

# **DISEÑO DE SISTEMAS HIPERCAÓTICOS PARA IMPLEMENTACIÓN EN DISPOSITIVOS LÓGICOS PROGRAMABLES ENFOCADO A APLICACIONES DE SEGURIDAD**

***Jorge Gustavo Vázquez Duran***

Tecnológico Nacional de México/Instituto Tecnológico de Tijuana  
*jorge.vazquez.duran@gmail.com*

***Ramón Ramírez Villalobos***

Tecnológico Nacional de México/Instituto Tecnológico de Tijuana  
*ramon.ramirez@tectijuana.edu.mx*

***Luis Néstor Coria de los Ríos***

Tecnológico Nacional de México/Instituto Tecnológico de Tijuana  
*luis.coria@tectijuana.edu.mx*

***Manuel de Jesús García Ortega***

Tecnológico Nacional de México/Instituto Tecnológico de Tijuana  
*manuel.garcia@tectijuana.mx*

## **Resumen**

Un sistema caótico es un sistema que experimenta una dinámica no repetitiva. Aparece tener un comportamiento aleatorio, sin embargo, dicha dinámica está muy lejos de serlo. Este tipo de sistema exhibe una estructura definida que resulta aparente con el tiempo. Un sistema hipercaótico, es un sistema que presenta una dinámica más compleja que un sistema caótico. Debido a las características anteriormente mencionadas, los sistemas hipercaóticos sugieren complejidad para su implementación física. En este documento se presenta el procedimiento de diseño e implementación de un sistema hipercaótico tipo Lorenz utilizando un dispositivo lógico programable. La metodología descrita puede ser utilizada para la implementación de otros sistemas caóticos o hipercaóticos. El procedimiento de

diseño requiere la utilización de ®MATLAB/SIMULINK en conjunto con la librería *Xilins System Generator*. Para una futura implementación se considera un dispositivo lógico programable de la familia Spartan de la compañía ®Xilinx. Se realizaron simulaciones numéricas y se comparan con las señales adquiridas.

**Palabras Claves:** Dispositivo lógico programable, sistemas caóticos.

## **Abstract**

*A chaotic system is a system with non-repetitive dynamics. Apparently, it seems not to have a random behavior, however, such dynamics are far from being. This type of system exhibits a definite structure that appears over time. A hyperchaotic system is a system that presents a more complex dynamic than a chaotic system. Due to the aforementioned characteristics, the physical implementation of hyperchaotic systems suggests complexity. This paper describes the procedure for design and implementation of a Lorenz-type hyperchaotic system by using a programmable logic device. The described methodology can be useful for the implementation of different chaotic or hyperchaotic systems. The design procedure requires the use of ®MATLAB / SIMULINK and the Xilinx System Generator library. For future implementation is considered a programmable logic device of the Spartan family from the Xilinx Company. Numerical simulations were performed and compared with the acquired signals.*

**Keywords:** Chaotic system, Field Programmable Gate Array.

## **1. Introducción**

En las últimas décadas los sistemas caóticos han recibido gran atención por parte de la comunidad científica. Edward Norton Lorenz, matemático y meteorólogo, fue el pionero en el estudio y desarrollo de los sistemas caóticos. Lorenz construyó un modelo matemático simplificado, en el cual estudiaba el comportamiento climático [Lorenz, 1963], durante su estudio se dio cuenta de que cualquier alteración por más mínima que fuera en las condiciones iniciales afectaría drásticamente las condiciones finales, este sensible fenómeno es conocido como efecto mariposa.

Los sistemas hipercaóticos son intensivamente estudiados en la literatura, debido a las propiedades dinámicas de los sistemas hipercaóticos (por ejemplo, puntos de equilibrios infinitos) [Lassoued, 2016]. Esto se debe a las potenciales aplicaciones que tienen en distintas ramas de la ciencia e ingeniería, por ejemplo, en áreas como: robótica móvil, redes neuronales, sistemas de comunicaciones seguras y encriptamiento en sistemas biométricos.

