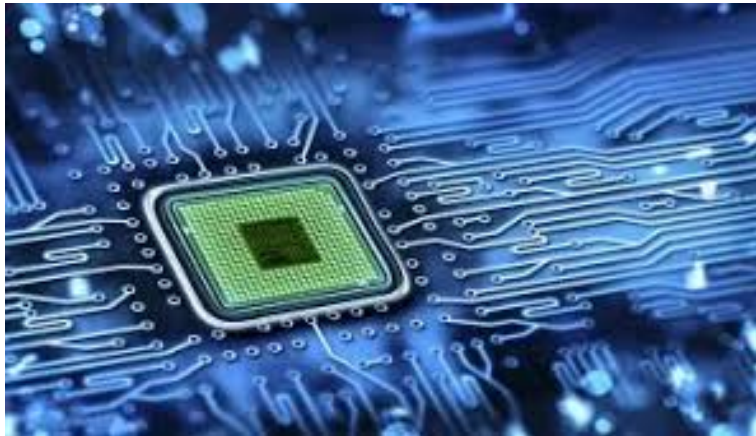


# Electrónica Digital

Ingeniería Informática – FICH, UNL  
Leonardo Giovanini



## Circuitos combinatoriales para gestión de datos

*Multiplexores y códigos*

En esta se estudiarán los siguientes temas:

- Multiplexor;
- Demultiplexor
- Códigos detectores/correctores de errores;
- Códigos de línea.

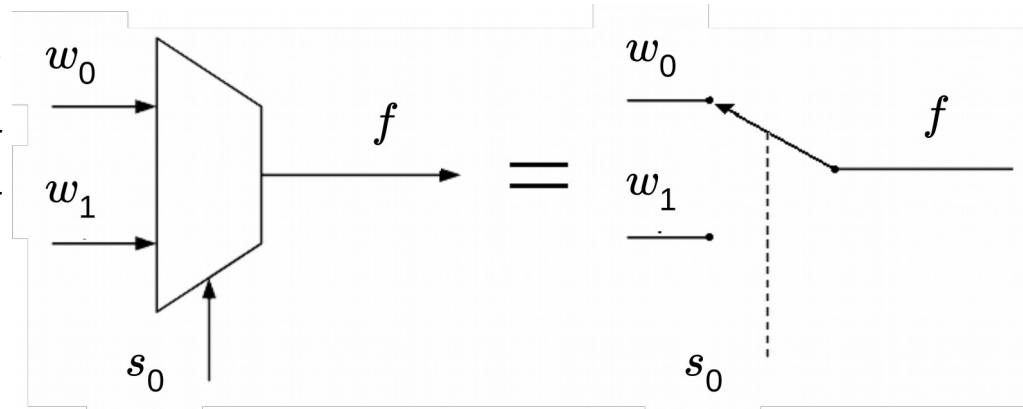
En los sistemas digitales usualmente es necesario **seleccionar información entre un número dado de fuentes**.

Los **multiplexores**, también conocidos como **selector de datos**, son circuitos combinacionales con varias líneas de entrada ( $w_i$   $i=0, \dots, 2^n-1$ ) y una única de salida ( $f$ ) de datos. Están dotados de  $n$  entradas de control ( $s_i$   $i=0, \dots, n-1$ ) que permiten seleccionar la entrada de datos  $w_i$  que se conectará con la salida  $f$ .

Cada combinación de las entradas de selección  $s_i$  corresponde a una entrada de datos, y la salida del multiplexor será al valor de la entrada seleccionada.

Un multiplexor se puede utilizar para:

- **Seleccionar de entradas;**
- **Serializar datos** – convertir datos de formato paralelo al formato serie;
- **Multiplexar de datos** – transmitir datos diferentes orígenes utilizando una sola conexión;
- **Realización de funciones lógicas** – implementar funciones lógicas de manera más compactas y sencilla que con compuertas tradicionales.



Supongamos que tenemos dos fuentes de datos, provenientes de las entradas  $w_0$  y  $w_1$ . El valor de estas entradas cambia con el tiempo de modo que tenemos una secuencia de unos y ceros aplicadas a las entradas  $w_0$  y  $w_1$ .

