"CONGRESO INTERNACIONAL DE INVESTIGACIÓN E INNOVACIÓN 2014"
Multidisciplinario
10 y 11 de abril de 2014, Cortazar, Guanajuato, México
ISBN: 978-607-95635

Comparativa de Encriptadores de 32 bits basados en LFSRs Implementados en Hardware

N. A. Rodríguez*, A.G Hernández, J.A. Soto–Cajiga, L. Nava
Rodrigo Hernández y Hugo Jiménez
Centro de Ingeniería y Desarrollo Industrial, Av. Playa Pie de la Cuesta 702,
Desarrollo San Pablo, 76130 Querétaro, Qro., México
{noeamir,alex.02.02.88,rodrigoherz,hugo.jimenez}@gmail.com
{Jsoto, lnava}@cidesi.mx
http://www.cidesi.com

Abstract. This paper presents the implementation of one encryptor maximum equi-distribution of pseudo-random number of 32 bits based on the combination of Linear Feedback Shift Registers (LFSRs),on FPGA, this encryptor is selected from the comparison of three LFSRs. the selected encryptor is implemented for 3 independent data channels parallel, the simulation was performed using MATLAB ® and Xilinx Design Tools ®.

Keywords: LFSR, Pseudo-random, encryptor, FPGA

1 INTRODUCCIÓN

El desarrollo de encriptadores de datos es comúnmente realizado en software debido a ala fácil actualización, portabilidad y flexibilidad que se tiene en el desarrollo de los algoritmos; sin embargo las implementaciones en software ofrecen limitantes físicas de

ISBN: 978-607-95635

seguridad, principalmente con respecto al almacenamiento de la llave [1]. Los dispositivos con hardware reconfigurables como los Arreglos de Compuertas Programables en Campo (FPGA) ofrecen una alternativa prometedora para la implementación de encriptadores debido a sus potenciales ventajas como son la agilidad, actualización y modificación que inyectan al algoritmo, la eficiencia de la arquitectura, el flujo de datos y la eficiencia de costo [2]; sin embargo para

ISBN: 978-607-95635

aprovechar las altas capacidades que ofrece el FPGA es necesario implementar algoritmos de encriptación donde la paralelizacón de las operaciones pueda ser a nivel bit [3].

Los Registros de Desplazamiento con Retroalimentación Lineal (LFSRs) también llamados generadores Tausworthe, están basados sobre una recurrencia lineal de módulo 2 con características de un polinomio primitivo [4], con operaciones de retroalimentación a nivel bit, estos LFSRS son ampliamente utilizados en sistemas de comunicaciones para verificar la correcta transmisión de los datos, además, gracias a sus propiedades estadísticas los LFSRs son comúnmente utilizados para codificar información contenida en una paquete de datos, por cuestiones de seguridad o para reducir los efectos del ruido [5], permiten un alto flujo de datos, producen secuencias de períodos muy largos, tienen buenas propiedades estadísticas y pueden ser fácilmente analizados con algebra abstracta [6].

Existen numerosos algoritmos de encriptación de datos basados en LFSRs para 32 bits, sin embargo en sistemas desarrollados sobre un FPGA donde el cifrado de datos es sólo una pequeña parte del sistema, es necesario que los encriptadores sean de bajo consumo de espacio y de buenas características estadísticas. 3 encriptadores de 32 bits basados sobre LFSRs han sido seleccionados, implementados y comparados en este trabajo. El primero fue un LFSR de 32 bits de periodicidad máxima con retroalimentación polinómica [7] el cual presenta la máxima periodicidad de un generador aleatorio a 32 bits, el segundo fue un generador aleatorio de un LFSR combinado con un autómata celular [8] el cual ya pasó las pruebas de aleatoriedad DIEHARD [9], y el tercero es una combinación de LFSRs que permiten obtener una máxima equi-distribución de los datos generados mediante LFSRs trinomiales [4].

El resto de este trabajo está organizado de la manera siguiente. En la sección 2 se describe la implementación de los encriptadores ,en la sección 3 se presenta la selección

ISBN: 978-607-95635

del encriptador a utilizar para los tres canales paralelos, en la sección 4 se presenta la arquitectura del encriptador seleccionado para los 3 canales, en la sección 5 se presentan los resultados obtenidos con el sistema para los tres canales. Finalmente, en la sección 6 se presentan las conclusiones de este trabajo

ISBN: 978-607-95635

2 Implementación de los encriptadores.

2.1 LFSR de 32 bits de periodicidad máxima

El LFSR de 32-bits de periodicidad máxima, es un LFSR de implementación de estilo Fibonacci [10] con los taps expresados como el polinomio $X^{32} + X^{22} + X^2 + X^1 + 1$, el cual presenta $4294967295\ (2^{32} - 1)$ salidas aleatorias. Al valor inicial de este generador se le denomina semilla, que para encriptar datos será utilizado como una llave de 32 bits con valor diferente de 0x00000000. La arquitectura propuesta para la implementación en hardware es mostrada en la Fig. 1.

