

MÓDULOS DE PROCESAMIENTO DE SEÑALES EN VHDL APLICANDO REDES NEURONALES PARA SISTEMAS SDR

CECILIA SANDOVAL-RUIZ

Instituto de Matemática y Cómputo Aplicado. Universidad de Carabobo, Valencia. Venezuela
e-mail: cecisandova@yahoo.com

Recibido: mayo 2017

Aprobado para publicación: junio 2017

RESUMEN

Este trabajo presenta el modelado, en lenguaje descriptor de hardware VHDL, de un conjunto de componentes en el dominio digital (en banda base) aplicables para radio definido por software, y una red neuronal reconfigurable para modelado neuronal descrita en VHDL. Se considera que estos modelos requieren ser configurados en el hardware, en este caso bajo la tecnología de arreglo de compuertas programables FPGA. El método empleado consistió en la descripción y prueba de las etapas del receptor. Se obtiene como resultado la descripción en VHDL de una gama de componentes que pueden ser configurados de acuerdo a combinaciones específicas de parte del sistema de comunicación. Estos diseños fueron validados a través de simulación y la red reconfigurable fue entrenada para obtener como resultados los parámetros de cada módulo diseñado. La investigación permitió concluir que el modelado en VHDL provee de versatilidad a los sistemas de comunicaciones definidos por software, en los que se puede incorporar elementos cognitivos de inteligencia artificial, para el manejo óptimo de las etapas adaptadas. El principal aporte está dado por los códigos en VHDL, de forma paramétrica que permite avanzar en estos diseños y, a la vez, proveer de una base para el desarrollo tecnológico y nuevos conocimientos en la materia.

Palabras clave: modelado en VHDL, redes neuronales dinámicas, radio definido por software, tecnología FPGA

VHDL SIGNAL PROCESSING MODULES APPLYING NEURONAL NETWORKS FOR SDR SYSTEMS

ABSTRACT

This paper presents the modeling of a set of components in the digital domain (in base band) applicable for software defined radio, and a reconfigurable neural network for neuronal modeling described in VHDL hardware descriptor language. It is considered that these models need to be configured in the hardware, in this case, under the FPGA programmable gate arrangement technology. The method consisted in the description and test of the receiver stages. The result is the description in VHDL of a range of components that can be configured according to specific combinations of part of the communication system. These designs were validated through simulation and the reconfigurable network was trained to obtain as results the parameters of each designed module. The investigation allowed to conclude that the modeling in VHDL provides versatility to software defined communication systems, in which it is possible to add cognitive elements of artificial intelligence, for the optimal management of the adapted stages. The codes in VHDL, in parametric form that allow to advance in these designs, while providing a basis for technological development and new knowledge in the field represent the main contribution.

Keywords: VHDL modeling, Dynamic Neural Networks, Software Defined Radio SDR, FPGA Technology

INTRODUCCIÓN

Las nuevas tendencias en sistemas de tercera generación presentan un estudio de antenas inteligentes y moduladores adaptativos, como elementos de diseño para estos sistemas

inalámbricos inteligentes (Blogh y Hanzo, 2002). Este tipo de tecnología puede ser integrada con aplicaciones de redes neuronales (Hagan, Demuth y Beale, 1995). A fin de diseñar sistemas resultan altamente eficientes, tal como se presentan en algunas investigaciones que abordan el

diseño de codificadores y decodificadores a partir de redes neuronales (Wang y Wicker, 1996).

En las investigaciones en el área de sistemas de comunicación de datos, se ha encontrado, en los últimos años, el desarrollo de conceptos como Sistemas de Radio definido por Software (SDR, del inglés *Software Defined Radio*) y Radio Cognitivo (Fette, 2009). Estos conceptos tienen un punto en común, al proponer la reconfiguración de la arquitectura y parámetros del sistema, para adaptar las funciones de los módulos de forma dinámica, lo que ha motivado la indagación en sistemas de comunicación adaptativos y en la tecnología de soporte para diseños parametrizables y reconfigurables.

