

# BASTA USARE LA LOGICA

In questo Workshop introduciamo alcuni dei principi alla base della moderna elettronica digitale, che verificheremo con un semplice circuito.

**N**ei Workshop precedenti abbiamo parlato molto spesso di problematiche e argomenti legati all'elettronica; tuttavia la nostra attenzione è stata per lo più dedicata ai concetti del mondo **analogico**. Con questo articolo, invece, inizieremo a conoscere alcune delle basi fondamentali dell'**elettronica digitale** e cioè di un insieme di tecnologie che consentono di elaborare e gestire informazioni in forma numerica.

## DAL CONTINUO AL DISCRETO >>>

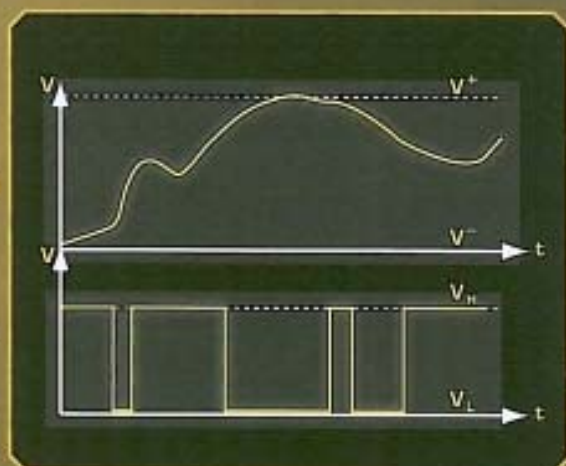
Come vedremo nel corso dei prossimi fascicoli, l'elettronica digitale è basata su principi molto diversi rispetto a quanto abbiamo osservato finora. Paragonare, infatti, il mondo analogico a quello digitale è, per certi versi, molto simile a mettere a confronto i concetti matematici di **continuo** o di **discreto**. Ciò deriva dal fatto che i sistemi analogici lavorano generalmente con **segnali di tipo continuo**, ossia che possono evolvere nel tempo

assumendo tutti i valori di tensione compresi tra limiti definiti

(solitamente uno superiore e uno inferiore). Un

esempio di segnale di questo tipo è mostrato nel primo grafico della figura sopra a destra. Le tecnologie digitali, al contrario,

lavorano sulla base di livelli logici, ossia tensioni (o range di tensioni) alle quali viene attribuito un numero limitato e preciso di significati numerici (vedi il grafico in basso della figura sotto).

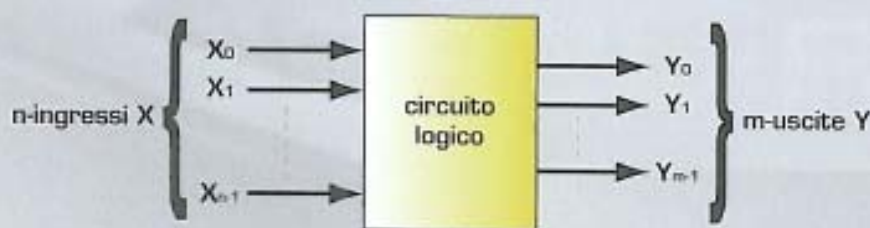


Un segnale analogico (in alto) assume valori compresi tra i limiti di tensione  $V^+$  e  $V^-$ . Quello digitale (in basso), può assumere esclusivamente i valori di tensione  $V_H$  e  $V_L$ .

## I SISTEMI LOGICI >>>

L'elettronica digitale getta le sue fondamenta su un particolare tipo di rappresentazione delle informazioni, la **codifica binaria**, nella quale i segnali e le variabili evolvono nel tempo assumendo solamente due valori: un livello 'alto' (indicabile con '1') e un livello 'basso' (indicabile con '0').

Un 'wafer' di silicio da cui vengono ricavati i moderni circuiti digitali.



