

## Análisis de circuitos combinacionales MSI

03

### En esta unidad aprenderás a:

- Identificar y caracterizar las funciones digitales más relevantes de carácter combinacional.
- Analizar funciones y circuitos combinacionales, interpretando los esquemas y describiendo su funcionamiento.
- Definir circuitos combinacionales, seleccionando los componentes precisos y aplicando los procedimientos de diseños necesarios.
- Montar, ensayar y poner a punto los circuitos formados por dispositivos combinacionales.

### 3. Análisis de circuitos combinacionales MSI

#### 3.1 Propiedades de los circuitos combinacionales



## 3.1 Propiedades de los circuitos combinacionales

En esta Unidad estudiaremos, al principio de cada bloque, los circuitos combinacionales integrados en una única pastilla MSI (sigla inglesa de «media escala de integración», véase Apartado 1.7.B.) que son de aplicación general. Sin embargo, cualquier diagrama lógico construido a partir de una o varias funciones, como las expuestas en las Unidades anteriores, es también un circuito de idénticas características. Un **circuito combinacional** es aquel que está formado por funciones lógicas elementales (Y, O, NAND, NOR, etc.), que tiene un determinado número de entradas y salidas cuyos valores dependen exclusivamente de los adoptados por las entradas.

Además del análisis de los bloques MSI, abordaremos el diseño de circuitos más complejos, a partir de las pastillas MSI estándar. Son ejemplos significativos de circuitos combinacionales de aplicación general, fabricados en una sola pastilla: *codificadores*, *decodificadores*, *multiplexadores*, *demultiplexadores* y *comparadores*. Son también circuitos combinacionales estándar los *generadores* y los *detectores de paridad* y los *convertidores de código*, aunque no sean dispositivos comerciales. De todo este tipo de funciones, así como de los dispositivos que las realizan, nos vamos a ocupar en los próximos apartados.

Es necesario indicar que existe un determinado número de circuitos que realizan la misma función. Nosotros estudiaremos los más representativos de cada tipo.

Los **codificadores** y los **decodificadores** son circuitos que realizan operaciones inversas. En el primer caso, el dispositivo transforma la información que utilizamos en nuestro lenguaje habitual (números decimales, letras, signos, etc.) en un conjunto de ceros (0) y de unos (1) para que sean procesados por el resto del sistema digital. Los *decodificadores* se encargan de proporcionar a su salida una información inteligible a partir de una combinación binaria presente en sus entradas.

De manera análoga, los **multiplexadores** y los **demultiplexadores** funcionan de forma complementaria. Los primeros se utilizan para enviar a su única salida la información presente en alguna de sus numerosas entradas. Los *demultiplexadores* hacen lo contrario, es decir, permiten enviar la información presente en su única entrada a cualquiera de sus múltiples salidas.

En ambos casos son necesarias unas entradas de control mediante las que se realiza la selección. Comercialmente, los demultiplexadores y los decodificadores son los mismos dispositivos.

Los **comparadores** admiten en sus entradas dos combinaciones binarias del mismo número de cifras e indican en sus salidas si esas combinaciones son iguales o no lo son. En este último caso, suelen indicar cuál de las dos es la mayor y cuál es la menor.

## 3.2 Sistema binario y códigos

Todos los circuitos digitales funcionan mediante la aplicación a sus entradas de señales digitales. Las salidas también proporcionan señales eléctricas de la misma forma. Este tipo de señales están formadas exclusivamente (véase Unidad 1) por dos niveles de tensión (nivel alto y nivel bajo) que se corresponden con los dos posibles estados estables (corte y saturación) de los elementos electrónicos básicos que constituyen las puertas y demás bloques integrados.

Cualquier información que se desee tratar, procesar o almacenar mediante sistemas digitales deberá ser traducida o **codificada** en un tipo de lenguaje apropiado. La forma correcta de hacerlo es convertir cualquier número, letra, signo, instrucción u operación en un conjunto de

señales eléctricas digitales que serán diferentes en cada caso. Cada uno de los datos estará constituido por una serie de unos y ceros que indicarán niveles altos o bajos de tensión. El número 9 (decimal), por ejemplo, podría representarse por *1001*, que es su equivalente en el *sistema binario*.

De la misma forma, a la hora de interpretar un resultado a la salida de un circuito digital, es necesario, tal como ya hemos señalado, traducir o **decodificar** el resultado transformando los ceros y los unos en datos que sean inteligibles. La codificación y la decodificación serán siempre operaciones imprescindibles en sistemas digitales que traten información, o en procesos industriales donde haya que suministrar datos o presentar resultados.



### 3. Análisis de circuitos combinacionales MSI

#### 3.2 Sistema binario y códigos

#### A. Sistema binario

Aunque en las Unidades anteriores hemos hecho alguna referencia a la equivalencia entre el sistema decimal y el sistema binario, éste es el momento de efectuar el estudio de este último sistema de numeración, describiendo los procesos de transformación de decimal a binario y viceversa.

El sistema en base dos utiliza únicamente los símbolos 0 y 1. A cada cifra o dígito de un número binario se le denomina **bit** (acrónimo de *Binary digiT*, «dígito binario»).

La combinación que representa un número, una letra, un signo o una orden, formada por un conjunto de bits, recibe el nombre de **palabra**. Como veremos en esta y en sucesivas Unidades, la *palabra* suele tener un formato concreto de 4, 8, 16, 32, ... bits.

Un número binario, como otro representado en cualquier base o sistema de numeración, puede presentarse de forma polinómica:

$$* a_n b^n + a_{n-1} b^{n-1} + \dots + a_1 b^1 + a_0 b^0 + a_{-1} b^{-1} + a_{-2} b^{-2} + \dots$$

Los coeficientes  $a_n, a_{n-1}, \dots, a_1, a_0$ , etc., representan ordenadamente las cifras del número binario y  $b$  es la base del sistema de numeración; es decir, en este caso:  $b = 2$ .

Si los términos se expresan en base diez y se suman todos ellos, se puede obtener el equivalente decimal del número binario al que representa.

Ejemplos:

- a) Pasar el número 101101 en base dos a decimal.

$$1 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 32 + 0 + 8 + 4 + 0 + 1 = 45$$

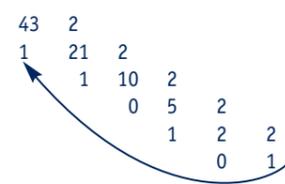
- b) Pasar 0,1001 a decimal.

$$1 \cdot 2^{-1} + 0 \cdot 2^{-2} + 0 \cdot 2^{-3} + 1 \cdot 2^{-4} = \frac{1}{2^1} + \frac{1}{2^4} = 0,5625$$

Para pasar un número decimal entero binario, se realizan divisiones sucesivas entre dos, hasta que el último cociente sea inferior a dos. El número binario será el formado por el último cociente, que será el bit de mayor peso, y los restos de cada división.

Ejemplo:

- Pasar 43 decimal a binario:



El número expresado en binario será 101011.

Para convertir a binario un número decimal fraccionario, se multiplica éste por dos. La parte decimal del resultado se vuelve a multiplicar por dos, y así sucesivamente, hasta que el resultado del producto sea un valor entero, o se obtenga la precisión deseada. El número binario quedará formado por la sucesión de las partes enteras resultantes de los productos.

Ejemplo:

- Pasar 0,5625 a binario:

$$\begin{array}{l} 0,5625 \cdot 2 = 1,1250 \\ 0,125 \cdot 2 = 0,250 \\ 0,25 \cdot 2 = 0,5 \\ 0,5 \cdot 2 = 1,0 \end{array} \downarrow$$

El número binario equivalente será 0,1001.

Como es lógico, con los números binarios también se pueden realizar operaciones matemáticas. De la *suma* y *resta* binaria nos ocuparemos en la Unidad 4, cuando examinemos los circuitos sumadores.

Otro sistema de numeración muy utilizado también en circuitos digitales es el hexadecimal, que desarrollaremos más adelante, en la Unidad 9, cuando tratemos la parte correspondiente a las memorias.

#### B. Códigos

Como hemos indicado anteriormente, la información que haya de ser procesada mediante circuitos digitales, o transmitida de unas unidades a otras, debe ser previamente codificada. En general, un **código** es un conjunto de unidades de información relacionadas de forma sistemática y biunívoca con otro conjunto de signos y símbolos según unas determinadas reglas de traducción fijadas de antemano. Los códigos que se utilizan en los sistemas digitales son binarios, es decir, combinaciones de unos y de ceros.

### 3. Análisis de circuitos combinatoriales MSI

#### 3.2 Sistema binario y códigos



La información que se puede codificar no se limita, exclusivamente, a cantidades numéricas, sino que se extiende a signos operativos, letras, instrucciones, etc. Cuando decíamos anteriormente que el número decimal 9 se representa en sistema binario como 1001, debe entenderse que el número 9 se codifica de esa manera para aplicar la información a un circuito lógico.

