

Unidad 2 / Escenario 4

Lectura Fundamental

Circuitos lógicos combinacionales II

Contenido

- 1 Multiplexores
- 2 Demultiplexores
- 3 Circuitos aritméticos
- 4 Comparadores de magnitud
- 5 Generadores y comprobadores de paridad

Referencias

Palabras clave: Lógica Combinacional, Multiplexores, Demultiplexores, Circuitos Aritméticos, Paridad

1. Multiplexores

Un multiplexor (MUX) digital, también conocido como selector de datos, “es un circuito lógico que acepta **varias entradas de datos digitales y selecciona una de ellas en un momento dado para pasarla a la salida**” (Tocci, Widmer, y Moss, 2007, p.599). Dicha selección es controlada mediante las líneas o entradas de selección. Siendo M la cantidad de líneas de selección, es posible controlar hasta $N = 2^M$ entradas. En la figura 1 se muestra un diagrama funcional del multiplexor, fíjese que el multiplexor actúa de manera similar que un selector.

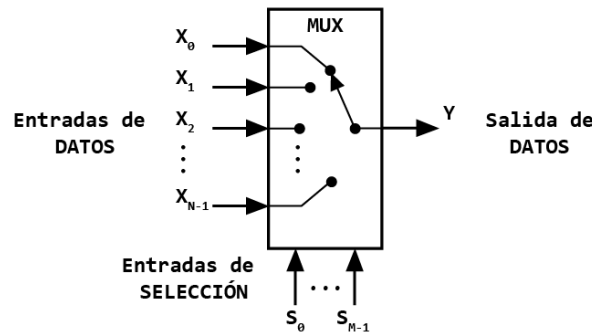


Figura 1: Diagrama funcional de un multiplexor. *Basado en Tocci y cols. (2007)*

En la figura 2 se encuentra la tabla de verdad para un multiplexor con 4 entradas de datos. Note que las líneas de datos y de salida deben ser del mismo tamaño (En este caso son de 1 bit). Se puede apreciar como, de acuerdo al código binario que se ponga en las dos entradas de selección S1 y S0, es posible mostrar una de las entradas X0, X1, X2 o X3 en la salida de datos. En la última columna de la tabla de verdad se encuentran además las expresiones lógicas para cada una de las posibles salidas, representadas en el circuito lógico mostrado (Este circuito se obtuvo utilizando la herramienta de Análisis Combinacional de Logisim, vista en el escenario anterior).

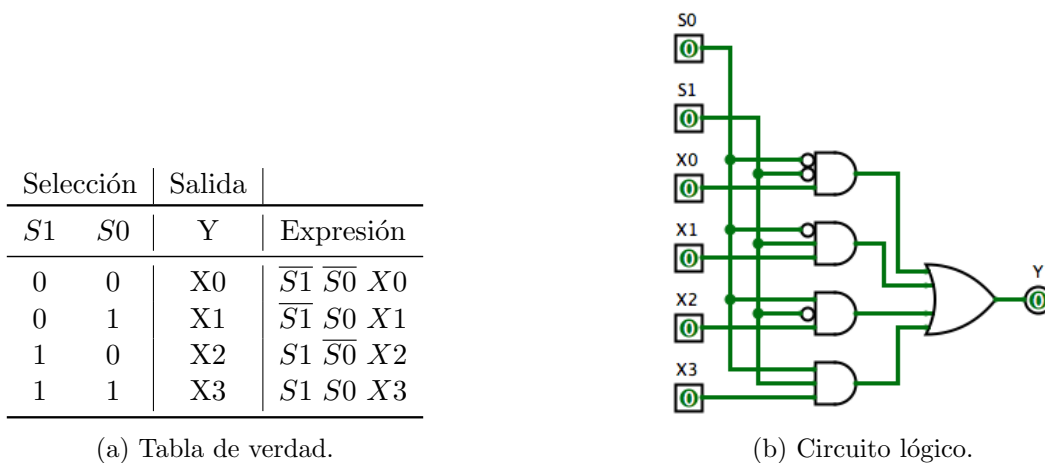


Figura 2: Funcionamiento de un Multiplexor de 4 entradas. *Elaboración propia.*

En la tabla 1 hay algunas referencias de multiplexores comerciales, junto con sus respectivas descripciones. El 74HC151 es un multiplexor con 8 entradas de 1 bit y 3 bits de selección (Figura 3a). Existen también multiplexores con entradas mayores a 1 bit, como el 74HC157 que cuenta con 2 líneas, cada una de 4 bits de entrada y 1 bit

de selección (Figura 3b). Estas referencias de multiplexor, al igual que otros circuitos integrados, cuentan con una entrada de habilitación \overline{EN} (Enable), la cual debe estar en BAJO (Lógica inversa) para que a la salida haya un dato válido.

Tipo	Descripción
74HC151	Multiplexor de 8 entradas (de 1 bit).
74HC157	Multiplexor cuádruple de 2 entradas (de 4 bits).

