

IPICYT

**INSTITUTO POTOSINO DE INVESTIGACIÓN
CIENTÍFICA Y TECNOLÓGICA, A.C.**

POSGRADO EN CONTROL Y SISTEMAS DINÁMICOS

**Diseño e implementación de arreglos de compuertas dinámicas
programables en campo y su aplicación en redes Booleanas**

Tesis que presenta

Roberto Rafael Rivera Durón

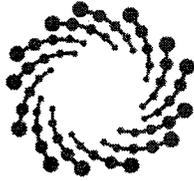
Para obtener el grado de

Doctor en Control y Sistemas Dinámicos

Director de la Tesis:

Dr. Eric Campos Cantón

San Luis Potosí, S.L.P., Abril de 2018



IPICYT

Constancia de aprobación de la tesis

La tesis *“Diseño e implementación de arreglos de compuertas dinámicas programables en campo y su aplicación en redes Booleanas”* presentada para obtener el Grado de Doctor en Control y Sistemas Dinámicos fue elaborada por **Roberto Rafael Rivera Durón** y aprobada el veintitrés de abril del dos mil dieciocho por los suscritos, designados por el Colegio de Profesores de la División de Matemáticas Aplicadas del Instituto Potosino de Investigación Científica y Tecnológica, A.C.

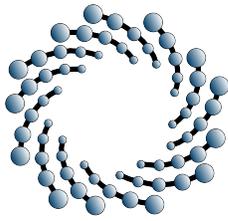
Dr. Eris Campos Cantón
Director de la tesis

Dr. Haret-Codratian Rosu Barbus
Jurando en el Examen

Dr. Diego Langarica Córdoba
Jurando en el Examen

Dr. Guillermo Huerta Cuéllar
Jurando en el Examen

Dr. Luis Javier Ontañón García Pimentel
Jurando en el Examen



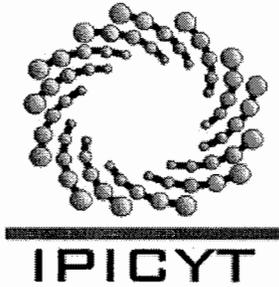
IPICYT

Créditos Institucionales

Esta tesis fue elaborada en la División de Matemáticas Aplicadas del Instituto Potosino de Investigación Científica y Tecnológica, A.C., bajo la dirección del Dr. Eric Campos Cantón.

Durante la realización del trabajo el autor recibió una beca académica del Consejo Nacional de Ciencia y Tecnología No. de registro CONACYT-262196 y del Instituto Potosino de Investigación Científica y Tecnológica, A. C.

Adicionalmente, se recibió apoyo por parte del Fondo Sectorial de Innovación Secretaría de Economía-CONACYT con número de proyecto 215812 y por parte del CONACYT a través del financiamiento al proyecto de ciencia básica con número 181002.



Instituto Potosino de Investigación Científica y Tecnológica, A.C.

Acta de Examen de Grado

El Secretario Académico del Instituto Potosino de Investigación Científica y Tecnológica, A.C., certifica que en el Acta 004 del Libro Primero de Actas de Exámenes de Grado del Programa de Doctorado en Control y Sistemas Dinámicos está asentado lo siguiente:

En la ciudad de San Luis Potosí a los 23 días del mes de abril del año 2018, se reunió a las 12:00 horas en las instalaciones del Instituto Potosino de Investigación Científica y Tecnológica, A.C., el Jurado integrado por:

Dr. Haret-Codratian Rosu Barbus	Presidente	IPICYT
Dr. Eric Campos Cantón	Secretario	IPICYT
Dr. Diego Langarica Córdoba	Sinodal	IPICYT
Dr. Guillermo Huerta Cuéllar	Sinodal externo	UdeG
Dr. Luis Javier Ontañón García Pimentel	Sinodal externo	UASLP

a fin de efectuar el examen, que para obtener el Grado de:

DOCTOR EN CONTROL Y SISTEMAS DINÁMICOS

sustentó el C.

Roberto Rafael Rivera Durón

sobre la Tesis intitulada:

Diseño e implementación de arreglos de compuertas dinámicas programables en campo y su aplicación en redes Booleanas

que se desarrolló bajo la dirección de

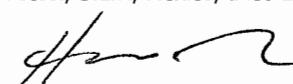
Dr. Eric Campos Cantón

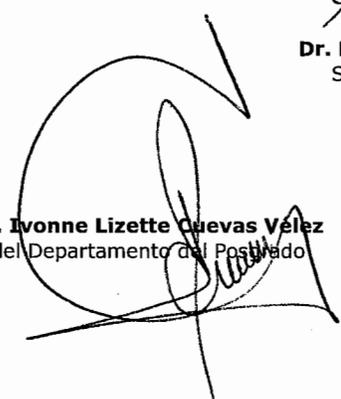
El Jurado, después de deliberar, determinó

APROBARLO

Dándose por terminado el acto a las 13:30 horas, procediendo a la firma del Acta los integrantes del Jurado. Dando fe el Secretario Académico del Instituto.

A petición del interesado y para los fines que al mismo convengan, se extiende el presente documento en la ciudad de San Luis Potosí, S.L.P., México, a los 23 días del mes de abril de 2018.


Dr. Horacio Flores Zúñiga
Secretario Académico


Mtra. Ivonne Lizette Cuevas Vélez
Jefa del Departamento del Posgrado



Dedico esta tesis a mi madre y mi hermana
por su apoyo incondicional.

Agradecimientos

En estas líneas quisiera expresar mi más sincero agradecimiento a quienes se sumaron a esta causa, y me ofrecieron un buen consejo, una mano de ayuda o una crítica objetiva que ayudaran a mi formación tanto personal como profesional.

Primeramente, a mi madre, por todo su apoyo y por ser la estrella polar que me recuerda el camino que debo seguir. A mi hermana, por ser un gran ejemplo para mí. Gran parte de que este objetivo se haya logrado es gracias a ustedes.

A mi asesor, el Dr. Eric Campos Cantón, por toda la confianza depositada en mí y por compartir su conocimiento conmigo.

Al Dr. Barajas y al Dr. Bonilla, quienes fungieron como parte de mi comité tutorial, por estar al tanto de mi desempeño académico y por sus comentarios acertados durante los avances de tesis.

A los doctores Haret Rosu, Guillermo Huerta, Diego Langarica y Luis Javier Ontañón, por haber aceptado ser miembros del jurado y por sus invaluable comentarios acerca de este trabajo.

A los investigadores de la División de Matemáticas aplicadas, por contribuir a mi formación. Al grupo de investigación de Dinámica no lineal y caos, por sus consejos y enseñanzas durante los seminarios.

Al CONACYT, por la beca académica otorgada con número 262196 y por el financiamiento al proyecto de ciencia básica con número 181002. Al Fondo Sectorial de Innovación Secretaría de Economía-CONACYT, por el apoyo económico recibido con número de proyecto 215812.

Al IPICYT y su personal, por permitirme realizar mis estudios de doctorado. A mis compañeros y amigos del instituto, por su compañía, apoyo y por haber creado el ambiente propicio para desarrollar esta investigación.

A mis amigos de toda la vida, por el apoyo incondicional de siempre y por estar al pendiente de mí.

Lista de tablas

2.1	Tabla de verdad de la función Booleana AND.	11
2.2	Tabla de verdad de la función Booleana OR.	12
2.3	Tabla de verdad de la función Booleana NOT.	12
2.4	Tabla de verdad de la función Booleana NAND.	13
2.5	Tabla de verdad de la función Booleana NOR.	13
2.6	Tabla de verdad de la función Booleana XOR.	14
2.7	Tabla de verdad de la función Booleana XNOR.	14
2.8	Tabla de verdad de la función F	15
5.1	Valores de $h(\mathbf{x})$ de la ecuación (5.1), y tablas de verdad para las compuertas lógicas AND, OR, NAND, NOR y XOR correspondientes a la salida Y en la ecuación (5.2).	38
5.2	Valores del parámetro b requeridos para la obtención de las diferentes compuertas lógicas.	42
5.3	Combinación en los interruptores para la selección de la funcionalidad de cada compuerta lógica dinámica.	43
5.4	Tablas de verdad de las funciones Booleanas NAND y NOR.	50
6.1	Línea de tiempo sobre el estudio del fenómeno de sincronización.	58
6.2	Tabla de verdad para la función \mathcal{F} obtenida a partir de la ecuación (6.18).	65
6.3	Valores de los retardos τ_{ij} para la simulación de dos ABNs acopladas unidireccionalmente.	66
6.4	Valores de los retardos τ_{ij} para la simulación de dos ABNs acopladas bidireccionalmente.	71
6.5	Valores de los parámetros para generar los diferentes retardos τ_{ij} en los enlaces.	76
6.6	Valores de parámetros para generar los diferentes retardos τ_{ij} en los enlaces para dos redes experimentales.	81

Lista de figuras

2.1	Mapa de Karnaugh para la función $F = BC + AC' + AB + BCD$	17
3.1	Clasificación de los sistemas dinámicos.	22
3.2	Evolución temporal del sistema conservativo $x(t) = x'(t - 1)$	24
3.3	Evolución temporal del sistema disipativo $x(t) = (x(t - 1) + x'(t - \frac{1}{3}))'$	25
4.1	Arquitectura básica de un FPGA.	32
5.1	Arquitectura básica de un FPDGA.	37
5.2	Intervalo de valores de b para emular la función lógica AND.	39
5.3	Intervalo de valores de b para emular la función lógica OR.	39
5.4	Intervalo de valores de b para emular la función lógica NAND.	40
5.5	Intervalo de valores de b para emular la función lógica NOR.	40
5.6	Intervalo de valores de b para emular la función lógica XOR.	41
5.7	Circuito electrónico de las compuertas lógicas dinámicas.	42
5.8	Diagrama de bloques de la implementación de la selección del parámetro de control.	44
5.9	Diagrama esquemático de la implementación de la selección de la funcionalidad de una compuerta lógica dinámica.	45
5.10	Diagrama de bloques de la implementación de la interconexión programable para una compuerta lógica reconfigurable.	47
5.11	Diagrama esquemático de la implementación de la interconexión programable para una compuerta lógica reconfigurable.	48
5.12	Implementación electrónica de los bloques del FPDGA.	51
5.13	Integración de los bloques del FPDGA.	52
5.14	Validación de las compuertas lógicas dinámicas.	53
6.1	Esquema de un par de sistemas con acoplamiento unidireccional.	59
6.2	Esquema de un par de sistemas con acoplamiento bidireccional.	59
6.3	Esquema de un par de sistemas forzados por señal externa.	59
6.4	Sincronización en fase.	61
6.5	Sincronización en antifase.	61
6.6	Red Booleana autónoma base.	62
6.7	Acoplamiento unidireccional entre un par de redes Booleanas autónomas.	64
6.8	Mapa de Karnaugh para la función \mathcal{F} de la ecuación (6.18).	66
6.9	Ajuste de los estados $x_1^M(t)$ y $x_1^S(t)$ por la acción directa de f	67
6.10	Ajuste de los estados $x_2^M(t)$ y $x_2^S(t)$ por la acción indirecta de f	67
6.11	Ajuste de los estados $x_3^M(t)$ y $x_3^S(t)$ por la acción indirecta de f	68

6.12	Acoplamiento bidireccional entre dos redes Booleanas autónomas.	69
6.13	Ajuste de los estados $x_1^1(t)$ y $x_1^2(t)$ por la acción de f_1	71
6.14	Ajuste de los estados $x_3^1(t)$ y $x_3^2(t)$ por la acción de f_2	72
6.15	Ajuste de los estados $x_2^1(t)$ y $x_2^2(t)$ por la acción indirecta de f_1 y f_2	73
6.16	Esquema de forzamiento por señal externa para k redes Booleanas autónomas.	74
6.17	Red Booleana autónoma con dinámica compleja de 5 nodos.	74
6.18	Evolución temporal en una ABN simulada medida en la salida de tres nodos.	77
6.19	Dos redes Booleanas autónomas controladas por una señal externa S_e	78
6.20	Evolución temporal de los voltajes de dos ABNs forzadas realizadas numéricamente.	80
6.21	Evolución temporal de los errores entre nodos correspondientes de dos ABNs forzadas y simuladas numéricamente.	82
6.22	Evolución temporal de una ABN experimental medida en las salidas de tres de sus nodos.	83
6.23	Variables Booleanas asociadas a V	83
6.24	Logaritmo de la distancia Booleana como una función del tiempo.	84
6.25	Evolución temporal de los voltajes de dos ABNs forzadas y realizadas experimentalmente.	85
6.26	Evolución temporal de los errores entre nodos correspondientes de dos ABNs realizadas experimentalmente.	86
A.1	Póster para el LVII Congreso Nacional de la Sociedad Mexicana de Física.	99
A.2	Póster para la 17a. Feria de Posgrados de Calidad.	100
A.3	Constancia de participación en el XLVIII Congreso Nacional de la Sociedad Matemática Mexicana.	100
A.4	Artículo de investigación publicado.	101

Índice general

Resumen	xxi
Abstract	xxiii
1 Introducción	1
1.1 Motivación e hipótesis	3
1.1.1 Motivación	3
1.1.2 Hipótesis	3
1.2 Objetivo general y objetivos particulares	3
1.3 Contribución de la tesis	4
1.4 Estructura de la tesis	5
2 Álgebra Booleana	7
2.1 Breve reseña histórica	7
2.2 Definición axiomática del álgebra Booleana	8
2.3 Teoremas básicos y propiedades del álgebra Booleana	9
2.4 Funciones Booleanas básicas	11
2.4.1 Función AND	11
2.4.2 Función OR	11
2.4.3 Función NOT	12
2.5 Lógica combinacional	12
2.5.1 Función NAND	12
2.5.2 Función NOR	13
2.5.3 Función XOR	13
2.5.4 Función XNOR	13
2.6 Representaciones algebraicas de funciones Booleanas	14
2.7 Simplificación de funciones Booleanas	16
2.7.1 Simplificación por manipulación algebraica	16
2.7.2 Simplificación por mapas de Karnaugh	16
3 Redes Booleanas y ecuaciones Booleanas con retardo	19
3.1 Redes Booleanas	19
3.1.1 Introducción a las redes Booleanas	19
3.1.2 Definición de red Booleana	20
3.1.3 Clasificación de las redes Booleanas	21
3.2 Ecuaciones Booleanas con retardo	21

3.2.1	Introducción a las BDEs	21
3.2.2	Definición formal de BDEs	23
3.2.3	Propiedades importantes en la BDEs	23
3.2.4	Clasificación de las BDEs	24
3.2.5	Comportamiento asintótico	24
4	Cómputo reconfigurable	27
4.1	Introducción	27
4.2	Definición de cómputo reconfigurable	28
4.3	Desafíos en el diseño de dispositivos reconfigurables	29
4.4	Campos de aplicación del cómputo reconfigurable	29
4.5	Evolución de los sistemas de cómputo reconfigurable	30
4.6	Trabajos anteriores enfocados a la obtención de dispositivos reconfigurables	32
5	Diseño e implementación de un FPDGA	35
5.1	Motivación, hipótesis y propuesta	35
5.1.1	Hipótesis	35
5.1.2	Propuesta	36
5.2	FPDGA: Arreglo de compuertas dinámicas programables en campo	36
5.3	Consideraciones de diseño del FPDGA	36
5.4	Compuertas lógicas dinámicas	37
5.4.1	Modelo matemático de la compuerta lógica dinámica	37
5.4.2	Implementación electrónica de las compuertas lógicas dinámicas	41
5.5	Estructura de ruteo programable	43
5.5.1	Selección de la funcionalidad de las compuertas lógicas dinámicas	43
5.5.2	Interconexión programable de compuertas	45
5.6	Bloque de entrada	46
5.7	Bloque de salida	46
5.8	Implementación electrónica de las partes que conforman el FPDGA	48
5.8.1	Fuentes de voltaje de uso general	48
5.8.2	Fuente de voltaje de uso específico	49
5.8.3	Compuertas lógicas dinámicas	49
5.8.4	Multiplexores analógicos	49
5.8.5	Multiplexores digitales	49
5.8.6	Interruptores	49
5.8.7	Bloques de entrada y bloques de salida	49
5.9	Integración y validación del FPDGA	50
6	Sincronización en redes Booleanas autónomas	55
6.1	Visión general del fenómeno de sincronización	55
6.2	Reseña histórica	56
6.3	Sincronización desde la perspectiva de sistemas dinámicos	57
6.4	Tipos de sistemas acoplados	57
6.5	Tipos de sincronización	60
6.5.1	Sincronización completa	60

6.5.2	Sincronización de fase	60
6.5.3	Sincronización con retardo o <i>lag</i>	61
6.5.4	Sincronización generalizada	61
6.6	Fenómeno de sincronización en redes Booleanas autónomas	62
6.6.1	Sincronización de redes Booleanas autónomas en acoplamiento unidireccional	63
6.6.2	Sincronización de redes Booleanas autónomas en acoplamiento bidireccional	68
6.6.3	Sincronización de redes Booleanas forzadas por una señal externa	73
7	Conclusiones y trabajo a futuro	87
7.1	Conclusiones	87
7.2	Trabajo a futuro	89
	Bibliografía	91
	Apéndice A Productividad	99

Resumen

A menudo, en el estudio de redes Booleanas es deseable poder llevar a cabo su realización electrónica. Con los dispositivos electrónicos comercialmente disponibles, este objetivo se puede lograr de dos maneras diferentes: mediante el uso de compuertas lógicas comerciales o por medio de un arreglo de compuertas programables en campo (FPGA). Sin embargo, cada una de estas metodologías presenta ciertos inconvenientes. Para superar estos inconvenientes, en esta tesis se diseña e implementa un arreglo de compuertas dinámicas programables en campo (FPDGA), el cual es un dispositivo reconfigurable conformado de compuertas lógicas dinámicas cuya funcionalidad e interconexión pueden ser configuradas por el usuario.

La teoría matemática que sustenta el funcionamiento dinámico de las compuertas del FPDGA es desarrollada en este trabajo de tesis y consiste en un sistema no lineal. Con esta teoría, se logra que cada compuerta sea capaz de emular cinco operaciones lógicas. También se propone una forma de crear la estructura de ruteo programable que interconecta las compuertas unas con otras. La múltiple funcionalidad de las compuertas, en conjunto con la estructura de ruteo programable, representa una ventaja sobre los dispositivos reconfigurables existentes debido a que la ejecución de una función lógica compleja puede realizarse de manera directa.

Aunado al diseño e implementación del FPDGA, en esta tesis es presentado un estudio analítico, numérico y experimental del fenómeno de sincronización entre un par de redes Booleanas autónomas acopladas bajo tres esquemas diferentes (unidireccional, bidireccional y forzamiento por una señal externa). Mediante un análisis basado en ecuaciones Booleanas con retardo, se calcula una señal de acoplamiento que permite a las redes alcanzar un estado de sincronización completa en cada uno de los esquemas utilizados. Estos resultados teóricos son corroborados por medio de simulaciones numéricas. En el caso del forzamiento por señal externa, los resultados también son verificados de manera experimental utilizando el FPDGA.

Abstract

Often, in the study of Boolean networks it is desirable to be able to carry out their electronic realization. With commercially available electronic devices, this objective can be achieved in two different ways: through the use of commercial logic gates or through a Field-Programmable Gate Array (FPGA). However, each of these methodologies has certain disadvantages. To overcome these disadvantages, in this thesis is designed and implemented a Field-Programmable Dynamical Gate Array (FPDGA), which is a reconfigurable device consisting of dynamical logic gates whose functionality and interconnection can be configured by the user.

The mathematical theory that supports the dynamical functioning of the FPDGA's gates is developed in this thesis work and consists of a non-linear system. With this theory, each gate is able to emulate five logical operations. A way to create the programmable routing structure that interconnects gates with each other is also proposed. The multiple functionality of gates, in conjunction with the programmable routing structure, represents an advantage over existing reconfigurable devices because the execution of a complex logic function can be performed directly.

In addition to the design and implementation of FPDGA, this thesis presents an analytical, numerical and experimental study of the phenomenon of synchronization between a pair of autonomous Boolean networks coupled under three different schemes (unidirectional, bidirectional and forcing by an external signal). By means of an analysis based on Boolean delay equations, a coupling signal is calculated which allows the networks to achieve a complete synchronization state in each of the schemes used. These theoretical results are corroborated by means of numerical simulations. In the case of forcing by external signal, the results are also verified experimentally using the FPDGA.

CAPÍTULO 1

Introducción

El mundo real es rico en conexiones e interacciones, es decir, no está constituido de sistemas aislados sino que es una vasta colección de agentes interrelacionados entre sí cuyo comportamiento y propiedades pueden ser muy complejos. A este conjunto de objetos interactuando se le da el nombre de red. Las cadenas alimentarias, los grupos de colaboración científica y el sistema nervioso del gusano *C. elegans* son ejemplos de redes a diferente escala encontradas en el mundo real.

Desde el punto de vista de teoría de redes, las redes del mundo real pueden ser representadas por una colección de nodos conectados mediante enlaces los cuales simbolizan la influencia que un nodo tiene sobre otro. Este tipo de redes son inherentemente difíciles de entender debido a que presentan alguna de las siguientes características [1]:

- Complejidad estructural. El diagrama de conexiones puede ser muy entramado.
- Evolución de la red. El diagrama de conexiones puede ir cambiando a través del tiempo.
- Diversidad de conexión. Los enlaces entre nodos pueden tener diferentes pesos, direcciones y signos.
- Complejidad dinámica. Los nodos por sí mismos pueden tener una dinámica complicada, por ejemplo, cada nodo puede ser un sistema dinámico no lineal.
- Diversidad de nodos. Los nodos pueden ser de diferentes tipos.
- Meta-complejidad. Pueden estar presentes en la red varias de las complicaciones anteriormente mencionadas.

Las interacciones exactas que presentan los sistemas del mundo real la mayoría de las veces son desconocidas, por lo tanto, el análisis de redes complejas requiere de modelos más simples, pero lo suficientemente ricos que modelen el comportamiento complejo de los

sistemas del mundo real. Con esta finalidad, la comunidad científica ha recurrido a simplificaciones para suprimir algunas de las complicaciones antes mencionadas. Una de esas simplificaciones son las redes Booleanas, las cuales son redes en las que el estado de sus nodos sólo puede adoptar un par de valores.

Las redes Booleanas fueron introducidas por Kauffman en el año 1969. En [2], Kauffman propone una red Booleana aleatoria como un modelo para las redes de regulación genética. En este modelo cada nodo está asociado con una variable de estado binario, determinada por una función lógica, lo que se puede entender como la definición más general de una red Booleana. A partir de este trabajo, las redes Booleanas se han convertido en herramientas poderosas para describir, analizar y simular redes celulares [3]. Además, se puede decir que el trabajo de Kauffman inspiró a muchos investigadores, no sólo del área de biología, sino de diferentes áreas como física, ciencias de la computación a estudiar las propiedades y aplicaciones de las redes Booleanas.

Las redes Booleanas han sido utilizadas desde dos enfoques diferentes. Por un lado, se les han empleado en el modelado de procesos en los que es útil asumir que las variables de estado del sistema toman valores binarios determinados por funciones Booleanas y que la transición de estado se realiza de manera síncrona como sucede por ejemplo, en las redes de regulación genética. En otras palabras, las redes Booleanas síncronas son aquellas en que la actualización de todos los estados de los nodos se realiza de manera simultánea. El espacio de estados de los modelos síncronos es discreto y finito mientras que el tiempo en que evolucionan es discreto, por lo tanto, estos modelos sólo pueden tener atractores periódicos [4]. Estos modelos síncronos incluyen un proceso externo, parecido a un reloj, que actualiza las variables de estado. Algunos trabajos de redes Booleanas bajo este enfoque pueden ser consultados en [5–8]. Por otro lado, las redes Booleanas han sido útiles herramientas en el modelado de sistemas que tienen realimentaciones en sus estados y que además presentan múltiples retardos en tiempo, tal como sucede en muchos sistemas del mundo real donde la información entre elementos se propaga con retardos que pueden ser diferentes para cada enlace. Estos sistemas pueden modelarse como redes Booleanas autónomas, las cuales son redes en las que los nodos actualizan su estado cuando existe una transición Booleana en sus entradas. En este tipo de redes, el comportamiento futuro es determinado por la especificación de los tiempos precisos en que las transiciones ocurrieron, por lo tanto el espacio de estados es continuo. Las matemáticas que describen a estos sistemas Booleanos autónomos han sido poco desarrolladas, pero se sabe que estos sistemas exhiben patrones aperiódicos si los elementos lógicos tienen tiempos de respuesta instantáneos [4], por esta razón esta tesis se enfoca al estudio de las redes Booleanas autónomas. Algunos trabajos bajo esta perspectiva pueden ser consultados en [9–12].

A menudo, en el estudio de redes Booleanas es deseable poder llevar a cabo su realización electrónica. Este objetivo se puede lograr de dos maneras diferentes:

- Mediante el uso de **compuertas lógicas comerciales**. En este método se utilizan circuitos integrados para la implementación de los nodos, mientras que los enlaces se realizan por medio de segmentos de alambre en una tablilla de prototipos o en su defecto, con segmentos de pista en un circuito impreso. Sin embargo, esta tarea se complica si se están estudiando diferentes topologías de redes Booleanas. Es decir, el trabajo de alambrado aumenta a medida que los nodos o los enlaces de la red varían o

si los nodos cambian su función lógica.

- Por medio de un **arreglo de compuertas programables en campo** (FPGA, por sus siglas en inglés). El FPGA consiste de una matriz bidimensional de bloques configurables los cuales pueden ser conectados mediante recursos de interconexión, de tal manera que el FPGA puede ser programado para implementar diferentes tareas. Sin embargo, debido a que los bloques lógicos que lo conforman son estáticos, para ejecutar una función lógica compleja es necesario poner todo en función de los bloques lógicos preconstruidos, esto implica que el FPGA requiere un cierto tiempo para su re-cableado, lo cual se ve reflejado en la velocidad que puede alcanzar. Además, el FPGA es una máquina de estados finitos en la cual un conjunto de reglas determina la transición de un estado hacia el siguiente cada ciclo de reloj, por esta razón solamente redes Booleanas síncronas pueden ser realizadas en este dispositivo, siendo imposible la realización de una red Booleana autónoma.

1.1. Motivación e hipótesis

1.1.1. Motivación

La motivación de esta tesis surge precisamente al analizar las desventajas que presentan cada una de las formas de implementar redes Booleanas en forma experimental, particularmente, aquellos inconvenientes relacionados con el FPGA. La pregunta por responder es, ¿se puede superar la limitante que impone la arquitectura estática en la próxima generación de arreglos de compuertas configurables? Para dar respuesta a este cuestionamiento, en este trabajo de tesis se partió del supuesto de que una nueva arquitectura con estructura dinámica puede ser la solución a las limitantes de los sistemas actuales. En la arquitectura dinámica los sistemas pueden cambiar con flexibilidad su configuración. Por lo tanto, la idea consiste en utilizar un sistema dinámico no lineal para emular diferentes compuertas lógicas ya que éstas son la base del cómputo de propósito general, y una vez obtenidas las compuertas lógicas, integrarlas en un dispositivo programable por el usuario, es decir, para crear un arreglo de compuertas dinámicas programables en campo (FPDGA, por sus siglas en inglés).

1.1.2. Hipótesis

La **dinámica no lineal** puede ayudar a la obtención de compuertas lógicas dinámicas. Estas compuertas junto con conexiones programables son aplicables en la construcción de un arreglo de compuertas dinámicas programables en campo (FPDGA). Este dispositivo puede ser utilizado como base para **la generación de redes Booleanas autónomas y el análisis del fenómeno de sincronización entre ellas**.

1.2. Objetivo general y objetivos particulares

El objetivo general de la tesis consiste en **sincronizar numéricamente y experimentalmente** dos redes Booleanas autónomas. Para validar experimentalmente el fenómeno de

sincronización, se propone **diseñar e implementar** una arquitectura nueva para la creación de un arreglo de compuertas dinámicas programables en campo (FPDGA). Esta arquitectura será obtenida a partir de un sistema no lineal.

Para alcanzar el objetivo general se establecieron los siguientes objetivos particulares:

1. Diseñar e implementar el dispositivo FPDGA.
 - Desarrollar la teoría matemática capaz de emular compuertas lógicas dinámicas a partir de un sistema no lineal.
 - Emular dispositivos de memoria aplicables a las compuertas lógicas dinámicas con el fin de retener su configuración.
 - Emular estructuras de interconexión programable entre compuertas lógicas dinámicas.
 - Emular dispositivos de entrada y salida.
2. Desarrollar modelos matemáticos de redes Booleanas autónomas implementables en el FPDGA.
3. Validar redes Booleanas autónomas en el FPDGA.
4. Sincronizar dos redes Booleanas autónomas mediante diferentes esquemas de acoplamiento.

1.3. Contribución de la tesis

El presente trabajo de tesis logra integrar dos aspectos importantes del quehacer científico: la investigación científica como generadora de conocimiento y la innovación tecnológica a partir del conocimiento.

Con respecto a la investigación científica, esta tesis hace las siguientes contribuciones:

- Presenta un sistema no lineal capaz de emular diferentes compuertas lógicas.
- Presenta un estudio analítico de las redes Booleanas autónomas modeladas a partir de ecuaciones Booleanas con retardo.
- Analiza diferentes topologías de redes Booleanas autónomas y define el grado de complejidad de las mismas por medio del cálculo de su exponente de Lyapunov.
- Acopla dos redes Booleanas autónomas mediante tres esquemas diferentes con la finalidad de lograr su sincronización.
- Da condiciones para la obtención de sincronización idéntica tanto para el esquema de acoplamiento de maestro-esclavo como para el esquema de forzamiento por señal externa.

Mientras que en lo referente a la innovación tecnológica, este trabajo de investigación también contribuye debido a que:

- Expone la metodología para implementar experimentalmente compuertas lógicas dinámicas obtenidas a partir de un sistema no lineal.
- Implementa experimentalmente estructuras de conexión programable.
- Muestra la manera de crear un dispositivo programable que contiene bloques lógicos cuya funcionalidad e interconexión pueden ser configuradas *in situ*.
- Registra en México y en Estados Unidos de América la solicitud de patente del dispositivo creado.
- Logra el otorgamiento de la patente del dispositivo FPDGA en Estados Unidos de América.

Estos últimos dos puntos son de suma importancia debido a que las tendencias globales toman a las solicitudes y el otorgamiento de patentes como indicadores significativos tanto de la actividad innovativa como de la producción de conocimiento de centros de investigación como el IPICYT.

1.4. Estructura de la tesis

El contenido restante de la presente tesis está estructurado de la siguiente manera:

- El Capítulo 2 incluye una reseña histórica y los preliminares teóricos sobre el álgebra de Boole.
- El Capítulo 3 aborda los aspectos generales de las redes Booleanas y las nociones básicas de las ecuaciones Booleanas con retardo útiles para el entendimiento de las redes Booleanas autónomas.
- En el Capítulo 4 se presenta un panorama general del estado del arte del hardware reconfigurable.
- El Capítulo 5 trata sobre el diseño e implementación de un arreglo de compuertas lógicas dinámicas programables en campo.
- El Capítulo 6 muestra el estudio numérico y experimental del fenómeno de sincronización de dos redes Booleanas autónomas mediante diferentes esquemas de acoplamiento.
- Finalmente, el Capítulo 7 resume las conclusiones de la tesis y plantea algunas directrices para trabajo a futuro.

CAPÍTULO 2

Álgebra Booleana

Como se mencionó en el capítulo introductorio, esta tesis aborda dos temas aparentemente disjuntos. Por un lado, trata sobre el estudio de redes Booleanas autónomas, particularmente sobre el fenómeno de sincronización entre ellas. Y por otro lado, aborda el diseño e implementación de un dispositivo electrónico reconfigurable capaz de realizar cómputo de propósito general y sobretodo en el cual se puedan programar realizaciones electrónicas de redes Booleanas autónomas. Por la naturaleza propia de las redes Booleanas, las funciones lógicas que rigen el comportamiento de sus estados está basada en el álgebra Booleana. Lo mismo sucede con la parte referente al diseño de dispositivos electrónicos digitales como es el caso del arreglo programable de compuertas dinámicas, ya que los postulados y teoremas del álgebra de Boole son de importancia absoluta en el análisis y diseño de dichos circuitos.

El objetivo de este capítulo es presentar los postulados y propiedades que definen al álgebra de Boole. Además se muestra la forma en que operan las funciones Booleanas básicas y cómo se puede obtener otras funciones más complejas a partir de la lógica combinacional. Finalmente, se explica la simplificación de funciones Booleanas por medio de mapas de Karnaugh.

2.1. Breve reseña histórica

La teoría del álgebra Booleana fue creada en 1854 por el matemático inglés George Boole. En su libro *The laws of thought* introdujo el primer tratado sistemático de lógica y desarrolló para este fin un sistema algebraico conocido hoy en día como álgebra Booleana [13]. La forma del álgebra creada por Boole dista de la versión moderna, la cual proviene del periodo comprendido entre 1864 y 1895 con las contribuciones de William Stanley Jevons, Augustus De Morgan, Charles Sanders Peirce y Ernst Schröder. Los fundamentos del álgebra Booleana como una disciplina algebraica abstracta, axiomatizada por un conjunto de ecuaciones fue dada por Edward Huntington en 1904 [14].

Varios contemporáneos de Boole idearon la forma de crear máquinas que implementaran

funciones Booleanas. William Stanley Jevons construyó un piano lógico en 1869. Posteriormente, el filósofo norteamericano Charles Sanders Peirce sugiere el uso de electricidad para implementar funciones Booleanas en 1886. Paul Ehrenfest propone utilizar el álgebra Booleana para crear interruptores automáticos para teléfonos en 1910, pero es hasta finales de la década de 1930 cuando esta idea fue desarrollada y marcó el inicio de la era digital [15].

Las computadoras a lo largo del tiempo han sido fabricadas con diferentes tipos de componentes. De cierto modo, se puede decir que los interruptores electro-mecánicos utilizados en telefonía fueron los antecesores de las computadoras actuales. Durante las décadas de 1940 y 1950 las computadoras fueron hechas mediante tubos de vacío. Posteriormente, con la invención del transistor, fueron hechas de varios de estos elementos interconectados. Hoy en día, están hechas de chips de semiconductores como silicio, germanio o galio. Sin embargo, independientemente del material con que han sido elaboradas las computadoras, los fundamentos teóricos que hacen posible su funcionamiento siguen siendo los mismos. Todas ellas están basadas en la lógica del álgebra Booleana.

2.2. Definición axiomática del álgebra Booleana

El álgebra de Boole es una estructura algebraica definida para un conjunto de elementos $\mathbb{B} = \{0, 1\}$ junto con dos operadores binarios $+$ y \cdot , de tal forma que se cumplen los postulados de Huntington que a continuación se enumeran [16]:

1. (a) Conjunto cerrado con respecto al operador $+$.
(b) Conjunto cerrado con respecto al operador \cdot .
2. (a) Un elemento de identidad con respecto al operador $+$ designado por el 0; tal que se cumple que $x + 0 = 0 + x = x$.
(b) Un elemento de identidad con respecto al operador \cdot designado por el 1; tal que se cumple que $x \cdot 1 = 1 \cdot x = x$.
3. (a) Conmutativo con respecto al operador $+$, tal que $x + y = y + x$.
(b) Conmutativo con respecto al operador \cdot , tal que $x \cdot y = y \cdot x$.
4. (a) \cdot es distributivo sobre $+$, tal que $x \cdot (y + z) = (x \cdot y) + (x \cdot z)$.
(b) $+$ es distributivo sobre \cdot , tal que $x + (y \cdot z) = (x + y) \cdot (x + z)$.
5. Para cada elemento $x \in \mathbb{B}$, existe un elemento $x' \in \mathbb{B}$, llamado complemento de x , tal que:
 - (a) $x + x' = 1$.
 - (b) $x \cdot x' = 0$.
6. Existen al menos dos elementos $x, y \in \mathbb{B}$ tales que $x \neq y$.

