



di GIORGIO OBER (GIOBE2000)

l'angolo di Mr A.KEER



(parte diciottesima)

PROGETTARE con le PORTE LOGICHE

Il Latch SR

Dopo diciassette puntate dedicate alla logica combinatoria è giunto il momento di affrontare la seconda grande categoria di dispositivi digitali: le macchine sequenziali, il primo passo verso la creazione dei microprocessori.

Quando osserviamo il funzionamento di taluni dispositivi logici importanti, può capitarci di percepire in essi un certo *grado di intelligenza*, come fossero capaci di prevedere la via migliore per portare avanti il loro compito; pensiamo al nostro orologio-sveglia che, non solo organizza impeccabilmente la visualizzazione dei secondi, dei minuti e delle ore (imballandoli in pacchetti di 60 e di 24, inconsueti per l'abitudine decimale umana), ma che sembra pure obbedire agli ordini, attivando una suadente melodia anche a distanza di ore, nella speranza di strapparci dalle braccia di Morfeo..

Inutile dire che qualunque oggetto dei nostri tempi è sostanzialmente di questo tipo, ma attribuire razionalità ad una macchina sembra esagerato (.. o forse prematuro?); di certo siamo in presenza di circuiti (noti come **macchine sequenziali**) in grado di **ricordare lo stato** da loro assunto in un **tempo** precedente, cioè di influenzare le loro uscite non solo in funzione dei valori degli ingressi ma anche di quelli proposti in precedenza.

Le Macchine Sequenziali

Le strutture logiche descritte nelle puntate precedenti hanno tutte la caratteristica di fornire in uscita un valore logico ottenuto dalla combinazione dei quelli forniti sui rispettivi ingressi; ciascuna di esse (per questa ragione dette **macchine combinatorie**) "insegue" ogni loro variazione, rielaborandola attraverso una o più porte logiche, unici dispositivi chiamati a realizzarle.

Se si potesse ritenere nullo il tempo di propagazione delle porte logiche attraversate nel percorso tra ingressi e uscite, la risposta sarebbe immediata, ma anche non volendolo trascurare il ritardo introdotto è oggettivamente comunque molto piccolo, garantendo (come si suol dire) risposta *in tempo reale*.

Le **macchine sequenziali** non possono esser molto diverse: la loro parte più sostanziosa sarà ancora una macchina combinatoria (alla quale, data la sua alta specializzazione, saranno sempre affidati i compiti più onerosi) ma dovendo tener conto anche della *storia passata*, disporrà anche di **elementi di memoria** chiamati (come detto) a ricordare lo stato del circuito.

La *Figura 1* mostra il modello fondamentale di una macchina sequenziale: la sua apparente semplicità non rende giustizia alle numerose considerazioni che esso può suscitare.

In primo luogo notiamo che, vista "da fuori", non sembra diversa da una comune macchina combinatoria, con le sue classiche linee d'ingresso e d'uscita: il dettaglio interno evidenzia però che ciò non è vero, perchè ora le *uscite esterne* sono frutto dell'elaborazione (combinatoria) non solo degli *ingressi esterni* ma anche di uno o più *ingressi interni* (indicati come *Stati presenti*) sui quali è stata scritta (e memorizzata) l'informazione generata a sua volta da altrettante *uscite interne*.

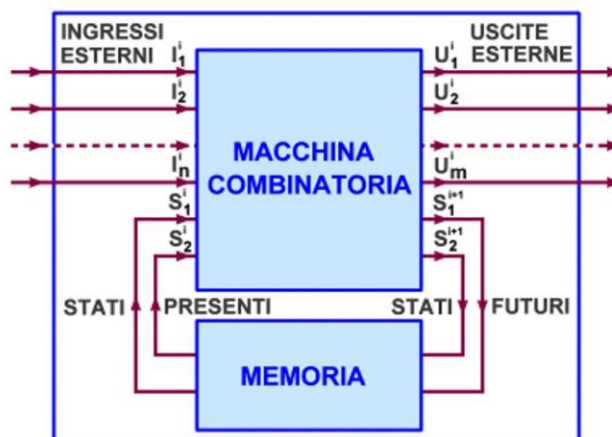


Figura 1 - Modello fondamentale di una Macchina Sequenziale

In secondo luogo possiamo sottolineare la caratteristica fondamentale di questa categoria: per tener conto dello stato precedente una parte delle informazioni presenti in uscita viene riportata in ingresso (linee di feedback).

Ci possiamo ora chiedere come sia possibile che una rete logica possa conservare informazioni relative ad eventi *precedenti* a quelli *attualmente* forniti sui suoi ingressi; già la formulazione di questa domanda lascia trasparire la necessità di disporre del "fattore tempo" come elemento essenziale per il funzionamento di una macchina sequenziale: ci sono eventi *passati* o *futuri* da coinvolgere con eventi *presenti*! Ma l'Algebra di Boole non ha dipendenza temporale, cioè le espressioni booleane non sono esprimibili in funzione del tempo.

Queste considerazioni portano dunque all'introduzione di un nuovo concetto: lo **stato** della rete definito come "**memoria** dell'evoluzione temporale" del dispositivo sequenziale; in questo modo si supera l'apparente incongruenza confondendo un *elemento di memoria* con un *elemento di ritardo*, adatto di fatto a conservare l'informazione ricevuta al tempo t fino ad un tempo successivo $t + \Delta t$; nella *Figura 2* il concetto di tempo è affidato simbolicamente all'istante presente " i " così da stabilire facilmente passato (" $i-1$ ") e futuro (" $i+1$ "): questa convenzione sarà mantenuta anche per ogni prossima considerazione

Queste macchine possono essere gestite (e classificate) in due modi diversi: sincrono e asincrono.

Un circuito **sequenziale** sarà **sincrono** se controllato da un segnale esterno, una forma d'onda quadra a frequenza costante (clock): in questo caso l'elaborazione combinatoria degli ingressi esterni e di quelli interni (forniti dalla memoria) avverrà in istanti ben precisi, $n \cdot \Delta t$, di solito coincidenti con uno dei fronti, di salita o di discesa, del clock, sempre con intervalli Δt

costanti, uguali al periodo del segnale di sincronismo; gli elementi di memoria sono dispositivi concreti, detti *flip-flop*.

Questi circuiti sono molto affidabili (non essendo influenzati dai possibili ritardi tra i vari segnali coinvolti) e molto utilizzati (spesso in forma integrata) per realizzare funzioni sequenziali complesse, come contatori, registri a scorrimento, memorie, logiche di controllo dei dati, microprocessori; il modello fondamentale adattato per una macchina sequenziale sincrona è proposto dalla *Figura 2*.

