



UNIVERSIDAD AUTÓNOMA DE SAN LUIS POTOSÍ

FACULTAD DE CIENCIAS

POSGRADO EN CIENCIAS APLICADAS

Instituto de Investigación en Comunicación Óptica

**“Diseño de un Circuito Electrónico para una
Red No Lineal Celular Reconfigurable”**

Tesis para obtener el grado de:
Maestro en Ciencias Aplicadas

Presenta:

I.E. Baltazar Cerda Cerda

Asesor:

Dr. Isaac Campos Cantón

San Luis Potosí, S.L.P., México

Septiembre de 2018



TESIS DE MAESTRÍA

NOMBRE DE LA TESIS

NOMBRE DEL ALUMNO

COMITÉ QUE ACEPTA LA TESIS

Sinodal

M.I. _____

Sinodal

Dra. _____

Sinodal

Dr. _____

Asesor

Dr. _____

Septiembre de 2018

Dedicatoria

A mi hijo Leonardo Baltazar

A mi esposa Gladis Jessahi

A mis padres Ma. Jesús y Baltazar

Agradecimientos

A Leonardo, por ser mi motor que me inspira a seguir día con día, gracias a ti he tenido la fortaleza de retomar este camino, te amo hijo.

A Jessahi, por estar siempre a mi lado en todo momento y no dejarme caer, solo tú sabes cuánto nos ha costado llegar hasta aquí, siempre juntos en las buenas y en las malas, te amo.

A mi madre por todo el cariño y apoyo que me has brindado en el caminar de esta vida, y sobre todo los consejos y el impulso que me has brindado para que el día de hoy este cumpliendo este cometido, te amo mamá.

A mi padre, por el cariño, la sabiduría, los consejos y por nunca dejarme caer, gracias por siempre predicar con el ejemplo, te amo papá.

A mis hermanos Cinthia y Jonathan, con los que me ha tocado compartir esta gran aventura llamada vida, y que a pesar de la distancia siempre están presentes en mi mente, de alguna u otra forma son parte de esto, los amo.

A mis abuelas, abuelos[†] y familia en general, por el cariño y apoyo que siempre me brindan.

A mi asesor el Dr. Isaac Campos por la paciencia y el apoyo recibido a través de los años, le agradezco infinitamente el poder llegar hasta aquí.

A mi comité evaluador, M.I. Grisela Rodríguez, Dra. Marcela Mejía y Dr. Oscar Núñez por todo el apoyo, tiempo y disposición, muchas gracias.

A mis profesores del IICO, por brindarme su conocimiento y apoyo durante estos

Agradecimientos

dos años, a todos y cada uno de ustedes, muchas gracias.

Al Dr. Raúl Balderas por su grande apoyo fuera del aula como coordinador del posgrado y aun después de terminar su gestión, muchas gracias.

A los doctores Manuel Camacho y Jorge Ortega por su gran apoyo como subcoordinadores del posgrado.

A la L.A. María Inés Torre García por su infinito apoyo en cada uno de los tramites a realizar y sobre todo en el de la titulación, muchas gracias.

Muy especialmente quiero agradecer mis compañeros y amigos de cubículo Lorena, Juan y Alberto con quienes comencé esta aventura y de los que siempre recibí un gran apoyo, de verdad ¡mil gracias!

A la Lic. Karla Marely Ortega González, por todo su apoyo y facilidades al buscar información sobre alguna bibliografía en especial, pero sobre todo por tu amistad, muchas gracias Karla.

A todos y cada uno de mis compañeros del IICO con los que he logrado formar una muy buena amistad y pasar tan buenos momentos, no los menciono a todos por que afortunadamente son muchos y temo en que se me pase alguno, sin embargo, ustedes sabes quienes son.

A los hermanos David, Miguel y Alex Domínguez, por su sincero apoyo antes y durante mi estancia en el posgrado, pero sobre todo su gran amistad que sigue creciendo al pasar de los años, de verdad les agradezco, y pues seguimos adelante carnales.

A Ursula Jiménez, Fabiola Vázquez y Javier Cruz por su gran apoyo en esa difícil etapa justo antes de ingresar, y por su amistad brindada a través de los años.

A Sergio Méndez por la confianza transmitida durante nuestra estancia en la academia, muchas gracias mi Serch.

A todos y cada uno de mis amigos, que como mencione anteriormente soy afortunado de no poder contarlos solo con los dedos de las manos, siempre de alguna u otra forma he recibido su apoyo, muchas gracias a todos.

A Aria Infuriata por ser una gran parte de mi vida, y por todas las satisfacciones que me ha y me sigue dando.

Al rock 'n' roll por llenar de alegrías mi alma y corazón.

Y al universo por permitirme estar aquí en este momento y por todas las alegrías y experiencias que día a día me brinda la oportunidad de vivir... infinitas gracias.

“Gracias Totales”
- Gustavo Cerati

Resumen

En el presente trabajo se realizó el análisis y diseño electrónico de una Red No lineal Celular Reconfigurable. Esto con el objetivo de tener un sistema físico que emule el comportamiento de un Autómata Celular en una forma básica (unidimensional con un vecindario de tres elementos); que sea capaz de presentar dinámica en cada una de las células que lo conforman. La Red No lineal Celular Reconfigurable es un sistema que genera una respuesta evolutiva de 5 elementos durante un determinado número de iteraciones. Esta red se construyó a partir de circuitos reconfigurables y un arreglo de flip-flops, el cual funciona como etapa de memoria para llevar a cabo la retroalimentación que genera el comportamiento evolutivo del Autómata Celular; donde la semilla del sistema se define mediante la configuración de “preset - reset” de los flip-flops. Se desarrolló de forma general la teoría propuesta para diseñar una compuerta lógica dinámica, lo cual sienta las bases para obtener el modelo matemático del sistema gracias a la similitud del operador módulo 2 (compuerta lógica *XOR*) y la *regla 90* de los Autómatas Celulares Unidimensionales; este dispositivo es un circuito reconfigurable que constituye el corazón de nuestro sistema, ya que cada una de las células de la red esta compuesta por uno de estos circuitos sintonizados en una función “*f*” determinada. Debido al comportamiento dinámico que presentan este tipo de circuitos, y al ser sintonizados todos con una misma función, se dice que el sistema obtenido es una *Red No lineal Celular*. Se realizaron simulaciones de un Autómata Celular con compuertas lógicas *XOR*, de la compuerta reconfigurable y del arreglo de la Red No Lineal Celular en el software *SPICE NI Multisim*. También se programó una simulación para evolucionar y observar el comportamiento de las distintas reglas de Autómatas Celulares con semillas definidas y aleatorias en *MATLAB*. Finalmente se obtuvieron los resultados del diseño electrónico, donde se pudo observar la evolución del sistema por medio de un arreglo de LEDs, de forma favorable, en comparación con las evoluciones obtenidas mediante simulaciones numéricas de las reglas 10, 90, 110, 126 y 254.

Índice general

Resumen	VII
Introducción	1
1. Marco Teórico	9
1.1. Sistemas Dinámicos	9
1.2. Autómatas Celulares	11
1.2.1. Unidimensionales	12
1.2.2. Bidimensionales	15
1.3. Redes No lineales Celulares	18
2. Sistema con célula de dos entradas	25
2.1. Compuerta Lógica Reconfigurable con dos entradas	26
2.1.1. Análisis matemático	27
2.1.2. Circuito electrónico	30
2.1.3. Resultados	34
2.2. Sistema empleando la compuerta reconfigurable	37
2.2.1. Circuito electrónico	39
2.2.2. Resultados	41
3. Sistema con célula de tres entradas	43
3.1. Célula con tres entradas	44
3.1.1. Análisis matemático	45
3.1.2. Circuito electrónico	47
3.1.3. Resultados	49
3.2. Diseño de la CNN	56
3.2.1. Circuito electrónico	57
3.2.2. Resultados	60

Índice general

Conclusiones	63
Bibliografía	XI
Anexo	XV
Índice de figuras	XIX
Índice de tablas	XXI

Introducción

Antecedentes y Estado del Arte

La necesidad de entender sistemas cuyo comportamiento es complejo, y hasta cierto punto impredecible, como pueden ser el cambio climático, el control de tráfico, la propagación de virus y enfermedades, la evolución de patologías dentro del cuerpo, e incluso la actividad cerebral, etc., nos ha llevado a crear modelos e idealizaciones que nos permitan emular dichos sistemas; para así poder entender, y en algunos casos manipular, la evolución que estos presentan al paso del tiempo.

Algunos de estos son los Autómatas Celulares y las Redes Neuronales. Estas últimas son, a grandes rangos, modelos computacionales compuestos por interconexiones entre elementos llamados neuronas artificiales, las cuales se organizan dentro de una estructura multicapa llamada red. La primera de estas capas contiene la información de entrada y la última muestra la respuesta del sistema. En medio de estas existe un arreglo con una o más “capas ocultas” compuestas por neuronas artificiales que llevan a cabo el procesamiento mediante una determinada *función de activación*. Estas neuronas tienen la capacidad de aprender y cambiar el peso de su *función de activación*, que por lo general es un número decimal entre 0 y 1 en intervalos decimales, y de esta forma se trata de simular la interacción entre las neuronas que genera el funcionamiento del cerebro. Los primeros en mostrar un modelo neuronal artificial fueron McCulloch y Pitts en 1943. En su ensayo “*A Logical Calculus of the Ideas Immanent in Nervous Activity*”, demostraron que en una red finita de neuronas, siendo estas la unidad básica del cerebro, se podía implementar un programa de la máquina de Turing [1, 2].

Por otra parte, la teoría de autómatas habla del estudio de dispositivos de cálculo abstractos [3]. McIntosh nos dice que si pensamos en un autómata como cualquier dispositivo capaz de realizar acciones complejas, entonces su historia es tan antigua que nos remonta hasta la revolución industrial, época en la que Babbage tuvo la inquietud

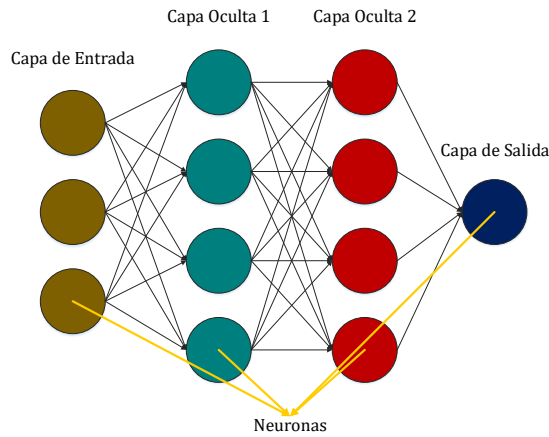


Figura I.1: Red Neuronal con dos capas ocultas

por generar una máquina que pudiera ser totalmente autónoma, pero debido a las limitaciones tecnológicas de la época esto no pudo convertirse en realidad [1].

Uno de los más grandes aportes a la teoría de autómatas lo hizo Alan Turing, quien durante los años 30 comenzó a trabajar en una máquina la cual tenía todas las capacidades que se pueden encontrar en un ordenador actual, en lo que a capacidad de cálculo se refiere. Esta máquina constaba de una cinta infinita dividida en casillas que almacenaban símbolos, por lo general 1's y 0's, un cabezal móvil para leer esos símbolos y un programa el cual daba las instrucciones al cabezal según la operación que se fuera a realizar; después Turing se dio cuenta que el programa se podía ir escribiendo en la misma cinta que los datos de entrada, lo que dio pauta a un dispositivo más general el cual llamamos “Máquina universal de Turing”. En las computadoras actuales los componentes equivalentes son la memoria, el procesador y los programas respectivamente. El objetivo principal de Alan Turing era demostrar el límite de lo que una máquina de cálculo podía y que no podía realizar [3]. Gracias a sus contribuciones Turing es considerado el padre de la Ciencias Computacionales y la Criptografía.

Los Autómatas Celulares (**C**ellular **A**utomata, **CA** por sus siglas en inglés) comenta Ilachinski que provienen de los primeros estudios de análisis matemático y teoría de sistemas realizados por investigadores rusos entre los años 30's y 40's [2]. Sin embargo el primero en desarrollar estos modelos matemáticos bajo el nombre de “Cellular Spaces” fue John Von Neumann a finales de los años 40's. Von Neumann pretendía crear fábricas auto-reproductivas, pero debido a las deficiencias mecánicas de la épo-

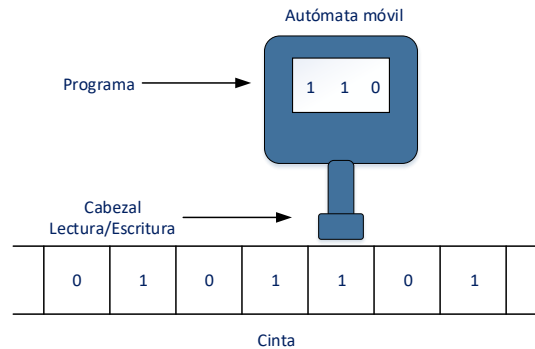


Figura I.2: Elementos de una máquina universal de Turing

ca tuvo muchos problemas; sin embargo su objetivo principal era diseñar dispositivos informáticos que emularan el comportamiento del cerebro humano donde la memoria y la unidad de procesamiento no trabajen por separado, si no que lo hicieran en paralelo y fueran capaces de reconstruirse al existir la materia prima necesaria [15, 16]. Para materializar estas ideas, su colega Stanislaw Ulam le propuso diseñar un modelo matemático abstracto, el cual sería más susceptible a demostración. Este modelo planteaba un universo discreto construido sobre dos mallas bidimensionales de máquinas de estado finitas llamadas celdas las cuales están interconectadas localmente entre ellas y pueden cambiar su estado dependiendo de los estados de las células vecinas, cada célula podía tomar 1 de 29 estados posibles. Este universo es lo que ahora conocemos como Autómatas Celulares [1].

La historia de los CA se puede resumir en tres acontecimientos principales :

- Los autómatas auto-reproductivos de Von Neumann
- El *juego de la vida* de Jhon Conway
- La clasificación de Stephen Wolfram

En 1970 John Conway presenta un juego llamado "*Life*", con el interés de encontrar una configuración más simple a los trabajos realizados por Von Neumann. Este juego se popularizó gracias a que Martin Gardner lo publica en su columna "Mathematical Games", de la revista "Scientific American", la cual contaba con muchos lectores en esa

época. En 1976 cuando se comienzan a popularizar las micro computadoras, la compañía Cromenco desarrolla una interfaz llamada “Dazzler” que cuenta con un controlador de vídeo a color, y en la cual se podían observar las evoluciones de life. Más allá de ofrecer un nuevo autómeta al repertorio, la aportación de Conway fue mostrar el potencial de los comportamientos que estos pueden llegar a tener [1].

En los años 80’s Stephen Wolfram comienza a realizar estudios y formular reglas que más adelante serían de gran interés para los científicos inmersos en esta área de investigación; esto se ve plasmado en una serie de artículos publicados del ’82 al ’88 [2].

A pesar de que Conway y Von Neumann sabían que existían reglas alternativas de evolución, decidieron concentrarse en una que sirviera a su propósito y explorar sus consecuencias a detalle [1]. Es aquí donde entra el más básico pero a su vez relevante aporte de Wolfram, ya que es el primero en organizar a los Autómatas Celulares Unidimensionales (**1D-CA**) de manera empírica en cuatro clases basándose en observar el comportamiento y patrones visuales que estos generan al evolucionar usando distintas reglas, incluso llegando a formar fractales. Un ejemplo claro de esto lo encontramos en el triangulo de Pascal, el cual se forma al evolucionar un CA usando la regla 90.

Los estudios de Wolfram se basaron en encontrar propiedades de los autómatas unidimensionales por medio de herramientas computacionales, guiado por conceptos de dinámica no lineal y mecánica estadística. Esto se ve reflejado en su artículo de 1983 “Statistical Mechanics of Cellular Automata”, donde en primera instancia nos adentra al mundo de estos modelos matemáticos y las múltiples aplicaciones que estos tienen en las diferentes ramas de la ciencia. En éste mismo artículo, nos hace mención del fenómeno de auto-organización que se presenta en los CA al utilizar semillas iniciales con valores aleatorios generados mediante funciones de probabilidad. En general Wolfram nos acerca a la representación matemática, así como a las propiedades locales y globales que presentan los Autómatas Celulares Unidimensionales [10]. Todo el trabajo acerca de Autómatas Celulares realizado por Wolfram quedo documentado en su obra maestra, el Atlas del 2002 “*A New Kind of Science*” [5].

Los Autómatas Celulares y las Rede Neuronales históricamente se han venido desarrollado de forma paralela y aunque aparentemente no tienen una conexión muy evidente, con el paso del tiempo se ha comprobado que pertenecen a la misma corriente del conocimiento [1]. A finales de la década de los 80’s el científico Leon O. Chua desarrollo una nueva clase de sistemas para procesamiento de información en tiempo real

basado en una Red Neuronal, el cual cuenta con las capacidades de procesamiento en paralelo de un Autónoma Celular, al que llamo Red Neuronal Celular (**C**ellular **N**eural **N**etwork, **CNN** por sus siglas en inglés). Chua nos dice que este modelo combina lo mejor de ambos mundos, ya que, gracias a que trabaja en tiempo continuo se puede hacer procesamiento de señales en tiempo real, y por otra parte, la interconexión local entre sus elementos lo hace un perfecto candidato para implementar en integración a muy grande escala (**V**ery **L**arge **S**cale **I**ntegration), en otras palabras implementar en un circuito integrado [11, 12].

Las CNN se pueden nombrar de distintas formas dependiendo del contexto en que éstas sean aplicadas, siendo los más comunes :

- Redes **Neuronales** Celulares \longrightarrow Ciencias del cerebro.
- Redes **No lineales** Celulares \longrightarrow Sistemas dinámicos acoplados.

En el año 2002, Chua publica una serie de artículos titulada “*A Nonlinear Dynamics Perspective of Wolfram’s New Kind of Science*”, con la intención de proveer un enfoque de dinámica no lineal a todo el trabajo realizado por Wolfram, el cual menciona que esta basado en observaciones empíricas de simulaciones por computadora. En el primero de estos artículos llamado “Threshold of Complexity” (umbral de complejidad) introduce el concepto de “índice complejo”, el cual caracteriza a una estructura tridimensional llamada “cubo booleano” sobre la que se definen geoméricamente todas las reglas de los Autómatas Celulares Unidimensionales. Chua enfatiza en la estrecha relación que existe entre los CA y las CNN, llamando a los primeros un caso especial de los segundos. La contribución más importante de este artículo es que se llega a un teorema donde se afirma que cada una de las 256 reglas propuestas por Wolfram para los 1D-CA, puede ser obtenida por una ecuación diferencial no lineal.

Una observación muy interesante que hace Chua en éste artículo, es que debemos estar conscientes de que aunque los Autómatas Celulares teóricamente evolucionan en tiempo discreto, los ordenadores donde estos se simulan son sistemas de tiempo continuo ya que están compuestos por transistores, los cuales trabajan junto a un sistema de ecuaciones que rige el comportamiento del circuito interno del ordenador, y que por lo tanto a pesar de que la velocidad de procesamiento de este último sea muy pequeña nunca existe un tiempo cero [13],

Luigi Fortuna es un científico Italiano que ha seguido muy de cerca el trabajo de Chua, incluso en el año de 2009 publicó un libro titulado “Chua’s Circuit Implementa-

tions: Yesterday, Today and Tomorrow” donde describe la importancia de este circuito el cual menciona es el mas sencillo en el que se puede demostrar “caos”, así como sus distintas implementaciones y aplicaciones; dedicando el tercer capítulo a la implementación del circuito basado en CNN’s. Fortuna menciona que la mayoría de los circuitos presentados en su libro pueden ser implementados en el mas sencillo laboratorio de electrónica o incluso en cualquier hogar y pueden ser probados con instrumentos de bajo costo o con algún dispositivo periférico para computadora [6]. En 2010 con la colaboración de Chua publican un artículo donde basándose en el primer tomo de “A Nonlinear Dynamics Perspective of Wolfram’s New Kind of Science”, y con ayuda de un microcontrolador (en el caso concreto un Arduino Nano) y una matriz de LED’s, implementan un Autómata Celular de Hardware Programable, el cual es capaz de emular el comportamiento de todas las reglas de los Autómatas Celulares Elementales sin necesidad de usar un ordenador, pudiendo observar sus evoluciones en la matriz de LED’s [15]. A este sistema lo llamaron “*The Wolfram Machine*” o *La Maquina de Wolfram*, y es de gran importancia ya que fue el primer Autómata Celular implementado de forma física, documentado hasta el momento.

También en el año 2010, como parte de su investigación en sistemas no lineales y caos, el Dr. Isaac Campos Cantón de la Universidad Autónoma de San Luis Potosí, junto a sus colaboradores comienza a publicar resultados acerca de un dispositivo que tiene la capacidad de cambiar su respuesta de acuerdo a sus parámetros de control. El sistema denominado “*Compuerta Lógica Reconfigurable*”, esta formado por un circuito electrónico con un amplificador operacional y un comparador, y como su nombre lo indica emula el comportamiento de estos circuitos lógicos. En el caso de los artículos presentados se hace el análisis para que el sistema presente la respuesta de las compuertas AND, OR y NOT, sin embargo se menciona que lograda la construcción de estas, podemos generar funciones lógicas más complejas y así estar en la posibilidad de generar una nueva generación de arquitecturas computacionales, lo que se conoce como cómputo caótico debido a que hacen uso de componentes no lineales [14, 24].

