⌛

Tempi di clock, di setup, di hold e frequenza massima

▼ Creatore originale: @Gianbattista Busonera 😉

- @Gianbattista Busonera(12/04/2025): Trattazione tramite video e testo di un argomento filosofico del corso, utile in numerosi esercizi.
- @Giacomo Dandolo (13/04/2025): aggiunta di piccole descrizioni, miglioramento dell'evidenziatura del testo, in modo da catturare meglio le parti importanti.

👉 Il materiale di questa pagina è stato preso:

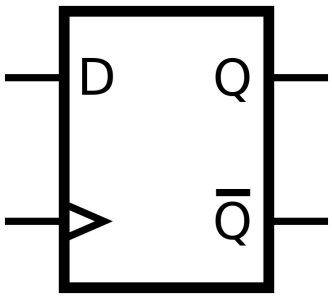
- dalle lezioni di ELAP 2018: LINK
- da video di indiani su Youtube, di cui allego il principale: LINK

https://youtu.be/rTB2H5gs9vA

Spiegazione di questo argomento in formato video (NB: il materiale mostrato in video potrebbe non essere 1:1 con ciò che è mostrato in questa pagina, ma i concetti sono equivalenti)

Tutte le porte logiche, quindi anche quelle che formano i flip-flop, introducono dei ritardi.

Prendiamo in considerazione un Flip Flop di tipo D che commuta sul fronte di salita del clock.



Visualizzazione del componente Flip-Flop di tipo D

In un mondo ideale, senza ritardi di alcun tipo, otterremmo questo risultato:

Questo perché il Flip Flop commuta sul fronte di salita del clock e, di conseguenza, mantiene il valore alto (1 logico) per tutto il tempo in cui D resta alto, senza alcun ritardo.

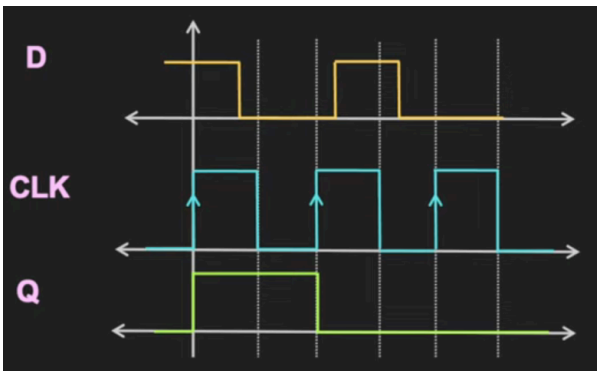
Fino ad ora stiamo assumendo che, al fronte di salita del clock (CLK), qualsiasi sia il valore di D (ingresso), quest'ultimo verrà campionato istantaneamente (in Q).

Ritardo di propagazione del clock

In un modello ideale, l'uscita Q del flip flop assume istantaneamente il valore

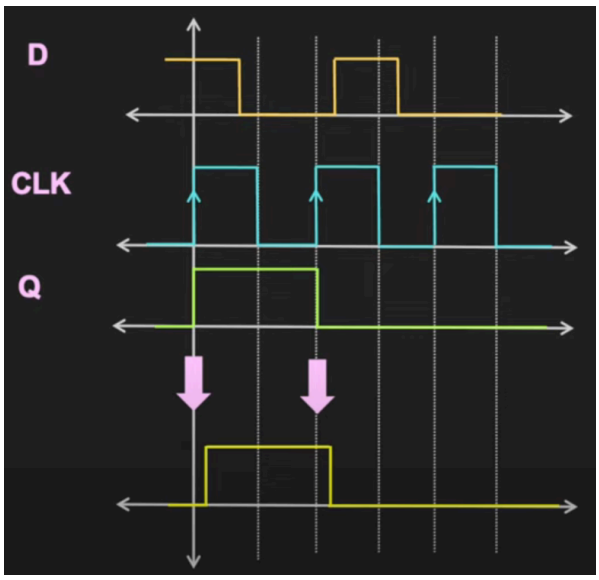
In un modello reale, l'uscita Q commuta con un certo ritardo, detto

di D, come visto precedentemente.



