

1.4 Elettronica di base - Logica digitale

Segnali digitali

Osserviamo che in alcuni casi i segnali che trattiamo sono per natura segnali discreti, come ad esempio lo stato di un interruttore. In questi casi l'uso dell'elettronica digitale, circuiti che trattano segnali con solo due stati possibili, 0 e 1, è naturale. In altri casi invece il segnale è per natura analogico: un segnale audio oppure una tensione ricavata da un sensore che misura una grandezza fisica (temperatura, luminosità...). Vedremo che in molti casi sarà conveniente tramite un opportuno circuito, l'ADC (Analog to Digital Converter), convertire segnali analogici in segnali digitali. Questo perché diventa possibile effettuare calcoli sui dati (tramite un computer), immagazzinare l'informazione facilmente sotto forma di numero e trasmettere il segnale senza essere soggetti alla degradazione del segnale a causa del rumore, inerente in ogni sistema analogico.

È possibile convertire un segnale analogico in una serie di numeri che rappresentano l'ampiezza ad un dato tempo e trasmettere questi numeri come segnali digitali (questa tecnica si chiama PCM, Pulse Code Modulation). Il segnale non viene degradato dal rumore fino a che ciò non impedisce il riconoscimento degli 0 ed 1 offrendo una maggior resistenza rispetto ad un segnale analogico. I CD audio, ed in generale tutte le tecnologie di trasmissione di audio digitale come l'HDMI usano questa tecnica.

Vediamo ora quali sono gli elementi di base che caratterizzano l'elettronica digitale

Stati logici: HIGH e LOW

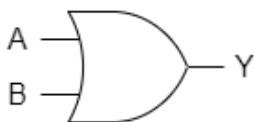
Chiamiamo i due stati possibili dei circuiti elettronici digitali HIGH e LOW: questi stati corrispondono a dei valori di tensione che dipendono dalla famiglia logica in uso, ma generalmente cadono rispettivamente vicino alla tensione di alimentazione e GND. Questi stati sono collegati in qualche modo agli stati booleani VERO e FALSO. Se si utilizza la convenzione HIGH = TRUE il segnale si dice *active HIGH*, mentre se si sceglie la convenzione LOW = TRUE il segnale viene detto *active LOW* ed è usualmente indicato con una linea sopra al nome del segnale (ad esempio \overline{RESET} o a volte *nRESET*). Questa situazione può portare a confusione perché spesso nei circuiti segnali diversi usano convenzioni diverse.

Funzioni logiche elementari

Vediamo le funzioni che sono realizzabili tramite *gate* logici ovvero dispositivi la cui uscita dipende solo dallo stato degli ingressi (logica combinatoria). Per i *gate* viene utilizzata la convenzione *active HIGH*.

Riportiamo di seguito il simbolo e la tabella di verità delle funzioni logiche di base: OR, AND, NOT, NAND, NOR e XOR.

Gate OR

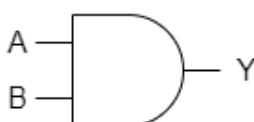


A	B	Y
0	0	0
1	0	1
0	1	1
1	1	1

L'uscita di un gate OR è HIGH se uno qualsiasi degli ingressi (o entrambi) è HIGH. (effettua l'operazione o booleana)

A sinistra è rappresentato un gate OR a due ingressi.

Gate AND

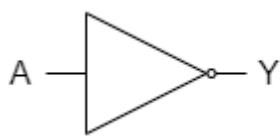


A	B	Y
0	0	0
1	0	0
0	1	0
1	1	1

L'uscita di un gate AND è HIGH solo se tutti gli ingressi sono HIGH. (effettua l'operazione e booleana)

Sia i gate OR che i gate AND sono disponibili con 2,3,4 e a volte anche più ingressi.

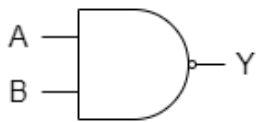
Gate NOT (inverter)



A	Y
0	1
1	0

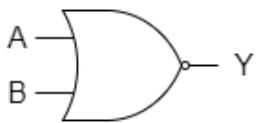
Il gate NOT inverte lo stato logico: per un ingresso HIGH l'uscita sarà LOW e viceversa (effettua la negazione booleana)

Gate NAND e NOR



A	B	Y
0	0	1
1	0	1
0	1	1
1	1	0

La funzione NOT può essere combinata con un gate AND per produrre un gate NAND e con un gate OR per produrre un gate NOR. Questi due gate sono più utilizzati dei gate AND e OR perché permettono di costruire qualsiasi altro gate logico combinando gate NOR o NAND in maniera opportuna.