Tabla 1: Referencias de multiplexores comerciales. *Elaboración propia.*

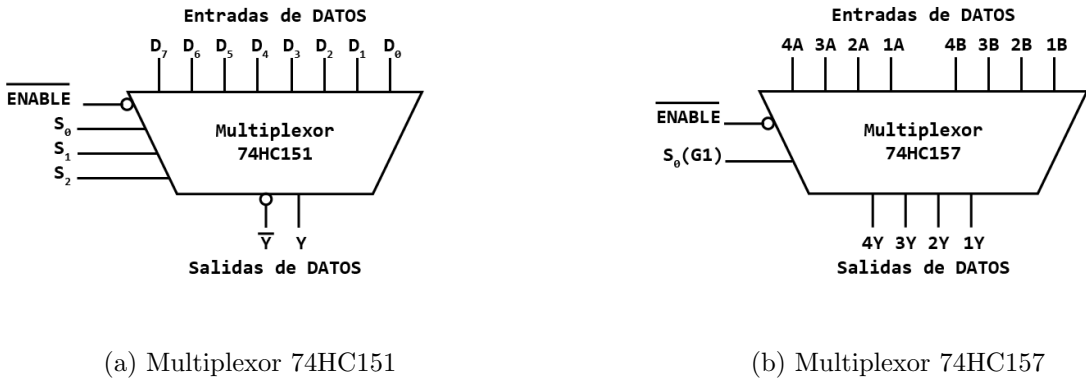


Figura 3: Diagramas funcionales de dos multiplexores digitales. *Elaboración propia.*

1.1. Aplicaciones

Existen variadas aplicaciones para los multiplexores: “la selección de datos, el enrutamiento de datos, la secuencia de operaciones, la conversión de paralelo a serial, la generación de formas de onda y la generación de funciones lógicas” (Tocci y cols., 2007, p.604). A continuación se estudian dos de ellas.

1.1.1. Enrutamiento de datos

Es la aplicación principal de un multiplexor. Se cuenta con varias fuentes de información y los datos obtenidos de esas fuentes se quieren llevar a un único elemento que procese o analice dicha información. Un ejemplo clásico sucede cuando se cuenta con dos displays de 7 segmentos y se desea utilizar un solo conversor de BCD a 7 segmentos (74LS47). Es posible poner en las entradas del multiplexor (74HC157) los dos números a visualizar, y sincronizar la entrada de selección con la alimentación de los displays (mediante un decodificador 74LS139, A1 en BAJO enciende el primer display y en ALTO el segundo). Esto se puede ver en la figura 4.

1.1.2. Generación de funciones lógicas

La implementación de funciones lógicas mediante multiplexores se hace partiendo de la tabla de verdad. El primer paso es la evaluación de los minterminos para la generación de una expresión Suma de Productos. Se seleccionan

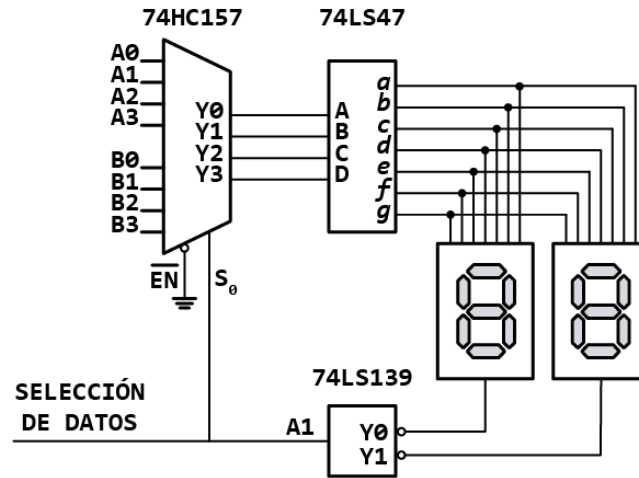


Figura 4: Enrutamiento de datos - Displays de 7 segmentos. *Basado en Floyd (2010).*

las entradas de datos que corresponden con un mintermino y se deja su estado en ALTO, para todas las demás el estado se deja conectado en BAJO. Las variables de entrada de la tabla de verdad se conectan a las entradas de selección del multiplexor.

En la figura 5a se presenta una tabla de verdad para un problema particular. En este caso la Suma de Productos está dada por:

$$Y = m(0, 1, 4, 7) = m_0 + m_1 + m_4 + m_7$$

$$Y = \overline{C} \overline{B} \overline{A} + \overline{C} \overline{B} A + C \overline{B} \overline{A} + C B A$$

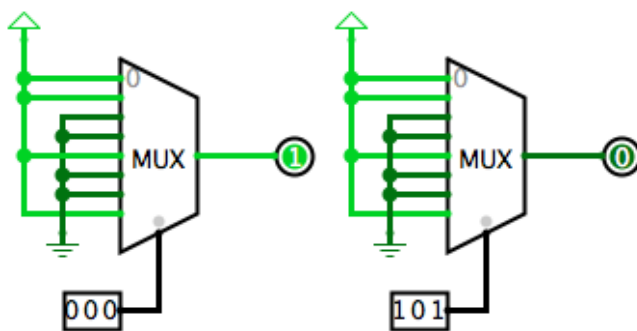
Estos estados corresponden con las entradas D_0, D_1, D_4 y D_7 del multiplexor y se deben poner en ALTO, los demás se deben poner en BAJO. En la figura 5(b) se muestra el circuito implementado con dos ejemplos de valores en la entrada de selección. Note que cuando la entrada de selección se encuentra en 0 0 0, se muestra en la salida el valor de la entrada D_0 (estado $\overline{C} \overline{B} \overline{A}$) que está conectado en ALTO. Por otro lado, cuando la entrada es 1 0 1 ($C \overline{B} A$) la salida es 0.

