Síntesis e implementación en FPGA de un mapa caótico con PDF Gaussiana

L. De Micco y H. A. Larrondo
Departamentos de Física y de Ingeniería Electrónica
Facultad de Ingeniería, Universidad Nacional de Mar del Plata,
Mar del Plata, Argentina. - CONICET
Email: {ldemicco, larrondo}@fi.mdp.edu.ar

***Abstract*—Los generadores de Ruido Blanco Gaussiano AWGN (Additive White Gaussian Noise) constituyen un insumo básico para la medición de los sistemas de comunicaciones digitales. En la actualidad existen varios métodos de generación de AWGN que parten de secuencias aleatorias con PDF (Probability Density Function) uniforme y requieren implementar en hardware operaciones complejas. En este trabajo se diseña e implementa en hardware un generador de ruido gaussiano basado en un mapa caótico. La ventaja radica en el hecho de que los sistemas caóticos deterministas son descriptos por ecuaciones alineales simples, y por lo tanto son sencillos de implementar en hardware. Para lograr que la secuencia generada presente la PDF deseada se sintetiza el mapa caótico, que será el corazón del sistema, mediante un método basado en la teoría de las matrices positivas. La *calidad* de las secuencias generadas es evaluada mediante cuantificadores de aleatoriedad. La implementación en hardware se realiza en una FPGA Cyclone III EP3C120F780C7, empleando la placa de desarrollo 3C120 Development Board de ALTERA.**

I. Introducción

El canal con ruido gaussiano es un estandard en la evaluación de los sistemas de comunicaciones ya que constituye una buena aproximación a muchos canales reales. Los generadores de ruido gaussiano son entonces un elemento básico para la medición y prueba de los sistemas digitales. La mayoría de los métodos de generación propuestos parten de una serie temporal con histograma constante (PDF uniforme) [1]. Aplicando luego el algoritmo de Box-Muller o el método basado en el Teorema del limite central, se obtiene la serie temporal con PDF gaussiana. Un inconveniente para la implementación en hardware de estos algoritmos es que requieren la implementación de las funciones sinusoidal y logarítmica.

La implementación de sistemas caóticos es en general más simple que la de sistemas estocásticos, ya que el caos determinista se genera mediante ecuaciones no lineales sencillas. Por lo tanto, es natural que se procure utilizar señales caóticas como generadores de ruido (PRNG, Pseudo Random Number Generator) en aplicaciones en hardware. Las secuencias generadas por mapas caóticos ergódicos, luego de un transitorio (que depende de la propiedad de mixing del mapa), convergen a una única Función Densidad de Probabilidades Invariante (IPDF). Esta distrubución, como también el parametro de mixing, están reflejados por el Operador de Perron Frobenius (PF) que depende de la estructura del mapa.

En este trabajo se implementó en hardware un mapa caótico con IPDF aproximada a gaussiana y constante de mixing pequeña. La misma metodología empleada puede extenderse para implementar otros generadores, con PDFs arbitrarias, aproximadas por tramos.

II. Creación de mapa caótico con IPDF pre-especificada

La síntesis de mapas caóticos a partir de una densidad invariante deseada es un problema conocido como *el problema Inverso de Perron-Frobenius* (IFPP) [2]. En [3] se presenta una solución al IFPP, basada en la teoría de matrices positivas. Allí se utiliza una matriz estocástica, llamada matriz A, que describe la dinámica del mapa caótico. El autovector principal de la matriz A es la densidad invariante del mapa. Por lo tanto el IFPP se reduce a sintetizar una matriz que posea el autovector deseado.

Este método permite sintetizar mapas que posean densidades invariantes lineales por tramos arbitrarias y con valores de mixing también arbitrarios.

A. Matriz A

En esta matriz cada componente expresa la probabilidad de transición de un intervalo a otro. Para definir la matriz se particiona el intervalo unidad en n subintervalos, $I_1, ..., I_n$.