Debe notarse que los postulados de Huntington no incluyen la ley asociativa, pero esta ley es válida para el álgebra de Boole.

2.3. Teoremas básicos y propiedades del álgebra Booleana

Algunos de los postulados anteriores fueron listados en incisos, es decir, contienen una parte (a) y una parte (b). Una parte de los postulados puede ser obtenida a partir de la otra si los operadores binarios y los elementos identidad son intercambiados [16]. Lo anterior está enunciado en el siguiente teorema.

Teorema 2.3.1 Principio de dualidad. *Las expresiones algebraicas deducidas de los postulados del álgebra de Boole permanecen siendo válidas si se intercambian los operadores $+$ y \cdot , y elementos de identidad 0 y 1 .*

A continuación se enumerarán los postulados y teoremas básicos del álgebra de Boole. Los postulados son axiomas básicos de la estructura algebraica y no necesitan demostración, mientras que los teoremas se deben probar a partir de los postulados.

1. Postulado 1: Propiedades de cerradura

$$x + y \in \mathbb{B} = \{0, 1\}, \quad (2.1)$$

$$x \cdot y \in \mathbb{B} = \{0, 1\}. \quad (2.2)$$

2. Postulado 2: Propiedades del elemento identidad

$$0 + x = x, \quad (2.3)$$

$$1 \cdot x = x. \quad (2.4)$$

3. Postulado 3: Leyes conmutativas

$$x + y = y + x, \quad (2.5)$$

$$x \cdot y = y \cdot x. \quad (2.6)$$

4. Postulado 4: Leyes distributivas de $+$ y \cdot

$$x + (y \cdot z) = (x + y) \cdot (x + z), \quad (2.7)$$

$$x \cdot (y + z) = (x \cdot y) + (x \cdot z). \quad (2.8)$$

5. Postulado 5: Leyes del complemento

$$x + x' = 1, \quad (2.9)$$

$$x \cdot x' = 0. \quad (2.10)$$

6. Teorema 1: Propiedades especiales del 0 y 1

$$1 + x = 1, \quad (2.11)$$

$$0 \cdot x = 0. \quad (2.12)$$

7. Teorema 2: Leyes de idempotencia

$$x + x = x, \quad (2.13)$$

$$x \cdot x = x. \quad (2.14)$$

8. Teorema 3: Ley de involución

$$(x')' = x. \quad (2.15)$$

9. Teorema 4: Leyes asociativas

$$(x + y) + z = x + (y + z), \quad (2.16)$$

$$(x \cdot y) \cdot z = x \cdot (y \cdot z). \quad (2.17)$$

10. Teorema 5: Leyes de De Morgan

$$(x + y)' = x' \cdot y', \quad (2.18)$$

$$(x \cdot y)' = x' + y'. \quad (2.19)$$

11. Teorema 6: Leyes de absorción

$$x + (x \cdot y) = x, \quad (2.20)$$

$$x \cdot (x + y) = x, \quad (2.21)$$

$$x + (x' \cdot y) = x + y, \quad (2.22)$$

$$x \cdot (x' + y) = x \cdot y, \quad (2.23)$$

$$(x \cdot y) + (x \cdot y') = x, \quad (2.24)$$

$$(x + y) \cdot (x + y') = x. \quad (2.25)$$

2.4. Funciones Booleanas básicas

El álgebra Booleana es la parte de las matemáticas que concierne a ceros, unos y la manipulación de éstos utilizando operaciones llamadas funciones Booleanas, también conocidas como funciones lógicas. Dentro de esta álgebra existen unos objetos llamados bits o variables binarias que pueden tomar el valor de 0 ó 1, existen además varias funciones que operan sobre los bits para producir diferentes bits. El bit o bits sobre los cuales una función Booleana opera son llamados **bits de entrada**, y el bit que resulta de la aplicación de la función a los bits de entrada se conoce como **bit de salida**. En general, las funciones Booleanas son simples y existen tres de ellas que son básicas: AND, OR y NOT. En este punto es importante mencionar que aunque existe una traducción exacta de cada una de las funciones Booleanas básicas, a lo largo de esta tesis se referirá a ellas en idioma inglés y se les denotará en mayúsculas. Lo anterior con la finalidad de evitar confundirlas con las palabras *y*, *o* y *no*.

2.4.1. Función AND

La función AND toma dos bits de entrada y produce un bit de salida. Y su forma de operar es muy simple, la salida será 1 si y sólo si ambos bits de entrada son 1, en cualquier otro caso la salida será 0.

El símbolo que representa a la función AND es \wedge , aunque también se suele utilizar el símbolo “ \cdot ” o simplemente no se utiliza ningún símbolo entre los dos bits de entrada. Por lo tanto las expresiones $x \wedge y$, $x \cdot y$, xy significan x AND y .

La tabla de verdad de la función AND para todas las combinaciones de entradas se muestra en la Tabla 2.1.

x	y	$x \cdot y$
0	0	0
0	1	0
1	0	0
1	1	1

Tabla 2.1: Tabla de verdad de la función Booleana AND.

2.4.2. Función OR

La función OR toma dos bits de entrada y produce un bit de salida. La forma en que opera es muy simple, la salida será 0 si y sólo si ambos bits de entrada son 0, en cualquier otro caso la salida será 1.

El símbolo que representa a la función OR es \vee , aunque también se suele utilizar el símbolo $+$. De tal modo que las expresiones $x \vee y$, $x + y$ significan indistintamente x OR y .

En la Tabla 2.2 se muestra la tabla de verdad de la función OR para todas las combinaciones de entradas.

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

Tabla 2.2: Tabla de verdad de la función Booleana OR.

2.4.3. Función NOT

La función NOT a diferencia de las funciones AND y OR, es una función unaria, esto significa que opera con un sólo bit. Esta función lo que hace es invertir, complementar o negar bits, es decir, un 1 lo convierte en 0 y un 0 lo convierte en 1.

Los símbolos utilizados para representar a la función NOT son: $'$, $\bar{}$, \neg . Por lo tanto, las expresiones x' , \bar{x} , $\neg x$ significan NOT x .

La tabla de verdad de la función NOT se muestra en la Tabla 2.3. Debe notarse que ésta consta solamente de 2^1 posibles combinaciones debido a que es una función unaria.

x	x'
0	1
1	0

Tabla 2.3: Tabla de verdad de la función Booleana NOT.

2.5. Lógica combinacional

Como se mencionó en la Sección 2.2, el álgebra de Boole por definición consiste de sólo tres operadores: AND, OR y NOT; pero existen además otras funciones Booleanas que son obtenidas a partir de estas funciones básicas y que son conocidas como **lógica combinacional**. Con frecuencia, existen diferentes maneras de combinar las funciones básicas para obtener una misma función compuesta. A cada una de estas distintas formas se les da el nombre de **realizaciones** [15].

A continuación, se presentan algunas de las funciones Booleanas más utilizadas y que aparecen recurrentemente en los capítulos de esta tesis.

2.5.1. Función NAND

La función NAND es una operación de dos bits de entrada que surge de evaluar primeramente una función AND para posteriormente, aplicar la función NOT al bit de salida resultante. Su forma de operar es muy simple, la salida será 0 si y sólo si ambos bits de entrada son 1, en cualquier otro caso la salida será 1.

Las expresiones $\overline{(x \wedge y)}$, $\overline{x \cdot y}$, \overline{xy} , $(x \wedge y)'$, $(x \cdot y)'$, $(xy)'$, $\neg(x \wedge y)$, $\neg(x \cdot y)$, $\neg(xy)$ significan x NAND y . La tabla de verdad de la función NAND para todas las combinaciones de entradas se muestra en la Tabla 2.4.

x	y	$(x \cdot y)'$
0	0	1
0	1	1
1	0	1
1	1	0

Tabla 2.4: Tabla de verdad de la función Booleana NAND.

2.5.2. Función NOR

La función NOR es una operación de dos bits de entrada que surge de evaluar primeramente una función OR y luego, aplicar la función NOT al bit de salida resultante. La salida de esta función será 1 si y sólo si ambos bits de entrada son 0, en cualquier otro caso la salida será 0.

Las expresiones $\overline{(x \vee y)}$, $\overline{(x + y)}$, $(x \vee y)'$, $(x + y)'$, $\neg(x \vee y)$, $\neg(x + y)$ significan x NOR y . La tabla de verdad de la función NOR para todas las combinaciones de entradas se muestra en la Tabla 2.5.

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

Tabla 2.5: Tabla de verdad de la función Booleana NOR.

2.5.3. Función XOR

La función XOR es una operación binaria un tanto más compleja respecto a las enumeradas anteriormente. Ésta surge de la evaluación de la expresión $(x \wedge y') \vee (x' \wedge y)$, de tal manera que su salida será 1 cuando los bits de entrada tengan distinto valor de verdad.

El símbolo para denotar a esta función es \oplus y es único. La tabla de verdad de la función XOR para todas las combinaciones de entradas se muestra en la Tabla 2.6.

2.5.4. Función XNOR

También conocida como equivalencia, coincidencia o NOR-exclusiva, la función XNOR que opera sobre dos bits de entrada y que surge de evaluar primeramente una función XOR

x	y	$x \oplus y$
0	0	0
0	1	1
1	0	1
1	1	0

Tabla 2.6: Tabla de verdad de la función Booleana XOR.

para posteriormente, aplicar la función NOT al bit de salida resultante. La salida de esta función será 1 cuando ambos bits de entrada tengan el mismo valor de verdad.

El símbolo para denotar a esta función es \odot . La tabla de verdad de la función XNOR para todas las combinaciones de entradas se muestra en la Tabla 2.7.

x	y	$x \odot y$
0	0	1
0	1	0
1	0	0
1	1	1

Tabla 2.7: Tabla de verdad de la función Booleana XNOR.

2.6. Representaciones algebraicas de funciones Booleanas

Toda la teoría presentada en esta sección puede ser consultada en [17].

Un **término de producto** se obtiene de utilizar la función AND sobre dos o más variables. Por ejemplo: $AB'C$, $P'QR'$ y XYZ son ejemplos de términos de productos.

Un **término de suma** se obtiene al aplicar la función OR sobre dos o más variables. $A + B' + C$, $P' + Q + R'$ y $X + Y + Z$ son ejemplos de términos de sumas.

Cada ocurrencia de una variable, ya sea en su forma verdadera o en su forma complementada, se le denomina **literal**. El término $AB'C$ tiene tres literales, mientras que el término $A + B' + C + D' + E$ tiene cinco literales.

Un término de producto X se dice que está incluido en otro término de producto Y , si Y tiene cada literal que esté en X . Por analogía, un término de suma X se dice que está incluido en otro término de suma Y , si Y tiene cada literal que esté en X . Por ejemplo, $AB'C$ está incluido en $AB'CD'$, mientras que XYZ no está incluido en $XY'Z$.

Una función Booleana puede expresarse ya sea como una suma de varios términos de producto, conocida como suma de productos o como un producto de varios términos de suma, conocido como producto de sumas. Por ejemplo, $F(A, B, C) = AB + ABC + ABC'$ está en la forma de suma de productos, mientras que $G(A, B, C) = (A+B)(A+B+C)(A+B+C')$ está en la forma de producto de sumas. En los dos ejemplos anteriores las variables de las funciones están enumeradas de forma explícita en el lado izquierdo de la igualdad, sin

embargo, esta enumeración no es necesaria si las variables se pueden determinar a partir del lado derecho de la igualdad.

Si ninguno de los términos de producto de la función expresada en la forma de suma de productos está incluido en los otros términos de productos de la función, se dice que la función está en **forma normal de suma de productos**. Por ejemplo, $F = AB + AC'$, $G = PQS + PQR$, $H = ABC + AB'C + A'BC$ están en la forma normal de suma de productos.

De igual manera, si ninguno de los términos de suma de la función expresada en la forma de producto de sumas está incluido en los otros términos de sumas de la función, se dice que la función está en **forma normal de producto de sumas**. Por ejemplo, $F = (A+B)(A+C')$, $G = (P+Q+S)(P+Q+R)$, $H = (A+B+C)(A+B'+C)(A'+B+C)$ están en la forma normal de producto de sumas.

Una función en la forma suma de productos se dice que está en la forma canónica de suma de productos si cada uno de los términos de productos de la función contiene todas y cada una de las variables componentes, en su forma verdadera o en su forma complementada. Es decir, si la función es de n variables, cada término de producto en la representación canónica tendrá n literales distintas. Por ejemplo, $Q = ABC + A'BC + A'B'C'$ está en la forma canónica de suma de productos, mientras que $Q = XY + X'YZ + XYZ'$ no lo está.

De manera análoga, una función en la forma producto de sumas se dice que está en la forma canónica de producto de sumas si cada uno de los términos de sumas de la función contiene todas y cada una de las variables componentes, en su forma verdadera o en su forma complementada. Es decir, si la función es de n variables, cada término de suma en la representación canónica tendrá n literales distintas. Por ejemplo, $Q = (A + B + C)(A' + B + C)(A' + B' + C')$ está en la forma canónica de producto de sumas, mientras que $Q = (X + Y)(X' + Y + Z)(X + Y + Z')$ no lo está.

Cada término en la forma canónica de suma de productos se denomina **minitérmino**. Cada término en la forma canónica de producto de sumas se denomina **maxitérmino**. Los minitérminos y maxitérminos son los equivalentes decimales de las combinaciones binarias correspondientes de los valores de función 1 y 0, respectivamente. Por ejemplo, considérese la tabla de verdad para F , una función de A , B y C mostrada en la Tabla 2.8.

A	B	C	F
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

Tabla 2.8: Tabla de verdad de la función F .

De la tabla de verdad se puede ver que F es 1 cuando ocurren cada uno de los siguientes casos: $(A = 0, B = 0, C = 0)$, $(A = 1, B = 0, C = 0)$, $(A = 1, B = 0, C = 1)$ y $(A = 1, B = 1, C = 1)$. De tal manera que la función F expresada en la forma canónica

de suma de productos es $F = A'B'C' + AB'C' + AB'C + ABC$. La forma en suma de minitérminos es $F = \sum m(0, 4, 5, 7)$.

2.7. Simplificación de funciones Booleanas

Un punto importante relacionado con las funciones Booleanas es encontrar la forma más simple de las mismas. Existen diferentes formas de simplificar funciones Booleanas pero en esta tesis sólo se ahondará en un par de ellas: la simplificación por manipulación algebraica y la simplificación por mapas de Karnaugh, también conocidos como mapas K.

2.7.1. Simplificación por manipulación algebraica

Este método consiste en utilizar los postulados y teoremas del álgebra de Boole para manipular las expresiones Booleanas con el objeto de llevar al mínimo el número de literales. Sin embargo, la manipulación puede resultar un tanto compleja debido a que no existe un algoritmo que dicte el orden en que ésta se debe realizar.

Por ejemplo, para simplificar la función $F = BC + AC' + AB + BCD$ se requieren hacer los siguientes pasos:

1. Reacomodar la función $F = AC' + AB + BC + BCD$.
2. Utilizar las leyes de absorción del Teorema 6. Se sabe que $BC + BCD = BC$, por lo que $F = AC' + AB + BC$.
3. Aplicar las leyes del complemento del Postulado 5: $F = AC' + AB(C + C') + BC(A + A') = AC' + ABC + ABC' + A'BC$.
4. Por las leyes de absorción del Teorema 6 se sabe que $AC' + AC'B = AC'$ y $BCA' + BCA = BC$, por lo tanto $F = AC' + BC$.

2.7.2. Simplificación por mapas de Karnaugh

Este método es gráfico y se realiza por medio de un diagrama dividido en cuadros. Cada cuadro representa un minitérmino. Ya que cualquier función Booleana puede ser expresada como una suma de minitérminos, es posible representarla gráficamente en el mapa de Karnaugh a partir del área encerrada por aquellos cuadros cuyos minitérminos se incluyen en la función. De hecho, el mapa de Karnaugh presenta en forma gráfica todas las posibles formas en que puede ser expresada una función Booleana en su forma normalizada. La idea principal es encontrar la expresión algebraica más simple, es decir, la que tiene el mínimo número de literales [16].

Un mapa de Karnaugh de n variables contiene 2^n cuadros correspondiendo cada uno a un minitérmino de la función de variables n .

La propiedad principal de los mapas de Karnaugh es que sólo una variable cambia al desplazarse de un cuadro a un cuadro adyacente. Se dice que dos minitérminos son lógicamente adyacentes cuando difieren sólo por una variable. El mapa de Karnaugh transforma

así la adyacencia lógica en adyacencia física de minitérminos. Es decir, los minitérminos que se representan físicamente adyacentes entre sí en el mapa de Karnaugh son también lógicamente adyacentes [17]. Esta propiedad es de suma importancia para la minimización de funciones.

El algoritmo que permite minimizar las funciones Booleanas consiste en:

1. Llenar el mapa de Karnaugh con los minitérminos que se incluyen en la función. Este paso resulta más fácil si se tiene la tabla de verdad de la función a minimizar. Sin embargo, algunas veces es necesario deducir los minitérminos a partir de una función expresada en forma no canónica.
2. Hacer grupos de minitérminos que sean contiguos unos con otros en conjuntos cuya cardinalidad sea potencia de dos, es decir, hacer conjuntos de 1, 2, 4, . . . minitérminos.
3. Por cada grupo hecho, aparecerá un término de producto resultante de eliminar aquellas variables que han cambiado su valor lógico y de anotar solamente aquellas variables que no cambian su valor de verdad.

Como ejemplo se simplificará nuevamente la función $F = BC + AC' + AB + BCD$ mostrada en la Subsección 2.7.1. Los pasos para hacerlo son los siguientes:

Antes de construir el mapa de Karnaugh correspondiente, se debe notar que la función a simplificar está expresada en una forma no canónica. Esto significa que se deben deducir cada uno de los minitérminos que no aparecen explícitamente en la función. Por ejemplo, en el término BC de F no aparecen explícitamente las variables A y D , razón por la cual se debe completar este término de manera que aparezcan todas las variables de la función en todas sus posibles combinaciones de valor de verdad. Por lo tanto, los minitérminos que incluyen al término BC son: $ABCD$, $A'BCD$, $ABCD'$ y $A'BCD'$. Ahora, para el término AC' , los minitérminos que lo incluyen son: $ABC'D$, $AB'C'D$, $ABC'D'$ y $AB'C'D'$. Para el término AB , los minitérminos que lo incluyen son: $ABCD$, $ABC'D$, $ABCD'$ y $ABC'D'$. Finalmente, para el término BCD , los minitérminos que lo incluyen son: $ABCD$ y $A'BCD$.

Una vez que se han deducido todos los minitérminos que hacen verdadera a la función, se procede a llenar el mapa de Karnaugh con dichos minitérminos. El mapa de Karnaugh de la función F se muestra en la Figura 2.1.

		CD			
		00	01	11	10
AB	00	0	0	0	0
	01	0	0	1	1
	11	1	1	1	1
	10	1	1	0	0

Figura 2.1: Mapa de Karnaugh para la función $F = BC + AC' + AB + BCD$.

Posteriormente, se agrupan todos los minitérminos que aparecen en el mapa. Para ello se forman dos conjuntos de cuatro minitérminos cada uno, estos conjuntos están coloreados en azul cielo y azul marino. Para el conjunto coloreado en azul cielo se puede observar que las variables A y D cambian su valor de verdad, por lo tanto éstas no se incluyen en el término resultante, mientras que las variables B y C permanecen con el mismo valor de verdad. En consecuencia, el término resultante asociado al conjunto coloreado en azul cielo es BC . Ahora, para el conjunto coloreado en azul marino, las variables B y D cambian, mientras que A se mantiene en 1 y C se mantiene en 0. Por lo tanto, el término resultante asociado a este conjunto es AC' . De tal modo que la función simplificada queda como $F = BC + AC'$.

Redes Booleanas y ecuaciones Booleanas con retardo

En el capítulo anterior fue dada la teoría elemental del álgebra de Boole útil para el entendimiento tanto de las redes Booleanas como del diseño de sistemas reconfigurables.

El presente capítulo tiene por objeto presentar las nociones básicas referentes al tema de redes Booleanas. Uno de los tópicos a tratar es la clasificación de las redes Booleanas con respecto a la forma en que actualizan sus estados ya que de esto depende la forma en que serán modeladas. Finalmente, debido a que en esta tesis se estudia particularmente el caso de las redes Booleanas autónomas es importante conocer la teoría de ecuaciones Booleanas con retardo, misma que será presentada en este capítulo.

3.1. Redes Booleanas

3.1.1. Introducción a las redes Booleanas

En el estudio de sistemas dinámicos se pueden encontrar sistemas en los cuales sus estados adquieren un gran número de valores. El comportamiento dinámico resultante de estos sistemas puede ser muy complicado. Por eso es importante contar con herramientas que ayuden a simplificar y a comprender el comportamiento de los sistemas dinámicos. Unas de estas simplificaciones o aproximaciones son las redes Booleanas.

En el año 1966, Walker y Ashby se interesan en el estudio del comportamiento de sistemas dinámicos con muchas partes interactuando intrincadamente, los cuales tienen la apariencia de ser lo que posteriormente se conocería como una red Booleana. Sin embargo, ellos no acuñaron este concepto para referirse a su objeto de estudio. Los sistemas presentados por estos autores están contruidos por elementos, los cuales ejecutan funciones lógicas recursivas. Cada elemento tiene dos entradas binarias y un estado interno binario el cual es también la salida del elemento [18].

El concepto de red Booleana fue propuesto originalmente en el año 1969 por Stuart Kauffman para modelar la regulación genérica a nivel celular. El tipo de red propuesto

consiste de N nodos, los cuales pueden tomar valores Booleanos. El estado de cada nodo está determinado por K conexiones de entrada que provienen del mismo o de otros nodos. Las conexiones son hechas de manera aleatoria y permanecen fijas. La manera en que los nodos se afectan unos a otros no solamente es determinada por sus conexiones, sino también por funciones lógicas, que también son asignadas de manera aleatoria y permanecen fijas [2].

Posteriormente, en el año 1984, Dee y Ghil proponen las ecuaciones Booleanas con retardo, también llamadas ecuaciones Booleanas en diferencias, como una forma de modelar aquellos sistemas biológicos y físicos que presentan mecanismos de realimentación que dependen de umbrales asociados con las variables de estado [19]. Un año más tarde, Ghil y Mullhaupt estudian por métodos numéricos, analíticos y algebraicos las ecuaciones Booleanas con retardo. Muestran que las soluciones dependen de la condición inicial y de la naturaleza de los retardos. Si todos los retardos son racionales, entonces el sistema sólo tendrá soluciones periódicas. Pero si cuenta con retardos inconmensurables, entonces tendrá soluciones aperiódicas [20].

En el año 1998, Glass y Hill analizan las transiciones entre la dinámica ordenada y desordenada de redes Booleanas modeladas por medio de ecuaciones diferenciales lineales a pedazos, las cuales cuentan con funciones constantes a pedazos cuyos valores pueden cambiar conforme las variables de estado crucen determinado umbral [21].

Desde su aparición, las redes Booleanas han resultado ser útiles herramientas en varias disciplinas. Por ejemplo, en biología han sido utilizadas para modelar redes de regulación genética. En ciencias de la tierra, han sido utilizadas para modelar el clima durante la era cuaternaria [22, 23]; en la creación de un modelo de sismicidad [24, 25]; y para describir el fenómeno climático de El Niño [26]. En economía, han sido utilizadas para crear modelos que consideren el papel de los proveedores y productores que aparecen en la economía real [27, 28]. En criptografía y comunicaciones, las redes Booleanas han sido empleadas como generadores de código Hash ya que han demostrado ser seguras y de un bajo costo computacional [29].

3.1.2. Definición de red Booleana

Una red Booleana puede ser definida como:

- Una composición de nodos que pueden estar en uno de dos estados (0 ó 1) y enlaces que conectan a los nodos. La dinámica de la red está determinada por funciones Booleanas de los estados Booleanos.
- Un número de nodos conectados unos con otros a través de enlaces dirigidos. Cada nodo está asociado con una variable de estado Booleana y su valor está determinado por una función lógica Booleana que evalúa los argumentos de todos los nodos que son entrada del nodo en cuestión.

Con base en lo anterior, matemáticamente se define una red Booleana en general como una terna

$$BN = (G = (V, E), Q = \{0, 1\}, F = \{f_1, \dots, f_n\}), \quad (3.1)$$

donde G es un grafo dirigido, Q es el conjunto de los estados y F es el conjunto de las funciones de activación local o también llamadas tablas de verdad.

3.1.3. Clasificación de las redes Booleanas

Un punto importante en el estudio de las redes Booleanas es la forma en que éstas actualizan sus estados, ya que la dinámica de la red está estrechamente relacionada con este aspecto. Con base en este criterio y de acuerdo con [30], las redes Booleanas se clasifican en:

- **Síncronas.** Son redes Booleanas en las cuales todos los estados de sus nodos se actualizan de manera simultánea.
- **Asíncronas.** Esta categoría incluye aquellas redes Booleanas en las cuales los nodos son individualmente escogidos para ser actualizados en un orden elegido aleatoriamente. Pero además incluyen aquellas redes Booleanas que tienen múltiples retardos no uniformes de tal manera que los nodos actualizan su estado cuando existe una transición Booleana en sus entradas; este tipo de redes son comúnmente conocidas como redes Booleanas autónomas.

Los esquemas de actualización síncrono simplifican el análisis matemático y permiten simulaciones numéricas exactas; sin embargo, son muy poco probables en las redes del mundo real. Mientras que la actualización asíncrona de las redes Booleanas autónomas parece estar más apegada a la realidad.

Las redes Booleanas autónomas son redes con estados discretos y cuya evolución temporal es continua. Para modelarlas se han utilizado principalmente dos enfoques: las ecuaciones diferenciales lineales a pedazos [21] y las ecuaciones Booleanas con retardo [19, 20, 22, 26].

En esta tesis se estudia el fenómeno de sincronización de redes Booleanas autónomas bajo el enfoque de las ecuaciones Booleanas con retardo (BDEs por sus siglas en inglés de Boolean Delay Equations), cuya teoría se detalla en las siguientes secciones de este capítulo.

3.2. Ecuaciones Booleanas con retardo

3.2.1. Introducción a las BDEs

En varios sistemas de la vida real, como lo son los sistemas biológicos y los sistemas físicos, las interacciones entre las variables del sistema son altamente no lineales. Para algunas de estas variables, se pueden asociar umbrales. Entonces se puede describir el estado del sistema usando un vector de variables Booleanas, es decir, variables que sólo pueden tomar valores de 0 y 1. Las interacciones pueden ser descritas por funciones con valores booleanos.

Las ecuaciones Booleanas con retardo representan una herramienta de modelado especial creada para la formulación matemática de modelos conceptuales de los sistemas cuyo comportamiento puede estar basado en umbrales, poseen múltiples retroalimentaciones y distintos retardos. Las ecuaciones Booleanas con retardo son sistemas dinámicos semi-discretos o híbridos, donde las variables de estado son discretas, generalmente Booleanas, mientras que

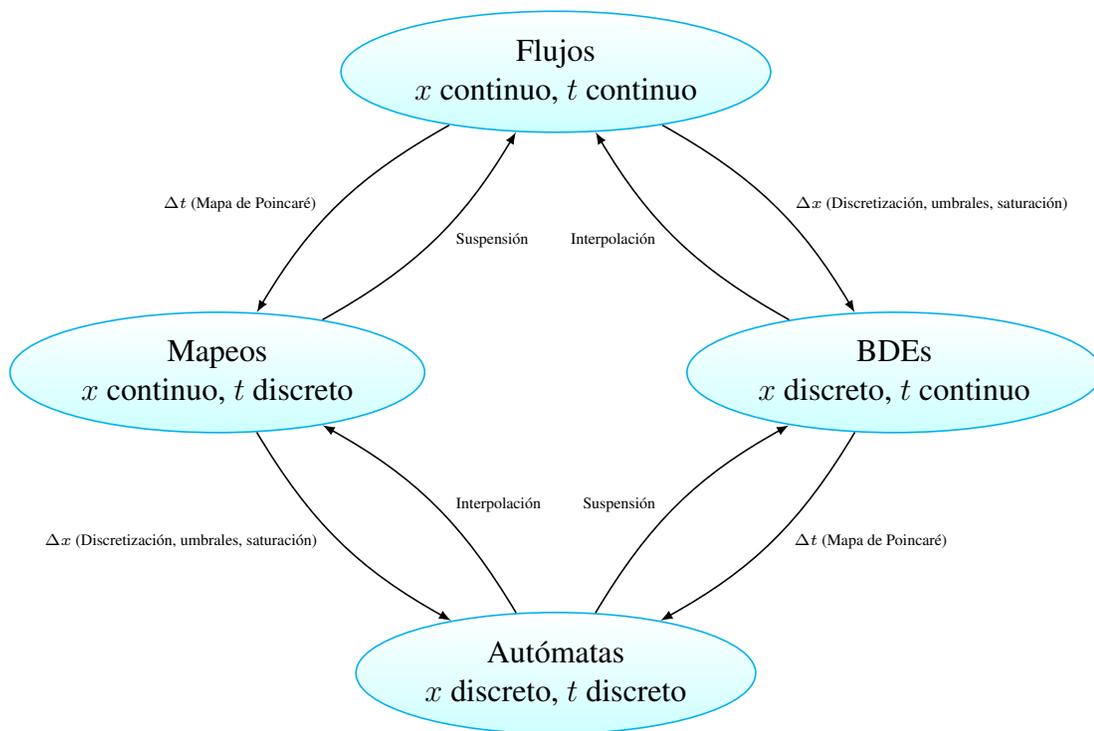


Figura 3.1: Clasificación de los sistemas dinámicos.

el tiempo en que evolucionan es continuo. En otras palabras, las BDEs son ecuaciones de evolución continua para un conjunto de variables discretas.

Es posible clasificar a los sistemas dinámicos respecto al tipo de variables de estado que poseen y a la forma en que dichos sistemas evolucionan a través del tiempo. Es decir, las variables de estado pueden ser continuas o discretas; de igual manera, el tiempo en que evolucionan también puede ser continuo o discreto. Dicha clasificación se muestra en la Figura 3.1.

Los sistemas en los cuales tanto las variables de estado como el tiempo son continuos son llamados flujos. Éstos están representados en la esquina superior del romboide de la Figura 3.1. Pertenecen a esta clase los campos vectoriales, las ecuaciones diferenciales ordinarias y parciales, las ecuaciones diferenciales funcionales y con retardo, y las ecuaciones diferenciales estocásticas.

Los sistemas con variables de estado continuas y tiempo discreto son conocidos como mapas. Están representados en la esquina izquierda de la Figura 3.1 e incluyen ecuaciones en diferencia ordinarias y parciales.

En los autómatas tanto las variables de estado como el tiempo son discretos. Este grupo está representado en la esquina inferior del romboide de la Figura 3.1. Los autómatas celulares y todas las máquinas de Turing, incluyendo las computadoras del mundo real, pertenecen a este grupo.

Finalmente, las ecuaciones Booleanas con retardo, la lógica cinética y conservativa completan la clasificación y ocupan la esquina derecha del romboide de la Figura 3.1.

3.2.2. Definición formal de BDEs

Considerando un sistema con n variables de estado continuas y evaluadas en los reales $\vec{v} = (v_1, v_2, \dots, v_n) \in \mathbb{R}^n$ para el cual existen umbrales $q_i \in \mathbb{R}$, entonces a cada variable $v_i \in \mathbb{R}$ se le asocia una variable Booleana, $x_i \in \mathbb{B} = \{0, 1\}$, es decir, una variable que puede ser un 0 lógico ó 1 lógico, dada por:

$$x_i(v_i) = \begin{cases} 0, & v_i \leq q_i \\ 1, & v_i > q_i \end{cases}, i = 1, 2, \dots, n. \quad (3.2)$$

El conjunto de variables Booleanas $\{x_1, x_2, \dots, x_n\}$ da una descripción cualitativa simple del sistema original, y reduce el número de posibles estados a 2^n . Es importante mencionar que existen además sistemas cuyas variables de estado son de manera inherente Booleanas. Para estos sistemas ya no es necesario definir un umbral y la teoría que a continuación se presenta aplica indistintamente.

Ahora bien, las ecuaciones que describen la evolución del vector de estado Booleano $\vec{x} = (x_1, x_2, \dots, x_n) \in \mathbb{B}^n$ debido a las interacciones retardadas entre las variables Booleanas $x_i \in \mathbb{B}$ tiene la forma:

$$\begin{cases} x_1(t) = f_1(x_1(t - \theta_{11}), x_2(t - \theta_{12}), \dots, x_n(t - \theta_{1n})), \\ x_2(t) = f_2(x_1(t - \theta_{21}), x_2(t - \theta_{22}), \dots, x_n(t - \theta_{2n})), \\ \vdots \\ x_n(t) = f_n(x_1(t - \theta_{n1}), x_2(t - \theta_{n2}), \dots, x_n(t - \theta_{nn})). \end{cases} \quad (3.3)$$

Aquí, cada variable Booleana x_i es una función del tiempo t y las funciones $f_i : \mathbb{B}^n \rightarrow \mathbb{B}$, $1 \leq i \leq n$, son las ecuaciones Booleanas que involucran operadores lógicos y retardos. Cada valor de $\theta_{ij} \in \mathbb{R}^+$, donde $1 \leq i, j \leq n$, es el tiempo que le toma a la variable x_j en afectar a la variable x_i .

El espacio de condiciones iniciales para los modelos con actualización autónoma difiere de los modelos síncronos y asíncronos. Para que una serie de tiempo sea especificada de manera única no basta con especificar todos los valores del estado en un instante dado, sino que es necesario especificar las series de tiempo de todos los nodos para una determinada duración de tiempo. Por ejemplo, para determinar las series de tiempo del nodo i comenzando en $t = 0$ se requiere del conocimiento de todas las señales que comenzaron a viajar a dicho nodo desde cada nodo j a partir del instante $t = -\tau_{ij}$. Por lo tanto, una condición inicial para (3.3) está definida como las series de tiempo de todos los nodos en el intervalo $[-\max\{\tau_{i1}, \tau_{i2}, \dots, \tau_{in}\}, 0)$.

3.2.3. Propiedades importantes en la BDEs

Los principales resultados acerca de las ecuaciones Booleanas con retardo obtenidos por Ghil y Mullhaupt en [20] son:

- Existencia y unicidad de las soluciones. Considere un sistema de ecuaciones Booleanas con retardo y condiciones iniciales constantes a pedazos en un intervalo de longitud

igual al mayor de los retardos. Se puede entonces probar por construcción la existencia de una solución única mostrando la ausencia de soluciones con un número infinito de saltos entre 0 y 1 en cualquier intervalo de tiempo finito.

