

Emulación en FPGA de un memristor y su aplicación en un circuito caótico

Luis Jothan Gámez Palacios², Opeyemi Micheal Afolabi¹,
José Cruz Núñez Pérez¹

¹ Instituto Politécnico Nacional, IPN-CITEDI,
México

² Tecnológico Nacional de México,
Instituto Tecnológico de Tijuana,
México

luis.gamez@tectijuana.edu.mx, oafolabi@citedi.mx,
nunez@citedi.mx

Resumen. En este artículo se presenta el diseño y emulación de osciladores caóticos basados en memristores de orden entero y fraccional (IOMR y FOMR, respectivamente), implementados en el FPGA XC7A200T-2FBG676C mediante Vivado Design Suite. Los modelos matemáticos que describen el comportamiento de estos sistemas caóticos fueron validados mediante simulaciones numéricas, y posteriormente se diseñaron bloques HDL para compilar el modelo en VHDL. Para evaluar la respuesta y rendimiento del emulador, se analizan distintas configuraciones del oscilador caótico, variando el orden fraccional del sistema. El estudio de recursos y potencia en FPGA mostró que el oscilador caótico basado en IOMR demandó un 279% más potencia que el oscilador caótico basado en FOMR, cuya potencia nominal es de 1.42 W con un uso total de 15,891 recursos (entre LUTs, FFs, DSPs, IOs y BUFGs). La diferencia se atribuye a que el oscilador caótico basado en IOMR requiere 4.25 veces más recursos que el circuito oscilador basado en FOMR, debido a las técnicas de aproximación utilizadas para la solución de ambos sistemas caóticos.

Palabras Clave: FPGA, Memristor, Oscilador caótico, Simulink, VHDL, Vivado.

Memristor Emulation on FPGA and its Application in Chaotic System Circuits

Abstract. This paper presents the design and emulation of chaotic oscillators based on integer-order and fractional-order memristors (IOMR and FOMR, respectively), implemented on the XC7A200T-2FBG676C FPGA using the Vivado Design Suite. The mathematical models describing the behavior of these chaotic systems were validated through numerical simulations, and HDL blocks were subsequently designed to compile the model in VHDL. To evaluate the emulator's response and performance, different configurations of the chaotic oscillator were analyzed by varying the system's fractional order. FPGA resource

and power analysis showed that the IOMR-based chaotic oscillator required 279% more power than the FOMR-based oscillator, which had a nominal power consumption of 1.42 W and a total usage of 15,891 resources (including LUTs, FFs, DSPs, IOs, and BUFGs). The difference is attributed to the fact that the IOMR-based oscillator circuit demands 4.25 times more resources than the FOMR-based oscillator, due to the approximation techniques used to solve both chaotic systems.

Keywords. FPGA, Memristor, Chaotic oscillator, Simulink, VHDL, Vivado.

1. Introducción

En [1], Leon O. Chua analiza, desde la teoría de circuitos, los tres elementos básicos de los circuitos de dos terminales, definidos por relaciones entre dos de las cuatro variables fundamentales: voltaje (v), corriente (i), carga (q) y flujo magnético (ϕ). Tres de las seis relaciones posibles corresponden a la resistencia, el capacitor y el inductor. Chua argumenta que, desde una perspectiva matemática y simétrica, existe un vacío teórico en las relaciones restantes, lo que lo lleva a postular un cuarto elemento.

El concepto de dispositivo con memoria fue introducido por Chua en 1971 con la postulación teórica del memristor (MR), que presentaba una relación no lineal entre carga (q) y flujo magnético (ϕ), permitiéndole almacenar la carga que había fluido a través de él [2]. Aunque la teoría era sólida, no fue hasta 2008 que se logró crear físicamente el memristor en los laboratorios de HP bajo la dirección de Dmitri Strukov, confirmando las teorías de Chua casi 40 años después [3].

Para estudiar el comportamiento dinámico de circuitos con dispositivos con memoria, como los MRs, se han desarrollado modelos lineales por partes [4, 5] y macromodelos SPICE [6-8], útiles para la simulación de sistemas antes de su implementación física. En 2010, Pershin y Di Ventra [9] propusieron un emulador de MR basado en microcontroladores, lo que marcó un avance importante para el desarrollo de emuladores más sofisticados. Muthuswamy [10] propuso un emulador no lineal continuo suave basado en amplificadores operacionales y multiplicadores analógicos. En 2015, Yang *et al.* [11] desarrollaron un emulador de MR HP con características avanzadas como un amplio rango de memristancia, operación bimodal, largo período de no volatilidad y operación flotante. Zhao *et al.* [12] propusieron un emulador universal de memristores controlado por carga y voltaje, ofreciendo dos configuraciones principales, conectada a tierra o flotante, destacando por su versatilidad.

La integración de dispositivos con memoria en circuitos caóticos ha abierto un abanico de nuevas posibilidades de diseño y la implementación de sistemas con comportamientos dinámicos complejos introduciendo no linealidades y propiedades de memoria [13].

