

Latch e Flip-Flop*

Corso di Architetture dei Sistemi di Elaborazione

Prof. Antonino Mazzeo

11 dicembre 2018

Univ. di Napoli Federico II

*Si avvisa che il presente capitolo è in bozza e può contenere errori e imprecisioni

Notazioni utilizzate

I valori vero o falso delle variabili booleane sono indicati dal nome senza alcun segno, se veri e dal nome con apposta una barra superiore (\overline{X}) se falsi. Oltre a tale denotazione, in molti testi, spesso per problemi tipografici, per rappresentare le variabili negate sono utilizzate altre notazioni quali: un apice (X') o un punto esclamativo ($X!$) o un asterisco, (X^*) posposto al nome della variabile e, in taluni casi, anche con una barra a esso preposta (\overline{X}). Nel presente testo si utilizzeranno o la notazione con barra superiore o quella con apice (X').

Gli operatori AND e OR sono indicati rispettivamente con i simboli \cdot e $+$, ad esempio ($X \cdot Y$) e ($X + Y$); gli operatori NAND e NOR rispettivamente con una freccia in alto ($X \uparrow Y$) e una freccia in basso ($X \downarrow Y$).

1 Introduzione

Questo capitolo è dedicato alla presentazione dei componenti bistabili (i flip-flop e i latch), **realizzati** con tecnologie elettroniche. I dispositivi ampiamente impiegati nella progettazione dei sistemi digitali e alla base della costruzione dei registri e delle memorie, rappresentando essi una cella elementare di memoria della capacità di un bit. I flip-flop e i latch sono realizzati a partire da un particolare circuito elettronico analogico detto multivibratore *bistabile*, composto da componenti elettronici attivi (inizialmente costruito a valvole e, successivamente, con dispositivi a semiconduttore di varie tecnologie quali i transistor bipolari e quelli MOS).

I flip-flop e i latch, possono essere studiati con i metodi tradizionali di analisi e progettazione dei circuiti elettronici o come dispositivi logici, utilizzando i metodi di analisi e progettazione tipici delle reti logiche (reti combinatorie e sequenziali sincrone e asincrone) e sintetizzati utilizzando come componenti elementari le porte logiche di tipo inverter, AND, OR, NAND, NOR, etc..

Nel seguito, è presentata, dopo una prima sintetica illustrazione dei circuiti bistabili, la classificazione e l'illustrazione delle caratteristiche funzionali dei latch e flip-flop, il procedimento per la loro analisi e sintesi, anche con riferimento ad alcuni esempi di cui è anche proposta una descrizione in linguaggio VHDL e mostrati i risultati delle relative simulazioni.

I sistemi digitali si compongono, come visto, di reti logiche che possono essere di tipo combinatorio o sequenziale. **Le reti combinatorie sono, come è noto, sistemi “senza memoria”, nel senso che esse implementano una funzione booleana (mono, multi valore), di variabili booleane. Le reti sequenziali, siano esse asincrone o sincrone, sono invece reti dotate di memoria che ne caratterizza nel tempo lo stato. Lo stato contribuisce, unitamente agli ingressi, alla definizione dei valori delle uscite e dello stato successivo.**

Gli elementi di memoria sono stati realizzati nel tempo a partire dalle tecnologie abilitanti di cui si poteva disporre e con cui realizzare l'elemento base che, come detto, è un elemento bistabile. Flip-flop e latch rappresentano, ad oggi, una soluzione dominante per la sintesi di tali tipi di rete e sono diffusamente usati in tutti i progetti di sintesi di apparati digitali.

2 Bistabili: latch e flip-flop

I circuiti bistabili rappresentano uno dei componenti base per la realizzazione dei latch e dei flip-flop. I bistabili fanno parte della famiglia di circuiti noti come multivibratori. Essi sono circuiti elettronici retroazionati e possono essere di tipo monostabile, astabile e bistabile.

I *multivibratori monostabili* hanno un'uscita che può assumere o un valore di riposo detto "neutro" o un valore "attivo". Il valore attivo dura per una costante di tempo predefinita e caratteristica dello specifico dispositivo. Di fatto il monostabile si comporta da timer, esso mantiene lo stato attivo per un tempo preassegnato e poi ritorna allo stato di riposo dove permane finché un segnale di controllo in ingresso non lo faccia ricommutare allo stato attivo.

I *multivibratori astabili* sono circuiti elettronici a due stati, ma con una uscita. Essi sono progettati in modo tale da permanere in ciascuno dei due stati per un tempo preassegnato. Tali circuiti sono, pertanto, degli oscillatori il cui periodo e il cui duty cycle sono determinati dalle due costanti di permanenza allo stato 1 e a quello 0. I valori di tali costanti di tempo dipendono dal valore di elementi resistivi e capacitivi del circuito.

I *multivibratori bistabili* sono idealmente circuiti elettronici che possono permanere solo in uno di due stati stabili. Per ciascuno stato le uscite assumono valori (complementari nel senso che se un'uscita assume un potenziale alto, l'altra ne assume quello basso e viceversa) che mantengono indefinitamente fino a che un nuovo segnale di comando ne forza il cambiamento di stato. Uno o più linee di ingresso permettono, quindi, di forzare l'equilibrio raggiunto portando il circuito a operare fra uno stato stabile e l'altro. Tale dispositivo si comporta quindi come una cella di memoria di un bit e per la sua importanza sarà trattato nel dettaglio di seguito.

2.1 Il Multivibratore Bistabile

Il multivibratore bistabile I circuiti bistabili sono realizzati mediante una cella base a due transistori. Il funzionamento di principio del circuito può essere spiegato con riferimento alla fig.2.1. Il dispositivo presenta due ingressi indicati con A e B e due uscite Q e \overline{Q} . Un interruttore bipolare, collega l'uno o l'altro degli ingressi collegati alle basi dei transistor a massa. Quando l'interruttore è connesso al punto A, la base del transistor TR1 è connessa a massa e il transistor è in interdizione e la sua uscita Q si porta al valore V_{cc} (che in logica positiva rappresenta lo stato 1 logico). Di contro il transistor TR2, polarizzato direttamente, si porta in saturazione e presenta all'uscita \overline{Q} un valore di tensione (V_{cesat}) prossimo a 0 Volt (associato allo stato 0 logico). Viceversa, quando l'interruttore è spostato sulla posizione B, TR2 va in interdizione e TR1 in conduzione, commutando l'uscita Q allo stato 0 e la \overline{Q} a 1. Le uscite Q e \overline{Q} , nelle due situazioni di

2 Bistabili: latch e flip-flop

stabilità, sono l'una negata dell'altra. Durante il transitorio (si veda fig.2.1), a seguito di una commutazione di breve durata τ , le uscite, con continuità, si portano da un valore all'altro, passando per un punto di equilibrio instabile (o metastabile), in cui le relative tensioni sono uguali. Tale punto è critico e se non controllato potrebbe portare a forme di indeterminazione delle uscite con successivi malfunzionamenti di un circuito.

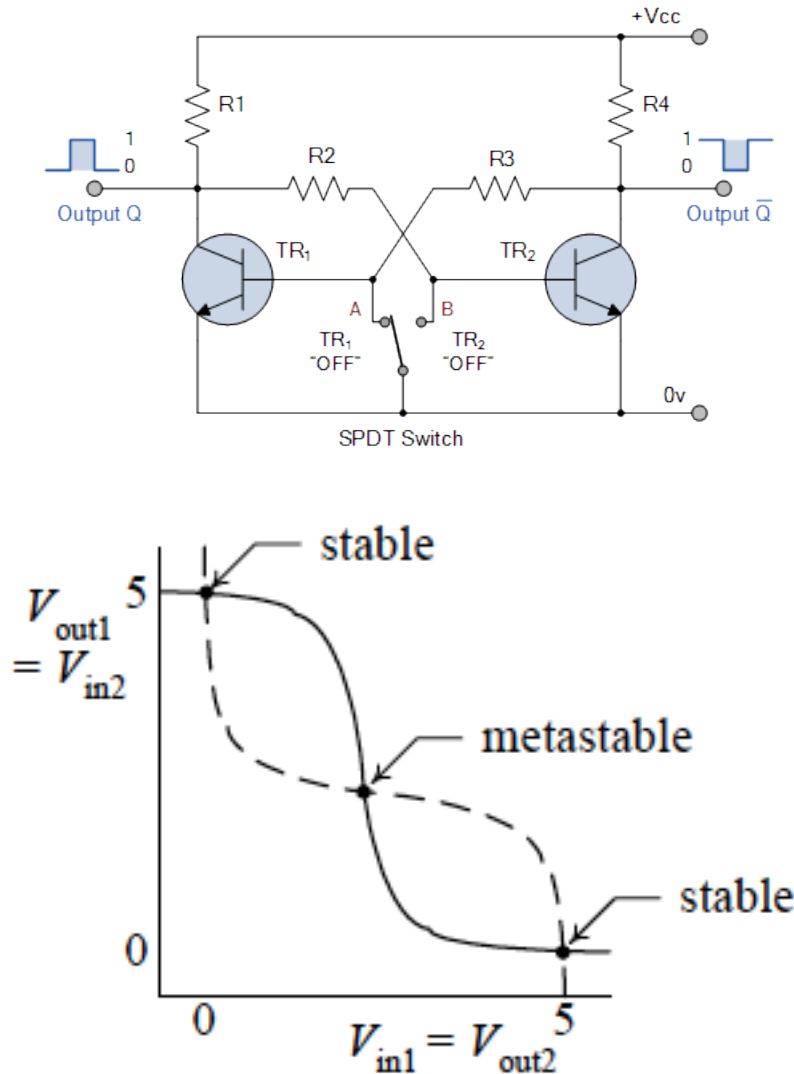


Figura 2.1: Circuito multivibratore bistabile e stati delle uscite

Come si può notare sempre dalla fig.2.1, il bistabile è realizzato controreazionando due invertitori (ciascuno dei due transistor in figura opera da invertitore).

2.2 Bistabili come elementi di memoria

Un elemento di memoria svolge nel tempo la funzione di contenitore di un dato binario (bit). Tale elemento può operare in modo statico o dinamico. Nel modo statico, durante il periodo di funzionamento, la memoria è non volatile (in talune tecnologie, come quella delle memorie magnetiche, la non volatilità resta anche a dispositivo non alimentato e la memoria si dice essere persistente), nel modo dinamico la memoria è volatile nel senso che perde il suo valore nel tempo e si richiede il ricorso a operazioni cicliche di rinfresco del valore (refresh) per mantenerlo.

I primi elementi di memoria sono stati realizzati ricorrendo a tecniche di memorizzazione dei dati dinamiche e basate sull'uso di linee elettriche di ritardo. In esse, la memorizzazione di un bit è effettuata sfruttando l'effetto di persistenza, dovuto alla propagazione del segnale, del valore a esso associato. Una volta immesso un segnale in una linea elettrica di lunghezza L opportuna, questo si propaga nella linea in un tempo di ritardo τ (la propagazione avviene alla velocità della luce nel mezzo e il cui valore dipende dalla lunghezza e dalle proprietà elettriche della linea stessa, ad es. 0,8c). Dopo tale tempo, terminata la propagazione, il segnale raggiunge l'altro estremo della linea posto a una lunghezza L , riflettendosi indietro e subendo effetti di attenuazione per gli effetti resistivi e di degrado per gli effetti capacitivi e induttivi della linea stessa. Occorre pertanto, mediante un apposito meccanismo (realizzato con un circuito elettronico che provvede a rigenerarne forma e livelli elettrici), procedere alla sua reimmissione (una sorta di refresh) nella linea per compiere un nuovo ciclo che, se ripetuto, ne garantisce il mantenimento del valore nel tempo. Tale meccanismo di refresh provvede a ripristinare e mantenere, quindi, nel tempo il valore del segnale memorizzato.

Al posto di una linea di ritardo si può utilizzare, mantenendo lo stesso meccanismo di memorizzazione del dato, un buffer (fig.2.2), avente un ritardo di propagazione (o di trasporto) t_{pd} e in cui l'uscita è collegata in retroazione con l'ingresso. Una volta immesso in esso un valore 0 o 1, questo resterà memorizzato a causa dell'effetto di rigenerazione dinamico che ciclicamente propaga il segnale fra l'uscita e l'ingresso. Lo stato binario è memorizzato, quindi, in cicli di durata t_{pd} , il tempo impiegato dal segnale per propagarsi fra l'ingresso e l'uscita del circuito (si fa osservare che allo stato corrente della tecnologia dei componenti, il ritardo t_{pd} è dell'ordine di 1 ns o meno).

Ovviamente, tale soluzione è di principio e di scarsa utilità sia per la difficoltà che si incontra nel forzare un nuovo valore da memorizzare nel buffer sia per i consumi del circuito che, per mantenere memorizzato un bit, deve essere mantenuto sempre alimentato.

Per la realizzazione di una cella di memoria elementare si preferisce, pertanto, ricorrere ai circuiti bistabili, realizzati come visto con circuiti elettronici detti multivibratori bistabili o con circuiti elettronici a riempimento-svuotamento di carica (come ad esempio nelle celle a transistor MOS, in cui lo stato booleano è associato alla presenza o meno di una certa quantità di carica elettrica in un dispositivo).

