Calcolatori Elettronici
Reti Sequenziali Sincrone

Ing. Gestionale e delle Telecomunicazioni
A.A. 2009/10
Gabriele Cecchetti

Reti Sequenziali Sincrone

- Sommario:
  - Introduzione, tipi e definizione
  - Condizioni per il corretto funzionamento
  - Rete Sequenziale Sincronizzate di Moore
  - Registri, Flip-flop J-K
  - Rete Sequenziale Sincronizzata di Mealy ritardato
  - Rete Sequenziale Sincronizzata di Mealy
- Riferimenti
  - G. Corsini “Dalle porte AND OR NOT al sistema calcolatore: un viaggio nel mondo delle reti logiche”: cap. “Reti Sequenziali Sincrone”
  - C. Hamacher ““Introduzione all’architettura del Calcolatore”, cap. 2, sez. 2.7, 2.8.

Frequenza di clock
Segnale di clock
Sincronizzazione e commutazione

INTRODUZIONE ALLE RETI SEQUENZIALI SINCRONE

Introduzione alle Reti Sequenziali Sincrone

- In molte situazioni è necessario che lo stato di un Flip-Flop possa cambiare solo in determinati istanti di tempo o intervalli di tempo.
- Le **Reti Sequenziali Sincrone** (RSS) si differenziano dalle RSA per il fatto che lo stato può evolvere solo a determinati istanti.
- Questi istanti sono determinati dalla transizione da 0 a 1 di una particolare variabile detta ***clock***.

Frequenza di clock

- Quindi, per quanto riguarda l'evoluzione dello stato è come se il tempo fosse *discreto* e non continuo.

- La *frequenza di clock* indica quante volte il clock passa da 0 a 1 e viceversa nell'unità di tempo.

  Esempio 1Mhz = 1 ciclo di clock ogni ***μs***.

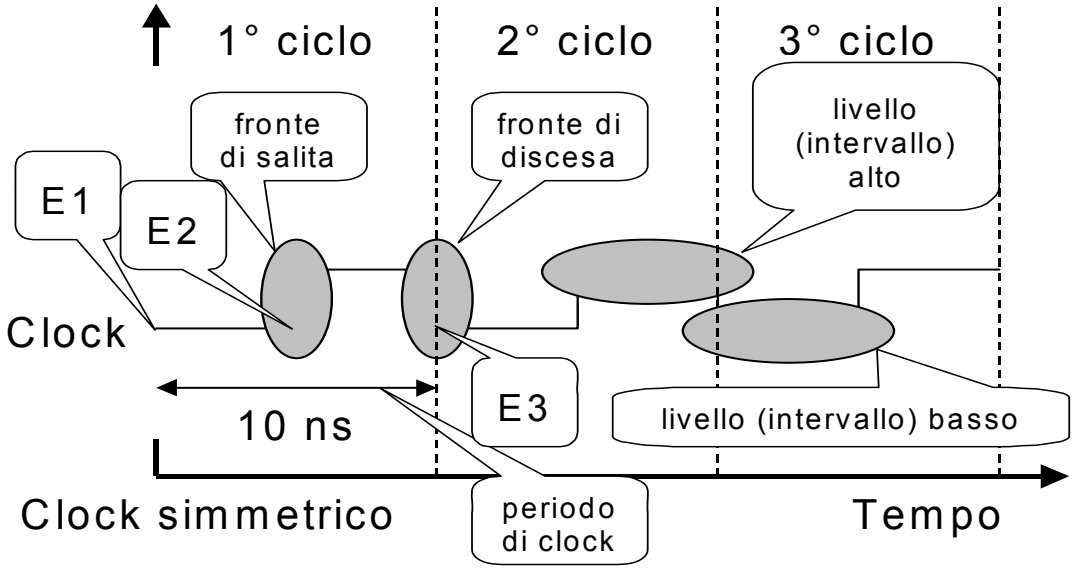
Segnale di Sincronizzazione

- Per realizzare reti sequenziali sincrone pertanto *occorre*:
  - disporre di un **segnale di clock** (o di sincronizzazione) che scandisca gli istanti o intervalli di tempo quando le transizioni di stato possono avvenire,
  - **sincronizzare il bistabile con il clock**.

- Il segnale di clock è un segnale binario, con andamento periodico nel tempo.

- Il segnale di clock è una successione di impulsi:
  - *ogni impulso ha una larghezza costante e due impulsi consecutivi stanno a una distanza costante.*

Segnale di Clock



frequenza di clock = 1 / periodo di clock = 1 / 10 ns = 100 MHz

il ciclo di clock contiene 3 eventi (E1, E2, E3)

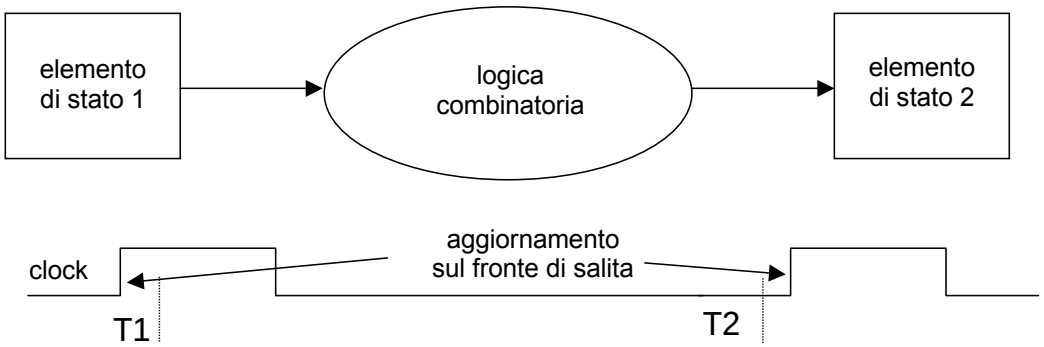
Sincronizzazione (1/2)

- I fattori che differenziano i bistabili riguardano due aspetti:
  - la relazione ingresso-stato (quando gli ingressi sono efficaci)
  - la relazione stato-uscita (quando cambiano le uscite)
- La relazione ingresso-stato (tipo di sincronizzazione) definisce quando gli ingressi modificano lo stato interno del bistabile:
  - sincronizzazione basata sul livello del segnale di sincronizzazione
  - sincronizzazione basata sul fronte del segnale di sincronizzazione

Sincronizzazione (2/2)

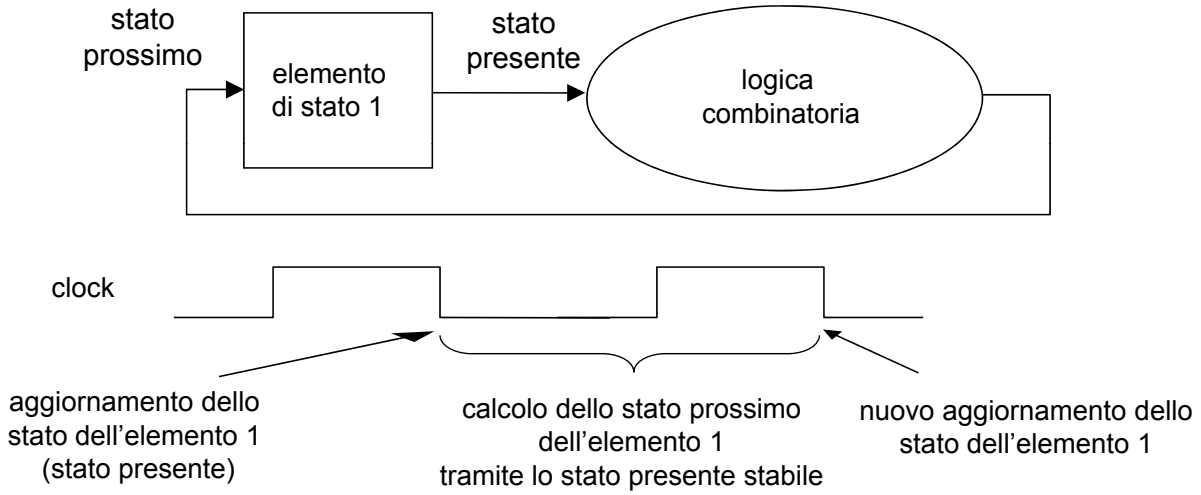
- Sincronizzazione basata sul **livello**:
  - durante tutto l'intervallo di tempo in cui il segnale di sincronizzazione è attivo, qualsiasi variazione sui segnali di ingresso influenza il valore dello stato interno del bistabile (bistabili con commutazione a livello)
- Sincronizzazione basata sul **fronte**:
  - il valore dello stato interno del bistabile viene aggiornato solamente in corrispondenza di un fronte del segnale di sincronizzazione (bistabili con commutazione sul fronte di salita oppure di discesa)

Commutazione sul Fronte (1/2)



il valore dello stato memorizzato nell'elemento 1 al tempo T1 viene utilizzato (con il valore degli ingressi primari) per determinare tramite la rete combinatoria il valore di stato che verrà memorizzato nell'elemento 2 al tempo T2

Commutazione sul Fronte (2/2)



lettura e scrittura dello stato in un ciclo di clock

Specifica di funzionamento

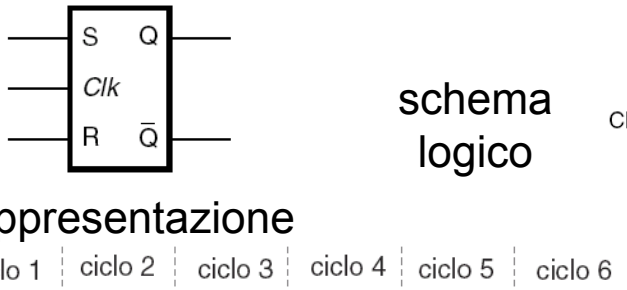
Diagramma temporale

IL FLIP-FLOP SR SINCRONO

Bistabile SR Sincrono (SR-latch)

