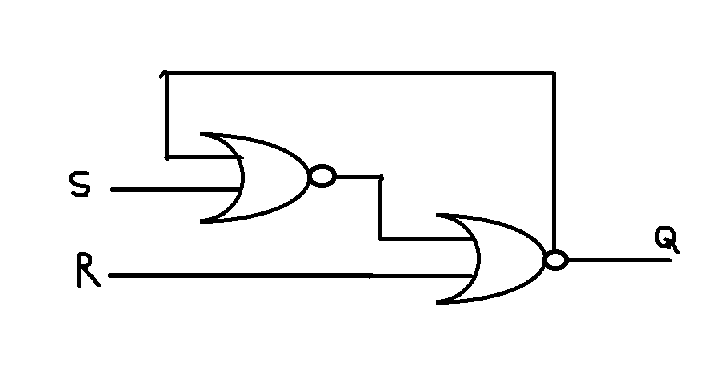
Circuiti sequenziali. In cosa si differenzia da un circuito combinatorio?

Poniamo di avere un circuito di questo tipo:



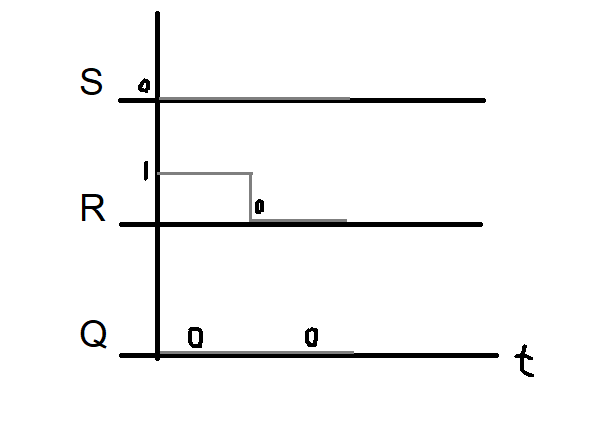
Cosa succede? Controlliamo la tavola di verità

|  |  |  |
| --- | --- | --- |
| S | R | Q |
| 0 | 0 | ? |
| 0 | 1 | 0 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

Se sia S che R sono 0 non si può determinare quale sia il risultato. Se l’uscita fosse 1 allora l’ingresso S nel secondo NOR varrebbe 0, e ciò varrebbe se l’ingresso Q nel NOR vicino ad S vale 1, e ciò ha senso. Ma se l’uscita fosse 0 allora i 2 0 in ingresso al NOR di S produrrebbero un 1, che produrrebbe uno 0 al NOR di R.

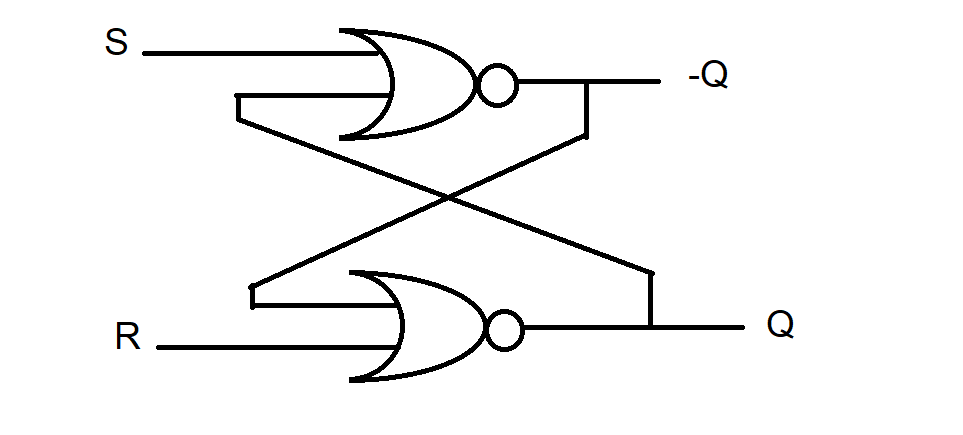
I valori 0 e 1 sono quindi entrambi possibili: la configurazione dei valori in ingresso non è sufficiente a determinare l’uscita. Questo dispositivo ha la capacità di ricordarsi che cosa è successo prima.