Uscita Q commuta istantaneamente (modello ideale)

ritardo di propagazione $t_{\text{CK-Q}}$.



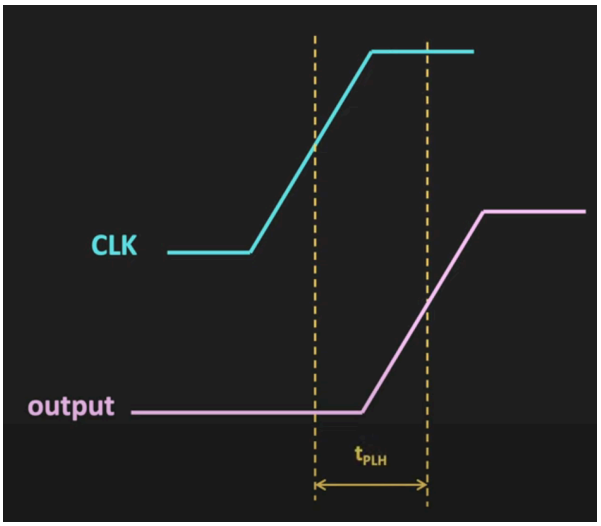
Uscita Q che commuta con un certo ritardo (reale)

Il ritardo di propagazione $t_{\text{CK-Q}}$ è dovuto al fatto che il clock non commuta istantaneamente da uno stato basso a uno alto, ma quest'ultimo deve passare una certa soglia V_T affinché si possa effettivamente considerare "commutato".

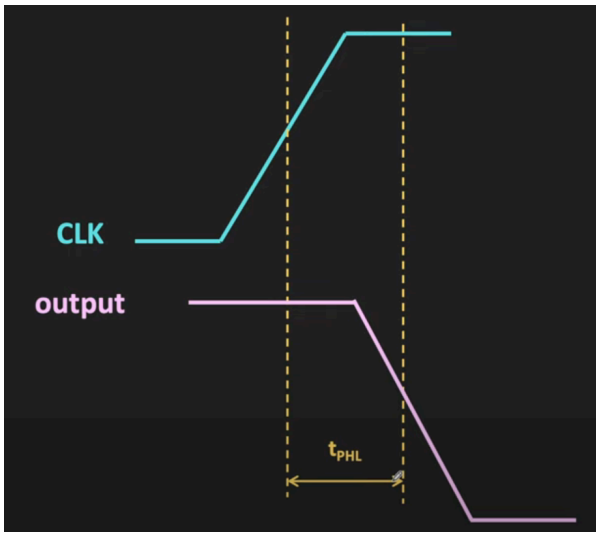
Ritardi di propagazione su transizione L→H e H→L

I tempi descritti in questa sezione sono quelli di propagazione sulla transizione L→H ($t_{\text{CK},L\to H}$) e sulla transizione H→L ($t_{\text{CK},H\to L}$), dove L indica lo stato basso (0), mentre H indica lo stato alto (1).

Come visibile in foto, l'output inizia a cambiare stato solo quando il clock raggiunge effettivamente lo stato alto. Inoltre, l'uscita non commuta istantaneamente al valore d'ingresso, ma è necessario un certo tempo transitorio, cioè un breve tempo necessario affinché anche l'output possa effettivamente commutare (in quanto anche lui ha un certo ritardo di propagazione da uno stato alto a basso o viceversa).



Visualizzazione della commutazione di output quando il valore di clock arriva allo stato alto (ideale)



Visualizzazione della commutazione di output quando il valore di clock arriva allo stato alto (reale)

Finestra di campionamento

In un modello reale, l'input dovrebbe rimanere stabile per una certa quantità di tempo, detto tempo di acquisizione o finestra di campionamento.

In particolare, tale finestra è suddivisa in:

- tempo di setup t_{su};
- tempo di hold t_{hold}.