- Lema *Pigeon-hole* o lema del casillero. Todo sistema de ecuaciones Booleanas con retardo que posee solamente retardos racionales puede verse como un autómata celular finito. La conmensurabilidad de los retardos crea una partición en el eje del tiempo en segmentos sobre los cuales las variables de estado permanecen constantes y cuya longitud es un múltiplo entero del mínimo común denominador de los retardos. Como existe un número finito de posibles asignaciones de dos valores a estos segmentos, la repetición debe ocurrir, y los únicos comportamientos asintóticos posibles son una eventual solución constante o una solución periódica en tiempo.

3.2.4. Clasificación de las BDEs

Ghil y Mullhaupt hicieron una clasificación de los sistemas de ecuaciones Booleanas con retardo, dividiéndolos en dos categorías.

- Conservativos. Son todos aquellos sistemas cuyas soluciones son inmediatamente periódicas para cualquier conjunto de retardos racionales y cualquier condición inicial. Un ejemplo simple de un sistema conservativo es $x(t) = x'(t - 1)$, cuya evolución temporal se encuentra graficada en la Figura 3.2. Las condiciones iniciales de este sistema son $x(t) = 0$ para $t < 0$. Aunque en la figura sólo se muestra la gráfica para un pequeño lapso de tiempo, las oscilaciones entre 0 y 1 permanecen infinitamente.

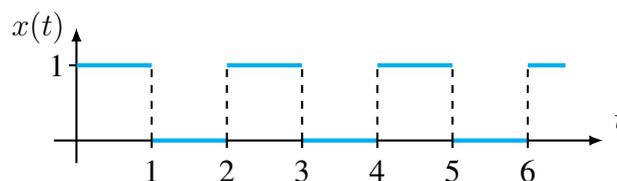


Figura 3.2: Evolución temporal del sistema conservativo $x(t) = x'(t - 1)$.

- Disipativos. Son aquellos sistemas que para algunos retardos racionales presentan un comportamiento transitorio antes de llegar a la eventual periodicidad. Un ejemplo de un sistema disipativo es $x(t) = (x(t - 1) + x'(t - \frac{1}{3}))'$. La Figura 3.3 muestra la evolución temporal de este sistema. Las condiciones iniciales para este sistema son $x(t) = 1$ para $t < 0$. De la gráfica se puede notar que existe un periodo transitorio en $0 < t < 1$. A partir de $t = 1$ el sistema comienza a oscilar entre 0 y 1 de manera periódica.

3.2.5. Comportamiento asintótico

De acuerdo con [26], los siguientes tipos de comportamiento asintótico se pueden observar en los sistemas de BDEs:

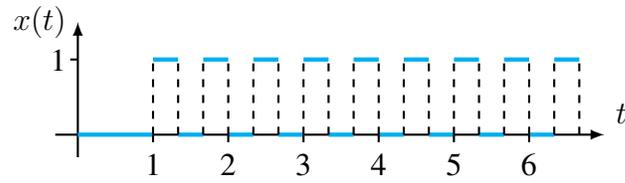


Figura 3.3: Evolución temporal del sistema disipativo $x(t) = (x(t - 1) + x'(t - \frac{1}{3}))'$.

- Punto fijo. La solución llega a uno de los posibles puntos de estado finitos y permanece ahí.
- Ciclo límite. La solución se convierte en periódica después de que transcurre un tiempo finito.
- Cuasi-periodicidad. La solución es la suma de varios “modos” inconmensurables.
- Complejidad creciente. El número de cambios de estado por unidad de tiempo se incrementa. Este número crece como una potencia positiva y fraccional del tiempo.

Cómputo reconfigurable

En el capítulo anterior fue abordada la teoría elemental de las redes Booleanas y particularmente, se ahondó en una manera de modelar por medio de ecuaciones Booleanas con retardo un tipo de red Booleana que evoluciona en el tiempo de forma continua.

Ahora bien, como se mencionó en el capítulo introductorio, esta tesis tiene además una parte aplicada que consta en el diseño e implementación de un dispositivo reconfigurable en el cual puedan ser realizadas diferentes topologías de redes Booleanas. El presente capítulo tiene como objetivo proveer las nociones básicas y un panorama general del estado del arte del cómputo reconfigurable.

4.1. Introducción

En el campo de la computación y de la electrónica, existen tres maneras diferentes de realizar cómputo:

- **Microprocesadores de alto rendimiento.** Proveen un medio para lograr requerimientos de procesamiento de alta velocidad. Son flexibles ya que cuentan con un conjunto amplio de instrucciones que permiten la implementación de tareas computables [31]. Desafortunadamente para muchas aplicaciones, un simple procesador no es lo suficientemente rápido. Además, el alto consumo de potencia y el costo elevado para su producción los deja fuera de ser una buena opción para muchas aplicaciones embebidas.
- **Circuitos integrados de aplicación específica (ASICs).** Son dispositivos de hardware altamente optimizados para realizar rápidamente tareas, pero están permanentemente configurados a un número pequeño de aplicaciones o incluso a sólo una. Para una tarea dada, los ASICs logran un alto desempeño, requieren menor área de silicio y consumen menor potencia que los microprocesadores programables. Sin embargo, carecen de

flexibilidad ya que si la tarea cambia, implica que un nuevo ASIC deba ser desarrollado [32].

- **Sistemas de cómputo reconfigurable.** Estos dispositivos contienen uno o más procesadores junto con una estructura reconfigurable sobre la cual unidades funcionales personalizadas pueden ser construidas [31]. A diferencia de un ASIC, no se requiere mandar a fabricar uno cada vez que se realiza una nueva aplicación. La flexibilidad y la configurabilidad de estos sistemas, junto con la creciente integración de tecnología, ha permitido crear plataformas sofisticadas que facilitan la reconfiguración, el prototipado rápido y la verificación temprana del diseño [33].

La investigación en la arquitectura de sistemas de cómputo ha sido una constante preocupación por parte de la gente que trabaja en ciencias computacionales como en la electrónica. Recientemente, el cómputo reconfigurable se está convirtiendo en una de las áreas de investigación con mayor interés por parte de los investigadores. Esto se debe a que se estima que el cómputo reconfigurable tiene el potencial de proveer soluciones eficientes para lograr un mejor desempeño, un menor consumo de potencia y robustez ante fallas en los sistemas de cómputo del futuro.

4.2. Definición de cómputo reconfigurable

Hasta el momento en este documento se ha mencionado al cómputo reconfigurable como una promisoriosa herramienta para diseñar e implementar mejores equipos en el futuro, sin embargo, no se le ha definido de forma explícita.

El cómputo reconfigurable puede ser definido como:

- El uso de dispositivos reconfigurables para propósitos computacionales [31].
- El estudio de la computación utilizando dispositivos reconfigurables [34].
- El proceso de explotar de mejor manera el potencial del hardware reconfigurable [35].

Las tres definiciones arriba mencionadas coinciden en el uso, explotación o utilización de dispositivos reconfigurables. Un dispositivo reconfigurable es un circuito de propósito general el cual tiene una estructura interna que puede ser modificada por el usuario (o a petición de éste, por el fabricante) para realizar una amplia gama de aplicaciones.

De acuerdo con [34], existe una sutil diferencia entre configuración y reconfiguración. Si el proceso de cambiar la estructura de un dispositivo reconfigurable ocurre al inicio del tiempo de ejecución se le conoce como un proceso de configuración. Mientras que reconfiguración es el proceso de cambiar la estructura de un dispositivo reconfigurable durante el tiempo de ejecución. Lo cierto es que dentro de la comunidad científica al no existir claridad entre estos dos términos se suelen utilizar de manera indistinta.

4.3. Desafíos en el diseño de dispositivos reconfigurables

Al diseñar un dispositivo reconfigurable es deseable que éste cumpla con las siguientes características [36]:

- **Eficiencia en consumo de potencia.** Un eficiente consumo de potencia aumentaría la gama de aplicaciones en que los dispositivos reconfigurables pueden ser empleados, por citar un ejemplo, podrían ser utilizados en sistemas embebidos como los celulares.
- **Confiabilidad.** La flexibilidad inherente de los dispositivos reconfigurables puede ser utilizada para detectar y corregir fallas de sí mismos.
- **Desempeño esperado.** En este punto es deseable que los dispositivos reconfigurables cumplan con los requerimientos de aplicación y del sistema.
- **Reconfiguración.** Es deseable que los dispositivos reconfigurables manejen velocidades de reconfiguración mayores con la finalidad de cumplir con los requerimientos de cómputo en tiempo real.

Las tres primeras características son limitantes de diseño y requerimientos del sistema difíciles de alcanzar con las tendencias tecnológicas actuales. Por otro lado, la reconfiguración es una propiedad deseada que hace que un dispositivo reconfigurable ofrezca ventajas significativas sobre los enfoques de los circuitos integrados estáticos.

4.4. Campos de aplicación del cómputo reconfigurable

A continuación se enumeran algunos de los campos de aplicación en los cuales el cómputo reconfigurable puede ser de gran importancia [34]:

- **Prototipado rápido.** Este proceso permite probar un dispositivo en hardware real antes de su producción de forma tal que los errores pueden ser detectados y corregidos sin afectar las piezas ya producidas. Un dispositivo reconfigurable es útil aquí porque puede ser utilizado varias veces para implementar diferentes versiones del producto final hasta que éste no presente errores.
- **Personalización en sistema.** Un dispositivo reconfigurable proporciona la capacidad de ser actualizado en campo. La personalización en el sistema también se puede utilizar para actualizar sistemas que se implementan en lugares no accesibles o de muy difícil acceso.
- **Sistemas de cómputo adaptativo.** Las necesidades de cómputo actuales requieren de sistemas que sean capaces de adaptar su comportamiento y estructura para modificar las condiciones de operación o de ambiente. El cómputo reconfigurable podría ayudar a la realización de sistemas adaptativos.

4.5. Evolución de los sistemas de cómputo reconfigurable

Gerald Estrin introdujo el concepto de cómputo reconfigurable en el año 1959. Sin embargo, ha tomado décadas lograr desarrollos importantes en esta área. Muchos investigadores han contribuido a la creación de dispositivos lógicos que puedan ser programados fácil y rápidamente para ejecutar una gran variedad de tareas.

A lo largo del tiempo han existido diferentes dispositivos reconfigurables. Éstos se pueden clasificar en dos grupos: de grano fino y de grano grueso. Los primeros incluyen a todo aquel dispositivo de hardware cuya funcionalidad puede modificarse a un muy bajo nivel de granularidad, es decir, el dispositivo puede ser modificado al añadir o remover un simple inversor o una sola compuerta NAND de dos entradas. Los dispositivos de grano grueso consisten generalmente de un conjunto de elementos de grano grueso, como puede ser una unidad aritmética-lógica (ALU), que permiten la implementación eficiente de una función de flujo de datos [34]. A continuación se enumeran algunos de los dispositivos reconfigurables que se han desarrollado:

- **Computadora con estructura Fija-más-variable.** Esta computadora fue propuesta en [37] por Estrin y Viswanathan en el año 1962. El sistema completo operaba bajo la configuración Fija-más-variable de tal forma que los mismos elementos utilizados para dar solución al problema específico de calcular los eigenvalores y eigenvectores de matrices reales simétricas podían ser reorganizados para otros problemas. La estructura Fija-más-variable consistía de una computadora de propósito general de alta velocidad (parte fija) operando a la par con un conjunto de sub-estructuras digitales de alta velocidad (parte variable). La cooperación de las dos partes ocurría bajo la dirección de un control supervisorio. La parte variable estaba hecha de unidades funcionales optimizadas para un problema específico. Los módulos podían ser insertados en cualquiera de las 36 posiciones de la tarjeta madre. La funcionalidad de la tarjeta madre podía ser modificada sustituyendo algunos bloques por unos nuevos.
- **Computadora Rammig.** Esta máquina fue propuesta por Franz J. Rammig en [38] en el año 1977. La idea principal era construir un sistema el cual, sin interferencia mecánica o manual, permitiera la construcción, modificación, procesamiento y destrucción de hardware digital real. Para lograr su objetivo, Rammig desarrolló un editor de hardware el cual estaba compuesto de un conjunto de módulos M , un conjunto de pines P y un mapeo uno a uno b sobre el conjunto de pines. Para implementar el editor de hardware, existía un arreglo de selectores que conectaba las salidas de los módulos con las entradas de los selectores y las salidas de los selectores con las entradas de los módulos.
- **Memoria activa programable (PAM).** Propuesta en [39], la PAM es un arreglo uniforme de celdas idénticas conectadas de manera repetitiva. Cada celda, llamada bit activo programable PAB, era lo suficientemente general de tal manera que cualquier circuito digital síncrono podía ser realizado en una PAM para velocidades de reloj suficientemente pequeñas.
- **Arreglo lógico programable (PLA) y lógica de arreglo programable (PAL).** Ambos forman parte de los dispositivos lógicos programables PLDs. Consisten de un plano

de compuertas AND conectadas a un plano de compuertas OR. Tanto las señales de entrada como sus negaciones están conectadas a las compuertas AND. Las salidas de las compuertas AND estaban conectadas a las entradas de las compuertas OR cuyas salidas correspondían a las salidas del PLA/PAL. En la PAL el plano de compuertas AND es programable y el plano de compuertas OR es fijo, mientras que en el PLA ambos planos son programables. Estos dispositivos constaban de unos pocos cientos de compuertas lógicas.

- **Dispositivo lógico programable complejo (CPLD).** Consiste de un conjunto de macro-celdas, bloques de entrada/salida y una red de interconexión. Una macro-celda contiene varios PLAs y flip-flops. La red de interconexión sirve para comunicar a los bloques de entrada/salida con macro-celdas y macro-celdas entre sí. Los CPLDs aumentaron su capacidad a varios miles de cientos de compuertas lógicas comparados con los PLDs.
- **Arreglo de compuertas programables en campo (FPGA).** Es un dispositivo reconfigurable que puede ejecutar una gran variedad de aplicaciones de hardware. Un software especializado transforma una aplicación de hardware en un flujo de bits de programación que puede ser fácil e instantáneamente descargado a un FPGA [40]. Fue introducido por Xilinx en 1985 y su aparición marca el inicio de la era de reutilización del hardware lo cual condujo a una implementación más flexible de los circuitos integrados.

Los FPGAs son dispositivos revolucionarios que combinan los beneficios del software y hardware. Al igual que una computadora, los FPGAs pueden computar simultáneamente millones de operaciones utilizando los recursos distribuidos dentro del chip [35]. Por otro lado, a diferencia de los ASICs en lo que la programación es hecha por el fabricante, los FPGAs tienen la ventaja de poder ser programados por el usuario las veces que sea necesario.

La arquitectura básica de un FPGA se muestra en la Figura 4.1. Los bloques lógicos configurables están ordenados en una matriz bidimensional y están interconectados por una red de ruteo programable. Los bloques de entrada-salida están situados en la periferia del chip del FPGA y también se encuentran conectados a la red de ruteo programable. Estos recursos de ruteo incluyen segmentos de pistas e interruptores programables para enlazar bloques lógicos a pistas, bloques de entrada-salida a pistas o pistas entre sí.

Los bloques lógicos configurables contienen unidades de procesamiento para realizar lógica combinacional simple, además cuentan también con flip-flops para implementar lógica secuencial. Debido a que las unidades de procesamiento a menudo son simplemente memorias, cualquier función de cinco o seis entradas puede ser implementada en cada bloque lógico reconfigurable. [35].

La red de ruteo programable suele ocupar el 80-90 % del área del chip del FPGA. Una red de ruteo basado en malla consiste de pistas de ruteo horizontales y verticales las cuales están interconectadas por medio de cajas de interruptores. Los bloques lógicos configurables están conectados a la red de ruteo a través de cajas de conexión.

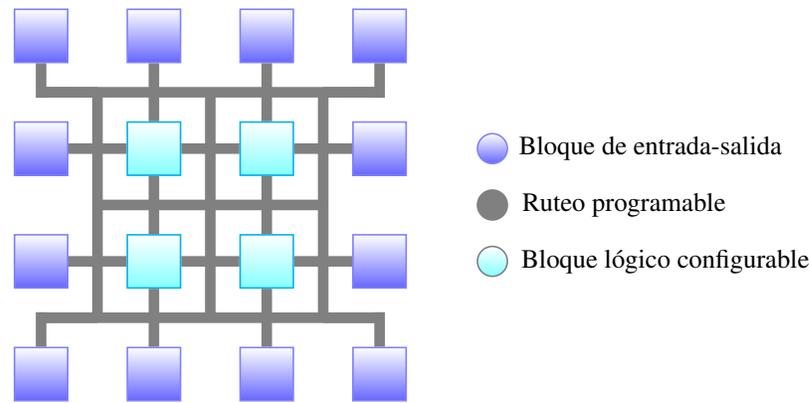


Figura 4.1: Arquitectura básica de un FPGA.

Los bloques de entrada-salida implementan la interfaz de los pines del empaquetado con la red de ruteo programable y con los bloques lógicos configurables. La comunicación con señales externas es hecha por medio de *buffers* tri-estado cuya dirección y polaridad son determinados por bits de configuración específica.

La flexibilidad de los FPGAs se debe a la naturaleza configurable de los bloques lógicos que lo componen y a la manera en que éstos se pueden interconectar mediante la red de ruteo programable. Una aplicación deseada puede ser mapeada a estos recursos reconfigurables utilizando el software específico. Sin embargo, el esquema reconfigurable de los FPGAs es típicamente lento debido a que requiere de cierto tiempo para re-cablearse [41]. Esto es debido en gran medida a que los bloques lógicos configurables dentro de los FPGAs son un tanto estáticos, lo cual representa una desventaja ya que para ejecutar una función lógica compleja es necesario poner todo en función de esos bloques lógicos estáticos, y esto se ve reflejado en el rendimiento.

4.6. Trabajos anteriores enfocados a la obtención de dispositivos reconfigurables

La comunidad científica dedicada al diseño y realización de arquitecturas reconfigurables ha explorado y manipulado diferentes fenómenos para obtener nuevas formas de hacer cómputo. La idea básica principal es crear dispositivos a partir de los patrones inherentes de estos fenómenos para codificar entradas y subsecuentemente obtener una salida deseada [42]; y así emular las compuertas lógicas que son la base del cómputo de propósito general.

En seguida se presentan algunos trabajos que resultan interesantes y que de cierto modo representan el estado del arte en cuanto a hardware reconfigurable se refiere.

- En 1998, Sinha y Ditto demuestran la capacidad de los látices de mapeos caóticos acoplados para realizar cálculos sencillos. Estos autores proponen una red de mapeos logísticos con la que logran emular compuertas lógicas, codificar números, sumar y multiplicar [43]. Un año más tarde, extienden el esquema que propusieron a un sistema continuo multidimensional [44].

- En 2002, Munakata *et al.* presentan los principios básicos para implementar las funciones de cómputo fundamentales (compuertas AND, OR, XOR y NOT y operaciones aritméticas bit a bit) por medio del empleo de elementos caóticos. En este caso utilizan el mapeo logístico para ejemplificar su procedimiento [45].
- En 2003, Murali *et al.* muestran la implementación directa de la compuerta lógica NOR por medio del circuito de Chua. Con su experimento muestran además la capacidad de los sistemas caóticos de realizar cómputo de propósito general. Utilizando un umbral y niveles de entrada adecuados, se pueden emular también otras compuertas lógicas [46].
- En el año 2005, Murali *et al.* reportan la primera realización experimental de una celda lógica reconfigurable capaz de emular las compuertas lógicas NOT, AND, OR y XOR utilizando un circuito caótico. Un mecanismo de umbral permite al circuito conmutar entre diferentes comportamientos y así emular diferentes compuertas [47].
- En 2008, Peng *et al.* presenta tres métodos basados en sistemas lineales a pedazos para lograr la obtención de compuertas lógicas, sumadores y elementos de memoria. El primer de los esquemas está basado en un sistema lineal discreto, mientras que los otros dos restantes se basan en la sincronización de sistemas lineales continuos [48].
- En el año 2009, Murali *et al.* proponen la implementación flexible y directa de compuertas lógicas utilizando la evolución dinámica de un sistema no lineal. El concepto presentado involucra la observación del estado del sistema en diferentes tiempos para obtener las diferentes salidas. Con la metodología expuesta logran emular las compuertas NAND, AND, NOR, OR y XOR, pero además la compuertas XOR y XNOR con múltiples entradas [49].
- En 2010, Guerra *et al.* presentan un dispositivo nanomecánico capaz de funcionar como una compuerta lógica reprogramable. La función lógica es programada ajustando los parámetros de operación de un resonador controlado por la intensidad de ruido [50].
- En 2010, Peng *et al.* explora los sistemas lineales por pedazos para construir compuertas lógicas. El esquema que proponen puede discriminar dos señales de entrada y obtener 16 operaciones lógicas por medio de diferentes combinaciones de parámetros y condiciones para determinar la salida [51].
- En 2011, Peng *et al.* presentan un método de umbral de ventana para construir una arquitectura lógica dinámica. Específicamente, proponen, analizan e implementan compuertas lógicas con múltiples entradas y múltiples salidas [52].
- En 2012, Yuan y Liu exploran redes neuronales celulares (CNN) para construir una arquitectura dinámica. Con ayuda de una CNN para emular diferentes compuertas lógicas. Los esquemas que proponen pueden discriminar las dos señales de entrada y conmutar entre 16 roles operacionales simplemente cambiando algunos parámetros [53].
- En 2013, Li *et al.* proponen un esquema para obtener funciones lógicas ajustando algunos parámetros específicos del sistema dinámico $\dot{x} = -px + p(I - k)^2$. Por medio de un mecanismo de umbral determinan la salida de la compuerta lógica [41].

- En 2013, Rothenbuhler *et al.* presentan la implementación de compuertas lógicas de umbral (TLG) usando dispositivos memresistivos de plata-calco-genuro y circuitos CMOS. Las TLGs son redes neuronales artificiales de un solo nodo diseñadas específicamente para emular una operación lógica [54].

Diseño e implementación de un FPDGA

En el capítulo anterior se presentó un panorama general de algunos de los dispositivos reconfigurables que han sido construidos y/o comercializados al día de hoy. Además, se expusieron varias líneas de investigación que están siendo exploradas con la finalidad de lograr implementar cómputo reconfigurable.

Tal como se mencionó en el capítulo anterior, uno de los dispositivos reconfigurables que presenta el problema de arquitectura estática es el FPGA. Esto se debe a que los bloques lógicos que integran la matriz están confinados a realizar una tarea específica, de tal manera que para ejecutar operaciones lógicas más complejas es necesario poner todo en función de esos bloques estáticos, lo cual se ve reflejado en el rendimiento.

El presente capítulo tiene como objetivo presentar la motivación e hipótesis de esta tesis, así como exponer a detalle la propuesta que da respuesta al cuestionamiento que motiva este trabajo.

5.1. Motivación, hipótesis y propuesta

Al ver la naturaleza estática de los bloques lógicos dentro de un FPGA, surgió la pregunta que motiva la presente tesis. ¿Se puede superar la limitante que impone la arquitectura estática en la próxima generación de arreglos de compuertas configurables?

5.1.1. Hipótesis

La **dinámica no lineal** puede ayudar a la obtención de compuertas lógicas dinámicas. Estas compuertas junto con conexiones programables son aplicables en la construcción de un arreglo de compuertas dinámicas programables en campo (FPDGA). Este dispositivo puede ser utilizado como base para **la generación de redes Booleanas autónomas y el análisis del fenómeno de sincronización entre ellas.**

5.1.2. Propuesta

Utilizar un sistema dinámico no lineal para emular diferentes compuertas lógicas ya que éstas son la base del cómputo de propósito general, y una vez obtenidas las compuertas lógicas, integrar estos elementos en un dispositivo programable por el usuario, es decir, para crear un arreglo de compuertas dinámicas programables en campo (FPDGA).

5.2. FPDGA: Arreglo de compuertas dinámicas programables en campo

Un arreglo de compuertas dinámicas programables en campo FPDGA es un dispositivo programable que contiene bloques lógicos cuya funcionalidad e interconexión puede ser configurada *in situ* mediante un lenguaje de descripción especializado. El FPDGA convencionalmente consiste de cuatro partes principales: compuertas lógicas dinámicas, estructura de ruteo programable, bloques de entrada y bloques de salida. La arquitectura de las compuertas lógicas del FPDGA está basada en un sistema no lineal y estas compuertas se pueden configurar para realizar cualquiera de las siguientes funciones lógicas: AND, OR, XOR, NAND, NOR, XNOR y NOT.

Este dispositivo puede ser visto como una versión simplificada de un FPGA. De hecho, su diseño está basado en la arquitectura del FPGA.

La principal característica del FPDGA es su capacidad de doble programación:

- La funcionalidad de cada compuerta lógica puede ser configurada.
- La interconexión entre compuertas lógicas puede ser personalizada.

5.3. Consideraciones de diseño del FPDGA

Para el diseño del FPDGA se hicieron las siguientes consideraciones:

- La arquitectura del FPDGA consiste de cuatro elementos principales: bloques de entrada, bloques de salida, compuertas lógicas dinámicas y estructura de ruteo programable tal como se muestra en la Figura 5.10. Es de suma importancia notar que en este nuevo dispositivo los bloques de entrada y los bloques de salida están separados en contraste con el FPGA.
- Los elementos lógicos dentro del FPDGA son llamados compuertas lógicas dinámicas (DLG). En esta primera realización, éstas pueden implementar de forma directa las funciones lógicas AND, OR, XOR, NAND y NOR solamente variando un parámetro de control. Y de forma indirecta pueden operar como funciones XNOR y NOT.
- Las DLGs contenidas en el FPDGA están confinadas a implementar operaciones lógicas binarias, es decir, cada DLG tiene dos entradas y una salida. Las formas de implementar la operación unaria NOT y la operación binaria XNOR serán discutidas posteriormente.

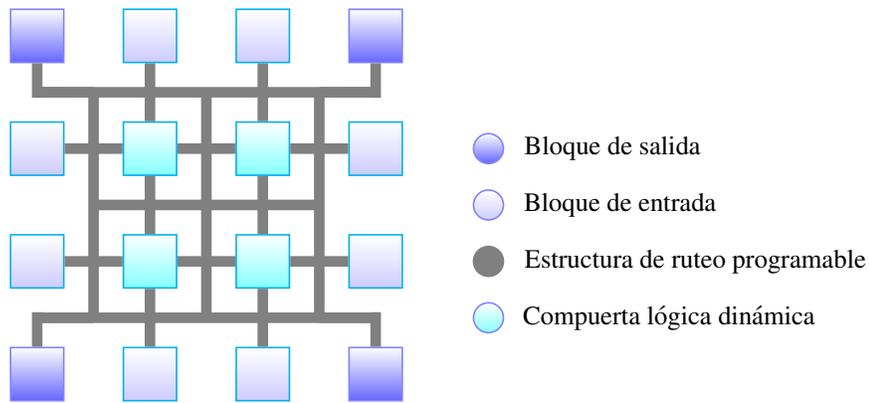


Figura 5.1: Arquitectura básica de un FPDGA.

- La estructura de ruteo programable proporciona un medio para interconectar las entradas de las DLGs ya sea con los bloques de entrada o con las salidas de las DLGs. Además incluyen la manera de configurar el parámetro de control de cada DLG por medio de multiplexores analógicos.
- En esta primera realización, el dispositivo FPDGA completo consiste de cuatro compuertas lógicas dinámicas, ocho bloques de entrada, cuatro bloques de salida y una estructura de ruteo programable. Esta última consiste de cuatro multiplexores analógicos y veinticuatro multiplexores digitales. Es importante hacer notar que estas cantidades son solamente ilustrativas, mas no restrictivas, de tal forma que el dispositivo puede ser escalable.

5.4. Compuertas lógicas dinámicas

El núcleo del FPDGA son las llamadas compuertas lógicas dinámicas ya que éstas son las encargadas de realizar el cómputo de propósito general que sea programado por el usuario. Reciben el adjetivo de dinámica porque cada una de estas compuertas se puede programar para implementar varias operaciones lógicas después de que un parámetro de control se ha establecido. Mediante el ajuste de este parámetro es posible obtener directamente una de las siguientes operaciones lógicas: AND, OR, XOR, NAND y NOR. Mientras que de forma indirecta pueden ser implementadas las operaciones lógicas XNOR y NOT.

5.4.1. Modelo matemático de la compuerta lógica dinámica

Con la finalidad de desarrollar un dispositivo capaz de emular el comportamiento de diferentes compuertas lógicas, en esta tesis se considera el siguiente sistema lineal afín:

$$h(\mathbf{x}) = \mathbf{a}\mathbf{x} + b, \quad (5.1)$$

donde $\mathbf{x} = (x_1, x_2)^T$ es un vector de estados con $x_i \in \mathbb{B} = \{0, 1\}$, $\mathbf{a} = (a_1, a_2) \in \mathbb{R}^2$, $h : \mathbb{B}^2 \rightarrow \mathbb{R}$ es la suma de un producto escalar más un *offset* o ajuste dado por el parámetro $b \in \mathbb{R}$. La salida del sistema está dada por $Y : \mathbb{R} \rightarrow \mathbb{B}$ como sigue:

$$Y(h) = \begin{cases} 1, & \text{si } |h| < k; \\ 0, & \text{en otro caso.} \end{cases} \quad (5.2)$$

donde $k \in \mathbb{R}$.

Entonces, la ecuación (5.1) puede tener el resultado mostrado en la tercer columna de la Tabla 5.1, una vez que los respectivos valores de x_1 y x_2 han sido sustituidos.

x_1	x_2	$h(\mathbf{x})$	AND	OR	NAND	NOR	XOR
0	0	b	0	0	1	1	0
0	1	$a_2 + b$	0	1	1	0	1
1	0	$a_1 + b$	0	1	1	0	1
1	1	$a_1 + a_2 + b$	1	1	0	0	0

Tabla 5.1: Valores de $h(\mathbf{x})$ de la ecuación (5.1), y tablas de verdad para las compuertas lógicas AND, OR, NAND, NOR y XOR correspondientes a la salida Y en la ecuación (5.2).

Por lo tanto, la salida de la ecuación (5.2) sólo produce valores del conjunto $\{0, 1\}$ dependiendo de si el valor absoluto de $h(\mathbf{x})$ es menor que el valor de k . El sistema dado por las ecuaciones (5.1) y (5.2) puede comportarse como una compuerta lógica dinámica dependiendo de los valores de los parámetros a_1 , a_2 , b y k que se elijan. Esto significa que estos valores no son únicos, sino que existen varios que pueden satisfacer las ecuaciones (5.1) y (5.2).

Los parámetros a_1 , a_2 , b y k son fundamentales para el funcionamiento de la compuerta lógica dinámica. Sin embargo, la idea principal es que la funcionalidad de la compuerta dependa solamente del parámetro b . Para elegir el parámetro b , es necesario resolver un sistema de desigualdades para cada operación lógica que se quiera obtener.

Por ejemplo, para la compuerta AND, el sistema de desigualdades a resolver es el siguiente:

$$-k > b \quad \vee \quad k < b, \quad (5.3)$$

$$-k - a_2 > b \quad \vee \quad k - a_2 < b, \quad (5.4)$$

$$-k - a_1 > b \quad \vee \quad k - a_1 < b, \quad (5.5)$$

$$-k - a_1 - a_2 < b \quad \wedge \quad k - a_1 - a_2 > b. \quad (5.6)$$

Con la finalidad de simplificar el procedimiento para obtener el valor del parámetro b , en esta tesis se supuso que el valor de $a_1 = a_2 = 2$ y $k = 1.5$, de tal manera que al sustituir estos valores en el sistema de desigualdades dado por (5.3)-(5.6), se obtiene:

$$-1.5 > b \quad \vee \quad 1.5 < b, \quad (5.7)$$

$$-3.5 > b \quad \vee \quad -0.5 < b, \quad (5.8)$$

$$-3.5 > b \quad \vee \quad -0.5 < b, \quad (5.9)$$

$$-5.5 < b \quad \wedge \quad -2.5 > b. \quad (5.10)$$

Debido a que se supuso que $a_1 = a_2$, las desigualdades (5.8) y (5.9) son idénticas. Por lo tanto se puede prescindir de una de ellas y el sistema de desigualdades (5.7)-(5.10) se puede reescribir de la siguiente manera:

$$-1.5 > b \quad \vee \quad 1.5 < b, \quad (5.11)$$

$$-3.5 > b \quad \vee \quad -0.5 < b, \quad (5.12)$$

$$-5.5 < b \quad \wedge \quad -2.5 > b. \quad (5.13)$$

La Figura 5.2 muestra la solución de cada una de las desigualdades del sistema (5.11)-(5.13). En color azul marino y línea continua, se encuentra graficada la solución de (5.11); en gris y línea discontinua, la solución de (5.12); mientras que en azul cielo y línea punteada, la solución de (5.13). Para elegir el valor del parámetro b , es necesario que se satisfagan las tres desigualdades simultáneamente. Esto se muestra en la región sombreada de la Figura 5.2, por lo tanto, cualquier valor en el intervalo $-5.5 < b < -3.5$ sirve para emular la compuerta AND.

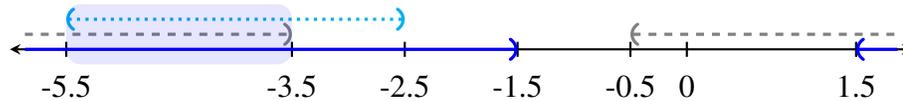


Figura 5.2: Intervalo de valores de b para emular la función lógica AND.

En el caso de la compuerta OR, el sistema de desigualdades que se debe resolver es el siguiente:

$$-1.5 > b \quad \vee \quad 1.5 < b, \quad (5.14)$$

$$-3.5 < b \quad \wedge \quad -0.5 > b, \quad (5.15)$$

$$-5.5 < b \quad \wedge \quad -2.5 > b. \quad (5.16)$$

En la Figura 5.3, se muestra la solución para cada una de las desigualdades del sistema (5.14)-(5.16). En color azul marino y línea continua, se encuentra graficada la solución de (5.14); en gris y línea discontinua, la solución de (5.15); mientras que en azul cielo y línea punteada, la solución de (5.16). El valor del parámetro b necesario para emular la función lógica OR debe estar dentro del intervalo $-3.5 < b < -2.5$. Gráficamente, este intervalo corresponde a la región sombreada de la Figura 5.3.

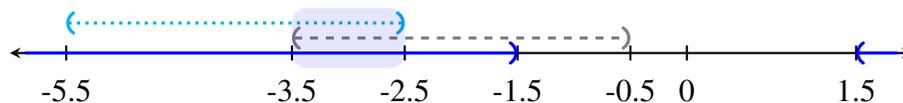


Figura 5.3: Intervalo de valores de b para emular la función lógica OR.

Para obtener la compuerta NAND, se debe resolver el siguiente sistema de desigualdades:

$$-1.5 < b \quad \wedge \quad 1.5 > b, \quad (5.17)$$

$$-3.5 < b \quad \wedge \quad -0.5 > b, \quad (5.18)$$

$$-5.5 > b \quad \vee \quad -2.5 < b. \quad (5.19)$$

La solución a cada una de estas desigualdades se muestra en la Figura 5.4. En color azul marino y línea continua, se encuentra graficada la solución de (5.17); en gris y línea discontinua, la solución de (5.18); mientras que en azul cielo y línea punteada, la solución de (5.19). Por lo tanto, para emular la compuerta NAND, es necesario elegir un valor para el parámetro b dentro del intervalo $-1.5 < b < -0.5$, el cual corresponde a la región sombreada de la Figura 5.4.

