SECUENCIAS BINARIAS PSEUDO ALEATORIAS GENERADAS POR UN MAPA CAÓTICO 2D

C.M. González, H. A. Larrondo, C. A. Gayoso, L. J. Arnone

Facultad de Ingeniería. Universidad Nacional de Mar del Plata – Argentina
Juan B. Justo 4302. C. P. 7600. Mar del Plata – Argentina
cmgonzal@fi.mdp.edu.ar

RESUMEN

Este trabajo presenta el desarrollo y la implementación práctica de un generador de secuencias binarias pseudo aleatorias basado en un mapa caótico 2D, el bien conocido "Cat Map". La realización física se implementa en un Dispositivo Lógico Programable de pequeño tamaño. Para mejorar su período y sus características estadísticas se lo combina con otro generador de bits pseudo aleatorios de amplia utilización, un LFSR de 32 bits. Las características más importantes de este sistemas son: a) se utiliza un mapa 2D en lugar de mapas 1D; b) su sencilla arquitectura hace que los recursos de hardware utilizados sean pocos; c) la velocidad de generación de bits es alta, debido a que sólo está limitada por el retardo de los sumadores utilizados. Se incluye la comparación con generadores standard, los resultados de pruebas estadísticas y la simulación temporal del hardware diseñado.

.

SECUENCIAS BINARIAS PSEUDO ALEATORIAS GENERADAS POR UN MAPA CAÓTICO 2D.

C.M. González, H. A. Larrondo, C. A. Gayoso, L. J. Arnone

Facultad de Ingeniería. Universidad Nacional de Mar del Plata – Argentina
Juan B. Justo 4302. C. P. 7600. Mar del Plata – Argentina
cmgonzal@fi.mdp.edu.ar

1.INTRODUCCIÓN

Los generadores de números aleatorios (y pseudo aleatorios) son utilizados en muchas áreas del conocimiento tales como Ingeniería, Economía, Física, Estadística, etc. En particular son muy utilizados en simulaciones, método de Monte Carlo, criptografía y telecomunicaciones. Ninguna secuencia producida por una máquina de estados finitos puede ser realmente aleatoria, debido a que son determinísticas por naturaleza y su salida es predecible (*"Anyone who considers arithmetical methods of producing random digits is, of course, in a state of sin," John von Neumann*). Por lo tanto, por estos medios, se generan secuencias pseudo aleatorias con un período finito, pero si ese período es suficientemente largo la pseudo aleatoriedad es suficiente para muchas de estas aplicaciones[1].

En este trabajo se presenta un método de generación de secuencias binarias pseudo aleatorias basado en un clásico mapa 2D de comportamiento caótico, estudiado y desarrollado por el matemático ruso Vladimir I. Arnold. Este mapa es denominado en la bibliografía *"Arnold's cat map"*[2], debido a que el matemático ruso utilizó imágenes de un gato para ejemplificar su comportamiento. Para lograr una implementación de bajo costo se utiliza un dispositivo lógico programable de pequeño tamaño.

La estructura de este trabajo es la siguiente: en la sección 2 se estudia el comportamiento del mapa 2D propuesto utilizado como generador de bits pseudo aleatorios y se proponen estrategias para mejorar las características estadísticas y el período de las secuencias generadas. En el punto 3 se efectúa el análisis estadístico de las secuencias binarias pseudo aleatorias generadas y se comparan los resultados con los obtenidos con otros sistemas conocidos. En el punto 4 se desarrolla la implementación física empleando el dispositivo lógico programable EPF10K20TC144-3 de Altera.

2. GENERADOR DE BITS PSEUDO ALEATORIOS BASADO EN UN MAPA 2D

El mapa 2D que se utilizó como base, es el muy bien estudiado "cat map" introducido por V. I. Arnold. La evolución de este mapa, de comportamiento caótico, sobre un cuadrado de lado unitario suele explicarse con la fotografía de un gato. La fórmula matemática es:

$$\begin{cases} x_{n+1} = (x_n + y_n) \bmod 1 \\ y_{n+1} = (x_n + 2y_n) \bmod 1 \end{cases} \quad (1)$$

donde la operación $a \bmod 1$ elimina la parte entera del número a.