Tradicionalmente un sistema caótico es implementado utilizando componentes analógicos, por ejemplo, el Sistema de Lorenz en [Cuomo, 1993]. La implementación de este tipo de sistemas utilizando componentes analógicos presenta dificultades, debido a las propiedades y características de los sistemas caótico, por ejemplo, un sistema caótico al tener sensibilidad en sus condiciones iniciales se ve afectado por la tolerancia de los componentes analógicos.

En los últimos años ha surgido gran interés en el desarrollo de procedimiento para la implementación de sistemas caóticos utilizando *Field Programmable Gate Array* (FPGA). En la literatura se puede encontrar artículos relacionados con el desarrollo e implementación de sistemas caóticos en un FPGA, mediante la discretización del sistema de ecuaciones diferenciales ordinarias [Gonzalez, 2005]. Además, se han propuesto procedimientos de modelado y simulación de osciladores caóticos enfocados para la transición de algoritmos desarrollados en ©MATLAB hacia su implementación [Tlelo, 2007].

La implementación de sistemas caóticos utilizando FPGA's provee ventajas comparando con los implementados utilizando componentes electrónicos analógicos. Una ventaja de utilizar FPGA's es que al estar compuesto por procesadores reprogramables los parámetros del modelo matemático pueden variarse con un cambio en el programa, lo que en circuitos analógicos representa un reemplazo físico de elementos pasivos. Además, un FPGA no se ve afectado por tolerancia de los componentes. Igualmente, puede utilizarse para implementar diferentes sistemas caóticos sin importar su complejidad [Sivaranakrishnan, 2007]. Por otro lado, una ventaja a destacar de un FPGA, comparado con otros dispositivos digitales, es su capacidad de configuración y la alta velocidad de datos. Además, el costo de desarrollo e implementación son menores en este tipo

de dispositivos. La implementación de sistemas caóticos en dispositivos digitales puede ser utilizada en comunicaciones seguras [Shuangxia, 2006], en caotificación de motores para su aplicación en lavadoras automáticas de ropa, bandas transportadoras, lavadoras [Sivaranakrishnan, 2007], sistemas de comunicaciones seguras [Xiao, 2009], encriptamiento en sistemas biométricos, entre otras.

El objetivo del presente artículo es el desarrollo de un procedimiento alternativo para obtener las dinámicas de los sistemas hipercaóticos, que permita su implementación utilizando FPGA's. En la presente investigación, se propone un método de implementación utilizando @MATLAB/SIMULINK en conjunto con la librería *Xilinx System Generator* (XSG). La metodología consiste en implementar las ecuaciones diferenciales del sistema caótico, mediante la programación de bloques de @MATLAB/SIMULINK. Posteriormente, mediante la librería XSG transformar la programación de bloques generada en código VHDL implementable en un FPGA. La programación en bloques se realizó en MATLAB R2011, la cual es compatible con Xilinx 14.7. Debido a sus características y la compatibilidad con el software utilizado, para una futura implementación se ha considerado el FPGA Spartan 3AN.

## 2. Métodos

En sección se presenta el modelo matemático que describe la dinámica del sistema hipercaótico de Lorenz-Stenflo. Así como el procedimiento de diseño del oscilador hipercaótico.

### Sistema hipercaótico de Lorenz-Stenflo

El sistema Lorenz-Stenflo, presentado en [Stenflo, 1996], es la generalización de un modelo simplificado que describe ondas de gravedad acústica y está por ecuaciones 1 a la 4.

$$\dot{x} = -ax + ay + cv, \quad (1)$$

$$\dot{y} = rx - y - xz, \quad (2)$$

$$\dot{z} = xy - bz, \quad (3)$$

$$\dot{v} = -x - av, \quad (4)$$

Donde los parámetros  $a, b, c, r$  son positivos. El sistema exhibe un comportamiento hipercaótico para algunos valores de parámetros, incluyendo los valores  $a = 2, b = 0.7, c = 1.5, r = 26$ . En la figura 1 se muestra las proyecciones  $(x, y, z)$  y  $(x, y, v)$ , del sistema de ecuaciones 1 a la 4 para los valores de parámetros descritos en (5).

