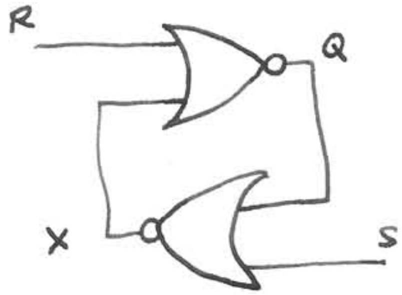
lez. 22/10/14
(Merc., 3h)

Sono ripartito dal latch di NOR :



SRQ	Q'	
000	0	} Hold
001	1	
010	0	} Reset
011	0	
100	1	} Set
101	1	
110	0	} "Forbidden"
111	0	

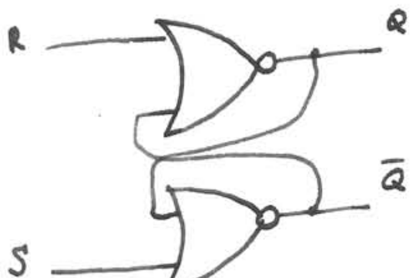
↑ queste sono le transizioni con esito STABILE. In realtà, poiché USCIRE dalla SR = 11 "porta" più a instabilità / ambiguità

allora $Q'(1,1,Q) = X$

con X = "don't care"

(it: "condizioni di indifferenza"; UK: "chi se ne frega?")

$\equiv$



L'eq. caratteristica (trans. stato) per l'SR è

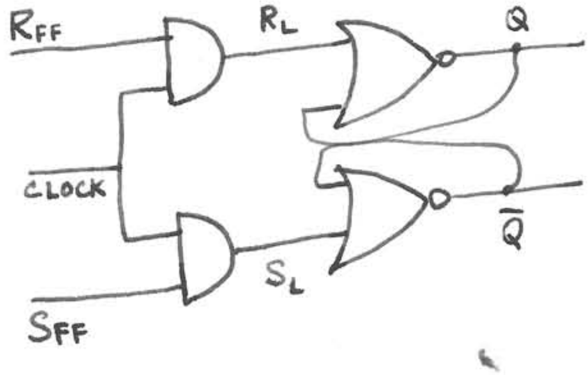
$Q' = S\bar{R} + \bar{R}Q$ (se i don't care sono presi dalla tabella originale, cioè 0)

oppure (più semplice)

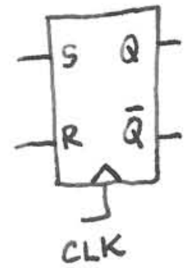
$Q' = S + \bar{R}Q$ (se i don't care sono entrambi a 1)

SR \ Q	0	1
00	0	1
01	0	0
11	X	X
10	1	1

Il FLIP-FLOP SR è la versione "gated" del latch:



$\equiv$



Notiamo come

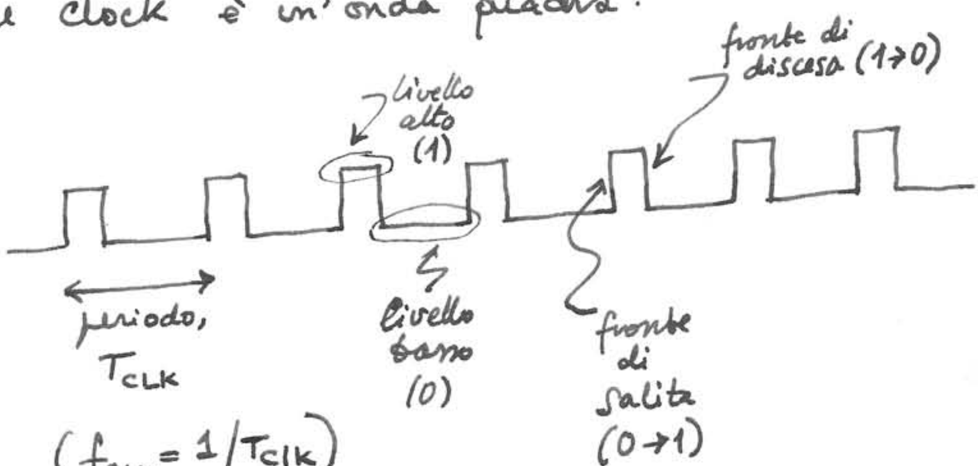
$$Q'_{FF} = \overline{CLK}\, Q + CLK\, Q'_{LATCH}$$

ossia per il Flip-flop SR abbiamo

$$Q' = \begin{cases} S + \bar{R}Q & \text{se } CLK = 1 \\ Q & \text{se } CLK = 0 \end{cases}$$

