

Introduzione Circuiti Sequenziali

Riassunto teorico:

“I circuiti sequenziali sono caratterizzati dal fatto che, in un determinato istante, le uscite non dipendono solo dai livelli logici presenti sugli ingressi nello stesso istante ma anche dai livelli logici assunti dalle uscite negli istanti precedenti. Il concetto può essere meglio precisato introducendo la variabile tempo. Se ad un certo istante T_n gli ingressi e le uscite si trovano in un determinato stato, una successiva variazione dei valori logici di ingresso individua un nuovo istante T_{n+1} . In un sistema sequenziale il nuovo stato delle uscite dipende sia dallo stato degli ingressi all’istante T_{n+1} sia da quello delle uscite stesse all’istante precedente T_n . Ciò significa che il circuito è in grado di conservare memoria della sequenza degli stati assunti dagli ingressi e dalle uscite

Alcune strutture circuitali che realizzano questa funzione di memoria sono varie, ma contengono tutte un collegamento di reazione (feedback) che riporta in ingresso lo stato delle uscite. Anche le modalità con cui le variazioni degli ingressi vengono recepite dai circuiti sono varie: in alcuni (latch) la semplice variazione dello stato degli ingressi determina un nuovo stato delle uscite; in altre strutture (latch con abilitazione) il circuito avverte la variazione degli ingressi solo se è attivato un ingresso supplementare di abilitazione. In molti altri casi (flip flop) è invece presente un segnale di comando o di cadenza, chiamato clock, che controlla la risposta del circuito e definisce esattamente la temporizzazione delle commutazioni delle uscite.”

Esercizi di riepilogo

Per *rinfrancare la memoria* vi proponiamo di rifare i circuiti visti a lezione.

LATCH SR

Il circuito LATCH SR presenta due terminali di ingresso S e R, iniziali rispettivamente di Set (impostare) e Reset (azzerare) e due terminali di uscita Q e \bar{Q} (ricordatevi che, normalmente, le uscite sono complementate in questo tipo di circuiti). Ricordando ciò che avete visto a lezione compilate la tabella sottostante indicando la funzione di ogni combinazione di ingresso e successivamente disegnare su logisim il circuito LATCH SR utilizzando due porte logiche NOR, collegando degli opportuni pin di input ed output verificare il comportamento.

S	R	Q_{n+1}	\bar{Q}_{n+1}	Funzione
0	0	—	—	—
0	1	—	—	—
1	0	—	—	—
1	1	—	—	Non usato

Sapreste rifare lo stesso circuito con porte NAND? E' possibile? Come cambierebbe il circuito e la tabella delle relazioni tra ingresso ed uscita?

Nozioni extra: i latch rivestono una grande importanza nell'elettronica digitale in quanto costituiscono l'elemento base di tutti i circuiti sequenziali e trovano applicazione perfino come elementi singoli (es. circuiti antirimbazzo).

Alcuni video che possono essere utili, ne trovate a migliaia online!

<https://www.youtube.com/watch?v=nSEZN7DV7zY>

<https://www.youtube.com/watch?v=-aQH0ybMd3U>

LATCH SR gated e LATCH D

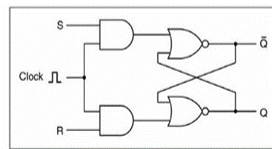
I latch SR possono essere “dotati” di un ulteriore ingresso, detto “di abilitazione”, o come viene presentato sulle slide del corso di un segnale di “sincronizzazione”. Implementate questo LATCH SR con in logisim e completate nuovamente la tavola di verità. In particolare, i simboli X nella prima riga esprimono una condizione di indifferenza.

Notate che nella slide è indicato un “clock” ovvero un segnale che con cadenza costante cambia il proprio valore assumendo i valori 0 ed 1. Il concetto di “clock” o “segnale di enable” in questo caso è indifferente, la modifica ottenuta con questo circuito consiste nel dare alla circuiteria un ulteriore funzionalità, ovvero la possibilità di essere “attivato”!