Los sistemas de comunicaciones SDR (Reis, Barros, Lenzi, Meloni y Barbin, 2012) nacen a partir del interés en esta área por parte de los radioaficionados, lo que motivó una investigación que dio origen al término *software radioK*. Al ser un modelo novedoso en los que la mayoría de las funciones son soportadas a través de software, estos sistemas presentan alta flexibilidad, ya que pueden ser adaptados, de acuerdo a las necesidades de los usuarios o condiciones del canal, reconfigurando sus modos de operación y características para alcanzar un mejor rendimiento. En esta investigación se propone un nuevo concepto en el que el control del sistema esté dado por Redes Neuronales (NNDR por sus siglas en inglés, *Neural Network Defined Radio*), con características de configuración de los sistemas SDR, para el manejo de los módulos de radiocomunicación. Los sistemas de comunicaciones orientados a radio cognitivo (Bonello, Cheny Hanzo, 2009; Hernández, Martínez, Fernando, Sarmiento yHernán, 2016) presentan un avance en el uso eficiente del canal de comunicación, donde se logra una disminución de la potencia de transmisión, gracias a la conmutación inteligente de la configuración de los módulos operativos, sean estos codificadores o moduladores. Esto con base en sistemas inteligentes, con características cognitivas donde se puede aplicar las redes neuronales, a fin de mejorar la eficiencia de la comunicación.

Es así como resultan de interés los avances en sistema de comunicaciones, con características adaptativas en los esquemas de modulación y codificación, que pueden ser ajustados dinámicamente a las condiciones del canal para mejorar el rendimiento del sistema (Noordin, Ali, Ismail y Jamuar, 2004), basados en arquitecturas *Very Large Scale Integration* (VLSI) (Mora, 2008), con una tendencia hacia la ingeniería basada en dispositivo reconfigurables.

Trabajos previos (Podosinov, 2011) presentan el desarrollo de funciones sobre tecnología FPGA para aplicaciones de

SDR. Así mismo, se han realizado investigaciones acerca de nuevos métodos de modelado en VHDL (Sandoval-Ruiz, 2017a). En este trabajo, se plantea como objetivo el desarrollo de modelos que pueden ser incorporados de manera selectiva, en la configuración de sistemas dinámicos de comunicación digital, a fin de establecer un aporte en el área de diseño reconfigurable y sistemas de radio inteligente, para eficiencia energética.

Antecedentes

En Sandoval (2007) se diseñaron las etapas del transmisor del sistema de comunicación digital en VHDL, alcanzando resultados en la codificación de fuente (Sandoval y Fedon, 2007a), codificadores de canal concatenados (Sandoval, 2008), con énfasis en el código Reed Solomon (Sandoval-Ruiz, 2017b), optimizaciones en RS(255,k) (Sandoval-Ruiz, 2013; Sandoval-Ruiz y Fedon-Rovira, 2014) y la etapa de modulación adaptativa (Diaz, Fernández y Sandoval-Ruiz, 2012), con lo cual se maneja un avance en materia de diseños en lenguaje VHDL, para el soporte de esta tecnología.

FUNDAMENTOS DEL MODELADO VHDL DE LOS SISTEMAS SDR

El estudio consiste en diseñar los componentes de un sistema de comunicaciones digitales, los módulos que pueden ser definidos por software (en banda base) y su configuración a través del control neuronal, así como la implementación de estos componente en VHDL a partir del modelo de comportamiento, como del diseño neuronal de módulo (Figura 1).

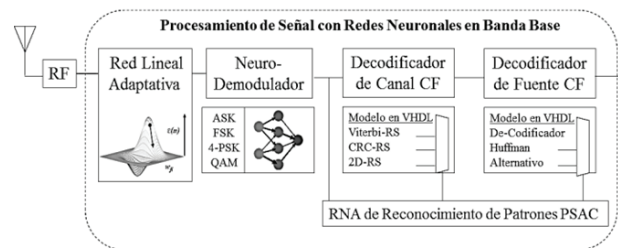


Figura 1. Esquema del sistema reconfigurable NNDS

Principios de funcionamiento de los módulos del SDR

Los módulos de estudio corresponden a una red neuronal lineal adaptativa, para el tratamiento de las señales de un arreglo de antenas inteligentes (Castellanos, Sandoval y Azpúrua, 2014), aplicando el algoritmo LMS, para entrenar las redes neuronales que puedan ajustar los pesos

sinápticos, a partir de la entrada de cada antena, y obtener a la salida del arreglo de antenas adaptativo la señal deseada. El esquema es presentado en la Figura 2.

