UNIVERSIDAD DE GUADALAJARA

CUCEI

CARRERA: **INGENERIA EN COMPUTACION**

INVESTIGACION 2: **ALGEBRA DE BOOLE Y CIRCUITOS LÓGICOS COMBINACIONALES**

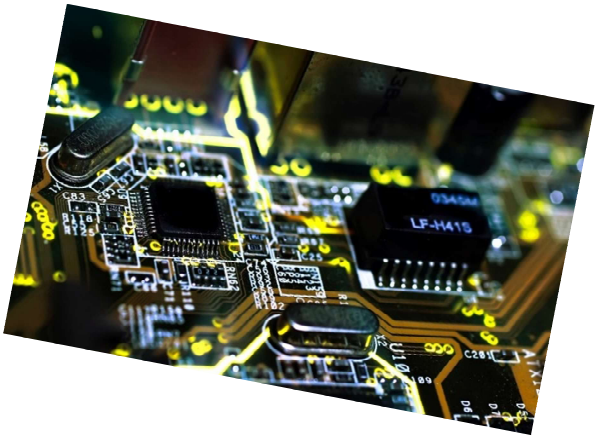
ALUMNO: **EFRAIN ROBLES PULIDO**

CODIGO: **221350095**

NOMBRE DE LA MATERIA: **ARQUITECTURA DE COMPUTADORAS**

SECCIÓN: **D10** *CALENDARIO:* **2021B**

NOMBRE DE LA PROFESORA: **THELMA ISABEL MORALES RAMIREZ**



Algebra de Boole

George Boole (1815-1864), matemático inglés autodidacta, que fue el primero en definirla como parte de un sistema lógico, inicialmente en un pequeño folleto: The Mathematical Analysis of Logic, publicado en 1847, en respuesta a una controversia en curso entre Augustus De Morgan y Sir William Hamilton. El álgebra de Boole fue un intento de utilizar las técnicas algebraicas para tratar expresiones de la lógica proposicional.

En la actualidad, el álgebra de Boole se aplica para el diseño electrónico. Esta lógica se puede aplicar a dos campos:

1. Al análisis, porque es una forma concreta de describir cómo funcionan los circuitos.
2. Al diseño, ya que teniendo una función lógica aplicamos dicha álgebra para poder desarrollar una implementación de la función.

Como toda álgebra, la de Boole parte de un cuerpo axiomático, el cual puede adquirir diversas formas, variando la cantidad y calidad de los axiomas.

El álgebra de Boole son las matemáticas de los sistemas digitales, que está formada por un conjunto de variables Booleanas, (0, 1).

operador + --> operador OR

operador · --> operador AND

operador ‘ --> operador NOT

Una ***variable*** es un símbolo (normalmente una letra mayúscula en cursiva) utilizado para representar magnitudes lógicas como 0 o 1. El ***complemento*** es el inverso de la variable y se indica mediante una barra encima de la misma (complemento de A es $\bar{A}$, A=1 entonces $\bar{A}$=0, o si A=0 entonces $\bar{A}$=1), en algunas ocasiones se utiliza un apostrofe en vez de una barra encima. Y un ***literal*** es una variable o el complemento de una variable.

Axiomas

1. En un conjunto G de objetos, sujetos a una relación de equivalencia, denotada por "=" satisface el principio de sustitución. Es decir, a = b, “b” sustituye “a” en cualquier expresión que la contenga, sin alterar la validez de la expresión.

2. La regla de combinación "+" en donde a + b está en G siempre que al menos a o b lo estén. También se define una regla de combinación "·" en donde a · b está en G siempre que tanto a como b lo estén.
3. Neutros.

Existe un elemento 0 en G tal que para cada a de G: a + 0 = a. También un elemento 1 en G tal que para cada a de G: a · 1 = a.

4. Conmutativos

Para todo par de elementos a y b pertenecientes a G se cumple:

$a + b = b + a$ $a \cdot b = b \cdot a$

5. Distributivos

Para toda terna de elementos a, b, c pertenecientes a G se cumple:

$a + (b \cdot c) = (a + b) \cdot (a + c)$ $a \cdot (b + c) = (a \cdot b) + (a \cdot c)$

6. Complemento

Para cada elemento a de G existe un elemento ā tal que:

$a \cdot \bar{a} = 0$ $a + \bar{a} = 1$

7. Existen por lo menos dos elementos x, y en G tal que x <> y

Muchos de estos postulados tienen una similitud con los del álgebra común. Sin embargo, la primera de las reglas distributivas (sobre la suma) y la existencia del complemento se diferencian en forma fundamental al álgebra común.

Teoremas

- **Teorema 1:** el elemento complemento Ā es único.
- **Teorema 2 (Elementos nulos):** para cada elemento de G se verifica:
  $A+1 = 1$ $A \cdot 0 = 0$
- **Teorema 3:** cada elemento identidad es el complemento del otro.
  $\bar{0}=1$ $\bar{1}=0$
- **Teorema 4 (Idempotencia):** para cada elemento de G, se verifica:
  $A+A=A$ $A \cdot A=A$
- **Teorema 5 (Involución):** para cada elemento de G, se verifica:

$\overline{(\bar{A})} = A$

- **Teorema 6 (Absorción):** para cada par de elementos de G, se verifica:
  $A+A\cdot B=A$ $A\cdot(A+B)=A$
- **Teorema 7:** para cada par de elementos de G, se verifica:
  $A + \bar{A} \cdot B = A + B$ $A \cdot (\bar{A} + B) = A \cdot B$
- **Teorema 8 (asociatividad):** cada uno de los operadores binarios (+) y (·) cumple la propiedad asociativa:
  $A+(B+C) = (A+B) + C$ $A\cdot(B\cdot C) = (A \cdot B) \cdot C$
- **Leyes De Morgan:** para cada par de elementos de G, se verifica:
  $\overline{(A + B)} = \bar{A} \cdot \bar{B}$ $\overline{(A \cdot B)} = \bar{A} + \bar{B}$

Propiedades

- **Dualidad**

Como se presentan de a pares y en tal forma que uno de la pareja se obtiene de otro cambiando "0" por "1" junto con "+" por "·" (y viceversa). Entonces cada propiedad que se demuestre en esta algebra tiene una "dual" que también es cierta (para demostrar la dual bastaría con repetir la demostración realizada sustituyendo cada postulado o propiedad utilizada por su dual).

