

Unidad 2 / Escenario 3

Lectura Fundamental

Circuitos lógicos combinacionales I

Contenido

- 1 Puertas lógicas compuestas
- 2 Circuitos lógicos combinacionales
- 3 Circuitos decodificadores
- 4 Circuitos codificadores
- 5 Circuitos conversores

Referencias

1. Compuertas lógicas compuestas

Mediante la combinación de las operaciones lógicas básicas AND, OR y NOT, es posible obtener otro tipo de expresiones más complejas, representadas en compuertas lógicas. Las compuertas NAND y NOR son dos compuertas compuestas de gran importancia, debido a que su fabricación es de menor complejidad que la de las compuertas AND y OR (Mano, 2014). Otras compuertas compuestas, de gran importancia en el diseño de circuitos digitales son las XOR y XNOR, principalmente para labores de comparación y suma de los valores de entrada.

1.1. Compuerta NAND

Se trata de una compuerta AND que tiene su salida negada (el término NAND proviene de la unión entre NOT y AND). En la figura 1 se presenta el símbolo de una compuerta NAND (la salida negada se representa por un pequeño círculo, para simplificación). En el cuadro 1 se encuentra la relación entre entradas y salidas para la compuerta, al igual que los niveles lógicos obtenidos como resultado de la operación mediante compuertas lógicas. Básicamente, **la compuerta NAND genera un valor ALTO cuando *alguna* de sus entradas está en nivel BAJO, y un nivel BAJO cuando *todas* sus entradas están en nivel ALTO.**

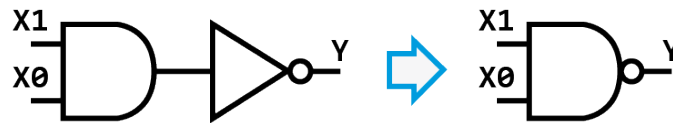
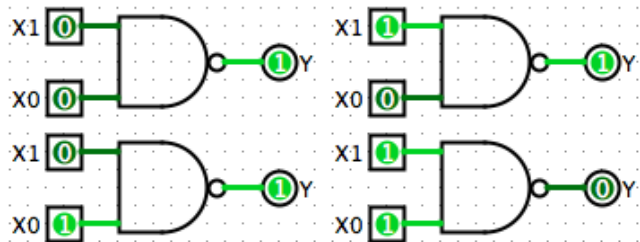


Figura 1: Símbolo de la compuerta NAND. *Elaboración propia.*

Entradas		AND	NAND
X1	X0	$X1 \cdot X0$	$Y = \overline{X1 \cdot X0}$
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	0

(a) Tabla de verdad



(b) Representación de la lógica en el circuito

Cuadro 1: Funcionamiento de la compuerta NAND. *Elaboración propia.*

1.1.1. Equivalente Negativa-OR

La compuerta NAND puede representarse como una compuerta OR que tiene todas sus entradas negadas (Figura 2). Esta representación se obtiene al aplicar el teorema de DeMorgan: $Y = \overline{X1 \cdot X0} = \overline{X1} + \overline{X0}$. Dado este equivalente, una compuerta NAND puede ser usada como una compuerta OR de lógica inversa, es decir, la salida se pone en ALTO cuando cualquiera de las entradas está en nivel BAJO.



Figura 2: Equivalencia entre una NAND y una Negativa-OR. *Elaboración propia.*

1.1.2. NAND como elemento lógico universal

Mediante la combinación de compuertas NAND es posible implementar cualquier otro tipo de compuerta lógica, por lo cuál se puede decir que se trata de una compuerta universal (Floyd, 2010). En la figura 3 se encuentran los circuitos para las compuertas NOT, AND y OR implementados usando únicamente compuertas NAND.