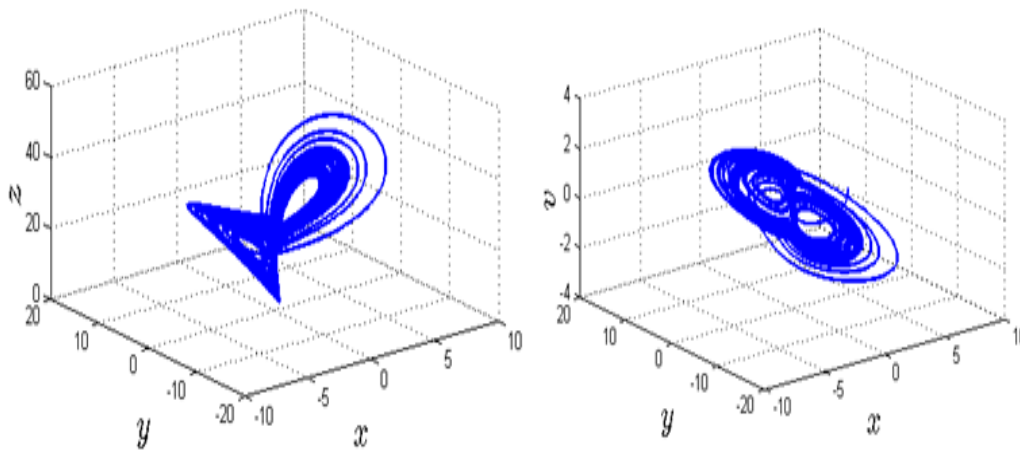


Figura 1 Proyecciones del sistema hipercaótico Lorenz-Stenflo.

### Diseño en Simulink

Los sistemas caóticos parten de ecuaciones matemáticas que gobiernan el comportamiento de su dinámica. Dichas ecuaciones definidas en el sistema (1)-(4) para el sistema Lorenz-Stenflo, fueron representadas mediante la programación de bloques de MATLAB/SIMULINK. En la figura 2 se muestra la implementación del sistema, considerando los valores de parámetros descritos en (5).

Para la implementación del sistema de ecuaciones 1 a la 4 se utilizaron bloques multiplicadores, de ganancia, suma e integradores.

### Diseño con la Librería XSG

Una vez implementado el sistema de ecuaciones 1 a la 4 en Simulink, se realizó la implementación con los bloques de la librería de XSG para tener la compatibilidad de los sistemas con una tarjeta FPGA Spartan 3AN.