Cada elemento $a_{i,j}$ de la matriz A denotará la probabilidad de transición del intervalo I_i al intervalo I_j, designada $p_{i,j}$:

$$A = \begin{pmatrix} \beta_1 + \alpha_1(1-\beta_1) & \alpha_1(1-\beta_1) & \dots & \alpha_1(1-\beta_1) \\ \alpha_2(1-\beta_1) & \beta_2 + \alpha_2(1-\beta_1) & & \\ \dots & & & \dots \\ \alpha_n & \dots & & \beta_n + \alpha_n(1-\beta_n) \end{pmatrix} \quad (1)$$

La matriz A es una matriz estocástica, y es estrictamente positiva cuando $\alpha_i \geq 0$ y $0 < \beta_i < 1 \ \forall \ i \ \epsilon \ 1, ..., n$. De la teoría de matrices positivas se sabe que uno de los autovalores de la matriz A es la unidad, y su autovector asociado, x_p, corresponde a la densidad invariante del proceso gobernado por A, y tiene la siguiente forma:

$$x_p = (\frac{\alpha_1}{1-\beta_1}, ..., \frac{\alpha_n}{1-\beta_n}) \quad (2)$$

Claramente, es posible controlar este autovector eligiendo debidamente los valores de α_i y β_i de forma tal que el autovector correspondiente al autovalor unitario presente la forma deseada, y así obtener una densidad invariante deseada.

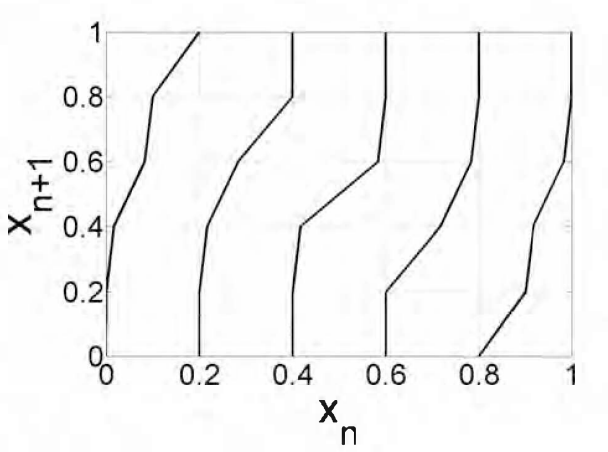

Fig. 1. Mapa caótico.

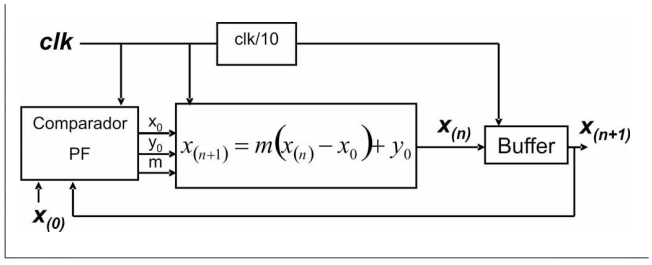

Fig. 2. Flujo de datos.

Este método tiene la limitación de que la IPDF obtenida es lineal por tramos, sin embargo, es posible aproximar cualquier curva mediante tramos lineales, cuantos más tramos se tomen mejor sera la aproximación.

En el caso de este trabajo se implementó una primera aproximación de la curva gaussiana, se tomaron cinco puntos, esto significa una matriz A de 5x5, como se puede ver en (3), y el correspondiente mapa caótico (Fig. 1).

$$A = \begin{pmatrix} 0.5013 & 0.0013 & 0.0013 & 0.0013 & 0.0013 \\ 0.0828 & 0.5828 & 0.0828 & 0.0828 & 0.0828 \\ 0.3319 & 0.3319 & 0.8319 & 0.3319 & 0.3319 \\ 0.0828 & 0.0828 & 0.0828 & 0.5828 & 0.0828 \\ 0.0013 & 0.0013 & 0.0013 & 0.0013 & 0.5013 \end{pmatrix} \quad (3)$$

III. Implementación en Hardware

En el caso de los sistemas caóticos el proceso de digitalización es crítico, errores de truncamiento y redondeo pueden producir la pérdida del comportamiento caótico.

Por esto la elección de la arquitectura y cantidad de bits empleados para la representación de los datos es un tema muy importante.

En un trabajo anterior [4] se implementó este sistema empleando punto flotante (estandard IEEE 754 de precision simple). Es bien sabido que con punto flotante se mejora la presicion, pero se consumen más recursos y la operaciones requieren más ciclos de clock.

