

Circuiti sequenziali sincroni

Capitolo 7

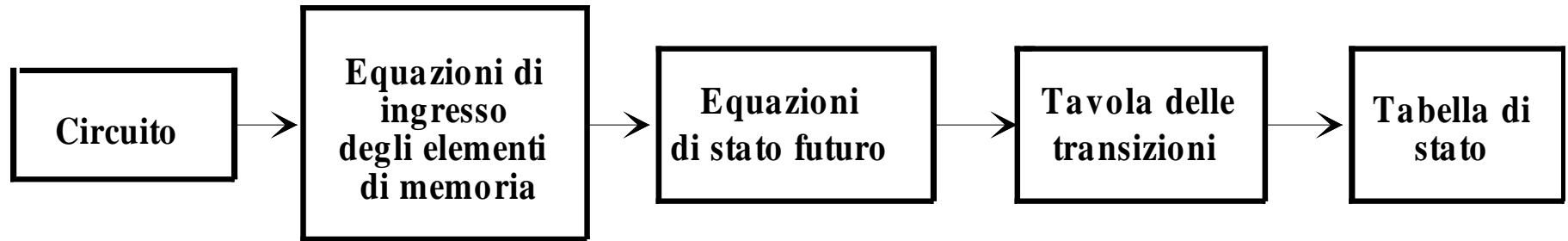


Introduzione

- Differenze fondamentali con i c.s.a.
 - I segnali di stato sono sincronizzati (clock)
 - Presenza di Flip Flop (JK, SR, T, D) per mantenere memoria dello stato
 - Assenza di Alee o Corse
 - Eventuale presenza di Reset
 - Sviluppo in forma di strutture particolari (a memoria finita)



Analisi



Esempio di Analisi (cont.)

Equazioni di stato Futuro

$$y_1^{n+1} = x^n \cdot \overline{y_1^n} + x^n \cdot y_1^n = x^n$$

$$y_2^{n+1} = (x^n + y_1^n) \cdot \overline{y_2^n} + [(x^n + y_1^n) \cdot \overline{y_2^n}] y_2^n = (x^n + y_1^n) \cdot \overline{y_2^n}$$

Equazione di uscita

$$y_2^{n+1} = z$$

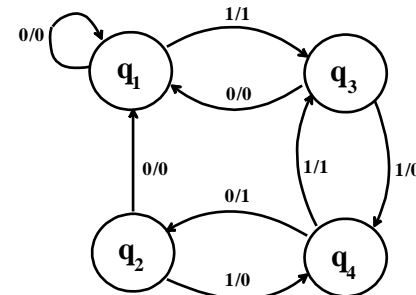
Tavola di transizione

		x^n	
		0	1
$y_1^n y_2^n$	00	00/0	11/1
	01	00/0	10/0
	11	00/0	10/0
	10	01/1	11/1

$y_1^{n+1} y_2^{n+1} / z^n$

Tavola e diagramma di stato

		Ingresso	
		$x^n=0$	$x^n=1$
Stato	q_1	$q_1, 0$	$q_3, 1$
	q_2	$q_1, 0$	$q_4, 0$
	q_3	$q_1, 0$	$q_4, 0$
	q_4	$q_2, 1$	$q_3, 1$



(a)

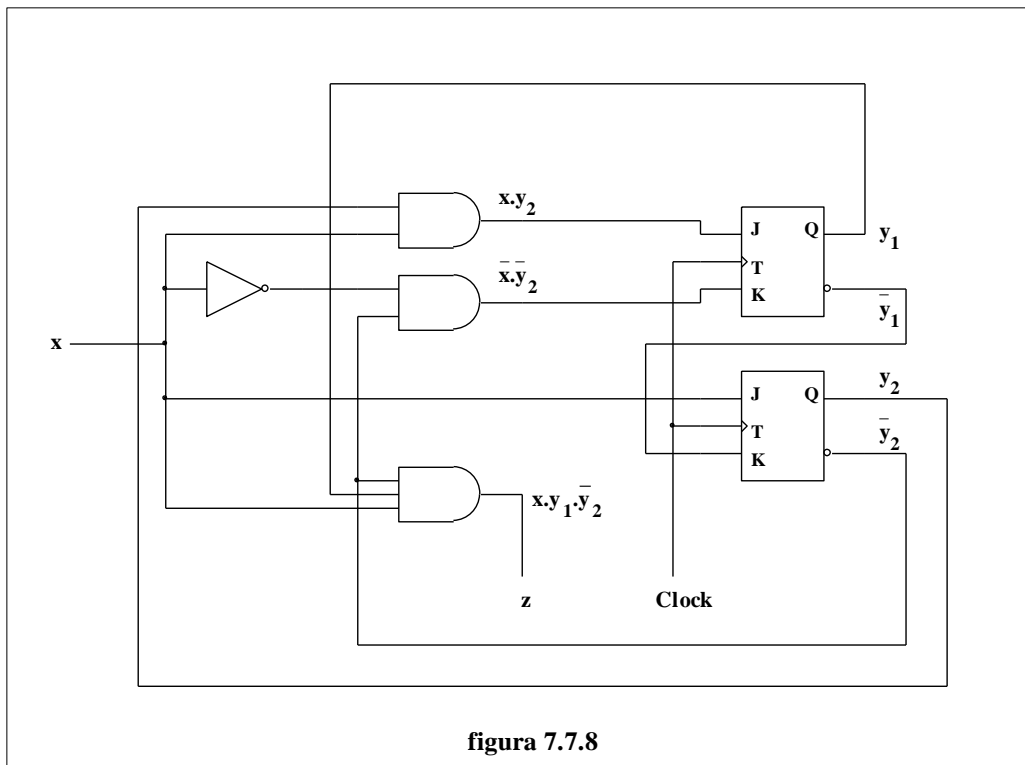
(b)

figura 7.2.2



Analisi

■ Esempio 2



Equazioni di ingresso dei FF

$$J_1 = x \cdot y_2$$

$$K_1 = \bar{x} \cdot \bar{y}_2$$

$$J_2 = x$$

$$K_2 = \bar{y}_1$$

Equazioni di stato Futuro

$$\begin{aligned} y_1^{n+1} &= J_1 \bar{y}_1^n + \bar{K}_1 y_1^n = \\ &= x \cdot y_2^n \cdot \bar{y}_1^n + (x + y_2^n) y_1^n \end{aligned}$$

$$\begin{aligned} y_2^{n+1} &= J_2 \bar{y}_2^n + \bar{K}_2 y_2^n = \\ &= x \cdot \bar{y}_2^n + y_1^n \cdot y_2^n \end{aligned}$$



Analisi, Esempio 2 (cont.)

Equazioni di stato Futuro e di uscita

$$\begin{aligned}y_1^{n+1} &= J_1 \overline{y_1^n} + \overline{K_1} y_1^n = \\ &= x \cdot y_2^n \cdot \overline{y_1^n} + (x + y_2^n) y_1^n\end{aligned}$$

$$\begin{aligned}y_2^{n+1} &= J_2 \overline{y_2^n} + \overline{K_2} y_2^n = \\ &= x \cdot \overline{y_2^n} + y_1^n \cdot y_2^n\end{aligned}$$

$$z^n = x y_1^n \overline{y_2^n}$$

Tavola di Transizione

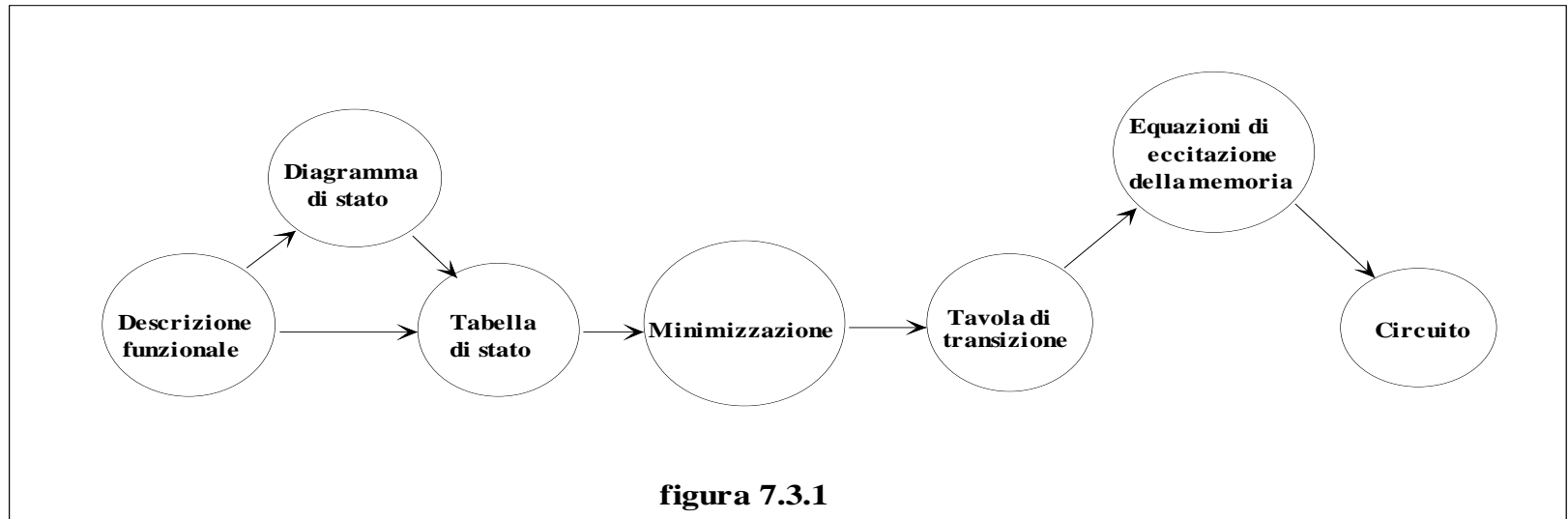
$y_1 y_2 \setminus x$	0	1
00	00/0	01/0
01	00/0	10/0
11	11/0	11/0
10	00/0	11/1

