Artículo de Investigación. Revista Ciencia e Ingeniería. Vol. 41, No. 2 pp. 197-204, abril-julio, 2020.
ISSN 1316-7081. ISSN Elect. 2244-8780 Universidad de los Andes (ULA)

Operador matemático LFC(*n,k*) en campos finitos basado en concatenación fractal para GF(2m) – Extendido

LFC (*n,k*) mathematical operator in finite fields based on fractal concatenation for GF (2m) - Extended

Sandoval-Ruiz, Cecilia
Facultad de Ingeniería / Dirección de Postgrado
Universidad de Carabobo, Venezuela.
cecisandova@yahoo.com

Resumen

Esta investigación presenta el estudio de las operaciones extendidas en campos finitos LFC - Linear FeedBack Concatenated, a partir de aplicaciones computacionales como los códigos Reed Solomon, que optimiza el manejo de funciones matemáticas compuestas, desde el tratamiento fractal de los operadores, para campo finito álgebra GF(2^m). Se ha determinado una ecuación que describe la concatenación fractal de estos operadores en campos finitos:

$$r_t = \&_{i=0}^{n-k}\, r_{t-1}\,(i\text{-}1) \oplus [(d(i) \oplus r_{t-1}\,(n\text{-}k\text{-}1)) \otimes g(i)]$$

Donde *R(x)* corresponde a la operación matemática LFC (*n,k*), que resulta entre *D(x)* y *G(x)*, aplicando concatenación frac-tal de productos en campos finitos, siendo reformulada por el operador matemático:

$$r_t = \&_{i=0}^{n-k}\, r_{t-1}\,(i\text{-}1) \oplus (\oplus_{t=1}^{m} d(i) \oplus r_{t-1}\,(n\text{-}k\text{-}1)_t \text{ and } g(i)_t)$$

De esta forma, el producto interno se obtiene por un elemento del campo finito GF (2^m), para cada coeficiente del polinomio generador del LFC. El operador definido aquí presenta un campo de aplicación en sistemas de control con variables acotadas. Se ha estudiado el campo de aplicación en la ingeniería del modelo matemático desarrollado, reorganizando los operadores basados en la función LFC (*n,k*), para sistemas regenerativos. Se proporcionó la configuración del modelo con la etapa de adaptación del sistema de transitorio y la retroalimentación lineal en el régimen permanente. Se establece así, un modelo único que simplifica la adaptación a nuevas aplicaciones.

Palabras claves: concatenación fractal, operador matemático LFC (*n, k*), álgebra de campo finito, códigos Reed Solomon, sistemas de funciones iteradas.

Abstract

This research presents the study of the extended operations in finite fields LFC - *Linear FeedBack Concatenated*, starting from computational applications such as the Reed Solomon codes, which optimizes the handling of compound mathematical functions, from the fractal treatment of the mathematical operators, for finite field algebra GF(2^m). An equation describing the fractal concatenation of these operators on finite fields has been determined:

$$r_t = \&_{i=0}^{n-k}\, r_{t-1}\,(i\text{-}1) \oplus [(d(i) \oplus r_{t-1}\,(n\text{-}k\text{-}1)) \otimes g(i)]$$

Where *R(x)* corresponds to the mathematical operation *LFC (n,k)*, resulting between *D(x)* and *G(x)*, applying fractal concatenation of products in finite fields, being reformulated by the mathematical operator:

$$r_t = \&_{i=0}^{n-k}\, r_{t-1}\,(i\text{-}1) \oplus (\oplus_{t=1}^{m} d(i) \oplus r_{t-1}\,(n\text{-}k\text{-}1)_t \text{ and } g(i)_t)$$

In this way, the internal product is obtained by an element of the finite field GF(2^m), for each coefficient of the generating polynomial of the LFC. The operator defined here presents a field of application in control systems with bounded variables. The field of application in the engineering of the developed mathematical model has been studied, rearranging the operators based on the LFC (*n,k*) function, for regenerative systems. A description algorithm VHDL - seed code of self-generation was provided for the configuration of the model with the adaptation stage of the transit system and the linear feedback in the permanent regime. A unique model is established that simplifies the mode and adaptation for new applications.

Keywords — *Fractal Concatenation, Mathematical Operator LFC (n,k), Finite Field Algebra, Reed Solomon Codes, Iterated Function Systems.*

1 Introducción

Actualmente, la implementación computacional de funciones matemáticas, que demandan alta capacidad de cómputo, están siendo objeto de estudio con el objetivo de lograr su implementación eficiente. Las investigaciones en el área de hardware reconfigurable y tecnología FPGA – *Field Programmable Gate Arrays* (Rodríguez y col., 2017), buscan desarrollar modelos circuitales orientados a la optimización del dichas funciones matemáticas compuestas, esto a través de arquitectura sistólica (Yeh, y col., 1984), arquitecturas fractales (Sandoval y col., 2013), así como operaciones definidas para hardware reconfigurable. En este sentido, el estudio de los sistemas de funciones iteradas (Rivera y col., 2012) ha permitido identificar estructuras (Sandoval 2016) y modelos para el análisis de procesos dinámicos, tal es el caso del diseño de códigos Reed Solomon (Sandoval 2017), sistemas MIMO, redes neuronales (Sandoval, 2020 c), con funciones específicas, como el multiplicador en campos Finitos de Galois (Sandoval 2010), (Nazar 2004), suma de convolución, etc., con características de auto-similitud entre sus componentes. Es a partir del desarrollo de estos modelos que se ha identificado la formulación de operaciones particulares de concatenación fractal, sobre campos finitos.