Propuesta y objetivo

El objetivo de este trabajo es realizar el diseño de un circuito reconfigurable, compuesto por componentes electrónicos básicos, el cual sea capaz de emular el comportamiento de las reglas definidas por Wolfram de los Autómatas Celulares.

El corazón del sistema esta compuesto por Compuertas Lógicas Reconfigurables,

ya que con estas podremos sintonizar en tiempo real las funciones que determinan las distintas reglas de evolución. Al tener comportamiento dinámico no lineal en dichas compuertas, y siendo estas los elementos que fungen como las células del sistema, de acuerdo a lo establecido por Chua podemos decir que al estar todas sintonizadas bajo la misma regla, estamos hablando de una Red No lineal Celular.

Se pretende que éste diseño al ser implementado físicamente sea el primero que por su naturaleza análoga trabaje completamente en tiempo real y sea un parte aguas para aplicaciones futuras que así lo requieran. El trabajo se centra en el concepto de un sistema dinámico aplicado a una **Red No lineal Celular** que presente comportamiento reconfigurable. A continuación se describen las contribuciones por capítulo:

Capítulo 1. Marco Teórico : Se describe la teoría de los elementos involucrados en el sistema: Sistemas dinámicos, Autómatas Celulares y Redes No lineales Celulares.

Capítulo 2. Sistema con célula de dos entradas : Se presenta el análisis matemático y el diseño electrónico de una compuerta reconfigurable con dos entradas, la cual fungirá más adelante como la célula del sistema. Posteriormente se hace el diseño del circuito de un Autómata Celular tomando como regla de evolución la número 90, usando primero compuertas lógicas *XOR*, para posteriormente cambiar estas por *Compuertas Lógicas Reconfigurables*.

Se presenta el análisis matemático del sistema, se diseña el circuito donde las células del mismo contengan dos entradas y se presentan los resultados obtenidos.

Capítulo 3. Sistema con célula de tres entradas : Se presenta el análisis matemático del sistema, se diseña el circuito donde las células del mismo contengan tres entradas y se presentan los resultados obtenidos.

En las [conclusiones](#) se discuten los resultados finales, dificultades y trabajo a futuro.

Capítulo 1

Marco Teórico

Este capítulo abarca las bases teóricas de los diferentes modelos involucrados en el sistema propuesto, Autómatas Celulares y Redes No lineales Celulares; así como la estrecha relación que existe entre ellos. Es importante mencionar que el sistema cuenta con la capacidad de cambiar su comportamiento al modificar sus parámetros de entrada, y así lograr sintonizar un total de 256 funciones por circuito; este comportamiento es muy parecido al funcionamiento de circuitos reconfigurables como los PLD's y FPGA's, solo que con un diseño electrónico que involucre solamente componentes básicos, esto con el fin de tener una mayor robustez y estabilidad en futuras implementaciones. También es importante recalcar que aunque se puede establecer una función distinta para cada célula, es necesario que todas y cada una de estas sean sintonizadas bajo una misma función, para que así se cumpla con el criterio de una CNN. Ambos modelos pertenecen a una clase de sistemas llamados “*sistemas dinámicos*”.

1.1. Sistemas Dinámicos

La dinámica es una rama de la física que estudia los cambios que presentan ciertos fenómenos naturales al paso del tiempo, ya sea que lleguen al equilibrio, se repita de forma cíclica o comiencen a presentar comportamientos complejos [7]. Así, los sistemas dinámicos son aquellos que representa los cambios en dichos fenómenos durante cierto periodo de tiempo, ya sea de forma continua o discreta. Estos evolucionan en base a estados, y los cambios están regidos por una ley de control con la que podemos conocer cualquier estado en algún tiempo anterior [25]. Es importante mencionar que independientemente de como se comporte el sistema, la dinámica del mismo es la que se utiliza para analizar su comportamiento. El concepto de sistema dinámico, a pesar

de ser un tema con gran impacto en la actualidad, se remonta al siglo XVII cuando Isaac Newton introduce las ecuaciones diferenciales y sus leyes del movimiento y gravitación universal, las cuales sirven de herramienta para llegar a demostraciones de las leyes del movimiento planetario enunciadas por Kepler casi 80 años antes. Para el siglo XIX Poincaré comienza con lo que sería la teoría moderna de sistema dinámicos. El pensamiento de Poincaré se basa en resolver cuestiones con una perspectiva más cualitativa, distinto a lo que se planteaba en la época, por ejemplo, pretendía llegar a tener conocimiento acerca de la existencia de estabilidad en la evolución del sistema solar, o si por el contrario algunos planetas eventualmente se perderían en el infinito. La contribución de Poincaré a la teoría de sistemas dinámicos fue desarrollar un modelo geométrico lo suficientemente robusto para resolver este tipo de problemas; comenta Michael Bick que la evolución de los estados del sistema en el modelo de Poincaré se conoce como órbita debido a su analogía con el estudio de la mecánica celeste [8, 9].

Dependiendo de como es tratada la variable de tiempo, los sistemas dinámicos se dividen en dos tipos:

- **Discretos.** Se describen mediante ecuaciones en diferencias, y como su nombre lo indica, evolucionan en pasos de tiempo discreto. En electrónica, estos se diseñan mediante circuito digitales, este es el caso de los Autómatas Celulares.

$$x_{t+1} = f(x_t), \quad t \in \mathbb{Z}, \mathbb{N} \quad (1.1)$$

- **Continuos** - Su evolución esta descrita por una ecuación diferencial, y como su nombre lo dice la indica, esta los describe en tiempo continuo. De forma electrónica estos se diseñan con circuitos análogos, como es el caso de las Redes No lineales Celulares.

$$\frac{dx}{dt} = \dot{x} = f(x), \quad t \in \mathbb{R} \quad (1.2)$$

Donde x es el estado del sistema y se representa en su espacio fase [7, 9].

Es importante mencionar que al conjunto de valores numéricos sobre los que el sistema tiende a evolucionar se le llama “*atractor*”, y que dichos atractores geométricamente pueden ser puntos fijos, órbitas periódicas, toroides o estructuras de tipo fractal llamadas “atractores extraños”.

1.2. Autómatas Celulares

Los autómatas celulares son modelos matemáticos que evolucionan temporal y espacialmente en forma discreta siguiendo un conjunto de reglas específicas. Estos representan sistemas dinámicos, lo que permite estudiar el comportamiento de fenómenos físicos complejos. Así, se pueden describir como un conjunto de elementos, que se actualizan por sí mismos en paralelo mediante una regla de evolución [25] y que poseen una fuerte interacción local entre ellos.

La evolución de los Autómatas Celulares se genera sobre espacios rectangulares bidimensionales llamados mallas o enrejados, donde están distribuidos n número de elementos llamados células. Dependiendo del número de dimensiones del CA, se pueden ocupar desde 1 hasta n número de enrejados; por lo general los 1D y 2D-CA evolucionan sobre un solo enrejado, la diferencia está en la forma como lo hacen, ya que los 1D-CA evolucionan continuamente sobre un solo eje partiendo de un estado inicial definido en la parte superior y generando las nuevas generaciones hacia la parte inferior del enrejado; mientras que los 2D-CA lo hacen iniciando desde cualquier parte del enrejado, extendiéndose en todas direcciones hacia cualquier otra parte del mismo. Para los 3D-CA se usan dos enrejados que se cruzan por el centro, y así sucesivamente dependiendo del orden de dimensiones donde evolucione el sistema.

Por medio del siguiente polinomio característico podemos representar la configuración completa de un Autómata Celular con N número de celdas en un tiempo t :

$$A^t(x) = \sum_{i=0}^{N-1} a_i(t)x^i \quad (1.3)$$

donde el coeficiente $a_i(t)$ representa el valor del sitio i .

Las células o sitios, pueden tomar dos o más valores que se actualizan simultáneamente basándose en los valores de sus vecinos en pasos de tiempo anterior de acuerdo a las reglas locales de evolución que sean especificadas. Los 1D-CA son considerados como computadoras de procesamiento en paralelo, y su nombre se debe a su similitud con el crecimiento celular, donde las células crecen, se reproducen y mueren con el paso del tiempo [26]. Tienen un sin número de aplicaciones en ramas como la física, biología, teoría de números, etc., ya que muestran muchos de los comportamientos complejos de un sistema continuo

1.2.1. Unidimensionales

Son los CA más estudiados y con más aplicaciones hasta el momento; como se mencionó con anterioridad, existe un extenso trabajo realizado por Stephen Wolfram acerca de este modelo. Son Autómatas Celulares que evolucionan en una dimensión, esto quiere decir que lo hacen linealmente sobre uno de los ejes pertenecientes al enrejado. La forma para llevar a cabo esta evolución es colocando la semilla inicial en la parte superior del enrejado, sobre el eje de las x , se aplica la regla a cada una de las células que componen la semilla y de esta manera se forma un nuevo vector, el cual se coloca una posición por debajo de la semilla anterior sobre el eje y , y así sucesivamente. Como podemos observar en la figura 1.3, el eje y representa el tiempo y crece en forma descendiente sobre el enrejado.

Los 1D-CA por lo general evolucionan por medio de un vecindario de radio 1, esto quiere decir que el valor de un sitio es determinado por la regla aplicada a él mismo y sus vecinos contiguos más cercanos en un tiempo anterior $(-1, 0, 1)$ como lo podemos ver en la figura 1.1; si el sitio se encuentra ya sea al principio o al final del enrejado, se toma como vecino la célula al otro extremo del mismo como si se tratara de un anillo. Cuando además de esto, las células solo pueden tomar dos estados 0, 1, se le conoce con el nombre de Autómata Celular Elemental.

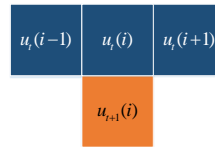


Figura 1.1: Vecindario de un Autómata Celular Elemental

En el contexto de los Autómatas Celulares Fundamentales, existen $2^3 = 8$ posibles patrones, en otras palabras, los ocho números binarios de tres bits, que pueden cambiar el estado de la célula, por lo tanto, Wolfram definió un total de $2^8 = 256$ reglas de evolución diferentes para este tipo de modelos.

Wolfram agrupo estas 256 reglas en 4 clases según su comportamiento, como lo podemos observar en la figura 1.2:

- **Clase 1: Fijas** - Casi todas las condiciones iniciales conducen hacia el mismo estado final, esto quiere decir que después de cierto periodo de transición, los

valores de las células no cambiaran.

- **Clase 2: Periódicas** - Existen muchos estados finales distintos, pero todos ellos consisten solo de ciertas estructuras sencillas, que ya sea que se mantengan iguales por siempre o bien se repitan cada cierto tiempo.
- **Clase 3: Caóticas** - El comportamiento de estos modelos es más complejo ya que parece tener comportamientos aleatorios. Si el estado inicial no tiene un orden, la sucesión de valores para ciertos sitios puede darse de manera caótica; sin embargo, figuras como triángulos y otras estructuras de pequeña escala siempre podrán observarse.
- **Clase 4: Complejas** - Involucra una combinación de orden y aleatoriedad, durante su evolución pueden observarse estructuras relativamente sencillas, pero que la interacción entre ellas se da bajo una variedad de comportamientos complejos.

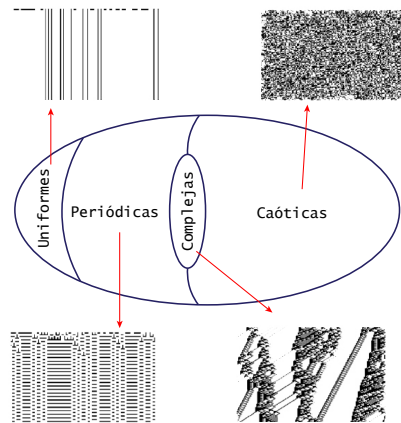


Figura 1.2: Clasificación de Wolfram de los Autómatas Celulares

Wolfram nos menciona que existen dos restricciones que, aunque no se consideran como esenciales, son aplicadas al conjunto de las 256 reglas de los 1D-CA,

1. Una regla puede ser considerada ilegal a menos que un estado inicial “nulo” o “inactivo” que consista únicamente de 0 permanece sin cambios. Esto prohíbe las reglas cuya especificación binaria termina con un 1.
2. Las reglas deben ser de reflexión simétrica, de modo que 100 y 001 (y 110 y 011) arrojen valores idénticos. En pocas palabras, una regla se considera legal si es simétrica y asigna la configuración nula a sí misma.

1.2. Autómatas Celulares

Estas restricciones nos dejan con un total de 32 reglas “legales”: 0, 4, 18, 22, 32, 36, 50, 54, 72, 76, 90, 94, 104, 108, 122, 126, 128, 132, 146, 150, 160, 164, 178, 182, 200, 204, 218, 222, 232, 236, 250, 254.

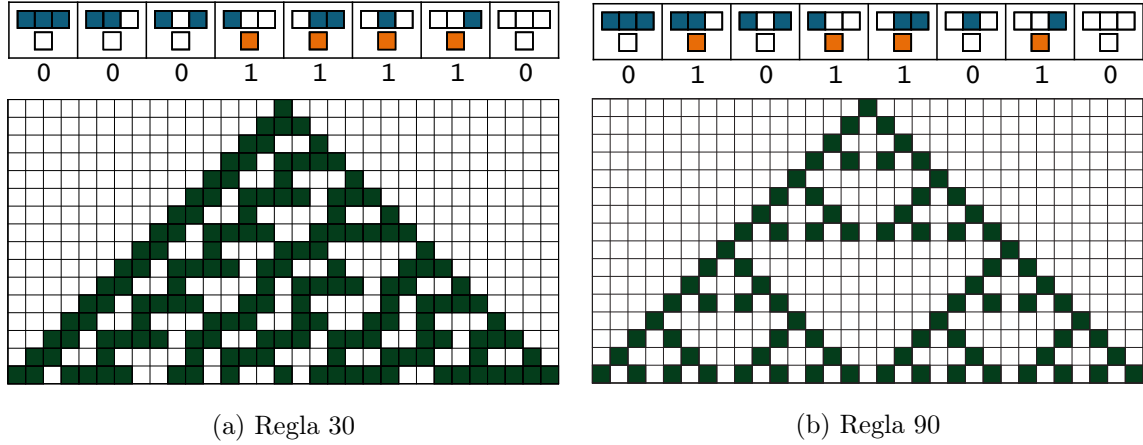


Figura 1.3: Ejemplos de Autómatas Celulares Unidimensionales: a) Regla 30 (Illegal), b) Regla 90 (legal).

Wolfram también nos menciona que entre estas reglas hay cuatro que presentan la propiedad de adición aditiva, esto quiere decir que la evolución que se genera a partir de estas reglas satisface el principio de superposición, estas reglas son la 0, 90, 150 y 204, de las cuales 0 y 204 son triviales ya que 0 borra cualquier valor de la semilla inicial, mientras que 204 lo mantiene sin cambios. La regla 90 es el operador suma módulo 2, mejor representado en lógica digital como la compuerta XOR, el cual se aplica entre los vecinos contiguos a la izquierda y a la derecha de la célula en un estado anterior.

$$f_{90} = u_t(i - 1) \oplus u_t(i + 1) \quad (1.4)$$

Con la regla 150 pasa algo similar, pero en este caso la operación se aplica tanto a los vecinos como a la célula en dicho estado.

$$f_{150} = u_t(i - 1) \oplus u_t \oplus u_t(i + 1) \quad (1.5)$$

Otra propiedad importante mencionada por Wolfram, es que algunos CA son “periféricos”, es decir que el estado de la célula depende solo del valor de sus vecinos en el paso de tiempo anterior pero no del suyo. Estas reglas son la 0, 90, 160 y 250. Nótese

que la regla 90 cumple con todas las propiedades antes mencionadas. Wolfram menciona que si observamos los patrones generados por las reglas legales, se puede observar que existen reglas donde el 1 inicial es borrado desde la primera iteración como puede ser en las reglas 0 y 60, u otras donde este 1 inicial permanece sin cambios como en la regla 4 y 30, esto es debido a que el patrón de su regla local esta dado por $100 \rightarrow 0$ o $001 \rightarrow 0$ lo cual previene la propagación del 1 inicial,

En resumen, un Autómata Celular Unidimensional se puede considerar como una secuencia de valores que evolucionan de acuerdo a reglas predeterminadas que involucran el valor de los estados de sus vecinos cercanos. La dinámica de los 1D-CA regularmente se muestra en diagramas de espacio - tiempo, figura 1.4.

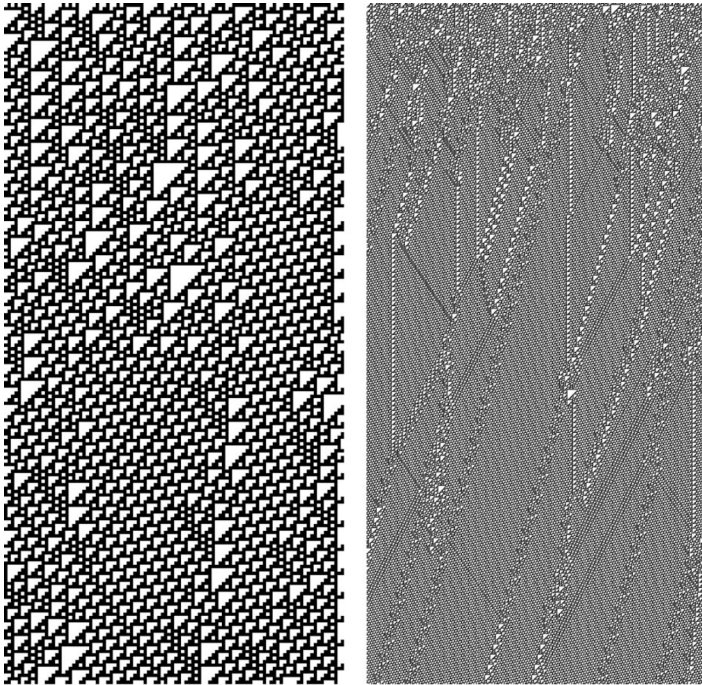


Figura 1.4: Diagramas de espacio - tiempo de la regla 110 en diferentes niveles de amplificación

Como parte del proyecto se realizó un programa en Matlab para correr [simulaciones](#) de un CA unidimensional. El código del programa se puede consultar en la sección [anexo](#).

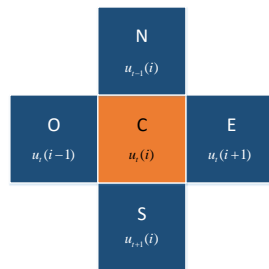
1.2.2. Bidimensionales

A pesar de que este trabajo estuvo enfocado concretamente al modelo unidimensional de Wolfram, es importante mencionar que existen estudios sobre modelos de más

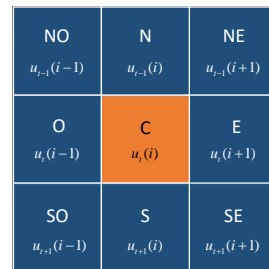
1.2. Autómatas Celulares

de una dimensión, principalmente en 2D y más reciente mente en 3D. Le evolución de estos, al igual que ocurre con los 1D-CA, se genera sobre un enrejado rectangular, pero en este caso los patrones avanzan sobre los ejes x y y obedeciendo las regla establecidas, y la semilla del sistema se puede colocar en cualquier parte del enrejado antes de la primera iteración. Al igual que los CA unidimensionales, los elementos pueden tomar distintos estados dependiendo de la regla evolutiva, recordemos que en el Autómata de Von Neumann cada célula podía tomar hasta 29 estados.

Los vecindarios más usados para la evolución de CA bidimensionales son el de Von Neumann donde los vecinos son las cuatro celdas que rodean ortogonalmente al sitio central, y la de Moore que es muy parecida a Von Neumann solo que se agregan las 4 celdas que faltan para completar el perímetro que rodea a la celda central como lo podemos observar en la figura 1.5.



(a) Vecindad de Von Neumann



(b) Vecindad de Moore

Figura 1.5: Vecindarios más usados en los Autómatas Celulares Bidimensionales

Un ejemplo de 2D-CA, pero desde otro enfoque evolutivo, lo podemos ver en el artículo “*2-Variable Boolean Operation - Its Use in Pattern Formation*” de Sahoo. En él se muestra como obtener patrones de tipo fractal utilizando operaciones booleanas de 2 variables (2-VBO), aplicando alguna de las $2^4 = 16$ reglas de evolución generadas por las posibles combinaciones binarias entre 2 variables. Un ejemplo de como realizar una 2-VBO es la siguiente:

- Aplicar la regla f_2 entre $(43)_{10} = (101011)_2$ y $(24)_{10} = (11000)_2$

siendo

$$f_2(0, 0) = 0 \quad f_2(0, 1) = 1 \quad f_2(1, 0) = 0 \quad f_2(1, 1) = 0$$

El resultado de la operación es 16 decimal, y lo podemos ver a continuación:

$$\begin{array}{rcccccc} 1 & 0 & 1 & 0 & 1 & 1 & (43)_{10} \\ 0 & 1 & 1 & 0 & 0 & 0 & (24)_{10} \\ \hline 0 & 1 & 0 & 0 & 0 & 0 & (16)_{10} \end{array} \quad (1.6)$$

Ahora, conociendo como se resuelven las 2-VBO, el procedimiento para realizar la evolución del 2D-CA es el siguiente:

1. Se construye una matriz bidimensional de M , de (i) renglones por (j) columnas.
2. Se escoge una de las 16 reglas de evolución para llevar a cabo la 2-VBO.
3. Se calcula el valor en cada elemento de la matriz como: $M_{i,j} = 2VBO(i, j)$, tomando en cuenta la regla de evolución.
4. Se colocan los resultados de las operaciones en cada célula en forma decimal, y se observan los números que se repiten formando patrones.
5. Se asignan colores a los números que forman los fractales y se colorean sus casillas.

