



Universidad de San Carlos de Guatemala  
Facultad de Ingeniería  
Escuela de Ingeniería Mecánica Eléctrica

**VIRTUALIZACIÓN DE CONCEPTOS TEÓRICOS DEL CURSO ELECTRÓNICA 3,  
IMPARTIDO EN LA ESCUELA DE MECÁNICA ELÉCTRICA, FACULTAD DE INGENIERÍA,  
UNIVERSIDAD DE SAN CARLOS DE GUATEMALA**

**Jorge Mario Cardona Oliva**

Asesorado por el Ing. Carlos Eduardo Guzmán Salazar

Guatemala, noviembre de 2018

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



FACULTAD DE INGENIERÍA

**VIRTUALIZACIÓN DE CONCEPTOS TEÓRICOS DEL CURSO  
ELECTRÓNICA 3, IMPARTIDO EN LA ESCUELA DE MECÁNICA  
ELÉCTRICA, FACULTAD DE INGENIERÍA, UNIVERSIDAD DE SAN CARLOS  
DE GUATEMALA**

TRABAJO DE GRADUACIÓN

PRESENTADO A LA JUNTA DIRECTIVA DE LA  
FACULTAD DE INGENIERÍA  
POR

**JORGE MARIO CARDONA OLIVA**

ASESORADO POR EL ING. CARLOS EDUARDO GUZMAN SALAZAR

AL CONFERÍRSELE EL TÍTULO DE

**INGENIERO ELECTRÓNICO**

GUATEMALA, NOVIEMBRE DE 2018

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA  
FACULTAD DE INGENIERÍA



**NÓMINA DE JUNTA DIRECTIVA**

DECANO	Ing. Pedro Antonio Aguilar Polanco
VOCAL I	Ing. Angel Roberto Sic García
VOCAL II	Ing. Pablo Christian de León Rodríguez
VOCAL III	Ing. José Milton de León Bran
VOCAL IV	Br. Oscar Humberto Galicia Nuñez
VOCAL V	Br. Carlos Enrique Gómez Donis
SECRETARIA	Inga. Lesbia Magalí Herrera López

**TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO**

DECANO	Ing. Pedro Antonio Aguilar Polanco
EXAMINADOR	Ing. José Aníbal Silva de los Ángeles
EXAMINADOR	Ing. Luis Eduardo Durán Córdova
EXAMINADOR	Ing. Julio Cesar Solares Peñate
SECRETARIA	Inga. Lesbia Magalí Herrera López

## **HONORABLE TRIBUNAL EXAMINADOR**

En cumplimiento con los preceptos que establece la ley de la Universidad de San Carlos de Guatemala, presento a su consideración mi trabajo de graduación titulado:

**VIRTUALIZACIÓN DE CONCEPTOS TEÓRICOS DEL CURSO  
ELECTRÓNICA 3, IMPARTIDO EN LA ESCUELA DE MECÁNICA  
ELECTRICA, FACULTAD DE INGENIERÍA, UNIVERSIDAD DE SAN CARLOS  
DE GUATEMALA**

Tema que me fuera asignado por la Dirección de la Escuela de Ingeniería Mecánica Eléctrica, con fecha 18 de noviembre 2016.

**Jorge Mario Cardona Oliva**

Guatemala, 5 de julio de 2018

Ingeniero

**Julio César Solares Peñate**

Coordinador Área de Electrónica

Escuela de Ingeniería Mecánica Eléctrica

Facultad de Ingeniería

Universidad de San Carlos de Guatemala

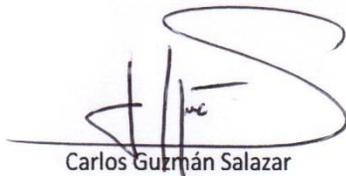
Estimado ingeniero Solares:

Hago de su conocimiento que he concluido la revisión del trabajo de graduación del estudiante **Jorge Mario Cardona Oliva**, titulado: **Virtualización de conceptos teóricos del curso Electrónica 3, impartido en la Escuela de Mecánica Eléctrica, Facultad de Ingeniería, Universidad de San Carlos de Guatemala.**

Considerando que el mismo ha cumplido a cabalidad con los objetivos que se plantearon para su elaboración. Por lo que, apruebo el contenido del mismo, pudiendo el estudiante Cardona Oliva continuar con el trámite que corresponda para culminar su carrera universitaria.

Así mismo, indico que tanto el estudiante Cardona Oliva, como el suscrito, somos enteramente responsables del contenido del trabajo de graduación.

Reciba un cordial saludo,



Carlos Guzmán Salazar

ASESOR





FACULTAD DE INGENIERIA

REF. EIME 56. 2018.  
11 DE JULIO 2018.

Señor Director  
Ing. Otto Fernando Andriano González  
Escuela de Ingeniería Mecánica Eléctrica  
Facultad de Ingeniería, USAC.

Señor Director:

Me permito dar aprobación al trabajo de Graduación titulado: **VIRTUALIZACIÓN DE CONCEPTOS TEÓRICOS DEL CURSO ELECTRÓNICA 3, IMPARTIDO EN LA ESCUELA DE MECÁNICA ELÉCTRICA, FACULTAD DE INGENIERÍA, UNIVERSIDAD DE SAN CARLOS DE GUATEMALA** del estudiante; **Jorge Mario Cardona Oliva**, que cumple con los requisitos establecidos para tal fin.

Sin otro particular, aprovecho la oportunidad para saludarle.

Atentamente,  
**ID Y ENSEÑAD A TODOS**

  
Ing. Julio César Solares Peñate  
Coordinador de Electrónica





REF. EIME 56. 2018.

El Director de la Escuela de Ingeniería Mecánica Eléctrica, después de conocer el dictamen el Asesor, con el Visto Bueno del Coordinador de Área, al trabajo de Graduación del estudiante: **JORGE MARIO CARDONA OLIVA** titulado: **VIRTUALIZACIÓN DE CONCEPTOS TEÓRICOS DEL CURSO ELECTRÓNICA 3, IMPARTIDO EN LA ESCUELA DE MECÁNICA ELÉCTRICA, FACULTAD DE INGENIERÍA, UNIVERSIDAD DE SAN CARLOS DE GUATEMALA,** procede a la autorización del mismo.

  
Ing. Otto Fernando Andriño González



GUATEMALA, 10 DE AGOSTO 2018.

Universidad de San Carlos  
de Guatemala

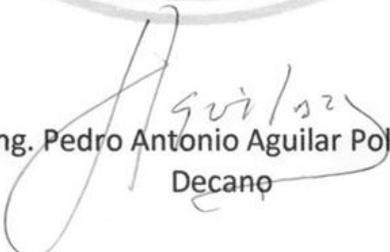


Facultad de Ingeniería  
Decanato

DTG. 498.2018

El Decano de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer la aprobación por parte del Director de la Escuela de Ingeniería Mecánica Eléctrica, al Trabajo de Graduación titulado: **VIRTUALIZACIÓN DE CONCEPTOS TEÓRICOS DEL CURSO DE ELECTRÓNICA 3, IMPARTIDO EN LA ESCUELA DE MECÁNICA ELÉCTRICA, FACULTAD DE INGENIERÍA, UNIVERSIDAD DE SAN CARLOS DE GUATEMALA**, presentado por el estudiante universitario: **Jorge Mario Cardona Oliva**, y después de haber culminado las revisiones previas bajo la responsabilidad de las instancias correspondientes, autoriza la impresión del mismo.

IMPRÍMASE:

  
Ing. Pedro Antonio Aguilar Polanco  
Decano

Guatemala, noviembre de 2018

/gdech



## **ACTO QUE DEDICO A:**

- Dios** Por ser una fuente infinita de sabiduría, guiar mi camino y darme la fuerza en los momentos de flaqueza.
- Mis padres** Hugo Cardona Rojas y María Amparo Oliva Muralles, por el apoyo incondicional, el amor, la paciencia y esfuerzo que han puesto para que esta meta se cumpla. No me alcanzará la vida para agradecerse los.
- Mis hermanos** Kevin Roberto y Hugo Estuardo, por ser esos pilares en los cuales puedo apoyar, por los consejos que se me brindaron y porque siempre creyeron en mí.
- A mi familia** Por brindarme siempre su comprensión y apoyo a lo largo de esta etapa académica de mi vida.
- A mis amigos** Por el cariño que siempre me dieron.

## **AGRADECIMIENTOS A:**

**Universidad de San  
Carlos de Guatemala**

Por haberse convertido en mi casa de estudio e inculcarme un sentido social en favor del pueblo guatemalteco.

**Faculta de Ingeniería**

Por brindarme la oportunidad de alcanzar este preciado éxito

## ÍNDICE GENERAL

ÍNDICE DE ILUSTRACIONES.....	XIII
LISTA DE SÍMBOLOS.....	XXIII
GLOSARIO.....	XXV
RESUMEN.....	XXVII
OBJETIVOS .....	XXIX
INTRODUCCIÓN.....	XXXI
1. MARCO TEÓRICO.....	1
1.1. Aprendizaje.....	1
1.1.1. Teoría conductista .....	2
1.1.2. Teoría cognitivista.....	2
1.1.3. Teoría sociocultural.....	2
1.1.4. Teoría constructivista .....	3
1.2. Método científico .....	4
1.3. Positivismo .....	5
1.4. Constructivismo .....	5
1.4.1. Teorías constructivistas .....	5
1.4.2. Características del constructivismo .....	6
1.4.3. Roles de aprendizaje en el constructivismo .....	7
1.4.3.1. Rol del estudiante .....	7
1.4.3.2. Rol del catedrático .....	7
1.4.4. El constructivismo en la educación superior.....	9
1.5. Positivismo frente a constructivismo .....	10
1.6. Tecnologías de la información y comunicación (TIC).....	11
1.6.1. Características de las TIC .....	12

1.6.2.	Desventajas de las TIC.....	13
2.	CONCEPTOS TEÓRICOS DEL CURSO ELECTRÓNICA 3.....	15
2.1.	Sistemas de numeración.....	15
2.1.1.	Reseña histórica.....	15
2.1.2.	Sistemas de numeración importantes en la electrónica digital.....	16
2.1.2.1.	Sistema de numeración decimal .....	16
2.1.2.2.	Sistema de numeración binario.....	18
2.1.2.3.	Sistema de numeración octal.....	19
2.1.2.4.	Sistema de numeración hexadecimal ..	19
2.1.3.	Conversión de base numérica .....	21
2.1.3.1.	Conversión de base r a decimal.....	21
2.1.3.1.1.	Conversión de binario a decimal .....	22
2.1.3.1.2.	Conversión de octal a decimal .....	22
2.1.3.1.3.	Conversión de hexadecimal a decimal .....	23
2.1.3.2.	Conversión de decimal a base r.....	23
2.1.3.2.1.	Conversión de decimal a binario.....	24
2.1.3.2.2.	Conversión de decimal a octal.....	25
2.1.3.2.3.	Conversión de decimal a hexadecimal .....	27

2.1.3.3.	Relación de números binarios, octales y hexadecimales .....	28
2.1.3.3.1.	Binario a octal .....	28
2.1.3.3.2.	Binario a hexadecimal ...	30
2.1.3.3.3.	Octal y hexadecimal a binario .....	32
2.1.4.	Operaciones aritméticas de números binario, octales y hexadecimales .....	33
2.1.4.1.	Aritmética binaria .....	33
2.1.4.1.1.	Suma.....	33
2.1.4.1.2.	Resta.....	34
2.1.4.1.3.	Multiplicación.....	36
2.1.4.1.4.	División .....	37
2.1.4.2.	Aritmética octal.....	38
2.1.4.2.1.	Suma.....	39
2.1.4.2.2.	Resta.....	39
2.1.4.2.3.	Multiplicación.....	39
2.1.4.2.4.	División .....	41
2.1.4.3.	Aritmética hexadecimal .....	42
2.1.4.3.1.	Suma.....	43
2.1.4.3.2.	Resta.....	43
2.1.4.3.3.	Multiplicación.....	43
2.1.4.3.4.	División .....	45
2.1.5.	Complementos.....	45
2.1.5.1.	Complemento a la base disminuida.....	46
2.1.5.1.1.	Números decimales.....	47
2.1.5.1.2.	Números binarios .....	47
2.1.5.1.3.	Números octales .....	48

	2.1.5.1.4.	Números hexadecimales.....	49
	2.1.5.2.	Complemento a la base .....	50
	2.1.5.2.1.	Números decimales .....	50
	2.1.5.2.2.	Números binarios.....	51
	2.1.5.2.3.	Números octales.....	51
	2.1.5.2.4.	Números hexadecimales.....	52
	2.1.5.3.	Resta con complementos .....	52
2.1.6.		Números binarios con signo .....	55
	2.1.6.1.	Notación con signo magnitud.....	57
	2.1.6.2.	Notación con complemento a uno.....	58
	2.1.6.3.	Notación con complemento a dos.....	59
	2.1.6.4.	Notación en exceso a K.....	59
2.1.7.		Códigos binarios.....	59
	2.1.7.1.	Binario natural .....	60
	2.1.7.2.	Código BCD 8421.....	61
	2.1.7.3.	Código 2421 .....	63
	2.1.7.4.	Código exceso 3.....	64
	2.1.7.5.	Código 8 4 -2 -1 .....	64
	2.1.7.6.	Código Gray .....	65
	2.1.7.7.	Código ASCII.....	66
2.1.8.		Códigos para detectar errores .....	67
2.1.9.		Lógica binaria.....	68
2.1.10.		Compuertas lógicas.....	69
	2.1.10.1.	Compuerta lógica YES.....	71
	2.1.10.2.	Compuerta lógica NOT .....	72
	2.1.10.3.	Compuerta lógica AND .....	73
	2.1.10.4.	Compuerta lógica NAND.....	74

	2.1.10.5.	Compuerta lógica OR.....	75
	2.1.10.6.	Compuerta lógica NOR .....	76
	2.1.10.7.	Compuerta lógica XOR .....	77
	2.1.10.8.	Compuerta lógica XNOR.....	78
2.2.		Álgebra booleana.....	79
	2.2.1.	Reseña histórica .....	79
	2.2.2.	Definición .....	79
	2.2.3.	Álgebra booleana de dos valores .....	80
	2.2.4.	Principio de dualidad.....	82
	2.2.5.	Teoremas básicos del álgebra booleana.....	83
		2.2.5.1. Teorema 1.....	83
		2.2.5.2. Teorema 2.....	84
		2.2.5.3. Teorema 3, involución .....	85
		2.2.5.4. Teorema 4, asociatividad .....	85
		2.2.5.5. Teorema 5, DeMorgan .....	85
		2.2.5.6. Teorema 6, absorción .....	86
	2.2.6.	Funciones booleanas .....	87
		2.2.6.1. Simplificación de funciones booleanas con álgebra de Boole .....	91
		2.2.6.2. Complemento de una función.....	93
	2.2.7.	Niveles en compuertas lógicas.....	95
	2.2.8.	Formas canónicas.....	95
		2.2.8.1. Minitérminos y maxitérminos .....	95
	2.2.9.	Formas estándar .....	97
		2.2.9.1. Suma de productos .....	97
		2.2.9.2. Producto de sumas .....	98
	2.2.10.	Compuertas lógicas digitales .....	99
		2.2.10.1. Lógica positiva y negativa .....	102
	2.2.11.	Circuitos integrados .....	103

	2.2.11.1.	Niveles de integración.....	103
	2.2.11.2.	Familias de lógica digital.....	104
2.3.		Minimización de funciones booleanas .....	105
	2.3.1.	Introducción.....	105
	2.3.2.	Mapa de Karnaugh .....	105
	2.3.2.1.	Mapa de dos variables.....	106
	2.3.2.2.	Mapa de tres variables.....	108
	2.3.2.3.	Mapa de cuatro variables.....	112
	2.3.2.4.	Mapa para cinco variables .....	115
	2.3.2.5.	Mapa de seis variables .....	118
	2.3.3.	Simplificación a producto de sumas .....	121
	2.3.4.	Condición indiferencia .....	123
	2.3.5.	Implementación con compuertas NAND y NOR ...	125
	2.3.5.1.	Implementación de compuertas NAND y NOR.....	125
	2.3.5.2.	Compuerta NAND .....	125
	2.3.5.3.	Compuerta NOR.....	127
	2.3.6.	Compuerta OR exclusiva.....	127
2.4.		Lógica combinacional.....	129
	2.4.1.	Introducción.....	129
	2.4.2.	Procedimiento de análisis.....	130
	2.4.3.	Procedimiento de diseño .....	131
	2.4.4.	Aplicaciones utilizando lógica combinacional.....	133
	2.4.4.1.	Semisumador .....	133
	2.4.4.2.	Sumador completo.....	134
	2.4.4.3.	Sumador binario .....	137
	2.4.4.4.	Propagación del acarreo.....	138
	2.4.4.5.	Restador binario .....	142
	2.4.4.6.	Sumador-restador binario .....	142

2.4.4.7.	Sumador BCD de cuatro bits .....	143
2.4.4.8.	Multiplicador binario .....	146
2.4.4.8.1.	Multiplicador binario con sumador simple ....	146
2.4.4.8.2.	Multiplicador binario con sumadores completos.....	148
2.4.4.9.	Comparador de magnitudes .....	148
2.4.4.10.	Decodificadores .....	153
2.4.4.11.	Demultiplexores .....	153
2.4.4.11.1.	Entradas habilitadoras.	154
2.4.4.11.2.	Demultiplexor .....	154
2.4.4.11.3.	Decodificador / demultiplexor.....	156
2.4.4.12.	Codificadores .....	156
2.4.4.12.1.	Codificador con prioridad .....	159
2.4.4.13.	Multiplexor.....	160
2.5.	Lógica secuencial .....	163
2.5.1.	Circuito secuencial sincrónico .....	164
2.5.2.	Circuito secuencial asincrónico .....	165
2.5.3.	<i>Flip-Flops</i> .....	166
2.5.3.1.	<i>Latches</i> .....	166
2.5.3.1.1.	Latch SR .....	166
2.5.3.1.2.	Latch D.....	169
2.5.3.1.3.	Símbolos gráficos de <i>latches</i> .....	170
2.5.3.2.	<i>Flip-flops</i> .....	171

	2.5.3.2.1.	<i>Flip-flop D</i> disparado por flanco.....	172
	2.5.3.2.2.	<i>Flip-flop JK</i> .....	175
	2.5.3.2.3.	<i>Flip-Flop T</i> .....	177
2.5.4.		Entradas asíncronas.....	177
2.5.5.		Análisis de circuitos secuenciales.....	179
	2.5.5.1.	Ecuaciones de estado.....	180
	2.5.5.2.	Tabla de estados .....	181
	2.5.5.3.	Diagrama de estados.....	183
	2.5.5.4.	Análisis con <i>flip-flops</i> .....	184
		2.5.5.4.1. Análisis con <i>flip-flop D</i> .	185
		2.5.5.4.2. Análisis con <i>flip-flop JK</i> .....	186
		2.5.5.4.3. Análisis con <i>flip-flop T</i> .	192
2.5.6.		Máquinas de estado .....	193
2.5.7.		Reducción de estados .....	194
2.5.8.		Asignación de estados .....	198
	2.5.8.1.	Diseño de circuitos secuenciales .....	199
		2.5.8.1.1. Síntesis con <i>flip-flops D</i> .....	201
		2.5.8.1.2. Tablas de excitación .....	202
		2.5.8.1.3. Síntesis con <i>flip-flop JK</i> .....	204
		2.5.8.1.4. Síntesis con <i>flip-flops T</i> .....	206
2.6.		Registros y contadores.....	209
	2.6.1.	Registros .....	209
		2.6.1.1. Registros con carga paralela .....	210
		2.6.1.2. Registros de desplazamiento.....	211

	2.6.1.2.1.	Transferencia en serie.	212
	2.6.1.3.	Registro de desplazamiento universal.....	214
	2.6.1.4.	Suma en serie .....	216
2.6.2.	Contadores .....		217
	2.6.2.1.	Contadores de rizo .....	219
	2.6.2.1.1.	Contador binario de rizo .....	219
	2.6.2.1.2.	Contador BCD de rizo .	220
	2.6.2.2.	Contadores sincrónicos .....	223
	2.6.2.2.1.	Contador binario.....	223
	2.6.2.2.2.	Contador binario ascendente- descendente.....	225
	2.6.2.2.3.	Contador BCD.....	226
	2.6.2.2.4.	Contador binario con carga paralela .....	230
	2.6.2.2.5.	Contador con estados no utilizados .....	232
	2.6.2.2.6.	Contador anular .....	236
	2.6.2.2.7.	Contador Johnson .....	239
3.	PREGUNTAS Y EJERCICIOS CONCEPTUALES DEL CURSO ELECTRÓNICA 3.....		243
	3.1.	Preguntas y ejercicios conceptuales de sistemas de numeración.....	243
	3.1.1.	Preguntas conceptuales.....	243
	3.1.2.	Ejercicios conceptuales.....	245
	3.2.	Preguntas y ejercicios conceptuales del álgebra booleana ....	247

3.2.1.	Preguntas conceptuales .....	247
3.2.2.	Ejercicios conceptuales .....	250
3.3.	Preguntas y ejercicios conceptuales de minimización de funciones booleanas .....	253
3.3.1.	Preguntas conceptuales .....	253
3.3.2.	Ejercicios conceptuales .....	254
3.4.	Preguntas y ejercicios conceptuales de la lógica combinacional .....	258
3.4.1.	Preguntas conceptuales .....	258
3.4.2.	Ejercicios conceptuales .....	259
3.5.	Preguntas y ejercicios conceptuales de la lógica secuencial .	263
3.5.1.	Preguntas conceptuales .....	263
3.5.2.	Ejercicios conceptuales .....	265
3.6.	Preguntas y ejercicios conceptuales de los registros y contadores .....	266
3.6.1.	Preguntas conceptuales .....	266
3.6.2.	Ejercicios conceptuales .....	269
4.	EXÁMENES CORTOS DE CONCEPTOS TEÓRICOS DEL CURSO	
	ELECTRÓNICA 3 .....	271
4.1.	Exámenes de sistemas de numeración .....	271
4.1.1.	Examen uno .....	271
4.1.2.	Examen dos .....	274
4.1.3.	Examen tres .....	275
4.2.	Exámenes de álgebra booleana .....	277
4.2.1.	Examen uno .....	277
4.2.2.	Examen dos .....	279
4.2.3.	Examen tres .....	281
4.3.	Exámenes de minimización de funciones booleanas.....	283

4.3.1.	Examen uno.....	284
4.3.2.	Examen dos.....	286
4.3.3.	Examen tres.....	288
4.4.	Exámenes de lógica combinacional.....	292
4.4.1.	Examen uno.....	292
4.4.2.	Examen dos.....	294
4.4.3.	Examen tres.....	298
4.5.	Exámenes de lógica secuencial.....	300
4.5.1.	Examen uno.....	300
4.5.2.	Examen dos.....	301
4.5.3.	Examen tres.....	302
4.6.	Exámenes de sistemas de registros y contadores.....	303
4.6.1.	Examen uno.....	303
4.6.2.	Examen dos.....	305
4.6.3.	Examen tres.....	307
5.	VIDEOS DE CONCEPTOS TEÓRICOS DEL CURSO DE ELECTRÓNICA 3.....	311
5.1.	Videos del capítulo 1: sistemas de numeración.....	311
5.2.	Videos del capítulo 2: álgebra booleana.....	315
5.3.	Videos del capítulo 3: minimización de funciones booleanas.....	317
5.4.	Videos del capítulo 4: lógica combinacional.....	318
5.5.	Videos del capítulo 5: lógica secuencial.....	320
5.6.	Videos del capítulo 6: registros y contadores.....	322
	CONCLUSIONES.....	325
	RECOMENDACIONES.....	327
	BIBLIOGRAFÍA.....	329



## ÍNDICE DE ILUSTRACIONES

### FIGURAS

1.	Clasificación de las teorías de aprendizaje.....	1
2.	Diagrama de bloques del método científico.....	3
3.	Diagrama del constructivismo en educación superior.....	8
4.	Intervalo de voltaje de compuertas lógicas.....	70
5.	Simbología de compuestas lógicas.....	70
6.	Circuito equivalente de compuerta lógica yes.....	71
7.	Circuito equivalente de compuerta lógica not.....	72
8.	Circuito equivalente de compuerta lógica and.....	73
9.	Circuito equivalente de compuerta lógica nand.....	74
10.	Circuito equivalente de compuerta lógica or.....	75
11.	Circuito equivalente de compuerta lógica nor.....	76
12.	Circuito equivalente de compuerta lógica xor.....	77
13.	Circuito equivalente de compuerta lógica xnor.....	78
14.	Ley distributiva algebra booleana sobre +.....	81
15.	Ley distributiva algebra booleana sobre •.....	82
16.	Postulados y teoremas del álgebra booleana.....	83
17.	Teorema 5 a).....	86
18.	Representación de función f en compuertas lógicas.....	90
19.	Circuitos de ejemplo 36.....	90
20.	DeMorgan para una función de 3 variables.....	93
21.	Ejemplo de nivel 1.....	94
22.	Ejemplo de nivel 2.....	94
23.	Circuito equivalente de la función f1.....	97

24.	Circuito equivalente de función $f_2$ .....	98
25.	Circuito equivalente de función $f_3$ .....	99
26.	Compuertas lógicas digitales .....	101
27.	Lógica positiva y negativa .....	102
28.	Tabla de verdad y compuerta lógica en lógica positiva .....	102
29.	Tabla de verdad y compuerta lógica en lógica negativa .....	103
30.	Ejemplo de mapa de dos variables 1 .....	106
31.	Ejemplo de mapa de dos variables 2 .....	107
32.	Mapa de ejemplo 45 .....	108
33.	Mapa de tres variables .....	109
34.	Mapa de ejemplo 46 .....	110
35.	Mapa de ejemplo 47 .....	112
36.	Mapa de cuatro variables .....	113
37.	Mapa de ejemplo 51 .....	115
38.	Mapa de cinco variables .....	116
39.	Mapa de ejemplo 49 .....	117
40.	Mapa de seis variables .....	118
41.	Mapa 1 de ejemplo 50 .....	119
42.	Mapa 2 de ejemplo 50 .....	119
43.	Mapa 3 de ejemplo 50 .....	120
44.	Mapa de ejemplo 51 .....	122
45.	Circuitos equivalentes de ejemplo 51 .....	123
46.	Mapa de ejemplo 52 .....	124
47.	Compuertas and y or utilizando nand .....	126
48.	Compuerta and y or utilizando nor .....	126
49.	Compuerta or exclusiva con and-or-not y nand .....	128
50.	Diagrama de bloques de circuito combinacional .....	129
51.	Mapa del ejemplo 53 .....	132
52.	Circuito lógico de ejemplo 53 .....	133

53.	Circuito lógico del semisumador .....	134
54.	Mapa de karnaugh de sumador completo .....	135
55.	Circuitos lógicos de sumador completo .....	136
56.	Circuito lógico de sumador completo con xor .....	136
57.	Diagrama de bloques y funcionamiento de sumador binario .....	137
58.	Circuito de cada sumador completo en el sumador binario .....	138
59.	Diagrama lógico del generador del acarreo anticipado .....	139
60.	Funciones booleanas de acarreo de salida .....	140
61.	Sumador binario con propagación del acarreo .....	141
62.	Diagrama de bloques de resta binaria .....	142
63.	Sumador-restador de cuatro bits .....	143
64.	Suma binaria y suma bcd .....	144
65.	Diagrama de bloques de sumador bcd .....	145
66.	Multiplicación binaria .....	146
67.	Diagrama lógico de multiplicador binario .....	147
68.	Diagrama del multiplicador binario con sumador completo .....	149
69.	Comparador de magnitudes de cuatro bits .....	151
70.	Funciones booleanas de $a > b$ y $a < b$ .....	152
71.	Decodificador de 3 a ocho líneas .....	152
72.	Circuito con entrada habilitadora .....	154
73.	Diagrama de bloques de demultiplexor de 8 bits .....	155
74.	Diagrama lógico de demultiplexor de 8 bits .....	155
75.	Diagrama lógico del codificador de octal a binario .....	158
76.	Mapas del codificador con prioridad .....	159
77.	Diagrama lógico de codificador con prioridad .....	160
78.	Diagrama lógico y diagrama de bloques de un multiplexor de 2 a 1 ...	161
79.	Diagrama lógico de multiplexor de 4 líneas a 1 .....	162
80.	Diagrama de bloques de un circuito secuencial .....	163
81.	Diagrama de bloques de circuito secuencial con reloj .....	164

82.	Diagrama de bloques de circuito secuencial asíncrono.....	165
83.	<i>Latch sr</i> con compuertas nor.....	167
84.	<i>Latch sr</i> con compuertas nand.....	168
85.	<i>Latch s'r'</i> con entrada de control .....	169
86.	<i>Latch d</i> .....	170
87.	Símbolos gráficos de <i>latches sr, s'r'</i> y <i>d</i> .....	170
88.	Tipos de respuestas de reloj en un <i>latch</i> y un <i>flip-flop</i> .....	171
89.	<i>Flip-flop d</i> con dos <i>latches d</i> .....	173
90.	<i>Flip-flop d</i> disparado por borde positivo .....	173
91.	Símbolo gráfico y tabla característica de <i>flip-flop d</i> .....	174
92.	Circuito equivalente con <i>flip-flop d</i> y simbología de <i>flip-flop jk</i> .....	175
93.	Tabla de funcionamiento de <i>flip-flop jk</i> .....	176
94.	Circuito equivalente con <i>flip-flop d</i> y <i>jk</i> y simbología de <i>flip-flop t</i> .....	176
95.	Tabla de funcionamiento de <i>flip-flop t</i> .....	177
96.	<i>Flip-flop d</i> con restablecimiento asíncrono .....	178
97.	Circuito secuencial.....	179
98.	Tabla de estados .....	181
99.	Segunda forma de la tabla de estados.....	182
100.	Diagrama de estados.....	183
101.	Circuito secuencial con <i>flip-flop d</i> .....	185
102.	Diagrama de estados.....	186
103.	Circuito secuencial con <i>flip-flop jk</i> .....	187
104.	Diagrama de estados de circuito con <i>flip-flop jk</i> .....	190
105.	Circuito secuencial con <i>flip-flops t</i> .....	191
106.	Ejemplo de diagrama de estados.....	194
107.	Secuencia de entrada .....	195
108.	Tabla de estados con siete estados.....	196
109.	Tabla de estados con cinco estados .....	197
110.	Secuencia de entrada .....	197

111.	Diagrama de estados con cinco estados .....	198
112.	Posibles asignaciones binarias de estados .....	199
113.	Diagrama de estados para el detector de sucesiones .....	200
114.	Mapas para el detector de sucesiones .....	202
115.	Diagrama del detector de sucesiones.....	203
116.	Mapas para las ecuaciones de entrada j y k.....	205
117.	Diagrama lógico para el circuito secuencial con <i>flip-flops jk</i> .....	206
118.	Diagrama de estados de contador de tres bits .....	207
119.	Mapa de karnaugh de contador de 3 bits .....	208
120.	Diagrama lógico de contador binario de tres bits .....	209
121.	Registro de 4 bits en paralelo .....	210
122.	Registro de desplazamiento de cuatro bits .....	211
123.	Diagrama de bloques de un circuito de transferencia de registros en serie .....	212
124.	Diagrama de temporización.....	213
125.	Registro de desplazamiento universal .....	214
126.	Sumador en serie .....	216
127.	Contador binario de rizo de 4 bits.....	218
128.	Contador bcd de rizo .....	221
129.	Diagrama de estados de contador bcd .....	222
130.	Diagrama de bloques de un contador bcd de 3 décadas .....	222
131.	Contador binario sincrónico.....	224
132.	Contador binario ascendente-descendente de 4 bits.....	225
133.	Mapa de karnaugh 1 de contador sincrónico bcd .....	227
134.	Mapa de karnaugh 2 de contador sincrónico bcd .....	228
135.	Contador sincrónico bcd.....	229
136.	Contador binario con carga paralela.....	230
137.	Contador bcd empleando un contador con carga paralela.....	231
138.	Mapas de karnaugh de contador .....	234

139.	Diagrama lógico del contador .....	235
140.	Diagrama de estados del contador .....	236
141.	Contador anular de 3 bits a.....	237
142.	Mapas de karnaugh del contador anular .....	238
143.	Contador johnson .....	240
144.	Contador anular con extremos conmutados de cuatro etapas .....	241
145.	Circuito del ejercicio conceptual 1 capítulo 1 .....	245
146.	Diagrama de ejercicio conceptual 1 capítulo 2 .....	250
147.	Diagrama de ejercicio conceptual 3 capítulo 2 .....	251
148.	Diagrama de ejercicio conceptual 4 capítulo 2 .....	251
149.	Respuesta del ejercicio conceptual 4 capítulo 2.....	252
150.	Ejercicio conceptual 2 capítulo 3.....	255
151.	Respuesta del ejercicio conceptual 2 capítulo 3.....	256
152.	Circuito de ejercicio conceptual 4 capítulo 3 .....	256
153.	Respuesta del ejercicio conceptual 4 capítulo 3.....	257
154.	Circuito de ejercicio conceptual 5 capítulo 3 .....	257
155.	Circuito equivalente del ejercicio conceptual 5 capítulo 3 .....	257
156.	Ecuaciones del ejercicio conceptual 3 capítulo 4 .....	261
157.	Circuito del ejercicio conceptual 4 capítulo 4 .....	262
158.	Diagrama de bloques del ejercicio conceptual 5 capítulo 4 .....	262
159.	Diagrama del ejercicio conceptual 1 capítulo 6 .....	269
160.	Circuito del ejercicio conceptual 2 capítulo 6 .....	270
161.	Diagrama del enunciado 3 capítulo 2 examen 2 .....	280
162.	Diagrama del enunciado 6 capítulo 2 examen 3 .....	282
163.	Diagrama del enunciado 7 capítulo 2 examen 3 .....	282
164.	Respuesta del enunciado 7 capítulo 2 examen 3.....	283
165.	Mapa del enunciado 6 capítulo 3 examen 1 .....	285
166.	Mapa de karnaugh del enunciado 7 capítulo 3 examen 1 .....	285
167.	Respuesta del enunciado 7 capítulo 3 examen 1 .....	286

168.	Enunciado 4 capítulo 3 examen 2 .....	287
169.	Mapa de karnaugh del enunciado 5 capítulo 3 examen 2.....	287
170.	Respuesta del enunciado 5 capítulo 3 examen 2 .....	288
171.	Enunciado 3 capítulo 3 examen 3 .....	289
172.	Enunciado 4 capítulo 3 examen 3 .....	289
173.	Respuesta del enunciado 4 capítulo 3 examen 3 .....	290
174.	Enunciado 6 capítulo 3 examen 3 .....	290
175.	Respuesta del enunciado 6 capítulo 3 examen 3 .....	291
176.	Enunciado 7 capítulo 3 examen 3 .....	291
177.	Respuesta de enunciado 7 capítulo 3 examen 3 .....	291
178.	Circuito 1 del enunciado 5 capítulo 4 examen 1 .....	293
179.	Circuito 2 del enunciado 5 capítulo 4 examen 1 .....	293
180.	Circuito del enunciado 5 capítulo 4 examen 2 .....	295
181.	Ecuaciones enunciado 5 capítulo 4 examen 2.....	297
182.	Respuesta del enunciado 6 capítulo 4 examen 2 .....	297
183.	Enunciado 5 capítulo 4 examen 1 .....	299
184.	Enunciado 6 capítulo 4 examen 3 .....	299
185.	Enunciado 6 capítulo 6 examen 1 .....	305
186.	Enunciado 6 capítulo 6 examen 2 .....	307
187.	Mapas de karnaugh de enunciando 4 capítulo 6 examen 3.....	309
188.	Circuito de enunciando 4 capítulo 6 examen 3.....	309

## TABLAS

I.	Diferencias entre positivismo y constructivismo.....	11
II.	Valores de letras en números hexadecimales .....	20
III.	Tabla para resolución de ejemplo 8.....	29
IV.	Tabla para resolución de ejemplo 9.....	30

V.	Tabla de conversión de números decimales, binarios, octales y hexadecimales.....	31
VI.	Tabla para resolución de ejemplo 10 .....	32
VII.	Datos importantes para la suma de números binarios .....	34
VIII.	Datos importantes para la resta de números binarios .....	36
IX.	Datos importantes para la multiplicación de números binarios.....	37
X.	Tabla para suma de números octales .....	38
XI.	Tabla para suma de números hexadecimales.....	42
XII.	Complementos a la base r y a la base disminuidas de sistemas de numeración.....	46
XIII.	Fórmula de complementos a la base disminuida .....	46
XIV.	Fórmula de complementos a la base disminuida .....	49
XV.	Ejemplo de números decimales y números binarios con y sin signo ....	56
XVI.	Ejemplo de notación con complemento a uno.....	56
XVII.	Ejemplo con notación con complemento a dos .....	57
XVIII.	Ejemplo con notación en exceso a k.....	58
XIX.	Dígitos en codificación binario natural.....	60
XX.	Dígitos en codificación bcd .....	61
XXI.	Dígitos en codificación 2421 .....	62
XXII.	Dígitos en codificación xs3.....	63
XXIII.	Dígitos en codificación 8 4 -2 -1 .....	64
XXIV.	Dígitos en codificación gray .....	65
XXV.	Lista de código ascii a.....	66
XXVI.	Lista de código ascii b.....	67
XXVII.	Ejemplo de paridad .....	67
XXVIII.	Tabla de verdad de compuerta lógica yes.....	71
XXIX.	Tabla de verdad de compuerta lógica not .....	72
XXX.	Tabla de verdad de compuerta lógica and .....	73
XXXI.	Tabla de verdad de compuerta lógica nand .....	74

XXXII.	Tabla de verdad de compuerta lógica or .....	75
XXXIII.	Tabla de verdad de compuerta lógica nor .....	76
XXXIV.	Tabla de verdad de compuerta lógica xor.....	77
XXXV.	Tabla de verdad de compuerta lógica xnor.....	78
XXXVI.	Complemento .....	82
XXXVII.	Asociatividad .....	85
XXXVIII.	Teorema 5 b) .....	86
XXXIX.	Tabla de verdad de la función f1 .....	87
XL.	Tabla de ejemplo 36.....	89
XLI.	Maxitérminos y minitérminos para tres variables binarias.....	96
XLII.	Tablas de verdad para las 16 funciones de dos variables binarias .....	100
XLIII.	Tabla de verdad de ejemplo 45 .....	107
XLIV.	Tabla de verdad de ejemplo 46 .....	109
XLV.	Tabla de verdad de ejemplo 47 .....	111
XLVI.	Tabla de verdad de ejemplo 48 .....	114
XLVII.	Tabla de verdad del ejemplo 53 .....	132
XLVIII.	Tabla de verdad de semisumador .....	134
XLIX.	Tabla de verdad de sumador completo .....	135
L.	Tabla de verdad de decodificador de 3 a 8 líneas .....	153
LI.	Tabla de demultiplexor de 8 bits.....	156
LII.	Tabla de verdad del codificador octal a binario.....	157
LIII.	Tabla de verdad de codificador de cuatro entradas con prioridad.....	158
LIV.	Tabla de verdad de multiplexor de 4 líneas a 1 .....	161
LV.	Tabla de estados.....	186
LVI.	Entradas de <i>flip-flop</i> .....	188
LVII.	Tabla de estados del circuito secuencial con <i>flip-flop jk</i> .....	189
LVIII.	Tabla de estados de circuito con <i>flip-flop t</i> .....	192
LIX.	Tabla de estados con seis estados .....	196
LX.	Tabla de estados del detector de sucesiones.....	201

LXI.	Tablas de excitación de <i>flip-flops jk</i> y <i>t</i> .....	203
LXII.	Tabla de estados y entradas de circuito con <i>flip-flop jk</i> .....	204
LXIII.	Tabla de estados del contador de tres bits.....	207
LXIV.	Transferencia en serie .....	213
LXV.	Funciones del registro de desplazamiento universal de 4 bits.....	215
LXVI.	Comportamiento de contador binario de rizo de 4 bits .....	220
LXVII.	Tabla de comportamiento de contador sincrónico bcd .....	226
LXVIII.	Funcionamiento del contador binario con carga paralela .....	231
LXIX.	Tabla de estados de contador con estados no utilizados .....	232
LXX.	Comportamiento del contador anular .....	237
LXXI.	Funcionamiento del contador johnson .....	239
LXXII.	Tabla del enunciado 7 capítulo 1 examen 1 .....	273
LXXIII.	Tabla del enunciado 6 capítulo 1 examen 3.....	276
LXXIV.	Tabla del resultado del enunciado 6 capítulo 1 examen 3.....	277
LXXV.	Tabla del enunciado 6 capítulo 3 examen 1 .....	285
LXXVI.	Enunciado 4 capítulo 6 examen 3 .....	308

## LISTA DE SÍMBOLOS

Símbolo	Significado
$\oplus$	Función XOR.
$\bullet$	Multiplicación en álgebra booleana.



## GLOSARIO

<b>Acarreo</b>	Se usa cuando un dígito ha sido transferido de una columna de dígitos a otra columna de mayor potencia en un algoritmo de cálculo.
<b>Analógico</b>	Cualquier señal que tenga una cantidad infinita de magnitudes.
<b>Bit</b>	Es el acrónimo de <i>binary digit</i> y es un dígito en el sistema de numeración binaria.
<b>Código</b>	Conjunto de símbolos y combinaciones de ceros y unos, que sirven para representar información numérica y alfabética.
<b>Digital</b>	El término surgió cuando las computadoras o sistemas estaban diseñados para tratar dígitos.
<b>Dígitos</b>	Símbolo que se utiliza para representar alguna cantidad numérica.
<b>Elemento discreto</b>	Forma particular de codificación que toma un símbolo o un paquete de información.

<b>Informática</b>	Serie de recursos, herramientas y dispositivos que permiten que la información sea procesada de manera automática.
<b>Postulado</b>	Suposición de la cual se deduce reglas, teorías y propiedades.
<b>Sistemas digitales</b>	Sistemas diseñados para el procesamiento de dígitos.
<b>Tabla de verdad</b>	Es una lista de combinaciones de unos y ceros asignados a las variables binarias y una columna que muestra el valor de la función para cada combinación binaria.
<b>Telecomunicaciones</b>	Es toda transmisión y recepción de señales de cualquier naturaleza, típicamente electromagnéticas, que contengan signos, sonidos, imágenes o, en definitiva, cualquier tipo de información que se desee comunicar a cierta distancia
<b>Teorema</b>	Afirmación comprobable mediante postulados.
<b>Transductores</b>	Es un dispositivo capaz de transformar o convertir una determinada manifestación de energía de entrada, en otra diferente a la salida.

## RESUMEN

Las tecnologías de la información cada vez están abarcando más ámbitos laborales; y la educación no es la excepción. En el primer capítulo se describe un poco de la historia acerca de los procedimientos de enseñanza que han surgido en la historia, iniciando desde el positivismo hasta la teoría constructivista, la cual es en la que está enfocado el trabajo de graduación.

En el capítulo dos se hace énfasis en los temas que abarca el curso de electrónica 3. En él se describen los conceptos del manejo de números binarios, álgebra de Boole, reducción de expresiones booleanas, lógica combinacional, lógica secuencial y el análisis y diseño de circuitos digitales. En cada uno de los temas, se han realizado una gran variedad de ejemplos que el estudiante puede tomar en cuenta cuando desee estudiar el tema más a fondo. Los conceptos expuestos cuentan con reseñas históricas para que el alumno del curso de Electrónica 3, pueda conocer en donde iniciaron estos conceptos.

En el tercer capítulo se realizó una variedad de problemas resueltos seleccionados para que el estudiante pueda ampliar sus conocimientos en la resolución de problemáticas utilizando los conceptos aprendidos en el curso de Electrónica 3.

En el cuarto capítulo se expone una serie de pruebas propuestas para que el catedrático del curso pueda comprobar el conocimiento de los alumnos de los conceptos teóricos del curso. Cada examen está compuesto de una variedad de preguntas conceptuales y ejercicios en donde el alumno podrá demostrar sus conocimientos. Las pruebas están divididas de forma modular, para que cada

prueba evalúe conceptos distintos del curso. Las pruebas podrán tener o no valor según el catedrático lo vea conveniente.

En el capítulo 5 estarán expuestos videos educativos para que el estudiante del curso de Electrónica 3 pueda tener otra vía de aprendizaje. Estos videos cubrirán los conceptos descritos en el capítulo dos. Los videos fueron hechos con el afán de darles a los alumnos una alternativa más audiovisual, y de este modo fomentar el autoaprendizaje y la proactividad estudiantil.

# OBJETIVOS

## General

Proporcionar información digital para facilitar el aprendizaje de los conceptos teóricos del curso de Electrónica 3 a los estudiantes de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala.

## Específicos

1. Proveer tiempo extra al catedrático que imparte el curso de Electrónica 3 para que pueda profundizar en los temas impartidos.
2. Dar al catedrático que imparte el curso de Electrónica 3 tiempo extra para que pueda proveer de una mayor cantidad de ejemplos prácticos utilizando electrónica digital.
3. Mejorar la calidad académica de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala.
4. Dejar una brecha amplia para que puedan llegar a expandir el contenido del curso.



## INTRODUCCIÓN

Actualmente, todo el contenido de las cátedras de la Escuela de Mecánica Eléctrica de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, son impartidas de forma presencial. Esto dificulta mucho a las personas que residen en lugares allende a la universidad. Al impartir los conceptos teóricos en las cátedras, el ingeniero a cargo del curso pierde tiempo valioso, en el cual podría dar énfasis a los problemas de aplicación que, por lo general, tienen un mayor grado de dificultad. Un curso como Electrónica 3 requiere de una gran dedicación y es necesaria la asistencia a la mayoría de las cátedras para entender la complejidad de su contenido. También es necesario el estudio profundo de las aplicaciones prácticas que ofrece la electrónica digital.

Aprovechando las tecnologías de la información TICS, se puede otorgar al estudiante universitario otra alternativa para el aprendizaje de los conceptos teóricos de los cursos. Esto con el afán de ayudarlo a que tenga una plataforma virtual donde pueda tener acceso a la información 24/7 y acceder a ella desde cualquier computador.

Por ello se presentan los conceptos teóricos del curso Electrónica 3, para que el alumno pueda tener acceso a la información desde cualquier dispositivo que tenga la capacidad de conectarse a internet. Esto fue hecho en base al libro que el catedrático utiliza para impartir el curso. Ayudará a que el catedrático pueda distribuir mejor su tiempo y no utilizarlo para la explicación de los conceptos teóricos (los cuales el alumno puede leerlos por su cuenta) sino enfocarse en las aplicaciones actuales reales de la electrónica digital.



# 1. MARCO TEÓRICO

## 1.1. Aprendizaje

Es la adquisición de conocimientos, habilidades o actitudes que el individuo internaliza y luego las utiliza.

Las teorías del aprendizaje son construcciones teóricas que proponen como aprende el ser humano desde distintos puntos de vista y argumentos explicativos que integran elementos biológicos, sociales, psicológicos y culturales.

Figura 1. Clasificación de las teorías de aprendizaje



Fuente: EDUTICPERU, 01 *Teorías del aprendizaje*.  
<https://goo.gl/Ew2ox3>. Consulta: 17 de noviembre de 2016.

Como se ve hay un gran número de teorías del aprendizaje, que con el pasar de los años se han ido incrementando en conocimiento y en personas que las aplican.

### **1.1.1. Teoría conductista**

Desde sus orígenes se centra en la conducta observable intentando hacer un estudio totalmente empírico de la misma, y queriendo controlar y predecir esta conducta.

### **1.1.2. Teoría cognitivista**

Es una teoría psicológica cuyo objeto de estudio es como la mente procesa, interpreta y almacena la información en la memoria. Se interesa por la forma en como la mente humana piensa y aprende. Esta se centra en el sujeto que hace el pensamiento más que en lo que lo rodea.

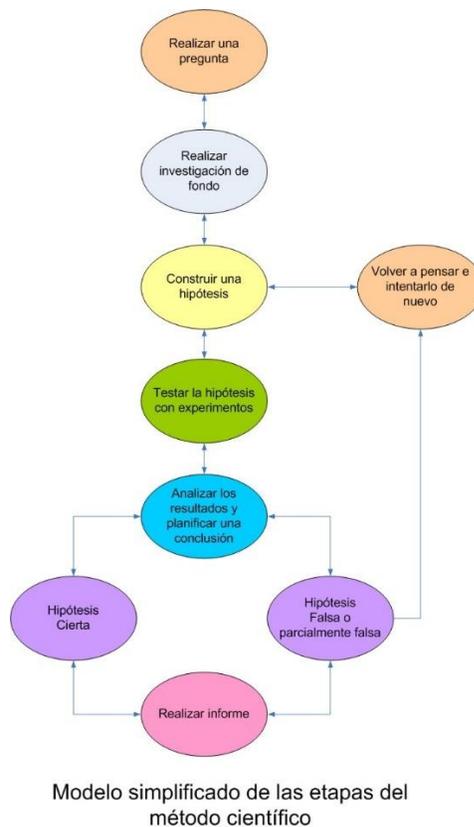
### **1.1.3. Teoría sociocultural**

Esta pone el acento en la participación proactiva con el ambiente que lo rodea. Establece que los niños desarrollan el aprendizaje con la interacción social y poco a poco van incrementando sus habilidades cognitivas. Esto quiere decir que el lugar donde resida el individuo es factor para su aprendizaje.

### 1.1.4. Teoría constructivista

Es la teoría del aprendizaje que destaca la importancia de la acción, es decir, del proceder activo en proceso de aprendizaje. Esta se basa en que para que exista un aprendizaje real debe ser construido y/o reconstruido por el mismo sujeto que aprende, y esto solo se logra a través de la acción y no simplemente algo que se pueda transmitir.

Figura 2. Diagrama de bloques del método científico



Fuente: *Método científico*. <https://goo.gl/TqVkvG>. Consulta: 28 de noviembre de 2016.

## 1.2. Método científico

Es no dogmático ya que se basa en leyes deducidas por el hombre y no en principios revelados. Sus leyes son siempre rechazadas si los hechos contradicen lo que afirman. Su validez la confirma la experiencia diaria de su uso.

El método científico está sustentado por 2 pilares fundamentales los cuales son:

- Reproducible: esto quiere decir que se debe repetir por cualquier otra persona teniendo similitud de circunstancias.
- Refutabilidad: significa que toda proposición científica debe ser susceptible a ser falsada, es decir, que no acepta verdades absolutas.

Al ser un método no dogmático se dice que este no establece una serie de pasos que se deben seguir con rigurosidad, pero en base a la experiencia, se ha detectado que hay una serie de pasos los cuales los científicos realizan al hacer una investigación, y estos son:

- Observación
- Inducción
- Hipótesis
- Experimentación
- Teoría
- Ley

### **1.3. Positivismo**

El positivismo es una corriente o escuela filosófica que afirma que el único conocimiento auténtico es el conocimiento científico. Esta afirmación positiva de las teorías es a través del método científico.

### **1.4. Constructivismo**

El constructivismo, es una posición epistemológica o manera de explicar cómo el ser humano, a lo largo de su historia personal, va desarrollando su intelecto y conformando sus conocimientos. O en palabras más sencillas es un modelo pedagógico que dice que el conocimiento no es una copia de la realidad, sino una construcción del ser humano. Esta construcción se realiza con los esquemas que la persona ya posee, por lo que utiliza los conocimientos previos. Para el constructivismo no es lo más importante el conocimiento en sí, sino adquirir competencias con él, es decir, aplicar lo ya conocido a una situación nueva.

#### **1.4.1. Teorías constructivistas**

- Jean Piaget: ha contribuido grandemente a comprender cómo se aprende y con ello cómo se puede generar procesos de aprendizaje. Afirma que es a través de una interacción activa con las cosas cuando realmente hay aprendizaje. Es allí cuando la persona es actor y protagonista del propio aprendizaje.

- Lev Vygotsky: propone que es necesaria una interrelación entre las personas y su ambiente para que se generen aprendizajes. En las interacciones se van ampliando las estructuras mentales, se construyen conocimientos, valores, actitudes y habilidades. La cultura juega un papel importante, pues proporciona a la persona las herramientas necesarias para modificar su ambiente.
- Jerome Bruner: plantea que el niño es un ser social con una cultura y una serie de conocimientos (conocimientos previos) los cuales organiza en estructuras mentales al realizar alguna actividad y aprende cuando descubre a través de lo que se ha realizado. Resalta mucho la importancia de la acción, del hacer, del descubrir a través de la ayuda del docente.
- David Ausubel: el nuevo conocimiento debe tener en cuenta los conocimientos previos que cada estudiante tenga, alguna imagen mental, un símbolo o concepto que sea relevante o importante para él o ella. Para ello se debe relacionar con su vida, con su ambiente; lo que se relaciona con los conocimientos que ha aprendido de su entorno le será significativo.

#### **1.4.2. Características del constructivismo**

- Permite que el estudiante dirija el aprendizaje, cambie la estrategia y cuestione el contenido.
- Fomenta el dialogo y la colaboración entre los alumnos, los alumnos y el maestro.
- Alimenta la curiosidad de los estudiantes a través del uso frecuente del modelo de aprendizaje.

- Estimula y acepta la iniciativa y autonomía del educando.

### **1.4.3. Roles de aprendizaje en el constructivismo**

En la enseñanza siempre estarán involucrados dos sujetos, una persona que está aprendiendo (estudiante) y una persona que está enseñando (catedrático). En el constructivismo cada uno tendrá un rol importante para que la enseñanza sea efectiva.

#### **1.4.3.1. Rol del estudiante**

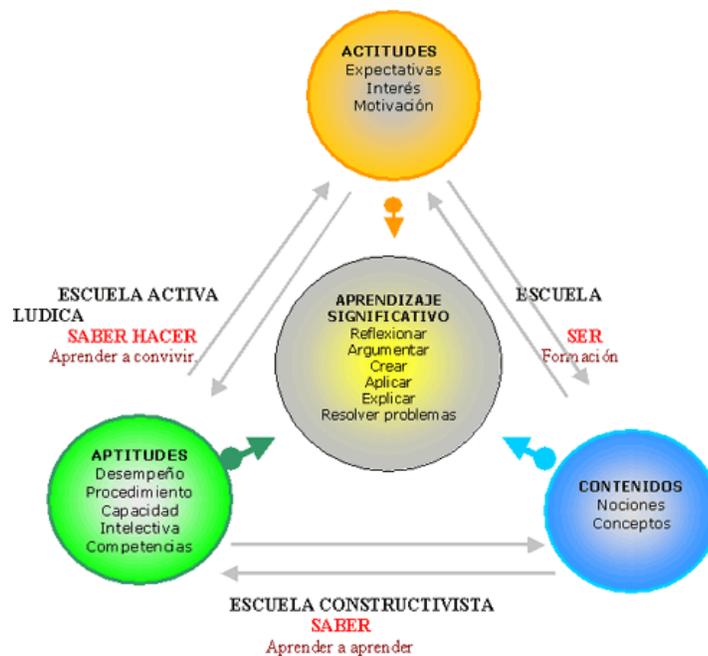
- Es el responsable de su propio proceso de aprendizaje.
- Construye el conocimiento por sí mismo y nadie puede sustituirle en esta tarea.
- Relaciona la información nueva con los conocimientos previos.
- Da un significado a las informaciones que recibe.
- Reconoce que necesita apoyos y asume una actitud de búsqueda colaborativa.
- Asume un papel activo en su proceso de aprendizaje.
- Reconoce la interacción, como condición fundamental en su proceso.
- Valora las diferentes opciones que se le presentan para sumir posiciones críticas.

#### **1.4.3.2. Rol del catedrático**

- Mediador del aprendizaje.

- Responsable de inculcar en el alumno el interés por desarrollar un proceso crítico, honesto y responsable, donde el alumno ha de adquirir una serie de hábitos y habilidades.
- Motivar a construir su conocimiento investigando, compartiendo y descubriendo nuevos temas a partir de la interacción con otras personas.

Figura 3. **Diagrama del constructivismo en educación superior**



Fuente: DEWEY, John. *Constructivismo en educación superior*. <https://goo.gl/3frZzT>.

Consulta: 29 de noviembre de 2016.

#### **1.4.4. El constructivismo en la educación superior**

El constructivismo aún no se ha adoptado como dogma en la educación superior, pero poco a poco ha ido abarcando a una gran comunidad de profesores que creen que esta es la mejor manera de enseñar. Se han dado cuenta que el rol de profesor no solo consiste en transmitir una teoría y lograr que sea memorizada sino el ser tutor y guía para que, al momento de necesitar los conocimientos, puedan comprenderlos y aplicarlos, en especial en la profesión. Sabiendo esto, es necesario la formación de conocimientos a través de una estrecha relación entre alumnos y catedrático; esto para tener un ambiente listo para realizar discusiones y crear pensamientos críticos en los estudiantes.

En el proceso del aprendizaje un alumno de pregrado encontrará que tendrá que adaptar una gran cantidad de conocimientos y actitudes para lograr responder a dificultades y problemas presentados en sus experiencias tanto cotidianas como laborales. Con estas experiencias las habilidades aprendidas adquieren significado, en donde puede dimensionar lo aprendido y enfrentarse a la realidad.

Algo que es sumamente necesario que el catedrático comprenda es que siempre tiene que iniciar con los conceptos básicos del tema que se está impartiendo. De este modo el estudiante comprenderá mejor y cuando se esté trabajando con algo más práctico y engorroso obtendrá mejores resultados. Una buena manera de aplicar el constructivismo es proporcionar al alumno la información que luego tiene que ser aplicada; se le dan las herramientas que podrá aplicar según su criterio y trabajar en conjunto con sus compañeros.

Es de mucha relevancia saber que los alumnos construyen su propio conocimiento a partir de todas las percepciones vividas y de todos sus aprendizajes anteriores para motivar el aprendizaje en base de nuevas experiencias. Si un alumno fue educado en base a la teoría constructivista de enseñanza, se esperaría que tuviera las siguientes características:

- Capacidad de trabajo independiente
- Capacidad de trabajo en equipos multidisciplinarios
- Capacidad de autoaprendizaje
- Capacidad para aceptar y respetar diversidad

Para lograr estas competencias profesionales es necesario que el proceso educativo gire en torno a siete principios de buenas prácticas en la enseñanza.

- Fomento del contacto entre profesor y estudiante
- Desarrollo de la reciprocidad y cooperación entre los estudiantes
- Motivación del aprendizaje activo
- Retroalimentación pronta
- Énfasis en la programación de las tareas
- Comunicación de expectativas altas
- Respeto por la diversidad de talentos y estilos de aprendizaje

### **1.5. Positivismo frente a constructivismo**

El positivismo y constructivismo son ideologías bastante diferentes, pero hay algunos puntos clave en los cuales es prudente resaltar y estos son la universalidad, la falsedad, la repetibilidad y el reduccionismo. Estas diferencias están expuestas en la siguiente tabla.

Tabla I. **Diferencias entre positivismo y constructivismo**

	<b>Constructivismo</b>	<b>Positivismo</b>
<b>Universalidad</b>	Lo que se considera verdad depende del contexto social y cultural del lugar y el momento	Sostiene que la verdad puede conocerse a través de leyes generales y universales
<b>Falsedad</b>	La decisión de refutar una teoría en particular sosteniendo que es falsa requiere de un juicio de valor formado por creencias condicionadas culturalmente	El conocimiento es reunido a través de la observación y la prueba de hipótesis, sujetos a los ensayos científicos de la falsedad
<b>Repetibilidad</b>	Las condiciones para observar los fenómenos cambian en la medida que las sociedades desarrollan nuevas instituciones y revisan sus valores culturales, lo cual lleva a diferente interpretación de los datos	Es algo que se puede explicar y repetir
<b>Reduccionismo</b>	Enfatiza el carácter holístico, relativo e interdependiente de cualquier sujeto bajo investigación científica.	Ve el conocimiento como un cuerpo de hechos, cada uno de los cuales puede tomarse teóricamente por separado de los demás, y que las aparentes unidades de las experiencias sensoriales están reducidas a sus constituyentes elementales

Fuente: elaboración propia, empleando Excel.

## 1.6. **Tecnologías de la información y comunicación (TIC)**

Las tecnologías de la información y comunicación tienen mucho que ver con la informática y las telecomunicaciones, las cuales permiten que la información pueda ser procesada automáticamente y que exista comunicación entre redes y visualización de recursos en internet, respectivamente.

Algunos ejemplos de TIC son:

- Bases de datos
- Internet
- Celulares
- Computadora
- Videos
- Video conferencias
- Páginas web
- Correo electrónico

#### **1.6.1. Características de las TIC**

Las principales características que se pueden mencionar de las TIC son las siguientes:

- **Inmaterialidad:** las TIC no se componen exactamente de algún material físico como el papel, sino de información que se encuentra almacenada en algún dispositivo representado en ceros y unos para que un computador pueda entenderlas, transferirlas y convertirlas a un medio que sea entendible por el humano. Las TIC han facilitado muchos aspectos de la vida de las personas gracias a esta característica. Con la automatización de tareas se puede, por ejemplo, programar actividades que realizan automáticamente los ordenadores con total seguridad y efectividad.
- **Instantaneidad:** se puede conseguir información y comunicarse instantáneamente, no importando la distancia del emisor y el receptor.

- **Interactividad:** la digitalización permite disponer de información inmaterial, para almacenar grandes cantidades en pequeños soportes o acceder a información ubicada en dispositivos. Las nuevas TIC se caracterizan por permitir la comunicación bidireccional, entre personas sin importar donde se encuentre.

### **1.6.2. Desventajas de las TIC**

A pesar de que las TIC facilitan, a una gran parte de la población, el aprendizaje a diario, tiene algunas desventajas las cuales siempre hay que tomar en cuenta:

- Hace una mayor diferencia entre las personas que no tienen la posibilidad del acceso a internet y a las que sí tienen acceso. Esto hace que la competitividad laboral en las empresas actuales sea favorable para las personas que saben utilizarlas.
- Provoca aislamiento por parte de las personas que pasan una gran parte de su tiempo diario en estas tecnologías, y no precisamente trabajando; como en redes sociales, juegos en línea, pornografía entre otros.
- Hay una mayor facilidad para realizar fraude dado a que no se puede detectar con facilidad con qué persona se está interactuando



## **2. CONCEPTOS TEÓRICOS DEL CURSO ELECTRÓNICA 3**

### **2.1. Sistemas de numeración**

Los sistemas de numeración son la base de todas las ciencias técnicas y científicas y en cada región del mundo han evolucionado de distinta manera.

#### **2.1.1. Reseña histórica**

Desde hace aproximadamente 10 000 años el hombre es sedentario y con el pasar del tiempo logra desarrollar su lenguaje y sistemas de comunicación. Para sobrevivir a la hambruna, comienza a cultivar plantas, hierbas y vegetales, además de criar animales. Creó herramientas y materiales. Sin embargo, a consecuencia de lo anterior, surge la necesidad de contar animales, realizar trueques con otras familias, llevar la noción del paso de los días según la luna, entre otros. Así comenzaron a llevar el registro de cantidades pequeñas, asignando a cada objeto un dedo, una piedra, una marca o un nudo.

Sin embargo, para cantidades mayores, este método de conteo se vuelve inadecuado, por lo que los distintos pueblos y culturas tuvieron que idear sistemas de representación más complejos. En base a esta necesidad, muchas civilizaciones empezaron a realizar su propio sistema de numeración en base a símbolos que representaban una cierta cantidad numérica, y cada una tenía una distinta cantidad de símbolos que se basaban en sus necesidades. Algunos de ellos son:

- Sistema de numeración babilónico
- Sistema de numeración egipcio
- Sistema de numeración chino-japonés
- Sistema de numeración romana
- Sistema de numeración maya
- Sistema de numeración inca
- Sistema de numeración mapuche

Tras la constante globalización, se fueron adaptando características de cada uno de los sistemas de numeración hasta convertirse en los sistemas de numeración utilizados hoy día.

### **2.1.2. Sistemas de numeración importantes en la electrónica digital**

En la electrónica digital son necesarios simplemente una cantidad finita de sistemas de numeración, los cuales se describen a continuación.

#### **2.1.2.1. Sistema de numeración decimal**

Los sistemas de numeración ayudan a representar cualquier cantidad numérica con una cantidad específica de dígitos. La cantidad de dígitos dependerá del sistema numérico que se esté utilizando. Se puede decir que una de las propiedades que define un sistema numérico es la cantidad de dígitos que se utilicen. Por otra parte, los números (independientemente del sistema de numeración) son escritos de la siguiente manera:

$$a_5 a_4 a_3 a_2 a_1 a_0 . a_{-1} a_{-2}$$

Otra forma más general es la siguiente:

$$q * a_5 + w * a_4 + e * a_3 + e * a_2 + r * a_1 + y * a_0 \cdot a_{-1} + q * a_{-2}$$

Los cuales son multiplicados por un número base elevado a una potencia, dependiendo de la posición que pertenezca. Si el número base es, por ejemplo, diez, se puede decir que el sistema de numeración utilizado es de base diez. En base a esto, otra propiedad que define a un sistema numérico es el número base que se utilice.

Ejemplo 1: escribir el número 3561.75 con exponenciales de base diez.

$$a_3 * 10^3 + a_2 * 10^2 + a_1 * 10^1 + a_0 * 10^0 + a_{-1} * 10^{-1} + a_{-2} * 10^{-2}$$
$$3 * 10^3 + 5 * 10^2 + 6 * 10^1 + 1 * 10^0 + 7 * 10^{-1} + 5 * 10^{-2}$$

El sistema de numeración es decimal porque utiliza diez dígitos y se utiliza una base del número diez. Ahora bien, sería tedioso indicar una cantidad numérica utilizando para ello una gran sumatoria con números con exponenciales, por ello, se utiliza la siguiente nomenclatura:

$$(3561.75)_{10}$$

El subíndice diez indica que es un sistema decimal, pero como se ha adoptado como el sistema de numeración favorito, se ha desvanecido la necesidad de incluir el diez en número escrito en el sistema de numeración decimal.

