

Circuiti combinatori

- In elettronica digitale il compito da svolgere è generare uscite digitali da ingressi digitali, come ad esempio:

1. moltiplicare due numeri binari
2. convertire un numero binario in caratteri decimali
3. generare un bit di parità per controllare la trasmissione di dati
4. confrontare lo stato effettivo di un sistema con quello desiderato

tutti questi compiti sono di tipo **combinatorio**, nel senso che le uscite sono funzioni predeterminate degli ingressi, e possono essere realizzati mediante combinazioni delle porte elementari già viste; i circuiti sommatore rientrano in questa categoria.

- Invece, per la classe di problemi che richiedono di conoscere anche lo stato degli ingressi nel passato (esempio: *produrre un impulso in uscita ogni due impulsi in ingresso*), sarà necessario introdurre una qualche forma di **memoria digitale** per costruire un circuito di tipo **sequenziale**; vedremo che l'elemento base è il **flip-flop**.
- Tornando alla logica combinatoria, esistono varie tecniche per **ottimizzare** il circuito che realizza una certa funzione, cioè per trovare il circuito più semplice, o il più conveniente sotto certe condizioni. Per problemi con ≤ 4 ingressi (il numero di uscite è irrilevante) la tecnica più utile è quella delle **mappe di Karnaugh**: vediamola con un esempio.
- *Problema*: Progettare una macchina per contare i voti (simultanei) di tre elettori in un referendum. La decisione deve essere presa a maggioranza. Vediamo la soluzione in 4 fasi.
 1. costruire la tabella di verità:

A	B	C	Q
0	0	0	0
0	0	1	0
0	1	0	0
1	0	0	0
1	1	0	1
1	0	1	1
0	1	1	1
1	1	1	1

2. costruire la mappa di Karnaugh - si tratta di una tabella a due dimensioni, in cui gli indici di riga e colonna corrispondono ai possibili valori degli ingressi, e i contenuti delle celle al valore dell'uscita:

AB→	00	01	11	10
C ↓				
0	0	0	1	0
1	0	1	1	1

nel nostro caso i primi due bit di ingresso (A e B) sono stati raggruppati, ed è stata seguita la regola generale per cui *soltanto uno* dei bit di ingresso *può cambiare* quando ci si sposta da una cella ad una delle celle adiacenti (sia in orizzontale che in verticale)

3. individuare blocchi di 1 (o in alternativa di 0), cioè rettangoli composti da celle con contenuto 1 nella mappa, e le corrispondenti espressioni logiche:

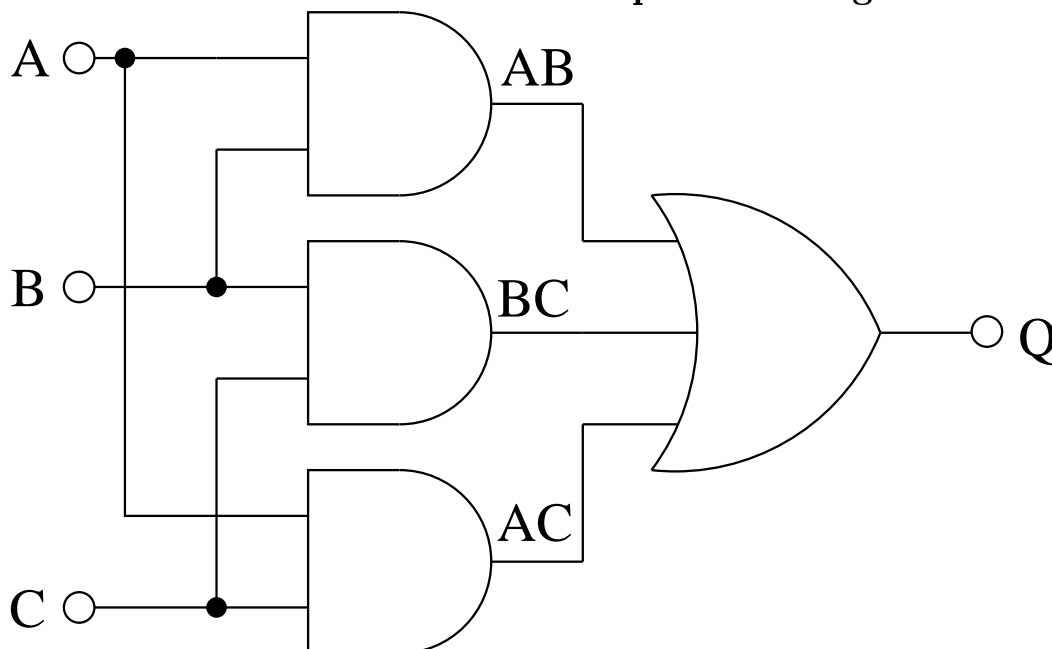
$$AB = \text{blocco verticale di 1 in terza colonna} \quad (27)$$

