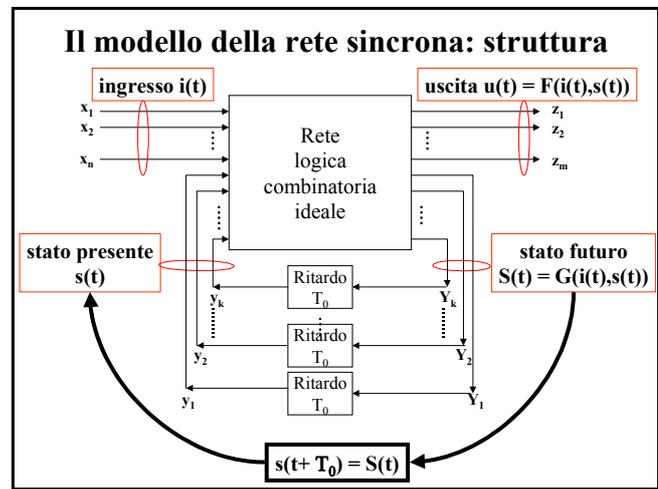


Capitolo 7: Reti sincrone

7.1 – Elaborazione sincrona

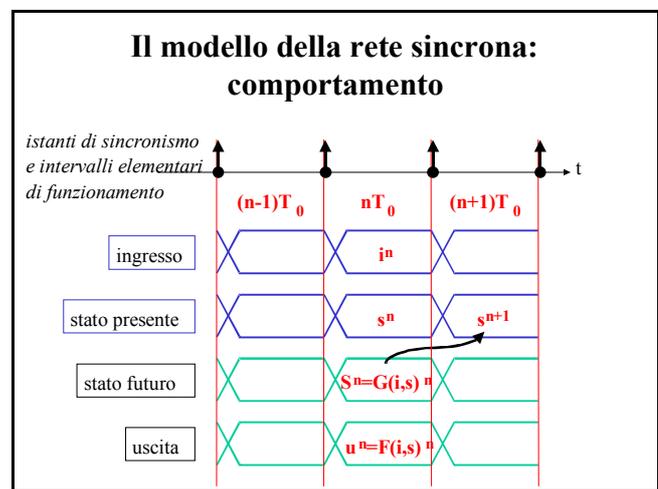
Esigenze e vincoli

Il modello della rete sincrona prevede la retroazione delle variabili di stato attraverso un componente che introduce un ritardo. Tale ritardo è pilotato da un orologio che permette l'aggiornamento dello stato presente solo in predeterminati *istanti di sincronismo*, distanziati tra loro di un tempo T_0 . Se è verificata l'ipotesi di ingressi sincroni, cioè che gli ingressi possano variare (e quindi indurre variazioni dello stato futuro e dell'uscita) solo in corrispondenza degli istanti di sincronismo, ciò equivale funzionalmente ad avere sui rami di retroazione un ritardo di durata T_0 .



Con queste retroazioni **non dirette** la macchina sequenziale sincrona appare a prima vista solo come una versione più lenta della macchina asincrona: occorre infatti aspettare sempre T_0 prima che il valore dello stato futuro venga reso disponibile sullo stato presente. In realtà ciò allarga enormemente la gamma dei comportamenti realizzabili: la cadenzata modifica dello stato interno consente infatti di mettere in conto anche il trascorrere del tempo. La misura del tempo è indispensabile ogniqualvolta l'informazione è associata non solo alla **sequenza**, ma anche alla **durata** dei simboli d'ingresso e/o d'uscita. In questo caso occorre infatti saper svolgere **azioni diverse ad istanti diversi**.

In ogni **intervallo elementare di funzionamento** compreso tra due istanti di sincronismo consecutivi la macchina fornisce un simbolo d'uscita di pari durata e si calcola un nuovo riassunto di ciò che ha fatto fino a quel momento; l'aggiornamento dello stato presente avviene all'inizio dell'intervallo successivo. Il vincolo del perfetto sincronismo dei segnali è reso meno stringente grazie all'uso di **flip-flop** per la realizzazione del ritardo. Il suo corretto impiego richiede infatti **soltanto** (v. pag. 113) che il segnale da campionare abbia e mantenga il valore



desiderato un po' di tempo prima (il **set-up time**, usualmente indicato con il simbolo τ_{SU}) ed un po' di tempo dopo (l' **hold time**, indicato con τ_H) il verificarsi di un fronte di salita del clock.

Prendendo come riferimento l'istante t in cui si verifica il fronte del clock ed andando a ritroso nel tempo occorre dunque che il blocco combinatorio preposto al calcolo dello stato futuro abbia sicuramente completato i suoi calcoli all'istante

$$t - \tau_{SU}$$

Indichiamo ora con τ_{RC} il ritardo, **nel caso peggiore**, di questa rete combinatoria. I suoi ingressi, cioè le configurazioni associate al simbolo che proviene dall'esterno ed al simbolo di stato interno, devono avere valori a regime all'istante

$$t - \tau_{SU} - \tau_{RC}$$

I segnali d'ingresso devono dunque aver completato il loro transitorio prima di questo istante. La stessa condizione deve valere anche per i segnali di stato presente, cioè per le uscite dei flip-flop. Sappiamo però che il flip-flop impiega un certo tempo a partire dal fronte del clock (il **response time**, usualmente indicato con τ_R) per completare l'esecuzione del comando di scrittura: tale comando deve quindi essere stato dato prima dell'istante

$$t - \tau_{SU} - \tau_{RC} - \tau_R.$$

Quanto abbiamo detto è illustrato in figura ed individua la massima frequenza attribuibile al clock di una rete sequenziale sincrone. Il periodo dell'oscillatore deve dunque essere un po' **più grande** della durata del transitorio sull'anello di retroazione:

$$T_0 > \tau_R + \tau_{RC} + \tau_{SU}$$

Il rispetto del vincolo che il segnale in ingresso al flip-flop mantenga il valore campionato per almeno τ_H è garantito dal fatto che nelle realizzazioni integrate si ha sempre:

$$\tau_H < \tau_R.$$

Quando serve, si può dunque, collegare direttamente l'uscita di un flip-flop all'ingresso di un altro ($\tau_{RC} = 0$) senza temere che il valore campionato scompaia prima del tempo.

Una volta rispettata la condizione sulla frequenza di funzionamento della rete sincrone (cosa che da ora in poi daremo per scontata), il progetto del blocco combinatorio che realizza la funzione G diventa molto più semplice di quello richiesto nelle reti asincrone:

- non occorre preoccuparsi dei **fenomeni di alea**, statica e dinamica, dato che possono verificarsi solo all'inizio dell'intervallo e che saranno quindi sicuramente terminati all'istante di campionamento;
- non occorre preoccuparsi dell'**adiacenza** delle configurazioni consecutive d'ingresso e di stato: le configurazioni spurie causate da segnali che non cambiano proprio allo stesso istante si verificano solo all'inizio dell'intervallo e non verranno quindi mai campionate.