Tavola di Stato

$S \setminus x$	0	1
A	A/0	B/0
B	A/0	D/0
C	C/0	C/0
D	A/0	C/1



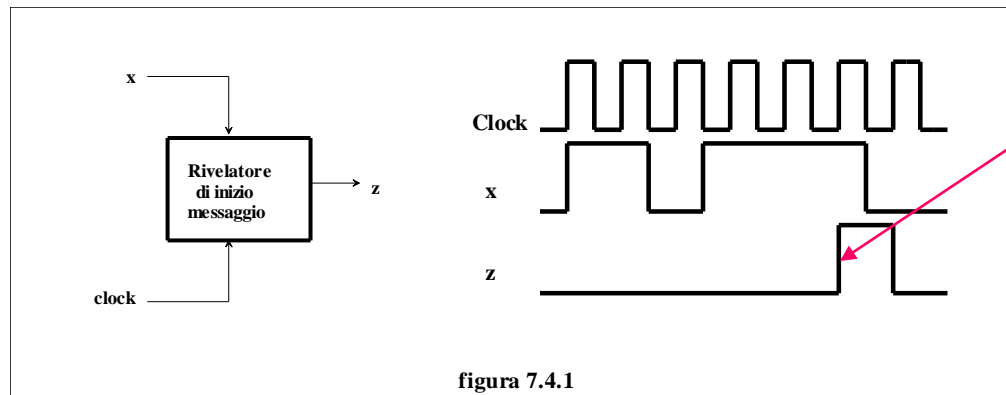
Sintesi



- Va in direzione opposta all'analisi
 - Prevede qualche passo/scelta in piu'
 - Scelta della logica (es. tipo di FF da impiegare)
 - Minimizzazione degli stati
 - Codifica degli stati
 - Stato di reset
 - tramite linea dedicata (non esplicitata negli ingressi)
 - tramite un apposito ingresso

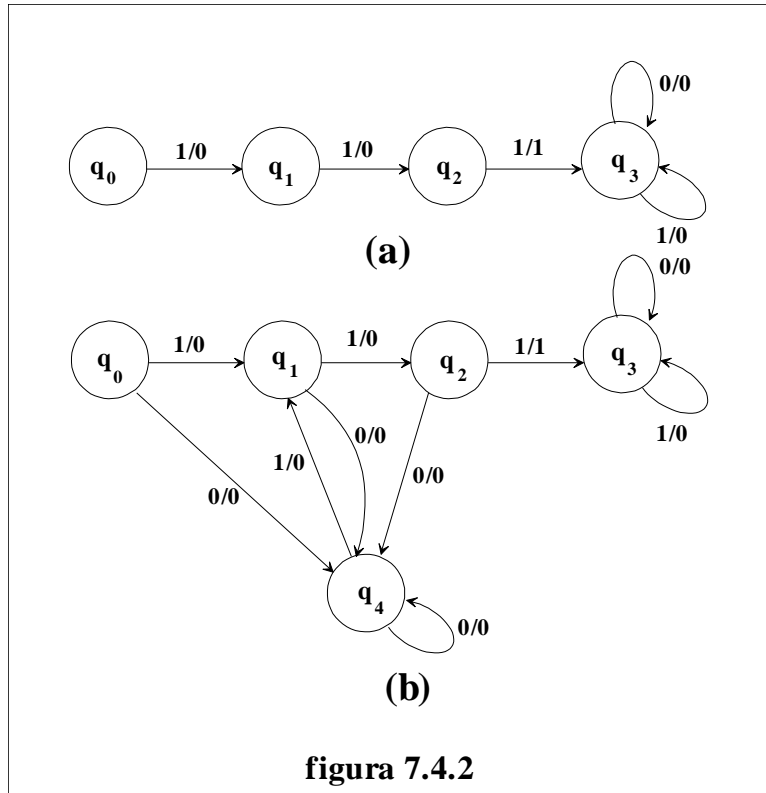
Esempio 1 (Reset implicito)

- Riconoscitore di tre uni consecutivi
- RESET tramite comando dedicato (non esplicitato in tabella)
- Ingresso sincronizzato su un clock



all'arrivo del III
uno il sistema
attiva l'uscita

Esempio 1 (cont.)



Stato	Ingresso	
	0	1
q_0	$q_4, 0$	$q_1, 0$
q_1	$q_4, 0$	$q_2, 0$
q_2	$q_4, 0$	$q_3, 1$
q_3	$q_3, 0$	$q_3, 0$
q_4	$q_4, 0$	$q_1, 0$

figura 7.4.3

Tavola degli stati

Diagramma degli stati parziale e completo

Notare la presenza di uno stato doppio (che poi potrà essere semplificato)
 Notare l'assenza di una linea dedicata al reset



Esempio 2 (reset esplicito)

- Riconoscitore di 0 (000) e 7 (111)
- Ingressi a pacchetti di 3
(Es: la sequenza 3,4 (011,100) **NON** deve essere riconosciuta)
- Ingresso particolare indicante l'inizio del messaggio (**d**)
- Ingresso sincronizzato su un clock

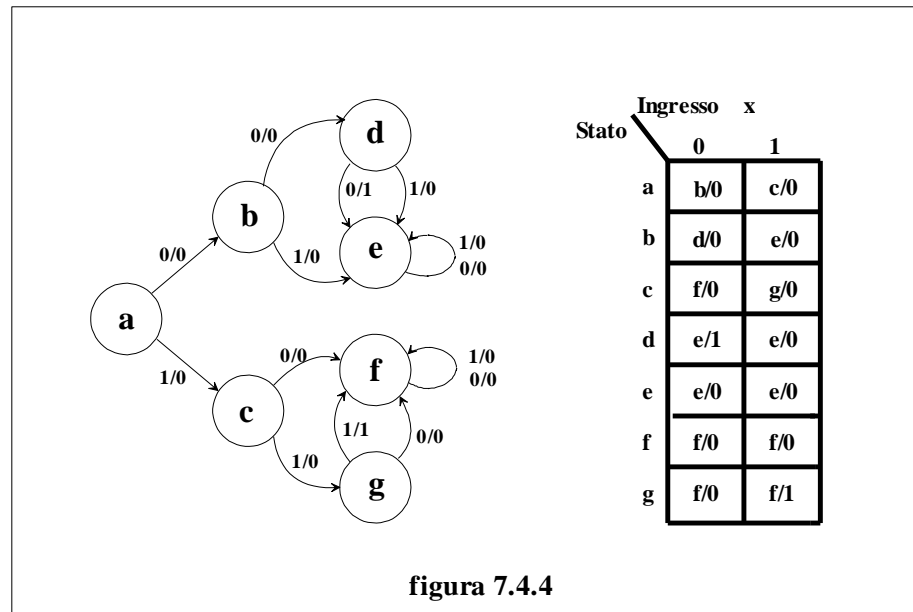


Diagramma e tavola degli stati senza la linea (d)

Notare che si potrebbe eliminare uno stato tra 'e' ed 'f'



Esempio 2 (cont.)

d.x	X
00	1
01	2
10	3
11	4

(a)

Stato	X			
	1	2	3	4
a	b/0	c/0	a/0	a/0
b	d/0	e/0	a/0	a/0
c	f/0	g/0	a/0	a/0
d	e/1	e/0	a/0	a/0
e	e/0	e/0	a/0	a/0
f	f/0	f/0	a/0	a/0
g	f/0	f/1	a/0	a/0

(b)

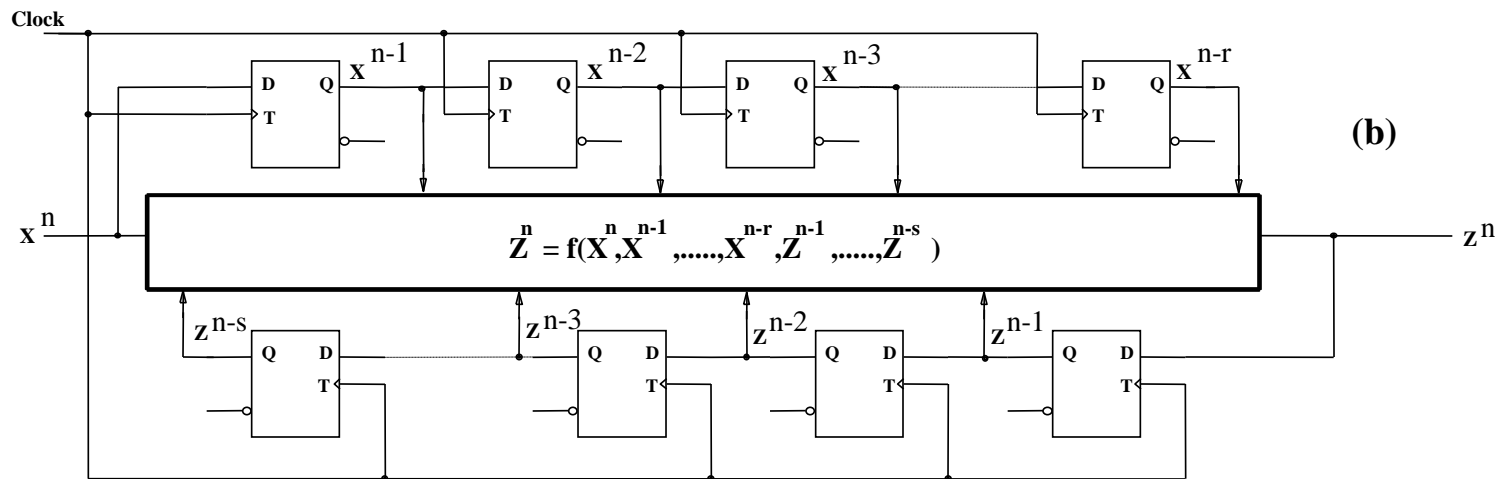
figura 7.4.5

Tavola degli stati completata dalla linea (d)



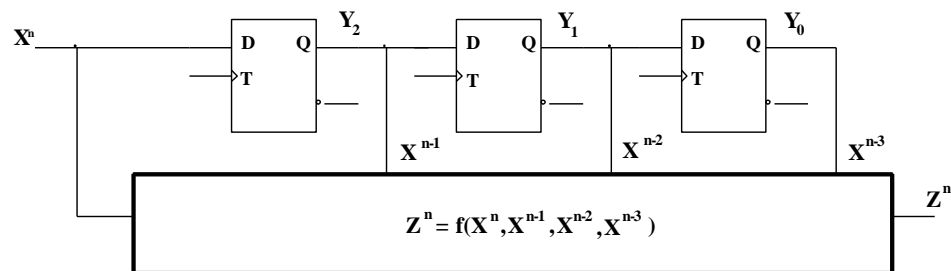
Circuiti a memoria finita

- Sono circuiti con una struttura diversa dal modello fondamentale
- Essi mantengono una memoria finita sugli ingressi e sulle uscite
- L'uscita dipende da 'r' ingressi passati e da 's' uscite passate
- I circuiti di questo tipo possono risultare piu' facili da progettare
- NON tutti i circuiti seq. sono a memoria finita!!!



