

METODOLOGÍAS PARA DISEÑO DE CIRCUITOS LADDER CON BASE EN SISTEMAS  
SECUENCIALES Y COMBINACIONALES

MARIO ALBERTO BRITO SALDARRIAGA  
JOAN SEBASTIÁN GIRALDO BETANCOURT

UNIVERSIDAD TECNOLÓGICA DE PEREIRA  
FACULTAD DE INGENIERÍAS: ELÉCTRICA, ELECTRÓNICA, FÍSICA  
Y CIENCIAS DE LA COMPUTACIÓN

PROGRAMA DE INGENIERÍA ELÉCTRICA

PEREIRA, JULIO 2010

METODOLOGÍAS PARA DISEÑO DE CIRCUITOS LADDER CON BASE EN SISTEMAS  
SECUENCIALES Y COMBINACIONALES

MARIO ALBERTO BRITO SALDARRIAGA  
JOAN SEBASTIÁN GIRALDO BETANCOURT

Proyecto de grado presentado como requisito parcial para optar al título de  
Ingeniero Electricista.

Director: M.Sc. Mauricio Holguín Londoño

UNIVERSIDAD TECNOLÓGICA DE PEREIRA  
FACULTAD DE INGENIERÍAS: ELÉCTRICA, ELECTRÓNICA, FÍSICA  
Y CIENCIAS DE LA COMPUTACIÓN  
PROGRAMA DE INGENIERÍA ELÉCTRICA

PEREIRA, JULIO 2010

Nota de aceptación:

---

---

---

---

---

Firma del jurado

## DEDICATORIA

A...

DIOS, por darnos la oportunidad de vivir.

NUESTRAS FAMILIAS, por su apoyo incondicional.

NUESTROS AMIGOS, por su ayuda en esta etapa universitaria.

## AGRADECIMIENTOS

Al ingeniero y magister en ingeniería eléctrica Mauricio Holguín Londoño, director de este proyecto de grado, por su acompañamiento, paciencia y consejos.

Este proyecto de grado fue realizado mediante L $\gamma$ X©1.6.5 disponible en <http://www.lyx.org>



# Índice general

<b>I. INTRODUCCIÓN</b>	<b>1</b>
1. DEFINICIÓN DEL PROBLEMA	3
2. JUSTIFICACIÓN	5
3. OBJETIVOS	9
3.1. OBJETIVO GENERAL . . . . .	9
3.2. OBJETIVOS ESPECÍFICOS . . . . .	9
<b>II. PRELIMINARES</b>	<b>11</b>
4. LÓGICA DE PREDICADOS	13
5. <b>ÁLGEBRA DE BOOLE</b>	<b>17</b>
5.1. POSTULADOS DEL <b>ÁLGEBRA DE BOOLE</b> . . . . .	17
5.2. PRINCIPIO DE DUALIDAD . . . . .	18
5.3. TEOREMAS DEL <b>ÁLGEBRA DE BOOLE</b> . . . . .	18
5.4. FUNCIONES BOOLEANAS . . . . .	20
5.4.1. Forma canónica suma de productos . . . . .	21
5.4.2. Forma canónica producto de sumas . . . . .	21
5.4.3. Términos Don't Care . . . . .	22
6. <b>SIMPLIFICACIÓN DE FUNCIONES</b>	<b>25</b>
6.1. MÉTODO ALGEBRAICO . . . . .	25
6.2. MAPAS DE KARNAUGH . . . . .	26
6.3. QUINE-McCLUSKEY . . . . .	31
6.4. ALGORITMO DE PETRICK . . . . .	33
7. <b>FUNCIONES LÓGICAS</b>	<b>39</b>
7.1. COMPUERTA LÓGICA OR . . . . .	39

7.2. COMPUERTA LÓGICA AND . . . . .	40
7.3. COMPUERTA LÓGICA NOT . . . . .	41
7.4. COMPUERTA LÓGICA NAND . . . . .	42
7.5. COMPUERTA LÓGICA NOR . . . . .	43
7.6. COMPUERTA LÓGICA XOR . . . . .	44
7.7. COMPUERTA LÓGICA XNOR . . . . .	45
<b>8. LÓGICA SECUENCIAL</b>	<b>49</b>
8.1. BIESTABLE S-R . . . . .	49
8.2. BIESTABLE S-R CON HABILITACIÓN . . . . .	52
8.3. BIESTABLE D . . . . .	54
8.4. FLIP FLOPS . . . . .	56
8.4.1. Flip flop S-R maestro/esclavo . . . . .	56
8.4.2. Flip flop J-K maestro/esclavo . . . . .	58
8.4.3. Flip flop D maestro/esclavo . . . . .	60
8.4.4. Flip flop T . . . . .	61
<b>9. MÁQUINA DE ESTADOS FINITOS</b>	<b>65</b>
9.1. DIAGRAMA DE ESTADOS . . . . .	65
9.2. MÁQUINAS DE MEALY Y DE MOORE . . . . .	67
9.3. SIMPLIFICACIÓN . . . . .	67
9.3.1. Inspección . . . . .	67
9.3.2. Partición . . . . .	69
9.3.3. Tabla de Implicación . . . . .	71
<b>10. LÓGICA CABLEADA</b>	<b>77</b>
10.1. EL CONTACTOR . . . . .	78
10.1.1. Principales partes del contactor . . . . .	78
10.1.2. Funcionamiento del contactor . . . . .	82
10.1.3. Categoría de empleo . . . . .	82
10.2. EL RELÉ . . . . .	83
10.3. RELÉ DE ENCLAVAMIENTO . . . . .	83
10.4. CONTACTOR CON BOBINA DE AUTO-RETENCIÓN . . . . .	83
10.5. RELÉS DE TEMPORIZACIÓN . . . . .	84
10.6. ELEMENTOS DE MANDO . . . . .	85

<b>III. METODOLOGÍAS DE DISEÑO</b>	<b>87</b>
<b>11. METODOLOGÍA POR SEÑALES DE MANDO</b>	<b>89</b>
11.1. INTRODUCCIÓN	89
11.2. FUNCIONES BÁSICAS	90
11.2.1. Función pulsador (mando monoestable)	90
11.2.2. Función interruptor (mando biestable)	92
11.2.3. Función seguidor	93
11.2.4. Función NOT (inversor)	94
11.2.5. Función AND	96
11.2.6. Función OR	97
11.2.7. Función temporización a la excitación	98
11.2.8. Función temporización a la excitación negada	99
11.2.9. Función temporización a la desexcitación	101
11.2.10. Función temporización a la desexcitación negada	102
11.2.11. Función detector flancos de subida	103
11.2.12. Función detector flancos de bajada	105
11.2.13. Función refresco	106
11.2.14. Función relé de enclavamiento	108
11.2.15. Función relé de auto-retención	109
11.3. FUNCIONES DERIVADAS	112
11.3.1. Introducción	112
11.3.2. Función temporización a la excitación y desexcitación	112
11.3.3. Función temporización a la excitación y desexcitación negada	113
11.3.4. Función detector de flancos de subida y bajada	115
11.3.5. Función toggle	116
11.3.6. Función contador flancos de subida	118
11.3.7. Función contador flancos de bajada	119
11.3.8. Función contador flancos de subida y bajada	121
11.4. PASOS PARA DISEÑAR	122
11.5. EJEMPLOS BÁSICOS	123
11.5.1. <i>Ejemplo 1.</i>	123
11.5.2. <i>Ejemplo 2.</i>	129
11.5.3. <i>Ejemplo 3.</i>	132
<b>12. METODOLOGÍA POR MÁQUINAS DE ESTADO</b>	<b>137</b>
12.1. INTRODUCCIÓN	137

12.2. EQUIVALENCIAS . . . . .	137
12.2.1. Entradas . . . . .	137
12.2.2. Estados . . . . .	139
12.2.3. Transiciones Temporizadas . . . . .	141
12.2.4. Salidas . . . . .	142
12.2.5. Inicialización de la máquina de estados . . . . .	143
12.3. PASOS PARA DISEÑAR . . . . .	145
12.4. EJEMPLOS BÁSICOS . . . . .	145
12.4.1. <i>Ejemplo 4. Sistema de encendido y apagado de una lámpara</i> . . . . .	145
12.4.2. <i>Ejemplo 5. Desplazamiento de un móvil entre dos límites de recorrido</i> . . . . .	148
<b>IV. RESULTADOS Y CONCLUSIONES</b>	<b>155</b>
<b>13. RESULTADOS</b>	<b>157</b>
13.1. <i>Ejemplo 6.</i> . . . . .	157
13.2. <i>Ejemplo 7.</i> . . . . .	162
<b>14. CRITERIOS DE VALIDEZ Y CONFIABILIDAD</b>	<b>169</b>
14.1. <i>Ejemplo 8.</i> . . . . .	169
14.2. <i>Ejemplo 9.</i> . . . . .	172
14.3. <i>Ejemplo 10.</i> . . . . .	177
<b>15. CONCLUSIONES</b>	<b>189</b>

# Índice de figuras

6.1. Diagrama de Venn y mapa de Karnaugh equivalente para dos variables . . . . .	26
6.2. Diagrama de Venn y mapa de Karnaugh equivalente para tres variables . . . . .	27
6.3. Mapa de Karnaugh función $F = \sum m(0, 2, 3, 4, 7)$ . . . . .	28
6.4. Mapa de Karnaugh función $F = \prod (1, 5, 6)$ . . . . .	28
6.5. Mapa de Karnaugh términos don't care . . . . .	29
6.6. Minimización mapa de Karnaugh sin usar términos Don't Care . . . . .	30
6.7. Minimización mapa de Karnaugh usando términos Don't Care . . . . .	30
6.8. Mapas de Karnaugh para el ejemplo de Quine-McCluskey . . . . .	32
7.1. Compuerta lógica OR . . . . .	40
7.2. Compuerta lógica AND . . . . .	41
7.3. Compuerta lógica NOT . . . . .	42
7.4. Compuerta lógica NAND . . . . .	43
7.5. Compuerta lógica NOR . . . . .	44
7.6. Compuerta lógica XOR . . . . .	45
7.7. Compuerta lógica XNOR . . . . .	46
8.1. Biestable S-R con compuertas NOR . . . . .	50
8.2. Diagrama de estados Biestable S-R . . . . .	51
8.3. Biestable S-R . . . . .	51
8.4. Biestable S-R con habilitación con compuertas NAND . . . . .	52
8.5. Diagrama de estados Biestable S-R con habilitación . . . . .	53
8.6. Biestable S-R con habilitación . . . . .	54
8.7. Biestable D con compuertas NAND . . . . .	54
8.8. Diagrama de estados Biestable D . . . . .	55
8.9. Biestable D . . . . .	56
8.10. Flip flop S-R maestro/esclavo usando Biestables S-R . . . . .	57
8.11. Diagrama de estados flip flop S-R maestro/esclavo . . . . .	58
8.12. Símbolo flip flop S-R maestro/esclavo . . . . .	58
8.13. Flip flop J-K mestros/esclavo empleando Biestables S-R . . . . .	58
8.14. Diagrama de estados flip flop J-K maestro/esclavo . . . . .	59

8.15. Símbolo flip flop J-K maestro/esclavo . . . . .	60
8.16. Flip flop D maestro/esclavo utilizando Biestables D . . . . .	60
8.17. Diagrama de estados flip flop D maestro/esclavo . . . . .	61
8.18. Símbolo flip flop D maestro/esclavo . . . . .	61
8.19. Flip flop T empleando Flip flop J-K . . . . .	62
8.20. Diagrama de estados flip flop T . . . . .	63
8.21. Símbolo flip flop T . . . . .	63
9.1. Diagrama de estados . . . . .	66
9.2. Autómata de Moore y autómata de Mealy . . . . .	67
9.3. Circuito 1 . . . . .	68
9.4. Circuito 1 reducido . . . . .	68
9.5. Circuito 2 . . . . .	68
9.6. Circuito 2 reducido . . . . .	69
9.7. Circuito secuencial método de partición . . . . .	69
9.8. Equivalencia de estados mediante partición . . . . .	71
9.9. Tabla de implicación . . . . .	71
9.10. Partición de salida . . . . .	72
9.11. Tabla terminada . . . . .	72
9.12. Tabla terminada . . . . .	72
9.13. Partición de equivalencia . . . . .	73
10.1. Partes del contactor . . . . .	79
10.2. Accionamiento contactos temporizados . . . . .	84
11.1. Señal de mando y señal de salida . . . . .	90
11.2. Diagrama de tiempo función pulsador . . . . .	91
11.3. Diagrama Ladder función pulsador . . . . .	91
11.4. Bloque función pulsador . . . . .	91
11.5. Diagrama de tiempo función interruptor . . . . .	92
11.6. Diagrama Ladder función interruptor . . . . .	92
11.7. Bloque función interruptor . . . . .	93
11.8. Diagrama de tiempo función seguidor . . . . .	93
11.9. Diagrama Ladder función seguidor . . . . .	94
11.10. Bloque función seguidor . . . . .	94
11.11. Diagrama de tiempo función NOT . . . . .	95
11.12. Diagrama Ladder función NOT . . . . .	95
11.13. Bloque función NOT . . . . .	95
11.14. Diagrama de tiempo función AND . . . . .	96

11.15Diagrama Ladder función AND . . . . .	96
11.16Bloque función AND . . . . .	97
11.17Diagrama de tiempo función OR . . . . .	97
11.18Diagrama Ladder función OR . . . . .	97
11.19Bloque función OR . . . . .	98
11.20Diagrama de tiempo función temporización a la excitación . . . . .	98
11.21Diagrama Ladder función temporización a la excitación . . . . .	99
11.22Bloque función temporización a la excitación . . . . .	99
11.23Diagrama de tiempo función temporización a la excitación negada . . . . .	100
11.24Diagrama Ladder función temporización a la excitación negada . . . . .	100
11.25Bloque función temporización a la excitación negada . . . . .	101
11.26Diagrama de tiempo función temporización a la desexcitación . . . . .	101
11.27Diagrama Ladder función temporización a la desexcitación . . . . .	101
11.28Bloque función temporización a la desexcitación . . . . .	102
11.29Diagrama de tiempo función temporización a la desexcitación negada . . . . .	102
11.30Diagrama Ladder función temporización a la desexcitación negada . . . . .	103
11.31Bloque función temporización a la desexcitación negada . . . . .	103
11.32Diagrama de tiempo función detector flancos de subida . . . . .	104
11.33Diagrama Ladder función detector flancos de subida . . . . .	104
11.34Bloque función detector flancos de subida . . . . .	105
11.35Diagrama de tiempo función detector flancos de bajada . . . . .	105
11.36Diagrama Ladder función detector flancos de bajada . . . . .	106
11.37Bloque función detector flancos de bajada . . . . .	106
11.38Diagrama de tiempo función refresco . . . . .	107
11.39Diagrama Ladder función refresco . . . . .	107
11.40Bloque función refresco . . . . .	108
11.41Diagrama de tiempo función relé de enclavamiento . . . . .	108
11.42Diagrama Ladder función relé de enclavamiento . . . . .	109
11.43Diagrama de bloque función relé de enclavamiento . . . . .	109
11.44Diagrama de tiempo función relé de auto-retención . . . . .	110
11.45Diagrama Ladder función relé de auto-retención . . . . .	111
11.46Diagrama de bloque función relé de auto-retención . . . . .	111
11.47Diagrama de tiempo función temporización a la excitación y desexcitación . . . . .	112
11.48Diagrama Ladder función temporización a la excitación y desexcitación . . . . .	113
11.49Bloque función temporización a la excitación y desexcitación . . . . .	113
11.50Diagrama de tiempo función temporización a la excitación y desexcitación negada . . . . .	114
11.51Diagrama Ladder función temporización a la excitación y desexcitación negada . . . . .	114
11.52Bloque función temporización a la excitación y desexcitación negada . . . . .	115

11.53	Diagrama de tiempo función detector de flancos de subida y bajada	115
11.54	Diagrama Ladder función detector de flancos de subida y bajada	116
11.55	Bloque función detector de flancos de subida y bajada	116
11.56	Diagrama de tiempo función toggle	117
11.57	Diagrama Ladder función toggle	117
11.58	Bloque función toggle	118
11.59	Diagrama de tiempo función contador flancos de subida	118
11.60	Diagrama Ladder función contador flancos de subida	119
11.61	Bloque función contador flancos de subida	119
11.62	Diagrama de tiempo función contador flancos de bajada	120
11.63	Diagrama Ladder función contador flancos de bajada	120
11.64	Diagrama de bloque función contador flancos de bajada	121
11.65	Diagrama de tiempo función contador flancos de subida y bajada	121
11.66	Diagrama Ladder función contador flancos de subida y bajada	122
11.67	Diagrama de bloque función contador flancos de subida y bajada	122
11.68	Ejemplo 1	123
11.69	Mapa de Karnaugh encendido de la lámpara ejemplo 1	125
11.70	Mapa de Karnaugh apagado de la lámpara ejemplo 1	125
11.71	Diagrama de bloques ejemplo 1	126
11.72	Diagrama Ladder de cada bloque ejemplo 1	127
11.73	Diagrama Ladder simplificado ejemplo 1	129
11.74	Diagrama de bloques ejemplo 2	130
11.75	Diagrama Ladder de cada bloque ejemplo 2	131
11.76	Diagrama Ladder simplificado ejemplo 2	132
11.77	Ejemplo 3	133
11.78	Diagrama de bloques ejemplo 3	133
11.79	Diagrama Ladder de cada bloque ejemplo 3	134
11.80	Diagrama Ladder simplificado ejemplo 3	135
12.1.	Diagrama Ladder de una entrada mediante la utilización de un biestable	138
12.2.	Máquina de dos estados y una entrada	138
12.3.	Diagrama Ladder equivalente para una entrada de una máquina de estados	139
12.4.	Diagrama Ladder control de estados	139
12.5.	Diagrama Ladder bobina de activación	140
12.6.	Diagrama Ladder bobina de desactivación	140
12.7.	Diagrama Ladder bobina de activación modificado	140
12.8.	Diagrama ladder control de estados modificado	141
12.9.	Diagrama Ladder activación estado temporizado	141

12.10	Diagrama Ladder bobina de activación a través de un contacto temporizado . . . . .	142
12.11	Diagrama Ladder bobina de desactivación a través de un contacto temporizado . . . . .	142
12.12	Diagrama Ladder salidas directas . . . . .	143
12.13	Diagrama Ladder salida varios estados . . . . .	143
12.14	Inicialización máquina de estados a través de un pulsador . . . . .	144
12.15	Inicialización máquina de estados directa . . . . .	144
12.16	Diagrama de estados ejemplo 4 . . . . .	145
12.17	Entrada ejemplo 4 . . . . .	146
12.18	Bobinas de activación ejemplo 4 . . . . .	147
12.19	Bobinas de desactivación ejemplo 4 . . . . .	147
12.20	Bobinas estados ejemplo 4 . . . . .	147
12.21	Salida ejemplo 4 . . . . .	148
12.22	Sistema ejemplo 5 . . . . .	149
12.23	Diagrama de estados ejemplo 5 . . . . .	150
12.27	Bobinas de estados ejemplo 5 . . . . .	150
12.24	Entradas ejemplo 5 . . . . .	151
12.25	Bobinas de activación ejemplo 5 . . . . .	152
12.26	Bobinas de desactivación ejemplo 5 . . . . .	152
12.28	Salidas ejemplo 5 . . . . .	153
13.1.	Diagrama de estados ejemplo 6 . . . . .	158
13.2.	Entradas ejemplo 6 . . . . .	159
13.3.	Bobinas de activación ejemplo 6 . . . . .	160
13.4.	Bobinas de desactivación ejemplo 6 . . . . .	161
13.5.	Bobinas estados ejemplo 6 . . . . .	161
13.6.	Salidas ejemplo 6 . . . . .	162
13.7.	Sistema empacado de esferas . . . . .	163
13.8.	Entradas ejemplo 7 . . . . .	164
13.9.	Bobinas de estados ejemplo 7 . . . . .	164
13.10	Bobinas de activación ejemplo 7 . . . . .	165
13.11	Bobinas de desactivación ejemplo 7 . . . . .	165
13.12	Bobinas transición ejemplo 7 . . . . .	166
13.13	Salidas ejemplo 7 . . . . .	167
14.1.	Diagrama de estados ejemplo 8 . . . . .	169
14.2.	Entradas ejemplo 8 . . . . .	170
14.3.	Bobinas de activación ejemplo 8 . . . . .	170
14.4.	Bobinas de desactivación ejemplo 8 . . . . .	171

14.5. Bobinas de estados ejemplo 8 . . . . .	171
14.6. Salidas ejemplo 8 . . . . .	172
14.7. Diagrama de estados ejemplo 9 . . . . .	173
14.8. Entradas de la puerta ejemplo 9 . . . . .	173
14.9. Bobinas de activación de la puerta ejemplo 9 . . . . .	174
14.10 Bobina de desactivación de la puerta ejemplo 9 . . . . .	174
14.11 Bobinas de estados de la puerta ejemplo 9 . . . . .	174
14.12 Entradas del horno ejemplo 9 . . . . .	175
14.13 Bobinas de activación del horno ejemplo 9 . . . . .	175
14.14 Bobinas de desactivación del horno ejemplo 9 . . . . .	176
14.15 Bobinas de estados del horno ejemplo 9 . . . . .	176
14.16 Salidas del horno ejemplo 9 . . . . .	176
14.17 Entrada del modo de cocción ejemplo 9 . . . . .	177
14.18 Bobina de estado del modo de cocción ejemplo 9 . . . . .	177
14.19 Salida modo de cocción ejemplo 9 . . . . .	177
14.20 Diagrama de estados ejemplo 10.1 . . . . .	178
14.21 Diagrama de estados ejemplo 10.2 . . . . .	178
14.22 Diagrama de estados ejemplo 10.3 . . . . .	179
14.23 Diagrama de estados ejemplo 10.4 . . . . .	180
14.24 Diagrama de estados reducido ejemplo 10 . . . . .	181
14.25 Diagrama de estados secuencia de entrada correcta ejemplo 10 . . . . .	181
14.26 Diagrama de estados completo segundo desarrollo ejemplo 10 . . . . .	182
14.27 Diagrama de estados reducido segundo desarrollo ejemplo 10 . . . . .	183
14.28 Entradas ejemplo 10 . . . . .	183
14.29 Bobinas de activación ejemplo 10 . . . . .	184
14.30 Bobinas de desactivación ejemplo 10 . . . . .	185
14.31 Bobinas de estados ejemplo 10 . . . . .	185
14.32 Función contador para secuencia de entrada correcta . . . . .	186
14.33 Salidas ejemplo 10 . . . . .	186

# Índice de tablas

5.1. Tabla de verdad ejemplo términos Don't Care . . . . .	23
6.1. Minimización mediante Quine-McCluskey . . . . .	32
6.2. Implicantes primos contra Mintérminos . . . . .	34
6.3. I.P contra mintérminos con algoritmo de Petrick . . . . .	36
7.1. Compuerta lógica OR . . . . .	40
7.2. Compuerta lógica AND . . . . .	41
7.3. Compuerta lógica NOT . . . . .	42
7.4. Compuerta lógica NAND . . . . .	43
7.5. Compuerta lógica NOR . . . . .	44
7.6. Compuerta lógica XOR . . . . .	45
7.7. Compuerta lógica XNOR . . . . .	46
8.1. Tabla de transición para el Biestable S-R . . . . .	51
8.2. Tabla de excitación para el Biestable S-R . . . . .	51
8.3. Tabla de transición para el Biestable S-R con habilitación . . . . .	53
8.4. Tabla de excitación para el Biestable S-R con habilitación . . . . .	53
8.5. Tabla de transición del Biestable D . . . . .	55
8.6. Tabla de excitación para el Biestable D . . . . .	55
8.7. Tabla de transición del flip flop S-R maestro/esclavo . . . . .	57
8.8. Tabla de excitación del flip flop S-R maestro/esclavo . . . . .	57
8.9. Tabla de transición para el flip flop J-K maestro/esclavo . . . . .	59
8.10. Tabla de excitación del flip flop J-K maestro/esclavo . . . . .	59
8.11. Tabla de transición para el flip flop D maestro/esclavo . . . . .	61
8.12. Tabla de excitación del flip flop D . . . . .	61
8.13. Tabla de transición para el flip flop T . . . . .	62
8.14. Tabla de excitación del flip flop T . . . . .	62
9.1. Tabla de estados . . . . .	66
11.1. Tabla de verdad encendido de la lámpara ejemplo 1 . . . . .	124

11.2. Tabla de verdad apagado de la lámpara ejemplo 1 . . . . .	125
12.1. Tabla de estados ejemplo 4 . . . . .	146
12.2. Tabla de estados ejemplo 5 . . . . .	149
13.1. Tabla de estados ejemplo 6 . . . . .	158
14.1. Tabla de estados ejemplo 10 . . . . .	180
14.2. Tabla de estados reducida ejemplo 10 . . . . .	180
14.3. Tabla de estados segundo desarrollo ejemplo 10 . . . . .	182
14.4. Tabla de estados reducida segundo desarrollo ejemplo 10 . . . . .	182

# NOTACIONES

$x, y, z$	Últimas letras del alfabeto minúsculas para representar variables
$a, b, c$	Primeras letras del alfabeto minúsculas para representar constantes
$\wedge$	Cuantificador universal, se lee “para todo”
$\vee$	Cuantificador existencial, se lee “existe”
$ $	Descriptor, se lee “tal que”
$\bar{X}$	Negación, se lee “no”
$\bullet$	Conjunción, se lee “Y”
$+$	Disyunción, se lee “O”
$\oplus$	Conectiva lógica XOR
$\odot$	Conectiva lógica XNOR
$\rightarrow$	Implicación, se lee “si entonces”
$\leftrightarrow$	Doble implicación o equivalencia, se lee “si y sólo si”
$\in$	Pertenece
$>$	Símbolo que identifica el estado inicial de una máquina de estados finitos



**Parte I.**

# **INTRODUCCIÓN**



# 1. DEFINICIÓN DEL PROBLEMA

Los automatismos a través del tiempo han alcanzado un tamaño y una complejidad considerables debido a la industrialización del mundo moderno que busca realizar rutinas de producción con mayor eficiencia y en condiciones óptimas de seguridad tanto para las personas como para las máquinas, lo que ha llevado a buscar herramientas para el diseño de circuitos automáticos que logren satisfacer dichas necesidades.

Con el pasar de los años se han logrado implementar circuitos automáticos que realizan funciones básicas con la interconexión de pulsadores, interruptores, contactos, temporizadores, bobinas; y lo que se busca es que estos sirvan como punto de partida para el diseño final de los automatismos requeridos, lo que los ha convertido en un estándar. Cabe anotar que el ser un estándar no significa que se puedan utilizar en todo tipo de aplicaciones, por lo que se hace necesario acondicionar los circuitos dependiendo de la aplicación que se quiera, de ahí la dificultad de los diseñadores para encontrar el circuito que cumpla con los requerimientos exigidos.

En la actualidad las metodologías de diseño se han visto representadas en la prueba de circuitos simples a los cuales se le adicionan ciertas características que van siendo probadas para encontrar fallas e ir corrigiéndolas hasta encontrar el funcionamiento deseado, pero no se cuenta con una metodología que guíe al diseñador paso a paso, es decir, si se busca un circuito que permita implementar una acción, el encargado de su implementación se enfrenta a un problema que sólo podrá superar por medio de su habilidad y experiencia en el campo.

Cuando se diseñan circuitos Ladder con contactos temporizados surge un problema adicional ya que estas acciones no pertenecen a una descripción combinacional como comportamiento de un circuito.

Al momento de utilizar las nuevas plataformas para diseños de bajo costo, se presentan algunos problemas debido a que estas sólo admiten el lenguaje Ladder; si se tiene en cuenta que este tipo de programación contiene un gran nivel de razonamiento lógico y es altamente dependiente de la experiencia, se hace muy difícil para una persona inexperta brindar soluciones a través de este método sin antes haber realizado algunos estudios y obtener ciertos conocimientos; más aún al querer ajustar metodologías de diseño como las redes de Petri o las máquinas de estado a una programación de este tipo.



## 2. JUSTIFICACIÓN

La automatización se concibe en nuestros días como indispensable en el desarrollo de la industria debido a que esta permite la competitividad de las empresas en cuanto a tiempo de elaboración y calidad de sus productos, además de propiciar un ahorro en el costo de producción en cuanto a operarios se refiere. Es importante destacar que los procesos automáticos se encuentran en distintas formas desde unas básicas como el encendido de un motor hasta otras más complejas como la elaboración de automóviles; motivo por el cual en el diseño de automatismos se necesitan resultados óptimos y rápidos al momento de querer realizar nuevos procesos de producción.

Teniendo en cuenta lo anterior se hace necesario plantear una rutina de diseño que permita a partir de un conjunto de requerimientos previamente señalados, obtener un circuito en lenguaje Ladder capaz de ejecutar las acciones solicitadas, ya que a pesar del avance en la automatización de procesos, hoy en día no se cuenta con un método que asegure que con seguir unas instrucciones se podrá llegar a un circuito o diseño esperado, evitando así la pérdida de tiempo y el incremento del estrés no deseado por estos días.

Existen nuevas tecnologías como los relés inteligentes Zelio Logic proveniente de una marca registrada como Telemecanique y diseñados para pequeños sistemas de automatismos. Se usan tanto en los sectores de la industria como en el de servicios. Su tamaño compacto y facilidad de ajuste hacen de ellos una alternativa a las soluciones basadas en lógica de cableado o tarjetas específicas. La simplicidad de su programación, es garantizada por el uso de dos lenguajes (LADDER y FBD), cumpliendo así con las exigencias en la automatización y con las expectativas del electricista. Cabe resaltar que en la gama baja de los relés Zelio el lenguaje Ladder es el único aceptado para su programación, de ahí la importancia de contar con una metodología de diseño.[1]

Los relés inteligentes Compactos son convenientes para sistemas de automatismos simples de hasta 20 E/S. Si es requerido, en los relés modulares pueden agregarse extensiones de E/S y un módulo de comunicación Modbus con un mayor desempeño y flexibilidad, desde 10 hasta 40 E/S.[1]

Los PLC Koyo son los primeros micro PLC que dependiendo de su aplicación (discreta, análoga o módulos de comunicación) pueden combinar 20 entradas y 16 salidas con cuatro tarjetas de expansión en un mismo conjunto. Tienen la más flexible configuración de puertos en su clase y cuentan con los

mismos protocolos de comunicación que otras familias de PLC's, [1] estos independientemente de su tamaño o módulos adicionales, se programan exclusivamente en Ladder.

Hoy en día los PLC o Controlador de lógica programable, son dispositivos electrónicos muy usados en la automatización industrial. Su historia se remonta a finales de la década de 1960 cuando la industria buscó en las nuevas tecnologías electrónicas una solución más eficiente para reemplazar los sistemas de control basados en circuitos eléctricos con relés, interruptores y otros componentes comúnmente utilizados para el control de los sistemas de lógica combinacional. Los PLC actuales pueden comunicarse con otros controladores y computadoras en redes de área local, y son una parte fundamental de los modernos sistemas de control distribuido, en cuanto a la programación de estos se hace presente el lenguaje Ladder, preferido por los electricistas, el cual consta de una lista de instrucciones y programación por estados. Dado que los PLC de clase económica sólo se pueden programar haciendo uso del Ladder, además de los problemas al momento de implementar temporizaciones, se deben buscar metodologías que faciliten el diseño sin depender fuertemente de la experiencia del diseñador.

De lo anterior nace la siguiente pregunta de investigación: ¿Es posible describir el comportamiento de automatismos con base en metodologías secuenciales y combinacionales con el fin de permitir implementaciones en lenguaje Ladder?.

Los diagramas Ladder presentados en el desarrollo de este trabajo de grado han sido diseñados bajo la norma IEC 61131-3.





## **3. OBJETIVOS**

### **3.1. OBJETIVO GENERAL**

- Desarrollar metodologías para diseño de circuitos Ladder con base en sistemas secuenciales y combinacionales, señales de temporización y máquinas de estados finitos.

### **3.2. OBJETIVOS ESPECÍFICOS**

- Describir de manera clara una metodología de diseño Ladder con base en circuitos combinacionales y en señales de temporización.
- Exponer claramente una metodología para diseño de circuitos Ladder con base en sistemas secuenciales y máquinas de estados finitos.
- Desarrollar un modelo híbrido que integre las metodologías anteriores para diseñar sistemas con circuitos Ladder más complejos.



**Parte II.**

# **PRELIMINARES**



## 4. LÓGICA DE PREDICADOS

El instrumento fundamental de comunicación humana es el lenguaje, formado por frases de tipo interrogativo, imperativo y declarativo. Estas últimas constituyen el elemento básico de descripción del conocimiento.

La lógica es la disciplina que estudia los métodos de formalización del conocimiento humano. En lógica se estudian, por tanto, métodos de formalización de frases declarativas. Para ello existen dos niveles de abstracción según el grado de detalle que se quiera formalizar: lógica proposicional y lógica de predicados.[18]

La lógica proposicional o lógica de enunciados toma como elemento básico las frases declarativas simples o proposiciones que son aquellos elementos de una frase que constituyen por si solos una unidad de comunicación de conocimientos y pueden ser considerados verdaderos o falsos.[18] La lógica de predicados estudia las frases declarativas con mayor grado de detalle, considerando la estructura interna de las proposiciones. Se toman como elementos básicos, los objetos y las relaciones entre dichos objetos, es decir, se distingue:[18]

- Qué se afirma (predicado o relación)
- De quién se afirma (objeto)

El alfabeto de la lógica de predicados está formado por los siguientes conjuntos de símbolos:[2, 17]

**Variabes:** está formado por las últimas letras del alfabeto minúsculas. También se utilizan subíndices, por ejemplo:  $x, y, z, x_1, y_1, z_1, \dots, x_n, y_n, z_n$ .

**Constantes:** primeras letras del alfabeto minúsculas (con subíndice), por ejemplo:  $a, b, c, a_1, b_1, c_1, \dots, a_n, b_n, c_n$ .

**Funciones:** está formado por las letras  $f, g, h$ . También se puede incluir subíndices para diferenciar distintas funciones:  $f_1, g_1, h_1, \dots, f_n, g_n, h_n$ .

**Relatores:** se representa mediante letras mayúsculas por ejemplo P, Q, R, K. Igualmente se puede indicar la aridad<sup>1</sup> de un relator por medio de un superíndice.

---

<sup>1</sup>La aridad de una función o de un predicado se define como el número de argumentos que tiene.

**Cuantificadores:** son signos que brindan al lenguaje formal una mayor fuerza. Se emplean conjuntamente con las variables y principalmente son dos clases. El cuantificador universal indica que algo es cierto para todos los individuos. Sea  $A$  una expresión y  $x$  una variable, si se desea indicar que  $A$  es verdadero para todos los posibles valores de  $x$ , se escribe  $(\forall x) A$ . en donde  $(\forall x)$  es el cuantificador universal,  $A$  es el ámbito<sup>2</sup> del cuantificador, y el símbolo  $\forall$  se lee “para todo”. Por otro lado se tiene el cuantificador existencial ( $\exists$ ) el cual se lee “existe” por ejemplo  $(\exists x)$  algo. Quiere decir que existe un  $x$  de manera que ese  $x$  pertenece a algo, de forma general, ese algo es cierto si la variable  $x$  se interpreta de forma adecuada.

**Funtor:** un funtor es un signo que lleva objetos a objetos, en contraste con un relator que da lugar a una afirmación.

**Descriptor:** se representa por  $(|)$  y se lee “*tal que*”.

Cuando se trata con proposiciones más complejas se hace necesario el uso de los conectivos lógicos con el objetivo de enlazar las frases declarativas simples, estos conectivos lógicos son:

**Negación ( $\neg$ ):** se lee como “*no*” o “*es falso que*”.

**Conjunción ( $\bullet$ ):** se lee como “*Y*”.

**Disyunción ( $+$ ):** se lee como “*O*”, en algunos casos la disyunción tiene la interpretación como, lo uno, lo otro o ambos, y en otras ocasiones se interpreta como, lo uno, lo otro, pero no ambos, en la lógica de predicados se emplea en el primero de los sentidos.

**Implicación ( $\rightarrow$ ):** se lee como “*si entonces*”.

**Doble implicación o equivalencia ( $\leftrightarrow$ ):** se lee como “*si y sólo si*”, quiere decir, que lo que es válido para una afirmación lo es para la otra.

---

<sup>2</sup>Contexto que tiene una expresión o valor dentro de una oración.





## 5. ÁLGEBRA DE BOOLE

El álgebra booleana, estudiada por primera vez en detalle por George Boole, constituye un área de las matemáticas que ha pasado a ocupar un lugar prominente con el advenimiento de la computadora digital. Es usada ampliamente en el diseño de circuitos de distribución y computadoras, y sus aplicaciones van en aumento en muchas otras áreas. En el nivel de lógica digital de una computadora, lo que comúnmente se llama hardware, y que está formado por los componentes electrónicos de la máquina, se trabaja con diferencias de tensión, las cuales generan funciones que son calculadas por los circuitos que forman el nivel. Éstas funciones, en la etapa de diseño del hardware, son interpretadas como funciones de Boole.

En el álgebra de Boole se tratan las funciones booleanas, haciendo una correlación con las fórmulas proposicionales. Así mismo, se plantean dos formas canónicas de las funciones booleanas, que son útiles para varios propósitos, tales como el de determinar si dos expresiones representan o no la misma función. Pero para otros propósitos son a menudo engorrosas, por tener más operaciones de las necesarias. Particularmente, cuando se está construyendo un circuito electrónico con el cual implementar funciones booleanas, el problema de determinar una expresión mínima para una función es a menudo crucial. No resultan de la misma eficiencia en dinero y tiempo, principalmente, dos funciones las cuales calculan lo mismo pero donde una tiene menos variables y lo hace en menor tiempo. Como solución a este problema, se plantea un método de simplificación, que hace uso de unos diagramas especiales llamados mapas o diagramas de Karnaugh, y el cual tiene la limitación de poder trabajar adecuadamente sólo con pocas variables. [19]

### 5.1. POSTULADOS DEL ÁLGEBRA DE BOOLE

#### Postulado 1: Presentación

El álgebra de Boole  $(\vec{B})$ , es un sistema cerrado distribuido y complementario tal que se cumple:

$$a, b \in \vec{B} \mid a + b \in \vec{B} \wedge a \bullet b \in \vec{B}$$

**Postulado 2: Existencia de elementos neutros**

$$a \in \vec{B} \mid a+0 = a \wedge a \bullet 1 = a$$

**Postulado 3: Conmutatividad**

$$a, b \in \vec{B} \mid a+b = b+a \wedge a \bullet b = b \bullet a$$

**Postulado 4: Asociatividad**

$$a, b, c \in \vec{B} \mid a+(b+c) = (a+b)+c \wedge a \bullet (b \bullet c) = (a \bullet b) \bullet c$$

**Postulado 5: Distributividad**

$$a, b, c \in \vec{B} \mid a+(b \bullet c) = (a+b)(a+c) \wedge a \bullet (b+c) = a \bullet b + a \bullet c$$

**Postulado 6: Complemento**

$$\text{Para todo } a \in \vec{B} \mid \forall \bar{a} \in \vec{B} \text{ se cumple } a+\bar{a} = 1 \text{ y } a \bullet \bar{a} = 0$$

## 5.2. PRINCIPIO DE DUALIDAD

Establece que toda expresión verdadera en el álgebra de Boole, posee una expresión dual verdadera, más no necesariamente igual. La expresión dual se obtiene reemplazando el operador  $+$  por el operador  $\bullet$ , el operador  $\bullet$  por el operador  $+$ , los 1 por 0, los 0 por 1, y conservando las precedencias relacionadas por los paréntesis.[2]

## 5.3. TEOREMAS DEL ÁLGEBRA DE BOOLE

**Teorema 1: Idempotencia**

$$a+a = a$$

$$a \bullet a = a$$

**Teorema 2: Elementos Neutros**

$$a + 1 = 1$$

$$a \bullet 0 = 0$$

**Teorema 3: Involución**

$$\overline{\overline{a}} = a$$

$$\overline{\overline{\overline{a}}} = \overline{a}$$

**Teorema 4: Primer Teorema de Absorción**

$$a + a \bullet b = a$$

$$a \bullet (a + b) = a$$

**Teorema 5: Segundo Teorema de Absorción**

$$a + \overline{a} \bullet b = a + b$$

$$a \bullet (\overline{a} + b) = a \bullet b$$

**Teorema 6: Tercer Teorema de Absorción**

$$a \bullet b + a \bullet \overline{b} = a$$

$$(a + b) \bullet (a + \overline{b}) = a$$

**Teorema 7: Cuarto Teorema de Absorción**

$$a \bullet b + a \bullet \overline{b} \bullet c = a \bullet b + a \bullet c$$

$$(a + b) \bullet (a + \bar{b} + c) = (a + b) \bullet (a + c)$$

**Teorema 8: Teorema de DeMorgan**

$$\overline{a + b} = \bar{a} \bullet \bar{b} \implies \overline{a + b + c + d + \dots} = \bar{a} \bullet \bar{b} \bullet \bar{c} \bullet \bar{d} \bullet \dots$$

$$\overline{a \bullet b} = \bar{a} + \bar{b} \implies \overline{a \bullet b \bullet c \bullet d \bullet \dots} = \bar{a} + \bar{b} + \bar{c} + \bar{d} + \dots$$

Este teorema define realmente dos nuevas funciones lógicas de gran importancia que son utilizadas como elementos básicos para la realización de los sistemas digitales. Estas dos funciones vistas anteriormente, se denominan respectivamente NOR y NAND.

