# Progetto di Un Flip Flop di tipo T di tipo DDR sincronizzato sia sul fronte di salita che di discesa del clock e successiva trasformazione in Flip Flop JK

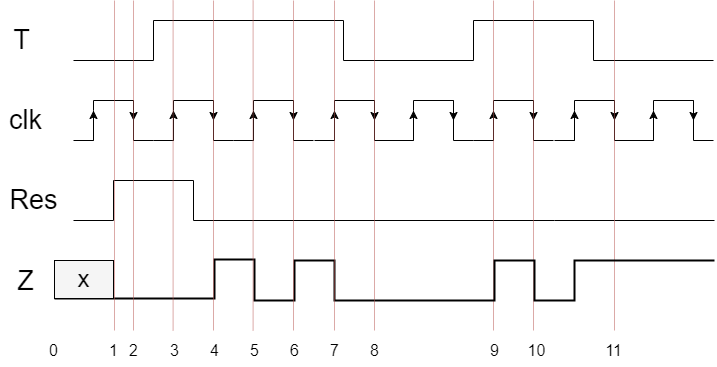
Individuare i segnali coinvolti:

Visto dall’esterno il sistema si presenta dotato di 2 ingressi e di un’uscita:  
- un segnale a livelli *T* che quando alto abilita la commutazione del FlipFlop negli istanti di sincronismo  
- un segnale *clk* i cui fronti di salita e di discesa scandiscono gli istanti nei quali il FF viene a commutare

Inoltre anche se non richiesto sarà opportuno introdurre un segnale di “reset” asincrono in grado di forzare il FF in una condizione nota.

Descrizione sintetica

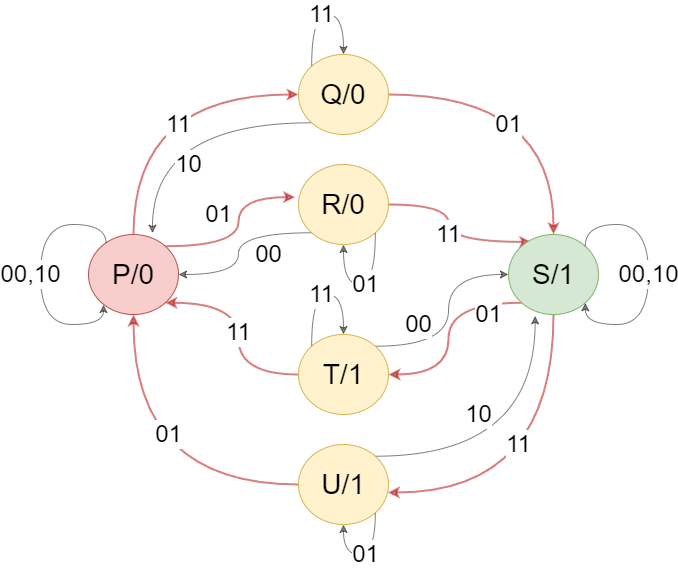
Il sistema è un elemento bistabile che prevede due condizioni di stabilità: lo stato ALTO e lo stato BASSO.  
Per passare dall’uno all’altro devono verificarsi 2 condizioni: il segnale T deve essere ALTO e deve presentarsi un fronte di clock (di salita o di discesa)



1. All’istante di partenza il sistema può accendersi in una qualsiasi delle sue condizioni e non vi è modo di prevedere se sarà nella condizione alta o bassa
2. All’istante 1 si attiva il segnale di reset asincrono quindi il sistema si riporta nel più breve tempo possibile nella condizione con uscita bassa
3. All’istante 2 si presenta un fronte di clock (di discesa) ma essendo il segnale di Reset ancora attivo (oltre ad essere il segnale T basso) l’uscita non subisce modifiche
4. All’istante 3 si sarebbero le condizioni per una commutazione (T alto e fronte di clk) ma il perdurare del segnale di Reset alto ne blocca l’evoluzione
5. All’istante 4 il reset è spento il segnale T è alto, si presenta un fronte di clock ed l’uscita commuta il uso stato, così come nei successivi istanti 5,6,7.
6. All’istante 8 pur essendo presente un istante di sincronismo il segnale di eccitazione del Flip Flop (T) è basso e la commutazione viene inibita. Questa situazione perdura fino all’istante 9
7. Negli istanti 9 e 10 si ritrovano le condizioni di T alto e presenza dei fronti di clock, il che porta nuovamente il FF a commutare ogni volta.
8. Nell’istante 11 T è nuovamente basso e l’uscita mantiene il suo stato.

