

Capitolo 6

Reti asincrone

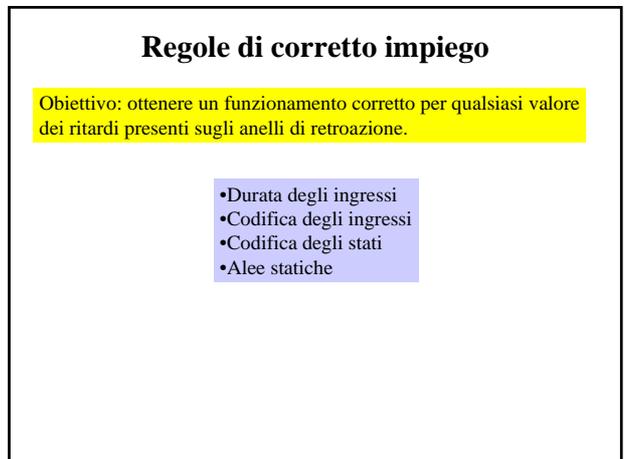
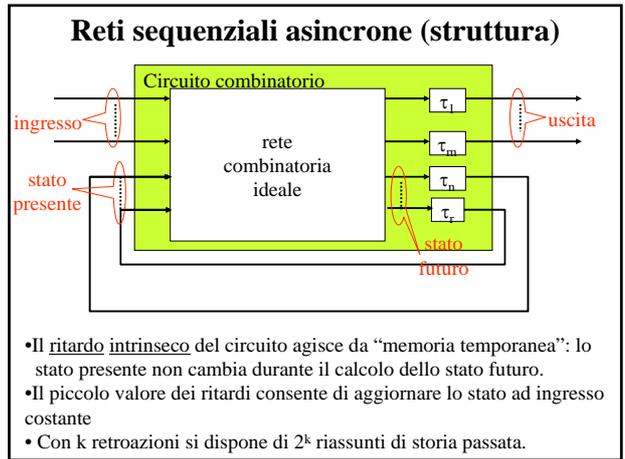
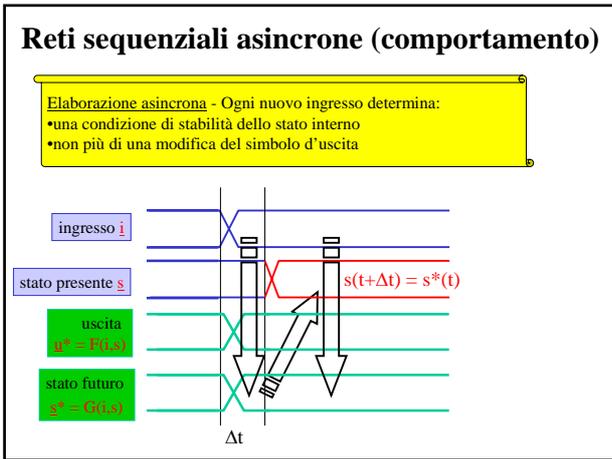
6.1 – Elaborazione asincrona

6.2 – Memorie binarie

6.3 – Analisi e Sintesi

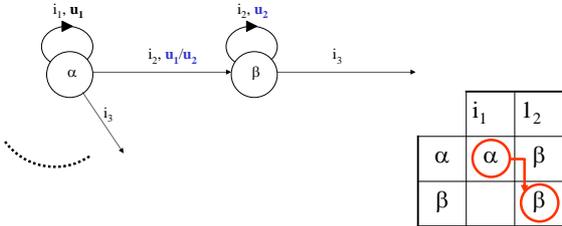
6.1

Elaborazione asincrona



1. Durata degli ingressi

L'ingresso può essere modificato solo dopo che il circuito ha raggiunto la nuova stabilità

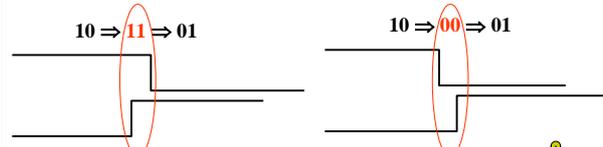


Il passaggio da una ad un'altra condizione di stabilità è detto funzionamento in **modo fondamentale**

2. Codifica degli ingressi

I segnali d'ingresso devono cambiare di valore uno solo alla volta

Esempio di situazione d'ingresso pericolosa : 10 ⇒ 01

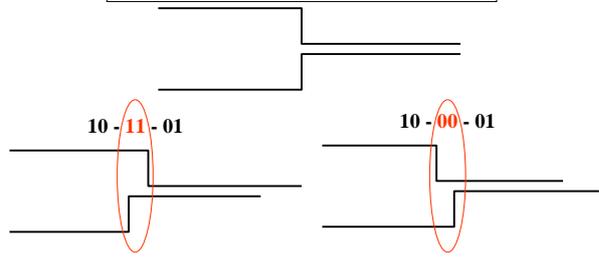


La codifica dei simboli d'ingresso non può essere arbitraria: configurazioni consecutive devono essere **adiacenti**

3. Codifica degli stati

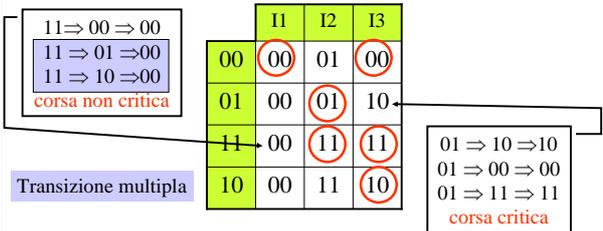
Codifica degli stati interni - Per eliminare a priori la presenza di stati interni "spuri" le config. Condizione restrittiva lo stato presente ed a tutti i suoi possibili stati Vincolo eccessivo per il valore di un solo bit

Esempio di situazione pericolosa : 10 - 01



Corse e Corse critiche

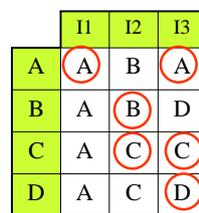
Segnali in retroazione per cui è stata prevista una modifica **contemporanea** di valore si trovano in una situazione di **corsa**: nel circuito i cambiamenti si verificheranno infatti a istanti diversi e con un ordine dettato dai ritardi interni. Una **corsa** è **critica** se si possono raggiungere stabilità diverse.



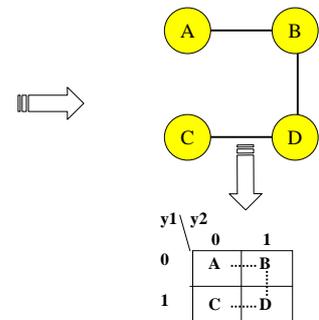
Eliminazione a priori delle corse critiche (1)

- Nelle colonne con **una sola stabilità** si inserisce il simbolo dello stato stabile al posto di eventuali condizioni d'indifferenza
- Per le sole colonne **con più stabilità** si traccia il **grafo delle adiacenze**: ogni coppia stato presente-stato futuro è indicata con due nodi connessi da un ramo orientato.
- Si sovrappone il grafo ad una mappa per il minimo numero di variabili di stato e si verifica se è possibile **assegnare configurazioni adiacenti ad ogni coppia di stati coinvolta in una transizione**
- (segue)

