

Volfango Furgani

CIRCUITI SEQUENZIALI

23 August 2002

global \$id,\$section; ?>

Circuiti sequenziali:

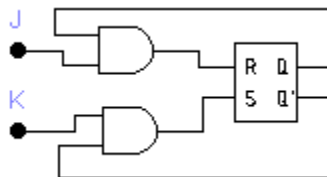
- [Funzionamento sincrono dei circuiti sequenziali](#)
- [Il segnale di clock](#)
- [La corretta durata del livello alto del clock](#)

Funzionamento sincrono dei circuiti sequenziali

(Dom. 104) Abbiamo visto il latch SR realizzato con soli nor definito dalla tabella di verità:

S R	$Q_n + 1$	$n+1$ indica l'istante temporale (non piove, tranquilli) successivo a Q
0 0	Q_n	condizione di memoria. La frase: "l'uscita rimane com'era" è rivelatrice della natura sequenziale del problema
0 1	0	condizione di reset od azzeramento o di scrittura di zero
1 0	1	condizione di set o posizionamento ad 1 o di scrittura di uno
1 1	x	condizione proibita perché risulta $Q = Q'$ (situazione imbarazzante)

Supponiamo di volere eliminare la condizione proibita e di sostituirla con la condizione di commutazione. Il circuito seguente sembra adatto allo scopo: infatti R ed S non possono valere contemporaneamente 1 perché o Q o Q' vale 0.



Quindi la condizione proibita è certamente evitata. Si può verificare (esercizio) che per $J = K = 1$ il latch cambia stato rispondendo quindi alla tabella:

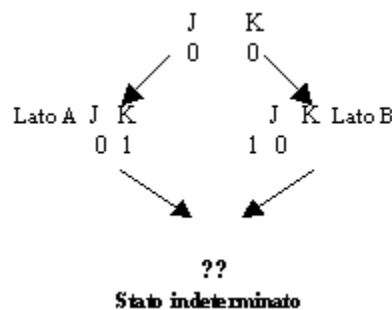
J K	$Q_n + 1$
0 0	Q_n
0 1	0
1 0	1
1 1	Q_n commutazione

Commutare significa cambiare stato, cioè passare da 0 ad 1 o da 1 a 0.

Questo circuito, il molto ipotetico latch J K, non si sogna di funzionare per due maledettissimi motivi.

Il segnale di clock

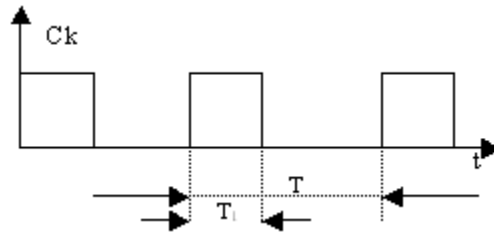
Questa è una rete asincrona soggetta al seguente inconveniente. Non è affatto probabile che J e K cambino valore contemporaneamente, cioè entro il tempo t_{pd} (ad esempio di 10 nanosecondi). Questo significa che, se J e K debbono cambiare stato "contemporaneamente" (ad esempio da 00 ad 11), è molto probabile che uno dei due modifichi il suo valore prima dell'altro dando luogo alle due possibilità transitorie di figura:



Nel caso che K vada ad 1 prima di J seguendo il Lato A, Q si azzerava. Nel caso contrario ($J = 1$ e $K = 0$ seguendo il Lato B) Q si porta a 1. (Vedi tabella.) In sostanza si sa da dove si parte ,non si sa dove si arriva. Non è bello. Questo comportamento si chiama **corsa critica** e, per evitarlo ci sono due possibilità

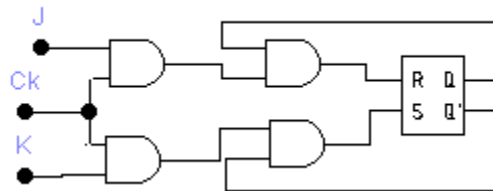
- 1) si impone che vari un solo segnale per volta (funzionamento asincrono) o
- 2) si usa una sbarra metaforica (segnale di clock) che blocca J e K finchè raggiungono il loro stato finale definitivo (non vuol dire che sono morti). Poi la

sbarra si alza ed il circuito legge i segnali d'ingresso e si comporta correttamente di conseguenza. Il segnale di clock è rappresentato in figura.



Il livello basso corrisponde alla sbarra abbassata in cui i segnali cambiano senza influenzare il sistema. Durante il livello alto i valori di J e K vengono letti e possono influenzare il flip-flop. Tale segnale è caratterizzato dal periodo T, dalla frequenza $f = 1/T$ e dal duty-cycle $d\% = (T_1/T) \cdot 100$. Se $d\% = 50\%$ l'onda si dice quadra.