- **Asociativa**

$a + (b + c) = (a + b) + c$ $a \cdot (b \cdot c) = (a \cdot b) \cdot c$

- **Idempotencia**

Para todo elemento en G se cumple:

$a + a = a$ $a \cdot a = a$

Demostración:

$a + a = (a + a) \cdot 1$ (3b)
$a + a = (a + a) \cdot (a + \bar{a})$ (6)
$a + a = a + (a \cdot \bar{a})$ (5a)
$a + a = a + 0$ (6)
$\Rightarrow a + a = a$

$\Rightarrow a \cdot a = a$ (Dualidad)

- **Neutros Cruzados**

Para todo elemento en G se cumple:

a + 1 = 1 a · 0 = 0

Demostración:

$a+1=a+(a+\bar{a})$ (6)
$a+1=(a+a)+\bar{a}$ (asociativa)
$a+1=a+\bar{a}$ (idempotencia)
$\Rightarrow a+1=1$

$\Rightarrow a\cdot 0=0$ (Dualidad)

- **Complemento de complemento**

Para cada elemento de G se cumple: a = $\bar{\bar{a}}$. Para todo par de elementos de G se cumple:

a + ab = a a (a+b) =a

Para todo par de elementos de G se cumple:

a + $\bar{a}$b = a + b a ($\bar{a}$ + b) = ab

- **Ley De Morgan**

Para todo par de elementos de G se cumple:

$\overline{(a+b)}=\bar{a}\bar{b}$ $\overline{(ab)}=\bar{a}+\bar{b}$

Estas reglas de De Morgan pueden generarse para cualquier número de variables.

$\overline{(a_1+a_2+\cdots a_n)}=\overline{a_1}\,\overline{a_2}\ldots\overline{a_n}$ $\overline{(a_1\,a_2\ldots a_n)}=\overline{a_1}+\overline{a_2}+\cdots\overline{a_n}$

Circuitos Lógicos Combinacionales

Los circuitos combinacionales son circuitos digitales con varias entradas y varias salidas, en los cuales la relación entre cada salida y las entradas puede ser expresada mediante una función lógica (expresiones algebraicas, tablas de verdad, circuito con puertas lógicas, etc.). Entonces, cada salida en un instante de tiempo determinado depende exclusivamente de las

entradas al circuito en el mismo instante de tiempo, pero no depende de las entradas que hubo en instantes de tiempo anteriores (no tiene "memoria").

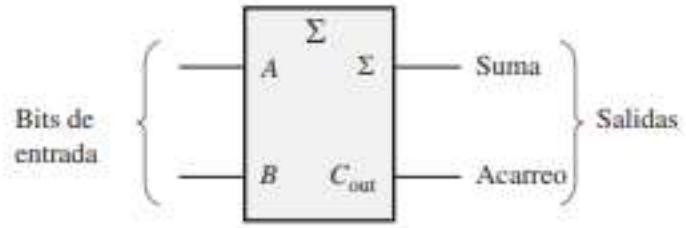
Las *multifunciones* son aquellas funciones que tienen varias salidas, por lo que habrá una expresión lógica para cada salida.

Entonces los circuitos combinacionales realizan funciones específicas, por dos razones:

1. Los circuitos muy complejos pueden descomponerse en circuitos o bloques más elementales, como los que interconectan entre sí para formar el circuito (“Divide y vencerás” o diseño jerárquico).
2. Estos circuitos se encuentran disponibles comercialmente, integrados en una sola pastilla.

Circuitos semi-sumadores

Suma dos dígitos binarios en sus entradas y genera dos dígitos binarios en sus salidas: un bit de suma y un bit de acarreo.



A partir del funcionamiento lógico de un semi-sumador, las expresiones correspondientes a la suma y al acarreo de salida se pueden obtener como funciones de las entradas. En la salida de acarreo (Cout) es 1 sólo cuando A y B son 1; por tanto, Cout puede expresarse como una operación AND de las variables de entrada.

$0 + 0 = 0$
$0 + 1 = 1$
$1 + 0 = 1$
$1 + 1 = 10$

$$C_{out} = AB$$

La salida correspondiente a la suma (Σ) es 1 sólo si las variables A y B son distintas. Por tanto, la suma puede expresarse como una operación OR-exclusiva de las variables de entrada

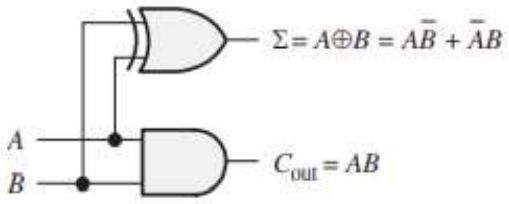
$$\Sigma = A \oplus B$$

La salida de acarreo se produce mediante una puerta AND, siendo A y B sus dos entradas, y la salida de la suma se obtiene mediante una puerta OR-exclusiva.

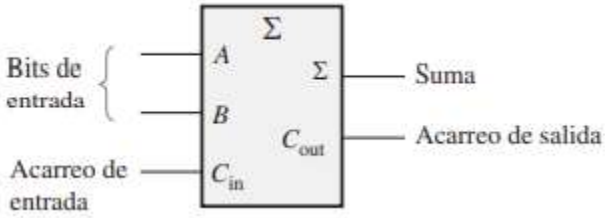
La operación OR-exclusiva se implementa con puertas AND, una puerta OR e inversores.

A	B	C_{OUT}	Σ
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Σ = suma
C_{out} = acarreo de salida
A y B = variables de entrada (operandos)



Circuitos sumadores completo



Un sumador acepta dos bits de entrada y un acarreo de entrada, y genera una salida de suma y un acarreo de salida. Y tiene un acarreo de entrada mientras que el semi-sumador no.

