

METODOLOGÍA PARA LA IMPLEMENTACIÓN DEL MPM EN VHDL Y LA EMULACIÓN DE AMPLIFICADORES DE POTENCIA EN UNA TARJETA FPGA

Edgar Allende Chávez

Tecnológico Nacional de México/Instituto Tecnológico de Tijuana
edgar.allende@tectijuana.edu.mx

José Ricardo Cárdenas Valdez

Tecnológico Nacional de México/Instituto Tecnológico de Tijuana
jose.cardenas@tectijuana.edu.mx

José Alejandro Galaviz Aguilar

Instituto Politécnico Nacional, IPN-CITEDI
jgalaviz@citedi.mx

Andrés Calvillo Téllez

Instituto Politécnico Nacional, IPN-CITEDI
calvillo@citedi.mx

José Cruz Núñez Pérez

Instituto Politécnico Nacional, IPN-CITEDI
nunez@citedi.mx

Resumen

El presente trabajo muestra el diseño e implementación en VHDL del modelo polinomial con memoria que fue seleccionado para la emulación del comportamiento de amplificadores de potencia con el propósito de proporcionar una plataforma de pruebas y evaluación para el modelado matemático y su posterior uso en pre-distorsión digital. Las mediciones de un amplificador real modelo NXP de 10 W medido a 2 GHz se utilizaron para la obtención del modelo matemático el cual fue implementado en una tarjeta de evaluación y desarrollo

DSP-FPGA Altera Stratix III. Además el artículo describe el desarrollo de un conjunto de funciones, para la manipulación de números complejos, necesario para la implementación del modelo. Los resultados muestran un desempeño adecuado del modelo en VHDL el cual es capaz de emular las curvas de distorsión en amplitud y fase AM-AM y AM-PM. Finalmente a modo de validación la implementación se compara con una simulación en Matlab.

Palabras Claves: Amplificador de potencia, emulación, FPGA, modelo polinomial con memoria, VHDL.

Abstract

This paper shows the design and implementation in VHDL of the memory polynomial model which was selected for emulating the behavior of power amplifiers with the purpose of providing a test and evaluation test bed for mathematical modeling and its later use in digital predistortion. Measurements of a real power amplifier model NXP 10W at 2 GHz were used for obtaining the mathematical model which was implemented in the DSP-FPGA development kit, Stratix III Edition by Altera. This paper also describes the development of a function set for complex numbers manipulation which is needed for the implementation of the model. Results show a correct performance of the VHDL model which can emulate distortion curves for amplitude and phase AM-AM and AM-PM. Finally a comparison is done between VHDL model and Matlab simulation.

Keywords: Emulation, FPGA, Memory polynomial model, Power amplifier, VHDL.

1. Introducción

El amplificador de potencia para radiofrecuencia (RF-PA) es el circuito electrónico de entrada que permite incrementar el nivel de potencia de la señal RF que se desea enviar, antes de que la misma llegue a la antena. Lo anterior con el fin de que la señal sea transmitida y sobrepase la sensibilidad del receptor, lo que garantiza la demodulación de la información [Núñez, 2014]. Sin embargo el RF-PA es un elemento inherentemente no lineal sobre todo cuando se trabaja en la zona

de saturación y la amplificación lograda no corresponde a una ganancia lineal de la señal de entrada [Wood, 2016]. Aunado a esto, se han desarrollado nuevos esquemas de modulación que buscan hacer un uso más eficiente del ancho de banda disponible tales como acceso múltiple por división de código de banda ancha (WCDMA) y multiplexación por división de frecuencias ortogonales (OFDM) las señales resultantes de los anteriores esquemas tienen una envolvente no constante y un factor de potencia pico-promedio (PAPR) sumamente grande generalmente de 10 dB [Roblin, 2013]. Este tipo de multiplexaciones digitales lleva a trabajar el RF-PA en su región no lineal lo cual tiene efectos no deseados en la señal de salida tales como: productos de intermodulación (IMD), efectos de memoria, recrecimiento espectral así como interferencia en canales adyacentes; lo cual puede llevar a sanciones por parte de los organismos nacionales e internacionales reguladores de las telecomunicaciones [Wood, 2016], [Kiran, 2016], en el caso de México la comisión federal de telecomunicaciones (COFETEL).