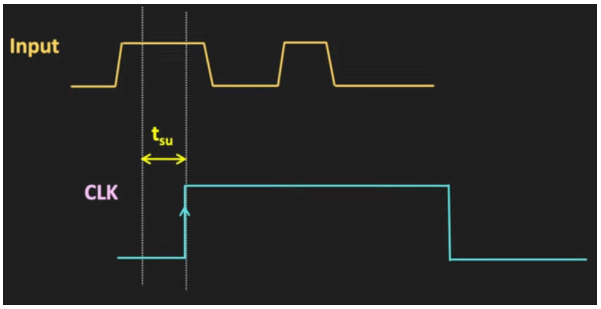
> ⚠️ La finestra di campionamento e, quindi, i tempi di setup t_{su} e di hold t_{hold} di cui è composta, riguarda solamente la quantità di tempo per cui l'input deve rimanere stabile.

Tempo di setup

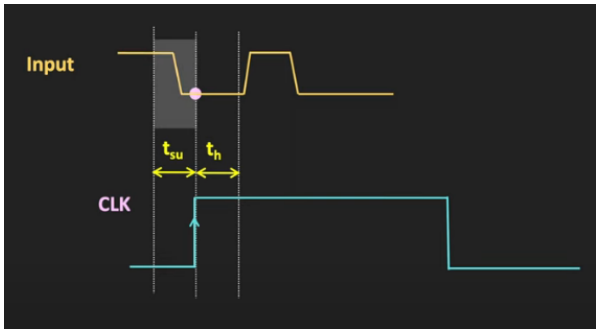
Il tempo di setup t_{su} è l'intervallo di tempo per cui l'ingresso D (Input) deve rimanere stabile (non può commutare) **prima** del fronte di salita del clock.

Sono presenti due casi:

1. nel primo diagramma temporale, si può vedere che l'input è stabile al valore alto per il tempo di setup. Di conseguenza, non sono presenti violazioni e il dato può essere campionato;
2. nel secondo diagramma temporale, si può vedere che l'input non è stabile per il tempo di setup. Di conseguenza, sono presenti violazioni e il dato risulta inconsistente con ciò che ci si aspetta.



Soddisfacimento del tempo di setup, in quanto l'input NON commuta durante il tempo di setup



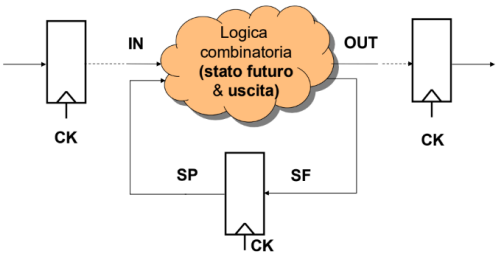
Violazione del tempo di setup, in quanto l'input commuta durante il tempo di setup

⚠ Si noti che il tempo di setup dipende unicamente dalle caratteristiche intrinseche del flip flop ed è un dato che ci verrà sempre fornito.

Calcolo del periodo di clock minimo (ritardi massimi)

Consideriamo il seguente modello di circuito, in cui il flip flop a sinistra (d'ora in poi chiamato FF1) deve trasferire il suo valore (IN, equivalente a Q1) nella logica combinatoria, la quale farà le dovute trasformazioni su tale dato.

Una volta processato, il dato (OUT) è pronto per essere immagazzinato nel flip flop a destra (FF2) come input.