La diferencia principal entre un sumador completo y un semi-sumador es que el sumador completo acepta un acarreo de entrada.

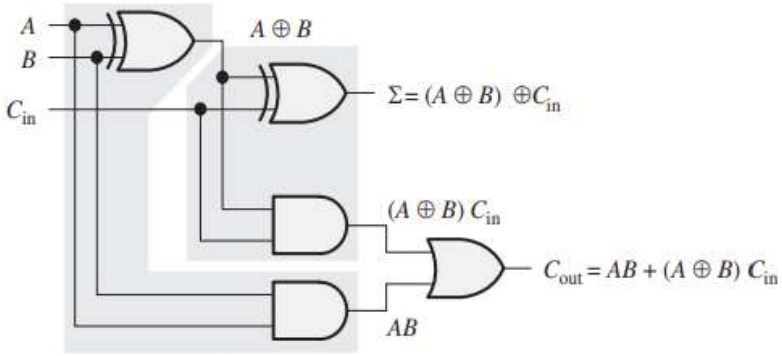
Entonces el sumador completo tiene que sumar dos bits de entrada y un acarreo de entrada. Del semi-sumador sabemos que la suma de los bits de entrada A y B es la operación OR-exclusiva de esas dos variables, A ⊕ B. Para sumar el acarreo de entrada (Cin) a los bits de entrada, hay que aplicar de nuevo la operación OR-exclusiva, obteniéndose la siguiente ecuación para la salida de suma del sumador completo:

$$\Sigma = (A \oplus B) \oplus C_{in}$$

Esto significa que para implementar la función del sumador completo se pueden utilizar dos puertas OR-exclusiva de 2 entradas. La primera tiene que generar el término A ⊕ B, y la segunda tiene como entradas la salida de la primera puerta XOR y el acarreo de entrada.

A	B	C_{in}	C_{OUT}	Σ
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

C_{in} = acarreo de entrada. Algunas veces se designa como *CI*.
C_{out} = acarreo de salida. Algunas veces se designa como *CO*.
Σ = suma
A y B = variables de entrada (operandos)

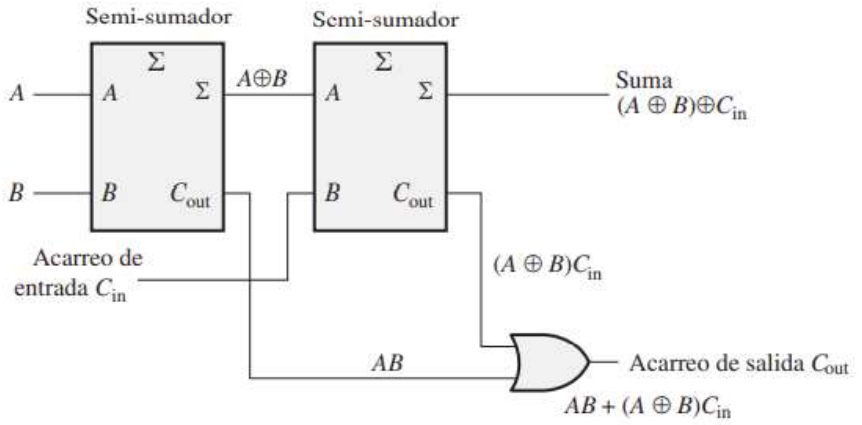


(b) Circuito lógico de un sumador completo (cada semi-sumador se representa por un área sombreada)

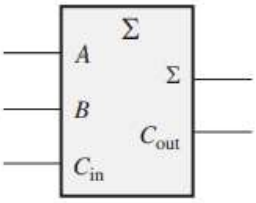
El acarreo de salida es 1 cuando las dos entradas de la primera puerta XOR están a 1 o cuando las dos entradas de la segunda puerta XOR están a 1. El acarreo de salida del sumador completo se obtiene por tanto del producto lógico (AND) de las entradas A y B, y del producto lógico de A ⊕ B y C in. Después se aplica la operación OR a estos dos términos. Esta función se implementa y se combina con la lógica de la suma para formar un circuito sumador completo.

$$C_{out} = AB + (A \oplus B)\, C_{in}$$

Existen dos semi-sumadores conectados, cuyos acarreos de salida se aplican a una puerta OR.



(a) Dos semisumadores para formar un sumador completo



(b) Símbolo lógico del sumador completo

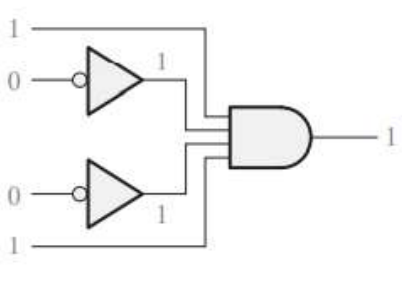
Circuitos decodificadores

Detectan la presencia de una determinada combinación de bits (código) en sus entradas y señalan la presencia de este código mediante un cierto nivel de salida. En su forma general, un decodificador posee n líneas de entrada para gestionar n bits y en una de las 2n líneas de salida indica la presencia de una o más combinaciones de n bits.

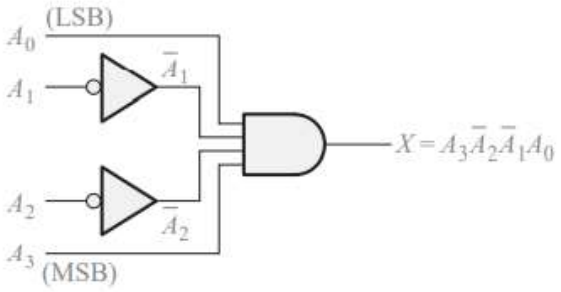
Por ejemplo:

- *El decodificador binario básico*

Supongamos que necesitamos determinar cuándo aparece el número binario 1001 en las entradas de un circuito digital. Se puede utilizar una puerta AND como elemento básico de decodificación, ya que produce una salida a nivel ALTO sólo cuando todas sus entradas están a nivel ALTO. Por tanto, debe asegurarse de que todas las entradas de la puerta AND estén a nivel ALTO cuando se introduce el número 1001, lo cual se puede conseguir invirtiendo los dos bits centrales (cuyos bits son 0).