El LFSR de 32 bits de periodicidad máxima está compuesto por un Registro de 32 bits, la retroalimentación es de 4 bits en cada ciclo de reloj en las posiciones 32, 22, 2 y 1 mediante la operación booleana or-exclusiva, los 32 bits del LFSR son tomados por cada ciclo de reloj para generar el dato aleatorio.

2.2 Generador aleatorio de un LFSR con un autómata celular

El generador aleatorio de un LFSR con un autómata celular se basa en un LFSR de estilo Fibonacci [10] de 37 bits, con los taps expresados como el polinomio $X^{37} + X^5 + X^4 + X^3 + X^2 + X^1 + 1$ y un autómata celular basado en la mezcla de las reglas 90 y 150 [8], con estas características este generador ya pasó las pruebas de aleatoriedad DIEHARD.

El generador aleatorio híbrido de un LFSR con un autómata celular mostrado en la Figura 2 está compuesto por un LFSR de 37 bits, y un autómata celular (CA)

ISBN: 978-607-95635

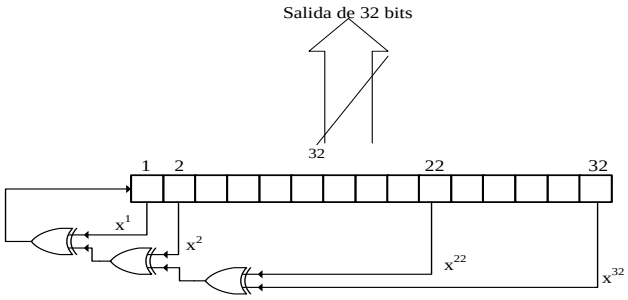


Fig. 1 Diagrama del circuito de LFSR de 32 bits para periodicidad máxima

de 16 bits, el cual opera bajo las reglas 90 y 150 las cuales van alternadas para tener la máxima aleatoriedad [11]. Este generador permite obtener 8 bits por cada ciclo de reloj, por lo cual para

ISBN: 978-607-95635

formar una palabra de 32 bits se requieren 4 ciclos de reloj, partiendo del tiempo $t = 1$ para el primer byte (B) generado, la estructura de la palabra generada es de la forma $B_{t+3}B_{t+2}B_{t+1}B_t$. La llave o semilla del encriptador son los 37 bits iniciales del LFSR.

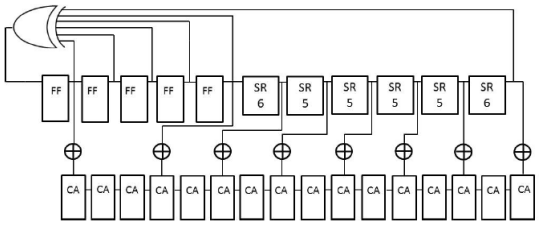


Fig. 2 Generador aleatorio mediante LFSR de 37 bits y un CA de 16 bits, 8 bits de salida **[8]**.

Tabla 1. Operadores para implementación en hardware

Operador	Descripción
$\oplus$	Operación or-exclusiva bit a bit
$\otimes$	Operación and bit bit
$\oplus$	Operación or bit a bit
>>n	Corrimiento a la derecha de n bits de un registro
<<n	Corrimiento a la izquierda de n bits de un registro

2.3 Combinación de LFSRS trinomiales para máxima equi-distribución

Implementar LFSRs basados en generadores trinomiales tiene importantes defectos estadísticos, pero combinando varios de ellos se puede generar un encriptador relativamente rápido y robusto. Con polinomios trinomiales de la forma $P_j(z) = z^{k_j} - z^{q_j}$ cumpliendo con la desigualdad $0 < 2_{q_j} < k_j$, y con paso de tamaño s_j satisfaciendo las condiciones $0 < s_j \leq k_j - q_j < k_j \leq L$. Dando los valores de $L = 32$ y $J = 4$, y utilizando

ISBN: 978-607-95635

a $s_1 = 18,\ s_2 = 2,\ s_3 = 7$ y $s_4 = 13$ y $q_1 = 6,\ \ q_2 = 2,\ q_3 = 13$ y $q_4 = 3$ se tiene entonces una periodicidad de 2^{113} CITATION Lec99 \l 1033 | [4]}.