Le conseguenze sono notevoli:

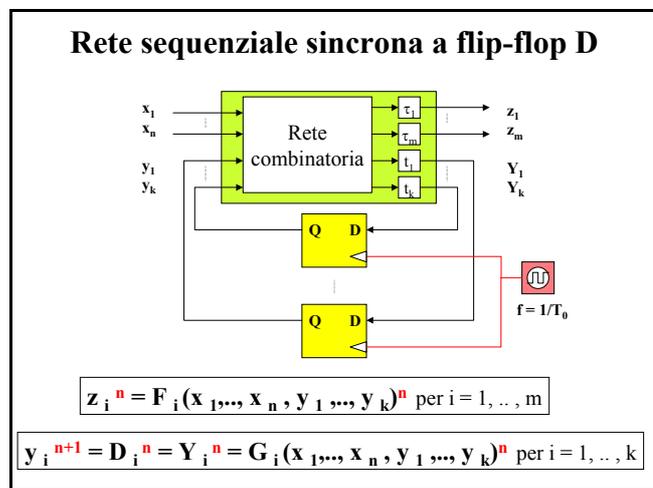
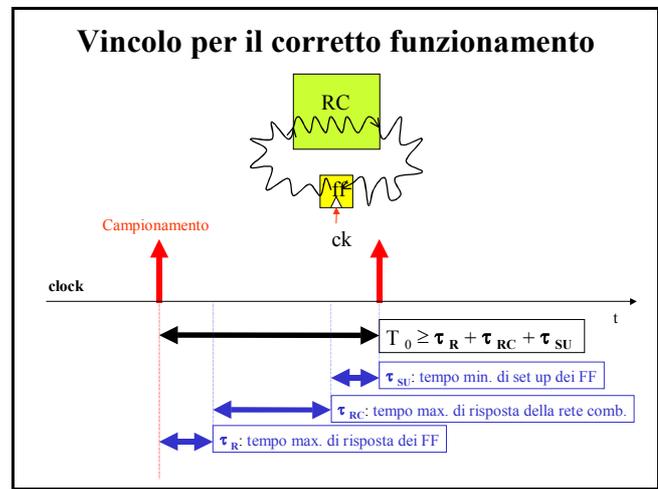
1. è possibile **impiegare espressioni minime** nella realizzazione delle funzioni di stato e d'uscita;
2. è possibile **codificare arbitrariamente** i simboli d'ingresso e d'uscita.

Tipi di flip-flop

Per chiudere le retroazioni di un circuito sincrone non c'è solo il **flip-flop D** che abbiamo studiato finora. Nell'ambito dei circuiti elettronici integrati sono stati infatti realizzati anche il **flip-flop JK** ed il **flip-flop T**. A parità di tecnologia, tutti hanno pressoché la stessa durata del transitorio.

Le modalità d'impiego del flip-flop D discendono direttamente dal suo comportamento come ritardo T_0 . Le variabili di stato futuro, calcolate da un'apposita rete combinatoria, sono connesse agli ingressi **D** di altrettanti flip-flop; le variabili di stato presente sono disponibili sulle loro uscite **Q** e riproducono con un intervallo di ritardo la codifica dello stato futuro:

$$Q_i^{n+1} = D_i^n = G_i(x_1, \dots, x_n, Q_1, \dots, Q_i, \dots, Q_k)^n$$



Il flip-flop JK ha due ingressi (detti appunto **J** e **K**), tramite i quali è possibile imporgli quattro differenti comportamenti:

1. **J=0, K=0** (comando di **hold**) determina $Q^{n+1} = Q^n$
2. **J=1, K=0** (comando di **set**) determina $Q^{n+1} = 1$
3. **J=0, K=1** (comando di **reset**) determina $Q^{n+1} = 0$
4. **J=1, K=1** (comando di **toggle**) determina $Q^{n+1} = Q'^n$

In figura le quattro regole sono state riportate su una mappa, da cui è stata poi dedotta la seguente **equazione caratteristica**:

$$Q^{n+1} = (J \cdot Q' + K' \cdot Q)^n$$

Anche il flip-flop JK è in realtà una rete sequenziale asincrona azionata dai fronti di un clock.

L'equazione caratteristica ne fornisce un'immagine astratta (quella di una rete sincrona) ed evidenzia il ritardo T_0 con cui ubbidisce ai suoi comandi.

La formula è d'aiuto nel procedimento di sintesi. Supponiamo ad esempio di non disporre del flip-flop JK e di volerne ottenere il comportamento a partire da un flip-flop D.

Tutto quello che serve è una rete combinatoria con ingressi J, K, Q e con uscita

$$D = J \cdot Q' + K' \cdot Q$$

Viceversa, avendo a disposizione un flip-flop JK, è possibile costruirsi un flip-flop D connettendo l'ingresso J al segnale D e l'ingresso K al suo complemento D':

$$Q^{n+1} = (D \cdot Q' + (D') \cdot Q)^n = (D \cdot Q' + D \cdot Q)^n = D^n$$

Il flip-flop T ha il solo ingresso **T** e quindi due sole modalità di funzionamento:

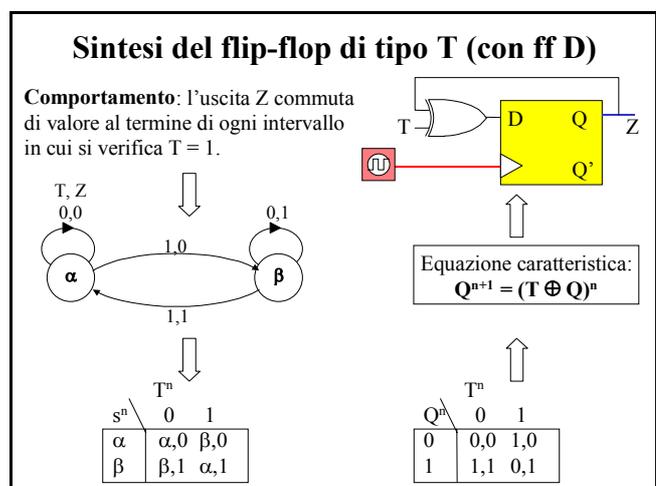
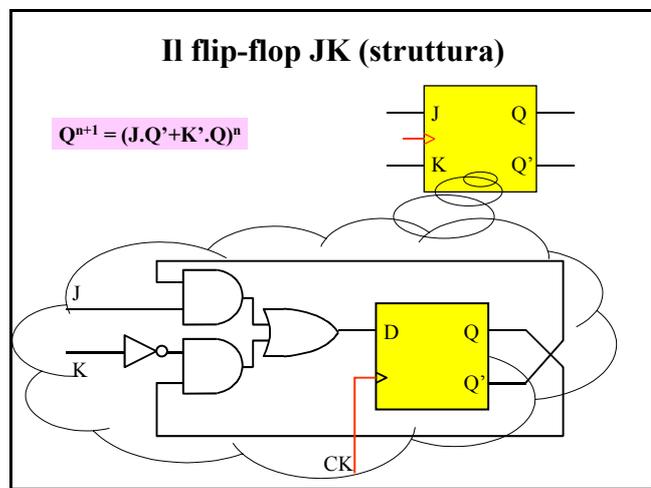
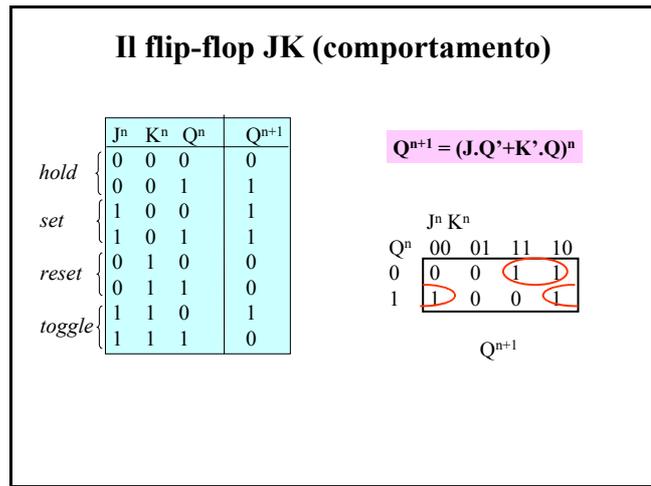
1. **T=0** (comando di **hold**) determina $Q^{n+1} = Q^n$
2. **T=1** (comando di **toggle**) determina $Q^{n+1} = Q'^n$

In figura è mostrato come questo comportamento possa essere ottenuto con un flip-flop D. Il grafo di questa macchina sincrona elementare richiede due stati: la costanza dell'uscita per un intervallo consente l'uso del modello di Moore, come si può evincere dal fatto che i rami uscenti da ogni nodo hanno un uguale valore d'uscita.