Questo discorso può essere riassunto in un grafico:

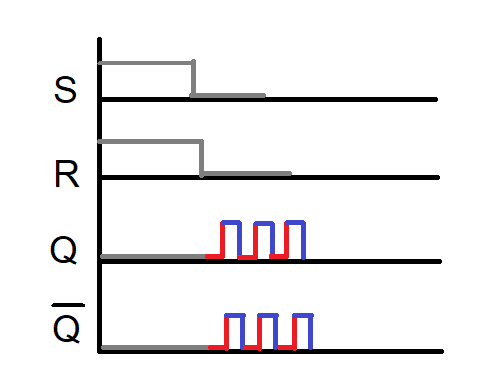


I circuiti sequenziali quindi si ricordano dei valori precedenti. Essi sono caratterizzati dal presentare dei circuiti chiusi e dal fatto che per conoscere il loro valore in uscita bisogna “tornare indietro nel tempo” e conoscere l’esatta sequenza iniziale dei valori inviati in ingresso.

Questo è un Flip-Flop di tipo S(set)-R(reset).



Se si guarda la tavola di verità, l’unico valore non ammissibile è 1,1. Poiché ciò produrrebbe 0 0 e Q non sarebbe più l’opposto di -Q. (anche se in realtà non si può stabilire una tavola di verità per un circuito sequenziale). Inoltre, se i valori di S e R fossero 1,1. Il grafico temporale sarebbe:



Ecco perché i valori 1,1 sono sbagliati. Dopo averli ricevuti se si volesse cambiare la rappresentazione di S e R a 0,0 il flip flop diventerebbe un clock che continua a inviare valori 1,1 e 0,0. Questo perché dopo un tempo Epsilon le due funzioni si aggiornerebbero contemporaneamente al valore 1,1, ma poiché sono dei NOR e ricevono in ingresso l’uno l’output dell’altro, dopo un altro tempo Epsilon entrambi si aggiornerebbero al valore 0,0, ma questo riprovocherebbe la situazione iniziale e si avrebbe un loop infinito. Ecco perché i valori 1, 1 sono “vietati”.

Questo non si verifica se si passa da 1,1 a 1,0 o 0,1. Ciononostante, il valore 1,1 non dovrebbe mai venire inviato all’ingresso di un flip-flop s-r.

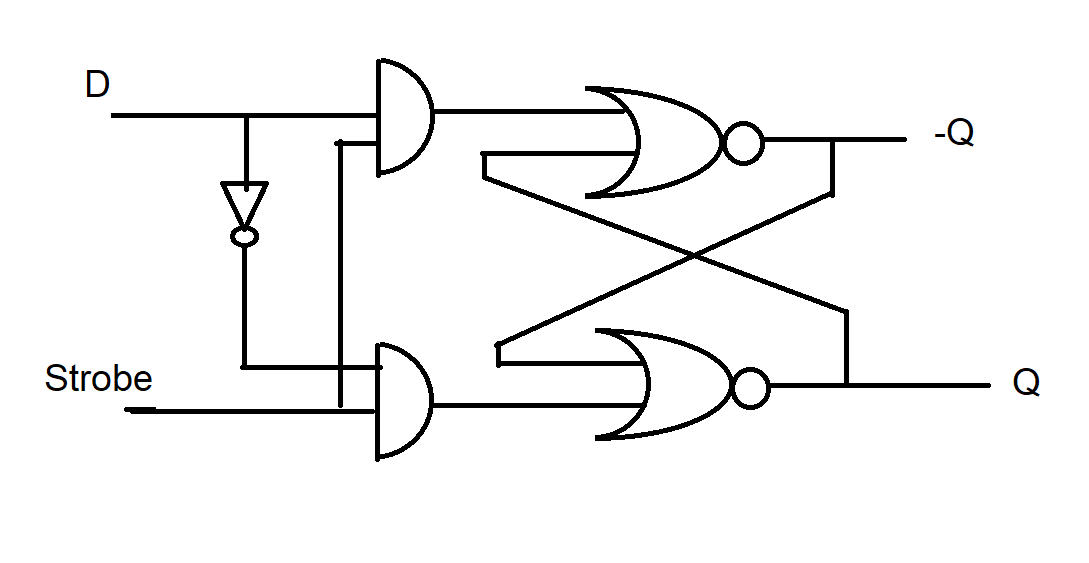
Descriviamo le uscite del circuito sequenziale (non è una tavola di verità ma ci assomiglia):

|  |  |  |  |
| --- | --- | --- | --- |
| S | R | Q | -Q |
| 0 | 0 | Q | -Q |
| 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 0 |

Notare che manca la riga per S= 1 R= 1.