Una codifica priva di corse critiche

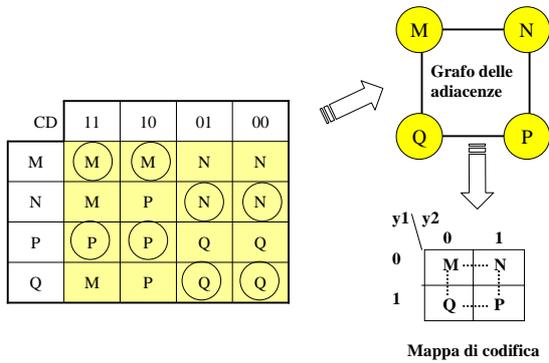


OK



Mappa di codifica

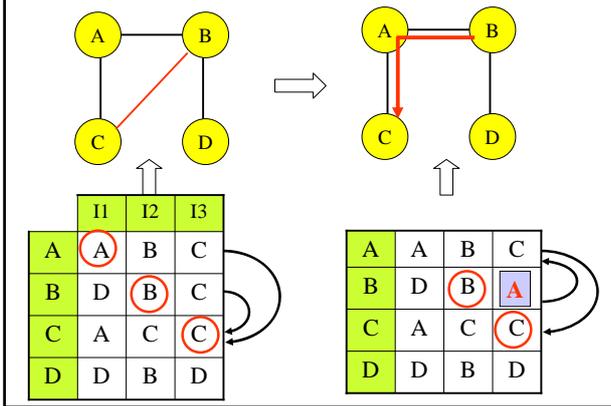
Codifica degli stati per il flip-flop (pag. 40)



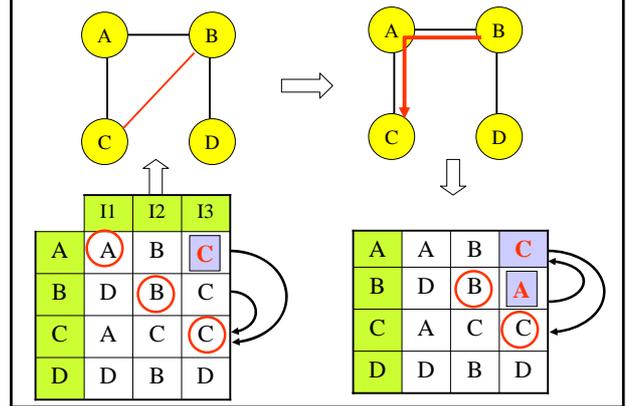
Eliminazione a priori delle corse critiche

4. Se è impossibile soddisfare tutte le richieste di adiacenza, si cerca di ridurle introducendo **transizioni multiple**
5. Se non ci si riesce, si **incrementa il numero delle variabili di stato** e si ritorna a 4.

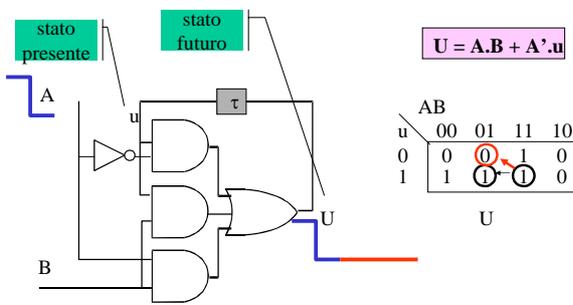
Transizioni multiple (1)



Transizioni multiple (2)



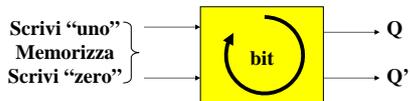
Eliminazione a priori delle alee statiche



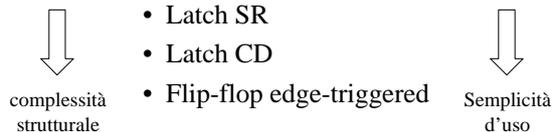
Sulle retroazioni si deve sempre rispettare T14

6.2 Memorie binarie

Memorie binarie

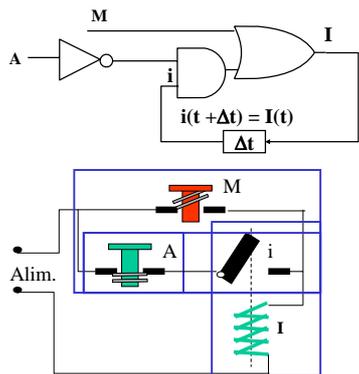


Cosa scrivere e **quando** scrivere \Rightarrow sequenza d'ingresso



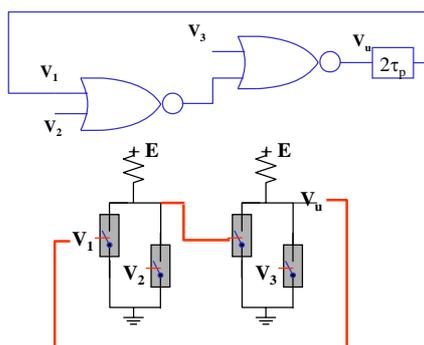
astrazione

$$I = f(M, A, i) = M + A' \cdot i$$



Rete ideale, ritardo e retroazione

$$V_u = V_3 \downarrow (V_1 \downarrow V_2) \quad V_1(t + 2\tau_p) = V_u(t)$$



Sintesi del latch Set-Reset



S	R	Q
0	0	Q
1	0	1
0	1	0

Latch SR: rete sequenziale asincrona che memorizza il valore di un bit. Il valore 1 del comando di set ordina la memorizzazione del valore 1. Il valore 1 del comando di reset ordina la memorizzazione del valore 0. Per S=0, R=0, lo stato Q ricorda l'ultimo comando ricevuto.