Un dispositivo bistabile realizza, quindi, fintantochè il suo circuito è alimentato e funzionante, una memoria di un bit, mantenendo stabili nel tempo i valori elettrici associati

2 Bistabili: latch e flip-flop

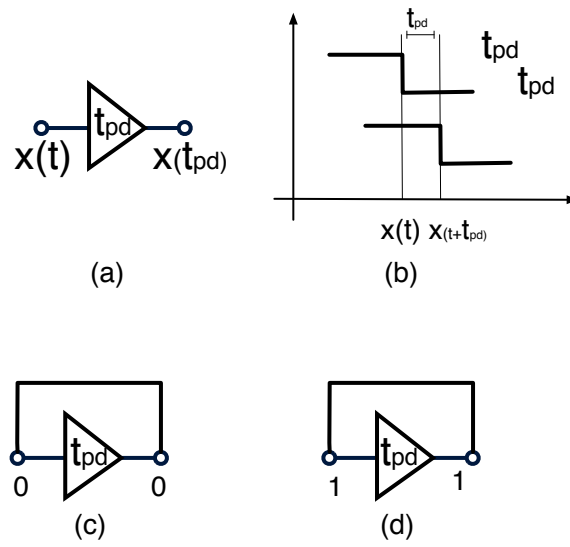


Figura 2.2: Buffer con ritardo per memorizzare valori booleani

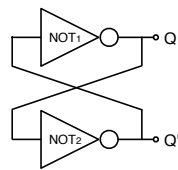


Figura 2.3: Circuito bistabile

2 Bistabili: latch e flip-flop

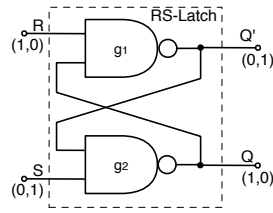


Figura 2.4: Bistabile realizzato con due porte NAND

a ciascuno dei suoi due stati logici.

La fig.2.3, mostra un bistabile di principio realizzato logicamente mediante due inverter (che operano da amplificatori invertitori). Le due uscite in tensione V_H e V_L ne rappresentano rispettivamente gli stati logici 1 e 0.

Il funzionamento del bistabile può essere descritto concettualmente come segue: supponendo che al tempo t_0 sia $Q=V_H$, essendo Q collegato all'ingresso dell'inverter NOT₂, l'uscita da tale inverter si porterà al valore $Q'=V_L$ che, a sua volta, reazionato all'ingresso del NOT₁, genererà sulla sua uscita il valore $Q=V_H$, chiudendo così l'anello di reazione in modo consistente fra i valori degli ingressi e delle uscite e soddisfacendo il vincolo del flip-flop $Q=\text{Not } Q'$.

Il bistabile sopra descritto è dotato sole delle due uscite Q e Q' , e non possiede ingressi. Esso è solo di principio in quanto in tal caso i valori di Q e Q' manterrebbero per tutto il periodo di funzionamento l'uno o l'altro dei due valori di regime raggiunto in modo non determinabile dopo il transitorio che segue l'alimentazione del circuito. Per rendere concretamente utilizzabile tale bistabile, occorre inserire due ulteriori linee di ingresso tramite cui poter controllare i valori delle uscite e, quindi, la commutazione dello stato.

Ciò può ottenersi sostituendo i due inverter con due porte NAND o con due porte NOR così come illustrato in fig.2.4. Se si usano le porte NAND, i segnali di input, per rispettare le equazioni logiche del flip-flop, devono operare in logica negativa se NOR in logica positiva (per chiarezza si indicheranno con R , S gli ingressi codificati in logica 1-attiva e con R' , S' , quelli in logica 0-attiva).

2.2.1 Il bistabile visto come rete asincrona

Il circuito elettronico bistabile può essere anche studiato come rete asincrona, operante secondo il modello fondamentale di Huffman (per tale motivo esso è noto come latch RS fondamentale). Per tale macchina, gli ingressi in grado di produrre il cambiamento di stato sono detti di Set (S), utilizzato per forzare lo stato Q a 1, e di Reset (R), per forzarlo a 0. Se essi assumono contemporaneamente il valore neutro (il valore non attivo), lo stato resta invariato, se assumono invece il valore attivo (condizione logica proibita dalla specifica del latch RS), le uscite del bistabile assumono entrambe il medesimo valore V_H , se esso è realizzato con porte NOR, o il valore V_L se con NAND, violando quindi la condizione logica $Q = \text{Not } Q'$. Volendo impedire tale anomalia occorre garantire che gli

2 Bistabili: latch e flip-flop

ingressi S e R soddisfino il vincolo $SR=0$ e, quindi, che non siano mai simultaneamente attivi.

Il bistabile, memoria elementare di 1 bit, rappresenta l'elemento base per la sintesi di componenti di memoria più complessi realizzati come macchine sincrone o asincrone.

Essi sono noti come *latch* e *flip-flop*. Spesso tali due termini sono utilizzati nei testi scientifici e tecnici come sinonimi, ma si preferisce, seguendo la linea di gran parte degli autori, indicare con *latch* i bistabili che, se abilitati, commutano il loro stato al variare degli ingressi con continuità e restano insensibili alle variazioni degli ingressi quando disabilitati; con *flip-flop* i bistabili che commutano lo stato solo in ben precisi istanti di tempo, coincidenti con il fronte di salita o di discesa di un segnale di sincronizzazione esterno (clock), e restano inattivi quando l'abilitazione assume il valore a livello 0 o 1.

La principale differenza fra latch e flip-flop sta quindi nel fatto che in un latch abilitato a operare lo stato evolve con l'evoluzione degli ingressi, esso cambia il suo stato al variare dei suoi ingressi e, quando disabilitato, mantiene l'ultimo stato memorizzato. Di contro, un flip-flop cambia stato soltanto sul fronte attivo del suo segnale di abilitazione, e cioè in un preciso istante di tempo e non in una finestra temporale, così come avviene per il latch. Se il fronte è quello caratterizzato dalla variazione $0 \rightarrow 1$ di un segnale di *Enable*, il flip-flop è detto sensibile (o attivo) sul fronte di salita (rising edge), se quello caratterizzato dalla variazione $1 \rightarrow 0$, è detto sensibile al fronte di discesa (falling edge). Passato il fronte attivo, il flip-flop diventa insensibile a qualunque variazione dei suoi ingressi e mantiene costante il suo stato.

Come rete sequenziale asincrona, il funzionamento di un bistabile può essere descritto dalla:

- *equazione caratteristica*, che esprime lo stato prossimo in funzione dello stato corrente e dello stato degli ingressi;
- *tabella delle transizioni di stato*, che riporta nelle righe lo stato corrente, nelle colonne gli ingressi e negli incroci lo stato prossimo;
- *tabella invertita in forma compatta* che riporta nelle righe gli ingressi e nelle colonne lo stato prossimo;
- *tabella delle eccitazioni* che riporta nelle righe le combinazioni possibili di stato corrente-stato prossimo e nelle colonne i valori degli ingressi che permettono la specifica transizione di stato.

Si fa osservare, infine, che per la sintesi di reti sequenziali sincrone, è preferibile utilizzare, come elementi di memoria nelle linee di retroazione, i flip-flop e non i latch, in quanto essi consentono di sincronizzare il trasferimento di un nuovo stato in un ben preciso istante di tempo, coincidente con uno dei due fronti di un segnale di abilitazione (che se regolare è detto clock).

2.3 Registri k-stabili

In generale un registro è un dispositivo atto a memorizzare dati D di un assegnato tipo T. La cardinalità k del registro è data dalla cardinalità k del tipo T.

Ad esempio un dato D di tipo intero nell'intervallo $\{0..9999\}$ avrà **ua** cardinalità $k=10000$, essendo 10000 i valori che D può assumere in esso (i valori interi memorizzabili). Al fine di memorizzare un dato D in un registro, questo deve poter essere rappresentato fisicamente in esso, occorre pertanto una codifica dei valori che D può assumere.

Un registro k-stabile è organizzato come un array di n celle elementari in grado di memorizzare un digit di b valori, con b la base. Esso è una macchina sequenziale che per un registro con k stati $q_0..q_{k-1}$ sarà composto da $n = \lceil \log_b k \rceil$ celle, essendo b la base di rappresentazione di ciascun digit nella cella. Se il registro è strutturato in celle contenenti 4 digit decimali (ogni cella è di fatto un registro elementare 10-stabile), il registro permetterà di memorizzare $k=10^4$ configurazioni differenti del numero decimale D_{10} . Analogamente se si opera con un registro composto da n celle binarie, la cardinalità sarà $k=2^n$ e ciascuna cella sarà costituita da un bistabile.

Registro binario è, pertanto, un dispositivo composto da un array finito di n celle bistabili avente funzioni di scrittura e lettura simultanea delle n celle. Le operazioni di memorizzazione del dato D sono legate alla tipologia di bistabile di cui è fatta ciascuna cella e caratterizza il tipo di registro con riferimento alla sincronizzazione delle operazioni.

I registri possono essere costituiti da celle di tipo latch o sensibili al fronte flip-flop (ad esempio n D-latch o n D-flip-flop).

2.4 I dispositivi bistabili Latch

I latch sono dispositivi bistabili sintetizzabili come una rete sequenziale asincrona operante in modo fondamentale¹.

Come detto il latch base è il bistabile RS. Da esso è possibile derivare, con l'aggiunta di una logica di controllo, tutti gli altri tipi di latch e flip-flop.

I bistabili, oltre che come latch e flip-flop possono essere in generale classificati come dispositivi a memorizzazione e a commutazione dello stato. I dispositivi a memorizzazione dello stato, mantengono il valore del loro stato (Q e Q') come definito dagli ingressi, quelli a commutazione, cambiano lo stato a causa di un evento in ingresso (se lo stato è Q=1, il successivo sarà Q=0 e viceversa). Inoltre, in base alla tipologia degli ingressi possono anche essere classificati come:

- RS: latch base, ha due ingressi detti di Set (S) e Reset (R). Gli ingressi S=1, R=0 forzano l'uscita a Q=1, mentre S=0, R=1, a Q=0. Gli ingressi S=0, R=0 sono neutri e non generano commutazione di stato, mentre non è ammessa la configurazione S=1, R=1 degli ingressi;

¹e cioè una macchina che ha sequenze di ingresso a livello e vincolati a cambiare uno alla volta e in cui una sola uscita può cambiare a seguito di un passaggio di stato

2 Bistabili: latch e flip-flop

- D: ha un ingresso detto di Delay (D) ; il valore di D si propaga in uscita Q dopo un ritardo noto e caratteristico del latch;
- T: ha un unico segnale di ingresso T (abbreviazione di Toggle che in inglese significa “cambio di stato in un funzionamento) che se attivo fa commutare lo stato Q del latch;
- JK: ha due ingressi detti J (che corrisponde a S per il latch RS) e K (che corrisponde a R). Il latch JK opera come RS, ma a differenza di questo, ammette la configurazione J=1, K=1 degli ingressi, in tal caso esso si comporta da T.
- RS, D, T, JK abilitati: tutti i latch di cui sopra possono avere un segnale di abilitazione solitamente indicato con clock (Clk), essi sono detti latch abilitati.

2.4.1 Analisi e progettazione del latch RS fondamentale

Il latch RS descritto è una rete sequenziale asincrona operante in modo fondamentale e cioè una macchina con sequenze di ingresso a livello in cui gli ingressi sono vincolati a cambiare uno alla volta e in cui una sola uscita può cambiare a seguito di un passaggio di stato. La specifica verbale del latch RS fondamentale è la seguente:

Una macchina asincrona con due ingressi S e R (Set e Reset) e due uscite Q e Q'. L'ingresso attivo S pone lo stato di Q al valore attivo, l'ingresso attivo R al valore non attivo, l'ingresso neutro (00 per i segnali 1-attivi e 11 per quelli 0-attivi) lascia lo stato del latch invariato. C'è il vincolo sulle uscite di essere $Q = \overline{Q'}$, vincolo che si traduce nel vincolo sugli ingressi R ed S di non assumere contemporaneamente il valore attivo. Nel caso di segnali 1-attivi tale vincolo è dato da $R \cdot S = 0$, nel caso di segnali 0-attivi dalla condizione duale $R + S = 1$.

Da tale descrizione funzionale si può dedurre direttamente, riferendosi ad esempio a segnali 1-attivi, l'equazione di stato logica del latch di seguito riportata (si è indicato con Q_p il valore corrente della variabile di stato e con Q il valore successivo):

$$Q = S + \overline{R}Q_p \text{ con vincolo } R \cdot S = 0$$

Tale equazione asserisce che lo stato Q assume il valore vero quando il segnale di set è vero o quando lo stato precedente è vero e il segnale di reset è falso.

In modo analogo, per segnali 0-attivi, Q assume il valore vero quando il segnale di set è falso o quando lo stato precedente è vero e il segnale di reset è vero, così come indicato dall'equazione di stato seguente:

$$Q = S' + RQ_p \text{ con il vincolo } R + S = 1$$

Il latch RS con ingressi 0-attivi, realizzato come si vedrà di seguito con porte NAND, è del tutto analogo nel funzionamento a quello con ingressi 1-attivi, realizzato con porte NOR, per esso, il valore R=0 forza l'uscita Q=0, mentre S=0 la forza a Q=1. La

2 Bistabili: latch e flip-flop

R	S	Q_p	Q
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	-
1	1	1	-

R	S	Q	\overline{Q}
0	0	Q	\overline{Q}
0	1	1	0
1	0	0	1
1	1	-	-

Tabella 2.1: Tabella caratteristica in forma estesa e compatta per il latch RS

	Q_p/RS	00	01	11	10	$Q' Q$
Stato di reset con ingr. 00	s_0	s_0	s_1	-	s'_0	10
Stato di reset con ingr. 10	s'_0	s_0	s_1	-	s'_0	10
Stato di set con ingr 01	s_1	s'_1	s_1	-	s'_0	01
Stato di set con ingr 00	s_1	s'_1	s_1	-	s'_0	01

Tabella 2.2: Tabella primitiva del latch RS

la differenza sta solo nel valore dello stato neutro degli ingressi ($R=S=0$ per 1-attivo e $R=S=1$ per 0-attivo) e nei valori che esprimono il vincolo sui valori non ammessi ($R=S=1$ per 1-attivo e $R=S=0$ per 0-attivo). Tale comportamento è il duale del latch realizzato a porte NOR.

