



individuare la posizione dell'errore attraverso l'intersezione riga-colonna. In tabella 2 si mostra la determinazione della parità orizzontale e verticale.

**Tabella 2**

Messaggio		Parità orizzontale
1	0001 0001	0
2	0001 1100	1
3	0010 1010	1
4	0011 1011	1
5	0000 1111	0
6	0101 0101	0
7	0111 0001	0
8	0000 1010	0
Parità verticale		0011 1101

### Codice di Hamming

A differenza del codice di parità, aumentando i bit ridondanti, è possibile conoscere la posizione e correggere l'errore.

Si definisce **distanza di Hamming** (D) il numero di 0 e 1 che si devono cambiare per trasformare una combinazione in un'altra. Attraverso un codice con distanza di Hamming D si possono rilevare D-1 errori e correggere  $(D - 1)/2$  errori. Per correggere 1 bit in una parola binaria ad n bit si deve usare un codice di Hamming con  $D = 3$  che deve contenere m bit di controllo aggiuntivi inseriti opportunamente nel messaggio binario. Indicando con 1,2,3,4,5,6... le posizioni dei bit del messaggio, i bit di controllo vengono introdotti solo nelle posizioni di potenza 2. Affinché la correzione possa avvenire, il numero di bit del messaggio ed il numero di bit di controllo, devono soddisfare la seguente disuguaglianza:  $2^n - 1 \geq n + m$ .

In tabella 3, ad esempio, si mostra il posizionamento dei bit di controllo  $P_1, P_2, P_4$  nel caso in cui il messaggio sia costituito dai 4 bit  $B_0, B_1, B_2, B_3$ .

**Tabella 3**

Posizione dei bit	1	2	3	4	5	6	7
Parola da trasmettere	$P_1$	$P_2$	$B_0$	$P_4$	$B_1$	$B_2$	$B_3$

Inviato il messaggio, il ricevitore decodifica la parola, generando un numero di bit che dipende dai bit di controllo inviati. Questi nuovi bit generati hanno la funzione di rilevare l'eventuale errore introdotto nel messaggio; se il messaggio risulta corretto i bit generati, in questo caso  $K_4, K_2$  e  $K_1$ , assumeranno i valori 000. In tabella 4 si mostra come vengono individuati gli errori.

**Tabella 4**

$K_4$	$K_2$	$K_1$	Significato
0	0	0	Messaggio corretto
0	0	1	$P_1$ errato
0	1	0	$P_2$ errato
0	1	1	$B_0$ errato
1	0	0	$P_4$ errato
1	0	1	$B_1$ errato
1	1	0	$B_2$ errato
1	1	1	$B_3$ errato

Dalla tabella 4 si evince che:

- 1) il bit  $K_1 = 1$  individua l'errore nelle posizioni 1,3,5,7;
- 2) il bit  $K_2 = 1$  individua l'errore nelle posizioni 2,3,6,7;
- 3) il bit  $K_4 = 1$  individua l'errore nelle posizioni 4,5,6,7.

In una trasmissione priva di errori, utilizzando l'operatore OR-esclusivo (XOR), si ha che:

- 1)  $K_1 = P_1 \oplus B_0 \oplus B_1 \oplus B_3 = 0$ ;
- 2)  $K_2 = P_2 \oplus B_0 \oplus B_2 \oplus B_3 = 0$ ;
- 3)  $K_4 = P_4 \oplus B_1 \oplus B_2 \oplus B_3 = 0$ .

Da queste uguaglianze si ricavano  $P_1$ ,  $P_2$  e  $P_4$ :

- 1)  $P_1 = B_0 \oplus B_1 \oplus B_3$ ;
- 2)  $P_2 = B_0 \oplus B_2 \oplus B_3$ ;
- 3)  $P_4 = B_1 \oplus B_2 \oplus B_3$ .