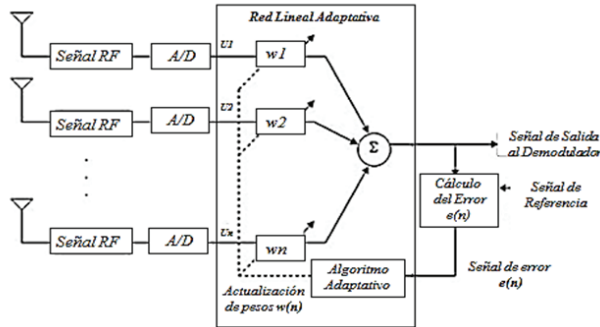


Figura 2. Esquema del arreglo de antenas inteligentes con Red Lineal Adaptativa

Red lineal adaptativa

Esta red neuronal está compuesta por una sola capa, donde se realiza la suma ponderada de las entradas y una polarización (*bias*). Su función de salida es lineal., La Ecuación 1 define su comportamiento.

$$a = \sum_{i=0}^R w_{j,i} p_i = \sum_{i=1}^R W_{j,i} P_i + b \quad (1)$$

La característica adaptativa viene dada por un algoritmo de aprendizaje LMS, el cual se basa en la actualización continua de los pesos sinápticos, proporcional al error cometido por la neurona. Su definición está dada por la Ecuación 2.

$$w_{new} = w_{old} + \alpha (t_q - a_q) p_q^T \quad (2)$$

La red adaptativa lineal puede ser configurada para diversas aplicaciones como estáticas o dinámicas. En este caso está definido como un filtro adaptativo lineal (una red dinámica), que incorpora una línea de retardo en la entrada de la red $p(k)$, cuya salida $a(k)$ está definida en la Ecuación 3.

$$a(k) = \sum_{i=0}^R w_{j,i} p_i(k-i) \quad (3)$$

Se puede observar la similitud del modelo neuronal con las estructuras LFSR (Sandoval-Ruiz, 2017a). Los siguientes elementos del estudio corresponden con los códigos CRC,

decodificación Viterbi y Reed Solomon, como elementos de configuración selectiva para códigos compuestos y demodulación basado en redes neuronales, para ampliar la gama de esquemas de codificación / modulación, proponiendo algoritmos de entrenamiento de la red neuronal de control, para ajuste de parámetros selectivos de cada módulo. Es decir, cada componentes tendrá ganancia ajustable, de manera que se seleccionará la configuración que participe en la red definida por el sistema NNDR.

Para la elaboración de los codificadores reconfigurables se describieron en VHDL los códigos base de los esquemas concatenados (Sandoval y Fedón, 2008; Sandoval, 2008). En este punto, el código de estudio correspondió al codificador secuencial y el esquema de codificación de bloques, como es el caso de los códigos Reed Solomon (Sandoval & Fedón, 2007b), así como la incorporación de uno de los códigos para la detección de errores más comunes y potentes como lo son los CRC (Stalling, 1997). Dado un mensaje de k bits, en el transmisor se genera una secuencia de n bits, denominada secuencia de comprobación de la trama (FCS, “*frame check sequence*”). El receptor realizará el mismo procesamiento de los datos para comprobar si la información es correcta, lo que equivale a dividir la trama recibida por el polinomio generador. La comprobación de los datos consiste en no producir residuo.

En el decodificador Viterbi se definen los casos de posibles secuencias erradas, donde la corrección se realiza en función de los estados presentes y las señales recibidas actuales y futuras. Estas son manejadas con un retardo sobre la señal de entrada de dos bits, es así como se establecen las correspondientes salidas, considerando el diagrama de estados del decodificador (Sandoval y otro, 2008). En el decodificador Reed Solomon, se establece el diagrama de bloques con los componentes, basados en funciones matemáticas para el cálculo del síndrome, se determina el error de los símbolos recibidos y la posición, para su corrección. Todos estos esquemas y comportamientos se definen en VHDL.

En el caso de los demoduladores, tomando en consideración moduladores adaptativos (Díaz y otros, 2012), se cuenta con la posibilidad de variar el esquema de modulación. Para la etapa de demodulación se parte del esquema de una red neuronal dinámica, en la cual se pueden manejar un número de muestras de la señal recibida, sobre la cual la red se entrena para obtener las salidas de-demoduladas, obteniendo una red que se entrena para cada esquema de modulación. En el caso de la modulación ASK (*Amplitudes Shift Keying*), se define la salida para la señal recibida, tal como se presenta en la Ecuación 4.

$$D(t)=\begin{cases} 1 & \text{Lógico para la señal } V_p \text{ sen}(2\pi f_1 t) \\ 0 & \text{Lógico para la señal } 0 \end{cases} \quad (4)$$

En tanto que para el esquema FSK (*Frequency-shift keying*), se aplica la Ecuación 5 en función del entrenamiento de la red neuronal.

$$D(t)=\begin{cases} 1 & \text{Lógico para } V_p \text{ sen}(2\pi f_1 t) \\ 0 & \text{Lógico para } V_p \text{ sen}(2\pi f_1 t) \end{cases} \quad (5)$$

Estos modelos neuronales, permiten implementar los módulos, sin necesidad de tener implementada en *hardware* cada configuración. Al mismo tiempo, pueden ser extrapolados para otros esquemas de demodulación, considerando aspectos de sincronización en los casos de señales moduladas por fase de la señal.

