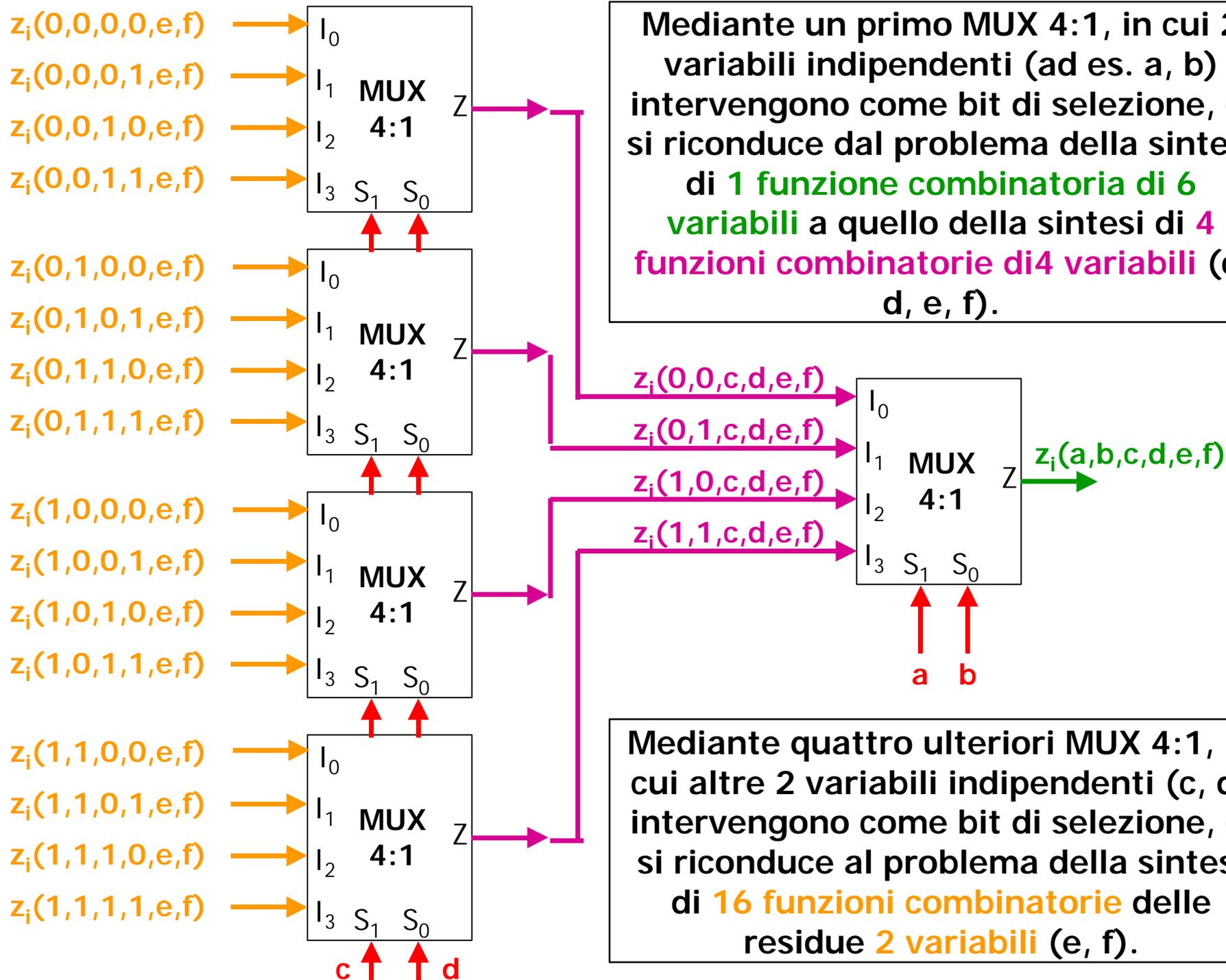
Esercizio 3

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.

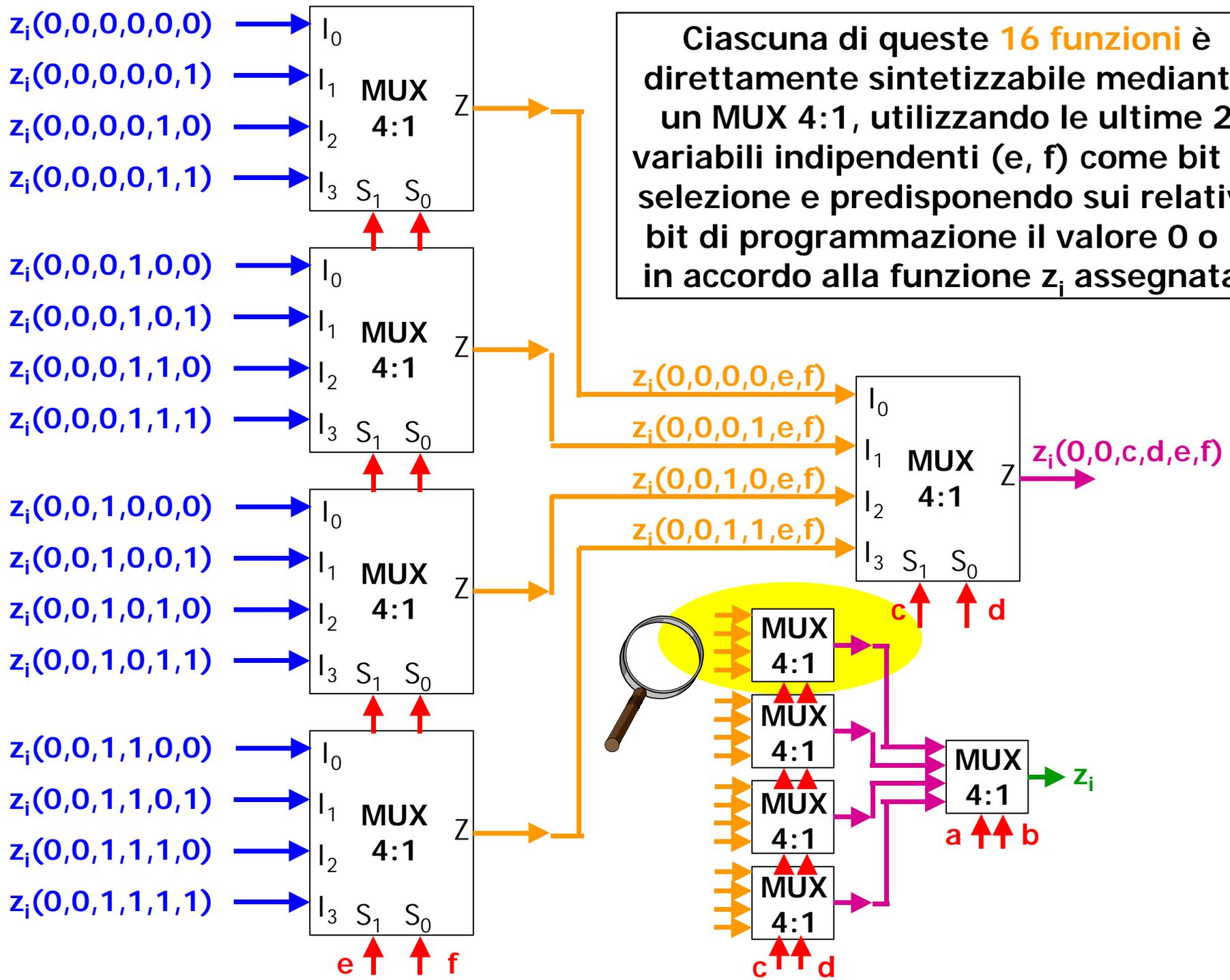
Ciascuna sottorete deve essere strutturata in maniera tale da dare luogo ad un comportamento equivalente a quello di un MUX con 6 bit di selezione e $2^6=64$ bit di programmazione.





Mediante un primo MUX 4:1, in cui 2 variabili indipendenti (ad es. a, b) intervengono come bit di selezione, ci si riconduce dal 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).



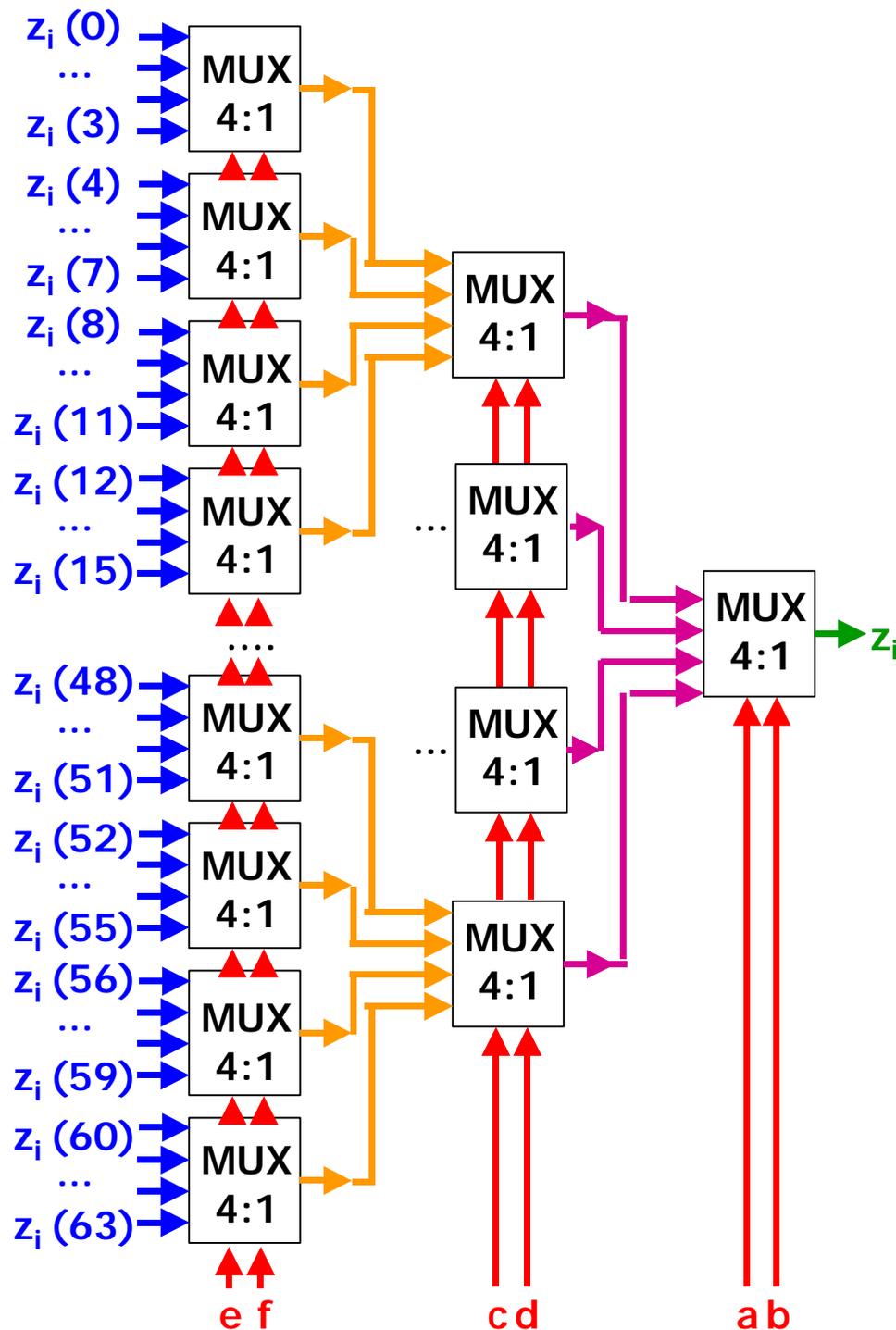
Ciascuna di queste **16 funzioni** è direttamente sintetizzabile mediante un MUX 4:1, utilizzando le ultime 2 variabili indipendenti (e, f) come bit di selezione e predisponendo sui relativi bit di programmazione il valore 0 o 1 in accordo alla funzione z_i assegnata.

$z_i(0,0,0,0,e,f)$
 $z_i(0,0,0,1,e,f)$
 $z_i(0,0,1,0,e,f)$
 $z_i(0,0,1,1,e,f)$

$a \uparrow \uparrow b$

$c \uparrow \uparrow d$

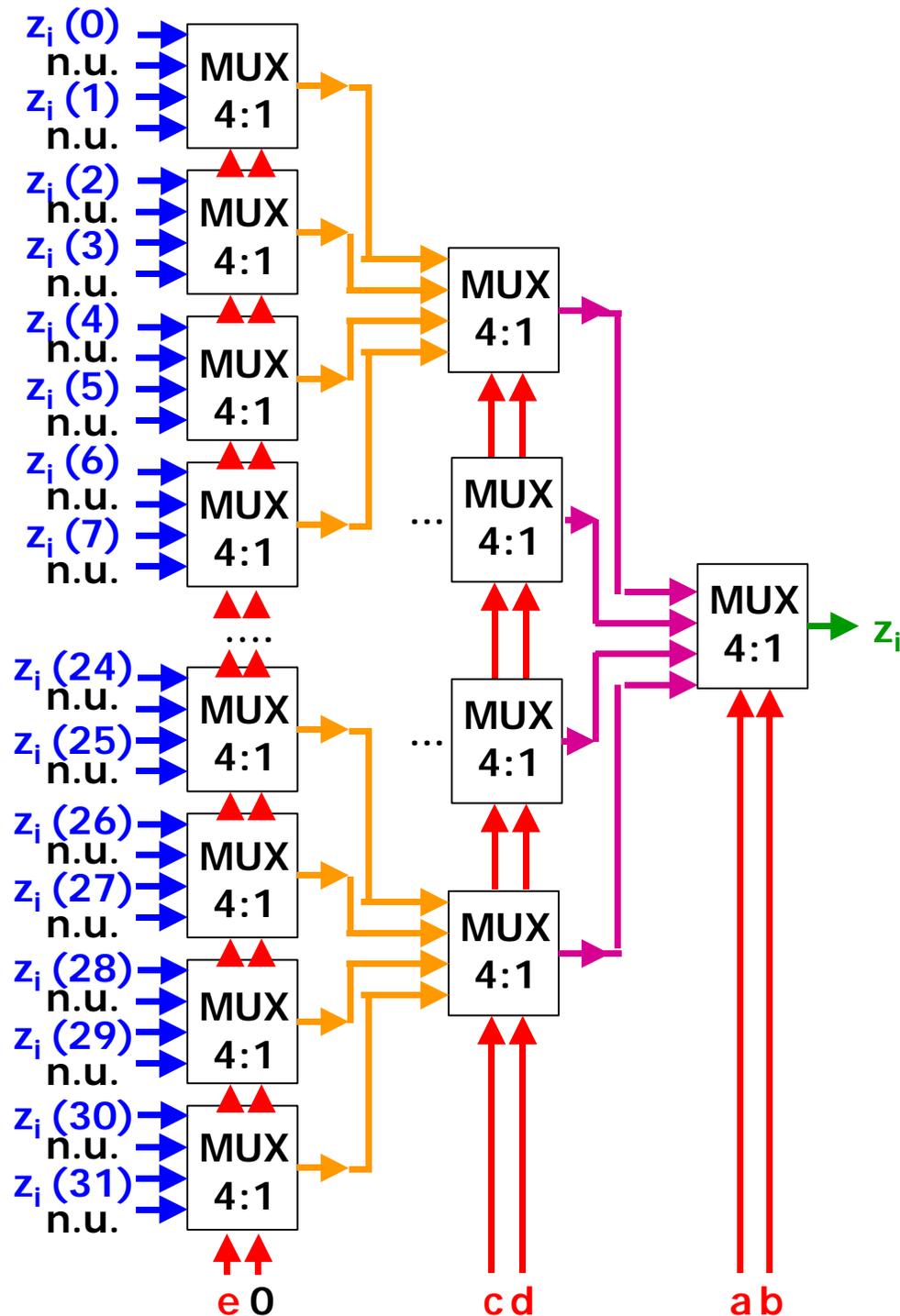
z_i



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

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



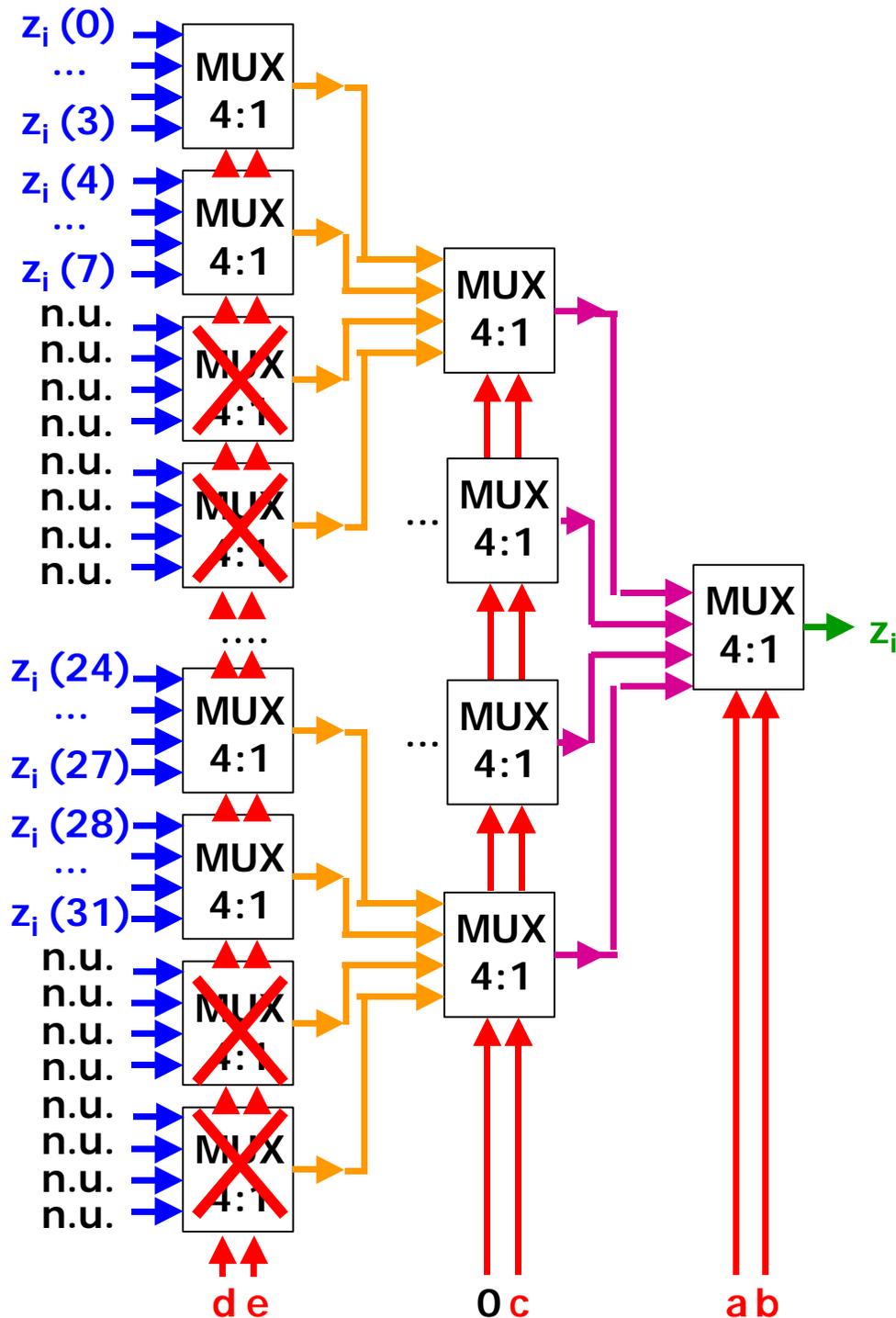
e se $(n \bmod k) > 0$???

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, e 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: $n=5, k=2$

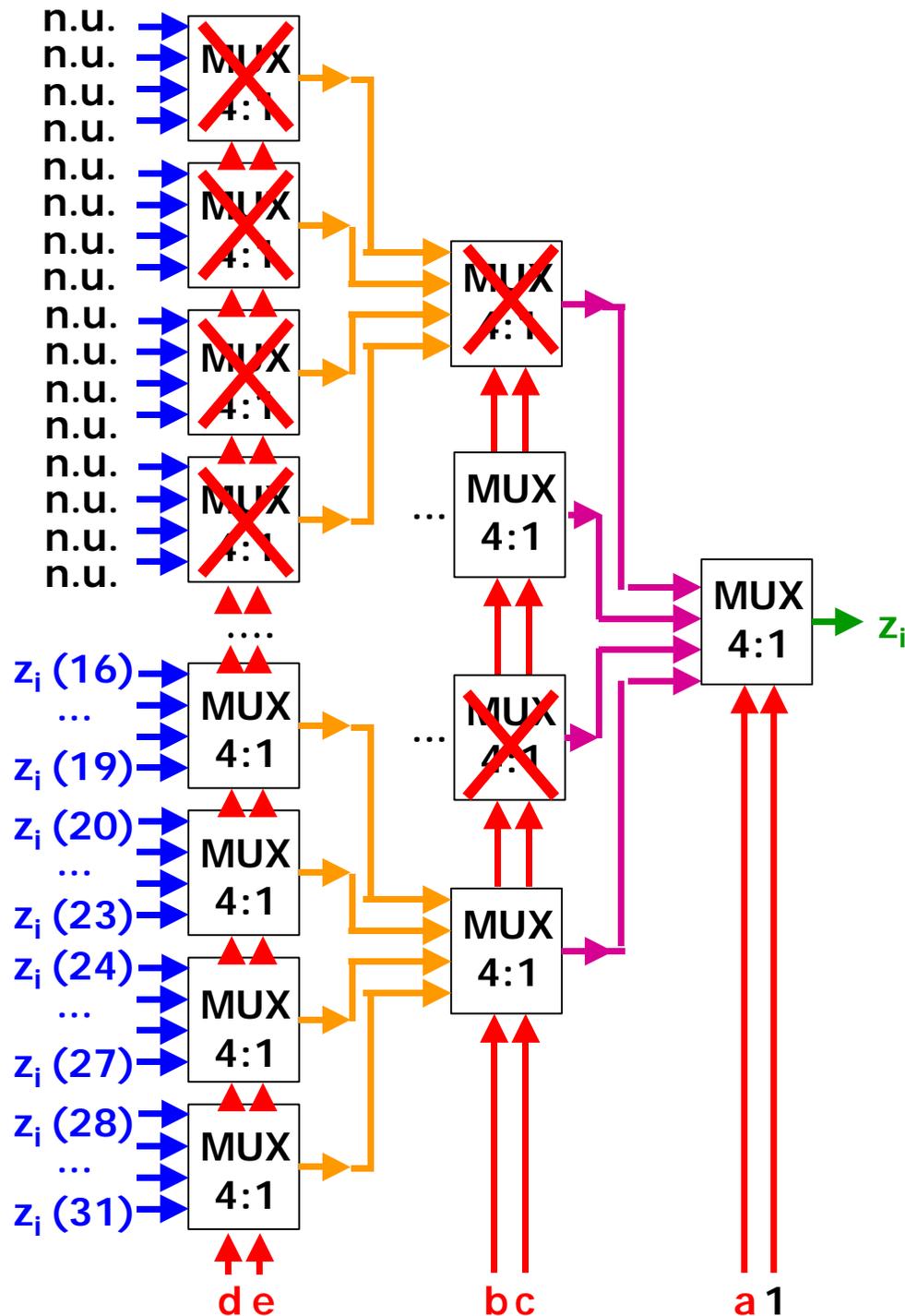
$z_i(a,b,c,d,e)$

Una seconda soluzione:

2 variabili indipendenti (a, b) intervengono come bit di selezione al 1° livello, 1 altra (c) al 2° livello, e le ultime 2 (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, e 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.

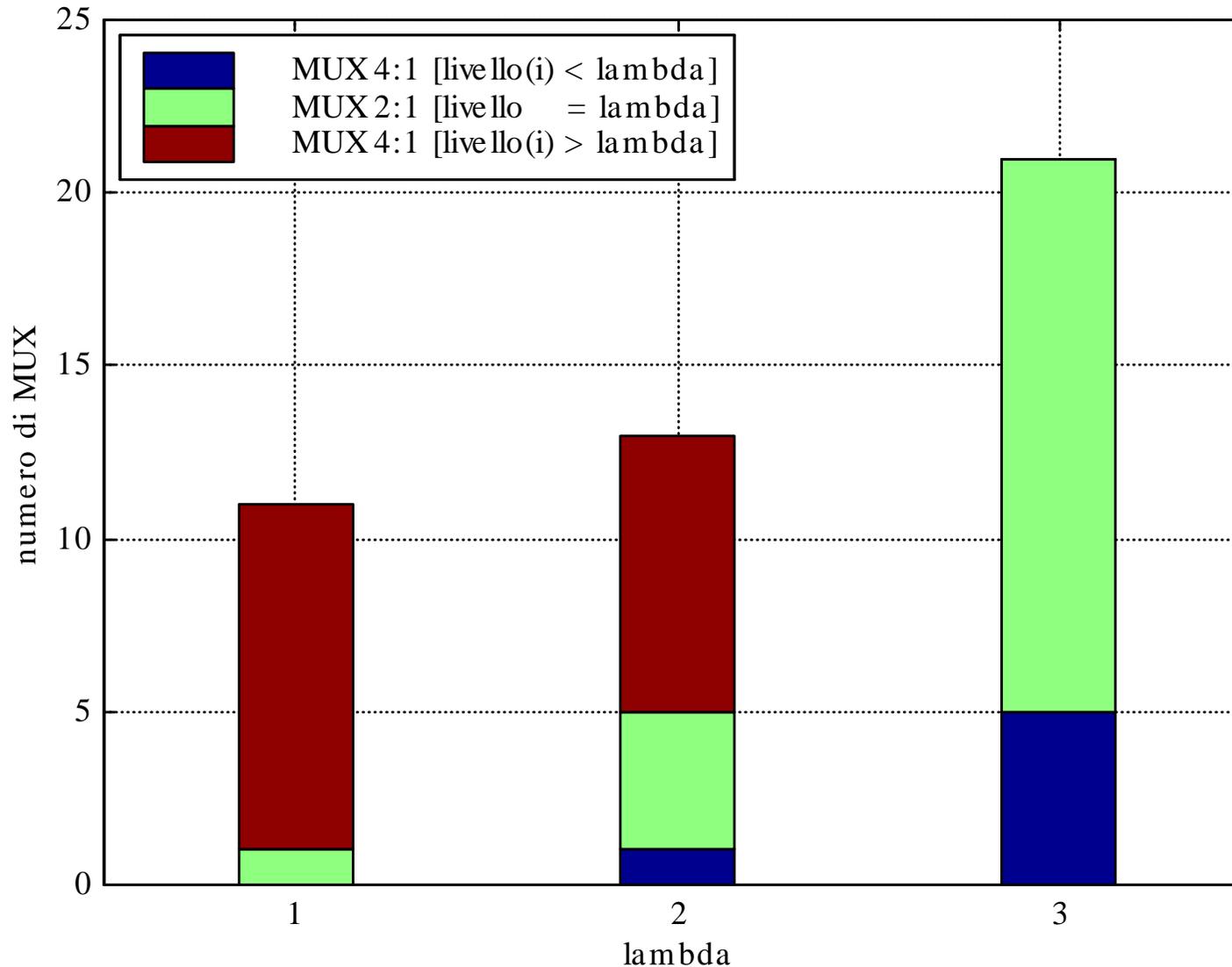
Occorrono 11 (10+1) MUX.

In generale, per " coppia (n, k) con
 $n = i k + h, i = \lceil n/k \rceil > 0, h = n \bmod k > 0,$
 si hanno $L=i+1$ livelli di elaborazione. In ogni livello intervengono
 come bit di selezione k variabili indipendenti, ad eccezione di un
 livello l (l = 1, 2, ..., L) in cui le variabili sono solo h.

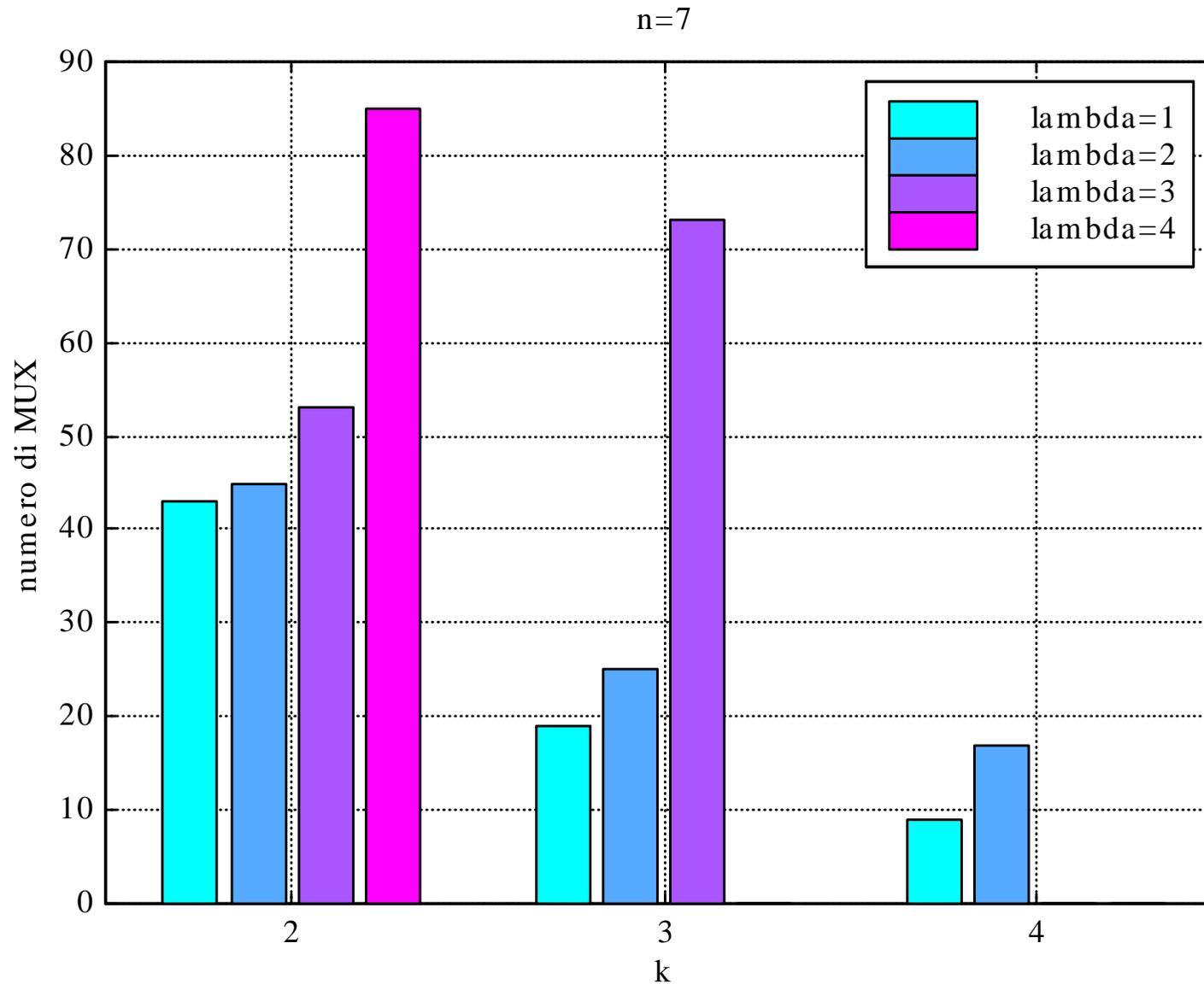
livello	L	...	l+2	l+1	l	l-1	...	2	1
n° variabili	k	...	k	k	h	k	...	k	k
n° MUX	$2^{(L-2)k+h}$...	2^{1k+h}	$2^{(l-1)k+h}$	$2^{(l-1)k}$	$2^{(l-2)k}$...	2^k	1
tipologia MUX	$2^k:1$			$2^h:1$	$2^k:1$				
n° MUX $2^h:1$	$2^{(\lambda-1)k}$								
n° MUX $2^k:1$	$\frac{2^n - 2^{(\lambda-1)k+h}}{2^k - 1} + \frac{2^{(\lambda-1)k} - 1}{2^k - 1} = \frac{2^n - 1 - 2^{(\lambda-1)k} (2^h - 1)}{2^k - 1}$								
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 l aumenta il numero di MUX $2^k:1$, ma contemporaneamente diminuisce, ed in maniera più rilevante, il numero di MUX $2^h:1$.

$n=5, k=2$: tipologie di MUX e corrispondenti livelli



Al fine di ottenere la soluzione che coinvolge il minor numero di MUX conviene pertanto scegliere $l = 1$, 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 $l=1$ 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 ... n/k

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 $2^k:1$

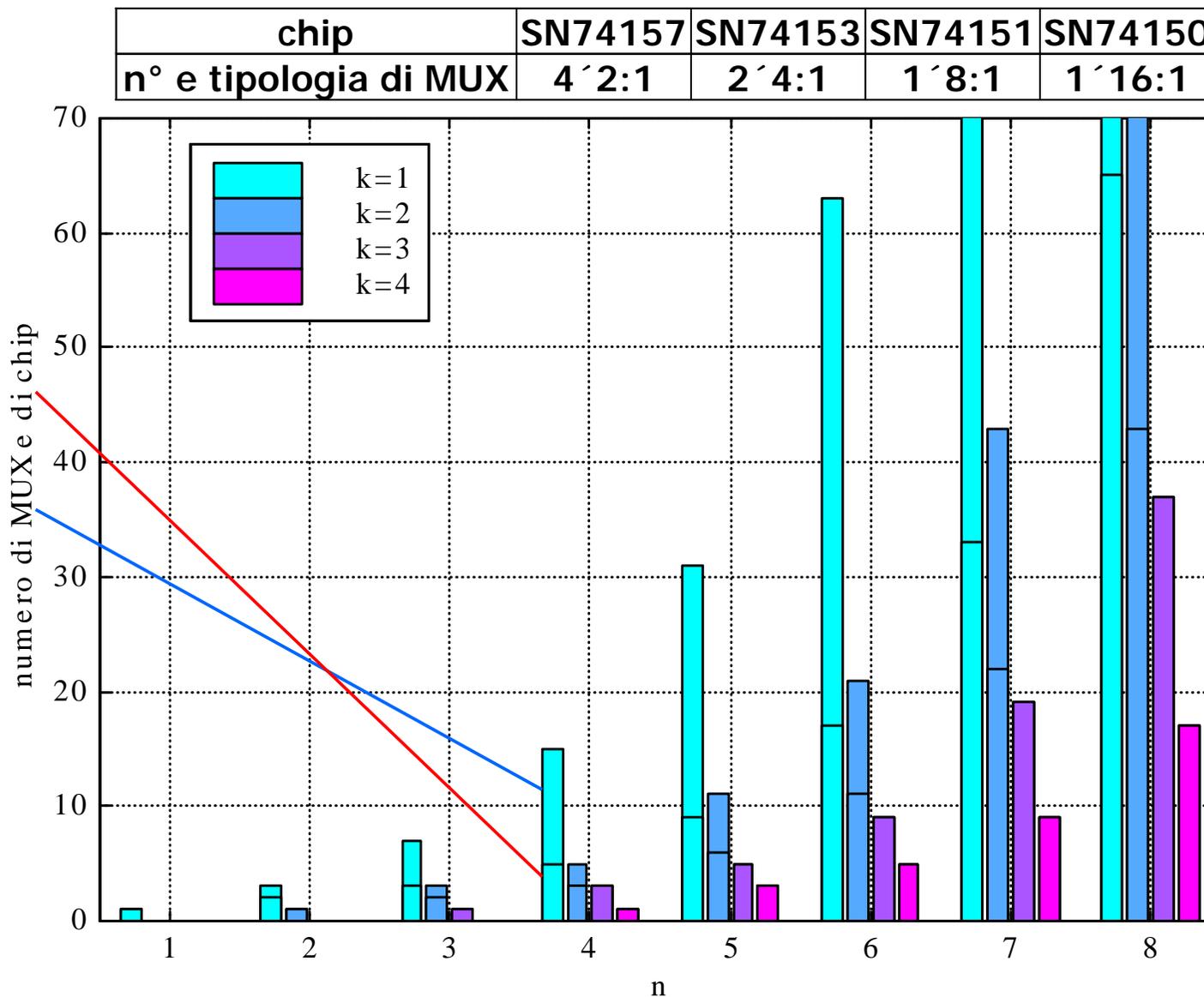
livello

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 :

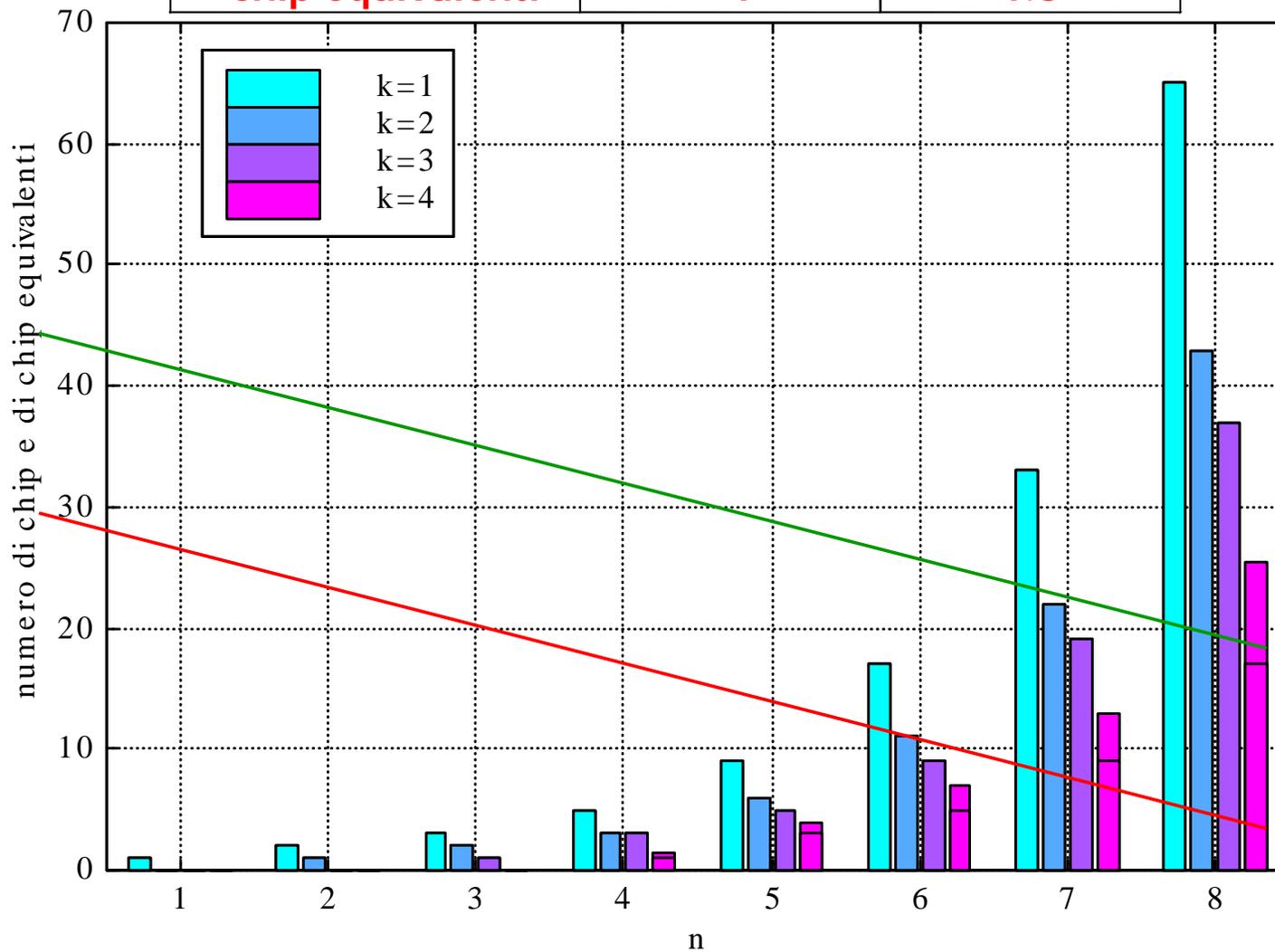
$$\min \{n_{\text{MUX}}(n, k)\} = n_{\text{MUX}}(n, k_{\text{max}}).$$

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 ...