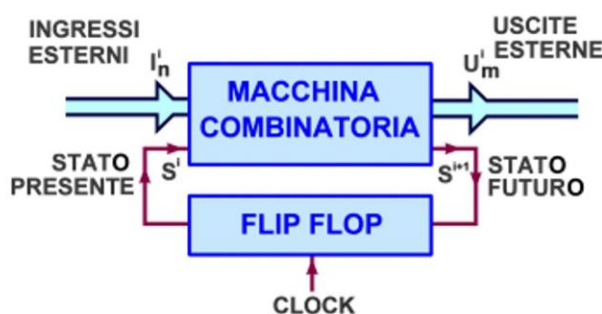


Figura 2 - Modello fondamentale di una Macchina Sequenziale Sincrona

Un circuito **sequenziale asincrono** non dispone invece di alcun clock esterno e non utilizza elementi di memoria come i flip-flop: di solito è realizzato con semplici porte logiche dalle cui uscite sia garantito almeno un rientro (feedback) in ingresso; il modello fondamentale è facilmente adattabile anche a questa modalità (vedi *Figura 3*) se si suppone di "estrarre" i ritardi di propagazione dagli elementi logici coinvolti, concentrandoli in un unico ritardo Δt , al fine di rappresentare l'elemento di memoria necessario al modello; ora il contenuto della memoria (in concreto il livello S^{i+1} dello Stato futuro) sarà proposto quasi istantaneamente insieme agli ingressi esterni per essere immediatamente con essi elaborato dalla rete combinatoria.

Ogni ingresso (esterno o interno) potrà ora arrivare in modo sordinato (asincrono) rispetto agli altri, ma dopo breve tempo le uscite assumeranno un valore stabile, in attesa di una nuova situazione



Figura 3 - Modello fondamentale di una Macchina Sequenziale Asincrona

In definitiva gli **elementi di memoria** sono dispositivi **bistabili**, una delle tre funzioni (con quella monostabile e astabile, entrambe meritevoli, in futuro, di un approfondimento dedicato) associabili ai circuiti **multivibratori**; il percorso che ci accingiamo a percorrere ci porterà alla scoperta di numerose macchine sequenziali elementari, classificabili sotto diverse tipologie, in

funzione delle importanti applicazioni che saranno chiamate a supportare, a cominciare dai dispositivi SR, oggetto di questa puntata.

Il termine **bistabile** è sufficiente per far capire che sulla loro uscita potrà essere presente stabilmente uno dei 2 valori logici possibili e che esso sarà mantenuto (memorizzato) senza variazioni fino a quando il circuito rimarrà alimentato; si tratta dunque di una memoria volatile, il cui valore potrà variare solo in presenza di un evento esterno.

A questo proposito vale la pena anticipare una importante differenziazione che distingue gli elementi di memoria in **latch** e **flip-flop**: i primi (il cui nome deriva dal verbo inglese che significa "mettere sotto chiave") sono *sensibili al livello* logico dei rispettivi ingressi, in base al valore dei quali cambiano l'uscita; i secondi si appoggiano ad un segnale di clock esterno e cambiano stato solo *in presenza di un suo fronte attivo*.

Latch SR

Il circuito logico che andiamo a descrivere è a mio avviso uno dei più affascinanti di tutta la letteratura digitale; la sua definizione nasce dal fatto che per implementare un dispositivo di memoria sono sufficienti 2 inverter, in serie tra loro ed inseriti in un anello di retroazione (Figura 4): se sull'ingresso del primo è presente un 1 logico tale valore (invertito 2 volte) viene "rigenerato" dal circuito e riproposto tale e quale in ingresso (con un ritardo pari a 2 tempi di propagazione) e memorizzato in uscita indefinitamente; lo stesso vale se sul primo inverter è presente uno 0 logico.

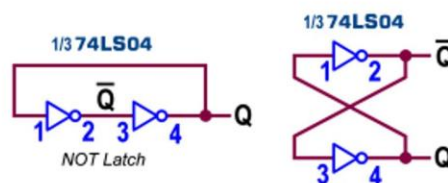


Figura 4 - Elemento di memoria con 2 inverter

Il problema di questo circuito sta nel fatto che non esiste modo di modificare il valore iniziale dell'informazione proposta sul primo inverter, assunta casualmente quando viene fornita l'alimentazione e mantenuta all'infinito dalla sequenza logica appena descritta; per risolverlo è sufficiente sostituire gli inverter con una porta invertente, NOR o NAND, al fine di disporre anche di ingressi esterni (in aggiunta di quelli interni necessari per assicurare la gestione della retroazione).

Ma andiamo con ordine; per prima cosa possiamo notare che l'uscita Q di questi dispositivi di memoria è sempre disponibile sia in forma diretta, sia in forma negata.

Vediamo dunque di progettare un semplice circuito sequenziale in *logica positiva*, in grado di memorizzare una informazione binaria a un bit a partire dal valore predisposto su due ingressi esterni, entrambi *attivi alti*, ai quali è affidato tradizionalmente il nome S (SET) e R (RESET), per sottolineare che il primo dovrà forzare l'uscita diretta Q a 1 e l'altro dovrà riportarla a 0; stiamo per realizzare un **latch SR asincrono**, dato che lo stato della sua uscita cambierà in funzione dei soli segnali di ingresso e non è presente alcun segnale di clock esterno.

Nonostante si tratti di una macchina sequenziale la sua natura sostanzialmente combinatoria ("con memoria") ci consente di mantenerle le buone abitudini stilando per essa una Tabella di Verità (Figura 5a) capace di sintetizzare il suo funzionamento; l'unica concessione alla novità è la presenza del "fattore tempo", come anticipato rappresentato dall'esponente "i" (istante

presente) e "i+1" (istante futuro); in queste condizioni la tabella può essere letta così: quale sarà il valore dell'uscita (futuro "i+1", Q^{i+1}) se ora (presente, "i") applico una certa combinazione agli ingressi (S^i , R^i)?

In dettaglio: un 1 su S^i (con $R^i=0$) lascia o forzerà l'uscita Q^{i+1} a 1; un 1 su R^i (con $S^i=0$) lascia o forzerà l'uscita Q^{i+1} a 0; coerentemente, quando entrambi gli ingressi non sono attivi ($S^i=0$, $R^i=0$) lo stato del bistabile resta invariato, mentre non ha senso pensare che essi possano essere attivi contemporaneamente: a che valore si assesterà l'uscita Q^{i+1} se entrambi gli ingressi ne reclamano modifiche opposte?

