



INSTITUTO SUPERIOR
UNIVERSITARIO

SUPE

**GUÍA GENERAL DE ESTUDIO
DE LA ASIGNATURA DE
ELECTRÓNICA DIGITAL**



Guía general de estudio de la asignatura de Electrónica Digital

Edgar Javier Cajas Oña

Alex Fabricio Oña Ñacata

Mayra Alejandra Sarzosa Villarroel

Robinson Lema Parco

2024

Esta publicación ha sido sometida a revisión por pares académicos específicos por:

Nelson Rolando Sinchiguano Chilingua
Instituto Superior Universitario Cotopaxi

Corrección de estilo:

- Karla Jaramillo - Docente - Sucre
- Freddy Centeno - Docente - Sucre
- Ana Llumiyinga - Docente - Sucre

Diseño y diagramación:

- Ronny Chaguay - Docente - Sucre
- Diego Bonilla - Docente - Sucre

Primera Edición
Quito - Ecuador

ISBN: 978-9942-676-26-9

Esta publicación está bajo una licencia de Creative Commons
Reconocimiento-No Comercial-Compartir Igual 4.0 Internacional.

Los contenidos de este trabajo están sujetos a una licencia internacional Creative Commons Reconocimiento-NoComercial-CompartirIgual 4.0 (CC BY-NC-SA 4.0). Usted es libre de Compartir — copiar y redistribuir el material en cualquier medio o formato. Adaptar — remezclar, transformar y construir a partir del material citando la fuente, bajo los siguientes términos: Reconocimiento- debe dar crédito de manera adecuada, brindar un enlace a la licencia, e indicar si se han realizado cambios. Puede hacerlo en cualquier forma razonable, pero no de forma tal que sugiera que usted o su uso tienen el apoyo de la licenciante. No Comercial-no puede hacer uso del material con propósitos comerciales. Compartir igual-Si remezcla, transforma o crea a partir del material, debe distribuir su contribución bajo la misma licencia del original. No puede aplicar términos legales ni medidas tecnológicas que restrinjan legalmente a otras a hacer cualquier uso permitido por la licencia.
<https://creativecommons.org/licenses/by-nc-sa/4.0/>



Reconocimiento-NoComercial-CompartirIgual 4.0 Internacional (CC BY-NC-SA 4.0)

Usted acepta y acuerda estar obligado por los términos y condiciones de esta Licencia, por lo que, si existe el incumplimiento de algunas de estas condiciones, no se autoriza el uso de ningún contenido.



MISIÓN

Ser una Institución Superior Universitaria con estándares de calidad académica e innovación, reconocida a nivel nacional con proyección internacional.

VISIÓN

Formamos profesionales competentes con espíritu emprendedor, capaces de contribuir al desarrollo integral del país.

ÍNDICE

ÍNDICE	5
Presentación de la asignatura	7
Resultados del aprendizaje	7
UNIDAD 1 SISTEMAS DE NUMERACIÓN Y OPERACIONES LÓGICAS	8
Sistemas de numeración.....	8
Conversión de sistemas de numeración	8
Aritmética de sistemas de numeración	12
Complementos.....	13
Resta utilizando complementos	13
Ejercicios propuestos.....	15
UNIDAD 2 ÁLGEBRA BOOLEANA Y COMPUERTAS LÓGICA	17
Álgebra Booleana	17
Compuertas lógicas	17
Tablas de verdad	17
Tipo de compuertas lógicas	17
Identidades del álgebra de Boole	18
Estandarización de funciones.....	19
Mapas de Karnaugh.....	22
Simplificación de funciones	23
Ejercicios prácticos propuestos	24
UNIDAD 3 FUNCIONES Y CIRCUITOS DE LÓGICA COMBINACIONAL	25
Decodificadores	25
Aplicaciones de los decodificadores	26
Multiplexores.....	27
Demultiplexor	30
Circuitos digitales sumadores	31
Circuitos digitales comparadores	32
Ejercicios Propuestos.....	34
UNIDAD 4 FUNCIONES Y CIRCUITOS DE LÓGICA SECUENCIAL	35

Flip-Flop D (data).....	35
Flip-Flop T	35
Flip-Flop JK.....	35
Flip-Flop SR	36
Ejercicio Propuesto.....	36
Contadores	37
Registro de desplazamiento.....	40
Máquinas de estado	44
Autoevaluación	46
Prueba de diagnóstico	47
Referencias bibliográficas	48

Presentación de la asignatura

La asignatura de electrónica digital tiene como objetivo introducir los elementos básicos y las técnicas necesarias para analizar y diseñar circuitos lógicos combinacionales y secuenciales. También se enfoca en el funcionamiento de diversos circuitos integrados MSI, que son utilizados en diseños de mayor complejidad para distintas aplicaciones.

Resultados del aprendizaje

Utiliza adecuadamente sistemas numéricos para la comprensión de la estructura de los sistemas de cómputo.

Demuestra y compara las diferentes tecnologías electrónicas utilizadas en la implementación de circuitos y sistemas digitales.

Discute sobre las características más importantes de los diseños tales como tiempos de conmutación, consumo de energía, velocidad de procesamiento de datos, etc.

Desarrolla simplificaciones de expresiones booleanas e implementa circuitos sintetizados. Con todo esto, estará en capacidad de organizar, dirigir, ejecutar y controlar tareas en el sector industrial en las que el diseño, implementación, mantenimiento y reparación de sistemas electrónicos digitales sea la base tecnológica requerida.

UNIDAD 1 SISTEMAS DE NUMERACIÓN Y OPERACIONES LÓGICAS

Sistemas de numeración

En electrónica digital, los sistemas de numeración más comúnmente utilizados son el decimal, binario, octal y hexadecimal, cada sistema tiene una base específica que determina la cantidad de dígitos únicos que utiliza y cómo se representan los valores numéricos. En la Tabla 1 se muestran los sistemas más comunes en el ámbito de la electrónica.

Tabla 1
Tipos de sistemas de numeración

<u>Nombre del sistema</u> <u>en función de su base</u>	<u>Definición</u>
<u>Sistema Decimal (Base 10)</u>	<u>El sistema decimal es el más comúnmente utilizado en la vida diaria. Está basado en 10 dígitos (0-9). Cada posición de un número en este sistema tiene un valor basado en potencias de 10.</u>
<u>Sistema Binario (Base 2)</u>	<u>El sistema binario es fundamental en electrónica digital y computación. Utiliza solo dos dígitos: 0 y 1. Cada posición en un número binario representa una potencia de 2.</u>
<u>Sistema Octal (Base 8)</u>	<u>El sistema octal utiliza ocho dígitos (0-7). Es útil en electrónica digital porque es más compacto que el binario y se puede convertir fácilmente a binario y viceversa.</u>
<u>Sistema Hexadecimal (Base 16)</u>	<u>El sistema hexadecimal utiliza dieciséis dígitos (0-9 y A-F, donde A=10, B=11, ..., F=15). Es ampliamente utilizado en informática y electrónica digital porque se puede convertir fácilmente a binario y es más compacto.</u>

Conversión de sistemas de numeración

Conversión directa

En base a la equivalencia entre los sistemas más comunes, se puede realizar una conversión directa, en la Tabla 2, se describen las equivalencias entre los cuatros sistemas más comunes, para sus 16 primeros números.

Tabla 2
Tipos de sistemas de numeración

<u>Decimal</u>	<u>Octal</u>	<u>Hexadecimal</u>	<u>Binario</u>
<u>0</u>	<u>0</u>	<u>0</u>	<u>0000</u>
<u>1</u>	<u>1</u>	<u>1</u>	<u>0001</u>
<u>2</u>	<u>2</u>	<u>2</u>	<u>0010</u>
<u>3</u>	<u>3</u>	<u>3</u>	<u>0011</u>
<u>4</u>	<u>4</u>	<u>4</u>	<u>0100</u>
<u>5</u>	<u>5</u>	<u>5</u>	<u>0101</u>
<u>6</u>	<u>6</u>	<u>6</u>	<u>0110</u>
<u>7</u>	<u>7</u>	<u>7</u>	<u>0111</u>
<u>8</u>	<u>10</u>	<u>8</u>	<u>1000</u>
<u>9</u>	<u>11</u>	<u>9</u>	<u>1001</u>
<u>10</u>	<u>12</u>	<u>A</u>	<u>1010</u>
<u>11</u>	<u>13</u>	<u>B</u>	<u>1011</u>
<u>12</u>	<u>14</u>	<u>C</u>	<u>1100</u>
<u>13</u>	<u>15</u>	<u>D</u>	<u>1101</u>
<u>14</u>	<u>16</u>	<u>E</u>	<u>1110</u>
<u>15</u>	<u>17</u>	<u>F</u>	<u>1111</u>

Conversión de binario a octal y hexadecimal

Para transformar a octal se debe separar en 3 dígitos el número binario.

Para transformar a hexadecimal se debe separar en 4 dígitos el número binario.

Métodos de sustitución para la conversión de números

Este método nos permite transformar números de cualquier base a base 10, y se puede representar de la siguiente manera:

$$N)r = \sum_{i=-m}^{n-1} a_i r^i; \text{ donde: } n = \text{Parte entera y } m = \text{Parte decimal.}$$

Ejemplo: Transformar el siguiente número hexadecimal $(2AB, 12C)_{16}$ aplicando el método de sustitución a un número decimal.

$$\begin{aligned}
 & \underline{(2AB, 12C)_{16}} \\
 & \quad \underline{n=3} \\
 & \quad \underline{m=3} \\
 & \quad \underline{r=16} \\
 (2AB, 12C)_{16} &= 2x16^2 + Ax16^1 + Bx16^0 + 1x16^{-1} + 2x16^{-2} + \\
 & \quad \underline{Cx16^{-3}} \\
 &= 2x16^2 + 10x16^1 + 11x16^0 + 1x16^{-1} + 2x16^{-2} + 12x16^{-3} \\
 &= 2x256 + 10x16 + 11 + \frac{1}{16} + \frac{2}{256} + \frac{12}{4096} \\
 &= 512 + 160 + 11 + \frac{1}{16} + \frac{2}{256} + \frac{12}{4096} \\
 &= 683 + 0,0734863 \\
 &= \underline{(683,0732422)_{10}}
 \end{aligned}$$

Método de Divisiones y Multiplicación Sucesivas

Este método permite convertir números de base 10 a cualquier otra base. Para la parte entera, se realizan divisiones sucesivas: se divide el número por la base deseada y se toma el residuo como el dígito menos significativo. El cociente se divide nuevamente hasta que sea menor que la base, siendo el último residuo el dígito más significativo. Para la parte fraccionaria, se realizan multiplicaciones sucesivas: se multiplica la parte fraccionaria por la base deseada, separando el resultado en parte entera y fraccionaria. Este proceso se repite hasta que la parte fraccionaria sea cero.

