

A FPGA implementation of a LMS adaptative algorithm for smart antenna arrays

**Javier Alexander Castellanos Hernández¹,
Cecilia Esperanza Sandoval Ruiz², Marco Aurelio Azpúrua Auyanet³**

¹Coordinación de Electrónica, UNEFA. Carretera Maracay-Mariara, frente a la base Aérea "Mariscal Sucre", Edo. Aragua, Venezuela. Tlf: 058+0412-0783140. jachplay@hotmail.com

²Facultad de Ingeniería, Universidad de Carabobo. Naguanagua, sector Bárbula. Edo. Carabobo, Venezuela. Tlf: 058+0414-9447293 cecisandova@yahoo.com

³Laboratorio de Electromagnetismo Aplicado, Fundación Instituto de Ingeniería para Investigación y Desarrollo Tecnológico. Carretera Baruta-Hoyo de la Puerta, Urb. Monte Elena II (Entrada IDEA), Altos de Sartenejas, Baruta, Edo. Miranda, Venezuela. Tlf: 058+0412-8714029. bazpurua@fii.org

Abstract

This paper presents a design of an adaptive algorithm, proposed for applications in arrays of antennas with adjustable weight. This algorithm was selected because of its simplicity, efficiency and performance; which makes it idoneous to be implemented on a reconfigurable hardware device. The methodology developed begins with the design of the study case and the analysis of the LMS algorithm in the adaptive control of the radiation pattern for the four element antenna array, using MATLAB™ v.7.6., then the definition of the model, the components, the description of the behavior using the syntax in VHDL for the design on a FPGA device, using the ISE 10.1 Xilinx tool. Finally, the validation of the designed circuit is presented through simulations using ModelSim 5.7. Resulting in synthesis reports resources used in the design developed.

Keywords: adaptive algorithm LMS, Very High Speed Hardware Description Language (VHDL), Field Programmable Gate Arrays (FPGA).

Implementación sobre FPGA de un Algoritmo LMS para un arreglo de antenas inteligentes

Resumen

Este artículo presenta el diseño de un algoritmo adaptativo, propuesto para aplicaciones en arreglos de antenas con pesos ajustables. Este algoritmo fue seleccionado por su simplicidad, eficiencia y desempeño, haciendo idónea su implementación sobre dispositivos de hardware reconfigurable. La metodología desarrollada inicia con el diseño del caso de estudio y la el análisis del algoritmo LMS en el control adaptativo del patrón de radiación del arreglo de antenas de cuatro elementos, usando MATLAB™ v.7.6., seguido de la definición del modelo, los componentes, la descripción del comportamiento usando sintaxis VHDL para su diseño sobre un FPGA, usando la herramienta de Xilinx ISE 11.1. Finalmente, la validación del circuito diseñado se presentó a través de la simulación con ModelSim 5.7. Obteniendo como resultado los reportes de síntesis de recursos empleados en el diseño desarrollado.

Palabras clave: algoritmo adaptativo LMS, Lenguaje descriptor de hardware para dispositivos de muy alta escala de integración (VHDL), arreglos de compuertas programables por campo (FPGA).

Introducción

Las antenas inteligentes son un conjunto de elementos radiadores organizados en forma de arreglos [1], en donde las señales de cada uno de estos elementos, son procesadas y controladas para formar un patrón radiación ajustable. Esta técnica se denomina conformación de haces adaptativos, la cual focaliza las señales transmitidas o recibidas por un arreglo de antenas hacia una dirección específica, modificando la dirección del lóbulo principal del patrón de radiación, a partir de la dirección deseada; permitiendo minimizar la interferencia co-canal ocasionada por los haces de las señales provenientes de otras direcciones [2]. La conformación de haces adaptativos consiste en asignar una ponderación dinámica, para cada señal recibida de los elementos del arreglo de antenas, denominados pesos, los cuales se calculan iterativamente usando algoritmos adaptativos complejos.

El mayor desafío para la implantación de sistemas de antenas inteligentes en las comunicaciones inalámbricas, es la gran velocidad de adaptación que deben poseer dichos sistemas, para responder a un entorno altamente cambiante. Esto requiere que el cálculo de los pesos que controlan el patrón de radiación, se ejecuten muy rápidamente, y por lo tanto se requiere una alta velocidad de procesamiento y elevada capacidad de cómputo. Por ende, la implementación a nivel de hardware de dichos algoritmos es una alternativa que permite que estos sistemas sean una realidad en las comunicaciones inalámbricas. En este sentido, una de las tecnologías más versátiles a la hora de procesar señales a nivel de hardware son las FPGA, ya que permiten sintetizar circuitos específicos, a partir de código escrito en el lenguaje descriptor de hardware VHDL.