Il latch, rappresentato come macchina di Moore, è descritto dalla tabella caratteristica estesa o da quella compatta (tab.2.1), dalla tabella primitiva (2.2) e dalla tabella delle transizioni di stato (2.3) utilizzata per la sintesi delle rete.

Stante la semplicità del circuito, una prima sintesi diretta può essere effettuata a partire dalla tabella delle transizioni di stato ridotta di tab.2.3 o dalle equazioni logiche del latch, che porta alla realizzazione del cosiddetto latch dinamico in cui lo stato è codificato con un'unica variabile. Tale tipo di latch ha il vantaggio di non presentare alee combinatorie, e quindi corse, ma presenta lo svantaggio, avendo un solo segnale di stato-uscita Q , di dover derivare quello complementare Q' da Q mediante l'uso di un inverter, cosa questa che rende asimmetrica l'architettura del latch e che ritarda la presentazione del nuovo valore di Q' rispetto a Q . Tale soluzione è pertanto solo di principio e raramente utilizzata.

La sintesi del latch RS può essere fatta sistematicamente, utilizzando, a partire dalle tabelle di transizione di stato, il classico procedimento di sintesi delle macchine asincrone di Paull-Hunger che prevede la minimizzazione dello spazio degli stati e un'appropriata codifica dei segnali di ingresso, stato e uscita.

Si fa osservare la natura *asincrona impulsiva* di tale macchina. Essa, infatti, opera

2 Bistabili: latch e flip-flop

Q_p/RS	00	01	11	10	Q Q'
S_0	S_0	S_1	-	S_0	01
S_1	S_1	S_1	-	S_0	10

Tabella 2.3: Tabella delle transizioni di stato del latch RS fondamentale

Q_p/RS	00	01	11	10	Q,Q'
S_0	S_0	S	-	S_0	10
S	-	S_1	-	S_0	00
S_1	S_1	S_1	-	S	01

 \Rightarrow

y_1y_2/RS	00	01	11	10
10	10	00	-	10
00	-	01	-	10
01	01	01	-	00

Figura 2.5: Tabella delle transizioni e tabella in codice con aggiunta dello stato instabile S per eliminare la corsa (caso di un latch a NOR)

solo con sequenze di ingresso intervallate dallo stato neutro. Ad esempio, nel caso del latch a NAND, il cui stato neutro è dato da $R=S=1$, si possono presentare sequenze del tipo “11-10-11-01...”. Come detto non è ammessa la sequenza proibita $R=S=0$ che, se applicata, porterebbe a cambiamenti di stato non definiti.

Onde pervenire alla realizzazione del latch simmetrico con le due uscite Q e Q', occorre utilizzare una codifica con due variabili di stato, y_1 e y_2 , cosa che porta, come si vedrà, alla realizzazione del latch con due porte di tipo NOR (se gli ingressi R e S sono 1-attivi) o di tipo NAND (se gli ingressi sono 0-attivi).

In particolare, la codifica dello stato S_0 con $y_1=1$, $y_2=0$ e dello stato S_1 con $y_1=0$, $y_2=1$, se garantisce che le variabili di stato y_1 e y_2 coincidano con quelle delle uscite Q e Q', nel contempo genera una corsa nella transizione da uno stato all'altro, stante la presenza dell'alea $01 \leftrightarrow 10$ evidenziata in tabella e dal cambiamento simultaneo dei codici delle due variabili di stato da 10 a 01 o viceversa.

Per eliminare tale alea, e l'eventuale corsa che ne deriverebbe, si introduce un nuovo stato *instabile*, denotato con S, che può essere codificato con una delle due configurazioni libere $y_1=0$ e $y_2=0$ oppure $y_1=1$ e $y_2=1$. Se si codifica con $y_1=0$ e $y_2=0$ si perviene alla sintesi di un latch a NOR, se con $y_1=1$ e $y_2=1$, di un latch a NAND, così come di seguito mostrato:

Dalla tabella in codice di fig.2.5 si derivano le due mappe di Karnaugh per y_1 e y_2 di cui alla fig.2.6. La sintesi della rete combinatoria è effettuata a partire da tali mappe, selezionando per y_1 e per y_2 i primi implicant marcati in grassetto e da cui deriva, come detto, la sintesi del latch 1-attivo con porte NOR (si veda fig.2.9).

Se si codifica, invece, lo stato instabile con $y_1=1$ e $y_2=1$, si perviene, in modo analogo, alle seguenti equazioni che portano alla sintesi del latch 0-attivo con porte NAND (2.7 e 2.8):

2 Bistabili: latch e flip-flop

$y_1 \Rightarrow$	$y_1 y_2 / RS$	00	01	11	10
	00	-		-	1
	01			-	
	11	-	-	-	-
	10	1		-	1

$y_2 \Rightarrow$	$y_1 y_2 / RS$	00	01	11	10
	00	-	1	-	
	01	1	1	-	
	11	-	-	-	-
	10			-	

$$Y_1 = \overline{S} \cdot \overline{y_2} = S \downarrow y_2$$

$$Y_2 = \overline{R} \cdot \overline{y_1} = R \downarrow y_1$$

Figura 2.6: Mappe di Karnaugh per y_1 e y_2 (sintesi con NOR)

Q_p / RS	00	01	11	10	Q, Q'
S_0	-	S	S_0	S_0	10
S	-	S_1	-	S_0	00
S_1	-	S_1	S_1	S	01

 \Rightarrow

$y_1 y_2 / RS$	00	01	11	10
10	-	11	10	10
11	-	01	-	10
01	-	01	01	11

Figura 2.7: Tabella delle transizioni e tabella in codice con incluso lo stato instabile per eliminare la corsa (latch a NAND)

$y_1 \Rightarrow$	$y_1 y_2 / RS$	00	01	11	10
	00	-	-	-	-
	01	-			1
	11	-		-	1
	10	-	1	1	1

$y_2 \Rightarrow$	$y_1 y_2 / RS$	00	01	11	10
	00	-	-	-	-
	01	-	1	1	1
	11	-	1	-	
	10	-	1		

$$Y_1 = \overline{y_2} + \overline{S} = y_2 \uparrow S$$

$$Y_2 = \overline{y_1} + \overline{R} = y_1 \uparrow R$$

Figura 2.8: Mappe di Karnaugh per y_1 e y_2 (sintesi con NAND)

2 Bistabili: latch e flip-flop

$Q_p \backslash SR$	00	01	11	10	$Q_p \backslash SR$	00	01	11	10
0	0	0	-	1	0	-	1	0	0
1	1	0	-	1	1	-	1	1	0

Tabella 2.4: Mappa di Karnaugh per il calcolo dell'equazione caratteristiche del latch RS con ingressi 1-attivi (a sinistra) e 0-attivi (a destra)

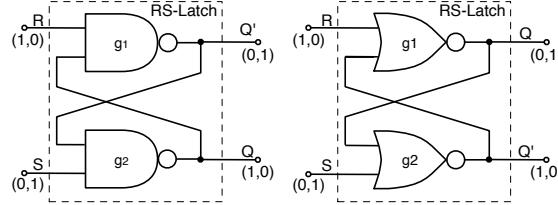


Figura 2.9: Latch RS simmetrico

Stante la semplicità di tale latch, se ne può effettuare la sintesi per tramite delle sue *equazioni caratteristiche* (sia per ingressi 1-attivi sia 0-attivi) ottenute minimizzando le mappe di Karnaugh di tab.2.4 derivate dalla tabella in codice riportate in fig.2.7:

$$Q = S + \overline{R}Q_p \text{ con vincolo } R \cdot S = 0 \text{ per la 1-attiva e}$$

$$Q = \overline{S} + RQ_p \text{ con il vincolo } R + S = 1 \text{ per la 0-attiva}$$

Da tali espressioni, dopo una semplice manipolazione algebrica, si ottengono le seguenti due forme in soli NOR e in soli NAND corrispondenti ai circuiti del latch di fig.2.9:

$$Q = S + \overline{R}Q_p = \overline{\overline{S} + \overline{R}Q_p} = \overline{\overline{S} \cdot \overline{RQ_p}} = S \downarrow (R \downarrow \overline{Q_p}) \text{ con vincolo } R \cdot S = 0$$

$$Q = \overline{S} + RQ_p = \overline{\overline{\overline{S} + RQ_p}} = \overline{\overline{S} \cdot \overline{RQ_p}} = S \uparrow (R \uparrow Q_p) \text{ con il vincolo } R + S = 1$$

Come si può osservare, le espressioni finali sono, ovviamente, identiche a quelle derivate precedentemente con la procedura di sintesi.

Si fa osservare che il vincolo $R=S=1$ per i latch 1-attivi e $R=S=0$ per quelli 0-attivi presente nella specifica del latch RS, deve essere posto e rispettato dai circuiti a monte che definiscono i valori di R e S nel circuito in cui il latch è inserito. Infatti, se gli ingressi R e S assumessero uno stato proibito, il circuito, operando in modo improprio, forzerebbe lo stato delle uscite a uno stesso valore (0 o 1). In particolare, con riferimento a un latch a NAND (0-attivo) con $R=S=1$ come stato neutro, ponendo gli ingressi $R=S=0$, lo stato delle uscite si porterebbe a $Q=Q'=1$, mentre per una realizzazione a NOR, con $R=S=0$ come stato neutro, si avrebbe, a fronte di $R=S=1$ (1-attivo), uno stato delle uscite forzato a $Q=Q'=0$.

Tale condizione, non ammessa, potrebbe generare una condizione di errore nei circuiti a valle, allorquando RS si riportano allo stato neutro a causa dei ritardi nelle linee di

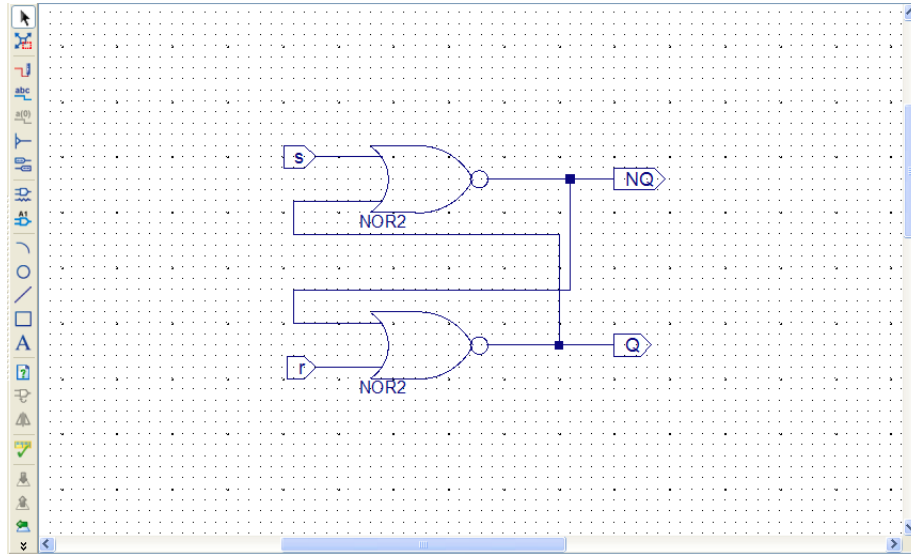


Figura 2.10: Schematico del latch RS

retroazione (che per un latch fisicamente realizzato sono certamente fra loro differenti). In un latch ideale simulato, dove tali ritardi possono supporre essere uguali, il passaggio dalla condizione proibita allo stato neutro genererebbe un fenomeno oscillatorio delle uscite del latch, con una sequenza di transizione 00->11 o 11->00.

Per verificare concretamente tale comportamento, sono riportati di seguito i risultati della simulazione di un latch RS². Il latch è stato disegnato mediante lo schematico disponibile nell'ambiente ISE di Xilinx e poi simulato in modo behavioral e nei modi *non-behavioral post-translation, post-mapping, post-place-and-route*.

I dettagli di tali modi di simulazione sono riportati in appendice B. Il test bench per verificare il latch fornisce agli ingressi q e nq uno specifico pattern per ciascuno di essi. Tale pattern è generato mediante il seguente processo in VHDL:

```
tb : PROCESS
BEGIN
    s <= '0', '1' after 20 ns, '0' after 40 ns, '1' after 65 ns,
        '0' after 80 ns, '1' after 100 ns;
    r <= '0', '1' after 45 ns, '0' after 80 ns, '1' after 90 ns;
    WAIT; -- attesa infinita

END PROCESS;
```

In fig.2.10 è riportato lo schematico del latch disegnato con l'apposito editor dell'ambiente ISE.