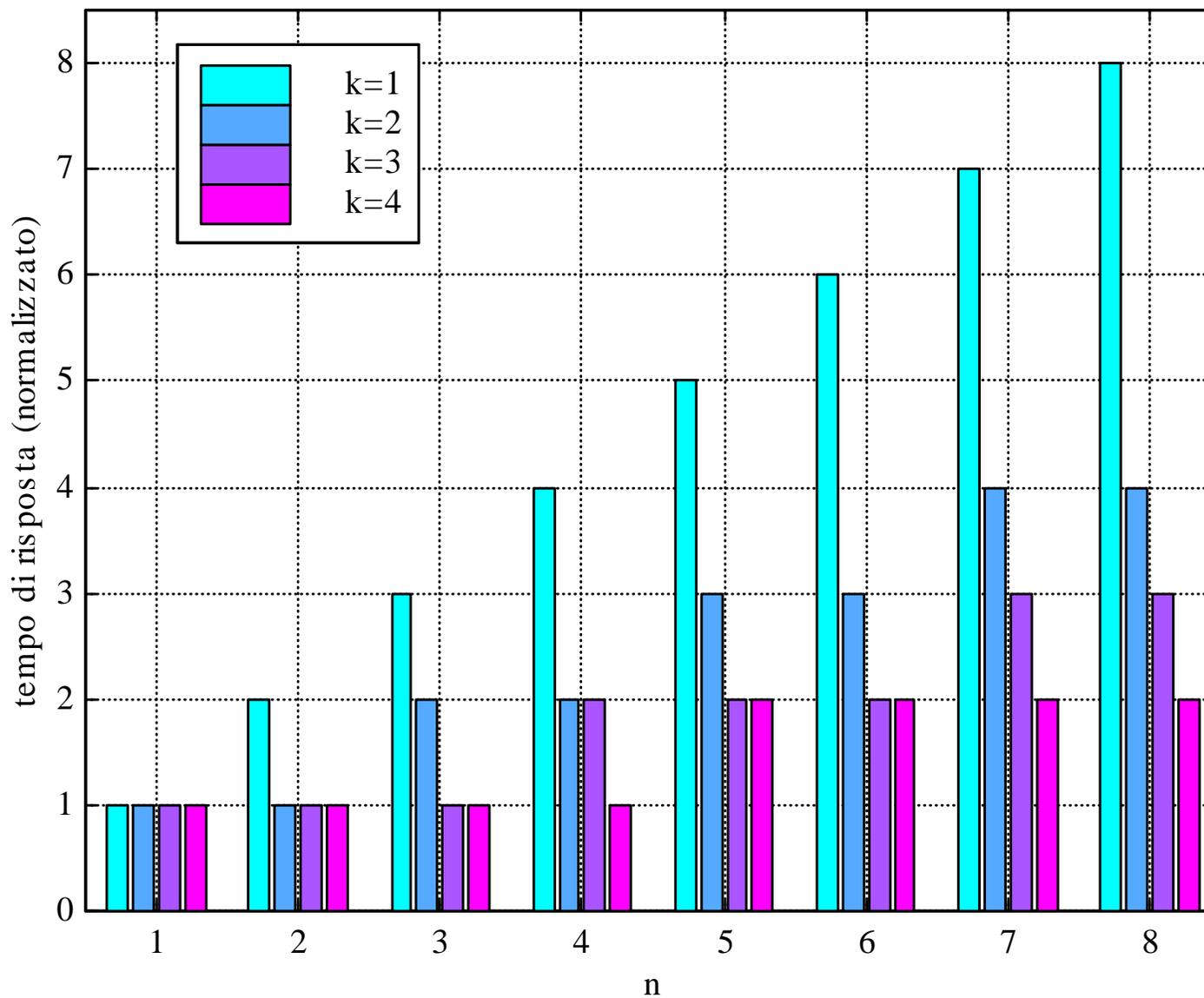


... ed aumentano le dimensioni , il n° di pin ed il consumo dei chip.

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.



A yellow starburst shape with a black outline, containing the text "Reti sequenziali asincrone".

**Reti
sequenziali
asincrone**

Esercizio 1

Una rete sequenziale asincrona è caratterizzata da due segnali di ingresso (E, X) e da un segnale di uscita (Z). I segnali di ingresso non variano mai contemporaneamente, né può X cambiare di valore allorché $E = 1$. Z deve assumere il valore 0 allorché $E = 0$. Quando $E = 1$, Z deve assumere il valore 1 se e soltanto se X presenta lo stesso valore assunto nel precedente intervallo di attivazione di E.

Si determini:

- il diagramma degli stati della rete;
- la realizzazione di costo minimo della rete mediante NAND.

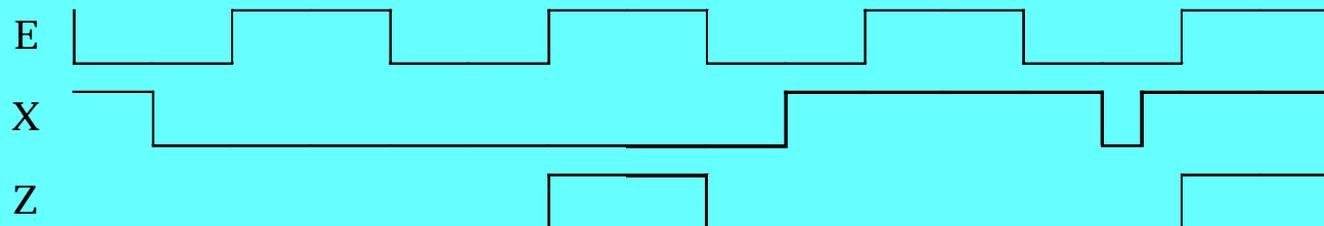
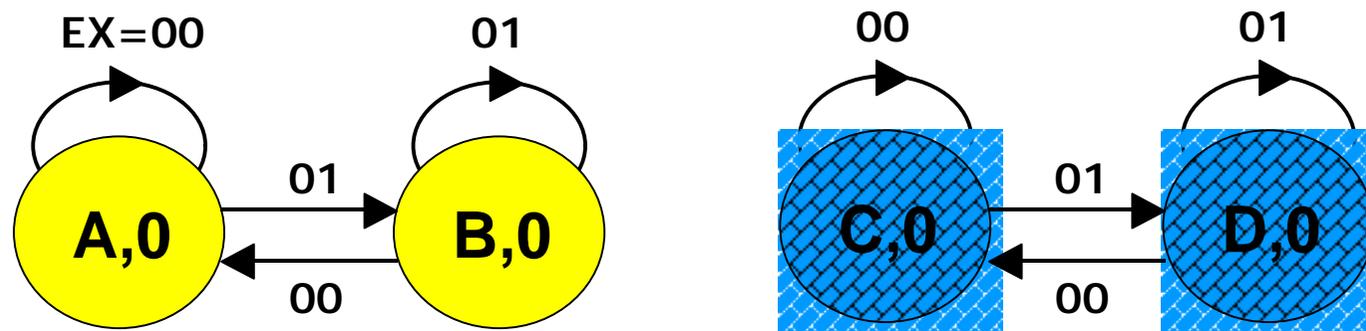


Diagramma degli stati primitivo (modello di Moore)

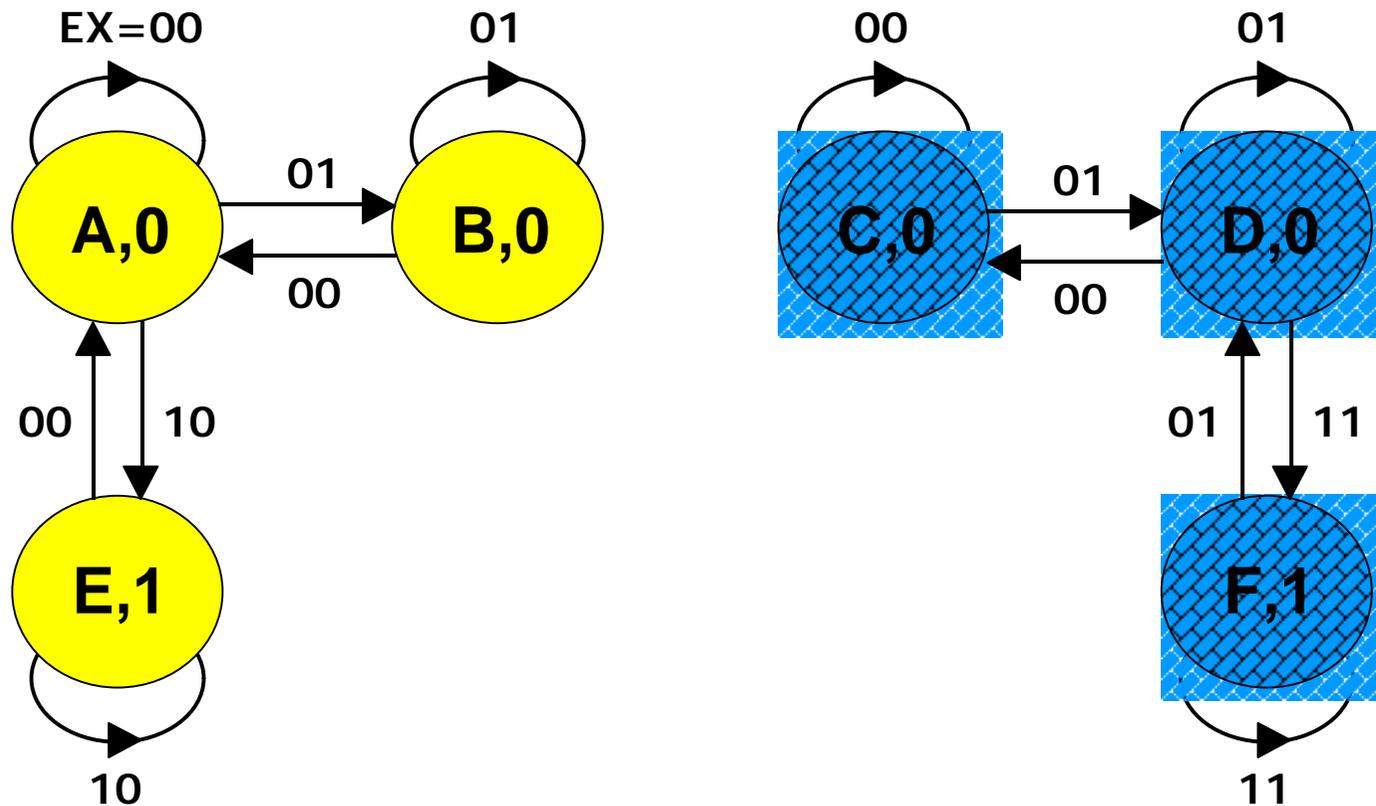
Il diagramma degli stati può essere costruito considerando innanzitutto le diverse situazioni in cui può trovarsi ad operare la rete allorché in ingresso si ha $E=0$. Al di là del fatto che l'uscita Z deve comunque valere 0, occorre prevedere due distinte coppie di stati, contraddistinte dal valore (0 o 1) precedentemente assunto da X nell'ultimo intervallo di attivazione di E . Nell'ambito di ciascuna coppia, eventuali variazioni di X provocano semplicemente la transizione tra i due stati, stabili l'uno per $EX=00$, l'altro per $EX=01$.



Valore di X nell'ultimo intervallo di attivazione di E :



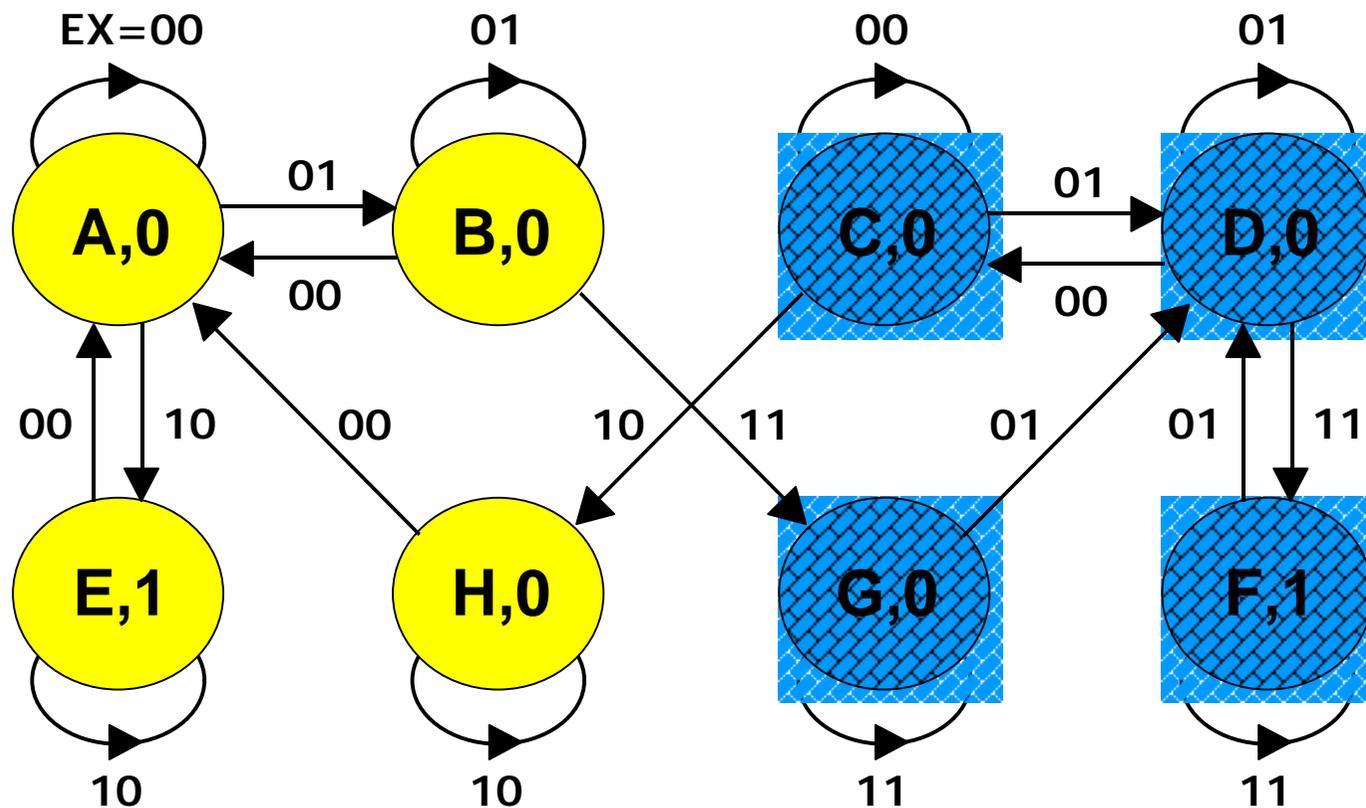
Supponiamo ora che E assuma il valore 1 e che X presenti lo stesso valore assunto nel precedente intervallo di attivazione di E. Per ciascuna coppia di stati già definiti occorre prevedere la transizione verso un nuovo stato caratterizzato da $Z=1$, dal quale si effettuerà poi la transizione opposta non appena E si riporterà a 0.



Valore di X nell'ultimo intervallo di attivazione di E:



Supponiamo infine che, quando E assume il valore 1, X abbia un valore diverso da quello corrispondente al precedente intervallo di attivazione di E. Per ciascuna coppia di stati inizialmente definiti occorre prevedere la transizione verso un nuovo stato, caratterizzato ancora da $Z=0$, dal quale si transiterà poi nell'opportuno stato dell'altra coppia quando E si riporterà a 0.



Valore di X nell'ultimo intervallo di attivazione di E:



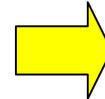
Tabella di flusso

modello di Moore

modello di Mealy

EX

	00	01	11	10	Z
A	A	B	-	E	0
B	A	B	G	-	0
C	C	D	-	H	0
D	C	D	F	-	0
E	A	-	-	E	1
F	-	D	F	-	1
G	-	D	G	-	0
H	A	-	-	H	0



EX

	00	01	11	10
A	A,0	B,0	-, -	E, -
B	A,0	B,0	G,0	-, -
C	C,0	D,0	-, -	H,0
D	C,0	D,0	F, -	-, -
E	A, -	-, -	-, -	E, 1
F	-, -	D, -	F, 1	-, -
G	-, -	D,0	G,0	-, -
H	A,0	-, -	-, -	H,0

situazioni di equilibrio

Il valore dell'uscita di norma può non essere specificato durante le transizioni di stato cui corrisponde una variazione dell'uscita.

Tabella triangolare

B							
C	BD EH	AC BD					
D	AC BD	AC FG					
E				AC			
F	BD						
G	BD	BD		FG			
H	EH		AC	AC			
	A	B	C	D	E	F	G

**modello
di Mealy**

B							
C	BD EH	AC BD					
D	AC BD	AC FG					
E							
F							
G	BD	BD		FG			
H	EH		AC	AC			
	A	B	C	D	E	F	G

**modello
di Moore**

stati compatibili

stati incompatibili per
almeno una
configurazione d'ingresso

stati compatibili per ogni
sequenza d'ingresso di
lunghezza unitaria, ma
incompatibili per almeno
una sequenza d'ingresso
di lunghezza superiore

Classi massime di compatibilità

modello di Mealy

{G,H} {F,H} {E,F} {E,G}
{C,D,F} {C,G} {A,B,E} {B,H}

{G,H} {E,F} {C,D} {C,G}
{B,H} {A,B}

modello di Moore

Tabella di flusso minima

	EX			
	00	01	11	10
ABE≡α	α,0	α,0	γ,0	α,1
CDF≡β	β,0	β,0	β,1	γ,0
GH≡γ	α,0	β,0	γ,0	γ,0

3 stati

situazioni di equilibrio

	EX				
	00	01	11	10	Z
AB≡α	α	α	δ	γ	0
CD≡β	β	β	γ	δ	0
EF≡γ	α	β	γ	γ	1
GH≡δ	α	β	δ	δ	0

4 stati

L'automata minimo di una rete sintetizzata secondo il modello di Moore coinvolge in generale un numero di stati superiore rispetto all'automata minimo di Mealy:
 diagramma degli stati ® modello di Moore
 tabella di flusso ® modello di Mealy

Diagramma delle adiacenze e mappa di codifica (modello di Mealy)

$$\alpha \Leftrightarrow \gamma$$

$$\updownarrow$$

$$\beta$$

y_1/y_2	0	1
0	α	γ
1	-	β

Tabella delle transizioni

y_1y_2/EX	00	01	11	10
00	00,0	00,0	01,0	00,1
01	00,0	11,0	01,0	01,0
11	11,0	11,0	11,1	01,0
10	--,-	--,-	--,-	--,-

Y_1Y_2, Z

Rete combinatoria di uscita mediante NAND

$$Z = y_1 E X$$

$$+ y_2' E X'$$

y_1y_2/EX	00	01	11	10
00	0	0	0	1
01	0	0	0	0
11	0	0	1	0
10	-	-	-	-

Z

$$Z = (y_1 - E - X)$$

$$- (y_2' - E - X')$$

Rete combinatoria di aggiornamento dello stato mediante NAND

y_1y_2/EX	00	01	11	10
00	0	0	0	0
01	0	1	0	0
11	1	1	1	0
10	-	-	-	-

Y_1

$$Y_1 = y_1 E' + y_1 X + y_2 E' X = (y_1 - E') - (y_1 - X) - (y_2 - E' - X)$$

y_1y_2/EX	00	01	11	10
00	0	0	1	0
01	0	1	1	1
11	1	1	1	1
10	-	-	-	-

Y_2

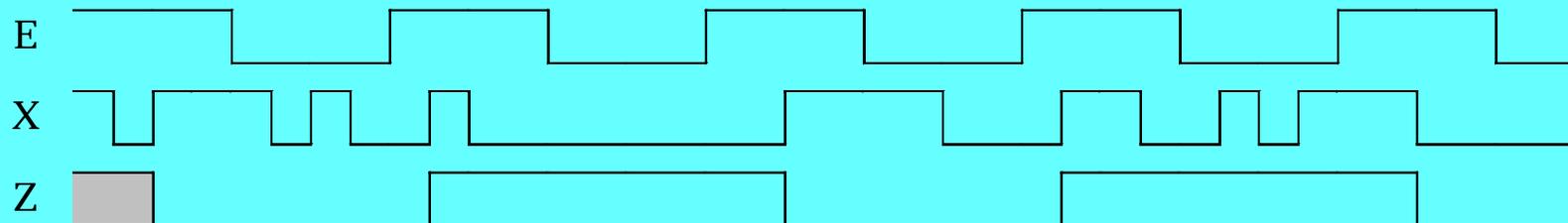
$$Y_2 = y_2 X + y_2 E + EX + y_1 = (y_2 - X) - (y_2 - E) - (E - X) - y_1'$$

Esercizio 2

Una rete sequenziale asincrona è caratterizzata da due segnali di ingresso X , E , i quali non variano mai contemporaneamente, e da un segnale di uscita Z . La rete deve generare Z tenendo conto esclusivamente dei fronti di salita e di discesa presentati da X allorché $E=1$. In particolare Z deve assumere il valore 1 se il penultimo fronte significativo di X è stato un fronte di salita, il valore 0 se il penultimo fronte significativo di X è stato un fronte di discesa.

Si identifichi:

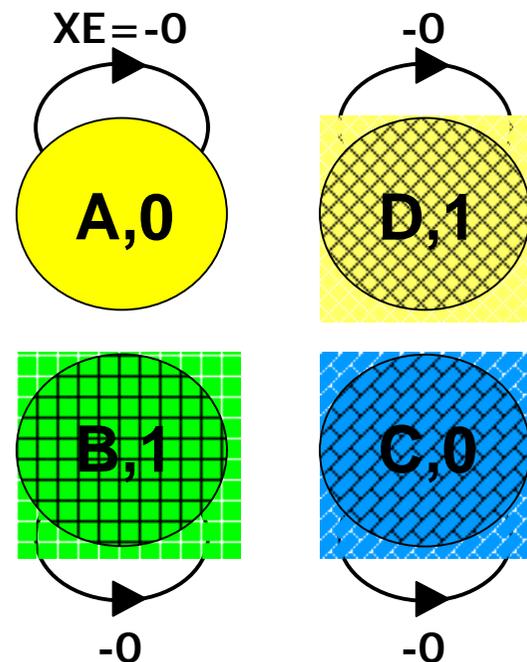
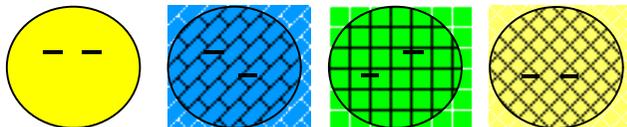
- il diagramma degli stati della rete;
- la realizzazione di costo minimo della rete mediante NOR.



Un primo diagramma degli stati non primitivo (modello di Moore)

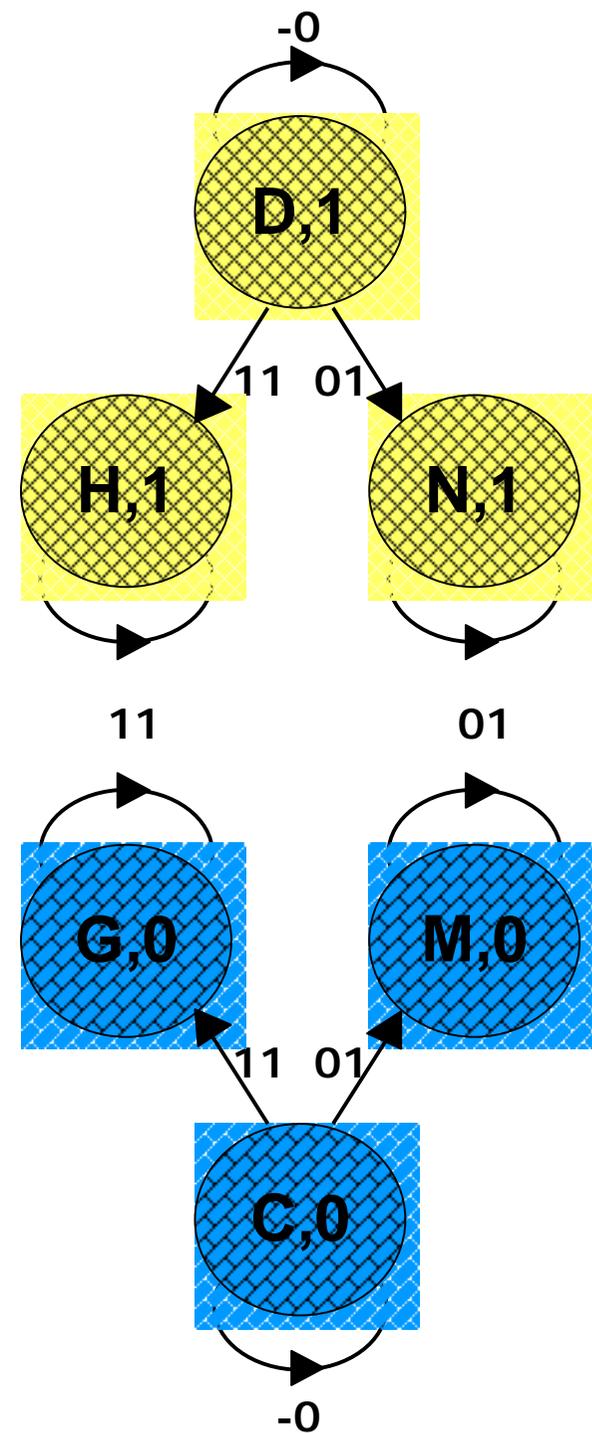
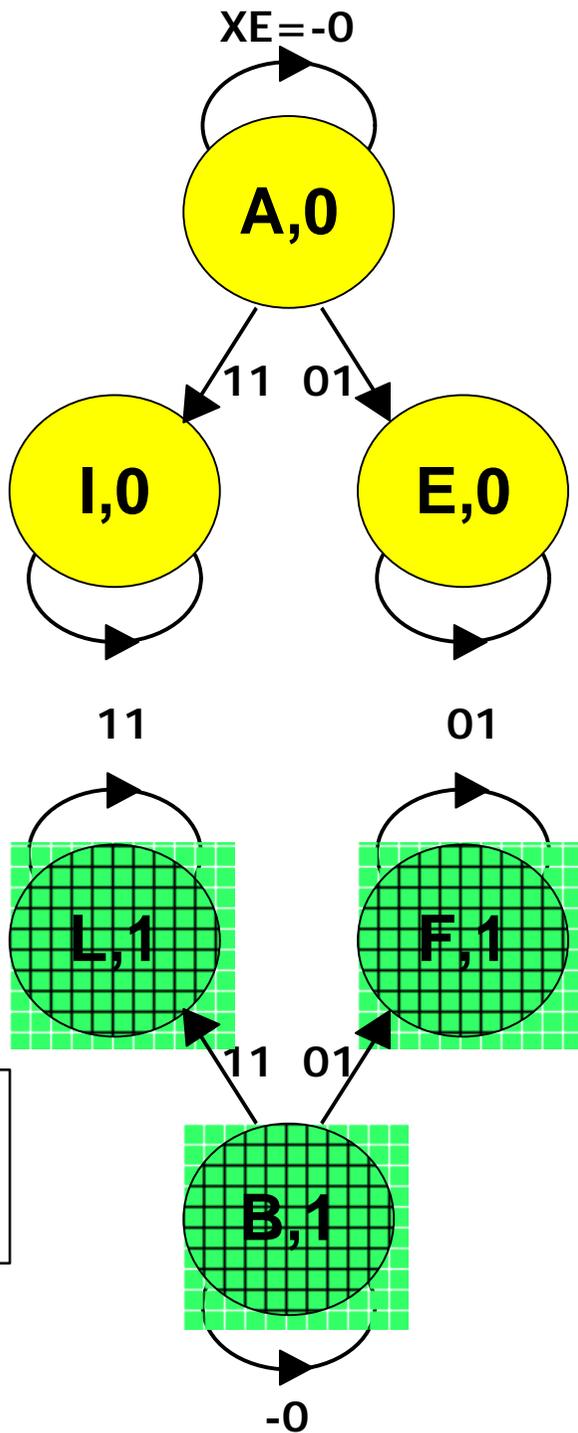
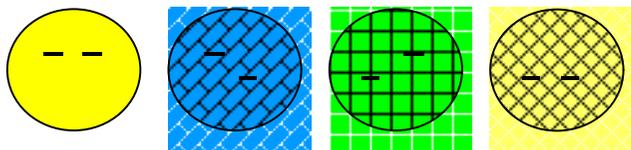
Il diagramma degli stati può essere costruito considerando innanzitutto le situazioni di ingresso caratterizzate da $E=0$, e ritenendo conseguentemente ininfluenti eventuali variazioni di X . Occorre prevedere quattro distinti stati, tutti stabili per $XE=-0$, ciascuno dei quali corrisponde ad una ben precisa combinazione degli ultimi due fronti significativi di X . Il penultimo fronte significativo di X associato a ciascuno stato ne individua univocamente il relativo valore di Z .

Penultimo e ultimo fronte significativo di X :

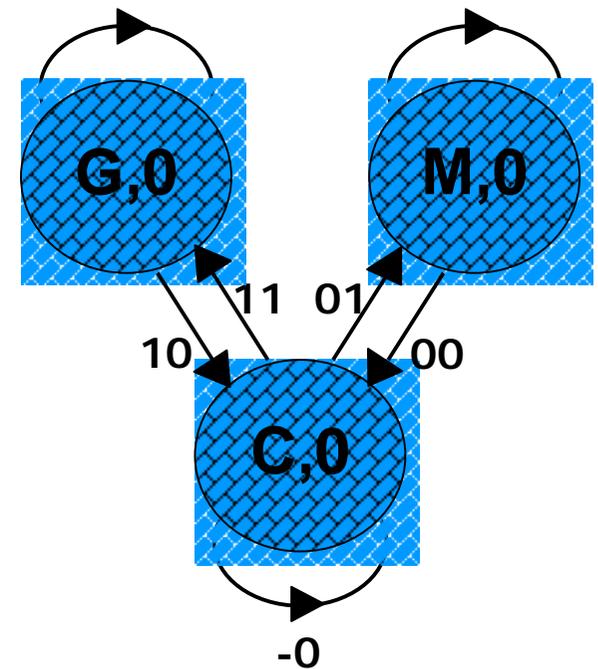
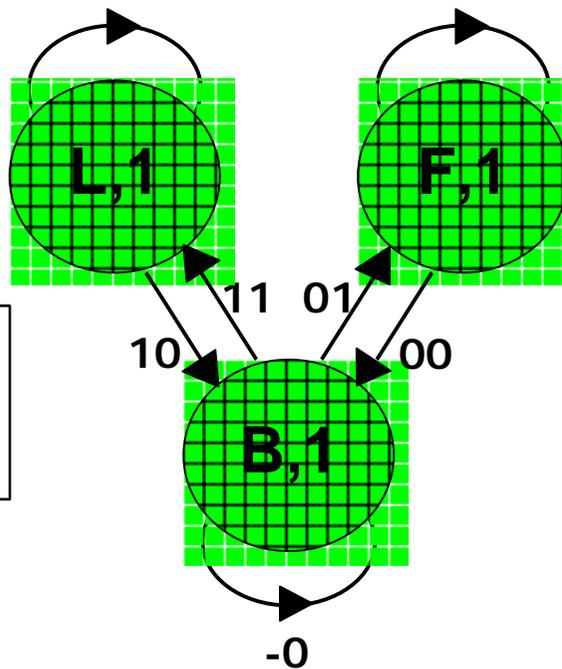
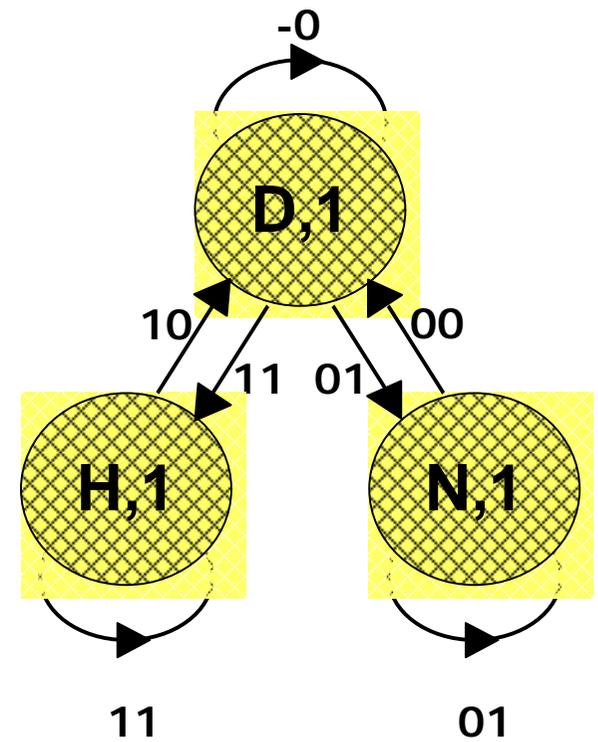
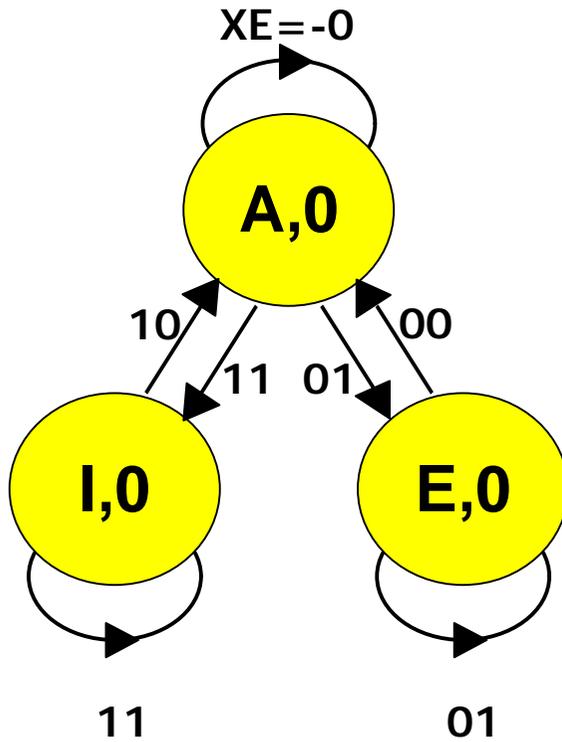


Supponiamo ora che E assuma il valore 1. Al fine di poter rilevare eventuali successive variazioni (ora significative) di X, occorre prevedere per ogni stato già definito la transizione verso due nuovi stati, caratterizzati dallo stesso valore di Z e stabili rispettivamente per X=0 e X=1:

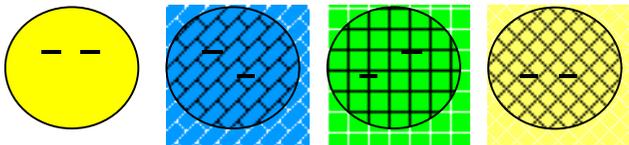
Penultimo e ultimo fronte significativo di X:



Se E si riporta a 0 prima che X cambi di valore, gli ultimi due fronti significativi di X rimangono quelli precedentemente assunti. Da tale considerazione discendono le seguenti transizioni di stato:



Penultimo e ultimo fronte significativo di X:



Il diagramma viene completato considerando infine variazioni di X in presenza del valore 1 di E, ed indicando coerentemente le transizioni di stato in base all'attuale ed al precedente fronte significativo di X.

Penultimo e ultimo fronte significativo di X:

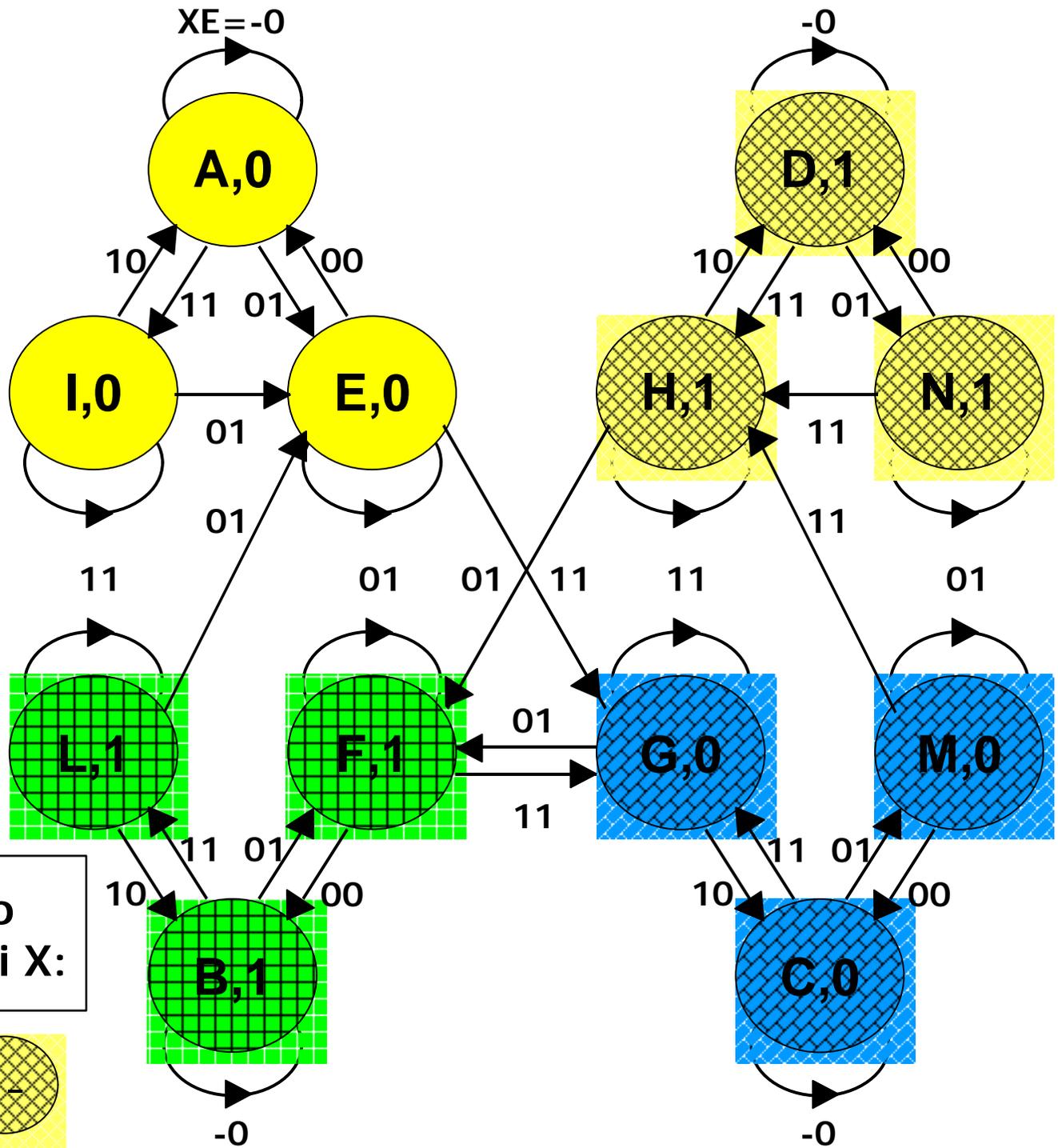
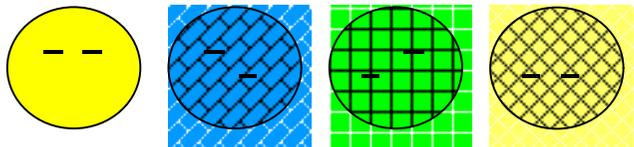
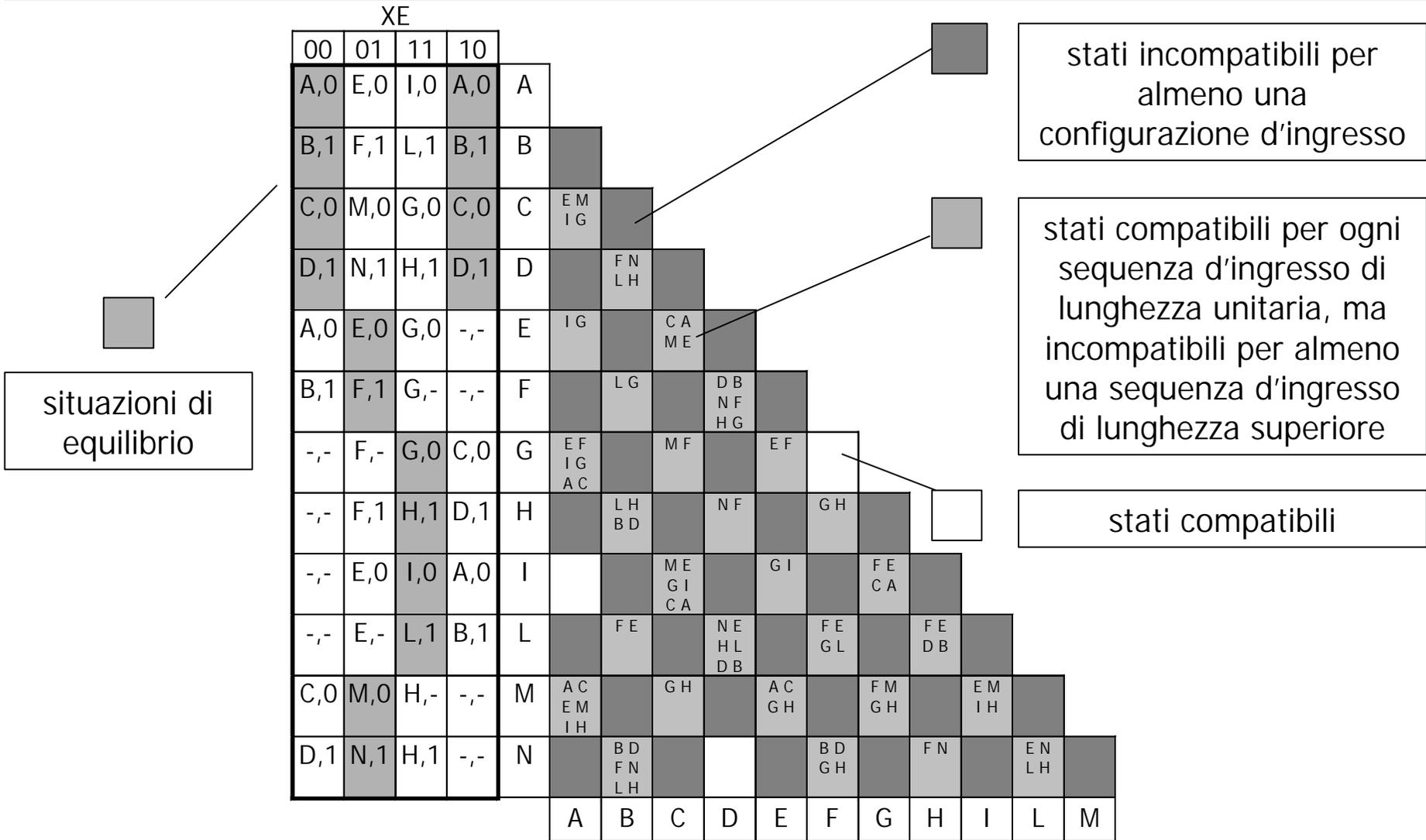


