

MODELO DE ESTRUCTURAS RECONFIGURABLES CON REGISTRO DESPLAZAMIENTO PARA LENGUAJE DESCRIPTOR DE HARDWARE VHDL

CECILIA SANDOVAL-RUIZ

Universidad de Carabobo, Facultad de Ingeniería, Dirección de Postgrado.

e-mail: cecisandova@yahoo.com

Recibido: febrero 2016

Aprobado para publicación: julio 2016

RESUMEN

En esta investigación se presenta el modelo generalizado para hardware reconfigurable de generadores de secuencia, basados en registros desplazamiento con realimentación lineal – LFSR, con el objetivo de lograr su descripción en VHDL (Very high speed integrated circuit Hardware Description Language). La metodología empleada para el proceso de modelado consistió en un análisis comparativo entre varios circuitos operadores de datos: convolución, correlación, reducción modular y codificación Reed Solomon. Se realizó la descripción de su comportamiento en VHDL y se establece la similitud entre las diversas estructuras. Entre los resultados se obtuvo la validación a través de simulación del modelo VHDL para la operación de convolución y multiplicación en campos finitos de Galois. Se obtuvieron las ecuaciones que describen cada aplicación, se estableció la correspondencia entre los componentes del modelo, en función de las variables k muestras de entrada e i posiciones de las ramas en la arquitectura, empleando como método la generalización de la expresión matemática y la aplicación de los operadores lógicos y el operador “concatenación” disponibles en VHDL. La investigación permitió establecer el modelo optimizado de un LFSR reconfigurable para diversas aplicaciones en el área de la ingeniería y la ciencia, con ventajas como la eficiencia energética de los circuitos y la implementación en hardware de manera concurrente.

Palabras clave: modelo LFSR reconfigurable, VHDL, generadores de secuencia, operador de convolución, operador de correlación, multiplicador GF, codificador reed solomon

MODEL OF RECONFIGURABLE STRUCTURES WITH SHIFT REGISTER, FOR HARDWARE DESCRIPTION LANGUAGE VHDL

ABSTRACT

This research presents the generalized model for reconfigurable hardware of sequence generators, based on Linear Feedback Shift Register - LFSR, in order to obtain its description in VHDL (Very high speed integrated circuit Hardware Description Language). The methodology used for the modeling process consisted of a comparative analysis among several circuits data operators: convolution, correlation, modular reduction and Reed Solomon coding. The description of its behavior in VHDL was made, and the similarity between the various structures was established. Among the results, the validation was obtained through simulation of the VHDL model for the convolution and multiplication operations in Galois finite fields. The equations that describe each application were obtained, the correspondence between the components of the model was established, as a function of the variables k input samples and the i positions of the branches in the architecture, using as method the generalization of the mathematical expression and the application of the logical operators and the “concatenation” operator available in VHDL. The research allowed establishing the optimized model of a reconfigurable LFSR for various applications in the area of engineering and science, with advantages such as the energy efficiency of circuits and hardware implementation concurrently.

Keywords: model reconfigurable LFSR, VHDL, sequence generators, convolution operator, correlation operator, GF multiplier, reed solomon encoder

INTRODUCCIÓN

Actualmente, se evidencia una necesidad de reducir el consumo de energía de los circuitos de procesamiento, así como promover los criterios de diseño eficiente. Por lo que se plantea el desarrollo de un modelo para lenguaje descriptor de hardware -VHDL, partiendo de identificar en la estructura circuital de diversas aplicaciones, la similitud entre sus componentes modulares, que correspondan con la misma arquitectura, aunque con diferente dimensión entre sus operadores.

Estudios previos (Goresky & Klapper, 2004, 2002) han generado una arquitectura alternativa para dispositivos de estado finito, conocida como FCSR – Feedback with carry shift registers. Recientemente, (Arnault, Berger, Lauradoux, Minier, & Pousse, 2009) han propuesto una versión que generaliza las representaciones de Fibonacci y Galois para la arquitectura del FCSR. Así mismo, en (Pérez & García, 2007) se ha desarrollado el diseño de un modelo generalizado para la estructura del codificador de convolución, CRC y Reed Solomon, para su descripción en hardware, lo que ratifica la tendencia hacia la generalización de estructuras configurables.

En este sentido, se puede observar que los circuitos generadores de secuencia basados en estructuras LFSR (Linear Feedback Shift Register) corresponden al componente básico de operadores matemáticos sobre secuencias. Esta estructura puede configurarse con características específicas para aplicaciones como: operadores de convolución, y correlación, operación de reducción modular presente en multiplicadores en álgebra de Galois (Sandoval, 2010), así como en filtros transversales de procesamiento digital de señales y codificadores Reed Solomon (Sandoval & Fedón, 2007). A partir de esta similitud en la arquitectura de dichas aplicaciones, se ha considerado pertinente desarrollar un modelo generalizado para la función de generación de secuencias que permita la optimización de los diseños asociados.