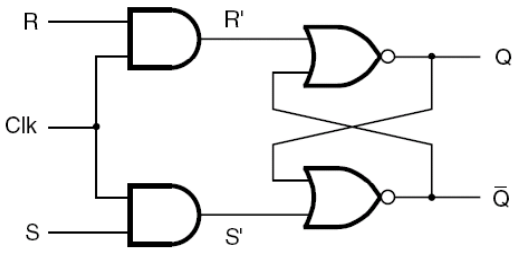
- *Il bistabile SR sincronizzato sul livello ha*:
  - due ingressi S e R (che costituiscono i segnali Set e Reset),
  - *un ingresso di sincronizzazione* (clock),
  - un'uscita Q, il cui valore rappresenta lo stato del bistabile, e un'uscita /Q (stato complementato).
- *Nel bistabile SR sincronizzato sul livello*:
  - *se il clock vale 0*, **gli ingressi S e R non hanno alcun effetto** (latch SR opaco), e *il bistabile mantiene lo stato corrente;*
  - *se il clock vale 1*, **gli ingressi S e R sono efficaci** (latch SR trasparente), e *il comportamento è lo stesso descritto per il bistabile SR asincrono.*

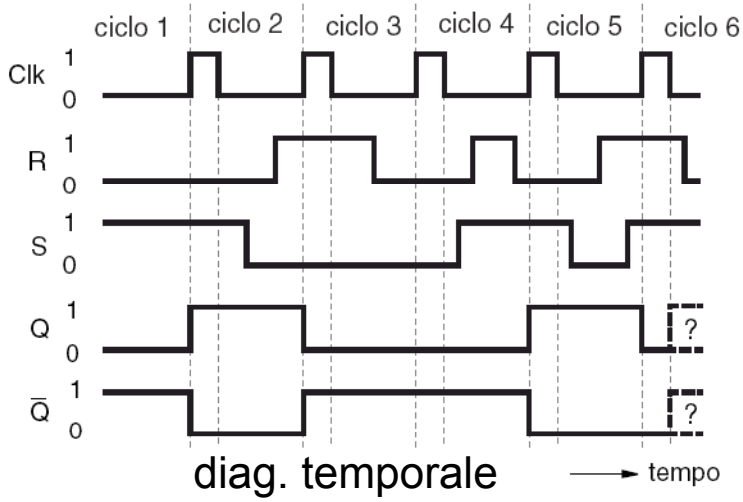
Bistabile SR Sincrono (SR-latch)



rappresentazione

schema logico





diag. temporale

stato futuro

Clk	S	R	Q(t+1)
0	x	x	Q (t) (non cambia)
1	0	0	Q (t) (non cambia)
1	0	1	0
1	1	0	1
1	1	1	x

stato corrente

tab. di transizione

Specifica di funzionamento

Sintesi

Diagramma temporale

IL FLIP-FLOP D-LATCH

Flip-Flop D-latch (o D sincrono)

- Il bistabile D ha:
  - un ingresso D (che rappresenta il dato che verrà memorizzato)
  - un ingresso di sincronizzazione (clock)
  - un'uscita Q, il cui valore rappresenta lo stato del bistabile, e un'uscita /Q (stato complementato)
- Nel bistabile D sincronizzato:
  - *se il clock vale 0*, **l'ingresso D non ha alcun effetto** (*latch D opaco*), e il bistabile mantiene memorizzato il suo stato corrente;
  - *se il clock vale 1*, **l'ingresso D è efficace** (*latch D trasparente*), e *il bistabile memorizza il valore logico* (0 oppure 1) *presente sull'ingresso D.*

Flip-Flop D-Latch

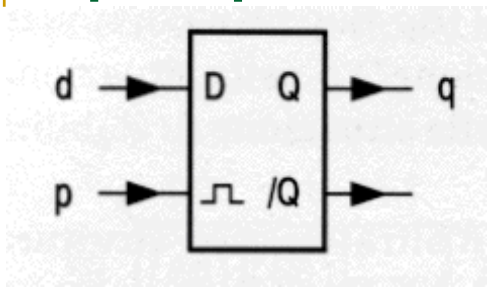
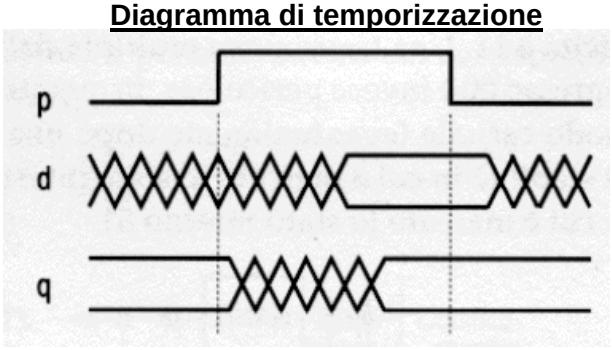


Diagramma di temporizzazione



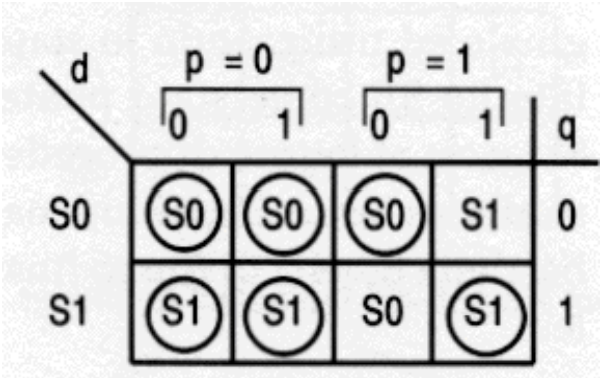
- E' una RSA che si evolve in accordo alle seguenti specifiche:
  - **se p=1 q=d (il FF è in *trasparenza*),**
  - **se p=0 q=q' (il FF è in *conservazione*).**

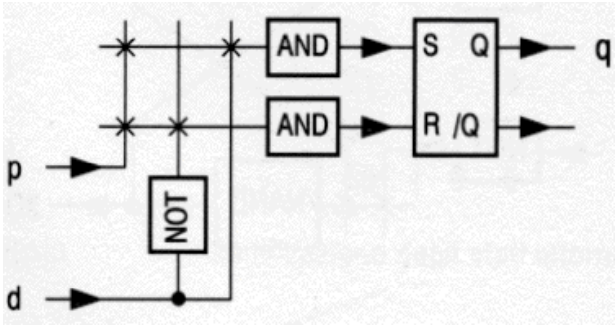
Clk	D	Q(t + 1)
0	x	Q(t)
1	0	0
1	1	1

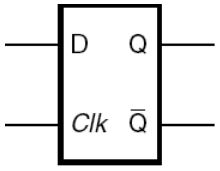
- *Questo FF non può essere utilizzato in un contesto in cui essendo esso in trasparenza il valore dell'uscita Q influenzi attraverso un anello di reazione la var. d.*

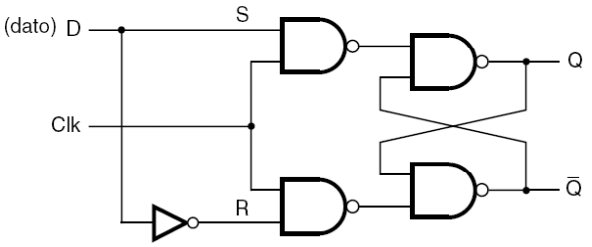
Flip-Flop D-Latch: sintesi

- Tabella degli stati
- Sintesi circuitale

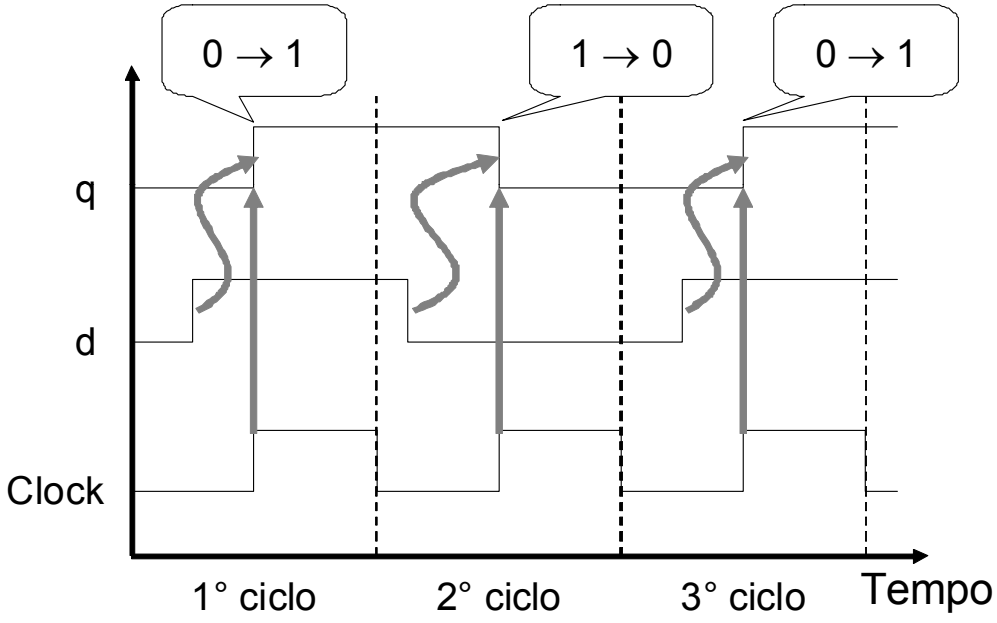








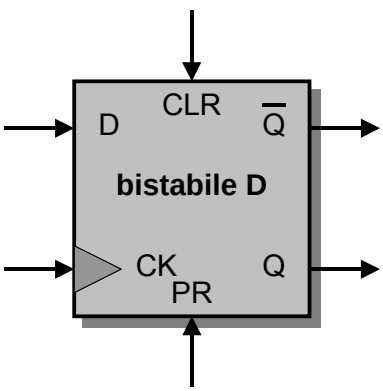
Esempio di diagramma temporale



le variazioni dell'uscita sono sincronizzate con il clock

Comandi di Ripristino e di Precarica

- *Tutti i tipi di bistabile dispongono di varianti dotate di comando di ripristino CLR* (**clear** o reset), *che forza lo stato a 0*.
- Il comando di ripristino è molto utile per (re)inizializzare lo stato.
- *Alcuni tipi di bistabile dispongono anche del comando di precarica PR* (**preset**), *che forza lo stato a 1*.

