

Capitolo 6

Reti asincrone

6.1 – Elaborazione asincrona

6.2 – Memorie binarie

6.3 – Analisi e Sintesi

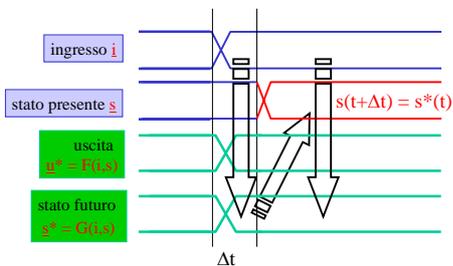
6.1

Elaborazione asincrona

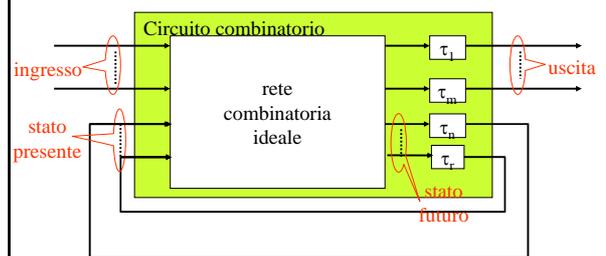
Reti sequenziali asincrone (comportamento)

Elaborazione asincrona - Ogni nuovo ingresso determina:

- una condizione di stabilità dello stato interno
- non più di una modifica del simbolo d'uscita



Reti sequenziali asincrone (struttura)



- Il **ritardo intrinseco** del circuito agisce da "memoria temporanea": lo stato presente non cambia durante il calcolo dello stato futuro.
- Il piccolo valore dei ritardi consente di aggiornare lo stato ad ingresso costante
- Con k retroazioni si dispone di 2^k riassunti di storia passata.

Esigenze di elaborazioni asincrone

Problema - **Discriminare e ricordare** l'ordine temporale con cui due o più segnali binari modificano il loro valore

Soluzione - **macchina seq. asincrona**

Sensori intelligenti

Flip-flop & Reti sincrone

Buffer & RAM

Gestione delle risorse

Proprietà - **Velocità nel prendere decisioni**

Pericolo di malfunzionamento e vincoli di corretto impiego

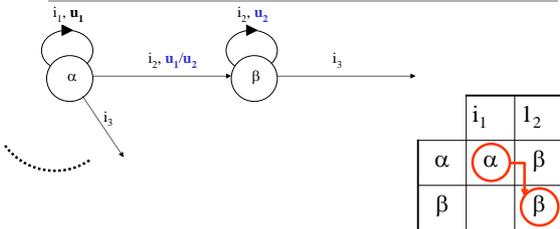
Regole di corretto impiego

Obiettivo: ottenere un funzionamento corretto per qualsiasi valore dei ritardi presenti sugli anelli di retroazione.

- Durata degli ingressi
- Codifica degli ingressi
- Codifica degli stati
- Alee statiche

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 : $10 \Rightarrow 01$

$10 \Rightarrow 11 \Rightarrow 01$

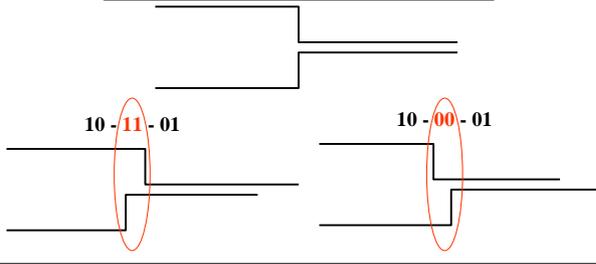
$10 \Rightarrow 00 \Rightarrow 01$

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

3. Codifica degli stati

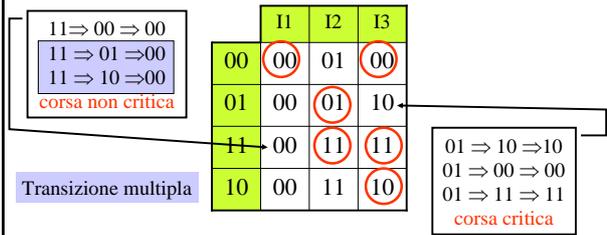
Codifica degli stati interni – Per eliminare a priori la presenza di stati interni "spuri" le config. Condizione restrittiva lo stato presente ed a tutti i suoi possibili stati Vincolo eccessivo per il valore di un solo bit

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.



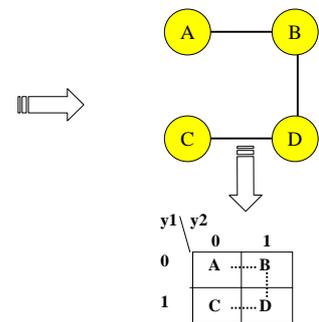
Eliminazione a priori delle corse critiche (1)

1. Nelle colonne con **una sola stabilità** si inserisce il simbolo dello stato stabile al posto di eventuali condizioni d'indifferenza
2. 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.
3. 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**
4. (segue)

Una codifica priva di corse critiche

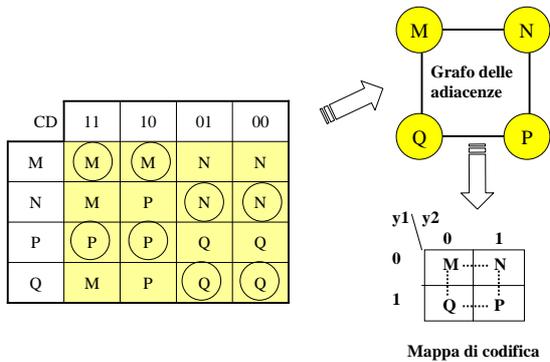
	I1	I2	I3
A	A	B	A
B	A	B	D
C	A	C	C
D	A	C	D