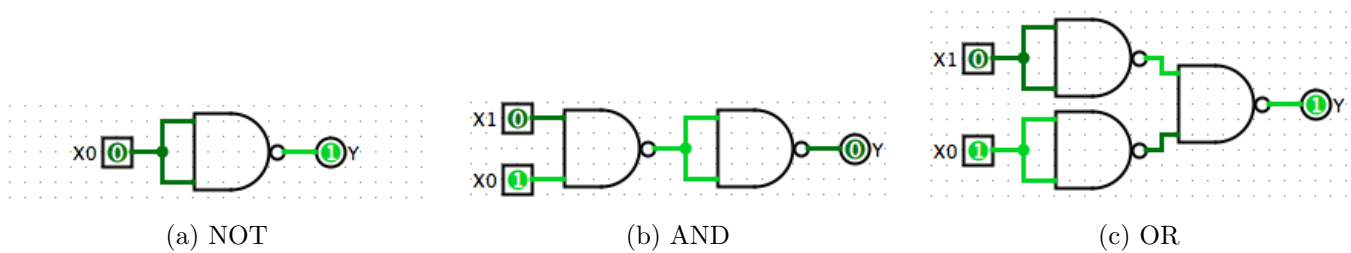


Figura 3: NAND como compuerta universal. *Basado en (Floyd, 2010).*

1.2. Compuerta NOR

Se trata de una compuerta OR que tiene su salida negada (el término NOR proviene de la unión entre NOT y OR). En la figura 4 se presenta el símbolo de una compuerta NOR. En el cuadro 2 se encuentra la relación entre entradas y salidas para la compuerta, al igual que los niveles lógicos obtenidos como resultado de la operación mediante compuertas lógicas. **La compuerta NOR genera un valor ALTO cuando *todas* sus entradas están en nivel BAJO, y un nivel BAJO cuando *alguna* de sus entradas están en nivel ALTO.**

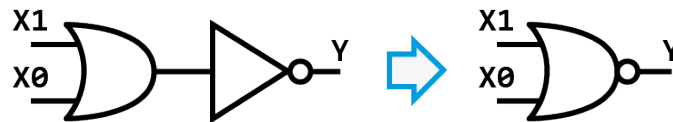


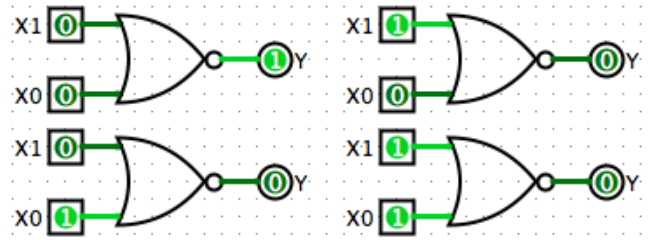
Figura 4: Símbolo de la compuerta NOR. *Elaboración propia.*

1.2.1. Equivalente Negativa-AND

La compuerta NOR puede representarse como una compuerta AND que tiene todas sus entradas negadas (Figura 5). Esta representación se obtiene al aplicar el teorema de DeMorgan: $Y = \overline{X1 + X0} = \overline{X1} \cdot \overline{X0}$. Dado este equivalente, una compuerta NOR puede ser usada como una compuerta AND de lógica inversa, es decir, la salida se pone en ALTO sólo cuando las diferentes entradas está en nivel BAJO.

Entradas		OR	NOR
X1	X0	$X1 + X0$	$Y = \overline{X1 + X0}$
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0

(a) Tabla de verdad



(b) Representación de la lógica en el circuito

Cuadro 2: Funcionamiento de la compuerta NOR. *Elaboración propia.*



Figura 5: Equivalencia entre una NOR y una Negativa-AND. *Elaboración propia.*

1.2.2. NOR como elemento lógico universal

Al igual que con la compuerta NAND, es posible obtener cualquiera de las operaciones booleanas utilizando únicamente compuertas NOR. En la figura 6 se encuentran los circuitos para las compuertas NOT, AND y OR.