Los *Campos Finitos de de Galois* (Sandoval y col., 2008), (Sandoval 2017) constituyen un área específica de la matemática desarrollada por E. Galois, donde el campo es especificado a través de un elemento primo *p*, base del campo; y un entero positivo *m*, longitud del elemento del campo. Se cumple que p^m corresponde al número de elementos del campo, y las operaciones aritméticas sobre el campo finito dan como resultado un elemento que pertenece al mismo (Nazar 2004). Para la implementación circuital del operador se deben identificar las etapas del multiplicador en campos finitos. En primer lugar se estudia el modelo propuesto por (Tejera y col., s/f), que comprende una etapa de reducción modular sobre los resultados, dividido en cuatro niveles de operación sobre los datos.

Una presentación ampliamente utilizada es la forma polinomial, en la cual se define un polinomio generador del campo, conocido como polinomio irreducible *p(x)*, el cual se operará módulo – *mod*, a través del polinomio irreducible del campo (Nazar 2004), con los resultados de las operaciones para llevar el resultado a la longitud fija definida para el campo. La reducción modular – *mod* puede ser definida como un circuito lineal realimentado LFSR – *Linear FeedBack Shift Register*, se analizó la realimentación en *ciclos iterados* para la paralelización del procesamiento, logrando la definición de un elemento circuital *Linear FeedBack Concurrent Structure* - LFCS (Sandoval 2012), aplicando el procesamiento por bloques, para un tratamiento del conjunto de datos.

Estas estructuras son utilizadas en aplicaciones de códigos Reed-Solomon (Reed & Solomon 1940), (Sandoval 2007), (Xilinx 2018), criptografía, entre otras, se observa la similitud de esta operación y se describe su implementación circuital a través de estas estructuras, con operadores anidados. Para su descripción matemática la estructura circuital fractal, ha sido definida como un modelo de funciones iteradas SFI (Snadoval 2013), estableciendo un operador compuesto para la ecuación resultante.

Propiedades de los Campos Finitos de Galois GF (2^m)

El cuerpo GF(2^m) es una extensión del cuerpo GF(2), denominado cuerpo binario finito, en que cada elemento de GF(2^m) puede representarse mediante un vector de m elementos en GF(2). Para cada elemento $a \epsilon$ GF(2^m) podemos afirmar que:

$$a = \sum_{i=1}^{m-1} a_i \alpha_i \quad donde\ a_i \epsilon \{0,1\} \tag{1}$$

El vector $\{\alpha_0, \alpha_1, \ldots, \alpha_{m-1}\}$ se denomina base de GF(2^m) sobre GF(2). Para la definición del producto en campos finitos. Dado un elemento $a \epsilon$ GF(2^m) la operación de desplazamiento a la izquierda $x \cdot a(x) mod\ p(x)$ puede realizarse de la siguiente forma:

$$x \cdot a(x) mod p(x) = \begin{cases} \sum_{j=1}^{m-1} a_{j-1} x^j & si\ a_{m-1} = 0 \\ \sum_{j=1}^{m-1} (a_{j-1} + p_j) x^j + p_0 & si\ a_{m-1} \neq 0 \end{cases} \tag{2}$$

Siendo a_{m-1}, la realimentación del LFSR.

1.1. Operador matemático LFC (n,k)

El operador definido en este estudio corresponde al procesamiento del esquema circuital LFSR, en configuración Galois (ver Fig.1). Los coeficientes del código del polinomio generador (para el campo extendido) operarán los datos de entrada, definiendo los elementos del campo.El interés de esta aplicación es permitir la definición matemática formal del campo extendido usando un operador circular de composición fractal, para sistemas regenerativos. Éste modelo soporta la concatenación de estructuras autosimilares y simplifica la descripción de operaciones complejas.

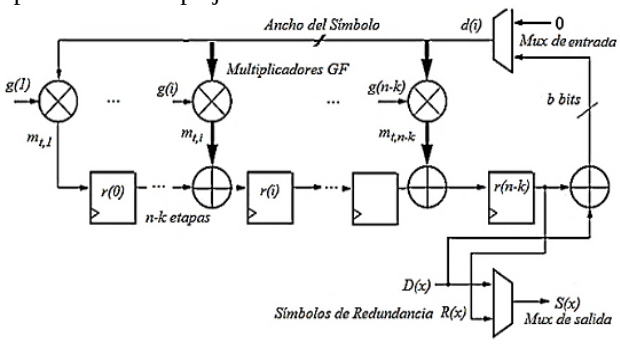

Fig.1 Arquitectura del circuito LFSR(*n*,*k*)

Un tratamiento similar se puede establecer para operaciones fractales con base en el campo finito extendido de

GF(2). Éste será definido como el campo extendido, donde el producto cumple las condiciones definidas, con la función *g(x)*, de los elementos $d_i \in GF(2^m)$.
En este punto se introduce el concepto de Torres de Campos Finitos (Velazquez y col., 2014), éstas son una extensión a su vez de un campo de extensión $\mathbb{F}_p[x]/(f(x))$ con característica p, es decir, un campo primo $\mathbb{F}_p$ que ha sido extendido módulo un polinomio irreducible $f(x)$ de grado n. El campo de extensión $\mathbb{F}_p[x]/(f(x))$ se extiende módulo otro polinomio irreducible $g(x)$ de grado n /m, generando un campo con un número mayor de elementos. La representación en torres de campos permite la ejecución de las operaciones en forma más eficiente.
En (Magaña y col., 2011) se presenta un algoritmo fractal mediante el cual se puede hacer remalleo automático de redes bidimensionales de elemento finito, con base en éste se considera la autogeneración de modelos circuitales fractales (Sandoval 2017), a partir de elementos finitos de operadores básicos en campos finitos de Galois, con lo que se pueden obtener campos extendidos que cumplen con los criterios de diseño. A partir de la matematización, el proceso de construcción de un modelo matemático para casos prácticos, seleccionando el estudio de la aplicación Reed Solomon (Sandoval 2014 *a,b*).