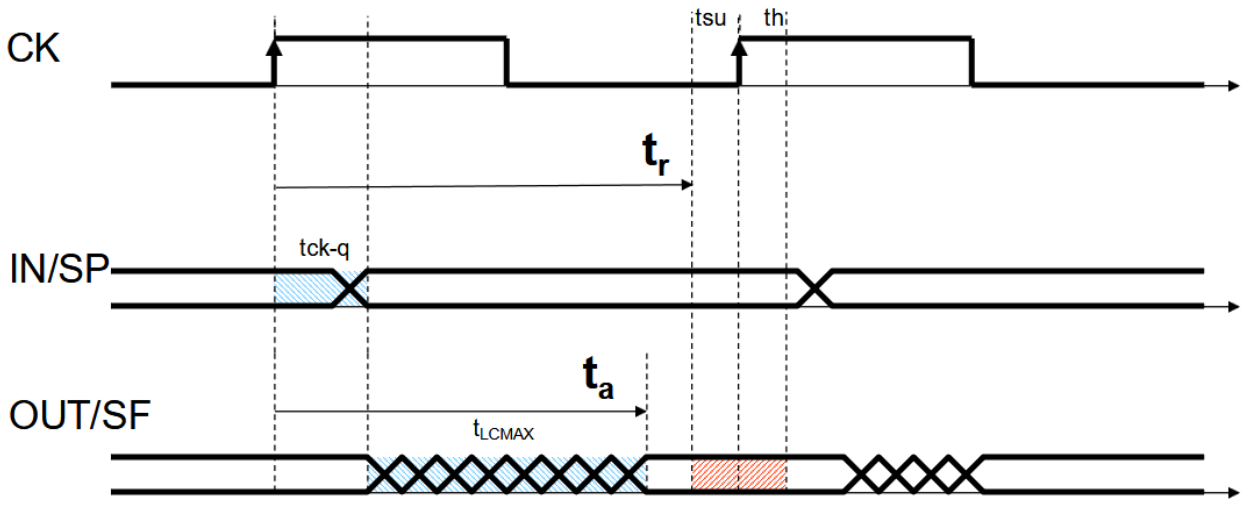
Circuito con 3 flip flop di tipo D e una logica combinatoria

Valutiamo un diagramma temporale di esempio che ci permette di vedere le variazioni su IN e OUT tra FF1 e FF2:

1. concentriamoci sull'input: esso non viene propagato istantaneamente sulla logica combinatoria, ma deve aspettare un tempo $t_{\text{CK-Q}}$.
2. per un certo periodo di tempo $t_{\text{LC,max}}$ (definito come il tempo massimo di elaborazione della logica combinatoria), il dato resta variabile prima di stabilizzarsi a un certo valore.

> ⚠️ Si prende il ritardo massimo della logica combinatoria in quanto vogliamo considerare il caso peggiore per il quale dobbiamo garantire il corretto funzionamento del circuito e, quindi, determinare un periodo di clock minimo.

E' importante che il dato si stabilizzi prima del tempo di setup del periodo di clock successivo (OUT è l'input D del FF2, e deve soddisfare i vincoli della finestra di campionamento).



Consideriamo il primo Flip Flop a sinistra come già "stabile"; il problema è l'input del flip flop a destra, il quale risente dei ritardi del primo flip flop ($t_{\text{CK-Q}}$) e dei ritardi della logica combinatoria ($t_{\text{LC,MAX}}$).

Il dato che parte dall'uscita del FF1 si propaga in ingresso nella logica combinatoria (come in via indiretta in ingresso del FF2) dopo un certo $t_{\text{arrivo}} = t_a$ pari a:

$$t_{\text{a}} = t_{\text{CK-Q}} + t_{\text{LC,max}}$$

Però, è necessario che tale dato sia memorizzato nel FF2 in un certo $t_{\text{richiesta}} = t_r$ pari a:

$$t_{\text{r}} = T_{\text{CK}} - t_{\text{su}}$$

> ⚠️ Il dato che parte dall'uscita del FF1, per essere memorizzato nel FF2, deve restare stabile per un tempo di setup prima del successivo colpo di clock. In tal senso, si parla di tempo minimo richiesto per evitare violazioni.

Per quanto detto in precedenza, consideriamo la seguente relazione:

$$t_a \leq t_r$$

Riportando i valori di t_a e t_r definiti in precedenza ed esplicitando il valore di T_{CK}, si ottiene:

$$t_{\text{CK-Q}} + t_{\text{LC,max}} \leq T_{\text{CK}} - t_{\text{su}} \Rightarrow T_{\text{CK}} \geq t_{\text{CK-Q}} + t_{\text{LC,max}} + t_{\text{su}}$$