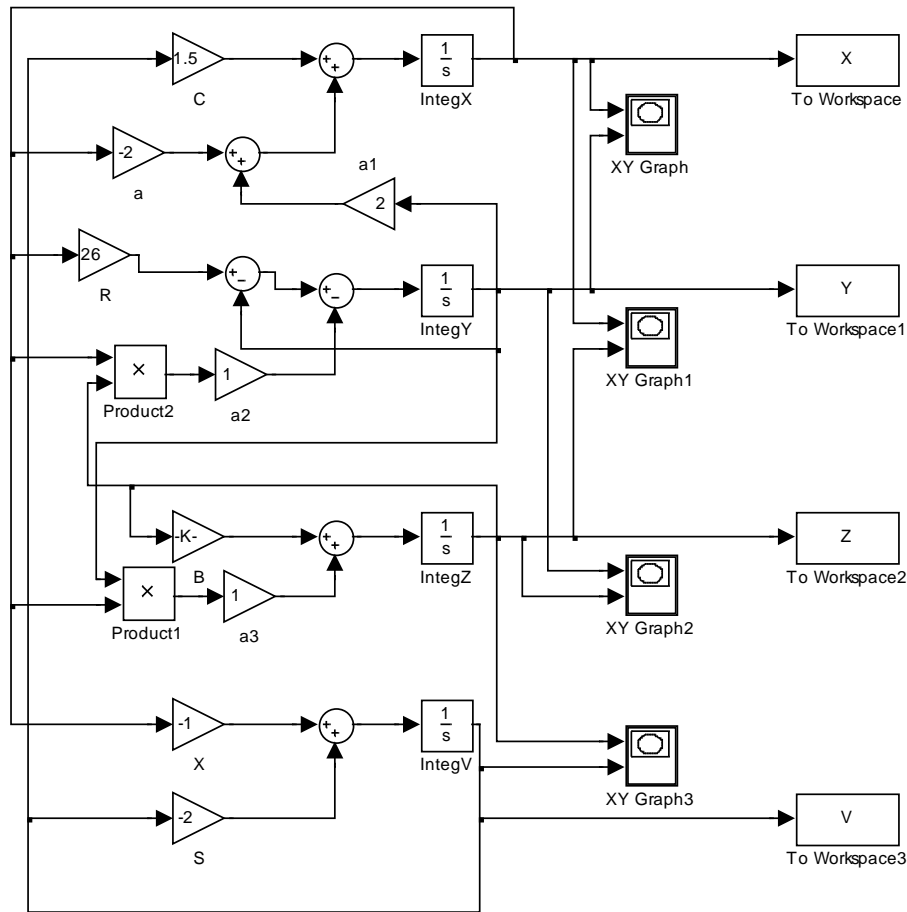


Figura 2 Sistema Lorenz-Stenflo implementado en Simulink.

De la misma manera que la programación a bloques Simulink, la librería XSG cuenta con una gama de bloques muy amplia, la mayoría son similares a los de la librería de Simulink. Aunque no se cuenta con todos los bloques necesarios para llevar a cabo un sistema caótico, en este caso, el bloque integrador. Para llevar a cabo una integración eficiente con la librería XSG, fue necesario construir el integrador con los bloques de la librería XSG, como se muestra en la figura 3.

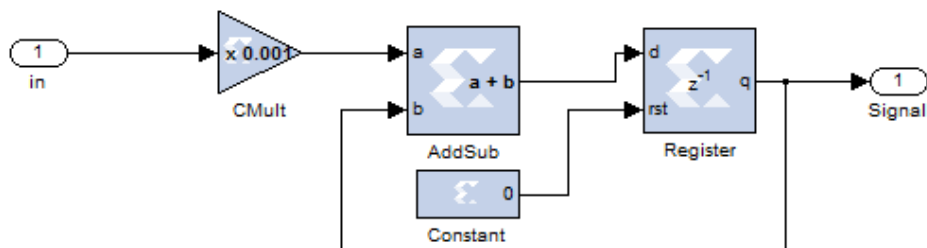


Figura 3 Integrador con la librería XSG.

Una vez implementado el bloque integrador, se procedió a implementar el sistema (1)-(4) utilizando los bloques de la librería XSG, como se muestra en la figura 4.