Tabella di flusso (modello di Mealy) e tabella triangolare



Classi massime di compatibilità

{A,I} {B} {C} {D,N} {E} {F,G} {H} {L} {M}

Tabella di flusso ridotta

**9
stati**

		X E			
		00	01	11	10
{A,I}	A	A,0	E,0	A,0	A,0
	B	B,1	F,1	L,1	B,1
	C	C,0	M,0	F,0	C,0
{D,N}	D	D,1	D,1	H,1	D,1
	E	A,0	E,0	F,0	-, -
{F,G}	F	B,1	F,1	F,0	C,0
	H	-, -	F,1	H,1	D,1
	L	-, -	E, -	L,1	B,1
	M	C,0	M,0	H, -	-, -

situazioni di equilibrio

Diagramma delle adiacenze e mappa di codifica

A	↔	E	←	L
		↓		↑
C	↔	F	↔	B
↑		↑		
M	⇒	H	↔	D

y ₁ y ₂ /y ₃ y ₄	00	01	11	10
00	A	E	L	-
01	C	F	B	-
11	M	H	D	-
10	-	-	-	-

Tabella delle transizioni

$y_1y_2y_3y_4/XE$	00	01	11	10
0000	0000,0	0001,0	0000,0	0000,0
0001	0000,0	0001,0	0101,0	----,-
0011	----,-	0001,-	0011,1	0111,1
0010	----,-	----,-	----,-	----,-
0100	0100,0	1100,0	0101,0	0100,0
0101	0111,1	0101,1	0101,0	0100,0
0111	0111,1	0101,1	0011,1	0111,1
0110	----,-	----,-	----,-	----,-
1100	0100,0	1100,0	1101,-	----,-
1101	----,-	0101,1	1101,1	1111,1
1111	1111,1	1111,1	1101,1	1111,1
1110	----,-	----,-	----,-	----,-
1000	----,-	----,-	----,-	----,-
1001	----,-	----,-	----,-	----,-
1011	----,-	----,-	----,-	----,-
1010	----,-	----,-	----,-	----,-

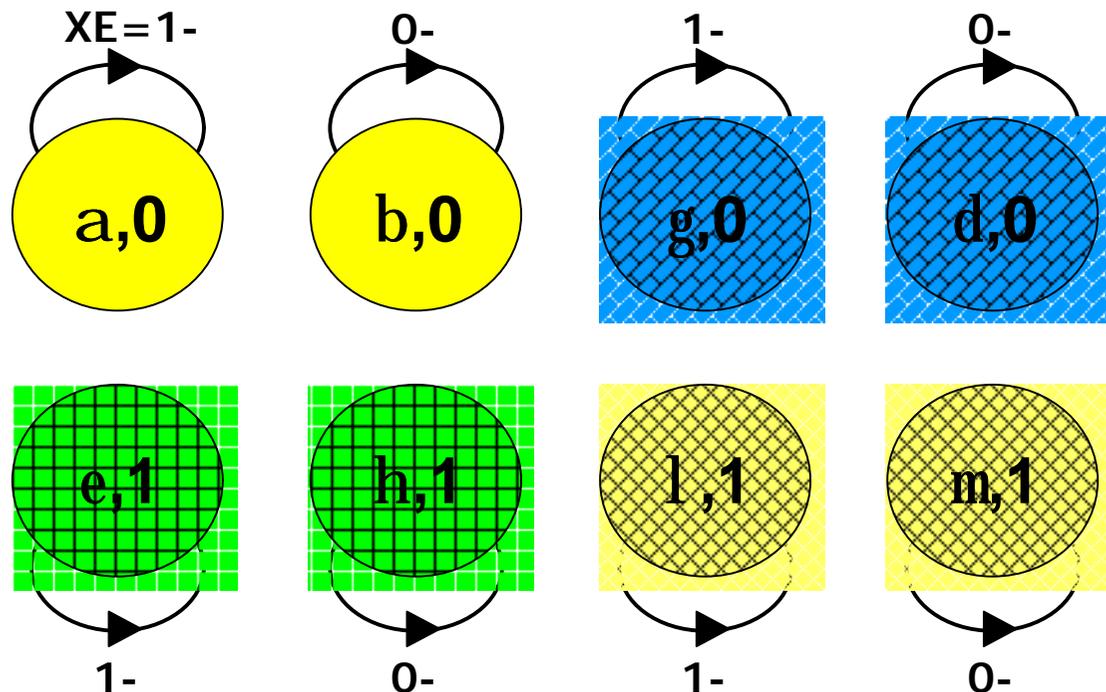
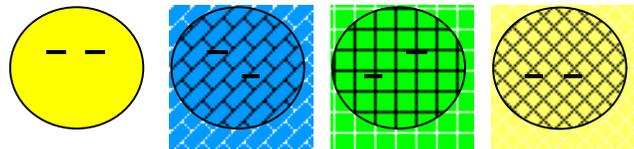
$Y_1Y_2Y_3Y_4, Z$

4 variabili di stato

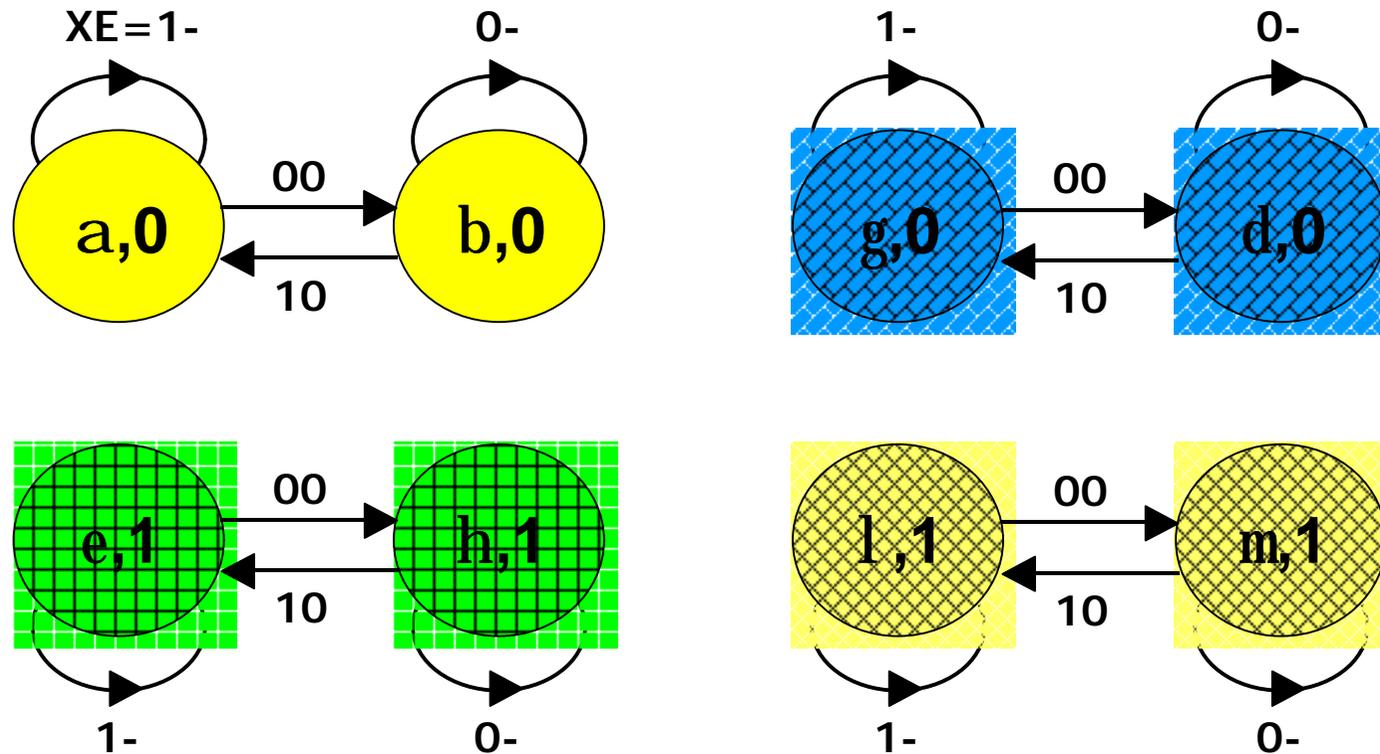
Un secondo diagramma degli stati non primitivo (modello di Moore)

Il diagramma può essere costruito in maniera alternativa partendo dalla constatazione che il valore di E è irrilevante se X non varia. Definiamo pertanto, per ognuna delle quattro combinazioni degli ultimi due fronti significativi del segnale X, una coppia di stati, l'uno stabile per $XE=1-$, l'altro per $XE=0-$. Il penultimo fronte significativo di X associato a ciascuno stato ne individua univocamente il relativo valore di Z.

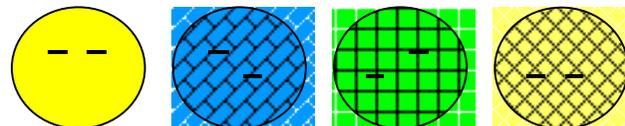
Penultimo e ultimo fronte significativo di X:



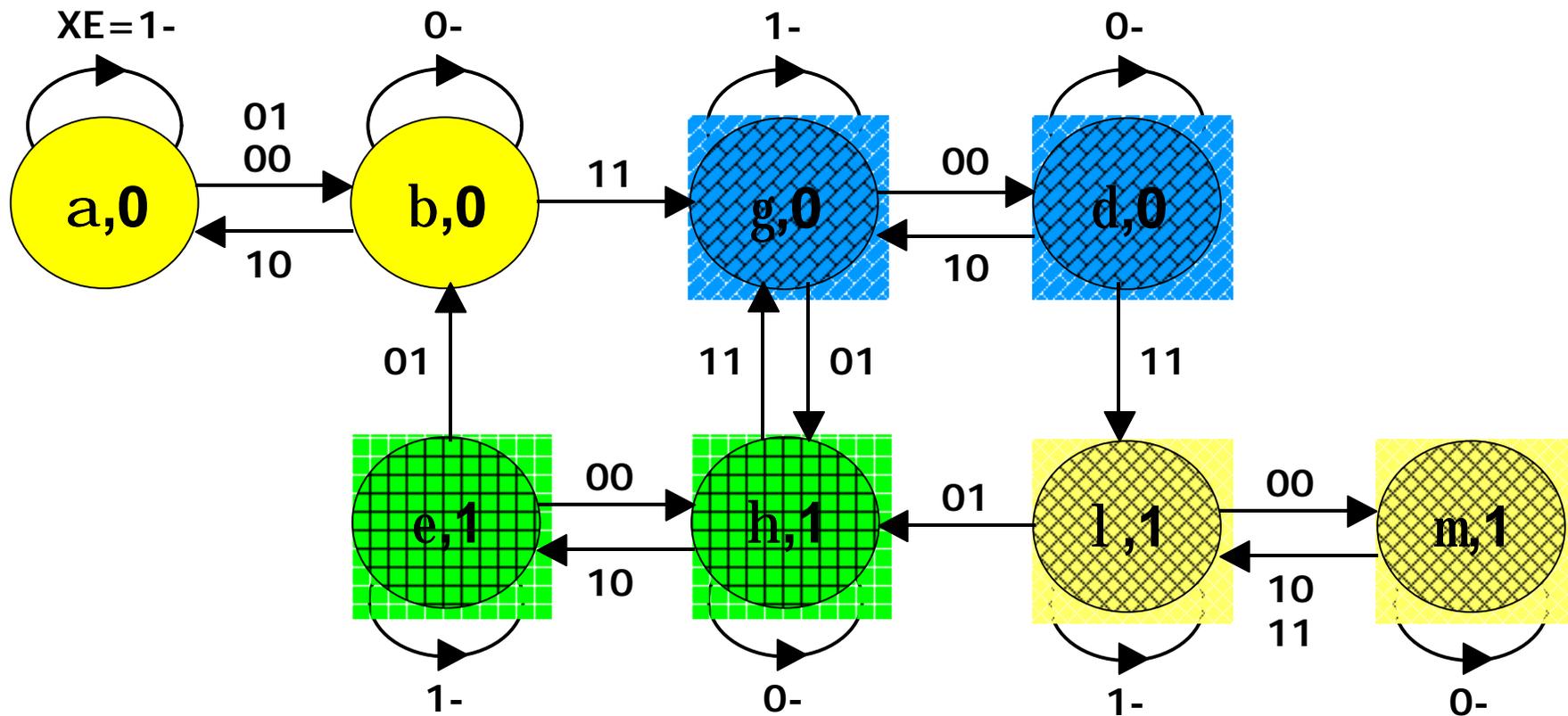
Variazioni di X in presenza del valore 0 di E sono da intendersi non significative. Gli ultimi due fronti significativi di X rimangono quelli precedentemente assunti.



Penultimo e ultimo fronte significativo di X:



Il diagramma viene completato considerando infine variazioni di X in presenza del valore 1 di E, ed indicando coerentemente le transizioni di stato in base all'attuale ed al precedente fronte significativo di X.



Penultimo e ultimo fronte significativo di X:

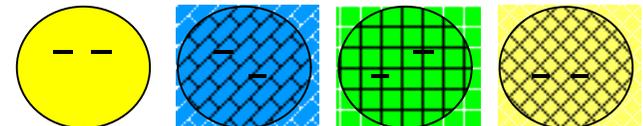


Tabella di flusso (modello di Mealy)

		X E			
		00	01	11	10
situazioni di equilibrio	α	$\beta, 0$	$\beta, 0$	$\alpha, 0$	$\alpha, 0$
	β	$\beta, 0$	$\beta, 0$	$\gamma, 0$	$\alpha, 0$
	γ	$\delta, 0$	$\eta, -$	$\gamma, 0$	$\gamma, 0$
	δ	$\delta, 0$	$\delta, 0$	$\lambda, -$	$\gamma, 0$
	ε	$\eta, 1$	$\beta, -$	$\varepsilon, 1$	$\varepsilon, 1$
	η	$\eta, 1$	$\eta, 1$	$\gamma, -$	$\varepsilon, 1$
	λ	$\mu, 1$	$\eta, 1$	$\lambda, 1$	$\lambda, 1$
	μ	$\mu, 1$	$\mu, 1$	$\lambda, 1$	$\lambda, 1$

tabella non
riducibile

8 (anziché 9)
stati

Diagramma delle adiacenze e mappa di codifica

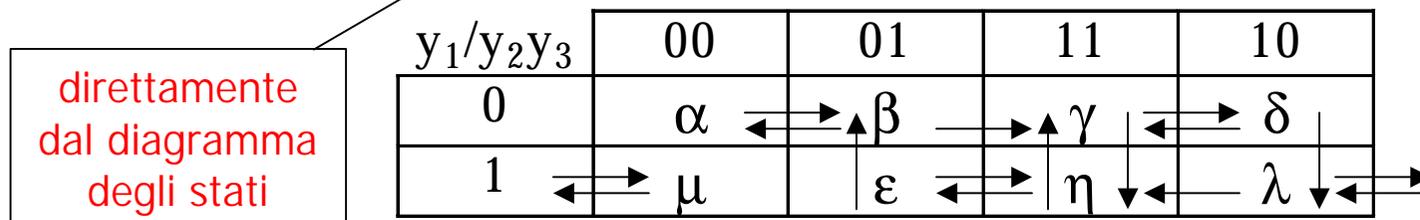


Tabella delle transizioni

		X E			
		00	01	11	10
$y_1y_2y_3$	000	001,0	001,0	000,0	000,0
	001	001,0	001,0	011,0	000,0
	011	010,0	111,-	011,0	011,0
	010	010,0	010,0	110,-	011,0
	100	100,1	100,1	110,1	110,1
	101	111,1	001,-	101,1	101,1
	111	111,1	111,1	011,-	101,1
	110	100,1	111,1	110,1	110,1

$Y_1Y_2Y_3, Z$

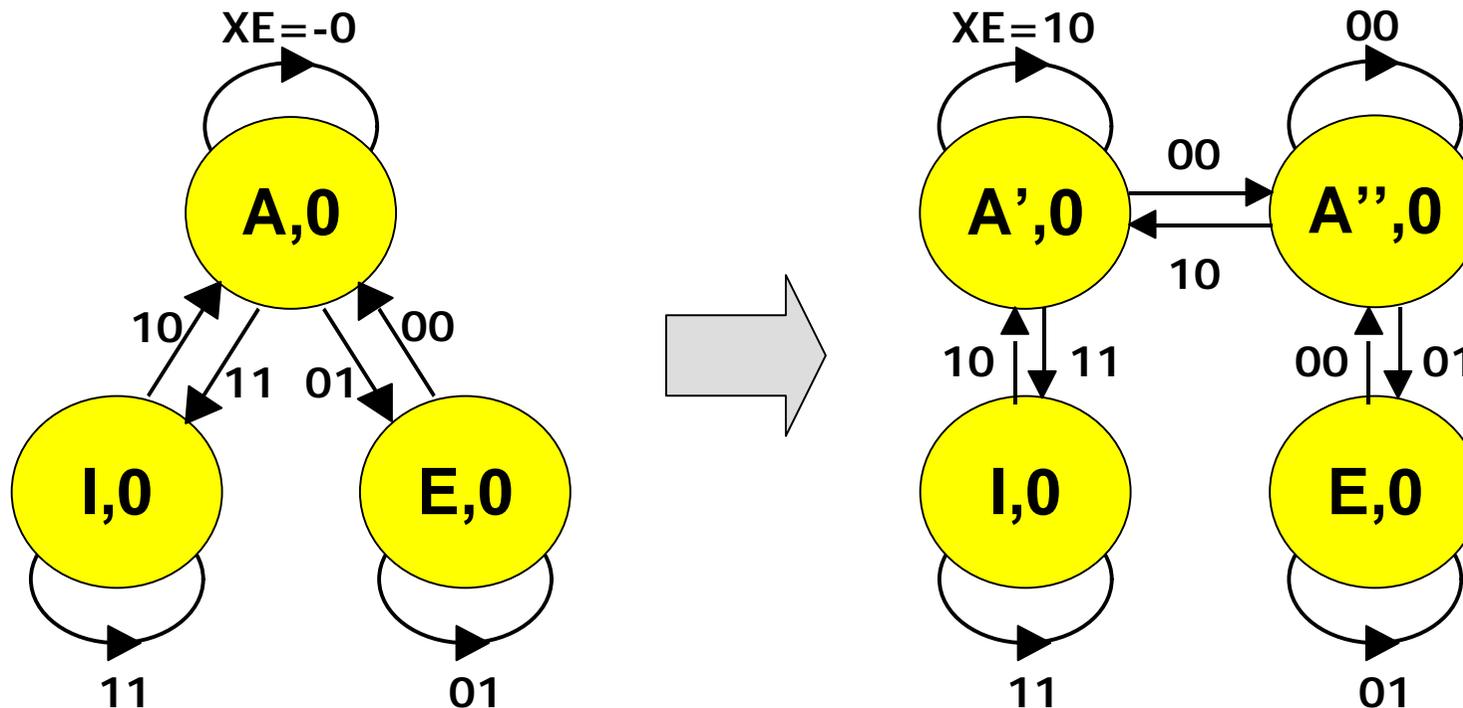
3 (anziché 4) variabili di stato

La costruzione del diagramma degli stati in una forma non primitiva non necessariamente conduce all'identificazione dell'automa minimo.

La costruzione del diagramma degli stati in forma primitiva necessariamente conduce, una volta applicato l'algoritmo di riduzione, all'identificazione dell'automa minimo.

Il diagramma degli stati primitivo (modello di Moore)

Il diagramma primitivo è alquanto simile al primo diagramma non primitivo. Occorre soltanto *sdoppiare* i quattro stati A, B, C e D, stabili per le due configurazioni d'ingresso XE=10 e 00, ciascuno in una coppia di stati, l'uno stabile per XE=10, l'altro per XE=00.



L'applicazione dell'algoritmo di riduzione conduce immediatamente alla tabella di flusso minima, coincidente con quella associata al secondo diagramma non primitivo.

Rete combinatoria di costo minimo mediante NOR

Rete combinatoria di uscita

$Z = y_1$
direttamente dalla mappa
di codifica degli stati

Rete combinatoria di aggiornamento dello stato

$$\begin{aligned}
 Y_1 &= (y_2 + y_3' + X + E') \\
 &(y_2' + y_3' + X' + E') \\
 &(y_1 + y_3 + X) \\
 &(y_1 + E) \\
 &(y_1 + y_2) \\
 &(y_1 + y_3' + X')
 \end{aligned}$$

	X E			
	00	01	11	10
000	0	0	0	0
001	0	0	0	0
011	0	1	0	0
010	0	0	1	0
100	1	1	1	1
101	1	0	1	1
111	1	1	0	1
110	1	1	1	1

Y_1

$$\begin{aligned}
 Y_1 &= (y_2^- y_3'^- X^- E')^- \\
 &(y_2'^- y_3'^- X'^- E')^- \\
 &(y_1^- y_3^- X)^- \\
 &(y_1^- E)^- \\
 &(y_1^- y_2)^- \\
 &(y_1^- y_3'^- X')^-
 \end{aligned}$$

per evitare alee statiche

$$Y_2 = (y_1' + y_3 + X + E)$$

$$(y_1' + y_3' + X' + E)$$

$$(y_1 + y_2 + y_3)$$

$$(y_1 + y_2 + E)$$

$$(y_2 + X + E')$$

$$(y_1' + y_2 + y_3' + X')$$

$$(y_1' + y_2 + y_3' + E')$$

$$(y_2 + y_3 + X)$$

$$(y_1 + y_2 + X)$$

$$(y_2 + y_3' + X' + E)$$

	X E			
	00	01	11	10
000	0	0	0	0
001	0	0	1	0
011	1	1	1	1
010	1	1	1	1
100	0	0	1	1
101	1	0	0	0
111	1	1	1	0
110	0	1	1	1

Y_2

	X E			
	00	01	11	10
000	0	0	0	0
001	0	0	1	0
011	1	1	1	1
010	1	1	1	1
100	0	0	1	1
101	1	0	0	0
111	1	1	1	0
110	0	1	1	1

Y_2

per evitare alee statiche

la mappa è istanziata due volte per motivi di leggibilità

$$Y_2 = (y_1'^- y_3^- X^- E)^- \\
(y_1'^- y_3'^- X'^- E)^- \\
(y_1^- y_2^- y_3)^- \\
(y_1^- y_2^- E)^- \\
(y_2^- X^- E')^- \\
(y_1'^- y_2^- y_3'^- X')^- \\
(y_1'^- y_2^- y_3'^- E')^- \\
(y_2^- y_3^- X)^- \\
(y_1^- y_2^- X)^- \\
(y_2^- y_3'^- X'^- E)^-$$

$$Y_3 = (y_1 + y_2' + X + E)$$

$$(y_1 + y_2 + X' + E)$$

$$(y_3 + X' + E')$$

$$(y_1 + y_2' + y_3 + X)$$

$$(y_1' + y_2 + y_3)$$

$$(y_1' + y_3 + E)$$

$$(y_1 + y_2' + y_3 + E')$$

$$(y_2 + y_3 + X')$$

$$(y_1' + y_3 + X')$$

$$(y_2' + y_3 + X + E)$$

		X E			
		00	01	11	10
y ₁ y ₂ y ₃	000	1	1	0	0
	001	1	1	1	0
	011	0	1	1	1
	010	0	0	0	1
	100	0	0	0	0
	101	1	1	1	1
	111	1	1	1	1
	110	0	1	0	0

		X E			
		00	01	11	10
y ₁ y ₂ y ₃	000	1	1	0	0
	001	1	1	1	0
	011	0	1	1	1
	010	0	0	0	1
	100	0	0	0	0
	101	1	1	1	1
	111	1	1	1	1
	110	0	1	0	0

per evitare alee statiche

la mappa è istanziata due volte per motivi di leggibilità

$$Y_3 = (y_1^- y_2'^- X^- E)^-$$

$$(y_1^- y_2^- X'^- E)^-$$

$$(y_3^- X'^- E')^-$$

$$(y_1^- y_2'^- y_3^- X)^-$$

$$(y_1'^- y_2^- y_3)^-$$

$$(y_1'^- y_3^- E)^-$$

$$(y_1^- y_2'^- y_3^- E')^-$$

$$(y_2^- y_3^- X')^-$$

$$(y_1'^- y_3^- X')^-$$

$$(y_2'^- y_3^- X^- E)^-$$

Esercizio 3

Una rete sequenziale asincrona è caratterizzata da due segnali di ingresso (X_1 , X_2) e da un segnale di uscita (Z). I segnali di ingresso non variano mai contemporaneamente, né possono essere contemporaneamente attivi (livello logico 1), né può X_2 attivarsi più di una volta fra due successive attivazioni di X_1 . Z deve assumere il valore 0 quando $X_1=0$. Quando $X_1=1$, Z può assumere il valore 1 soltanto una volta ogni *due* attivazioni di X_1 , e ciò se nell'intervallo di tempo precedente delimitato dal penultimo fronte di discesa di X_1 stesso, si è avuto un numero pari di attivazioni di X_2 .

Si determini:

- il diagramma degli stati della rete;
- una possibile tabella delle transizioni minima priva di corse critiche.

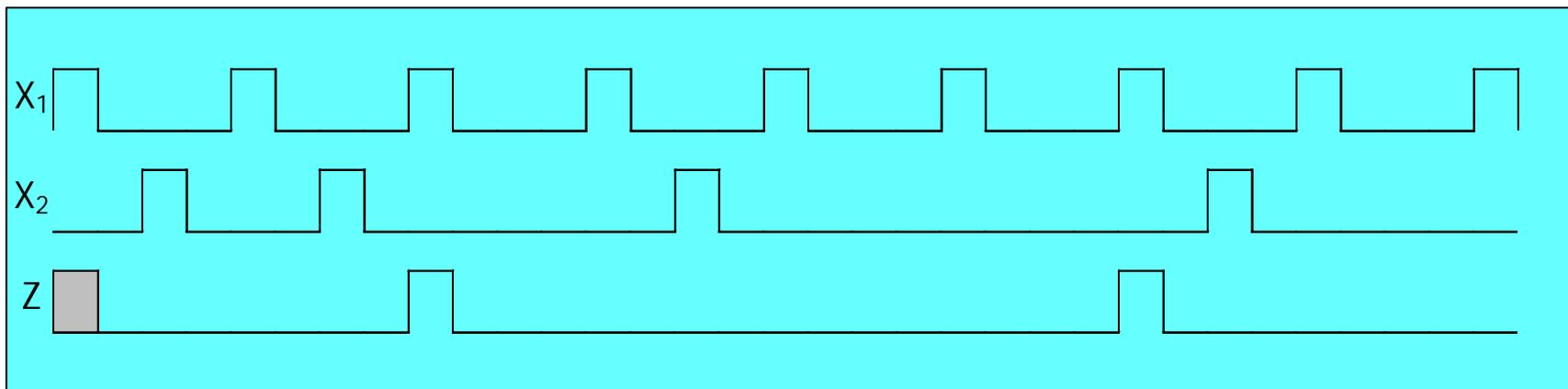
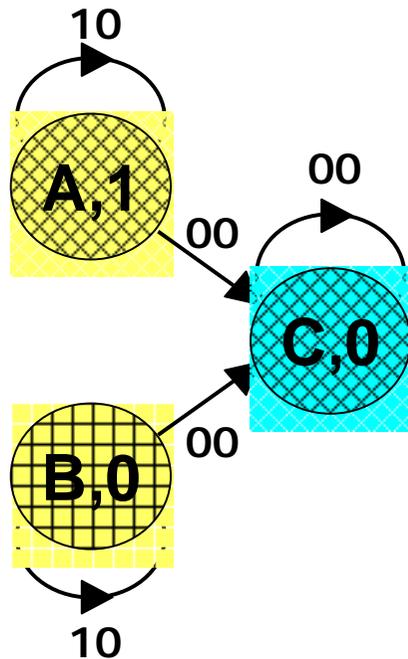


Diagramma degli stati primitivo (modello di Moore)



Il diagramma degli stati può essere costruito considerando inizialmente la configurazione di ingresso $X_1X_2=10$ e supponendo che essa corrisponda alla seconda attivazione del segnale X_1 . Occorre prevedere due distinti stati A e B, entrambi stabili per $X_1X_2=10$ e caratterizzati rispettivamente da $Z=1$ e $Z=0$, onde tener conto del fatto che il numero di attivazioni del segnale X_2 può essere stato in precedenza pari o dispari.

Allorché il segnale X_1 si disattiva, dagli stati A e B si transita nello stato C in attesa della prima attivazione di X_1 , eventualmente preceduta da quella di X_2 .

Numero di attivazioni di X_1 :

0

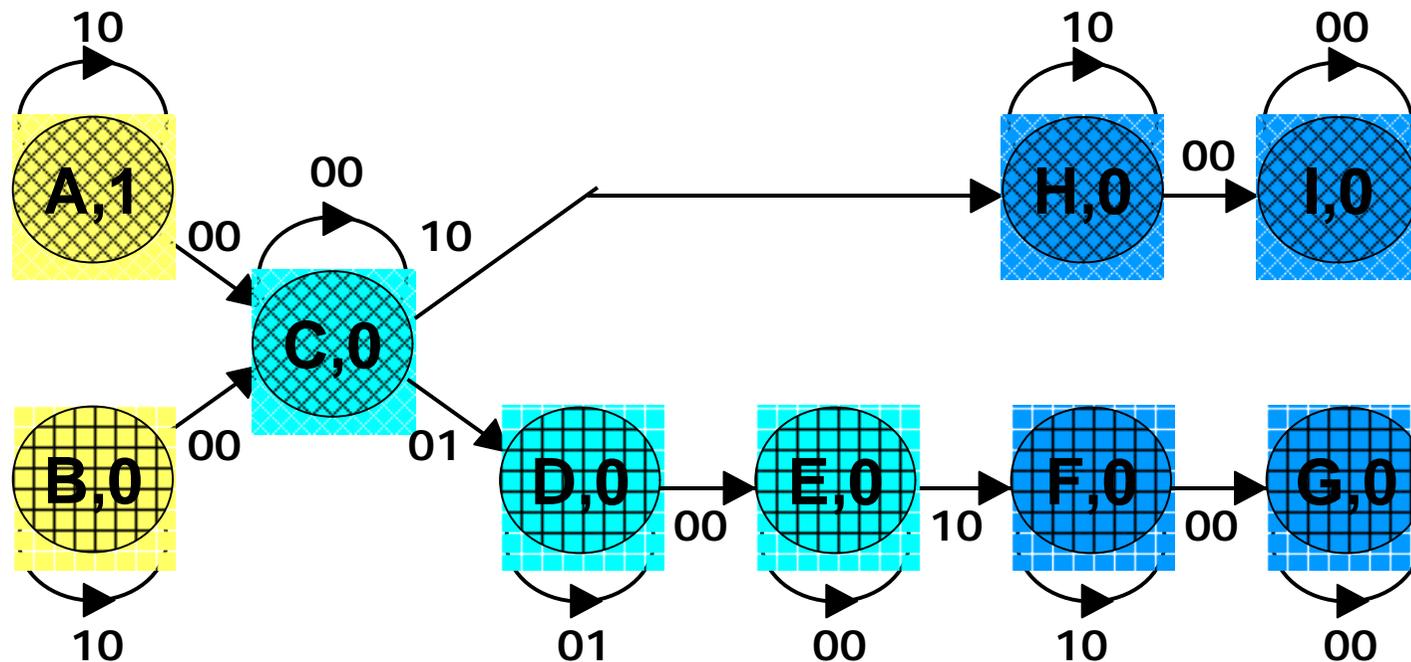
2

Numero di attivazioni di X_2 :

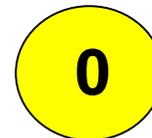
P

D

Lo stato C viene abbandonato non appena si attiva l'uno o l'altro segnale di ingresso. Occorre prevedere due distinte sequenze di stati (D, E, F, G e H, I rispettivamente) per tener traccia del fatto che la prima attivazione di X_1 sia stata o meno preceduta da quella di X_2 .



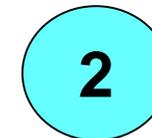
Numero di attivazioni di X_1 :



0

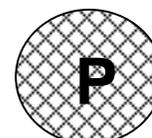


1

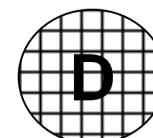


2

Numero di attivazioni di X_2 :

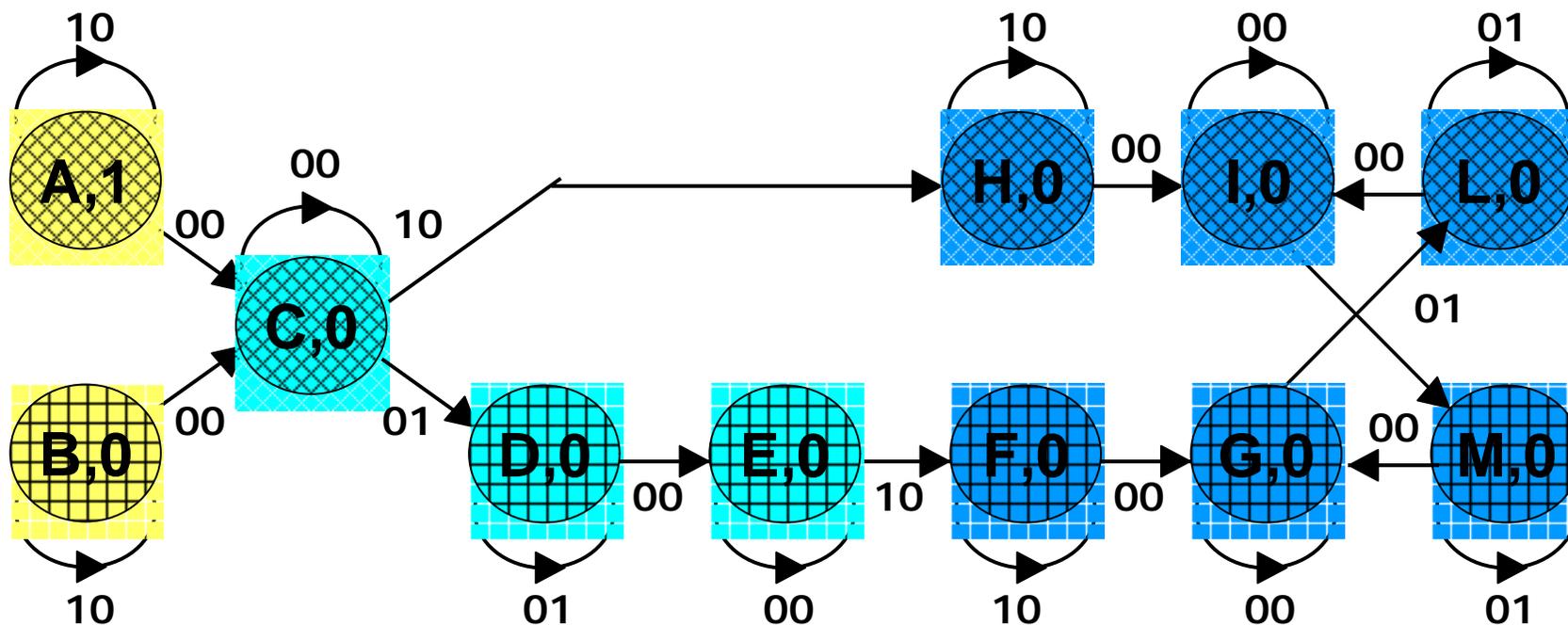


P



D

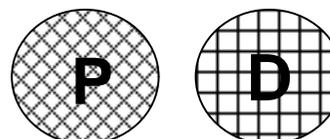
Occorre ora prevedere l'eventuale attivazione di X_2 prima della seconda attivazione di X_1 . Il numero di attivazioni di X_2 risulterà complessivamente pari se in precedenza ne è stata rilevata un'altra (transizione dallo stato G in I via L), dispari in caso contrario (transizione dallo stato I in G via M).



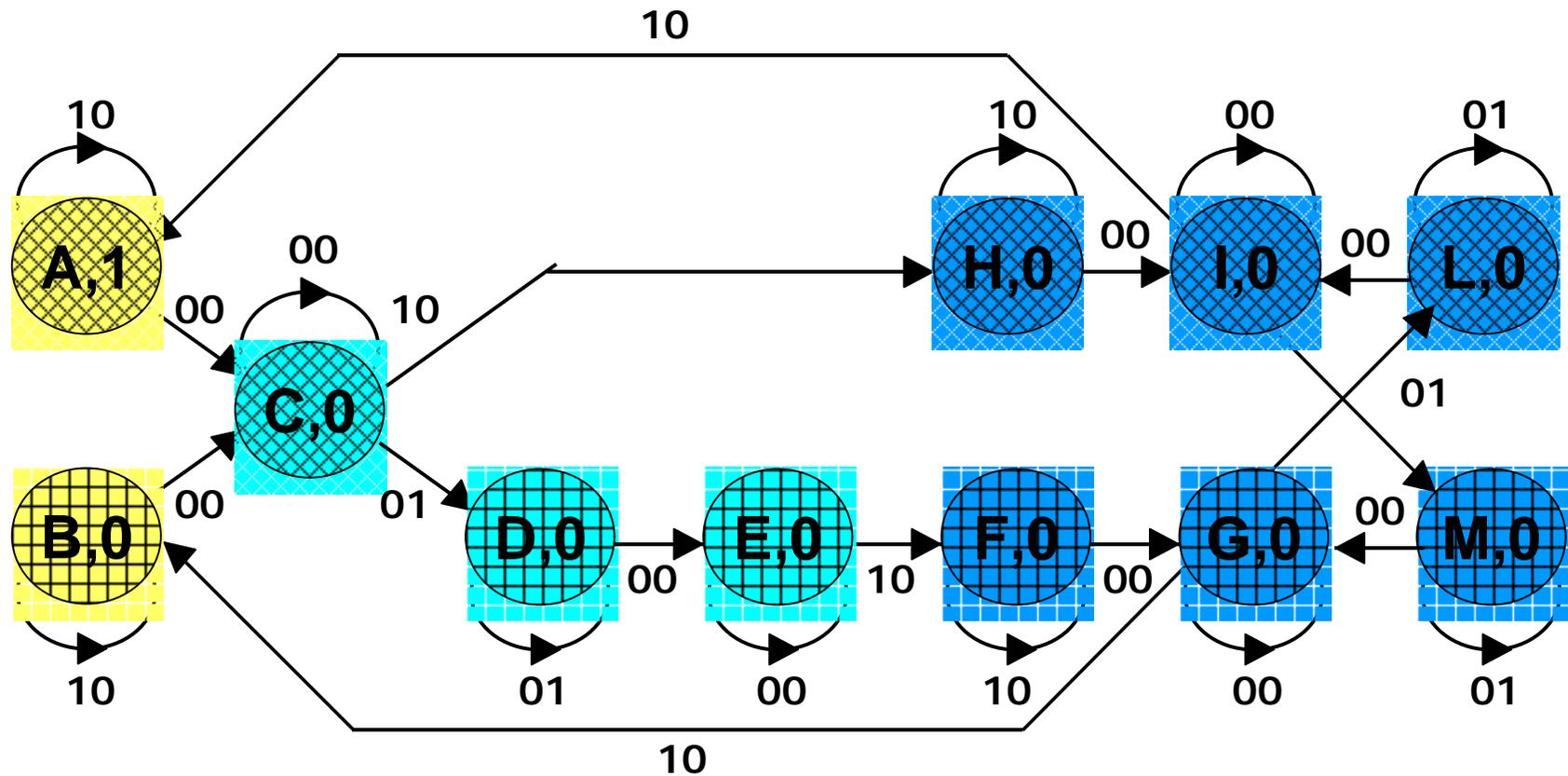
Numero di attivazioni di X_1 :



Numero di attivazioni di X_2 :



Il diagramma viene completato prevedendo la seconda attivazione di X_1 . Dallo stato I (nessuna o due attivazioni di X_2) occorre transitare in A, dallo stato G (una sola attivazione di X_2) in B.