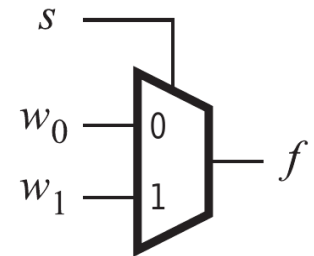
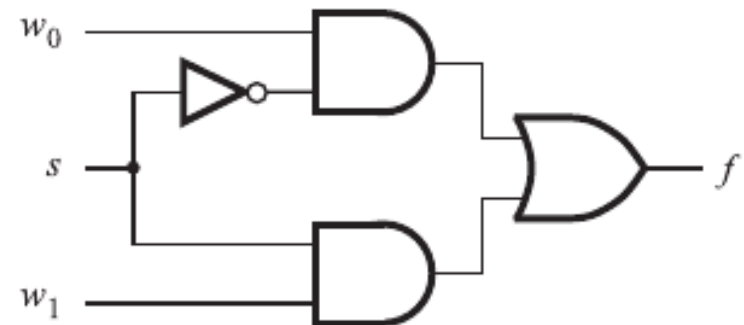
Un **multiplexor de dos entradas** es un circuito combinacional cuya salida  $f$  tiene el mismo valor que  $w_0$  y  $w_1$ , dependiendo del valor de la entrada de selección  $s_0$ . De modo que su tabla de verdad está dada por

$s$	$f$
0	$w_0$
1	$w_1$

La función lógica que describe el comportamiento del multiplexor es

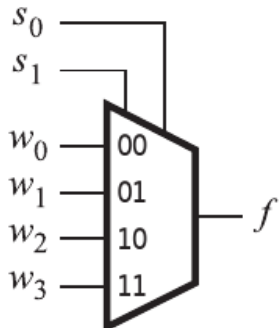
$$f = \bar{s} w_0 + s w_1,$$

y su implementación circuital está dada por



De manera similar podemos diseñar un multiplexor de cuatro entradas de datos ( $w_0$  a  $w_3$ ) y dos de selección ( $s_0$  y  $s_1$ ). La tabla de verdad está dada por

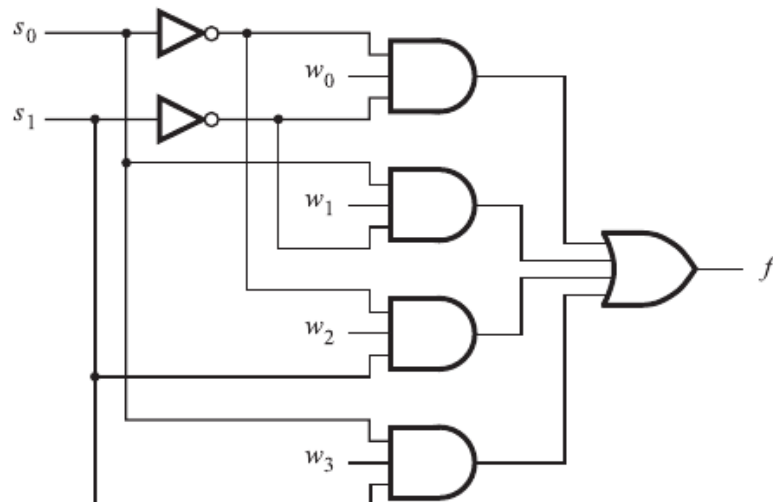
$s_1$	$s_0$	$f$
0	0	$w_0$
0	1	$w_1$
1	0	$w_2$
1	1	$w_3$