Aplicaciones en Ingeniería del operador LFC(n,k)
El estudio del operador LFC(*n,k*) inició de la aplicación de códigos Reed Solomon 2D-RS para el área de telecomunicaciones, en el cual se identificó la correspondencia entre la función del producto $GF(2^m)$ y la implementación del arreglo circutal del código Reed Solomon.
El concepto para un tipo de red neuronal basada en esta arquitectura FNN- Fractal Neural Network (Sandoval 2020), en el cual se simplifica las técnicas de entrenamiento profundo a través del particionado de las capas de la red en etapas con objetivo conocido, considerando la reorganización de la arquitectura LFSR en configuración Galois.
En el area de ingeniería se propone la extrapolación del operador LFC, en el estudio y correspondencia entre components para la adaptación de diversos modelos matemáticos.

1. Hardware reconfigurable (Sandoval 2019), tecnologia circular y reciclaje electrónico.
2. Procesamiento digiral de señales definido por software (Sandoval 2017).
3. Aplicaciones en sistemas regenerativos y control de energías renovables (Sandoval, 2018 a,b), configuración adaptativa en sistemas ERNC (Sandoval 2019 a,b) y su extension a sistemas fotovoltaicos (Sandoval 2020 a,b).

Entre otras aplicaciones de control y comunicaciones, con elementos de realimentación lineal.

2 Preliminares

Este análisis parte del estudio del modelo matemático de la representación polinomial, en el que se enuncia: Si *p(x)* es el polinomio irreducible, entonces la multiplicación de dos elementos del campo, representados como los polinomios *A(x)* y *B(x)* es el producto algebraico de los dos polinomios, y la operación módulo del polinomio *P(x), también conocido* como reducción modular:

$$C(x) = A(x).B(x) \leftrightarrow C(x) = A(x)\ x\ B(x)\ (mod\ p(x)) \quad (3)$$

La multiplicación de polinomios es asociativa, conmutativa y distributiva con respecto a la adicción por lo cual se obtienen:

$$C(x)=B(x)(\Sigma_{i=0}^{m-1} A_i\, x^i)\ mod\ p(x) \rightarrow$$

$$C(x)= \Sigma_{i=0}^{m-1} B_i\,(A(x)\ x^i\ mod\ p(x)) \quad (4)$$

Donde *A(x)* y *B(x)* corresponden a la representación polinomial de los operandos y *p(x)* es el polinomio irreducible del campo de *Galois*. En el caso del codificador RS el multiplicando *A(x)* corresponde a un coeficiente del polinomio generador del código (para una doble extensión de $\mathbb{F}_p$, *A(x)* en el campo extendico permite generar los elementos del códigos Reed Solomon, así como *p(x)* en el polinomio irreducible) y el multiplicador *B(x)* a la entrada de datos. Para realizar el cálculo de la reducción modular se emplea el concepto de la división de polinomios sobre campos finitos.
Donde $r(x) = A(x)\ mod\ p(x)$, corresponde al residuo de la división entre el operando *A(x)* de la multiplicación y el polinomio irreducible del campo finito $GF(2^m)$. Esta operación de reducción modelar sobre el campo finito, es definida:

$$a(x) = x^i\ A(x)\ mod\ p(x) \quad (5)$$

La expresión matricial para el producto de símbolos en campos finitos de Galois, puede ser expresada como:

$$A(x)\cdot B(x) = \begin{bmatrix} a_{1,0} & \cdots & a_{1,m-1} \\ \vdots & \ddots & \vdots \\ a_{m,1} & \cdots & a_{m,m-1} \end{bmatrix} . \begin{bmatrix} b_{1,0} & \cdots & b_{1,m-1} \\ \vdots & \ddots & \vdots \\ b_{m,1} & \cdots & b_{m,m-1} \end{bmatrix} = [c_0 \quad \ldots \quad c_{m-1}],$$

$$con\ \ a_{t,i} = a_{t\text{-}1}(i\text{-}1)\ xor\ (a_{t\text{-}1}(m\text{-}1)\ and\ p(i))\ y\ b_{t,i} = b(i)$$

Por otra parte, en la definición de la palabra de código Reed Solomon, expresada a través de un polinomio generador, donde cada palabra debe ser múltiplo del polinomio generador *G(x)*, expresada ésta en su forma sistemática, corresponde al bloque de información *D(x)*, adicionando los símbolos de redundancia calculados sobre el bloque de información. Este cálculo es el bloque resultante como residuo de la operación de división entre el polinomio generador *G(x)*, dada como $R_{g(x)}[.]$ aplicada sobre los símbolos

de datos.

$$C(x) = x^{n-k}\, D(x) + R_{g(x)}\,[D(x)\, x^{n-k}\,] \qquad (6)$$

Siendo $C(x)$ - la palabra de código, n - el número de símbolos de la palabra de código y k - el número de símbolos de la palabra de datos. La expresión matemática corresponde a ensamblar dos polinomios con desplazamiento, definido como: $c = (D << (n\text{-}k)) + (D << (n\text{-}k))\ \%\ g$; donde se desplaza el polinomio de datos de información $n\text{-}k$ posiciones a la izquierda, y los $n\text{-}k$ símbolos menos significativos son completados con el residuo de la operación *mod* del polinomio $G(x)$. De manera tal que la expresión polinomial de la palabra de código queda definida como la suma de los polinomios mencionados.