Las tres funciones elementales: suma, producto e inversión lógica pueden ser realizadas mediante las funciones NOR y NAND.[20]

**Teorema 9: Teorema de Consenso**

$$a \bullet b + \bar{a} \bullet c + b \bullet c = a \bullet b + \bar{a} \bullet c$$

$$(a + b) \bullet (\bar{a} + c) \bullet (b + c) = (a + b) \bullet (\bar{a} + c)$$

**Teorema 10: Teorema de Desarrollo de Shannon**

$$f(x_1, x_2, \dots, x_n) = x_1 \bullet f(1, x_2, \dots, x_n) + \bar{x}_1 \bullet f(0, x_2, \dots, x_n)$$

$$f(x_1, x_2, \dots, x_n) = [x_1 + f(0, x_2, \dots, x_n)] \bullet [\bar{x}_1 + f(1, x_2, \dots, x_n)]$$

**5.4. FUNCIONES BOOLEANAS**

Una función del álgebra de Boole es una variable binaria cuyo valor es igual al de una expresión algebraica en la que se relacionan entre si las variables binarias por medio de las operaciones básicas. Producto lógico, suma lógica e inversión.

Se representa una función lógica por la expresión  $f(a, b, c, \dots)$ ; el valor lógico de  $f$ , depende de las variables  $a, b, c, \dots$

Se llama término canónico de una función lógica a todo producto o suma en la cual aparece todas las variables en su forma directa o inversa una sola vez. Al primero de ellos se les llama producto canónico (mintérminos) y al segundo suma canónica (maxtérminos).[21]

### 5.4.1. Forma canónica suma de productos

Es aquella constituida exclusivamente por términos canónicos productos (mintérminos) sumados que aparecen una sola vez.

*Ejemplo:*  $F(x, y, z) = \bar{x} \bullet \bar{y} \bullet z + x \bullet \bar{y} \bullet \bar{z} + x \bullet \bar{y} \bullet z + x \bullet y \bullet \bar{z} + x \bullet y \bullet z$

Para simplificar la escritura en forma de suma de productos, se utiliza una notación especial. A cada mintérmino se le asocia un número binario de  $n$  bits resultantes de considerar como 0 las variables complementadas y como 1 las variables no complementadas. Así por ejemplo el mintérmino  $\bar{x} \bullet \bar{y} \bullet z$  corresponde a combinación  $x = 0, y = 0, z = 1$  que representa el número binario 001, cuyo valor decimal es 1. Este mintérmino se identifica entonces como  $m_1$ .

De esta forma la función:  $F(x, y, z) = \bar{x} \bullet \bar{y} \bullet z + x \bullet \bar{y} \bullet \bar{z} + x \bullet \bar{y} \bullet z + x \bullet y \bullet \bar{z} + x \bullet y \bullet z$

Se puede expresar como:  $F(x, y, z) = \sum m(1, 4, 5, 6, 7)$  que quiere decir la sumatoria de los mintérminos 1, 4, 5, 6, 7.[21]

### 5.4.2. Forma canónica producto de sumas

Es aquella constituida exclusivamente por términos canónicos sumas (maxtérminos) multiplicados que aparecen una sola vez.

*Ejemplo:*  $F(x, y, z) = (x + y + z) \bullet (x + \bar{y} + z) \bullet (x + \bar{y} + \bar{z})$

Análogamente al caso anterior, se puede simplificar la expresión de la función, indicando los maxtérminos. Sin embargo, en este caso se hace al contrario que antes. A cada maxtérmino se le asocia un número binario de  $n$  bits resultantes de considerar como 1 las variables complementadas y como 0 las variables no complementadas. Así por ejemplo el maxtérmino  $x + y + z$  corresponde a combinación  $x = 0, y = 0, z = 0$  que representa el número binario 000, cuyo valor decimal es 0. A este maxtérmino se identifica entonces como  $M_0$ .

De esta forma, la función:  $F(x, y, z) = (x + y + z) \bullet (x + \bar{y} + z) \bullet (x + \bar{y} + \bar{z})$

Se puede expresar como:  $F(x, y, z) = \prod M(0, 2, 3)$  que quiere decir el producto de los maxtérminos 0, 2, 3.[21]

### 5.4.3. Términos Don't Care

La especificación básica de una función de conmutación (función booleana) es la tabla de verdad, que muestra la lista de todas las combinaciones posibles de las variables y el valor que asumirá la o las salidas para todas esas combinaciones. Hasta ahora se ha supuesto que los valores de verdad se especifican estrictamente para todas las  $2^n$  combinaciones de entradas posibles, siendo  $n$  el número de variables de entrada. Sin embargo, no siempre es así. Existe la posibilidad que ciertas combinaciones de entrada, debido a restricciones externas, no se produzcan nunca. Esto no quiere decir que si estas entradas prohibidas se producen, el circuito no responde de alguna forma, de hecho cualquier circuito de conmutación responde de alguna forma a cualquier entrada. Sin embargo, dado que la entrada no puede ocurrir nunca, no importa si el circuito responde a la salida con un *cerro* o con un *uno* a esta combinación de entrada prohibida. Cuando se presentan estas situaciones se dice que la salida es NO IMPORTA (Don't care en inglés). Esto se indica en la tabla de verdad y en el mapa de Karnaugh correspondiente con una  $d$  en lugar del 1 o 0.[2, 22]

*Ejemplo.* Diseñar un circuito que detecte los números primos entre 1 y 9.

Para representar los números entre el 1 y el 9 se necesitan 4 bits. Suponiendo  $A, B, C, D$ , siendo  $A$  el bit más significativo.

Se realiza la tabla de verdad colocando un 1 en los números primos del 1 al 9 y un 0 en los números que no sean primos. Nótese que el rango de combinaciones que se quiere cubrir está entre 1 y 9 ambos inclusive, lo que hace un total de 9 combinaciones.

Con 4 variables se puede tener 16 combinaciones, por lo que existen 7 combinaciones para las cuales "no importa" (don't care) la entrada, porque nunca se van a dar. En esas combinaciones se coloca una  $d$  en la tabla 5.1.

Tabla 5.1.: Tabla de verdad ejemplo términos Don't Care

Número	A	B	C	D	f
0	0	0	0	0	<i>d</i>
1	0	0	0	1	1
2	0	0	1	0	1
3	0	0	1	1	1
4	0	1	0	0	0
5	0	1	0	1	1
6	0	1	1	0	0
7	0	1	1	1	1
8	1	0	0	0	0
9	1	0	0	1	0
10	1	0	1	0	<i>d</i>
11	1	0	1	1	<i>d</i>
12	1	1	0	0	<i>d</i>
13	1	1	0	1	<i>d</i>
14	1	1	1	0	<i>d</i>
15	1	1	1	1	<i>d</i>



## 6. SIMPLIFICACIÓN DE FUNCIONES

### 6.1. MÉTODO ALGEBRAICO

Para la simplificación por este método no sólo basta con conocer todas las propiedades y teoremas del álgebra de Boole, además se debe desarrollar una cierta habilidad lógico matemática.

Se comienza simplificando la siguiente función:

$$F = \bar{A} \cdot \bar{C} + A \cdot B \cdot C + B \cdot \bar{C} + \bar{A} \cdot \bar{B} \cdot C + \bar{A} \cdot B \cdot C$$

Observando cada uno de los sumandos se puede ver que hay factores comunes en los sumandos 2<sup>do</sup> con 5<sup>to</sup> y 4<sup>to</sup> con 5<sup>to</sup> que conllevan simplificación. Luego:

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

El término 5<sup>to</sup> se ha tomado dos veces, de acuerdo con el teorema 3 visto anteriormente  $a + a = a$ , y aplicando el teorema 6 también ya descrito se tiene:

$$F = \bar{A} \cdot \bar{C} + B \cdot \bar{C} + B \cdot C + \bar{A} \cdot C$$

Repitiendo nuevamente el proceso:

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

$$F = \bar{A} + B$$

Como se puede apreciar, el método algebraico no resulta cómodo y lo que es peor, una vez simplificada una ecuación pueden quedar serias dudas acerca de si se ha conseguido simplificarla al máximo. El método gráfico resuelve estos inconvenientes.

## 6.2. MAPAS DE KARNAUGH

Si se quiere realizar eficazmente la simplificación de las funciones de conmutación se debe contar con un método sistemático que proporcione un camino para lograr el objetivo de manera segura, un método de este tipo son los mapas de Karnaugh, que pueden ser aplicados en funciones de conmutación hasta de seis variables.

Los mapas de Karnaugh no son más que una extensión de los conceptos de las tablas de verdad, diagramas de Venn y mintérminos, lo que se evidencia en la transformación de un diagrama de Venn en un mapa de Karnaugh.

Se considera un diagrama de Venn de dos variables  $A$  y  $B$  (figura 6.1a) representadas mediante las subdivisiones del conjunto universal.

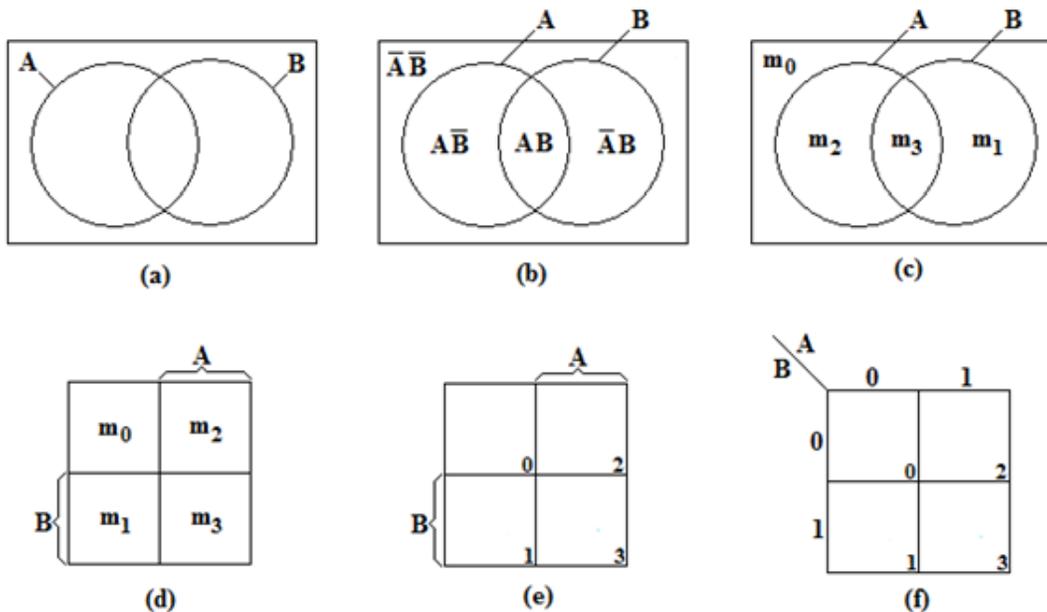


Figura 6.1.: Diagrama de Venn y mapa de Karnaugh equivalente para dos variables

En la figura 6.1b se observan las subdivisiones ajenas únicas del diagrama de Venn representadas por las intersecciones:  $AB$ ,  $A\bar{B}$ ,  $\bar{A}B$ ,  $\bar{A}\bar{B}$ ; siendo estas últimas no más que los mintérminos de dos variables:  $m_0$ ,  $m_1$ ,  $m_2$ ,  $m_3$  (figura 6.1c). Se puede ajustar las áreas del diagrama de Venn de manera que todas sean iguales (figura 6.1d) conservando la característica de que las áreas adyacentes en el diagrama de Venn también lo son en la figura 6.1d, pero ahora una mitad del diagrama representa a la variable  $A$  y la otra mitad a la variable  $B$ . Puesto que cada cuadrado representa un mintérmino se puede omitir

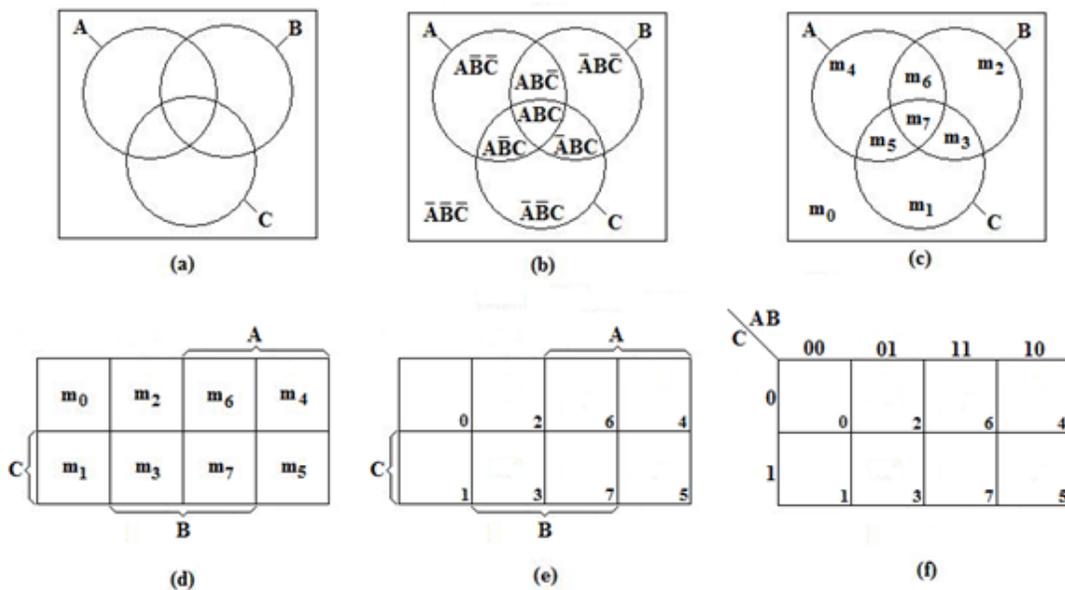


Figura 6.2.: Diagrama de Venn y mapa de Karnaugh equivalente para tres variables

la letra  $m$  y dejar sólo el subíndice (figura 6.1e) siendo esta una forma del mapa de Karnaugh. En la figura 6.1f se puede observar otra forma del mapa de Karnaugh en el que la asociación de un cuadrado de un mapa con una variable en particular, por ejemplo  $A$ , se indica como 0 para  $\bar{A}$  y 1 para  $A$ .

Es de notar la correspondencia de los mapas de Karnaugh con las tablas de verdad ya que por cada mintermino existe una fila en la tabla de verdad, mientras que en el diagrama existe un cuadrado; esta observación se extiende también para los maxtérminos.

En la figura 6.2 se observa el desarrollo de un mapa de Karnaugh para tres variables. Algo que se debe tener en cuenta es que en este mapa tanto los extremos izquierdo y derecho como los extremos superior e inferior se deben considerar como la misma línea y así lograr que los minterminos que son adyacentes en el diagrama de Venn lo sean en el mapa de Karnaugh.

Para simplificar una función lógica por el método de Karnaugh se llevan a cabo los siguientes pasos: [23]

1. Se dibuja el diagrama correspondiente al número de variables de la función a simplificar.
2. Se coloca un 1 en los cuadros correspondientes a los términos canónicos que forman parte de la función en el caso de los minterminos, mientras que cuando se trabaja con maxtérminos se pone un 0.
3. Se agrupan mediante lazos los 1 de casillas adyacentes siguiendo estrictamente las siguientes reglas:

- a) Dos casillas son adyacentes cuando se diferencian únicamente en el estado de una sola variable.
  - b) Cada lazo debe contener el mayor número de 1 posibles, siempre que dicho número sea potencia de 2 (1, 2, 4, etc.).
  - c) Los lazos pueden quedar superpuestos y no importa que haya cuadrículas que pertenezcan a dos o más lazos diferentes.
  - d) Se debe tratar de conseguir el menor número de lazos con el mayor número de 1 posibles.
4. La función simplificada tendrá tantos términos como lazos posea el diagrama. Cada término se obtiene eliminando la o las variables que cambien de estado en el mismo lazo.

*Ejemplo.* Para aclarar la teoría descrita anteriormente se realizan dos simplificaciones de una misma función a partir de sus dos formas canónicas.

$$F = \sum m(0, 2, 3, 4, 7)$$

$$F = \prod (1, 5, 6)$$

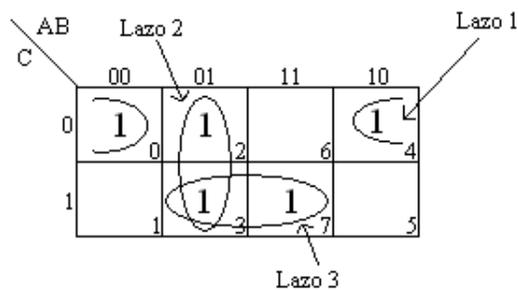


Figura 6.3.: Mapa de Karnaugh función  $F = \sum m(0, 2, 3, 4, 7)$

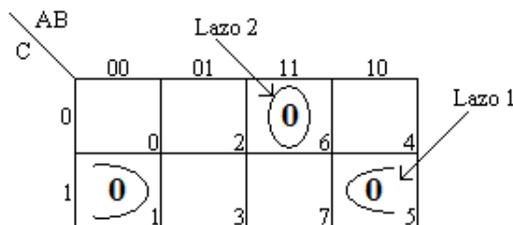


Figura 6.4.: Mapa de Karnaugh función  $F = \prod (1, 5, 6)$

La función simplificada tiene tres sumandos en un caso y dos productos en el otro. Al examinar el mapa de Karnaugh correspondiente a la suma de productos (figura 6.3), se observa que en el lazo 1 cambia la variable  $A$  (en la celda 0 es negada y en la 4 directa), en el lazo 2 es la  $C$  y en el lazo 3 vuelve a ser  $A$ . Teniendo en cuenta que en el caso de los mintérminos una variable negada corresponde a 0 y una variable directa corresponde a 1, por lo tanto, la ecuación simplificada es:

$$F = \bar{B} \cdot \bar{C} + \bar{A} \cdot B + B \cdot C$$

Razonando de modo similar en el mapa de productos de sumas (figura 6.4), con la salvedad que en el caso de los maxtérminos una variable negada corresponde a 1 y una variable directa corresponde a 0, se obtiene:

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

Al igual que en la tabla de verdad los términos don't care se representan en el mapa de Karnaugh con una  $d$  y se utiliza como un comodín, ya que puede hacerse valer como un 0 o 1 según convenga a la hora de minimizar. Cuando se quiere simplificar una función utilizando mapas de Karnaugh, estas condiciones de don't care ayudan a formar grupos de "unos" más grandes que generan términos con productos menores.[2, 22]

Retomando el ejemplo de los *números primos* mostrado anteriormente en la subsección 5.4.3 se realiza la simplificación de esta función mediante el mapa de Karnaugh.

De la tabla de verdad se obtiene el mapa de Karnaugh (figura 6.5), colocando los 1 y las  $d$ .

	AB			
CD	00	01	11	10
00	$d$ 0		$d$ 12	
01	1 1	1 5	$d$ 13	
11	1 3	1 7	$d$ 15	$d$ 11
10	1 2		$d$ 14	$d$ 10

Figura 6.5.: Mapa de Karnaugh términos don't care

Se muestra la diferencia de tomar las  $d$  en el proceso de agrupación como más convenga para la minimización. Ya que las combinaciones indicadas con  $d$  no importan, porque nunca van a estar presentes, se toman como 1 o 0 si ayudan a obtener un menor número de términos o términos con menos literales.

AB \ CD	00	01	11	10
00	$d$ <sub>0</sub>		$d$ <sub>12</sub>	
01	1 <sub>1</sub>	1 <sub>5</sub>	$d$ <sub>13</sub>	
11	1 <sub>3</sub>	1 <sub>7</sub>	$d$ <sub>15</sub>	$d$ <sub>11</sub>
10	1 <sub>2</sub>		$d$ <sub>14</sub>	$d$ <sub>10</sub>

Figura 6.6.: Minimización mapa de Karnaugh sin usar términos Don't Care

Sin incluir los términos don't care en la simplificación (figura 6.6) se obtiene:

$$F = \bar{A} \cdot \bar{B} \cdot C + \bar{A} \cdot D$$

AB \ CD	00	01	11	10
00	$d$ <sub>0</sub>		$d$ <sub>12</sub>	
01	1 <sub>1</sub>	1 <sub>5</sub>	$d$ <sub>13</sub>	
11	1 <sub>3</sub>	1 <sub>7</sub>	$d$ <sub>15</sub>	$d$ <sub>11</sub>
10	1 <sub>2</sub>		$d$ <sub>14</sub>	$d$ <sub>10</sub>

Figura 6.7.: Minimización mapa de Karnaugh usando términos Don't Care

Al incluir los términos don't care en la simplificación (figura 6.7) se obtiene:

$$F = \bar{A} \cdot \bar{B} + \bar{A} \cdot D$$

### 6.3. QUINE-McCLUSKEY

El Algoritmo Quine–McCluskey es un método de simplificación de funciones booleanas desarrollado por Willard Van Orman Quine y Edward J. McCluskey. Es funcionalmente idéntico a la utilización del mapa de Karnaugh, pero su forma tabular lo hace más eficiente para su implementación en lenguajes computacionales, y provee un método determinístico de conseguir la mínima expresión de una función booleana.[12]

El método consta de dos pasos:

1. Encontrar todos los implicantes primos de la función.
2. Usar esos implicantes en una tabla de implicantes primos para encontrar los implicantes primos esenciales, los cuales son necesarios y suficientes para generar la función.

Aunque es más práctico que el mapa de Karnaugh, cuando se trata de trabajar con más de cuatro variables, el tiempo de resolución del algoritmo Quine-McCluskey crece de forma exponencial con el aumento del número de variables. Se puede demostrar que para una función de  $n$  variables, el límite superior del número de implicantes primos es  $3^n/n$ . Si  $n = 32$  habrá más de  $5,75 \bullet 10^{15}$  implicantes primos. Funciones con un número grande de variables tienen que ser minimizadas con otros métodos heurísticos.[12]

*Ejemplo.* Minimizar las funciones:

$$f\alpha(A, B, C, D) = \sum m(0, 2, 7, 10) + d(12, 15)$$

$$f\beta(A, B, C, D) = \sum m(2, 4, 5) + d(6, 7, 8, 10)$$

$$f\gamma(A, B, C, D) = \sum m(2, 7, 8) + d(0, 5, 13)$$

Utilizando mapas de Karnaugh para cada función (figura 6.8) se obtienen los siguientes resultados:

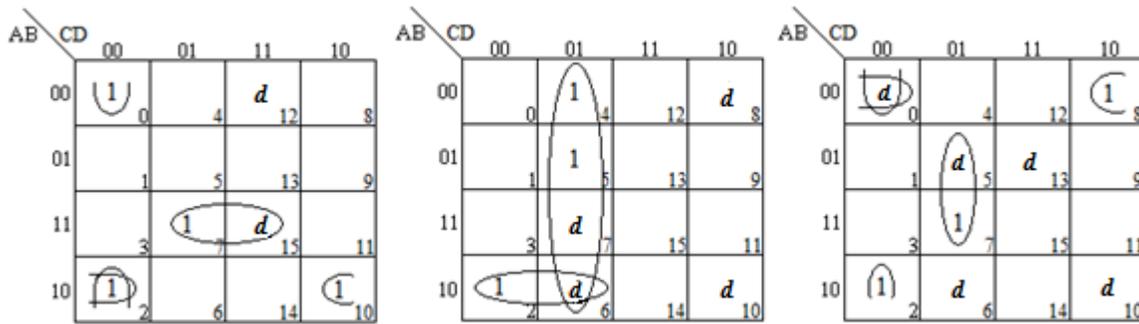


Figura 6.8.: Mapas de Karnaugh para el ejemplo de Quine-McCluskey

De donde:

$$f\alpha = \bar{A} \cdot \bar{B} \cdot \bar{D} + \bar{B} \cdot C \cdot \bar{D} + B \cdot C \cdot D$$

$$f\beta = \bar{A} \cdot B + \bar{A} \cdot C \cdot \bar{D}$$

$$f\gamma = \bar{A} \cdot B \cdot D + \bar{B} \cdot \bar{C} \cdot \bar{D} + \bar{A} \cdot \bar{B} \cdot \bar{D}$$

En la tabla 6.1 se muestra la minimización mediante Quine-McCluskey.

Tabla 6.1.: Minimización mediante Quine-McCluskey

1s	Minterminos	1er Cubo	2do Cubo
0	$m_0$ 0000 $\alpha\gamma^*$	$m_0, m_2$ 00 – 0 $\alpha\gamma$ IP2 $m_0, m_8$ –000 $\gamma$ IP3	$m_4, m_5, m_6, m_7$ 01 – – $\beta$ IP1
1	$m_2$ 0010 $\alpha\beta\gamma$ IP10 $m_4$ 0100 $\beta^*$ $m_8$ 1000 $\beta\gamma$ IP11	$m_2, m_6$ 0 – 10 $\beta$ IP4 $m_2, m_{10}$ –010 $\alpha\beta$ IP5 $m_4, m_5$ 010 – $\beta^*$ $m_4, m_6$ 01 – 0 $\beta^*$ $m_8, m_{10}$ 10 – 0 $\beta$ IP6	
2	$m_5$ 0101 $\beta\gamma^*$ $m_6$ 0110 $\beta^*$ $m_{10}$ 1010 $\alpha\beta^*$ $m_{12}$ 1100 $\alpha$ IP12	$m_5, m_7$ 01 – 1 $\beta\gamma$ IP7 $m_5, m_{13}$ –101 $\gamma$ IP8 $m_6, m_7$ 011 – $\beta^*$	
3	$m_7$ 0111 $\alpha\beta\gamma$ IP13 $m_{13}$ 1101 $\gamma^*$	$m_7, m_{15}$ –111 $\alpha$ IP9	
4	$m_{15}$ 1111 $\gamma^*$		

La Comparación de mintérminos adyacentes exige mínimo tener una función en común. Un término está cubierto sólo si todas las funciones a las que pertenece están cubiertas.

Ahora se procede a realizar la tabla 6.2 de implicantes primos contra mintérminos, pero organizando los mintérminos por función.

Se procede a completar la cubierta de todas las funciones con el menor número de implicantes primos por medio de un algoritmo llamado el algoritmo de Petrick.

## 6.4. ALGORITMO DE PETRICK

Como ya se ha señalado, los métodos mencionados anteriormente para seleccionar una cubierta mínima son heurísticos y, por tanto, no garantizan la obtención de una solución óptima. En particular, los últimos pasos de la minimización por mapas de Karnaugh y minimización por Quine-McCluskey se basan en el talento del diseñador para identificar un conjunto de implicantes mínimo para completar una cubierta, después de identificar los implicantes primos esenciales. Con frecuencia, se utiliza un método de prueba y error para identificar y evaluar las diversas cubiertas posibles.

Se han inventado varios métodos que pueden generar soluciones óptimas de manera directa, uno de estos métodos es el algoritmo de petrick, el cual utiliza un método algebraico para generar todas las cubiertas posibles de una función. El siguiente algoritmo para determinar cubiertas mínimas se basa en el algoritmo de petrick.

1. Utilizar la minimización por mapas de Karnaugh o minimización por Quine-McCluskey para determinar todos los implicantes primos de la función.
2. Crear una tabla de implicantes primos y después identificar y eliminar todos los implicantes primos esenciales, del método de Quine-McCluskey.
3. Para la tabla restante, escribir una expresión POS lógica que represente todas las cubiertas posibles, como sigue:
  - a) Para cada columna de mintérmino  $m_i$ , escribir una expresión para la suma (OR) de todos los implicantes primos que cubren a  $m_i$ . Esta expresión indica que se puede elegir cualquiera o varios de estos implicantes primos para cubrir a  $m_i$ .
  - b) Formar el producto (AND) de todos los términos suma del paso a). Este producto indica que todos los mintérminos deben quedar cubiertos.

Tabla 6.2.: Implicantes primos contra Minterminos

	IPE	$f_{\alpha}$				$f_{\beta}$				$f_{\gamma}$	
		$m_0^*$	$m_2^*$	$m_7$	$m_{10}^*$	$m_2^*$	$m_4^*$	$m_5^*$	$m_2^*$	$m_7$	$m_8$
IP1 $m_4, m_5, m_6, m_7, 01 - \beta$	*					X	X				
IP2 $m_0, m_2, 00-0 \alpha\gamma$	*	X	X						X		
IP3 $m_0, m_8, -000 \gamma$											X
IP4 $m_2, m_6, 0-10 \beta$						X					
IP5 $m_2, m_{10}, -010 \alpha\beta$	*		X		X	X					
IP6 $m_8, m_{10} \beta$											
IP7 $m_5, m_7 \beta\gamma$								X		X	
IP8 $m_5, m_{13} \gamma$											
IP9 $m_7, m_{15} \alpha$				X							
IP10 $m_2 \alpha\beta\gamma$			X			X			X		
IP11 $m_8 \beta\gamma$											X
IP12 $m_{12} \alpha$											
IP13 $m_7 \alpha\beta\gamma$				X						X	

4. Convertir la expresión lógica del paso 2 del formato POS al SOP mediante la ley distributiva, y después simplificar la expresión usando involución y absorción para eliminar los términos redundantes. Cada término producto de la expresión SOP representa una posible cubierta.
5. Seleccionar la cubierta con el menor costo; en este caso, el costo se calcula como el número de implicantes primos en el término producto y el número de literales en cada implicante primo.

Tomando el ejemplo anterior y por medio del algoritmo de Petrick se procede a sacar las funciones simplificadas (tabla 6.3).

$$(IP9 + IP13)(IP7 + IP13)(IP13 + IP11) = (IP9 \bullet IP7 + IP9 \bullet IP13 + IP7 \bullet IP13 + IP13)(IP13 + IP11) = IP3 \bullet IP9 \bullet IP7 + IP3 \bullet IP9 \bullet IP13 + IP3 \bullet IP7 \bullet IP13 + IP3 \bullet IP13 + IP11 \bullet IP9 \bullet IP7 + IP11 \bullet IP9 \bullet IP13 + IP11 \bullet IP7 \bullet IP13 + IP11 \bullet IP13$$

$IP3 \bullet IP13$  Elimina 1 variable (combinación óptima)

$IP11 \bullet IP13$  No elimina variables.

De esta manera se tienen las siguientes funciones simplificadas:

$$f\alpha = \bar{A} \bullet \bar{B} \bullet \bar{D} + \bar{B} \bullet C \bullet \bar{D} + \bar{A} \bullet B \bullet C \bullet D$$

$$f\beta = \bar{A} \bullet B + \bar{B} \bullet C \bullet \bar{D}$$

$$f\gamma = \bar{A} \bullet B \bullet C \bullet D + \bar{B} \bullet \bar{C} \bullet \bar{D} + \bar{A} \bullet \bar{B} \bullet \bar{D}$$

Tabla 6.3.: I.P contra mintérminos con algoritmo de Petrick

	IPE	$f_\alpha$					$f_\beta$				$f_\gamma$		
		m0*	m2*	m7	m10*	m2*	m4*	m5*	m2*	m7	m8		
IP1 $m_4, m_5, m_6, m_7$ 01- $\beta$	*						X	X					
IP2 $m_0, m_2$ 00-0 $\alpha\gamma$	*	X	X							X			
IP3 $m_0, m_8$ -000 $\gamma$		■										X	
IP4 $m_2, m_6$ 0-10 $\beta$						X							
IP5 $m_2, m_{10}$ -010 $\alpha\beta$	*		X		X	X							
IP6 $m_8, m_{10}$ $\beta$													
IP7 $m_5, m_7$ $\beta\gamma$								X			X		
IP8 $m_5, m_{13}$ $\gamma$													
IP9 $m_7, m_{15}$ $\alpha$				X									
IP10 $m_2$ $\alpha\beta\gamma$			X			X				X			
IP11 $m_8$ $\beta\gamma$												X	
IP12 $m_{12}$ $\alpha$													
IP13 $m_7$ $\alpha\beta\gamma$		■		X						X			





## 7. FUNCIONES LÓGICAS

Se denomina lógica combinacional a todo sistema digital en el que el estado de sus salidas se encuentra definido exclusivamente por el valor de sus entradas en un momento determinado, sin que intervengan en ningún caso estados anteriores de las entradas o de las salidas. Las funciones (OR, AND, NOT, NAND, NOR, XOR, XNOR) son booleanas donde cada función se puede representar en una tabla de la verdad, por lo tanto, carecen de memoria y de realimentación.

Dentro de la electrónica digital, existe un gran número de problemas a resolver que se repiten normalmente. Por ejemplo, es muy común que al diseñar un circuito electrónico se necesite tener el valor opuesto al de un punto determinado, o que cuando un cierto número de pulsadores estén activados, una salida permanezca apagada. Todas estas situaciones pueden ser expresadas mediante ceros y unos, y tratadas mediante circuitos digitales. Los elementos básicos de cualquier circuito digital son las Compuertas Lógicas[8, 9, 10]

A continuación se muestran las funciones básicas de la lógica combinacional:[8, 9, 10]

### 7.1. COMPUERTA LÓGICA OR

La función booleana que realiza la compuerta OR es la asociada a la suma, y su notación es “+”. Esta compuerta presenta un estado alto en su salida cuando al menos una de sus entradas también está en estado alto. En cualquier otro caso, la salida será 0.

**Símbolo:**

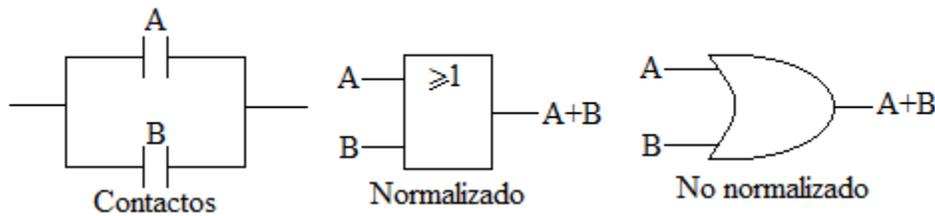


Figura 7.1.: Compuerta lógica OR

**Ecuación:**

$$S = A + B$$

**Tabla de verdad:**

Tabla 7.1.: Compuerta lógica OR

Entrada A	Entrada B	Salida S
0	0	0
0	1	1
1	0	1
1	1	1

## 7.2. COMPUERTA LÓGICA AND

Al observar la tabla de verdad de la compuerta lógica AND (tabla 7.2), se advierte que la operación AND es exactamente igual que la multiplicación ordinaria. Siempre que  $A$  o  $B$  sean cero, su producto será cero; cuando  $A$  y  $B$  sean 1, su producto será 1. Por lo tanto se puede decir que en la operación AND el resultado será 1 solo si todas las entradas son 1; en los demás casos el resultado será cero.

**Símbolo:**

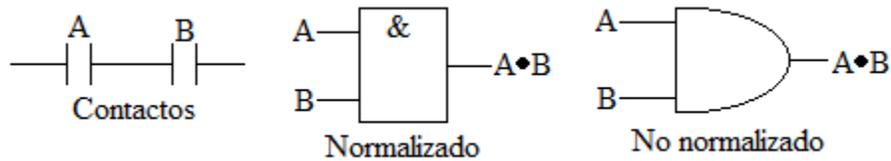


Figura 7.2.: Compuerta lógica AND

**Ecuación:**

$$S = A \bullet B$$

**Tabla de verdad:**

Tabla 7.2.: Compuerta lógica AND

Entrada A	Entrada B	Salida S
0	0	0
0	1	0
1	0	0
1	1	1

## 7.3. COMPUERTA LÓGICA NOT

La compuerta NOT entrega en su salida el inverso (opuesto) de la entrada. Esto significa que: si en la entrada se tiene un "1" lógico, en la salida habrá un "0" lógico, y si a la entrada se tiene un "0" lógico, en la salida habrá un "1" lógico.

**Símbolo:**

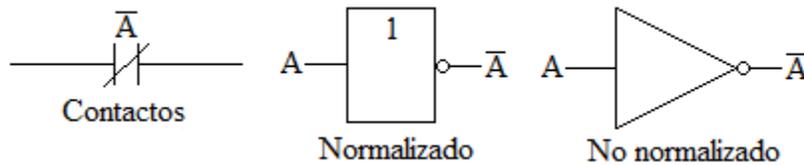


Figura 7.3.: Compuerta lógica NOT

**Ecuación:**

$$S = \bar{A}$$

**Tabla de verdad:**

Tabla 7.3.: Compuerta lógica NOT

Entrada A	Salida S
0	1
1	0

## 7.4. COMPUERTA LÓGICA NAND

La función NO-Y, llamada más comúnmente NAND es la negación de la función Y (AND) precedente. Así como en una puerta Y (AND) se necesita que exista nivel 1 en todas las entradas para obtener el mismo nivel en la salida, en una NAND el nivel de la salida es 0 en las mismas condiciones. Por el contrario, cuando hay un nivel 0 en alguna de las entradas de una puerta Y (AND) la salida está a nivel 0, mientras que en iguales circunstancias en una puerta NAND el nivel de salida es 1.

**Símbolo:**

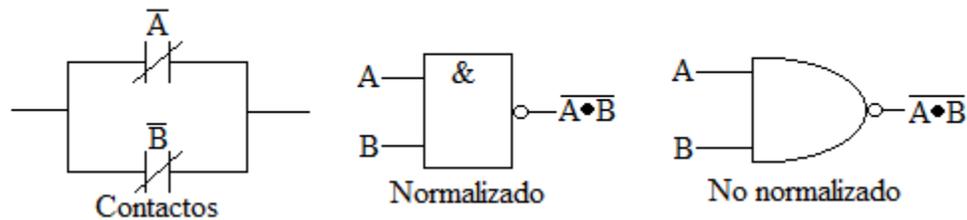


Figura 7.4.: Compuerta lógica NAND

**Ecuación:**

$$S = \overline{A \bullet B} = \overline{A} + \overline{B}$$

**Tabla de verdad:**

Tabla 7.4.: Compuerta lógica NAND

Entrada A	Entrada B	Salida S
0	0	1
0	1	1
1	0	1
1	1	0

## 7.5. COMPUERTA LÓGICA NOR

La función NOR es la negación de la función OR, es decir, que en la función NOR se obtiene un "1" lógico si las entradas se encuentran en un estado lógico bajo, de lo contrario su salida será un "0" lógico.