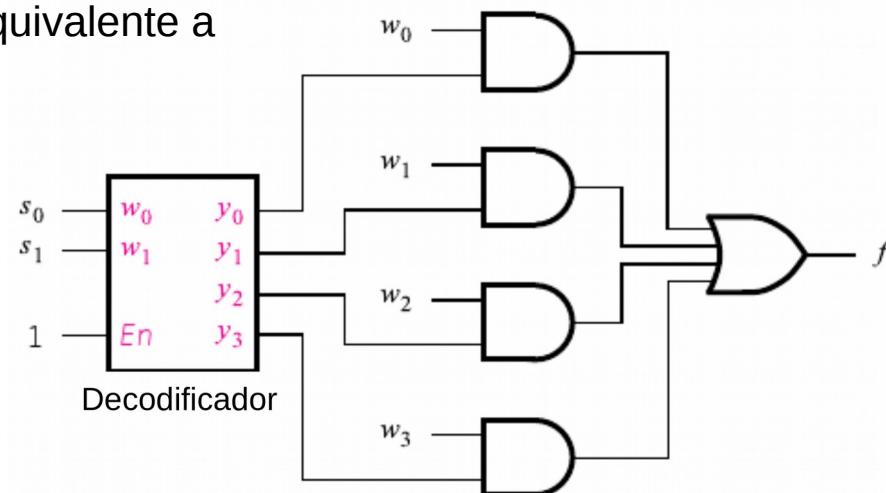
La función lógica que describe el comportamiento del multiplexor es

$$f = \bar{s}_0 \bar{s}_1 w_0 + s_0 \bar{s}_1 w_1 + \bar{s}_0 s_1 w_2 + s_0 s_1 w_3,$$

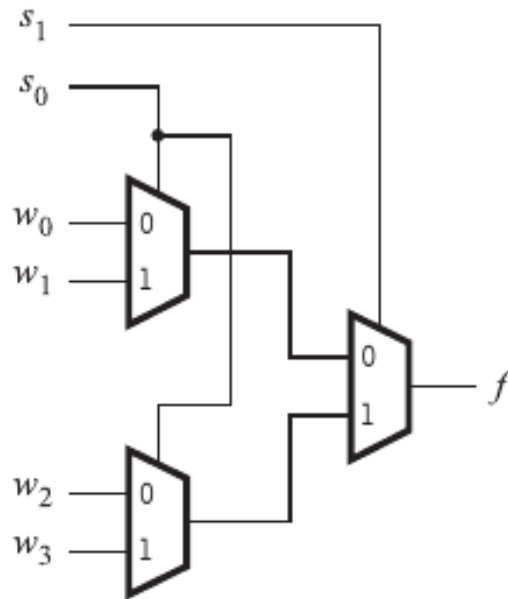
y su implementación circuital está dada por



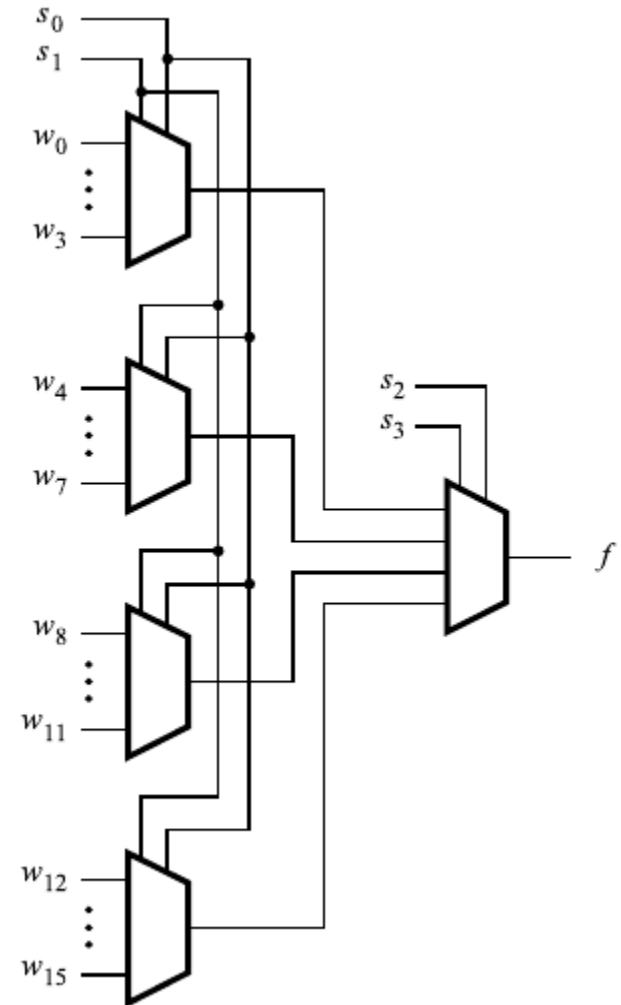
y es equivalente a



De manera similar a los decodificadores, que se pueden utilizar para implementar un multiplexor, multiplexores de muchas entradas se pueden construir a partir de multiplexores de una menor cantidad de entradas.