Latch SR Sincronizzato

- Circuito a porte logiche di un SR



ENABLE	S	R	Qn+1	Funzione
0	X	X	_____	Memoria
1	0	0	_____	_____
1	0	1	_____	_____
1	1	0	_____	_____
1	1	1	_____	_____

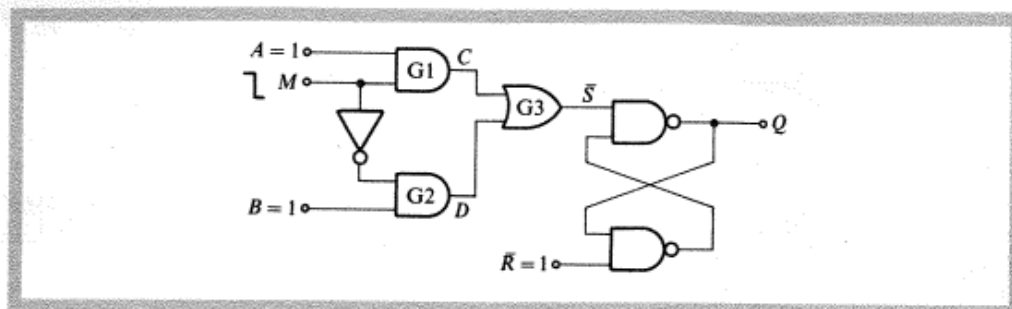
Fino ad ora abbiamo visto i latch SR con due input *Set* e *Reset*. Possiamo pensare di usare un singolo input in quale modo? Come andrebbe a modificarsi il circuito indicato nella slide precedente?

Per capire di più: vantaggi e limiti dei latch con abilitazione. A che servono? Esempio pratico.

5.2.1 Vantaggi e limiti dei latch con abilitazione

Nella memorizzazione dei dati, l'impiego di latch con ingresso di abilitazione rappresenta un notevole miglioramento rispetto a soluzioni con latch statici di tipo *SR* poiché consente di attuare la memorizzazione solo quando i dati di ingresso sono stabili e validi. Ciò è particolarmente evidente se si considera che, generalmente, i dati sono forniti ai latch da altri elementi circuitati, i cui tempi di risposta non sono nulli e non sono uguali fra di loro.

Fig. 5.6
Circuito logico
che genera
un'alea.



Si consideri ad esempio la situazione illustrata in fig. 5.6. Il circuito costituito dalle porte G1, G2, G3 comanda un latch $\bar{S}\bar{R}$ a porte NAND, con ingressi attivi bassi, che è stato precedentemente azzerato ($Q=0$). Con i valori indicati per A , B e \bar{R} , si vuole che la commutazione di M da 1 a 0 non sposti l'uscita Q dallo stato $Q=0$. Inizialmente sia $M=1$; essendo $A=B=1$, si ha $C=1$ e $D=0$ cosicché $\bar{S}=1$; il latch si trova in stato di memoria ($\bar{S}=\bar{R}=1$). Quando il segnale applicato ad M passa a livello 0, le uscite di G1 e G2 dovrebbero diventare rispettivamente $C=0$ e $D=1$ e quindi \bar{S} dovrebbe rimanere ad 1 mantenendo il latch in stato di memoria con $Q=0$, come richiesto. In realtà però la porta G1 commuta prima della porta G2 poiché la variazione di livello di M viene avvertita da G2 con un lieve ritardo (pari al tempo di propagazione t_{pd} della NOT) rispetto a G1. Commutando prima G1, si viene ad avere $C=0$ mentre D è ancora $D=0$: per un breve istante l'uscita di G3 viene ad essere $\bar{S}=0$ causando il set del latch ($Q=1$). Non appena G2 compie la sua commutazione, \bar{S} ritorna ad 1 bloccando il latch nello stato, indesiderato ed errato, $Q=1$.

L'inconveniente illustrato rientra nella vasta casistica delle *alee* ossia di quelle commutazioni indesiderate dovute non solo ai diversi percorsi di propagazione dei segnali, come nell'esempio precedente, ma anche ai diversi tempi di commutazione di porte, pure se dello stesso tipo. Gli impulsi spuri che nascono sono in qualche caso eliminabili, ma più spesso sono difficilmente prevedibili; si noti poi che se nei circuiti combinatori essi sono indesiderati, ma in certa misura tollerabili, in quelli sequenziali sono assolutamente dannosi in quanto possono causare, come nell'esempio visto, la memorizzazione di uno stato non valido e quindi una situazione di errore permanente. Ritornando all'esempio di fig. 5.6, l'inconveniente potrebbe essere evitato usando un latch di tipo gated e attivando l'ingresso di abilitazione con un certo ritardo rispetto alle commutazioni di M , in modo da consentire una completa stabilizzazione del dato su \bar{S} .

