

Architettura degli Elaboratori

Lezione 13 – Logica sequenziale sincrona

Giuseppe Cota

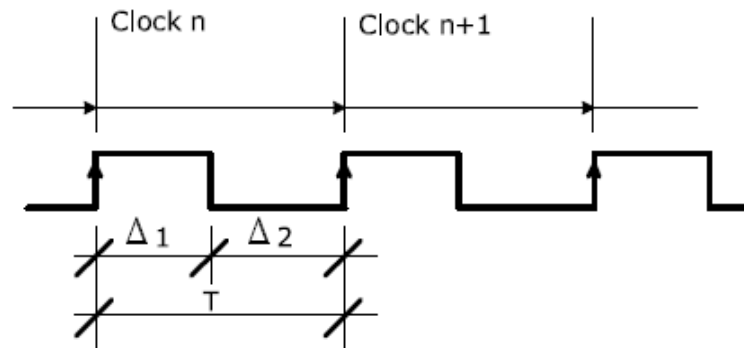
Dipartimento di Scienze Matematiche Fisiche e Informatiche
Università degli Studi di Parma

Indice

- ❑ Flip-flop sincroni
- ❑ Reti sequenziali sincrone basate sui flip-flop

Reti sequenziali sincrone

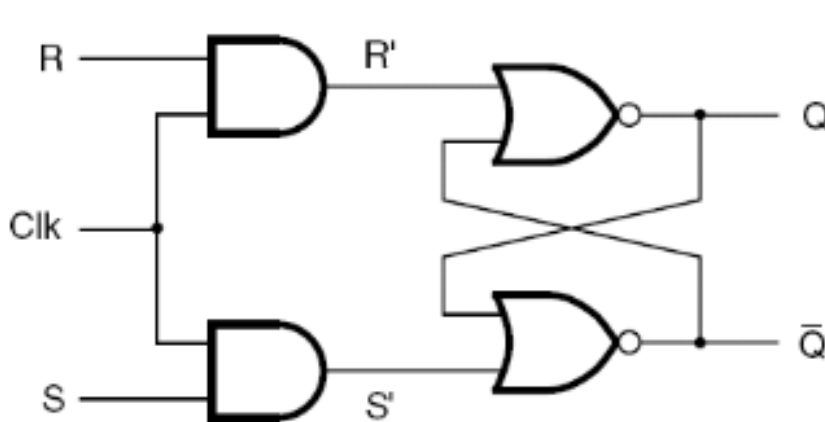
- Le **reti sequenziali asincrone** reagiscono immediatamente alle variazioni dell'ingresso.
- Spesso è necessario che lo stato di un flip-flop possa cambiare in determinati istanti temporali.
- Le **reti sequenziali sincrone** variano il loro stato solo in determinati istanti.
- Questi istanti sono determinati da un particolare segnale chiamato **clock** che varia il suo valore da 0 a 1 periodicamente.
 - La frequenza di clock indica quante volte il clock compie un ciclo in un secondo.
 - Si usano gli Hertz (Hz) come misura della frequenza di clock
 - Ad esempio: 1 MHz = un ciclo di clock ogni μs , ossia in un secondo il clock compie un milione di cicli.



Flip-flop sincroni

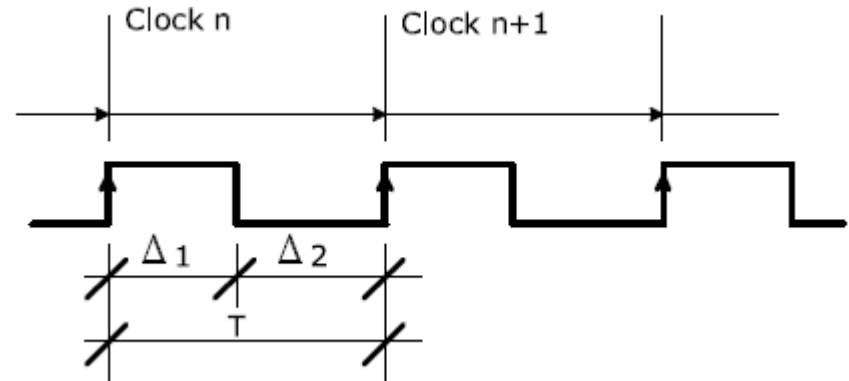
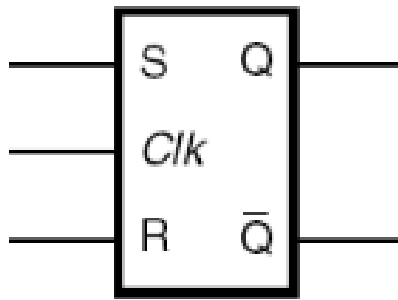
Latch NOR (flip-flop SR) sincrono