Actualmente, los algoritmos de filtrado adaptativo se utilizan para diversas aplicaciones que requieren ajustes de parámetros durante procesos dinámicos, para optimizar la respuesta de los sistemas. En consecuencia, es posible mejorar las prestaciones de los sistemas de telecomunicaciones inalámbricas en aspectos como la relación señal a ruido y el consumo de potencia, aplicando técnicas adaptativas sobre sus sistemas de antenas. En tal sentido, resulta de interés una comparación entre los diferentes tipos de filtros adaptativos considerados en la implementación de un

algoritmo para el ajuste de los pesos, aplicado a un arreglo de antenas, considerando como factores de selección: la tasa de convergencia, robustez, seguimiento y requerimiento computacional.

Este trabajo presenta el proceso de diseño de un procesador digital con algoritmo de adaptación aplicado a un arreglo de antenas, describiendo su comportamiento en VHDL, para sintetizarlo sobre dispositivos de tecnología de hardware reconfigurable, FPGA. Para el cálculo de los pesos ponderados se ha seleccionado el algoritmo adaptativo LMS (del acrónimo para el término inglés *Least-Mean-Squares*), debido a que aun cuando posee una menor tasa de convergencia que el RLS (del acrónimo para el término inglés *Recursive-Least-Squares*), presenta mayor estabilidad, robustez, y menor complejidad computacional [3], lo que lo hace apropiado, en aplicaciones de tiempo real. Una vez seleccionado el algoritmo, se estudiaron los arreglos, en el cual se ha seleccionado el arreglo *scanning* de cuatro elementos, como caso de estudio, se realizaron pruebas de convergencia del algoritmo sobre la aplicación seleccionada [4].

Aplicación del algoritmo LMS

A partir del estudio de la arquitectura de los filtros digitales adaptativos [5], se pueden definir las ecuaciones que describen el comportamiento del sistema, para su aplicación sobre un arreglo de antenas. En la Figura 1 se presenta la estructura de un filtro adaptativo, con un algoritmo de control LMS, para la interpretación de la relación entre las señales.

De la estructura del filtro, se puede deducir la ecuación que define el comportamiento de la salida, como se presenta en la ecuación 1.

$$y(t) = w^H * x(t), \quad (1)$$

donde w^H es la transpuesta conjugada compleja del vector de ponderación de pesos w y $x(t)$ es la señal de entrada a la etapa de control adaptativo. El error cuadrático medio entre la salida del filtro digital (conformador de haz) y la señal de referencia puede ser expresado a través de la ecuación 2.

$$\varepsilon^2(t) = [d(t) - w^H * x(t)]^2, \quad (2)$$

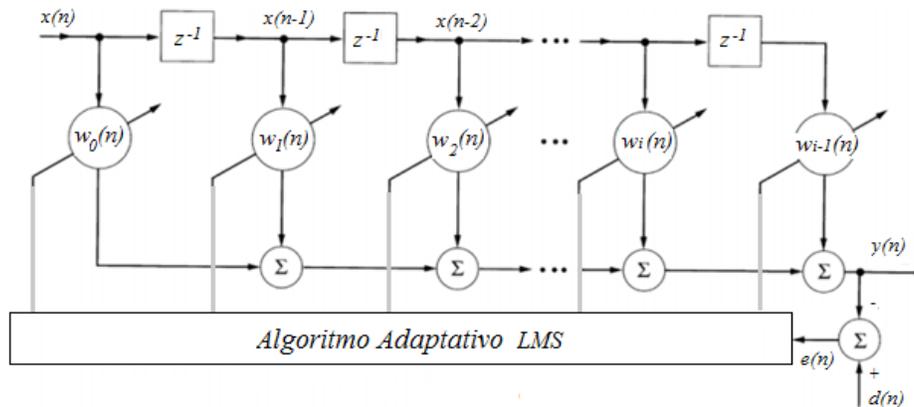


Figura 1. Estructura del Filtro Adaptativo y Algoritmo LMS.

siendo ε el error resultante, de la diferencia entre la señal de salida $y(t)$, (calculada mediante la ecuación 1) y la señal deseada $d(t)$. Para obtener el mínimo error se requiere implementar un algoritmo a fin de obtener la ponderación de los pesos del sistema. El algoritmo LMS [6], es altamente usado por su simplicidad, eficiencia computacional y buen rendimiento bajo condiciones de operación variable. A partir del método de gradiente descendente se desarrollan las ecuaciones de soporte del mencionado algoritmo, tratadas en el dominio discreto para el procesamiento digital de las señales muestreadas, como se observa en la ecuación 3.