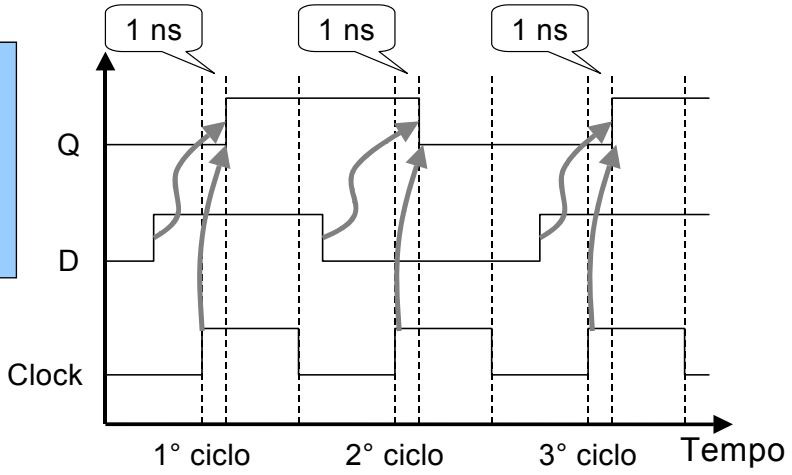


Bistabile di tipo D, dotato di comandi di ripristino e di precarica. Di norma tali comandi sono asincroni, cioè agiscono immediatamente, non appena vengono attivati, senza attendere il clock.

Ritardo di Commutazione

- I bistabili (sincronizzati o no), come le porte logiche, presentano un ritardo di commutazione dell'uscita:
  - la commutazione dell'uscita avviene con un certo ritardo rispetto alla variazione degli ingressi o rispetto al fronte di clock che hanno indotto la transizione di stato
  - Il ritardo di commutazione dipende dalla tecnologia

il bistabile sincrono di tipo D ha un ritardo di commutazione di 1 *ns* dell'uscita rispetto al fronte del clock



flip-flop D a memoria ausiliaria

FLIP-FLOP D MASTER-SLAVE

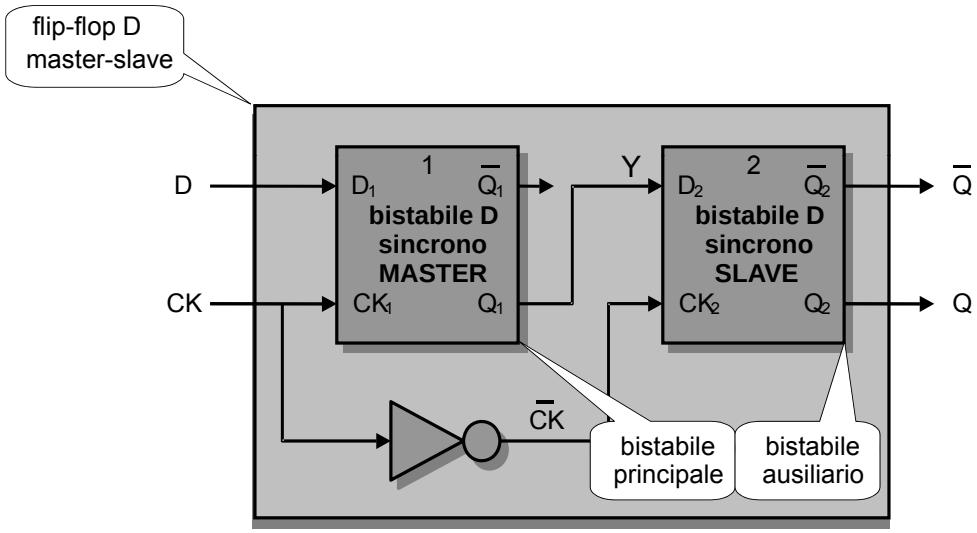
Trasparenza (1/2)

- I latch sincroni (SR o D) presentano, *durante l'intervallo di tempo in cui il clock è attivo*, il fenomeno (indesiderato) di **trasparenza delle uscite**:
  - **in tale intervallo, se gli ingressi si modificano le uscite seguono subito la modifica** (o solo con un lieve ritardo di commutazione),
  - è come se, nell'intervallo attivo del clock, i bistabili non esercitassero alcuna funzione effettiva di memorizzazione.

Trasparenza (2/2)

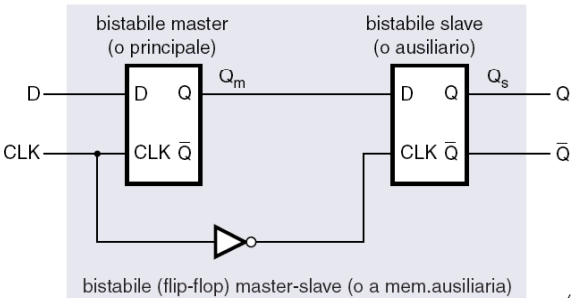
- *Per evitare la trasparenza* **si usano i flip-flop** (D o SR), **costituiti da due bistabili** (D o SR) **sincroni in cascata** (latch), ***così che lo stato modifichi le uscite solo in corrispondenza di un fronte del segnale di sincronizzazione***.
- Nei **flip-flop** si ha:
  - relazione stato-uscita (aggiornamento dell'uscita):
    - sul fronte
  - relazione ingresso-stato (aggiornamento dello stato):
    - a livello (flip-flop master-slave, o a memoria ausiliaria)
    - a fronte (flip-flop edge-triggered) (per i dettagli, vedi il testo)

Flip-Flop D Master-Slave (1/2)

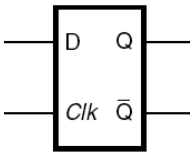


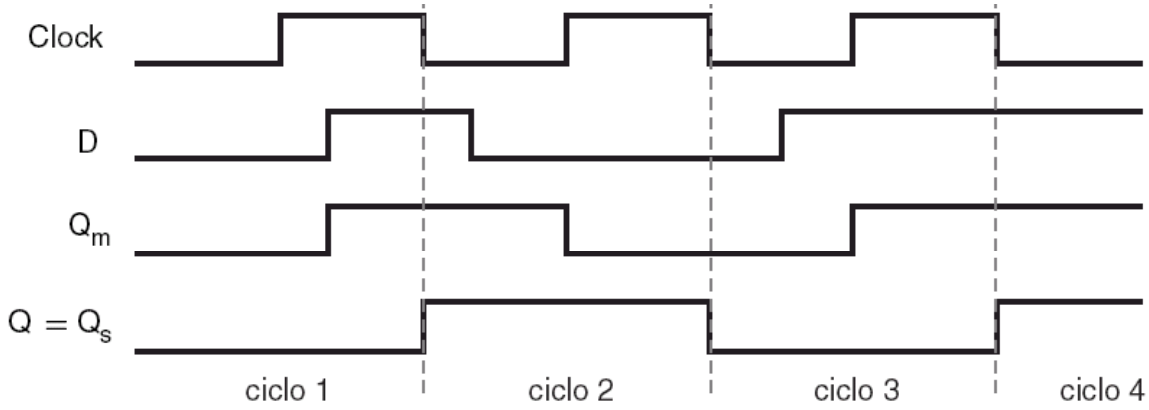
coppia di bistabili sincroni D trasparenti in cascata con clock invertiti
l'insieme dei due non presenta il fenomeno della trasparenza

Flip-Flop D Master-Slave (2/2)



a commutazione sul
fronte di discesa





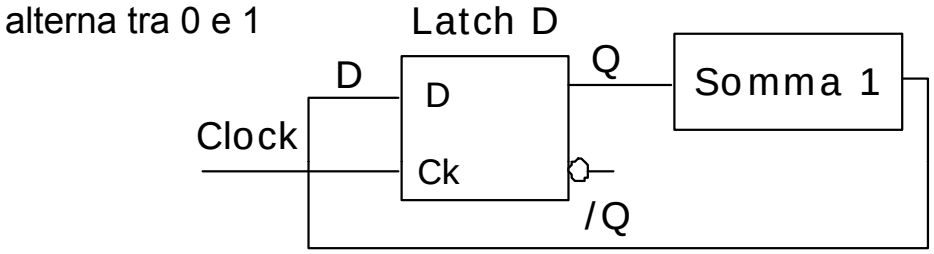
Funzionamento

- Il bistabile principale campiona l'ingresso principale D durante l'intervallo alto del clock, lo emette sull'uscita intermedia Q_m e lo manda all'ingresso D del bistabile ausiliario.
- Il bistabile ausiliario campiona l'ingresso D = Q_m durante l'intervallo basso del clock e lo emette sull'uscita principale Q.
- L'uscita principale Q può variare solo nell'istante del fronte di discesa del clock.
- Perché si evita il fenomeno di trasparenza:
  - nell'intervallo basso del clock, il bistabile SLAVE è in stato di trasparenza
  - nell'intervallo alto del clock, il bistabile MASTER è in stato di trasparenza
  - se l'ingresso D varia durante l'intervallo alto del clock, il bistabile MASTER si comporta in modo trasparente
  - ma il bistabile SLAVE no, perché il suo clock si trova nell'intervallo basso

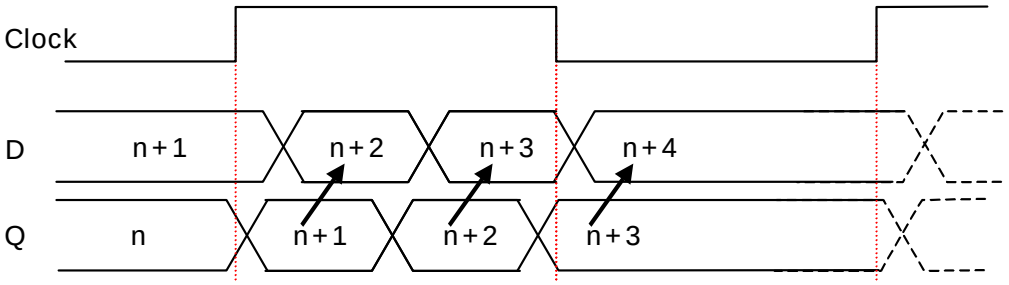
Circuito Sequenziale

- Ricordando la struttura generale del circuito sequenziale (rete combinatoria retroazionata su elementi di memoria), si noti che in genere esso contiene proprio flip-flop master-slave (in genere di tipo D, ma talvolta anche di altro tipo).
- Infatti, l'uso di semplici bistabili sincronizzati (come il bistabile sincrono D o SR) porterebbe a effetti indesiderati, a causa del fenomeno di trasparenza.
- Di seguito, esempio con contatore a un bit (alterna 0 e 1 a ogni ciclo di clock), prima con bistabile D sincrono (malfunzionamento!), poi con flip-flop D.