2. Demultiplexores

Los demultiplexores (DEMUX) realizan la tarea inversa de los multiplexores. Un demultiplexor permite la distribución de datos, desde una única entrada hacia diversas salidas. En la figura 6 se muestra el diagrama funcional de un demultiplexor. Note cómo en este caso, solo hay una línea de entrada para N salidas. La relación entre las M entradas de selección y las N salidas es equivalente a la del multiplexor: $N = 2^M$. Recuerde que, al igual que en los multiplexores, las líneas de entrada y salida de datos pueden contar con más de 1 bit.

Entradas			Salida
C	B	A	Y
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

(a) Tabla de verdad.



(b) Circuito lógico.

Figura 5: Generación de funciones lógicas. *Elaboración propia.*

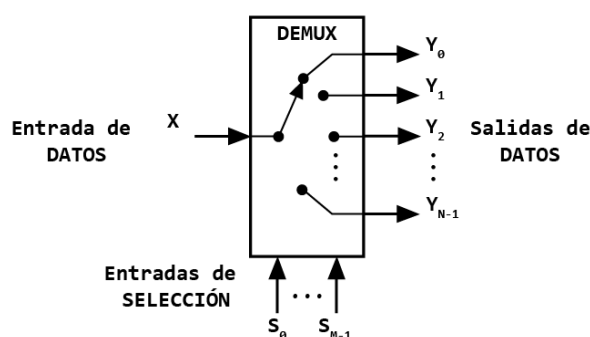


Figura 6: Diagrama funcional de un demultiplexor. *Basado en Tocci y cols. (2007)*

¿Sabías qué...

los circuitos integrados que se utilizan como demultiplexores, normalmente tienen la capacidad de ser usados también como decodificadores?

En la tabla 2 hay algunas referencias de demultiplexores comerciales, junto con sus respectivas descripciones. El 74HC138 es un demultiplexor con 1 entrada de 1 bit y 3 bits de selección, para 8 salidas (Figura 7(a)). El 74HC139 funciona como un demultiplexor doble, de 2 a 4 líneas (Figura 7(b)) y el 74HC154 cuenta con 4 bits de selección, para 16 salidas (Figura 7(c)). En el caso de usar estos circuitos como demultiplexores, lo que se hace normalmente es usar las entradas de selección A (*Address*), para determinar hacia qué salida se direccionan los datos, mientras que los datos en sí se aplican a las entradas de habilitación (*Enable*), dejando una entrada para datos y las otras en un nivel activo.

Tipo	Descripción
74LS138	Decodificador/demultiplexor de 1 a 8 líneas,
74HC139	Doble decodificador/demultiplexor de 2 a 4 líneas.
74HC154	Decodificador/demultiplexor de 4 a 16 líneas.

Tabla 2: Referencias de demultiplexores comerciales. *Elaboración propia.*

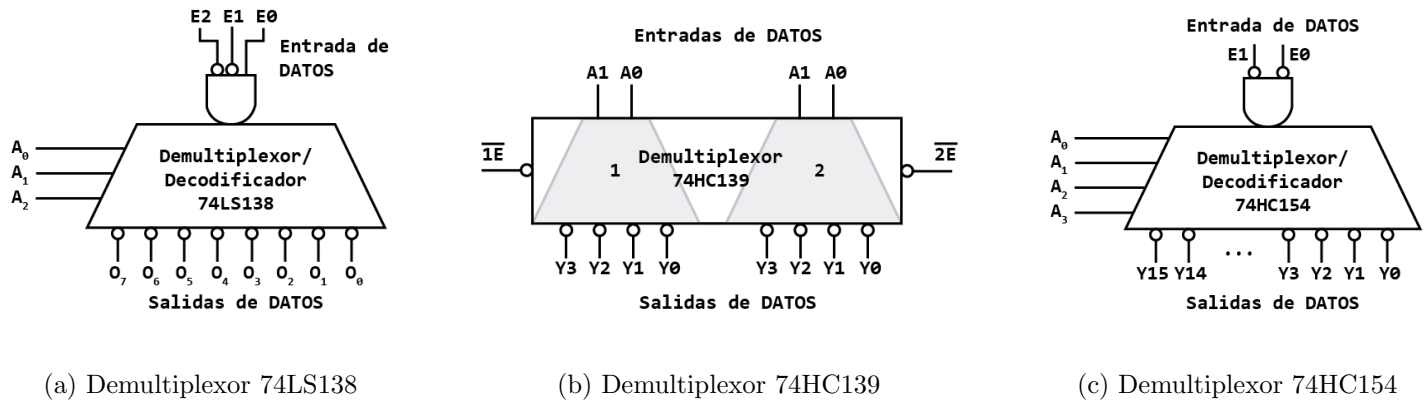


Figura 7: Diagramas funcionales de tres demultiplexores comerciales. *Elaboración propia.*

En este momento es necesario regresar al escenario. Ahí podrá encontrar una lectura que permitirá complementar lo visto hasta ahora con respecto a generación de funciones lógicas con multiplexores, utilizando también decodificadores. Además, podrá realizar una actividad para verificar que lo visto hasta ahora esté claro.