**Símbolo:**

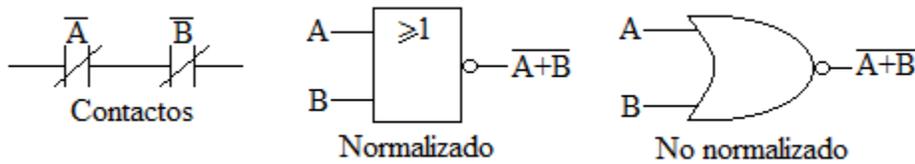


Figura 7.5.: Compuerta lógica NOR

**Ecuación:**

$$S = \overline{A+B} = \bar{A} \bullet \bar{B}$$

**Tabla de verdad:**

Tabla 7.5.: Compuerta lógica NOR

Entrada A	Entrada B	Salida S
0	0	1
0	1	0
1	0	0
1	1	0

## 7.6. COMPUERTA LÓGICA XOR

A diferencia de la compuerta OR, la compuerta XOR tiene una salida igual a "0" cuando sus entradas son iguales a 1 o si ninguna entrada es 1.

Si se comparan las tablas de verdad de la compuerta OR y la compuerta XOR se observa que la compuerta XOR tendrá un "1" en su salida cuando la suma de los "1" en las entradas sea igual a un número impar.

**Símbolo:**

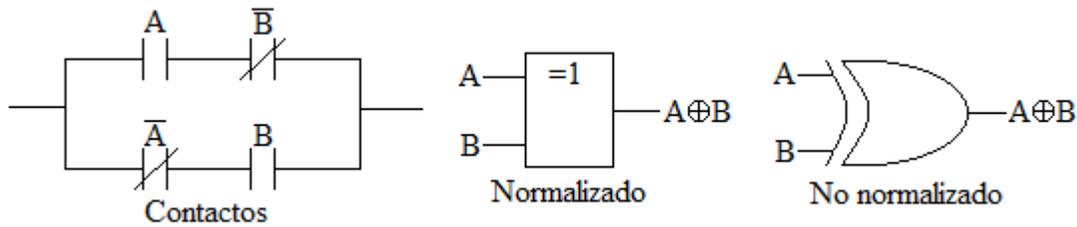


Figura 7.6.: Compuerta lógica XOR

**Ecuación:**

$$S = A \oplus B$$

$$S = \bar{A} \cdot B + A \cdot \bar{B}$$

**Tabla de verdad:**

Tabla 7.6.: Compuerta lógica XOR

Entrada A	Entrada B	Salida S
0	0	0
0	1	1
1	0	1
1	1	0

## 7.7. COMPUERTA LÓGICA XNOR

Como se puede deducir de los casos anteriores, una compuerta XNOR es una compuerta XOR con su salida negada, por lo que su salida estará en estado alto solamente cuando sus entradas son iguales, y en estado bajo para las demás combinaciones posibles.

**Símbolo:**

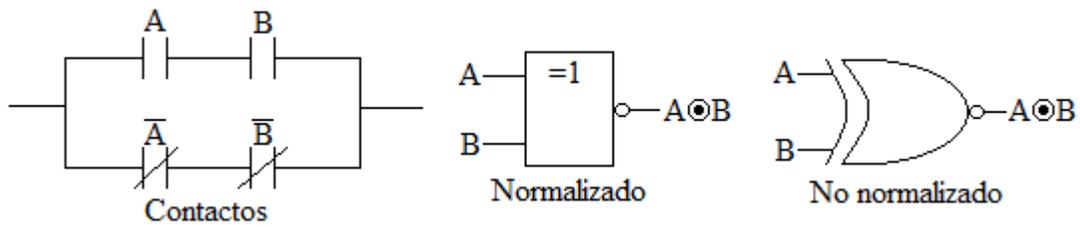


Figura 7.7.: Compuerta lógica XNOR

**Ecuación:**

$$S = \overline{A \oplus B}$$

**Tabla de verdad:**

Tabla 7.7.: Compuerta lógica XNOR

Entrada A	Entrada B	Salida S
0	0	1
0	1	0
1	0	0
1	1	1





## 8. LÓGICA SECUENCIAL

A diferencia de los sistemas combinacionales, en los sistemas secuenciales, los valores de las salidas, en un momento dado, no dependen exclusivamente de los valores de las entradas en dicho momento, sino también dependen del estado interno. El sistema secuencial más simple es el biestable, de los cuales, el de tipo  $D$  (o cerrojo) es el más utilizado actualmente.[11]

La mayoría de los sistemas secuenciales están gobernados por señales de reloj. A éstos se los denomina "síncronos" o "sincrónicos", a diferencia de los "asíncronos" o "asincrónicos" que son aquellos que no son controlados por señales de reloj.

En todo sistema secuencial se encuentra:[12, 13]

- Un conjunto finito,  $n$ , de variables de entrada ( $X_1, X_2, \dots, X_n$ ).
- Un conjunto finito,  $m$ , de estados internos, de aquí que los estados secuenciales también sean denominados autómatas finitos. Estos estados proporcionan  $m$  variables internas ( $Y_1, Y_2, \dots, Y_m$ ).
- Un conjunto finito,  $p$ , de funciones de salida ( $Z_1, Z_2, \dots, Z_p$ ).

Los biestables y los flip flops son los bloques constitutivos de la mayoría de los circuitos secuenciales. Los sistemas digitales típicos usan biestables y flip flops que son dispositivos preempaquetados y especificados de manera funcional en un circuito integrado estándar. En el ambiente ASIC (Application Specific Integrated Circuit), los biestables y los flip flops son celdas típicamente predefinidas especificadas por el vendedor de ASIC. Sin embargo, dentro de un IC (Integrated Circuit) estándar o un ASIC, cada biestable o flip flop está diseñado como un circuito secuencial retroalimentado mediante compuertas lógicas individuales y lazos de retroalimentación.[13]

A continuación se muestran los elementos básicos de la lógica secuencial:[11, 12, 13]

### 8.1. BIESTABLE S-R

Un biestable  $S$ - $R$  (set-reset) basado en compuertas NOR tiene dos entradas,  $S$  y  $R$  y dos salidas,  $Q$  y  $\bar{Q}$ , donde  $\bar{Q}$  es el complemento de  $Q$ .

Si  $S$  y  $R$  son ambos cero, el circuito se comporta como un elemento biestable; se tiene un lazo de retroalimentación que retiene uno de los estados lógicos,  $Q = 0$  o  $Q = 1$ ;  $S$  o  $R$  pueden ser excitados para forzar al lazo de retroalimentación al estado deseado.  $S$  inicia (set) o preinicia la salida  $Q$  a 1;  $R$  despeja (reset) o aclara la salida  $Q$  a 0. Luego que se nieguen las entradas  $S$  y  $R$ , el biestable permanece en el estado al que fue forzado. La figura 8.1 describe el comportamiento funcional de un biestable S-R para una secuencia típica de entradas.

En la tabla 8.1 se observa que el estado siguiente para las entradas de  $R = S = 1$  no es permitido, esto es, por que las salidas  $Q$  y su complemento  $\bar{Q}$  toman el valor de 0, lo cual crea entre las compuertas una condición de competencia y, dependiendo de la estructura física de éstas, no se puede asegurar el valor de la salida. Por tal razón la combinación  $R = S = 1$  no está permitida para este sistema.

Un posible diseño para el biestable S-R con compuertas NOR podría ser el mostrado en la figura 8.1.

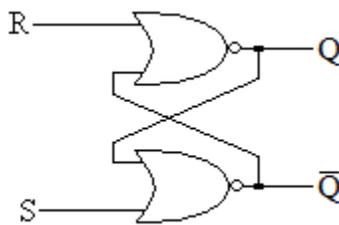


Figura 8.1.: Biestable S-R con compuertas NOR

La ecuación característica que representa el comportamiento del biestable S-R es la siguiente:

$$Q(t+1) = S + \bar{R} \bullet Q(t)$$

Los resultados obtenidos del estado futuro al evaluar las entradas  $S$ ,  $R$  y  $Q(t)$  en la ecuación anterior, son mostrados en la tabla 8.1, y la relación que existe entre el estado presente y el estado futuro se observa en la tabla 8.2.

Tabla 8.1.: Tabla de transición para el Biestable S-R

Entrada S	Entrada R	Estado Presente Q(t)	Estado Futuro Q(t+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	x
1	1	1	x

Tabla 8.2.: Tabla de excitación para el Biestable S-R

Estado Actual	Entradas S-R			
	00	01	10	11
0	0	0	1	x
1	1	0	1	x

El diagrama de estados mostrado en la figura 8.2 hace referencia al comportamiento de la tabla 8.2.

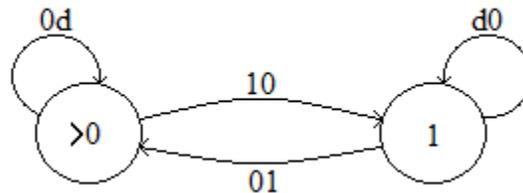


Figura 8.2.: Diagrama de estados Biestable S-R

El símbolo que representa el biestable S-R se muestra en la figura 8.3.

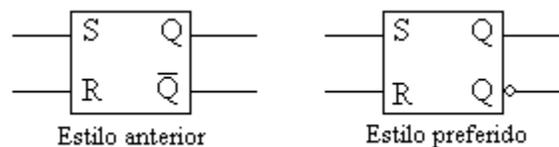


Figura 8.3.: Biestable S-R

## 8.2. BIESTABLE S-R CON HABILITACIÓN

Un biestable  $S$ - $R$  o  $\bar{S}$ - $\bar{R}$  es sensible a sus entradas  $S$  y  $R$  en todo momento. Sin embargo, puede modificarse fácilmente para crear un dispositivo que sea sensible a estas entradas sólo cuando este activa una entrada de habilitación  $C$ . Tal biestable  $S$ - $R$  con habilitación se muestra en la figura 8.4, además como se observa en la tabla 8.3, el circuito se comporta como un biestable  $S$ - $R$  cuando  $C$  está en 1 y retiene su estado previo cuando  $C$  está en 0. Si  $S$  y  $R$  son 1 cuando  $C$  cambia de 1 a 0, el circuito se comporta como un biestable  $S$ - $R$  en el que  $S$  y  $R$  se niegan simultáneamente; el siguiente estado es impredecible y la salida puede llegar a ser metaestable<sup>1</sup>.

Un posible diseño para el biestable S-R con habilitación utilizando compuertas NAND se muestra en la figura 8.4.

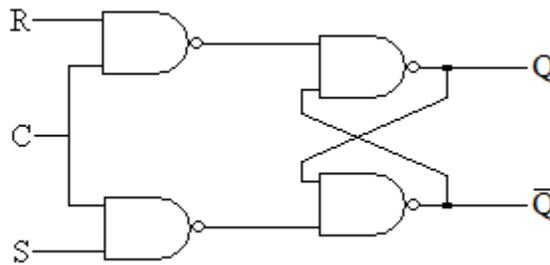


Figura 8.4.: Biestable S-R con habilitación con compuertas NAND

La ecuación característica que representa el comportamiento del biestable S-R con habilitación es la siguiente:

$$Q(t+1) = (S \bullet C) + (\bar{R} \bullet Q(t)) + (\bar{C} \bullet Q(t))$$

Los resultados obtenidos del estado futuro al evaluar las entradas  $S$ ,  $R$ ,  $C$  y  $Q(t)$  en la ecuación anterior, son mostrados en la tabla 8.3, y la relación que existe entre el estado presente y el estado futuro se observa en la tabla 8.4.

<sup>1</sup>Es la indefinición del valor de salida de un biestable durante un cierto periodo de tiempo.

Tabla 8.3.: Tabla de transición para el Biestable S-R con habilitación

Entrada C	Entrada S	Entrada R	Estado Presente Q(t)	Estado Futuro Q(t+1)
0	d	d	0	0
0	d	d	1	1
1	0	0	0	0
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	x
1	1	1	1	x

Tabla 8.4.: Tabla de excitación para el Biestable S-R con habilitación

Estado Actual	Entradas S-C-R							
	000	001	010	011	100	101	110	111
0	0	0	0	0	0	0	1	x
1	1	1	1	1	1	0	1	x

El diagrama de estados mostrado en la figura 8.5 hace referencia al comportamiento de la tabla 8.4.

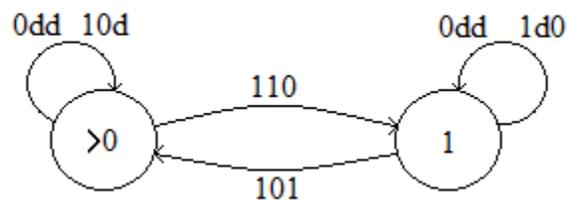


Figura 8.5.: Diagrama de estados Biestable S-R con habilitación

El símbolo que representa el biestable S-R con habilitación se muestra en la figura 8.4.

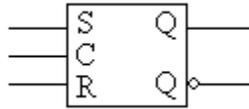


Figura 8.6.: Biestable S-R con habilitación

### 8.3. BIESTABLE D

Los biestables  $S$ - $R$  son útiles en especificaciones de control, donde se piensa, a menudo, en términos de poner en 1 al biestable en respuesta a alguna condición y ponerlo en 0 cuando esta cambia; así, se controlan las entradas de inicio y despeje de manera algo independiente. Sin embargo, a menudo se necesitan biestables simplemente para almacenar bits de información: cada bit de información se presenta en una línea de información y se necesita almacenarlo en alguna parte. Un biestable  $D$  puede usarse para tal aplicación.

El diagrama lógico del biestable  $D$  es reconocible como un biestable  $S$ - $R$  con habilitación, con el agregado de un inversor para generar las entradas  $S$  y  $R$  de la única entrada  $D$  (dato). Esto elimina la situación molesta de los biestables  $S$ - $R$ , donde  $S$  y  $R$  pueden estar en 1 simultáneamente. La entrada de control de un biestable  $D$ , etiquetada  $C$ , se conoce como ENABLE, CLK o G y está activa en baja en algunos diseños de biestables  $D$ .

Un posible diseño para el biestable  $D$  utilizando compuertas NAND se muestra en la figura 8.7.

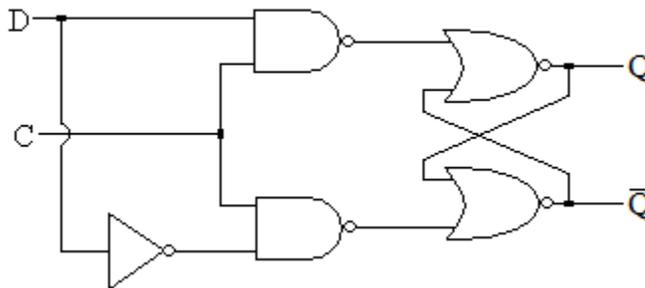


Figura 8.7.: Biestable D con compuertas NAND

La ecuación característica que representa el comportamiento del biestable D es la siguiente:

$$Q(t+1) = (D \bullet C) + (\bar{C} \bullet Q(t))$$

Los resultados obtenidos del estado futuro al evaluar las entradas  $D$ ,  $C$ , y  $Q(t)$  en la ecuación anterior, son mostrados en la tabla 8.5, y la relación que existe entre el estado presente y el estado futuro se observa en la tabla 8.6.

Tabla 8.5.: Tabla de transición del Biestable D

Entrada C	Entrada D	Estado Presente Q(t)	Estado Futuro Q(t+1)
0	d	0	0
0	d	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

Tabla 8.6.: Tabla de excitación para el Biestable D

Estado Actual	Entradas C-D			
	00	01	10	11
0	0	0	0	1
1	1	1	0	1

El diagrama de estados mostrado en la figura 8.8 hace referencia al comportamiento de la tabla 8.6.

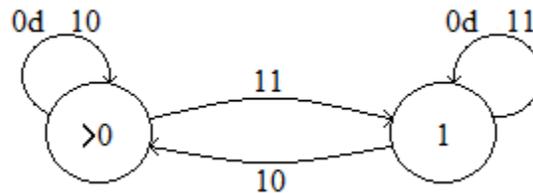


Figura 8.8.: Diagrama de estados Biestable D

El símbolo que representa el biestable D se muestra en la figura 8.9.

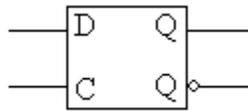


Figura 8.9.: Biestable D

En los biestables se encuentra un problema con las entradas, ya que, ante cualquier cambio en ellas se produce de manera inmediata el paso al estado siguiente, por lo que se hace necesario otro tipo de dispositivo, como lo son los flip flops que se muestra en la siguiente sección.

## 8.4. FLIP FLOPS

Los flip flops son utilizados en circuitos que requieren de una realimentación o de una memoria, y la utilidad de estos radica en que realizan el cambio a un estado siguiente en sincronismo con los pulsos de una señal de reloj, lo que no ocurre con los biestables. Es por esto que nace la necesidad de tener flip flops, ya que se pueden presentar cambios en el sistema de acuerdo con las entradas, pero al mismo tiempo con una señal de reloj.

### 8.4.1. Flip flop S-R maestro/esclavo

Se indicó antes que los biestables *S-R* son útiles en aplicaciones de “control”, donde se pueden tener condiciones independientes para iniciar (poner en 1) y despejar (poner en 0) un bit de control. Si se supone que el bit de control sólo cambia en ciertos momentos con respecto a una señal de reloj, entonces se necesita un flip-flop *S-R* que, a semejanza de un biestable *D*, sólo cambia sus salidas con cierto borde de la señal de reloj.

Al igual que un biestable *D*, el flip flop *S-R* sólo cambia sus salidas en el borde descendente de la señal de control *C*. Sin embargo, el nuevo valor de salida depende de los valores de entrada, no sólo durante el flanco descendente, sino durante todo el intervalo en el que *C* está ALTO previo al flanco descendente.

Un posible diseño para el flip flop *S-R* maestro/esclavo utilizando biestables *S-R* se muestra en la figura 8.10.

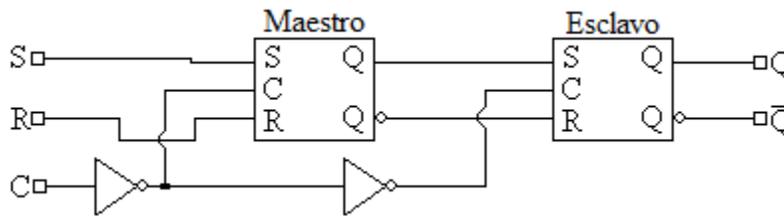


Figura 8.10.: Flip flop S-R maestro/esclavo usando Biestables S-R

La ecuación característica que representa el comportamiento del flip flop S-R maestro/esclavo es la siguiente:

$$Q(t + 1) = S + \bar{R} \bullet Q(t)$$

Los resultados obtenidos del estado futuro al evaluar las entradas  $S$ ,  $R$ , y  $Q(t)$  en la ecuación anterior, son mostrados en la tabla 8.7, y la relación que existe entre el estado presente y el estado futuro se observa en la tabla 8.8.

Tabla 8.7.: Tabla de transición del flip flop S-R maestro/esclavo

Entrada S	Entrada R	Estado Presente Q(t)	Señal de Activación C	Estado Futuro Q(t+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	↓	x
1	1	1	↓	x

Tabla 8.8.: Tabla de excitación del flip flop S-R maestro/esclavo

Estado Actual	Entradas S-R			
	00	01	10	11
0	0	0	1	x
1	1	0	1	x

El diagrama de estados mostrado en la figura 8.11 hace referencia al comportamiento de la tabla 8.8.

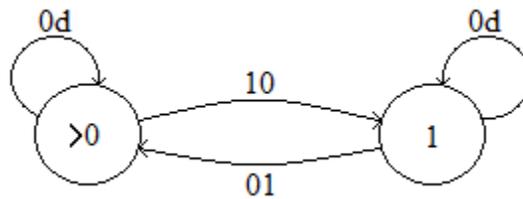


Figura 8.11.: Diagrama de estados flip flop S-R maestro/esclavo

El símbolo que representa el flip flop S-R maestro/esclavo se muestra en la figura 8.12.

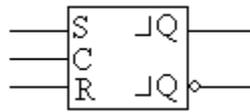


Figura 8.12.: Símbolo flip flop S-R maestro/esclavo

#### 8.4.2. Flip flop J-K maestro/esclavo

El problema de qué hacer cuando  $S$  y  $R$  están activas simultáneamente se resuelve con un flip flop J-K maestro/esclavo. Las entradas  $J$  y  $K$  son análogas a  $S$  y  $R$ ; sin embargo, el hecho de que  $J$  esté activo, activa la entrada  $S$  del biestable maestro sólo si la salida  $\bar{Q}$  del flip flop es 1 en ese momento (o sea,  $Q$  es 0), y al activar  $K$ , se activa la entrada  $R$  del biestable maestro sólo si  $Q$  es 1 en ese momento. Por lo tanto, si  $J$  y  $K$  están activas simultáneamente el flip flop pasa al estado opuesto de su estado presente.

Un posible diseño para el flip flop J-K maestro/esclavo empleando biestables S-R se muestra en la figura 8.13.

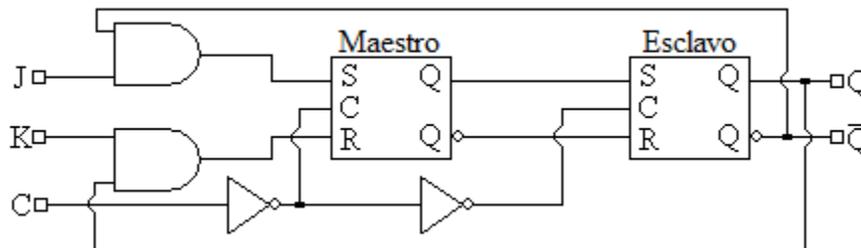


Figura 8.13.: Flip flop J-K maestro/esclavo empleando Biestables S-R

La ecuación característica que representa el comportamiento del flip flop S-R maestro/esclavo es la siguiente:

$$Q(t+1) = \bar{K} \cdot Q(t) + J \cdot \overline{Q(t)}$$

Los resultados obtenidos del estado futuro al evaluar las entradas  $J$ ,  $K$ , y  $Q(t)$  en la ecuación anterior, son mostrados en la tabla 8.9, y la relación que existe entre el estado presente y el estado futuro se observa en la tabla 8.10.

Tabla 8.9.: Tabla de transición para el flip flop J-K maestro/esclavo

Entrada J	Entrada K	Estado Presente Q(t)	Señal de Activación C	Estado Futuro Q(t+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	↓	0

Tabla 8.10.: Tabla de excitación del flip flop J-K maestro/esclavo

Estado Actual	Entradas J-K			
	00	01	10	11
0	0	0	1	1
1	1	0	1	0

El diagrama de estados mostrado en la figura 8.14 hace referencia al comportamiento de la tabla 8.10.

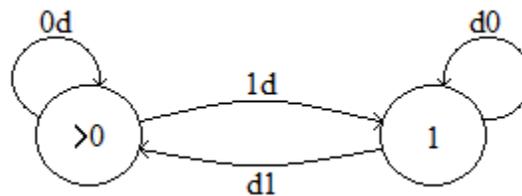


Figura 8.14.: Diagrama de estados flip flop J-K maestro/esclavo

El símbolo que representa el flip flop J-K maestro/esclavo se muestra en la figura 8.15.

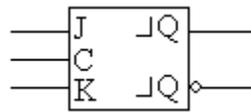


Figura 8.15.: Símbolo flip flop J-K maestro/esclavo

### 8.4.3. Flip flop D maestro/esclavo

Un flip flop D disparado por flanco positivo combina un par de biestables D, para crear un circuito que muestre sus entradas  $D$  y cambie sus salidas  $Q$  y  $\bar{Q}$  sólo en el borde de ascenso de una señal  $C$  de control. Al primer biestable se le conoce como el maestro; está abierto y sigue la entrada cuando  $C$  está bajo. Cuando  $C$  pasa a alto, el biestable maestro cierra y su salida se transfiere al segundo biestable, llamado el biestable esclavo. El biestable está abierto todo el tiempo que  $C$  está alto, aunque cambia al inicio de este intervalo, debido a que el biestable maestro se cierra y no cambia durante el resto del intervalo.

Un posible diseño para el flip flop D maestro/esclavo haciendo uso de biestables D se muestra en la figura 8.16.

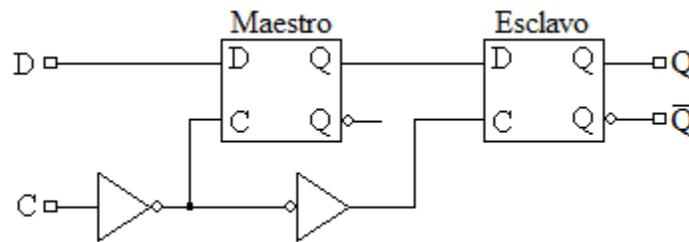


Figura 8.16.: Flip flop D maestro/esclavo utilizando Biestables D

La ecuación característica que representa el comportamiento del flip flop D maestro/esclavo es la siguiente:

$$Q(t+1) = D$$

Los resultados obtenidos del estado futuro al evaluar la entrada  $D$  en la ecuación anterior, son mostrados en la tabla 8.11, y la relación que existe entre el estado presente y el estado futuro se observa en la tabla 8.12.

Tabla 8.11.: Tabla de transición para el flip flop D maestro/esclavo

Entrada	Estado Presente	Señal de Activación	Estado Futuro
D	Q(t)	C	Q(t+1)
0	0	↓	0
0	1	↓	0
1	0	↓	1
1	1	↓	1

Tabla 8.12.: Tabla de excitación del flip flop D

Estado Actual	Entrada D	
	0	1
0	0	1
1	0	1

El diagrama de estados mostrado en la figura 8.17 hace referencia al comportamiento de la tabla 8.12.

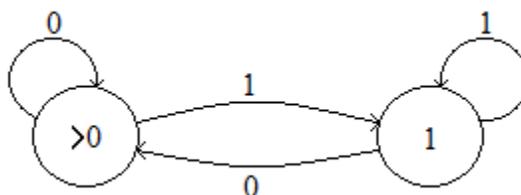


Figura 8.17.: Diagrama de estados flip flop D maestro/esclavo

El símbolo que representa el flip flop D maestro/esclavo se muestra en la figura 8.18.

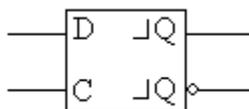


Figura 8.18.: Símbolo flip flop D maestro/esclavo

### 8.4.4. Flip flop T

Un Flip flop T, es el mismo Flip flop J-K con sus entradas equivalente  $J = K$ . Este Flip flop T tiene el comportamiento de una función toggle, cuando el valor  $T$  toma el estado lógico 1 y retendrá el estado actual si el valor de la entrada es 0.

Un posible diseño para el flip flop T haciendo uso de un flip flop J-K se muestra en la figura 8.16.

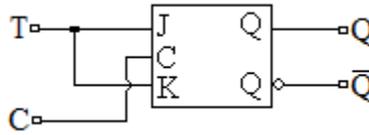


Figura 8.19.: Flip flop T empleando Flip flop J-K

La ecuación característica que representa el comportamiento del flip flop T es la siguiente.

$$Q(t+1) = \bar{T} \bullet Q(t) + T \bullet \overline{Q(t)}$$

Los resultados obtenidos del estado futuro al evaluar la entrada  $T$ ,  $Q(t)$  en la ecuación anterior, son mostrados en la tabla 8.13, y la relación que existe entre el estado presente y el estado futuro se observa en la tabla 8.14.

Tabla 8.13.: Tabla de transición para el flip flop T

Entrada	Estado Presente	Señal de Activación	Estado futuro
T	Q(t)	C	Q(t+1)
0	0	┘	0
0	1	┘	1
1	0	┘	1
1	1	┘	0

Tabla 8.14.: Tabla de excitación del flip flop T

Estado Actual	Entrada T	
	0	1
0	0	1
1	1	0

El diagrama de estados mostrado en la figura 8.20 hace referencia al comportamiento de la tabla 8.14.

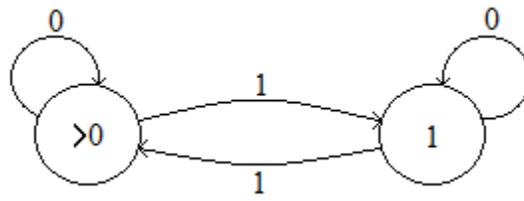


Figura 8.20.: Diagrama de estados flip flop T

El símbolo que representa el flip flop T se muestra en la figura 8.21.

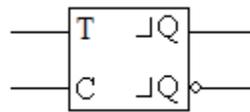


Figura 8.21.: Símbolo flip flop T



## 9. MÁQUINA DE ESTADOS FINITOS

Se denomina máquina de estado a un modelo de comportamiento de un sistema con entradas y salidas, en donde las salidas dependen no sólo de las señales de entrada actuales sino también de las anteriores. Las máquinas de estados se definen como un conjunto de estados que sirve de intermediario en esta relación de entradas y salidas, haciendo que el historial de señales de entrada defina un estado, que en pocas palabras son el único medio de memoria en las máquinas de estados finitos, haciendo que las salidas dependan únicamente del estado y las entradas actuales.

Una máquina de estados se denomina máquina de estados finitos, si el conjunto de estados de la máquina es finito, este es el único tipo de máquinas de estados que se puede modelar en la actualidad a través de una computadora. Un ejemplo de una máquina de estados infinitos es una Máquina universal de Turing la cual se puede definir teóricamente con una "cinta" o memoria infinita. [9, 11]

Es posible clasificar las máquinas de estados en aceptoras o transductoras:

- **Aceptoras** (también llamadas **reconocedoras** o **discriminadoras**): Son aquellas en donde la salida es binaria (0 ó 1), depende únicamente del estado en que se encuentran, además sólo posee un estado inicial. Se puede decir, entonces, que cuando la máquina produce una salida positiva (es decir, un "1"), es porque ha reconocido o aceptado la secuencia de entrada. En las máquinas de estados aceptoras, los estados con salida positiva se denominan estados finales.
- **Transductoras**: Las máquinas de estado transductoras son las más generales, convierten una secuencia de señales de entrada en una secuencia de salida, no importa si esta es binaria o de carácter más compleja, sólo depende de la entrada actual sin importar el estado inicial en que se encuentre.

La representación de una máquina de estados se hace mediante diagramas de estados.

### 9.1. DIAGRAMA DE ESTADOS

Un diagrama de estados es una representación gráfica que indica la secuencia de los estados que se presentan en un circuito secuencial, teniendo en cuenta las entradas y salidas. El diagrama se forma

con círculos y líneas. Los círculos representan los estados del circuito secuencial y cada uno de ellos contiene un número o una letra que identifica su estado, adicional a esto el estado inicial de la máquina va marcado por el símbolo (>). Las líneas que salen de los estados se denominan transiciones. El símbolo (/) se utiliza en las transiciones cuando es una máquina de Mealy, separando dos números que corresponden a la entrada y salida respectivamente, mientras que en la máquina de Moore se utiliza que en la máquina de Moore se utiliza en los estados, realizando un proceso similar.

En el diagrama de estado que se observa en la figura 9.1, la línea que une los estados 00 y 01 marcada como 1/0 indica que el circuito secuencial se encuentra en el estado 00 mientras la entrada sea 0 y la salida también sea 0, y que después de que ocurra una transición en la señal de reloj el estado cambia a 01.

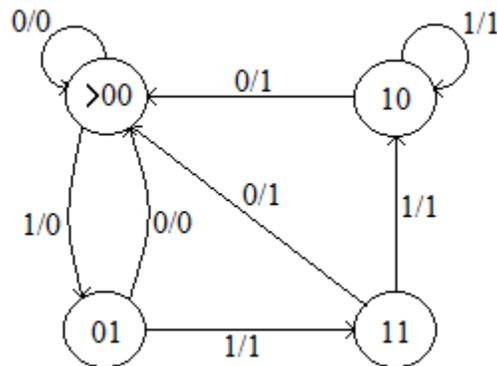


Figura 9.1.: Diagrama de estados

Otra forma de representación de un sistema secuencial es mediante la tabla de estados, en ella las columnas corresponden a los valores de las variables de entrada y las filas a los estados presentes del sistema. En cada celda se relaciona para el valor de entrada y del estado actual el estado siguiente y el valor de las salidas.

Para el diagrama mostrado en la figura 9.1 la tabla de estados correspondiente es la que se muestra en la tabla 9.1:

Tabla 9.1.: Tabla de estados

	Entrada	
Estado Actual	0	1
00	00/0	01/0
01	00/0	11/1
10	00/1	10/1
11	00/1	11/1

## 9.2. MÁQUINAS DE MEALY Y DE MOORE

Los circuitos secuenciales se clasifican dentro de una categoría conocida como máquinas de estados finitos con salida, de la cual se distinguen comúnmente dos tipos:

- **Máquina de Mealy:** En esta máquina de estados las salidas se encuentran determinadas por el estado interno del sistema y por las entradas no sincronizadas con el circuito. El diagrama de bloques representativo de esta máquina se muestra en la figura 9.2, donde se observa que las salidas del sistema son tanto sincrónicas como asincrónicas.
- **Máquina de Moore:** Las salidas solo dependen del estado interno y de cualquier entrada sincronizada con el circuito, como se observa en la figura 9.2, donde las salidas del sistema son únicamente sincrónicas. Un ejemplo de este tipo de máquinas de estado son los contadores.

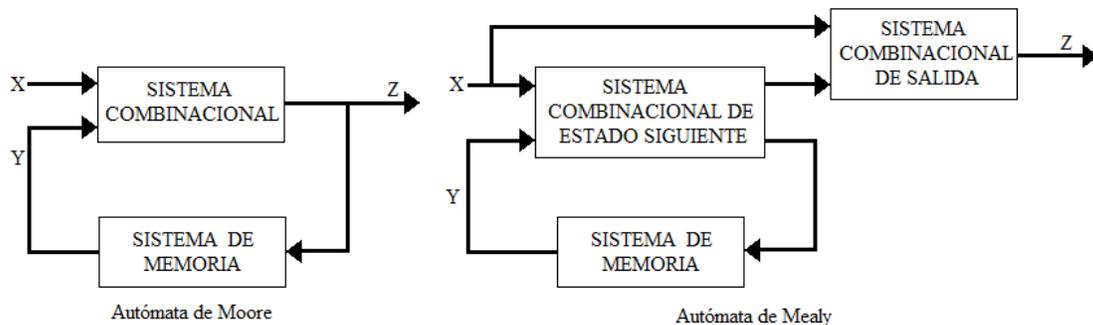


Figura 9.2.: Autómata de Moore y autómata de Mealy

## 9.3. SIMPLIFICACIÓN

Existen tres técnicas para determinar estados equivalentes en los circuitos secuenciales las cuales son explicadas a continuación.

### 9.3.1. Inspección

La técnica más sencilla es la de reconocer los estados equivalentes por inspección. En este método basta con reconocer varias filas en la tabla de estados que realicen la misma función, para después eliminar los estados redundantes.

**Ejemplo**

En este ejemplo, hay dos situaciones de filas equivalentes. En primer lugar, se examina el circuito secuencial definido mediante la tabla de estados de la figura 9.3. Ésta se puede reducir por inspección, ya que los estados *B* y *D* realizan exactamente la misma función. Por tanto, se puede eliminar el estado *D* de la tabla, para lo cual basta con quitar la fila *D* de la tabla y reemplazarlo por su estado equivalente *B*. Este procedimiento produce la tabla de estados de la figura 9.4.

	0	1
A	B/0	C/1
B	C/0	A/1
C	D/1	B/0
D	C/0	A/1

Figura 9.3.: Circuito 1

	0	1
A	B/0	C/1
B	C/0	A/1
C	B/1	B/0

Figura 9.4.: Circuito 1 reducido

	0	1
A	B/0	C/1
B	B/0	A/1
C	D/1	B/0
D	D/0	A/1

Figura 9.5.: Circuito 2

	0	1
A	B/0	C/1
B	B/0	A/1
C	B/1	B/0

Figura 9.6.: Circuito 2 reducido

Analizando el circuito secuencial de la figura 9.5. Si el circuito está en el estado  $B$  y recibe un 0 lógico, permanecerá en el mismo estado con un 0 como salida, y si recibe una entrada 1 lógico pasará al estado  $A$  con una salida 1. Por otro lado, si el circuito está en el estado  $D$ , se puede afirmar lo mismo: con una entrada 0 lógico permanece en el mismo estado con una salida 0, y con una entrada 1 lógico pasará al estado  $A$  con una salida 1. Por tanto, los estados  $B$  y  $D$  son equivalentes por inspección. La figura 9.6 muestra la tabla de estados reducida.

### 9.3.2. Partición

El método de partición implica la determinación sucesiva de particiones  $P_k, K = 1, 2, 3, \dots, I$ , donde  $P_k$  está compuesto por varios bloques, cada bloque consta de un grupo de uno o más estados. Los estados contenidos dentro de un bloque dado de  $P_k$  son  $K$ -equivalentes, es decir, dado un circuito secuencial con estados  $S_1, S_2, S_3, S_4, S_5$ , si  $P_k = (S_1 \bullet S_3) \bullet (S_2 \bullet S_4) \bullet (S_5)$ , entonces  $P_k$  contiene tres bloques y  $S_1$  y  $S_3$  son  $K$  equivalentes, al igual que  $S_2$  y  $S_4$ ,  $S_5$  no es  $K$ -equivalente a algún otro estado del circuito secuencial. Por claridad, se toma como ejemplo el circuito secuencial mostrado a continuación para describir el procedimiento de partición.

	0	1
A	C/1	B/0
B	C/1	E/0
C	B/1	E/0
D	D/0	B/1
E	E/0	A/1

Figura 9.7.: Circuito secuencial método de partición

**Procedimiento de partición**

- Paso 1: Se forma la primera partición  $P_1$  colocando dos o más estados en el mismo bloque de  $P_1$  si y sólo si, su salida es idéntica para cada entrada, para el ejemplo de la figura 9.7  $P_1 = (A \bullet B \bullet C) \bullet (D \bullet E)$  y, por tanto, los estados dentro de cada bloque son 1-equivalentes. El paso 1 garantiza, que cada bloque de  $P_1$  satisface la condición 1 para los estados equivalentes.
  
- Paso 2: Se obtiene las particiones  $P_k, K = 2, 3, 4, \dots, I$  de manera sucesiva, colocando dos o más estados en el mismo bloque de  $P_k$  si y sólo si para cada valor de entrada sus estados siguientes están en un único bloque de  $P_k$ . Este proceso de carácter iterativo es sugerido por la condición 2 para los estados equivalentes.
  
- Paso 3: Cuando  $P_{k+1} = P_k$ , es decir, una vez que la partición se repite, los estados de cada bloque de  $P_k$  que son  $K$ -equivalentes son  $(k+1)$ -equivalentes,  $(k+2)$ -equivalentes, etc. y se dice que  $P_k$  es una partición de equivalencia. En el ejemplo, una rápida verificación indica que  $P_4 = P_3$  y, por tanto, los estados  $B$  y  $C$  son  $K$ -equivalentes para toda  $K$ ; es decir, son equivalentes.  $P_k$  satisface entonces la condición 2 para los estados equivalentes.

