



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

**Diseño e implementación de un sistema basado en hardware reconfigurable
como herramienta de estudio de la sincronización de circuitos caóticos para el
Instituto Técnico Superior Carlos Cisneros**

ROLANDO BOLÍVAR DONOSO REAL

**Trabajo de Titulación modalidad Proyectos de Investigación y Desarrollo, presentado ante
el Instituto de Posgrado y Educación Continua de la ESPOCH, como requisito parcial para
la obtención del grado de**

**MAGÍSTER EN MATEMÁTICA MENCIÓN MODELACIÓN Y
DOCENCIA**

RIOBAMBA - ECUADOR

Noviembre 2022

©2022, Rolando Bolívar Donoso Real.

Se autoriza la reproducción total o parcial, con fines académicos, por cualquier medio o procedimiento, incluyendo la cita bibliográfica del documento, siempre y cuando se reconozca el Derecho de Autor



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

EL TRIBUNAL DE TRABAJO DE TITULACIÓN CERTIFICA QUE:

El Trabajo de Titulación modalidad Proyectos de Investigación y Desarrollo, titulado Diseño e implementación de un sistema basado en hardware reconfigurable como herramienta de estudio de la sincronización de circuitos caóticos para el Instituto Técnico Superior Carlos Cisneros, de responsabilidad del señor Rolando Bolívar Donoso Real ha sido prolijamente revisado y se autoriza su presentación.

**DRA. SILVIA MARIANA HARO RIVERA; MAG.
PRESIDENTE**



Firmado electrónicamente por:
**SILVIA
MARIANA HARO**

**ING. LEONARDO FABIÁN RENTERÍA
BUSTAMANTE; PH. D.
DIRECTOR**



Firmado electrónicamente por:
**LEONARDO FABIAN
RENERIA
BUSTAMANTE**

**DR. KLEVER HERNAN TORRES RODRIGUEZ;
MAG.
MIEMBRO**



Firmado electrónicamente por:
**KLEVER HERNAN
TORRES
RODRIGUEZ**

**LIC. JOSÉ ALEJANDRO SÁNCHEZ SÁNCHEZ;
MAG.
MIEMBRO**

**JOSE
ALEJANDRO
SANCHEZ
SANCHEZ** Firmado
digitalmente por
JOSE ALEJANDRO
SANCHEZ SANCHEZ
Fecha: 2022.11.08
08:50:12 -05'00'

Riobamba, noviembre 2022

DERECHOS INTELECTUALES

Yo, Rolando Bolívar Donoso Real, declaro que soy responsable de las ideas, doctrinas y resultados expuestos en el **Trabajo de Titulación modalidad Proyectos de Investigación y Desarrollo**, y que el patrimonio intelectual generado por la misma pertenece exclusivamente a la Escuela Superior Politécnica de Chimborazo.

ROLANDO Bolívar Donoso Real
BOLIVAR
DONOSO
REAL

Firmado digitalmente por
ROLANDO BOLIVAR DONOSO REAL
Fecha: 2022.11.08 08:18:14 -05'00'

Rolando Bolívar Donoso Real

No. Cédula 0604098541

DECLARACIÓN DE AUTENTICIDAD

Yo, Rolando Bolívar Donoso Real, declaro que el presente proyecto de investigación es de mi autoría y que los resultados de este son auténticos y originales. Los textos constantes en el documento que provienen de otras fuentes están debidamente citados y referenciados.

Como autor, asumo la responsabilidad legal y académica de los contenidos de este Trabajo de Titulación de Maestría.

ROLANDO
BOLIVAR
DONOSO
REAL



Firmado
digitalmente por
ROLANDO BOLIVAR
DONOSO REAL
Fecha: 2022.11.08
08:19:22 -05'00'

Rolando Bolívar Donoso Real

No. Cédula 0604098541

AGRADECIMIENTO

Agradezco a la Escuela Superior Politécnica de Chimborazo, a la Facultad de Mecánica y de manera especial a la Escuela de Ingeniería Mecánica, la misma que me enseñó a superar dificultades para conseguir las metas deseadas. Mencionar también al Instituto de Posgrado y Educación Continua quienes siguen formando profesionales de un alto nivel académico reconocidos a nivel nacional. Al Instituto Técnico Superior Carlos Cisneros por haber permitido el desarrollo de la tesis.

De la misma manera mi gratitud al Dr. Leonardo Rentería por haber confiado en mi persona, por la paciencia y dirección en el proceso de elaboración de la tesis. Al Dr. Klever Rodríguez por compartir sus conocimientos en esta investigación, al Mag. José Sánchez por el apoyo incondicional.

Un agradecimiento a mis padres, a mis suegros, a mis hermanas y en especial a mi hermosa esposa Angela Mayacela, quien gracias a su paciencia es un apoyo para seguir cumpliendo sueños. Mencionar a mis princesas Sofía y Emma Donoso, la una me cuida desde el cielo y la otra me brinda sus alegrías aquí en la tierra.

TABLA DE CONTENIDO

RESUMEN.....	xiv
SUMMARY	xv

CAPÍTULO I

1	INTRODUCCIÓN	1
1.1	Formulación del problema.	1
1.2	Preguntas directrices o específicas de la investigación.	1
1.3	Justificación de la Investigación:	1
1.3.1	Justificación Teórica:	1
1.3.2	Justificación practica:	2
1.3.3	Justificación Metodológica	3
1.3.4	Justificación Social.....	3
1.4	OBJETIVOS	3
1.4.1	Objetivo General	3
1.4.2	Objetivos específicos	3
1.5	Planteamiento de la hipótesis	4
1.5.1	Hipótesis General	4
1.5.2	Hipótesis específicas	4

CAPÍTULO II

2	MARCO TEÓRICO.....	5
2.1	Situación problemática	5
2.2	Antecedentes del problema:	5
2.2.1	Modelado matemático de sistemas dinámicos en epidemiología.....	5
2.2.2	Sistema open hardware y open source aplicados a la enseñanza de la electrónica.	6
2.2.3	Spontaneous Synchronization in Two Mutually Coupled Memristor-Based Chua's Circuits: Numerical Investigations.....	6
2.2.4	Diseño e implementación sobre hardware reconfigurable de una arquitectura para la emulación en tiempo real de redes neuronales celulares.....	6

2.2.5	Programación VHDL de algoritmos de codificación para dispositivos de hardware reconfigurable	7
2.2.6	Implementación de una red neuronal en FPGA para modelado de sistemas.	7
2.3	Sistemas dinámicos	8
2.4	Clasificación de los sistemas dinámicos.	9
2.4.1	Sistemas con parámetros concentrados y con parámetros distribuidos.....	9
2.4.2	Sistemas lineales y no lineales	9
2.4.3	Sistemas variantes en el tiempo e invariantes en el tiempo.	10
2.4.4	Sistemas continuos y sistemas discretos:	10
2.4.5	Respuesta del sistema:.....	11
2.5	Sistemas dinámicos caóticos.	11
2.6	Regresión lineal simple	12
2.7	Método de Euler	14
2.8	Octave	16
2.9	Circuito de Chua	17
2.10	Sincronización del circuito de Chua.....	18
2.11	Hardware reconfigurable.....	19
2.12	Hardware reconfigurable libre:	19
2.13	FPGAS	20
2.14	Estructura de las FPGA	20
2.15	Representación de un punto fijo.....	21
2.15.1	Representación de números fraccionarios en DSP de bajo costo.....	21
2.15.2	El formato Q.....	22
2.15.3	Elegir la posición del punto binario	22
2.16	Marco conceptual:	23

CAPÍTULO III

3	METODOLOGÍA DE LA INVESTIGACIÓN.....	25
3.1	Metodología:	25
3.1.1	Tipo y diseño de investigación:.....	25
3.1.2	Cuasi experimental.....	25
3.1.3	Inductivo-Deductivo	25
3.1.4	Correlacional	25

3.2	Métodos de investigación.....	26
3.3	Enfoque de la investigación:	26
3.4	Alcance de la investigación:.....	26
3.5	Técnicas de la investigación:	27
3.5.1	Investigación bibliográfica:.....	27
3.5.2	Estudios correlacionales:.....	27
3.5.3	Estudios causales comparativos a partir de teorías fundamentadas	27
3.6	Tamaño de la muestra	28
3.7	Identificación de variables	28

CAPÍTULO IV

4	RESULTADOS Y DISCUSIÓN	30
4.1	ANÁLISIS DE RESULTADOS	30

CAPÍTULO V

5	PROPUESTA.....	34
5.1	SINCRONIZACIÓN DEL CIRCUITO DE CHUA.....	34
5.2	Planteamiento del modelo	34
5.2.1	Modelo de un circuito de Chua.	34
5.2.2	Modelo de sincronización de dos circuitos de Chua.	35
5.2.3	Modelo de sincronización de n circuitos de Chua.....	36
5.3	Implementación del circuito de Chua en Octave	38
5.4	Implementación del circuito de Chua en la FPGA.....	41
5.5	Análisis del modelo para n circuitos de Chua usando FPGA.....	43
	CONCLUSIONES	47
	RECOMENDACIONES	48

GLOSARIO

BIBLIOGRAFÍA

ANEXOS

ÍNDICE DE TABLAS

Tabla 1-3. Operacionalización de variables.....	29
Tabla 1-4. Coeficiente de correlación y modelo lineal de las variables de estado de la FPGA (xx, yy, zz) y del software de simulación (x, y, z).	33

ÍNDICE DE GRÁFICOS

Gráfico 1-4: Diagrama de cajas de las variables de estado, x , y , z (Octave) & xx , yy , zz . (FPGA) .	31
Gráfico 2-4: Gráfica dispersión de la relación lineal $xx1$ vs $x1$	32
Gráfico 1-5: Variables de estado en función del tiempo.....	39
Gráfico 2-5. Relación de las variables de estado x , y , z	40
Gráfico 3-5. Variables de estado de tres nodos acoplados en función del tiempo.....	41
Gráfico 4-5. RTL del sistema implementado en la FPGA para un circuito de Chua.....	42
Gráfico 5-5. Variables de estado en función del tiempo obtenidas en la FPGA	43
Gráfico 6-5. RTL del sistema implementado en la FPGA para tres circuitos de Chua. $\text{Alpha}=15$, $\text{beta}= 28$, $m1= -1.143$, $m0= -0.714$, $x1= 0.7$, $x2 = 0$, $x3 = 0$	44
Gráfico 7-5. Variables de estado $x1$, $x2$, $x3$ con respecto al tiempo	44
Gráfico 8-5. Relación entre variables de estado calculadas con la FPGA (xx , yy , zz) y las calculadas con Octave (x , y , z).....	46

ÍNDICE DE FIGURAS

Figura 1-2. Regresión lineal volumen vs costos	12
Figura 2-2: Posibles diagramas de dispersión.	13
Figura 3-2: Representación de los mínimos cuadrados	14
Figura 4-2: Curva Solución	15
Figura 5-2: Circuito de Chua.	17
Figura 6-2. Estructura interna de un FPGA	21
Figura 1-4: Recursos utilizados por la FPGA Spartan 3E luego de la síntesis e implementación de tres circuitos de Chua.	33
Figuras 1-5. Circuito Caótico de Chua	34
Figura 2-5. Dos circuitos de Chua acoplados a través de la variable x	36
Figura 3-5. N circuitos de Chua acoplados a través de la variable x	37
Figura 4-5: Spartan 3E FPGA	42

ÍNDICE DE ANEXOS

Anexo A: Código fuente para el análisis de un circuito de Chua en el programa Octave

Anexo B : Código fuente para el análisis de la sincronización de tres circuitos de Chua en el programa Octave

Anexo C : Modulo verilog del divisor de frecuencia, fdiv.v

Anexo D : Modulo verilog de un circuito de chua , chua.v

Anexo E : Modulo verilog de la unidad bajo prueba de un circuito de chua, uutChua.v

Anexos F: Módulo Verilog de un circuito de Chua con entradas de sincronización, chuaN.v

Anexo G : Modulo VERILOG de la unidad bajo prueba de tres circuitos de Chua, uutChuaN.v

Anexo H : Elementos y equipos utilizados en el sistema

RESUMEN

El objetivo del presente trabajo es diseñar e implementar un sistema basado en hardware reconfigurable como herramienta de estudio de la sincronización de circuitos caóticos para el Instituto Técnico Superior Carlos Cisneros. La investigación es de tipo cuasi experimental, inductivo-deductivo y correlacional. La investigación está basada en datos cualitativos a más de hipótesis y variables. Para el desarrollo del trabajo se estableció modelos matemáticos para uno, dos y n circuitos de Chua, estos modelos fueron llevados a lenguaje de programación para implementarlos en el software de simulación numérica Octave. Después se realizó un script basado en el método de resolución de ecuaciones diferenciales de Euler para implementarlo en la tarjeta FPGA spartan 3E usando un lenguaje verilog, este sistema a su vez cumple la función de hardware. Este sistema resolvió las ecuaciones de estado satisfactoriamente y los datos obtenidos fueron comparados estadísticamente con los datos arrojados por el software de simulación numérica; los resultados del estudio mostraron que no existió una diferencia significativa entre las variables de estado del sistema implementado y del software, obteniendo un coeficiente de correlación de Pearson aproximado a 1, por lo que se puede concluir que los dos sistemas son funcionales para la resolución de circuitos de Chua. Es recomendable encontrar una aplicación práctica en los diferentes campos de estudio de los sistemas caóticos.

Palabras clave: <MODELO MATEMÁTICO>, <SOFTWARE>, < SINCRONIZACIÓN >, < SISTEMA >, < PROGRAMA >, < HARDWARE >, < CIRCUITO >, <SIMULACIÓN>.



13-10-2022

0144-DBRA-UPT-IPEC-2022

SUMMARY

The purpose of this work is to design and implement a system based on reconfigurable hardware as a study tool for the synchronization of chaotic circuits for the Carlos Cisneros Higher Technical Institute. The research is of a quasi-experimental, inductive-deductive, and correlational type. The research is based on qualitative data in addition to hypotheses and variables. For the development of the work, mathematical models were established for one, two and n Chua circuits, these models were converted into a programming language to be implemented in the Octave numerical simulation software. Afterwards, a script based on Euler's differential equation resolution method was created to be implemented on the spartan 3E FPGA card using a Verilog language, this system in turn fulfills the hardware function. This system solved the equations of state satisfactorily and the data obtained were statistically compared with the data produced by the numerical simulation software; The results of the study showed that there was no significant difference between the state variables of the implemented system and the software, obtaining a Pearson correlation coefficient of approximately 1, so it can be concluded that the two systems are functional for solving Chua circuits. It is advisable to find a practical application in the different fields of study of chaotic systems.

Keywords: <MATHEMATICAL MODEL>, <SOFTWARE>, <SYNCHRONIZATION>, <SYSTEM>, <PROGRAM>, <HARDWARE>, <CIRCUIT>, <SIMULATION>.

CAPÍTULO I

1 INTRODUCCIÓN

La teoría del caos es una rama de las matemáticas que se ocupa de los sistemas que parecen ser ordenados (deterministas) pero, de hecho, albergan comportamientos caóticos. También se ocupa de los sistemas que parecen ser caóticos, pero, habitualmente, tienen orden subyacente. Es así que, nos permite comprender los fenómenos de la naturaleza y los patrones de comportamiento que obedece. Más allá de esto, también aparece como una valiosa herramienta para comprender el comportamiento humano, los fenómenos sociales, económicos, la evolución de la tecnología y la actividad industrial.

Los principios de la teoría del caos se han utilizado con éxito para describir y explicar diversos fenómenos naturales y fenómenos artificiales. Por citar algunos ejemplos, predecir decomisos, el comportamiento de los mercados financieros, modelar producción de sistemas de fabricación, boletines meteorológicos, creación de Fractales, entre otros.

1.1 Formulación del problema.

¿Es posible utilizar una herramienta de simulación basada en hardware reconfigurable para el estudio de la sincronización de circuitos dinámicos caóticos?

1.2 Preguntas directrices o específicas de la investigación.

- a) ¿Se puede estudiar fenómenos físicos mediante la simulación de circuitos caóticos?
- b) ¿Es posible utilizar un hardware reconfigurable como herramienta para el análisis de la sincronización de circuitos dinámicos caóticos?
- c) ¿Se puede verificar estadísticamente que utilizando el hardware reconfigurable como herramienta se obtendrán iguales resultados que los obtenidos en el software de análisis numérico?

1.3 Justificación de la Investigación:

1.3.1 Justificación Teórica:

A través del modelado y la simulación de redes complejas se alcanzan grandes resultados que de otra forma no se pudieran obtener. El modelado y simulación son de gran importancia al momento de plantear o proponer soluciones que impliquen investigaciones o comprobaciones puesto que permiten la reducción del tiempo y consecuentemente de costos. Con el apoyo de este tipo de herramientas se implementan mecanismos de predicción y análisis de rendimientos, funcionamiento bajo distintos escenarios, variaciones y análisis de sensibilidad sobre las distintas formas del comportamiento del sistema. En este contexto, existen diversas herramientas de software abierto y propietario que se utilizan en el estudio de sistemas dinámicos no lineales tales como: Matlab, Mathematica, Python, Octave, entre otras.