Evidenziare i meccanismi di funzionamento

Come evidenziato sopra le condizioni che portano ad una modifica dell’uscita possono essere individuate in particolari sequenze presenti sui segnali di ingresso: in particolare il passaggio da una condizione all’altra può avvenire secondo due precise sequenze: chiamati in ordine cT i segnali di clock e di eccitazione tali sequenze sono rispettivamente: 11 🡪 01 (fronte di discesa del clock e T alto) oppure 11 🡪 01 (fronte di salita del clock e T alto). Per evidenziare tale funzionamento in un diagramma di Moore si potrebbe pensare ad una soluzione quale quella riportata qui di seguito ove le suddette sequenze che inducono ad una commutazione sono evidenziate in rosso.



Da cui si evince che:

* Per distinguere entrambe le possibili sequenze vi sia bisogno di usare complessivamente 6 stati di cui 4 intermedi per salvare memoria del primo passo della sequenza di commutazione e nel quale il sistema rimane stabile fino al modificarsi ulteriore dei segnali di ingresso.   
  Qui si presentano 2 possibilità:
  + Se la sequenza non dovesse completarsi si ritorna allo stato di partenza
  + Se la sequenza si completa si completa la transizione
  + Si noti che non viene presa in considerazione la commutazione simultanea di entrambi i segnali di ingresso.
* Vi è inoltre da notare che la coppia di segnali che completano la transizione potrebbe essere interpretata anche come l’inizio della transizione inversa. Infatti a sequenza ultimata la macchina non permane nello stato P o S ma prosegue immediatamente verso uno degli stati intermedi ove eventualmente completare la sequenza inversa.
* Negli stati S e P il sistema si ferma solo nel momento in cui il segnale T dovesse abbassarsi inibendo quindi qualsiasi transizione.

La suddetta macchina può essere descritta efficacemente anche attraverso la Tavola di Huffmann:



Applicare il metodo di Ginsburg a tale macchina per ricercare eventuali semplificazioni non porta ad alcun risultato, ma se si apportano alcune modifiche: ovvero

* Si raddoppiano gli Stati P e S in due stati tra loro compatibili P1-P2 ed S1-S2



* E si nota che lo stato P1 (stabile con 00) non potrà mai raggiungere lo stato Q perché la transizione richiederebbe la modifica contemporanea di entrambi gli ingressi
* Analogamente lo stato P2 non può raggiungere R
* Analogamente lo stato S1 non può raggiungere U
* Analogamente lo stato S2 non può raggiungere T
* Si eliminano le transizioni multiple sugli stati S e P andando direttamente negli stati finali   
    
  (queste modifiche sono evidenziate in rosso)



Si può notare che la suddetta macchina ammette ora la semplificazione secondo il metodo di Ginsburg, riconoscendo la compatibilità rispettivamente di: (P1,R), (P2,Q), (S1T),(S2U)



La macchina così semplificata potrebbe essere descritta verbalmente con le seguenti considerazioni:

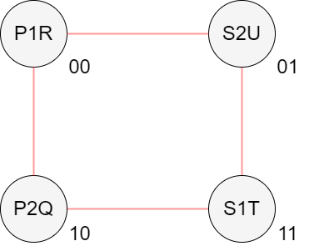
* I passaggi di stato avvengono solamente quando si modifica il segnale di clock, ovvero fintanto che il clock è stabile, qualsiasi modifica sul segnale T non altera lo stato.
* Vi si possono identificare 4 stati di cui 2 con uscita bassa e 2 con uscita alta, inoltre 2 stati sono stabili con il clock basso e 2 con il clock alto
* Ad ogni modifica del clock (fronte di salita o fronte di discesa) in base al valore presente sul segnale T la macchina sceglie se andare in nello stato la cui uscita è 0 oppure in quello la cui uscita è uno in base alla logica desiderata.