OK



Mappa di codifica

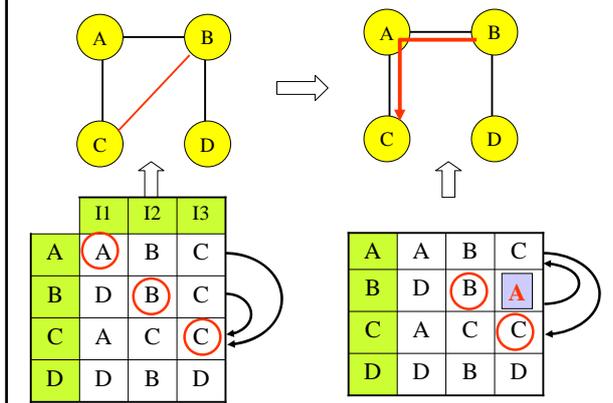
Codifica degli stati per il flip-flop (pag. 40)



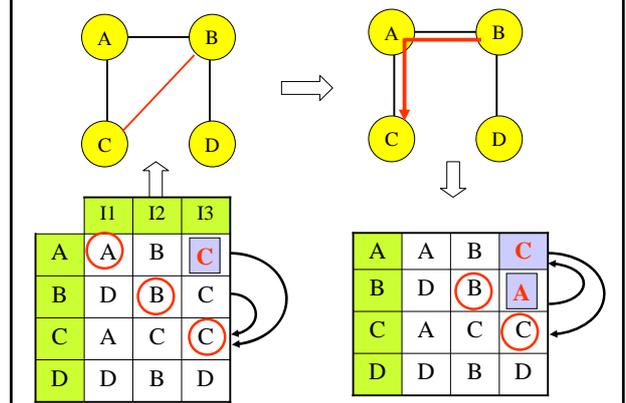
Eliminazione a priori delle corse critiche

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.

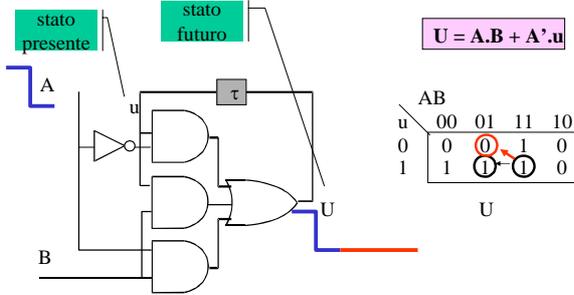
Transizioni multiple (1)



Transizioni multiple (2)



Eliminazione a priori delle alee statiche

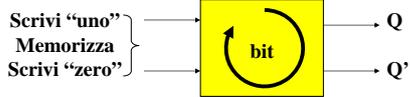


$$U = A.B + A'.u$$

Sulle retroazioni si deve sempre rispettare T14



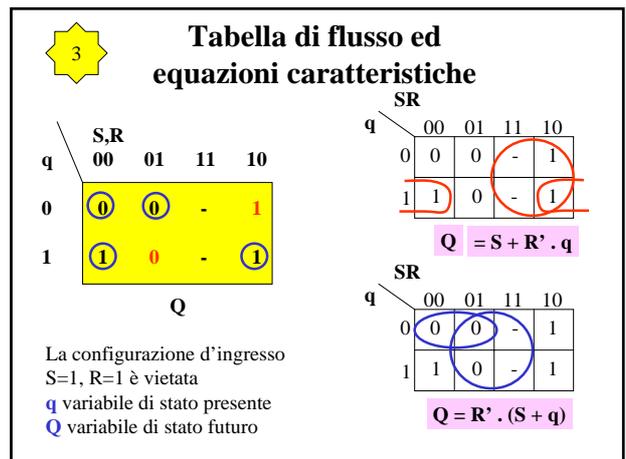
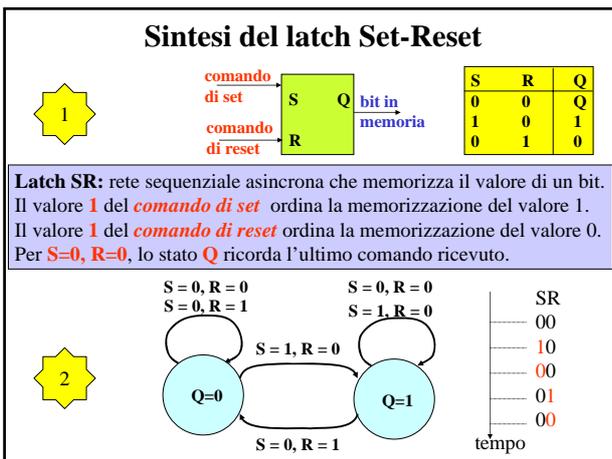
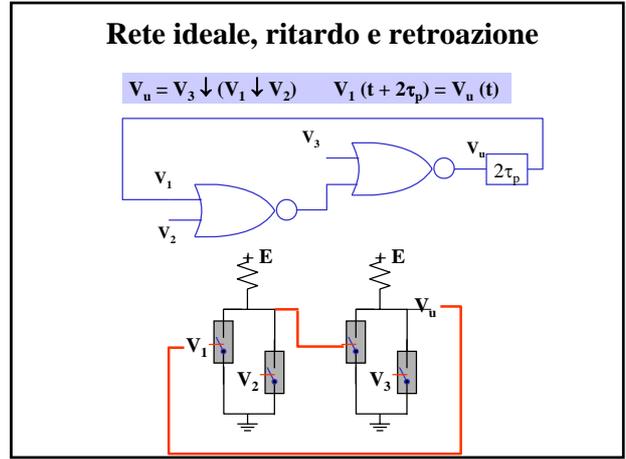
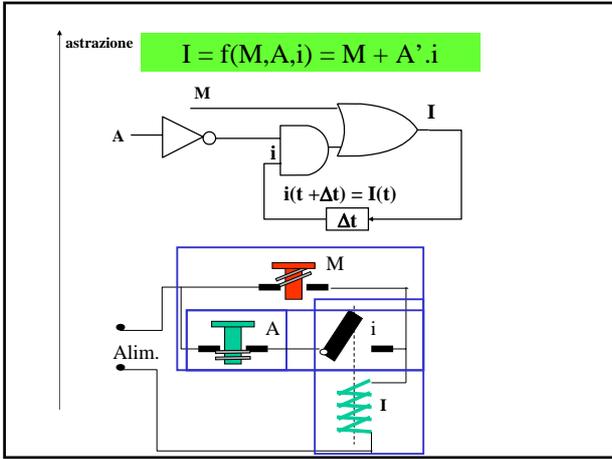
Memorie binarie