$$C(x) = x^{n-k}\, D(x) + x^{n-k}\, D(x)\ mod\ g(x) \qquad (7)$$

Encontrando así la expresión matemática del generador de símbolos de redundancia

$$R(x) = x^{n-k}\, D(x)\ mod\ g(x) \qquad (8)$$

Similar a la expresión de los polinomios generados como residuos parciales en la operación de multiplicación en campos finitos (ec. 5), lo que evidencia un tratamiento similar de los operandos en el caso de multiplicador de dimensión m y en el caso del generador de redundancia de dimensión extendida $m{\cdot}(n\text{-}k)$, donde cada una de las iteraciones del cálculo de convolución puede ser estimado en un componente matricial de la forma concurrente. El modelo desarrollado se presenta en la Figura 2.

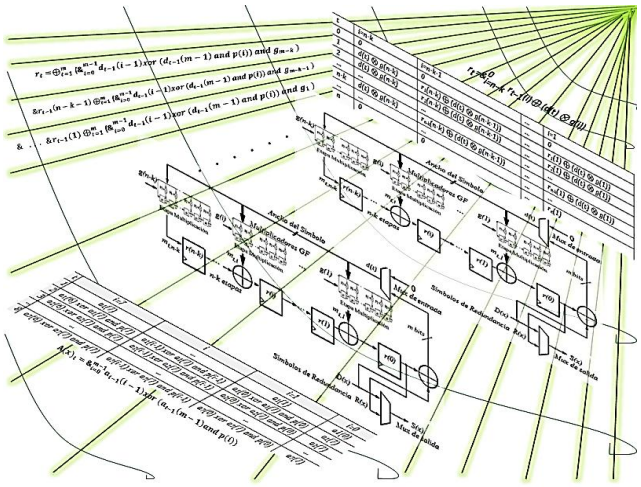

Fig.2. Modelo Fractal LFC (n,k)

El esquema representa la superposición de operadores en una distribución tiempo-espacio, para la asignación a las aplicaciones específicas.

3 Resultados

Los principales resultados de esta investigación se resumen a continuación:

Siendo $D(x)$ una palabra de datos de tamaño k, la cual es operada por el polinomio $G(x) = px^{n-k} + qx^{n-k-1} + \ldots + w$, un polinomio irreducible de longitud $n\text{-}k$. Se obtendrá una palabra de código $C(x)$ de longitud $n\text{-}k$, definiendo el resultado de la operación dentro del campo finito extendido por una operación fractal definida LFC – *Linear FeedBack Concatenated*, con operaciones internas de multiplicadores GF(2^m).

Teorema 1.1. Sea $C(x) = \&_{i=0}^{k}\, D(i)\ \&\ R(x)$, un elemento del campo finito definido por la operación de reducción modular, $R(x) = \&_{i=k}^{n}\ R_i\,(0)$, sobre el polinomio $G(x) = g_{n-k}\, x^{n-k} + g_{n-k-1}\, x^{n-k-1} + \ldots + g_1$, del elemento $D(x)$. De esta manera, se define una transformación del elemento dato, así como un elemento resultante de la operación modular de éste por el polinomio del código. Ésta puede ser interpretada como la concatenación de los k símbolos de datos, con los $n\text{-}k$ símbolos de redundancia. Estos últimos corresponden a la salida de un generador de secuencia LFSR en su representación de *Galois*.

Demostración: Cada elemento resultante será siempre un elemento dentro del campo definido por la dimensión de $G(x)$, dado que la magnitud del elemento final, será $n\text{-}k$, dado por la magnitud del polinomio, entendiendo que cada elemento $D(x)$ cuenta con una longitud definida por m bits, cuyo producto por los elementos de $G(x)$, se realizan sobre el campo finito definido por el polinomio irreducible: $p(x) = p_m\, x^m + p_{m-1}\, x^{m-1} + \ldots + p_1$

Teorema 1.2. Sea el operador LFC – *Linear FeedBack Concatenated*, correspondiente a la concatenación de productos en campos finitos, se tiene que:

$$LFC(D(x)) = \sum_{i=0}^{n} \left(\sum_{x=0}^{m} \left(d_k(x)\ xor\ (d_k(m\text{-}1))\ and\ p(x)\right)\right] and_m\ g(n\text{-}i)) \qquad (9)$$

Demostración: Partiendo de la ecuación de convolución, basado en el esquema LFSR – *Linear FeedBack Shift Register* en la representación de *Galois*, se sustituye la salida de datos $y(n)$ por el vector de símbolos de redundancia $R(x)$, la entrada de datos $x(k)$ por los datos a codificar (compuesto con la realimentación) $d(t)$, y los coeficientes de la función de transferencia $h(n\text{-}k)$ por los coeficientes del polinomio generador del código $g(n\text{-}k)$, obteniendo así una expresión de la forma:

$$LFC(D(x)) = R(x) = \sum_{k=0}^{n} d(t)\cdot g(n-k) \quad (10)$$

para los n-k símbolos

Para dicha expresión se ha empleado un término *d(t)* que corresponde a un arreglo compuesto entre *d(k)* y la realimentación del residuo en la posición menos significativa del polinomio $r_{k-1}(0)$, esto con el propósito de conservar la similitud de la expresión matemática (sin realimentación), al sustituir en función de la entrada del codificador *d(k)*, correspondiente a una muestra *k* del vector de datos, se obtiene:

$$R(x) = \sum_{k=0}^{n} (d(k) \oplus r_{k-1}(n-k)).g(n-k) \quad (11)$$

Desarrollando el producto en campos finitos de *Galois*, se tiene:

$$R(x) = \sum_{k=0}^{n} [(d(k) \oplus r_{k-1}(n-k)) \bmod p(x)].[\, g(n-k)] \quad (12)$$