Il che a ben vedere può essere preso come metodo generale per descrivere molte tipologie di macchine che risultino sincronizzate su un solo specifico segnale di sincronismo, indipendentemente se di questo si debba analizzare il fronte di salita, di discesa oppure entrambi (DDR) o ancora fosse impulsivo.

Sintesi

La macchina così descritta può ora essere facilmente sintetizzata:

Si Inizia individuando una opportuna codifica per gli stati che minimizzi le variabili di stato impiegate e che sia priva di corse critiche. Si può notare dalla mappa delle transizioni



Che una volta garantita l’adiacenza degli stati come sopra rappresentato, il passaggio dall’uno all’altro avverrà sempre con la modifica di una ed una sola variabile, il che rende il sistema al riparo da eventuali corse.

Il che porta ad una tavola di flusso quale quella a seguire:



La quale porta ed evidenziare le seguenti funzioni per quanto riguarda le variabili di stato



Tali funzioni possono essere realizzate in diversi modi, ad esempio impiegando due piccole memorie ROM (LUT – Look Up Tables) a 16 bit ciascuna nelle quali precaricare le uscite desiderate.

Si noti peraltro che anche se non espressamente richiesto è sempre bene dotare tutti i circuiti di un segnale di Reset esplicito che possa forzarli in una condizione nota. Tale segnale è utile sia in fase realizzativa per garantire che il circuito parta da una condizione nota, sia in fase di simulazione per consentire al simulatore di uscire da una condizione che altrimenti sarebbe indeterminata.

Se scegliamo che lo stato di Reset coincida con lo stato P1R (codificato con 00), forzare il circuito in tale stato attraverso un segnale asincrono può essere ottenuto modificando le equazioni di cui sopra con:

Così facendo, quando il segnale “Res” dovesse essere alto le variabili di stato vengono entrambe forzate al valore 0, mentre quando il segnale è basso il suo effetto è ininfluente sulle equazioni sopra trovate.

Il sistema può essere descritto in linguaggio Verilog HDL ad esempio col seguente codice:

module FFT\_ddr\_v2(T,clk,Res,Z);

input wire T;

input wire clk;

input wire Res;

output wire Z;

wire y1,y2;

wire y1r,y2r;

assign y2r = y2 & ~Res;

assign y1r = y1 & ~Res;

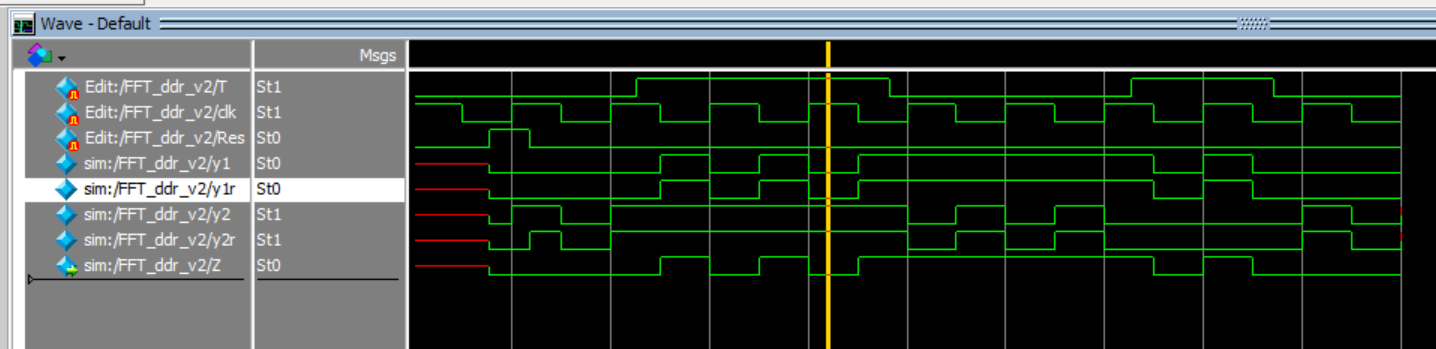
assign y1 = (y1r&~T) | (y2r&T&~clk) | (~y2r&clk&T);