Utilizzando le precedenti relazioni è possibile costruire, nel trasmettitore i bit di controllo  $P_1$ ,  $P_2$ ,  $P_4$  e al ricevitore, i bit  $K_4$ ,  $K_2$  e  $K_1$  che individuano la posizione dell'eventuale errore.

## ESECUZIONE PRATICA

### Codice di parità

Questa esperienza è iniziata col disegno e l'esecuzione del circuito in figura 1 in ambiente EWB, che raffigura un circuito per la rilevazione di errori utilizzando il codice di parità di un messaggio a 4 bit ABCD. Come si può notare dallo schema in figura 1, il blocco DATO TRASMESSO, genera i bit della parola da trasmettere; il blocco SIMULATORE DI ERRORE immette nel messaggio un errore provocato dalla apertura degli switch FGHI; il blocco GENERATORE DI PARITÀ ha il compito di generare un bit P che può assumere il valore 1 se il numero di 1 nel messaggio è pari o valore 0 se il numero di 1 nel messaggio è dispari. Il blocco CONTROLLORE DI PARITÀ fornisce in uscita (OUT) il valore 0 se non vi sono errori ed il valore 1 se vi sono errori.

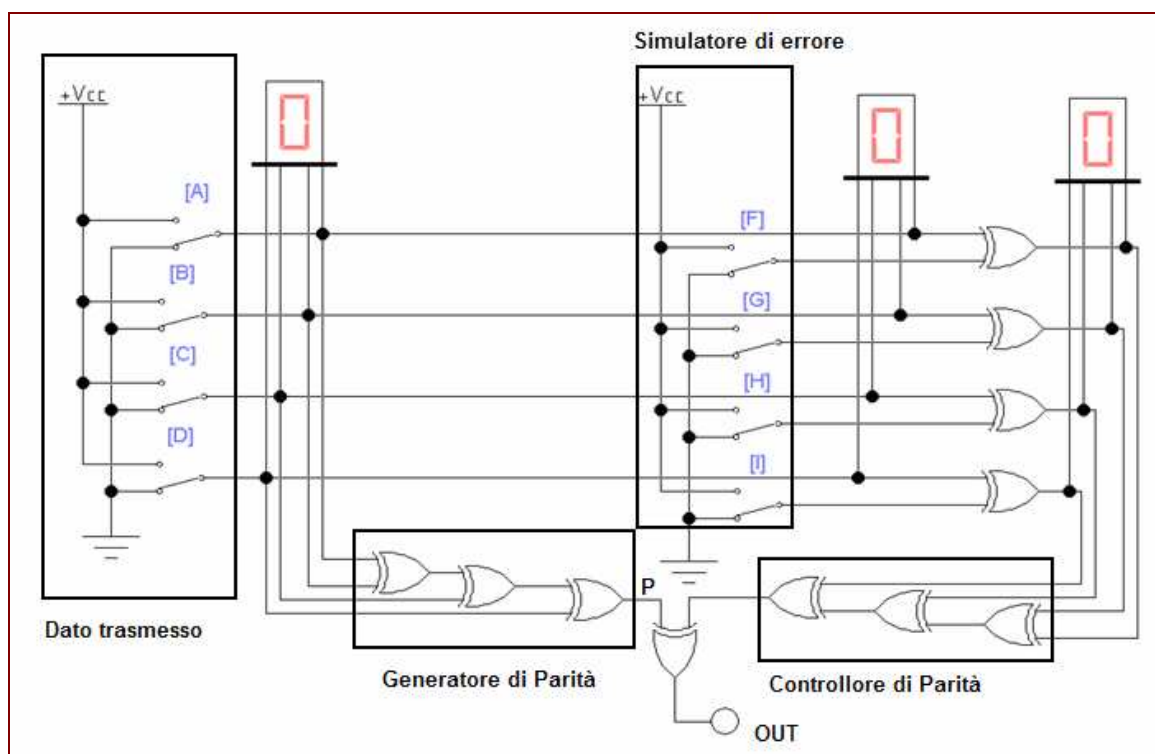


Figura 1

Ricordiamo che la porta XOR può essere come invertitore di 1 bit utilizzando il secondo ingresso come controllo, come mostrato in figura 2.