$$AC = \text{blocco orizzontale di 1 in seconda riga} \quad (28)$$

$$BC = \text{blocco orizzontale di 1 in seconda riga} \quad (29)$$

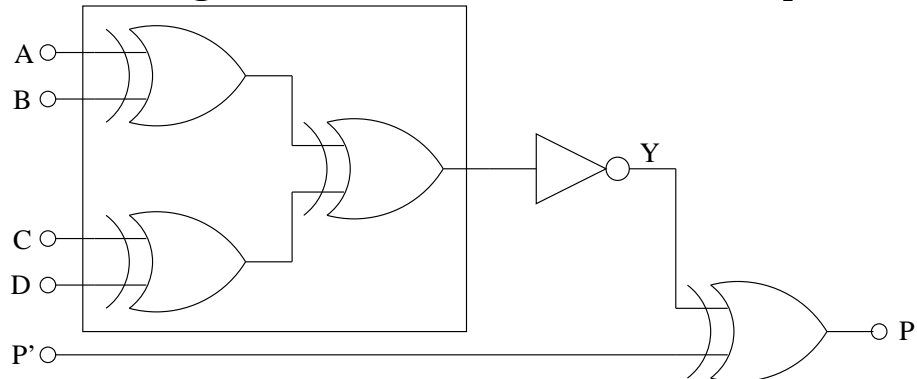
$$Q = AB + AC + BC \quad (30)$$

4. realizzare il circuito in base all'espressione logica trovata:



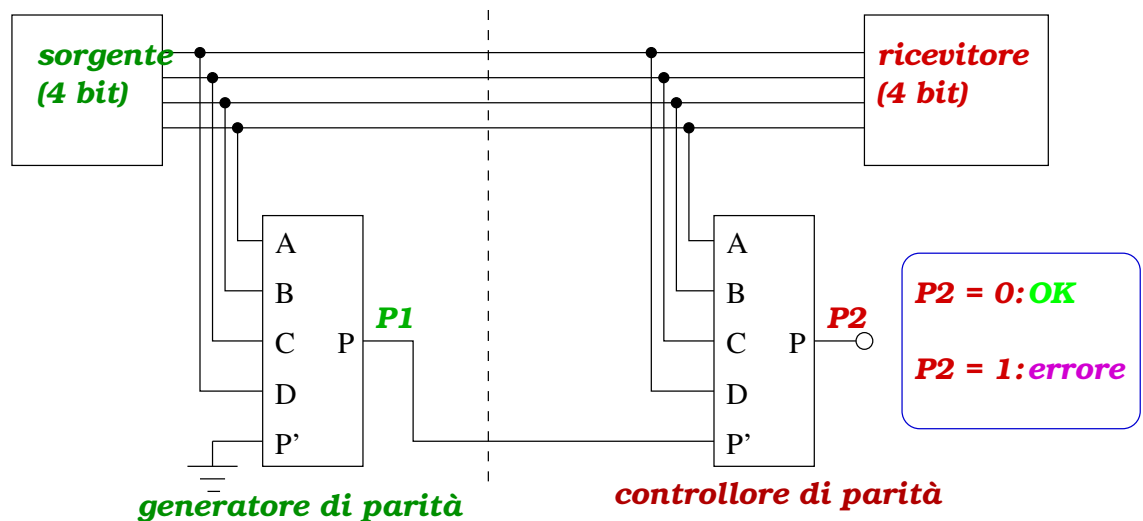
Controllore/generatore di parità

- La **parità** di una parola a N bit è definita come **pari** se la parola contiene un numero pari di 1, risp. **dispari** se ne contiene un numero dispari. Un circuito basato su una catena di porte XOR svolge la funzione di **controllore** di parità:



se $P' = 0$ (una delle due possibili convenzioni), allora $P = 1$ per parità **pari** della parola ($ABCD$), e $P = 0$ per parità **dispari**.