*I circuiti logici possono essere paragonati a funzioni matematiche, che elaborano 'n' parametri forniti come ingressi, per produrre una combinazione di 'm' uscite.*

Per capire meglio come vengono trattati 'elettricamente' i livelli logici, ti rimandiamo al FOCUS DN delle prossime pagine. Ma come è possibile sfruttare una tecnologia 'semplice', in grado di gestire esclusivamente '0' e '1', per ottenere sistemi complessi come un calcolatore o il 'cervello' di un robot? Alla base delle moderne tecnologie digitali vi è un particolare ramo della matematica chiamata **algebra di Boole**, la quale mette a disposizione regole e operatori capaci di trattare variabili che assumono i **valori logici 'vero'** (equivalente a '1') e **'falso'** ('0'). Ovviamente, in quest'opera non ci addentreremo nelle questioni più strettamente matematiche, ma ci limiteremo a osservare come gli operatori offerti dall'algebra booleana possano essere applicati per i nostri scopi. In particolare, questi strumenti matematici ci permetteranno di realizzare

fisicamente quelli che chiameremo **'circuiti logici'** e che possiamo classificare in **tre distinte categorie**: i circuiti logici **combinatori**, i circuiti logici **sequenziali** e i circuiti **di memoria**. La differenza sostanziale tra questi tre tipi di sistemi deriva dai principi di funzionamento. **I circuiti combinatori e sequenziali** possono essere **paragonati a 'funzioni matematiche' dotate di un numero 'n' di ingressi binari** (ognuno dei quali sarà indicato con  $X_n$ ) e di un numero **'m' di uscite, anch'esse binarie** (indicate con  $Y_m$ ). Sono, quindi, accomunati dalla capacità di **elaborare gli ingressi** per produrre un **risultato in uscita** (vedi immagine in alto). Vi è, tuttavia, una importante differenza nel loro processo di funzionamento: **nei circuiti logici combinatori**, infatti, **l'uscita dipende esclusivamente dalle variabili d'ingresso**. Al contrario,

all'interno dei circuiti sequenziali sono presenti **elementi di memoria** (chiamati **flip-flop** e **latch**) che tengono traccia dell'evoluzione del sistema. **In essi, quindi, l'uscita dipende sia dall'ingresso che dall'evoluzione storica (stato attuale) del sistema stesso**. Detto questo, è importante tenere presente che il modo in cui evolvono i sistemi sequenziali non è casuale o universalmente determinato, ma deve essere deciso in fase di progettazione, stabilendo **l'insieme degli stati desiderati** e le condizioni che causano il passaggio da uno stato all'altro (**transizioni**), ma di questo ci occuperemo in futuro. **I circuiti di memoria**, invece, sono circuiti digitali il cui unico scopo è quello di **memorizzare e restituire informazioni binarie** (in modo praticamente analogo alle RAM dei computer). Iniziamo il nostro 'viaggio' nell'elettronica digitale

### LE TABELLE DI VERITÀ»»

Le **tabelle di verità** sono strumenti utilizzati per visualizzare il comportamento delle reti logiche combinatorie. Sono **fondamentalmente suddivise in quattro riquadri**: in quelli **superiori** sono visualizzati i nomi delle variabili di ingresso (a sinistra) e di uscita (a destra). **Nei riquadri inferiori**, invece, sono mostrati i valori assunti. In particolare nel riquadro in basso a sinistra devono essere elencate **tutte le combinazioni delle variabili binarie di ingresso** [per 'n' variabili sono necessarie  $2^n$  righe], mentre in basso a destra sono indicate, riga per riga, le uscite corrispondenti.

$X_2$	$X_1$	$X_0$	$Y_1$	$Y_0$
0	0	0	0	0
0	0	1	0	1
0	1	0	1	0
0	1	1	1	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	0

3 combinazioni di ingressi      3 combinazioni di uscite



**porta 'NOT'**

operatore unario di 'negazione logica' (nega la variabile di ingresso)

X	Y
0	1
1	0

matematicamente  $Y = \bar{X}$



**porta 'AND'**

Operatore binario 'prodotto logico' (è vero solo se tutti gli ingressi sono veri, falso altrimenti)

X <sub>1</sub>	X <sub>2</sub>	Y
0	0	0
0	1	0
1	0	0
1	1	1

matematicamente  $Y = X_1 \cdot X_2$



**porta 'OR'**

Operatore binario 'somma logica' (è vero se almeno uno degli ingressi è vero, falso altrimenti)