Por lo tanto, en la carrera de tecnología superior electrónica del IST Carlos Cisneros manejan varias de estas herramientas de software abierto para el análisis de circuitos, redes neuronales, sistemas de ultrasonido, control de módulos funcionales, redes de inteligencia artificial, sistemas de gestión de eficiencia energética y muchos temas más.

Estos temas de análisis están basados en circuitos y modelos matemáticos los cuales deben ser analizados a través de software o hardware debido a su grado de complejidad; el hardware es el más cotizado pues no necesita de una gran cantidad de memoria para analizar los resultados, a más que su tiempo de análisis es menor en comparación a un software. En consecuencia, el diseño y la implementación de un hardware reconfigurable como herramienta ayudaría a los estudiantes de la carrera de electrónica y telecomunicaciones a realizar el análisis de fenómenos físicos de un modo práctico y no solo en un modo teórico.

1.3.2 Justificación práctica:

Si bien es cierto, las computadoras son ampliamente utilizadas en la ingeniería y la ciencia, la computación digital actual es muy pasiva y los algoritmos basados en aproximaciones no pueden proporcionar todas las posibles respuestas complicadas que existen en los sistemas dinámicos, como los movimientos regulares y caóticos causados por la bifurcación. Además, una simulación de este tipo puede requerir muchos cálculos que un solo procesador requeriría días o incluso años para completarse. Tal es así, que los científicos se han visto obligados a investigar y desarrollar nuevas herramientas que faciliten la simulación de los sistemas dinámicos para obtener resultados más rápidos y analizar los fenómenos de una forma más eficiente. Es por ello que se necesitan de más herramientas tecnológicas para las diferentes investigaciones en los diferentes campos de la ciencia. La

implementación de hardware en los laboratorios de institutos superiores se ve necesario para el desarrollo de las practicas.

1.3.3 Justificación Metodológica

La metodología planteada en la investigación se basa en tres tipos de investigación, deductivo-inductivo, cuasi experimental y correlacional. Es así que se plantea la manipulación de la variable independiente que es el número de nodos, partiendo de conceptos que se asumen que son verdad como el funcionamiento del circuito de Chua. Por lo tanto, se parte de lo particular como es la sincronización de dos, tres, cuatro o más circuitos caóticos para poder generalizar el fenómeno físico que ocurre al sincronizar circuitos caóticos. Por último, se relaciona conceptos con el fin de poder realizar predicciones de resultados que se puedan obtener.

1.3.4 Justificación Social.

El análisis de los fenómenos físicos en el campo de la electrónica puede tener varias aplicaciones es por eso que el diseño y la implementación de hardware reconfigurable en la carrera de tecnología superior electrónica del IST Carlos Cisneros, ayudara a investigaciones en fenómenos de alta complejidad a su vez este hardware puede ser modificado y reproducido de diferentes formas para el análisis de fenómenos en las diversas carreras de especialidad.

1.4 OBJETIVOS

Los objetivos planteados están de acuerdo con la investigación desarrollada.

1.4.1 Objetivo General

Diseñar e implementar un sistema basado en hardware reconfigurable como herramienta de estudio de la sincronización de circuitos caóticos para el Instituto Técnico Superior CARLOS CISNEROS.

1.4.2 Objetivos específicos

- a) Diagnosticar los fenómenos físicos desarrollados mediante la electrónica en hardware reconfigurable.

- b) Diseñar el sistema de simulación numérica usando tecnología de hardware reconfigurable.
- c) Verificar estadísticamente que los resultados obtenidos mediante el hardware reconfigurable son correlacionales con los resultados obtenidos mediante el uso de un software de cálculo numérico.

1.5 Planteamiento de la hipótesis

1.5.1 Hipótesis General

Los resultados obtenidos de las variables de estado con la herramienta de hardware reconfigurable están correlacionados con los resultados obtenidos con el software de análisis numérico.

1.5.2 Hipótesis específicas

- a) Existen fenómenos físicos analizados y desarrollados mediante circuitos eléctricos.
- b) La herramienta de hardware reconfigurable permite encontrar las variables de estado en la sincronización de un número finito de nodos.
- c) Estadísticamente los valores obtenidos de las variables de estado mediante la herramienta de hardware reconfigurable están correlacionados con los valores de las variables de estado obtenidos mediante el software de análisis numérico

CAPÍTULO II

2 MARCO TEÓRICO

2.1 Situación problemática.

Los sistemas complejos son sistemas en los que el comportamiento colectivo de sus partes conlleva el surgimiento de propiedades que difícilmente, si no en absoluto, pueden inferirse de las propiedades de las partes. Entre los ejemplos de sistemas complejos se incluyen los hormigueros, las propias hormigas, las economías humanas, el clima, los sistemas nerviosos, las células y los seres vivos, incluidos los seres humanos, así como la energía moderna o las infraestructuras de telecomunicaciones.

En los últimos años, se están llevando a cabo estudios sobre sistemas no lineales entre ellos los sistemas complejos dinámicos caóticos, no obstante, es bastante difícil dar soluciones analíticas a este tipo de sistemas. Los sistemas caóticos presentan cambios bruscos al variar sus condiciones iniciales y no se puede prever su comportamiento hasta llegar a un resultado a partir del cálculo.

Por lo general, el estudio de estos sistemas se enfoca más en el aspecto teórico debido a que requieren un alto conocimiento de la matemática para entenderlos y desarrollarlos. El circuito de Chua es uno de los sistemas caóticos más estudiados en la actualidad. Su modelo matemático se describe a través de un sistema de ecuaciones diferenciales, en consecuencia, es necesario el desarrollo de programas de cálculo numérico que permitan modelar y simular este tipo de sistemas.

En este sentido la carrera de tecnología superior electrónica del IST Carlos Cisneros ve necesario la creación de una herramienta para poder estudiar los circuitos dinámicos en menor tiempo y con mayor facilidad de manejo de datos.

2.2 Antecedentes del problema:

En los últimos años se han desarrollado varios trabajos de investigación con respecto al tema propuesto en la presente tesis de los cuales se puede mencionar los siguientes:

2.2.1 *Modelado matemático de sistemas dinámicos en epidemiología*

Referencia: (Esther & Garaluz, 2014)

En este trabajo se estudiaron la identificación de sistemas dinámicos mediante técnicas de inteligencia computacional, este estudio está basado en el modelado y estimación de los parámetros de las epidemias de VIH/SIDA y dengue para el caso de Cuba. Para este análisis de los parámetros de estimación en este modelo se utilizó un método de estimación basado en las redes neuronales de Hopfield. Este método tiene una alta capacidad resolutive en lo problemas de optimización. El trabajo realizado permite evaluar la eficacia de las medidas de control de la epidemia.

2.2.2 Sistema open hardware y open source aplicados a la enseñanza de la electrónica.

Referencia:(Arango et al., 2014)

En este estudio realizaron el diseño y la implementación de un sistema open source y open hardware como herramientas. El uso de esta herramienta permite la construcción de sistemas complejos haciendo uso de software y hardware diseñados por terceros, estas herramientas son una opción económica, eficiente y flexible para implementar y desarrollar tecnología en el aula. Logran ajustar el comportamiento de varias aplicaciones obteniendo varios beneficios como: reducción del tiempo de desarrollo expansión de aplicaciones, modularidad y corrección de problemas de una manera práctica, reducen costos en el desarrollo de las investigaciones. Los sistemas con los que realizaron las aplicaciones open hardware utilizan el mismo sistema de conexión de los módulos arduino, lo que permite la conexión simultanea de varias tarjetas y así poder analizar sistemas de gran complejidad.

2.2.3 Spontaneous Synchronization in Two Mutually Coupled Memristor-Based Chua's Circuits: Numerical Investigations

Referencia: (Bilotta et al., 2014)

En este artículo se muestra el estudio del circuito de Chua, que es investigado mediante simulaciones numéricas; en este estudio se investigó la sincronización del comportamiento dinámico espontaneo de dos circuitos interactuando mutuamente a través de una resistencia de acoplamiento, analizando sus resultados mediante simuladores por ordenador con el fin de estudiar el fenómeno de sincronización auto organizada mediante ecuaciones que describen el sistema dinámico del mismo. Los resultados del análisis deducen que llega un tiempo en donde se sincronizan los dos circuitos.

2.2.4 Diseño e implementación sobre hardware reconfigurable de una arquitectura para la emulación en tiempo real de redes neuronales celulares.

Referencia: (*Universidad Politécnica de Cartagena, 2012*)

En este estudio analizan el diseño y la implementación de hardware reconfigurable de una arquitectura para la emulación en tiempo real de redes neuronales celulares (CNN), en el cual se llega a la conclusión que cuando la RNA demanda el uso de tecnología analógica, la solución apropiada, por sus inmejorables prestaciones en cuanto a velocidad de funcionamiento y consumo de área, es la basada en circuitos integrados de aplicación específica (ASIC, Application Specific Integrated Circuits). Si las técnicas de implementación son digitales, además de utilizar ASIC digitales, se podría emplear dispositivos hardware reconfigurables como las FPGA (Field Programmable Gate Array). Se pudo observar también que esta arquitectura facilita la emulación de red CNN complejas, compuesta por cientos de miles de millones de neuronas; el excelente compromiso alcanzado entre la velocidad de procesamiento y consumo de recursos hardware, hace que sea una interesante solución para considerar frente a otras alternativas.

2.2.5 Programación VHDL de algoritmos de codificación para dispositivos de hardware reconfigurable

Referencia: (Sandoval Ruiz & Fedón, 2008)

Este trabajo de investigación implementa un hardware reconfigurable de módulos de operación en álgebra de campos finitos de Galois, seleccionando el área de la codificación puesto que esta área en la matemática es la base de varios algoritmos en los campos de corrección de errores y procesamientos digitales de señales para criptografía; esto conlleva a mejorar su desempeño y seguridad. A estos algoritmos son preferible implementarlos en hardware sobre dispositivos reconfigurables. Para el desarrollo de este sistema se utilizó lenguaje descriptivo de hardware (VHDL) y la captura del diseño se lo realizó sobre dispositivos de compuertas programables (FPGA); para poder validar la salida del diseño fue utilizado ModelSim 5.7 a través de simuladores. En este trabajo se llegó a concluir que mediante la simulación en hardware reconfigurable se ha podido depurar y optimizar el procesamiento paralelo de la data, destacando como principal ventaja, el máximo nivel de paralelismo ofrecido por los FPGA.

2.2.6 Implementación de una red neuronal en FPGA para modelado de sistemas.

Referencia: (Cornejo, 2021)

En esta investigación buscan alternativas que permitan actualizar modelos matemáticos de forma continua, es así que desarrollo un trabajo investigativo que consistió en desarrollar una red neuronal para el modelo de sistemas dinámicos en un FPGA (Field Programmable Gate Array), para esto se utilizó el poder del Simulink de Matlab así como el Vivado mediante el software de System Generator. Esta red cuenta con una entrada de dos neuronas y una salida de una neurona. Se utilizó un método de aprendizaje de backpropagation (retro programación) cuya programación en FPGA no resulta trivial. Para poder validar la red diseñada se lo pudo hacer con el desarrollo de ecuaciones diferenciales. También se lo desarrollo en un microcontrolador utilizando el programa C++ con el objetivo de comparar ambas alternativas usando la misma configuración de red y el mismo conjunto de datos de entrenamiento. Mediante esta investigación se pudo concluir que el tiempo de ejecución en FPGA fue 31 veces más rápido y su entrenamiento fue 175 veces más rápido que en un microcontrolador.

2.3 Sistemas dinámicos

Fundamentalmente los estudios de los sistemas dinámicos están basados en el modelamiento matemático, el análisis y la simulación de sistemas físicos que le interesa a la ingeniería, tales como: los sistemas eléctricos, mecánicos, hidráulicos, neumáticos y térmicos. Al mismo tiempo son de particular importancia los sistemas híbridos que son el resultado de la combinación de dos o más sistemas de los mencionados anteriormente. De la misma forma, la teoría de los sistemas dinámicos puede ser aplicada a otros tipos de sistemas tales como: los sistemas químicos, económicos, biológicos, entre otros. Los modelos matemáticos de los sistemas dinámicos pueden ser representados básicamente de dos formas. Mediante ecuaciones diferenciales de primer orden conocidas como ecuaciones de estado o por medio de ecuaciones diferenciales de n-ésimo orden (Buitrago, 2009) (Moreno & Becerra ,2012).

Por otro lado, la mayoría los sistemas dinámicos se pueden caracterizar mediante ecuaciones diferenciales las mismas que se basan en las leyes físicas, tal es el caso de las leyes de Newton, Kirchhof entre otras leyes. Se puede definir un modelo matemático como la descripción matemática del comportamiento del sistema. En varias ocasiones al comenzar el estudio y análisis del sistema se obtiene un modelo matemático simple de interpretar y resolver puesto que se puede ignorar la no linealidad y los parámetros distribuidos (como en el caso de líneas de transmisión eléctrica), esto, con el propósito de obtener ecuaciones diferenciales lineales y de parámetros concentrados. Estos modelos pueden ser validos en operaciones de baja frecuencia y no a frecuencia alta. Uno de los aspectos de real importancia en el estudio de sistemas dinámicos es el concerniente a los que tienen

carácter no lineal puesto que estos permiten estudiar no solamente cuestiones relacionadas a la estabilidad del sistema sino también a posibles comportamientos caóticos y operaciones a altas frecuencias. Este estudio de sistemas no lineales se analiza a menudo de manera numérica puesto que en general es muy complicado encontrar soluciones analíticas.

Son parte básica de la denominada dinámica no lineal algunos aspectos tales como bifurcaciones, periodicidades y caos existiendo varias formas de poder abordarlos (Moreno & Becerra, 2012) (Buitrago, 2009) (Amicos, 2011). En este sentido, estudiar los sistemas dinámicos involucra el modelamiento matemático, el análisis y la simulación de sistemas físicos compuestos de un conjunto de magnitudes medibles que evolucionan con el tiempo (Zadunaisky & Dinámicos, 2009).

2.4 Clasificación de los sistemas dinámicos.

Los sistemas dinámicos pueden ser clasificados de varias formas dentro de las cuales se pueden mencionar las siguientes:

2.4.1 Sistemas con parámetros concentrados y con parámetros distribuidos.

Estos se pueden subdividir en dos grandes grupos de acuerdo con la naturaleza de sus parámetros, en el caso que la excitación depende apenas del tiempo se los denomina sistemas de parámetros concentrados y son descritos por ecuaciones diferenciales ordinarias. Cuando la excitación y la respuesta dependen del tiempo y de las coordenadas espaciales se los denomina sistemas con parámetros distribuidos y son descritos por ecuaciones diferenciales parciales (más de una variable) (Moreno & Becerra 2012).

2.4.2 Sistemas lineales y no lineales

En (Moreno & Becerra, 2012) se menciona que una de las propiedades que tiene profundas implicaciones en el análisis de sistemas es la linealidad. Si se considera la relación entre la entrada $r(t)$ y la salida $c(t)$ mediante un diagrama de bloques y considerando dos pares de entradas y salidas $r_1(t), c_1(t)$ y $r_2(t), c_2(t)$. Entonces para el mismo sistema, se tiene una entrada $r_3(t)$. Por lo cual se obtiene una combinación lineal de $r_1(t)$ y $r_2(t)$ donde α_1 y α_2 son constantes:

$$r_3(t) = \alpha_1 r_1(t) + \alpha_2 r_2(t) \quad (2-1)$$

Si la salida $c_3(t)$ representa una combinación lineal expresada en la siguiente forma se puede decir que el sistema es lineal.

$$c_3(t) = \alpha_1 c_1(t) + \alpha_2 c_2(t) \quad (2-2)$$

Caso contrario si tenemos la siguiente expresión decimos que se trata de un sistema no lineal.

$$c_3(t) \neq \alpha_1 c_1(t) + \alpha_2 c_2(t) \quad (2-3)$$

Se puede concluir que para un sistema lineal las respuestas que se den a su excitación se las puede hallar de forma separada y después combinarlas linealmente; a esto se le conoce como el principio de superposición que es el principio fundamental de la teoría de los sistemas lineales.

Una de las ventajas en trabajar con sistemas lineales es que su modelo matemático esta descrito por un sistema de ecuaciones diferenciales lineales el cual es fácil en su solución analítica. Por otra parte, el modelo matemático de los sistemas no lineales es descrito por ecuaciones diferenciales no lineales las cuales son de complicada solución analítica; estas se pueden desarrollar imponiendo ciertas hipótesis de simplificación o apelando a métodos numéricos aproximados como son Euler o Runge-Kutta, los cuales se pueden desarrollar en software de simulación como Matlab.

2.4.3 Sistemas variantes en el tiempo e invariantes en el tiempo.

De acuerdo con (Buitrago, 2009), en los modelos matemáticos los parámetros que resultan del análisis del sistema aparecen en ecuaciones diferenciales en forma de coeficientes, es así que, de acuerdo a la forma de sus ecuaciones de estado se puede decir:

Si un sistema es no lineal y variante con el tiempo, las ecuaciones de estado se pueden escribir como:

$$\dot{x}(t) = a(x(t), u(t), t) \quad (2-4)$$

Si el sistema es no lineal e invariante con el tiempo las ecuaciones de estado son de la forma:

$$\dot{x}(t) = a(x(t), u(t)) \quad (2-5)$$

Si el sistema es lineal y variante con el tiempo sus ecuaciones de estado son:

$$\dot{x}(t) = A(t)x(t) + B(t)u(t) \quad (2-6)$$

Siendo $A(t)$ y $B(t)$ matrices de dimensiones $n \times m$ y $n \times p$ respectivamente con elementos variantes en el tiempo.