Cuando establecemos una relación análoga entre los números 0, 1, 2, 3, 4, 5, etc., y un conjunto de combinaciones binarias como la indicada, estamos creando uno de los múltiples códigos posibles. Los códigos más comunes utilizados para convertir cantidades numéricas son:

- Binario natural.
- Decimal Codificado en Binario o BCD (*Binary Coded Decimal*):
  - Natural.
  - Exceso tres.
  - Aiken.

Veremos su estructura y composición seguidamente, y comprobaremos sus ventajas e inconvenientes en la Unidad siguiente, cuando abordemos la suma y la resta binarias. En muchas ocasiones es necesario utilizar códigos *alfanuméricos* que conviertan números, letras, signos e instrucciones. El más utilizado es el **ASCII**.

Por otra parte, cuando la información deba ser transmitida, es conveniente utilizar códigos que verifiquen la veracidad de la comunicación *detectando* el error, en caso de producirse, o incluso *corrigiéndolo*. Los códigos detectores más utilizados son los de **paridad**. También son comunes otros, conocidos como *2 entre 5* y *2 entre 7* o *biquinario*. En estos dos últimos casos, las combinaciones están formadas por cinco y siete bits, respectivamente; de todos ellos, en ambos casos, solamente dos son unos.

Los códigos más habituales que se emplean para corregir un dato numérico mal transmitido son los de la familia **Hamming**.

#### ● Código binario natural

Consiste simplemente en representar, por el sistema de equivalencia deducido en el Apartado 3.2.A, cualquier número decimal (base diez) mediante la combinación binaria correspondiente.

#### ● Familia de códigos BCD

Como ya se ha indicado, esta familia de códigos es la más utilizada para representar información numérica. Para codificar un número decimal mediante este sistema, se representa por separado cada una de sus cifras. La cantidad de bits necesaria para representar cada cifra es cuatro. Con ellos se pueden efectuar  $2^4 = 16$  combinaciones distintas. Como en el sistema decimal el número de caracteres diferentes es igual a diez (del 0 al 9), siempre quedarán seis combinaciones inutilizadas.

La diferencia entre los códigos de esta familia, *natural*, *exceso tres* y *Aiken*, reside en las diez (de las dieciséis) combinaciones distintas que emplea cada uno de ellos:

- En **BCD natural** se utilizan, en orden creciente, las *diez primeras* combinaciones.
- En **BCD exceso tres** no se utilizan ni las tres primeras ni las tres últimas; en consecuencia, está formado por las *diez combinaciones intermedias*.
- En **BCD Aiken** se emplean las *cinco primeras* y las *cinco últimas*.

En la Tabla 3.1 se representan todos ellos y su equivalencia con el sistema decimal.

Sistema decimal	BCD natural	BCD exceso tres	BCD Aiken
0	0000	0011	0000
1	0001	0100	0001
2	0010	0101	0010
3	0011	0110	0011
4	0100	0111	0100
5	0101	1000	1011
6	0110	1001	1100
7	0111	1010	1101
8	1000	1011	1110
9	1001	1100	1111

Tabla 3.1. Equivalencia entre el sistema decimal y los códigos de la familia BCD.



### 3. Análisis de circuitos combinacionales MSI

#### 3.2 Sistema binario y códigos

0	0011	1
1	0100	0
2	0101	1
3	0110	1
4	0111	0
5	1000	0
6	1001	1
7	1010	1
8	1011	0
9	1100	1

Tabla 3.3. Código de paridad impar construido.

#### ASCII

El Código Normalizado Estadounidense para Intercambio de Información, aceptado internacionalmente y mucho más conocido como **ASCII** (*American Standard Code for Information Interchange*), permite representar cifras decimales, caracteres alfabéticos, signos especiales y diversas órdenes de control para periféricos (impresoras, pantallas, etcétera).

En la Tabla 3.2 se muestra el ASCII formado por ocho bits, de los cuales el de la izquierda es un bit de paridad. El bit  $b_7$  es el más significativo o de mayor peso de la combinación. En consecuencia, el formato queda constituido de la siguiente manera:

$P$	$b_7$	$b_6$	$b_5$	$b_4$	$b_3$	$b_2$	$b_1$
-----	-------	-------	-------	-------	-------	-------	-------

#### Códigos de paridad

Los **códigos de paridad** se forman añadiendo un bit más a los del código base. Pueden ser de paridad **par** o **impar**. En el primer caso, el número de unos, incluido el de paridad, debe ser par y en el segundo impar. En la Tabla 3.3 se muestra, como ejemplo, el código de paridad impar formado a partir del BCD exceso tres.

El **bit de paridad** se genera mediante un circuito combinacional muy sencillo denominado *generador de paridad*, que se construye con puertas 0 exclusiva. La detección se realiza comprobando (mediante un circuito detector, formado también por puertas 0 exclusiva) que el número de unos en cada combinación es siempre par o impar, según el caso.

$b_4$	$b_3$	$b_2$	$b_1$	$b_7$	0	1	2	3	4	5	6	7
0	0	0	0	0	0	1	2	3	4	5	6	7
0	0	0	1	1	0	1	2	3	4	5	6	7
0	0	1	0	2	0	1	2	3	4	5	6	7
0	0	1	1	3	0	1	2	3	4	5	6	7
0	1	0	0	4	0	1	2	3	4	5	6	7
0	1	0	1	5	0	1	2	3	4	5	6	7
0	1	1	0	6	0	1	2	3	4	5	6	7
0	1	1	1	7	0	1	2	3	4	5	6	7
1	0	0	0	8	0	1	2	3	4	5	6	7
1	0	0	1	9	0	1	2	3	4	5	6	7
1	0	1	0	10	0	1	2	3	4	5	6	7
1	0	1	1	11	0	1	2	3	4	5	6	7
1	1	0	0	12	0	1	2	3	4	5	6	7
1	1	0	1	13	0	1	2	3	4	5	6	7
1	1	1	0	14	0	1	2	3	4	5	6	7
1	1	1	1	15	0	1	2	3	4	5	6	7

Tabla 3.2. ASCII de 7 bits.

### 3. Análisis de circuitos combinacionales MSI

#### 3.2 Sistema binario y códigos



#### ◆ Código Hamming

Los **códigos correctores** proporcionan el lugar que ocupa el bit erróneo. Mediante el circuito adecuado se puede corregir automáticamente el fallo detectado en la información recibida. Se utilizan, fundamentalmente, en procesos industriales.

El código corrector más utilizado es el **Hamming**, en el cual cada combinación está formada por siete bits y para cuya construcción se parte también de los códigos de la familia BCD. En la Tabla 3.4 aparece el código Hamming formado a partir del BCD natural.

Las columnas  $b_7$ ,  $b_6$ ,  $b_5$  y  $b_3$  corresponden al código BCD natural. Las  $b_7$ ,  $b_2$  y  $b_4$  se construyen de forma que en cada una de las siguientes combinaciones el número de unos sea par:

$$b_1-b_3-b_5-b_7$$

$$b_2-b_3-b_6-b_7$$

$$b_4-b_5-b_6-b_7$$

El bit  $b_1$  se puede obtener a partir de la siguiente expresión:

$$b_4 = b_3 \oplus b_5 \oplus b_7$$

Es decir, mediante una función 0 exclusiva de tres variables. De la misma manera,  $b_2$  y  $b_4$  se generan con puertas del mismo tipo, tomando como referencia las funciones:

$$b_2 = b_3 \oplus b_6 \oplus b_7$$

$$b_4 = b_5 \oplus b_6 \oplus b_7$$

Por otra parte, el detector de errores debe generar tres funciones que cumplan las condiciones siguientes:

$$C_1 = b_1 \oplus b_3 \oplus b_5 \oplus b_7$$

$$C_2 = b_2 \oplus b_3 \oplus b_6 \oplus b_7$$

$$C_3 = b_4 \oplus b_5 \oplus b_6 \oplus b_7$$

El número decimal equivalente a la combinación binaria  $C_3 C_2 C_1$  indicará el lugar del bit cuyo valor se ha invertido. El propio sistema receptor debe ser capaz de cambiar el valor del bit erróneo o mal transmitido por su valor adecuado, mediante un circuito *corrector*. Cuando no exista error en la transmisión, el valor de las funciones  $C_1$ ,  $C_2$  y  $C_3$  será cero en los tres casos, porque, en las combinaciones que se obtienen al asignar valores a los diferentes bits que las constituyen, el número de unos será par.

Para comprender la forma de operar, supongamos que al transmitir el número *siete decimal*, cuyo valor codificado mediante Hamming es 0110100, se produce un error y la combinación recibida es 0110000. El valor de cada una de las funciones  $C_1$ ,  $C_2$  y  $C_3$  será:

$$C_1 = 0 \oplus 0 \oplus 1 \oplus 0 = 1$$

$$C_2 = 0 \oplus 0 \oplus 1 \oplus 0 = 1$$

$$C_3 = 0 \oplus 1 \oplus 1 \oplus 0 = 0$$

La combinación  $C_3 C_2 C_1$  será en este caso 011, que equivale al número tres decimal. Esto quiere decir que el bit equivocado es el  $b_3$ , es decir, el tercero comenzando a contar por la derecha.