Si capisce, dunque, che il periodo di clock minimo affinché il circuito funzioni senza violazioni di setup è pari a:

$$T_{\text{CK,min}} = t_{\text{CK-Q}} + t_{\text{LC,max}} + t_{\text{su}}$$

Calcolo della frequenza massima

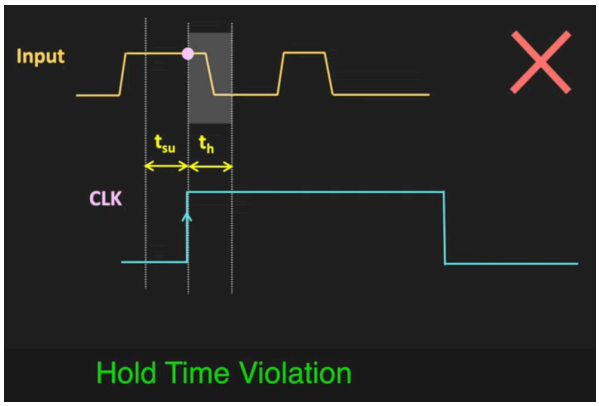
La frequenza massima per la quale un circuito può funzionare senza problemi è facilmente calcolabile come segue (ricordando che la frequenza è l'inverso del periodo):

$$F_{\max} = \frac{1}{T_{\text{CK,min}}} = \frac{1}{t_{\text{CK-Q}} + t_{\text{LC,max}} + t_{\text{su}}}$$

Tempo di hold

Il tempo di hold t_{hold} è l'intervallo di tempo per cui l'ingresso D (input) deve rimanere stabile dopo il fronte di salita del clock.

Nel diagramma temporale di esempio, si può notare che l'input non è stabile per il tempo di hold, in quanto passa da alto a basso. Di conseguenza, è presente una violazione di hold.



Violazione del tempo di hold, in quanto l'input commuta durante il tempo di hold

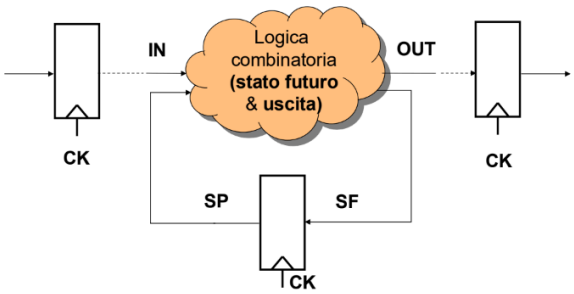
⚠️ Si noti che il tempo di hold è più subdolo del tempo di setup, in quanto non dipende unicamente dal flip flop, ma anche dal contesto esterno (ritardi di propagazione, logica combinatoria, e via dicendo)

Un'analogia interessante per questo concetto risulta essere nella macchina fotografica:

- per catturare una foto con tanta luce, il tempo è breve (contesto favorevole);
- per catturare una foto con poca luce, il tempo risulta lungo (contesto sfavorevole).

Calcolo del tempo di hold massimo (ritardi minimi)

Consideriamo il seguente modello di circuito, descritto già quando si è parlato del calcolo del periodo di clock minimo nella sezione del tempo di setup.