### 2.1.2.2. Sistema de numeración binario

El sistema de numeración binario es más antiguo de lo que se podría creer. Inició en el tercer siglo y tuvo varias aplicaciones distintas de las que se conocen hoy día. Una de las más relevantes es que fue utilizado para encriptar mensajes en tiempo de guerra. Luego en 1937 hicieron su primera aparición en los circuitos, utilizando relés y conmutadores. Ahora utilizando los mismos conceptos se realizan los diseños digitales más sofisticados.

Ahora, en la sección anterior, se escribe un valor numérico en el sistema binario, el cual está conformado por una cantidad de dos dígitos y se utiliza de base el número dos. Los dígitos utilizados en este caso son uno (1) y cero (0).

Ejemplo 2: escribir el número 3561.75 en el sistema de numeración binario.

$$\begin{aligned} & a_{11} * 2^{11} + a_{10} * 2^{10} + a_9 * 2^9 + a_8 * 2^8 + a_7 * 2^7 + a_6 * 2^6 + a_5 * 2^5 + a_4 * 2^4 + a_3 \\ & \quad * 2^3 + a_2 * 2^2 + a_1 * 2^1 + a_0 * 2^0 + a_{-1} * 2^{-1} + a_{-2} * 2^{-2} \\ & 1 * 2^{11} + 1 * 2^{10} + 0 * 2^9 + 1 * 2^8 + 1 * 2^7 + 1 * 2^6 + 1 * 2^5 + 0 * 2^4 + 1 * 2^3 + \\ & \quad 0 * 2^2 + 0 * 2^1 + 1 * 2^0 + 1 * 2^{-1} + 1 * 2^{-2} = 3561.75 \\ & \quad (110111101001.11)_2 \end{aligned}$$

Lo que hace es obtener los valores que multiplican los valores elevados a potencias y ponerlos en ese orden. El sistema de numeración es binario porque utiliza dos dígitos y una base del número dos. Para otros casos que no sea el sistema decimal, es necesario indicar en qué sistema numérico se encuentra. En este caso, por ser el sistema de numeración binario, se indica por medio del subíndice dos.

### 2.1.2.3. Sistema de numeración octal

El sistema octal surgió de la necesidad de hacer más sencilla la comunicación de un usuario a la computadora empleando números binarios. Y dado a que un dígito octal equivale a tres dígitos binarios, el usuario tendría que interpretar solo una cifra en comparación de tres. Este tendrá una cantidad de ocho dígitos y se utilizará de base el número ocho (8). Los dígitos utilizados son: cero (0), uno (1), dos (2), tres (3), cuatro (4), cinco (5), seis (6) y siete (7).

Ejemplo 3: escribir el número 3561.75 en el sistema de numeración octal.

$$\begin{aligned} a_4 * 8^4 + a_3 * 8^3 + a_2 * 8^2 + a_1 * 8^1 + a_0 * 8^0 + a_{-1} * 8^{-1} \\ 6 * 8^3 + 7 * 8^2 + 5 * 8^1 + 1 * 8^0 + 6 * 8^{-1} = 3561.75 \\ (6751.6)_8 \end{aligned}$$

El sistema es octal porque utiliza ocho dígitos y se utiliza una base del número 8. Para otros casos que no sea el sistema decimal, es necesario indicar en que sistema numérico se encuentra. En este caso, por ser el sistema octal, se indica por medio del subíndice ocho.

### 2.1.2.4. Sistema de numeración hexadecimal

El sistema hexadecimal surgió de la necesidad de hacer más sencilla la comunicación de un usuario a la computadora empleando números binarios. Y dado a que un dígito hexadecimal equivale a cuatro dígitos binarios, el usuario tendría que interpretar solo una cifra en comparación de cuatro. Este tendrá una cantidad de 16 dígitos y se utilizará de base el número 16.

¿Pero que otros símbolos numéricos existen aparte de los números del cero (0) al nueve (9)? para los seis dígitos restantes se utilizarán las primeras letras mayúsculas del abecedario. Los dígitos utilizados son: cero, uno, dos, tres, cuatro, cinco, seis, siete, ocho, nueve, A, B, C, D, E y F, teniendo las letras los valores expuestos en la tabla II.

Tabla II. **Valores de letras en números hexadecimales**

Letra	Valor
A	10
B	11
C	12
D	13
E	14
F	15

Fuente: elaboración propia, empleando Excel.

Ejemplo 4: escribir el número 3561,75 en el sistema de numeración hexadecimal.

$$a_2 * 16^2 + a_1 * 16^1 + a_0 * 16^0 + a_{-1} * 16^{-1}$$

$$D * 16^2 + E * 16^1 + 9 * 16^0 + C * 16^{-1} = 3561.75$$

$$(DE9.C)_{16}$$

El sistema es hexadecimal porque utiliza 16 dígitos y se utiliza una base del número 16. Para otros casos que no sea el sistema decimal, es necesario indicar en que sistema numérico se encuentra. En este caso, por ser el sistema de numeración hexadecimal, se indica por medio del subíndice 16.

Como se ve el número 3561,75 (y cualquier otro número) puede ser escrito la cantidad de maneras como sistemas de numeración existan.

Los sistemas de numeración que serán de mucho interés para este curso, y se dará mayor énfasis al sistema binario. Esto no significa que no exista una mayor cantidad de sistemas de numeración. Se acostumbra tomar del sistema decimal la cantidad de dígitos requeridos, si la base del número es menor a diez y utilizar las letras del alfabeto para complementar los diez dígitos decimales, si la base del número es mayor a diez.

### **2.1.3. Conversión de base numérica**

A veces es necesario convertir del sistema numérico decimal a otro sistema de numeración y viceversa. Inmediatamente se explicará los procedimientos para ambos casos.

#### **2.1.3.1. Conversión de base $r$ a decimal**

Como se ha podido apreciar la conversión de un número de base  $r$  a decimal, se efectúa expandiendo el número a una serie de potencias y sumando todos los términos. Y esto es así para cualquier sistema numérico que se utilice. Los pasos para convertir de una base  $r$  a sistema decimal son:

- Escribir el número  $r$  que se desea convertir.
- Si la cantidad por convertir tiene un punto, separar la cantidad que precede del punto, con la que prosigue del punto. Cada cantidad se convierte independientemente de la otra.

- Multiplicar cada casilla de número  $r$  por el número base elevado al exponente correspondiente a su casilla.
- Sumar los resultados.
- Situar las cantidades en su lugar (respecto del punto).

#### **2.1.3.1.1. Conversión de binario a decimal**

Los pasos son los siguientes:

- Escribir el número que se desea convertir.
- Si la cantidad por convertir tiene un punto, separar la cantidad que precede del punto, con la que prosigue del punto. Cada cantidad se convierte independientemente de la otra.
- Multiplicar cada casilla de número  $r$  por dos elevado al exponente correspondiente a su casilla.
- Sumar los resultados.
- Situar las cantidades en su lugar (respecto del punto).

#### **2.1.3.1.2. Conversión de octal a decimal**

Los pasos son los siguientes:

- Escribir el número  $r$  que se desea convertir
- Si la cantidad a convertir tiene un punto, separar la cantidad que precede del punto, con la que prosigue del punto. Cada cantidad se convierte independientemente de la otra.

- Multiplicar cada casilla de número  $r$  por ocho elevado al exponente correspondiente a su casilla.
- Sumar los resultados.
- Situar las cantidades en su lugar (respecto del punto).

### **2.1.3.1.3. Conversión de hexadecimal a decimal**

Los pasos son los siguientes:

- Escribir el número  $r$  que se desea convertir.
- Si la cantidad por convertir tiene un punto, separar la cantidad que precede del punto, con la que prosigue del punto. Cada cantidad se convierte independientemente de la otra.
- Multiplicar cada casilla de número  $r$  por dieciséis elevado al exponente correspondiente a su casilla.
- Sumar los resultados.
- Situar las cantidades en su lugar (respecto del punto).

### **2.1.3.2. Conversión de decimal a base $r$**

Ahora se presentará un procedimiento más general que será de convertir un número decimal a un número de base  $r$ . La conversión de un entero decimal a un número base  $r$ , se efectúa dividiendo el número y todos sus cocientes sucesivos entre  $r$  y acumulando los residuos.

### 2.1.3.2.1. Conversión de decimal a binario

Los pasos son los siguientes:

- Si la cantidad por convertir tiene un punto decimal, separar la cantidad que precede del punto, con la que prosigue del punto. Cada cantidad se convierte independientemente de la otra.
- Dividir la cantidad dentro de dos hasta que el cociente de la división sea uno.
- Por último, tomar el cociente de la última división y luego agregar los resultados de los módulos en orden de la última división realizada a la primera división realizada.
- Situar las cantidades en su lugar (con respecto al punto).

Ejemplo 5: escribir el número 521,36 en sistema numérico binario.

Se inicia convirtiendo el 521:

$$521 \div 2 = 260 \quad \text{mod} = 1$$

$$260 \div 2 = 130 \quad \text{mod} = 0$$

$$130 \div 2 = 65 \quad \text{mod} = 0$$

$$65 \div 2 = 32 \quad \text{mod} = 1$$

$$32 \div 2 = 16 \quad \text{mod} = 0$$

$$16 \div 2 = 8 \quad \text{mod} = 0$$

$$8 \div 2 = 4 \quad \text{mod} = 0$$

$$4 \div 2 = 2 \quad \text{mod} = 0$$

$$2 \div 2 = 1 \quad \text{mod} = 0$$

Al leer el resultado de abajo hacia arriba se obtiene:

$$(1000001001)_2$$

Se prosigue convirtiendo el 36:

$$36 \div 2 = 18 \quad \text{mod} = 0$$

$$18 \div 2 = 9 \quad \text{mod} = 0$$

$$9 \div 2 = 4 \quad \text{mod} = 1$$

$$4 \div 2 = 2 \quad \text{mod} = 0$$

$$2 \div 2 = 1 \quad \text{mod} = 0$$

Al leer el resultado de abajo hacia arriba se obtiene:

$$(100100)_2$$

Y al juntar ambas partes se obtiene lo siguiente:

$$521.36 = (1000001001.100100)_2$$

### **2.1.3.2.2. Conversión de decimal a octal**

Los pasos son los siguientes:

- Separar la cantidad antes del punto decimal de la cantidad luego del punto decimal.

- Dividir la cantidad dentro de ocho hasta que el cociente de la división sea menor a ocho.
- Por último, tomar el cociente de la última división y luego agregar los resultados de los módulos en orden de la última división realizada a la primera división realizada.
- Situar las cantidades en su lugar (respecto del punto).

Ejemplo 6: convertir el número 521,36 a su equivalente en el sistema de numeración octal.

Se inicia convirtiendo el 521:

$$521 \div 8 = 65 \quad \text{mod} = 1$$

$$65 \div 8 = 8 \quad \text{mod} = 1$$

$$8 \div 8 = 1 \quad \text{mod} = 0$$

Al leer el resultado de abajo hacia arriba se obtiene:

$$(1011)_8$$

Se prosigue convirtiendo el 36:

$$36 \div 8 = 4 \quad \text{mod} = 4$$

Al leer el resultado de abajo hacia arriba se obtiene:

$$(44)_8$$

Y al juntar ambas partes se obtiene:

$$521,36 = (1011,44)_8$$

### 2.1.3.2.3. Conversión de decimal a hexadecimal

Los pasos son los siguientes:

- Separar la cantidad antes del punto decimal de la cantidad luego del punto decimal.
- Dividir la cantidad dentro de dieciséis hasta que el cociente de la división sea menor a dieciséis.
- Luego tomar el cociente de la última división y agregar los resultados de los módulos en orden de la última división realizada a la primera división realizada.

Ejemplo 7: convertir el número 521,36 a su equivalente en el sistema de numeración hexadecimal. Se inicia convirtiendo el 521.

$$521 \div 16 = 32 \quad \text{mod} = 9$$

$$32 \div 16 = 2 \quad \text{mod} = 0$$

Al leer el resultado de abajo hacia arriba se obtiene:

$$(209)_{16}$$

Se prosigue convirtiendo el 36:

$$36 \div 16 = 2 \quad \text{mod} = 4$$

Al leer el resultado de abajo hacia arriba se obtiene:

$$(24)_{16}$$

Y al juntar ambas partes se obtiene:

$$521,36 = (209.24)_{16}$$

### **2.1.3.3. Relación de números binarios, octales y hexadecimales**

Las conversiones entre binarios, octales y hexadecimales desempeñan un papel importante en las computadoras digitales. Dado a que  $2^3 = 8$  y  $2^4 = 16$ , cada dígito octal corresponde a tres dígitos binarios y cada dígito hexadecimal corresponde a cuatro dígitos binarios.

Por esto la conversión del sistema de numeración binaria a uno de estos sistemas de numeración, es rápido y fácil de entender.

#### **2.1.3.3.1. Binario a octal**

Los pasos para la conversión del sistema binario al sistema octal son:

- Escribir el número binario que se desea convertir.
- Si la cantidad a convertir tiene un punto, separar la cantidad que precede del punto, con la que prosigue del punto. Cada cantidad se convierte independientemente de la otra.
- Para ambas cantidades, agrupar los dígitos en grupos de tres de derecha a izquierda.
- Por cada grupo en binario convertir a su equivalente en el sistema octal.
- Luego unificar ambas cantidades con el punto decimal. Cada una en la misma posición inicial.

Ejemplo 8: convertir del sistema de numeración binario al sistema de numeración octal  $(10110001101011,111100000110)_2$ .

Tabla III. **Tabla para resolución de ejemplo 8**

<b>Binario</b>	<b>Octal</b>
010	2
110	6
001	1
101	5
011	3
.	.
111	7
100	4
000	0
110	6

Fuente: elaboración propia, empleando Excel.

Parte antes del punto:  $(10110001101011)_2$

Parte después del punto:  $(111100000110)_2$

Parte antes del punto agrupada en 3 dígitos:  $(010\ 110\ 001\ 101\ 011)_2$

Parte después del punto agrupada en 3 dígitos:  $(111\ 100\ 000\ 110)_2$

Entonces se obtiene como resultado:  $(26153,7406)_8$

### 2.1.3.3.2. Binario a hexadecimal

Los pasos para la conversión del sistema binario al sistema hexadecimal son:

- Escribir el número binario que se desea convertir.
- Separar la cantidad antes del punto decimal de la cantidad después del punto. Esto porque se convertirán por separado.
- Para ambas cantidades, agrupar los dígitos en grupos de cuatro de derecha a izquierda.
- Por cada grupo en binario convertir a su equivalente en el sistema octal.
- Luego unificar ambas cantidades con el punto decimal. Cada una en la misma posición inicial.

Tabla IV. **Tabla para resolución de ejemplo 9**

Binario	Hexadecimal
10	2
1100	C
1100	6
1011	B
.	.
1111	F
0000	0
110	2

Fuente: elaboración propia, empleando Excel.

Ejemplo 9: convertir del sistema de numeración binaria al sistema de numeración hexadecimal  $(10110001101011.111100000110)_2$ .

Parte antes del punto:  $(10110001101011)_2$

Parte después del punto:  $(111100000110)_2$

Parte antes del punto agrupada en 4 dígitos:  $(10\ 1100\ 0110\ 1011)_2$

Parte después del punto agrupada en 4 dígitos:  $(1111\ 0000\ 0110)_2$

Entonces se obtiene como resultado:  $(2C6B.F02)_{16}$

Tabla V. **Tabla de conversión de números decimales, binarios, octales y hexadecimales**

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

Fuente: elaboración propia, empleando Excel.

### 2.1.3.3.3. Octal y hexadecimal a binario

Como se aprecia la conversión entre números binarios, octales y hexadecimales es sencilla; y se puede volver más sencilla con la ayuda de la tabla V. La conversión de octal o hexadecimal a binario, se hace invirtiendo el procedimiento anterior. Cada dígito octal se convierte a su equivalente a binario de tres dígitos. Asimismo, cada dígito hexadecimal se convierte en su equivalente binario de cuatro dígitos.

Es difícil trabajar con números binarios porque se componen con más dígitos que sus equivalentes decimales. Por ejemplo, el número binario  $111111111111_2$  equivale a 4095 decimal. No obstante, las computadoras digitales emplean números binarios y hay ocasiones que el operador necesita comunicarse directamente con la máquina empleando números binarios. Aprovechando la relación con los sistemas binarios, octal y hexadecimal, el usuario piensa en términos de números octales y hexadecimales, y efectúa la conversión requerida por inspección cuando se hace necesaria la comunicación directa con los sistemas digitales.

Tabla VI. **Tabla para resolución de ejemplo 10**

Hexadecimal	Binario
3	11
0	0
6	0110
.	.
D	1101

Fuente: elaboración propia, empleando Excel.

Ejemplo 10: convertir  $(306.D)_{16}$  al sistema de numeración binaria.

$(306.D)_{16}$

#### **2.1.4. Operaciones aritméticas de números binario, octales y hexadecimales**

Cada sistema numérico tiene una leve diferencia en el procedimiento de operaciones aritméticas, pero se dará cuenta que los procesos son similares a los que utilizan en el sistema decimal. Las operaciones que se estudiarán para cada sistema numérico son:

- Suma
- Resta
- Multiplicación
- División

##### **2.1.4.1. Aritmética binaria**

Ahora se explicarán la suma, resta, multiplicación y división en el sistema de numeración binario.

##### **2.1.4.1.1. Suma**

Los pasos para resolver una sumatoria en el sistema de numeración binario son similares a los del sistema de numeración decimal.

Tabla VII. **Datos importantes para la suma de números binarios**

Datos de suma
0+0=0
1+0=1
0+1=1
1+1=0 y se lleva 1

Fuente: elaboración propia, empleando Excel.

Ejemplo 11: realizar la siguiente suma:  $(11011)_2 + (11100)_2$ .

$$\begin{array}{r}
 \phantom{+} 1 \phantom{1} \phantom{0} \phantom{1} \phantom{1} \\
 1 \phantom{1} 0 1 1 \\
 + 1 1 1 0 0 \\
 \hline
 1 1 0 1 1 1
 \end{array}$$

Hay que colocar los dos números a sumar uno encima del otro y luego se irá sumando los números de derecha a izquierda. Si hay una suma de 1+1, el uno que se lleva hay que sumarlo en la casilla izquierda siguiente.

Una manera de corroborar el resultado es de convertir los números a decimal y ver si es el mismo resultado sumando los números decimales.

#### 2.1.4.1.2. Resta

Como se verá en el ejemplo 12 todos los números que están en azul representan ya sea el minuendo o sustraendo y lo que está en color rojo representa los números “devueltos” tras la operación 0-1 y los resultados de las operaciones con los números “devueltos”. El “devuelvo 1” significa que se llevará un número uno imaginario a la siguiente casilla.

Ejemplo 12: realizar la siguiente resta:  $(111100)_2 - (110011)_2$

$$\begin{array}{r} 111100 \\ - 110011 \\ \hline 001001 \end{array}$$

Para la primera casilla:

$$\begin{array}{r} 1 \\ 0 \\ - \frac{1}{1} \end{array} \rightarrow \begin{array}{l} 0 - 1 = 1 \text{ y devuelvo } 1 \\ 1 \end{array}$$

Para la segunda casilla:

$$\begin{array}{r} 1 \\ 0 \\ - \frac{1}{0} \end{array} \rightarrow \begin{array}{l} 0 - 1 = 1 \text{ y devuelvo } 1 \\ 1 - 1 = 0 \\ 0 \end{array}$$

Para la tercera casilla:

$$\begin{array}{r} 1 \\ 1 \\ - \frac{0}{0} \end{array} \rightarrow \begin{array}{l} 1 - 1 = 0 \text{ y devuelvo } 1 \\ 0 - 0 = 0 \\ 0 \end{array}$$

Para la cuarta casilla:

$$\begin{array}{r} 1 \\ - \frac{0}{1} \end{array} \rightarrow \begin{array}{l} 1 - 0 = 1 \\ 1 \end{array}$$

Para la quinta y sexta casilla:

$$\begin{array}{r} 1 \\ - \frac{1}{0} \end{array} \rightarrow \begin{array}{l} 1 - 1 = 0 \\ 0 \end{array}$$

Tabla VIII. **Datos importantes para la resta de números binarios**

<b>Datos de resta</b>
0-0=0
1-0=1
0-1=1 y devuelvo 1
1-1=0

Fuente: elaboración propia, empleando Excel.

Ahora se ve que cuando ocurre 0-1 hay que hacer otras operaciones extras aparte de las del minuendo y sustrayendo iniciales. Cuando se está trabajando con el minuendo y los números uno devueltos, se debe usar como minuendo (para las operaciones extras) el número de la casilla actual y como sustraendo el uno devuelto; la diferencia se utilizará en la siguiente casilla. Cuando se trabaja con el sustraendo, el minuendo es la diferencia de la operación anterior y el sustrayendo es el número de la casilla actual; la diferencia será la que exista (la resta en general) en la casilla.

#### **2.1.4.1.3. Multiplicación**

Como se verá en el procedimiento del ejemplo 13 los pasos (comparados con los de la multiplicación en el sistema decimal) son los mismos. De igual manera se suman al final todos los números con las mismas reglas de la suma.

Tabla IX. **Datos importantes para la multiplicación de números binarios**

Datos de multiplicación
$0*0=0$
$1*0=0$
$0*1=0$
$1*1=1$

Fuente: elaboración propia, empleando Excel.

Ejemplo 13: realizar la siguiente multiplicación:  $(1010)_2 * (110)_2$ .

$$\begin{array}{r}
 1010 \\
 \times 110 \\
 \hline
 0000 \\
 1010 \\
 \hline
 1010 \\
 \hline
 111100
 \end{array}$$

Como se aprecia el procedimiento comparado con el sistema decimal es el mismo. De igual manera se suman al final todos los números con las mismas reglas de la suma.

#### 2.1.4.1.4. División

Para la división se debe recordar las reglas de la multiplicación y las reglas de la resta. Como se verá en el procedimiento del ejemplo 14 los pasos (comparados con los de la división en el sistema decimal) son los mismos. De igual manera se va construyendo el cociente; de izquierda a derecha. Cada resta realizada se hace con las reglas de la resta ya establecidas.



### 2.1.4.2.1. Suma

Ejemplo 15: realizar la siguiente suma:  $(405)_8 + (753)_8 + (766)_8$ .

$$\begin{array}{r} 2 \ 1 \ 1 \\ 4 \ 0 \ 5 \\ 7 \ 5 \ 3 \\ + 7 \ 6 \ 6 \\ \hline 2 \ 3 \ 4 \ 6 \end{array}$$

Para la primera casilla se utiliza la tabla X para ir recorriendo los números iniciando desde el cero. Cada vez que se pase por el número cero se generará un acarreo de uno.

### 2.1.4.2.2. Resta

Ejemplo 16: restar los siguientes números:  $(4327)_8 - (1541)_8$ .

$$\begin{array}{r} 8 \ 8 \\ 4 \ 3 \ 2 \ 7 \\ - 1 \ 5 \ 4 \ 1 \\ \hline 2 \ 5 \ 6 \ 6 \end{array}$$

El procedimiento es muy similar a restar números decimales, pero con la diferencia que al momento de prestar un número (cuando el número entre casillas es mayor el sustraendo que el minuendo), no será diez, sino que será ocho.

### 2.1.4.2.3. Multiplicación

La multiplicación es del mismo modo que en el sistema decimal, pero con la siguiente variación.

Hay que restar ocho a nuestro resultado de multiplicación la cantidad de veces necesarias para que el resultado sea menor a ocho. Cada vez que se hace una resta, se generará un acarreo. Para simplificar este procedimiento se puede dividir dentro de 8 el resultado de cada multiplicación y el cociente sería la cantidad de acarreos y el residuo el número a escribir en la casilla correspondiente. Con la ayuda del siguiente ejemplo, esto quedará más claro.

Ejemplo 17: multiplicar los siguientes números:  $(354)_8 * (23)_8$

$$\begin{array}{r}
 1\ 2\ 1 \\
 3\ 5\ 4 \\
 \times 2\ 3 \\
 \hline
 1\ 3\ 0\ 4 \\
 7\ 3\ 0 \\
 \hline
 1\ 0\ 6\ 0\ 4
 \end{array}$$

Para la primera multiplicación:

$$3 * 4 = 12$$

$12 - 8 = 4$  y llevo "1" porque doce sobrepasa una vez la base 8.

Para la segunda multiplicación:

$$3 * 5 = 15$$

$15 + 1 = 16$  El "1" se agregó de la primera multiplicación.

$16 - 8 = 8$  Se lleva "1" pero aún se sobrepasa, se debe volver a restar.

$8 - 8 = 0$  Se lleva otro "1"; entonces se llevan 2.

Para la tercera multiplicación

$$3 * 3 = 9$$

$9 + 2 = 11$  El "2" se agregó de la segunda multiplicación.

$11 - 8 = 3$  Se lleva "1" porque se sobrepasó una vez la base 8.

Para la multiplicación con el número 2 se seguirá el mismo procedimiento, y para la suma de ambos resultados se utilizará las reglas de la suma previamente establecidas.

#### 2.1.4.2.4. División

La división sigue básicamente las mismas reglas del sistema de numeración decimal, pero hay que tomar en cuenta que se está en el sistema de numeración octal y cada operación que sea necesario hacer en la división (resta y multiplicación) será en base a las reglas ya preestablecidas.

Ejemplo 18: realizar la siguiente división:  $(65431)_8 / (45)_8$ .

$$\begin{array}{r} 1344 \\ 45 \overline{) 65431} \\ \underline{- 45} \phantom{1} \\ 204 \phantom{1} \\ \underline{- 157} \phantom{1} \\ 0253 \phantom{1} \\ \underline{- 224} \phantom{1} \\ \phantom{0}271 \phantom{1} \\ \underline{- 271} \phantom{1} \\ \phantom{00}000 \end{array}$$

Tabla XI. **Tabla para suma de números hexadecimales**

0	
1	
2	
3	
4	
5	
6	
7	
8	
9	
A	10
B	11
C	12
D	13
E	14
F	15

Fuente: elaboración propia, empleando Excel.

### **2.1.4.3. Aritmética hexadecimal**

Las operaciones en el sistema numérico hexadecimal son muy parecidas a los sistemas ya vistos, pero se debe recordar el uso de los dieciséis dígitos que son utilizados en este sistema de numeración. Se utilizará la tabla XI para las operaciones aritméticas en sistema hexadecimal. La tabla se recorrerá de forma ascendente numéricamente hablando, y cada vez que en una operación pase por 0, se generará un acarreo.



Cada vez que se hace una resta se tendrá un acarreo. Para simplificar este procedimiento se podrá dividir dentro de dieciséis el resultado de cada multiplicación y el cociente será la cantidad de acarreos y el residuo el número por escribir.

Ejemplo 21: realizar la siguiente multiplicación  $(5FA)_{16} * (F4)_{16}$

$$\begin{array}{r}
 5 \ E \ 9 \quad \text{Para F} \\
 1 \ 3 \ 2 \quad \text{Para 4} \\
 \hline
 \begin{array}{r}
 5 \ F \ A \\
 X \quad F \ 4 \\
 \hline
 1 \ 1 \quad \leftarrow \text{Acarreo} \\
 1 \ 7 \ E \ 8 \\
 + \ 5 \ 9 \ A \ 6 \\
 \hline
 5 \ B \ 2 \ 4 \ 8
 \end{array}
 \end{array}$$

Para la primera multiplicación:

$$4 * A = 40$$

$$40/16 = 2 \text{ Con residuo } 8 \text{ y se lleva "2"}$$

Para la segunda multiplicación:

$$4 * F = 60$$

$$60 + 2 = 62 \text{ El "2" se agregó de la primera multiplicación.}$$

$$62/16 = 3 \text{ Con residuo } E \ (14), \text{ se lleva "3"}$$

Para la tercera multiplicación

$$4 * 5 = 20$$

$$20 + 3 = 23 \text{ El "3" se agregó de la segunda multiplicación.}$$

$$23/16 = 1 \text{ con residuo } 7 \text{ y se lleva "1" porque sobrepasó una vez la base.}$$

Para la multiplicación con el número 2 se seguirá el mismo procedimiento, y para la suma de ambos resultados se utilizará las reglas de la suma.

#### 2.1.4.3.4. División

La división sigue básicamente las mismas reglas del sistema de numeración decimal, pero hay que tomar en cuenta que se utiliza el sistema de numeración hexadecimal y cada operación que sea necesaria hacer en la división (resta y multiplicación) será en base a las reglas ya preestablecidas.

Ejemplo 22: realizar la siguiente división:  $(F5C1)_{16}/(A3)_{16}$

$$\begin{array}{r}
 \phantom{A} 3 \overline{) F5C1} \\
 \underline{- A3} \phantom{0} \\
 52C \\
 \underline{- 518} \\
 0141 \\
 \underline{\phantom{0}A3} \\
 09E
 \end{array}$$

Es el mismo procedimiento que en el sistema decimal, solo hay que recordar que se utiliza el sistema hexadecimal y que se hacen las restas y multiplicaciones con las reglas ya establecidas.

#### 2.1.5. Complementos

En las computadoras digitales se usan complementos para simplificar la operación de resta y para efectuar manipulaciones lógicas. Hay dos tipos de complementos para cada sistema base  $r$ : el complemento a la base y el

complemento a la base disminuida. El primero se le denomina complemento a  $r$ , mientras que al segundo es el complemento a  $(r-1)$ .

Tabla XII. **Complementos a la base  $r$  y a la base disminuidas de sistemas de numeración**

	Complemento base $r$	Complemento base disminuida
<b>Decimal</b>	10	9
<b>Binario</b>	2	1
<b>Octal</b>	8	7
<b>Hexadecimal</b>	16	15

Fuente: elaboración propia, empleando Excel.

Tabla XIII. **Fórmula de complementos a la base disminuida**

Sistema numérico	Fórmula
Decimal	$(10^n - 1) - N$
Binario	$[(10^n - 1)_2 - N]_2$
Octal	$[(10^n - 1)_8 - N]_8$
Hexadecimal	$[(10^n - 1)_{16} - N]_{16}$

Fuente: elaboración propia, empleando Excel.

### 2.1.5.1. Complemento a la base disminuida

El complemento a la base disminuida de un número  $N$  que tiene  $n$  dígitos y una base  $r$  está definido a continuación. Con ello se puede definir la tabla XIII.

$$(r^n - 1) - N$$

#### 2.1.5.1.1. Números decimales

Ejemplo 23: comprobar el complemento a la base disminuida del número 566.

$$r = 10$$

$$n = 3$$

$$N = 566$$

$$(10^3) - N = 1000000 - 104513 = 895487$$

Entonces el complemento a 9 de 566 es 433.

#### 2.1.5.1.2. Números binarios

El complemento de base disminuida es utilizar la segunda fórmula de la Tabla XIII. Otra manera de obtenerlo es invirtiendo cada uno de los dígitos del número.

Ejemplo 24: comprobar el complemento a la base disminuida del número  $(1000110110)_2$ .

Para  $(1000110110)_2$

$$r = 2$$

$$n = 10$$

$$N = 1000110110$$

Así se reemplaza en la expresión anterior:

$$\begin{aligned}(10^{10} - 1) - 1000110110 &= 10000000000 - 1 - 1000110110 \\ &= 111111111 - 1000110110 = 0111001001\end{aligned}$$

Entonces el complemento a 1 de  $(1000110110)_2$  es  $(0111001001)_2$ . Como se puede observar el complemento a uno de cualquier número binario es invertir cada uno de los números.

### 2.1.5.1.3. Números octales

El complemento de base disminuida es utilizar la tercera fórmula de la tabla XIII.

Otra manera de obtener el complemento a siete de un número octal es restar el número octal a un número cuya cantidad de sietes, sea igual a la cantidad de dígitos del número octal.

Ejemplo 25: comprobar el complemento a la base disminuida del número  $(31474)_8$ .

Ahora se resta cada dígito a 7.

$$\begin{array}{r} 7\ 7\ 7\ 7\ 7 \\ -\ 3\ 1\ 4\ 7\ 4 \\ \hline 4\ 6\ 3\ 0\ 3 \end{array}$$

Entonces el complemento a siete de  $(31477)_8$  es  $(46303)_8$ . También se pudo haber utilizado la fórmula de la tabla XIII, dará el mismo resultado.

### 2.1.5.1.4. Números hexadecimales

El complemento de base disminuida es utilizar la cuarta fórmula de la tabla XIII. Otra manera de obtener el complemento a quince de un número hexadecimal es restar el número hexadecimal a un número cuya cantidad de efes (F), sea igual a la cantidad de dígitos del número hexadecimal.

Ejemplo 26: comprobar el complemento a la base disminuida del número  $(E5C1AB0)_{16}$ .

Ahora se resta cada dígito a F.

$$\begin{array}{r} F F F F F F F \\ - E 5 C 1 A B 0 \\ \hline 1 A 2 E 4 3 F \end{array}$$

Entonces, el complemento a 15 de  $(E5C1AB0)_{16}$  es  $(1A2E43F)_{16}$ . También se pudo haber utilizado la fórmula de la tabla XIII, dará el mismo resultado.

Tabla XIV. **Fórmula de complementos a la base disminuida**

Sistema numérico	Fórmula
Decimal	$10^n - N$
Binario	$[10^n - N]_2$
Octal	$[10^n - N]_8$
Hexadecimal	$[10^n - N]_{16}$

Fuente: elaboración propia, empleando Excel.

### 2.1.5.2. Complemento a la base

El complemento a  $r$  de un número  $N$  de  $n$  dígitos es:

$$r^n - N$$

Y se aprecia que sería el complemento de base disminuida más 1.

$$r^n - N = [(r^n - 1) - N] + 1$$

Con ello se puede obtener las fórmulas de la tabla XIV. Hay que recordar que el diez a la  $n$  está en el sistema de numeración decimal.

Como se vio, el complemento de base disminuida más uno es igual al complemento de la base, entonces para calcular el mismo simplemente se tendría que sumar un uno al complemento de base disminuida.

#### 2.1.5.2.1. Números decimales

Para el sistema decimal se puede obtener el complemento de la base de otra manera. Puede formarse dejando como están todos los ceros menos significativos, restando a 10 el primer dígito menos significativo distinto de cero, y restando a nueve los demás dígitos de la izquierda.

Ejemplo 27: obtener el complemento de la base de 104513 y de 257600.

$$\begin{array}{r} 9\ 9\ 9\ 9\ 9\ 10 \\ -\ 1\ 0\ 4\ 5\ 1\ 3 \\ \hline 8\ 9\ 5\ 4\ 8\ 7 \end{array} \qquad \begin{array}{r} 9\ 9\ 9\ 10\ 0\ 0 \\ -\ 2\ 5\ 7\ 6\ 0\ 0 \\ \hline 7\ 4\ 2\ 4\ 0\ 0 \end{array}$$

Para 104513 su complemento a la base es 895487.

Para 257600 su complemento a la base es 742400.

#### **2.1.5.2.2. Números binarios**

De forma similar, el complemento a dos se forma dejando como están todos los ceros menos significativos y el primer uno, y sustituyendo los unos por ceros y los ceros por unos en las demás posiciones a la izquierda.

También se pudo haber utilizado la segunda fórmula de la tabla XIV para su cálculo.

Ejemplo 28: obtener el complemento de la base de  $(10111010)_2$  y de  $(01100111)_2$ .

Para  $(10111010)_2$  su complemento a la base es  $(01000110)_2$

Para  $(01100111)_2$  su complemento a la base es  $(10011001)_2$

#### **2.1.5.2.3. Números octales**

En este caso se calculará el complemento de la base disminuida y posteriormente se le sumará uno. El uno que se agrega es con numeración octal.

Ejemplo 29: obtener el complemento a la base de  $(54723)_8$ .

$$\begin{array}{r}
 7\ 7\ 7\ 7\ 7 \\
 -\ 5\ 4\ 7\ 2\ 3 \\
 \hline
 2\ 3\ 0\ 5\ 4
 \end{array}$$

$$23054 + 1 = 23055$$

Para  $(54723)_8$  el complemento de la base es  $(23055)_8$

#### 2.1.5.2.4. Números hexadecimales

En este caso se calculará el complemento de la base disminuida y posteriormente se le sumará uno, el uno que se agrega es con numeración hexadecimal.

Ejemplo 30: obtener el complemento de la base de  $(FE0505)_{16}$ .

$$\begin{array}{r}
 F\ F\ F\ F\ F\ F \\
 -\ F\ E\ 0\ 5\ 0\ 5 \\
 \hline
 0\ 1\ F\ A\ F\ A
 \end{array}$$

$$01FAFA + 1 = 01FAFB$$

Para  $(FE0505)_{16}$  el complemento de la base es  $(01FAFB)_{16}$

#### 2.1.5.3. Resta con complementos

Cuando se realiza una resta a lápiz se tiene el concepto de préstamos, donde se pide prestado un uno de la siguiente posición cuando el dígito del minuendo es menor que el del sustraendo. El método funciona bien cuando se resta con lápiz y papel, pero cuando la resta se implementa en *hardware* digital, el método es más eficiente si se usan complementos. La resta de dos números de  $n$  dígitos,  $M - N$  en base  $r$  se efectúa así:

- Sumar el minuendo  $M$  el complemento a  $r$  del sustraendo  $N$ . Esto es  $M + (r^n - N) = M + r^n - N$ .  $n$  siempre será la cantidad de dígitos del número más grande.
- Si  $M \geq N$  la suma producirá un acarreo final  $r^n$  que puede desecharse; lo que queda es el resultado  $M - N$ .
- Si  $M < N$  la suma no produce un acarreo final y es igual a  $r^n - (N - M) = M + (r^n - N)$ , que es el complemento a  $r$  de  $(N - M)$ . Para obtener la respuesta en una forma conocida, se toma el complemento a  $r$  de la suma y se le antepone un signo menos.

Ejemplo 31: restar  $45466 - 775897$  usando complemento a 10.

$$M = 45466 ; N = 775897 ; n = 6$$

$$M < N$$

M=	0 4 5 4 6 6
Complemento a 10 de N=	+ 2 2 4 1 0 3
Suma	2 6 9 5 6 9

$$1000000 - 269569 = -730431$$

Se puede observar que el procedimiento final es obtener el complemento a diez de la suma y agregar signo negativo. Puesto a que está manejando números sin signo, en realidad es imposible obtener un resultado sin signo para este caso.

Ejemplo 32: restar  $775897 - 45466$  usando complemento a 10.

$$M = 775897 ; N = 45466 ; n = 6$$

$$M \geq N$$

$$\begin{array}{r}
M = \quad 7 \ 7 \ 5 \ 8 \ 9 \ 7 \\
\text{Complemento a 10 de } N = \quad + \ 9 \ 5 \ 4 \ 5 \ 3 \ 4 \\
\text{Total} \quad \quad \quad 1 \ 7 \ 3 \ 0 \ 4 \ 3 \ 1 \\
\text{Desechar acarreo final } 10^6 \quad - \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \\
\text{Respuesta} \quad \quad \quad 7 \ 3 \ 0 \ 4 \ 3 \ 1
\end{array}$$

Ejemplo 33: teniendo los números binarios  $A = 111011$  y  $B = 10001$  realizar las siguientes restas empleando complemento a dos:  $A - B$  y  $B - A$ .

Para  $A - B$ :

$$\begin{array}{r}
A \geq B \\
A = \quad 1 \ 1 \ 1 \ 0 \ 1 \ 1 \\
\text{Complemento a 2 de } B = \quad + \quad 0 \ 1 \ 1 \ 1 \ 1 \\
\text{Total} \quad \quad \quad 1 \ 0 \ 0 \ 1 \ 0 \ 1 \ 0 \\
\text{Desechar acarreo final } 2^6 \quad - \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \\
\text{Respuesta} \quad \quad \quad 0 \ 0 \ 1 \ 0 \ 1 \ 0
\end{array}$$

Para  $B - A$ :

$$\begin{array}{r}
A < B \\
B = \quad 0 \ 1 \ 0 \ 0 \ 0 \ 1 \\
\text{Complemento a 2 de } A = \quad + \ 0 \ 0 \ 0 \ 1 \ 0 \ 1 \\
\text{Suma} \quad \quad \quad 0 \ 1 \ 0 \ 1 \ 1 \ 0 \\
100000 - 10110 = -1010
\end{array}$$

Se puede observar que para ambos casos es el mismo procedimiento que utilizando números decimales. La resta de números sin signo también se puede efectuar usando el complemento a  $(r - 1)$ ; tanto para números decimales como para binarios. Esto sería a complemento a nueve y complemento a uno respectivamente.

Ejemplo 34: repetir lo mismo para el ejemplo anterior solo que utilizando complemento a 1.

Para  $A - B$ :

$$\begin{array}{r}
 A \geq B \\
 A = \quad \quad \quad 1 \ 1 \ 1 \ 0 \ 1 \ 1 \\
 \text{Complemento a 1 de } B = \quad + \quad \quad 0 \ 1 \ 1 \ 1 \ 1 \\
 \hline
 \text{Total} \quad \quad \quad 1 \ 0 \ 0 \ 1 \ 0 \ 1 \ 0 \\
 \text{Acarreo} \quad + \quad \quad \quad \quad \quad \quad \quad 1 \\
 \text{Desechar acarreo final } 2^6 \quad - \quad 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \\
 \hline
 \text{Respuesta} \quad \quad \quad 0 \ 0 \ 1 \ 0 \ 1 \ 1
 \end{array}$$

Para  $B - A$ :

$$\begin{array}{r}
 B < A \\
 B = \quad \quad \quad 0 \ 1 \ 0 \ 0 \ 0 \ 1 \\
 \text{Complemento a 1 de } A = \quad + \quad 0 \ 0 \ 0 \ 1 \ 1 \ 0 \\
 \hline
 \text{Suma} \quad \quad \quad 0 \ 1 \ 0 \ 1 \ 1 \ 1 \\
 100000 - 10111 = -1011
 \end{array}$$

### 2.1.6. Números binarios con signo

Por limitaciones de hardware las computadoras deben representar todo en dígitos binarios. Los enteros positivos incluido el cero se representan como números sin signo. Sin embargo, para representar números enteros negativos se necesita una notación que distinga a los valores negativos.

Tabla XV. **Ejemplo de números decimales y números binarios con y sin signo**

Binario	Decimal	
	Con signo	Sin signo
1001	-1	9
1111	-7	15
1010	-2	10
1011	-3	11
0100	+4	4
0011	+3	3
0111	+7	7
1101	-5	13

Fuente: elaboración propia, empleando Excel.

Tabla XVI. **Ejemplo de notación con complemento a uno**

Binario	Decimal
00000000	0
00000001	1
00000010	2
...	...
01111101	125
01111110	126
01111111	127
10000000	-127
10000001	-126
10000010	-125
...	...
11111101	-2
11111110	-1
11111111	-0

Fuente: elaboración propia, empleando Excel.

Es importante aclarar que los números binarios, tanto con signo como sin él, se representan en las computadoras con una cadena de bits. El usuario determina si el número tiene signo o no.

Tabla XVII. **Ejemplo con notación con complemento a dos**

<b>Binario</b>	<b>Decimal</b>
00000000	0
00000001	1
00000010	2
...	...
01111101	125
01111110	126
01111111	127
10000001	-127
10000010	-126
10000011	-125
...	...
11111110	-2
11111111	-1

Fuente: elaboración propia, empleando Excel.

### **2.1.6.1. Notación con signo magnitud**

Se acostumbra representar el signo con un bit colocado en la posición extrema izquierda del número. La interpretación más general es que si el bit es cero, el número es positivo y si es uno, el número es negativo.

Si el número binario posee signo, el bit de la extrema izquierda representará el signo y el resto de los bits representaran el número. Si se supone que el número binario carece de signo, el bit de la extrema izquierda será el bit más significativo del número. Ejemplo en la tabla XV.

### 2.1.6.2. Notación con complemento a uno

Esta notación consiste en que el negativo de un número N será el complemento a uno de dicho número N. Recordar que el complemento a uno de un número binario es el inverso de cada uno de sus dígitos. Ejemplo en la tabla XVI.

Tabla XVIII. Ejemplo con notación en exceso a K

Binario	Decimal
00000000	-127
00000001	-126
00000010	-125
...	...
01111110	-1
01111111	0
10000000	1
10000001	2
10000010	3
...	...
11111101	126
11111110	127
11111111	128

Fuente: elaboración propia, empleando Excel.

### **2.1.6.3. Notación con complemento a dos**

Esta notación consiste en que el negativo de un número N será el complemento a dos de dicho número N. Recordar que el complemento a dos de un número binario es el complemento a uno, más uno. Ejemplo en la tabla XVII.

### **2.1.6.4. Notación en exceso a K**

Esta notación consiste en sumar un valor K a todos los números. El valor K no tiene un valor estandarizado, pero usualmente se utilizan los siguientes valores:

$$2^{n-1}$$
$$2^{n-1} - 1$$

Donde n es la cantidad de dígitos del número binario. Por ejemplo, si se va a utilizar dos a la n menos 1, menos 1 como K, se va a sumar 2 a la n menos 1 menos 1 a cada uno de los números que se desean representar, y a partir de ello se van formando los positivos y negativos. Ejemplo en la tabla XVIII.

### **2.1.7. Códigos binarios**

Los sistemas digitales emplean señales que tienen dos valores distintos, y elementos que tienen dos estados estables. Existe una analogía directa entre señales binarias, elementos binarios de circuito y dígitos binarios. Un número binario de n dígitos, por ejemplo, podría representarse con n elementos binarios de circuitos, cada uno de los cuales tiene la señal de salida equivalente a 1 o 0.

Los sistemas digitales representan y manipulan no solo números binario, sino también muchos otros elementos discretos de información. Cualquier elemento discreto de información dentro de un grupo de cantidades puede ser representado con un código binario. Los códigos deben estar en binario porque las computadoras solo pueden almacenar unos y ceros. Se debe entender que los códigos binarios simplemente cambian los símbolos, no el significado de los elementos de información que representan.

### 2.1.7.1. Binario natural

Consiste en representar directamente el número decimal en binario, o lo que es lo mismo cada número decimal tiene su correspondiente en binario.

Es la codificación que hasta ahora se ha conocido. Ejemplos en la tabla XIX.

Tabla XIX. **Dígitos en codificación binario natural**

Binario Natural	Decimal
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	10
...	...

Fuente: elaboración propia, empleando Excel

### 2.1.7.2. Código BCD 8421

El código BCD cuyas siglas tienen un origen inglés (*Binary Coded Decima*) es la codificación de un número decimal a uno binario. La diferencia que posee con el código binario natural, es que cada dígito del número decimal, es representado por cuatro dígitos binarios. Por ejemplo, se puede representar el número 145 como:  $(0001_2 0100_2 0101_2)$ .

Tabla XX. Dígitos en codificación BCD

Binario	Decimal
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9

Fuente: elaboración propia, empleando Excel.

El código BCD se usa en dispositivos digitales en donde los datos de entrada se generan en un teclado decimal y las salidas se muestran en pantallas numéricas, como son las calculadoras digitales, relojes, multímetros, variadores de frecuencia entre otros. Las computadoras digitales modernas no procesan en BCD por estos motivos:

- Para representar un número en BCD se requieren más bits que para un número representado en binario natural.
- Las operaciones aritméticas son más complicadas que en binario natural.

Ejemplo 35: representar los números 10, 12, 497, 1526 en código BCD.

$$10 = 0001\ 0000$$

$$12 = 0001\ 0010$$

$$400 = 0100\ 0000\ 0000$$

$$1526 = 0001\ 0101\ 0010\ 0110$$

Tabla XXI. **Dígitos en codificación 2421**

Código 2421	Decimal
0000	0
0001	1
0010	2
0011	3
0100	4
1011	5
1100	6
1101	7
1110	8
1111	9

Fuente: elaboración propia, empleando Excel.

### 2.1.7.3. Código 2421

También llamado código Aiken, es un código que también utiliza cuatro dígitos binarios para representar un dígito decimal. Cada posición de cada dígito binario tiene un valor en específico. La posición menos significativa tiene un valor de uno, una posición mayor tiene un valor de 2, la siguiente un valor de cuatro y la posición más significativa tiene un valor de 2. La razón de esta codificación es conseguir simetría entre ciertos números; entre: cuatro y cinco; tres y seis; dos y siete; uno y ocho y entre cero y nueve. En la tabla XXI se aprecia que las parejas de números mencionadas se relacionan invirtiendo cada uno de sus dígitos, por ejemplo, los números tres ( $0011_2$ ) y seis ( $1100_2$ ). Esta codificación es útil para la realización de resta y división.

Tabla XXII. Dígitos en codificación XS3

Código XS3	Decimal
0011	0
0100	1
0101	2
0110	3
0111	4
1000	5
1001	6
1010	7
1011	8
1100	9

Fuente: elaboración propia, empleando Excel.

#### 2.1.7.4. Código exceso 3

Es otro código BCD común, a menudo se abrevia como XS3. Este código representa a un número decimal en 4 bits, solo que se le añade tres a cada dígito decimal antes de efectuar la conversión, por ejemplo, el cero se en codifica en exceso 3 como 0011. Se ven ejemplos en la tabla XXII.

#### 2.1.7.5. Código 8 4 -2 -1

Es un ejemplo de la asignación de pesos tanto positivos como negativos a un código decimal. Es este caso, la combinación de bits 0110 se interpreta como un dos en el sistema de numeración decimal y se calcula de la siguiente manera:  $8 * 0 + 4 * 1 + (-2) * 1 + (-1) * 0 = 2$ . Se ven ejemplos en la tabla XXIII.

Tabla XXIII. Dígitos en codificación 8 4 -2 -1

Código 84-2-1	Decimal
0000	0
0111	1
0110	2
0101	3
0100	4
1011	5
1010	6
1001	7
1000	8
1111	9

Fuente: elaboración propia, empleando Excel.

### 2.1.7.6. Código Gray

En este código existe solo un cambio de un bit entre dos números sucesivos. Los códigos que tienen estas características generalmente sus aplicaciones se extienden a los campos de la instrumentación, transductores, convertidores analógicos/digital, en codificadores de desplazamiento lineal y angular. El código Gray se emplea en aplicaciones en las que la sucesión normal de números binarios podría provocar error o ambigüedad durante la transacción de un número al siguiente. La ventaja del código Gray sobre la sucesión continua de números binarios es que la diferencia entre dos números consecutivos cualesquiera en código Gray es de un solo bit. Por ejemplo, al cambiar del 7 al 8 en código Gray cambia de 0100 a 1100.

Tabla XXIV. **Tabla de dígitos en codificación Gray**

Código Gray	Decimal	Código Gray	Decimal
0000	0	1100	8
0001	1	1101	9
0011	2	1111	10
0010	3	1110	11
0110	4	1010	12
0111	5	1011	13
0101	6	1001	14
0100	7	1000	15

Fuente: elaboración propia, empleando Excel.

### 2.1.7.7. Código ASCII

Muchas aplicaciones de las computadoras digitales requieren manipulación de datos que no solo son números, sino también letras. En un esfuerzo por estandarizar los códigos de intercambio de información, los fabricantes de equipo relacionado a esta rama acordaron usar el código ASCII (*American Standard Code for Information Interchange*). Este código puede representar hasta 128 caracteres diferentes y usa 7 bits.

Tabla XXV. Lista de código ASCII A

		CONTROL		CARACTERES ALFANUMERICOS						
		0	1	2	3	4	5	6	7	
765 ↓ 4321 →	000	NUL	DLE	SP	0	@	P	'	p	
	0001	SOH	DC1	!	1	A	Q	a	q	
	0010	STX	DC2	"	2	B	R	b	r	
	0011	ETX	DC3	#	3	C	S	c	s	
	0100	EOT	DC4	\$	4	D	T	d	t	
	0101	ENQ	NAK	%	5	E	U	e	u	
	0110	ACK	SYN	&	6	F	V	f	v	
	0111	BEEP	ETB	'	7	G	W	g	w	
	1000	BS	CAN	(	8	H	X	h	x	
	1001	HT	EM	)	9	I	Y	i	y	
	1010	LF	SUB	*	:	J	Z	j	z	
	1011	VT	ESC	+	:	K	[	k	{	
	1100	FF	FS	,	<	L	\	l	:	
	1101	CR	GS	-	=	M	]	m	}	
	1110	SO	RS	.	>	N	^	n	~	
	1111	SI	US	/	?	O	_	o	DEL	

Fuente: LEAL CHAPA, César Augusto. *Fundamentos de diseño digital*. p 96.

Tabla XXVI. Lista de código ASCII B

NULL	Null Idle	CR	Carriage return
SOM	Start of message	SO	Shift out
EOA	End of address	SI	Shift in
EOM	End of message	DC <sub>0</sub>	Device control ① Reserved for data link escape
EOT	End of transmission	DC <sub>1</sub> -DC <sub>3</sub>	Device control
WRU	"Who are you?"	ERR	Error
RU	"Are you ...?"	SYNC	Synchronous idle
BELL	Audible signal	LEM	Logical end of media
FE	Format effector	SO <sub>0</sub> -SO <sub>7</sub>	Separator (information) Word separator (blank, normally non-printing)
HT	Horizontal tabulation	ACK	Acknowledge
SK	Skip (punched card)	②	Unassigned control
LF	Line feed	ESC	Escape
V/TAB	Vertical tabulation	DEL	Delete Idle
FF	Form feed		

Fuente: LEAL CHAPA, César Augusto. *Fundamentos de diseño digital*. p 97.

Tabla XXVII. Ejemplo de paridad

Entrada	Paridad par	Paridad impar
ASCII A=1000001	01000001	11000001
ASCII T=1010100	11010100	01010100

Fuente: elaboración propia, empleando Excel.

### 2.1.8. Códigos para detectar errores

Cuando se desea detectar errores en la comunicación y en el procesamiento de datos, a veces se añade un octavo bit al carácter ASCII para indicar su paridad.

El bit de paridad es un bit adicional que se incluye en un mensaje de modo que el número total de unos sea par o impar. Considere el caso de la tabla XXVII. En ambos casos, se inserta un bit extra en la posición extrema izquierda del código para producir un número par de unos en el carácter, si se está usando paridad par, o un número impar de unos en el carácter, si se utiliza paridad impar. En general, se adopta solo una de las dos paridades, siendo más común la paridad par.

Para la paridad par, la detección de errores se efectúa generando, para cada carácter, un bit de paridad par en el extremo transmisor. Los caracteres de ocho bits que incluyen bits de paridad se transmiten a su destino. En el extremo receptor se verifica la paridad de cada carácter recibido. Si un carácter no tiene un número par de unos, quiere decir que por lo menos un bit cambio de valor durante la transmisión. Este método detecta un error, tres errores o cualquier combinación impar de errores en cada carácter transmitido. Si hay un número par de errores no se detectan. Podrían requerirse códigos adicionales de detección de errores para detectar una combinación par de errores.

### **2.1.9. Lógica binaria**

La lógica binaria trata de las operaciones lógicas con variables que adoptan sólo 2 valores posibles (cero y uno), tomando como referencia que el valor cero corresponde a un valor que se puede denominar: no, falso, bajo, abierto, y el valor uno como: si, verdadero, alto, cerrado, en dependencia del sistema a que se aplique. Por tanto, los factores que intervienen en una operación lógica solo pueden tomar 2 valores, verdadero o falso, y el resultado de dicha operación lógica solo puede tener un valor verdadero o falso.

El paralelismo entre las operaciones binarias de la lógica y las operaciones binarias de los sistemas digitales ha convertido a las puertas lógicas en los elementos básicos que componen los circuitos digitales. La cantidad de combinaciones que se pueden establecer en una operación de lógica binaria depende de la cantidad de variables que intervengan en ella. Como las variables solo pueden tomar 2 valores posibles, las combinaciones posibles serían 2 elevada a la cantidad de variables. Por ejemplo, para 4 variables serían 16 combinaciones ( $2^4 = 16$ ). El resultado de dicha operación va a depender del tipo de función que se aplique. La lógica binaria consiste en variables binarias y operaciones lógicas. Las variables se designan con letras del alfabeto, como A, B, C, x, y, z, etc., y cada variable tiene dos y solo dos posibles valores: 1 y 0. Las operaciones básicas son:

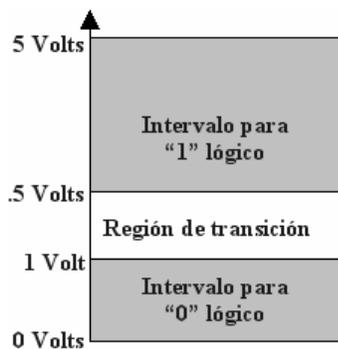
- AND: esta operación se representa con un punto u omitiendo el operador. Por ejemplo  $x \cdot y = z$  o  $xy = z$  y se lee “x AND y es igual a z”.
- OR: esta operación se representa con un signo más. Por ejemplo,  $x + y = z$  y se lee “x OR y es igual a z”.
- NOT: esta operación se representa con un apóstrofo o con una testa. Por ejemplo,  $x' = z$  o  $\bar{x} = z$  y se lee “NOT de x es igual a z”.

### 2.1.10. Compuertas lógicas

Las compuertas lógicas son circuitos electrónicos que operan con una o más señales de entrada para producir una señal de salida. En los sistemas digitales, las señales eléctricas podrían ser voltajes o corrientes que existen con uno de dos valores reconocibles. Los circuitos operados por voltaje responden a dos niveles de voltaje distintos que representan una variable binaria cuyo valor es 1 lógico o 0 lógico. En la práctica, cada nivel de voltaje tiene un intervalo aceptable. Como se ve en la figura 4, hay un rango de voltajes en donde los

dispositivos electrónicos detectan un 1 lógico y 0 lógico y otro rango en el que no detectarán nada o detectarán cualquiera de las dos opciones. En las siguientes subsecciones se usará a “S” como salida y a “a” y “b” como entradas para las tablas de verdad.

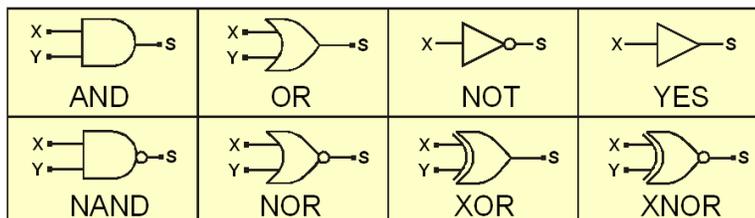
Figura 4. **Intervalo de voltaje de compuertas lógicas**



Fuente: Compuertas lógicas. <https://goo.gl/wtYypa>. Consulta: 25 de diciembre de 2016.

Se puede hacer una lista de las compuertas lógicas que actualmente existen, estas son: YES, NOT, AND, NAND, OR, NOR, XOR, XNOR.

Figura 5. **Simbología de compuestas lógicas**



Fuente: MORENO, Erwin. *Electrónica digital*. <https://goo.gl/RhWwcl>. Consulta: 25 de diciembre de 2016.

### 2.1.10.1. Compuerta lógica YES

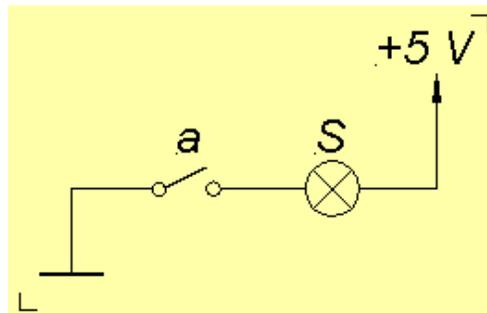
También conocida como *Buffer* o *EQUAL*, es aquella donde el resultado S es igual a la entrada a. Esta compuerta solo cuenta con una entrada.

Tabla XXVIII. **Tabla de verdad de compuerta lógica YES**

<b>a</b>	<b>S</b>
<b>1</b>	<b>1</b>
<b>0</b>	<b>0</b>

Fuente: *Lógica binaria electrónica digital*. <https://goo.gl/xkdbsh>. Consulta: 25 de diciembre de 2016.

Figura 6. **Circuito equivalente de compuerta lógica YES**



Fuente: *Lógica binaria electrónica digital*. <https://goo.gl/xkdbsh>. Consulta: 25 de diciembre de 2016.

### 2.1.10.2. Compuerta lógica NOT

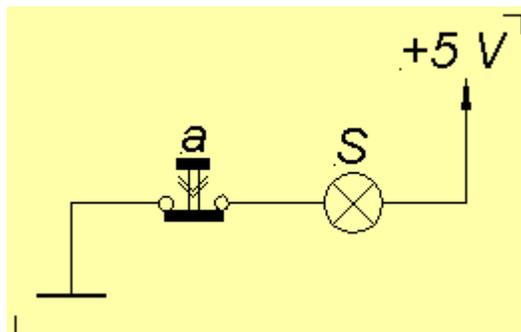
Es aquella donde el resultado  $S$  es el opuesto a la entrada  $a$ . Esta compuerta solo cuenta con una entrada.

Tabla XXIX. Tabla de verdad de compuerta lógica NOT

$a$	$S$
1	0
0	1

Fuente: *Lógica binaria electrónica digital*. <https://goo.gl/xkdbsh>. Consulta: 25 de diciembre de 2016.

Figura 7. Circuito equivalente de compuerta lógica NOT



Fuente: *Lógica binaria electrónica digital*. <https://goo.gl/xkdbsh>. Consulta: 25 de diciembre de 2016.

### 2.1.10.3. Compuerta lógica AND

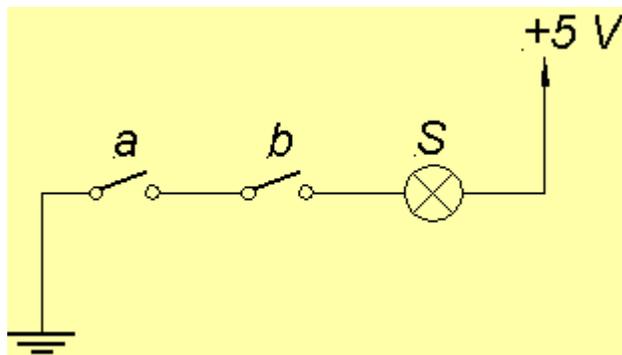
La compuerta lógica AND significa que  $S=1$  si y solo si  $a=1$  y  $b=1$ ; de lo contrario  $S=0$ .

Tabla XXX. Tabla de verdad de compuerta lógica AND

$a$	$b$	$S$
0	0	0
0	1	0
1	0	0
1	1	1

Fuente: *Lógica binaria electrónica digital*. <https://goo.gl/xkdbsh>. Consulta: 25 de diciembre de 2016.

Figura 8. Circuito equivalente de compuerta lógica AND



Fuente: *Lógica binaria electrónica digital*. <https://goo.gl/xkdbsh>. Consulta: 25 de diciembre de 2016.

#### 2.1.10.4. Compuerta lógica NAND

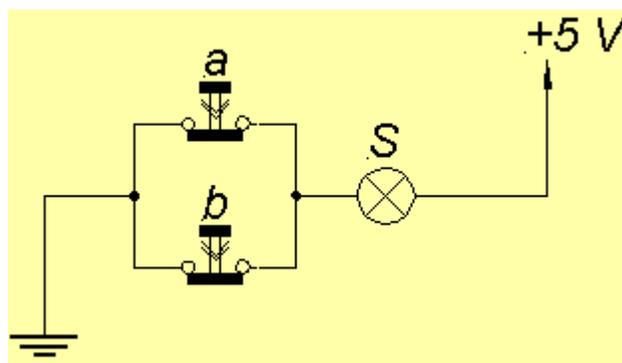
En la salida de una compuerta AND se coloca una compuerta NOT y se obtiene el resultado de esta compuerta.

Tabla XXXI. Tabla de verdad de compuerta lógica NAND

$a$	$b$	$S$
0	0	1
0	1	1
1	0	1
1	1	0

Fuente: *Lógica binaria electrónica digital*. <https://goo.gl/xkdbsh>. Consulta: 25 de diciembre de 2016.

Figura 9. Circuito equivalente de compuerta lógica NAND



Fuente: *Lógica binaria electrónica digital*. <https://goo.gl/xkdbsh>. Consulta: 25 de diciembre de 2016.

### 2.1.10.5. Compuerta lógica OR

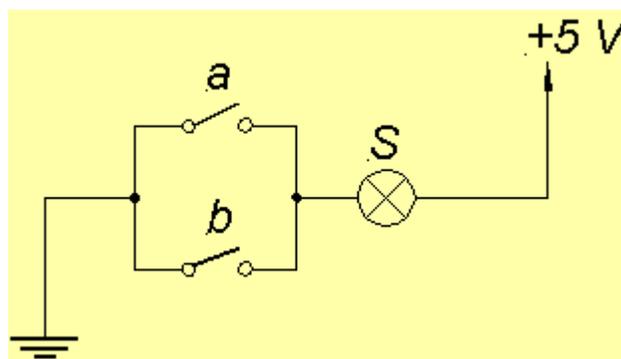
La compuerta lógica OR significa que  $S=0$  si y solo si  $a=0$  y  $b=0$ ; de lo contrario  $S=1$ .

Tabla XXXII. Tabla de verdad de compuerta lógica OR

$a$	$b$	$S$
0	0	0
0	1	1
1	0	1
1	1	1

Fuente: *Lógica binaria electrónica digital*. <https://goo.gl/xkdbsH>. Consulta: 25 de diciembre de 2016.

Figura 10. Circuito equivalente de compuerta lógica OR



Fuente: *Lógica binaria electrónica digital*. <https://goo.gl/xkdbsH>. Consulta: 25 de diciembre de 2016.

### 2.1.10.6. Compuerta lógica NOR

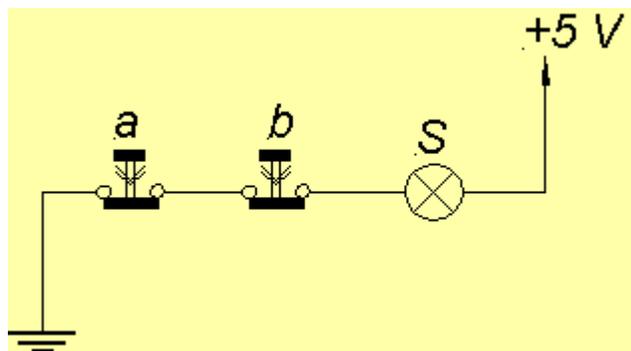
En la salida de una compuerta OR se coloca una compuerta NOT.