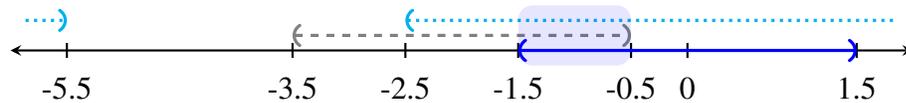


Figura 5.4: Intervalo de valores de b para emular la función lógica NAND.

Para emular la compuerta NOR, se requiere resolver el siguiente sistema de desigualdades:

$$-1.5 < b \quad \wedge \quad 1.5 > b, \quad (5.20)$$

$$-3.5 > b \quad \vee \quad -0.5 < b, \quad (5.21)$$

$$-5.5 > b \quad \vee \quad -2.5 < b. \quad (5.22)$$

En la Figura 5.5, se muestra la solución de cada una de estas desigualdades. En color azul marino y línea continua, se encuentra graficada la solución de (5.20); en gris y línea discontinua, la solución de (5.21); mientras que en azul cielo y línea punteada, la solución de (5.22). De tal manera que para emular la compuerta NOR, es necesario elegir un valor para el parámetro b dentro del intervalo $-0.5 < b < 1.5$, el cual corresponde a la región sombreada de la Figura 5.5.

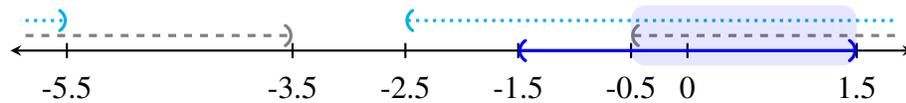


Figura 5.5: Intervalo de valores de b para emular la función lógica NOR.

Tal como se mencionó en la Sección 5.3, una de las consideraciones de diseño de la compuerta lógica dinámica es que ésta pueda emular cinco operaciones lógicas diferentes de manera directa. De tal manera que la última de las compuertas que se puede obtener con los valores de a_1 , a_2 y k previamente seleccionados es la función lógica XOR. Para emular esta compuerta, es preciso resolver el siguiente sistema de desigualdades:

$$-1.5 > b \quad \vee \quad 1.5 < b, \quad (5.23)$$

$$-3.5 < b \quad \wedge \quad -0.5 > b, \quad (5.24)$$

$$-5.5 > b \quad \vee \quad -2.5 < b. \quad (5.25)$$

En la Figura 5.6 se encuentra graficada la solución a cada una de estas desigualdades. El trazo en color azul marino y línea continua corresponde a la solución de (5.23); el trazo gris y línea discontinua, a la solución de (5.24); mientras que el trazo en color azul cielo y línea punteada, a la solución de (5.25). Por lo tanto, si se desea emular la compuerta XOR, es requisito indispensable seleccionar el valor del parámetro b dentro del intervalo $-2.5 < b < -1.5$. Dicho intervalo satisface el sistema de desigualdades dado por (5.23)-(5.25) y está representado por la región sombreada de la Figura 5.6.

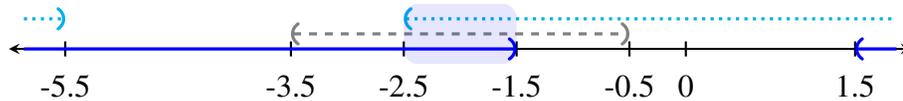


Figura 5.6: Intervalo de valores de b para emular la función lógica XOR.

5.4.2. Implementación electrónica de las compuertas lógicas dinámicas

El siguiente paso en el diseño de una compuerta lógica dinámica es llevar el modelo matemático a su implementación electrónica. El circuito electrónico mostrado en la Figura 5.7 corresponde al sistema con estructura dinámica de las ecuaciones (5.1) y (5.2). Éste comprende tres bloques principales:

- **Un bloque de comparadores.** Consta de un par de comparadores que determinan si los voltajes de entrada son señales en “alto” o son señales en “bajo”, es decir, determinan si se trata de un 1 lógico o un 0 lógico, respectivamente.
- **Un bloque sumador.** Está acompañado de una señal de compensación la cual está definida como parámetro de control para determinar el comportamiento de la compuerta lógica dinámica. El bloque sumador primeramente suma las señales de salida provenientes del bloque de entrada con la señal de compensación y posteriormente, invierte la suma.
- **Un bloque detector de ventana.** Determina si la suma proveniente del bloque sumador cae en un intervalo determinado por un voltaje de umbral definido por el usuario, y genera un nivel “alto” (1 lógico) si la señal recibida del bloque sumador está en el intervalo, en caso contrario genera un nivel “bajo” (0 lógico).

Desde un punto de vista electrónico, el circuito consiste de cuatro comparadores LM311, un amplificador operacional TL081, siete resistencias y fuentes de voltaje. Para el funcionamiento del circuito se considera que un 0 lógico y un 1 lógico son representados por 0 Volts y 5 Volts, respectivamente. La salida del primer comparador (LM311a) será aproximadamente

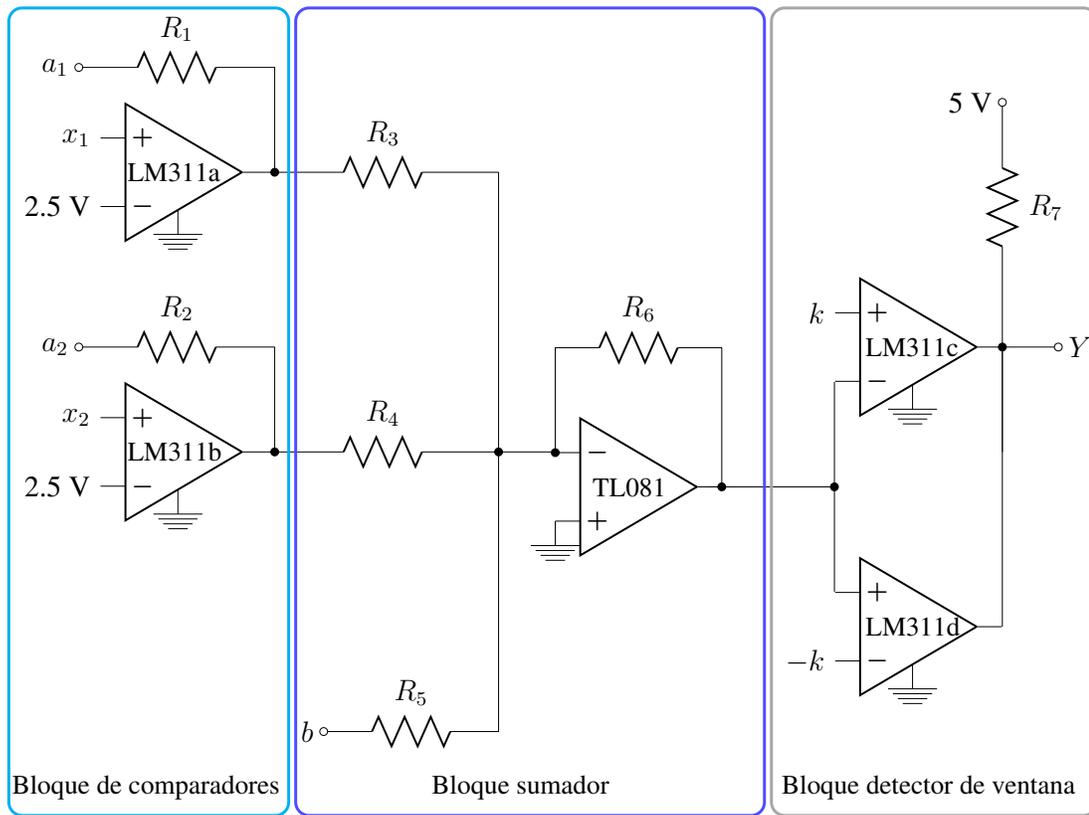


Figura 5.7: Circuito electrónico de las compuertas lógicas dinámicas. Los valores requeridos son: $R_{1;2;7} = 470 \Omega$, $R_{3;4;5;6} = 10 \text{ k}\Omega$, $k = 1.5 \text{ V}$, $a_{1,2} = 2.0 \text{ V}$.

el valor de a_1 , si $x_1 > 2.5 \text{ V}$. En otro caso, la salida del primer comparador será de 0 V. De igual manera, la salida del segundo comparador (LM311b) será aproximadamente el valor de a_2 , si $x_2 > 2.5 \text{ V}$. En otro caso, la salida del segundo comparador será de 0 V. Las salidas de los comparadores LM311a y LM311b, junto con un voltaje denotado como b (parámetro de control) son las entradas de un amplificador operacional (TL081). Este amplificador operacional trabaja en configuración sumador-inversor. El valor de la suma puede tener cuatro valores b , $a_1 + b$, $a_2 + b$, y $a_1 + a_2 + b$. Finalmente, dos comparadores (LM311c y LM311d) son conectados en configuración detector de ventana y tienen como entrada el valor de la suma. Si el resultado de la suma (salida del TL081) cae dentro del intervalo $(-k, k)$, entonces la salida del detector de ventana será un 1 lógico, de otra forma será un 0 lógico.

En esta primera realización se fijaron los valores de $a_1 = a_2 = 2.0 \text{ V}$, $k = 1.5 \text{ V}$ y los valores para el parámetro b para la obtención de las diferentes compuertas lógicas se muestran en la Tabla 5.2.

Parámetro	AND	OR	XOR	NAND	NOR
b	-4.0 V	-3.0 V	-2.0 V	-1.0 V	0.0 V

Tabla 5.2: Valores del parámetro b requeridos para la obtención de las diferentes compuertas lógicas.

5.5. Estructura de ruteo programable

Dentro del FPDGA esta estructura está encargada de dos tareas específicas. Por un lado incluye los medios para configurar el parámetro de control de cada DLG, es decir, permite la selección de la funcionalidad de cada compuerta. Y por otro lado, proporciona los recursos para interconectar las entradas de las DLGs ya sea con los bloques de entrada o con las salidas de las DLGs, en otras palabras, hace posible la interconexión programable de compuertas.

5.5.1. Selección de la funcionalidad de las compuertas lógicas dinámicas

La Figura 5.8 es un diagrama de bloques que muestra la forma en que se implementó la selección del parámetro de control para las compuertas lógicas dinámicas. Como se mencionó cada compuerta lógica reconfigurable cuenta con su propio parámetro de control para seleccionar cualquiera de las cinco operaciones lógicas que puede realizar (AND, OR, XOR, NAND, NOR). Para lograr esta multifuncionalidad, los cinco valores de voltaje necesarios son entregados por una fuente de voltaje; dichos valores de tensión se introducen a las entradas de cuatro multiplexores analógicos. Una palabra de control es utilizada para seleccionar cuál de los cinco voltajes saldrá de cada multiplexor analógico hacia cada parámetro de control b . En esta realización, una palabra de control consta de tres bits para cada uno de los multiplexores analógicos. Esta combinación de tres bits ingresa a los multiplexores analógicos que definirán cuál de los cinco voltajes pasará al parámetro de control b de cada compuerta. De tal manera que una posible realización de esta programación se muestra en la Tabla 5.3. Debe notarse que hay cinco voltajes que se pueden elegir y un total de ocho combinaciones, por lo tanto, hay tres combinaciones que por defecto llevan a obtener la compuerta NOR.

SW3	SW2	SW1	Operación lógica
0	0	0	AND
0	0	1	OR
0	1	0	XOR
1	1	1	NAND
1	0	0	NOR
1	0	1	NOR
1	1	0	NOR
1	1	1	NOR

Tabla 5.3: Combinación en los interruptores para la selección de la funcionalidad de cada compuerta lógica dinámica.

El diagrama esquemático de la implementación de la selección del parámetro de control b para una DLG se muestra en la Figura 5.9. Es importante mencionar que al ser idénticas cada una de las DLGs que forman el FPGA, este proceso es el mismo para cada una de ellas. Una fuente de voltaje proporciona los voltajes mostrados en la Tabla 5.2. Estos voltajes ingresan a las entradas del multiplexor analógico ADG608. Debe notarse que el resto de las entradas

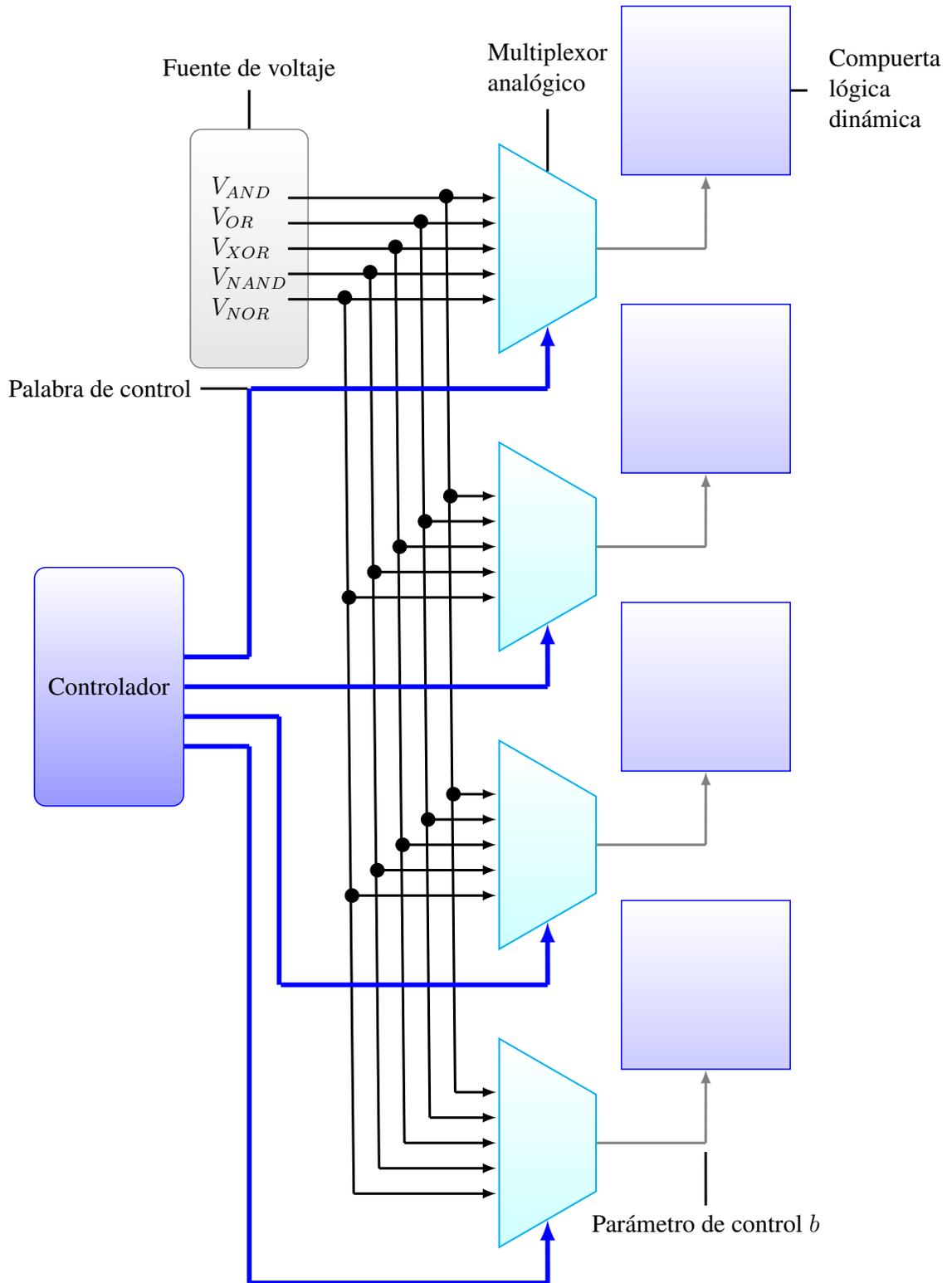


Figura 5.8: Diagrama de bloques de la implementación de la selección del parámetro de control.

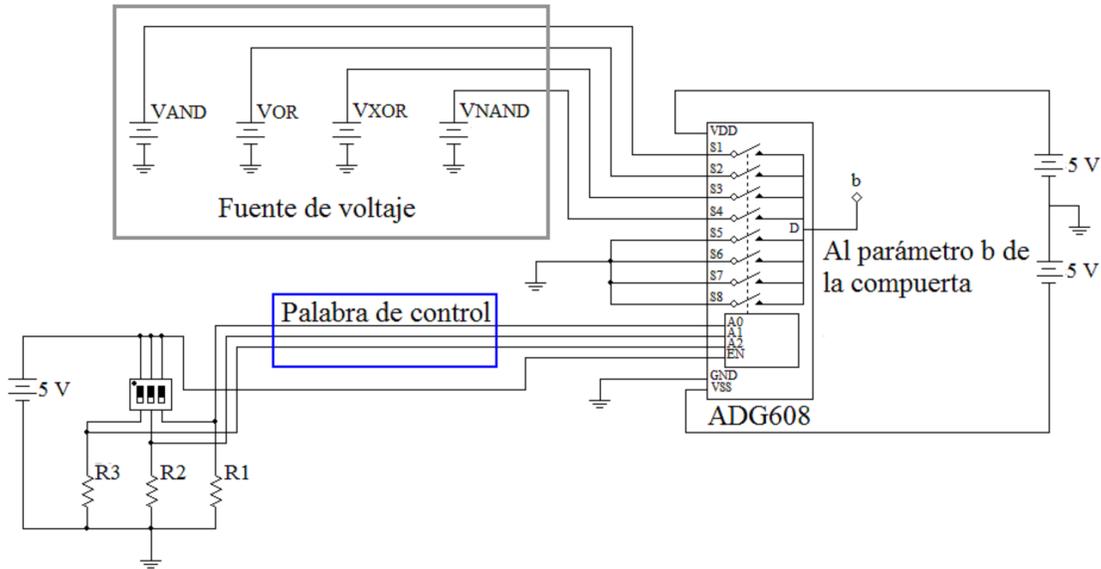


Figura 5.9: Diagrama esquemático de la implementación de la selección de la funcionalidad de una compuerta lógica dinámica.

del multiplexor fueron puestas a tierra. Mediante interruptores se hacen las posibles combinaciones binarias para la selección del voltaje que se tendrá a la salida del multiplexor. Esta combinación es la palabra de control y consta de tres bits. Las combinaciones posibles se encuentran en la Tabla 5.3. El multiplexor ADG608 cuenta con una entrada de habilitación denotada como EN que está conectada a un nivel alto de manera que siempre está habilitado el multiplexor. La salida del multiplexor se conecta a la terminal del parámetro b de la compuerta lógica dinámica. De este modo se hace la programación de la selección de la funcionalidad.

5.5.2. Interconexión programable de compuertas

Constituye la segunda tarea de las que se encarga la estructura de ruteo programable. Dicha tarea consiste en proveer a cada entrada de cada compuerta lógica dinámica una señal que puede provenir de dos diferentes orígenes:

- **De una señal externa.** La señal a la entrada de la compuerta proviene del exterior a través del bloque de entrada.
- **De la salida de una compuerta lógica dinámica.** La señal entrante a la compuerta lógica dinámica proviene de la salida de otra compuerta lógica dinámica o incluso puede provenir de la salida de esa misma compuerta creando así, un lazo de realimentación.

El diagrama a bloques que se muestra en la Figura 5.10 corresponde a la forma en que fue implementada la interconexión programable de compuertas dentro del FPDGA. A continuación se explica específicamente para el caso de una compuerta lógica dinámica, pero debe

tenerse en cuenta que al ser idénticas las compuertas dentro del FPDGA, este procedimiento aplica para cualquier cantidad de compuertas lógicas dinámicas.

La interconexión programable de compuertas lógicas dinámicas se realiza de la siguiente manera. Primeramente, el bloque de entrada recibe la cantidad total de señales externas que pasan a un primer par de multiplexores digitales denotados como M1 y M2. Luego, un bloque contenedor almacena todas las salidas de las diferentes compuertas lógicas dinámicas; y estas salidas se conectan a un segundo par de multiplexores M3 y M4. En seguida, la salida del multiplexor M1 junto con la salida del multiplexor M3 se conectan a un quinto multiplexor M5, mientras que la salida del multiplexor M3 junto con la salida del multiplexor M4 se conectan a un sexto multiplexor M6. Un primer controlador C1 se utiliza para enviar una palabra de control de m bits a los multiplexores M1 y M3, y también envía un bit adicional al multiplexor M5. De manera similar, un segundo controlador C2 se utiliza para enviar una palabra de control de m bits a los multiplexores M2 y M4, y también envía un bit adicional al multiplexor M6. La salida del multiplexor M5 se conecta a la primera entrada de la compuerta lógica dinámica y la salida del multiplexor M6 se conecta a la segunda entrada de la compuerta lógica dinámica.

El diagrama esquemático de la implementación de la interconexión programable para una compuerta lógica dinámica se muestra en la Figura 5.11. Al ser idénticas cada una de las compuertas lógicas reconfigurables que forman el FPDGA, el proceso es el mismo para cada una de ellas. Para esta parte del diseño y construcción del dispositivo se utilizaron multiplexores 74AC151. Para hacer la selección de las señales que llegarán a las entradas de las compuertas, fueron utilizados interruptores. En otras palabras, los interruptores funcionan como los controladores C1 y C2 descritos previamente.

5.6. Bloque de entrada

Este bloque tiene como finalidad comunicar señales externas con el interior del FPDGA. Consta simplemente de un conector de ocho pines e igual cantidad de amplificadores operacionales en configuración seguidora. Un punto importante a resaltar es que al no contar con un mecanismo de protección, lo único que se pide a las señales de entrada es que sean de corriente directa y cuya amplitud no supere los voltajes de alimentación de los amplificadores operacionales para no llevarlos a un estado de saturación.

5.7. Bloque de salida

Este bloque permite la comunicación de señales provenientes del FPDGA con el exterior. En otras palabras, con el fin de observar las salidas de las compuertas lógicas dinámicas, el bloque de salida proporciona un medio para hacer el seguimiento de dichas señales. Consta de cuatro amplificadores operacionales en configuración seguidora conectados a un conector de igual número de pines.

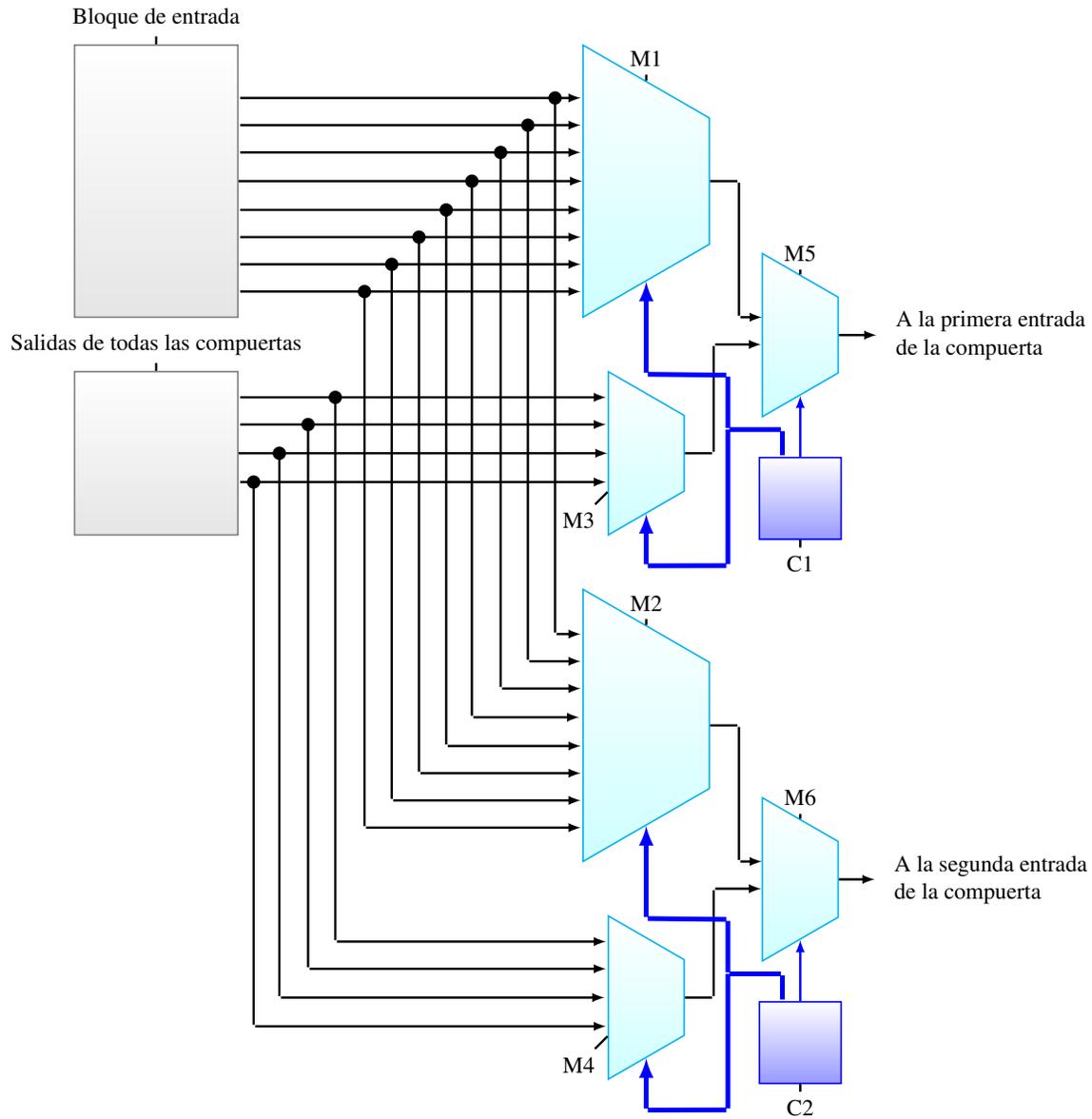


Figura 5.10: Diagrama de bloques de la implementación de la interconexión programable para una compuerta lógica reconfigurable.

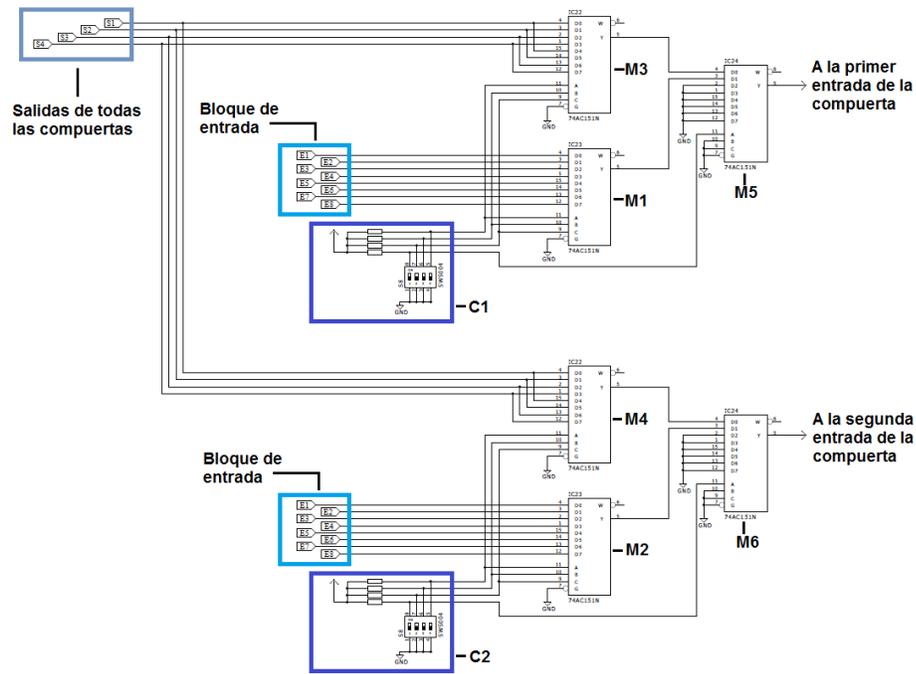


Figura 5.11: Diagrama esquemático de la implementación de la interconexión programable para una compuerta lógica reconfigurable.

5.8. Implementación electrónica de las partes que conforman el FPDGA

En las Secciones 5.4-5.7 de este capítulo, se ha explicado, en la medida de lo posible, cada uno de los bloques principales de los que consta el FPDGA y cómo fueron diseñados para que pudieran cumplir su función. El siguiente paso en la construcción del dispositivo FPDGA fue llevar a la realización electrónica dichos bloques por separado para su posterior integración. A continuación, se presenta una breve descripción de esos bloques y se muestran fotografías de la implementación de los mismos.

5.8.1. Fuentes de voltaje de uso general

Principalmente, se encargan de proveer los voltajes de alimentación al resto de la circuitería del dispositivo. Son fuentes bipolares que alimentan a los amplificadores operacionales con una tensión de ± 12 V; y al resto de los circuitos con 5 V. Pero además, entregan voltajes para el funcionamiento de los otros circuitos. Por ejemplo, suministran a cada compuerta lógica dinámica los voltajes de $a_1 = a_2 = 2$ V, $k = 1.5$ V, entre otros. En la Figura 5.12 (a), se muestra el circuito impreso construido para una fuente de voltaje de uso general, específicamente, se trata de la fuente de ± 12 V.

5.8.2. Fuente de voltaje de uso específico

Esta fuente es la encargada de proporcionar cada uno de los voltajes necesarios para realizar las cinco funciones lógicas de la compuerta. Es decir, suministra cada uno de los voltajes que se mostraron en la Tabla 5.2. El circuito impreso de esta fuente consta de reguladores de voltaje negativo y de resistencias variables (*trimpots*) para ajustar los voltajes de salida. Una fotografía del circuito impreso de esta fuente se encuentra en la Figura 5.12 (b).

5.8.3. Compuertas lógicas dinámicas

En un mismo circuito impreso fueron integradas cuatro compuertas lógicas dinámicas, las cuales representan el núcleo del FPDGA, ya que son las encargadas de hacer el cómputo de propósito general. La Figura 5.12 (c) muestra un arreglo de cuatro compuertas lógicas dinámicas.

5.8.4. Multiplexores analógicos

Forman parte de la estructura de ruteo programable. Proporcionan un medio para la selección de la funcionalidad de las compuertas. Cada multiplexor permite escoger uno de los cinco voltajes necesarios para realizar las diferentes operaciones lógicas. Este voltaje seleccionado será entregado al parámetro b de una compuerta lógica dinámica. En un mismo circuito impreso fueron integrados cuatro multiplexores analógicos, el cual se muestra en la Figura 5.12 (d).

5.8.5. Multiplexores digitales

También son parte de la estructura de ruteo programable. Proveen la manera de realizar la interconexión programable entre compuertas. Para cada entrada de una compuerta lógica dinámica se requiere de tres multiplexores digitales. Esto con la finalidad de poder seleccionar si a la entrada de la compuerta llegará una señal proveniente de un bloque de entrada o de la salida de una compuerta. En la Figura 5.12 (e), se muestra el arreglo total de multiplexores digitales para realizar la interconexión entre compuertas.

5.8.6. Interruptores

Proveen el recurso para programar tanto los multiplexores analógicos como los multiplexores digitales. En otras palabras, mediante estos interruptores se introducen las palabras de control que seleccionan la funcionalidad y la interconexión entre compuertas. Un arreglo de estos interruptores se encuentra en la Figura 5.12 (f).

5.8.7. Bloques de entrada y bloques de salida

Aunque dentro de la arquitectura del FPDGA se encuentran separados, la forma en que los bloques de entrada y los bloques de salida fueron implementados electrónicamente es

idéntica. Simplemente, se trata de un arreglo de amplificadores operacionales en configuración seguidora, o bien, esto mismo también se puede hacer con *buffers* triestados. Una fotografía del circuito impreso de un bloque de entrada se muestra en la Figura 5.12 (g).

5.9. Integración y validación del FPDGA

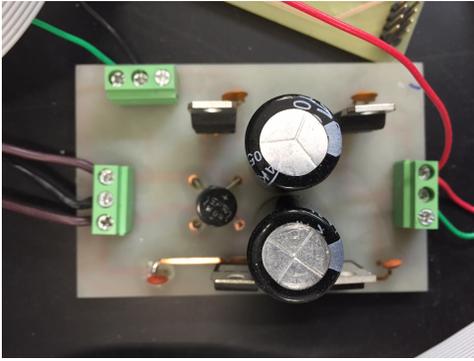
La última fase en la construcción física del FPDGA fue la integración de todos los circuitos que lo conforman. Esta fase consistió en colocar sobre una base de acrílico todos los circuitos impresos e interconectarlos. Posteriormente, se realizaron pruebas para detectar fallas en el diseño y soldadura de los circuitos impresos, o la interconexión de los mismos. Diferentes etapas de la fase de integración se muestran en la Figura 5.13.

Una vez que se integraron todos los componentes del FPDGA, la siguiente tarea fue la validación del prototipo. Para verificar que el dispositivo en su totalidad funcionara adecuadamente, se utilizó uno de los bloques de entrada para introducir una señal cuadrada de 1 kHz, en otro bloque de entrada se introdujo una señal de 2 kHz. Con esta combinación de señales se tienen las cuatro posibles combinaciones que se pueden hacer con dos bits. En seguida, se programaron los multiplexores digitales mediante los interruptores para que la señal de 1 kHz llegara a la primera entrada de las compuertas. De igual manera se hizo llegar la señal de 2 kHz a la segunda entrada de las compuertas. Posteriormente, se programaron los multiplexores analógicos para cambiar la funcionalidad de cada compuerta. Capturas de la pantalla del osciloscopio al realizar las mediciones a una compuerta se muestran en la Figura 5.14. En las Figuras 5.14 (a)-5.14 (e), el trazo superior corresponde a la señal de 1 kHz; el trazo medio corresponde a la señal de 2 kHz; y el trazo inferior corresponde a la salida de la compuerta.

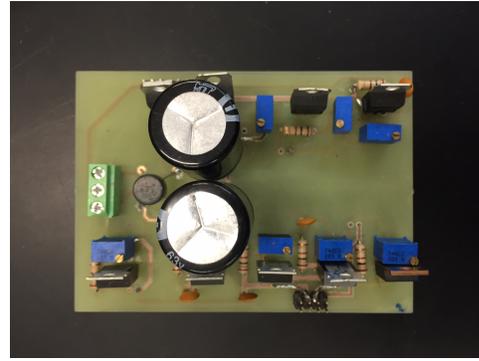
La obtención de la compuerta NOT se puede lograr por dos medios. Tomando como base las tablas de verdad de las compuertas NAND y NOR, mismas que se muestran en la Tabla 5.4, es posible observar que cuando las dos entradas tienen el mismo valor de verdad, la salida es la negación de dichas entradas. Por lo tanto, para que la compuerta lógica dinámica pueda emular la operación NOT, primeramente, se deben programar los multiplexores digitales para que la misma señal llegue a las dos entradas de la compuerta; y posteriormente, programar los multiplexores analógicos para que en el parámetro b de la compuerta aparezca el voltaje necesario para realizar la función NAND o NOR. La captura de la pantalla del osciloscopio al realizar las mediciones en una compuerta ejecutando la función lógica NOT se muestra en la Figura 5.14 (f).

x_1	x_2	NAND	NOR
0	0	1	1
0	1	1	0
1	0	1	0
1	1	0	0

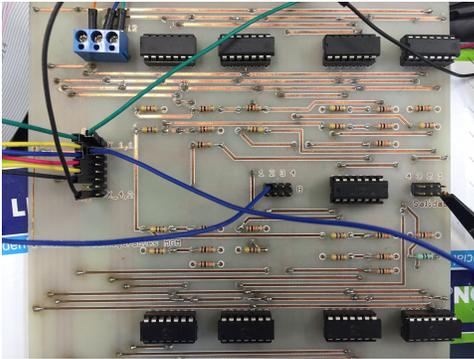
Tabla 5.4: Tablas de verdad de las funciones Booleanas NAND y NOR.