Dalla tabella delle transizioni si ottiene l'**equazione caratteristica** di questa memoria:

$$Q^{n+1} = (T \oplus Q)^n$$

Nella realizzazione occorre dunque un gate EX-OR retroazionato da un flip-flop D.



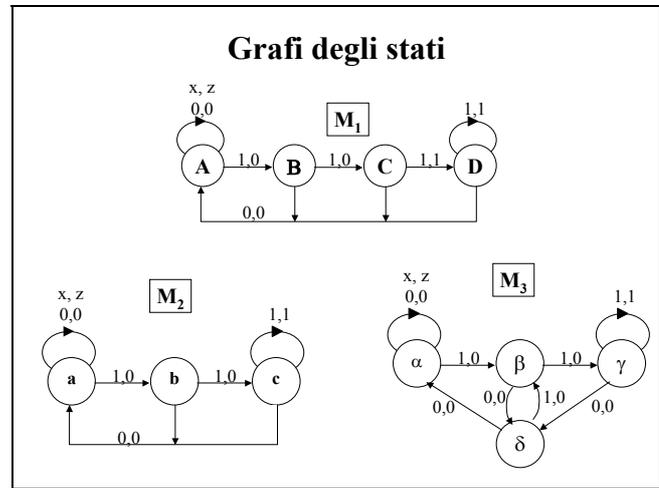
- **Passo 2: individuazione del grafo degli stati** - Si procede alla costruzione di **uno dei diversi possibili grafi** in grado di descrivere il comportamento desiderato: usuale è l'impiego del modello di Mealy, per specificare puntualmente il simbolo d'uscita che la macchina deve generare in ogni transizione e quindi in ogni intervallo elementare del suo funzionamento.

ESEMPIO - Nel grafo M1 indicato in figura il riconoscimento dei tre "uni" consecutivi dopo uno "zero" è affidato alle transizioni da **A** a **B**, da **B** a **C** e da **C** a **D**.

La stabilità in **D** (si noti che nelle reti sincrone, a differenza di quelle asincrone, non è richiesto che ogni stato sia stabile per l'ingresso che lo ha determinato) consente di considerare ogni ulteriore "uno" come ultimo di una terna.

Le transizioni da **B,C,D** ad **A** e la stabilità in **A** consentono infine di scartare tutte le sequenze in cui appare almeno uno "zero".

Il riconoscimento di tre "uni" consecutivi dell'ingresso **x** può essere fatto altrettanto bene con i grafi M2, M3 indicati in figura. M1, M2 e M3 sono dunque tre possibili descrizioni grafiche di uno stesso comportamento: in casi di questo genere, si dice che gli automi corrispondenti sono **equivalenti**.

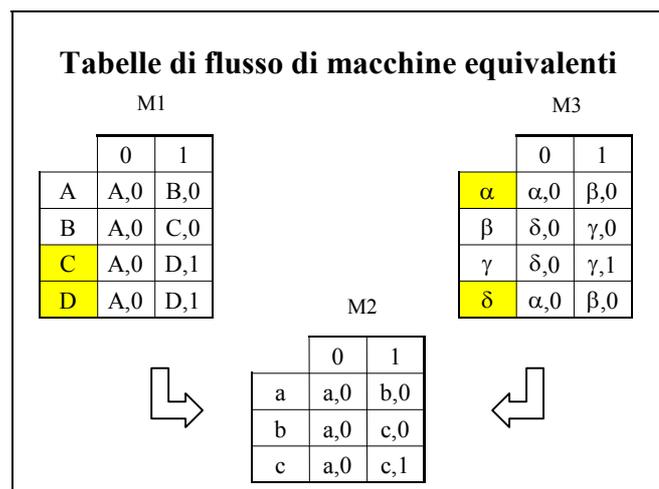


- **Passo 3: definizione della tabella di flusso** - Il grafo individuato nel passo 2 viene trasformato in tabella di flusso. Per realizzare un circuito con il minimo numero possibile di retroazioni, si procede poi alla ricerca dell'**automa minimo**: a tal fine si individua l'eventuale presenza di stati **equivalenti** (sono detti equivalenti stati che, assunti come iniziali, generano sequenze d'uscita identiche per ogni possibile sequenza d'ingresso) e si costruisce una tabella di flusso che contiene un'unica riga al posto di tutte quelle che sono state individuate come equivalenti. Una semplice regola di equivalenza applicabile alle tabelle di flusso delle reti sincrone è: **“due righe della tabella di flusso rappresentano stati equivalenti se, per ogni ingresso, i simboli d'uscita sono identici ed i simboli di stato futuro sono o identici, o gli stessi della coppia in esame, o quelli di stati per cui è già stata dimostrata l'equivalenza”**.

ESEMPIO - Le tabelle di flusso dei grafi M1, M3 e M2 sono indicate in figura. La tabella M1 ha due righe identiche: C e D sono dunque stati equivalenti. Analogamente sono equivalenti gli stati α e δ della tabella M3. Nella tabella M2 non sono presenti stati equivalenti: tale tabella descrive dunque l'**automa minimo**.

Se uno non avesse individuato l'automa minimo nel passo 2, può sempre farlo nel passo 3. M2 può infatti essere ottenuta da M1 eliminando la riga D (oppure la C) e sostituendo poi il simbolo **a** al simbolo **A**, il simbolo **b** al simbolo **B** ed il simbolo **c** ai simboli **C,D**. Se il punto di partenza è invece M3, M2 si ottiene eliminando la riga δ e sostituendo poi il simbolo **a** ai simboli α, δ , il simbolo **b** al simbolo β ed il simbolo **c** al simbolo γ .

In questo caso l'impiego di M2 non produce vantaggio: come per M1 e per M3 occorrono infatti 2 bit di codifica e quindi 2 flip-flop.



- **Passo 4: codifica degli stati e definizione della tabella delle transizioni** - La scelta di un periodo di clock superiore alla somma dei ritardi presenti sugli anelli di retroazione elimina a priori il pericolo delle **corse critiche** e rende quindi **arbitraria** la codifica degli stati. Una volta stabilito il codice, di norma **non ridondante**, la tabella delle transizioni si ottiene sostituendo i simboli di stato con le corrispondenti configurazioni.