MÉTODO DE MODELADO EN VHDL

El método de diseño consistió en la descripción en VHDL de los componentes del sistema, iniciando por la red lineal del arreglo de antenas, siendo de interés el algoritmo adaptativo para el ajuste de los pesos de la neurona, se presenta el algoritmo para adaptación de la red en el circuito (Tabla 1).

Tabla 1. Descripción VHDL del algoritmo adaptativo

```
Nueva_ITERACIÓN: --Nueva pasada por el algoritmo
i <= i+1; -- iteración
n <= 0; -- patrón
Adaptación:
--Rutina de entrenamiento de los q patrones
x(n) <= leer puerto; -- patrón a entrenar (vector de R entradas)
q <= q + 1; -- incrementa el número de patrones
y(n) <= w(n) * x(n) + b(n); -- calcular la salida de la red
e(n) <= t(n) - y(n); -- error de la neurona para el patrón n
ea <= e(n) + ea; -- error acumulado
For i=1 to R -- para cada entrada i del patrón n
wi(n+1) <= wi(n) + u. xi(n). e(n); -- vector de 1 a R elementos
b(n+1) <= b(n) + u. e(n);
next
if clk' event and clk=1 then
w(n) <= w(n+1); -- actualiza el vector de pesos sinápticos
n <= n+1 -- próximo patrón
endif;
if n < q then -- patrones presentados de forma incremental
Goto Adaptación
endif;
et(i) <= ea / n; -- error promedio de la iteración i
if et(i) > 0.00999 then -- comprobación del error para detener el
entrenamiento
Goto nueva_ITERACIÓN -- si el error es mayor
else
salida_red <= y(n); -- mostrar por puerto la salida w(n) actualizado
endif;
```

El diseño de los módulos es para la configuración selectiva, en la que se habilita el módulo correspondiente según la señal recibida y reconocida por la red neuronal de control. De esta manera, se seleccionaron para la descripción en VHDL los decodificadores Viterbi, CRC y Reed-Solomon, como elementos básicos para las concatenaciones por su importancia y desempeño. En la Tabla 2, se presenta la descripción de comportamiento del decodificador Viterbi para corrección de errores.

Tabla 2. Descripción VHDL del decodificador Viterbi

IF ((A='1') and (R="01" or R="10"))then	IF ((C='1') and (R="01" or R="10")) then
if ((prox="01") or (prox="10")) then	if ((prox="00") or (prox="11")) then
next_s<='1';	next_s<='0';
next_B<='1';	next_A<='1';
elsif (prox="00" or prox="11") then	elsif (prox="01" or prox="10") then
next_s<='0';	next_s<='1';
next_A<='1';	next_B<='1';
end if;	end if;
end if;	end if;
IF ((B='1') and (R="00" or R="11")) then	IF ((D='1') and (R="00" or R="11")) then
if ((prox="01") or (prox="10")) then	if ((prox="00") or (prox="11")) then
next_s<='1';	next_s<='0';
next_D<='1';	next_C<='1';
elsif (prox="00" or prox="11") then	elsif (prox="01" or prox="10") then
next_s<='0';	next_s<='1';
next_C<='1';	next_D<='1';
end if;	end if;

En el caso del código CRC, este ha sido descrito en la Tabla 3. Definiendo como divisor $P = 110101$, (un patrón de $n+1$ bits, con n bits de comprobación) se establece el polinomio $P(x) = x^5 + x^4 + x^2 + 1$.

Tabla 3. Descripción VHDL del código CRC

```

architecture Behavioral of CRCciclico is
process (clk)
begin
...
elsif clk='1' and clk'event then
count <= count + '1';
q0<=datain xor q4;
q1<=q0;
q2<=q1 xor q4;
q3<=q2;
q4<=q3 xor q4;
q<=q4&q3&q2&q1&q0;
if count < "1010" then
mensaje<= mensaje(8 downto 0) & datain;
end if;
code<=mensaje & q;
end if;
end process;
end Behavioral;

```

En el análisis de comportamiento del decodificador Reed Solomon (n, k), se establece la jerarquía entre componentes y las señales que interactúan dentro de cada módulo. El diagrama de bloque del modelo del decodificador RS permite la descripción de los componentes en VHDL (Figura 3). Así mismo, este modelo permite definir la tabla de entrenamiento para un neuro-decodificador RS (Sandoval-Ruiz, 2017c).