Esempio 1

- Riconoscitore di due '1' oppure quattro '0' consecutivi
- Si può usare una memoria finita dell'ingresso (4 tap - 3 FF - 8 stati)
 - Il circuito può essere suddiviso in
 - Una memoria d'ingresso
 - Una rete combinatoria
- Eventualmente si può semplificare la tavola o cambiare codifica agli stati
 - ma il circuito viene a cambiare struttura



Schema di principio

$y_2 y_1 y_0$	X	0	1
000		000,1	100,0
001		000,0	100,0
010		001,0	101,0
011		001,0	101,0
100		010,0	110,1
101		010,0	110,1
111		011,0	111,1
110		011,0	111,1

(b)

Stato	X	0	1
a		a,1	e,0
b		a,0	e,0
c		b,0	f,0
d		b,0	f,0
e		c,0	g,1
f		c,0	g,1
g		d,0	h,1
h		d,0	h,1

(c)

Tavola di transizione e degli stati



Considerazione

- Il circuito a mem. finita
 - non e' il circuito minimo
 - puo' essere visto come un convertitore seriale parallelo + rete combinatoria
 - ES: Se si volesse riconoscere una seq. di 128 '1' in ingresso, l'uso di 128 FF e' ovviamente ridondante
 - Solo se l'uscita dovesse differire per ogni possibile seq. in ingresso allora il circuito e' anche minimo (per riconosere 2^n seq. ci vogliono 2^n stati)
 - Se la forma del circuito e' quella a mem. finita la tavola delle transizioni e' unica (rappresentano degli shift)



Tavola degli stati per circuiti a mem. finita dell'ingresso

$x^{n-1}x^{n-2}$	q^n	x^n	
		0	1
00	a	a	d
01	b	a	d
11	d	b	c
10	c	b	c

$x^{n-1}x^{n-2}x^{n-3}$	q^n	x^n	
		0	1
000	a	a	e
001	b	a	e
011	c	b	f
010	d	b	f
100	e	d	h
101	f	d	h
110	g	c	g
111	h	c	g

$x^{n-1}x^{n-2}x^{n-3}x^{n-4}$	q^n	x^n	
		0	1
0000	a	a	i
0001	b	a	i
0010	c	b	l
0011	d	b	l
0100	e	c	m
0101	f	c	m
0110	g	d	n
0111	h	d	n
1000	i	e	p
1001	l	e	p
1010	m	f	q
1011	n	f	q
1100	p	g	r
1101	q	g	r
1110	r	h	s
1111	s	h	s

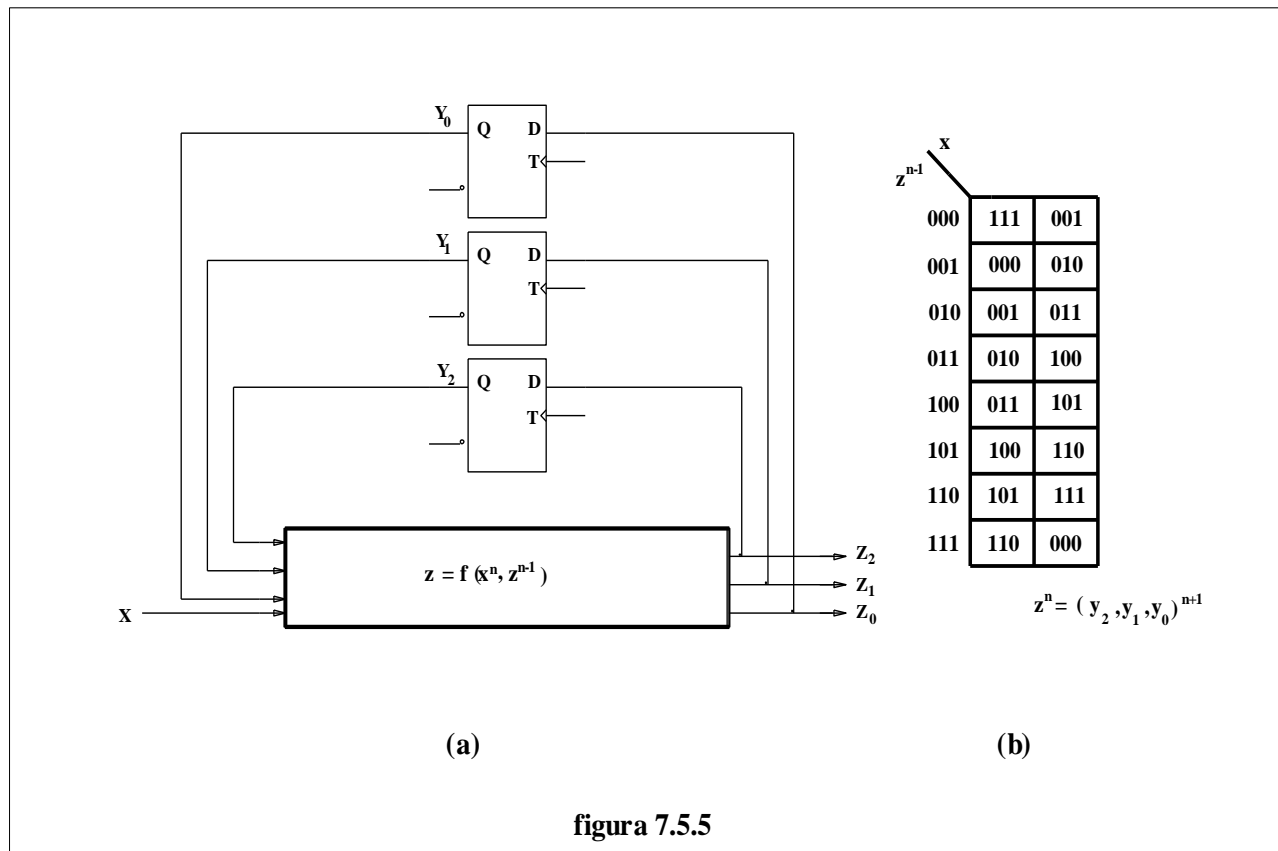
Es: 1 | 0101 \rightarrow 1010 ovvero 1 | F \rightarrow M
 0 | 0101 \rightarrow 0010 ovvero 0 | F \rightarrow C



Esempio 2

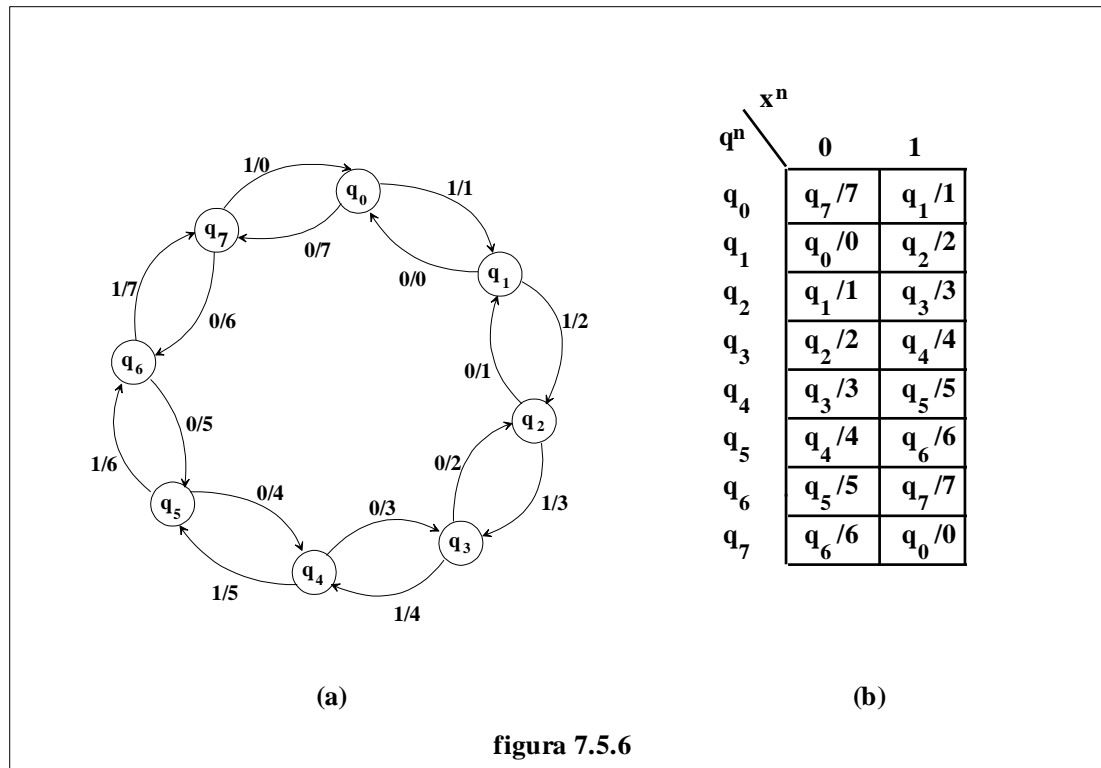
■ Circuito a mem. finita dell'uscita

- Contatore binario a 3 bit bidirezionale (up/down) con clock
- L'uscita dipende solo dal valore precedente dell'uscita e dalla "direzione" (basta memorizzare l'uscita precedente - 3 bit)



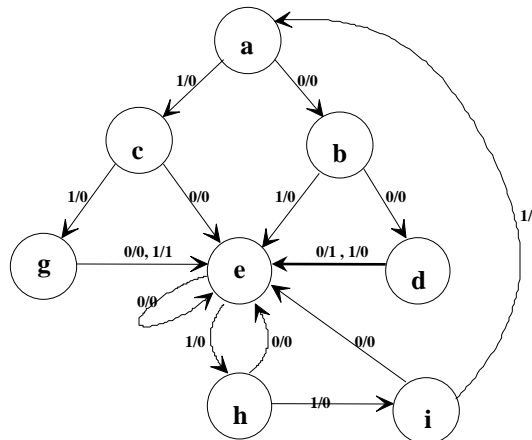
Esempio 2 (cont)

- La tavola e' minima
- Vi possono esistere altre tipologie di realizzazioni
 - ad. es. con diversi tipi di FF
- Il diagramma degli stati e' quanto riportato
 - La codifica degli stati "potrebbe" essere diversa