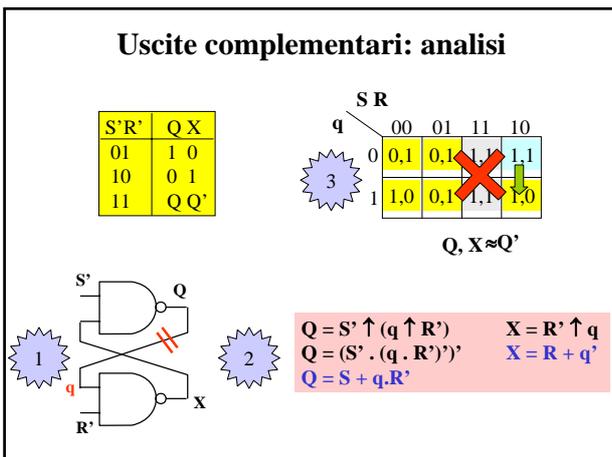
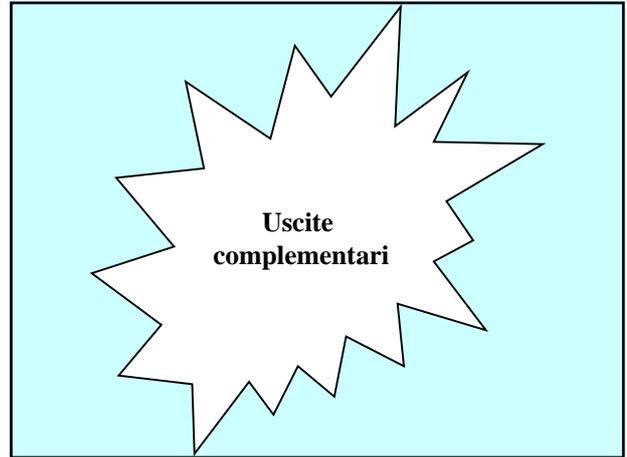
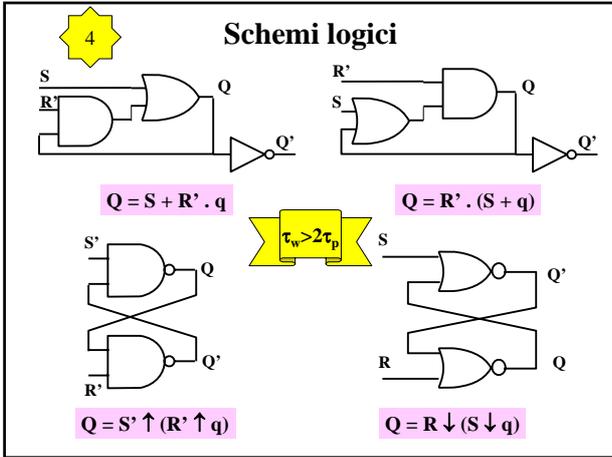


Cosa scrivere e **quando** scrivere ⇒ sequenza d'ingresso

- | | | |
|-------------------------|--|------------------|
| ↓ | <ul style="list-style-type: none"> • Latch SR • Latch CD • Flip-flop edge-triggered | ↓ |
| complessità strutturale | | Semplicità d'uso |



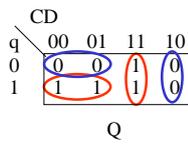




Una memoria binaria più utile

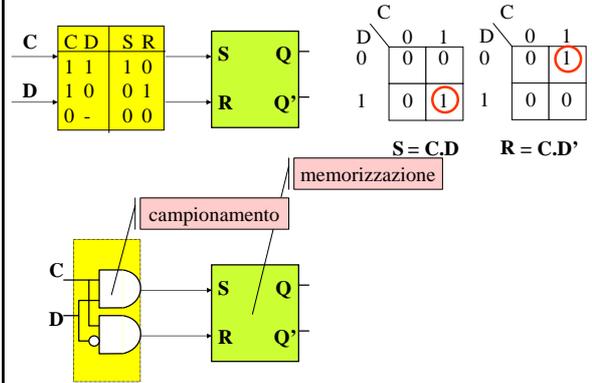
Latch CD - Memoria binaria in cui il valore 1 di un segnale **C** individua "quando scrivere", il valore contemporaneo di un segnale **D** "cosa scrivere".