3. Circuitos aritméticos

En esta sección se revisarán algunos circuitos combinacionales que permiten realizar operaciones aritméticas binarias. Inicialmente se estudiará todo lo relacionado con los sumadores, luego se hará una explicación de cómo se puede realizar una resta a partir de un circuito sumador y finalmente, una multiplicación binaria. La importancia de estas operaciones radica en que los sistemas digitales en algunas ocasiones deben realizarlas para el procesamiento de datos numéricos.

3.1. Semisumador

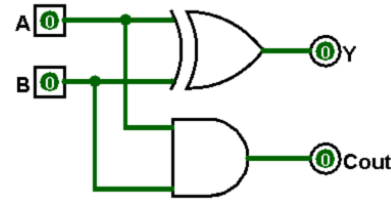
Un Semisumador es un circuito combinacional que cuenta con dos bits de entrada (operandos) y dos bits de salida, uno de suma y otro de acarreo. **Se le llama semisumador, pues no tiene en cuenta si existe o no un bit de acarreo para la entrada.** Para poder comprender el circuito del semisumador, es necesario recordar cómo se realiza la suma binaria. En la figura 8(a) se presenta su tabla de verdad.

La última columna de la tabla de verdad (Y) muestra el resultado de la suma entre A y B , sin tener en cuenta el acarreo. Es posible ver que este resultado es equivalente a una operación XOR entre A y B . La tercera columna (C_{out}) representa el acarreo como resultado de la suma, agregando o no un dígito a la siguiente potencia. Este acarreo de salida corresponde a aplicar la operación AND entre A y B . De acuerdo a lo anterior, podría decirse entonces que:

$$Y = A \oplus B \quad y \quad C_{out} = A B$$

$A + B$			
A	B	C_{out}	Y
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

(a) Tabla de verdad.



(b) Circuito.

Figura 8: Semisumador. Basado en (Mano, 2014).

Dadas las expresiones para Y y C_{out} se obtiene la representación mediante circuitos combinacionales de lo que se conoce como un semisumador (figura 8b).

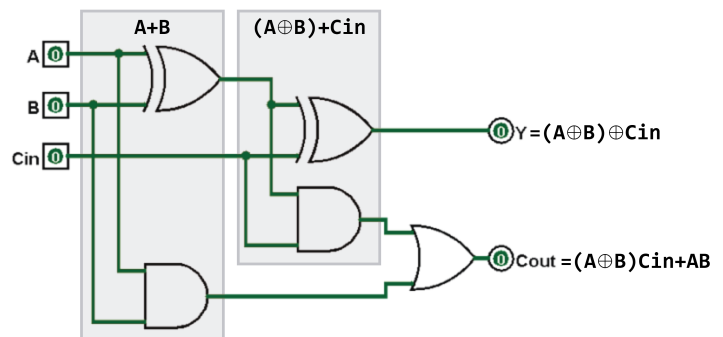
3.2. Sumador completo

Un sumador completo es un circuito que, además de recibir en sus entradas los bits de los operandos A y B , tiene un bit para el acarreo de entrada (C_{in}), que normalmente proviene de otra suma para un dígito de menor valor. Esto permite hacer, como su nombre lo indica, una suma completa entre dos dígitos de un bit. La tabla de verdad para este circuito se encuentra en la figura 9a, para las salidas Y y C_{out} .

Mediante la simplificación de las tablas de verdad por medio de mapas de Karnaugh y álgebra booleana, es posible obtener el circuito de la figura 9b. Este circuito consta de dos semisumadores que realizan las operaciones $A + B$ y $(A \oplus B) + C_{in}$. El bit de acarreo del sumador completo resulta de aplicar la operación OR a los dos bits de acarreo de salida de cada semisumador.

$(A + B) + C_{in}$				
A	B	C_{in}	C_{out}	Y
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

(a) Tabla de verdad.



(b) Circuito.

Figura 9: Sumador completo. Basado en (Mano, 2014).

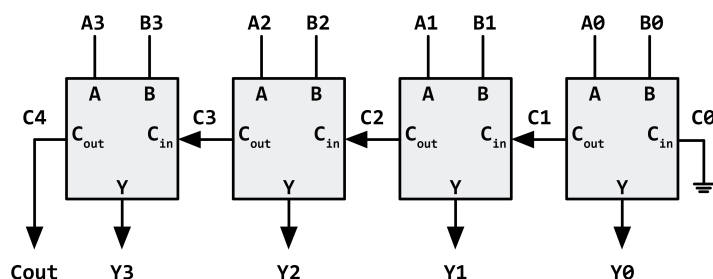
3.3. Sumador binario en paralelo

Utilizando n sumadores completos, es posible realizar la suma de 2 números binarios de n bits. La suma se realiza bit a bit, teniendo en cuenta los acarros de entrada y de salida. Al igual que en la suma binaria vista en la unidad

1, si el resultado de la suma genera un acarreo de salida, el acarreo se ve representado en el siguiente dígito como un acarreo de entrada.