En la figura 1.6 podemos observar la evolución de un 2D-CA, usando la regla 14 que se comporta como un operador OR, en una malla de 70×70 . Si prestamos atención en la imagen podremos observar que $y = 31$ para $x \in [32, 63]$, esto se puede apreciar mejor sobre el área del ovalo color negro [17].

Por otra parte, como se menciona en el capitulo anterior, uno de los 2D-CA más importantes es el juego de la vida desarrollado por John Conway. La evolución de este 2D-CA, como la mayoría de estos modelos, esta determinada por el estado inicial, y no requiere entradas posteriores. Cada una de las células del autómata puede tener uno de dos estados, viva o muerta, y cada una de estas interactúa con sus ocho vecinos aledaños bajo el criterio del vecindario de Moore.

Las reglas de evolución son las siguientes:

- 1) Cada célula viva con menos de dos vecinos vivos, muere por falta de población
- 2) Cada célula viva con dos o tres vecinos vivos, prevalece a la siguiente generación.
- 3) Cada célula viva con más de tres vecinos vivos, muere por sobrepoblación.

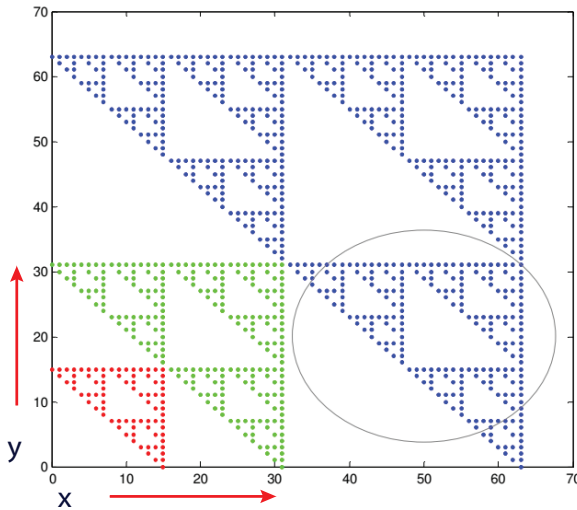


Figura 1.6: Patrones de tipo Fractal obtenidos mediante la regla 14, usando operaciones booleanas de 2 variables. Rojo = 15, Verde = 31, Azul= 63

- 4) Cada célula muerta con exactamente tres vecinos vivos, se torna en una célula viva por reproducción.

El patrón inicial constituye la semilla del sistema, y al aplicarle las reglas de evolución a cada una de sus células se van creando las generaciones posteriores. Conway reviso las reglas de una manera muy cuidadosa, para evitar los extremos en los cuales las células tienen un crecimiento sin límite o por otro lado donde las células van desapareciendo y eventualmente todas mueren [1].

Los 2D-CA han sido especialmente utilizados en el campo del procesamiento de imágenes, encriptación y reconocimiento de patrones [10].

1.3. Redes No lineales Celulares

Las Redes No lineales Celulares son modelos computacionales introducidos por Leon Chua a finales de los años 80's. Estos sistemas tienen la peculiaridad de que por una parte son circuitos analógicos no lineales que procesan señales en tiempo real, como las Redes Neuronales, y al mismo tiempo se forman sobre un enrejado compuesto por células con la misma distancia de separación, las cuales en su interior llevan clones

de un mismo circuito, que se comunican localmente entre sí mediante sus vecinos más cercanos, igual que los Autómatas Celulares, por esto es que las CNN son muy eficaces para realizar procesamiento de señales en paralelo de alta velocidad. Una CNN se define básicamente por dos conceptos matemáticos:

1. Una colección espacialmente discreta de sistemas dinámicos no lineales llamados células, donde la información puede ser encriptada en cada célula a través de tres variables independientes llamadas entrada, umbral y estado inicial.
2. Una ley de acoplamiento que relaciona una o más variables, tales como estado, salida, etc., de cada celda C_{ij} con todas las celdas vecinas ubicadas dentro de una esfera de influencia de radio r , centrada en C_{ij} .

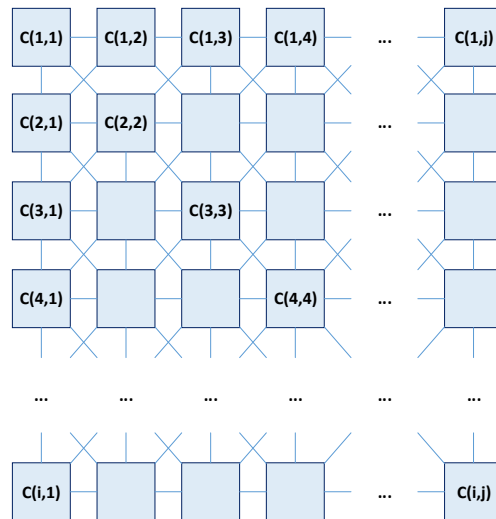


Figura 1.7: CNN bidimensional básica de $i \times j$ elementos

La estructura básica de una CNN, al igual que un CA, está formada de células adyacentes agrupadas en vecindarios; estas células son la unidad básica de una CNN y solo pueden comunicarse con sus vecinos, sin embargo, células que no tienen conexión directa pueden verse afectadas debido a los efectos de propagación de la dinámica de tiempo continuo que presenta la red. En el contexto de las CNN's, los vecindarios se definen de acuerdo al radio de células alrededor de la célula central, es decir, un vecindario de radio 1 ($r = 1$), es aquel donde se tienen una célula interna y sus 8 vecinos que lo rodean, a este tipo de vecindario por lo general se le llama de 3×3 ; así

1.3. Redes No lineales Celulares

entonces un vecindario de $r = 2$ es aquel que partiendo de la célula central se toman las 2 células hacia el exterior en las 8 direcciones al que también se le llama de 5×5 , y así sucesivamente.

El vecindario de cada célula C_{ij} de una CNN está definido por la siguiente ecuación:

$$N_r(i, j) = \{C(k, l) \mid \max\{|k - i|, |l - j|\} \leq r, 1 \leq k \leq M, 1 \leq l \leq N\} \quad (1.7)$$

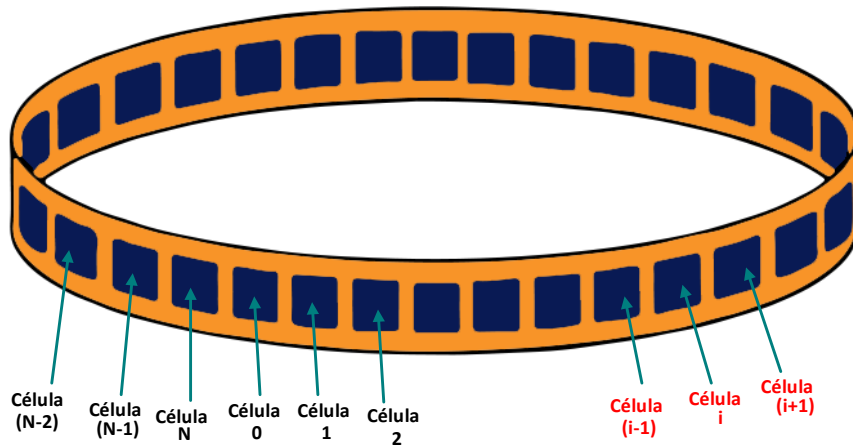
donde r es un número positivo, un ejemplo se muestra en la figura 1.8.



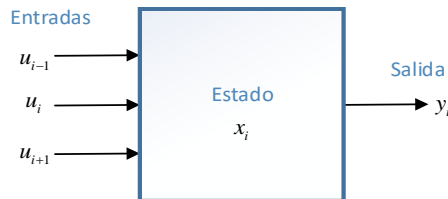
Figura 1.8: Vecindarios de 3×3 y 5×5 para cualquier célula $C(i, j)$ perteneciente a la figura 1.7, a) $r=1$ y b) $r=2$

En una CNN, todas las células internas tienen las mismas estructuras de circuito, con los mismos valores en sus elementos, ya que la red está caracterizada por el conjunto de todas las ecuaciones diferenciales no lineales de las células que conforman el circuito. Estas células que llaman internas son aquellas que tienen $2r + 1$ número de vecinos, las demás células del vecindario se les conoce como células límite. En el primer artículo donde Chua habla acerca de las CNN hace una propuesta de célula compuesta por un capacitor, dos resistencias, tres fuentes de corriente controladas por voltaje, una fuente de voltaje independiente y un elemento no lineal para el que usa una fuente de corriente controlada por voltaje lineal a trozos [13], sin embargo, el mismo define que una célula de CNN contiene en su interior cualquier sistema no lineal que su comportamiento este regido por una ecuación diferencial. Es importante mencionar que las CNN tienen retroalimentación de salida y mecanismos de control a la entrada, y al igual que en los CA teóricamente podemos tener CNN's de cualquier número de dimensiones.

En su artículo del 2002, Chua hace un análisis en el cual se demuestra la estrecha relación entre las Redes Neuronales Celulares y los Autómatas Celulares, incluso menciona que los 1D-CA son un caso especial de CNN. Para esto primeramente define un 1D-CA como un anillo formado por células contiguas, y asume que cada célula contiene en su interior un sistema dinámico como se puede observar en la figura 1.9, que evoluciona de acuerdo a un comportamiento definido y a su propia escala de tiempo; cuando las células se acoplan el sistema completo evoluciona de tal forma que obedece a su propia regla, así como a la regla de interacción impuesta por reglas de acoplamiento.



(a)



(b)

Figura 1.9: a) 1D-CA construido por $N + 1$ células idénticas que solo interactúa con sus vecinos próximos a su derecha y a la izquierda, b) cada célula “ i ” tiene una variable de estado $x_i(t)$, una de salida $y_i(t)$ y tres entradas binarias constantes u_{i-1} , u_i y u_{i+1}

Para caracterizar las reglas booleanas que surgen de los 1D-CA, Chua define un

1.3. Redes No lineales Celulares

entero, que nombra como “*índice de complejidad*” representado por la letra k , con el cual puede cuantificar el grado de no linealidad de los 1D-CA. Este sistema dinámico menciona que es equivalente a un mapa no lineal que se describe con la siguiente tabla de verdad:

Tabla 1.1: Tabla de verdad para la función $y_i = N(u_{i-1}, u, u_{i+1})$

	u_{i-1}	u_i	u_{i+1}	y_i
0	-1	-1	-1	y_0
1	-1	-1	1	y_1
2	-1	1	-1	y_2
3	-1	1	1	y_3
4	1	-1	-1	y_4
5	1	-1	1	y_5
6	1	1	-1	y_6
7	1	1	1	y_7

Como se puede observar, en la tabla de verdad los resultados en la salida corresponden a $y_N \in \{-1, 1\}$, en vez de usar $\{0, 1\}$ como se hace con las señales binarias; esto lo hacen por que usa los atractores del sistema para codificar la tabla de verdad binaria ya que el estado y la salida evolucionan en tiempo real a través de una ecuación diferencial ordinaria, la cual está diseñada para que una vez que la solución alcance un estado estable la salida tienda a $\{-1, 1\}$. Cada una de estas reglas locales puede ser representada por un cubo, donde cada uno de sus vértices representa a uno de los ocho posibles estados de las vecindades de 3 células en los 1D-CA, por lo tanto, a los vértices se les asignan dos colores para representar el valor de la salida en el “cubo booleano”.

Después de un extenso análisis realizado en [13], cada regla puede ser mapeada en un sistema dinámico no lineal definido por la siguiente ecuación de estados:

$$\dot{x} = (-x_i + (|x_i + 1| - |x_i - 1|)) + \{z2 + c2|(z1 + c1|z0 + b_1u_{i-1} + b_2u_i + b_3u_i)|\}$$

$$i = 0, 1, \dots, L$$

$$x_i(0) = 0$$
(1.8)

donde $b_1, b_2, b_3, c_1, c_2, z_0, z_1$ y z_2 son números reales que caracterizan las reglas de evolución, u_i representa el estado binario del 1D-CA, y x_i es el estado del sistema

equivalente de tiempo continuo del 1D-CA. Con estos parámetros, el sistema converge a un punto de equilibrio $x_i(Q)$ para $i = 0, 1, \dots, L$, y la salida cuidadosamente definida como :

$$y_i(t) = 0,5(|x_i(t) + 1| - |x_i(t) - 1|) \quad (1.9)$$

converge a un valor booleano dado por la siguiente expresión:

$$y_i(Q) = \text{sgn}\{z_2 + c_2|(z_1 + c_1|z_0 + b_1u_{i-1} + b_2u_i + b_3u_{i+1}|)|\} \quad (1.10)$$

Así, la evolución del estado binario en tiempo discreto de las células esta dada por la siguiente relación:

$$u_i^{t+1} = \text{sgn}\{z_2 + c_2|(z_1 + c_1|z_0 + b_1u_{i-1}^t + b_2u_i^t + b_3u_{i+1}^t)|\} \quad (1.11)$$

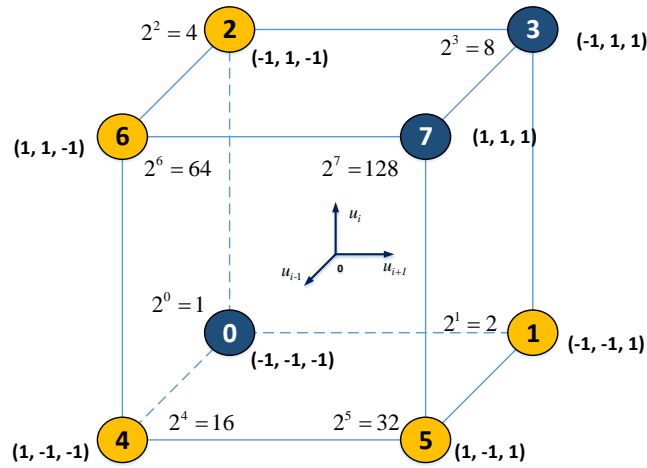


Figura 1.10: Cubo Booleano de la regla 110, azul= 0 y amarillo=1

Capítulo 2

Sistema con célula de dos entradas

En este capítulo se aborda primeramente el diseño de una Compuerta Lógica Reconfigurable (CLR) de dos entrada que funciona como célula para el sistema. Posteriormente se realiza el diseño del circuito de la red empleando compuertas reconfigurables con el propósito de obtener capacidad de reconfiguración y así poder sintonizar el circuito con cualquiera de las 16 funciones posibles con dos entradas. Finalmente se agrega elementos de memoria que permiten guardar el estado de un tiempo anterior y así poder generar la evolución del Autómata, cabe mencionar que la semilla del sistema se define dentro de este arreglo de *Flip-Flops* y no en las entradas de las compuertas, lo que permite que una vez arrancado el sistema no haya necesidad de manipularlo manualmente.

Como se mencionó en el capítulo anterior, el vecindario básico de un 1D-CA consta de tres elementos, sin embargo sabemos que la evolución de la regla 90 solo necesita del estado de sus vecinos y no hace uso de su propio estado para poder evolucionar, de manera similar podemos observar esta propiedad en las reglas 160 y 250, que comportan como un operador AND y OR de dos entradas respectivamente, se podrá observar que la regla 160 al ser del tipo $100 \rightarrow 0$ previene la propagación del 1 en la semilla y por lo tanto no existe evolución. Por esta razón y aprovechando esta propiedad es que se implemento el Autómata Celular con elementos de dos entradas y de lo cual a continuación mostramos los resultados obtenidos. Posteriormente en el siguiente capítulo se realiza el diseño para el vecindario de tres elementos.

2.1. Compuerta Lógica Reconfigurable con dos entradas

Las compuertas lógicas son bloques que realizan la función de operadores booleanos, por lo tanto operan con números binarios y solamente responden a niveles bajos y altos de voltaje, 0 y 5 volts respectivamente, los cuales son representados con valores lógicos 0 y 1.

Para que la red planteada pueda tener un comportamiento reconfigurable, es necesario contemplar un elemento dinámico el cual al variar sus parámetros sea capaz de modificar su respuesta en tiempo real, para este propósito se plantea usar una compuerta lógica reconfigurable como célula para el sistema. Este dispositivo es capaz de cambiar su respuesta al modificar sus parámetros de control, con lo cual es posible generar compuertas lógicas básicas y funciones booleanas según sus combinaciones de entrada. Es preciso mencionar que generar este tipo de dispositivos dinámicos existe un extenso trabajo realizado por Campos-Cantón para obtener distintas funciones en un espacio de estados bidimensionales, del cual se pueden consultar algunos resultados en [14]. Por simplicidad para diseño del circuito electrónico en este trabajo se diseña mediante un enfoque de funciones lineales sobre un arreglo matricial.

A continuación se hace el análisis matemático de la compuerta lógica reconfigurable con dos entradas y se muestran los resultados al obtener las compuertas OR, AND y XOR cuyas respuestas teóricas se observan en la siguiente tabla de verdad:

Tabla 2.1: Tabla de verdad de las funciones que representan a las compuertas lógicas básicas

Entradas		Función		
x_1	x_2	$f_6(XOR)$	$f_8(AND)$	$f_{14}(OR)$
0	0	0	0	0
0	1	1	0	1
1	0	1	0	1
1	1	0	1	1

2.1.1. Análisis matemático

Para la construcción de las compuertas hacemos uso de funciones lineales en un arreglo matricial y un vector de entradas binarias, así consideramos el siguiente sistema lineal:

$$\dot{X} = AX + B \quad (2.1)$$

Donde $A \in \mathbb{R}^2 \times \mathbb{R}^2$ y $X, B \in \mathbb{R}^2$ son matrices reales; A y B contienen los parámetros del sistema, X es el vector de entradas binarias ($x_i \in \{0, 1\}$) y $Y \in \mathbb{R}^2$ es la salida ($y_i \in \{0, 1\}$) definida por:

$$Y(f) = \begin{cases} 1, & |f + C| < k \\ 0, & \text{de otra forma} \end{cases} \quad (2.2)$$

siendo $C, k \in \mathbb{R}^n$ los valores del centro y el radio de una circunferencia respectivamente, donde el valor de la salida esta determinado dependiendo si el valor de la función cae dentro o fuera de esta circunferencia, a la que llamamos umbral.

Así, si tenemos que:

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}, \quad B = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}, \quad X = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad \text{y} \quad Y = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \quad (2.3)$$

al sustituir los valores de 2.3 en 2.1 , el sistema queda descrito de la siguiente forma:

$$\begin{bmatrix} f_1 \\ f_2 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \quad (2.4)$$

y lo podemos expresar en forma de polinomios:

$$\begin{aligned} f_1 &= a_{11}x_1 + a_{12}x_2 + b_1 \\ f_2 &= a_{21}x_1 + a_{22}x_2 + b_2 \end{aligned} \quad (2.5)$$

Entonces, de los polinomios obtenidos podemos decir que la salida 2.2 queda expresada de la forma:

$$Y_1(f_1) = \begin{cases} 1, & |f_1 + C_1| < k_1 \\ 0, & \text{de otra forma} \end{cases} \quad (2.6)$$

$$Y_2(f_2) = \begin{cases} 1, & |f_2 + C_2| < k_2 \\ 0, & \text{de otra forma} \end{cases}$$

Si observamos bien el sistema, nos podemos percatar que mediante el arreglo matricial básicamente estamos consiguiendo obtener dos salidas iguales pero que dependen de funciones distintas, y al ser nosotros los que definimos el valor que debe tener la salida de cada función, estamos en la posibilidad de generalizar el sistema para calcular cualquier función a la vez si lo replanteamos de la siguiente forma:

$$f(x) = A^T X + b \quad (2.7)$$

donde $b \in \mathbb{R}$ y $A^T, X \in \mathbb{R}^2$, así podemos formar los vectores con los siguientes valores:

$$A^T = [a_1 \ a_2], \quad X = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad (2.8)$$

y al sustituir estos valores en la ecuación 2.7 obtenemos:

$$f(x) = [a_1 \ a_2] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + b \quad (2.9)$$

que en forma de polinomio lo podemos expresar como:

$$f(x) = a_1 x_1 + a_2 x_2 + b \quad (2.10)$$

Siendo $x_1, x_2 \in \{0, 1\}$ las entradas del sistema, podemos sustituir su valor en las diferentes combinaciones de dos entradas y generar la siguiente tabla de verdad:

Tabla 2.2: Tabla de verdad para obtener las ecuaciones que se generan con las distintas combinaciones en una CLR de dos entradas

x_1	x_2	$f(x)$
0	0	b
0	1	$a_2 + b$
1	0	$a_1 + b$
1	1	$a_1 + a_2 + b$

Ahora bien, el valor de la salida se define como:

$$Y(f) = \begin{cases} 1, & |f(x)| < k \\ 0, & \text{de otra forma} \end{cases} \quad (2.11)$$

Donde, si el valor de la función se encuentra dentro de los límites del umbral establecido k , en la salida tendremos un 1 lógico, pero si el valor de la función está fuera de los límites, tendremos un 0 lógico. Dicho esto, podemos definir el diagrama de bloques de la CLR de dos entradas como se observa en la figura 2.1.