C	D	Q
1	1	1
1	0	0
0	-	Q

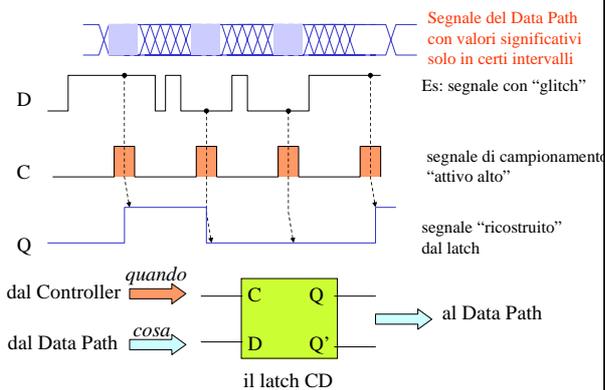


Equazioni caratteristiche
 $Q = C.D + C'.q$
 $Q = (C + q).(C' + D)$

Dal latch SR al latch CD



Campionamento e Memorizzazione



Problemi del latch CD

- Alea statica
- Durata del transitorio
- Uscite trasparenti



Alea statica nel latch CD

		CD			
q		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

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

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

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

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

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

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

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

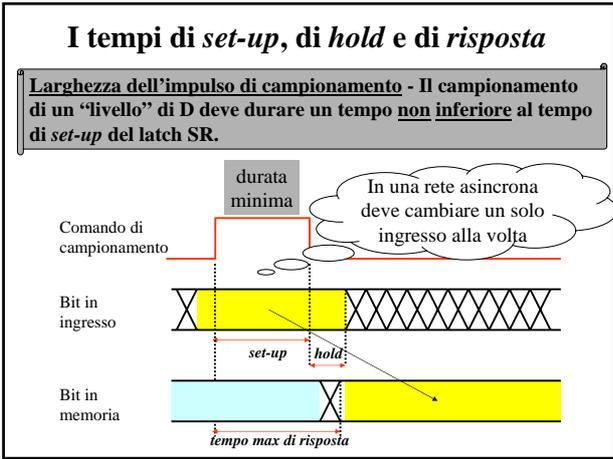
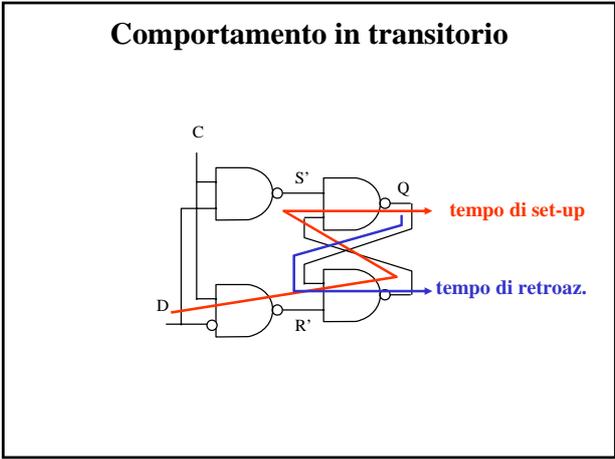
Una quarta soluzione

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

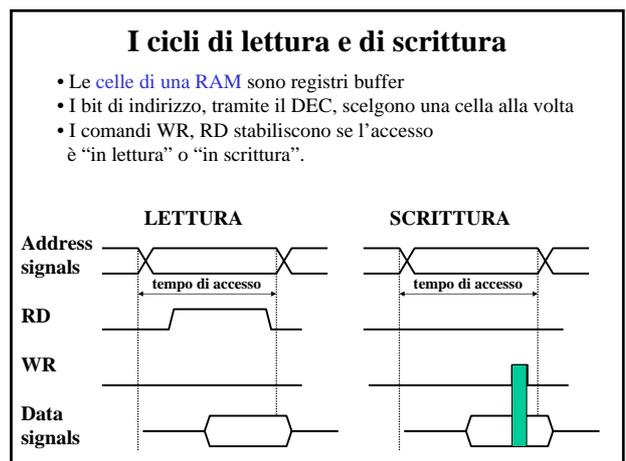
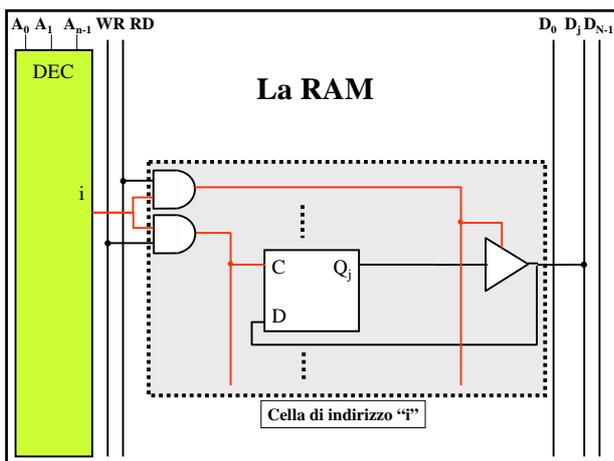
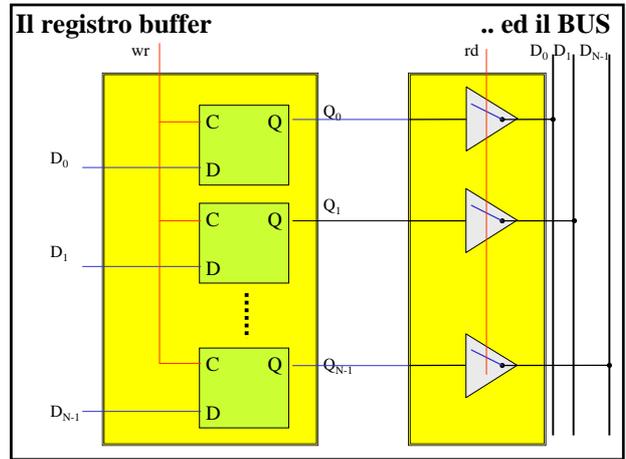
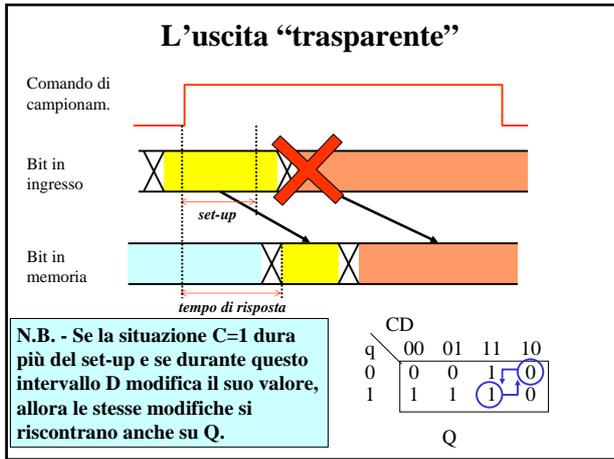
$$Q = C.D + C'.q$$