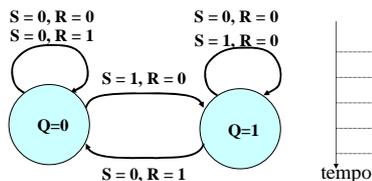


Tabella di flusso ed equazioni caratteristiche

q \ S,R	00	01	11	10
0	0	0	-	1
1	1	0	-	1

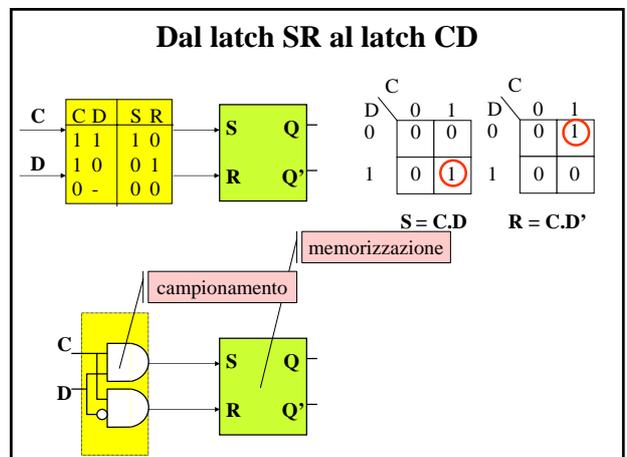
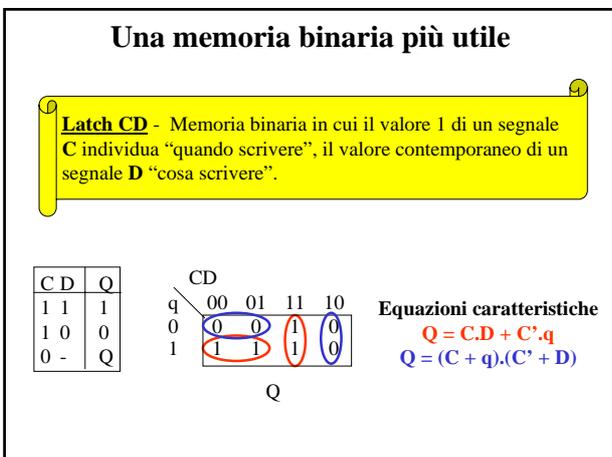
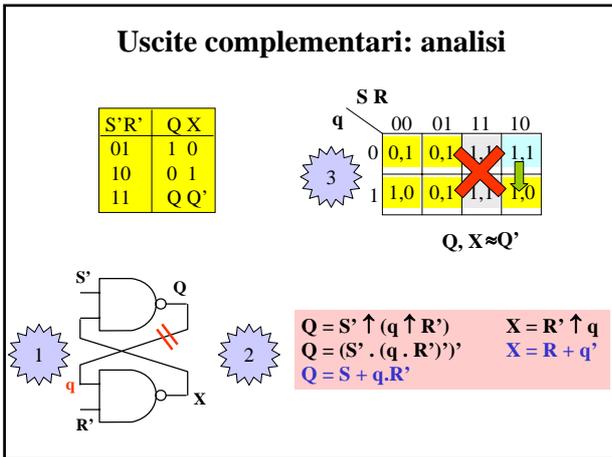
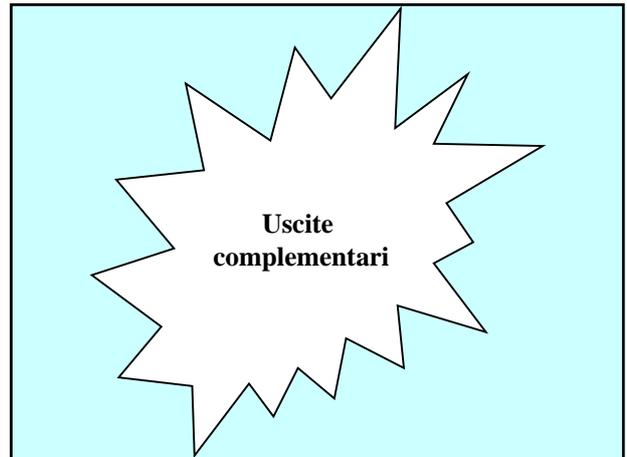
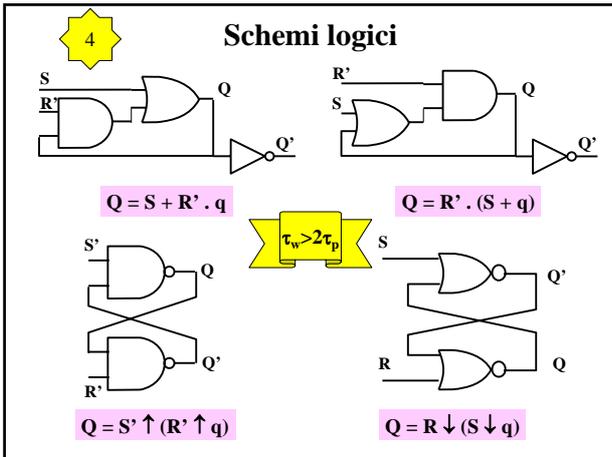
q \ SR	00	01	11	10
0	0	0	-	1
1	1	0	-	1

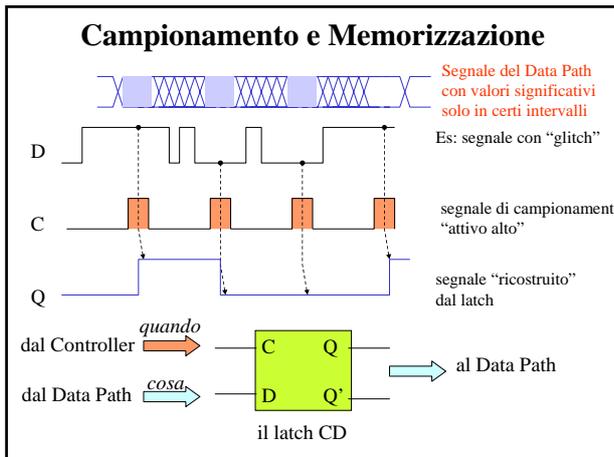
$$Q = S + R' \cdot q$$

q \ SR	00	01	11	10
0	0	0	-	1
1	1	0	-	1

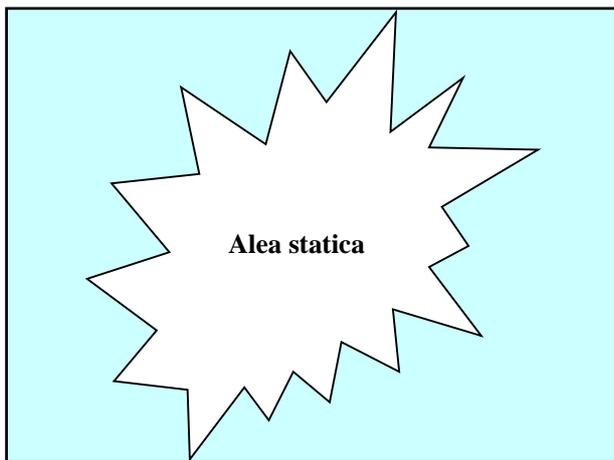
$$Q = R' \cdot (S + q)$$

La configurazione d'ingresso S=1, R=1 è vietata
 q variabile di stato presente
 Q variabile di stato futuro





- ### Problemi del latch CD
- Alea statica
 - Durata del transitorio
 - Uscite trasparenti



Alea statica nel latch CD

q \ CD	00	01	11	10
0	0	0	1	0
1	1	1	1	0

Q

Equazioni caratteristiche "ridondanti"