$$w(n+1) = w(n) + \frac{1}{2} \mu \left[-\nabla \left(E(\varepsilon^2(n)) \right) \right], \quad (3)$$

donde $w(n+1)$ es el peso calculado, $w(n)$ corresponde al peso actual, μ son los tamaños del paso de adaptación, que caracterizan la convergencia del algoritmo LMS (comprendido en un rango entre 0 y 1) y $\varepsilon^2(n)$ el error cuadrático. Luego haciendo uso de la versión en el dominio discreto de la ecuación 2, tenemos $\varepsilon^2(n)$, el error cuadrático entre la salida $y(n)$ y la señal de referencia $d(n)$, a partir del cual se calcula el vector gradiente con respecto a la ponderación de los pesos, y sustituyendo en (3) se obtiene la actualización de los mismos con la ecuación 4.

$$w(n+1) = w(n) + \mu \cdot x(n) \cdot \varepsilon(n), \text{ con} \\ \varepsilon(n) = d(n) - x^H(n)w(n) \quad (4)$$

Esta ecuación describe el comportamiento del algoritmo LMS, utilizando la solución del

mínimo error cuadrático alcanzable (en sentido estadístico), bajo la asunción de un filtrado lineal en un entorno estacionario [7], donde los coeficientes o pesos w , corresponden a la respuesta impulsional del filtro, calculado en función de n muestras, debido a que está sujeto a un proceso de aprendizaje y por ello debe ser actualizado en el tiempo, a partir de la señal de error estimado con respecto a la señal deseada $d(n)$ y el vector de datos $x(n)$ de la señal de entrada. La cantidad μ denominado paso de adaptación, determina la velocidad de aprendizaje que se desea imprimir al sistema. Los algoritmos de gradiente emplean el error cuadrático mínimo, para proporcionar la actualización de los coeficientes.

Diseño del algoritmo LMS para un arreglo de antenas

En esta sección se presenta la evaluación de la convergencia y la capacidad de seguimiento, del algoritmo LMS, ya que al momento de diseñar, es importante la selección adecuada del parámetro μ , el cual controla la velocidad de convergencia, se debe acotar que existe una velocidad máxima de convergencia o valor máximo del parámetro, para que el algoritmo no diverja. Si μ es muy pequeño el algoritmo converge muy lentamente, en el caso contrario converge rápidamente pero puede volverse inestable [6]. La evaluación de los parámetros se ha realizado a través de modelos desarrollados en MATLAB™, para determinar la respuesta del algoritmo, basado en un estudio de los conceptos de arreglo de antenas [8] y su comportamiento [9], partiendo de las etapas del conformador de haces.

La primera etapa del conformador de haces, corresponde al **arreglo de antenas**, con las señales: $s_N(t)$, representan las señales incidentes en los elementos de antenas y $n(t)$ representa el ruido aditivo, de donde se obtiene la señal $x(t)$, la cual corresponde a la entrada en banda base para el control adaptativo, como se muestra en la ecuación 5.

$$x(t) = s(t) * a(\theta_0) + \sum_{i=1}^{N_u} u_i(t) * a(\theta_i) + n(t), \quad (5)$$

donde $s(t)$ es la señal incidente al arreglo con un ángulo θ_0 ; $u_i(t)$ denota a las N_u señales interferentes que llegan al arreglo con un ángulo θ_i ; $a(\theta_i)$ es el vector de propagación del arreglo de las i -ésimas señales de interferencia y $a(\theta_0)$ es el vector de propagación del arreglo de la señal deseada.

La segunda etapa está constituida por el **procesador digital**, encargado generar la señal de salida a partir de las señales que intervienen en el arreglo de antenas y calcular la actualización de los pesos ponderados óptimos (w_n), a través de un algoritmo adaptativo, procesado por un DPS o FPGA, a fin de obtener el patrón de radiación deseado [1] (con una máxima radiación hacia una dirección definida y nulos en la dirección de señales interferentes).

En la Figura 2 se presenta (a) la arquitectura del conformador de haces adaptativos, con un algoritmo LMS, y (b) la respuesta del diseño, contrastando la señal estimada $y(n)$, con la señal deseada $d(n)$.

En este caso, se ha seleccionado un arreglo de antenas lineal de cuatro elementos, con una distancia entre cada elemento de un cuarto de longitud de onda $0,25\lambda$. Esto con el propósito de evaluar el diseño del algoritmo LMS con un paso de adaptación μ de 0,002, obteniendo como respuesta el seguimiento de la señal deseada. Para una señal deseada $d=4$, la señal estimada se estabiliza aproximadamente en $y=3,995$, el algoritmo converge en 102 muestras, con $w_1=0,33317$, $w_2=0,33316$, $w_3=0,33314$ y $w_4=0,3331$. Luego, al cambiar la señal deseada $d=2$ (a partir $n=200$ muestras), el algoritmo tiene la capacidad de hacer seguimiento a la señal deseada aproximadamente en $n=300$ muestras, obteniendo como resultado, la señal estimada $y=2,0011$ y los pesos actualizados $w_1=0,16683$, $w_2=0,16683$, $w_3=0,16682$ y $w_4=0,16681$, comprobando así que el algoritmo tiene un buen comportamiento en ambientes dinámicos.