Tabla XXXIII. Tabla de verdad de compuerta lógica NOR

<i>a</i>	<i>b</i>	<i>S</i>
0	0	1
0	1	0
1	0	0
1	1	0

Fuente: *Lógica binaria electrónica digital*. <https://goo.gl/xkdbsh>. Consulta: 25 de diciembre de 2016.

Figura 11. Circuito equivalente de compuerta lógica NOR



Fuente: *Lógica binaria electrónica digital*. <https://goo.gl/xkdbsh>. Consulta: 25 de diciembre de 2016.

### 2.1.10.7. Compuerta lógica XOR

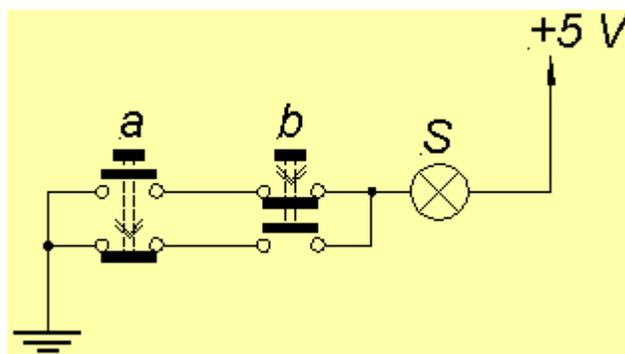
La compuerta lógica XOR también se conoce como OREX u OR exclusiva. XOR nos indica que  $S=1$  si  $a=1$  o  $b=1$  pero  $S=0$  si  $a=1$  y  $b=1$ .

Tabla XXXIV. Tabla de verdad de compuerta lógica XOR

$a$	$b$	$S$
0	0	0
0	1	1
1	0	1
1	1	0

Fuente: *Lógica binaria electrónica digital*. <https://goo.gl/xkdbsh>. Consulta: 25 de diciembre de 2016.

Figura 12. Circuito equivalente de compuerta lógica XOR



Fuente: *Lógica binaria electrónica digital*. <https://goo.gl/xkdbsh>. Consulta: 25 de diciembre de 2016.

### 2.1.10.8. Compuerta lógica XNOR

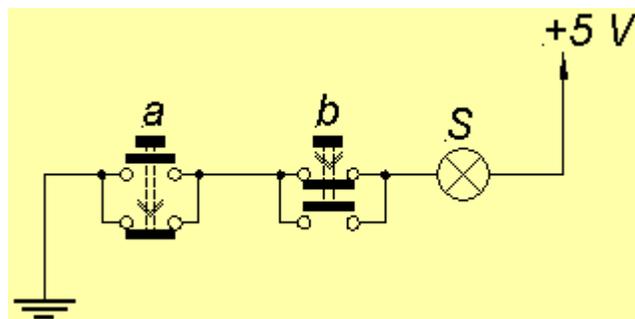
En la salida de una compuerta XNOR se coloca una compuerta NOT.

Tabla XXXV. **Tabla de verdad de compuerta lógica XNOR**

<i>a</i>	<i>b</i>	<i>S</i>
0	0	1
0	1	0
1	0	0
1	1	1

Fuente: *Lógica binaria electrónica digital*. <https://goo.gl/xkdbsh>. Consulta: 25 de diciembre de 2016.

Figura 13. **Circuito equivalente de compuerta lógica XNOR**



Fuente: *Lógica binaria electrónica digital*. <https://goo.gl/xkdbsh>. Consulta: 25 de diciembre de 2016.

## **2.2. Álgebra booleana**

Se denomina así en honor a George Boole, matemático inglés autodidacta, quien fue el primero en definirla como parte de un sistema lógico, inicialmente en un pequeño folleto, *The Mathematical Analysis of Logic*, publicado en 1847.

El álgebra de Boole fue un intento de utilizar las técnicas algebraicas para tratar expresiones de la lógica proposicional. Más tarde, fue conocido como un libro más importante: *An Investigation of the Laws of Thought on Which are Founded the Mathematical Theories of Logic and Probabilities*, publicado en 1854.

### **2.2.1. Reseña histórica**

Una vez que los circuitos implementados por medio de relevadores electromagnéticos adquirieron popularidad, fue necesario su estudio y sistematización. Eran redes formadas por interruptores y contactares de relevadores que por medio de combinaciones de circuitos abiertos y cerrados que desarrollaban funciones específicas. En la actualidad, el álgebra de Boole se aplica de forma generalizada en el ámbito del diseño electrónico.

### **2.2.2. Definición**

Es una estructura algebraica definida por un conjunto de elementos, junto con dos operadores binarios,  $+$  y  $\bullet$ , a condición de que se satisfagan los postulados siguientes (de Huntington):

- Cerradura
  - Cerradura respecto del operador +
  - Cerradura respecto del operador ●
- Elemento identidad
  - Un elemento de identidad respecto de +, designado por 0:  $x + 0 = 0 + x = x$
  - Un elemento de identidad respecto de ●, designado por 1:  $x \cdot 1 = 1 \cdot x = x$
- Conmutativa
  - Conmutativa respecto de +:  $x + y = y + x$
  - Conmutativa respecto de ●:  $x \cdot y = y \cdot x$
- Distributiva
  - ● distributiva sobre +:  $x \cdot (y + z) = (x \cdot y) + (x \cdot z)$
  - + distributiva sobre ●:  $x + (y \cdot z) = (x + y) \cdot (x + z)$
- Para cada elemento  $x$ , existe un elemento  $x'$  (llamado complemento de  $x$ ) tal que:
  - $x + x' = 1$
  - $x \cdot x' = 0$
- Existen al menos dos elementos tales que  $x \neq y$
- El álgebra de Boole es asociativa y se puede derivar, para ambos operadores, de los otros postulados.

### 2.2.3. Álgebra booleana de dos valores

El álgebra booleana de dos valores se define sobre un conjunto de dos elementos,  $B = \{1,0\}$ , con las reglas para los dos operadores binarios + y ●, que se muestran en las siguientes tablas. Como se verá estas se comparan con las compuertas AND, OR y NOT. También se puede ver que se cumple el

postulado de cerradura. Ahora se demostrará que los postulados anteriores se cumplen para  $B = \{1,0\}$ .

- La cerradura es obvia por la figura 14, pues el resultado de todas las operaciones son 1 o 0 (que son números naturales).
- En la figura 14 se ve que  $0 + 0 = 0$  y  $1 \cdot 1 = 1$  (elementos de identidad).
- Las leyes conmutativas son obvias por la simetría de las tablas de los operadores binarios.
- Para la ley distributiva se construyen las tablas que muestra la figura 14 y figura 15.
- El complemento se verifica en la tabla XXXVI.
- El sexto postulado satisface porque el álgebra booleana de dos valores tiene elementos, 1 y 0, y  $1 \neq 0$

Figura 14. **Ley distributiva algebra booleana sobre +**

$x$	$y$	$z$	$y + z$	$x \cdot (y + z)$	$x \cdot y$	$x \cdot z$	$(x \cdot y) + (x \cdot z)$
0	0	0	0	0	0	0	0
0	0	1	1	0	0	0	0
0	1	0	1	0	0	0	0
0	1	1	1	0	0	0	0
1	0	0	0	0	0	0	0
1	0	1	1	1	0	1	1
1	1	0	1	1	1	0	1
1	1	1	1	1	1	1	1

Fuente: MANO, Morris. *Diseño digital*. p 37.

Tabla XXXVI. **Complemento**

x	x'	x+x'	x*x'
0	1	1	0
1	0	1	0

Fuente: elaboración propia, empleando Excel.

#### 2.2.4. Principio de dualidad

Establece que toda expresión algebraica que puede deducirse de los postulados del algebra booleana seguirá siendo válida si se intercambian los operadores y los elementos de identidad. En el álgebra booleana de 2 valores, los elementos de identidad y los elementos son: 1 y 0. El principio de dualidad tiene muchas aplicaciones. Si se desea el dual de una expresión algebraica, simplemente se intercambian los operadores OR y AND y se sustituyen los unos por ceros y los ceros por los unos.

Figura 15. **Ley distributiva algebra booleana sobre •**

x	y	z	yz	x+yz	x+y	x+z	(x+y)(x+z)
0	0	0	0	0	0	0	0
0	0	1	0	0	0	1	0
0	1	0	0	0	1	0	0
0	1	1	1	1	1	1	1
1	0	0	0	1	1	1	1
1	0	1	0	1	1	1	1
1	1	0	0	1	1	1	1
1	1	1	1	1	1	1	1

Fuente: elaboración propia, empleando Excel.

## 2.2.5. Teoremas básicos del álgebra booleana

En la figura 16 se puede apreciar que se tiene una cantidad de seis teoremas con cuatro postulados. Respecto de los postulados, no será necesaria su demostración algebraica dado a que son expresiones muy sencillas, pero se demostrará posteriormente los teoremas.

Figura 16. Postulados y teoremas del álgebra booleana

Postulado 2	a)	$x + 0 = x$	b)	$x \cdot 1 = x$
Postulado 5	a)	$x + x' = 1$	b)	$x \cdot x' = 0$
Teorema 1	a)	$x + x = x$	b)	$x \cdot x = x$
Teorema 2	a)	$x + 1 = 1$	b)	$x \cdot 0 = 0$
Teorema 3, involución		$(x')' = x$		
Postulado 3, conmutatividad	a)	$x + y = y + x$	b)	$xy = yx$
Teorema 4, asociatividad	a)	$x + (y + z) = (x + y) + z$	b)	$x(yz) = (xy)z$
Postulado 4, distributividad	a)	$x(y + z) = xy + xz$	b)	$x + yz = (x + y)(x + z)$
Teorema 5, DeMorgan	a)	$(x + y)' = x'y'$	b)	$(xy)' = x' + y'$
Teorema 6, absorción	a)	$x + xy = x$	b)	$x(x + y) = x$

Fuente: MANO, Morris. *Diseño digital*. p 38.

### 2.2.5.1. Teorema 1

Se procede a demostrar el teorema 1 a).

$$x + x = x$$

$$x + x = (x + x) * 1 \quad 2b)x * 1 = x$$

$$x + x = (x + x)(x + x') \quad 5a)x + x' = 1$$

$$x + x = x + xx' \quad 4b)x + yz = (x + y)(x + z)$$

$$x + x = x + 0 \quad 5b)x * x' = 0$$

$$x + x = x \quad 2a)x + 0 = x$$

Se procede a demostrar el teorema 1 b).

$$x * x = x$$

$$x * x = xx + 0 \quad 2a) x + 0 = x$$

$$x * x = xx + xx' \quad 5b) x * x' = 0$$

$$x * x = x(x + x') \quad 4a) x(y + z) = xy + xz$$

$$x * x = x * 1 \quad 5a) x + x' = 1$$

$$x * x = x \quad 2b) x * 1 = x$$

### 2.2.5.2. Teorema 2

Se procede a demostrar el teorema 2 a).

$$x + 1 = 1$$

$$x + 1 = 1 * (x + 1) \quad 2b) x * 1 = x$$

$$x + 1 = (x + x')(x + 1) \quad 5a) x + x' = 1$$

$$x + 1 = x + x' * 1 \quad 4b) x + yz = (x + y)(x + z)$$

$$x + 1 = x + x' \quad 2b) x * 1 = x$$

$$x + 1 = 1 \quad 5a) x + x' = 1$$

Ahora se procede a demostrar el teorema 2 b). Para ello se debe recordar el principio de dualidad.

$$x * 0 = 0$$

### 2.2.5.3. Teorema 3, involución

Del postulado 5, se tiene  $x + x' = 1$  y  $x * x' = 0$ , lo que define al complemento de  $x$ . El complemento de  $x'$  es  $x$  y también es  $(x')'$ . Por tanto, dado que el complemento es único, se tiene que  $(x')' = x$ .

### 2.2.5.4. Teorema 4, asociatividad

En la tabla XXXVII se puede apreciar el teorema 4 a) y el teorema 4 b).

Tabla XXXVII. Asociatividad

x	y	z	$x+(y+z)$	$(x+y)+z$	$x(yz)$	$(xy)z$
0	0	0	0	0	0	0
0	0	1	1	1	0	0
0	1	0	1	1	0	0
0	1	1	1	1	0	0
1	0	0	1	1	0	0
1	0	1	1	1	0	0
1	1	0	1	1	0	0
1	1	1	1	1	1	1

Fuente: elaboración propia, empleando Excel.

### 2.2.5.5. Teorema 5, DeMorgan

El teorema de DeMorgan puede ser visto en la figura 17 y en la tabla XXXVIII.

Figura 17. **Teorema 5 a)**

$x$	$y$	$x + y$	$(x + y)'$	$x'$	$y'$	$x'y'$
0	0	0	1	1	1	1
0	1	1	0	1	0	0
1	0	1	0	0	1	0
1	1	1	0	0	0	0

Fuente: MANO, Morris. *Diseño digital*. p 40.

Tabla XXXVIII. **Teorema 5 b)**

$x$	$y$	$xy$	$(xy)'$	$x'$	$y'$	$x'+y'$
0	0	0	1	1	1	1
0	1	0	1	1	0	1
1	0	0	1	0	1	1
1	1	1	0	0	0	0

Fuente: elaboración propia, empleando Excel.

### 2.2.5.6. **Teorema 6, absorción**

Se procede a demostrar el teorema 6 a).

$$x + xy = x$$

$$x + xy = x * 1 + xy$$

$$x + xy = x(1 + y)$$

$$x + xy = x(y + 1)$$

$$2b) x * 1 = x$$

$$4a) x(y + z) = xy + xz$$

$$3a) x + y = y + x$$

$$x + xy = x * 1$$

$$2a) x + 1 = 1$$

$$x + xy = x$$

$$2b) x * 1 = x$$

Ahora se demuestra el teorema 6 b). Para ello hay que recordar el principio de dualidad.

$$x(x + y) = x$$

### 2.2.6. Funciones booleanas

El álgebra booleana se ocupa de variables binarias y operaciones lógicas. Una función booleana descrita por una expresión algebraica consta de variables binarias, las constantes uno y cero, y los símbolos lógicos de operaciones. Para un valor de las variables binarias, la función puede ser igual a uno o bien a cero. Un ejemplo de una función booleana es:

$$F = x + zy'$$

Tabla XXXIX. **Tabla de verdad de la función F1**

$x$	$y$	$z$	$F_1$
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

Fuente: MANO, Morris. *Diseño digital*. p 41.

Estas ayudarán a simplificar las tablas de verdad y a tener un mejor orden en el diseño de circuitos digitales. Las tablas de verdad es un modo más visual de representar expresiones booleanas. Estas llevan un conteo de modo que la función pueda tomar en cuenta cada uno de los casos que el número de variables pueda ofrecer. Si existen tres variables, la tabla de verdad tendrá  $2^3 = 8$  casos (renglones). El orden de conteo es por medio de la codificación natural y el primer caso es el número cero.

Cuando se compara la función con la tabla de verdad, se ve que la función solo define cuando  $F_1$  es igual a uno, dado a que estos valores son los de interés. Cabe la posibilidad que los valores de interés puedan ser ceros y dependerá del diseño que se esté implementando, pero por lo general y si no se dice lo contrario el valor de interés siempre será uno. Una función puede ser representada de una manera mucho más práctica y reducida que la tabla de verdad. Consiste en una sumatoria donde se indiquen las posiciones en donde será igual a uno la función. Para explicar esto se expresará de esta forma la función  $F_1$ :

$$F_1 = \sum (1,4,5,6,7)$$

La expresión representa que será uno en las posiciones uno, cuatro, cinco, seis y siete de la tabla de verdad. Hay que recordar que la posición inicial de la tabla de verdad es cero y la final  $2^n - 1$  donde  $n$  es el número de variables que tiene la función  $F$ . Otro formato que se puede encontrar es el siguiente:

$$F_1(a, b, c) = \sum (1,4,5,6,7)$$

Este significa que es 1 en las posiciones 1, 4, 5, 6, 7 y F1 está en función de las variables a, b y c.

Ejemplo 36: encuentre la función de la tabla XL.

Dado a que una función se encuentra en forma algebraica puede ser expresada de varias maneras. La expresión específica empleada para designar la función también determinará la interconexión de compuertas en el diagrama de circuitos lógicos. Manipulando una expresión booleana según las reglas del álgebra booleana, a veces es posible obtener una expresión más simple para la misma función y así reducir el número de compuertas del circuito y el número de entradas de las compuertas. Esto se realiza con los postulados y teoremas vistos con anterioridad.

Tabla XL. **Tabla de ejemplo 36**

x	y	z	F
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0

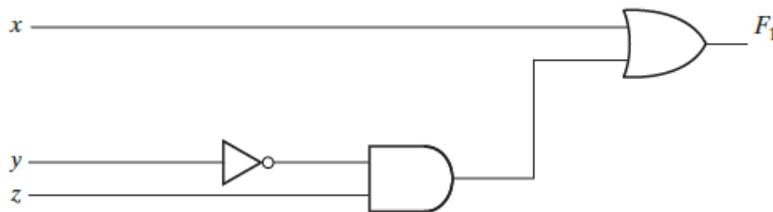
Fuente: elaboración propia, empleando Excel.

$$F = x'y'z' + x'yz + xyz'$$

$$F = x + y'z$$

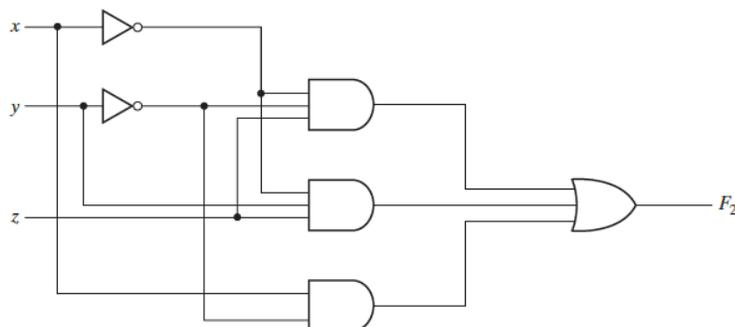
Una función booleana se puede transformar de una expresión algebraica a un diagrama de circuitos hechos con compuertas lógicas. Por ejemplo, se realizará el diagrama de la siguiente función F.

Figura 18. **Representación de función F en compuertas lógicas**

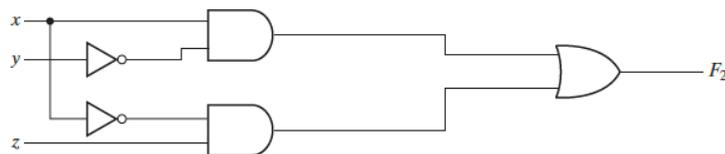


Fuente: MANO, Morris. *Diseño digital*. p 41.

Figura 19. **Circuitos de ejemplo 36**



a)  $F_2 = x'y'z + x'yz + xy'$



b)  $F_2 = xy' + x'z$

Fuente: MANO, Morris. *Diseño digital*. p 42.

### 2.2.6.1. Simplificación de funciones booleanas con álgebra de Boole

La simplificación de funciones booleanas por medio del álgebra de Boole es un proceso que consiste en hacer más sencilla una expresión matemática para que el circuito que corresponde a esa expresión, sea más compacto. Y esto utilizando los postulados y teoremas vistos anteriormente.

Ejemplo 37: simplificar la siguiente expresión con la ayuda de los teoremas y postulados del álgebra de Boole.  $F_2 = x'y'z + x'yz + xy'$ .

$$F_2 = x'y'z + x'yz + xy'$$

$$F_2 = z(x'y' + x'y) + xy'$$

$$F_2 = z(x'(y' + y)) + xy'$$

$$F_2 = zx' + xy'$$

Como se ve en la figura anterior, así sería la diferencia entre la función inicial y la función simplificada.

Ejemplo 38: simplificar la siguiente expresión:  $x(x' + y)$ .

$$x(x' + y)$$

$$xx' + yx = 0 + xy = xy$$

Ejemplo 39: simplificar la siguiente expresión:  $x + x'y$ .

$$x + x'y$$

$$(x + x')(x + y) = 1(x + y) = x + y$$

Para este ejemplo se utilizó la ley distributiva.

Ejemplo 40: simplificar la siguiente expresión:  $(x + y)(x + y')$ . Para este ejemplo se utilizó la ley distributiva.

$$(x + y)(x + y')$$

$$x + yy' = x + 0 = x$$

Ejemplo 41: simplificar la siguiente expresión:  $xy + x'z + yz$ .

$$xy + x'z + yz$$

$$xy + x'z + yz(x + x') = xy + x'z + yzx + yzx'$$

$$xy(1 + z) + x'z(1 + y) = xy(1) + x'z(1)$$

$$xy + x'z$$

Ejemplo 42: simplificar la siguiente expresión:  $(x + y)(x' + z)(y + z)$ .

$$(x + y)(x' + z)(y + z)$$

$$(x + y)(x'y + xz + zy + zz)$$

$$(x + y)(x'y + xz + zy + z)$$

$$(x + y)(x'y + z(1 + x + y))$$

$$(x + y)(x'y + z)$$

$$xz + xx'y + yz + x'yy = xz + yz + x'y$$

$$xz + yz(x + x') + x'y = xz + x'y + yzx + yzx'$$

$$xz(1 + y) + x'y(1 + y) = xz + x'y$$

### 2.2.6.2. Complemento de una función

El complemento de una función  $F$  es  $F'$  y se obtiene intercambiando ceros por unos y unos por ceros en el valor de  $F$ . El complemento de una función puede deducirse algebraicamente empleando el teorema de DeMorgan. El teorema de DeMorgan de 2 variables fue visto en tablas anteriores, pero en los siguientes ejemplos se podrá verificar que se puede implementar en funciones de más de una variable. En la siguiente figura se puede apreciar cómo sería la aplicación del teorema de DeMorgan para una expresión booleana de 3 variables.

Figura 20. **DeMorgan para una función de 3 variables**

$$\begin{aligned}(A + B + C)' &= (A + x)' \\ &= A'x' \\ &= A'(B + C)' \\ &= A'(B'C)' \\ &= A'B'C'\end{aligned}$$

Fuente: MANO, Morris. *Diseño digital*. p 43.

Siendo  $B + C = x$

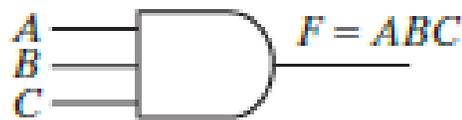
Ejemplo 43: encontrar el complemento de  $F = x'yz' + x'y'z$ .

$$\begin{aligned}F' &= (x'yz' + x'y'z)' \\ &= (x'yz')'(x'y'z)' \\ &= (x + y' + z)(x + y + z')\end{aligned}$$

Ejemplo 44: encontrar el complemento de  $F = x(y'z' + yz)$ .

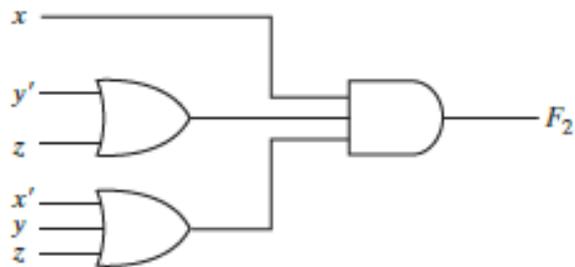
$$\begin{aligned}
 F' &= (x(y'z' + yz))' \\
 &= x' + (y'z' + yz)' \\
 &= x' + (y'z')'(yz)' \\
 &= x' + (y + z)(y' + z')
 \end{aligned}$$

Figura 21. **Ejemplo de nivel 1**



Fuente: MANO, Morris. *Diseño digital*. p 56.

Figura 22. **Ejemplo de nivel 2**



Fuente: MANO, Morris. *Diseño digital*. p 50.

### **2.2.7. Niveles en compuertas lógicas**

Cuando se habla de niveles en compuertas lógicas, hace referencia a la complejidad con que el circuito esté diseñado. Para entender esto se llamará “salto” al proceso que consiste en pasar a través de una compuerta lógica. El nivel inicial es uno y se refiere a que todas las variables booleanas de entrada de la función tienen que realizar solo un salto para llegar a la salida de la función.

Cuando se posee un circuito con muchos niveles de compuertas lógicas surge la problemática que el retardo será mayor comparado con uno que tiene pocos niveles. La figura 21 muestra un ejemplo de nivel 1, dado a que las entradas pasan por solo una compuerta lógica para llegar a la salida. Por otro lado, la figura 22 muestra un nivel 2 porque por lo menos una de las entradas tiene que pasar por dos compuertas lógicas para llegar a la salida.

### **2.2.8. Formas canónicas**

Para la representación de funciones booleanas hay distintas formas para su representación, a continuación, se verán las formas de minitérminos y maxitérminos.

#### **2.2.8.1. Minitérminos y maxitérminos**

Una variable binaria podría aparecer en su forma normal ( $x$ ) o en su forma complementada ( $x'$ ). Considere ahora dos variables binarias  $x$  y  $y$  que se combinan con una operación AND. Puesto que cada variable podría aparecer en cualquiera de sus formas, hay cuatro combinaciones posibles:  $x'y'$ ,  $x'y$ ,  $xy'$  y  $xy$ .

Cada uno de estos cuatro términos AND es un minitérmino o producto estándar. De manera similar, se puede combinar  $n$  variables para formar  $2^n$  minitérminos.

Tabla XLI. **Maxitérminos y minitérminos para tres variables binarias**

$x$	$y$	$z$	Minitérminos		Maxitérminos	
			Términos	Designación	Términos	Designación
0	0	0	$x'y'z'$	$m_0$	$x + y + z$	$M_0$
0	0	1	$x'y'z$	$m_1$	$x + y + z'$	$M_1$
0	1	0	$x'yz'$	$m_2$	$x + y' + z$	$M_2$
0	1	1	$x'yz$	$m_3$	$x + y' + z'$	$M_3$
1	0	0	$xy'z'$	$m_4$	$x' + y + z$	$M_4$
1	0	1	$xy'z$	$m_5$	$x' + y + z'$	$M_5$
1	1	0	$xyz'$	$m_6$	$x' + y' + z$	$M_6$
1	1	1	$xyz$	$m_7$	$x' + y' + z'$	$M_7$

Fuente: MANO, Morris. *Diseño digital*. p 45.

Asimismo,  $n$  variables que forman un término OR, donde cada variable puede tener apóstrofo o no, dan pie a  $2^n$  posibles combinaciones, llamadas maxitérminos o sumas estándar. Un dato que hay que tomar en cuenta es que cada maxitérminos es el complemento de su minitérmino correspondiente y viceversa. Como se ve en la tabla XLI para cada minitérmino, si la variable tiene un apóstrofo, será un cero y si no lo tiene será un uno. Por otro lado, para cada maxitérmino las variables significaran uno cuando tengan un apóstrofo y cero cuando no lo tengan. Se puede deducir que cualquier función booleana puede ser representada por minitérminos o maxitérminos o una combinación de ambos.

## 2.2.9. Formas estándar

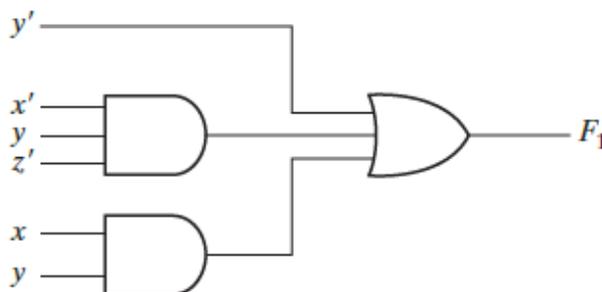
Las dos formas canónicas del álgebra booleana son formas básicas que se obtienen al leer una función de su tabla de verdad, pero casi nunca son las que tiene el número mínimo de literales, porque cada minitérmino o maxitérmino debe de contener, por definición, todas las variables, complementadas o sin complementar. En esta configuración, los términos que forman la función podrían contener, una, dos o cualquier número de literales. Hay dos tipos de forma estándar: la suma de productos y el producto de sumas.

### 2.2.9.1. Suma de productos

La suma de productos es una expresión booleana que contiene términos AND, llamados términos de productos, con una o más literales cada uno. La suma denota el OR de esos términos.

$$F1 = y' + xy + x'yz'$$

Figura 23. Circuito equivalente de la función F1



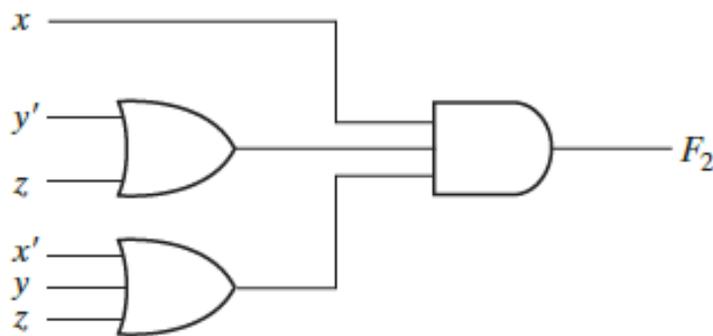
Fuente: MANO, Morris. *Diseño digital*. p 50.

### 2.2.9.2. Producto de sumas

Un producto de sumas es una expresión booleana que contiene términos OR, llamados términos de suma. Cada término puede tener cualquier cantidad de literales. El producto denota el AND de esos términos.

$$F_2 = x(y' + z)(x' + y + z)$$

Figura 24. **Circuito equivalente de función F2**

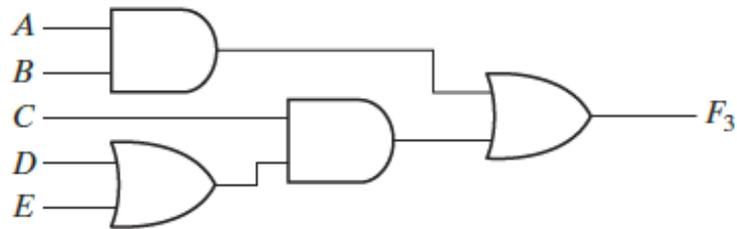


Fuente: MANO, Morris. *Diseño digital*. p 50.

Las funciones booleanas también pueden ser representadas en forma no estándar, por ejemplo:

$$F_3 = AB + C(D + E)$$

Figura 25. **Circuito equivalente de función F3**



Fuente: MANO, Morris. *Diseño digital*. p 51.

### 2.2.10. **Compuertas lógicas digitales**

Para una cierta cantidad de variables existe una cierta cantidad de funciones que se pueden generar. Con la siguiente expresión matemática se puede calcular la cantidad de funciones en base al número de variables que esta posea:

$$2^{2^n}$$

Aquí  $n$  es la cantidad de variables de la función. En base a esto se puede obtener la tabla XLII. En base a esta tabla se puede sustraer los casos para las compuertas lógicas que se han visto anteriormente. Estos se muestran en la figura 26. Puesto que todas las funciones booleanas se expresan en términos de operaciones AND, OR y NOT, es más fácil implementar una función booleana con estos tipos de compuertas.

Las compuertas AND y OR son conmutativas y asociativas, por lo que gracias a estas propiedades matemáticas es posible utilizarlas con más de 2 variables booleanas de entrada. Por parte de las compuertas NAND y NOR son

igualmente conmutativas, pero no asociativas. Para superar este problema, se define la compuerta NOR o NAND múltiple como una compuerta OR o AND complementada.

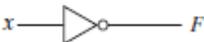
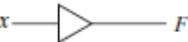
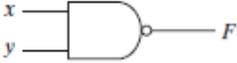
Tabla XLII. **Tablas de verdad para las 16 funciones de dos variables binarias**

$x$	$y$	$F_0$	$F_1$	$F_2$	$F_3$	$F_4$	$F_5$	$F_6$	$F_7$	$F_8$	$F_9$	$F_{10}$	$F_{11}$	$F_{12}$	$F_{13}$	$F_{14}$	$F_{15}$
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

Fuente: MANO, Morris. *Diseño digital*. p 52.

La compuerta OR exclusivo y de equivalencia es tanto conmutativa como asociativa y se pueden extender a más de dos entradas. No obstante, las compuertas OR exclusivo de varias entradas son poco comunes en hardware.

Figura 26. **Compuertas lógicas digitales**

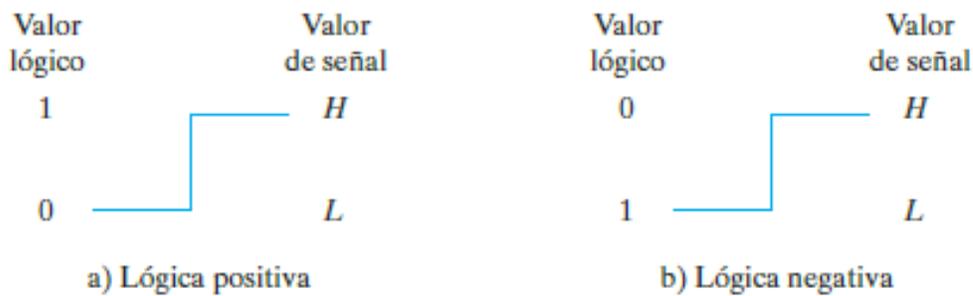
Nombre	Símbolo gráfico	Función algebraica	Tabla de verdad															
AND		$F = xy$	<table border="1"> <thead> <tr> <th>x</th> <th>y</th> <th>F</th> </tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </tbody> </table>	x	y	F	0	0	0	0	1	0	1	0	0	1	1	1
x	y	F																
0	0	0																
0	1	0																
1	0	0																
1	1	1																
OR		$F = x + y$	<table border="1"> <thead> <tr> <th>x</th> <th>y</th> <th>F</th> </tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </tbody> </table>	x	y	F	0	0	0	0	1	1	1	0	1	1	1	1
x	y	F																
0	0	0																
0	1	1																
1	0	1																
1	1	1																
Inversor		$F = x'$	<table border="1"> <thead> <tr> <th>x</th> <th>F</th> </tr> </thead> <tbody> <tr><td>0</td><td>1</td></tr> <tr><td>1</td><td>0</td></tr> </tbody> </table>	x	F	0	1	1	0									
x	F																	
0	1																	
1	0																	
Búfer		$F = x$	<table border="1"> <thead> <tr> <th>x</th> <th>F</th> </tr> </thead> <tbody> <tr><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td></tr> </tbody> </table>	x	F	0	0	1	1									
x	F																	
0	0																	
1	1																	
NAND		$F = (xy)'$	<table border="1"> <thead> <tr> <th>x</th> <th>y</th> <th>F</th> </tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td></tr> </tbody> </table>	x	y	F	0	0	1	0	1	1	1	0	1	1	1	0
x	y	F																
0	0	1																
0	1	1																
1	0	1																
1	1	0																
NOR		$F = (x + y)'$	<table border="1"> <thead> <tr> <th>x</th> <th>y</th> <th>F</th> </tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>0</td></tr> </tbody> </table>	x	y	F	0	0	1	0	1	0	1	0	0	1	1	0
x	y	F																
0	0	1																
0	1	0																
1	0	0																
1	1	0																
OR exclusivo (XOR)		$F = xy' + x'y$ $= x \oplus y$	<table border="1"> <thead> <tr> <th>x</th> <th>y</th> <th>F</th> </tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td></tr> </tbody> </table>	x	y	F	0	0	0	0	1	1	1	0	1	1	1	0
x	y	F																
0	0	0																
0	1	1																
1	0	1																
1	1	0																
NOR exclusivo o equivalencia		$F = xy + x'y'$ $= (x \oplus y)'$	<table border="1"> <thead> <tr> <th>x</th> <th>y</th> <th>F</th> </tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </tbody> </table>	x	y	F	0	0	1	0	1	0	1	0	0	1	1	1
x	y	F																
0	0	1																
0	1	0																
1	0	0																
1	1	1																

Fuente: MANO, Morris. *Diseño digital*. p 54.

### 2.2.10.1. Lógica positiva y negativa

La señal binaria en las entradas y salidas de cualquier compuerta tiene uno de dos valores exceptuando cuando se encuentra en transición. Un valor de señal representa al uno lógico y el otro al cero lógico.

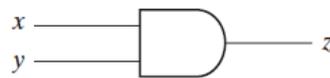
Figura 27. Lógica positiva y negativa



Fuente: MANO, Morris. *Diseño digital*. p 57.

Figura 28. Tabla de verdad y compuerta lógica en lógica positiva

$x$	$y$	$z$
0	0	0
0	1	0
1	0	0
1	1	1



Fuente: MANO, Morris. *Diseño digital*. p 58.

Puesto que se asignan dos valores de señal a dos valores lógicos, puede haber dos asignaciones distintas de nivel de señal al valor lógico.

Uno denotado como alto y otro denotado como bajo. Se puede representar a uno lógico con la señal en alto y representar al cero lógico con la señal en bajo, a esto se le conoce como lógica positiva. Por otro lado, se puede representar a uno lógico como la señal en bajo y representar al cero lógico con la señal en alto; a esto se le conoce como lógica negativa.

Figura 29. **Tabla de verdad y compuerta lógica en lógica negativa**

$x$	$y$	$z$
1	1	1
1	0	1
0	1	1
0	0	0



Fuente: MANO, Morris. *Diseño digital*. p 57.

### 2.2.11. Circuitos integrados

Un circuito integrado (CI) es un cristal semiconductor de silicio, llamado chip, que contiene los componentes electrónicos para construir compuertas digitales.

#### 2.2.11.1. Niveles de integración

Los circuitos integrados digitales suelen clasificarse según la complejidad de sus circuitos, la cual se mide por el número de compuertas lógicas incluidas en el paquete. Con base en este tipo de característica se realizan las siguientes categorías:

- SSI (*small-scale integration*): contienen varias compuertas independientes en un solo paquete, estas son de una cantidad de menor a diez compuertas lógicas. Pueden ser CI de compuertas lógicas.
- MSI (*medium-scale integration*): estos contienen una complejidad de diez a mil compuertas en un solo paquete. Pueden ser decodificadores, sumadores y multiplexores.
- LSI (*large-scale integration*): contienen miles de compuertas en un solo paquete. Se puede interpretar en el rango de mil a novecientos noventa y nueve mil novecientos noventa y nueve compuertas. Incluyen procesadores, chips de memoria y dispositivos de lógica programable.
- VLSI (*very large-scale integration*) este grupo contiene cientos de miles de compuertas en un solo paquete. Como por ejemplo las grandes matrices de memoria de los microprocesadores complejos.

#### **2.2.11.2. Familias de lógica digital**

Los circuitos integrados también pueden ser clasificados por la tecnología de circuitos utilizada en su construcción; y se le llamará familia lógica digital. Las más populares son:

- TTL, lógica transistor-transistor: es una familia lógica que ha estado en operación mucho tiempo y se le considera estándar. Estos utilizan transistores bipolares.
- ECL, lógica acoplada por emisor: resulta ventajoso en sistemas que deben operar a alta velocidad. Utiliza un amplificador diferencial hechos con BJT.
- MOS, metal-óxido-semiconductor: es apropiado para circuitos que requieren una densidad elevada de componentes. Utilizan MOS.

- CMOS metal-óxido-semiconductor complementario: es preferible en sistemas que requieren bajo consumo de energía. Utilizan CMOS.

### **2.3. Minimización de funciones booleanas**

En los capítulos anteriores se observa cómo, a partir de una tabla de verdad, se puede obtener la expresión booleanas y viceversa, que representa el comportamiento de un bloque digital. Esta expresión no siempre está en su forma más simple. Hay otras maneras con las cuales se reduce la cantidad de entradas booleanas en un circuito y, por ende, se hace un circuito más eficiente. En este capítulo se explorarán los métodos matemáticos que se tienen para minimizar al máximo la circuitería.

#### **2.3.1. Introducción**

La complejidad de las compuertas lógicas digitales que implementan una función booleana, está relacionada directamente con la complejidad de la expresión algebraica a partir de la cual se implementa la función. Aunque la representación de una función como tabla de verdad es única, hay muchas formas de expresarla algebraicamente.

#### **2.3.2. Mapa de Karnaugh**

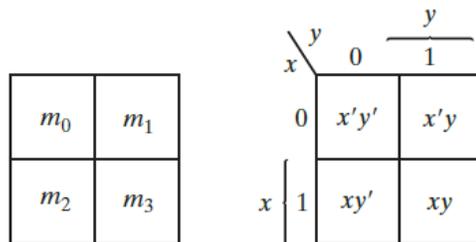
El mapa de Karnaugh es un método gráfico para la representación y minimización de funciones booleanas. Se usa para simplificar funciones de dos, tres y cuatro variables, pero puede extenderse de funciones de cinco a seis variables. Su operación se basa en la combinación de minitérminos los cuales difieren en solo una variable como:  $AB' + AB = A(B + B') = A$ .

Un mapa para una función de  $N$  variables consiste en  $2^n$  cuadros, donde cada cuadro representa a un minitérmino, además entre los minitérminos de cuadros adyacentes debe de haber un solo cambio en solo una de las variables. Un mapa para una función de 2 variables tiene  $2^2 = 4$  cuadros, para 3 variables  $2^3 = 8$ , para 4 variables  $2^4 = 16$ , para 5 variables  $2^5 = 32$  y para 6 variables  $2^6 = 64$ .

### 2.3.2.1. Mapa de dos variables

En la figura 30 se presenta un mapa de dos variables. Hay cuatro minitérminos para dos variables. Por lo tanto, el mapa consiste en cuatro cuadrados, uno para cada minitérmino.

Figura 30. Ejemplo de mapa de dos variables 1



Fuente: MANO, Morris. *Diseño digital*. p 65.

En la figura de la derecha se ha dibujado el mapa de modo que muestre la relación entre los cuadrados y las dos variables  $x$  y  $y$ . El cero y el uno que se marcan en cada fila y columna indican los valores de las variables. La variable  $x$  aparece con apóstrofo en la fila cero y sin apóstrofo en la fila uno. De forma similar,  $y$  aparece con apóstrofo en la columna cero y sin él en la columna uno.

Teniendo en cuenta que la función que se está utilizando es  $F = x'y + xy' + xy$  se puede deducir que el cuadro de la figura 31 donde será uno en los minitérminos que contiene la función. Para su resolución hay que observar, en orientación vertical u horizontal, cuando existe un solo cambio de una sola de las variables. Cuando esto ocurre los dos minitérminos se simplifican a la variable que no cambió.

Figura 31. **Ejemplo de mapa de dos variables 2**

		$y$	
		0	1
$x$	0		1
	1	1	1

Fuente: MANO, Morris. *Diseño digital*. p 65.

Ejemplo 45: para la siguiente tabla de verdad, obtener la expresión más simplificada de la función.

Tabla XLIII. **Tabla de verdad de ejemplo 45**

x	y	F
0	0	1
0	1	1
1	0	1
1	1	0

Fuente: elaboración propia, empleando Excel.

Por ser una función de 2 variables, será un mapa de Karnaugh de cuatro minitérminos y cuatro cuadros. Se ve que en la función  $F$  solo está definida para 3 minitérminos, los cuales tienen un uno en su cuadro. Se inicia por simplificar los minitérminos superiores (color amarillo), que dan como resultado  $x'$ ; esto porque la variable  $x'$  no cambia entre los dos cuadros resaltados. Luego se procede a simplificar la columna izquierda (color rojo) que da como resultado  $y'$ ; esto porque la variable  $y'$  no cambia en los cuadros resaltados.

Figura 32. **Mapa de ejemplo 45**

x\y	0	1
0	x'y'	x'y
1	xy'	xy

x\y	0	1
0	1	1
1	1	0

x\y	0	1
0	1	1
1	1	0

x\y	0	1
0	1	1
1	1	0

Fuente: elaboración propia, empleando Excel.

$$F = x' + y'$$

### 2.3.2.2. Mapa de tres variables

Para el mapa de tres variables, que se muestra en la figura 33 hay ocho minitérminos, por lo tanto, hay ocho cuadrados. Hay que advertir que los minitérminos no están acomodados en sucesión binario, sino en una sucesión similar al código Gray. La característica de esta sucesión es que solo un bit cambia de valor entre dos columnas adyacentes. Los pasos son los mismos solo que ahora se tiene que tomar en cuenta la relación de tres variables. El comportamiento de las variables en el mapa de Karnaugh está representado en la figura 33.

Figura 33. Mapa de tres variables

$m_0$	$m_1$	$m_3$	$m_2$
$m_4$	$m_5$	$m_7$	$m_6$

		$y$			
		$xz$	$00$	$01$	$11$
$x$	$0$	$x'y'z'$	$x'y'z$	$x'yz$	$x'yz'$
	$1$	$xy'z'$	$xy'z$	$xyz$	$xyz'$

Fuente: MANO, Morris. *Diseño digital*. p 66.

Ejemplo 46: para la siguiente tabla de verdad, obtener la expresión más simplificada de la función.

Tabla XLIV. **Tabla de verdad de ejemplo 46**

x	y	z	F
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

Fuente: elaboración propia, empleando Excel.

Utilizando de referencia la tabla de verdad de la función F se puede deducir la siguiente función:

$$F = x'y'z + x'yz' + x'yz + xy'z' + xy'z + xyz$$

Como se ve es el mismo procedimiento para mapas de 3 variables y mapas de 2 variables, solo que implementa una regla nueva, la cual consiste en que se intente reducir la cantidad de  $2^n$  minitérminos a la vez, por lo que se agrupan  $2^n$  cuadrados cada vez que desea reducir un grupo.

Figura 34. Mapa de ejemplo 46

x\yz	00	01	11	10
0	x'y'z'	x'y'z	x'yz	x'yz'
1	xy'z'	xy'z	xyz	xyz'

x\yz	00	01	11	10
0	0	1	1	1
1	1	1	1	0

x\yz	00	01	11	10
0	0	1	1	1
1	1	1	1	0
		Z		

x\yz	00	01	11	10
0	0	1	1	1
1	1	1	1	0
		xy'		

x\yz	00	01	11	10
0	0	1	1	1
1	1	1	1	0
		x'y		

Fuente: elaboración propia, empleando Excel.

$$F = z + xy' + x'y$$

Se inicia colocando los uno correspondientes usando como base la tabla de verdad de la función, luego se inicia agrupando los cuatro minitérminos del centro, verificando que cumple con la restricción que una variable se mantendrá constante; en este caso la z.

Luego de esto se agrupa la pareja de minterminos de la izquierda y se tiene como resultado  $xy'$ ; esto porque las variables  $x$  y  $y$  no cambiaron. Por último, se simplifica los minterminos de la derecha y se obtiene  $x'y$  por las mismas razones que el caso anterior.

Ejemplo 47: para la siguiente tabla de verdad, obtenga la expresión más simplificada.

Tabla XLV. **Tabla de verdad de ejemplo 47**

x	y	z	F
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

Fuente: elaboración propia, empleando Excel.

Utilizando de referencia la tabla de verdad de la función  $F$  se puede deducir la siguiente función:

$$F = xy'z' + xy'z + xyz' + xyz$$

$$F = xyz + x'y' + xyz' + x'yz + xy'z'$$

Como se ve, en este problema se tiene algo distinto que el caso anterior. Se inicia colocando los minterminos en los cuadros respectivos y colocando los unos necesarios usando de referencia la tabla de verdad. Ahora se ve que los

minitérminos  $xy'z'$  y  $xyz'$  están de extremo a extremo del mapa. Pero al analizar se puede apreciar que dos de las variables se mantienen constantes ( $xz$ ) y solo una es la que hace un cambio ( $y$ ), entonces se puede simplificar estos dos minitérminos. También se podría realizar si los minitérminos fueran los que se encuentran ubicados en la parte de arriba, o inclusive simplificar los cuatro minitérminos de las orillas.

Figura 35. **Mapa de ejemplo 47**

x\yz	00	01	11	10
0	$x'y'z'$	$x'y'z$	$x'yz$	$x'yz'$
1	$xy'z'$	$xy'z$	$xyz$	$xyz'$

x\yz	00	01	11	10
0			1	
1	1		1	1

x\yz	00	01	11	10
0			1	
1	1		1	1

x\yz	00	01	11	10
0			1	
1	1		1	1

Fuente: elaboración propia, empleando Excel.

$$F = yz + xz'$$

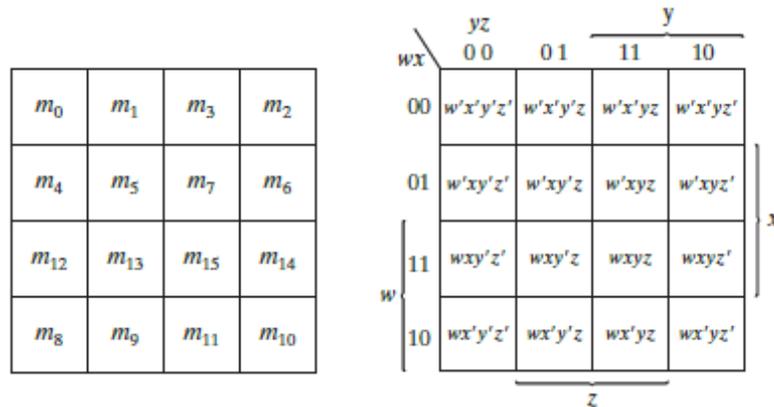
### 2.3.2.3. Mapa de cuatro variables

El mapa para las funciones booleanas de cuatro variables se ve en la figura 37. Se presentan los 16 minitérminos y los cuadrados asignados a cada uno.

Las filas y las columnas se numeran en orden según el código Gray, de modo que solo un dígito cambie de valor entre dos filas o columnas adyacentes. Se verá que el procedimiento es similar a los mapas vistos con anterioridad,

solo que en este caso habrá una mayor cantidad de variables, minitérminos y cuadros.

Figura 36. **Mapa de cuatro variables**



Fuente: MANO, Morris. *Diseño digital*. p 70.

Ejemplo 48: para la siguiente tabla de verdad, obtener la función F más simplificada.

Como se había dicho antes, es el mismo procedimiento solo que ahora con cuatro variables. Se inicia simplificando los  $2^3 = 8$  minitérminos que se tienen al lado izquierdo (ver figura 51). Como se ve la variable que se mantiene constante es la  $y'$ , por ende, este será la simplificación de ese grupo. Se prosigue con los cuatro cuadros superiores extremos que se simplifican a  $w'z'$ . Por último, los cuadros extremos centrales que se simplifican a  $xz'$ .

Se ve que para los últimos minitérminos que se simplifican se utilizaron la mayor cantidad de cuadros posibles, esto es para que la expresión final sea la más simplificada posible. El orden en que haga las simplificaciones no importa, dado a que se puede empezar desde cualquier parte del mapa, sin embargo, es

siempre recomendable iniciar desde el número mayor de minitérminos que se simplifican de una sola vez.

$$F = y' + w'z' + xz'$$

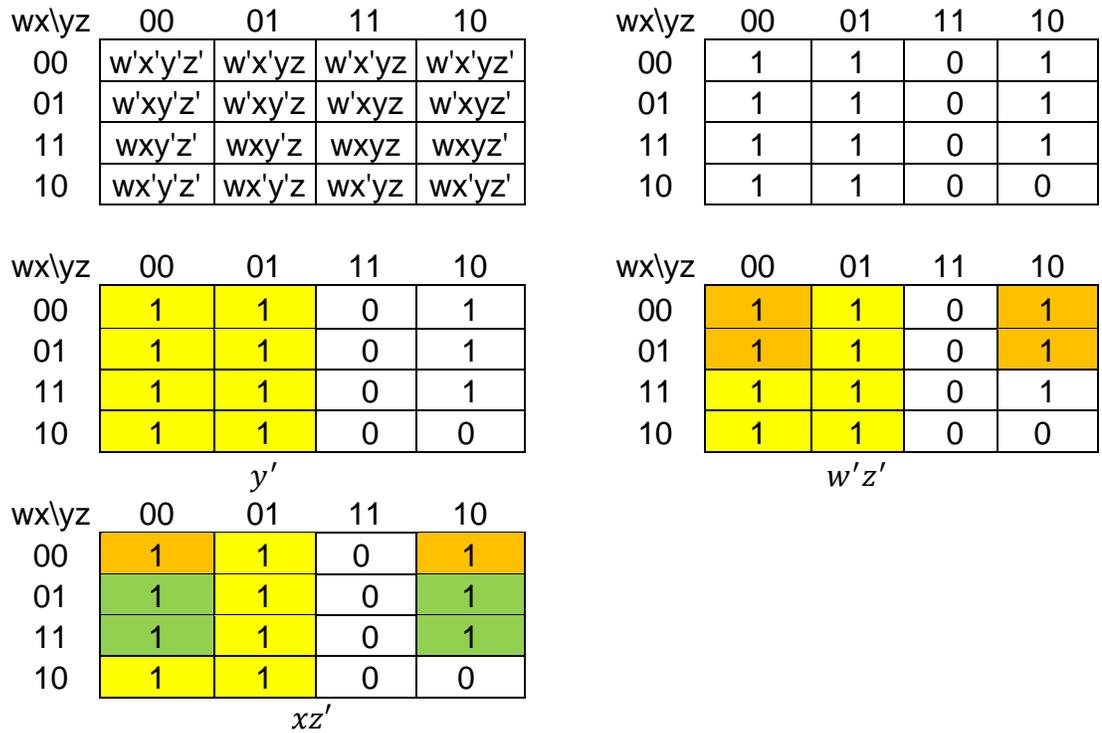
Tabla XLVI. **Tabla de verdad de ejemplo 48**

w	x	y	z	F
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	0
1	0	0	0	1
1	0	0	1	1
1	0	1	0	0
1	0	1	1	0
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

Fuente: elaboración propia, empleando Excel.

$$F = w'x'y'z' + w'x'y'z + w'x'yz + w'xy'z' + w'xy'z + w'xyz' + wx'y'z' + wx'y'z + wx'y'z' + wxy'z + wxyz'$$

Figura 37. Mapa de ejemplo 51



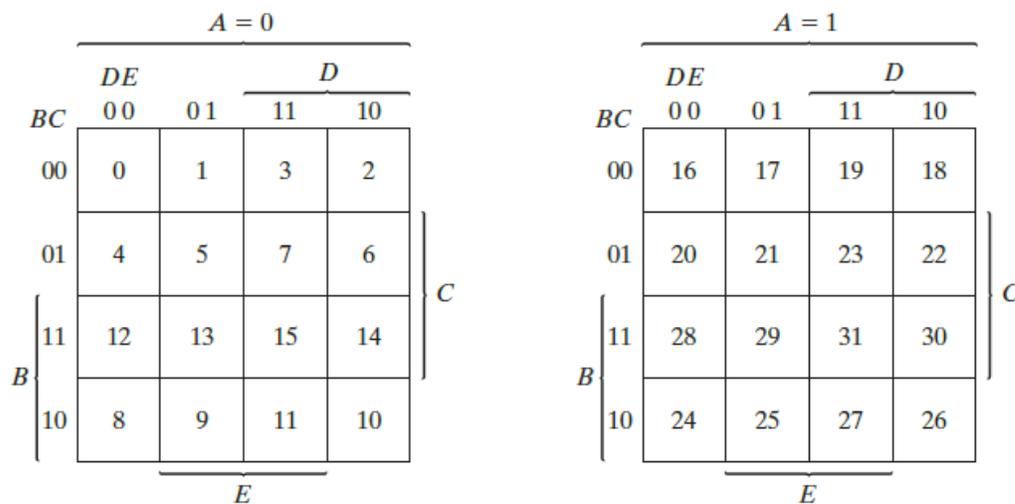
Fuente: elaboración propia, empleando Excel.

### 2.3.2.4. Mapa para cinco variables

El uso de mapas para más de cuatro variables se va tornando cada vez más difícil porque la cantidad de minitérminos y cuadros se eleva. Para un mapa de 5 variables se necesitan 32 cuadros. El mapa de cinco variables se muestra en la figura 39 y está compuesto por dos mapas de cuatro variables. Se puede ver que la variable *A* es la que diferencia estos dos mapas.

El mapa de cuatro variables de la izquierda representa los 16 cuadrados en los que  $A = 0$ ; el otro representa los cuadrados cuando  $A = 1$ . Los minterminos 0 al 15 corresponden al mapa de la izquierda los minterminos 16 al 31 corresponden al mapa de la derecha. Cada mapa de cuatro variables conserva la adyacencia que se conoce. Cada cuadro del mapa  $A = 0$  es adyacente al cuadro correspondiente al mapa  $A = 1$ . Por ejemplo, el mintermino 0 es adyacente al mintermino 16 y, por ello, se puede simplificarlos entre mapas.

Figura 38. Mapa de cinco variables



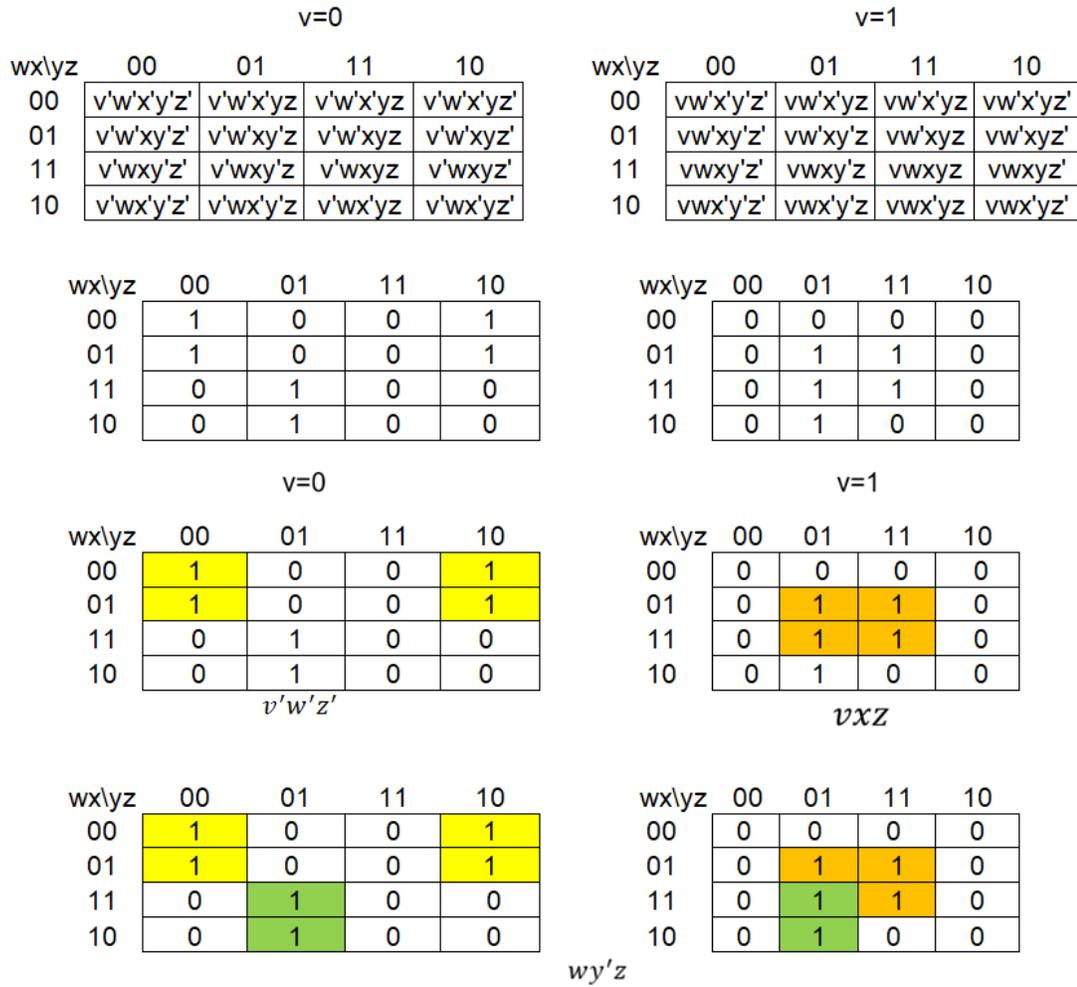
Fuente: MANO, Morris. *Diseño digital*. p 75.

Ejemplo 49: para la siguiente función, encuentre la manera más simplificada de expresarla.

$$F(v, w, x, y, z) = \sum (0, 2, 4, 6, 9, 13, 21, 23, 25, 29, 31)$$

$$F = v'w'z' + vxz + wy'z$$

Figura 39. Mapa de ejemplo 49



Fuente: elaboración propia, empleando Excel.

Se ve que se inicia de igual manera que los pasos anteriores, pero al final (cuadros verdes) se aplica la nueva regla para este tipo de mapa. Hay que recordar que los minitérminos son adyacentes en correspondencia a la posición entre ambos mapas de cuatro variables. Entonces agrupar los minitérminos 13, 9, 31 y 27 siempre de la misma manera que se ha estado trabajando.

Figura 40. **Mapa de seis variables**

		u=0 v=0						u=0 v=1			
wxlyz		00	01	11	10	wxlyz		00	01	11	10
00		0	1	3	2	00		16	17	19	18
01		4	5	7	6	01		20	21	23	22
11		12	13	15	14	11		28	29	31	30
10		8	9	11	10	10		24	25	27	26

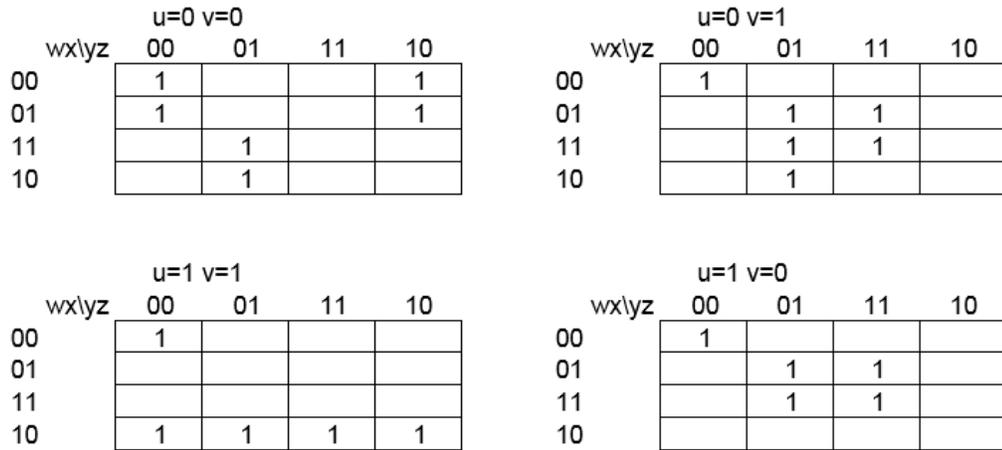
		u=1 v=1						u=1 v=0			
wxlyz		00	01	11	10	wxlyz		00	01	11	10
00		48	49	51	50	00		32	33	35	34
01		52	53	55	54	01		36	37	39	38
11		60	61	63	62	11		44	45	47	46
10		56	57	59	58	10		40	41	43	42

Fuente: elaboración propia, empleando Excel.

### 2.3.2.5. Mapa de seis variables

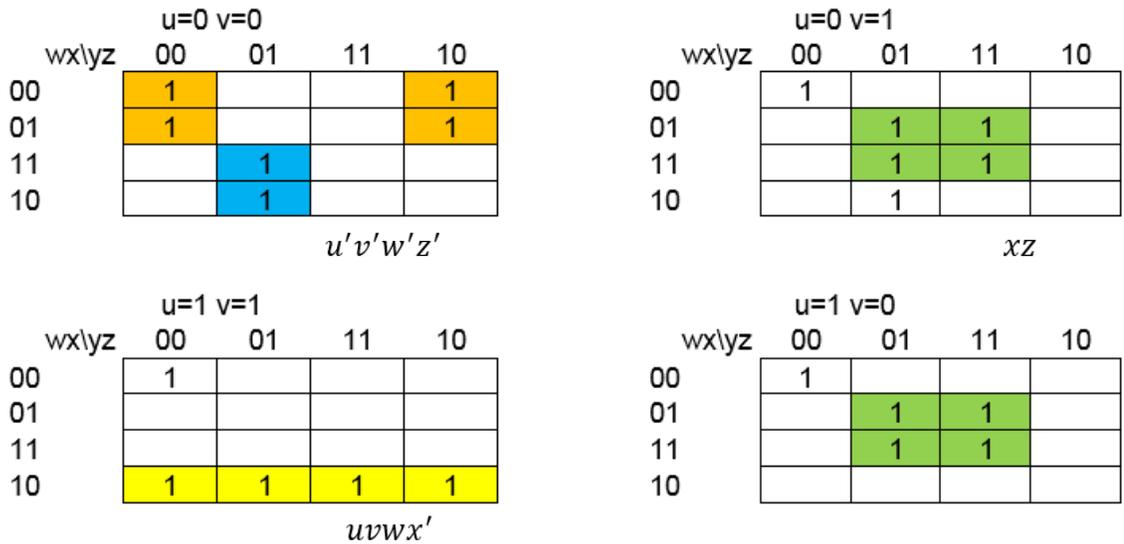
Tomando como referencia el ejercicio anterior, es posible llegar a trabajar un mapa que contenga la cantidad de seis variables, pero inclusive para cinco variables es poco práctico la realización de mapas de Karnaugh a mano. Para este tipo de casos se utiliza páginas de internet en donde el sitio web es capaz de realizar los cálculos para que darnos el resultado.