Circuito con 3 flip flop di tipo D e una logica combinatoria

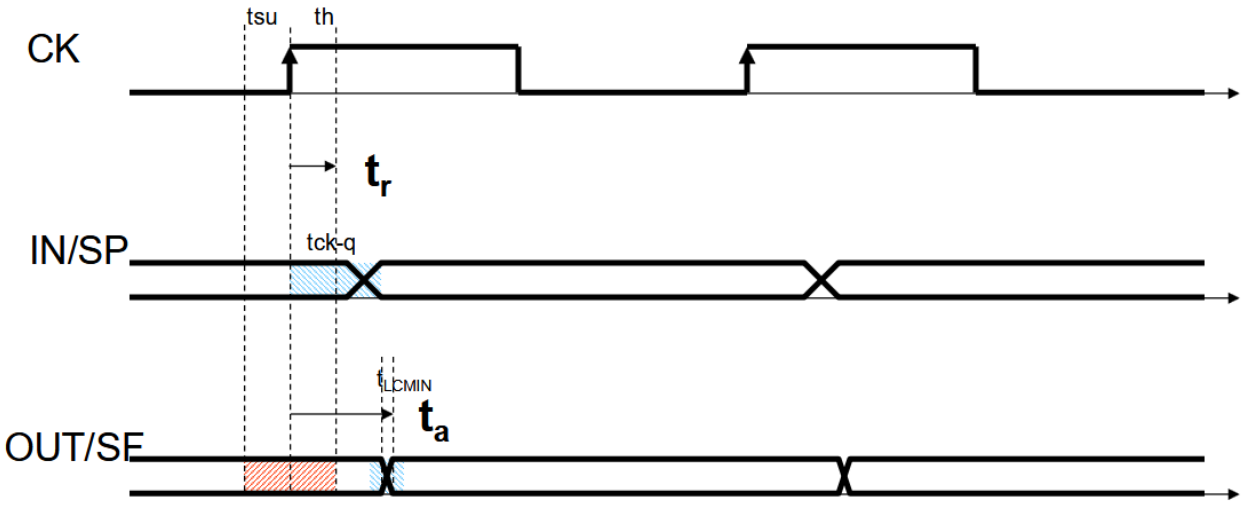
Valutiamo un diagramma temporale di esempio che ci permette di vedere le variazioni su IN e OUT tra FF1 e FF2:

1. concentriamoci sull'input: esso non viene propagato istantaneamente sulla logica combinatoria all'arrivo del fronte di salita del clock, ma deve aspettare un tempo $t_{\text{CK-Q}}$.
2. per un certo periodo di tempo $t_{\text{LC,MIN}}$ (definito come il tempo minimo di elaborazione della logica combinatoria), il dato resta variabile prima di stabilizzarsi a un certo valore.

> ⚠️ Si prende il ritardo minimo della logica combinatoria in quanto vogliamo considerare il caso peggiore per il quale dobbiamo garantire il corretto funzionamento del circuito e, quindi, determinare un tempo di hold massimo.

E' importante che il FF2 mantenga il dato precedente (area evidenziata in arancione) per un certo tempo di hold prima che assuma un nuovo valore. Il tempo di hold, infatti, ha come vincolo il periodo di clock attuale (e non il successivo, come nel caso del tempo di setup).

Se il nuovo dato (IN) arriva troppo in fretta (dati i tempi $t_{\text{CK-Q}}$ e $t_{\text{LC,min}}$ troppo brevi), rischiamo che OUT commuti durante il suo tempo di hold, commettendo una violazione di hold.



Consideriamo il primo Flip Flop a sinistra (FF1) come già “stabile”; il problema sta nel valore “precedente” del flip flop a destra, il quale deve rimanere stabile per almeno il tempo di hold del periodo di clock attuale

Il dato che parte dall’uscita del FF1 si propaga in ingresso della logica combinatoria (e quindi in via indiretta in ingresso del FF2) dopo un certo $t_{\text{arrivo}} = t_a$ pari a:

$$t_a = t_{\text{CK-Q}} + t_{\text{LC,min}}$$

Però è necessario che tale dato arrivi nel FF2 solo dopo che sia passato il tempo di hold per il suo precedente valore e, quindi, dopo un certo $t_{\text{richiesta}} = t_r$ pari a:

$$t_r = t_{\text{hold}} = t_{\text{h}}$$

> ⚠️ Il dato che parte dall’uscita del FF1, deve evitare di arrivare troppo in fretta all’ingresso del FF2, altrimenti violeremmo il tempo di hold del FF2 per il precedente valore (riferito allo stesso periodo di clock). In tal senso, si parla di tempo massimo di hold richiesto per evitare violazioni.