Numero di attivazioni di X_1 :



Numero di attivazioni di X_2 :

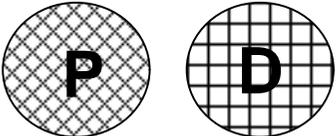


Tabella di flusso (modello di Mealy) e tabella triangolare

	$X_1 X_2$			
	00	01	11	10
A	C,-	-, -	-, -	A,1
B	C,0	-, -	-, -	B,0
C	C,0	D,0	-, -	H,0
D	E,0	D,0	-, -	-, -
E	E,0	-, -	-, -	F,0
F	G,0	-, -	-, -	F,0
G	G,0	L,0	-, -	B,0
H	I,0	-, -	-, -	H,0
I	I,0	M,0	-, -	A,-
L	I,0	L,0	-, -	-, -
M	G,0	M,0	-, -	-, -

situazioni di equilibrio

Classi massime di compatibilità

{A} {B} {C} {D,E} {F,M}
{G} {H,L} {I}

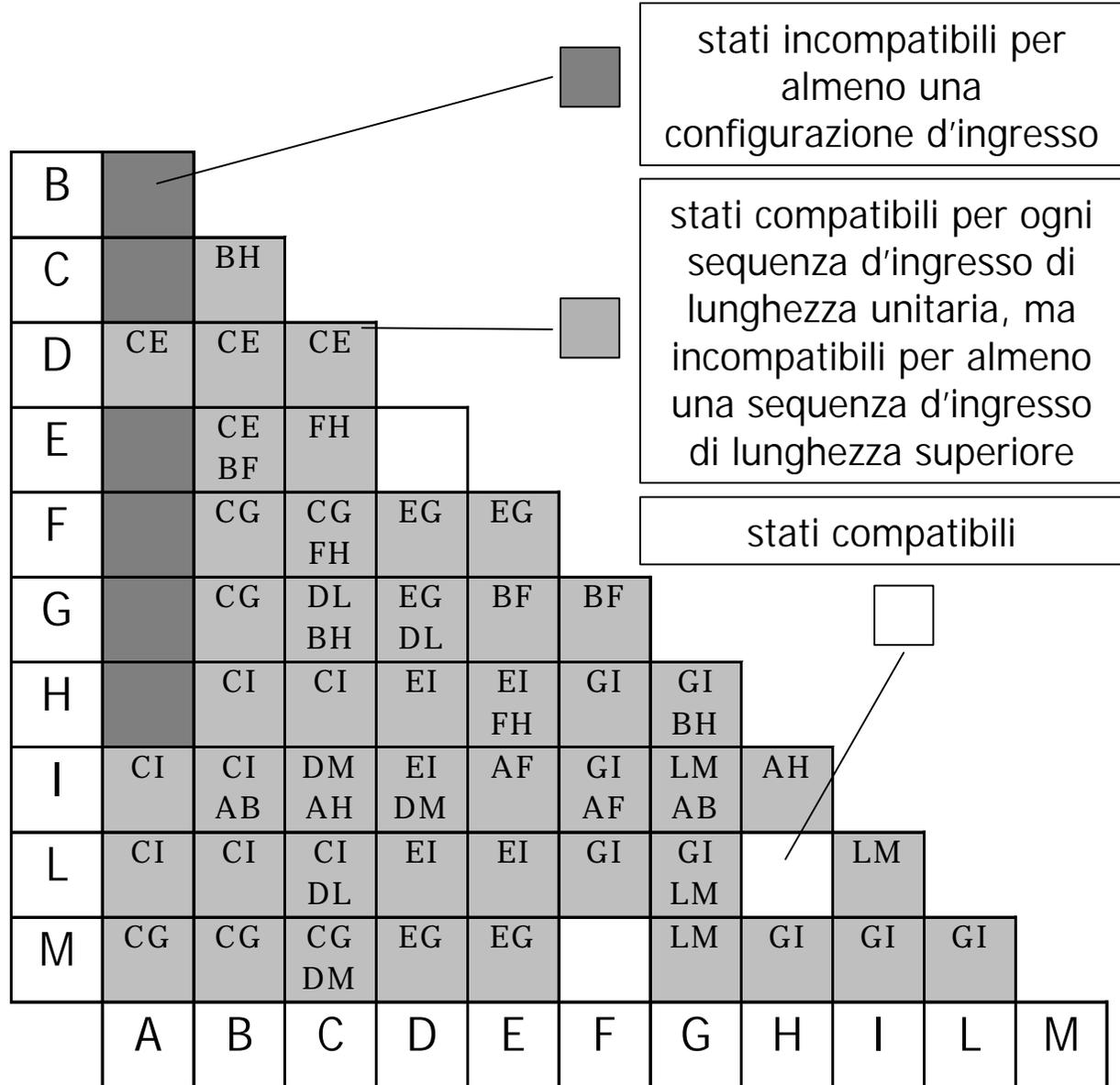


Tabella di flusso minima

		$X_1 X_2$			
		00	01	11	10
{D,E}	A	C,-	-, -	-, -	A,1
	B	C,0	-, -	-, -	B,0
{F,M}	C	C,0	D,0	-, -	H,0
	D	D,0	D,0	-, -	F,0
{H,L}	F	G,0	F,0	-, -	F,0
	G	G,0	H,0	-, -	B,0
	H	I,0	H,0	-, -	H,0
	I	I,0	F,0	-, -	A,-

e mappa di codifica

		$y_2 y_3$			
		00	01	11	10
y_1	0	A	C	B	
	1	G	H	D	F

		$y_2 y_3$			
		00	01	11	10
y_1	0	A	C	B	D
	1	I	H	G	F

Diagramma delle adiacenze

stato

vincoli di adiacenza

transizioni indirette

A	$A \Rightarrow C$	
B	$B \Rightarrow C$	
C	$C \Rightarrow D, C \Rightarrow H$	$C \Rightarrow B \Rightarrow D$
D	$D \Rightarrow F$	
F	$F \Rightarrow G$	
G	$G \Rightarrow H, G \Rightarrow B$	
C	$C \Rightarrow D, C \Rightarrow H$	$C \Rightarrow B \Rightarrow D$
D	$D \Rightarrow F$	
F	$F \Rightarrow G$	
G	$G \Rightarrow H, G \Rightarrow B$	
H	$H \Rightarrow I$	
I	$I \Rightarrow F, I \Rightarrow A$	



Un metodo più sistematico per individuare una codifica degli stati che non dia luogo a corse critiche consiste nell'esaminare i vincoli di adiacenza colonna per colonna, differenziandoli in "vincoli di adiacenza forti" e "vincoli di adiacenza deboli".

Tabella di flusso minima

		$X_1 X_2$			
		00	01	11	10
	A	C,-	-, -	-, -	A,1
	B	C,0	-, -	-, -	B,0
	C	C,0	D,0	-, -	H,0
{D,E}	D	D,0	D,0	-, -	F,0
{F,M}	F	G,0	F,0	-, -	F,0
	G	G,0	H,0	-, -	B,0
{H,L}	H	I,0	H,0	-, -	H,0
	I	I,0	F,0	-, -	A,-

È sufficiente che uno dei due stati A, B sia adiacente a C, non necessariamente entrambi

Esistono due condizioni di indifferenza che possono essere utilizzate per transizioni indirette

vincoli di adiacenza

forti

deboli

$X_1 X_2 = 10$

C \Rightarrow H
D \Rightarrow F
G \Rightarrow B
I \Rightarrow A

$X_1 X_2 = 00$

F \Rightarrow G
H \Rightarrow I

A \Rightarrow C
B \Rightarrow C

$X_1 X_2 = 01$

C \Rightarrow D
G \Rightarrow H
I \Rightarrow F

Tabella di flusso minima

		$X_1 X_2$			
		00	01	11	10
{D,E} {F,M} {H,L}	A	C,-	-, -	-, -	A,1
	B	C,0	-, -	-, -	B,0
	C	C,0	D,0	-, -	H,0
	D	D,0	D,0	-, -	F,0
	F	G,0	F,0	-, -	F,0
	G	G,0	H,0	-, -	B,0
	H	I,0	H,0	-, -	H,0
	I	I,0	F,0	-, -	A,-

Tabella delle transizioni

		$y_1 y_2 y_3 / X_1 X_2$			
		00	01	11	10
A	000	010,-	100,0	---,-	000,1
I	001	001,0	101,0	---,-	000,-
H	011	001,0	011,0	---,-	011,0
C	010	010,0	000,0	---,-	011,0
D	100	100,0	100,0	---,-	101,0
F	101	111,0	101,0	---,-	101,0
G	111	111,0	011,0	---,-	110,0
B	110	010,0	---,-	---,-	110,0

$Y_1 Y_2 Y_3, Z$

Mappa di codifica

vincoli di adiacenza forti

$C \Rightarrow H$ $D \Rightarrow F$ $G \Rightarrow B$
 $I \Rightarrow A$ $F \Rightarrow G$ $H \Rightarrow I$

$y_1/y_2 y_3$	00	01	11	10
0	C \Rightarrow H	\Rightarrow I	\Rightarrow A	
1	D \Rightarrow F	\Rightarrow G	\Rightarrow B	

tutte transizioni dirette

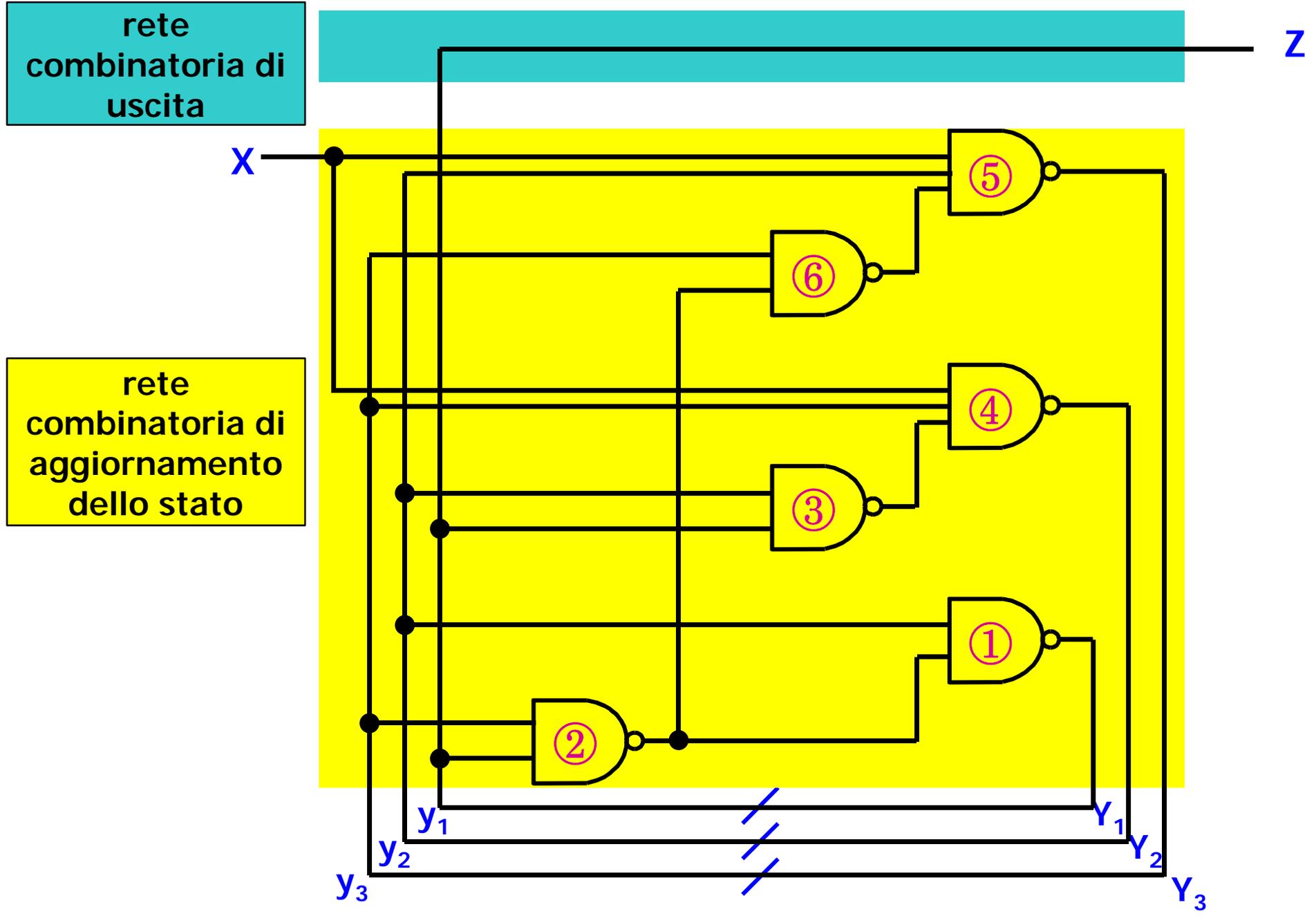
vincoli di adiacenza deboli

$C \Rightarrow D$ $G \Rightarrow H$ $I \Rightarrow F$
 $A \Rightarrow C$ $B \Rightarrow C$

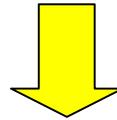
$y_1/y_2 y_3$	00	01	11	10
0	A \Leftarrow I	\Leftarrow H	\Leftarrow C	
1	D \Rightarrow F	\Rightarrow G	\Rightarrow B	

una transizione indiretta $C \Rightarrow A \Rightarrow D$

Lo schema logico ridisegnato secondo il modello di riferimento



Schema logico



Espressioni delle variabili di stato interno futuro e di uscita in funzione delle variabili di stato interno presente e di ingresso

$$Y_1 = y_{2-1} - (y_{1-1} - y_{3-1})$$

$$Y_1 = y_2' + y_1 y_3$$

$$Y_2 = X_{-1} - y_{3-1} - (y_{1-2} - y_{2-2})$$

$$Y_2 = X' + y_3' + y_1 y_2$$

$$Y_3 = X_{-1} - y_{2-1} - (y_{3-2} - (y_{1-3} - y_{3-3}))$$

$$Y_3 = X' + y_2' + y_3 (y_1' + y_3')$$

indice di livello

$$Z = y_1$$

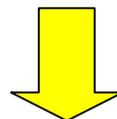


Tabella delle transizioni

Tabella delle transizioni

		X	
		0	1
$y_1y_2y_3$	000	111,0	111,0
	001	111,0	101,0
	011	011,0	001,0
	010	011,0	010,0
	100	111,1	111,1
	101	111,1	101,1
	111	111,1	110,1
	110	011,1	010,1

$Y_1Y_2Y_3,Z$

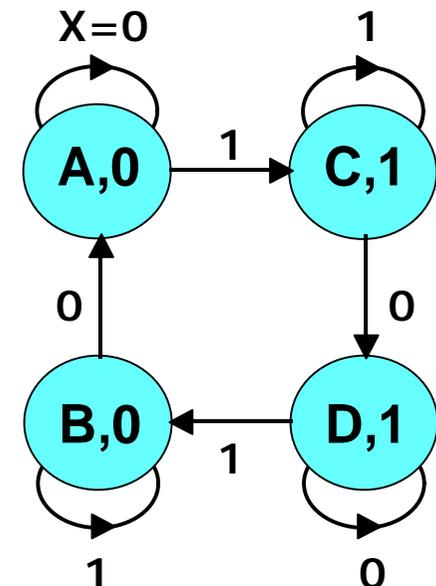
- situazioni di equilibrio
- transizioni dirette
- transizioni multiple (per evitare corse critiche)
- stati irraggiungibili

Tabella di flusso

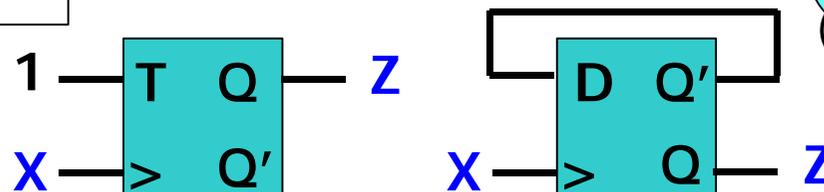
	X	
	0	1
011 \equiv A	A,0	C,-
010 \equiv B	A,0	B,0
101 \equiv C	D,1	C,1
111 \equiv D	D,1	B,-

Il segnale di uscita Z commuta ad ogni fronte di salita del segnale di ingresso X

Diagramma degli stati



Reti equivalenti





**Reti
sequenziali
sincrone**

Esercizio 1

Gli accessi da parte di due processori P_1 e P_2 ad una risorsa condivisa sono coordinati da una rete sequenziale sincrona (*arbitro*), caratterizzata da due segnali di ingresso R_1 e R_2 (richiesta di accesso da parte di P_1 e P_2), e da due segnali di uscita C_1 e C_2 (consenso all'accesso per P_1 e P_2). In assenza di richieste di accesso, l'arbitro deve mantenere disattivi (livello logico 0) sia C_1 che C_2 . Se un processore presenta una richiesta di accesso allorché la risorsa condivisa è libera, l'arbitro deve prontamente attivare il relativo consenso, confermandolo poi per tutto il tempo di attivazione della richiesta stessa. Se un processore presenta una richiesta di accesso allorché la risorsa non è libera (*conflitto di accesso*), l'arbitro deve fornire il relativo consenso soltanto al completamento dell'accesso in corso. Richieste di accesso contemporanee da parte di P_1 e P_2 , infine, debbono essere gestite dall'arbitro in maniera tale da privilegiare il processore che per ultimo ha dovuto attendere a seguito di un conflitto di accesso. Nell'ipotesi che P_1 e P_2 , una volta attivata una richiesta di accesso, rimangano comunque in attesa del relativo consenso confermando la richiesta stessa, si determini:

- l'automa minimo dell'arbitro secondo il modello di Moore;
- una possibile realizzazione mediante FF-JK e gate elementari.

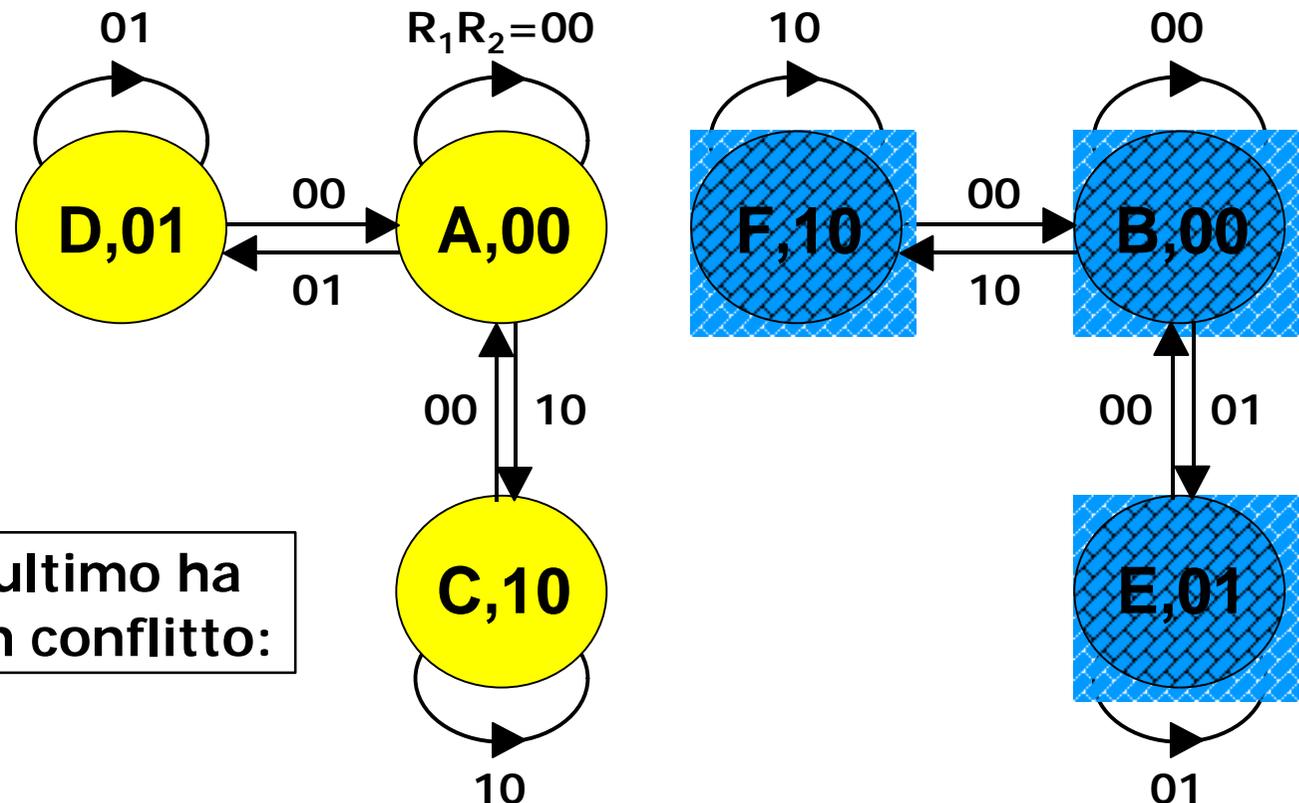
Diagramma degli stati (modello di Moore)

Il diagramma può essere costruito a partire dalla situazione corrispondente a risorsa libera e richieste di accesso disattive ($R_1R_2=00$). Occorre prevedere due distinti stati, entrambi caratterizzati da $C_1C_2=00$, per discriminare il processore che per ultimo ha atteso a seguito di un conflitto.

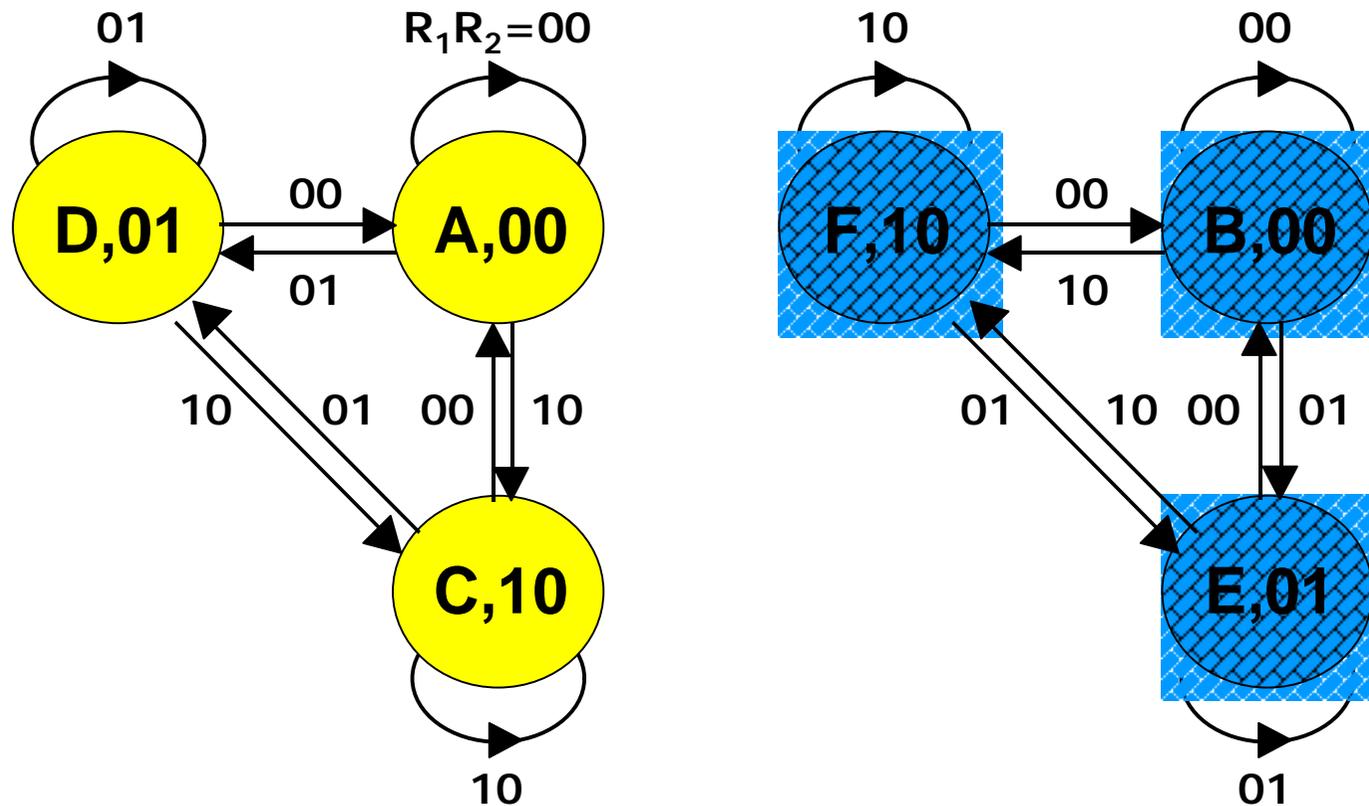
A fronte di una sola richiesta di accesso, da ciascuno stato si transiterà verso una nuova coppia di stati, l'uno caratterizzato da $C_1C_2=10$ (attivazione di R_1), l'altro da $C_1C_2=01$ (attivazione di R_2).

Terminato l'accesso ($R_1R_2=00$), da ogni stato si effettuerà poi la transizione opposta, non essendosi verificato per ipotesi un nuovo conflitto.

processore che per ultimo ha atteso a seguito di un conflitto:



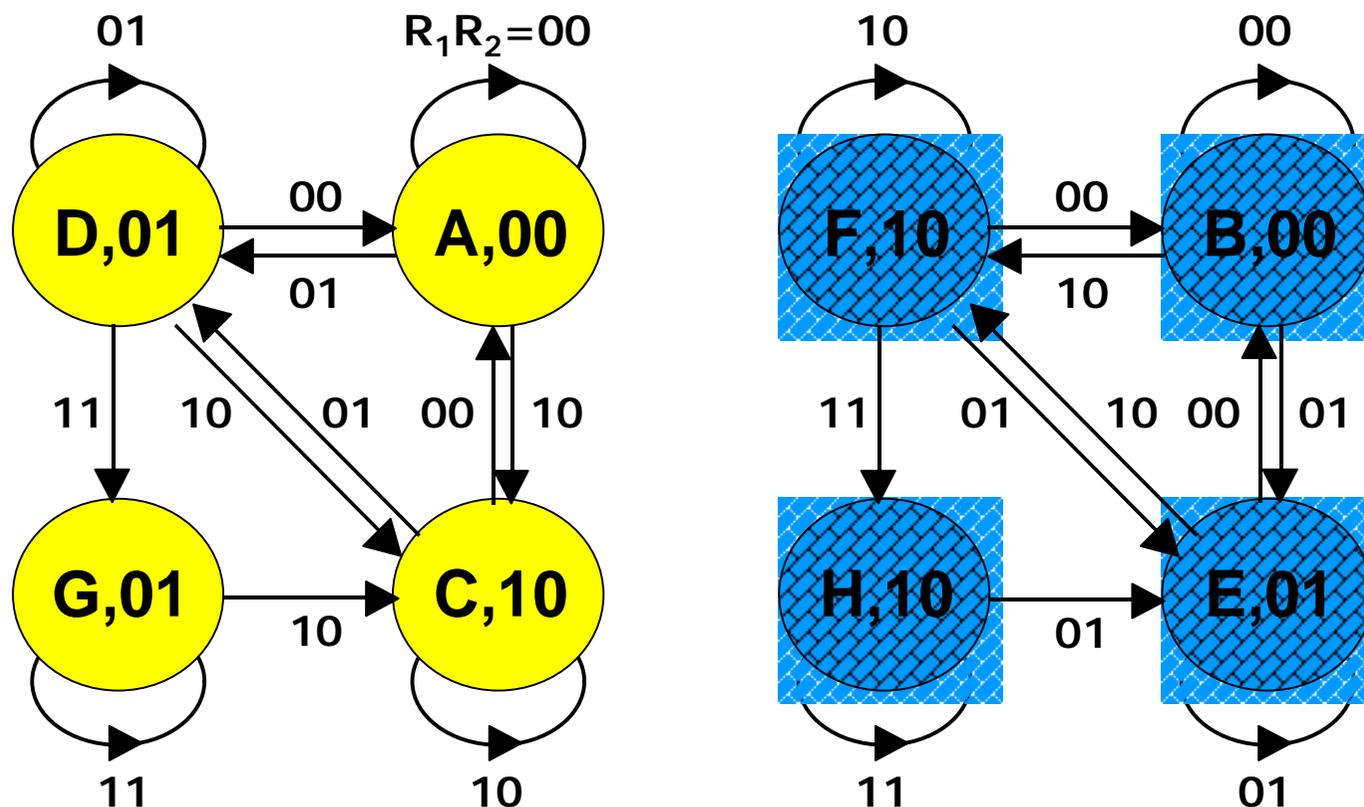
Non si ha conflitto anche nel caso in cui la richiesta di accesso alla risorsa condivisa da parte di un processore si manifesta nello stesso intervallo di clock in cui termina l'accesso da parte dell'altro processore. In tal caso il processore che per ultimo ha dovuto attendere a seguito di un conflitto di accesso continua ad essere quello precedentemente assunto.



processore che per ultimo ha dovuto attendere a seguito di un conflitto di accesso:



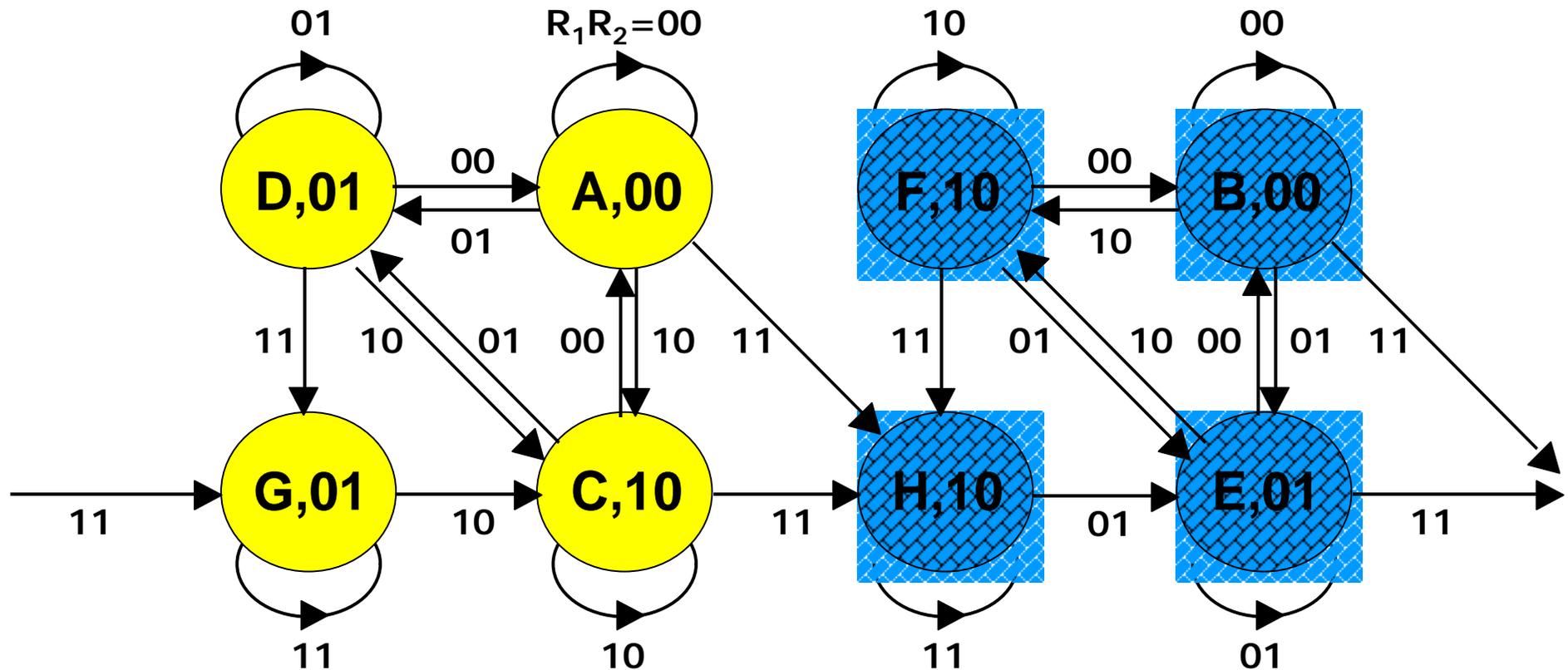
Nel caso in cui la risorsa sia già occupata e si manifesti una richiesta di accesso da parte dello stesso processore che in precedenza ha dovuto attendere a seguito di un conflitto, occorre ritardare l'attivazione del relativo consenso fino al completamento dell'accesso in corso, e confermare tale processore come l'ultimo cui è stato temporaneamente negato l'accesso.



processore che per ultimo ha dovuto attendere a seguito di un conflitto di accesso:



Similmente occorre operare nel caso in cui la risorsa sia già occupata e la richiesta di accesso pervenga da parte dell'altro processore. L'unica variante consiste nell'aggiornare lo stato in modo tale da riflettere quest'ultimo conflitto. Occorre infine prevedere le due transizioni di stato corrispondenti alla contemporanea attivazione di R_1 e R_2 allorché la risorsa è libera. La precedenza nell'accesso deve essere attribuita al processore penalizzato nell'ultimo conflitto.



processore che per ultimo ha dovuto attendere a seguito di un conflitto di accesso:



Tabella di flusso e tabella triangolare

	R_1R_2				
	00	01	11	10	C_1C_2
A	A	D	H	C	00
B	B	E	G	F	00
C	A	D	H	C	10
D	A	D	G	C	01
E	B	E	G	F	01
F	B	E	H	F	10
G	-	-	G	C	01
H	-	E	H	-	10

stati incompatibili per almeno una configurazione d'ingresso

stati compatibili per ogni sequenza d'ingresso di lunghezza unitaria, ma incompatibili per almeno una sequenza d'ingresso di lunghezza superiore

stati compatibili

Classi massime di compatibilità

{A}	{B}	{C}
{D,G}	{E}	{F,H}

B	DE	GH	CF				
C							
D							
E				AB	CF		
F			AB	DE			
G					CF		
H			DE				
	A	B	C	D	E	F	G

Tabella di flusso minima

Mappa di codifica

		R_1R_2				
		00	01	11	10	C_1C_2
{DG}≡ {FH}≡	A	A	D	F	C	00
	B	B	E	D	F	00
	C	A	D	F	C	10
	D	A	D	D	C	01
	E	B	E	D	F	01
	F	B	E	F	F	10

y_0/y_1y_2	00	01	11	10
0	A	D	-	C
1	B	E	-	F

$y_0 =$ 0 1

priorità di accesso:



Tabella delle transizioni

		$(R_1R_2)^n$				
		00	01	11	10	$(C_1C_2)^n$
$(y_0y_1y_2)^n$	000	000	001	110	010	00
	001	000	001	001	010	01
	011	-	-	-	-	-
	010	000	001	110	010	10
	100	100	101	001	110	00
	101	100	101	001	110	01
	111	-	-	-	-	-
	110	100	101	110	110	10

$(y_0y_1y_2)^{n+1}$

Reti combinatorie di uscita e di aggiornamento dello stato

$$C_1^n = y_1^n \quad C_2^n = y_2^n$$

		$(R_1R_2)^n$				$(R_1R_2)^n$				$(R_1R_2)^n$			
		00	01	11	10	00	01	11	10	00	01	11	10
$(y_0y_1y_2)^n$	000	0	0	1	0	0	0	1	1	0	1	0	0
	001	0	0	0	0	0	0	0	1	0	1	1	0
	011	-	-	-	-	-	-	-	-	-	-	-	-
	010	0	0	1	0	0	0	1	1	0	1	0	0
	100	1	1	0	1	0	0	0	1	0	1	1	0
	101	1	1	0	1	0	0	0	1	0	1	1	0
	111	-	-	-	-	-	-	-	-	-	-	-	-
	110	1	1	1	1	0	0	1	1	0	1	0	0
		y_0^{n+1}				y_1^{n+1}				y_2^{n+1}			

$$J_0^n = (R_1R_2y_2')^n$$

$$J_1^n = (R_1R_2' + R_1y_2'y_0')^n$$

$$J_2^n = (R_2R_1' + R_2y_1'y_0')^n$$

$$K_0^n = (R_1R_2y_1')^n$$

$$K_1^n = (R_1')^n$$

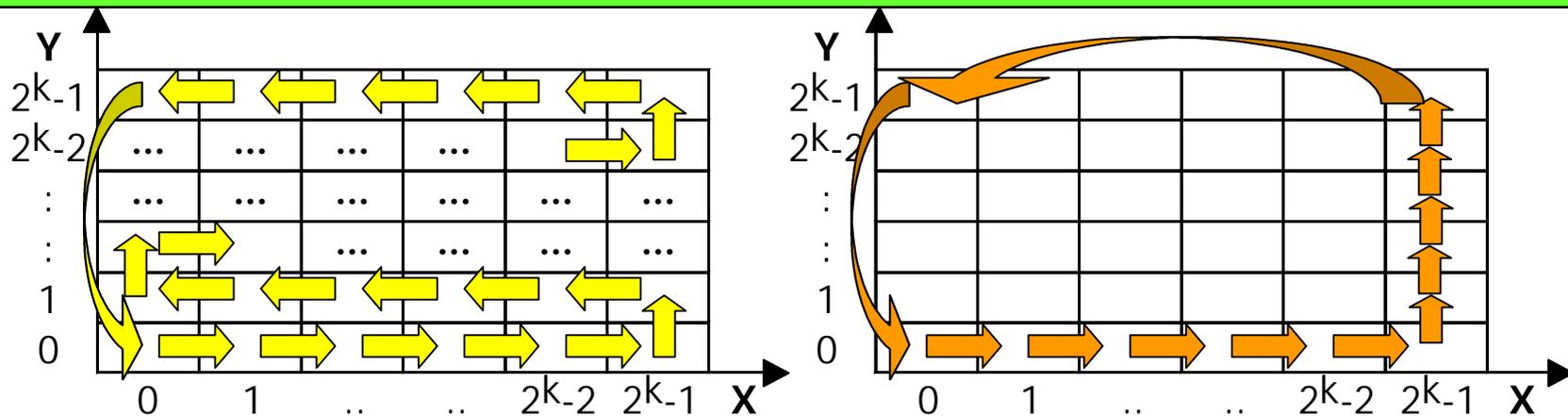
$$K_2^n = (R_2')^n$$

Esercizio 2

La posizione di un oggetto nel piano X, Y è definita dalle uscite di 2 contatori binari avanti-indietro (C_x, C_y) aventi base di conteggio 2^k : le uscite x_{k-1}, \dots, x_1, x_0 di C_x stabiliscono il valore della coordinata X , le uscite y_{k-1}, \dots, y_1, y_0 di C_y il valore della coordinata Y . Una rete sequenziale sincrona operante in base allo stesso clock dei contatori ha il compito di gestire i segnali E_x, E_y (secondo il modello di Mealy) e $U_x/D_x', U_y/D_y'$ (secondo il modello di Moore) dei contatori, in modo tale che l'oggetto descriva nel piano X, Y le due traiettorie indicate in figura, una di seguito all'altra e senza soluzione di continuità. Si determini l'automa minimo della rete ed una possibile realizzazione tramite FF-T e gate elementari, nell'ipotesi che siano disponibili in ingresso unicamente i segnali

$$Z_x = x_{k-1} \dots x_1 x_0 U_x/D_x' + x_{k-1}' \dots x_1' x_0' (U_x/D_x')',$$

$$Z_y = y_{k-1} \dots y_1 y_0 U_y/D_y' + y_{k-1}' \dots y_1' y_0' (U_y/D_y')'.$$



In effetti occorre progettare una rete a due sole uscite ($E_x, U_x/D_x'$), in quanto si può direttamente imporre $U_y/D_y'=1$, dovendo il contatore C_y contare soltanto in avanti, ed assumere $E_y = E_x'$, non essendo mai i contatori contemporaneamente abilitati o disabilitati.