Nel caso di S e R = 0 il circuito mantiene la configurazione precedente -> è un circuito sequenziale.

Ecco un altro flip-flop (è un evoluzione del primo):

 Questo è chiamato Flip-Flop di tipo D. Esso è costruito in modo che in ingresso alla parte del circuito sequenziale non arrivino mai in ingresso i valori 1,1. Infatti, se controlliamo il funzionamento del circuito otteniamo:

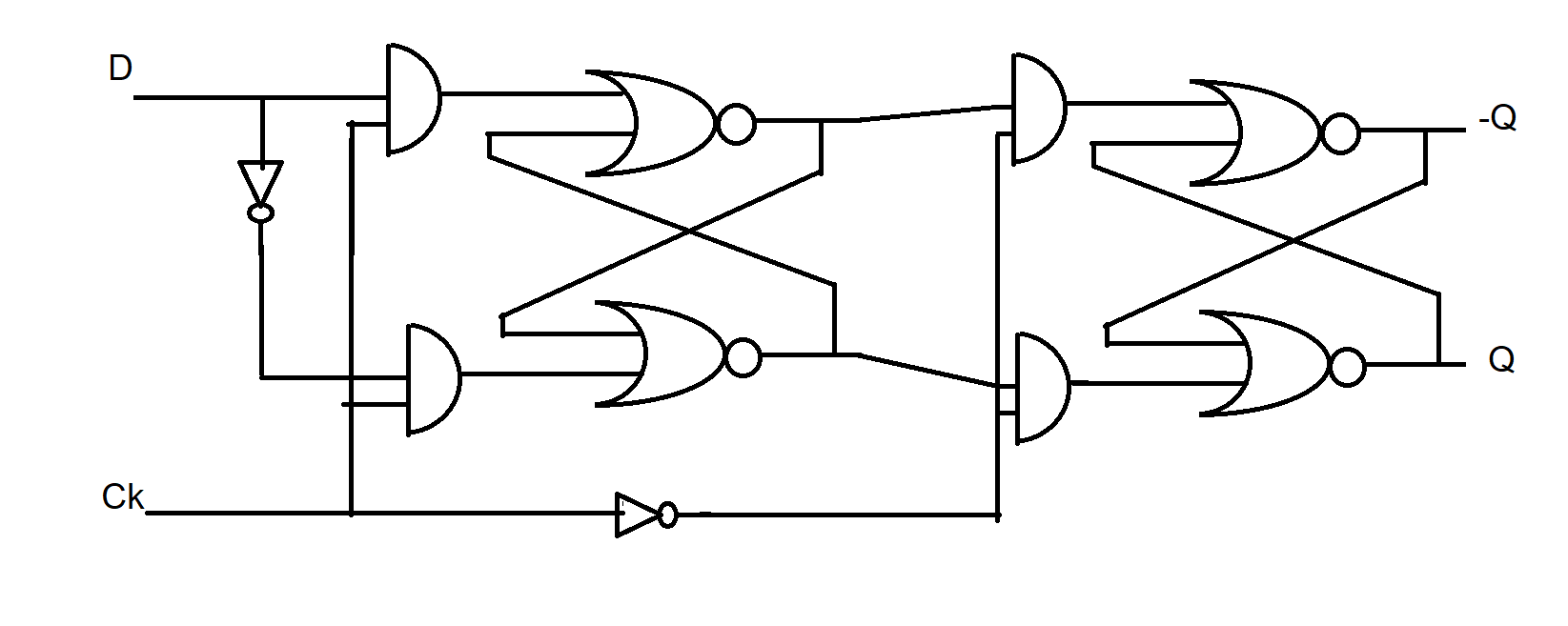
|  |  |  |  |
| --- | --- | --- | --- |
| D | Strobe | Q | -Q |
| 0 | 0 | Q | -Q |
| 1 | 0 | Q | -Q |
| 0 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 |

Questo “aggeggio” può correttamente memorizzare UN bit di informazioni: esso è il passo base per la costruzione di una memoria (se si fissa Strobe si può conservare il valore di D).