Este artículo está organizado de la siguiente manera, la sección 2 contiene teoría relevante para desarrollar los objetivos de este artículo, destacando los dispositivos con memoria, cálculo fraccional, métodos numéricos y sistemas caóticos. En la sección 3 se muestran las simulaciones numéricas en Matlab-Simulink de los osciladores caóticos basados en memristores y las métricas obtenidas del emulador implementado en FPGA mediante Vivado Design Suite. Por último, en la sección 4 se presentan las conclusiones

finales del artículo realizando comparaciones entre los resultados obtenidos con ambos programas.

2. Marco teórico

En esta sección se abordan los fundamentos teóricos necesarios para el desarrollo del presente trabajo. Se introduce el concepto de dispositivo con memoria, haciendo énfasis en el memristor, y se presentan los principios del cálculo fraccional y de sistemas caóticos.

2.1 Dispositivo de memoria

Los dispositivos con memoria, también conocidos como memelementos (ME), representan una clase de componente que posee la capacidad de almacenar su estado anterior debido a una relación no lineal entre ciertas variables, como voltaje, corriente, carga o flujo magnético. En [13] Di Ventra *et al.*, se establece que un ME controlado por m y descrito por un conjunto de k variables de estado x en un instante t se representa matemáticamente mediante la ecuación (1):

$$\begin{aligned} y(t) &= f(x, m, t) m(t), \\ \dot{x} &= g(x, m, t). \end{aligned} \quad (1)$$

donde $y(t)$ y $m(t)$ denotan la salida y la entrada del ME, respectivamente, f es la respuesta generaliza del sistema y g es una función vectorial continua en un espacio k dimensional.

Memristor. Di Ventra *et al.* [13] presentaron un modelo matemático para describir el comportamiento de un MR controlado por carga, en el cual la relación ideal que define su funcionamiento bajo la aplicación de un voltaje en sus terminales se expresa mediante la ecuación (2):

$$\begin{aligned} v_m(t) &= M(x, q, t) i(t), \\ \dot{x} &= g(x, q, t). \end{aligned} \quad (2)$$

donde M representa la memristancia que depende de la cantidad total de carga eléctrica y de la derivada de la función de flujo magnético con respecto a la carga acumulada en el tipo de MR considerando según la ecuación (3):

$$M(q) = \frac{d\phi(q)}{dq}. \quad (3)$$

El modelo matemático correspondiente al MR controlado por carga de la ecuación (2) se generaliza al orden arbitrario utilizando cálculo fraccional, obteniendo la expresión (4):

$$\begin{aligned} {}_0D_t^\alpha v_m(t) &= M(x, q, t) i(t), \\ \dot{x} &= g(x, q, t). \end{aligned} \quad (4)$$

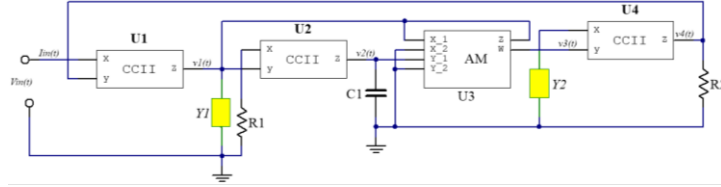


Fig. 1. Modelo universal de tipo conexión a tierra para memelementos de orden entero.

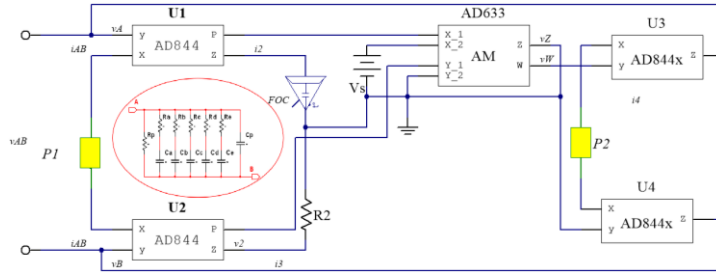


Fig. 2. Modelo universal de tipo conexión flotante para memelementos de orden fraccional.

Por otro lado, para el MR controlado por voltaje, el modelo matemático que define el comportamiento del dispositivo se expresa en la ecuación (5):

$$\begin{aligned} i_m(t) &= G(x, v, t) v(t), \\ \dot{x} &= g(x, v, t). \end{aligned} \quad (5)$$

donde G es el valor de la meminductancia del MR en el tiempo t . El dispositivo controlado por voltaje, al igual que el MR controlado por carga, se generaliza a valores arbitrarios tal como se describe en la ecuación (6):

$$\begin{aligned} {}_0 D_t^\alpha i_m(t) &= G(x, v, t) v(t), \\ \dot{x} &= g(x, v, t). \end{aligned} \quad (6)$$

Modelo universal de memelementos. En la obra de Zhao *et al.* [12], se propone un modelo universal de ME de tipo conexión a tierra de orden entero empleando transportadores de corriente de segunda generación de tipo positivo (CCII+), multiplicadores analógicos (AM), como se expone en la Fig. 1.