Figura 41. Mapa 1 de ejemplo 50



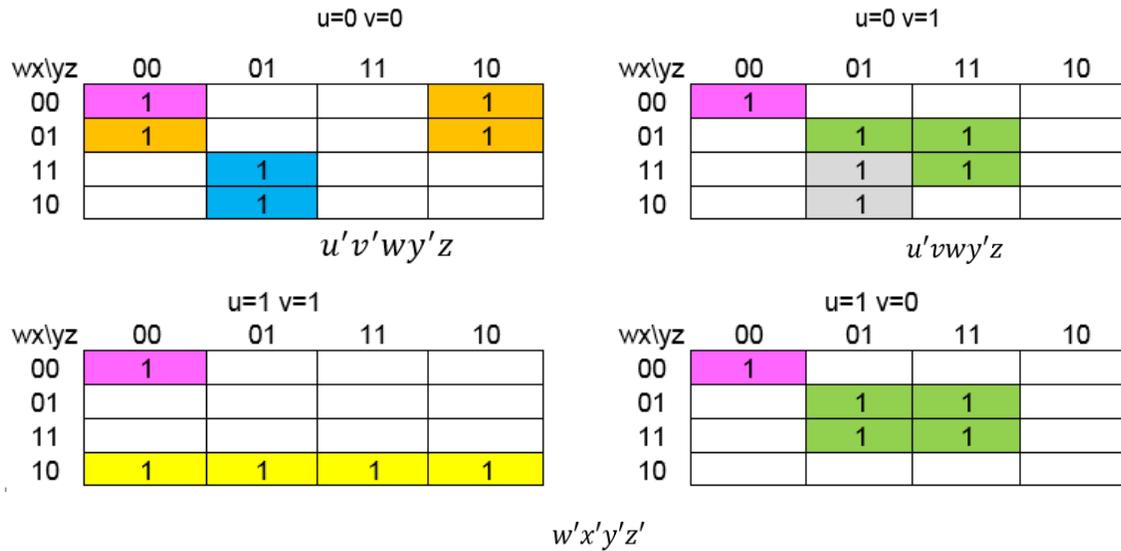
Fuente: elaboración propia, empleando Excel.

Figura 42. Mapa 2 de ejemplo 50



Fuente: elaboración propia, empleando Excel.

Figura 43. Mapa 3 de ejemplo 50



Fuente: elaboración propia, empleando Excel.

Con fines didácticos se ejemplificará como realizar uno de estos mapas. Como se ve se tienen 64 minterminos y el mapa está conformado por 64 cuadros. El cambio es en la adyacencia de los cuadros.

Ahora los cuatro mapas tendrán adyacencia en la posición correspondiente, sin excluir las reglas de adyacencia de cada mapa de cuatro variables. Por ejemplo, el mintermino 0, 16, 32 y 48 son adyacentes entre sí. Como se verá es un procedimiento largo en el que es posible cometer varios errores dados a la cantidad de minterminos que se manejan.

Como se ve el procedimiento es extenso y puede haber errores con facilidad, por esto se recomienda utilizar algún *software* especializado para resolver este tipo de problemas.

Ejemplo 50: para la siguiente función, encuentre la manera más simplificada de expresarla.

$$F(v, w, x, y, z) = \sum (0,2,4,6,9,13,16,21,23,25,29,31,32,37,39,45,47,48,56,57,58,59)$$

Se inicia colocando cada uno de los unos de la función en las casillas correspondientes del mapa (figura 42). Luego se reducen los grupos de unos adyacentes (figura 43 y figura 44).

$$F = u'v'w'z' + uvwx' + xz + u'v'wy'z + u'vwy'z + w'x'y'z'$$

### 2.3.3. Simplificación a producto de sumas

Como se ha visto el mapa de Karnaugh ofrece una gran ayuda para la simplificación de compuertas. Este método ofrece los resultados con una suma de productos, pero para el diseño digital a veces es necesario que la simplificación sea dada en productos de suma. Con una modificación al proceso de simplificación del método del mapa se puede llegar a tener como resultado un producto de suma.

El procedimiento consiste en obtener una función de los valores que no se desean, en este caso los cero de la función. Esto dará un resultado de suma de productos, pero dado a que es el complemento de una función se puede aplicar el teorema de DeMorgan para convertirlo en producto de sumas.

Ejemplo 51: simplificar la siguiente función booleana en forma de producto de sumas.

$$F(w, x, y, z) = \sum (0,1,2,5,8,9,10)$$

Haciendo el mismo procedimiento que en los mapas anteriores, pero recordar que los valores de interés son los ceros se obtiene:

$$F' = wx + yz + xz'$$

Figura 44. Mapa de ejemplo 51

wx\yz	00	01	11	10
00	w'x'y'z'	w'x'yz	w'x'yz	w'x'yz'
01	w'xy'z'	w'xy'z	w'xyz	w'xyz'
11	wxy'z'	wxy'z	wxyz	wxyz'
10	wx'y'z'	wx'y'z	wx'yz	wx'yz'

wx\yz	00	01	11	10
00	1	1	0	1
01	0	1	0	0
11	0	0	0	0
10	1	1	0	1

wx\yz	00	01	11	10
00	1	1	0	1
01	0	1	0	0
11	0	0	0	0
10	1	1	0	1

wx\yz	00	01	11	10
00	1	1	0	1
01	0	1	0	0
11	0	0	0	0
10	1	1	0	1

Fuente: elaboración propia, empleando Excel.

Que será el complemento de la función  $F$ , luego de esto se aplica el teorema de DeMorgan:

$$F' = (wx + yz + xz')$$

$$F = (wx + yz + xz')'$$

$$F = (wx)'(yz)'(xz)'$$

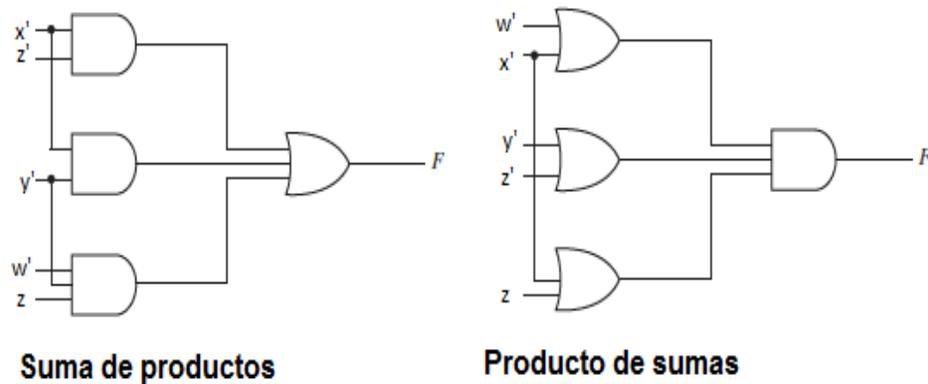
$$F = (w' + x')(y' + z')(x' + z)$$

Si se hubiera utilizado una suma de productos el resultado hubiera sido el siguiente:

$$F = x'z' + x'y' + w'y'z$$

Con cada uno se obtiene los mismos resultados con la función F, pero difieren en el uso de componentes:

Figura 45. **Circuitos equivalentes de ejemplo 51**



Fuente: MANO, Morris. *Diseño digital*. p 78.

### 2.3.4. Condición indiferencia

Como se ha visto anteriormente, las funciones están definidas respecto de los valores de interés de alguna combinación específica de valores de las variables booleanas, pero muchas veces hay combinaciones que realmente no inciden en su resultado. Estas también se pueden representar en un mapa de Karnaugh. A esto se le llama condición de indiferencia y ayuda a disminuir más aún el mapa de Karnaugh.

Ejemplo 52: simplifique la siguiente función booleana

$$F(w, x, y, z) = \sum (1,3,7,11,15)$$

Que tiene las condiciones de indiferencia

$$d(w, x, y, z) = \sum (0,2,5)$$

Figura 46. Mapa de ejemplo 52

wx\yz	00	01	11	10
00	w'x'y'z'	w'x'yz	w'xyz	w'xyz'
01	w'xy'z'	w'xy'z	w'xyz	w'xyz'
11	wxy'z'	wxy'z	wxyz	wxyz'
10	wx'y'z'	wx'y'z	wx'yz	wx'yz'

wx\yz	00	01	11	10
00	X	1	1	X
01	0	X	1	0
11	0	0	1	0
10	0	0	1	0

wx\yz	00	01	11	10
00	X	1	1	X
01	0	X	1	0
11	0	0	1	0
10	0	0	1	0

wx\yz	00	01	11	10
00	X	1	1	X
01	0	X	1	0
11	0	0	1	0
10	0	0	1	0

Fuente: elaboración propia, empleando Excel.

$$F = w'x' + yz$$

Los minterminos que tengan esta condición serán llamados mintermino indiferente. Para distinguir los minterminos indiferentes en un mapa de Karnaugh y/o en una tabla de verdad se usa una X en lugar de unos y ceros.

Así, una  $X$  en un cuadro del mapa indica que no importa si se asigna el valor de cero o de uno a  $F$  para el minitérmino en cuestión. Al escoger cuadrados adyacentes para simplificar la función, se puede suponer que los minitérminos indiferentes son cero o uno, lo que más convenga.

Como se vio en el ejemplo anterior, dos de los minitérminos indiferentes, se han trabajado como si fueran igual a uno. De este mismo modo, de ser necesario, se hubieran podido trabajar como si fueran cero.

### **2.3.5. Implementación con compuertas NAND y NOR**

Muchos circuitos digitales se construyen con compuertas NAND y NOR en lugar de compuertas AND y OR. Las primeras son fáciles de fabricar con componentes electrónicos y son las compuertas básicas empleadas en todas las familias de lógica de CI.

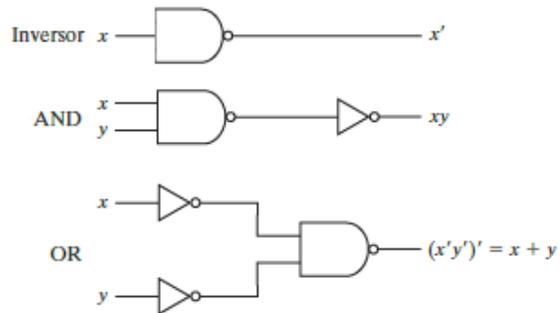
#### **2.3.5.1. Implementación de compuertas NAND y NOR**

Ahora se verá la aplicación de las compuertas NAND y NOR en los circuitos digitales. Se apreciará que son compuertas fundamentales para la creación de hardware digital.

#### **2.3.5.2. Compuerta NAND**

Las compuertas NAND son más sencillas de fabricar que su contraparte negada, por ello ha ganado popularidad a la hora de realizar piezas de circuitería digital.

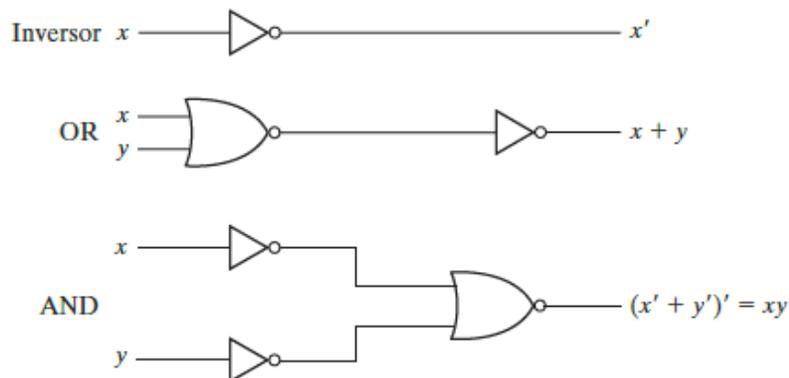
Figura 47. **Compuertas AND y OR utilizando NAND**



Fuente: MANO, Morris. *Diseño digital*. p 82.

Se dice que la compuerta NAND es una compuerta universal porque cualquier sistema digital se puede implementarse con ella. Como se muestra en la figura 48 se puede llegar a representar cualquier otra compuerta lógica con la ayuda de la compuerta NAND.

Figura 48. **Compuerta AND y OR utilizando NOR**



Fuente: MANO, Morris. *Diseño digital*. p 87.

### 2.3.5.3. Compuerta NOR

La compuerta NOR es otra compuerta universal que sirve para implementar cualquier función booleana. Al igual que la compuerta NAND es mucho más sencilla de implementar que su contraparte negada. Como se muestra en la figura 49 se puede llegar a representar cualquier otra compuerta lógica con la ayuda de la compuerta NOR.

### 2.3.6. Compuerta OR exclusiva

La función OR exclusiva, se denota por el símbolo  $\oplus$ , es una operación lógica que efectúa la operación booleana siguiente:

$$x \oplus y = xy' + x'y$$

Es igual a uno, si solo  $x$  es igual a uno o solo  $y$  es igual a uno, pero no si ambas son uno. El NOR exclusivo, también llamada equivalencia, realiza la operación booleana siguiente:

$$(x \oplus y)' = xy + x'y'$$

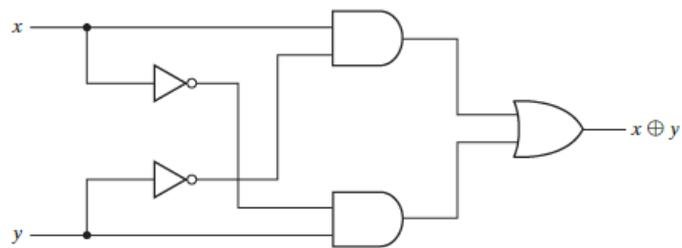
Con ayuda de manipulación algebraica se puede llegar a las siguientes conclusiones:

$$\begin{aligned}(x \oplus y)' \\ (xy' + x'y)' \\ (x' + y)(x + y')\end{aligned}$$

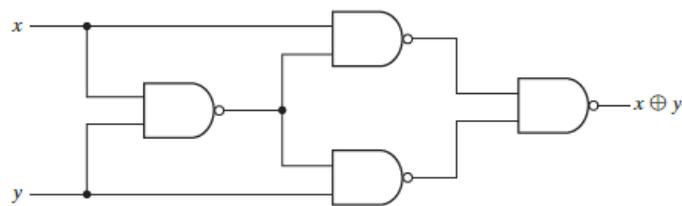
$$x'x + xy + x'y' + yy'$$

$$xy + x'y'$$

Figura 49. **Compuerta OR exclusiva con AND-OR-NOT y NAND**



a) Con compuertas AND-OR-NOT



b) Con compuertas NAND

Fuente: MANO, Morris. *Diseño digital*. p 95.

Algunas de las propiedades que se pueden resaltar son las siguientes:

$$x \oplus 0 = x$$

$$x \oplus 1 = x'$$

$$x \oplus x = 0$$

$$x \oplus x' = 1$$

$$x \oplus y' = x' \oplus y = (x \oplus y)'$$

$$A \oplus B = B \oplus A$$

$$(A \oplus B) \oplus C = A \oplus (B \oplus C) = A \oplus B \oplus C$$

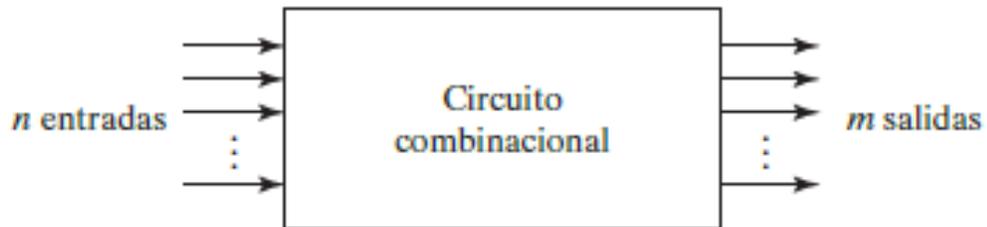
## 2.4. Lógica combinacional

Un circuito combinacional consiste en compuertas lógicas cuyas salidas en cualquier momento están determinadas por la combinación actual de entradas. Consiste en variables de entrada, compuertas lógicas y variables de salida.

### 2.4.1. Introducción

Cuando se trabaja una circuitería combinacional, es cuando se tiene una cierta cantidad de entradas, luego un grupo de compuertas lógicas interpretan estas señales de entradas, para dar lugar a señales binarias de salida.

Figura 50. Diagrama de bloques de circuito combinacional



Fuente: MANO, Morris. *Diseño digital*. p 111.

Con  $n$  variables de entrada hay  $2^n$  posibles combinaciones de entradas binarias.

Para cada una de esas combinaciones, hay un posible valor de salida. Por tanto, es posible especificar un circuito combinacional con una tabla de verdad que presenta los valores de salida para cada combinación de variables de entrada. También es factible describir un circuito combinacional con  $m$  funciones booleanas, una para cada variable de salida. Cada función de salida se expresa en términos de las  $n$  variables de entrada.

#### **2.4.2. Procedimiento de análisis**

Este proceso parte de un diagrama lógico dado y culmina en un conjunto de funciones booleanas, una tabla de verdad o una posible explicación del funcionamiento del circuito. Se procede a analizar el diagrama para verificar que sea combinacional. Si cuenta con algún elemento de almacenamiento o circuito de retroalimentación, será un diagrama secuencial, que se verá más adelante.

Si lo que se tiene es un diagrama lógico los pasos para obtener la función o funciones booleanas de salida es el siguiente:

- Rotular con símbolos arbitrarios todas las salidas de compuertas que son funciones variables de entrada. Determinar las funciones booleanas para cada salida de compuerta.
- Rotular con otros símbolos arbitrarios las compuertas que son funciones de variables de entrada y de compuertas previamente rotuladas. Obtener las funciones booleanas de estas compuertas.
- Repetir el proceso de bosquejo en el paso 2 hasta obtener las salidas del circuito.
- Por situaciones repetidas de funciones previamente definidas, obtener las funciones booleanas de salida en términos de variables de entrada.

Para obtener la tabla de verdad directamente del diagrama lógico sin tener que deducir las funciones booleanas, se procede así.

- Determinar el número de variables de entrada del circuito. Para  $n$  entradas, forme las  $2^n$  posibles combinaciones y haga una lista de los números binarios de 0 a  $2^n - 1$  en una tabla.
- Rotular las salidas de compuertas selectas con símbolos arbitrarios.
- Obtener la tabla de verdad para las salidas de aquellas compuertas que son función únicamente de las variables de entrada.
- Obtener la tabla de verdad para las salidas de aquellas compuertas que son función de valores previamente definidos, hasta llenar las columnas de todas las salidas.

### **2.4.3. Procedimiento de diseño**

Cuando se diseña un circuito combinacional, se inicia partiendo del problema y termina con un diagrama lógico de circuitos o un conjunto de funciones booleanas. El procedimiento es el siguiente:

- Dar las especificaciones del circuito, y deducir el número requerido de entradas y salidas; asignar un símbolo a cada una.
- Deducir la tabla de verdad que define la relación requerida entre las entradas y las salidas.
- Obtener las funciones booleanas simplificadas para cada salida en función de las variables de entrada.
- Dibujar el diagrama lógico y verificar que el diseño sea correcto.

Ejemplo 53: se quiere implementa un sistema de una luz de alarma y tres sensores digitales. Se llamará "A" a la luz de alarma, y "x", "y" y "z" a los sensores digitales. La alarma A tiene que encender si al menos dos de los sensores es activado.

Tabla XLVII. **Tabla de verdad del ejemplo 53**

x	y	z	A
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Fuente: elaboración propia, empleando Excel.

Figura 51. **Mapa del ejemplo 53**

x\yz	00	01	11	10
0	x'y'z'	x'y'z	x'yz	x'yz'
1	xy'z'	xy'z	xyz	xyz'

x\yz	00	01	11	10
0	0	0	1	0
1	0	1	1	1

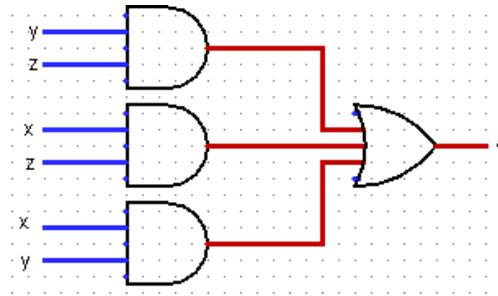
x\yz	00	01	11	10
0			1	
1	0	1	1	1

x\yz	00	01	11	10
0	0	0	1	0
1	0	1	1	1

Fuente: elaboración propia, empleando Excel.

$$A = yz + xz + xy$$

Figura 52. **Circuito lógico de ejemplo 53**



Fuente: elaboración propia, empleando Logisim.

#### 2.4.4. Aplicaciones utilizando lógica combinacional

Al adentrarse en la lógica combinacional, se aprecia que hay una gran cantidad de aplicaciones que se pueden implementar en los circuitos. Entre las principales se tienen algunas operaciones aritméticas como la suma y la resta.

##### 2.4.4.1. Semisumador

Este es un circuito combinacional que realiza la suma de dos bits. Este necesita dos entradas binarias y dos salidas binarias. Las variables de entrada designan los bits sumados y las de salida, la suma y el acarreo. En la tabla XLVIII se muestra la tabla de verdad del sumador. Como se ve  $x$  y  $y$  son los números por sumar,  $S$  es el bit menos significativo de la suma y  $C$  es el más significativo. Con la tabla de verdad también se puede deducir la función booleana.

Tabla XLVIII. **Tabla de verdad de semisumador**

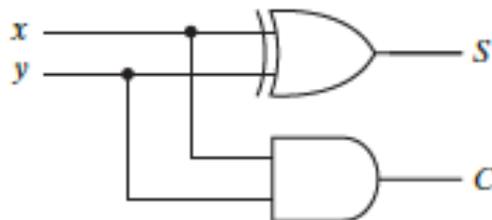
$x$	$y$	$C$	$S$
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Fuente: MANO, Morris. *Diseño digital*. p 119.

$$S = x'y + xy'$$

$$C = xy$$

Figura 53. **Circuito lógico del semisumador**



Fuente: MANO, Morris. *Diseño digital*. p 120.

#### 2.4.4.2. **Sumador completo**

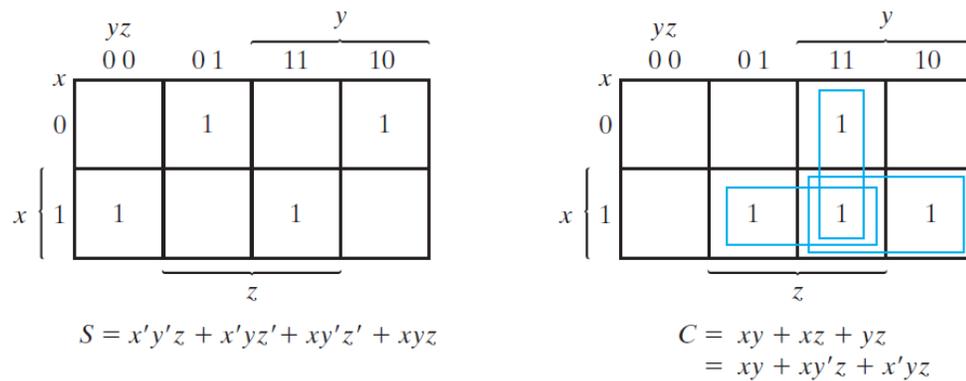
Es un circuito combinacional que forma la suma aritmética de tres bits. Tiene tres entradas y dos salidas. Las entradas  $x$ ,  $y$  y  $z$  son los números que se desean sumar,  $S$  el resultado de la suma y  $C$  el acarreo generado en la suma.

Tabla XLIX. **Tabla de verdad de sumador completo**

<i>x</i>	<i>y</i>	<i>z</i>	<i>C</i>	<i>S</i>
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

Fuente: MANO, Morris. *Diseño digital*. p 120.

Figura 54. **Mapa de Karnaugh de sumador completo**

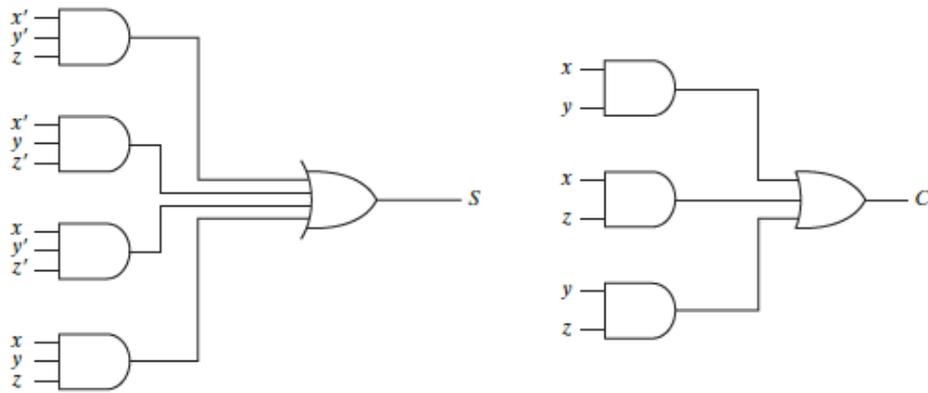


Fuente: MANO, Morris. *Diseño digital*. p 121.

$$S = x'y'z + x'yz' + xy'z' + xyz$$

$$C = xy + xz + yz$$

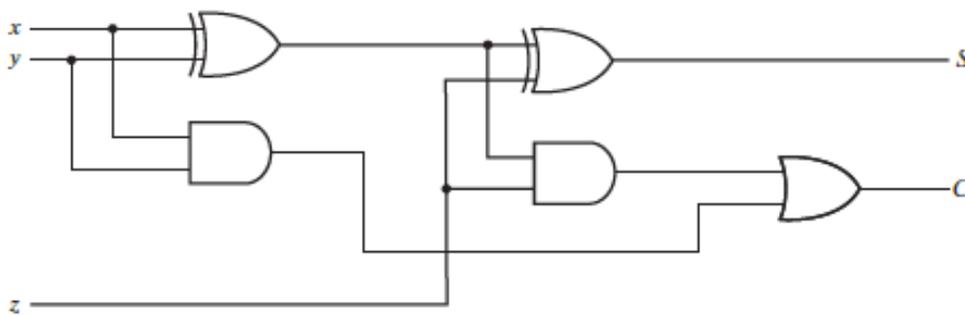
Figura 55. **Circuitos lógicos de sumador completo**



Fuente: MANO, Morris. *Diseño digital*. p 121.

Simplificando un poco y utilizando la compuerta XOR se puede llegar al circuito de la figura 56. Observar que este circuito tiene un nivel de compuertas mayor.

Figura 56. **Circuito lógico de sumador completo con XOR**



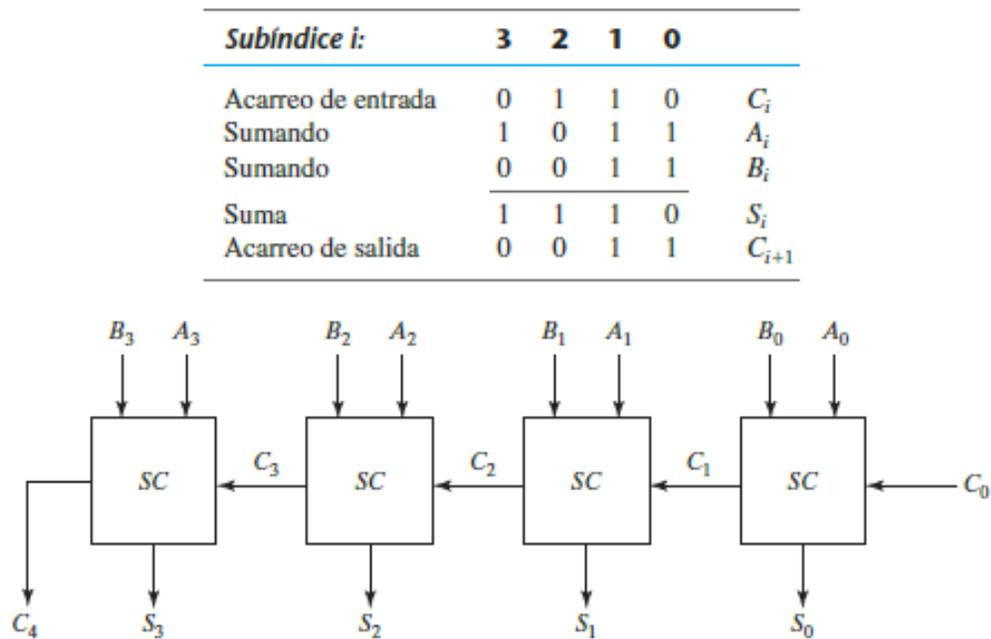
Fuente: MANO, Morris. *Diseño digital*. p 122.

### 2.4.4.3. Sumador binario

Es un circuito digital que produce la suma aritmética de dos números binarios de  $n$  bits. Este se construye en base a sumadores completos, donde se conectan en cascada, y el acarreo de salida irá conectado al bit menos significativo del siguiente sumador completo de la cadena. A este bit menos significativo se le llamará acarreo de entrada.

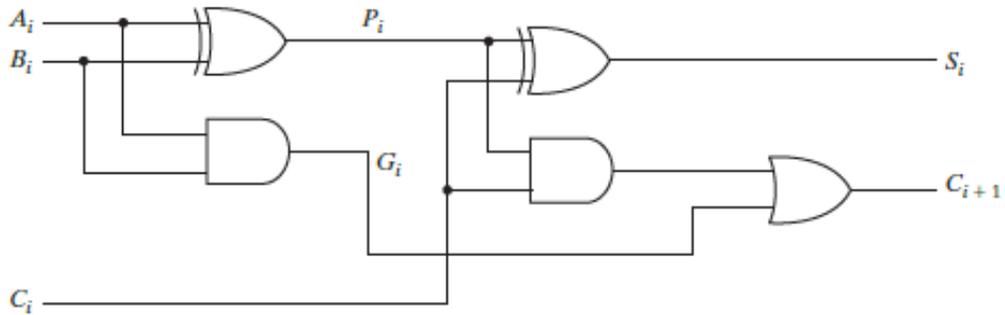
En la figura 57 se vislumbra el diagrama de bloques del sumador binario, así como un análisis para la construcción del sumador completo en cada etapa.

Figura 57. Diagrama de bloques y funcionamiento de sumador binario



Fuente: MANO, Morris. *Diseño digital*. p 122.

Figura 58. **Circuito de cada sumador completo en el sumador binario**



Fuente: MANO, Morris. *Diseño digital*. p 123.

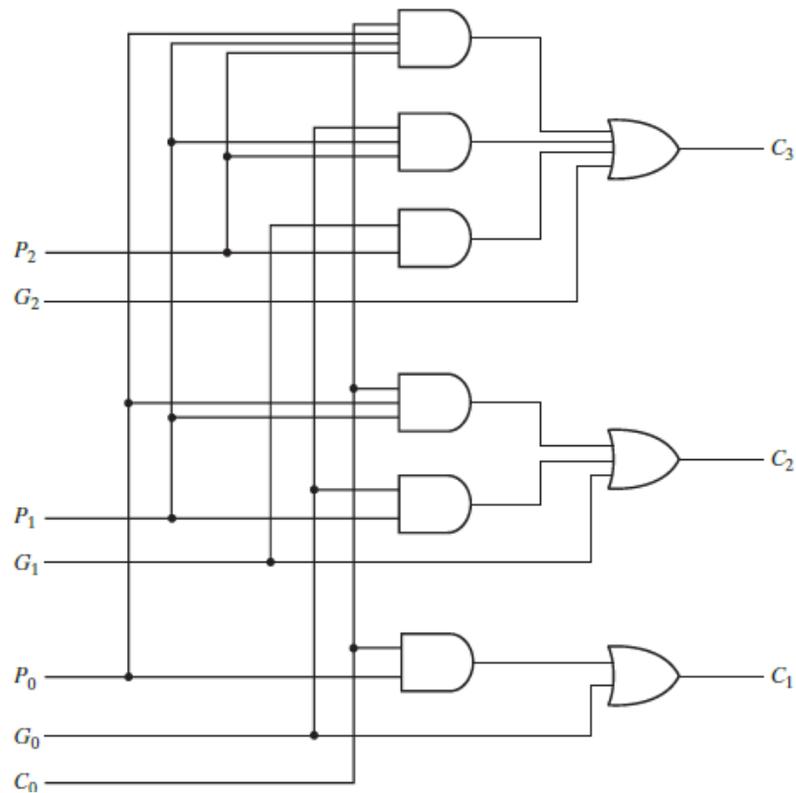
Las letras A y B representan los bits de los dos números que se desean sumar. La letra C representa los acarreos que se generan en las sumas. Las letras S representan el resultado de la suma en cada bit, donde el bit de la derecha es el menos significativo. En la figura 58 se aprecia cómo sería la distribución de entradas y salidas en un circuito del sumador completo cuando se desea implementarlo en el sumador binario. Si es un sumador de cuatro bits se tendría que tener cuatro sumadores completos.  $C_{i+1}$  será el acarreo que se genera tras la suma y  $C_i$  el acarreo de entrada de la suma.  $C_{i+1}$  será el acarreo  $C_i$  en la siguiente casilla.

#### 2.4.4.4. Propagación del acarreo

Como se ve en la figura 58, el acarreo de la suma de la casilla actual ( $C_i$ ), afectará a la suma de la casilla siguiente ( $C_{i+1}$ ). El acarreo  $C_{i+1}$  es generado por un circuito de dos niveles (compuertas AND y OR). Luego en la siguiente casilla, el acarreo  $C_{i+1}$  actual será el acarreo  $C_i$ , y tendrá que pasar por las compuertas AND y OR nuevamente. Sucesivamente hasta llegar al bit más significativo.

Por lo que se entiende que el acarreo inicial tiene que pasar por  $2n$  niveles. Por otro lado, la salida S solo pasará por una compuerta de dos niveles en cada una de las casillas. Entonces las casillas S llegarán antes que los acarros C. Esto genera un atraso y hace que los cálculos, al inicializar el circuito, se tornen impredecibles.

Figura 59. **Diagrama lógico del generador del acarreo anticipado**



Fuente: MANO, Morris. *Diseño digital*. p 125.

Una solución para esta problemática es la aplicación del acarreo anticipado, que consiste en encontrar otra forma que el acarreo pueda generarse con un circuito de dos niveles.

Para esto revisar nuevamente la figura 59 y se concentrará en las salidas  $P_i$  y  $G_i$ .  $G_i$  se llama acarreo generado y produce un acarreo de uno si tanto  $A_i$  como  $B_i$  son uno y  $P_i$  se llama el acarreo propagado porque es el término asociado a la propagación del acarreo  $C_i$  y  $C_{i+1}$ .

$$P_i = A_i \oplus B_i$$

$$G_i = A_i B_i$$

Figura 60. **Funciones booleanas de acarreo de salida**

$C_0 =$  acarreo de entrada

$$C_1 = G_0 + P_0 C_0$$

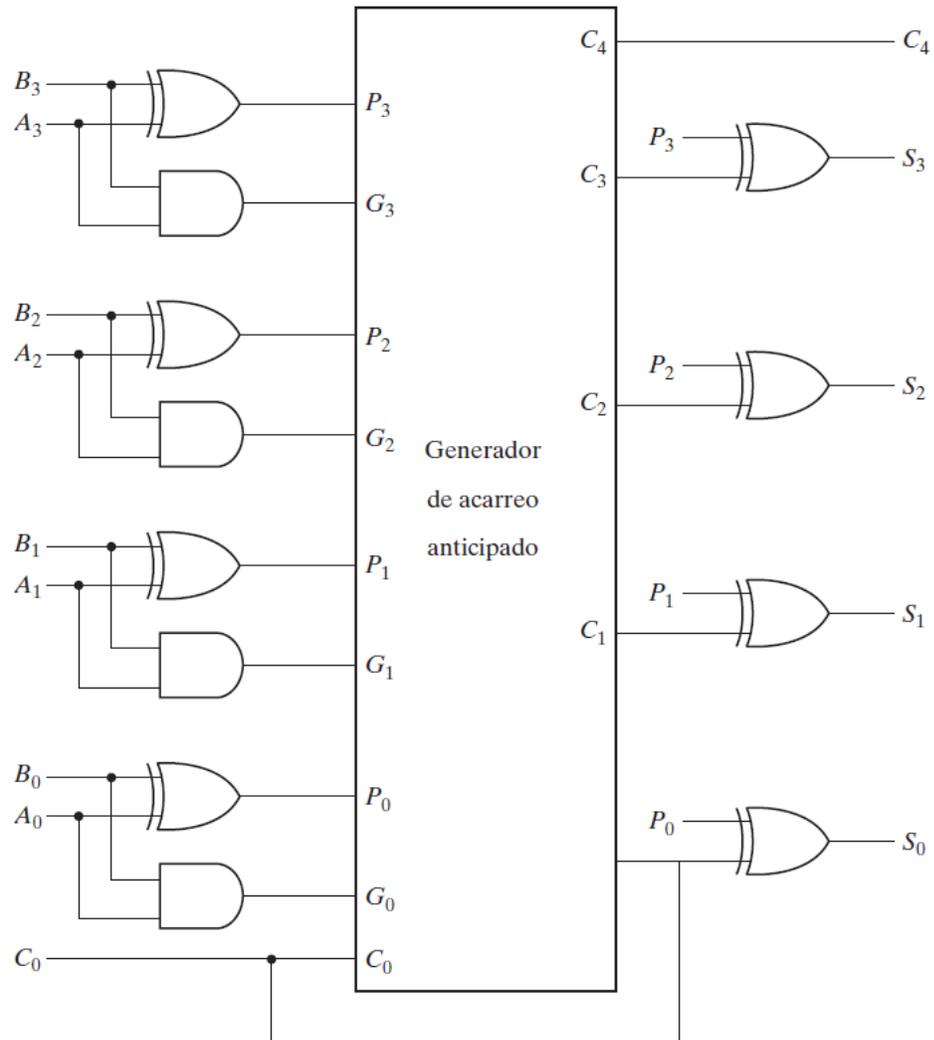
$$C_2 = G_1 + P_1 C_1 = G_1 + P_1(G_0 + P_0 C_0) = G_1 + P_1 G_0 + P_1 P_0 C_0$$

$$C_3 = G_2 + P_2 C_2 = G_2 + P_2 G_1 + P_2 P_1 G_0 + P_2 P_1 P_0 C_0$$

Fuente: MANO, Morris. *Diseño digital*. p 124.

Ahora se escriben las funciones booleanas para el acarreo de salida en cada etapa: En la figura 60 se muestra las funciones booleanas para el acarreo de salida en cada una de las etapas. En la figura 60 se muestra el diagrama lógico de las funciones booleanas. Como se ve, ahora se tiene un circuito de nivel dos para un sumador de cuatro bits. Esto ayudará a que no exista ningún retardo entre los resultados de las sumas  $S$  y los acarreo  $C$ . El circuito de la propagación del acarreo se puede ver en la figura 62.

Figura 61. Sumador binario con propagación del acarreo

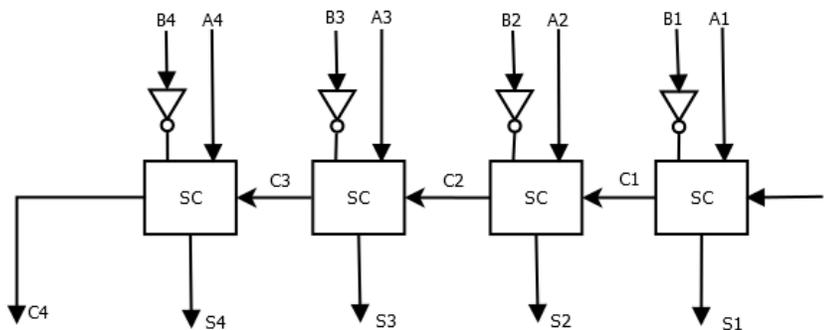


Fuente: MANO, Morris. *Diseño digital*. p 126.

#### 2.4.4.5. Restador binario

El proceso ideal para ejecutar la resta entre dos números binarios sin signo es utilizando complementos, como se explicó en secciones anteriores. Hay que recordar que la resta  $A - B$  se efectúa obteniendo el complemento a dos de  $B$  y sumándole  $A$ . El circuito para restar  $A - B$  consiste en un sumador completo con inversores colocados entre cada entrada de datos  $B$  y la entrada correspondiente del sumador completo y una suma de uno en el bit menos significativo. Como se ve es muy parecido al sumador completo, solo que con la diferencia de invertir las variables de entrada de  $B$  y que el acarreo inicial sea cero.

Figura 62. Diagrama de bloques de resta binaria



Fuente: elaboración propia, empleando Dia.

#### 2.4.4.6. Sumador-restador binario

Las operaciones de suma y resta se pueden combinar en un solo circuito. Esto se realiza agregando una compuerta XOR a cada sumador completo del sumador de cuatro bits que se ha visto previamente.



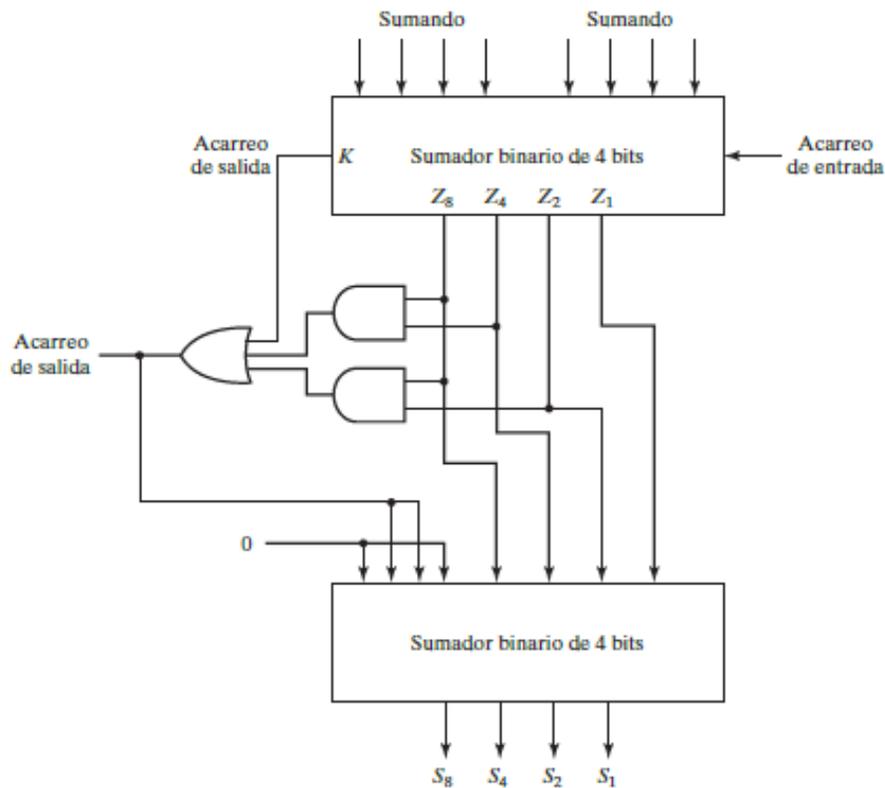
Figura 64. **Suma binaria y suma BCD**

Suma binaria					Suma BCD					Decimal
K	Z <sub>8</sub>	Z <sub>4</sub>	Z <sub>2</sub>	Z <sub>1</sub>	C	S <sub>8</sub>	S <sub>4</sub>	S <sub>2</sub>	S <sub>1</sub>	
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	1	1
0	0	0	1	0	0	0	0	1	0	2
0	0	0	1	1	0	0	0	1	1	3
0	0	1	0	0	0	0	1	0	0	4
0	0	1	0	1	0	0	1	0	1	5
0	0	1	1	0	0	0	1	1	0	6
0	0	1	1	1	0	0	1	1	1	7
0	1	0	0	0	0	1	0	0	0	8
0	1	0	0	1	0	1	0	0	1	9
0	1	0	1	0	1	0	0	0	0	10
0	1	0	1	1	1	0	0	0	1	11
0	1	1	0	0	1	0	0	1	0	12
0	1	1	0	1	1	0	0	1	1	13
0	1	1	1	0	1	0	1	0	0	14
0	1	1	1	1	1	0	1	0	1	15
1	0	0	0	0	1	0	1	1	0	16
1	0	0	0	1	1	0	1	1	1	17
1	0	0	1	0	1	1	0	0	0	18
1	0	0	1	1	1	1	0	0	1	19

Fuente: MANO, Morris. *Diseño digital*. p 129.

Ahora hay que suponer que la suma aritmética de dos números BCD junto con un acarreo de entrada de alguna etapa previa. Dado a que los números BCD no son mayores de nueve, entonces  $9 + 9 + 1 = 19$  será el número más grande que se necesitaría representar. El uno extra es por el bit de acarreo inicial. Se ve en la figura 65 que de los números del cero al nueve la representación binaria en ambos casos es la misma, pero de los números mayores a nueve cambia la representación. Como se ve hay una cierta discrepancia en la representación de números binarios con el sistema binario tradicional y la codificación BCD, por lo que no se puede utilizar el sumador binario sin antes realzar una pequeña modificación en el circuito.

Figura 65. Diagrama de bloques de sumador BCD



Fuente: MANO, Morris. *Diseño digital*. p 130.

Si la suma binaria es mayor a 9, entonces se le sumará un seis (0110), el cual convertirá el número en representación tradicional a su valor equivalente en BCD. Esto para utilizar el sumador binario y representarlo en BCD. Si la suma es menor a nueve no existirá ninguna variación en el procedimiento. Con la tabla precedente, se ve que los valores mayores a nueve tienen la casilla  $Z_8$  igual a uno. Pero los números 8 ( $1000_2$ ) y 9 ( $1001_2$ ) también lo tienen, por lo que hay que especificar que cuando tenga la casilla  $Z_8$  con  $Z_4$  o  $Z_2$  con uno, será un número mayor a nueve.

Otra condición es cuando sucede un acarreo en la suma, en el que (por obvias razones) será mayor a nueve. Esto se hace con una compuerta OR con las condiciones AND establecidas.

#### 2.4.4.8. Multiplicador binario

Hay que recordar que la multiplicación binaria se realiza de la misma manera que para números decimales.

Figura 66. **Multiplicación binaria**

$$\begin{array}{r}
 \phantom{A_1} B_1 \phantom{A_0} \\
 \phantom{A_1} A_1 \phantom{A_0} \\
 \hline
 A_0 B_1 \phantom{A_0} \\
 A_1 B_1 \phantom{A_0} \\
 A_1 B_0 \\
 \hline
 C_3 \phantom{C_2} \phantom{C_1} \phantom{C_0} \\
 C_2 \phantom{C_1} \phantom{C_0} \\
 C_1 \phantom{C_0} \\
 C_0
 \end{array}$$

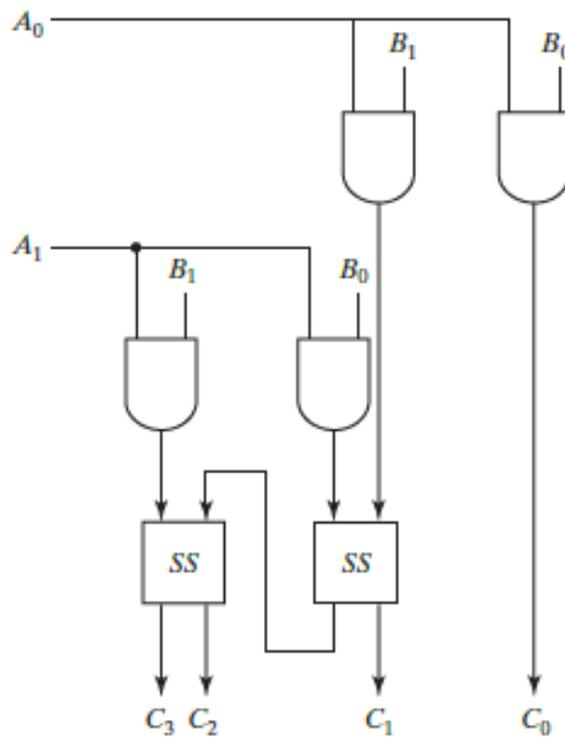
Fuente: MANO, Morris. *Diseño digital*. p 131.

##### 2.4.4.8.1. Multiplicador binario con sumador simple

El multiplicando se multiplica por cada bit del multiplicador, comenzando con el bit menos significativo. Cada una de estas multiplicaciones forma un producto parcial. Los productos parciales sucesivos se desplazan una posición izquierda.

El producto final se obtiene sumando los productos parciales. Como se ve en la figura 68, el multiplicador binario está compuesto por compuertas AND y sumadores simples. El bit menos significativo es simplemente una compuerta AND, debido al corrimiento de la multiplicación. Luego los siguientes bits, son obtenidos tras la suma de un sumador simple de los resultados obtenidos de las compuertas AND. Con la ayuda de un sumador completo se puede extender los bits del multiplicador.

Figura 67. **Diagrama lógico de multiplicador binario**



Fuente: MANO, Morris. *Diseño digital*. p 131.

#### 2.4.4.8.2. Multiplicador binario con sumadores completos

Se puede construir de forma similar un multiplicador binario de más bits con circuitos combinatoriales. Se obtiene el AND de un bit del multiplicador y cada bit del multiplicando en tantos niveles como haya bits en el multiplicador. La salida binaria de cada nivel de compuertas AND se suma al producto parcial del nivel anterior para formar un nuevo producto parcial. El último nivel genera el producto. Si el multiplicador tiene  $J$  bits y el multiplicando tiene  $K$  bits, se necesitarán  $J \times K$  compuertas AND y  $J - 1$  sumadores de  $K$  bits para obtener un producto de  $J + K$  bits. El circuito se puede visualizar en la figura 69.

#### 2.4.4.9. Comparador de magnitudes

Cuando se comparan dos números lo que se desea saber es que si son iguales o uno mayor que el otro. Un comparador de magnitudes es un circuito combinatorial que compara dos números  $A$  y  $B$ , y determina si  $A > B$ ,  $A < B$  o  $A = B$ . Este circuito cuenta con una cantidad de entradas de  $2^{2n}$  donde  $n$  es la cantidad de bits. Para ejemplificar se hará un comparador de magnitudes con dos números de cuatro bits ( $A$  y  $B$ ).

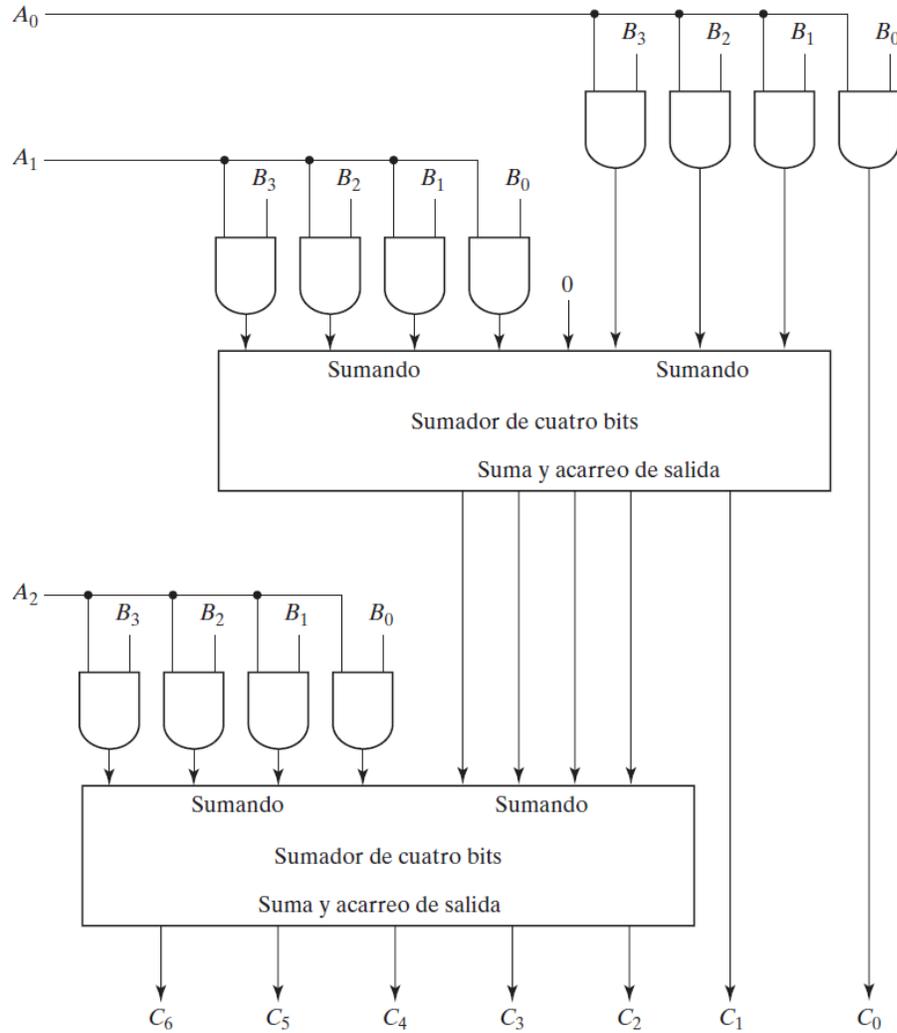
$$A = A_3A_2A_1A_0$$

$$B = B_3B_2B_1B_0$$

Donde  $A_i$  y  $B_i$  son cada dígito del número de cuatro bits. Para determinar si  $A = B$ , se necesita que  $A_i = B_i$  sean iguales. Esto se puede determinar con una compuerta XOR negada.

$$x_i = (A_i \oplus B_i)' = A_i B_i + A'_i B'_i$$

Figura 68. Diagrama del multiplicador binario con sumador completo



Fuente: MANO, Morris. *Diseño digital*. p 132.

Donde  $x_i = 1$  cuando ambos dígitos sean cero o uno. Ahora, esto sería por cada uno de los dígitos, y como se necesitan que se cumplan todas, las salidas  $x_i$  irán conectadas hacia una compuerta AND.

$$(A = B) = (A_3B_3 + A'_3B'_3)(A_2B_2 + A'_2B'_2)(A_1B_1 + A'_1B'_1)(A_0B_0 + A'_0B'_0)$$

Para determinar si A es mayor o menor que B se inspeccionará las magnitudes relativas de pares de dígitos significativos, comenzando de la posición más significativa. Si los dos dígitos son iguales se comparará el siguiente par de dígitos menos significativos. Esta comparación continúa hasta encontrar un par de dígitos distintos. Si el dígito correspondiente de A es uno y el de B es cero, se concluye que  $A > B$ . Si el dígito correspondiente de A es 0 y el de B es 1, se concluye que  $A < B$ . Por ende, se puede realizar el siguiente listado de condiciones para cada uno de los casos:

**Para  $A > B$**

$$A_3 = 1 \cdot B_3 = 0$$

$$(A_3 = 1 \cdot B_3 = 1) \cdot (A_2 = 1 \cdot B_2 = 0)$$

$$(A_3 = 1 \cdot B_3 = 1) \cdot (A_2 = 1 \cdot B_2 = 1) \cdot (A_1 = 1 \cdot B_1 = 0)$$

$$(A_3 = 1 \cdot B_3 = 1) \cdot (A_2 = 1 \cdot B_2 = 1) \cdot (A_1 = 1 \cdot B_1 = 1) \cdot (A_0 = 1 \cdot B_0 = 0)$$

**Para  $A < B$**

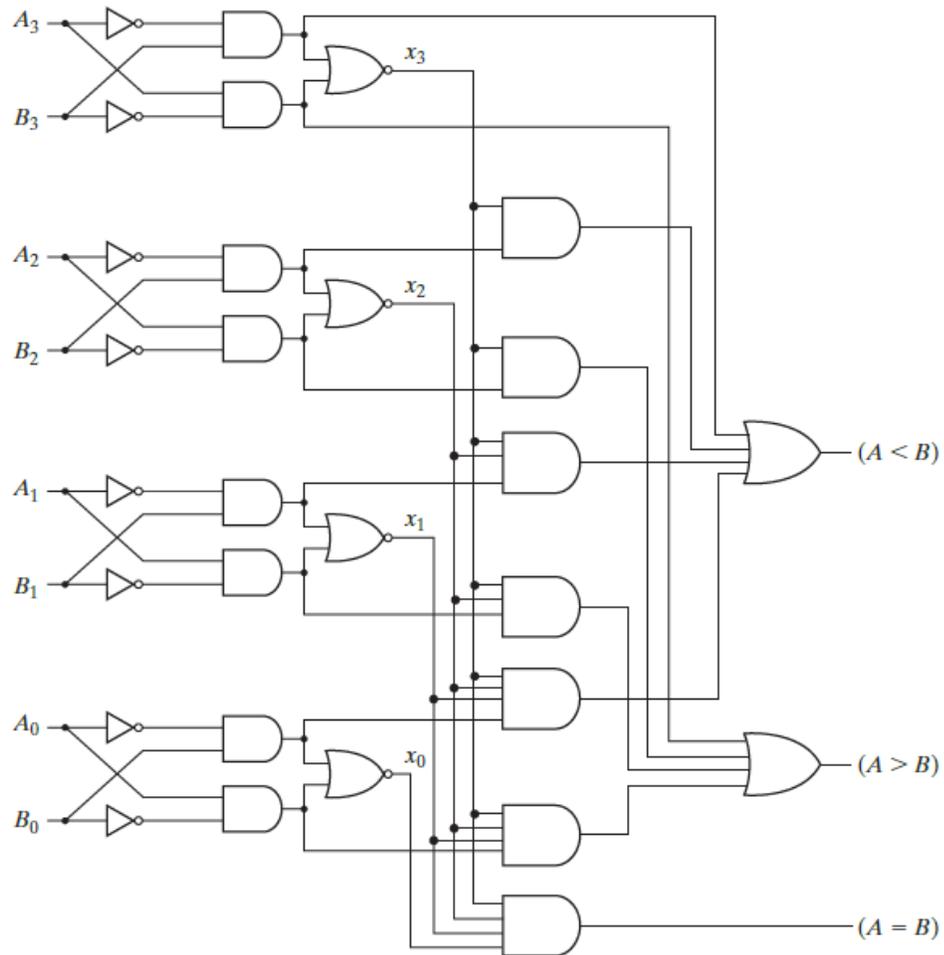
$$A_3 = 0 \cdot B_3 = 1$$

$$(A_3 = 1 \cdot B_3 = 1) \cdot (A_2 = 0 \cdot B_2 = 1)$$

$$(A_3 = 1 \cdot B_3 = 1) \cdot (A_2 = 1 \cdot B_2 = 1) \cdot (A_1 = 0 \cdot B_1 = 1)$$

$$(A_3 = 1 \cdot B_3 = 1) \cdot (A_2 = 1 \cdot B_2 = 1) \cdot (A_1 = 1 \cdot B_1 = 1) \cdot (A_0 = 0 \cdot B_0 = 1)$$

Figura 69. Comparador de magnitudes de cuatro bits



Fuente: MANO, Morris. *Diseño digital*. p 134.

Si alguna de estas condiciones es cumplida indica que A es mayor a B o bien, que A es menor que B, por ello se deduce que estas condiciones tienen la forma de suma de productos. En base a esto se deducen las funciones booleanas que se muestran en la figura 70. Donde  $x_i$  son las condiciones de igualdad de cada par de dígitos.

Por medio de las tres funciones obtenidas se obtiene el siguiente diagrama lógico.

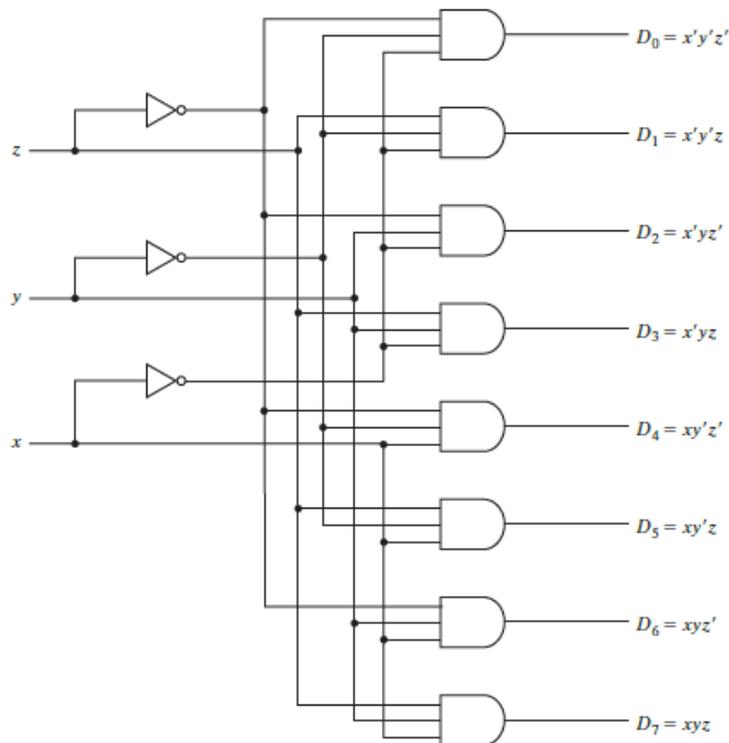
Figura 70. **Funciones booleanas de  $A > B$  y  $A < B$**

$$(A > B) = A_3B'_3 + x_3A_2B'_2 + x_3x_2A_1B'_1 + x_3x_2x_1A_0B'_0$$

$$(A < B) = A'_3B_3 + x_3A'_2B_2 + x_3x_2A'_1B_1 + x_3x_2x_1A'_0B_0$$

Fuente: MANO, Morris. *Diseño digital*. p 133.

Figura 71. **Decodificador de 3 a ocho líneas**



Fuente: MANO, Morris. *Diseño digital*. p 135.

#### 2.4.4.10. Decodificadores

Un decodificador es un circuito combinacional que convierte información binaria de  $n$  líneas de entrada a un máximo de  $2^n$  líneas de salida distintas. También se usa para referirse a otros convertidores de códigos, como un decodificador de BCD de siete segmentos. Es  $2^n$  el máximo de salidas que puede tener un decodificador, pero no necesariamente hay que usar todas las salidas, puede trabajarse con menos de  $2^n$ . En la tabla L se puede ver su tabla de verdad.

Tabla L. **Tabla de verdad de decodificador de 3 a 8 líneas**

Entradas			Salidas							
$x$	$y$	$z$	$D_0$	$D_1$	$D_2$	$D_3$	$D_4$	$D_5$	$D_6$	$D_7$
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

Fuente: MANO, Morris. *Diseño digital*. p 136.

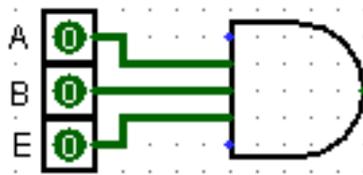
#### 2.4.4.11. Demultiplexores

Ahora se explicará el concepto de demultiplexor, pero antes de ello se explicará el concepto de entrada habilitadora dado a que es una característica intrínseca en este dispositivo.

#### 2.4.4.11.1. Entradas habilitadoras

Algunos de los decodificadores que se trabajan, contienen una entrada booleana llamada “habilitación” o “*enable*”. Esta entrada ayuda para habilitar las salidas del circuito combinacional que envíe información. Como se ve en el circuito siguiente, la multiplicación de las entradas A y B no se mostrará hasta que la entrada habilitadora E sea igual a 1.

Figura 72. Circuito con entrada habilitadora

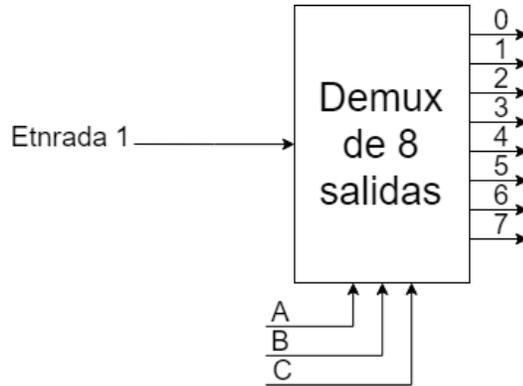


Fuente: elaboración propia, empleando Logisim.

#### 2.4.4.11.2. Demultiplexor

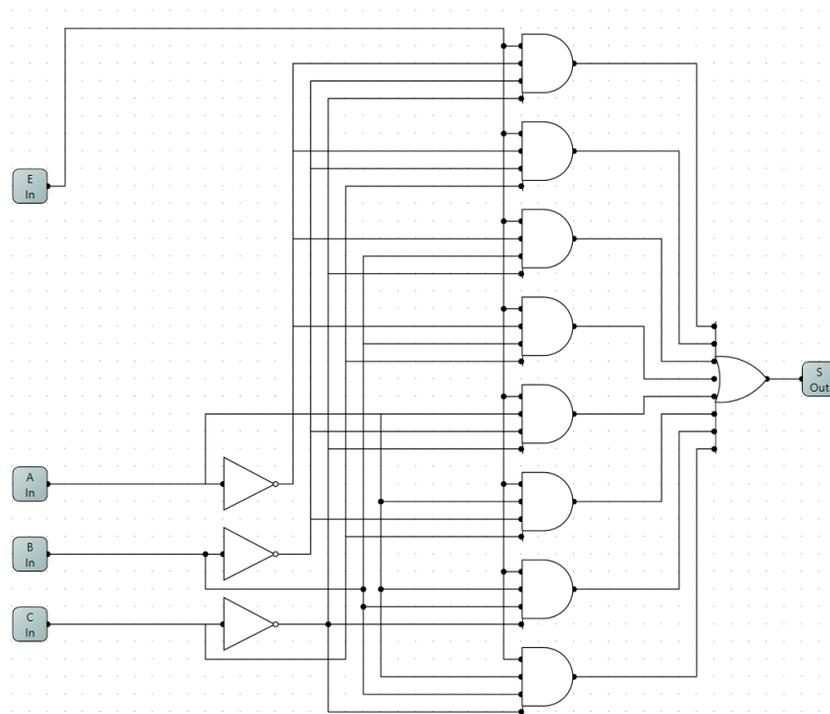
Es un circuito que recibe información de una sola línea y la dirige a una de  $2^n$  posibles líneas de salida, usando para la selección de la salida una cantidad de  $n$  líneas habilitadoras. Esto es parecido a un circuito decodificador, solo que con la diferencia que solo hay una línea de entrada de data y es controlado por una  $n$  cantidad de entradas habilitadoras hacia una de  $2^n$  salidas. El diagrama de bloques de un demultiplexor de 8 bits se puede ver en la figura 74, el circuito lógico en la figura 75 y la tabla de verdad en la tabla LI.

Figura 73. **Diagrama de bloques de demultiplexor de 8 bits**



Fuente: elaboración propia, empleando Dia.