Por otra parte, se realizó la evaluación del patrón de radiación, aplicando dos valores de μ , en este arreglo inciden 3 señales recibidas, dos de interferencias y una señal deseada. La señal deseada está a un ángulo de 120° y las señales de interferencias están con un ángulo de 30° y 90° respectivamente. Aplicando el algoritmo LMS con un paso de adaptación de $\mu=0,02$, se obtuvo como resultado el desajuste de la señal estimada (Figura 3.a), por ende la divergencia del algoritmo, lo que quiere decir que los valores de los pesos se encuentran alejados de los óptimos, trayendo

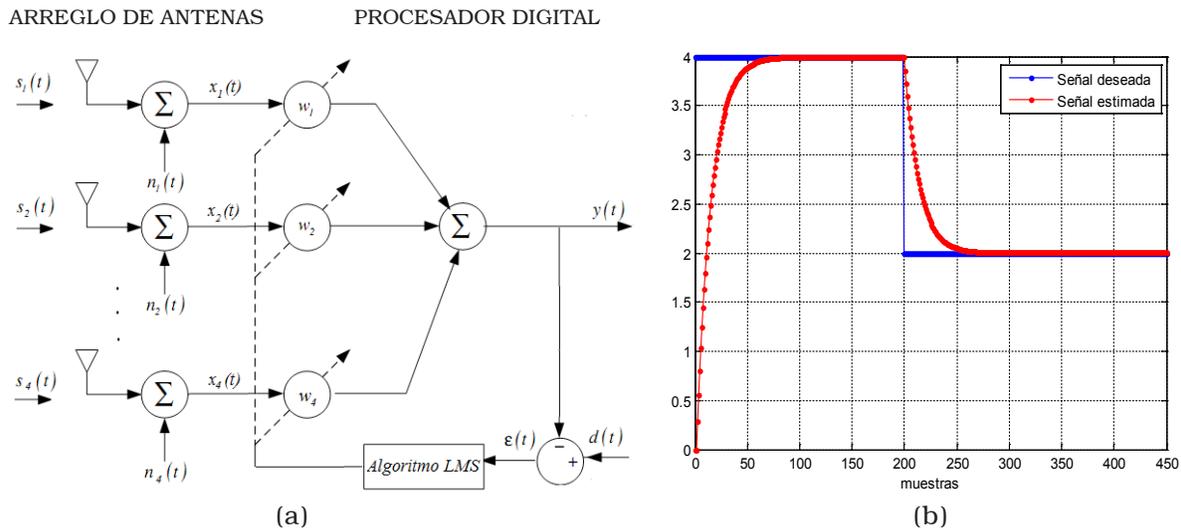


Figura 2. (a) Arquitectura del conformador de haces adaptativos y (b) respuesta del algoritmo LMS.

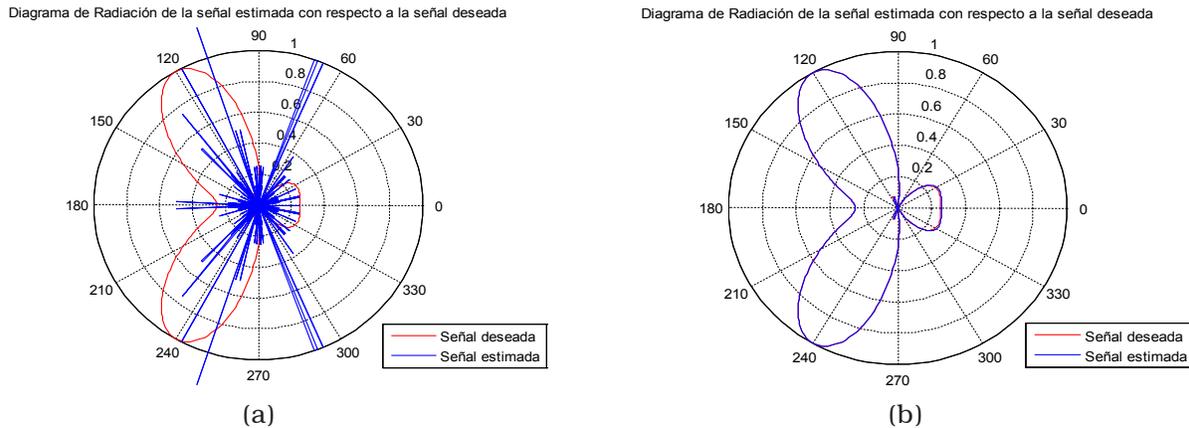


Figura 3. Patrón de radiación de la señal estimada con respecto a la señal deseada (a) con $\mu=0,02$ (b) con $\mu=0,002$.

como conclusión que no es apropiado el factor de convergencia. Luego, se modificó el factor de convergencia a $\mu=0,002$, en la cual se pudo observar (Figura 3.b) como la señal estimada se aproxima a la señal deseada (esto lo logró en $n=300$ muestras), eliminando así las interferencias recibidas.