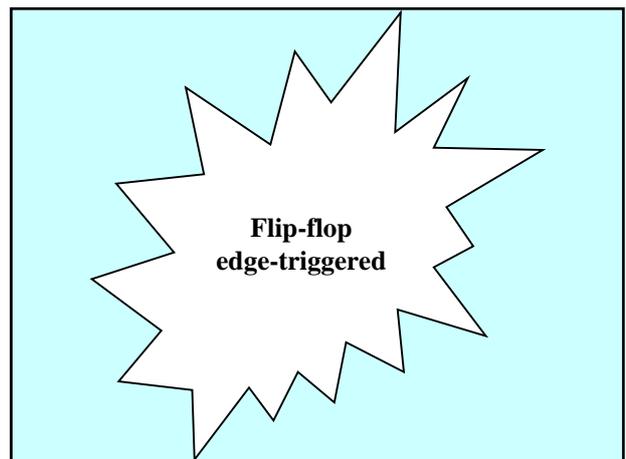
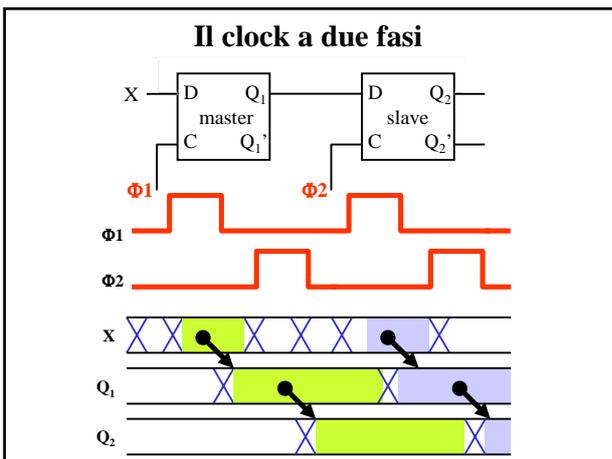
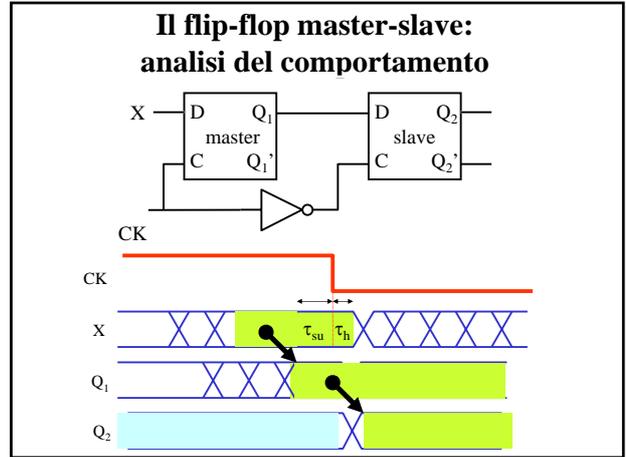
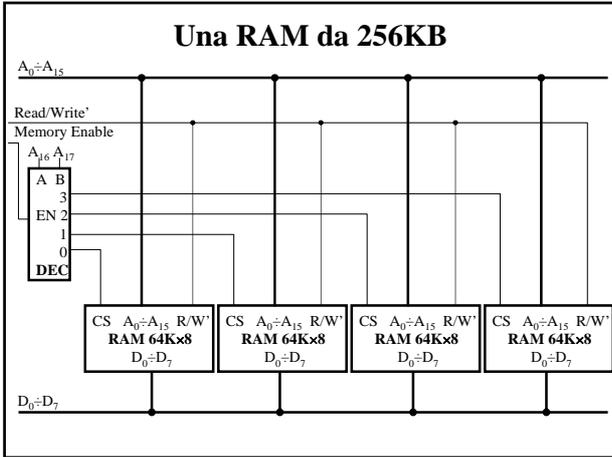
N.B. - Il ritardo inerziale dei due NOT elimina l'eventuale "glitch" generato dall'alea statica del MUX .

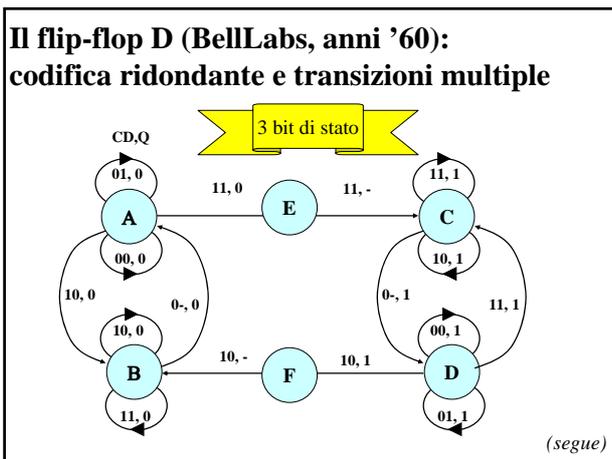
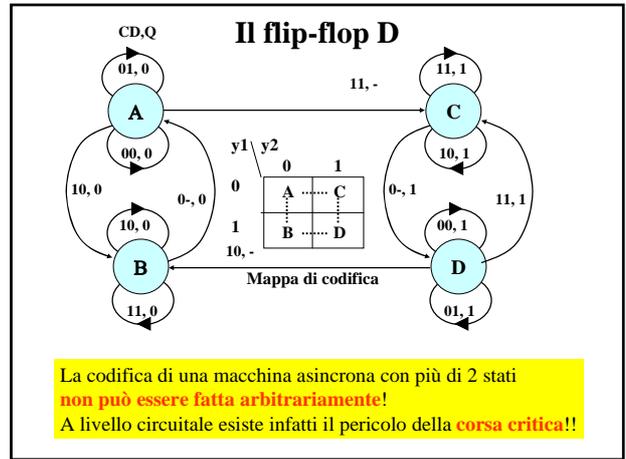
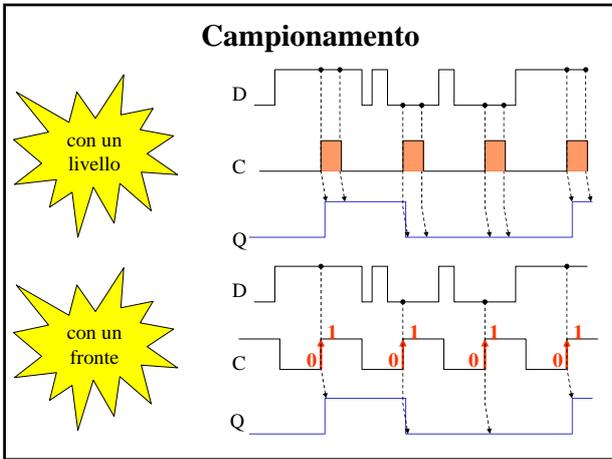
Il transitorio



Uscite trasparenti







La tabella delle transizioni

stato	CD			
	00	01	11	10
000	---	---	---	---
E = 001	---	---	101,-	---
A = 011	011,0	011,0	001,0	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)

Realizzazione a NAND

Sintesi a NAND (1)