En la figura 10a se presenta el circuito para un sumador de 4 bits. En esta imagen hay varios aspectos a tener en cuenta. Note que cada caja representa un sumador completo, con sus entradas A y B; a la derecha el acarreo de entrada C_{in} y a la izquierda el acarreo de salida C_{out} ; en la parte inferior se encuentra la salida Y. Con respecto a los acarreos, Para el bit menos significativo no hay acarreo de entrada (C_0 se pone en un nivel BAJO). La salida está conformada por los 4 bits resultantes de cada suma individual. El bit más significativo de la salida corresponde al último acarreo de salida (C_4).

Con el fin de comprender mejor el funcionamiento del circuito, la figura 10b muestra el ejemplo de la suma de los números $A = 1101$ y $B = 1001$, cuyo resultado es 10110.



(a) Sumador de 4 bits. Basado en (Mano, 2014).

$Y = (A + B)$				
Sumador	3	2	1	0
C_{in}	0	0	1	0
A	1	1	0	1
B	1	0	0	1
Y	0	1	1	0
C_{out}	1	0	0	1

(b) Suma binaria.

Figura 10: Sumador completo. Basado en (Mano, 2014).

Esta forma de conectar los acarreos entre sumadores se conoce como *acarreo en serie*. Dado que el acarreo se debe propagar etapa por etapa, se presenta un retardo debido a los tiempos de propagación de la señal de acarreo. Existe otro método llamado *acarreo anticipado*, que permite disminuir estos retardos. El método consiste en determinar mediante lógica combinatorial, según el valor de las entradas y del acarreo de entrada, si el sumador va a generar un acarreo de salida.

En la tabla 3 se presentan dos referencias de sumadores comerciales. Las dos referencias son básicamente iguales, sin embargo, el 74LS283 es más utilizado debido a que los pines de conexión están ubicados de una manera estándar (Tierra en pin 8, VCC en pin 16).

Tipo	Descripción
74LS83	Sumador paralelo de 4 bits.
74LS283	Sumador paralelo de 4 bits.

Tabla 3: Referencias de sumadores. *Elaboración propia*.

3.4. Restadores

No existe como tal un circuito integrado que realice la operación de resta. El procedimiento consiste entonces en utilizar un sumador como restador. Para esto, el sustraendo se debe representar con su complemento a dos.

Cabe recordar que "el complemento a dos se obtiene calculando el complemento a uno y sumando 1 al bit menos significativo" (Mano, 2014, p.126). Este método permite representar números con signo, de la siguiente manera (Tocci y cols., 2007, p.300):

- Cuando el número es positivo, se utiliza en el bit más significativo un '0' como signo. El resto de los bits representan el valor real en binario del número. Así, el número 90_{10} (90 en base 10) se representaría como 01011010.
- Cuando el número es negativo, en el bit más significativo se pone un '1' para representar el signo. El resto de los bits representan el valor, en complemento a 2, del número en binario. Ahora, el número -90_{10} se representaría como 10100101.

En la figura 11a se muestra una resta utilizando el complemento a 2. Para implementarla en un circuito (figura 11b), es necesario complementar a uno el sustraendo (En este caso, invertir B) y sumarle 1 (este paso implica agregar un acarreo de entrada para convertir dicho número en complemento a 2). El resultado obtenido corresponde a la resta entre A y B.

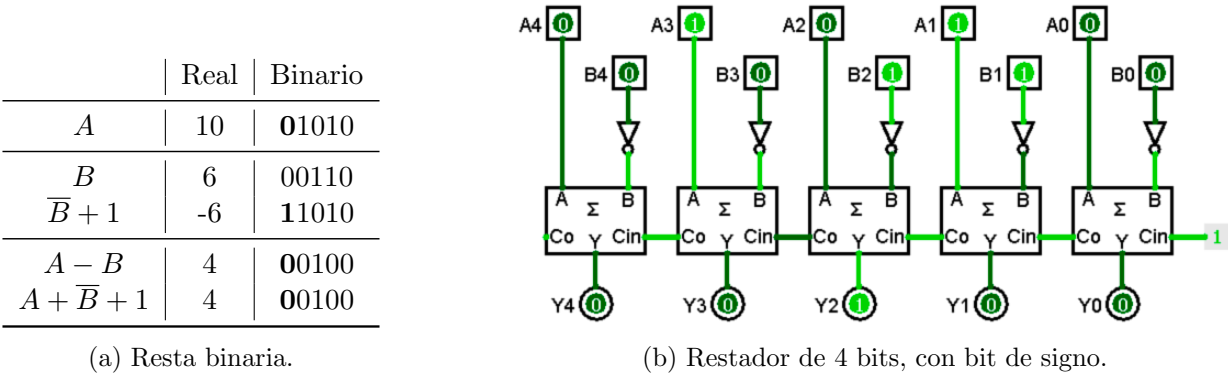


Figura 11: Sumador completo. *Elaboración propia.*

Para el circuito restador mostrado, si A es mayor que B, el resultado será un número positivo (como en el ejemplo). Si B es mayor que A, el resultado será el bit de signo en 1 y el complemento a 2 del resultado.