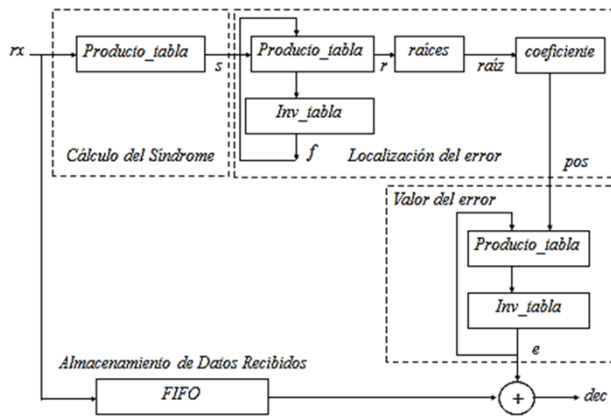


Figura 3. Esquema Modular del Decodificador RS

Se puede observar cómo se realiza la decodificación a través de tres módulos. El primero se denomina cálculo de síndrome, en el cual, conocidas las características del código, aplica la matriz inversa y se obtiene $2t$ símbolos que corresponde al síndrome. Estos valores actúan como coeficiente de un sistema de ecuaciones que permiten obtener las incógnitas f_1, f_2 a través de las cuales se genera el polinomio localizador de error. Este permite obtener los valores r_1, r_2 correspondientes a las raíces del polinomio $f(x)$ y a través de estos valores se obtiene la posición de los

error, pos y los valores de los errores e_1, e_2 . Estos, sumados en la posición de los errores a la trama recibida, permitirán obtener los datos corregidos, es decir; decodificada.

Finalmente, se estableció el entrenamiento de las redes neuronales a partir de los datos de entrada y salidas deseadas, basadas en el diseño en VHDL de una red dinámica de dos capas. Esta red es definida como red reconfigurable de propósito general, con parámetros ajustables de número de neuronas, retardos, matriz de pesos de entrada y capa para la realimentación, como se muestra en la Figura 4.

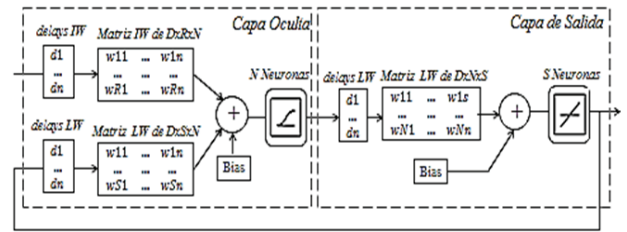


Figura 4. Red neuronal configurable para el diseño de los módulos

En esta red reconfigurable (de propósito general), se permite la configuración de acuerdo a las características del módulo funcional a implementar. Se considera que se pueden definir redes de una sola capa con función de salida escalón o lineal, siendo estas funciones tablas de búsqueda diseñadas en VHDL para obtener la salida de las neuronas. También se pueden configurar redes multicapa, en este caso de dos capas, en las que se pueden definir como redes dinámicas, con los retardos de las señales de entrada o entre capas, sean estas realimentadas (redes recurrentes) o de la capa oculta hacia la capa de salida (red con interacción hacia adelante) El número de neuronas, donde la dimensión de la matriz de pesos sinápticos vendrá dada por el número de las R entradas, por los D retardos y las N neuronas de la capa.

Para el caso de los codificadores secuenciales se definió una red dinámica, con retardos a la entrada y manejo de estados internos de la red. En el caso del codificador de bloque se seleccionó una configuración de red MPL paralela. Para la implementación de los demoduladores se seleccionó en el caso del ASK una red dinámica con retardos de entrada y realimentación y en el FSK una red con retardos distribuidos TDNN (Beale, Hagan y Demuth, 2011), para definir el entrenamiento a partir de las entradas/salidas del módulo.

RESULTADOS

Al hacer la simulación del decodificador de Viterbi diseñado, se observa la salida (trama recuperada) en función de los datos con los bits de redundancia insertados, tal como se muestra en la Figura 5.