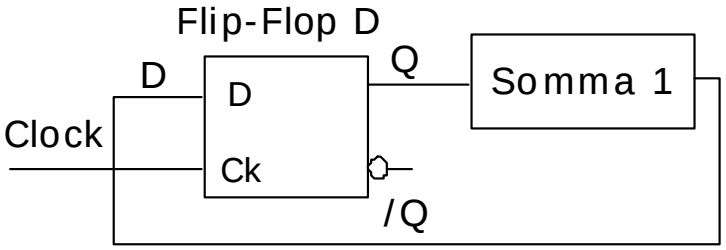
Esempio di Trasparenza

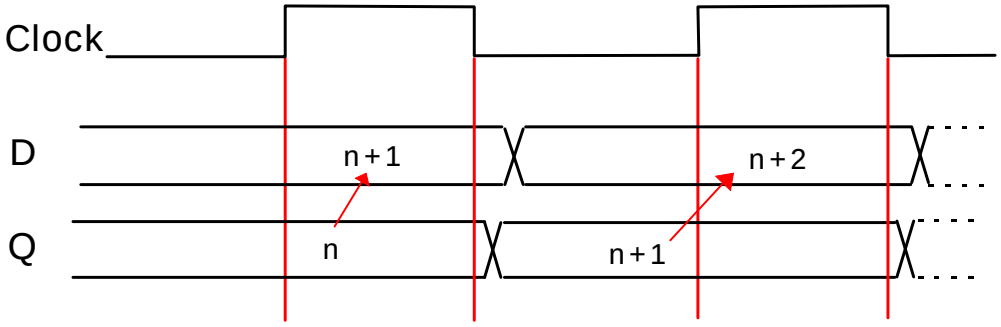


ma non funziona … corre più del necessario
(alterna 0 e 1 più volte in un ciclo !)



Contatore a un Bit



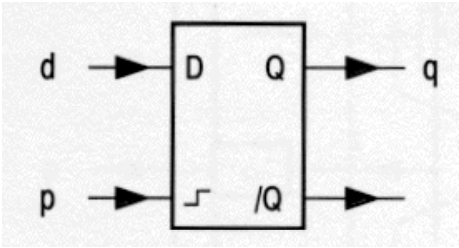


Specifica e sintesi

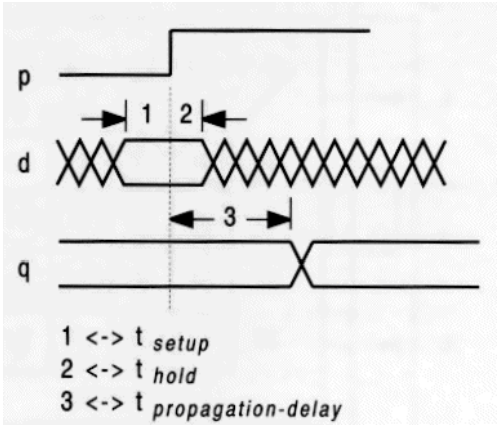
FLIP-FLOP D-POSITIVE EDGE TRIGGERED

Flip-Flop D-positive-edge-triggered

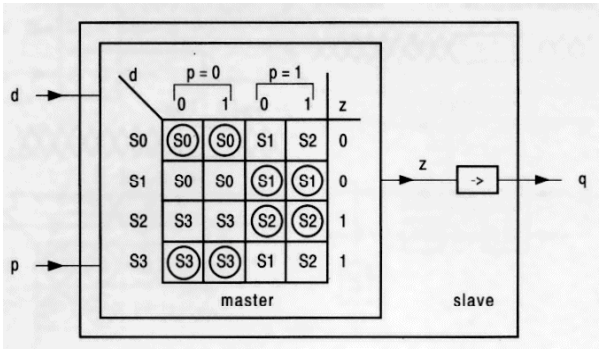
E' una RSA che si evolve in accordo alle seguenti specifiche:



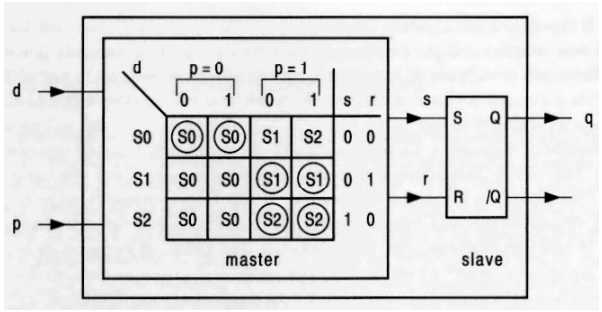
- **Il FF è sempre in conservazione**, **ad esclusione** di quando il valore della var. di ingresso ***p* passa da 0 a 1**.
- Al verificarsi di tale evento il **FF fotografa** il valore della var. di ingresso e dopo un certo tempo lo presenta in uscita come nuovo valore della var. *q*.
- Quando *q* assume questo nuovo valore il FF non è già più sensibile al valore della var. di ingresso *d*.



Flip-Flop D-positive-edge-triggered: realizzazione master-slave

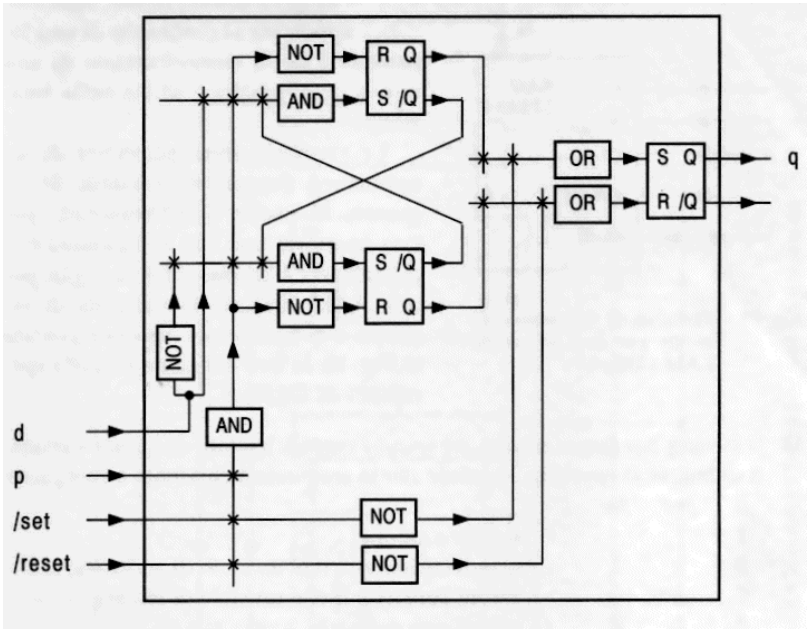


- Slave come elemento di ritardo



- Slave come FF SR

Flip-Flop D-positive-edge-triggered: sintesi



- I piedini di **/set** e **/reset** sono utilizzabili per l'inizializzazione del FF

qualche nozione di base

CIRCUITO INTEGRATO

Circuiti integrati (1/2)

- Un circuito integrato è una piastrina di silicio (o chip), quadrata o rettangolare, sulla cui superficie vengono realizzati e collegati transistori e dunque porte logiche, che complessivamente realizzano uno o più circuiti digitali:
  - di solito la piastrina di silicio di un circuito integrato ha dimensioni comprese tra:
    - 5×5 mm, e
    - 1×1 cm (di rado superiore);
  - la piastrina di silicio integra i transistori, i collegamenti tra transistori e i collegamenti con i morsetti di ingresso/uscita del chip.

Circuiti integrati (2/2)

- Su un singolo chip si possono integrare:
  - porte logiche sparse e indipendenti,
  - una o più reti combinatorie, con funzioni definite,
  - una o più reti sequenziali,
  - unità funzionali complesse: memoria, processore, unità di controllo di periferiche, e così via.

Famiglie di Circuiti Integrati

- I circuiti integrati (CI) sono classificati in base alle loro dimensioni, cioè al numero di porte logiche contenute:
  - **SSI** (*Small Scale Integrated*): CI a scala di integrazione piccola, da 1 a 10 porte
  - **MSI** (*Medium Scale Integrated*): CI a scala di integrazione media, da 10 a 100 porte
  - **LSI** (*Large Scale Integrated*): CI a scala di integrazione grande, da 100 a 100.000 porte
  - **VLSI** (*Very Large Scale Integrated*): CI a scala di integrazione molto grande, > 100.000 porte
- Ogni famiglia ha degli usi caratteristici nei calcolatori e in generale nei dispositivi elettronici, che dipendono dalle dimensioni, cioè dalla quantità di porte presenti nel circuito integrato stesso.

Usi Caratteristici

- **SSI**: piccoli circuiti digitali di contorno a circuiti più grandi e impegnativi (*glue logic*, logica di “incollamento”).
- **MSI**: circuiti digitali semplici, dotati di un’unica funzione ben definita (ad. es. scambio di due o più segnali, confronto di numeri, addizione o sottrazione di numeri);
- **LSI**: circuiti digitali complessi, dotati di funzionalità multiple, eventualmente programmabili (ad es. un intero insieme di op. aritmetiche – addizione, sottrazione, moltiplicazione, divisione o ALU – memorie di piccole dimensioni, processori semplici).
- **VLSI**: circuiti digitali molto complessi o di grandi dimensioni, spesso programmabili (ad es. processori (dai microcontrollori ai processori di uso generale - Pentium, SPARC, memorie, da qualche Kbyte in su, unità di controllo delle periferiche).La progettazione di un circuito integrato VLSI è di tipo modulare, dove ogni modulo realizza una funzione.

Blocchi funzionali sequenziali di base:
registro parallelo e registro parallelo con caricamento

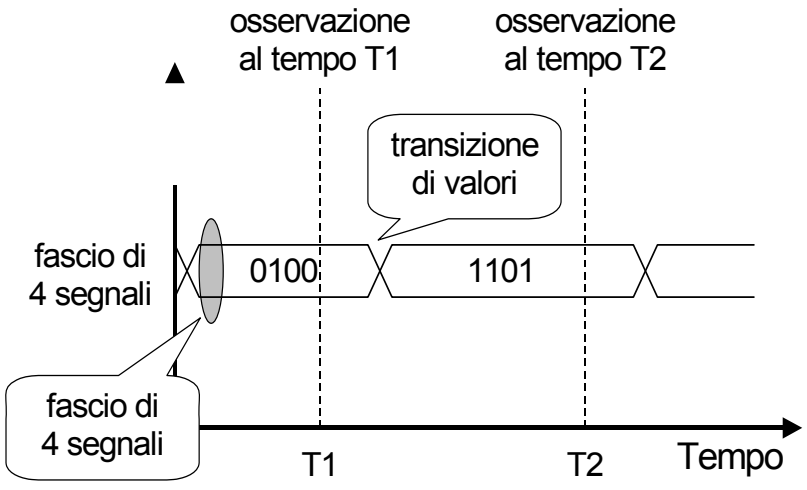
BLOCCHI FUNZIONALI LOGICI SEQUENZIALI

Blocchi Funzionali Sequenziali

- Tipici principali componenti sequenziali di libreria:
  - registro parallelo
  - registro a scorrimento
  - banco di registri (vedi capitolo relativo)
  - memoria (vedi capitolo relativo)
- Ognuno di questi blocchi ammette numerose versioni e varianti.