Esempio 3

- Riconoscitore di 0 (000) e 7 (111) e permanga nello stato
- Ingressi a pacchetti di 3
(Es: la sequenza 3,4 (011,100) **NON** deve essere riconosciuta)
- sequenza di reset (inizio trasmissione) e' 111 (**d**)
- Ingresso sincronizzato su un clock



(a)

x \ q	0	1
a	b,0	c,0
b	d,0	e,0
c	e,0	g,0
d	e,1	e,0
e	e,0	h,0
g	e,0	e,1
h	e,0	i,0
i	e,0	a,0

(b)



Minimizzazione degli stati

- La riduzione del numero degli stati e' stata gia' trattata
- Non sempre ridurre gli stati semplifica il circuito
 - Es per codificare 9 o 16 stati ci vogliono comunque 4 bit (FF)

Stato		Coppie α - compatibili		Ingresso	
		0	1	0	1
q_0	q_1	$q_0/1$	$q_4/0$	q_0, q_1	q_4, q_5
q_1	q_2	$q_0/0$	$q_4/0$	q_0, q_1	q_4, q_5
q_2	q_3	$q_1/0$	$q_5/0$	q_1, q_1	q_5, q_5
q_3	q_4	$q_1/0$	$q_5/0$	q_2, q_2	q_6, q_6
q_4	q_5	$q_2/0$	$q_6/1$	q_2, q_3	q_6, q_7
q_5	q_6	$q_2/0$	$q_6/1$	q_2, q_3	q_6, q_7
q_6	q_7	$q_3/0$	$q_7/1$	q_2, q_3	q_6, q_7
q_7		$q_3/0$	$q_7/1$	q_2, q_3	q_6, q_7
				q_3, q_3	q_7, q_7

NO

NO

Classi equivalenti :

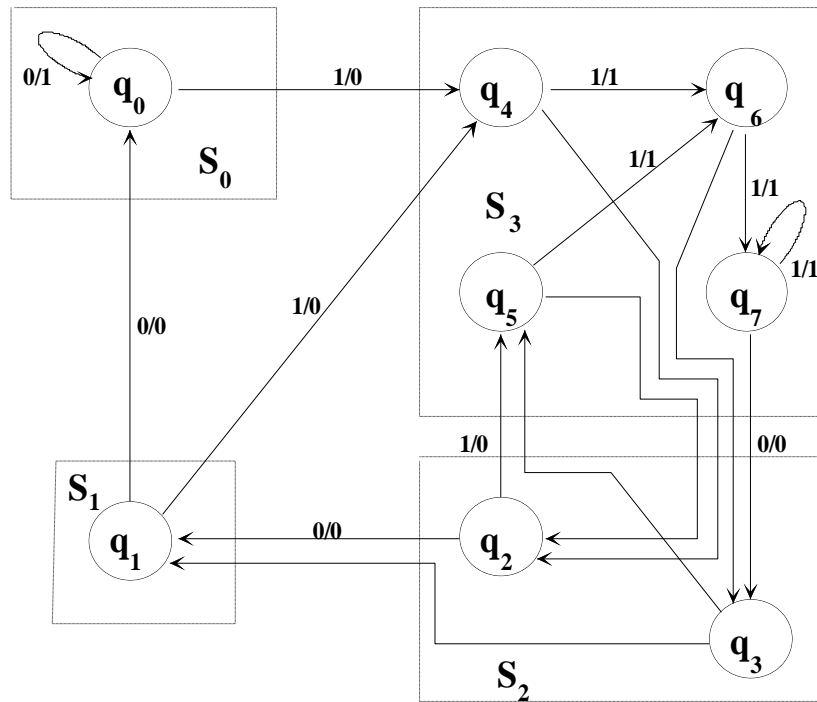
$$S0 = \{q_0\} \quad S1 = \{q_1\}$$

$$S2 = \{q_2, q_3\}$$

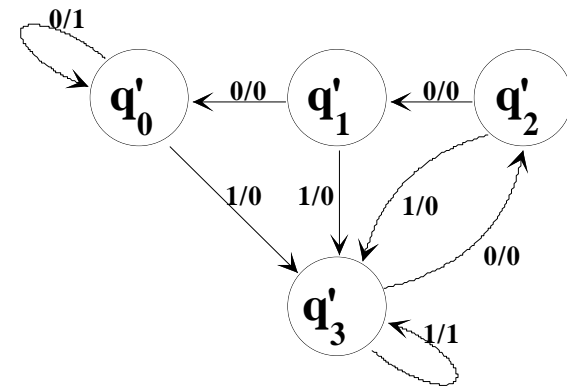
$$S3 = \{q_4, q_5, q_6, q_7\}$$



Minimizzazione degli stati (cont.)



(a)



(b)

figura 7.6.3

Codifica dello stato ed eq. di eccitazione

- Non sempre ridurre gli stati semplifica il circuito
 - Potrebbe non diminuire il numero di FF
 - Potrebbe complicare la rete combinatoria
 - Se si partiva da una struttura a mem. finita e la riduzione del numero di stati non riduce i FF merita mantenere la struttura
 - Si elimina in tal modo sicuramente la rete combinatoria per lo stato futuro
 - Se la struttura a mem. finita non e' appropriata la soluzione va trovata "per tentativi" ma con qualche accortezza
- Utilizzando 'r' FF il numero di stati realizzabile e' 'm'

$$2^{r-1} < m \leq 2^r$$

- Oppure viceversa se si devono codificare 'm' stati usando 'r' FF si possono fare molte scelte nella codifica di questi



Codifica degli stati

- Usando 'r' FF per codificare m stati le scelte possibili sono:
 1. come scegliere tra le 2^r combinazioni quali rappresentano uno stato:
 - La codifica dello stato 1 si puo' scegliere in 2^r modi
 - La codifica dello stato 2 in 2^{r-1}
 - ...
$$\frac{2^r!}{(2^r - m)!}$$
 2. inoltre si possono eliminare tutte le codifiche ottenute dalle precedenti per
 1. eventuale scambio tra le variabili (r!)
 2. eventuali inversioni di variabili (2^r)
- Esempio:
 - come codificare 3 stati con 2 bit
 1. Le scelte possibili sono $4 \cdot 3 \cdot 2 = 24$
 2. di cui 4 ottenibili per inversione delle variabili $\rightarrow 6$
 3. di cui i risultati sono a 2 a 2 sono speculari (2 possibili permutazioni delle variabili) $\rightarrow 3$



Codifica degli Stati

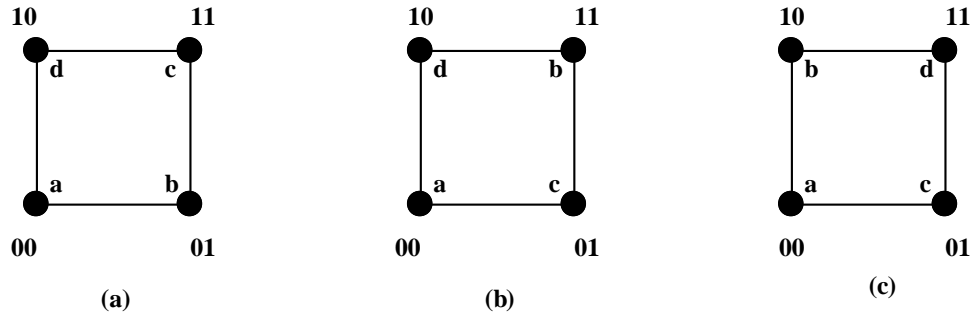


figura 7.7.1

Numero di stati	Numero di variabili	Assegnazione distinte
2	1	1
3	2	3
4	2	3
5	3	140
6	3	420
7	3	840
8	3	840
9	4	10.810.800

figura 7.7.2

Fino a 4 stati si possono provare le varie codifiche e cercare la soluzione ottimale, per più di 4 stati l'impresa è impossibile, bisogna trovare qualche "strategia"



Esempio

- Sistema per rilevare 3 uni consecutivi

Stato	Ingresso	
	0	1
a	a/0	c/0
b	a/0	d/1
c	a/0	b/0
d	d/0	d/0

figura 7.7.3

stato attuale	Ingresso	
	0	1
a	00	11
b	01	10
c	11	01
d	10	10

(a)

stato attuale	Ingresso	
	0	1
a	00	01
b	11	10
c	01	11
d	10	10

(b)

stato attuale	Ingresso	
	0	1
a	00	01
b	10	11
c	01	10
d	11	11

(c)

figura 7.7.4

Con le 3 codifiche possibili

In base alla scelta del FF adottato queste mappe devono trasformarsi in equazioni di eccitazione per i FF



Esempio (cont.)

- Bisogna far riferimento alle “liste di transizione dei FF

$Q^n \text{--} Q^{n+1}$	J	K
0	0	-
$\bar{1}$	1	-
$\bar{0}$	-	1
1	-	0

$Q^n \text{--} Q^{n+1}$	S	R
0	0	-
$\bar{1}$	1	0
$\bar{0}$	0	1
1	-	0

$Q^n \text{--} Q^{n+1}$	D
0	0
$\bar{1}$	1
$\bar{0}$	0
1	1

$Q^n \text{--} Q^{n+1}$	T
0	0
$\bar{1}$	1
$\bar{0}$	1
1	0

- Rappresentano quali siano gli ingressi da fornire ai vari FF per ottenere una certa uscita ovvero:
 - 0: mantenere lo 0
 - 1: mantenere l' 1
 - $\bar{0}$: transizione da $1 \rightarrow 0$
 - $\bar{1}$: transizione da $0 \rightarrow 1$



Esempio (cont.)