Con el fin de corregir los efectos no deseados de la amplificación no lineal del RF-PA se han desarrollado varias técnicas de linealización entre ellas la linealización por anticipación, por retro alimentación y, una de las más estudiadas por su flexibilidad y exactitud, la pre-distorsión digital (DPD) [Wood, 2016], [Braithwaite, 2015]. La técnica de DPD consiste en que la señal de entrada sea acondicionada antes de ser aplicada al RF-PA, el tratamiento de la señal será entonces que pase por un elemento que tenga un comportamiento inverso del RF-PA [Hammi, 2014]. Para lograr obtener un bloque con el comportamiento inverso del RF-PA es necesario en primera instancia contar con un modelo matemático que describa adecuadamente el dispositivo [Liu, 2014], [Moon, 2011]. La literatura muestra una gran cantidad de modelos adoptados para modelado de RF-PAs entre ellos encontramos: modelos basados en series de Volterra, redes neuronales, sistemas neuro-difusos y recientemente modelos generados con programación genética [Fehri, 2014], [Mkadem, 2010], [Zhai, 2008], [Cárdenas, 2017].

Usualmente para la implementación del bloque de comportamiento inverso se elige usar tarjetas de desarrollo sobre todo basadas en FPGA, donde se

aprovechan las bondades de flexibilidad. La implementación de estos modelos en hardware generalmente se hace mediante el uso de tablas de búsqueda (LUT) como se muestra en [Gilabert, 2008] y [Cárdenas, 2015]. Sin embargo el uso de la metodología anterior no permite utilizar señal compleja la cual resulta necesaria al aplicar DPD como un solo modelo donde se incluya el comportamiento en fase y amplitud. Un factor a tomar en cuenta es la complejidad del modelo elegido ya que está impactará directamente en la cantidad de recursos necesarios ya sea por complejidad de computo o por necesidades de almacenamiento, este aspecto es sumamente importante dado que en el FPGA se tienen recursos limitados y la optimización resulta crucial [Renteria, 2016].

VHDL es el lenguaje de descripción de hardware de circuitos integrados de alta velocidad. Mediante este lenguaje se puede describir el comportamiento y la estructura de los sistemas electrónicos y es particularmente adecuado para describir la estructura de los diseños en hardware electrónico digital e implementarlos en plataformas tales como ASICs y FPGAs [Rushton, 2011].

El presente trabajo está organizado de la siguiente manera: se muestra la teoría del modelo polinomial con memoria (MPM) para la realización del modelado de un amplificador NXP 10W a 2GHz, se describe el proceso de codificación del MPM en VHDL así como el desarrollo de un conjunto de funciones que permitiera el uso de números complejos y la representación elegida para el uso de números fraccionales. En la sección 3 se muestra como resultado la simulación en Modelsim con una señal de entrada de amplitud modulada misma que se compara con gráficas de la simulación del modelo en Matlab. En la sección 4 se muestra la viabilidad de los módulos obtenidos para su uso en el modelado de RF-PAs así como DPD. Finalmente en la sección 5 se presentan las conclusiones de este trabajo de investigación.

2. Métodos

Dentro de las técnicas más difundidas para el modelado matemático de los RF-PAs se encuentran los modelos basados en series de Volterra debido a que consideran tanto la no linealidad del dispositivo como los efectos de memoria que

el mismo pudiera tener. Las series de Volterra describen la relación de una entrada con una salida en un sistema no lineal y para un caso discreto con datos del tipo complejos su representación matemática está dada por la ecuación 1.

$$y(n) = \sum_{k=1}^K \sum_{i_1}^Q \dots \sum_{i_{2k-1}}^Q h_{2k-1}(i_1, i_2, \dots, i_{2k-1}) \prod_{j=1}^{k+1} x(n - i_j) \prod_{k+2}^{2k+1} x^*(n - i_j) \quad (1)$$

Dónde:

$()^*$ - denota el complejo conjugado, $x(n)$ – es la entrada discreta de la muestra n , $y(n)$ – Es la salida discreta de la muestra n , K - es el orden de no linealidad del modelo, Q - es la profundidad de memoria del modelo y $h_{2k-1}(i_1, i_2, \dots, i_{2k-1})$ – Es el coeficiente de Volterra de orden k .

El modelo Polinomial con Memoria

A pesar de que las Series de Volterra modelan con exactitud el comportamiento del RF-PA, su uso no resulta práctico al momento de implementar el modelo en hardware debido a su gran complejidad computacional, la cual queda manifiesta cuando al incrementar el orden de no linealidad o la profundidad de memoria el número de coeficientes necesarios para el modelo crece de manera exponencial. Para evitar lo antes mencionado existen modelos derivados de ellas que permiten disminuir la complejidad del modelo sin sacrificar tanta exactitud, uno de estos modelos es el MPM. El MPM consiste en fases de retardo y solo considera los coeficientes de la diagonal principal de Volterra [Nuñez, 2014]. La ecuación 2 muestra el MPM utilizado.

$$y(n) = \sum_{q=0}^Q \sum_{k=1}^K a_{2k-1,q} |x(n - q)|^{2(k-1)} x(n - q) \quad (2)$$

Dónde:

$x(n)$ – es la entrada discreta de la muestra n , $y(n)$ – es la salida discreta de la muestra n , K - es el orden de no linealidad del modelo, Q - es la profundidad de memoria del modelo y $a_{2k-1,q}$ es el coeficiente correspondiente.