La versión discretizada de este mapa se obtiene simplemente cambiando el rango de las variables x e y , éstas estarán confinadas en la red de números enteros NxN[3]. Por lo tanto el mapa será ahora:

$$\begin{cases} X_{n+1} = (X_n + Y_n) \bmod N \\ Y_{n+1} = (X_n + 2Y_n) \bmod N \end{cases} \quad (2)$$

donde la operación $A \bmod N$ es el resto de la división entera entre A y N.

Este mapa discretizado tiene un período definido y por lo tanto tiene un comportamiento pseudo caótico.

Para simplificar la implementación en hardware de este mapa se busca que el número N sea una potencia de 2 es decir del tipo 2^k (con k=1,2,.....). Esta característica hace que sólo sea necesario implementar dos sumadores enteros, ya que las operaciones de multiplicación y $mod\ 2^k$ se realizan sólo desplazando bits.

Para pequeños valores de $N=2^k$ (8, 16, 32 y 64) se hizo un estudio exhaustivo de todas las secuencias posibles. El período P de este mapa, considerando condiciones iniciales en las que al menos una de las dos variables sea un número impar, dio como resultado:

$$P = \frac{3}{4} 2^k \quad (3)$$

Utilizando este resultado como hipótesis, se generaliza para cualquier valor de $N=2^k$. En particular, para $N=2^{32}$ este resultado fue confirmado experimentalmente, implementando el mapa en la placa de desarrollo *UP1 Education Board* de Altera y midiendo el período de repetición partiendo de las condiciones iniciales ya mencionadas.

El mapa utilizado será entonces, uno con $N=2^{32}$ y período $P=3x2^{30}$. Se eligió el bit más significativo de la variable X como fuente de la secuencia de bits.

Analizando secuencias de bits generadas de este modo se concluyó que si bien estas secuencias pasan las pruebas estadísticas básicas de aleatoriedad, no pasan pruebas más exigentes tales como la batería de tests denominada "Diehard"[4] desarrollada por G. Marsaglia, de amplia utilización en la bibliografía sobre el tema. Además el período puede no ser lo suficientemente grande para algunas aplicaciones.

Por lo tanto se propone una modificación que mejora estas características. La modificación consiste en combinar, mediante una OR EXCLUSIVA, la secuencia de bits propuesta anteriormente con una secuencia de bits generada por un LFSR (Linear Feedback Shift Register) .

Los LFSR[5] son generadores de secuencias de bits pseudo aleatorias de fácil implementación en hardware, ya que sólo se trata de un registro de desplazamiento y compuertas OR EXCLUSIVA. Como ejemplo se muestra en la figura 1 un LFSR de $q=3$ etapas. El período de estos generadores es 2^q-1 .

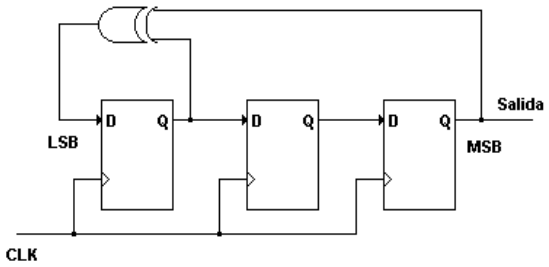

Figura 1. LFSR de tres etapas

La modificación propuesta es por lo tanto la siguiente: se combina, mediante una OR EXCLUSIVA, el bit más significativo del mapa 2D propuesto con el bit más significativo de un LFSR de $q=32bits$.

El período P_T de la secuencia resultante será:

$$P_T = \frac{P_1 \times P_2}{MCD(P_1, P_2)} \quad (4)$$

,donde P_1 y P_2 son los períodos individuales de cada generador, y $MCD(P_1,P_2)$ es el máximo común divisor entre ambos períodos.