	$b_7$	$b_6$	$b_5$	$b_4$	$b_3$	$b_2$	$b_1$
0	0	0	0	0	0	0	0
1	0	0	0	0	1	1	1
2	0	0	1	1	0	0	1
3	0	0	1	1	1	1	0
4	0	1	0	1	0	1	0
5	0	1	0	1	1	0	1
6	0	1	1	0	0	1	1
7	0	1	1	0	1	0	0
8	1	0	0	1	0	1	1
9	1	0	0	1	1	0	0

Tabla 3.4. Código Hamming formado a partir del BCD natural.



### 3.3 Análisis de codificadores



#### Actividad en el aula

##### 1 Ensayo y experimentación con un circuito codificador 74148

En la Figura 3.1 se muestra el símbolo y el *diagrama de conexión* del codificador prioritario 74148, fabricado en tecnología TTL.

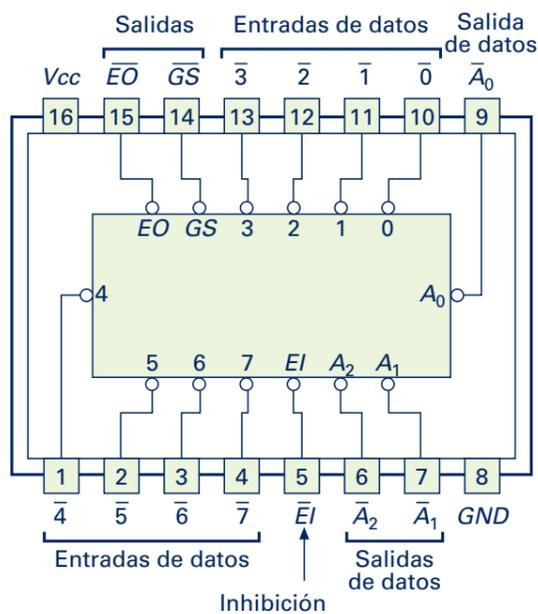


Fig. 3.1. Símbolo lógico y diagrama de conexión del 74148.

- Conectar las salidas a tres puertas inversoras y completar la Tabla 3.5.
- Conectar la entrada  $\bar{E}I$  (Enable Input) a un nivel bajo de tensión.
- Dibujar el circuito completo (incluyendo el CI 7404, los dispositivos de entrada y los señalizadores de salida).
- Responder a las siguientes preguntas:
  - ¿Cuál es el *nivel activo* que hay que aplicar a las entradas?
  - ¿Qué quiere decir que el decodificador es prioritario? Comprobar lo que ocurre cuando se activan simultáneamente las Entradas 2, 3 y 4.
- Observando la Tabla 3.5, indicar:
  - ¿Qué función cumple la entrada  $\bar{E}I$ ? Observar qué ocurre cuando aparece el nivel bajo de tensión en esta entrada.
  - ¿Qué información suministran las salidas EO (Enable Output) y  $\bar{G}S$  (Group Signal)?

Entradas								Salidas		
0	1	2	3	4	5	6	7	$A_2$	$A_1$	$A_0$
X	X	X	X	X	X	X	0			
X	X	X	X	X	X	0	1			
X	X	X	X	X	0	1	1			
X	X	X	X	0	1	1	1			
X	X	0	1	1	1	1	1			
X	0	1	1	1	1	1	1			
0	1	1	1	1	1	1	1			

Tabla 3.5.

#### A. Estudio del dispositivo 74148

El codificador 74148 es uno de los pocos circuitos de este tipo, fabricados en tecnología MSI, que se pueden encontrar en los catálogos de componentes. Es un dispositivo con ocho líneas de entrada y tres salidas. Con un solo circuito de estas características es posible codificar en binario los ocho primeros números del sistema decimal; sin embargo, es posible conectar en cascada varios dispositivos para codificar una cantidad mayor de números. En cualquier caso, en este tipo

de dispositivos o circuitos, las entradas y las salidas deben estar relacionadas mediante la expresión:  $N = 2^n$ , donde  $N$  es el número de entradas y  $n$  es el número de salidas.

En la Tabla 3.6 se muestran todas las salidas y todas las entradas del codificador. En este caso, la tabla de verdad se diferencia de las representadas en los capítulos anteriores, ya que, en la parte de la izquierda, no aparecen todas las combinaciones que se pueden construir con las variables de entrada.

### 3. Análisis de circuitos combinatoriales MSI

#### 3.3 Análisis de codificadores



En la tabla del codificador 74148 es suficiente con recoger aquellas combinaciones que se corresponden con los ocho primeros números del sistema decimal. Por otra parte, veremos que aparecen entradas y salidas complementarias o de control que completan las prestaciones de los dispositivos.

Como se puede observar en este caso, el *nivel activo* a la entrada es el 0. Por otra parte, las salidas proporcionan el valor codificado de la entrada activa en *forma negada*.

Por las razones expuestas, las variables de entrada y las funciones de salida aparecen con un signo de inversión en la Tabla 3.6 y con un círculo o un triángulo en el diagrama de conexión de la Figura 3.1.

Estos signos serán frecuentes en los diagramas y tablas de todos los dispositivos a partir de ahora. El terminal  $\overline{EI}$  es una entrada de *habilitación* o inhibición que permite codificar al dispositivo cuando se le aplica un nivel bajo ( $L$ , de *low*).

En caso contrario, es decir, cuando está a nivel alto ( $H$ , de *high*), sea cual sea el estado de las entradas, el circuito se inhibe y no codifica, apareciendo un nivel alto en todas y cada una de las salidas, incluidas  $\overline{EO}$  y  $\overline{GS}$ .

La salida habilitada,  $\overline{EO}$ , indica, mediante un nivel bajo, que ninguna entrada está activada, estando habilitado el dispositivo para codificar ( $\overline{EI} = L$ ).

Por el contrario, si al menos una de las líneas está activada,  $\overline{EO}$  responderá con un nivel alto. Por último,  $\overline{GS}$  muestra un nivel bajo cuando alguna de las entradas está activada, siendo su estado el inverso cuando todas las entradas están inactivas o el circuito está inhibido.

$\overline{EI}$	Entradas								Salidas				
	0	1	2	3	4	5	6	7	$\overline{A_2}$	$\overline{A_1}$	$\overline{A_0}$	$\overline{GS}$	$\overline{EO}$
H	X	X	X	X	X	X	X	X	H	H	H	H	H
L	H	H	H	H	H	H	H	H	H	H	H	H	L
L	X	X	X	X	X	X	X	L	L	L	L	L	H
L	X	X	X	X	X	X	L	H	L	L	H	L	H
L	X	X	X	X	X	L	H	H	L	H	L	L	H
L	X	X	X	L	H	H	H	H	L	H	L	L	H
X	X	X	L	H	H	H	H	H	H	L	L	L	H
L	X	X	L	H	H	H	H	H	H	L	H	L	H
L	X	L	H	H	H	H	H	H	H	H	L	L	H
L	H	H	H	H	H	H	H	H	H	H	H	L	H

Tabla 3.6. Tabla de verdad del codificador 74148.

Como veremos, estos tres terminales juegan un papel importante cuando se conectan varios dispositivos en cascada. Observando la parte de las líneas de entrada ( $\overline{0}$  a  $\overline{7}$ ) de la Tabla 3.6, podemos comprobar que aparece una cantidad considerable de términos indiferentes. Esto indica que, cuando una determinada entrada está activada (nivel  $L$ ), el dispositivo codifica el número correspondiente a esa entrada sea cual sea el estado de todas las líneas que se encuentren a su izquierda; es decir, el circuito reconoce la entrada de valor más alto. Por esta razón se dice que el 74148 es un codificador *prioritario*.

Es posible obtener las funciones del codificador ( $\overline{A_2}$ ,  $\overline{A_1}$ ,  $\overline{A_0}$ ,  $\overline{GS}$  y  $\overline{EO}$ ) a partir de las variables de entrada ( $\overline{EI}$ ,  $\overline{0}$ ,  $\overline{1}$ ,  $\overline{2}$ ,  $\overline{3}$ ,  $\overline{4}$ ,  $\overline{5}$ ,  $\overline{6}$  y  $\overline{7}$ ). Para ello, es necesario considerar los términos indiferentes de las entradas como ceros y como unos, de tal manera que cada uno de ellos dará lugar a dos combinaciones diferentes. El proceso es largo y complejo. Como todo circuito combinatorial, el decodificador 74148 está constituido por un conjunto de puertas lógicas de varios tipos. En la Figura 3.2 se muestra el diagrama lógico completo. Las puertas que tienen un círculo a la entrada son inversores normales. El símbolo indica que el nivel activo es el cero, pero su funcionamiento es idéntico al de la función complemento analizada en la Unidad 1.