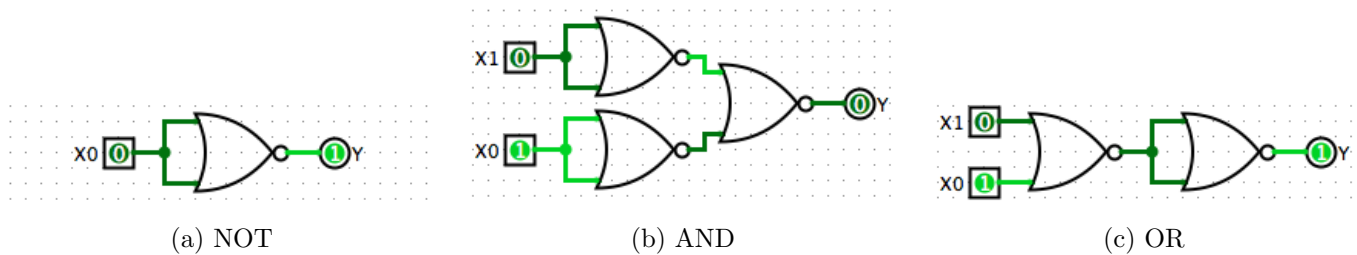


Figura 6: NOR como compuerta universal. *Basado en (Floyd, 2010).*

1.3. Compuerta XOR

La compuerta XOR, también conocida como *OR exclusivo*, es un tipo de circuito utilizado frecuentemente, formado por la combinación de compuertas lógicas. La expresión de salida de una XOR está dada por la siguiente relación entre sus entradas:

$$Y = \overline{X1} \cdot X0 + X1 \cdot \overline{X0}.$$

Esta ecuación suele abreviarse con el símbolo \oplus , por lo que la expresión final sería:

$$Y = X1 \oplus X0.$$

En la figura 7 se encuentra el circuito completo, basado en la expresión anterior, y el símbolo asignado para la compuerta XOR.

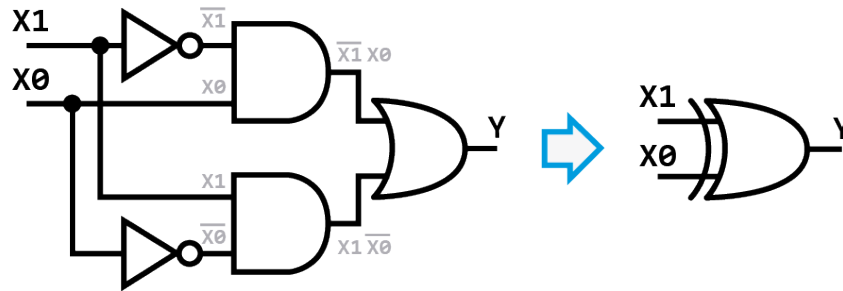
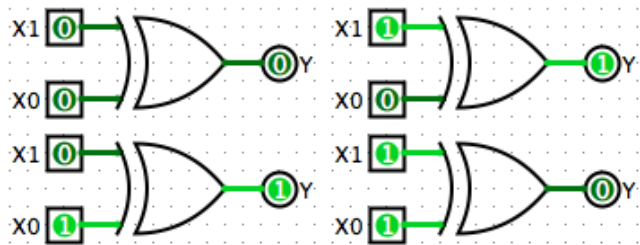


Figura 7: Circuito XOR y su símbolo. *Elaboración propia.*

Dada la expresión anterior y el circuito resultante, es posible obtener la tabla de verdad de la compuerta XOR, así como la verificación de su funcionamiento (Cuadro 3). Nótese que **en la compuerta XOR se obtiene un nivel ALTO en la salida, únicamente cuando sus dos entradas tienen niveles diferentes. Si ambas entradas están en un mismo nivel, la salida es BAJO.**