Descripción en VHDL del algoritmo LMS

La implementación del algoritmo seleccionado sobre el FPGA, se realizó partiendo de la estructura estudiada, definiendo la secuencia de operaciones en base a un modelo generalizado [10], considerando el manejo de las actualizaciones de los pesos bajo procesos dependientes de la señal de reloj y un procesamiento de las muestras basado en componentes descritos en lenguaje VHDL [11], con el propósito de procesar los términos de la ecuación 4, de una manera eficiente a nivel de hardware.

Inicialmente, se definió una entidad (*entity lms*) la cual es la representación de la unidad del dispositivo *controlador adaptativo*, en la que se realizó la definición de los puertos, tanto de entradas (*in*), salidas (*out*) como entradas-salidas (*inout*), representando los parámetros del algoritmo. Se trabajó con vectores lógicos, definidos por la librería IEEE (*IEEE.std_logic_1164*). Las señales de entrada fueron definidas como vectores *xn1*, *xn2*, *xn3* y *xn4* respectivamente con una dimensión de 8 bits los cuales representan una parte entera y una parte decimal, es decir, los 5 bits más significativos como la parte entera y los 3

bits restantes como la parte decimal (*xxxxx.xxx*), se definió un vector de entrada, el cual representa la señal de referencia (*dn*). En vista de requerir realimentación para uno de los componentes, se definieron algunos vectores como entrada-salida, entre ellos: el vector de la señal de error (*error*) con las mismas dimensiones al vector de la señal de referencia (*8 bits: xxxxx.xxx*).

Por otra parte, se definieron los vectores de pesos actuales (*wn*, *wn2*, *wn3* y *wn4*) con dimensiones de 6 bits, los 3 bits más significativos como la parte entera y los 3 bits menos significativos como la parte decimal (*xxx.xxx*). Finalmente, 4 vectores con las mismas dimensiones de los antes mencionados, que representan los vectores en t^+ , una muestra siguiente (*wnt*, *wnt2*, *wnt3* y *wnt4*). Por último, se realizó la definición de unos vectores de 6 bits (*factor*, *factor2*, *factor3* y *factor4*) que representan el producto entre los vectores de la señal de error con la señal de entrada de cada elemento.

La señal de salida al filtro (*yn*) está representada por un vector de salida de 8 dimensiones con las mismas características al vector de la señal de referencia (*8 bits: xxxxx.xxx*). Igualmente, se tomó como criterio de diseño el factor de convergencia μ por ser un valor considerablemente pequeño, se definió un vector con una dimensión de 11 bits, el cual es interpretado como los dos bit más significativos como la parte entera y los 9 bits restantes se interpretan como la parte decimal, obteniendo una resolución binaria de 512 combinaciones para la parte decimal y 4 para la parte entera. Se definió una entrada de habilitación (*hab*), con el

fin de poder realizar la respectiva inicialización del proceso y una entrada de sincronización; la señal de reloj (*clk*), que permitió manejar el circuito secuencial.

Debido a que lenguaje VHDL, cuenta con la característica de modularidad, se modelo el algoritmo LMS para programarlo, a través de componentes que jerarquizan la arquitectura del diseño. Se realizó la definición de los componentes: *delay* y *signo*, permitiendo así ejecutar en paralelo a través de un proceso (*process*), varios circuitos secuenciales, dejando en la arquitectura principal, la estructura combinacional.

Componente *delay*, el cual actualiza su salida cuando ocurre un cambio en la señal de reloj (*clk*), realizando la asignación y actualización de las señales y vectores que intervienen en el diseño. Esta actualización es realizada cuando el bit de habilitación se encuentra en bajo (*hab=0*), haya ocurrido un cambio en el reloj y el mismo se encuentre en alto (*clk=1*). Se realiza la inicialización de los vectores en caso que el bit de habilitación se encuentre en alto (*hab=1*). También, en dicho componente se realiza la operación del cálculo del error entre el vector de la señal deseada (*dn*) y la señal *temp*, la cual representa el valor de la señal de salida del filtro en una determinada muestra. Como se puede observar en la Tabla 1, se presenta el *process* de dicho componente, donde la inicialización de las señales, actualización y los cálculos realizados se hacen dentro de una estructura *IF-THEN-ELSE*, el cual sólo es aplicado en un *process*, debido a la presencia de instrucciones secuenciales.