Nella simulazione behavioral, fig.2.11, le porte logiche sono considerate ideali e istantanee nel loro funzionamento per cui passando dalla condizione R=S=1 a quella R=S=0

²la simulazione è stata effettuata con ISIM, il simulatore fornito con l'ambiente Xilinx ISE utilizzato nel laboratorio di ASE

2 Bistabili: latch e flip-flop

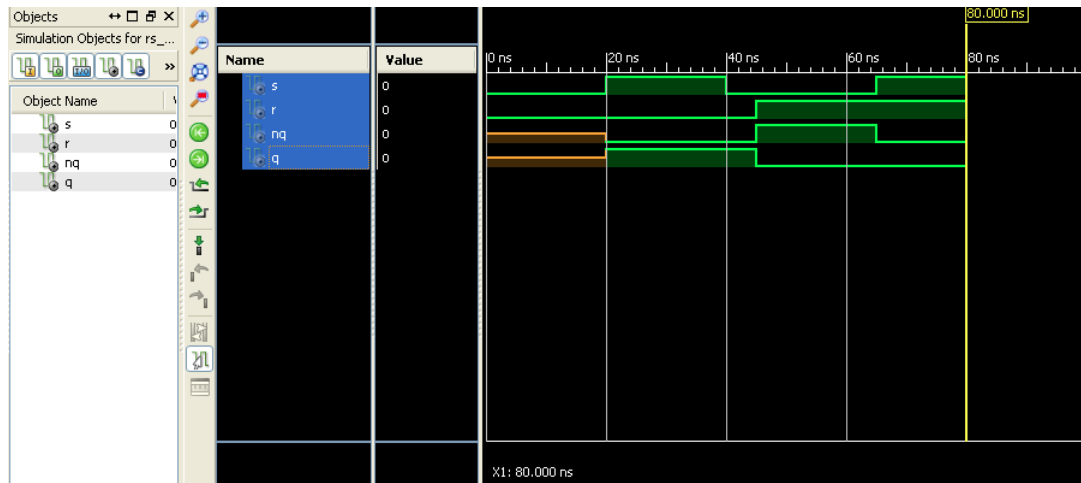


Figura 2.11: Simulazione del latch RS Behavioural

si generano eventi oscillatori tutti posti all'istante di commutazione dei due segnali che producono equivalenti cicli di delta cycle. Tali cicli non sono rappresentabili dal simulatore in quanto il tempo di simulazione per essi non avanza. Ciò, quindi, genera una segnalazione di errore del simulatore per raggiunto limite di iterazioni³

La simulazione post-translate, introduce nella descrizione del circuito da simulare componenti logici di una libreria Xilinx ma sempre ideali nei tempi di commutazione considerati istantanei. I diagramma di fig.2.12 è pertanto uguale a quello precedente e si manifesta, ovviamente, lo stesso errore di raggiungimento del limite di iterazioni delta cycle.

La simulazione post-map, di cui alla fig.2.13, avviene dopo che nel processo di sintesi si sono inseriti componenti logici con gli associati ritardi ma senza tenere conto dei ritardi delle connessioni fra essi considerate istantanee. Tale situazione in cui è mantenuta la simmetria delle porte è l'unica che permette di evidenziare nel diagramma temporale prodotto dal simulatore il fenomeno delle oscillazioni sulle uscite all'atto della transizione degli ingressi R e S a 0 logico. Si fa notare che tale caso è ideale e si manifesta in quanto la rete di connessione fra i componenti opera a ritardo nullo.

La simulazione post-route, è la più dettagliata e vicina a quello che è il comportamento reale del circuito sintetizzato sul dispositivo FPGA selezionato. In essa si tiene anche conto dei ritardi delle connessioni dipendenti dallo specifico routing effettuato nella fase di mapping tecnologico. Ciò inserisce certamente delle asimmetrie nel circuito che de-

³ERROR: at 80 ns(10000): Iteration limit 10000 is reached. Possible zero delay oscillation detected where simulation can not advance in time because signals can not resolve to a stable value in File "E:/Ambienti-ASE/flip-flop/Latch_SR/latch_rs_nor/latch_rs_nor/RS_nor.vhf" Line 48. Please correct this code in order to advance past the current simulation time.

2 Bistabili: latch e flip-flop

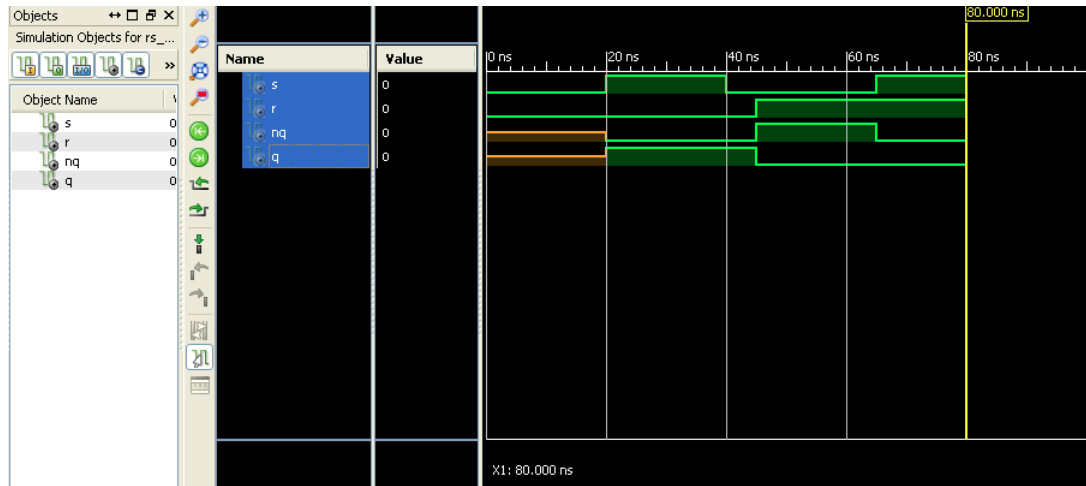


Figura 2.12: Simulazione del latch RS Post-Translate model

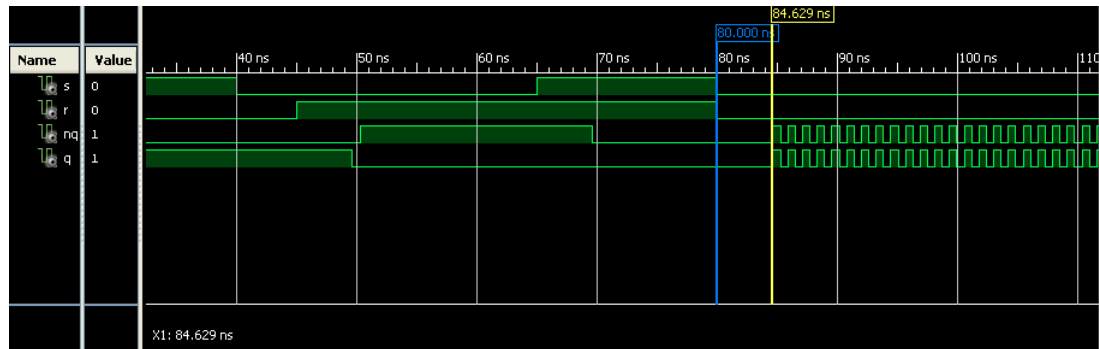


Figura 2.13: Simulazione del latch RS Post-Map

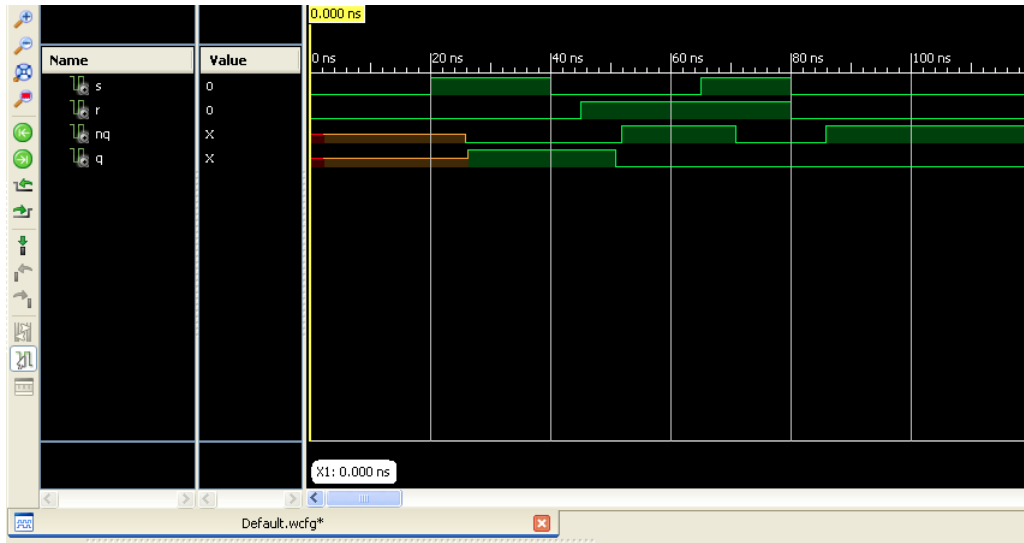


Figura 2.14: Simulazione del latch RS Post-Route

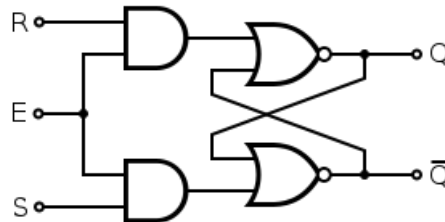


Figura 2.15: Latch RS abilitato

terminano lo stato delle uscite al raggiungimento dello stato $R=S=0$. In fig.2.14 si può, infatti, notare che lo stato delle uscite si pone a $q=0$ e $nq=1$.

2.4.2 Latch fondamentale RS abilitato

Il latch RS abilitato è derivato dal latch RS di cui ne eredita il comportamento quando il segnale di abilitazione E (Enable) è attivo, mentre conserva l'ultimo stato memorizzato quando E è neutro (in tale stato risulta insensibile alle variazioni degli ingressi R e S). In fig.2.15 il valore attivo di E è quello alto, mentre il valore basso, forza le uscite delle due porte And a 0-logico, valore neutro questo per il latch RS a valle del circuito. La progettazione di tale latch può essere effettuata considerando tale sistema composto da un latch RS fondamentale e da una rete combinatoria di abilitazione composta dalle due porte AND e dal segnale abilitante E.

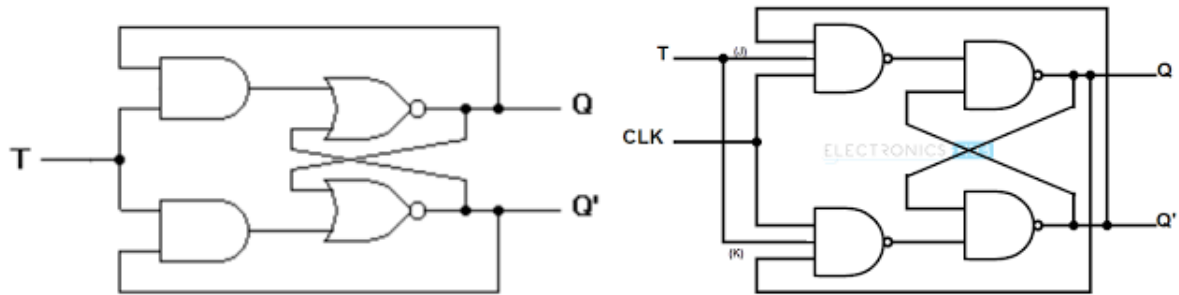


Figura 2.16: Latch T fondamentale, T abilitato e T derivato da latch D

2.4.3 Latch a commutazione trigger T

Il latch T è un latch a commutazione, comunemente usato come circuito divisore di frequenza e impiegato nella sintesi dei contatori. In tal caso, la versione correntemente utilizzata introduce un secondo ingresso detto di clock (clk) posto in AND con l'ingresso T. Se l'ingresso $T=0$, il latch trigger mantiene lo stato precedente, se $T=1$, il latch segue il valore dell'ingresso clk che lo fa commutare, per cui l'eventuale segnale di clk in ingresso è riportato in uscita a frequenza dimezzata. In fig.2.16 sono riportati due schemi di tale dispositivo, quello di base e quello abilitato da un ingresso di clk. Il dispositivo funziona con segnali a livello.

Come macchina asincrona operante in modo fondamentale, il latch T può essere sintetizzato a partire da un latch RS base e da una rete logica combinatoria posta a monte. Il latch RS assolve le funzioni di elemento di memorizzazione, mentre la rete combinatoria a monte, con tre ingressi (Q , Q' e T) e due uscite (R ed S) composta dalle due porte AND, di rete di controllo della commutazione (è la reazione in ingresso delle uscite Q e Q' che forza la commutazione dello stato). Il valore neutro di T lascia invariato lo stato del latch (nel caso riportato in figura di latch a NOR, lo stato neutro è lo 0 logico per cui il valore $T=0$ genera la condizione $R=S=0$, valore neutro per il latch RS posto valle). Il valore attivo (il valore 1 per la porta NOR), abilitando la retroazione dell'uscita Q sull'ingresso R e dell'uscita negata Q' sull'ingresso K , fa commutare il latch, sempre che si rispettino i vincoli della macchina fondamentale asincrona che opera in tal caso come macchina impulsiva (0-1-0). La durata dell'impulso (e cioè il tempo di permanenza nello stato attivo di T) deve essere compresa fra un T_{min} e un T_{max} . T_{min} deve essere sufficientemente grande per consentire al circuito di commutare, T_{max} sufficientemente piccolo per evitare effetti di ulteriori commutazioni che, per tale tipo di reti, portano a un'instabilità che si manifesta con un comportamento oscillatorio delle due uscite Q e Q_n .

L'equazione caratteristica del latch T è data da: $Q = T'Qp + TQp' = T \oplus Qp$, con Q lo stato prossimo e con Qp lo stato precedente.

La forma $Q=T \oplus Qp$ permette di sintetizzare il latch T a partire dal un D latch (che memorizza lo stato Qp) a cui anteporre una rete combinatoria costituita da un semplice

2 Bistabili: latch e flip-flop

Tabella di verità di JK				
J	K	Q_{n+1}		
0	0	Q_n		
0	1	0		
1	0	1		
1	1	Q'_n		

Mappa di Karnaugh per Q_{n+1}				
JK/ Q_n	00	01	11	10
0			1	1
1	1			1

$$Q_{n+1} = JQ'_n + K'Q_n$$

Figura 2.17: Equazione caratteristica del latch J-K

OR-Esclusivo.

2.4.4 Latch JK

Il latch JK fondamentale è simile nel funzionamento a quello SR ove J corrisponde a S e K a R. Esso, quando si pone $J=K=1$, si comporta però da latch trigger (si ricorda che per il latch RS a Nor, $R=S=1$ è, invece, la configurazione non ammessa).