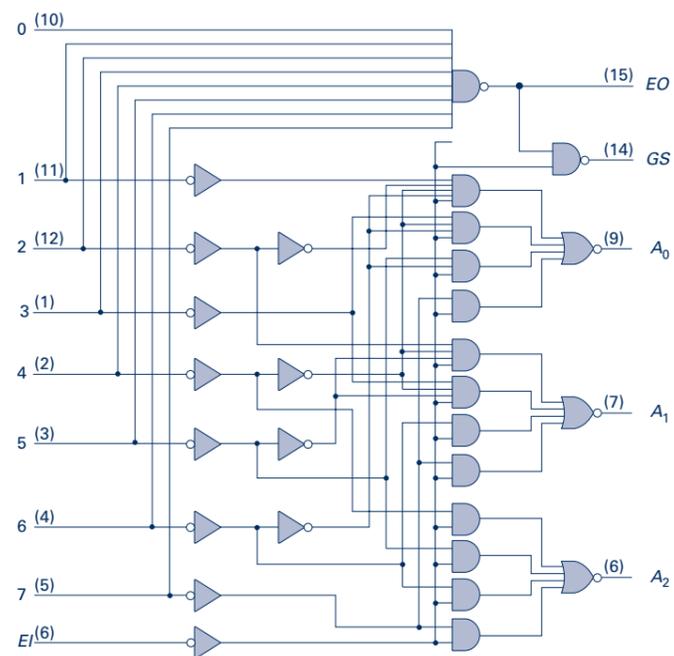


Fig. 3.2. Diagrama lógico del codificador 74148.



### 3. Análisis de circuitos combinacionales MSI

#### 3.4 Análisis de decodificadores

## 3.4 Análisis de decodificadores



### Actividad en el aula

#### 2 Ensayo y experimentación con un circuito formado por un decodificador BCD/7 segmentos y un display 3

En la Figura 3.3 se muestra el símbolo y el diagrama de conexión del decodificador 7447 y en la Figura 3.4 el circuito de conexión a un *display* de 7 segmentos. Este elemento está constituido por siete LED (diodos emisores de luz) cuyos ánodos están conectados a un punto común,  $V_{cc}$ , de alimentación.

- Representar, en la Tabla 3.7, la relación entre todas las posibles combinaciones binarias aplicadas a las entradas y la información suministrada por el *display*, rellenando los segmentos que se iluminan en cada caso.

- Responder a las siguientes preguntas:
  - ¿Qué función realiza la entrada  $\overline{LT}$  (*Lamp Test*)? Aplicar un nivel bajo a esta entrada.
  - ¿Qué función realiza la entrada  $\overline{RBI}$  (*Ripple Blanking Input*)? Conectarla a nivel bajo y aplicar la combinación 0000 a las entradas *ABCD*.
  - Observar la Tabla 3.9 e indicar en qué casos  $\overline{RB}/\overline{RBO}$  (*Blanking Input/Ripple Blanking Output*) se comporta como entrada y en qué otros como salida.

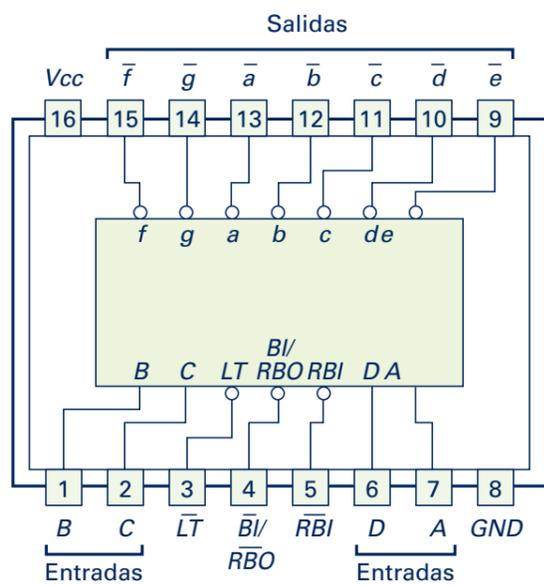


Fig. 3.3. Símbolo lógico y diagrama de conexión del decodificador 7447.

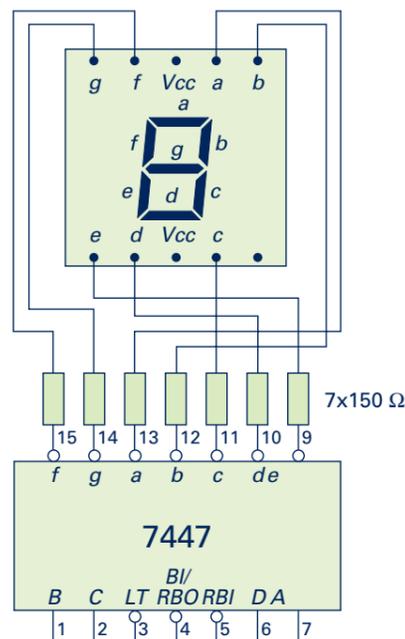


Fig. 3.4. Conexión del decodificador 7447 a un *display* de 7 segmentos.

0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

Tabla 3.7.

### 3. Análisis de circuitos combinatoriales MSI

#### 3.4 Análisis de decodificadores



#### Estudio del circuito y del decodificador 7447

El decodificador 7447 que forma parte del circuito de la Figura 3.4 es un dispositivo que dispone de cuatro entradas de datos y dos de control, siete salidas y otro terminal que puede ser utilizado como entrada o como salida. La Tabla 3.8 muestra el comportamiento de este componente. Este elemento permite visualizar en un *display* los números comprendidos entre el cero y el nueve, cuando se aplica a sus entradas la combinación binaria correspondiente de cuatro bits. En general, al contrario que en los codificadores, los decodificadores disponen de  $N$  salidas y de  $n$  entradas de manera que  $N = 2^n$ . En este caso concreto, son necesarias cuatro entradas para poder decodificar diez signos diferentes, pero el número de salidas (como debería corresponderle) no es de 16, sino que está determinado por el dispositivo externo al que se conecta, que dispone de siete entradas (una para cada segmento).

En consecuencia, aquí no se cumple la norma. Esto otorga al 7447 el sobrenombre de *convertidor de código*.

Como se puede comprobar, en este caso el nivel activo a la entrada es el uno, es decir, las combinaciones binarias aplicadas a las entradas  $A$ ,  $B$ ,  $C$  y  $D$  deben presentarse de forma directa. Sin embargo, las salidas que activan en cada caso los segmentos correspondientes al número decimal equivalente a la entrada binaria muestran un nivel lógico cero.

La entrada  $\overline{LT}$  (*Lamp Test*) se utiliza para comprobar, mediante un nivel activo bajo, el estado de los diodos que constituyen los segmentos del *display*. En condiciones normales,  $\overline{LT}$  debe permanecer a nivel lógico alto y, cuando se aplica un nivel bajo, sea cual sea el estado de las otras entradas, todas las salidas pasan a nivel bajo, iluminándose los siete segmentos.

N.º decimal o función decimal	Entradas						BI/RBO	Salidas						
	$\overline{LT}$	$\overline{RBI}$	D	C	B	A		$\overline{a}$	$\overline{b}$	$\overline{c}$	$\overline{d}$	$\overline{e}$	$\overline{f}$	$\overline{g}$
0	H	H	L	L	L	L	H	L	L	L	L	L	L	H
1	H	X	L	L	L	H	H	H	L	L	H	H	H	H
2	H	X	L	L	H	L	H	L	L	H	L	L	H	L
3	H	X	L	L	H	H	H	L	L	L	L	H	H	L
4	H	X	L	H	L	L	H	H	L	L	H	H	L	L
5	H	X	L	H	L	H	H	L	H	L	L	H	L	L
6	H	X	L	H	H	L	H	H	H	L	L	L	L	L
7	H	X	L	H	H	H	H	L	L	L	H	H	H	H
8	H	X	H	L	L	L	H	L	L	L	L	L	L	L
9	H	X	H	L	L	H	H	L	L	L	H	H	L	L
10	H	X	H	L	H	L	H	H	H	H	L	L	H	L
11	H	X	H	L	H	H	H	H	H	L	L	H	H	L
12	H	X	H	H	L	L	H	H	L	H	H	H	L	L
13	H	X	H	H	L	H	H	L	H	H	L	H	L	L
14	H	X	H	H	H	L	H	H	H	H	L	L	L	L
15	H	X	H	H	H	H	H	H	H	H	H	H	H	H
BI	X	X	X	X	X	X	L	H	H	H	H	H	H	H
RBI	H	L	L	L	L	L	L	H	H	H	H	H	H	H
LT	L	X	X	X	X	X	H	L	L	L	L	L	L	L

Tabla 3.8. Tabla de verdad del decodificador 7447.



### 3. Análisis de circuitos combinacionales MSI

#### 3.5 Análisis de multiplexadores

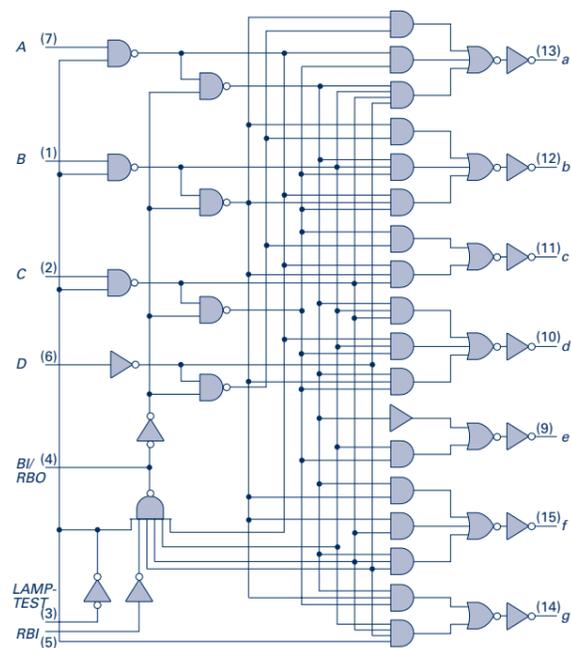


Fig. 3.5. Diagrama lógico del decodificador 7447.