Las características terminales de las unidades CCII+ se describen en la ecuación (7), mientras que las características del AM se definen mediante la expresión (8):

$$v_x = v_y, \quad i_y = 0, \quad i_z = i_x, \quad (7)$$

$$v_w = \frac{(v_{x1} - v_{x2})(v_{y1} - v_{y2})}{10} + v_z. \quad (8)$$

Para un MR utilizando el modelo universal de la Fig. 1, los componentes Y_1 e Y_2 se sustituyen por los resistores R_0 y R_3 , respectivamente. El voltaje que atraviesa las terminales del MR de tipo conexión a tierra está dado por la expresión (9):

$$v_{in}(t) = -\frac{R_0 R_2}{R_3} i_{in}(t) + \frac{R_0^2 R_2}{10 R_1 R_3 C_1} q(t) i_{in}(t). \quad (9)$$

Por otro lado, el voltaje que atraviesa las terminales el MR de tipo conexión flotante se describe por la ecuación (10):

$$v_{in}(t) = \left(\frac{R_0 R_4}{R_5} - \frac{R_0 R_2}{R_3} \right) i_{in}(t) + \frac{R_0^2 R_2}{10 R_1 R_3 C_1} q(t) i_{in}(t). \quad (10)$$

En la obra de Li et al. [14], se propone un modelo universal de ME de tipo conexión flotante de orden fraccional empleando transportadores de corriente de segunda generación de tipo positivo y seguidores de tensión (CCII+), un multiplicador analógico (AM), un condensador de orden fraccionario (FOC), como se expone en la Fig. 2. Las propiedades de los AD833 y AD844 utilizados en el diseño se expresan mediante las ecuaciones (11) y (12), respectivamente:

$$v_x = v_y, \quad v_p = v_z, \quad i_y = 0, \quad i_z = i_x, \quad (11)$$

$$v_w = \frac{(v_{x1} - v_{x2})(v_{y1} - v_{y2})}{10} + v_z. \quad (12)$$

Para un MR utilizando el modelo universal de la Fig. 2, los componentes P_1 y P_2 se sustituyen por los resistores R_1 y R_3 , respectivamente. El corriente que atraviesa las terminales del MR de tipo conexión flotante está dado por la expresión (13):

$$i_{AB}(t) = \left(-\frac{R_2}{10 C_\alpha R_3 R_1} J^\alpha v_{AB}(t) - \frac{R_2 V_s}{10 R_1 R_3} \right) v_{AB}(t). \quad (13)$$

donde $v_{AB}(t)$ representa la entrada de voltaje. J^α es el operador de la integral fraccional de orden α , la cual se describe mediante la ecuación (14). V_s es la tensión de puerto del terminal x_2 del multiplicador AD633 y C_α es el valor del FOC:

$$J^\alpha v_{AB}(t) = \frac{1}{\Gamma(\alpha)} \int_0^t (t-\tau)^{\alpha-1} f(\tau) d\tau \quad (14)$$

2.2 Sistemas dinámicos

Los sistemas dinámicos son modelos matemáticos que describen la evolución de un sistema a lo largo del tiempo. Modelan las variables de estado en función del tiempo y las condiciones iniciales [15]. Los sistemas dinámicos de orden entero se describen mediante ecuaciones diferenciales ordinarias (ODE), como se describe en la ecuación (15):

$$\dot{x} = f(x, t, u), \quad (15)$$

donde $\dot{x}$, denota la derivada de x respecto al tiempo, f describe el comportamiento del sistema. Por otro lado, los sistemas dinámicos de orden fraccional se describen mediante ecuaciones diferenciales fraccionales (FDE), como se describe en la ecuación (16):

$$D_t^\alpha x(t) = f(x, t, u), \quad (16)$$

donde D_t^α es el operador de cálculo fraccional de orden α . El orden arbitrario puede ser un número real o racional y se define como $0 < \alpha \leq n$ y $f(x, t, u)$ representa la función que describe la dinámica del sistema.

2.3 Cálculo fraccional

El cálculo fraccional es una extensión natural de las matemáticas clásicas y se casi tan evidente tan pronto como se conocieron las ideas del cálculo básico de Leibniz, permitiendo explorar la viabilidad de aplicar potencias reales a los operadores integral y derivativo, en lugar de restringir dichos operadores a órdenes de naturaleza entera [15]. El origen del cálculo fraccional se remonta a 1695 con la notación $d^n f(x)/dx^n$ que introdujo Leibniz a L'Hospital para la derivada n -ésima de la función $f(x)$ para valores arbitrarios de n [15, 16]. Desde su introducción, se han estudiado muchos tipos de definiciones de derivadas e integrales de orden fraccional, por mencionar solo algunas, Laplace [17] obtuvo expresiones para ciertas derivadas fraccionales de casos particulares y Lacroix [16], quien este último proporcionó por primera vez una solución de derivada de orden m de una función potencia $f(x) = x^p$, tal como se desarrolla en la ecuación (17):

$$\frac{d^m}{dx^m}(x^p) = \frac{p!}{(p-m)!} x^{p-m}. \quad (17)$$

La aparición del operador factorial en la ecuación (17) sugiere la posibilidad de una generalización utilizando la función gamma, reescribiendo se obtiene la expresión (18):

$$\frac{d^m}{dx^m}(x^p) = \frac{\Gamma(p+1)}{\Gamma(p-m+1)} x^{p-m}, \quad (18)$$