$$Q = C.D + C'.q + D.q$$

$$Q = (C + q).(C' + D).(D + q)$$

Realizzazione con latch SR

$$Q = S + R'.q$$

$$= C.D + (C.D)'.q$$

$$= C.D + C'.q + \bar{D}.q$$

Il latch CD a NAND ed a NOR

Realizzazione a NAND

$$Q = C.D + C'.q + D.q$$

$$= C.D + ((C' + D).q)$$

$$= (C \uparrow D) \uparrow ((C' \uparrow D) \uparrow q)$$

Realizzazione a NOR

$$Q = (C + q).(C' + D).(D + q)$$

$$= (C' + D).((C.D) + q)$$

$$= (C' \downarrow D) \downarrow ((C' \downarrow D) \downarrow q)$$

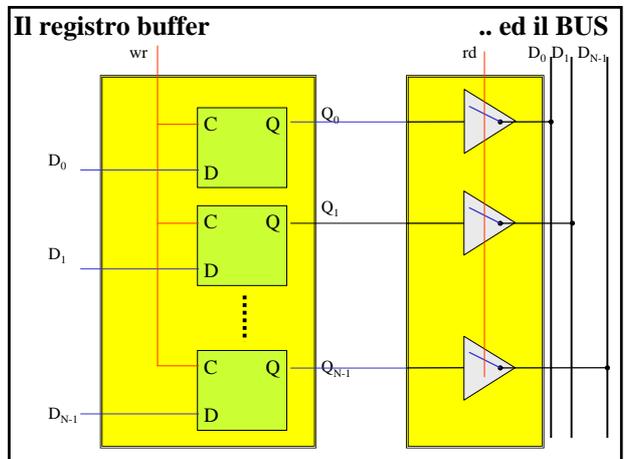
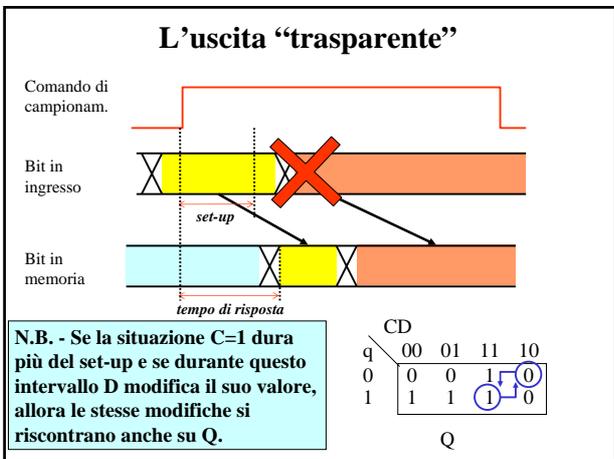
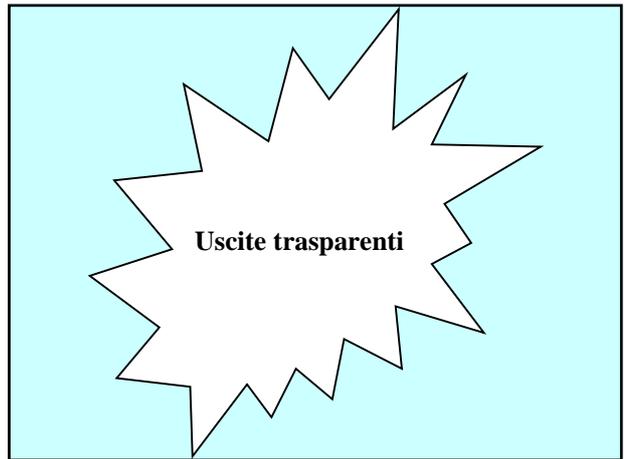
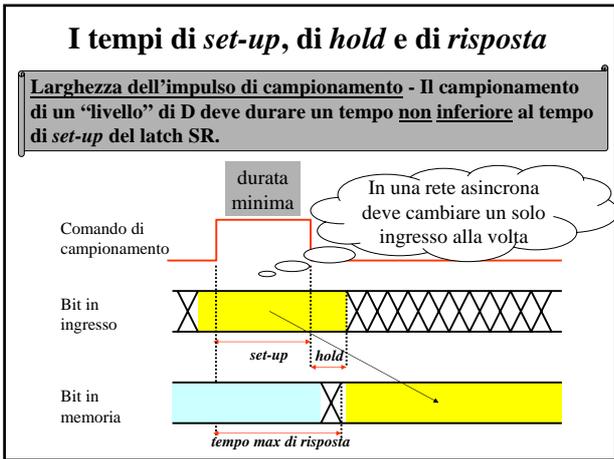
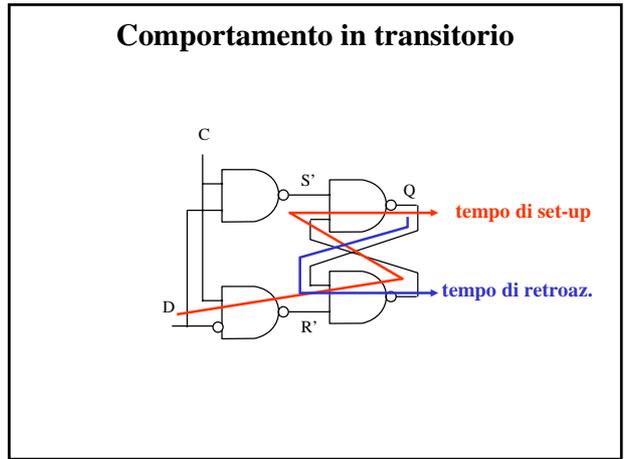
Una quarta soluzione

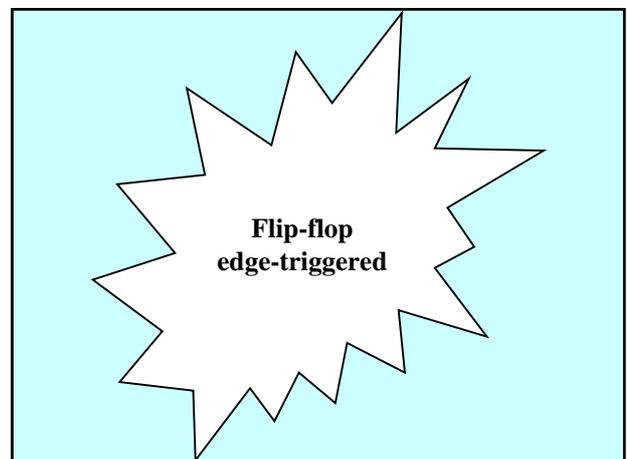
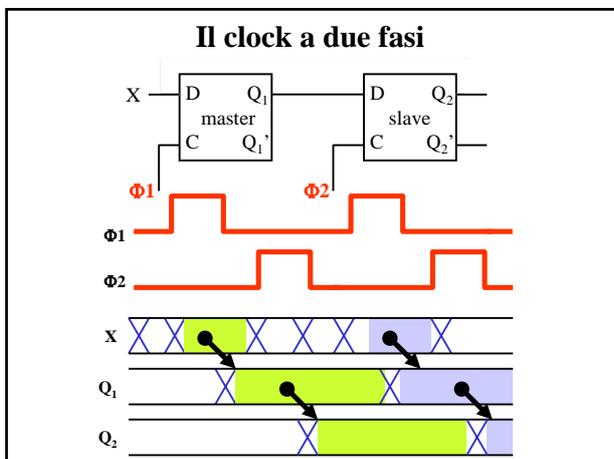
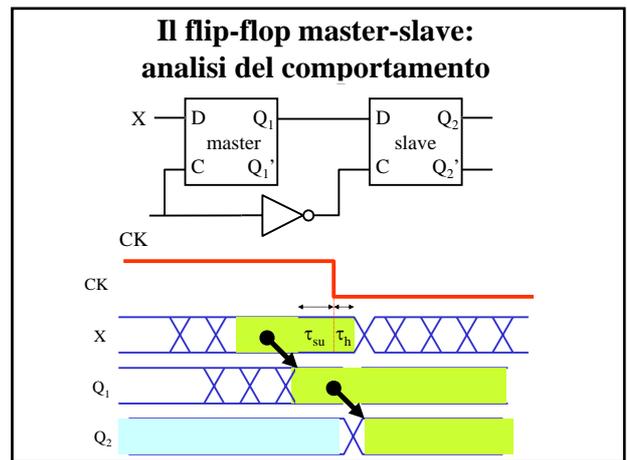
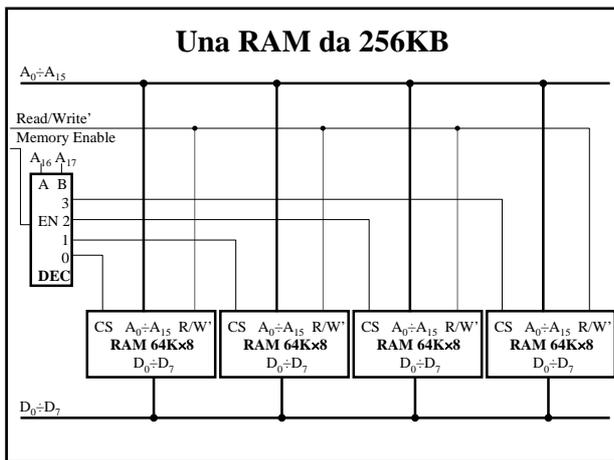
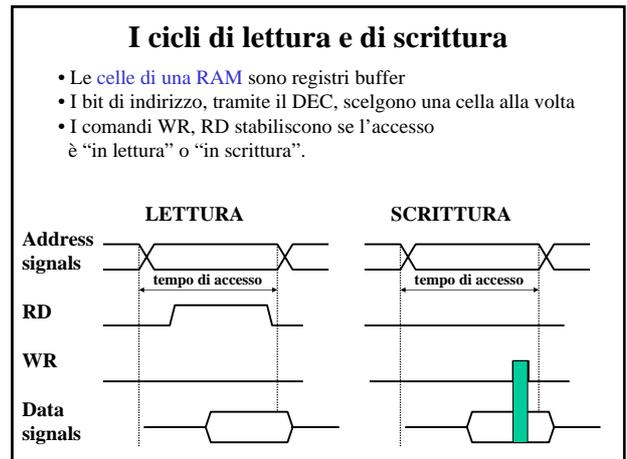
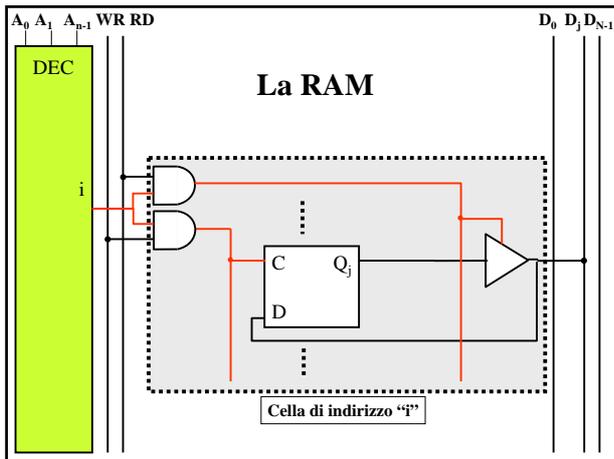
q \ CD	00	01	11	10
0	0	0	1	0
1	1	1	1	0