Evoluzione temporale dei segnali di ingresso e di uscita ($k=2$)

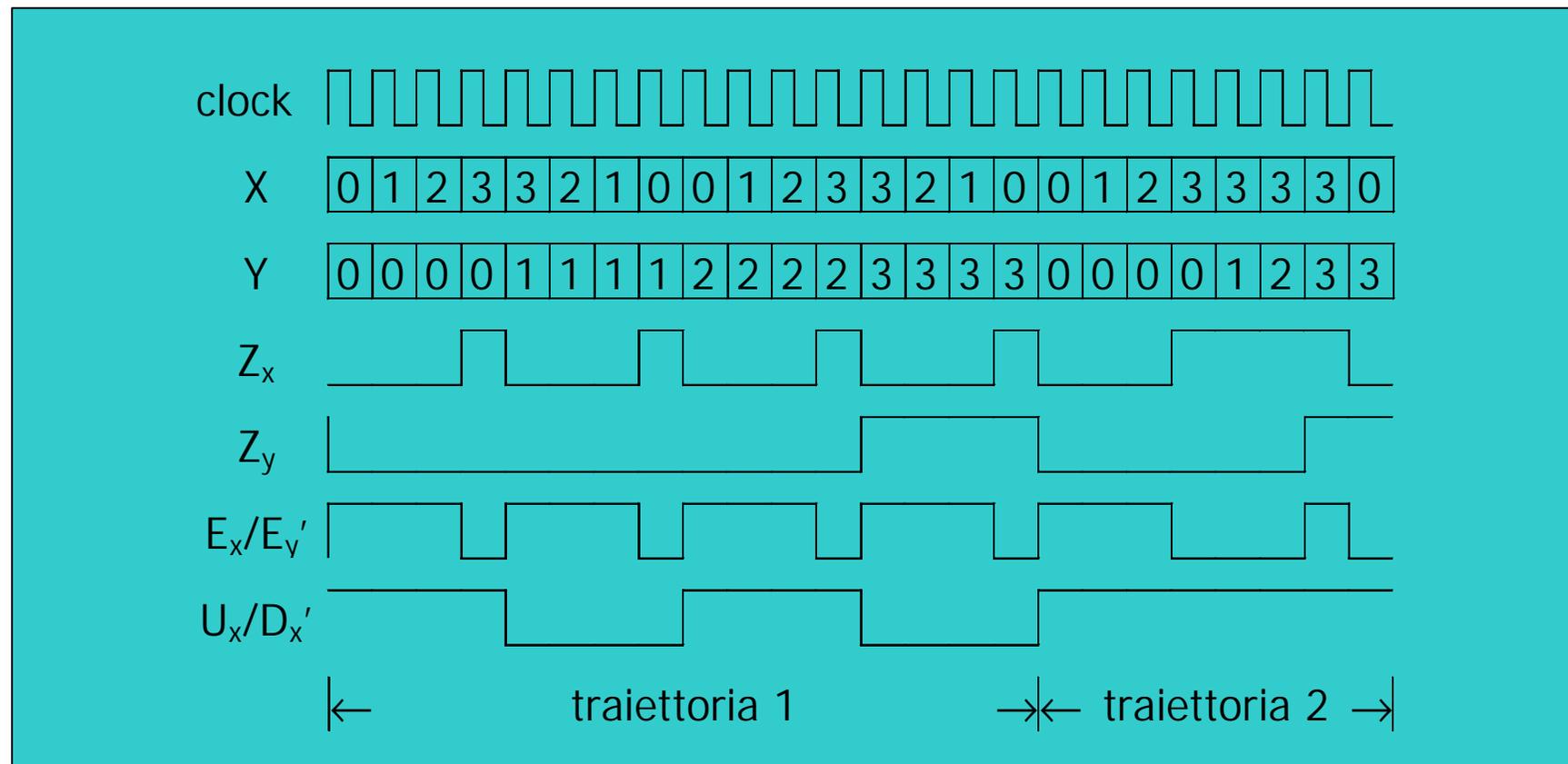
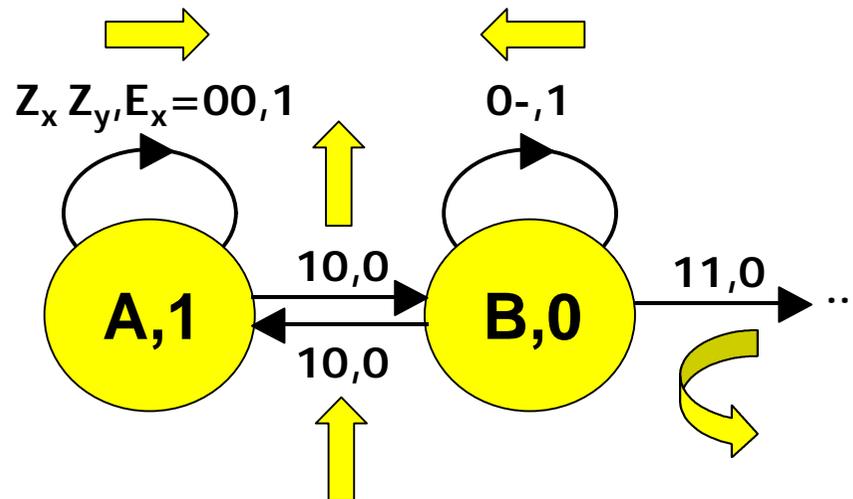


Diagramma degli stati

Il diagramma può essere costruito ipotizzando inizialmente che l'oggetto si trovi nell'origine del piano X, Y e che debba descrivere la prima traiettoria. A tal fine occorre eseguire nell'ordine i seguenti passi:

- abilitare il contatore C_x in avanti fino a che $X=X_{\max}$;
- abilitare il contatore C_y per un intervallo di clock cosicché $Y=Y+1$;
- abilitare il contatore C_x all'indietro, fino a che $X=X_{\min}$;
- incrementare il contatore C_y e ripetere i passi precedenti se $Y < Y_{\max}$



Traiettoria:

1

Affinché l'oggetto descriva nel piano X, Y la seconda traiettoria, occorre selezionare la modalità di conteggio in avanti anche per il contatore C_x ed eseguire ordinatamente i seguenti passi:

- abilitare il contatore C_x fino a che $X=X_{max}$;
- abilitare il contatore C_y fino a che $Y=Y_{max}$;
- incrementare il contatore C_x cosicché $X=X_{min}$;
- incrementare il contatore C_y cosicché $Y=Y_{min}$ (ritorno nell'origine).

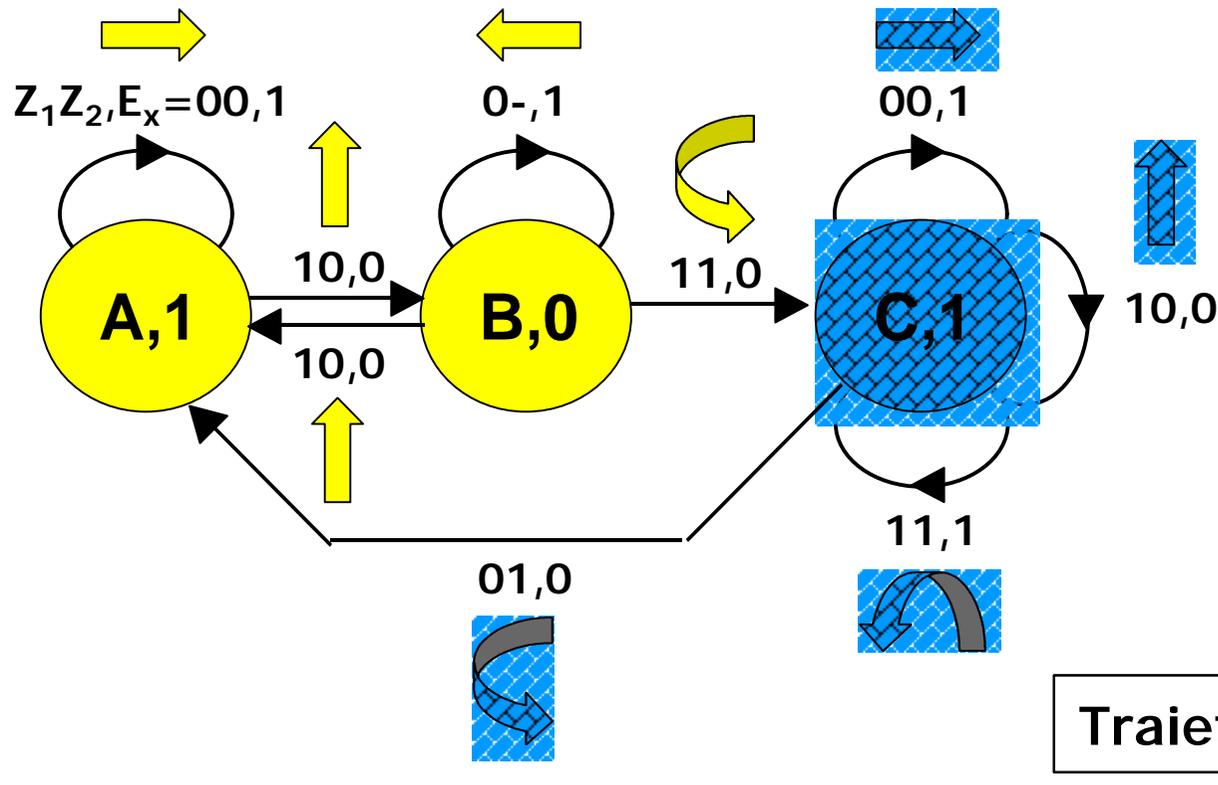


Tabella di flusso e tabella delle transizioni

		$Z_x Z_y$				
s.p.		00	01	11	10	U_x/D_x'
A	A,1	-, -	-, -	B,0		1
B	B,1	B,1	C,0	A,0		0
C	C,1	A,0	C,1	C,0		1

s.f., E_x

		$(Z_x Z_y)^n$				
$(y_1 y_2)^n$		00	01	11	10	$(U_x/D_x')^n$
$A \equiv 00$	00,1	-, -	-, -	01,0		1
$B \equiv 01$	01,1	01,1	10,0	00,0		0
11	-, -	-, -	-, -	-, -		-
$C \equiv 10$	10,1	00,0	10,1	10,0		1

$(y_1 y_2)^{n+1}, E_x^n$

Reti combinatorie di aggiornamento dello stato e d'uscita

$$T_1^n = (Z_x' Z_y y_1 + Z_x Z_y y_2)^n$$

$$T_2^n = (Z_x y_1')^n$$

$$E_x^n = (Z_x' Z_y' + Z_x' y_1' + Z_x Z_y y_1)^n$$

$$(U_x/D_x')^n = (y_2')^n$$

Esercizio 3

Una rete sequenziale sincrona è caratterizzata da un unico segnale di ingresso (X) e da un unico segnale di uscita (Z), entrambi sincroni. In ogni intervallo di clock l'uscita Z deve assumere il valore logico 1 soltanto se gli ultimi quattro valori di X costituiscono una palindrome. Si identifichi:

- l'automa minimo della rete secondo il modello di Mealy;
- una possibile realizzazione della rete con FF-D e gate elementari.

X	0	1	1	0	1	0	0	1	1	1	1	1	...
Z				1	0	0	0	1	0	0	1	1	...

Diagramma degli stati (modello di Mealy)

Occorre prevedere otto distinti stati, ciascuno dei quali identifica una ben precisa configurazione dei valori assunti da X nei tre precedenti intervalli di clock. Denotando gli stati direttamente con tali configurazioni binarie, ed assumendo che in esse i bit siano ordinati in modo tale che l'ultimo a destra si riferisca all'ultimo valore assunto da X , è immediato costruire il diagramma degli stati:

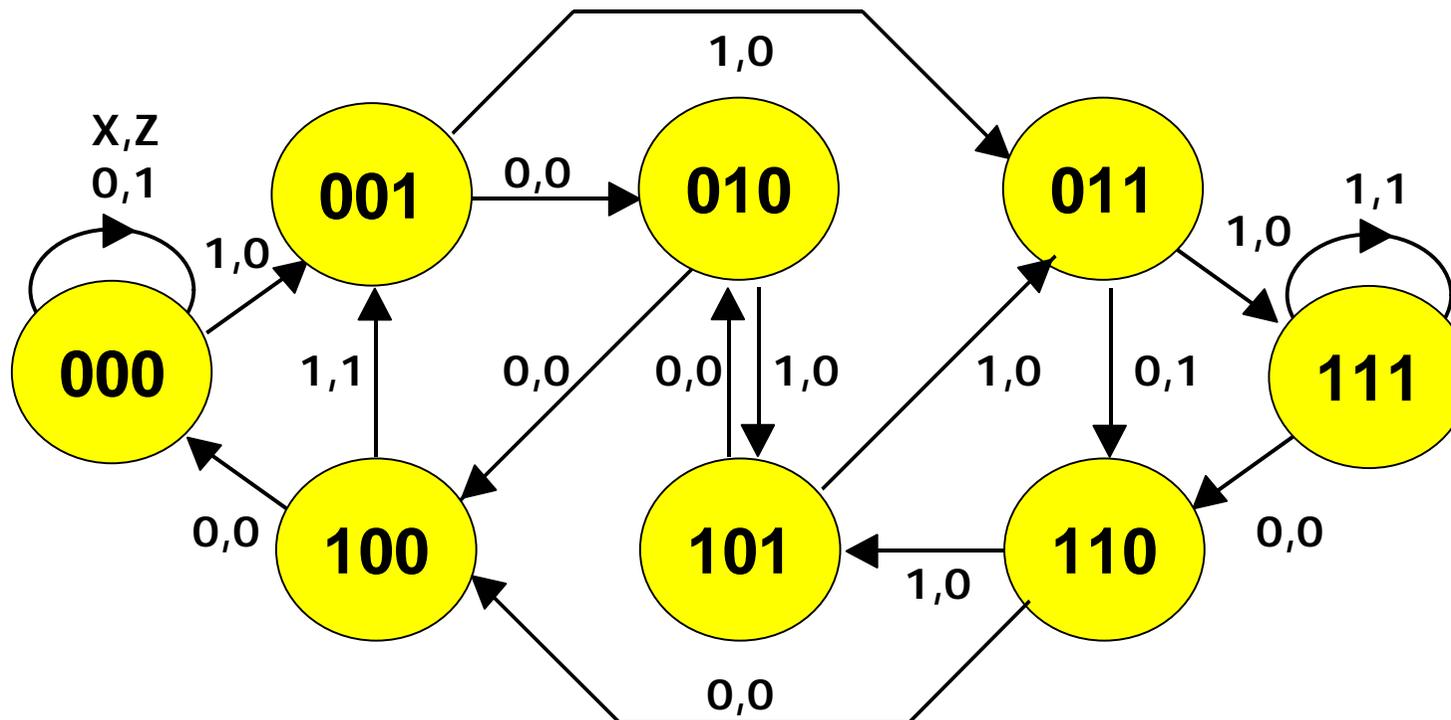


Tabella di flusso e tabella triangolare

	X	
	0	1
0	0,1	1,0
1	2,0	3,0
2	4,0	5,0
3	6,1	7,0
4	0,0	1,1
5	2,0	3,0
6	4,0	5,0
7	6,0	7,1

1							
2		24 35					
3	06 17						
4							
5			24 35				
6		24 35			24 35		
7					06 17		
	0	1	2	3	4	5	6

stati distinguibili per almeno una configurazione d'ingresso

stati indistinguibili per ogni sequenza d'ingresso di lunghezza unitaria, ma distinguibili per almeno una sequenza d'ingresso di lunghezza superiore

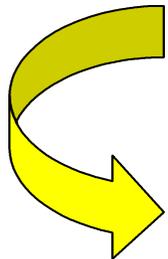
stati indistinguibili

Classi di indistinguibilità

{0} {1,5} {2,6} {3} {4} {7}

Tabella di flusso minima

	X	
	0	1
0	0,1	1,0
1	2,0	3,0
2	4,0	1,0
3	2,1	7,0
4	0,0	1,1
7	2,0	7,1



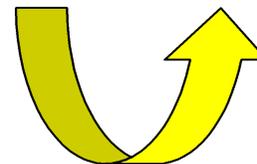
Mappa di codifica

y_3/y_2y_1	00	01	11	10
0	0	1	3	2
1	4	-	7	-

Tabella delle transizioni

	X^n	
$(y_3y_2y_1)^n$	0	1
000	000,1	001,0
001	010,0	011,0
011	010,1	111,0
010	100,0	001,0
100	000,0	001,1
101	---,-	---,-
111	010,0	111,1
110	---,-	---,-

$(y_3y_2y_1)^{n+1}, Z^n$



Reti combinatorie di aggiornamento dello stato e d'uscita

	X^n	
$(y_3y_2y_1)^n$	0	1
000	0	0
001	0	0
011	0	1
010	1	0
100	0	0
101	-	-
111	0	1
110	-	-

	X^n	
$(y_3y_2y_1)^n$	0	1
000	0	0
001	1	1
011	1	1
010	0	0
100	0	0
101	-	-
111	1	1
110	-	-

	X^n	
$(y_3y_2y_1)^n$	0	1
000	0	1
001	0	1
011	0	1
010	0	1
100	0	1
101	-	-
111	0	1
110	-	-

	X^n	
$(y_3y_2y_1)^n$	0	1
000	1	0
001	0	0
011	1	0
010	0	0
100	0	1
101	-	-
111	0	1
110	-	-

$$D_3^n = y_3^{n+1} = (X'y_1'y_2 + Xy_1y_2)^n$$

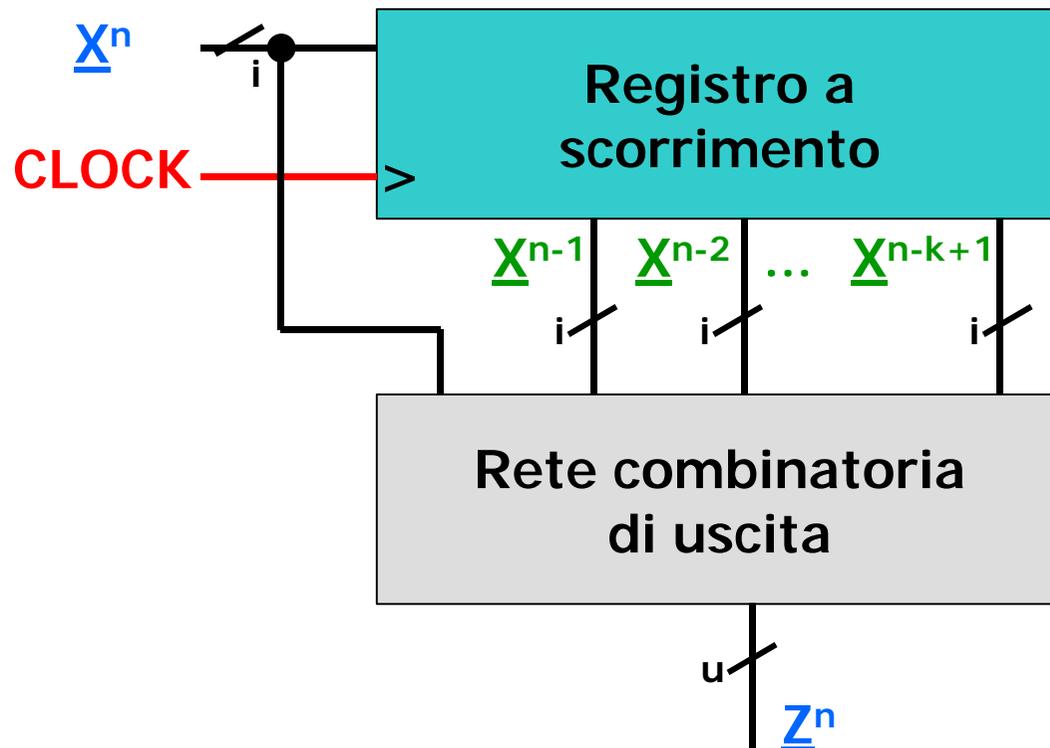
$$D_2^n = y_2^{n+1} = (y_1)^n$$

$$D_1^n = y_1^{n+1} = (X)^n$$

$$Z^n = (X'y_1'y_2'y_3' + X'y_1y_2y_3' + Xy_3)^n$$

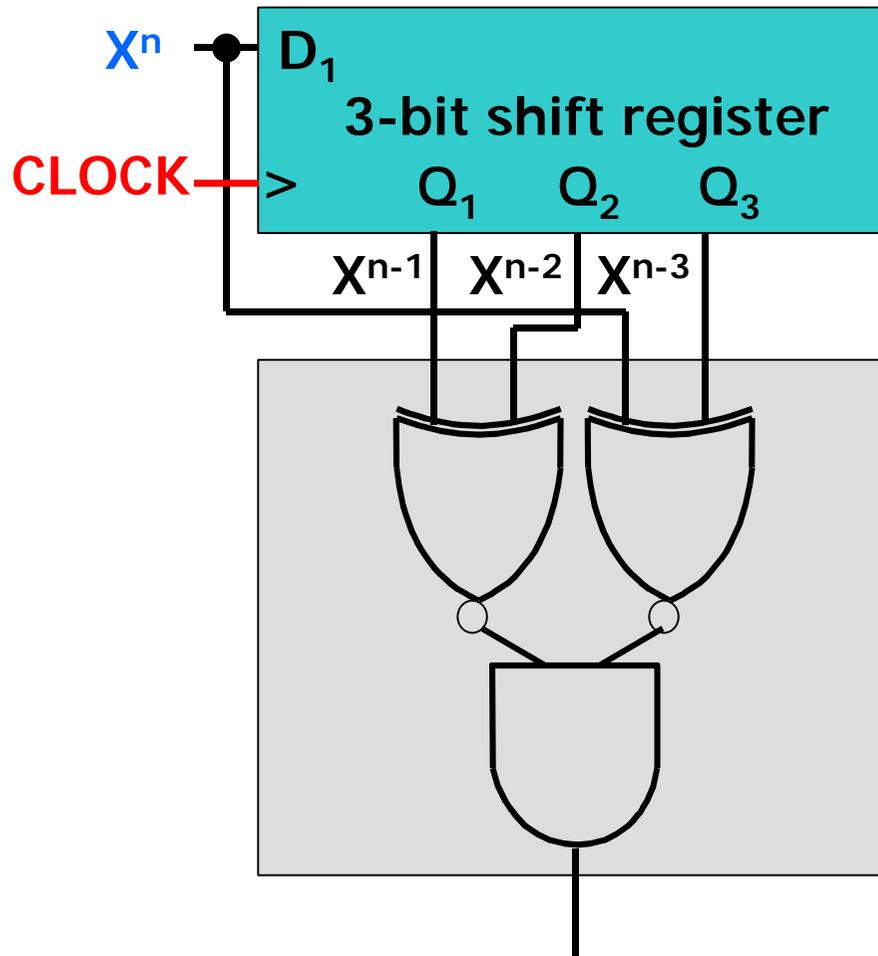
Quando i segnali di uscita $\underline{Z} = \{Z_1, Z_2, \dots, Z_u\}$ di una rete sequenziale sincrona dipendono in ogni intervallo di clock direttamente dalle ultime k configurazioni dei segnali di ingresso $\underline{X} = \{X_1, X_2, \dots, X_i\}$, è possibile eseguire il progetto in maniera rapida, strutturata e flessibile applicando il seguente

1° modello di riferimento
per macchine con ampiezza di memoria finita



capacità [bit] =
 $i \cdot (k-1)$

Applicazione del modello al problema in esame (i=u=1, k=4):

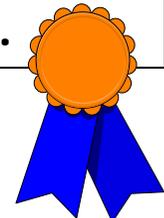


Unità di memoria

Unità di elaborazione

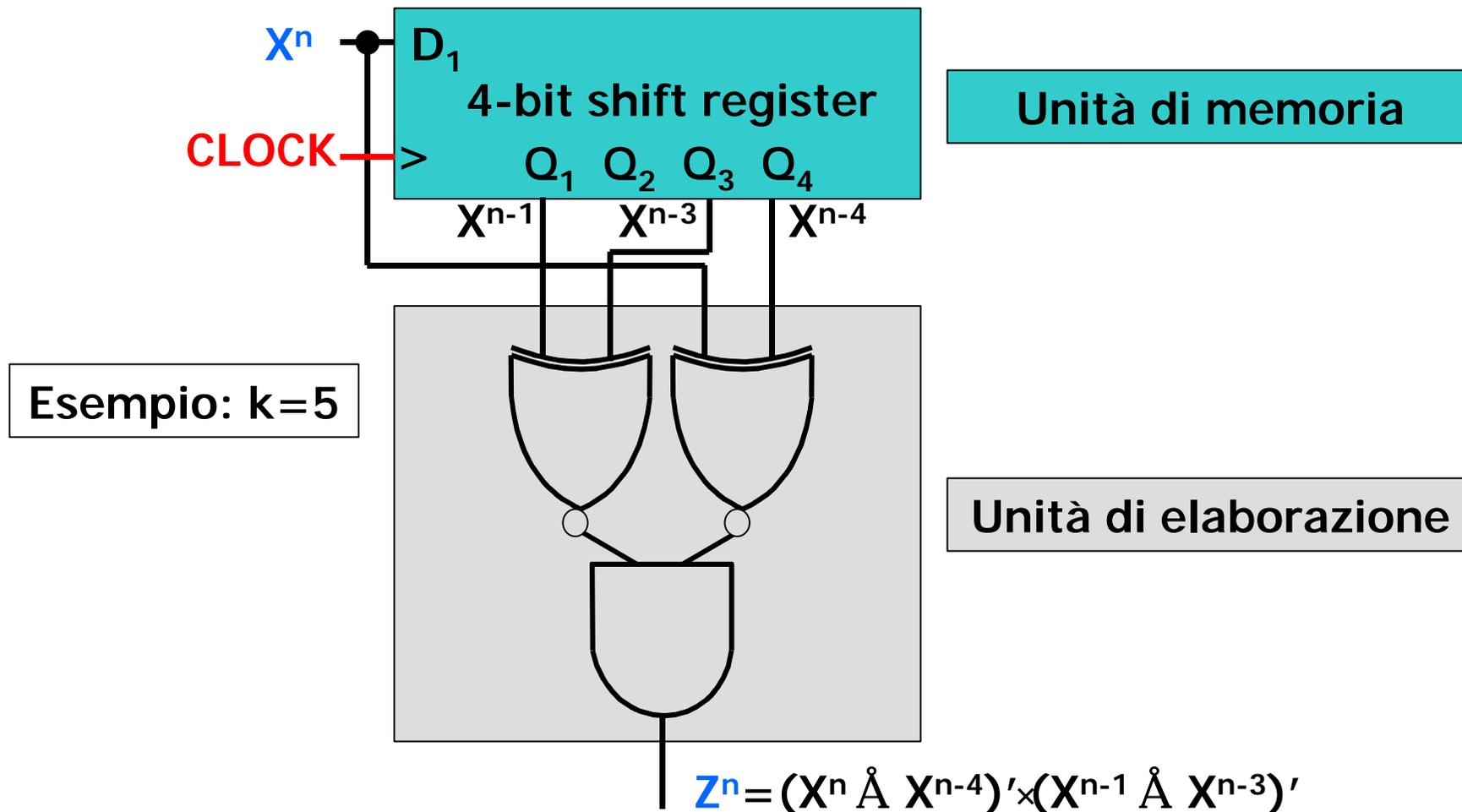
Rispetto alla realizzazione precedentemente identificata seguendo l'approccio convenzionale, questa soluzione è non solo più immediata ed economica, ma anche più flessibile ...

$$Z^n = (X^n \dot{\wedge} X^{n-3})' \times (X^{n-1} \dot{\wedge} X^{n-2})'$$



La flessibilità del modello

Se infatti la specifica di progetto richiedesse di attivare Z se non più gli ultimi 4, bensì gli ultimi k (" k) valori di X costituiscono una palindroma, basterebbe ricorrere ad un registro a scorrimento di k-1 bit ed adattare coerentemente il comparatore in uscita:



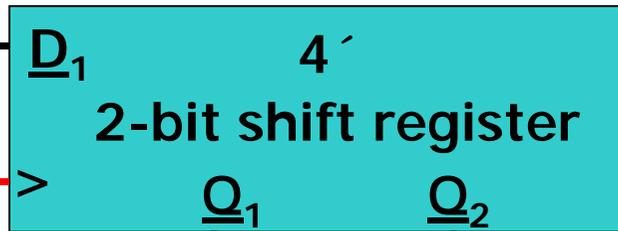
Esercizio 4

Una rete sequenziale sincrona è caratterizzata da quattro segnali di ingresso (X_3, X_2, X_1, X_0) e da un segnale di uscita (Z), tutti sincroni. In ogni intervallo di clock la rete riceve in ingresso una cifra decimale codificata in BCD (X_3 rappresenta il bit più significativo, X_0 il bit meno significativo). L'uscita della rete nel generico n-esimo intervallo di clock (Z^n) dipende dalle ultime tre cifre applicate in ingresso (N^n, N^{n-1}, N^{n-2}). In particolare $Z^n=1$ se una soltanto o tutte e tre le cifre sono prime. In caso contrario $Z^n=0$.

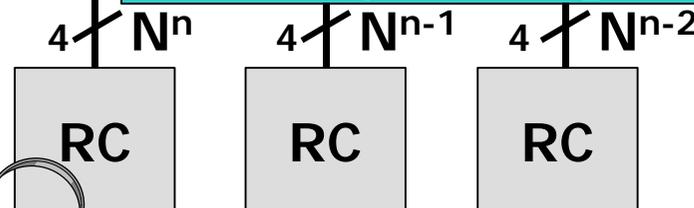
La soluzione in accordo al 1° modello introdotto nell'esercizio precedente

$(X_3 X_2 X_1 X_0)^n$

CLOCK



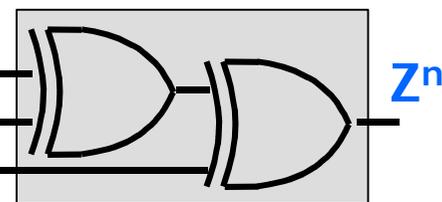
Unità di memoria



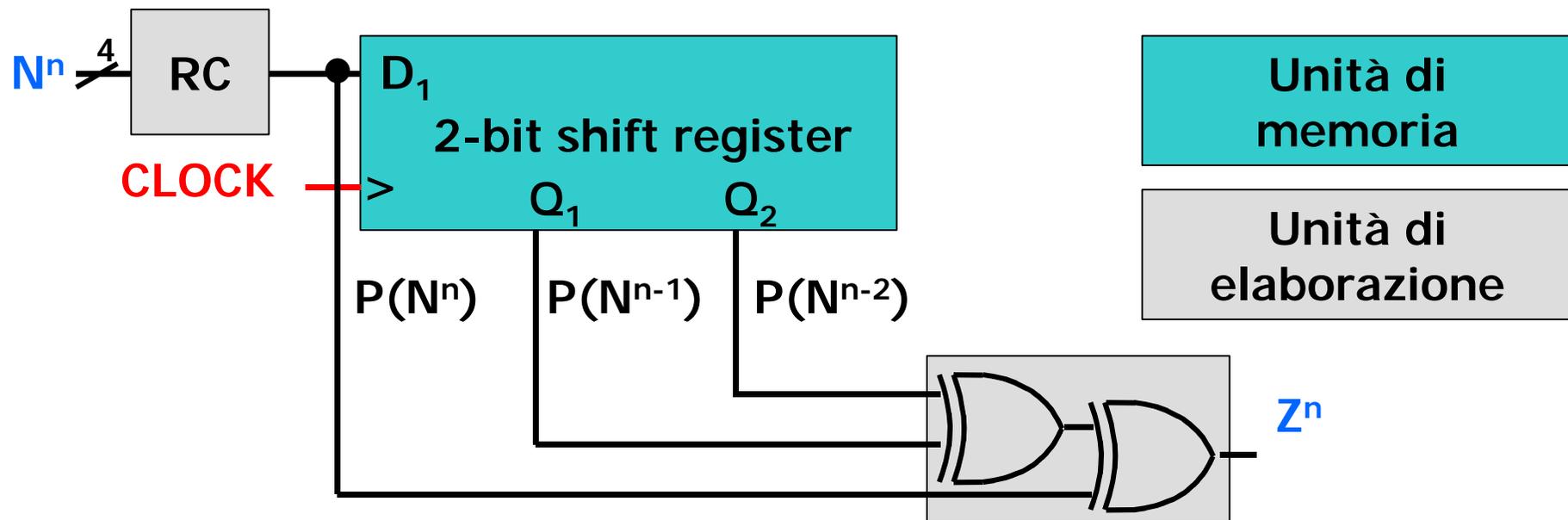
Unità di elaborazione

		$X_1 X_0$			
		00	01	11	10
$X_3 X_2$	00	0	1	1	1
	01	0	1	1	0
	11	-	-	-	-
	10	0	0	-	-

$$P(X_3, X_2, X_1, X_0) = X_3' X_0 + X_2' X_1$$



In realtà, ai fini della generazione del segnale di uscita Z non è necessario memorizzare direttamente le cifre applicate in ingresso e quindi procedere alla loro elaborazione. Si può più convenientemente preelaborare ciascuna cifra, verificando se soddisfa o meno la proprietà richiesta (ovvero se è prima), e memorizzare solo il risultato di tale verifica:



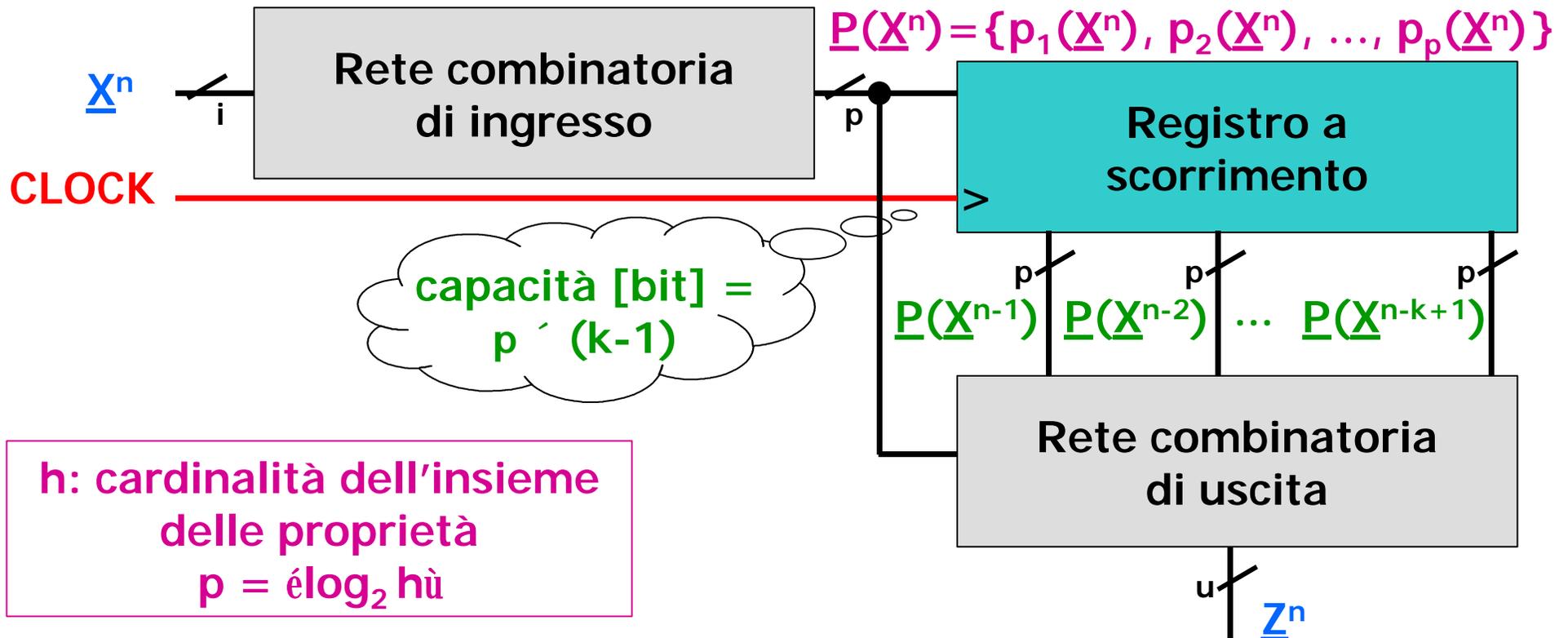
La capacità di memoria ora richiesta è pari al 25%.
La rete combinatoria RC non è più inutilmente replicata.



La soluzione è immediatamente configurabile per qualunque valore di k (basta ricorrere ad un registro a scorrimento di $k-1$ bit ed adattare coerentemente il solo generatore di parità).

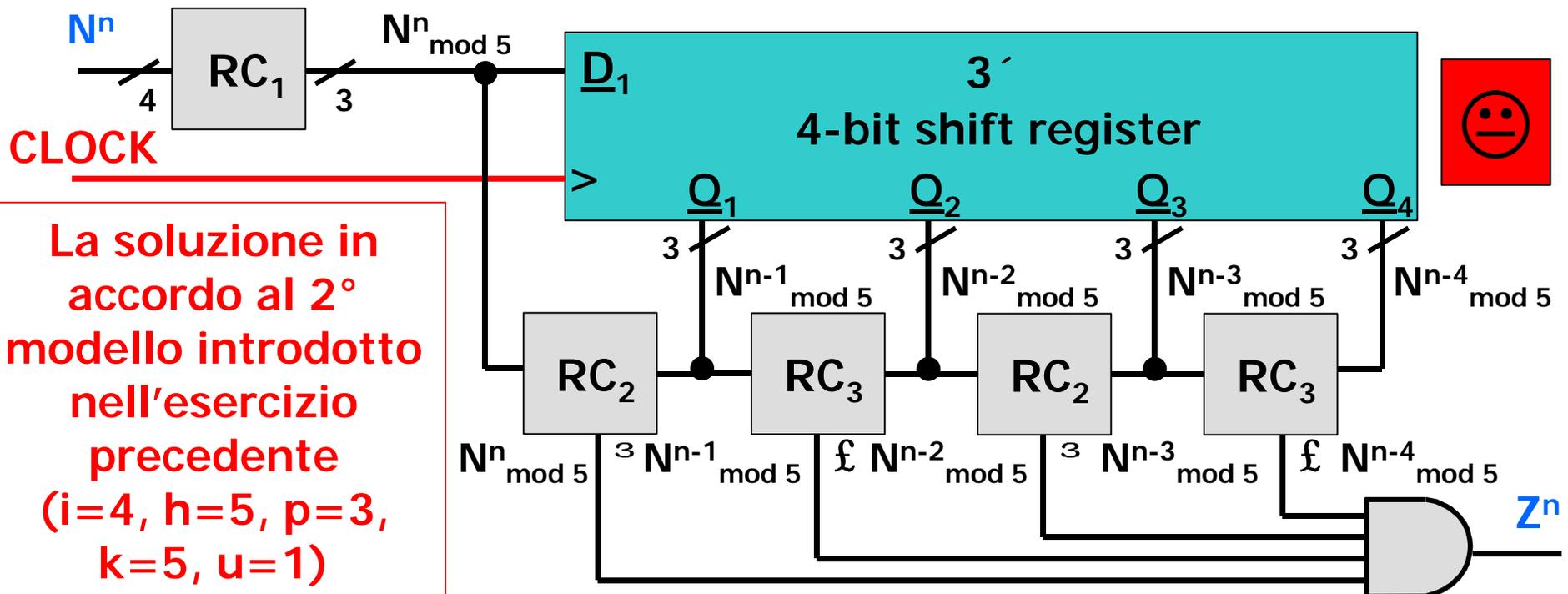
Quando i segnali di uscita $\underline{Z} = \{Z_1, Z_2, \dots, Z_u\}$ di una rete sequenziale sincrona dipendono in ogni intervallo di clock da una qualche proprietà associata alle ultime k configurazioni dei segnali di ingresso $\underline{X} = \{X_1, X_2, \dots, X_i\}$, è possibile eseguire il progetto in maniera rapida, strutturata e flessibile applicando il seguente

2° modello di riferimento
per macchine con ampiezza di memoria finita



Esercizio 5

Una rete sequenziale sincrona è caratterizzata da quattro segnali di ingresso (X_3, X_2, X_1, X_0) e da un segnale di uscita (Z), tutti sincroni. In ogni intervallo di clock la rete riceve in ingresso una cifra decimale codificata in BCD (X_3 rappresenta il bit più significativo, X_0 il bit meno significativo). L'uscita della rete nel generico n -esimo intervallo di clock (Z^n) dipende dalle ultime cinque cifre applicate in ingresso ($N^n, N^{n-1}, N^{n-2}, N^{n-3}, N^{n-4}$). In particolare $Z^n=1$ se e soltanto se $N^{n-j+1} \bmod 5 \geq N^{n-j} \bmod 5, j = 1, 3, N^{n-j+1} \bmod 5 \leq N^{n-j} \bmod 5, j = 2, 4$ (ovvero se $N^n \geq N^{n-1} \leq N^{n-2} \geq N^{n-3} \leq N^{n-4}$).