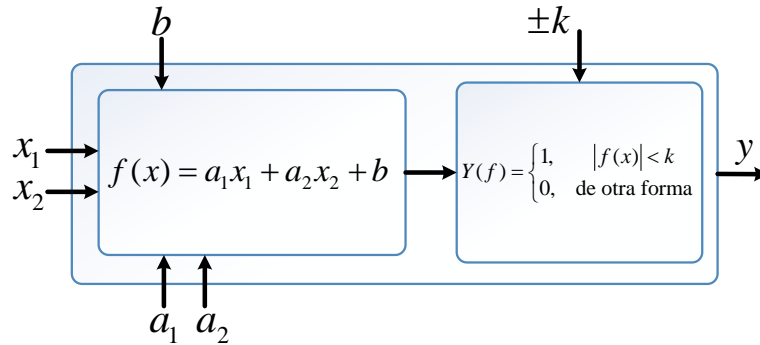


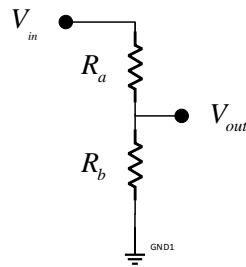
Figura 2.1: Diagrama de bloques de una compuerta lógica reconfigurable de dos entradas.

Se puede observar que en la parte izquierda se genera el valor de la función, mientras que en la parte derecha se encuentra la etapa de umbral que define el valor a la salida de la compuerta.

2.1.2. Circuito electrónico

El circuito electrónico propuesto básicamente se compone de dos etapas como se puede observar en el diagrama de la figura 2.1.

En la primera etapa se encuentran las señales de entrada x_1 y x_2 , las cuales pueden tomar un valor de voltaje alto o bajo. Estas señales están conectadas a los potenciómetros R_1 y R_2 respectivamente, que son los parámetros a_1 y a_2 del sistema 2.9. Estos potenciómetros actúan como divisores de tensión lo que permite realizar la multiplicación $A^T X$.



$$V_{out} = \frac{R_b}{R_a + R_b} V_{in} \quad (2.12)$$

Figura 2.2: Divisor de voltaje

Junto a estas señales tenemos también una señal de *offset* generada por una fuente de voltaje x_b en polarización negativa y que conectada en serie con el potenciómetro R_3 dan como resultado el parámetro b . El uso de potenciómetros permite variar el valor de los parámetros de entrada, y así poder sintonizar la compuerta para cualquier función. Es importante mencionar que todos los potenciómetros tienen el mismo valor nominal, $R_1 = R_2 = R_3 = 10k\Omega$.

Los voltajes que se obtiene a la salida de los potenciómetros entran, cada uno, a un amplificador operacional en configuración de seguidor unitario con el único propósito de acoplar impedancias, ya que en el sistema final va a existir interacción entre varias de estas compuertas lo que puede ocasionar pérdidas de tensión.

Una vez limpias las señales, todas se unen en la entradas negativa de un amplificador operacional en configuración de “*sumador inversor*” donde intencionalmente hacemos que las resistencias, incluyendo la de retroalimentación R_7 , tengan el mismo valor $R_4 = R_5 = R_6 = R_7 = 1k\Omega$, con esto logramos que la ganancia sea unitaria $A_v = 1$ y

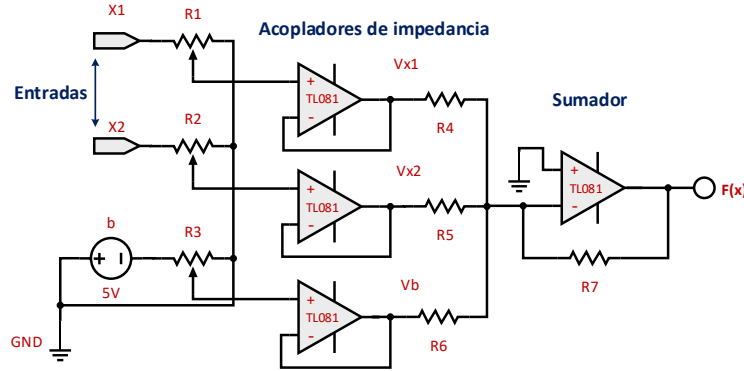


Figura 2.3: Primera parte del circuito: Divisores de voltaje y amplificador sumador

por lo tanto se desprecian los valores de las resistencias:

$$\frac{V_{x1}}{R_4} + \frac{V_{x2}}{R_5} + \frac{V_b}{R_6} = -\frac{V_f}{R_7} \quad (2.13)$$

despejando el voltaje de salida tenemos que:

$$V_f = -R_7 \left(\frac{V_{x1}}{R_4} + \frac{V_{x2}}{R_5} + \frac{V_b}{R_6} \right) \quad (2.14)$$

sabiendo que todas las resistencias tienen el mismo valor, la expresión queda de la siguiente forma:

$$V_f = -(V_{x1} + V_{x2} + V_b) \quad (2.15)$$

que es equivalente a la ecuación 2.10. De esta manera la primera parte del circuito queda como se puede observar en la figura 2.3, página 31.

En la segunda parte del circuito, encontramos el elemento que le va a dar la “no linealidad” al sistema, nos referimos a un comparador de tipo ventana, el cual esta compuesto por dos comparadores LM311 que junto a dos fuentes de referencia, con 1V, definen el umbral de salida $\pm k$. Así, si el voltaje en el nodo V_f se encuentra dentro de este umbral, el valor que tomará la salida Y será de 5V o bien un *nivel alto*, de lo

2.1. Compuerta Lógica Reconfigurable con dos entradas

contrario el valor que tome Y será de $0V$ o *nivel bajo*, tal como lo especifica la ecuación 2.11.

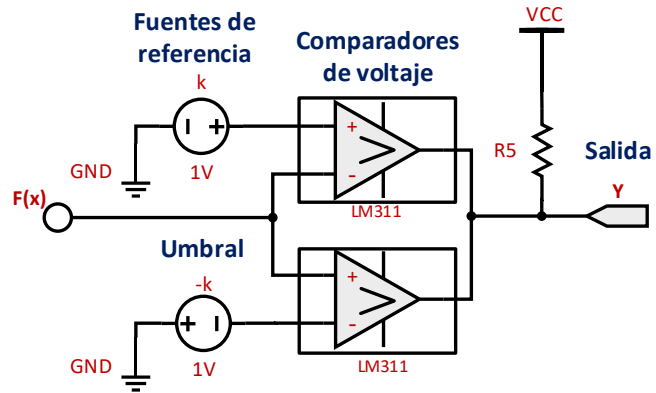


Figura 2.4: Segunda parte del Circuito: Elemento no lineal (comparador de ventana)

Teniendo bien definidas las dos partes del circuito, estas se unen por medio del nodo $f(x)$, dando como resultado el circuito para la CLR de dos entradas como se muestra en la figura 2.5.

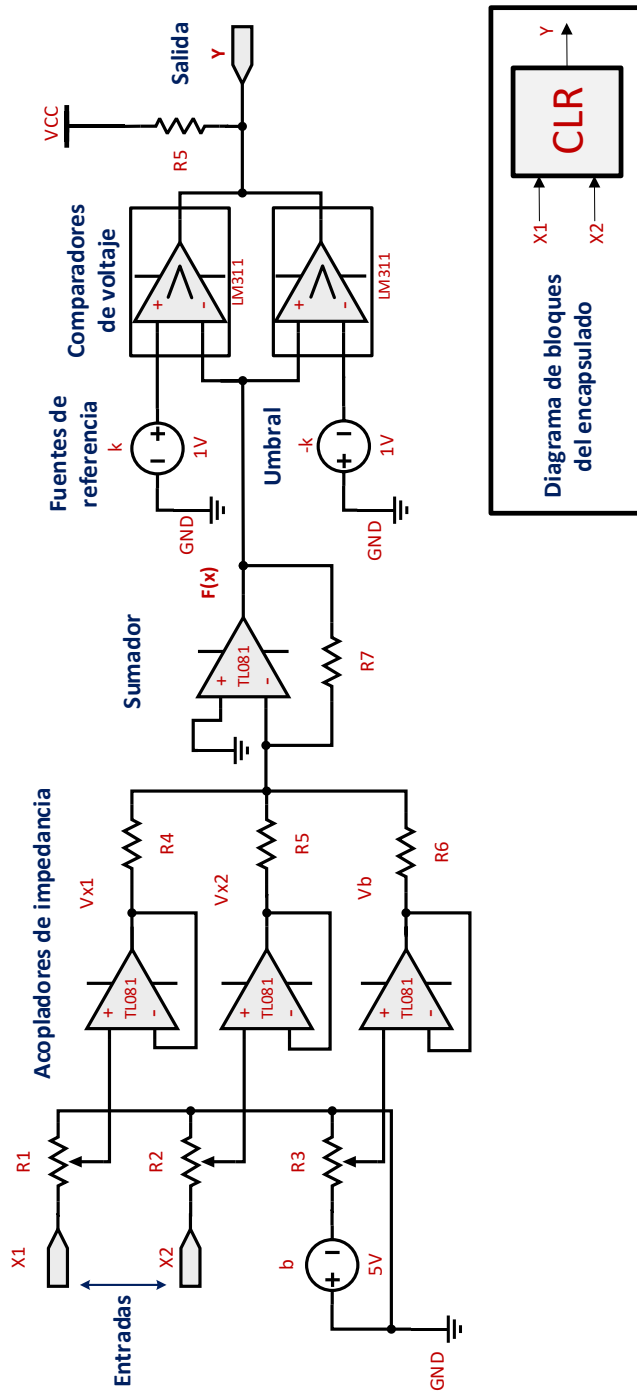


Figura 2.5: Circuito de la compuerta lógica reconfigurable de dos entradas

2.1.3. Resultados

En esta sección se presentan los resultados obtenidos al sintonizar las compuertas *AND*, *OR* y *XOR* en el dispositivo, los valores que deben tomar los parámetros de entrada se pueden consultar en la siguiente tabla:

Tabla 2.3: Valores para los parámetros de entrada del sistema

	x_1	x_2	b
AND	1.3	1.3	-2.5
OR	1.3	1.3	-2
XOR	2.5	2.5	-2.5

Sintonizando la compuerta AND

En la siguiente tabla se puede observar que los resultados para $f(x)$ solo caen dentro del umbral unicamente cuando x_1 y x_2 tienen un valor lógico de 1, dando como resultado el comportamiento del operador AND.

Tabla 2.4: Resultados de las ecuaciones del sistema al sintonizar la compuerta lógica *AND*

x_1	x_2	ecuación	$f(x)$	$y(x)$
0	0	b	-2.5	0
0	1	$a_2 + b$	-1.2	0
1	0	$a_1 + b$	-1.2	0
1	1	$a_1 + a_2 + b$	0.1	1

En la figura 2.6a se observa la señal que se obtiene en el nodo $f(x)$, donde se observa ahora experimentalmente los cambios de la función respecto a los valores de la entrada, donde solamente cae una vez dentro del umbral, cuando ambos valores son 5V. Esto se reafirma en 2.6b, donde se pueden observar los valores lógicos de las entradas en los canales 1 y 2, y la respuesta de la salida $y(x)$ en el canal 3.

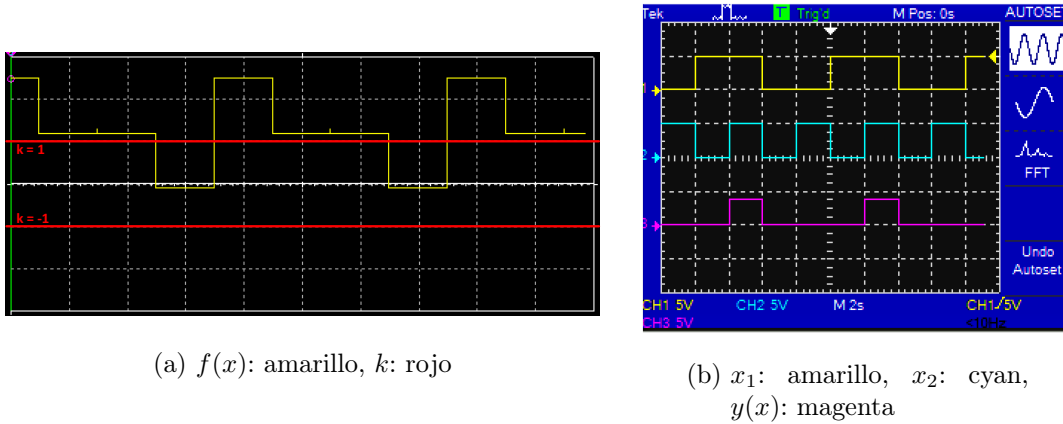


Figura 2.6: Compuerta lógica AND

Sintonizando la compuerta OR

En la tabla 2.5 se puede observar que los resultados para $f(x)$ caen dentro del umbral cuando x_1 , x_2 o ambos tienen un valor lógico de 1, dando como resultado el comportamiento del operador OR.

Tabla 2.5: Resultados de las ecuaciones del sistema al sintonizar la compuerta lógica OR

x_1	x_2	ecuación	$f(x)$	$y(x)$
0	0	b	-2.0	0
0	1	$a_2 + b$	-0.7	1
1	0	$a_1 + b$	-0.7	1
1	1	$a_1 + a_2 + b$	0.6	1

En la figura 2.7a se observa experimentalmente la señal que se obtiene en el nodo $f(x)$, donde solamente cae fuera del umbral cuando x_1 y x_2 valen 0V. En 2.7b se puede observar los valores lógicos de las entradas en los canales 1 y 2, y la respuesta de la salida $y(f)$ en el canal 3.

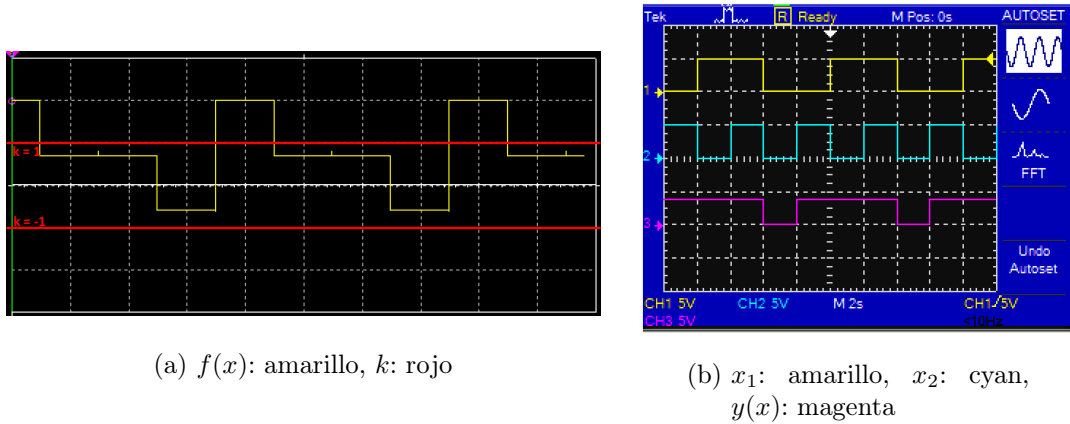


Figura 2.7: Compuerta lógica OR

Sintonizando la compuerta XOR

En la tabla 2.5 se muestra que el resultado para $f(x)$ es 1 solamente cuando x_1 o x_2 , pero no ambos, tienen un valor lógico de 1. Dando así como resultado el operador XOR. Este mismo comportamiento se puede observar en la regla 90 de los Autómatas Celulares.

Tabla 2.6: Resultados de las ecuaciones del sistema al sintonizar la compuerta lógica XOR

x_1	x_2	ecuación	$f(x)$	$y(x)$
0	0	b	-2.5	0
0	1	$a_2 + b$	0	1
1	0	$a_1 + b$	0	1
1	1	$a_1 + a_2 + b$	-2.5	0

En la figura 2.8a se observa que la señal $f(x)$ solamente cae dentro del umbral cuando alguna de las entradas esta en alto, pero no ambas. En 2.8b se puede observar los valores lógicos de las entradas en los canales 1 y 2, y la respuesta de la salida $y(f)$ en el canal 3.

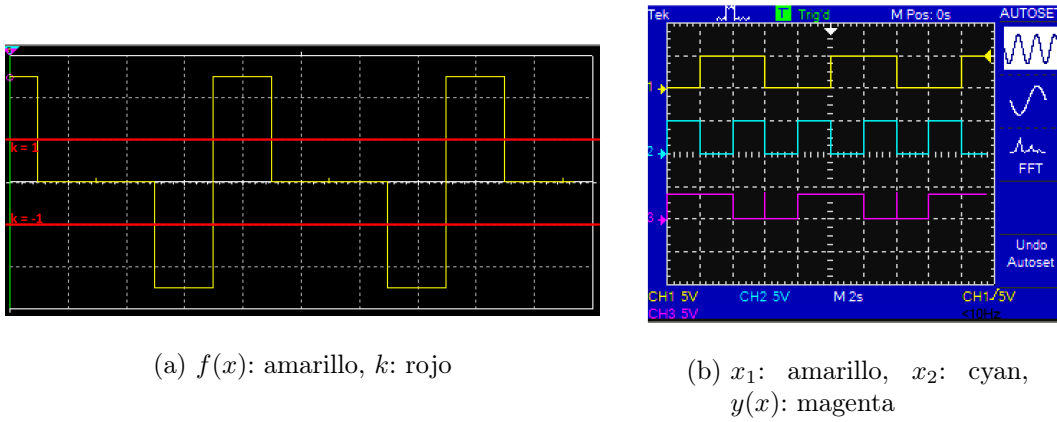


Figura 2.8: Compuerta lógica *XOR*

2.2. Sistema empleando la compuerta reconfigurable

Teniendo diseñada la CLR, procedemos a implementar un sistema previo, en el cual aprovechando que las reglas 90, 160 y 250 solo necesitan de los estados de sus vecinos podemos generar evolución de un 1D-CA con la CLR de 2 entradas.

$$r_{90} = x_t(i - 1) \oplus x_t(i + 1)$$

$$r_{160} = x_t(i - 1) \bullet x_t(i + 1)$$

$$r_{250} = x_t(i - 1) + x_t(i + 1)$$

El sistema se compone de dos etapas, en la primera tenemos un arreglo de cinco CLR, el cual podemos asociar al anillo de la figura 2.9 donde cada célula es una CLR, y de este modo se forma el elemento básico para el 1D-CA del sistema.

La segunda etapa del circuito es un arreglo de *flip-flops* tipo D que conforman un elemento de memoria; este es necesario para guardar el valor del 1D-CA en un instante de tiempo previo y que de esta forma se pueda generar la evolución de manera periódica en cada ciclo de reloj.

2.2. Sistema empleando la compuerta reconfigurable

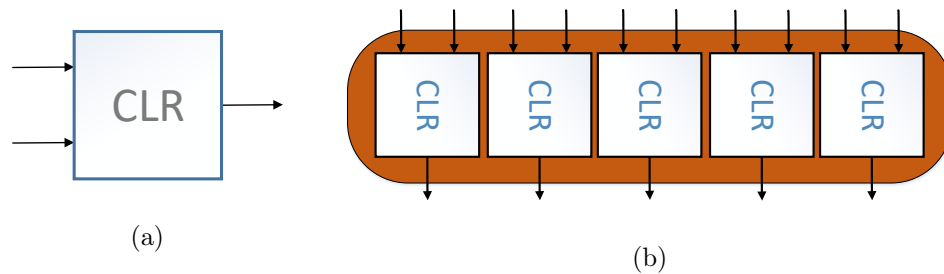


Figura 2.9: a) Representación en bloque de la *CLR* de dos entradas (célula del sistema), b) Representación del *1D-CA* con semilla de 5 elementos

Las razones por la que se usaron *flip-flops* tipo D son varias, en primer lugar la respuesta que otorga a la salida Q en configuración normal o “*set*” es el mismo valor que tiene en la entrada D , con lo que se logra guardar el dato sin modificar el sistema; en segundo lugar el *flip-flop* gracias a su entrada de reloj nos aporta una escala de tiempo del sistema, según la frecuencia de reloj que entre por aquí será la base de tiempo discreto sobre la que se genera cada iteración; y en tercer lugar, y de gran relevancia, es que gracias a los modos *preset* y *clear* con que cuenta el *flip-flop* se puede definir y guardar la semilla del sistema antes del primer ciclo de reloj, ya que como se puede observar en la tabla de verdad 2.7, estos modos se activan de manera asíncrona al recibir una señal en bajo sin importar lo que se tenga en las demás entradas. Al activar *preset* siempre se tendrá un nivel alto a la salida Q , y al activar *clear* siempre existirá un nivel bajo en Q .