El MPM puede ser construido a partir de $Q + 1$ funciones con la estructura de la segunda sumatoria las cuales tendrán como entrada la señal retrasada q veces según sea su caso. En la figura 1, se muestra la estructura a bloques de una función con la estructura de la segunda sumatoria de la ecuación 2, mientras que en la figura 2 se muestra el diagrama a bloques del MPM completo.

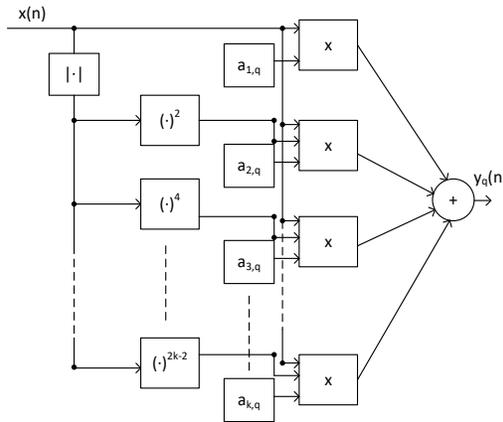


Figura 1 Diagrama a bloques de una función básica del MPM.

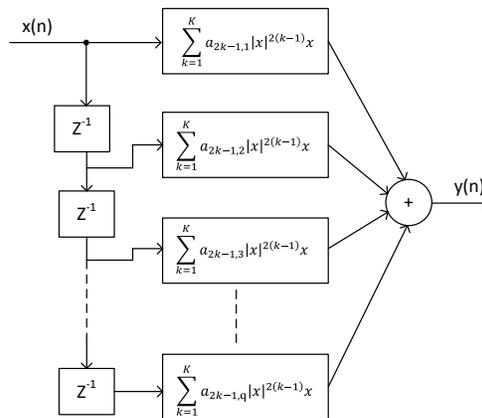


Figura 2 Diagrama a bloques del MPM completo.

Con el fin de extraer los coeficientes que mejor ajustan el modelo, se realiza una regresión lineal basada en mínimos cuadrados de la siguiente manera; la ecuación 2 se representa en forma matricial de modo que adquiere la forma de la ecuación 3 [Ku, 2003].

$$Y = H * a \tag{3}$$

Donde \mathbf{Y} es el vector de salidas, \mathbf{H} es la matriz de observación construida a partir de cada uno de los términos del MPM donde se tiene la unidad como coeficientes y \mathbf{a} es vector de coeficientes cuyo estimador puede ser calculado mediante la ecuación 4.

$$\hat{\mathbf{a}} = \mathbf{H}^+ * \mathbf{Y} \quad (4)$$

Donde \mathbf{H}^+ representa la pseudo-inversa de la matriz de observación.

Se utilizó un MPM con $K = 5$ y $Q = 2$ para el modelado de comportamiento de un amplificador NXP 10W medido a 2 GHz cuyas características eléctricas se muestran en la tabla 1.

Tabla 1 Características eléctricas del NXP 10W.

Parámetro	NXP 10W RF-PA @ 2 GHz
Ganancia (1dBm)	36 dBm
Clase	AB ($V_{ds}=50$ V, $I_{ds}=54$ mA)
Frecuencia de operación	500-2500 MHz
Eficiencia de drenado (η_d)	21%

Se cuenta con un total 65,536 muestras para este amplificador mismas que se usaron para la obtención de las curvas de distorsión, estas curvas caracterizan los efectos que tiene el RF-PA en la señal de entrada tanto en amplitud como en fase y permiten visualizar que también se ajusta el modelo a los valores reales medidos, las curvas de distorsión del NXP 10W fueron realizadas en Matlab y se muestran en los resultados. El cálculo de coeficientes para el MPM fue hecho en Matlab con el objetivo de insertar estos últimos en el modelo escrito en VHDL. Un parámetro numérico que permite cuantificar la calidad del modelo matemático es el error cuadrático medio normalizado (NMSE) el cual está dado por la ecuación 5.

$$NMSE = \frac{\sum_{n=0}^{N-1} (\hat{y}(n) - y(n))^2}{\sum_{n=0}^{N-1} (y(n))^2} \quad (5)$$

En donde: N es el número de muestras, $y(n)$ es el valor real de la salida en la muestra n y $\hat{y}(n)$ es el valor estimado por el modelo para la muestra n . El NMSE suele expresarse en decibeles para lo cual se utiliza la ecuación 6.

$$NMSE(dB) = 10 \log_{10}(NMSE) \quad (6)$$

Codificación del MPM en VHDL

Una de las necesidades para una implementación adecuada del MPM en VHDL es el uso de números fraccionales, por lo que se eligió la representación en punto fijo, esto debido a la ventaja en el tiempo de procesamiento con respecto a la representación en punto flotante.

