

AUTÓMATAS CELULARES SOBRE FPGA

CELULAR AUTOMATON ON FPGA

Brayan Sáenz, Ivan Ladino, Ruth Suárez y Oscar Penagos

Fundación Universitaria Los Libertadores

Programa de ingeniería electrónica

Programa de ingeniería industrial

Universidad Nacional de Colombia

Maestría de ingeniería eléctrica

Bogotá D.C, Colombia.

E-mail: {bjsaenzc, idladinov, rmsuarezc}@libertadores.edu.co,
openagose@unal.edu.co

Abstract: Reconfigurable Hardware is generally associated with FPGAs. Strictly reconfigurable devices must support reconfiguration without the need to reprogram the devices. Cellular automata are made up of a very dense collection of reconfigurable cells, where each one performs a simple operation but together they can perform complex operations. The evolved automata, unlike the evolutionary ones, are reconfigured based on rules that are the result of an evolutionary process external to the automaton itself; The Genetic Machine can be implemented on a processor or on a PC, as is the case in this work, in which an automaton was implemented in two layers, one for linear operations and the other for non-linear operations. The Genetic machine, for practical convenience, was designed on an image filtering application and allows the rules that are reconfigured in the automaton implemented on the FPGA to be evolved.

Keywords: Cellular Automaton, Reconfigurable Hardware, Genetic Algorithms, Genetic Machine.

Resumen: Generalmente el Hardware reconfigurable se asocia a las FPGAs. Los dispositivos reconfigurables en sentido estricto deben soportar la reconfiguración sin la necesidad de reprogramar los dispositivos. Los autómatas celulares están constituidos por una colección muy densa de celdas reconfigurables, donde cada una realiza una operación simple pero en conjunto pueden ejecutar operaciones complejas. Los autómatas evolucionados a diferencia de los evolutivos se reconfiguran con base en reglas que son fruto de un proceso evolutivo externo al autómata en sí mismo; la Máquina Genética puede implementarse sobre un procesador o sobre un PC como es el caso de este trabajo, en el que se implementó un autómata en dos capas una para operaciones lineales y la otra para no lineales. La máquina Genética por conveniencia práctica, se diseñó sobre una aplicación de filtrado de imágenes y permite evolucionar las reglas que se reconfiguran en el autómata implementado sobre la FPGA.

Palabras clave: Autómata Celular, Hardware Reconfigurable, Algoritmos Genéticos, Máquina Genética.

I. INTRODUCCIÓN

Para aplicaciones como la criptografía, simulaciones físicas, procesamiento de señales y de

imágenes entre otras, se requieren cada vez más, de plataformas concurrentes y de alto computo que permitan obtener tiempos de ejecución inferiores, de tal forma que se reduzcan o eliminen los procesos

secuenciales como los implementados sobre plataformas basadas en PC.

Las aplicaciones sobre plataformas digitales involucra actualmente la necesidad de apropiar técnicas relacionadas con la inteligencia artificial y los sistemas bio-inspirados, debido al aumento de la complejidad de los problemas [1], [2], [6]. El concepto de complejidad [4], plantea la necesidad de emplear estrategias que disminuyan la complejidad en dos sentidos: uno el de la disminución de la dependencia entre los requerimientos funcionales, y el otro, la reducción de la entropía, es decir, el promedio de incertidumbre asociado con las probabilidades de cumplir con los requerimientos funcionales. Lo anterior significa que se deben apropiar nuevos paradigmas, un conjunto nuevo de conceptos y modelos de las matemáticas, sin los cuales es difícil comprender los conceptos subyacentes y concebir soluciones a problemas prácticos.

Una estrategia para plantear soluciones viables a problemas difíciles, consiste en dividir o descomponer la solución en una colección interrelacionada de componentes mucho más simples, uniformes e independientes. Ese es el caso de los Autómatas celulares que se representan básicamente por una colección de celdas elementales conectadas entre sí, con la característica de que para cada celda su estado o valor se actualiza a partir del estado de las celdas circunscritas en su vecindad, aplicando un conjunto de reglas específicas [5]. Así, un autómata en dos dimensiones por ejemplo, se percibe como una malla donde sus nodos corresponden a las celdas, y las aristas o lazos de conexión entre nodos representan la relación de vecindad para cada celda. Cada nodo contiene un valor o estado que cambia a medida que transcurre el tiempo discreto, de acuerdo a la aplicación de las operaciones o reglas a los estados de la celda y sus celdas vecinas (ver figura 1).