Tabla 2.7: Tabla de verdad de un Flip-Flop tipo D

Operación	Entradas				Salidas	
	Preset	Clear	CLK	D	Q	Q'
Set	1	1	▲	1	1	0
Reset	1	1	▲	0	0	1
Preset	0	1	x	x	1	0
Clear	1	0	x	x	0	1
Hold	1	1	0	x	Q_0	Q'_0

2.2.1. Circuito electrónico

Teniendo en cuenta lo anterior, el procedimiento para realizar el circuito queda de la siguiente forma:

1. Se forma un 1D-CA en forma de anillo de cinco celdas contiguas, la salida de cada celda se conecta a una entrada de *flip-flop*.
2. Las entradas de *preset* y *clear* de todos los *flip-flops* van conectadas mediante un dip switch a una fuente de alimentación de 5V, así podemos generar distintas semillas al activar y desactivar estas entradas en los distintos *flip-flops*. Para este trabajo, por generalidad, solo se activa el *preset* del tercer *flip-flop* con lo que el elemento central de la semilla es el único que se pone en estado alto antes de la primera iteración.
3. En las entradas de reloj se conecta cualquier fuente que nos genere pulsos cuadrados continuos a una frecuencia determinada y con ciclo de trabajo 50% - 50%, como lo pueden ser un generador de ondas cuadradas o un circuito 555.
4. Como último paso se genera la retroalimentación del circuito conectando las salidas de los *flip-flops* a las entradas de las CLR, esto permite que cada estado que se va guardando en los *flip-flops* sea una nueva semilla para el instante de tiempo siguiente y así el sistema evoluciona de manera autónoma hasta que dejemos de proveerle la señal de reloj o en su defecto cortemos la alimentación del circuito.

Las conexiones entre las salidas Q_n de los *flip-flops* y las entradas x_1 y x_2 de cada compuerta se realizan de la siguiente forma:

$$\begin{array}{l}
 Q_1 \left\{ \begin{array}{l} \text{Compuerta 5} \Rightarrow x_2 \\ \text{Compuerta 2} \Rightarrow x_1 \end{array} \right. \\
 \\
 Q_2 \left\{ \begin{array}{l} \text{Compuerta 1} \Rightarrow x_2 \\ \text{Compuerta 2} \Rightarrow x_1 \end{array} \right. \\
 \\
 Q_3 \left\{ \begin{array}{l} \text{Compuerta 2} \Rightarrow x_2 \\ \text{Compuerta 4} \Rightarrow x_1 \end{array} \right. \\
 \\
 Q_4 \left\{ \begin{array}{l} \text{Compuerta 3} \Rightarrow x_2 \\ \text{Compuerta 5} \Rightarrow x_1 \end{array} \right. \\
 \\
 Q_5 \left\{ \begin{array}{l} \text{Compuerta 4} \Rightarrow x_2 \\ \text{Compuerta 1} \Rightarrow x_1 \end{array} \right.
 \end{array}$$

2.2. Sistema empleando la compuerta reconfigurable

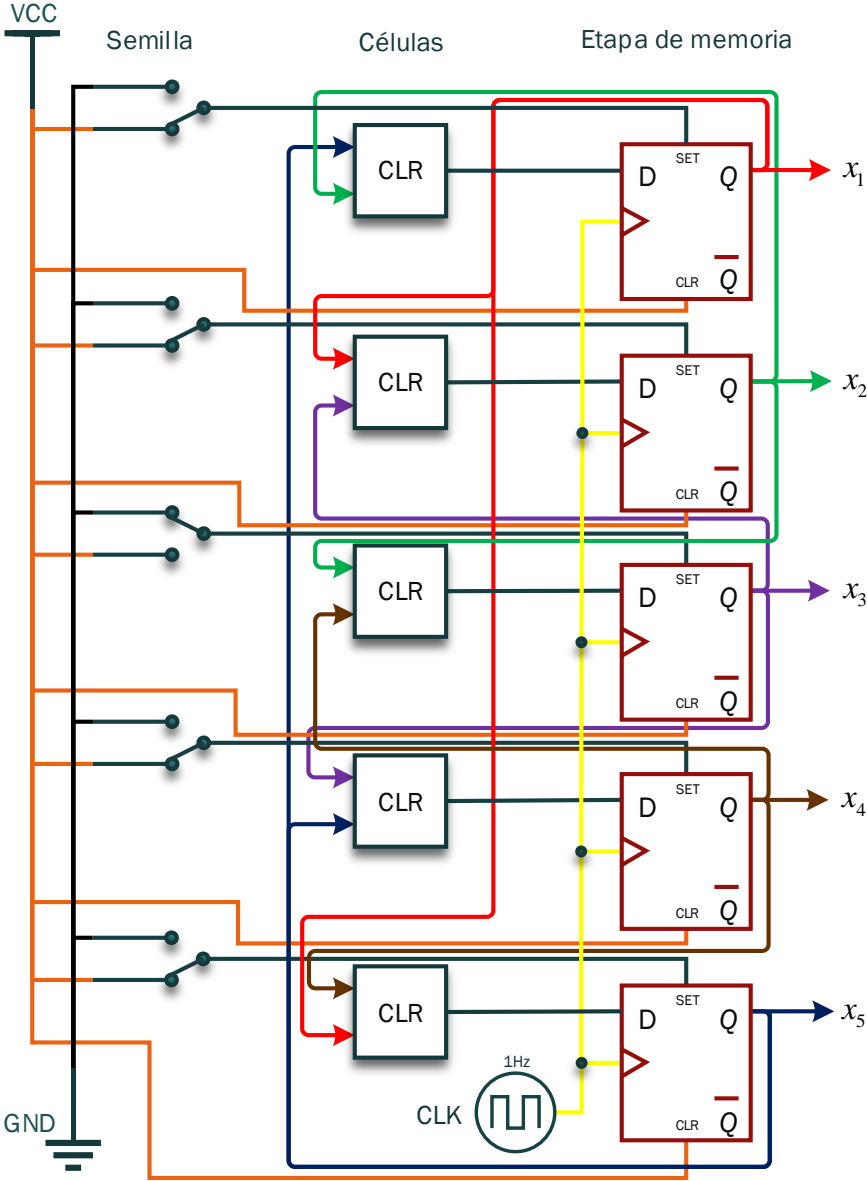


Figura 2.10: Circuito electrónico del sistema empleando la CLR de dos entradas

2.2.2. Resultados

A continuación se presentan los resultados obtenidos, en la parte izquierda de las figuras se observan simulaciones matemáticas en 20 iteraciones de las reglas 160, 250 y 90, las cuales tienen el comportamiento de los operadores *AND*, *OR* y *XOR*, de dos entradas, respectivamente. En el lado derecho de las figuras se muestra una secuencia de imágenes con la evolución del sistema mostrado en una barra de LED's, donde se puede observar que a partir de la quinta iteración el patrón de evolución se repite. Se puede comprobar que la evolución se lleva a cabo de forma correcta al comparar la secuencia de leds con la simulación del 1D-CA.

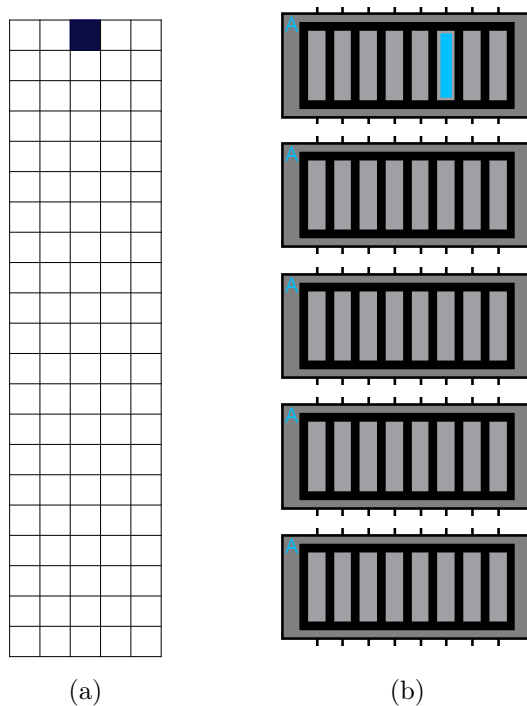


Figura 2.11: Evolución de la Regla 160 ($x_1 \bullet x_3$): a) Simulación matemática, b) Respuesta del sistema

Figura 2.12: Evolución de la Regla 250, $(x_1 + x_3)$: a) Simulación matemática, b) Respuesta del sistema

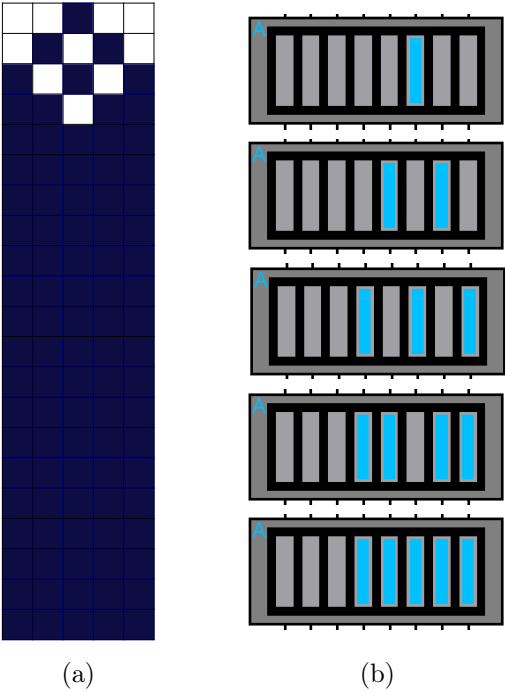
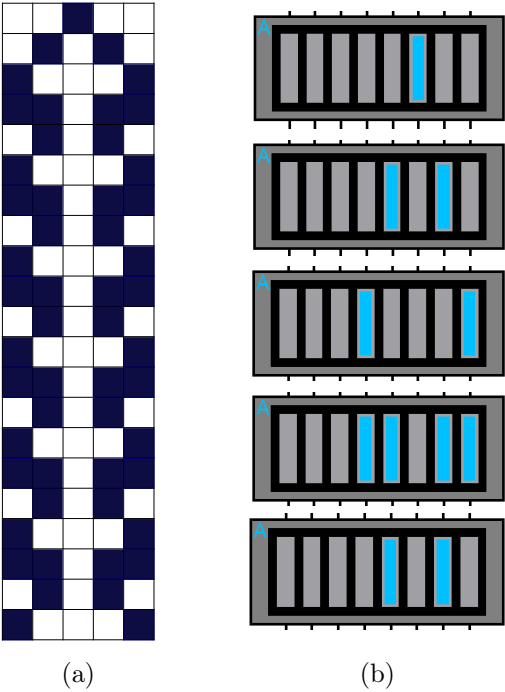


Figura 2.13: Evolución de la Regla 90, $(x_1 \oplus x_3)$: a) Simulación matemática, b) Respuesta del sistema



Capítulo 3

Sistema con célula de tres entradas

El objetivo principal del proyecto, es diseñar un circuito con el cual se pueda generar “físicamente” la evolución de las reglas propuestas por Wolfram para los 1D-CA, y que a la vez tenga la capacidad de reconfigurarse en tiempo real. Como se menciona en el capítulo anterior es imposible generar todas las reglas con un dispositivo de dos entradas, ya que el vecindario elemental de los 1D-CA consta de 3 elementos, sin embargo aprovechando la naturaleza de algunas reglas, como la $r90$, nos fue posible lograr obtener resultados con un sistema más sencillo.

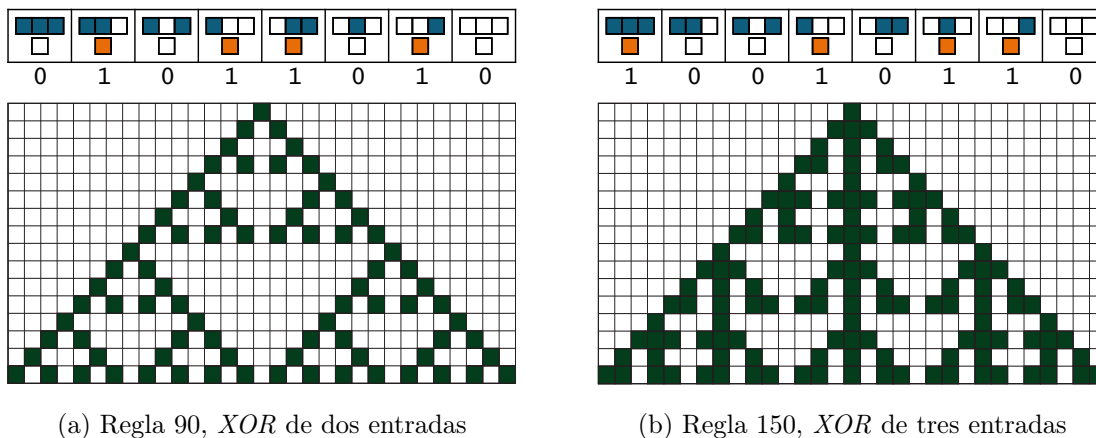


Figura 3.1: Reglas de evolución (arriba) y diagramas *espacio-tiempo* (abajo) de Autómatas Celulares Unidimensionales

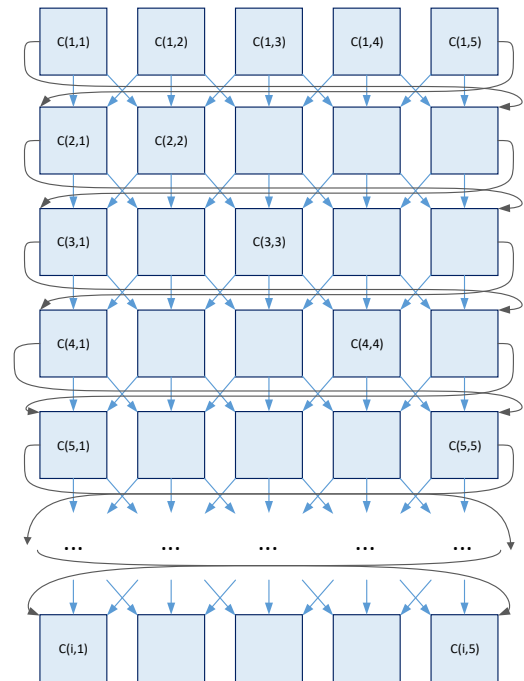
Así, basados en el diseño de la compuerta lógica reconfigurable de dos entradas, modificamos el diseño para realizar un dispositivo agregando una tercera entrada, el

3.1. Célula con tres entradas

cual nos permita conformar un vecindario de tres elementos, con el cual evolucionamos los Autómatas Celulares que obedecen a las reglas 10, 90, 110, 126 y 254. Aunque solo se presentan resultados para las reglas mencionadas, gracias a la inclusión de la tercera entrada es posible generalizar el sistema a cualquiera de las 256 reglas sintonizando de forma adecuada los parámetros de las compuertas lógicas que funcionan como células del sistema.

Es importante mencionar que aunque las unidades de tiempo en la red son discretas, cada una de sus células trabaja mediante sistemas lineales en tiempo continuo, lo que nos da la posibilidad de cambiar su comportamiento en tiempo real, debido a esto podemos decir que el sistema es una Red No lineal Celular (fig.3.2).

Figura 3.2: Topología básica de la CNN diseñada, contiene elementos en la semilla que evoluciona durante n número de iteraciones, su evolución la realiza como lo hacen los 1D-C



3.1. Célula con tres entradas

A continuación se realiza el análisis matemático de la célula reconfigurable con tres entradas y se muestran los resultados al obtener las funciones de las reglas 10, 90, 110,

126 y 254 cuyas respuestas teóricas podemos observar en la siguiente tabla de verdad:

Tabla 3.1: Tabla de verdad para las reglas 10, 90,110, 126 y 254

Entradas			Función				
x_1	x_2	x_3	r_{10}	r_{90}	r_{110}	r_{126}	r_{254}
0	0	0	0	0	0	0	0
0	0	1	1	1	1	1	1
0	1	0	0	0	1	1	1
0	1	1	1	1	1	1	1
1	0	0	0	1	0	1	1
1	0	1	0	0	1	1	1
1	1	0	0	1	1	1	1
1	1	1	0	0	0	0	1

3.1.1. Análisis matemático

Tomando el sistema generalizado que se propuso en la ecuación 2.7,

$$f(x) = A^T X + b$$

agregamos un elemento más a los vectores A^t y X , con lo cual $b \in \mathbb{R}$ y $A^T, X \in \mathbb{R}^3$. Por lo tanto el sistema ahora queda de la siguiente forma:

$$f(x) = \begin{bmatrix} a_1 & a_2 & a_3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + b \quad (3.1)$$

que equivalentemente podemos representarlo como:

$$f(x) = a_1x_1 + a_2x_2 + a_3x_3 + b \quad (3.2)$$

Al contar con un elemento más en el vector X se duplica el numero de combinaciones posibles, por lo tanto se eleva el número de funciones que se pueden obtener pasando de $2^4 = 16$ en un sistema de dos entradas a $2^8 = 256$. En la siguiente tabla de verdad se aprecian las ecuaciones resultantes para cada combinación de entradas $x_1, x_2, x_3 \in \{0, 1\}$.

3.1. Célula con tres entradas

Tabla 3.2: Tabla de verdad para obtener las ecuaciones que se generan con las distintas combinaciones en una CLR de tres entradas

x_1	x_2	x_3	ecuación
0	0	0	b
0	0	1	$x_3 + b$
0	1	0	$x_2 + b$
0	1	1	$x_2 + x_3 + b$
1	0	0	$x_1 + b$
1	0	1	$x_1 + x_3 + b$
1	1	0	$x_1 + x_2 + b$
1	1	1	$x_1 + x_2 + x_3 + b$

El valor de la salida lo podemos definir de la misma forma que en la ecuación 2.11 como:

$$Y(f) = \begin{cases} 1, & |f(x)| < k \\ 0, & \text{de otra forma} \end{cases}$$

Donde se obtiene un nivel alto si el valor de $f(x)$ cae dentro del umbral $\pm k$, de lo contrario la salida toma un nivel bajo. Así, podemos definir el diagrama de bloques de la CLR de tres entradas como se observa en el diagrama de bloques 3.3.

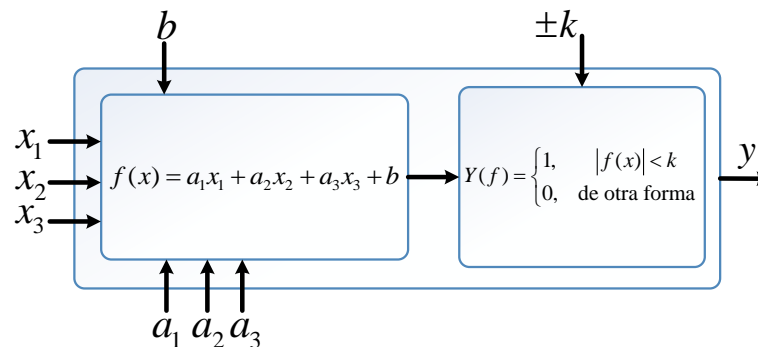


Figura 3.3: Diagrama de bloques de una compuerta lógica reconfigurable de tres entradas.

3.1.2. Circuito electrónico

El añadir los parámetros x_3 al vector X y a_3 a A^T , podemos diseñar el circuito de la siguiente forma:

En la primera etapa tenemos tres señales de entrada representadas por x_1, x_2 y x_3 , las cuales solo pueden tomar un valor de voltaje alto o bajo. Posteriormente estas entradas se conectan en serie a los potenciómetros R_1, R_2 y R_3 que actúan como los parámetros a_1, a_2 y a_3 del sistema 3.1: su función es dividir la tensión de las entradas, con lo que se obtiene la multiplicación $A^T X$. Nuevamente nos encontramos con la señal de *offset* generada por una fuente de voltaje x_b en polarización negativa, la cual al ser conectada en serie con el potenciómetro R_4 nos arroja el parámetro b , resultado del divisor de tensión. Al igual que en el sistema de dos entradas los potenciómetros tienen el mismo valor nominal, $R_1 = R_2 = R_3 = R_4 = 10k\Omega$. Los voltajes que se obtiene a la salida de los potenciómetros entran a amplificadores operacionales conectados como seguidor unitario con el propósito de acoplar impedancias para evitar fugas de tensión.