Nel caso della prima codifica

stato attuale	Ingresso	
	0	1
a 00	00	11
b 01	00	10
c 11	00	01
d 10	10	10

con FF tipo JK

$Q^n \cdot Q^{n+1}$	J	K
0	0	-
$\bar{1}$	1	-
$\bar{0}$	-	1
1	-	0

$y_1 y_2$	0	1
00	0	1
01	0	1
11	0	0
10	1	1

y'_1

$y_1 y_2$	0	1
00	0	1
01	0	1
11	-	-
10	-	-

J_{y_1}

$y_1 y_2$	0	1
00	-	-
01	-	-
11	1	1
10	0	0

K_{y_1}

$y_1 y_2$	0	1
00	0	1
01	0	0
11	0	1
10	0	0

y'_2

$y_1 y_2$	0	1
00	0	1
01	-	-
11	-	-
10	0	0

J_{y_2}

$y_1 y_2$	0	1
00	-	-
01	1	1
11	1	0
10	-	-

K_{y_2}

figura 7.7.7

$$J_{y_1} = X$$

$$K_{y_1} = y_2$$

$$J_{y_2} = X \cdot \bar{y}_1$$

$$K_{y_2} = \bar{X} + \bar{y}_1$$

$$Z = X \cdot \bar{y}_1 \cdot y_2$$



Esempio (cont.)

stato attuale		Ingresso	
		0	1
a	00	00	01
b	10	00	11
c	01	00	10
d	11	11	11

Nel caso della terza codifica

con FF tipo JK

$Q^n - Q^{n+1}$	J	K
0	0	-
1	1	-
0	-	1
1	-	0

			J_1			K_1		
			$y_1, y_2 \setminus x$	0	1	$y_1, y_2 \setminus x$	0	1
y_1	$y_1, y_2 \setminus x$		00	0	0	00	0	0
			01	0	1	01	0	1
			11	1	1	11	-	-
			10	0	1	10	-	-
y_2	$y_1, y_2 \setminus x$		00	0	1	00	0	1
			01	0	0	01	-	-
			11	1	1	11	-	-
			10	0	1	10	0	1

$$J_{y1} = x \cdot y_2 \quad J_{y2} = x$$

$$K_{y1} = \overline{x} \cdot \overline{y_2} \quad K_{y2} = \overline{y_1}$$

$$z = x \cdot \overline{y_1} \cdot \overline{y_2}$$



Esempio 2

- Riconoscitore di 2 uni o 4 zeri consecutivi da realizzarsi con FF tipo SR (la macchina semplificata e')

		Ingresso	
		0	1
Stato	a	a/1	c/0
	b	d/0	c/0
	c	b/0	c/1
	d	a/0	c/0

(a)

		Ingresso	
		0	1
Stato	00	00/1	01/0
	01	11/0	01/1
	11	10/0	01/0
	10	00/0	01/0

(b)

figura 7.7.9

$$S_{y1} = \bar{x} \cdot y_2$$

$$R_{y1} = x + \bar{y}_2$$

$$S_{y2} = x$$

$$R_{y2} = \bar{x} \cdot y_1$$

$$z = \bar{x} \cdot \bar{y}_1 \cdot \bar{y}_2 + x \cdot \bar{y}_1 \cdot y_2$$

le altre codifiche portano ad eq. piu' complesse

		Ingresso	
		0	1
Stato	00	0	0
	01	1	0
	11	-	0
	10	0	0

S_{y1}

		Ingresso	
		0	1
Stato	00	-	-
	01	0	-
	11	0	1
	10	1	1

R_{y1}

		Ingresso	
		0	1
Stato	00	0	1
	01	-	-
	11	0	-
	10	0	1

S_{y2}

		Ingresso	
		0	1
Stato	00	-	0
	01	0	0
	11	1	0
	10	-	0

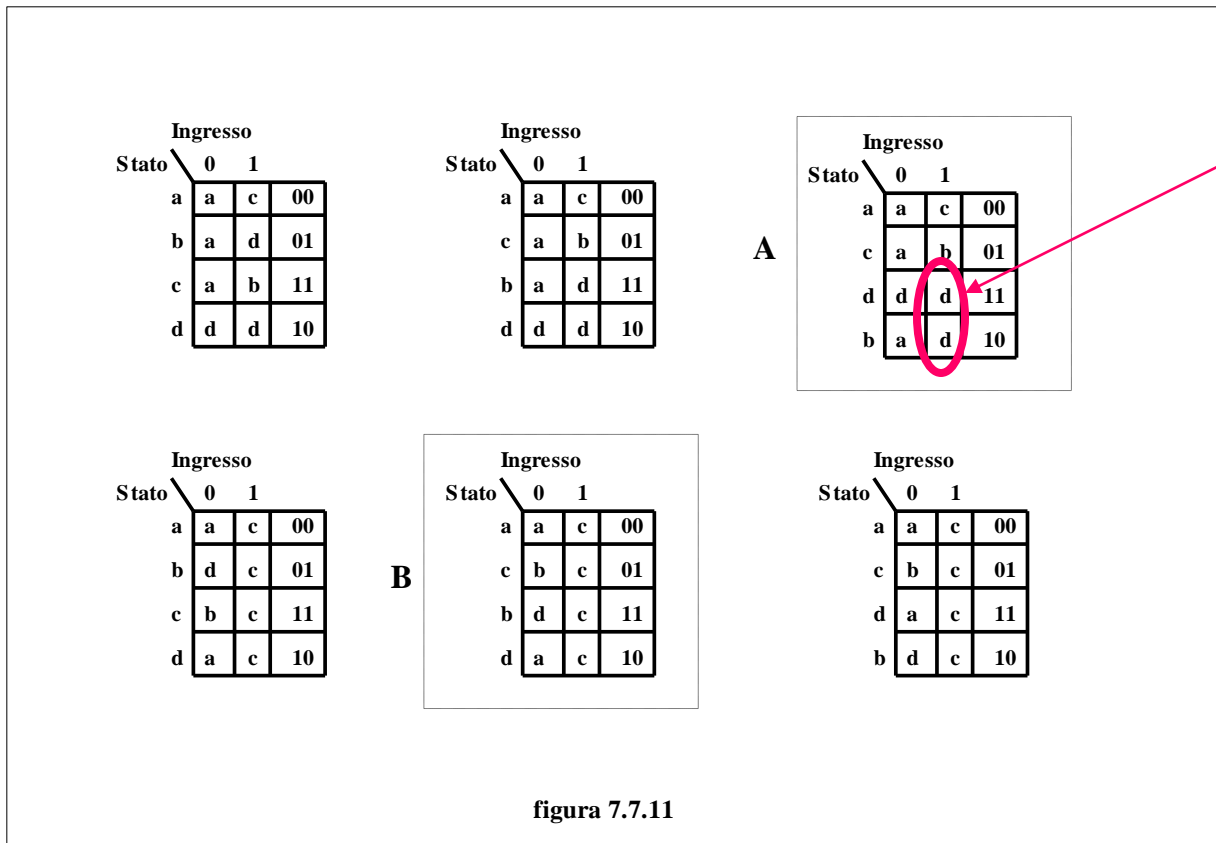
R_{y2}

figura 7.7.10



Considerazioni

- Per semplificare il circuito le mappe di eccitazione devono permettere ampi raggruppamenti
- Ovvero bisogna mettere vicino ingressi con stadi futuri uguali



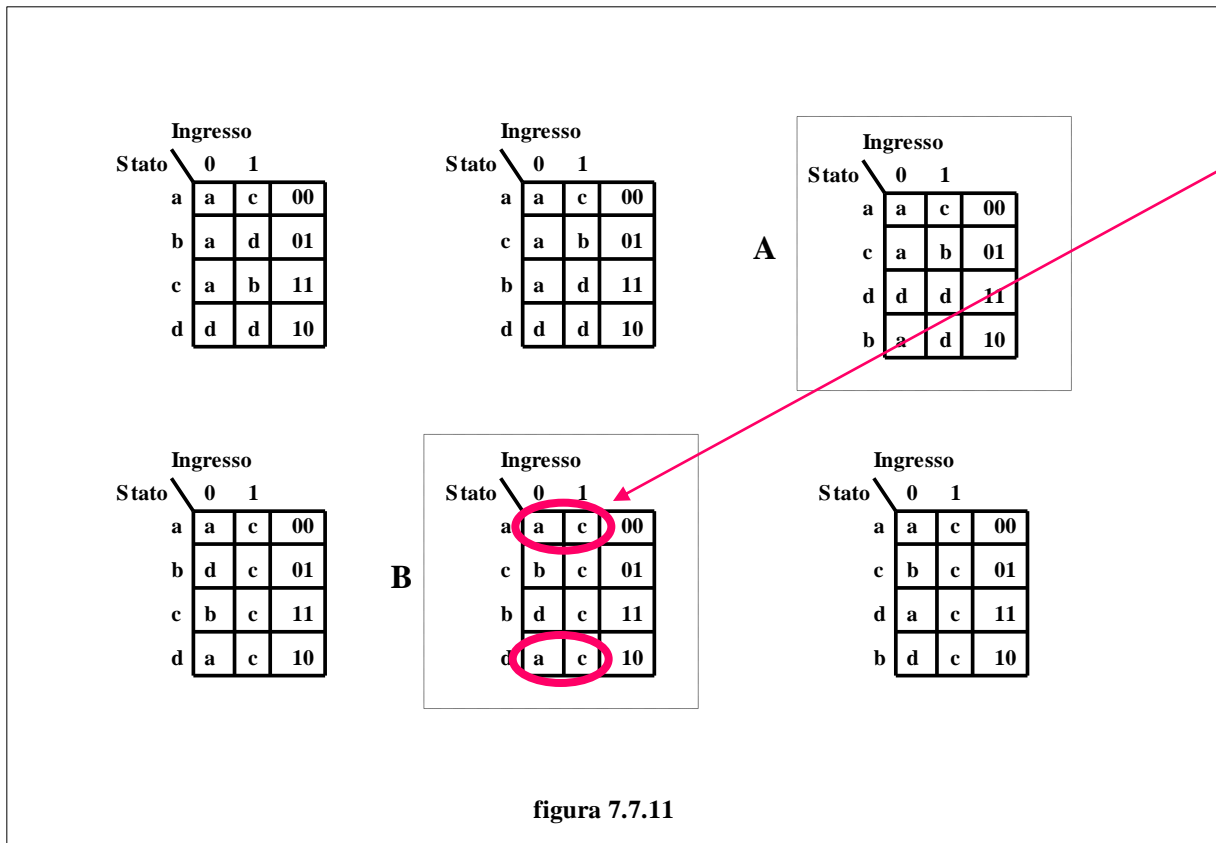
Stati futuri uguali
(con stessa codifica)
per stati adiacenti
(una var. a comune)
per quella variabile la
transizione sarà
uguale e quindi
uguale eccitazione

figura 7.7.11



Considerazioni

- Per semplificare il circuito le mappe di eccitazione devono permettere ampi raggruppamenti
- Ovvero bisogna mettere vicino ingressi con stadi futuri uguali



Se a e d sono
adiacenti e gli stati
futuri sono uguali si
crea semplificazione

Inoltre se anche a e c
sono adiacenti una
variabile sarà
comune a tutte e 4 le
posizioni



Considerazioni

		Ingresso	
		0	1
Stato	a	c	a
	b	d	c
	c	b	a
	d	a	c

figura 7.7.12

Se a e d sono
adiacenti e gli stati
futuri sono a loro
volta adiacenti si ha
maggiori probabilita'
di creare dei
raggruppamenti