- La rete sequenziale in basso è chiamata **latch di NOR sincrono** oppure **flip-flop sincrono SR**.
 - Due ingressi Set (S) e Reset (R)
 - Un ingresso di sincronizzazione (Clock)
 - Due uscite Q e il suo complemento \bar{Q}
- Quando il clock vale 0, gli ingressi S e R non hanno alcun effetto (latch SR opaco), di conseguenza lo stato non cambia.
- Quando il clock vale 1, gli ingressi S e R sono efficaci (latch SR trasparente) e il latch si comporta come se fosse asincrono.
 - Flip-flop level triggered**, l'output può cambiare per tutto il tempo in cui il clock è a 1

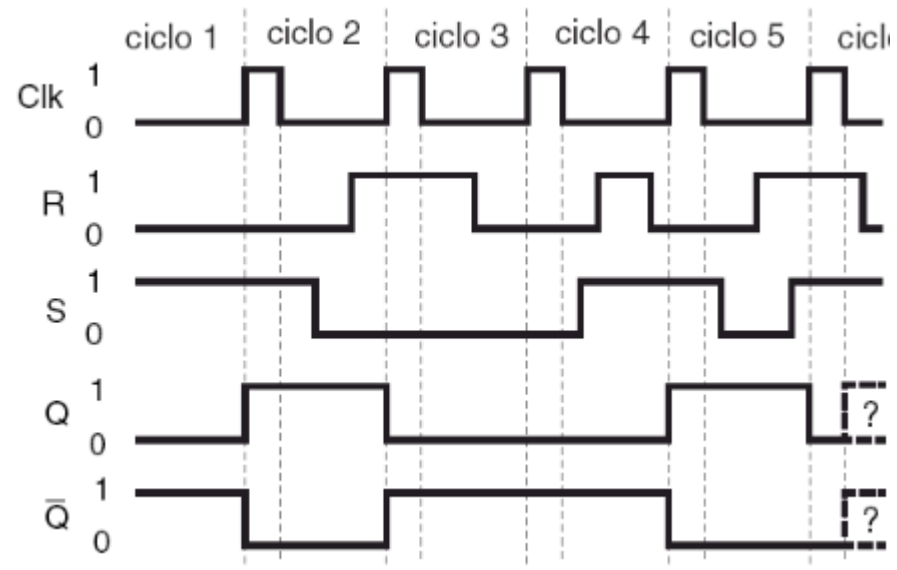


Clk	S	R	Q_{i+1}	stato futuro
0	x	x	Q_i	(non cambia)
1	0	0	Q_i	(non cambia)
1	0	1	0	stato corrente
1	1	0	1	
1	1	1	x	

Flip-flop SR (FFSR) sincrono level triggered



- Se T è il periodo di clock e sia Δ_1 il tempo in cui il clock è a 1 e Δ_2 il tempo in cui il clock è a 0. Si suppone che:
 - Durante Δ_1 S e R non cambino
 - Δ_1 sia sufficiente per stabilizzare l'output del flip-flop
- $Q_{i+1} = S + \bar{R}Q_i$