Mostrando una estructura en árbol similar a la de los decodificadores.



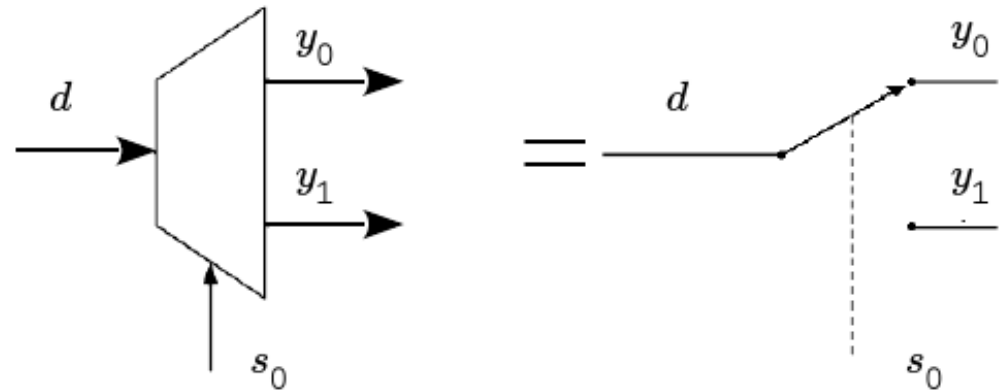
Un demultiplexor es un circuito combinacional Toma datos de una línea y los distribuye a un determinado número de líneas de salida. Por este motivo, el demultiplexor se conoce también como distribuidor de datos.

Los **demultiplexores** son circuitos combinacionales con un línea de entrada ( $d$ ) y varias salidas ( $w_i$   $i=0, \dots, 2^n-1$ ) de datos. Están dotados de  $n$  entradas de control ( $s_i$   $i=0, \dots, n-1$ ) que permiten direccionar el dato disponible a la entrada a cualquiera de las salidas  $w_i$ .

Cada combinación de las entradas de selección  $s_i$  corresponde a una salida de datos seleccionada, la cual tendrá el valor en la entrada.

Un demultiplexor se puede utilizar para:

- **Distribuir datos;**
- **Paralelizar datos** - convertir datos de formato serie al formato paralelo;
- **Demultiplexar de datos** – reorganizar datos multiplexados, asignados el canal original;
- **Realización de decodificadores** – implementar decodificadores de manera más compactas y sencilla que con compuertas tradicionales.



Un demultiplexor de cuatro salidas de datos ( $w_0$  a  $w_3$ ) y dos de selección ( $s_0$  y  $s_1$ ). Las funciones lógicas que describen el comportamiento del demultiplexor son

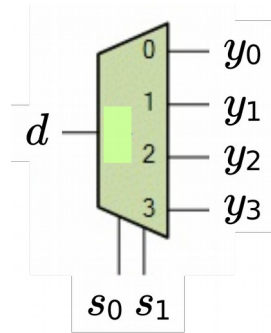
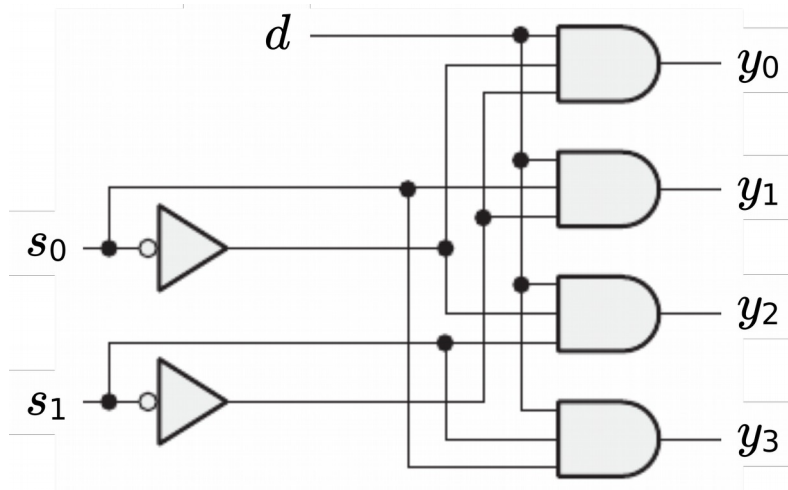
$$y_0 = \bar{s}_0 \bar{s}_1 d;$$