assign y2 = (y2r&T) | (y1r&~T&~clk) | (~y1r&clk&~T);

assign Z = y1;

endmodule

e successivamente simulato



Oppure, per avere maggior aderenza con la realtà introducendovi ipotetici tempi di ritardo:

module FFT\_ddr\_v2(T,clk,Res,Z);

input wire T;

input wire clk;

input wire Res;

output wire Z;

wire y1,y2;

wire y1r,y2r;

assign #2 y2r = y2 & ~Res;

assign #2 y1r = y1 & ~Res;

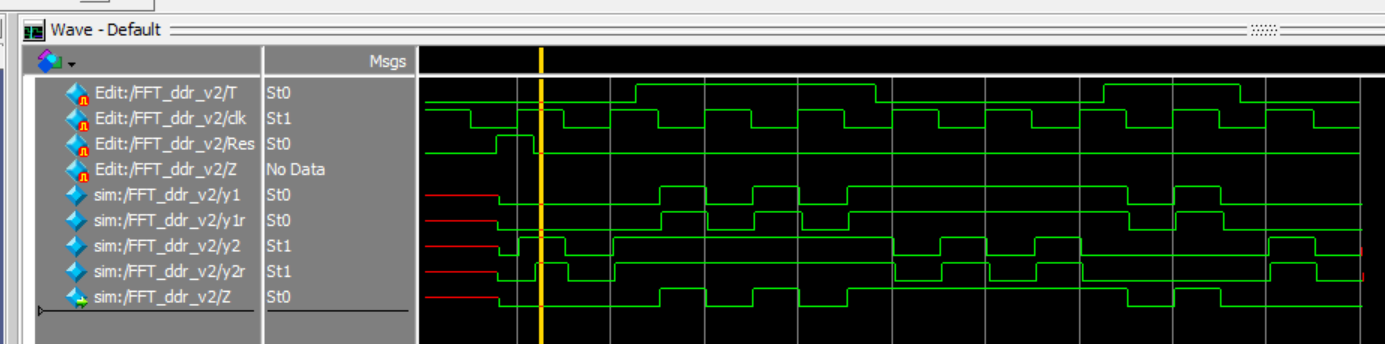
assign #2 y1 = (y1r&~T) | (y2r&T&~clk) | (~y2r&clk&T);

assign #2 y2 = (y2r&T) | (y1r&~T&~clk) | (~y1r&clk&~T);

assign Z = y1;

endmodule

Che comunque porta ad un risultato corretto, fintato che i tempi di ritardo sono significativamente inferiori al periodo del clock.



Fase 2. Trasformazione tra Flip-Flop

Quando già si disponga di una tipologia qualsiasi di Flip Flop, sincronizzato da un certo segnale di sincronismo, la sua trasformazione in un'altra tipologia, che sia SEMPRE sincronizzata sul MEDESIMO segnale di sincronismo può essere eseguita senza dover modificare la macchina asincrona nel suo INTERNO, ma solamente aggiungendovi alcuni elementi ESTERNI alla stessa.

In particolare si può notare che un Flip Flop può essere visto come una macchina SINCRONA elementare e come tale può essere progettato ricorrendo all’impiego di qualsiasi tipologia di FF usati come elementi di memoria, pur di fornire ad essi la corretta variabile di eccitazione.