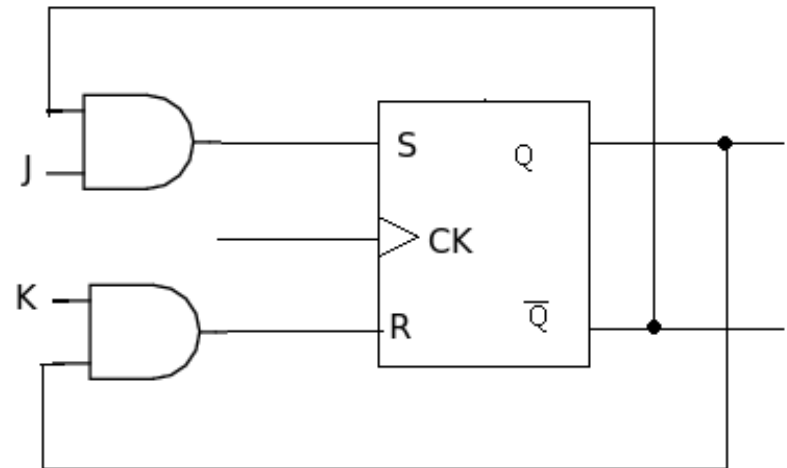


Flip-flop JK sincrono

- Simile al FFSR. Il comportamento è identico per gli ingressi 00, 01 e 10, mentre per l'ingresso 11 il flip-flop JK (FFJK) cambia sempre stato

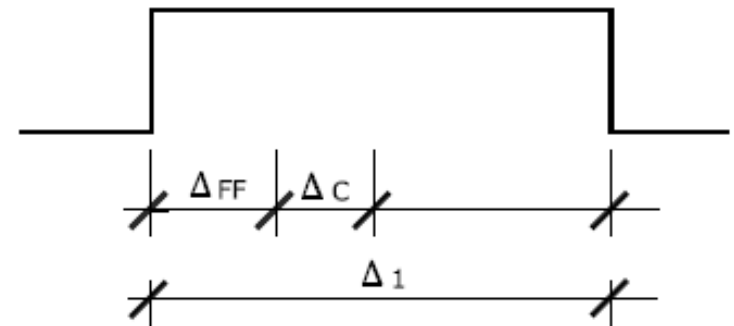
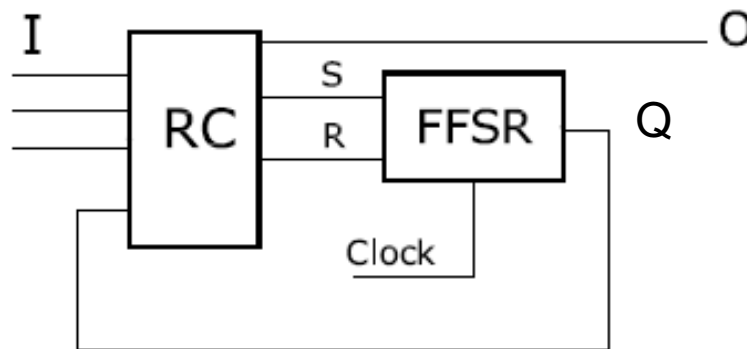
J	K	Q_{i+1}	\bar{Q}_{i+1}
0	0	Q_i	\bar{Q}_i
0	1	0	1
1	0	1	0
1	1	\bar{Q}_i	Q_i