3.5. Multiplicadores binarios

De la misma manera que con los restadores, no existe un circuito integrado comercial para la multiplicación, por lo tanto se debe realizar esta operación utilizando lógica combinatorial. Recordando lo visto en la Unidad 1, la multiplicación de dos números binarios A y B será:

	A_1	A_0
x	B_1	B_0
	$B_0 * A_1$	$B_0 * A_0$
$B_1 * A_1$	$B_1 * A_0$	
Y =	$B_1 * A_1$	$B_0 * A_1 + B_1 * A_0$
		$B_0 * A_0$

Cada uno de los términos del resultado tienen una posición específica en la salida (Y), así $Y_0 = B_0 * A_0$, $Y_1 = B_0 * A_1 + B_1 * A_0$, $Y_2 = B_1 * A_1$ y el acarreo final de esta suma sería el término restante Y_3 de ser necesario. Recuerde que la multiplicación binaria de 1 bit corresponde a la operación AND. Este procedimiento es similar para multiplicadores de más bits. En la figura 12 se encuentra el circuito combinacional resultante, para la multiplicación binaria de 2 bits. Este circuito está montado utilizando semisumadores, razón por la cual no se utiliza el acarreo de entrada. Nótese que se el máximo valor posible de la multiplicación entre 2 números de 2 bits es el 9_{10} , resultado de multiplicar $3_{10} * 3_{10}$ ($11_2 * 11_2$ en base binaria). Este valor es el que se muestra en el circuito de la figura.

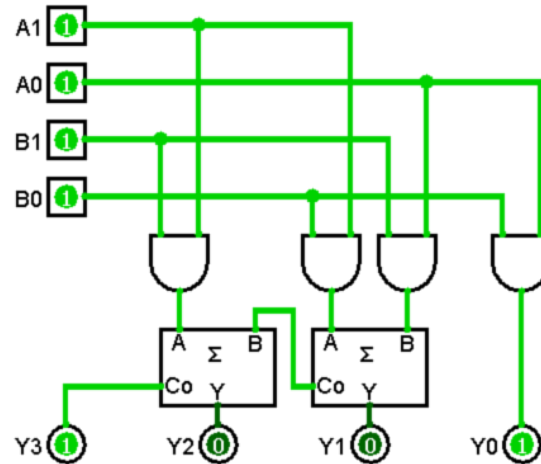


Figura 12: Multiplicador de 2 bits. Basado en (Mano, 2014)

4. Comparadores de magnitud

Este tipo de circuitos compara dos números binarios (A y B) de n bits en sus entradas. Hay 3 posibilidades: A y B son iguales ($A = B$), A es mayor que B ($A > B$) o A es menor que B ($A < B$). Para evaluar la igualdad, recuerde que una compuerta XOR indica si sus dos entradas son diferentes o no. Al negar este resultado se puede verificar la igualdad entre sus entradas (figura 13a). Esto se puede evaluar bit a bit y pasarlo por una compuerta AND para verificar que en todos se cumpla la condición.

Con respecto a si un número es mayor que otro, se realiza la evaluación desde el bit superior (n), si $A_n = 1$ y $B_n = 0$ entonces $A > B$ (figura 13b). De lo contrario, si $A_n = 0$ y $B_n = 1$ entonces $A < B$ (figura 13c). Si ambos bits son iguales, se procede a revisar el bit inmediatamente inferior ($n - 1$).

El circuito integrado 74HC85 es un comparador de 4 bits, se puede ver su estructura general en la figura 14a. El comparador cuenta con 2 entradas de 4 bits y las 3 salidas mencionadas: $A = B$, $A > B$ y $A < B$. Además, cuenta con 3 entradas de cascada, que permiten su funcionamiento para comparación de números con más de 4 bits. Las salidas del comparador se conectan a las entradas de cascada de otro comparador (Que podría estar comparando los bits $A_7A_6A_5A_4$ con $B_7B_6B_5B_4$, si A y B son de 8 bits en total). Note que el comparador de bits menos significativos tiene sus entradas en cascada $A > B$ y $A < B$ conectadas a tierra, mientras que $A = B$ está conectada a un nivel ALTO.

En este momento lo invito a volver al escenario, donde encontrará una actividad que le permitirá poner a prueba