(a)

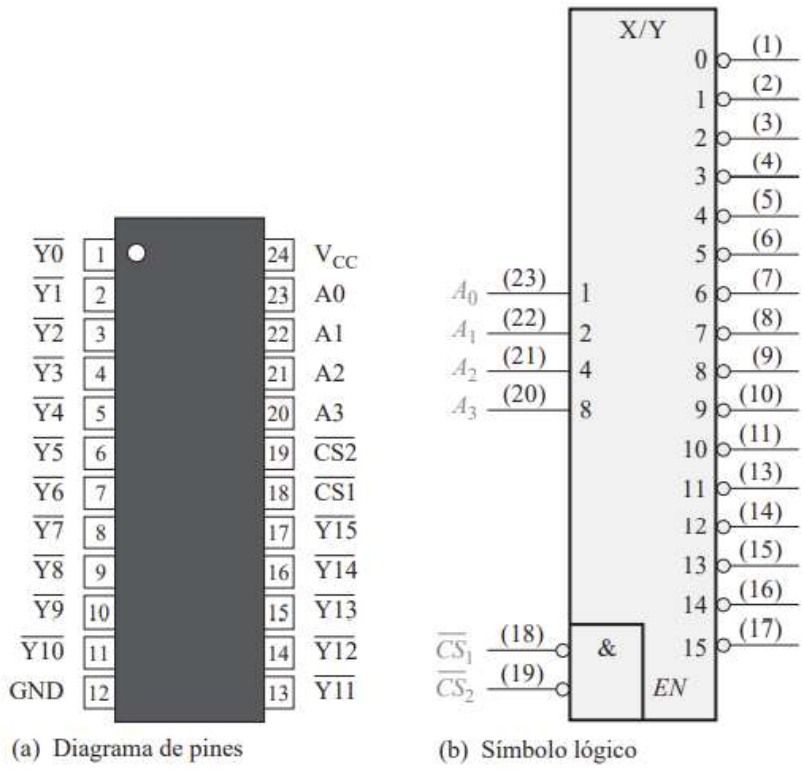


(b)

Se debe comprobar que la salida es siempre 0 excepto cuando se aplican las entradas A0 = 1, A1 = 0, A2 = 0 y A3 = 1. A0 es el bit menos significativo y A3 el más significativo. En este libro, cuando se representa un número binario o cualquier otro código de pesos, el bit menos significativo siempre es el situado más a la derecha cuando el número se escribe en sentido horizontal, y el de más arriba cuando se escribe en vertical, a menos que se indique lo contrario. Si se utiliza una puerta NAND en lugar de una AND una salida a nivel BAJO indicará la presencia del código binario adecuado, que en este caso es 1001.

- *El decodificador 1 de 16 74HC154*

En este tipo de dispositivo existe una función de activación (enable, EN), que se implementa mediante una puerta NOR utilizada como negativa-AND. En las entradas de selección del chip, se requiere un nivel BAJO para obtener en la salida de la puerta de activación (EN, enable) un nivel ALTO. La salida de la puerta de activación se conecta a una entrada de cada puerta NAND del decodificador, por lo que debe estar a nivel ALTO para que las puertas NAND se activen. Si la puerta de activación no se activa mediante un nivel BAJO en ambas entradas, entonces las dieciséis salidas (Y) del decodificador estarán a nivel ALTO independientemente del estado de las cuatro variables de entrada A0, A1, A2 y A3.



(a) Diagrama de pines (b) Símbolo lógico

Aplicaciones:

Las computadoras se tienen que comunicar con una gran variedad de dispositivos externos, denominados periféricos, enviando y/o recibiendo datos a través de lo que se conoce como puertos de entrada/salida (E/S). Estos dispositivos externos incluyen impresoras, modems, escáneres, unidades de disco externas, teclados, monitores y otras computadoras. A

continuación, se emplea un decodificador para seleccionar el puerto de E/S determinado por la computadora, de forma que los datos puedan ser enviados o recibidos desde algún dispositivo externo concreto.