Ejemplo: Transformar el siguiente número decimal $1452,671875_{10}$ a un número de base "8", utilizando el método de divisiones y multiplicación sucesivas.

1452	8			
65	181	8		
12	21	22	8	
(4)	(5)	(6)	(2)	

← 2654₈

0,671875	× 8 =	5,375	5 ↓
0,375	× 8 =	3	3 ↓

Respuesta: $2654,53_8$

Ejemplo: Transformar el siguiente número binario: 011110101110110110,101111110101 a octal y hexadecimal.

<u>A octal</u>	$\begin{array}{cccccccccccc} \underline{011} & \underline{110} & \underline{101} & \underline{110} & \underline{110} & \underline{110}, & \underline{101} & \underline{111} & \underline{110} & \underline{101} \\ 3 & 6 & 5 & 6 & 6 & 6, & 5 & 7 & 6 & 5 \end{array}$
<u>Dividimos en tres dígitos el número binario</u>	<u>365666,5765₈</u>
<u>A hexadecimal</u>	$\begin{array}{cccccccc} \underline{0001} & \underline{1110} & \underline{1011} & \underline{1011} & \underline{0110}, & \underline{1011} & \underline{1111} & \underline{0101} \\ 1 & E & B & B & 6, & B & F & 5 \end{array}$
<u>Dividimos en cuatro dígitos el número binario</u>	<u>1EBB6,BF5₁₆</u>

Conversión de binario a decimal

Es conveniente tratar el caso particular de convertir un número binario a decimal por ser uno de los más utilizados en sistemas digitales y porque el método puede ser simplificado.

Para ello es conveniente memorizar algunas potencias de “2”, como se observa en la Tabla 3.

Tabla 3
Cuadro de potencias

<u>Posición</u>	<u>...</u>	<u>9</u>	<u>8</u>	<u>7</u>	<u>6</u>	<u>5</u>	<u>4</u>	<u>3</u>	<u>2</u>	<u>1</u>	<u>0</u>	<u>Punto</u>	<u>-1</u>	<u>...</u>
<u>Peso</u>	<u>...</u>	<u>2⁹</u>	<u>2⁸</u>	<u>2⁷</u>	<u>2⁶</u>	<u>2⁵</u>	<u>2⁴</u>	<u>2³</u>	<u>2²</u>	<u>2¹</u>	<u>2⁰</u>	.	<u>2⁻¹</u>	<u>...</u>
<u>Valor</u>	<u>...</u>	<u>512</u>	<u>256</u>	<u>128</u>	<u>64</u>	<u>32</u>	<u>16</u>	<u>8</u>	<u>4</u>	<u>2</u>	<u>1</u>	.	<u>0.5</u>	<u>...</u>

Ejemplo: Utilizando la conversión binaria a decimal y transforma el siguiente número (10111101)₂

128 64 32 16 8 4 2 1

2⁷ 2⁶ 2⁵ 2⁴ 2³ 2² 2¹ 2⁰

1 0 1 1 1 1 0 1 Sumar solo los casilleros que están pintados.

= (128+32+16+8+4+1) = (189)₁₀

Complementos

Los complementos de números binarios sirven para transformar la resta en una suma y se puede obtener por medio de los complementos A_1 y A_2 .

Complemento A_1 (CA_1)

Para obtener los complementos A_1 de un número binario se combina los unos por ceros y los ceros por unos.

1 0 1 1 1 0 1 1 0 1 1 1, 1 0 1 1 0

0 1 0 0 0 1 0 0 1 0 0 0, 0 1 0 0 1: CA_1

Complemento A_2 (CA_2)

Primera forma: Para obtener el complemento de A_2 de un número binario, primero se obtiene el complemento A_1 y luego se suma "1" al dígito menos significativo del componente A_1 .

$$\begin{array}{r}
 , \\
 CA_1 , \\
 \hline
 , \\
 CA_2 ,
 \end{array}$$

Dígito menos significativo

Segunda forma: Para obtener el complemento A_2 de un número binario debe ubicarse en el dígito menos significativo del número. Luego nos movemos hacia la izquierda hasta encontrar el primer "1", los dígitos no se invierten dejamos intacto este "1" y luego invertimos todos.

$$\begin{array}{r}
 , \\
 CA_2 , \\
 , \\
 ,
 \end{array}$$

muevo hacia la izquierda

Dígito menos significativo

Resta utilizando complementos

Resta con complemento A_1

Para realizar la resta utilizando CA_1 debe seguir los siguientes pasos: Iguala el número de dígitos del minuendo y sustraendo, obtén el CA_1 del sustraendo, suma el minuendo con el CA_1 del sustraendo,

verifica el carry o acarreo: Si no existe carry se obtiene el CA_1 del resultado del paso 3 y se agrega el signo negativo (-). Si existe carry, el valor del carry le sumamos al dígito menos significativo del resultado obtenido del paso 3 y el resultado es positivo (+).

Ejemplo: Resuelve el siguiente ejercicio planteado aplicando la resta con complemento A1.

1 0 1 1 1 0 1 1 1, 1 1 0 1 1 Minuendo (M)

1 0 1 1, 0 1 1 1 1 Sustraendo (S)

1 0 1 1 1 0 1 1 1, 1 1 0 1 1 M
0 0 0 0 0 1 0 1 1, 0 1 1 1 1 S

Carry 1

1 1 1 1 1 0 1 0 0, 1 0 0 0 0 CA_1 "S"
+ 1 0 1 1 1 0 1 1 1, 1 1 0 1 1 M
1 0 1 1 0 1 1 0 0, 0 1 0 1 1
+ 1 Carry
1 0 1 1 0 1 1 0 0, 0 1 1 0 0

Resta con complemento A2

Para realizar la resta utilizando CA_2 debe seguir los siguientes pasos: Iguala el número de dígitos tanto minuendo como el sustraendo, obtén el CA_2 del sustraendo, suma el minuendo con el CA_2 del sustraendo, verifica el carry: Si no existe carry obtener el CA_2 del resultado del paso 3 y agrega el signo negativo, Si existe carry se lo descarta o se elimina y el resultado es positivo.

Ejemplo: Resuelve el siguiente ejercicio planteado aplicando la resta con complemento A2.

1 0 1 1 1 0 0, 1 0 1 1 0 M

1 1 0 1 0 1 0 1 1 1, 1 0 1 S

0 0 0 1 0 1 1 1 0 0, 1 0 1 1 0 M
1 1 0 1 0 1 0 1 1 1, 1 0 1 0 0 S

0 0 1 0 1 0 1 0 0 0, 0 1 1 0 0 CA₂S
+ 0 0 0 1 0 1 1 1 0 0, 1 0 1 1 0 M

0 1 0 0 0 0 0 1 0 1, 0 0 0 1 0 RESULTADO

No existe

carry

-1 0 1 1 1 1 1 0 1 0, 1 1 1 1 0 CA₂ RESULTADO

Ejercicios propuestos

1. Convertir los siguientes números binarios a sus equivalentes en base 10: $(1110011)_2$, $(01001)_2$, $(010011)_2$
2. Convertir los siguientes números decimales a sus equivalentes en binario: $(69)_{10}$, $(35)_{10}$, $(1,25)_{10}$, $(20,5)_{10}$
3. Convertir los siguientes números enteros hexadecimales a decimal: $(5E2)_{16}$, $(F0BE)_{16}$
4. Convertir los siguientes números reales hexadecimales a decimal: $(E2, A)_{16}$, $(B7, D)_{16}$
5. Convertir el número hexadecimal a decimal y a binario: $(11,5)_{16}$
6. Convertir el siguiente número hexadecimal a binario, octal y decimal (considerar 5 cifras fraccionarias): $(6416213A, 17B)_{16}$
7. Convertir el siguiente número binario a base hexadecimal: $(001111110011101010100)_2$
8. Convertir el siguiente número binario a base octal: $(1001011010111001)_2$
9. Resuelve la siguiente suma binaria: $11100111 + 101001$
10. Resuelve la siguiente resta binaria: $111001 - 1011$
11. Resuelve las multiplicaciones y divisiones binarias: 110110×1101 , $11010 \div 1101$
12. Determina el complemento A1 del siguiente número binario: 111100011110111

13. Encontrar el complemento A2 del siguiente número binario: 1100010110
14. Realizar las siguientes restas por complementos: 11100111 – 00010011

UNIDAD 2 ÁLGEBRA BOOLEANA Y COMPUERTAS LÓGICA

Álgebra Booleana

Es un sistema matemático que modela la lógica binaria, donde las variables y operaciones toman valores de 1 o 0 y se manipulan según las reglas del álgebra.

Compuertas lógicas

Son bloques fundamentales de construcción de circuitos digitales y pueden ser AND, NAND, OR, NOT, XOR, NAND, NOR.

Tablas de verdad

Es un formato tabular que muestra todas las combinaciones posibles valores de entrada junto con los resultados que surgen de una función lógica específica, ver Tabla 5.

Tabla 5

Tabla de verdad de tres variables

<u>A</u>	<u>B</u>	<u>C</u>	<u>F (A, B, C)</u>
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

Tipo de compuertas lógicas

Las compuertas lógicas se utilizan para diseñar y construir circuitos digitales más complejos, cada compuerta lógica sigue una tabla de verdad que da las posibles combinaciones de entrada y la respectiva salida obtenida, ver

Figura 1 y Tabla 6.