Diagramma Temporale

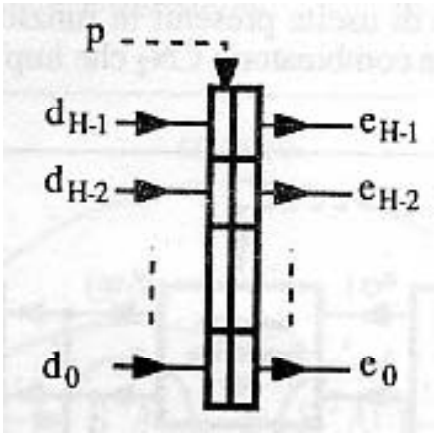
Come rappresentare un fascio di segnali:



al tempo T1 i 4 segnali valgono 0100, al tempo T2 i 4 segnali valgono 1101

Registri

- Ugual numero di variabili di ingresso e uscita.
- Lo stato di uscita all'istante t_i coincide con lo stato d'ingresso presente all'istante t_{i-1}.
- E' dotato di un ingresso per il reset asincrono (tutte le varabili di uscita vengono fissate al livello 0).



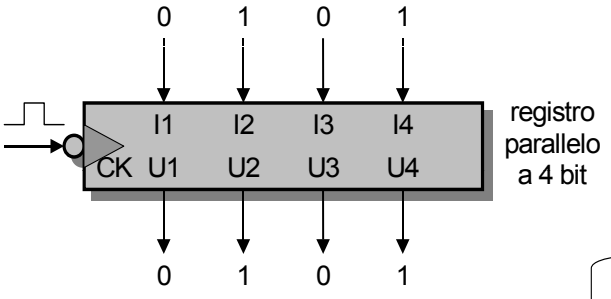
- **Il registro è una collezione di FF D+edge triggered temporizzati in parallelo.**

Registro Parallelo

- Il registro parallelo è una schiera di $n \geq 1$ flip-flop di tipo D (molto spesso master-slave). Ha:
  - $n \geq 1$ ingressi $I_1, \ldots, I_n$
  - $n \geq 1$ uscite $U_1, \ldots, U_n$
  - e naturalmente l'ingresso di clock CK

- A ogni ciclo di clock, il registro legge e memorizza nel suo stato la parola di *n* bit presente in ingresso, e la presenta sulle *n* uscite nel *ciclo successivo*.

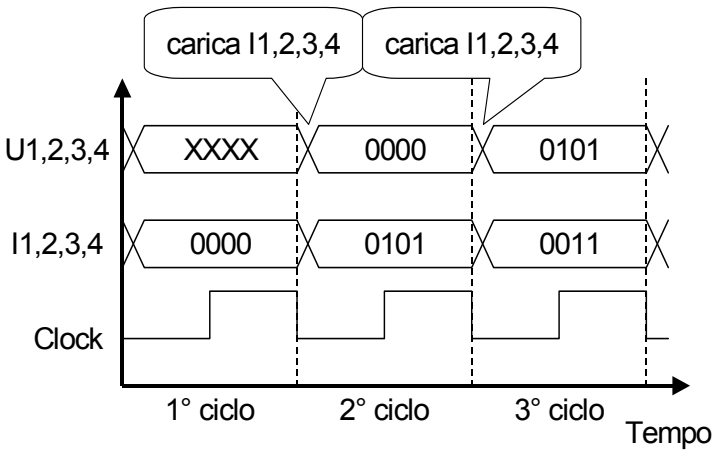
Simbolo e Funzionamento



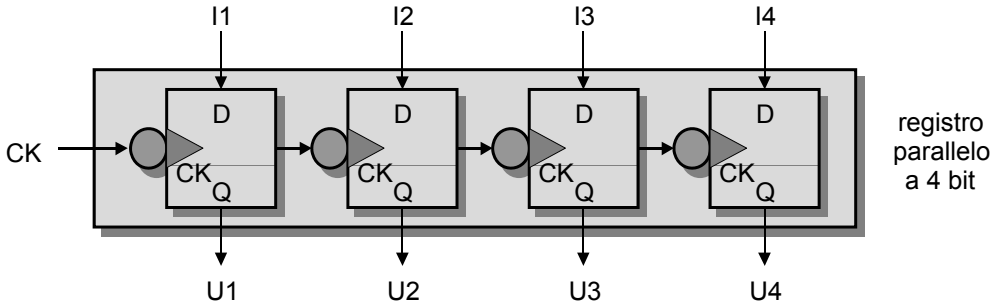
registro parallelo a 4 bit

- carica 0000
- carica 0101
- ecc ...

diagramma temporale



Progetto in Stile Funzionale



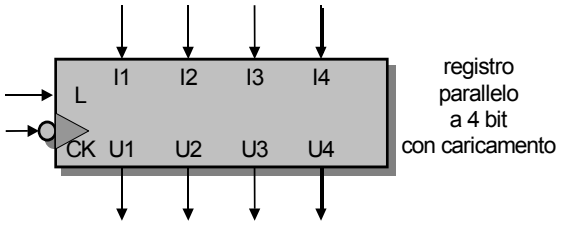
registro parallelo a 4 bit

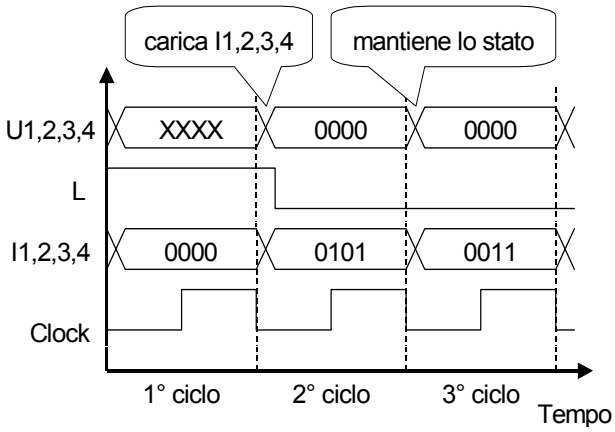
registro parallelo progettato in stile funzionale, usando 4 flip-flop D sincroni sul fronte (di discesa)

Nota bene: se si usassero dei bistabili D trasparenti (sincronizzati sul livello), durante il livello alto del clock il registro sarebbe esso stesso del tutto trasparente, e dunque non si comporterebbe come un registro ...

Registro con Caricamento

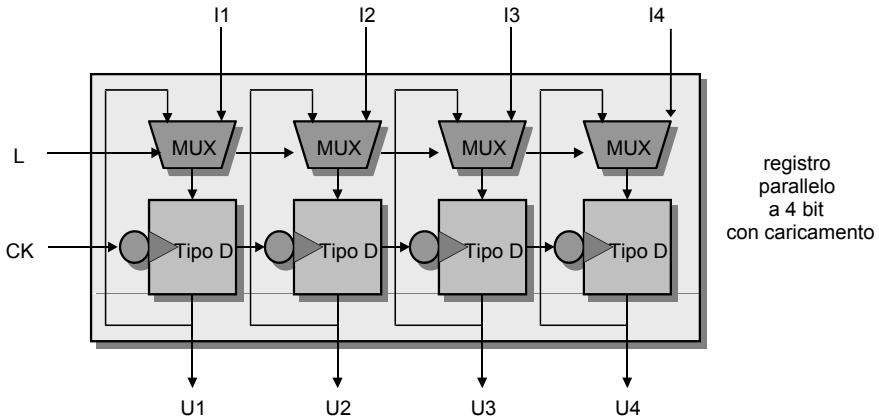
- Funziona come il registro parallelo, ma ha in aggiunta un ingresso di comando di caricamento (L, ingresso di *Load*):
  - se il comando L è attivo (p. es. L = 1), la parola in ingresso al registro viene memorizzata nel registro stesso e presentata in uscita nel ciclo successivo;
  - altrimenti (cioè L = 0), il registro mantiene il suo stato corrente di memorizzazione.





- carica 0000
- mantiene 0000
- ecc ...

Progetto in Stile Funzionale



registro parallelo progettato in stile funzionale,
usando 4 flip-flop D sincroni sul fronte (di discesa),
e 4 multiplexer a un ingresso di selezione e due ingressi dati

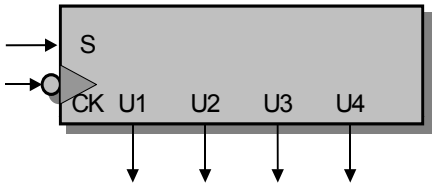
Varianti e Integrazioni

- Registro parallelo con comando di ripristino, per azzerarne il contenuto.
- Registro parallelo con comando di ripristino e di precarica.
- Registro parallelo universale, riunisce le funzioni di tutti i registri precedenti: comandi di caricamento, comando di ripristino e comando di precarica.

Registro a Scorrimento

- Il registro a scorrimento è una successione di $n \geq 1$ flip-flop di tipo D (sempre di tipo master-slave) collegati in cascata. Ha:
  - un ingresso seriale S
  - $n \geq 1$ uscite parallele $U_1, \ldots, U_n$
  - e naturalmente l'ingresso di clock
- A ogni ciclo di clock, fa scorrere di un bit verso destra la parola memorizzata, perdendo il bit più a destra e aggiungendo a sinistra il bit presente sull'ingresso seriale.