Entradas		XOR
X1	X0	$X1 \oplus X0$
0	0	0
0	1	1
1	0	1
1	1	0

(a) Tabla de verdad



(b) Representación de la lógica en el circuito

Cuadro 3: Funcionamiento de la compuerta XOR. *Elaboración propia.*

1.4. Compuerta XNOR

La compuerta XNOR, también conocida como *NOR exclusivo*, es otra compuerta compuesta bastante usada con un funcionamiento opuesto a la XOR. La expresión de salida de una XNOR está dada por la siguiente relación entre sus entradas:

$$Y = X1 \cdot X0 + \overline{X1} \cdot \overline{X0}.$$

Esta ecuación suele abreviarse con el símbolo \otimes , por lo que la expresión final sería:

$$Y = X1 \otimes X0.$$

En la figura 8 se encuentra el circuito completo, basado en la expresión anterior, y el símbolo asignado para la compuerta XNOR.

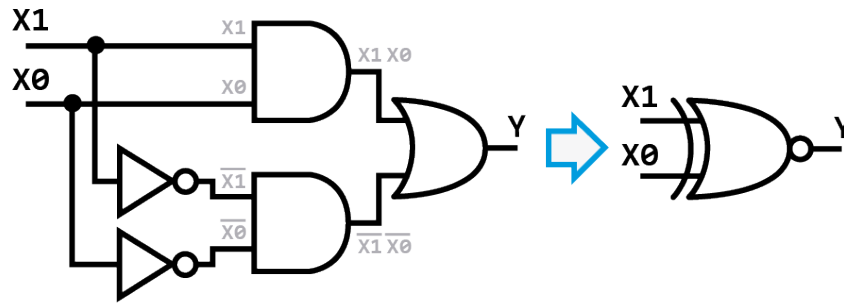
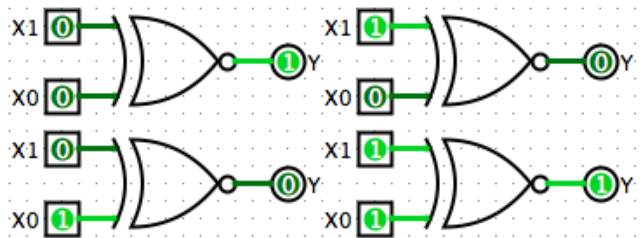


Figura 8: Circuito XNOR y su símbolo. *Elaboración propia.*

Dada la expresión anterior y el circuito resultante, es posible obtener la tabla de verdad de la compuerta XNOR, así como la verificación de su funcionamiento (Cuadro 4). Nótese que a diferencia de la compuerta XOR, **en una compuerta XNOR se obtiene un nivel ALTO en la salida, únicamente cuando sus dos entradas tienen el mismo nivel (BAJO o ALTO).**

Entradas		XNOR
X1	X0	$X1 \otimes X0$
0	0	1
0	1	0
1	0	0
1	1	1

(a) Tabla de verdad



(b) Representación de la lógica en el circuito

Cuadro 4: Funcionamiento de la compuerta XNOR. *Elaboración propia.*

1.5. Circuitos integrados digitales

Las compuertas lógicas vistas hasta ahora, así como muchas de las funciones lógicas que se verán a lo largo del curso, están implementadas mediante circuitos integrados (Figura 9), usualmente construidos bajos dos tipos de tecnología: **TTL** y **CMOS**. El tipo de tecnología afecta la tensión de alimentación, velocidad de respuesta y otras características, pero las funciones lógicas y conexiones son iguales.

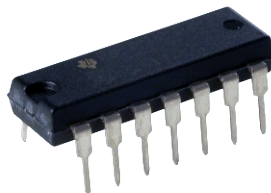


Figura 9: Circuito integrado

La nomenclatura que permite diferenciar entre diferentes tipos de integrados y tecnologías se encuentra estandarizada. Los primeros números corresponden al tipo de serie, normalmente se trabaja con la **74**. Luego de este número

puede o no haber algunas letras, las cuales hacen referencia a la tecnología usada, por ejemplo **74LS** corresponde a TTL, mientras que **74HC** es CMOS. Finalmente se encuentran dos o tres dígitos que hacen referencia al tipo de compuerta o circuito. En el cuadro 5 se encuentran algunos tipos y su descripción. En estas tablas se puede ver que una NAND cuádruple de dos entradas podría tener la denominación 7400, 74LS00 y 74HC00.