$$Q_{i+1} = J\bar{Q}_i + KQ_i$$



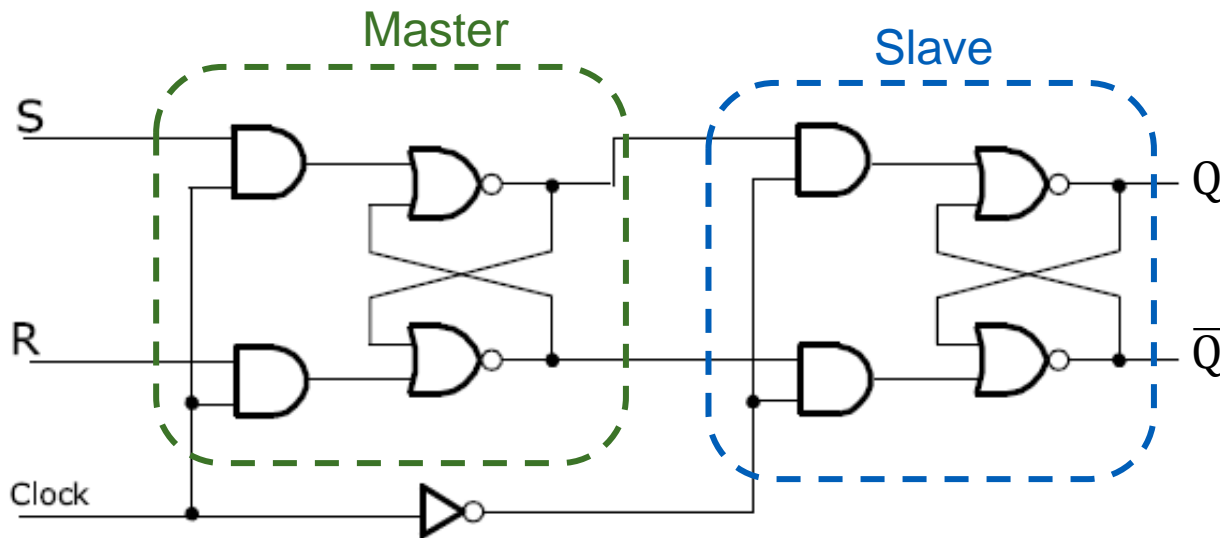
Problemi con flip-flop SR sincrono level triggered

- Supponiamo di avere un rete come quella in figura e siano:
 - Δ_C il tempo di commutazione della rete combinatoria (RC)
 - Δ_{FF} il tempo di commutazione del FFSR
- I è stabile per tutto il tempo Δ_1 (tempo in cui il clock è a 1)
- Trascorso il tempo Δ_{FF} dal fronte di salita del clock, il nuovo Q commuta
 - Gli ingressi di RC sono cambiati (Q è un ingresso di RC) e quindi potrei ottenere dei nuovi valori per R e S (R e S sono degli output di RC), che a loro volta potrebbero cambiare il valore di Q, e così via...



FFSR Master-slave

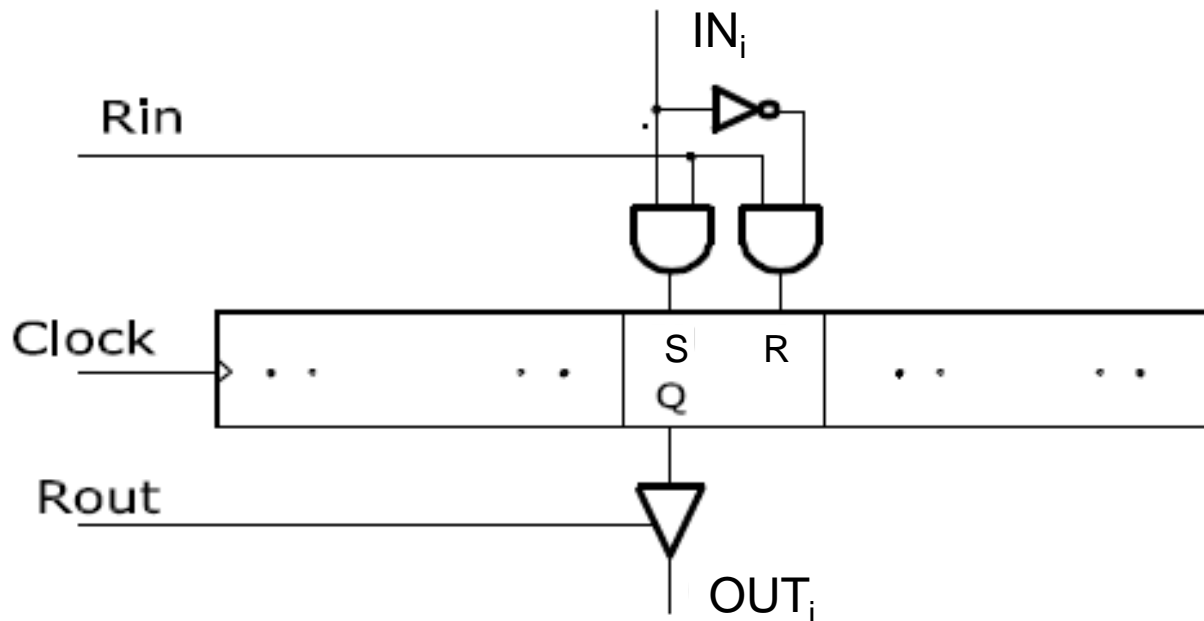
- Quando il clock è a 1: il master può cambiare stato, mentre lo slave resta invariato perché ha tutti gli ingressi a 0.
- Quando il clock passa a 0: il master non può commutare e lo slave si porta nello stato raggiunto dal master.
 - **Edge triggered FFSR**
- Se Δ_1 è il tempo in cui il clock è a 1, allora Δ_1 deve essere lungo almeno quanto il tempo massimo di commutazione del master.