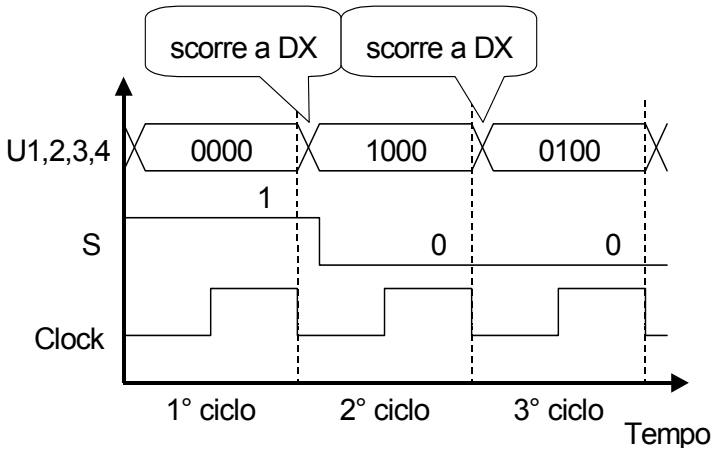
Simbolo e Funzionamento



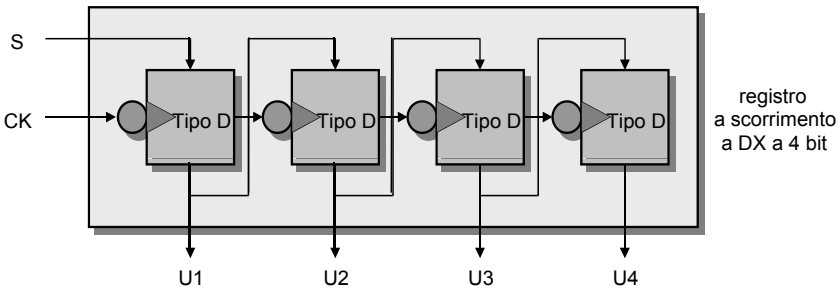
registro
a scorrimento
a DX a 4 bit

- scorre a DX
- scorre a DX
- ecc ...

diagramma temporale



Progetto in Stile Funzionale



registro a scorrimento a DX progettato in stile funzionale,
usando 4 flip-flop D sincroni sul fronte (di discesa) collegati in cascata

Nota bene:

se si usassero dei bistabili D trasparenti (sincronizzati sul livello), durante il livello alto del clock un bit potrebbe propagarsi lungo l'intera catena di bistabili … non sarebbe un comportamento accettabile!

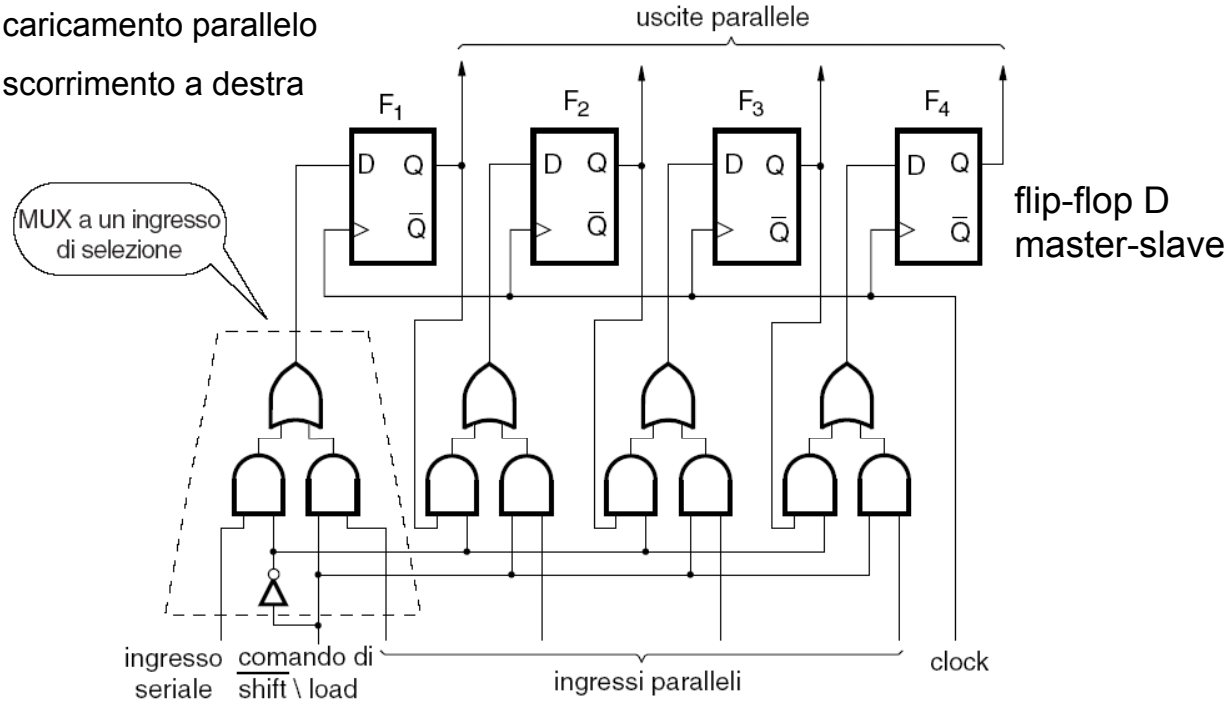
Varianti e Integrazioni

- Registro a scorrimento a SX (sinistra).
- Registro a scorrimento universale: DX (destra) e SX (ha un comando di scelta del verso di scorrimento).
- Registro a scorrimento (DX o SX) con funzione di caricamento parallelo.
- Registro parallelo / a scorrimento universale: riunisce le funzioni dei registri parallelo e a scorrimento universali.
- Registro IN seriale / OUT seriale.
- Registro IN parallelo / OUT seriale.
- Registro IN parallelo/seriale OUT parallelo/seriale.

Esempio

Funzioni:

- caricamento parallelo
- scorrimento a destra



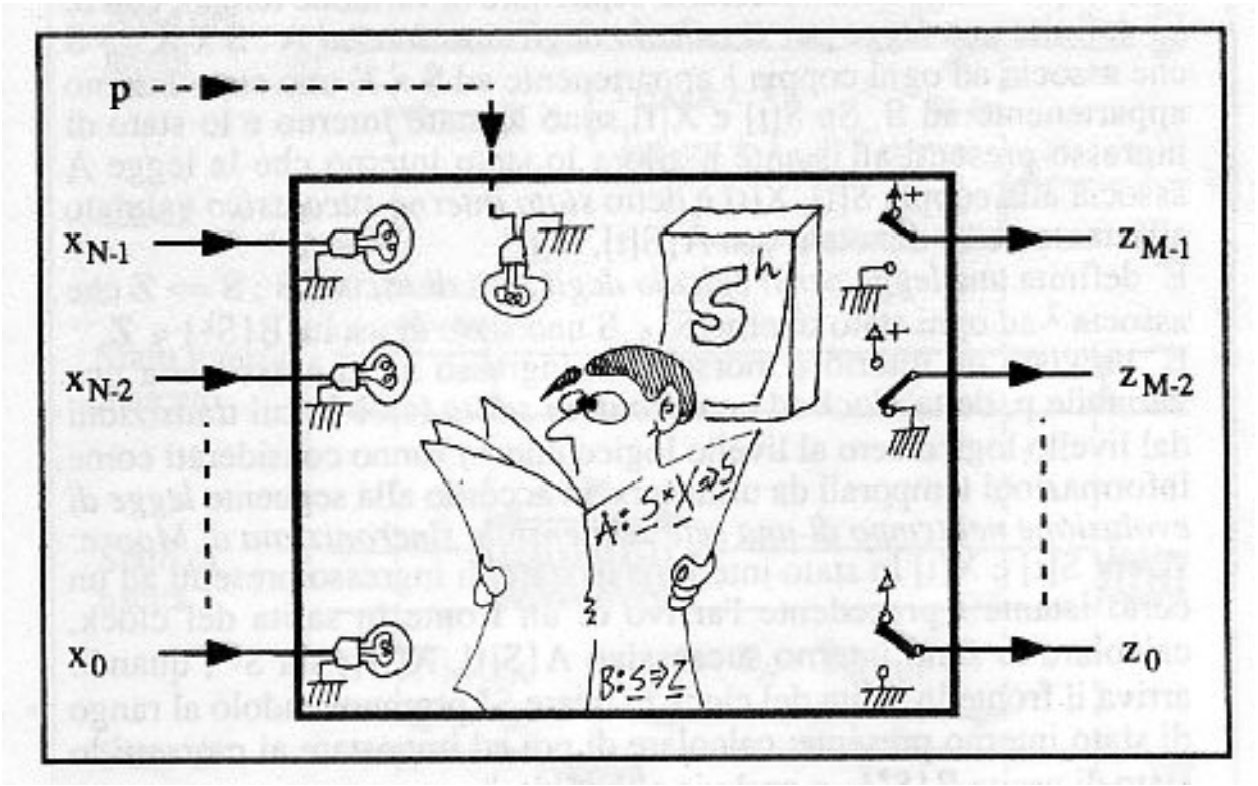
RSS di Moore, Flip-Flop J-K
RSS di Mealy
RSS di Mealy ritardato

MODELLI STRUTTURALI PER RSS

Tipi di Reti Sequenziali Sincrone

- **Moore**: l'uscita dipende solo dallo stato
- **Mealy**: l'uscita dipende sia dallo stato che dall'ingresso

Rete sequenziale sincrone di Moore



RSS di Moore: definizione

RSS di Moore è una qualunque struttura che soddisfa ai seguenti requisiti:

- è dotata di *N var. d'ingresso* $x_{N-1},x_{N-2},\ldots,x_0$
- è dotata di *M var. di uscita* $z_{M-1},z_{M-2},\ldots,z_{0a'}$
- è dotata di un *meccanismo di marcatura* che seleziona ad ogni istante uno ed un solo elemento appartenente ad un opportuno insieme $S=\{S^0,S^1,\ldots,S^{K-1}\}$,
- implementa una *legge per gli stati interni* **A: SxX ⇒ S,**
- implementa una *legge per gli stati di uscita* **B: S ⇒ Z,**
- è dotata di *un ingresso p*, detto **clock**, le cui transizioni dal 0 a 1 vanno considerate come informazioni temporali da utilizzarsi in accordo alla seguente ***legge di evoluzione del tempo***:
  "detti X[t] e S[t] sono lo stato d'ingresso presente e lo stato interno presenti ad un certo istante *t* precedente l'arrivo di un fronte in salita del clock, *calcolare* lo stato lo stato interno successivo **S*=A{S[t],X[t]}**; *quando arriva il fonte in salita del clock marcare S** promuovendolo al rango di stato interno presente; *quindi calcolare ed impostare lo stato di uscita* **Z=B{S*}**, e così via all'infinito.