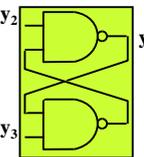
CD	00	01	11	10
$y_1y_2y_3$				
000	-	-	-	-
001	-	-	-	-
011	0	0	0	0
010	0	0	0	0
100	-	-	-	-
101	1	1	1	1
111	1	1	1	1
110	-	-	-	-

Q

$Q = y_1$

Sintesi a NAND (2)

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

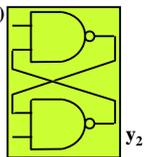


$y_1 = Q$

$Y_1 = y_2' + y_3 \cdot y_1$
 $Y_1 = y_2 \uparrow (y_3 \uparrow y_1)$

Sintesi a NAND (3)

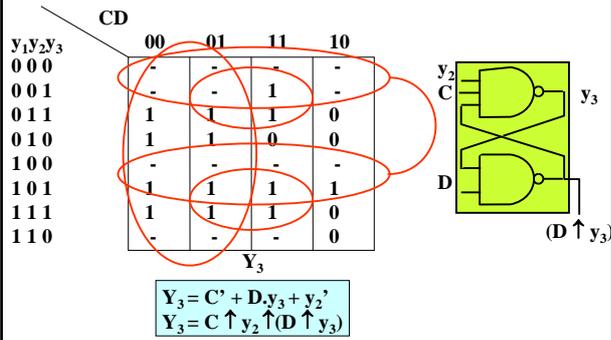
CD	00	01	11	10
$y_1y_2y_3$				
000	-	-	-	-
001	-	-	0	-
011	1	1	0	1
010	1	1	1	1
100	-	-	-	-
101	1	1	0	0
111	1	1	0	1
110	-	-	-	1



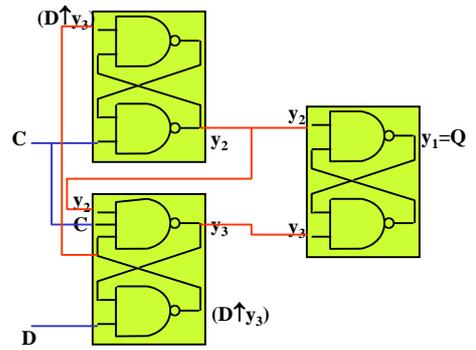
$(D \uparrow y_3)$
 C
 y_2

$Y_2 = C' + D' \cdot y_2 + y_2 \cdot y_3' = C' + y_2 \cdot (D' + y_3')$
 $Y_2 = C \uparrow (y_2 \uparrow (D \uparrow y_3))$

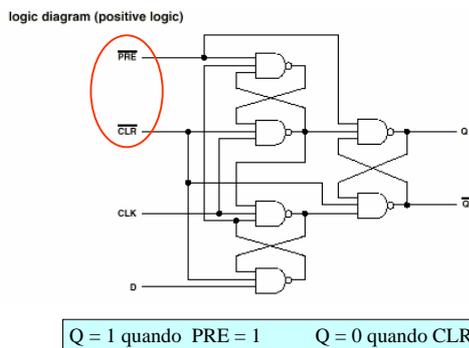
Sintesi a NAND (4)



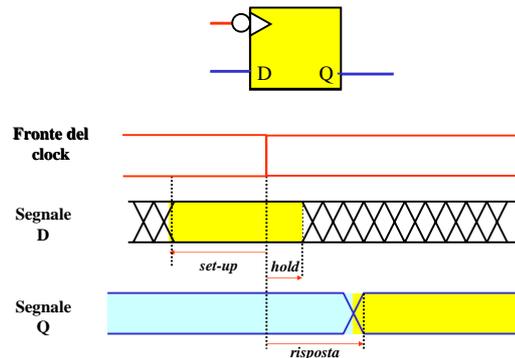
Il flip-flop D edge triggered



Il flip-flop di tipo D della famiglia TTL: comandi sincroni e asincroni ('74)



I tempi di *set-up*, di *hold* e di *risposta*



6.3 Analisi e Sintesi

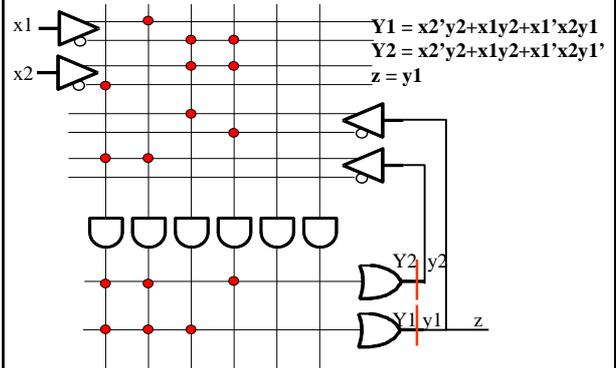
Analisi

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 (1&2)



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

Y₁

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

Y₂

Eliminazione alee statiche: y₁y₂ in Y₁, y₁'y₂ in Y₂

Una PLA con due retroazioni (4&5)

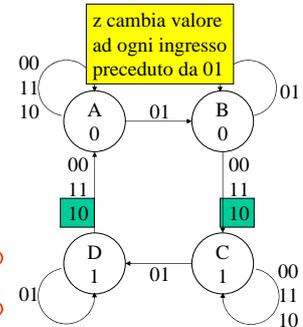
y ₁ y ₂	x ₁ x ₂			
	00	01	11	10
A	A,0	B,0	A,0	A,0
B	C,0	B,0	C,0	C,0
C	C,1	D,1	C,1	C,1
D	A,1	D,1	A,1	A,1

Funziona in modo fondamentale

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

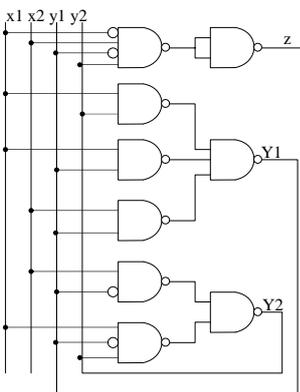
Non ci sono corse critiche

Modello di Mealy



Modello di Moore

Una rete asincrona con 2 retroazioni



$$Y_1 = (x_1 \uparrow y_2) \uparrow (x_1 \uparrow y_1) \uparrow (x_2 \uparrow y_1)$$

$$= x_1.y_2 + x_1.y_1 + x_2.y_1$$

$$Y_2 = (x_2 \uparrow y_1') \uparrow (x_1 \uparrow y_1' \uparrow y_2)$$

$$= x_2.y_1' + x_1.y_1'.y_2$$

$$z = x_1'.x_2.y_1'.y_2$$

$$Y_1 = x_1.y_2 + x_1.y_1 + x_2.y_1 \quad Y_2 = x_2.y_1' + x_1.y_1'.y_2 \quad z = x_1'.x_2.y_1'.y_2$$