Como $P_1=3x2^{30}$ y $P_2=2^{32}-1$ y $MCD(P_1,P_2)=3$:

$$P_T = 2^{30} \times (2^{32} - 1) \approx 4{,}61 \times 10^{18}$$

Este período es sustancialmente mayor que el propio del mapa 2D. El comportamiento estadístico de las secuencias generadas será analizado en el punto siguiente.

3. COMPORTAMIENTO ESTADÍSTICO DEL GENERADOR DE SECUENCIAS DE BITS PSEUDO ALEATORIAS

En la figura 2 puede verse la autocovarianza de una secuencia de 65536 bits del generador propuesto, en ella no se "ve" correlación entre la secuencia y la secuencia desplazada. También se ve en la figura 3, que es la transformada rápida de Fourier de la misma secuencia anterior, que el espectro es similar al del ruido. Este comportamiento justifica la elección realizada en el diseño pero sólo aporta una visión cualitativa.

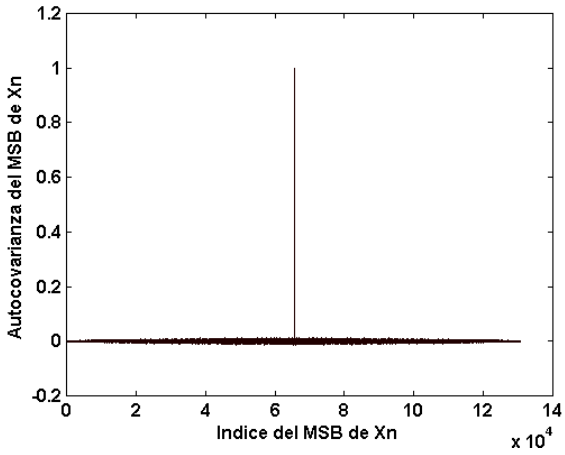

Figura 2. Autocovarianza del MSB de X_n.

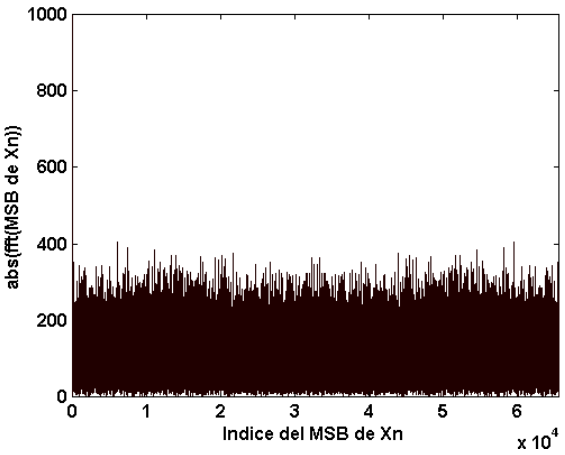

Figura 3. FFT del MSB de X_n

Para determinar las características de la secuencia generada se realizaron cinco pruebas estadísticas básicas a 5 muestras de 65536 bits cada una (prueba de frecuencia, prueba serie, prueba del Poker, prueba de rachas y autocorrelación), comúnmente utilizadas (estas pruebas son condiciones necesarias pero no suficientes para demostrar que una secuencia "luzca" aleatoria) [6][7]. Estas pruebas dieron resultados satisfactorios.

Se realizaron pruebas más exhaustivas, tales como la batería de tests denominada "Diehard" desarrollada por G. Marsaglia mencionada anteriormente. Esta batería de tests fue también utilizada para analizar secuencias de bits generadas por otros generadores pseudo aleatorios de amplia utilización. Los generadores analizados fueron:

- El generador Cat Map-LFSR propuesto.
- El generador LFSR utilizado anteriormente en forma individual.
- Un LCG (Linear Congruential Generator). Algoritmo determinístico para ser utilizado en software, en particular se utilizó el algoritmo de Lehmer usado por la función rand() de Matlab.
- Algoritmo de Marsaglia, determinístico, para ser utilizado en software.