La ecuación define el producto de *a(x)* correspondiente al primer operando *mod* el polinomio generador del campo *p(x)* y el vector b(x) correspondiente al segundo operando. Siendo: $a(x) = d(i) \oplus r_{i-1}(0)$ Se puede expresar la operación *mod* como la convolución realimentada.

$$a(x) \bmod p(x) = \sum_{k=0}^{m} a(k).p(n-k) \quad (13)$$

Donde se puede expresar en correspondencia con:

$$a_m(x) = \sum_{i=0}^{m} (a(i) \oplus a(m)).p(n-i) \quad (14)$$

De esta manera, el tratamiento de la operación *mod* será dada como la convolución con *p(x)* del elemento *a(m)* realimentado. Luego se sustituye en la ecuación del producto en campos finitos GF, en la ecuación del codificador, de forma iterativa, con lo que se genera la expresión general:

$$R(x) = \sum_{i=0}^{n} (\sum_{x=0}^{m} (d_k(x) \text{ xor } (d_k(m-1)) \text{ and } p(x))] \text{ and}_m\ g(n-i)) \quad (15)$$

Para finalmente, expresar la ecuación matemática del modelo matricial del codificador RS, como un arreglo de símbolos de redundancia.

$$R(x) = \begin{bmatrix} r_{1,0} & \cdots & r_{n-k} \\ \vdots & \ddots & \vdots \\ r_{n,0} & \cdots & r_{n,n-k} \end{bmatrix};$$

$$con\ r_{t,i} = r_{t-1}(i-1) \oplus [(d(i) \oplus r_{t-1}(n-k-1)) \otimes g(i)]$$

Obteniendo una representación matricial, para su implementación concurrente.
Igualmente, se puede identificar la correspondencia entre la ecuación descriptiva del arreglo de términos r_t y el arreglo de términos a_t, como se observa:

$$a_t = \&_{i=0}^{m-1}\ a_{t-1}(i-1) \text{ xor } (a_{t-1}(m-1) \text{and } p(i)) \quad (16)$$

$$r_t = \&_{i=0}^{n-k}\ r_{t-1}(i-1) \oplus [(d(i) \oplus r_{t-1}(n-k-1)) \otimes g(i)] \quad (17)$$

La expresión del producto en campo finito GF se puede sustituir como se muestra en la ecuación:

$$r_t = \&_{i=0}^{n-k}\ r_{t-1}(i-1) \oplus (\oplus_{t=1}^{m}\ a_t \text{ and}_m\ b_t) \quad (19)$$

Luego, sustituyendo a_t y b_t, *en función de la realimentación* d_{t-1} *y los coeficientes g(i)* desarrollando la ecuación, se obtiene la ecuación 20. Se puede notar que de la generalización se debe operar el término $r_{t-1}(n-k)$, pero de acuerdo a la estructura este término es nulo.

$$\begin{aligned} r_t = {} & r_{t-1}(n-k-1) \oplus [\ \oplus_{t=1}^{m}\ [\&_{i=0}^{m-1}\ d_{t-1}(i-1)\ \text{xor} \\ & (d_{t-1}(m-1) \text{and } p(i))]\ \text{and}_m\ g(n-k)\ ]\ \&\ r_{t-1}(i-1) \oplus \\ & [\ \oplus_{t=1}^{m}\ [\&_{i=0}^{m-1}\ d_{t-1}(i-1) \text{ xor } (d_{t-1}(m-1) \text{and } p(i))] \\ & \text{and}_m\ g(n-k-1)\ ]\ \&\ \ldots\ r_{t-1}(0) \oplus [\ \oplus_{t=1}^{m}\ [\&_{i=0}^{m-1}\ d_{t-1}(i-1) \\ & \text{xor } (d_{t-1}(m-1) \text{and } p(i))]\ \text{and}_m\ g(1)\ ] \end{aligned} \quad (20)$$

De esta manera se obtiene la expresión detallada del modelo desarrollado, correspondiente a la versión extendida de la ecuación de concatenación, ésta representa la aplicación del modelo concurrente del LFSR.

$$\begin{aligned} r_t = {} & r_{i-1}(n-k-1) \oplus d(t) \otimes g(n-k) \& r_{i-1}(n-k) \oplus \\ & d(t) \otimes g(n-k-1)) \& \ldots d(t) \otimes g(1)) \end{aligned} \quad (21)$$

Teorema 1.3. Sea la función *f* es una operación definida como el producto en campos finitos sobre el polinomio *p(x)*, y la función *g* es una operación de concatenación de productos sobre el polinomio *g(x)*, entonces:

$$g(f(x)) = \&_{i=n-k}^{k}\ g_{i-1} + g_i\,(f(x)) \quad (22)$$

Al sustituir cada operador en el operador fractal, se obtiene el producto de convolución para ecuaciones matemáticas. Si la función *g(x)* corresponde a *n-k* elementos operados en el producto de campo finito de la función *f*, $f(x) \in GF(2^m)$, entonces se obtiene que $g(x) \in GF(2^{m \cdot n-k})$.

Demostración: La operación matemática da como resultado elementos de longitud *n-k*, lo que puede ser expresado como GF[GF$(2^m)^{n-k}$], dado por la concatenación fractal de los operadores definidos para productos en campos finitos, con una definición de campo finito extendido, sobre elementos ϵ GF(2^m).