ESTRUCTURA DE LOS GENERADORES DE SECUENCIA

El objeto de estudio corresponde con los elementos operadores de secuencia, los cuales generan una salida específica (según la función de la aplicación), de acuerdo a los coeficientes de un polinomio característico. La arquitectura de éstos corresponde a una configuración de operaciones sobre un registro desplazamiento con realimentación lineal.

Para el modelado del LFSR es necesario expresar el comportamiento del circuito secuencia de n etapas con m bits cada una de forma genérica. Siendo necesario definir los parámetros de escalabilidad del circuito, en función del número de etapas y tamaño del bus. En base a esta necesidad, se han encontrado desarrollos en técnicas de factorización de las funciones y operaciones comunes que dan paso al concepto de LFSR reconfigurables (Alaus, Oguet, & Alicot, 2008), lo que permite la reutilización de hardware del circuito R_LFSR (Figura 1).

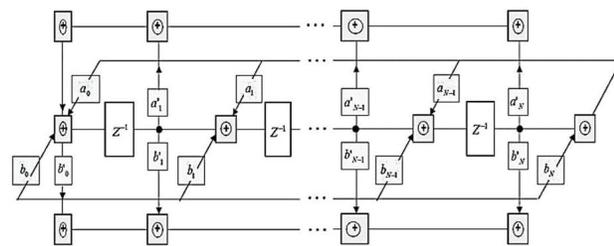
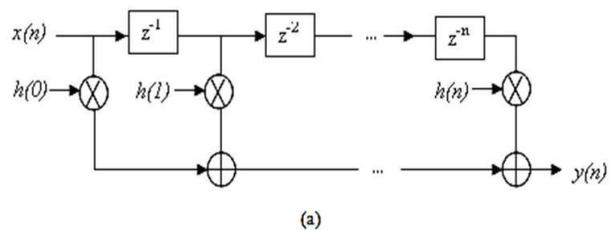


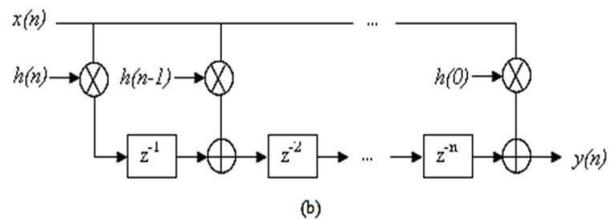
Figura 1. Arquitectura del R-LFSR

Fuente: Alaus et al., 2008

La parametrización del LFSR puede ser aplicada para la configuración de hardware en VHDL, con un tratamiento genérico de sus componentes, y describir las variantes de configuración, a través de habilitadores y elementos de memoria, operadores y longitud de elementos variable. En el caso de la configuración Fibonacci (Figura 2.a) se selecciona el sentido de desplazamiento L-M, mientras que en el caso de la configuración Galois (Figura 2.b) se selecciona M-L.



(a)



(b)

Figura 2. Esquema LFSR (a) Configuración Fibonacci. (b) Configuración Galois

MÉTODO DE DESCRIPCIÓN DEL GENERADOR DE SECUENCIA EN VHDL

El primer paso consistió en establecer la correspondencia entre las aplicaciones para definir un circuito con módulos selectivos para cada generador de secuencia, seguido de la descripción en VHDL del comportamiento secuencial, generalizado bajo el concepto de arquitectura base LFSR y su descripción concurrente.

I caso de estudio: operador de convolución

La convolución es la operación matemática que permite obtener la respuesta $y(n)$ de un sistema lineal invariante en el tiempo como función de la entrada $x(n)$ y la respuesta del sistema al impulso unitario $h(n)$, quedando expresada como $x(n)*h(n)$. Para el análisis de la implementación de este operador, se requirió una re-interpretación de su arquitectura, partiendo de su modelo matemático, dado por:

$$y(n) = \sum_{k=-\infty}^{+\infty} x(n-k) \cdot h(k) \quad (1)$$

La cual también se puede implementar bajo la configuración expresada como:

$$y(n) = \sum_{k=-\infty}^{+\infty} x(k) \cdot h(n-k) \quad (2)$$

Ésta corresponde a la sumatoria de los productos de las k muestras de la entrada de datos $x(n)$ con los n elementos de la respuesta al impulso unitario $h(n)$, reflejada y desplazada según el número de muestra. De esta manera, se encuentra la expresión definida por la operación de convolución entre las funciones estudiadas y la similitud entre el modelo circuital basado en la arquitectura del LFSR bajo la representación de Galois. Destacando que la secuencia de salida será representada en VHDL bajo la forma:

$$y_n = \&zk(i), \text{ con } k \text{ desde } 0 \text{ a } k+n-1 \quad (3)$$

Se realizó la descripción del hardware para el circuito de comportamiento secuencial de la operación convolución, para ambos esquemas, bajo la configuración Galois (a) y la configuración Fibonacci (b). El código configurado en VHDL para el procesamiento de las señales es presentado en la Tabla 1.