Si el sistema es lineal e invariante con el tiempo son de la forma:

$$\dot{x}(t) = Ax(t) + Bu(t) \quad (2-7)$$

Siendo A y B matrices constantes.

2.4.4 Sistemas continuos y sistemas discretos:

Se llama sistema continuo si es sometido a una entrada continua con el tiempo $r(t)$ y presenta una salida continua $c(t)$; su modelo matemático está constituido por ecuaciones diferenciales. Por otro

lado se llama sistema discreto si está sometido a una entrada discreta en el tiempo $\{r_K\}$ (una secuencia de números) y presenta también una salida discreta, $\{c_K\}$ (otra secuencia de números); su modelo matemático está formado por ecuaciones diferenciales finitas (Moreno & Becerra, 2012).

2.4.5 Respuesta del sistema:

Para conocer el comportamiento del sistema cuando es sometido a una excitación con condiciones iniciales (desplazamiento inicial, velocidad inicial), basta con la solución de la ecuación diferencial del modelo matemático. La solución de la ecuación diferencial consiste en dos partes: la solución homogénea y la solución particular

Solución homogénea: corresponde al caso en donde la excitación externa a ser nula pudiendo el sistema entrar en movimiento solo cuanto se le imponga condiciones iniciales caso contrario permanece en reposo.

Solución particular: corresponde a la parte de la respuesta debida enteramente a la excitación externa, considerando las condiciones iniciales nulas. La naturaleza de la respuesta depende de las características del sistema dinámico, así como de la excitación utilizada, por lo cual, conviene distinguir entre respuesta permanente y respuesta transitoria.

Respuesta permanente: Matemáticamente, es la parte de la respuesta total que permanece cuando el tiempo tiende al infinito.

Respuesta transitoria: matemáticamente, es la parte de la respuesta total que desaparece cuando el tiempo tiende al infinito, este tipo de respuesta depende fuertemente del tiempo (Moreno & Becerra, 2012).

2.5 Sistemas dinámicos caóticos.

Los sistemas dinámicos deterministas en los cuales el ámbito de aplicación cubre todas las ramas de la ciencia, poseen movimientos de tal complejidad que se hace imposible toda predicción, por lo tanto, reciben el nombre de caóticos.

La mayoría de los sistemas dinámicos caóticos poseen movimientos tan complejos con sus trayectorias entrecruzándose de una forma tan errática y turbulenta lo que hace imposible tener una predicción detallada para tiempos grandes y es extremadamente difícil su estudio. En esos casos se habla de caos, de comportamiento caótico, turbulento o estocástico. Su característica más importante constituye la extrema sensibilidad de los movimientos al desarrollar pequeñas variaciones en las posiciones iniciales que son imposibles de eliminar, es así que surge la necesidad de aproximarlas

aplicando los métodos de cálculo numérico. Esto significa que dos trayectorias posibles, que en el instante inicial están muy próximas pueden llegar a separarse de manera brusca y violenta al cabo de un tiempo; sin ninguna intervención exterior, esto conlleva a que si se da una pequeña imprecisión en el dato inicial hace que se pueda confundir con la otra lo que significa un error grande o incluso un desconocimiento completo del estado final. Desde hace miles de años se utilizan estos sistemas en diferentes campos como por ejemplo los juegos de azar (dados, monedas, ruletas) o para fines mágicos o adivinatorios (cartomancia). En esos casos se podría hacer predicciones válidas para tiempos cortos, pero se hace muy difícil cuanto se tiene tiempos prolongados (*MovimientoCaotico011.Pdf*, n.d.).

2.6 Regresión lineal simple

Los factores que intervienen en un experimento pueden ser cualitativos o cuantitativos, por lo tanto, un factor cuantitativo puede asociarse con puntos en una escala numérica como, por ejemplo, la temperatura, la presión, el tiempo, entre otros. Mientras que los factores cualitativos son aquellos que no pueden ordenarse por magnitud; en consecuencia, para analizar estos factores se puede utilizar la regresión lineal simple que no es más que, el cálculo de la ecuación correspondiente a la línea que mejor describe la relación entre la respuesta y la variable que la explica. Esta ecuación va a representar la línea que mejor se ajusta a los puntos en un gráfico de dispersión, figura 1-2 (Carrasquilla-Batista et al., 2016).

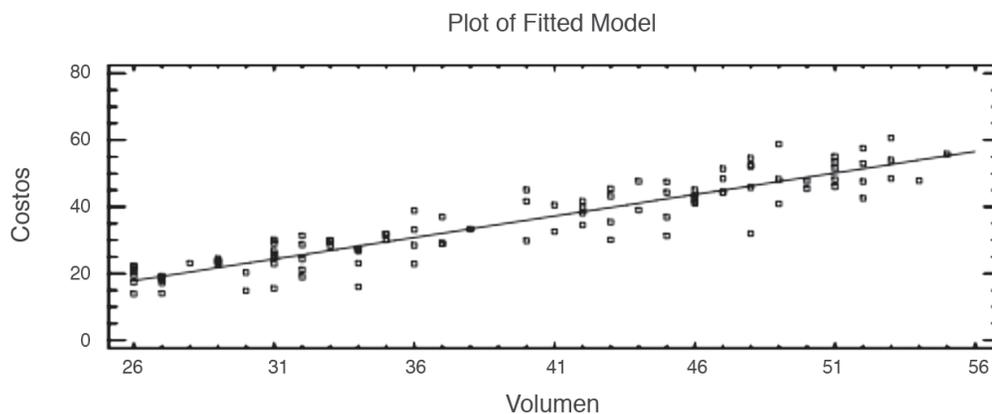


Figura 1-2. Regresión lineal volumen vs costos

Realizado por: (Carrasquilla-Batista et al., 2016)

Se debe recordar que en la regresión lineal simple se tiene una única variable predictoría. En caso de que se tenga dos o más variables regresoras o predictorías se debe recurrir a la regresión lineal múltiple.

Previo a todo análisis, es recomendable realizar una inspección visual de los datos con el objetivo de comprobar la conveniencia o no de la utilización del modelo de regresión lineal simple. El diagrama de dispersión o nube de puntos dará una idea si existe una relación o no entre las variables o si tiene una tendencia lineal o no. También se puede extraer información de interés como el grado de estrechez de la nube de puntos, indicadora de la intensidad de la relación, si existe valores anómalos que distorsionen la posible relación o si la distorsión de los datos es uniforme, de este modo, se tiene algunos diagramas de dispersión, figura 2-2 (Molina, n.d.).

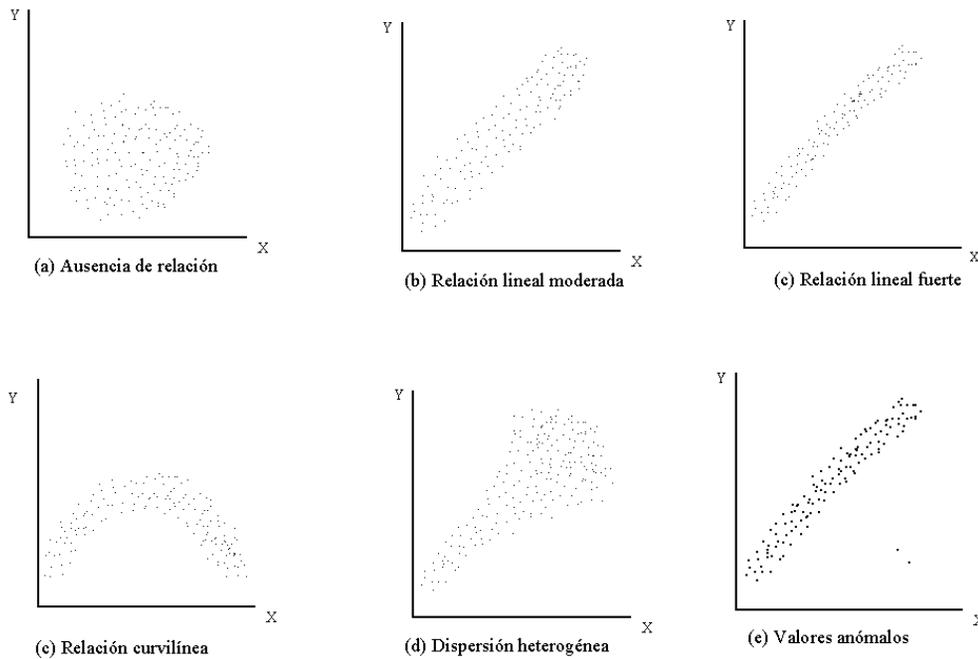


Figura 2-2: Posibles diagramas de dispersión.

Realizado por : (Molina, n.d.)

La ecuación para una línea recta es

$$y = b_0 + b_1x \tag{2-8}$$

Dónde:

- Y representa la variable de respuesta
- X representa la variable periódica

- b_0 representa la intersección con el eje y cuando x tiene un valor de 0
- b_1 representa la pendiente, por lo tanto, determina la cantidad en que cambia y cuando x se incrementa en una unidad

Existen también, los residuos que son las distancias entre los puntos y la línea de regresión. Estos van a representar la porción de respuesta que no es explicada por la ecuación de regresión, es decir que la diferencia entre el valor observado y el valor aproximado es el residuo.

En la regresión lineal se utilizan los mínimos cuadrados, a los cuales se conoce también como mínimos cuadrados de regresión; estos van a determinar la línea que minimiza la suma de las distancias verticales cuadradas desde los puntos hacia la recta, figura 3-2.

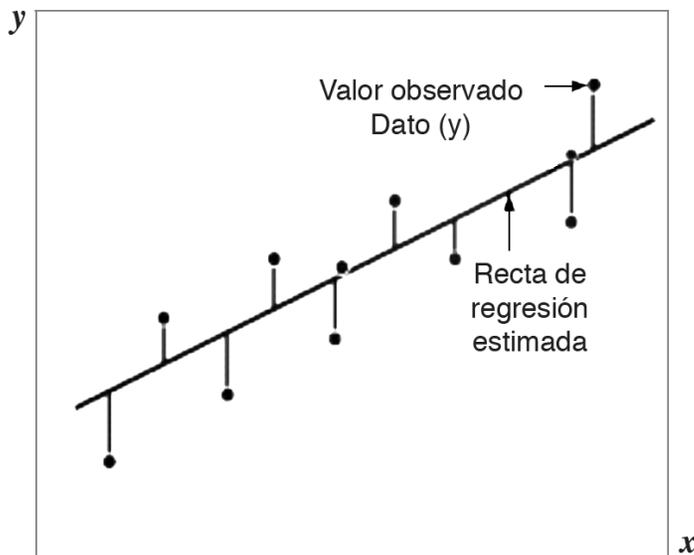


Figura 3-2: Representación de los mínimos cuadrados

Realizado por: (Carrasquilla-Batista et al., 2016)

2.7 Método de Euler

Las ecuaciones diferenciales aparecen cuando se va a modelar situaciones físicas ya sea en el área de las ciencias naturales, ingeniería, electrónica y otras disciplinas donde se encuentran razones de cambio de una o varias funciones desconocidas con respecto a una o varias funciones independientes (Guzman et al., n.d.).

Se conoce como el método de Euler al método numérico que consiste en ir incrementando paso a paso la variable tomada como independiente y encontrando la siguiente imagen con la derivada.

La primera derivada va a dar como resultado una estimación directa de la pendiente X_i

$$\phi = f(X_i, Y_i) \quad (2-9)$$

Por lo tanto $f(X_i, Y_i)$ es la ecuación diferencial evaluada en X_i y Y_i , tal estimación podrá substituirse en la ecuación.

$$Y_{i+1} = Y_i + f(X_i, Y_i)h \quad (2-10)$$

Esta fórmula representa el método de Euler (punto medio). Se predice un nuevo valor de Y por medio de la pendiente (igual a la primera derivada en el valor original de X).

Existen dos tipos de errores en el método al dar solución numérica a las ecuaciones ordinarias (EDO), los cuales se mencionan a continuación:

- **Error de truncamiento o discretización**, estos son causados por la naturaleza de las técnicas que son empleadas para aproximar los valores de y.
- **Error de Redondeo**, estos son causados por el número límite de cifras significativas que pueden retener una computadora.

Existe también el método de Euler mejorado, el mismo que se basa en la misma idea que el método anterior, pero este método realiza un refinamiento en la aproximación, tomando un promedio entre algunas pendientes, para lo cual se cuenta con la siguiente formula.

$$Y_{n+1} = Y_n + h \left[\frac{f(X_n, Y_n) + f(X_{n+1}, Y_{n+1})}{2} \right] \quad (2 - 11)$$

Dónde

$$Y_{n+1} = Y_n + f(X_n, Y_n)h \quad (2-12)$$

Se analiza el primer paso de aproximación con la siguiente gráfica para poder comprender la formula, figura 4-2.

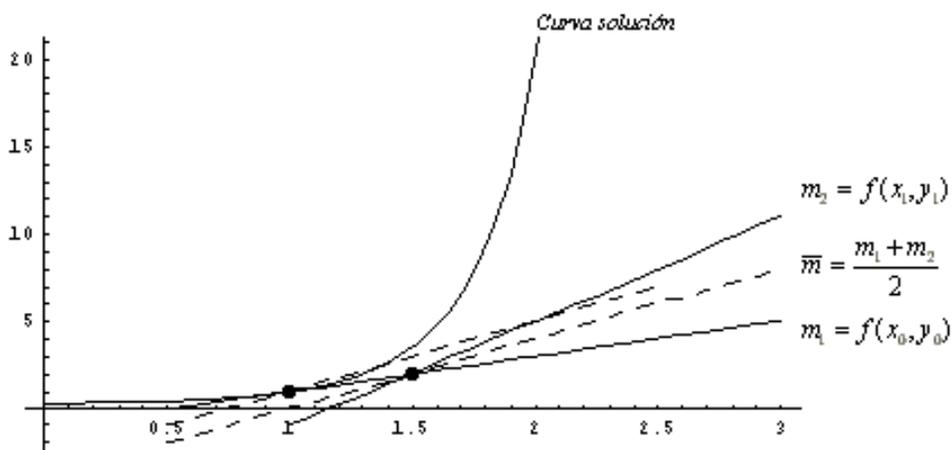


Figura 4-2: Curva Solución

Realizado por: (Guzman et al., n.d.)

En la gráfica se puede observar que la pendiente promedio corresponde a la pendiente de la bisectriz de la recta tangente a la curva en el punto de la condición inicial y la recta tangente a la curva en el punto

$$(X_i, Y_i)$$

Donde Y_i es la aproximación obtenida con la primera fórmula de Euler. Por último, esta recta bisectriz se traslada paralelamente hasta el punto de la condición inicial, y se considera el valor de esta recta en el punto $X = X_1$ como la aproximación de Euler mejorada.

2.8 Octave

En su propia documentación Octave se describe como un lenguaje de programación de alto nivel que está orientado al Cálculo Numérico. Proporciona una consola para resolver problemas lineales y no lineales con el ordenador y para desarrollar experimentos numéricos. Tiene una sintaxis similar al paquete de cálculo numérico MATLAB, este es prácticamente compatible hasta las versiones 4.

Octave fue pensado originalmente para ser un software de acompañamiento de un libro de texto que trata sobre reactores químicos, este libro fue escrito por James B. Rawling de la Universidad de Wisconsin- Madison y John G. Ekerdt de la Universidad de Texas (Castro, 2013).

Este paquete se caracteriza porque el tipo de datos básicos es la matriz matemática de dos dimensiones, para las mismas tiene implementadas gran cantidad de operaciones. Puede también manejar cadenas con caracteres string y otros contenedores de datos más complejos como estructuras arrays de celdas o listas. Debido a que este paquete tiene la facilidad para la creación de funciones con números de entrada y salida, ha podido ampliarse con conjuntos de funciones toolbox; estas abordan numerosos problemas en los campos de las ciencias y las ingenierías relacionadas al cálculo numérico, estadístico, procesamiento de señales, control de sistemas entre otros (Castro, 2013).

Gracias a la utilización de la aplicación Gnuplot tiene capacidades para realizar gráficos bidimensionales bastantes complejas a más de posibles gráficas tridimensionales un poco más sencillas.

Octave fue diseñado para ser una herramienta dentro del sistema operativo GNU, esto quiere decir que puede ser copiado modificado y redistribuido bajo los términos de la licencia GNU GPL. Este paquete en un principio fue un lenguaje de programación independiente, pero ha ido convergiendo a Matlab hasta el punto de buscar la compatibilidad con él. Octave tiene la capacidad de ejecutar la mayoría de los códigos escritos en Matlab. Una de las diferencias que más se evidencian es que están escritos en lenguajes de programación distintas, Matlab en C y Octave en C++ (Torres, n.d.).

Octave hoy en día es una herramienta de una calidad altísima para el desarrollo de pequeños proyectos, es inferior a la herramienta Matlab, pero para ser un programa gratuito es muy eficiente en el desarrollo de modelos matemáticos.