Por el contrario, la función  $\overline{RBI}$  (*Ripple Blanking Input*), a través de la entrada del mismo nombre, apaga todos los segmentos del *display* aplicando un nivel alto a las salidas, cuando todas las entradas *A*, *B*, *C* y *D* se encuentran en ese mismo estado lógico y la entrada *LT* a nivel alto.

Por tanto, esta línea se utiliza para suprimir el cero.  $\overline{RB}/\overline{RBI}$  (*Blanking Input-Ripple Blanking Output*) es una salida que siempre muestra un nivel alto, salvo en las condiciones expuestas anteriormente, es decir, cuando  $\overline{RBI}$  y *A*, *B*, *C* y *D* están a nivel bajo.

Por otra parte, este mismo terminal puede ser utilizado como entrada, de tal manera que, cuando se aplica directamente un nivel bajo, todas las salidas muestran un nivel alto, sea cual sea el estado de todas las demás entradas.

Como en el caso del apartado anterior, es posible deducir las funciones de salida a partir de las variables de entrada, pero el proceso también es aquí muy laborioso. Por último, en la Figura 3.5 de la página siguiente se muestra el diagrama lógico del decodificador 7447, formado por un conjunto de puertas de diferentes tipos.

### 3.5 Análisis de multiplexadores

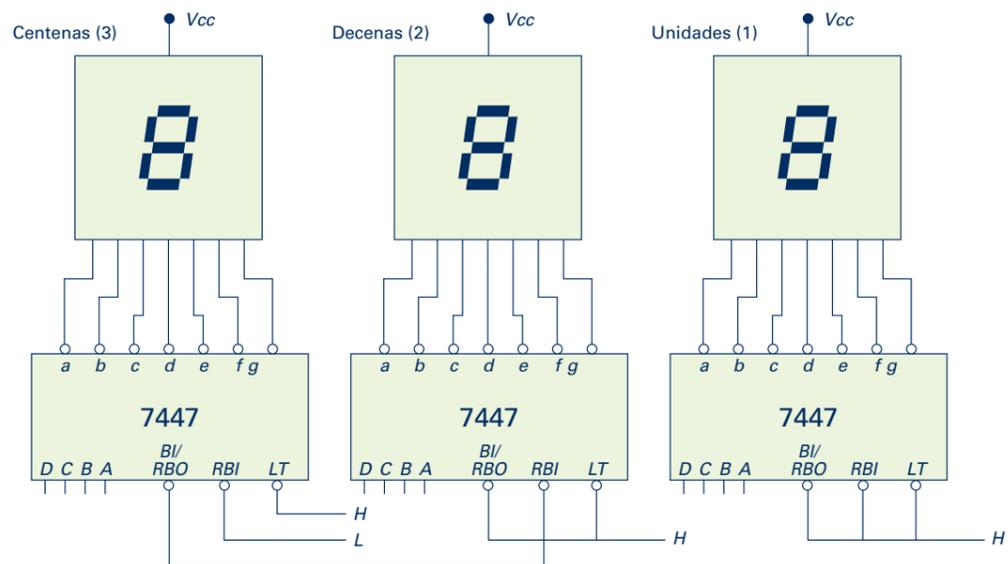


Fig. 3.6. Visualizador numérico de tres dígitos.

### 3. Análisis de circuitos combinatoriales MSI

#### 3.5 Análisis de multiplexadores



#### Actividad en el aula

##### 3 Ensayo y experimentación con un multiplexador 74151

En la Figura 3.7 se muestra el símbolo y el diagrama de conexión del multiplexador 74151 fabricado en tecnología TTL.

- Conectar la entrada de inhibición  $\bar{S}$  a nivel bajo y completar la Tabla 3.9.

- Dibujar el circuito completo (incluyendo los dispositivos de entrada y los señalizadores de salida).
- Comparar los valores obtenidos en  $Y$  con los de las entradas de datos para cada combinación de  $CBA$ .

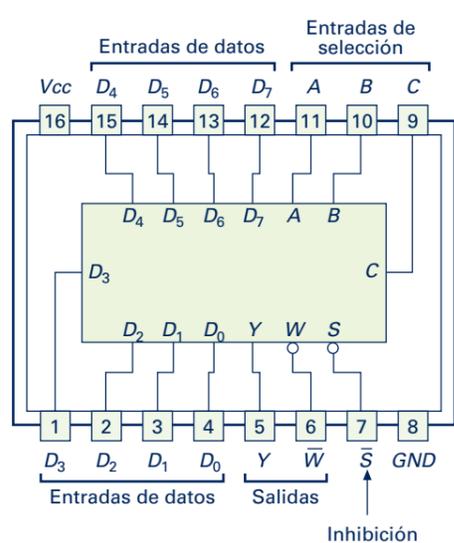


Fig. 3.7. Símbolo y diagrama de conexión del multiplexador 74151.

Líneas de selección			Entradas								Salidas	
C	B	A	D <sub>0</sub>	D <sub>1</sub>	D <sub>2</sub>	D <sub>3</sub>	D <sub>4</sub>	D <sub>5</sub>	D <sub>6</sub>	D <sub>7</sub>	Y	W
0	0	0	0	X	X	X	X	X	X	X		
0	0	0	1	X	X	X	X	X	X	X		
0	0	1	X	0	X	X	X	X	X	X		
0	0	1	X	1	X	X	X	X	X	X		
0	1	0	X	X	0	X	X	X	X	X		
0	1	0	X	X	1	X	X	X	X	X		
0	1	1	X	X	X	0	X	X	X	X		
0	1	1	X	X	X	1	X	X	X	X		
1	0	0	X	X	X	X	0	X	X	X		
1	0	0	X	X	X	X	1	X	X	X		
1	0	1	X	X	X	X	X	0	X	X		
1	0	1	X	X	X	X	X	1	X	X		
1	1	1	0	X	X	X	X	X	0	X		
1	1	0	X	X	X	X	X	X	1	X		
1	1	1	X	X	X	X	X	X	X	0		
X	1	1	1	X	X	X	X	X	X	1		

Tabla 3.9.

##### Estudio del dispositivo 74151

El 74151 es un multiplexador de ocho líneas de entrada ( $D_0$  a  $D_7$ ), tres entradas de selección o control ( $A$ ,  $B$  y  $C$ ), una entrada  $\bar{S}$  que inhibe o habilita al dispositivo y dos salidas complementarias:  $Y$  y  $\bar{W}$ .

En un circuito de este tipo, la relación entre el número de líneas de entrada de información y el número de entradas de selección  $n$  debe ajustarse siempre a la siguiente regla:  $N = 2^n$ .

La Tabla 3.10 es una síntesis de la tabla de verdad del dispositivo en la que se muestra, de manera ingeniosa, el comportamiento del multiplexador.

Si se representase la tabla de verdad completa, sería necesario ordenar todas las combinaciones que se pueden obtener a partir de 11 variables (las tres entradas de selección, más las ocho líneas de datos).



### 3. Análisis de circuitos combinacionales MSI

#### 3.6 Análisis de multiplexadores

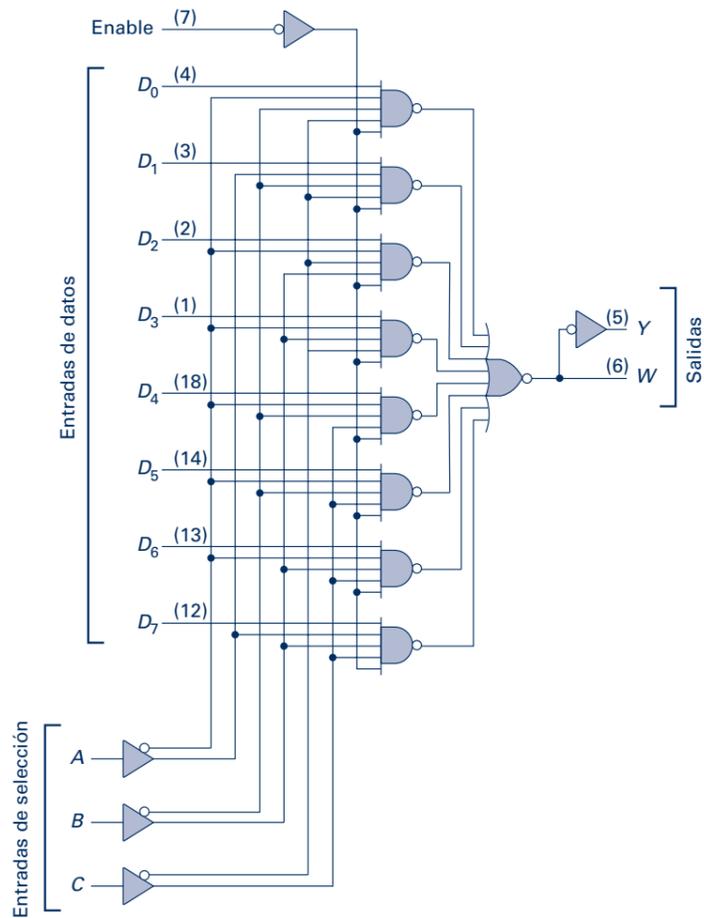


Fig. 3.8. Diagrama lógico del multiplexador 74151.