L'equazione caratteristica può derivarsi dalla tabella

Il latch JK è molto simile come architettura al latch T. Così come mostrato in fig.2.18 esso, al posto di un unico segnale T di ingresso collegato alle due porte AND, presenta due ingressi separati di J e K ciascuno connesso con una sola porta AND. Analogamente al latch T, sulle due porte AND sono reazionate le due uscite Q e Q' del latch RS (Q verso l'ingresso di reset K e Q' verso quello di set J). A destra della fig.2.18 è anche riportata la versione che fa uso di un segnale di abilitazione clk. Il segnale clk, quando attivo (e cioè allo stato alto per la porta AND di cui alla figura) abilita la commutazione del latch, quando neutro, la disabilita e si permane nello stato precedente.

In fig.2.19 è mostrata una variante del JK con clock che fa uso di un latch RS abilitato da clk, in tal caso il latch JK utilizza in ingresso porte NAND senza il clk, così come invece riportato nella fig.2.18.

I bistabili JK commerciali, hanno anche due ingressi detti di Preset e Clear, come mostrato in fig.2.20, utilizzati per inizializzare il dispositivo forzandolo in uno dei due stati possibili.

2.4.4.1 Simulazione del latch J-K

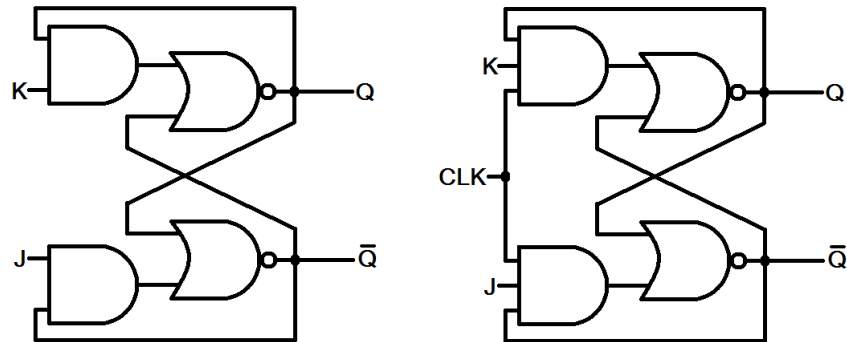


Figura 2.18: Latch JK fondamentale e JK abilitato

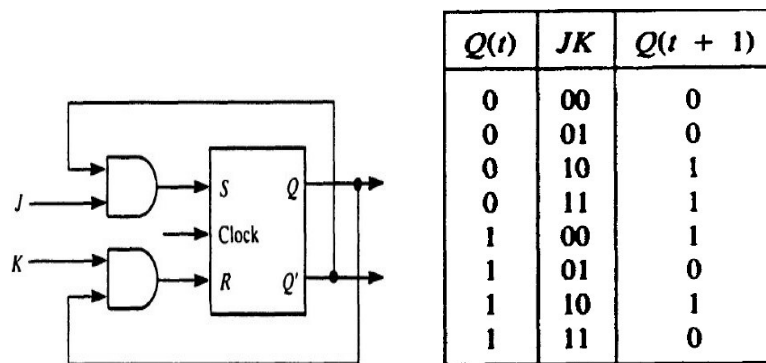


Figura 2.19: Schema di un bistabile JK

2 Bistabili: latch e flip-flop

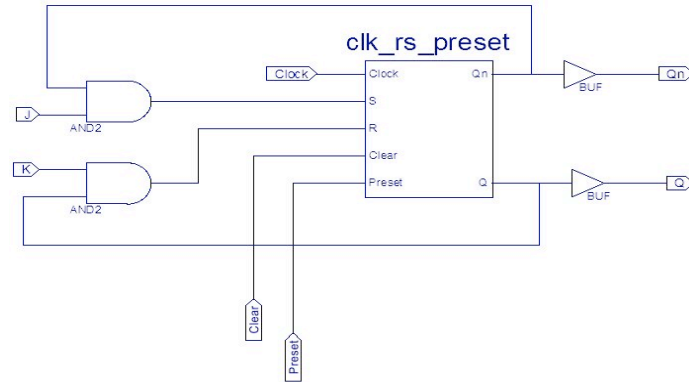


Figura 2.20: Schema di un flip-flop JK commerciale derivato da un RS

```
Preset <= '1','0' after 5 ns;
Clear <= '0';
Clock <= '1','0' after 75 ns;
J <= '0', '1' after 10 ns, '0' after 20 ns, '1' after 35 ns, '0' after 70 ns,
    '1' after 80 ns, '0' after 90 ns, '1' after 105 ns, '0' after 140 ns;
K <= '0', '1' after 25 ns, '0' after 70 ns, '1' after 95 ns, '0' after 140 ns;
```

Ottenendo:

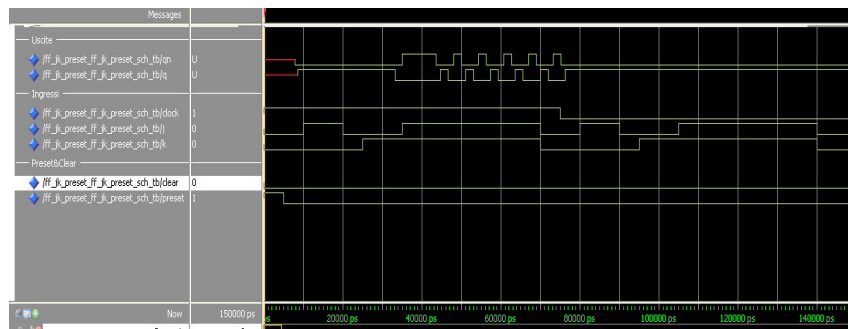


Figura 2.21:

Dalla simulazione si può notare come il comportamento sia del tutto analogo a quello del latch SR, tranne che nel caso in $J = K = 1$, in questo caso infatti le uscite del JK commutano continuamente per un numero di volte che dipende dalla durata per cui i segnali J e K permangono alti. È estremamente critico riuscire ad ottenere una sola commutazione, che, come detto, richiede di far permanere i segnali $J=K=1$ per un tempo sufficientemente grande per generare la commutazione, ma abbastanza piccolo da non innescare instabilità con la continua commutazione delle uscite. Essendo piccolo tale intervallo di tempo, il rispettare tale vincolo risulta molto critico e, pertanto, non si ha un utilizzo del JK latch così realizzato e si preferisce usare soluzioni di tipo edge triggered.

2 Bistabili: latch e flip-flop

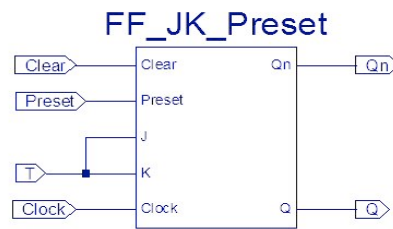


Figura 2.22: Latch JK usato da Trigger

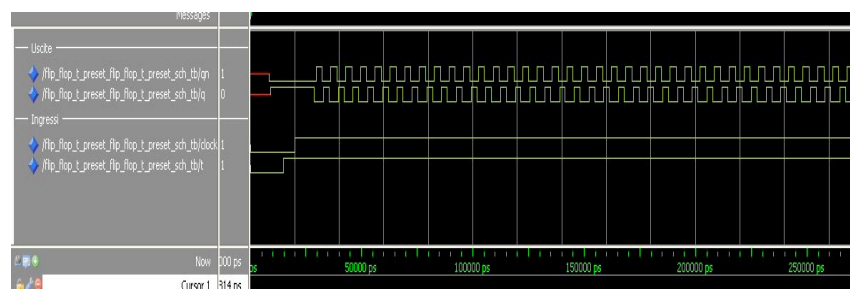


Figura 2.23:

2.4.5 Latch T derivato dal latch JK

Dal JK si può derivare il latch T semplicemente collegando i due ingressi J e K in un unico ingresso T (fig.2.22).

In questo caso la tabella di verità diventa:

T	Q	Q'
0	Q _p	Q _p '
1	Q _p '	Q _p

Dove per Q_p si indica il valore precedente dell'uscita.

Il circuito così ottenuto può essere simulato post-route con le seguenti waveform (fig.2.4.5)

```
T <='0', '1' after 15 ns; Clock <='0','1' after 20 ns; Preset<='1', '0' a
```

Che mette ancora più in evidenza il problema prima citato, versioni funzionanti di questo flip flop si ottengono a partire da JK Edge Triggered o Master-Slave

2.5 Flip Flop Edge triggered e master-slave

Il funzionamento dei latch visti precedentemente dipende in maniera critica dalla durata dei segnali a livello che ne determinano la commutazione (ad es il clock di abilitazione). È possibile rendere i tali dispositivi insensibili alla durata o al valore di detti segnali, facendo sì che le transizioni di stato avvengano solo quando il segnale commuta da un livello all'altro (variazione sui fronti). Tale effetto può realizzarsi o con ricorso al modello a variazione sul fronte (edge triggered) o mediante l'architettura detta master-slave.

2.5.1 Analisi e sintesi di un flip flop D edge triggered (a variazione sul fronte)

Il flip flop *D edge triggered* è un dispositivo bistabile con due ingressi, di cui uno D (dato) e uno clk (clock), e due uscite Q e Q'. Esso effettua la transizione di stato all'occorrenza di una variazione sul fronte attivo, di salita o di discesa, del segnale di clock e dello stato degli ingressi in quell'istante. Tale dispositivo può essere analizzato e progettato o con un procedimento di sintesi dei circuiti elettronici, e cioè come circuito elettronico digitale retroazionato, o con i classici metodi di sintesi usati per le macchine sequenziali asincrone o come sistema di reti sequenziali composto da tre sottosistemi latch di tipo RS.

Di seguito sono riportate l'analisi e la sintesi del flip-flop D edge triggered che, come si vedrà sarà composto da tre latch RS fondamentali (realizzati con porte NAND, se il flip flop è attivo sul fronte di salita del clock, con porte NOR, se su quello di discesa).

2.5.1.1 Analisi del funzionamento

Per l'analisi del funzionamento del flip-flop D al variare dei due segnali di ingresso clk e D ci si riferirà allo schema a porte NOR, sensibile ai fronti di discesa del clock, è riportato in fig.2.24. Esso impiega solo 6 porte logiche per un totale di 26 transistor, a differenza dalle 11 porte (con 38 transistor) richieste per la sintesi di un analogo flip-flop operante secondo il modello master-slave. Le sei porte NOR realizzano tre latch RS fondamentali, indicati come *latch1*, *latch2* e *latch3* e fra loro interconnessi.

Si ricorda che una porta NOR è 0-attiva, essa presenta un'uscita alta solo quando i suoi ingressi sono entrambi bassi, negli altri casi la porta fornisce l'uscita bassa (la NOR riconosce, quindi, lo stato in cui tutti i suoi ingressi sono bassi), per cui se si pone un valore logico '0' a uno dei suoi due ingressi (un valore che consideriamo neutro), il valore logico dell'uscita dipenderà soltanto da quello dell'altro ingresso.

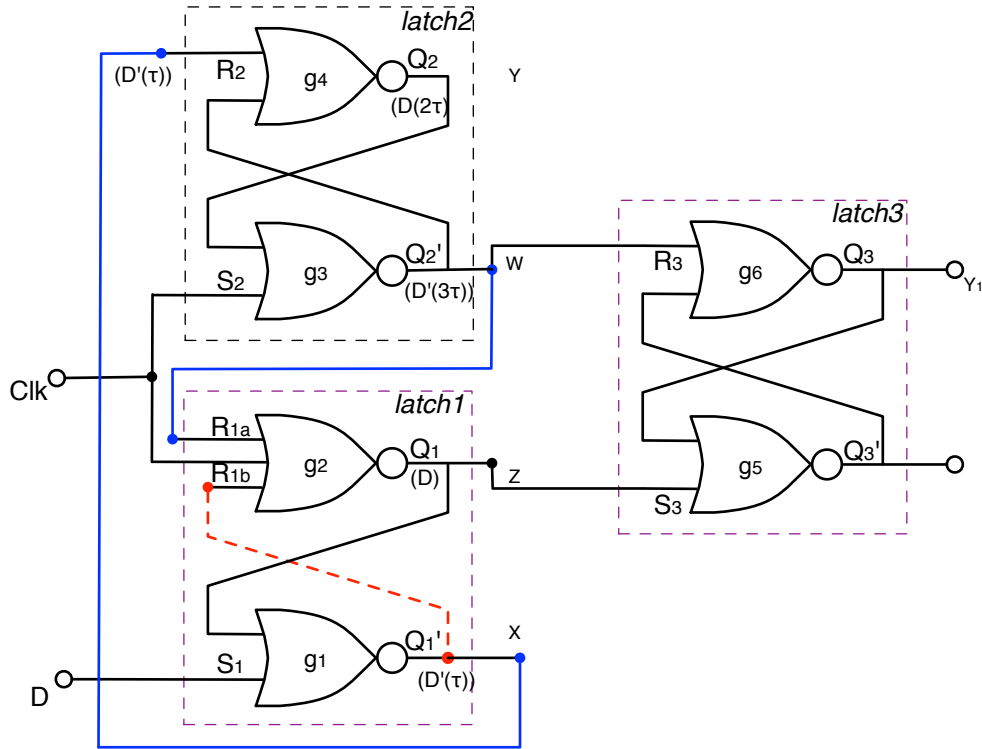


Figura 2.24: flip-flop D edge triggered sul fronte di discesa

I latch *latch1* e *latch2* di cui alla fig.2.24, sono fra loro interconnessi in anello chiuso. La reazione avviene per tramite delle linee evidenziate in blu che collegano le due uscite negate Q_1' e Q_2' rispettivamente con gli ingressi R_2 e R_{1a} ⁴.