2.9 Circuito de Chua

El circuito de Chua, es uno de los ejemplos clásicos de sistemas caóticos, esto se debe a su robustez, bajos costos y gran variedad de regímenes dinámicos. Este es un de los pocos sistemas que su comportamiento caótico tiene comprobación teórica, experimental y numérica. Por su fácil construcción permite analizar varios ejemplos de fenómenos caóticos, debido a la fácil manipulación de sus parámetros. Su estudio, permite nuevas formas de la investigación como el estudio de la eliminación de caos, estudiar el fenómeno de sincronización de fases, entre otros (Físicas , 2016).

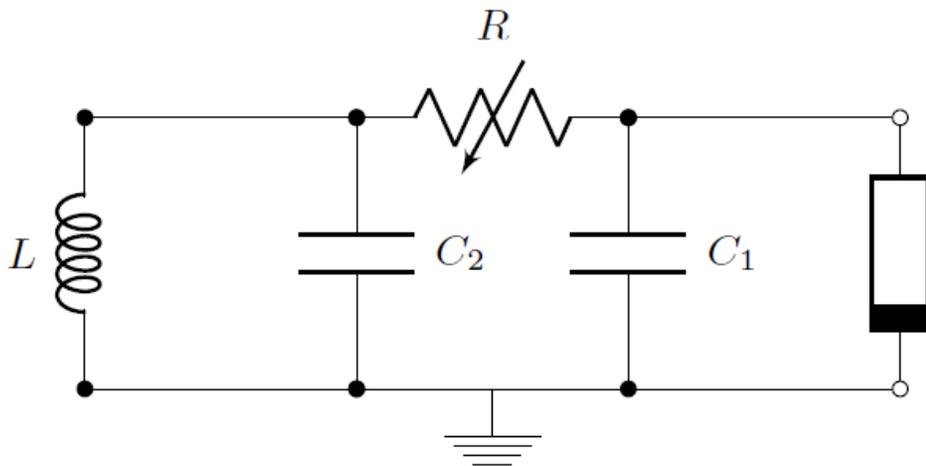


Figura 5-2: Circuito de Chua.

Realizado por: (Físicas , 2016)

El circuito de Chua, figura 5-2, consiste en un inductor L , dos capacitores C_1 y C_2 , un diodo de Chua $g(V_1)$ y una resistencia variable R_v . Utilizando las leyes de Kirchhof, se presentan las siguientes ecuaciones para el análisis del sistema.

$$C_1 \frac{dV_1}{dt} = \frac{1}{R_v} (V_2 - V_1) - gV_1 \quad (2-13)$$

$$C_2 \frac{dV_2}{dt} = \frac{1}{R} (V_1 - V_2) + I_L \quad (2-14)$$

$$L \frac{dI_L}{dt} = -V_2 \quad (2-15)$$

Dónde la corriente gV_1 que atraviesa el diodo de Chua está definido por.

$$g(V_1) = \begin{cases} G_b V_1 + (G_b - G_a)E, & V_1 \leq -E \\ G_a V_1 & |V_1| < E \\ G_b V_1 + (G_a - G_b)E, & V_1 \geq E \end{cases} \quad (2-16)$$

Dónde:

V_1, V_2 , son diferencia de potencial en los capacitores C_1 y C_2 respectivamente.

I_L , es la corriente del inductor L

G_b y G_a , son las conductancias que estrictamente significan las pendientes de las rectas cuando se grafica la corriente g versus V_1 .

2.10 Sincronización del circuito de Chua

Es interesante pensar en que sucederá al acoplar dos o más sistemas caóticos. Uno de los propósitos principales de la tesis es investigar que pasara si conectamos de dos o más circuitos caóticos tomando como referencia el circuito de Chua.

- En (Nari & Chua, 2002), se decide acoplar de forma lineal dos circuitos de Chua por varias razones.
- Este acople puede representar una señal de sincronía para realizar transmisiones entre circuitos de Chua que se encuentran en localidades remotas.
- El acople realizado por resistencias puede servir como una primera aproximación para los diferentes acoples de circuitos de Chua en un sistema de comunicación inalámbricas.
- En este tipo de acople es sencillo el análisis y no añade complejidad al circuito.

Para analizar el sistema se procede a encontrar sus ecuaciones de estado obteniendo un sistema de seis ecuaciones diferenciales acopladas.

$$\begin{aligned} \frac{dI_{3A}}{dt} &= -\frac{1}{L_A} V_{2A} \\ \frac{dV_{2A}}{dt} &= \frac{1}{G_A} I_{3A} - \frac{1}{C_{2A}} G_A (V_{2A} - V_{1A}) \\ \frac{dV_{1A}}{dt} &= \frac{1}{C_{1A}} G_A (V_{2A} - V_{1A}) - \frac{1}{C_{1A}} f(V_{1A}) - \frac{1}{C_{1A}} G_C (V_{1A} - V_{1B}) \\ \frac{dI_{3B}}{dt} &= -\frac{1}{L_A} V_{2B} \\ \frac{dV_{2B}}{dt} &= \frac{1}{C_{2B}} I_{3B} - \frac{1}{C_{2B}} G_B (V_{2B} - V_{1B}) \\ \frac{dV_{1B}}{dt} &= \frac{1}{C_{1B}} G_C (V_{1A} - V_{1B}) - \frac{1}{C_B} f(V_{1B}) + \frac{1}{C_{1B}} G_B (V_{2B} - V_{1B}) \end{aligned} \quad (2-17)$$

Dónde las funciones $f(V_{1A}), f(V_{1B})$ representan la respuesta de la resistencia no lineal o diodo de Chua. Esta resistencia esta descrita matemáticamente por la ecuación

$$f(V_r) = m_1 V_r + \frac{1}{2}(m_o - m_1)(|V_r + E| - |V_r - E|) \quad (2-18)$$

La resolución de este sistema no resulta trivial por lo cual es necesario un software o hardware para obtener los datos deseados a diferentes condiciones iniciales.

2.11 Hardware reconfigurable.

Un sistema basado en hardware reconfigurable se obtiene, tomando en cuenta por un lado los lenguajes HDL y por otro el elemento físico como por ejemplo las FPGAs. Según (Arce Ferandez Fuentes, 2008), la metodología del diseño al utilizar hardware programable es similar a la de un sistema digital, salvo que al final se va a obtener un fichero ejecutable que se puede enviar al FPGA para ser reconfigurado, implementando así el diseño.

Durante el diseño se debe tomar en cuenta varias etapas de simulación. Inicialmente la descripción RTL o en HDL, se realiza mediante la creación de bancos de prueba para poder simular el sistema y llegar a los resultados. Se mapea el diseño en una netlist que es luego traducida a un nivel descrito por compuertas lógicas. La simulación se repite para confirmar que el proceso de síntesis no tenga errores. Por último, el diseño es introducido en la FPGA y se vuelve a simular la. Todo el desarrollo de este proceso es automático y controlado por el software especializado.

2.12 Hardware reconfigurable libre:

En (González., 2003) se menciona que un hardware reconfigurable se puede trabajar exactamente igual a un software por lo tanto las características que posee son:

- a) Se le pueden ofrecer las 4 mismas libertades del software libre a los ficheros HDL.
- b) Aparecen comunidades hardware que van a compartir información, las cuales son OpenCore [17] y Open Collector [18]
- c) Se puede implementar repositorios hardware, lo que va a permitir que muchas personas puedan participar en el desarrollo.
- d) Pueden aparecer distribuidores que puedan recopilar todo el hardware libre existente.

Para poder ejecutar el software se necesita de una máquina que tenga procesador, para poder probar el hardware es necesario una plataforma FPGA en la que se descargue el diseño. La tarjeta JPS es una opción de ellas, que además esté es libre.

2.13 FPGAS

Son dispositivos que permiten implementar circuitos digitales. Estas están compuestas por bloques iguales configurables llamados CLBs los mismos que se van a unir de una forma dinámica según como se especifique en la memoria de configuración. De esta forma, cambiando el contenido de esta memoria se establecen diferentes uniones entre los CLBs, permitiendo así, obtener un dispositivo u otro (González , 2003).

El fichero que contiene la configuración se denomina bitstream. La característica fundamental de las FPGAs es que son dispositivos universales. Esto permite que se puedan convertir en cualquier diseño digital según el bitstream que sea cargado en su memoria de configuración (González et al., 2003).

2.14 Estructura de las FPGA

Internamente una FPGA es una serie de pequeños dispositivos lógicos, que por algunos fabricantes son llamados CLB (Configurable Logic Block) organizados por filas y columnas, figura 6. Entre los CLB existe un gran número de elementos de interconexión, líneas que pueden unir unos CLB con otros y con partes del FPGA, en estos se pueden hallar líneas de distintas velocidades. También existen varios elementos en cada uno de los pines del chip para poder definir la forma en la que se está trabajando (entrada, salida o ambos), se lo suelen llamar IOB (Input/Output Block) Según se especifique en la memoria de configuración los CLB se unen dinámicamente, de esta forma al cambiar el contenido de esta memoria, se establecen unas uniones diferentes entre los CLBs, obteniendo un dispositivo u otro. La característica fundamental de las FPGAs es que son dispositivos universales, se los puede convertir en cualquier diseño digital, según el bitstream que se cargue en su memoria de configuración (Arce Fernandez Fuentes, 2008).

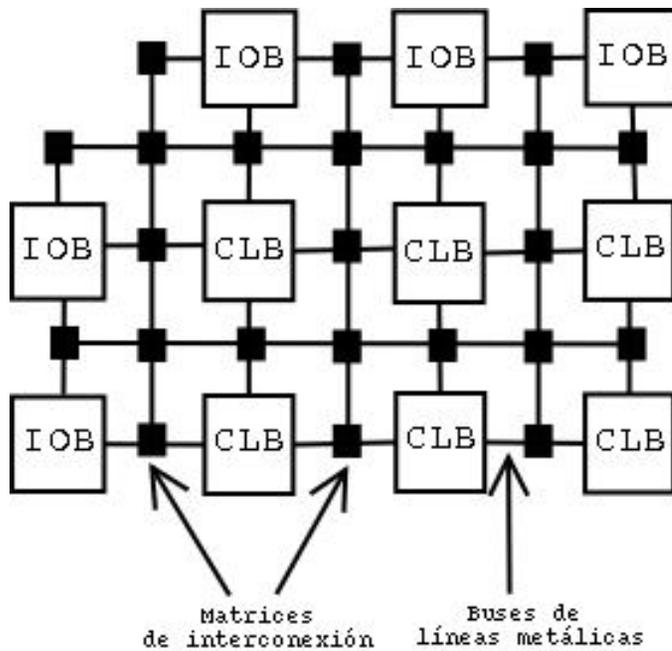


Figura 6-2. Estructura interna de un FPGA

Realizado por: (Arce Ferandez Fuentes, 2008)

2.15 Representación de un punto fijo.

La representación de punto fijo permite usar números fraccionarios en hardware de enteros de bajo costo. A continuación, se revisa el formato Q para representar números fraccionarios.

2.15.1 Representación de números fraccionarios en DSP de bajo costo

Para reducir el costo de la implementación, muchos procesadores de señales digitales están diseñados para realizar operaciones aritméticas solo con números enteros. Para representar números fraccionarios en estos procesadores, es posible usar un punto binario implícito.

Por ejemplo, la palabra de ocho bits $a=01010110_2$ representa 86_{10} cuando se interpreta como un número entero; sin embargo, podemos considerar un punto binario implícito para a e interpretarlo como un número fraccionario.

Suponiendo que el punto binario está entre las posiciones de bit cuarto y quinto, es decir, $a=0101.0110_2$, se obtiene el valor decimal equivalente de este número como:

$$a = 0x2^3 + 1x2^2 + 0x2^1 + 1x2^0 + 0x2^{-1} + 1x2^{-2} + 1x2^{-3} + 0x2^{-4} = 5,375$$

Con esta interpretación, estamos usando cuatro bits para representar la parte entera del número y otros cuatro bits para representar la parte fraccionaria.

Como se puede observar, la posición del primer bit a la derecha del punto binario tiene un peso de 0,5, la posición del segundo bit tiene un peso de 0,25 y así sucesivamente. Se debe tomar en cuenta que este punto binario implícito no está especificado en el hardware. Y el programador necesita considerar un factor de escala apropiado para interpretar correctamente el resultado de los cálculos. Para el ejemplo anterior, el hardware utiliza ocho elementos de almacenamiento para representar la palabra de ocho bits $a = 01010110_2$. Ahora bien, si el programador pretende representar 5.375 con a , debe recordar que un factor de escala de 2^{-4} debe aplicarse adecuadamente al resultado de cualquier cálculo que utilice la variable a .

Como otro ejemplo, se puede considerar el punto binario de a entre las posiciones de bit quinto y sexto, es decir, $a = 010.10110_2$. En este caso, el valor decimal equivalente será $a = 2,6875_{10}$

2.15.2 El formato Q

Dependiendo de dónde se supone que está el punto binario, un número dado puede interpretarse como varios valores diferentes. Para simplificar la programación, generalmente se utiliza un punto binario fijo en todo el algoritmo. Para especificar fácilmente cuántos bits se usan para representar las partes enteras y fraccionarias del número, se usa una notación llamada formato Q. Por ejemplo, para especificar que se están usando tres bits para la parte entera y cuatro bits para la parte fraccionaria, es posible decir que los números están en formato Q3.4.

Otra notación posible es especificar solo la longitud de la parte fraccionaria. Esto se basa en la suposición de que se conoce la longitud de palabra para un procesador determinado. Por ejemplo, cuando se trabaja con un procesador que tiene una longitud de palabra de 16 bits, se puede decir simplemente que estamos usando el formato Q15 para representar los números. Esto significa que se están colocando 15 bits a la derecha del punto binario y un bit a su izquierda. En este caso, el formato Q15 es equivalente al formato Q1.15.

2.15.3 Elegir la posición del punto binario

Para elegir la posición del punto binario, se debe considerar dos factores principales:

- a) El número más grande que a representará en un algoritmo dado
- b) El ruido de cuantificación tolerable

El primero especifica cuántos bits se deben usar para la parte entera y el segundo determina la longitud de la parte fraccionaria.

Tenga en cuenta que, además de usar un factor de escala implícito, el formato Q no tiene nada nuevo en comparación con el conocido concepto de representar números en una computadora digital. Como resultado, también se puede usar el formato Q para representar números en complemento a dos con signo. En este caso, solo necesitamos asignar el bit más significativo (MSB) al signo y usar la forma de complemento a dos para los números negativos.

2.16 Marco conceptual:

Circuito: Es un procesador de señal o energía, está conformado por la interconexión de elementos o dispositivos simples. La energía en un circuito puede ser absorbida, suministrada, almacenada o convertida (Gomez, 2008).

Hardware: Es el término que se utiliza para hacer referencia a cualquiera de las partes de la computadora, incluso se lo menciona cuando se habla de la computadora completa, en otras palabras, son componentes electrónicos que funcionan como dispositivos de entrada, salida, almacenamiento y procesamiento (Contaduría, 2010).

Hardware reconfigurable. Es aquél que está descrito mediante un lenguaje HDL (Hardware Description Language). Este se desarrolla de una manera muy similar a como se lo hiciera con un software cuyos diseños son ficheros de texto que contienen el código fuente. Se les puede aplicar directamente una licencia libre, como la GPL (González., 2003).

HDL: Se refiere a un tipo de lenguaje como por ejemplo VHDL, Handel C y Verilog. De esta manera los ficheros se van a convertir en ficheros de texto ASCII (“código de fuente”) que describen tanto la estructura del diseño como el comportamiento de sus partes integrantes. Se pueden desarrollar también librerías de componentes que luego se pueden usar en diseños más complejos. Inicialmente estos lenguajes son empleados para describir de una forma no ambigua el hardware y poder desarrollar simulaciones (González et al., 2003).

Programa: Es el término utilizado para referirse a los elementos de información (software) que van a ejecutar una serie de instrucciones para realizar las operaciones en la computadora (Contaduría, 2010).

Simulación Numérica. La simulación numérica va a integrar las herramientas matemáticas que permiten modelar, simular, o predecir el comportamiento de varios dispositivos, productos y procesos en el campo de la ingeniería y ciencias aplicadas (Luis Ricarod Orozco Villarruel, 2019).

Sincronización: Se define a la sincronización como la satisfacción de restricciones temporales en la interacción de los procesos. Se puede mencionar también que sincronizar se refiere a que dos o más elementos, fenómenos, eventos u operaciones sean programados para que ocurran en un orden y/o momentos predefinidos de lugar o tiempo (Alcalá Minerva., 2013).

Sistema: Es el conjunto de programas que van a controlar y administrar los recursos del hardware, tales como el procesador, la memoria principal, los dispositivos de entrada, salida, almacenamiento y todo en cuanto se refiere a el manejo de la información (Contaduría, 2010).

Software: Es el término que se utiliza de manera general para referirse a los programas computacionales y en algunas formas también a los archivos de datos, en otras palabras, son elementos de información que utiliza la computadora (Contaduría, 2010).

CAPÍTULO III

3 METODOLOGÍA DE LA INVESTIGACIÓN

3.1 Metodología:

3.1.1 *Tipo y diseño de investigación:*

Se establece un tipo de investigación cuasi experimental, inductivo-deductivo y correlacional.