ESEMPIO – In figura sono riportate le codifiche a 2 bit degli stati di M1, M3, M2 e le conseguenti tabelle delle transizioni

Il codice per M1 è stato scelto a caso: incidentalmente è capitato che tutte le transizioni in colonna $x=1$ comportano la variazione di un solo bit, ma ciò non era assolutamente necessario.

Il codice per M3 non è invece stato scelto a caso: le sue proprietà verranno discusse più avanti.

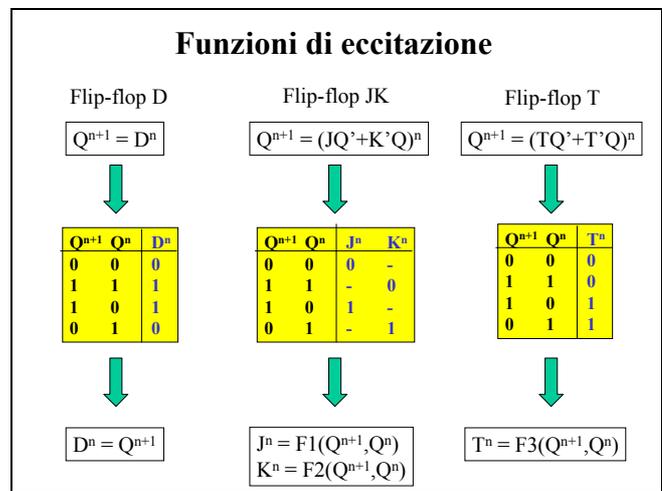
La tabella di M2 merita un commento: l'automata ha solo tre stati, ma i bit di codifica devono essere due. Il circuito avrà dunque due retroazioni e potrà eventualmente assumere come stato interno anche la configurazione 01, non utilizzata nel codice. E' dunque opportuno che questa riga sia resa instabile per ogni ingresso: a questo punto del progetto è però di solito riempita con condizioni d'indifferenza.

Tabelle delle transizioni di M1, M3, M2

		M1		M3		M2			
$y_1^n y_2^n$		0	1	$y_1^n y_2^n$	0	1	$y_1^n y_2^n$	0	1
A:00		00,0	10,0	α :00	00,0	10,0	a:00	00,0	10,0
B:10		00,0	11,0	β :10	01,0	11,0	b:10	00,0	11,0
C:11		00,0	01,1	γ :11	01,0	11,1	c:11	00,0	01,1
D:01		00,0	01,1	δ :01	00,0	10,0	01	--,-	--,-
		$y_1^{n+1} y_2^{n+1} z^n$		$y_1^{n+1} y_2^{n+1} z^n$		$y_1^{n+1} y_2^{n+1} z^n$			

➤ **Passo 5: scelta dei flip-flop e sintesi delle funzioni di eccitazione** - Ogni variabile di stato presente della macchina è realizzata dall'uscita Q di un flip-flop; la scelta del tipo è **arbitraria**.

Il progetto dei segnali da inviare in ingresso al flip-flop (D, JK, T) dipende dal tipo prescelto. Ciascuno di essi deve infatti realizzare una **funzione di eccitazione** $F(Q^n, Q^{n+1})$, atta ad imporre a Q, ad ogni istante di sincronismo e nel rispetto della **tabella delle transizioni**, o la variazione da 0 a 1, o la variazione da 1 a 0, o il mantenimento di 1, o il mantenimento di 0. I valori necessari nei quattro casi discendono dall'equazione caratteristica del flip-flop e sono indicati in figura.



Con queste tabelle, come vedremo tra poco, è agevole dedurre le funzioni di eccitazione dalla **tabella delle transizioni** e realizzarle poi con **espressioni minime**. In generale la complessità del circuito dipende dal **grafo**, dalla **codifica degli stati** e dal **tipo di flip-flop**.

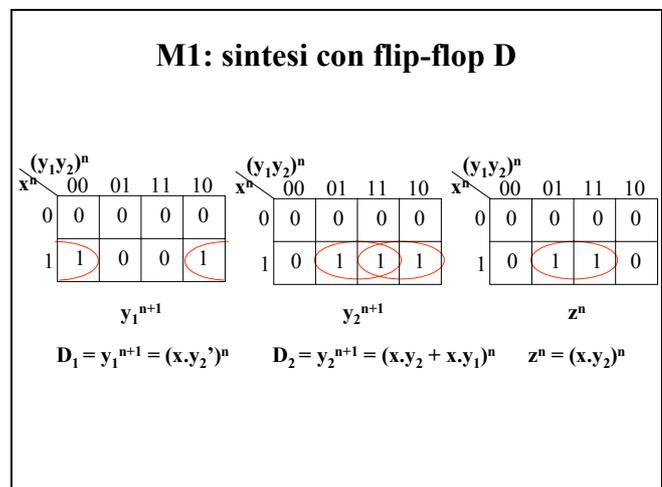
Sintesi con flip-flop D

La funzione di eccitazione del **flip-flop D** coincide con la funzione che calcola la variabile di stato futuro. Nel generico intervallo n deve quindi essere: $D^n = Q^{n+1}$.

ESEMPIO – Supponiamo di dover realizzare con flip-flop D, AND, OR e NOT la precedente tabella delle transizioni M1.

In figura sono mostrate le mappe delle due variabili di stato futuro e della variabile d'uscita. Su ciascuna mappa sono indicati i raggruppamenti necessari per una copertura minima degli "uni". Al di sotto appaiono le corrispondenti espressioni minime di tipo SP.

La realizzazione del circuito richiede dunque due flip-flop D, tre AND a due ingressi (*perché?*), un OR a due ingressi, nessun NOT (*perché?*).



ESEMPIO – Prendiamo ora in considerazione la sintesi della precedente tabella delle transizioni M3.

Dalle mappe di figura discende che la realizzazione delle due funzioni di eccitazione non richiede alcun gate. Questo risultato notevole dipende dalla particolare codifica degli stati precedentemente adottata: esiste un procedimento formale per individuarla, ma l'argomento esula dai limiti di questo corso.

In figura è mostrato lo schema logico ed i passaggi che consentono di verificare la correttezza del suo comportamento.

La struttura ottenuta è detta **registro a scorrimento a 2 bit** ed ha la proprietà notevole che i bit di stato memorizzano successivi valori del bit d'ingresso.

Tutte le **macchine con memoria finita** (v. cap. 3, pag. 36) possono essere realizzate da un registro a scorrimento che memorizza la sequenza d'ingresso e da una rete combinatoria che l'elabora.

M3: sintesi con flip-flop D

Ipotesi: reti minime di tipo SP

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

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

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

$D_1^n = y_1^{n+1} = x^n$ $D_2^n = y_2^{n+1} = y_1^n$ $z^n = x^n \cdot y_1^n \cdot y_2^n$

Verifica del comportamento:
 $y_1^{n+1} = x^n$
 $y_2^{n+1} = y_1^n = x^{n-1}$
 $z^n = (x \cdot y_1 \cdot y_2)^n = x^n \cdot x^{n-1} \cdot x^{n-2}$

ESEMPIO – La sintesi a AND, OR, NOT e flip-flop D della tabella delle transizioni M2 è riportata nella figura a lato. Si noti che, coerentemente con quanto fatto prima, lo stato 01 è stato considerato **impossibile**: le variabili di stato futuro hanno infatti in corrispondenza valore indifferente.