Tabla 1. Descripción VHDL del circuito secuencial

Modelo 1: $y(n) = \sum x(k) \cdot h(n-k)$

```

if hab = '1' then
  yn <= "0000";
else
  -- Producto de x(k) * h(i)
  mk0 <= x(k) * h(0);
  mk1 <= x(k) * h(1);
  ...
  mkn <= x(k) * h(n);
  -- Desplazamiento de los registros r(n-k)
  if (clk'event and clk='1') then
    r_v(n) := r_v(n-1);
    ...
    r_v(1) := r_v(n-1) + mk1;
    r_v(0) := r_v(n-1) + mk0;
    y_v(k) := r_v(0); -- sumatoria k de sec. producto
  end if;
end if;
--Concatenación de yk(0)
yn <= y_v7(0) & y_v6(0) & ... & y_v0(0);
-- en el ejemplo yn = v0 + v1 + v2 - v3

```

Modelo 2: $y(n) = \sum x(n-k) \cdot h(k)$

```

if hab = '1' then
  yn <= "0000";
else
  -- Desplazamiento de la entrada x(n-k)
  if (clk'event and clk='1') then
    x_v(n) := x_v(n-1);
    ...
    x_v(2) := x_v(1);
    x_v(1) := x_v(0);
    x_v(0) := xn;
  --Producto de la entrada retardada con h(k)
  v0 <= x_v(0) * h(0); -- ejemplo h(0) = "01"
  v1 <= x_v(1) * h(1); -- ejemplo h(1) = "10"
  ... -- ejemplo h(2) = "01"
  vn <= x_v(n) * h(n); -- ejemplo h(3) = "01"
  end if;
end if;
--Sumatoria de las secuencias producto
yk <= v0 + v1 + ... + vn;

```

Se define la operación de convolución entre las secuencias dadas por: $x(n)=1,2,3,1$ y $h(n)=1,2,1,-1$ (Figura 3).

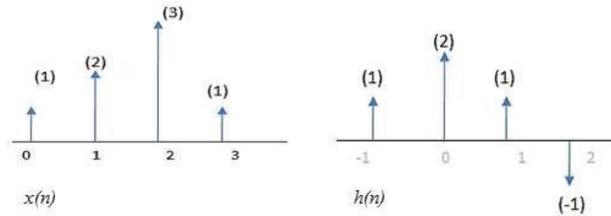


Figura 3. Representación gráfica de las secuencias

Siguiendo el procedimiento correspondiente a: (1) se refleja una de las secuencias: $h(n) \Rightarrow h(-k)$, (2) se desplaza la secuencia n muestras: $h(n-k)$, (3) se multiplica por la otra secuencia a operar, a fin de obtener la secuencia producto vn y, finalmente, (4) se suman los términos de la secuencia producto (Proakis & Manoslakis, 1998). Se obtuvo así la salida correspondiente $y(n)$ presentada en la Tabla 2.

Tabla 2. Comportamiento del sistema para k muestras

k	y_3	y_2	y_1	$y(n)$
1	-1	1	2	1
2	-2	1	5	4
3	-3	1	7	8
4	-1	-2	3	8
5	0	-1	-2	3
6	0	0	-1	-2
7	0	0	0	-1

Se realizó la implementación circuital del operador de convolución bajo la estructura circuital LFSR, aplicado en la configuración Galois. Se definió $x(n)$ como la entrada al circuito generador de secuencia y los coeficientes asignados a las ramas de la estructura, fueron la secuencia $h(-n)$; es decir, la secuencia $h(n)$ reflejada, como se observa en la Figura 4.

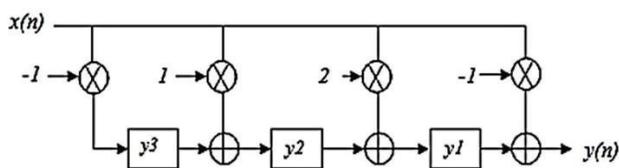


Figura 4. Esquema circuital de implementación

A partir de este esquema circuital, en el que se reconoce la correspondencia entre el operador de convolución y la estructura del generador de secuencia, se puede extrapolar la paralelización del circuito para definir la convolución de manera concurrente, a fin de optimizar el modelo en VHDL, obteniendo un procesamiento paralelo. La descripción VHDL del modelo concurrente es presentada en la Tabla 3.