Nel caso specifico se si volesse trasformare il FF-T appena realizzato in un FF-JK (COL MEDESIMO SEGNALE DI SINCRONISMO – pertanto DDR) si può seguire la seguente procedura:

1. Si descriva il FlipFlop da realizzare come una macchina SINCRONA

Ad esempio impiegando la tavola di Huffmann



1. Si scelga una codifica opportuna per gli stati. In questo caso risulta piuttosto evidente che la codifica può, per semplicità, coincidere con l’uscita stessa.



1. Si evidenziono le transizioni. In questo caso viene evidenziata attaverso la presenza di un apice una “alterazione” della variabile di stato (0🡪1: 1’ o 1🡪0: 0’), mentre l’assenza dell’apice indica un “mantenimento” della variabile di stato (0🡪0: 0 o 1🡪1: 1’).



1. Note che siano le mappe di eccitazione del Flip Flop T (quello che abbiamo appena progettato nella prima parte dell’esercizio)



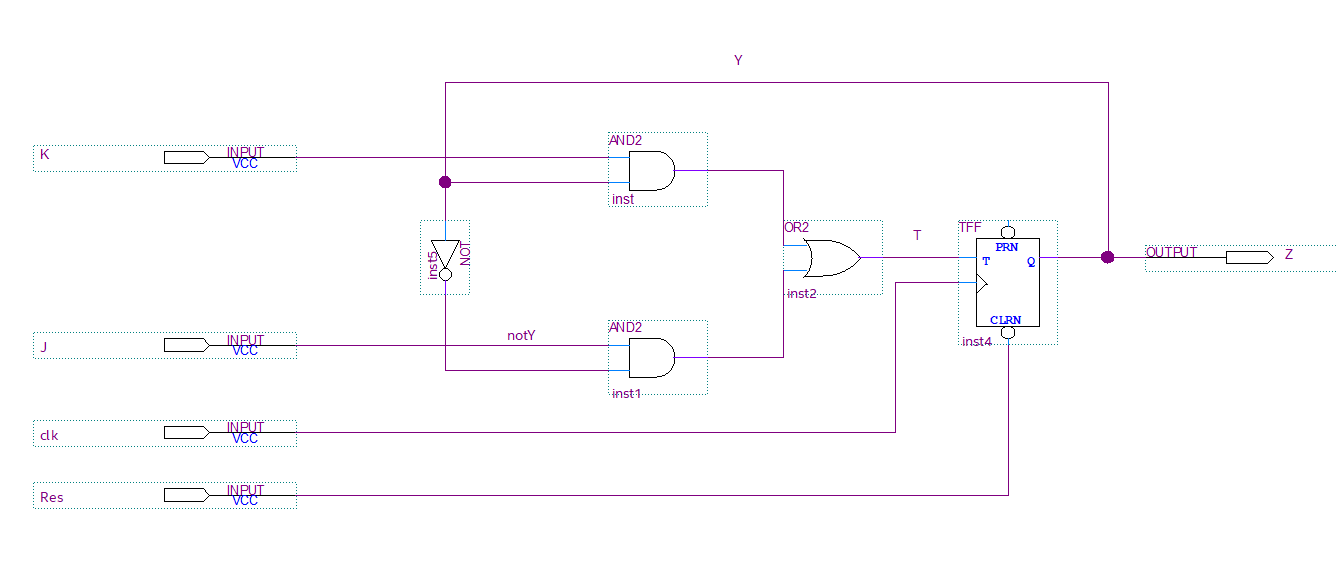
1. Si può (sostituendo alla transizione desiderata l’eccitazione che la condiziona) arrivare a definire quale sia la corretta eccitazione da far arrivare al FF T per indurre la transizione desiderata:



1. E da essa pervenire all’equazione di Eccitazione



1. E con essa allo schema di Trasformazione



Che trova la seguente descrizione in Verilog HDL, imponendo ipotetici ritardi:

module FFTinJK(K,J,clk,Res,Z);

input wire K,J;

input wire clk;

input wire Res;

output wire Z;

wire notY;

wire T;

wire Y;

wire WIRE\_0;

wire WIRE\_1;

assign #2 WIRE\_1 = K & Y;

assign #2 WIRE\_0 = J & notY;

assign #2 T = WIRE\_0 | WIRE\_1;