Una volta fatta la copertura, tali valori indifferenti sono però stati fissati a 0: si noti infatti che le due celle non sono state incluse in alcun RR(1). Nel circuito lo stato 01 ha dunque sempre 00 come stato futuro, cioè è **instabile per ogni ingresso**, come si era auspicato quando era stata definita la tabella delle transizioni.

La proprietà molto importante secondo la quale le configurazioni di stato non utilizzate dall'automa vengono abbandonate per qualsiasi configurazione di ingresso, riportando la macchina in un tempo finito ad una condizione di funzionamento prevista dal modello, è detta di **autoinizializzazione**. Può essere ottenuta a priori imponendo valori appropriati dello stato futuro, al posto delle indifferenze che appaiono in corrispondenza di stati presenti impossibili, a discapito della flessibilità offerta dalle indifferenze in fase di copertura, oppure come nell'esempio precedente può essere verificata a posteriori, dopo aver sfruttato liberamente le indifferenze per realizzare la copertura minima.

M2: sintesi con flip-flop D

Ipotesi: si cercano reti minime di tipo SP

$(y_1 y_2)^n$	
x^n	00 01 11 10
0	0 - 0 0
1	1 - 0 1

$(y_1 y_2)^n$	
x^n	00 01 11 10
0	0 - 0 0
1	0 - 1 1

$(y_1 y_2)^n$	
x^n	00 01 11 10
0	0 - 0 0
1	0 - 1 0

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

$D_1 = y_1^{n+1} = (x \cdot y_2)^n$ $D_2 = y_2^{n+1} = (x \cdot y_1)^n$ $z^n = (x \cdot y_2)^n$

Sintesi con flip-flop JK

Per individuare le funzioni J e K è utile dapprima “ingrossare” i valori di y^{n+1} che differiscono dai corrispondenti valori di y^n e generare poi due mappe in cui i quattro differenti simboli che così si ottengono sono sostituiti dalle appropriate configurazioni di comando.

- ⌈ $\Rightarrow J = 1, K = -$, 1 $\Rightarrow J = -, K = 0$,
- ⓪ $\Rightarrow J = -, K = 1$, 0 $\Rightarrow J = 0, K = -$.

L'uso del flip-flop JK al posto del D determina spesso espressioni più semplici.

ESEMPIO – In figura, a sinistra, sono indicate le due funzioni di stato futuro che discendono dalla tabella delle transizioni di M1.

Su ciascuna di queste mappa appaiono ingrossati i valori futuri diversi dai valori presenti.

Nella parte destra della figura sono mostrate le mappe delle funzioni di eccitazione, quali discendono dalle quattro sostituzioni dei simboli “promemoria”.

M1: sintesi con flip-flop JK

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

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

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

y_1^{n+1} $J_1^n = x \cdot y_2'$ $K_1^n = x' + y_2$

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

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

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

y_2^{n+1} $J_2^n = x \cdot y_1$ $K_2^n = x'$

CASO DI STUDIO – La rete sequenziale sincrona di figura deve continuamente contare, modulo 2, gli intervalli di tempo in cui si verifica $x = 0$.

Il risultato del conteggio appare su z e viene aggiornato solo al termine di ogni intervallo in cui non si è contato ($x = 1$).

I valori 0 e 1 di z indicano rispettivamente che la rete ha visto, in tutta la sua storia passata, un numero “pari” ed un numero “dispari” di intervalli con $x=0$.

Passo 1- Le forme d’onda di figura evidenziano un caso. La rete, inizialmente con $z = 0$, vede per un intervallo $x = 0$; l’uscita, come richiesto, è portata a 1 con due intervalli di ritardo. Successivamente arrivano due intervalli con $x = 0$; il risultato del conteggio complessivo è di nuovo 1 (dispari) e tale valore viene quindi confermato su z .

Passo 2 - La rete può ricevere sequenze comunque lunghe di “uni” durante le quali, a seconda di cosa è successo prima, deve fornire o uscita 0, o uscita 1.

Il tracciamento del grafo può dunque utilmente partire dagli stati **a** e **d**, entrambi stabili per $x = 1$.

All’arrivo del primo zero occorre prenderne atto, transitando dallo stato **a** allo stato **b** in un caso, dallo stato **d** allo stato **e** nell’altro.

Il conteggio modulo due di eventuali zeri successivi è effettuato rispettivamente dalle transizioni **b** → **c** → **b** ...

e → **f** → **e**

Al primo uno le transizioni **c** → **a** e **f** → **d** consentono alla rete di confermare il valore di z , le transizioni **b** → **d** e **e** → **a**, di modificarlo.

Passo 3 - Per sapere se è stato impiegato il minimo numero possibile di stati conviene fare riferimento alla tabella di flusso.

La tabella, indicata in figura in alto a sinistra, consente infatti di individuare a colpo d’occhio l’eguaglianza delle coppie di righe {**a,c**} e {**d,f**}.

Denominando **a** la prima coppia e **d** la seconda, si ottiene la tabella non più riducibile indicata a lato.

Al di sotto è indicato il grafo che corrisponde a tale tabella minima.

Passo 4- Qualsiasi codice a 2 bit va bene.

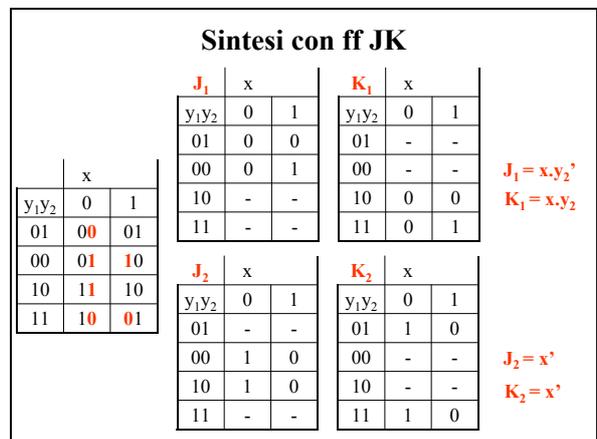
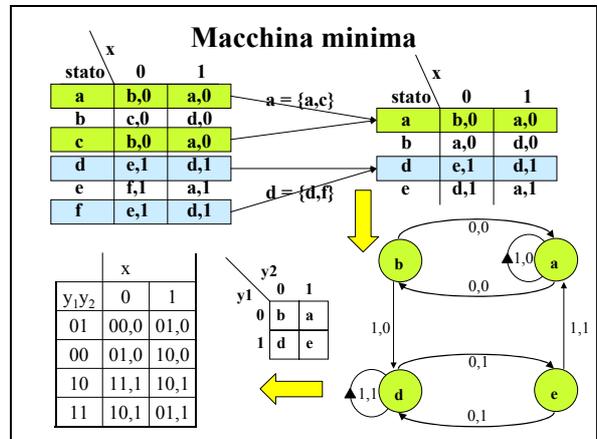
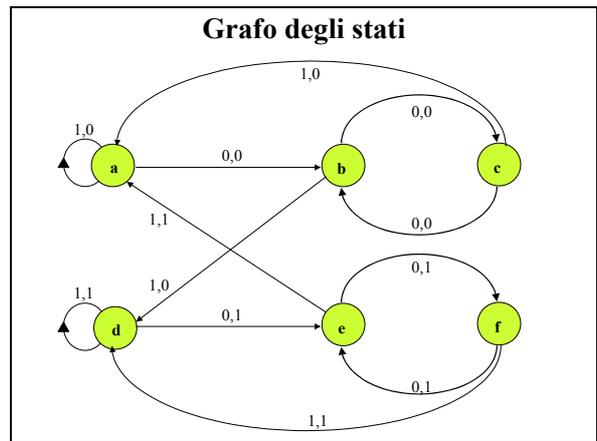
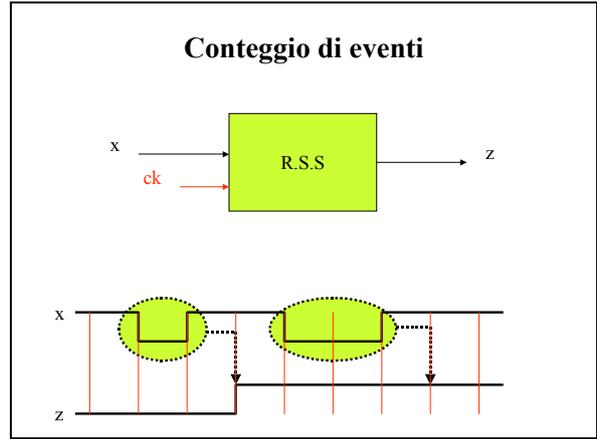
In figura è evidenziata una mappa di codifica in cui si è privilegiato il criterio di far variare un solo bit alla volta. La tabella delle transizioni che ne discende è indicata a lato.