Figura 1
Símbolo de compuertas lógicas

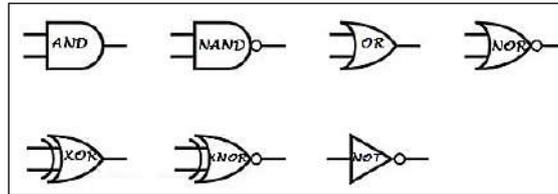


Tabla 6
Tabla de verdad compuerta lógicas

IN		AND	OR	NOT	XOR	NAND	NOR	XNOR
A	B	A.B	A+B	\bar{A}	$A \oplus B$	$\overline{A.B}$	$\overline{A+B}$	$\overline{A \oplus B}$
0	0	0	0	1	0	1	1	1
0	1	0	1	1	1	1	0	0
1	0	0	1	0	1	1	0	0
1	1	1	1	0	0	0	0	1

Identidades del álgebra de Boole

Permite la simplificación y optimización de expresiones booleana mediante las operaciones AND, OR, NOT y otras más complejas como la AND, XOR, NOR, XNOR, ver Tabla 7.

Tabla 7
Algebra de Boole

Identidad	Dualidad	Commutativa	Asociativa	Distributiva	Morgan	Identidades Adicionales
$A + 0 = A$	$A * 1 = A$	$A + B = B + A$	$A + (B + C)$	$A * (B + C)$	$\overline{\overline{A + B + C + \dots}}$	$A + A * B = A$
			$= (A + B) + C$	$= A * B + A * C$	$= \bar{A} * \bar{B} * \bar{C} \dots$	
$A + 1 = 1$	$A * 0 = 0$	$A * B = B * A$				$A + \bar{A} * B$
						$= A + B$
			$A * (B * C)$	$A + (B * C)$	$= \bar{A} + \bar{B} + \bar{C} \dots$	
$A + A = A$	$A * A = A$		$= (A * B) * C = (A + B) * (A + C)$			$(A * B)$
						$+ (\bar{A} * \bar{B}) * C$
$A + \bar{A} = 1$	$A * \bar{A} = 0$					$= (A * B) + C$
$\bar{\bar{A}} = A$						$\bar{A} + (A * \bar{B})$
						$= \bar{A} + \bar{B}$

Ejemplos: Simplificar la siguiente función lógica aplicando el algebra booleana, para tal efecto encontrar el circuito lógico de la ecuación simplificada.

$$F(A, B, C, D) = \bar{A}BC + ABC + AC\bar{D}$$

$$F(A, B, C, D) = (\bar{A}BC + ABC) + AC\bar{D}$$

$$F(A, B, C, D) = BC(\bar{A} + A) + AC\bar{D}$$

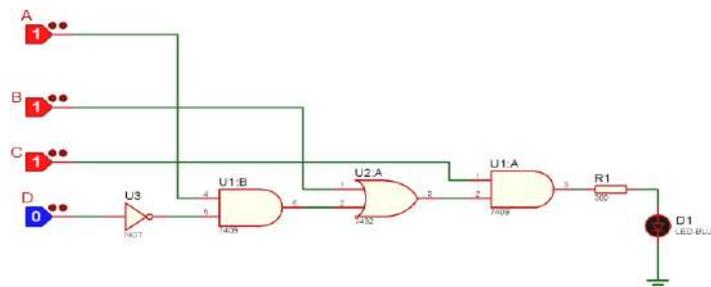
$$F(A, B, C, D) = BC(1) + AC\bar{D}$$

$$F(A, B, C, D) = BC + AC\bar{D}$$

$$F(A, B, C, D) = C(B + A\bar{D}), \text{ (ver Figura 2).}$$

Figura 2

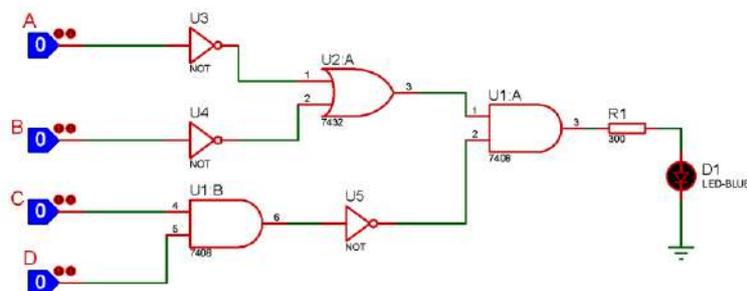
Circuito lógico ejemplo planteado



Ejemplo: En base al circuito lógico que se muestra en la Figura 3, encuentre la ecuación de salida.

Figura 3

Circuito lógico ejemplo planteado



$$F(A, B, C, D) = (\bar{A} + \bar{B})(\bar{C}\bar{D}).$$

Estandarización de funciones

Cada función lógica debe contener todas las variables de entrada, dicha función a ser estandarizada se la puede representar en forma de suma de productos o como productos de sumas.

Suma de productos

Para estandarizar como suma de productos una función, aplicamos las siguientes identidades:

$$A + A = A$$

$$A + \bar{A} = 1$$

$$A(B + C) = AB + AC$$

Al realizar la suma de productos se encuentran los términos mínimos los mismos que cumplen la función de asignar el dígito “1” a la variable sin complementar y el dígito “0” a la variable complementada.

Ejemplo: encuentre los términos mínimos y la tabla de verdad de la siguiente función, para tal efecto utilizar suma de productos, ver Tabla 8.

$$F(A, B, C) = AB + A\bar{C} + \bar{B}C$$

Completo cada función con la variable de entrada que le falta.

$$F(A, B, C) = AB + A\bar{C} + \bar{B}C$$

$$F(A, B, C) = AB(C + \bar{C}) + A\bar{C}(B + \bar{B}) + \bar{B}C(A + \bar{A})$$

$$F(A, B, C) = ABC + AB\bar{C} + A\bar{C}B + A\bar{C}\bar{B} + \bar{B}CA + \bar{B}C\bar{A}$$

Ordena las variables de entrada de cada ecuación.

$$F(A, B, C) = ABC + AB\bar{C} + A\bar{C}B + A\bar{C}\bar{B} + \bar{B}CA + \bar{B}C\bar{A}$$

Términos semejantes

$$F(A, B, C) = ABC + AB\bar{C} + A\bar{B}\bar{C} + A\bar{B}C + \bar{A}\bar{B}C$$

A esta última función se la llama **TÉRMINOS MÍNIMOS**.

$$F(A, B, C) = 111 \quad 110 \quad 100 \quad 101 \quad 001$$

$$F(A, B, C) = m7 \quad m6 \quad m4 \quad m5 \quad m1$$

$$F(A, B, C) = \sum m(1,4,5,6,7)$$

Tabla 8

Tabla de verdad ejemplo planteado

A	B	C	F(A,B,C)
0	0	0	
0	0	1	1
0	1	0	
0	1	1	
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

Producto de sumas

Para estandarizar una función mediante un producto de sumas utilizamos las siguientes leyes del álgebra de Boole.

$$A + BC = (A + B)(A + C)$$

$$A * A = A$$

$$A * \bar{A} = 0$$

$$A * 1 = A$$

A cada término de la función estandarizada se la conoce como el **termino máximo** los mismos que cumplen la función de asignar el dígito “1” a la variable complementada y el dígito “0” a la variable sin complementar.

Ejemplo: Encuentre los términos máximos y la tabla de verdad de la siguiente función, para tal efecto utilizar productos de sumas.

$$F(D, E, F) = \bar{F}D + \bar{F}E + DF$$

Completo cada función con la variable de entrada que le falta.

$$F(D, E, F) = (\bar{F}D + \bar{F}E) + DF$$

$$F(D, E, F) = \bar{F}(D + E) + DF$$

$$F(D, E, F) = [\bar{F}(D + E) + D][\bar{F}(D + E) + F]$$

$$F(D, E, F) = [D + \bar{F}(D + E)][F + \bar{F}(D + E)]$$

$$F(D, E, F) = [(D + \bar{F})(D + (D + E))][(F + \bar{F})(F + (D + E))]$$

$$F(D, E, F) = [(D + \bar{F})(D + D + E)][(F + \bar{F})(F + D + E)]$$

$$F(D, E, F) = [(D + \bar{F})(D + E)][(1)(F + D + E)]$$

$$F(D, E, F) = [(D + \bar{F})(D + E)][F + D + E]$$

$$F(D, E, F) = (D + \bar{F} + E * \bar{E})(D + E + F * \bar{F})(F + D + E)$$

$$F(D, E, F) = (D + \bar{F} + E)(D + \bar{F} + \bar{E})(D + E + F)(D + E + \bar{F})(F + D + E)$$

$$F(D, E, F) = (D + E + \bar{F})(D + \bar{E} + \bar{F})(D + E + F)(D + E + \bar{F})(D + E + F)$$

Términos semejantes

$$F(D, E, F) = (D + E + \bar{F})(D + \bar{E} + \bar{F})(D + E + F)$$

$$F(D, E, F) = (0 \quad 0 \quad 1)(0 \quad 1 \quad 1)(0 \quad 0 \quad 0)$$

$$F(D, E, F) = (M1)(M3)(M0)$$

$$F(D, E, F) = \prod M(0,1,3).$$

Tabla 9

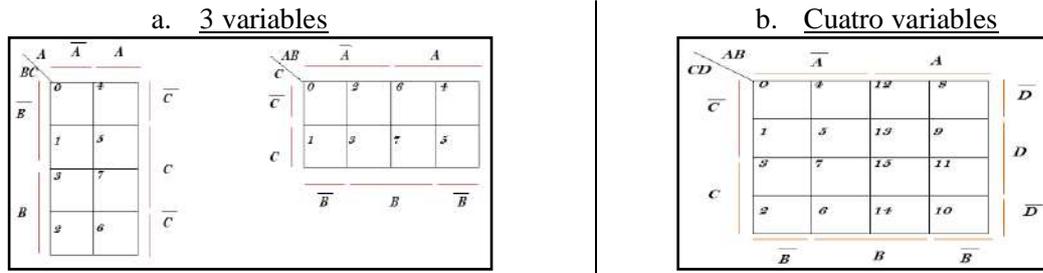
Tabla de verdad ejemplo planteado

<u>D</u>	<u>E</u>	<u>F</u>	<u>F(D,E,F)</u>
0	0	0	0
0	0	1	0
0	1	0	
0	1	1	0
1	0	0	
1	0	1	
1	1	0	
1	1	1	

Mapas de Karnaugh

Son herramientas graficas empleadas para el diseño y simplificación de expresiones lógicas, estos mapas representan cuadros de dos dimensiones, donde las variables booleanas se organizan en filas y columnas, representando cada celda una combinación específica de valores de entrada. En la Figura 4 se observa el mapa de Karnaugh para tres variables y cuatro variables.