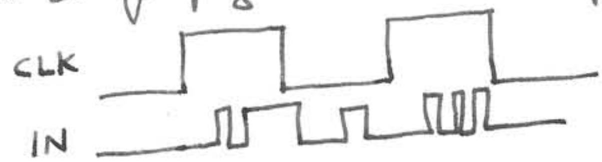
(cioè il clock consente al latch di commutare solo sui livelli alti) altrimenti fa HOLD

Il clock è un'onda quadra:



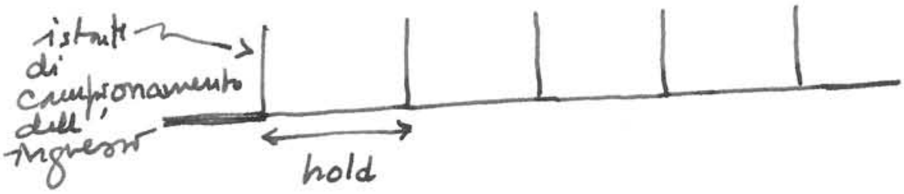
Il "duty cycle" è la frazione di periodo in cui il livello è alto:

Nelle applicazioni pratiche, è conveniente che l'ingresso (necessario alla commutazione) sia campionato in istanti molto piccoli di tempo, per evitare che — con un d.c. troppo grande — valori spuri o instabili di commutazione si propaghino ad altre parti di un sistema:

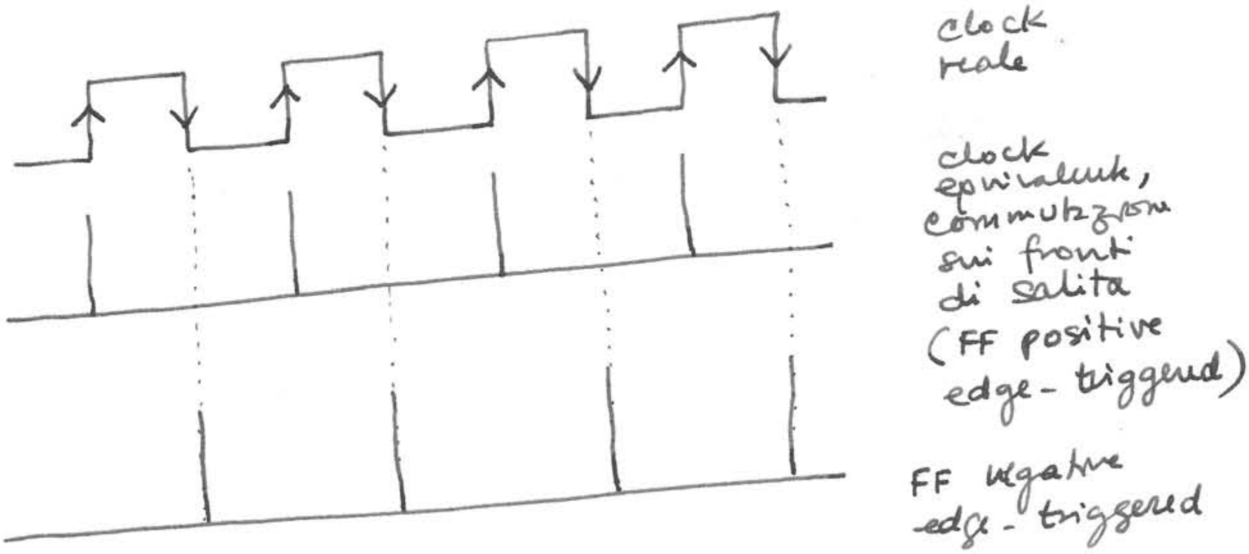


%

Quindi, idealmente si vorrebbe disporre di un clock con d.c. pari a 0 (= δ di Dirac):