$$y_1 = s_0 \bar{s}_1 d;$$

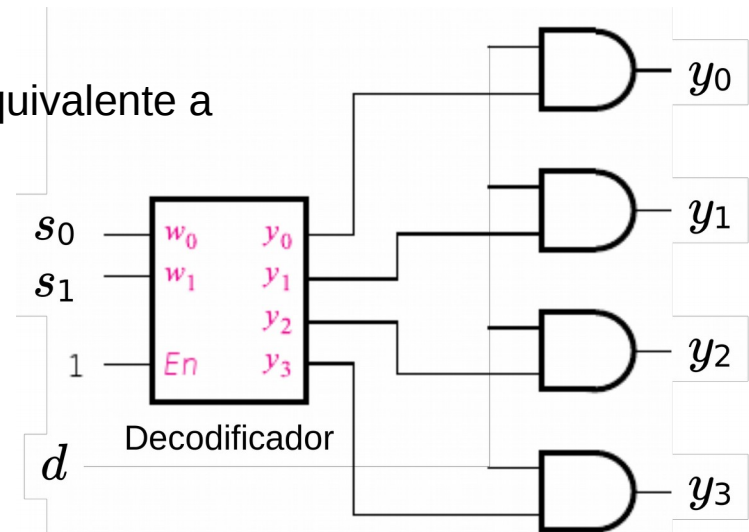
$$y_2 = \bar{s}_0 s_1 d;$$

$$y_3 = s_0 s_1 d;$$

y su implementación circuital está dada por



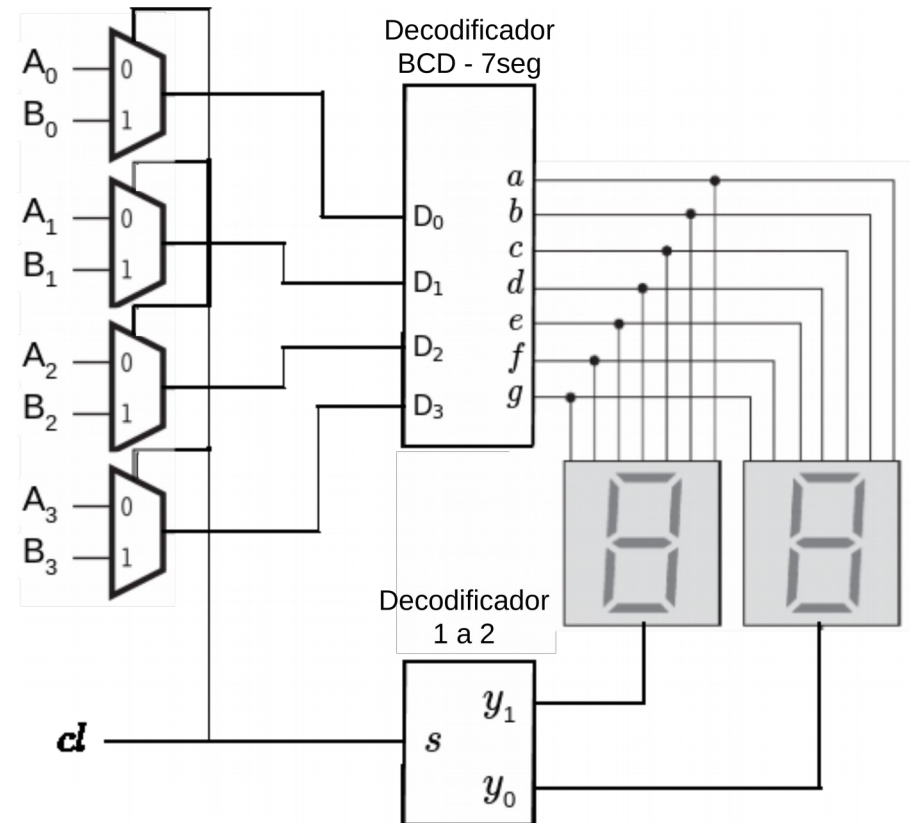
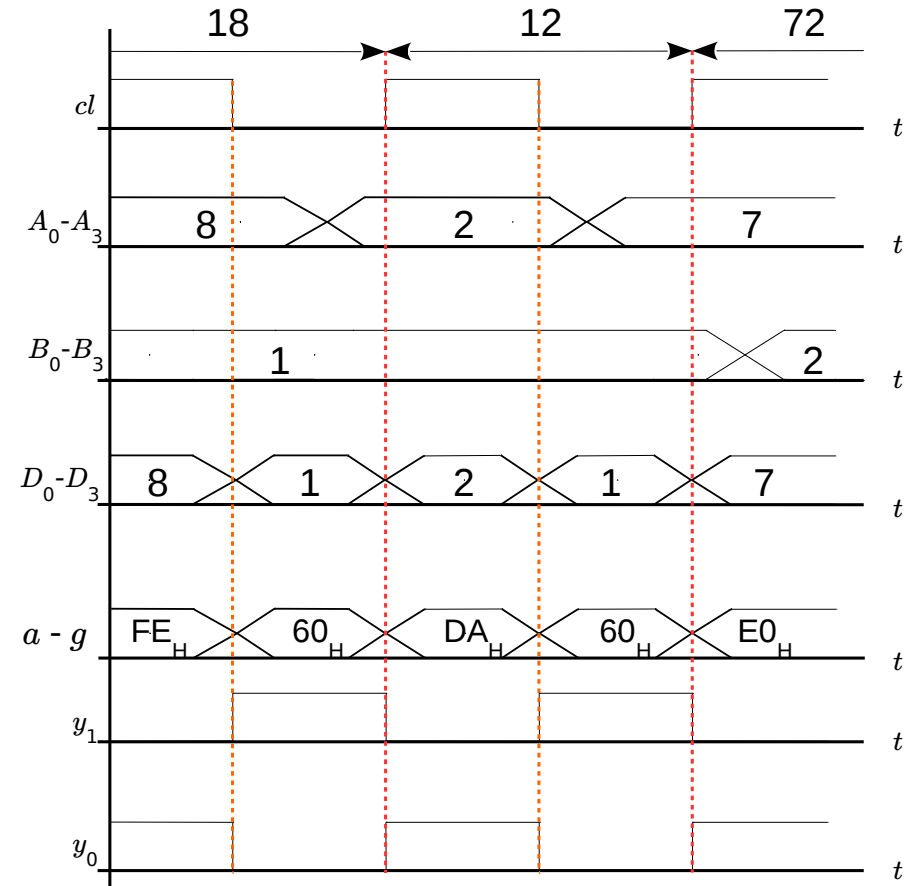
y es equivalente a





# Circuitos Combinacionales – Multiplexores

## Multiplexión de displays



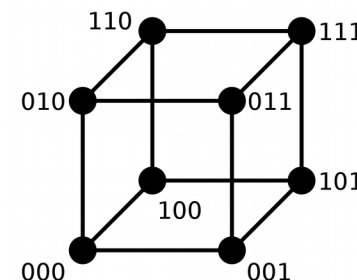
Una de las aplicaciones más importante de la codificación es cambiar la representación de la información para **introducir redundancia** y poder **detectar y corregir** los cambios que se produzcan en la información cuando se la transmite o almacena, entre otras.

Un código de detección de errores **muy simple y muy utilizado** en la práctica es **la paridad**.

Se lo utiliza en las memorias de las computadoras y en las transmisiones serie a baja velocidad.

A	B	$f_{par}$	$f_{impar}$
0	0	0	1
0	1	1	0
1	0	1	0
1	1	0	1

La paridad se calcula a partir de agregar un bit de información de modo que la cantidad de 1 en la palabra sea par (ó impar). La tabla de verdad muestra como se calcula la paridad para una palabra binaria de dos bits.



La función lógica resultante es de mucha utilidad en el diseño de circuitos combinacionales, por lo que se le asigna una compuerta lógica específica: la compuerta OR Exclusiva (XOR).