donde m es el orden de la derivada la función x^p válida para exponentes p , mediante la relación $m \leq p$. La función gamma extiende la funcionalidad del número factorial k , denotado por $k!$ al dominio de números reales y complejos positivos [18]. La función gamma de un número complejo z se denota por $\Gamma(z)$, y es definida por la integral de la ecuación (19):

$$\Gamma(z) = \int_0^\infty e^{-t} t^{z-1} dt. \quad (19)$$

Definición de Riemann-Liouville. En 1832, Liouville estudió la derivada fraccional para $f(x) = e^{ax}$, pero su definición tenía restricciones de convergencia y generaba números complejos para ciertos casos. Optó por analizar primero la integral fraccional para luego definir la derivada fraccional. Paralelamente, Riemann obtuvo resultados similares mediante series de Taylor [15, 19]. Las investigaciones posteriores de

Bernhard Riemann y Joseph Liouville sobre los estudios que implican la tasa de cambio y las antiderivadas condujeron a la conocida derivada e integral de Riemann-Liouville de orden arbitrario, ecuaciones (20) y (21) respectivamente, para una función continua $f: [a, b] \rightarrow \mathbb{R}$, donde $m \in \mathbb{N}$, establecida por $(m-1) < \alpha \leq m$:

$${}^{\text{RL}}D_t^\alpha f(t) = \frac{1}{\Gamma(m-\alpha)} \left(\frac{d}{dt} \right)^m \int_a^t (t-k)^{m-\alpha-1} f(k) dk, \quad (20)$$

$${}^{\text{RL}}I_a^\alpha f(t) = \frac{1}{\Gamma(\alpha)} \int_a^t (t-k)^{\alpha-1} f(k) dk. \quad (21)$$

Definición de Caputo. Las definiciones de Riemann-Liouville son esenciales en el cálculo fraccional principalmente en el marco de un formalismo puramente matemático, pero presentan dificultades al modelar fenómenos reales [19]. En 1969, Michele Caputo propuso una nueva derivada fraccional, adecuada para problemas con valores iniciales o de frontera en sistemas físicos [18, 19]. La derivada fraccional de Caputo de orden $\alpha > 0$ de una función continua $f(t) \in L_1[a, b]$, con $a \geq 0$, es definida por la ecuación (22),

$${}^{\text{C}}D_t^\alpha f(t) = \begin{cases} \frac{1}{\Gamma(m-\alpha)} \int_a^t \frac{f^{(m)}(\tau)}{(t-\tau)^{\alpha+1-m}} d\tau, \\ \frac{d^m}{dt^m} f(t) \end{cases} \quad (22)$$

donde a y t son los límites del operador diferencial de orden fraccional, Γ denota la función gamma, el orden fraccional está definido por la $a=m$, y la expresión $(m-1) < \alpha \leq m$.

2.4 Métodos numéricos

Existen diversos métodos numéricos para aproximar la solución de sistemas de ecuaciones; sin embargo, cuando el problema a resolver es no lineal, estos métodos suelen implicar un alto costo computacional. Por esta razón, se emplean métodos paso a paso, como el método de Runge-Kutta y sus variantes, que permiten mejorar la eficiencia en la resolución de este tipo de problemas [20].

Runge – Kutta de orden entero. Fue desarrollado en el siglo XX por los matemáticos Carl Runge y Martin Kutta, basado en el método Euler para analizar ecuaciones diferenciales ordinarias. El método Runge – Kutta de cuarto orden (RK4) suele implementarse más a menudo debido a su alta tasa de convergencia para resolver problemas de la forma en la expresión (23), con condiciones iniciales en (24) definida en un dominio uniforme distante y un tamaño de paso $h = x_n - x_{n-1}$, donde $n=1, 2, \dots, N$ [18].

$$\frac{dy}{dx} = f(x, y), \quad (23)$$

$$y(x_0)=y_0. \quad (24)$$

El algoritmo de Runge – Kutta de cuarto orden está definido por la ecuación (25), donde k_n y h son coeficientes reales.

$$\begin{aligned} k_1 &= hf(x_n, y_n), \\ k_2 &= hf\left(x_n + \frac{h}{2}, y_n + \frac{k_1}{2}\right), \\ k_3 &= hf\left(x_n + \frac{h}{2}, y_n + \frac{k_2}{2}\right), \\ k_4 &= hf(x_n + h, y_n + k_3), \\ y_{n+1} &= y_n + \frac{(k_1 + 2k_2 + 2k_3 + k_4)}{6}. \end{aligned} \quad (25)$$

Runge – Kutta de orden fraccional. Considerando un problema con condición inicial definido por la derivada fraccional de Caputo en la ecuación (26), donde $0 < \alpha \leq 1$, $y(t) \in C[t_0, T]$, $f(t, y(t)) \in C[t_0, T]$ y t_0 es el punto base de la derivada fraccionaria [18]:

$$\begin{cases} {}^{C_0}D_t^\alpha y(t) = f(t, y(t)), & t \in [t_0, T], \\ y(t_0) = y_0. \end{cases} \quad (26)$$