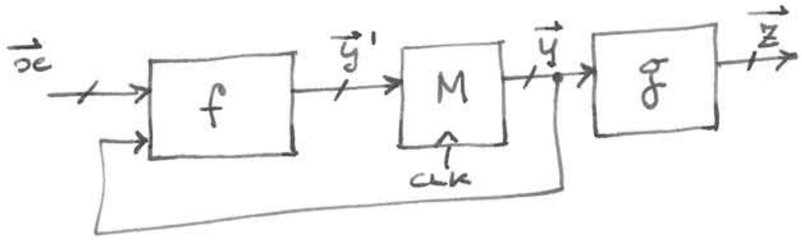


In pratica, si ottiene questo comportamento modificando opportunamente il FF appena visto (che commuta SUI LIVELLI) trasformandolo in un FF che commuta SUI FRONTI (cfr. Hamacher):



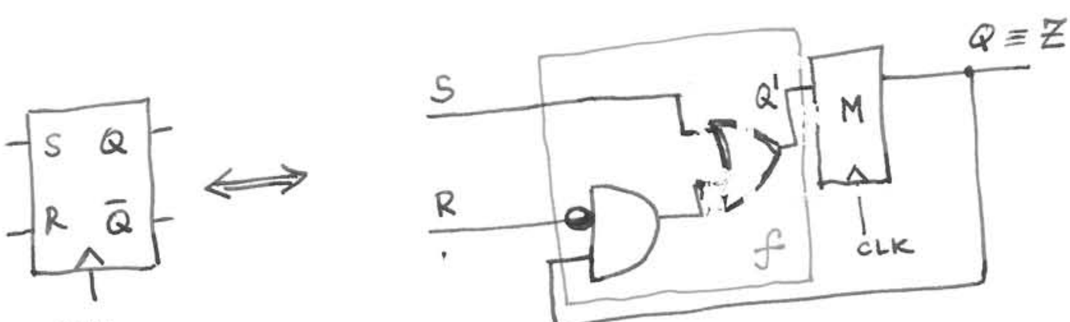
NOTA: Il valore campionato è quello immediatamente <u>precedente</u> all'arrivo del fronte.

Torniamo al FF SR, e vediamo qual è il suo legame con lo schema a blocchi generale (visto in precedenza) di una macchina sequenziale di Moore:



Nel caso del FF SR, lo stato è direttamente visibile in uscita. Quindi $\vec{z} \equiv \vec{y}$, ovvero $z = Q$. La funzione g è dunque l'IDENTITA'. Questo accadrà per tutti i FF che vedremo.
La funzione $\vec{y}' = f(\vec{x}, \vec{y})$ invece non è banale. Infatti abbiamo $Q' = S + \bar{R}Q$, con $\vec{x} = (S, R)$ e $\vec{y} = Q$.

In sostanza



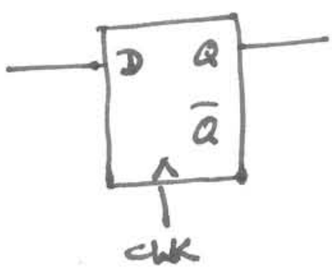
Ci chiediamo ora a quale speciale macchina sequenziale corrisponda il blocco M. La risposta è che il blocco M è descritto dalla coppia di eq.

$\vec{y}' = f(\vec{x}, \vec{y}) \equiv \vec{x}$ (ossia STATO FUTURO $\equiv$ INGRESSO)

$\vec{z} = g(\vec{y}) \equiv \vec{y}$ (ossia USCITA $\equiv$ STATO PRESENTE)

./.

