

Capitolo 6

Reti asincrone

6.1 – Elaborazione asincrona

6.2 – Memorie binarie

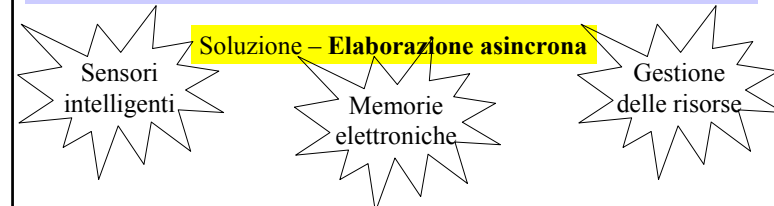
6.3 – Analisi e Sintesi

6.1 Elaborazione asincrona

Esigenze e vincoli

Esigenze

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



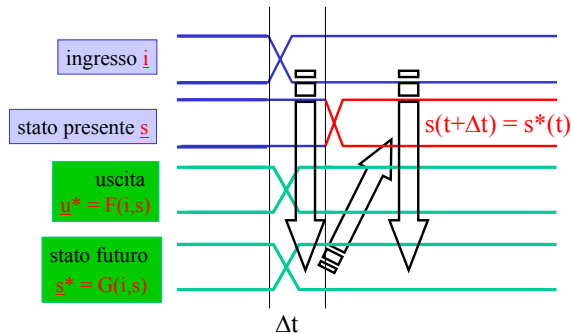
Proprietà – **Velocità nel prendere decisioni**

Pericolo di malfunzionamento
e vincoli di corretto impiego

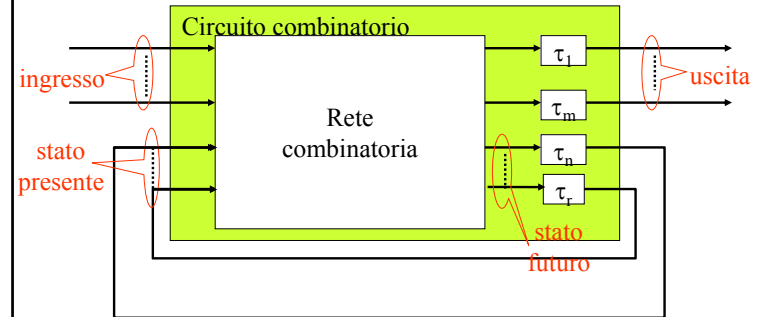
Reti sequenziali asincrone (comportamento)

Elaborazione asincrona - Ogni nuovo ingresso determina:

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



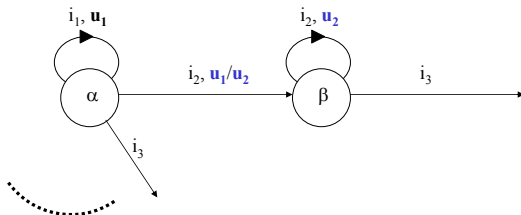
Reti sequenziali asincrone (struttura)



- Il ritardo intrinseco del circuito agisce da “memoria temporanea” dello stato presente durante il calcolo dello stato futuro.
- Con k retroazioni si dispone di 2^k riassunti di storia passata.

Vincoli di corretto impiego (1)

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

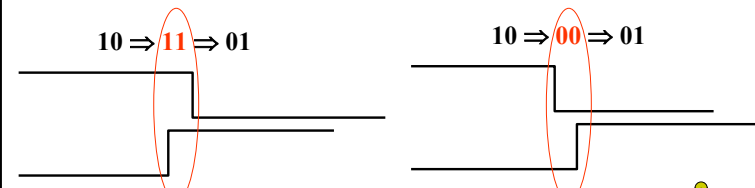


Il passaggio da una ad un'altra condizione di stabilità è detto funzionamento in **modo fondamentale**

Vincoli di corretto impiego (2)

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

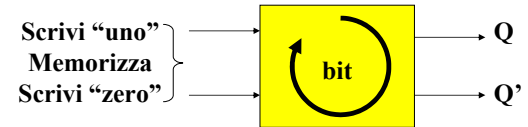
Esempio di situazione d'ingresso pericolosa :
10 ⇒ 01



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

6.2 Memorie binarie

Memorie binarie



Cosa scrivere e quando scrivere \Rightarrow sequenza d'ingresso

↓
complessità
strutturale

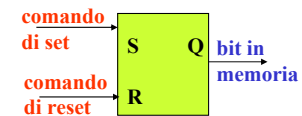
- Latch SR
- Latch CD
- Flip-flop edge-triggered

↓
Semplicità
d'uso

Latch SR

Sintesi del latch Set-Reset

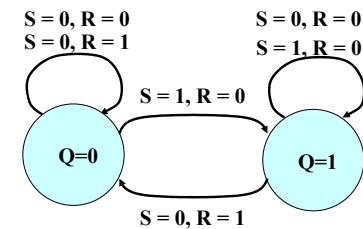
1



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.