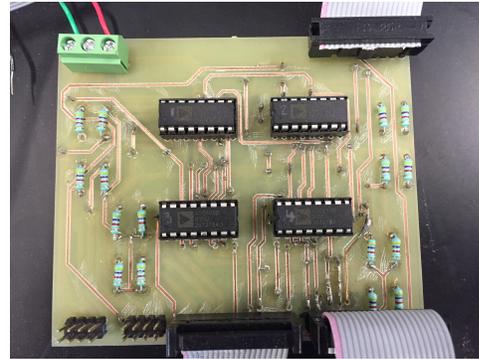
(a) Fuente de voltaje de uso general.



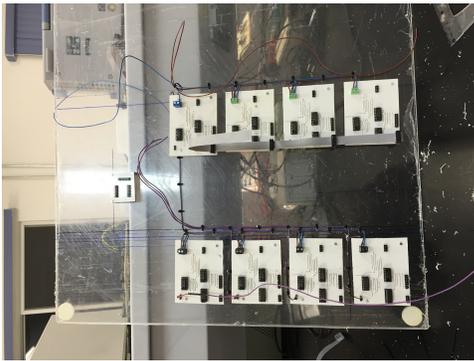
(b) Fuente de voltaje de uso específico.



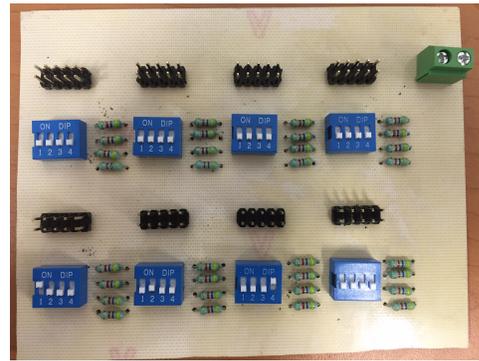
(c) Cuatro compuertas lógicas dinámicas.



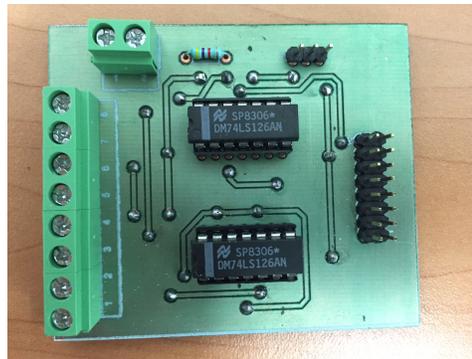
(d) Arreglo de multiplexores analógicos.



(e) Arreglo de multiplexores digitales.

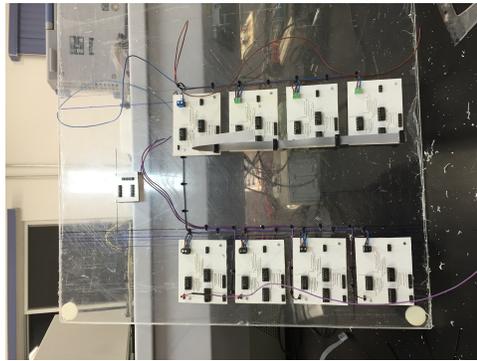


(f) Arreglo de interruptores.



(g) Bloque de entrada.

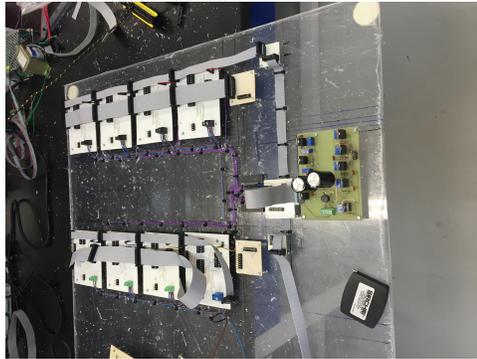
Figura 5.12: Implementación electrónica de los bloques del FPDGA.



(a) Colocación de los multiplexores digitales.



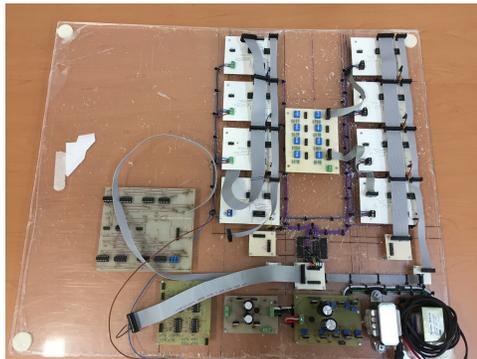
(b) Interconexión de los multiplexores digitales.



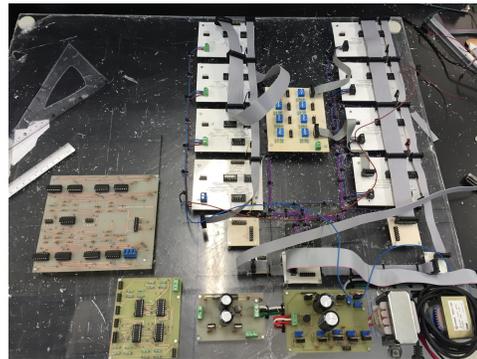
(c) Colocación de la fuente de uso específico.



(d) Colocación de la fuente de uso general.



(e) Colocación de los multiplexores analógicos.



(f) Colocación de las compuertas.

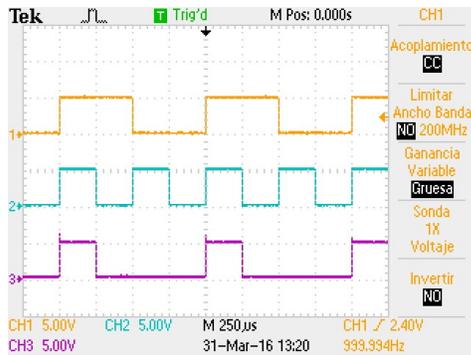


(g) Colocación de los interruptores.

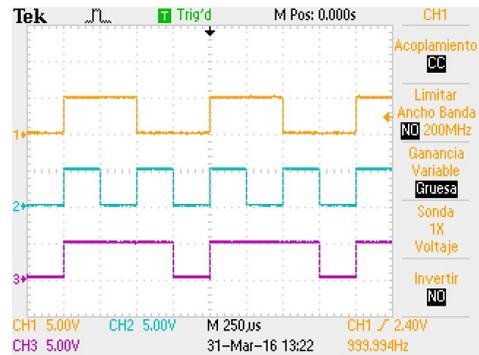


(h) Estructura de ruteo finalizada.

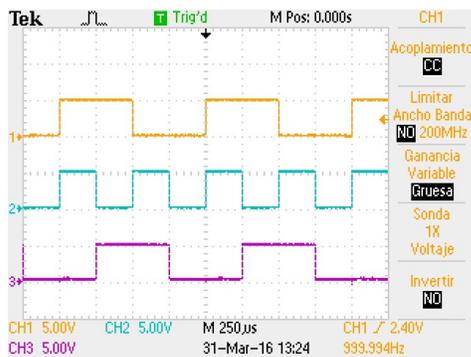
Figura 5.13: Integración de los bloques del FPDGA.



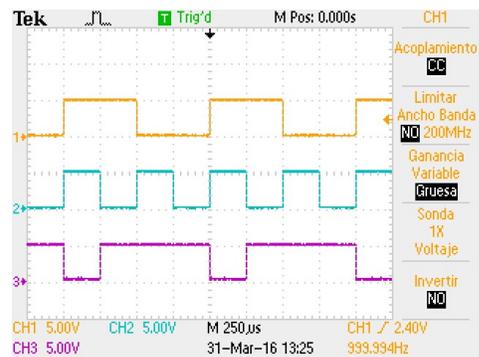
(a) Compuerta AND.



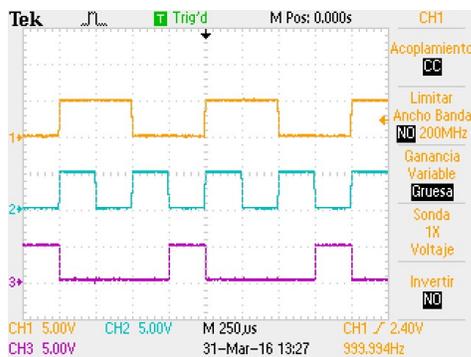
(b) Compuerta OR.



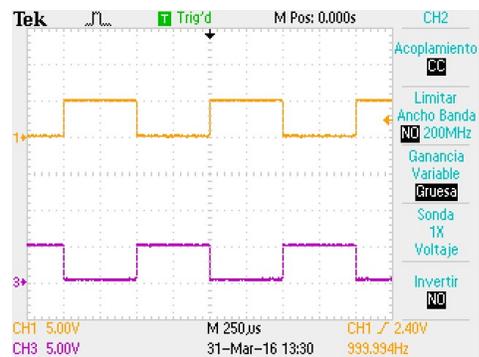
(c) Compuerta XOR.



(d) Compuerta NAND.



(e) Compuerta NOR.



(f) Compuerta NOT.

Figura 5.14: Validación de las compuertas lógicas dinámicas.

Sincronización en redes Booleanas autónomas

En el presente capítulo se aborda el fenómeno de sincronización desde varias perspectivas. Primeramente, en la Sección 6.1 se trata desde un punto de vista general debido a que la sincronización puede ser encontrada tanto en numerosos eventos de la naturaleza como en diferentes áreas del conocimiento humano. Además, se ha incluido en la Sección 6.2 una reseña histórica que ayude a entender desde cuándo la sincronización ha llamado el interés humano y cuáles han sido los principales trabajos científicos enfocados a su comprensión. Posteriormente, en la Sección 6.3 diferentes conceptos afines al área de sistemas dinámicos son presentados con la finalidad de hacer más entendible la sincronización de redes Booleanas autónomas. Finalmente, como parte medular de este capítulo, los resultados numéricos y experimentales del análisis de la sincronización de dos redes Booleanas autónomas bajo distintos esquemas de acoplamiento son discutidos en las secciones restantes.

6.1. Visión general del fenómeno de sincronización

La palabra sincronización a menudo es encontrada tanto en el ámbito científico como en el coloquial. Esta palabra proviene de dos vocablos griegos, *syn* y *cronos*, que significan común y tiempo, respectivamente [55]. Con base en lo anterior, se puede inferir que el significado que tuvo para los antiguos griegos fue: “ocurriendo al mismo tiempo”. Este significado se ha preservado hasta la actualidad y por sincronización se entiende entonces, a una correlación en tiempo de procesos diferentes.

Los fenómenos de sincronización son comunes en la vida diaria y es posible encontrarlos a diferentes escalas. Por ejemplo, las luciérnagas en los bosques destellan y dejan de hacerlo de manera síncrona; los aplausos del público en algún evento al principio son asíncronos pero luego de un lapso de tiempo, el ritmo de los aplausos se vuelve unísono. Incluso en el cuerpo humano es posible encontrar casos de sincronización. La epilepsia es uno de los trastornos neurológicos crónicos más comunes, la cual se caracteriza por descargas eléctricas paroxísticas (repentinas y recurrentes) de la corteza cerebral que dan lugar a alteraciones

intermitentes de la función cerebral [56]. Estas alteraciones consisten en el incremento y sincronización anormales de la actividad eléctrica neuronal. Es decir, el comportamiento sincronizado en la actividad del cerebro lejos de representar un signo de salud, representa un estado de enfermedad.

6.2. Reseña histórica

Desde un punto de vista histórico, la noción de la sincronización data del siglo IV a. de C. cuando Andróstenes de Thasos, escriba de Alejandro Magno, da testimonio de un sistema biológico rítmico. En su reporte de una expedición hecha a la India describe cómo las hojas de los árboles de tamarindo se mantenían abiertas durante el día, mientras que en la noche permanecían cerradas [57]. Sin embargo, es hasta el siglo XVII cuando se hace un análisis más formal del fenómeno de sincronización. En 1665, Huygens descubrió un extraño tipo de comportamiento entre dos péndulos conectados uno con el otro mediante una viga. Si los péndulos acoplados poseían pequeñas oscilaciones con las mismas condiciones iniciales o la diferencia inicial entre fases era nula, los dos péndulos se sincronizaban. Si la diferencia inicial entre fases era de 180 grados, los péndulos oscilaban exactamente a la misma frecuencia pero en anti-sincronización [58]. Huygens dedujo que la interacción crucial para este efecto se debía a los movimientos imperceptibles de la viga que sostenía a los dos péndulos, estos resultados fueron publicados en *Horologium oscillatorium: sive, De motu pendulorum ad horologia aptato demonstrationes geometricae* en el año 1673 [59].

A mediados del siglo XIX, John William Strutt, Lord Rayleigh, describe en su tratado *La Teoría del Sonido* el fenómeno de sincronización en sistemas acústicos. Rayleigh observó sincronización mutua en los órganos cuando dos tubos diferentes comenzaban a sonar al unísono, también observó el efecto de extinción cuando el acoplamiento llevaba a la supresión de las oscilaciones [55].

En los años 20 del siglo pasado el estudio de la sincronización estuvo enfocada al estudio de la propagación de las ondas eléctricas y de radio [58]. En 1920, W. H. Eccles y J. H. Vincent descubrieron el fenómeno de sincronización en el generador triodo, un dispositivo eléctrico basado en el tubo de vacío. Sin embargo, el estudio sistemático propiamente dicho del fenómeno de sincronización, tanto de manera teórica como experimental, comienza con E. Appleton en 1922 y B. van der Pol en 1927. Ellos mostraron que la frecuencia de un generador triodo puede ser sincronizada por una débil señal externa con una frecuencia ligeramente diferente. Estos trabajos fueron de gran importancia debido a que tales generadores se convirtieron en los elementos básicos de los sistemas de comunicación por radio.

En las últimas dos décadas del siglo XX hubo un surgimiento en el interés por analizar el comportamiento en sistemas acoplados. Una de esas investigaciones fue publicada por Fujisaka y Yamada en 1983 [60]. En su artículo muestran cómo dos osciladores caóticos idénticos bajo la variación de la fuerza de acoplamiento pueden lograr un estado de sincronización completa en el cual la dinámica del sistema acoplado ocupa un subespacio invariante del espacio fase total [61]. Además, muestran que si la fuerza de acoplamiento es debilitada, el sistema acoplado puede exhibir diferentes tipos de comportamientos no sincronizados. Los detalles de esta transición entre regímenes de movimientos fueron analizados para un par de osciladores de Lorenz acoplados. Por otro lado, en 1990 Pecora y Carroll [62] presentan un

criterio basado en los exponentes de Lyapunov para lograr la sincronización de dos sistemas caóticos conectados mediante una señal común. Desde su publicación, este artículo se ha convertido en un referente para la comunidad que estudia el fenómeno de sincronización, la cual se ha enfocado a desarrollar esquemas de control para lograr que dos o más sistemas dinámicos bajo ciertas restricciones logren un comportamiento sincronizado.

Los datos presentados en esta sección fueron elegidos por considerarlos de suma importancia para el estudio de la sincronización y están resumidos en la línea del tiempo mostrada en la Tabla 6.1.

6.3. Sincronización desde la perspectiva de sistemas dinámicos

En términos más específicos del área de sistemas dinámicos y de control, la sincronización puede definirse como:

- Un ajuste en los ritmos de objetos oscilatorios debido a su débil interacción [55].
- Un ajuste en las escalas de tiempo de las oscilaciones debido a la interacción entre procesos oscilatorios [63].
- El proceso en el cual dos o más agentes que interactúan, cada uno con condiciones iniciales diferentes, gradualmente aproximen sus estados conforme el tiempo avanza [64].
- Un proceso en el cual dos o más sistemas idénticos o no idénticos ajusten una cierta propiedad de su dinámica a un comportamiento común, debido a algún acoplamiento o forzamiento [65].
- Una completa o casi completa coincidencia de los estados de sistemas interactuando [66].

Las cuatro definiciones arriba mencionadas coinciden en que la sincronización implica el ajuste, coincidencia o aproximación de los estados de objetos interactuando. Además, cualquiera de estas definiciones es aplicable a todo tipo de sistema dinámico, es decir, aplica de igual manera para flujos, mapeos, autómatas o sistemas modelados con ecuaciones Booleanas con retardo como lo son las redes Booleanas autónomas.

6.4. Tipos de sistemas acoplados

Un primer aspecto a destacar al estudiar el fenómeno de sincronización de sistemas es la forma en que éstos están acoplados. Esto es importante porque existe una diferencia significativa en el proceso para llegar a estados sincronizados, dependiendo de la configuración de acoplamiento [65]. En este contexto, se pueden distinguir tres casos principales:



Tabla 6.1: Línea de tiempo sobre el estudio del fenómeno de sincronización.

- Sistemas con acoplamiento unidireccional o sistema maestro-esclavo. En este esquema, el sistema global acoplado está formado por un par de sistemas que se comportan como maestro y esclavo. Esto implica que uno de los sistemas se comporta libremente y dirige la evolución del otro. Como resultado, el sistema esclavo es perturbado con la dinámica o una función propia de la dinámica del sistema maestro, el cual actúa como una fuerza externa sobre el sistema esclavo. En general, es posible tener un sistema maestro y cualquier número de sistemas esclavos que pueden ser perturbados por el sistema maestro.

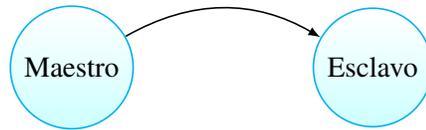


Figura 6.1: Esquema de un par de sistemas con acoplamiento unidireccional.

- Sistemas con acoplamiento bidireccional o mutuo. Este esquema está dado por un par de sistemas acoplados de tal manera que los sistemas son perturbados uno con la dinámica del otro. Este tipo de acoplamiento puede llevar a un comportamiento mutuo de sincronización.

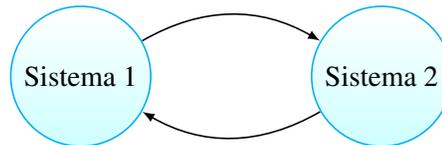


Figura 6.2: Esquema de un par de sistemas con acoplamiento bidireccional.

- Sistemas forzados por señal externa. En este tipo de acoplamiento dos o más sistemas son forzados a tener un comportamiento correlacionado mediante la inyección de una señal externa. Esta señal forzadora es independiente de los estados de los sistemas a forzar.

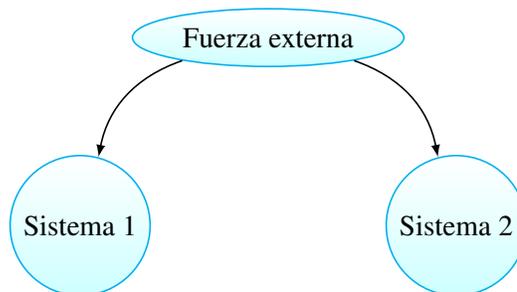


Figura 6.3: Esquema de un par de sistemas forzados por señal externa.

6.5. Tipos de sincronización

Un segundo aspecto importante en el estudio de la sincronización es el que se refiere al tipo de comportamiento sincronizado que los sistemas alcanzan, independientemente del tipo de acoplamiento con que estén conectados. Diferentes tipos de sincronización han sido estudiados y reportados. El caso más simple y obvio ocurre cuando los estados de todos los sistemas se comportan de manera idéntica. En otros casos, sólo correlaciones de algunos aspectos individuales de los sistemas se presentan, de tal modo que la sincronización es menos evidente. De acuerdo con [65], algunos de los diversos tipos de sincronización son: la sincronización completa o sincronización idéntica, la sincronización de fase, la sincronización con retardo, la sincronización generalizada, entre otras.

6.5.1. Sincronización completa

También es llamada sincronización idéntica. Consiste en una perfecta coincidencia de las trayectorias de los sistemas involucrados debido a la señal de acoplamiento. Este tipo de sincronización ocurre cuando las órbitas $\mathbf{x}^{(i)}(t)$ y $\mathbf{x}^{(j)}(t)$ satisfacen que:

$$\lim_{t \rightarrow \infty} |\mathbf{x}^{(i)}(t) - \mathbf{x}^{(j)}(t)| = 0, \quad (6.1)$$

para $i \neq j$, e $i, j \in \{1, 2, \dots, k\}$, con $\mathbf{x}^{(i)}(0) \neq \mathbf{x}^{(j)}(0)$ y $\mathbf{x}(t)$ es la trayectoria en el tiempo t , es decir, después de un tiempo transitorio, las trayectorias de los sistemas son idénticas e independientes de las condiciones iniciales.

6.5.2. Sincronización de fase

La sincronización en fase se utiliza para sistemas de osciladores no idénticos que pueden alcanzar un régimen intermedio donde se produce una unión de las fases, mientras la correlación entre las amplitudes permanece débil, es decir, amplitudes descorrelacionadas. La idea principal de la sincronización en fase es que la frecuencia puede ser forzada por una señal periódica externa, o al ser acoplada con otro oscilador. Este tipo de sincronización puede ser encontrada cuando las trayectorias oscilan alrededor de un valor constante. Para corroborar esto, primeramente, se define la fase de una trayectoria con la siguiente relación:

$$\phi_i(t) = \arctan \left[\frac{x_v^{(i)}(t)}{x_u^{(i)}(t)} \right], \quad (6.2)$$

donde $x_v^{(i)}(t), x_u^{(i)}(t) \in \mathbf{x}^{(i)}(t)$, y $v \neq u$. Si ϕ_i y ϕ_j con $i \neq j$ denotan las fases de las trayectorias $\mathbf{x}^{(i)}(t)$ y $\mathbf{x}^{(j)}(t)$, respectivamente, entonces la sincronización de fase ocurre cuando se satisface la siguiente desigualdad:

$$|m_1 \phi_i(t) - m_2 \phi_j(t)| < c, \quad (6.3)$$

para $m_1, m_2 \in \mathbb{Z}$ y c es una constante.

Por ejemplo, si dos péndulos se mueven en la misma dirección y si transcurrido un tiempo alcanzan la misma posición, entonces sus fases ϕ_1 y ϕ_2 están muy cerca de tal modo que

se dice que los péndulos están en sincronización en fase. Por lo contrario, si dos péndulos se mueven en dirección contraria y si el primer péndulo alcanza la extrema derecha, mientras que el segundo alcanza la extrema izquierda simultáneamente, entonces se dice que los péndulos están en antifase. Las Figuras 6.4 y 6.5 muestran la sincronización en fase y en antifase, respectivamente. Algunos trabajos referentes a este tipo de sincronización pueden ser consultados en [67–71].

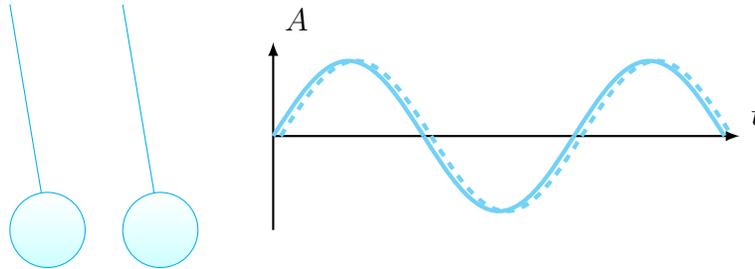


Figura 6.4: Régimen de sincronización en fase entre dos péndulos cuando la diferencia entre sus fases $\phi_2 - \phi_1 \approx 0$.

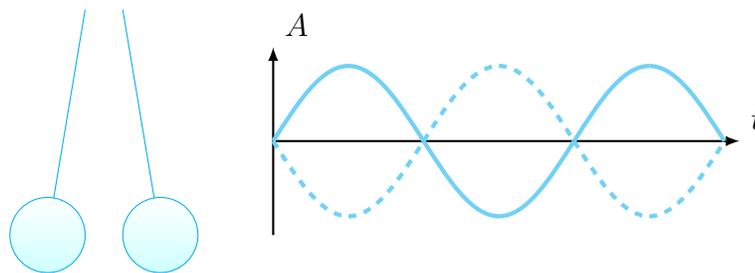


Figura 6.5: Régimen de sincronización en antifase entre dos péndulos cuando la diferencia entre sus fases $\phi_2 - \phi_1 = \pi$.

6.5.3. Sincronización con retardo o *lag*

Se puede ver a este tipo de sincronización como un paso entre la sincronización de fase y la sincronización completa e implica la existencia de un límite asintótico entre el estado $y(t)$ de un sistema en el tiempo t y el estado $x(t - \tau)$ del otro sistema retardado un tiempo τ . En otras palabras, la sincronización con retardo aparece como una coincidencia entre los estados de sistemas recorridos en el tiempo, $y(t) \rightarrow x(t - \tau), t \rightarrow \infty$, con τ positivo. Para ahondar más en este tipo de sincronización se puede consultar en [72–76]

6.5.4. Sincronización generalizada

Se dice que dos sistemas caóticos están en sincronización generalizada si existe una relación funcional entre los estados de los dos sistemas [65]. Es importante notar que en este tipo de sincronización los sistemas pueden ser no idénticos. En el caso particular del acoplamiento maestro-esclavo, el estado del subsistema esclavo, $y(t)$ es asintóticamente una

función del estado $x(t)$ del subsistema maestro. Por lo tanto, la sincronización generalizada ocurre cuando $y = h(x)$ [77]. Más detalles sobre este tópico pueden ser consultados en [78–81].

6.6. Fenómeno de sincronización en redes Booleanas autónomas

Tal cual se mencionó en el Capítulo 1, el objetivo general de esta tesis es sincronizar, tanto numéricamente como experimentalmente, dos redes Booleanas autónomas mediante diferentes esquemas de acoplamiento y validar las simulaciones numéricas mediante el arreglo programable en campo de compuertas dinámicas FPDGA cuyo diseño e implementación fue explicado en el Capítulo 5.

Antes de profundizar en los diferentes acoplamientos utilizados para analizar el fenómeno de sincronización en redes Booleanas autónomas, es preciso describir a detalle la red Booleana autónoma que fue tomada como base para dicho análisis.

La red Booleana autónoma base de esta tesis es la propuesta por Zhang *et al.* en [4]. Se trata de una red Booleana autónoma con tres nodos, cada uno de los cuales tiene dos entradas y una salida que se propaga a dos nodos diferentes. El tiempo que le toma a una señal propagarse al nodo i desde el nodo j se denota por τ_{ij} , con $i, j = 1, 2, 3$. Los nodos 1 y 2 ejecutan la operación lógica OR-exclusiva XOR, mientras que el nodo 3 ejecuta la operación OR-exclusiva negada XNOR. La red no presenta puntos fijos, lo que implica que la red siempre se mantiene oscilando. La topología de esta red se muestra en la Figura 6.6.

Específicamente hablando de la implementación electrónica de esta red, cada retardo del tiempo es una combinación del retardo intrínseco asociado con cada compuerta y del tiempo de la señal de propagación a lo largo del enlace. Para aumentar el tiempo de propagación, los autores incorporan un número par de compuertas negadoras NOT o de Schmitt *triggers* conectados en serie, cualquiera de estos dos procesos se comporta como un retardo en tiempo.

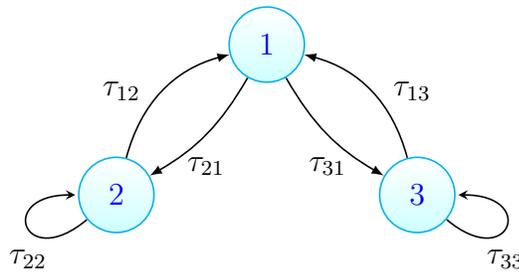


Figura 6.6: Red Booleana autónoma base.

Las ecuaciones Booleanas con retardo que describen el comportamiento de la red Booleana base son:

$$x_1(t) = x_2(t - \tau_{12}) \oplus x_3(t - \tau_{13}), \quad (6.4)$$

$$x_2(t) = x_1(t - \tau_{21}) \oplus x_2(t - \tau_{22}), \quad (6.5)$$

$$x_3(t) = x_1(t - \tau_{31}) \oplus x_3(t - \tau_{33}) \oplus 1, \quad (6.6)$$

donde x_i es el estado del i -ésimo nodo y el término $\oplus 1$ en la ecuación (6.6) ejecuta la función NOT.

6.6.1. Sincronización de redes Booleanas autónomas en acoplamiento unidireccional

El primer esquema que se analizó en este trabajo de tesis es el correspondiente a la sincronización de un par de redes Booleanas autónomas en acoplamiento unidireccional. Para estudiar este tipo de acoplamiento se consideraron dos redes Booleanas autónomas con la topología mostrada en la Figura 6.6. Antes de que ambas redes sean acopladas, cada una tiene su dinámica propia impuesta por sus condiciones iniciales. Por lo tanto, antes del acoplamiento, las ecuaciones Booleanas con retardo que describen la dinámica de la red que se comportará como maestro son:

$$x_1^M(t) = x_2^M(t - \tau_{12}) \oplus x_3^M(t - \tau_{13}), \quad (6.7)$$

$$x_2^M(t) = x_1^M(t - \tau_{21}) \oplus x_2^M(t - \tau_{22}), \quad (6.8)$$

$$x_3^M(t) = x_1^M(t - \tau_{31}) \oplus x_3^M(t - \tau_{33}) \oplus 1. \quad (6.9)$$

Mientras que la dinámica de la red que se comportará como esclava está dada por las ecuaciones Booleanas con retardo siguientes:

$$x_1^S(t) = x_2^S(t - \tau_{12}) \oplus x_3^S(t - \tau_{13}), \quad (6.10)$$

$$x_2^S(t) = x_1^S(t - \tau_{21}) \oplus x_2^S(t - \tau_{22}), \quad (6.11)$$

$$x_3^S(t) = x_1^S(t - \tau_{31}) \oplus x_3^S(t - \tau_{33}) \oplus 1. \quad (6.12)$$

La forma en que fueron acopladas las dos redes Booleanas autónomas se muestra en la Figura 6.7. La red de la izquierda se comporta como maestra y puede dirigir el comportamiento de la red Booleana de la derecha, la cual funge como esclava y cuya dinámica ya no depende de sus condiciones iniciales sino de la señal que la red maestra le transmite. Es importante mencionar que ambas redes son idénticas, lo que implica que los retardos de la red maestra son iguales a los retardos de la red esclava, es decir, $\tau_{ij}^M = \tau_{ij}^S$.

Para lograr acoplar las dos redes Booleanas autónomas como se muestra en la Figura 6.7, se partió del supuesto de que éstas alcanzan la sincronización completa. Esto implica que los estados entre los nodos correspondientes de las redes tengan el mismo valor de verdad por la acción de una señal de acoplamiento, la cual en esta aproximación involucra los estados de los nodos 1 de ambas redes. De tal modo que las siguientes ecuaciones se cumplen:

$$x_2^M(t - \tau_{12}) \oplus x_3^M(t - \tau_{13}) = x_2^S(t - \tau_{12}) \oplus x_3^S(t - \tau_{13}) \oplus f(x_2^M(t - \tau_{12}), x_3^M(t - \tau_{13}), x_2^S(t - \tau_{12}), x_3^S(t - \tau_{13}), \mathcal{C}), \quad (6.13)$$

$$x_1^M(t - \tau_{21}) \oplus x_2^M(t - \tau_{22}) = x_1^S(t - \tau_{21}) \oplus x_2^S(t - \tau_{22}), \quad (6.14)$$

$$x_1^M(t - \tau_{31}) \oplus x_3^M(t - \tau_{33}) \oplus 1 = x_1^S(t - \tau_{31}) \oplus x_3^S(t - \tau_{33}) \oplus 1. \quad (6.15)$$

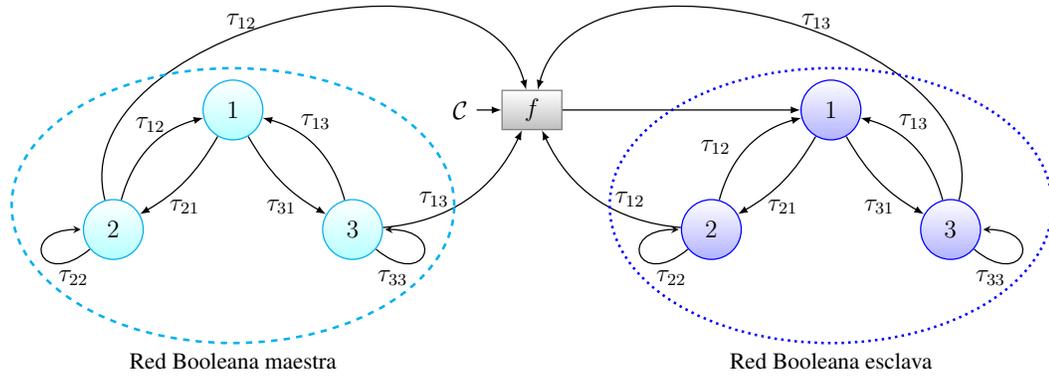


Figura 6.7: Acoplamiento unidireccional entre un par de redes Booleanas autónomas.

La función $f : \mathbb{B}^5 \rightarrow \mathbb{B}$ es la señal de acoplamiento y $\mathcal{C} \in \mathbb{B} = \{0, 1\}$ es una señal que habilita la acción de la señal de acoplamiento f . Si $\mathcal{C} = 0$, las redes se encuentran desacopladas, su comportamiento está descrito por las ecuaciones Booleanas con retardo (6.7)-(6.9) y (6.10)-(6.12), para la red maestra y esclava, respectivamente. Pero si $\mathcal{C} = 1$, las redes se encuentran acopladas y la señal de acoplamiento f comienza a actuar, es decir, la red maestra comienza a influir sobre la red esclava.