En el caso de este trabajo emplearemos una arquitectura de punto fijo, utilizándose 10 bits para representar la parte entera, y 30 bits para la parte decimal.

En la Fig. 2 puede verse el flujo de datos del diseño. Algunos bloques para las cuales no existen modelos prediseñados de ALTERA se programaron mediante VHDL, tales como los bloques *Comparador PF* y *Buffer*. Siempre que fue posible, como es aconsejable, se utilizaron bloques de ALTERA ya que éstos están optimizados para trabajar con las placas de este fabricante. Se emplearon estos cores para implementar las operaciones de suma y multiplicación en punto fijo (LPM_MULT, LPM_ADD_SUB), como también el PLL ($ALTCLKLOCK$) para obtener el clock de salida del sistema.

Para la programación en VHDL se utilizó QUARTUS II Web Edition [5]. Este software permite no sólo compilar el diseño de forma funcional, sino también realizar el análisis temporal, la distribución en la placa, etc. Se realizó la programación del dispositivo y para verificar el correcto funcionamiento se almacenaron los datos de salida empleando el analizador lógico embebido SignalTap que provee ALTERA [6].

Como el mapa caótico es lineal por tramos (Fig. 1) se realizó la implementación de una forma muy sencilla: cada valor x_{k+1} a generar es el iterado del valor actual x_k, tomando en cuenta la recta correspondiente al tramo en el que se encuentre:

$$x_{(k+1)} = m(x_{(k)} - x_0) + y_0 \quad (4)$$

Como se dijo anteriormente, en este trabajo se implementó una primera aproximación de la curva gaussiana, se realizó un análisis para determinar cuál es la mínima cantidad de puntos necesarios para obtener una buena aproximación. Para esto primero se simuló empleando la herramienta Fixed-Point Toolbox de $Matlab$© [7]. Se generaron archivos con una extensión de más de 150000 valores. En la Fig. 3 puede verse el histograma obtenido aproximando la curva gaussiana con 5, 21 y 101 puntos y con una secuencia generada mediante la función Randn de $Matlab$©. Allí puede verse que empleando 21 puntos en la aproximación de la gaussiana (Fig. 3.e)) se obtiene una curva similar a la generada por la función Randn (Fig. 3.j)). En esta figura puede verse que las secuencias obtenidas con el mapa caótico presentan estructuras internas en la representación de embedding $3D$ (Figs. 3.c, 3.f y 3.i) que no estan presentes en el caso de la secuencia de $Matlab$© (Fig. 3.k) y no son detectadas por el histograma (Figs. 3.b, 3.e, 3.h y 3.j).

Para poder cuantificar el grado de aleatoriedad obtenido mediante las distintas aproximaciones y de esta forma conseguir un parámetro confiable para la selección de la cantidad de puntos a utilizar en la aproximación, se utilizaron cuantificadores de aleatoriedad para evaluar las secuencias obtenidas mediante los mapas caóticos [8].