La representación en punto fijo es una representación de un número fraccionario, el cuál se almacena en la memoria, en este caso el número se almacena como un entero con signo en el formato de dos complementos. Sobre lo anterior, se aplica una separación del vector localizando el punto base que separa la parte entera de la fraccional un número fijo de bits a la izquierda de su posición inicial, lo anterior se ilustra en el diagrama de la figura 3. Cuando se interpretan los bits del entero con signo almacenado en la memoria, se reposiciona el punto de base multiplicando el entero almacenado por un factor de escala fijo en este caso una potencia de dos ya sea positiva (parte entera) o negativa (parte fraccional).

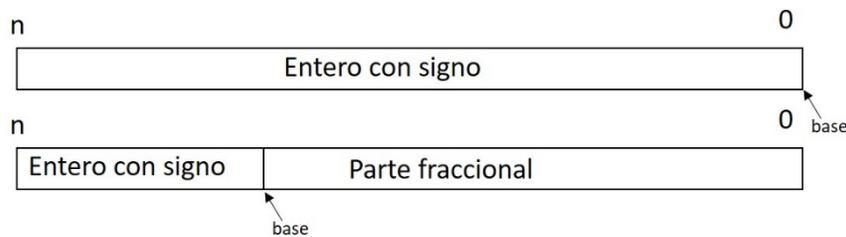


Figura 3 Representación de números fraccionales en punto fijo.

A partir de la versión del 2008, el estándar de VHDL se definen los tipos de datos de punto fijo y punto flotante, sin embargo el entorno de desarrollo QUARTUS II de Altera hace uso de la versión de 1993 por lo que fue necesario el uso de la librería de compatibilidad que se recomienda en [Rushton, 2011]. En dicha librería se tienen disponibles las diferentes funciones de operación para el trabajo con números de punto fijo con y sin signo, así como funciones de conversión y escalamiento.

Con el objetivo de que el módulo escrito pudiera procesar señal compleja, se escribió un conjunto de funciones en VHDL para el manejo de números complejos, en el que se define lo siguiente:

- Definición del tipo de dato complejo
- Suma de números complejos.
- Resta de números complejos.
- Multiplicación de números complejos.
- Absoluto de números complejos.
- Raíz cuadrada de números en representación de punto fijo.

Se planeó representar el tipo de dato complejo en su forma cartesiana, por lo que está constituido por un arreglo de dos vectores de 32 bits los cuales representan dos números fraccionales en punto fijo con signo, el número de bits para la parte entera y la fraccional es configurable con el fin de tener diversos niveles de exactitud y rangos de representación, para este trabajo se normalizó la señal compleja con el fin de tener valores de magnitud a la salida entre 0 y 1.

En el caso de la raíz cuadrada se escribió en VHDL el algoritmo mediante restas, descrito en [Paeth, 2014] y cuyo pseudocódigo aparece en la figura 4, las funciones de operación entre números complejos se escribieron utilizando las definiciones matemáticas para las formas binomiales de números complejos. Para todo lo anterior se eligió lo siguiente: en caso de un desbordamiento del número fraccional que hubiese un comportamiento de saturación y en el caso de que la resolución no fuera suficiente para la representación del número se emplea el redondeo al número más cercano que fuera posible representar. Todas las operaciones con números complejos codificadas tienen parámetros para el escalamiento del resultado.

Con el fin de mejorar el uso de recursos se modificó la parte del MPM en la que se realizan potencias de la magnitud de la entrada, la modificación hecha se muestra en la figura 5, ya que era deseable que la entrada se procesara en un solo ciclo de reloj y no era posible hacer multiplicaciones secuenciales se optó por un diseño donde se toma ventaja de las potencias pares y en lugar de usar 16

multiplicadores solo fueron necesarios 4 para cada una de estas etapas con una no linealidad de 5.

```
1 Definir tipo de dato "FXP" de 32 bits
2 con 2 bits para la parte entera
3 Sqrt(FXP x)
4 {FXP raiz, remHi, remLo, testDiv,
contador;
5 //inicializar parámetros
6 raiz = 0; remHi = 0; remLo = x; count =
30;
7 hacer {//obtener 2 bits del argumento
8 remHi = (remHi<<2) or (remLo>>30);remLo
<<= 2;
9 raiz <<= 1; //preparar el próximo bit de
la raiz
```

Figura 4 Pseudocódigo para extraer la raíz cuadrada de un número de punto fijo de 32 bits.