2



SR
— 00
— 10
— 00
— 01
— 00
tempo

3

Tabella di flusso ed equazioni caratteristiche

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

Q

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

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

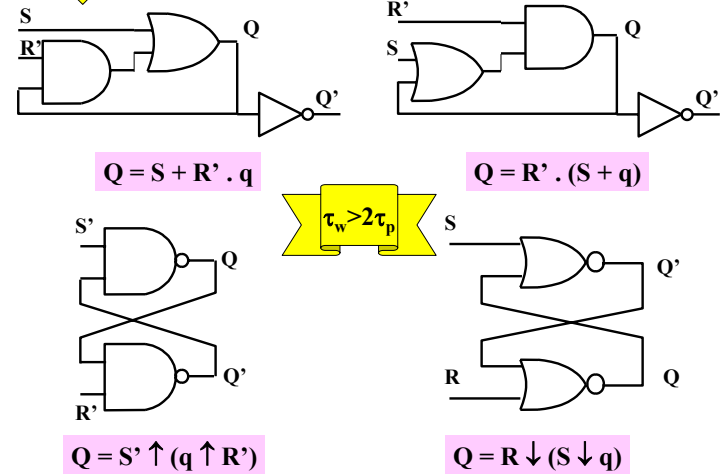
Q = S + R' . q

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

Q = R' . (S + q)

4

Schemi logici



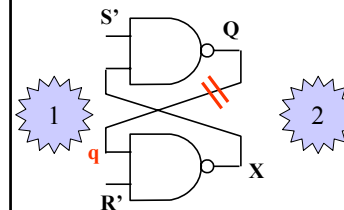
Uscite
complementari

Analisi del latch SR a NAND

S'R'	Q X
01	1 0
10	0 1
11	Q Q'

q	SR			
	00	01	11	10
0	0,1	0,1	1,1	1,1
1	1,0	0,1	1,1	1,0

Q, X ≈ Q'



$$\begin{aligned}
 Q &= S' \uparrow (q \uparrow R') & X &= R' \uparrow q \\
 Q &= (S' \cdot (q \cdot R'))' & X &= R + q' \\
 Q &= S + q \cdot R'
 \end{aligned}$$

Latch CD

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

	00	01	11	10
q	0	0	1	0
Q	1	1	1	0

Equazioni caratteristiche

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

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

Dal latch SR al latch CD

C	D	S	R
1	1	1	0
1	0	0	1
0	-	0	0

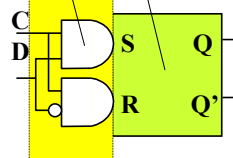
memorizzazione

campionamento

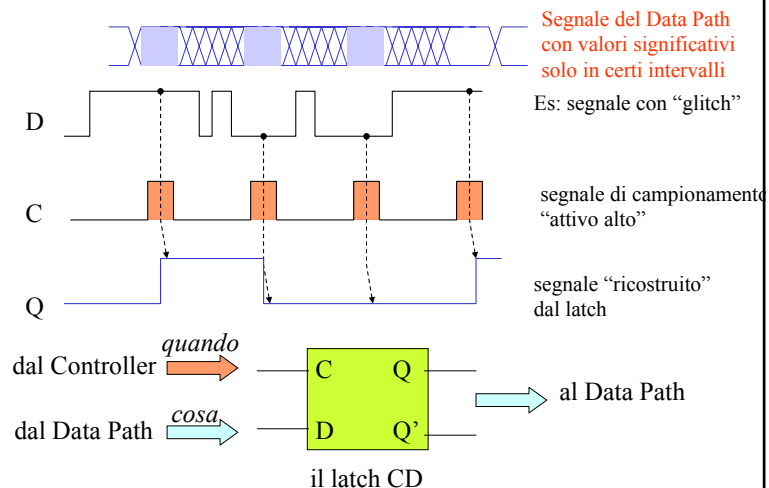
D	C	0	1
0	0	0	0
1	0	0	1

$$S = C.D$$

$$R = C.D'$$



Campionamento e Memorizzazione



Problemi del latch CD

- Alea statica
- Durata del transitorio
- Uscite trasparenti

Alea statica

Alea statica nel latch CD

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

Q

Realizzazione con latch SR

$$\begin{aligned}
 Q &= S + R'.q \\
 &= C.D + (C.D')'.q \\
 &= C.D + C'.q + \cancel{D.q}
 \end{aligned}$$

Equazioni caratteristiche “ridondanti”

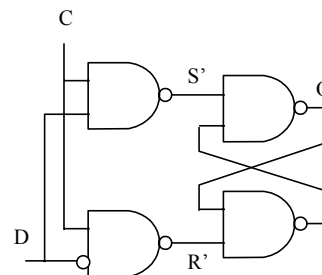
$$Q = C.D + C'.q + D.q = C.D + (C' + D).q$$

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

Il latch CD a NAND ed a NOR

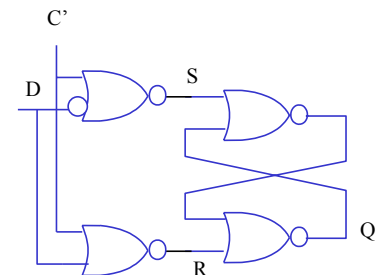
Realizzazione a NAND

$$\begin{aligned}
 Q &= C.D + (C' + D).q \\
 &= (C \uparrow D) \uparrow ((C \uparrow D)' \uparrow q)
 \end{aligned}$$



Realizzazione a NOR

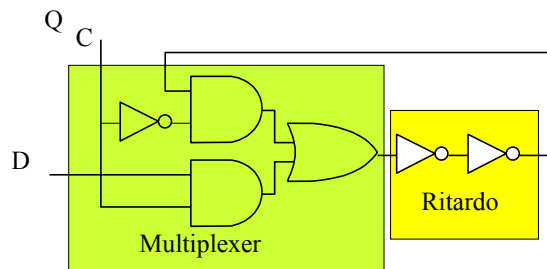
$$\begin{aligned}
 Q &= (C' + D).((C.D) + q) \\
 &= (C' \downarrow D) \downarrow ((C' \downarrow D)' \downarrow q)
 \end{aligned}$$