El algoritmo Runge – Kutta explícito de 3 estados de orden fraccional (EFORK) se describe mediante la ecuación (27) con coeficientes y pesos determinados por la expresión (28):

$$\begin{aligned} k_1 &= h^\alpha f(t, y), \\ k_2 &= h^\alpha f(t + c_2 h, y + a_{21} k_1), \\ k_3 &= h^\alpha f(t + c_3 h, y + a_{31} k_1 + a_{32} k_2), \\ y_{n+1} &= y_n + w_1 k_1 + w_2 k_2 + w_3 k_3. \end{aligned} \quad (27)$$

$$\begin{aligned} a_{21} &= \frac{1}{2\Gamma(\alpha+1)^2}, \\ a_{31} &= \frac{\Gamma(1+\alpha)^2 \Gamma(2\alpha+1) + 2\Gamma(2\alpha+1)^2 \cdot \Gamma(3\alpha+1)}{4\Gamma(\alpha+1)^2 (2\Gamma(2\alpha+1)^2 \cdot \Gamma(3\alpha+1))}, \\ a_{32} &= -\frac{\Gamma(2\alpha+1)}{4(\Gamma(\alpha+1)^2 \cdot \Gamma(3\alpha+1))}, \\ c_2 &= \left(\frac{1}{2\Gamma(\alpha+1)}\right)^{1/\alpha}, \\ c_3 &= \left(\frac{1}{4\Gamma(\alpha+1)}\right)^{1/\alpha}, \\ w_1 &= \frac{8\Gamma(\alpha+1)^3 \Gamma(2\alpha+1)^2 - 6\Gamma(\alpha+1)^3 \Gamma(3\alpha+1) + \Gamma(2\alpha+1) \Gamma(3\alpha+1)}{\Gamma(\alpha+1) \Gamma(2\alpha+1) \Gamma(3\alpha+1)}, \\ w_2 &= \frac{2\Gamma(\alpha+1)^2 (4\Gamma(2\alpha+1)^2 - \Gamma(3\alpha+1))}{\Gamma(\alpha+1) \Gamma(3\alpha+1)}, \\ w_3 &= -\frac{8\Gamma(\alpha+1)^2 (2\Gamma(2\alpha+1)^2 - \Gamma(3\alpha+1))}{\Gamma(\alpha+1) \Gamma(3\alpha+1)}. \end{aligned} \quad (28)$$

2.5 Sistemas caóticos

Los sistemas caóticos son modelos dinámicos no lineales que presentan un comportamiento aparentemente impredecible, aunque sus fundamentos matemáticos

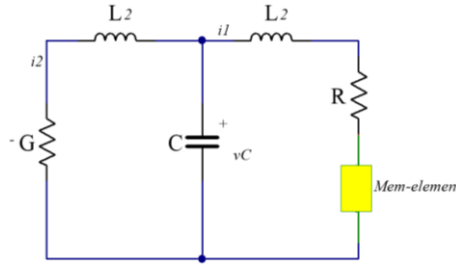


Fig. 3. Circuito oscilador caótico universal de memelementos de orden entero. sean completamente deterministas y se caracterizan por su sensibilidad a las condiciones iniciales [18].

Oscilador caótico universal de memelementos. En [12], Zhao *et al.* propusieron un circuito oscilador universal de ME de orden entero, expuesto en la Fig. 3. El circuito consiste de una conductancia negativa $-G$, los inductores L_1 y L_2 , un resistor R , un capacitor C , y un modelo de ME caracterizado por su voltaje $v_m(q)$.

Aplicando ley de Kirchhoff, el sistema dinámico del circuito oscilador para un ME se deriva en la ecuación (29),

$$\begin{cases} L_1 \frac{di_1}{dt} = v_c - Ri_1 - v_m(t), \\ C \frac{dv_c}{dt} = i_2 - i_1, \\ L_2 \frac{di_2}{dt} = \frac{i_2}{G} - v_c, \\ \frac{dq}{dt} = i_1. \end{cases} \quad (29)$$

Cuando el ME se sustituye por un IOMR en el circuito de la Fig. 3, y empleando la ecuación (9), se deriva la ecuación (30) donde i_1 , i_2 , q , y v_c son variables de estado en el sistema dinámico.

$$\begin{cases} L_1 \frac{di_1}{dt} = v_c - Ri_1 - \left(-\frac{R_0 R_2}{R_3} + \frac{R_0^2 R_2}{10 R_1 R_3 C_1} q \right), \\ C \frac{dv_c}{dt} = i_2 - i_1, \\ L_2 \frac{di_2}{dt} = \frac{i_2}{G} - v_c, \\ \frac{dq}{dt} = i_1, \end{cases} \quad (30)$$

Realizando un escalamiento en el dominio del tiempo del sistema de la ecuación (30), la representación se simplifica al sistema dinámico de la ecuación (31) utilizando $t = L_2 d\tau$, $x = i_1$, $y = v_c$, $z = i_2$, $w = q/L_2$, $a = L_1/L_2$, $b = L_2/C$, $c = R$, $k = 1/G$, $m = -R_0 R_2/R_3$ y $n = R_0^2 R_2/R_1 R_3 C_1$, donde x, y, z, w son las variables de estado y a, b, c, k, m, n , son valores constantes reales.