Figura 74. **Diagrama lógico de demultiplexor de 8 bits**



Fuente: elaboración propia, empleando LogicCircuit.

### 2.4.4.11.3. Decodificador / demultiplexor

Dado a que se obtienen operaciones de decodificador y demultiplexor con el mismo circuito, se dice que un decodificador con entradas de habilitación es un decodificador/demultiplexor.

Tabla LI. **Tabla de demultiplexor de 8 bits**

Entrada	Entradas Habilitadoras			Salidas							
	A	B	C	0	1	2	3	4	5	6	7
E	0	0	0	E	0	0	0	0	0	0	0
E	0	0	1	0	E	0	0	0	0	0	0
E	0	1	0	0	0	E	0	0	0	0	0
E	0	1	1	0	0	0	E	0	0	0	0
E	1	0	0	0	0	0	0	E	0	0	0
E	1	0	1	0	0	0	0	0	E	0	0
E	1	1	0	0	0	0	0	0	0	E	0
E	1	1	1	0	0	0	0	0	0	0	E

Fuente: elaboración propia, empleando Excel.

### 2.4.4.12. Codificadores

Es un circuito digital que efectúa la operación inversa de la que efectúa un circuito decodificador. El codificador tiene  $2^n$  líneas de entrada y n líneas de salida.

Un ejemplo sencillo es el codificador de 8 bits, el cual tiene 8 entradas y una cantidad de 3 salidas. Esta muestra los números del cero al siete en el sistema binario. Por ello se tiene la limitante que solo una entrada puede estar activa en un momento dado. Por medio de la tabla de verdad (tabla LII) se pueden deducir las siguientes funciones booleanas:

$$z = D_1 + D_3 + D_5 + D_7$$

$$y = D_2 + D_3 + D_6 + D_7$$

$$x = D_4 + D_5 + D_6 + D_7$$

Tabla LII. **Tabla de verdad del codificador octal a binario**

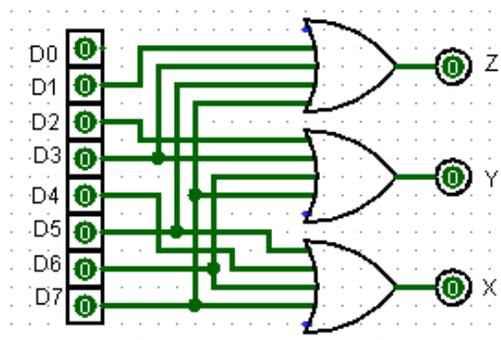
Entradas								Salidas		
$D_0$	$D_1$	$D_2$	$D_3$	$D_4$	$D_5$	$D_6$	$D_7$	$x$	$y$	$z$
1	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	1	1
0	0	0	0	1	0	0	0	1	0	0
0	0	0	0	0	1	0	0	1	0	1
0	0	0	0	0	0	1	0	1	1	0
0	0	0	0	0	0	0	1	1	1	1

Fuente: MANO, Morris. *Diseño digital*. p 139.

La limitante es que solo una entrada pueda ser activada a la vez. Ahora suponiendo que las entradas  $D_3$  y  $D_6$  son activadas al mismo tiempo. Esto daría como resultado que  $x = y = z = 1$  (111), lo cual solo debería suceder cuando  $D_7 = 1$ . Como se ve no representa a  $D_3$  ni a  $D_6$ .

Para evitar esta ambigüedad los circuitos deben contar con alguna prioridad de entrada, para que sea garantizado que solo se codifique una de las entradas. Otra ambigüedad es que se genera una salida de tres ceros cuando todas las entradas son cero. Esto solo debería de ocurrir cuando  $D_0 = 1$ .

Figura 75. Diagrama lógico del codificador de octal a binario



Fuente: elaboración propia, empleando Logisim.

Tabla LIII. Tabla de verdad de codificador de cuatro entradas con prioridad

Entradas				Salidas		
$D_0$	$D_1$	$D_2$	$D_3$	$x$	$y$	$V$
0	0	0	0	X	X	0
1	0	0	0	0	0	1
X	1	0	0	0	1	1
X	X	1	0	1	0	1
X	X	X	1	1	1	1

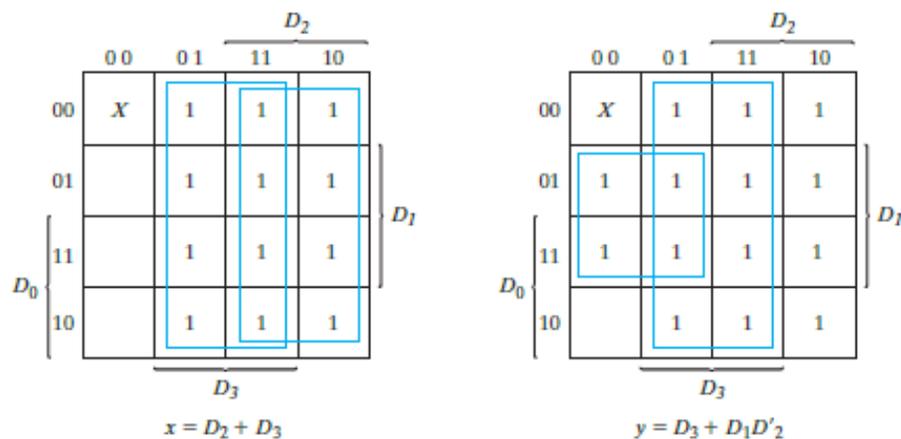
Fuente: MANO, Morris. *Diseño digital*. p 140.

### 2.4.4.12.1. Codificador con prioridad

Un codificador con prioridad es un circuito codificado que incluye la función de prioridad. Su funcionamiento consiste en que, si hay una cantidad mayor a una de entradas, en la salida mostrará solo la que tenga mayor prioridad.

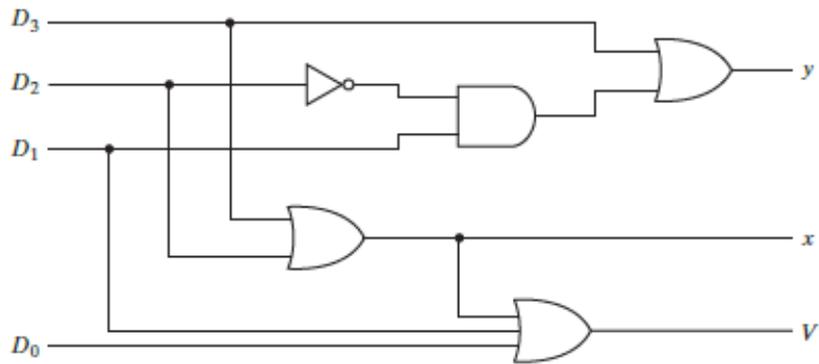
Para resolver los problemas del circuito codificador tradicional se implementaron nuevas condiciones como muestra la tabla de verdad del circuito (tabla LIII). Se integró una nueva salida booleana, la cual es una validación que será uno cuando uno o más de las entradas sean uno. Esto para evitar una lectura no deseada cuando todas las entradas sean cero. Luego, para que no haya ambigüedad en la salida, se realizó la tabla de verdad con condiciones de indiferencia, denotando que mientras mayor sea la posición de la salida tendrá una mayor prioridad. Si  $D_3 = 1$  no importará que los demás sean uno. Si  $D_3 = 0$ , el siguiente en verificarse será  $D_2$  y así sucesivamente.

Figura 76. Mapas del codificador con prioridad



Fuente: MANO, Morris. *Diseño digital*. p 140.

Figura 77. **Diagrama lógico de codificador con prioridad**



Fuente: MANO, Morris. *Diseño digital*. p 141.

$$x = D_2 + D_3$$

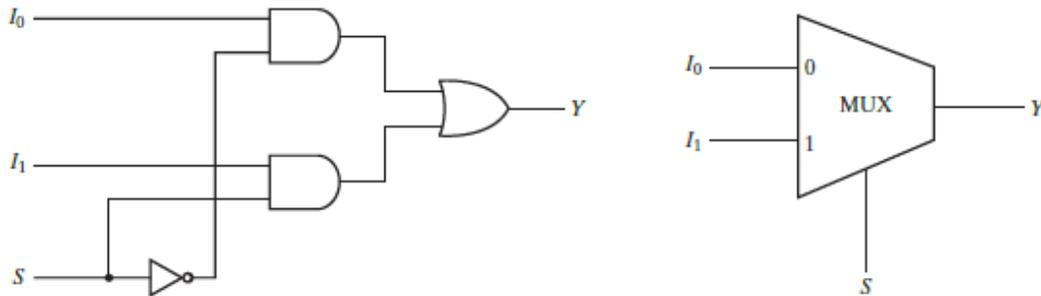
$$y = D_3 + D_1 D'_2$$

Hay que recordar que al evaluar con condiciones de indiferencia se puede elegir que las X sean unos o ceros, según la conveniencia del diseño del circuito. En la figura 78 se puede apreciar el circuito de un codificador con prioridad.

#### 2.4.4.13. Multiplexor

Un multiplexor es un circuito combinacional que selecciona información de una de muchas líneas de entrada y la envía a una sola línea de salida. La selección de una sola línea de entrada dada se controla con un conjunto de líneas de selección. Normalmente, hay  $2^n$  líneas de entrada con n líneas de selección cuyas combinaciones de bits determinan cuál entrada se selecciona.

Figura 78. **Diagrama lógico y diagrama de bloques de un multiplexor de 2 a 1**



Fuente: MANO, Morris. *Diseño digital*. p 142.

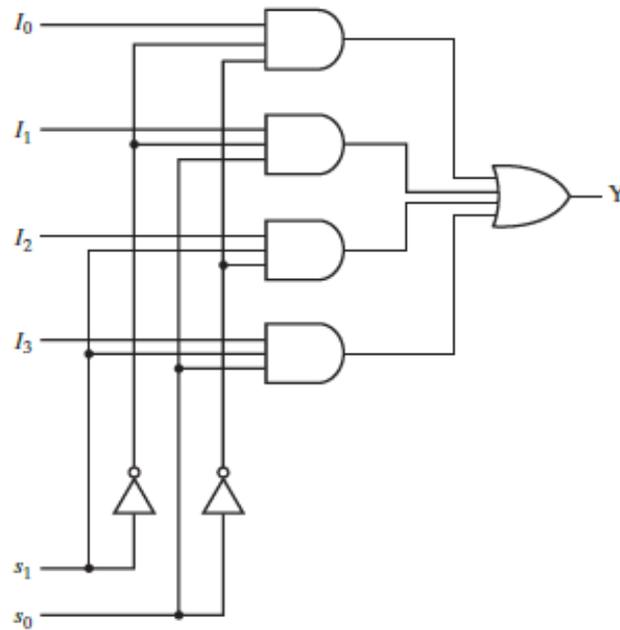
Cuando se hace referencia a un multiplexor, normalmente se le refiere diciendo “de 2 líneas a 1”, esto quiere decir que tendrá 2 entradas las cuales se pueden elegir por medio de un selector y una sola salida donde saldrá la entrada deseada. Cuando se tiene una cantidad de mayor de entradas, se tiene que tener una mayor cantidad de selectores; recordando que si son  $2^n$  entradas, será un selector de n bits.

Tabla LIV. **Tabla de verdad de multiplexor de 4 líneas a 1**

$s_1$	$s_0$	$Y$
0	0	$I_0$
0	1	$I_1$
1	0	$I_2$
1	1	$I_3$

Fuente: MANO, Morris. *Diseño digital*. p 142.

Figura 79. **Diagrama lógico de multiplexor de 4 líneas a 1**



Fuente: MANO, Morris. *Diseño digital*. p 142.

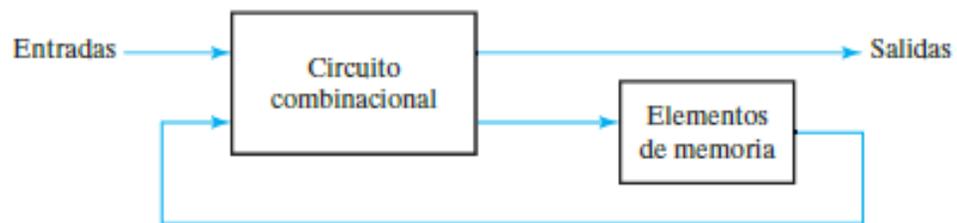
$$Y = I_0S_0'S_1' + I_1S_0'S_1 + I_2S_0'S_1 + I_3S_0S_1$$

En la tabla LIV se visualiza la tabla de verdad de un multiplexor de cuatro líneas a una, teniendo un selector de dos bits. Esta muestra el funcionamiento de la salida Y con respecto a los selectores  $x_0$  y  $x_1$ . En la figura 80 se observa el circuito lógico del multiplexor. Se ve que en el caso de los multiplexores no son bits, sino que son las salidas  $I_i$ .

## 2.5. Lógica secuencial

Hasta ahora solo se han estudiado circuitos que dependen solamente de sus entradas actuales. Pero ahora se estudiará circuitos donde se tendrán elementos de almacenamiento que guardarán estados pasados.

Figura 80. Diagrama de bloques de un circuito secuencial



Fuente: MANO, Morris. *Diseño digital*. p 167.

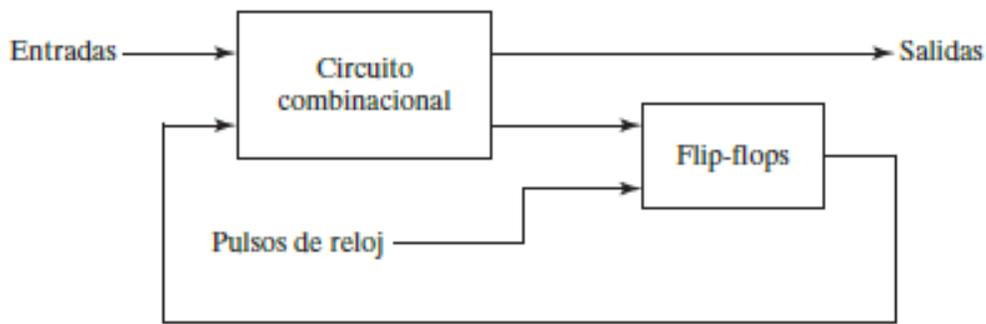
Un circuito secuencial consiste en un circuito combinacional al que se conecta elementos de almacenamiento para formar una trayectoria de retroalimentación. Los elementos de almacenamiento son dispositivos capaces de guardar información binaria. La información almacenada en estos elementos en cualquier momento dado define el estado del circuito secuencial en ese momento.

El circuito secuencial recibe información binaria de entradas externas. Esas entradas, junto con el estado actual de los elementos de almacenamiento, determinan el valor binario de las salidas. También determinan la condición para cambiar el estado de los elementos de almacenamiento.

### 2.5.1. Circuito secuencial síncrono

Es un sistema que su comportamiento se define conociendo sus señales en instantes discretos. Utiliza señales que afectan a los dispositivos de almacenamiento únicamente en instantes discretos. La sincronización se logra con un elemento de sincronización llamado generador de reloj, el cual produce un tren de pulsos de reloj. Un tren de pulsos son bits distribuidos en forma serial, los cuales alternan sus valores (101010101<sub>2</sub>). Estos cambios de valor de los bits del tren de pulso permiten cambiar el estado del dispositivo de almacenamiento.

Figura 81. Diagrama de bloques de circuito secuencial con reloj



a) Diagrama de bloques



b) Diagrama de temporización de los pulsos de reloj

Fuente: MANO, Morris. *Diseño digital*. p 168.

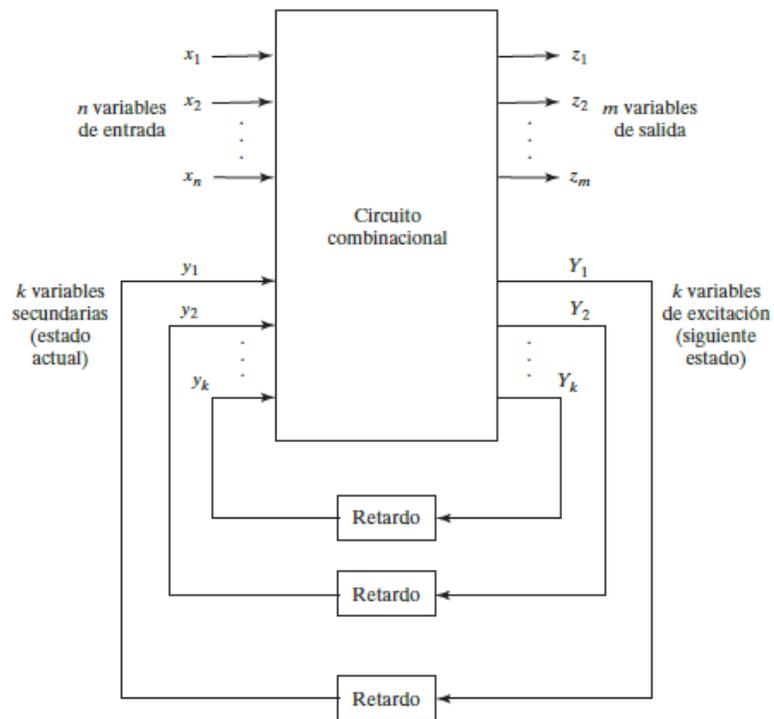
Un circuito síncrono que utiliza pulsos de reloj es llamado: circuito secuencial con reloj. Los dispositivos de almacenamiento que son empleados

en los circuitos secuenciales con reloj se llaman *flip-flop*, o *flip-flops* en plural. Este es un dispositivo que puede almacenar un bit de información.

### 2.5.2. Circuito secuencial asíncrono

El comportamiento de este circuito está definido para cualquier instante del tiempo dado. Es decir, no se liga a ningún tren de pulsos. Para que los dispositivos de almacenamiento realicen un cambio en su estado, es necesario que ocurra un cambio en las entradas del circuito secuencial.

Figura 82. Diagrama de bloques de circuito secuencial asíncrono



Fuente: MANO, Morris. *Diseño digital*. p 343.

Los dispositivos de almacenamiento que se utilizan para estos circuitos son *flip-flops* sin reloj, o bien circuitos de retardo. Los circuitos de retardo son dispositivos que generan un retardo de la señal del circuito. Esto porque, al ser un circuito con retroalimentación, es probable que existan retrasos con la señal en algunos dispositivos. Para asegurar que todos los dispositivos cuenten con el mismo retardo, se utilizan los dispositivos de retardo.

### **2.5.3. Flip-Flops**

Es un dispositivo con la capacidad de mantenerse en uno de dos estados posibles durante un tiempo indefinido. Hay una variedad de tipos de *flip-flops*, los cuales tienen una función específica. Las principales diferencias entre los diversos tipos de *flip-flops*, radican en el número de entradas que tienen y la forma en que las entradas afectan al estado binario.

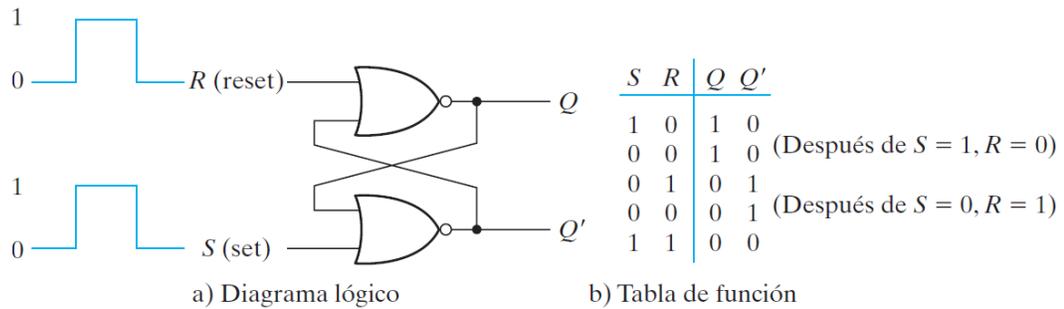
#### **2.5.3.1. Latches**

Los *latches* son los tipos más básicos de *flip-flops* y operan con niveles de señal. También son los circuitos más básicos con los cuales se construyen *flip-flops* más complejos. La diferencia principal con los *flip-flops* más complejos es que estos no cuentan con una entrada de reloj.

##### **2.5.3.1.1. Latch SR**

Este es un circuito con dos compuertas NOR o dos compuertas NAND acopladas en cruz. Cuentan con una entrada S (*set* de establecer), y una entrada R (*reset* de restablecer).

Figura 83. **Latch SR con compuertas NOR**



Fuente: MANO, Morris. Diseño Digital. p 169.

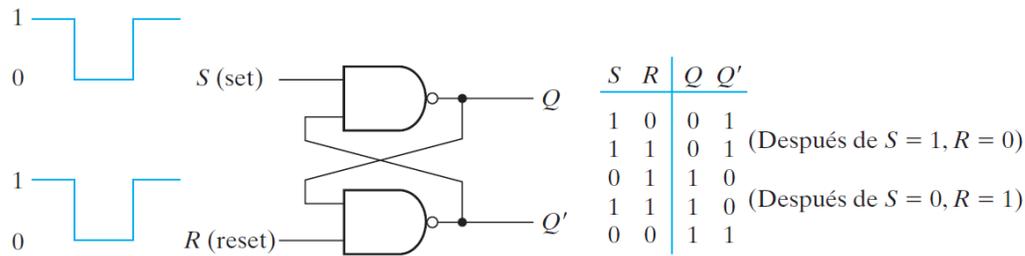
El *latch SR* hecho con compuertas NOR posee 3 estados; el estado establecido, que es cuando  $Q=1$  y  $Q'=0$ ; el estado restablecido, que es cuando  $Q'=1$  y  $Q=0$ ; y el estado indefinido, que es cuando  $Q$  y  $Q'$  son igual a cero. El *latch* hecho con compuertas NOR, opera con ambas entradas normalmente en cero. Al aplicar momentáneamente un uno en  $S$ , entrará al estado establecido. Para cambiar al estado restablecido hay que aplicar cero a la entrada  $S$  y uno en la entrada  $R$ .

Para retornar al estado establecido hay que aplicar cero en la entrada  $R$  y uno en la entrada  $S$ . Si ambas entradas son cero, el estado del *flip-flop* será el estado previo. Si ambas entradas son uno, el estado del *flip-flop* será indefinido. Para el estado indefinido, no importa en qué estado previo se encuentre el *flip-flop*. Es indefinido porque es imposible predecir cuál será el siguiente estado cuando ambas entradas vuelvan a cero.

El *latch SR* hecho con compuertas NAND (o *latch S'R'*) posee tres estados; el estado establecido, que es cuando  $Q=0$  y  $Q'=1$ ; el estado

restablecido, que es cuando  $Q'=0$  y  $Q=1$ ; y el estado indefinido, que es cuando  $Q$  y  $Q'$  son igual a uno.

Figura 84. **Latch SR con compuertas NAND**



a) Diagrama lógico

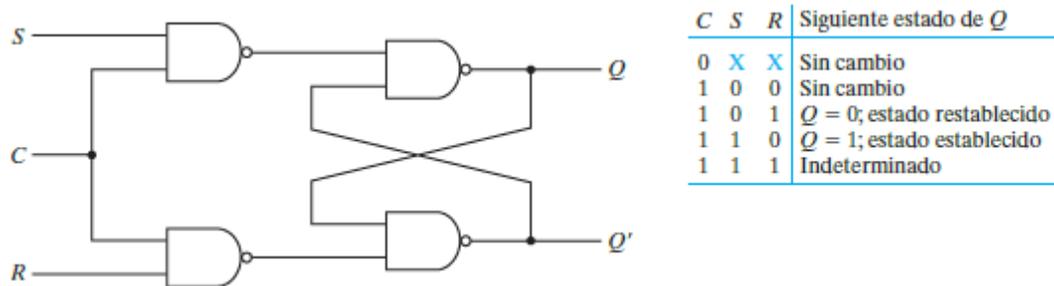
b) Tabla de función

Fuente: MANO, Morris. *Diseño digital*. p 170.

El *latch* hecho con compuertas NAND opera con ambas entradas normalmente en uno. Al aplicar momentáneamente un cero en  $S$ , entrará al estado establecido. Para cambiar al estado restablecido hay que aplicar uno a la entrada  $S$  y cero en la entrada  $R$ . Para retornar al estado establecido hay que aplicar uno en la entrada  $R$  y cero en la entrada  $S$ . Si ambas entradas son uno, el estado del *flip-flop* será el estado previo. Si ambas entradas son cero, el estado del *flip-flop* será indefinido. Para el estado indefinido, no importa en qué estado previo se encuentre el *flip-flop*. Es indefinido porque es imposible predecir cuál será el siguiente estado cuando ambas entradas vuelvan a cero.

A veces es necesario poder tener una entrada de control para cuando se desea cambiar de estado de los *latches*. Por este motivo se diseñó un *latch* que tenga una entrada habilitadora de control ( $C$ ), para controlar cuando se desea cambiar de estado al *latch*, o cuando se desee mantenerlo en el mismo estado.

Figura 85. **Latch S'R' con entrada de control**



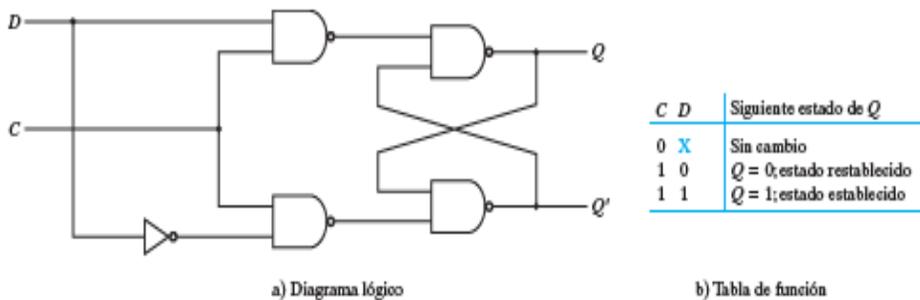
Fuente: MANO, Morris. *Diseño digital*. p 171.

Cuando la entrada de control se mantenga en cero, las salidas no tendrán ningún cambio. Cuando se torne a uno, será posible realizar un cambio en el estado del *latch*. Cambia al estado establecido cuando C=1, S=1 y R=0. Cambia al estado restablecido cuando C=1, S=0 y R=0. Se obtiene un estado indeterminado cuando C=1, S=1 y R=1. Cuando C=1, S=0 y R=0, el estado del *latch* permanece sin cambios.

### 2.5.3.1.2. Latch D

Para evitar la problemática del estado indefinido se ha diseñado el *latch* tipo D. Este tipo de *latch* solo tiene dos entradas: la entrada D (de datos) y la entrada C (de control). La entrada D pasa directamente a la S y su complemento a la entrada R. Ahora no importa lo que pase, la única manera de cambiar el estado del *flip-flop* es habilitando la entrada C. Cambia al estado establecido cuando C=1 y D=1. Cuando C=1 y D=0, cambia al estado restablecido. Cuando C=0, el estado permanece sin cambios.

Figura 86. **Latch D**

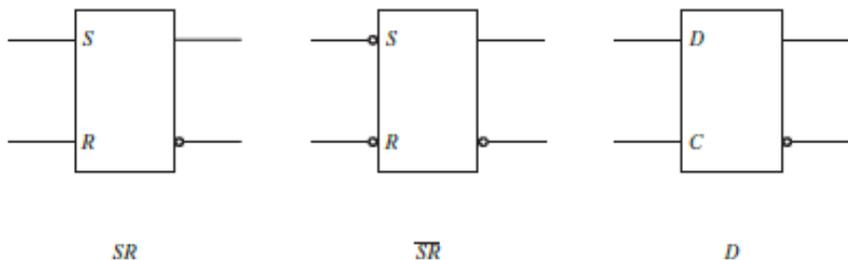


Fuente: MANO, Morris. *Diseño digital*. p 171.

### 2.5.3.1.3. Símbolos gráficos de latches

Los *latches* se representan por medio de rectángulos verticales que poseen las entradas al lado izquierdo y las salidas en el lado derecho. Las salidas Q son las que no poseen la salida invertida y las salidas Q' son las que poseen las salidas invertidas.

Figura 87. **Símbolos gráficos de latches SR, S'R' y D**

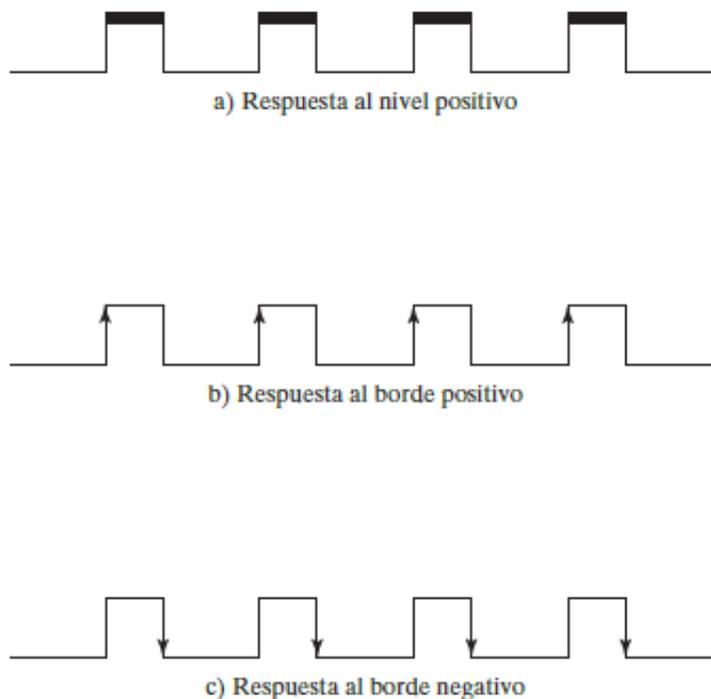


Fuente: MANO, Morris. *Diseño digital*. p 172.

### 2.5.3.2. *Flip-flops*

Hasta ahora se ha estudiado el comportamiento de los *latches*, los cuales pueden almacenar un bit de información y cambiar de estado, cuando su entrada de control lo permite. Mientras tanto la entrada se mantenga en ese nivel, cualquier cambio en la entrada de datos hará que cambie la salida y el estado del *latch*. Al recordar el diagrama de bloques de un circuito secuencial, se puede vislumbrar que posee una trayectoria de retroalimentación. Esto quiere decir que habrá salidas de *latches* conectadas, por medio de un circuito combinacional, a las entradas de otros *latches*.

Figura 88. **Tipos de respuestas de reloj en un *latch* y un *flip-flop***



Fuente: MANO, Morris. *Diseño digital*. p 173.

Ahora, se supone que se tiene un circuito secuencial con *latches* con un tren de pulsos de reloj aplicado en la entrada de control. Las transiciones de los estados de los *latches* se inician siempre y cuando el reloj es un uno lógico. El nuevo estado del *latch* aparece en la salida mientras el pulso aún está activo. Estas salidas se conectan a las entradas de los *latches* por medio de un circuito combinacional. Si las entradas aplicadas a los *latches* cambian mientras el pulso de reloj todavía está en uno lógico, los *latches* responderán a nuevos valores y podría presentarse un nuevo estado de salida. El resultado es una situación impredecible, ya que el estado de los *latches* podría seguir cambiando durante todo el tiempo que el pulso de reloj se mantiene en el estado activo.

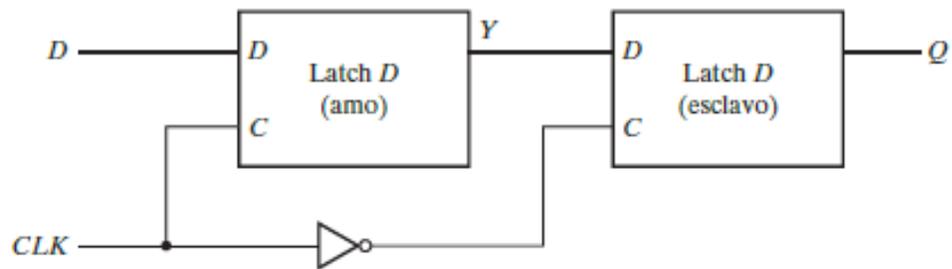
Este problema se ha eliminado permitiendo que el *flip-flop* pueda cambiar de estado en un tiempo de reloj menos prolongado. Esto será cuando ocurra una transición de señal. Una transición de señal es el cambio del valor de la señal. Puede ser una transición positiva (de cero a uno), que es llamada al borde o flanco positiva; o una transición negativa (de uno a cero), que es llamada al borde o flanco negativo. Al permitir que el circuito secuencial solo cambie en una transición de señal, se soluciona el problema anterior.

#### **2.5.3.2.1. *Flip-flop D* disparado por flanco**

Este tipo de *flip-flop* posee la característica que solo se podrá hacer un cambio de estado cuando ocurra una transición. Está construido por medio de 2 *latches*; un amo y un esclavo. Este se dispara por medio de una transición negativa. Cuando el reloj (CLK) es cero, el *latch* amo está inhabilitado y el *latch* esclavo muestra en su salida Q, el valor de su entrada D. Cuando CLK=1, el *latch* amo está habilitado y puede dejar pasar a su salida Y, el valor de su entrada D.

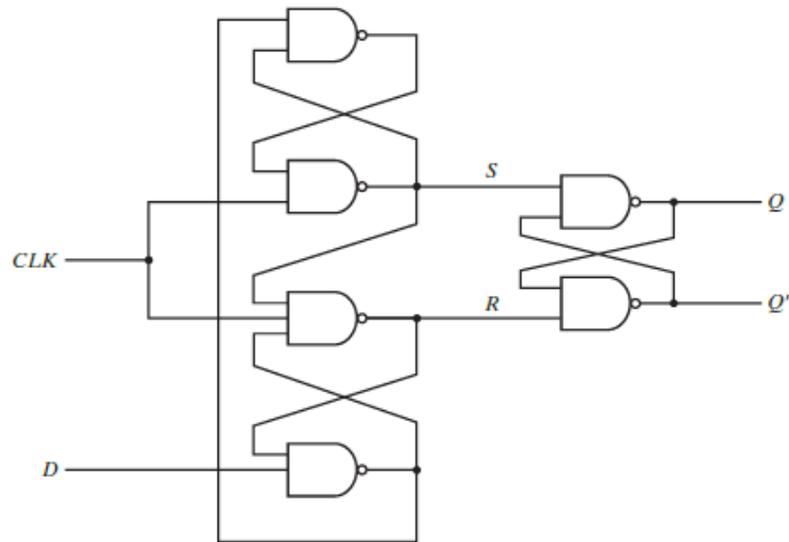
Ahora, como la salida  $Y$  es la entrada  $D$  del *latch* esclavo, mostrará el resultado justo al momento que  $CLK$  vuelva a tener el valor de cero.

Figura 89. **Flip-flop  $D$  con dos *latches*  $D$**



Fuente: MANO, Morris. *Diseño digital*. p 174.

Figura 90. **Flip-flop  $D$  disparado por borde positivo**

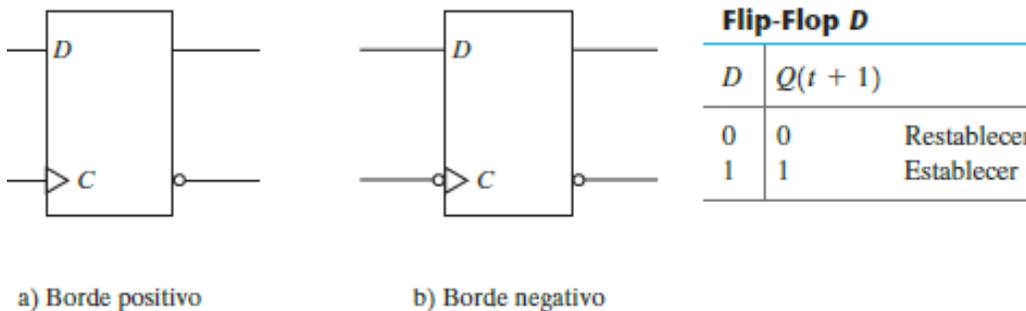


Fuente: MANO, Morris. *Diseño digital*. p 174.

Cuando el reloj es cero, la salida del inversor es uno. El *latch* esclavo queda habilitado y su salida Q es igual a la salida Y del amo. El *latch* amo queda inhabilitado cuando CLK=0. Cuando CLK=1, el dato de la entrada D externa se transfiere al amo. Cuando el pulso vuelva a cero, el amo quedará inhabilitado y aislado de la entrada D. Al mismo tiempo, el esclavo estará habilitado y el valor Y se transferirá a la salida Q del *flip-flop*. El *flip-flop D* mostrado anteriormente tiene respuesta al flanco positivo, pero agregando una compuerta NOT en la entrada del reloj, puede convertirlo en un *flip-flop* con respuesta al borde positivo.

Una construcción más eficiente del *flip-flop D* de disparo por borde consta de tres *latches SR*. Cuando el reloj de entrada del *flip-flop* disparado por borde realiza una transición positiva, el valor de D se transfiere a Q. Una transición negativa de uno a cero no afecta la salida y tampoco lo hace cuando CLK está estable en cero o uno lógico. Por tanto, este tipo de *flip-flop* solo se dispara en una transición positiva.

Figura 91. **Símbolo gráfico y tabla característica de *flip-flop D***



Fuente: MANO, Morris. *Diseño digital*. p 175.



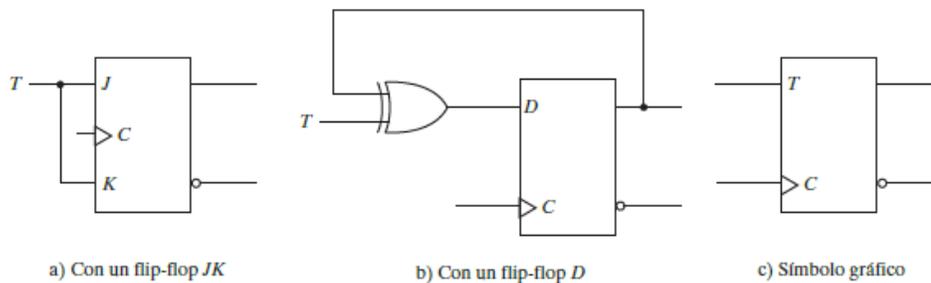
$$D = JQ' + K'Q$$

Figura 93. **Tabla de funcionamiento de *flip-flop JK***

<b>Flip-Flop JK</b>			
<i>J</i>	<i>K</i>	$Q(t + 1)$	
0	0	$Q(t)$	Sin cambio
0	1	0	Restablecer
1	0	1	Establecer
1	1	$Q'(t)$	Complementar

Fuente: MANO, Morris. *Diseño digital*. p 177.

Figura 94. **Circuito equivalente con *flip-flop D* y *JK* y simbología de *flip-flop T***



Fuente: MANO, Morris. *Diseño digital*. p 177.

### 2.5.3.2.3. Flip-Flop T

Este *flip-flop* también es construido con un *flip-flop D* disparado por borde. Este es un *flip-flop* complementador y también puede construirse con un *flip-flop JK*. Su ecuación característica es la siguiente:

$$D = T \oplus Q = TQ' + T'Q$$

Figura 95. Tabla de funcionamiento de *flip-flop T*

Flip-Flop T		
T	Q(t + 1)	
0	Q(t)	Sin cambio
1	Q'(t)	Complementar

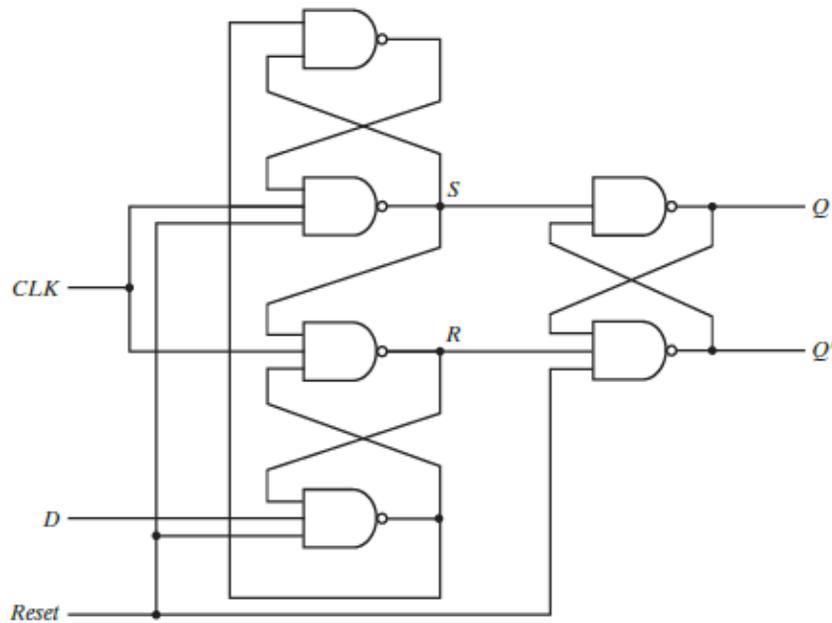
Fuente: MANO, Morris. *Diseño digital*. p 177.

### 2.5.4. Entradas asíncronas

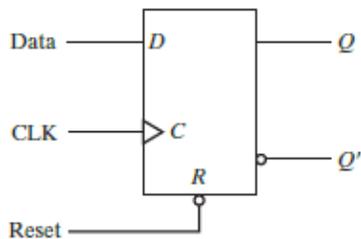
Algunos *flip-flop* tienen entradas directas asíncronas que sirven para forzar al *flip-flop* a un estado independiente del reloj. La entrada que pone el *flip-flop* en uno se llama establecimiento directo (*preset*). La entrada que pone a cero el *flip-flop* se llama restablecimiento directo (*clear*). Cuando se enciende un sistema digital, se desconoce el estado de los *flip-flops*. Las entradas directas sirven para poner todos los *flip-flops* del sistema en un estado inicial conocido antes del funcionamiento del reloj.

Como el ejemplo presentado en la figura 97 es de restablecimiento asíncrono con entrada invertida, se ve que se activa con un cero lógico. Cuando  $Reset=0$ , no se toman en cuenta las entradas del control ni de datos.

Figura 96. **Flip-Flop D con restablecimiento asíncrono**



a) Diagrama de circuito



b) Símbolo gráfico

$R$	$C$	$D$	$Q$	$Q'$
0	X	X	0	1
1	↑	0	0	1
1	↑	1	1	0

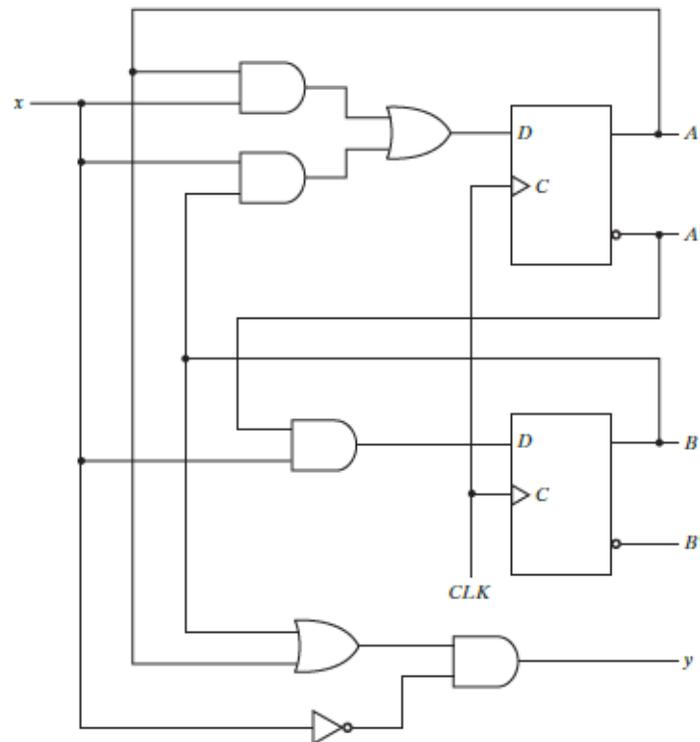
c) Tabla de función

Fuente: MANO, Morris. *Diseño digital*. p 177.

### 2.5.5. Análisis de circuitos secuenciales

Ahora se analizará cómo representar circuitos secuenciales por medio de ecuaciones utilizando el álgebra de Boole, tablas representado el estado del circuito y diagramas gráficos visualmente más amigables. El comportamiento de cualquier circuito secuencial lo determinan sus entradas, salidas y estado de sus *flip-flops*. El siguiente estado está determinado por sus entradas y estado actual de sus *flip-flops*. Los *flip-flop* a analizar pueden ser de cualquier tipo y el diagrama lógico podría tener o no, un circuito combinacional.

Figura 97. **Circuito secuencial**



Fuente: MANO, Morris. *Diseño digital*. p 181.

### 2.5.5.1. Ecuaciones de estado

También llamadas ecuación de transición, es una expresión algebraica que especifica el siguiente estado de un *flip-flop*. Esto en función de sus entradas y estados actuales de los *flip-flops*. En la figura 98 se puede apreciar un circuito secuencial con dos *flip-flops*. Se llamará al que se encuentra en la parte superior A y al otro B. Por ser *flip-flops* tipo D, su estado siguiente está denotado por su entrada D (datos). La parte izquierda de la ecuación denotará el estado siguiente del *flip-flop*, un borde de reloj después. El lado derecho de la ecuación será una expresión booleana que especifica el estado actual y las condiciones de entrada que harán que el siguiente estado sea uno.

$$A(t + 1) = A(t)x(t) + B(t)x(t)$$

$$B(t + 1) = A'(t)x(t)$$

En donde (t+1) significa un estado siguiente y (t) significa el estado actual. Sabiendo que son ecuaciones de estado y que del lado derecho son los estados actuales, se pueden hacer las ecuaciones un poco más compactas:

$$A(t + 1) = Ax + Bx$$

$$B(t + 1) = A'x$$

Las ecuaciones mostradas anteriormente sirven para obtener la salida de los *flip-flops* que conforman los circuitos analizados, sin embargo, estos no son la salida del circuito. La salida y, sí lo es. La salida estará en función de los estados actuales, y por ello sería de la siguiente manera:

$$y(t) = [A(t) + B(t)]x'(t)$$

Sabiendo que todo está en función de estados actuales, se puede hacer más compacta:

$$y = (A + B)x'$$

### 2.5.5.2. Tabla de estados

También llamada tabla de transición, es una tabla compacta donde se muestra los posibles estados siguientes en función de las entradas y los estados actuales. Consta de cuatro secciones: estado actual, entrada, siguiente estado y salida.

Figura 98. Tabla de estados

Estado actual		Entrada <i>x</i>	Siguiete estado		Salida <i>y</i>
<i>A</i>	<i>B</i>		<i>A</i>	<i>B</i>	
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	0	1
0	1	1	1	1	0
1	0	0	0	0	1
1	0	1	1	0	0
1	1	0	0	0	1
1	1	1	1	0	0

Fuente: MANO, Morris. *Diseño digital*. p 182.

Para formarla hay que deducir todas las posibles combinaciones con las entradas y los estados actuales de los *flip-flops*. Para ejemplificar esto se utilizan las siguientes ecuaciones de estado:

$$A(t + 1) = Ax + Bx$$

$$B(t + 1) = A'x$$

$$y = Ax' + Bx'$$

Se puede apreciar una tabla de estados en la figura 99. Al ser dos *flip-flops* y una sola entrada, se tiene ocho posibles combinaciones (de 000 a 111). En general, un circuito secuencial con  $m$  *flip-flops* y  $n$  entradas necesita  $2^{m+n}$  filas en la tabla de estados. Hay otra forma de representar la tabla de estados, en la que consiste en omitir una columna específica para las entradas (figura 100). Para cada estado actual hay dos siguientes estados y salidas posibles, dependiendo del valor de las entradas. Elegir el formato no implica cambios en los resultados, pero para ciertas aplicaciones es conveniente utilizar alguno de los formatos.

Figura 99. **Segunda forma de la tabla de estados**

Estado actual	Siguiete estado		Salida	
	$x = 0$	$x = 1$	$x = 0$	$x = 1$
<i>AB</i>	<i>AB</i>	<i>AB</i>	<i>y</i>	<i>y</i>
00	00	01	0	0
01	00	11	1	0
10	00	10	1	0
11	00	10	1	0

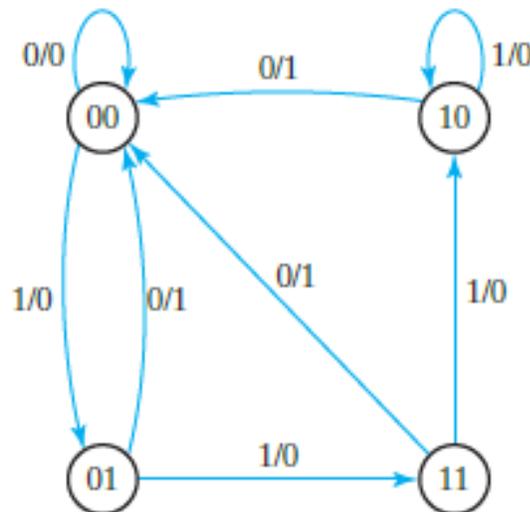
Fuente: MANO, Morris. *Diseño digital*. p 182.

### 2.5.5.3. Diagrama de estados

La información contenida en una tabla de estados se puede representar gráficamente en forma de diagrama de estados. En este tipo de diagramas, un estado se representa con un círculo, y las transiciones entre estados se indican con flechas que conectan a los círculos. Los números binarios dentro de cada círculo identifican el estado de los *flip-flops*.

Las flechas se rotulan con dos números separados por una diagonal. Primero se da el valor de entrada durante el estado actual, y el número después de la diagonal indica la salida durante el estado actual con esa entrada. Por ejemplo, la flecha del estado 00 a 01 lleva el rótulo 1/0, lo que significa que cuando el circuito secuencial está en el estado actual 00 y la entrada es 1, la salida es 0. Después del siguiente ciclo de reloj, el circuito pasa al siguiente estado, 01.

Figura 100. Diagrama de estados



Fuente: MANO, Morris. *Diseño digital*. p 183.

#### 2.5.5.4. Análisis con *flip-flops*

La parte del circuito combinacional que genera salidas externas se describe algebraicamente con un conjunto de funciones booleanas llamadas ecuaciones de salida. La parte del circuito que genera las entradas a los *flip-flops* se describe algebraicamente con un conjunto de funciones booleanas llamadas ecuaciones de entrada de *flip-flops*. Se adoptará la nomenclatura de usar el símbolo de entrada del *flip-flop* para denotar la variable de ecuación de entrada y un subíndice para indicar el nombre de la salida del *flip-flop*. Por ejemplo, hay que ver la siguiente ecuación:

$$D_Q = x + y$$

Esto quiere decir que el *flip-flop D* tiene a  $x + y$  de entradas y una salida Q. Si no dice que hay una salida adicional, se deduce que la salida del circuito secuencial es Q. Para referirse a una ecuación de entrada (en este caso al circuito combinacional que se encuentra en la entrada D) se utiliza la siguiente nomenclatura:

$$D_A = Ax + Bx$$

$$D_B = A'x$$

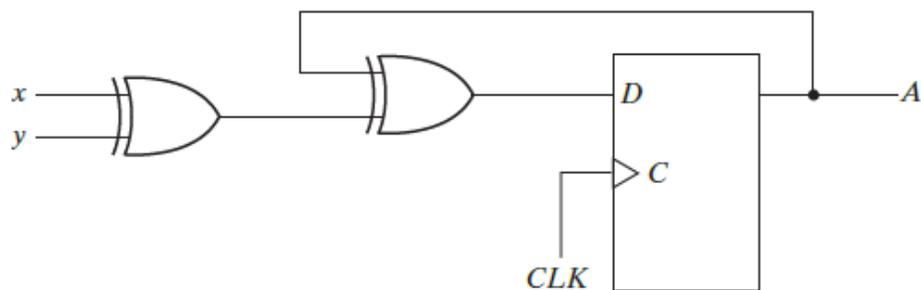
En este caso se tiene dos *flip-flops* (A y B) en el circuito.

### 2.5.5.4.1. Análisis con *flip-flop D*

Para aprender a cómo analizar al *flip-flop D*, hay que ver la figura 102, la cual es un circuito secuencial con un circuito combinacional de entrada. La ecuación de entrada es:

$$D_A = A \oplus x \oplus y$$

Figura 101. Circuito secuencial con *flip-flop D*



Fuente: MANO, Morris. *Diseño digital*. p 185.

Las variables  $x$  y  $y$  son las entradas del circuito y  $A$  es la salida del *flip-flop*. En este caso no se especifica una salida, por lo que se asume que la salida del circuito secuencial es  $A$ . Al recordar que la entrada  $D$  es la que indicará el valor del siguiente estado, se deduce la siguiente:

$$A(t + 1) = A \oplus x \oplus y$$

Con ello se puede deducir la tabla de estados y el diagrama de estados.

Tabla LV. **Tabla de estados**

Estado actual	Salidas		Siguiente estado
<i>A</i>	<i>x</i>	<i>y</i>	<i>A</i>
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

Fuente: MANO, Morris. *Diseño digital*. p 185.

Figura 102. **Diagrama de estados**



Fuente: MANO, Morris. *Diseño digital*. p 185.

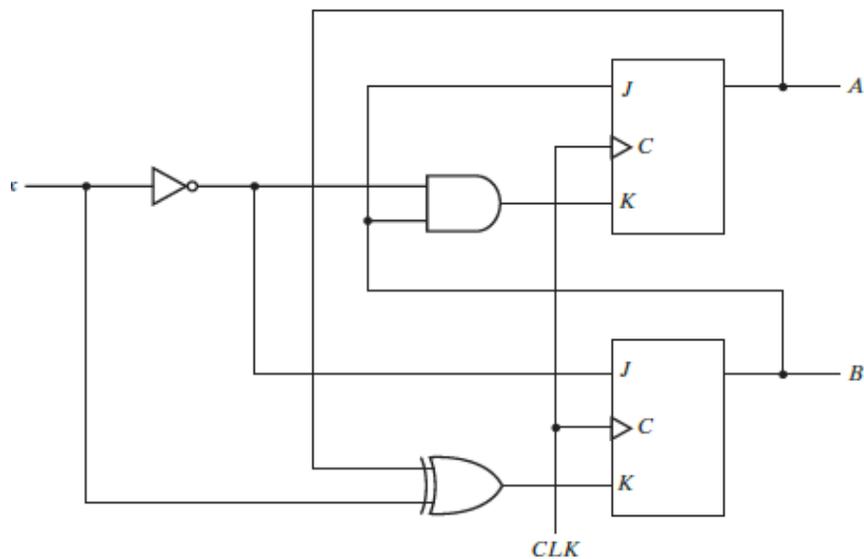
#### 2.5.5.4.2. Análisis con *flip-flop JK*

Cuando se utiliza un *flip-flop JK* es necesario consultar la tabla característica o la ecuación característica para conocer el valor del estado siguiente. Esto para obtener la tabla de estados.

Para ello se puede partir por dos caminos: utilizando la tabla característica o las ecuaciones de estado. A continuación, se hará utilizando la tabla característica y posteriormente con la ecuación de estados. Los pasos son los siguientes:

- Determinar las ecuaciones de entrada del flip-flop en términos del estado actual y las variables de entrada.
- Enumerar los valores binarios de cada ecuación de entrada.
- Usar la tabla característica del flip-flop en cuestión para determinar los valores del siguiente estado de la tabla de estados.

Figura 103. **Circuito secuencial con *flip-flop JK***



Fuente: MANO, Morris. *Diseño digital*. p 186.

Para ejemplificar esto se analizará el circuito de la figura 104. Como se ve es un circuito secuencial que tiene dos *flip-flops JK*. Las salidas del circuito son las salidas de los *flip-flops*. Ahora se prosigue a tener el primer paso el cual es de obtener las ecuaciones de entrada de los *flip-flop*. Estas son:

$$J_A = B$$

$$J_B = x'$$

$$K_A = Bx'$$

$$K_B = A \oplus x$$

Teniendo las ecuaciones de entrada, se prosigue a encontrar los valores binarios de entrada; el cual es el segundo paso.

Tabla LVI. **Entradas de *flip-flop***

<b>Entradas de flip-flop</b>			
$J_A$	$K_A$	$J_B$	$K_B$
0	0	1	0
0	0	0	1
1	1	1	0
1	0	0	1
0	0	1	1
0	0	0	0
1	1	1	1
1	0	0	0

Fuente: MANO, Morris. *Diseño digital*. p 186.

Existen dos maneras para obtener los estados siguientes, verificando la tabla característica, o las ecuaciones de estado. Cuando se utiliza la tabla característica hay que recordar que cuando  $J=1$  y  $K=0$ , el siguiente estado es uno. Cuando  $J=0$  y  $K=1$ , el siguiente estado es 0. Cuando  $J=K=0$ , no hay cambio en el estado y el valor siguiente es igual al estado actual. Cuando  $J=K=1$ , el bit del siguiente estado es el complemento del bit del estado actual.

Tabla LVII. **Tabla de estados del circuito secuencial con *flip-flop JK***

Estado Actual		Entrada	Siguiete estado		Entradas de flip-flop			
A	B		A	B	$J_A$	$K_A$	$J_B$	$K_B$
0	0	0	0	1	0	0	1	0
0	0	1	0	0	0	0	0	1
0	1	0	1	1	1	1	1	0
0	1	1	1	0	1	0	0	1
1	0	0	1	1	0	0	1	1
1	0	1	1	0	0	0	0	0
1	1	0	0	0	1	1	1	1
1	1	1	1	1	1	0	0	0

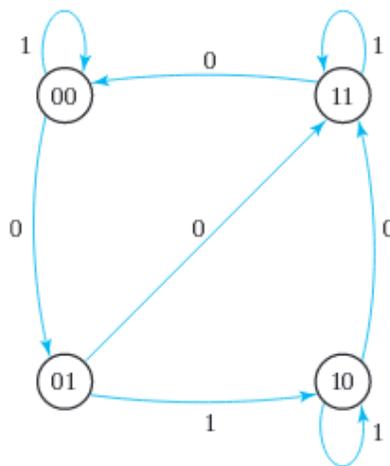
Fuente: MANO, Morris. *Diseño digital*. p 186.

Cuando se desea utilizar las ecuaciones características se pueden seguir los siguientes pasos:

- Obtener las ecuaciones de entrada de *flip-flop* en términos del estado actual y las variables de entrada.

- Sustituir las ecuaciones de salida en la ecuación característica del *flip-flop* para obtener ecuaciones de estado.
- Usar las ecuaciones de estado correspondientes para determinar los valores del siguiente estado de la tabla de estados.

Figura 104. **Diagrama de estados de circuito con *flip-flop JK***



Fuente: MANO, Morris. *Diseño digital*. p 187.

Las ecuaciones de entrada fueron obtenidas para el caso anterior. Para el segundo paso se prosigue a obtener las ecuaciones de estado

$$A(t + 1) = JA' + K'A$$

$$B(t + 1) = JB' + K'B$$

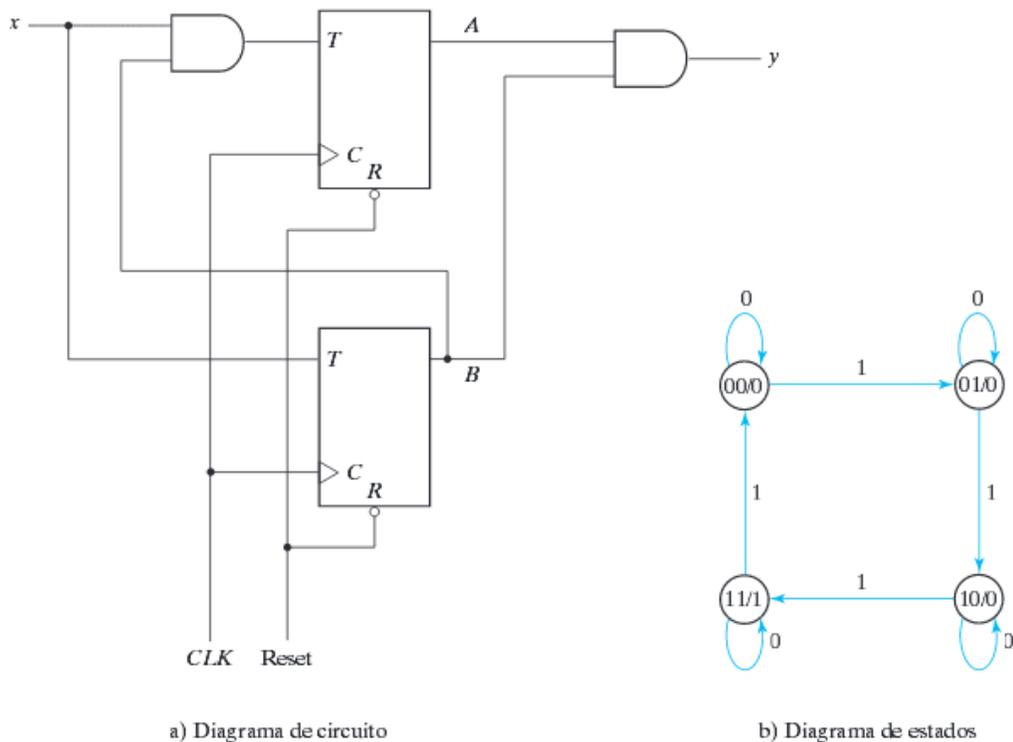
Ahora sustituyendo los valores de  $J_A$ ,  $K_A$ ,  $J_B$  y  $K_B$  se obtiene:

$$A(t + 1) = BA' + (Bx')'A = A'B + (B' + x)A = A'B + AB' + Ax$$

$$B(t + 1) = x'B' + (A \oplus x)'B = B'x' + BAx + BA'x'$$

Ya teniendo las ecuaciones de estado se puede deducir el estado siguiente del *flip-flops*. Ya sea que se obtenga el siguiente estado de los *flip-flops* por medio de una tabla de estados o ecuaciones de estado, es recomendable realizar un diagrama de estados; esto para saber el siguiente estado de una manera más rápida y, por ende, hacer el trabajo más eficientemente.

Figura 105. **Circuito secuencial con *flip-flops T***



Fuente: MANO, Morris. *Diseño digital*. p 188.

### 2.5.5.4.3. Análisis con *flip-flop T*

El análisis de un circuito secuencial con *flip-flops T* sigue siendo el mismo procedimiento que se delineó para los *flip-flops JK*. Considerar el circuito secuencial de la figura 106. Tiene dos *flip-flops T*, A y B, una entrada x y una salida y. Se puede conocer el siguiente estado del *flip-flop* (y por ende el de la salida) con la ayuda de las ecuaciones características o de la tabla característica.

Si se utiliza la tabla característica del *flip-flop T*, recordar que cuando  $T=0$   $Q(t+1)=Q(t)$  y cuando  $T=1$   $Q(t+1)=Q'(t)$ . Con esta información se puede deducir la tabla de estados de la tabla LVIII.

Tabla LVIII. **Tabla de estados de circuito con *flip-flop T***

Estado actual		Entrada x	Siguiete estado		Salida y
A	B		A	B	
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	1	0
0	1	1	1	0	0
1	0	0	1	0	0
1	0	1	1	1	0
1	1	0	1	1	1
1	1	1	0	0	1

Fuente: MANO, Morris. *Diseño digital*. p 189.

Se ve que en este caso la salida del circuito es la salida y. La ecuación característica del *flip-flop T* es:

$$Q(t + 1) = T'Q + TQ' = T \oplus Q$$

Y hay que reemplazar con las entradas del circuito:

$$T_A = Bx$$

$$T_B = x$$

$$y = AB$$

Reemplazando en cada ecuación de cada *flip-flop*:

$$A(t + 1) = (Bx)'A + (Bx)A' = AB' + Ax' + A'Bx$$

$$B(t + 1) = x \oplus B$$

Con esto ya se puede deducir un diagrama de estados y visualizar de una mejor manera el circuito.

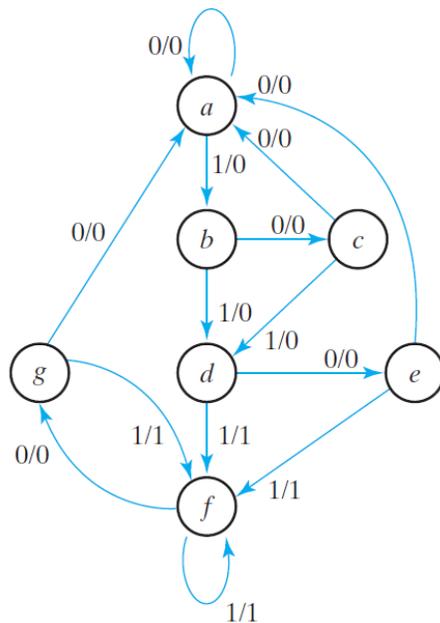
### 2.5.6. Máquinas de estado

El modelo más general de un circuito secuencial tiene entradas, salidas y estados internos. Se acostumbra distinguir entre dos modelos de circuitos secuenciales: el modelo Mealy y el modelo Moore. Difieren en la forma en que se genera la salida. En el modelo Mealy, la salida es función tanto del estado actual como de la entrada. En el modelo Moore, la salida solo es función del estado actual. Al tratar los dos modelos, algunos libros y otras fuentes técnicas ven el circuito secuencial como una máquina de estados finitos (*FSM*). El modelo Mealy de un circuito secuencial es una *FSM Mealy* o máquina Mealy. El modelo Moore es una *FSM Moore* o máquina Moore.

### 2.5.7. Reducción de estados

La reducción en el número de *flip-flops* de un circuito secuencial se le conoce como reducción de estados. El algoritmo de reducción es un procedimiento que puede reducir la cantidad de estados en una tabla de estados, pero no afecta en las salidas del circuito. Un efecto impredecible al reducir el número de *flip-flops* es que a veces el circuito equivalente podría requerir más compuertas combinatorias. El procedimiento consiste en encontrar dos estados en la tabla de estados que, con las mismas entradas, den como resultado la misma salida y que lleven al mismo estado.

Figura 106. **Ejemplo de diagrama de estados**



Fuente: MANO, Morris. *Diseño digital*. p 199.