Una quarta soluzione

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

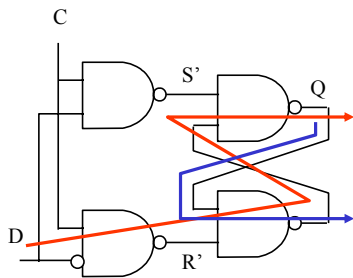
$$Q = C.D + C'.q$$



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

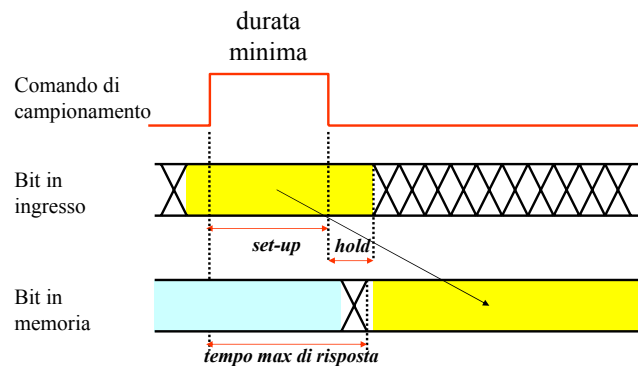
Il transitorio

Comportamento in transitorio



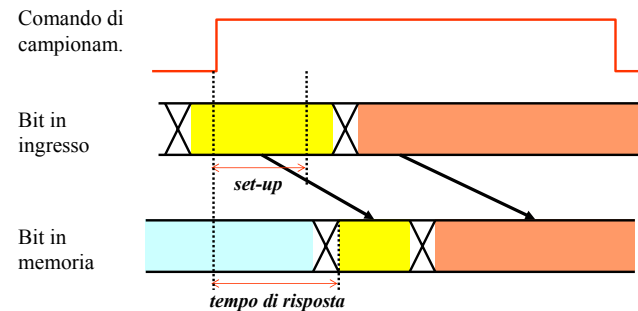
I tempi di *set-up*, di *hold* e di *risposta*

Larghezza dell'impulso di campionamento - Il campionamento di un "livello" di D deve durare un tempo non inferiore al tempo di *set-up* del latch SR.



Uscite trasparenti

L'uscita "trasparente"



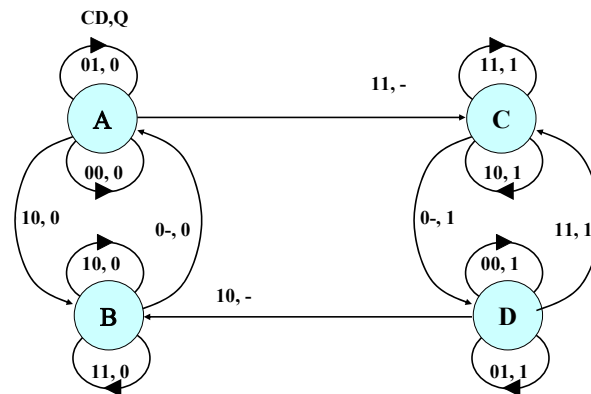
N.B. - Se la situazione $C=1$ dura più del set-up e se durante questo intervallo D modifica il suo valore, allora le stesse modifiche si riscontrano anche su Q.

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

Q

Flip-flop edge-triggered

Il grafo degli stati del flip-flop D



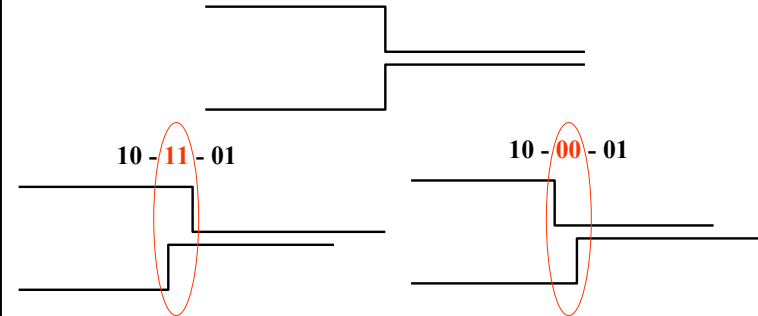
La codifica di più di 2 stati non può essere fatta arbitrariamente!
A livello circuitale esiste infatti il pericolo della **corsa critica**!!

Corse critiche

Vincoli sulla codifica degli stati

Codifica degli stati interni - Le configurazioni associate a stati interni consecutivi devono differire 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.

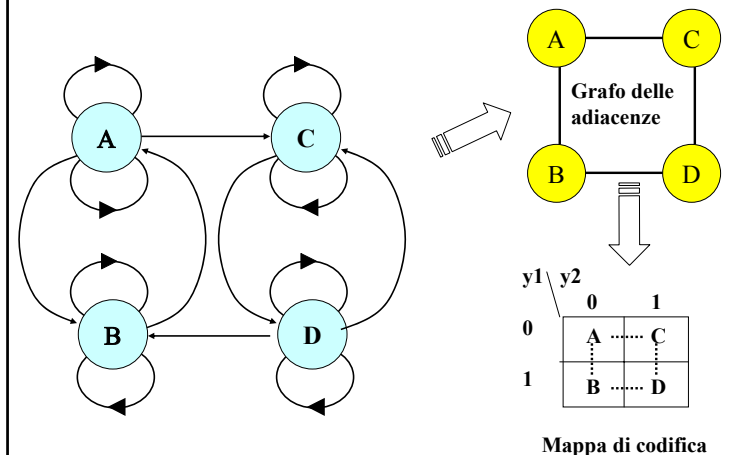
Una **corsa** è **critica** se si possono raggiungere stabilità diverse.