Condizioni per il corretto funzionamento di una RSS di Moore (1/2)

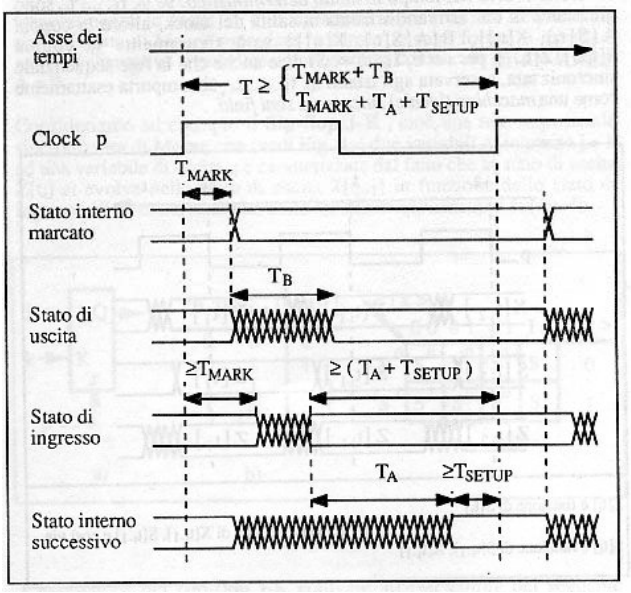
Il tempo T fra due fronti in saluta del clock

sia sufficiente a che la rete possa:

- utilizzare il meccanismo di marcatura (per promuovere S* a S;
- calcolare il nuovo stato di uscita (sia T_B il tempo necessario);
- calcolare il nuovo stato interno successivo (sia T_A il tempo necessario);
- avere ancora un margine di tempo T_{SETUP} legato alla natura del meccanismo di marcatura.

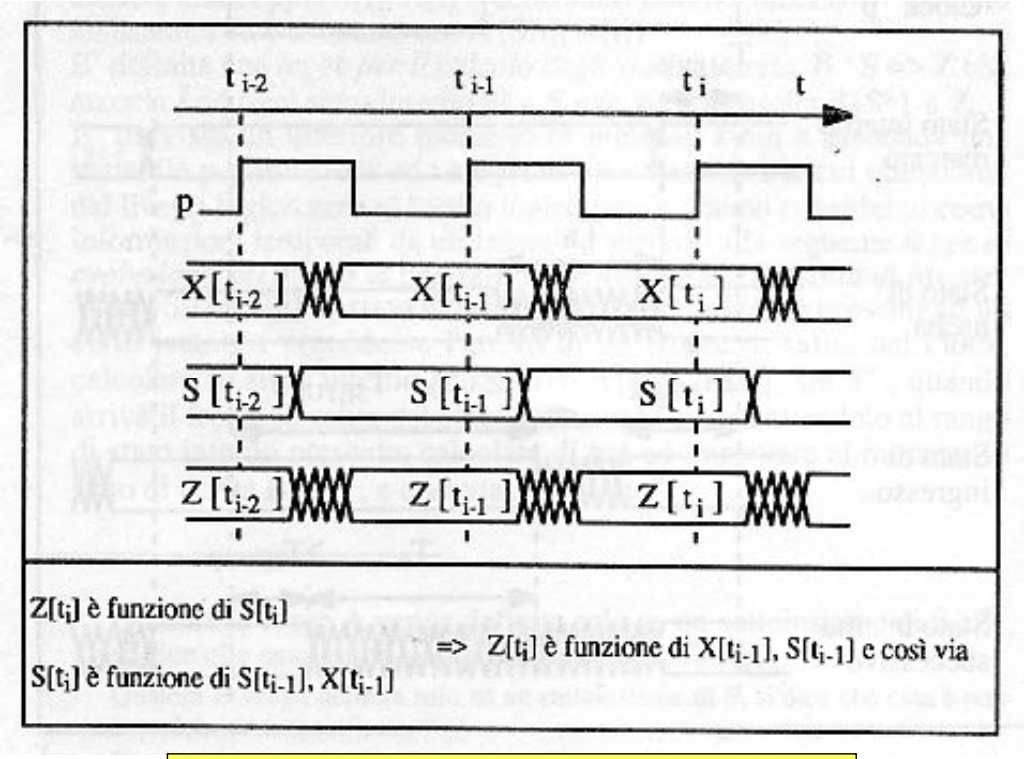
Condizioni per il corretto funzionamento di una RSS di Moore (2/2)

- $T-(T_{MARK}+T_A+T_{ASETUP})$ è il tempo a disposizione per cambiare X e far estinguere tutti i transitori.
- T_A+T_{ASETUP} è il tempo in cui X non deve essere rimosso (calcolo S* e marcatura)



In questo modo la rete si comporta in modo deterministico.

Evoluzione temporale di una RSS di Moore



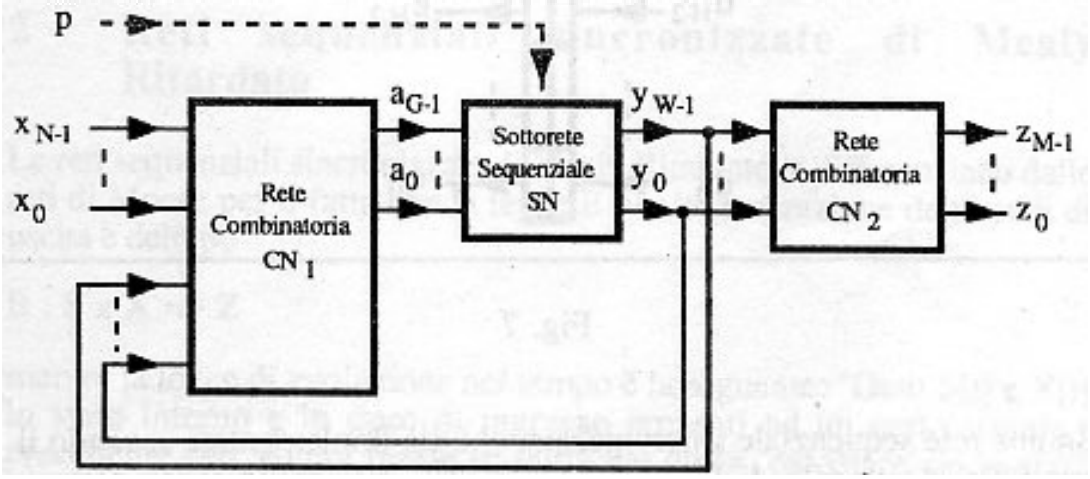
Macchina (ideale) sincrona a stati finiti.

Non trasparenza delle RSS di Moore

- Lo stato di uscita non cambia negli intervalli di tempo:

$$t_i - T_{SETUP} \, , \, t_i + T_{MARK}$$

Modelli strutturali per le RSS di Moore (1/2)



- S = $\{S^0, S^1, \ldots, S^{K-1}\}$ sono codificati tramite K delle 2^W combinazioni dei valori delle var. di stato $y_{w-1}, y_{w-2}, \ldots, y_0$ con $W \geq \lceil \log_2 K$;
- La sottorete sequenziale SN marca all'uscita lo stato interno;
- CN_1 + SN implementano sia la legge A che il meccanismo di marcatura (operazione non istantanea, SN produce stati spuri che però non creano malfunzionamenti perché SN non è trsparente;
- CN_2 implementa la legge B.

Modelli strutturali per le RSS di Moore (2/2)

- SN (RSS primitiva) può essere implementata con
  - registri,
  - flip-flop SR sincronizzati,
  - flip-flop J-K sincronizzati.

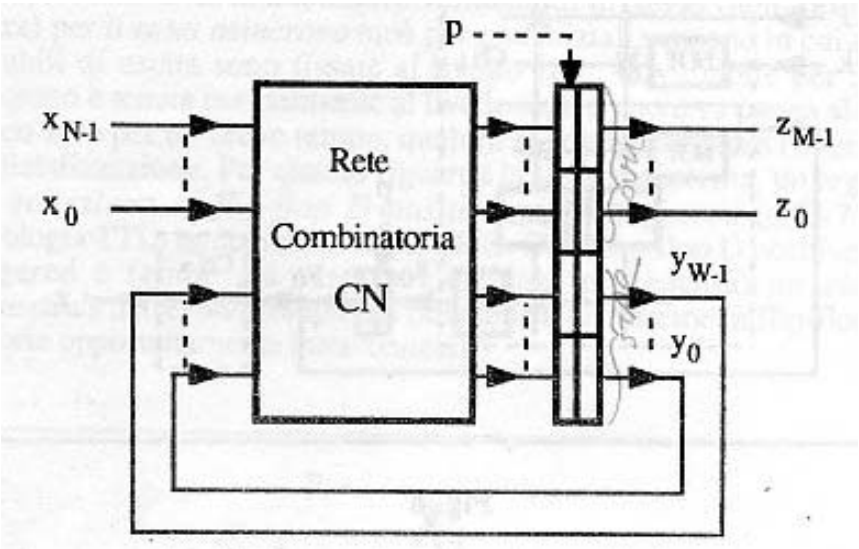
Reti sequenziali sincronizzate di Mealy Ritardato

Differenza rispetto alle RSS di Moore:

- **B : S x X → Z**
- Legge di evoluzione nel tempo:

  "Detti X[t] e S[t] lo stato d'ingresso presente e lo stato interno presenti ad un certo istante t precedente l'arrivo di un fronte in salita del clock, *calcolare* **sia** lo stato lo *stato interno successivo* S*=A{S[t],X[t]} **che** lo *stato di uscita* Z=B{S[t],X[t]} e *promuoverli al rango di stati presenti solo quando arriva il fronte in salita del clock*, e così via all'infinito."

Modello strutturale per RSS di Mealy Ritardato



- Definiamo:
  - registro STAR, la porzione di registro che memorizza lo stato S;
  - registro OUTR, la porzione di registro che memorizza il prossimo stato di uscita.