3.1.2 *Cuasi experimental*

Este tipo de investigación tiene como objetivo poner a prueba una hipótesis casual mediante la manipulación de al menos una variable independiente. Por lo tanto, la variable que se pretende manipular es el número de nodos, estos nodos no pueden ser analizados de forma aleatoria que es otra propiedad de este tipo de investigación. La asignación de condiciones iniciales se lleva a cabo por autoselección, identificando grupos de comparación. En consecuencia, los resultados obtenidos se los puede comparar con resultados preestablecidos en la investigación.

3.1.3 *Inductivo-Deductivo*

El método deductivo parte de afirmaciones generales y de hechos que ya existen. Es así como, se parte de estudios ya realizados asumiendo que son verdad, así se asume que el circuito de Chua es un circuito dinámico caótico. Estos circuitos pueden acoplarse entre sí formando un sistema caótico que puede ser sincronizado. Por lo tanto, mediante el hardware reconfigurable se puede sincronizar dichos circuitos aplicando las reglas de la lógica en su desarrollo y poder probar que un sistema de hardware funciona para este fenómeno de sincronización. Partiendo de casos particulares para dos, tres o más circuitos de Chua se pueden llegar a generalizar por medio de un estudio inductivo.

3.1.4 *Correlacional*

La investigación correlacional se basa en relacionar conceptos o variables mediante pruebas de hipótesis correlacionales y la aplicación de técnicas estadísticas para estimar su correlación. Es así como, en el tema de estudio planteado se relacionan los conceptos de sincronización, sistemas, caos.

En base al estudio de estos conceptos y con el análisis de las hipótesis planteadas se pueden llegar a correlacionar los mismos.

3.2 Métodos de investigación.

El análisis del tema planteado parte del estudio bibliográfico sobre los sistemas dinámicos caóticos. Mediante este estudio se puede deducir que existen circuitos dinámicos caóticos basados en el circuito de Chua. Estos circuitos pueden ser sincronizados mediante la interconexión de dos o más circuitos caóticos y pueden ser estudiados mediante software o hardware, por lo tanto, se procede de la siguiente manera lógica.

Como primer paso está el análisis y determinación de las ecuaciones diferenciales que gobiernan el circuito de Chua. Se plantea el sistema de ecuaciones diferenciales obtenidas al interconectar a través de una resistencia dos o más circuitos de Chua y finalmente se llevará a cabo la simulación numérica. Seguidamente se implementará el sistema de simulación numérica utilizando hardware reconfigurable, utilizando el lenguaje de descripción de hardware (HDL). La implementación inicia con la simulación funcional, mediante este método se asegura que el diseño introducido funciona como debería en términos de su operación lógica antes de sintetizarlo en un diseño hardware. Se procede a realizar la síntesis y optimización para reducir la cantidad de recursos consumidos por el sistema. Finalmente se implementa y se lleva a cabo la simulación temporal para determinar la frecuencia máxima de operación del sistema. En la última fase se llevará a cabo la simulación numérica en el sistema basado en hardware implementado en la fase anterior y los resultados serán validados estadísticamente a través de la comparación con los resultados obtenidos en la simulación numérica funcional.

3.3 Enfoque de la investigación:

Se parte de un enfoque cuantitativo por lo que se desea describir, explicar y predecir fenómenos, así como generar y demostrar hipótesis. La meta fundamental de esta investigación es generar un conocimiento que sirva para la solución de problemas teóricos – prácticos; es decir que, mediante la implementación de la herramienta de hardware reconfigurable, se proporciona un método alternativo equivalente para la obtención de los valores de las variables de estado en la sincronización de circuitos caóticos. Los datos obtenidos serán sometidos a un análisis estadístico riguroso.

3.4 Alcance de la investigación:

En la presente investigación se pretende realizar un análisis de la sincronización utilizando hardware reconfigurable varios nodos y comparar los resultados con los obtenidos a través de la simulación numérica tradicional utilizando software.

3.5 Técnicas de la investigación:

De acuerdo con el tema planteado destacamos las siguientes técnicas que son necesarias para llevar a cabo la investigación.

3.5.1 Investigación bibliográfica:

La investigación bibliográfica es aplicada a cualquier tipo de investigación debido a que es necesario asegurar la veracidad de los estudios mediante fuentes confiables de información, con esto se asegura la originalidad de los temas investigados. Es así que, es necesario en la investigación del tema planteado puesto que, se parte de información veraz que está confirmada y aprobada.

3.5.2 Estudios correlacionales:

Esta técnica de investigación se refiere a los procedimientos investigativos que tratan de determinar la relación entre dos o más variables de estudio. En consecuencia, en la creación del hardware reconfigurable como herramienta se relacionan varias variables tales como: los resultados de las variables de estado, tiempos de sincronización y análisis. También se tiene la relación de conceptos tales como: sistemas caóticos, circuitos caóticos, sistemas dinámicos. Mediante las relaciones tanto de las variables como de los conceptos mencionados se puede obtener conclusiones al final de la investigación, así como también predicciones para futuras investigaciones.

3.5.3 Estudios causales comparativos a partir de teorías fundamentadas

Esta técnica de investigación abarca dos o más casos de estudio de forma que producen conocimiento más generalizado. En el tema de estudio planteado se propone realizar el estudio comparativo entre las teorías fundamentales de la sincronización de circuitos caóticos con los resultados obtenidos de las variables de estado en el programa de análisis numérico y los resultados obtenidos de las variables de estado en el hardware reconfigurable. Esta comparación implica el análisis y la síntesis de las

similitudes, diferencias y patrones que comparten una meta común. Partiendo de la recolección de datos a través del análisis documental y experimental, se puede relacionar datos tanto cualitativos como cuantitativos obteniendo un análisis teórico-práctico.

3.6 Tamaño de la muestra

Población: En el caso del estudio planteado se tiene una población infinita debido a que se pueden tomar datos de forma infinita.

Muestra: Para la selección de la muestra partimos de la ecuación

$$n = \frac{z^2 pq}{e^2} \quad (3 - 1)$$

Dónde

z: nivel de confianza 95% $z=1,96$

p: probabilidad a favor 50% $p=0,5$

q: Probabilidad en contra 50% $q=0,5$

e= Margen de error 0,1%

$$n = \frac{(1,96)^2(0,5)(0,5)}{(0,01)^2}$$
$$n = 9604$$

En consecuencia, la muestra es de 9604 datos.

Selección de la muestra

La selección de la muestra se desarrollará de forma aleatoria.

3.7 Identificación de variables

Variable independiente: Tiempo

Variable dependiente: Variables de estado

Tabla 1-3. Operacionalización de variables

VARIABLE	TIPO DE VARIABLE	CONCEPTO	INDICADOR	DESCRIPCION	INSTRUMENTOS
Tiempo	Independiente	Representa el incremento temporal entre puntos sucesivos para calcular los valores de a variables de estado en dichos puntos.	Numero	El tiempo está definido por el número de pasos del método numérico utilizado para la resolución de las ecuaciones de estado.	Software, Hardware
Variables de estado	Dependiente	El circuito de Chua consiste de un inductor L, dos capacitores C_1 y C_2 , un diodo de Chua $g(V_1)$ y una resistencia variable R_v	Valor numérico	Dependen del valor inicial, son únicas para cada circuito y con la sincronización tienden a un mismo valor.	Software, Hardware

Realizado por: Donoso R. 2022

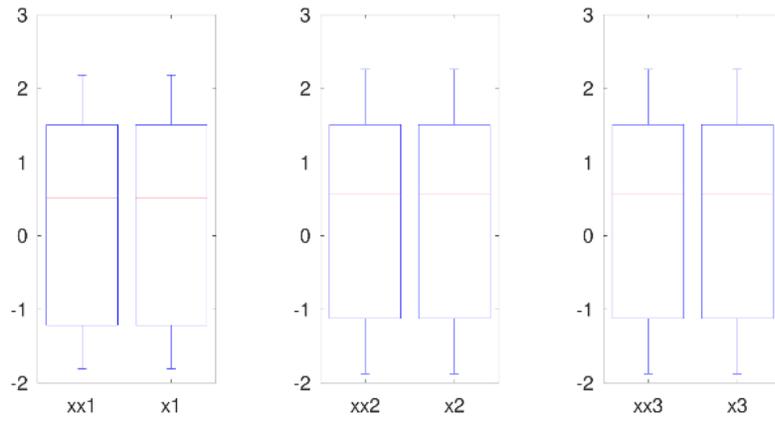
CAPÍTULO IV

4 RESULTADOS Y DISCUSIÓN

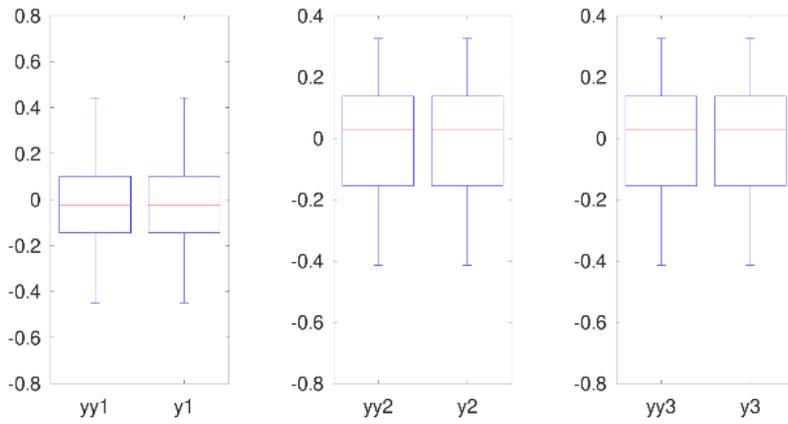
4.1 ANÁLISIS DE RESULTADOS

En esta sección se presentan los resultados obtenidos luego de implementar el circuito de Chua en la FPGA y se comparan con los resultados obtenidos utilizando en el software de simulación numérica octave.

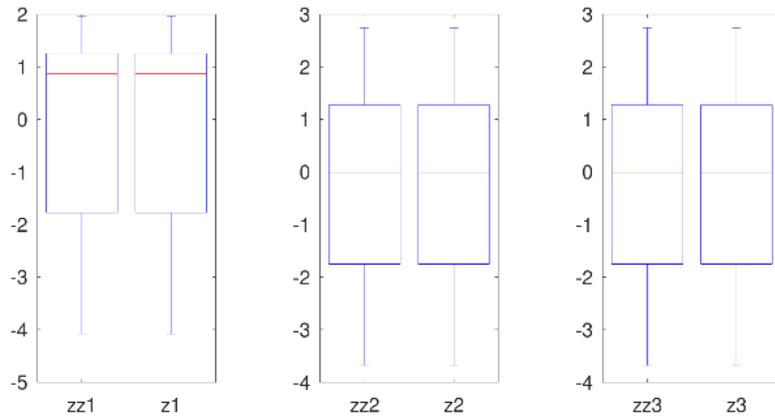
En la gráfica 1-4 (a)(b)(c) se muestran la distribución y asimetría de las variables de estado x , y , z calculadas con el software de simulación y con la FPGA, en estas se puede apreciar que las medianas, la distribución y la simetría son idénticas.



a) x & xx



b) y & yy



c) z & zz

Gráfico 1-4: Diagrama de cajas de las variables de estado, x, y, z (Octave) & xx, yy, zz. (FPGA)

Realizado por: Donoso R. 2022

Para ratificar la fiabilidad de los resultados obtenidos; se llevó a cabo además un análisis de la relación estadística entre los datos obtenidos mediante la FPGA y el software de simulación. El gráfico 2-7 muestra una relación lineal entre las variables x y xx correspondientes al software de simulación y a la FPGA respectivamente. Así mismo, como se puede apreciar en la tabla 1-4, el coeficiente de correlación de Pearson en 1 o muy cercano a 1, esto indica categóricamente que las variables están fuertemente correlacionadas.

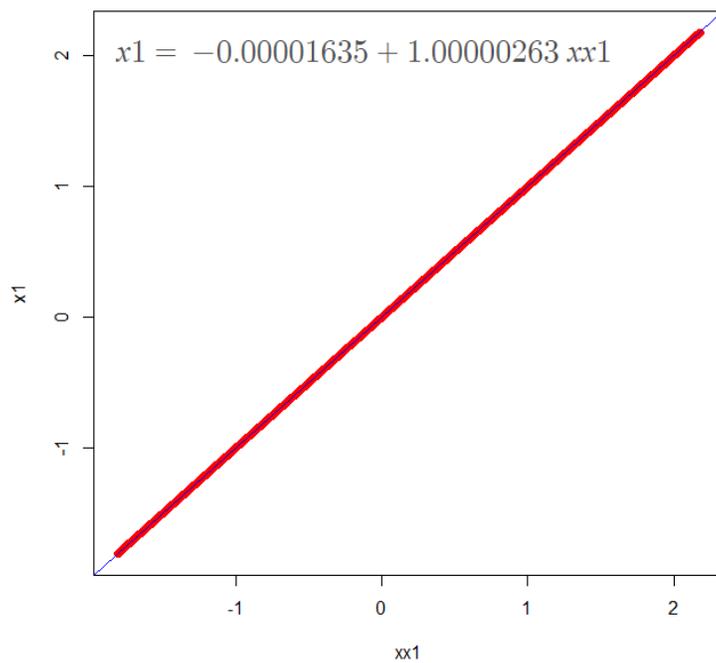


Gráfico 2-4: Gráfica dispersión de la relación lineal xx1 vs x1

Realizado por: Donoso R. 2022

Tabla 1-4. Coeficiente de correlación y modelo lineal de las variables de estado de la FPGA (xx, yy, zz) y del software de simulación (x, y, z).

Variable 1 (x)	Variable 2 (y)	r	Modelo lineal y=b+mx	
			b	m
xx1	x1	1	-0.00001635	1.00000263
xx2	x2	1	-0.00001541	1.00000245
xx3	x3	1	-0.00001541	1.00000245
yy1	y1	0.9999887	0.00001353	1.00001967
yy2	y2	0.9999498	0.00002214	1.00002846
yy3	y3	0.9999498	0.00002214	1.00002846
zz1	z1	0.9999788	0.0001184	1.0000158
zz2	z2	0.9999925	0.00005549	1.00001217
zz3	z3	0.9999925	0.00005549	1.00001217

Realizado por: Donoso R. 2022

Finalmente, la figura 1-4, muestra el porcentaje de recursos utilizados en la FPGA luego de la síntesis e implementación de tres circuitos de Chua, el mismo que no supera el 4% es decir la FPGA utilizada tiene una capacidad de resolver aproximadamente 75 nodos.

Device Utilization Summary					I-1
Logic Utilization	Used	Available	Utilization	Note(s)	
Number of Slice Flip Flops	29	9,312	1%		
Number of 4 input LUTs	16	9,312	1%		
Number of occupied Slices	25	4,656	1%		
Number of Slices containing only related logic	25	25	100%		
Number of Slices containing unrelated logic	0	25	0%		
Total Number of 4 input LUTs	46	9,312	1%		
Number used as logic	16				
Number used as a route-thru	30				
Number of bonded IOBs	3	232	1%		
Number of BUFGMUXs	1	24	4%		
Average Fanout of Non-Clock Nets	2.56				

Figura 1-4: Recursos utilizados por la FPGA Spartan 3E luego de la síntesis e implementación de tres circuitos de Chua.

Realizado por: Donoso R. 2022

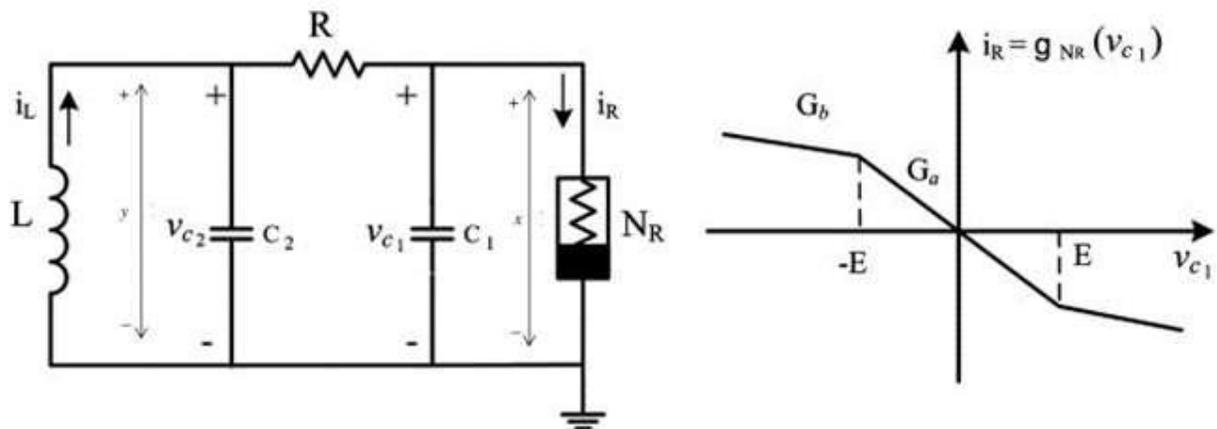
CAPÍTULO V

5 PROPUESTA

5.1 SINCRONIZACIÓN DEL CIRCUITO DE CHUA

El concepto de sincronización caótica considera que tanto el emisor como el receptor son dos sistemas caóticos y oscilan de manera sincronizada. Varios enfoques diferentes, incluidos algunas técnicas de control lineal convencionales y esquemas de control no lineales avanzados ya han sido aplicados con éxito a problemas de sincronización. Esto puede parecer extraño para los sistemas caóticos, ya que las trayectorias de dos sistemas idénticos con casi condiciones iniciales idénticas divergen exponencialmente. Sin embargo, es posible que los sistemas caóticos se sincronicen entre sí bajo ciertas condiciones.