Tabla 3. Descripción VHDL del circuito concurrente

$y(i) = y(i-1) + h(i) \cdot x(k)$ con $h(i) = h(k)$ y $fb = 0$	
...	
UM31: mult port map	(x3, h1,m31);
UM32: mult port map	(x3, h2,m32);
UM33: mult port map	(x3, h3,m33);
UM34: mult port map	(x3, h4,m34);
$y1(3) \leq m14;$	$y3(3) \leq m34;$
$y1(2) \leq y0(3) \text{ xor } m13;$	$y3(2) \leq y2(3) \text{ xor } m33;$
$y1(1) \leq y0(2) \text{ xor } m12;$	$y3(1) \leq y2(2) \text{ xor } m32;$
$y1(0) \leq y0(1) \text{ xor } m11;$	$y3(0) \leq y2(1) \text{ xor } m31;$
$yn \leq yk(i) \& \dots \& yk(1) \& yk(0);$	

Para validar el modelo desarrollado, se simularon las aplicaciones descritas en VHDL, donde se obtiene la salida $y(n) = 1, 4, 8, 8, 3, -2, -1$, cumpliendo con el resultado teórico correspondiente, destacando que el simulador expresa los resultados negativos en complemento-1 (Figura 5).

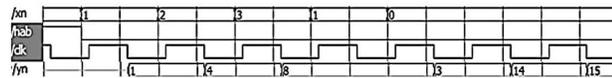


Figura 5. Simulación de un caso de convolución basado en la descripción en VHDL

II caso de estudio: operador de correlación

En (M. Pérez, 2009) se estudia la correlación de códigos, considerando que en su modo más sencillo un filtro acoplado, para detectar una señal en un medio con ruido blanco gaussiano, se puede implementar a través de un operador de correlación paralelo, observando su amplia utilidad en procesamiento digital de señales. La correlación de señales es una operación que se realiza en distintas áreas como radar, sonar, comunicaciones digitales, etc. Siendo esta operación matemática que permite medir el parecido entre dos señales. Esta operación es muy parecida a la convolución, ambas operan sobre dos secuencias (Proakis & Manoslakis, 1998).

De esta manera, los resultados en el modelado VHDL para el operador de convolución pueden ser aplicados en adaptaciones de otros operadores matemáticos para procesamiento digital de señales, tal es el caso del operador de correlación, en el cual, considerando que $x_1(n)$ y $x_2(n)$ son dos secuencias, entonces el operador de correlación Rx_1x_2 se define como:

$$Rx_1x_2 = \sum_{n=0}^L x_1(n) \cdot x_2(n-1) \quad (4)$$

Donde l es el parámetro de desplazamiento o retardo en el tiempo. Es decir, la secuencia $x_1(n)$ se mantiene fija y la secuencia $x_2(n)$ se desplaza l muestras, quedando expresada como $x_2(n-l)$, hacia la derecha si l es positivo y hacia la izquierda si l es negativo. Las secuencias se multiplican punto a punto.

Las similitudes entre el cálculo de la correlación cruzada y la convolución de dos secuencias son evidentes. En la correlación no se refleja la señal, sino que se realizan las mismas operaciones; es decir, solo se desplaza, se multiplica por la otra secuencia y finalmente, se suman los términos de la secuencia producto (Proakis & Manoslakis, 1998). Esta expresión en términos de la operación de convolución viene dada por:

$$Rx_1x_2 = x_1(l) * x_2(-l) \quad (5)$$

De esta manera, sustituyendo $h(n)$ por la secuencia $x_2(-l)$, se obtiene la descripción VHDL para esta operación de correlación, aplicando el modelo generado en VHDL para el primer caso de estudio por estar muy vinculados en la similitud de la estructura hardware. Su implementación puede ser de forma paralela, en el que se observa la configuración de Fibonacci (Figura 6).

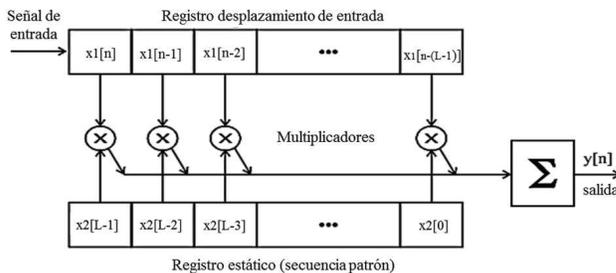


Figura 6. Diagrama de bloques de un correlador paralelo digital. Fuente: M. Pérez, 2009

El comportamiento de este circuito puede ser descrito en VHDL, definiendo la operación de forma paralela de los productos y la sumatoria final, descrito en la Tabla 4.