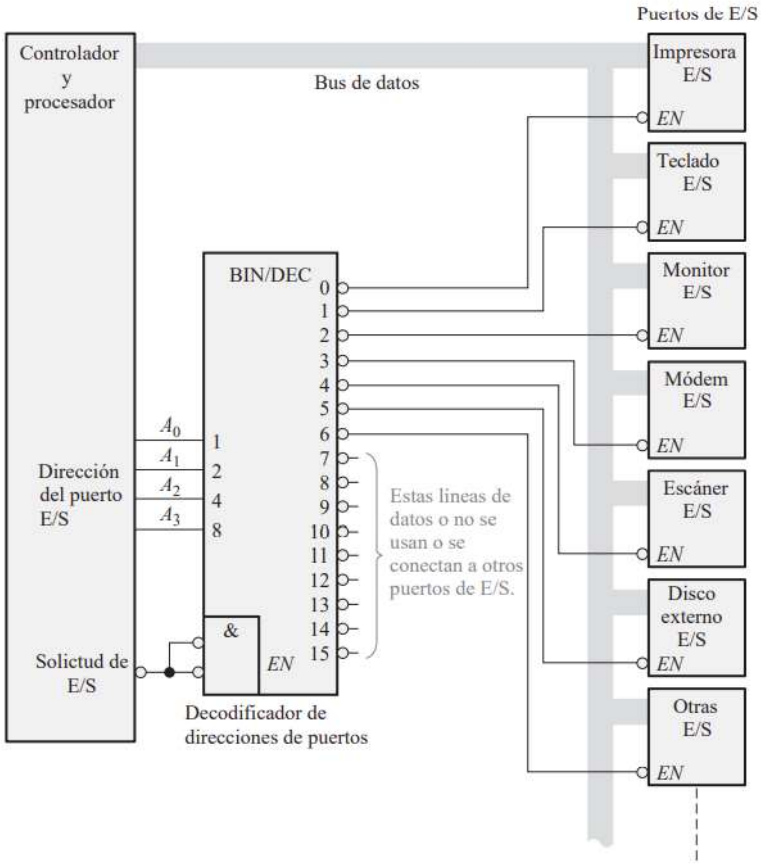


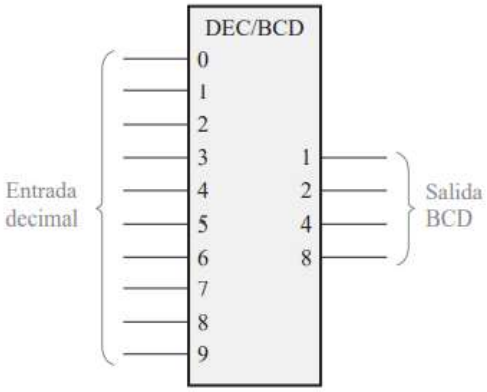
FIGURA 6.31 Un sistema simplificado de puertos de E/S con un decodificador de direcciones de puerto con sólo cuatro líneas de dirección.

Circuitos codificadores

Realizan la función "inversa" del decodificador. Un codificador permite que se introduzca en una de sus entradas un nivel activo que representa un dígito, como puede ser un dígito decimal u octal, y lo convierte en una salida codificada, como BCD o binario. Los codificadores se pueden diseñar también para codificar símbolos diversos y caracteres alfabéticos. El proceso de conversión de símbolos comunes o números a un formato codificado recibe el nombre de codificación.

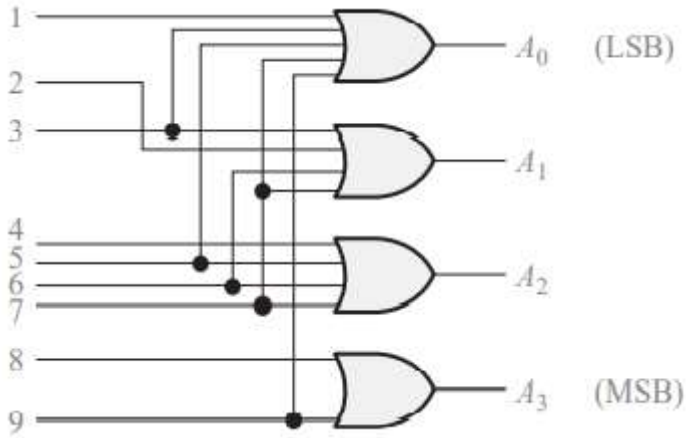
Por ejemplo:

- *Codificador decimal-BCD*



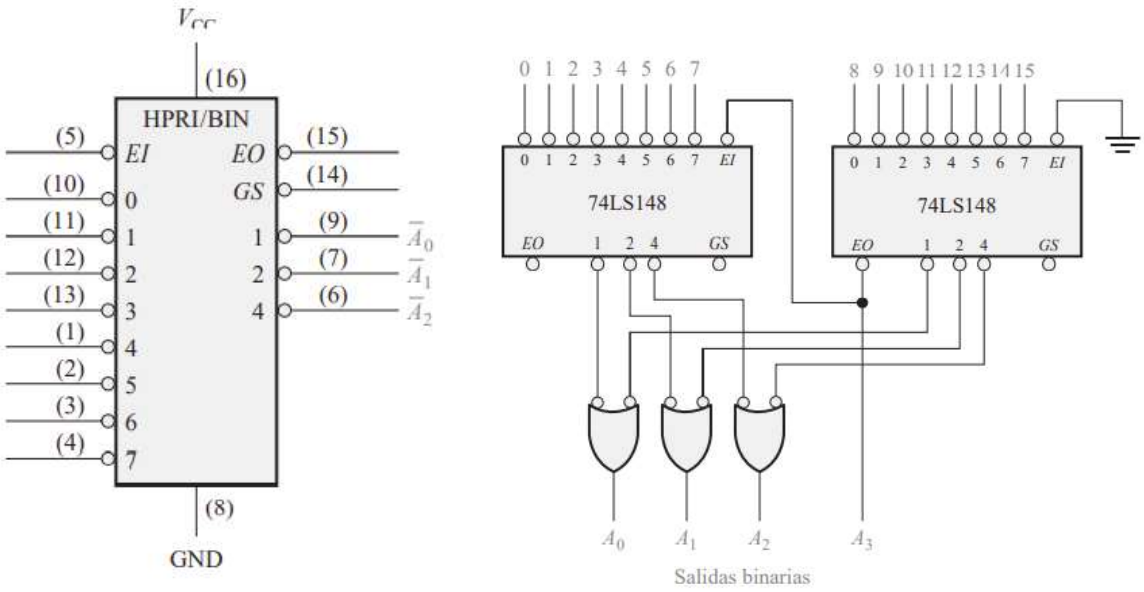
Este tipo de codificador tiene diez entradas, una para cada dígito decimal, y cuatro salidas que corresponden al código BCD. Este es un codificador básico de 10-líneas a 4-líneas. A partir de esta tabla podemos determinar la relación entre cada bit BCD y los dígitos decimales, con el fin de analizar la lógica.

Dígito decimal	Código BCD			
	A_3	A_2	A_1	A_0
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1



- *El codificador 8-líneas a 3-líneas 74ls148*

Es un codificador con prioridad que tiene ocho entradas activas a nivel BAJO y tres salidas binarias activas a nivel BAJO. Este dispositivo se puede utilizar para convertir entradas octales (recuerde que los dígitos octales son del 0 al 7) en código binario de 3 bits. Para activar este dispositivo, la entrada de activación, (Enable Input, EI) tiene que estar activa a nivel BAJO. También tiene una EO (salida de activación, Enable Output) y una salida GS para permitir la ampliación. La salida EO está a nivel BAJO cuando la entrada EI está a nivel BAJO y ninguna de las entradas (de 0 a 7) se encuentra activada. GS está a nivel BAJO cuando EI está a nivel BAJO y cualquiera de las entradas se encuentra activada.