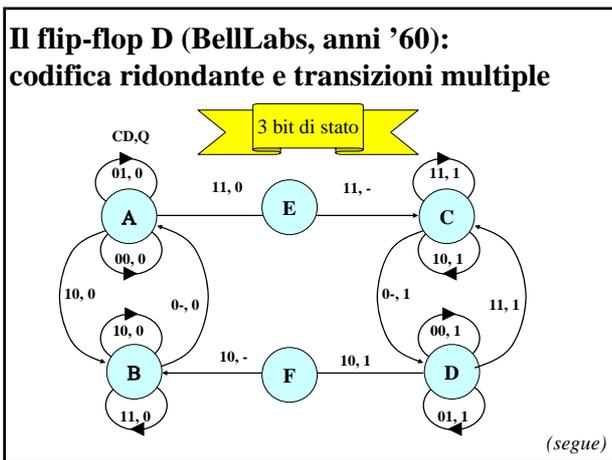
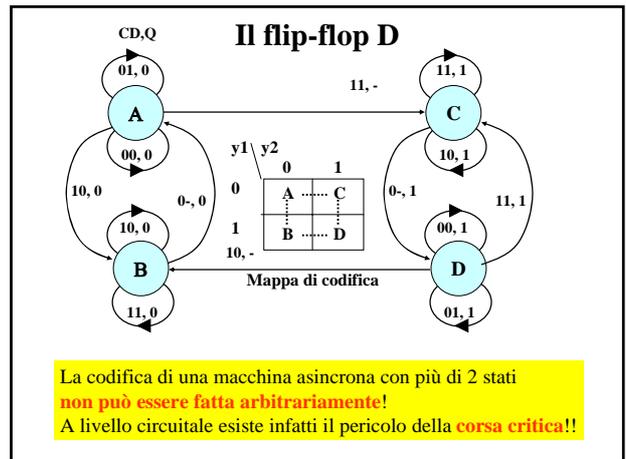
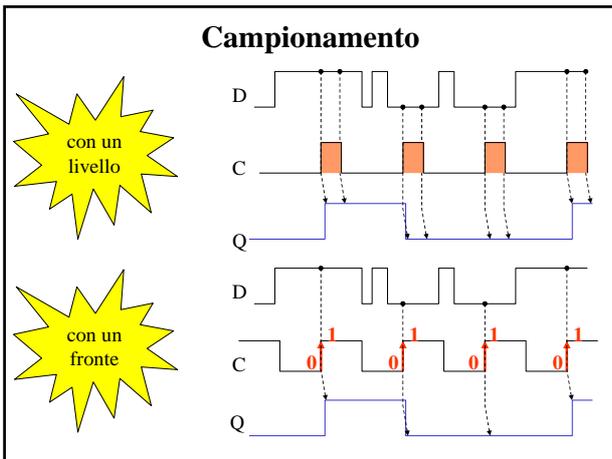
Q

$$Q = C.D + C'.q$$

N.B. - Il ritardo inerziale dei due NOT elimina l'eventuale "glitch" generato dall'alea statica del MUX.







La tabella delle transizioni

stato		CD			
		00	01	11	10
E =	001	---	---	---	---
A =	011	011,0	011,0	101,0	010,0
B =	010	011,0	011,0	010,0	010,0
	100	---	---	---	---
C =	101	111,1	111,1	101,1	101,1
D =	111	111,1	111,1	101,1	110,1
F =	110	---	---	---	010,-

$y_1 y_2 y_3$ $Y_1 Y_2 Y_3, Q$

(segue)