11 \Rightarrow 00 \Rightarrow 00
 11 \Rightarrow 01 \Rightarrow 00
 11 \Rightarrow 10 \Rightarrow 00
 corsa non critica

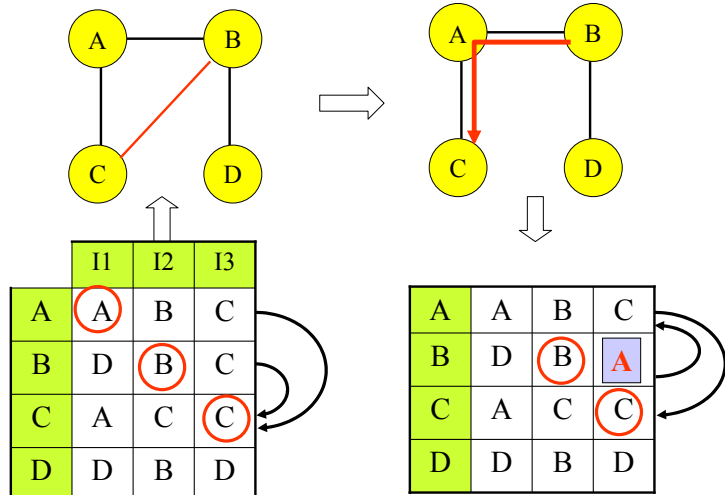
	11	12	13
00	00	01	00
01	00	01	10
11	00	11	11
10	00	11	10

01 \Rightarrow 10 \Rightarrow 10
 01 \Rightarrow 00 \Rightarrow 00
 01 \Rightarrow 11 \Rightarrow 11
 corsa critica

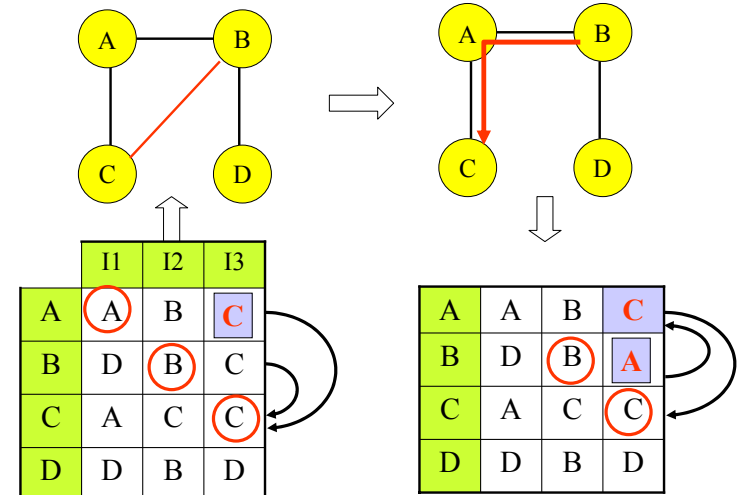
Un flip-flop privo di corse critiche



Transizioni multiple (1)



Transizioni multiple (2)

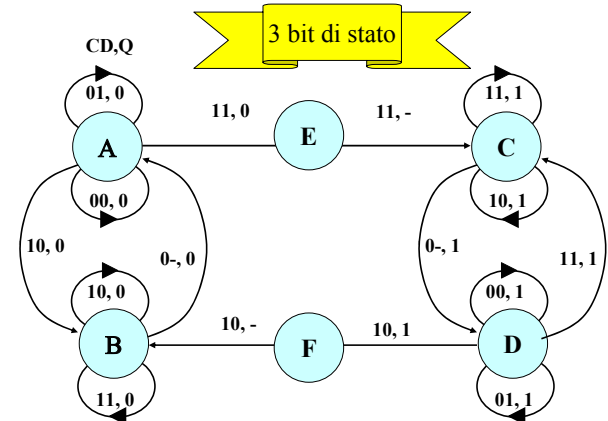


Transizioni multiple (3)

RIASSUNTO

- 1- Si adotta il n° minimo di variabili di stato
- 2- Se esiste una codifica che garantisca l'adiacenza di ogni coppia stato presente/stato futuro, allora FINE
- 3- Si controlla la colonna in cui è specificata la transizione tra due stati che non si riesce a rendere adiacenti: se esiste un altro stato che deve andare nello stato desiderato e se si può renderlo adiacente agli stati della coppia, lo si adopera come stato di transizione e poi FINE
- 4- Se nella colonna esiste uno stato con stato futuro indifferente e se si può renderlo adiacente agli stati della coppia, lo si adopera come stato di transizione e poi FINE
- 5- Si incrementa di uno il n° di variabili di stato e si torna a 4

Il flip-flop D (BellLabs, anni '60): codifica ridondante e transizioni multiple



(segue)

La tabella delle transizioni

		CD			
stato		00	01	11	10
E =	000	---,-	---,-	---,-	---,-
	001	---,-	---,-	101,-	---,-
	011	011,0	011,0	001,0	010,0
	010	011,0	011,0	010,0	010,0
C =	100	---,-	---,-	---,-	---,-
	101	111,1	111,1	101,1	101,1
	111	111,1	111,1	101,1	110,1
	110	---,-	---,-	---,-	010,-
$y_1y_2y_3$		$Y_1Y_2Y_3,Q$			

(segue)

Realizzazione
a NAND

Sintesi a NAND (1)

		CD			
$y_1y_2y_3$		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			

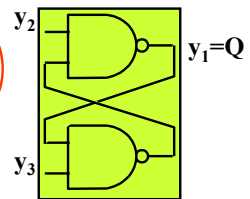
$$Q = y_1$$

Sintesi a NAND (2)