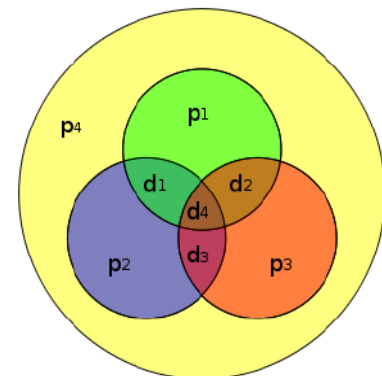
$$f_{par} = \bar{A}B + A\bar{B}$$

	B	
A	0	1
0	0	1
1	1	0

Un código **detector/corrector** de errores muy utilizado en los sistemas digitales es el **código de Hamming**.

Este código añade al mensaje bits detectores-correctores de error ordenados de modo los bits erróneos pueden ser identificados.

Para mejorar su desempeño, se le agrega un bit de paridad al código calculado. Este código se conoce como **código de Hamming extendido**.



La regla de asignación de bits y cálculo de las paridad del código de Hamming se muestra en la siguiente tabla

Posición	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Tipo	p1	p2	d1	p4	d2	d3	d4	p8	d5	d6	d7	d8	d9	d10	d11	p16	d12	d13	d14	d15
Cálculo de paridad	p1	X		X		X		X		X		X		X		X		X		X
	p2		X	X			X	X			X	X			X	X			X	X
	p4				X	X	X	X					X	X	X	X				X
	p8								X	X	X	X	X	X	X					
	p16																X	X	X	X

Entonces, el circuito combinacional para implementar un **codificar de Hamming** consistirá en circuitos **calculen las paridades** correspondientes (compuerta XOR) y organizar los datos y las paridades en los lugares correspondientes.

Por ejemplo, si queremos detectar 2 errores y corregir 1 en un sistema de memoria cuyo ancho de palabra es 8 bits, deberemos agregar 4 bits de paridad y uno de paridad global. Los circuitos combinacionales que determinarán la paridad de los grupos definidos en la tabla serán calculados con compuertas XOR con cantidad de entradas pertinentes

$$p_1 = d_1 \oplus d_2 \oplus d_4 \oplus d_5 \oplus d_7;$$

$$p_2 = d_1 \oplus d_3 \oplus d_4 \oplus d_6 \oplus d_7;$$

$$p_4 = d_2 \oplus d_3 \oplus d_4 \oplus d_8;$$

$$p_8 = d_5 \oplus d_6 \oplus d_7 \oplus d_8;$$

$$p_g = p_1 \oplus d_1 \oplus p_2 \oplus p_4 \oplus d_2 \oplus d_3 \oplus d_4 \oplus p_8 \oplus d_5 \oplus d_6 \oplus d_7 \oplus d_8;$$

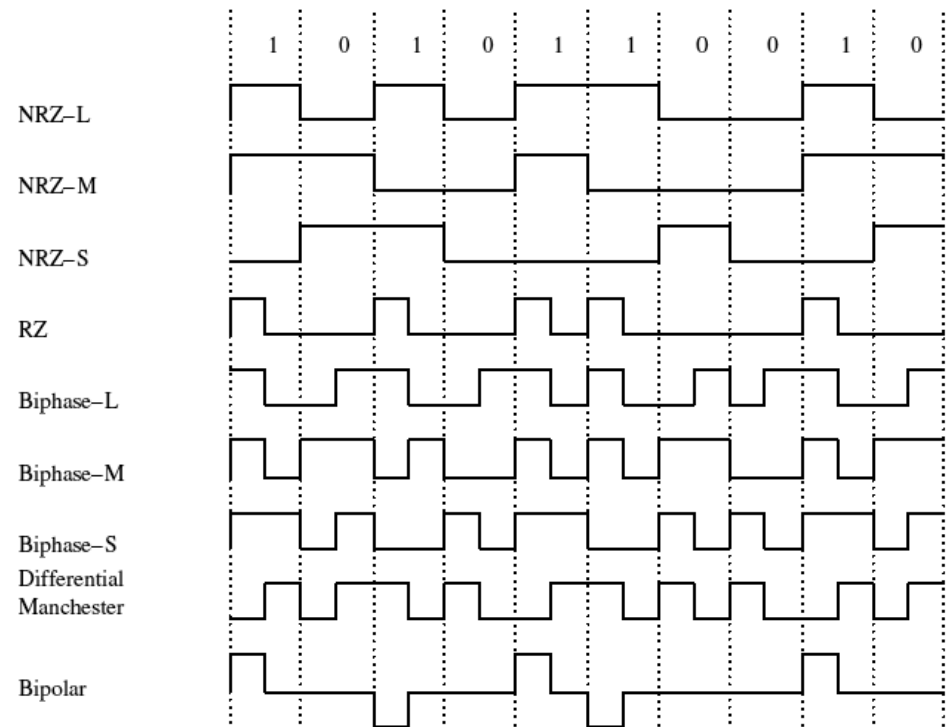
$$\text{Dato} = [ p_1, d_1, p_2, p_4, d_2, d_3, d_4, p_8, d_5, d_6, d_7, d_8, p_g ];$$