INGRESSI			USCITA
S^i	R^i	Q^{i+1}	
0	0	Q^i	
0	1	0	
1	0	1	
1	1	Vietata	

(a)

INGRESSI				USCITA
S^i	R^i	Q^i	Q^{i+1}	
0	0	0	0	
0	1	0	0	
1	0	0	1	
1	1	0	Vietata	
0	0	1	1	
0	1	1	0	
1	0	1	1	
1	1	1	Vietata	

(b)

Figura 5 - Latch RS asincrono in Logica positiva: Tabella di Verità

La condizione $S^i=1$, $R^i=1$ è detta vietata, non tanto perchè il fornirla possa danneggiare in alcun modo il circuito, ma perchè è ovviamente illogica; a partire da queste considerazioni, è possibile riscrivere la Tabella di Verità (Figura 5b) tenendo presente il modello fondamentale che prevede il rientro in ingresso dell'uscita, dopo esser stato "ritardato", per coinvolgerlo con gli ingressi esterni.

$S^i R^i$					
		00	01	11	10
Q^i	0	0	0	X	1
	1	1	0	X	1

$Q^{i+1} = S^i + \bar{R}^i \cdot Q^i$ $S^i \cdot R^i = 0$

Figura 6 - Latch RS asincrono in Logica positiva: Equazione fondamentale

Da essa, con l'aiuto delle tecniche di minimizzazione di Karnaugh, si può dedurre l'equazione fondamentale del **latch SR** (Figura 6); è interessante notare che, in questo caso, alla formula viene affiancata la precisazione $S^i R^i = 0$: $S^i R^i$ è il mintermine che corrisponde alla combinazione 11 e l'imposizione di ritenerlo nullo ribadisce il concetto che tale combinazione non è ammessa e non si deve fornire.

La magia del progetto digitale sta nel fatto che ogni formula si traduce immediatamente in circuito; anche dall'equazione fondamentale possiamo trarre un primo circuito per il **latch SR** (Figura 7); naturalmente (con riferimento al modello fondamentale delle macchine sequenziali) l'elemento di ritardo Δt rappresenta il concentrato dei tempi di propagazione di tutte le porte coinvolte.

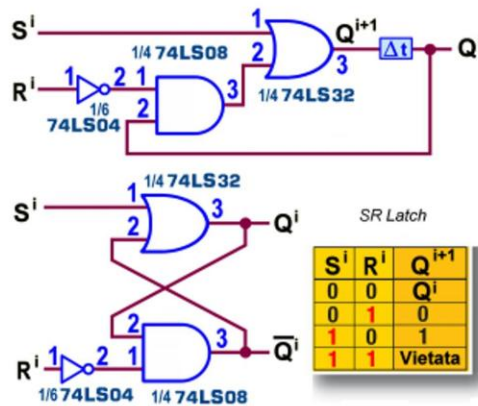


Figura 7 - Latch RS asincrono in Logica positiva: primo schema

Il progetto è a tutti gli effetti un **latch SR asincrono** (dato che lo stato della sua uscita cambierà in funzione dei soli segnali di ingresso e non è presente alcun segnale di clock esterno) in logica positiva (essendo *attivi alti* sia gli ingressi che le uscite), ma appare evidente la sua scarsa praticità: per realizzarlo sono necessari 3 integrati TTL, tra l'altro utilizzati pochissimo (una sola porta per integrato); è necessario dunque prendere provvedimenti ...

La Figura 8 mostra la versione classica di un **latch SR** in logica positiva, realizzato con solo 2 delle 4 NOR contenute nell'integrato 74LS02: sembra incredibile poter disporre della possibilità di memorizzare un bit con sole 2 porte logiche; per la sua struttura detto anche "**Cross-Coupled NOR Gates**".

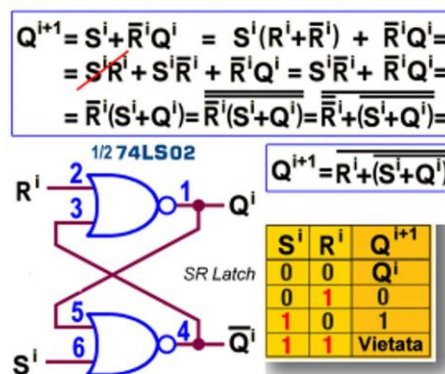


Figura 8 - Latch RS asincrono in Logica positiva: secondo schema

La procedura analitica che ha portato a questa semplificazione non è banale; ecco alcune considerazioni: a) moltiplicare S^i per $(R^i + \overline{R^i})$ lascia inalterato il valore di S^i dato che $(R^i + \overline{R^i})$ è sempre uguale a 1; b) il termine $S^i R^i$ può essere eliminato, essendo uguale a 0, per quanto detto in precedenza; c) i dettagli finali si giustificano ricordando che sottoponendo una variabile booleana a doppia negazione non ne cambia la valore.

Importante sottolineare che la disposizione delle variabili in ingresso non è casuale: perchè funzioni questa versione di **latch SR** deve avere il segnale di Reset (R^i) sulla porta NOR che fornisce l'uscita diretta Q^i e quello di Set (S^i) in ingresso all'altra; in queste condizioni è facile

verificare che il suo funzionamento è in accordo con le specifiche presentate in occasione della stesura della Tabella di Verità.

Cominciamo dalla condizione di riposo (entrambi gli ingressi sono disattivati, $S^i=0$, $R^i=0$): le uscite diretta e negata del dispositivo non vengono influenzate e mantengono perennemente il loro valore attuale; sia per $Q^i=1$ (Figura 9a) che per $Q^i=0$ (Figura 9b) la logica NOR di entrambe le porte ribadisce sulle rispettive uscite lo stesso valore già presente.

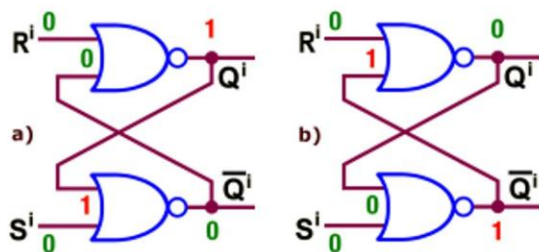


Figura 9 - Latch RS asincrono in Logica positiva: ingressi non attivi

Se viene attivato uno solo degli ingressi il suo valore influenzerà le uscite in accordo con le specifiche di progetto; se si attiva solo l'ingresso di Reset ($S^i=0$, $R^i=1$) e l'uscita diretta è già a 0 ($Q^i=0$, Figura 10a) la logica NOR di entrambe le porte ribadisce lo 0 già presente; se invece l'uscita diretta è a 1 ($Q^i=1$, Figura 10b) l'uscita della NOR N rimane momentaneamente al suo valore attuale, 0, ma riportata in ingresso alla NOR M ne impone il cambiamento, da 1 a 0; a sua volta il nuovo valore 0 della NOR M impone il cambiamento anche all'uscita della NOR N che da 0 passa a 1; da questo momento in poi la nuova situazione si stabilizza per sempre.