Al seguir este procedimiento en el ejemplo de la figura 9.7 se deben revisar los grupos de estados en cada bloque de  $P_1$ . En el primer bloque, los estados siguientes para  $A$ ,  $B$  y  $C$  con  $x = 0$  están todos en el mismo bloque de  $P_1$ , sin embargo, para  $x = 1$ , el estado siguiente de  $A$  está en un bloque de  $P_1$  distinto del bloque en que se encuentran los estados siguientes de  $B$  y  $C$ , por tanto, el bloque  $(A \bullet B \bullet C)$  contenido en  $P_1$  se divide en los bloques  $(A) \bullet (B \bullet C)$  en  $P_2$ . Los estados siguientes de los estados  $D$  y  $E$  están en el mismo bloque de  $P_1$  para  $x = 0$  y  $x = 1$ , por lo que  $D$  y  $E$  permanecen en el mismo bloque de  $P_2$ . Así  $P_2 = (A) \bullet (B \bullet C) \bullet (D \bullet E)$  y los estados dentro de cada bloque son 2-equivalentes.

Se obtiene la partición  $P_3$  examinando cada bloque de  $P_2$ . Los estados siguientes  $B$  y  $C$  se encuentran en el mismo bloque de  $P_2$  para cada entrada, por lo que el bloque  $(B \bullet C)$  permanece intacto en  $P_3$ . Sin embargo, los estados siguientes  $D$  y  $E$  con  $x = 1$  están en bloques diferentes de  $P_2$ , por lo que estos dos estados deben aparecer en bloques distintos de  $P_3$ . Por tanto,  $P_3 = (A) \bullet (B \bullet C) \bullet (D) \bullet (E)$ . Así que sólo los estados  $B$  y  $C$  son 3-equivalentes.

Este procedimiento de obtención de particiones sucesivas se repite hasta lograr la condición establecida en el paso 3.

	Bloques de partición	Acción
Partición P <sub>0</sub>	(A•B•C•D)	
Salida para X=0	11100	Separar (A•B•C) y (D•E)
Salida para X=1	00011	Separar (A•B•C) y (D•E)
Partición P <sub>1</sub>	(A•B•C)   (D•E)	
Estado siguiente para X=0	C•C•B   D•E	
Estado siguiente para X=1	B•E•E   B•A	Separar (A) y (B•C)
Partición P <sub>2</sub>	(A)   (D•E)	
Estado siguiente para X=0	C   C•B   D•E	
Estado siguiente para X=1	E   E•E   B•A	Separar (D) y (E)
Partición P <sub>3</sub>	(A)   (BC)   (D)   (E)	
Estado siguiente para X=0	C   C•B   D   E	
Estado siguiente para X=1	E   E•E   B   A	
Partición P <sub>4</sub> =P <sub>3</sub>	(A)   (BC)   (D)   (E)	

Los estados B y C son equivalentes

Figura 9.8.: Equivalencia de estados mediante partición

### 9.3.3. Tabla de Implicación

La tabla de implicación es otra herramienta que puede servir para determinar la equivalencia de estados. Esta técnica es más general, pues también es aplicada a los circuitos secuenciales con especificación incompleta; sin embargo, puede ocupar mayor tiempo que el método de partición.

El método de la tabla de implicación se explica retomando el ejemplo de la figura 9.7.

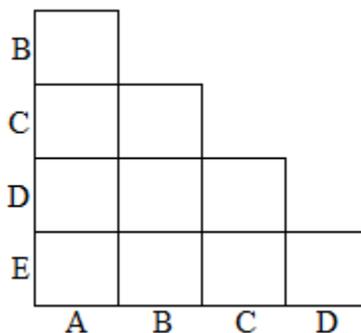


Figura 9.9.: Tabla de implicación

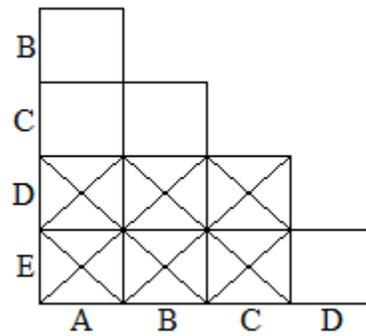


Figura 9.10.: Partición de salida

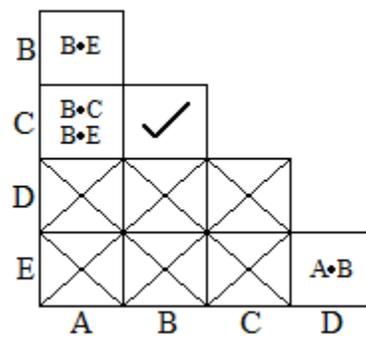


Figura 9.11.: Tabla terminada

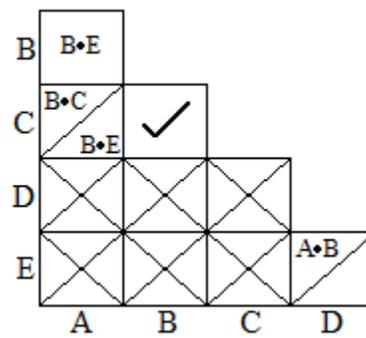


Figura 9.12.: Tabla terminada

$$\begin{array}{c|c}
 A & \\
 B & (B \cdot C) \\
 C & \\
 D & \\
 \hline
 P_k = & (A) \cdot (B \cdot C) \cdot (D) \cdot (E)
 \end{array}$$

Figura 9.13.: Partición de equivalencia

### Procedimiento de la tabla de implicación

- Paso 1: Se forma una tabla con la estructura que se muestra en la figura 9.9, que se obtiene listando en forma vertical todos los estados de la tabla excepto el último y de forma horizontal se anotan todos los estados de la tabla, excepto el primero. El resultado de la tabla muestra todas las combinaciones posibles de dos estados, y por tanto, cada celda de la tabla correspondiente a la intersección de una fila y una columna representa dos estados cuya equivalencia se quiere verificar.
- Paso 2: Puesto que sólo los estados con salidas idénticas podrían ser equivalentes, se coloca una cruz en las celdas correspondientes a los pares de estados cuyas salidas no son iguales para todas las entradas. En el caso del ejemplo esto se puede observar en la figura 9.10.
- Paso 3: Se utiliza la condición 2 de los estados equivalentes para completar las celdas vacantes de la figura 9.11. Se coloca en estos bloques los pares de estados siguientes cuya equivalencia es implicada por los dos estados que define la celda por la intersección. Como muestra de esto, se considera la celda definida por los estados  $A$  y  $B$ . De la tabla de estados se puede ver que, para que  $A$  y  $B$  sean estados equivalentes,  $B$  y  $E$  deben ser equivalentes. Dado esto, se anota el par  $B \bullet E$  en la celda definida por  $A$  y  $B$ , como se muestra en la figura 9.11. Una aclaración importante es que si los estados del par implicado,  $B$  y  $E$ , no son equivalentes, entonces existe una cadena de entrada que comienza con  $x = 0$  y produce salidas distintas dependiendo de si el estado inicial es  $A$  o  $B$ , lo que quiere decir que  $A$  y  $B$  no son equivalentes. Si los pares implicados para cada celda sólo contienen los estados que definen a la celda o si los estados siguientes de los dos estados que definen a la celda son el mismo estado para una entrada dada, se coloca una marca de verificación (✓) en la celda, lo cual indica que los dos estados que definen a la celda son equivalentes por inspección e independientes de cualquier par implicado. La figura 9.11 muestra esta condición mediante la celda definida por los estados  $B$  y  $C$ .
- Paso 4: Una vez que se encuentre llena la tabla, esta se recorre en forma sucesiva para determinar si hay que tachar otras celdas distintas a las ya tachadas en el paso 2. Se tacha una celda de la

tabla si contiene al menos un par implicado que defina una celda de la tabla y tachada con anterioridad. Hecha esta operación para el ejemplo, se obtiene la tabla de la figura 9.12. Por ejemplo, la celda definida por  $A$  y  $B$  se ha tachado debido que contiene el par  $B \bullet E$ , el cual define una celda ya eliminada. Este proceso se repite hasta que ya no se puedan tachar más celdas.

- Paso 5: Por último, se obtiene la tabla de la figura 9.13 enumerando en una columna los estados que definen la fila horizontal de la tabla de implicación. Seguidamente se analiza la tabla de implicación, columna por columna, de izquierda a derecha, con el fin de verificar si no se han tachado otras celdas. Los estados que definen a las celdas no tachadas son equivalentes y se enumeran como pares equivalentes en la tabla de la figura 9.13, los pares se combinan mediante la transitividad.  $(S_i, S_j)(S_j, S_k) \rightarrow (S_i, S_j, S_k)$ . En el ejemplo, todas las celdas de las columnas  $A$ ,  $C$  y  $D$  se tachan y, por tanto, se escriben guiones en las filas correspondientes de la tabla de la figura 9.13. En la columna  $B$  de la tabla de implicación, no se tacha la celda definida por los estados  $B$  y  $C$  y, por tanto, se coloca el par  $(B \bullet C)$  en la fila  $B$  de la tabla. Así la partición de equivalencia consta de todos los estados equivalentes determinados en la tabla; es decir,  $(B \bullet C)$  junto con los demás estados del circuito que no son equivalentes a algún otro estado. Observese que esta partición de equivalencia es idéntica a la obtenida anteriormente mediante el método de partición.





## 10. LÓGICA CABLEADA

La lógica cableada industrial es la técnica de diseño de pequeños a complejos autómatas utilizados en plantas industriales, básicamente con relés cableados. Para los técnicos en telecomunicaciones y en informática, la lógica cableada utiliza compuertas lógicas discretas (TTL, CMOS, HCMOS), para implementar circuitos digitales de comunicaciones y computadores. La lógica cableada industrial consiste en el diseño de automatismos con circuitos cableados entre contactos auxiliares de relés electromecánicos, contactores de potencia, relés temporizados, diodos, relés de protección, válvulas óleo-hidráulicas o neumáticas y otros componentes. Los cableados incluyen funciones de comando y control, de señalización, de protección y de potencia. La potencia además de circuitos eléctricos comprende a los circuitos neumáticos (mando por aire a presión) u óleo-hidráulicos (mando por aceite a presión). Crea automatismos rígidos, capaces de realizar una serie de tareas en forma secuencial, sin posibilidad de cambiar variables y parámetros. Si se ha de realizar otra tarea, es necesario realizar un nuevo diseño. Se emplea en automatismos pequeños, o en lugares críticos, donde la seguridad de personas y máquinas, no puede depender de la falla de un programa de computación. En sistemas mayores también se emplea el autómata programable, entre los que se encuentran los PLC (controlador lógico programable), la RTU (Unidad Terminal Remota) o los relés programables, computadoras o servidores de uso industrial. Estos autómatas no se programan en lenguajes tradicionales como cualquier computador, generalmente para sistemas de bajo costo se utiliza el lenguaje ladder, en dicho lenguaje las instrucciones no son otra cosa que líneas de lógica cableada. Así el conocimiento de la lógica cableada es de fundamental importancia para quien programa un autómata programable o PLC. La lógica cableada más que una técnica, hoy en día constituye una filosofía que permite estructurar circuitos en forma ordenada, prolija y segura, sea en circuitos cableados o programados. La práctica de la lógica cableada ha sido asimilada por otras ramas de la tecnología como las telecomunicaciones y la informática, con la introducción del cableado estructurado en edificios, oficinas y locales comerciales, lugares donde es poco usual el manejo de esquemas y dibujos de las instalaciones eléctricas, excepto la de potencia, la elaboración de proyectos de detalle y el cableado en forma ordenada mediante el uso de borneras y regletas, que pasaron a llamarse “patcheras” en el caso de las redes de datos y telefonía.[2, 4]

## 10.1. EL CONTACTOR

Se puede definir un contactor como un aparato mecánico de conexión y desconexión eléctrica, accionado por cualquier forma de energía, menos manual, capaz de establecer, soportar e interrumpir corrientes en condiciones normales del circuito, incluso las de sobrecarga.

Las energías utilizadas para accionar un contactor pueden ser muy diversas: mecánicas, magnéticas, neumáticas, hidráulicas, etc. Los contactores corrientemente utilizados en la industria son accionados mediante la energía magnética proporcionada por una bobina.[5]

Un contactor accionado por energía magnética, consta de un núcleo magnético y de una bobina capaz de generar un campo magnético suficientemente grande como para vencer la fuerza de los muelles antagonistas que mantienen separada del núcleo una pieza, también magnética, solidaria al dispositivo encargado de accionar los contactos eléctricos.

Una característica importante de un contactor será la tensión a aplicar a la bobina de accionamiento, así como su intensidad ó potencia. Según sea el fabricante, se dispone de una extensa gama de tensiones de accionamiento, tanto en corriente continua como en corriente alterna. El tamaño de un contactor, depende de la intensidad que es capaz de establecer, soportar e interrumpir, así como del número de contactos que dispone (normalmente cuatro). El tamaño del contactor también depende de la tensión máxima de trabajo que puede soportar.[14]

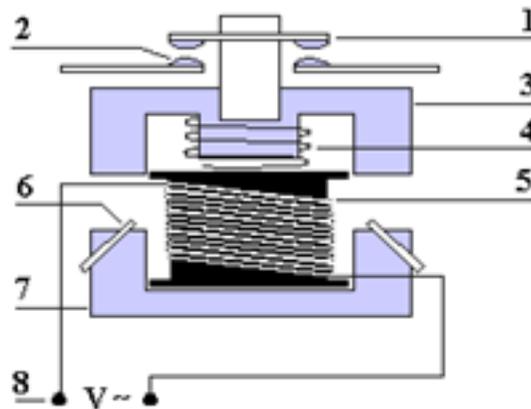
### 10.1.1. Principales partes del contactor

En la figura 10.1 se pueden apreciar algunas partes del contactor.

**Carcasa:** La carcasa es el elemento en el cual se fijan todos los componentes conductores del contactor, para lo cual es fabricada en un material no conductor con propiedades como la resistencia al calor, y un alto grado de rigidez. Uno de los materiales más utilizados es la fibra de vidrio pero tiene un inconveniente y es que este material es quebradizo y por lo tanto su manipulación es muy delicada.[5]

**Electroimán:** También es denominado circuito electromagnético, y es el elemento motor del contactor. Está compuesto por una serie de elementos cuya finalidad es transformar la energía eléctrica en un campo magnético muy intenso mediante el cual se produce un movimiento mecánico aprovechando las propiedades electromagnéticas de ciertos materiales.[2]

**Bobina:** Consiste en un arrollamiento de alambre de cobre con unas características muy especiales, con un gran número de espiras y de sección muy delgada para producir un campo magnético. El flujo magnético produce un par magnético que vence los pares resistentes de los muelles de manera que la armadura se puede juntar con el núcleo estrechamente.[5]



1. Contactos móviles. 2. Contactos fijos. 3. Hierro móvil. 4. Muelle antagonista. 5. Bobina. 6. Espira de sombra (en corriente alterna). 7. Hierro fijo. 8. Alimentación bobina.

Figura 10.1.: Partes del contactor

- **Bobina energizada con CA:** Para el caso cuando una bobina se energiza con corriente alterna, se produce una corriente de magnitud muy alta puesto que sólo se cuenta con la resistencia del conductor, ya que la reactancia inductiva de la bobina es muy baja debido al gran entrehierro que existe entre la armadura y el núcleo, esta corriente tiene factor de potencia por consiguiente alto, del orden de 0.8 a 0.9 y es conocida como *corriente de llamada*. Esta corriente elevada produce un campo magnético muy grande capaz de vencer el par ejercido por los muelles o resorte que los mantiene separados y de esta manera se cierra el circuito magnético uniéndose la armadura con el núcleo trayendo como consecuencia el aumento de la reactancia inductiva y así la disminución de hasta aproximadamente diez veces la corriente, produciéndose entonces una corriente llamada *corriente de mantenimiento* con un factor de potencia más bajo pero capaz de mantener el circuito magnético cerrado.[4]
- **Bobina energizada con CC:** En este caso no se presenta el fenómeno anterior puesto que las corrientes de llamada y de mantenimiento son iguales. La única resistencia presente es la resistencia de la bobina misma por lo cual las características y la construcción de estas bobinas son muy especiales. La bobina puede ser energizada por una fuente de alimentación o por una fuente independiente.[14]

**El núcleo:** Su función es concentrar y aumentar el flujo magnético con el fin de atraer la armadura eficientemente. Está construido de láminas de acero al silicio superpuestas y unidas firmemente unas con otras con el fin de evitar las corrientes parásitas. Cuando circula una corriente alterna por la bobina, se supone que cuando la corriente pasa por el valor cero, el núcleo se separa de la armadura puesto que el flujo también es cero pero como esto sucede 120 veces en un segundo (si la frecuencia es de 60Hz) por lo cual en realidad no hay una verdadera separación pero esto sin embargo genera vibraciones y

un zumbido además del aumento de la corriente de mantenimiento; por esto las bobinas que operan con corriente alterna poseen unos dispositivos llamados espiras de sombra las cuales producen un flujo magnético desfasado con el principal de manera que se obtiene un flujo continuo similar al producido por una corriente continua.[14]

**Armadura:** Es un elemento móvil muy parecido al núcleo pero no posee espiras de sombra, su función es la de cerrar el circuito magnético ya que en estado de reposo se encuentra separada del núcleo. Este espacio de separación se denomina *entrehierro* o *cota de llamada*.

Tanto el cierre como la apertura del circuito magnético suceden en un espacio de tiempo muy corto (10 milisegundos aproximadamente), todo debido a las características del muelle, por esto se pueden presentar dos situaciones.[5]

- Cuando el par resistente es mayor que el par electromagnético, no se logra atraer la armadura.
- Si el par resistente es débil no se logra la separación rápida de la armadura.

Cada una de las acciones de energizar o desenergizar la bobina y por consiguiente la atracción o separación de la armadura, es utilizada para accionar los contactos que obran como interruptores, permitiendo o interrumpiendo el paso de la corriente. Estos contactos están unidos mecánicamente pero son separados eléctricamente.[5]

**Contactos:** El objeto de estos elementos es permitir o interrumpir el paso de la corriente, son elementos conductores, los cuales se accionan tan pronto se energiza o se desenergiza la bobina por lo que se les denomina contactos instantáneos. Esta función la cumplen tanto en el circuito de potencia como en el circuito de mando. Los contactos están compuestos por tres partes, dos de las cuales son fijas y se encuentran ubicadas en la carcasa y una parte móvil que une estas dos y posee un resorte para garantizar el contacto.[14]

Las partes que entran en contacto deben tener unas características especiales puesto que al ser accionados bajo carga, se presenta un arco eléctrico el cual es proporcional a la corriente que demanda la carga, estos arcos producen sustancias que deterioran los contactos pues traen como consecuencia la corrosión

- **Contactos principales:** Son los encargados de permitir o interrumpir el paso de la corriente en el circuito principal, es decir que actúa sobre la corriente que fluye de la fuente hacia la carga. Es recomendable verificar la separación de estos, ya que permiten que las partes fijas y móviles se junten antes de que el circuito magnético se cierre completamente, esta distancia se le denomina *cota de presión*.

Debido a que los contactos operan bajo carga, es determinante poder extinguir el arco que se produce puesto que esto deteriora el dispositivo debido a las temperaturas extremadamente altas,

para esto, los contactos se encuentran instalados dentro de la llamada *cámara apagachispas*. La extinción del arco se logra mediante diferentes mecanismos.[4, 5]

- **Soplado por auto-ventilación:** Este dispositivo consiste en dos aberturas, una grande y una pequeña, al calentarse el aire, este sale por la abertura pequeña entrando aire fresco por la abertura grande y este movimiento de aire hace que se extinga la chispa.[2]
  - **Cámaras desionizadoras:** Estas cámaras consisten en un recubrimiento metálico que actúa como un disipador de calor y por esto el aire no alcanza la temperatura de ionización. Este método suele acompañarse por el soplado por auto-ventilación.[4]
  - **Transferencia y fraccionamiento del arco:** Consiste en dividir la chispa que se produce, de manera que es más fácil de extinguir chispas más pequeñas.[4]
  - **Soplo magnético:** Este método emplea un campo magnético que atrae la chispa hacia arriba de la cámara aumentando de esta manera la resistencia. Este método suele ir acompañado del soplado por auto-ventilación y se debe realizar en un tiempo no muy largo pero tampoco extremadamente corto.[5]
- **Contactos secundarios:** Estos contactos secundarios se encuentran dimensionados para corrientes muy pequeñas porque estos actúan sobre la corriente que alimenta la bobina del contactor o sobre elementos de señalización.[5]

Dado que en ocasiones deben trabajar con los PLC, estos contactos deben tener una confiabilidad muy alta.

Gran parte de la versatilidad de los contactores depende del correcto uso y funcionamiento de los contactos auxiliares. Normalmente los contactos auxiliares son:[5]

- **Instantáneos:** Actúan tan pronto se energiza la bobina del contactor.
- **De apertura lenta:** La velocidad y desplazamiento del contacto móvil es igual al de la armadura.
- **De apertura positiva:** Los contactos abiertos y cerrados no pueden coincidir cerrados en ningún momento.

Sin embargo se encuentran contactores auxiliares con adelanto al cierre o a la apertura y con retraso al cierre o a la apertura. Estos contactos actúan algunos milisegundos antes o después que los contactos instantáneos. Existen dos clases de contactos auxiliares:

- **Contacto normalmente abierto (NA ó NO):** llamado también contacto instantáneo de cierre, cuya función es cerrar un circuito tan pronto se energiza la bobina del contactor. En estado de reposo se encuentra abierto.[2]
- **Contacto normalmente cerrado (NC):** llamado también contacto instantáneo de apertura, cuya función es abrir un circuito tan pronto se energiza la bobina del contactor. En estado de reposo se encuentra cerrado.[2]

### 10.1.2. Funcionamiento del contactor

Cuando la bobina se energiza genera un campo magnético intenso, de manera que el núcleo atrae a la armadura, con un movimiento muy rápido. Con este movimiento todos los contactos del contactor, principales y auxiliares, cambian inmediatamente y de forma solidaria de estado.

Existen dos consideraciones en cuanto a las características de los contactores:[5]

- **Poder de cierre:** Valor de la corriente independientemente de la tensión, que un contactor puede establecer en forma satisfactoria y sin peligro que sus contactos se suelden.
- **Poder de corte:** Valor de la corriente que el contactor puede cortar, sin riesgo de daño de los contactos y de los aislantes de la cámara apagachispas.

Para que los contactos vuelvan a su posición anterior es necesario desenergizar la bobina. Durante esta desenergización o desconexión de la bobina (carga inductiva) se producen sobretensiones de alta frecuencia, que pueden producir interferencias en los aparatos electrónicos.

### 10.1.3. Categoría de empleo

Para establecer la categoría de empleo se tiene en cuenta el tipo de carga controlada y las condiciones en las cuales se efectúan los cortes.

Las categorías más usadas en AC son:[2, 4]

- **AC1:** Cargas no inductivas (resistencias, distribución) o débilmente inductivas, cuyo factor de potencia sea por lo menos 0.95.
- **AC2:** Se refiere al arranque, al frenado en contracorriente y a la marcha por impulso permanente de los motores de anillos. Al cierre el contactor establece el paso de corrientes de arranque equivalentes a más o menos 2.5 la corriente nominal del motor. A la apertura el contactor debe cortar la intensidad de arranque, con una tensión inferior o igual a la tensión de la red.

- **AC3:** Para el control de motores jaula de ardilla (motores de rotor en cortocircuito) que se apagan a plena marcha. Al cierre se produce el paso de corrientes de arranque, con intensidades equivalentes a 5 o más veces la corriente nominal del motor. A la apertura corta el paso de corrientes equivalentes a la corriente nominal absorbida por el motor. Es un corte relativamente fácil.
- **AC4:** Se refiere al arranque, al frenado en contracorriente y a la marcha por impulso permanente de los motores de jaula. Al cierre se produce el paso de la corriente de arranque, con intensidades equivalentes a 5 o más veces la corriente nominal del motor. Su apertura provoca el corte de la corriente nominal a una tensión, tanto mayor como tanto mayor es la velocidad del motor. Esta tensión puede ser igual a la tensión de la red.

## 10.2. EI RELÉ

Un relé es un conmutador eléctrico especializado que permite controlar dispositivos de gran potencia (por ejemplo un motor) mediante un dispositivo de potencia mucho menor. Su constitución y finalidad es similar a las de un contactor, la diferencia radica en que en el relé, todos sus contactos son auxiliares por lo que se emplea en el control de circuitos que requieran auto-mantenimiento, enclavamiento de contactos, protección y señalización.[2]

## 10.3. RELÉ DE ENCLAVAMIENTO

Esta clase de relé posee dos bobinas, una principal de funcionamiento normal, la cual al efectuarse un paso de corriente por ella cambia el estado de los contactos auxiliares, pero una vez desenergizada, los contactos no vuelven a su posición original de reposo debido a la operación de un dispositivo de enclavamiento. Para que los contactos vuelvan a su estado inicial (reposo) es necesario energizar la segunda bobina para eliminar el enclavamiento.[4]

Este relé es empleado como función de memoria para procesos, permitiendo así que la información o los pasos de este no se pierdan al efectuarse un corte de energía, y reanudar en la posición correcta.

## 10.4. CONTACTOR CON BOBINA DE AUTO-RETENCIÓN

Es un contactor que posee una bobina adicional además de la bobina principal, la bobina adicional es de bloqueo o auto-retención. La función de esta bobina es evitar el paso de corriente por la bobina

principal, y por ende los contactos se encuentran en reposo. Si se energiza la bobina de bloqueo los contactos no actúan así este energizada la bobina principal. Un inconveniente con la bobina de bloqueo es que, trabaja impidiendo el recorrido de la bobina principal desde su estado de reposo hasta su energización, es decir, no sirve como medio para que los contactos regresen a su estado de reposo.[5]

## 10.5. RELÉS DE TEMPORIZACIÓN

Son relés en los cuales abren o cierran determinados contactos llamados contactos temporizados. Los contactos temporizados actúan después de cierto tiempo que se ha programado previamente en el dispositivo. En el momento de energizar un temporizador tipo *ON*, se tiene que, los contactos temporizados asociados a la bobina de temporización, siguen en la misma posición de reposo y solamente cuando ha transcurrido el tiempo programado, cambian de estado, es decir que el contacto *NA* se cierra y el contacto *NC* se abre.[4, 14]

En los relés tipo *OFF* se presenta una acción de apertura o cierre instantáneo cuando la bobina es energizada, y una conmutación de apertura o cierre temporizado al momento de la desenergización. Cabe decir que los relés tipo *ON* como los relés tipo *OFF* pueden tener contactos auxiliares que actúan de manera instantánea al presentarse una energización o desenergización de la bobina.

En la figura 10.2 se puede apreciar la manera como se accionan los contactos temporizados según el tipo del relé de temporización.

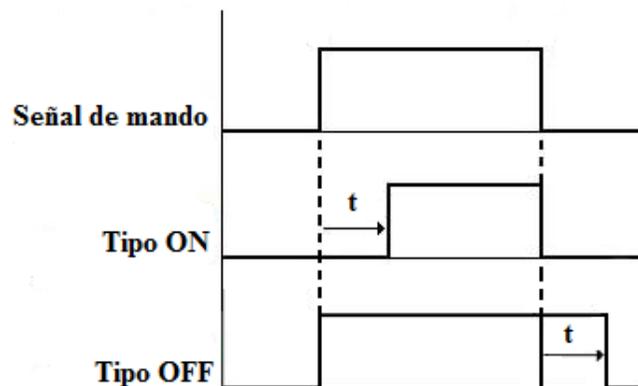


Figura 10.2.: Accionamiento contactos temporizados

## 10.6. ELEMENTOS DE MANDO

Son elementos empleados para ingresar las órdenes hacia los dispositivos de control con el fin de actuar sobre los órganos receptores, los cuales en general son elementos de potencia que forman la parte operativa del sistema. A continuación se muestran los más utilizados:[2, 4]

**Breaker:** Es un dispositivo que permite conectar o desconectar la alimentación de un circuito ya sea en condiciones normales o anormales (cortocircuito y sobrecarga), además de proveer protección (térmica y magnética).

**Pulsadores:** Elementos que permiten el paso o interrupción de la corriente mientras son accionados. Cuando ya no se actúa sobre ellos vuelven a su estado de reposo que puede ser normalmente cerrado *NC* o normalmente abierto *NA*.

**Controles de nivel:** Los controles de nivel son dispositivos o estructuras hidráulicas cuya finalidad es la de garantizar el nivel de un fluido en un rango de variación preestablecido.

**Controles de presencia:** Los sensores de presencia tienen como objetivo determinar la presencia de un objeto en un intervalo de distancia especificada.



**Parte III.**

# **METODOLOGÍAS DE DISEÑO**



# 11. METODOLOGÍA POR SEÑALES DE MANDO

## 11.1. INTRODUCCIÓN

La metodología que se muestra a continuación se centra en la obtención de una ecuación de salida  $S$  en función de las señales intermedias de mando, de manera que  $S(f_1, f_2, \dots, f_n)$  y cada función intermedia en función de las señales de mando. Existen casos en los que sea necesaria la utilización de varias señales intermedias de mando.

Entre las señales de mando se destacan las provenientes de pulsadores, interruptores, breakers, sensores de nivel o de presencia, quienes controlan la aparición de las señales intermedias de mando como lo son las pertenecientes a los temporizadores, de manera que se ejecute el desarrollo de un automatismo y se logre el buen funcionamiento de este, lo que se ve representado en una adecuada señal de salida, siendo esta última quien registra el comportamiento de los órganos receptores presentes en un automatismo.

Un ejemplo de los conceptos anteriores se evidencia en el caso del control de una bombilla a través de un pulsador, en este caso el pulsador representa la señal de mando, ya que este es quien controla la activación del elemento receptor, pero además es el encargado de propiciar la aparición de una señal intermedia de mando, que a su vez ejerce control sobre la bombilla, mientras que el comportamiento de esta última se ve representado por una señal de salida. De manera gráfica la descripción anterior puede verse en la figura 11.1.

A través del manejo de las señales de mando se logra la implementación de algunas funciones, divididas en *funciones básicas* y *funciones derivadas*, su diferencia radica en que las primeras no se obtienen como combinación de otras funciones, mientras que las segundas se obtienen a partir de arreglos de las funciones básicas, sin embargo las funciones derivadas se especifican gracias a su gran utilidad. Tanto las funciones básicas como las funciones derivadas pueden ser utilizadas según convenga para el desarrollo de automatismos ya que permiten encontrar un diagrama de contactos que cumple con los requerimientos exigidos por una situación que se desea realizar de manera automática.

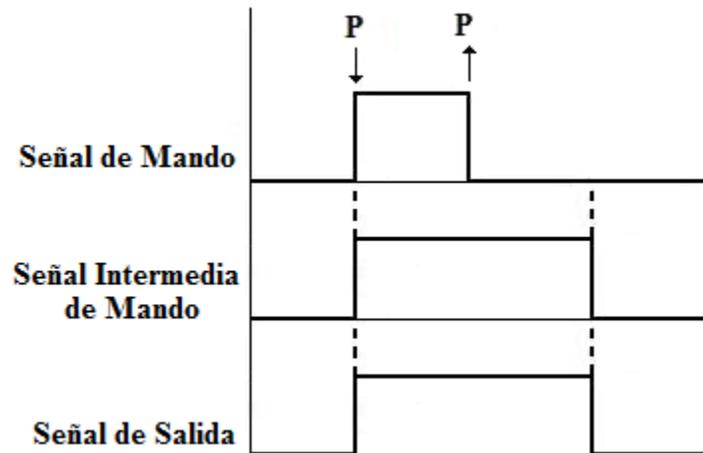


Figura 11.1.: Señal de mando y señal de salida

Dentro de las funciones básicas se destacan las funciones *pulsador* e *interruptor* , ya que estas permiten la activación de las demás funciones.

## 11.2. FUNCIONES BÁSICAS

### 11.2.1. Función pulsador (mando monoestable)

La salida es una señal que toma un valor lógico de “1” mientras la señal de mando este presente y en “0” en caso contrario.

Esta señal la entrega directamente cualquier señal monoestable de mando, pero se presenta por ser necesaria como elemento de mando generado por señales intermedias.

Diagrama de tiempo:

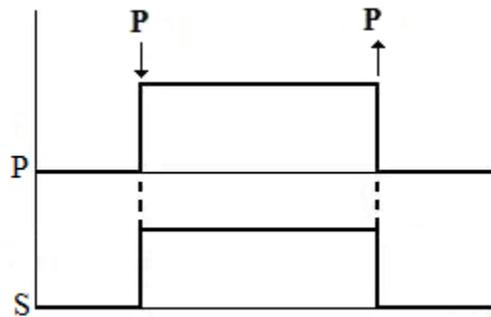


Figura 11.2.: Diagrama de tiempo función pulsador

Ecuación:

$$S = P$$

Diagrama Ladder:



Figura 11.3.: Diagrama Ladder función pulsador

El circuito Ladder de ésta función consta de una bobina asociada al pulsador de mando que al ser presionado se obtiene la señal de salida.

Diagrama de bloque:

El diagrama de bloque mostrado en la figura 11.4 representa el diagrama Ladder de la función pulsador.

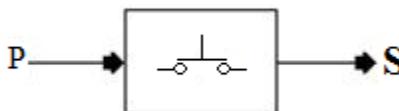


Figura 11.4.: Bloque función pulsador

### 11.2.2. Función interruptor (mando biestable)

La salida es una señal que toma un valor lógico de “1” desde la activación de una señal monoestable de mando  $A$  y hasta la activación (o desactivación) de una segunda señal de mando monoestable  $P$ .

Diagrama de tiempo:

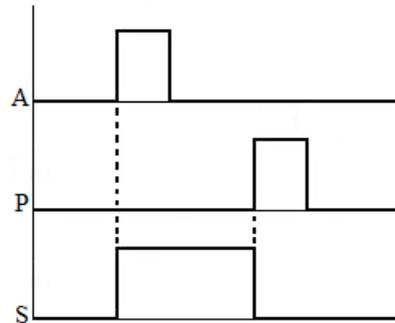


Figura 11.5.: Diagrama de tiempo función interruptor

Ecuación:

$$S = (A + S) \cdot \bar{P}$$

Diagrama Ladder:

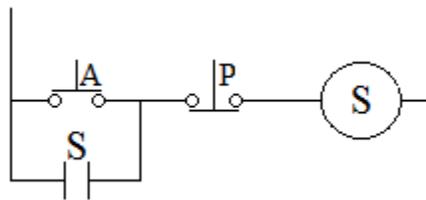


Figura 11.6.: Diagrama Ladder función interruptor

El circuito Ladder de esta función contiene dos pulsadores monoestables, uno de activación  $A$  y otro de desactivación  $P$ . Cuando se acciona  $A$  permite la energización de un relé y se mantiene en ese estado hasta el momento en el que se pulsa  $P$  para la desenergización.

Diagrama de bloque:

El diagrama de bloque mostrado en la figura 11.7 representa el diagrama Ladder de la función interruptor.

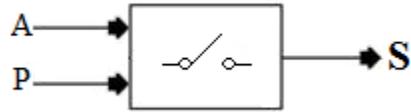


Figura 11.7.: Bloque función interruptor

### 11.2.3. Función seguidor

La salida es una señal que presenta un comportamiento idéntico a la señal de mando.

Diagrama de tiempo:

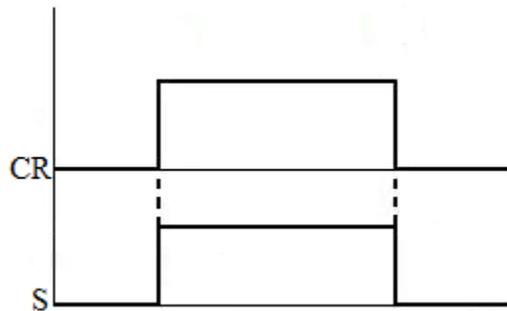


Figura 11.8.: Diagrama de tiempo función seguidor

Ecuación:

$$S = CR$$

Diagrama Ladder:



Figura 11.9.: Diagrama Ladder función seguidor

El circuito Ladder contiene un pulsador que al ser accionado energiza una bobina cerrando un contacto *NA*, para así obtener la señal de salida.

Diagrama de bloque:

El diagrama de bloque mostrado en la figura 11.10 representa el diagrama Ladder de la función seguidor.



Figura 11.10.: Bloque función seguidor

#### 11.2.4. Función NOT (inversor)

La salida es una señal que presenta un comportamiento contrario a la señal de mando.

Diagrama de tiempo:

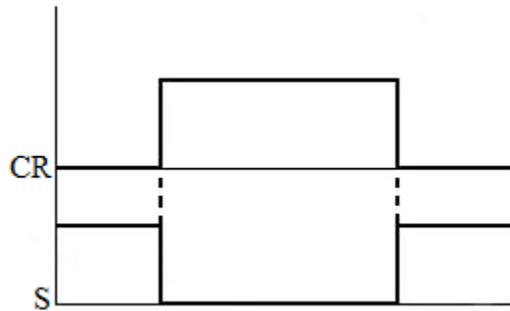


Figura 11.11.: Diagrama de tiempo función NOT

Ecuación:

$$S = \overline{CR}$$

Diagrama Ladder:



Figura 11.12.: Diagrama Ladder función NOT

El circuito Ladder contiene un pulsador, que al ser accionado energiza una bobina abriendo un contacto *NC*, permitiendo la desenergización de la señal de salida, llevándola de un estado lógico “1” a un estado lógico “0”, volviendo a su estado original una vez se deje de presionar el pulsador.

Diagrama de bloque:

El diagrama de bloque mostrado en la figura 11.13 representa el diagrama Ladder de la función NOT.

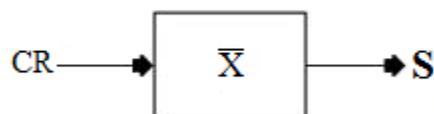


Figura 11.13.: Bloque función NOT

### 11.2.5. Función AND

La salida es una señal que depende de dos o más señales de entrada, las cuales deben de estar en estado lógico “1” para que la salida se presente, de lo contrario la señal de salida permanece en estado lógico “0”.

Diagrama de tiempo:

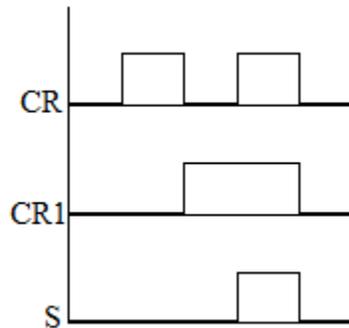


Figura 11.14.: Diagrama de tiempo función AND

Ecuación:

$$S = CR \bullet CR1$$

Diagrama Ladder:

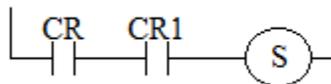


Figura 11.15.: Diagrama Ladder función AND

El circuito Ladder permite el paso de corriente a la señal de salida siempre y cuando los contactos de la bobina a la que se encuentre asociados estén cerrados.

Diagrama de bloque:

El diagrama de bloque mostrado en la figura 11.16 representa el diagrama Ladder de la función AND.

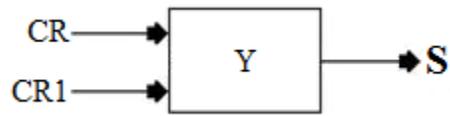


Figura 11.16.: Bloque función AND

### 11.2.6. Función OR

La salida es una señal que depende de dos o más señales de entrada, de las cuales, al menos una debe estar en estado lógico “1” para que la salida se presente. La salida es “0” cuando todas las señales de entrada sean “0”.