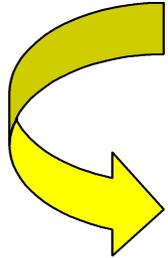


RC₂

$(p_2 p_1 p_0)^{n-j}$

	000	001	011	010	100	101	111	110
000	1	0	0	0	0	-	-	-
001	1	1	0	0	0	-	-	-
011	1	1	1	1	0	-	-	-
$(p_2 p_1 p_0)^{n-j+1}$ 010	1	1	0	1	0	-	-	-
100	1	1	1	1	1	-	-	-
101	-	-	-	-	-	-	-	-
111	-	-	-	-	-	-	-	-
110	-	-	-	-	-	-	-	-

$$(r_j)^n = (p_2)^{n-j+1} + (p_2')^{n-j} (\dots)$$

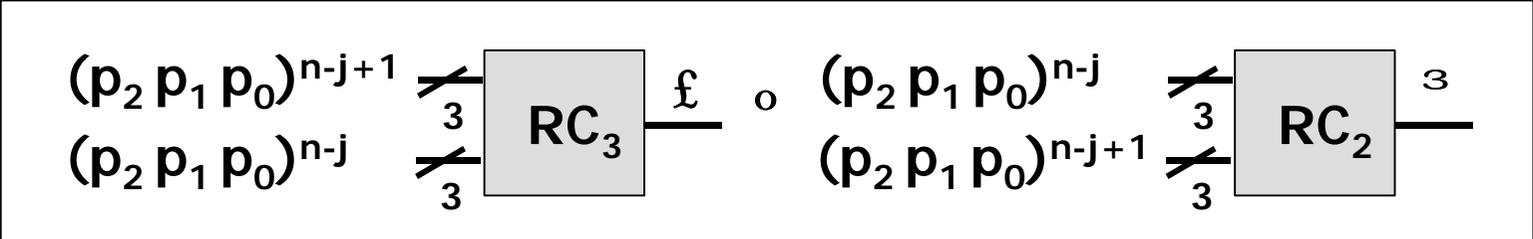


$(p_1 p_0)^{n-j}$

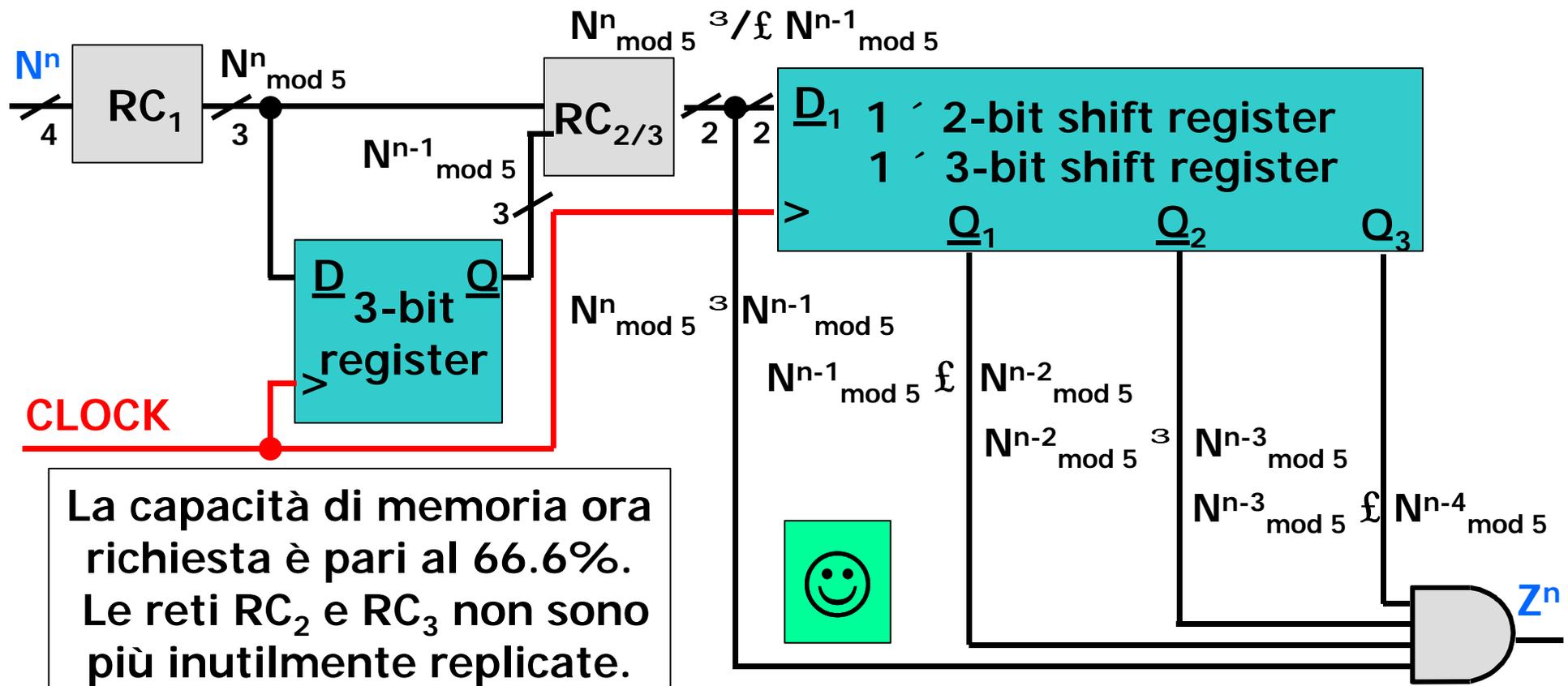
	00	01	11	10
00	1	0	0	0
01	1	1	0	0
11	1	1	1	1
10	1	1	0	1

$$(r_j)^n = ((p_2 p_1 p_0)^{n-j+1} \geq (p_2 p_1 p_0)^{n-j})$$

$$\begin{aligned} \dots &= (p_1 p_0)^{n-j+1} \\ &+ (p_1)^{n-j+1} (p_1')^{n-j} \\ &+ (p_1)^{n-j+1} (p_0')^{n-j} \\ &+ (p_0)^{n-j+1} (p_1')^{n-j} \\ &+ (p_1' p_0')^{n-j} \end{aligned}$$

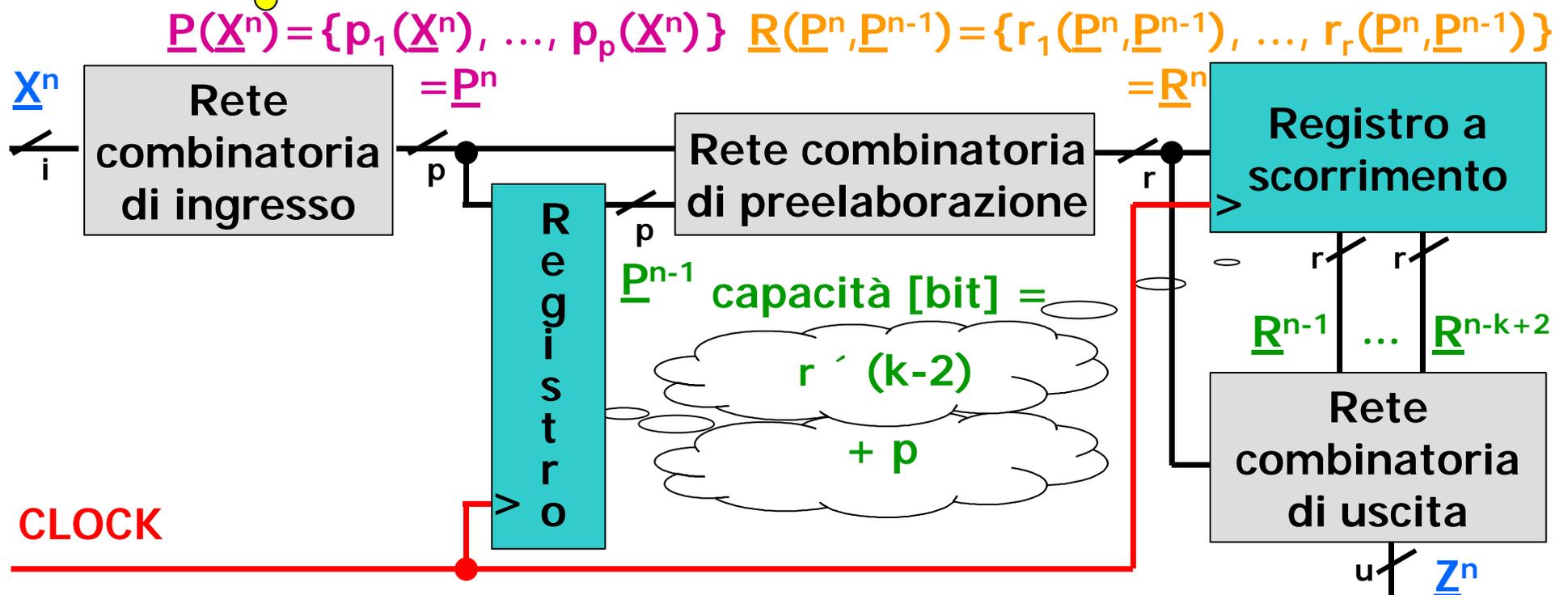


Ai fini della generazione del segnale di uscita non è necessario memorizzare via via i bit rappresentativi della proprietà (il resto della divisione per cinque) associata alle 4 cifre precedentemente applicate in ingresso e quindi procedere alle comparazioni. Si può più convenientemente memorizzare i bit relativi soltanto all'ultima cifra, confrontarli con quelli associati alla cifra attualmente in ingresso, e generare Z in base al risultato di tale confronto, unitamente al risultato dei tre precedenti analoghi confronti:



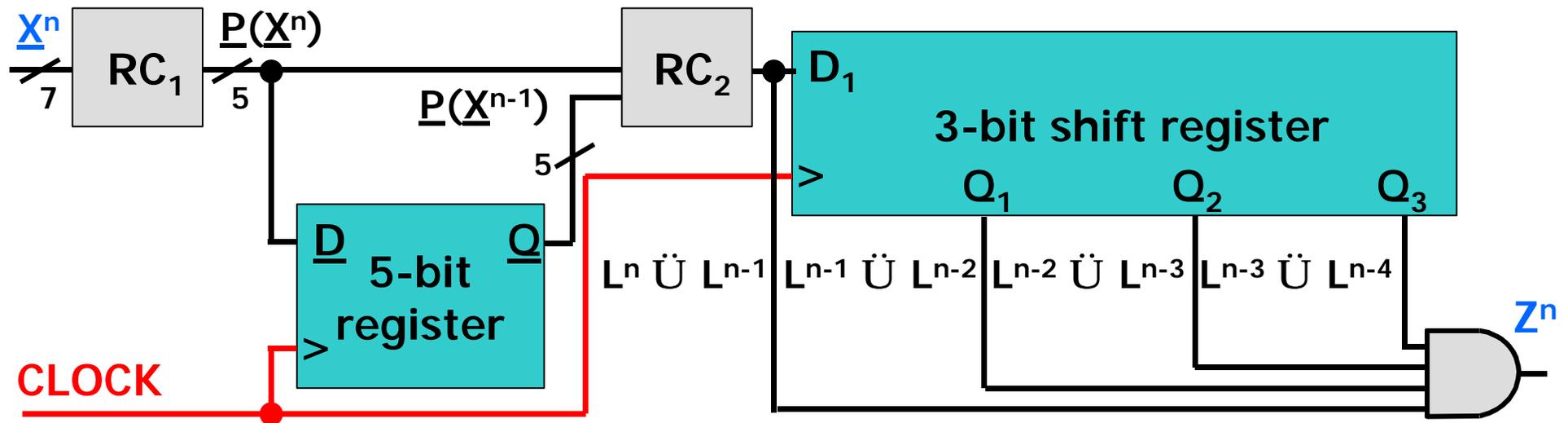
Quando i segnali di uscita $\underline{Z} = \{Z_1, Z_2, \dots, Z_u\}$ di una rete sequenziale sincrona dipendono in ogni intervallo di clock dalle (o da una proprietà delle) ultime k configurazioni dei segnali di ingresso $\underline{X} = \{X_1, X_2, \dots, X_i\}$ secondo relazioni che coinvolgono iterativamente ciascuna delle $k-1$ coppie di (proprietà di) configurazioni consecutive, è possibile eseguire il progetto in maniera rapida, strutturata e flessibile applicando il seguente

3° modello di riferimento
per macchine con ampiezza di memoria finita



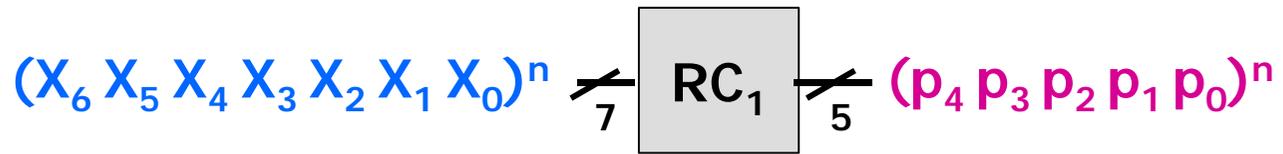
Esercizio 6

Una rete sequenziale sincrona è caratterizzata da sette segnali di ingresso ($X_6, X_5, X_4, X_3, X_2, X_1, X_0$) e da un segnale di uscita (Z), tutti sincroni. In ogni intervallo di clock la rete riceve in ingresso un simbolo alfanumerico rappresentato secondo il codice ASCII (X_6 rappresenta il bit più significativo, X_0 il bit meno significativo). L'uscita della rete nel generico n -esimo intervallo di clock (Z^n) dipende dagli ultimi 5 simboli applicati in ingresso ($\underline{X}^n, \underline{X}^{n-1}, \underline{X}^{n-2}, \underline{X}^{n-3}, \underline{X}^{n-4}$). In particolare Z^n deve valere 1 solo se tali simboli corrispondono a lettere maiuscole, diverse fra loro, ed in ordine alfabetico.

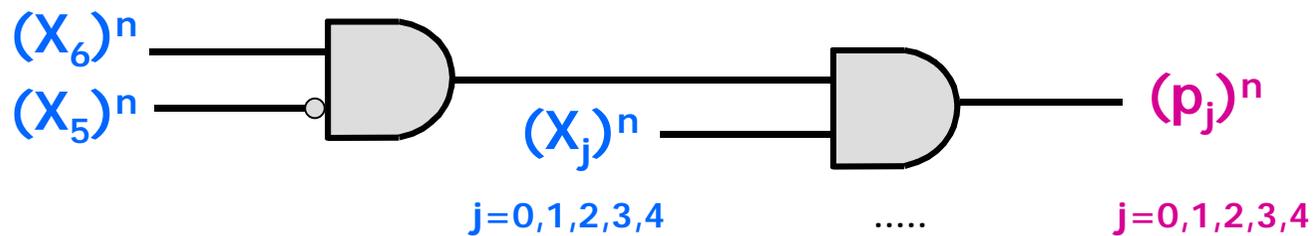


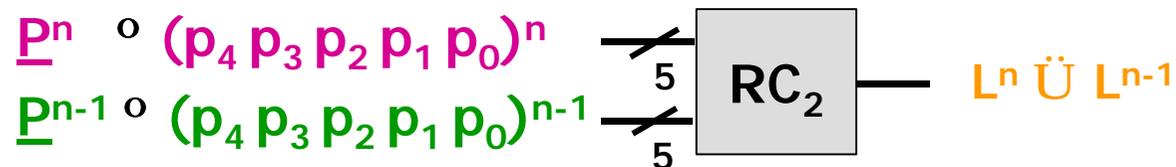
La soluzione in accordo al 3° modello introdotto nell'esercizio precedente ($i=7, h=27, p=5, r=1, k=5, u=1$)





	X_6	X_5	X_4	X_3	X_2	X_1	X_0	p_4	p_3	p_2	p_1	p_0
0	0	1	1	0	0	0	0	0	0	0	0	0
1	0	1	1	0	0	0	1	0	0	0	0	0
...	0	1	1	0	0	0	0	0
9	0	1	1	1	0	0	1	0	0	0	0	0
A	1	0	0	0	0	0	1	0	0	0	0	1
B	1	0	0	0	0	1	0	0	0	0	1	0
...	1	0
Z	1	0	1	1	0	1	0	1	1	0	1	0
a	1	1	0	0	0	0	1	0	0	0	0	0
b	1	1	0	0	0	1	0	0	0	0	0	0
...	1	1	0	0	0	0	0
z	1	1	1	1	0	1	0	0	0	0	0	0





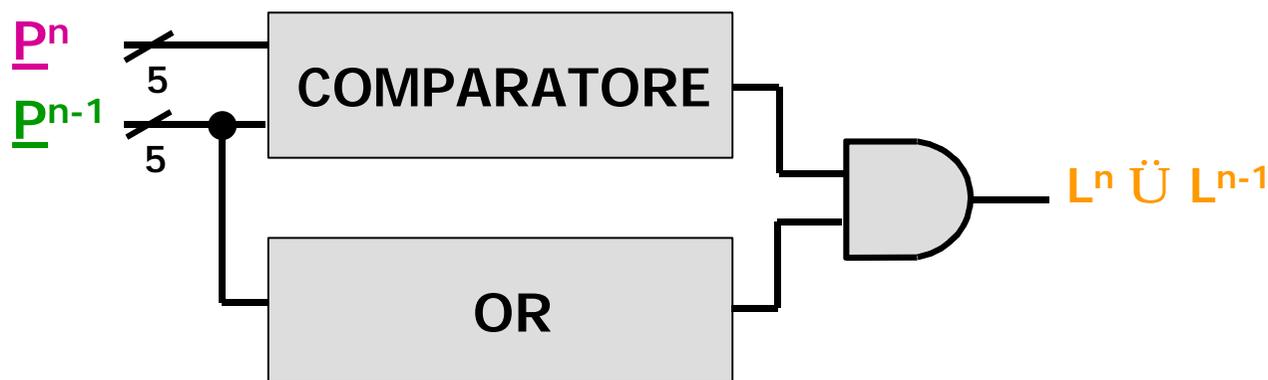
$L^n \dot{\cup} L^{n-1}$ (coppia di lettere maiuscole in ordine alfabetico)

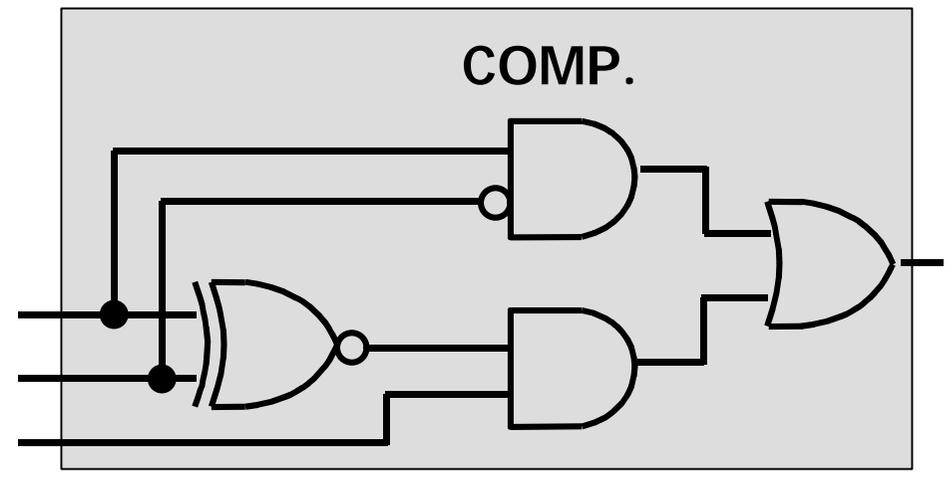
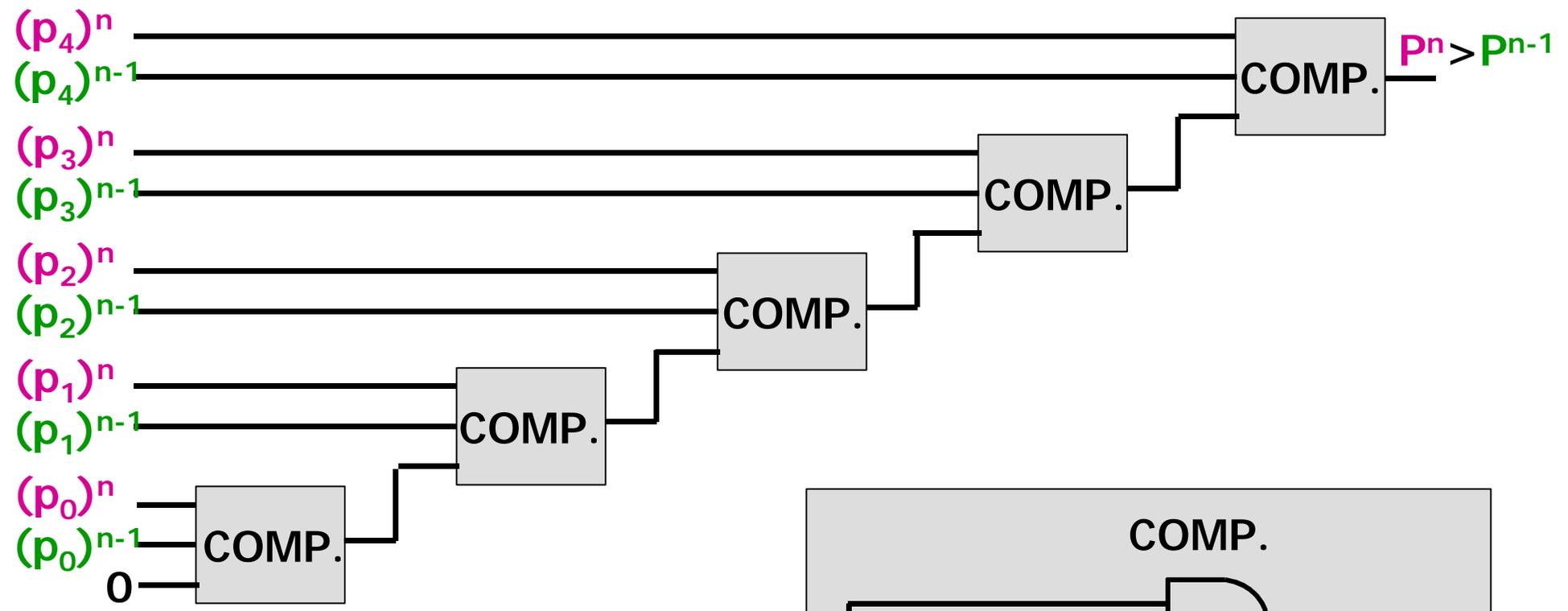
se $(\underline{P}^n > \underline{P}^{n-1}) \ \& \ (\underline{P}^n > 0) \ \& \ (\underline{P}^{n-1} > 0)$

ovvero se $(\underline{P}^n > \underline{P}^{n-1}) \ \& \ (\underline{P}^{n-1} > 0)$

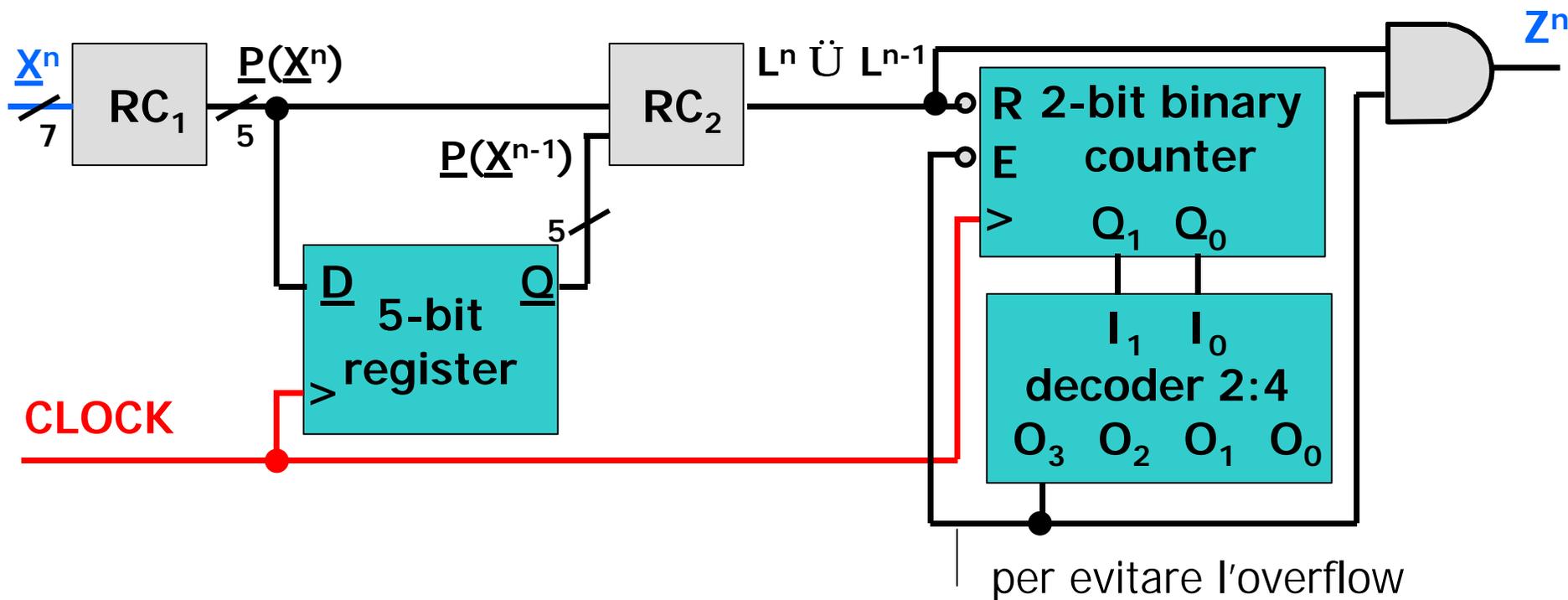
$$L^n \dot{\cup} L^{n-1} = \{ (p_4)^n (p_4')^{n-1} + ((p_4)^n \dot{\wedge} (p_4)^{n-1})' [(p_3)^n (p_3')^{n-1} + ((p_3)^n \dot{\wedge} (p_3)^{n-1})' [(p_2)^n (p_2')^{n-1} + ((p_2)^n \dot{\wedge} (p_2)^{n-1})' [(p_1)^n (p_1')^{n-1} + ((p_1)^n \dot{\wedge} (p_1)^{n-1})' [(p_0)^n (p_0')^{n-1}]]]] \}$$

$$\{ (p_4 + p_3 + p_2 + p_1 + p_0)^{n-1} \}$$





Ai fini della generazione del segnale di uscita non è necessario memorizzare il risultato dei singoli confronti e combinarli poi attraverso un operatore AND. Basta contarli !!!



La capacità di memoria ora richiesta è pari all'87.5 %.



La soluzione è immediatamente configurabile per qualunque valore di k (basta modificare la base del contatore $(k-1)$ ed eventualmente il decoder). Rispetto alla soluzione precedente, la capacità di memoria richiesta si riduce da $5 + (k-2)$ bit a $5 + \lceil \log_2(k-1) \rceil$ bit.

Quando il segnale di uscita Z di una rete sequenziale sincrona dipende in ogni intervallo di clock dalle (o da una proprietà delle) ultime k configurazioni dei segnali di ingresso $\underline{X} = \{X_1, X_2, \dots, X_i\}$ secondo una relazione che coinvolge iterativamente ed identicamente ciascuna delle $k-1$ coppie di (proprietà di) configurazioni consecutive, è possibile eseguire il progetto in maniera rapida, strutturata e flessibile applicando il seguente

4° modello di riferimento
per macchine con ampiezza di memoria finita

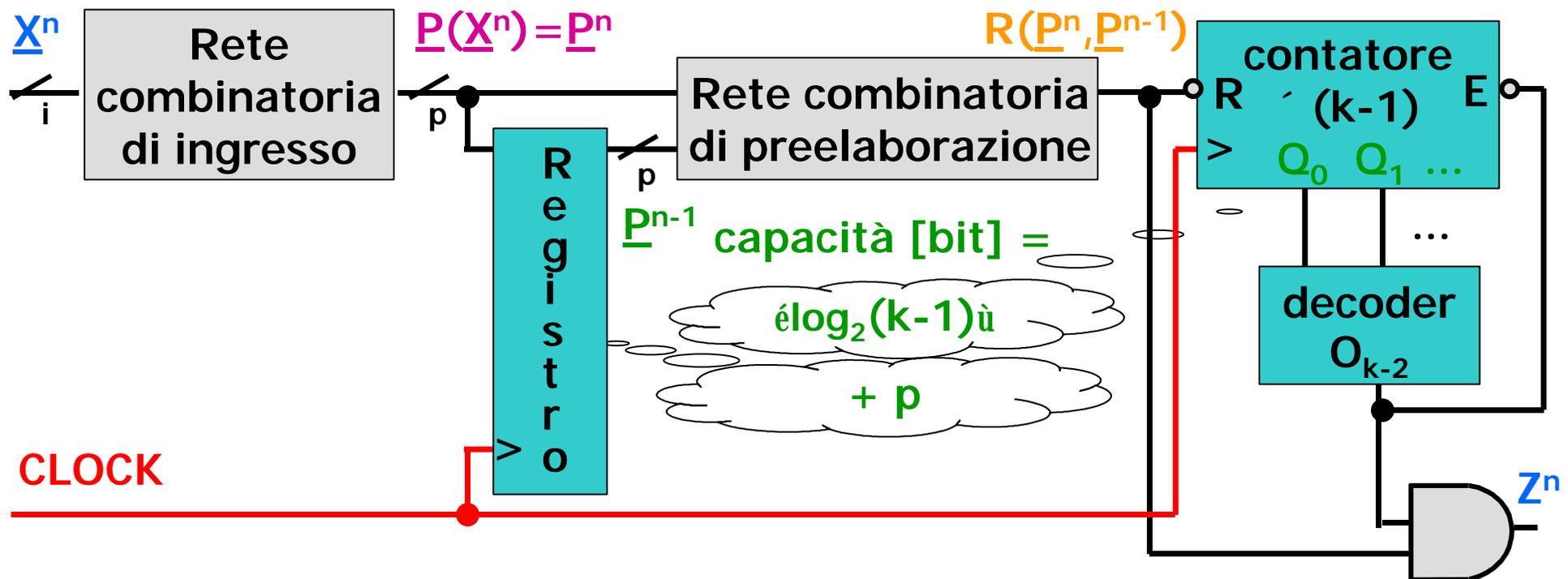
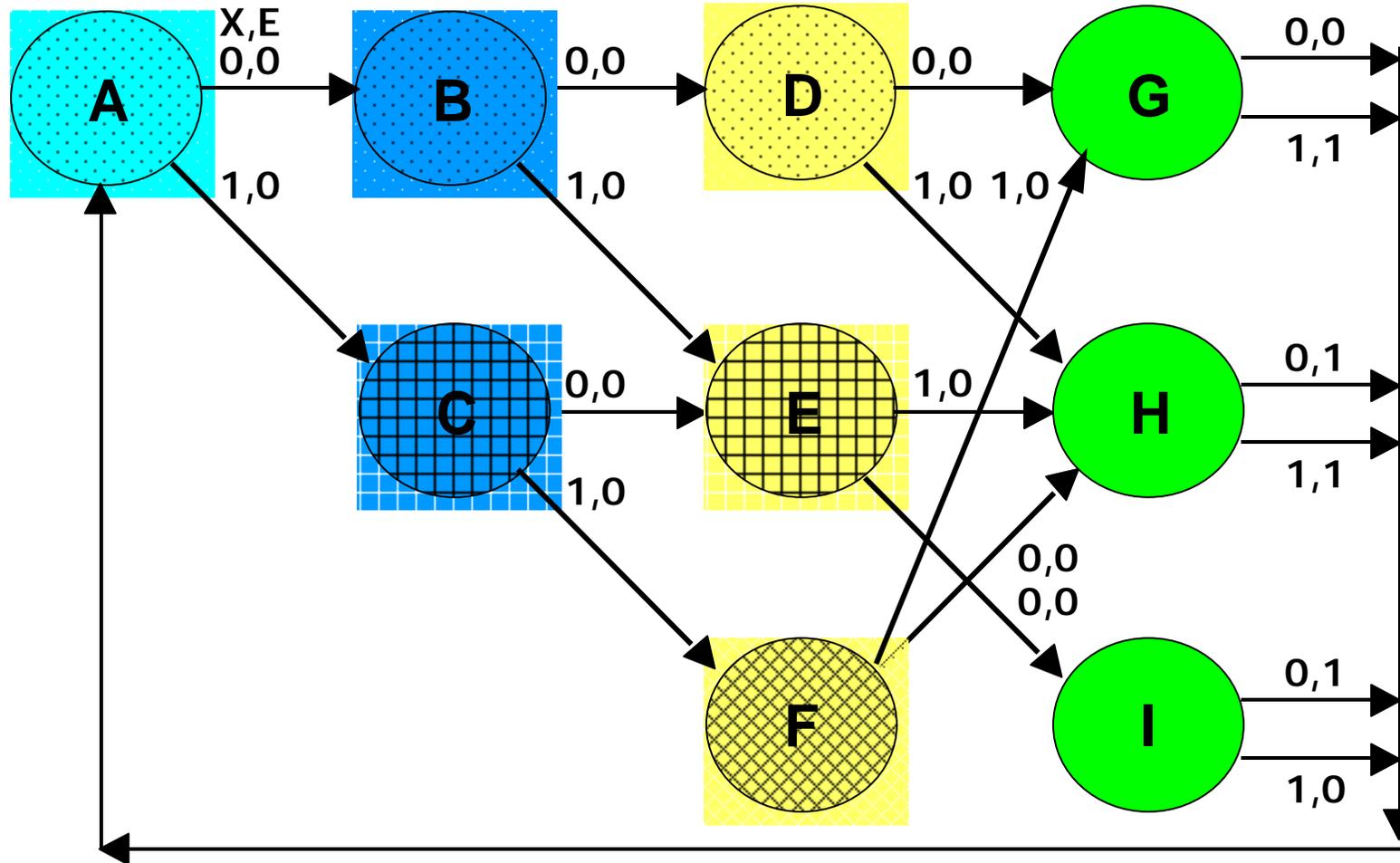
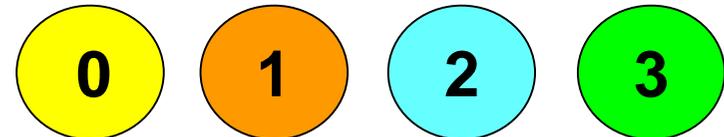


Diagramma degli stati (modello di Mealy)



Numero di bit già ricevuti nell'ambito di una parola:



Numero di bit di informazione uguali a 1 già rilevati nell'ambito di una parola:

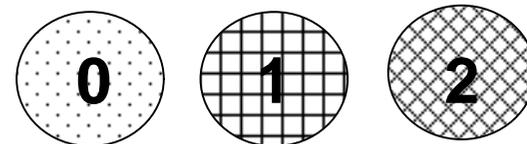


Tabella di flusso e tabella triangolare

	X	
	0	1
A	B,0	C,0
B	D,0	E,0
C	E,0	F,0
D	G,0	H,0
E	I,0	H,0
F	H,0	G,0
G	A,0	A,1
H	A,1	A,1
I	A,1	A,0

B	BD CE							
C	BE CF	DE EF						
D	BG CH	DG EH	EG FH					
E	BI CH	DI EH	EI FH	GI				
F	BH CG	DH EG	EH FG	GH	HI GH			
G								
H								
I								
	A	B	C	D	E	F	G	H

stati distinguibili per almeno una configurazione d'ingresso

stati indistinguibili per ogni sequenza d'ingresso di lunghezza unitaria, ma distinguibili per almeno una sequenza d'ingresso di lunghezza superiore

Tabella di flusso non riducibile

Mappa di codifica e tabella delle transizioni

y_3y_4

	00	01	11	10	
y_1y_2	00	A	B	G	D
	01	-	C	H	E
	11	-	-	-	-
	10	-	-	I	F

nessun vincolo

$(y_3y_4)^n$

	00	01	11	10	
$(Xy_1y_2)^n$	000	0001,0	0010,0	0000,0	0011,0
	001	----,-	0110,0	0000,1	1011,0
	011	----,-	----,-	----,-	----,-
	010	----,-	----,-	0000,1	0111,0
	100	0101,0	0110,0	0000,1	0111,0
	101	----,-	1010,0	0000,1	0111,0
	111	----,-	----,-	----,-	----,-
	110	----,-	----,-	0000,0	0011,0

$(y_1y_2y_3y_4)^{n+1}, Z^n$

Rete combinatoria di aggiornamento dello stato ...

$(y_3y_4)^n$

	00	01	11	10
$(Xy_1y_2)^n$	000	0	0	0
	001	-	0	1
	011	-	-	-
	010	-	0	0
	100	0	0	0
	101	-	1	0
	111	-	-	-
	110	-	0	0
		y_1^{n+1}		

$$J_1^n = (X'y_2y_4' + Xy_2y_3')^n$$

$$K_1^n = 1$$

$(y_3y_4)^n$

	00	01	11	10
$(Xy_1y_2)^n$	0	0	0	0
	-	1	0	0
	-	-	-	-
	-	-	0	1
	1	1	0	1
	-	0	0	1
	-	-	-	-
	-	-	0	0
		y_2^{n+1}		

$$J_2^n = (X'y_1y_4' + Xy_3' + Xy_1'y_4')^n$$

$$K_2^n = (X'y_1'y_3 + Xy_4)^n$$

... Rete combinatoria di aggiornamento dello stato

$(y_3y_4)^n$

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

y_3^{n+1}

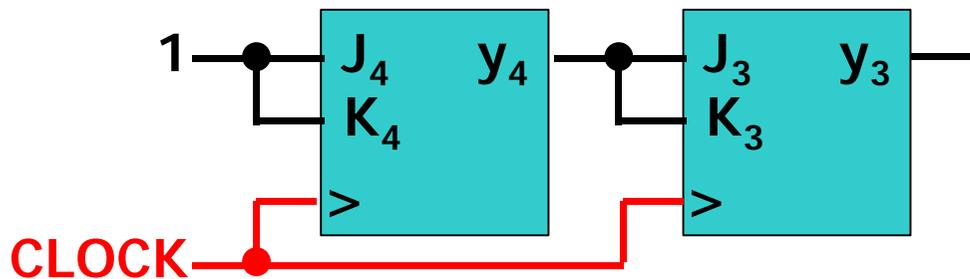
$$J_3^n = K_3^n = y_4^n$$

$(y_3y_4)^n$

	00	01	11	10
1	0	0	0	1
-	0	0	0	1
-	-	-	-	-
-	-	-	0	1
1	0	0	0	1
-	0	0	0	1
-	-	-	-	-
-	-	0	0	1

y_4^{n+1}

$$J_4^n = K_4^n = 1$$



La particolare codifica selezionata conduce ad una rete nella quale è identificabile la presenza di un contatore binario $\times 4$.
Esso ha il compito di partizionare in parole i bit applicati in ingresso.

Rete combinatoria di uscita

$(y_3y_4)^n$

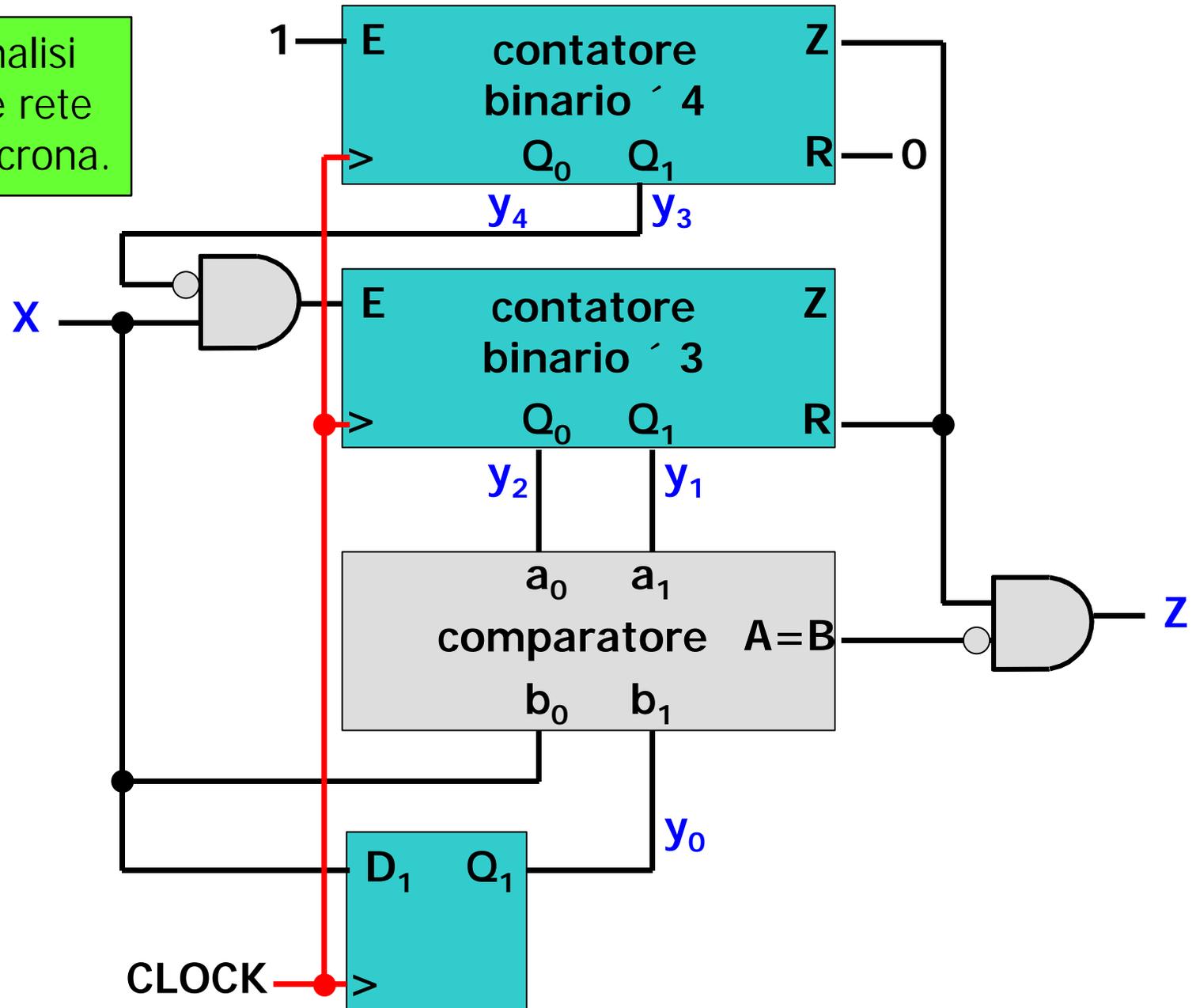
	00	01	11	10
000	0	0	0	0
001	-	0	1	0
011	-	-	-	-
$(Xy_1y_2)^n$ 010	-	-	1	0
100	0	0	1	0
101	-	0	1	0
111	-	-	-	-
110	-	-	0	0

Z^n

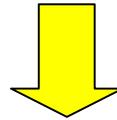
$$Z^n = (y_2y_3y_4 + Xy_1'y_3y_4 + X'y_1y_4)^n$$

Esercizio 8

Si esegua l'analisi della seguente rete sequenziale sincrona.