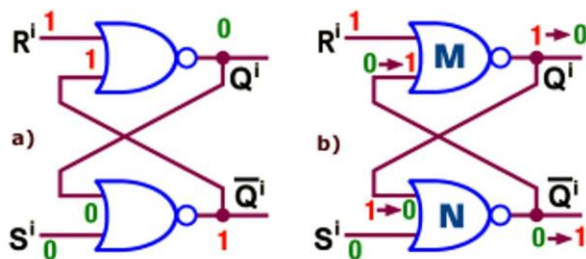


Figura 10 - Latch RS asincrono in Logica positiva: attivo solo Reset

Facile verificare anche la situazione duale che vede attivo solo l'ingresso di Set ($S^i=1$, $R^i=0$): l'uscita diretta sarà lasciata (se già lo è) o forzata a 1, con un meccanismo analogo a quello appena descritto, mantenendo "memoria" di questo valore per sempre (in attesa di una eventuale nuova situazione d'ingresso).

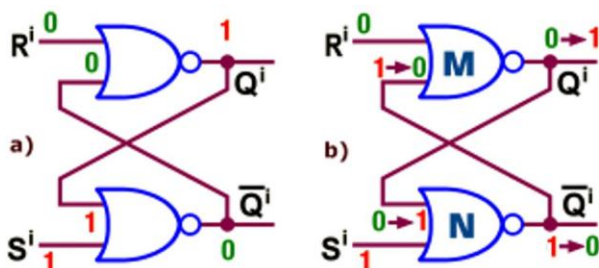


Figura 10BIS - Latch RS asincrono in Logica positiva: attivo solo Set

Interessante anche osservare che succede se si trasgredisce all'ordine e si applica la condizione vietata ($S^i=1, R^i=1$): non scoppia nulla e il circuito non brucia! Semplicemente la presenza di (almeno) un 1 in ingresso alle 2 NOR forza le uscite di entrambe a 0 e ciò è decisamente in contrasto con l'ipotesi che le due uscite di **latch** (o di un **flip-flop**) debbano sempre essere complementari l'una dall'altra.

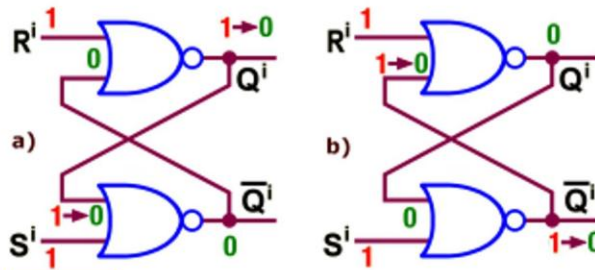


Figura 10TER - Latch RS asincrono in Logica positiva: attivi Set e Reset

L'applicazione della condizione vietata provoca un'altra curiosa (e spiacevole) situazione: se i livelli alti vengono tolti da S e da R contemporaneamente (cosa per altro poco probabile ..) le uscite di entrambe le NOR tenderanno al 1 (per presenza istantanea di uno 0 sui loro 2 ingressi), ma l'inevitabile differenza (pur lieve) tra i rispettivi tempi di propagazione farà sì che una di esse (la più lenta) prima o poi assesti la sua uscita a livello basso; questo significa che non sarà possibile prevedere lo stato successivo del latch.

Nonostante questo circuito sia quello classicamente previsto dalla letteratura per rappresentare un **latch SR asincrono** non si tratta (a mio avviso) della scelta migliore; essendo (come ogni altro stereotipo didattico) progettato in logica positiva esso, lasciato a riposo con ingressi fluttuanti, si pone automaticamente in condizione vietata, "sentendo" degli 1 su entrambi gli ingressi.

Sebbene sia sempre cosa buona e giusta vincolare ad un morsetto dell'alimentazione gli ingressi non collegati, in questo caso (invece dei consueti resistori di pull-up) si rendono necessari collegamenti di pull-down, per tenere a 0 sia Reset che Set in attesa che vengano premuti gli omonimi tasti di controllo (Figura 11).

Con riferimento alla serie LS TTL (a cui appartiene l'integrato **Quad 2-input NOR Gates 74LS02**, utilizzato nello schema di figura) questo fatto introduce un'altro problema che non dovrebbe essere preso alla leggera: la resistenza del resistore di pull-down deve essere più piccola, rispetto a quella tradizionalmente affidata ad un resistore di un pull-up, perché la corrente I_{IL} erogata in ingresso a livello basso è di solito molto più grande (al massimo 400 μA) di quella assorbita in ingresso a livello alto (I_{IH}); non è ragionevole prenderla più grande di 1 kohm, per evitare che la caduta ai suoi capi non sia sufficientemente bassa da essere ritenuta uno 0 logico (di fatto la V_{IL} massima, pari a circa 0,8 V).

D'altro canto non è certo il caso di prenderla troppo piccola, perché la potenza dissipata in calore su di essa ($P_D = V_{cc}^2/R$) aumenta in maniera esponenziale al diminuire del suo valore; in conclusione, con la scelta di figura ($R=560 \text{ ohm}$) quando il pulsante è aperto la tensione rilevata ai capi del resistore è di 0,25 V (ben al di sotto del valore V_{IL} massimo) e quando è chiuso la potenza dissipata è di $(5V)^2/560\Omega = 45 \text{ mW}$.

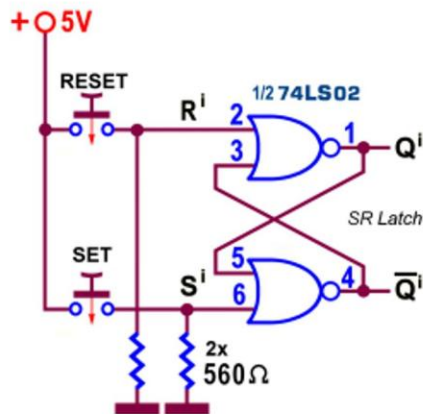


Figura 11 - Latch RS asincrono in Logica positiva: circuito applicativo

Naturalmente il problema non si pone con riferimento alla serie CMOS; utilizzando nello schema di figura l'integrato **Quad 2-input NOR Gates 4001** il valore consigliato per la resistenza del resistore di pull-down può essere liberamente scelto nell'ordine di alcuni kohm.

Una variante molto gradita è quella offerta in *Figura 12*: la presenza del condensatore garantisce la possibilità di assicurare un livello logico prestabilito, subito dopo che il circuito è stato alimentato; nello schema proposto si realizza il "reset automatico" dell'uscita diretta Q^i , ma nulla vieta di spostarlo sull'altro resistore, se si desidera invece forzare Q^i a 1, all'accensione.

L'azzeramento dei dispositivi non appena si fornisce l'alimentazione è sempre tecnica consigliabile, anche nel progetto di circuiti sequenziali più importanti, per evitare che i processi inizino da valori casuali, assunti dalle logiche di memoria per effetto dei loro diversi ritardi intrinseci, come già abbiamo avuto modo di sottolineare per il semplice **latch SR**.