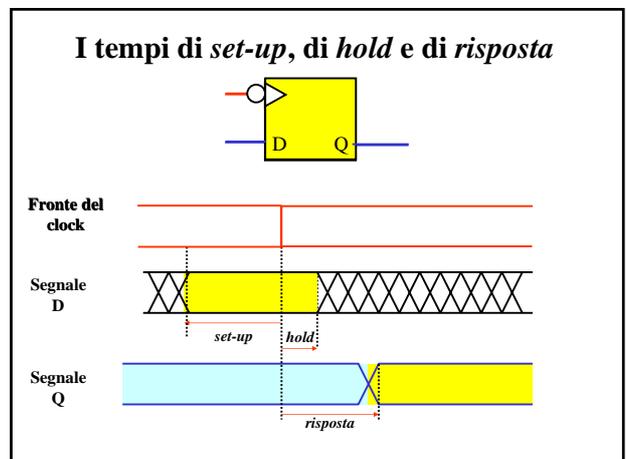
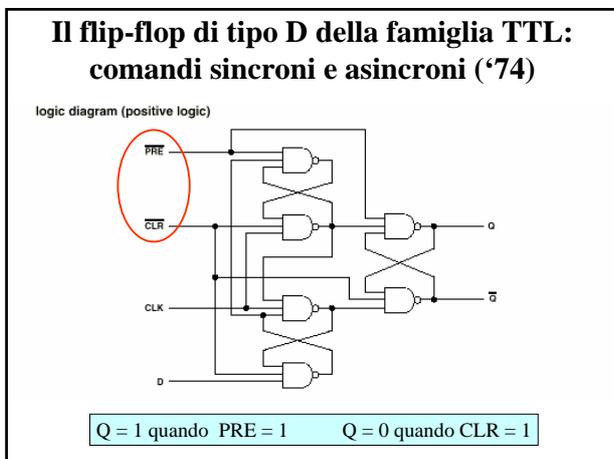
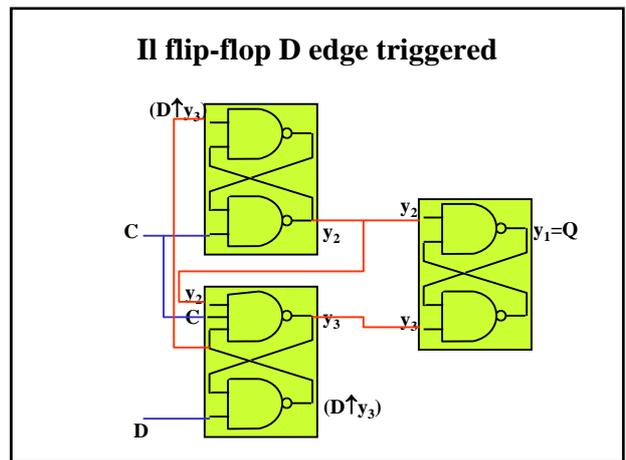
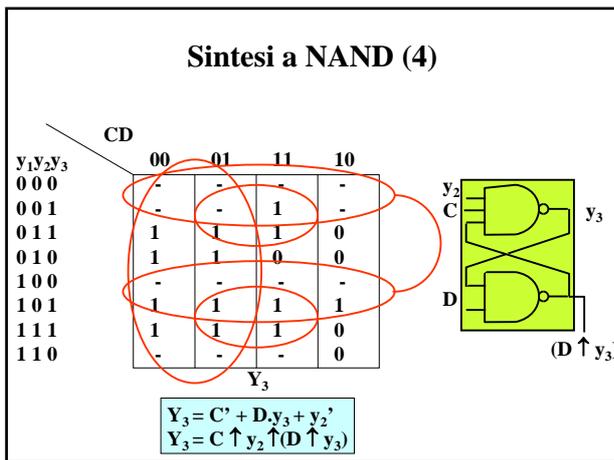
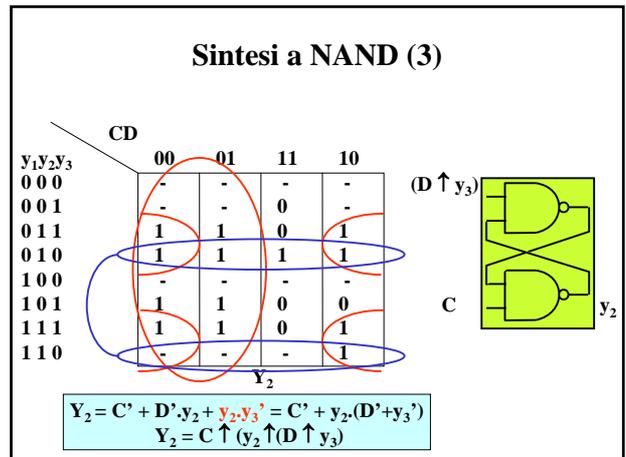
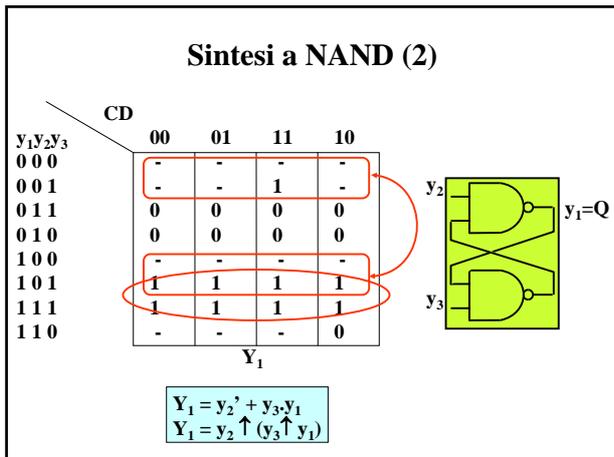


Sintesi a NAND (1)

$y_1 y_2 y_3$	CD			
	00	01	11	10
000	-	-	-	-
001	-	-	-	-
011	0	0	0	0
010	0	0	0	0
100	-	-	-	-
101	1	1	1	1
111	1	1	1	1
110	-	-	-	-

Q

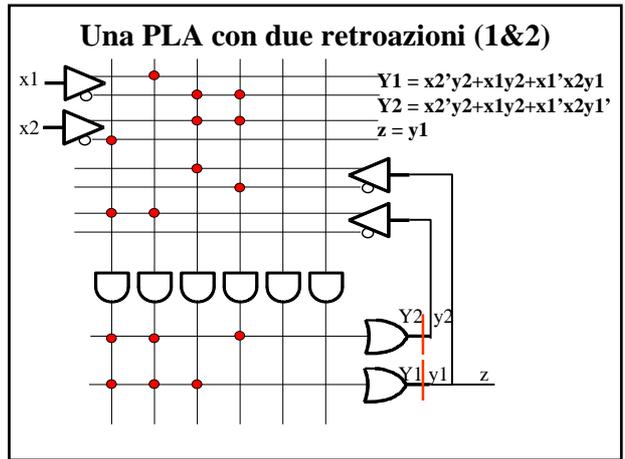
$Q = y_1$



6.3 Analisi e Sintesi

Analisi

- ### Il procedimento di analisi
- Il procedimento di analisi di una rete sequenziale asincrona è formato da 5 passi e consente di dedurre il comportamento dallo schema logico:
- 1: individuazione delle variabili di stato,
 - 2: analisi della parte combinatoria,
 - 3: individuazione della tabella delle transizioni,
 - 4: studio delle condizioni di stabilità,
 - 5: individuazione della tabella di flusso e del grafo degli stati.