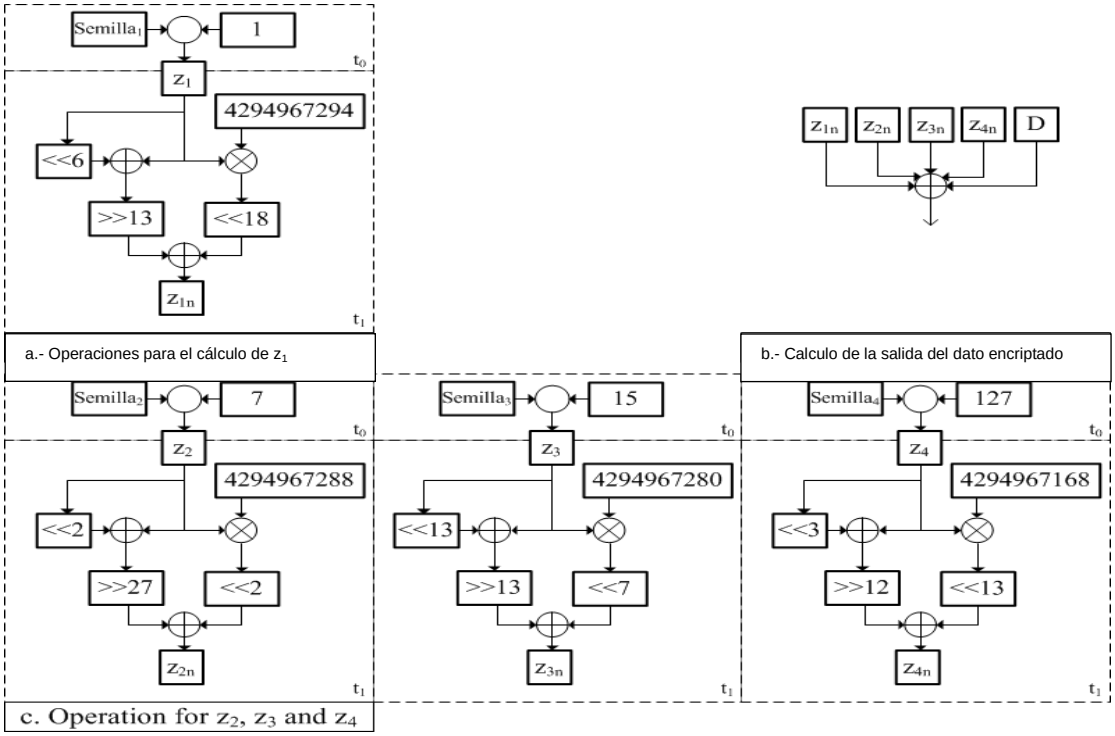
La implementación de LFSRs trinomiales para tener una máxima equi-distribución se realiza mediante 4 registros de 32 bits denominados z_1, z_2, z_3 y z_4. Las operaciones se realizan de forma paralela para mejorar el rendimiento del encriptador, estas operaciones de implementación se definen en la Tabla 1. Los cálculos sobre z_1 se muestran en la Figura 3(a), donde para un tiempo t_0 se realiza la operación or entre el valor semilla y el valor numérico 1, ya que z1 al menos debe valer 1 [4].

ISBN: 978-607-95635

Se realizaron operaciones similares para z_2, z_3 y z_4 en t_0 como se muestra en la Figura 3(c), los cuales deben valer al menos z_2=7, z_3=15 y z_4=127 [4]; por lo cual, la llave o semilla es de 4 registros de 32 bits, la arquitectura propuesta cumple con la propiedad de generar un dato aleatorio en cada ciclo de reloj debido a la paralelización de las operaciones para los cálculos de las z_ns de acuerdo a la operación mostrada en la Figura 3(b), dónde D corresponde al dato de 32 bits que se desea encriptar.

3 Comparación de resultados en la implementación

Para la comparación de los 3 generadores, se realizaron 5 pruebas, todos los diseños fueron sintetizados en un FPGA Xilinx Virtex-4^{MR} modelo XC4VLX25 de 10 752 slices [12], con el objetivo de tener un elemento en hardware en común para realizar la comparación. Las semillas o llaves utilizadas en todas las pruebas fueron las mismas para cada encriptador, recordando que en ninguno de los encriptadores se aceptará el valor semilla de 0, para el LFSR de 32 bits de longitud máxima se utilizó como semilla el valor de 2^{31} que es el valor mediano de todos los posibles valores que se pueden colocar como semilla. Para el generador aleatorio de un LFSR con un autómata celular se utiliza como valor inicial del



ISBN: 978-607-95635

c.- Operaciones para z_2, z_3, z_4

Fig. 3. (a) Operaciones para el cálculo de z_1. (b) Calculo de la salida del dato encriptado. (c) Operaciones para z_2, z_3, z_4

ISBN: 978-607-95635

autómata celular el valor 0x0180 [11], la semilla en el LFSR es el valor 2^{36} ya que es el valor mediano de todos los posibles valores que se pueden colocar. Para la combinación de LFSRs trinomiales se utiliza como semilla para $z_1=z_2=z_3=z_4=2^{31}$ que es el valor mediano de los posibles valores semilla a utilizar. La primera prueba se basa en el rendimiento en términos del flujo de datos, frecuencia máxima y el área consumida (en términos del FPGA basado en slices) [13].