Nell'istante dell'accensione il resistore su S^i "tiene a massa" questo ingresso (assicurandogli il livello 0) mentre il condensatore (inizialmente scarico e quindi con caduta nulla ai suoi capi) lascia tutti i 5V ai capi del resistore su R^i , assicurando su questo ingresso il livello 1; sono presenti le condizioni ($S^i=0$, $R^i=1$) per la commutazione a 0 dell'uscita Q^i .

Da questo istante iniziale il condensatore comincia a caricarsi con costante di tempo $\tau=RC$, sottraendo esponenzialmente tensione ai capi del resistore su R^i , fino a ridurla a 0V; di fatto non è per altro necessario arrivare a questo punto: è sufficiente che la tensione ai capi del resistore si sia ridotta sotto il limite V_{IL} massimo, ritenuto 0 logico dalla porta coinvolta (pari a 0,8V, con riferimento all'integrato **74LS02** LS TTL impiegato nello schema).

Dopo un certo tempo anche l'ingresso R^i torna dunque a 0, ponendo il **latch SR** in condizione di attesa, con uscita stabilmente a 0; possiamo osservare che lasciando passare 2 costanti di tempo la tensione è certamente adatta per essere ritenuta 0 logico per cui, avendo scelto un resistore da 560 ohm, per avere un tempo (per esempio) di 2 ms sarà necessario un condensatore da 1,8 microfarad ($t=2\tau=2RC=2*560*1,8*10^{-6}$); il tempo richiesto non è critico e può essere anche molto più piccolo.

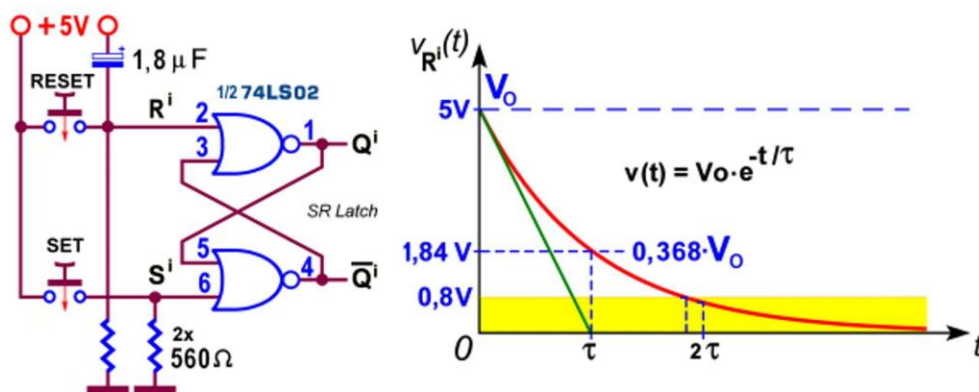


Figura 12 - Latch RS asincrono in Logica positiva: circuito con reset automatico

Nel progetto delle macchine sequenziali diventa importante abituarsi a tracciare il *diagramma temporale* degli eventi coinvolti dal dispositivo; nel nostro caso esso è piuttosto semplice ma non per questo il suo comportamento nel tempo (Figura 13a) è meno interessante: **a)** osserviamo la situazione imprevedibile delle uscite (zona tratteggiata) prima dell'istante t_0 , in cui si applica uno degli ingressi (in questo caso Reset) dopo l'applicazione dell'alimentazione all'istante 0; **b)** di seguito l'uscita Q^i (e la sua complementare negata) si stabilizza ai valori logici suggeriti dalla Tabella di Verità, in presenza o del Set o del Reset; **c)** entrambe le uscite (diretta e negata) vanno a 0 se si ignora il divieto di attivare contemporaneamente entrambi gli ingressi; **d)** supponendo di rilasciare gli stessi nello medesimo istante t_1 lo stato delle uscite torna imprevedibile (zona tratteggiata a destra) attestandosi (come già detto in precedenza) a valori legati alla velocità relativa di una NOR rispetto all'altra.

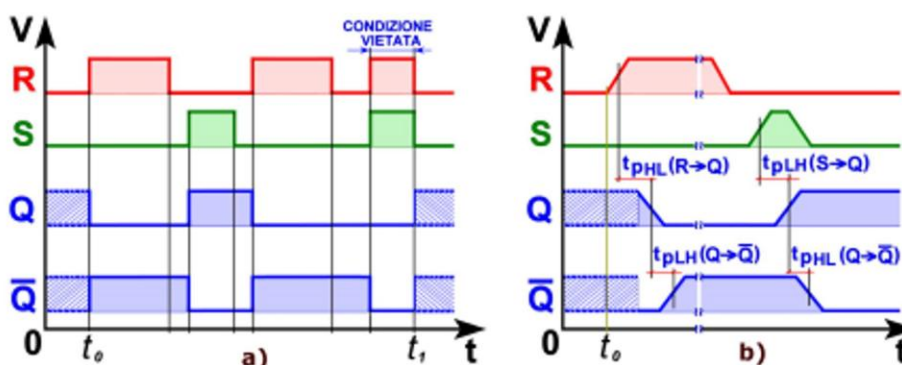


Figura 13 - Latch RS asincrono in Logica positiva: diagramma temporale

Va detto per amor del vero che le transizioni dei segnali coinvolti, da un livello all'altro, non sono né istantanee né simultanee: prendendo come riferimento il fronte di salita di R^i all'istante t_0 ci vorrà qualche nanosecondo perché l'uscita diretta passi a 0 e ulteriori nanosecondi perché l'uscita negata si assesti a 1 (Figura 13b); talvolta i manuali riportano diagrammi di questo tipo, ma in genere forniscono le caratteristiche dinamiche in forma numerica, raccogliendo tempi di propagazione e tempi di transizione in tabelle dedicate.

Torniamo ora brevemente sul problema sollevato dalla necessità di non lasciare fluttuanti, a riposo, gli ingressi del **latch SR** realizzato con porte NOR; se si potesse "rovesciare" la logica d'utilizzo chi comanda sarebbero gli 0 e non gli 1, cosicché lasciando scollegati i due ingressi TTL ("sentendo" entrambi un 1) il **latch SR** si assesterebbe in condizione stabile, d'attesa.

Ed è proprio quello che succede con la nuova versione di questo oggetto (*Figura 14*), decisamente più coerente e consigliabile: la sua disponibilità è giustificata dall'equazione fondamentale, negando 2 volte la quale si perviene alla formula di progetto, realizzato con solo 2 delle NAND contenute nell'integrato 74LS00.