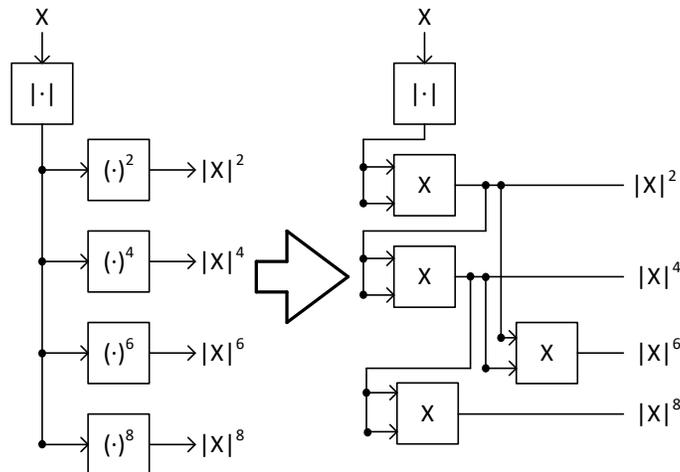


Figura 5 Optimización propuesta para el cálculo de las potencias.

Se desarrolló también el código para un bloque de retardo el cuál a cada ciclo de reloj entrega a la salida la entrada del ciclo anterior y hace una lectura de la nueva para tenerla disponible en memoria para la repetición del ciclo con un valor de 0 para la primera salida. Finalmente, se realizó el diseño de la entidad principal cuyo diagrama a nivel de transferencia de registros (RTL) puede verse en la figura 6 y en la que son claras las similitudes con el diagrama general del MPM mostrado

anteriormente. En la figura 6, puede observarse que el tipo de dato complejo en la entrada del sistema pasa hacia las diferentes funciones no lineales después de ser retrasada la cantidad de veces necesarias para finalmente realizar una suma de las salidas particulares de cada función y con ello lograr obtener la salida del MPM completo.

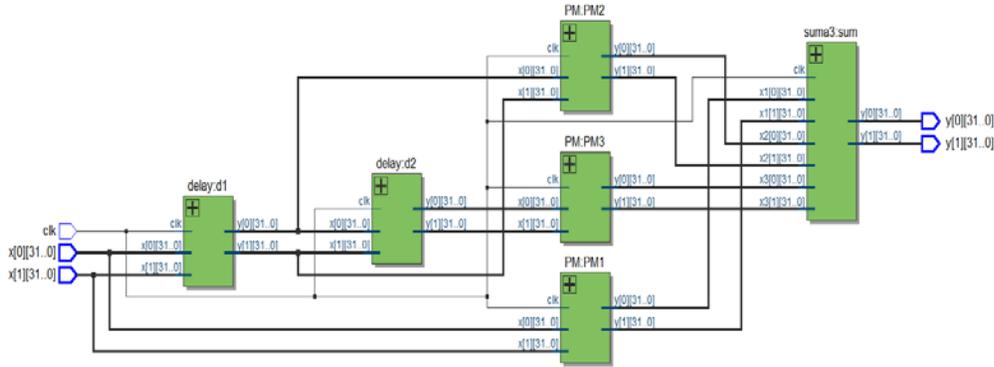


Figura 6 Diagrama RTL del MPM implementado mediante VHDL.

3. Resultados

La figura 7 muestra las curvas de distorsión AM-AM y AM-PM obtenidas por el MPM con $K = 5$ y $Q = 2$, mismo que fue implementado en VHDL. El NMSE obtenido por este modelo fue de -19.8256 dB y como puede observarse ajusta de manera correcta las mediciones del amplificador.

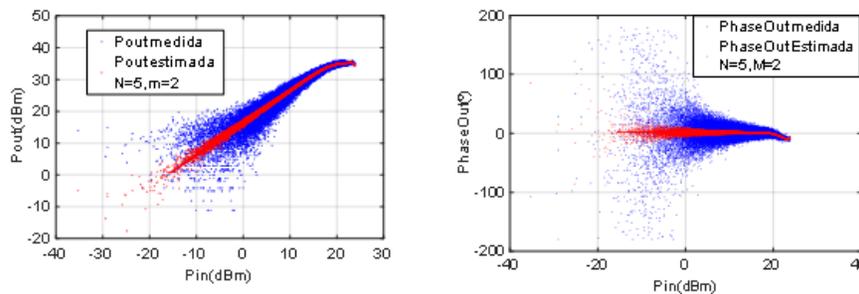


Figura 7 Curvas de distorsión AM-AM y AM-PM del amplificador NXP 10W.