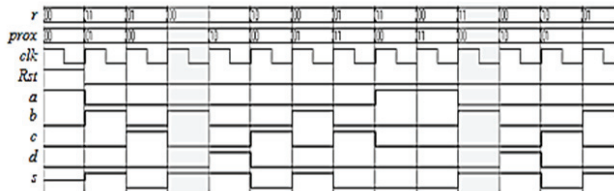


Figura 5. Simulación del decodificador Viterbi

Al analizar el comportamiento del decodificador Viterbi, se puede observar la señal recibida retrasada en el tiempo (r), la señal de entrada próxima ($prox$), que corresponde a la entrada recibida actualmente, los estados de la secuencia (a, b, c, d) y la señal de salida del decodificador (s). En el caso de la quinta entrada r , el estado corresponde a d , lo cual detecta un error que no está entre las entradas para la decodificación de este estado, sino entre las opciones $r=01$ y $r=10$. Es en base al dato $prox=10$ que se puede definir que el dato correcto a decodificar corresponde a $s=1$, con lo cual se corrige el error en la trama transmitida. Efectivamente, se ha logrado la correcta decodificación de los datos, incluso se ha introducido a propósito un error en los datos recibidos posición 5 y 12; de acuerdo al código creado este ha sido detectado y corregido, lo que permite validar el diseño.

Para el caso del decodificador CRC, se define una entrada de mensaje $M=1000010001101$, se obtiene el polinomio de mensaje $M(x)=x^9+x^7+x^3+x^2+1$. A este se le realiza el procedimiento a fin de implementar $T=2^nM+R$. Al llegar al receptor se realiza el mismo procedimiento y si hay resto de la operación de división se identificará el error (Figura 6).

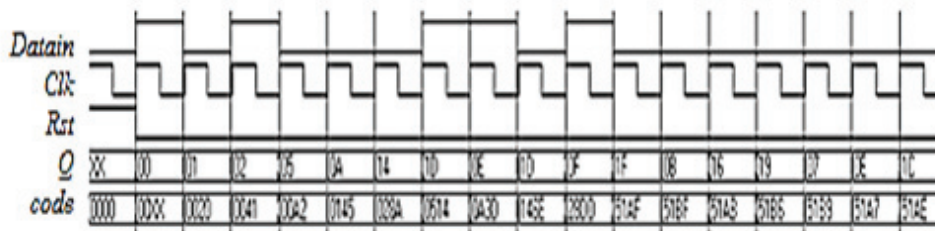


Figura 6. Simulación del decodificador CRC

Se observa la corrección del error en los resultados esperados, es decir; reprodujo a la salida el residuo correspondiente. Así mismo, al concatenar el mensaje con el FCS se obtiene $code=1010001101|01110$. Esto se traduce en 51AE en formato hexadecimal, como resultado de la simulación, lo que efectivamente permite concluir que el CRC programado genera los bits de redundancia de forma coherente para el mensaje de entrada, basado en la descripción de comportamiento realizada bajo sintaxis VHDL.

Al momento de simular el decodificador Reed Solomon, definido por la estructura (Figura 3), se tiene como resultado la simulación, con las señales del error y la posición para su corrección, como se observa en la Figura 7.

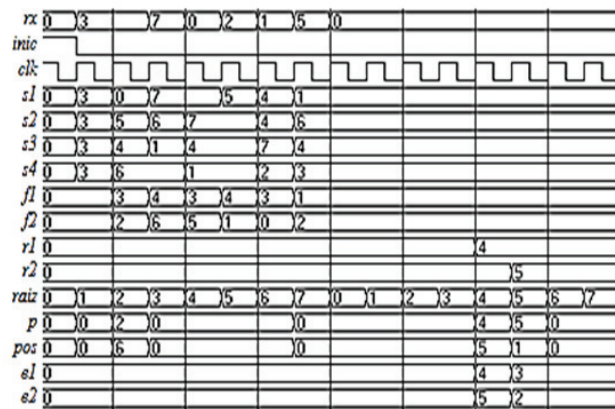


Figura 7. Simulación del decodificador Reed Solomon

Se obtuvo el modelado neuronal de los componentes estudiados, siendo este un aporte para el diseño de los sistemas de comunicaciones adaptativos, definidos a través de redes neuronales, en los que se encuentran las configuraciones de los módulos de cada etapa, como se muestra en la Tabla 4.