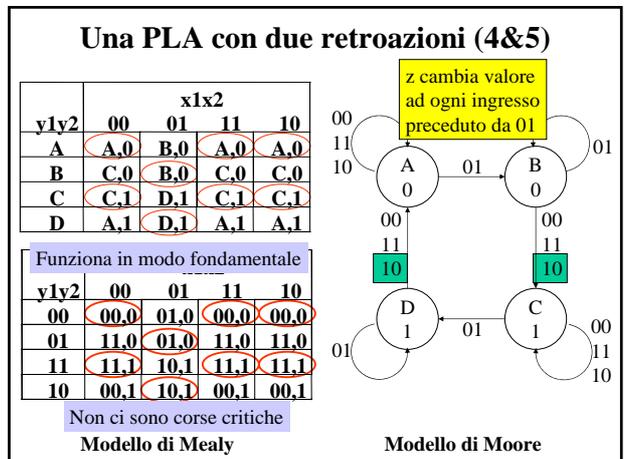


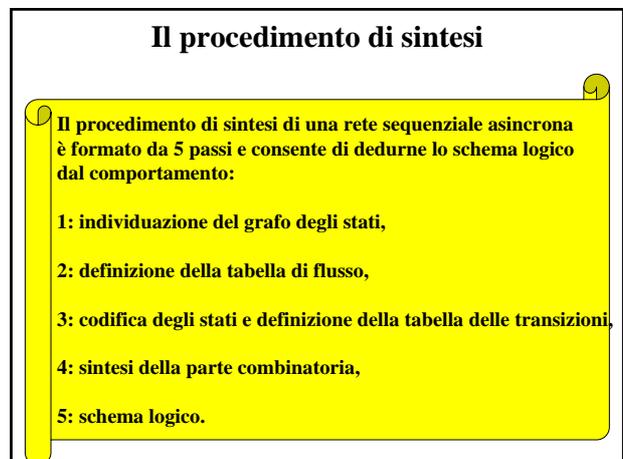
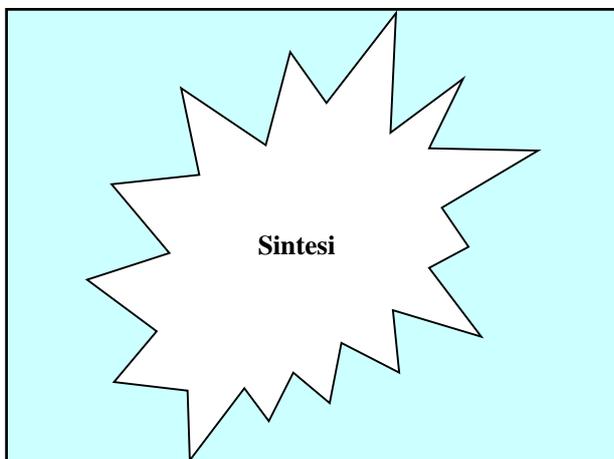
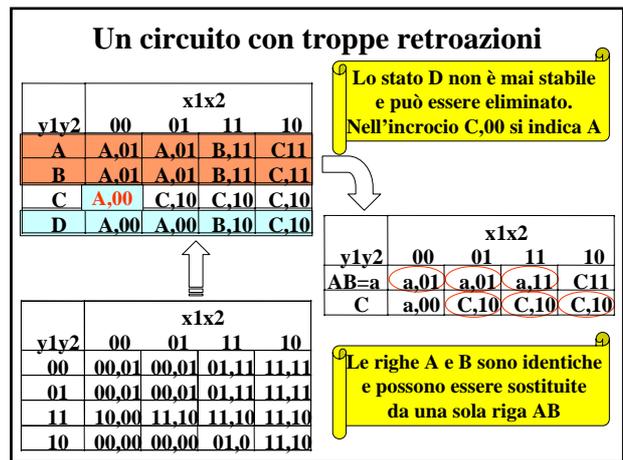
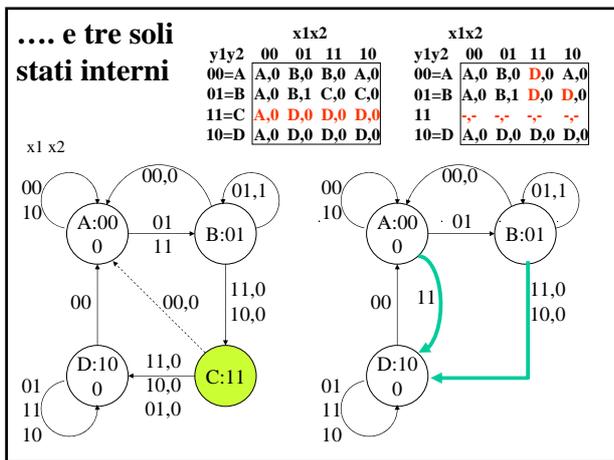
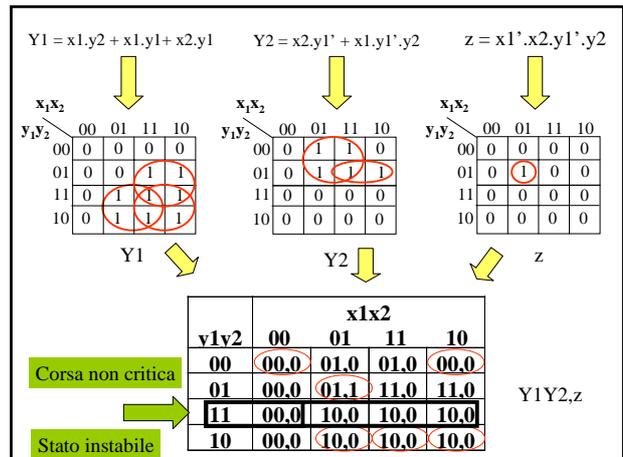
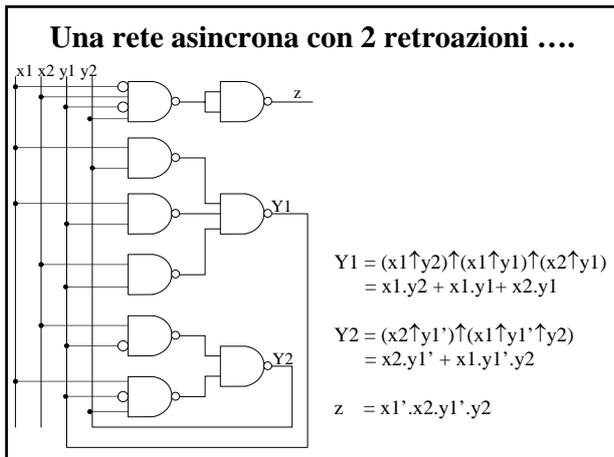
Una PLA con due retroazioni (3)

$Y_1 = x_2'y_2 + x_1y_2 + x_1'x_2y_1$
 $Y_2 = x_2'y_2 + x_1y_2 + x_1'x_2y_1'$
 $z = y_1$

y_1y_2	x_1x_2				Y_1	Y_2
	00	01	11	10		
00	0	0	0	0	0	0
01	1	0	1	1	1	1
11	1	1	1	1	1	1
10	0	1	0	0	1	0

Eliminazione alee statiche: y_1y_2 in Y_1 , $y_1'y_2$ in Y_2

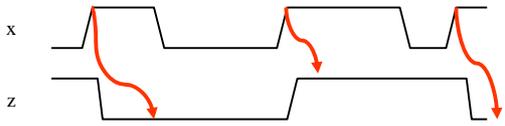




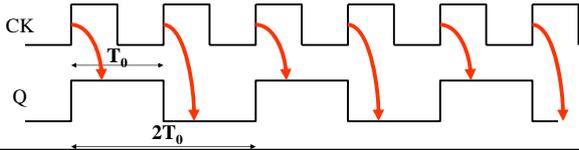
Esempio di sintesi

Comportamento: "z cambia valore ad ogni fronte di salita di x"

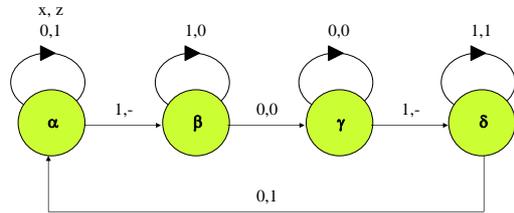
1 - Lampada da tavolo



2 - Divisore x2 della frequenza di un segnale periodico



Esempio (1: grafo degli stati)



Stabilità - Ogni stato è stabile per l'ingresso che lo genera.
Indifferenza sull'uscita - La modifica di uscita può avvenire già durante la transizione oppure, indifferentemente, essere rinviata al raggiungimento della stabilità.

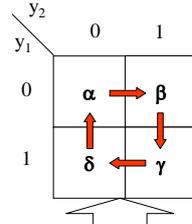
Esempio (2: tabella di flusso)

	x	
stato	0	1
α	α,1	β,-
β	γ,0	β,0
γ	γ,0	δ,-
δ	α,1	δ,1

CONTROLLI FORMALI

- In ogni riga ci deve essere almeno una condizione di stabilità.
- In ogni colonna si deve raggiungere sempre una stabilità.
- Le situazioni di instabilità devono indicare uno stato futuro stabile nella colonna (assenza di transizioni multiple).

Esempio (3: tabella delle transizioni)



Grafo delle adiacenze e mappa di codifica

	x	
y1,y2	0	1
α:00	00,1	01,-
β:01	11,0	01,0
γ:11	11,0	10,-
δ:10	00,1	10,1

Y₁, Y₂, z

Codifica degli stati - A stati consecutivi (stato presente e futuro) si devono assegnare configurazioni adiacenti.