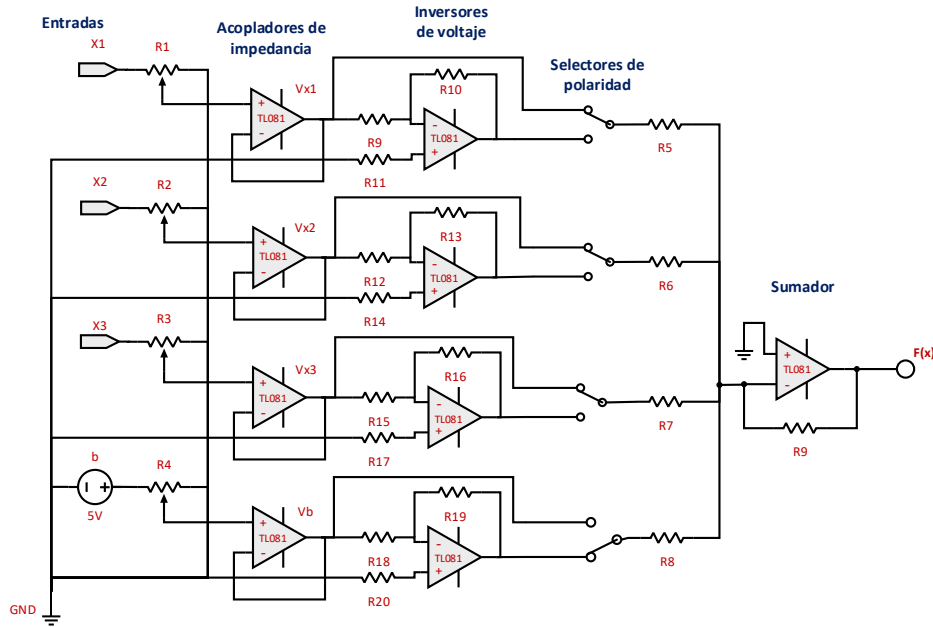


Figura 3.4: Primera parte del circuito: Divisores de voltaje, inversores de polaridad y amplificador sumador

3.1. Célula con tres entradas

Las señales que salen de los acopladores de impedancia se unen a la entradas negativa de un amplificador operacional TL084 conectado como sumador inversor, e intencionalmente hacemos que las resistencias, junto con la de retroalimentación R_9 , tengan el mismo valor $R_5 = R_6 = R_7 = R_8 = R_9 = 1k\Omega$, para de esta forma tener ganancia unitaria $A_v = 1$ y así poder sumar los voltajes tal como vienen de las entradas, por lo tanto podemos tener las siguientes ecuaciones:

$$\frac{V_{x1}}{R_5} + \frac{V_{x2}}{R_6} + \frac{V_{x3}}{R_7} + \frac{V_b}{R_8} = -\frac{V_f}{R_9} \quad (3.3)$$

despejando el voltaje de salida tenemos que:

$$V_f = -R_9 \left(\frac{V_{x1}}{R_5} + \frac{V_{x2}}{R_6} + \frac{V_{x3}}{R_7} + \frac{V_b}{R_8} \right) \quad (3.4)$$

como sabemos que todas las resistencias son iguales, nos queda la siguiente ecuación:

$$V_f = V_{x1} + V_{x2} + V_{x3} + V_b \quad (3.5)$$

que es equivalente a la ecuación 3.2.

La segunda parte del circuito es en teoría idéntica a la del sistema de dos entradas, sin embargo se le realizó una ligera modificación para optimizar el circuito y sobre todo reducir costos para su implementación física.

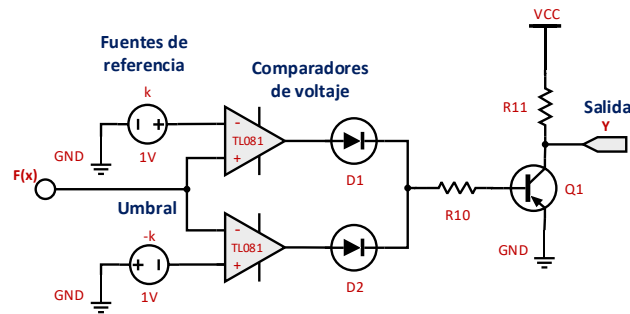


Figura 3.5: Comparador de ventana con amplificadores operacionales TL081 y transistor

Esta consta de cambiar los comparadores LM311 por un diseño con amplificadores operacionales convencionales, en este caso TL084, que son utilizados por el sistema en general. La conexión de las fuentes de referencia al TL084 se hace para k por la terminal negativa y para $-k$ por la positiva; para evitar que las salidas de los diodos se sobrecarguen al juntarse, es preciso conectar un par de diodos 1N4007 a estas respectivamente; finalmente se conecta un transistor que sirve como inversor de voltaje y a su vez satura cuando V_f cae dentro del umbral haciendo que $Y = 5V$, de lo contrario $Y = 0V$, tal como lo especifica la ecuación 2.11, página 29.

El circuito de la compuerta lógica reconfigurable de tres entradas lo podemos observar en la figura 3.6, página 50.

3.1.3. Resultados

En esta sección se presentan los resultados obtenidos al sintonizar las reglas 10, 90, 110, 126 y 254 en el dispositivo. Los valores que deben tomar los parámetros de entrada se pueden consultar en la siguiente tabla:

Tabla 3.3: Valores para los parámetros de entrada del sistema

	x_1	x_2	x_3	b
regla 10	0.8	-0.2	3	-1.1
regla 90	2	0.5	1.5	-2
regla 126	1.2	1.2	2	-2
regla 254	0.7	0.7	0.7	-1.5

Estos valores al ser substituidos en la ecuación $f(x) = a_1x_1 + a_2x_2 + a_3x_3 + b$ (3.2) dan como resultado el comportamiento de cada regla a obtener como se definen en las tablas 3.4, 3.5, 3.6, 3.7, 3.8 y se observa en las figuras 3.7, 3.8, 3.9, 3.10, 3.11.

3.1. Célula con tres entradas

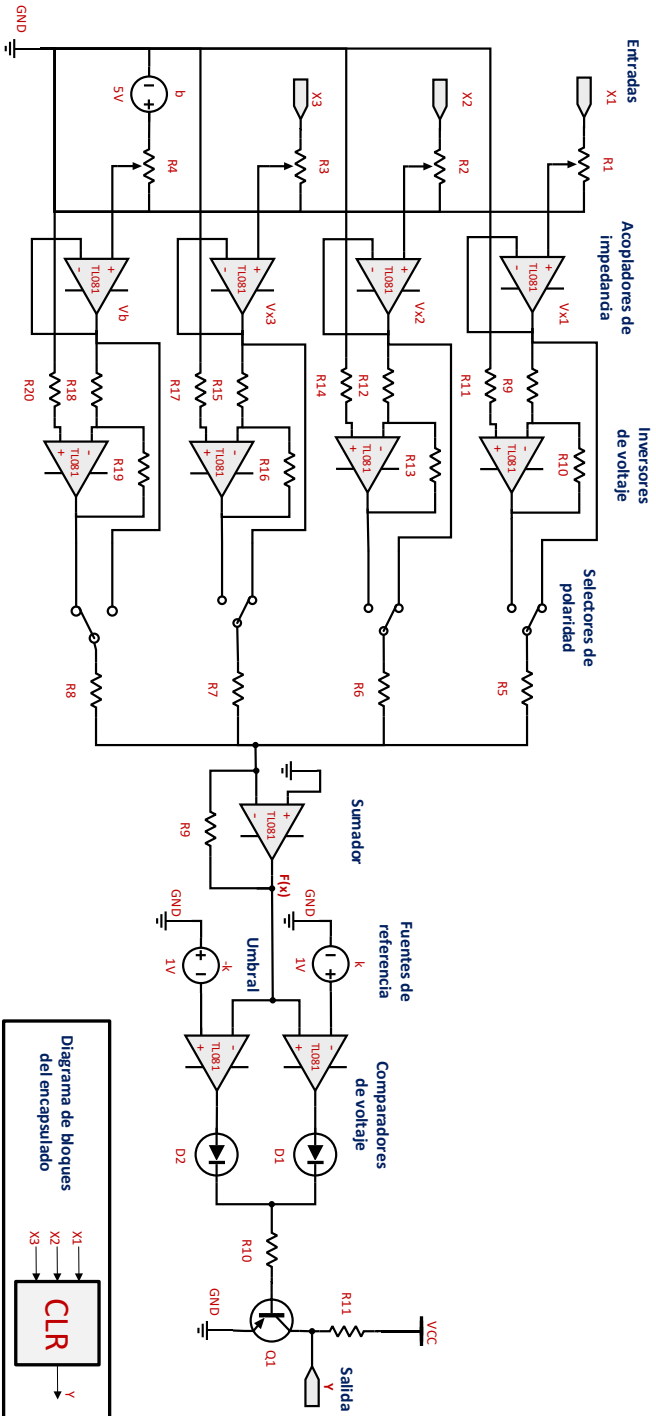


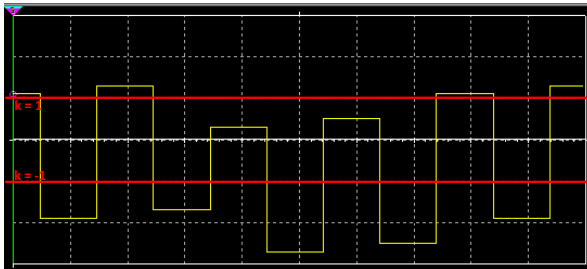
Figura 3.6: Circuito de la compuerta lógica reconfigurable de tres entradas

Sintonizando la regla 10

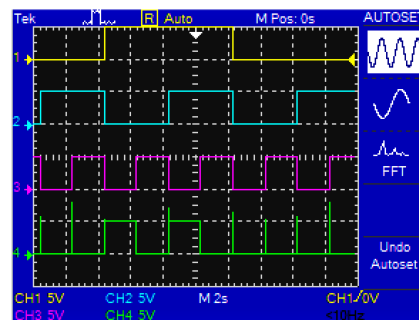
La tabla 3.4 muestra que teóricamente solo debe existir un nivel alto a la salida para las combinaciones $x_1 = 0, x_2 = 0, x_3 = 1$ y $x_1 = 0, x_2 = 1, x_3 = 1$. Al observar lo que pasa en el $f(x)$ en la figura 3.7a comprobamos que efectivamente solo se encuentra dentro del umbral cuando ocurren estas combinaciones, así podemos ver en la figura 3.7b el valor lógico de la función con respecto a las distintas combinaciones en las entradas del sistema. Cabe mencionar que, como se ve en la figura 3.15a, por falta de simetría esta regla es considerada “*ilegal*”.

Tabla 3.4: Tabla para obtener los valores que sintonizan la regla 10

x_1	x_2	x_3	ecuación	r_{10}	valor
0	0	0	b	0	-1.1
0	0	1	$x_3 + b$	1	-0.3
0	1	0	$x_2 + b$	0	-1.3
0	1	1	$x_2 + x_3 + b$	1	-0.5
1	0	0	$x_1 + b$	0	1.9
1	0	1	$x_1 + x_3 + b$	0	2.7
1	1	0	$x_1 + x_2 + b$	0	1.7
1	1	1	$x_1 + x_2 + x_3 + b$	0	2.5



(a) $f(x)$: amarillo, k : rojo



(b) x_1 : amarillo, x_2 : cian, x_3 : magenta, $y(x)$: verde

Figura 3.7: Regla 10

Sintonizando la regla 90

La regla 90 de los autómatas celulares es una de las más estudiadas ya que presenta ciertas características, algunas de ellas son:

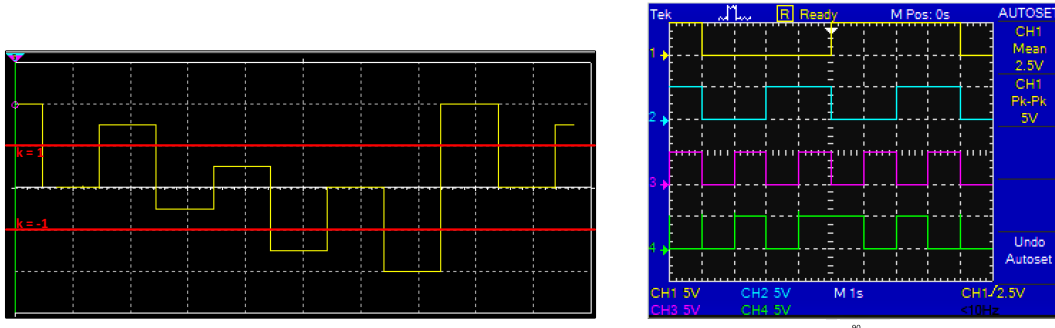
- Cumple con superposición aditiva.
- Es de comportamiento periférico, es decir, que su evolución solo depende de sus vecinos y no de ella misma en un estado anterior.
- Es el “operador módulo dos”, mejor conocido como *XOR*, entre dos valores booleanos.

Como podemos observar en la tabla, su vector de resultados tiene reflexión simétrica ya que la primera mitad de los valores se repiten pero en forma de espejo $0101 \rightarrow 1010$. Otro patrón interesante es que si observamos los cuatro valores en la parte central del vector de resultados podemos ver claramente la respuesta que se obtiene al usar el operador *XOR* de dos entradas, y esto se puede observar nuevamente al juntar los primeros dos y los últimos dos valores de este mismo vector; también es interesante observar que al tener reflexión simétrica, los últimos cuatro valores del vector tienen el comportamiento de un operador *XNOR* de dos entradas.

Tabla 3.5: Tabla de verdad para obtener los valores que sintonizan la regla 90

x_1	x_2	x_3	ecuación	r_{90}	valor
0	0	0	b	0	-2
0	0	1	$x_3 + b$	1	-0.5
0	1	0	$x_2 + b$	0	-1.5
0	1	1	$x_2 + x_3 + b$	1	0
1	0	0	$x_1 + b$	1	0
1	0	1	$x_1 + x_3 + b$	0	1.5
1	1	0	$x_1 + x_2 + b$	1	0.5
1	1	1	$x_1 + x_2 + x_3 + b$	0	2

En la figura 3.8a se pueden observar los valores que caen dentro del umbral, y en 3.8b observamos la respuesta del sistema en función de los valores que toman las entradas.



(a) $f(x)$: amarillo, k : rojo

(b) x_1 : amarillo, x_2 : cian, x_3 : magenta, $y(x)$: verde

Figura 3.8: Regla 90

Sintonizando la regla 110

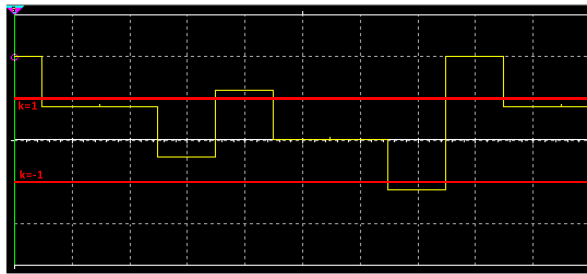
La regla 110 también es de las más estudiadas, ya que a pesar de ser considerada “*illegal*” debido su falta de simetría reflectiva, ésta presenta comportamientos de mucho interés en el campo de la encriptación y seguridad informática debido a su carácter *complejo*. Se dice que esta regla es la única de las 256 que se ha comprobado ser “*Turing Completa*”, esto quiere decir que tiene el poder computacional de una Máquina Universal de Turing. Matthew Cook probó que la regla 110 es capaz de soportar computación universal, siendo esta un sistema lo suficientemente simple para indicar que sistemas físicos que ocurren en la naturaleza también pueden ser universales, esto significa que muchas de sus propiedades pueden ser indeseables y pueden no tener soluciones matemáticas de forma cerrada.

Al observar la tabla 3.6, nos podemos percatar que de las tres combinaciones que nos producen un nivel bajo, dos son cuando o todas las entradas están activas o bien todas inactivas, y la tercera es cuando solamente x_3 se encuentra en alto. Esto se corrobora en la figura 3.9 donde vemos en 3.9a los valores que caen dentro del umbral, y en 3.9b la respuesta del sistema.

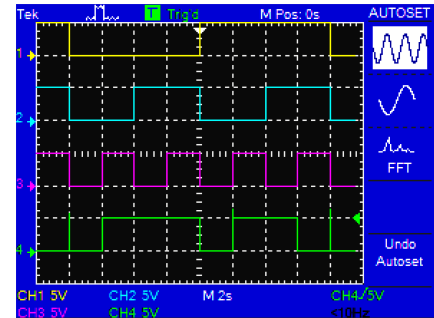
3.1. Célula con tres entradas

Tabla 3.6: Tabla de verdad para obtener los valores que sintonizan la regla 110

x_1	x_2	x_3	ecuación	r_{254}	valor
0	0	0	b	0	-2
0	0	1	$x_3 + b$	1	-0.8
0	1	0	$x_2 + b$	1	-0.8
0	1	1	$x_2 + x_3 + b$	1	0.4
1	0	0	$x_1 + b$	0	-1.2
1	0	1	$x_1 + x_3 + b$	1	0
1	1	0	$x_1 + x_2 + b$	1	0
1	1	1	$x_1 + x_2 + x_3 + b$	0	1.2



(a) $f(x)$: amarillo, k : rojo



(b) x_1 : amarillo, x_2 : cian, x_3 : magenta, $y(x)$: verde

Figura 3.9: Regla 110

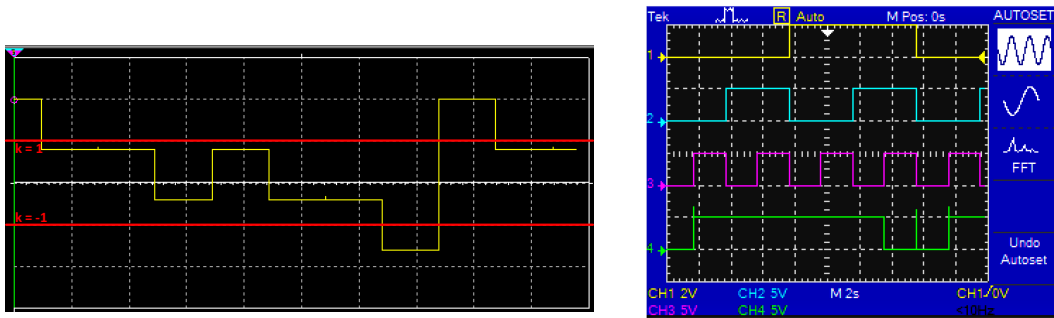
Sintonizando la regla 126

Al igual que la regla 90, el comportamiento de esta regla se considera caótico. Se puede observar en su tabla de verdad que, a excepción de cuando todas las entradas son activas o inactivas, las demás combinaciones producen un nivel alto, lo que conduce a un patrón evolutivo interesante; observamos que si la evolución se inicia con una semilla donde la única celda activa sea la central, las generaciones siguientes se van formando de la siguiente forma: 1, 7, 27, 127, 387, ..., etc, que en forma binaria se representan como: 1, 111, 11011, 1111111, 110000011, ..., etc. (fig. 3.17).

Tabla 3.7: Tabla de verdad para obtener los valores que sintonizan la regla 126

x_1	x_2	x_3	ecuación	r_{126}	valor
0	0	0	b	0	-2
0	0	1	$x_3 + b$	1	-0.8
0	1	0	$x_2 + b$	1	-0.8
0	1	1	$x_2 + x_3 + b$	1	0.4
1	0	0	$x_1 + b$	1	-0.8
1	0	1	$x_1 + x_3 + b$	1	0.4
1	1	0	$x_1 + x_2 + b$	1	0.4
1	1	1	$x_1 + x_2 + x_3 + b$	0	1.6

En el osciloscopio se pueden ver muy marcados sus dos estados inactivos, quedando fuera del umbral (fig 3.10a). En la figura 3.10b se puede observar la respuesta del sistema con respecto a los cambios en las entradas.



(a) $f(x)$: amarillo, k : rojo

(b) x_1 : amarillo, x_2 : cian, x_3 : magenta, $y(x)$: verde

Figura 3.10: Regla 126

Sintonizando la regla 254

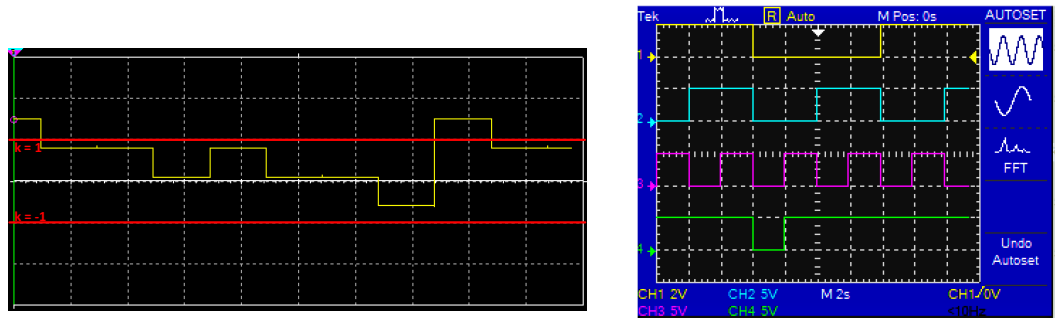
La regla 254 es de carácter *fijo*, lo que se puede observar en su tabla de verdad donde la única combinación que nos genera un estado bajo en la función es cuando todas las entradas se encuentran inactivas, esto se puede corroborar en la imagen 3.11, donde se observa en 3.11a que a excepción de una, todas las combinaciones están dentro del umbral, y en 3.11b podemos constatar que esta combinación es $x_1 = 0$, $x_2 = 0$,

3.2. Diseño de la CNN

$x_3 = 0$. Esto da como resultado que por cada célula que se encuentre activa en un estado de tiempo anterior, se activan la misma célula y sus vecinos contiguos en un paso de tiempo siguiente.