Una macchina siffatta si chiama REGISTRO DATI, e nella versione a 1 bit di stato ($y=Q$) ha equazione $Q' = x$, dove x è l'ingresso esterno applicato. La macchina prende il nome di FLIP-FLOP D (dove D sta per "delay", perché per l'appunto il blocco M ritarda di 1 colpo di clock lo stato futuro). L'ingresso x viene usualmente chiamato D:



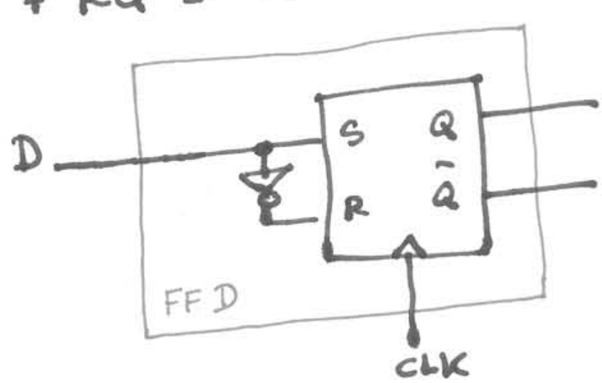
$Q' = D$

D	Q'	
0	0	Reset
1	1	Set

NOTA: Questo FF, il più semplice che esista, ha uno stato futuro indipendente dallo stato presente, e quindi non c'è alcun valore dell'ingresso che realizza un HOLD. Di fatto, l'HOLD viene realizzato dal clock nei semiperiodi bassi.

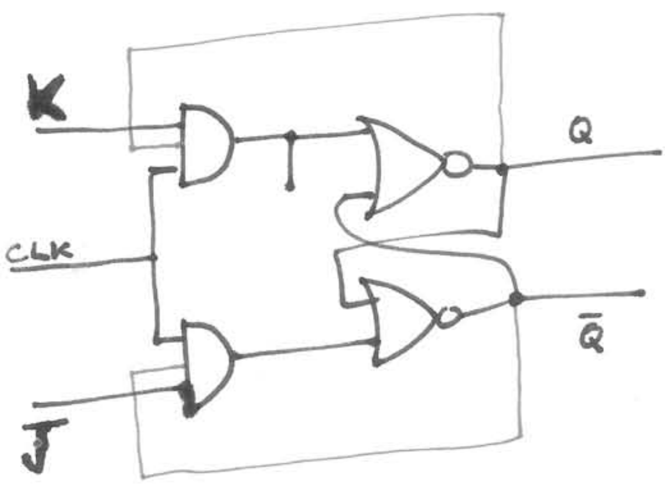
Si può ottenere il FF D dal FF SR ponendo $D = S = \bar{R}$. Infatti

$$Q'_{SR} = S + \bar{R}Q = D + DQ = D$$



Notiamo che questo FF evita automaticamente la configurazione $S=R=1$. Infatti, è sempre $S=\bar{R}$

Un altro modo per evitare la SR = 11 è di mettere in AND a S e R due valori complementari. E' ciò che si fa nel FLIP FLOP JK (da Jack Kilby):



Definiamo

$S = J\bar{Q}$

$R = KQ \Rightarrow \bar{R} = \bar{K} + \bar{Q}$

Dall'eq. caratt. del FF SR:

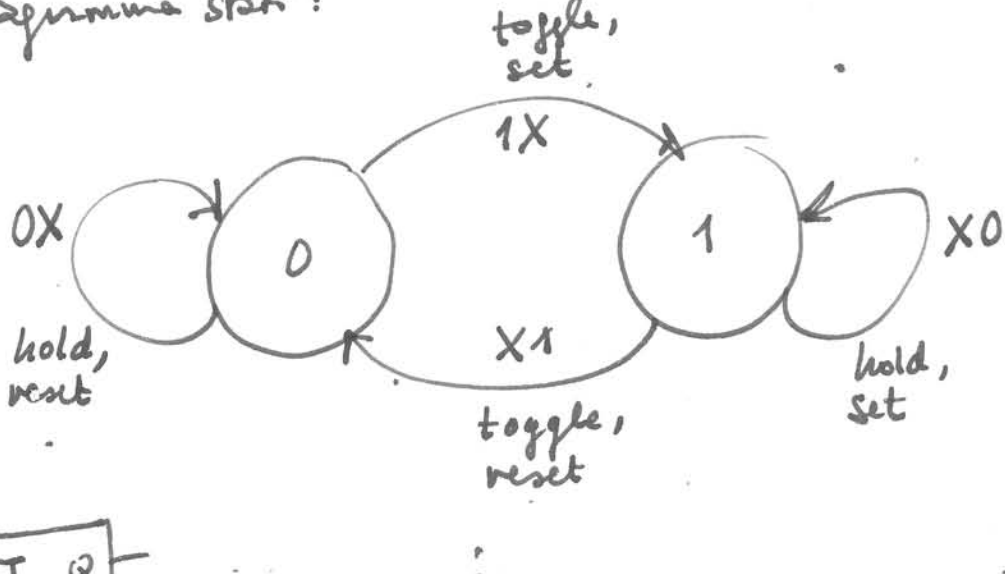
$$Q' = S + \bar{R}Q = J\bar{Q} + \bar{K}Q$$