Los vectores que son actualizados en dicho componente son los vectores de pesos (*w*), los cuales debido a la ecuación 6 requieren el valor de la muestra actual y de la muestra anterior, así como también se realiza la actualización del valor

del vector de la señal de salida del filtro (*yn*), el vector del error (*error*).

Componente *signo*, el cual se ejecuta a través de un *process* que depende de una lista de señales sensibles como son: *dn* la señal *patrón* y el bit de habilitación. En el mismo componente, se realizan los cálculos respectivos a la suma aritmética del valor del peso actual de cada elemento con el vector *factor* correspondiente. En la Tabla 2, se muestran las asignaciones y cálculos realizados dentro del proceso que permitieron obtener un circuito secuencial.

El comportamiento del controlador adaptativo se describe en VHDL mediante el mapeo de las señales sobre los componentes definidos y sus realimentaciones respectivas, así como; los cálculos de los productos correspondientes a las ecuaciones del algoritmo, como por ejemplo ($\mu * error * x_n$) y ($w_n * x_n$), se realizó una multiplicación binaria considerando la interpretación que se está dando a cada vector, es decir, la partición del vector para una parte entera y la parte decimal, con truncamiento de los bits nulos que se generan en la multiplicación, manteniendo el aporte de los pesos w_n calculados para el procesado de la señal y_n .

Análisis de resultados

Se realizó la simulación usando Modelsim 5,7, para cada cambio de la señal deseada, se pudo observar que para el valor de $dn = 11001,100$ (25,1); el algoritmo realizó la convergencia aproximadamente en 7050,01ns (Figura 4.a), obteniendo como salida $yn = 11000,110$ (24,3) y valores de los pesos de $w_n = 010,111$, $w_n2 = 000,101$, $w_n3 = 000,101$ y $w_n4 = 010,110$.

Luego, al cambiar el valor de $dn = 10101001$ (21,4), se pudo observar como la señal estimada

Tabla 1
Estructura del Componente Delay

process (clk)			$w_n <= wnt;$
begin			$w_n2 <= wnt2;$
if hab = '1' then	error<="00000000";	if clk = '1' and clk'event then	$w_n3 <= wnt3;$
--proceso de inicialización	patron<="00000000";	if dn > temp then	$w_n4 <= wnt4;$
$w_n <= "001000";$	en<="00000000";	en <= dn - temp;	error <= en;
$w_n2 <= "001000";$	else	else	patron <= temp;
$w_n3 <= "001000";$	-- caso contrario	en <= temp - dn;	end if;
$w_n4 <= "001000";$	proceso de cálculo	end if;	end if;
			end process;

Tabla 2
Estructura del process del componente signo

```

process (hab,dn,patron)
begin
if hab = '1' then
temp3<="000000";
temp7<="000000";
temp8<="000000";
temp9<="000000";
wnt<="000000";
wnt2<="000000";
wnt3<="000000";
wnt4<="000000";
else
if dn > patron then
temp3<= factor1 + wn;
temp7<= factor2 + wn2;
temp8<= factor3 + wn3;
temp9<= factor4 + wn4;
wnt <= temp3(5 downto 0);
wnt2 <= temp7(5 downto 0);
wnt3 <= temp8(5 downto 0);
wnt4 <= temp9(5 downto 0);
else
if wn >= factor then
temp3<= wn - factor1;
temp7<= wn2 - factor2;
temp8<= wn3 - factor3;
temp9<= wn4 - factor4;
wnt <= temp3(5 downto 0);
wnt2 <= temp7(5 downto 0);
wnt3 <= temp8(5 downto 0);
wnt4 <= temp9(5 downto 0);
else
temp3 <= factor1 - wn;
temp7<= factor2 - wn2;
temp8<= factor3 - wn3;
temp9<= factor4 - wn4;
wnt <= temp3(5 downto 0);
wnt2 <= temp7(5 downto 0);
wnt3 <= temp8(5 downto 0);
wnt4 <= temp9(5 downto 0);
end if;
...
end process;
    
```

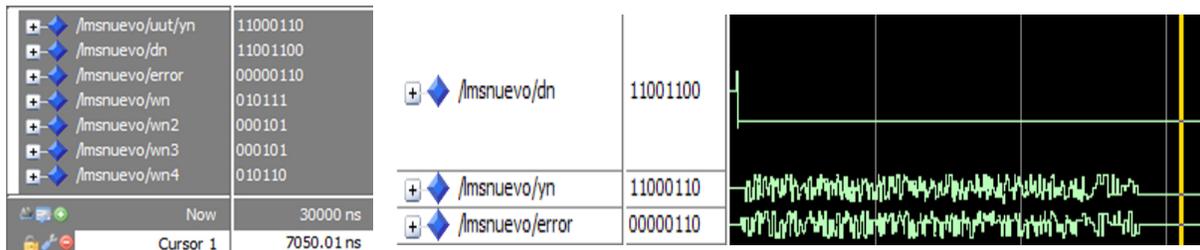


Figura 4.a. Simulación con $dn=25,1$ (a) en $t=7050,01$ ns (b) comportamiento en el tiempo.