In un certo senso Il Flip-Flop D funziona come una vecchia macchina fotografica (di quelle che per funzionare dovevi togliere e rimettere il tappo rapidamente). Quindi Strobe dovrebbe restare fisso a 0, poi far passare Strobe da 0 a 1 e mentre accade ciò assicurarsi che il valore di D NON cambi. Dopodiché si fa ripassare rapidamente il valore di Strobe a 0 e ciò permette di mantenere salvato il valore che D aveva mentre Strobe valeva.

Il problema di questo genere di dispositivo è che, per evitare di causare errori (come scattare una foto a un oggetto in movimento) causati dal “movimento” della variabile D, il circuito avrebbe bisogno di un po’ di tempo per sincronizzarsi, ma questo rallenta il funzionamento del calcolatore.

Per ovviare a ciò è necessario l’equivalente di un otturatore per le macchine fotografiche (che riduca il tempo di esposizione). Questo nei circuiti sequenziali si realizza usando una configurazione chiamata master/slave. Se si vuole un Flip-Flop di tipo Master/Slave bisogna replicare il circuito che abbiamo visto fin’ora e fare un paio di aggiustamenti:



Il primo flip-flop è il master, il secondo è lo slave. Ck è l’ingresso “Clock”. Il funzionamento è che nel momento in cui il Clock cambia valore, da 1 a 0, il primo flip flop si blocca sul valore che conteneva e ciò fa sì che il secondo flip-flop memorizzi il valore nel master (poiché l’ordine sullo slave di memorizzare arriva dopo un tempo Espilon che è stato detto al master di bloccarsi).

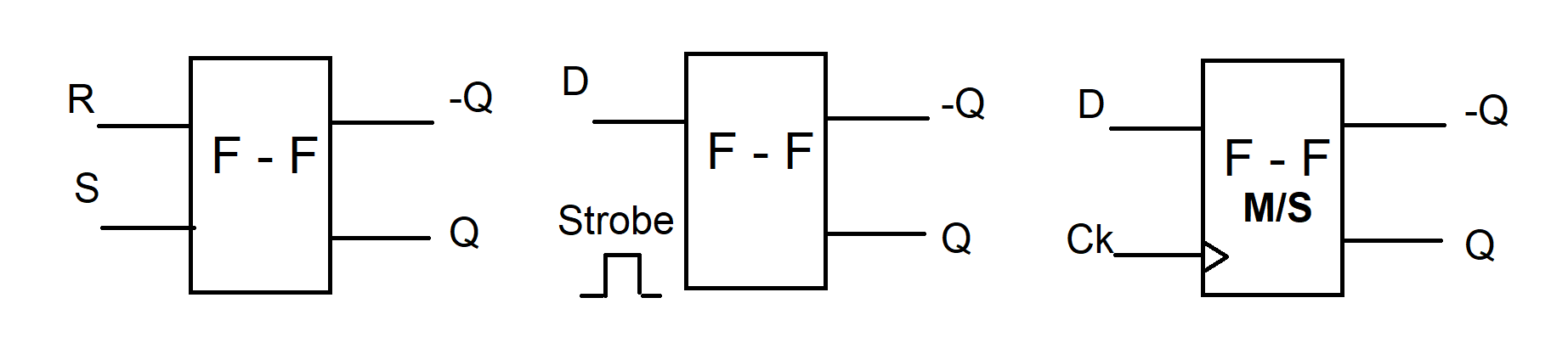
La differenza tra un Flip-Flop normale e un master/slave è quindi che al flip-flop master/slave quando vuole salvare un valore non importa se il valore D continua a cambiare durante il salvataggio (perché “l’istantanea” verrà “scattata” in un preciso istante).

Quindi:

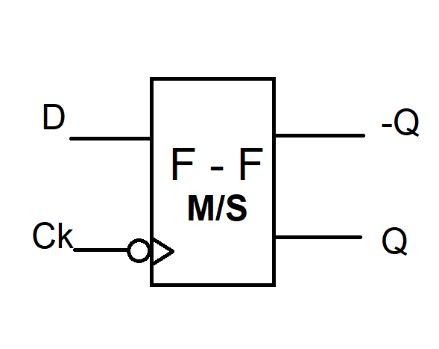
|  |  |  |  |
| --- | --- | --- | --- |
| D | Ck | Q | -Q |
| 0 | 1 -> 0 | 1 | 0 |
| 1 | 1 -> 0 | 0 | 1 |
| Tutti gli altri casi | Tutti gli altri casi | Q | -Q |

Un comportamento di questo tipo si chiama EDGE TRIGGERED (“comandato dai fronti”).