De acuerdo con la Tabla 2, se puede apreciar que el LFSR de periodicidad máxima tiene los mejores resultados para su implementación debido a su relación flujo/área, esto debido a que en cada ciclo de reloj de él se toman los 32 bits; el segundo mejor encriptador es el LFSR basado en trinomios debido a su relación flujo/área ya que también permite tomar los 32 bits por cada ciclo de reloj, mediante el criterio de flujo/área el LFSR con autómata celular queda en último lugar debido a que en cada ciclo de reloj de él sólo se toman 8 bits.

La segunda prueba se basa en la descomposición de valores singulares (por sus siglas en inglés SVD) de una matriz de datos generada por cada encriptador.

Tabla 2. Comparación de rendimiento y área de los encriptadores

Encriptador	Área (slices)	Frecuencia (MHz)	Flujo de datos (Mbps)	Flujo/Área
LFSR periodicidad máxima	16	757.0	24224	1514
LFSR con autómata celular	67	434.9	3479.2	51.9
LFSRs trinomiales	88	522.4	16716.8	189.9

ISBN: 978-607-95635

$$A = \begin{bmatrix} W_1 & W_{8193} & . & W_{57344} \\ . & . & . & . \\ . & . & . & . \\ . & . & . & . \\ W_{8192} & . & . & W_{65536} \end{bmatrix}$$

Fig. 4. Matriz A de 65,536 palabras de 32 bits

Cada encriptador genera 65536 palabras de 32 bits lo que equivale a 256 Kbytes de información organizados en una matriz A de 8192x8 como se muestra en la Figura 4. El concepto de la prueba es la compresión de datos y se basa en analizar la matriz A (de 8192x8) de los datos

ISBN: 978-607-95635

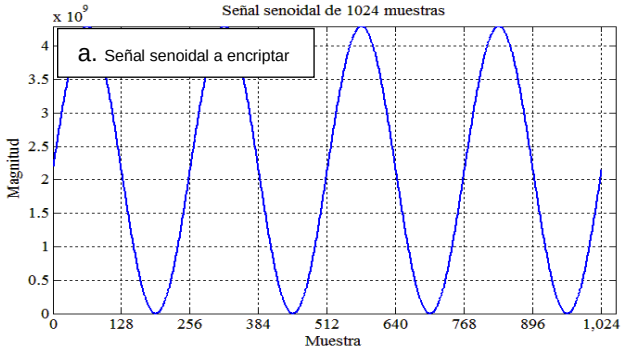
numéricos y tratar de describir una aproximación de A con un número mucho menor que los datos originales. La prueba trata a los datos como una tabla de números y pretende encontrar una aproximación que capture los rasgos más significativos [11].

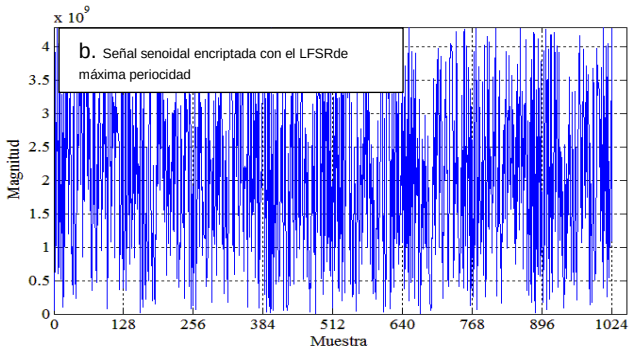
En la Tabla 3 se muestran los valores singulares ordenados y normalizados de 0 a 1 de mayor a menor de la matriz ∑ para cada encriptador; además, se realizó la matriz de 8192x8 en MATLAB con la función rand para comparar con una matriz de datos uniformemente distribuidos.