La Tabla I resume la batería de pruebas realizada[8]. En todos los casos se utilizaron secuencias de 8x10^7 bits.

Generador / **Prueba**	Cat Map-LFSR	LFSR	LCG	Marsaglia
Birthday Spacings	Pasa	Pasa	No Pasa	Pasa
Binary Rank Test 31x31 y 32x32	Pasa	No Pasa	Pasa	Pasa
Binary Rank Test 6x8	Pasa	No Pasa	Pasa	Pasa
Monkey Tests on 20-bit Words	Pasa	Pasa	No Pasa	Pasa
Monkey Tests OPSO,,OQSO,,DNA	Pasa	No Pasa	Pasa	Pasa
Count the 1`s in a Stream of Bytes	Pasa	No Pasa	No Pasa	Pasa
Count the 1`s in Specific Bytes	Pasa	No Pasa	Pasa	Pasa
Parking Lot Test	Pasa	No Pasa	Pasa	Pasa
Minimum Distance Test	Pasa	No Pasa	Pasa	Pasa
Random Spheres Test	Pasa	Np Pasa	Pasa	Pasa
The Sqeeze Test	Pasa	Pasa	No Pasa	Pasa
Overlapping Sums Test	Pasa	Pasa	No Pasa	Pasa
Runs Up and Down Test	Pasa	Pasa	Pasa	Pasa
The Craps Test	Pasa	No Pasa	No Pasa	Pasa

Tabla I. Batería de pruebas "Diehard" aplicadas a cuatro generadores distintos.

De los resultados obtenidos se observa que la muestra ensayada del generador propuesto y la del algoritmo de Marsaglia pasan con éxito la batería de pruebas utilizada, mientras que el LFSR y el LCG no la pasan.

Es de notar que el generador propuesto es útil para ser implementado en hardware y que la complejidad de diseño no es muy elevada, por lo que su implementación física se podrá realizar utilizando pocos recursos de hardware, como veremos en el punto siguiente.

4. IMPLEMENTACIÓN EN DISPOSITIVOS LÓGICOS PROGRAMABLES

El generador fue implementado en el circuito integrado EPF10K10TC144-3 de Altera, utilizando el software de desarrollo Max-Plus II.

La figura 4 muestra la arquitectura del diseño realizado, en ella se distinguen dos bloques principales denominados CAT MAP y LFSR_32. El primero realiza el mapa 2D de la ecuación (2) con *N=32*, está implemantado en lenguaje VHDL y utiliza para la realización de los sumadores los Módulos Parametrizados de Biblioteca (LPM's) suministrados por el software. El segundo bloque es el LFSR, también está realizado en VHDL y consta de un registro de desplazamiento de 32 bits y compuertas XOR.