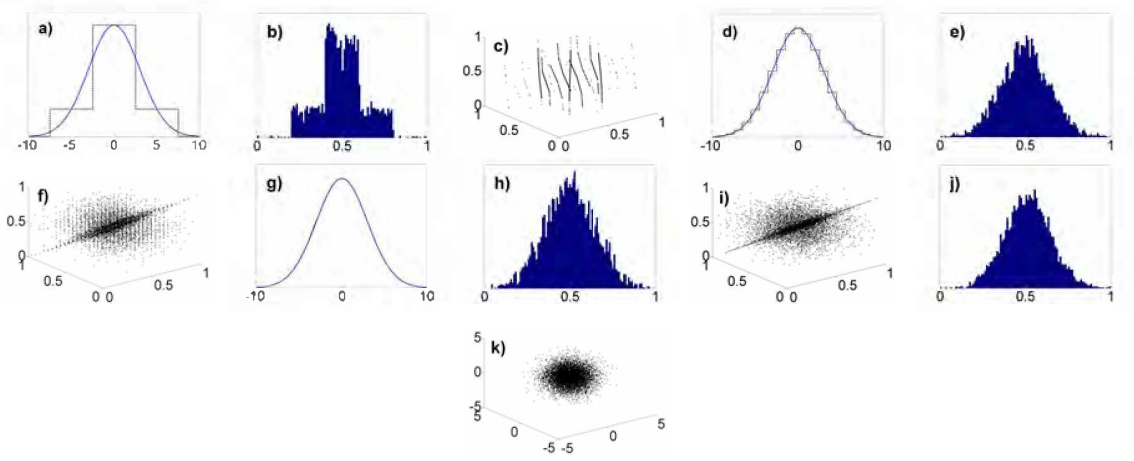

Fig. 3. Aproximación con 5 puntos: (a) gaussiana, (b) histograma. (c) $3D$ embedding; Aproximación con 21 puntos: (d) gaussiana. (e) histograma, (f) $3D$ embedding; Aproximación con 101 puntos: (g) gaussiana, (h) histograma, (i) $3D$ embedding; Empleo de la función Randn() de $Matlab^{©}$: (j) histograma . (k) $3D$ embedding.

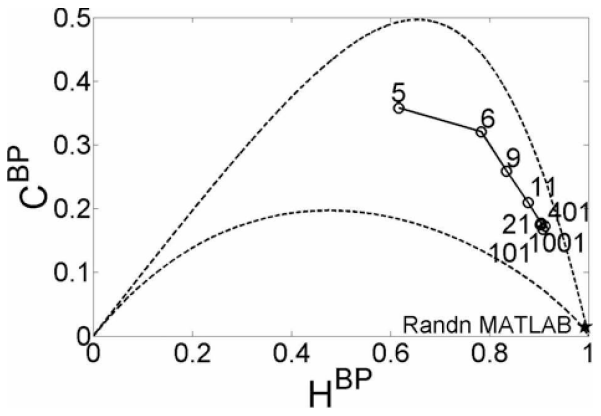

Fig. 4. Cuantificadores C^{BP} vs H^{BP} para las secuencias generadas mediante aproximaciones de la curva gaussiana con 5, 6, 9, 11, 21, 101, 401 y 1001 puntos y mediante la función Randn de $Matlab^{©}$.

IV. Cuantificadores de aleatoriedad

Basados en resultados de investigaciones previas [8] se adoptan la entropía H y la complejidad estadística C como cuantificadores para caracterizar el determinismo o estocasticidad del sistema caótico. Una discusión respecto de la conveniencia de utilizar estos cuantificadores está fuera del alcance de este trabajo pero existe una extensa bibliografía [9], [10], [11].

Los cuantificadores empleados caracterizan una dada PDF. Existen diversos métodos para asignar una PDF a una serie temporal [12], [13], [14], [15], [16], en este caso empleamos el método de Bandt y Pompe ya que produce una PDF causal. Bandt y Pompe propusieron una descripción de grano grueso usando palabras formadas por D valores consecutivos. Cada palabra es reemplazada por un número que representa el *patrón de permutación* [15], [17].

Se calculó para todas las series temporales de cada archivo el valor de H^{BP} y C^{BP} y se los representó en el plano entropía-complejidad.

V. Resultados

Los resultados de la compilación en $QUARTUS$ II muestran que el diseño ocupa un 2% de los elementos lógicos del dispositivo, un 3% de los multiplicadores embebidos de 9 bits y un 33% de los bits de memoria totales. Se consiguio una frecuencia de salida máxima de 10MHz.

En cuanto al análisis de *calidad* de las secuencias generadas con las distintas aproximaciones de la curva, los resultados obtenidos se muestran en la Fig. 4. Las curvas punteadas indican la región accesible para el caso $D = 6$ [11].

En este plano se puede apreciar la secuencia generada mediante la función Randn presenta un comportamiento cercano al punto $C^{BP} = 0$ y $H^{BP} = 1$.

Por otro lado a medida que se incrementa la cantidad de puntos empleados en la aproximación, los valores de entropía y complejidad se acercan al valor de la secuencia generada con la función Randn, sin embargo a partir de los 21 puntos se produce una saturación y aunque se mejore la aproximación no se consigue mejora en el "mezclado" interno de las secuencias.