De la Tabla 3 se puede apreciar que no es viable comprimir los datos en ninguno de los encriptadores, por lo cual todos pasan la prueba, sin embargo,

Tabla 3. Descomposición de valores singulares de los encriptadores

Propiedad	LFSR longitud máxima	LFSR con CA	LFSR de trinomios	Matriz de MATLAB
Eigenvalores	0.4189	0.4182	0.4173	0.4172
	0.0855	0.0846	0.0848	0.0850
	0.0842	0.0840	0.0840	0.0846
	0.0841	0.0836	0.0836	0.0841
	0.0830	0.0834	0.0833	0.0829
	0.0820	0.0828	0.0827	0.0826
	0.0815	0.0827	0.0824	0.0820
	0.0807	0.0807	0.0819	0.0817





ISBN: 978-607-95635

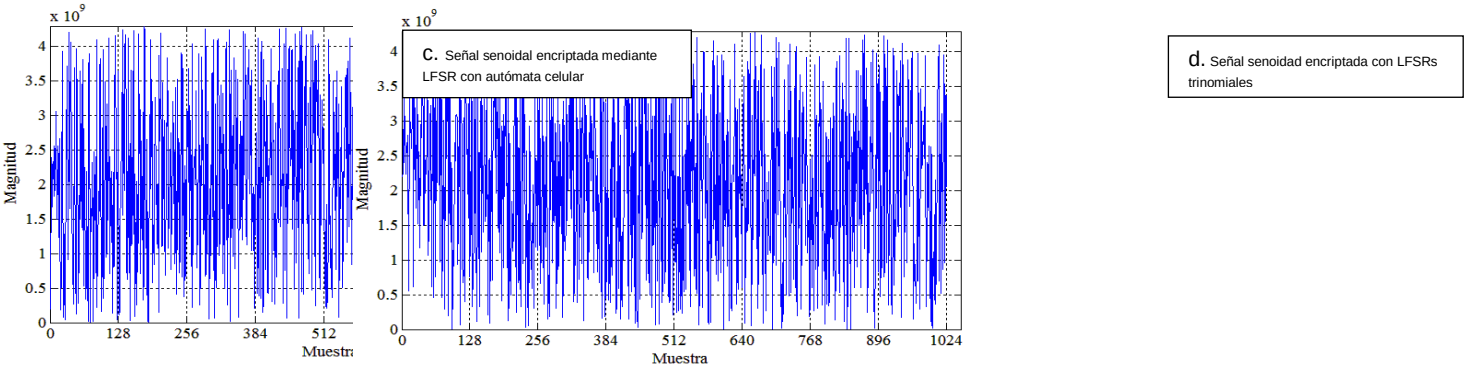


Fig. 5. (a) Señal senoidal a encriptar. (b) Señal senoidal encriptada con el LFSRde máxima periocidad. (c) Señal senoidal encriptada mediante LFSR con autómata celular. (d) Señal senoidad encriptada con LFSRs trinomiales.

ISBN: 978-607-95635

si se quitasen los 2 eigenvalores más pequeños a cada uno de los encriptadores, perderían más información el LFSR de trinomios y LFSR con autómata celular, en esta prueba el LFSR de trinomios presenta menos compresibilidad que los otros dos encriptadores.

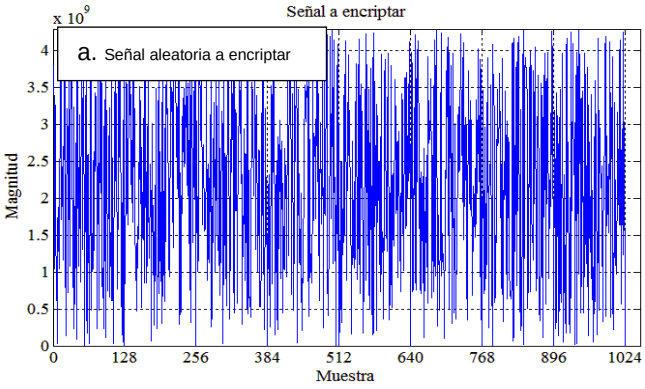
La tercera prueba se basa en encriptar una señal senoidal de 4 Hz, con 256 muestras por periodo de tiempo, lo que implica un total de 1024 palabras de 32 bits, la Figura 5(a) muestra la señal senoidal a encriptar. El objetivo de la prueba es analizar gráficamente cómo se comporta el encriptador para analizar si se puede predecir el dato a enviar. En la Figura 5(b) se muestra la señal senoidal encriptada por el LFSR de 32 bits de máxima periodicidad. En la Figura 5(c) se muestra la señal senoidal encriptada por el LFSR con autómata celular. En la Figura 5(d) se muestra la señal senoidal encriptada por el LFSR de trinomios.