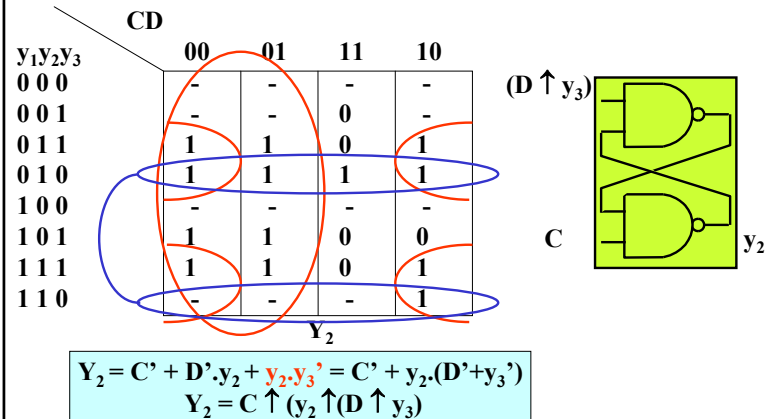
		CD			
$y_1y_2y_3$		00	01	11	10
000		-	-	-	-
001		-	-	1	-
011		0	0	0	0
010		0	0	0	0
100		-	-	-	-
101		1	1	1	1
111		1	1	1	1
110		-	-	-	0
		Y_1			

$$Y_1 = y_2' + y_3 \cdot y_1$$

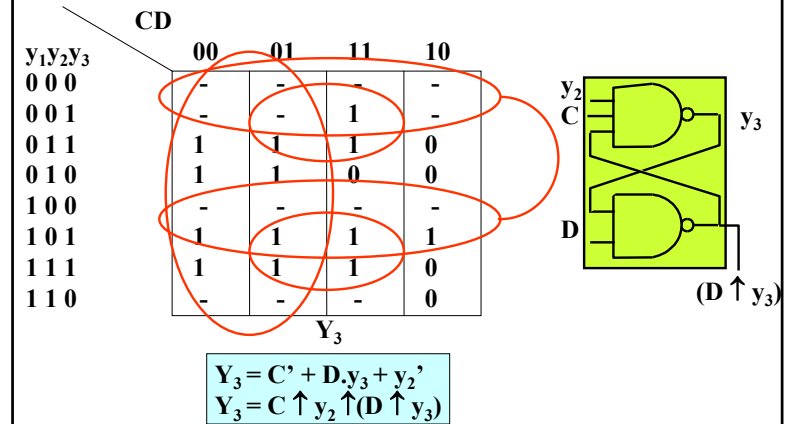
$$Y_1 = y_2 \uparrow (y_3 \uparrow y_1)$$



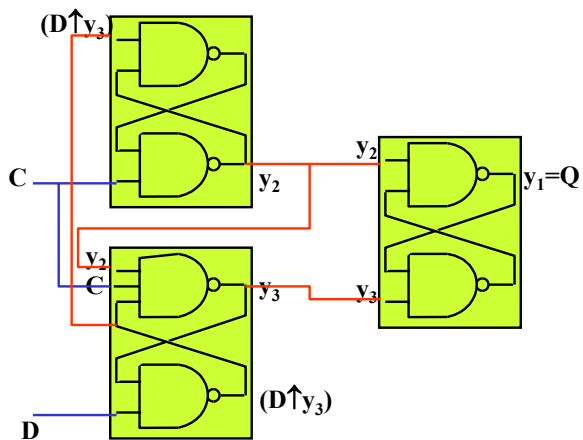
Sintesi a NAND (3)



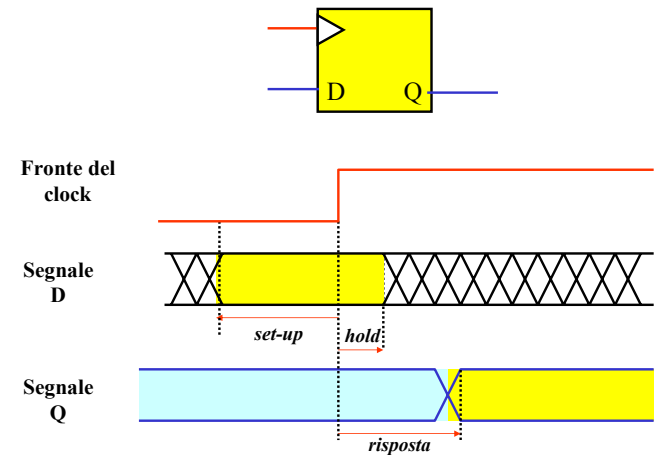
Sintesi a NAND (4)



Il flip-flop D edge triggered

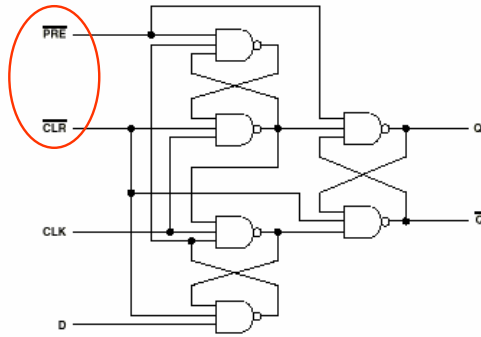


I tempi di *set-up*, di *hold* e di *risposta*



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

logic diagram (positive logic)