X <sub>1</sub>	X <sub>2</sub>	Y
0	0	0
0	1	1
1	0	1
1	1	1

matematicamente  $Y = X_1 + X_2$



**porta 'XOR'**

Operatore binario 'OR esclusivo' (è vero solo se gli ingressi sono differenti tra loro, falso altrimenti)

X <sub>1</sub>	X <sub>2</sub>	Y
0	0	0
0	1	1
1	0	1
1	1	0

matematicamente  $Y = X_1 \oplus X_2$

**LE PORTE LOGICHE FONDAMENTALI**

In questo box sono elencate le sette porte logiche fondamentali utilizzate in elettronica digitale. Sei di esse sono operatori binari, ossia richiedono due ingressi, mentre una (NOT) è unaria, ossia opera su un solo ingresso. In commercio sono reperibili anche porte 'composte', come AND o OR a più di due ingressi. Ricordiamo che '1' può essere letto come 'vero' e '0' come 'falso'.

**porta 'NAND'**

Operatore binario 'prodotto logico negato' (è vero solo se almeno uno degli ingressi è falso, falso altrimenti)

X <sub>1</sub>	X <sub>2</sub>	Y
0	0	1
0	1	1
1	0	1
1	1	0

matematicamente  $Y = \overline{X_1 \cdot X_2}$

**porta 'NOR'**

Operatore binario 'somma logica negata' (è vero solo se tutti gli ingressi sono falsi, falso altrimenti)

X <sub>1</sub>	X <sub>2</sub>	Y
0	0	1
0	1	0
1	0	0
1	1	0

matematicamente  $Y = \overline{X_1 + X_2}$

**porta 'XNOR'**

Operatore binario 'OR esclusivo negato' (è vero solo se i due ingressi sono uguali, falso altrimenti)

X <sub>1</sub>	X <sub>2</sub>	Y
0	0	1
0	1	0
1	0	0
1	1	1

matematicamente  $Y = \overline{X_1 \oplus X_2}$

partendo proprio dalla teoria dei circuiti logici combinatori.

**LE PORTE LOGICHE ELEMENTARI >>>**

I sistemi logici elettronici sono fondamentalmente reti di microsistemi basati su transistor (BJT o MOSFET, in base alla famiglia del circuito), che implementano gli operatori dell'algebra booleana. Tali circuiti sono noti con il nome di 'porte logiche'. Le porte logiche sono quindi i 'mattoni' alla base dei circuiti combinatori. Vediamo ora l'elenco degli operatori booleani e delle porte corrispondenti. Il loro funzionamento è mostrato ricorrendo a un semplice ausilio grafico chiamato **tabella di verità**. Le tabelle di verità visualizzano il comportamento delle reti logiche combinatorie e delle corrispondenti espressioni logiche (dalla singola porta semplice alla rete più complessa) associando a ogni combinazione possibile degli ingressi l'uscita, o le uscite, corrispondenti. Per ogni porta (vedi box a fianco) vengono presentati, oltre alla tabella di verità, anche il simbolo circuitale corrispondente e l'operatore matematico che utilizzeremo nella scrittura delle espressioni.