In genere, per risolvere il problema di una corretta temporizzazione dei diversi dispositivi, e in particolare degli elementi di memoria, che compongono un sistema digitale, si ricorre ad un *segnale di sincronismo comune*, che rappresenta il riferimento temporale per tutte le commutazioni e in sostanza scandisce i passi del processo di elaborazione. Questo segnale, costituito da un treno di impulsi periodici, è chiamato, proprio per la sua funzione, segnale di *clock* (orologio) o di *cadenza*.

L'uso di latch abilitati dal livello alto o basso di un segnale di abilitazione EN , anche se sincronizzato e attentamente correlato al clock comune del sistema, non sempre è sufficiente a risolvere tutti i problemi.

Il circuito di fig. 5.7a, ad esempio, presenta un funzionamento non accettabile. Partendo da una condizione iniziale che vede tutte le uscite $Q_i=0$, si vuole che al primo impulso di clock commuti il primo latch ($Q_1=1$), al secondo ciclo commuti il secondo latch ($Q_2=1$) e così via, come illustra il diagramma temporale di fig. 5.7b. In altri termini il dato ($D=1$) deve scorrere lungo la catena di latch avanzando di una posizione per ogni impulso di clock. Si noti che la funzionalità richiesta, così come la struttura proposta, sono tipiche dei registri a scorrimento, di cui si parlerà diffusamente nel seguito del capitolo. I latch di tipo D sono abilitati dal livello alto del segnale EN coincidente con il clock (Ck). All'istante $t=t_0$ si ha $D=1$, $Q_i=0$, $Ck=0$. Nell'istante $t=t_1$, Ck raggiunge il livello 1 abilitando tutti i latch e in

particolare il primo, la cui uscita diventa $Q_1 = 1$. In realtà la commutazione di Q_1 si verifica con un lieve ritardo, pari al tempo di propagazione t_{pd} del latch. Quando però $Q_1 = 1$, se il clock è ancora alto, anche il secondo latch, con un ritardo pari a t_{pd} , commuta nello stato $Q_2 = 1$. Se a questo punto il clock è ancora alto, si verifica la commutazione del terzo latch e di eventuali altri che siano inseriti nella catena. I diagrammi di fig. 5.7c illustrano la situazione reale nel caso che il clock rimanga ad 1 per un tempo molto superiore a t_{pd} . Come si vede, il comportamento del sistema è del tutto diverso da quello desiderato. Inoltre, supponendo di avere N latch in cascata, non si può ragionevolmente prevedere a quale latch si fermerà lo scorrimento del dato durante un ciclo di clock; questo infatti dipende dal rapporto fra il tempo durante il quale il clock rimane alto e il tempo di propagazione t_{pd} , che per di più non è uguale per tutti i latch.

Una soluzione sembra essere quella di abilitare i latch per un tempo brevissimo, inferiore al minimo tempo di propagazione t_{pd} , utilizzando come segnale di clock un treno di impulsi molto stretti ovvero con duty cycle ridottissimo. In questo modo il segnale Ck ritorna a 0 prima che la risposta del primo latch si manifesti sull'uscita Q_1 e quindi il secondo latch commuta solo al ciclo di clock successivo. Nascono però degli inconvenienti pratici che rendono questo metodo inaccettabile: impulsi tanto brevi non sono generalmente in grado di garantire corrette commutazioni dei latch; inoltre la trasmissione nel sistema digitale di impulsi così rapidi è difficilmente realizzabile e poco affidabile.

La soluzione corretta a questo tipo di problemi è pertanto quella di impiegare, invece dei latch, dispositivi le cui uscite commutino soltanto quando gli ingressi sono già disabilitati. Questa caratteristica è tipica di altri circuiti bistabili chiamati propriamente *flip-flop*.