Tabla 4. Descripción VHDL de la correlación

y(i)= y(i-1)+h(i) · x(k) con h(i)=h(k) y fb=0

```
begin
...
if (clk'event and clk='1')then
    x1_v(n):= x_v(n-1);
    ...
    x1_v(2):= x1_v(1);
    x1_v(1):= x1_v(0);
    x1_v(0):=x1n;
--Producto de las secuencias
    v0<= x1_v(0) * x2(0);
    v1<= x1_v(1) * x2(1);
    ...
    vn<= x1_v(n) * x2(n);
end if;
end if;
--Sumatoria de los productos parciales
y1<= v0 + v1 + ... + vn;
yn<=y0 & y1 &... yL-1
```

Para tal caso, se requiere de un procesador de altas prestaciones. Para una secuencia con un gran número de muestras, como alternativa se puede implementar de forma secuencial, a través de un correlador serie, donde para cada ciclo de reloj se computa una muestra con el resultado acumulado (M. Pérez, 2009).

III caso de estudio: reducción modular en GF

Así mismo, se puede notar la correspondencia en la función de reducción modular mod, con la estructura LFSR Galois. Siendo $P(x)$ el polinomio irreducible (define los coeficientes del LFSR), 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 reducción modular con $P(x)$, siendo está implementada en un LFSR a través de registros para obtener los n -vectores con corrimiento de los bits. En esta configuración se basa el análisis del comportamiento del circuito en m ciclos de reloj, donde m es igual al número de bits de la palabra. Esto permite expresar el modelo matemático para definir el comportamiento del multiplicador sobre campo finito orientado a la descripción en VHDL.

Este circuito permite la implementación del 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. Se puede expresar la operación mod como la convolución realimentada, expresada como:

$$a(x) \bmod p(x) = \sum_{k=0}^m a(k) \cdot p(n-k) \quad (6)$$

De esta manera, las secuencias a operar corresponden al operando $a(x)$ y $p(x)$. Esta última secuencia se comporta como los coeficientes del LFSR que permite definir las propiedades del multiplicador por medio de la reducción del resultado dentro del campo finito. La importancia de esta expresión en la definición del modelo consiste en establecer una relación entre el comportamiento secuencial del procesamiento con un modelo matemático concurrente a fin de lograr la descripción de forma eficiente. La principal diferencia entre ambos circuitos de implementación viene dada por la realimentación presentada en el caso de la operación módulo (Figura 7).

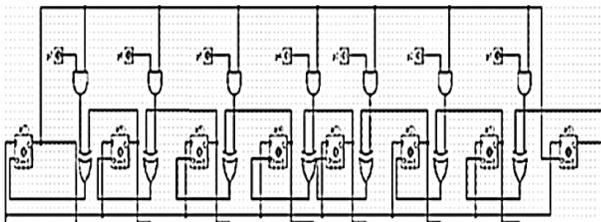


Figura 7. Implementación circuital del operador mod en el multiplicador en campos finitos GF

De esta manera, el tratamiento de la operación mod será dada como la convolución con $p(x)$ del elemento $a(m)$ realimentado. Donde se puede expresar como:

$$a_m(x) = \sum_{i=0}^m (a(i) \oplus a(m)) \cdot p(n-i) \quad (7)$$

Al desarrollar la implementación circuital del multiplicador en VHDL, es de vital importancia el comportamiento del generador de secuencia correspondiente a la operación de reducción modular (a nivel de operadores lógicos) basados en la descripción de cada término (para cada muestra de la secuencia de entrada $a(x)$ y su concatenación, con el propósito de realizar una implementación de manera concurrente, estableciendo así la descripción VHDL del operador mod, presentado en la Tabla 5.

Tabla 5. Reducción Mod concurrente en VHDL

```
p<="100011101";
-- Primitive polynomial = D^8+D^4+D^3+D^2+1 (285)
u0: a1<=D_dato
-- para los términos cuyo coeficiente de p(x) ≠ 0:
u1: a2<=a1(6 downto 4)&(a1(3)xor a1(7))&(a1(2)xor
a1(7))&(a1(1)xor a1(7))&a1(0)& a1(7);
...
u7: a8<=a7(6 downto 4)&(a7(3)xor a7(7))&(a7(2)xor
a7(7))&(a7(1)xor a7(7))&a7(0)& a7(7);
```

De la simulación del comportamiento del multiplicador GF descrito en VHDL se obtuvo la salida de datos correspondiente a los valores teóricos esperados para el producto de la secuencia de 0 a 16 por el coeficiente 13 (Figura 8).