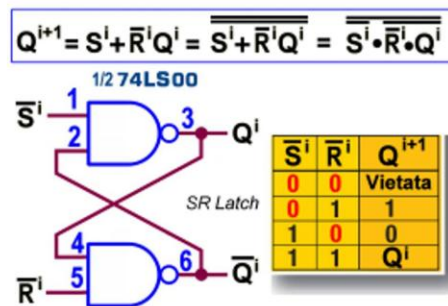


Figura 14 - Latch RS asincrono in Logica negativa: schema

Anche in questo caso è **importantissimo** sottolineare la posizione delle variabili in ingresso: ora è il segnale Set (S^i , attivo basso) quello collegato sulla porta NAND che fornisce l'uscita diretta Q^i , esattamente l'opposto di quanto succede nella versione realizzata con le NOR; anche la Tabella di Verità sarà ovviamente diversa da quella vista in precedenza, evidenziando il fatto che ora gli ingressi sono attivi con lo 0 logico.

In dettaglio: uno 0 su S^i (con $R^i=1$) lascia o forzerà l'uscita Q^i a 1; uno 0 su R^i (con $S^i=1$) lascia o forzerà l'uscita Q^i a 0; con entrambi gli ingressi non attivi ($S^i=1$, $R^i=1$) lo stato del latch resta invariato, mentre la presenza attiva di entrambi gli ingressi ($S^i=0$, $R^i=0$) è ancora da evitare decisamente (condizione vietata) perchè forza entrambe le uscite a 1, contravvenendo all'ipotesi che le due uscite debbano essere complementari l'una dall'altra.

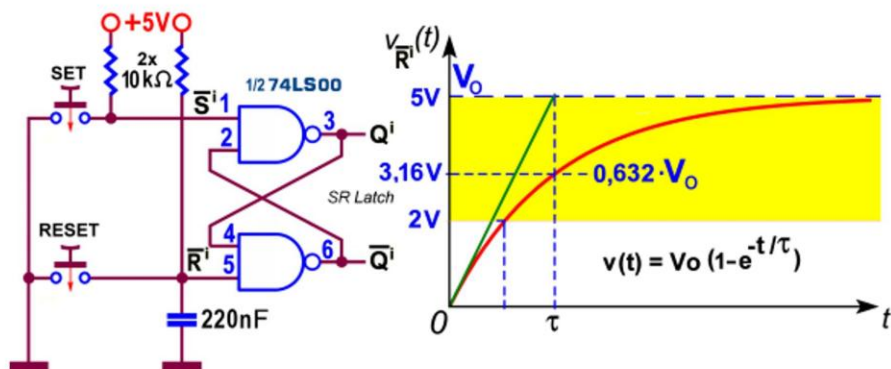


Figura 15 - Latch RS asincrono in Logica negativa: circuito applicativo con reset automatico

La *Figura 15* mostra lo schema applicativo comprensivo di reset automatico, basato sull'integrato LS TTL **Quad 2-input NAND Gates 74LS00**; la presenza dei resistori di pull-up è ora caldamente consigliata (ma non strettamente necessaria) e il valore di resistenza utilizzato è tipicamente dell'ordine delle decine di Kohm, essendo di solito molto piccola (al massimo 20 μA) la corrente I_{IH} assorbita in ingresso a livello alto, tale da imporre una caduta ai suoi capi del tutto compatibile con il livello logico da assicurare (di fatto la V_{IH} minima, pari a circa 2V).

La presenza del condensatore garantisce la possibilità di assicurare il "reset automatico" dell'uscita diretta Q^i ; non appena si accende l'alimentatore il resistore su S^i tiene questo ingresso a V_{cc} (assicurandogli il livello 1) mentre il condensatore, inizialmente scarico, tiene a massa l'ingresso R^i (assicurando il livello 0): in queste condizioni ($S^i=1$, $R^i=0$) l'uscita Q^i commuta a 0.

La successiva carica del condensatore (con costante di tempo $\tau=RC$) riporta lentamente a V_{cc} anche l'ingresso R^i e, non appena la tensione ai capi del condensatore raggiunge il valore V_{TH} minimo (come detto pari a 2V per le NAND LS TTL), viene garantito anche su di esso il desiderato 1 logico, ponendo il **latch SR** nella condizione di attesa, con uscita stabilmente a 0.

La *Figura 15* conferma che l'intervallo di una costante di tempo è ampiamente sufficiente per essere ritenuto 1 logico per cui, con un resistore da 10Kohm, per avere lo stesso tempo di prima (2 ms) sarà necessario un condensatore da 220 nanofarad ($t=\tau=RC=10^4*220*10^{-9}$).

La *Figura 16* mostra il *diagramma temporale* per la versione **latch SR** in logica negativa; le considerazioni sono analoghe a quella già discusse per la versione con le NOR.

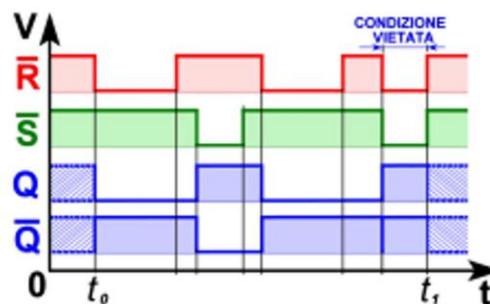


Figura 16 - Latch RS asincrono in Logica negativa: diagramma temporale

La *Figura 17* mostra infine il simbolo logico di entrambe le proposte asincrone del **latch SR**.

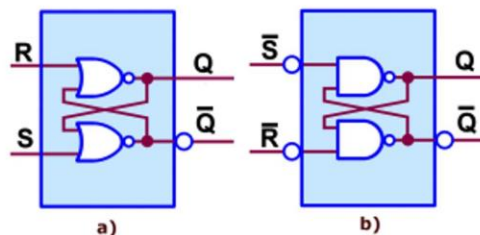


Figura 17 - Latch RS asincrono: Simbolo logico

Il dispositivo oggetto della nostra trattazione è importante perchè consente di disporre di un elemento di memoria (spesso indispensabile in molti progetti digitali) con sole 2 porte logiche invertenti, ampiamente disponibili sia con tecnologia TTL che in tecnologia CMOS; sebbene sia consuetudine utilizzare operatori a 2 ingressi [come le NOR **74LS02** (TTL) e **4001** (CMOS) e le NAND **74LS00** (TTL) e **4011** (CMOS)] può sempre capitare la necessità di avere più ingressi, per esempio 3 [nelle NOR **74LS27** (TTL) e **4025** (CMOS) e nelle NAND **74LS10** (TTL) e **4023** (CMOS)]. Sono comunque a disposizione anche integrati specificatamente progettati per ospitare fino a 4 **latch SR** contemporaneamente.

Latch SR (TTL): 74LS279

Alla serie **TTL LS** appartiene il **74LS279**, definito dai costruttori **Quad S-R Latches**; ospita 4 **latch SR** in logica negativa (cioè realizzati con NAND e soggetti a ingressi attivi bassi, indipendenti tra loro) 2 dei quali hanno un input S aggiuntivo, in aggiunta a quello principale.