Figura 4
Mapa de Karnaugh para 3 y 4 variables



Simplificación de funciones

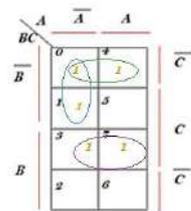
Para simplificar funciones se realiza agrupaciones de los casilleros lógicamente adyacentes en el orden de 2^n ; donde n nos indica el número de variables que se simplifica.

Para que una variable se simplifique en la agrupación de términos adyacentes lógicos deben estar tanto en la variable complementada como en la variable sin complementar.

Ejemplo: Utilizando mapas K realizar la siguiente simplificación de funciones y diseña el circuito lógico de la función simplificada: $F(A, B, C, D) = \sum m(0,1,3,4,7)$, como se muestra en la Figura 5 y Figura 6.

Figura 5

Mapa K $F(A, B, C, D) =$



$\sum m(0, 1, 3, 4, 7)$

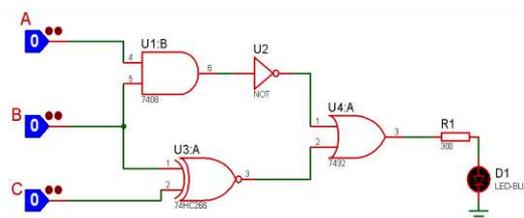
$$F(A, B, C) = \overline{B}\overline{C} + \overline{A}\overline{B} + BC$$

$$\overline{A}\overline{B} + (\overline{B}\overline{C} + BC)$$

$$\overline{A}\overline{B} + (\overline{B} \otimes C)$$

Figura 6

Circuito lógico solución ejemplo planteado



Ejercicios prácticos propuestos

1. Dada la expresión $F(D,E,F,G)=\sum m(1,2,6,9,10,14,15)$ de términos mínimos simplificar la función utilizando los mapas de Karnaugh, también hallar la ecuación lógica simplificada, tabla de verdad y de la función simplificada diseñar el circuito lógico.
2. Dada la expresión $F(H,I,J,K)=\pi M(0,3,4,5,11,15)$ de términos máximos simplificar la función utilizando los mapas de Karnaugh, también hallar la ecuación lógica simplificada, tabla de verdad y de la función simplificada diseñar el circuito lógico.
3. Utilizando mapas K realizar la simplificación del siguiente ejercicio: Dos números con 2 bits cada uno, a la salida obtener la suma de dichos números, obtenga la ecuación de salida, tabular la tabla de verdad, implementa el diseño del circuito lógico en Proteus.
4. Dadas la expresión $F(A, B, C, D) = A\bar{B}C + \bar{A}C + \bar{A}\bar{B}\bar{C}$, determine el circuito lógico, determina su tabla de verdad, presenta el circuito funcionando en protoboard.
5. Diseña un circuito lógico que encienda una salida cada vez que en sus cuatro entradas se produzca un múltiplo de dos, incluir la combinación cero: determina su tabla de verdad, presenta el circuito funcionando en protoboard, considera los mapas K para la simplificación.

UNIDAD 3 FUNCIONES Y CIRCUITOS DE LÓGICA COMBINACIONAL

Decodificadores

Es un circuito que convierte una determinada combinación de n bits de entrada en una salida digital específica de máximo 2^n salidas.

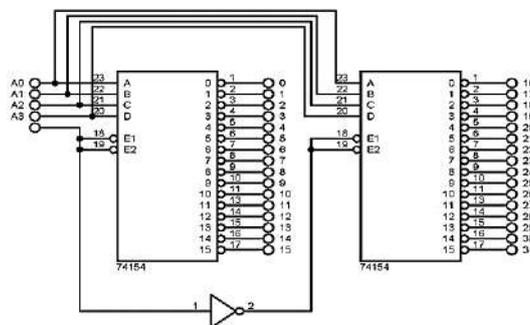
Decodificador de 4 a 16 líneas

El esquema de la **¡Error! No se encuentra el origen de la referencia.** se denomina decodificador de 4 a 16 líneas (binario a decimal), en la cual, para un código binario dado en la entrada se activa una de las 16 líneas de salida. Comercialmente se tiene el CI-74154 que es un decodificador de 4 a 16.

Ejemplo: Se requiere codificar un número de 5 bits usando los CI-74154. El número binario se representa con la combinación $A_4A_3A_2A_1A_0$. El quinto bit A_4 se conecta a las entradas de selección del chip $\overline{CS1}$ y $\overline{CS2}$ de uno de los decodificadores y $\overline{A_4}$ se conecta a las entradas de activación $\overline{CS1}$ y $\overline{CS2}$ del otro decodificador. Con esto cuando el número decimal es menor o igual a 15, A_4 es 0, por lo que el decodificador de menor orden se activa y el otro se desactiva. El circuito Decodificador de 5 a 32 líneas final se observa en la Figura 7:

Figura 7

Decodificador binario a decimal



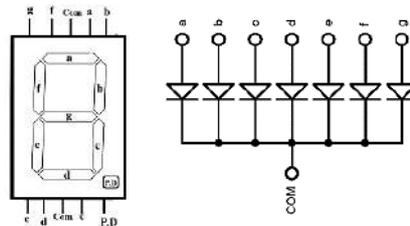
Aplicaciones de los decodificadores

Decodificador BCD de 7 segmentos

Todos los sistemas digitales necesitan mostrar información que sea perceptible por el usuario, generalmente en formato numérico, para tal efecto se utilizan los display de 7 segmentos, los mismos que tienen alfabéticamente etiquetas como a, b, c, d, e, f, g y el punto decimal (P.D), tal como se muestra en la Figura 8. Los tipos de display pueden ser de ánodo común y cátodo común.

Figura 8

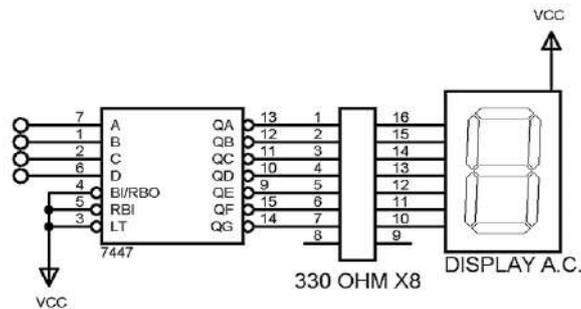
Denominación de los segmentos del display a 7 segmentos



Comercialmente se tienen los CI-7446 o el CI-7447, como son tecnología TTL se recomienda usar resistencias limitadoras cuyas conexiones de muestran en la Figura 9.

Figura 9

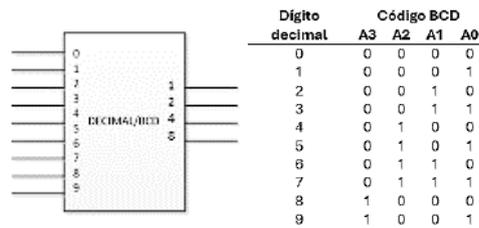
Decodificador BCD a decimal con display ánodo común



Codificador

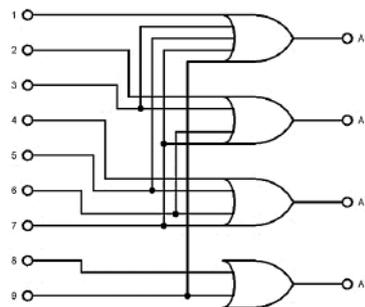
Es un circuito lógico que permite el ingreso en una de sus entradas un nivel lógico que represente un dígito y lo convierte en una salida codificada. Como ejemplo se tiene el codificador decimal a BCD (Figura 10), el cual dispone tiene 10 entradas y cuatro salidas correspondientes al código BCD.

Figura 10
Diagrama y tabla de verdad del Codificador de 10 líneas a 4 líneas Decimal/BCD



La implementación en compuertas OR se muestra en la Figura 11, en la cual se va verificando salida a salida a qué combinación corresponde. El modelo comercial es el CI-74147.

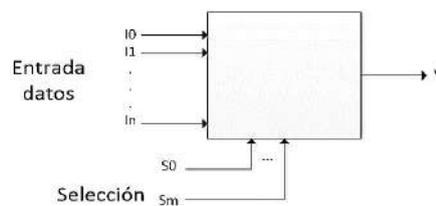
Figura 11
Implementación con compuertas del codificador de 10 a 4 líneas



Multiplexores

Es un circuito lógico que contiene varias líneas de entrada y una única línea de salida y su función es conmutar la información digital a una única línea de salida (selector de datos), la selección para la salida se lo hace por medio de entradas de dirección o control (S_x), ver Figura 12.

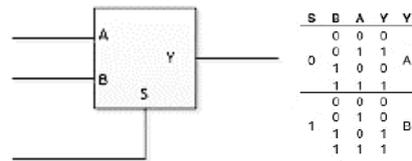
Figura 12
Diagrama de un multiplexor



Multiplexor 2 - 1

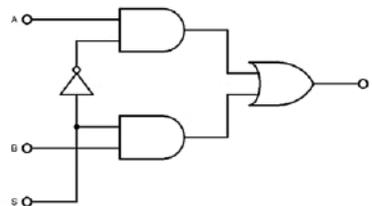
Este multiplexor es el más simple porque tiene una salida, dos entradas y una entrada de selección (S), Figura 13.

Figura 13
Tablas de verdad del Multiplexor 2 - 1



La función de salida $Y = \bar{S}A + SB$, se obtiene de la tabla de verdad, considerando que el circuito esta implementado con compuertas lógicas como se aprecia en la Figura 14.