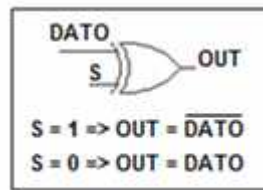


Figura 2

Dopo la simulazione in ambiente Electronics WorkBench, si è proceduto con il montaggio del circuito in figura 3 su basetta di sperimentazione K-H IDL – 800 (figura 4) alimentata con tensione pari a 220 ed accessoriata con display, switch e LED.

Di questi sono stati utilizzati due LED e 4 switch per la simulazione dell'errore.

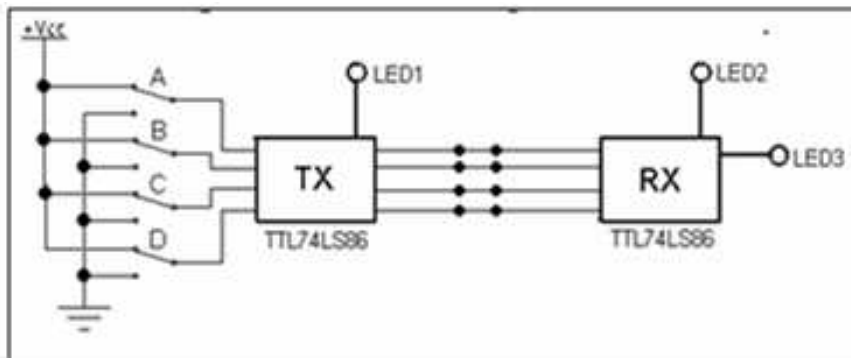


Figura. 3



Figura. 4

Per il montaggio sono stati utilizzati anche due integrati TTL 74 LS 7486 che al loro interno contengono quattro porte XOR come mostrato in figura 5.

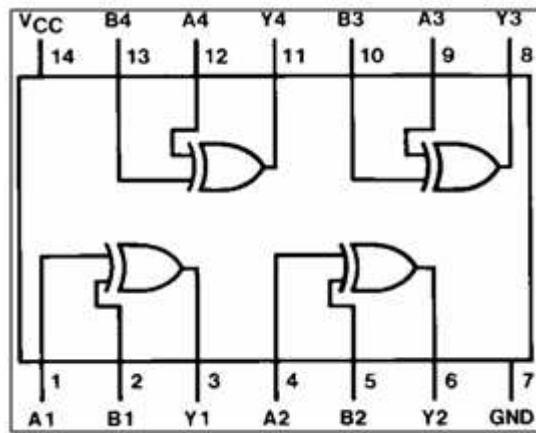


Figura 5

Secondo lo schema di figura 3 il LED1 mostra lo stato dell'uscita del generatore di parità, il LED2 raffigura lo stato dell'uscita del controllore di parità ed il LED3 mostra l'operazione di XOR tra lo stato logico del LED1 e del LED2.

In figura 6 si mostra l'intero cablaggio.