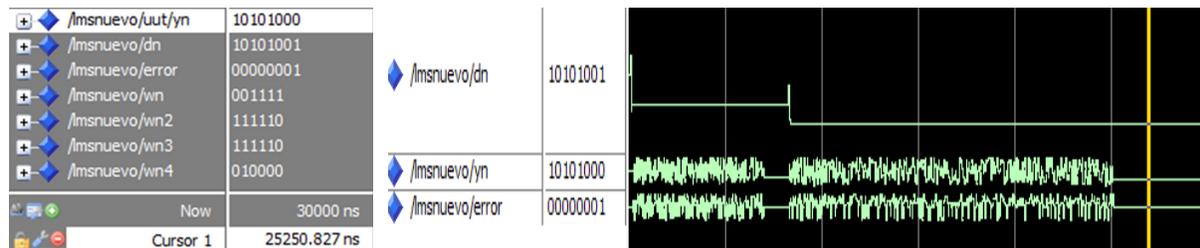


Figura 4.b. Simulación con $dn=21,4$ (a) en $t=25250,827$ ns (b) comportamiento en el tiempo.

sigue a la deseada hasta alcanzarla en 25250,83ns (Figura 4.b), realizando su respectiva convergencia y obteniendo como salida $yn=10101,000$ (21,0), para unos pesos óptimos de $wn=001,111$; $wn2=111,110$; $wn3=111,110$; y $wn4=010,000$.

Una vez validado el comportamiento del diseño, se procedió a obtener los diagramas esquemáticos de los componentes diseñados a través de la herramienta RTL Schematyc del ISE 11.1, donde se observa el diagrama de bloques de la implementación del algoritmo LMS sobre el dispositivo FPGA, tal como se presenta en la Figura 5.

En la Tabla 3 se presentan los porcentajes de utilización que ocupa el diseño sobre cada uno de los dispositivos seleccionados Spartan 3 (XC3S200 FT256) y Virtex 5 (XC5VLX50-3FF324).

El reporte de síntesis permitió comprobar la factibilidad de implementación del diseño sobre los dispositivos FPGA, donde se obtiene un bajo porcentaje en el consumo de recursos de hardware (slices, tablas de búsqueda LUTs, elementos de memoria FF, IOs, multiplicadores o DSP dedicados). Este tratamiento puede ser aplicado en diseños de sistemas MIMO con etapas adaptativas [12], técnicas de redes neuronales [13], y algoritmos modificados como el ECLMS [14], a fin de

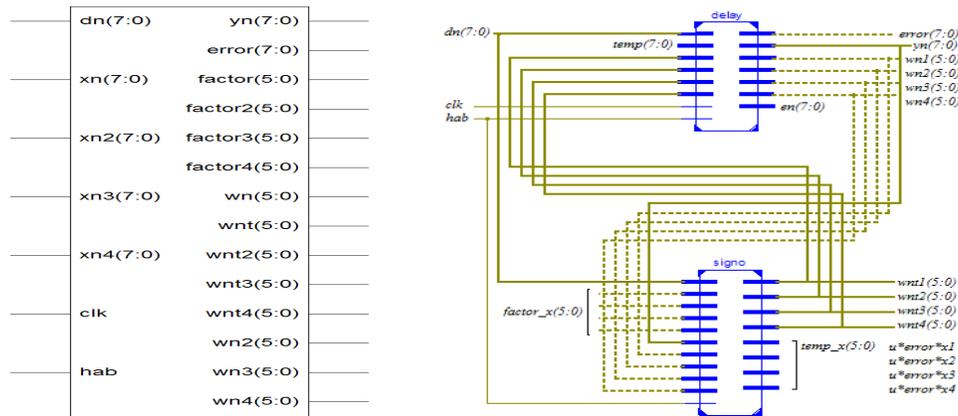


Figura 5. Diagrama de bloques de la Implementación sobre el FPGA.

Tabla 3
Reporte de Síntesis del LMS sobre dispositivos FPGA

XC3S200 FT256		XC5VLX50-3FF324	
Number of Slices:	80 out of 1920 4%	Number of Slices:	48 out of 28800 0%
Number of Flip Flops	48 out of 3840 1%	Number of LUTs-FF	43 out of 157 23%
Number of 4 input LUTs:	146 out of 3840 3%	Number of LUTs:	152 out of 28800 0%
Number of bonded IOBs:	130 out of 173 75%	Number of bonded IOBs:	130 out of 220 59%
Number of Mult 18x18	13 out of 12 108%	Number of DSP48Es	8 out of 48 16%
Number of GCLKs	1 out of 8 12%	Number of GCLKs	1 out of 32 3%

contrastar la eficiencia de sus implementaciones sobre hardware reconfigurable.