Schema logico



Espressioni delle variabili di stato interno futuro e di uscita in funzione delle variabili di stato interno presente e di ingresso:

$$y_0^{n+1} = X^n$$

$$y_1^{n+1} = ((y_1 \dot{\wedge} (X y_3' y_2)) (y_1 (y_2 + X y_3') + y_3 y_4)')^n$$

$$y_2^{n+1} = ((y_2 \dot{\wedge} (X y_3')) (y_1 (y_2 + X y_3') + y_3 y_4)')^n$$

$$y_3^{n+1} = (y_3 \dot{\wedge} y_4)^n$$

$$y_4^{n+1} = (y_4')^n$$

$$Z^n = (y_3 y_4 ((y_1 \dot{\wedge} y_0) + (y_2 \dot{\wedge} X)))^n$$

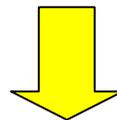


Tabella delle transizioni

Tabella delle transizioni

		$(Xy_3y_4)^n$							
		000	001	011	010	100	101	111	110
$(y_0y_1y_2)^n$	000	00001,0	00010,0	00000,0	00011,0	10101,0	10110,0	10000,1	10011,0
	001	00101,0	00110,0	00000,1	00111,0	11001,0	11010,0	10000,0	10111,0
	011	00001,0	00010,0	00000,1	00011,0	10001,0	10010,0	10000,1	10011,0
	010	01001,0	01010,0	00000,1	01011,0	10001,0	10010,0	10000,1	11011,0
	100	00001,0	00010,0	00000,1	00011,0	10101,0	10110,0	10000,1	10011,0
	101	00101,0	00110,0	00000,1	00111,0	11001,0	11010,0	10000,1	10111,0
	111	00001,0	00010,0	00000,1	00011,0	10001,0	10010,0	10000,0	10011,0
	110	01001,0	01010,0	00000,0	01011,0	10001,0	10010,0	10000,1	11011,0

$(y_0y_1y_2y_3y_4)^{n+1}, Z^n$

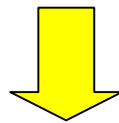


Tabella di flusso

ciascuno stato è identificato con il valore decimale (0, 1, ..., 31)
corrispondente alla relativa rappresentazione binaria (00000, 00001, ...,
11111):

Tabella di flusso

(in una forma, date le dimensioni, non del tutto convenzionale per motivi di disegno)

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
X	0	1,0	2,0	3,0	0,0	5,0	6,0	7,0	0,1	9,0	10,0	11,0	0,1	1,0	2,0	3,0	0,1
	1	21,0	22,0	19,0	16,1	25,0	26,0	23,0	16,0	17,0	18,0	27,0	16,1	17,0	18,0	19,0	16,1
X	0	1,0	2,0	3,0	0,1	5,0	6,0	7,0	0,1	9,0	10,0	11,0	0,0	1,0	2,0	3,0	0,1
	1	21,0	22,0	19,0	16,1	25,0	26,0	23,0	16,1	17,0	18,0	27,0	16,1	17,0	18,0	19,0	16,0
		16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31

Eventuali stati che contraddistinguono il comportamento della rete in transitorio devono essere rimossi dalla tabella di flusso *prima* di applicare l'algoritmo di riduzione.

stati irraggiungibili:

4, 8, 12, 13, 14, 15, 20, 24, 28, 29, 30, 31

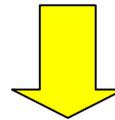


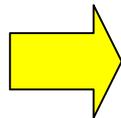
Tabella di flusso ridotta

Tabella di flusso ridotta

	X	
	0	1
0	1,0	21,0
1	2,0	22,0
2	3,0	19,0
3	0,0	16,1
5	6,0	26,0
6	7,0	23,0
7	0,1	16,0
9	10,0	18,0
10	11,0	27,0
11	0,1	16,1
16	1,0	21,0
17	2,0	22,0
18	3,0	19,0
19	0,1	16,1
21	6,0	26,0
22	7,0	23,0
23	0,1	16,1
25	10,0	18,0
26	11,0	27,0
27	0,0	16,1

**stati ora
irraggiungibili**

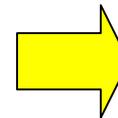
5
9
17
25



	X	
	0	1
0	1,0	21,0
1	2,0	22,0
2	3,0	19,0
3	0,0	16,1
6	7,0	23,0
7	0,1	16,0
10	11,0	27,0
11	0,1	16,1
16	1,0	21,0
18	3,0	19,0
19	0,1	16,1
21	6,0	26,0
22	7,0	23,0
23	0,1	16,1
26	11,0	27,0
27	0,0	16,1

**stati ora
irraggiungibili**

10
18



	X	
	0	1
0	1,0	21,0
1	2,0	22,0
2	3,0	19,0
3	0,0	16,1
6	7,0	23,0
7	0,1	16,0
11	0,1	16,1
16	1,0	21,0
19	0,1	16,1
21	6,0	26,0
22	7,0	23,0
23	0,1	16,1
26	11,0	27,0
27	0,0	16,1

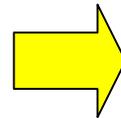
Tabella di flusso ridotta

	X	
	0	1
0	1,0	21,0
1	2,0	22,0
2	3,0	19,0
3	0,0	16,1
6	7,0	23,0
7	0,1	16,0
11	0,1	16,1
16	1,0	21,0
19	0,1	16,1
21	6,0	26,0
22	7,0	23,0
23	0,1	16,1
26	11,0	27,0
27	0,0	16,1

È possibile ridurre ulteriormente e direttamente la tabella osservando che alcune **righe** sono **identiche**:

{0,16} {3,27} {6,22} {11,19,23}

Gli **stati** corrispondenti sono pertanto **indistinguibili**.



	X	
	0	1
0	1,0	21,0
1	2,0	6,0
2	3,0	11,0
3	0,0	0,1
6	7,0	11,0
7	0,1	0,0
11	0,1	0,1
21	6,0	26,0
26	11,0	3,0

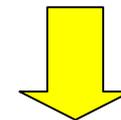


Tabella triangolare

Tabella triangolare e tabella di flusso minima

	X	
	0	1
0	1,0	21,0
1	2,0	6,0
2	3,0	11,0
3	0,0	0,1
6	7,0	11,0
7	0,1	0,0
11	0,1	0,1
21	6,0	26,0
26	11,0	3,0

1	1,2 21,6							
2	1,3 21,11	2,3 6,11						
3								
6	1,7 21,11	2,7 6,11	3,7					
7								
11								
21	1,6 21,26	2,6 6,26	3,6 11,26		7,6 11,26			
26	1,11 21,3	2,11 6,3	3,11		7,11 11,3			6,11 26,3
	0	1	2	3	6	7	11	21

stati distinguibili per almeno una configurazione d'ingresso

stati indistinguibili per ogni sequenza d'ingresso di lunghezza unitaria, ma distinguibili per almeno una sequenza d'ingresso di lunghezza superiore

Tabella di flusso non riducibile

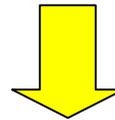
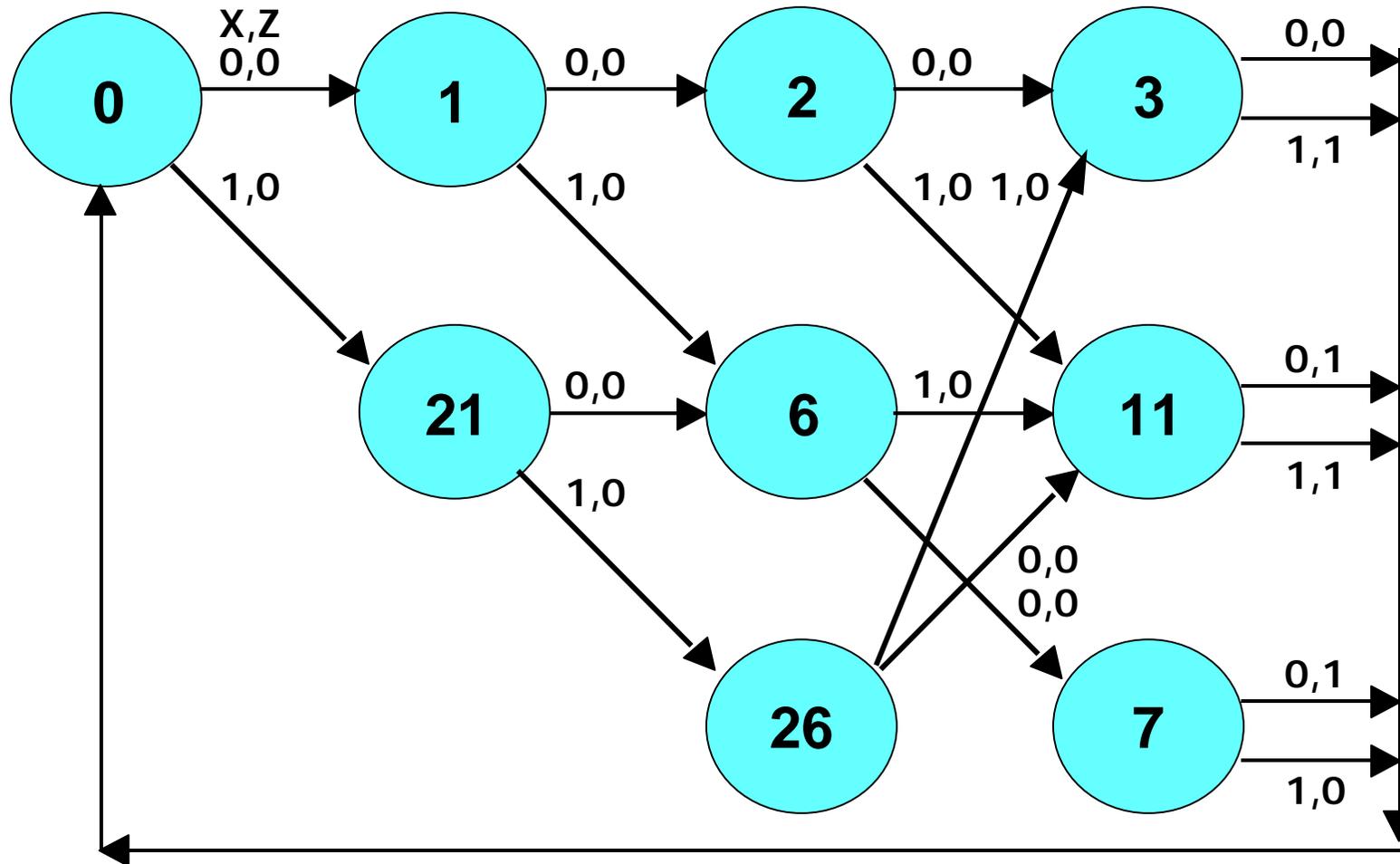


Diagramma degli stati

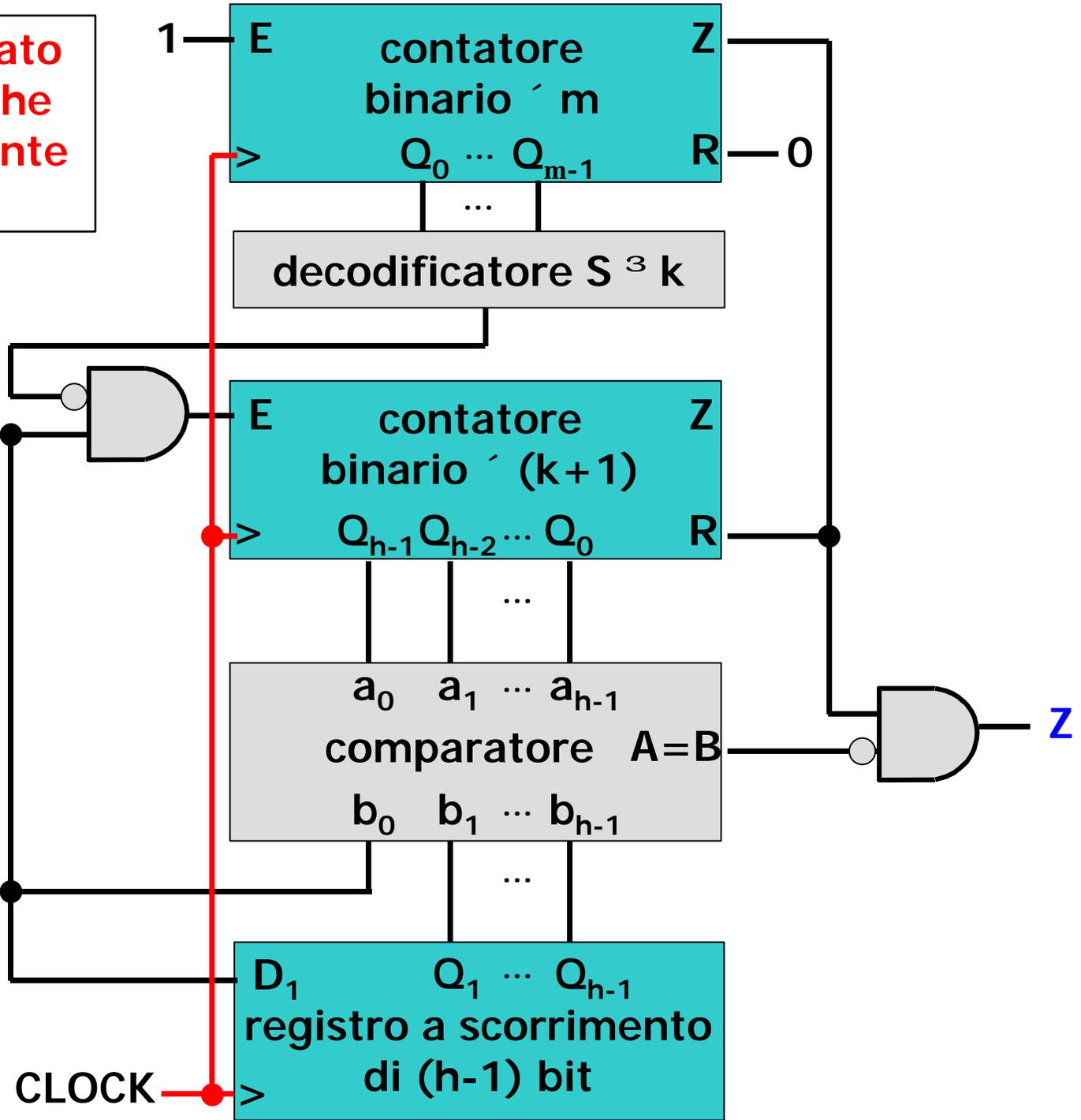
Diagramma degli stati



La rete in esame presenta, a regime, lo stesso comportamento di quella progettata nell'esercizio precedente !

Lo schema analizzato ha una struttura che può essere facilmente configurabile

per gestire parole di $m=k+h$ bit, costituite da un numero qualsiasi k di bit di informazione e da $h=\lceil \lg_2(k+1) \rceil$ bit di rilevazione di errore.



Esercizio 9

Una rete sequenziale sincrona è caratterizzata da un segnale di ingresso (X) e da un segnale di uscita (Z), entrambi sincroni. Attraverso l'ingresso X la rete riceve serialmente, ed una di seguito all'altra, parole di 32 bit. Ogni parola, costituita da otto cifre decimali rappresentate secondo il codice BCD, identifica una data in termini di anno ($A \equiv \{A_m A_c A_d A_u\}$), mese ($M \equiv \{M_d M_u\}$) e giorno ($G \equiv \{G_d G_u\}$). Le cifre identificative di ciascuna data sono ricevute secondo l'ordine $G_d, G_u, M_d, M_u, A_m, A_c, A_d, A_u$. I bit di ciascuna cifra sono ricevuti a partire da quello più significativo. La rete ha il compito di verificare per ciascuna data se l'anno è bisestile. Solo in tal caso, ed in corrispondenza dell'intervallo di ricezione dell'ultimo bit di una data, Z deve assumere il valore 1; in ogni altro intervallo di clock Z deve assumere il valore 0.

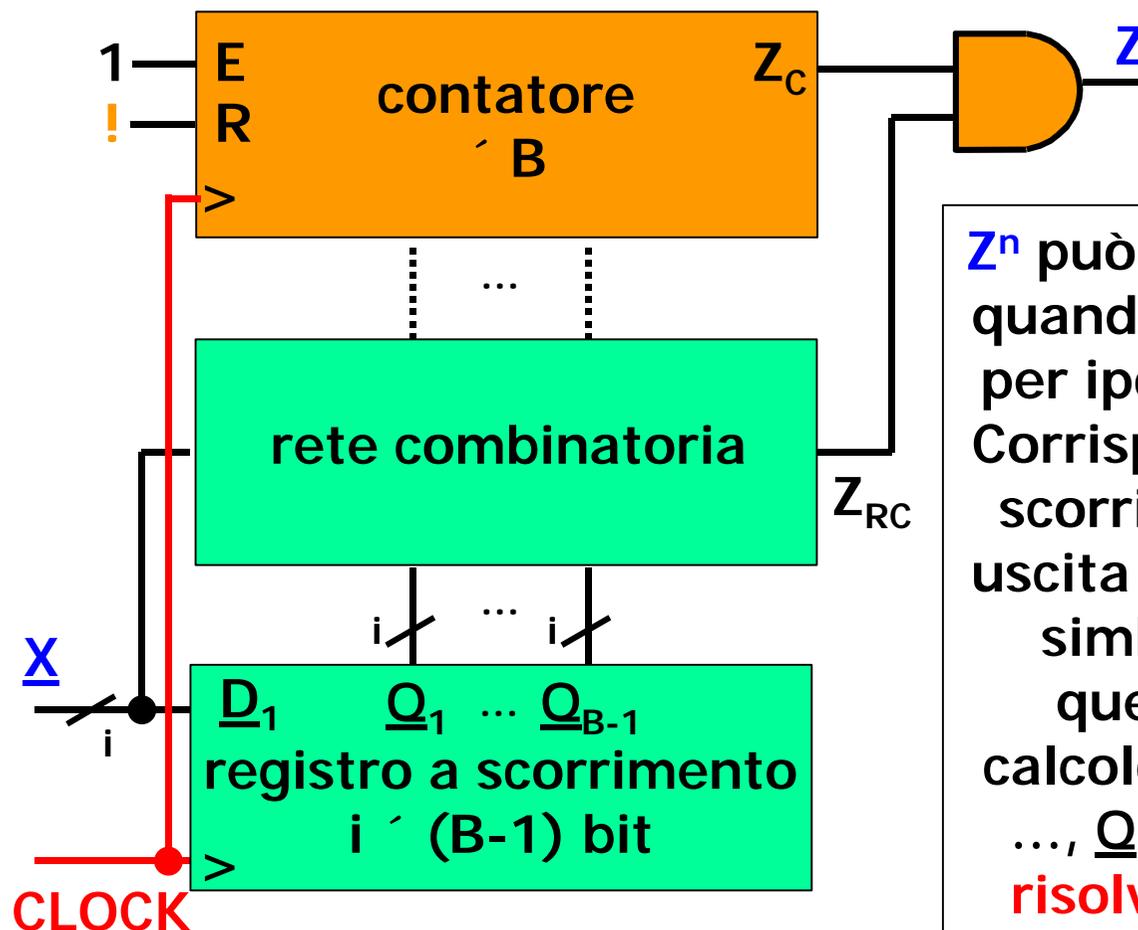
Un anno è bisestile se è divisibile per quattro, a meno che sia il primo anno di un secolo non multiplo di quattro:

$$(A_{\text{mod } 4} = 0) \text{ and not } ((A_{\text{mod } 100} = 0) \text{ and } ((A / 100)_{\text{mod } 4} \neq 0))$$

ovvero:

$$((A_d A_u)_{\text{mod } 4} = 0) \text{ and } ((A_d \neq 0) \text{ or } (A_u \neq 0) \text{ or } ((A_m A_c)_{\text{mod } 4} = 0))$$

Per risolvere problemi di questo tipo si può applicare il modello introdotto nel precedente esempio: una **rete di sincronizzazione** (basata su un contatore opportunamente inizializzato e con base di conteggio B pari al numero di simboli per parola) provvede a partizionare in parole la sequenza di simboli applicati in ingresso; una **rete di elaborazione** (strutturata come macchina con ampiezza di memoria finita) svolge il compito di gestire ciascuna parola.

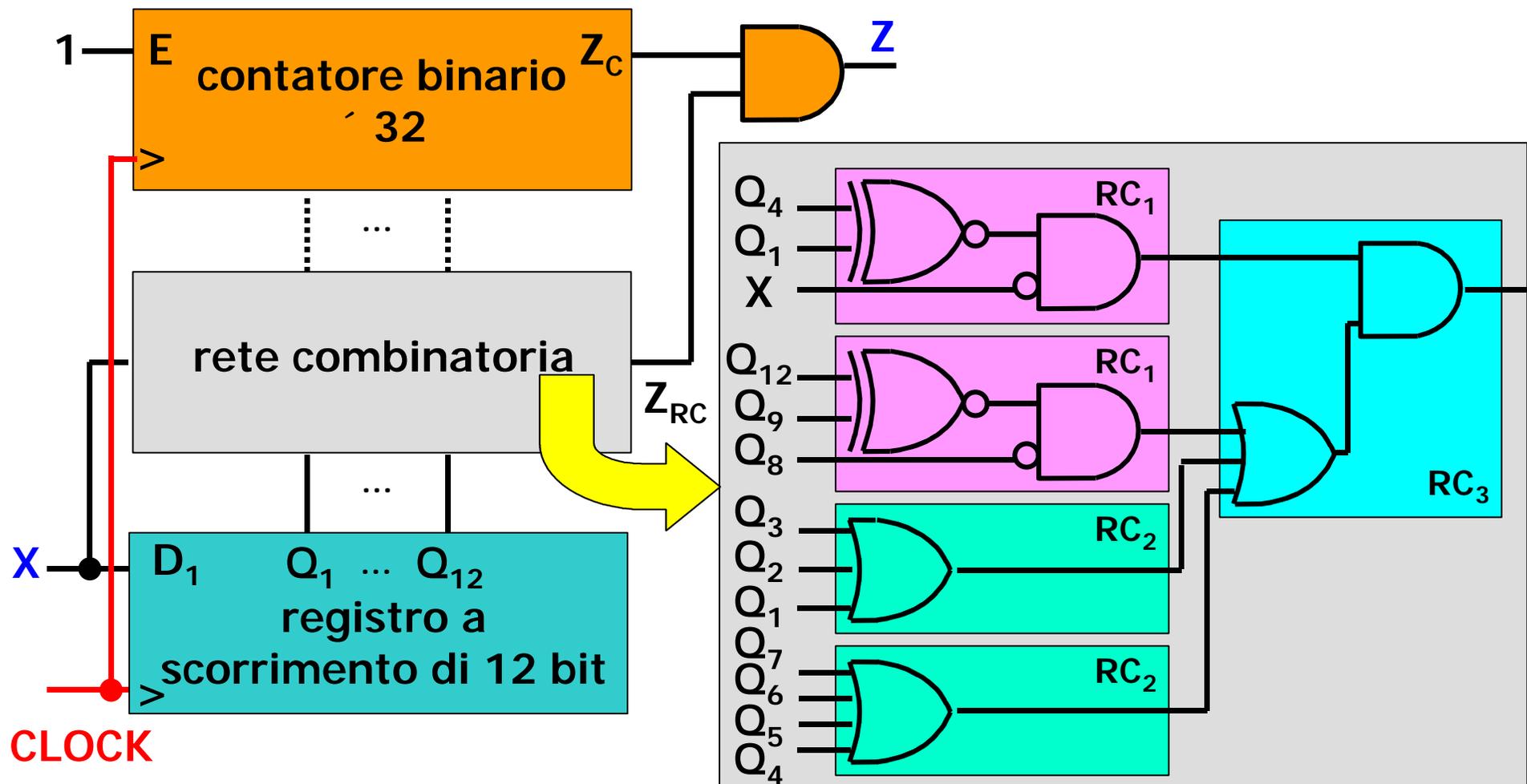


Z^n può valere 1 solo se $Z_c^n = 1$, cioè quando il simbolo X^n in ingresso è per ipotesi l'ultimo di una parola. Corrispondentemente il registro a scorrimento rende disponibili in uscita ordinatamente i precedenti simboli di una parola (o tutti quelli significativi ai fini del calcolo di Z^n): $Q_1^n = X^{n-1}$, $Q_2^n = X^{n-2}$, ..., $Q_{B-1}^n = X^{n-B+1}$. **Il problema da risolvere è ora combinatorio !!!**

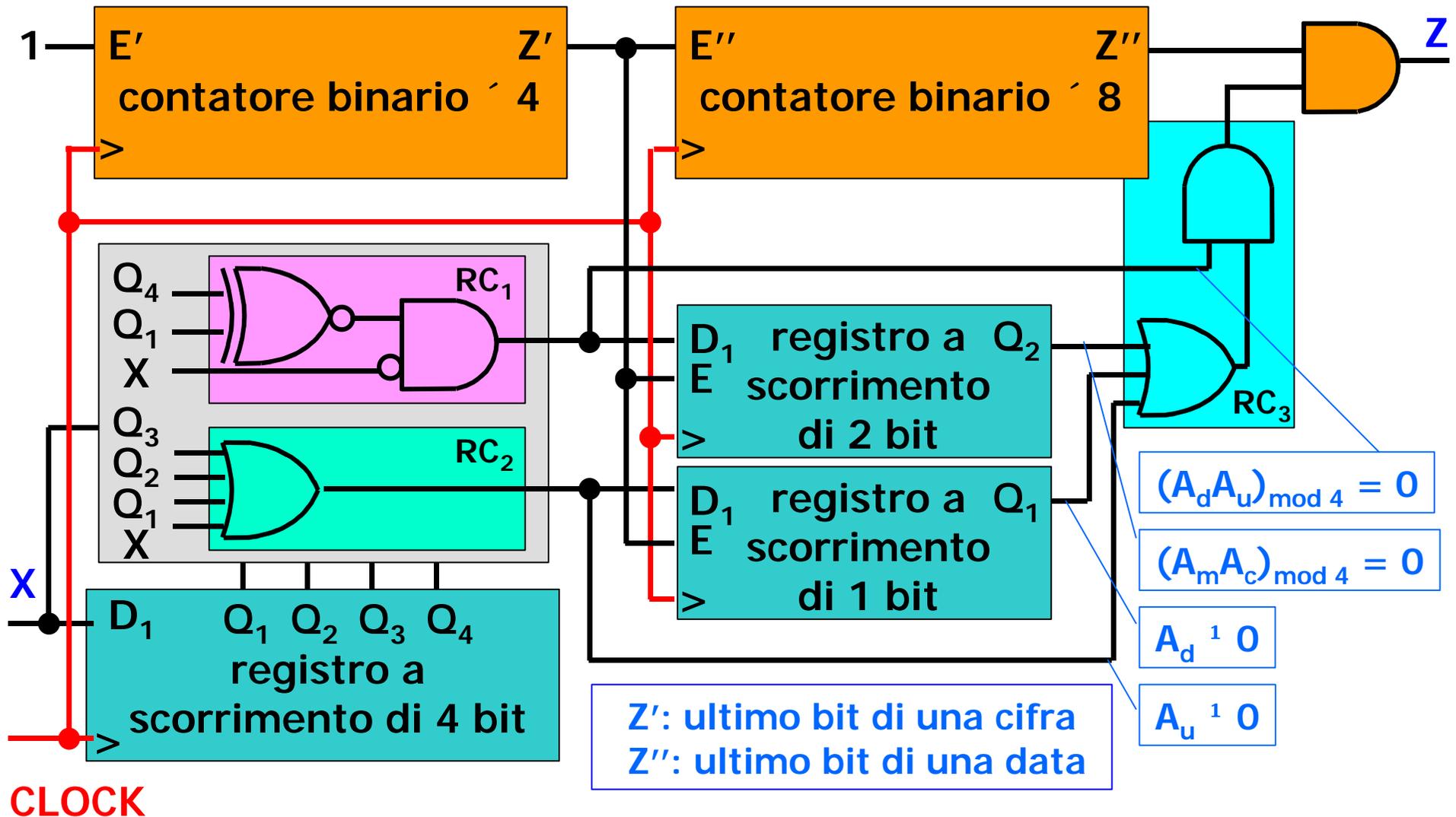
Nel caso specifico basta selezionare un contatore ~ 32 ed un registro a scorrimento di soli 12 bit, poiché né i quattro bit di G_d, G_u, M_d, M_u , né i tre bit più significativi di A_m intervengono nella relazione di ingresso-uscita della rete combinatoria :

$$Z_{RC} = A_{u0}' (A_{u1} \dot{\wedge} A_{d0})'$$

$$(A_{u1} + A_{u2} + A_{u3} + A_{d0} + A_{d1} + A_{d2} + A_{d3} + A_{c0}' (A_{c1} \dot{\wedge} A_{m0})')$$



Facendo riferimento ai modelli già discussi a proposito delle macchine con ampiezza di memoria finita, è possibile identificare una soluzione più efficiente, che evita di replicare inutilmente le due reti combinatorie RC_1 e RC_2 , e nel contempo permette di ridurre la capacità di memoria richiesta (58.3 %).



Esercizio 10

Una rete sequenziale sincrona è caratterizzata da due segnali di ingresso (E, X) e da un segnale di uscita (Z), tutti sincroni rispetto al clock della rete stessa. Attraverso l'ingresso X la rete riceve serialmente parole di k bit. Il segnale E, attivo (valore logico 1) per k intervalli di clock, segnala la fase di ricezione di ciascuna parola. L'uscita Z della rete può assumere il valore logico 1 soltanto in corrispondenza dell'intervallo di clock immediatamente successivo a quello di ricezione dell'ultimo bit di ciascuna parola, e ciò se la parola comprende almeno tre 1 consecutivi, ma non due 0 consecutivi. Si identifichi:

- l'automa minimo della rete secondo il modello di Mealy;
- la realizzazione della rete con FF-JK e gate elementari.

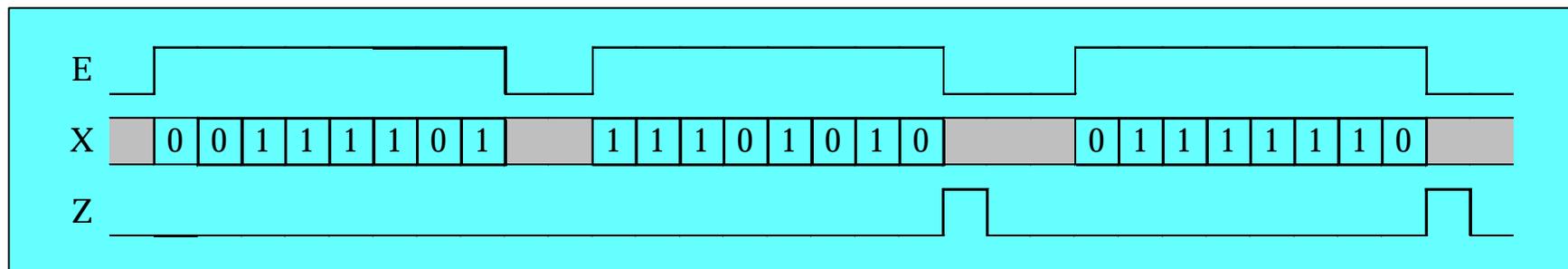
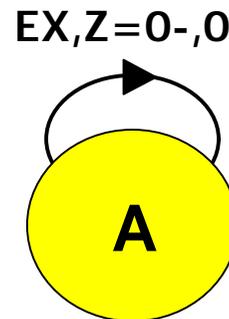
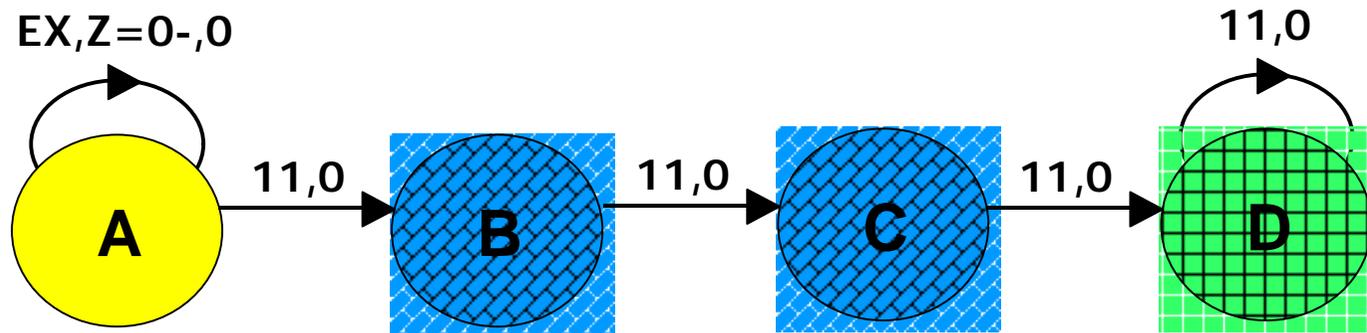


Diagramma degli stati (modello di Mealy)

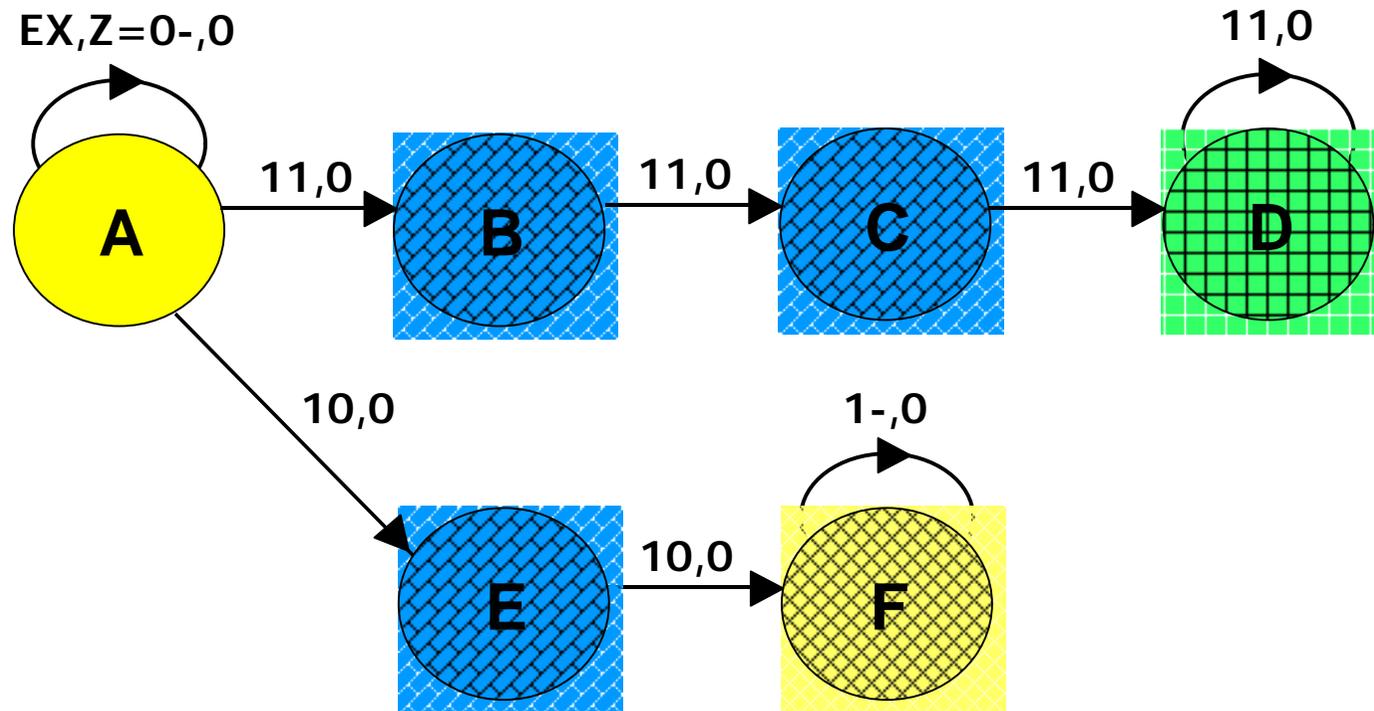
Il diagramma degli stati può essere costruito a partire dalla situazione in cui si trova ad operare la rete allorché in attesa di una parola da elaborare ($E=0, X=-, Z=0$).



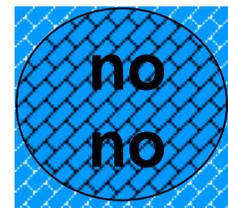
Supponiamo ora che, quando E si attiva, la parola in ingresso presenti subito la sequenza desiderata, ovvero che X assuma il valore 1 per (almeno) tre intervalli di clock consecutivi.



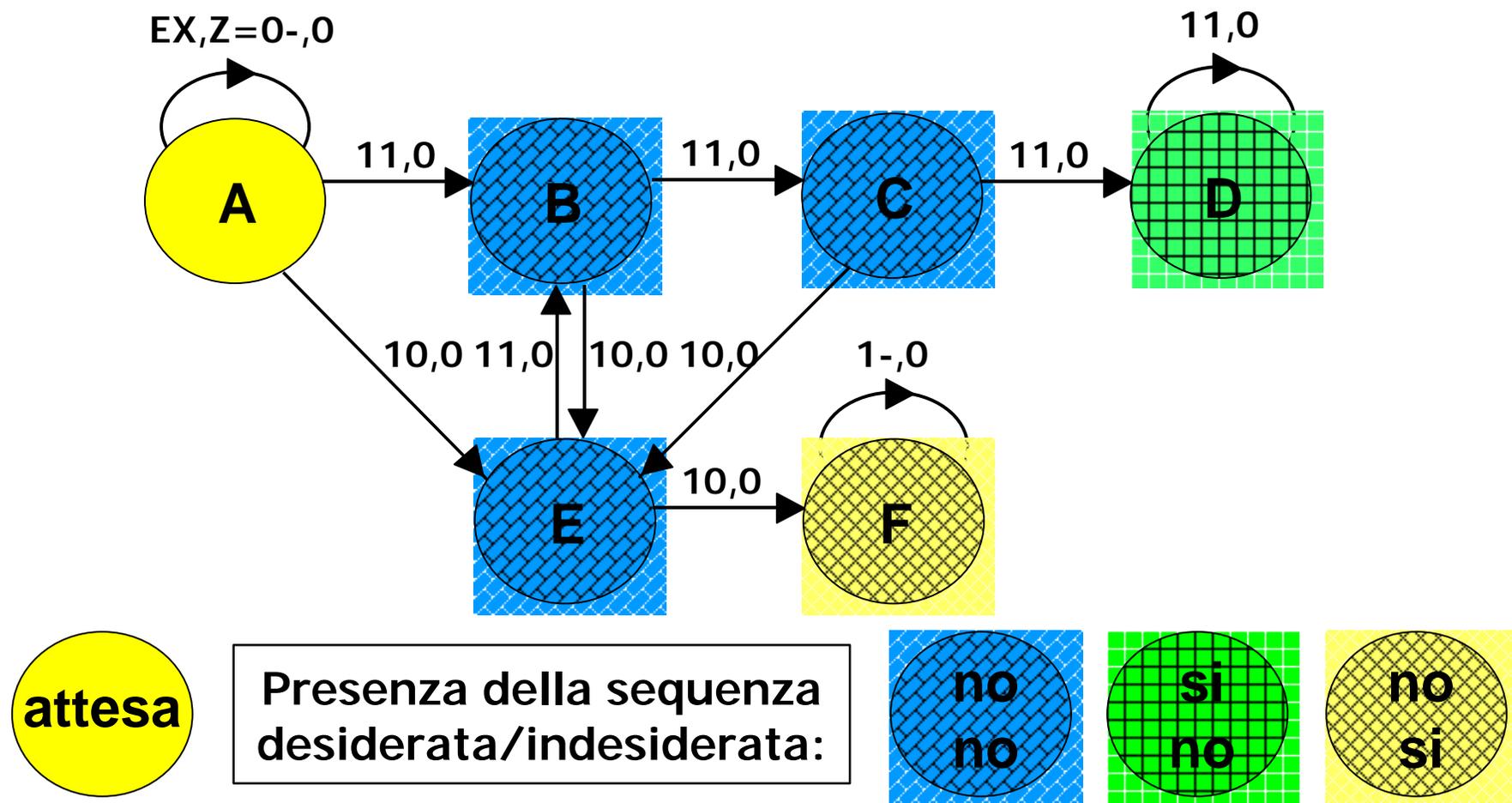
Supponiamo ora che, quando E si attiva, la parola in ingresso presenti subito la sequenza indesiderata, ovvero che X assuma il valore 0 per due intervalli di clock consecutivi. In tal caso i successivi valori di X nell'ambito di una parola risultano influenti.



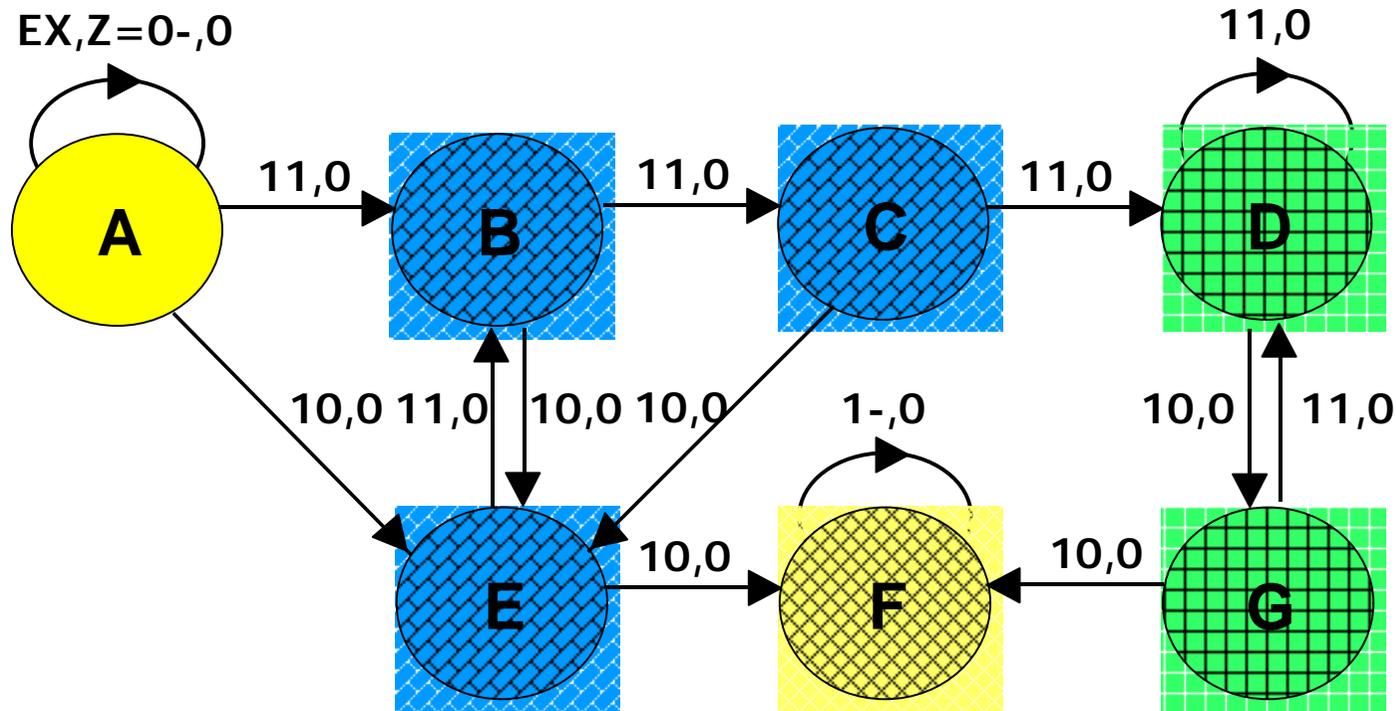
Presenza della sequenza desiderata/indesiderata:



Supponiamo ora che nella parola in corso di elaborazione non sia stata ancora rilevata né la sequenza desiderata, né quella indesiderata, e che tale situazione venga confermata dai successivi simboli applicati in ingresso. Se dopo uno (stato B) o due 1 consecutivi (stato C) si presenta il simbolo 0, da tali stati occorre transitare nello stato E. Se dopo uno 0 (stato E) si presenta il simbolo 1, da tale stato occorre transitare nello stato B.