Tabla 4. Resultados de los módulos neuronales

CAPA OCULTA			
APLICACIÓN	RETARDOS	Neuronas	FUNCIÓN
Modulador ASK	Delays IW: -	20	sigmoide
	Delays LW: 10		
Modulador FSK	Delays IW: 4	5	sigmoide
	Delays LW: -		
Decodificador Viterbi	Delays IW: 1	4	sigmoide
	Delays LW: 1		
Decodificador Reed Solomon(7,3)	Delays IW: -	20	sigmoide
	Delays LW: -		
Arreglo de Antenas	Delays IW: -	-	-
	Delays LW: -		
CAPA DE SALIDA			
APLICACIÓN	RETARDOS	Neuronas	FUNCIÓN
Modulador ASK		1	Lineal
	Delays LW: 10		
Modulador FSK		1	Lineal
	Delays LW: 3		
Decodificador Viterbi		5	Lineal
	Delays LW: -		
Decodificador Reed Solomon(7,3)		3	Lineal
	Delays LW: -		
Arreglo de Antenas		1	Lineal

Los resultados de los modelos permitieron definir la estimación de neuronas requerida por cada una de las aplicaciones estudiadas. Se presentan los componentes de la arquitectura neuronal, en función del tipo de red, donde destacan la complejidad del diseño dado por las redes dinámicas con elementos de memoria. A través de estos se manejan los retardos y las redes multicapas, que presentan una configuración de neuronas en la capa oculta, así como redes lineales adaptativas para el caso del arreglo de antenas.

CONCLUSIONES

Gracias al diseño del conjunto de componentes del sistema de comunicación digital, bajo la descripción en VHDL para su implementación en hardware reconfigurable, se cuenta con los módulos descriptivos de comportamiento validados, para la configuración selectiva de diversos modelos de sistemas, así como la herramienta basada en una red de propósito general en VHDL para la configuración neuronal de los componentes modelados. Esto ofrece gran versatilidad al momento de adaptación en el circuito, a las condiciones del canal de comunicación.

De esta manera, se obtiene un nuevo concepto que parte de los sistemas de radio definidos por software, para establecer sistemas de radio definidos por redes neuronales. Este

enfoque innovador permite el desarrollo de esta tecnología incorporando en la base de los SDR, componentes modelados en lenguaje descriptor de hardware y redes neuronales como solución para la eficiencia del sistema dinámico.

Incluir en este trabajo el tema de una red neuronal de propósito general como elemento de partida para el diseño y entrenamiento de los componentes codificadores y moduladores, resulta un aporte importante, ya que permite definir estos módulos de alta complejidad sobre un hardware generalizado, facilitando la implementación de sistemas híbridos y concatenaciones diversas de forma particular. Al mismo tiempo, la incorporación de los algoritmos de entrenamiento en hardware permiten que el sistema pueda adaptarse en tiempo de ejecución. Para este caso se trató el algoritmo LMS en VHDL, dejando la posibilidad de definir otros modelos para hardware que optimicen la velocidad de convergencia, como es el caso del algoritmo NLFSR (Sandoval-Ruiz, 2017b).

Otro aspecto de interés corresponde a los logros que se derivan de este método de diseño enfocado a NNDR, como la optimización en el desempeño del sistema, donde se puede manejar una señal de calidad, de forma eficiente con respecto a la potencia de transmisión, lo que se traduce en avances en materia de ahorro energético y disminución de la potencia disipada. Esto permite crear un espacio de investigación y desarrollo de nuevos conocimientos en la materia.

Se tiene, finalmente, que la investigación desarrollada es un aporte para aplicaciones sobre tecnología FPGA para la implementación en hardware reconfigurable, ya que facilita la adaptabilidad de los productos finales. Este aspecto es importante en los procesos continuos de actualización, además de cumplir con la sintaxis en VHDL, lo que a su vez ofrece portabilidad de la generación de secuencias a diversas plataformas.

REFERENCIAS

- BEALE, M. H., HAGAN, M. T., & DEMUTH, H. B. (2011). *Neural Network Toolbox*.
- BLOGH, J. S., & HANZO, L. S. (2002). *Third-Generation Systems and Intelligent Wireless Networking, smart antennas and adaptive modulation*. England: John Wiley & Sons.