Posterior al desarrollo de la implementación en VHDL del MPM se realizó un análisis de los recursos utilizados para la síntesis del mismo en una tarjeta de evaluación y desarrollo DSP-FPGA Stratix III de Altera, los resultados obtenidos se muestran en la tabla 2.

Tabla 2 Recursos utilizados por el MPM escrito en VHDL.

Recurso	Cantidad usada
ALUTs combinacionales	32,661 / 86,000 (38 %)
ALUTs de memoria	0 / 43,000 (0 %)
Registros lógicos dedicados	512 / 86,000 (< 1 %)
Total de registros	512
Total de pines	129 / 488 (26 %)
Total de pines virtuales	0
Total de bloques de memoria	0 / 4,303,872 (0 %)
Elementos DSP de 18 bits	214/ 288 (74 %)
Total de PLLs	0 / 4 (0 %)
Total de DLLs	0 / 4 (0 %)

Con el fin de conocer el desempeño del módulo escrito, se realizó la codificación de un banco de pruebas, en el que se suministra a la entrada una onda modulada en amplitud (AM) con portadora de 5 MHz y mensaje de 500 kHz muestreada a la frecuencia del reloj del FPGA, la cual es de 50 MHz. Para poder visualizar la salida se verificó la amplitud de la misma mediante el uso del valor absoluto del número complejo, en la figura 8 se muestra la simulación en Modelsim en donde se visualizan las magnitudes de la onda de la entrada y la onda a la salida del módulo que emula el comportamiento del RF-PA mediante el MPM.

Para validar los resultados obtenidos en el modelo MPM escrito en VHDL se realizó la codificación del mismo usando Matlab y se le aplicó la misma señal de amplitud modulada a la entrada, obteniendo los resultados observados en las figuras 9 y figura 10. Como puede observarse al comparar las gráficas tanto de la simulación en VHDL como de Matlab tienen exactamente la misma forma de onda. A nivel de valores en el binomio que representa el número complejo las variaciones con relación a la simulación fueron pequeños, puesto que en la representación en punto fijo se tenía una resolución mínima de 2^{-30} .

Finalmente, se realizó la programación de la tarjeta DSP-FPGA Stratix III para la visualización de los resultados, mismos que se muestran en las figuras 11 y 12, se acondicionó la señal resultante de modo que pudiera ser utilizada por el convertidor digital analógico de 14 bits de la tarjeta de adquisición de datos Terasic HSMC AD/DA.

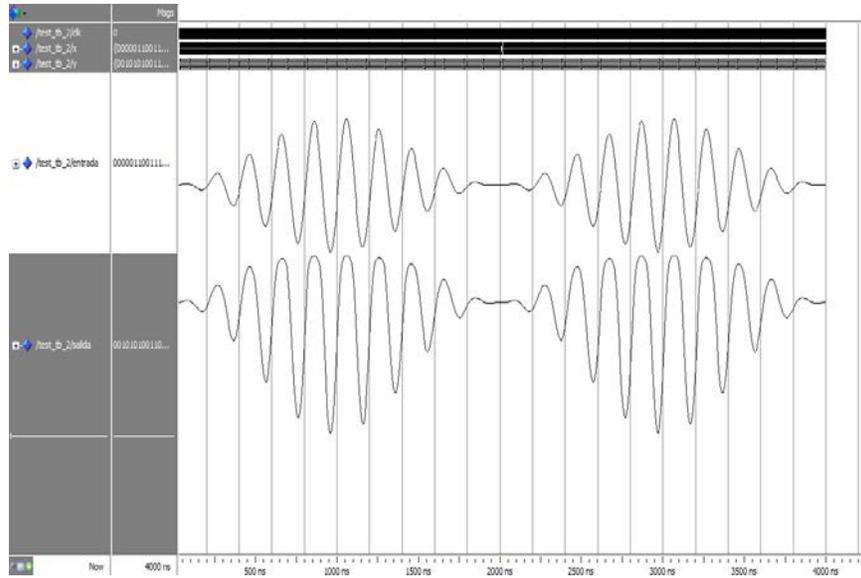


Figura 8 Simulación del modelo MPM a una entrada AM.

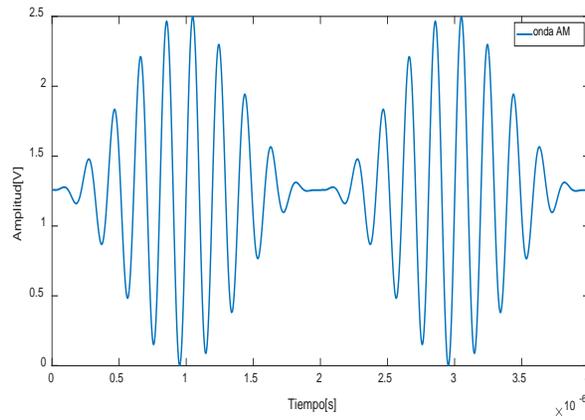


Figura 9 Entrada AM al modelo MPM escrito en Matlab.