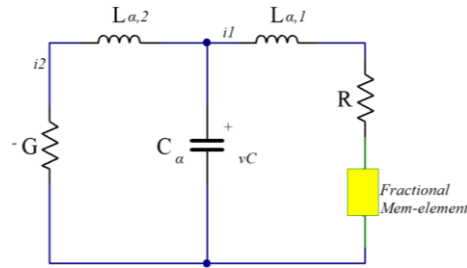


Fig. 4. Circuito oscilador caótico universal de memelementos de orden fraccional.

$$\begin{cases} \dot{x} = a(y - cx - (m + nw)x), \\ \dot{y} = b(z - x), \\ \dot{z} = kz - y, \\ \dot{w} = x, \end{cases} \quad (31)$$

Al considerar el FOMR, la topología del circuito de la Fig. 3 se modifica por elementos de orden arbitrario. Zhao *et al.* proponen en [12] un circuito oscilador universal de ME que considera el orden arbitrario, expuesto en la Fig. 4. El circuito consiste de un capacitor de orden fraccional C_α , los inductores $L_{\alpha,1}$ y $L_{\alpha,2}$, un resistor R , una conductancia negativa $-G$ y un modelo de ME de orden fraccional caracterizado por su voltaje $v_m(q)$.

La dinámica del circuito oscilador caótico se modela matemáticamente como un sistema fraccional conmensurado utilizando leyes de Kirchhoff como se describe la ecuación (32),

$$\begin{cases} L_{\alpha,1} {}^C D_{t_0}^\alpha i_1 = v_C - Ri_1 - v_m(t), \\ C_\alpha {}^C D_{t_0}^\alpha v_C = i_2 - i_1, \\ L_{\alpha,2} {}^C D_{t_0}^\alpha i_2 = \frac{i_2}{G} - v_C, \\ {}^C D_{t_0}^\alpha q = i_1, \end{cases} \quad (32)$$

Sustituyendo el ME por un FOMR y realizando un escalamiento en el dominio del tiempo, como en las ecuaciones (30) y (31), se obtiene el sistema dinámico con la incorporación del orden arbitrario en la ecuación (33),

$$\begin{cases} {}^C D_{t_0}^\alpha x = a(y - cx - (m + nw)x), \\ {}^C D_{t_0}^\alpha y = b(z - x), \\ {}^C D_{t_0}^\alpha z = kz - y, \\ {}^C D_{t_0}^\alpha w = x, \end{cases} \quad (33)$$

donde x, y, z, w son las variables de estado y a, b, c, k, m, n , son valores constantes reales.

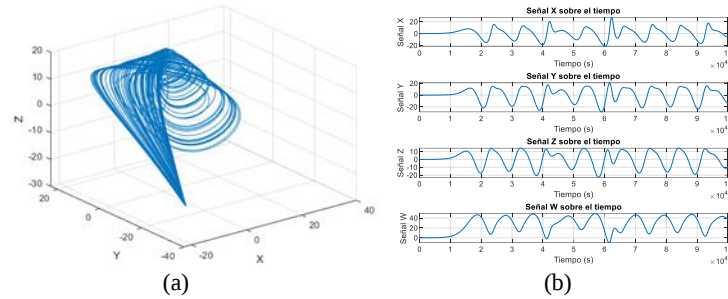


Fig. 5. Respuesta del emulador para un oscilador caótico basado en IOMR: a) gráfico 3D de la relación $x, y, y z$, b) variables sobre el tiempo.

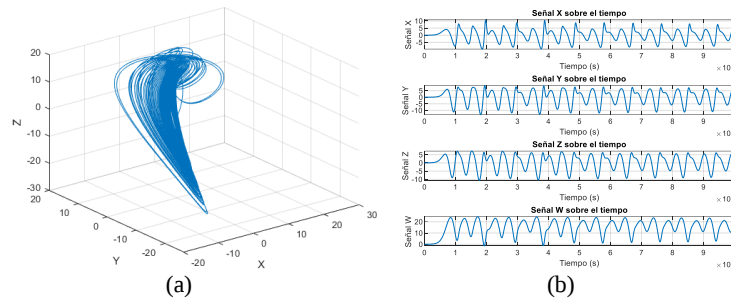


Fig. 6. Respuesta del emulador para un oscilador caótico basado en FOMR cuando $\alpha=0.9$: a) gráfico 3D de la relación $x, y, y z$, b) variables sobre el tiempo.

3. Diseño digital de los osciladores caóticos basados en memristores

En este trabajo se diseñó mediante bloques HDL los osciladores caóticos basados en memristores de orden entero y fraccional en la plataforma de Matlab-Simulink, permitiendo compilar el diseño en VHDL para implementarlo en hardware mediante Vivado Design Suite.

3.1 Oscilador caótico basado en IOMR

El oscilador caótico basado en IOMR descrito en la ecuación (31), se define bajo los siguientes parámetros: $a=2$, $b=1$, $c=0.2$, $k=0.92$, $m=-0.002$ y $n=0.04$. El sistema se inicializa con los estados $x(0)=y(0)=z(0)=w(0)=0.01$. Para aproximar la solución al sistema de ecuaciones, se emplea RK4, con un tamaño de paso $h=0.001$. En la Fig. 5a se expone la relación de $x, y, y z$ que rigen la dinámica del oscilador caótico en 3D, mientras que en la Fig. 5b se muestra el comportamiento de las variables sobre el tiempo obtenidas por el emulador.