4 Aplicaciones en Ingeniería

4.1. Operador en Hardware Reconfigurable

A partir del modelo es possible establecer una ecuación generatriz para campos extendidos o concatenación de funciones autosimilares, definidos en término del operador LFC(*n*,*k*) con definición de sus parámetros, en una matriz de lógica reconfigurable. Cada uno de los módulos de la matriz, tendrá elementos selectores para la habilitación de las ramas de operación y así configurar en hardware la arquitectura del operador lógico-matemático para las aplicaciones específicas en el campo de la ingeniería.

El desarrollo del operador está basado en la identificación de correspondencia entre los circuitos fractales y la simplificación de la estructura para su descripción en hardware, a fin de lograr aplicaciones definidas por software. Este modelo presenta un aporte para ser implementado sobre tecnología FPGA, con módulos LFC como se presenta en la Figura 3.

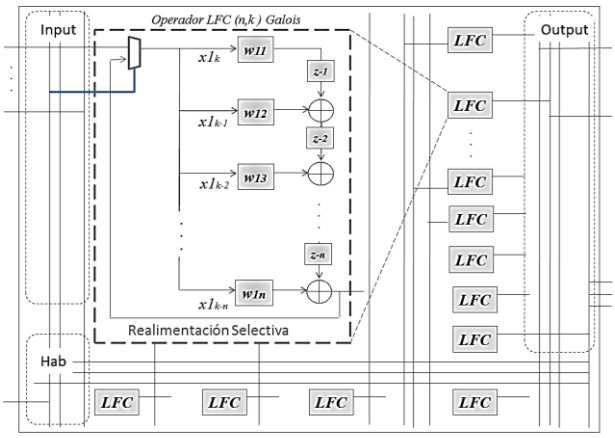

Fig. 3. Arreglo Reconfigurable de operadores LFC(n,k)

La arquitectura propuesta está orientada a implementación electrónica de hardware con el operador LFC(*n*,*k*), basada en el desarrollo matemático expuesto en la presenta investigación, con optimización de su arquitectura a través del operador, con la simplificación de una ecuación generalizada para las estructuras hardware que se adapte a la longitud de los datos y el número de etapas.

4.2. Aplicaciones en ERNC

Entre las aplicaciones para energías renovables no convencionales – ERNC se tiene el modelo matemático generalizado, basado en la estructura LFSR para diversos arreglos de convertidores (Sandoval 2019).

El operador LFC(*n*,*k*) se identifica en sistemas desarrollados como fotovoltaica a nivel de células componentes y a nivel de la unidad de generación fotovoltaica, así como en un sistema óptico, en función de la correspondencia LFSR. Se plantean sistemas de heliostatos, basados en los principios de direccionamiento de la radiación solar por principio óptico, así como la optimización respecto a la altura del arreglo en una estación aérea (definiendo una altura *h* óptima), que permita re-direccionar la radiación solar con cometas (función similar a las repetidoras de señales en telecomunicaciones). Entre sus funciones:

a. Concentración la radiación para una longitud de onda selectiva, conocido como sensibilización para el elemento fotovoltaico, aplicado en el caso de HCPV.
b. Filtración de componentes de radiación UV (con aplicaciones específicas), así proteger áreas de glaciares y ambientes forestales sensibles de radiación directa.
c. Re-direccionamiento de la radiación hacia el campo fotovoltaico terreno, con el fin de solventar limitaciones de la topografía.
d. Guía de onda, aplicaciones con fibra óptica para concentración solar y transmisión de forma eficiente.

La correspondencia de los operadores concatenados, almacenamiento y secuencia se presenta en la Tabla 1.

Tabla 1. Correspondencia del SFV con elementos LFSR

Sistemas Fotovoltaicos definidos por Software SDF	Células Fotovoltaicas	Capas reemplazables de materiales fotovoltaicos
		Arreglos Tándem semiconductores
		U-Condensadores de almacenamiento
		Dopaje de electrones libres
	Paneles Fotovoltaicos	Matriz configurable de células FV
		PV Bifacial, Concentrador multiplexado
		Elevador de Tensión, seguimiento solar
		PERC capa reflectante de realimentación
	Arreglos Fotovoltaicos	Matriz de conmutación serie-paralelo
		MPPT redes neuronales control
		Almacenamiento transitorio Ultra-Condensadores
		Realimentación del sist. de control
	Red Eléctrica	Redes Reconfigurables ERNC
		Redes Neuronales para optimización
		Almacenamiento de ERNC
		Smart Grid

La implementación consiste en un optimizador multiplexado, como se muestra en la Fig. 4.

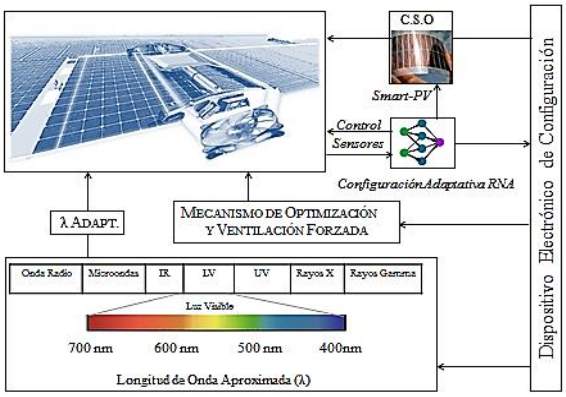

Fig.4. Sistema de Implmentación del Optimizador

A partir del diseño del adaptador para sistemas fotovoltaicos reconfigurables, se define el modelo representado en la Fig. 5.