Para comprender de una mejor manera esto se hará el siguiente ejemplo. Se partirá de un circuito secuencial cuyo diagrama de estados es el de la figura 107. Se puede ver que los estados están representados con letras y no con números binarios, dado a que los estados internos solo sirven para producir las sucesiones requeridas.

Sabiendo que hay un número infinito de sucesiones de entrada que se pueden ingresar al circuito, se ingresa la sucesión 01010110100 partiendo del estado inicial a. Cada entrada de cero o uno produce una salida de cero o uno y hace que el circuito pase al siguiente estado. Esto dará una idea del funcionamiento del circuito. Esto se puede visualizar en la figura 108.

Figura 107. **Secuencia de entrada**

Estado	<i>a</i>	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>f</i>	<i>g</i>	<i>f</i>	<i>g</i>	<i>a</i>
Entrada	0	1	0	1	0	1	1	0	1	0	0	
Salida	0	0	0	0	0	1	1	0	1	0	0	

Fuente: MANO, Morris. *Diseño digital*. p 199.

Con los datos obtenidos tras la sucesión e inspeccionando el diagrama de estados, se puede proceder a realizar la tabla de estados de la figura 109. Suponga que se ha hallado un circuito secuencial cuyo diagrama de estados tiene menos de siete estados y se desea compararlo con el circuito cuya tabla de estados es la anterior. Si se les aplica sucesiones de entrada idénticas y los dos circuitos producen salidas idénticas para todas las sucesiones de entrada, se dice que los dos circuitos son equivalentes, y se puede sustituir uno con el otro.

Figura 108. **Tabla de estados con siete estados**

<b>Estado actual</b>	<b>Siguiente estado</b>		<b>Salida</b>	
	<b><math>x = 0</math></b>	<b><math>x = 1</math></b>	<b><math>x = 0</math></b>	<b><math>x = 1</math></b>
<i>a</i>	<i>a</i>	<i>b</i>	0	0
<i>b</i>	<i>c</i>	<i>d</i>	0	0
<i>c</i>	<i>a</i>	<i>d</i>	0	0
<i>d</i>	<i>e</i>	<i>f</i>	0	1
<i>e</i>	<i>a</i>	<i>f</i>	0	1
<i>f</i>	<i>g</i>	<i>f</i>	0	1
<i>g</i>	<i>a</i>	<i>f</i>	0	1

Fuente: MANO, Morris. *Diseño digital*. p 200.

Tabla LIX. **Tabla de estados con seis estados**

<b>Estado actual</b>	<b>Siguiente estado</b>		<b>Salida</b>	
	<b><math>x = 0</math></b>	<b><math>x = 1</math></b>	<b><math>x = 0</math></b>	<b><math>x = 1</math></b>
<i>a</i>	<i>a</i>	<i>b</i>	0	0
<i>b</i>	<i>c</i>	<i>d</i>	0	0
<i>c</i>	<i>a</i>	<i>d</i>	0	0
<i>d</i>	<i>e</i>	<i>f</i>	0	1
<i>e</i>	<i>a</i>	<i>f</i>	0	1
<i>f</i>	<i>e</i>	<i>f</i>	0	1

Fuente: MANO, Morris. *Diseño digital*. p 200.

Ahora se procede a buscar estados equivalentes en la tabla de estados. Se dice que dos estados son equivalentes si, para cada miembro del conjunto de entradas, dan la misma salida y pasan el circuito al mismo estado o a un estado equivalente.

Se puede visualizar que los estados e y g cumplen con esta característica y por ello se dice que son equivalentes y pueden ser reducidos a un solo estado. Ahora se tiene una tabla de estados con solo seis estados. El estado se ha tomado lugar del estado g. Por ello, se debe reemplazar cualquier estado g por el estado e. Al hacer esto, se ve que se tiene dos estados equivalentes, los estados d y f. Se puede realizar de nuevo otra reducción. Se comprueba que con la misma secuencia de estados que los resultados de las salidas y las transiciones entre estados son las mismas. Con esto se comprueba que los circuitos son equivalentes.

Figura 109. **Tabla de estados con cinco estados**

Estado actual	Siguiete estado		Salida	
	$x = 0$	$x = 1$	$x = 0$	$x = 1$
<i>a</i>	<i>a</i>	<i>b</i>	0	0
<i>b</i>	<i>c</i>	<i>d</i>	0	0
<i>c</i>	<i>a</i>	<i>d</i>	0	0
<i>d</i>	<i>e</i>	<i>d</i>	0	1
<i>e</i>	<i>a</i>	<i>d</i>	0	1

Fuente: MANO, Morris. *Diseño digital*. p 201.

Figura 110. **Secuencia de entrada**

Estado	<i>a</i>	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>d</i>	<i>d</i>	<i>e</i>	<i>d</i>	<i>e</i>	<i>a</i>
Entrada	0	1	0	1	0	1	1	0	1	0	0	
Salida	0	0	0	0	0	1	1	0	1	0	0	

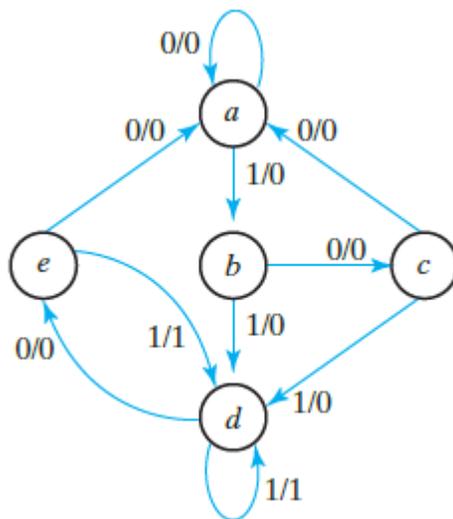
Fuente: MANO, Morris. *Diseño digital*. p 201.

Como se ve se pasó de una tabla de estados con siete estados, a una tabla de estados con cinco estados. No obstante, el hecho de que en una tabla de estados se haya reducido menos estados no garantiza un ahorro en el número de *flip-flops* o compuertas.

### 2.5.8. Asignación de estados

Al momento de diseñar un circuito secuencial con componentes físicos, es necesario asignar valores binarios codificados a los estados. En el caso de un circuito de  $m$  estados los circuitos deben de contener  $n$  bits, donde  $2^n \geq m$ . Hay una variedad de codificaciones en las cuales se puede elegir, y con respecto a la codificación elegida, será el tamaño del circuito combinacional. Entre las más comunes se tiene la asignación binaria, el código gray y un solo uno. Esta última utiliza una cantidad de bits igual a la cantidad de estados.

Figura 111. Diagrama de estados con cinco estados



Fuente: MANO, Morris. *Diseño digital*. p 201.

Figura 112. Posibles asignaciones binarias de estados

Estado	Asignación 1 Binaria	Asignación 2 Código Gray	Asignación 3 Un solo uno
<i>a</i>	000	000	00001
<i>b</i>	001	001	00010
<i>c</i>	010	011	00100
<i>d</i>	011	010	01000
<i>e</i>	100	110	10000

Fuente: MANO, Morris. *Diseño digital*. p 202.

### 2.5.8.1. Diseño de circuitos secuenciales

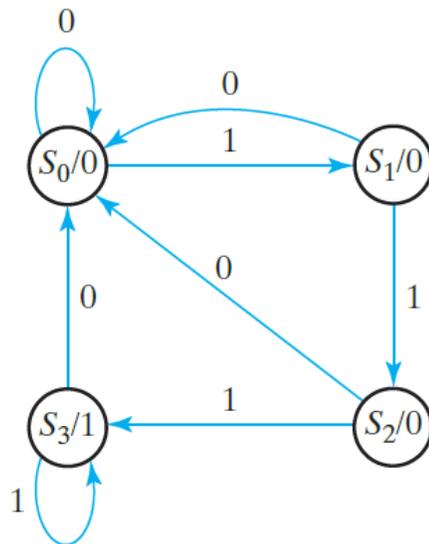
El análisis de circuitos secuenciales parte de un diagrama de circuitos y culmina en una tabla o diagrama de estados. El diseño de un circuito secuencial parte de un conjunto de especificaciones y culmina en un diagrama lógico. En contraste con los circuitos combinacionales, que se especifican cabalmente con una tabla de verdad, los circuitos secuenciales requieren una tabla de estados para su especificación.

Su diseño parte de un conjunto de especificaciones y culmina en un diagrama lógico o una lista de funciones booleanas de la cual puede obtenerse un diagrama lógico. Este se compone de *flip-flops* y circuitos combinacionales. El procedimiento para diseñar circuitos secuenciales sincrónicos es:

- Deducir, de la descripción textual y las especificaciones del funcionamiento deseado, un diagrama de estados para el circuito.
- Reducir el número de estados si es necesario.
- Asignar valores binarios a los estados.

- Obtener la tabla de estados codificados en binario.
- Escoger el tipo de *flip-flops* que se usarán
- Deducir las ecuaciones simplificadas de entrada y de salida de los *flip-flops*.
- Dibujar el diagrama lógico.

Figura 113. **Diagrama de estados para el detector de sucesiones**



Fuente: MANO, Morris. *Diseño digital*. p 204.

Para ejemplificar esto se diseñará un circuito que detecte tres o más unos consecutivos en una cadena de bits que lleguen por una línea de entrada. En este caso se tiene cuatro estados: cero unos recibidos ( $S_0$ ), un uno recibido ( $S_1$ ), dos unos consecutivos recibidos ( $S_2$ ) y tres o más unos recibidos ( $S_3$ ). Al recibir un cero en cualquiera de los estados, el siguiente estado será  $S_0$ . Se tiene una sola entrada y una sola salida. La salida será igual a uno solo si hay una sucesión de tres de más unos.

Con estas condiciones aclaradas se puede realizar el diagrama de estados que se muestra en la figura 114. Posterior a esto, se debe asignar

códigos binarios a los estados y hacer una tabla de estados. Se escogió la asignación binaria pero la codificación no influye en el resultado del circuito. La tabla de estados se muestra en la tabla LX.

Tabla LX. **Tabla de estados del detector de sucesiones**

Estado actual		Entrada <i>x</i>	Sigulente Estado		Salida <i>y</i>
<i>A</i>	<i>B</i>		<i>A</i>	<i>B</i>	
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	0	0
0	1	1	1	0	0
1	0	0	0	0	0
1	0	1	1	1	0
1	1	0	0	0	1
1	1	1	1	1	1

Fuente: MANO, Morris. *Diseño digital*. p 205.

### 2.5.8.1.1. Síntesis con *flip-flops D*

Se escogieron dos *flip-flops D* para representar los estados de la tabla de estados anterior. Hay una entrada *x* y una salida *y*. Los minitérminos de interés son:

$$A(t + 1) = D_A(A, B, x) = \sum (3,5,7) = A'Bx + AB'x + ABx$$

$$B(t + 1) = D_B(A, B, x) = \sum (1,5,7) = A'B'x + AB'x + ABx$$

$$y(A, B, x) = \sum (6,7) = ABx' + ABx$$

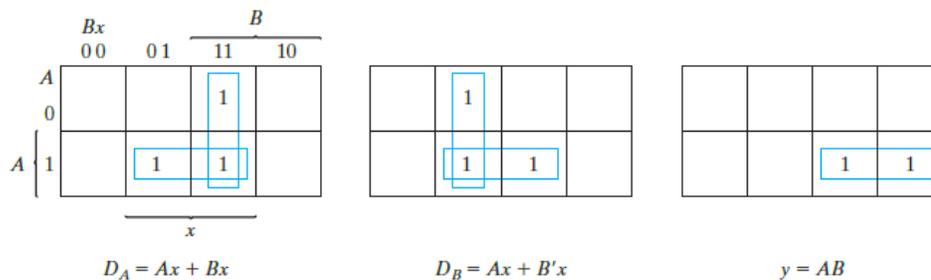
Ahora con la ayuda de mapas de Karnaugh (figura 114), se puede reducir las expresiones booleanas.

$$D_A = Ax + Bx$$

$$D_B = Ax + B'x$$

$$y = AB$$

Figura 114. Mapas para el detector de sucesiones



Fuente: MANO, Morris. *Diseño digital*. p 205.

Con las expresiones reducidas se puede realizar el diagrama lógico que se muestra en la figura 115.

### 2.5.8.1.2. Tablas de excitación

El diseño de un circuito secuencial con *flip-flops* de otro tipo que no sea D se complica con el hecho de que las ecuaciones de entrada del circuito se deben deducir de manera indirecta de la tabla de estados. Para determinar las ecuaciones de entrada de estos *flip-flops* es necesario deducir una relación funcional entre la tabla de estados y las ecuaciones de entrada. Durante el proceso de diseño, normalmente se conoce la transición de estado actual a siguiente estado y se desea conocer las condiciones de entrada del *flip-flop* que



### 2.5.8.1.3. Síntesis con *flip-flop JK*

El procedimiento de síntesis de circuitos secuenciales con *flip-flops JK* es el mismo que con *flip-flops D*, excepto que la ecuación de entrada se debe evaluar a partir de la transición de estado actual al siguiente estado deducida de la tabla de excitación.

Tabla LXII. **Tabla de estados y entradas de circuito con *flip-flop JK***

Estado Actual		Entrada <i>x</i>	Siguiete Estado		Entradas del <i>flip-flop</i>			
<i>A</i>	<i>B</i>		<i>A</i>	<i>B</i>	<i>J<sub>A</sub></i>	<i>K<sub>A</sub></i>	<i>J<sub>B</sub></i>	<i>K<sub>B</sub></i>
0	0	0	0	0	X	0	X	
0	0	1	0	1	0	X	1	X
0	1	0	1	0	1	X	X	1
0	1	1	0	1	0	X	X	0
1	0	0	1	0	X	0	0	X
1	0	1	1	1	X	0	1	X
1	1	0	1	1	X	0	X	0
1	1	1	0	0	X	1	X	1

Fuente: MANO, Morris. *Diseño digital*. p 208.

Además de tener columnas para el estado actual, la entrada y el siguiente estado, como en una tabla convencional de estados, la tabla también muestra las condiciones de entrada del *flip-flop* de las que se deducen en las ecuaciones de entrada. Con esto se deduce los siguientes minitérminos de interés, los cuales están en función a las entradas y los estados actuales de los *flip-flop*:

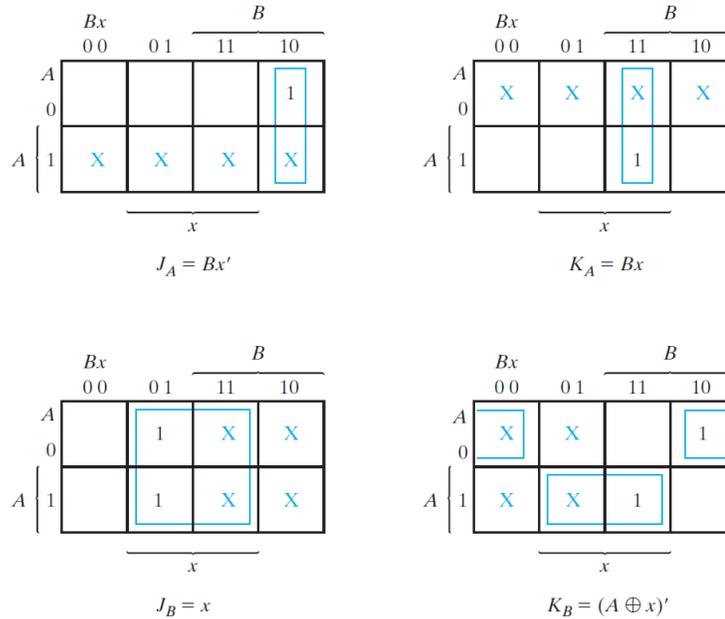
$$J_A(A, B, x) = \sum (2) = A'Bx'$$

$$K_A(A, B, x) = \sum (7) = ABx$$

$$J_B(A, B, x) = \sum (1) = A'B'x$$

$$K_B(A, B, x) = \sum (2,7) = A'Bx' + ABx$$

Figura 116. Mapas para las ecuaciones de entrada J y K



Fuente: MANO, Morris. *Diseño digital*. p 208.

Ahora se formarán los minterminos en base de las entradas de los *flip-flops*. A diferencia del *flip-flop D* que era con respecto al estado siguiente; siempre en función de las entradas y estados actuales. Por medio de la tabla se pueden formar minterminos de interés. Con la ayuda de los mapas de Karnaugh se puede simplificar ecuaciones y obtener un circuito mucho más simplificado.

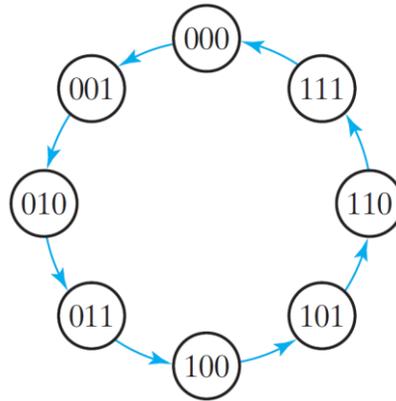
$$J_A = A'Bx'$$

$$K_A = ABx$$

$$J_B = A'B'x + AB'x$$



Figura 118. **Diagrama de estados de contador de tres bits**



Fuente: MANO, Morris. *Diseño digital*. p 209.

Como se ve en este diagrama no se cuenta con ninguna entrada externa, pero el cambio está dado por las transiciones reloj del circuito. El siguiente estado de un contador depende exclusivamente de un estado actual, y la transición de estado se efectúa cada vez que el reloj tiene una transición. En la tabla LXIV se vislumbra la tabla de estados del contador de tres bits.

Tabla LXIII. **Tabla de estados del contador de tres bits**

Estado actual			Siguiete estado			Entradas de los flip-flops		
$A_2$	$A_1$	$A_0$	$A_2$	$A_1$	$A_0$	$T_{A2}$	$T_{A1}$	$T_{A0}$
0	0	0	0	0	1	0	0	1
0	0	1	0	1	0	0	1	1
0	1	0	0	1	1	0	0	1
0	1	1	1	0	0	1	1	1
1	0	0	1	0	1	0	0	1
1	0	1	1	1	0	0	1	1
1	1	0	1	1	1	0	1	1
1	1	1	0	0	0	1	1	1

Fuente: MANO, Morris. *Diseño digital*. p 210.

Aquí se toma en cuenta las entradas de los *flip-flops* en la tabla de estados. En base a las entradas de los *flip-flops* se obtienen los minitérminos de interés.

$$T_{A_2}(A, B, x) = \sum (3,7) = A'_2A_1A_0 + A_2A_1A_0$$

$$T_{A_1}(A, B, x) = \sum (1,3,5,6,7) = A'_2A'_1A_0 + A'_2A_1A_0 + A_2A'_1A_0 + A_2A_1A'_0 + A_2A_1A_0$$

$$\begin{aligned} T_{A_0}(A, B, x) &= \sum (0,1,2,3,4,5,6,7) \\ &= A'_2A'_1A'_0 + A'_2A'_1A_0 + A'_2A_1A'_0 + A'_2A_1A_0 + A_2A'_1A'_0 + A_2A'_1A_0 \\ &\quad + A_2A_1A'_0 + A_2A_1A_0 \end{aligned}$$

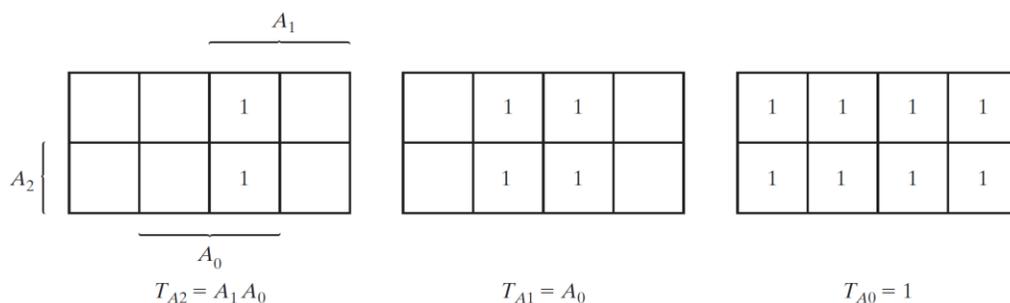
Con la ayuda de mapas de Karnaugh se reducen las expresiones booleanas. Esto se ve en la figura 120. Con ello se obtienen las siguientes expresiones booleanas:

$$T_{A_2} = A_1A_0$$

$$T_{A_1} = A_0$$

$$T_{A_0} = 1$$

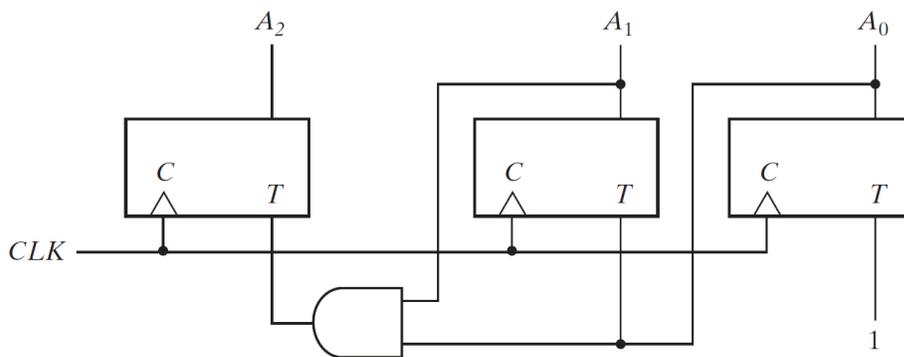
Figura 119. **Mapa de Karnaugh de contador de 3 bits**



Fuente: MANO, Morris. *Diseño digital*. p 210.

Con las expresiones booleanas se puede hacer el diagrama del circuito lógico del contador, el cual se muestra en la figura 121.

Figura 120. **Diagrama lógico de contador binario de tres bits**



Fuente: MANO, Morris. *Diseño digital*. p 211.

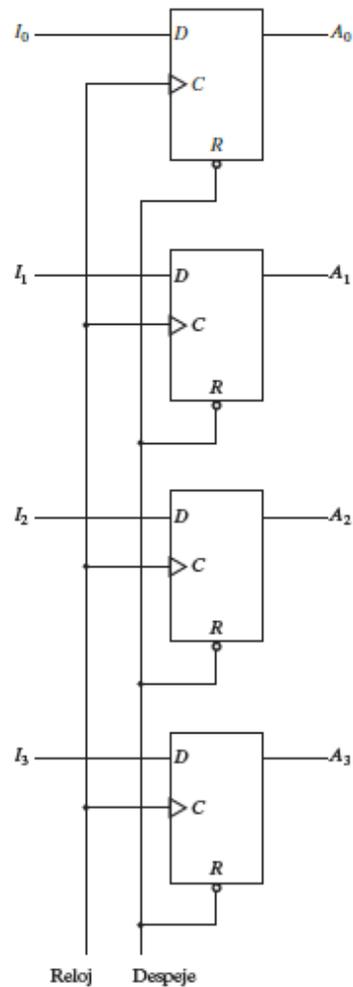
## 2.6. Registros y contadores

Los circuitos que incluyen *flip-flops* por lo regular se clasifican según la función que desempeñan, más que por el nombre del circuito secuencial. Dos de esas clasificaciones son los registros y contadores.

### 2.6.1. Registros

Un registro consiste en un grupo de *flip-flops* y compuertas que efectúan una transición. Los *flip-flops* contiene la información binaria y las compuertas determinan como se transfiere la información al registro.

Figura 121. **Registro de 4 bits en paralelo**



Fuente: MANO, Morris. *Diseño digital*. p 218.

### 2.6.1.1. **Registros con carga paralela**

Si todos los bits de un registro se cargan simultáneamente, con un pulso de reloj común se dice que la carga se efectúa en paralelo. Esto quiere decir que ocurre una transferencia de registro al mismo tiempo con una misma señal. Como se ve en la figura 121, es un registro con carga paralela de 4 bits

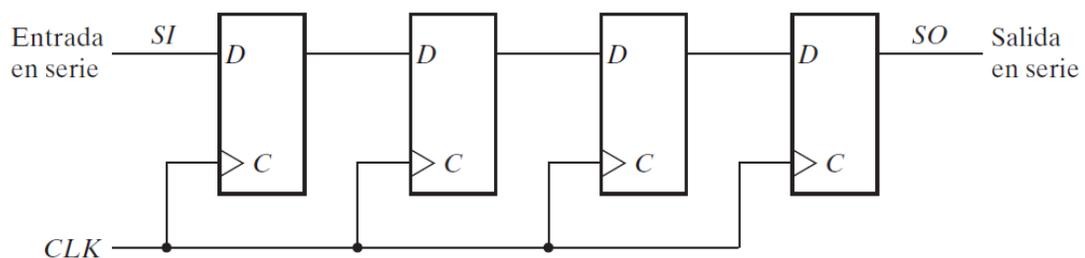
utilizando *flip-flops D*. Todas las entradas *C* de los *flip-flops* están conectadas al mismo reloj. En la entrada *D* se puede o no tener un circuito combinacional, esto dependerá de lo que se desee hacer.

### 2.6.1.2. Registros de desplazamiento

Un registro capaz de desplazar su información binaria en una dirección o en la otra se llama registro de desplazamiento. La configuración lógica de un registro de desplazamiento consiste en una cadena de *flip-flops*, con la salida de un *flip-flop* conectada a la entrada del siguiente *flip-flop*. Todos tienen el mismo reloj, que activan el desplazamiento de la etapa siguiente.

La figura anterior muestra el ejemplo más sencillo de un registro de desplazamiento (sin ningún circuito combinacional). Este se puede ver en la figura 122.

Figura 122. Registro de desplazamiento de cuatro bits



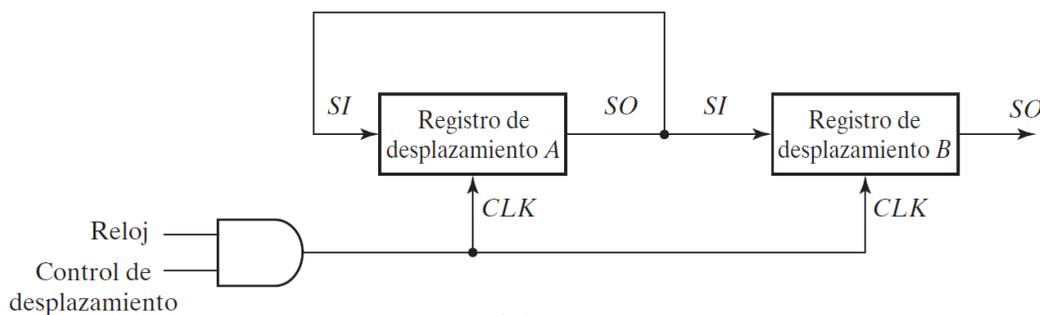
Fuente: MANO, Morris. *Diseño digital*. p 220.

### 2.6.1.2.1. Transferencia en serie

Un sistema digital opera en modo serie cuando la información se transfiere y manipula bit por bit. La información se transfiere bit por bit desplazando los bits del registro de origen hacia el registro de destino. Esto contrasta con la transferencia en paralelo, en la que todos los bits del registro se transfieren al mismo tiempo.

En la figura 123 se ve un diagrama de bloques de un circuito de transferencia de registros en serie. Este está compuesto por dos registros de desplazamiento A y B conectados en serie. Los datos que entran en el registro A se pierden si la información sigue ingresando, pero puede ser almacenada en el registro B. Si la información sigue ingresando y el reloj oscilando los registros siempre cambiarán, por ello es necesario agregar una entrada de control que llamará control de desplazamiento. Esta será una compuerta AND que permitirá hacer un desplazamiento de registros solo cuando esté activa.

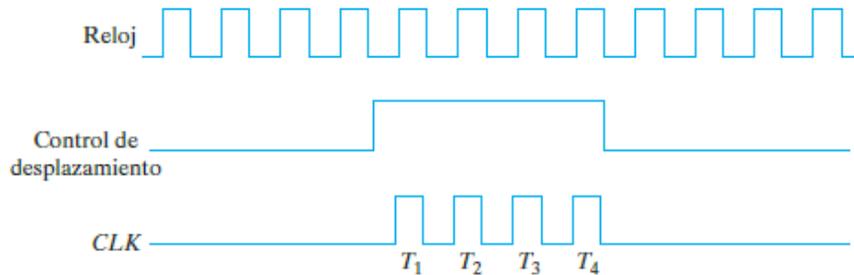
Figura 123. **Diagrama de bloques de un circuito de transferencia de registros en serie**



a) Diagrama de bloques

Fuente: MANO, Morris. *Diseño digital*. p 221.

Figura 124. Diagrama de temporización



Fuente: MANO, Morris. *Diseño digital*. p 221.

Ahora, se analizará un poco este tipo de registros con un caso real. Se supone que los registros de desplazamiento son de cuatro bits. Si se desea transferir los 4 bits del registro A, a los cuatro bits del registro B, serán necesarios cuatro pulsos de reloj para lograrlo.

Este proceso puede ser observado en la tabla LXIV y en la figura 124. Para esto la entrada de control de desplazamiento tiene que estar activa y debe desactivarse cuando hayan pasado los cuatro tiempos de reloj.

Tabla LXIV. Transferencia en serie

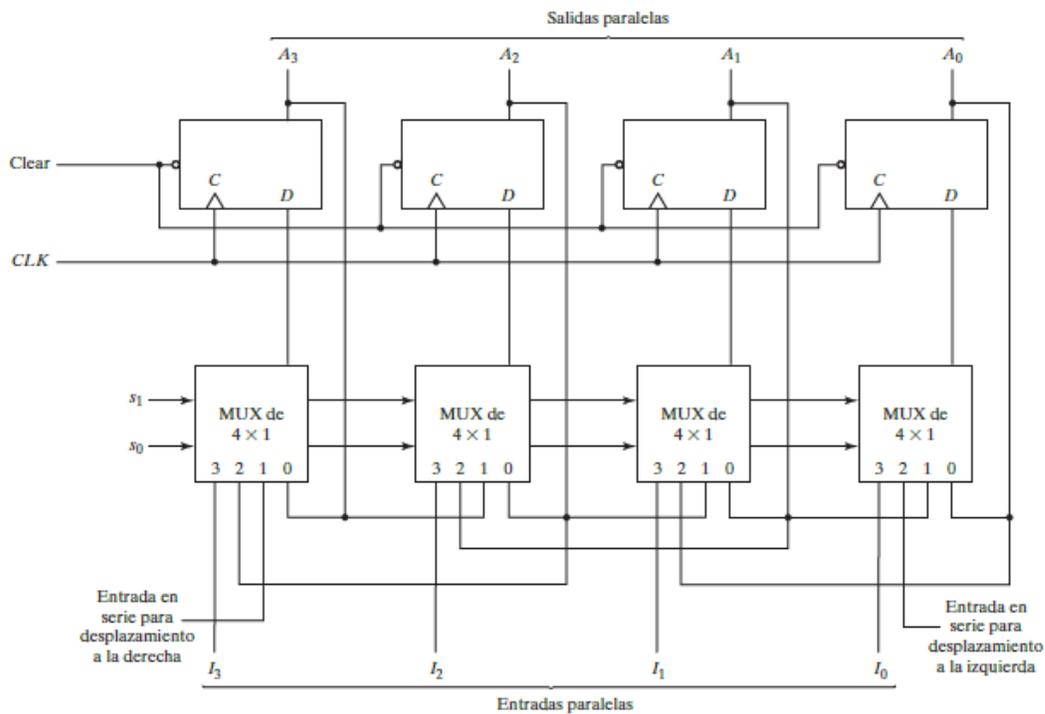
<b>Pulso de temporización</b>	<b>Registro de desplazamiento A</b>	<b>Registro de desplazamiento B</b>
Valor inicial	1 0 1 1	0 0 1 0
Después de $T_1$	1 1 0 1	1 0 0 1
Después de $T_2$	1 1 1 0	1 1 0 0
Después de $T_3$	0 1 1 1	0 1 1 0
Después de $T_4$	1 0 1 1	1 0 1 1

Fuente: MANO, Morris. *Diseño digital*. p 222.

### 2.6.1.3. Registro de desplazamiento universal

Estos registros pueden almacenar información de forma paralela y de forma bidireccional en serial. Cuando un registro es unidireccional puede almacenar información ya sea de derecha a izquierda y de izquierda a derecha. Un registro bidireccional permite almacenar información de izquierda a derecha y de derecha a izquierda.

Figura 125. Registro de desplazamiento universal



Fuente: MANO, Morris. *Diseño digital*. p 226.

Sus características son las siguientes:

- Una entrada de restablecimiento.
- Una entrada de reloj.
- Un control de desplazamiento a la derecha para habilitar el desplazamiento a la derecha.
- Un control de desplazamiento a la izquierda para habilitar el desplazamiento a la izquierda.
- Un control de carga en paralelo para habilitar la transferencia en paralelo.
- n líneas de salida en paralelo.
- Un control de estado para dejar los registros con los estados actuales.

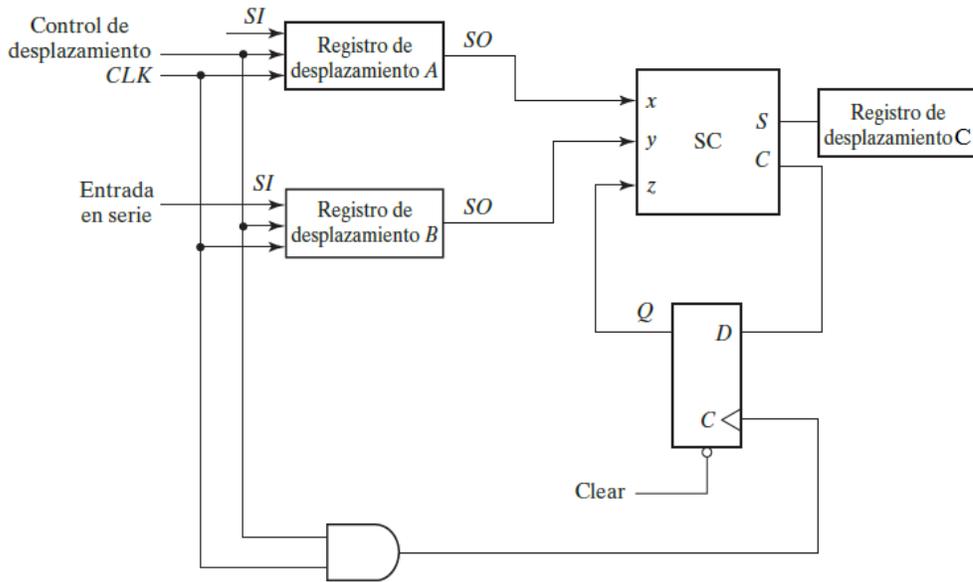
En la figura anterior se puede apreciar un registro de desplazamiento universal. Como se ve está compuesto de *flip-flops D* y multiplexores 4X1. Cuenta con dos entradas de control las cual permite realizar las siguientes funciones:

Tabla LXV. **Funciones del registro de desplazamiento universal de 4 bits**

<b>Control de modo</b>		<b>Operación del registro</b>
$s_1$	$s_0$	
0	0	Sin cambio
0	1	Desplazamiento a la derecha
1	0	Desplazamiento a la izquierda
1	1	Carga en paralelo

Fuente: MANO, Morris. *Diseño digital*. p 227.

Figura 126. **Sumador en serie**



Fuente: MANO, Morris. *Diseño digital*. p 223.

#### 2.6.1.4. **Suma en serie**

Las operaciones de las computadoras digitales por lo regular se efectúan en paralelo porque este modo de operación es más rápido. Las operaciones en serie son más lentas, pero tienen la ventaja de requerir menos *hardware*. Como se vio con anterioridad se puede realizar una suma en paralelo, donde todos los bits de los números binarios por sumar entran al mismo tiempo al sumador. Ahora se verá una versión de un sumador donde los sumandos serán ingresados de forma serial.

En la figura 126 se puede visualizar el circuito, el cual está compuesto por dos registros de desplazamiento, una compuerta AND para control, un sumador completo y un *flip-flop D* para el acarreo. Los dos números binarios

que se sumarán en serie se almacenan en dos registros de desplazamiento. Los bits se suman par por par utilizando un solo circuito de sumador completo (SC) como se observa en el diagrama. El acarreo de salida del sumador completo se transfiere a un *flip-flop D*. La salida de este *flip-flop* se utiliza entonces como acarreo de entrada para el siguiente par de bits significativos. El bit de suma de la salida *S* del sumador completo se transfiere a un tercer registro de desplazamiento *C*. El sumador en serie funciona como sigue. Inicialmente, el registro *A* contiene el primer sumando, el registro *B* contiene el segundo sumando y el *flip-flop* de acarreo está en 0. Las salidas (*SO*) de *A* y *B* alimentan un par de bits significativos al sumador completo en *x* y *y*. La salida *Q* del *flip-flop* alimenta el acarreo de entrada en *z*.

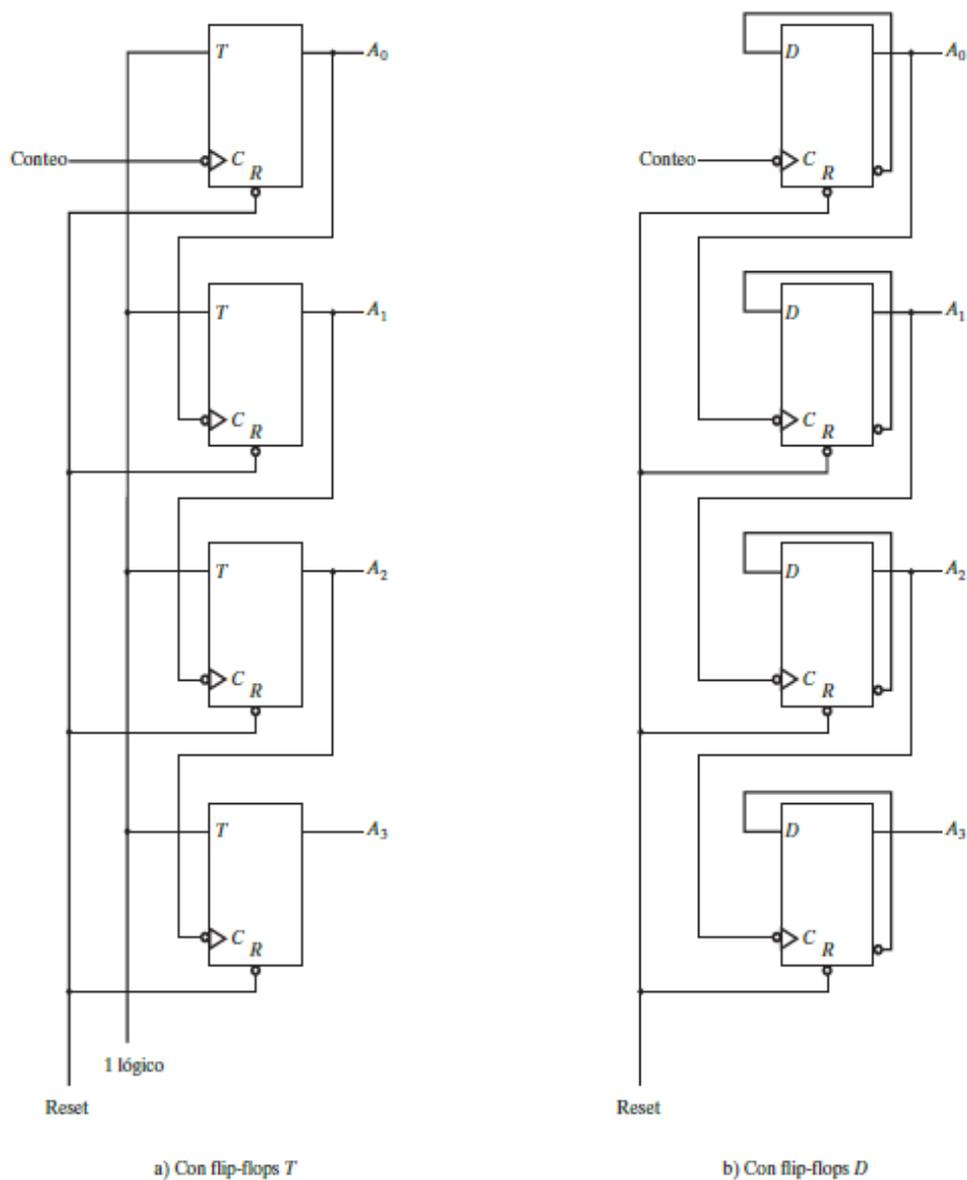
El control de desplazamiento habilita ambos registros y el *flip-flop* de acarreo, de modo que, en el siguiente pulso de reloj, ambos registros se desplazarán una vez a la derecha; el bit de suma de *S* ingresará en el *flip-flop* de extrema izquierda de *C*, y el acarreo de salida se transferirá al *flip-flop Q*. El control de desplazamiento habilita los registros durante un número de pulsos de reloj igual al número de bits que hay en los registros. Con cada pulso de reloj sucesivo, un nuevo bit de suma se transfiere a *C*, un nuevo acarreo se transfiere a *Q* y ambos registros se desplazan una vez a la derecha.

### **2.6.2. Contadores**

Un contador es básicamente un registro que pasa por una sucesión predeterminada de estados. Las compuertas del contador están conectadas de tal manera que producen la sucesión prescrita de estados binarios. La sucesión de estados podría seguir la sucesión numérica binaria o cualquier otro orden. Un contador que sigue la sucesión numérica binaria es un contador binario. Un contador binario de *n* bits puede contar en binario desde 0 hasta  $2^n - 1$ . Los

contadores pueden estar ligados a un reloj o no, es decir, pueden ser asíncronos o síncronos.

Figura 127. **Contador binario de rizo de 4 bits**



Fuente: MANO, Morris. *Diseño digital*. p 229.

### 2.6.2.1. Contadores de rizo

Son dispositivos contadores que tienen conectados los *flip-flops* en forma asíncrona, es decir, que no tienen conectadas las entradas de reloj en paralelo. En un contador de rizo la transición de salida del *flip-flop* sirve como disparador de otros *flip-flops*. Es decir, que la entrada C de algunos de los *flip-flops* no está conectada al reloj, sino que a la salida de otros *flip-flops*. Un contador de rizo es un circuito secuencial asíncrono.

#### 2.6.2.1.1. Contador binario de rizo

Un contador binario de rizo es un contador donde los *flip-flops* están conectados en serie y cuentan de manera ascendente o descendente (según configuración). La salida de cada *flip-flop* se conecta a la entrada C del siguiente *flip-flop* de orden superior. En la figura 127 se pueden apreciar dos contadores de rizo de cuatro bits, los cuales están compuestos con *flip-flops T* o *flip-flops D*. Se puede llegar a utilizar *flip-flops JK* uniendo las entradas JK y verificando la tabla característica del *flip-flop*. Estos *flip-flops* son con respuesta al borde negativo. Su funcionamiento es el de la tabla LXVI

Cada vez que  $A_0$  pasa de uno a cero, complementa a  $A_1$ . Cada vez que  $A_1$  pasa de uno a cero, complementa a  $A_2$ . Cada vez que  $A_2$  pasa de uno a cero, complementa a  $A_3$ . Y así sucesivamente con los demás bits de orden más alto que tenga el contador. Cuando el contador llega a quince, regresa al estado inicial (0000).

Este contador puede llegar a contar de forma descendente con una pequeña modificación. Un contador que cuenta de forma descendente es llamado contador de cuenta regresiva. En él, el estado inicial es quince y la

cuenta disminuye de uno a uno. Cuando el contador llega a 0, regresa al estado inicial (1111). Para ello es el mismo diagrama de la figura 128, solo que los *flip-flops* tienen que dispararse en el borde positivo. Si se usan *flip-flops* con el borde negativo, la entrada C de cada uno deberá conectarse a la salida del complemento del flip-flop anterior.

Tabla LXVI. **Comportamiento de contador binario de rizo de 4 bits**

$A_3$	$A_2$	$A_1$	$A_0$
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0

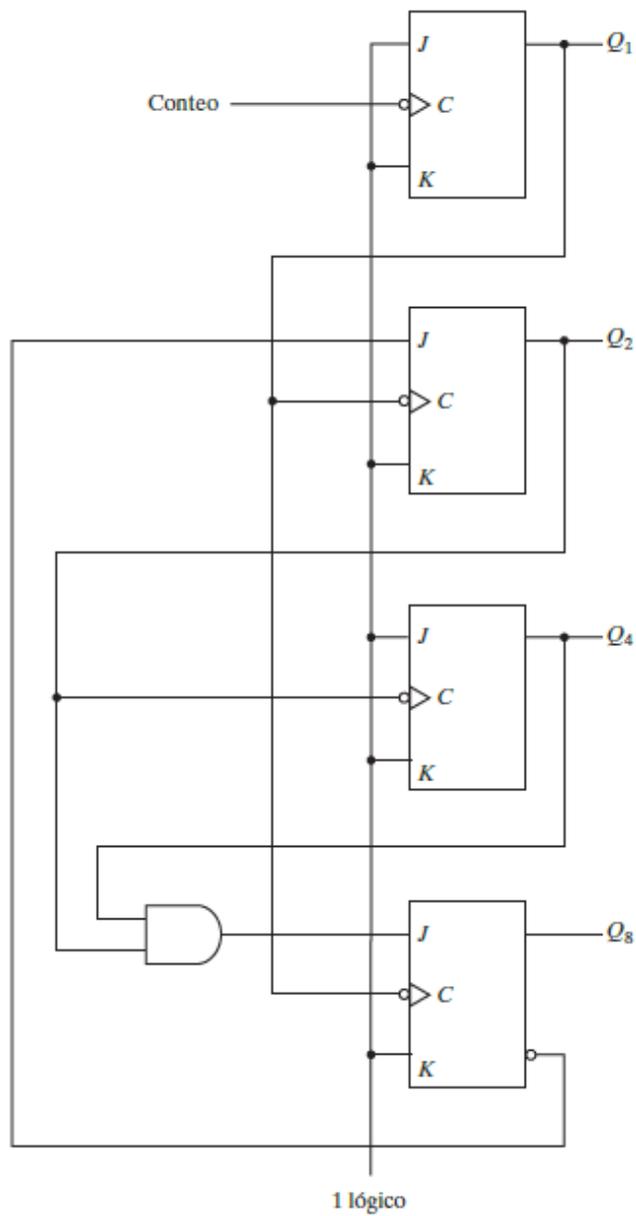
Fuente: MANO, Morris. *Diseño digital*. p 230.

#### 2.6.2.1.2. Contador BCD de rizo

Este es un contador decimal que tiene diez estados, que son los números del cero al nueve. Este es muy parecido al contador binario, con la diferencia que el siguiente estado del número nueve (1001) es el número cero (0000). Su diagrama de estados puede ser visualizado en la figura 130. En la figura 128 se puede ver un contador BCD de rizo el cual está compuesto por *flip-flops JK* y una compuerta AND. Los subíndices de las salidas Q indican el peso binario del bit correspondiente en el código BCD. Se ve que la salida  $Q_1$  se aplica a las entradas C tanto de  $Q_2$  como de  $Q_8$ , y que la salida de  $Q_2$  se aplica a la entrada

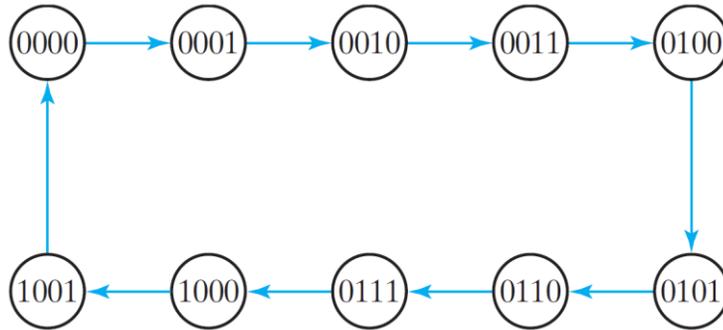
C de  $Q_4$ . Las entradas J y K se conectan con una señal de uno lógico permanente o bien a salidas de otros *flip-flops*.

Figura 128. **Contador BCD de rizo**



Fuente: MANO, Morris. *Diseño digital*. p 231.

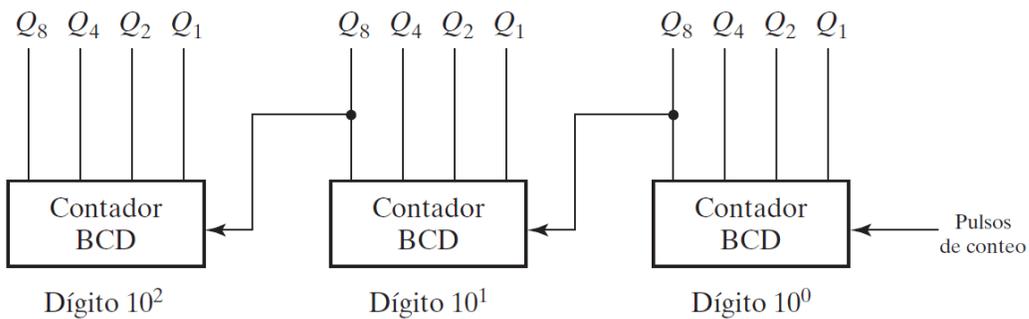
Figura 129. **Diagrama de estados de contador BCD**



Fuente: MANO, Morris. *Diseño digital*. p 230.

Muchas veces es necesario representar números mayores a solo una década. La figura 130 muestra una manera sencilla de conectar los contadores BCD para contar 3 décadas. De la misma manera se podría agregar una mayor cantidad de contadores BCD para una mayor cantidad de décadas.

Figura 130. **Diagrama de bloques de un contador BCD de 3 décadas**



Fuente: MANO, Morris. *Diseño digital*. p 231.

### 2.6.2.2. Contadores sincrónicos

En un contador sincrónico las salidas C de todos los *flip-flops* están conectadas al reloj común. Un solo contador dispara todos los *flip-flops* simultáneamente en vez de hacerlo uno por uno sucesivamente como en los contadores de rizo.

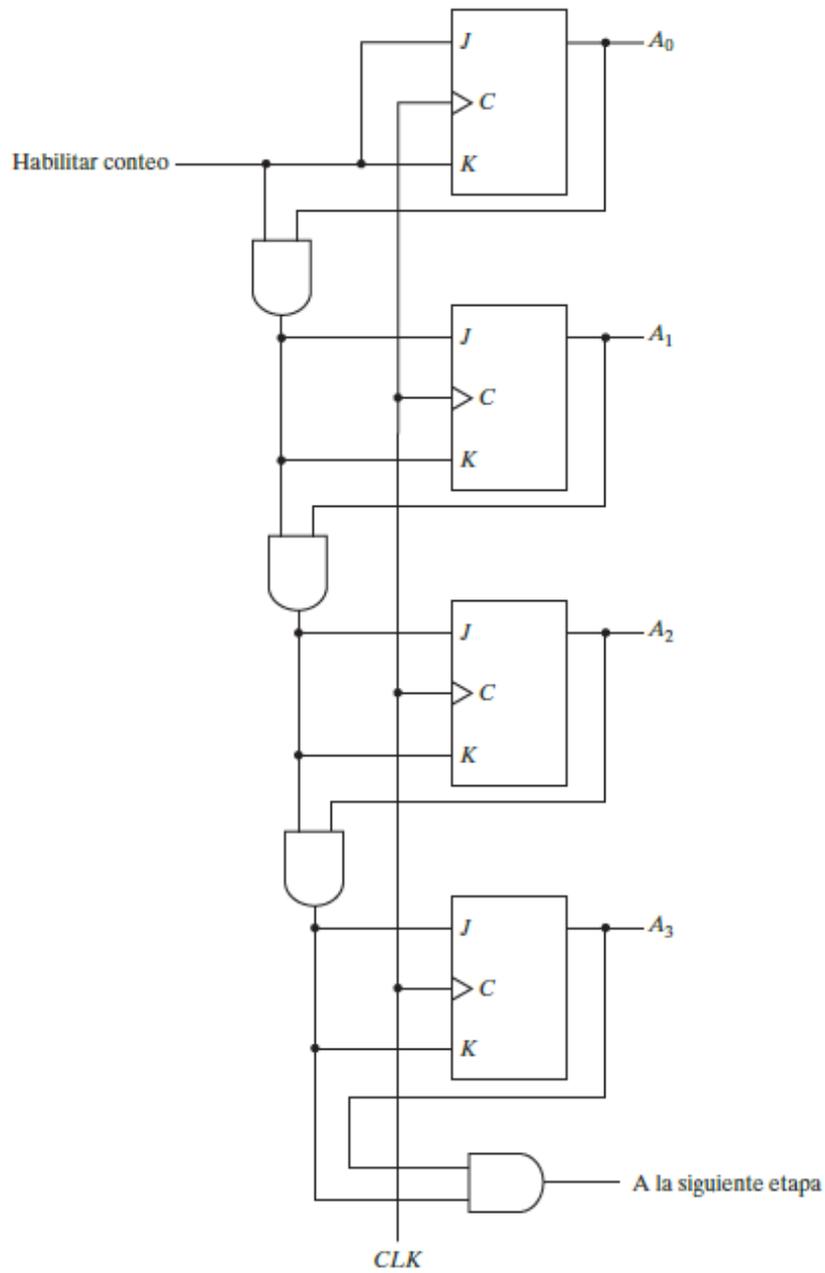
#### 2.6.2.2.1. Contador binario

En un contador binario sincrónico, el *flip-flop* de la posición menos significativa se complementa con cada pulso. Un *flip-flop* en cualquier posición se complementa cuando todos los bits de las posiciones significativas inferiores son uno. En la figura 132 se puede apreciar un contador binario sincrónico de cuatro bits, el cual está construido por *flip-flops JK*. Sus entradas J y K fueron unidas, por lo que se pudo haber utilizado *flip-flops T*.

Las entradas C de todos los *flip-flops* se conectan a un reloj común. El contador se habilita con la entrada de habilitar contador. Si esta entrada es cero, todas las entradas JK son cero, y el contador no cambia de estado. La primera etapa  $A_0$  tiene uno en J y K si el contador está habilitado. Las otras entradas JK tienen uno si todas las etapas anteriores menos significativa, producen uno y el conteo está habilitado.

El contador sincrónico binario se dispara con el borde positivo del reloj o el borde negativo (depende el *flip-flop* que se utilice). De igual manera que el contador de rizo, se puede volver descendente. Al utilizar las salidas de complemento en las compuertas AND se habilita este funcionamiento.

Figura 131. **Contador binario sincrónico**

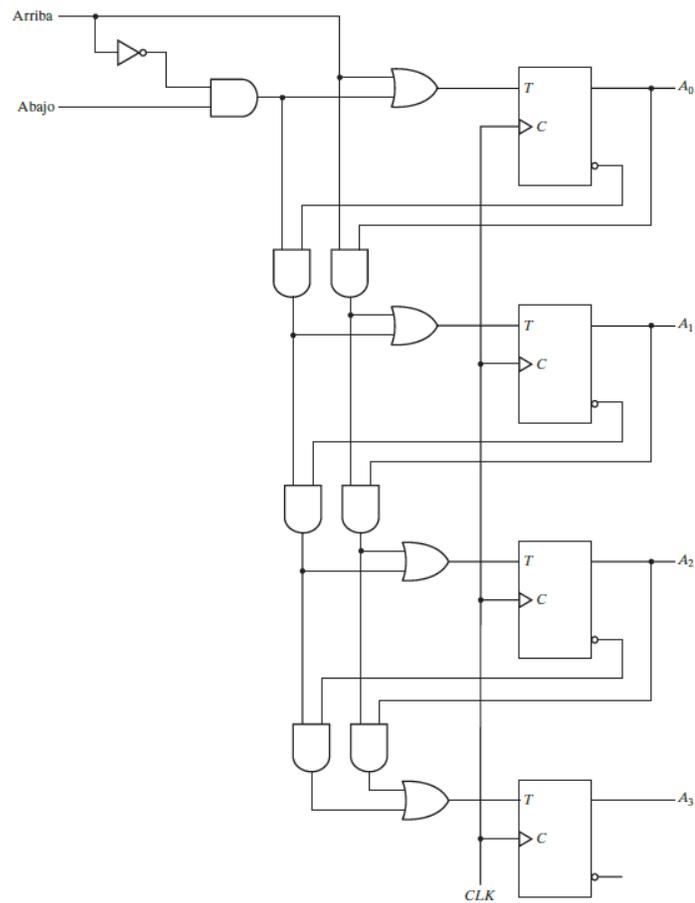


Fuente: MANO, Morris. *Diseño digital*. p 233.

### 2.6.2.2. Contador binario ascendente-descendente

Es posible combinar las operaciones de contador ascendente y contador descendente en un solo circuito. Este se presenta en la figura 132. Este circuito está compuesto con *flip-flops T*.

Figura 132. Contador binario ascendente-descendente de 4 bits



Fuente: MANO, Morris. *Diseño digital*. p 235.

Este tiene una entrada de control para conteo ascendente (Arriba) y una entrada de control para el conteo descendente (Abajo). Cuando la entrada de arriba es uno, el circuito cuenta hacia arriba, porque las entradas T reciben sus señales de las salidas normales de los *flip-flops* anteriores. Cuando la entrada abajo es uno y la entrada arriba es cero, el circuito cuenta hacia abajo, porque se aplican a las entradas T las salidas complementarias de los *flip-flops* anteriores. Si ambas entradas, arriba y abajo son cero, el circuito no cambia de estado. Si ambas entradas son uno, el circuito cuenta hacia arriba.

### 2.6.2.2.3. Contador BCD

Los contadores BCD cuentan en decimal codificado en binario, de 0000 hasta 1001 y luego regresan a 0000. La tabla LXVII muestra el funcionamiento del contador BCD sincrónico. La salida  $y$  se utiliza si se desea contar en más de una década. En base a esta tabla se puede obtener los siguientes minitérminos.

Tabla LXVII. **Tabla de comportamiento de contador sincrónico BCD**

Estado actual				Siguiete estado				Salida	Entradas de flip-flop			
$Q_8$	$Q_4$	$Q_2$	$Q_1$	$Q_8$	$Q_4$	$Q_2$	$Q_1$	$y$	$TQ_8$	$TQ_4$	$TQ_2$	$TQ_1$
0	0	0	0	0	0	0	1	0	0	0	0	1
0	0	0	1	0	0	1	0	0	0	0	1	1
0	0	1	0	0	0	1	1	0	0	0	0	1
0	0	1	1	0	1	0	0	0	0	1	1	1
0	1	0	0	0	1	0	1	0	0	0	0	1
0	1	0	1	0	1	1	0	0	0	0	1	1
0	1	1	0	0	1	1	1	0	0	0	0	1
0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	0	0	1	0	0	0	0	1
1	0	0	1	0	0	0	0	1	1	0	0	1

Fuente: MANO, Morris. *Diseño digital*. p 236.

$$\begin{aligned}
T_{Q1} &= Q'_8 Q'_4 Q'_2 Q'_1 + Q'_8 Q'_4 Q'_2 Q_1 + Q'_8 Q'_4 Q_2 Q'_1 + Q'_8 Q'_4 Q_2 Q_1 + Q'_8 Q_4 Q'_2 Q'_1 \\
&\quad + Q'_8 Q_4 Q'_2 Q_1 + Q'_8 Q_4 Q_2 Q'_1 + Q'_8 Q_4 Q_2 Q_1 + Q_8 Q'_4 Q'_2 Q'_1 \\
&\quad + Q_8 Q'_4 Q'_2 Q_1 \\
T_{Q2} &= Q'_8 Q'_4 Q'_2 Q_1 + Q'_8 Q'_4 Q_2 Q_1 + Q'_8 Q_4 Q'_2 Q_1 + Q'_8 Q_4 Q_2 Q_1 \\
T_{Q4} &= Q'_8 Q'_4 Q_2 Q_1 + Q'_8 Q_4 Q_2 Q_1 \\
T_{Q8} &= Q'_8 Q_4 Q_2 Q_1 + Q'_8 Q_4 Q_2 Q'_1 \\
y &= Q_8 Q'_4 Q'_2 Q_1
\end{aligned}$$

Ahora se procede a utilizar mapas de Karnaugh para simplificar las ecuaciones y de ese modo obtener un circuito más simplificado. Los mapas se muestran en la figura 133 y en la figura 134.

Figura 133. **Mapa de Karnaugh 1 de contador sincrónico BCD**

Para TQ1

Q8Q4\Q2Q1	00	01	11	10
00	1	1	1	1
01	1	1	1	1
11	X	X	X	X
10	1	1	X	X

Q8Q4\Q2Q1	00	01	11	10
00	1	1	1	1
01	1	1	1	1
11	X	X	X	X
10	1	1	X	X

Para TQ2

Q8Q4\Q2Q1	00	01	11	10
00	0	1	1	0
01	0	1	1	0
11	X	X	X	X
10	0	0	X	X

Q8Q4\Q2Q1	00	01	11	10
00	0	1	1	0
01	0	1	1	0
11	X	X	X	X
10	0	0	X	X

Fuente: elaboración propia, empleando Excel.

Figura 134. Mapa de Karnaugh 2 de contador sincrónico BCD

Para TQ4

Q8Q4\Q2Q1	00	01	11	10
00	0	0	1	0
01	0	0	1	0
11	X	X	X	X
10	0	0	X	X

Q8Q4\Q2Q1	00	01	11	10
00	0	0	1	0
01	0	0	1	0
11	X	X	X	X
10	0	0	X	X

Para TQ8

Q8Q4\Q2Q1	00	01	11	10
00	0	0	0	0
01	0	0	1	0
11	X	X	X	X
10	0	1	X	X

Q8Q4\Q2Q1	00	01	11	10
00	0	0	0	0
01	0	0	1	0
11	X	X	X	X
10	0	1	X	X

Para y

Q8Q4\Q2Q1	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	X	X	X	X
10	0	1	X	X

Q8Q4\Q2Q1	00	01	11	10
00	0	0	0	0
01	0	0	1	0
11	X	X	X	X
10	0	1	X	X

Fuente: elaboración propia, empleando Excel.

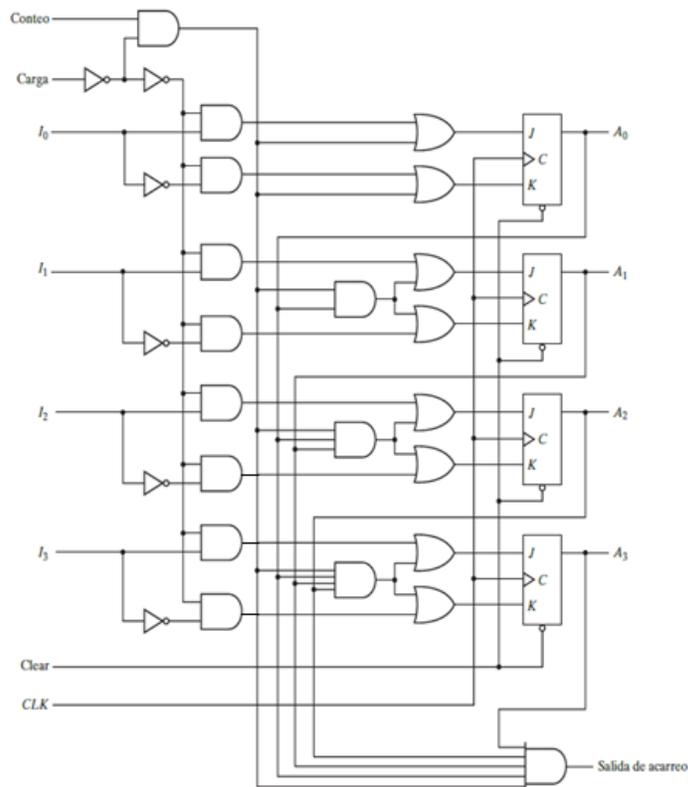
Con la reducción se obtienen las siguientes ecuaciones y el circuito de la figura 135.



#### 2.6.2.2.4. Contador binario con carga paralela

Es muy común que los contadores empleados en sistemas digitales requieran una capacidad de carga paralela para transferir un número binario inicial al contador antes de la operación de conteo. La figura 137 muestra el contador binario con carga paralela, el cual puede iniciar el conteo desde el número que le asigne. Su funcionamiento según sus entradas se describe en la tabla LXVIII.

Figura 136. Contador binario con carga paralela



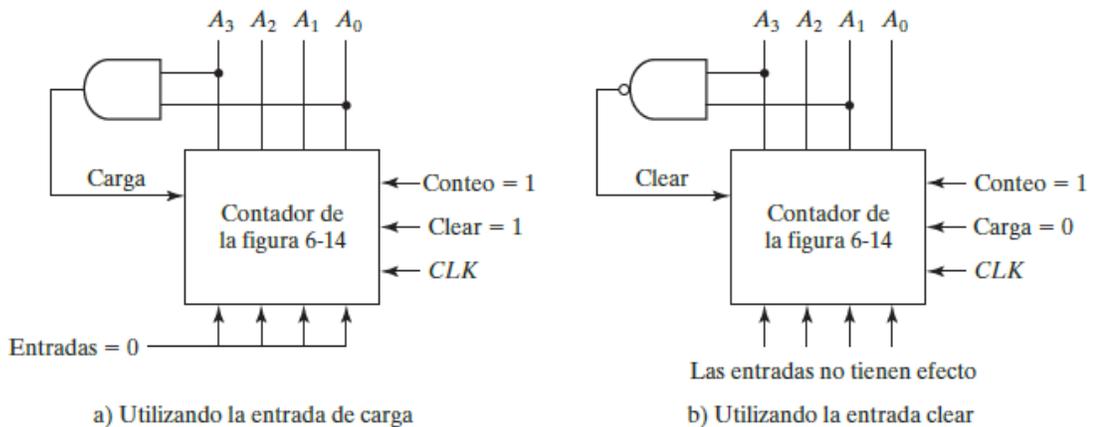
Fuente: MANO, Morris. *Diseño digital*. p 237.