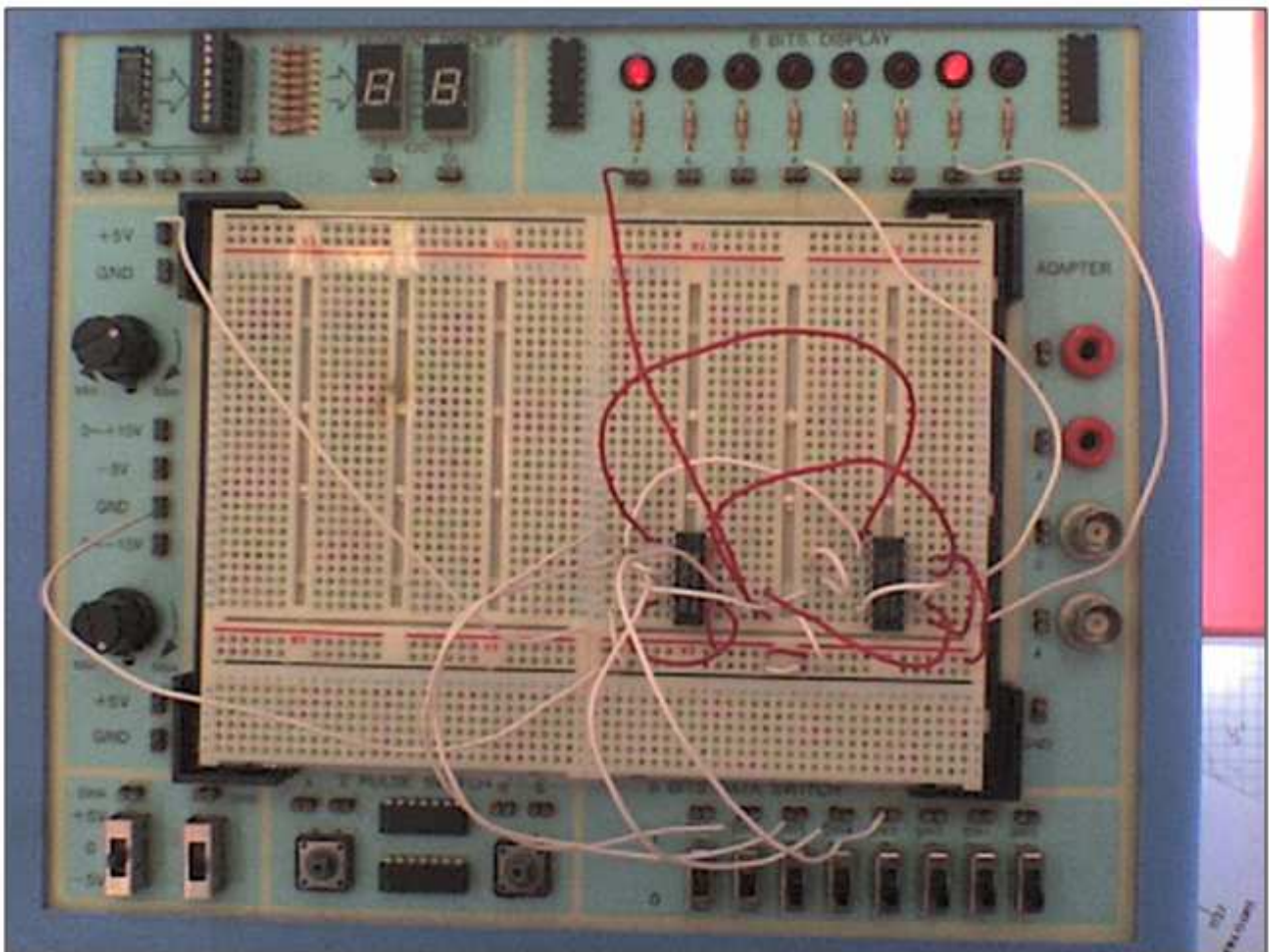


Figura 6

Successivamente si è simulato l'invio di un messaggio ed un errore, staccando un cavetto in corrispondenza di un pin d'ingresso dell'integrato che rappresenta il controllore di parità.

Si sono analizzati i seguenti casi:

- 1) Il LED1 e il LED2 risultano accesi quando nella parola d'ingresso si ha un numero di bit 1 dispari;
- 2) il LED 1 e il LED 2 risultano spenti quando nella parola d'ingresso si ha un numero di bit 1 pari;
- 3) nel momento in cui si simula l'errore, viene provocata l'accensione del LED 3, mentre lo stato del LED 1 e del LED 2 non coincide.
- 4) La simulazione di due errori contemporanei non viene rilevata.

Il codice di parità, nonostante possa solo rilevare l'errore su di un bit del messaggio senza correggerlo, è molto adoperato nei sistemi di trasmissione, poiché la probabilità che si possano avere 2 bit errati contemporaneo è molto bassa.