- BONELLO, N., CHEN, S., & HANZO, L. (2009). On the Design of Pilot Symbol Assisted Codes. In *Vehicular Technology Conference Fall (VTC 2009-Fall)*, 2009 IEEE 70th (pp. 1–5). IEEE. Retrieved from http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5378698
- CASTELLANOS, J., SANDOVAL, C., & AZPURUA, M. (2014). A FPGA implementation of a LMS adaptative algorithm for smart antenna arrays. *Revista Técnica de La Facultad de Ingeniería de La Universidad de Zulia, Venezuela*, 37(3), 270–278.
- DIAZ, M., FERNANDES, V., & SANDOVAL-RUIZ, C. (2012). Sistema Eficiente de Modulación Digital basado en Radio Cognitivo. In *"I Congreso Venezolano de Ciencia, Tecnología e Innovación"* (p. 285). Retrieved from [http://www.oncti.gob.ve/images/pdf/Tomo1-resumenes\(interactivo\).pdf](http://www.oncti.gob.ve/images/pdf/Tomo1-resumenes(interactivo).pdf)
- FETTE, B. A. (2009). *Cognitive radio technology. Engineering* (Second Edi). Academic Press.
- HAGAN, M. T., DEMUTH, H. B., & BEALE, M. H. (1995). Neural Network Design. *Boston Massachusetts PWS*, 2, 734.
- HERNÁNDEZ, C., MARTÍNEZ, P., FERNANDO, L., SARMIENTO, M., & HERNÁN, F. (2016). Algoritmos para asignación de espectro en redes de radio cognitiva. *Tecnura*, 20(48), 69–88.
- MORA, A. (2008). *Estudio de Arquitecturas VLSI de la Etapa de Predicción de la Compensación de Movimiento, para Compresión de Imágenes y Video con Algoritmos. Aplicación al Estándar H.264/AVC Tesis doctoral*. Universidad Politécnica de Valencia.
- NOORDIN, N. K., ALI, B. M., ISMAIL, N., & JAMUAR, S. S. (2004). Adaptive Techniques in Orthogonal Frequency Division Multiplexing in Mobile Radio Environment. *International Journal of Engineering*, 1(2), 115–123.
- PODOSINOV, V. S. (2011). *A Hybrid DSP and FPGA System for Software Defined Radio Applications*. Polytechnic Institute and State University.
- REIS, A. L. G., BARROS, A. F., LENZI, K. G., MELONI, L. G. P., & BARBIN, S. E. (2012). Introduction to the software-defined radio approach. *IEEE Latin America Transactions*, 10(1), 1156–1161.
- 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. (2016). Modelo de Estructuras Reconfigurables con Registro Desplazamiento, para Lenguaje Descriptor de Hardware VHDL. *Revista Fac Ing UCV*, 31(3), 109–118.
- SANDOVAL-RUIZ, C. (2017). Modelo Neuro-Adaptativo en VHDL, basado en circuitos NLFSR, para control de un Sistema Inteligente de Tecnología Sostenible. *Universidad, Ciencia Y Tecnología*, 21(85), 140–149.
- SANDOVAL-RUIZ, C. E. (2017a). Análisis de Circuitos Fractales y Modelado a través de Sistema de Funciones Iteradas para VHDL. *Revista Ciencia E Ingeniería*, 38(1), 3–16.
- SANDOVAL-RUIZ, C. E. (2017b). Logical-Mathematical Model of Encoder 2D-RS for Hardware Description in VHDL. *Revista Ingeniería UC*, 24(1), 28–39.
- SANDOVAL-RUIZ, C. E. (2017c). Modelo en VHDL de Redes Neuronales Configurables Aplicadas a Decodificación en Radio Cognitivo. *Revista Ingeniería UC*, 24(3).
- SANDOVAL-RUIZ, C., & FEDÓN-ROVIRA, A. (2014). Efficient RS (255 ,k) encoder over reconfigurable systems. *Rev. Téc.Ing.Zulia*, 37(2), 151–159.
- SANDOVAL, C. (2007). *Diseño Modular de un Sistema para Procesamiento y Comunicación Digital en Banda Base usando Programación en VHDL*. Universidad de Carabobo.
- SANDOVAL, C., & FEDÓN, A. (2007a). Codificador de fuente programado en VHDL para dispositivos de hardware reconfigurable. *Revista Ciencia E Ingeniería*, 28(1), 37–40.
- SANDOVAL, C., & FEDÓN, A. (2007b). Codificador y decodificador digital Reed-Solomon programados para hardware reconfigurable. *Ingeniería Y Universidad*, 11(1), 17–32.
- SANDOVAL, C., & FEDÓN, A. (2008). Descripción modular de un esquema de codificación concatenado para corrección de errores con programación de hardware. *Ingeniare*, 16(2), 310–317.

- SANDOVAL RUIZ, C. E. (2008). Diseño de Módulos Funcionales para Implementación de Códigos Concatenados Paralelos sobre FPGA. In *VI Congreso de Investigación Universidad de Carabobo* (pp. 535–539). Venezuela: Universidad de Carabobo.
- STALLING, W. (1997). *Data and Computer Communication* (5° Edición). Prentice Hall.
- WANG, X. A., & WICKER, S. B. (1996). An artificial neural net Viterbi decoder. *IEEE Transactions on Communications*, 44(2), 165–171.