Se il segnale di clock è nello stato alto ($clk=1$), il nodo Z della porta g_2 e quello W della g_3 sono forzati al livello basso, indipendentemente dai valori presenti negli altri ingressi delle due porte. Tale stato ($Z = W=0$) rappresenta il valore neutro per gli ingressi R_3 e S_3 del *latch3* che, pertanto, continua a mantenere immutato lo stato delle sue uscite Q_3 e Q_3' . Le due porte g_1 e g_4 , essendo il valore 0 presente in Z e W un valore neutro per esse, operano da inverter e sono, in tale condizione, libere di commutare e di seguire lo stato dell'ingresso D. In particolare, l'uscita di g_1 assumerà il valore D' con il ritardo τ proprio della porta, mentre l'uscita di g_4 il valore D, ma con un ritardo 2τ dovuto ai tempi di propagazione del segnale D attraverso g_1 e g_4 .

Quando il segnale di clock commuta a $clk=0$ (fronte attivo $1 \rightarrow 0$), le due porte g_2 e g_3 , operano esse da semplici inverter. In particolare, g_3 inverte Y (l'uscita W assume il valore di D con un ritardo di 2τ in quanto il path del segnale attraversa le due porte g_1 e g_4), mentre g_2 inverte W (l'uscita Z assume il valore D' con un ritardo di 3τ in quanto il path del segnale attraversa le tre porte g_1 , g_4 e g_3). Quindi, da quando $clk=0$, il nodo

⁴Il motivo per cui il *latch1* ha due ingressi di reset e non uno, introducendo una asimmetria nel circuito, sarà spiegato nel seguito

W, collegato al reset R_3 , assumerà il valore D' dopo 3τ ; il nodo Z, collegato al set S_3 il valore D dopo 4τ . $W=D'$ e $Z=D$, connessi rispettivamente agli ingressi di reset e set del *latch3*, determineranno lo stato delle sue uscite: $D=Z=1$ forzerà $Q_3=1$ e $D'=W=0$ forzerà $Q_3=0$.

Il flip-flop descritto, senza la connessione marcata in rosso, supponendo, come detto, uguale a τ il ritardo delle sue porte, impiega 4τ per la commutazione di *latch1* e *latch2* e 2τ per la commutazione del *latch3*, per un totale di 6τ che rappresenta il tempo di setup t_{setup} ⁵

Inserendo la connessione in rosso, il ritardo massimo diviene di 3τ per la commutazione di *latch1* e *latch2* e di 5τ in totale.

Sostituendo le sei porte NOR con altrettante porte NAND si perviene a un flip-flop D sensibile sul fronte di salita del clock. La NAND, dualmente alla NOR, è 1-attiva in quanto presenta l'uscita bassa solo se i suoi ingressi sono entrambi alti mentre negli altri casi fornisce l'uscita alta.

2.5.1.2 Analisi flip-flop D visto come sistema di reti sequenziali

L'analisi del flip-flop D può essere anche fatta considerandolo come composto da un sistema di reti sequenziali con tre latch RS (ovviamente tale soluzione è del tutto equivalente circuitualmente a quella a 6 NOR precedentemente descritta). Il circuito generato può essere visto come composto da tre latch di tipo RS fondamentale, di cui i primi due (*latch1* e *latch2*) formano la rete di controllo (PC) adibita a sincronizzare i valori dell'ingresso D con il fronte di salita del segnale clk, e il terzo, il *latch3*, rappresenta la parte operativa (PO) adibita alla sola memorizzazione del dato D acquisito. La rete PC ha un solo ingresso e le due uscite R_3 e S_3 atte a pilotare il *latch3* di tipo RS.

La fig.2.25 mostra nella parte superiore tale soluzione indicando con un riquadro tratteggiato ciascuno dei tre latch RS, mentre la figura in basso schematizza, senza il dettaglio costruttivo di ciascuno dei latch, la stessa architettura, non si riportando per semplicità il terzo latch RS pilotato dalle due linee di comando S_3 e R_3 , collegate rispettivamente a Q_1' e Q_2 e inessenziale per la spiegazione del funzionamento del circuito.

La tabella 2.5 riporta gli stati degli ingressi e delle uscite dei due latch al variare degli ingressi D e Clk. Come già visto nell'analisi del circuito a NOR sopra riportata, quando $\text{Clk}=1$, il *latch2*, se $D=0$, ha gli ingressi $R_2=S_2=1$ e, di conseguenza, le sue uscite assumeranno entrambe il valore $Q_2=Q_2'=0$, analoga cosa avviene per il *latch1* quando $D=1$. In tali condizioni i primi due latch seguono le variazioni di D (sono trasparenti a D), in particolare nel primo caso il flip-flop si predispone per memorizzare l'ingresso $D=0$, nell'altro il $D=1$. Tale valore di D sarà trasferito in uscita al verificarsi del fronte attivo 1-0 di Clk, infatti Quando si perviene alla condizione $\text{Clk}=0$, è attivo il circuito colorato in blu nella figura, quindi il *latch2* è pilotato solo dall'ingresso R_2 che assume il valore

⁵Si ricorda che l'uso di un tale tipo di flip-flop come elemento di memoria in circuiti digitali reazionati richiede che sia soddisfatta la condizione che il tempo di ritardo t_d nell'anello di reazione soddisfi la condizione $t_d > t_{\text{setup}}$

neutro 0 per il *latch2* se $D=1$ e il valore 1 se $D=0$. Riepilogando, il *latch1* presenterà lo stato $Q_1=S_3=1$ se $D=1$ e il *latch2* lo stato $Q_2'=R_3=1$ se $D=0$, pilotando così in modo coerente il *latch3*.

Clk	D	R ₁	S ₁	Q ₁ =S ₃	Q ₁ '	R ₂	S ₂	Q ₂	Q ₂ '=R ₃
1	0	1	0	0	1	1	1	0	0
1	1	1	1	0	0	0	1	1	0
0	0	0	0	0	1	1	0	0	1
0	1	0	1	1	0	0	0	1	0

Tabella 2.5: Tabella analisi del flip-flop D visto come sistema di tre latch RS

2.5.1.3 Sintesi del flip-flop D sensibile al fronte di salita del clock

Si riporta di seguito il procedimento di sintesi del flip-flop D sensibile al fronte di salita (e quindi realizzato con porte NAND), effettuata seguendo il tradizionale procedimento di sintesi delle macchine sequenziali asincrone. Tale procedimento si avvia con la costruzione della tabella primitiva (Tab.2.6) che riporta, a partire da uno stato iniziale stabile, le evoluzioni di stato del sistema al variare dello stato dei suoi ingressi (D e clk) e il valore assunto dall'uscita (Q)⁶.

Nella tabella primitiva, si rappresentano, per ciascuna configurazione degli ingressi, l'evoluzione della rete al variare delle possibili configurazioni delle uscite. Per una rete con n ingressi e m uscite si potranno, quindi, avere al massimo 2^{n+m} configurazioni possibili, per cui la tabella di flusso primitiva avrà altrettante righe e, per ciascuna riga un solo stato stabile. Tale tabella iniziale di descrizione della rete asincrona, per come è costruita, risulterà ridondante nel numero di stati. Da essa, applicando l'algoritmo di minimizzazione basato sulla ricerca degli stati compatibili, si perviene a una tabella ridotta, eventualmente con un minor numero di stati. Si procede, quindi, con il passo di codifica degli stati, degli ingressi e delle uscite e, infine, con quello della progettazione delle reti combinatorie per il calcolo dello stato prossimo e delle uscite (dipendente dal modello Mealy o Moore che si intende applicare), pervenendo in tal modo alla sintesi finale della rete corrispondente alla specifica formale espressa mediante la tabella di flusso iniziale.

Nel caso specifico, essendo la rete che si intende realizzare semplice, si possono semplificare i passi di sintesi precedentemente elencati osservando che essa presenta solo due ingressi (D e clk) e due uscite Q e Q'). La tabella primitiva avrebbe pertanto $2^{2+2}=16$ righe, di cui ciascuna con uno stato stabile. Essendo però le uscite Q e Q' l'una inversa dell'altra, si può semplificare la costruzione della tabella considerando soltanto una sola delle due uscite. Essa, seppur ancora ridondante, si riduce pertanto alla metà delle righe

⁶Si ricorda che la tabella primitiva può essere dedotta a partire da un diagramma temporale che associa differenti stati q_i a tutte le combinazioni differenti dei segnali di ingresso e di quelli delle uscite.

2 Bistabili: latch e flip-flop

D, clk					Q
Stato	00	01	11	10	
S1	S1	S2	-	S4	0
S2	S1	S2	S3	-	0
S3	-	S2	S3	S4	0
S4	S1	-	S7	S4	0
S5	S5	S2	S3	S8	1
S6	S5	S6	S7	-	1
S7	-	S6	S7	S8	1
S8	S5	-	S7	S8	1

Stati equivalenti e codifica degli stati					y1y2
Q ₀ (S1, S2, S3)					00
Q ₁ (S4)					01
Q ₂ (S5)					10
Q ₃ (S6, S7, S8)					11

D, clk↑					Q
	00	01	11	10	
Q₀	Q₀	Q₀	Q₀	Q ₁	0
Q₁	Q ₀	-	Q ₃	Q₁	0
Q₃	Q ₂	Q₃	Q₃	Q₃	1
Q₂	Q₂	Q ₀	-	Q ₃	1

D, clk↑					Q
y1y2	00	01	11	10	
00	00	00	00	01	0
01	00	-	11	01	0
11	10	11	11	11	1
10	10	00	-	11	1

Tabella 2.6: Tab. di flusso primitiva del ff D-Edge Trigg. sul fronte di salita, tab. ridotta con fusione righe e tab. in codice

($2^{2+1}=8$). Applicando, quindi, la minimizzazione degli stati, la tabella si trasforma in una tabella equivalente di sole quattro righe, come mostrato in fig.2.6.

Procedendo, quindi, con la codifica degli stati equivalenti si pervenire alla tabella in codice mostrata sempre nella stessa figura.

Dalla tabella in codice si derivano le due tabelle di verità per la sintesi delle variabili di stato secondarie y1 e y2, così come riportate in fig.2.7:

Minimizzando (si selezionano per Y1 gli implicant primari {5,7,9,11} {8,9,10,11} {8,10,12,15} e per Y2 {5,7,9,11} {2,6,10,14}), si derivano per Y₁ e Y₂ le seguenti equazioni:

2 Bistabili: latch e flip-flop

Tabella per Y1	D, clk				Q	Tabella per Y2	D, clk				Q
y1y2	00	01	11	10		y1y2	00	01	11	10	
00	0	0	0	0	0	00	0	0	0	1	0
01	0	-	1	0	0	01	0	-	1	1	0
11	1	1	1	1	1	11	0	1	1	1	1
10	1	0	-	1	1	10	0	0	-	1	1

Tabella 2.7: Tabelle di verità relative alle variabili Y1 e Y2

$$Y_1 = y_1 \cdot y_2 + y_2 \cdot clk + y_1 \cdot \overline{clk} = y_1 \cdot (y_2 + \overline{clk}) + y_2 \cdot clk = y_2 \cdot clk + y_1 \overline{clk}$$

$$Y_2 = y_2 \cdot clk + y_2 \cdot D + D \cdot \overline{clk} = D(y_2 + \overline{clk}) + y_2 \cdot clk$$

che espresse in forma di soli NAND divengono:

$$Y_1 = (y_1 \uparrow (\overline{y_2} \uparrow clk)) \uparrow (y_2 \uparrow clk)$$

$$Y_2 = (D \uparrow (\overline{y_2} \uparrow clk)) \uparrow (y_2 \uparrow clk)$$

Da esse, dopo aver fatto le assegnazioni di seguito riportate, indicando con g₁-g₆ le uscite delle sei porte Nand utilizzate, si perviene al circuito di fig.2.26 (privo della connessione indicata con la linea in rosso, che come vedremo è inserita per eliminare un'alea nella rete):

$$y1 \equiv Q; y2 \equiv Q2$$

$$g_4 \equiv Q_2 : (g_1 \uparrow g_3)$$

$$g_3 \equiv Q'_2 : (y_2 \uparrow clk)$$

$$g_2 \equiv Q_1 : (\overline{y_2} \uparrow clk)$$

$$g_1 \equiv Q'_1 : (D \uparrow g_2)$$

$$g_6 \equiv Q_3 \equiv Q : (g_5 \uparrow g_3) = y_1$$

$$g_5 \equiv Q'_3 \equiv Q' : (y_1 \uparrow g_2) = y_2$$

Il circuito generato può essere visto come composto da tre latch di tipo RS fondamentale, di cui i primi due (latch1 e latch2) formano la rete di controllo (PC) adibita a sincronizzare i valori dell'ingresso D con il fronte di salita del segnale clk, e il terzo, il latch3, rappresenta la parte operativa (PO) adibita alla sola memorizzazione del dato D acquisito. La rete PC ha un solo ingresso e le due uscite R₃ e S₃ atte a pilotare il latch3 di tipo rS.

Dalla sintesi precedentemente mostrata deriva che il *latch1* è privo della retroazione indicata con la linea in rosso, è stata inserita per evitare una alea.

Ragionando in modo analogo a quanto visto nell'analisi del flip-flop a porte NOR, in quello a porte NAND, l'ingresso clk=0 forza i nodi w e z, collegati agli ingressi del latch3, ad assumere il valore w=z=1, mentre i nodi w e z seguono le variazioni di D essendo in tale condizione y=D dopo 2τ e x=D' dopo τ (essi memorizzano il valore di D all'arrivo

del fronte 0-1 di clk). Quando si verifica la transizione $clk=1$, si ha ancora $y=D$ $x=D'$, mentre w e z assumeranno i valori $w=D'$ dopo (dopo 3τ) $z=D$ (dopo 4τ) che vanno a pilotare coerentemente gli ingressi $R3$ e $S3$ del latch3 per memorizzare il nuovo stato D .