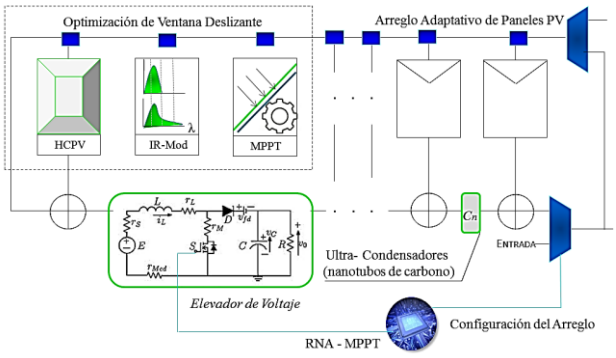

Fig.5. LFSR del Sistemas Fotovoltaico con optimización HCPV

El modelo de la aplicación HCPV se expresa en la ec. 23.

$$y(t) = \sum_{i=1}^{n} f_i(x) * x(t) + b_s * y_0(t-1) \qquad (23)$$

Donde $f_c(t)$ corresponde a la ganancia dinámica del operador, sea este concentrador, convertidores FV, convertidor de almacenamiento, configuración del sistema, destacando que un operador puede ser un circuito LFSR con la función (1) concatenada en su estructura. En tanto que $x_c(t)$ corresponde a la señal de entrada radiación solar incidente, radiación solar concentrada (en este caso la señal es iterativa sobre el operador), energía convertida para almacenamiento, etc. Y el parámetro $y_c(t-1)$ representa la realimentación de la salida.

$$r(t) = \sum_{i=1}^{n} c_i\left(\sum_{i=1}^{m} f_i(x) * x(t)\right) + c_i * y_0(t-1) + b_s * r_0(t-1) \qquad (24)$$

Adicionalmente, la expresión tiene selectores de realimentación b_s y relés de selección de ramas de operadores c_i, de manera que el coeficiente de una rama define el peso correspondiente o la selección on/off de la rama de operación, en el caso de ser un operador LFSR, el coeficiente multiplicara la ecuación de convolución.

5 Conclusiones

Finalmente, se ha establecido la relación del operador LFC a una aplicación práctica y extensión del diseño en sistemas de dinámica compleja, como son los sistemas de energías renovables, detallado en (Sandoval 2020). Donde se introduce el tratamiento de optimización secuencial y almacenamiento transitorio de energía, en correspondencia con la arquitectura LFSR, con la simplificación asociada. El reconocimiento de similitud entre las funciones, ha permitido definir un operador compuesto, con las ventajas de un tratamiento matemático generalizado. Las ecuaciones desarrolladas en esta investigación representan un aporte para optimización del cómputo en aplicaciones de códigos correctores de error, criptografía y para el diseño de sistemas complejos, con realimentación, siendo de especial interés en sistemas multidimensionales como códigos 2D-RS. Ofrece una base para el desarrollo de aplicaciones en hardware usando VHDL, estableciendo una formulación matemática para modelar sistemas de información, control avanzado con salidas enmarcadas en un espacio finito y otros esquemas innovadores, siendo un avance científico sobre modelos no acotados.

Referencias

Magaña del Toro R, Hermosillo- Arteaga AR, Romo-Organista MP, Carrera-Bolaños, J 2011,Análisis con elemento finito y remalleo fractal en geotecnia, Ing. Investig. y Tecnol., vol. XII, no. I, pp. 103–118.

Nazar AS, 2004, Implementación Eficiente de Algoritmos Criptográficos en Dispositivos de Hardware Reconfigurable, Centro de Investigación y Estudios Avanzados del Instituto Politécnico Nacional, México.

Reed & Solomon, 1960. Polynomial codes over certain finite fields, J. Soc. Ind. Appl. Math., vol. 8, no. 2, pp. 300–304.

Rivera E, López HR, 2012, Evidencia de propiedades fractales en la sucesión de Fibonacci usando wavelets, Sci. Tech., vol. 17, no. 52, pp. 122–128.

Rodríguez Andina JJ, Torre Arnanz E, Valdés Peña MD, 2017, FPGAs Fundamentals, Advanced Features, and Applicationes in Industrial Electronics. CRC Press.

Sandoval C, Fedón A, 2007, Codificador y decodificador digital Reed-Solomon programados para hardware reconfigurable, Ing. y Univ., vol. 11, no. 1, pp. 17–32.

Sandoval C, Fedón A, 2008, Programación VHDL de algoritmos de codificación para dispositivos de hardware reconfigurable, Rev.Int.Mét.Num.Cálc.Dis.ing., vol. 24, no. 1, pp. 3–11.

https://upcommons.upc.edu/bitstream/handle/2099/10415/I%20-%20V24N1.pdf

Sandoval C, 2010, Multiplicador paralelo en campos finitos de Galois GF (2m), Congr. Investig. UC, no. 1, pp.