Tabla 3.8: Tabla de verdad para obtener los valores que sintonizan la regla 254

x_1	x_2	x_3	ecuación	r_{254}	valor
0	0	0	b	0	-1.5
0	0	1	$x_3 + b$	1	-0.8
0	1	0	$x_2 + b$	1	-0.8
0	1	1	$x_2 + x_3 + b$	1	-0.1
1	0	0	$x_1 + b$	1	-0.8
1	0	1	$x_1 + x_3 + b$	1	-0.1
1	1	0	$x_1 + x_2 + b$	1	-0.1
1	1	1	$x_1 + x_2 + x_3 + b$	1	0.6



(a) $f(x)$: amarillo, k : rojo

(b) x_1 : amarillo, x_2 : cian, x_3 : magenta, $y(x)$: verde

Figura 3.11: Regla 254

3.2. Diseño de la CNN

Para la construcción de la CNN se tomó como base el sistema realizado en el capítulo anterior, simplemente se cambia la CLR de dos entradas por la recién diseñada de tres entradas; lo que como ya se ha mencionado nos permite generar el vecindario

elemental de los 1D-CA, de esta manera se procedió a implementar un sistema con el que se tiene la posibilidad de generar la evolución para cualquier regla de 1D-CA. Al igual que en el sistema previo el circuito final se compone de dos etapas, un arreglo de cinco CLR de tres entradas que forman la semilla del 1D-CA, la cual después de n iteraciones formará la *red* del sistema, seguido del mismo arreglo de *flip-flops* tipo D que actúan como elemento de memoria para llevar a cabo la retroalimentación, y así poder generar la evolución.

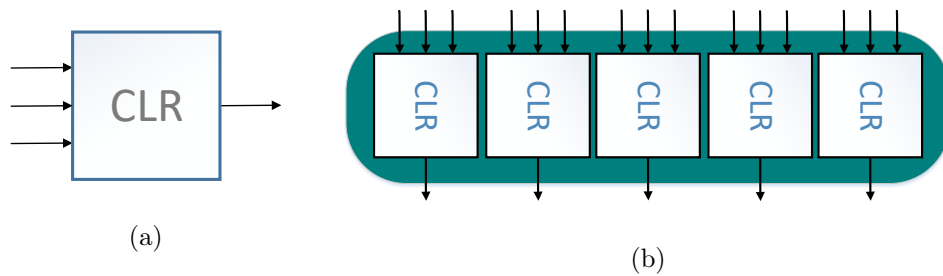


Figura 3.12: a) Representación en bloque de la CLR con tres entradas (célula del sistema), b) Representación del 1D-CA con semilla de 5 elementos

3.2.1. Circuito electrónico

Al igual que en el sistema anterior, el procedimiento para realizar el circuito es el siguiente:

1. Se forma un 1D-CA en forma de anillo de cinco celdas contiguas, la salida de cada celda se conecta a una entrada de *flip-flop*.
2. Las entradas de *preset* y *clear* de todos los *flip-flops* van conectadas mediante un dip switch a una fuente de alimentación de 5V, así podemos generar distintas semillas al activar y desactivar estas entradas en los distintos *flip-flops*. Para este trabajo, por generalidad, solo se activa el *preset* del tercer *flip-flop* con lo que el elemento central de la semilla es el único que se pone en estado alto antes de la primera iteración.
3. Se conecta una señal de reloj por la entrada CLK de los flip flops
4. Como último paso se genera la retroalimentación del circuito conectando las salidas de los *flip-flops* a las entradas de las CLR, esto permite que cada estado

que se va guardando en los *flip-flops* sea una nueva semilla para el instante de tiempo siguiente y así el sistema evoluciona de manera autónoma hasta que dejemos de proveerle la señal de reloj o en su defecto cortemos la alimentación del circuito.

Bajo el mismo criterio utilizado en el [capítulo 2](#) se procede a realizar el circuito de la CNN, en base al Autómata Celular de la sección anterior y compuertas lógicas reconfigurables.

Las conexiones entre las salidas Q_n de los *flip-flops* y las entradas x_1 , x_2 y x_3 de cada compuerta se realizan de la siguiente forma:

$$\begin{array}{l}
 Q_1 \left\{ \begin{array}{l} \text{Compuerta 5} \implies x_3 \\ \text{Compuerta 1} \implies x_2 \\ \text{Compuerta 2} \implies x_1 \end{array} \right. \\
 \\
 Q_2 \left\{ \begin{array}{l} \text{Compuerta 1} \implies x_3 \\ \text{Compuerta 2} \implies x_2 \\ \text{Compuerta 3} \implies x_1 \end{array} \right. \\
 \\
 Q_3 \left\{ \begin{array}{l} \text{Compuerta 2} \implies x_3 \\ \text{Compuerta 3} \implies x_2 \\ \text{Compuerta 4} \implies x_1 \end{array} \right. \\
 \\
 Q_4 \left\{ \begin{array}{l} \text{Compuerta 3} \implies x_3 \\ \text{Compuerta 4} \implies x_2 \\ \text{Compuerta 5} \implies x_1 \end{array} \right. \\
 \\
 Q_5 \left\{ \begin{array}{l} \text{Compuerta 4} \implies x_3 \\ \text{Compuerta 5} \implies x_2 \\ \text{Compuerta 1} \implies x_1 \end{array} \right.
 \end{array}$$

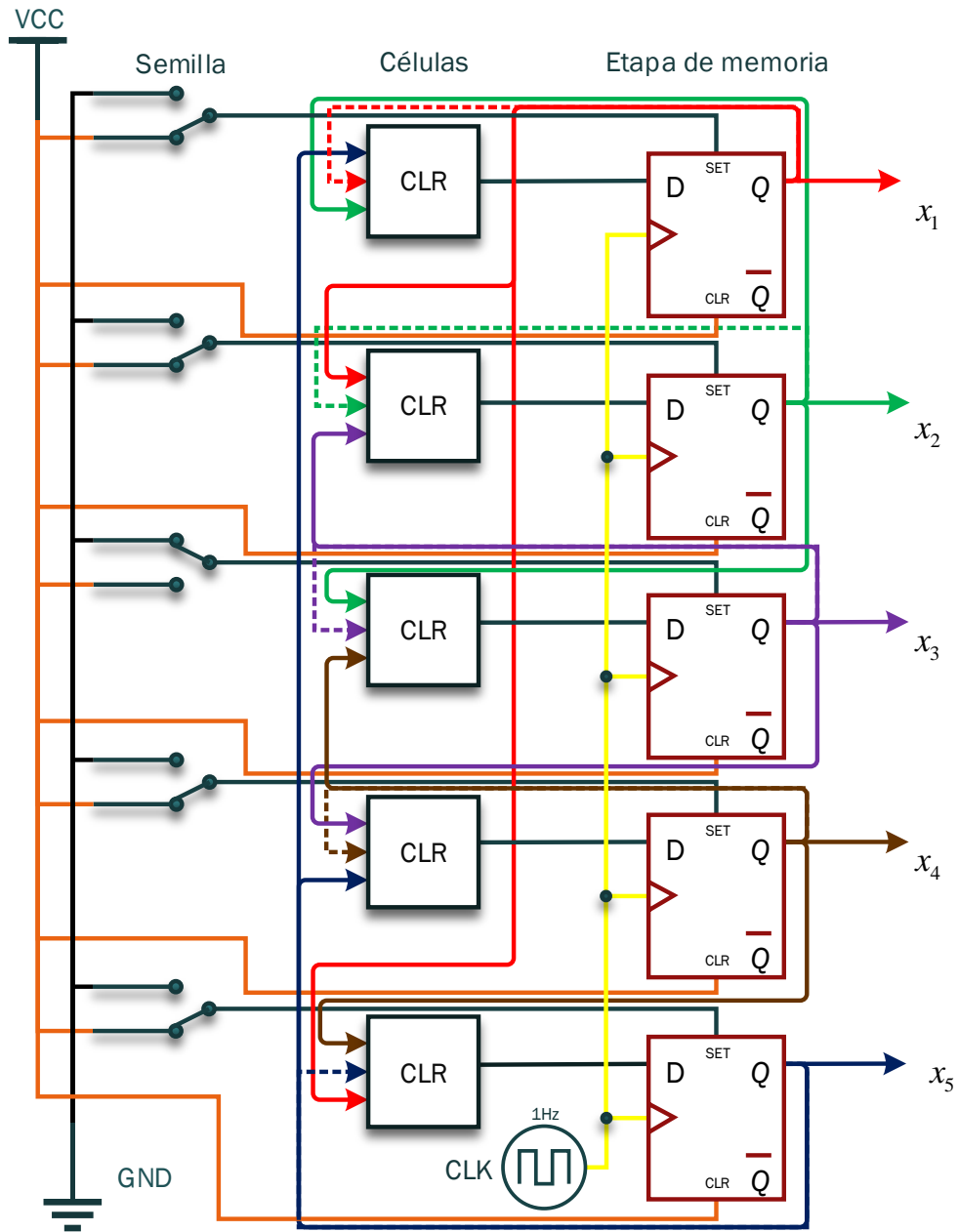


Figura 3.13: Circuito electrónico del sistema empleando la CLR de tres entradas

3.2.2. Resultados

La regla 254 (fig. 3.14) pertenece a la *clase I* de los 1D-CA, considerados de carácter *fijo*. Podemos observar en la figura 3.14a que debido al tamaño de nuestra semilla a partir de la cuarta iteración se mantiene un patrón constante, y de forma experimental en 3.14b podemos observar que este comportamiento se cumple de igual manera en nuestras barras de LED's. Por otra parte la regla 10 (fig. 3.15) pertenece a la *clase II* de carácter *periódico*, este comportamiento lo podemos corroborar de una manera muy marcada en las imágenes de la figura 3.15a, tanto en el diagrama espacio - tiempo obtenido de la simulación, como de forma experimental fig. 3.15b, observando que la respuesta arrojada por el sistema es satisfactoria.

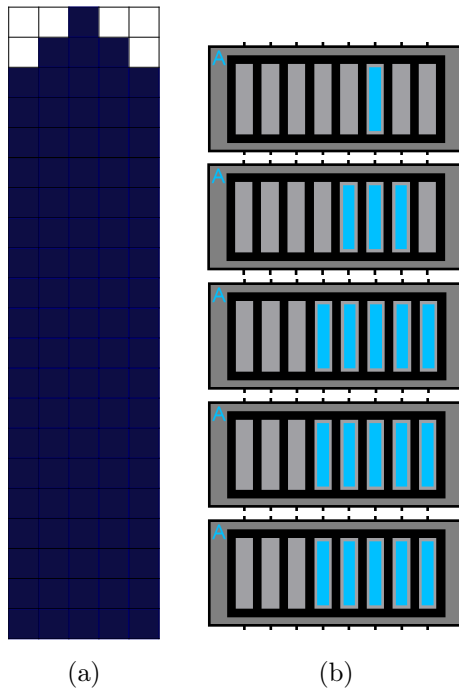


Figura 3.14: Evolución de la Regla 254 perteneciente a la clase I de los 1D-CA - *Fija*, a) Simulación matemática, b) Respuesta del sistema

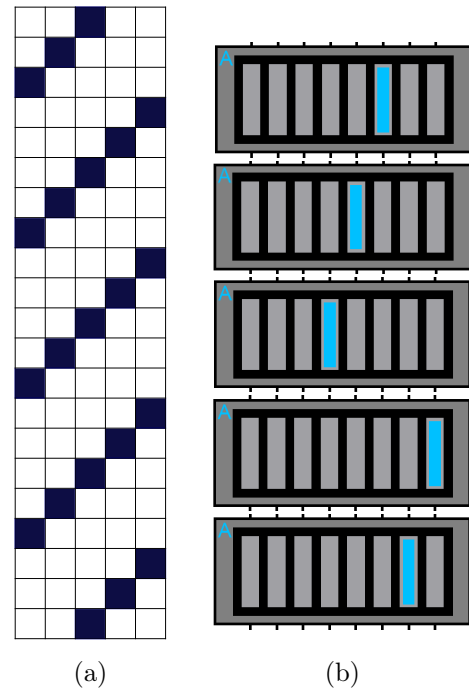


Figura 3.15: Evolución de la Regla 10 perteneciente a la clase II de los 1D-CA - *Periódica*, a) Simulación matemática, b) Respuesta del sistema

Las reglas 90 y 126 ambas pertenecen a la *clase III* de los 1D-CA, clasificadas de carácter *caótico*. En la primera (fig. 3.16a) podemos observar que se forma un patrón triangular que se repite a partir de la quinta iteración, esto debido a la longitud de nuestra semilla, ya que si se contara con una de mayor longitud, se podría observar como este patrón crece hasta formar el triángulo de Sierpinski como se puede observar en el diagrama espacio - tiempo de la figura 3.1a. Se puede observar en la figura 3.16b que el sistema genera de forma adecuada la evolución de la regla. Caso similar ocurre con la regla 126, solo que debido a su simetría, el fractal de evolución contiene más estados en activo y la ausencia de huecos hacen que no se puedan formar los triángulos como en el caso de la regla 90, sin embargo si extendemos la semilla, se puede observar que cada 2^n iteraciones aparece una línea horizontal de estados activos como si fuese una separación entre regiones [20]. En la figura 3.17 se puede observar que la respuesta del sistema cumple satisfactoriamente con los resultados esperados.

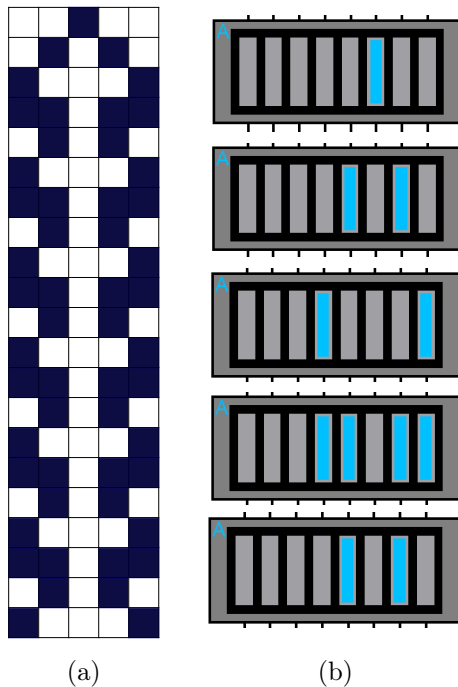


Figura 3.16: Evolución de la Regla 90 perteneciente a la clase III de los 1D-CA - *Caótica*, a) Simulación matemática, b) Respuesta del sistema

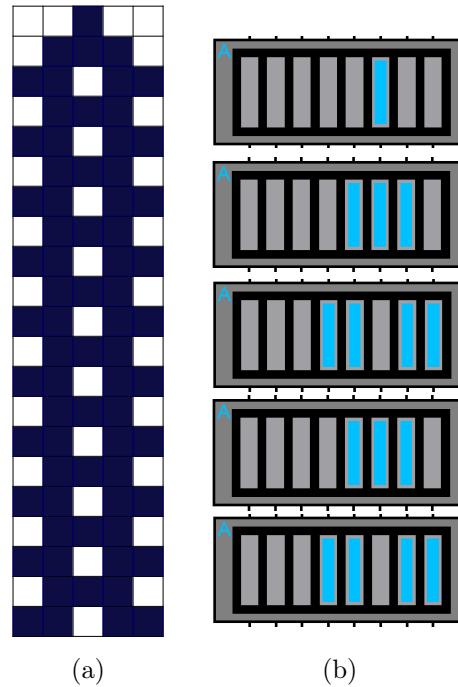


Figura 3.17: Evolución de la Regla 126 perteneciente a la clase III de los 1D-CA - *Caótica*, a) Simulación matemática, b) Respuesta del sistema

3.2. Diseño de la CNN

La regla 110 pertenece a la *clase IV*, la cual es considerada de carácter *complejo* debido a que su comportamiento no es completamente estable pero tampoco caótico. En esta clase de reglas se pueden observar que todos los patrones iniciales evolucionan en estructuras definidas que interactúa de formas complejas e interesantes, sin embargo para que un patrón se pueda repetir es necesario que transcurran muchas iteraciones. Esta regla junto al juego de la vida (*“Life”*), son hasta la fecha los únicos Autómatas Celulares comprobados con lo que se puede realizar cualquier cálculo computacional. La figura 3.18a es el diagrama espacio - tiempo de esta regla, donde podemos observar que los huecos forman figuras en forma de “naves espaciales” que se repiten cada cierto número de iteraciones; por otra parte nos podemos percatar de la ausencia de células activas hacia el lado derecho de la evolución, esto indica que la regla no tiene simetría reflectiva y por lo tanto se considera una regla *“ilegal”*. Desafortunadamente debido al tamaño de la semilla, a partir de la quinta iteración desaparece el patrón de evolución (fig. 3.18b), sin embargo se puede observar en la figura 3.18c que en estas cinco iteraciones el sistema responde de forma correcta.

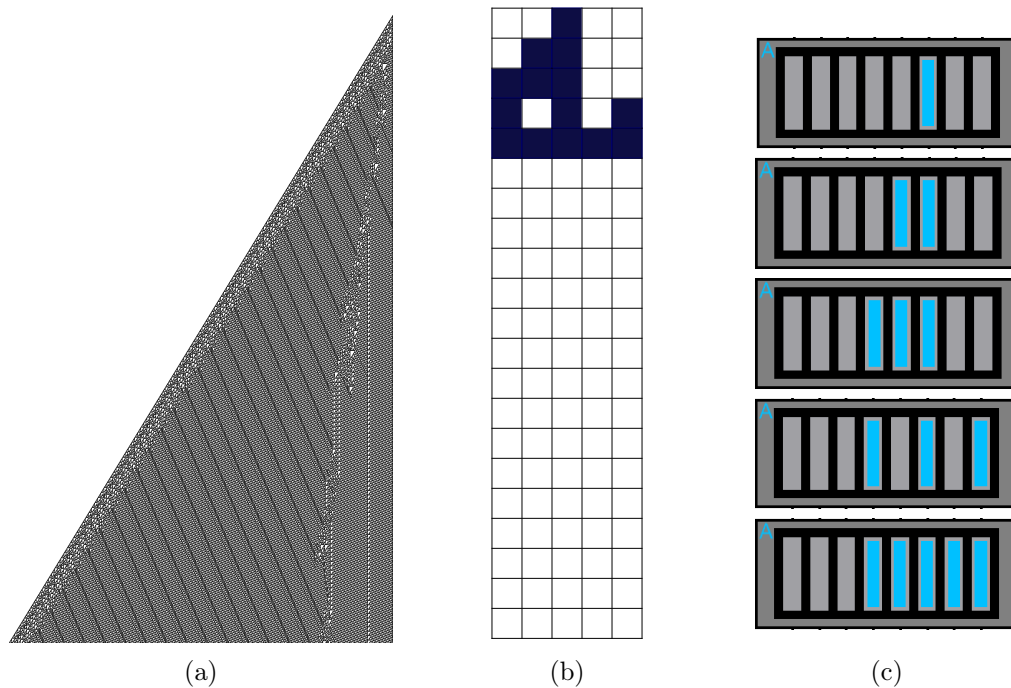


Figura 3.18: Evolución de la Regla 110 perteneciente a la clase IV de los 1D-CA - *Compleja*, a) Diagrama Espacio - Tiempo, b) Simulación matemática con semilla de cinco elementos, c) Respuesta del sistema

Conclusiones

El sistema fue capaz de reproducir y evolucionar reglas pertenecientes a las cuatro clases propuestas por Wolfram, de igual forma se pudo constatar su funcionamiento con reglas legales e ilegales lo que permite extendernos a las 256 posibilidades.

Dicho lo anterior cabe mencionar que dentro de las reglas que se muestran en los resultados, y algunas que no se reportan, la regla 41 perteneciente a la *clase IV - compleja* es la única que ha logrado hacer que el sistema falle al sintonizarla con los valores obtenidos de forma teórica..

La máquina de Wolfram fue de gran motivación para nuestro trabajo, ya que hasta el momento es el único sistema físico en el que se implementa un Autómata Celular y que además genera a su salida cualquiera de las 256 reglas haciendo uso de un microcontrolador. Sin embargo nuestra propuesta va más allá debido a que esta construida por componentes electrónicos básicos, esta no depende de una programación específica, si no que basta con cambiar los parámetros a la entrada del sistema para que esta modifique su comportamiento y al ser un sistema analógico lo podemos hacer en tiempo real. Por otra parte la desventaja que tiene nuestro sistema comparada con la máquina de Wolfram es que tenemos una gran limitante en el número de elementos que forman la semilla, ya que cada célula esta formada por un circuito.

Es importante mencionar que Chua también fue el primero en mostrar “caos” con un circuito relativamente sencillo de elementos pasivos al cual se le agrega un elemento no lineal denominado "*Diodo de Chua*" [6], Esto es de gran relevancia ya que en su modelo de CNN también utiliza elementos no lineales, solo que en el caso de las CNN's son fuentes controladas de voltaje, esto indica que puede existir una relación entre las CNN y los osciladores caóticos, por lo que nos sirve de inspiración para trabajos futuros.

En cuanto a lo que se refiere de compuertas lógicas reconfigurables, nos pudimos

dar cuenta que al obtener las ocho primeras funciones, estamos en la posibilidad de obtener las siguientes ocho por simetría, ya que estas son las inversas de la primeras; dicho de esta forma podemos obtener físicamente estas funciones antes de calcularlas si agregamos un inversor de voltaje a la salida del circuito, y así obtener los valores de las señales resultantes.