Passo 5 – Supponiamo di volere un circuito con flip-flop JK sulle retroazioni e AND, OR, NOT nella parte combinatoria.

In figura, da sinistra verso destra, sono indicati i risultati delle tre azioni da fare in sequenza:

- 5.1 l’ingrossamento degli “uni” e degli “zeri” sulla tabella delle transizioni,
- 5.2 l’individuazione delle due funzioni di eccitazione di ciascun flip-flop,
- 5.3 la deduzione delle loro espressioni minime SP.

L’uscita deve valere 1 quando la macchina si trova o nello stato **d**, o nello stato **e**. Non occorre dunque l’ausilio di una mappa per individuare l’espressione minima: **$z = y_1$** .



Sintesi con flip-flop T

Anche per individuare la funzioni di eccitazione del **flip-flop T**, è utile dapprima “ingrossare” i valori di y^{n+1} che differiscono dai corrispondenti valori di y^n , esattamente come si fa nella sintesi con flip-flop JK.

Le sostituzioni da operare successivamente sono però diverse:

- 1 ⇒ T = 1,
- 0 ⇒ T = 1,
- 1 ⇒ T = 0,
- 0 ⇒ T = 0.

ESEMPIO – In figura è trattato il caso della macchina M1 precedentemente definita.

Una volta che la tabella delle transizioni è stata scomposta nelle tabelle y_1^{n+1} e y_2^{n+1} , l’ingrossamento dei simboli individua quattro esigenze di variazione di valore per la prima variabile di stato e tre per la seconda; altrettanti sono dunque gli “uni” presenti nelle mappe delle due funzioni di eccitazione.

La realizzazione di M1 con flip-flop T è in questo caso più complessa di quelle ottenute in precedenza. Ovviamente non è sempre così.

M1: sintesi con flip-flop T

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

y_1^{n+1}

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

$T_1^n = x \cdot y_2' \cdot y_1' + y_1 \cdot y_2 + x' \cdot y_1$

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

y_2^{n+1}

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

$T_2^n = x \cdot y_2' \cdot y_1 + x' \cdot y_2$

Esercitazione N.16

Progettare il semaforo per una direttrice di marcia con AND, OR, NOT e flip-flop D.

Il semaforo: sintesi con ff D e reti minime SP

$(y_1 y_2 y_3)^n$	$(y_1 y_2 y_3)^{n+1}$	$(z_1 z_2 z_3)^n$
0 0 0	0 0 1	1 0 0
0 0 1	0 1 0	1 0 0
0 1 0	0 1 1	1 0 0
0 1 1	1 0 0	0 1 0
1 0 0	1 0 1	0 0 1
1 0 1	1 1 0	0 0 1
1 1 0	0 0 0	0 0 1

$y_2^n y_3^n$		00	01	11	10
y_1^n	0				
	1				

$y_1^{n+1} =$

$y_2^n y_3^n$		00	01	11	10
y_1^n	0				
	1				

$z_1^n =$

$y_2^n y_3^n$		00	01	11	10
y_1^n	0				
	1				

$y_2^{n+1} =$

$y_2^n y_3^n$		00	01	11	10
y_1^n	0				
	1				

$z_2^n =$

$y_2^n y_3^n$		00	01	11	10
y_1^n	0				
	1				

$y_3^{n+1} =$

$y_2^n y_3^n$		00	01	11	10
y_1^n	0				
	1				

$z_3^n =$

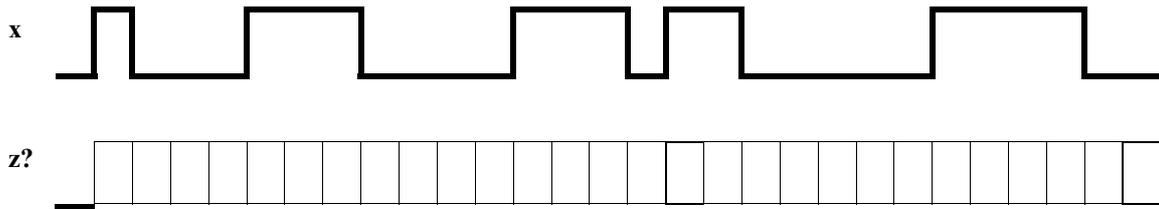
137

Esercitazione N.17

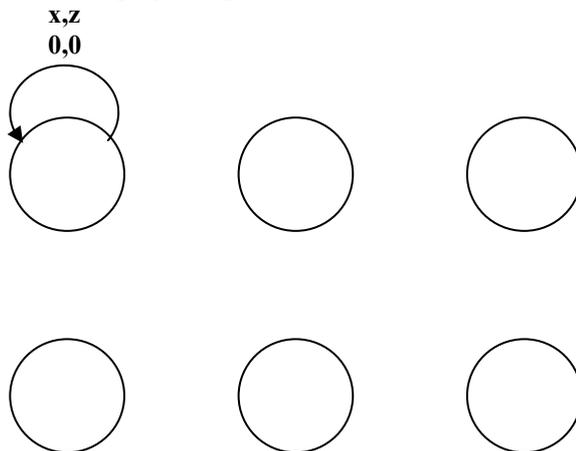
Una rete sequenziale sincrona ha il compito di riprodurre sulla sua uscita z , con un ritardo di due intervalli di clock, il valore presente sul suo ingresso x , a condizione però che tale valore perduri per più di due intervalli.

Se il valore di x è presente solo per uno o per due intervalli, l'uscita z deve ignorarlo e mantenere il valore che aveva prima della variazione di x .

DOMANDA N.1 - Indicare la forma d'onda del segnale d'uscita z in corrispondenza della forma d'onda del segnale d'ingresso x mostrata in figura.



DOMANDA N.2 – Completare il grafo degli stati .