Condizioni per il corretto funzionamento

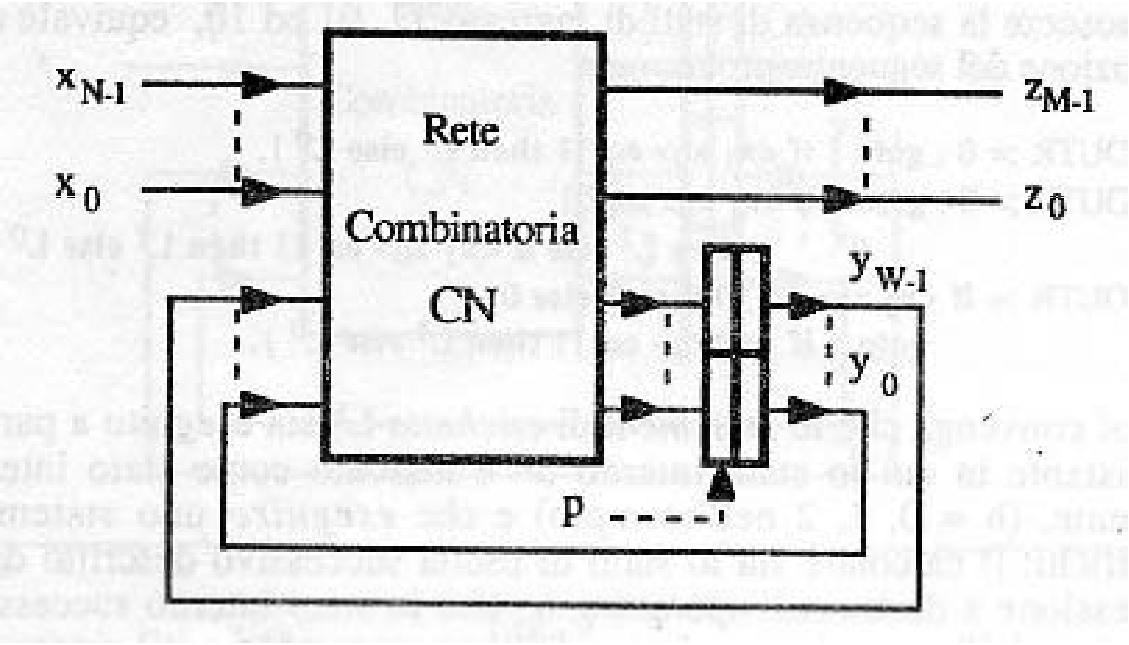
- Condizioni per il corretto funzionamento:
  - $T \geq T_{MARK} + \max\{T_A, T_B\} + T_{SETUP}$
  - X[t] stabile da $\max\{T_A, T_B\} + T_{SETUP}$ secondi prima a T_{MARK} secondi dopo
- Rispettando queste condizioni la rete si evolve in modo deterministico e
  $Z[t_i]$ è funzione di $X[t_{i-1}]$, $X[t_{i-1}]$, …, $X[t_0]$, $S[t_0]$
- Vantaggi:
  - minor numero di stati interni a parità di problema risolto,
  - si adatta meglio ad essere implementato sulle PLA

Reti sequenziali sincronizzate di Mealy

Differenza rispetto alle RSS di Mealy ritardato:

- le variabili di uscita sono prelevate direttamente dalla rete combinatoria
  - → l'ingresso influenza direttamente l'uscita
  - → svantaggio: troppi transitori sull'ucita
    - → svantaggio: rischio di creare anelli instabili di reti combinatorie se più reti di Mealy vengono interconnesse nel costruire unità complesse
  - → Per questi motivi non vengono utilizzate.

Modello strutturale per RSS di Mealy



Specifica di funzionamento

Sintesi

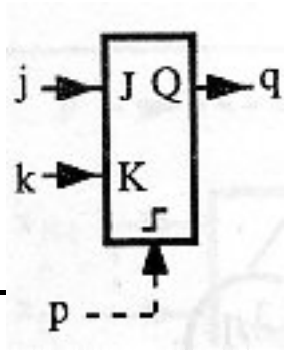
Esempio applicativo

FLIP-FLOP J-K

Flip Flop J-K : specifica di funzionamento

- Specifica di funzionamento:

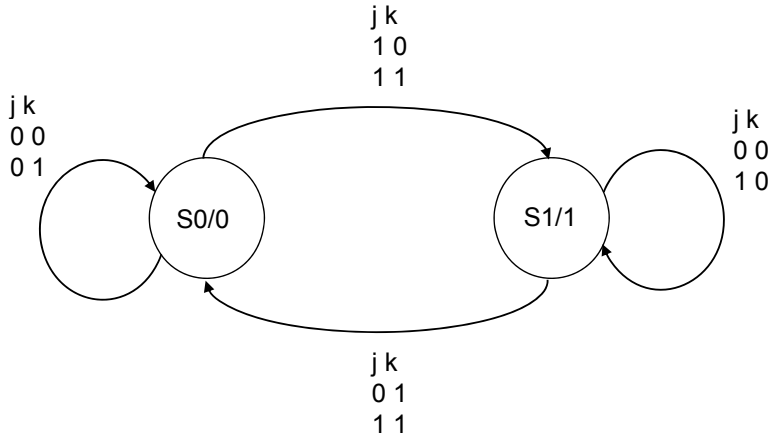
j	k	q	
1	0	1	*Set*
0	1	0	*Reset*
0	0	q'=q	*Conservazione*
1	1	q'=/q	*Commutazione*



- Tabella di applicazione

q	q'	j	k
0	0	0	-
0	1	1	-
1	1	-	0
1	0	-	1

Flip-Flop J-K: diagramma a stati e tabella di flusso



<jk>	00	01	11	10	<q>
S^0	S^0	S^0	S^1	S^1	0
S^1	S^1	S^0	S^0	S^1	1

Flip-Flop J-K: considerazioni

- Non ci sono problemi di stabilità perchè il tempo è discreto. **Lo stato rimane sempre staile per un intero ciclo di clock e così anche l'uscita.**
- **Non è importante se CN_1 ha alee oppure no.**
- **Non c'è il problema della corsa delle variabili di stato, quindi in generale sono ammesse transizioni multiple in ingresso.**

→ Quindi *l'introduzione del clock semplifica la realizzazione dei circuiti* a patto che la temporizzazione sia rispettata.

Flip-Flop J-K: sintesi

1. Modello strutturale scelto
   - La sottorete SN è un registro
2. Codifica degli stati

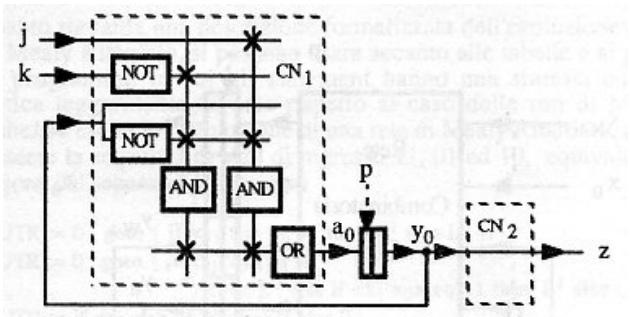
   S_0 = <0>

   S_1 = <1>
   - Con questa codifica degli stati CN_2 è un corto circuito
3. Tabella degli stati

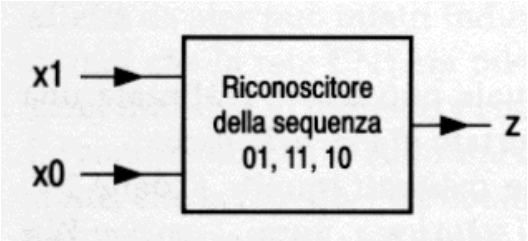
y jk	00	01	11	10	q
0	0	0	1	1	0
1	1	0	0	1	1

4. Sintesi CN_1

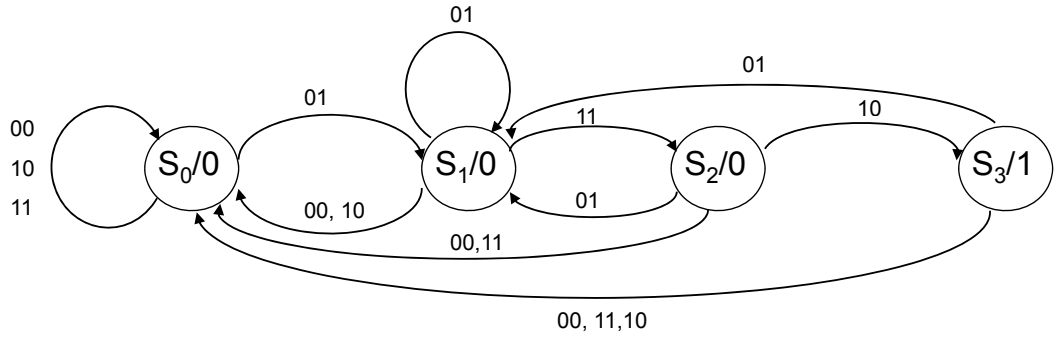
   a = j/y + /ky



Riconoscitore di sequenza 01,11,10 (1/3)



- Diagramma di flusso



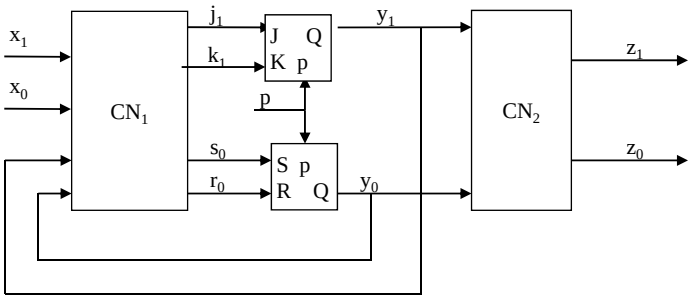
Riconoscitore di sequenza 01,11,10 (2/3)

1. Tabella di flusso
2. Codifica degli stati interni
3. Tabella di flusso con gli stati interni codificati

stato interno	codifica y1 y0
S0	0 0
S1	0 1
S2	1 1
S3	1 0

Riconoscitore di sequenza 01,11,10 (3/3)

- Modello strutturale



- Leggi caratterizzanti CN1 e CN2
  - CN1: a partire dalla tabella degli stati codificati,
    1) si calcola calcolano il valore che deve essere impostato sui piedini j1,k1 e j0,k0 affinché avvengano le transizioni di stato richieste;
    2) si dividono le tabelle di pilotaggio così ottenute in modo da ottenere le espressioni algebriche di j1, k1, j0 e k0.
  - CN2: a partire dalla codifica degli stati e delle uscite corrispondenti (in questo caso $z=y_1/y_0$)