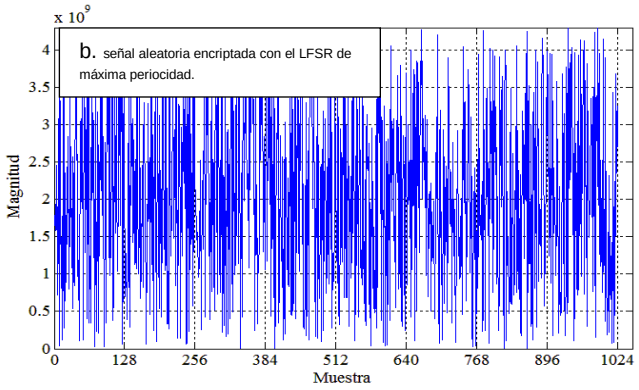
Los 3 encriptadores funcionaron correctamente al encriptar la señal senoidal, la simulación fue realizada considerando un reloj de 200MHz, para el encriptador basado en LFSR de periodicidad máxima y los LFSRs de trinomios se simularon 5.125 microsegundos para un pulso de reset y la lectura de los 1024 datos; para el encriptador basado en LFSR y el autómata celular se simularon 20.48 microsegundos. Para el proceso de des-encriptado los 3 presentaron como respuesta la señal senoidal mostrada en la Figura 5(a).

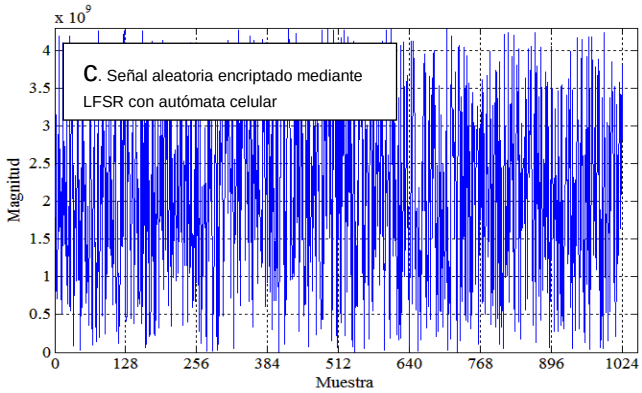
La cuarta prueba se basa en encriptar una señal de 1024 datos aleatorios de distribución uniforme en donde se cambia al valor 0 cada 128 datos para analizar el comportamiento de los encriptadores contra una señal pseudoaleatoria y su comportamiento contra un dato igualado a 0. En la Figura 6(a) se muestra la señal aleatoria a encriptar.

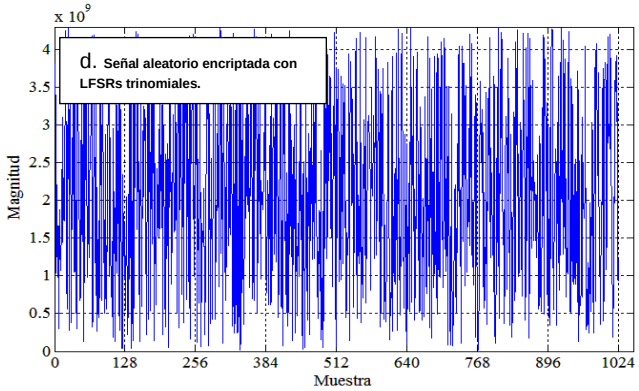
ISBN: 978-607-95635

ISBN: 978-607-95635









ISBN: 978-607-95635

Fig. 6. (a) Señal aleatoria a encriptar. (b) señal aleatoria encriptada con el LFSR de máxima periocidad. (c) Señal aleatoria encriptado mediante LFSR con autómata celular. (d) Señal aleatorio encriptada con LFSRs trinomiales.

En la Figuea 6(b) se muestra la señal aleatoria encriptada mediante el LFSR de periodicidad máxima. La Figura 6(c) muestra la señal aleatoria encriptada mediante el LFSR con Automata celular y la Figura 6(d) muestra la señal encriptada mediante los LFSRs trinomiales.

Los 3 encriptadores funcionaron correctamente al encriptar la señal aleatoria, no presentaron ningún problema al encriptar el dato cero, para todos los encriptadores el proceso de des-encriptado dio como respuesta la señal aleatoria de la Figura 6 (a).

La quinta prueba consiste en encriptar una imagen, para esta prueba se toma como ejemplo a la imagen de la Figura 7(a), la imagen es de 197x350 pixeles, el dato que se encripta es el valor de intensidad en escala de grises para tratar menos datos y no tener una simulación tan compleja si se tomarán las componentes de color.