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

Y₁

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

Y₂

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

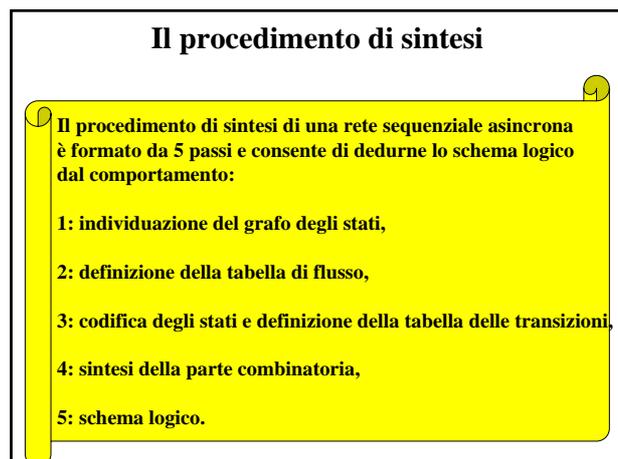
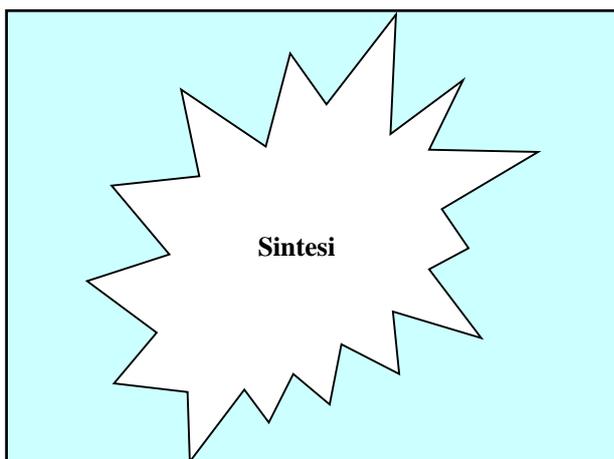
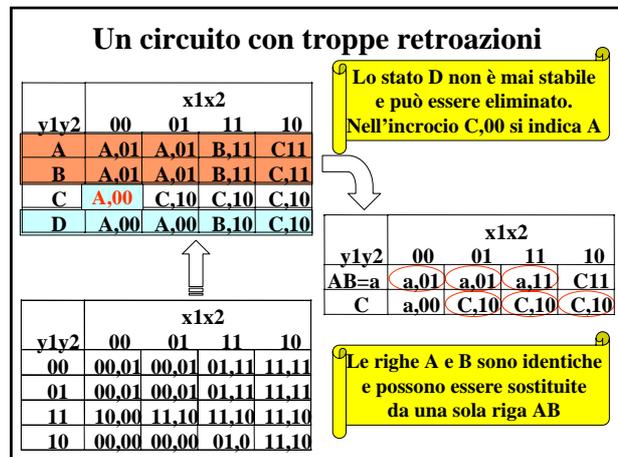
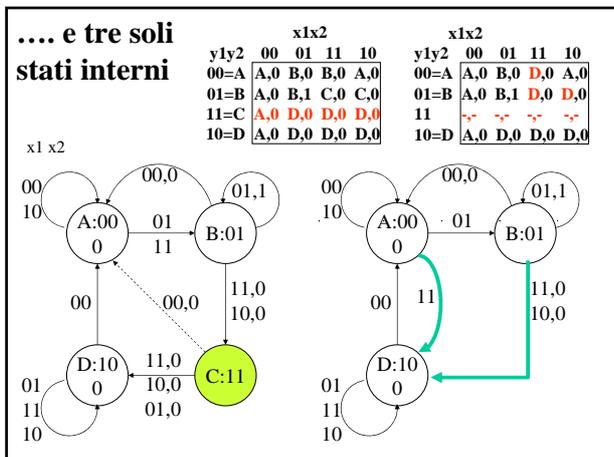
z

Corsa non critica

Stato instabile

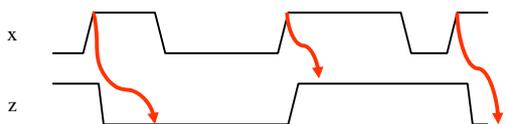
y ₁ y ₂	x ₁ x ₂			
	00	01	11	10
00	00,0	01,0	01,0	00,0
01	00,0	01,1	11,0	11,0
11	00,0	10,0	10,0	10,0
10	00,0	10,0	10,0	10,0

Y₁Y₂,z

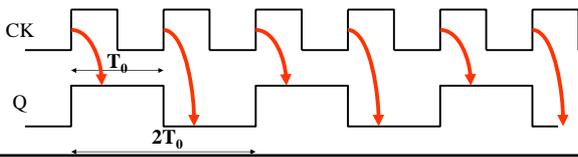


Esempio di sintesi

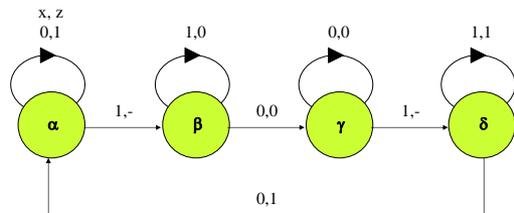
Comportamento: "z cambia valore ad ogni fronte di salita di x"
 1 - Lampada da tavolo



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.