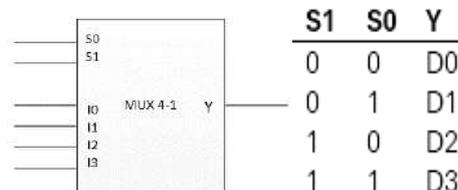
Figura 14
Circuito lógico del Multiplexor 2 - 1



Multiplexor de 4 - 1

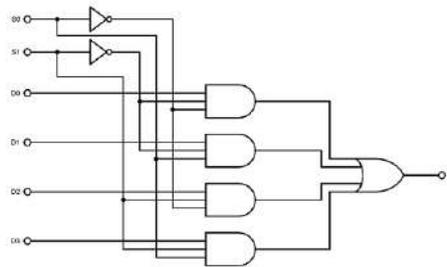
Este multiplexor tiene una salida, 4 entradas y requiere de dos líneas de control (S1 y S0) que son necesarias para seleccionar la entrada de datos que va a la salida, ver Figura 15.

Figura 15
Tabla de verdad del Multiplexor 4 - 1



Por ejemplo, si se aplica un código binario $S1=0$ y $S0=0$, la entrada de datos D0 será la seleccionada para pasar a la salida. De acuerdo con la tabla, cada salida depende de las entradas de selección, por lo que se puede generar la expresión $Y = D_0\bar{S}_1\bar{S}_0 + D_1\bar{S}_1S_0 + D_2S_1\bar{S}_0 + D_3S_1S_0$. A nivel comercial se tienen varias opciones de multiplexores, por ejemplo, el CI 74157 posee 4 MUX de 2 - 1 con una entrada de selección común mientras que el CI 74151 es un MUX de 8 - 1. Por tal razón el circuito con compuertas lógicas se tiene puede visualizar en la Figura 16

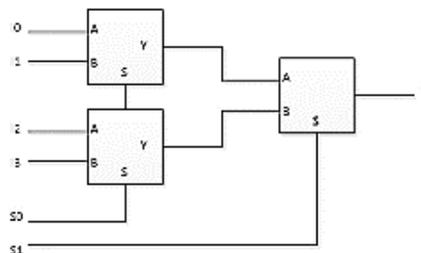
Figura 16
Circuito lógico del Multiplexor 4 - 1



Multiplexores y sus aplicaciones

En la Figura 17 se puede observar el diagrama de bloques de un multiplexor 4 – 1 en función de un multiplexor de 2 – 1.

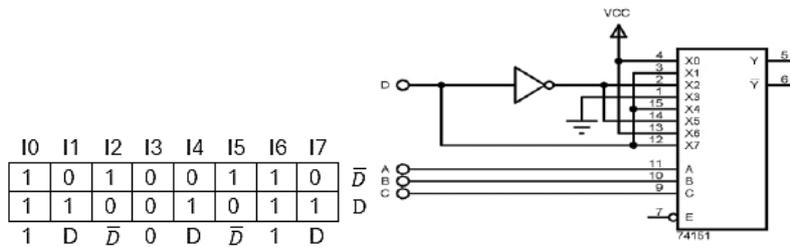
Figura 17
Multiplexor 4 - 1 con multiplexores de 2 – 1



Ejemplo: Implementar la siguiente función booleana $F < D, C, B, A > = \sum m(0,2,5,6,8,9,12,14,15)$.

Si la función tiene n variables, entonces se requiere de un MUX de (n-1) entradas de selección, para la variable faltante se usa la conexión de las entradas del MUX. Para el ejemplo se requiere un MUX 8-1 de 3 entradas de selección, con la función se genera una tabla de 2 filas y tantas columnas tenga el MUX (Figura 18), y se va llenando con la función a implementar empezando desde la fila de arriba y de izquierda a derecha. La fila de arriba corresponde a la variable \bar{D} y la fila de abajo corresponde al valor de la variable D. Si en una columna se tienen dos 1s, debajo se pone un 1, si son dos 0s se pone un 0, en otro caso se pone la variable de la fila correspondiente donde se encuentra el 1. Estos valores de la parte inferior se usan para la implementación de la función, tal como se muestra en la Figura 18, en donde los 1s se ponen a VCC y los 0s a GND.

Figura 18
Diseño e implementación de función booleana con MUX 8 - 1.



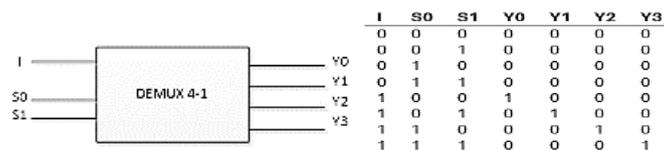
Demultiplexor

Tiene una función inversa al multiplexor, es decir, se tiene una sola línea de entrada de datos y varias líneas de salida de datos, siendo una la seleccionada para sacar los datos de entrada, la selección se hace por medio de líneas de selección.

DEMUX 1 - 4

Como se tienen 4 salidas, se requieren 2 líneas de selección (S0 y S1) ver Figura 19, mientras que en la Figura 20 se observa el circuito lógico implementado.

Figura 19
Tabla de verdad del Demultiplexor 1 - 4



Las funciones para cada salida quedan:

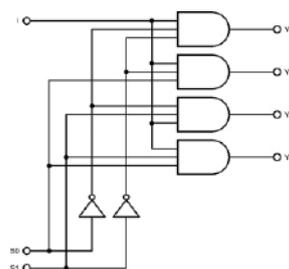
$$Y_0 = \bar{S}_1 \bar{S}_0 I$$

$$Y_1 = \bar{S}_1 S_0 I$$

$$Y_2 = S_1 \bar{S}_0 I$$

$$Y_3 = S_1 S_0 I$$

Figura 20
Circuito lógico del Demultiplexor 1 - 4



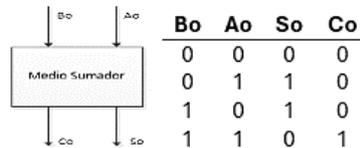
Circuitos digitales sumadores

Diseño del medio sumador

El bloque del medio sumador y la tabla para el diseño se muestra en la Figura 21.

Figura 21

Bloque medio sumador – tabla de verdad



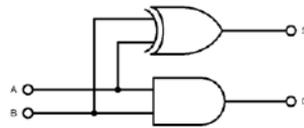
De la tabla de verdad se concluye que:

$$S_o = A_o \oplus B_o$$

$$C_o = A_o \cdot B_o \text{ (ver Figura 22)}$$

Figura 22

Implementación con compuertas de medio sumador

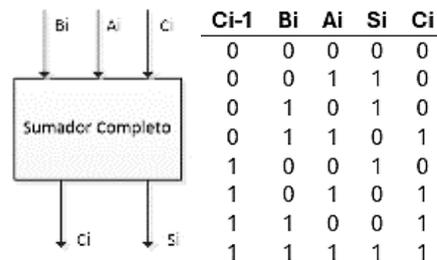


Diseño del sumador completo

El bloque del sumador completo y la tabla para el diseño se muestra en la Figura 23.

Figura 23

Bloque medio sumador – tabla de verdad

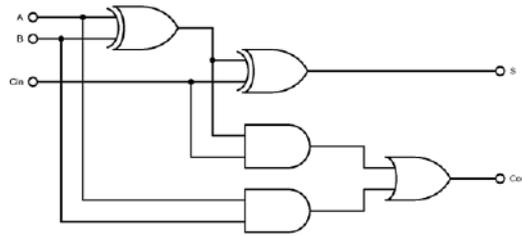


De la tabla se obtiene:

$$S_i = A_i \oplus B_i \oplus C_i$$

$$C_i = \overline{\overline{A_i} \cdot B_i \cdot (\overline{A_i \oplus B_i}) \cdot C_i} \text{ (ver figura 24)}$$

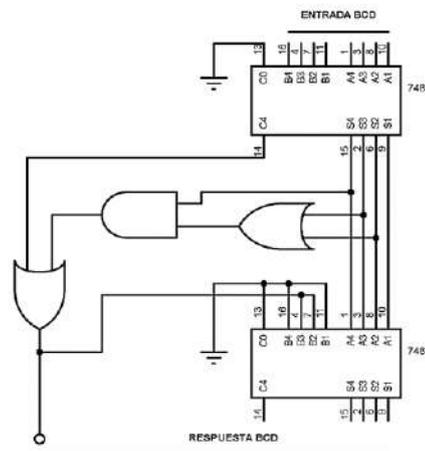
Figura 24
Implementación con compuertas de un sumador completo



Ejemplo sumador BCD: En algunas ocasiones se requiere sumar números en BCD (A y B), sin embargo, el sumador produce únicamente un resultado en binario natural, por tal motivo se requiere realizar un ajuste decimal (AD), el cual consiste en sumar 0 (0000) cuando el resultado de la suma A+B sea menor que 10, y se suma 6 (0110) cuando el resultado sea mayor que 9. La tabla de verdad y la implementación se muestra en la Figura 25. De la tabla se obtiene la función $AD = S_3(S_2 + S_1)C_0$.

Figura 25
Tabla de verdad e implementación de sumador BCD

Co	S3	S2	S1	S0	AD
0	0	0	0	0	0
0	0	0	0	1	0
0	0	0	1	0	0
0	0	0	1	1	0
0	0	1	0	0	0
0	0	1	0	1	0
0	0	1	1	0	0
0	0	1	1	1	0
0	1	0	0	0	0
0	1	0	0	1	0
0	1	0	1	0	1
0	1	0	1	1	1
0	1	1	0	0	1
0	1	1	0	1	1
0	1	1	1	0	1
0	1	1	1	1	1
1	0	0	0	0	1
1	0	0	0	1	1
1	0	0	1	0	1



Nota: El estudiante como ejercicio en clases debe obtener la expresión usando mapas de Karnaugh.

Circuitos digitales comparadores

Tiene la función de comparar magnitudes de dos cantidades binarias con el fin de establecer su relación (si $A > B$, $A = B$ o $A < B$), esto se puede lograr con circuitos lógicos.