Conclusiones

En este trabajo se diseñó y configuró el algoritmo adaptativo LMS sobre un dispositivo FPGA con aplicación en arreglos de antenas inteligentes, obteniéndose resultados satisfactorios tanto en el comportamiento del algoritmo adaptativo LMS, como en el uso de los recursos del dispositivo reconfigurable. Uno de los aportes más significativos, consistió en el desarrollo y documentación de la metodología de diseño modular en VHDL, se realizó un tratamiento matemático particular, basado en truncamiento de bits no representativos para mantener longitud constante en palabras de longitud creciente debido a la naturaleza iterativa en el tiempo del algoritmo. Los módulos diseñados permiten implementar en un mismo FPGA m filtros adaptativos en paralelo para un arreglo de n antenas, haciendo uso de la característica de reusabilidad del código diseñado.

Se logró obtener excelentes tiempos de convergencia y estabilidad en el algoritmo, para la aplicación en la conformación de haz adaptativo en arreglos de antenas inteligentes. Se consiguió el control del patrón de radiación para el arreglo de antenas lineales de cuatro elementos, a través de la obtención del paso de adaptación μ óptimo para el estudio realizado, el cual fue probado tanto en la simulación del arreglo como en el diseño sobre hardware. De esta manera, los resultados alcanzados representan una alternativa para mejorar la calidad de las comunicaciones inalámbricas, reducir la potencia de transmisión requerida y mejorar la relación señal a ruido durante el proceso de recepción.

Agradecimiento

Agradecemos a DIOS por guiar el desarrollo de la investigación.

Los autores de este trabajo reconocen a la Fundación Instituto de Ingeniería para Investigación y Desarrollo tecnológico (FII) por el apoyo

brindado a este proyecto, adscrito al Laboratorio de Electromagnetismo Aplicado.

Referencias

1. P. Espinosa Díaz and C. Villarroel González, "Proposición y simulación de un algoritmo adaptativo para sistemas de antenas inteligentes," *Ingeniare*, Vol. 15, N° 3, (2007) 344-350.
2. O. Hirokazu, K. Minseok, and A. Hiroyuki, "FPGA Implementation of LMS and N-LMS Processor for Adaptive Array Applications," in *International Symposium on Intelligent Signal Processing and Communications*, 2006, 485-488.
3. B. Allen, *Adaptive Array Systems*. John Wiley & Sons, 2005.
4. J. Castellanos, *Sistema de Control sustentado en Algoritmos Adaptativos para un arreglo lineal de Antenas Inteligentes utilizando tecnología FPGA*, N° 0243(2009).
5. W. Zelaya, "Diseño de un filtro digital adaptativo como cancelador de ruido basado en el algoritmo LMS," *Universidad del Salvador*, 2004.
6. D. G. Manolakis, *Statistical and Adaptive Signal Processing*. McGraw-Hill, 2005.
7. L. Azpicueta, "Aplicación de algoritmos combinados de filtrado adaptativo. Tesis Doctoral," *Universidad Carlos III de Madrid*, 2011.
8. C. Balanis, *Antenna Theory, Analysis and Design*, Tercera. Sons, John Wiley &, 2005.
9. C. Balanis, *Introduction to Smart Antennas*, Primera Ed. Morgan & Claypool, 2007.
10. C. Sandoval-Ruiz, "Modelo concurrente de generadores de secuencia para descripción de hardware reconfigurable," in *VIII Congreso Nacional y 2do Congreso Internacional de la UC*, 2013.
11. C. Sandoval, "Diseño Modular de un Sistema para Procesamiento y Comunicación Digital en Banda Base usando Programación en VHDL," *Universidad de Carabobo*, 2007.
12. C. Sandoval, "Esquema de Codificación Adaptativa para Sistemas de Comunicaciones MIMO" in *4to Congreso Iberoamericano de Estudiantes de Ingeniería Eléctrica*, 2010, 25-30.
13. C. Sandoval, "FPGA prototyping of neuro-adaptive decoder," *Proceedings of the 9th WSEAS international*, pp. 99-104, 2010.
14. J. V. López, J. Gerardo, Á. Ochoa, and C. Sánchez, "Comparación e Implementación de los Algoritmos ECLMS y OAELMS en un DSP Comparison and Implementation on a DSP of ECLMS and OAELMS Algorithm," *Computación y Sistemas*, Vol. 14, N° 2, (2010) 175-185.

Recibido el 10 de Enero de 2013

En forma revisada el 14 de Julio de 2014