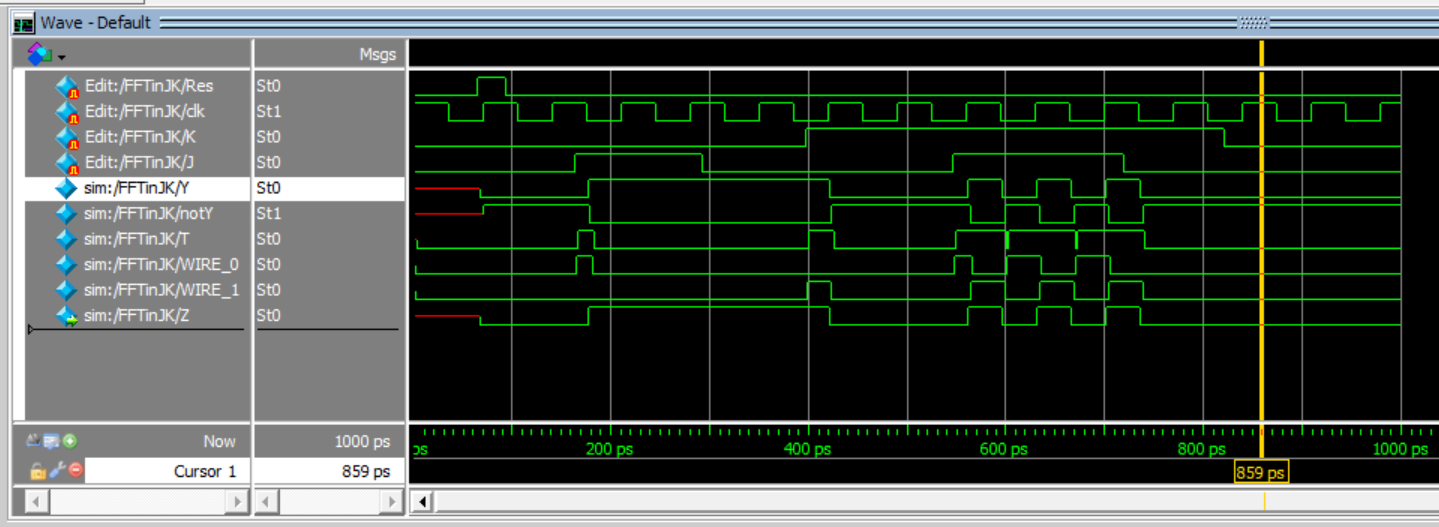
assign #2 notY = ~Y;

assign Z = Y;

FFT\_ddr\_v2 FFTinst(.T(T),.clk(clk),.Res(Res),.Z(Y));

endmodule

E che in fase di simulazione porta al seguente risultato.



Vi è da notare che sebbene la strategia seguita abbia seguito le tracce per il disegno di circuiti sincroni ciò che ne esce globalmente può essere visto come un circuito asincrono con tre loop di reazione (di cui due interni al Flip Flop) ed uno esterno, quindi una sua analisi porterebbe ad evidenziare la presenza tre variabili di stato per conseguenti otto stati.

Inoltre un progetto accurato richiederebbe quantomeno di verificare che siano rispettate tutte le tempistiche per un corretto funzionamento del FF e che in particolare il segnale d’ingresso T del medesimo sia stabile in un opportuno intervallo attorno l’istante di campionamento compreso tra il tempo di Set-up (*Tsu*) ed il tempo di Hold (*Th*) . Si notino ad esempio i “glitches” presenti sul segnale T che è portato all’ingresso del FF: con i tempi di ritardo finora ipotizzati questi arrivano sufficientemente lontani dal fronte del clock ed il dispositivo funziona, ma se essi dovessero ad esempio arrivare troppo ravvicinati ad esso, questi potrebbero mettere il sistema in condizioni di metastabilità, inficiandone il corretto funzionamento.