En telecomunicaciones, un código de línea es un código utilizado en un sistema de digital para facilitar Son usados para el transporte y almacenamiento de datos digital.

Estos códigos consisten en representar la señal digital transportada en alguna magnitud de la señal que la transporta (amplitud, frecuencia, fase ó combinaciones).

En la transmisión y almacenamiento de datos es conveniente que la secuencia de los símbolos transmitidos no presente una componente continua, ya que introduce errores al detectar el código.

Por ello se codifica la señal de modo que elimine la componente continua, sincronice el transmisor con el receptor y permita detectar errores.



Un ejemplo de este tipo de códigos es el código 4B5B, el cual mapea grupos de 4 bits de datos en palabras código de 5 bits para su transmisión o almacenamiento.

Estas palabras código se obtienen de una tabla predeterminada y se eligen para obtener las propiedades deseadas (sincronismo, componente de DC mínima, etc.).

Un efecto colateral de la codificación es que utilizamos un 25% más de datos para enviar la misma información.

Además, la información no utilizada explícitamente por el código, puede emplearse para detectar errores en el paquete de datos.

Un codificador binario a 4B5B es circuito combinatorial compuesto por cinco subcircuitos que calculan cada uno de los cinco bits de la palabra código a partir de los 4bits de entrada según lo muestra la tabla de códigos, que en este caso también es la tabla de verdad del codificador.

Dato		Código 4B5B
Hex	Binario	
0	0000	11110
1	0001	01001
2	0010	10100
3	0011	10101
4	0100	01010
5	0101	01011
6	0110	01110
7	0111	01111
8	1000	10010
9	1001	10011
A	1010	10110
B	1011	10111
C	1100	11010
D	1101	11011
E	1110	11100
F	1111	11101

El codificador está dado por los siguientes circuitos combinacionales

$$f_4 = A + C\bar{B} + \bar{B}\bar{D}$$

	C D			
AB	0 0	0 1	1 1	1 0
0 0	1	0	1	1
0 1	0	0	0	0
1 1	1	1	1	1
1 0	1	1	1	1

$$f_1 = \bar{A}B + A\bar{B} + A\bar{C} + \bar{A}\bar{C}\bar{D}$$

	C D			
AB	0 0	0 1	1 1	1 0
0 0	1	0	0	0
0 1	1	1	1	1
1 1	1	1	0	0
1 0	1	1	1	1

$$f_2 = CD + \bar{A}\bar{C} + \bar{A}\bar{B}\bar{D}$$

	C D			
AB	0 0	0 1	1 1	1 0
0 0	1	0	1	1
0 1	0	0	1	1
1 1	0	0	1	1
1 0	0	0	1	1

$$f_3 = \bar{A}B + \bar{A}\bar{C} + A\bar{B}$$

	C D			
AB	0 0	0 1	1 1	1 0
0 0	1	1	0	0
0 1	1	1	1	1
1 1	0	0	0	0
1 0	1	1	1	1

$$f_0 = D$$

	C D			
AB	0 0	0 1	1 1	1 0
0 0	0	1	1	0
0 1	0	1	1	0
1 1	0	1	1	0
1 0	0	1	1	0