Tabla LXVIII. **Funcionamiento del contador binario con carga paralela**

Clear	CLK	Carga	Conteo	Función
0	X	X	X	Poner en ceros
1	↑	1	X	Cargar entradas
1	↑	0	1	Contar al siguiente estado binario
1	↑	0	0	Sin cambio

Fuente: MANO, Morris. *Diseño digital*. p 238.

Figura 137. **Contador BCD empleando un contador con carga paralela**



Fuente: MANO, Morris. *Diseño digital*. p 239.

La salida del acarreo es uno si todas las salidas de los *flip-flops* son igual a uno y la entrada de conteo está habilitada. Las cuatro entradas de control: *clear*, CLK, carga y conteo, determinan el siguiente estado. La entrada *clear* es asincrónica y, si es cero, hace que el contador se despeje sin importar si hay pulsos de reloj y otras entradas.

Esto se indica con las entradas X, que representan condiciones de indiferencia para las demás entradas. La entrada *clear* debe tener uno para que sean habilitadas todas las demás funciones. Si la entrada de carga y conteo son cero, el contador no cambiará de estado. Cuando carga=1 causa una transferencia de las entradas paralelas. La entrada de carga debe ser cero para que la entrada de conteo controle el funcionamiento del contador. Haciendo una pequeña modificación se puede hacer que este circuito sea un contador BCD. Este se muestra en la figura 138. Como se ve, simplemente se agrega una compuerta AND al circuito.

#### 2.6.2.2.5. Contador con estados no utilizados

Hasta ahora se ha visto contadores que tienen una sucesión binaria, pero se puede diseñar contaderas que generen cualquier sucesión de estados deseada. Un contador con n *flip-flops* puede llegar a tener una cantidad de  $2^n$  estados.

Tabla LXIX. **Tabla de estados de contador con estados no utilizados**

Estado actual			Siguiete estado			Entradas de flip-flops					
A	B	C	A	B	C	$J_A$	$K_A$	$J_B$	$K_B$	$J_C$	$K_C$
0	0	0	0	0	1	0	X	0	X	1	X
0	0	1	0	1	0	0	X	1	X	X	1
0	1	0	1	0	0	1	X	X	1	0	X
1	0	0	1	0	1	X	0	0	X	1	X
1	0	1	1	1	0	X	0	1	X	X	1
1	1	0	0	0	0	X	1	X	1	0	X

Fuente: MANO, Morris. *Diseño digital*. p 240.

En algunos circuitos digitales no es necesario incluir todos los estados que pueden otorgar el número de *flip-flops*. Un contador que no utilice todos sus estados es llamado contador con estados no utilizados. Para diseñar este tipo de circuitos no es necesario agregar en la tabla de estados los estados que no se vayan a utilizar. En la tabla LXIX se puede observar una tabla de estados de un circuito con estados no utilizados. Para este circuito se utilizan tres *flip-flops JK*.

Se puede observar que los estados no tienen un orden específico y se han omitido los estados 011 y 111. Por medio de la tabla de estados se puede obtener las siguientes ecuaciones:

$$J_A = A'BC'$$

$$K_A = ABC'$$

$$J_B = A'B'C + AB'C$$

$$K_B = A'BC' + ABC'$$

$$J_C = A'B'C' + AB'C'$$

$$K_C = A'B'C + AB'C$$

Con la ayuda de mapas de Karnaugh (proceso expuesto en la figura 139) se simplifican las ecuaciones anteriores y obtienen las siguientes:

$$J_A = B$$

$$K_A = B$$

$$J_B = C$$

$$K_B = 1$$

$$J_C = B'$$

$$K_C = 1$$

Figura 138. Mapas de Karnaugh de contador

Para JA

A\BC	00	01	11	10
0	0	0	X	1
1	X	X	X	X

A\BC	00	01	11	10
0	0	0	X	1
1	X	X	X	X

Para KA

A\BC	00	01	11	10
0	X	X	X	X
1	0	0	X	1

A\BC	00	01	11	10
0	X	X	X	X
1	0	0	X	1

Para JB

A\BC	00	01	11	10
0	0	1	X	X
1	0	1	X	X

A\BC	00	01	11	10
0	0	1	X	X
1	0	1	X	X

Para KB

A\BC	00	01	11	10
0	X	X	X	1
1	X	X	X	1

A\BC	00	01	11	10
0	X	X	X	1
1	X	X	X	1

Para JC

A\BC	00	01	11	10
0	1	X	X	0
1	1	X	X	0

A\BC	00	01	11	10
0	1	X	X	0
1	1	X	X	0

Para KC

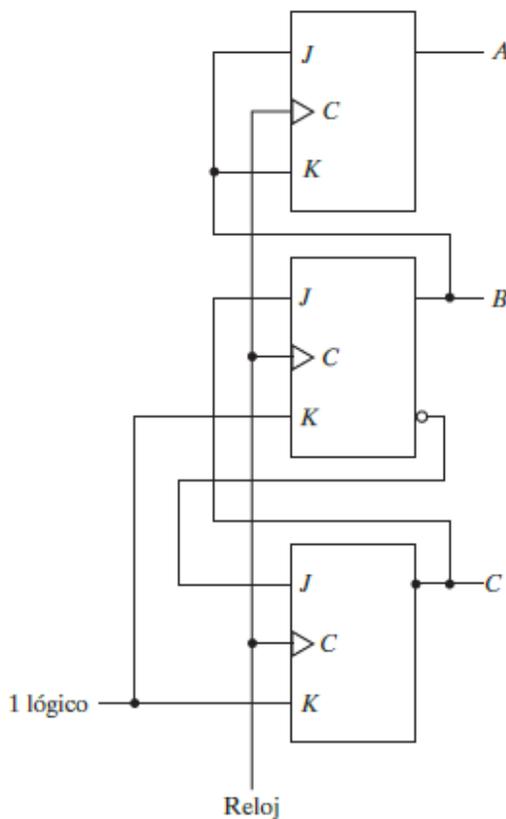
A\BC	00	01	11	10
0	X	1	X	X
1	X	1	X	X

A\BC	00	01	11	10
0	X	1	X	X
1	X	1	X	X

Fuente: elaboración propia, empleando Excel.

Ahora con las ecuaciones obtenidas, se puede realizar el diagrama lógico del circuito, que es la figura 140. Para el diseño de este tipo de circuitos hay que tener cierto tipo de cuidado, dado a que es probable que los circuitos puedan llegar a los estados omitidos (en este caso 011 y 111). Por ello hay que asegurarse que, si por alguna razón el circuito caiga es uno de estos estados, pueda regresar a uno de los estados conocidos. Un circuito que tiene esta propiedad se llama contador con autocorrección.

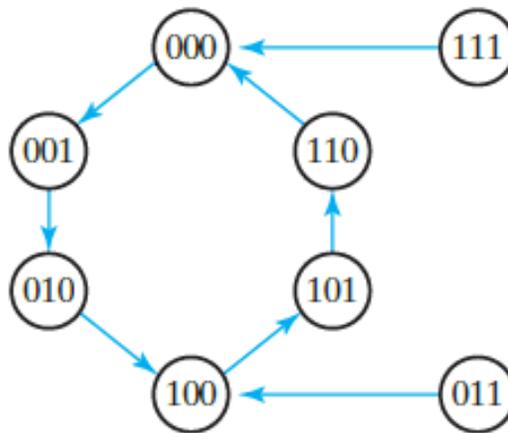
Figura 139. **Diagrama lógico del contador**



Fuente: MANO, Morris. *Diseño digital*. p 241.

Analizando el circuito anterior, se puede observar que si el circuito llega a caer en el estado 011, las entradas de JK de los *flip-flops* A y B tienen uno por lo que se complementarían al próximo pulso de reloj. El *flip-flop* C tendrá la entrada K=1 y la entrada J=0; lo que indica que su siguiente estado será cero. Entonces, el siguiente estado de 011 será 100, el cual sí se encuentra en los estados. De manera similar sucede con el estado 111, que su estado siguiente es 000. Con esta información se puede deducir el siguiente diagrama de estados del circuito.

Figura 140. **Diagrama de estados del contador**



Fuente: MANO, Morris. *Diseño digital*. p 241.

#### 2.6.2.2.6. Contador anular

El contador anular o de anillo es un circuito en donde solo un *flip-flop* está establecido y todos los demás *flip-flops* están restablecidos. Esto quiere decir que solo uno de los *flip-flop* estará en uno y los demás que componen el contador en cero. Para ejemplificar esto se utilizará como ejemplo un contador anular de tres bits. Los estados del contador anular son: 100, 010, 001.

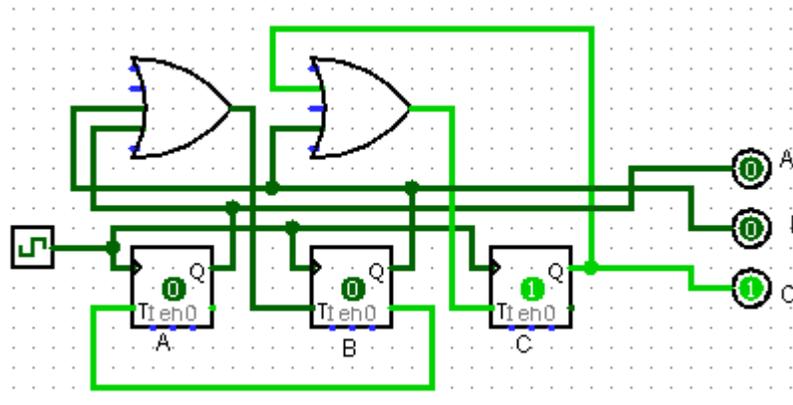
También se ha incluido el estado 000 porque el contador debe de saber qué hacer si todos los *flip-flops* están restablecidos. En la tabla LXX se puede apreciar el comportamiento del contador anula de tres bits y se puede deducir las ecuaciones que le siguen.

Tabla LXX. **Comportamiento del contador anular**

Estado actual			Estado siguiente			Entradas <i>flip-flops</i>		
A	B	C	A	B	C	Ta	Tb	Tc
0	0	0	1	0	0	1	0	0
1	0	0	0	1	0	1	1	0
0	1	0	0	0	1	0	1	1
0	0	1	1	0	0	1	0	1

Fuente: elaboración propia, empleando Excel.

Figura 141. **Contador anular de 3 bits a**



Fuente: elaboración propia, empleando Logisim.

$$T_A = A'B'C' + AB'C' + A'B'C$$

$$T_A = AB'C' + A'BC'$$

$$T_A = AB'C' + A'B'C$$

Ahora con la ayuda de mapas de Karnaugh se obtienen las siguientes ecuaciones. Los mapas de Karnaugh están expuestos en la figura 142.

$$T_A = B'$$

$$T_B = A + B$$

$$T_C = B + C$$

Figura 142. **Mapas de Karnaugh del contador anular**

Para Ta

A\BC	00	01	11	10
0	1	1	X	0
1	1	X	X	X

A\BC	00	01	11	10
0	1	1	X	0
1	1	X	X	X

Para Tb

A\BC	00	01	11	10
0	0	0	X	1
1	1	X	X	X

A\BC	00	01	11	10
0	0	0	X	1
1	1	X	X	X

Para Tc

A\BC	00	01	11	10
0	0	1	X	1
1	0	X	X	X

A\BC	00	01	11	10
0	0	1	X	1
1	0	X	X	X

Fuente: elaboración propia, empleando Excel.

Con las ecuaciones obtenidas se puede realizar el circuito correspondiente (figura 142). Este circuito también es posible realizarlo con la ayuda de un

contador binario y un decodificador. En este caso la necesidad (en número) de *flip-flops* se reduce, pero la circuitería combinacional se aumenta.

Este tipo de circuito tiene un defecto, el cual es que son necesarios muchos componentes para su realización. Si se desea tener dieciséis estados hay que utilizar dieciséis *flip-flops*. Si se desea utilizar un contador binario, son necesarias dieciséis compuertas AND.

### 2.6.2.2.7. Contador Johnson

El contador Johnson es un contador anular con extremos conmutados de  $k$  bits provistos de  $2k$  compuertas decodificadoras para generar salidas correspondientes a  $2k$  señales de temporización. Un conmutador anular con extremo conmutado es un registro de desplazamiento circular en el que la salida del complemento del último *flip-flop* está conectada a la entrada del primer *flip-flop*. Se vislumbra un contador anular con extremos conmutados en la figura 144.

Tabla LXXI. **Funcionamiento del contador Johnson**

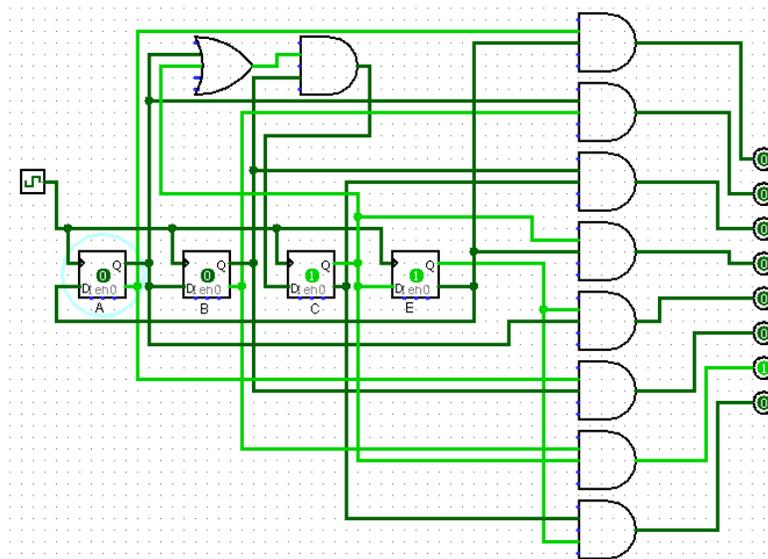
Número sucesivo	Salidas de los flip-flops				Compuerta AND requerida para la salida
	A	B	C	E	
1	0	0	0	0	$A'E'$
2	1	0	0	0	$AB'$
3	1	1	0	0	$BC'$
4	1	1	1	0	$CE'$
5	1	1	1	1	$AE$
6	0	1	1	1	$A'B$
7	0	0	1	1	$B'C$
8	0	0	0	1	$C'E$

Fuente: MANO, Morris. *Diseño digital*. p 243.

Su comportamiento lo expone la tabla LXXI, el cual se ve que posee una cantidad de cuatro *flip-flops* y ocho salidas distintas.

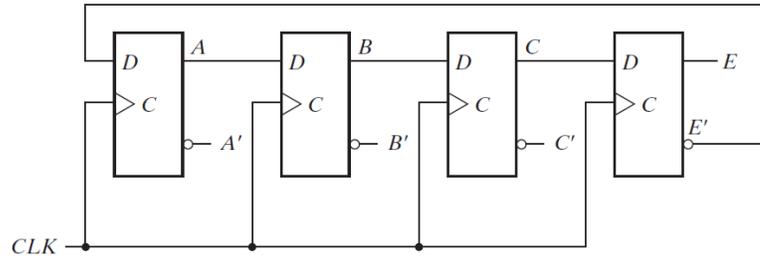
Se ve que su tabla de funcionamiento es muy similar a la de un registro de desplazamiento. Habrá una compuerta AND que decodifique las salidas de los *flip-flop* para dar paso a la salida requerida. Este circuito está compuesto por *flip-flops D*.

Figura 143. **Contador Johnson**



Fuente: elaboración propia, empleando Logisim.

Figura 144. **Contador anular con extremos conmutados de cuatro etapas**



Fuente: MANO, Morris. *Diseño digital*. p 243.

En este caso solo se necesitan cuatro *flip-flops* y ocho compuertas AND para tener 8 estados en el contador anular. Aparte de la circuitería expuesta en la tabla, es necesario agregar un arreglo combinacional para este circuito. Esto porque al pasar a un estado no válido, el circuito persistirá a pasar a otro estado no válido, y nunca llegará a un estado conocido. Para evitar esta condición indeseable, es necesario agregar el siguiente arreglo al *flip-flop C*.

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



### **3. PREGUNTAS Y EJERCICIOS CONCEPTUALES DEL CURSO ELECTRÓNICA 3**

#### **3.1. Preguntas y ejercicios conceptuales de sistemas de numeración**

Al finalizar el capítulo de sistemas de numeración el alumno deberá ser capaz de resolver las preguntas y ejercicios propuestos a continuación.

##### **3.1.1. Preguntas conceptuales**

Las preguntas conceptuales son preguntas directas que contienen los temas del capítulo de sistemas de numeración.

- ¿Qué es un sistema de numeración? Es un conjunto de símbolos y reglas que permiten representar todos los números válidos.
- ¿Cuál es el sistema de numeración que se utiliza cotidianamente? El sistema decimal.
- ¿Qué dígitos se utilizan en el sistema binario? Uno (1) y cero (0).
- ¿Qué indica el subíndice de un número y porque no se utiliza en el sistema de numeración decimal? El subíndice indica cual es el sistema numérico que se utilizó para representar esa cantidad. No es utilizado en el sistema de numeración decimal porque este es utilizado en forma cotidiana. Si el número no presenta ningún subíndice, se sobreentiende que se utilizó el sistema de numeración decimal para representar dicha cantidad.
- ¿Cuántos dígitos binarios corresponden a un dígito octal? ¿Y a uno hexadecimal? Corresponden 3 y 4 dígitos binarios respectivamente.

- ¿De qué necesidad surgieron los sistemas de numeración octal y hexadecimal? Algunas veces es necesario que el operador se comunique con la computadora a través de números binarios. Aprovechando la relación que hay entre los tres sistemas de numeración, es más sencilla la comprensión de números hexadecimales y octales a comparación de un número binario.
- ¿Cuál es la forma más rápida de obtener el complemento a uno de un número binario? Invirtiendo cada uno de los dígitos del número binario.
- ¿Cuál es una manera de obtener el complemento a 15 de un número hexadecimal? Restar el número hexadecimal a un número cuya cantidad de sietes, sea igual a la cantidad de dígitos del número hexadecimal.
- ¿Por qué es importante la resta de complementos? Cuando se desea restar con hardware, es mucho más eficiente el uso de complementos.
- ¿Hay diferencia entre los valores de un número positivo con signo a un número positivo sin signo? Explique su respuesta. Dependerá con qué número (cero o uno) el usuario haya decidido representar el signo negativo. Si el número uno representa el signo negativo, no habría diferencia, pero si el número uno representa un signo positivo, sí habría diferencia.
- ¿Cuál es la aplicación (no un dispositivo) de la codificación BCD? Una representación en decimal de un número binario.
- ¿Por qué razones no es utilizada la codificación BCD para el procesamiento de datos? Porque utiliza una mayor cantidad de bits que el código binario natural y que las operaciones aritméticas son más complicadas que en la codificación binario natural.
- ¿En qué difiere la codificación BCD y la codificación 2421? En el valor numérico que representa cada dígito de los cuatro bits. En la codificación BCD 8 4 2 1 y en la codificación 2421 2 4 2 1.

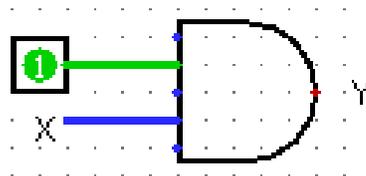
- ¿Cuál es la principal ventaja del código Gray con el código BCD? La principal ventaja que existe en el código Gray es que, entre dos números sucesivos, existe un cambio de solo un bit.
- ¿Por qué la codificación de detección de errores no puede detectar una cantidad par de errores? Al tener una cantidad par de errores, la cantidad de unos en la trama sería de paridad igual a la paridad que se toma como correcta.
- ¿Cuál es la región de transición de las compuertas lógicas? De 1V a 5V.
- ¿Cuáles son las compuertas lógicas? YES/Buffer/EQUAL, OR, AND, XOR/OR EXCLUSIVA/OREX, NOT, NOR, NAND, XNOR.

### 3.1.2. Ejercicios conceptuales

Los ejercicios conceptuales son preguntas prácticas que contienen los temas del capítulo de sistemas de numeración.

- ¿Qué ocurre si en la entrada de una compuerta lógica hay un valor de entre 1 voltio a 1,5 voltios? El resultado sería impredecible porque el valor se encuentra en la región de transición.

Figura 145. **Circuito del ejercicio conceptual 1 capítulo 1**



Fuente: elaboración propia, empleando Logisim.

- Tomar de referencia el circuito de la figura 146. Si la entrada uno (color verde) de la compuerta AND es uno y la entrada 2 (X) de la compuerta AND se encuentra en la región de transición, ¿Cuál sería el valor de la salida Y? ¿Qué diferencia habría si la compuerta fuera una OR? Explique su respuesta. Dado a que el valor de la salida de AND es uno solo si todas sus entradas son igual a uno, no se puede estar seguros de ningún valor de la salida Y, por el valor incierto de la entrada X. Si fuera una compuerta OR el valor de la salida sería uno, dado a que la compuerta OR es uno si alguna de sus entradas es igual a uno.
- Pablo tiene una venta de tomates en la terminal, y necesita una calculadora para llevar la cuenta de los tomates que tiene. Él no tiene una pero tiene los conocimientos y componentes suficientes para crear una. La calculadora que fue diseñada es de ocho bits y decide no agregar un bit de signo para utilizar los ocho bits para el conteo. ¿Fue esto prudente (suponiendo que la calculadora sola tendrá la función establecida)? Explique su respuesta. Si fue prudente, dado a que en el conteo de algo siempre serán cantidades positivas y con los ocho bits podrá contar mucho más que con siete bits. Claro, si se hubiera decidido que el número tuviera signo, se tendría la misma cantidad de estados, pero una magnitud menor
- Juan es un ingeniero electrónico y funda su empresa de dispositivos electrónicos. Al ver el éxito de Juan, Pedro también funda su empresa de dispositivos electrónicos. Juan utiliza la codificación BCD para el procesamiento de datos en sus dispositivos y Pedro utiliza la codificación binario natural para la misma tarea. ¿Cuál de las dos será el empresario más exitoso (sin tomar en cuenta cualquier otro factor no mencionado)? Explique su respuesta. La empresa de Pedro será más exitosa porque, al utilizar la codificación binario natural para el procesamiento de datos, sus dispositivos serán más eficientes y baratos que los de Juan. Esto porque

necesitará menor cantidad de bits para su procesamiento y las operaciones aritméticas serán más sencillas.

### 3.2. Preguntas y ejercicios conceptuales del álgebra booleana

Al finalizar el capítulo de algebra booleana el alumno deberá ser capaz de resolver las preguntas y ejercicios propuestos a continuación.

#### 3.2.1. Preguntas conceptuales

Las preguntas conceptuales son preguntas directas que contienen los temas del capítulo de álgebra booleana.

- ¿Cuáles son los elementos de Huntington?
  - Cerradura
    - Cerradura con respecto al operador +
    - Cerradura con respecto al operador •
  - Elemento identidad
    - Un elemento de identidad respecto de +, designado por 0:  $x + 0 = 0 + x = x$
    - Un elemento de identidad respecto de •, designado por 1:  $x \cdot 1 = 1 \cdot x = x$
  - Conmutativa
    - Conmutativa respecto de +:  $x + y = y + x$
    - Conmutativa respecto de •:  $x \cdot y = y \cdot x$
  - Distributiva
    - • distributiva sobre +:  $x \cdot (y + z) = (x \cdot y) + (x \cdot z)$
    - + distributiva sobre •:  $x + (y \cdot z) = (x + y) \cdot (x + z)$

- Para cada elemento  $x$ , existe un elemento  $x'$  (llamado complemento de  $x$ ) tal que:
    - $x + x' = 1$
    - $x \cdot x' = 0$
  - Existen al menos dos elementos tales que  $x \neq y$
- ¿Qué establece el principio de dualidad? Establece que toda expresión algebraica que puede deducirse de los postulados del algebra booleana seguirá siendo válida si se intercambian los operadores y los elementos de identidad.
  - ¿Cuáles son las 3 formas de expresar una función booleana? Expresión algebraica, tabla de verdad y sumatoria.
  - ¿Cuál es la finalidad de simplificar las expresiones algebraicas en el álgebra de Boole? Utilizar la menor cantidad de hardware posible obteniendo los mismos resultados.
  - ¿Qué información se puede obtener de la siguiente sumatoria? Que la función  $F$  que depende de las variables  $M$ ,  $N$ , y  $L$ , es igual a uno en los reglones: cero, uno, tres, siete, ocho, diez, 16, 20, 25, 30, 35 y 60.

$$F(M, N, L) = \sum (0,1,3,7,8,10,16,20,25,30,35,60)$$

- ¿Cuál es el complemento de una función? En un circuito lógico ¿Cómo se puede obtener el complemento de una función? El complemento de una función  $F$  es  $F'$  y se obtiene intercambiando ceros por unos y unos por ceros en la salida de la función  $F$ . Colocando una compuerta NOT en la salida  $F$  del circuito lógico se puede obtener la función  $F'$ .

- ¿Cuál es la diferencia entre los maxitérminos y los minitérminos? Los minitérminos son productos donde las variables son cada uno de los factores y los maxitérminos son sumas que sus sumandos son cada una de las variables.
- Comprobar que los maxitérminos son el complemento de los minitérminos correspondientes y viceversa. Para comprobar esto se debe aplicar el teorema de DeMorgan a los minitérminos (o maxitérmino) para obtener el complemento correspondiente.

$$\begin{aligned}
 & x'y'z' \\
 (x'y'z')' &= (x + y + z) \\
 (x + y + z)' & \\
 (x)'(y)'(z)' &= x'y'z'
 \end{aligned}$$

- ¿Cuál es la diferencia entre producto de sumas y suma de productos? El producto de suma está compuesto por términos AND que forman un término OR y la suma de productos son términos OR que forman un término AND.
- ¿Cuál es la diferencia entre lógica positiva y lógica negativa? La lógica positiva es aquella que el uno lógico es representado por el valor más alto de voltaje, y la lógica negativa es aquella que el uno lógico es representado del valor más bajo de voltaje.
- ¿Qué es un CI? Es un cristal semiconductor de silicio, llamado chip, que contiene los componentes electrónicos para construir compuertas digitales.
- ¿Cuándo es ventajoso utilizar los CI ECL? Resulta ventajoso en sistemas que deben operar a alta velocidad.
- ¿Cuándo es ventajoso utilizar los CI MOS? Resulta conveniente para circuitos que se requieren una densidad elevada de componentes.

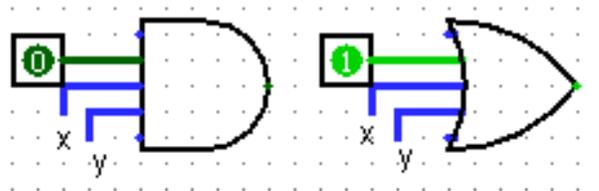
- ¿Cuándo es ventajoso utilizar CI CMOS? Resulta ventajoso cuando no se cuenta con altos niveles de potencia.

### 3.2.2. Ejercicios conceptuales

Los ejercicios conceptuales son preguntas prácticas que contienen los temas del capítulo de álgebra booleana.

- ¿Qué principio relaciona los dos siguientes diagramas? El principio de dualidad. La expresión que representa al primer diagrama es:  $x * y * 0 = 0$  y la expresión que representa el segundo diagrama es:  $1 + x + y = 1$ .

Figura 146. Diagrama de ejercicio conceptual 1 capítulo 2

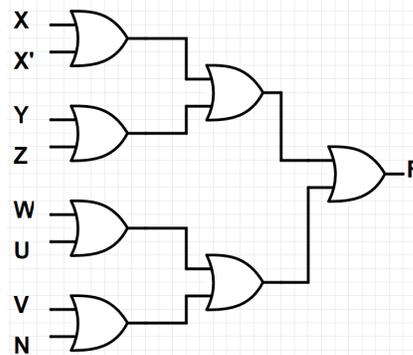


Fuente: elaboración propia, empleando Logisim.

- ¿Cuál es el valor de la salida F del circuito de la figura 148? Explique su respuesta. F siempre será igual a uno porque:

$$\begin{aligned}
 F &= X + X' + Y + Z + W + U + V + N \\
 F &= (X + X') + Y + Z + W + U + V + N \\
 F &= 1 + Y + Z + W + U + V + N \\
 F &= 1
 \end{aligned}$$

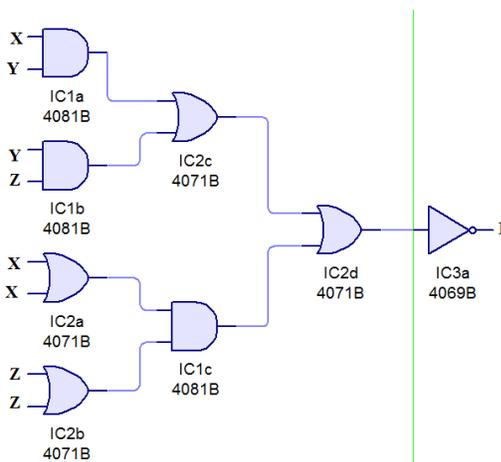
Figura 147. Diagrama de ejercicio conceptual 3 capítulo 2



Fuente: elaboración propia, empleando Logisim.

- Si se tiene un circuito el cual está conformado por 8 compuertas ¿Qué tipo de CI utilizaría (deducir con respecto al nivel de integración)? Dado a que se deduce por medio de los niveles de integración, el tipo de CI es SSI.

Figura 148. Diagrama de ejercicio conceptual 4 capítulo 2



Fuente: elaboración propia, empleando Logisim.

- Encontrar la tabla de verdad, la expresión algebraica y la sumatoria del diagrama lógico de la figura 149. La expresión algebraica escribirla por medio de minitérminos.

$$F = (xy + yz + xz)'$$

$$F = (xy)'(yz)'(xz)'$$

$$F = (x' + y')(y' + z')(x' + z')$$

$$F = x'y' + x'z' + y'y' + y'z'(x' + z')$$

$$F = x'x'y' + x'x'z' + x'y'y' + x'y'z' + x'y'z' + x'z'z' + y'y'z' + y'z'z'$$

$$F = x'y' + x'z' + x'y' + x'y'z' + x'y'z' + x'z' + y'z' + y'z'$$

$$F = x'y' + x'z' + x'y'z' + y'z'$$

Figura 149. **Respuesta del ejercicio conceptual 4 capítulo 2**

x	y	z	F
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	0

Fuente: elaboración propia, empleando Excel.

$$F(x, y, z) = \sum (0,1,2,3,4)$$

### **3.3. Preguntas y ejercicios conceptuales de minimización de funciones booleanas**

Al finalizar el capítulo de funciones booleanas el alumno deberá ser capaz de resolver las preguntas y ejercicios propuestos a continuación.

#### **3.3.1. Preguntas conceptuales**

Las preguntas conceptuales son preguntas directas que contienen los temas del capítulo de álgebra booleana.

- ¿Qué es un mapa de Karnaugh? Es una herramienta matemática que ayuda a simplificar funciones booleanas.
- ¿Cuándo es necesaria la utilización de mapas de Karnaugh? Luego de haber simplificado la función booleana por medio de los postulados y teoremas del álgebra de Boole.
- ¿Por qué es recomendable resolver los mapas de Karnaugh de mayor de 5 variables por medio de software? La cantidad de minterminos es  $2^5 = 32$ , y al hacerlo a mano el resultado es muy susceptible a errores.
- Al momento de realizar un mapa de Karnaugh ¿Qué codificación se utiliza para ordenar los minterminos? El código Gray porque al hacer el mapa de Karnaugh se busca que, entre los cuadros adyacentes, solo una variable cambie de estado.
- Se desea obtener un producto de sumas al simplificar una función con la ayuda de mapas de Karnaugh ¿Cuál es el procedimiento? Es muy similar al del mapa de Karnaugh, pero con la diferencia que los valores de interés son los cero. Luego de esto se le aplica al resultado el teorema de DeMorgan.

- ¿Cuándo es conveniente obtener como resultado un producto de sumas? Dependerá de las limitaciones de hardware que se tengan. En el diseño digital a veces es conveniente tener un cierto tipo de compuertas lógicas.
- Cuando se habla de los mapas de Karnaugh, ¿A qué se refiere el término indiferencia? Hace referencia de los minterminos o maxiterminos que no influyen en el resultado de una función. Tienen tan poca influencia que pueden ser cambiados al estado que el usuario desee (ya sea cero o uno).
- ¿Qué es un mintermino indiferente? Es un mintermino el cual su valor no es relevante en una función booleana. Puede ser uno o cero según se necesite y es representado por la letra X.
- ¿A qué se le llama salto cuando se habla de niveles de compuertas lógicas? Es el proceso que consiste en pasar a través de una compuerta lógica.
- ¿Por qué es preferible utilizar las compuertas NAND y NOR en vez de las compuertas AND y OR? Porque son fáciles de fabricar con componentes electrónicos y son las compuertas básicas empleadas en todas las familias de lógica de CI.

### 3.3.2. Ejercicios conceptuales

Los ejercicios conceptuales son preguntas prácticas que contienen los temas del capítulo de álgebra booleana.

- ¿Qué diferencia hay entre una salida que tiene un nivel alto de nivel de compuertas con una que tiene un nivel bajo del nivel de compuertas? La salida que tiene un nivel alto tendrá un mayor retraso con respecto a la que tiene un nivel bajo. Esto porque tiene que pasar por una mayor cantidad de componentes electrónicos.

- Se tiene una tabla de verdad de 3 variables ¿Cuál sería la respuesta más simplificada si los únicos minterminos de interés son el cero y el uno y los minterminos cuatro y cinco son minterminos indiferentes (los demás minterminos son igual a cero lógico)? Los valores de interés son uno lógico. Se procede a realizar la tabla de verdad con los valores descritos y hacer el mapa de Karnaugh de tres variables. Luego se puede obtener la función.

Figura 150. **Ejercicio conceptual 2 capítulo 3**

x	y	z	F
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	X
1	0	1	X
1	1	0	0
1	1	1	0

x\yz	00	01	11	10
0	x'y'z'	x'y'z	x'yz	x'yz'
1	xy'z'	xy'z	xyz	xyz'

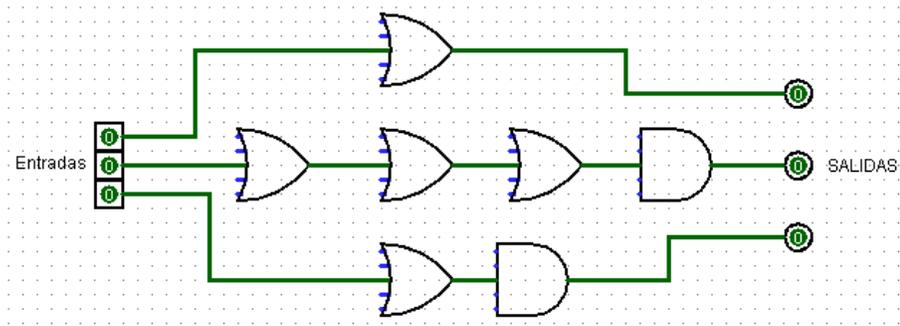
x\yz	00	01	11	10
0	1	1	0	0
1	X	X	0	0

$$F = y$$

Fuente: elaboración propia, empleando Excel.

- ¿Cuál es el nivel de compuertas lógicas respecto de las salidas del siguiente circuito? El nivel de compuertas lógicas respecto de la salida es de cuatro, esto porque al menos una de las entradas tiene que pasar por cuatro compuertas lógicas

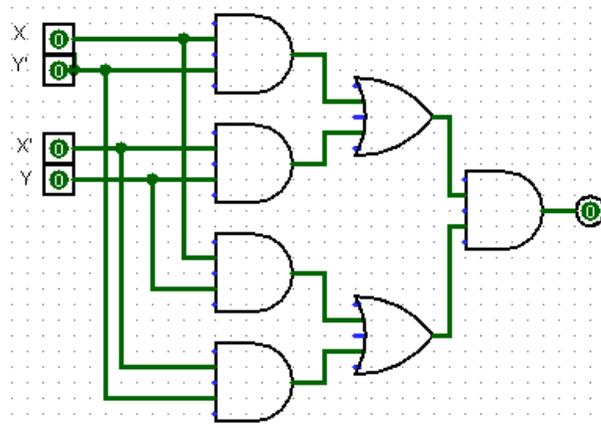
Figura 151. Respuesta del ejercicio conceptual 2 capítulo 3



Fuente: elaboración propia, empleando Logisim.

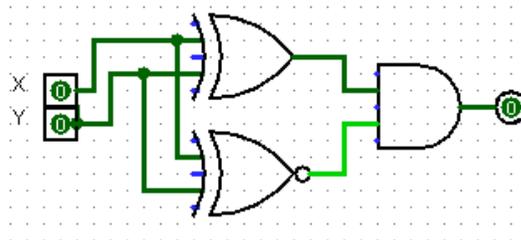
- Dibujar el circuito equivalente de la figura 152 utilizando compuertas XOR para el siguiente circuito.

Figura 152. Circuito de ejercicio conceptual 4 capítulo 3



Fuente: elaboración propia, empleando Logisim.

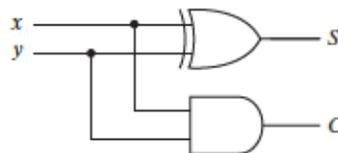
Figura 153. **Respuesta del ejercicio conceptual 4 capítulo 3**



Fuente: elaboración propia, empleando Logisim.

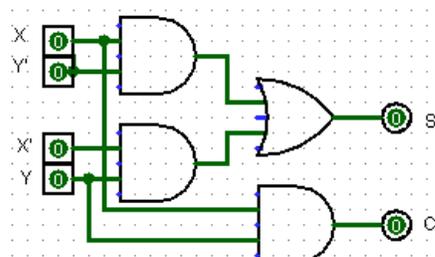
- ¿De qué otra forma se puede representar el circuito de la figura 155? La forma solicitada no es la más simplificada.

Figura 154. **Circuito de ejercicio conceptual 5 capítulo 3**



Fuente: MANO, Morris. *Diseño digital*. p 120.

Figura 155. **Circuito equivalente del ejercicio conceptual 5 capítulo 3**



Fuente: elaboración propia, empleando Logisim.

### 3.4. Preguntas y ejercicios conceptuales de la lógica combinacional

Al finalizar el capítulo de lógica combinacional el alumno deberá ser capaz de resolver las preguntas y ejercicios propuestos a continuación.

#### 3.4.1. Preguntas conceptuales

Las preguntas conceptuales son preguntas directas que contienen los temas del capítulo de lógica combinacional.

- ¿En función de qué está la salida de un circuito combinacional? En función de las entradas del circuito.
- Cuando se tienen  $n$  variables de entrada en un circuito combinacional, ¿Cuántas combinaciones de entradas binarias se tienen en un circuito? Se tienen  $2^n$  combinaciones.
- ¿Cuál es la diferencia entre un semisumador y un sumador completo? El semisumador, por limitaciones de hardware, solo puede sumar dos bits y el sumador completo puede realizar la suma aritmética de hasta tres bits.
- ¿Qué problemática resuelve el circuito de la propagación del acarreo? Permite que el valor  $S$  y el valor  $C$  lleguen al mismo tiempo, dado que de otro modo llegarían en tiempos distintos por los retrasos que pueda generar en pasar una cantidad  $n$  de compuertas lógicas.
- ¿Por qué es más conveniente un circuito sumador-restador binario, a comparación de los circuitos: sumador y restador binario por aparte? Al utilizar el sumador y restador binario se puede llegar a ahorrar una gran cantidad de compuertas lógicas y, por ende, haciéndolo un método más eficiente.
- ¿Por qué hay que sumar seis a la conversión de binario y BCD? ¿A cuáles números hay que sumarlo? Para representar el número en binario en la

codificación en BCD. Al sumar seis genera el acarreo necesario para que la década siguiente aumente en uno, y hace que el valor de la década actual se muestre en un valor acorde a un valor en el sistema decimal. Hay que sumarlos a los números mayores a nueve porque los números menores no cambian con respecto a la codificación de binario natural

- ¿Qué es un decodificador? Es un circuito combinacional que convierte información binaria de  $n$  salidas de entrada a un máximo de  $2^n$  líneas de salida distintas.
- Teniendo un decodificador de  $n$  entradas ¿Es necesario utilizar las  $2^n$  salidas? No es necesario, se pueden utilizar solo los minitérminos de interés.
- ¿Qué es una entrada habilitadora? Es una entrada que permite habilitar la salida del circuito.
- ¿Qué es un circuito decodificador/desmultiplexor? Es un circuito decodificador con entradas de habilitación.
- ¿Cuál es la diferencia entre un codificador con prioridad y un codificador tradicional y que problemática resuelve el codificador con prioridad? El codificador con prioridad permite seleccionar una salida en base a prioridades establecidas en el diseño del circuito. Permite resolver problemas con ambigüedad, y estar seguros de cuál será la salida con ciertas entradas.
- ¿Qué es la multiplicación? Es la capacidad de seleccionar la salida entre dos o más entradas con la ayuda de in circuito de selección.

### **3.4.2. Ejercicios conceptuales**

Los ejercicios conceptuales son preguntas prácticas que contienen los temas del capítulo de lógica combinacional.

- Demostrar que los circuitos de la figura 56 y la figura 57 son equivalentes. Utilice algebra de Boole para su respuesta

$$S = x'y'z + x'yz' + xy'z' + xyz$$

$$z(x'y' + xy) + z'(x'y + xy')$$

$$z(x \oplus y)' + z'(x \oplus y)$$

$$z \oplus (x \oplus y)$$

$$C = xy + xz + yz$$

$$z(x + y) + xy$$

- Sabiendo que un sumador binario se construye en base a sumadores completos, ¿qué pasaría si se sustituyera el sumador completo con un semisumador del bit menos significativo del sumador binario? ¿habría algún cambio? Explique su respuesta. Hay que recordar que la diferencia entre un sumador completo y el semisumador es que el sumador completo tiene la capacidad de sumar tres bits y el semisumador solo dos. Al no dejar una entrada de acarreo en el bit menos significativo, impide que este sumador binario pueda utilizarse junto a otros sumadores y de ese modo sumar dos números de mayor cantidad de bits.
- Explicar el funcionamiento del comparador de magnitudes.

Para ejemplificar se hará un comparador de magnitudes con dos números de cuatro bits (A y B).

$$A = A_3A_2A_1A_0$$

$$B = B_3B_2B_1B_0$$

Donde  $A_i$  y  $B_i$  son cada dígito del número de cuatro bits. Para determinar si  $A = B$ , se necesita que  $A_i = B_i$  sean iguales. Esto se puede determinar con una compuerta XOR.

$$x_i = A_i \oplus B_i = A_i B_i + A'_i B'_i$$

Donde  $x_i = 1$  cuando ambos números sean cero o uno. Ahora, esto sería por cada uno de los dígitos, y como se necesita que se cumplan todas, las salidas  $x_i$  irán conectadas hacia una compuerta AND.

$$(A = B) = (A_3 B_3 + A'_3 B'_3)(A_2 B_2 + A'_2 B'_2)(A_1 B_1 + A'_1 B'_1)(A_0 B_0 + A'_0 B'_0)$$

Para determinar si  $A > B$  se inspeccionará las magnitudes relativas de pares de dígitos significativos, comenzando de la posición más significativa. Si los dos dígitos son iguales se comparará el siguiente par de dígitos menos significativos. Esta comparación continúa hasta encontrar un par de dígitos distintos. Si el dígito correspondiente de A es 1 y el de B es 0, se concluye que  $A > B$ . Si el dígito correspondiente de A es 0 y el de B es 1, se concluye que  $A < B$ .

Figura 156. **Ecuaciones del ejercicio conceptual 3 capítulo 4**

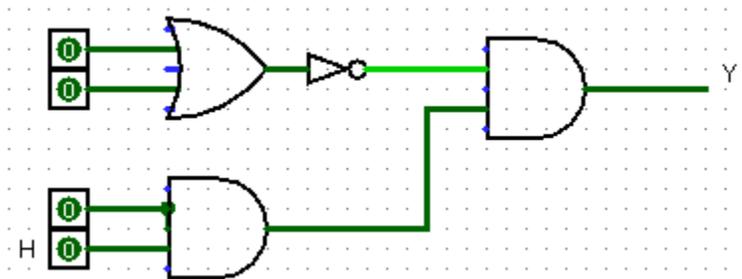
$$\begin{aligned} (A > B) &= A_3 B'_3 + x_3 A_2 B'_2 + x_3 x_2 A_1 B'_1 + x_3 x_2 x_1 A_0 B'_0 \\ (A < B) &= A'_3 B_3 + x_3 A'_2 B_2 + x_3 x_2 A'_1 B_1 + x_3 x_2 x_1 A'_0 B_0 \end{aligned}$$

Fuente: MANO, Morris. *Diseño digital*. p 124.

- Basándose en el comportamiento del circuito de la figura 158, ¿Qué tipo de entrada sería la entrada H? Como se ve la entrada H está conectada hacia

una compuerta AND, cuya salida es hacia otra compuerta AND. Se puede ver que, si H es igual a cero, la salida Y queda inhabilitada, por ello se dice que H es una compuerta habilitadora.

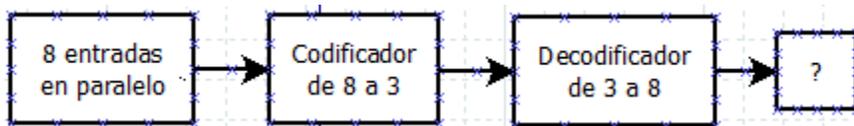
Figura 157. **Circuito del ejercicio conceptual 4 capítulo 4**



Fuente: elaboración propia, empleando Logisim.

- Con respecto al diagrama de bloques de la figura 159, ¿Cuál sería el resultado final? En base a su respuesta anterior ¿Con que circuito equivalente lo reemplazaría? Dado a que es un circuito codificador conectado a un circuito decodificador el resultado sería las entradas al circuito codificador (8 salidas en paralelo). Es decir, que daría lo mismo incluir el codificador y el decodificador. El circuito equivalente serían cables conectados directamente a las salidas del circuito.

Figura 158. **Diagrama de bloques del ejercicio conceptual 5 capítulo 4**



Fuente: elaboración propia, empleando Dia.

### 3.5. Preguntas y ejercicios conceptuales de la lógica secuencial

Al finalizar el capítulo de lógica secuencial el alumno deberá ser capaz de resolver las preguntas y ejercicios propuestos a continuación.

#### 3.5.1. Preguntas conceptuales

Las preguntas conceptuales son preguntas directas que contienen los temas del capítulo de lógica secuencial.

- ¿Qué es un circuito secuencial? Es un circuito el cual su salida depende de sus entradas y de la información que tengan almacenada sus dispositivos de almacenamiento.
- ¿Qué son los elementos de almacenamiento? Son dispositivos capaces de guardar información binaria.
- ¿Qué es un circuito secuencial sincrónico? Es un sistema que su comportamiento se define conociendo sus señales en instantes discretos.
- ¿Qué es un circuito secuencial asincrónico? Es un sistema que su comportamiento se define en cualquier instante del tiempo, por lo que no está definido por ningún tren de pulsos.
- ¿Qué es un *flip-flop*? Es un dispositivo con la capacidad de mantenerse en uno de dos estados posibles durante un tiempo indefinido.
- ¿Cuál es la diferencia principal entre un *latch* y un *flip-flop*? Un *latch* no tiene entrada de reloj, y un *flip-flop* si cuenta con ella.
- ¿Para qué sirve una entrada de control en un *latch*? Ayuda a controlar cuando se desea que el circuito cambie de estado. Si esta entrada no es activada, el *latch* no cambiara de estado.

- En un *latch* SR ¿Qué es un estado indefinido? Es cuando no se sabe cuál será el estado siguiente del *latch*. Si es un *latch* SR con compuertas NOR, esto ocurrirá cuando  $S=R=0$ ; si es un *latch* SR con compuertas NAND, esto ocurrirá cuando  $S=R=1$ . Las entradas de los *latches* afectaran al siguiente estado y no al estado actual.
- ¿Cuál es la diferencia entre un *latch* SR y un *latch* D? El *latch* D no puede llegar a caer a un estado indefinido por su compuerta NOT en la entrada D.
- ¿Qué son las entradas asíncronas? Son entradas que no dependen del reloj para poder funcionar. Permiten establecer o restablecer *flip-flops* en un momento deseado.
- ¿Qué es una ecuación de estado? Es una expresión algebraica que especifica el siguiente estado de un *flip-flop*. Esto en función de sus entradas y estados actuales de los *flip-flops*.
- ¿Qué es una tabla de estados? Es una tabla compacta donde se muestra los posibles estados siguientes en función de las entradas y los estados actuales de los *flip-flops*.
- ¿Qué es un diagrama de estados? Es una representación gráfica de las transiciones de los circuitos secuenciales en base a las entradas y los estados actuales de los *flip-flops* que componen el circuito.
- ¿Qué es la reducción de estados? Es un procedimiento que puede reducir la cantidad de estados en una tabla de estados, pero no afecta en las salidas del circuito.
- ¿Qué es una tabla de excitación? Es una relación funcional entre la tabla de estados y las ecuaciones de entrada, que dará una vista rápida y directa del comportamiento del *flip-flop*.

### 3.5.2. Ejercicios conceptuales

Los ejercicios conceptuales son preguntas prácticas que contienen los temas del capítulo de lógica secuencial.

- Usted está diseñando un circuito secuencial, si su circuito es sincrónico ¿qué dispositivos de almacenamiento utilizaría? Y si su circuito asincrónico ¿Cuál utilizaría? Explique por qué. Si el circuito por diseñar es sincrónico, es recomendable utilizar *flip-flops*, porque estos poseen una entrada de reloj. Al diseñar un circuito asincrónico es necesario utilizar *latches* como dispositivos de almacenamiento porque estos no cuentan con ninguna entrada de reloj.
- Entre los *flip-flops* D, JK y T ¿Cuál elegiría para realizar su diseño? Explique su respuesta. Dependerá del circuito que se desee realizar. Ningún *flip-flop* es superior al otro, simplemente que cada uno tiene su función en específico. El *flip-flop D* es fácil de implementar dado a que no es necesaria una tabla de excitación. El *flip-flop JK* puede llegar a ser muy versátil. Por parte del *flip-flop T*, es muy útil para el diseño de contadores binarios.
- Al aplicar un tren de pulsos a la entrada de control de un *latch* ¿Actuaría de la misma manera que un *flip-flop*? Explique su respuesta. Al conectar un tren de pulsos en un *latch*, se tendría como resultado que cada vez que la entrada de control sea igual a uno, podría realizarse un cambio en el estado del *latch* (respuesta a nivel positivo). En cambio, un *flip-flop* solo puede cambiar su estado cuando existe un cambio de estado en el reloj (respuesta al borde positivo o negativo). Por ello un *latch* nunca tendrá el mismo comportamiento que un *flip-flop*.

### 3.6. Preguntas y ejercicios conceptuales de los registros y contadores

Al finalizar el capítulo de registros y contadores el alumno deberá ser capaz de resolver las preguntas y ejercicios propuestos a continuación.

#### 3.6.1. Preguntas conceptuales

Las preguntas conceptuales son preguntas directas que contienen los temas del capítulo de registros y contadores.

- ¿Qué es un registro? Es un grupo de *flip-flops* y compuertas que efectúan una transición. Los *flip-flops* contiene la información binaria y las compuertas determinan como se transfiere la información al registro.
- ¿Qué es un registro con carga paralela? Es un registro que se carga simultáneamente con un pulso de reloj común. Esto quiere decir que ocurre una transferencia de registro de todos los *flip-flops* al mismo tiempo con una misma señal. Todas las entradas C de los *flip-flops* están conectadas al mismo reloj.
- ¿Qué es un registro de desplazamiento? Es un registro capaz de desplazar su información binaria en una dirección o en la otra. La configuración lógica de un registro de desplazamiento consiste en un grupo de *flip-flops* conectados en cadena, con la salida de un *flip-flop* conectada a la entrada del siguiente *flip-flop*.
- ¿Qué es una entrada de control de desplazamiento? Cuando se habla de registros de desplazamiento, surge la problemática que los datos siguen ingresando al registro y, por ser seriales, reemplazarán los registros más antiguos. Esto se resuelve con una entrada de control de desplazamiento en

un *flip-flop*. Esta será una compuerta AND que permitirá hacer un desplazamiento de registros solo cuando esté activa.

- ¿Qué es un registro de desplazamiento universal? ¿Cuáles son sus características? Es un registro que puede almacenar información de forma paralela y de forma bidireccional en serial. Cuando un registro es unidireccional puede almacenar información ya sea de derecha a izquierda y de izquierda a derecha. Un registro bidireccional permite almacenar información de izquierda a derecha y de derecha a izquierda. Estos registros cuentan con las siguientes características:
  - Una entrada de restablecimiento.
  - Una entrada de reloj.
  - Un control de desplazamiento a la derecha para habilitar el desplazamiento a la derecha.
  - Un control de desplazamiento a la izquierda para habilitar el desplazamiento a la izquierda.
  - Un control de carga en paralelo para habilitar la transferencia en paralelo.
  - Una cantidad de  $n$  líneas de salida en paralelo.
  - Un control de estado para dejar los registros con los estados actuales.
- ¿Qué es un contador? Es básicamente un registro que pasa por una sucesión predeterminada de estados. Las compuertas del contador están conectadas de tal manera que producen la sucesión prescrita de estados binarios.

- ¿Qué es un contador de rizo? Son dispositivos contadores que tienen conectados los *flip-flops* en forma asíncrona, es decir, que no tienen conectadas las entradas de reloj en paralelo. En un contador de rizo la transición de salida del *flip-flop* sirve como disparador de otros *flip-flops*.
- ¿Qué es un contador binario de rizo? Un contador binario de rizo es un contador donde los *flip-flops* están conectados en serie y cuentan de manera ascendente o descendente (según configuración).
- ¿Qué es un contador sincrónico? En un contador sincrónico las salidas C de todos los *flip-flops* están conectadas al reloj común. Un solo contador dispara todos los *flip-flops* simultáneamente en vez de hacerlo uno por uno sucesivamente como en los contadores de rizo.
- ¿Qué problemática resuelve el contador binario de carga paralela? Muchas veces es necesario iniciar el conteo desde un valor que no fue pre establecido con anterioridad. Es decir, un valor el cual no haya sido incluido en el diseño, pero que se encuentre en el rango que el contador pueda soportar (que dependerá en la cantidad de bits del contador). Esto lo permite el contador binario con carga paralela.
- ¿Qué es un contador con autocorrección? Es un contador que tiene la capacidad de regresar a alguno de sus estados conocidos, si por error callo a un estado no conocido. Esto pudo ser causado por algún fallo en el sistema o de mal uso por el usuario.
- ¿Qué es un contador de estados no utilizados? Son contadores los cuales tienen la peculiaridad de que sus estados no son necesariamente sucesivos ni se utilizan todos los estados que los *flip-flops* pueden otorgar. Los estados de interés dependerán del diseño del usuario.

- ¿Qué es un contador anular? El contador anular o de anillo es un circuito en donde solo un *flip-flop* está establecido y todos los demás *flip-flops* están restablecidos. Por ejemplo, si se tienen tres *flip-flops* se tendría cuatro estados los cuales serían: 000, 100, 010, 001. Se incluyó el estado 000 como estado inicial.

### 3.6.2. Ejercicios conceptuales

Los ejercicios conceptuales son preguntas prácticas que contienen los temas del capítulo de registros y contadores.

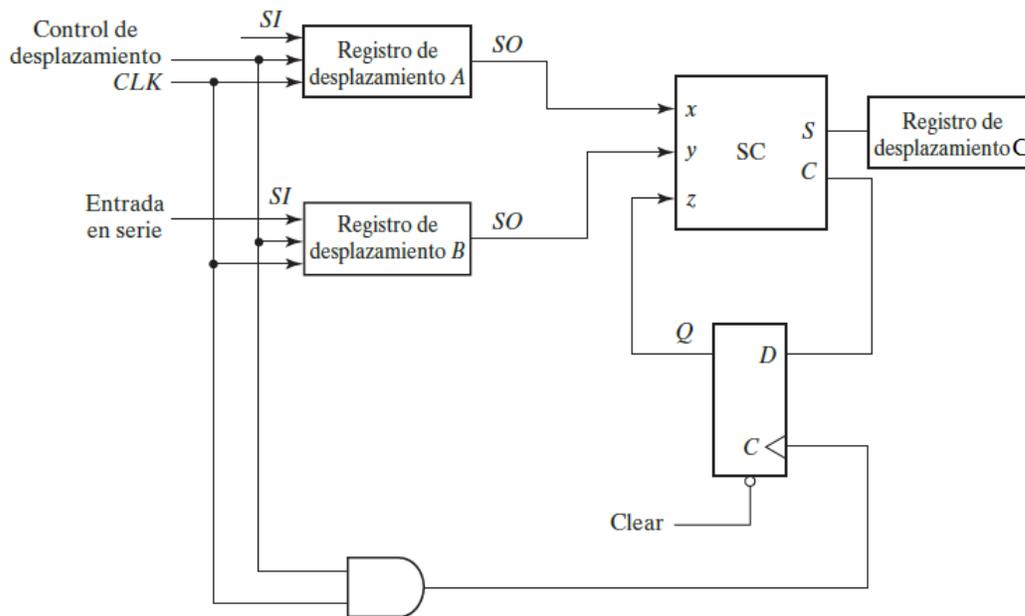
- ¿Cuál es la función del registro de desplazamiento B (figura 160)? Es la de almacenar la información que el registro de desplazamiento A contiene. Dado a que la información en el registro A sigue ingresando, de no guardarse, se perderían los datos.

Figura 159. Diagrama del ejercicio conceptual 1 capítulo 6



Fuente: MANO, Morris. *Diseño digital*. p 221.

Figura 160. **Circuito del ejercicio conceptual 2 capítulo 6**



Fuente: MANO, Morris. *Diseño digital*. p 223.

- Explique el funcionamiento del sumador en serie (figura 161). Está compuesto por dos registros de desplazamiento, una compuerta AND para control, un sumador completo y un *flip-flop D* para el acarreo. Los números de ambos sumandos serán ingresados de forma serial por las entradas SI de los registros de desplazamiento. La suma es realizada bit por bit y los resultados se almacenarán en un registro de desplazamiento C. Esto siempre será controlado por la entrada de reloj CLK. El acarreo es manejado por un *flip-flop D*, el cual al momento de haber un acarreo lleva ese uno a la siguiente suma del siguiente par de bits. La compuerta AND sirve para control para saber cuándo sumar.

## **4. EXÁMENES CORTOS DE CONCEPTOS TEÓRICOS DEL CURSO ELECTRÓNICA 3**

Los exámenes de este capítulo son una propuesta con el fin de poder comprobar los conocimientos aprendidos del curso de Electrónica 3. Cada uno tendrá una dificultad media y podrán ser realizados en un tiempo no mayor de 50 minutos. Será decisión del catedrático darles o no una ponderación.

### **4.1. Exámenes de sistemas de numeración**

A continuación, se presentan una cantidad de tres exámenes propuestos del capítulo sistemas de numeración.

#### **4.1.1. Examen uno**

Este es el primer examen propuesto y está compuesto por una cantidad de siete enunciados.

- ¿Qué indica el subíndice de un número y porque no se utiliza en el sistema de numeración decimal? El subíndice indica cual es el sistema numérico que se utilizó para representar esta cantidad. No es utilizado en el sistema de numeración decimal porque este sistema es utilizado en forma cotidiana. Si el número no presenta ningún subíndice, se sobreentiende que se utilizó el sistema de numeración decimal para representar dicha cantidad.

- ¿Cuántos dígitos binarios corresponden a un dígito octal? ¿Y a uno hexadecimal? Corresponden 3 y 4 dígitos binarios respectivamente.
- ¿De qué necesidad surgieron los sistemas de numeración octal y hexadecimal? Algunas veces es necesario que el operador se comunique con la computadora a través de números binarios. Aprovechando la relación que hay entre los tres sistemas de numeración, es más sencilla la comprensión de números hexadecimales y octales a comparación de un número binario.

- Escribir el número 3561.75 en el sistema de numeración octal.

$$a_4 * 8^4 + a_3 * 8^3 + a_2 * 8^2 + a_1 * 8^1 + a_0 * 8^0 + a_{-1} * 8^{-1}$$

$$6 * 8^3 + 7 * 8^2 + 5 * 8^1 + 1 * 8^0 + 6 * 8^{-1} = 3561.75$$

$$(6751.6)_8$$

- Convertir el número 325.33 a su equivalente en binario

$$325 \div 2 = 162 \quad mod = 1$$

$$162 \div 2 = 81 \quad mod = 0$$

$$81 \div 2 = 40 \quad mod = 1$$

$$40 \div 2 = 20 \quad mod = 0$$

$$20 \div 2 = 10 \quad mod = 0$$

$$10 \div 2 = 5 \quad mod = 0$$

$$5 \div 2 = 2 \quad mod = 1$$

$$2 \div 2 = 1 \quad mod = 0$$

$$33 \div 2 = 16 \quad mod = 1$$

$$16 \div 2 = 8 \quad mod = 0$$

$$8 \div 2 = 4 \quad mod = 0$$

$$4 \div 2 = 2 \quad mod = 0$$

$$2 \div 2 = 1 \quad \text{mod} = 0$$

$$325.33 = 101000101.100001_2$$

- Convertir el número 18667.55 a su equivalente en hexadecimal

$$18667 \div 16 = 1166 \quad \text{mod} = B$$

$$1166 \div 16 = 72 \quad \text{mod} = E$$

$$72 \div 16 = 4 \quad \text{mod} = 8$$

$$55 \div 16 = 3 \quad \text{mod} = 7$$

$$18667.55 = 48EB.37_{16}$$

- Convertir del sistema de numeración binario al sistema de numeración hexadecimal.

$$(10110001101011.111100000110)_2$$

Tabla LXXII. **Tabla del enunciado 7 capítulo 1 examen 1**

Binario	Hexadecimal
10	2
1100	C
1100	6
1011	B
.	.
1111	F
0000	0
110	2

Fuente: elaboración propia, empleando Excel.

Parte antes del punto:  $(10110001101011)_2$

Parte después del punto:  $(111100000110)_2$

Parte antes del punto agrupada en 4 dígitos:  $(10\ 1100\ 0110\ 1011)_2$

Parte después del punto agrupada en 4 dígitos:  $(1111\ 0000\ 0110)_2$

$(2C6B.F02)_{16}$

#### 4.1.2. Examen dos

Este es el segundo examen propuesto y está compuesto por una cantidad de siete enunciados.

- ¿Cuál es la aplicación (no un dispositivo) de la codificación BCD? Una representación en decimal de un número binario.
- ¿Por qué razones no es utilizada la codificación BCD para el procesamiento de datos? Porque utiliza una mayor cantidad de bits que el código binario natural y que las operaciones aritméticas son más complicadas que en la codificación binario natural.
- ¿Cuál es la principal ventaja del código Gray con el código BCD? La principal ventaja que existe en el código Gray es, que entre dos números sucesivos, existe un cambio de solo un bit.
- Realizar la siguiente suma:  $(10011)_2 + (11100)_2$

$$\begin{array}{r} \phantom{+} 1 \\ 1\ 1\ 0\ 1\ 1 \\ + 1\ 1\ 1\ 0\ 0 \\ \hline 1\ 1\ 0\ 1\ 1\ 1 \end{array}$$

- Realizar la siguiente resta:  $(11100)_2 - (11011)_2$

$$\begin{array}{r} 111100 \\ - 110011 \\ \hline 001001 \end{array}$$

- Comprobar el complemento a la base disminuida del número  $(1000110110)_2$ .

$$r = 2$$

$$n = 10$$

$$N = 1000110110$$

$$\begin{aligned} (2^{10} - 1) - 1000110110 &= 10000000000 - 1 - 1000110110 \\ &= 111111111 - 1000110110 = 0111001001 \end{aligned}$$

- Obtener el complemento de la base de  $(10111010)_2$  y de  $(01100111)_2$ .

Para  $(10111010)_2$  su complemento a la base es  $(01000110)_2$

Para  $(01100111)_2$  su complemento a la base es  $(10011001)_2$

#### 4.1.3. Examen tres

Este es el tercer examen propuesto y está compuesto por una cantidad de siete enunciados.

- ¿Por qué la codificación de detección de errores no puede detectar una cantidad par de errores? Al tener una cantidad par de errores, la cantidad de unos en la trama sería de paridad igual a la paridad que se toma como correcta.
- ¿Por qué razones no es utilizada la codificación BCD para el procesamiento de datos? Porque utiliza una mayor cantidad de bits que el código binario natural y que las operaciones aritméticas son más complicadas que en la codificación binario natural.

- ¿Cuál es la región de transición de las compuertas lógicas? De 1V a 5V.
- ¿Cuáles son las compuertas lógicas? YES/Buffer/EQUAL, OR, AND, XOR/OR EXCLUSIVA/OREX, NOT, NOR, NAND, XNOR.
- Describir el código Gray. En este código existe solo un cambio de un bit entre dos números sucesivos. Los códigos que tienen estas características generalmente sus aplicaciones se extienden a los campos de la instrumentación, transductores, convertidores analógicos/digital, en codificadores de desplazamiento lineal y angular.
- Complete la tabla LXXIII.

Tabla LXXIII. **Tabla del enunciado 6 capítulo 1 examen 3**

Binario	Decimal	
	Con signo	Sin signo
1001		
1111		
1010		
1011		
0100		
0011		
0111		
1101		

Fuente: elaboración propia, empleando Excel.

- ¿Qué es un código binario? Es una representación de un texto y/o valores numéricos con números binarios. Solo se utilizarán ceros y unos para representar el texto y/o cantidades deseadas.

Tabla LXXIV. **Tabla del resultado del enunciado 6 capítulo 1 examen 3**

Binario	Decimal	
	Con signo	Sin signo
1001	-1	9
1111	-7	15
1010	-2	10
1011	-3	11
0100	+4	4
0011	+3	3
0111	+7	7
1101	-5	13

Fuente: elaboración propia, empleando Excel.

## 4.2. Exámenes de álgebra booleana

A continuación, se presentan una cantidad de tres exámenes propuestos del capítulo álgebra booleana.