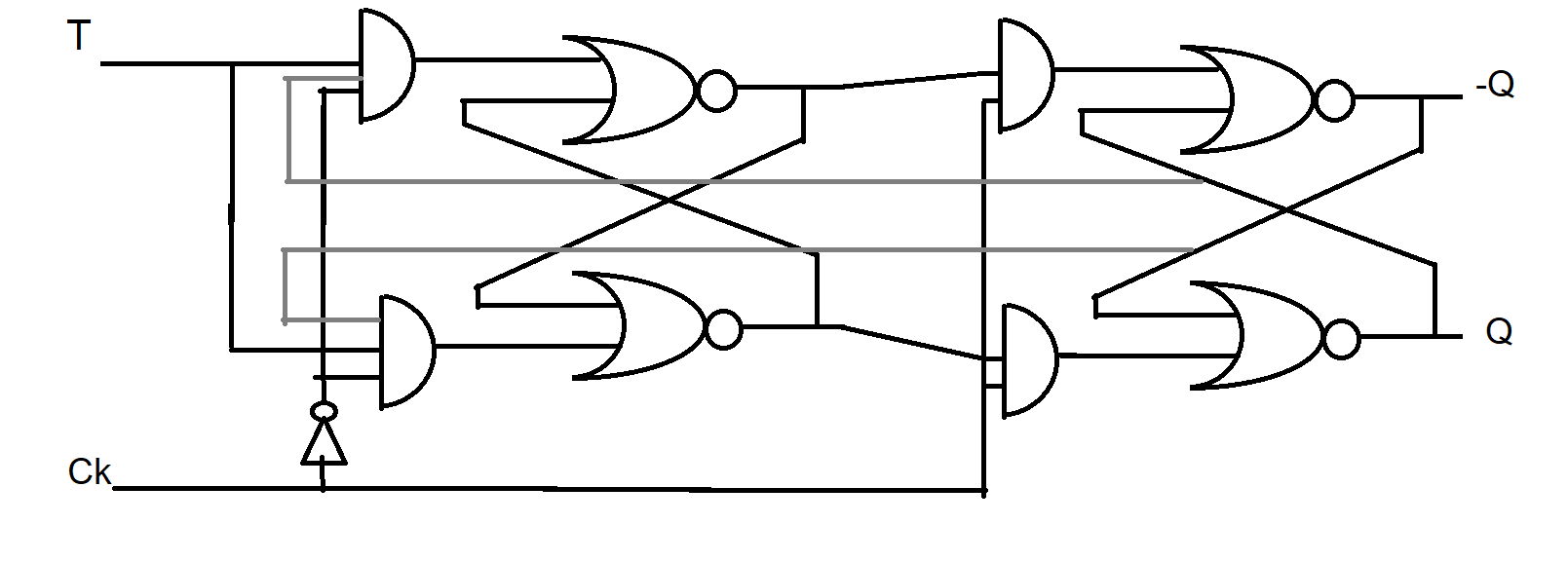
Riassumendo: Un circuito sequenziale è un dispositivo le cui uscite non dipendono solo dai valori in ingresso, ma anche dai valori inseriti in ingresso precedentemente. I circuiti sequenziali possono essere costruiti a partire da circuiti combinatori attraverso collegamenti tra uscite e ingressi che rendono il circuito chiuso. In generale un circuito sequenziale si costruisce a partire da un flip flop (che si può rappresentare a sto punto come una scatola chiusa). Esiste più di un tipo di flip flop però, Il flip-flop s-r, il flip-flop d e il flip-flop master/slave. Si possono rappresentare schematicamente così:



Il triangolino nell’ingresso Ck (Clock) indica che è EDGE TRIGGERED. Di base vuol dire che è suscettibile alle salite, se si aggiunge un pallino prima del triangolo significa che è suscettibile alle discese.