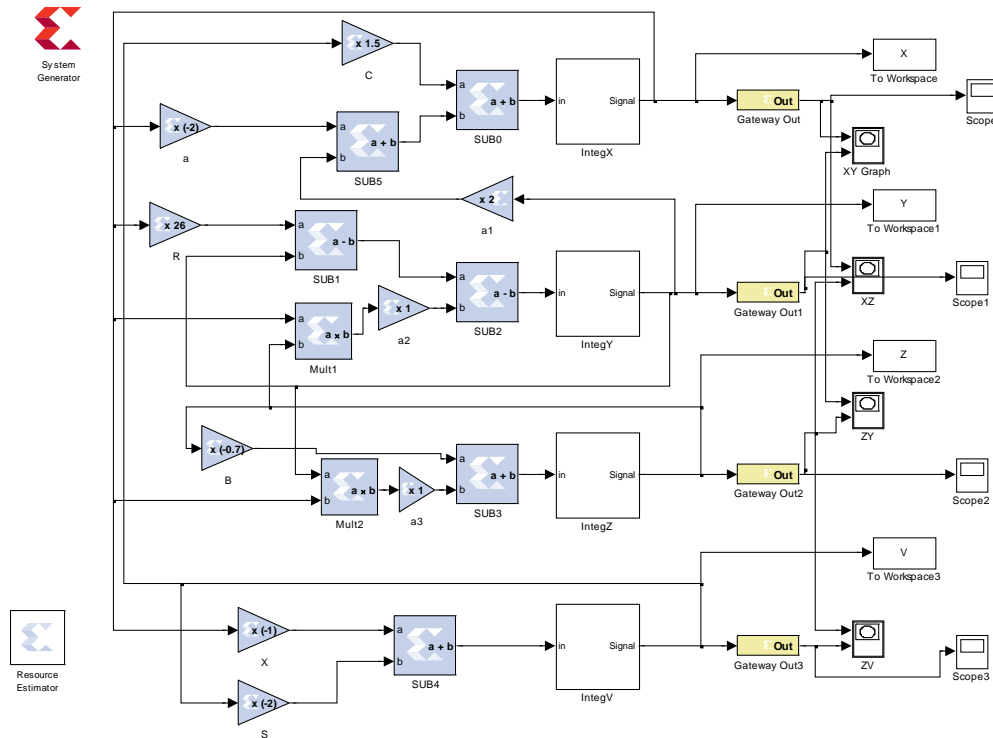


Figura 4 Sistema Lorenz-Stenflo implementado con la librería XSG.

### 3. Resultados

En la presente sección se muestra los resultados obtenidos del diseño del sistema hipercaótico Lorenz-Stenflo para su implementación en FPGAs, utilizando el procedimiento presentado en la sección anterior.

Para la compilación del diseño, que se muestra figura 4, en la en el FPGA se utilizó la librería XSG. De la emulación realizada en la librería XSG se obtuvieron los datos de las señales de las variables de estado y posteriormente las proyecciones de atractor hipercaótico, las cuales muestra en la figura 5.

Comparando los resultados que se muestra en la figura 1 y la figura 5 se visualiza que la compilacion del diseño tiene una dinamica caótica. Por lo tanto, de esta manera se demuestra que la combinacion de ®MATLAB/SIMULINK y la librería XSG de Xilinx permite desarrollar un procedimiento alternativo para la emulación de las dinámicas de sistemas caóticos.

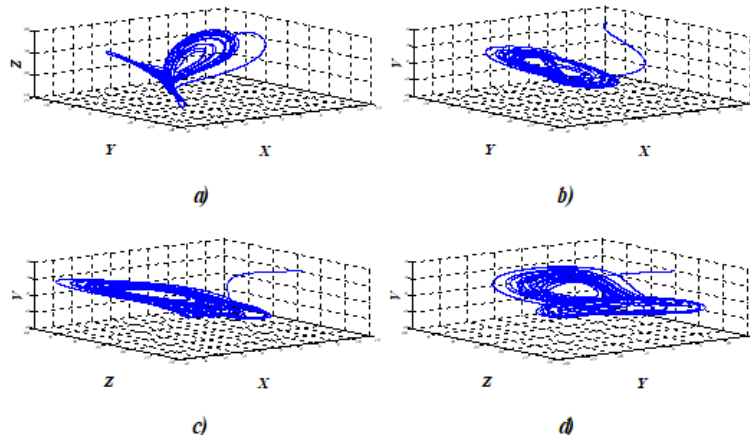


Figura 5 Proyecciones del atractor de Lorenz-Stenflo implementado en la librería XSG.