A través de las tres entradas de control, este componente permite seleccionar cada una de sus ocho líneas de datos y presentar la información correspondiente en las salidas  $Y$  y  $\bar{W}$ .

Como se puede comprobar, un nivel alto en la entrada  $\bar{S}$  deja fijo el valor de las salidas  $Y$  y  $\bar{W}$ , sea cual sea la combinación binaria presente en las entradas de selección. Un nivel bajo aplicado a dicha entrada habilita al dispositivo para su funcionamiento como multiplexador.

La función del multiplexador 74151, obtenida de la Tabla 3.10, puede expresarse de la siguiente manera cuando la entrada de inhibición se encuentra a nivel bajo:

$$F = \bar{A}\bar{B}\bar{C}D_0 + \bar{A}\bar{B}\bar{C}D_1 + \bar{A}\bar{B}\bar{C}D_2 + \bar{A}\bar{B}\bar{C}D_3 + \bar{A}\bar{B}\bar{C}D_4 + \bar{A}\bar{B}\bar{C}D_5 + \bar{A}\bar{B}\bar{C}D_6 + \bar{A}\bar{B}\bar{C}D_7$$

Entradas				Salidas	
Selección			Inhibición	$Y$	$\bar{W}$
$C$	$B$	$A$	$\bar{S}$		
X	X	X	X	X	H
L	L	L	L	$D_0$	$\bar{D}_0$
L	L	H	L	$D_1$	$\bar{D}_1$
L	H	L	L	$D_2$	$\bar{D}_2$
L	H	H	L	$D_3$	$\bar{D}_3$
H	L	L	L	$D_4$	$\bar{D}_4$
H	L	H	L	$D_5$	$\bar{D}_5$
H	H	L	L	$D_6$	$\bar{D}_6$
H	H	H	L	$D_7$	$\bar{D}_7$

Tabla 3.10. Tabla de verdad del multiplexador 74151.

El diagrama lógico, en el que se contemplan todas las entradas y todas las salidas, es el que se muestra en la Figura 3.8.



### 3.6 Análisis de decodificadores/demultiplexadores



#### Actividad en el aula

##### 4 Ensayo del circuito 7442 utilizado como decodificador

En la Figura 3.9 se muestra el símbolo lógico y el diagrama de conexión del decodificador/demultiplexador 7442.

- Completar la Tabla 3.11.
- Dibujar el circuito completo (incluyendo los dispositivos de entrada y los señalizadores de salida).

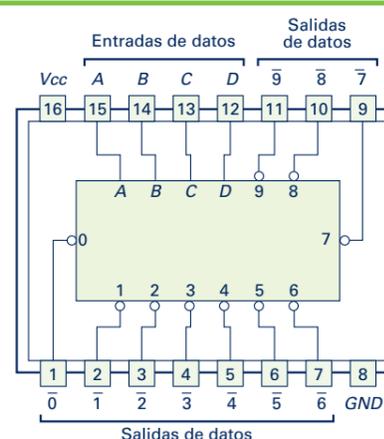


Fig. 3.9. Símbolo lógico y diagrama de conexión del circuito 7442.



#### Actividad en el aula

##### 5 Ensayo del circuito 7442 como demultiplexador

Montar el circuito utilizando  $A$ ,  $B$  y  $C$  como entradas de selección y la línea  $D$  como entrada de datos. Completar la Tabla 3.12.

Entradas				Salidas									
$D$	$C$	$B$	$A$	$\bar{0}$	$\bar{1}$	$\bar{2}$	$\bar{3}$	$\bar{4}$	$\bar{5}$	$\bar{6}$	$\bar{7}$	$\bar{8}$	$\bar{9}$
0	0	0	0										
0	0	0	1										
0	0	1	0										
0	0	1	1										
0	1	0	0										
0	1	1	0										
0	1	1	1										
1	0	0	0										
1	0	0	1										

Tabla 3.11.

Entradas				Salidas							
$D$	$B$	$A$	$D$	$\bar{0}$	$\bar{1}$	$\bar{2}$	$\bar{3}$	$\bar{4}$	$\bar{5}$	$\bar{6}$	$\bar{7}$
0	0	0	0								
0	0	0	1								
0	0	1	0								
0	0	1	1								
0	1	0	0								
0	1	0	1								
0	1	1	0								
0	1	1	1								
1	0	0	0								
1	0	0	1								
1	0	1	0								
1	0	1	1								
1	1	0	0								
1	1	0	1								
1	1	1	0								
1	1	1	1								

Tabla 3.12.



### 3. Análisis de circuitos combinacionales MSI

#### 3.6 Análisis de decodificadores/demultiplexadores

##### Estudio del dispositivo 7442

El 7442 es un dispositivo que tiene cuatro líneas de entrada y diez de salida. En la Tabla 3.13 se muestra la relación entre entradas y salidas.

Como se puede comprobar, el nivel activo de salida es el bajo (L). Del análisis de dicha tabla deduciremos su funcionamiento como decodificador y como demultiplexador.

Entradas				Salidas									
D	C	B	A	0	1	2	3	4	5	6	7	8	9
L	L	L	L	L	L	H	H	H	H	H	H	H	H
L	L	L	H	H	L	H	H	H	H	H	H	H	H
L	L	H	L	H	H	L	H	H	H	H	H	H	H
L	L	H	H	H	H	L	H	H	H	H	H	H	H
L	H	L	L	H	H	H	L	H	H	H	H	H	H
L	H	L	H	H	H	H	L	H	H	H	H	H	H
L	H	H	L	H	H	H	H	L	H	H	H	H	H
L	H	H	H	H	H	H	H	L	H	H	H	H	H
H	L	L	L	H	H	H	H	H	H	H	L	H	H
H	L	L	H	H	H	H	H	H	H	H	H	L	H
H	L	H	L	H	H	H	H	H	H	H	H	H	L
H	L	H	H	H	H	H	H	H	H	H	H	H	L
H	H	L	L	H	H	H	H	H	H	H	H	H	L
H	H	L	H	H	H	H	H	H	H	H	H	H	L
H	H	H	L	H	H	H	H	H	H	H	H	H	L
H	H	H	H	H	H	H	H	H	H	H	H	H	L

Tabla 3.13. Tabla de verdad del 7442.

##### Funcionamiento como decodificador

Las diez primeras líneas de la Tabla 3.13 muestran el comportamiento del 7442 como decodificador. Al presentar en sus cuatro entradas  $D$ ,  $C$ ,  $B$  y  $A$  las combinaciones binarias comprendidas entre la 0000 (LLLL) y la 1001 (HLLH), se activa la salida correspondiente al número decimal equivalente. Por esta razón, se conoce como *decodificador BCD/decimal*.

##### Funcionamiento como demultiplexador

Recordemos que un demultiplexador funciona de forma inversa a un multiplexador; es decir, la información presente en su única entrada puede aparecer en cualquiera de sus múltiples salidas mediante la conveniente *selección* de las entradas destinadas a tal fin.

En la Figura 3.10 se muestra el diagrama lógico del decodificador 7442.

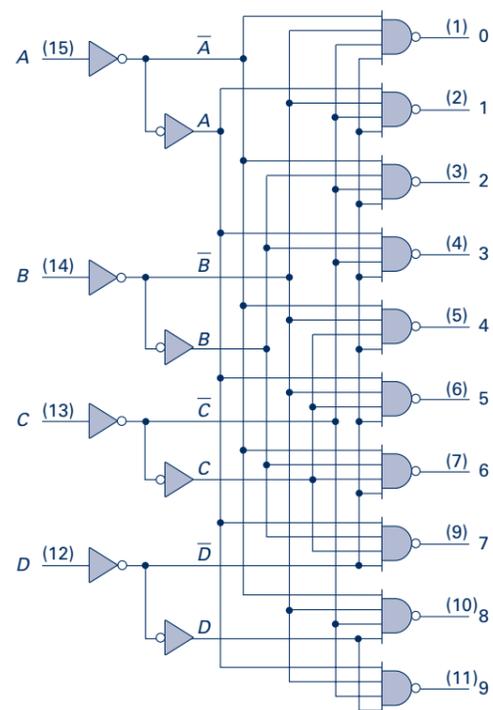


Fig. 3.10. Diagrama lógico del decodificador/demultiplexador 7442.

En el caso del 7442, la línea  $D$  puede ser utilizada como entrada de datos y las  $C$ ,  $B$  y  $A$  como entradas de selección. El dispositivo así configurado dispone de ocho salidas (de la 0 a la 7).