Diagrama de tiempo:

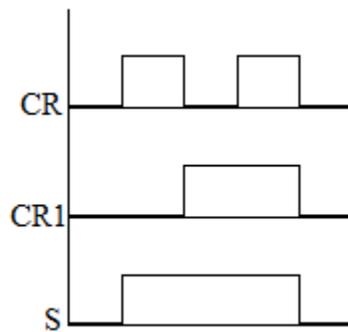


Figura 11.17.: Diagrama de tiempo función OR

Ecuación:

$$S = CR + CR1$$

Diagrama Ladder:

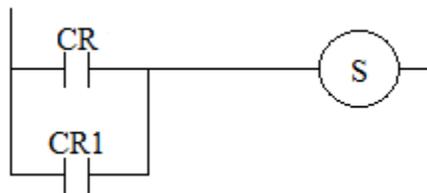


Figura 11.18.: Diagrama Ladder función OR

El circuito Ladder permite el paso de corriente a la señal de salida siempre y cuando al menos uno o más contactos asociados a la bobina previamente energizada estén cerrados.

Diagrama de bloque:

El diagrama de bloque mostrado en la figura 11.19 representa el diagrama Ladder de la función OR.



Figura 11.19.: Bloque función OR

### 11.2.7. Función temporización a la excitación

La salida es una señal que hace el paso del estado lógico “0” al estado lógico “1” un tiempo  $t$  después de que lo haya hecho la señal de mando, y se mantiene en ese estado hasta que la entrada se desenergice, es en ese momento cuando la salida vuelve a su estado original.

Diagrama de tiempo:

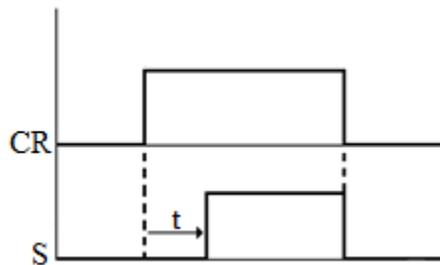


Figura 11.20.: Diagrama de tiempo función temporización a la excitación

Ecuación:

$$S = CR^{te}$$

Diagrama Ladder:

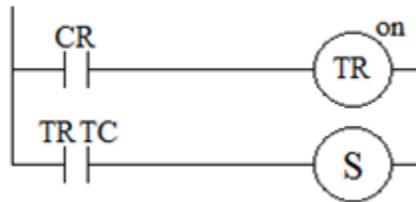


Figura 11.21.: Diagrama Ladder función temporización a la excitación

El circuito Ladder contiene un interruptor, el cual, al ser presionado energiza una bobina cerrando así un contacto *NA* asociado a la bobina previamente energizada. Este contacto como se encuentra en estado cerrado, permite el paso de la corriente, energizando la bobina de temporización *TR ON*, la cual retarda un tiempo  $t$  programado para que se presente el cierre del contacto *TR TC*, para obtener finalmente la señal de salida.

Diagrama de bloque:

El diagrama de bloque mostrado en la figura 11.22 representa el diagrama Ladder de la función temporización a la excitación.

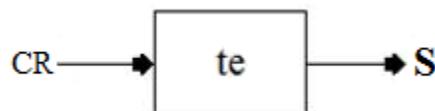


Figura 11.22.: Bloque función temporización a la excitación

### 11.2.8. Función temporización a la excitación negada

La salida es una señal que hace el paso del estado lógico “1” al estado lógico “0” un tiempo  $t$  después de que lo haya hecho la señal de mando, y se mantiene en ese estado hasta que la entrada se desenergice, es en ese momento cuando la salida vuelve a su estado original.

Diagrama de tiempo:

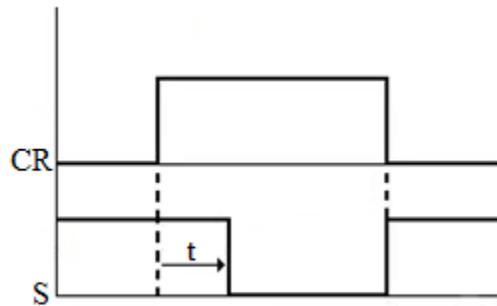


Figura 11.23.: Diagrama de tiempo función temporización a la excitación negada

Ecuación:

$$S = \overline{CR}^{te}$$

Diagrama Ladder:

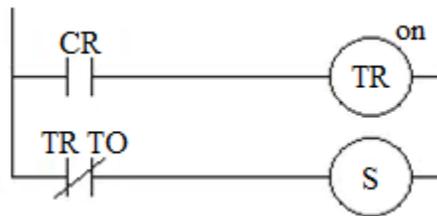


Figura 11.24.: Diagrama Ladder función temporización a la excitación negada

El circuito Ladder contiene un interruptor, el cual, al ser presionado energiza un bobina cerrando así un contacto *NA* asociado a la bobina previamente energizada. Este contacto como se encuentra en estado cerrado, permite el paso de la corriente, energizando la bobina de temporización *TR ON* la cual retarda un tiempo *t* programado para abrir el contacto *TR TO*, desenergizando así la señal de salida.

Diagrama de bloque:

El diagrama de bloque mostrado en la figura 11.25 representa el diagrama Ladder de la función temporización a la excitación negada.

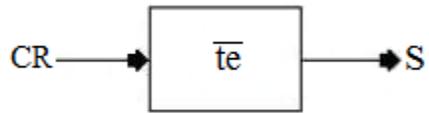


Figura 11.25.: Bloque función temporización a la excitación negada

### 11.2.9. Función temporización a la desexcitación

La salida es una señal que hace el cambio del estado lógico “0” al estado lógico “1” cuando la señal de mando hace su aparición, el comportamiento de esta función se observa una vez la señal de entrada es desenergizada, es entonces cuando la salida se retarda un tiempo  $t$  para que efectúe el cambio de estado lógico “1” al estado lógico “0” o vuelva a su posición original.

Diagrama de tiempo:

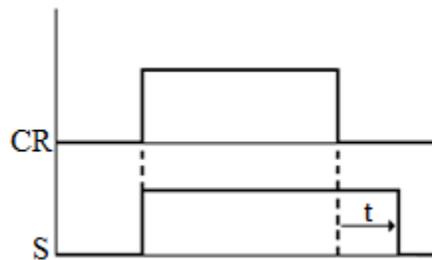


Figura 11.26.: Diagrama de tiempo función temporización a la desexcitación

Ecuación:

$$S = CR^{td}$$

Diagrama Ladder:

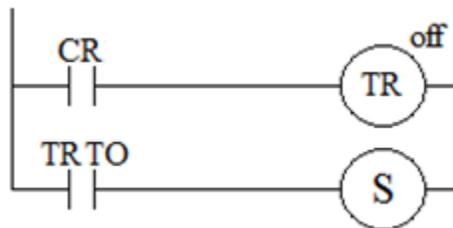


Figura 11.27.: Diagrama Ladder función temporización a la desexcitación

El circuito Ladder contiene un pulsador, que al ser accionado energiza una bobina que cierra un contacto *NA*, permitiendo el flujo de corriente energizando la bobina de temporización *TR OFF*, el contacto *TR TO* asociado a esta bobina funciona como un contacto instantáneo, haciendo que se presente la señal de salida. La función muestra su comportamiento una vez se deje de presionar el pulsador, es decir, que la bobina de temporización este desenergizada, es entonces, cuando actúa contando un tiempo *t* programado para desenergizar la señal de salida.

Diagrama de bloque:

El diagrama de bloque mostrado en la figura 11.28 representa el diagrama Ladder de la función temporización a la desexcitación.

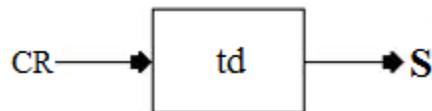


Figura 11.28.: Bloque función temporización a la desexcitación

### 11.2.10. Función temporización a la desexcitación negada

La salida es una señal que cambia del estado lógico “1” al estado lógico “0” una vez la señal de mando haga su aparición, cuando la entrada es desenergizada, la salida se retarda un tiempo *t* para hacer el cambio del estado lógico “0” al estado lógico “1”.

Diagrama de tiempo:

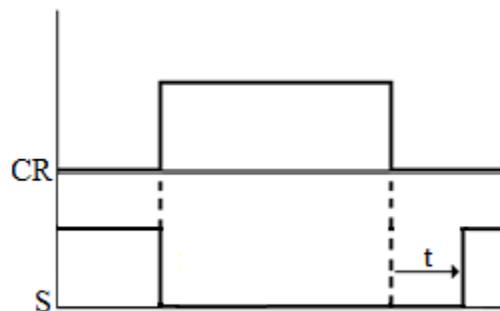


Figura 11.29.: Diagrama de tiempo función temporización a la desexcitación negada

Ecuación:

$$S = \overline{CR^{td}}$$

Diagrama Ladder:

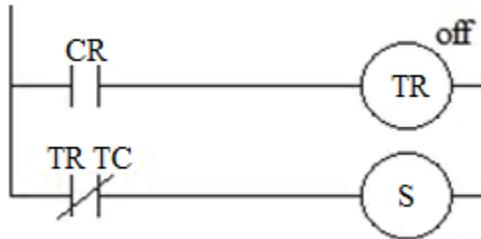


Figura 11.30.: Diagrama Ladder función temporización a la desexcitación negativa

El circuito Ladder contiene un pulsador, que al ser presionado energiza un contacto *NA* energizando la bobina de temporización. El contacto *TR TC* asociado a ésta bobina, actúa de forma instantánea desenergizando la señal de salida. El comportamiento de esta función se presenta cuando se deja de accionar el pulsador y la bobina temporizada se desenergiza, ahí es cuando, transcurrido el tiempo *t* programado, el contacto *TR TC* se cierra y se presenta nuevamente la señal de salida.

Diagrama de bloque:

El diagrama de bloque mostrado en la figura 11.31 representa el diagrama Ladder de la función temporización a la desexcitación negativa.

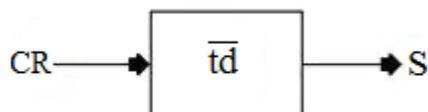


Figura 11.31.: Bloque función temporización a la desexcitación negativa

### 11.2.11. Función detector flancos de subida

La salida es una señal que cambia del estado lógico “0” al estado lógico “1” cuando se presenta la señal de mando, independientemente si la entrada permanece energizada o no, la salida sólo permanece energizada un lapso de tiempo muy corto antes de que vuelva a su estado original.

Diagrama de tiempo:

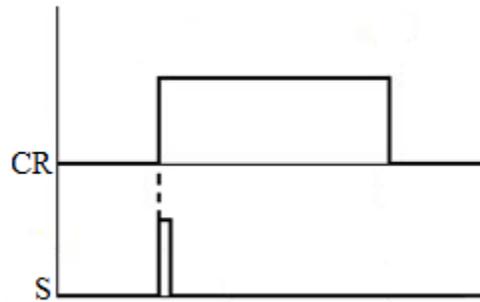


Figura 11.32.: Diagrama de tiempo función detector flancos de subida

Ecuación:

$$S = CR^{\uparrow}$$

Diagrama Ladder:

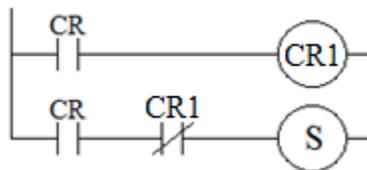


Figura 11.33.: Diagrama Ladder función detector flancos de subida

El circuito ladder contiene un pulsador, que al ser presionado energiza una bobina cerrando un contacto *NA*, en esta parte del circuito es cuando se presenta el flanco de subida, ya que, al cerrarse el contacto, da paso a la energización de otra bobina que abre casi de forma simultánea un contacto *NC*, es en ese momento donde los tiempos de conmutación entre un contacto y otro son muy cortos, por lo que la señal de salida sólo está presente un lapso de tiempo *t* mínimo.

Diagrama de bloque:

El diagrama de bloque mostrado en la figura 11.34 representa el diagrama Ladder de la función detector de flancos de subida.



Figura 11.34.: Bloque función detector flancos de subida

### 11.2.12. Función detector flancos de bajada

Cuando la señal de entrada es energizada, la salida permanece en estado lógico “0” y sólo cambia al estado lógico “1” cuando la entrada es desenergizada, pero sólo se presenta durante un lapso de tiempo muy corto antes de que vuelva a su estado original.

Diagrama de tiempo:

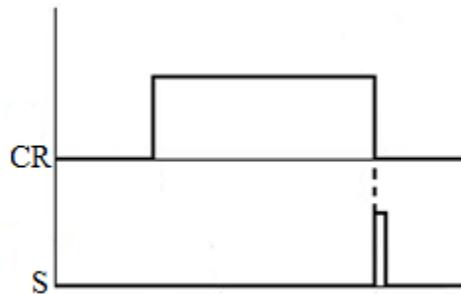


Figura 11.35.: Diagrama de tiempo función detector flancos de bajada

Ecuación:

$$S = CR^{\downarrow}$$

Diagrama Ladder:

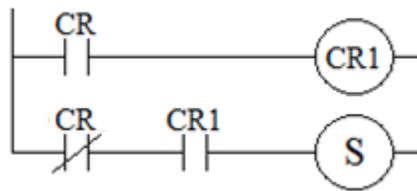


Figura 11.36.: Diagrama Ladder función detector flancos de bajada

El funcionamiento del circuito ladder es similar al circuito ladder del detector de flancos de subida, la diferencia, es que este efectúa el flanco cuando se suelta el pulsador, de igual manera los tiempos de conmutación entre un contacto y otro son muy pequeños haciendo que el pulso de la señal de salida se presente un tiempo  $t$  mínimo.

Diagrama de bloque:

El diagrama de bloque mostrado en la figura 11.37 representa el diagrama Ladder de la función detector flancos de bajada.



Figura 11.37.: Bloque función detector flancos de bajada

### 11.2.13. Función refresco

La salida es una señal que hace su aparición una vez lo haga la señal de mando y cuando es desenergizada la entrada, la salida se tarda un tiempo  $t$  en hacer el cambio de estado lógico "1" al estado lógico "0", pero si la entrada presenta un estado lógico alto antes de que transcurra el tiempo  $t$  para que la salida vuelva a su estado original, el conteo del tiempo  $t$  se reinicia.

Diagrama de tiempo:

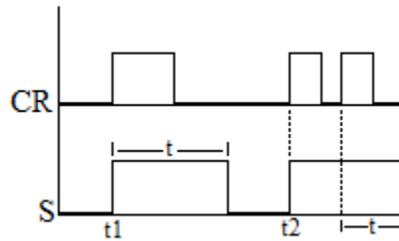


Figura 11.38.: Diagrama de tiempo función refresco

Ecuación:

$$S = CR^R$$

Diagrama Ladder:

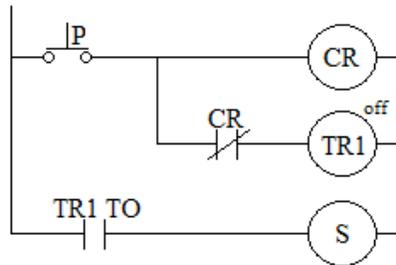


Figura 11.39.: Diagrama Ladder función refresco

El diagrama Ladder contiene un pulsador, que al ser accionado energiza dos bobinas de manera simultánea, una bobina *CR* y un temporizador *TR OFF*, cuando se energiza la bobina *CR* el contacto que está asociado a esta bobina se abre, haciendo que el tiempo programado en el temporizador empiece a transcurrir. Previamente el contacto asociado al temporizador se ha cerrado, por lo que la salida ha pasado a tener un estado alto. Si se acciona de nuevo el pulsador antes de que transcurra el tiempo *t* para que la salida se desenergice, el temporizador comienza de nuevo su conteo.

Diagrama de bloque:

El diagrama de bloque mostrado en la figura 11.40 representa el diagrama Ladder de la función refresco.

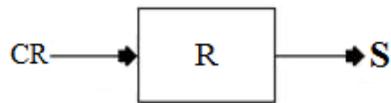


Figura 11.40.: Bloque función refresco

### 11.2.14. Función relé de enclavamiento

La salida es una señal que hace el cambio del estado lógico “0” al estado lógico “1” cuando la primera señal de entrada es energizada, la señal de salida se mantiene en ese estado hasta que una segunda señal de entrada se presente, es en ese momento cuando la salida se desenergiza pasando del estado lógico “1” al estado lógico “0”.

Diagrama de tiempo:

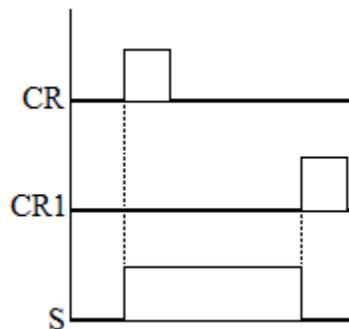


Figura 11.41.: Diagrama de tiempo función relé de enclavamiento

Ecuación:

$$S = CR \bullet \overline{CR1}$$

Diagrama Ladder:

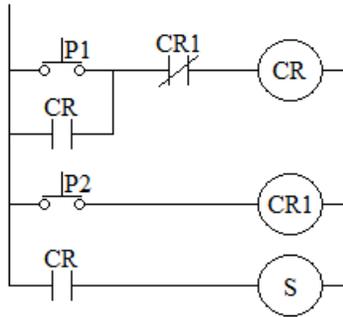


Figura 11.42.: Diagrama Ladder función relé de enclavamiento

El circuito Ladder contiene dos pulsadores, un pulsador *P1* encargado de energizar una bobina principal *CR*, que se mantiene en ese estado gracias a su contacto de sello, la cual a su vez permite la energización de la salida. Cuando se quiere desenergizar la salida es necesario presionar el pulsador *P2*, ya que este energiza la bobina *CR1* que es la encargada de desenergizar la bobina principal.

Diagrama de bloque:

El diagrama de bloque mostrado en la figura 11.43 representa el diagrama Ladder de la función relé de enclavamiento.

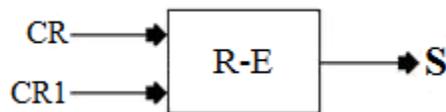


Figura 11.43.: Diagrama de bloque función relé de enclavamiento

### 11.2.15. Función relé de auto-retención

La salida es una señal que se activa siempre y cuando se presente una señal de mando que produzca el cambio de un estado lógico “0” a un estado lógico “1” además de que la señal de bloqueo se encuentre desactivada. La señal de bloqueo sólo interrumpe el cambio de un estado bajo a un estado alto en la señal de salida, por tal motivo, al momento de querer desactivar la salida se hace necesaria la utilización de otra señal de mando.

Diagrama de tiempo:

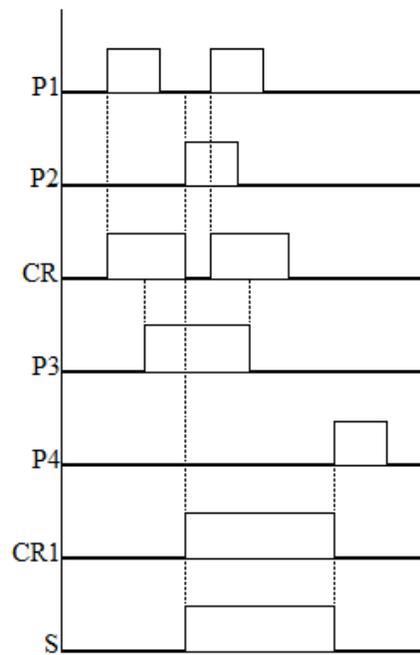


Figura 11.44.: Diagrama de tiempo función relé de auto-retención

Ecuación:

$$S = (P_3 \bullet \overline{(P_1 + CR) \bullet P_2}) + CR1 \bullet \overline{P_4}$$

Diagrama Ladder:

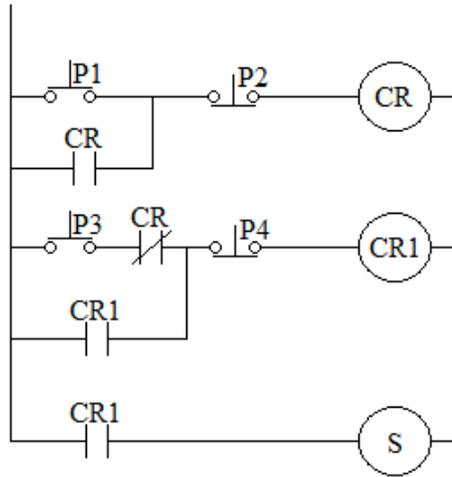


Figura 11.45.: Diagrama Ladder función relé de auto-retención

El circuito Ladder contiene dos pulsadores  $P1$  y  $P2$  que controlan la activación y desactivación de la bobina de bloqueo  $CR$ , y otros dos pulsadores  $P3$  y  $P4$  que controlan la energización y desenergización de la bobina  $CR1$ . Cuando se acciona  $P1$ , se activa la bobina  $CR$  cerrando su contacto de sello y abriendo el contacto  $NC$  asociado a esta bobina, en ese momento, aunque se presione  $P3$  la bobina  $CR1$  no se activa por lo que la salida  $S$  no se presenta, cuando se presiona  $P2$  la bobina  $CR$  se desenergiza y sus contactos vuelven a su estado original. Si se presiona  $P3$  de nuevo se activa la bobina  $CR1$  cerrando sus contactos asociados  $NA$  y permitiendo que la salida  $S$  pase a un estado lógico alto, y se mantiene en ese estado hasta que se presione  $P4$ , sin importar que la bobina  $CR$  se energice nuevamente y abra el contacto  $NC$  asociado a ella.

Diagrama de bloque:

El diagrama de bloque mostrado en la figura 11.46 representa el diagrama Ladder de la función relé de auto-retención.



Figura 11.46.: Diagrama de bloque función relé de auto-retención

## 11.3. FUNCIONES DERIVADAS

### 11.3.1. Introducción

Las funciones derivadas tienen un uso importante dentro de la metodología para el diseño de automatismos, ya que, son funciones que se obtienen a partir de la combinación de las funciones básicas. El desarrollo de un diseño solamente por medio de funciones básicas resulta ser, en ocasiones, un poco extenso, mientras que con las funciones derivadas puede resumirse un poco. A continuación se exponen las funciones derivadas.

### 11.3.2. Función temporización a la excitación y desexcitación

La salida es una señal que hace el cambio del estado lógico “0” al estado lógico “1” un tiempo  $t$  después de que la señal de mando lo haya hecho, de igual manera cuando la entrada se desenergice la salida se retarda un tiempo  $t$  en hacer el cambio del estado lógico “1” al estado lógico “0”.

Diagrama de tiempo:

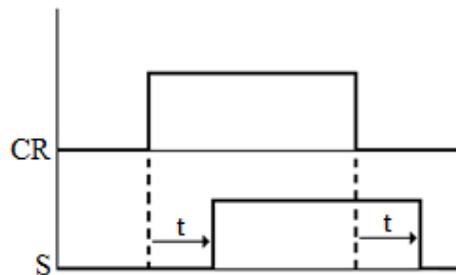


Figura 11.47.: Diagrama de tiempo función temporización a la excitación y desexcitación

Ecuación:

$$S = CR^{ted}$$

Diagrama Ladder:

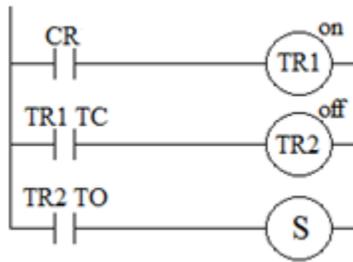


Figura 11.48.: Diagrama Ladder función temporización a la excitación y desexcitación

El circuito ladder contiene un pulsador, que al ser presionado energiza una bobina cerrando un contacto *NA*, este contacto permite el paso de corriente energizando la bobina temporizada *TR ON*, que después de un *t* programado cierra el contacto *TR1 TC* energizando la bobina temporizada *TR OFF*, al estar energizada, el contacto *TR2 TO* actúa de forma instantánea y sólo presenta su retardo una vez la bobina esté desenergizada, es así, como la señal de salida se activa un tiempo *t* después de que se presente la señal de mando y se retarda un tiempo *t* en desactivarse.

Diagrama de bloque:

El diagrama de bloque mostrado en la figura 11.49 representa el diagrama Ladder de la función temporización a la excitación y desexcitación.

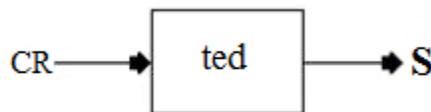


Figura 11.49.: Bloque función temporización a la excitación y desexcitación

### 11.3.3. Función temporización a la excitación y desexcitación negada

La salida es una señal que hace el cambio del estado lógico “1” al estado lógico “0” un tiempo *t* después de que la señal de mando lo haya hecho, de igual manera cuando la entrada se desenergice la salida se retarda un tiempo *t* en hacer el cambio del estado lógico “0” al estado lógico “1”.

Diagrama de tiempo:

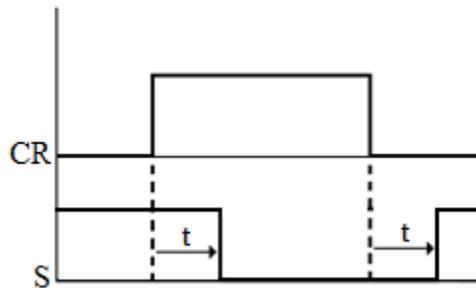


Figura 11.50.: Diagrama de tiempo función temporización a la excitación y desexcitación negada

Ecuación:

$$S = \overline{CR^{ted}}$$

Diagrama Ladder:

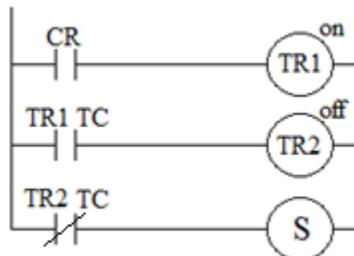


Figura 11.51.: Diagrama Ladder función temporización a la excitación y desexcitación negada

El circuito Ladder contiene un pulsador, que al ser accionado energiza una bobina la cual cierra un contacto *NA*, este contacto energiza la bobina temporizada *TR ON*, que cierra un contacto *TRI TC* después de transcurrir un tiempo *t*, este contacto energiza la bobina *TR OFF* haciendo que el contacto asociado a esta bobina actúe de forma instantánea, abriéndose y desenergizando la señal de salida. En otras palabras la señal de salida es desactivada un tiempo *t* cuando se presiona el pulsador, y permanece en ese estado hasta que se suelta el pulsador, es en ese momento cuando transcurre un tiempo *t* antes de que la señal de salida vuelva a activarse.

Diagrama de bloque:

El diagrama de bloque mostrado en la figura 11.52 representa el diagrama Ladder de la función temporización a la excitación y desexcitación negada.

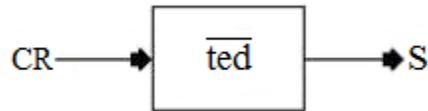


Figura 11.52.: Bloque función temporización a la excitación y desexcitación negada

#### 11.3.4. Función detector de flancos de subida y bajada

La salida es una señal que hace el cambio del estado lógico “0” al estado lógico “1” cuando la señal de mando hace su aparición, pero la salida se mantiene energizada un lapso de tiempo muy corto antes de que vuelva a su posición original, así la señal de entrada permanezca energizada, y sólo vuelve la salida a tener éste comportamiento una vez la señal de entrada se desenergice.

Diagrama de tiempo:

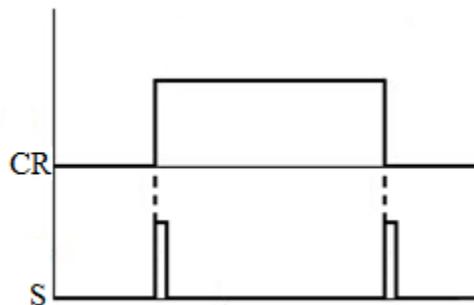


Figura 11.53.: Diagrama de tiempo función detector de flancos de subida y bajada

Ecuación:

$$S = CR^{\uparrow\downarrow}$$

Diagrama Ladder:

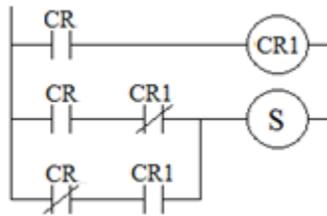


Figura 11.54.: Diagrama Ladder función detector de flancos de subida y bajada

El circuito Ladder contiene un pulsador, que al ser presionado energiza una bobina, esta cierra un contacto *NA*. En el momento que sucede esto se presenta el flanco de subida activando otra bobina y abriendo un contacto *NC*, en el momento que se suelta el pulsador, se desenergiza la bobina y los contactos vuelven a su posición original, es en ese momento cuando se presenta un flanco de bajada. Los tiempos de conmutación y los pulsos tanto para flancos de subida como de bajada tienen una duración mínima.

Diagrama de bloque:

El diagrama de bloque mostrado en la figura 11.55 representa el diagrama Ladder de la función detector flancos de subida y bajada.

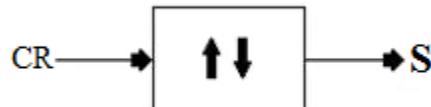


Figura 11.55.: Bloque función detector de flancos de subida y bajada

### 11.3.5. Función toggle

La salida es una señal que cambia de estado lógico “0” al estado lógico “1” cuando la señal de mando sólo efectúe un flanco de subida y debe permanecer en ese estado hasta que la entrada vuelva a ser energizada, es ahí, cuando la salida cambia del estado lógico “1” al estado lógico “0”, y vuelve a energizarse ante una nueva energización de la señal de entrada.

Diagrama de tiempo:

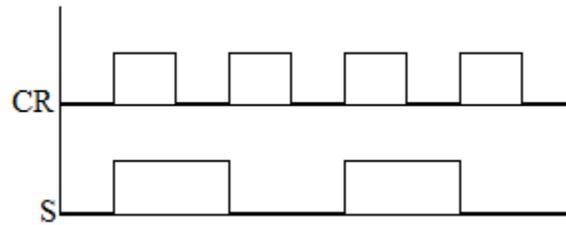


Figura 11.56.: Diagrama de tiempo función toggle

Ecuación:

$$S = CR^T$$

Diagrama Ladder:

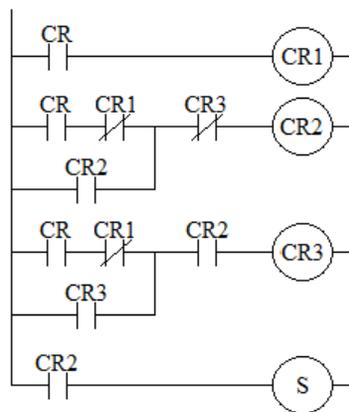


Figura 11.57.: Diagrama Ladder función toggle

El circuito Ladder de la función toggle funciona con un sólo pulsador ó elemento de mando, y consiste en hacer que una carga cambie de estado lógico, ya sea de alto a bajo ó de bajo a alto cada vez que detecta un flanco de subida.

Diagrama de bloque:

El diagrama de bloque mostrado en la figura 11.58 representa el diagrama Ladder de la función toggle.

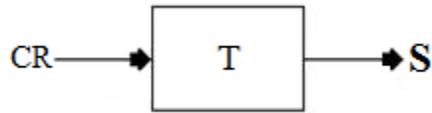


Figura 11.58.: Bloque función toggle

### 11.3.6. Función contador flancos de subida

La cantidad de veces que se desea contar cuando se presenta un flanco de subida, es el número de salidas que se van a obtener en el diagrama de tiempo.

La salida es una señal que hace el cambio del estado lógico “0” al estado lógico “1” una vez la señal de mando se energice.

Diagrama de tiempo:

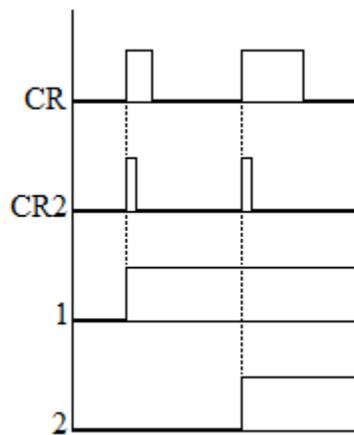


Figura 11.59.: Diagrama de tiempo función contador flancos de subida

Ecuación:

$$S = CR^{\#C\uparrow}$$

Diagrama Ladder:

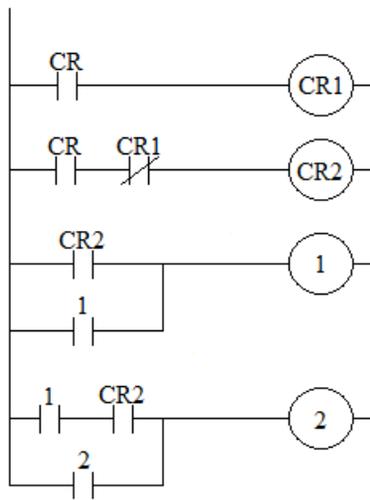


Figura 11.60.: Diagrama Ladder función contador flancos de subida

El circuito Ladder cuenta la veces que se presiona un pulsador por medio del conteo de los de flancos de subida.

Diagrama de bloque:

El diagrama de bloque mostrado en la figura 11.61 representa el diagrama Ladder de la función contador flancos de subida.



Figura 11.61.: Bloque función contador flancos de subida

### 11.3.7. Función contador flancos de bajada

La cantidad de veces que se desea contar cuando se presenta un flanco de bajada, es el número de salidas que se van a obtener en el diagrama de tiempo.

La salida es una señal que hace el cambio del estado lógico “0” al estado lógico “1” una vez la señal de mando se desenergice.

Diagrama de tiempo:

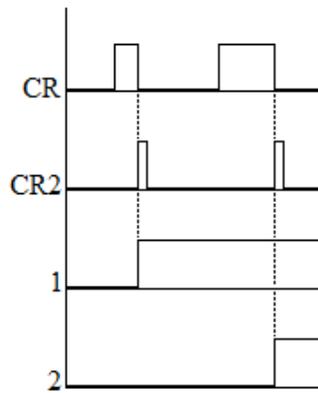


Figura 11.62.: Diagrama de tiempo función contador flancos de bajada

Ecuación:

$$S = CR \# C\downarrow$$

Diagrama Ladder:

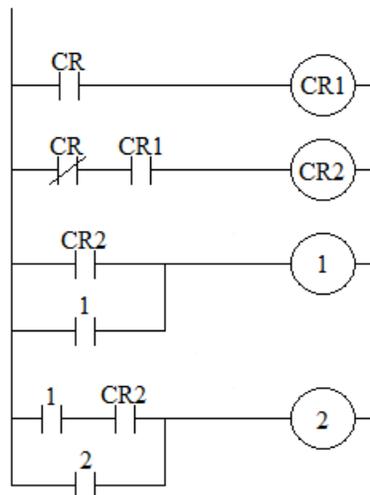


Figura 11.63.: Diagrama Ladder función contador flancos de bajada

El circuito ladder cuenta las veces que se suelta un pulsador a través del conteo de flancos de bajada.

El diagrama de bloque mostrado en la figura 11.64 representa el diagrama Ladder de la función contador flancos de bajada.

Diagrama de bloque:



Figura 11.64.: Diagrama de bloque función contador flancos de bajada

### 11.3.8. Función contador flancos de subida y bajada

En esta función, el número de salidas está determinado por la cantidad de veces que se desea contar tanto los flancos de subida como los flancos de bajada.

Una señal de salida hace el cambio de estado lógico “0” al estado lógico “1” cuando la señal de mando es activada, y otra señal de salida efectúa el mismo cambio de estado lógico, cuando la señal de mando es desactivada.

Diagrama de tiempo:

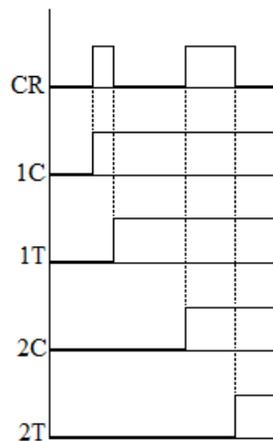


Figura 11.65.: Diagrama de tiempo función contador flancos de subida y bajada

Ecuación:

$$S = CR \# C \uparrow \downarrow$$

Diagrama Ladder:

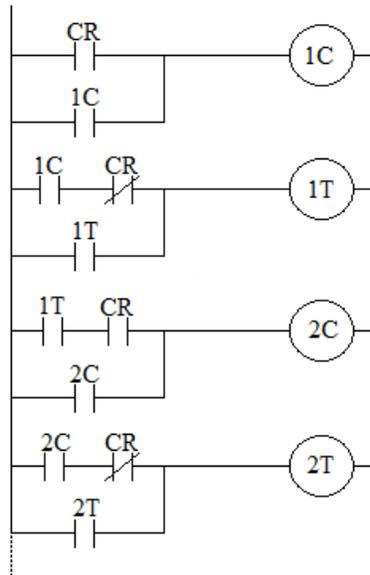


Figura 11.66.: Diagrama Ladder función contador flancos de subida y bajada

El circuito Ladder cuenta las veces que se presiona un pulsador y las veces que se suelta, haciendo uso del conteo de flancos de subida y flancos de bajada.

Diagrama de bloque:

El diagrama de bloque mostrado en la figura 11.67 representa el diagrama Ladder de la función contador flancos de subida y bajada.



Figura 11.67.: Diagrama de bloque función contador flancos de subida y bajada

## 11.4. PASOS PARA DISEÑAR

- Paso 1: Inicialmente se identifica en el enunciado del problema, el número de entradas y salidas del automatismo.

- Paso 2: Plantear ecuaciones de salida (en el caso que se tengan varias) de la forma  $S(f_1, f_2, \dots, f_n)$ , donde  $f_1, f_2, \dots, f_n$  representan las señales intermedias de mando.
- Paso 3: Escoger el método más adecuado de simplificación de funciones, ya sea, por el método algebraico o mapas de Karnaugh cuando se tiene sólo una función, o minimización por Quine-McCluskey cuando son varias funciones, de tal manera que se obtenga una ecuación simplificada.
- Paso 4: Realizar un diagrama de bloques correspondiente a cada ecuación obtenida en el Paso 3 con las funciones expuestas en la metodología de diseño por señales de mando.
- Paso 5: Una vez se tenga el diagrama de bloques del automatismo, se procede a realizar el diagrama ladder correspondiente a cada bloque.
- Paso 6: Simplificar el diagrama ladder resultante del Paso 5.

## 11.5. EJEMPLOS BÁSICOS

### 11.5.1. Ejemplo 1.

Se desea realizar el encendido de una lámpara ubicada en un pasillo con tres entradas de tal forma que:

Los interruptores P1, P3, P5 enciendan la lámpara y los pulsadores P2, P4, P6 la apaguen.

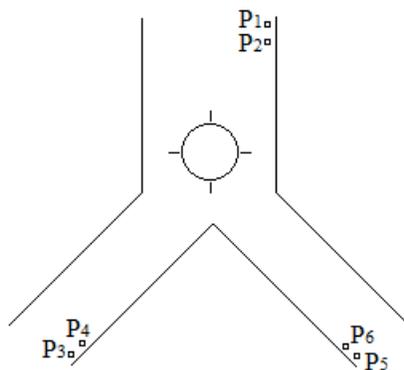


Figura 11.68.: Ejemplo 1

**Solución:**

Según las condiciones expuestas en el enunciado, se identifican seis entradas, correspondientes a los pulsadores de encendido y los pulsadores de apagado, y una única salida representada por la lámpara. Una ecuación que indica el encendido y apagado de la lámpara puede obtenerse por el método de Quine-McCluskey, pero bien se sabe, que este método es para la solución de más de una función y en este caso, sólo se va a obtener una sola ecuación. Una solución sencilla son los mapas de Karnaugh, pero se hace complicado dibujar una tabla de verdad para seis entradas, ya que se obtienen sesenta y cuatro posibles combinaciones, pero se puede solucionar realizando dos tablas de verdad, una para el encendido de la lámpara y otra para el apagado, así se facilita la solución del problema. Si se observa, tanto para el encendido como para el apagado de la lámpara se tienen tres entradas, lo que resulta en ocho posibles combinaciones, lo que hace el método de mapas de Karnaugh una forma sencilla de dar solución a este problema.