#### 4. Discusión

Primeramente, uno de los mayores retos de este proyecto fue construir un integrador, utilizando los bloques de la librería XSG, capaz de llevar a cabo el funcionamiento del bloque integrador de Simulink, ya que es necesario conocer dos cosas:

- La función exacta o muy aproximada del bloque de Simulink. Para esto fue necesario leer a fondo las propiedades del bloque para estudiar su comportamiento.
- Los bloques con los que cuenta la librería de XSG. Conociendo el funcionamiento preciso del bloque integrador de Simulink, se debe indagar en la librería de XSG para revisar los bloques que se deben usar para construir el integrador de modo que, además de construirlo, los parámetros no afecten al resultado esperado.

Después de haber construido el integrador, es importante tener en cuenta el tiempo de muestreo del mismo para que la visualización del atractor caótico sea la esperada, dependiendo del sistema a considerar.

Finalmente, el FPGA Spartan 3AN es considerado para futuras implementaciones enfocadas a sistemas de comunicaciones seguras y encriptamiento en sistemas biométricos. Lo anterior, debido a la compatibilidad con el software utilizado en el desarrollo de este proyecto, por las características que tiene (por ejemplo,



conectividad, alta frecuencia de procesamiento) y al ser un dispositivo digital enfocado a aplicaciones de automatización industrial, video y gráficos. Sin embargo, la implementación de sistemas hipercaótico en FPGAs utilizando el procedimiento presentado en este artículo, no está limitado solo a la utilización del FPGA Spartan 3AN.

## 5. Conclusiones

En el presente artículo se presentó un procedimiento alternativo para la emulación de las dinámicas de sistemas caóticos. El procedimiento permite implementar un oscilador caótico en un FPGA Spartan 3AN. La metodología consistió en implementar las ecuaciones diferenciales del sistema hipercaótico Lorenz-Stenflo, mediante la programación de bloques de  $\text{\textcircled{R}}$ MATLAB/SIMULINK. Posteriormente, mediante la librería XSG se transformó la programación de bloques generada en SIMULINK en un código VHDL implementable en un FPGA Spartan 3AN. Como trabajo futuro se pretende realizar la implementación y realizar pruebas validación.

## 6. Bibliografía y Referencias

- [1] Cuomo, K., & Oppenheim, A., Synchronization of Lorenz-based chaotic circuits with applications to communications. *IEEE Transaction on Circuits and Systems II*, No. 40, 626-633, 1993.
- [2] Gonzalez, C. M., Larrondo, H. A., Gayoso, C. A. & Arnone, L. J. Implementación de sistemas caóticos en dispositivos lógicos programables. *XI Workshop IBERCHIP*, 2005.
- [3] Lassoued A., & Boubaker, O. On new chaotic and hyperchaotic systems: A literature surver. *Nonlinear Analysis: Modelling and Control*, No. 21(6), 770-789, 2016.
- [4] Lorenz, E., Deterministic nonperiodic flow. *Journal of the Atmospheric Sciences*. No. 20, 130-141, 1963.
- [5] Shuangxia, Y. Chaotization of a single-phase induction motor for washing machines. *Industry Applications Conference*, No. 1, 855-960, 2006.

- [6] Sivaranakrishnan, R., A new approach on discrete chaotic cryptography using TMS320C6713 digital signal processor. *International Journal of Applied Engineering Research*, No. 2, 545-556, 2007.
- [7] Stenflo, L., Nonlinear equations for acoustic gravity waves, *Physics Letters A*, No. 222(6), 178-380, 1996.
- [8] Tlelo-Cuautle, E., Duarte-Villaseñor, M.A., García-Ortega, J.M., Modelado y Simulación de un Oscilador Caótico usando Matlab. *IEEE Latin America Transactions*. No. 5(2), 95-98, 2007.
- [9] Xiao, Z., A hard disk encryption system realized by the digital signal processor. *International Conference on Computational Intelligence and Security*, 312-314, 2009.