El 74LS148 puede ser ampliado a un codificador de 16-líneas a 4-líneas conectando la salida EO del codificador de mayor orden a la entrada EI del codificador de menor orden, y aplicando la operación negativa-OR a las correspondientes salidas binarias, como se muestra en la Figura 6.41. La salida EO se utiliza como cuarto y más significativo bit. Esta configuración particular produce salidas activas a nivel ALTO para los números binarios de cuatro bits.

Aplicación:

El típico ejemplo de aplicación es un codificador de teclado. Por ejemplo, los diez dígitos decimales del teclado de una computadora tienen que codificarse para poder ser procesados por el circuito lógico. Cuando se pulsa una de las teclas, el dígito decimal se codifica a su correspondiente código BCD.

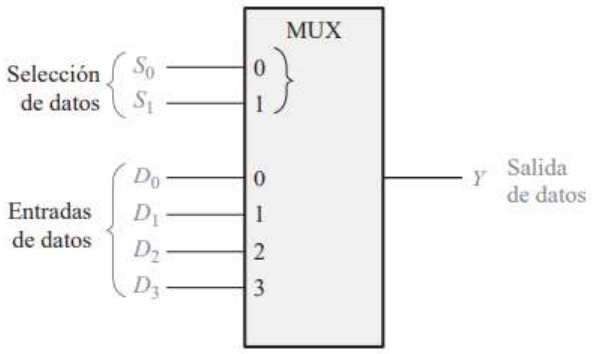
Circuitos multiplexores

Es un circuito lógico que pasa los datos digitales procedentes de varias líneas de entrada a una única línea de salida según una secuencia de tiempos específica. Funcionalmente, un multiplexor puede representarse mediante una operación de conmutación electrónica que conecta secuencialmente cada una de las líneas de entrada a la línea de salida.

También posee entradas de selección de datos, que permiten conmutar los datos digitales provenientes de cualquier entrada hacia la línea de salida. A los multiplexores también se les conoce como selectores de datos.

Por ejemplo:

- *El multiplexor (MUX) de cuatro entradas y una salida*

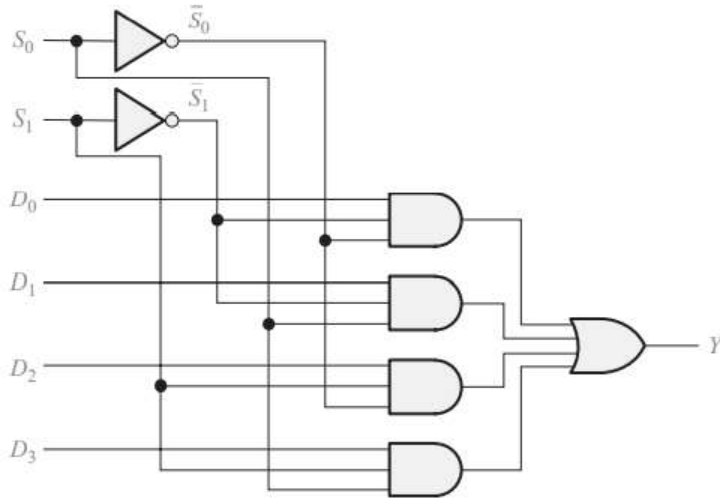


Dispone de dos líneas de selección de datos, dado que con dos bits se puede seleccionar cualquiera de las cuatro líneas de entrada de datos.

Un código binario de dos bits en las entradas de selección de datos (S) va a permitir que los datos de la entrada seleccionada pasen a la salida de datos. Si aplicamos un 0 binario (S1 = 0 y S0 = 0) a las líneas de selección de datos, los datos de la entrada D0 aparecerán en la línea de datos de salida. Si aplicamos un 1 binario (S1 = 0 y S0 = 1), los datos de la entrada D1 aparecerán en la salida de datos. Si se aplica un 2 binario (S1 = 1 y S0 = 0), obtendremos en la salida los datos de D2. Si aplicamos un 3 binario (S1 = 1 y S0 = 1), los datos de D3 serán conmutados a la línea de salida.

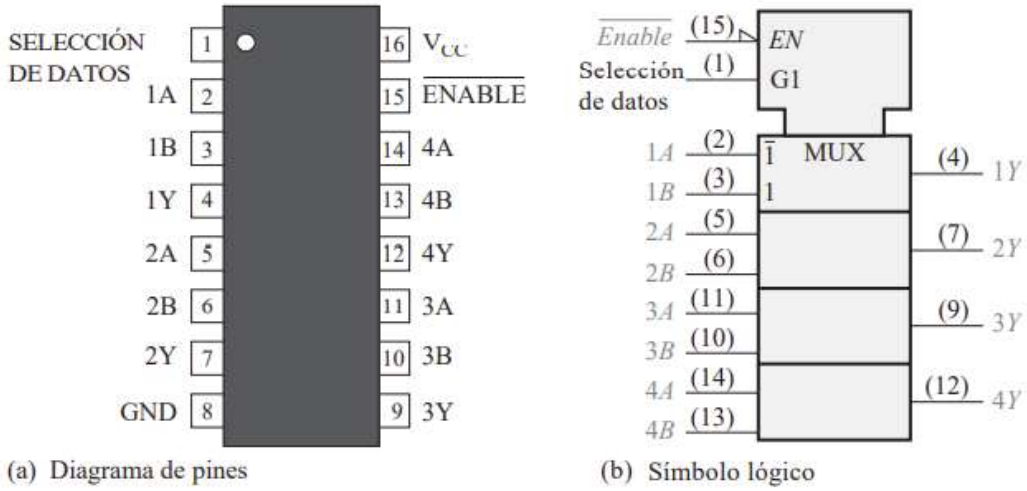
Entradas de selección de datos		Entrada seleccionada
S_1	S_0	
0	0	D_0
0	1	D_1
1	0	D_2
1	1	D_3

La implementación de esta ecuación requiere cuatro puertas AND de tres entradas, una puerta OR de cuatro entradas y dos inversores para generar los complementos de S1 y S0. Dado que los datos pueden ser seleccionados desde cualquier línea de entrada, se conoce también a este circuito con el nombre de selector de datos.