Per quanto detto in precedenza, consideriamo la seguente relazione:

$$t_a \geq t_r$$

Riportando i valori di t_a e t_r definiti in precedenza ed esplicitando il valore di t_{hold}, si ottiene:

$$t_{\text{CK-Q}} + t_{\text{LC,min}} \geq t_{\text{hold}} \Rightarrow t_{\text{hold}} \leq t_{\text{CK-Q}} + t_{\text{LC,min}}$$

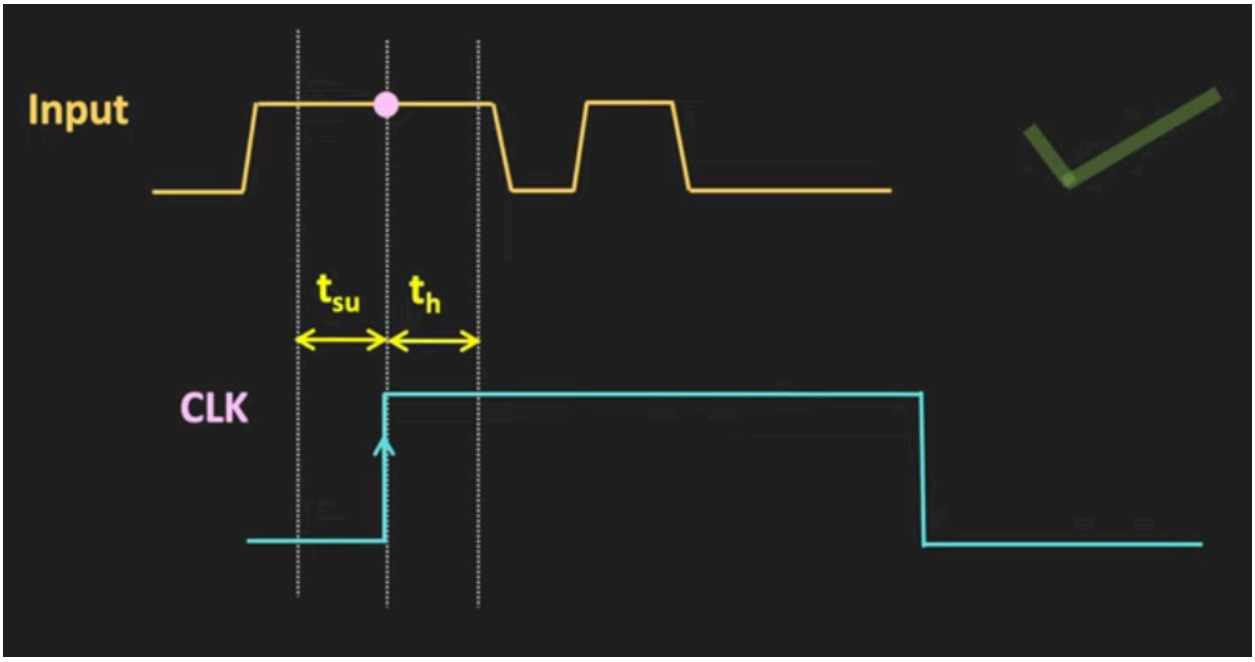
Si capisce, dunque, che il tempo di hold massimo affinché il circuito funzioni senza violazioni di hold è pari a:

$$t_{\text{hold,max}} = t_{\text{CK-Q}} + t_{\text{LC,min}}$$

⚠ In pratica, il problema è che un segnale può arrivare nel FF2 così rapidamente da modificare un suo ingresso che, invece, doveva rimanere stabile per un tempo di hold.

Esempio - Soddisfacimento delle condizioni di setup e hold

L'input descritto nel diagramma temporale soddisfa entrambe le condizioni di setup e di hold.



Esempio di soddisfacimento di tempi di setup e di hold

⚠️ Un’analogia interessante per descrivere la finestra temporale di acquisizione risulta essere nella macchina fotografica: immaginiamo di utilizzare una macchina fotografica e di voler scattare una foto al Prof. “C” (ovvero il nostro input D del FF).

Il fronte di salita del clock è dato dalla pressione del bottone per scattare la foto, e vogliamo che il soggetto della nostra foto (input D):

- resti fermo per un dato periodo di tempo prima dello scatto, così che non faccia smorfie poco prima che sia scattata e si possa scegliere l’istante giusto, definendo il tempo di setup;
- resti fermo un istante dopo lo scatto, cosicché la macchina fotografica abbia il tempo di catturare correttamente l’immagine, definendo il tempo di hold.