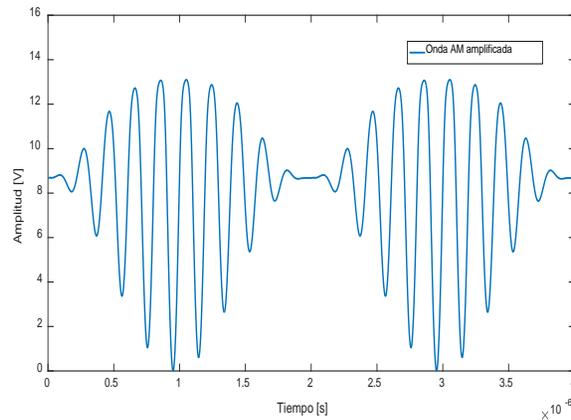


Figura 10 Salida del modelo MPM escrito en Matlab.

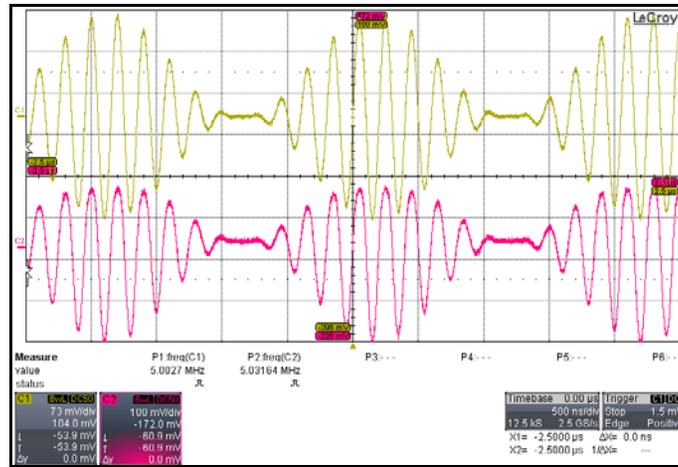


Figura 11 Entrada y salida del modelo MPM implementado en una tarjeta FPGA Stratix III.

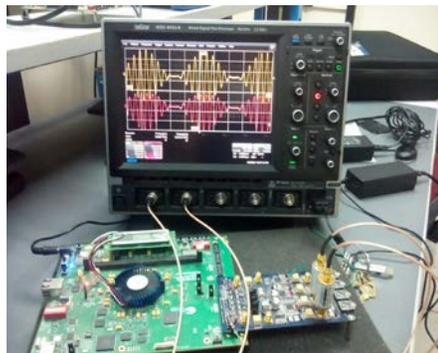


Figura 12 Banco de pruebas con la tarjeta Stratix III emulando el amplificador NXP 10 W.

4. Discusión

Después de haber realizado la implementación del MPM completo en VHDL, es posible observar que a diferencia de aquellas implementaciones basadas en LUTs tiene un considerable aumento en el uso de recursos lógicos. Sin embargo tiene la capacidad de utilizar señal compleja tanto a la entrada como a la salida logrando tener la capacidad de modelar en una sola implementación, los efectos del RF-PA tanto en amplitud como en fase, además de poder trabajar en un rango mayor en la entrada ya que no está limitada a una cierta cantidad de direcciones como en el caso mencionado.

El MPM codificado tiene las características de no linealidad 5 y profundidad de memoria 2. Sin embargo, estas características pueden ser modificadas según se requiera para el modelado de otros modelos de RF-PA con lo que queda

manifiesta la flexibilidad del uso de la plataforma FPGA para este tipo de aplicaciones, si bien el MPM utilizado se optimizó en la parte de potenciación del valor absoluto de la entrada, para otras aplicaciones con un MPM de parámetros fijos, el sistema puede optimizarse de una manera más profunda con lo que se puede lograr un uso óptimo de los recursos disponibles en el FPGA.

5. Conclusiones

El MPM permite un modelado correcto del comportamiento del RF-PA tanto en amplitud como en fase logrando en el caso del NXP 10W un NMSE de -19.8256 dB con un ajuste de coeficientes mediante regresión lineal simple. Además de ser implementado de manera exitosa en la plataforma FPGA con un uso mediano de recursos, por lo que representa un modelo que tiene las características de complejidad moderada y exactitud aceptable, ambas deseables para la aplicación de DPD.