Ahora bien, el término $f(x_2^M(t - \tau_{12}), x_3^M(t - \tau_{13}), x_2^S(t - \tau_{12}), x_3^S(t - \tau_{13}), \mathcal{C})$ de la ecuación (6.13) puede ser dividido en dos partes: La primera de ellas es la parte que depende de las variables $x_2^M(t - \tau_{12}), x_3^M(t - \tau_{13}), x_2^S(t - \tau_{12}), x_3^S(t - \tau_{13})$, la cual se denotará como \mathcal{F} y la segunda es la parte dependiente de \mathcal{C} ; por lo que puede reescribirse como $f(\mathcal{F}, \mathcal{C})$. Si $\mathcal{C} = 0$, las redes Booleanas permanecen desacopladas, lo que implica que:

$$f(\mathcal{F}, 0) = 0, \quad (6.16)$$

pero si $\mathcal{C} = 1$, entonces la red maestra perturba a la red esclava, lo que significa que:

$$f(\mathcal{F}, 1) \mapsto \mathbb{B} = \{0, 1\}. \quad (6.17)$$

Una función Booleana que puede satisfacer (6.16) es la función AND, ya que cuando alguno de sus operandos es cero, su salida es cero. Pero además, también puede satisfacer (6.17) debido a que $f = \mathcal{F} \cdot 1 = \mathcal{F}$. Por lo tanto, ahora es preciso encontrar la función \mathcal{F} . Si se supone a $f = \mathcal{F} \cdot \mathcal{C}$, cuando $\mathcal{C} = 1$ la ecuación (6.13) se reduce a:

$$x_2^M(t - \tau_{12}) \oplus x_3^M(t - \tau_{13}) = x_2^S(t - \tau_{12}) \oplus x_3^S(t - \tau_{13}) \oplus \mathcal{F}. \quad (6.18)$$

Una primera manera de encontrar \mathcal{F} consiste en construir la tabla de verdad con las 16 posibles combinaciones de $x_2^M(t - \tau_{12}), x_3^M(t - \tau_{13}), x_2^S(t - \tau_{12}), x_3^S(t - \tau_{13})$ que son las variables de entrada, y los valores de \mathcal{F} que satisfacen la ecuación (6.18) como variable de salida. Dicha tabla de verdad se muestra en la Tabla 6.2.

Posteriormente, a partir de la Tabla 6.2 se construye el mapa de Karnaugh asociado con la finalidad de encontrar la forma más simple de \mathcal{F} . Este mapa se muestra en la Figura 6.8. Por simple inspección del mismo, se puede ver que no es posible agrupar ninguno de los

$x_2^M(t - \tau_{12})$	$x_3^M(t - \tau_{13})$	$x_2^S(t - \tau_{12})$	$x_3^S(t - \tau_{13})$	F
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	1
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

Tabla 6.2: Tabla de verdad para la función \mathcal{F} obtenida a partir de la ecuación (6.18).

minitérminos debido a que no están en posiciones contiguas, razón por la cual la función \mathcal{F} no puede ser simplificada y queda como se muestra en la ecuación (6.19).

$$\begin{aligned}
\mathcal{F} = & x_2^{IM}(t - \tau_{12}) \cdot x_3^{IM}(t - \tau_{13}) \cdot x_2^{IS}(t - \tau_{12}) \cdot x_3^S(t - \tau_{13}) \\
& + x_2^{IM}(t - \tau_{12}) \cdot x_3^{IM}(t - \tau_{13}) \cdot x_2^S(t - \tau_{12}) \cdot x_3^{IS}(t - \tau_{13}) \\
& + x_2^{IM}(t - \tau_{12}) \cdot x_3^M(t - \tau_{13}) \cdot x_2^{IS}(t - \tau_{12}) \cdot x_3^{IS}(t - \tau_{13}) \\
& + x_2^{IM}(t - \tau_{12}) \cdot x_3^M(t - \tau_{13}) \cdot x_2^S(t - \tau_{12}) \cdot x_3^S(t - \tau_{13}) \\
& + x_2^M(t - \tau_{12}) \cdot x_3^M(t - \tau_{13}) \cdot x_2^{IS}(t - \tau_{12}) \cdot x_3^S(t - \tau_{13}) \\
& + x_2^M(t - \tau_{12}) \cdot x_3^M(t - \tau_{13}) \cdot x_2^S(t - \tau_{12}) \cdot x_3^{IS}(t - \tau_{13}) \\
& + x_2^M(t - \tau_{12}) \cdot x_3^{IM}(t - \tau_{13}) \cdot x_2^{IS}(t - \tau_{12}) \cdot x_3^{IS}(t - \tau_{13}) \\
& + x_2^M(t - \tau_{12}) \cdot x_3^{IM}(t - \tau_{13}) \cdot x_2^S(t - \tau_{12}) \cdot x_3^S(t - \tau_{13}).
\end{aligned} \tag{6.19}$$

La segunda metodología para obtener la función \mathcal{F} es mucho más sencilla y directa. Basta con observar la ecuación (6.18) e inferir que:

$$\mathcal{F} = x_2^M(t - \tau_{12}) \oplus x_3^M(t - \tau_{13}) \oplus x_2^S(t - \tau_{12}) \oplus x_3^S(t - \tau_{13}). \tag{6.20}$$

Las ecuaciones (6.19) y (6.20) son equivalentes y se pueden utilizar de manera indistinta. Aunque si se desea llevarlas a su implementación electrónica, resulta mucho más conveniente la realización de (6.20) debido a que emplea un menor número de compuertas.

Una vez obtenida la función \mathcal{F} es necesario sustituirla en $f = \mathcal{F} \cdot \mathcal{C}$, dando como resultado:

		$x_2^S(t - \tau_{12})$		$x_3^S(t - \tau_{13})$	
		00	01	11	10
$x_2^M(t - \tau_{12})$ $x_3^M(t - \tau_{13})$	00	0	1	0	1
	01	1	0	1	0
	11	0	1	0	1
	10	1	0	1	0

Figura 6.8: Mapa de Karnaugh para la función \mathcal{F} de la ecuación (6.18).

$$f = [x_2^M(t - \tau_{12}) \oplus x_3^M(t - \tau_{13}) \oplus x_2^S(t - \tau_{12}) \oplus x_3^S(t - \tau_{13})] \cdot \mathcal{C}. \quad (6.21)$$

Para corroborar que la señal de acoplamiento f calculada favorece la sincronización completa de dos redes Booleanas autónomas en acoplamiento unidireccional, se realizaron las simulaciones numéricas correspondientes. En ellas, la señal de control \mathcal{C} se activa en $t = 50$ s, lo que da lugar a que la red maestra comience a dirigir el comportamiento de la red esclava por acción de la señal de acoplamiento f . Los valores de los retardos para esta simulación se muestran en la Tabla 6.3 y están expresados en segundos. Las condiciones iniciales para la red maestra son: $x_1^M(t) = x_2^M(t) = x_3^M(t) = 0 \forall t < 0$; mientras que para la red esclava, $x_1^M(t) = 1$, $x_2^M(t) = x_3^M(t) = 0 \forall t < 0$.

	τ_{12}	τ_{13}	τ_{21}	τ_{22}	τ_{31}	τ_{33}
$\tau^{M,S}$	π s	$\frac{\pi}{\sqrt{7}}$ s	$\sqrt{2}$ s	$\sqrt{5}$ s	$\sqrt{3}$ s	$\frac{\sqrt{5}}{\sqrt{2}}$ s

Tabla 6.3: Valores de los retardos τ_{ij} para la simulación de dos ABNs acopladas unidireccionalmente.

La Figura 6.9 muestra la manera en que los estados $x_1^M(t)$ y $x_1^S(t)$ se ajustan por la acción directa de la señal de acoplamiento f . La evolución temporal de $x_1^M(t)$ se encuentra graficada en la Figura 6.9 (a), mientras que la dinámica de $x_1^S(t)$ se ha graficado en la Figura 6.9 (b). En la Figura 6.9 (c) se muestra el error de sincronización entre los estados $x_1^M(t)$ y $x_1^S(t)$, es decir, $x_1^M(t) \oplus x_1^S(t)$. Este error converge a cero inmediatamente después de que la señal de control \mathcal{C} es activada en $t = 50$ s. La Figura 6.9 (d) corresponde a la gráfica de la señal de control \mathcal{C} . En el instante en que \mathcal{C} es llevada a un nivel “alto”, la señal de acoplamiento f , mostrada en Figura 6.9 (e), comienza a actuar sobre el nodo 1 de la red esclava. En esta última gráfica debe notarse que los picos o disparos repentinos que se aprecian denotan el forzamiento aplicado, mismos que desaparecen una vez que las dos redes han alcanzado la sincronización completa.

La Figura 6.10 muestra cómo los estados $x_2^M(t)$ y $x_2^S(t)$ se van ajustando por la acción indirecta de f . La evolución temporal de $x_2^M(t)$ está graficada en la Figura 6.10 (a), mientras

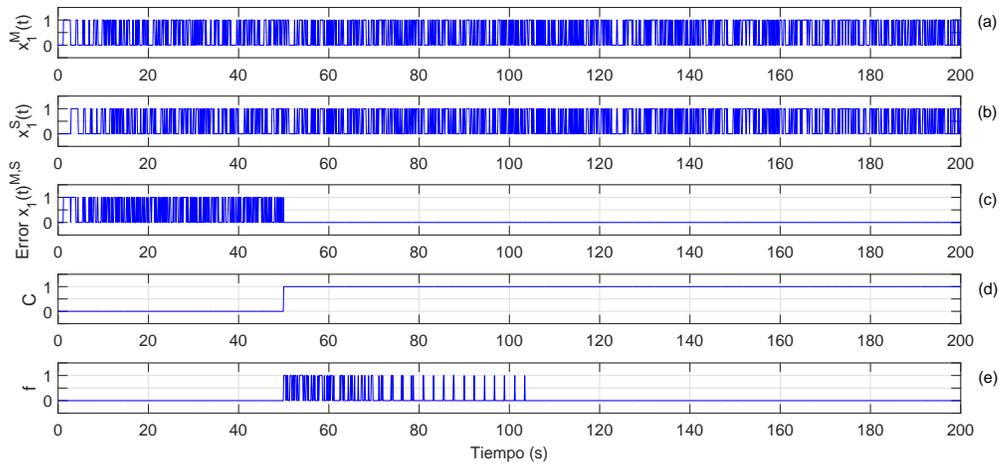


Figura 6.9: Ajuste de los estados $x_1^M(t)$ y $x_1^S(t)$ por la acción directa de f .

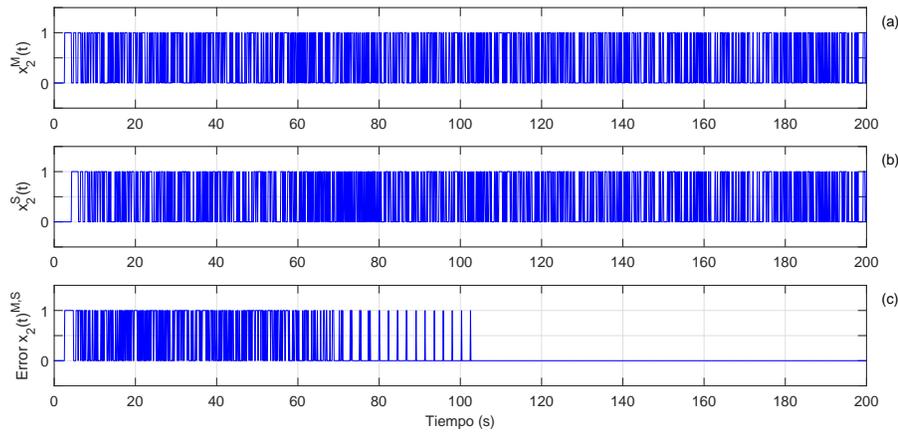


Figura 6.10: Ajuste de los estados $x_2^M(t)$ y $x_2^S(t)$ por la acción indirecta de f .

que la dinámica de $x_2^S(t)$ se encuentra en la Figura 6.10 (b). La medición del error de sincronización entre $x_2^M(t)$ y $x_2^S(t)$ se puede ver en la Figura 6.10 (c) y corresponde al cálculo de $x_2^M(t) \oplus x_2^S(t)$. En esta gráfica es posible apreciar breves instantes de desincronización que se van haciendo cada vez menos recurrentes, hasta que al final desaparecen.

En la Figura 6.11 se muestra el ajuste de los estados $x_3^M(t)$ y $x_3^S(t)$ por la acción indirecta de f . En la Figura 6.11 (a) está graficada la evolución temporal de $x_3^M(t)$, mientras que la dinámica de $x_3^S(t)$ se encuentra en la Figura 6.11 (b). El error de sincronización entre $x_3^M(t)$ y $x_3^S(t)$ se puede ver en la Figura 6.11 (c) y corresponde al cálculo de $x_3^M(t) \oplus x_3^S(t)$.

Cada una de las redes Booleanas autónomas a sincronizar se trata de una red conexa, esto quiere decir que para cualesquiera dos nodos existe un camino dirigido por el cual se puede transitar de ida y vuelta. Este aspecto es importante porque basta con aplicar la señal de acoplamiento en un único nodo para lograr que la red completa sincronice. Debido a que la señal de acoplamiento f viaja del nodo 1 de la red maestra solamente hacia el nodo 1 de la red esclava, estos nodos son los primeros en alcanzar la sincronización, lo cual ocurre

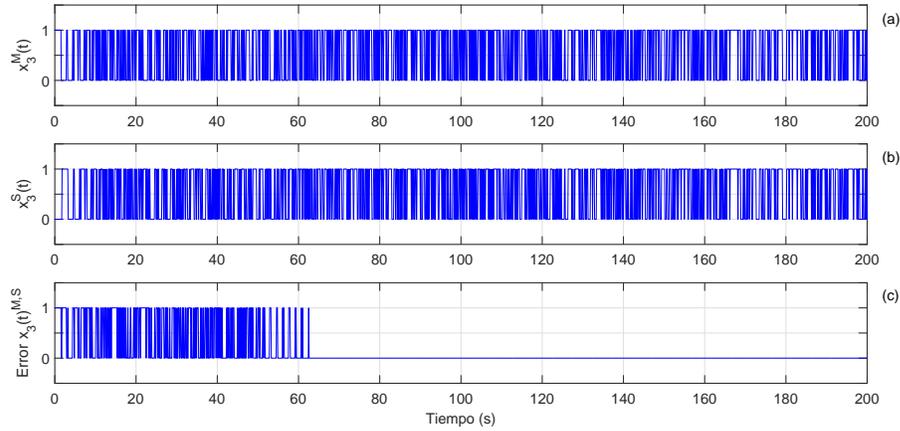


Figura 6.11: Ajuste de los estados $x_3^M(t)$ y $x_3^S(t)$ por la acción indirecta de f .

en $t = 50$ s. Los nodos 2 y 3 de la red esclava son controlados “virtualmente” por f . Es decir, indirectamente están siendo forzados a seguir la dinámica de los nodos 2 y 3 de la red maestra. Los nodos 2 de ambas redes alcanzan la sincronización en $t \sim 103$ s; mientras que los nodos 3, lo hacen en $t \sim 62$ s.

6.6.2. Sincronización de redes Booleanas autónomas en acoplamiento bidireccional

El siguiente esquema que se analizó fue el relacionado con acoplar dos redes Booleanas autónomas de forma bidireccional. La propuesta de acoplamiento se muestra en la Figura 6.12. La idea de esta propuesta es acoplar las dos redes por medio de dos señales, la primera denotada por f_1 que va del nodo 1 de la red 1 hacia el nodo 1 de la red 2; la segunda señal viaja del nodo 3 de la red 2 hacia el nodo 3 de la red 1. Es importante mencionar que ambas redes son idénticas, lo que implica que los retardos de la red 1 son iguales a los retardos de la red 2, es decir, $\tau_{ij}^1 = \tau_{ij}^2$.

Antes del acoplamiento, ambas redes se comportan independientes y su dinámica es descrita por el sistema de ecuaciones Booleanas con retardo (6.22)-(6.24) y (6.25)-(6.27), para la red 1 y la red 2, respectivamente.

$$x_1^1(t) = x_2^1(t - \tau_{12}) \oplus x_3^1(t - \tau_{13}), \quad (6.22)$$

$$x_2^1(t) = x_1^1(t - \tau_{21}) \oplus x_2^1(t - \tau_{22}), \quad (6.23)$$

$$x_3^1(t) = x_1^1(t - \tau_{31}) \oplus x_3^1(t - \tau_{33}) \oplus 1. \quad (6.24)$$

$$x_1^2(t) = x_2^2(t - \tau_{12}) \oplus x_3^2(t - \tau_{13}), \quad (6.25)$$

$$x_2^2(t) = x_1^2(t - \tau_{21}) \oplus x_2^2(t - \tau_{22}), \quad (6.26)$$

$$x_3^2(t) = x_1^2(t - \tau_{31}) \oplus x_3^2(t - \tau_{33}) \oplus 1. \quad (6.27)$$

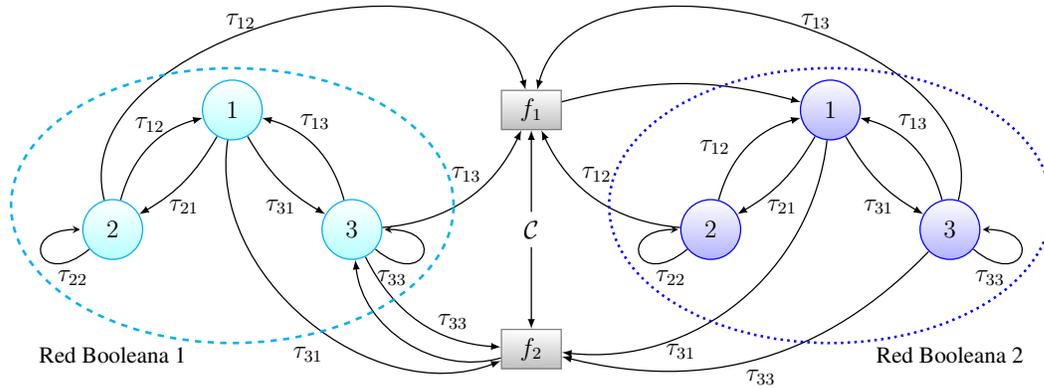


Figura 6.12: Acoplamiento bidireccional entre dos redes Booleanas autónomas.

De igual manera que se hizo para el caso del acoplamiento unidireccional, para acoplar las dos redes Booleanas autónomas en forma bidireccional se partió del supuesto de que éstas alcanzan la sincronización completa. Esto implica que los estados entre los nodos correspondientes de las redes tengan el mismo valor de verdad por la acción de un par de señales de acoplamiento. De tal modo que las siguientes ecuaciones se cumplen:

$$x_2^1(t - \tau_{12}) \oplus x_3^1(t - \tau_{13}) = x_2^2(t - \tau_{12}) \oplus x_3^2(t - \tau_{13}) \oplus f_1(x_2^1(t - \tau_{12}), x_3^1(t - \tau_{13}), x_2^2(t - \tau_{12}), x_3^2(t - \tau_{13}), C), \quad (6.28)$$

$$x_1^1(t - \tau_{21}) \oplus x_2^1(t - \tau_{22}) = x_1^2(t - \tau_{21}) \oplus x_2^2(t - \tau_{22}), \quad (6.29)$$

$$\begin{aligned} x_1^1(t - \tau_{31}) \oplus x_3^1(t - \tau_{33}) \oplus 1 \oplus f_2(x_1^1(t - \tau_{31}), x_3^1(t - \tau_{33}), x_1^2(t - \tau_{31}), x_3^2(t - \tau_{33}), C) &= x_1^2(t - \tau_{31}) \oplus x_3^2(t - \tau_{33}) \oplus 1, \end{aligned} \quad (6.30)$$

donde las funciones $f_1, f_2 : \mathbb{B}^5 \rightarrow \mathbb{B} = \{0, 1\}$ son las señales de acoplamiento y $C \in \mathbb{B} = \{0, 1\}$ es una señal de activación que precisa el momento en que f_1 y f_2 comienzan a actuar. Con este esquema, si $C = 0$, las redes se encuentran desacopladas, su comportamiento está descrito por las ecuaciones Booleanas con retardo (6.22)-(6.24) y (6.25)-(6.27), para la red 1 y red 2, respectivamente. Pero si $C = 1$, las redes se encuentran acopladas por medio de las funciones f_1 y f_2 .

El término $f_1(x_2^1(t - \tau_{12}), x_3^1(t - \tau_{13}), x_2^2(t - \tau_{12}), x_3^2(t - \tau_{13}), C)$ de la ecuación (6.28) puede ser dividido en dos partes: la primera de ellas es la parte que depende de las variables $x_2^1(t - \tau_{12}), x_3^1(t - \tau_{13}), x_2^2(t - \tau_{12}), x_3^2(t - \tau_{13})$, la cual se denotará como \mathcal{F}_1 y la segunda es la parte dependiente de C ; de tal forma que puede reescribirse como $f_1(\mathcal{F}_1, C)$. Si $C = 0$, las redes Booleanas deben permanecer desacopladas, lo que significa que:

$$f_1(\mathcal{F}_1, 0) = 0, \quad (6.31)$$

pero si $C = 1$, entonces la red 1 actúa sobre la red 2, lo que implica que:

$$f_1(\mathcal{F}_1, 1) \mapsto \mathbb{B} = \{0, 1\}. \quad (6.32)$$

Una función Booleana que puede satisfacer (6.31) es la función AND debido a que cuando alguno de sus operandos es cero, su salida es cero. Pero además, puede satisfacer (6.32) ya que $f_1 = \mathcal{F}_1 \cdot 1 = \mathcal{F}_1$. Por lo tanto, ahora es preciso encontrar la función \mathcal{F}_1 . Si se supone a $f_1 = \mathcal{F}_1 \cdot \mathcal{C}$, cuando $\mathcal{C} = 1$, la ecuación (6.28) se reduce a:

$$x_2^1(t - \tau_{12}) \oplus x_3^1(t - \tau_{13}) = x_2^2(t - \tau_{12}) \oplus x_3^2(t - \tau_{13}) \oplus \mathcal{F}_1. \quad (6.33)$$

De la ecuación (6.33) se puede determinar que:

$$\mathcal{F}_1 = x_2^1(t - \tau_{12}) \oplus x_3^1(t - \tau_{13}) \oplus x_2^2(t - \tau_{12}) \oplus x_3^2(t - \tau_{13}). \quad (6.34)$$

Una vez obtenida la función \mathcal{F}_1 es necesario sustituirla en $f_1 = \mathcal{F}_1 \cdot \mathcal{C}$, quedando de la siguiente manera:

$$f_1 = [x_2^1(t - \tau_{12}) \oplus x_3^1(t - \tau_{13}) \oplus x_2^2(t - \tau_{12}) \oplus x_3^2(t - \tau_{13})] \cdot \mathcal{C}. \quad (6.35)$$

Por otro lado, el término $f_2(x_1^1(t - \tau_{31}), x_3^1(t - \tau_{33}), x_1^2(t - \tau_{31}), x_3^2(t - \tau_{33}), \mathcal{C})$ de la ecuación (6.30) puede ser dividido en dos partes: la parte que depende de las variables $x_1^1(t - \tau_{31}), x_3^1(t - \tau_{33}), x_1^2(t - \tau_{31}), x_3^2(t - \tau_{33})$, la cual se denotará como \mathcal{F}_2 y la parte dependiente de \mathcal{C} ; de tal forma que se le puede reescribir como $f_2(\mathcal{F}_2, \mathcal{C})$. Si $\mathcal{C} = 0$, las redes Booleanas permanecen desacopladas, lo que significa que:

$$f_2(\mathcal{F}_2, 0) = 0, \quad (6.36)$$

pero si $\mathcal{C} = 1$, entonces la red 2 actúa sobre la red 1, lo que implica que:

$$f_2(\mathcal{F}_2, 1) \mapsto \mathbb{B} = \{0, 1\}. \quad (6.37)$$

La función Booleana AND satisface (6.36) debido a que cuando alguno de sus operandos es cero, su salida es cero. Pero además, puede satisfacer (6.37) ya que $f_2 = \mathcal{F}_2 \cdot 1 = \mathcal{F}_2$. Por lo tanto, ahora es preciso encontrar la función \mathcal{F}_2 . Si se supone a $f_2 = \mathcal{F}_2 \cdot \mathcal{C}$, cuando $\mathcal{C} = 1$, la ecuación (6.30) se reduce a:

$$x_1^1(t - \tau_{31}) \oplus x_3^1(t - \tau_{33}) \oplus 1 \oplus \mathcal{F}_2 = x_1^2(t - \tau_{31}) \oplus x_3^2(t - \tau_{33}) \oplus 1. \quad (6.38)$$

De la ecuación (6.38) se puede determinar que:

$$\mathcal{F}_2 = x_1^1(t - \tau_{31}) \oplus x_3^1(t - \tau_{33}) \oplus 1 \oplus x_1^2(t - \tau_{31}) \oplus x_3^2(t - \tau_{33}) \oplus 1, \quad (6.39)$$

la cual puede simplificarse aún más, de modo que queda como:

$$\mathcal{F}_2 = x_1^1(t - \tau_{31}) \oplus x_3^1(t - \tau_{33}) \oplus x_1^2(t - \tau_{31}) \oplus x_3^2(t - \tau_{33}). \quad (6.40)$$

Ahora que fue obtenida la función \mathcal{F}_2 se sustituye en $f_2 = \mathcal{F}_2 \cdot \mathcal{C}$, resultando en:

$$f_2 = [x_1^1(t - \tau_{31}) \oplus x_3^1(t - \tau_{33}) \oplus x_1^2(t - \tau_{31}) \oplus x_3^2(t - \tau_{33})] \cdot \mathcal{C}. \quad (6.41)$$

Para validar que dos redes Booleanas autónomas acopladas bidireccionalmente se sincronizan por la acción de las señales de acoplamiento f_1 y f_2 calculadas, se realizaron las simulaciones numéricas correspondientes. En ellas, la señal de activación \mathcal{C} se aplica en $t = 50$ s, lo que da lugar a que las señales de acoplamiento f_1 y f_2 comiencen a actuar. Los valores de los retardos para esta simulación se muestran en la Tabla 6.4 y se encuentran expresados en segundos. Las condiciones iniciales para la red 1 son: $x_1^1(t) = x_2^1(t) = x_3^1(t) = 0 \forall t < 0$; mientras que para la red 2, $x_1^2(t) = 1$, $x_2^2(t) = x_3^2(t) = 0 \forall t < 0$.

	τ_{12}	τ_{13}	τ_{21}	τ_{22}	τ_{31}	τ_{33}
$\tau^{1,2}$	π s	$\frac{\pi}{\sqrt{7}}$ s	$\sqrt{2}$ s	$\sqrt{5}$ s	$\sqrt{3}$ s	$\frac{\sqrt{5}}{\sqrt{2}}$ s

Tabla 6.4: Valores de los retardos τ_{ij} para la simulación de dos ABNs acopladas bidireccionalmente.

La Figura 6.13 muestra la manera en que los estados $x_1^1(t)$ y $x_1^2(t)$ gradualmente se ajustan por la acción de la señal de acoplamiento f_1 . La evolución temporal de $x_1^1(t)$ se encuentra graficada en la Figura 6.13 (a), mientras que la dinámica de $x_1^2(t)$ se halla graficada en la Figura 6.13 (b). En la Figura 6.13 (c) se muestra el error de sincronización entre los estados $x_1^1(t)$ y $x_1^2(t)$, es decir, $x_1^1(t) \oplus x_1^2(t)$. Este error converge a cero inmediatamente después de que la señal de activación \mathcal{C} es activada en $t = 50$ s. La Figura 6.13 (d) corresponde a la gráfica de la señal de activación \mathcal{C} . En el instante en que \mathcal{C} es llevada a un nivel “alto”, la señal de acoplamiento f_1 , mostrada en Figura 6.13 (e), comienza a actuar sobre el nodo 1 de la red 2. Los picos o disparos repentinos que se aprecian en Figura 6.13 (e) desaparecen una vez que las dos redes han alcanzado la sincronización completa.

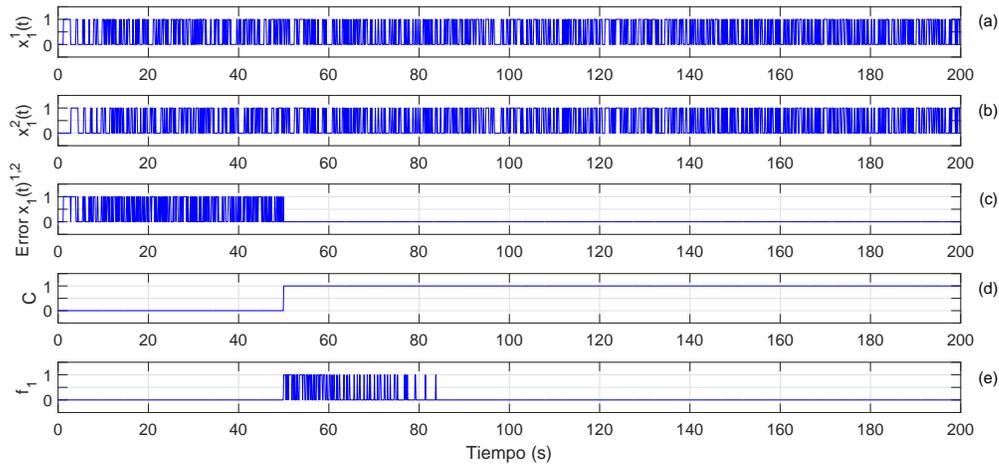


Figura 6.13: Ajuste de los estados $x_1^1(t)$ y $x_1^2(t)$ por la acción de f_1 .

La Figura 6.14 muestra la acción de la fuerza de acoplamiento f_2 para lograr el ajuste de los estados $x_3^1(t)$ y $x_3^2(t)$. La evolución temporal de $x_3^1(t)$ está graficada en la Figura 6.14 (a), mientras que la dinámica de $x_3^2(t)$ se encuentra en la Figura 6.14 (b). El error de sincronización entre $x_3^1(t)$, $x_3^2(t)$ se puede ver en la Figura 6.14 (c) y resulta del cálculo

de $x_3^1(t) \oplus x_3^2(t)$. Dicho error converge a cero inmediatamente después de que la señal de activación \mathcal{C} es aplicada en $t = 50$ s. La Figura 6.14 (d) corresponde a la gráfica de la señal de activación \mathcal{C} . En el instante en que \mathcal{C} es llevada a un nivel “alto”, la señal de acoplamiento f_2 , mostrada en Figura 6.14 (e), comienza a actuar sobre el nodo 3 de la red 1.

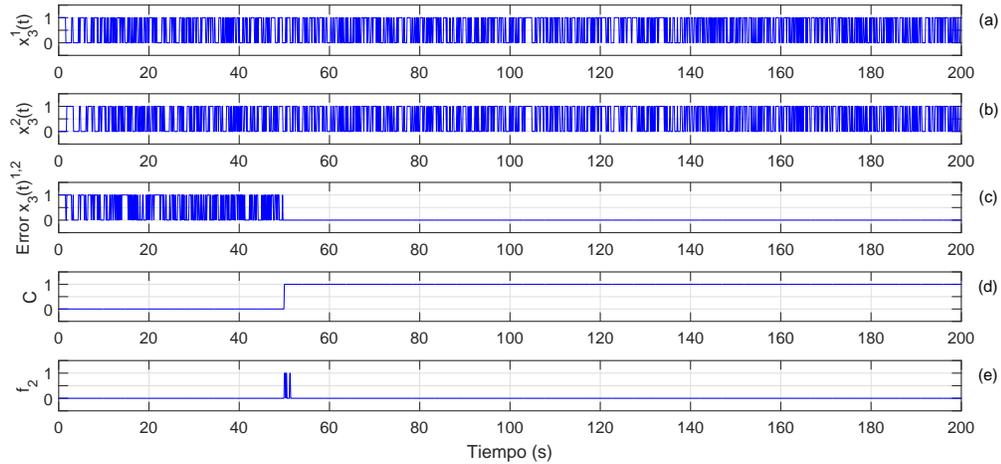


Figura 6.14: Ajuste de los estados $x_3^1(t)$ y $x_3^2(t)$ por la acción de f_2 .

La propuesta de acoplamiento bidireccional mostrado en la Figura 6.12 no considera una señal que controle directamente a los nodos 2. Por esta razón, estos nodos son los últimos en alcanzar un estado sincronizado, sin embargo, lo alcanzan debido a que el grafo asociado a la red es conexo, de tal manera que f_1 y f_2 actúan de forma indirecta sobre los nodos 2. La Figura 6.15 muestra el ajuste de los estados $x_2^1(t)$ y $x_2^2(t)$ por la acción indirecta de las fuerzas de acoplamiento f_1, f_2 . La evolución temporal de $x_2^1(t)$ está graficada en la Figura 6.15 (a), mientras que la dinámica de $x_2^2(t)$ se encuentra en la Figura 6.15 (b). El error de sincronización entre $x_2^1(t)$ y $x_2^2(t)$ se puede ver en la Figura 6.14 (c) y resulta del cálculo de $x_2^1(t) \oplus x_2^2(t)$. A diferencia de los errores en los estados controlados los cuales convergen a cero inmediatamente después de que la señal de control \mathcal{C} es activada, el error entre $x_2^1(t), x_2^2(t)$ converge a cero hasta $t \sim 82$ s. Este tiempo corresponde al instante en que desaparecen los picos repentinos en la gráfica de la fuerza de acoplamiento f_1 de la Figura 6.13 (e), lo que denota que efectivamente, las fuerzas de acoplamiento f_1 y f_2 indirectamente favorecen la sincronización.

Debido a que el esquema de acoplamiento bidireccional propuesto consta de dos señales conectando dos nodos diferentes, el tiempo que le tarda a un par de redes Booleanas autónomas acopladas bidireccionalmente es menor que el tiempo cuando éstas se encuentran acopladas de forma unidireccional. Esto puede verse en las gráficas presentadas, en acoplamiento unidireccional, las redes sincronizaron en $t \sim 103$ s; mientras que en acoplamiento bidireccional, la sincronización completa ocurre en $t \sim 82$ s.

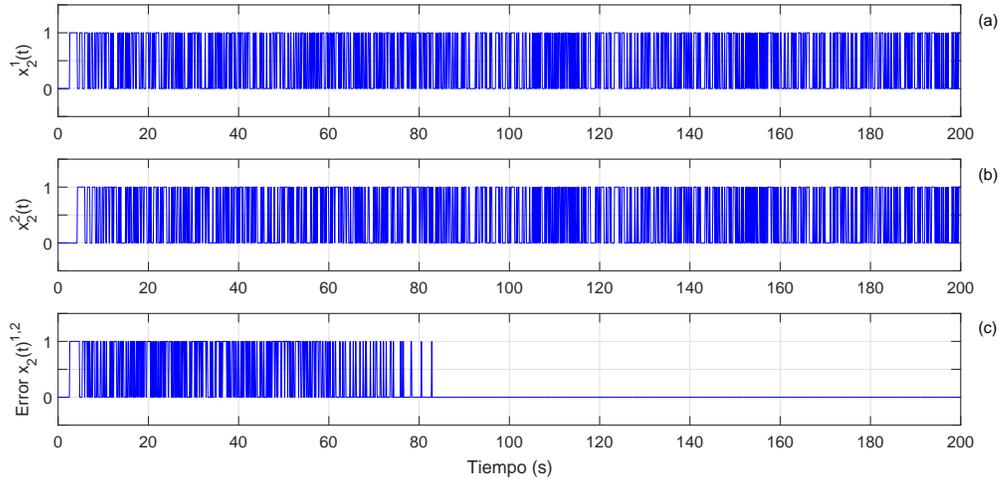


Figura 6.15: Ajuste de los estados $x_2^1(t)$ y $x_2^2(t)$ por la acción indirecta de f_1 y f_2 .

6.6.3. Sincronización de redes Booleanas forzadas por una señal externa

El último esquema de acoplamiento que se analizó en este trabajo de tesis es el relacionado con forzar dos redes Booleanas autónomas con una señal externa. La información aquí presentada puede ser consultada en [82].