### 4.2.1. Examen uno

Este es el primer examen propuesto y está compuesto por una cantidad de cinco enunciados.

- ¿Qué establece el principio de dualidad? Establece que toda expresión algebraica que puede deducirse de los postulados del algebra booleana seguirá siendo válida si se intercambian los operadores y los elementos de identidad.
- ¿Cuáles son las 3 formas de expresar una función booleana? Expresión algebraica, tabla de verdad y sumatoria.

- ¿Cuál es la finalidad de simplificar las expresiones algebraicas en el álgebra de Boole? Utilizar la menor cantidad de hardware posible obteniendo los mismos resultados.
- ¿Qué información se puede obtener de la siguiente sumatoria?

$$F(M, N, L) = \sum (0,1,3,7,8,10,16,20,25,30,35,60)$$

Que la función F que depende de las variables M, N, y L, es igual a uno en los reglones: cero, uno, tres, siete, ocho, diez, dieciséis, veinte, veinticinco, treinta, treintaicinco y sesenta.

- Simplificar las siguientes expresiones con la ayuda de los teoremas y postulados del álgebra de Boole.  $F_2 = x'y'z + x'yz + xy'$ ,  $xy + x'z + yz$ ,  $(x + y)(x' + z)(y + z)$ .

$$F_2 = x'y'z + x'yz + xy$$

$$F_2 = z(x'y' + x'y) + xy$$

$$F_2 = z(x'(y' + y)) + xy$$

$$F_2 = zx' + xy$$

$$xy + x'z + yz$$

$$xy + x'z + yz(x + x') = xy + x'z + yzx + yzx'$$

$$xy(1 + z) + x'z(1 + y) = xy(1) + x'z(1)$$

$$xy + x'z$$

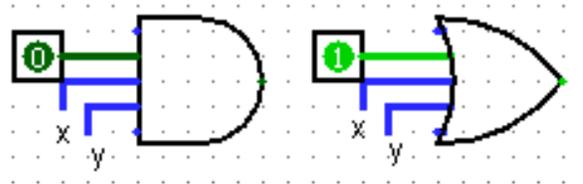
$$\begin{aligned}
& (x + y)(x' + z)(y + z) \\
& (x + y)(x'y + xz + zy + zz) \\
& (x + y)(x'y + xz + zy + zz) \\
& (x + y)(x'y + xz + zy + z) \\
& (x + y)(x'y + z(1 + x + y)) \\
& (x + y)(x'y + z) \\
& xz + xx'y + yz + x'yy = xz + yz + x'y \\
& xz + yz(x + x') + x'y = xz + x'y + yzx + yzx' \\
& xz(1 + y) + x'y(1 + y) = xz + x'y
\end{aligned}$$

#### 4.2.2. Examen dos

Este es el segundo examen propuesto y está compuesto por una cantidad de seis enunciados.

- ¿Cuál es el complemento de una función? En un circuito lógico ¿Cómo se obtiene el complemento de una función? El complemento de una función F es F' y se obtiene intercambiando ceros por unos y unos por ceros en la salida de la función F. Colocando una compuerta NOT en la salida F del circuito lógico se puede obtener nuestra función F'.
- ¿Cuál es la diferencia entre los maxitérminos y los minitérminos? Los minitérminos son productos donde las variables son cada uno de los factores y los maxitérminos son sumas que sus sumandos son cada una de las variables.

Figura 161. Diagrama del enunciado 3 capítulo 2 examen 2



Fuente: elaboración propia, empleando Logisim.

- ¿Qué principio relaciona los dos circuitos de la figura 162? El principio de dualidad. La expresión que representa al primer diagrama es:  $x * y * 0 = 0$  y la expresión que representa el segundo diagrama es:  $1 + x + y = 1$ .
- ¿Cuál es la diferencia entre producto de sumas y suma de productos? El producto de suma está compuesto por términos AND que forman un término OR y la suma de productos son términos OR que forman un término AND.
- ¿Cuál es la diferencia entre lógica positiva o lógica negativa? La lógica positiva es aquella que el uno lógico es representado por el valor más alto de voltaje, y la lógica negativa es aquella que el uno lógico es representado del valor más bajo de voltaje.
- Comprobar la equivalencia de las siguientes funciones:

$$AB'C + A'BC + A'B'C + A'B'C' + AB'C' = B' + A'C$$

$$AB'C + A'BC + A'B'C + A'B'C' + AB'C'$$

$$AB'C + A'BC + A'B'(C + C') + AB'C'$$

$$AB'C + A'BC + A'B' + AB'C'$$

$$AB'(C + C') + A'BC + A'B'$$

$$B'(A + A') + A'BC$$

$$B' + A'BC$$

$$B' + A'C$$

### 4.2.3. Examen tres

Este es el tercer examen propuesto y está compuesto por una cantidad de siete enunciados.

- ¿Qué es un CI? Es un cristal semiconductor de silicio, llamado chip, que contiene los componentes electrónicos para construir compuertas digitales.
- ¿Cuándo es ventajoso utilizar los CI ECL? Resulta ventajoso en sistemas que deben operar a alta velocidad.
- ¿Cuándo es ventajoso utilizar los CI MOS? Resulta conveniente para circuitos que se requieren una densidad elevada de componentes.
- ¿Cuándo es ventajoso utilizar CI CMOS? Resulta ventajoso cuando se requiere que el sistema posea un bajo consumo de energía.
- ¿Cuál es el valor de la salida F del circuito de la figura 163? Explique su respuesta. F siempre será igual a uno porque:

$$F = X + X' + Y + Z + W + U + V + N$$

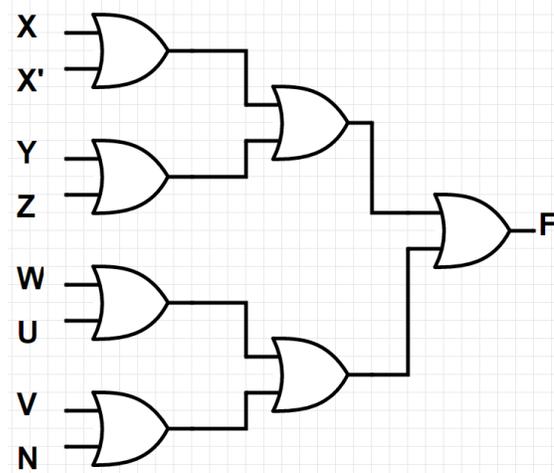
$$F = (X + X') + Y + Z + W + U + V + N$$

$$F = 1 + Y + Z + W + U + V + N$$

$$F = 1$$

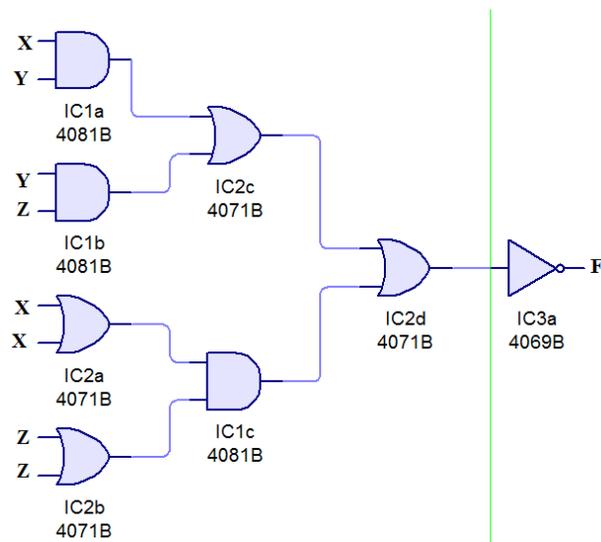
- Si se tiene un circuito el cual está conformado por 8 compuertas ¿Qué tipo de CI utilizaría (deducir con respecto al nivel de integración)? Dado a que se deduce por medio de los niveles de integración, el tipo de CI es SSI.
- Encontrar la tabla de verdad, la expresión algebraica y la sumatoria del diagrama lógico de la figura 164. La expresión algebraica escribirla por medio de minitérminos.

Figura 162. Diagrama del enunciado 6 capítulo 2 examen 3



Fuente: elaboración propia, empleando Logisim.

Figura 163. Diagrama del enunciado 7 capítulo 2 examen 3



Fuente: elaboración propia, empleando Logisim.

Figura 164. Respuesta del enunciado 7 capítulo 2 examen 3

x	y	z	F
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	0

Fuente: elaboración propia, empleando Excel.

$$F = (xy + yz + xz)'$$

$$F = (xy)'(yz)'(xz)'$$

$$F = (x' + y')(y' + z')(x' + z')$$

$$F = x'y' + x'z' + y'y' + y'z'(x' + z')$$

$$F = x'x'y' + x'x'z' + x'y'y' + x'y'z' + x'y'z' + x'z'z' + y'y'z' + y'z'z'$$

$$F = x'y' + x'z' + x'y' + x'y'z' + x'y'z' + x'z' + y'z' + y'z'$$

$$F = x'y' + x'z' + x'y'z' + y'z'$$

$$F(x, y, z) = \sum (0,1,2,3,4)$$

#### 4.3. Exámenes de minimización de funciones booleanas

A continuación, se presentan una cantidad de tres exámenes propuestos del capítulo funciones booleanas.

### 4.3.1. Examen uno

Este es el primer examen propuesto y está compuesto por una cantidad de siete enunciados.

- ¿Qué es un mapa de Karnaugh? Es una herramienta matemática que ayuda a simplificar funciones booleanas.
- ¿Cuándo es necesaria la utilización de mapas de Karnaugh? Luego de haber simplificado la función booleana por medio de los postulados y teoremas del álgebra de Boole.
- ¿Por qué es recomendable resolver los mapas de Karnaugh de mayor de 5 variables por medio de software? La cantidad de minterminos es  $2^5 = 32$ , y al hacerlo a mano el resultado está muy susceptible a errores.
- Al momento de realizar un mapa de Karnaugh ¿Qué codificación utiliza para ordenar los minterminos? El código Gray porque al hacer el mapa de Karnaugh se busca que, entre los cuadros adyacentes, solo una variable cambie de estado.
- ¿Qué diferencia hay entre una salida que tiene un nivel alto de nivel de compuertas con una que tiene un nivel bajo del nivel de compuertas? La salida que tiene un nivel alto tendrá un mayor retraso con respecto a la que tiene un nivel bajo. Esto porque tiene que pasar por una mayor cantidad de componentes electrónicos. Otro factor es que la que tiene un nivel alto necesitará mayor potencia por la mayor cantidad de componentes electrónicos.
- Para la siguiente tabla de verdad, obtener la expresión más simplificada de la función.

Tabla LXXV. **Tabla del enunciado 6 capítulo 3 examen 1**

x	y	F
0	0	1
0	1	1
1	0	1
1	1	0

Fuente: elaboración propia, empleando Excel.

Figura 165. **Mapa del enunciado 6 capítulo 3 examen 1**

x\y	0	1
0	x'y'	x'y
1	xy'	xy

x\y	0	1
0	1	1
1	1	0

x\y	0	1
0	1	1
1	1	0

x\y	0	1
0	1	1
1	1	0

Fuente: elaboración propia, empleando Excel.

$$F = x' + y'$$

- Ordenar cada uno de los minitérminos con las variables expuestas para el mapa de Karnaugh de tres variables de la figura 167.

Figura 166. **Mapa de Karnaugh del enunciado 7 capítulo 3 examen 1**

x\yz


Fuente: elaboración propia, empleando Excel.

Figura 167. **Respuesta del enunciado 7 capítulo 3 examen 1**

x\yz	00	01	11	10
0	$x'y'z'$	$x'y'z$	$x'yz$	$x'yz'$
1	$xy'z'$	$xy'z$	$xyz$	$xyz'$

Fuente: elaboración propia, empleando Excel.

#### 4.3.2. Examen dos

Este es el segundo examen propuesto y está compuesto por una cantidad de cinco enunciados.

- Se desea obtener un producto de sumas al simplificar una función con la ayuda de mapas de Karnaugh ¿Cuál es el procedimiento? Es muy similar al del mapa de Karnaugh, pero con la diferencia que los valores de interés son los cero. Luego de esto se le aplica al resultado el teorema de DeMorgan.
- ¿Cuándo es conveniente obtener como resultado un producto de sumas? Dependerá de las limitaciones de hardware que se tengan. En el diseño digital a veces es conveniente tener un cierto tipo de compuertas lógicas.
- Cuando se habla de los mapas de Karnaugh, ¿A qué se refiere el término indiferencia? Hace referencia de los minitérminos o maxitérminos que no influyen en el resultado de una función. Tienen tan poca influencia que pueden ser tomados al estado que el usuario desee (ya sea cero o uno).
- Tengo una tabla de verdad de 3 variables ¿Cuál sería la respuesta más simplificada si los únicos minitérminos de interés son el cero y el uno y los minitérminos cuatro y cinco son minitérminos indiferentes (los demás minitérminos son igual a cero lógico)? El valor de interés es un uno lógico. Se procede a realizar la tabla de verdad con los valores descritos y hacer el mapa de Karnaugh de tres variables. Luego se puede obtener la función.

$$F = y$$

Figura 168. **Enunciado 4 capítulo 3 examen 2**

x	y	z	F
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	X
1	0	1	X
1	1	0	0
1	1	1	0

x\yz	00	01	11	10
0	x'y'z'	x'y'z	x'yz	x'yz'
1	xy'z'	xy'z	xyz	xyz'

x\yz	00	01	11	10
0	1	1	0	0
1	X	X	0	0

Fuente: elaboración propia, empleando Excel.

- Ordenar cada uno de los minterminos con las variables expuestas para el mapa de Karnaugh de cuatro variables.

Figura 169. **Mapa de Karnaugh del enunciado 5 capítulo 3 examen 2**

wx\yz


Fuente: elaboración propia, empleando Excel.

Figura 170. **Respuesta del enunciado 5 capítulo 3 examen 2**

wx\yz	00	01	11	10
00	w'x'y'z'	w'x'yz	w'x'yz	w'x'yz'
01	w'xy'z'	w'xy'z	w'xyz	w'xyz'
11	wxy'z'	wxy'z	wxyz	wxyz'
10	wx'y'z'	wx'y'z	wx'yz	wx'yz'

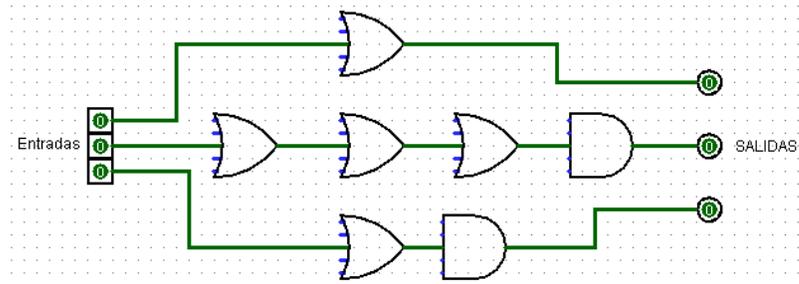
Fuente: elaboración propia, empleando Excel.

### 4.3.3. Examen tres

Este es el tercer examen propuesto y está compuesto por una cantidad de siete enunciados.

- ¿Qué es un minitérmino indiferente? Es un minitérmino el cual su valor no es relevante en la función booleana. Puede ser uno o cero según lo necesite el usuario y es representado por la letra X.
- ¿A qué se le llama salto cuando se habla de niveles de compuertas lógicas? Es el proceso que consiste en pasar a través de una compuerta lógica.
- ¿Cuál es el nivel de compuertas lógicas respecto de las salidas del circuito de la figura 172? El nivel de compuertas lógicas con respecto a la salida es de cuatro, esto porque al menos una de las entradas tiene que pasar por cuatro compuertas lógicas

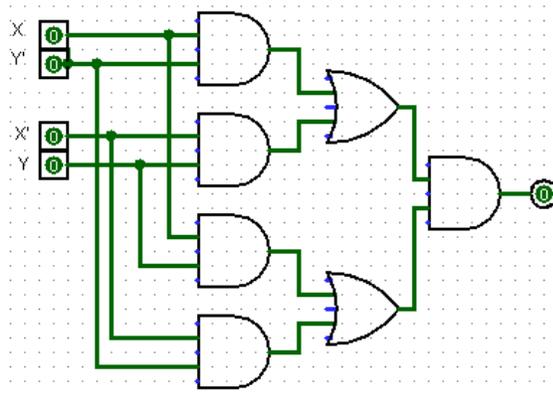
Figura 171. **Enunciado 3 capítulo 3 examen 3**



Fuente: elaboración propia, empleando Logisim.

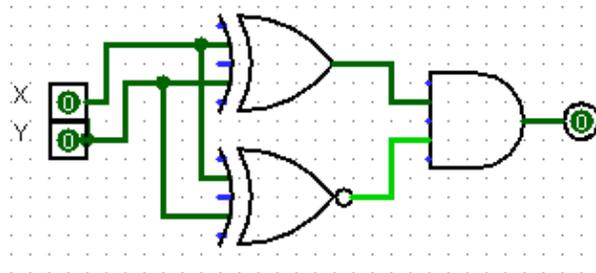
- Dibujar el circuito equivalente utilizando compuertas XOR para el circuito de la Figura 172.

Figura 172. **Enunciado 4 capítulo 3 examen 3**



Fuente: elaboración propia, empleando Logisim.

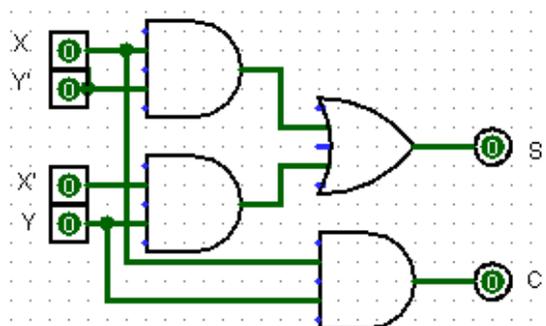
Figura 173. **Respuesta del enunciado 4 capítulo 3 examen 3**



Fuente: elaboración propia, empleando Logisim.

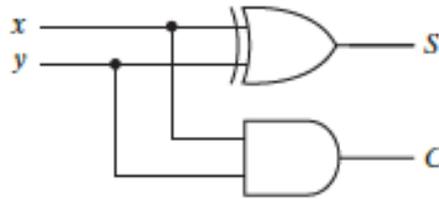
- ¿Por qué es preferible utilizar las compuertas NAND y NOR en vez de las compuertas AND y OR? Porque son fáciles de fabricar con componentes electrónicos y son las compuertas básicas empleadas en todas las familias de lógica de CI.
- ¿De qué otra forma se puede representar el circuito de la figura 176? La forma solicitada no es la más simplificada.

Figura 174. **Enunciado 6 capítulo 3 examen 3**



Fuente: elaboración propia, empleando Logisim.

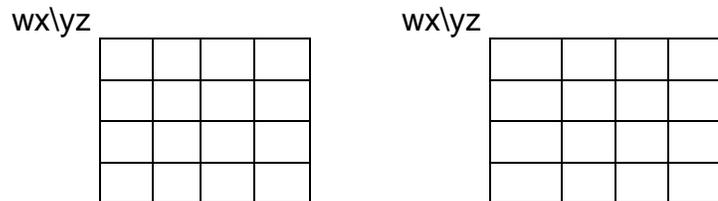
Figura 175. Respuesta del enunciado 6 capítulo 3 examen 3



Fuente: MANO, Morris. *Diseño digital*. p 120.

- Ordenar cada uno de los minterminos con las variables expuestas para el mapa de Karnaugh de cuatro variables.

Figura 176. Enunciado 7 capítulo 3 examen 3



Fuente: elaboración propia, empleando Excel.

Figura 177. Respuesta de enunciado 7 capítulo 3 examen 3

v=0					v=1				
wx\yz	00	01	11	10	wx\yz	00	01	11	10
00	v'w'x'y'z'	v'w'x'yz'	v'w'x'yz	v'w'x'yz'	00	vw'x'y'z'	vw'x'yz	vw'x'yz	vw'x'yz'
01	v'w'xy'z'	v'w'xy'z	v'w'xyz	v'w'xyz'	01	vw'xy'z'	vw'xy'z	vw'xyz	vw'xyz'
11	v'wxy'z'	v'wxy'z	v'wxyz	v'wxyz'	11	vwx'y'z'	vwx'y'z	vwx'yz	vwx'yz'
10	v'wx'y'z'	v'wx'y'z	v'wx'yz	v'wx'yz'	10	vwx'y'z'	vwx'y'z	vwx'yz	vwx'yz'

Fuente: elaboración propia, empleando Excel.

#### 4.4. Exámenes de lógica combinacional

A continuación, se presentan una cantidad de tres exámenes propuestos del capítulo lógica combinacional.

##### 4.4.1. Examen uno

Este es el primer examen propuesto y está compuesto por una cantidad de seis enunciados.

- ¿En función de qué está la salida de un circuito combinacional? En función de las entradas del circuito.
- Cuando se tienen  $n$  variables de entrada en un circuito combinacional, ¿Cuántas combinaciones de entradas binarias se tienen en un circuito? Se tienen  $2^n$  combinaciones.
- ¿Cuál es la diferencia entre un semisumador y un sumador completo? El semisumador, por limitaciones de hardware, solo puede sumar dos bits y el sumador completo puede realizar la suma aritmética de hasta tres bits.
- ¿Qué problemática resuelve el circuito de la propagación del acarreo? Permite que el valor  $S$  y el valor  $C$  lleguen al mismo tiempo, dado que de otro modo llegarían en tiempos distintos por los retrasos que pueden generar en pasar una cantidad  $n$  de compuertas lógicas.
- Demostrar que las salidas de la figura 179 son equivalentes a las salidas de la figura 180. Utilice algebra de Boole para su respuesta

$$S = x'y'z + x'yz' + xy'z' + xyz$$
$$z(x'y' + xy) + z'(x'y + xy')$$

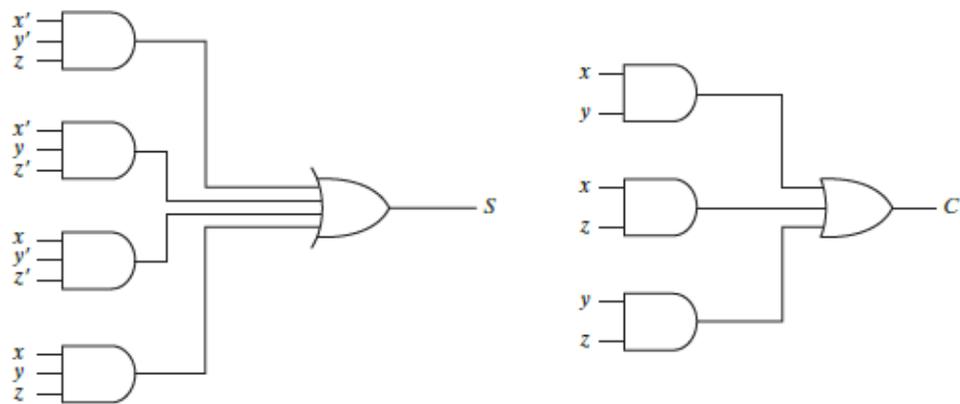
$$z(x \oplus y)' + z'(x \oplus y)$$

$$z \oplus (x \oplus y)$$

$$C = xy + xz + yz$$

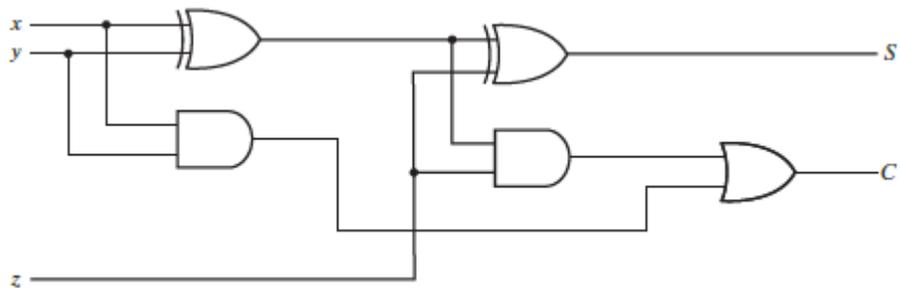
$$z(x + y) + xy$$

Figura 178. **Circuito 1 del enunciado 5 capítulo 4 examen 1**



Fuente: MANO, Morris. *Diseño digital*. p 121.

Figura 179. **Circuito 2 del enunciado 5 capítulo 4 examen 1**



Fuente: MANO, Morris. *Diseño digital*. p 122.

- Sabiendo que un sumador binario se construye en base a sumadores completos, ¿qué pasaría si se sustituyera el sumador completo con un semisumador del bit menos significativo del sumador binario? ¿Habría algún cambio? Explique su respuesta. Se hace referencia al semisumador. Hay que recordar la diferencia entre un sumador completo y el semisumador es que el sumador completo tiene la capacidad de sumar tres bits y el semisumador solo dos. Al no dejar una entrada de acarreo en el bit menos significativo del semisumador, impide que este sumador pueda utilizarse junto a otros sumadores y de ese modo sumar dos números de mayor cantidad de bits.

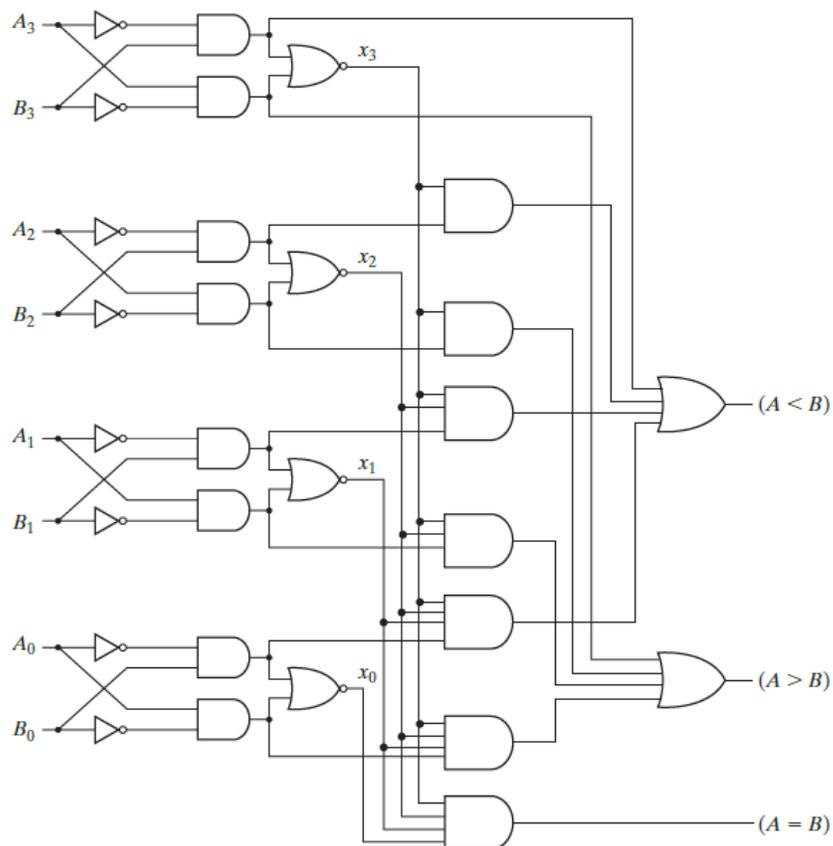
#### **4.4.2. Examen dos**

Este es el segundo examen propuesto y está compuesto por una cantidad de seis enunciados.

- ¿Por qué es más conveniente un circuito sumador-restador binario, a comparación de los circuitos: sumador y restador binario por aparte? Al utilizar el sumador y restador binario se puede ahorrar una gran cantidad de compuertas lógicas y, por ende, haciéndolo un método más eficiente.
- ¿Por qué hay que sumar seis a la conversión de binario y BCD? ¿A cuáles números hay que sumarlo? Para poder representar el número en binario en la codificación en BCD. Al sumar seis genera el acarreo necesario para que la década siguiente aumente en uno, y hace que el valor de la década actual se muestre en un valor acorde con un valor en el sistema decimal. Hay que sumarlos a los números mayores a nueve porque los números menores no cambian respecto de la codificación de binario natural

- ¿Qué es un decodificador? Es un circuito combinacional que convierte información binaria de  $n$  salidas de entrada a un máximo de  $2^n$  líneas de salida distintas.
- Teniendo un decodificador de  $n$  entradas ¿Es necesario utilizar las  $2^n$  salidas? No es necesario, se pueden utilizar solo los minitérminos de interés.
- Explicar el funcionamiento del comparador de magnitudes de cuatro bits.

Figura 180. **Circuito del enunciado 5 capítulo 4 examen 2**



Fuente: MANO, Morris. *Diseño digital*. p 134.

Para ejemplificar se hará un comparador de magnitudes con dos números de cuatro bits (A y B).

$$A = A_3A_2A_1A_0$$

$$B = B_3B_2B_1B_0$$

Donde  $A_i$  y  $B_i$  son cada dígito del número de cuatro bits. Para determinar si  $A = B$ , se necesita que  $A_i = B_i$  sean iguales. Esto se puede determinar con una compuerta XOR.

$$x_i = A_i \oplus B_i = A_iB_i + A'_iB'_i$$

Donde  $x_i = 1$  cuando ambos números sean cero o uno. Ahora, esto sería por cada uno de los dígitos, y como se necesita que se cumplan todas, las salidas  $x_i$  irán conectadas hacia una compuerta AND.

$$(A = B) = (A_3B_3 + A'_3B'_3)(A_2B_2 + A'_2B'_2)(A_1B_1 + A'_1B'_1)(A_0B_0 + A'_0B'_0)$$

Para determinar si  $A > B$  se inspeccionará las magnitudes relativas de pares de dígitos significativos, comenzando de la posición más significativa. Si los dos dígitos son iguales se comparará el siguiente par de dígitos menos significativos. Esta comparación continúa hasta encontrar un par de dígitos distintos. Si el dígito correspondiente de A es 1 y el de B es 0, se concluye que  $A > B$ . Si el dígito correspondiente de A es 0 y el de B es 1, se concluye que  $A < B$ .

Figura 181. **Ecuaciones enunciado 5 capítulo 4 examen 2**

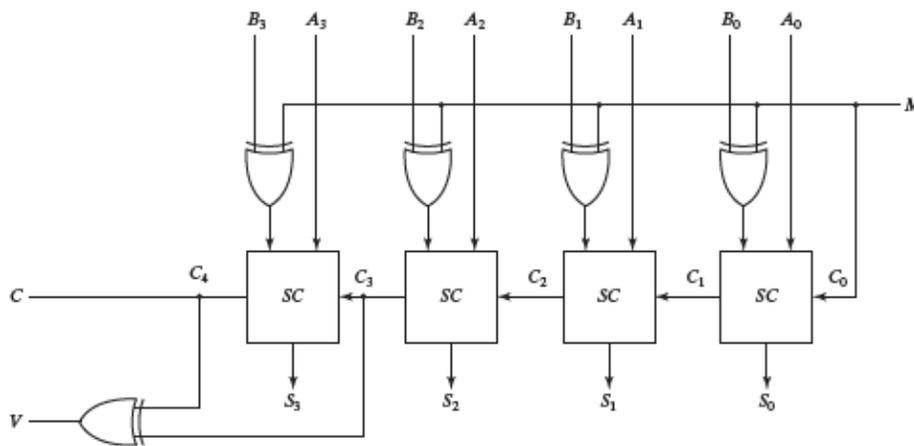
$$(A > B) = A_3B'_3 + x_3A_2B'_2 + x_3x_2A_1B'_1 + x_3x_2x_1A_0B'_0$$

$$(A < B) = A'_3B_3 + x_3A'_2B_2 + x_3x_2A'_1B_1 + x_3x_2x_1A'_0B_0$$

Fuente: Mano Morris M., *Diseño digital*. p.134.

- Dibujar el diagrama de bloques del sumador-restador binario y explique su funcionamiento. Las operaciones de suma y resta se pueden combinar en un solo circuito. Esto se realiza agregando una compuerta XOR a cada sumador completo del sumador de cuatro bits que se ha visto previamente. La entrada booleana  $M$  controla la funcionalidad del circuito. Si  $M = 0$  el circuito es sumador, y si  $M = 1$  el circuito es restador. En la compuerta XOR, cuando  $M = 0$  entonces  $B \oplus 0 = B$  y el acarreo inicial es 0, por lo que las entradas del sumador completo serían  $A$  y  $B$ . Cuando  $M = 1$  entonces  $B \oplus 1 = B'$  y el acarreo inicial es 1, por lo que la entrada sería  $A - B + 1$ , lo que es igual a la resta por medio de complementos.

Figura 182. **Respuesta del enunciado 6 capítulo 4 examen 2**



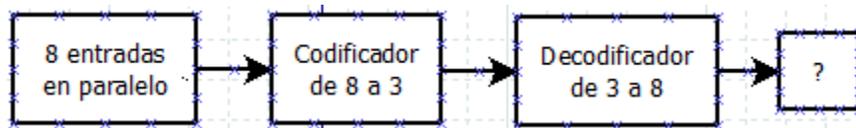
Fuente: MANO, Morris. *Diseño digital*. p 127.

#### 4.4.3. Examen tres

Este es el tercer examen propuesto y está compuesto por una cantidad de seis enunciados.

- ¿Qué es una entrada habilitadora? Es una entrada que permite habilitar la salida de un circuito.
- ¿Qué es un circuito decodificador/desmultiplexor? Es un circuito decodificador con entradas de habilitación.
- ¿Cuál es la diferencia entre un codificador con prioridad y un codificador tradicional y que problemática resuelve el codificador con prioridad? El codificador con prioridad permite seleccionar una salida en base a prioridades establecidas en el diseño del circuito. Permite resolver problemas con ambigüedad, y estar seguros de cuál será la salida con ciertas entradas.
- ¿Qué es la multiplexación? Es la capacidad de poder seleccionar la salida entre dos o más entradas con la ayuda de un circuito de selección.
- Respecto del diagrama de bloques de la figura 184, ¿Cuál sería el resultado final? En base a su respuesta anterior ¿Con que circuito equivalente lo reemplazaría? Dado a que es un circuito codificador conectado a un circuito decodificador el resultado sería las entradas al circuito codificador (8 salidas en paralelo). Es decir, que daría lo mismo incluir el codificador y el decodificador. El circuito equivalente sería una serie de cables conectados directamente a las salidas del circuito.

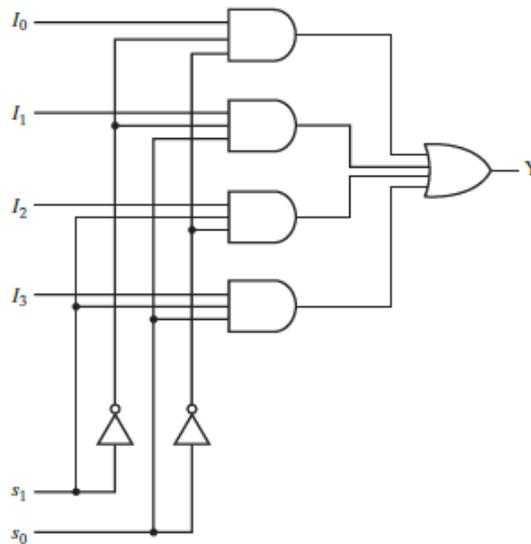
Figura 183. **Enunciado 5 capítulo 4 examen 1**



Fuente: elaboración propia, empleando Dia.

- Dibuje el diagrama lógico del multiplexor de 4 líneas a 1 y explique su funcionamiento. El circuito funciona por medio de las entradas habilitadoras  $S_1$  y  $S_2$ . Estas habilitan las entradas  $I_0$ ,  $I_1$ ,  $I_2$  o  $I_3$  Para que pueda para la señal hacia la salida  $Y$ .

Figura 184. **Enunciado 6 capítulo 4 examen 3**



Fuente: MANO, Morris. *Diseño digital*. p 142.

## 4.5. Exámenes de lógica secuencial

A continuación, se presentan una cantidad de tres exámenes propuestos del capítulo lógica secuencial.

### 4.5.1. Examen uno

Este es el primer examen propuesto y está compuesto por una cantidad de siete enunciados.

- ¿Qué es un circuito secuencial? Es un circuito el cual su salida depende de sus entradas y de la información que tengan almacenada sus dispositivos de almacenamiento.
- ¿Qué son los elementos de almacenamiento? Son dispositivos capaces de guardar información binaria.
- ¿Qué es un circuito secuencial sincrónico? Es un sistema que su comportamiento se define conociendo sus señales en instantes discretos.
- ¿Qué es un circuito secuencial asincrónico? Es un sistema que su comportamiento se define en cualquier instante del tiempo, por lo que no está definido por ningún tren de pulsos.
- ¿Qué es un *flip-flop*? Es un dispositivo con la capacidad de mantenerse en uno de dos estados posibles durante un tiempo indefinido.
- Usted está diseñando un circuito secuencial, si su circuito es sincrónico ¿qué dispositivos de almacenamiento utilizaría? Y si su circuito asincrónico ¿Cuál utilizaría? Explique porqué. Si el circuito a diseñar es sincrónico, es recomendable utilizar *flip-flops*, porque este tiene una entrada de reloj. Al diseñar un circuito asincrónico es necesario utilizar *latches* como dispositivos de almacenamiento porque no cuentan con ninguna entrada de reloj.

- Al aplicar un tren de pulsos a la entrada de control de un *latch* ¿Actuaría de la misma manera que un *flip-flop*? Explique su respuesta. Al conectar un tren de pulsos en un *latch*, se tendría como resultado que cada vez que la entrada de control sea igual a uno, podría realizarse un cambio en el estado del *latch* (respuesta a nivel positivo). En cambio, un *flip-flop* solo puede cambiar su estado cuando existe un cambio de estado en el reloj (respuesta al borde positivo o negativo). Por ello un *latch* nunca tendrá el mismo comportamiento que un *flip-flop*.

#### 4.5.2. Examen dos

Este es el segundo examen propuesto y está compuesto por una cantidad de cinco enunciados.

- ¿Cuál es la diferencia principal entre un *latch* y un *flip-flop*? Un *latch* no tiene entrada de reloj, y un *flip-flop* si cuenta con ella.
- ¿Para qué sirve una entrada de control en un *latch*? Ayuda a poder controlar cuando se desea que el circuito cambie de estado. Si esta entrada no es activada, el *latch* no cambiara de estado.
- En un *latch* SR ¿Qué es un estado indefinido? Es cuando no se sabe cuál será el estado siguiente del *latch*. Si es un *latch* SR con compuertas NOR, esto ocurrirá cuando  $S=R=0$ ; si es un *latch* SR con compuertas NAND, esto ocurrirá cuando  $S=R=1$ . Las entradas de los *latches* afectarán al siguiente estado y no al estado actual.
- ¿Cuál es la diferencia entre un *latch* SR y un *latch* D? El *latch* D no puede llegar a caer a un estado indefinido por su compuerta NOT en la entrada D.
- ¿Qué son las entradas asíncronas? Son entradas que no dependen del reloj para funcionar. Permiten establecer o restablecer los *flip-flops* en el momento que se desee.

### 4.5.3. Examen tres

Este es el tercer examen propuesto y está compuesto por una cantidad de seis enunciados.

- ¿Qué es una ecuación de estado? Es una expresión algebraica que especifica el siguiente estado de un *flip-flop*. Esto en función de sus entradas y estados actuales de los *flip-flops*.
- ¿Qué es una tabla de estados? Es una tabla compacta donde se muestra los posibles estados siguientes en función de las entradas y los estados actuales de los *flip-flops*.
- ¿Qué es un diagrama de estados? Es una representación gráfica de las transiciones de circuitos secuencial en base a las entradas y los estados actuales de los *flip-flops* que componen el circuito.
- ¿Qué es la reducción de estados? Es un procedimiento que puede reducir la cantidad de estados en una tabla de estados, pero no afecta en las salidas del circuito.
- ¿Qué es una tabla de excitación? Es una relación funcional entre la tabla de estados y las ecuaciones de entrada, que dará una vista rápida y directa del comportamiento de los *flip-flop*.
- Entre los *flip-flops* D, JK y T ¿Cuál elegiría para realizar su diseño? Explique su respuesta. Dependerá del circuito que se desee realizar. Ningún *flip-flop* es superior al otro, simplemente que cada uno tiene su función en específico. El *flip-flop D* es fácil de implementar dado a que no es necesaria una tabla de excitación. El *flip-flop JK* puede llegar a ser muy versátil. Por parte del *flip-flop T*, es muy útil para el diseño de contadores binarios.

## 4.6. Exámenes de sistemas de registros y contadores

A continuación, se presentan una cantidad de tres exámenes propuestos del capítulo lógica secuencial.

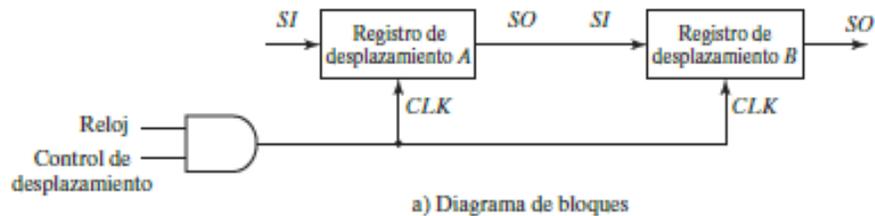
### 4.6.1. Examen uno

Este es el primer examen propuesto y está compuesto por una cantidad de seis enunciados.

- ¿Qué es un registro? Es un grupo de *flip-flops* y compuertas que efectúan una transición. Los *flip-flops* contiene la información binaria y las compuertas determinan como la información se transfiere al registro.
- ¿Qué es un registro con carga paralela? Es un registro que se carga simultáneamente con un pulso de reloj común. Esto quiere decir que ocurre una transferencia de registro de todos los *flip-flops* al mismo tiempo con una misma señal. Todas las entradas C de los *flip-flops* están conectadas al mismo reloj.
- ¿Qué es un registro de desplazamiento? Es un registro capaz de desplazar su información binaria en una dirección o en la otra. La configuración lógica de un registro de desplazamiento consiste en una cadena de *flip-flops* en cadena, con la salida de un *flip-flop* conectada a la entrada del siguiente *flip-flop*.
- ¿Qué es una entrada de control de desplazamiento? Cuando se habla de registros de desplazamiento, se tiene la problemática que los datos siguen ingresando al registro y, por ser seriales, reemplazarán los registros más antiguos. Esto se resuelve con una entrada de control de desplazamiento en un *flip-flop*. Esta será una compuerta AND que permitirá hacer un desplazamiento de registros solo cuando esté activa.

- ¿Qué es un registro de desplazamiento universal? ¿Cuáles son sus características? Es un registro que puede almacenar información de forma paralela y de forma bidireccional en serial. Cuando un registro es unidireccional puede almacenar información ya sea de derecha a izquierda y de izquierda a derecha. Un registro bidireccional permite almacenar información de izquierda a derecha y de derecha a izquierda. Estos registros cuentan con las siguientes características:
  - Una entrada de restablecimiento.
  - Una entrada de reloj.
  - Un control de desplazamiento a la derecha para habilitar el desplazamiento a la derecha.
  - Un control de desplazamiento a la izquierda para habilitar el desplazamiento a la izquierda.
  - Un control de carga en paralelo para habilitar la transferencia en paralelo.
  - Una cantidad de  $n$  líneas de salida en paralelo.
  - Un control de estado para dejar los registros con los estados actuales.
  
- ¿Cuál es la función del registro de desplazamiento B (ver figura 186)? Es la de almacenar la información que el registro de desplazamiento A contiene. Dado a que la información en el registro A sigue ingresando, de no guardarse, se perderían los datos.

Figura 185. Enunciado 6 capítulo 6 examen 1



Fuente: MANO, Morris. *Diseño digital*. p 221.

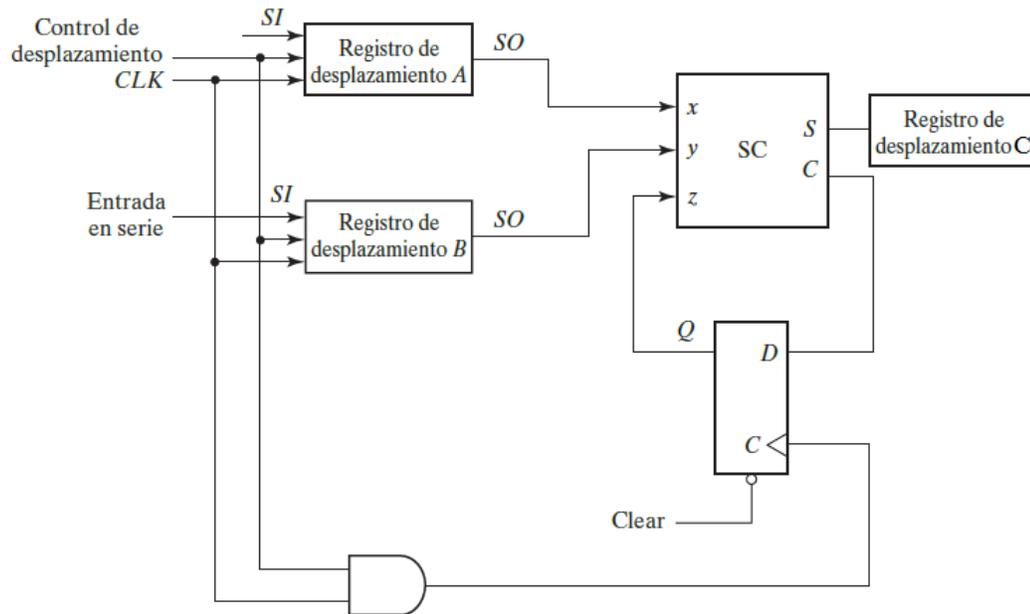
#### 4.6.2. Examen dos

Este es el segundo examen propuesto y está compuesto por una cantidad de seis enunciados.

- ¿Qué es un contador? Es básicamente un registro que pasa por una sucesión predeterminada de estados. Las compuertas del contador están conectadas de tal manera que producen la sucesión prescrita de estados binarios.
- ¿Qué es un contador de rizo? Son dispositivos contadores que tienen conectados los *flip-flops* en forma asíncrona, es decir, que no tienen conectadas las entradas de reloj en paralelo. En un contador de rizo la transición de salida del *flip-flop* sirve como disparador de otros *flip-flops*.
- ¿Qué es un contador binario de rizo? Un contador binario de rizo es un contador donde los *flip-flops* están conectados en serie y cuentan de manera ascendente o descendente (según configuración).

- ¿Qué es un contador sincrónico? En un contador sincrónico las salidas C de todos los *flip-flops* están conectadas al reloj común. Un solo contador dispara todos los *flip-flops* simultáneamente en vez de hacerlo uno por uno sucesivamente como en los contadores de rizo.
- ¿Qué problemática resuelve el contador binario de carga paralela? Muchas veces es necesario iniciar el conteo desde un valor que no fue pre establecido con anterioridad. Es decir, un valor el cual no haya sido incluido en el diseño, pero que se encuentre en el rango que el contador pueda soportar (que dependerá en la cantidad de bits del contador). Esto lo permite el contador binario con carga paralela.
- Dibujar el diagrama de bloques y explicar el funcionamiento del circuito de suma serial. Como se vio con anterioridad se puede realizar una suma en paralelo, donde todos los bits de los números binarios a sumar entran al mismo tiempo al sumador. Ahora se verá una versión de un sumador donde los sumandos serán ingresados de manera serial. En la figura 187 se puede visualizar el circuito, el cual está compuesto por dos registros de desplazamiento, una compuerta AND para control, un sumador completo y un *flip-flop D* para el acarreo. Los números de ambos sumandos serán ingresados de forma serial por las entradas SI de los registros de desplazamiento. La suma es realizada bit por bit y los resultados se almacenarán en un registro de desplazamiento C. Esto siempre será controlado por la entrada de reloj CLK. El acarreo es manejado por un *flip-flop D*, el cual al momento de haber un acarreo lleva ese uno a la siguiente suma del siguiente par de bits. La compuerta AND sirve para control de saber cuándo sumar.

Figura 186. Enunciado 6 capítulo 6 examen 2



Fuente: MANO, Morris. *Diseño digital*. p 223.

### 4.6.3. Examen tres

Este es el tercer examen propuesto y está compuesto por una cantidad de seis enunciados.

- ¿Qué es un contador con autocorrección? Es un contador que tiene la capacidad de regresar a alguno de sus estados conocidos, si por error callo a un estado no conocido. Esto pudo ser causado por algún fallo en el sistema o de mal uso por el usuario.

- ¿Qué es un contador de estados no utilizados? Son contadores los cuales tienen la peculiaridad de que sus estados no son necesariamente sucesivos ni se utilizan todos los estados que los *flip-flops* pueden otorgar. Los estados de interés dependerán del diseño del usuario.
- ¿Qué es un contador anular? El contador anular o de anillo es un circuito en donde solo un *flip-flop* está establecido y todos los demás *flip-flops* están restablecidos. Por ejemplo, si se tienen tres *flip-flops* se tendrían cuatro estados los cuales serían: 000, 100, 010, 001. Se incluyó el estado 000 como estado inicial.
- Realizar el diseño de un contador anular de 3 bits.

Tabla LXXVI. **Enunciado 4 capítulo 6 examen 3**

Estado actual			Estado siguiente			Entradas <i>flip-flops</i>		
A	B	C	A	B	C	Ta	Tb	Tc
0	0	0	1	0	0	1	0	0
1	0	0	0	1	0	1	1	0
0	1	0	0	0	1	0	1	1
0	0	1	1	0	0	1	0	1

Fuente: elaboración propia, empleando Excel.

Figura 187. **Mapas de Karnaugh de enunciando 4 capítulo 6 examen 3**

Para Ta

A\BC	00	01	11	10
0	1	1	X	0
1	1	X	X	X

A\BC	00	01	11	10
0	1	1	X	0
1	1	X	X	X

Para Tb

A\BC	00	01	11	10
0	0	0	X	1
1	1	X	X	X

A\BC	00	01	11	10
0	0	0	X	1
1	1	X	X	X

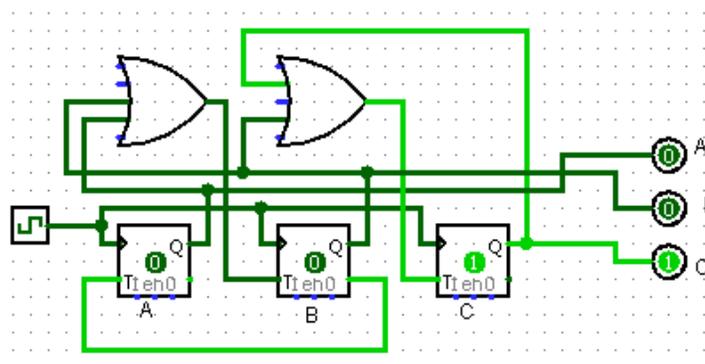
Para Tc

A\BC	00	01	11	10
0	0	1	X	1
1	0	X	X	X

A\BC	00	01	11	10
0	0	1	X	1
1	0	X	X	X

Fuente: elaboración propia, empleando Excel.

Figura 188. **Circuito de enunciando 4 capítulo 6 examen 3**



Fuente: elaboración propia, empleando Logisim.

$$T_A = A'B'C' + AB'C' + A'B'C$$

$$T_A = AB'C' + A'BC'$$

$$T_A = AB'C' + A'B'C$$

Ahora con la ayuda de mapas de Karnaugh se puede llegar a las siguientes ecuaciones:

$$T_A = B'$$

$$T_B = A + B$$

$$T_C = B + C$$

Con las ecuaciones obtenidas se puede realizar el diagrama del circuito correspondiente (figura 189).

- ¿Qué es un contador Johnson? El contador Johnson tiene la misma funcionalidad que el contador anular, pero con la diferencia que con este contador es necesario un número menor de componentes electrónicos.

## **5. VIDEOS DE CONCEPTOS TEÓRICOS DEL CURSO DE ELECTRÓNICA 3**

A continuación, se presentarán los videos educativos realizados para el curso de Electrónica 3. Cada video cubrirá algunos de los temas expuestos en el capítulo dos.

### **5.1. Videos del capítulo 1: sistemas de numeración**

- Video 1: Sistemas de numeración. Los temas cubiertos fueron:
  - Reseña histórica
  - Sistemas de numeración importantes en la electrónica digital
    - Sistema de numeración decimal
    - Sistema de numeración binario
    - Sistema de numeración octal
    - Sistema de numeración hexadecimal
  
- Video 2: Conversión de base  $r$  a decimal. Los temas cubiertos fueron:
  - Conversión de base  $r$  a decimal
    - Conversión de binario a decimal
    - Conversión de octal a decimal
    - Conversión de hexadecimal a decimal

- Video 3: Conversión de base decimal a base r. Los temas cubiertos fueron:
  - Conversión de decimal a base r
    - Conversión de decimal a binario
    - Conversión de decimal a octal
    - Conversión de decimal a hexadecimal
  
- Video 4: Conversiones entre binario, octal y hexadecimal. Los temas cubiertos fueron:
  - Relación de números binario, octales y hexadecimales
    - Binario a octal
    - Binario a hexadecimal
    - Octal y hexadecimal a binario
  
- Video 5: Aritmética binaria. Los temas cubiertos fueron:
  - Aritmética binaria
    - Suma
    - Resta
    - Multiplicación
    - División
  
- Video 6: Aritmética octal. Los temas cubiertos fueron:
  - Aritmética octal
    - Suma
    - Resta
    - Multiplicación
    - División

- Video 7: Aritmética hexadecimal. Los temas cubiertos fueron:
  - Aritmética hexadecimal
    - Suma
    - Resta
    - Multiplicación
    - División
  
- Video 8: Complementos. Los temas cubiertos fueron:
  - Complementos
    - Complemento a la base disminuida.
      - ✓ Números decimales
      - ✓ Números binarios
      - ✓ Números octales
      - ✓ Números hexadecimales
    - Complemento a la base disminuida
      - ✓ Números decimales
      - ✓ Números binarios
      - ✓ Números octales
      - ✓ Números hexadecimales
    - Resta con complementos
  
- Video 9: Números binarios con signo. Los temas cubiertos fueron:
  - Números binarios con signo
    - Notación con signo magnitud
    - Notación con complemento a uno
    - Notación con complemento a dos
    - Notación en exceso a K

- Video 10: Resta con complementos. Los temas cubiertos fueron:
  - Resta con complementos
  
- Video 11: Códigos binarios. Los temas cubiertos fueron:
  - Binario natural
  - Código BCD 8421
  - Código 2421
  - Código exceso 3
  - Código 8 4 -2 -1
  - Código Gray
  - Código ASCII
  - Códigos para detectar errores
  
- Video 12: Lógica binaria. Los temas cubiertos fueron:
  - Lógica binaria
  
- Video 13: Compuertas lógicas. Los temas cubiertos fueron:
  - Compuertas lógicas
    - Compuerta lógica YES
    - Compuerta lógica NOT
    - Compuerta lógica AND
    - Compuerta lógica NAND
    - Compuerta lógica OR
    - Compuerta lógica NOR
    - Compuerta lógica XOR
    - Compuerta lógica XNOR

## 5.2. Videos del capítulo 2: álgebra booleana

- Video 14: Álgebra de Boole. Los temas cubiertos fueron:
  - Reseña histórica
  - Sistema de numeración decimal
  - Sistema de numeración binario
  - Sistema de numeración octal
  - Sistema de numeración hexadecimal
  
- Video 15: Álgebra booleana de dos valores. Los temas cubiertos fueron:
  - Álgebra booleana de dos valores
  
- Video 16: Teoremas y propiedades básicas del álgebra booleana. Los temas cubiertos fueron:
  - Principio de dualidad
  - Teoremas básicos del álgebra booleana
    - Teorema 1
    - Teorema 2
    - Teorema 3, involución
    - Teorema 4, asociatividad
    - Teorema 5, DeMorgan
    - Teorema 6, absorción
  
- Video 17: Funciones booleanas. Los temas cubiertos fueron:
  - Funciones booleanas

- Video 18: Simplificación de funciones booleanas con álgebra de Boole. Los temas cubiertos fueron:
  - Simplificación de funciones booleanas con álgebra de Boole
  
- Video 19: Complementos de una función booleana. Los temas cubiertos fueron:
  - Complemento de una función booleana
  
- Video 20: Niveles en compuertas lógicas. Los temas cubiertos fueron:
  - Niveles de compuertas lógicas
  
- Video 21: Formas canónicas y estándar. Los temas cubiertos fueron:
  - Formas canónicas
    - Minitérminos y maxitérminos
  - Formas estándar
    - Suma de productos
    - Producto de sumas
  
- Video 22: Compuertas lógicas digitales. Los temas cubiertos fueron:
  - Compuertas lógicas digitales
    - Lógica positiva y negativa
  
- Video 23: Circuitos integrados. Los temas cubiertos fueron:
  - Circuitos integrados
    - Niveles de integración
    - Familias de lógica digital

### 5.3. Videos del capítulo 3: minimización de funciones booleanas

- Video 24: Minimización de funciones booleanas. Los temas por cubrir fueron los siguientes:
  - Introducción
  - Mapa de Karnaugh
  
- Video 25: Mapa de Karnaugh de dos variables. Los temas por cubrir fueron los siguientes:
  - Mapa de Karnaugh
    - Mapa de dos variables
  
- Video 26: Mapa de Karnaugh de tres variables. Los temas por cubrir fueron los siguientes:
  - Mapa de Karnaugh
    - Mapa de tres variables
  
- Video 27: Mapa de Karnaugh de cuatro variables. Los temas cubiertos fueron:
  - Mapa de Karnaugh
    - Mapa de cuatro variables
  
- Video 28: Mapa de Karnaugh de cinco variables. Los temas cubiertos fueron:
  - Mapa de Karnaugh
    - Mapa de cinco variables

- Video 29: Mapa de Karnaugh de seis variables. Los temas cubiertos fueron:
  - Mapa de Karnaugh
    - Mapa de seis variables
  
- Video 30: Simplificación de producto de sumas. Los temas cubiertos fueron:
  - Simplificación a producto de sumas
  
- Video 31: Condición indiferencia. Los temas cubiertos fueron:
  - Condición indiferencia
  
- Video 32: Implementación de compuertas NAND, NOR y XOR. Los temas cubiertos fueron:
  - Implementación de compuertas NAND y NOR
    - Implementación de compuertas NAND y NOR
    - Compuerta NAND
    - Compuerta NOR
  - Compuerta OR exclusiva

#### **5.4. Videos del capítulo 4: lógica combinacional**

- Video 33: Lógica combinacional. Los temas cubiertos fueron:
  - Introducción
  - Procedimiento de análisis
  - Procedimiento de diseño

- Video 34: Sumadores. Los temas cubiertos fueron:
  - Semisumador
  - Sumador completo
  - Sumador binario
  - Propagación del acarreo
  
- Video 35: Restador binario. Los temas cubiertos fueron:
  - Restador binario
  
- Video 36: Sumador-restador binario. Los temas cubiertos fueron:
  - Sumador-restador binario
  
- Video 37: Sumador BCD de cuatro bits. Los temas cubiertos fueron:
  - Sumador BCD de cuatro bits
  
- Video 38: Multiplicador binario. Los temas cubiertos fueron:
  - Multiplicador binario
    - Multiplicador binario con sumador simple
    - Multiplicador binario con sumadores completos
  
- Video 39: Comparador de magnitudes. Los temas cubiertos fueron:
  - Comparador de magnitudes
  
- Video 40: Decodificadores. Los temas cubiertos fueron:
  - Decodificadores

- Video 41: Demultiplexor. Los temas cubiertos fueron:
  - Demultiplexor
    - Entradas habilitadoras
    - Demultiplexor
    - Decodificador/Desmultiplexor
  
- Video 42: Codificadores. Los temas cubiertos fueron:
  - Codificadores
    - Codificador con prioridad
  
- Video 43: Multiplexores. Los temas cubiertos fueron:
  - Multiplexor

## 5.5. Videos del capítulo 5: lógica secuencial

- Video 44: Lógica secuencial. Los temas cubiertos fueron:
  - Introducción
    - Circuito secuencial sincrónico
    - Circuito secuencial asincrónico
  
- Video 45: Dispositivos de almacenamiento. Los temas cubiertos fueron:
  - *Flip-flops*
    - *Latches*
      - ✓ *Latch SR*
      - ✓ *Latch D*
    - *Flip-flops*

- ✓ *Flip-flop D* disparado por flanco
  - ✓ *Flip-flop JK*
  - ✓ *Flip-flop T*
  - Entradas asíncronas
- Video 46: Análisis de circuitos secuenciales. Los temas cubiertos fueron:
  - Análisis de circuitos secuenciales
    - Ecuaciones de estado
    - Tabla de estados
    - Diagrama de estados
- Video 47: Análisis con *flip-flops*. Los temas cubiertos fueron:
  - Análisis con *flip-flops*
    - Análisis con *flip-flop D*
    - Análisis con *flip-flop JK*
    - Análisis con *flip-flop T*
  - Máquinas de estado
- Video 48: Reducción de estados. Los temas por cubrir fueron los siguientes:
  - Reducción de estados
  - Asignación de estados
- Video 49: Diseño de circuitos secuenciales. Los temas por cubrir fueron los siguientes:

- Diseño de circuitos secuenciales
  - Síntesis con *flip-flops D*
  - Tablas de excitación
  - Síntesis con *flip-flops JK*
  - Síntesis con *flip-flops T*

## 5.6. Videos del capítulo 6: registros y contadores

- Video 50: Registros. Los temas cubiertos fueron:
  - Registros
    - Registros con carga paralela
    - Registros de desplazamiento
      - ✓ Transferencia en serie
    - Registros de desplazamiento universal
- Video 51: Suma en serie. Los temas cubiertos fueron:
  - Suma en serie
- Video 52: Contador de rizo. Los temas cubiertos fueron:
  - Contadores de rizo
    - Contador binario de rizo
    - Contador BCD de rizo

- Video 53: Contadores sincrónicos. Los temas cubiertos fueron:
  - Contadores sincrónicos
    - Contador binario
    - Contador binario ascendente-descendente
    - Contador BCD
    - Contador binario con carga paralela
    - Contador con estados no utilizados
    - Contador anular
    - Contador Johnson



## CONCLUSIONES

1. La gran mayoría de los catedráticos de FIUSAC no adaptan el constructivismo pedagógico a sus cátedras.
2. Las tecnologías de la información dan una gran ventaja en el ámbito de la enseñanza, dado a que se tiene acceso a la información en cualquier momento desde cualquier dispositivo con acceso a internet.
3. Actualmente, el curso de Electrónica 3 no profundiza en los temas impartidos, especialmente en los problemas de aplicación.
4. En el curso de Electrónica 3 se debe dar a conocer el tema de lógica secuencial asíncrona. Es un tema importante que debe ser conocido por un ingeniero electrónico, ingeniero eléctrico e ingeniero mecánico eléctrico.
5. Con el contenido en línea, el estudiante del curso de Electrónica 3 lo culminará con una mayor cantidad de conocimientos y estará más preparado para el ámbito profesional.



## RECOMENDACIONES

1. Para los catedráticos en FIUSAC, tomar en cuenta que el constructivismo es una metodología mucho más eficiente para enseñar, dado a que el estudiante retendrá por una mayor cantidad de tiempo los conceptos del curso.
2. Para el catedrático del curso Electrónica 3 y sus estudiantes: aprovechar las tecnologías de la información para buscar información relevante al curso. Hay que recordar que el ámbito tecnológico está en constante actualización.
3. Para el catedrático del curso Electrónica 3: dar a conocer a los alumnos del curso los videos conceptuales. Esto le ahorrará tiempo para que pueda dedicarse a temas más complejos como los problemas de aplicación utilizando para ello electrónica digital.
4. Para el catedrático del curso Electrónica 3: estudiar y dar a conocer el tema de lógica secuencial asíncrona, que es un tema de interés para el estudiantado.
5. Para el estudiantado que esté cursando el curso de Electrónica 3: dedicar el tiempo necesario para aprender a totalidad los conceptos del curso. Vean los videos y hagan preguntas al catedrático, que él les podrá resolver sus dudas.



## BIBLIOGRAFÍA

1. *Álgebra de Boole* [En línea]. <<https://goo.gl/VWq4jC>>[Consulta: 28 de marzo de 2018].
2. *Álgebra de Boole* [En línea]. <<https://goo.gl/XsxZ3H>>[Consulta: 25 de diciembre de 2016].
3. CARRANZA RAMÍREZ, Alba Irene. Ensayo el constructivismo como estrategia educativa: formación profesional en enfermería. [en línea]. <<http://goo.gl/gEgqWN>>. [Consulta: 29 de noviembre del 2016].
4. *Constructivismo (Filosofía)* [en línea]. <<https://goo.gl/EyTKFv>>[Consulta: 27 de noviembre de 2016].
5. *Constructivismo (Pedagogía)* [en línea]. <<https://goo.gl/y0IEf9>>. [Consulta: 28 de noviembre 2016].
6. *El Constructivismo* [en línea]. <<https://goo.gl/daq80C>>. [Consulta: 21 de noviembre de 2016].
7. *El constructivismo* [en línea]. <<https://goo.gl/eaBO2R>>. [Consulta: 20 de noviembre de 2016].
8. *Introducción a la lógica binaria* [en línea]. <<https://goo.gl/0mEoK9>>. [Consulta: 25 de diciembre de 2016].

9. *La inmaterialidad* [en línea]. <<https://goo.gl/67FoTa>>. [Consulta: 24 de diciembre de 2016].
10. *Lógica Binaria* [en línea]. <<https://goo.gl/3DHo6J>>. [Consulta: 25 de diciembre de 2016].
11. M.C. LEAL CHAPA, César Augusto. *Fundamentos de diseño digital*. País: Universidad Autónoma de Nuevo León, 2001. 315 p.
12. M. MANO, Morris, *Diseño digital*, 3a ed. País: California State University, Los Ángeles, 2003. 536 p.
13. *¿Qué es el constructivismo?* [en línea]. <<https://goo.gl/uj8hVk> >. [Consulta: 19 de noviembre de 2016].