Comparador para números de 2 bits

Dado los números $A=A_1A_0$ y $B=B_1B_0$, las salidas $A>B$, $A=B$ o $A<B$ toman el valor de uno lógico cuando se cumple dicha condición y cero lógico en otro caso (Figura 26).

Figura 26
Tabla de verdad de un comparador de 2 bits

Entradas				Salidas		
B1	B0	A1	A0	A>B	A=B	A<B
0	0	0	0	0	1	0
0	0	0	1	1	0	0
0	0	1	0	1	0	0
0	0	1	1	1	0	0
0	1	0	0	0	0	1
0	1	0	1	0	1	0
0	1	1	0	1	0	0
0	1	1	1	1	0	0
1	0	0	0	0	0	1
1	0	0	1	0	0	1
1	0	1	0	0	0	1
1	0	1	1	1	0	0
1	1	0	0	0	0	1
1	1	0	1	0	0	1
1	1	1	0	0	0	1
1	1	1	1	0	1	0

Considerando términos mínimos se tiene:

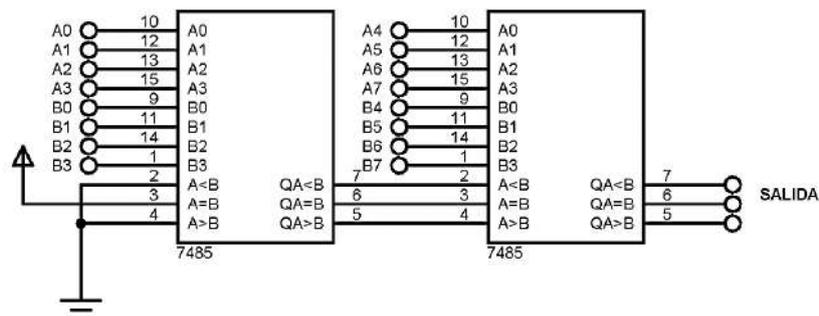
$$\begin{aligned}
 A > B &= \sum m(1,2,3,6,7,11) & A == B &= \sum m(0,5,10,15) & A < B \\
 & & & & = \sum m(4,8,9,12,13,14)
 \end{aligned}$$

Comercialmente se tiene el CI-7485 el cual es un comparador de 4 bits.

Ejemplo: Comparador de dos números de 8 bits usando los CI 7585 en cascada, ver

Figura 27.

Figura 27
Comparador de 8 bits



Ejercicios Propuestos

1. Realizar el esquema de conexiones para un decodificador de BCD a 7 segmentos cátodo común, para lo cual use el CI comercial 7448.
2. Usando un MUX apropiado implementar la función booleana $F < E, D, C, B, A > = \sum m(1,3,4,7,9,10,16,19,22,24,24,26,29,30,31)$
3. Implementar un MUX 8 a 1 empleando MUX 2 a 1.
4. Usar un decodificador del número de entradas adecuado para implementar un circuito que genere las siguientes funciones lógicas $F < D, C, B, A > = \sum m(0,3,6,7,9,12,15)$.

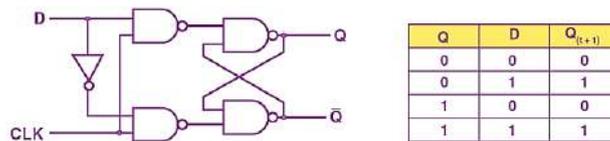
UNIDAD 4 FUNCIONES Y CIRCUITOS DE LÓGICA SECUENCIAL

Flip-Flop D (data)

El Flip - Flop D tiene una entrada de datos (D), una entrada de reloj (CLK) y dos salidas: una salida principal (Q) y una salida complementaria (\bar{Q}). Cuando la señal de reloj pasa de un nivel bajo a uno alto (flanco ascendente), el valor presente en la entrada D se transfiere a la salida Q, ver Figura 28.

Figura 28

Circuito y tabla de funcionamiento del Flip – Flop D

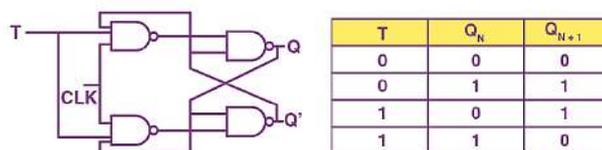


Flip-Flop T

Este tipo de Flip - Flop tiene una entrada de reloj (CLK), una entrada de datos (T) y dos salidas: Q y \bar{Q} . Cuando la señal de reloj pasa de un nivel bajo a uno alto, el Flip-Flop modifica su estado de salida si la entrada T está en nivel alto. Si T está en nivel bajo, el estado de salida permanece igual, ver Figura 29.

Figura 29

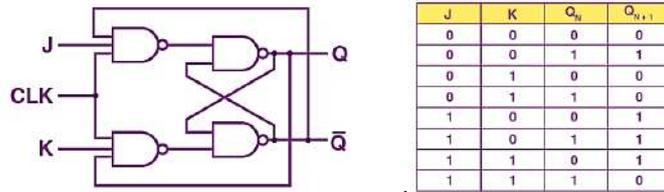
Circuito y tabla de funcionamiento del Flip – Flop T



Flip-Flop JK

Este tipo de Flip – Flop tiene una entrada de reloj (CLK), dos entradas de datos (J y K) y dos salidas: Q y \bar{Q} . Cuando la señal de reloj pasa de un nivel bajo a uno alto, el estado de salida del Flip-Flop se modifica de acuerdo con las entradas J y K siguiendo la tabla de verdad específica del Flip-Flop JK, ver Figura 30.

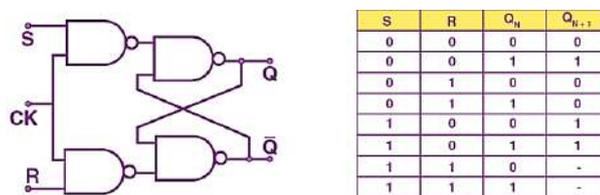
Figura 30
Circuito y tabla de funcionamiento del Flip – Flop JK



Flip-Flop SR

Este Flip - Flop tiene dos entradas de datos (S y R), una entrada de reloj (CLK) y dos salidas: Q y \bar{Q} . Cuando la señal de reloj cambia de bajo a alto, el estado de salida del Flip - Flop cambia según las entradas S y R según la tabla de verdad específica del Flip - Flop SR, ver Figura 31.

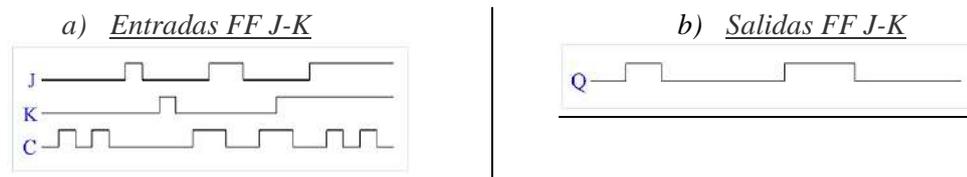
Figura 31
Circuito y tabla de funcionamiento del Flip – Flop SR



Ejercicio Propuesto

Obtenga los diagramas de tiempo a la salida del FF J-K, en las terminales Q y \bar{Q} a partir de los diagramas de tiempo de la Figura 32. El FF J-K está inicialmente en el estado $Q=0$ y los cambios son motivados por las transiciones de "1" a "0" en la terminal C y que en el estado $J=1$ y $K=1$ la terminal C no ejerce ninguna acción.

Figura 32
Diagramas de tiempo Flip – Flop



Solución: Suponiendo que el Flip - Flop J-K está inicialmente en el estado $Q=0$, la transición de "0" a "1" en la terminal C no produce efecto alguno, pero la caída subsecuente de "1" a "0" hace que el Flip - Flop J-K cambie de estado, pasando de $Q=0$ a $Q=1$. La siguiente transición negativa de "1" a "0" en la terminal C hace que la salida Q vuelva a cambiar de estado, pasando de $Q=1$ a $Q=0$. Tras esto, aparecen unos "pulsos" en las terminales J y K, pero estos no tienen efecto alguno al permanecer la terminal C en un nivel estático. Tras ascender nuevamente la terminal C de "0" a "1", hay un cambio en el valor de entrada en la terminal J, que es puesta a $J=1$. Esto prepara al Flip - Flop para que cuando haya una transición negativa de "1" a "0" en la terminal C el Flip - Flop J-K pase al estado $Q=1$ al tener $J=1$ y $K=0$ en sus terminales de entrada, que es precisamente lo que ocurre al final del tercer "pulso" en la terminal C. Tras esto, después de que la entrada en la terminal J cae de "1" a "0" sin producir cambio alguno puesto que los cambios los produce únicamente la terminal C, la terminal C sube de "0" a "1" sin producir todavía cambio alguno en la salida, y permanece en "1" mientras el valor en la terminal K sube de "0" a "1". Esto prepara al Flip - Flop para que cuando haya una transición negativa de "1" a "0" en la terminal C el Flip - flop J-K pase al estado $Q=0$ al tener $J=0$ y $K=1$ en sus terminales de entrada, que es precisamente lo que ocurre al final del cuarto "pulso" en la terminal C. Tras esto, la entrada J es elevada de "0" a "1", con lo cual ambas entradas J y K tienen el valor de "1", lo cual tiene como consecuencia directa que cuando ocurran los siguientes dos "pulsos" en la señal de "reloj" C el Flip - Flop J-K no cambie de estado.

Contadores

Contadores Síncronos

En este tipo de contador todos los Flip-Flops cambian de estado al mismo tiempo en respuesta a la señal de reloj. Estos Flip-Flops están conectados en cascada, de modo que la salida de un Flip-Flop se utiliza como entrada de reloj del siguiente. Por esta razón este tipo de contadores están sincronizados con una señal de reloj común.

Contadores Asíncronos

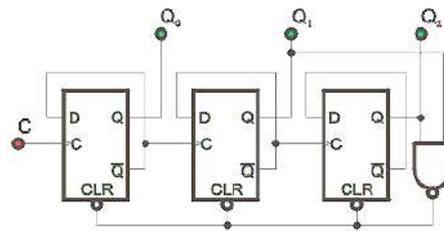
En un contador asíncrono, los Flip-Flops cambian de estado de manera independiente, sin depender de una señal de reloj común. Están conectados de forma que el cambio de estado de un Flip-Flop influye en el siguiente, creando una reacción en cadena.