Reti sequenziali sincrone basate sui flip-flop

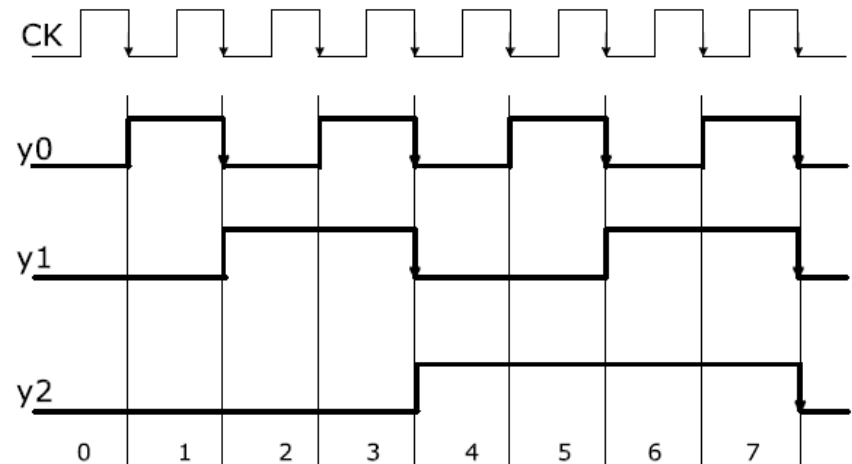
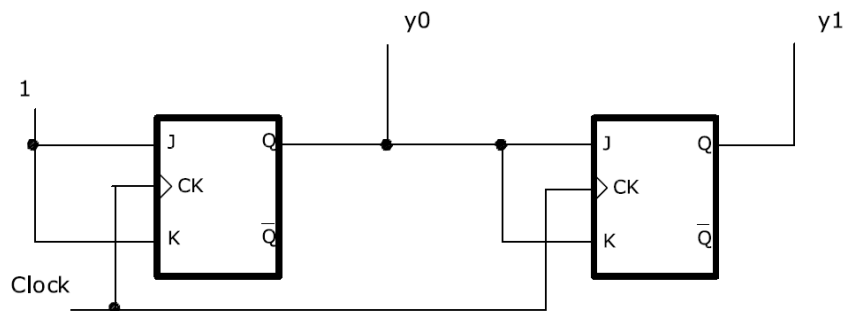
Registro a n bit

- Per registro si intende un insieme di n identici flip-flop sincronizzati tramite un unico clock.
 - Posso avere registri diversi in base al tipo di flip-flop utilizzati
- Rin può essere visto come una linea di comando per attivare un registro
- Rout ha funzione di Output Enable.



Contatore

- Se T è la durata di un ciclo di clock
- Ad ogni fronte di discesa del clock $\rightarrow y_0$ commuta
 - y_0 ha un periodo di $2T$
- Ad ogni fronte di discesa di $y_0 \rightarrow y_1$ commuta
 - y_1 ha un periodo di $4T$
- ...



Domande?

Riferimenti principali

- Appendice A di ***Calcolatori elettronici. Architettura e Organizzazione***, Giacomo Bucci. McGraw-Hill Education, 2017.
<http://highered.mheducation.com/sites/dl/free/8838675465/1098336/AppA.pdf> (download gratuito)