La Figura 7(b) muestra el resultado de encriptar la imagen mediante el LFSR de periodicidad máxima, la Figura 7(c) muestra la encriptación de la imagen mediante el LFSR con

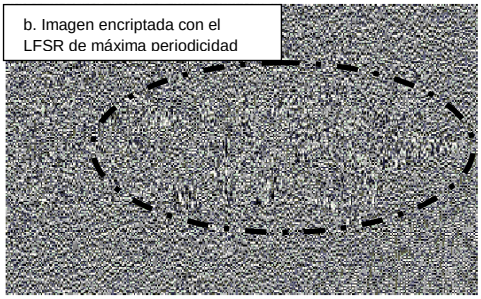
ISBN: 978-607-95635

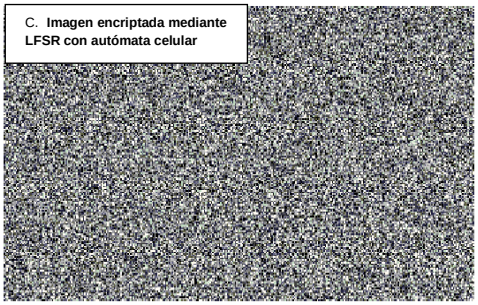
un autómata celular, la Figura 7(d) presenta el resultado de encriptar la imagen con los LFSRs basados en trinomios.

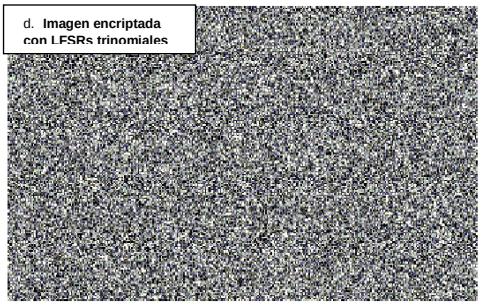
En esta prueba el LFSR de periodicidad máxima presenta pequeñas sombras, como se muestra en la Figura 7(b). En el proceso de des-encriptación los 3 encriptadores dieron como resultado la Figura 7(a).

4 Selección del encriptador para los 3 canales paralelos

Una vez realizadas las 5 pruebas en la sección anterior, no se da una calificación numérica a los encriptadores por prueba, sólo se le asigna un símbolo ✓ a los 2







ISBN: 978-607-95635

Fig. 7. (a) Imagen a encriptar. (b) Imagen encriptada con el LFSR de máxima periodicidad. (c) Imagen encriptada mediante LFSR con autómata celular. (d) Imagen encriptada con LFSRs trinomiales.

Tabla 4. Comparación de resultados en las 5 pruebas de los encriptadores

Encriptador	Prueba 1	Prueba 2	Prueba 3	Prueba 4	Prueba 5
LFSR periodicidad máxima	✓		✓	✓	
LFSR con autómata celular		✓	✓	✓	✓
LFSRs trinomiales	✓	✓	✓	✓	✓

mejores que hayan pasado la prueba, la Tabla 4 muestra los resultados obtenidos de las diferentespruebas.

ISBN: 978-607-95635

De acuerdo con la Tabla 4, para la prueba 1 el LFSR con autómata celular queda rezagado debido a su relación flujo/Área, para la prueba 2 el LFSR de periodicidad máxima es el más bajo debido a que su compresión de 2 eigenvalores quitaría menos información. Para las pruebas 3 y 4 no se detectaron anomalías en ninguno de los encriptadores, por lo cual a los 3 se les coloca el símbolo✓, para la prueba 5 el LFSR de periodicidad máxima no es de los mejores ya que la Figura 7(b) aún muestra sombras. Debido a esto el LFSR seleccionado para implementar en los 3 canales paralelos es de la combinación de LFSRs trinomiales.

5 Arquitectura del encriptador seleccionado para los 3 canales

En la Figura 8 se muestra el bloque del encriptador basado en LFSRs trinomiales, este bloque es colocado 3 veces en el FPGA con lo cual se tiene una independencia entre cada uno de ellos. Cada bloque se compone de las operaciones paralelas

Fig.8. Bloque de Encriptador basado en LFSRs trinomiales

mostradas en las Figuras 3(a) y (b), se le adicionan unas señales de entrada para el control como Inicio de operación (STO) con el cual se comienzan a generar aleatorios con cada ciclo de reloj, las entradas llave W1, W2, W3 y W4 de 32 bits cada una, con lo cual cada llave puede ser de hasta 128 bits, la entrada reset con la cual el encriptador se

ISBN: 978-607-95635

reinicia leyendo de nuevo el valor de las entradas llaves para comenzar una nueva encriptación, Din que es el dato de 32 bits a encriptar, clk que es la entrada del pulso del reloj y Dout que es el dato encriptado o des-encriptado según sea el bloque en el emisor o receptor.

Cada bloque de encriptación tiene una periodo aproximado de 2^{113} [4], para los 3 encriptadores los valores de los corrimientos (q_1, q_2, q_3, q_4) deben ser (6, 2, 13, 3) para tener la misma característica polinomial, sin embargo (s_1, s_2, s_3, s_4) pueden variar de acuerdo a las tablas establecidas en [4].