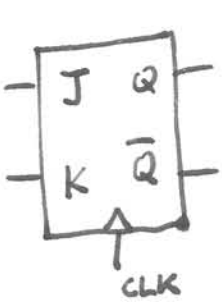
Tabella:

J	K	Q'	transizione
0	0	Q	hold
0	1	0	reset
1	0	1	set
1	1	$\bar{Q}$	toggle (nuova, al posto di "forbidden")

è il FF più generale, può fare tutto

Diagramma stati:

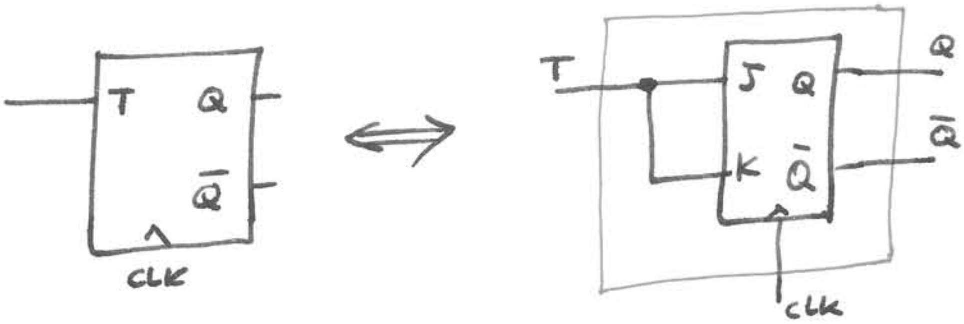




NOTA:

Un ultimo tipo di FF è il T, ottenuto con

$$J = K = T \quad :$$

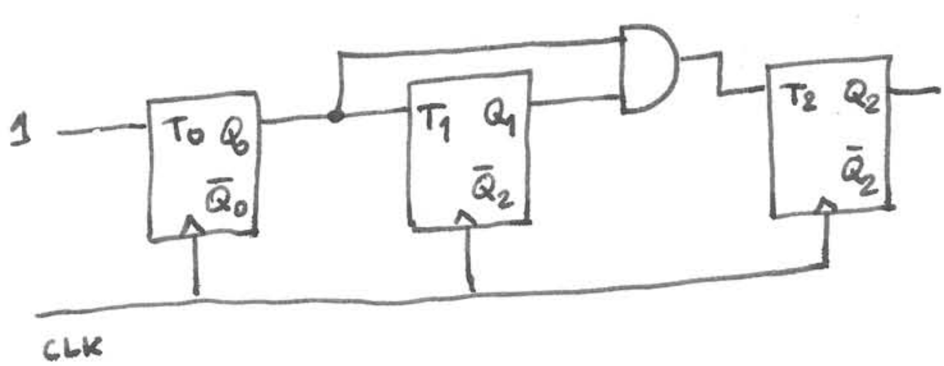


Come si vede dalla tabella del JK, questo FF è complementare al D, perché può fare solo hold e toggle:

T	Q'	trans.
0	Q	hold
1	$\bar{Q}$	toggle

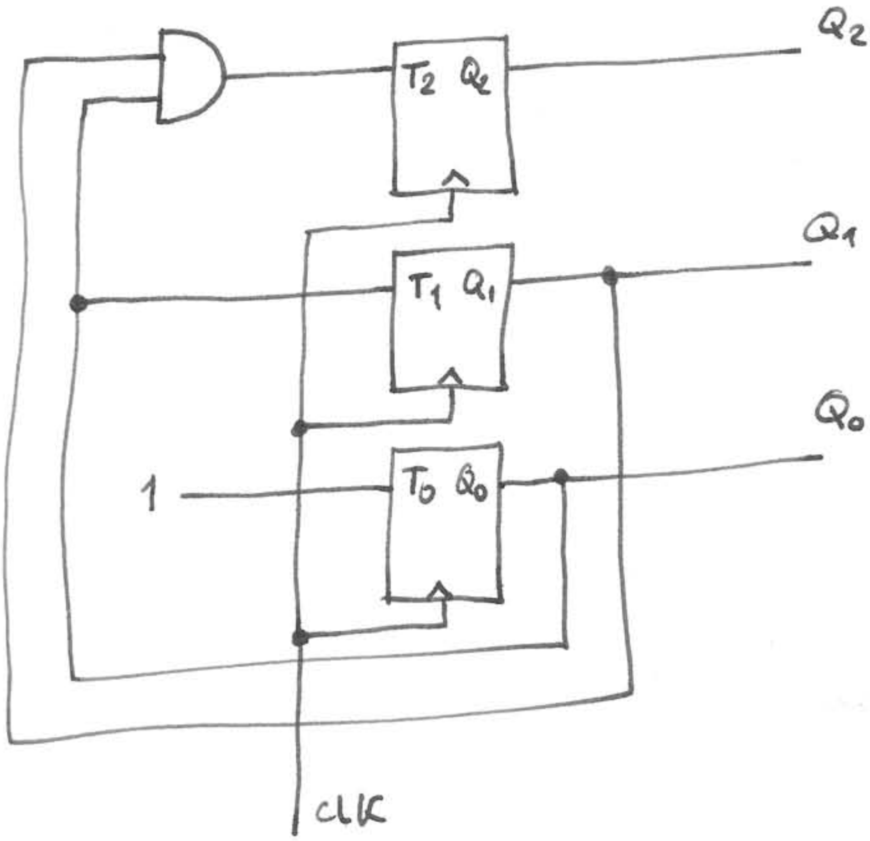
$$Q' = T\bar{Q} + \bar{T}Q = T \oplus Q$$

Questo FF trova impiego ad es. nella realizzazione di contatori. Ad es. per un contatore UP modulo 8 il FF0 è sempre in toggle, mentre FF1 e FF2 commutano solo se i FF_i precedenti hanno uscita 1:



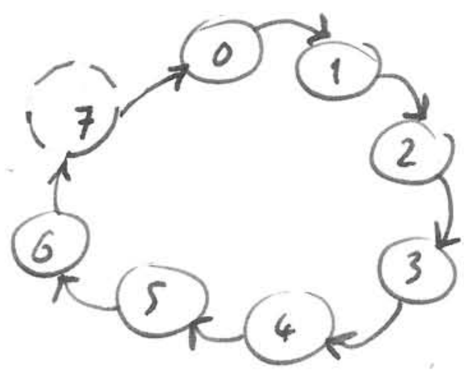
t	$Q_2Q_1Q_0$
0	0 0 0
1	0 0 1
2	0 1 0
3	0 1 1
4	1 0 0
5	1 0 1
6	1 1 0
7	1 1 1
8	0 0 0
...	...

Possiamo ridisegnare il contatore precedente con uno schema verticale, anziché orizzontale:



Notiamo che

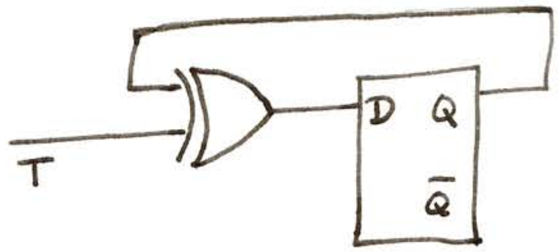
1) il contatore NON ha ingressi, è una cosiddetta "macchina autonoma", e come tale può avere solo un'evoluzione periodica, con transizioni obbligate. Infatti in qsto caso abbiamo:



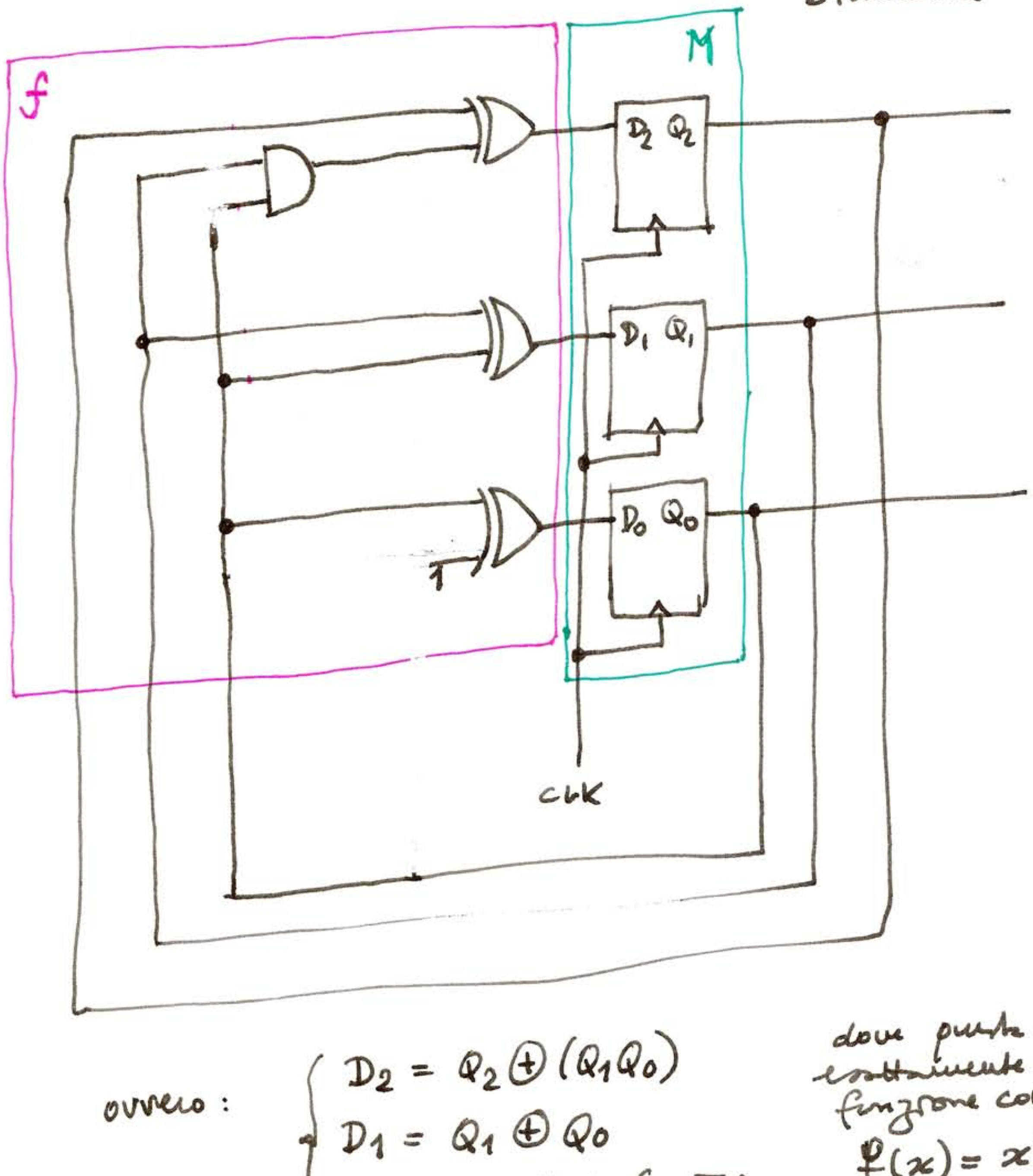
2) La parte combinatoria coinvolta dal contatore è quasi tutta (a parte la porta AND esterna) contenuta dentro i FF T:

%

Infatti in termini di schema a blocchi generali, la funzione f del FFT è $T \oplus Q$:



Quindi il disegno di prima è equivalente allo schema a blocchi standard:



ovvero:

$$\begin{cases} D_2 = Q_2 \oplus (Q_1 Q_0) \\ D_1 = Q_1 \oplus Q_0 \\ D_0 = Q_0 \oplus 1 \ (\equiv \overline{Q}_0) \end{cases}$$

dove questa è esattamente la funzione combinatoria

$$f(x) = x \text{ "+" } 1 \ (\mathrm{mod}\ 8)$$