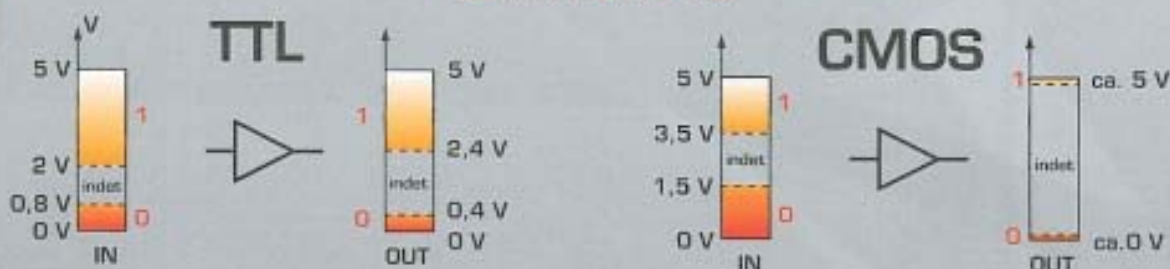
F O C U S O N

LE FAMIGLIE LOGICHE >>>

I circuiti logici (tra i quali troviamo anche le porte logiche) sono costruiti a partire da reti di transistor opportunamente configurate. Sebbene sia possibile realizzare 'manualmente' questo tipo di dispositivi utilizzando proprio dei semplici transistor (come è stato visto nel fascicolo 15, dove abbiamo realizzato una semplice porta NOT), sul mercato è reperibile una ricca offerta di circuiti integrati per l'elettronica digitale. Le porte logiche a cui faremo ricorso saranno proprio di tipo integrato e per poterle riconoscere sarà necessario avere una base, seppur elementare, della nomenclatura commerciale utilizzata. I nomi dei circuiti logici integrati più diffusi sono solitamente espressi secondo 'codici' del tipo **XXYYZZZ**. Il prefisso numerico **XX** indica il tipo di integrato. Generalmente per gli integrati commerciali si usano chip di tipo '74' o '40'. La seconda porzione codice **YY**, invece, è alfabetica ed esprime la sottofamiglia tecnologica dell'integrato. La dicitura 'LS', ad esempio, caratterizza le porte TTL (basate su BJT) di tipo 'Low Power Schottky' o, sempre come esempio, la sigla 'HC' identifica i circuiti 'High Speed CMOS' (una sottofamiglia a tecnologia MOSFET) e così via. La parte terminale del codice [ZZZ], invece, è di tipo numerico e permette di riconoscere la funzione logica del circuito. Per citare due esempi che utilizzeremo nello StepbyStep, se **ZZZ = '04'**

l'integrato conterrà delle porte NOT, mentre se **ZZZ = '32'**, esso racchiuderà delle porte OR. Nelle prossime pagine, ad esempio, utilizzeremo il circuito **74HC04**, ossia un integrato che mette a disposizione porte NOT di tipo High Speed CMOS. La presenza di diverse famiglie logiche, tuttavia, è anche fonte di numerose problematiche di interfacciamento. Ad esempio, se consideriamo le due più diffuse, ossia la TTL e la CMOS, vi è una netta differenza nelle specifiche di definizione dei range di tensione dei livelli logici. Non è quindi detto che circuiti 'ibridi' realizzati con dispositivi TTL e CMOS funzionino correttamente. Come esempio pratico riportiamo qui sotto, i livelli logici degli standard TTL e CMOS (serie 4xxx). Si osserva come le porte con tecnologia TTL interpretino come '1' tutto ciò che è al di sopra dei 2 V, e come '0' ciò che è al di sotto degli 0,8 V. In uscita, inoltre, le specifiche TTL prevedono un range da 2,4 a 5 V per i valori alti e da 0 a 0,4 V per quelli bassi. I livelli CMOS, invece, sono completamente diversi. Se collegassimo una porta CMOS 4xxx in cascata a una TTL, potrebbe verificarsi il caso in cui la tensione di uscita TTL si trovi a cadere in una zona di indeterminazione CMOS, con un conseguente malfunzionamento (in questa zona, infatti, il circuito non riconoscerebbe lo stato di ingresso).

LIVELLI LOGICI DELLE FAMIGLIE TTL E CMOS 4XXX (ALIMENTAZIONE 5 V)