Ejercicio Propuesto

Dado el siguiente circuito secuencial asíncrono construido a base de Flip - Flop D, obtener la tabla de secuencias para dicho circuito suponiendo que los Flip - Flop D utilizados son del tipo en los cuales los cambios de estado ocurren en la transición positiva (de "0" a "1") del pulso a la entrada de la terminal de reloj, ver Figura 33.

Figura 33

Diagrama secuencial Flip Flop D para ejercicio



Solución

Supondremos, que el estado inicial del circuito es $Q_0Q_1Q_2=000$.

Al inicio, durante la primera transición, con dos ceros en la puerta NAND, $Q_1Q_2=00$; esto genera un "1" en su salida, que es invertido por la puerta NOT en la terminal Clear (CLR) de los tres Flip-Flops D, por lo tanto, no afecta a ninguno de ellos. El primer Flip-Flop D cambia su estado de $Q_0=0$ a $Q_0=1$ porque su entrada está alimentada con su salida complementaria $Q_0=1$ (en efecto, el primer Flip-Flop se comportará todo el tiempo como un Flip-Flop T). Este cambio hace que Q_0 pase de "1" a "0", sin afectar al segundo Flip-Flop, ya que solo las transiciones de "0" a "1" en la terminal C provocan un cambio. Por lo tanto, el segundo Flip-Flop permanece en su estado $Q_1=0$, y lo mismo ocurre con el tercer Flip-Flop.

Durante la segunda transición de estados, el primer Flip-Flop, cuya entrada D siempre está conectada a su salida complementaria Q_0 , cambia su salida de $Q_0=1$ a $Q_0=0$. Sin embargo, el segundo Flip-Flop, al estar conectado a Q_0 , detecta el cambio de $Q_0=0$ a $Q_0=1$, provocando así un cambio de estado en el segundo Flip-Flop.

Manteniendo un análisis semejante, acumulamos la siguiente información.

Primera transición: El estado es $Q_0Q_1Q_2=100$.

Segunda transición: El estado es $Q_0Q_1Q_2=0100$.

Tercera transición: El estado es $Q_0Q_1Q_2=110$.

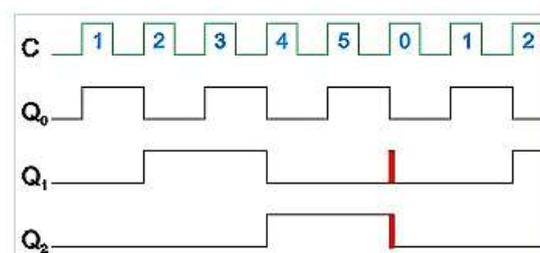
Cuarta transición: El estado es $Q_0Q_1Q_2=001$.

Quinta transición: El estado es $Q_0Q_1Q_2=101$.

Al llegar a la sexta transición, tenemos la situación $Q_1Q_2=11$, lo cual hace que la salida del NAND cambie de "1" a "0", y este "0" al ser invertido por las burbujas inversoras NOT en las terminales CLR de los tres Flip - Flop "limpia" al circuito regresándolo al estado $Q_0Q_1Q_2=000$. Esto se ve más claramente en el diagrama de la Figura 34.

Figura 34

Diagrama de tiempo de salidas para ejercicio



Todo el comportamiento del circuito se puede resumir entonces en la tabla 10

Tabla 10
Tabla de secuencias del ejercicio propuesto

<u>Q0</u>	<u>Q1</u>	<u>Q2</u>
<u>0</u>	<u>0</u>	<u>0</u>
<u>1</u>	<u>0</u>	<u>0</u>
<u>0</u>	<u>1</u>	<u>0</u>
<u>1</u>	<u>1</u>	<u>0</u>
<u>0</u>	<u>0</u>	<u>1</u>
<u>1</u>	<u>0</u>	<u>1</u>

Registro de desplazamiento

Registro de Desplazamiento Simple

Este tipo de registro realiza un desplazamiento de bits en una sola dirección, ya sea hacia la derecha o hacia la izquierda. En una operación de desplazamiento hacia la derecha, los bits en el extremo derecho se eliminan y se agrega un nuevo bit en el extremo izquierdo. Del mismo modo, en un desplazamiento hacia la izquierda, los bits en el extremo izquierdo se eliminan y se introduce un nuevo bit en el extremo derecho.

Ejemplo práctico: Si tenemos un registro de desplazamiento de 4 bits con el contenido inicial 1101, al realizar un desplazamiento a la derecha, el resultado sería 1110. Durante este proceso, el bit más a la derecha se elimina del registro y se introduce un nuevo bit en el extremo izquierdo.

Registro de Desplazamiento de Ciclo

Este registro realiza un desplazamiento de bits hacia la derecha o hacia la izquierda, y los bits que "salen" se vuelven a insertar en el extremo opuesto del registro. Es decir, los bits que se desplazan más allá de un extremo se recirculan al otro extremo.

Ejemplo práctico: Continuando con el ejemplo anterior y suponiendo que se tiene un registro de desplazamiento de ciclo de 4 bits, si se realiza un desplazamiento a la derecha en **1101**, se obtendría **1110** con el bit **1** que salió recirculado al extremo izquierdo.

Registro de Desplazamiento de Carga/Almacenamiento

Este tipo de registro permite cargar o almacenar datos en una posición específica del registro. Es útil cuando se necesita acceder a datos en ubicaciones específicas dentro del registro.

Ejemplo práctico: Suponiendo que se quiere cargar el valor **1010** en el tercer bit de un registro de desplazamiento de 4 bits. Utilizando un registro de desplazamiento de carga/almacenamiento, se podría colocar **1010** en la posición deseada sin afectar los otros bits.

Registro de Desplazamiento Universal

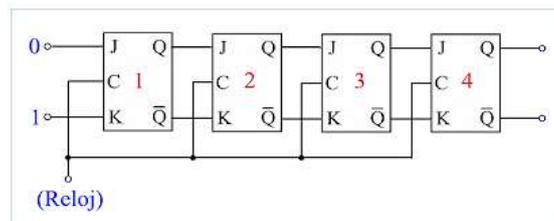
Este tipo de registro puede realizar cualquier tipo de desplazamiento (simple o cíclico) y también incluye otras operaciones como la rotación y la carga/almacenamiento condicional. Es altamente versátil y puede adaptarse a una variedad de aplicaciones.

Ejemplo práctico: Si se tiene un sistema en el que se necesita realizar operaciones de rotación en datos almacenados en un registro. Utilizando un registro de desplazamiento universal, se podría rotar los datos hacia la izquierda o hacia la derecha según sea necesario.

Ejercicio Propuesto

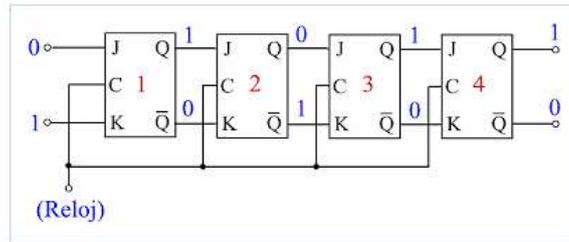
Analizar la siguiente configuración suponiendo que ésta recibe un tren simétrico de pulsos en la terminal de entrada. Supóngase que las entradas J y K del primer Flip - Flop mantienen un valor constante de "0" y "1" respectivamente. También, si el circuito comienza en el estado $Q_1Q_2Q_3Q_4=1011$, se debe elaborar una tabla de secuencias para la serie de eventos representados en la Figura 35.

Figura 35
Circuito con Flip Flop J-K



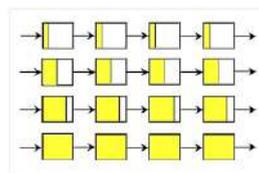
Con la condición inicial $Q_1Q_2Q_3Q_4=1011$, se observa el inicio de lo representado en la figura 36

Figura 36
Circuito con Flip Flop J-K y condición inicial



Las entradas en las terminales J y K de cada Flip Flop son inversas entre sí. En este caso, Q adoptará el valor de J después de que ocurra una transición de "1" a "0" en cada Flip Flop. En la primera transición, el Flip Flop 4 toma el valor Q_3 a la salida del Flip Flop 3, produciendo $Q_4=1$, el Flip Flop 3 toma el valor del Flip Flop 2 produciendo $Q_3=0$, el Flip Flop 2 toma el valor del Flip - Flop 1 produciendo $Q_2=1$ y el Flip Flop 1 toma el valor de "0" que está puesto en su entrada J produciendo $Q_1=0$. Este flujo de eventos es viable gracias a que cada Flip-Flop tiene un retardo interno que "retiene" la información el tiempo necesario después de la transición, permitiendo que la información en su salida pueda ser capturada por el Flip-Flop siguiente. Esta idea se ilustra en la Figura 37, donde se visualiza el recorrido de la información a través de cada Flip-Flop (inmediatamente después de comenzar una transición) representado por el área coloreada en amarillo.

Figura 37
Recorrido de información del circuito con Flip Flop J-K



El mecanismo mostrado anteriormente ilustra de manera básica cómo un Flip-Flop puede "leer" la información de otro Flip-Flop que le antecede, asegurando al mismo tiempo que la información se retenga en su salida el tiempo necesario para que el siguiente Flip-Flop la pueda procesar. Basándonos en esto, podemos proceder con el análisis para observar los eventos posteriores a cada transición, tal como se indica en la Tabla 11.