/test13/d_dato	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
/test13/datox	0	13	26	23	52	57	46	135	104	101	114	127	92	81	70	75	208

Figura 8. Validación del multiplicador GF

Luego de validar el funcionamiento, se estudió la optimización del multiplicador, donde la implementación para el caso $B(x) \bmod p(x) \cdot A(x)$, presenta un componente de potencia de la señal de 0,38 mW y en el caso de $A(x) \bmod p(x) \cdot B(x)$, presenta componente de potencia de señal de 0,35 mW (Figura 9). La optimización se consigue a través de la técnica de la reordenación de las señales y paralelización del diseño sin emplear señal de reloj clk .

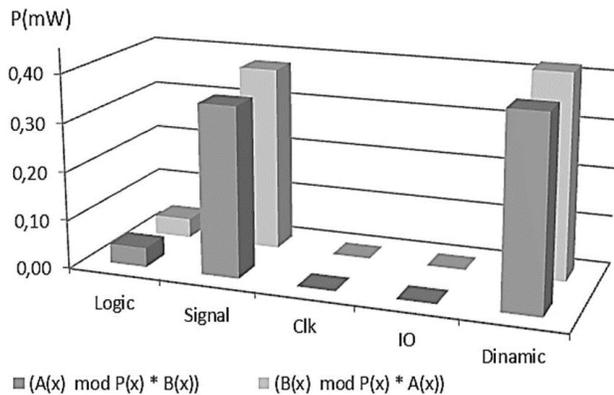


Figura 9. Optimización del multiplicador GF

Aplicación sobre símbolos en codificadores RS

Otra aplicación con estructura similar pero con operadores establecidos (producto GF). Es el caso del codificador Reed Solomon, en el cual se puede evidenciar la estructura LFSR, en la generación de los símbolos de redundancia, encontrando una correspondencia entre las estructuras del generador de código y sus operadores multiplicadores (Sandoval-Ruiz & Fedón-Rovira, 2013). Partiendo de la ecuación de convolución, basado en el LFSR de 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 $h(n-k)$ por los coeficientes del polinomio generador del código $g(n-k)$, obteniendo así una expresión para los $n-k$ símbolos generados, dada por:

$$R(x) = \sum_{k=0}^n d(t) \cdot g(n-k) \quad (8)$$

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 $rk-1(0)$. Esto con el propósito de conservar la similitud de la expresión matemática (sin realimentación). Por otra parte, la propuesta del esquema modular del elemento circuital LFSR es presentado en (Sandoval-Ruiz, 2013a) donde se destacan habilitadores para expandir el número de etapas. El modelo VHDL generalizado se presenta en la Tabla 6.

Tabla 6. Descripción VHDL del codificador RS

-- $r(i)=r(i-1)+g(i).(x(k)+r(n-1))$ con $c(i)=g(i)$	
UM11: mult port map	(d1,"100",m11);
...	
UM31: mult port map	(d3,"100",m31);
UM32: mult port map	(d3,"111",m32);
UM33: mult port map	(d3,"111",m33);
UM34: mult port map	(d3,"101",m34);
r1(3)<=m14;	r3(3)<=m34;
r1(2)<=r0(3) xor m13;	r3(2)<=r2(3) xor m33;
r1(1)<=r0(2) xor m12;	r3(1)<=r2(2) xor m32;
r1(0)<=r0(1) xor m11;	r3(0)<=r2(1) xor m31;
cn<= rk(i) & ... & rk(1) & rk(0);	

Finalmente, el comportamiento de los diversos casos de estudios considerados en esta investigación se pueden resumir en una estructura LFSR con características dadas

por la secuencia binaria o de símbolos (de m bits para el caso del codificador) y el componente a realimentar según sea la aplicación particular. En la Tabla 7 se presenta un resumen de los casos estudiados.

Tabla 7. Características de aplicaciones LFSR

Generador LFSR	Sec. LFSR	Sec. de Entrada	FB	Nro. bits
Convolución	$h(n)$	$x(n)$	-	1
Correlación	$x2(n)$	$x1(n)$	-	1
Reducción mod	$p(x)$	$a(x)$	$a(m)$	1
Codificador RS	$G(x)$	$D(x)$	$rk-1(0)$	8

De esta manera se puede visualizar a través de los parámetros definidos en cada uno de los casos estudiados, la correspondencia entre las estructuras, de acuerdo a la configuración de cada LFSR, con el propósito de establecer un modelo en VHDL generalizado para operaciones de generación de secuencias.