A	B	Y
0	0	1
1	0	0
0	1	0
1	1	0

Gate XOR



A	B	Y
0	0	0
1	0	1
0	1	1
1	1	0

L'uscita di un gate XOR è HIGH se e solo se uno dei due ingressi è HIGH. Questo gate ha sempre solo due ingressi.

La funzione XOR è equivalente alla addizione modulo-2 di due bit e quindi viene usata nei sommatore.

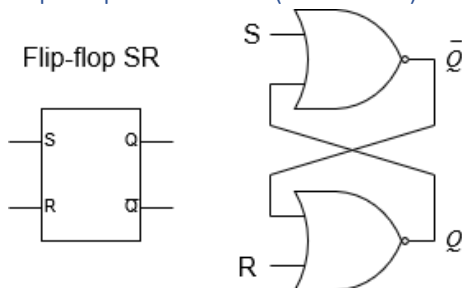
Circuiti integrati – gate logici

I gate logici illustrati sopra possono essere realizzati con circuiti discreti con l'impiego di transistor BJT o MOSFET ma più comunemente sono realizzati da circuiti integrati dedicati. La famiglia logica più diffusa è la famiglia 74xxxx (e in particolare la 74HCxx) che impiega tecnologia CMOS e quindi è a basso consumo di corrente. Ad esempio, il circuito integrato 74HC00 contiene 4 gate NAND, il 74HC04 6 gate NOT, il 74HC08 4 gate AND e il 74HC32 4 gate OR. Sempre nella stessa famiglia sono disponibili numerosi circuiti integrati che realizzano funzioni logiche più complesse¹.

Logica sequenziale: flip-flop

Introduciamo i flip-flop che permettono, in combinazione con i gate logici, di realizzare funzioni logiche molto complesse. Sono dispositivi che agiscono da memorie da 1 bit, ovvero conservano il loro stato. Permettono di realizzare contatori, registri a scorrimento e molte altre funzioni non realizzabili ricorrendo ai soli gate. Vediamo ora i tipi più diffusi di flip-flop

Flip-flop o latch SR (asincroni)



S	R	Q
0	0	=
1	0	1
0	1	0
1	1	!

Il più semplice flip-flop SR si può realizzare con due gate NOR collegati come in figura.

Il circuito ha due uscite simmetriche (l'una è la negazione dell'altra) e quindi ha 2 stati possibili: le due uscite non potranno mai avere lo stesso valore.

¹ Fino ad unità aritmetico-logiche, registri a scorrimento... per un elenco vedi https://en.wikipedia.org/wiki/List_of_7400-series_integrated_circuits

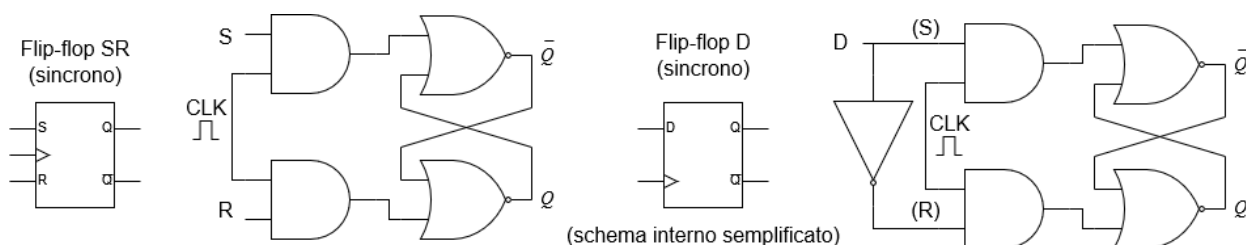
Ora chiamiamo gli ingressi rispettivamente SET e RESET. Notiamo che se gli ingressi sono entrambi LOW l'uscita mantiene il valore corrente: abbiamo ottenuto un dispositivo con memoria. Per cambiare lo stato basta portare momentaneamente uno degli ingressi a HIGH: se portiamo SET a HIGH l'uscita Q diventerà HIGH mentre se portiamo RESET a HIGH l'uscita Q diventerà LOW.

Questo circuito pone però una serie di problemi: non è possibile stabilire a priori lo stato iniziale del circuito, ovvero l'uscita all'applicazione dell'alimentazione e se i due ingressi sono simultaneamente HIGH il circuito diventa instabile e non è possibile prevedere quale stato assumerà l'uscita.

Il flip-flop SR che abbiamo visto qui funziona con logica *active HIGH*; è possibile costruire lo stesso circuito con logica *active LOW* impiegando gate NAND al posto dei gate NOR (ed è la configurazione più diffusa).