En vista de lo que el ejercicio requiere, la lámpara debe tener una función de encendido y una función de apagado, por tal razón se deduce que la ecuación general podría tener la forma  $S = f_1 \bullet \overline{f_2}$  siendo  $f_1$  la función de encendido y  $\overline{f_2}$  la función de apagado. Así pues, una vez se haya pensado el problema y se decide el método para dar solución se procede a realizar la tabla de verdad y el mapa de Karnaugh para  $f_1$  y  $\overline{f_2}$  respectivamente.

Para la función del encendido de la lámpara ( $f_1$ ) se plantea la tabla de verdad mostrada en la tabla 11.1.

Tabla 11.1.: Tabla de verdad encendido de la lámpara ejemplo 1

Entradas			Salida
P1	P3	P5	F1
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	1
1	1	1	1

Luego de obtener la tabla de verdad para la función de encendido se plantea el mapa de Karnaugh que se muestra en la figura 11.69.

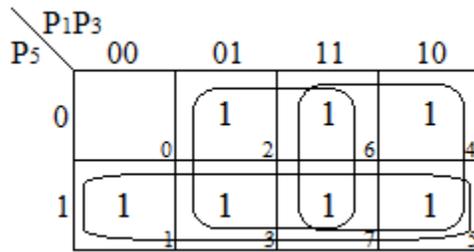


Figura 11.69.: Mapa de Karnaugh encendido de la lámpara ejemplo 1

A partir del mapa de Karnaugh anterior se obtiene una ecuación para  $f_1$  que representa el encendido de la lámpara, de manera que:

$$f_1 = P1 + P3 + P5$$

Se procede de la misma manera que el caso anterior para obtener la ecuación  $\overline{f_2}$  que representa el apagado de la lámpara, es por esto que se construye una tabla de verdad como la que se observa en la tabla 11.2 y un mapa de Karnaugh como el de la figura 11.70.

Tabla 11.2.: Tabla de verdad apagado de la lámpara ejemplo 1

Entradas			Salida
P2	P4	P6	F2
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	1
1	1	1	1

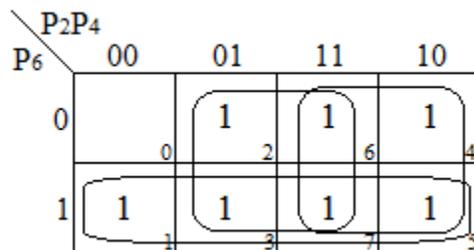


Figura 11.70.: Mapa de Karnaugh apagado de la lámpara ejemplo 1

A partir del mapa de Karnaugh anterior se obtiene la siguiente ecuación para  $f_2$ :

$$f_2 = P2 + P4 + P6$$

En este caso como se requiere  $\overline{f_2}$ , se realiza un proceso conocido como negación en ambos lados de la ecuación  $f_2$  y se aplica el teorema de Demorgan 5.3, con lo que se obtiene:

$$\overline{f_2} = \overline{P2} \bullet \overline{P4} \bullet \overline{P6}$$

Una vez se tienen las ecuaciones correspondientes al encendido y el apagado de la lámpara, se reemplazan en la ecuación general  $S$  mencionada anteriormente, con la cual se puede aplicar la metodología de diseño, de manera que se construye el diagrama de bloques de la figura 11.71 y posteriormente el diagrama Ladder de la figura 11.72.

$$S = (f_1) \bullet \overline{f_2}$$

$$S = (P1 + P3 + P5) \bullet \overline{P2} \bullet \overline{P4} \bullet \overline{P6}$$

**Diagrama de bloques:**

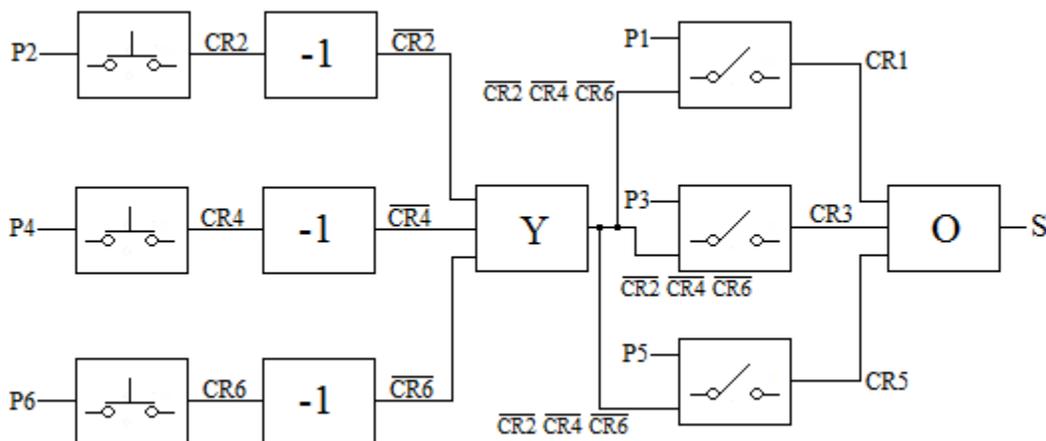


Figura 11.71.: Diagrama de bloques ejemplo 1

Diagrama Ladder de cada bloque:

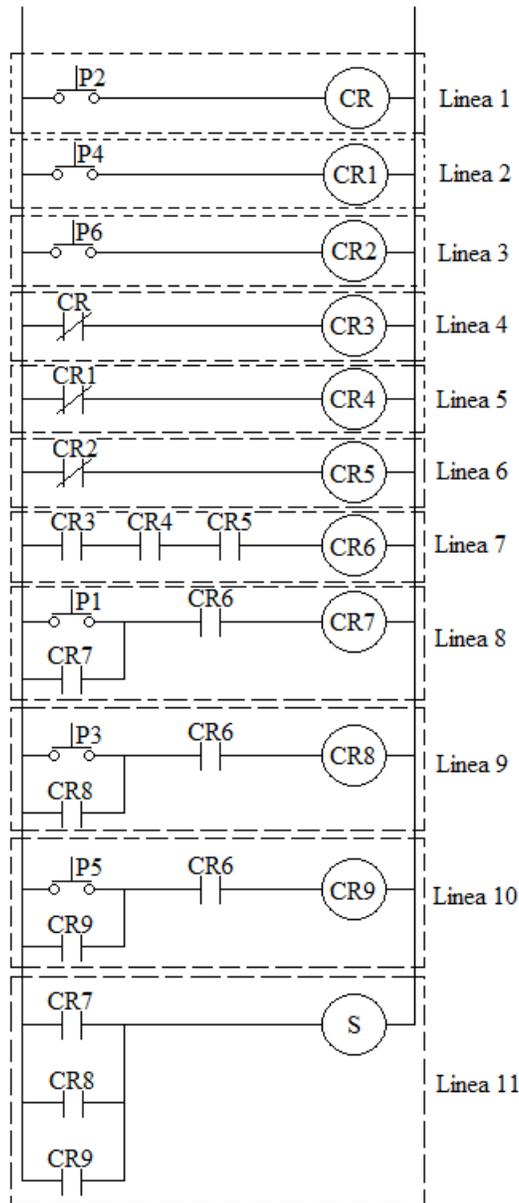


Figura 11.72.: Diagrama Ladder de cada bloque ejemplo 1

A continuación se procede a simplificar el circuito ladder de la figura 11.72 a partir de las ecuaciones representativas de cada línea del circuito escalera.

Línea 1  $P_2 = CR$

$$\text{Línea 2 } P_4 = CR1$$

$$\text{Línea 3 } P_6 = CR2$$

$$\text{Línea 4 } \overline{CR} = CR3$$

$$\text{Línea 5 } \overline{CR1} = CR4$$

$$\text{Línea 6 } \overline{CR2} = CR5$$

$$\text{Línea 7 } CR6 = CR3 \bullet CR4 \bullet CR5$$

$$\text{Línea 8 } CR7 = (P_1 + CR7) \bullet CR6$$

$$\text{Línea 9 } CR8 = (P_3 + CR8) \bullet CR6$$

$$\text{Línea 10 } CR9 = (P_5 + CR9) \bullet CR6$$

$$\text{Línea 11 } S = CR7 + CR8 + CR9$$

Se puede reemplazar en la ecuación de la línea 7 las ecuaciones correspondientes a las líneas 4, 5 y 6 ya que estas no se usan en ninguna otra parte, obteniendo una nueva ecuación  $CR6 = \overline{CR} \bullet \overline{CR1} \bullet \overline{CR2}$ .

En las ecuaciones de las líneas 8, 9, 10 puede reemplazarse la nueva ecuación de la línea 7, por lo que pasan a ser:

$$CR7 = (P_1 + CR7) \bullet \overline{CR} \bullet \overline{CR1} \bullet \overline{CR2}$$

$$CR8 = (P_3 + CR8) \bullet \overline{CR} \bullet \overline{CR1} \bullet \overline{CR2}$$

$$CR9 = (P_5 + CR9) \bullet \overline{CR} \bullet \overline{CR1} \bullet \overline{CR2}$$

Con las simplificaciones mostradas anteriormente, se logra reducir el diagrama Ladder en 4 líneas, por lo que se obtiene un nuevo diagrama Ladder que se muestra en la figura [11.73](#).

**Diagrama Ladder simplificado:**

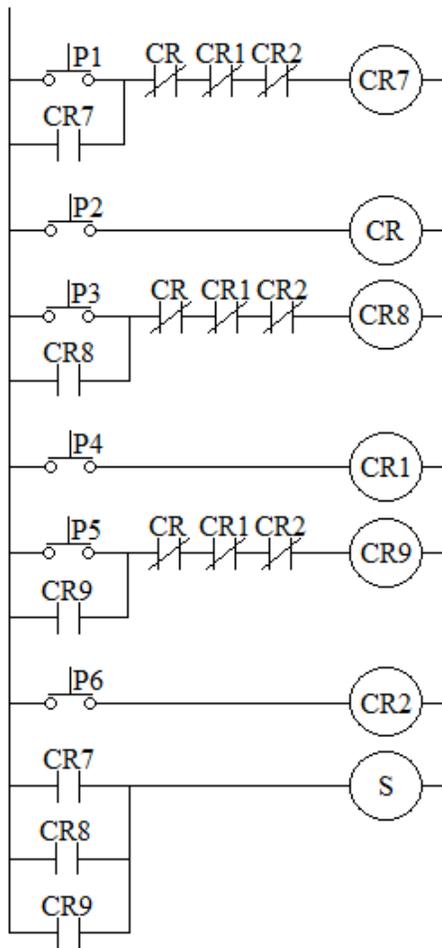


Figura 11.73.: Diagrama Ladder simplificado ejemplo 1

**11.5.2. Ejemplo 2.**

Adicional al requerimiento del ejemplo anterior en este caso se necesita que la lámpara permanezca encendida un tiempo  $t$  después del accionamiento de uno de los pulsadores de apagado.

**Solución:**

Como el ejemplo 2 tiene exactamente las mismas variables de entrada que el ejemplo 1 y aprovechando que ya se tiene una solución completa del ejemplo anterior, se toma como referencia dicha solución, la única diferencia que tiene este ejemplo es el apagado de la lámpara no se realiza de forma instantánea una vez se haya accionado cualquiera de los pulsadores de apagado, sino que, suceda luego de que trascorra un tiempo  $t$  determinado, para lograr este efecto se hace necesario un bloque adicional que corresponde a la función *temporización a la desexcitación*( $td$ ). En ese orden de ideas se procede a realizar la ecuación general que describe dicho comportamiento sugerido por el ejemplo 2.

$$S = [f_1 \bullet \overline{f_2}]^{td}$$

Como las funciones  $f_1 = P1 + P3 + P5$  y  $\overline{f_2} = \overline{P2} \bullet \overline{P4} \bullet \overline{P6}$  se desarrollaron en el ejemplo anterior, la ecuación que representa la solución para el ejemplo 2 es la siguiente:

$$S = [(P1 + P3 + P5) \bullet \overline{P2} \bullet \overline{P4} \bullet \overline{P6}]^{td}$$

A partir de la ecuación anterior se obtiene el diagrama de bloques mostrado en la figura 11.74 y el diagrama Ladder que se observa en la figura 11.75.

**Diagrama de bloques:**

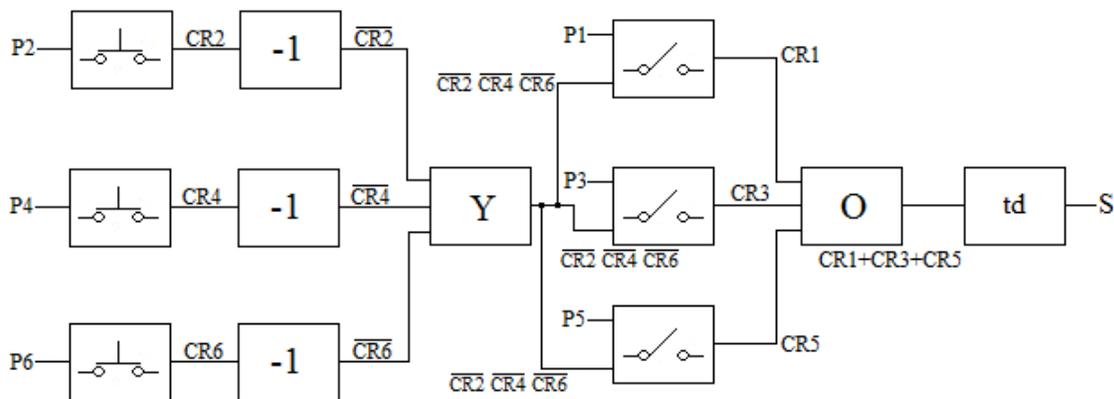


Figura 11.74.: Diagrama de bloques ejemplo 2

Diagrama Ladder de cada bloque:

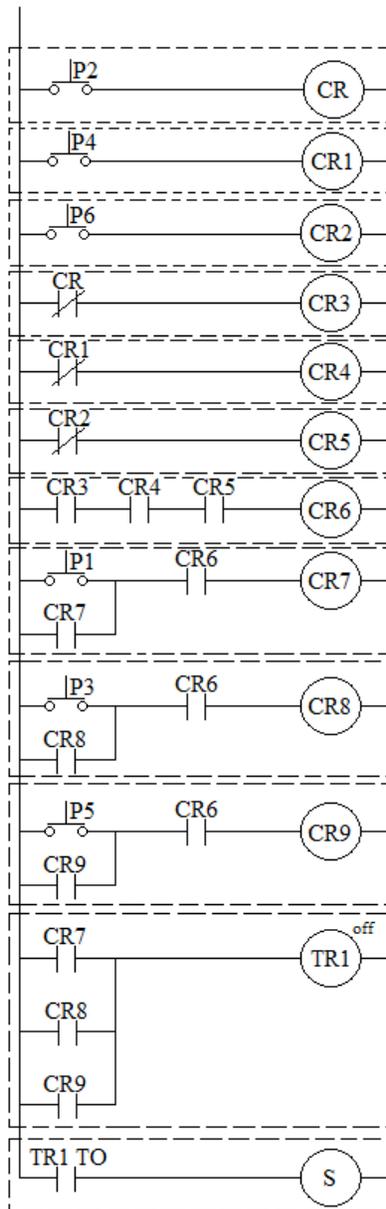


Figura 11.75.: Diagrama Ladder de cada bloque ejemplo 2

Para la simplificación de este circuito, se realiza un proceso similar al desarrollado en el ejemplo 1. De hecho el ejemplo 2 posee casi las mismas condiciones para el encendido y apagado de la lámpara, salvo que en este caso, la lámpara permanece encendida por un tiempo  $t$  una vez se presione uno de los pulsadores de apagado, por tal razón sólo existe una variante, que consiste en un temporizador a la desexcitación. En ese orden de ideas se obtiene el diagrama Ladder de la figura 11.76.

**Diagrama Ladder simplificado:**

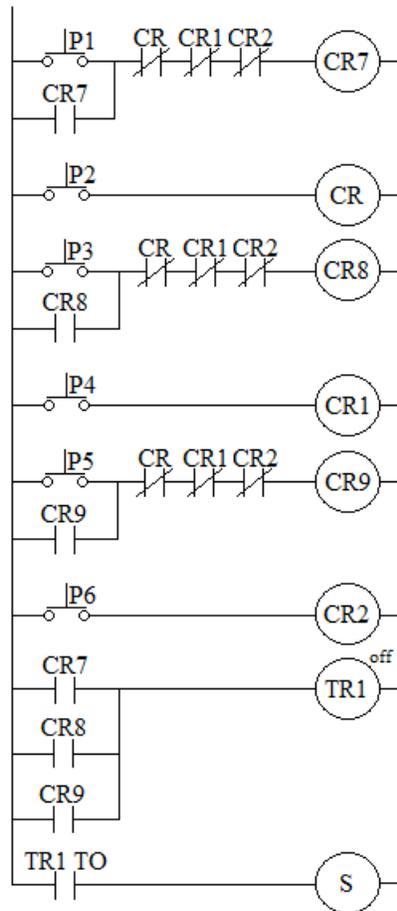


Figura 11.76.: Diagrama Ladder simplificado ejemplo 2

**11.5.3. Ejemplo 3.**

Realizando una variante al ejercicio del pasillo con la lámpara, ahora se quiere realizar el encendido de la lámpara a través de los pulsadores P1, P3, P5, y el apagado de manera automática luego de que trascurra un tiempo  $t$  programado.

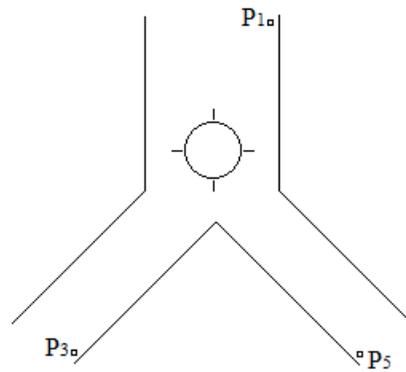


Figura 11.77.: Ejemplo 3

**Solución**

Este ejemplo presenta las mismas variables de entrada correspondiente al encendido de la lámpara de los ejemplos anteriores, mientras que el apagado se realiza con la ayuda de un bloque de temporización a la desexcitación, por lo tanto, tomando la ecuación de salida del ejemplo 2 y suprimiendo la función de apagado se obtiene lo siguiente:

$$S = [f_1]^{td}$$

La ecuación anterior representa la solución del ejemplo 3, en la que  $f_1 = P1 + P3 + P5$ , por lo que se obtiene:

$$S = [P1 + P3 + P5]^{td}$$

Al igual que en los ejemplos anteriores a partir de la ecuación de salida se dibuja el diagrama de bloques equivalente que se observa en la figura 11.78 y el diagrama Ladder mostrado en la figura 11.79.

**Diagrama de bloques**

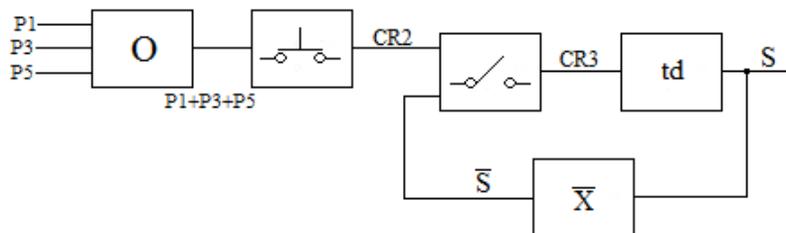


Figura 11.78.: Diagrama de bloques ejemplo 3

**Diagrama Ladder de cada bloque**

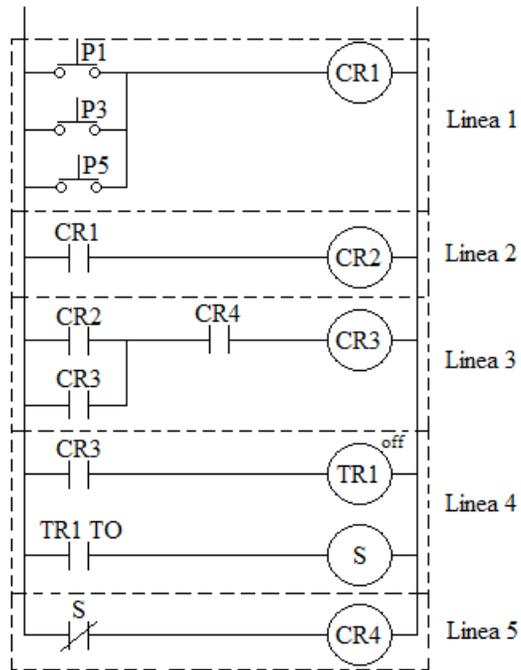


Figura 11.79.: Diagrama Ladder de cada bloque ejemplo 3

A continuación se procede a simplificar el circuito ladder de la figura 11.79 a partir de las ecuaciones representativas de cada línea del circuito escalera.

Línea 1  $P1 + P3 + P5 = CR1$

Línea 2  $CR1 = CR2$

Línea 3  $CR3 = (CR2 + CR3) \bullet CR4$

Línea 4  $S = CR3^{td}$

Línea 5  $CR4 = \bar{S}$

Se puede reemplazar la ecuación de la línea 2 en la ecuación de la línea 1 obteniendo una nueva ecuación  $P1 + P3 + P5 = CR2$ .

La ecuación de la línea 5 puede reemplazarse en la ecuación de la línea 3, por lo que la nueva ecuación para la línea 3 es  $CR3 = (CR2 + CR3) \bullet \bar{S}$

Con las simplificaciones mostradas anteriormente, se logra reducir el diagrama ladder en 2 líneas, por lo que se obtiene un nuevo diagrama Ladder que se muestra en la figura 11.80.

**Diagrama Ladder simplificado**

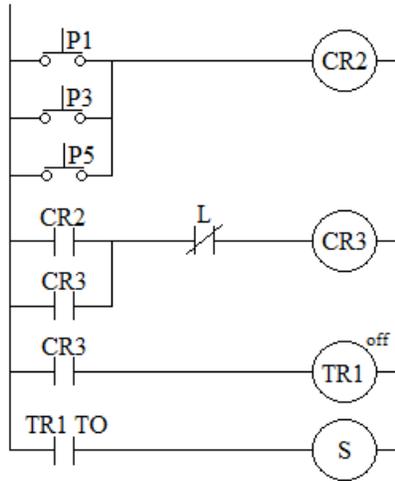


Figura 11.80.: Diagrama Ladder simplificado ejemplo 3



# 12. METODOLOGÍA POR MÁQUINAS DE ESTADO

## 12.1. INTRODUCCIÓN

Esta metodología permite realizar el diseño de circuitos Ladder que representan la solución a problemas de automatización haciendo uso de las máquinas de estado. Esto se logra a partir del hecho que casi todos los problemas con los que se enfrenta la automatización pueden modelarse de tal forma que se obtenga una máquina de estados, y es por eso que a continuación se muestran las equivalencias en Ladder para las partes que la conforman, entre las que se destacan las entradas, los estados, las salidas y la forma de inicializar la máquina.

## 12.2. EQUIVALENCIAS

### 12.2.1. Entradas

Las entradas en una máquina de estados permiten el cambio de estados según sea su valor *uno* o *cero*. Para su representación en el lenguaje Ladder se hace necesaria la utilización de pulsadores y bobinas de tal manera que su energización o desenergización hagan referencia a un cambio en ellas. Como la entrada tiene dos valores, *uno* y *cero* su accionar es similar al de un biestable accionado por dos pulsadores como se muestra en la figura 12.1, en ella puede apreciarse que el pulsador normalmente abierto *P1* energiza la bobina *E*, simulando así que la entrada presenta el valor de *uno* y manteniéndola de este modo gracias a un contacto de sello, hasta que se acciona el pulsador normalmente cerrado *P2* quien la desenergiza de manera que se muestra que la entrada a cambiado su estado a *cero*.

Para probar que el diseño de la figura 12.1 funcione como se requiere, se utiliza una máquina de dos estados *A* y *B* (figura 12.2), en la cual el cambio entre estados se realiza cuando su única entrada *E* presenta el valor de *uno* y permanece en un estado si *E* tiene el valor de *cero*. Al momento de implementar el diseño se llega a determinar que el diagrama Ladder no funciona como se desea ya

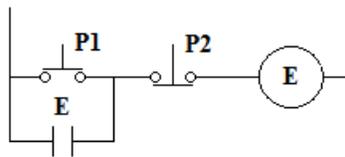


Figura 12.1.: Diagrama Ladder de una entrada mediante la utilización de un biestable

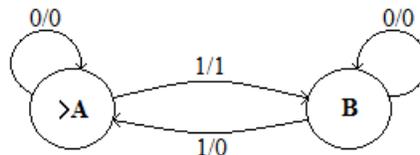


Figura 12.2.: Máquina de dos estados y una entrada

que al accionar el pulsador  $P1$  y mientras se presiona el pulsador  $P2$ , la máquina cambia varias veces de estado, esto se presenta debido a la dificultad de sincronizar la pulsación de estos dos elementos.

Es necesario replantear el diseño inicial de las entradas de manera que la activación de ellas se presente por un instante de tiempo. Una posible solución sería la utilización de flip flops, pero al momento de su implementación en Ladder no es práctico, de manera que esta opción se descarta.

Para la solución de este problema se suprime el pulsador normalmente cerrado  $P2$ , evitando así la necesidad de sincronización de los dos pulsadores y se conserva el pulsador normalmente abierto  $P1$ , identificando en él, su estado de reposo como que la entrada tiene un valor de *cero* y cuando se acciona, es decir, se cierra es porque el valor de la entrada a cambiado a *uno*. Adicional a esto, el pulsador se acompaña de un detector de flancos de subida o de bajada según como quiera realizarse el diseño, lo importante de la utilización del detector de flancos es que no permite que en el caso de que las transiciones de la máquina estén configuradas de la misma manera esta no cambie de estados sin control, sino que para cada cambio de estado se deba accionar el pulsador, indicando de nuevo el valor de las entradas para conseguir que la máquina avance hacia otro estado.

En el caso que la máquina de estados cuente con  $n$  entradas el número de pulsadores a utilizarse debe ser  $n$ , pero al momento de querer comprobar el diseño en un simulador para circuitos Ladder el número de pulsadores estará determinado por  $2^n - 1$ , de esta manera se asigna a cada posibilidad de que una o más entradas tomen el valor de *uno* a un pulsador, solucionando así el problema de no poder accionar dos o más pulsadores de manera simultánea.

En la figura 12.3, una entrada se representa por un pulsador marcado como  $P1$ , que acciona un detector de flancos de subida o bajada, que a su vez energiza una bobina  $PS1$  que dispone de contactos para realizar el cambio de estado según se desee.

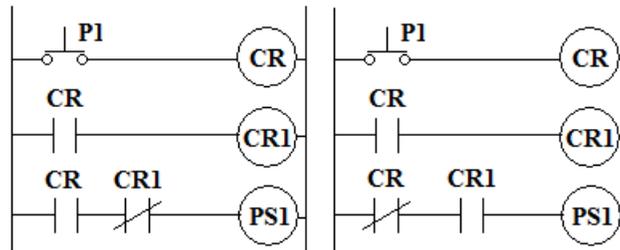


Figura 12.3.: Diagrama Ladder equivalente para una entrada de una máquina de estados

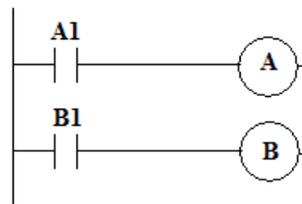


Figura 12.4.: Diagrama Ladder control de estados

### 12.2.2. Estados

Los estados se representan por bobinas marcadas con las primeras letras del alfabeto en mayúsculas  $A, B, C, D, \dots$  según tantos estados como se requiera, su funcionamiento se asemeja al de una memoria, ya que cuenta con una bobina que los activa, un contacto de sello que los mantiene en ese estado y otra bobina que los desactiva, de modo que satisfacen la ecuación  $M = (Set + M) \bullet \overline{Reset}$ .

La activación de un estado se realiza a través de un contacto de una bobina auxiliar (figura 12.4), llamada *bobina de activación* que se nombra como el estado que se quiere activar adicionándole el número 1, es decir,  $A1, B1, C1, D1, \dots$  que se energiza siempre y cuando el estado anterior al que se quiere llegar esté activo y que la combinación de las entradas corresponda a la requerida para la transición (figura 12.5); el estado permanece energizado gracias a un contacto de sello que tiene asociado la bobina de activación.

Para la desactivación de los estados se precisa de otra bobina auxiliar por cada estado, que recibe el nombre de *bobina de desactivación*, se marca de manera similar a la bobina de activación salvo el número, que en este caso es el 2, es decir,  $A2, B2, C2, D2, \dots$  como se muestra en la figura 12.6. Su energización se logra a través de un arreglo de contactos pertenecientes a una transición y a las bobinas de activación. La bobina de desactivación se energiza sólo por un instante, tiempo suficiente para realizar la desenergización del estado deseado. Para tal objetivo se introduce un contacto *NC* en el circuito de la bobina de activación, ya que con la apertura de este se logra desenergizar el estado deseado, cambiando así el circuito de la bobina de activación por el mostrado en la figura 12.7.

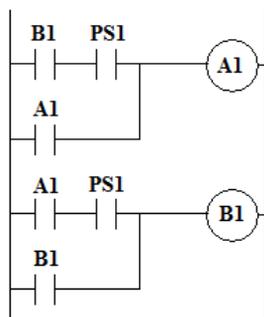


Figura 12.5.: Diagrama Ladder bobina de activación

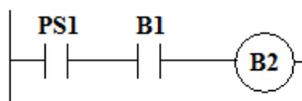


Figura 12.6.: Diagrama Ladder bobina de desactivación

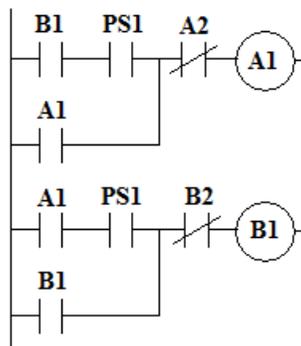


Figura 12.7.: Diagrama Ladder bobina de activación modificado

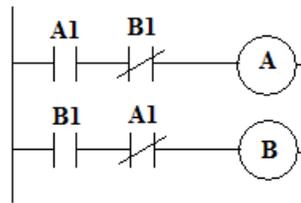


Figura 12.8.: Diagrama ladder control de estados modificado

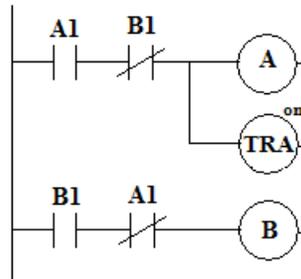


Figura 12.9.: Diagrama Ladder activación estado temporizado

Se garantiza que la máquina de estados se encuentre en un único estado en un instante de tiempo mediante la utilización de contactos *NC* de las bobinas de activación de los estados restantes, de manera que el circuito de control de los estados se modifica como se observa en la figura 12.8.

### 12.2.3. Transiciones Temporizadas

Es común encontrar en problemas de automatización la necesidad de implementar diseños en los que se incluyan dispositivos que controlen tiempos de operación, ya sea porque se realicen procesos repetitivos, porque se desee reducir la interacción de un operario, entre otras razones. Teniendo en cuenta lo anterior es como se concibe que en esta metodología de diseño se deben crear transiciones que realicen procesos por un tiempo determinado.

Para el diseño de las transiciones que involucran tiempos de activación o desactivación se utilizan temporizadores del tipo *ON*, debido a que estos se encuentran presentes en todos los *PLC* y simuladores de circuitos ladder.

Adicional a las configuraciones mostradas para los estados se deben considerar ciertos cambios para lograr el funcionamiento deseado, para empezar en el circuito en el que se activan los estados (figura 12.8), se coloca como se muestra en la figura 12.9 una bobina temporizada en paralelo a la bobina del estado correspondiente.

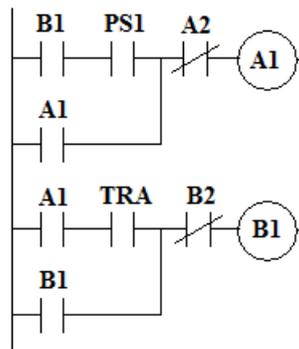


Figura 12.10.: Diagrama Ladder bobina de activación a través de un contacto temporizado

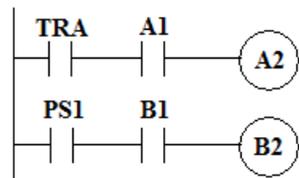


Figura 12.11.: Diagrama Ladder bobina de desactivación a través de un contacto temporizado

En el caso que el tiempo programado determine el cambio de estado en la máquina se deben realizar ajustes a los circuitos de las bobinas de activación y desactivación mostrados en las figuras 12.5 y 12.6 respectivamente, en ambos circuitos el contacto perteneciente al estado de las entradas que permite la transición hacia el estado siguiente se reemplaza por un contacto del temporizador, véase figuras 12.10 y 12.11.

#### 12.2.4. Salidas

Existen dos tipos de salidas, las primeras mostradas en la figura 12.12, se energizan directamente a través de contactos de las bobinas de los estados, permaneciendo de esta manera sólo el tiempo que el estado que las controla así lo hace, mientras que las segundas, haciendo uso de la función interruptor, permanecen energizadas durante varios estados consecutivos, ya que la energización se logra a través de un contacto de la bobina de estado en el que se activa por primera vez y la desenergización se realiza por medio de un contacto de una bobina de desactivación del estado final en que la salida debe estar presente, como se observa en la figura 12.13.

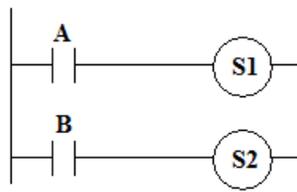


Figura 12.12.: Diagrama Ladder salidas directas

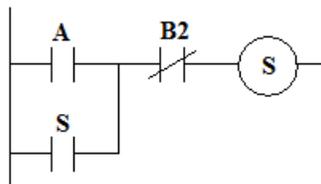


Figura 12.13.: Diagrama Ladder salida varios estados

### 12.2.5. Inicialización de la máquina de estados

Toda máquina de estados cuenta con un estado inicial a partir del cual según sea el valor de las entradas la máquina cambia de estado, pero si se observa la descripción para los estados mostrado en la subsección 12.2.2 esta condición no se considera ya que la ecuación  $M = (Set + M) \bullet \overline{Reset}$ , difiere de la necesaria para un estado inicial, ya que no incluye un término que garantice la activación de este estado en el momento que la máquina entra en funcionamiento. Para dar solución a este problema se incluye el término *Init* de manera que la ecuación para un estado inicial pasa a ser  $M = (Set + M) \bullet \overline{Reset} + Init$ , la inclusión del término *Init* representa cambios en el diagrama ladder de las bobinas de activación de los estados, por lo que se obtienen dos nuevas configuraciones, la primera se muestra en la figura 12.14 y consiste en la utilización de un pulsador extra *P* que energiza una bobina llamada *INIT* que a su vez energiza la bobina de activación (*AI*) ya que el estado *A* es el estado inicial de la máquina, mientras que en la segunda forma la energización de la bobina de activación del estado inicial (*AI*) se realiza de forma directa a través de una línea de contactos *NC* pertenecientes a las bobinas de activación de todos los estados de la máquina excluyendo al estado inicial como se observa en la figura 12.15, con estos contactos basta para garantizar la adecuada activación de este estado, ya que se activa sólo cuando los demás estados no lo están, lo que lleva a que la bobina de desactivación del estado inicial desaparezca.

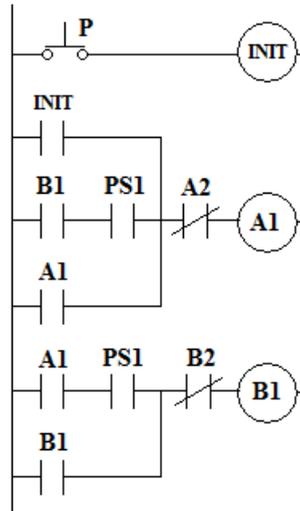


Figura 12.14.: Inicialización máquina de estados a través de un pulsador

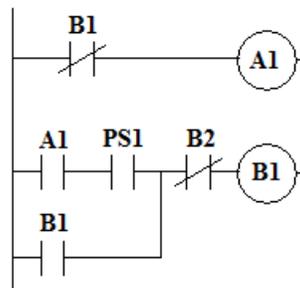


Figura 12.15.: Inicialización máquina de estados directa

## 12.3. PASOS PARA DISEÑAR

- Paso 1: Identificar en el enunciado, el número de entradas, salidas y estados del automatismo, además de definir el estado inicial del problema.
- Paso 2: Construir un diagrama de estados y una tabla de estados que permita identificar las relaciones existentes entre las entradas, los estados y las salidas identificadas en el paso 1.
- Paso 3: Implementar las equivalencias mostradas en la metodología de diseño por máquinas de estado, correspondiente a las entradas, estados (incluyendo el estado inicial) y las salidas necesarias para el buen funcionamiento del automatismo.

## 12.4. EJEMPLOS BÁSICOS

### 12.4.1. *Ejemplo 4. Sistema de encendido y apagado de una lámpara*

Se quiere implementar un automatismo de una entrada que controle el encendido y el apagado de una lámpara, de manera que, cuando la entrada presente un estado lógico alto la lámpara se encienda permaneciendo en ese estado hasta que la entrada presente de nuevo otro estado lógico alto, momento en el cual la lámpara se apaga.

#### Solución

Según las condiciones del sistema se pueden identificar dos estados, un estado A, que corresponde al estado inicial y en el que la lámpara se encuentra apagada y un estado B en el que la lámpara se encuentra encendida. También se puede apreciar una entrada como lo indica el enunciado, además de una salida representada por la lámpara.

Partiendo de la definición de los estados, las entradas y salidas del problema, se construye un diagrama de estados (figura 12.16) y una tabla de estados (tabla 12.1) en los que se aprecian las relaciones entre estos componentes.

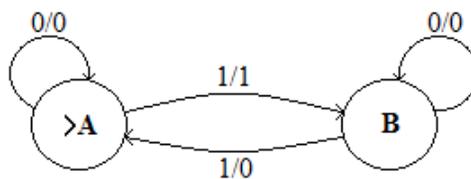


Figura 12.16.: Diagrama de estados ejemplo 4

Estado Actual	Entrada	
	0	1
A	A	B
B	B	A

Tabla 12.1.: Tabla de estados ejemplo 4

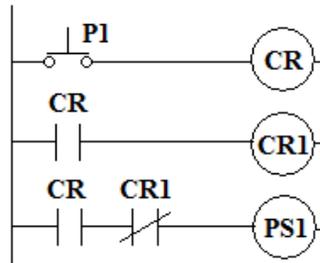


Figura 12.17.: Entrada ejemplo 4

En este diseño es suficiente con la utilización de un pulsador P1 para representar su única entrada. Como el pulsador que se utiliza es *NA*, su estado de reposo representa un estado bajo en la entrada, mientras que cuando se acciona, es decir, se cierra significa que la entrada presenta un estado alto. Acompañando el pulsador se encuentra detectores de flancos de subida que tiene el propósito de impedir que aunque el pulsador se mantenga presionado la entrada se permanezca en un estado alto, todo lo anterior se observa en la figura 12.17.