Q = 1 quando PRE = 1

Q = 0 quando CLR = 1

6.3 Analisi e Sintesi

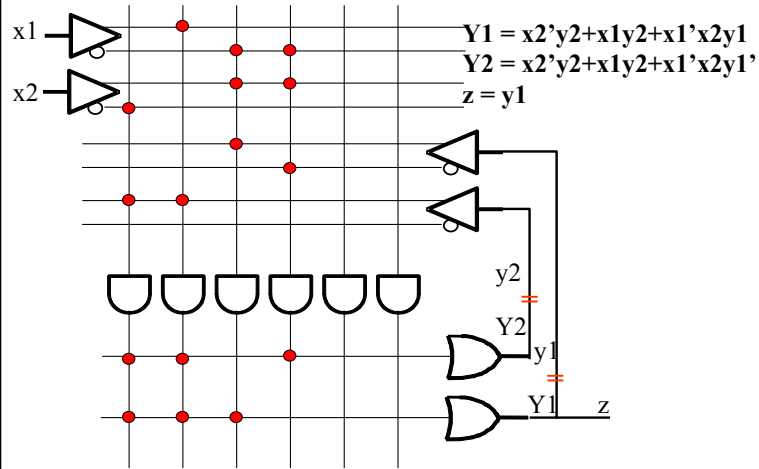
Analisi

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 (1&2)



Una PLA con due retroazioni (3)

$$Y_1 = x_2'y_2 + x_1y_2 + x_1'x_2y_1$$

$$Y_2 = x_2'y_2 + x_1y_2 + x_1'x_2y_1'$$

$$z = y_1$$

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

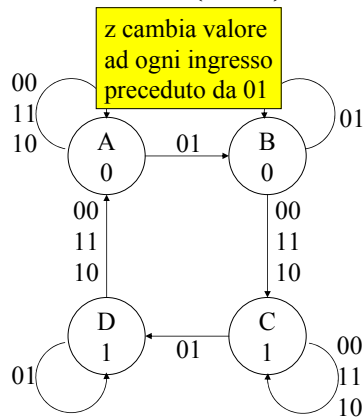
Y_1 Y_2

Una PLA con due retroazioni (4&5)

	x_1x_2			
y_1y_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

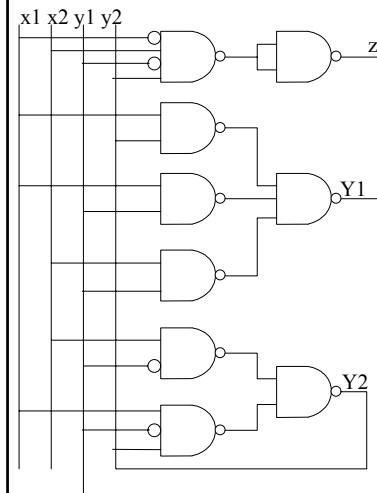
	x_1x_2			
y_1y_2	00	01	11	10
00	00,0	01,0	00,0	00,0
01	11,0	01,0	11,0	11,0
11	11,1	10,1	11,1	11,1
10	00,1	10,1	00,1	00,1

Modello di Mealy



Modello di Moore

Una rete asincrona con 2 retroazioni



$$Y1 = (x_1 \uparrow y_2) \uparrow (x_1 \uparrow y_1) \uparrow (x_2 \uparrow y_1)$$

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

$$Y2 = (x_2 \uparrow y_1') \uparrow (x_1 \uparrow y_1' \uparrow y_2)$$

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

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

$$Y1 = x1.y2 + x1.y1 + x2.y1 \quad Y2 = x2.y1' + x1.y1'.y2 \quad z = x1'.x2.y1'.y2$$

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

Y1

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

Y2

x_1x_2					
y_1y_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	

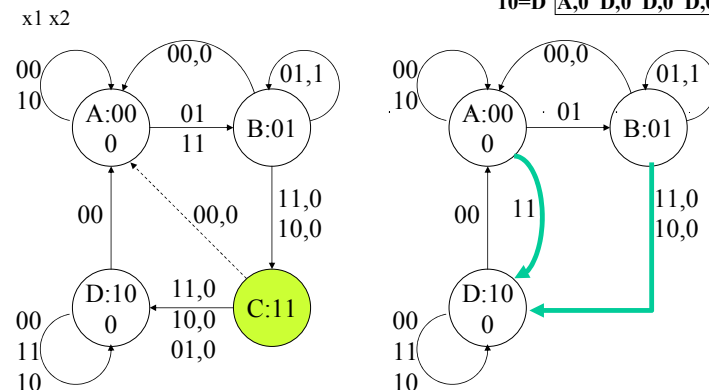
z

		$x1x2$			
$y1y2$		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

.... e tre soli
stati interni

	$x1x2$			
$y1y2$	00	01	11	10
00=A	A,0	B,0	B,0	A,0
01=B	A,0	B,1	C,0	C,0
11=C	A,0	D,0	D,0	D,0
10=D	A,0	D,0	D,0	D,0



Un circuito con troppe retroazioni

		$x1x2$			
$y1y2$		00	01	11	10
A		A,01	A,01	B,11	C11
B		A,01	A,01	B,11	C,11
C		A,00	C,10	C,10	C,10
D		A,00	A,00	B,10	C,10

Lo stato D non è mai stabile
e può essere eliminato.
Nell'incrocio C,00 si indica A

		$x1x2$			
$y1y2$		00	01	11	10
00		00,01	00,01	01,11	11,11
01		00,01	00,01	01,11	11,11
11		10,00	11,10	11,10	11,10
10		00,00	00,00	01,0	11,10

Le righe A e B sono identiche
e possono essere sostituite
da una sola riga AB

Sintesi

Il procedimento di sintesi

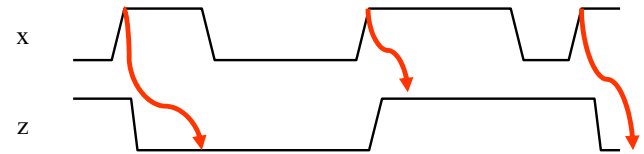
Il procedimento di sintesi di una rete sequenziale asincrona è formato da 5 passi e consente di dedurre lo schema logico dal comportamento:

- 1: individuazione del grafo degli stati,
- 2: definizione della tabella di flusso,
- 3: codifica degli stati e definizione della tabella delle transizioni,
- 4: sintesi della parte combinatoria,
- 5: schema logico.