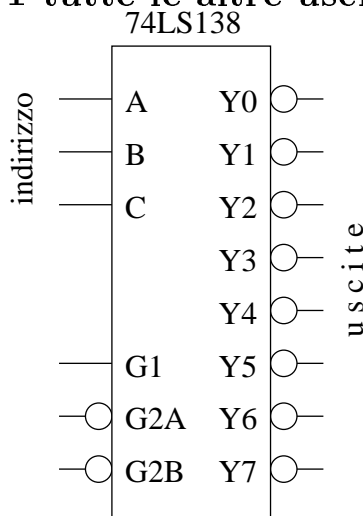
- Per evidenziare la presenza di errori di trasmissione tra una sorgente ed un ricevitore possiamo trasmettere un bit di parità insieme ai bit del dato (sullo stesso canale), e verificare alla ricezione che la parità dei bit ricevuti sia in accordo con il bit di parità; si può utilizzare lo stesso circuito visto sopra anche come **generatore** di parità:



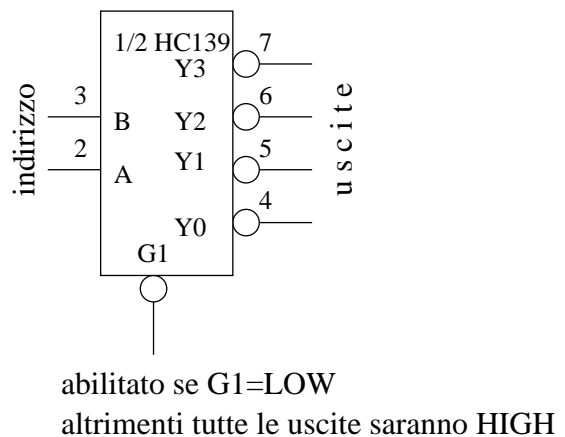
- Sono disponibili circuiti MSI per generare/controllare la parità, ad es. il 74HC280 accetta 9 bit di ingresso e genera in uscita sia il bit di parità pari sia quello dispari.

Decodificatore - demultiplexer

- Succede di frequente di dover attivare una (e una sola) tra N possibili operazioni sulla base di un'istruzione codificata:
 1. durante l'esecuzione di un programma in un microprocessore,
 2. per eseguire una sequenza predefinita di operazioni (distributore di bevande, lavatrice, ...),
 3. per pilotare un display a 7 segmenti con un numero BCD;
- il **decodificatore** (*decoder*) risolve questo problema, ad es. il '138 è un decodificatore *da 1 a 8* che pone = 0 l'uscita indicata dall'indirizzo codificato nei tre bit di ingresso, mantenendo a 1 tutte le altre uscite.



abilitato se $G1=HIGH$ e $G2A=G2B=LOW$
altrimenti tutte le uscite saranno HIGH



- Il '138 comprende anche 3 ingressi di abilitazione con diverse polarità (al suo interno vengono combinati con un AND). Il '139 contiene invece due decodificatori *da 1 a 4* con un ingresso di abilitazione per ciascuno. È possibile collegare in parallelo più decodificatori per aumentare il numero di linee di uscita.
- Un caso particolare è rappresentato dai decodificatori che accettano un ingresso BCD (4 bit) e attivano la corrispondente linea di uscita corrispondente alla cifra decimale prescelta (da 0 a 9); ad es. il 74HC42 è un decodificatore *da 4 a 10* che pone nello stato LOW l'uscita selezionata.
- Il **demultiplexer** è un circuito che accetta un ingresso di *segnale* oltre ai bit di *indirizzo*, e trasmette questo dato sulla linea di uscita selezionata; si può facilmente ottenere dal decodificatore inviando il segnale a uno degli ingressi di abilitazione.