2.5.1.4 Analisi degli Hazard e verifica mediante simulazione

Con riferimento all'implementazione con porte NAND, la rete presenta un'alea quando $D = 1$ e il clk effettua la transizione attiva da 0->1. In tal caso, le porte $g3$ e $g2$ operano da inverter dei segnali y e w collegati al loro ingresso. Tale transizione dovrebbe lasciare inalterato a 1 il valore già presente in z , valore forzato da $R1a$ che commuta al valore basso. Ma a causa del ritardo τ della porta $g3$, l'ingresso $R1a$ commuta solo dopo tale tempo, generando la tipica sequenza 1->0->1 dell'alea statica, che si manifesta con un glitch all'uscita di $g2$ (evidenziato nel diagramma di fig.2.27) e che potrebbe essere causa di un malfunzionamento. Tale alea è eliminata aggiungendo un ingresso ($R1b$) alla porta $g2$ e collegandolo all'uscita x di $g1$ ove già è presente il valore $D'=0$ (collegamento marcato in rosso), ciò fissa il valore $Z = 1$.

L'alea può essere evidenziata osservando i risultati della simulazione behavioral del D-latch riportata in fig.2.27 in assenza del collegamento marcato in rosso. In fig.2.28 è, invece, riportata la simulazione effettuata in presenza di tale connessione che elimina l'alea.

2.5.2 Flip Flop T edge triggered

In fig.2.29 è riportato lo schema di un flip-flop T derivato da un flip-flop D. Tale rete sequenziale può essere vista, al solito, come composta da un elemento di memorizzazione (il flip-flop D) e da un elemento di sincronizzazione e commutazione, in tal caso una semplice porta exclusive-OR. L'uscita Q reazionata in ingresso alla porta exclusive-OR forza la commutazione per ogni variazione del fronte attivo del flip-flop D.

2.5.3 Flip Flop JK edge triggered

Il flip-flop JK edge triggered, così come mostrato in fig.2.30, può essere derivato dal flip flop D edge triggered antemponendovi una rete di controllo con quattro ingressi (Q' , J , K , Q) e un'uscita D . Osservando che un flip flop D si limita a trasferire all'uscita Q il valore di D , la funzione booleana per D , in funzione di J , K e Qn è data dalla stessa dell'equazione caratteristica di JK. La rete di controllo esterna al flip flop D realizza, pertanto, la mappa i segnali JK con D in modo coerente alla logica del flip-flop JK, mentre il flip flop D assolve soltanto alle funzioni di sincronizzazione delle transizioni di stato con il fronte attivo del suo segnale di abilitazione ck .

2.5.4 Flip-Flop Master Slave

I master-slave, così come gli edge triggered sono anche essi flip-flop a variazione sul fronte. A differenza del modello edge-triggered precedentemente illustrato, il master-slave implementa la sensibilità al fronte (di salita o di discesa) utilizzando un circuito con due latch asincroni, così come mostrato per il flip-flop RS in fig.2.31, il primo opera da master e il secondo da slave. Il segnale abilitante C è inviato in forma diretta al master e invertito allo slave. Supponendo un'abilitazione 1-attiva, quando $C=1$ il master è abilitato e segue le variazioni di R ed S, mentre lo slave risulta disabilitato per cui l'uscita Q mantiene il valore precedentemente memorizzata. Quando $C=0$, la situazione si inverte: il master è disabilitato e non segue più i cambiamenti di R ed S, mentre lo slave è abilitato e memorizza lo stato catturato dal master sul fronte 1- \rightarrow 0 di C. Si fa osservare che, stante la presenza dell'inverter che presenta un suo proprio ritardo T i segnali C all'ingresso del master e dello slave non sono perfettamente sfasati di 180 gradi (uno inverso dell'altro) ma quello dello slave sfasa con il ritardo T. Ciò crea una finestra "detta di trasparenza" che seppur di breve durata T lascia abilitati sia il master sia lo slave, cosa che potrebbe creare un'anomalia sul funzionamento teorico del master-slave. Per garantire un corretto funzionamento occorre evitare, quindi, che i segnali R ed S siano stabili prima della transizione di C.