Para comprender el funcionamiento como demultiplexador, resulta de gran ayuda detenerse en las diagonales marcadas en la Tabla 3.13.

Se podrá observar, en las ocho primeras combinaciones de la parte de la derecha, que cada una de las salidas activas muestra un nivel bajo cuando la entrada de datos  $D$  está a nivel bajo, mientras que en las ocho últimas líneas, donde la entrada de datos  $D$  tiene nivel alto, la salida activa en cada combinación muestra un nivel alto.



## 3.7 Análisis de comparadores



### Actividad en el aula

#### 6 Ensayo y experimentación con un comparador 7485

En la Figura 3.11 se muestra el símbolo y el diagrama de conexión del comparador 7485.

- Conectar las entradas  $A > B$  y  $A < B$  a nivel bajo y la entrada  $A = B$  a nivel alto. Completar la Tabla 3.14.

- ¿Qué ocurre en las salidas, cuando las dos palabras son iguales, si las entradas  $A > B$ ,  $A < B$  y  $A = B$  se conectan a niveles HLL o LHL, respectivamente? Comprobar experimentalmente esta situación.

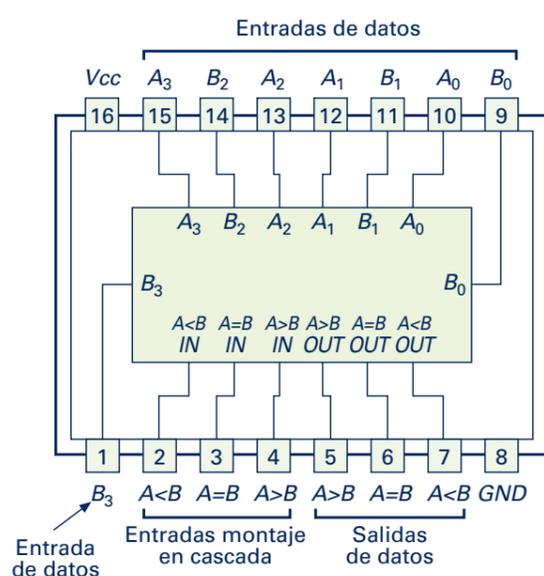


Fig. 3.11. Símbolo lógico y diagrama de conexión del comparador 7485.

Entradas								Salidas		
$A_3$	$A_2$	$A_1$	$A_0$	$B_3$	$B_2$	$B_1$	$B_0$	$A = B$	$A > B$	$A < B$
0	0	0	0	0	0	0	0			
1	X	X	X	0	X	X	X			
0	X	X	X	1	X	X	X			
1	0	0	0	1	0	0	0			
1	1	X	X	1	0	X	X			
1	0	X	X	1	1	X	X			
1	1	0	0	1	1	0	0			
1	1	1	X	1	1	0	X			
1	1	0	X	1	1	1	X			
1	1	1	0	1	1	1	0			
1	1	1	1	1	1	1	0			
1	1	1	0	1	1	1	1			

Tabla 3.14.

### A. Estudio del dispositivo 7485

El circuito integrado 7485 es un comparador de dos palabras de cuatro bits cada una. Como se puede comprobar en la Figura 3.12, el dispositivo dispone de ocho entradas de datos, tres salidas de información y tres entradas de expansión. En la Figura 3.13 se muestra el diagrama lógico.

El dispositivo puede operar individualmente, en serie o en paralelo con otros del mismo tipo. El funcionamiento en cualquiera de las tres modalidades se resume en la Tabla 3.15.

Las tres últimas líneas están dedicadas, exclusivamente, al funcionamiento en paralelo.

Si se observa la línea undécima de la tabla de verdad, comprobaremos que, en caso de utilizar un solo elemento, las entradas de expansión deben conectarse como se ha señalado en la Actividad de aula 7 de esta Unidad, es decir, las líneas  $A > B$  y  $A < B$  a nivel bajo y  $A = B$  a nivel alto.

Aunque, si miramos la línea siguiente, vemos que es suficiente con que la entrada  $A = B$  esté a nivel alto. De no ser así, la salida  $A = B$  nunca mostraría un nivel alto.



### 3. Análisis de circuitos combinacionales MSI

#### 3.7 Análisis de comparadores

#### B. Estudio de un circuito serie

La conexión en serie de dos 7485, tal como se muestra en la Figura 3.12, permite comparar dos palabras de ocho bits cada una. En este caso, las tres salidas del componente de los cuatro bits menos significativos deben conectarse a las tres entradas de expansión del dispositivo de los cuatro bits de mayor peso.

Esta técnica de conexionado puede ser empleada para comparar palabras de más bits, acoplando otros componentes.

El gran inconveniente de la conexión serie es la acumulación de los tiempos de conmutación de cada uno de los circuitos. La respuesta definitiva a las salidas de la última etapa no se produce hasta que no se comparan los bits menos significativos de todos los dispositivos anteriores.

Entradas				Entradas montaje paralelo			Salidas		
$A_3, B_3$	$A_2, B_2$	$A_1, B_1$	$A_0, B_0$	$A > B$	$A < B$	$A = B$	$A > B$	$A < B$	$A = B$
$A_3 > B_3$	X	X	X	X	X	X	H	L	L
$A_3 < B_3$	X	X	X	X	X	X	L	H	L
$A_3 = B_3$	$A_2 > B_2$	X	X	X	X	X	H	L	L
$A_3 = B_3$	$A_2 < B_2$	X	X	X	X	X	L	H	L
$A_3 = B_3$	$A_2 = B_2$	$A_1 > B_1$	X	X	X	X	H	L	L
$A_3 = B_3$	$A_2 = B_2$	$A_1 < B_1$	X	X	X	X	L	H	L
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 > B_0$	X	X	X	H	L	L
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 < B_0$	X	X	X	L	H	L
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 = B_0$	H	L	L	H	L	L
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 = B_0$	L	H	L	L	H	L
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 = B_0$	L	L	H	L	L	H
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 = B_0$	X	X	H	L	L	H
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 = B_0$	H	H	L	L	L	L
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 = B_0$	L	L	L	H	H	L

Tabla 3.15. Tabla de verdad del comparador 7485.



#### Actividad en el aula

##### 7 Ensayo y experimentación con un comparador de ocho bits

En la Figura 3.11 se muestra el símbolo y el diagrama de conexión del comparador 7485.

- Montar dos comparadores de 4 bits en serie, tal como se muestra en la Figura 3.12.
- Comprobar los valores de las salidas  $A = B$ ,  $A > B$  y  $A < B$  del conjunto, para varios valores de las palabras A y B.

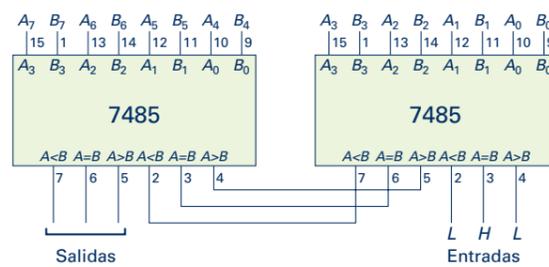


Fig. 3.12. Comparador de dos palabras de ocho bits con dispositivos 7485 en serie.