La codificación del MPM completo en VHDL ,permite una emulación más adecuada para el desarrollo de pre-distorsionadores que aquellas implementaciones basadas en LUTs , puesto que permite emular los efectos completos que induce el RF-PA en la señal de entrada, todo lo anterior con el fin de poder probarlos sin tener el RF-PA de manera física. Con la implementación del MPM completo en FPGA se abre la posibilidad del cálculo de coeficientes directamente en esta plataforma a través de diversos métodos de estimación particularmente la estimación por mínimos cuadrados secuenciales, además de poder realizar la DPD de manera adaptativa ajustando los parámetros del modelo inverso según cambie el comportamiento del RF-PA por calentamiento o envejecimiento de componentes.

6. Bibliografía y Referencias

- [1] Cárdenas-Valdez J. R. et al., Local Search Approach to Genetic Programming for RF-PAs Modeling Implemented en FPGA, Results of the Numerical and Evolutionary Optimization Workshop NEO 2015, Springer, pp. 67-88, 2017.

- [2] Cárdenas-Valdez J. R. et al., Modeling memory effects in RF power amplifiers applied to a digital pre-distortion algorithm and emulated on a DSP-FPGA board, *Integration, the VLSI Journal*, Volume 49, pp 49-64, 2015.
- [3] Fehri B. and Boumaiza S., Baseband Equivalent Volterra Series for Behavioral Modeling and Digital Predistortion of Power Amplifiers Driven With Wideband Carrier Aggregated Signals, *IEEE Transactions on Microwave Theory and Techniques*, vol. 62, no. 11, pp. 2594-2603, 2014.
- [4] Gilabert P.L. et al., Multi-Lookup Table FPGA Implementation of an Adaptive Digital Predistorter for Linearizing RF Power Amplifiers With Memory Effects, *IEEE Transactions on Microwave Theory and Techniques*, vol. 56, no. 2, pp. 372-384, 2008.
- [5] Hammi O. et al., A Digital Predistortion System With Extended Correction Bandwidth With Application to LTE-A Nonlinear Power Amplifiers, *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 61, no. 12, pp. 3487-3495, 2014.
- [6] Kiran V., ACPR reduction for better power efficiency using adaptive DPD, 2016 International Conference on Communication and Signal Processing (ICCSP), Melmaruvathur, pp. 0495-0498, 2016.
- [7] Ku H. y Kenney J. S., Behavioral modeling of nonlinear RF power amplifiers considering memory effects, *IEEE Transactions on Microwave Theory and Techniques*, vol. 51, no. 12, pp. 2495-2504, 2003.
- [8] Liu Y.J. et al., A Robust Augmented Complexity-Reduced Generalized Memory Polynomial for Wideband RF Power Amplifiers, *IEEE Transactions on Industrial Electronics*, vol. 61, no. 5, pp. 2389-2401, 2014.
- [9] Mkadem F. et al., Behavioral modeling and digital predistortion of Power Amplifiers with memory using Two Hidden Layers Artificial Neural Networks, 2010 IEEE MTT-S International Microwave Symposium, Anaheim, CA, pp. 656-659, 2010.
- [10] Paeth A., *Graphics Gems V (Macintosh Version)*, 1ra ed. Burlington, Elsevier Science, pp. 22-24, 2014.

- [11] Moon J. y Kim B., Enhanced Hammerstein Behavioral Model for Broadband Wireless Transmitters, *IEEE Transactions on Microwave Theory and Techniques*, vol. 59, no. 4, pp. 924-933, 2011.
- [12] Núñez Pérez J.C. et al., Flexible test bed for the behavioural modelling of power amplifiers, *COMPEL - The international journal for computation and mathematics in electrical and electronic engineering*, vol. 33, no. 1/2, pp. 355–375, 2014.
- [13] R. N. Braithwaite, A Comparison for a Doherty power amplifier linearized using digital predistortion and feedforward compensation, *2015 IEEE MTT-S International Microwave Symposium*, pp. 1-4, Phoenix, AZ, 2015.
- [14] Renteria J. et al., A novel configurable FPGA architecture for hardware implementation of multilayer feedforward neural networks suitable for digital pre-distortion technique, *2016 46th European Microwave Conference (EuMC)*, London, pp. 854-857, 2016.
- [15] Roblin P. et al, Concurrent linearization: The state of the art for modeling and linearization of multiband power amplifiers, *IEEE Microwave Magazine.*, vol. 14, no. 7, pp. 74–91, 2013.
- [16] Rushton A., *VHDL for logic synthesis*, 3ra ed. Chichester, John Wiley and Sons, 2011.
- [17] Wood J. et al., The Evolution of PA Linearization, *IEEE Microwave Magazine*, no. 2, pp. 32–40, 2016.
- [18] Zhai J. et al., Dynamic Behavioral Modeling of Power Amplifiers Using ANFIS-Based Hammerstein, en *IEEE Microwave and Wireless Components Letters*, vol. 18, no. 10, pp. 704-706, 2008.