MODELO DE LA ESTRUCTURA LFSR RECONFIGURABLE PARA VHDL

Apartir del estudio de señales, arquitectura y comportamiento de las aplicaciones, se reconocen las configuraciones requeridas para la generalización, como son los selectores para la entrada, salida de datos y realimentación de la señal, tal como se presenta en el modelo circuital (Figura 10).

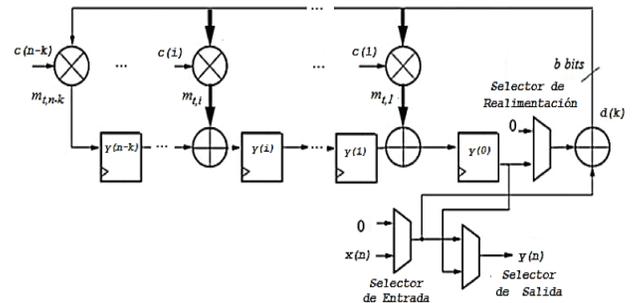


Figura 10. Esquema generalizado para operaciones de generación de secuencias

Luego de las características de cada arquitectura, se define el comportamiento para su descripción en VHDL tratada por el número de muestras k , mientras que la variable i permite tener una relación con respecto a la posición espacial del elemento en la estructura del polinomio, siendo la secuencia de salida de longitud $k+i-1$. El valor que tomará el vector de salida para cada posición frente a cada elemento de entrada será $y_k(i) = y_{k-1}(i+1)+d(k)$, se calcula según el valor del

elemento en su posición antecedente (según el sentido de desplazamiento), y el elemento realimentado $d(k)$.

Para el manejo de las aplicaciones se ingresa al circuito una señal de k elementos correspondiente a la entrada de datos $x(k)$ que será pre-seleccionada por el habilitador IN . Ésta será operada con el término de la realimentación según su habilitación para construir la señal realimentada $d(k)$ que será específica de acuerdo a los elementos seleccionados según la configuración; es decir, la señal resultante de la suma de la entrada (habilitada por IN) para los casos en que la aplicación procesa la entrada de datos y la señal retardada de la salida del arreglo de registros (habilitada por FB) en los casos donde que se aplica el circuito con realimentación lineal, destacando que el sentido de desplazamiento es selectivo por M/L el término más o menos significativo y que la salida corresponde a la concatenación de términos generados por el modelo:

$$y_k(i) = y_{k-1}(i \pm 1) + c(i) \cdot d(k),$$
$$\text{con } d(k) = [x(k) \cdot IN + y_{k-1}(M/L) \cdot FB] \quad (9)$$

Dicho modelo matemático representa la generalización de las ecuaciones estudiadas de los generadores de secuencia con base a la estructura LFSR que permite la descripción en VHDL del comportamiento secuencial de las aplicaciones estudiadas. Con éste se puede realizar la adaptación al modelo concurrente calculando los coeficientes para cada elemento, según sea el caso, a fin de configurar el hardware para procesamiento paralelo, tal como se realizó en el multiplicador GF.

CONCLUSIONES

La investigación desarrollada es un aporte para aplicaciones sobre tecnología de hardware reconfigurable, ya que facilita la adaptabilidad de los productos finales, aspecto importante en los procesos continuos de actualización. Además de cumplir con la sintaxis en VHDL, ésta a su vez ofrece portabilidad de los generadores de secuencias a diversas plataformas bajo el enfoque de implementaciones con hardware libre.

Se obtuvieron las ecuaciones que describen el código, con parámetros adaptables, ya que se determinó que los parámetros una vez asignados permiten optimizar el diseño en gran manera, motivo por el cual los resultados estarán dados en función de los coeficientes que definen el polinomio característico del circuito. Así se logra un diseño altamente eficiente y un modelo novedoso que aporta un método de configuración de hardware donde el caso de

estudio encontró una similitud matemática-circuital en las funciones de realimentación LFSR.

Se destaca que las aplicaciones estudiadas realizan su procesamiento bajo arreglos particulares de una arquitectura circuital común de generadores de secuencia configurado sin realimentación en el caso de los operadores de convolución y correlación, así como LFSR en los casos de la operación de reducción modular (C. Sandoval-Ruiz, 2017) y codificador Reed Solomon (C. E. Sandoval-Ruiz, 2017a), donde se busca mantener la estructura con selección de las condiciones específicas para establecer un modelo generalizado de las estructuras auto-similares (C. E. Sandoval-Ruiz, 2017b).