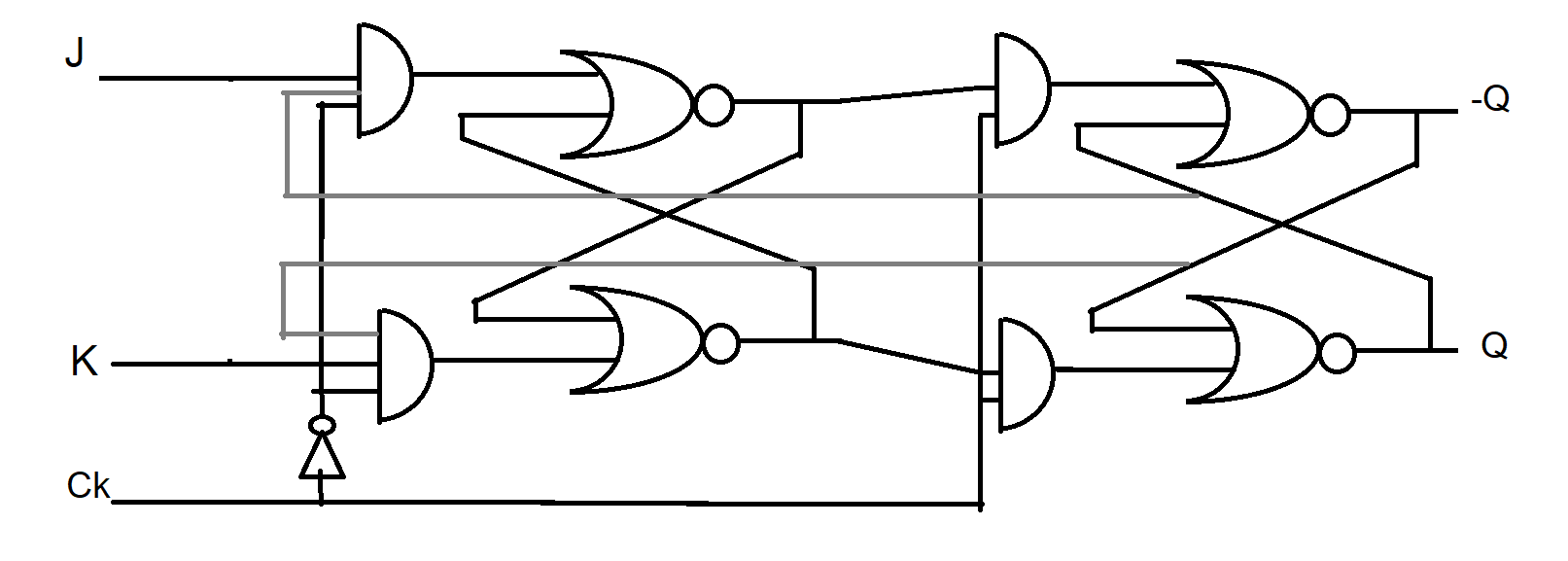
Esistono più Flip-Flop master/Slave. Due esempi sono il Flip-Flop T e il Flip-Flop J-K (Quest’ultimo è il più generico di tutti e può comportarsi come un qualsiasi altro flip-flop).



Ecco che gli AND del Master diventano a 3 ingressi. Se T vale 1, si ha una situazione simile a quello che accade con 1,1 nel flip-flop s-r. Tuttavia, in questo caso lo swap di valori tra Q e -Q non è più casuale (o = al tempo di esecuzione delle funzioni) ma è uguale alla frequenza delle variazioni del valore all’ingresso clock.

L’uso prevalente di questo tipo di flip-flop è quello di realizzare circuiti di tipo contatore (contando gli impulsi di clock).

Il flip-flop di tipo J-K invece si realizza così:



È molto simile al flip-flop di tipo T, solo che dipende da due variabili. Se J=K=0 si comporta come un Flip-Flop di tipo T con ingresso 0. Se J=K=1 si comporta come un flip-flop di tipo T con ingresso 1. Se J = 1 e K = 0 si comporta come un flip-flop di tipo D con ingresso 1 e se J = 0 e K = 1 si comporta come un flip-flop di tipo D con ingresso 0 (questo perché nel momento in cui J =\= K i collegamenti dalle uscite dello slave agli AND del master sono Inutili).

Tabellina:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| J | K | Ck | Q | -Q |
| 0 | 0 | 0 -> 1 | Q | -Q |
| 0 | 1 | 0 -> 1 | 1 | 0 |
| 1 | 0 | 0 -> 1 | 0 | 1 |
| 1 | 1 | 0 -> 1 | -Q | Q |

Come si vede I Flip-Flop di tipo D usano solo la seconda e la terza configurazione, mentre i flip-flop di tipo T usano solo la prima e la quarta configurazione.