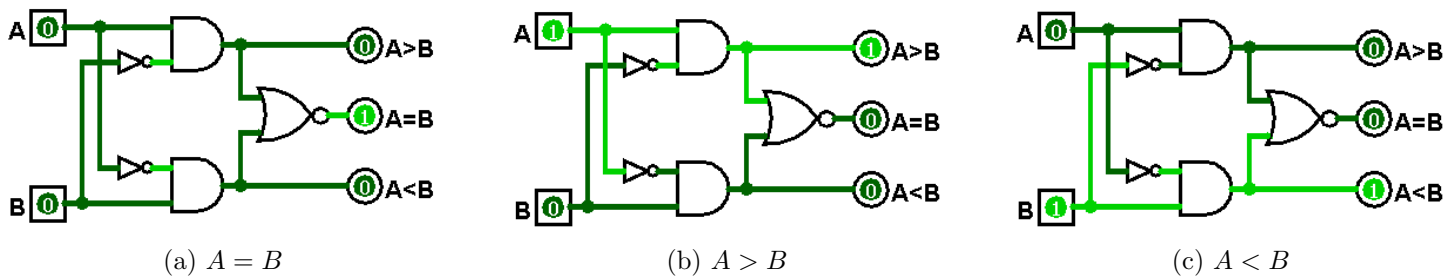


Figura 13: Circuito comparador de 1 bit. *Elaboración propia.*

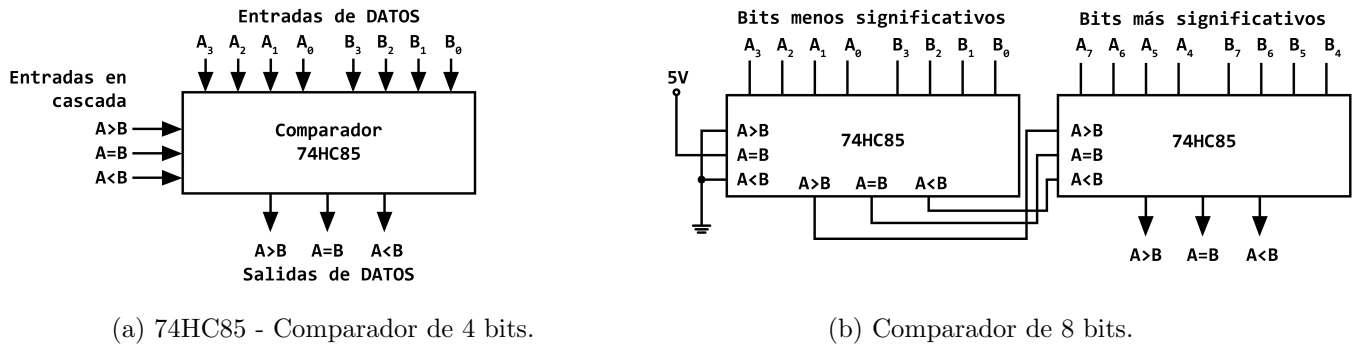


Figura 14: Comparador 74HC85. *Basado en (Tocci y cols., 2007)*

los conceptos vistos hasta ahora en cuento a circuitos aritméticos y comparadores.

5. Generadores y comprobadores de paridad

Al realizar comunicación entre dos dispositivos, es posible que se generen errores en la transmisión, modificando el mensaje original. Esto puede generar errores o problemas mayores en el destinatario del mensaje. Por esta razón, es necesario aplicar métodos que permitan la detección de errores en los mensajes recibidos. Uno de estos métodos consiste en agregar un bit de *paridad* al mensaje. Este bit puede indicar dos cosas:

- Paridad par: Hace que el total de unos en el mensaje sea par.
- Paridad impar: Hace que el total de unos en el mensaje sea impar.

Por ejemplo, si el mensaje a enviar es: 0010 000, y el envío tiene paridad par, se agrega un 1 al inicio: 1001 0000. Si por el contrario, la paridad es impar, se agrega un 0 al inicio: 0001 0000. El receptor del mensaje deberá saber con qué tipo de paridad fue enviado, para así entrar a verificar el número de unos. De esta forma, será posible detectar si hay alguna inconsistencia.

El circuito integrado 74LS280 permite la generación/comprobación de paridad de un número binario hasta 9 bits. A continuación, se presentan dos circuitos combinacionales que permiten comprender el funcionamiento de este

integrado. El primer circuito permite la generación del bit de paridad; el segundo, la detección o comprobación del mismo.

5.1. Generadores

Un circuito generador de paridad hace uso de compuertas XOR para realizar la suma (sin acarreo) de los bits en el mensaje. Cuando el número de unos es par ($1 + 1 = 0$), el resultado es cero. Cuando el número de unos es impar ($1 + 0 = 1$), el resultado será un uno. El circuito generador de paridad par, para un mensaje de 4 bits, se puede ver en la figura 15. Para un generador de paridad impar sólo se deberá invertir el resultado en el bit de paridad.