El circuito caótico simple es descrito por el circuito de Chua de la figura 1-5. Tiene tres señales 'x', 'y' y 'z' llamadas variables que representan el voltaje en el capacitor C1, el voltaje en el capacitor C2 y el corriente a través del inductor respectivamente



Figuras 1-5. Circuito Caótico de Chua

Realizado por: Donoso R. 2022

5.2 Planteamiento del modelo

5.2.1 Modelo de un circuito de Chua.

Se parte de las ecuaciones diferenciales que describen el circuito de Chua las mismas que se escriben a continuación.

$$C_1 \frac{dV_1}{dt} = \frac{1}{R_p}(V_2 - V_1) - gV_1 \quad (5-1)$$

$$C_2 \frac{dV_2}{dt} = \frac{1}{R}(V_1 - V_2) + I_L \quad (5-2)$$

$$L \frac{dI_L}{dt} = -V_2 \quad (5-3)$$

Según (Chen et al., 2015) las ecuaciones normalizadas del circuito de Chua se expresan de la siguiente manera.

$$\dot{x} = \alpha [y - x - h(x)] \quad (5-4)$$

$$\dot{y} = x - y + z \quad (5-5)$$

$$\dot{z} = -\beta y \quad (5-6)$$

Dónde:

$$h(x) = m_1 x + \frac{1}{2}(m_0 - m_1)[|x + 1| - |x - 1|] \quad (5-7)$$

Siendo m_1, m_2, y constantes reales positivas.

5.2.2 Modelo de sincronización de dos circuitos de Chua.

Tomando en cuenta que dos circuitos caóticos nunca tendrán las tres señales exactamente iguales en ningún momento, debido a la propiedad de "sensibilidad a las condiciones iniciales", en otras palabras, los circuitos nunca serán naturalmente sincronizado; se utilizan muchas formas de acoplar un sistema para la sincronización, pero las dos más populares son:

- a) Acoplamiento direccional (o excitación-respuesta), cuando el circuito excitador controla al esclavo (o respuesta).
- b) Acoplamiento no direccional cuando ambos circuitos se conectan entre sí, lo influyen para que se encienda.

La figura 2-5, muestra dos circuitos acoplados no direccionales a través de la variable de estado 'x' y una resistencia de acople R_c .

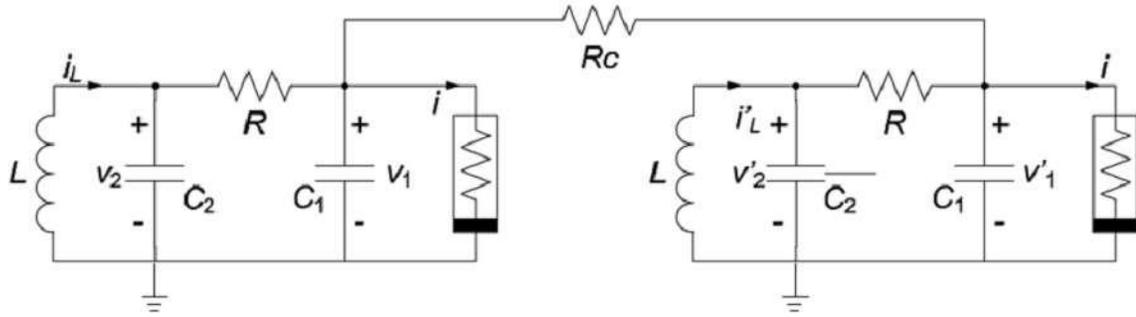


Figura 2-5. Dos circuitos de Chua acoplados a través de la variable x

Realizado por: Donoso R. 2022

Por lo tanto, se obtiene las siguientes ecuaciones:

Circuito de Chua 1

$$\dot{x} = \alpha(y - x - g(x)) + \delta_x(x' - x) \quad (5-8)$$

$$\dot{y} = x - y + z \quad (5-9)$$

$$\dot{z} = -\beta y \quad (5-10)$$

Circuito de Chua 2

$$\dot{x}' = \alpha(y' - x' - g(x')) + \delta_x(x - x') \quad (5-11)$$

$$\dot{y}' = x' - y' + z' \quad (5-12)$$

$$\dot{z}' = -\beta y' \quad (5-13)$$

Dónde

$$\delta_x = \frac{R\alpha}{R_c} \quad (5-14)$$

Cuando $\delta > 5,56$ los circuitos llegan a sincronizarse.

5.2.3 Modelo de sincronización de n circuitos de Chua

Considerando n circuitos de Chua como se muestra en la figura 3-5, las ecuaciones de estados ‘y’ y ‘z’ no tiene ningún efecto en la sincronización y el estado ‘x’ para el segundo circuito se puede calcular de la siguiente manera:

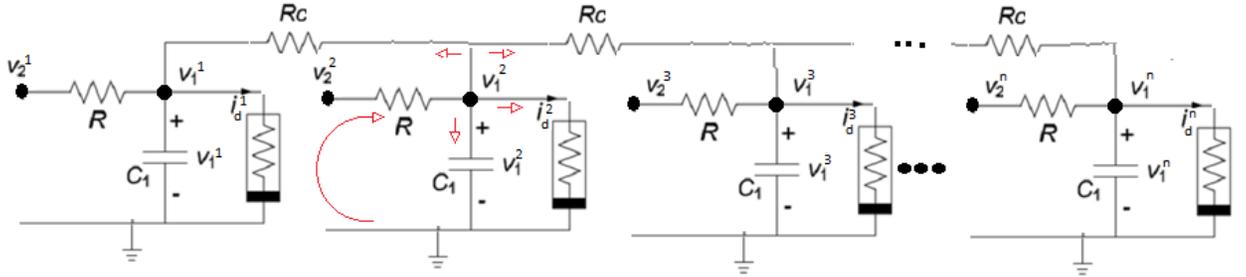


Figura 3-5. N circuitos de Chua acoplados a través de la variable x

Realizado por: Donoso R. 2022

Las flechas rojas muestran e flujo de corriente y para el KCL

$$\frac{V_2^2 - V_1^2}{R} = i_d^2 + C_1 \frac{dV_1^2}{dt} + \frac{V_1^2 - V_1^1}{R_c} + \frac{V_1^2 - V_3^1}{R_c} \quad (5 - 15)$$

$$C_1 \frac{dV_1^2}{dt} = \frac{V_2^2 - V_1^2}{R} - i_d^2 - \frac{V_1^2 - V_1^1}{R_c} - \frac{V_1^2 - V_3^1}{R_c} \quad (5 - 16)$$

$$\frac{dV_1^2}{dt} = \frac{V_2^2 - V_1^2}{RC_1} - \frac{i_d^2}{C_1} - \frac{1}{R_c C_1} (2V_1^2 - V_3^2 - V_3^1) \quad (5 - 17)$$

De manera similar a los dos circuitos de Chua, es posible realizar transformaciones de parámetros y encontrar el adimensional de la ecuación para x:

$$\dot{X}^2 = \alpha(y^2 - x^2 - g(x^2)) - \delta(2x^2 - x^1 - x^3) \quad (5 - 18)$$

Dónde $\delta = \frac{R\alpha}{R_c}$

En general, para una red de n circuitos de Chua, las j-ésimas ecuaciones de estado son:

$$\dot{X}^j = \alpha(y^j - x^j - g(x^j)) - \delta(2x^j - x^{j-1} - x^{j+1}) \quad (5 - 19)$$

$$\dot{Y}^j = x^j - y^j + z^j \quad (5 - 20)$$

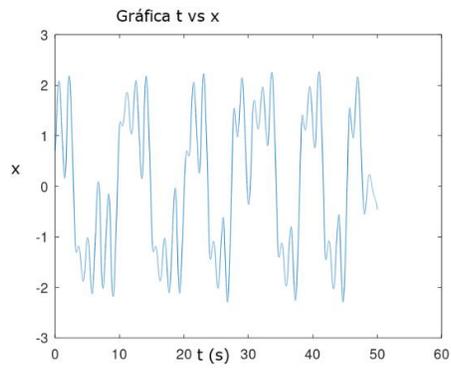
$$\dot{Z}^j = -\beta z^j \quad (5 - 21)$$

Tomando en cuenta que $g(x^j) = m_1 x^j + 0.5(m_0 - m_1)(|x^j + 1| - |x^j - 1|)$ (5 – 22)

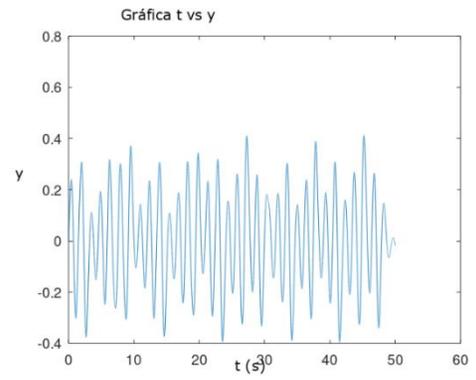
5.3 Implementación del circuito de Chua en Octave

Con base en el modelo matemático desarrollado para un circuito de Chua y considerando las ecuaciones de estado normalizadas, bajo los siguientes parámetros: $\alpha = 15.6$, $\beta = 28$, $m_0 = -1.143$, $m_1 = -0.714$ y condiciones iniciales $x_0 = 0.7$, $y_0 = 0$, $z_0 = 0$, se desarrolló un script en Octave que resuelve el sistema dinámico del circuito de Chua utilizando el método de Euler con un tamaño de paso de 0,005 y un total de 10000 muestras.

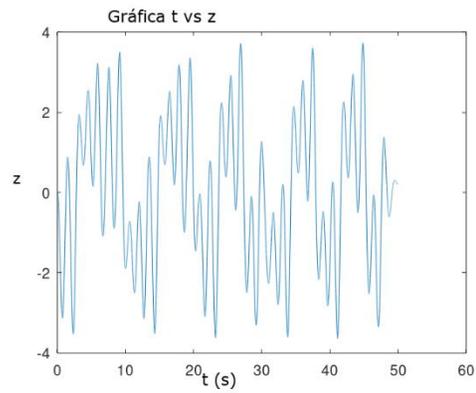
Las gráficas 1-5 (a)(b)(c), muestran la representación gráfica de las variables de estado x , y , z en función del tiempo, ratificando que se trata de un sistema dinámico caótico.



a) t vs x



b) t vs y

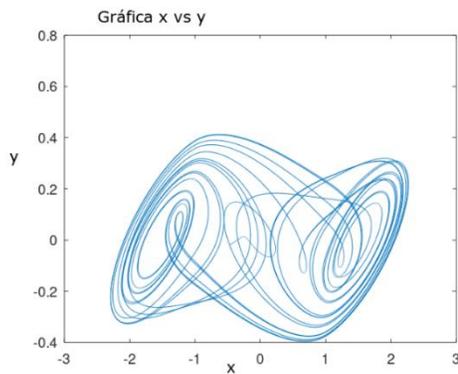


c) t vs z

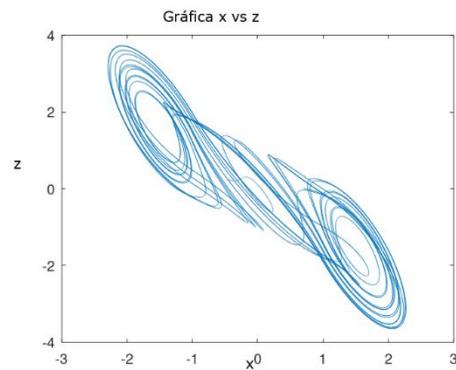
Gráfico 1-5: Variables de estado en función del tiempo

Realizado por: Donoso R. 2022

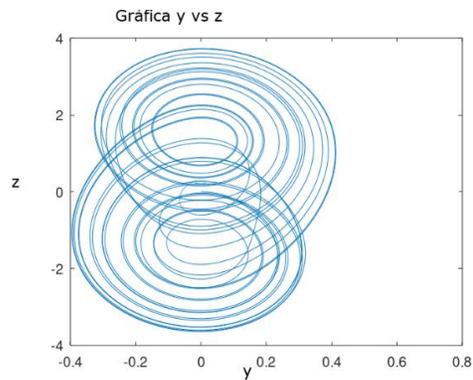
Las gráficas 2-5 (a)(b)(c), representan la relación que existe entre las variables de estado x , y , z , evidenciando que la trayectoria forma un atractor como era de esperarse.



a) x vs y



b) x vs z



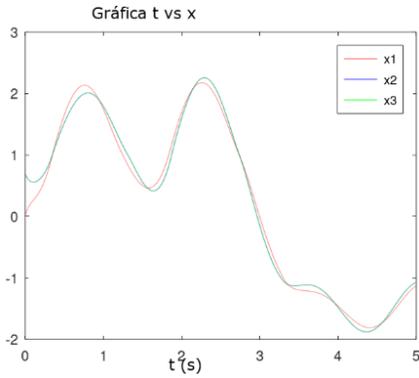
c) y vs z

Gráfico 2-5. Relación de las variables de estado x , y , z

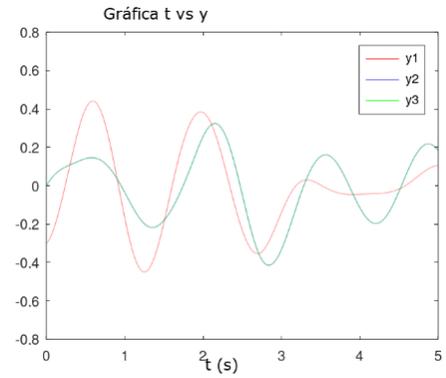
Realizado por: Donoso R. 2022

De la misma forma con base a la ecuación (6-22) para n circuitos de Chua y considerando los mismos parámetros mencionados anteriormente, se escribió un script para resolver las ecuaciones de estado correspondientes a la sincronización de tres circuitos de Chua.

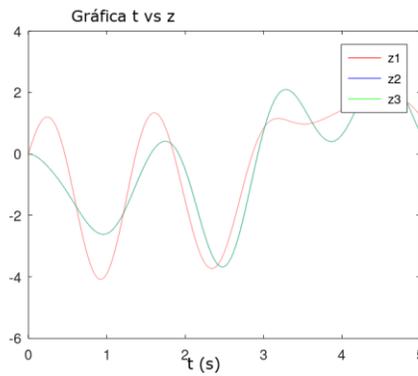
Las gráficas 3-5 (a)(b)(c), representan el comportamiento de las x_s , y_s y las z_s en función del tiempo. Visiblemente se puede observar que las variables tienden a sincronizarse (tienden al mismo valor) a medida que el tiempo transcurre.



a) t vs x_{ss}



b) t vs y_{ss}



c) t vs y_{ss}

Grafico 3-5. Variables de estado de tres nodos acoplados en función del tiempo

Realizado por: Donoso R. 2022

5.4 Implementación del circuito de Chua en la FPGA.

La implementación del circuito de Chua en el hardware reconfigurable se llevó a cabo con la ayuda de una herramienta de síntesis XILINKS ISE versión demo, se utilizó lenguaje de descripción de hardware Verilog y se cargó en una Spartan 3 E FPGA, figura 4-5.



Figura 4-5: Spartan 3E FPGA

Realizado por: Donoso R., 2022

El gráfico 4-5 muestra la arquitectura RTL del sistema implementado. Consta de dos elementos principales, el circuito de Chua y un divisor de frecuencia. El circuito de Chua resuelve las ecuaciones de estado x , y , z aplicando el método de Euler con un tamaño de paso de 0.005 mientras que, el divisor de frecuencia activa su salida cuando se han procesado un total de 10000 muestras. La frecuencia del sistema es de 50 Mhz.

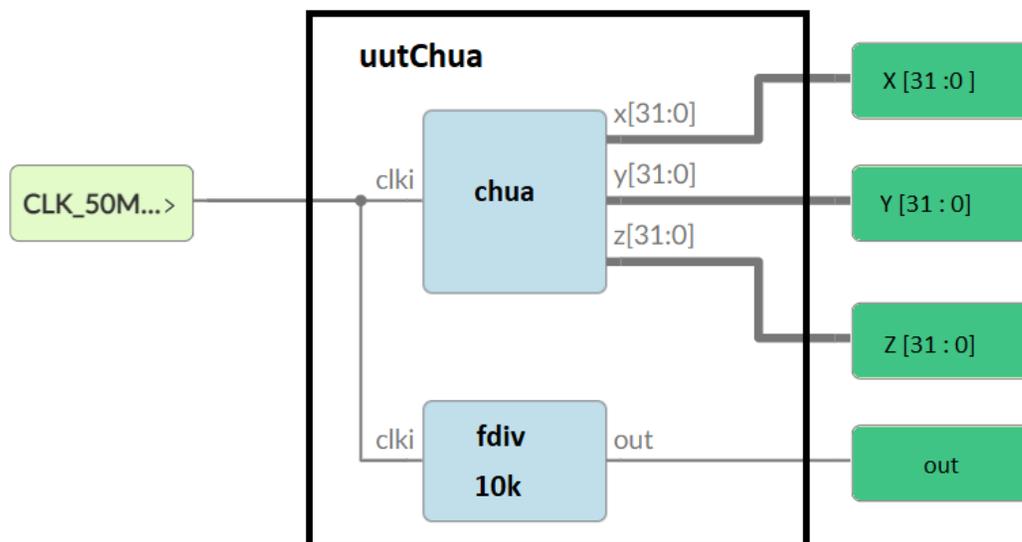


Gráfico 4-5. RTL del sistema implementado en la FPGA para un circuito de Chua.