Esempio (4: espressioni)

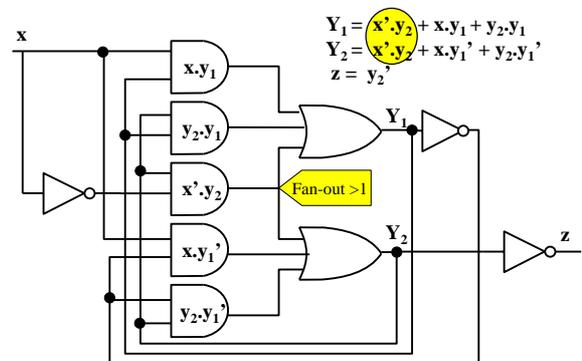
Ipotesi: si desiderano reti minime di tipo SP

	y ₁ y ₂			
x	00	01	11	10
0	0	1	1	0
1	0	0	1	1

$$Y_1 = x' \cdot y_2 + x \cdot y_1 + y_2 \cdot y_1 \quad Y_2 = x' \cdot y_2 + x \cdot y_1' + y_2 \cdot y_1' \quad z = y_2'$$

Coperture ridondanti - Per eliminare a priori il pericolo di alea statica (qui causato dal probabile sfasamento tra x e x') "ogni coppia di 1 adiacenti deve essere racchiusa in almeno un RR"; se occorre un RR "in più", si deve dargli dimensione massima.

Esempio (5: schema con retroazioni dirette)

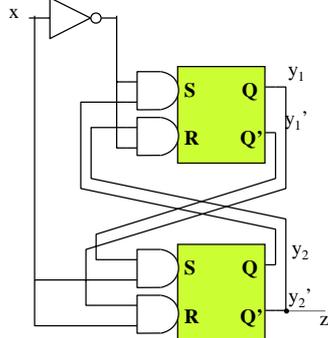


Esempio (5: schema con latch SR)

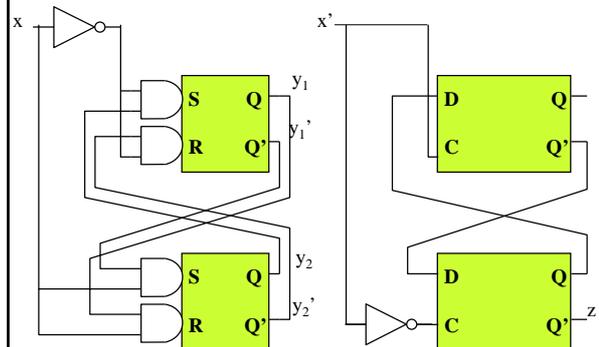
$$\begin{aligned}
 Y_1 &= x' \cdot y_2 + x \cdot y_1 + y_2 \cdot y_1 \\
 &= (x' \cdot y_2) + (x + y_2) \cdot y_1 \\
 &= (x' \cdot y_2) + (x' \cdot y_2)' \cdot y_1 \\
 &= S_1 + R_1' \cdot y_1
 \end{aligned}$$

$$\begin{aligned}
 Y_2 &= x' \cdot y_2 + x \cdot y_1' + y_2 \cdot y_1' \\
 &= x \cdot y_1' + (x' + y_1') \cdot y_2 \\
 &= x \cdot y_1' + (x \cdot y_1)' \cdot y_2 \\
 &= S_2 + R_2' \cdot y_2
 \end{aligned}$$

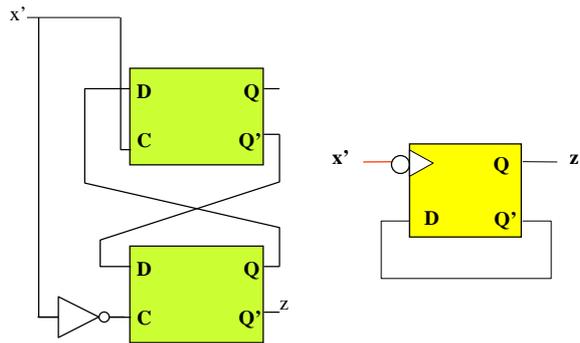
$$z = y_2'$$



Esempio (5: schema con latch CD)



Esempio (5: schema con flip-flop)



Grafi primitivi e non primitivi

Grafo primitivo

Grafo degli stati primitivo – Grafo in cui ogni stato è stabile per una ed una sola configurazione d'ingresso.

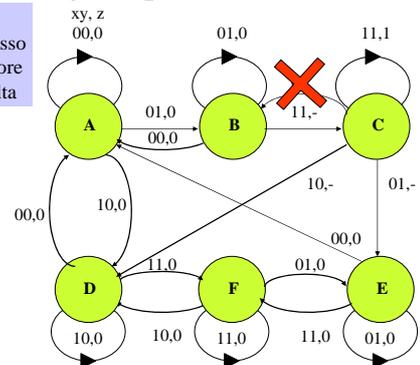
Per individuare le esigenze di stati interni poste dalla specifica di comportamento è spesso utile iniziare il progetto con un **grafo primitivo**.

Di norma il grafo primitivo **non ha il minimo numero possibile di stati interni**

Nota la tabella di flusso primitiva (una sola stabilità per riga), è abbastanza agevole individuare l'**automa minimo**

Riconoscitore della sequenza 00-01-11 grafo primitivo

IPOTESI
I segnali d'ingresso cambiano di valore uno solo alla volta



Riconoscitore della sequenza 00-01-11 grafo non primitivo

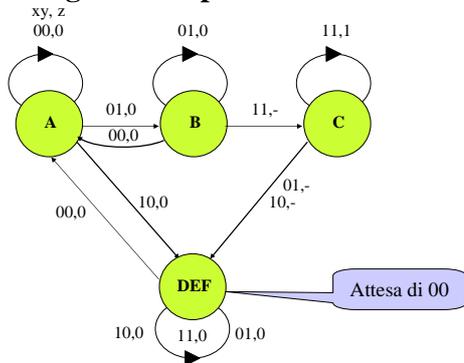


Tabelle di flusso

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

	00	01	11	10
A	A,0	B,0	-,-	S,0
B	A,0	B,0	C,-	-,-
C	-,-	S,-	C,1	S,-
S	A,0	S,0	S,0	S,0

	00	01	11	10
I	I,0	I,0	C,-	S,0
C	-,-	S,-	C,1	S,-
S	I,0	S,0	S,0	S,0

Tabella primitiva

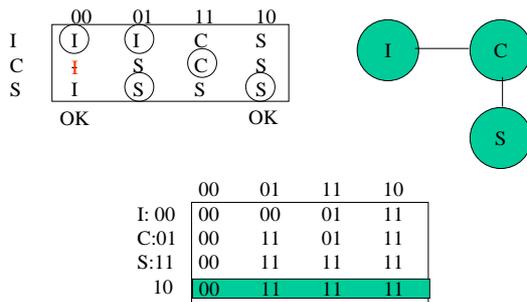
Tabella ridotta

Tabella minima

Due o più righe (stati) di una tabella di flusso possono sostituire da un'unica riga se, **in ogni colonna** (per ogni ingresso),

- le uscite sono **identiche** quando sono specificate
- gli stati futuri, quando sono specificati, sono o **identici**, o quelli corrispondenti alle **righe in esame**, o quelli di **righe che soddisfano questa condizione**.

Codifica degli stati



Stato "in più" 10: instabile per autoinizializzazione