Los resultados obtenidos pueden ser empleados en otras aplicaciones en el área de la ingeniería y la ciencia como un *filtro transversal* para control adaptativo (Castellanos, Sandoval, & Azpurua, 2014). Éste presenta una estructura LFSR Fibonacci, con pesos dados por coeficientes del generador de secuencia y k muestras de la señal de entrada, y para diversos generadores de secuencia en codificadores de datos y criptografía (Sandoval-Ruiz, 2013b). Todas éstas con ventajas del diseño generalizado.

Así mismo, la generalización lograda en esta investigación ha permitido extender los avances alcanzados para la descripción concurrente para cada circuito (Sandoval-Ruiz, 2013a) hacia un conjunto de aplicaciones que presentan arquitectura similar (Sandoval-Ruiz & Fedón-Rovira, 2014), donde se ha reconocido la correspondencia entre los términos del modelo matemático y se ha generado un modelo genérico para la descripción VHDL de estas estructuras. Todo esto con un desempeño eficiente que ofrece alta velocidad y mejor rendimiento en el consumo de energía (Sandoval-Ruiz, 2012).

REFERENCIAS

- ALAU, L. A., OGUET, D. N., & ALICOT, J. P. (2008). Extended Reconfigurable Linear Feedback Shift Register Operators for Software Defined Radio. *Gestion*, (1), 1–5.
- ARNAULT, F., BERGER, T., LAURADOUX, C., MINIER, M., & POUSSE, B. (2009). A New Approach for FCSRs. *Selected Areas in Cryptography: 16th Annual International Workshop*, 1–20.
- CASTELLANOS, J., SANDOVAL, C., & AZPURUA, M. (2014). A FPGA Implementation of a LMS Adaptive Algorithm for Smart Antenna Arrays. *Revista Técnica de la Facultad de Ingeniería de la Universidad de Zulia, Venezuela*,

- 37(3), 270–278.
- GOESKY, M., & KLAPPER, A. (2004). Fibonacci and Galois Representations of Feedback with Carry Shift Registers, 1–31.
- GOESKY, M., & KLAPPER, A. M. (2002). Fibonacci and Galois Representations of Feedback-with-carry Shift Registers. *IEEE Transactions Information Theory*, 48(11), 2826–2836.
- PÉREZ, C., & GARCÍA, J. (2007). Bloques funcionales del reverse traffic channel basados en un modelo generalizado LFSR. *Revista avances en Sistemas e Informática*, 4(2), 89–95.
- PÉREZ, M. (2009). Generación y correlación eficiente de códigos binarios derivados de conjuntos de secuencias complementarias para sistemas ultrasónicos. Tesis Doctoral, Universidad de Alcalá, España.
- PROAKIS, J. G., & MANOSLAKIS, D. G. (1998). Tratamiento digital de señales (3rd ed.). Madrid: Editorial Prentice Hall. pp. 75-85. 118-131.
- SANDOVAL, C. (2010). Multiplicador paralelo en campos finitos de Galois GF (2m). *Congreso de Investigación UC*, (1), 1706–1711.
- SANDOVAL, C., & FEDÓN, A. (2007). Codificador y decodificador digital Reed-Solomon programados para hardware reconfigurable. *Ingeniería y universidad*, 11(1), 17–32.
- SANDOVAL-RUIZ, C. (2012). Codificador RS (n,k) basado en LFCS : caso de estudio RS (7,3). *Rev. Fac. Ing. Univ. Antioquia*, 64, 68–78.
- SANDOVAL-RUIZ, C. (2013a). Modelo concurrente de generadores de secuencia para descripción de hardware reconfigurable Cecilia. VIII Congreso Nacional y 2do Congreso Internacional de Investigación Universidad de Carabobo, 25–34.
- SANDOVAL-RUIZ, C. (2013b). 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 Congreso Nacional y 2do Congreso Internacional de la UC, 1–12.
- SANDOVAL-RUIZ, C., & FEDÓN-ROVIRA, A. (2014). Efficient RS (255 ,k) Encoder over Reconfigurable Systems. *Rev. Téc.Ing.Zulia*, 37(2), 151–159.
- SANDOVAL-RUIZ, C. (2017). VHDL Optimized Model of a Multiplier in Finite Fields. *Ingeniería Y Universidad*, 21(2), 195–211.
- SANDOVAL-RUIZ, C. E. (2017a). Logical-Mathematical Model of Encoder 2D-RS for Hardware Description in VHDL. *Revista Ingeniería UC*, 24(1), 28–39.
- SANDOVAL-RUIZ, C. E. (2017b). Análisis de circuitos fractales y modelado a través de sistema de funciones iteradas para VHDL. *Revista Ciencia e Ingeniería*, 38(1), 3–16.