Realizado por: Donoso R., 2022

En el gráfico 5-5, se visualiza el comportamiento de las variables de estado x_1 , x_2 , x_3 con respecto al tiempo; su comportamiento es similar al de un circuito dinámico caótico.

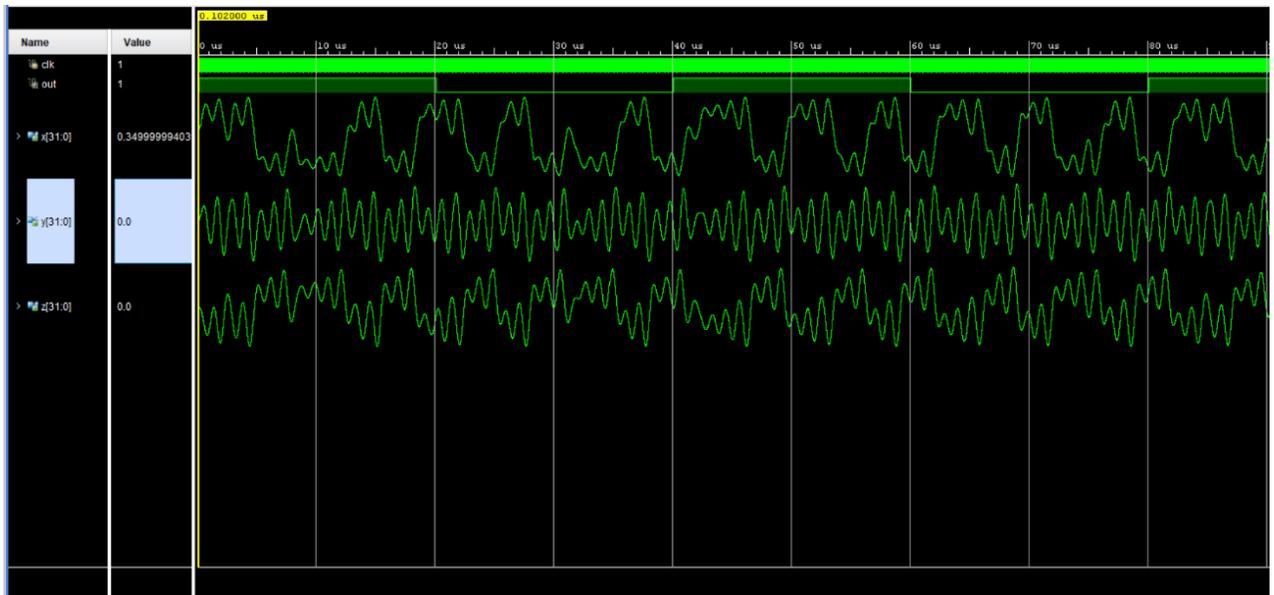


Gráfico 5-5. Variables de estado en función del tiempo obtenidas en la FPGA

Realizado por: Donoso R. 2022

5.5 Análisis del modelo para n circuitos de Chua usando FPGA.

El gráfico 6-5 muestra la arquitectura RTL creada en la FPGA para tres nodos sincronizados entre sí. Consta de cuatro elementos principales, tres circuitos de Chua y un divisor de frecuencia. Los tres circuitos de Chua resuelven las ecuaciones de estado x , y , z al mismo tiempo aplicando el método de Euler con un tamaño de paso de 0.005, mientras que, el divisor de frecuencia activa su salida cuando se han procesado un total de 10000 muestras. Utiliza la misma frecuencia mencionada anteriormente.

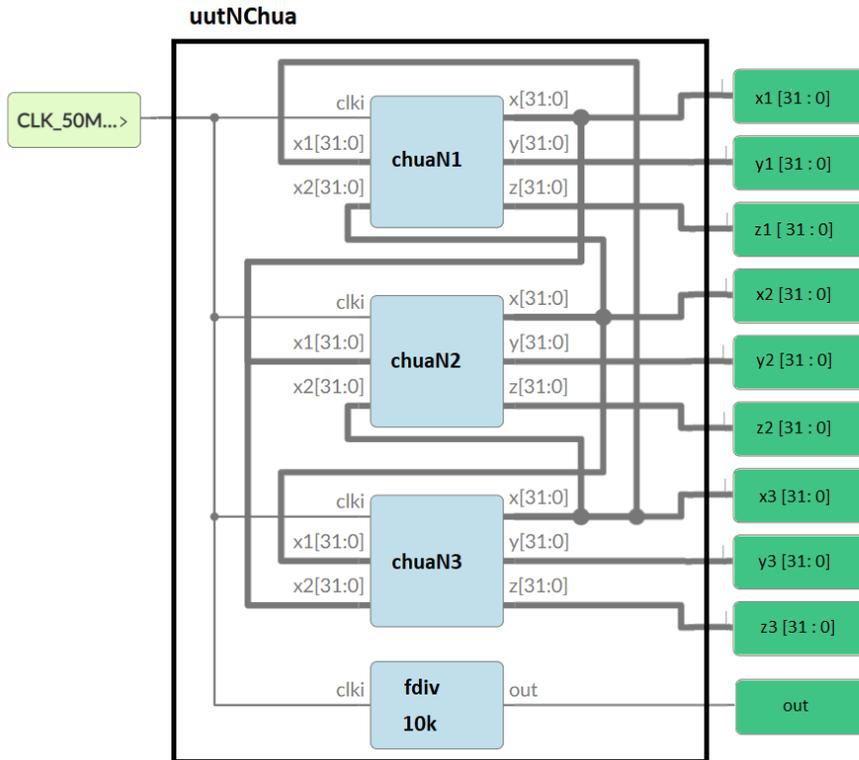


Gráfico 6-5. RTL del sistema implementado en la FPGA para tres circuitos de Chua. $\alpha=15$, $\beta=28$, $m_1=-1.143$, $m_0=-0.714$, $x_1=0.7$, $x_2=0$, $x_3=0$.

Realizado por: Donoso R. 2022

En el gráfico 7-5, se puede observar el comportamiento de las variables de estado x_1 , x_2 , x_3 correspondientes a los nodos 1,2,3 en función del tiempo respectivamente. Del mismo modo se puede apreciar que los valores tienden a sincronizarse a medida que avanza el tiempo

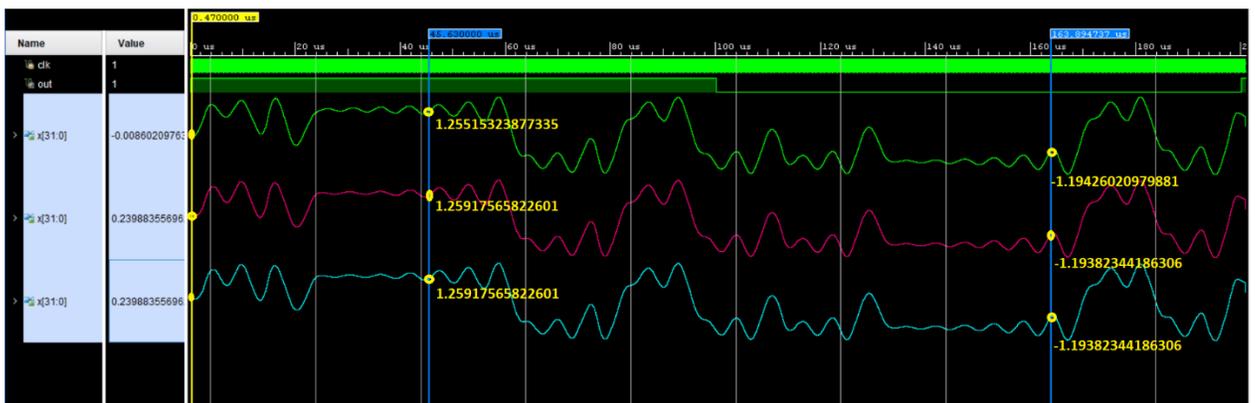
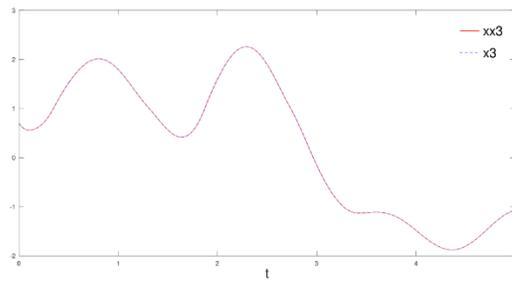
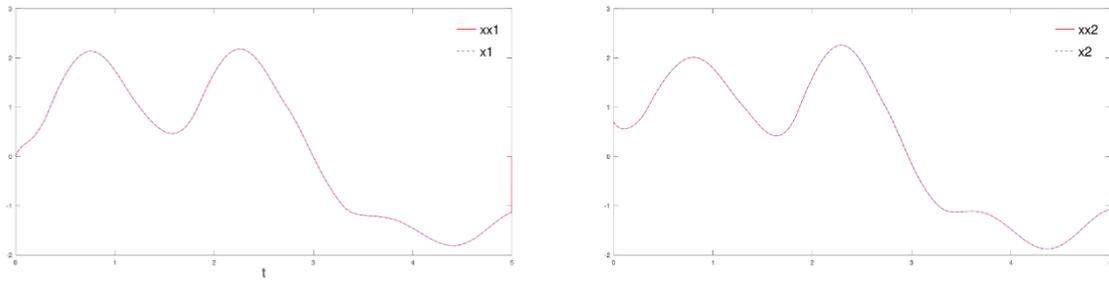


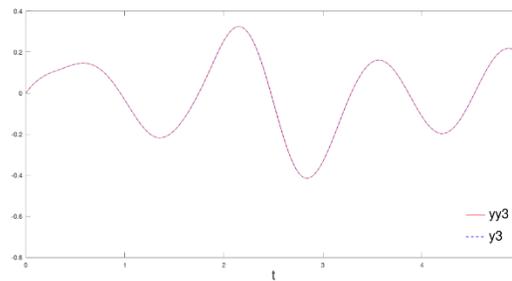
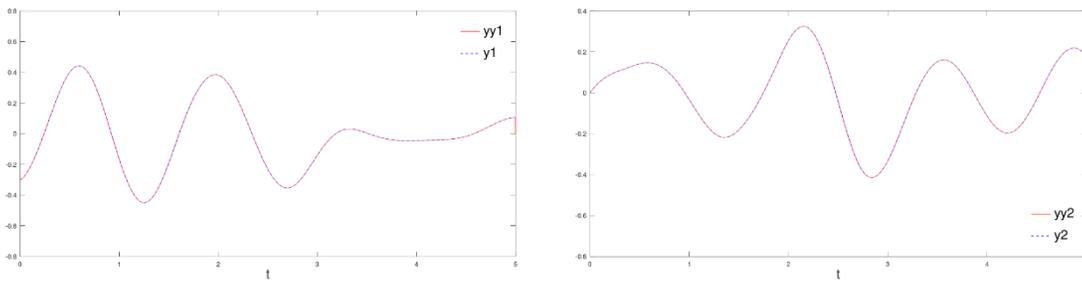
Gráfico 7-5. Variables de estado x_1 , x_2 , x_3 con respecto al tiempo

Realizado por: Donoso R. 2022

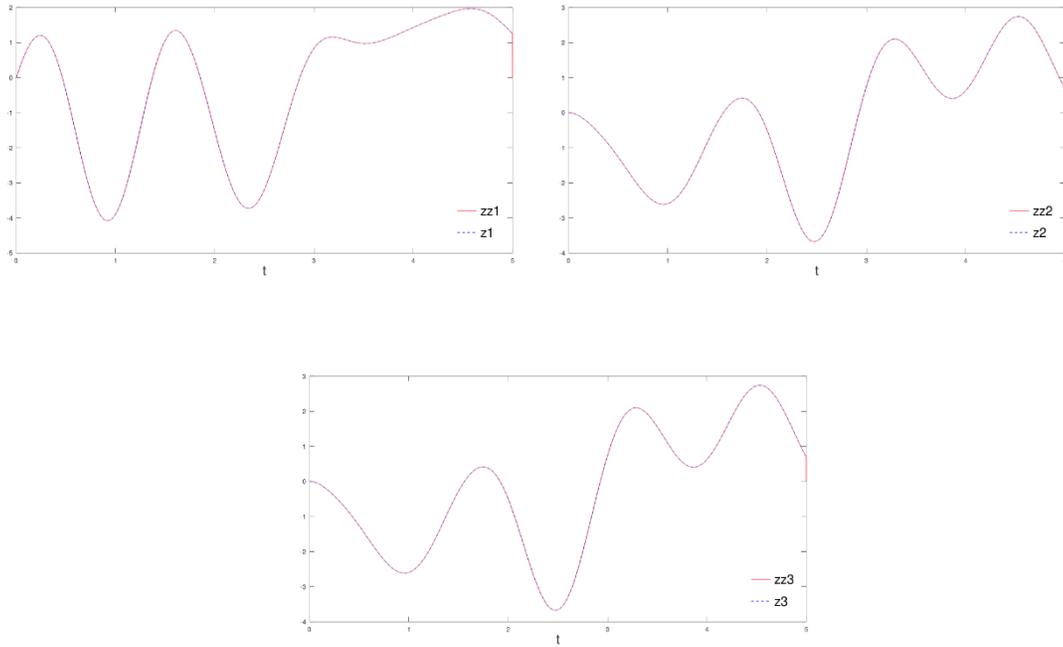
Finalmente, las gráficas 8-5 (a)(b)(c) se muestra el comportamiento temporal de las variables de estado obtenidas mediante la FPGA y el software de cálculo. Gráficamente se puede apreciar que no hay diferencias significativas entre los cálculos realizados con el software y el dispositivo electrónico.



a) t vs xs



b) t vs ys



c) t vs zs

Gráfico 8-5. Relación entre variables de estado calculadas con la FPGA (xx , yy , zz) y las calculadas con Octave (x , y , z).

Realizado por: Donoso R. 2022

CONCLUSIONES

A través de esta investigación se pudo estudiar el fenómeno de sincronización de varios circuitos de Chua utilizando hardware reconfigurables y los resultados obtenidos con la FPGA muestran un comportamiento caótico como se esperaba en este tipo de sistemas. La investigación se desarrolló de una manera sistemática, cumpliendo con el objetivo general, diseñando e implementado un sistema basado en hardware reconfigurable como una herramienta de estudio de circuitos caóticos. Los datos obtenidos fueron comparados estadísticamente con los resultados de la simulación numérica los mismos que no mostraron una diferencia significativa, es decir están altamente correlacionados. Esta herramienta ayudará a los estudiantes del Instituto Técnico Superior Carlos Cisneros en sus investigaciones posteriores acerca del sistema dinámicos.

RECOMENDACIONES

Experimentar este sistema con nuevos modelos matemáticos del circuito de Chua, diferentes al utilizado en este estudio.

Utilizar el sistema para verificar su funcionamiento dentro de una aplicación en específico por ejemplo en comunicaciones, redes neuronales o codificaciones.