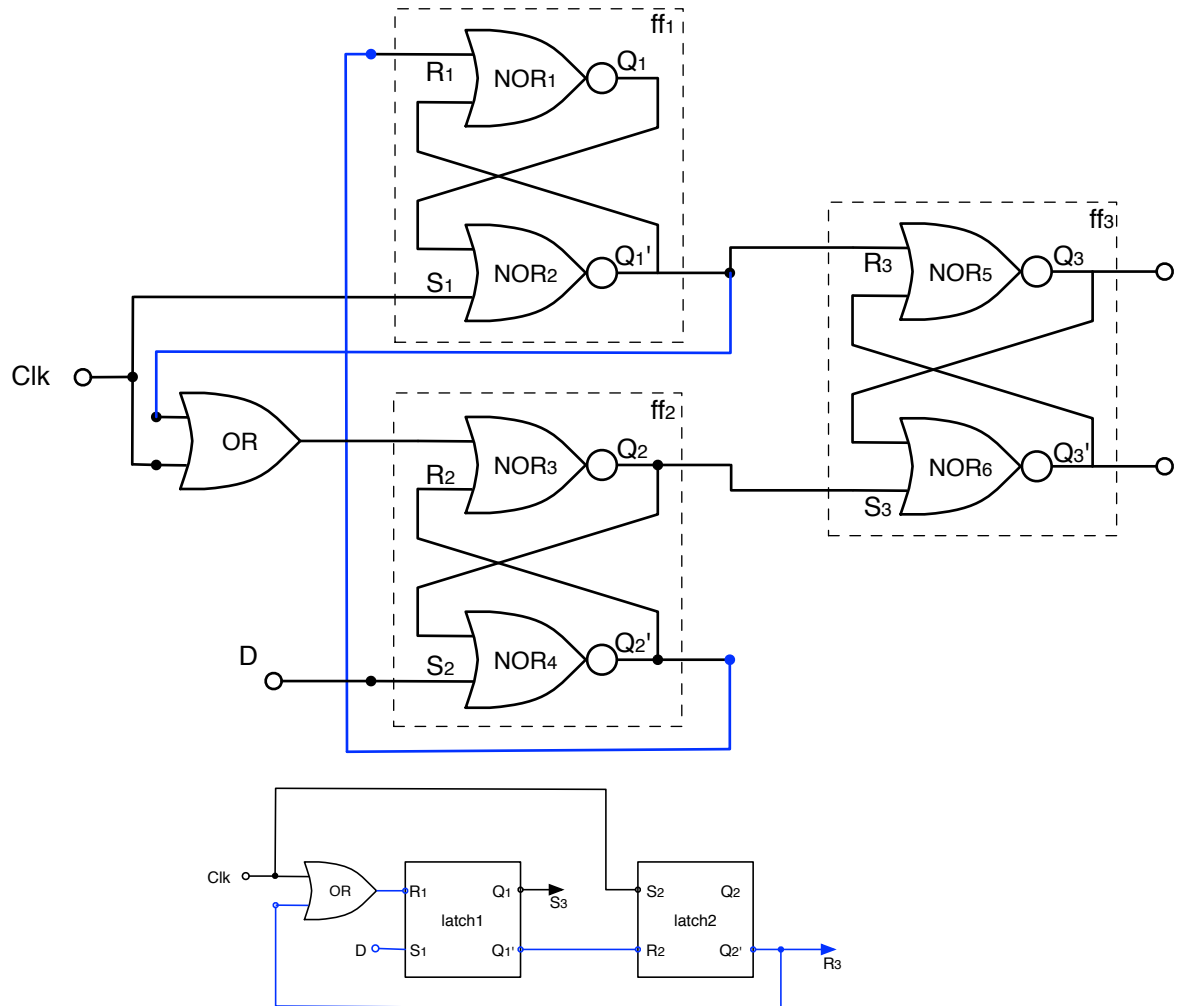


Figura 2.25: Flip-flop D realizzato con latch RS

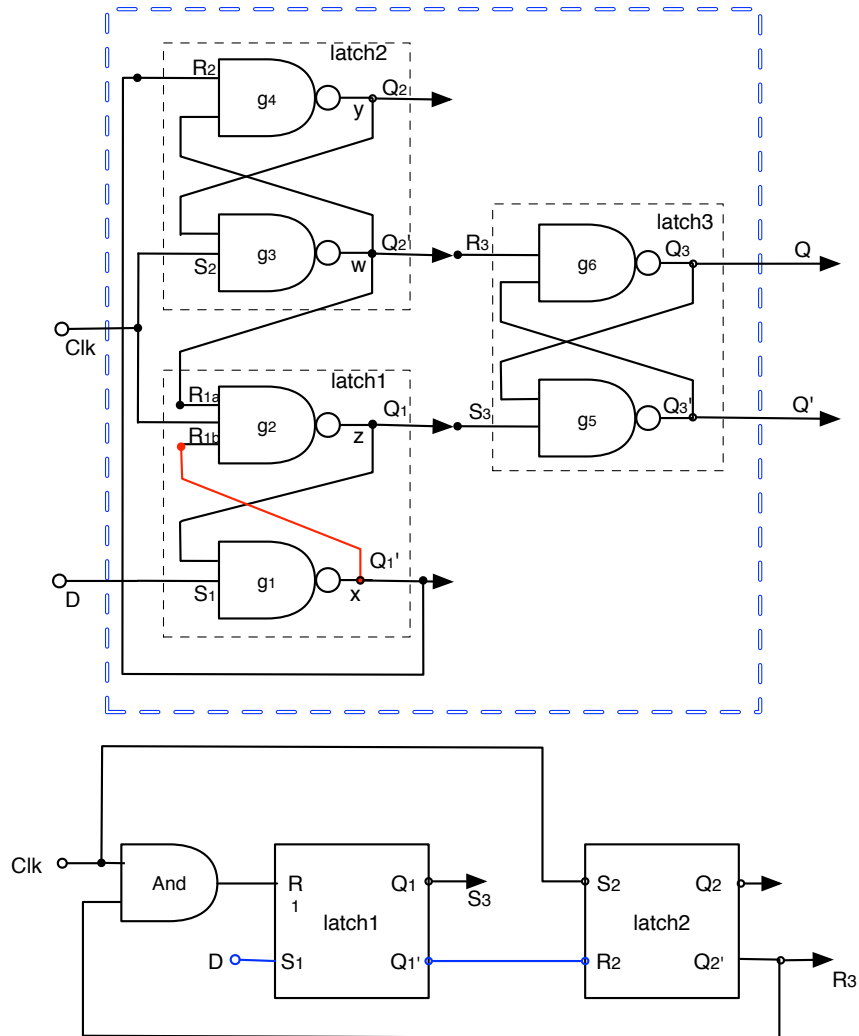


Figura 2.26: flip-flop D edge triggered

2 Bistabili: latch e flip-flop

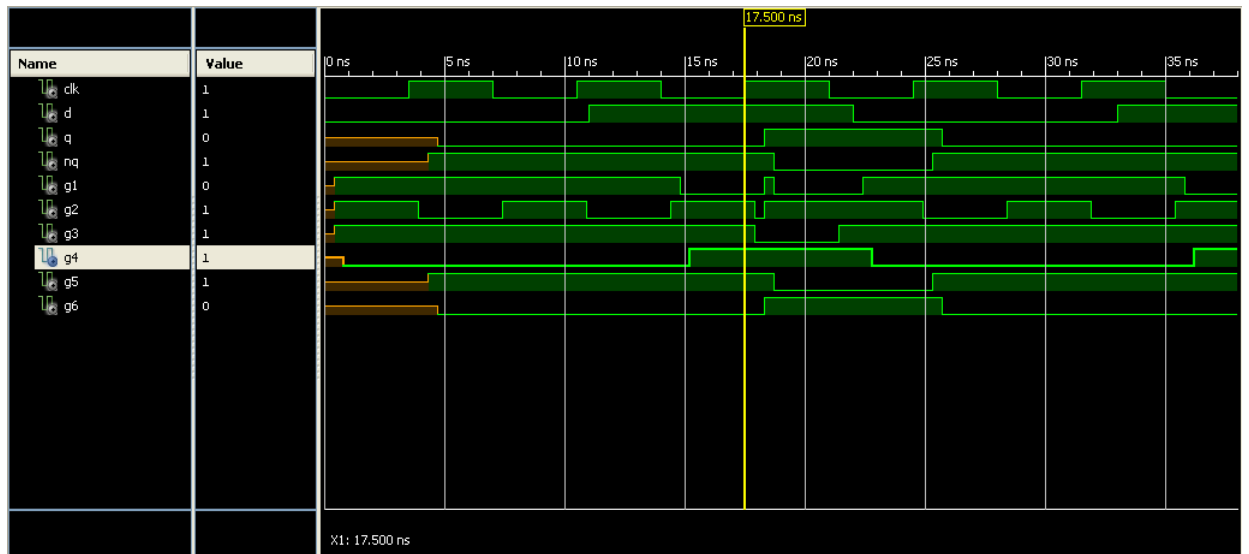


Figura 2.27: Latch RS edge triggered a NAND con alea

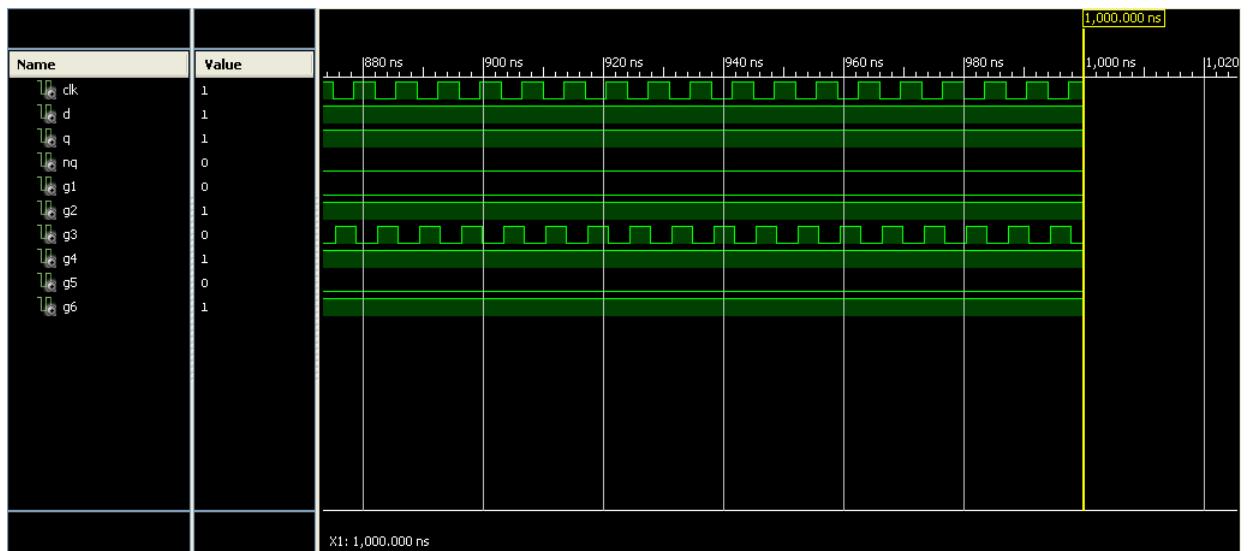


Figura 2.28: Latch RS edge triggered a NAND senza alea

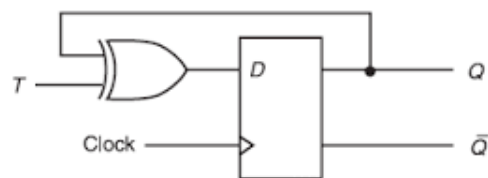


Figura 2.29: Flip-Flop T Edge triggered derivato dal flip-flop T

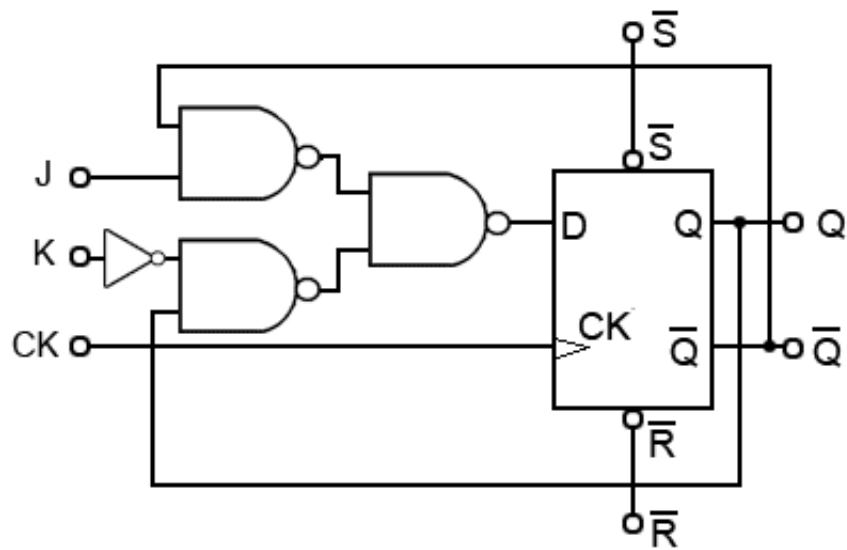


Figura 2.30: Jk-edge triggered

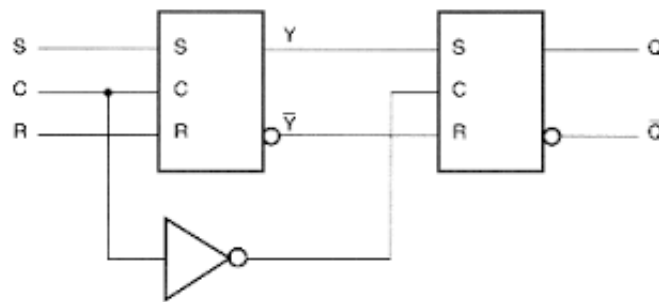


Figura 2.31: Flip-Flop RS Master-Slave

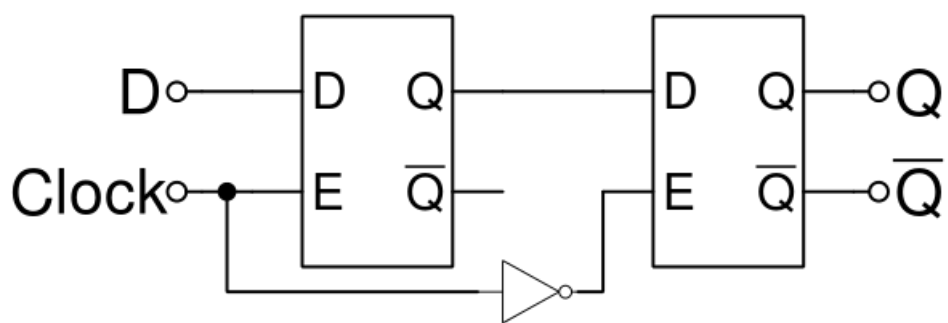


Figura 2.32: Flip-flop D master-slave

3 Esempi di descrizione in VHDL dei flip flop e dei latch illustrati

Come esercizi di laboratorio si faccia riferimento agli esempi posti nella cartella latch-flip-flop ove sono riportati i sorgenti in VHDL di taluni degli esempi discussi precedentemente che possono essere eseguiti per la simulazione nell'ambiente GHDL e per la sintesi su FPGA nell'ambiente ISE di Xilinx¹. Di seguito si allegano soltanto i sorgenti del flip-flop D edge triggered.

3.1 D Latch Edge Triggered

3.1.1 Listato del Latch in VHDL

```
-----
--
-- Title       : Flip-flop D_edge-triggered con NOR
-- Design      : D_EDGE
-- Author       : Antonino Mazzeo
-- Company      : DIS
--
-----
--
-- File         : d_edge_nor.vhd
-- Generated      : Friday oct 21, 22:10:44 2011
-- From          : interface description file
-- By            : Itf2Vhdl ver. 1.20
--
-----
--
-- Description : Il flip flop d implementato è di tipo edge triggered sul
               fronte
-- 1->0 di discesa (trailing edge) del clock C
--
-----

library IEEE;
use IEEE.STD_LOGIC_1164.all;

entity d_edge is
  generic (td: time:=0.0 ns);
end d_edge;
```

¹Si lascia allo studente l'esercizio di effettuare il porting degli stessi esempi nell'ambiente SystemC scaricabile dal sito Accelerata.

3 Esempi di descrizione in VHDL dei flip flop e dei latch illustrati

```
architecture d_edge of d_edge is

signal D, C : STD_LOGIC:= '0';
signal Q      : STD_LOGIC;
signal NQ     : STD_LOGIC;
signal G1     : STD_LOGIC;
signal G2     : STD_LOGIC;
signal G3     : STD_LOGIC;
signal G4     : STD_LOGIC;
signal G5     : STD_LOGIC;
signal G6     : STD_LOGIC;
signal G2b    : STD_LOGIC;

begin
    G1<=D nor G2 after td;
    G2<=not (G1 or G3 or C) after td; --nor(G1,G3,C)
    -- G2<=(G1 nor C) after td;
    G3<=C nor G4 after td;
    G4<=G3 nor G1 after td;

    G5<=G2 nor G6 after td;
    G6<=G5 nor G3 after td;
    Q<=G6; NQ<=G5;
    D <= '1' after 5 ns, '0' after 14 ns, '0' after 18 ns, '1' after 21 ns, '0'
        after 23 ns, '1' after 25 ns, '0' after 40 ns, '1' after 43 ns, '0' after
        48 ns, '0' after 52 ns, '1' after 57 ns, '0' after 61 ns, '1' after 65 ns
        , '0' after 69 ns;
    -- D<=not D after 6 ns;
    C <= '1' after 5 ns, '0' after 10 ns, '1' after 15 ns, '0' after 20 ns, '1'
        after 25 ns, '0' after 30 ns, '1' after 35 ns, '0' after 40 ns;
    --C<=not C after 1 ns;

end d_edge;
```

Appendice A

Segnali 0-attivi e 1-attivi

Solitamente, uno dei due valori di una variabile logica è associato all'attivazione di una qualche funzione di un sistema digitale. Un tale valore è detto valore *attivo* mentre l'altro detto valore *neutro*, è un valore di riposo che non crea alcun evento significativo per l'evoluzione del sistema digitale (non attiva alcuna funzione). La scelta di quali dei due valori deve essere attivo dipende dal tipo di sistema considerato e dalla scelta di progetto. In particolare, una variabile è di tipo 1-attiva se lo stato 1 è scelto come stato attivo, 0-attiva nel caso contrario (le variabili 1-attive sono indicate con X e quelle 0-attive con X').

Rappresentazione di una variabile booleana in logica positiva o negativa

Nella realizzazione di un circuito digitale, i due valori di una variabile logica 0 e 1 (indicati anche come L o H) sono associati ai valori di una grandezza di un segnale che, nel caso di tensione, può assumere due valori V1 e V0. Se $V1 > V0$ la logica utilizzata per rappresentare le variabili è detta *positiva*, se $V1 < V0$ è detta *negativa* (ad esempio, i valori di tensione +5 V associato al livello H e 0 V a quello L, individuano una logica positiva. Occorre fare l'assegnazione inversa se si vuole operare in logica negativa). Una funzione combinatoria f utilizzata in logica positiva, se utilizzata in logica negativa, opera in modo duale² e viceversa. Ad esempio, in logica negativa, l'operatore AND ($\overline{X} \cdot \overline{Y}$) opera, se usato in logica positiva, come un operatore OR ($\overline{X} \cdot \overline{Y} \iff X + Y$), una NAND fra due variabili in logica positiva $\overline{X \cdot Y}$ è equivalente alla sua duale NOR ($\overline{X \cdot Y} \iff \overline{X} + \overline{Y}$) fra variabili che operano in logica negativa. Ove occorra trasformare una variabile operante in logica positiva in una operante in logica negativa, e viceversa, occorre ricorrere all'uso di un inverter.

X	Y	AND	OR	NAND	NOR	
L	L	L	L	H	H	
L	H	L	H	H	L	
H	L	L	H	H	L	
H	H	H	H	L	L	
		OR	AND	NOR	NAND	

Figura 3.1: Operatori logici e loro duali

²Da una qualsiasi identità booleana se ne può trarre un'altra per dualità, sostituendo a ogni operatore e agli elementi 0 e 1 il rispettivo duale: il duale di + è *, il duale di 0 è 1. Il duale di una semplice variabile x è \overline{x} . Data una funzione $f(x_1, x_2, \dots, x_n)$ in qualsiasi forma, si definisce funzione duale di f , e si indica con δf , una funzione che ha per forma la forma duale di f , ad esempio: $y = \overline{a} + b(c + 0) \quad \delta f = a * (\overline{b} + \overline{c} * 1) = \overline{a} * (b + c * 1)$.

Si fa osservare che la funzione non cambia qualora non si effettui la negazione delle variabili. Devono essere negate solo le costanti 0 e 1.

Appendice B

Behavioral Simulation A behavioral simulation uses the Verilog source code you wrote in order to model the behavior of the module under test. Neither gate delays nor interconnect delays are modeled. Furthermore, functionality of the behavioral model may not match that of the synthesized logic, if you make use of Verilog language features that the synthesis engine cannot handle. (In particular, be careful about incomplete sensitivity lists.)

While behavioral simulation gives the least accurate prediction of how the final hardware implementation will perform, it is the most useful form of simulation during the initial debugging of a design. There is little point in running more realistic simulations until the behavioral model works correctly.

Non-Behavioral Simulations

Non-behavioral simulations are performed on some form of synthesized netlist. To simulate synthesized netlists of the fsm module, you need to fully synthesize the fsm module. You will need to set fsm as the top module first by right clicking on fsm1.

Post-Translation Simulation

A post-translation simulation uses the synthesized gate-level netlist to model the module under test. The functionality of the gates is modeled using a generic Xilinx library, but propagation delay is not modeled. The simulation should match the behavior of the actual hardware, but will assume the hardware is infinitely fast. (If you are worried about whether your design has synthesized correctly, start by checking the synthesis report file, as all common synthesis mismatch issues will generate warnings in the report file.)

Post-Mapping Simulation In a post-mapping simulation, the gates have been mapped to a library specific to the FPGA device being targeted. This library includes accurate gate delay information. However, interconnect delay is not modeled, because the design at this stage has not yet been placed and routed.

Post-Place-and-Route Simulation A post-place-and-route simulation models interconnect delay, as well as gate delay. This type of simulation will most accurately match the behavior of the actual hardware. However, for large designs, it can take a significant amount of time to extract the interconnect delay values from the place-and-route information, and a significant amount of time to run the actual simulation.

It really only makes sense to perform post-place-and-route simulations at the top level of a design. If you perform a post-place-and-route simulation on a submodule, the place-and-route process is rerun, using the submodule as the top-level of the design. The interconnect delays for the submodule simulation will therefore not match the interconnect delays for that submodule when it is laid out as part of the complete project.