Flip-flop SR sincroni e D

Una forma molto più usata nei circuiti digitali sono i flip-flop sincroni in cui lo stato può cambiare solo ad intervalli determinati, stabiliti da un segnale di clock. Possiamo ottenere un flip-flop SR sincrono semplicemente aggiungendo due gate AND agli ingressi di quello asincrono che fanno in modo di poter attivare gli input solo quando il clock è HIGH.



Questa è una approssimazione del comportamento dei flip-flop sincroni, in quanto la maggior parte permette il cambio di stato solo in corrispondenza di un fronte (*edge*) dell'onda di clock, che può essere sia quello di salita (*rising edge*) che quello di discesa (*falling edge*).

Tabella di verità
Flip-flop D

D	Q_{n+1}
0	0
1	1

Il tipo più usato di flip-flop sincrono è il tipo D in cui è presente un solo ingresso D corrispondente al SET, mentre l'ingresso RESET è sempre l'opposto dell'ingresso D. La funzione di questo flip-flop, illustrata dalla tabella di verità a sinistra, è estremamente semplice: ad ogni ciclo di clock il valore dell'ingresso D viene copiato all'output Q che mantiene lo stato fino al prossimo ciclo di clock.

La maggior parte dei flip-flop di tipo D però ha una struttura leggermente diversa, ovvero ha anche linee separate di SET e CLEAR (stessa funzione del RESET), particolarmente utili per impostare uno stato certo all'accensione del circuito.

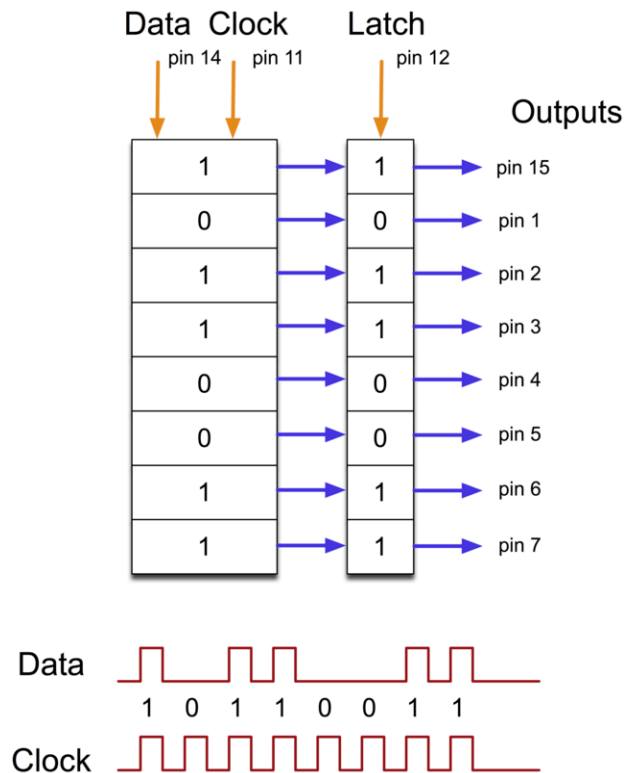
I flip-flop di tipo D hanno moltissimi usi e stanno alla base di dispositivi come contatori e registri a scorrimento (dispositivi di memoria). Vari tipi di flip-flop sono disponibili come circuiti integrati nella famiglia 74xxxx

Vediamo ora un paio di dispositivi che realizzano funzioni logiche più complesse combinando gli elementi visti finora. Successivamente introdurremo ancora altre funzioni logiche ma sempre in combinazione con l'uso di un microcontrollore.

Registri a scorrimento (74HC595)

La base di un registro a scorrimento (*shift register*) consiste in una serie di flip-flop collegate a cascata (ovvero l'uscita della prima è collegata all'entrata della seconda e così via): abbiamo sostanzialmente un dispositivo che accetta dati seriali e può dare in uscita dati paralleli (SIPO). Aggiungendo poi un flip-flop per ogni uscita che agisce da latch, ovvero memorizza lo stato fino ad un certo evento si ottiene il dispositivo

finale. Un dispositivo di questo genere, con 8 uscite, è il 74HC595. Per una spiegazione più dettagliata sulla struttura dei registri a scorrimento si rimanda al *datasheet* del componente.



Multiplexer e demultiplexer (74HC4052)

I multiplexer sono dispositivi che presentano un certo numero di ingressi (generalmente 4, 8 o 16) e una singola uscita. Un indirizzo binario (applicato ad ingressi aggiuntivi) viene usato per scegliere quale ingresso viene replicato dall'uscita. Il demultiplexer effettua l'opposto. I dispositivi che realizzano questa funzione possono essere adatti a gestire segnali digitali od analogici. Il 74HC4052 contiene due multiplexer 4:1 che permettono di gestire segnali analogici (o digitali di conseguenza) ed è reversibile, ovvero può agire anche da demultiplexer.