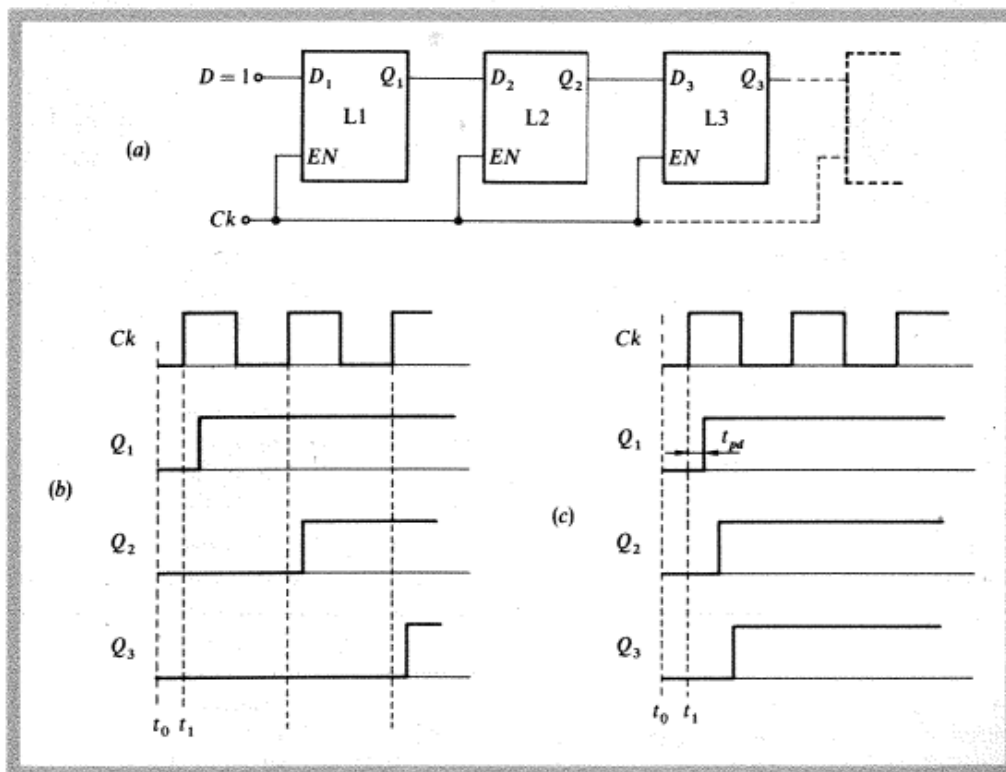
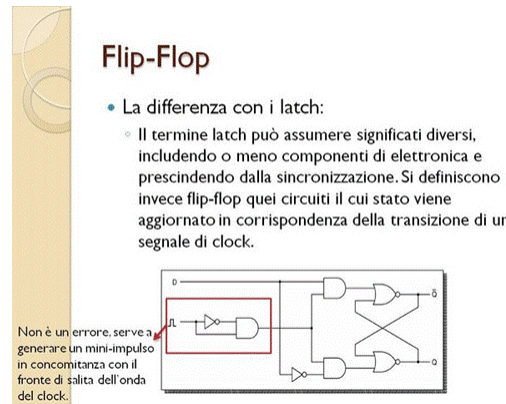


Fig. 5.7

(a) Latch disposti in cascata abilitati da un clock comune.
(b) Forme d'onda richieste.
(c) Forme d'onda reali.

4Flip Flop

I flip flop sono dispositivi bistabili provvisti di ingressi dati e di un ingresso supplementare a cui viene applicato un segnale di sincronizzazione, o clock! Essi sono caratterizzati dal fatto che la commutazione delle uscite, benché determinata dallo stato degli ingressi, si verifica solo in sincronismo con una ben precisa transizione del segnale di clock.



Provate a simulare la circuiteria di generazione di impulso seguendo le note della guida di logisim

<http://www.cburch.com/logisim/docs/2.6.0/en/guide/prop/delays.html>

Esercizi e domande:

1. Schematizzate i Flip Flop SR e successivamente il Flip Flop di tipo D, insieme con le tabelle funzionali.
2. Che differenza c'è tra un FLIP FLOP SR ed un FLIP FLOP JK a livello funzionale?