Considerazioni

■ Regole (in ordine d'importanza)

1. Regola 1

1. Stati attuali che hanno lo stesso stato futuro in ciascuna colonna devono essere resi adiacenti
2. Stati attuali che evolvono verso gli stessi stati futuri ma con diverso ordine nelle colonne devono essere adiacenti se anche gli stati futuri sono adiacenti
3. Stati relativi a righe che hanno stati futuri uguali solo per talune colonne vanno resi adiacenti (dando la precedenza a quelli dove piu' colonne hanno stati uguali in comune)

2. Regola 2

1. Stati futuri di una riga devono essere resi adiacenti

3. Regola 3

1. L'assegnazione deve semplificare per quanto possibile la mappa d'uscita



Tavola inversa di flusso

- Altra procedura di semplificazione:

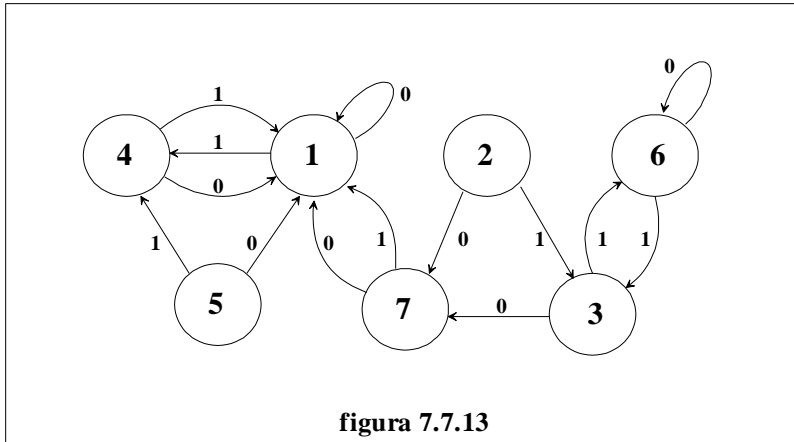


figura 7.7.13

Stato attuale	Stato precedente		n. termini
	x=0	x=1	
1	1,4,5,7	4,7	6
2	--	--	0
3	--	2,6	2
4	--	1,5	2
5	--	--	0
6	6	3	2
7	2,3	--	2

figura 7.7.14

Stato 1 : cod: 000 (nella tavola di flusso ci saranno molti zeri)

Adiacenze per righe: {2,6},{1,5},{2,3} : nella tavola di flusso evolvono verso lo stesso stato futuro

Adiacenze per colonne: {2,6} → {6,7}; {2,3} → {3,6}

Tutte queste adiacenze non sono realizzabili contemporaneamente

Una soluzione puo' essere:

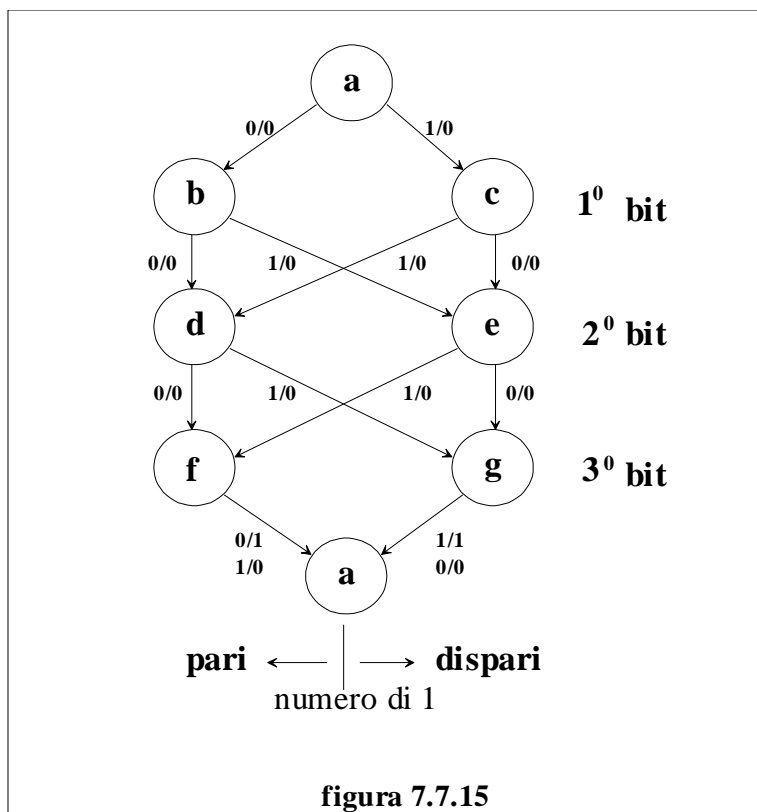
y_3	$y_1 y_2$			
	00	01	11	10
0	1	5	-	4
1	2	6	7	3

	x=0	x=1
1	1	4
2	7	3
3	7	6
4	1	1
5	1	4
6	6	3
7	1	1



Esempio

- Controllore di parità su 4 bit con reset
(l'uscita viene fornita solo sul 4⁰ bit) altrimenti e' sempre zero



stato	ingresso	
	0	1
a	b/0	c/0
b	d/0	e/0
c	e/0	d/0
d	f/0	g/0
e	g/0	f/0
f	a/1	a/0
g	a/0	a/1



Esempio (cont.)

Regole:

- 1.1 adiacenza {f, g}
- 1.2 adiacenza {d,e} se lo sono {f,g}
- 1.2 adiacenza {b,c} se lo sono {d,e}
- Stato a : reset : cod:000

		y_1y_2			
		00	01	11	10
y_3	0	a	f	g	-
	1	b	d	e	c

		x		
		0	1	
$y_1y_2y_3$	a	000	001/0	101/0
	b	001	011/0	111/0
	d	011	010/0	110/0
	f	010	000/1	000/0
		100	--/-	--/-
	c	101	111/0	011/0
	e	111	110/0	010/0
	g	110	000/0	000/1

		ingresso	
		0	1
stato	a	b/0	c/0
	b	d/0	e/0
	c	e/0	d/0
	d	f/0	g/0
	e	g/0	f/0
	f	a/1	a/0
	g	a/0	a/1

Nota: la tavola inversa di flusso in questo caso non aiuta molto



Partizione degli stati

- Due stati sono equivalenti se per qualunque ingresso le uscite coincidono e l'evoluzione li porta verso 2 stati che sono ancora equivalenti ovvero le uscite sono indistinguibili.

$$w(p, x) = w(q, x) \quad S(p, x) = S(q, x)$$

- Si consideri solo la II condizione:
 - Due stati appartengono alla stessa classe (**partizioni chiuse**) se per ciascun ingresso evolvono verso stati che appartengono alla medesima classe

Es: gli stati pari evolvono sempre verso stati dispari e quelli dispari sempre verso i pari
partizioni $\{0,2,4,6\}$ e $\{1,3,5,7\}$
ma anche
 $\{0,4\}, \{1,5\}, \{2,6\}, \{3,7\}$

stato	ingresso	
	0	1
0	7	1
1	0	2
2	1	3
3	2	4
4	3	5
5	4	6
6	5	7
7	6	0

figura 7.8.1



Partizioni degli stati e codifica

- Quando vi siano delle partizioni chiuse la codifica puo' risultare agevolata
 - Es: per l'insieme $\{0,2,4,6\}$ $y_1=0$ mentre per $\{1,3,5,7\}$ $y_1=1$
 - ne segue che $y_1^{n+1} = \overline{y_1^n}$
- Se si evidenziano partizioni chiuse con un opportuna codifica si puo'
 - Semplificare la logica che determina le variabili di stato
 - Semplificare il circuito (suddividendolo in sottocircuiti)
 - In pratica le variabili di stato dipendono solo da alcune variabili (ed al piu' dagli ingressi), ma non da tutte

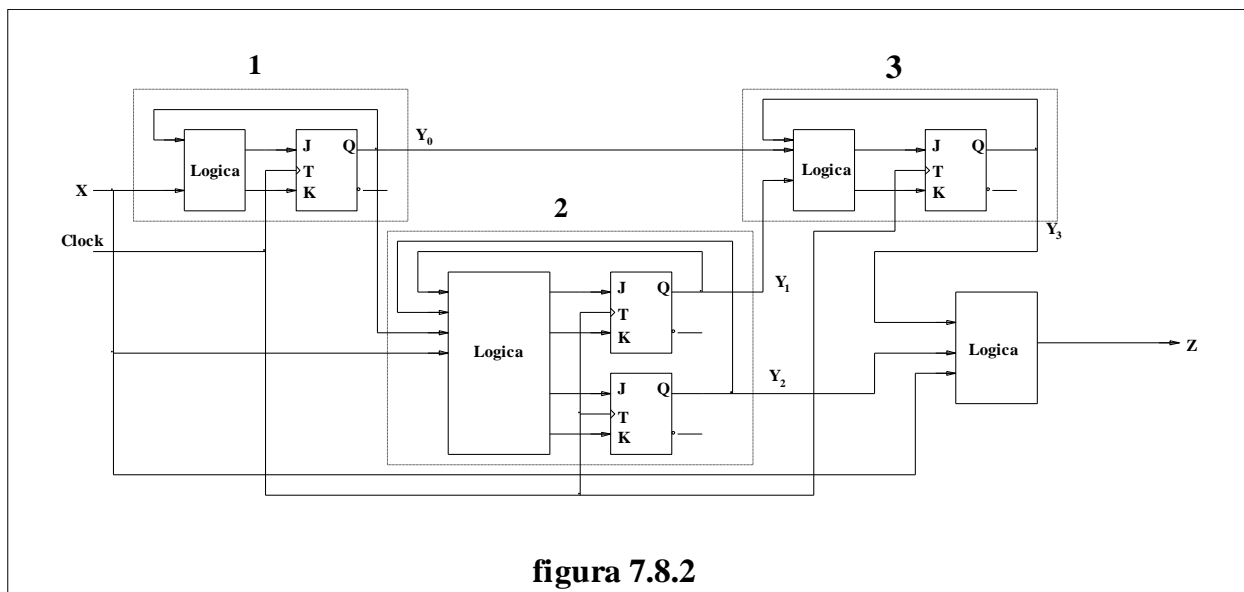


figura 7.8.2



Esempio 1

- Controllore di parità su 4 bits con uscita all'ultimo bit (già visto)

- Sotto-funzioni:

- Conteggio su 4 bit (2 FF)
- Controllo di parità (1 FF)

- y_1, y_2 non dipendono da x e y_0
- y_0 dipende da x, y_0, y_1, y_2
- z dipende da x, y_0, y_1, y_2

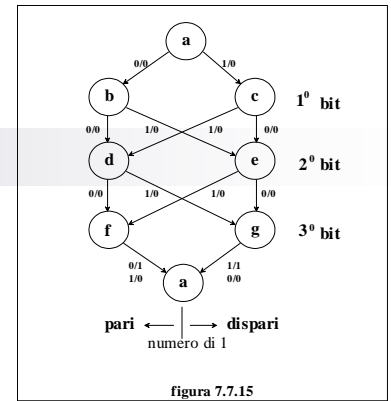


figura 7.7.15

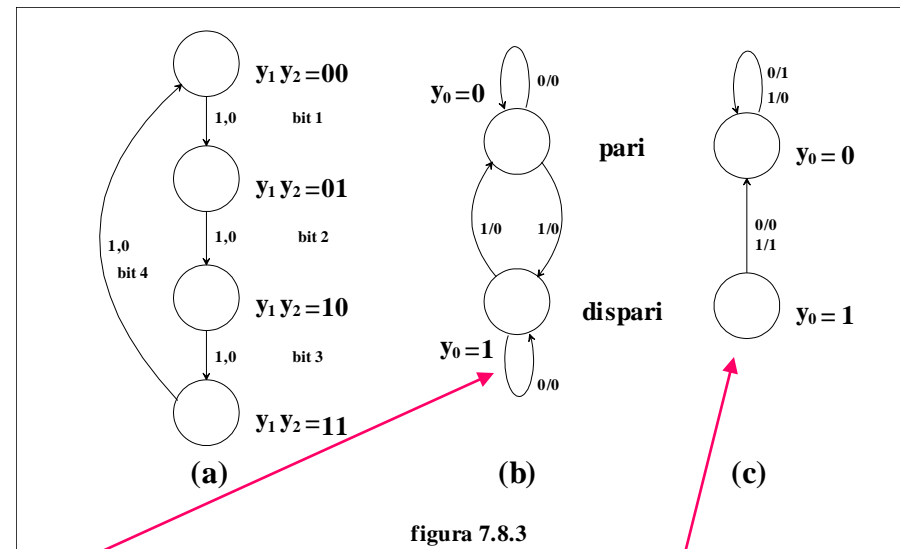


figura 7.8.3

Per i primi 3 bit l'uscita non cambia

Per l'ultimo bit si definisce l'uscita e si ritorna allo stato iniziale



Esempio 1 (cont)

- I diagrammi devono essere riuniti in una singola tabella

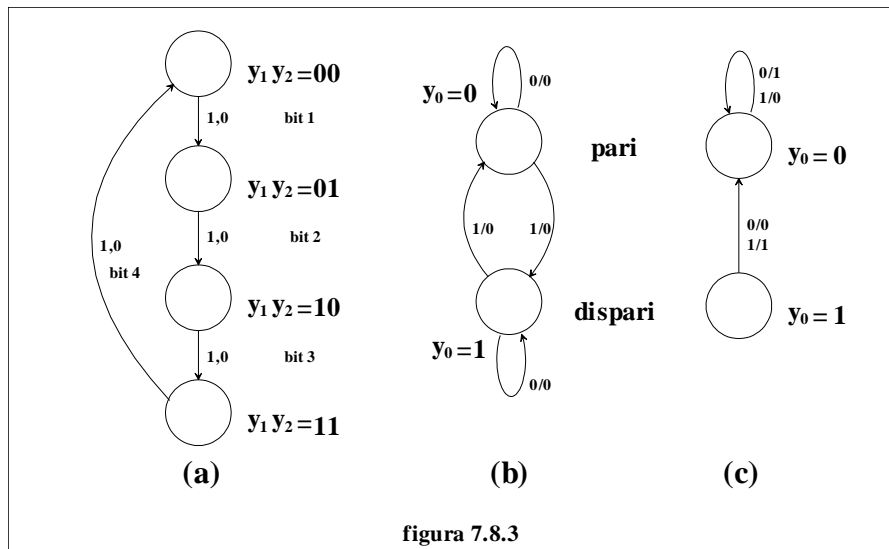


figura 7.8.3

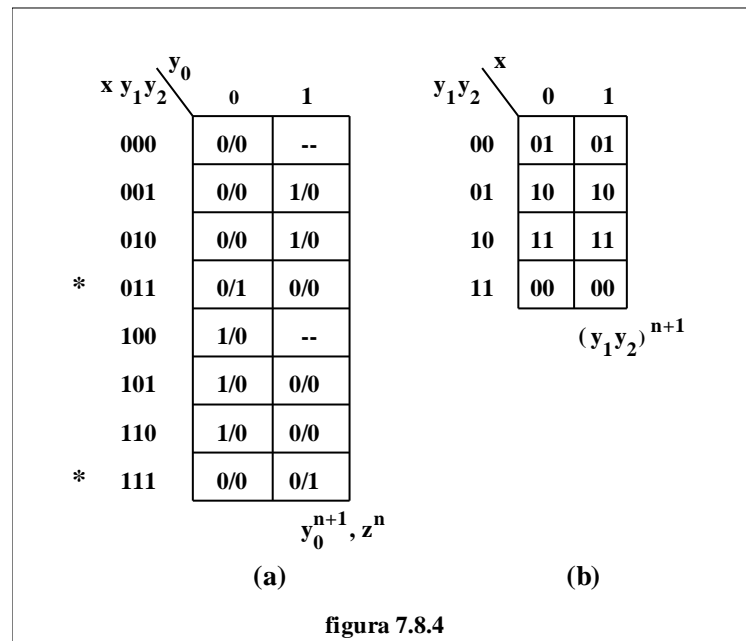


figura 7.8.4



Esempio 1 (cont)

$y_1 y_2 \backslash x$	0	1
00	01	01
01	10	10
10	11	11
11	00	00

y_1

$y_1 y_2 \backslash x$	0	1
00	0	0
01	1	1
10	1	1
11	0	0

J_1

$y_1 y_2 \backslash x$	0	1
00	0	0
01	1	1
10	-	-
11	-	-

K_1

$y_1 y_2 \backslash x$	0	1
00	-	-
01	-	-
10	0	0
11	1	1

y_2

$y_1 y_2 \backslash x$	0	1
00	1	1
01	0	0
10	1	1
11	0	0

J_2

$y_1 y_2 \backslash x$	0	1
00	1	1
01	-	-
10	1	1
11	-	-

K_2

$y_1 y_2 \backslash x$	0	1
00	-	-
01	1	1
10	-	-
11	1	1

$$J_{y_2} = K_{y_2} = 1$$

$$J_{y_1} = K_{y_1} = y_2$$



Esempio 1 (cont)

xy1y2\y0	0	1
000	0/0	-
001	0/0	1/0
010	0/0	1/0
011	0/1	1/0
100	1/0	-
101	1/0	0/0
110	1/0	0/0
111	0/0	0/1

xy1y2\y0	0	1
000	0	-
001	0	0
010	0	0
011	1	0
100	0	-
101	0	0
110	0	0
111	0	1

Z

J_0

xy1y2\y0	0	1
000	0	-
001	0	1
010	0	1
011	0	1
100	1	-
101	1	0
110	1	0
111	0	0

K_0

xy1y2\y0	0	1
000	0	-
001	0	-
010	0	-
011	0	-
100	1	-
101	1	-
110	1	-
111	0	-

xy1y2\y0	0	1
000	-	-
001	-	0
010	-	0
011	-	0
100	-	-
101	-	1
110	-	1
111	-	1

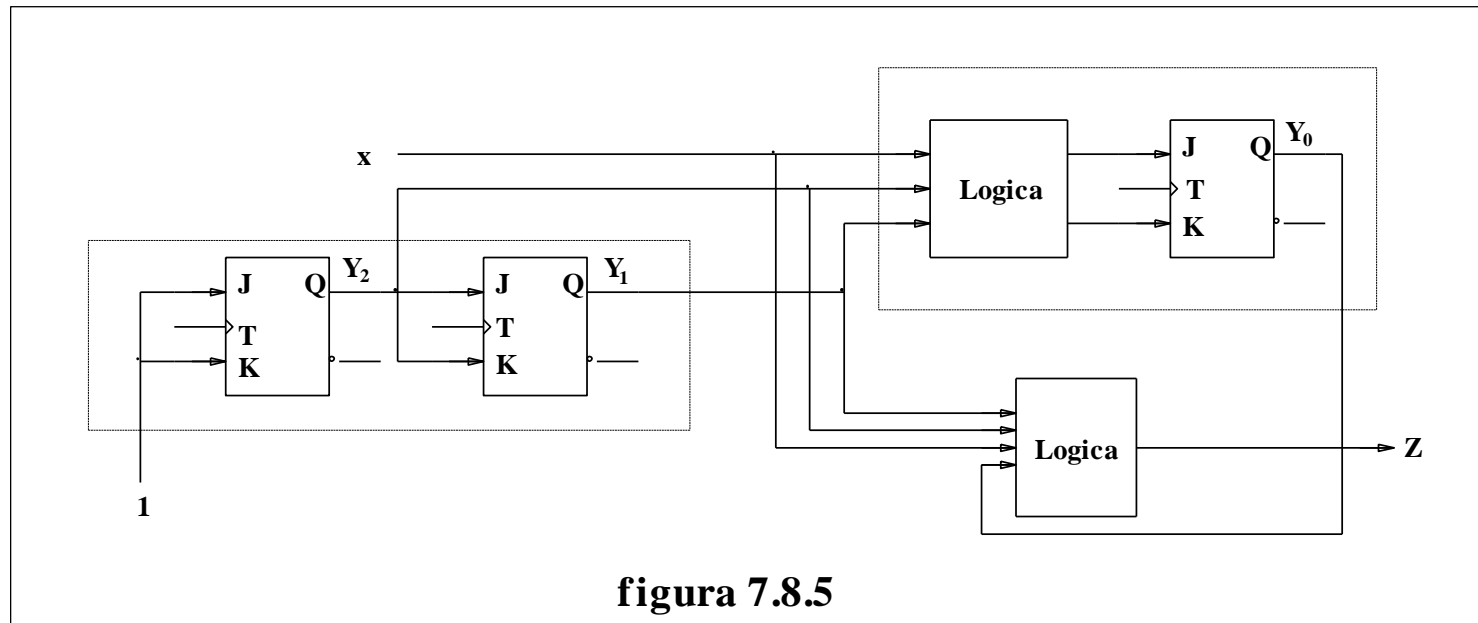
$$J_{y_0} = x \cdot (\overline{y_1} + \overline{y_2})$$

$$K_{y_0} = x + y_1 \cdot y_2$$

$$z = y_1 \cdot y_2 \cdot (x \cdot y_0 + \overline{x} \cdot \overline{y_0})$$



Esempio 1(cont)