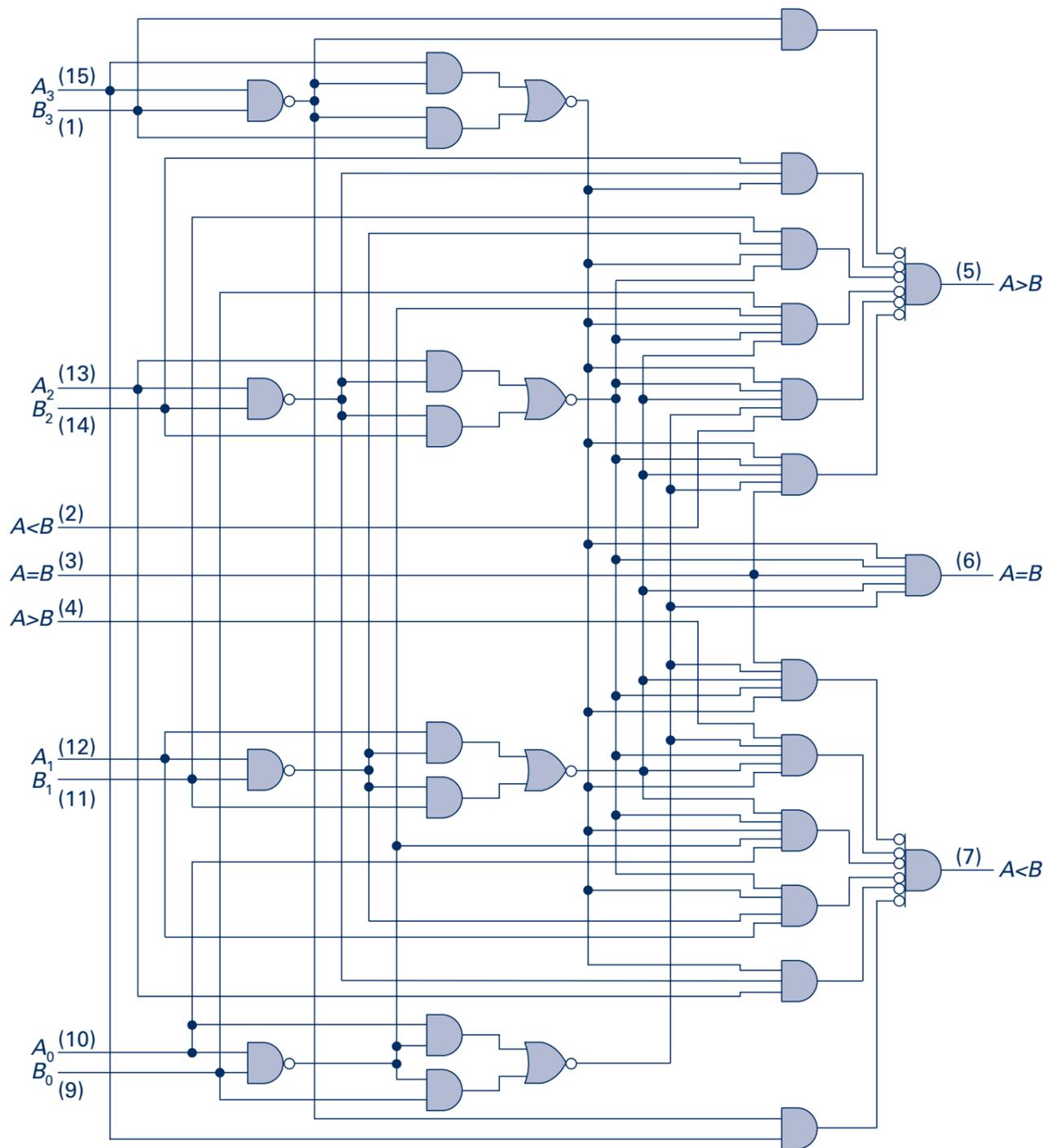


Fig. 3.13. Diagrama lógico del comparador 7485.



### 3. Análisis de circuitos combinacionales MSI

#### 3.7 Análisis de comparadores

#### C. Estudio de la conexión paralelo

Para remediar el inconveniente de retraso en la comparación, se utiliza la técnica de conexión paralelo, más compleja que la conexión serie. En la Figura 3.14 se muestra un circuito formado por seis dispositivos, capaz de comparar palabras de hasta 24 bits con un tiempo de respuesta similar al del circuito serie de la Figura 3.12.

En la conexión paralelo, las entradas de expansión  $A > B$  y  $A < B$  de gran parte de los dispositivos se utilizan como entradas de datos; de esta manera, el componente se convierte en un comparador de cinco bits. Las entradas  $A = B$  se deben conectar a nivel bajo. Como en todos los casos, las tres entradas de expansión del dispositivo correspondiente a los cuatro bits menos significativos de la palabra deben conectarse como ya se ha señalado, es decir,  $A = B$  a nivel alto y  $A > B$  y  $A < B$  a nivel bajo.

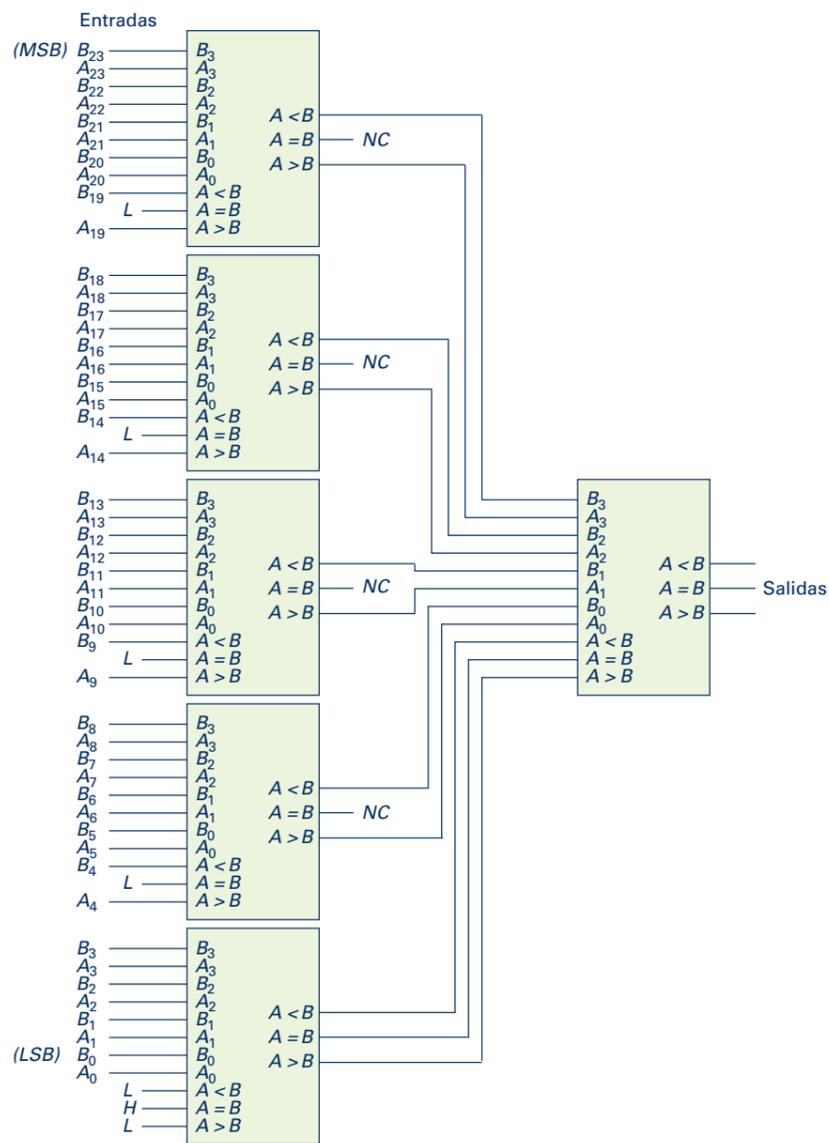


Fig. 3.14. Comparador de dos palabras de 24 bits con seis circuitos 7485 en conexión paralela.



## Actividades finales

1 Pasar de decimal a binario los siguientes números:

- a) 36,75
- b) 12,525
- c) 102,125
- d) 50,225

2 Calcular los números decimales equivalentes a los siguientes números binarios:

- a) 110011
- b) 10011110
- c) 001000,0101
- d) 11110000,1110

3 Codificar en BCD natural, exceso tres y Aiken las siguientes cantidades:

- a) 4 725
- b) 839
- c) 10 637
- d) 235

4 Construir un código de paridad par a partir del BCD Aiken.

5 Formar el código Hamming a partir del BCD exceso tres.

6 Expresar en ASCII:

- |       |        |
|-------|--------|
| a) B  | e) DEL |
| b) 7  | f) *   |
| c) h  | g) :   |
| d) LF |        |

7 Obtener las funciones de la Tabla 3.13 (Tabla de verdad del 7442).

8 **Ensayo y experimentación con un generador de paridad par para palabras de 8 bits.**

Montar un circuito como el de la Figura 3.15 y obtener los valores de salida (BP) para un determinado número de combinaciones de las variables de entrada.

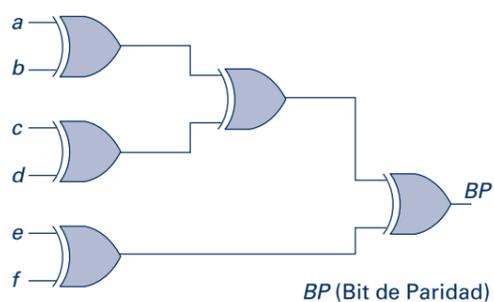


Fig. 3.15. Diagrama de un generador de paridad par para una palabra de 6 bits.

9 **Ensayo y experimentación con un generador/detector de paridad para palabras de 7 bits (incluido el de paridad).**

Montar un circuito detector como el de la Figura 3.16 y acoplar el generador de la Figura 3.15, simulando un fallo en la transmisión (mediante un conjunto de conmutadores). Observar el valor de la salida D.

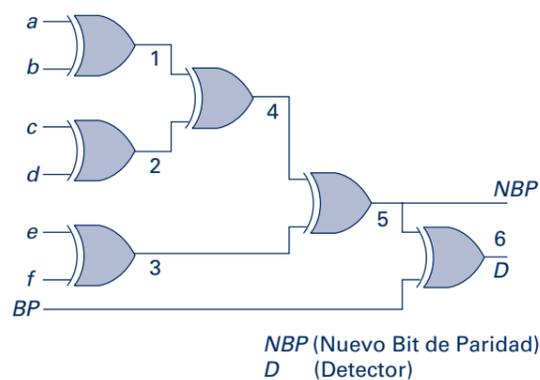


Fig. 3.16. Diagrama de un detector de paridad para palabras de 7 bits (incluido el bit de paridad, BP).

10 Diseñar un circuito, formado por puertas lógicas, que permita convertir el código Aiken a BCD natural. Montar el circuito y comprobar su funcionamiento.