DOMANDA N.3 - Riempire la tabella di flusso, individuare una codifica degli stati e riempire la tabella delle transizioni.

s^n	x^n	
	0	1
A		
B		
C		
D		
E		
F		

s^{n+1}, z^n

s^n	$y_1^n y_2^n y_3^n$	x^n	
		0	1
	0 0 0		
	0 0 1		
	0 1 1		
	0 1 0		
	1 0 0		
	1 0 1		
	1 1 1		
	1 1 0		

$y_1^{n+1} y_2^{n+1} y_3^{n+1}, z^n$

DOMANDA N.4 – Riempire le mappe delle funzioni di eccitazione J,K per il flip-flop che memorizza la variabile di stato y_1^n ed individuare le espressioni minime a NOR.

	$y_3^n \ x^n$			
$y_1^n \ y_2^n$	00	01	11	10
00				
01				
11				
10				

J_1

	$y_3^n \ x^n$			
$y_1^n \ y_2^n$	00	01	11	10
00				
01				
11				
10				

K_1

$J_1 =$
 $K_1 =$

Il procedimento di analisi

Il procedimento di analisi è formato da cinque passi.

- **Passo 1 – individuazione delle espressioni dei segnali in ingresso a ciascun flip-flop**
- **Passo 2 – individuazione delle espressioni di stato futuro e d’uscita**
- **Passo 3 – individuazione della tabella delle transizioni**
- **Passo 4 – deduzione e studio della tabella di flusso**
- **Passo 5 – deduzione e studio del grafo degli stati**

CASO DI STUDIO – Nello schema di figura sono presenti 4 flip-flop JK: il circuito ha dunque 16 stati.

In particolare si ha:

$$\begin{aligned}
 J_A^n &= K_A^n = X^n \\
 J_B^n &= K_B^n = (X \cdot Q_A \cdot Q_D)'^n \\
 J_C^n &= K_C^n = (X \cdot Q_A \cdot Q_B)^n \\
 J_D^n &= K_D^n = (X \cdot Q_A \cdot Q_B \cdot Q_C + X \cdot Q_A \cdot Q_D)^n
 \end{aligned}$$

I flip-flop JK sono in questo caso impiegati come flip-flop T; inserendo le precedenti espressioni nell’equazione caratteristica si ottiene:

$$\begin{aligned}
 Q_A^{n+1} &= (X \oplus Q_A)^n \\
 Q_B^{n+1} &= ((X \cdot Q_A \cdot Q_D)' \oplus Q_B)^n \\
 Q_C^{n+1} &= ((X \cdot Q_A \cdot Q_B) \oplus Q_C)^n \\
 Q_D^{n+1} &= ((X \cdot Q_A \cdot Q_B \cdot Q_C + X \cdot Q_A \cdot Q_D) \oplus Q_D)^n
 \end{aligned}$$

Una volta tracciata la tabella delle transizioni, è utile fare riferimento alle prime dieci righe ed interpretare lo stato come un numero binario a 4 bit:

$$(S)_2 = Q_D \cdot 2^3 + Q_C \cdot 2^2 + Q_B \cdot 2^1 + Q_A \cdot 2^0$$

Nella colonna $X = 0$ S non viene modificato (la proprietà è vera anche per le restanti sei righe).

Nella colonna $X = 1$ S viene incrementato di una unità, modulo 10.

Da entrambe le precedenti considerazioni discende che il circuito, una volta inizializzato con un valore di S non superiore a 9, non può mai assumere i valori 10, 11, 12, 13, 14, 15.

A regime si ha dunque:

$$(S)_2^{n+1} = (S + X \text{ mod } 10)_2^n$$

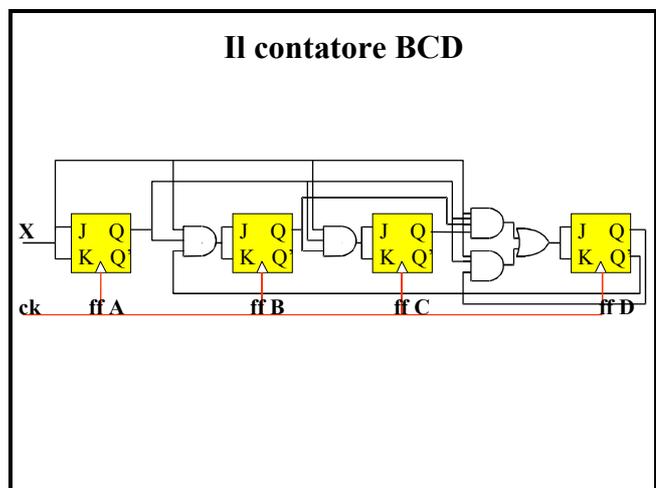


Tabella delle transizioni

	X	
$Q_D \ Q_C \ Q_B \ Q_A$	0	1
0 0 0 0	0 0 0 0	0 0 0 1
0 0 0 1	0 0 0 1	0 0 1 0
0 0 1 0	0 0 1 0	0 0 1 1
0 0 1 1	0 0 1 1	0 1 0 0
0 1 0 0	0 1 0 0	0 1 0 1
0 1 0 1	0 1 0 1	0 1 1 0
0 1 1 0	0 1 1 0	0 1 1 1
0 1 1 1	0 1 1 1	1 0 0 0
1 0 0 0	1 0 0 0	1 0 0 1
1 0 0 1	1 0 0 1	0 0 0 0
1 0 1 0	1 0 1 0	1 0 1 1
1 0 1 1	1 0 1 1	0 1 1 0
1 1 0 0	1 1 0 0	1 1 0 1
1 1 0 1	1 1 0 1	0 1 0 0
1 1 1 0	1 1 1 0	1 1 1 1
1 1 1 1	1 1 1 1	0 0 1 0
	$Q_D^{n+1} \ Q_C^{n+1} \ Q_B^{n+1} \ Q_A^{n+1}$	

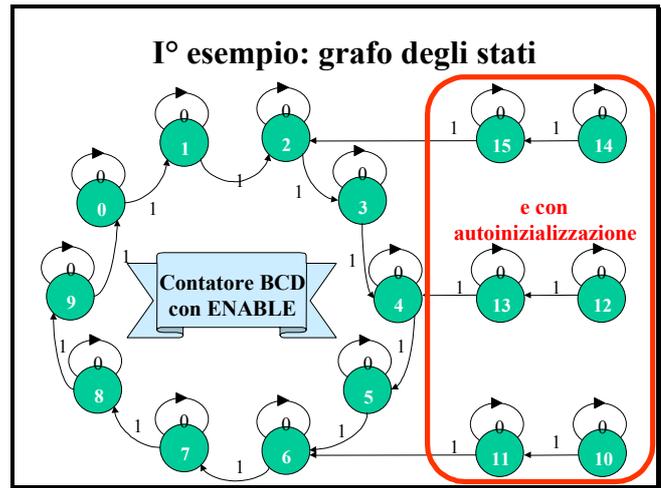
Ci troviamo quindi di fronte ad un **contatore** con **base 10** e stati codificati in **BCD**.

Il segnale X è detto comando di **abilitazione al conteggio**, o anche semplicemente **ENABLE**.

Per studiare meglio il comportamento del circuito conviene tracciare il grafo degli stati.

La caratteristica più evidente è l'anello di 10 stati percorso, un passo dietro l'altro, solo negli intervalli in cui X = 1, cioè quando è abilitato a farlo.

Il circuito presenta anche la già citata proprietà di **autoinizializzazione**: se lo stato iniziale è per caso una delle sei configurazioni non appartenenti al codice BCD ("10", "11", "12", "13", "14", "15"), cosa che può ad esempio capitare quando si fornisce l'alimentazione, sono sufficienti al più due intervalli con X = 1 per rientrare nel ciclo.