Tipo	Descripción
74	TTL estándar
74S	TTL Schottky
74LS	TTL Schottky de baja potencia
74HC	CMOS de alta velocidad
74AC	CMOS avanzada

(a) Nomenclatura según tecnología.

Tipo	Descripción
00	Cuádruple NAND de dos entradas
02	Cuádruple NOR de dos entradas
04	Inversor séxtuple
08	Cuádruple AND de dos entradas
32	Cuádruple OR de dos entradas
86	Cuádruple XOR
266	Cuádruple XNOR

(b) Numeración según tipo de circuito.

Cuadro 5: Nomenclatura de algunos integrados típicos. *Tomado de Floyd (2010)*

Antes de usar un circuito integrado, es necesario revisar las características de conexión del mismo. Toda la información referente a conexiones, características de operación, entre otras, se encuentra en la hoja de datos del integrado (*datasheet* en inglés), el cual puede encontrarse en una búsqueda por internet o en la página web del fabricante (Por ejemplo Texas Instruments).

En este momento lo invito a regresar al escenario, donde encontrará información complementaria acerca de las compuertas compuestas y su implementación mediante circuitos integrados.

2. Circuitos lógicos combinacionales

Los circuitos lógicos trabajados hasta ahora, incluyendo las compuertas lógicas compuestas, pueden clasificarse como circuitos lógicos combinacionales, para los cuales "el nivel lógico de la salida depende de la combinación de los niveles lógicos presentes en las entradas" (Tocci, Widmer, y Moss, 2007). Esto se puede ver en el diagrama general de la figura 10. En este tipo de circuitos no existe ningún almacenamiento de información o memoria, por lo tanto cualquier cambio en las entradas se verá reflejado directamente en la salida. Estos circuitos pueden representarse mediante funciones booleanas y tablas de verdad. Su implementación se realiza utilizando compuertas lógicas.

Existen circuitos combinacionales más complejos, con funciones específicas como codificación/decodificación, conversión y selección de información, así como de tipo aritméticas (sumadores, restadores, multiplicadores), entre otras.

2.1. Diseño de circuitos combinacionales

Si se tiene la expresión booleana o la tabla de verdad para un problema particular, es posible obtener el circuito lógico correspondiente de manera casi directa. Pero ¿Qué se debe hacer cuando se parte de un enunciado o

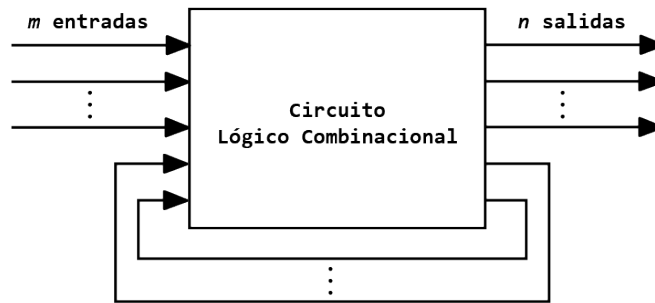


Figura 10: Circuitos lógicos combinacionales. *Elaboración propia.*

problema? A continuación se encuentra un procedimiento de diseño que permite la obtención de un circuito combinacional para un problema específico (Tocci y cols., 2007). Cabe mencionar que en este método se hace uso de los minterminos y la "Suma de Productos". El mismo procedimiento podría realizarse utilizando maxtérminos y "Producto de Sumas", haciendo los cambios correspondientes.

2.1.1. Análisis del problema

Para el diseño de circuitos combinacionales es necesario realizar un estudio del problema, con el fin de determinar, primero que todo, que éste sea solucionable mediante este tipo de circuitos. Esto es así cuando la salida depende únicamente del estado actual de las entradas. Al interpretar el problema, se debe hacer énfasis en identificar cuáles son las entradas y salidas del mismo. A cada una de las variables se les identifica con un nombre.