1706–1711.
Sandoval-Ruiz C, 2012, Codificador RS (n,k) basado en LFCS : caso de estudio RS (7,3), Rev. Fac. Ing. Univ. Antioquia, no. 64, pp. 68–78.
Sandoval-Ruiz C, 2013, Modelo Optimizado del Codificador Reed-Solomon (255,k) en VHDL a través de un LFSR paralelizado, Tesis Doctoral, Universidad de Carabobo, Venezuela.
Sandoval-Ruiz C, Fedón-Rovira A, 2013, Modelo fractal de un codificador Reed Solomon, VIII Congr. Nac. y 2do Congr. Int. la UC, pp. 1–12.
Sandoval-Ruiz C, Fedón-Rovira A, 2014, Efficient RS (255 ,k) encoder over reconfigurable systems, Rev.Téc.Ing.Zulia, vol. 37, no. 2, pp. 151–159. www.scielo.org.ve/pdf/rtfiuz/v37n2/art07.pdf
Sandoval-Ruiz C, 2014, Power Consumption Optimization in Reed Solomon Encoders over FPGA, Lat. Am. Appl. Res., vol. 44, no. 1, pp. 81–85.
Sandoval-Ruiz C, 2016, Modelo de Estructuras Reconfigurables con Registro Desplazamiento, para Lenguaje Descriptor de Hardware VHDL, Rev. Fac Ing UCV, vol. 31, no. 3, pp. 63–72.
http://saber.ucv.ve/ojs/index.php/rev_fiucv/article/view/15488/144814482166
Sandoval-Ruiz C, 2017, VHDL Optimized Model of a Multiplier in Finite Fields, Ing. y Univ., vol. 21, no. 2, pp. 195–211.
Sandoval-Ruiz C, 2017, Análisis de Circuitos Fractales y Modelado a través de Sistema de Funciones Iteradas para VHDL, Rev. Cienc. e Ing., vol. 38, no. 1, pp. 3–16.
Sandoval-Ruiz, 2017, VHDL Model of configurable neural networks applied to decoding in cognitive radio, Rev. Ing. UC, vol. 24(3). Pp. 290-301.
http://servicio.bc.uc.edu.ve/ingenieria/revista/v24n3/art02.pdf
Sandoval-Ruiz C, 2017, Logical-Mathematical Model of Encoder 2D-RS for Hardware Description in VHDL, Rev. Ing. UC, vol. 24, no. 1, pp. 28–39. http://servicio.bc.uc.edu.ve/ingenieria/revista/v24n1/art04-124.pdf
Sandoval-Ruiz C, 2017, Módulos de Procesamiento de Señales en VHDL aplicando Redes Neuronales para sistemas SDR, Revista Facultad de Ingeniería UCV, Vol. 32(1), pp. 17-26.
Sandoval-Ruiz C, 2018, Control de Micro-Redes de Energía Renovable a través de estructuras LFCS Reconfigurables en VHDL, Cienc. y Tecnol., Vol. 18, pp. 71–86.https://www.palermo.edu/ingenieria/pdf2018/CyT_18_05.pdf
Sandoval-Ruiz C, 2018, Códigos Reed Solomon para Sistemas Distribuidos de Energías Renovables y Smart Grids a través de Dispositivos Electrónicos Inteligentes sobre Tecnología FPGA, Memoria. Investigaciones en Ingeniería, Vol. 16, pp. 37-54.
http://www.um.edu.uy/docs/Codigos_Reed_Solomon_para_Sistemas_Distribuidos_de_Energias_Renovables_y_Smart_Grids.pdf
Sandoval-Ruiz C, 2019, Modelo VHDL de Control Neuronal sobre tecnología FPGA orientado a Aplicaciones Sostenible, Ingeniare. Rev. Chilena de Ingeniería, Vol. 27(3).
Sandoval-Ruiz C, 2019, Métodos Numéricos en Diferencias Finitas para la Estimación de Recursos de Hardware FPGA en arquitecturas LFSR(n,k) Fractales. Ingeniería Investigación y Tecnología, Vol. XX (03), pp. 1-10 http://www.revistaingenieria.unam.mx/numeros/2019/v20n3-08.pdf
Sandoval-Ruiz C, 2019. Plataforma de investigación de redes eléctricas reconfigurables de energías renovables aplicando modelos LFSR. Universidad Ciencia y Tecnología, 23(95), 103-115.
Sandoval-Ruiz C, 2020. Proyecto Cometa Solar – CS para optimización de Sistema Fotovoltaicos. Universidad, Ciencia y Tecnología, 24(99).
Sandoval-Ruiz C, 2020. Arreglos Fotovoltaicos Inteligentes con Modelo LFSR-Reconfigurable. Ingeniería: Revista de la Universidad de Costa Rica, 30(2).
Sandoval-Ruiz C, 2020. LFSR-Fractal ANN Modelapplied in R-IEDs for Smart Energy. IEEE Latin Ame-rica Transactions, VOL. 18.
Tejeda-calderón VC, MA García-martínez, and R Posada-gómez, Implementación en FPGA de un multiplicador por dígitos sobre Campos Finitos GF (2m), pp. 2–5.
Velásquez F, Castaño JF, 2014, Implementacion de aritmetica de torres de campos finitos binarios de extension 2. Visión electrónica, 7(2), 89-96. https://doi.org/10.14483/22484728.5513
Yeh CS, IS Reed, TK Truong, 1984, Systolic multipliers for finite fields GF (2m), IEEE Trans. Comput., vol. 4, pp. 367–360.
Xilinx 2018, Performance and Resource Utilization for Reed-Solomon Encoder v9.0 LogiCORE IP Product Guide, Design Suite Release 2018. https://www.xilinx.com/support/documentation/ip_documentation/ru/rs-encoder.html#virtex7.

Recibido: *08 de octubre de 2019*

Aceptado: *25 de enero de 2020*

Sandoval-Ruiz, Cecilia E. *Profesora en Postgrado de Ingeniería UC, egresada de la Universidad de Carabobo de Ingeniero Electricista en 2002, Magister en Ingeniería Eléctrica en 2007 y Doctora en Ingeniería en 2014. Investigadora acreditada en el PEII - Nivel C, áreas de investigación: diseño sostenible y configuración de hardware*.