Del funzionamento conosciamo ogni dettaglio; la *Figura 18* mostra il suo *pin-out*.

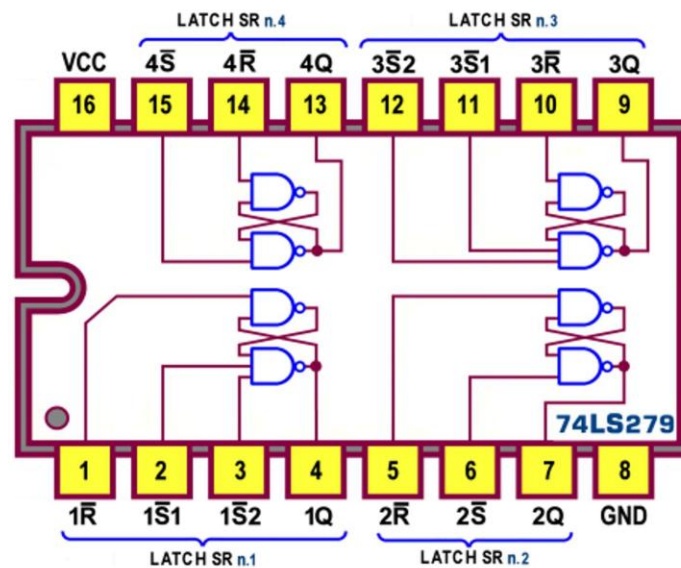


Figura 18 - Quad S-R Latches 74LS279: Pin-out

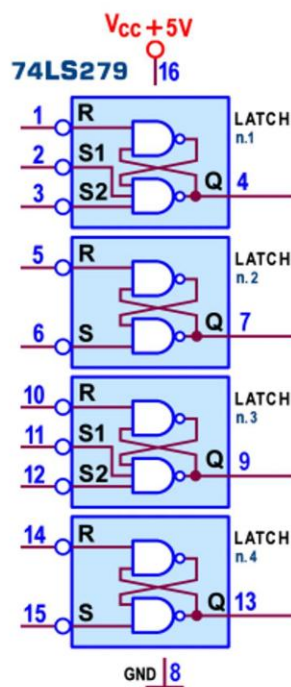


Figura 19 - Quad S-R Latches 74LS279: Schema funzionale

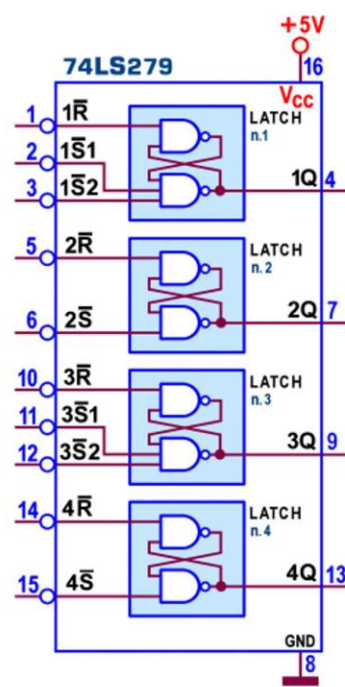


Figura 20 - Quad S-R Latches 74LS279: Schema pratico

Lo *schema funzionale* (Figura 19, *Logic Diagram*) e lo *schema pratico* (Figura 20) sono utili per la stesura dei progetti.

In Figura 21 è disponibile anche il *simbolo logico* predisposto dallo *standard IEEE*.

La corrente massima assorbita dalle uscite del **74LS279** è quella tipica **TTL LS** ($I_{OL}=8\text{ mA}$); la potenza dissipata massima è di soli **35 mW**; il ritardo di propagazione massimo (misurato con carico di **2Kohm/15pF**) tra gli ingressi S e uscita Q è di **22 ns** (t_{PLH}) e di **15 ns** (t_{PHL}) e un po' più grande (**27 ns** per entrambe le transizioni) tra gli ingressi R e uscita Q.

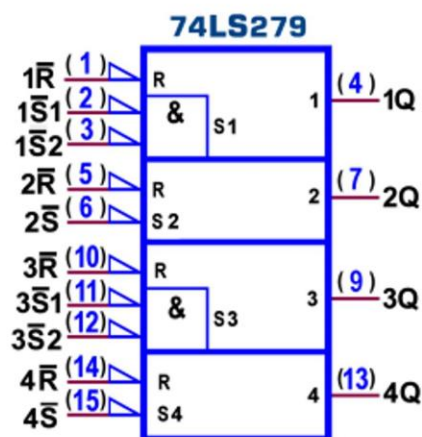


Figura 21 - Quad S-R Latches 74LS279: Simbolo logico ANSI/IEEE Std. 91-1984

Latch SR (CMOS): 4043, 4044

Alla serie **CMOS** appartengono il **4043** e il **4044**; entrambi contengono 4 **latch SR** e dispongono di un ingresso di controllo attivo alto (detto *Enable*); quando questa linea è a 0 logico gli elementi di memoria SR sono effettivamente scollegati dalle rispettive uscite (forzate in alta impedenza); questa caratteristica permette a questi componenti di concorrere in strutture "a bus", molto diffuse nelle logiche programmabili; al di fuori di esse *Enable* va dunque collegato a 1.

Naturalmente le regole teoriche descritte in precedenza sono perfettamente applicabili ad entrambi i dispositivi e, in particolare, con entrambi gli ingressi disattivi la memoria non cambia e mantiene il valore precedente, mentre deve essere evitata la situazione con entrambi gli ingressi attivi, che forzano le memorie in una situazione illogica (con entrambe le uscite forzate allo stesso valore).

Nel **4043**, definito dai costruttori **Quad NOR S-R Latches**, i 4 **latch SR** sono realizzati con delle NOR e gli ingressi S e R (indipendenti tra loro) sono attivi alti (logica positiva); la condizione vietata è dunque SR=11 e va assolutamente evitata.

L'utilizzo è documentato dalla Figura 22 (*pin-out*), dalla Figura 23 (*schema funzionale, Logic Diagram*) e dalla Figura 24 (*schema pratico*).