- *El cuádruple multiplexor/selector de datos de 2 entradas 74hc157*

Está formado por cuatro multiplexores de dos entradas. Cada uno de los cuatro multiplexores comparten una misma línea de selección de datos y una de habilitación (enable). Ya que sólo existen dos entradas de datos que puedan ser seleccionadas en cada multiplexor, es suficiente con tener una única entrada de selección. Un nivel BAJO en la entrada de habilitación permite al dato de entrada seleccionado pasar a la salida. Un nivel ALTO en la entrada evita que los datos pasen a la salida, es decir, inhabilita los multiplexores. Cuando la entrada de selección está a nivel ALTO, se seleccionan las entradas B de los multiplexores y, cuando la entrada de selección está a nivel BAJO, se seleccionan las entradas A. Para indicar dependencia AND siempre se usa una "G".



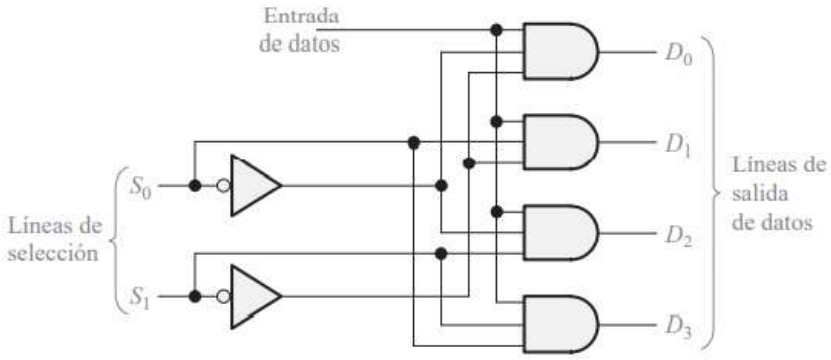
Circuitos demultiplexores

Es un circuito que pasa los datos digitales procedentes de una línea de entrada a varias líneas de salida según una determinada secuencia de tiempo. En esencia, el demultiplexor es un multiplexor invertido.

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. Como veremos, los decodificadores pueden utilizarse también como demultiplexores.

Por ejemplo:

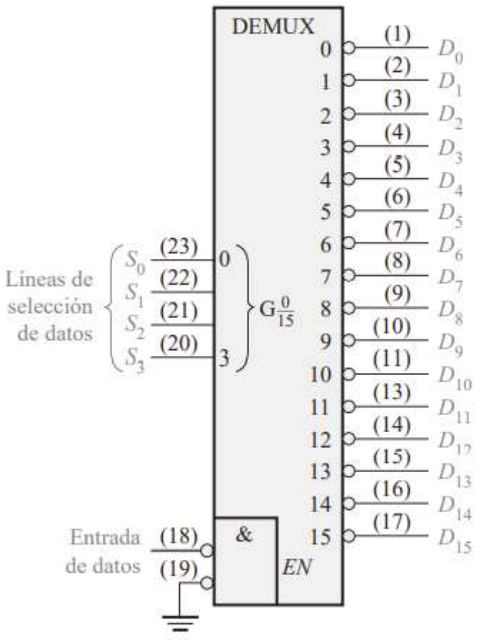
- *El circuito demultiplexor (DEMUX) de 1-línea a 4- líneas.*



La línea de entrada de datos está conectada a todas las puertas AND. Las dos líneas de selección de datos activan únicamente una puerta cada vez y los datos que aparecen en la línea de entrada de datos pasarán a través de la puerta seleccionada hasta la línea de salida de datos asociada.

- *El demultiplexor 74hc154*

Cuando se utiliza con este fin, se usan las líneas de entrada como líneas de selección de datos, una de las entradas de activación del chip se usa como línea de entrada de datos y la otra se mantiene a nivel BAJO, para activar la puerta interna negativa-AND.

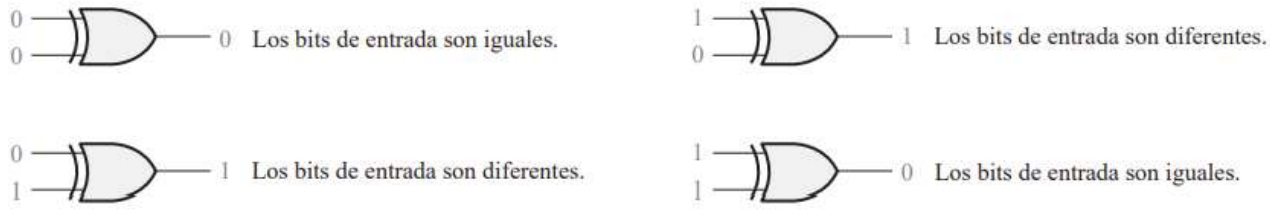


Circuitos comparadores

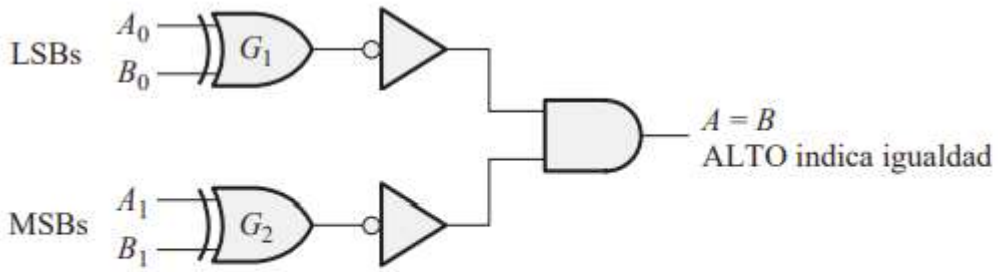
Consiste en comparar las magnitudes de dos cantidades binarias para determinar su relación. En su forma más sencilla, un circuito comparador determina si dos números son igual.