Supponiamo ora che nella parola in corso di elaborazione sia stata rilevata la sequenza desiderata, ma non ancora quella indesiderata. A fronte di un primo 0, dallo stato D occorre transitare in un nuovo stato G (non in E!). Da G si transiterà poi in D o in F a seconda che il successivo simbolo sia 1 o ancora 0, rispettivamente.



attesa

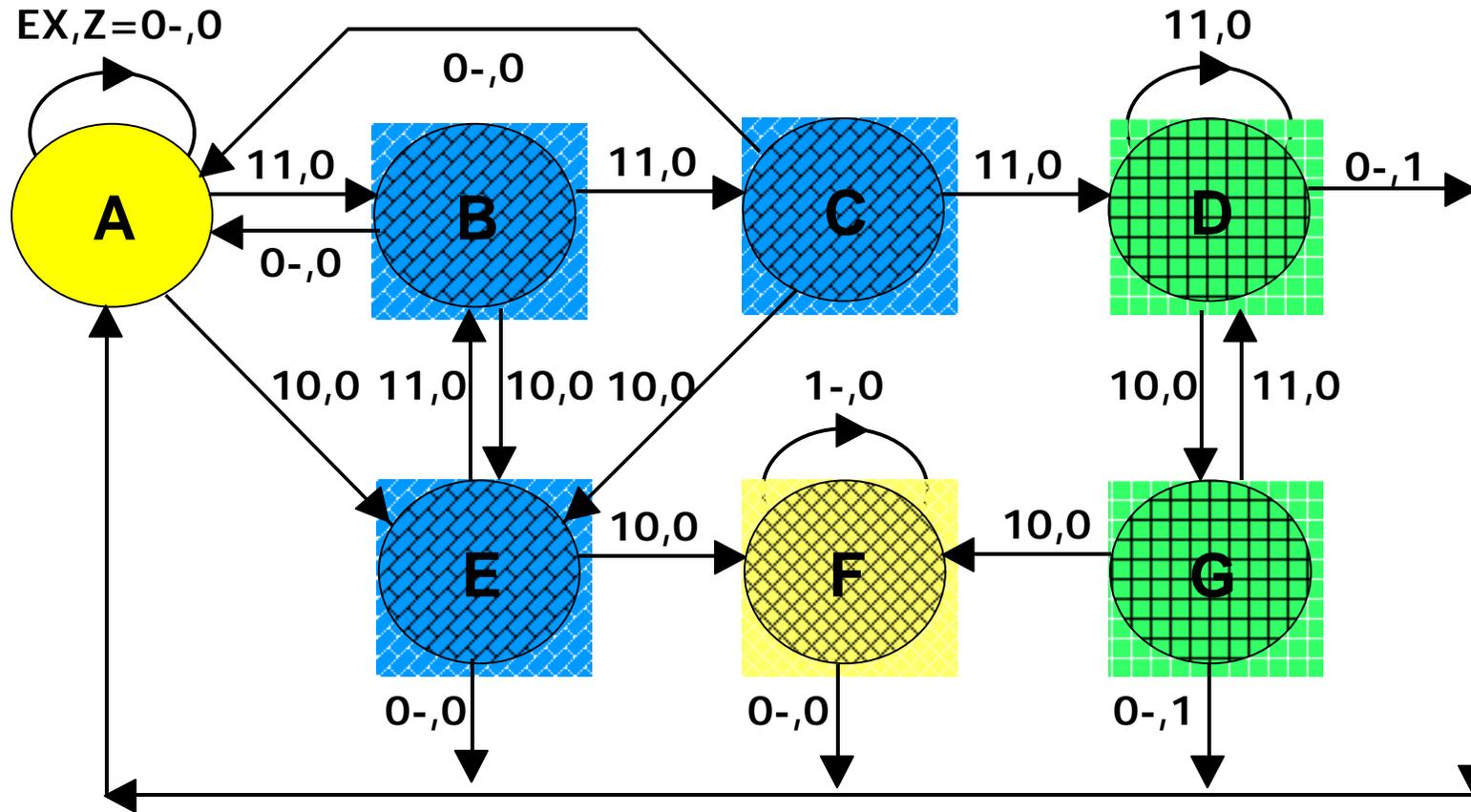
Presenza della sequenza desiderata/indesiderata:

no
no

si
no

~~no~~
si

Occorre infine prevedere, quando E si disattiva, il rientro nello stato di attesa A. Il valore da attribuire corrispondentemente all'uscita Z discende immediatamente, per ogni stato, dal significato associato allo stato stesso.



Presenza della sequenza desiderata/indesiderata:

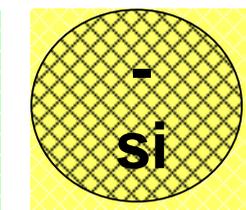
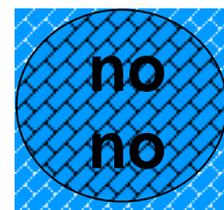


Tabella di flusso

	E X			
	00	01	11	10
A	A,0	A,0	B,0	E,0
B	A,0	A,0	C,0	E,0
C	A,0	A,0	D,0	E,0
D	A,1	A,1	D,0	G,0
E	A,0	A,0	B,0	F,0
F	A,0	A,0	F,0	F,0
G	A,1	A,1	D,0	F,0

**tabella non
riducibile**

Tabella delle transizioni

$(y_1y_2y_3 \setminus EX)^n$	00	01	11	10
A \equiv 000	000,0	000,0	001,0	100,0
B \equiv 001	000,0	000,0	010,0	100,0
D \equiv 011	000,1	000,1	011,0	111,0
C \equiv 010	000,0	000,0	011,0	100,0
E \equiv 100	000,0	000,0	001,0	101,0
F \equiv 101	000,0	000,0	101,0	101,0
G \equiv 111	000,1	000,1	011,0	101,0
110	---,-	---,-	---,-	---,-

$(y_1y_2y_3)^{n+1}, Z^n$

Rete combinatoria di uscita

$(y_1 y_2 y_3 \setminus EX)^n$	00	01	11	10
A \equiv 000	0	0	0	0
B \equiv 001	0	0	0	0
D \equiv 011	1	1	0	0
C \equiv 010	0	0	0	0
E \equiv 100	0	0	0	0
F \equiv 101	0	0	0	0
G \equiv 111	1	1	0	0
110	-	-	-	-

Z^n

$$Z^n = (y_2 y_3 E')^n$$

Rete combinatoria di aggiornamento dello stato

$(y_1 y_2 y_3 \setminus EX)^n$	00	01	11	10
A \equiv 000	0	0	0	1
B \equiv 001	0	0	0	1
D \equiv 011	0	0	0	1
C \equiv 010	0	0	0	1
E \equiv 100	0	0	0	1
F \equiv 101	0	0	1	1
G \equiv 111	0	0	0	1
110	-	-	-	-

y_1^{n+1}

$$J_1^n = (E X')^n$$

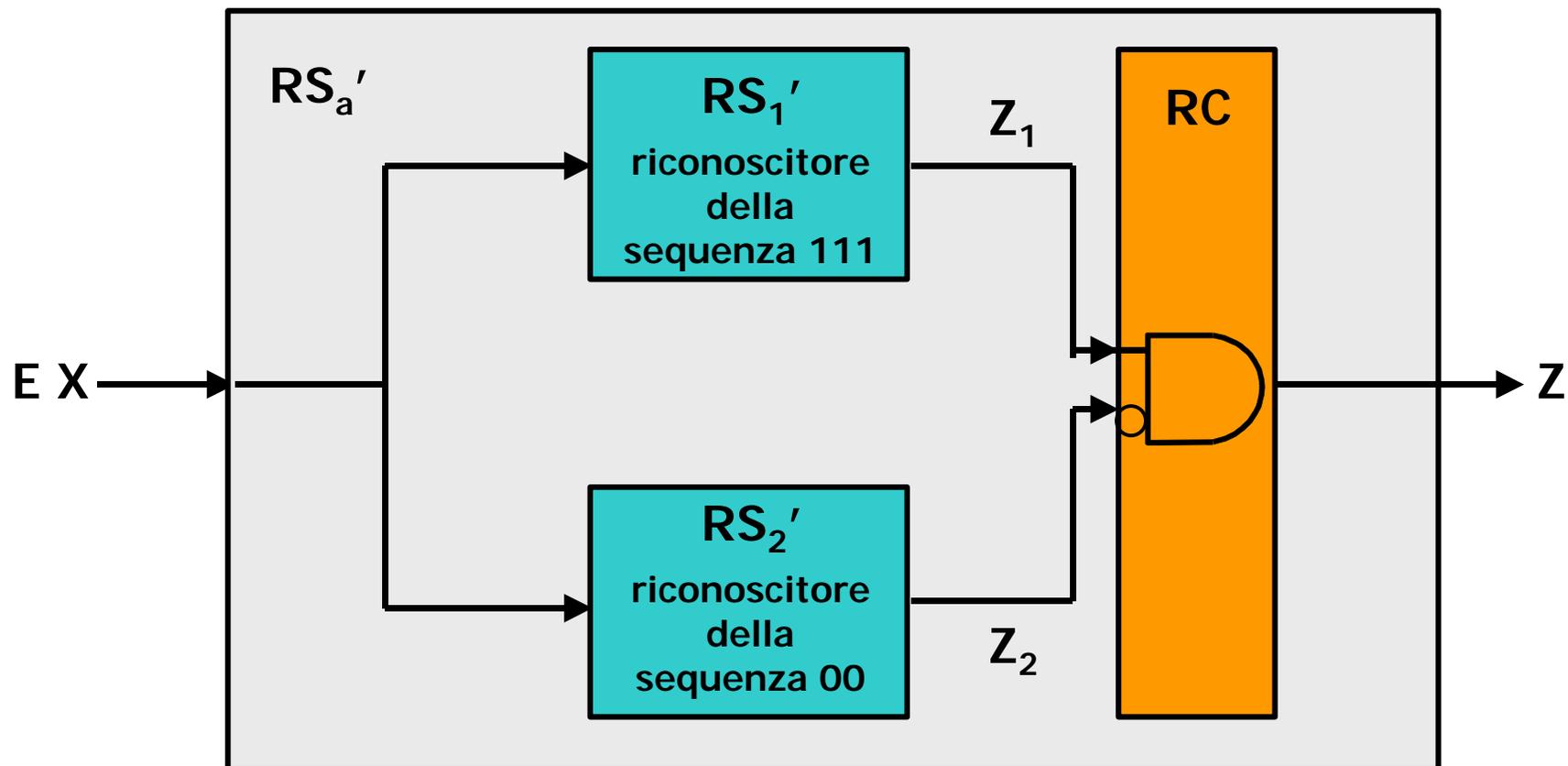
$$K_1^n = (E' + y_3' X + y_2 X)^n$$

similmente

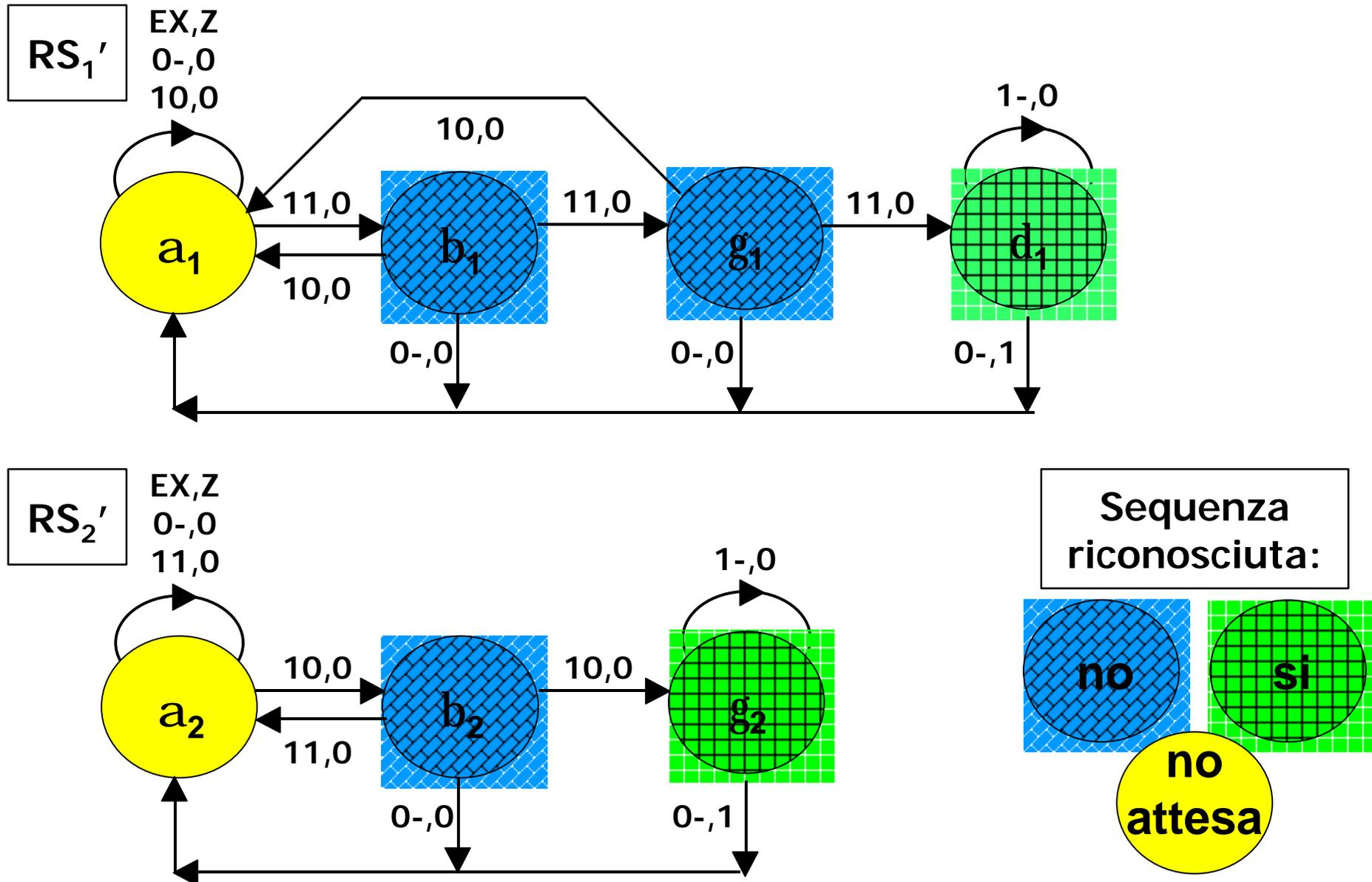
$$J_2^n, K_2^n, J_3^n, K_3^n$$

Alternativa alla realizzazione precedente (RS_a), è la seguente (RS_a') che deriva dall'applicazione del:

Principio di decomposizione: una rete sequenziale con N stati può essere decomposta in due reti sequenziali più semplici con N_1 e N_2 stati rispettivamente ($N_1 < N, N_2 < N, N_1 + N_2 \leq N$).

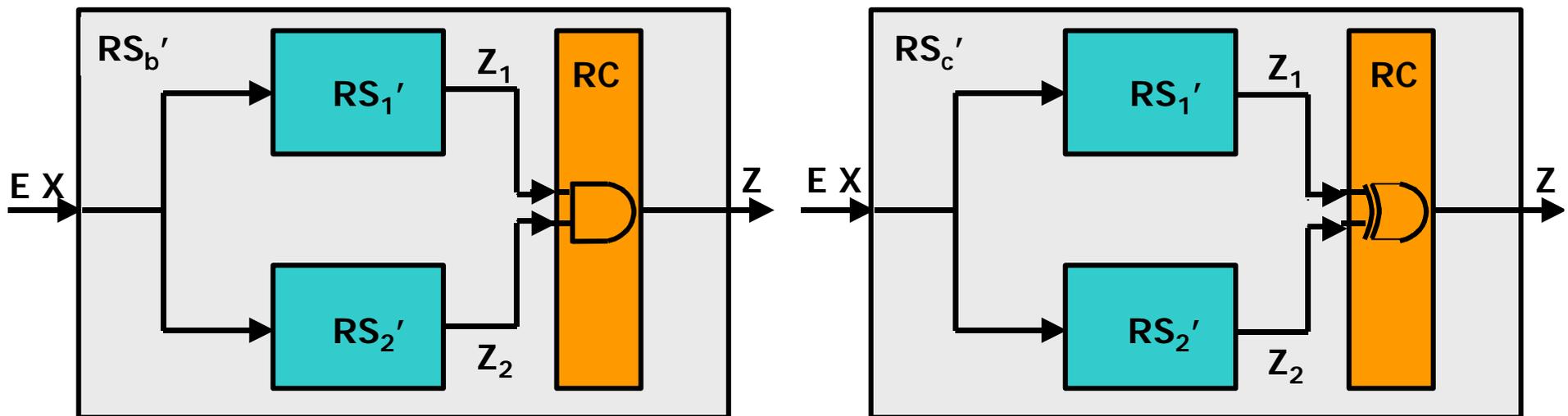


Il diagramma degli stati dei 2 riconoscitori di sequenza per la realizzazione della rete secondo il modello RS_a'



L'applicazione del principio di decomposizione consente non solo di semplificare e meglio strutturare il progetto, ma anche di riutilizzarne parti a fronte di eventuali variazioni delle specifiche.

Se, ad esempio, la specifica di progetto dovesse variare, prevedendo che l'uscita Z si attivi qualora una parola contenga non più (a) solo la sequenza 111 e non la sequenza 00, bensì (b) entrambe le sequenze, oppure (c) una qualunque delle due, la soluzione RS_a' precedentemente identificata applicando il principio di decomposizione sarebbe ampiamente riutilizzabile, essendo sufficiente modificare soltanto la rete combinatoria di uscita:

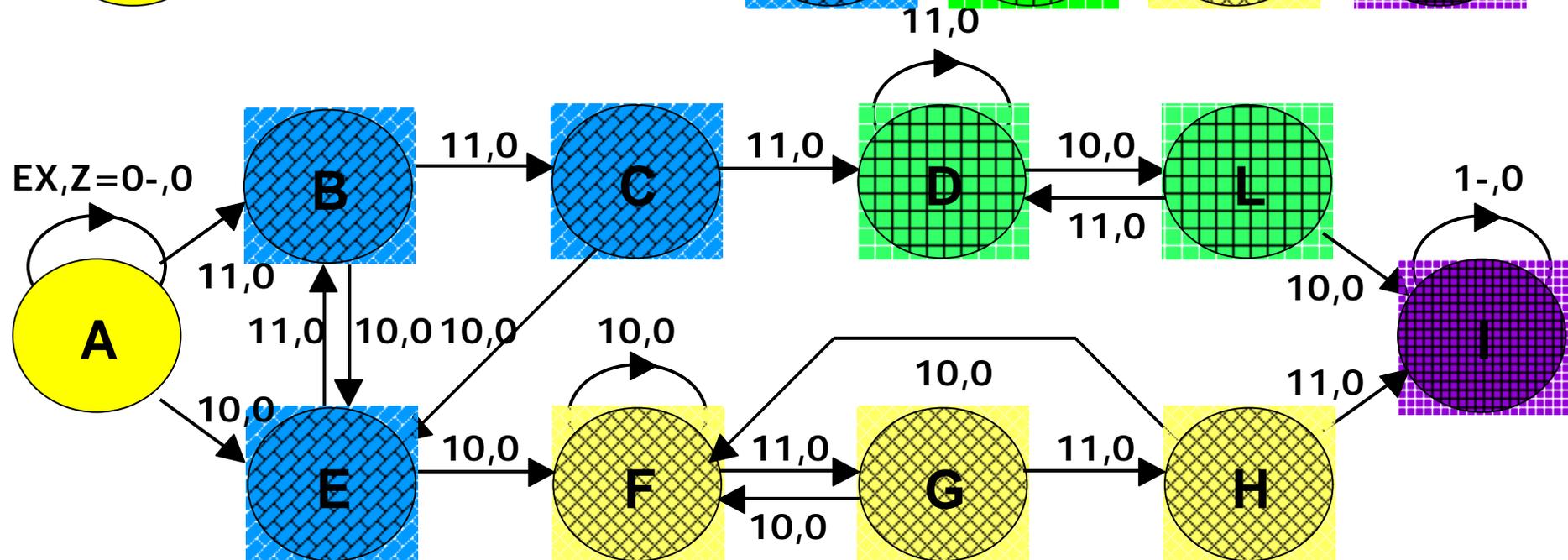
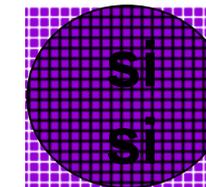


L'approccio convenzionale al contrario non consente il riutilizzo della soluzione RS_a ai fini della realizzazione delle nuove reti RS_b e RS_c .

I diagrammi degli stati per le reti RS_b e RS_c (... e RS_a)

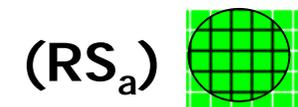
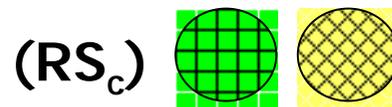


Presenza della sequenza 111/00:

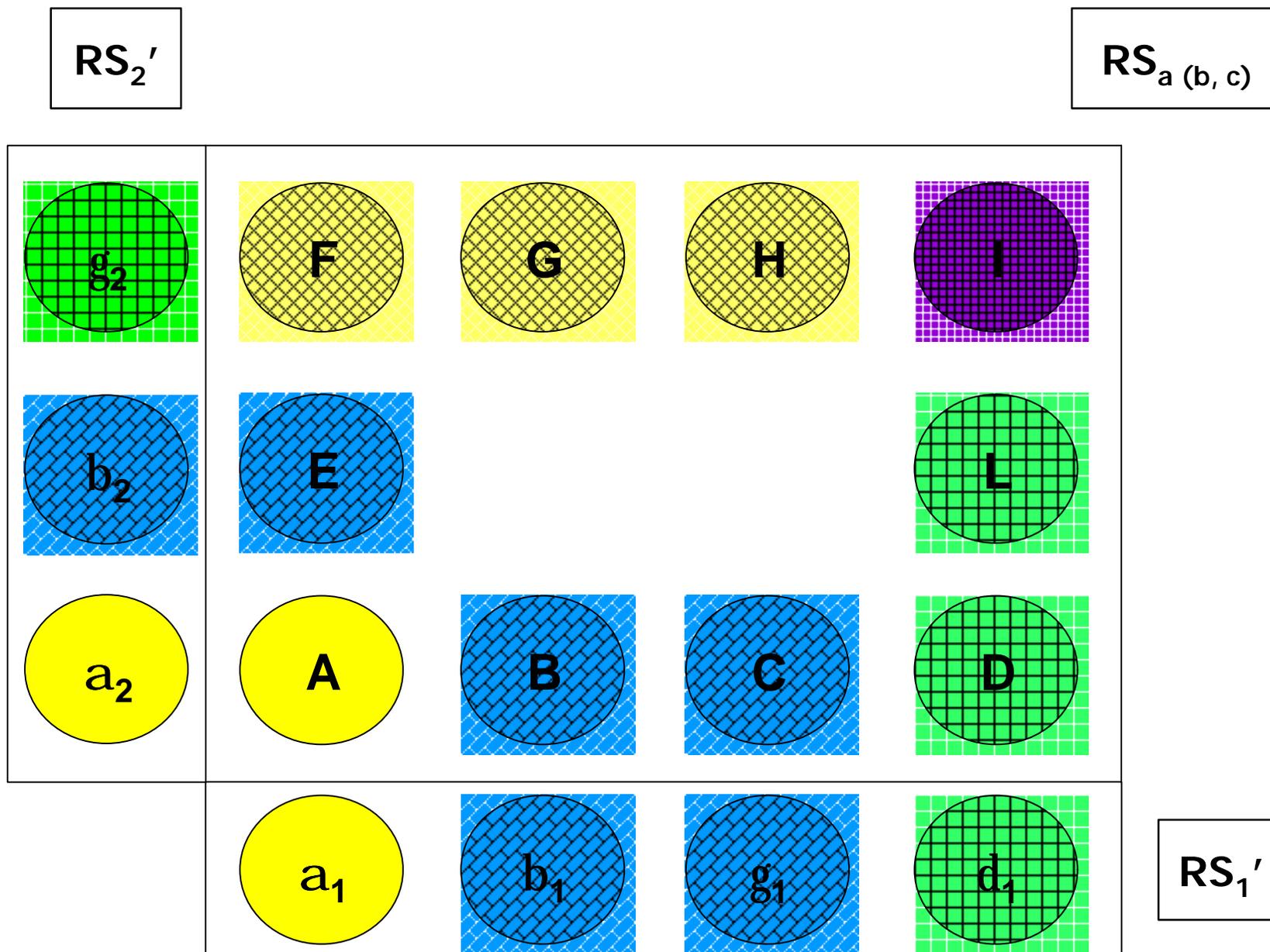


Occorre infine prevedere la transizione in A da ogni altro stato per $EX=0-$. Il valore da attribuire corrispondentemente a Z dipende dalla specifica di progetto, ovvero $Z=1$ se:

Il diagramma può essere utilizzato anche per la sintesi di RS_a . In tal caso $Z=1$ se:

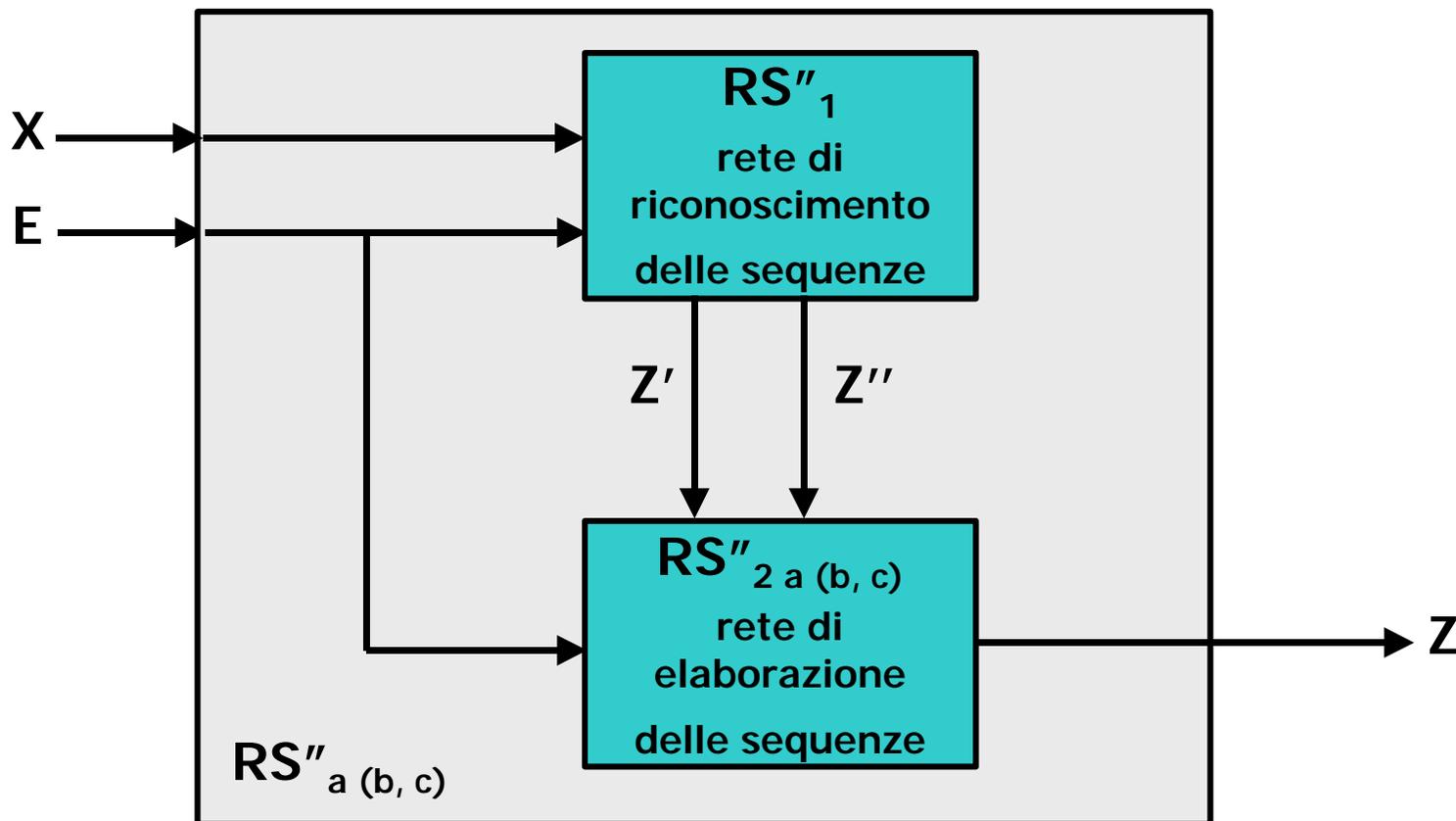


La corrispondenza tra gli stati di RS_1' e RS_2' e gli stati di RS_a (o di RS_b o di RS_c)



L'applicazione del principio di decomposizione, ovvero la strutturazione di una rete in sottoreti più semplici opportunamente interconnesse, in generale non è univoca.

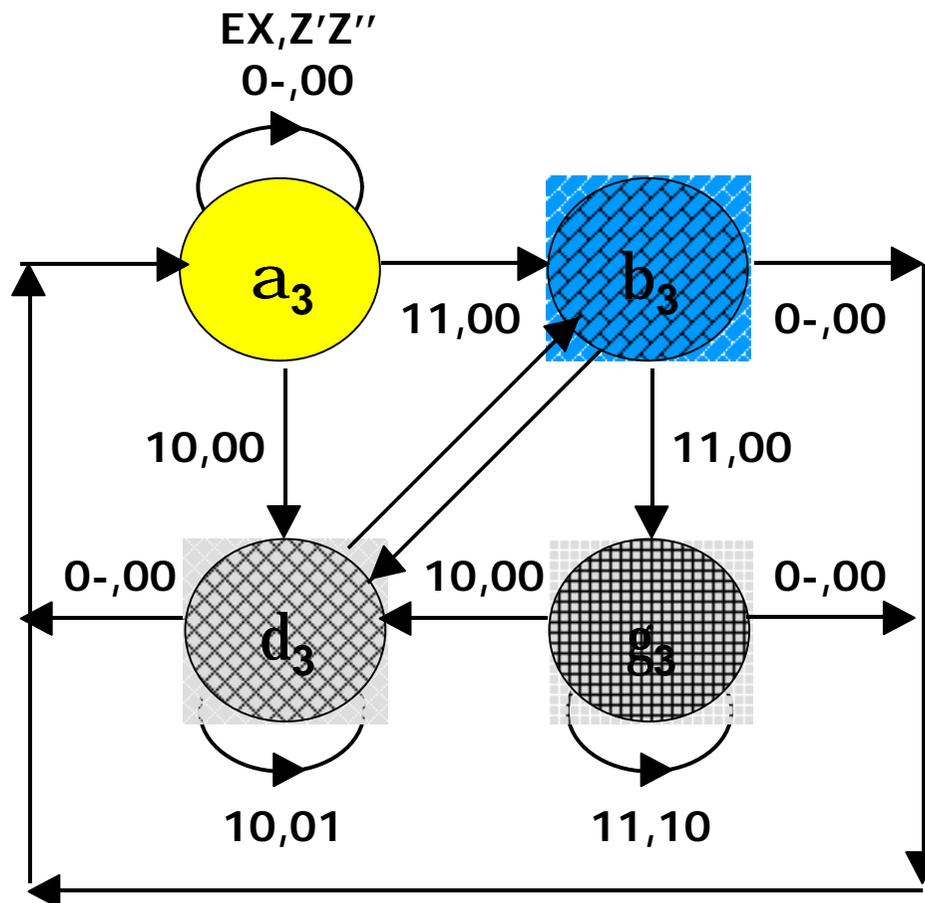
Si potrebbe infatti adottare anche il seguente modello, che prevede due sottoreti sequenziali contraddistinte da funzionalità più ortogonali rispetto al modello precedente:



La rete di riconoscimento delle sequenze (RS''_1)

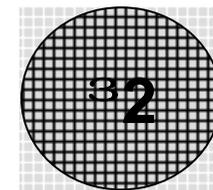
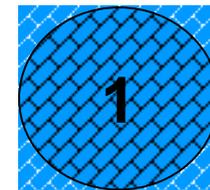
Tale rete ha il compito di attivare Z' o Z'' ogni qualvolta rileva in una parola la stringa 111 o 00, rispettivamente. Il comportamento della rete è invariante rispetto alla regola (a, b, c) di elaborazione delle sequenze selezionate.

Diagramma degli stati

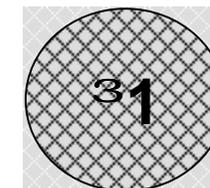


attesa

numero di 1 consecutivi
in precedenza rilevati:



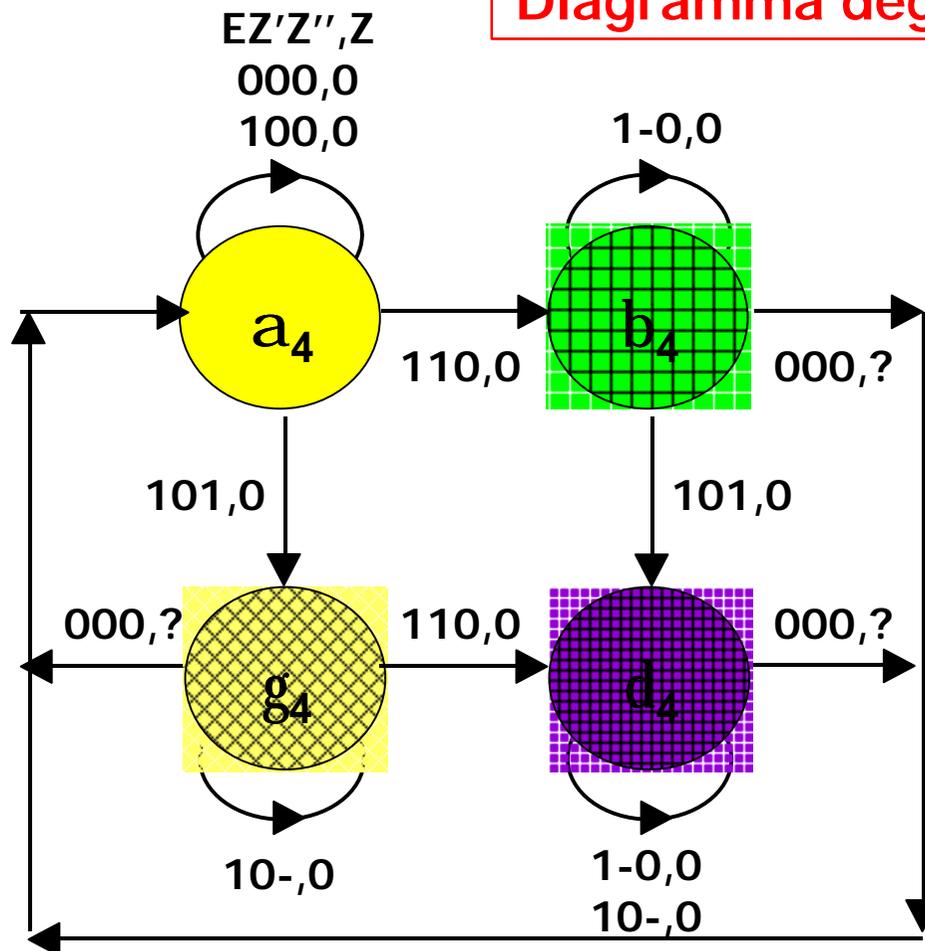
numero di 0 consecutivi
in precedenza rilevati:



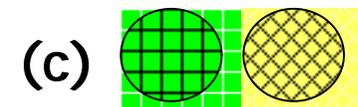
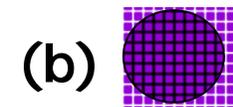
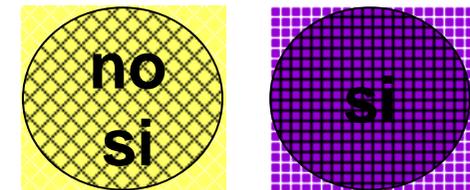
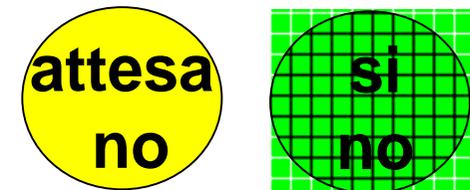
Le reti di elaborazione delle sequenze ($RS''_{2a(b,c)}$)

Tali reti hanno il compito di generare il segnale di uscita Z in accordo alla assegnata regola di gestione delle due sequenze. Il comportamento delle reti non dipende dalle sequenze selezionate.

Diagramma degli stati

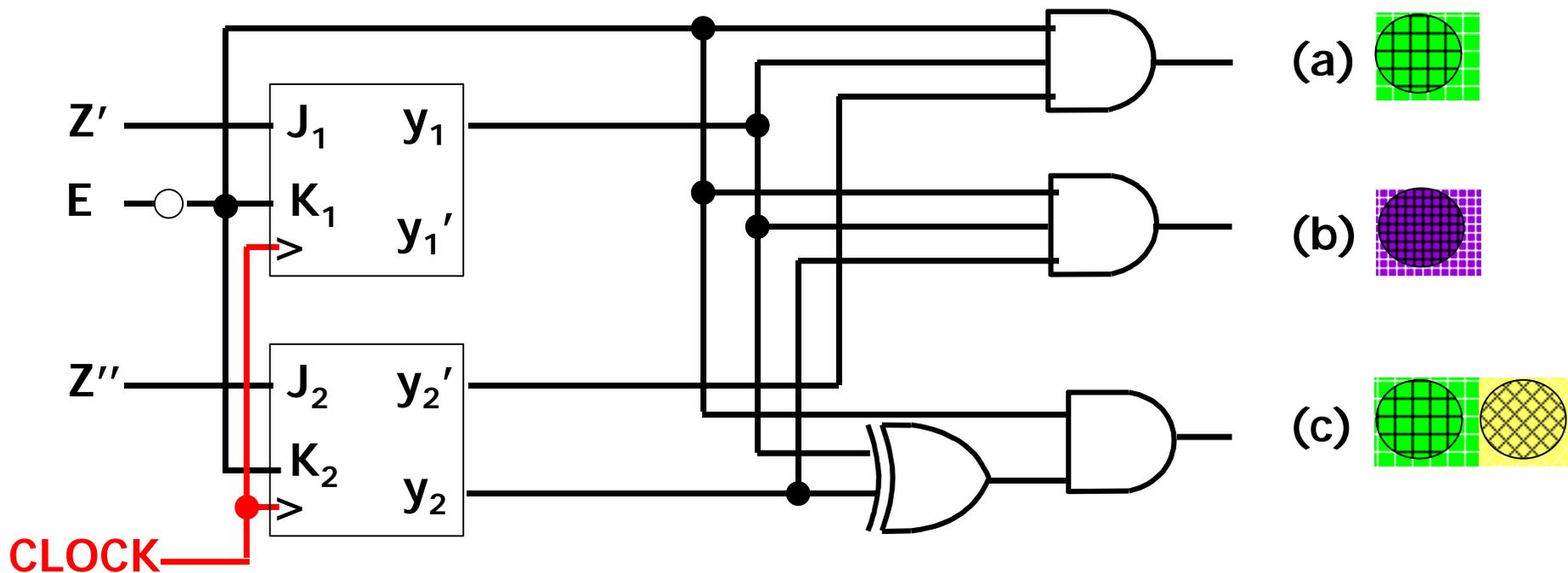
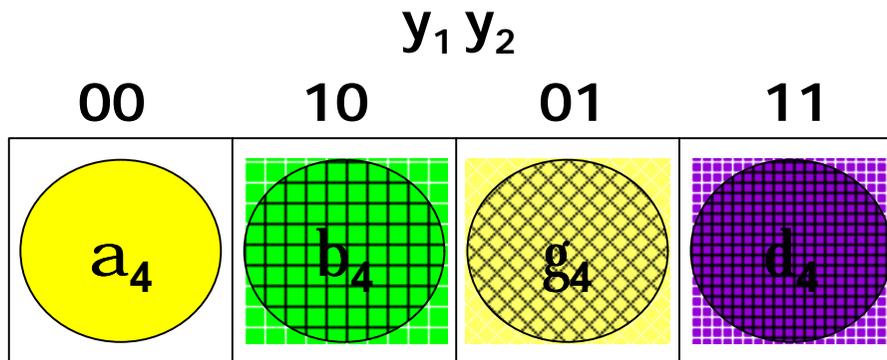


Presenza delle sequenze 111/00:



Le reti di elaborazione delle sequenze ($RS''_{2a(b,c)}$)

Schema logico



La corrispondenza tra gli stati di RS''_1 e RS''_{2a} (o RS''_{2b} o RS''_{2c}) e gli stati di RS_a (o RS_{2b} o RS_{2c})