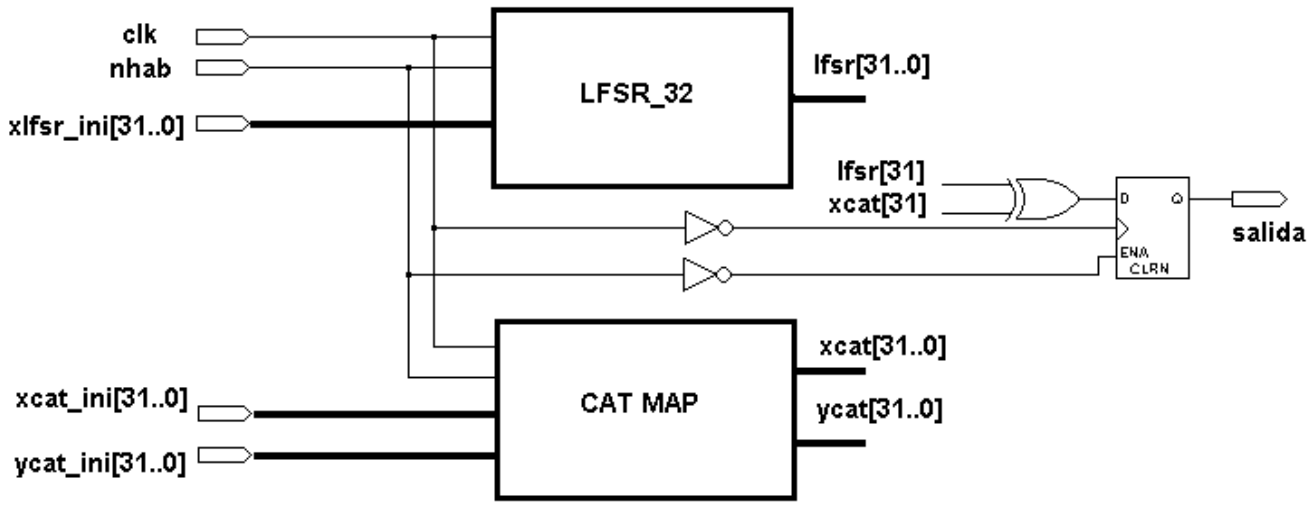

Figura 4. Diagrama de bloques del generador

El bit más significativo del bloque LFSR_32 denominado lfsr[31] se combina con el bit más significativo de la señal "xcat" de salida del bloque CAT MAP denominado xcat[31] mediante una XOR. La salida de esta XOR es registrada por un flip-flop D con el flanco descendente del reloj del sistema denominado "clk".

Las condiciones iniciales xlfsr[31..0], xcat_ini[31..1] e ycat_ini[31..0] actúan como semilla del sistema, dejando el bit menos significativo de xcat_ini en "1" para asegurar un número impar como condición inicial del mapa 2D.

El sistema completo utilizó 99 celdas lógicas, es decir sólo el 17% de los recursos del circuito integrado. La máxima frecuencia del reloj obtenida por simulación fue de 55,55 MHz.

5. CONCLUSIONES

En este trabajo se presentó un generador de secuencias binarias pseudo aleatorias realizado en hardware, obtenido a partir de la combinación de un mapa caótico 2D y un LFSR. El período obtenido es del orden de 2^62, además, se le realizaron pruebas de aleatoriedad siendo éstas satisfactorias, por lo que en principio puede ser utilizado como generador de secuencias de bits pseudo aleatorias.

En cuanto a su implementación física se realizó en un dispositivo lógico programable de pequeño tamaño, con una velocidad de generación de 55,55 Mbits/s, Es de notar que la frecuencia de trabajo está limitada por los sumadores, pudiéndose lograr velocidades mayores optimizando aquellos.

6. BIBLIOGRAFÍA.

[1] T. Stojanovski, L. Kocarev. "Chaos-Based Random Number Generators-Part. I: Analysis". IEEE Transaction on Circuits and Systems-I, vol. 48, N°3, March 2001, pp. 281-288.

[2] Gabriel Petersen. "Arnold's Cat Map".[Online] http://online.redwoods.cc.ca.us/instruct/darnold/maw/catmap.htm.

[3] Jiri Fridrich. "Symmetric Ciphers Based on Two-dimensional Chaotic Maps". International Journal of Bifurcation and Chaos, Vol. 8, No. 6 (1998) 1259-1284

[4] [Online]. http://stat.fsu.edu/~geo/diehard.html

[5] A. Menezes, P. Van Oorschot, S. Venstone. "Handbook of Applied Cryptography". CRC Press. 1997.

[6]] A, Papoulis. "Probability, Random Variables, and Stochastic Processes". Mc Graw-Hill, Inc.. Third Edition, 1991.

[7] C. González, H. Larrondo, C. Gayoso, L. Arnone. "Generación de Secuencias Binarias Pseudo Aleatorias por Medio de un Mapa Caótico 3D". . IX Workshop de IBERCHIP. 26 al 28 de marzo de 2003. La Habana, Cuba.

[8]L. Kocarev, G. Jakimoski. "Pseudorandom Bits Generated by Chaotic Maps". IEEE Transaction on Circuits and Systems-I, vol. 50, N°1, January 2003.