2.1.2. Obtención de la tabla de verdad

Se debe analizar cómo las variables de entrada modifican las variables de salida, y por ende, el resultado del sistema. Esta relación entre variables se debe expresar mediante una tabla de verdad.

2.1.3. Obtención de minterminos

De acuerdo a la tabla de verdad obtenida, es necesario verificar en qué casos la salida es 1 o "ALTO". De esta manera es posible obtener los minterminos que hacen parte de la solución del problema.

2.1.4. Obtención de la expresión Suma de Productos

Una vez se han detectado todos los minterminos, se unen para obtener la expresión Suma de Productos.

2.1.5. Simplificación de la expresión obtenida

Utilizando los métodos de simplificación vistos con anterioridad, bien sea mediante álgebra booleana o mediante métodos tabulares, como los mapas de Karnaugh.

2.1.6. Implementación del circuito dada la expresión

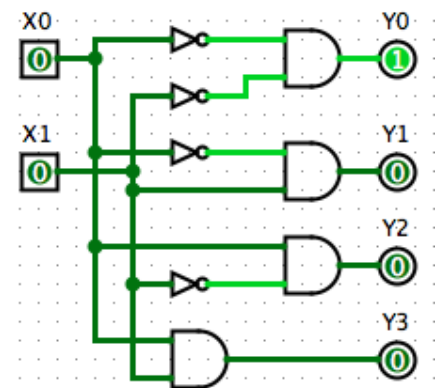
Para finalizar el diseño, se realiza la implementación del circuito según la expresión simplificada. Tenga en cuenta que una buena forma de verificar si un circuito es correcto es utilizando en lo posible en un programa de diseño asistido (*Logisim*). En este momento lo invito a volver al escenario, donde encontrará un recurso con un ejemplo de este paso a paso.

3. Circuitos decodificadores

De acuerdo a Floyd (2010): "La función básica de un decodificador es detectar la presencia de una determinada combinación de bits (código) en sus entradas y señalar la presencia de este código mediante un cierto nivel de salida." Un decodificador, de manera general, permite convertir la información de n líneas de entrada a 2^n líneas de salida. Su funcionamiento está dado por una tabla de verdad, que relaciona entradas con salidas.

Entradas		Salidas				Expresión de salida
X1	X0	Y0	Y1	Y2	Y3	
0	0	1	0	0	0	$Y0 = \overline{X1} \cdot \overline{X0}$
0	1	0	1	0	0	$Y1 = \overline{X1} \cdot X0$
1	0	0	0	1	0	$Y2 = X1 \cdot \overline{X0}$
1	1	0	0	0	1	$Y3 = X1 \cdot X0$

(a) Tabla de verdad



(b) Circuito lógico

Cuadro 6: Decodificador de 2 a 4. *Elaboración propia.*

En el cuadro 6 se muestra un ejemplo de decodificador, de 2 entradas y 4 salidas. Como se puede ver en la tabla de verdad, cada una de las salidas tiene una expresión específica, dada por el estado necesario en las entradas para activarla. Se hace uso de las compuertas AND e inversores, para formar las combinaciones necesarias para cada estado de activación.

Existen diferentes tipos de decodificadores, que siguen la misma estructura de diseño del visto previamente. En el cuadro 7 se pueden ver algunas referencias y sus respectivas descripciones. Además de convertir la información binaria a otro código, los decodificadores pueden ser usados como elementos de control y selección, para activar o desactivar diferentes elementos en un circuito, dado un código específico.