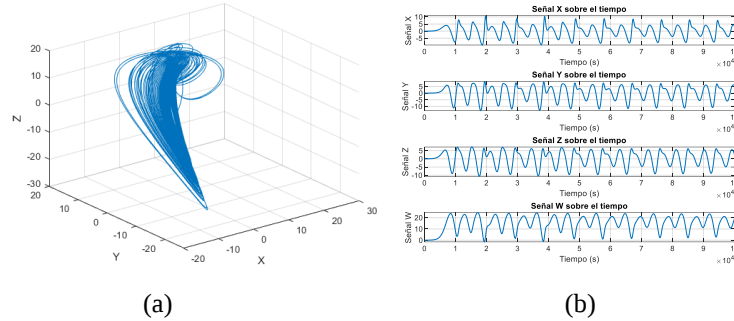


Fig. 7. Respuesta del emulador para un oscilador caótico basado en FOMR cuando $\alpha=0.95$: a) gráfico 3D de la relación x , y , y z , b) variables sobre el tiempo.

Tabla 1. Consumo de potencia del oscilador caótico basado en IOMR emulado en FPGA.

	Distribución de potencia	Oscilador caótico basado en IOMR
Dinamic	Clocks	0.006 W
	Signals	1.870 W
	Logic	1.259 W
	DSP	0.593 W
	I/O	0.090 W
Device static	---	0.143 W
	Total	3.960 W

3.2 Oscilador caótico basado en FOMR

El oscilador caótico basado en FOMR descrito en la ecuación (33), se define bajo los siguientes parámetros: $a=2$, $b=1$, $c=0.2$, $k=0.92$, $m=-0.002$ y $n=0.04$ para $\alpha=0.9$ y $\alpha=0.95$. El sistema se inicializa con los estados $x(0)=y(0)=z(0)=w(0)=0.01$. Para aproximar la solución al sistema de ecuaciones, se emplea EFORK, con un tamaño de paso $h=0.001$. En la Fig. 6a se expone la relación de x , y , y z que rigen la dinámica del oscilador caótico en 3D para $\alpha=0.9$, mientras que en la Fig. 6b se muestra el comportamiento de las variables sobre el tiempo obtenidas por el emulador.

En la Fig. 7a se expone la relación de x , y , y z que rigen la dinámica del oscilador caótico en 3D para $\alpha=0.95$, mientras que en la Fig. 7b se muestra el comportamiento de las variables sobre el tiempo obtenidas por el emulador.

4. Resultados

En esta sección se presentan los resultados obtenidos de la implementación del circuito oscilador caótico basado en memristores, tanto para el caso de orden entero como para el orden fraccional, utilizando una FPGA como plataforma de prueba. Se

Tabla 2. Consumo de potencia del oscilador caótico basado en FOMR cuando $\alpha=0.9$ emulado en FPGA.

	Distribución de potencia	Oscilador caótico basado en IOMR
Dinamic	Clocks	0.005 W
	Signals	0.486 W
	Logic	0.276 W
	DSP	0.422 W
	I/O	0.090 W
Device static	---	0.134 W
	Total	1.413 W

Tabla 3. Consumo de potencia del oscilador caótico basado en FOMR cuando $\alpha=0.95$ emulado en FPGA.

	Distribución de potencia	Oscilador caótico basado en IOMR
Dinamic	Clocks	0.005 W
	Signals	0.490 W
	Logic	0.277 W
	DSP	0.424 W
	I/O	0.090 W
Device static	---	0.134 W
	Total	1.420

comparan ambas implementaciones en términos de métricas de desempeño, considerando la cantidad de recursos utilizados y la potencia requerida por el FPGA.

4.1 Oscilador caótico basado en IOMR

Tras la implementación exitosa en FPGA del emulador, se analizan las métricas de este último. En la Tabla 1 se expone la distribución del consumo de potencia que demanda el emulador, utilizando un total de 3.96 W. El emulador emplea 67,553 recursos del FPGA, repartidos de la siguiente manera: 66,497 LUTs (49.7%), 168 FFs (0.06%), 717 DSPs (96.8%), 170 IO (42.5%) y 1 BUFG (3.13%).

4.2 Oscilador caótico basado en FOMR

En la Tabla 2 se expone la distribución del consumo de potencia que demanda el emulador, utilizando un total de 3.96 W. El emulador emplea 15,891 recursos del FPGA, repartidos de la siguiente manera: 14,932 LUTs (11.6%), 287 FFs (0.001%), 501 DSPs (67.7%), 170 IO (42.5%) y 1 BUFG (3.13%).

En la Tabla 3 se expone la distribución del consumo de potencia que demanda el emulador, utilizando un total de 1.42 W. Los recursos utilizados por el emulador para $\alpha=0.95$ coinciden con los utilizados por el emulador para $\alpha=0.9$.