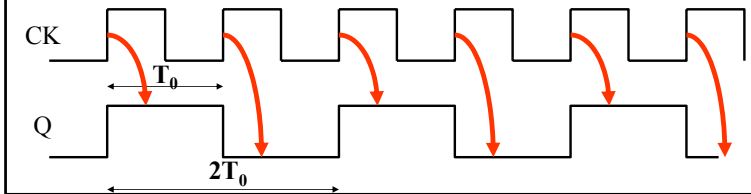
Esempio di sintesi

Comportamento: “z cambia valore ad ogni fronte di salita di x”

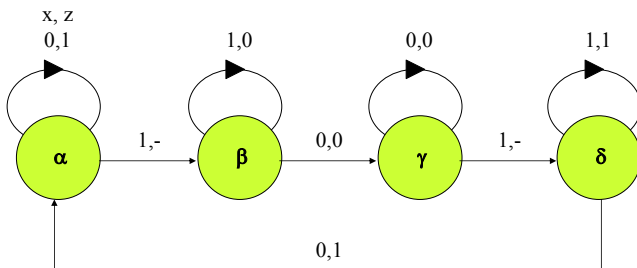
1 - Lampada da tavolo



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.

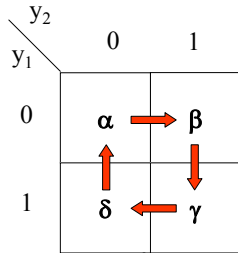
Esempio (2: tabella di flusso)

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

CONTROLLI FORMALI

1. In ogni riga ci deve essere almeno una condizione di stabilità.
2. In ogni colonna si deve raggiungere sempre una stabilità.
3. 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	0	1
$\alpha: 00$	00,1	01,-
$\beta: 01$	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)

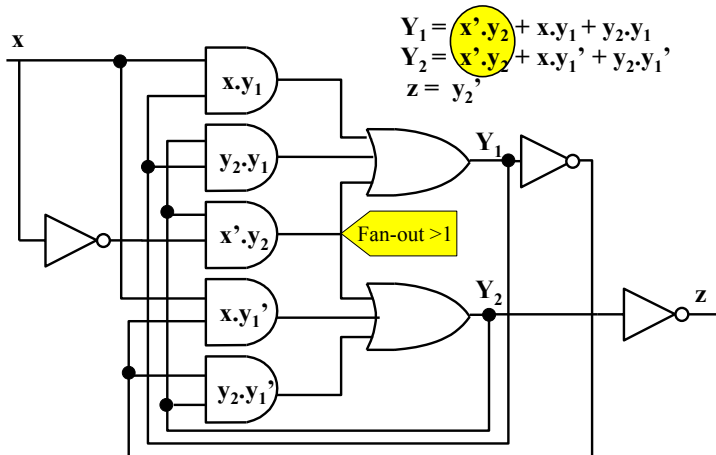
Ipotesi: si desiderano reti minime di tipo SP

$x \backslash y_1 y_2$	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)



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

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

$$z = y_2'$$

Esempio (5: schema con latch SR)

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

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

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

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

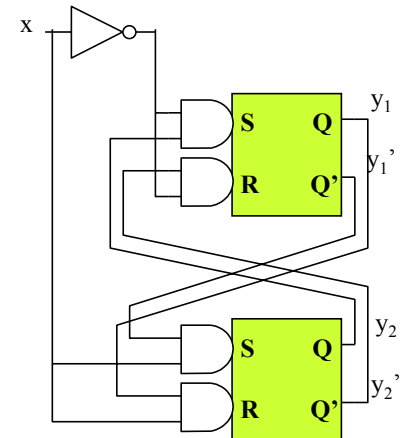
$$Y_2 = x' \cdot y_2 + x \cdot y_1' + y_2 \cdot y_1'$$

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

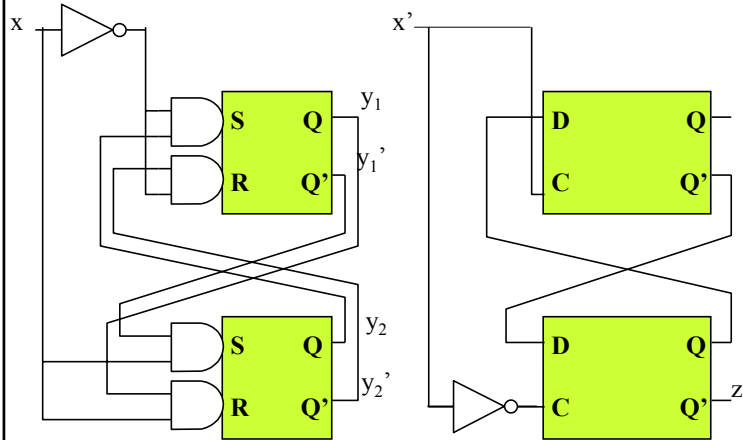
$$= x \cdot y_1' + (x \cdot y_1) \cdot y_2$$