GLOSARIO

A

Adimensional

Conjunto de variables que no tienen dimensiones 37

B

Bifurcación

División de una cosa en dos ramales, brazos o puntas.. 2

Bit

Unidad mínima de información 23

C

Comportamientos caóticos

Sistema determinístico regido por ecuaciones diferenciales que presenta un comportamiento dinámico aparentemente aleatorio..... 1

Compuertas lógicas

Son circuitos electrónicos conformados internamente por transistores que se encuentran con arreglos especiales con los que otorgan señales de voltaje como resultado o una salida de forma booleana, están obtenidos por operaciones lógicas binarias..... 19

Correlacionados

Correspondencia o relación recíproca que se da entre dos o más objetos de estudio. 4

F

Fenómenos artificiales

Cambios que son provocados por la acción humana..... 1

Fenómenos naturales

Cambios que no son provocados por la acción humana directamente 1

I

Inteligencia artificial

Se refiere a sistemas que imitan la inteligencia humana para realizar una tarea.....2

M

Modelado

Es una representación simplificada, a través de ecuaciones, funciones o fórmulas matemáticas.....6

O

Optimizar

Buscar la mejor manera de realizar una tarea.....7

P

Principio de superposición

Herramienta matemática que permite descomponer un problema lineal o de otro tipo en dos o más subproblemas más sencillos10

R

Recta tangente

Es la recta que toca a la función en un solo punto.....16

Redes complejas

Conjunto de de nodos y enlaces difíciles de comprender2

S

Solución homogénea

Cuando no se pueden distinguir sus componentes.....11

BIBLIOGRAFÍA

- Amicos, D. E. S. D. I. N. (2011).** Aplicación de las técnicas de hardware reconfigurable en un sistema digital de control dinámico.
- Arango, R., Navarro, Á. A., & Padilla B., J. B. (2014).** Sistemas Open Hardware Y Open Source Aplicados a La Enseñanza De La Electrónica. *Revista de Investigaciones Universidad Del Quindío*, 25(1), 126–133. <https://doi.org/10.33975/riuq.vol25n1.163>
- Bilotta, E., Chiaravalloti, F., & Pantano, P. (2014).** Spontaneous synchronization in two mutually coupled memristor-based Chua's circuits: Numerical investigations. *Mathematical Problems in Engineering*, 2014. <https://doi.org/10.1155/2014/594962>
- Carrasquilla-Batista, A., Chacón-Rodríguez, A., Núñez-Montero, K., Gómez-Espinoza, O., Valverde-Cerdas, J., & Guerrero-Barrantes, M. (2016).** Regresión lineal simple y múltiple: aplicación en la predicción de variables naturales relacionadas con el crecimiento microalgal. *Revista Tecnología En Marcha*, 29(8), 33. <https://doi.org/10.18845/tm.v29i8.2983>
- Castro, A. F. H. (2013).** *Introducción al Octave Índice*. 1–25.
- Chen, C., Ma, H., Chen, H., Meng, Y., & Ding, Q. (2015).** FPGA implementation of a UPT chaotic signal generator for image encryption. *Pacific Science Review A: Natural Science and Engineering*, 17(3), 97–102. <https://doi.org/10.1016/j.psra.2016.02.001>
- Esther, M., & Garaluz, G. (2014).** *Modelado matemático de sistemas dinámicos en epidemiología*.
- Físicas, F. D. C., Nacional, U., & San, M. De. (2016).** *Experimenting the Chua circuit Diseño*. 161902752, 1–5.
- González, I., González, J., & Gómez Arribas, F. (2003).** Hardware Libre : Clasificación y Desarrollo de Hardware Reconfigurable en Entornos GNU / Linux. *Universidad Autónoma de Madrid, Spain*, 1–13. <http://www.ii.uam.es>
- Guzman, D., Albarrán, F. R., Alexis, E., & Diego, G. (n.d.).** *Método de Euler Método de Euler*.
- LUIS RICAROD OROZCO VILLARRUEL. (2019).** Proyecto De Investigación Previo a La Obtención Del. *Universidad Estatal De Milagro*, 135. <http://repositorio.unemi.edu.ec/bitstream/123456789/5286/1/TESIS>
- Molina, L. R. (n.d.).** *Regresión lineal múltiple*.
- Nari, J. P., & Chua, E. De. (n.d.).** *circuito de Chua a la seguridad en comunicaciones Índice general*. M.
- Sandoval Ruiz, C., & Fedón, A. (2008).** Programación VHDL de algoritmos de codificación para dispositivos de hardware reconfigurable. *Revista Internacional de Métodos Numéricos Para Cálculo y Diseño En Ingeniería*, 24(4), 3–11.
- Torres, M. (n.d.).** *Matemáticas en Ingeniería con Matlab y Octave Release 0 . 1 Guillem Borrell i*

Nogueras et al Matemáticas en Ingeniería con Matlab y Octave.

Uito, C., En, D. E. C., Sistemas, S. D. E. L. O. S., Martínez, A., Berenice, M., & Francisco, Á. G.

(2013). *“Circ Uito De Chua En La Sincronización De Los Sistemas Caoticos.”*

Zadunaisky, P. E., & Dinámicos, S. (2009). *Fascículo 5.*

ANEXOS

Anexo A: Código fuente para el análisis de un circuito de Chua en el programa Octave

```
1. clc
2. clear all
3. tic
4. dt=0.005;
5. x(1) = 0.7;
6. y(1) = 0;
7. z(1)= 0;
8. t(1)=0;
9. N=9999;
10. alpha = 15.6;
11. beta = 28;
12. m0 = -1.143;
13. m1 = -0.714;
14. for i=1:N
15.   h = m1*x(i)+0.5*(m0-m1)*(abs(x(i)+1)-abs(x(i)-1));
16.   xdot = alpha*(y(i)-x(i)-h);
17.   ydot = x(i) - y(i)+ z(i);
18.   zdot = -beta*y(i);
19.   x(i+1)=x(i)+xdot*dt;
20.   y(i+1)=y(i)+ydot*dt;
21.   z(i+1)=z(i)+zdot*dt;
22.   t(i+1)=t(i)+dt;
23. end
24. toc
25. plot(x,y)
```

Anexo B: Código fuente para el análisis de la sincronización de tres circuitos de Chua en el programa Octave

```
1. clc
2. figure
3. tic
4. alpha = 15;
5. beta = 28;
6. fx=1;
7. n=3;
8. x=y=z=zeros(1,n);
9. x=y=z=zeros(1,n);
10. x=y=z=zeros(1,n);
11. y(1)=-0.3*fx;
12. x=[0.0 0.7*fx 0.7*fx];
13. dx=7*fx;
14. dt=0.0005;
15. function hf=h(x)
16. fx=1;
17. m0 = -1.143*fx;
18. m1 = -0.714*fx;
19. m0mm1x05 = -0.2145 ;
20. hf=m1*x+m0mm1x05*(abs(x+1)-abs(x-1));
21. endfunction
22. N=9999;
23. t(1)=0;
24. for j=1:N
25.     for i=1:n
26.         if i==1
27.             im1=n;
28.         else
29.             im1=i-1;
30.         end
31.         if i==n
32.             ip1=1;
33.         else
34.             ip1=i+1;
35.         end
36.         xdot(1,i) = alpha*(y(j,i)-x(j,i)-h(x(j,i)))-dx*(2*x(j,i)-x(j,im1)-
            x(j,ip1));
37.         ydot(1,i) = x(j,i) - y(j,i) + z(j,i);
```

```
38.         zdot(1,i) = -beta*y(j,i);
39.     end
40.     x(j+1,:)=x(j,:)+dt*xdot;
41.     y(j+1,:)=y(j,:)+dt*ydot;
42.     z(j+1,:)=z(j,:)+dt*zdot;
43.     t(j+1)=t(j)+dt;
44. end
45. toc
46. tx=1:length(t)
47. plot(t,x(:,1),"r;x1;",t,x(:,2),"b;x2;",t,x(:,3),"g;x3;")
48. h = legend ("location", "northeastoutside");
49. set (h, "fontsize", 10);
```

Anexo C : Módulo verilog del divisor de frecuencia, fdiv.v

```
`timescale 1ns / 1ps
module fDiv(clock_in, clock_out
    );
input clock_in; // input clock on FPGA
output reg clock_out; // output clock after dividing the input clock by divisor
reg[27:0] counter=28'd0;
parameter DIVISOR = 28'd10000;

always @(posedge clock_in)
begin
    counter <= counter + 28'd1;
    if(counter>=(DIVISOR-1))
        counter <= 28'd0;
    clock_out <= (counter<DIVISOR/2)?1'b1:1'b0;
end
endmodule
```

Anexo D: Modulo verilog de un circuito de chua , chua.v

```
//`timescale <time_units> / <precision>

module chua (clk,x, y, z);
    parameter integerBits    = 6;
    parameter fractionBits   = 25;
    parameter dtBits        = 16;
    parameter dtShift        = 30;
    parameter iteratorBits   = 16;
    parameter signed [integerBits + fractionBits : 0] dt=0.005 *(2.0 ** (fractionBits));
    parameter [iteratorBits - 1 : 0] skip=1024;
    parameter signed [integerBits + fractionBits : 0] sigma =      15.0 * 1*(2.0 **
(fractionBits));
    parameter signed [integerBits + fractionBits : 0] rho   =      1.0 * 1*(2.0 **
(fractionBits));
    parameter signed [integerBits + fractionBits : 0] beta  =      28.0 * 1*(2.0 **
(fractionBits));
    // parameter signed [integerBits + fractionBits : 0] m0   =      -1.143 * (2.0 **
$itor(fractionBits));
    parameter signed [integerBits + fractionBits : 0] m1    =      -0.714 * 1*(2.0 **
(fractionBits));
    parameter signed [integerBits + fractionBits : 0] m0mm1x05 =      -0.2145 *1* (2.0 **
(fractionBits));

    parameter signed [integerBits + fractionBits : 0] startX =      0.7 * 1*(2.0 **
fractionBits);
    parameter signed [integerBits + fractionBits : 0] startY =      0.0 * 1*(4.0 **
fractionBits);
    parameter signed [integerBits + fractionBits : 0] startZ =      0.0 * 1*(2.0 **
fractionBits);

    localparam totalBits = 1 + integerBits + fractionBits;

    input wire clk;

    //input wire [iteratorBits - 1 : 0] skip;
    output reg signed [totalBits - 1 : 0] x = startX;
    output reg signed [totalBits - 1 : 0] y = startY;
    output reg signed [totalBits - 1 : 0] z = startZ;
    //input wire [iteratorBits - 1 : 0] skip;
    //output reg signed [totalBits - 1 : 0] ab1 ;
    //output reg signed [totalBits - 1 : 0] ab2;
    //output reg signed [totalBits - 1 : 0] ho;

    reg [iteratorBits - 1 : 0] iterator = 0;
    reg signed [totalBits - 1 : 0] a = startX;
    reg signed [totalBits - 1 : 0] b = startY;
    reg signed [totalBits - 1 : 0] c = startZ;
    reg signed [totalBits * 2 - 1 : 0] h = 0;
```

```

reg signed [totalBits - 1 : 0] abs1 = 0;
reg signed [totalBits - 1 : 0] abs2 = 0;
reg signed [totalBits * 2 - 1 : 0] dxdt = 0;
reg signed [totalBits * 2 - 1 : 0] dydt = 0;
reg signed [totalBits * 2 - 1 : 0] dzdt = 0;
    reg signed [totalBits - 1 : 0] res = 0;
//absSum absp(x, rho, abs1);
//absSub absm(x, rho, abs2);

always @(posedge clk)
begin
    iterator <= iterator + 1;

    if (!iterator)
        begin
            x = a;
            y = b;
            z = c;
        end
    else
        begin
            if (iterator == skip)
                begin
                    a = x;
                    b = y;
                    c = z;
                end
            else
                begin
                    abs1=x+32'h04000000;
                    if(abs1<0) begin
                        abs1=-abs1;
                    end
                    abs2=(x-32'h04000000);
                    if(abs2<0) begin
                        abs2=-abs2;
                    end
                    end

            h=(m1*x+m0mm1x05*(abs1-abs2))>>>fractionBits;
                //ho=h;
                //ab1=abs1;
                //ab2=abs2;

            dxdt = (sigma * (y - x-h)) >>> fractionBits;
            dydt = x -y+z;
            dzdt = (-beta* y) >>> fractionBits;

            x = x + ((dxdt * dt) >>> fractionBits);
            y = y + ((dydt * dt) >>> fractionBits);
            z = z + ((dzdt * dt) >>> fractionBits);

        end
    end
endmodule

```

Anexo E: Módulo verilog de la unidad bajo prueba de un circuito de chua, uutChua.v

```
`timescale 1ns / 1ps
module uutChua(
    input CLK_50MHZ,
    output y
);

    fDiv divisor(CLK_50MHZ,y);
    chua chual(CLK_50MHZ);

endmodule
```

Anexo F: Módulo Verilog de un circuito de Chua con entradas de sincronización, chuaN.v

```
`timescale 1ns / 1ps
module chuaN(clk,x1,x2,x,y,z);
    parameter integerBits = 6;
    parameter fractionBits = 25;
    parameter dtBits = 16;
    parameter dtShift = 30;
    parameter iteratorBits = 16;
    parameter signed [integerBits + fractionBits : 0] dt=0.0005 *(2.0 ** (fractionBits));
    parameter [iteratorBits - 1 : 0] skip=1024;
    parameter signed [integerBits + fractionBits : 0] sigma = 15.0 * 1*(2.0 **
(fractionBits));
    parameter signed [integerBits + fractionBits : 0] rho = 1.0 * 1*(2.0 **
(fractionBits));
    parameter signed [integerBits + fractionBits : 0] beta = 28.0 * 1*(2.0 **
(fractionBits));
    // parameter signed [integerBits + fractionBits : 0] m0 = -1.143 * (2.0 **
$itor(fractionBits));
    parameter signed [integerBits + fractionBits : 0] m1 = -0.714 * 1*(2.0 **
(fractionBits));
    parameter signed [integerBits + fractionBits : 0] m0mm1x05 = -0.2145 *1* (2.0 **
(fractionBits));

    parameter signed [integerBits + fractionBits : 0] startX = 0.7 * 1*(2.0 **
fractionBits);
    parameter signed [integerBits + fractionBits : 0] startY = 0.0 * 1*(2.0 **
fractionBits);
    parameter signed [integerBits + fractionBits : 0] startZ = 0.0 * 1*(2.0 **
fractionBits);
    parameter signed [integerBits + fractionBits : 0] deltaX = 7 * (2.0 **
(fractionBits));
    parameter signed [integerBits + fractionBits : 0] uno = 1 * (2.0 **
(fractionBits));

    localparam totalBits = 1 + integerBits + fractionBits;

    input wire clk;
    input signed [totalBits - 1 : 0] x2;
    input signed [totalBits - 1 : 0] x1;

    //input wire [iteratorBits - 1 : 0] skip;
    output reg signed [totalBits - 1 : 0] x = startX;
    output reg signed [totalBits - 1 : 0] y = startY;
    output reg signed [totalBits - 1 : 0] z = startZ;
    //input wire [iteratorBits - 1 : 0] skip;
    //output reg signed [totalBits - 1 : 0] ab1 ;
    //output reg signed [totalBits - 1 : 0] ab2;
    //output reg signed [totalBits - 1 : 0] ho;

    reg [iteratorBits - 1 : 0] iterator = 0;
    reg signed [totalBits - 1 : 0] a = startX;
```

```

reg signed [totalBits - 1 : 0] b      = startY;
reg signed [totalBits - 1 : 0] c      = startZ;
reg signed [totalBits * 2 - 1 : 0] h   = 0;
reg signed [totalBits - 1 : 0] abs1    = 0;
reg signed [totalBits - 1 : 0] abs2    = 0;
reg signed [totalBits * 2 - 1 : 0] dxdt = 0;
reg signed [totalBits * 2 - 1 : 0] dydt = 0;
reg signed [totalBits * 2 - 1 : 0] dzdt = 0;
    reg signed [totalBits - 1 : 0] res  = 0;
//absSum  absp(x, rho, abs1);
//absSub  absm(x, rho, abs2);

always @(posedge clk)
begin
    iterator <= iterator + 1;

    if (!iterator)
        begin
            x = a;
            y = b;
            z = c;
        end
    else
        begin
            if (iterator == skip)
                begin
                    a = x;
                    b = y;
                    c = z;
                end
            else
                begin
                    abs1=x+uno;
                    if(abs1<0) begin
                        abs1=-abs1;
                    end
                    abs2=(x-uno);
                    if(abs2<0) begin
                        abs2=-abs2;
                    end
                    h=(m1*x+m0mm1x05*(abs1-abs2))>>>fractionBits;
                    //ho=h;
                    //ab1=abs1;
                    //ab2=abs2;
                    dxdt = (sigma * (y - x-h)-deltaX*(2*x-x1-x2)) >>> fractionBits;
                    dydt = x -y+z;
                    dzdt = (-beta* y) >>> fractionBits;

                    x = x + ((dxdt * dt) >>> fractionBits);
                    y = y + ((dydt * dt) >>> fractionBits);
                    z = z + ((dzdt * dt) >>> fractionBits);
                end
            end
        end
end
endmodule

```

Anexo G: Modulo VERILOG de la unidad bajo prueba de tres circuitos de Chua, uutChuaN.v

```
`timescale 1ns / 1ps

module uutNChua(
    input CLK_50MHZ,
    output y
);
// Inputs

    wire [31:0] x1,x2,x3;
    wire [31:0] y1,y2,y3;
    wire [31:0] z1,z2,z3;

    ///

    integer f;
    integer esc=6;
    wire clk;
    chuaN #
    (.startX (0),.startY (32'hfecccccd))
    chua1 (CLK_50MHZ,x3,x2,x1,y1,z1);
    chuaN chua2 (CLK_50MHZ,x1,x3,x2,y2,z2);
    chuaN chua3 (CLK_50MHZ,x2,x1,x3,y3,z3);
    fDiv divisor(CLK_50MHZ,clk);
    assign y=clk;

endmodule
```

Anexo H: Elementos y equipos utilizados en el sistema

CABLE DE DATOS



CABLE DE IMPRESORA



CARGADO 5 VOLTIOS



ARDUINO 1



SPARTAN 3E FPGA





epoch

Dirección de Bibliotecas y
Recursos del Aprendizaje

UNIDAD DE PROCESOS TÉCNICOS Y ANÁLISIS BIBLIOGRÁFICO Y
DOCUMENTAL

REVISIÓN DE NORMAS TÉCNICAS, RESUMEN Y BIBLIOGRAFÍA

Fecha de entrega: 09 / 11 / 2022

INFORMACIÓN DEL AUTOR/A (S)
Nombres – Apellidos: <i>Rolando Bolívar Donoso Real</i>
INFORMACIÓN INSTITUCIONAL
<i>Instituto de Posgrado y Educación Continua</i>
Título a optar: <i>Magíster en Matemática mención Modelación y Docencia</i>
f. Analista de Biblioteca responsable: Lic. Luis Caminos Vargas Mgs.



Firmado electrónicamente por:
**LUIS ALBERTO
CAMINOS
VARGAS**



0144-DBRA-UTP-IPEC-2022