Aplicaciones

Desde el punto de vista computacional, el juego de la vida puede realizar cualquier cosa que un ordenador pueda realizar. A pesar de su simplicidad este ha sido comprobado por la ciencia de la computación como una herramienta de cálculo universal. De la misma forma la regla 110 al ser Turing Completa, teóricamente puede emular cualquier máquina de cómputo que exista hasta este momento, inclusive computadoras cuánticas, es importante recordar que esta regla es muy utilizada en el área de encriptación y seguridad informática

Es importante mencionar que algunos patrones evolutivos de CA tienden a formar Fractales, y estos en los años 70's y 80's estuvieron en boga de la comunidad artística debido a su belleza gráfica. En nuestros días nuevamente esta resurgiendo este interés, sin embargo ahora enfocados hacia la música gracias a los avances tecnológicos en cuanto a herramientas de producción musical. Y no solo eso, si no que también están incursionando en este ámbito otro tipo de modelos, como lo son los circuitos caóticos, un ejemplo de esto lo podemos apreciar en el trabajo "The Sound of Chaos" del Dr. Edgar Tristán

Las CNN se han convertido en los últimos años en una herramienta muy útil para simular el comportamiento de osciladores caóticos.

Trabajo a futuro

Uno de los objetivos a futuro es tratar de optimizar el sistema para que funcione a la perfección e implementarlo de forma física. Además es imperativo para nuestro trabajo estudiar diferentes modelos de sistemas dinámicos y caóticos, como lo son los osciladores ya que también tienen una estrecha relación con las CNN y los CA, y así tratar de llegar a proponer algún modelo interesante.

Bibliografía

- [1] Harold V MacIntosh, *One Dimensional Cellular Automata*. Frome, UK: Luniver Press, 2009.
- [2] Andrew Ilachinski, *Cellular Automata A Discrete Universe*. Danvers, USA: World Scientific Publishing Co. Pte. Ltd., 2001.
- [3] John E Hopcroft, Rejeev Motwani, and Jeffrey D Ullman, *Introduction to Automata Theory, Languages, and Computation*. Reading, USA: Addison-Wesley, 2001.
- [4] Burton H Voorhees, *Computational Analysis of Cellular Automata*. Danvers, USA: World Scientific Publishing Co. Pte. Ltd., 1996.
- [5] Stephen Wolfram, *A New Kind of Science*. Champaign, USA: Wolfram media, 2002.
- [6] Luigi Fortuna, Mattia Frasca, and Maria Gabriella Xibilia, *Chua's Circuit Implementations Yesterday, Today and Tomorrow*. Danvers, USA: World Scientific Publishing Co. Pte. Ltd., 2009.
- [7] David K Arrowsmith, and Place M Colin, *An introduction to dynamical systems*. New Yory, USA: Cambridge University Press, 1990.
- [8] Michael Brin and Garrett Stuck, *Introduction to Dynamical Systems*. New York, USA: Cambridge University Press, 2002.
- [9] Steven H Strogatz, *Nonlinear dynamics and chaos: with applications to physics, biology, chemistry, and engineering*. Reading, USA: Perseus Books Publishing, 1994.
- [10] Stephen Wolfram, "Statistical Mechanics of Cellular Automata", *Rev. Mod. Phys.*, vol. 55, no. 3, 1983.

- [11] Leon Ong Chua, and Lin Yang, “Cellular Neural Networks: Theory”, *IEEE Transactions on circuits and systems*, vol. 35, no. 10, pp. 1257-1272., 1988.
- [12] Leon Ong Chua, and Lin Yang, “Cellular Neural Networks: Applications”, *IEEE Transactions on circuits and systems*, vol. 35, no. 10, pp. 1273-1290., 1988.
- [13] Leon Ong Chua, “A Nonlinear Dynamics Perspective of Wolfram’s New Kind of Science. Part I: Threshold of complexity”, *International Journal of Bifurcation and Chaos*, vol. 12, no. 12, 2002.
- [14] Isaac Campos Cantón, “Desarrollo de Celdas Lógicas por medio del Espacio de Estados en un Sistema Bidimensional”, *Revista Mexicana de Física*, vol. 27, no. 2, pp. 117-124, 2011.
- [15] Luigi Fortuna, Mattia Frasca, Angelo Sarra Fiore, and Leon Ong Chua, “The Wolfram Machine”, *International Journal of Bifurcation and Chaos*, vol. 20, no. 12, pp. 3863-3917, 2010.
- [16] Jarkko Kari, “Theory of Cellular Automata: A Survey”, *Theoretical Computer Science*, vol. 334, no. 1-3, pp. 3-33, 2005.
- [17] Sudhakar Sahoo, Ipsita Mohanty, Garisha Chowdhary, and Arpit Panigrahi, “2-Variable Boolean Operation – Its use in Pattern Formation”, *arXiv preprint arXiv:1008.2530*, 2010.
- [18] Sun Jian-Cheng, “Chaotic behavior learning of Chua’s Circuit”, *Chinese Physics B*, vol. 21, no. 10, p. 100502, 2012.
- [19] Fernando Sancho Caparrini. (2016, Octubre) Autómatas Celulares. [Online]. <http://www.cs.us.es/~fsancho/?e=66>
- [20] Stephen Wolfram. (2003, Agosto) The Wolfram Atlas of Simple Programs. [Online]. <http://atlas.wolfram.com>
- [21] Marcela Mejía Carlos, “Encriptación por Sincronización en Autómatas Celulares”, Universidad Autónoma de San Luis Potosí, Facultad de Ciencias, Instituto de Investigación en Comunicación Óptica, San Luis Potosí , Tesis Doctoral 2001.
- [22] Luz María Ortiz Rivera, “Compuertas Dinámicas en Dos y Tres Dimensiones”, Universidad Autónoma de San Luis Potosí, Facultad de Ciencias, Instituto de Investigación en Comunicación Óptica, San Luis Potosí, Tesis de Maestría 2015.

- [23] Samuel Salinas Martínez, “Construcción de Compuertas Lógicas Dinámicas con Sistemas Lineales a Trozos y Caos Booleano”, Universidad Autónoma de San Luis Potosí, Facultad de Ciencias, San Luis Potosí, Tesis de Licenciatura 2010.
- [24] Isaac Campos Cantón, and Eric Campos Cantón, “Compuertas lógicas reconfigurables OR, AND & NOT”, Universidad Autónoma de San Luis Potosí, Facultad de Ciencias, San Luis Potosí, Artículo 2010.
- [25] Paulina A. León Hernandez , and Rogelio Basurto Flores, “Introducción a los Sistemas Dinámicos y Autómatas Celulares”, Escuela Superior de Cómputo, Instituto Politécnico Nacional, México, Artículo Verano de Investigación 2008.
- [26] David A Reyes Gómez, “Descripción y Aplicaciones de los Autómatas Celulares”, Universidad Autónoma de Puebla, Departamento de Aplicación de Microcomputadoras, Puebla, Artículo Verano de Investigación 2011.
- [27] Álvaro Álvarez Parrilla, and Aurora Espinoza Váldez, “Autómatas Celulares: La regla 150”, Universidad Autónoma de Baja California, Facultad de Ciencias, Mexicali, Artículo Semana Nacional de Investigación y Docencia en Matemáticas 2009.

Anexo

Simulación de Autómatas Celulares en MATLAB

Como parte del estudio de Autómatas Celulares Unidimensionales, se decidió realizar un programa para observar los patrones obtenidos por Wolfram. Originalmente se realizaron dos versiones del programa el cual estaba hecho solo para observar la regla 182. En la primera se contaba con una semilla inicial donde la única celda activa se encuentra justamente a la mitad de esta. En la segunda versión se insertaban semillas cambiando una línea de código la cual contiene una función que inyecta 1's de forma aleatoria en un vector y la cantidad de estos se regula con un porcentaje de probabilidad. En el programa [A.2](#) también se obtiene una señal formada por la concentración de 1's en cada iteración del 1D-CA, y a esta señal se le extraen algunas propiedades como su magnitud, fase y transformada de Fourier.

Programación en MATLAB de la regla 182 con semilla de 512 elementos, célula central activa

Código

Programa A.1: Evolución de la regla 182, solo la célula central está activa

```
1 % Detección de la concentración de 1s en un Automata Celular de 512 x 512
2 % Primera Etapa
3 % Crecimiento del automata usando la regla 182 con una semilla "256"
4 clear all; close all; clc %cierra procesos, limpia variables y pantalla
5 %seed=de2bi(32768,31); %Semilla, decimal a binario, con "n" columnas
6
7 x=512; %tamaño de las filas de la red
8 y=480; %tamaño de las columnas de la red
9 seed(1,512)=0; %tamaño de la semilla (1 * 512)
```

```

10 seed(256)=1;
11 a=seed;           %asigna en "a" el valor de la semilla
12 b=0;             %inicializa una variable "b" en 0
13 automata=zeros(x); %Red de ceros para el automata
14
15 for(j=1:y)       % desde la fila 1 a la x ejecuta:
16     automata(j,:)=a; %Se coloca la semilla al inicio del automata
17     b(1)=or(or(and(a(x),not(a(1))),and(not(a(1)),a(2))), or(and(a(x),a(2)), and(and(
18         not(a(x)), a(1)), not(a(2)))); % aplica la regla para b0
19     b(x)=or(or(and(a(x-1),not(a(x))),and(not(a(x)),a(1))), or(and(a(x-1),a(1)), and(
20         and(not(a(x-1)), a(x)), not(a(1)))); % aplica la regla para b7
21     for(i=2:x-1) % aplica la regla para los demas
22         b(i)= or(or(and(a(i-1),not(a(i))),and(not(a(i)),a(i+1))), or(and(a(i-1),a(i+1)
23             ), and(and(not(a(i-1)), a(i)), not(a(i+1))));
24     end
25     a=b; % reemplaza el valor de a con lo que ahora tiene b
26 end

```

Resultados

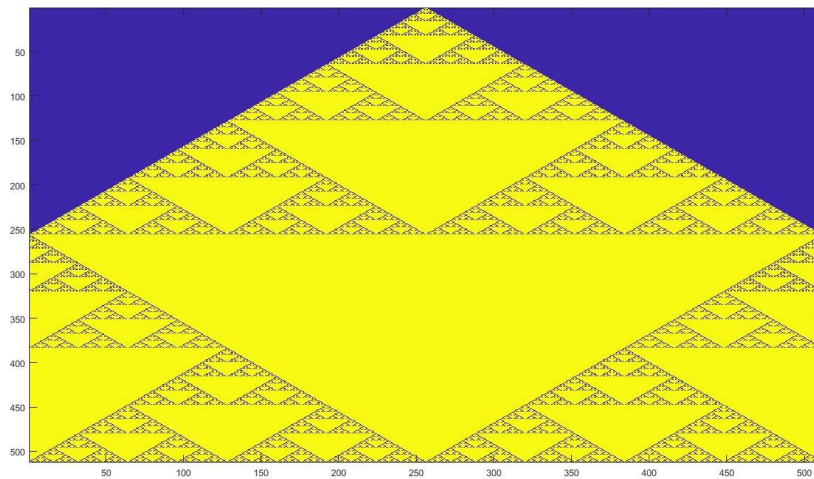


Figura A.1: Evolución de un Autómata Celular mediante la regla 182, en 480 iteraciones

Programación en MATLAB de la regla 182 con semilla de 512 elementos, condiciones iniciales aleatorias

Código

Programa A.2: Evolución de la regla 182 con condiciones iniciales aleatorias.

```

1 % Detección de la concentración de 1s en un Automata Celular de 512 x 512
2 % Total de elemento de la red : 262,144
3 % Crecimiento del automata usando la regla 182 con una semilla "256"
4 clear all; close all; clc %cierra procesos, limpia variables y pantalla
5 x=512; %tamaño de las filas de la red
6 y=512; %tamaño de las columnas de la red
7 %seed=rand(1,512)>.99; %Vector de 1's aleatorio, incrementar el numero para meos 1's y
   viceversa
8 %xlswrite('seed.xlsx',seed, 'Hoja1','A21'); %Escribe la semilla en excell
9 seed=xlsread('seed.xlsx', 'Hoja1','A1:SR1'); %Lee la semilla desde excell
10 a=seed; %asigna en "a" el valor de la semilla
11 b=0; %inicializa una variable "b" en 0
12 automata=zeros(x); %Red de ceros para el automata
13 unosM=nnz(automata); %cantidad de unos en el automata
14
15 for(j=1:y) % desde la fila 1 a la x ejecuta:
16     automata(j,:)=a; %Se coloca la semilla al inicio del automata
17     b(1)=or(or(and(a(x),not(a(1))),and(not(a(1)),a(2))), or(and(a(x),a(2)), and(and(
   not(a(x)), a(1)), not(a(2)))); % aplica la regla para b0
18     b(x)=or(or(and(a(x-1),not(a(x))),and(not(a(x)),a(1))), or(and(a(x-1),a(1)), and(
   and(not(a(x-1)), a(x)), not(a(1)))); % aplica la regla para b7
19     for(i=2:x-1) % aplica la regla para los demas
20         b(i)= or(or(and(a(i-1),not(a(i))),and(not(a(i)),a(i+1))), or(and(a(i-1),a(i+1)
   ), and(and(not(a(i-1)), a(i)), not(a(i+1))));
21     end
22     a=b; % reemplaza el valor de a con lo que ahora tiene b
23 end
24 s=sum(automata(1,:)); %concentración de 1's de la semilla
25 v=sum(automata,2); %vector concentración de 1's por renglon (señal)
26 Mv=abs(v); %Magnitu de la señal
27 Fv=angle(v); %Fase de la señal
28 TRFv=fft(v); %Transformada rapida de Fourier de la señal
29 MFv=abs(TRFv); %Magnitud de FFT de la señal
30 FFv=angle(TRFv); %Fase de la FFT de la señal
31 %[THAv, THDv]=haart(v); %Transformada de Haar a 5 niveles
32 %MHv=abs(THAv); %Magnitud de la matriz de aproximaciones de Haar de la señal
33 %FHv=angle(THAv); %Fase de la matriz de aproximaciones de Haar de la señal
34 frec=ifft(TRFv); %Fourier inversa

```

```
35 %rec=ihaart(MHv,THDv); %Haar inversa
36 E=Energia(v); %Energia de la señal
37 figure(1)
38 imagesc(automata)
39 figure(2)
40 subplot(3,2,1); plot(v)
41 subplot(3,2,2); plot(seed)
42 subplot(3,2,3); plot(MFv)
43 subplot(3,2,4); plot(FFv)
44 %subplot(5,2,5); plot(MHv)
45 %subplot(5,2,6); plot(FHv)
46 subplot(3,2,5); plot(frec)
47 %subplot(5,2,8); plot(xrec)
48 %subplot(3,2,6); plot(E)
49 hold on
50 %subplot(5,2,10); plot(MFv)
51 %subplot(5,2,10); plot(MHv)
52 hold off
```

Resultados

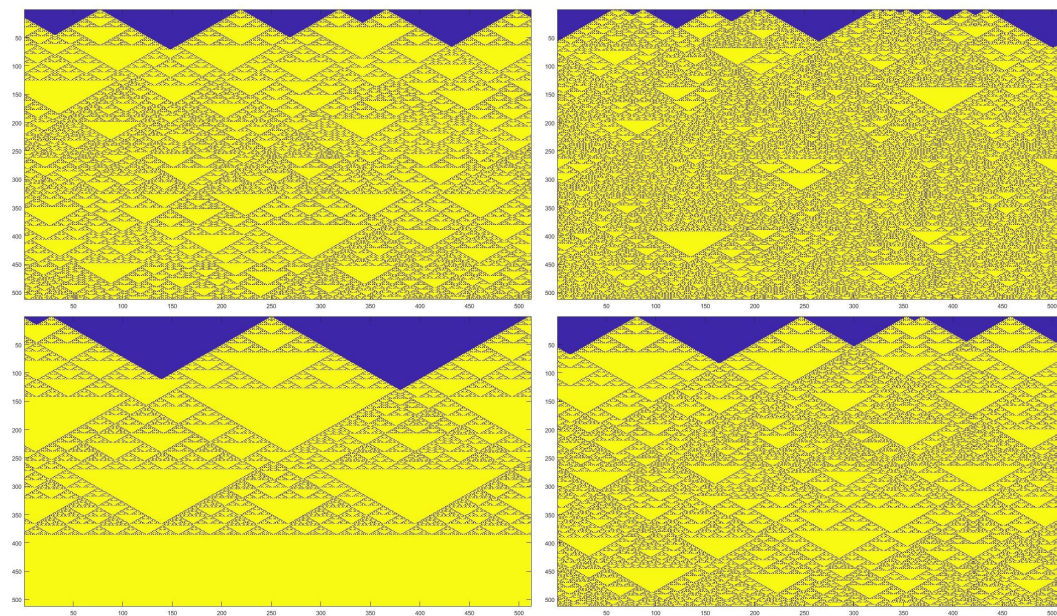


Figura A.2: Evolución de Autómatas Celulares mediante la regla 182, con semillas aleatorias de 512 elementos en 480 pasos de tiempo

Índice de figuras

I.1.	Red Neuronal con dos capas ocultas	2
I.2.	Elementos de una máquina universal de Turing	3
1.1.	Vecindario de un Autómata Celular Elemental	12
1.2.	Clasificación de Wolfram de los Autómatas Celulares	13
1.3.	Ejemplos de Autómatas Celulares Unidimensionales	14
1.4.	Diagramas de espacio - tiempo de la regla 110	15
1.5.	Vecindarios más usados en los Autómatas Celulares Bidimensionales	16
1.6.	Patrones de tipo Fractal obtenidos mediante la regla 14	18
1.7.	CNN bidimensional básica de $i \times j$ elementos	19
1.8.	Vecindarios de CNN's	20
1.9.	Anillo y célula de una CNN	21
1.10.	Cubo Booleano	23
2.1.	Diagrama de bloques de una compuerta lógica reconfigurable de dos entradas.	29
2.2.	Divisor de voltaje	30
2.3.	Primera parte del circuito: Divisores de voltaje y amplificador sumador	31
2.4.	Segunda parte del Circuito: Elemento no lineal (comparador de ventana)	32
2.5.	Circuito de la compuerta lógica reconfigurable de dos entradas	33
2.6.	Compuerta lógica <i>AND</i>	35
2.7.	Compuerta lógica <i>OR</i>	36
2.8.	Compuerta lógica <i>XOR</i>	37
2.9.	Representación de la CLR de dos entradas y el 1D-CA con semilla de 5 elementos	38
2.10.	Circuito electrónico del sistema empleando la <i>CLR</i> de dos entradas	40
2.11.	Evolución de la Regla 160	41
2.12.	Evolución de la Regla 250	42
2.13.	Evolución de la Regla 90	42

3.1. Reglas de evolución (arriba) y diagramas <i>espacio-tiempo</i> (abajo) de Autómatas Celulares Unidimensionales	43
3.2. Topología básica de la CNN diseñada	44
3.3. Diagrama de bloques de una compuerta lógica reconfigurable de tres entradas.	46
3.4. Primera parte del circuito: Divisores de voltaje, inversores de polaridad y amplificador sumador	47
3.5. Comparador de ventana con amplificadores operacionales TL081 y transistor	48
3.6. Circuito de la compuerta lógica reconfigurable de tres entradas	50
3.7. Regla 10	51
3.8. Regla 90	53
3.9. Regla 110	54
3.10. Regla 126	55
3.11. Regla 254	56
3.12. Representación de la CLR de tres entradas y el 1D-CA con semilla de 5 elementos	57
3.13. Circuito electrónico del sistema empleando la <i>CLR</i> de tres entradas	59
3.14. Evolución de la Regla 254	60
3.15. Evolución de la Regla 10	60
3.16. Evolución de la Regla 90	61
3.17. Evolución de la Regla 126	61
3.18. Evolución de la Regla 110	62
A.1. Evolución de un Autómata Celular mediante la regla 182	XVI
A.2. Evolución de Autómatas Celulares mediante la regla 182, con semillas aleatorias	XVIII

Índice de tablas

1.1.	Tabla de verdad para la función $y_i = N(u_{i-1}, u, u_{i+1})$	22
2.1.	Tabla de verdad de las funciones que representan a las compuertas lógicas básicas	26
2.2.	Tabla de verdad para obtener las ecuaciones que se generan con las distintas combinaciones en una CLR de dos entradas	29
2.3.	Valores para los parámetros de entrada del sistema	34
2.4.	Resultados de las ecuaciones del sistema al sintonizar la compuerta lógica <i>AND</i>	34
2.5.	Resultados de las ecuaciones del sistema al sintonizar la compuerta lógica <i>OR</i>	35
2.6.	Resultados de las ecuaciones del sistema al sintonizar la compuerta lógica <i>XOR</i>	36
2.7.	Tabla de verdad de un Flip-Flop tipo D	38
3.1.	Tabla de verdad para las reglas 10, 90,110, 126 y 254	45
3.2.	Tabla de verdad para obtener las ecuaciones que se generan con las distintas combinaciones en una CLR de tres entradas	46
3.3.	Valores para los parámetros de entrada del sistema	49
3.4.	Tabla para obtener los valores que sintonizan la regla 10	51
3.5.	Tabla de verdad para obtener los valores que sintonizan la regla 90	52
3.6.	Tabla de verdad para obtener los valores que sintonizan la regla 110	54
3.7.	Tabla de verdad para obtener los valores que sintonizan la regla 126	55
3.8.	Tabla de verdad para obtener los valores que sintonizan la regla 254	56