### Codice di Hamming

Come seconda prova, è stato realizzato e testato il circuito di figura 6 in ambiente EWB, che rappresenta un circuito per la rilevazione e correzione di un errore tramite il codice di Hamming per un messaggio a 4 bit ABCD.

Realizzato il circuito, si simula l'errore su di un bit del messaggio modificando lo stato di uno degli switch EFGH. Il circuito riconosce l'errore e tramite il generatore di indirizzi, genera tre bit chiamati  $K_4$ ,  $K_2$  e  $K_1$  che decodificati costituiscono l'indirizzo dell'errore, corretto attraverso l'invertitore. In questo modo il dato ricevuto corrisponde con il dato trasmesso.

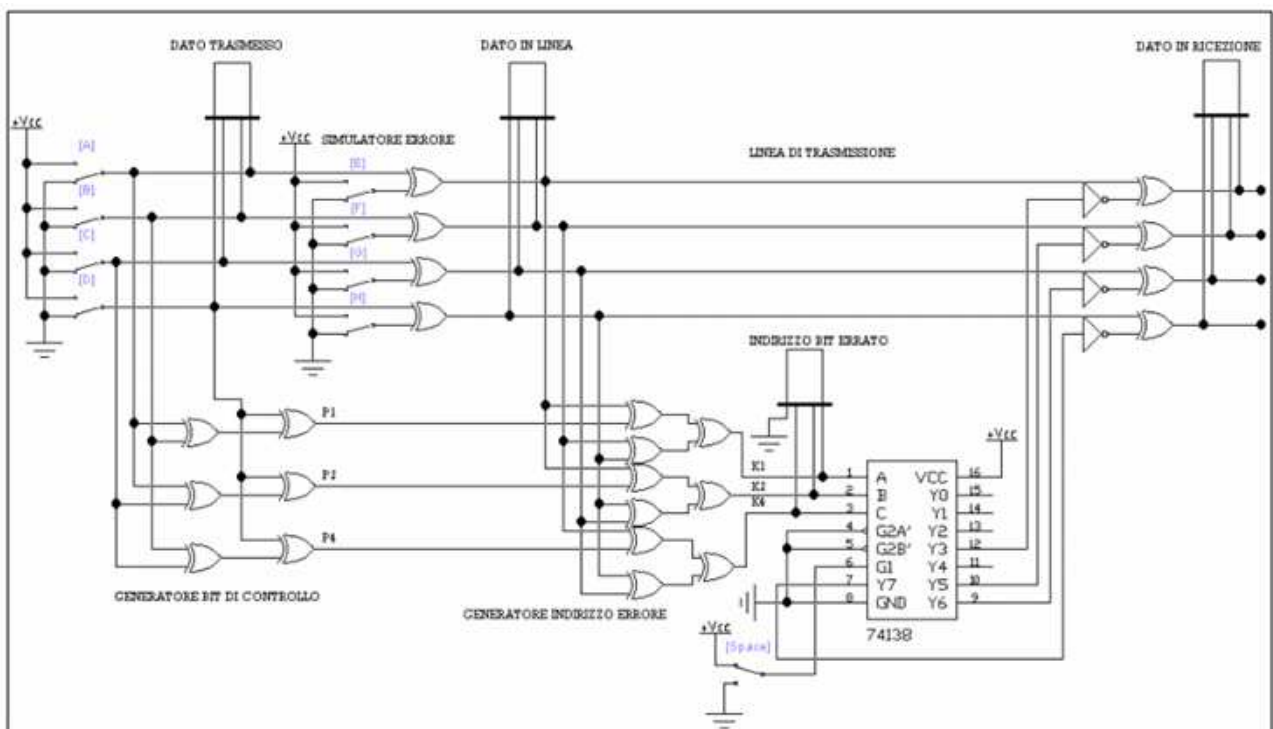


Figura 6