ISBN: 978-607-95635

6 Resultados obtenidos con el sistema de 3 canales de encriptación paralelos

El sistema completo de 3 canales consumió 278 slices de 10 752 slices con las que cuenta el FPGA de implementación lo que es un valor muy bajo con un excelente rendimiento, en la aplicación final las terminales de reset, clk y sto de cada máquina de estados puede interconectarse entre la terminal similar de cada máquina para generar datos aleatorios paralelos en el mismo ciclo de reloj. En el proceso de control primero se genera una señal de reset, para restablecer las señales internas, posteriormente se activa sto para iniciar la carga de las llaves en el siguiente flanco de subida. Al tercer flanco de subida del reloj se observa cómo se empiezan a generar los datos aleatorios en la salida del FPGA.

7 Conclusiones

En este trabajo 3 generadores pseudo-aleatorios son implementados en hardware y comparados bajo 5 pruebas que consideran desde el rendimiento en cuestión de velocidad-espacio, hasta su comportamiento en la encriptación de imágenes. Estos encriptadores fueron codificados en lenguaje VHDL y sintetizados en un FPGA. El encriptador seleccionado es el de LFRSs trinomiales debido a sus caracteristicas de periodicidad, sus operaciones se pueden paralelizar y con esto generar un dato aleatorio por cada ciclo de reloj, además, presenta muy baja compresibilidad. Finalmente la arquitectura propuesta para encriptar datos de forma aleatoria presenta un muy buen rendimiento en cuestión de espacio-velocidad, las llaves para cada generador pueden ser de 128 bits, y con una periodicidad de 2^{113}.

ISBN: 978-607-95635

ISBN: 978-607-95635

Referencias

[1] B. Schneier, Applied Cryptography., 2nd ed. ed., New york: John Wiley & Sons, 1996.

[2] A. J. Elbirt, W. Yip., B. Chetwynd and C. Paaer, "An FPGA-Based Performance Evaluation of the AES Block Cipher Candidater Algorithm Finalist.," *IEEE Transactions on Very Large Scale Integration(VLSI) Systems.,* vol. 9, no. 4, pp. 545-547, 2001.

[3] C. Cascaval, S. Chatterjee, H. Franke, K. j. Gildea and P. Pattnaik, "A taxonomy of accelerator architectures and their programinf models.," *IBM Journal of Research and Development.,* vol. 54, no. 5, pp. 1-10, 2010.

[4] P. L'ecuyer, "Tables of Maximally Equidistributed Combined LFSR Generators.," *Mathematics of Computation.,* vol. 68, no. 225, pp. 261-269, 1999.

[5] C. Mucci, L. Vanzolini, I. Mirimin, D. Gazzola, A. Deledda, S. Goller, J. Knaeblein, A. Schneider, L. Ciccarelli and F. Campi, "Implementation of Parallel LFSR-based Aplications on an Adaptive DSP featuring a Pipelined Configurable Gate Array," pp. 1444-1449, 2008.

[6] A. Molina-Rueda, F. Uceda-Ponga and C. Feregrina, "Extended period LFSR using variable TAP Function.," in *18th International Conference on Electronics, Communications and Computers.*, 2008.

[7] A. P. Kumar, P. Rajput and B. Shukla, "FPGA implementation of 8, 16 and 32 Bit LFSR with Maximum Length Feddback Polynomial using VHDL," in *2012 International Conference on Commication Systems and Network Technologies*, 2012.

ISBN: 978-607-95635

[8] J. C. Cerda, C. D. Martinez, J. M. Comer and D. k. Hoe, "An Efficient FPGA Random Number Generator using LFSRs and Celular Automata," in *2012 IEEE 55th International Mifwest Symposium on Circuits and Systems (MWSCAS)*, 2012.

[9] G. Marsaglia. [Online]. Available: http://stat.fsu.edu/-geo/diehard.html.

[10] S. Lim and A. Miller, "LFSRs as Functional Blocks in Wireless Aplicattions," 2001.

[11] P. D. Hortensius, R. D. MCLEOD, W. Pries, D. M. Miller and H. C. Card, "IEEE Transactions on Computer-aided Design," vol. 8, no. 8, pp. 842-859, 1989.

ISBN: 978-607-95635

[12] "Virtex-4 Family Overview," 2010.

[13] M. D. Galanis, P. Kitsos, G. Kostopoulos, N. Sklavos, Koufopavlou and C. E. Goutis, "Camparison of the hardware architectures and FPGA implementations of the stream ciphers.," in *Proccedings of the 2004 11th IEEE International Conference on Electronics, Circuits and Systems*, 2004.