Tipo	Descripción
74HC42	Decodificador BCD a decimal
74LS47	Decodificador BCD a 7 segmentos
74HC154	Decodificador de 4 líneas a 16 líneas
74HC238	Decodificador de 3 líneas a 8 líneas

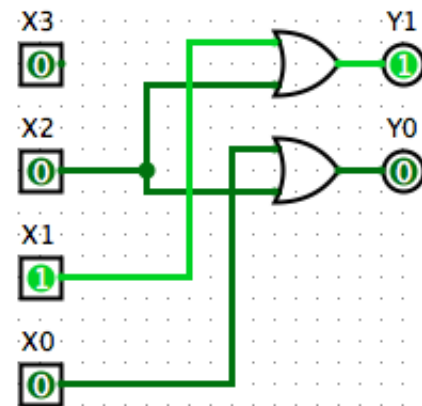
Cuadro 7: Referencias de decodificadores comunes.

4. Circuitos codificadores

Un circuito codificador se utiliza normalmente para hacer la tarea inversa del decodificador. Es decir, cuando se presenta una señal activa en alguna de sus entradas, el decodificador debe generar un código en binario (u otra codificación) en su salida. Existen además codificadores con prioridad, que evalúan si más de una señal en la entrada se encuentra activa y generan el código menor o mayor entre las opciones de entrada. En el cuadro 8 se encuentra el ejemplo de un codificador de 4 líneas a 2. Nótese que en este caso, las salidas se encuentran en BAJO cuando todas las entradas están en BAJO, mientras que cuando se activa alguna de las entradas X0 (1), X1 (2) O X2 (3), el código binario en la salida es acorde (Este codificador funciona con los dígitos decimales del 0 al 3). En el cuadro 9 se encuentran algunas referencias y sus respectivas descripciones.

Entradas				Salidas	
X0	X1	X2	X3	Y1	Y0
0	0	0	0	0	0
1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	0	1	1

(a) Tabla de verdad



(b) Circuito lógico

Cuadro 8: Decodificador de 2 a 4. *Elaboración propia.*

Tipo	Descripción
74HC147	Codificador de decimal a BCD con prioridad
74LS148	Codificador de 8 líneas a 3 líneas

Cuadro 9: Referencias de codificadores comunes.

5. Circuitos conversores

También conocidos como convertidores de código, son circuitos que cambian "los datos que se presentan en cierto tipo de código binario, a otro código binario" (Tocci y cols., 2007, p.624). Este es el caso del decodificador de BCD a 7 segmentos, pero también existen otros circuitos conversores comunes (Cuadro 10).

Tipo	Descripción
74LS47	BCD a 7 segmentos
74LS184	BCD a binario
74LS185A	Binario a BCD

Cuadro 10: Referencias de codificadores comunes.

En el caso de los conversores BCD a binario, normalmente su aplicación se requiere cuando se desea hacer una interfaz entre un sistema de usuario (basado en BCD) y un equipo de cómputo. Esta conversión se hace, debido a que los números en BCD (cuando tienen más de un dígito) pueden ocupar mayor cantidad de Bits que en binario.

Referencias

- Floyd, T. L. (2010). *Fundamentos de sistemas digitales*. Pearson Prentice Hall. (OCLC: 893578510)
- Mano, M. M. (2014). *Diseño digital: con una introducción a Verilog HDL*. Naucalpan de Juárez, México: Pearson Educación. (OCLC: 881629816)
- Tocci, R. J., Widmer, N. S., y Moss, G. L. (2007). *Sistemas digitales: principios y aplicaciones*. México [etc.: Pearson Educación. (OCLC: 804512544)

INFORMACIÓN TÉCNICA



Módulo: Sistemas digitales y ensambladores

Unidad 2: Circuitos combinacionales

Escenario 3: Circuitos lógicos combinacionales I

Autor: Gabriel Eduardo Ávila Buitrago

Asesor Pedagógico: Jeimmy Lorena Romero Perilla

Diseñador Gráfico: Leonardo Stiglitch Campos

Corrector de estilo: Jaime Posada

Asistente: Jhon Edwar Vargas Villa

*Este material pertenece al Politécnico Grancolombiano.
Por ende, son de uso exclusivo de las Instituciones
adscritas a la Red Ilumino. Prohibida su reproducción
total o parcial.*