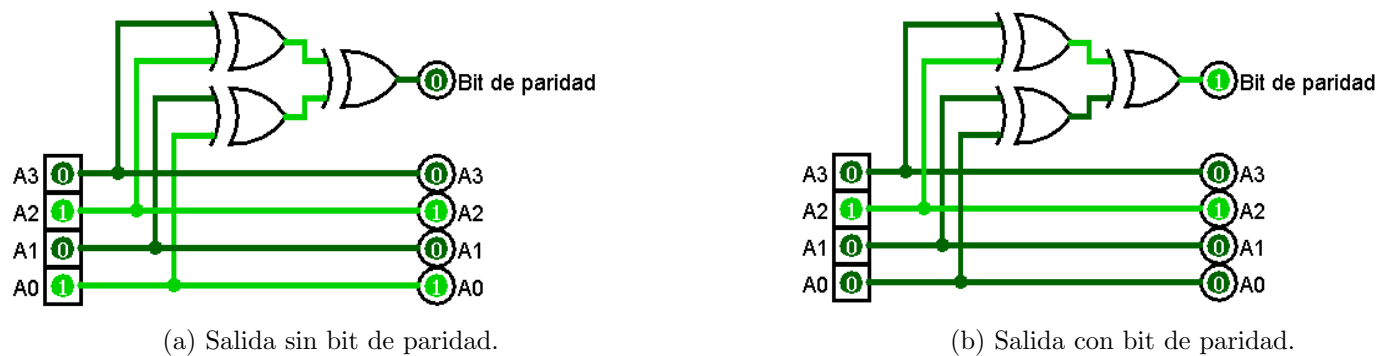


Figura 15: Generador de paridad. *Basado en (Tocci y cols., 2007)*

5.2. Comprobadores

Un circuito comprobador de paridad verifica que el mensaje tenga la cantidad de unos adecuados. Se utiliza el mismo principio de la suma de unos que en el generador. Si el número de unos en el mensaje es par, el resultado es cero; si es impar, el resultado es uno. En la figura 16 se presentan dos casos para el comprobador, enviando el mensaje 00101 que tiene paridad par. Si el bit de paridad es correcto, el circuito entrega como salida un cero, que indica que no hay error (figura 16a). Por el contrario, cuando el mensaje recibido es 10101, la cantidad de unos es impar y el circuito entrega en su salida un uno, indicando error (figura 16b)

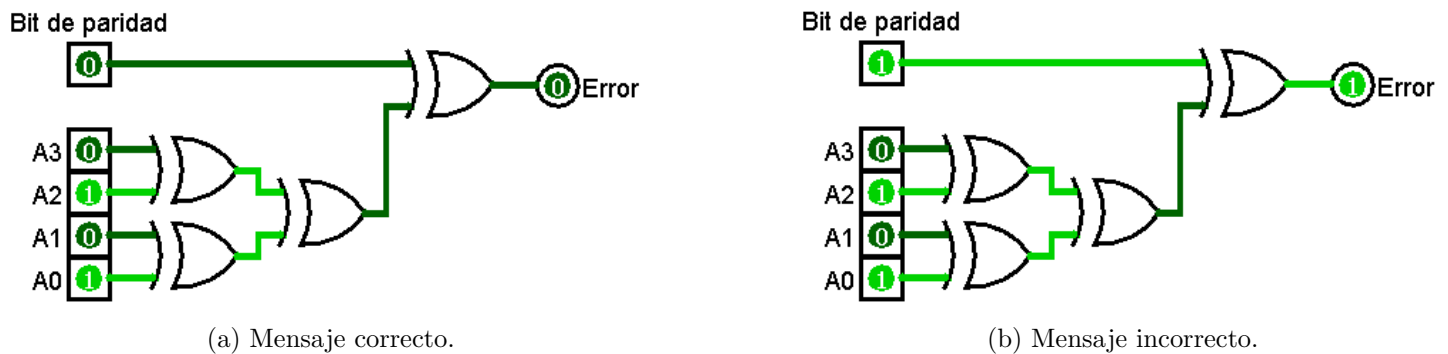


Figura 16: Comprobador de paridad. *Basado en (Tocci y cols., 2007)*

Índice de figuras

1	Diagrama funcional de un multiplexor. <i>Basado en Tocci y cols. (2007)</i>
2	Funcionamiento de un Multiplexor de 4 entradas. <i>Elaboración propia.</i>
3	Diagramas funcionales de dos multiplexores digitales. <i>Elaboración propia.</i>
4	Enrutamiento de datos - Displays de 7 segmentos. <i>Basado en Floyd (2010).</i>
5	Generación de funciones lógicas. <i>Elaboración propia.</i>
6	Diagrama funcional de un demultiplexor. <i>Basado en Tocci y cols. (2007)</i>
7	Diagramas funcionales de tres demultiplexores comerciales. <i>Elaboración propia.</i>
8	Semisumador. <i>Basado en (Mano, 2014).</i>
9	Sumador completo. <i>Basado en (Mano, 2014).</i>
10	Sumador completo. <i>Basado en (Mano, 2014).</i>
11	Sumador completo. <i>Elaboración propia.</i>
12	Multiplicador de 2 bits. <i>Basado en (Mano, 2014)</i>
13	Circuito comparador de 1 bit. <i>Elaboración propia.</i>
14	Comparador 74HC85. <i>Basado en (Tocci y cols., 2007)</i>
15	Generador de paridad. <i>Basado en (Tocci y cols., 2007)</i>
16	Comprobador de paridad. <i>Basado en (Tocci y cols., 2007)</i>

Índice de tablas

1	Referencias de multiplexores comerciales. <i>Elaboración propia.</i>
2	Referencias de demultiplexores comerciales. <i>Elaboración propia.</i>
3	Referencias de sumadores. <i>Elaboración propia.</i>

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 4: Circuitos lógicos combinacionales I

Autor: Gabriel Eduardo Ávila Buitrago

Asesor Pedagógico: Jeimmy Lorena Romero Perilla

Diseñador Gráfico: Leonardo Stiglitch Campos

Asistente: Jhon Edwar Vargas Villa

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