Como A es el estado inicial del sistema, la energización de la bobina A1 se realiza a través de un contacto *NC* de la bobina de activación del estado B, A1 se mantiene energizada gracias a un contacto de sello y su desenergización se realiza mediante un contacto de la bobina A2 que es la bobina de desactivación de este estado. Para el estado B se extrae de la máquina de estados (figura 12.16) la combinación necesaria para la activación de este, que en el caso del circuito Ladder equivale a contactos de PS1 (variación de la entrada) y de los contactos de A1 y B1 (bobinas de activación de los estados), además de sus respectivos contactos de sello y de desactivación. De todo lo anterior se obtiene el circuito Ladder de la figura 12.18.

Las bobinas de desactivación A2 y B2, que tienen como finalidad desenergizar los estados, se energizan mediante contactos que evidencian el cambio en las entradas (PS1) y la bobina de activación respectiva de cada estado (A1, B1), de tal manera que se obtiene el circuito de la figura 12.19.

Con los contactos *NA* de las bobinas de activación se alimentan los estados propiamente dichos, que se representan con las bobinas marcadas como A y B, adicional a esto se introducen contactos *NC* del estado contrario para garantizar que en un instante de tiempo sólo de encuentre un estado activado. Lo anterior se puede apreciar en la figura 12.20.

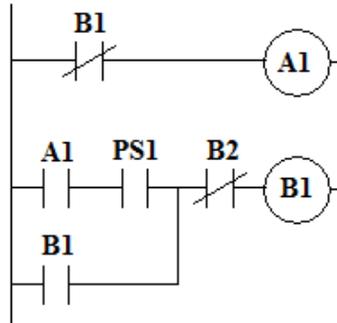


Figura 12.18.: Bobinas de activación ejemplo 4

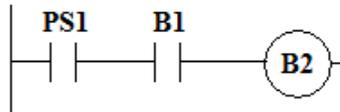


Figura 12.19.: Bobinas de desactivación ejemplo 4

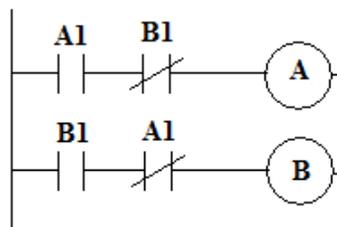


Figura 12.20.: Bobinas estados ejemplo 4



Figura 12.21.: Salida ejemplo 4

Por último teniendo cada uno de los estados representados en bobinas, se pueden utilizar contactos para realizar las acciones de encendido y apagado de las lámpara, de manera que se cumplan las condiciones requeridas por el sistema, es así como se concibe el circuito de la figura 12.21.

Con la integración de los circuitos Ladder mostrados en las figuras 12.17, 12.18, 12.19, 12.20 y 12.21 se completa el diseño requerido para el encendido y apagado de la lámpara.

#### 12.4.2. Ejemplo 5. Desplazamiento de un móvil entre dos límites de recorrido

Un móvil puede desplazarse hacia la derecha (movimiento  $D$ ), o hacia la izquierda (movimiento  $I$ ) hasta llegar a un mecanismo de limitación de recorrido (finales de carrera  $a$  y  $b$ ).

Se dispone además de tres pulsadores  $x$ ,  $l$  y  $r$  de manera que:

- Una presión en  $r$  excita  $D$  y suprime eventualmente a  $I$ .
- Una presión en  $l$  excita  $I$  y suprime eventualmente a  $D$ .
- Una presión en  $x$  suprime a  $D$  e  $I$ .

En la figura 12.22 se puede apreciar el sistema de manera gráfica.

#### Solución

Como ya se ha expuesto, es necesario transformar el enunciado en una máquina de estados. El objetivo fundamental es controlar el movimiento del móvil, por lo que se pueden identificar tres estados:

- Estado A. El móvil no se desplaza hacia ninguno de los dos lados.
- Estado B. Movimiento a la derecha  $D$ .
- Estado C. Movimeiento a la izquierda  $I$ .

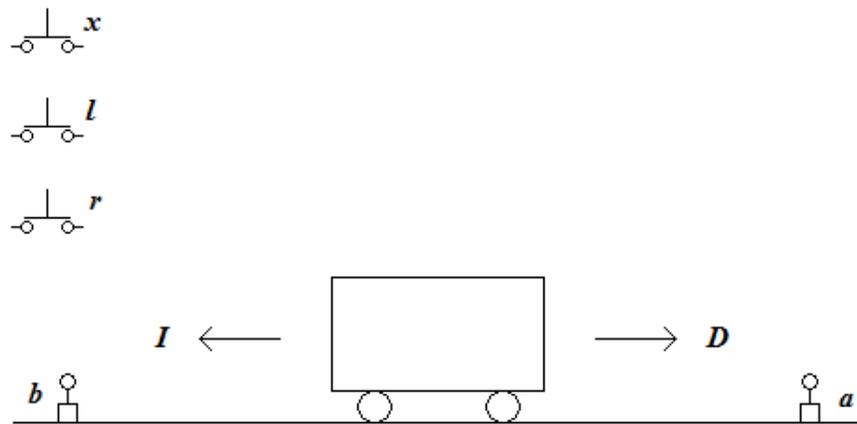


Figura 12.22.: Sistema ejemplo 5

Para identificar claramente las condiciones que se requieren para que haya un cambio de estado y no se presenten errores al momento de realizar los diagramas Ladder se puede construir una tabla de estados como la que se muestra en la tabla 12.2.

Estados	Entradas $x, l, r, a$ y $b$					
	10000	01000	00100	00010	00001	00000
A	A	C	B	-	-	A
B	A	C	B	-	C	B
C	A	C	B	B	-	C

Tabla 12.2.: Tabla de estados ejemplo 5

Con la tabla de estados anterior (tabla 12.2) se puede obtener un diagrama de estados como el que se muestra en la figura 12.23.

En este diagrama es de notar que existe la posibilidad de pasar de un estado a otro mediante dos condiciones diferentes, ya que existe un control interno del sistema (finales de carrera) y un control externo (pulsadores).

Para el diseño en contactos se cuenta con cinco pulsadores que representan las entradas  $x, l, r$  y los finales de carrera ( $a$  y  $b$ ) con sus respectivos detectores de flanco como se observa en la figura 12.24.

Como se tienen tres estados, se necesitan tres bobinas de activación (A1, B1 y C1). Como el estado A es el inicial, su bobina de activación se energiza mediante un arreglo de contactos *NC* de las demás bobinas de activación, para las bobinas B1 y C1 su energización se realiza a través de contactos *NA* de las entradas y los demás estados según las condiciones requeridas por el sistema, las tres bobinas

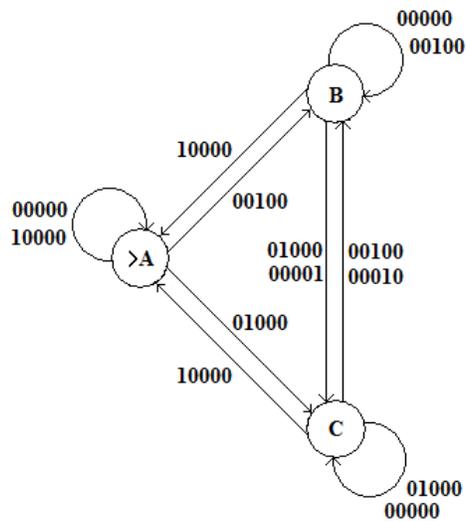


Figura 12.23.: Diagrama de estados ejemplo 5

están acompañadas por sus respectivos contactos de sello y contactos *NC* pertenecientes a la bobina de desactivación correspondiente a cada estado. Todo lo anterior está contenido en la figura 12.25.

En la figura 12.26 puede verse la configuración de las bobinas de desactivacion, en este circuito se incluyen contactos *NA* de las entradas y de las bobinas de activación de los estados como se ha explicado anteriormente.

La configuración para las bobinas de los estados siempre es la misma, en ella se incluye un contacto *NA* de la bobina de activación correspondiente al estado y contactos *NC* de las bobinas de activación de los demás estados como se muestra en la figura 12.27.

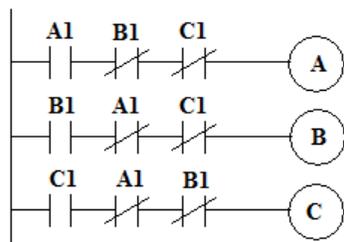


Figura 12.27.: Bobinas de estados ejemplo 5

Para representar el movimiento a la derecha o la izquierda se utilizan contactos *NA* de los estados B y C respectivamente ya que ese fue el planteamiento inicial, de esta manera se obtiene la figura 12.28, con lo que se concluye el diseño de este automatismo.

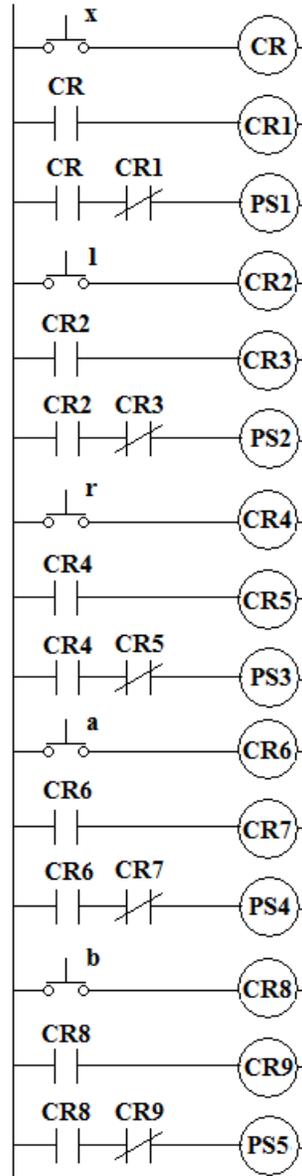


Figura 12.24.: Entradas ejemplo 5

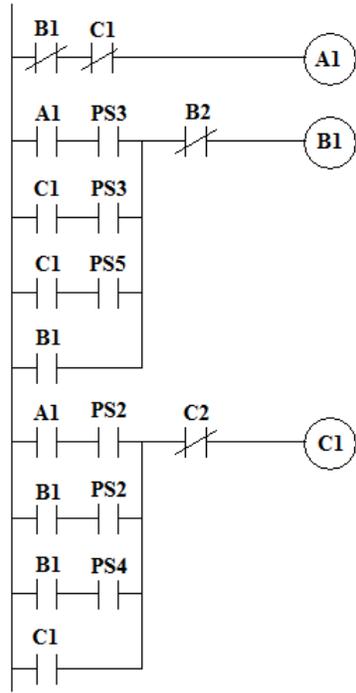


Figura 12.25.: Bobinas de activación ejemplo 5

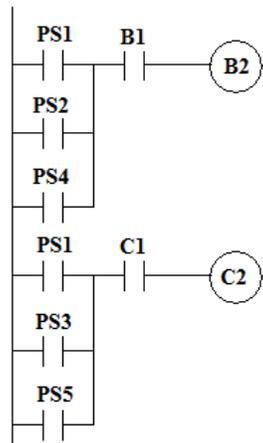


Figura 12.26.: Bobinas de desactivación ejemplo 5

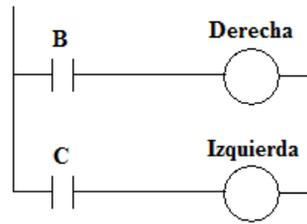


Figura 12.28.: Salidas ejemplo 5



**Parte IV.**

# **RESULTADOS Y CONCLUSIONES**



## 13. RESULTADOS

En este capítulo se muestran algunos ejemplos que tienen una complejidad mayor que los mostrados en los dos capítulos previos, su solución se realiza mediante el uso de una o la combinación de las dos metodologías mostradas en este documento.

### 13.1. *Ejemplo 6.*

Se desea implementar un sistema Ladder de dos entradas  $I1$  e  $I2$  para el control de tres lámparas  $L1$ ,  $L2$  y  $L3$ , según las siguientes condiciones:

- Inicialmente las tres lámparas se encuentran apagadas, y se van encendiendo una a una sin apagar la anterior cada vez que la entrada  $I1$  pase de estado bajo a alto, de manera que se enciende  $L1$  primero, después  $L2$  y por último  $L3$ , luego de que todas se encuentren encendidas con un nuevo cambio de la entrada  $I1$  se apagan las tres lámparas.
- Partiendo igualmente de que las tres lámparas se encuentran apagadas, si la entrada  $I2$  pasa de un estado bajo a uno alto las tres lámparas se encienden y con nuevos cambios de  $I2$  se van apagando una a una, de forma que primero se apaga  $L3$ , luego  $L2$  y finalmente  $L1$  hasta llegar al punto de partida.
- Si ambas entradas se encuentran en estados bajos no se presenta ningún cambio en el sistema, a diferencia de esto, si las dos entradas cambian de un estado bajo a uno alto en el mismo instante de tiempo, el sistema realiza un salto en la secuencia de apagado o encendido de las lámparas, de tal manera que si todas las lámparas se encuentran apagadas  $L1$  y  $L2$  se encienden, pero si estas se encuentran encendidas las dos se apagan, mientras que si  $L1$  está encendida y se realiza el cambio en las entradas mencionado,  $L2$  y  $L3$  se encienden, y en el caso de que las tres lámparas estén encendidas  $L2$  y  $L3$  se apagan.

**Solución**

Según las condiciones del sistema, se pueden identificar cuatro estados en él, de manera que:

- Estado A. Las tres lámparas  $L1$ ,  $L2$  y  $L3$  se encuentran apagadas.
- Estado B.  $L1$  está encendida, mientras que  $L2$  y  $L3$  permanecen apagadas.
- Estado C.  $L1$  y  $L2$  se encienden y  $L3$  está apagada.
- Estado D. Las tres lámparas  $L1$ ,  $L2$  y  $L3$  se encuentran encendidas.

A partir de la definición de los estados se puede deducir una máquina de estados (figura 13.1) como una tabla de estados (tabla 13.1) que representan el sistema de encendido y apagado de las lámparas que se quiere diseñar.

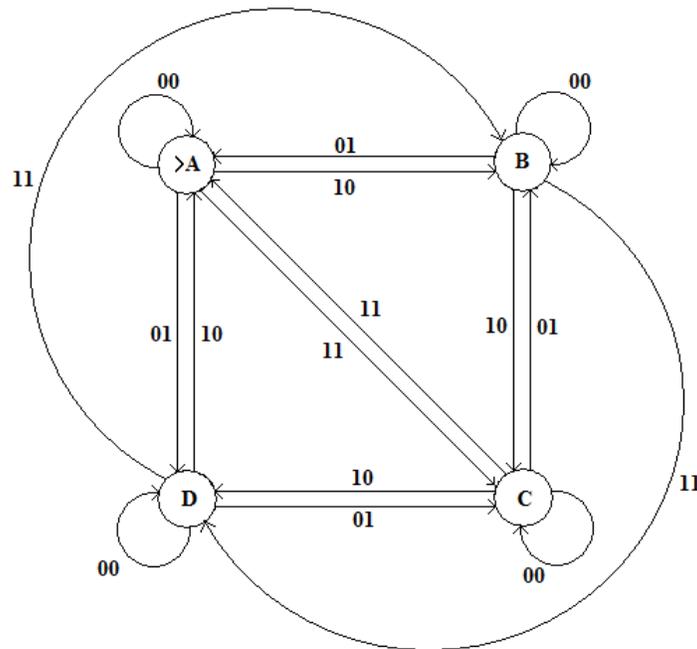


Figura 13.1.: Diagrama de estados ejemplo 6

Tabla 13.1.: Tabla de estados ejemplo 6

Estado Actual	Entradas I1 e I2			
	00	01	10	11
A	A	D	B	C
B	B	A	C	D
C	C	B	D	A
D	D	C	A	B

En este diseño basta con la utilización de dos pulsadores  $P1$  y  $P2$  para representar las entradas  $I1$  e  $I2$  respectivamente, pero como se requiere simular los circuitos obtenidos para comprobar su buen funcionamiento se incluye un tercer pulsador  $P3$  que permite identificar el momento en que las dos entradas cambian de un estado bajo a uno alto en el mismo instante de tiempo. Como los pulsadores que se utilizan son  $NA$ , su estado de reposo representa un estado bajo en las entradas, mientras que cuando se acciona, es decir, se cierra significa que las entradas presentan un estado alto. Acompañando los pulsadores se encuentran detectores de flancos de subida que tiene el propósito de impedir que aunque el pulsador se mantenga presionado la entrada se permanezca en un estado alto, todo lo anterior se observa en la figura 13.2.

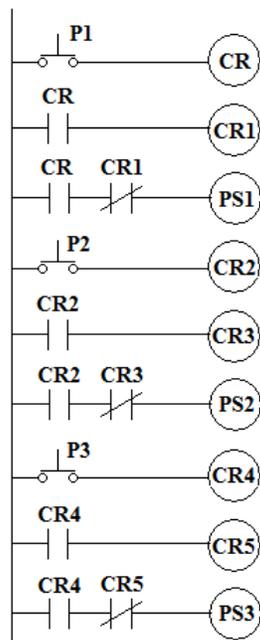


Figura 13.2.: Entradas ejemplo 6

Como  $A$  es el estado inicial del sistema, la energización de la bobina  $A1$  se realiza a través de contactos  $NC$  de las demás bobinas de activación de los estados restantes,  $A1$  se mantiene energizada gracias a un contacto de sello y su desenergización se realiza mediante un contacto de la bobina  $A2$  que es la bobina de desactivación de este estado. Para los demás estados se extraen de la máquina de estados (figura 13.1) las combinaciones necesarias para la activación de cada uno de ellos, que en el caso del circuito Ladder equivalen a contactos de  $PS1$ ,  $PS2$  y  $PS3$  (variación de las entradas  $I1$  e  $I2$ ) y de los contactos de  $A1$ ,  $B1$ ,  $C1$  y  $D1$  (bobinas de activación de los estados), además de sus respectivos contactos de sello y de desactivación. De todo lo anterior se obtiene el circuito Ladder de la figura 13.3.

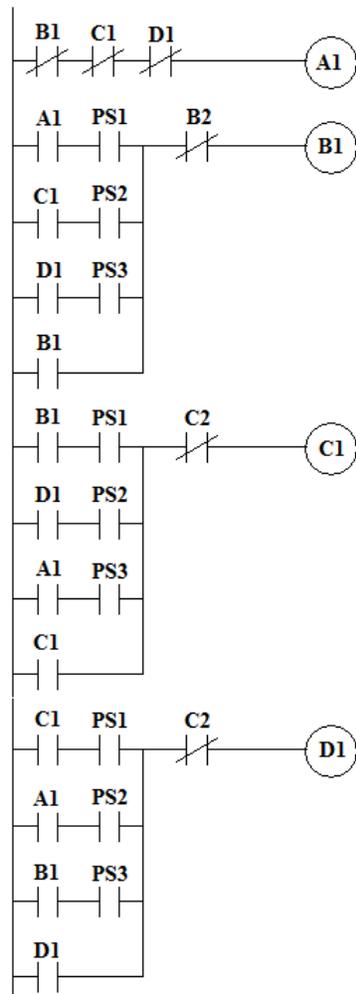


Figura 13.3.: Bobinas de activación ejemplo 6

Las bobinas de desactivación  $A2$ ,  $B2$ ,  $C2$  y  $D2$  que tienen como finalidad desenergizar los estados, se energizan mediante contactos que evidencian el cambio en las entradas ( $PS1$ ,  $PS2$  y  $PS3$ ) y la bobina de activación respectiva de cada estado ( $A1$ ,  $B1$ ,  $C1$  y  $D1$ ), de tal manera que se obtiene el circuito de la figura 13.4.

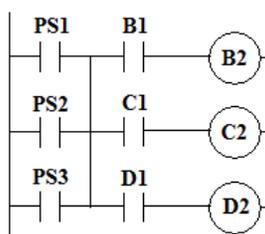


Figura 13.4.: Bobinas de desactivación ejemplo 6

Con los contactos *NA* de las bobinas de activación se alimentan los estados propiamente dichos, que se representan con las bobinas marcadas como *A*, *B*, *C* y *D*, adicional a esto se introducen contactos *NC* de los demás estados para garantizar que en un instante de tiempo sólo se encuentre un estado activado. Lo anterior se puede apreciar en la figura 13.5.

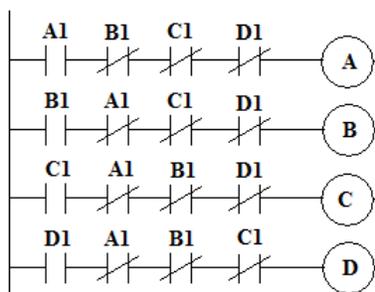


Figura 13.5.: Bobinas estados ejemplo 6

Por último teniendo cada uno de los estados representados en bobinas, se pueden utilizar contactos para realizar las acciones de encendido y apagado de las lámparas *L1*, *L2* y *L3* de manera que se cumplan las condiciones requeridas por el sistema, es así como se concibe el circuito de la figura 13.6.

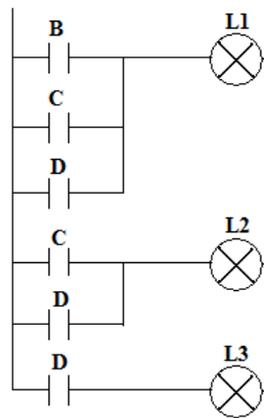


Figura 13.6.: Salidas ejemplo 6

Basta con unir los circuitos mostrados en las figuras 13.2, 13.3, 13.4, 13.5 y 13.6, para tener completo el diseño que satisface las necesidades planteadas para el encendido y apagado de las lámparas.

## 13.2. Ejemplo 7.

Se necesita la implementación de un automatismo que controle un sistema de envasado de objetos esféricos (figura 13.7), si su funcionamiento es el siguiente:

La orden de inicio se da mediante un pulsador, cuyo accionamiento provoca la apertura de la compuerta para la salida de esferas del contenedor principal, las cuales irán cayendo por su propio peso a través de un tubo. Al salir del tubo, cada objeto es detectado por un sensor de proximidad, el cual activa un cilindro de simple efecto que empuja la bola al interior de una caja. La caja está llena cuando se han depositado tres bolas en ella, momento en el cual es desplazada por un cilindro progresivo, movimiento que tarda dos segundos, hasta situar la caja sobre una banda transportadora, que está encargada de retirarla hacia un depósito, este último proceso toma un tiempo de seis segundos, quedando todo el sistema listo para un nuevo orden de inicio.

### Solución

Teniendo en cuenta las condiciones para el buen funcionamiento del sistema de empaquetado de esferas, se pueden identificar tres estados, un estado *A* en el que se realiza la apertura de la compuerta para la salida de las esferas, y el conteo de las mismas a la hora de introducirlas en su caja respectiva, un estado *B* en el cual se activa un cilindro progresivo y un estado *C* en el que entra en funcionamiento una banda transportadora. También se pueden apreciar dos entradas la primera identificable como un

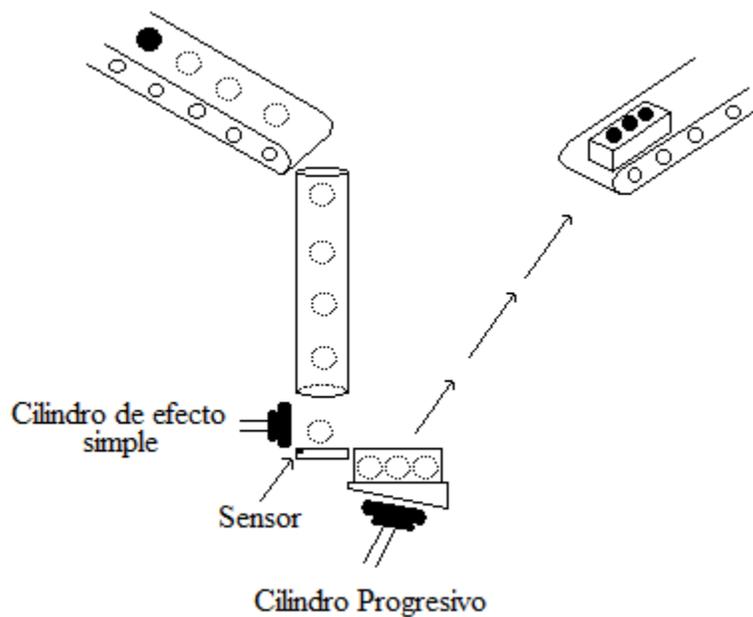


Figura 13.7.: Sistema empaquetado de esferas

botón de inicio que permite la activación del estado A y una segunda entrada que representa un sensor capaz de activar el cilindro de simple efecto controlado en el mismo estado. Teniendo en cuenta lo descrito anteriormente se aprecian cuatro salidas que representan la compuerta, el cilindro de simple efecto, el cilindro progresivo y la banda transportadora.

Para representar el botón de inicio y el botón sensor es necesario la implementación de dos pulsadores *Inicio* y *Sensor* respectivamente. Como los pulsadores que se utilizan son *NA*, su estado de reposo se representa como un “0” lógico en las entradas, y cuando se presionan, se representa como un “1” lógico. Acompañando los pulsadores se encuentran detectores de flancos de subida que tiene el propósito de impedir que aunque el pulsador se mantenga accionado, la entrada se mantenga en un estado alto, todo lo anterior se muestra en la figura 13.8.

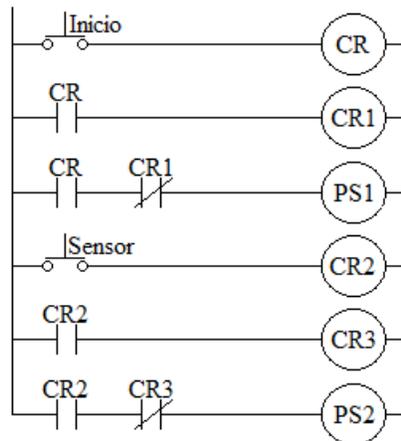


Figura 13.8.: Entradas ejemplo 7

Al igual como se ha trabajado anteriormente las bobinas de los estados se activan mediante contactos *NA* de las bobinas de activación respectivas, además de garantizar la activación de un único estado en un momento dado con la inclusión de contactos *NC* de las bobinas de activación de los estados restantes. Lo anterior se representa mediante el diagrama Ladder de la figura 13.9.

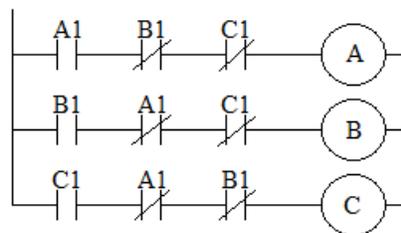


Figura 13.9.: Bobinas de estados ejemplo 7

Las bobinas de activación se energizan mediante la combinación de los estados precedentes y contactos de bobinas que permiten la transición hacia el estado siguiente, y su desenergización se logra mediante un contacto *NC* de las bobinas de desactivación de cada estado. Lo anterior se observa en la figura 13.10.

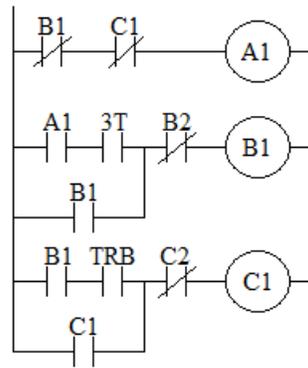


Figura 13.10.: Bobinas de activación ejemplo 7

Las bobinas de desactivación  $A2$ ,  $B2$  y  $C2$  tienen como finalidad desactivar cada uno de los estados. Su energización se logra por medio de contactos  $NA$  de las bobinas de activación de cada estado y de las transiciones. Lo anterior se muestra en la figura 13.11.

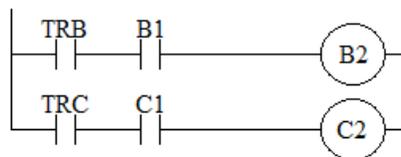


Figura 13.11.: Bobinas de desactivación ejemplo 7

En este ejemplo, las transiciones entre los estados están determinadas por las bobinas  $3T$ ,  $TRB$  y  $TRC$  estas dos últimas del tipo temporizadas. La bobina  $3T$  permite la transición del estado  $A$  al estado  $B$  ya que se activa cuando el conteo de esferas ingresadas a la caja corresponde al número tres. La transición del estado  $B$  a  $C$  se logra mediante los contactos de la bobina  $TRB$  que se encuentra temporizada a dos segundos, tiempo que tarda el cilindro progresivo en depositar la caja sobre la banda transportadora. Por último la bobina  $TRC$  determina la finalización del proceso luego de que el tiempo programado en ella, que es de seis segundos culmine, tiempo necesario para que la banda transportadora lleve la caja hasta su punto de almacenamiento. La configuración de estas tres bobinas se puede observar en la figura 13.12.

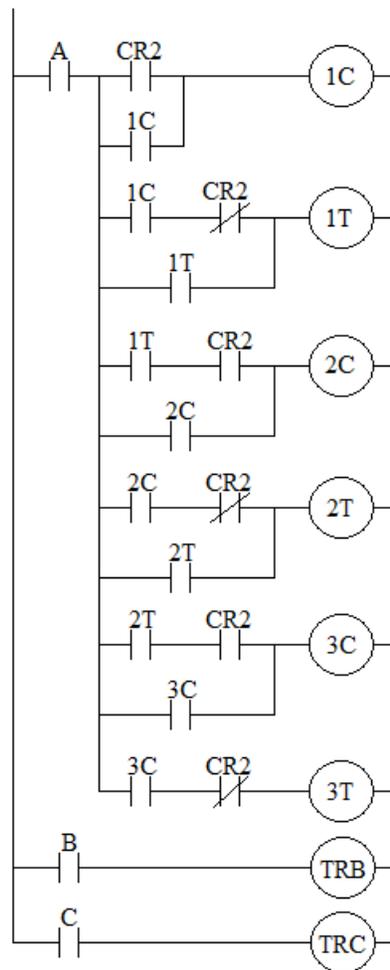


Figura 13.12.: Bobinas transición ejemplo 7

Como se había expuesto inicialmente en la solución de este problema, existen cuatro salidas, que se activan de la siguiente forma: el estado *A* permite la energización de las salidas compuerta y cilindro de simple efecto, como este último se activa cuando el sensor detecta una esfera, cuenta con un contacto adicional perteneciente a la bobina *PS2*. El estado *B* energiza el cilindro progresivo y el estado *C* permite el funcionamiento de la banda transportadora. El diseño anterior se muestra en la figura 13.13.

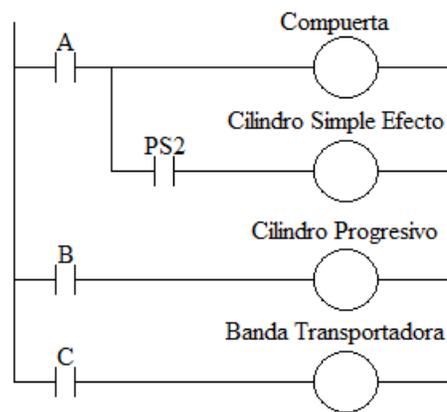


Figura 13.13.: Salidas ejemplo 7



## 14. CRITERIOS DE VALIDEZ Y CONFIABILIDAD

Para verificar la confiabilidad y validez de los resultados se implementaron con las metodologías desarrolladas, ejemplos de aplicación encontrados en algunas referencias como se muestra a continuación.

### 14.1. Ejemplo 8.

Este ejemplo se encuentra en la referencia [24], en él se quiere modelar el comportamiento de un horno microondas, como sigue:

El horno microondas posee una puerta. Si la puerta está cerrada, entonces puede estar o no en funcionamiento (según se prenda o apague). Estando prendido no es posible abrir la puerta del horno sin antes apagarlo. También, en cualquier momento es posible establecer el modo de cocción.

De este modo se identifican tres estados:

Estado *A*: Puerta abierta, encendido deshabilitado.

Estado *B*: Puerta cerrada, horno apagado.

Estado *C*: Puerta cerrada, horno prendido.

La descripción anterior puede apreciarse en la figura 14.1.

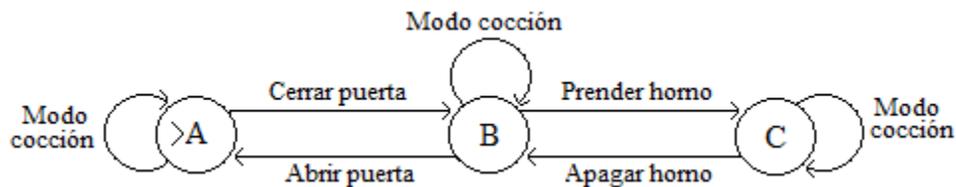


Figura 14.1.: Diagrama de estados ejemplo 8

**Solución:**

Se requieren cinco pulsadores para representar las entradas abrir puerta (A.P), cerrar puerta (C.P), prender horno (P.H), apagar horno (A.H) y modo cocción (M.C) como se observa en la figura 14.2.

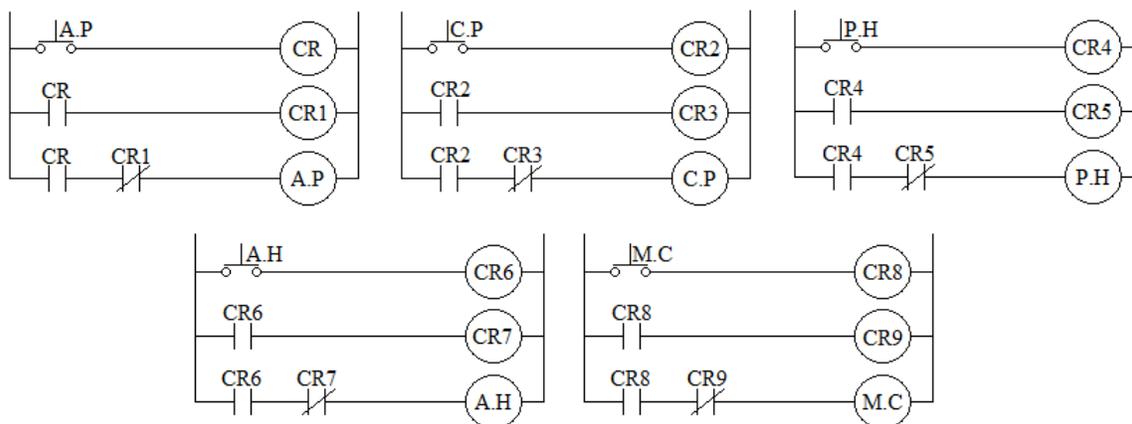


Figura 14.2.: Entradas ejemplo 8

Las bobinas de activación se energizan mediante la combinación de los estados anteriores y contactos *NA* de bobinas que permiten la transición hacia el estado siguiente, además de su correspondiente contacto de sello. Su desenergización se logra mediante un contacto *NC* de las bobinas de desactivación de cada estado. Lo anterior se observa en la figura 14.3.

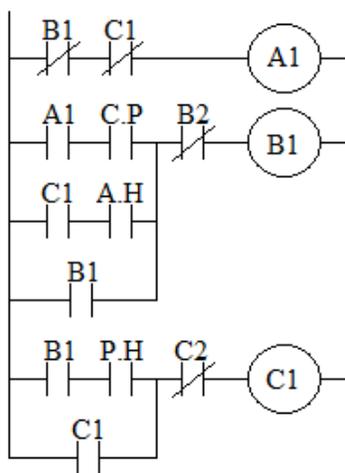


Figura 14.3.: Bobinas de activación ejemplo 8

Las bobinas de desactivación son las encargadas de desenergizar cada uno de los estados y se ener-

gizan mediante contactos *NA* de las bobinas de activación de cada estado y de sus correspondientes transiciones. La figura 14.4 muestra lo expuesto anteriormente.

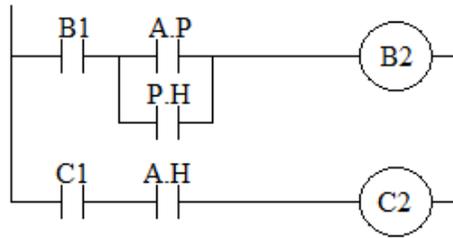


Figura 14.4.: Bobinas de desactivación ejemplo 8

La energización de las bobinas de estados se hace mediante contactos *NA* de las bobinas de activación, además de garantizar la energización de un único estado en un momento dado con la incorporación de contactos *NC* de las bobinas de activación de los estados restantes. Lo anterior se muestra en la figura 14.5.

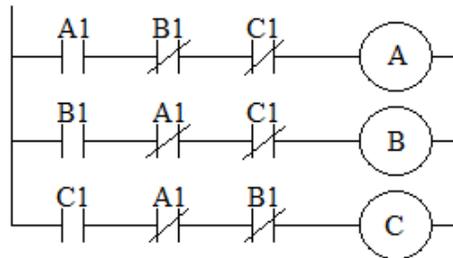


Figura 14.5.: Bobinas de estados ejemplo 8

Las salidas para este sistema se representan de la siguiente forma: el estado *A* permite la energización de la puerta abierta y el encendido deshabilitado, el estado *B* muestra la energización del horno apagado y la puerta cerrada y el estado *C* que activa el encendido del horno y la puerta cerrada, y una salida adicional la cual indica el modo de cocción en cualquier estado. Lo anterior se muestra en la figura 14.6.

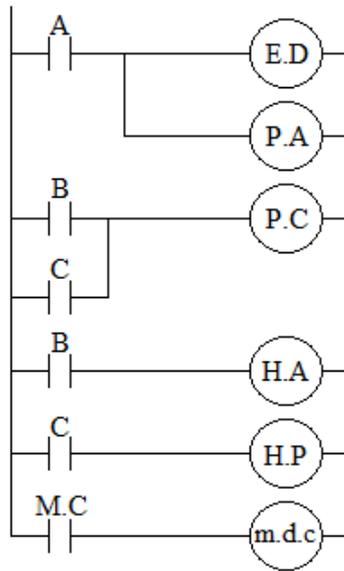


Figura 14.6.: Salidas ejemplo 8

## 14.2. Ejemplo 9.

Una variante sugerida en la referencia [24] para el ejemplo del horno microondas es la mostrada en la figura 14.7, en ella puede observarse el tratamiento de este problema por medio de tres diagramas de estados: en el primero se modela una puerta que se abre y se cierra, en el segundo un horno que se prende y apaga, y el encendido deshabilitado, y en el tercero el tratamiento del modo de cocción de manera independiente.

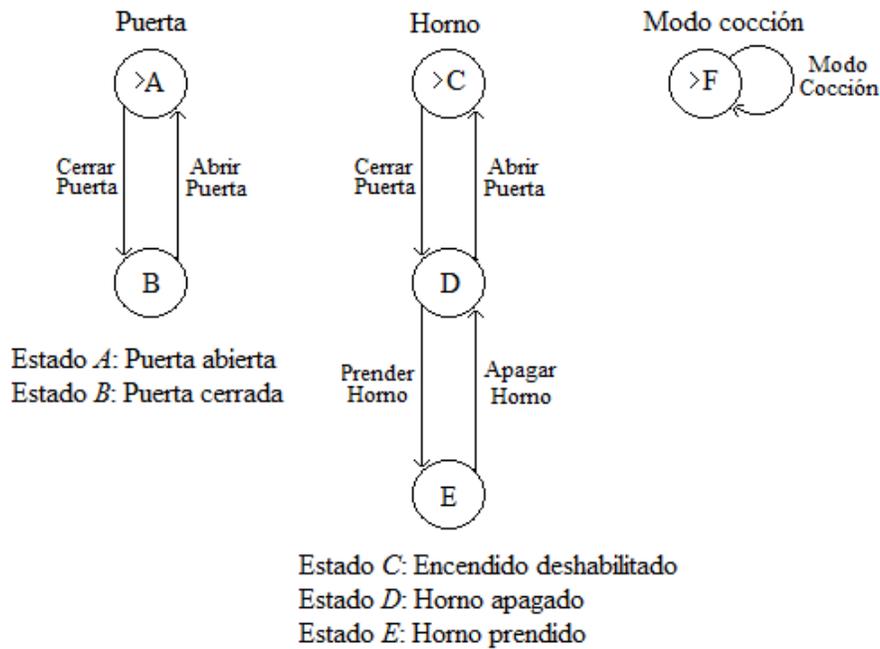


Figura 14.7.: Diagrama de estados ejemplo 9

**Solución:**

Diagrama de estados correspondiente a la *puerta*.