La idea general de la sincronización forzada se logra considerando sistemas esclavos que son forzados por una señal externa como se muestra en la Figura 6.16. Bajo este esquema, k redes Booleanas autónomas controladas por una señal externa S_e pueden ser descritas como sigue:

$$\begin{aligned}
 \mathbf{x}^{(1)}(t) &= \mathbb{F}(x_1^{(1)}(t), x_2^{(1)}(t), \dots, x_n^{(1)}(t), S_e), \\
 \mathbf{x}^{(2)}(t) &= \mathbb{F}(x_1^{(2)}(t), x_2^{(2)}(t), \dots, x_n^{(2)}(t), S_e), \\
 &\vdots \\
 \mathbf{x}^{(k)}(t) &= \mathbb{F}(x_1^{(k)}(t), x_2^{(k)}(t), \dots, x_n^{(k)}(t), S_e),
 \end{aligned} \tag{6.42}$$

donde $\mathbf{x}^{(i)}(t) \in \mathbb{B}^n$, para $i = 1, \dots, k$. De tal manera que el fenómeno de sincronización por forzamiento externo ocurre cuando k redes Booleanas autónomas dadas por (6.42) muestran comportamiento correlacionado debido a una señal externa S_e .

Definición 6.6.1 *Las redes Booleanas autónomas $\mathbf{x}^{(i)}(t)$ y $\mathbf{x}^{(j)}(t)$, $i, j = 1, 2, \dots, k$, logran sincronización completa por forzamiento externo o simplemente sincronización forzada, si para cualquier $\mathbf{x}^{(i)}(0), \mathbf{x}^{(j)}(0) \in \mathbb{B}^n$, existe un tiempo $t_k \geq 0$, tal que $\mathbf{x}^{(i)}(t) = \mathbf{x}^{(j)}(t)$, para $t \geq t_k$.*

En el mismo sentido que [4], en esta tesis se propuso una topología de red Booleana autónoma que consiste de cinco nodos como se muestra en la Figura 6.17. Cada nodo tiene dos entradas y una salida que se propaga a dos nodos diferentes. Así en este caso particular,

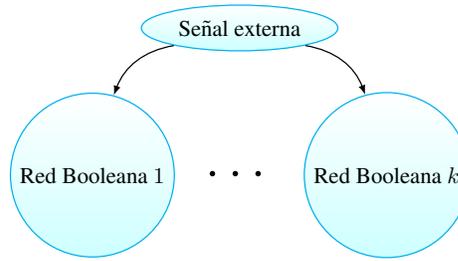


Figura 6.16: Esquema de forzamiento por señal externa para k redes Booleanas autónomas.

esta red Booleana autónoma puede ser descrita por el conjunto de funciones de activación local $f_i : \mathbb{B}^5 \rightarrow \mathbb{B}$, $i = 1, \dots, 5$, como sigue:

$$\begin{aligned}
 x_1(t) &= f_1(x_u(t - \tau_{iu}), x_v(t - \tau_{jv})), \\
 x_2(t) &= f_2(x_u(t - \tau_{iu}), x_v(t - \tau_{jv})), \\
 &\vdots \\
 x_5(t) &= f_5(x_u(t - \tau_{iu}), x_v(t - \tau_{jv})),
 \end{aligned} \tag{6.43}$$

donde $i, j, u, v = 1, \dots, 5$ y $x_{i_0}(t) = 0$ para toda la longitud de memoria del sistema. Los nodos 1 y 2 ejecutan la operación lógica XOR, el nodo 3 la operación lógica XNOR, mientras que los nodos 4 y 5 ejecutan la operación lógica OR. Para esta primera aproximación, debe notarse que un cero lógico es introducido a cada nodo que ejecuta la operación OR.

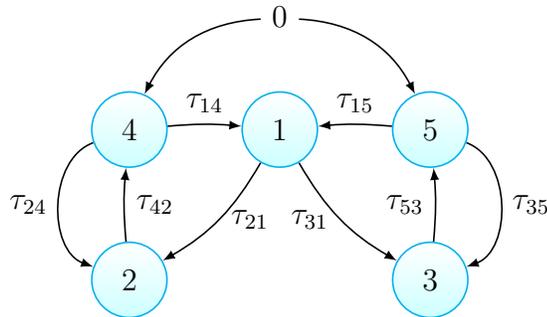


Figura 6.17: Red Booleana autónoma con dinámica compleja de 5 nodos.

Las ecuaciones Booleanas con retardo que describen a esta red Booleana autónoma son:

$$x_1(t) = x_4(t - \tau_{14}) \oplus x_5(t - \tau_{15}), \tag{6.44}$$

$$x_2(t) = x_1(t - \tau_{21}) \oplus x_4(t - \tau_{24}), \tag{6.45}$$

$$x_3(t) = x_1(t - \tau_{31}) \oplus x_5(t - \tau_{35}) \oplus 1, \tag{6.46}$$

$$x_4(t) = x_2(t - \tau_{42}) + 0, \tag{6.47}$$

$$x_5(t) = x_3(t - \tau_{53}) + 0. \tag{6.48}$$

Teorema 6.6.2 Para una red Booleana autónoma dada por el conjunto de ecuaciones (6.44)-(6.48), las trayectorias siempre permanecen oscilando.

Prueba. Para probar que esta red Booleana autónoma siempre genera oscilaciones en un espacio de fase infinito-dimensional, se debe mostrar que la red Booleana no tiene puntos fijos a los cuales la trayectoria pueda converger. Si una red con estados finitos tiene un punto fijo, entonces algunas de las trayectorias dejarán de oscilar debido a que éstas serán eventualmente fijas. Por contradicción, se demuestra este enunciado para una red Booleana autónoma. Se asume que existe un punto fijo $(x_1^*, x_2^*, \dots, x_5^*)$ del sistema, tal que:

$$\begin{aligned}x_1^*(t) &= x_1(t - \tau), \\x_2^*(t) &= x_2(t - \tau), \\x_3^*(t) &= x_3(t - \tau), \\x_4^*(t) &= x_4(t - \tau), \\x_5^*(t) &= x_5(t - \tau),\end{aligned}$$

para $t > \tau = \max\{\tau_{14}, \tau_{15}, \tau_{21}, \tau_{24}, \tau_{31}, \tau_{35}, \tau_{42}, \tau_{53}\}$.

Así, las ecuaciones (6.44)-(6.48) pueden ser reescritas como:

$$x_1(t) = x_4(t) \oplus x_5(t), \quad (6.49)$$

$$x_2(t) = x_1(t) \oplus x_4(t), \quad (6.50)$$

$$x_3(t) = x_1(t) \oplus x_5(t) \oplus 1, \quad (6.51)$$

$$x_4(t) = x_2(t), \quad (6.52)$$

$$x_5(t) = x_3(t). \quad (6.53)$$

La ecuación (6.52) implica que $x_4(t)$ tenga el mismo valor lógico que $x_2(t)$, y la ecuación (6.53) implica que $x_5(t)$ tiene el mismo valor lógico que $x_3(t)$. Usando este resultado se pueden reescribir las ecuaciones (6.49)-(6.51) como:

$$x_1(t) = x_2(t) \oplus x_3(t), \quad (6.54)$$

$$x_2(t) = x_1(t) \oplus x_2(t), \quad (6.55)$$

$$x_3(t) = x_1(t) \oplus x_3(t) \oplus 1. \quad (6.56)$$

Si se sustituye la ecuación (6.54) en (6.55) y (6.56), se obtiene:

$$x_2(t) = x_2(t) \oplus x_3(t) \oplus x_2(t), \quad (6.57)$$

$$x_3(t) = x_2(t) \oplus x_3(t) \oplus x_3(t) \oplus 1. \quad (6.58)$$

La ecuación (6.57) implica que $x_2(t) = x_3(t)$, pero la ecuación (6.58) implica que $x_3(t) = (x_2(t))'$. Por lo tanto, se tiene una contradicción, lo cual lleva a concluir que la red Booleana autónoma no tiene puntos fijos y oscilará permanentemente.

□

Para verificar la teoría anterior, se diseñó un circuito electrónico a través de simulación numérica utilizando SPICE con la topología mostrada en la Figura 6.17. Cada nodo de la red Booleana es una compuerta lógica dinámica discutida en la Sección 5.4. Los retardos en los enlaces se realizaron por medio de un circuito RC en configuración de filtro pasa-bajas. Con este circuito RC , se puede generar un conjunto de retardos de tiempo continuamente ajustables solamente variando los valores de R y C , aunque las distorsiones en la señal se vuelven más pronunciadas para retardos más grandes. Para un umbral lógico igual a la mitad de la diferencia entre el voltaje “alto” y el voltaje “bajo”, el retardo en tiempo está dado por $\tau = (\ln 2)RC$. Los valores de R y C usados para generar los diferentes retardos se muestran en la Tabla 6.5. El último renglón indica los valores de τ_{ij} . Los retardos τ_{42} y τ_{53} fueron considerados como un retardo intrínseco debido al tiempo de respuesta de la compuerta, el cual es ~ 60 ns .

	τ_{14}	τ_{15}	τ_{21}	τ_{24}	τ_{31}	τ_{35}
R	145 Ω	3.1 k Ω	1.1 k Ω	1.3 k Ω	2.3 k Ω	7.7 k Ω
C	22 pF	22 pF	22 pF	22 pF	22 pF	22 pF
τ	2.21 ns	47.27 ns	7.28 ns	19.82 ns	35.07 ns	117.41 ns

Tabla 6.5: Valores de los parámetros para generar los diferentes retardos τ_{ij} en los enlaces.

La Figura 6.18 muestra la evolución temporal de los voltajes de la red Booleana autónoma descrita previamente, la cual exhibe un comportamiento complejo y no repetitivo, posiblemente una manifestación de caos Booleano [4]. Como los nodos 4 y 5 ejecutan la operación lógica OR y una de sus entradas es un cero lógico, entonces estos nodos se comportan como *buffers* de retardo, es decir, las salidas de los nodos 4 y 5 son las mismas que las salidas de los nodos 2 y 3 respectivamente, pero retrasada un cierto tiempo para la red mostrada en la Figura 6.17.

La sincronización forzada de dos redes Booleanas autónomas se logra considerando dos sistemas esclavos dados por las ecuaciones (6.44)-(6.48) y añadiendo una señal Booleana externa forzadora. Bajo este esquema, el fenómeno de sincronización forzada ocurre cuando las oscilaciones de estos dos sistemas esclavos $\mathbf{x}^{(1)}(t)$ y $\mathbf{x}^{(2)}(t)$ muestran un comportamiento correlacionado debido a la señal externa S_e . La sincronización forzada completa puede ser determinada cuando el comportamiento asintótico de los sistemas esclavos está dada por:

$$\lim_{t \rightarrow \infty} |\mathbf{x}^{(1)}(t) \oplus \mathbf{x}^{(2)}(t)| = 0. \quad (6.59)$$

Por lo tanto, la sincronización forzada ocurre cuando las trayectorias $\mathbf{x}^{(1)}(t)$ y $\mathbf{x}^{(2)}(t)$ satisfacen la ecuación (6.59) con $\mathbf{x}^{(1)}(0) \neq \mathbf{x}^{(2)}(0)$, es decir, después de un tiempo transitorio, las trayectorias son idénticas e independientes de las condiciones iniciales.

En esta tesis, se acoplaron dos redes Booleanas autónomas esclavas como se muestra en la Figura 6.19, es decir, forzando a los nodos 4 y 5 de ambas redes con una señal Booleana externa, la cual tiene una frecuencia de 700 kHz.

Las redes Booleanas autónomas que están siendo forzadas pueden ser descritas como sigue:

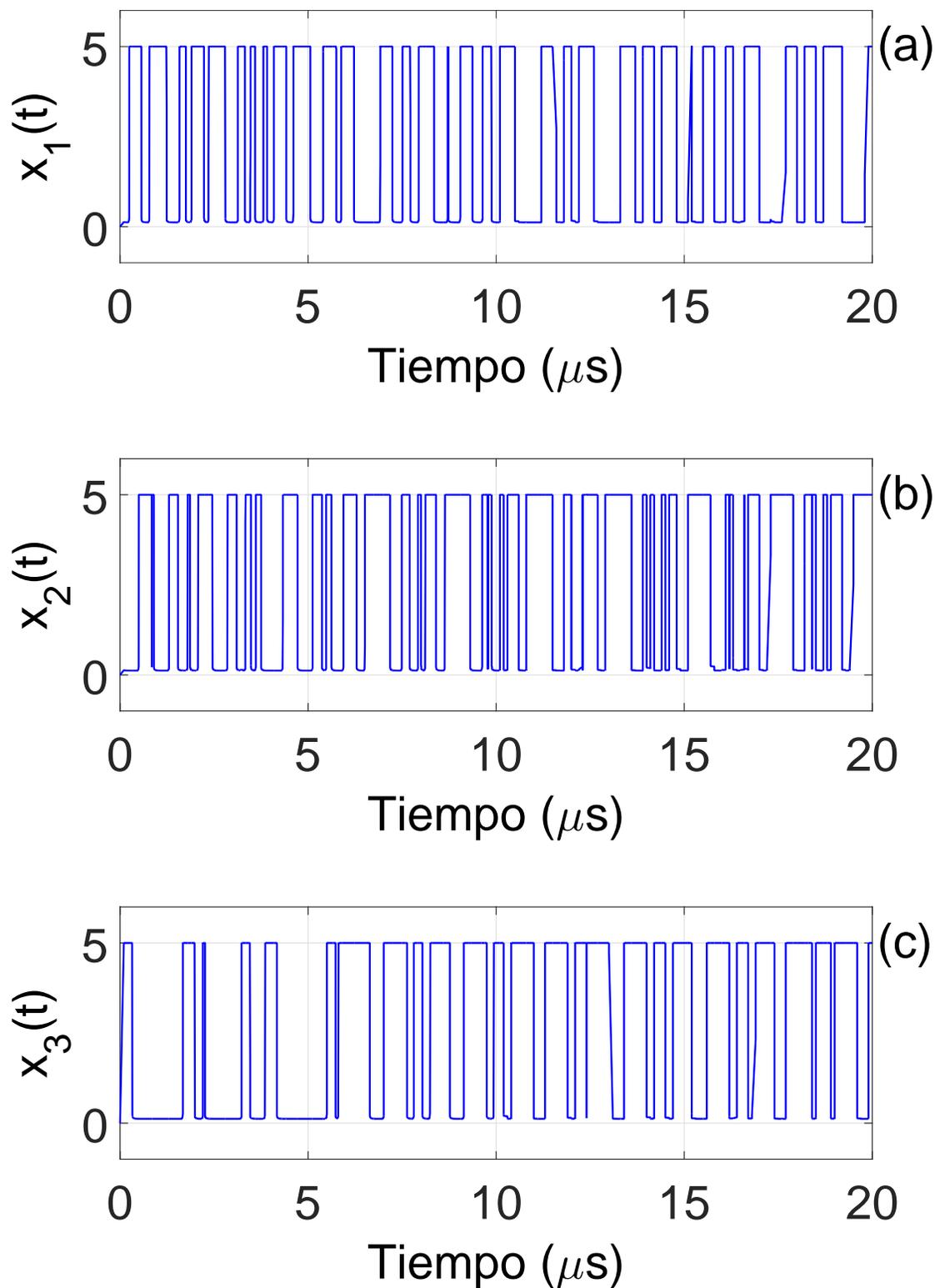
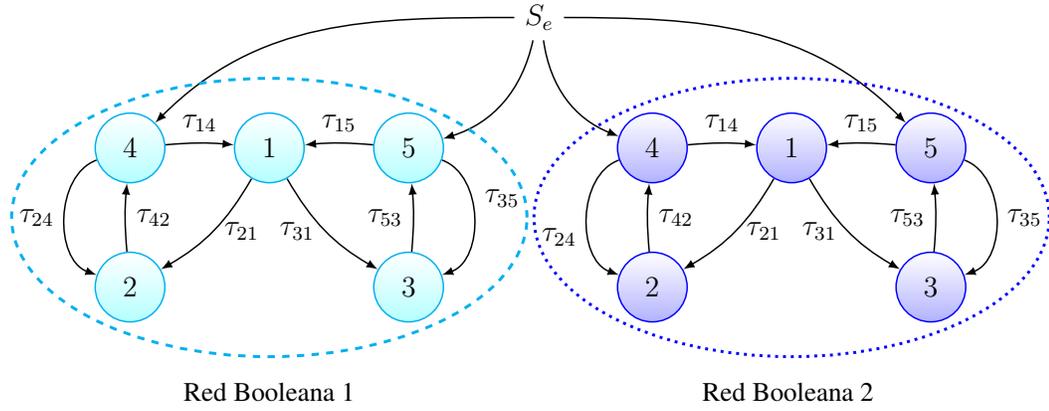


Figura 6.18: Evolución temporal en una ABN simulada medida en la salida de tres nodos. (a) Evolución temporal del nodo 1. (b) Evolución temporal del nodo 2. (c) Evolución temporal del nodo 3.


 Figura 6.19: Dos redes Booleanas autónomas controladas por una señal externa S_e .

$$x_1^i(t) = x_4^i(t - \tau_{14}) \oplus x_5^i(t - \tau_{15}), \quad (6.60)$$

$$x_2^i(t) = x_1^i(t - \tau_{21}) \oplus x_4^i(t - \tau_{24}), \quad (6.61)$$

$$x_3^i(t) = x_1^i(t - \tau_{31}) \oplus x_5^i(t - \tau_{35}) \oplus 1, \quad (6.62)$$

$$x_4^i(t) = x_2^i(t - \tau_{42}) + S_e, \quad (6.63)$$

$$x_5^i(t) = x_3^i(t - \tau_{53}) + S_e, \quad (6.64)$$

donde $i = 1, 2$.

Teorema 6.6.3 *Para una red Booleana autónoma forzada dada por el conjunto de ecuaciones (6.60)-(6.64), la señal externa determina los periodos de tiempo en los cuales las trayectorias están oscilando o convergiendo a un punto fijo.*

Prueba. La señal externa está oscilando entre cero y uno lógicos, entonces ésta tiene dos efectos diferentes sobre la red Booleana autónoma. Cuando la señal externa es cero lógico, los nodos 4 y 5 se comportan como fue explicado en el Teorema 6.6.2. Por lo tanto, las trayectorias siempre están oscilando. Sin embargo, cuando la señal externa es un uno lógico se necesita probar que la red Booleana autónoma tiene un punto fijo. Por lo tanto, se comienza describiendo el sistema como sigue:

$$x_1(t) = x_4(t - \tau_{14}) \oplus x_5(t - \tau_{15}), \quad (6.65)$$

$$x_2(t) = x_1(t - \tau_{21}) \oplus x_4(t - \tau_{24}), \quad (6.66)$$

$$x_3(t) = x_1(t - \tau_{31}) \oplus x_5(t - \tau_{35}) \oplus 1, \quad (6.67)$$

$$x_4(t) = x_2(t - \tau_{42}) + 1, \quad (6.68)$$

$$x_5(t) = x_3(t - \tau_{53}) + 1. \quad (6.69)$$

Nuevamente, se asume que $(x_1^*, x_2^*, \dots, x_5^*)$ es un punto fijo del sistema, tal que:

$$\begin{aligned}
x_1^*(t) &= x_1(t - \tau), \\
x_2^*(t) &= x_2(t - \tau), \\
x_3^*(t) &= x_3(t - \tau), \\
x_4^*(t) &= x_4(t - \tau), \\
x_5^*(t) &= x_5(t - \tau),
\end{aligned}$$

para $t > \tau = \max\{\tau_{14}, \tau_{15}, \tau_{21}, \tau_{24}, \tau_{31}, \tau_{35}, \tau_{42}, \tau_{53}\}$.

Así, las ecuaciones (6.65)-(6.69) pueden ser reescritas como sigue:

$$x_1(t) = x_4(t) \oplus x_5(t), \quad (6.70)$$

$$x_2(t) = x_1(t) \oplus x_4(t), \quad (6.71)$$

$$x_3(t) = x_1(t) \oplus x_5(t) \oplus 1, \quad (6.72)$$

$$x_4(t) = 1, \quad (6.73)$$

$$x_5(t) = 1. \quad (6.74)$$

Las ecuaciones (6.73) y (6.74) implican que $x_4(t) = 1$, $x_5(t) = 1$, por lo que las ecuaciones (6.70)-(6.74) pueden ser reescritas como:

$$x_1(t) = 1 \oplus 1, \quad (6.75)$$

$$x_2(t) = 1 \oplus 1 \oplus 1, \quad (6.76)$$

$$x_3(t) = 1 \oplus 1 \oplus 1 \oplus 1. \quad (6.77)$$

Por lo tanto, cuando la señal externa es un uno lógico, existe un único punto fijo en $x_1(t) = 0$, $x_2(t) = 1$, $x_3(t) = 0$, $x_4(t) = 1$, $x_5(t) = 1$.

□

Corolario 6.6.4 *Dos redes Booleanas autónomas forzadas descritas por el conjunto de ecuaciones (6.60)-(6.64) presentan sincronización completa cuando son forzadas por una señal Booleana externa.*

Cuando la señal externa es un uno lógico, ésta fuerza a los nodos 4 y 5 a actualizar sus salidas a un uno lógico. Cada vez que la señal externa cambia a uno, los estados lógicos del resto de los nodos de ambas redes Booleanas se van acercando hasta el tiempo en que la sincronización ocurre y todos los estados de las dos redes son iguales.

La Figura 6.20 muestra simulaciones numéricas de la evolución temporal de los voltajes de dos redes Booleanas medidas en las salidas de tres de los nodos. Se puede ver que el comportamiento complejo mostrado por cada nodo se va a un comportamiento periódico casi idéntico en ambas redes inmediatamente después de que la señal forzadora es aplicada. Además, debe notarse que el ciclo de trabajo de las ondas medidas en los nodos puede ser

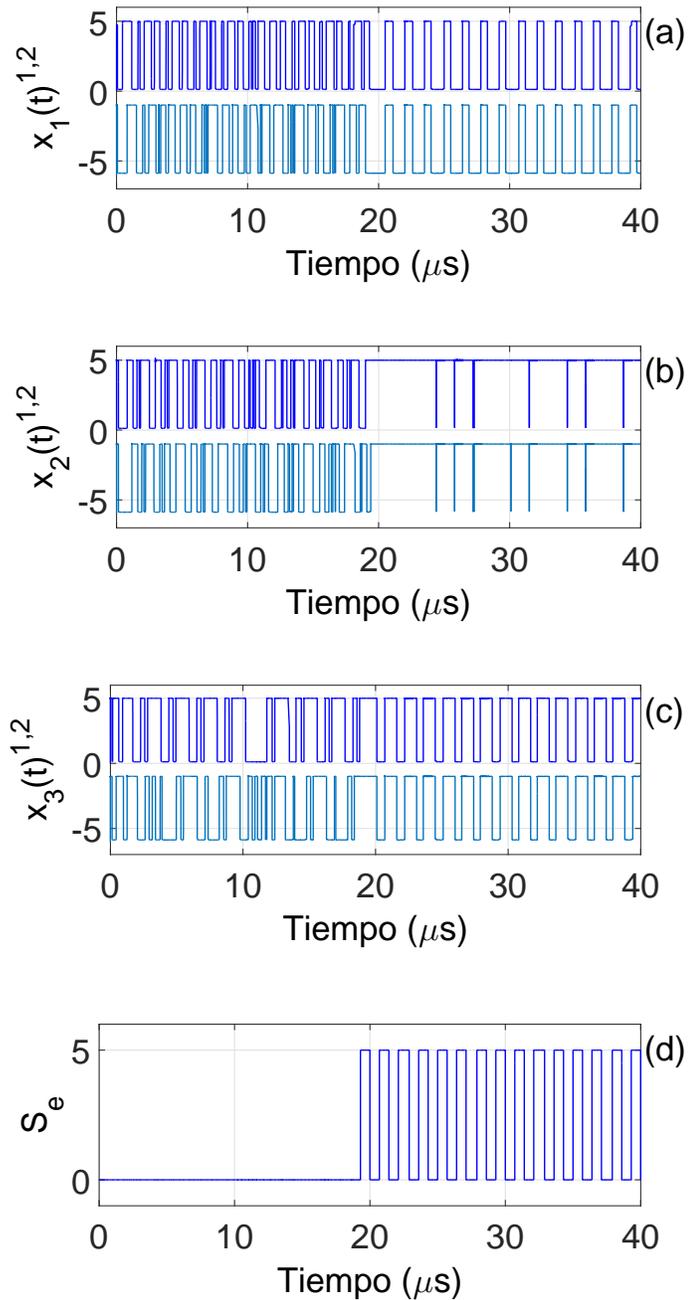


Figura 6.20: Evolución temporal de los voltajes de dos ABNs forzadas realizadas numéricamente. (a) Salidas de los nodos 1. (b) Salidas de los nodos 2. (c) Salidas de los nodos 3. En los tres casos la señal superior corresponde a la red 1, mientras que la señal inferior corresponde a la red 2 y esta señal tiene un *offset* de -6 V. (d) Señal forzadora externa.

diferente del ciclo de trabajo de la señal forzadora (50%), demostrando que la respuesta de cada nodo de la red Booleana autónoma no es igual a la señal forzadora para cualquier condición inicial.

Para detectar la sincronización forzada de dos redes Booleanas autónomas esclavas, se utilizaron compuertas XOR auxiliares cuyas entradas son los nodos correspondientes de ambas redes. Cuando la sincronización completa es alcanzada, la salida de estas compuertas auxiliares es un cero lógico. La Figura 6.21 muestra simulaciones de la evolución temporal de las salidas de las compuertas XOR auxiliares, donde se puede corroborar la sincronización completa después de un transitorio. Debe notarse que existen eventos de desincronización debidos a que la señal externa fuerza directamente a los nodos 4 y 5 a un uno lógico, e indirectamente fuerza al resto de los nodos de ambas redes a alcanzar un estado común. Después de un tiempo transitorio, la sincronización ocurre y todos los estados de ambas redes son iguales.

La red Booleana autónoma ilustrada en la Figura 6.17 fue realizada experimentalmente utilizando el FPDGA. Los retardos fueron hechos por medio de un filtro pasa-bajas. La Figura 6.22 muestra la evolución temporal de los voltajes de una red Booleana autónoma experimental.

Los valores de τ_{ij} utilizados para el experimento se muestran en las dos últimas líneas de la Tabla 6.6. Las etiquetas τ^1 y τ^2 corresponden a los valores utilizados para la primera y la segunda red Booleana autónoma forzada, respectivamente.

	τ_{14}	τ_{15}	τ_{21}	τ_{24}	τ_{31}	τ_{35}	τ_{42}	τ_{53}
τ^1	110 ns	232 ns	340 ns	150 ns	437 ns	220 ns	47 ns	51 ns
τ^2	109 ns	223 ns	352 ns	145 ns	419 ns	224 ns	40 ns	47 ns

Tabla 6.6: Valores de parámetros para generar los diferentes retardos τ_{ij} en los enlaces para dos redes experimentales.

El caos está caracterizado por la divergencia exponencial de trayectorias con condiciones iniciales muy cercanas, lo cual está indicado por un exponente de Lyapunov positivo. En [4] proponen un método para estimar el exponente de Lyapunov para sistemas dinámicos basados en ecuaciones en diferencias, la métrica utilizada para medir la divergencia de las trayectorias fue propuesta en [20] como sigue:

$$d(s) = \frac{1}{T} \int_s^{s+T} x(t' + t_a) \oplus y(t' + t_b) dt', \quad (6.78)$$

donde la distancia Booleana $d(s)$ es calculada sobre el intervalo T .

En esta tesis este método fue utilizado para estimar el exponente de Lyapunov más grande de una red Booleana autónoma implementada experimentalmente. El método está descrito por el siguiente algoritmo:

- Adquirir una serie de tiempo con suficientes datos V de voltaje de la red Booleana autónoma experimental.
- Transformar la serie de tiempo V en un serie Booleana de tiempo $x(t)$.

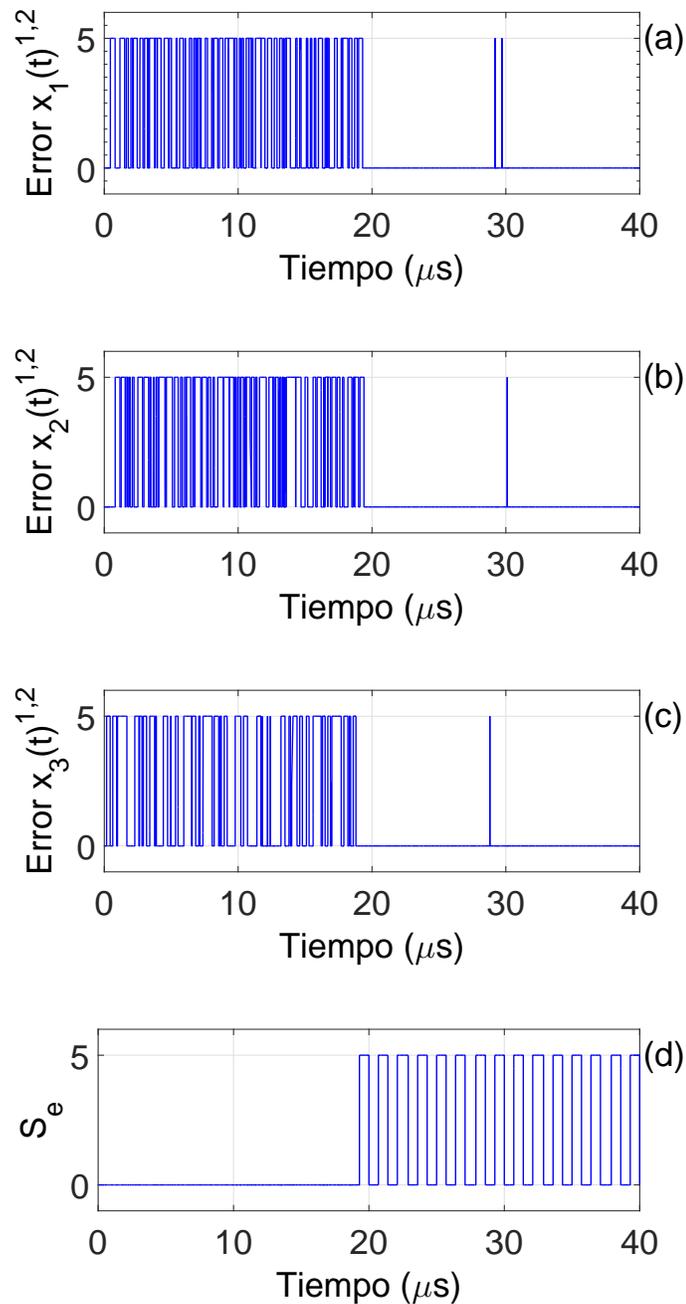


Figura 6.21: Evolución temporal de los errores entre nodos correspondientes de dos ABNs forzadas y simuladas numéricamente. (a) Error entre nodos 1. (b) Error entre nodos 2. (c) Error entre nodos 3. (d) Señal externa usada para forzar ambas ABNs.

- Para un $\delta > 0$ dado y $T = T_0$, buscar segmentos de $x(t)$ que comiencen en tiempo t_a y t_b tales que $d_0 = d(s) < \delta$.
- Calcular $d(s)$ para $s \in [0, k * T_0]$, utilizando los segmentos encontrados en el paso previo, para $k > 0$.

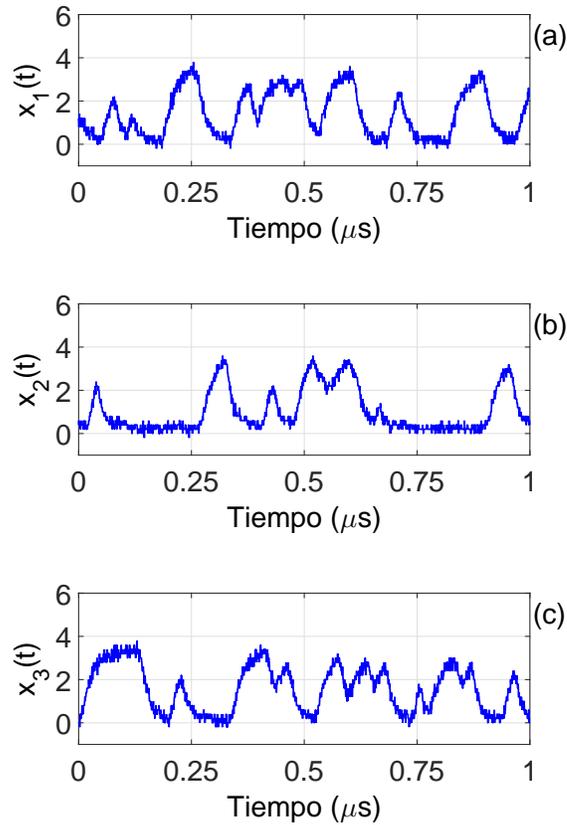


Figura 6.22: Evolución temporal de una ABN experimental medida en las salidas de tres de sus nodos. (a) Salida del nodo 1. (b) Salida del nodo 2. (c) Salida del nodo 3.

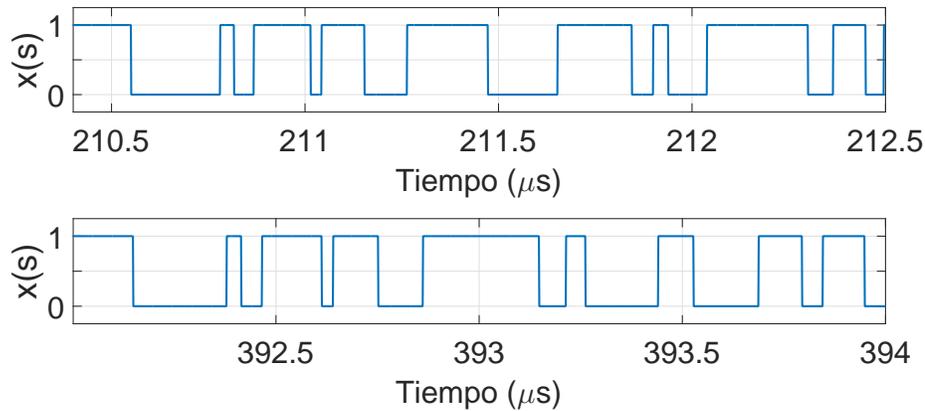


Figura 6.23: Variables Booleanas asociadas a V correspondientes al par empatado ($t_a = 2.10401 \cdot 10^{-4}$ s, $t_b = 3.92001 \cdot 10^{-4}$ s).

- Calcular $\langle \ln d(s) \rangle$, donde $\langle \rangle$ denota un promedio sobre todos los pares (t_a, t_b) que empatan.

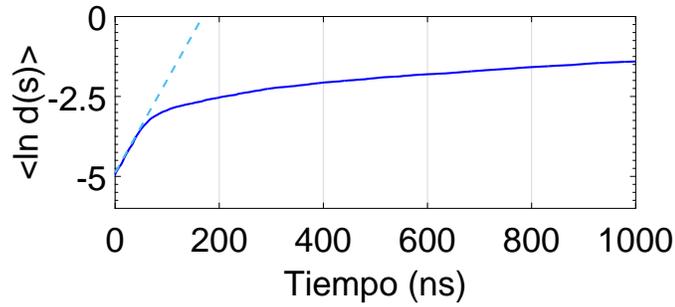


Figura 6.24: Logaritmo de la distancia Booleana como una función del tiempo.