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

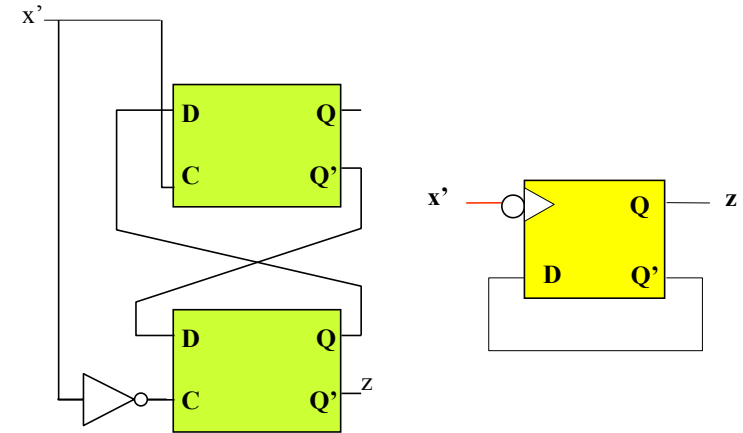
$$z = y_2'$$



Esempio (5: schema con latch CD)



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**.

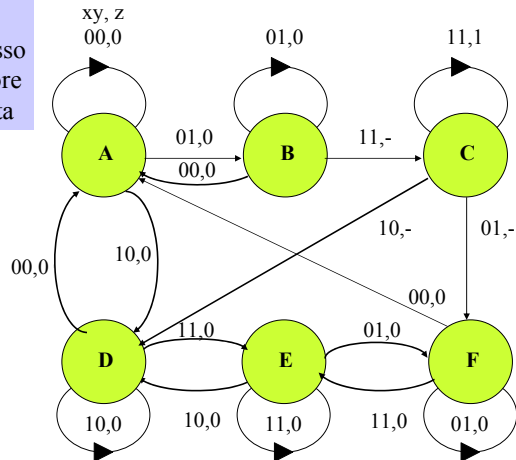
Di norma il grafo primitivo **non ha il minimo numero possibile di stati interni**

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

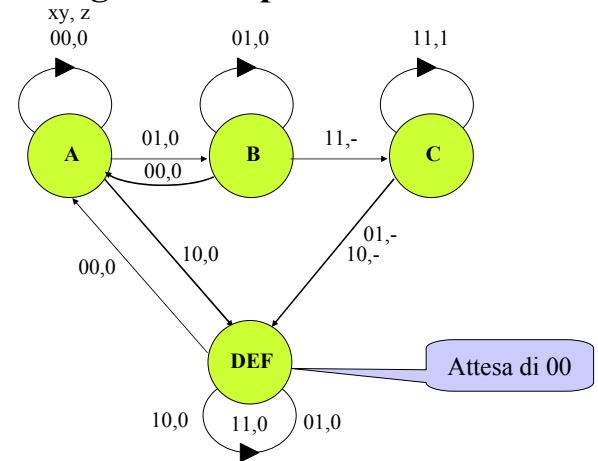


Tabelle di flusso

	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, -
D	A,0	S,0	S,0	S,0
E				
F				

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

Tabella primitiva

Tabella ridotta

Tabella minima

Due o più righe di una tabella di flusso possono essere compresse in un'unica riga se, per ogni ingresso, presentano simboli di stato futuro e di uscita **identici a meno di condizioni d'indifferenza**

Esercitazione N.12

Una RSA ha due ingressi x, y ed una uscita z . Gli ingressi non cambiano mai di valore contemporaneamente e **non presentano mai entrambi il valore 1**. L'uscita può cambiare di valore solo in corrispondenza dei **fronti di salita di x** . Il valore che z deve assumere e poi mantenere costante fino al fronte successivo è 0 se nel precedente intervallo $x=0$ **e non ha modificato il suo valore**, 1 nel caso opposto.

DOMANDA N.1 - forme d'onda

DOMANDA N.2 – grafo primitivo

DOMANDA N.3 – tabella di flusso

DOMANDA N.4 – codifica degli stati

DOMANDA N.5 – espressioni delle variabili di stato futuro

Esercitazione N.13

Una rete sequenziale asincrona è caratterizzata da due segnali di ingresso X_1 , X_2 (i quali non cambiano mai contemporaneamente) e da un segnale di uscita Z . Quando il segnale X_1 è disattivo (livello logico 0), Z deve assumere il valore 0. Quando il segnale X_1 è attivo (livello logico 1), Z deve assumere l'ultimo valore presentato dal segnale X_2 nel precedente intervallo di attivazione di X_1 .

(v. esempi di forme d'onda)

DOMANDA N.1 - grafo primitivo

DOMANDA N.2 – codifica degli stati e tabella delle transizioni

DOMANDA N.3 – espressione di una variabile di stato futuro

Esercitazione N.14

L'autoscuola GRATTA&PERDI, per addestrare meglio i suoi allievi, vi chiede di realizzare un simulatore dotato di due ingressi **f**, **c** e di una uscita **z**: **f** è il pedale della frizione (1 premuto, 0 rilasciato), **c** è la leva del cambio (1 marcia inserita, 0 folle), **z** è il comando di un segnalatore acustico (1 rumore di orrenda grattata, 0 nessun suono).

L'allievo può modificare un solo ingresso alla volta. Il rumore della grattata deve essere generato quando si inserisce una marcia con la frizione non premuta e fino a quando non venga dapprima disinserita la marcia con frizione premuta e poi rilasciata la frizione; quando si disinserisce una marcia con la frizione non premuta e fino a quando non venga dapprima inserita la marcia con frizione premuta e poi rilasciata la frizione.

DOMANDA N.1 - forme d'onda

DOMANDA N.2 – grafo degli stati

DOMANDA N.3 – tabella di flusso e delle transizioni

DOMANDA N.4 – espressione di una variabile di stato futuro