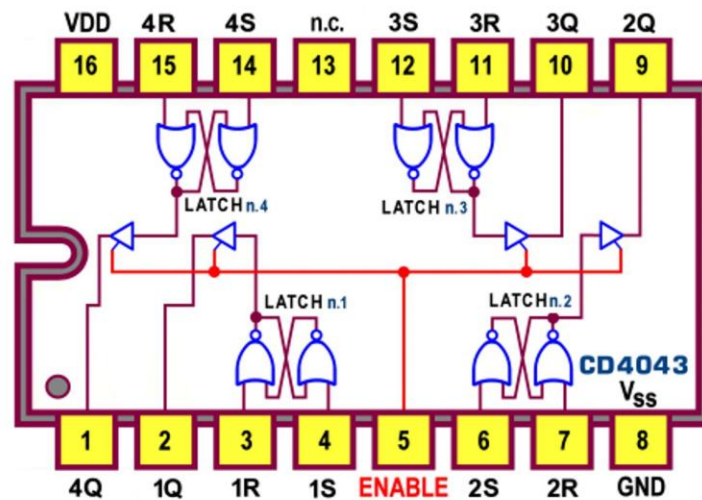


Figura 22 - Quad 3-State NOR R/S Latches 4043: Pin-out

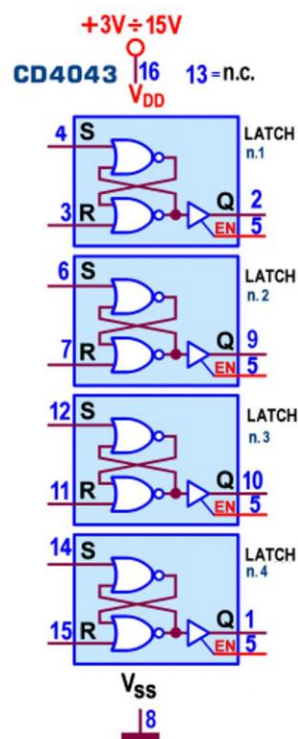


Figura 23 - Quad 3-State NOR R/S Latches 4043: Schema funzionale

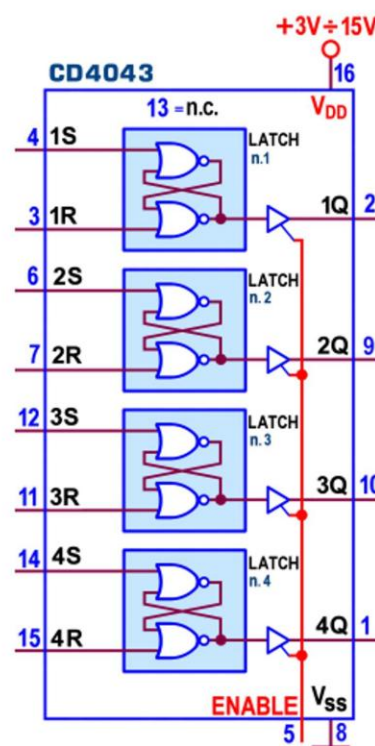


Figura 24 - Quad 3-State NOR R/S Latches 4043: Schema pratico

I 4 latch SR del **4044**, detto **Quad NAND S-R Latches**, i sono realizzati con delle NAND e gli ingressi S e R ora sono attivi bassi (logica negativa); la condizione vietata è ora $SR=00$ e va

applicata per nessun motivo; a suo supporto è, come di consueto, disponibile la *Figura 25* (*pin-out*), la *Figura 26* (*schema funzionale, Logic Diagram*) e la *Figura 27* (*schema pratico*).

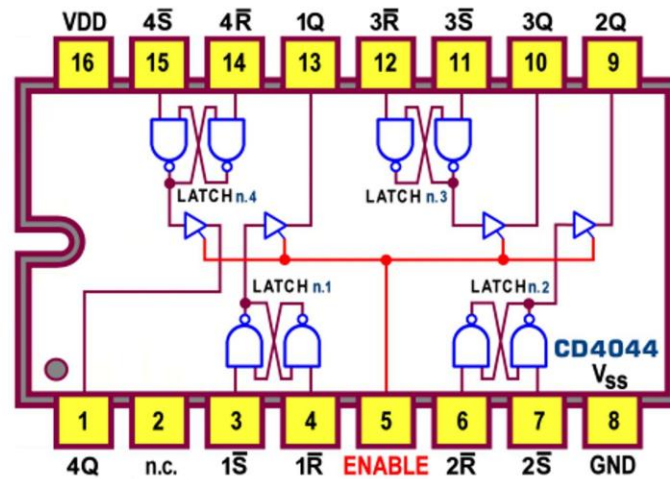


Figura 25 - Quad 3-State NAND R/S Latches 4044: Pin-out

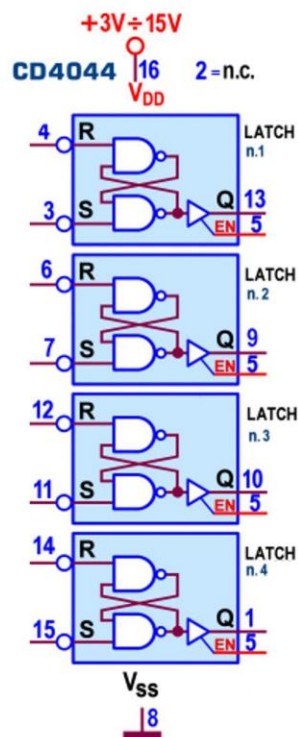


Figura 26 - Quad 3-State NAND R/S Latches 4044: Schema funzionale

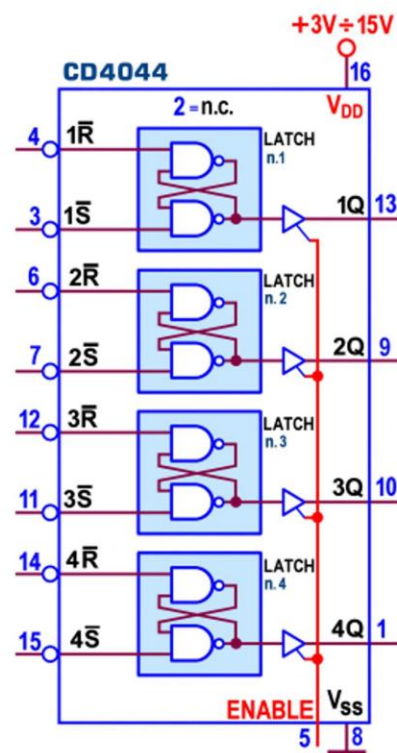


Figura 27 - Quad 3-State NAND R/S Latches 4044: Schema pratico

Come per gli altri componenti della famiglia **CMOS** la *potenza dissipata* è trascurabile mentre per entrambi gli integrati il *ritardo di propagazione* tra ingressi S o R verso l'uscita Q

(*Propagation Delay Time*, t_{PHL} e t_{PLH} , rilevato con carico di **200kohm/50pF**) è al massimo di **350ns** ($V_{DD}=5V$), **175ns** ($V_{DD}=10V$) e **100ns** ($V_{DD}=15V$); i valori *tipici* sono sostanzialmente la metà.

La definizione dei **latch SR asincroni** costituisce la base fondamentale per il progetto di memorie più evolute, supportate da segnali di sincronismo, come i **flip-flop**; l'argomento è molto vasto e richiede ulteriori approfondimenti; la prossima puntata completeremo la rassegna dedicata al **latch SR** con i circuiti applicativi asincroni e con lo studio delle sue varianti sincronizzate.