Il clock può essere inserito nel precedente circuito semplicemente per mezzo di due and:

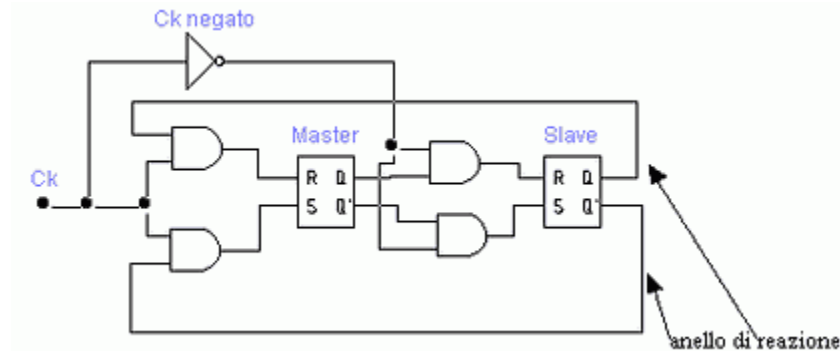


anello di reazione tipico dei circuiti sequenziali

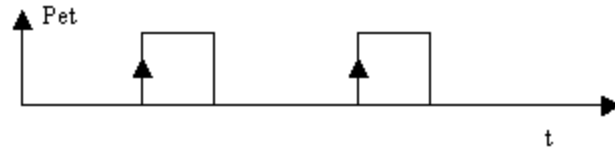
La corretta durata del livello alto del clock ([Dom. 105](#))

Il problema delle corse critiche (possibili nei circuiti sequenziali asincroni quando cambia simultaneamente più di un segnale di ingresso per volta) è così risolto. Se ne intravede un secondo all'orizzonte. Vediamolo con un esempio. Supponiamo che il tempo di commutazione T_c del ff J-K sia di 100 ns. Supponiamo inoltre che T_1 duri 400 ns e che sia $J = K = 1$. Il ff ha tempo di commutare 4 volte. Commutare un numero pari di volte significa balbettare per un po' ed alla fine non commutare. Le commutazioni a catena sono possibili solo per l'anello di reazione sempre chiuso che collega uscita e l'ingresso siamesi nei circuiti sequenziali. Riassumo: in sostanza con il comando $J = K = 1$ il ff JK può commutare un numero indefinito di volte (pari o dispari) in funzione del rapporto T_1/T_c . Due sono le vie per il Regno: **a)** rendere in ogni luogo ed in ogni tempo $T_1 = T_c$ (possibile solo in linea di principio) per dar tempo al ff di commutare una sola volta **b)** trovare la maniera di aprire l'anello in modo che l'ingresso non sappia ciò che l'uscita ha combinato. Ricordate il precetto: non sappia la destra ciò che fa la sinistra? (A proposito che sta facendo la tua destra, caro lettore? Meglio, molto meglio che la sinistra non sappia! O

sei mancino?) Il cambiamento di stato avviene una sola volta quando l'anello aperto si richiude. Il tipo di ff che così otteniamo si chiama master-slave. (Io master, tu slave)



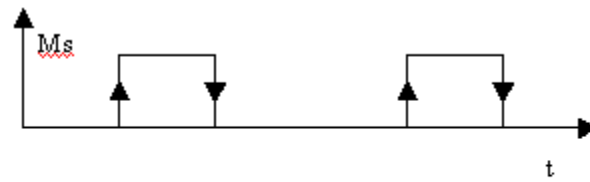
E' costituito da due ff RS che però non sono mai abilitati contemporaneamente dal segnale di clock a causa del not. In questo modo il ff master legge gli ingressi per tutto il tempo T1 senza mai trasmettere alcuna informazione alla slave che non è abilitata (anello aperto). Al termine di T1 il clock va a 0 e quindi, a causa del not, si abilita lo slave che legge le informazioni del master e cambia di conseguenza le uscite dello slave senza tuttavia che ciò influisca sul primo ff che non è abilitato (anello aperto). In questo modo si ha una sola commutazione. Complicato, ma non semplice. Questo circuito continua a presentare un punto debole. Se tenete la porta aperta per un tempo lungo, può entrare di tutto: rumore, spazzatura, disturbi, Casanova. Fuor di metafora gli and che leggono i segnali di ingresso sono attivi per tutto il tempo T1 durante il quale possono raccogliere disturbi elettrici e false letture. Il problema si risolve abilitando gli and non per tutto T1 o, come si dice, sul livello, ma riducendo il tempo di lettura ad un flash scattato sul **fronte del clock**. Riassumendo e completando: noi lavoreremo con tre tipi costruttivi di ff: il positive edge-triggered o pet, il negative edge-triggered o net ed il data lock-out che è un master-slave abilitato solo sui fronti. Chiameremo fronte di lettura quello in cui vengono letti gli ingressi e di scrittura quello in cui le uscite cambiano.



Brevemente diremo che il ff pet legge e scrive sul fronte di salita



Brevemente diremo che il ff net legge e scrive sul fronte di discesa



Brevemente diremo che il ff ms legge sul fronte di salita
e scrive sul fronte di discesa