CASO DI STUDIO – Si vuole individuare il comportamento della disposizione in cascata di 3 flip-flop D (**registro a scorrimento a 3 bit**).

Dalle equazioni caratteristiche si ottiene:

$$Q_0^{n+1} = x^n$$

$$Q_1^{n+1} = Q_0^n = x^{n-1}$$

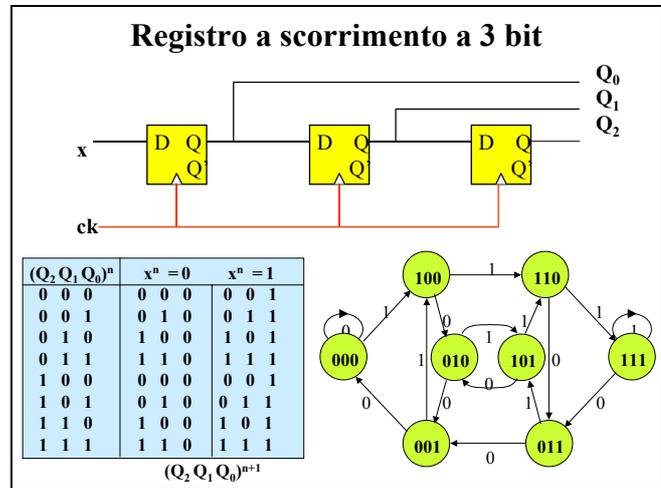
$$Q_2^{n+1} = Q_1^n = x^{n-2}$$

I valori dell'ingresso x scorrono dunque da sinistra verso destra, passando ad ogni clock da un flip-flop al successivo.

Da questa descrizione del comportamento discendono immediatamente la tabella delle transizioni ed il grafo degli stati indicati in figura.

Si notino sul grafo le due stabilità: quando la memoria del circuito è già tutta piena di 1 o di 0, l'arrivo di un valore dello stesso tipo non ne modifica più il contenuto.

Confrontando il risultato appena ottenuto con quanto abbiamo detto a proposito della realizzazione di M3 (v. pag. 135), possiamo trarre una conclusione del tutto generale: **la disposizione in cascata di N flip-flop D realizza in modo molto semplice la conversione da serie a parallelo di un dato di N bit.**



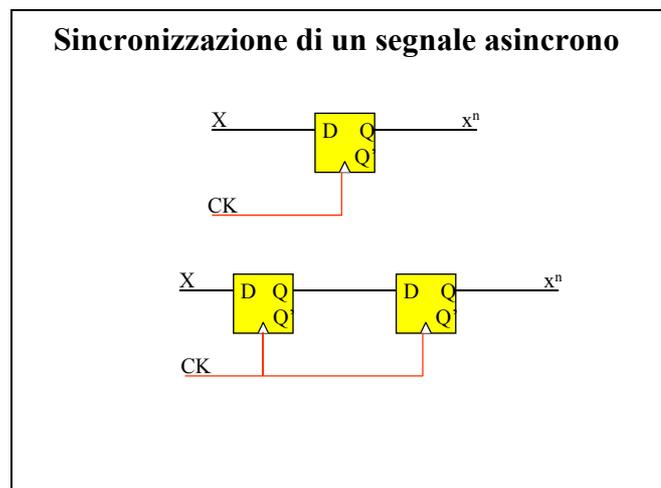
CASO DI STUDIO – La precedente affermazione parte dal presupposto che il registro a scorrimento abbia in ingresso un **segnale sincrono**.

Nei due casi di figura tale ipotesi **non è vera**: X è un **segnale asincrono** che può modificare il suo valore in qualsiasi istante del periodo T₀ di CK. Qual'è dunque in questi casi il ruolo del registro a scorrimento?

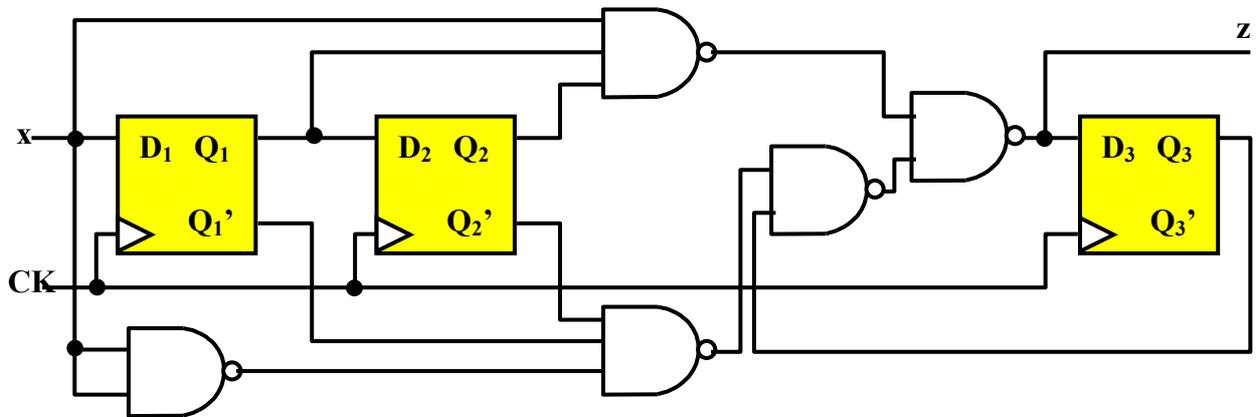
Prima di tutto facciamo l'ipotesi che il progettista conosca la massima frequenza di variazione di X e che la frequenza del clock sia stata di conseguenza da lui fissata ad un valore molto più alto.

Il registro a scorrimento a 1 bit indicato in alto provvede a campionare X ad ogni fronte del clock, restituendone su Q una versione lievemente ritardata, ma **con variazioni tutte allineate con gli istanti di sincronismo**. Il circuito viene quindi impiegato quando una rete sequenziale sincrona deve elaborare segnali d'ingresso asincroni.

APPROFONDIMENTO – Il flip-flop potrebbe però avere un malfunzionamento se la variazione di X capita proprio nell'intervallo $\tau_{SU} + \tau_{H}$. Di norma la cosa viene resa molto poco probabile, fissando ad un valore non troppo alto la frequenza di campionamento. I progettisti "pignoli" impiegano il registro a scorrimento a due bit indicato in basso in figura: il primo flip-flop non può mantenere per molto tempo un valore intermedio tra H e L (v. pag. 77) ed il secondo flip-flop presenta quindi sempre un valore corretto.



Esercitazione N.18



DOMANDA N.1 – Individuare le espressioni che descrivono il comportamento della rete sequenziale sincrona di figura.

$$z^n = \quad (\uparrow)$$

$$= \quad (+, \cdot, ')$$

$$Q_1^{n+1} =$$

$$Q_2^{n+1} =$$

$$Q_3^{n+1} =$$

DOMANDA N.2 – Dedurre dalle precedenti espressioni la tabella delle transizioni.

$Q_1^n \ Q_2^n \ Q_3^n$	$x^n=0$	$x^n=1$
0 0 0		
0 0 1		
0 1 1		
0 1 0		
1 0 0		
1 0 1		
1 1 1		
1 1 0		

$$Q_1^{n+1} \ Q_2^{n+1} \ Q_3^{n+1}, z^n$$