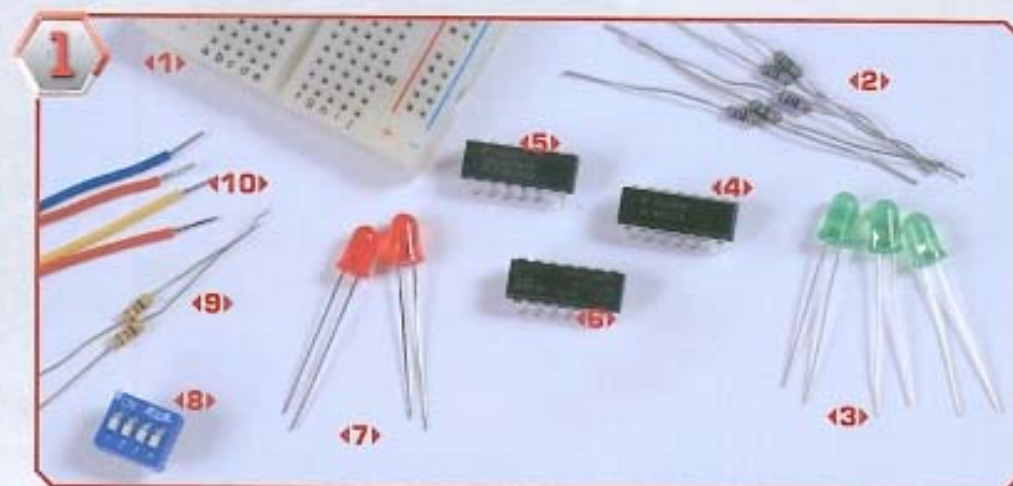


# STEP by STEP

## PROVIAMO LE PORTE LOGICHE

Prima di procedere, procuriamoci gli elementi necessari:

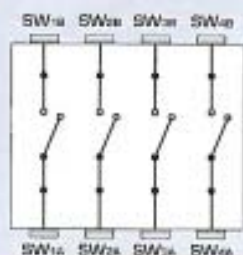
- 1) una breadboard
- 2) cinque resistori da 220 ohm (R)
- 3) tre LED verdi ( $D_{NOT}$ ,  $D_{AND}$ ,  $D_{OR}$ )
- 4) un 74HC32 ( $IC_{OR}$ )
- 5) un 74HC04 ( $IC_{NOT}$ )
- 6) un 74HC08 ( $IC_{AND}$ )
- 7) due LED rossi ( $D_1$ ,  $D_2$ )
- 8) uno switch multiplo di tipo DIL (SW)
- 9) due resistori da 4,7 kohm ( $R_{H1}$ )
- 10) fili per breadboard



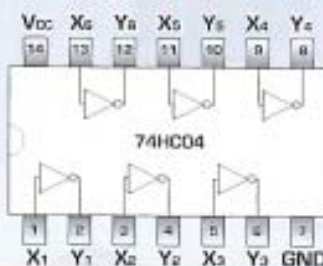
Nell'esperimento di oggi utilizzerai tre circuiti integrati che racchiudono tre tipi diversi di porte logiche. Come puoi vedere dagli schemi sottostanti, ogni integrato racchiude al suo interno più di una porta: dovrai, quindi, rispettare la piedinatura. Nel primo schema puoi vedere anche la struttura interna dello switch DIL, un dispositivo che racchiude al suo interno quattro interruttori.



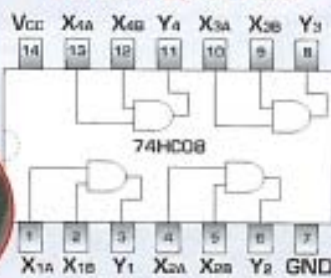
microswitch DIL (4 vie)



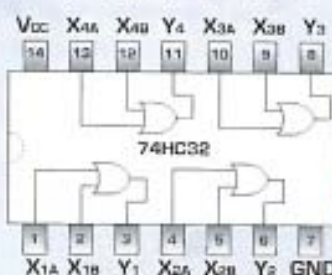
74HC04 (hex NOT)

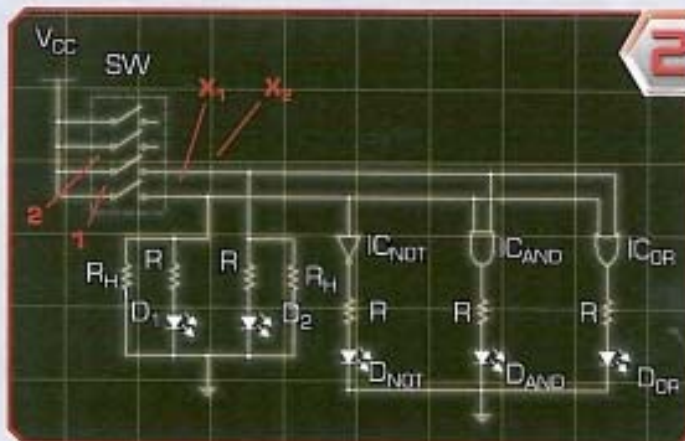


74HC08 (quad AND)



74HC32 (quad OR)