Esempio 1 (cont)

Verifica

- riunendo le tabelle in una sola
- Poiche' y_1 e y_2 non dipendono da y_0 si deve trovare una partizione chiusa per ogni combinazione di y_1y_2 con 2 stati per partizione (uno per $y_0=0$ ed uno per $y_0=1$)

x y_1y_2		y_0	
		0	1
000	0/0	--	
001	0/0	1/0	
010	0/0	1/0	
* 011	0/1	0/0	
100	1/0	--	
101	1/0	0/0	
110	1/0	0/0	
* 111	0/0	0/1	

y_0^{n+1}, z^n

(a)

y_1y_2		x	
		0	1
00	01	01	
01	10	10	
10	11	11	
11	00	00	

$(y_1y_2)^{n+1}$

(b)

figura 7.8.4

$y_0y_1y_2$		x	
		0	1
000	001	101	
001	010	110	
010	011	111	
011	000	010	
100	--	--	
101	110	010	
110	111	011	
111	000	000	

$(y_0y_1y_2)^{n+1}$

(a)

stato	ingresso	
	0	1
q_0	$q_1/0$	$q_5/0$
q_1	$q_2/0$	$q_6/0$
q_2	$q_3/0$	$q_7/0$
q_3	$q_0/1$	$q_0/0$
q_4	--/--	--/--
q_5	$q_6/0$	$q_2/0$
q_6	$q_7/0$	$q_3/0$
q_7	$q_0/0$	$q_0/1$

q^{n+1}, z^n

(b)

stati presenti	ingresso	
	0	1
0,4	1,5	1,5
1,5	2,6	2,6
2,6	3,7	3,7
3,7	0,4	0,4

stati futuri

(c)

figura 7.8.6



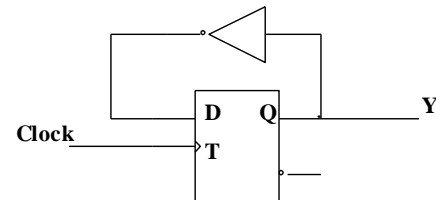
Esempio 2

■ Contatore Up/down a 3 bit

- Partizione evidente : stati pari e dispari $\{0,2,4,6\}$ e $\{1,3,5,7\}$
- Si assegni a queste partizioni la variab. y_1
 - $y_1=0 \rightarrow \{0,2,4,6\}$
 - $y_1=1 \rightarrow \{1,3,5,7\}$

$$y_1^{n+1} = \overline{y_1^n}$$

Stato	ingresso	
	0	1
0	7	1
1	0	2
2	1	3
3	2	4
4	3	5
5	4	6
6	5	7
7	6	0



blocco attuale	blocco attuale		y_1^{n+1}		
	0	1	y_1^n	0	1
(0,2,4,6)	(1,3,5,7)	(1,3,5,7)	0	1	1
(1,3,5,7)	(0,2,4,6)	(0,2,4,6)	1	0	0

figura 7.8.7



Esempio 2 (cont)

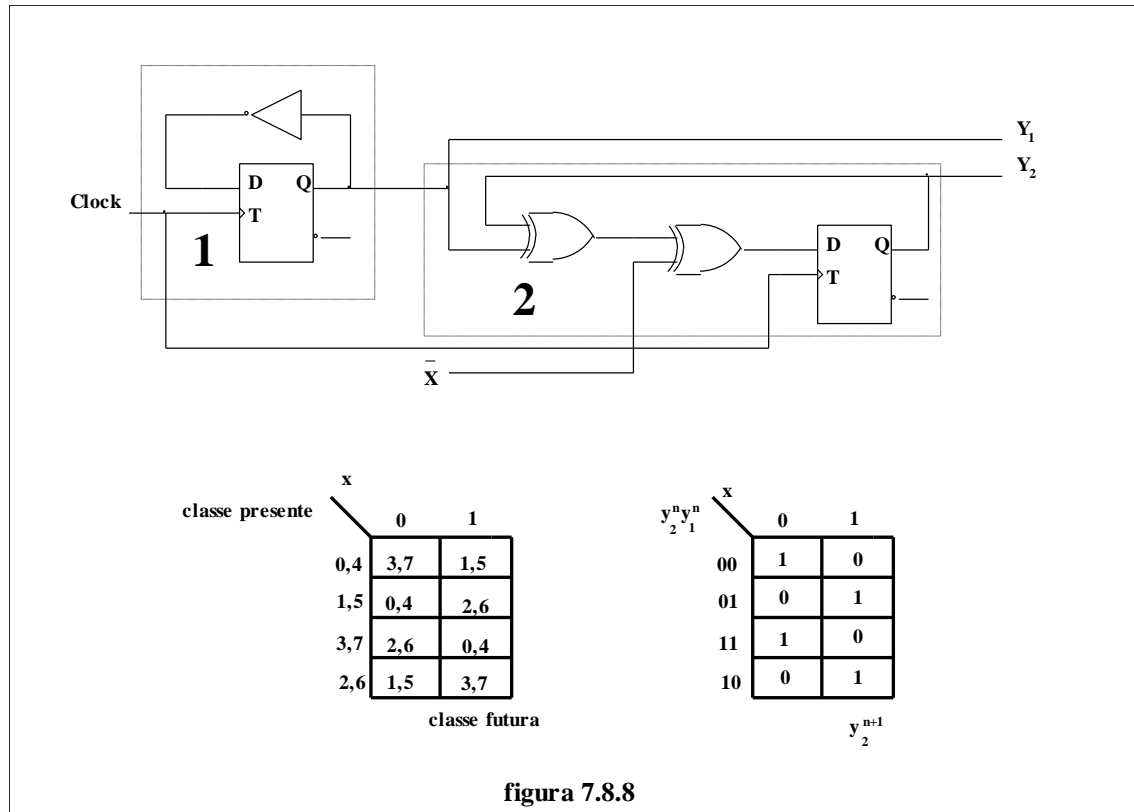
- Altre 2 variabili devono definire gli stati
 - $y_2=0$: primo e terzo del gruppo
 - $y_2=1$: secondo e quarto

$y_2 y_1$	Gruppo
00	$(0,1,4,5) \cap (0,2,4,6) = (0,4)$
01	$(0,1,4,5) \cap (1,3,5,7) = (1,5)$
11	$(2,3,6,7) \cap (1,3,5,7) = (3,7)$
10	$(2,3,6,7) \cap (0,2,4,6) = (2,6)$

- Anche questa e' una partizione chiusa
 - I valori futuri dipenderanno solo da queste variabili



Esempio 2 (cont)

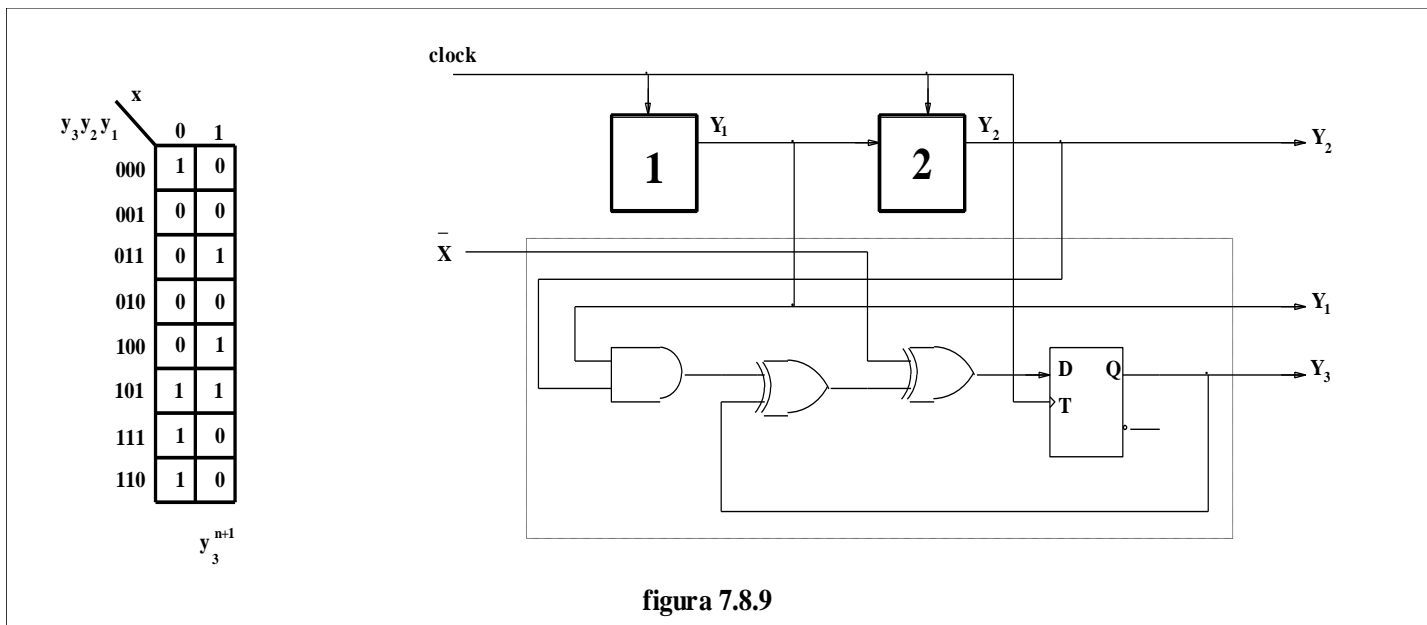


$$y_2^{n+1} = x^n \cdot (y_1^n \oplus y_2^n) + \overline{x^n} \cdot (\overline{y_1^n \oplus y_2^n}) = \overline{x^n} \oplus (y_1^n \oplus y_2^n)$$



Esempio 2 (cont)

- Infine y_3 deve individuare lo stato singolo
 - $y_3=0$: primo del gruppo
 - $y_3=1$: secondo del gruppo



$$y_3^{n+1} = ((y_1^n \cdot y_2^n) \oplus y_3^n) \oplus \overline{x}^n$$

Errore