Esto se debe a que los mapas caóticos presentan estructuras internas que son invisibles para el histograma pero son detectadas por estos cuantificadores. También pueden verse cualitativamente en las representaciones del embedding $3D$ de las secuencias (Figs. 3).

VI. Conclusión

En este trabajo se diseño e implementó una primera aproximación de un generdor de ruido gaussiano, empleando un mapa caótico. Se utilizó una FPGA Cyclone III EP3C120F780C7 de ALTERA.

Se obtuvo un PRNG en hardware con distribución lineal por tramos de acuerdo a la aproximación realizada de la curva

gaussiana. Se analizó y concluyó en que es posible obtener una buena aproximación a la PDF gaussiana a partir del empleo de 21 puntos. Para analizar las secuencias obtenidas se empleo el plano H^{BP} vs C^{BP}, mediante el cuale fue posible detectar la existencia de estructuras internas en las secuencias generadas, esto no se ve reflejado en el histograma.

AGRADECIMIENTOS

Este trabajo ha sido parcialmente financiado por Universidad Nacional del Mar del Plata y CONICET.

REFERENCES

[1] A. Ghazel. E. Boutillon, J. luc Danger. G. Gulak. and H. Laamari, "Design and performances analysis of high speed awgn communication channel emulator," in *IEEE PACRIM conference*, 2001, pp. 374–377.

[2] D. Pingel and P. Schmelcher, "Theory and examples of the inverse frobenius-perron problem for complete chaotic maps," *Chaos*, vol. 9, no. 3, pp. 357–366, 1999.

[3] A. Rogers, R. Shorten, and D. M. Heffernan. "Synthesizing chaotic maps with prescribed invariant densities," *Physics Letters A*, vol. 330, no. 6, pp. 435–441, 2004.

[4] L. D. Micco and H. Larrondo, "Implementación en fpga de ruido gaussiano para simulaciones en hardware," in *Proceedings Congreso Argentino de sistemas Embebidos CASE 2011*, 2011.

[5] ALTERA, *Quartus II Handbook Version 9.1*, 2009.

[6] ——, *Design Debugging Using the SignalTap II Logic Analyzer*, 2009.

[7] Matlab, "http://www.mathworks.com/help/toolbox/fixedpoint/."

[8] L. De Micco, H. A. Larrondo, A. Plastino, and O. A. Rosso, "Quantifiers for randomness of chaotic pseudo random number generators," *Philosophical Transactions of the Royal Society A*, vol. 367, pp. 3281–3296. 2009.

[9] O. A. Rosso, H. A. Larrondo. M. T. Martín, A. Plastino. and M. A. Fuentes. "Distinguishing noise from chaos." *Phys. Rev. Lett.*. vol. 99, pp. 154 102–154 106, 2007.

[10] L. De Micco, C. M. González, H. A. Larrondo, M. T. Martín, A. Plastino, and O. A. Rosso, "Randomizing nonlinear maps via symbolic dynamics," *Physica A*, vol. 387, pp. 3373–3383, 2008.

[11] M. T. Martín and A. Plastino, "Generalized statistical complexity measures: Geometrical and analytical properties," *Physica A*. vol. 369. pp. 439 462, 2006.

[12] K. Mischaikow, M. Mrozek, J. Reiss, and A. Szymczak, "Construction of symbolic dynamics from experimental time series," *Phys. Rev. Lett.*. vol. 82. pp. 1114–1147, 1999.

[13] W. Ebeling and R. Steuer, "Partition-based entropies of deterministic and stochastic maps," *Stochastics and Dynamics*, vol. 1, no. 1, pp. 1–17, 2001.

[14] C. Bandt and B. Pompe, "Permutation entropy: a natural complexity measure for time series," *Phys. Rev. Lett.*, vol. 88, pp. 174 102–1, 2002.

[15] K. Keller and M. Sinn. "Ordinal analysis of time series," *Physica A*. vol. 356, pp. 114–120, 2005.

[16] J. M. Amigó. L. Kocarev, and I. Tomovski, "Discrete entropy," *Physica D*, vol. 228, pp. 77–85., 2007.

[17] K. Keller and H. Lauffer, "Symbolic analysis of high-dimensional time series," *Int. J. Bifurcation and Chaos*, vol. 13, pp. 2657–2668, 2003.