5. Conclusiones

En este trabajo se compararon osciladores caóticos basados en IOMR y FOMR con órdenes arbitrarios $\alpha=0.9$ y $\alpha=0.95$. La emulación en FPGA permitió analizar el uso de recursos, consumo de potencia y eficiencia computacional. Se encontró que el oscilador basado en IOMR requiere más recursos que el de FOMR, debido al método de aproximación empleado. Aunque el método RK4 aumenta las iteraciones y el uso de recursos, el método EFORK de tres etapas optimizó parcialmente la demanda computacional. El oscilador basado en IOMR consume un 279% más de potencia que el de FOMR. En términos de lógica programable, el oscilador IOMR ocupa el 49.7% de las *LUTs* y el 96.8% de los *DSPs*, mientras que el FOMR solo usa el 11.16% de las *LUTs* y el 67.7% de los *DSPs*, evidenciando una mayor eficiencia en la implementación del FOMR en FPGA.

Agradecimientos. Los autores agradecen al Instituto Politécnico Nacional por el apoyo recibido a través del proyecto SIP 20251203.

Referencias

1. Chua, L.: Memristor-The Missing Circuit Element. *IEEE Transactions on Circuit Theory*, 18(5), pp. 507–519 (1971) doi: 10.1109/TCT.1971.1083337.
2. Khalil, N. A., Fouda, M. E., Said, L. A.: A Universal Fractional-Order Memelement Emulation Circuit. In: 2019 Novel Intelligent and Leading Emerging Sciences Conference (NILES), pp. 67–70 (2019) doi: 10.1109/NILES.2019.8909307.
3. Renping; W., Chunhua, W.: A New Simple Chaotic Circuit Based on Memristor. *International Journal of Bifurcation and Chaos*, 26(9) pp. 1650145 (2016) doi:10.1142/S0218127416501455.
4. Itoh M., Chua, L. O.: Memristor Oscillators. *International Journal of Bifurcation and Chaos*, 18(11), pp. 3183–3206 (2008).
5. Muthuswamy B., Kokate, P. P.: Memristor-based Chaotic Circuits. *IETE Technical Review*, 26, pp. 415–426 (2009)
6. Benderli S., Wey, T. A.: On SPICE Macromodelling of TiO₂ Memristor. *Electronics Letters*, 45, pp. 377–378 (2009) doi: 10.1049/el.2009.3511.
7. Rak A., Cserey, G.: Macromodeling of The Memristor in SPICE. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 29(4), pp. 632–636 (2010) doi: 10.1109/TCAD.2010.2042900.
8. Batas D., Fiedler, H.: A Memristor SPICE Implementation and A New Approach for Magnetic Flux-Controlled Memristor Modeling. *IEEE Transactions on Nanotechnology*, 10(2), pp. 250–255 (2011)
9. Pershin Y. V., Di Ventra, M.: Practical Approach to Programmable Analog Circuits with Memristors. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 57(8), pp. 1857–1864 (2010)
10. Muthuswamy B., Chua, L. O.: Simplest Chaotic Circuit. *International Journal of Bifurcation and Chaos*, 20(5), pp. 1567–1580 (2010)
11. Yang, C. J., Choi, H., Park, S.: A memristor Emulator as A Replacement of A Real Memristor. *Semiconductor Science and Technology*, 30, pp. 1–9 (2015)
12. Zhao, Q., Wang, C., Zhang, X.: A Universal Emulator for Memristor, Memcapacitor, and Meminductor and Its Chaotic Circuit. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 29(1), pp. 013–141 doi:10.1063/1.5081076.

13. Di Ventra, M., Pershin, Y. V., Chua, L. O.: Circuit Elements with Memory: Memristors, Memcapacitors, and Meminductors. *Proceedings of the IEEE*, 97(10), pp. 1717–1724 (2009) doi: 10.1109/JPROC.2009.2021077.
14. Li, Y., Xie, L., Zheng, C.: Modeling and Hardware Implementation of Universal Interface-Based Floating Fractional-Order Memelements. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 33(1) (2023) doi: 10.1063/5.0124793.
15. Strogatz, S. H.: *Nonlinear Dynamics and Chaos: With Applications to Physics, Biology, Chemistry, and Engineering*, 2nd ed. Boca Raton, FL, USA: CRC Press (2015)
16. Podlubny, I.: *Generalized Viscoelastic Models: Their Fractional Equations with Solutions*, vol. 198. New York, NY, USA: Academic Press (1998)
17. Ross, B.: The Development of Fractional Calculus 1695–1900, *Historia Mathematica*, 4(1), pp. 75–89 (1977) doi: 10.1016/0315-0860(77)90039-8.
18. Arfken, G. B., Weber, H. J., Harris, F. E.: *Mathematical Methods for Physicists*, 5th ed. San Diego, CA, USA: Academic Press (2001)
19. Oldham, K. B., Spanier, J.: *The Fractional Calculus*, vol. 17. New York, NY, USA: Dover, 1st ed. (1974)
20. Ghoreishi, F., Ghaffari, R., Saad, N.: Fractional Order Runge–Kutta Methods. *Fractal Fract.*, 7(3), pp. 245 (2023) doi: 10.3390/fractalfract7030245.