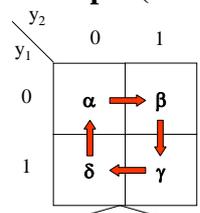
Esempio (2: tabella di flusso)

stato	x	
	0	1
α	$\alpha, 1$	$\beta, -$
β	$\gamma, 0$	$\beta, 0$
γ	$\gamma, 0$	$\delta, -$
δ	$\alpha, 1$	$\delta, 1$

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

y_1, y_2 \ x	0	1
	$\alpha: 00$	00,1
$\beta: 01$	11,0	01,0
$\gamma: 11$	11,0	10,-
$\delta: 10$	00,1	10,1

Y_1, Y_2, 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

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

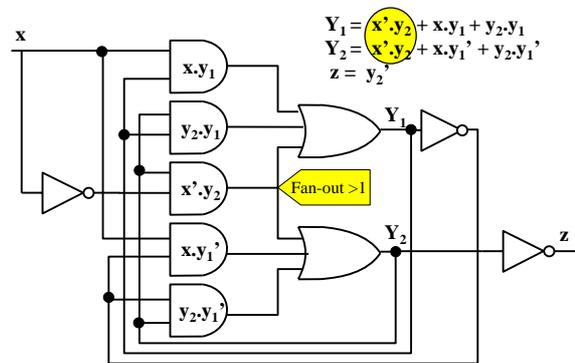
$y_1 y_2$	00	01	11	10
0	0	0	1	1
1	1	1	0	0

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

$$Y_1 = x' \cdot y_2 + x \cdot y_1 + y_2 \cdot y_1$$

$$= (x' \cdot y_2) + (x + y_2) \cdot y_1$$

$$= (x' \cdot y_2) + (x' \cdot y_2)' \cdot y_1$$

$$= S_1 + R_1' \cdot y_1$$

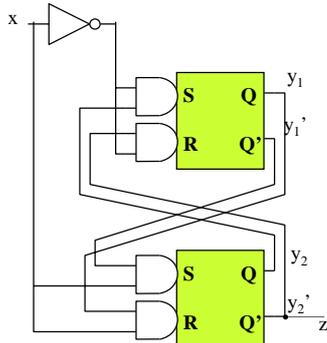
$$Y_2 = x' \cdot y_2 + x \cdot y_1' + y_2 \cdot y_1'$$

$$= x \cdot y_1' + (x' + y_1') \cdot y_2$$

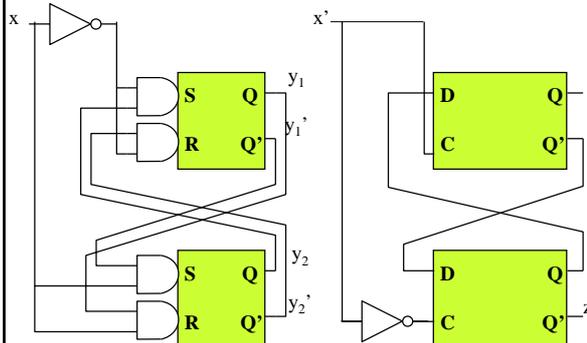
$$= x \cdot y_1' + (x \cdot y_1)' \cdot y_2$$

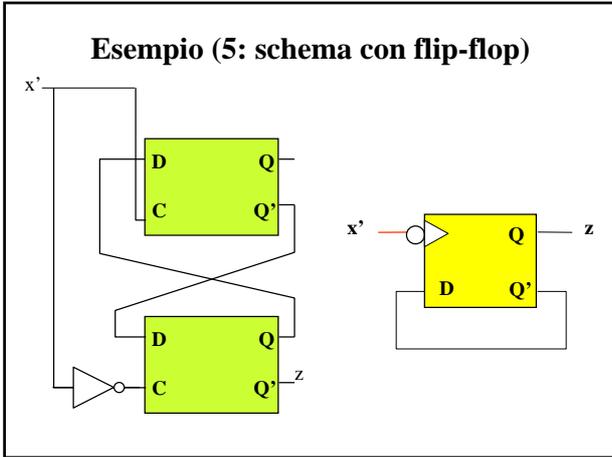
$$= S_2 + R_2' \cdot y_2$$

$$z = y_2'$$



Esempio (5: schema con latch CD)





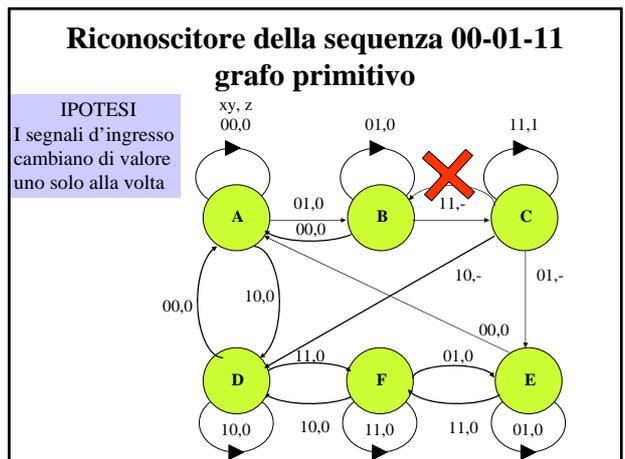
Grafo primitivo

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 non primitivo

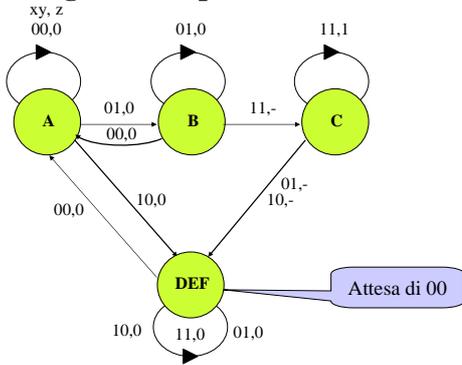


Tabelle di flusso

	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

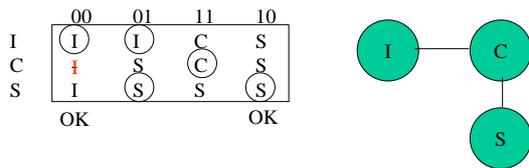
Tabella ridotta

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



	00	01	11	10
I: 00	00	00	01	11
C: 01	00	11	01	11
S: 11	00	11	11	11
10	00	11	11	11

Stato "in più" 10: instabile per autoinizializzazione