- Estimar el exponente de Lyapunov $\lambda_{ab} = (\ln d(s) - \ln d_0)/s$.

Para este experimento, $T_0 = 200$ ns y $\delta = 0.01$. Se adquirieron series de tiempo de 0.5 ms del voltaje en la salida del nodo 1, y se encontraron ~ 3600 pares correspondientes (t_a, t_b) que satisfacen que $d_0 = d(s) < 0.01$, $\ln d_0 = -4.6$. La Figura 6.23 muestra las variables Booleanas asociadas, $x(s + t_a)$ (gráfica superior) y $x(s + t_b)$ (gráfica inferior), para los segmentos típicos de $V(s + t_a)$ y $V(s + t_b)$, cuando $t_a = 2.10401 \times 10^{-4}$ s y $t_b = 3.92001 \times 10^{-4}$ s. La Figura 6.24 muestra la evolución temporal de $\langle d(s) \rangle$. El promedio de λ_{ab} sobre todos los pares de segmentos similares es el estimado del exponente mayor de Lyapunov λ del sistema. Se encontró que $\lambda = 0.029$ ns $^{-1}$, lo cual demuestra que la red es caótica.

También se realizó experimentalmente el acoplamiento de dos redes Booleanas esclavas forzando los nodos 4 y 5 de cada red con una señal Booleana externa con una frecuencia de 700 kHz, como se muestra en la Figura 6.19. Cuando la señal de 700 kHz con un ciclo de trabajo de 90 % es aplicada a ambas redes Booleanas, existen los dos comportamientos claramente distinguibles mostrados en la Figura 6.25. El primero, un periodo sin transiciones, ocurre cuando la señal externa está en “alto”, ésta fuerza directamente a los nodos 4 y 5 a tener el mismo valor lógico, e indirectamente fuerza a los nodos restantes. El segundo, caracterizado por los disparos cortos, ocurre cuando la señal externa está en “bajo” y cada red Booleana tiene una dinámica libre dada por sus propias condiciones. Además, debe notarse que el ciclo de trabajo de la función forzadora en simulación (50 %) es diferente del ciclo de trabajo de la señal forzadora en la implementación electrónica, debido a que en simulaciones las redes Booleanas son idénticas pero experimentalmente son diferentes.

Para determinar si las dos redes Booleanas realizadas experimentalmente alcanzaban la sincronización, se introdujeron los voltajes de los nodos correspondientes a compuertas XOR auxiliares. La Figura 6.26 muestra la evolución temporal del error entre tres nodos correspondientes después de que la señal forzadora externa es aplicada, de la figura es posible ver que el error prácticamente tiende a cero lo que indica que la sincronización completa es alcanzada luego de un tiempo transitorio.

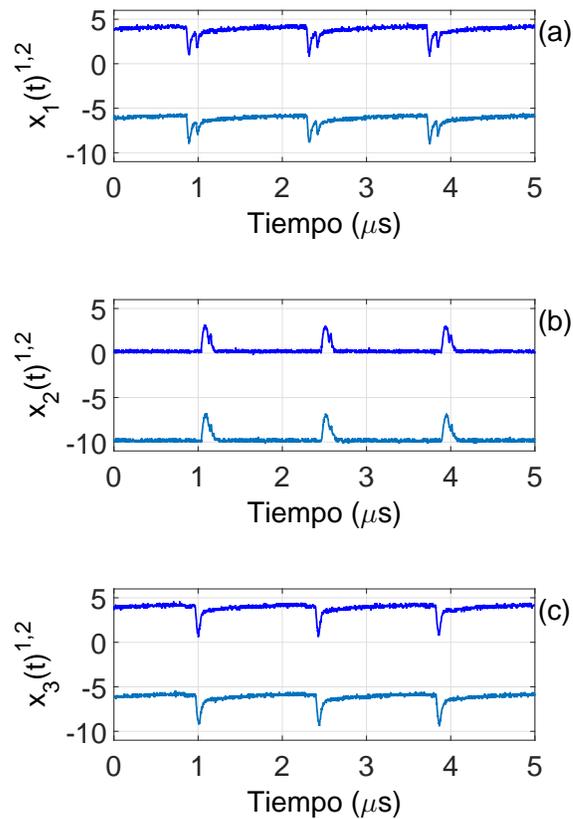


Figura 6.25: Evolución temporal de los voltajes de dos ABNs forzadas y realizadas experimentalmente. (a) Medidas experimentales de las salidas de los nodos 1. (b) Medidas experimentales de las salidas de los nodos 2. (c) Medidas experimentales de las salidas de los nodos 3. En los tres casos la señal superior corresponde a la red Booleana 1, mientras que la señal inferior corresponde a la red Booleana 2 y esta señal presenta un *offset* de -10 V.

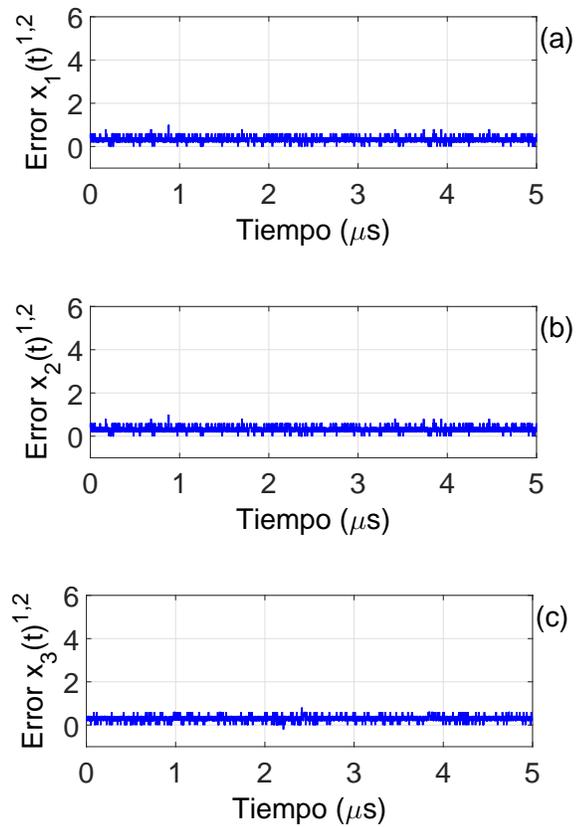


Figura 6.26: Evolución temporal de los errores entre nodos correspondientes de dos ABNs realizadas experimentalmente. (a) Error entre nodos 1. (b) Error entre nodos 2. (c) Error entre nodos 3.

Conclusiones y trabajo a futuro

7.1. Conclusiones

Antes de que comenzar a dar algunos comentarios concluyentes, es importante tener presente que:

- Esta tesis integra dos aspectos importantes. Por un lado, la investigación científica como generadora de conocimiento y por otro, la innovación tecnológica a partir del conocimiento.
- El objetivo de esta tesis consistió en sincronizar un par de redes Booleanas autónomas bajo diferentes esquemas de acoplamiento, tanto numéricamente como experimentalmente. La validación de los resultados numéricos se planeó hacerla a través de un dispositivo electrónico reconfigurable llamado arreglo de compuertas dinámicas programables en campo, para ello era necesario diseñar e implementar dicho dispositivo.

Con base en lo anterior, es posible dar las siguientes conclusiones:

- La hipótesis de la cual se partió pudo ser validada debido a que efectivamente, a partir de un sistema no lineal fue posible la obtención de compuertas lógicas capaces de cambiar su configuración. Dichas compuertas en conjunto con una red de interconexión basada en lógica combinacional fueron aplicadas en la construcción de un arreglo de compuertas dinámicas programables en campo.
- En esta investigación fue desarrollada la teoría matemática que respalda el funcionamiento de las compuertas lógicas dinámicas. El diseño de estas compuertas está basado en un sistema no lineal. Con respecto a otros enfoques que han utilizado sistemas caóticos para la obtención de compuertas configurables, el esquema propuesto en esta tesis tiene la ventaja de no ser sensible a condiciones iniciales ni a ruido.

- El arreglo de compuertas programables en campo es un dispositivo reconfigurable diseñado para que tanto la funcionalidad de cada una de las compuertas lógicas dinámicas como la interconexión entre ellas puedan ser programadas por el usuario. La programación de la funcionalidad de las compuertas lógicas dinámicas representa una ventaja al momento de realizar funciones lógicas complejas debido a que el número de compuertas necesario para implementarlas disminuye. Sin embargo, el número de compuertas en el interior del prototipo implementado es hasta el momento muy limitado, de tal manera que es prácticamente imposible compararse en ese aspecto con los dispositivos comerciales actuales, específicamente con los FPGAs.
- Para el modelado de las redes Booleanas autónomas se utilizaron ecuaciones Booleanas con retardo. Éstas resultan ser una buena aproximación para describir de manera cualitativa lo que está sucediendo en una red Booleana. Mediante simulaciones numéricas, la teoría de BDEs fue comprobada, particularmente aquella relacionada con los retardos. Es decir, se logró corroborar que si la red tiene solamente retardos racionales, entonces la red tiene un comportamiento periódico, mientras que si se tienen solamente retardos irracionales, entonces la red se comporta de manera compleja, esto es debido a que los retardos irracionales afectan directamente en número de transiciones de estado por unidad de tiempo. Sin embargo, con respecto a esto último, es preciso enfatizar que en las simulaciones numéricas hechas en computadora es prácticamente imposible implementar un retardo irracional, por lo que a lo sumo se tiene una representación aproximada del comportamiento.
- En cuanto a la sincronización de dos redes Booleanas, se analizaron los tres esquemas de acoplamiento comúnmente utilizados: acoplamiento unidireccional, acoplamiento bidireccional y el forzamiento de sistemas por señal externa. Para cada uno de ellos se presentó la forma de interconectar dos redes Booleanas preservando la esencia del acoplamiento. La topología de las redes Booleanas corresponde a una red de tres nodos, donde un par de ellos ejecutan la función XOR y el restante la función XNOR. Utilizando una métrica adecuada para calcular la distancia entre las trayectorias se observó que la diferencia entre los estados de la red tiende a cero, es decir, se logra la sincronización completa para los tres tipos de acoplamiento empleados.
- En el caso del acoplamiento unidireccional se determinó que dada la topología de la red a estudiar, basta con acoplar las redes en uno de sus estados para lograr la sincronización completa debido a que el grafo de la red es conexo. La señal forzadora depende de los estados de los nodos correspondientes en donde actúa el acoplamiento y sólo es transmitida de la red maestra a la red esclava.
- Para el caso del acoplamiento bidireccional, un par de señales es aplicada en ambos sentidos. Las señales de acoplamiento deben ser aplicadas en nodos diferentes, es decir, si la señal forzadora que va de la red uno hacia la red dos acopla determinado par de nodos de ambas redes, entonces la señal forzadora que va de la red dos hacia la red uno debe ser aplicada en otro par diferente de nodos. Cuando se acoplan adecuadamente dos redes Booleanas autónomas en forma bidireccional, el tiempo que tardan en sincronizar es menor que cuando son acopladas en esquema maestro-esclavo; esto

se debe al hecho de dos estados correspondientes de ambas redes alcanzan inmediatamente la sincronización luego de que las señales forzadoras son activadas quedando solamente uno de los estados de las redes por alcanzar la sincronización, mismo que alcanza el estado sincronizado luego de un breve instante.

- Finalmente, se logró la sincronización completa de dos redes Booleanas autónomas forzadas por una señal externa. Para lograr sincronizar las dos redes se hizo un pequeño cambio en la topología de la red Booleana tomada como base. Este cambio consistió en agregar a cada red a sincronizar un par de nodos, los cuales ejecutan la operación lógica OR. La señal externa aplicada a una de las entradas de los nodos OR determina los periodos de tiempo en los cuales las órbitas están oscilando o convergiendo a un punto fijo. Cuando la señal externa es un cero lógico, las redes se mantienen oscilando; mientras que cada vez que la señal externa es un uno lógico, las trayectorias son llevadas a una condición muy cercana, de tal modo que eventualmente se alcanzará la sincronización. La teoría matemática fue corroborada tanto con simulaciones numéricas como con la realización experimental en el FPDGA. Además, para determinar el grado de complejidad que tiene la nueva topología de red propuesta, se calculó el exponente de Lyapunov de las señales obtenidas experimentalmente, dando como resultado un exponente positivo $\lambda = 0.029 \text{ ns}^{-1}$.

7.2. Trabajo a futuro

Derivadas de este trabajo de tesis, a continuación se mencionan algunas de las directrices de trabajo a futuro que podrían ser realizables:

- Actualmente, el FPDGA tiene las siguientes limitantes:
 - La primera de ellas está relacionada con las dimensiones que ocupa. Por el hecho de estar implementado con componentes comerciales, su tamaño es muy grande. En el futuro, esta desventaja puede ser corregida por dos métodos diferentes. El primer método consiste en rediseñar el FPDGA empleando ahora dispositivos de montaje superficial. El segundo método implica hacer el diseño del FPDGA a nivel transistor, de tal manera que éste pueda ser llevado a una escala mínima y ser integrado en un chip de silicio.
 - La segunda limitante es el número de compuertas lógicas dinámicas en el interior del FPDGA. Este número es muy pequeño comparado con el número de bloques lógicos que contiene un FPGA. Para solucionar esta limitante es necesario incluir una mayor cantidad de compuertas dinámicas en el dispositivo, lo cual implica proponer una nueva metodología para interconectar eficientemente dichas compuertas. Este desafío sugiere el uso de teoría de grafos y optimización para mejorar la interconexión programable.
 - Las compuertas lógicas dinámicas en el FPDGA hoy en día son capaces de realizar cinco funciones lógicas de manera directa con los parámetros de diseño que

se establecieron en esta tesis. La idea es proponer un nuevo conjunto de parámetros con los cuales se pueden emular mayor cantidad de funciones lógicas, o en su defecto, proponer un sistema que permita obtener una mayor funcionalidad.

- Con respecto a la sincronización de dos redes Booleanas autónomas en un futuro sería importante saber bajo qué condiciones se pueden lograr otros tipos de sincronización, no sólo la completa, es decir, buscar la manera de lograr sincronización en fase, anti-fase, con retardo, etc.
- Por el comportamiento complejo que presentan las redes Booleanas autónomas con retardos inconmensurables, éstas pueden ser utilizadas en la generación de secuencias pseudo-aleatorias. La idea es realizar las pruebas estadísticas a dichas secuencias con la finalidad de garantizar que éstas pueden ser utilizadas en encriptación de información o en sistemas de comunicación.

Bibliografía

- [1] S. H. Strogatz, “Exploring complex networks,” *Nature*, vol. 410, no. 8625, pp. 268–276, 2001.
- [2] S. A. Kauffman, “Metabolic stability and epigenesis in randomly constructed genetic nets,” *Journal of theoretical biology*, vol. 22, no. 3, pp. 437–467, 1969.
- [3] D. Cheng, H. Qi, and Z. Li, *Analysis and Control of Boolean Networks: A Semi-tensor Product Approach*. Communications and Control Engineering, Springer London, 2010.
- [4] R. Zhang, H. L. D. S. Cavalcante, Z. Gao, D. J. Gauthier, J. E. S. Socolar, M. M. Adams, and D. P. Lathrop, “Boolean chaos,” *Phys. Rev. E*, vol. 80, p. 045202, 2009.
- [5] K. Klemm and S. Bornholdt, “Stable and unstable attractors in Boolean networks,” *Physical review. E, Statistical, nonlinear, and soft matter physics*, vol. 72, p. 055101, 2005.
- [6] D. Cheng and H. Qi, “Controllability and observability of Boolean control networks,” *Automatica*, vol. 45, no. 7, pp. 1659–1667, 2009.
- [7] C. J. Kuhlman, H. S. Mortveit, D. Murrugarra, and V. S. A. Kumar, “Bifurcations in Boolean networks,” *ArXiv e-prints*, 2011.
- [8] D. Cheng, H. Qi, Z. Li, and J. B. Liu, “Stability and stabilization of Boolean networks,” *International Journal of Robust and Nonlinear Control*, vol. 21, no. 2, pp. 134–156, 2011.
- [9] L. Glass, T. J. Perkins, J. Mason, H. T. Siegelmann, and R. Edwards, “Chaotic dynamics in an electronic model of a genetic network,” *Journal of Statistical Physics*, vol. 121, pp. 969–994, 2005.
- [10] J. Norrell, B. Samuelsson, and J. E. S. Socolar, “Attractors in continuous and Boolean networks,” *Phys. Rev. E*, vol. 76, p. 046122, 2007.

- [11] D. P. Rosin, D. Rontani, and D. J. Gauthier, “Ultrafast physical generation of random numbers using hybrid Boolean networks,” *Physical Review E*, vol. 87, p. 040902, 2013.
- [12] D. P. Rosin, D. Rontani, D. J. Gauthier, and E. Schöll, “Control of synchronization patterns in neural-like Boolean networks,” *Physical Review Letters*, vol. 110, no. 10, p. 104102, 2013.
- [13] J. Whitesitt, *Boolean algebra and its applications*. Addison-Wesley series in the engineering sciences, Addison-Wesley Pub. Co., 1961.
- [14] S. Givant and P. Halmos, *Introduction to Boolean algebras*. Undergraduate texts in Mathematics, Springer-Verlag New York, 1 ed., 2009.
- [15] J. Gregg, *Ones and zeros: understanding Boolean algebra, digital circuits, and the logic of sets*. IEEE Press understanding science & technology series, IEEE Press, 1998.
- [16] M. Mano, *Lógica digital y diseño de computadores*. Prentice-Hall Hispanoamericana, 1982.
- [17] S. Shiva and R. De la Parra, *Introducción al diseño lógico: circuitos digitales*. Trillas, 1998.
- [18] C. C. Walker and W. R. Ashby, “On temporal characteristics of behavior in certain complex systems,” *Kybernetik*, vol. 3, no. 2, pp. 100–108, 1966.
- [19] D. Dee and M. Ghil, “Boolean difference equations, I: Formulation and dynamic behavior,” *SIAM Journal on Applied Mathematics*, vol. 44, no. 1, pp. 111–126, 1984.
- [20] M. Ghil and A. Mullhaupt, “Boolean delay equations. II. Periodic and aperiodic solutions,” *Journal of Statistical Physics*, vol. 41, no. 1, pp. 125–173, 1985.
- [21] L. Glass and C. Hill, “Ordered and disordered dynamics in random networks,” *EPL (Europhysics Letters)*, vol. 41, no. 6, pp. 599–604, 1998.
- [22] M. Ghil, A. Mullhaupt, and P. Pestiaux, “Deep water formation and quaternary glaciations,” *Climate Dynamics*, vol. 2, no. 1, pp. 1–10, 1987.
- [23] D. G. Wright, T. F. Stocker, and L. A. Mysak, “A note on quaternary climate modelling using Boolean delay equations,” *Climate Dynamics*, vol. 4, no. 4, pp. 263–267, 1990.
- [24] I. Zaliapin, V. Keilis-Borok, and M. Ghil, “A Boolean delay equation model of colliding cascades. Part I: Multiple seismic regimes,” *Journal of Statistical Physics*, vol. 111, no. 3, pp. 815–837, 2003.
- [25] I. Zaliapin, V. Keilis-Borok, and M. Ghil, “A Boolean delay equation model of colliding cascades. Part II: Prediction of critical transitions,” *Journal of Statistical Physics*, vol. 111, no. 3, pp. 839–861, 2003.
- [26] M. Ghil, I. Zaliapin, and B. Coluzzi, “Boolean delay equations: A simple way of looking at complex systems,” *Physica D Nonlinear Phenomena*, vol. 237, pp. 2967–2986, 2008.

- [27] B. Coluzzi, M. Ghil, S. Hallegatte, and G. Weisbuch, “Boolean delay equations on networks: An application to economic damage propagation,” Papers 1003.0793, arXiv.org, 2010.
- [28] B. Coluzzi, M. Ghil, S. Hallegatte, and G. Weisbuch, “Boolean delay equations on networks in economics and the geosciences,” *International Journal of Bifurcation and Chaos*, vol. 21, p. 3511, 2011.
- [29] M. Zanin and A. Pisarchik, “Boolean networks for cryptography and secure communication,” *Nonlinear Sci. Lett. B.*, vol. 1, no. 1, pp. 25–32, 2011.
- [30] S. Squires, A. Pomerance, M. Girvan, and E. Ott, “Stability of boolean networks: The joint effects of topology and update rules,” *Physical review. E, Statistical, nonlinear, and soft matter physics*, vol. 90, p. 022814, 2013.
- [31] T. J. Todman, G. A. Constantinides, S. J. E. Wilton, and O. Mencer, “Reconfigurable computing: Architectures and design methods,” *IEE Proc. Comput. Digit. Tech.*, vol. 152, no. 2, pp. 193–207, 2005.
- [32] M. Platzner, “Reconfigurable computer architectures,” *e&i Elektrotechnik und Informationstechnik*, vol. 115, no. 3, pp. 143–148, 1998.
- [33] P. C. Diniz, E. Marques, K. Bertels, M. M. Fernandes, and J. M. P. Cardoso, *Reconfigurable Computing: Architectures, Tools and Applications. Third International Workshop, ARC 2007, Mangaratiba, Brazil, March 27-29, 2007, Proceedings*, vol. 4419 of *Theoretical Computer Science and General Issues*. Springer-Verlag Berlin Heidelberg, 2007.
- [34] C. Bobda, *Introduction to Reconfigurable Computing. Architectures, Algorithms, and Applications*. Springer Netherlands, 1 ed., 2007.
- [35] S. Hauck and A. DeHon, *Reconfigurable Computing: The Theory and Practice of FPGA-Based Computation*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2007.
- [36] J. Cardoso and M. Hübner, *Reconfigurable Computing: From FPGAs to Hardware/Software Codesign*. Springer Publishing Company Incorporated, 2014.
- [37] G. Estrin and C. R. Viswanathan, “Organization of a “fixed-plus-variable” structure computer for computation of eigenvalues and eigenvectors of real symmetric matrices,” *Journal of the ACM (JACM)*, vol. 9, pp. 41–60, 1962.
- [38] F. J. Rammig, “A concept for the editing of hardware resulting in an automatic hardware-editor,” in *Proceedings of the 14th Design Automation Conference, DAC ‘77*, (Piscataway, NJ, USA), pp. 187–193, IEEE Press, 1977.
- [39] P. Bertin, D. Roncin, and J. Vuillemin, “Introduction to programmable active memories,” in *Systolic Array Processors* (J. McCanny, J. McWhirter, and E. Swartzlander Jr., eds.), pp. 301–309, Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1989.

- [40] H. Parvez and H. Mehrez, *Application-Specific Mesh-based Heterogeneous FPGA Architectures*. Circuits and systems, Springer-Verlag New York, 1 ed., 2011.
- [41] L. Li, C. Yang, S. Hui, W. Yu, J. Kurths, H. Peng, and Y. Yang, “A reconfigurable logic cell based on a simple dynamical system,” *Mathematical Problems in Engineering*, vol. 2013, no. 735189, 2013.
- [42] V. Agrawal, S. S. Kang, and S. Sinha, “Realization of morphing logic gates in a repressilator with quorum sensing feedback,” *Physics Letters A*, vol. 378, no. 16, pp. 1099–1103, 2014.
- [43] S. Sinha and W. L. Ditto, “Dynamics based computation,” *Physical Review Letters*, vol. 81, pp. 2156–2159, 1998.
- [44] S. Sinha and W. L. Ditto, “Computing with distributed chaos,” *Physical review. E, Statistical physics, plasmas, fluids, and related interdisciplinary topics*, vol. 60, pp. 363–377, 1999.
- [45] T. Munakata, S. Sinha, and W. L. Ditto, “Chaos computing: Implementation of fundamental logical gates by chaotic elements,” *Circuits and Systems I: Fundamental Theory and Applications, IEEE Transactions on*, vol. 49, pp. 1629–1633, 2002.
- [46] K. Murali, S. Sinha, and W. L. Ditto, “Implementation of NOR gate by a chaotic Chua’s circuit,” *I. J. Bifurcation and Chaos*, vol. 13, pp. 2669–2672, 2003.
- [47] K. Murali, S. Sinha, and W. L. Ditto, “Construction of a reconfigurable dynamic logic cell,” *Pramana*, vol. 64, pp. 433–441, 2005.
- [48] H. Peng, Y. Yang, L. Li, and H. Luo, “Harnessing piecewise-linear systems to construct dynamic logic architecture,” *Chaos*, vol. 18, no. 3, p. 033101, 2008.
- [49] K. Murali, A. Miliotis, W. L. Ditto, and S. Sinha, “Logic from nonlinear dynamical evolution,” *Physics Letters A*, vol. 373, pp. 1346–1351, 2009.
- [50] D. N. Guerra, A. R. Bulsara, W. L. Ditto, S. Sinha, K. Murali, and P. Mohanty, “A noise-assisted reprogrammable nanomechanical logic gate,” *Nano Letters*, vol. 10, no. 4, pp. 1168–1171, 2010.
- [51] H. Peng, F. Liu, L. Li, Y. Yang, and X. Wang, “Dynamic logic architecture based on piecewise-linear systems,” *Physics Letters A*, vol. 374, no. 13, pp. 1450–1456, 2010.
- [52] H. Peng, G. Hu, L. Li, Y. Yang, and J. Xiao, “Constructing dynamic multiple-input multiple-output logic gates,” *Mathematical Problems in Engineering*, vol. 2011, no. 380345, 2011.
- [53] X. Yuan and W. Liu, “Implementation and improvement of dynamic logic gates based on cellular neural networks,” in *2012 Fifth International Workshop on Chaos-fractals Theories and Applications*, pp. 99–103, 2012.

- [54] A. Rothenbuhler, T. Tran, E. H. B. Smith, V. Saxena, and K. A. Campbell, “Reconfigurable threshold logic gates using memristive devices,” *Journal of Low Power Electronics and Applications*, vol. 3, no. 2, pp. 174–193, 2013.
- [55] A. Pikovsky, M. Rosenblum, and J. Kurths, *Synchronization: A Universal Concept in Nonlinear Sciences*. Cambridge Nonlinear Science Series, Cambridge University Press, 2003.
- [56] L. D. Iasemidis and J. C. Sackellares, “Review : Chaos theory and epilepsy,” *The Neuroscientist*, vol. 2, no. 2, pp. 118–126, 1996.
- [57] S. Strogatz, *Sync: How Order Emerges from Chaos in the Universe, Nature, and Daily Life*. Hachette Books, 2003.
- [58] A. C. J. Luo, *Dynamical system synchronization*. Nonlinear systems and complexity, Springer, 2013.
- [59] C. Huygens, *Horologium oscillatorium: sive, De motu pendulorum ad horologia aptato demonstrationes geometricae*. F. Muguet, 1673.
- [60] H. Fujisaka and T. Yamada, “Stability theory of synchronized motion in coupled-oscillator systems,” *Progress of Theoretical Physics*, vol. 69, no. 1, pp. 32–47, 1983.
- [61] E. Mosekilde, Y. Maistrenko, and D. Postnov, *Chaotic synchronization: Applications to living systems*. Series on Nonlinear Science, World Scientific, 2002.
- [62] L. M. Pecora and T. L. Carroll, “Synchronization in chaotic systems,” *Phys. Rev. Lett.*, vol. 64, no. 8, pp. 821–824, 1990.
- [63] A. Balanov, N. Janson, D. Postnov, and O. Sosnovtseva, *Synchronization. From simple to complex*. No. 1 in Springer Series in Synergetics, Springer-Verlag Berlin Heidelberg, 2009.
- [64] X. B. Lu and B. Z. Qin, *Synchronization in Complex Networks*. Computer networks series, Nova Science Publishers, 2011.
- [65] S. Boccaletti, J. Kurths, G. Osipov, D. Valladares, and C. Zhou, “The synchronization of chaotic systems,” *Physics Reports*, vol. 366, no. 1, pp. 1–101, 2002.
- [66] V. Anishchenko, A. Balanov, N. Janson, N. Igosheva, and G. V. Bordyugov, “Synchronization of cardiorythm by weak external forcing,” *Discrete Dynamics in Nature and Society*, vol. 4, pp. 201–206, 1999.
- [67] M. G. Rosenblum, A. S. Pikovsky, and J. Kurths, “Phase synchronization of chaotic oscillators,” *Phys. Rev. Lett.*, vol. 76, no. 11, pp. 1804–1807, 1996.
- [68] A. Pikovsky, M. Rosenblum, and J. Kurths, “Phase synchronization in regular and chaotic systems,” *International Journal of Bifurcation and Chaos*, vol. 10, no. 10, pp. 2291–2305, 2000.

- [69] U. E. Vincent, A. N. Njah, O. Akinlade, and A. R. T. Solarin, “Phase synchronization in bi-directionally coupled chaotic ratchets,” *Physica A: Statistical Mechanics and its Applications*, vol. 360, no. 2, pp. 186–196, 2006.
- [70] G. Erjaee, “On analytical justification of phase synchronization in different chaotic systems,” *Chaos, Solitons and Fractals*, vol. 39, no. 3, pp. 1195–1202, 2009.
- [71] A. L. Fradkov and B. Andrievsky, “Synchronization and phase relations in the motion of two-pendulum system,” *International Journal of Non-Linear Mechanics*, vol. 42, no. 6, pp. 895–901, 2007.
- [72] M. G. Rosenblum, A. Pikovsky, and J. Kurths, “From phase to lag synchronization in coupled chaotic oscillators,” *Phys. Rev. Lett.*, vol. 78, pp. 4193–4196, 1997.
- [73] E. M. Shahverdiev, S. Sivaprakasam, and K. A. Shore, “Lag synchronization in time-delayed systems,” *Physics Letters A*, vol. 292, no. 6, pp. 320–324, 2002.
- [74] C. Li, X. Liao, and K. wo Wong, “Chaotic lag synchronization of coupled time-delayed systems and its applications in secure communication,” *Physica D: Nonlinear Phenomena*, vol. 194, no. 3, pp. 187–202, 2004.
- [75] Z.-M. Ge, Y.-T. Wong, and S.-Y. Li, “Temporary lag and anticipated synchronization and anti-synchronization of uncoupled time-delayed chaotic systems,” *Journal of Sound and Vibration*, vol. 318, no. 1, pp. 267–278, 2008.
- [76] F. Zhang, “Lag synchronization of complex Lorenz system with applications to communication,” *Entropy*, vol. 17, pp. 4974–4985, 2015.
- [77] E. Campos, J. Urias, and N. F. Rulkov, “Multimodal synchronization of chaos,” *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 14, no. 1, pp. 48–54, 2004.
- [78] N. F. Rulkov, M. M. Sushchik, L. S. Tsimring, and H. D. I. Abarbanel, “Generalized synchronization of chaos in directionally coupled chaotic systems,” *Physical Review E*, vol. 51, no. 2, pp. 980–994, 1995.
- [79] H. D. I. Abarbanel, N. F. Rulkov, and M. M. Sushchik, “Generalized synchronization of chaos: The auxiliary system approach,” *Physical Review E*, vol. 53, no. 5, pp. 4528–4535, 1996.
- [80] L. Kocarev and U. Parlitz, “Generalized synchronization, predictability, and equivalence of unidirectionally coupled dynamical systems,” *Physical review letters*, vol. 76, no. 11, pp. 1816–1819, 1996.
- [81] N. F. Rulkov, V. S. Afraimovich, C. T. Lewis, J.-R. Chazottes, and A. Cordonet, “Multivalued mappings in generalized chaos synchronization,” *Physical Review E*, vol. 64, no. 1, p. 016217, 2001.
- [82] R. R. Rivera-Durón, E. Campos-Cantón, I. Campos-Cantón, and D. J. Gauthier, “Forced synchronization of autonomous dynamical Boolean networks,” *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 25, no. 8, p. 083113, 2015.

Apéndice

APÉNDICE A

Productividad

En este apéndice se muestran los productos obtenidos a partir del presente trabajo de tesis doctoral.

■ Carteles

- R. R. Rivera-Durón y E. Campos-Cantón, “Redes Booleanas con retardo implementadas con arquitectura flexible,” *LVII Congreso Nacional de la Sociedad Mexicana de Física*, 2014.

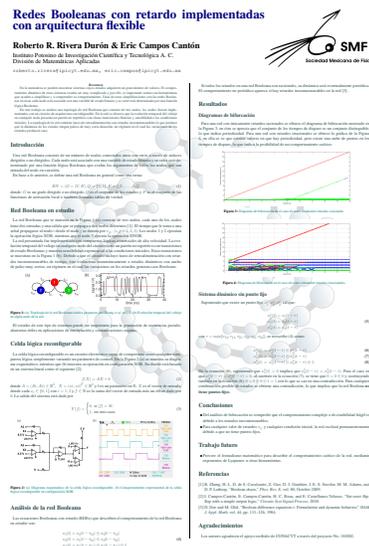


Figura A.1: Póster para el LVII Congreso Nacional de la Sociedad Mexicana de Física.

- R. R. Rivera-Durón y E. Campos-Cantón, “Sincronización forzada de redes Booleanas autónomas,” *XVII Feria de Posgrados de Calidad CONACYT*, 2016.

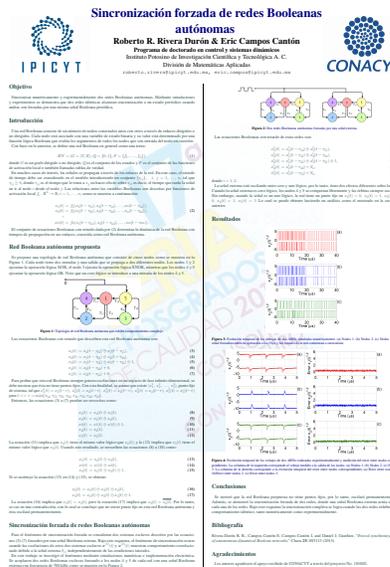


Figura A.2: Póster para la 17a. Feria de Posgrados de Calidad.

■ Exposiciones

- R. R. Rivera-Durón y E. Campos-Cantón, “Diferentes comportamientos en redes Booleanas autónomas,” *XLVIII Congreso Nacional de la Sociedad Matemática Mexicana*, 2015.



Figura A.3: Constancia de participación en el XLVIII Congreso Nacional de la Sociedad Matemática Mexicana.

■ Artículos en revistas indizadas en el JCR (*Journal Citation Report*)

- R. R. Rivera-Durón, E. Campos-Cantón, I. Campos-Cantón, and D. J. Gauthier, “Forced synchronization of autonomous dynamical Boolean networks,” *Chaos:*

An Interdisciplinary Journal of Nonlinear Science, vol. 25, no. 8, p. 083113, 2015.



Figura A.4: Artículo de investigación publicado.

■ Propiedad intelectual y transferencia de tecnología

- Solicitud de patente con folio MX/E/2016/060667 ante el Instituto Mexicano de la Propiedad Industrial. R. R. Rivera-Durón, E. Campos-Cantón y M. García-Martínez, “Método y circuito para integrar una matriz programable en campo de compuertas lógicas reconfigurables que emplean un sistema no lineal y un recableado programable eficaz”, 2016.
- Patente US 9,793,897. R. R. Rivera-Durón, E. Campos-Cantón, and M. García-Martínez, “Method and circuit for integrating a programmable matrix in the field of reconfigurable logic gates employing a non-linear system and an efficient programmable rewiring”, 2017.