Tabla 11
Proceso de transiciones, condiciones y esquema de conexión del ejercicio planteado

<u>Transiciones y condiciones</u>	<u>Esquema de conexión</u>
<p><u>Primera transición:</u></p> <p>La condición es $Q_1Q_2Q_3Q_4=0101$.</p>	
<p><u>Segunda transición:</u></p> <p>La condición es $Q_1Q_2Q_3Q_4=0010$.</p>	
<p><u>Tercera transición:</u></p> <p>La condición es $Q_1Q_2Q_3Q_4=0001$.</p>	
<p><u>Cuarta transición:</u></p> <p>La condición es $Q_1Q_2Q_3Q_4=0000$.</p>	

La quinta transición y las siguientes no lograrán cambiar el estado del circuito desde $Q_1Q_2Q_3Q_4=0000$, por lo que en este punto concluimos el análisis. La tabla de secuencias para cada transición sucesiva, comenzando con la condición inicial $Q_1Q_2Q_3Q_4=1011$, se basa en los resultados obtenidos y se muestra a continuación: Después de cada transición, observamos que la palabra almacenada originalmente se desplaza un bit hacia la derecha y es enviada al exterior por el último Flip-Flop (Flip-Flop 4). Este tipo de circuitos se conoce como registro de desplazamiento o registro de corrimiento, y tiene aplicaciones significativas en los circuitos lógicos. En este caso, el registro de desplazamiento se utilizó para leer una palabra almacenada de forma serial, es decir, bit por bit. Además, el registro de desplazamiento puede usarse para escribir una palabra, introduciéndola bit por bit en el primer Flip-Flop (Flip-Flop 1), para su posterior uso.

Máquinas de estado

Máquinas de Estado Finito Determinista (MEFD)

Las Máquinas de Estado Finito Determinista son modelos en los que el estado siguiente de la máquina está determinado de forma única por el estado actual y el evento que ocurre. Cada estado tiene transiciones salientes etiquetadas con eventos, que especifican qué evento debe ocurrir para realizar la transición a un nuevo estado.

Ejemplo: Consideremos una máquina expendedora de bebidas. Los estados podrían ser "esperando moneda", "seleccionando bebida", "entregando bebida", y "fuera de servicio". Los eventos podrían ser "moneda insertada", "bebida seleccionada", "bebida entregada", y "reinicio".

Estados: esperando moneda, seleccionando bebida, entregando bebida, fuera de servicio.

Transiciones:

Esperando moneda -> Moneda insertada -> Seleccionando bebida

Seleccionando bebida -> Bebida seleccionada -> Entregando bebida

Entregando bebida -> Bebida entregada -> Esperando moneda

Cualquier estado -> Reinicio -> Fuera de servicio

Máquinas de Estado Finito No Determinista (MEFND)

En las Máquinas de Estado Finito No Determinista, una entrada puede llevar a múltiples transiciones posibles desde un estado dado. Estas transiciones pueden ser condicionales y depender de la entrada actual, lo que significa que puede haber múltiples opciones de transición desde un mismo estado.

Ejemplo: Supongamos una máquina de búsqueda de texto en un editor de texto. Los estados podrían ser "esperando entrada", "buscando", y "mostrando resultados". Los eventos podrían ser "tecla presionada" y "búsqueda finalizada".

Estados: esperando entrada, buscando, mostrando resultados.

Transiciones:

Esperando entrada -> Tecla presionada -> Buscando

Buscando -> Búsqueda finalizada -> Mostrando resultados

Máquinas de Estado Finito con Salida (MEFS)

En las Máquinas de Estado Finito con Salida, además de cambiar de estado, la máquina produce una salida basada en el estado actual y el evento de transición. La salida puede ser cualquier tipo de información, desde un simple mensaje hasta un valor calculado.

Ejemplo: Consideremos una máquina de control de semáforos. Los estados podrían ser "verde", "amarillo", y "rojo". Los eventos podrían ser "tiempo transcurrido" y "botón de cambio".

Estados: verde, amarillo, rojo.

Transiciones:

Verde -> Tiempo transcurrido -> Amarillo (y produce la salida "cambiar a amarillo")

Amarillo -> Tiempo transcurrido -> Rojo (y produce la salida "cambiar a rojo")

Rojo -> Tiempo transcurrido -> Verde (y produce la salida "cambiar a verde")

Cualquier estado -> Botón de cambio -> Rojo (y produce la salida "cambiar a rojo")

Autoevaluación

En el siguiente apartado se muestra la evaluación de la materia, escoja la respuesta correcta según corresponda.

<p>1. ¿Cuál es la base del sistema binario? A) 8 B) 2 C) 10 D) 16</p>	<p>2. ¿Qué operación realiza la compuerta lógica AND? A) Produce 1 solo si todas las entradas son 1 B) Produce 1 solo si alguna entrada es 1 C) Invierte el valor de la entrada D) Produce 1 solo si todas las entradas son 0</p>
<p>3. ¿Cuál es el complemento A2 de 1101? A) 0010 B) 0100 C) 1110 D) 1011</p>	<p>4. ¿Qué se utiliza para convertir un número decimal a binario? A) Método de sustitución B) Método de divisiones y multiplicación sucesivas C) Método de agregación D) Método de sustracción</p>
<p>5. ¿Cuál es el código BCD del número decimal 9? A) 1001 B) 0110 C) 1010 D) 1111</p>	<p>6. ¿Cuál de las siguientes es una característica del contador síncrono? A) Los Flip - Flop cambian de estado de forma independiente B) Los Flip - Flop están sincronizados con una señal de reloj común C) Tienen menor consumo de energía que los contadores asíncronos D) No requieren señal de reloj</p>
<p>7. ¿Qué tipo de Flip - Flop tiene una entrada de datos (D) y copia su valor a la salida (Q) en el flanco de subida del reloj? A) Flip-Flop T B) Flip-Flop JK C) Flip-Flop D D) Flip-Flop SR</p>	<p>8. ¿Qué dispositivo convierte un código binario de entrada en una señal de salida específica, comúnmente usado para controlar displays de 7 segmentos? A) Multiplexor B) Decodificador C) Comparador D) Sumador</p>
<p>9. ¿Cuál es la salida de una compuerta OR si las entradas son 0 y 1? A) 0 B) 1 C) -1 D) 2</p>	<p>10. ¿Qué representa una máquina de estado finito determinista (MEFD)? A) Un sistema donde el estado siguiente está determinado de forma única por el estado actual y el evento que ocurre B) Un sistema donde una entrada puede llevar a múltiples transiciones posibles C) Un sistema que no produce ninguna salida D) Un sistema que nunca cambia de estado</p>
<p>RESPUESTAS</p>	
<p>1. <u>B</u> 2. <u>A</u> 3. <u>B</u> 4. <u>B</u> 5. <u>A</u></p>	<p>6. <u>B</u> 7. <u>C</u> 8. <u>B</u> 9. <u>B</u> 10. <u>A</u></p>

Prueba de diagnóstico

Para tener un buen rendimiento en el desarrollo de la asignatura Electrónica Digital, es importante el deseo del estudiante para aprender, pero también son requeridos algunos conocimientos previos; a través de este cuestionario, el estudiante puede poner a prueba este dominio.

11. ¿Qué es un bit?

- A) Un sistema de numeración.
- B) Un componente electrónico.
- C) La unidad básica de información en informática.
- D) Un tipo de señal analógica.

13. ¿Qué operación lógica realiza una puerta AND?

- A) La salida es 1 si cualquiera de las entradas es 1.
- B) La salida es 1 solo si todas las entradas son 1.
- C) La salida es 1 si ninguna de las entradas es 1.
- D) La salida es 1 si una y solo una de las entradas es 1.

15. ¿Qué representa el término "Flip - Flop" en electrónica digital?

- A) Un tipo de puerta lógica.
- B) Un contador de pulsos.
- C) Un circuito secuencial de almacenamiento.
- D) Un convertidor analógico a digital.

17. ¿Cuál es la principal diferencia entre la lógica combinacional y la lógica secuencial?

- A) La lógica combinacional no utiliza memoria.
- B) La lógica secuencial no utiliza puertas lógicas.
- C) La lógica combinacional es más rápida que la secuencial.
- D) La lógica secuencial no se usa en electrónica digital.

19. ¿Cuál es la función principal de un multiplexor en un circuito digital?

- A) Convertir una señal digital en analógica.
- B) Almacenar datos.
- C) Seleccionar una de varias entradas para enviarla a la salida.
- D) Realizar operaciones aritméticas.

12. ¿Cuál de los siguientes sistemas de numeración es utilizado en la electrónica digital?

- A) Decimal
- B) Binario
- C) Hexadecimal
- D) Todos los anteriores

14. ¿Cuál es el resultado de la suma binaria $1 + 1$?

- A) 0
- B) 1
- C) 10
- D) 11

16. ¿Qué es un bus en un sistema digital?

- A) Un tipo de memoria.
- B) Un grupo de cables que transfieren datos entre componentes.
- C) Un software de control.
- D) Un sistema de alimentación.

18. ¿Qué es un byte?

- A) Un único bit.
- B) Un conjunto de 8 bits.
- C) Un conjunto de 16 bits.
- D) Un grupo de señales analógicas.

20. ¿Qué tipo de componente es un transistor en la electrónica digital?

- A) Un amplificador de señal.
- B) Un interruptor controlado electrónicamente.
- C) Un componente pasivo.
- D) Un tipo de memoria.

Referencias bibliográficas

- Brown, S., & Vranesic, Z. (2009). *Fundamentals of digital logic with VHDL design* (3rd ed.). McGraw-Hill Education.
- Floyd, T. L. (2006). *Fundamentos de sistemas digitales* (9.a ed.). Ciudad de México: Pearson Educación.
- Mano, M. M., & Ciletti, M. D. (2013). *Digital design: With an introduction to the Verilog HDL, VHDL, and SystemVerilog* (6th ed.). Pearson.
- MCUs, D. (2023). *Circuitos integrados. Blog by Unit electronics*.
<https://blog.uelectronics.com/electronica/circuitos-integrados-compuertas-logicas-and-or-nand-xor-y-not/>
- Ndjountche, T. (2020). *Electrónica digital 3: Máquinas con números de estados finitos* (COLECCIÓN ELECTRÓNICA, Vol. III, No. 20).
- Tocci, R. J., Widmer, N. S., & Moss, G. L. (2016). *Digital systems: Principles and applications* (12th ed.). Pearson.
- Wakerly, J. F. (2017). *Digital design: Principles and practices* (5th ed.). Pearson.

SUCRE



ISBN: 978-9942-676-26-9



 SUCREInstitutooficial  @SUCREInstituto  @SUCREInstituto