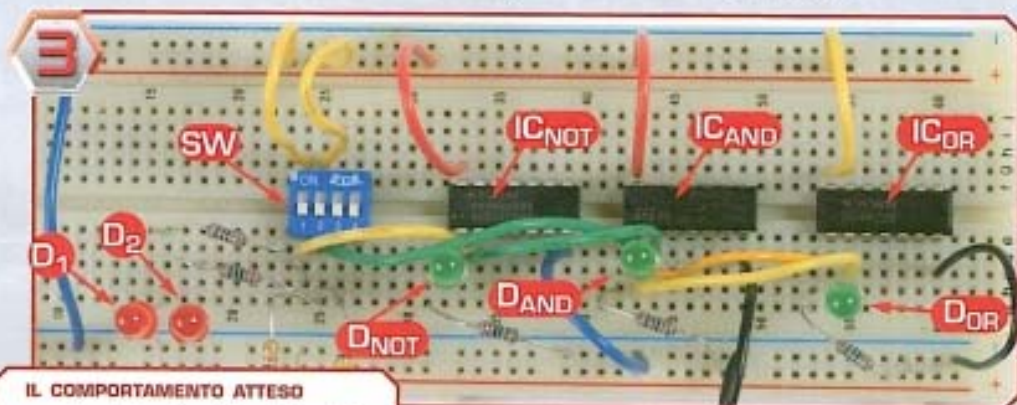




2

Il circuito che dovrai realizzare è quello mostrato nella figura accanto. Come al solito, per alimentarlo avrai bisogno del portatile da quattro stilo, che dovrà essere inserito collegando il polo positivo come terminale  $V_{CC}$  e il polo negativo come terminale di massa. Le tre porte logiche mostrate, invece,

sono messe a disposizione dai circuiti integrati che ti sei procurato. In particolare, la **NOT** è contenuta nel **74HC04**, la **AND** dal **74HC08** e la **OR** dal **74HC32**. Come puoi osservare dagli schemi interni dei circuiti che ti abbiamo mostrato nella pagina precedente, ogni integrato contiene più di una porta logica (comunque tutte dello stesso tipo). Spetta quindi a te scegliere quale utilizzare tra quelle messe a disposizione. Sempre dagli schemi puoi notare la presenza dei pin  $V_{CC}$  e  $GND$ , che sono necessari per alimentare gli integrati e che dovrai connettere alle corrispondenti linee elettriche del circuito a cui avrai collegato i terminali del portatile.



IL COMPORTAMENTO ATTESO

	D <sub>1</sub>	D <sub>2</sub>	D <sub>NOT</sub>	D <sub>AND</sub>	D <sub>OR</sub>
	●	●	●	●	●
	●	●	●	●	●
	●	●	●	●	●
	●	●	●	●	●

Ecco infine cosa dovrai attenderti dal circuito. Innanzitutto partiamo dallo switch **SW**. Esso costituisce il tuo strumento di interazione con il circuito stesso: gli **interruttori 1 e 2** che hai collegato ti permettono di impostare il livello logico degli ingressi delle porte. **Chiudendo gli interruttori** (ossia spostandoli verso la scritta 'ON') puoi portare la linee di ingresso corrispondenti a  $V_{CC}$ , ossia a un valore logico

'alto'. Al contrario, 'aprendoli' puoi imporre un livello logico basso (gli ingressi sono portati a massa dal resistore  $R_H$ ). Lo stato delle linee di ingresso è mostrato dai LED  $D_1$  e  $D_2$  (associati alle linee di input  $X_1$  e  $X_2$ ). I LED, infatti, si accenderanno in presenza di livelli logici 'alti' in ingresso, mentre si spegneranno in presenza di livelli logici 'bassi'. Lo stesso identico comportamento vale per i tre LED verdi che hai collegato ai pin di uscita delle porte logiche. Essi ti permetteranno di visualizzare lo stato assunto dalle porte in funzione degli ingressi presenti. Come puoi vedere, anche in questo caso, in uscita dai circuiti integrati, sono stati inseriti i resistori in serie ai diodi: servono, come al solito, a limitare la corrente emessa dall'integrato, in modo da non causarne la rottura. Nello schema puoi vedere riassunte le quattro combinazioni di ingresso impostabili tramite i due interruttori utilizzati con i relativi stati dei LED.