Igualdad

La puerta OR-exclusiva se puede emplear como un comparador básico, ya que su salida es 1 si sus dos bits de entrada son diferentes y 0 si son iguales.



Para comparar números binarios de dos bits, se necesita una puerta OR-exclusiva adicional. Los dos bits menos significativos (LSB) de ambos números se comparan mediante la puerta G1 y los dos más significativos (MSB) son comparados mediante la puerta G2. Si los dos números son iguales, sus correspondientes bits también lo son, y la salida de cada puerta OR-exclusiva será 0. Si los correspondientes conjuntos de bits no son idénticos, la salida de la puerta OR-exclusiva será un 1.

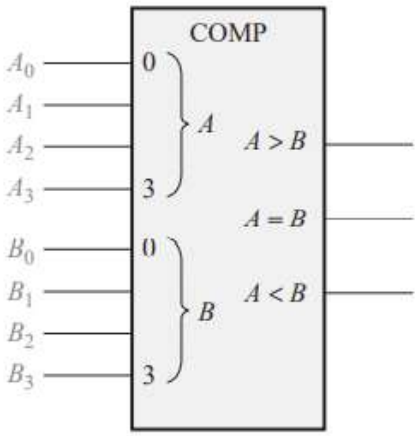


Formato general: Número binario $A \rightarrow A_1A_0$
Número binario $B \rightarrow B_1B_0$

Para obtener un único resultado de salida que indique la igualdad o desigualdad entre los dos números, se pueden usar dos inversores y una puerta AND. La salida de cada puerta OR-exclusiva se invierte y se aplica a la entrada de la puerta AND. Cuando los bits de entrada de cada OR-exclusiva son iguales, lo que quiere decir que los bits de ambos números son iguales, las entradas de la puerta AND son 1, por lo que el resultado a su salida también será 1. Cuando los

dos números no son iguales, al menos uno o ambos conjuntos de bits será distinto, lo que da lugar a, al menos, un 0 en una de las entradas de la puerta AND, y el resultado a su salida será 0. Por tanto, la salida de la puerta AND indica la igualdad (1) o desigualdad (0) entre dos números.

Desigualdad



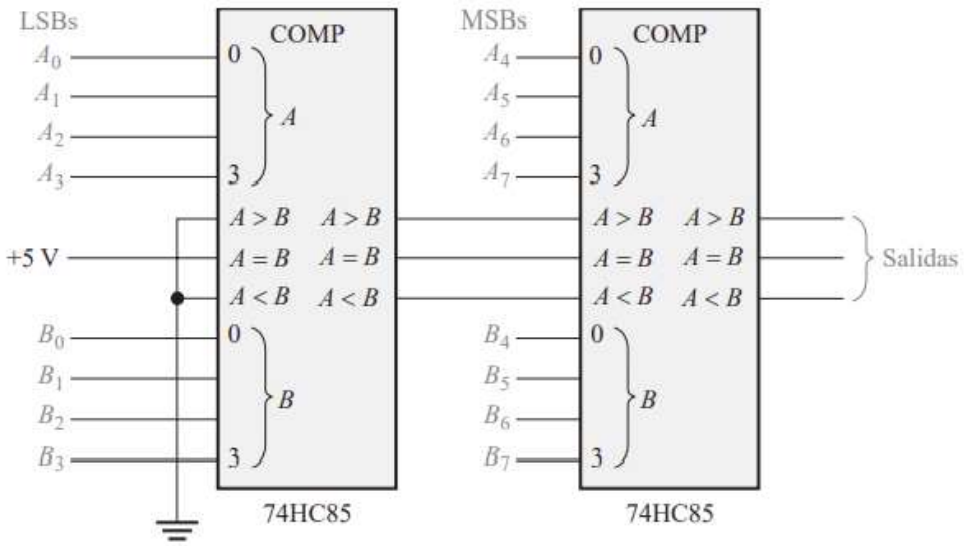
Muchos circuitos integrados comparadores tienen salidas adicionales que indican cuál de los dos números que se comparan es el mayor. Esto significa que existe una salida que indica cuándo el número A es mayor que el número B (A > B) y otra salida que indica cuándo A es menor que B (A < B).

El procedimiento general utilizado en un comparador consiste en comprobar una desigualdad en cualquier posición de bit, comenzando por los bits más significativos (MSB). Cuando se encuentra una desigualdad, la relación entre ambos números queda establecida y cualquier otra desigualdad entre bits con posiciones de orden menor debe ignorarse, ya que podrían indicar una relación entre los números completamente opuesta. La relación de más alto orden es la que tiene prioridad.

Por ejemplo:

- *El comparador de magnitud de 4 bits 74hc85*

Este dispositivo tiene todas las entradas y salidas del comparador visto anteriormente y, además, tiene tres entradas en cascada: A < B, A = B y A > B. Estas entradas permiten utilizar varios comparadores en cascada para la comparación de cualquier número binario con más de cuatro bits. Para ampliar el comparador, las salidas A < B, A = B y A > B del comparador de menor orden se conectan en cascada a las entradas del siguiente comparador de orden inmediatamente superior. El comparador de menor orden debe tener un nivel ALTO en la entrada A = B y un nivel BAJO en las entradas A < B y A > B.



Bibliografía

Alberto Brunete, P. (2021). 3.1 Álgebra de Boole | Introducción a la Automatización Industrial. Consultado el 27 de agosto de 2021 en https://bookdown.org/alberto_brunete/intro_automatica/algebraboole.html

Arquitectura de Computadoras. (2021). Algebra de Boole. Consultado el 27 de agosto de 2021 en https://www.fing.edu.uy/tecnoinf/mvd/cursos/arqcomp/material/teo/arq-teo03.pdf

Fundamentos de las Computadoras. (2021). TEMA 3. Álgebra de Boole. Consultado el 27 de agosto de 2021 en http://www.uhu.es/rafael.lopezahumada/descargas/tema3_fund_0405.pdf

Floyd, T.L. (2007) Fundamentos de sistemas Digitales. (9ª Edición). Madrid: Pearson EducaCon.