Para representar la apertura y cierre de la puerta es necesaria la utilización de dos pulsadores *NA*. La figura 14.8 muestra las entradas para este sistema.

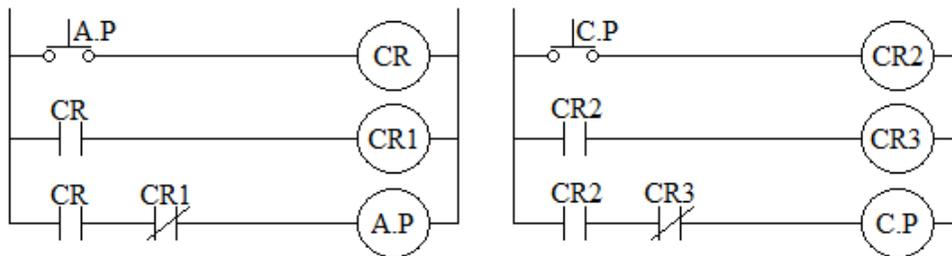


Figura 14.8.: Entradas de la puerta ejemplo 9

Las bobinas de activación se energizan por medio de contactos *NA* del estado anterior y de su correspondiente transición, además de su contacto de sello y un contacto *NC* de las bobbinas de desactivación de cada estado. lo anterior se observa en la figura 14.9.

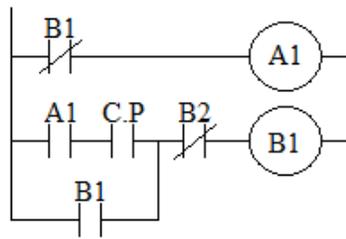


Figura 14.9.: Bobinas de activación de la puerta ejemplo 9

El sistema posee una sola bobina de desactivación la cual se energiza mediante contactos NA de la bobina de activación del estado B y su correspondiente transición. Lo anterior se aprecia en la figura 14.10.

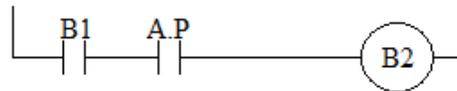


Figura 14.10.: Bobina de desactivación de la puerta ejemplo 9

La activación de las bobinas de cada estado A y B se hace por medio de contactos NA de sus correspondiente bobinas de activación y contactos NC de la bobina de activación del estado restante. Lo anterior se observa en la figura 14.11.

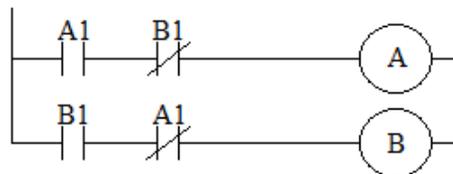


Figura 14.11.: Bobinas de estados de la puerta ejemplo 9

Las salidas para el sistema de la puerta se definen de la siguiente manera: el estado A es el encargado de energizar la bobina que abre la puerta, mientras que el estado B se encarga de controlar la bobina que cierra la puerta.

Diagrama de estados correspondiente al *horno*.

Para representar el encendido deshabilitado, el apagado y encendido del horno es necesaria la utilización de tres pulsadores NA. Como puede apreciarse en la figura 14.12 donde se muestran las entradas del sistema adicionales a las mostradas en la figura 14.8.

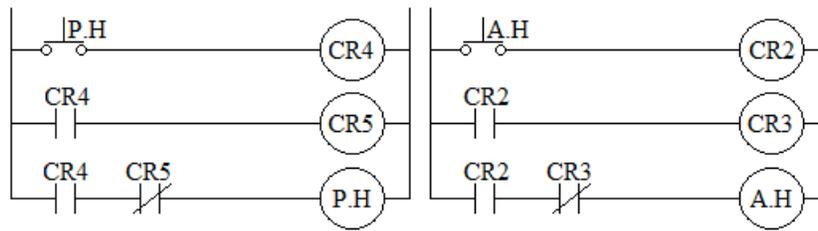


Figura 14.12.: Entradas del horno ejemplo 9

Las bobinas de activación se energizan mediante la combinación de los estados anteriores y contactos *NA* de bobinas que permiten la transición hacia el estado siguiente, además de su correspondiente contacto de sello. Su desenergización se logra mediante un contacto *NC* de las bobinas de desactivación de cada estado. Lo anterior se observa en la figura 14.13.

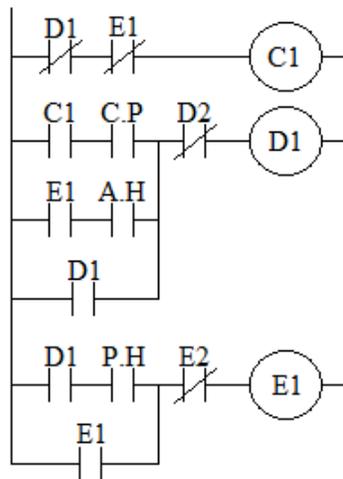


Figura 14.13.: Bobinas de activación del horno ejemplo 9

Las bobinas de desactivación son las encargadas de desenergizar cada uno de los estados y se energizan mediante contactos *NA* de las bobinas de activación de cada estado y de sus correspondientes transiciones. La figura 14.14 muestra lo mencionado anteriormente.

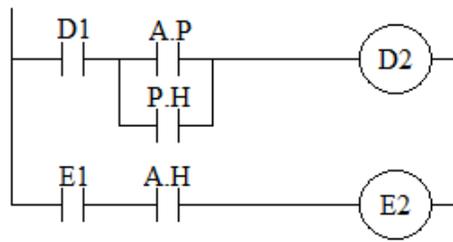


Figura 14.14.: Bobinas de desactivación del horno ejemplo 9

La energización de las bobinas de estados se hace mediante contactos *NA* de las bobinas de activación, además de garantizar la energización de un único estado en un momento dado con la incorporación de contactos *NC* de las bobinas de activación de los estados restantes. Lo anterior se muestra en la figura 14.15.

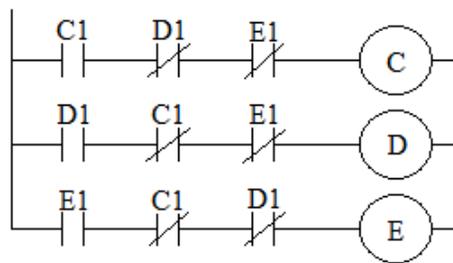


Figura 14.15.: Bobinas de estados del horno ejemplo 9

Las salidas para este sistema se representan de la siguiente forma: el estado *C* permite la energización del encendido deshabilitado, el estado *D* muestra la energización del horno apagado, el estado *E* energiza el encendido del horno. Lo anterior se muestra en la figura 14.16.

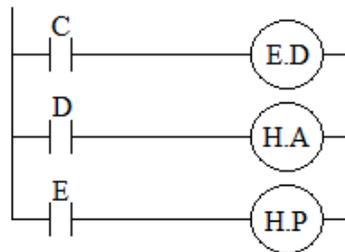


Figura 14.16.: Salidas del horno ejemplo 9

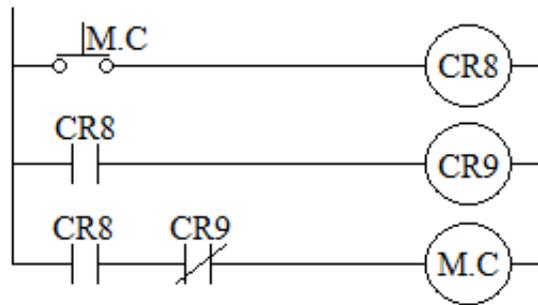


Figura 14.17.: Entrada del modo de cocción ejemplo 9

Diagrama de estados correspondiente al *modo de cocción*.

Este diagrama representado por un único estado precisa de la utilización de una entrada para realizar el accionamiento del modo de cocción (figura 14.17), de una bobina de estados, es este caso maracado como F (figura 14.18), y de una salida que indique que se ha realizado el activado el modo de cocción (figura 14.19).

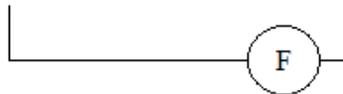


Figura 14.18.: Bobina de estado del modo de cocción ejemplo 9



Figura 14.19.: Salida modo de cocción ejemplo 9

### 14.3. Ejemplo 10.

Este ejemplo se encuentra en la referencia [9], en él se busca diseñar un sistema que reconozca una secuencia de entrada que consta precisamente de dos ceros seguidos por un 10. En otras palabras se debe obtener la siguiente secuencia de salida  $z$  a partir de la secuencia de entrada  $x$  dada:

$$x = 001001000010010$$

$$z = 000100100001001$$

### Solución

El desarrollo de este ejemplo en la referencia [9] se enfoca en el diseño es de un reconocedor de secuencias, en él se establece una cadena de transición de estado correspondiente a la secuencia de entrada correcta. A continuación se examina el diagrama de estados de la figura 14.20.

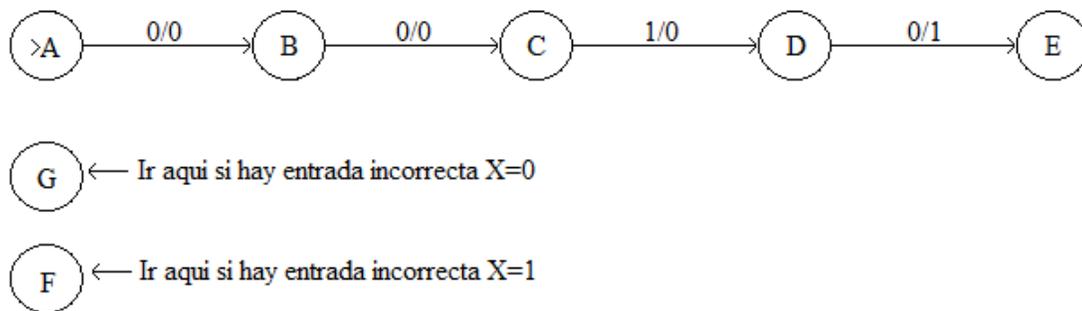


Figura 14.20.: Diagrama de estados ejemplo 10.1

Si el circuito comienza en el estado A, una secuencia de entrada 0010 devuelve una salida  $z = 1$  que coincide con el último  $x = 0$  de la secuencia. También se agregan dos estados que se utilizan como condiciones de error, es decir, que se envía a estos estados si la secuencia de entrada no es correcta. La figura 14.21 ilustra algunas transiciones obvias a los estados de error.

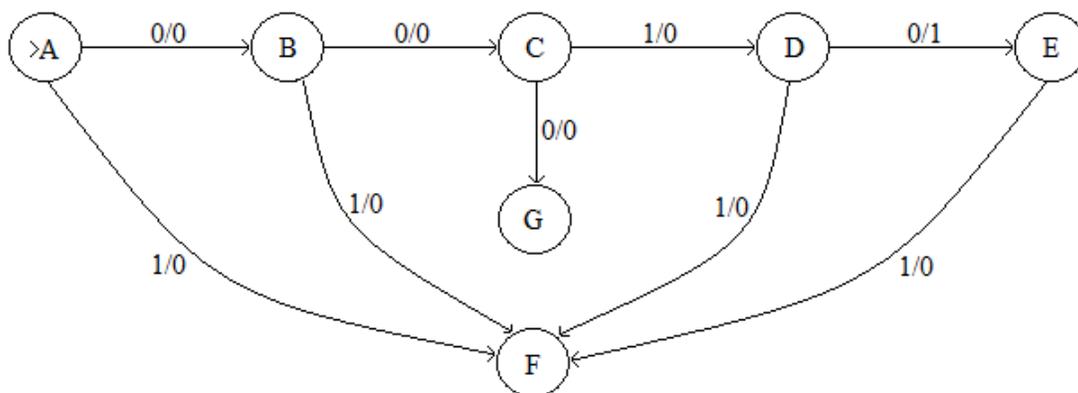


Figura 14.21.: Diagrama de estados ejemplo 10.2

Para completar el diagrama de estados, se deben tener dos arcos que salgan de cada estado, uno para cada condición de entrada. Aún no se ha definido el arco  $x = 0$  que sale del estado  $E$ . Este arco debe regresar al estado  $C$  para permitir el reconocimiento de las secuencias traslapadas, por lo que no debe de ir al estado  $G$ . Lo anterior se muestra en la figura 14.22.

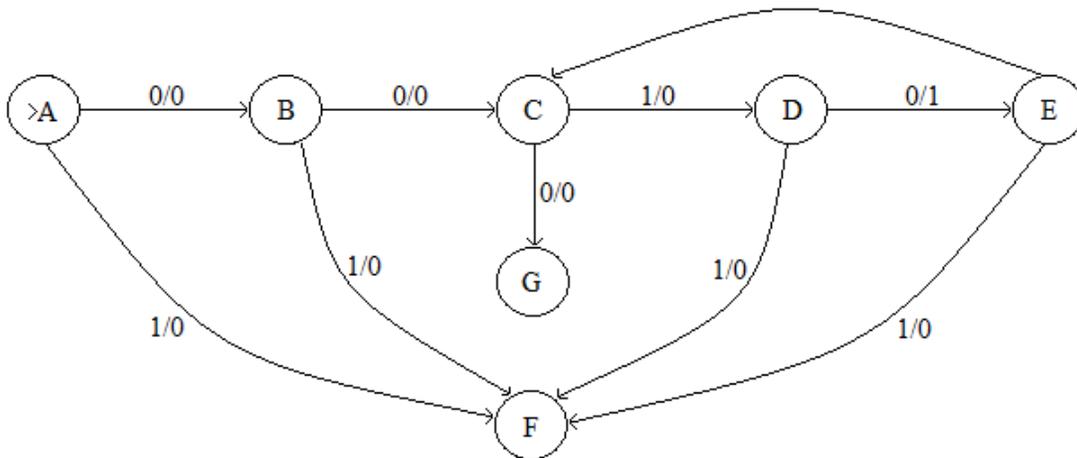


Figura 14.22.: Diagrama de estados ejemplo 10.3

Después se debe de completar las transiciones que sales de los estados de error  $F$  y  $G$ . Para el estado  $F$ , si se presenta una entrada  $x = 0$ , entonces existe un patrón de secuencia 10, que puede ser el inicio de una secuencia de entrada válida, por lo que se envía el circuito al estado  $B$  en la fila horizontal de estados del diagrama de estados. Las demás entradas  $x$  al circuito en los estados  $F$  y  $G$  causan transiciones dentro de estos estados de error, como se observa en la figura 14.23.

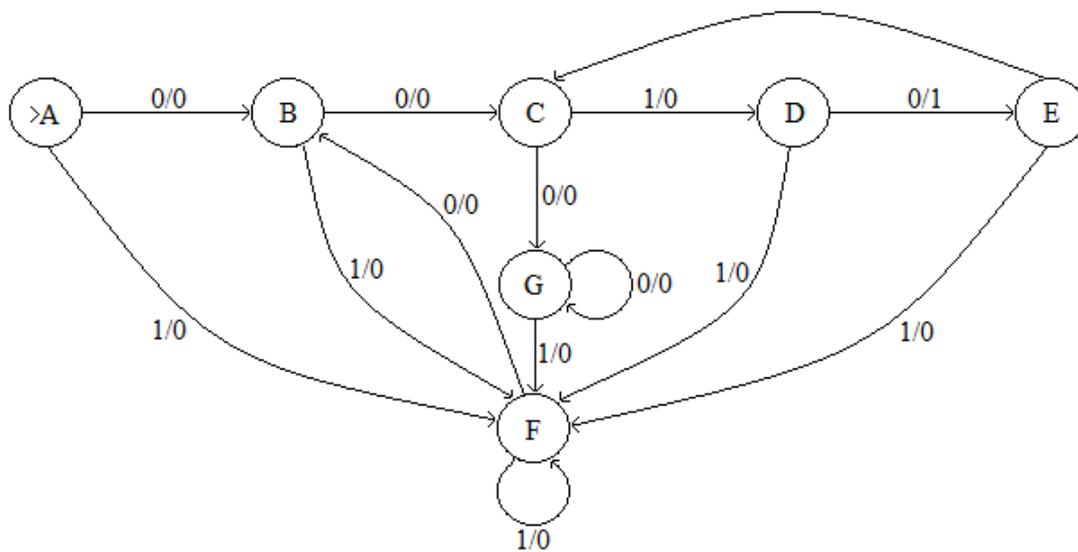


Figura 14.23.: Diagrama de estados ejemplo 10.4

Ahora se debe buscar los estados equivalentes en el diseño. En la tabla de estados (tabla 14.1) se observa que  $A = F$  y  $B = E$  ya que tienen filas idénticas, de esta manera se eliminan las filas  $E$  y  $F$ . Así, la tabla 14.2 y el diagrama de estados de la figura 14.24 muestran el circuito reducido.

Tabla 14.1.: Tabla de estados ejemplo 10

Estado Actual	Entrada x	
	0	1
A	B/0	F/0
B	C/0	F/0
C	G/0	D/0
D	E/1	F/0
E	C/0	F/0
F	B/0	F/0
G	G/0	F/0

Tabla 14.2.: Tabla de estados reducida ejemplo 10

Estado Actual	Entrada x	
	0	1
A	B/0	A/0
B	C/0	A/0
C	G/0	D/0
D	B/1	A/0
G	G/0	A/0

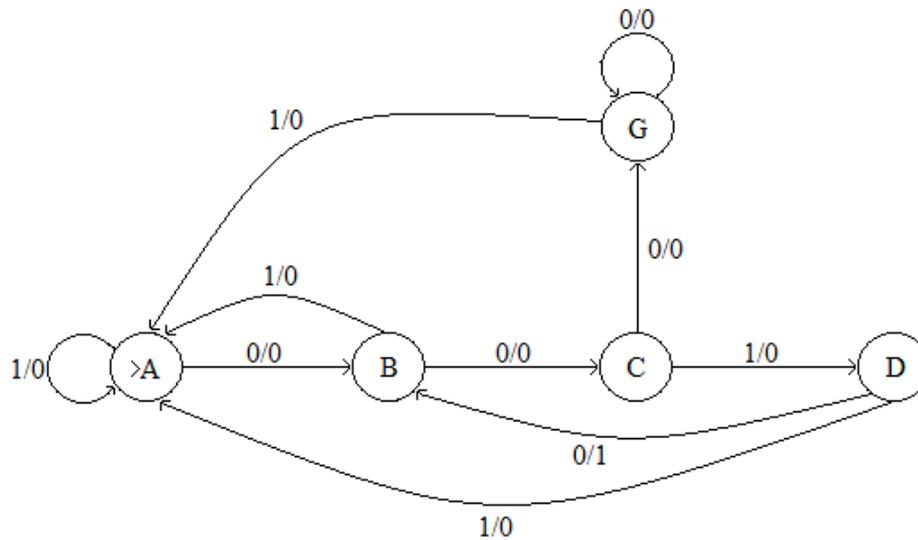


Figura 14.24.: Diagrama de estados reducido ejemplo 10

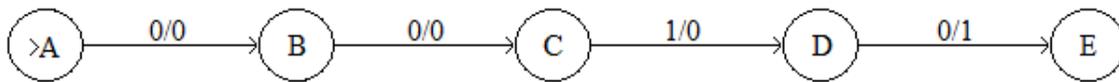


Figura 14.25.: Diagrama de estados secuencia de entrada correcta ejemplo 10

Debido a que este ejemplo se desea implementar con la metodología de diseño por máquinas de estado se sugiere el siguiente desarrollo, adicionándole a las condiciones planteadas en la referencia [9], la necesidad de observar además de que se ha introducido la secuencia correcta, el momento que dicha acción se ha ejecutado en tres ocasiones.

Este nuevo desarrollo parte del diagrama de estados mostrado en la figura 14.25, en él se observa una secuencia correcta de entrada cuando se activa el estado E.

Luego de que se ha introducido una secuencia correcta (estado E), se presentan dos casos, en el primero la siguiente entrada es un cero, razón por la cual la secuencia debe traslaparse llevando la máquina de estados al estado C, en cambio si la entrada siguiente corresponde a un uno la secuencia debe reiniciarse, por tal motivo la máquina pasa al estado A. Cuando la máquina se encuentra en los estado B ó D y la entrada siguiente corresponde a un uno, la secuencia es errónea, condición en la cual la máquina debe situarse en el estado A, para que se ingrese una nueva secuencia. Si la máquina se encuentra en el estado A y la entrada es un uno debe permanecer en dicho estado hasta que la entrada sea cero para pasara al estado B, caso similar se presenta en el estado C, la máquina debe mantenerse en ese estado mientras la entrada sea cero y sólo debe pasar al estado D cuando la entrada sea uno. Introduciendo los conceptos mencionados anteriormente, se obtiene el diagrama de estados mostrado

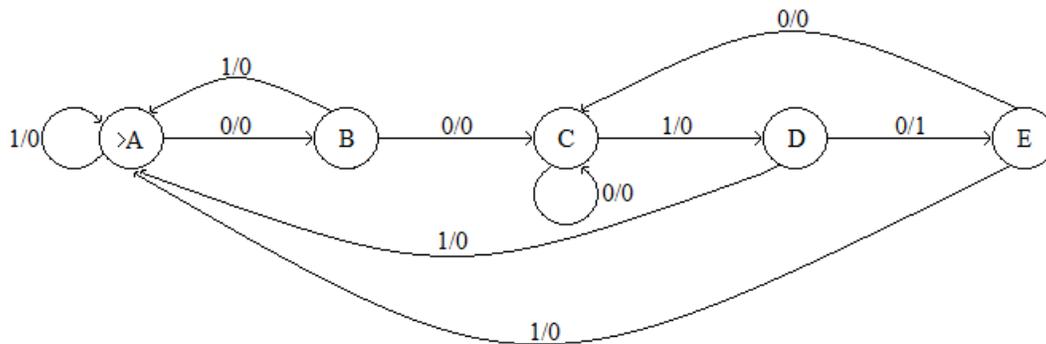


Figura 14.26.: Diagrama de estados completo segundo desarrollo ejemplo 10

Tabla 14.3.: Tabla de estados segundo desarrollo ejemplo 10

Estado Actual	Entrada x	
	0	1
A	B/0	A/0
B	C/0	A/0
C	C/0	D/0
D	E/1	A/0
E	C/0	A/0

en la figura 14.26.

Realizando una tabla de estados equivalente para el diagrama de estados de la figura 14.26, se obtiene la tabla 14.3, en ella puede apreciarse que los estados B y E son iguales, razón por la cual el estado E se elimina, obteniendo así una tabla de estados reducida mostrada en la tabla 14.4, y un diagrama de estados reducido que se observa en la figura 14.27.

Según las condiciones de diseño se requiere de un estado que indique que la secuencia de entrada es correcta, además de mostrar cuando se ha efectuado su ingreso en tres ocasiones, se precisa de la utilización del diagrama de estados de la figura 14.26, ya que el estado E brinda esa posibilidad al tener un único camino de activación, situación que no se presenta en el diagrama de la figura 14.27,

Tabla 14.4.: Tabla de estados reducida segundo desarrollo ejemplo 10

Estado Actual	Entrada x	
	0	1
A	B/0	A/0
B	C/0	A/0
C	C/0	D/0
D	B/1	A/0

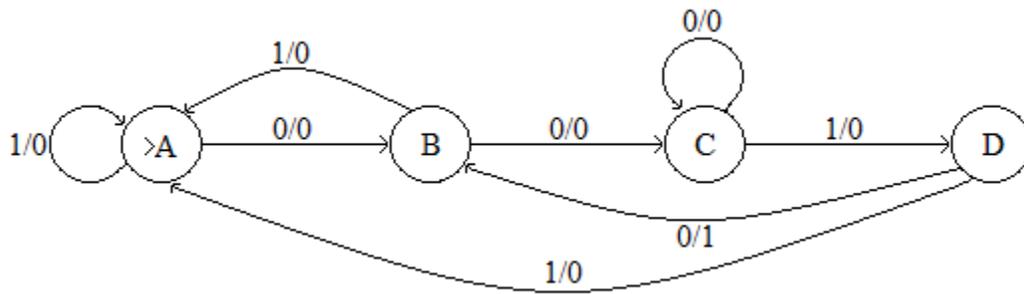


Figura 14.27.: Diagrama de estados reducido segundo desarrollo ejemplo 10

porque el estado B en ocasiones indica que la secuencia de entrada es correcta, mientras que en otro momento sólo sirve como estado de paso hacia otros estados según sea la entrada.

Comenzando con el diseño Ladder, para representar la secuencia de entrada por medio de ceros y unos se requiere la utilización de dos pulsadores *NA* para su representación como se observa en la figura 14.28.

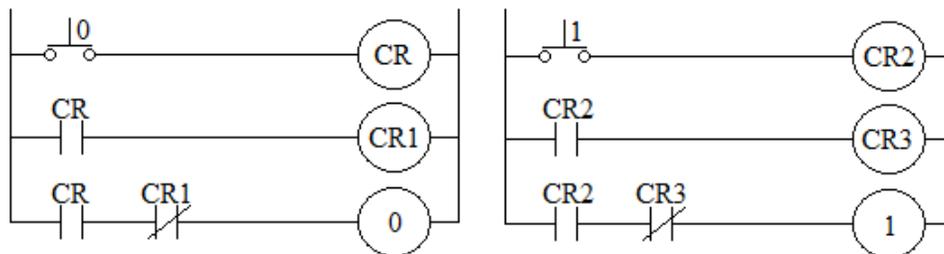


Figura 14.28.: Entradas ejemplo 10

Las bobinas de activación se energizan mediante la combinación de los estados precedentes y contactos de bobinas que permiten la transición hacia el estado siguiente, y su enegización se logra mediante un contacto *NC* de las bobinas de desactivación de cada estado. Lo anterior se observa en la figura 14.29.

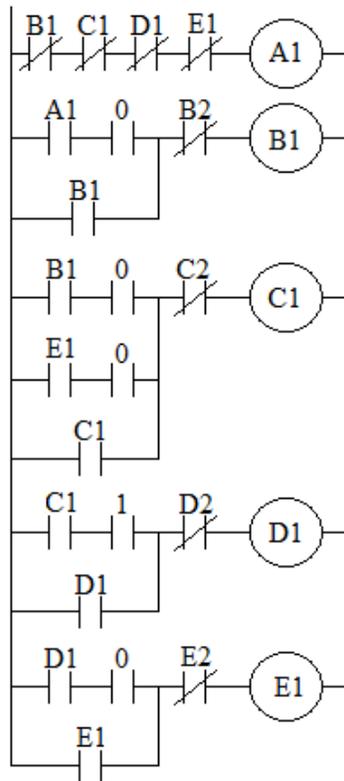


Figura 14.29.: Bobinas de activación ejemplo 10

Las bobinas de desactivación son las encargadas de desenergizar cada uno de los estados y se energizan mediante contactos *NA* de las bobinas de activación de cada estado y de sus correspondientes transiciones. La figura 14.30 muestra lo expresado anteriormente.

La energización de las bobinas de estados se hace mediante contactos *NA* de las bobinas de activación, además de garantizar la energización de un único estado en un momento dado con la incorporación de contactos *NC* de las bobinas de activación de los estados restantes. Lo anterior se muestra en la figura 14.31.

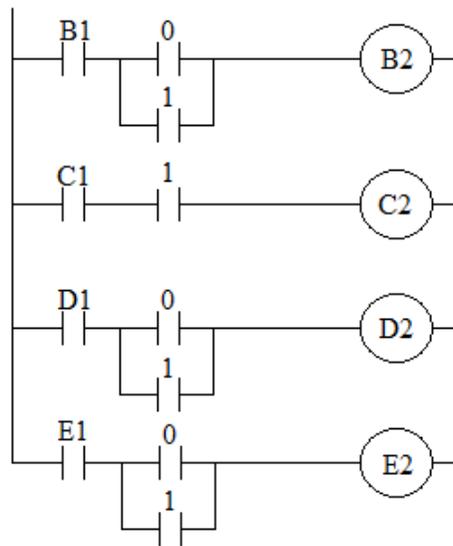


Figura 14.30.: Bobinas de desactivación ejemplo 10

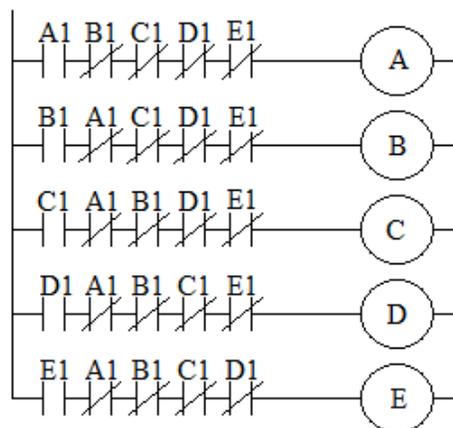


Figura 14.31.: Bobinas de estados ejemplo 10

En este diseño se precisa de la utilización de la función contador, descrita en la metodología de diseño por señales de mando para mostrar que se ha ingresado en tres ocasiones la secuencia de entrada correcta, dicha función es controlada por el estado E como se observa en la figura 14.32.

Las salidas para este ejemplo se muestran en la figura 14.33, en ella se observa una bobina SEC, controlada por el estado E, que indica que la secuencia de entrada se ha introducido de manera correcta, y una bobina TRES, controlada por la bobina 3T perteneciente al contador, que registra el momento en el que se ha ingresado de forma correcta la secuencia de entrada en tres ocasiones.

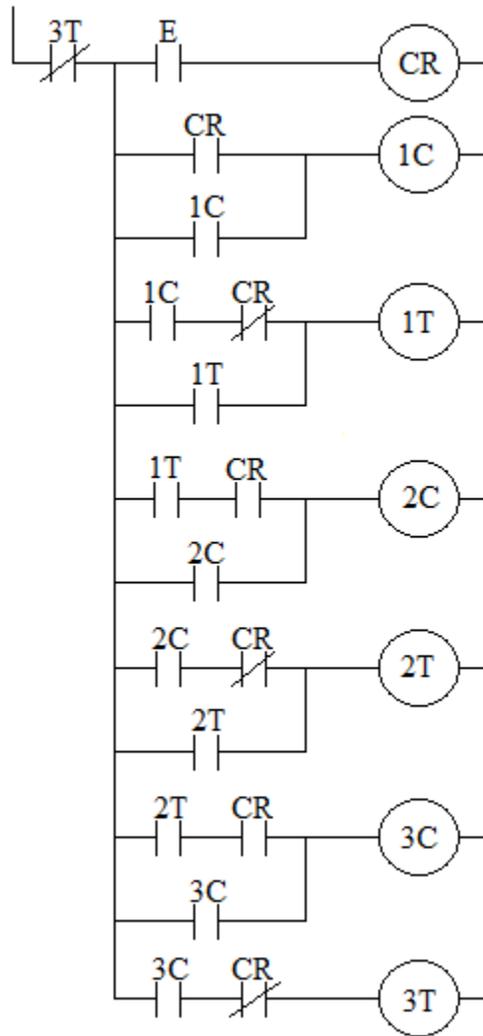


Figura 14.32.: Función contador para secuencia de entrada correcta

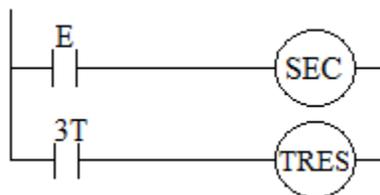


Figura 14.33.: Salidas ejemplo 10





## 15. CONCLUSIONES

- De la presente investigación, cabe resaltar que los automatismos industriales, han tenido una evolución importante en cuanto a su desarrollo e implementación, abarcando desde la utilización de relés para sistemas pequeños hasta la aparición de los PLC usados para sistemas de mayor tamaño. En la búsqueda de soluciones a problemas de automatización mediante diagramas Ladder, no existe una metodología de diseño clara, sino que se realizan diseños basados en prueba y error, lo que demanda un considerable tiempo cuando no se cuenta con una buena experiencia en el tema, lo que se ve reflejado en resultados no satisfactorios. Es por esta razón, que en esta investigación se presentaron dos metodologías de diseño para aquellas personas que no tengan práctica a la hora de dar una solución usando diagramas Ladder a un problema de automatización.
- Se comprobó, que por la metodología de diseño por señales de mando, la solución a los automatismos resulta de combinar funciones, ya que se cuenta con un número determinado de estas, siendo de gran ayuda a la hora de su aplicación para brindar solución a un problema de automatización por diagrama Ladder. Cabe destacar que no todos los sistemas y problemas de diseño se pueden desarrollar por medio de esta metodología, pues resulta un poco compleja cuando son problemas que requieren de alguna realimentación. Por esta razón la metodología de diseño por señales de mando sólo es aplicada a problemas que no requieran este tipo de situaciones.
- Se mostró como el diseño de circuitos Ladder puede asociarse a una máquina de estados, ya que se pueden transformar los enunciados de los problemas de automatización en situaciones claramente definidas, que para este caso equivalen a los estados, mientras que las variaciones en las entradas, representan las transiciones que son las que permiten el cambio de estado en la máquina. Es importante destacar que los diseños se realizaron pensando siempre en la utilización de simuladores que permitieran constatar la validez de estos.
- Al momento de realizar un diseño de mayor dificultad, resulta ser útil desarrollar un modelo híbrido que una las dos metodologías de diseño, en el que se mezclen las entradas, bobinas de los estados y salidas descritas en la metodología de diseño por máquinas de estados y las funciones, ya sean básicas o derivadas mostradas en la metodología de diseño por señales de mando.



# Bibliografía

- [1] BOLTON, W. Programmable Logic Controllers: Programming. Cuarta Edición. Reino Unido. Editorial Newnes. 2003. 288 Páginas. Oxford University. ISBN 0-7506-5986-6.
- [2] HOLGUIN, Londoño Mauricio. Automatismos Industriales: Fundamentos de los automatismos. Pereira. 2008. 254 Páginas. Universidad Tecnológica De Pereira. ISBN 978-958-8272-99-3.
- [3] ROMERAL, José Luis. Autómatas Programables: Automatización. España. Serie Mundo Electrónico. 1997. 439 Páginas. Universidad de Cataluña. ISBN 84-267-1089-1.
- [4] GAMIZ, Caro Juan. Introducción a Los Autómatas Programables: Autómatas Programables. Primera Edición. España. Editorial UOC. 2003. 300 Páginas. ISBN 74-8429-027-1.
- [5] KUPHALDT, Tony R. Lesson In Electric Circuit: Ladder Logic. Vol. IV. Noviembre 01. 2007.
- [6] CREUS, Solé Antonio. Fiabilidad Y Seguridad De Procesos Industriales: Fiabilidad De Sistemas. Segunda Edición. España. Editorial Marcombo S.A. 2005. 470 Páginas. Universidad de Málaga. ISBN 84-267-1362-9.
- [7] GARCIA, Moreno Emilio. Automatización De Procesos Industriales: Introducción A Los Automatismos. Edición Ceres-UPV. España. Servicio De Publicaciones Camino De Vera. 1999. 377 Páginas. Universidad Politécnica De Valencia. ISBN 84-7721-759-9.
- [8] TOCCI, Ronald J. Sistemas Digitales Principios y Aplicaciones: Compuertas Lógicas y Algebra Booleana. Octava Edición. México. Pearson Educación. 2003. 833 Páginas. Universidad de Grenoble. ISBN 970-26-0297-1.
- [9] NELSON, Víctor P. Análisis Y Diseño De Circuitos Lógicos Digitales: Circuitos de conmutación. Edición Ilustrada. Texas. Pearson Prentice Hall. 1996. 842 Páginas. Universidad De Arburn. ISBN 968-880-706-0.
- [10] MANDADO Pérez Enrique. Sistemas Electrónicos Digitales: Sistemas Combinacionales. Novena Edición. España. Alfaomega Grupo Editor. 2008. 883 Páginas. Universidad De Cataluña. ISBN 978-970-15-1304-0.

- [11] WAKERLY, John F. Diseño Digital Principios Y Prácticas: Principios Y Diseño De Lógica Secuencial. Tercera Edición. México. Pearson Educación. 2001. 946 Páginas. Stanford University. ISBN 970-17-0404-5.
- [12] ROTH, Charles H. Fundamentos de Diseño Lógico: Biestables y Latches, Método de Quine-McCluskey. Quinta Edición. México. Editorial Thomson. 2005. 700 Páginas. ISBN 970-686-373-7.
- [13] MORRIS, Mano M. Diseño Digital: Lógica Secuencial. Tercera Edición. México. Pearson Education. 2003. 521 Páginas. Universidad Autónoma de Guadalajara. ISBN 970-26-0438-9.
- [14] PERLOFF, Marjorie. Wittgenstein's Ladder: The Making Of The Tractatus Russell, Wittgenstein, And The Logic Of War. Chicago. 1996. 306 Páginas. The University Of Chicago. ISBN 0-226-66058-3.
- [15] DOMINGO, Peña Joan. Diseño Y Aplicaciones Con Autómatas Programables: Técnicas Y Diseño. Primera Edición. España. Editorial UOC. 2003. 358 Páginas. ISBN 978-84-8429-029-2.
- [16] DELHAYE, C. La Concepción Lógica de Automatismos Industriales: Relés Electromecánicos – Relés electrónicos. Única Edición. España. Marcombo S.A. 1971. 176 Páginas. ISBN 26.676-1968.
- [17] ALEGRE, Gil Carmen. Problemas de Matemática Discreta: Introducción a la Lógica de Predicados. España. Editorial Camino de Vera. 1997. 524 Páginas. Universidad Politécnica de Valencia. ISBN 84-7721-495-6.
- [18] FERRANDO, J C. Matemática Discreta: Lógica de Predicados y Sistemas Formales. Segunda Edición. Barcelona. Editorial Reverte. 2008. 326 Páginas. ISBN 84-291-5179-6.
- [19] PARDO, Collantes Daniel. Fundamentos de Electrónica Digital: Algebra de Boole Y Algebra Binaria. Primera Edición. España. 2006. 276 Páginas. Universidad de Salamanca. ISBN 978-84-7800-401-0.
- [20] GINZBURG, M C. Introducción a Las Técnicas Digitales Con Circuitos Integrados: Algebra de Boole Postulados Y Propiedades. Segunda Edición. Barcelona. Editorial Reverté. 2002. 505 Páginas. ISBN 84-291-3456-5.
- [21] MARTIN, Gonzales José Luis. Electrónica Digital: Funciones Booleanas. España. Editorial Delta. 2006. 386 Páginas. ISBN 84-96477-44-4.
- [22] PATTERSON, David A. Estructura y Diseño de Computadores: Lógica Combinatoria. Volumen 3. Barcelona. Editorial Reverte. 2000. 192 Páginas. Universidad Politécnica de Cataluña. ISBN 84-291-2619-8.

[23] ESTEVE, Bosch Raúl. Fundamentos de Electrónica Digital: Simplificación de Expresiones Lógicas. Primera Edición. Valencia. Editorial Universidad Politécnica de Valencia. 2004. 385 Páginas. Universidad Politécnica de Valencia. ISBN 84-9705-784-8.

[24] Máquinas de Estados Finitos. <http://ing.de.soft1.googlepages.com/IntroduccionAFSMs.pdf>