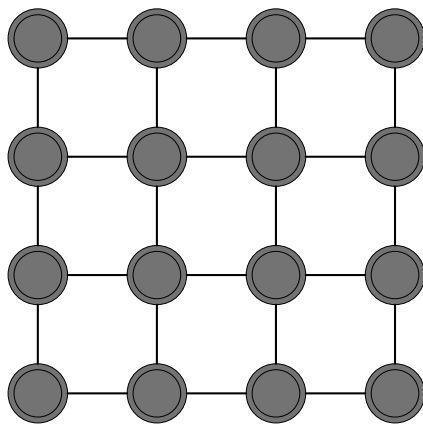


Fig. 1 Autómata Celular en dos dimensiones

La estructura de los autómatas de Von Neumann se ha considerado la base de diseño e implementación de las arquitecturas de los sistemas de computación clásicos como: microprocesadores, microcontroladores, y los PCs mismos. Por su parte los autómatas celulares como se conciben en la actualidad, determinan una granularidad (densidad de componentes) mayor que la disponible en los sistemas convencionales, dicha densidad o granularidad arrastra consigo los siguientes hechos fundamentales [5]:

- La necesidad de apropiar conceptos y modelos matemáticos que requieren el estudio de algunos temas propios de las matemáticas formales [8].
- El autómata celular funciona concurrentemente, ya que cada celda se actualiza simultáneamente con las demás celdas [5].
- La estructura de un autómata de una o dos dimensiones tiene semejanzas muy grandes con la estructura granular de los sistemas digitales discretos. [1].

Como expone Sekanina [1], [2] la herramienta ideal para sistemas digitales de esta naturaleza son las CELL MATRIX, que como se muestra en la figura 2, constan básicamente de una grilla de tablas de "Lookup" conmutables, es decir tablas de búsqueda, con entradas y salidas de datos hacia su vecindad (al igual que los autómatas celulares) y con, entrada y salida de parámetros de configuración.

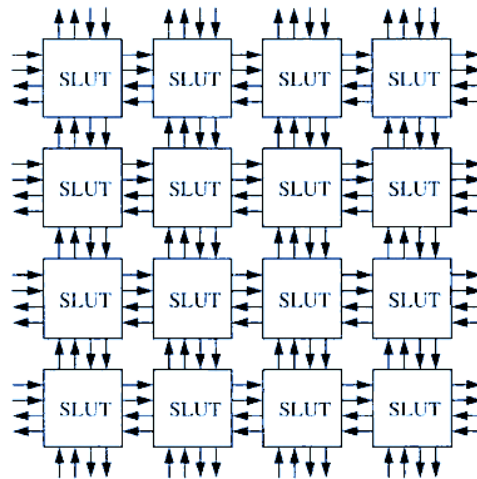


Fig.2 Cell Matrix

El inconveniente es que hoy en día, no se dispone de una herramienta de desarrollo comercial, en consecuencia, la alternativa es trabajar con sistemas de hardware reconfigurable como las plataformas de desarrollo basadas en FPGAs, pero es importante subrayar que aunque comparten ciertas características, tiene algunas diferencias que vale la pena subrayar:

- La granularidad desde el punto de vista de la densidad de componentes en una FPGA es muy inferior a la de un autómata celular o a la de una Cell Matrix [1].
- La memoria de configuración y las interconexiones ocupan el 80 a 90% del espacio en las FPGAs [1].
- La velocidad de una FPGA (Sekanina [1]) es aproximadamente 10 veces más baja que la de un ASIC [1].
- La FPGA es muy flexible sin embargo su arquitectura no coincide exactamente con la de un autómata celular.
- La diferencia más grande y sustancial entre las Cell Matrix y las FPGA, consiste en que las celdas se pueden reconfigurar sin necesidad de reprogramar el hardware, simplemente las nuevas reglas se cargan en el Cell matrix como si fuesen datos, mientras que en las FPGA se debe reprogramar cada vez que se requiere cambiar la configuración [1].
- En consecuencia de la anterior, las celdas de una Cell Matrix no solo comparten con su vecindad datos, sino además configuración.

II. AUTÓMATA EVOLUCIONADO

La alternativa planteada en este trabajo consiste en implementar autómatas reconfigurables, lo que implica que sus reglas deben depositarse en memoria para que puedan ser reescritas según se requiera, ello significa que se debe disponer de recursos de hardware adicionales para la memoria de reconfiguración y para las interconexiones. De otra parte las reglas de reconfiguración se obtienen a partir del proceso evolutivo de una Máquina Genética implementada sobre PC.

Por lo anterior se debió dividir el problema en dos componentes principales, uno el de la implementación de los autómatas reconfigurables sobre la FPGA, y el otro, la implementación de la máquina genética sobre la plataforma PC.

La limitante más grande en cuanto a recursos o hardware radica en la densidad de elementos lógicos de la FPGA. En las pruebas realizadas con autómatas en dos dimensiones y de dos estados se construyó en un proyecto anterior [9] un autómata de una capa que consume 4096 ALUTs (tabla de búsqueda adaptativa) para un total de 4096 celdas sobre una STRATIX II EP2S60 de Altera [7]. Sin embargo la implementación de autómatas para ser evaluados en el filtrado de imágenes requiere de un número de estados mucho mayor; si las imágenes en escala de grises emplean una representación típica de 256 niveles (8 bits), cada celda del autómata debe disponer entonces de 256 estados.

La actualización del estado de cada celda puede involucrar operaciones de tipo lineal y no lineal; las de

tipo lineal se puede representar básicamente por una convolución, es decir que la actualización del estado de la celda se realiza con base en una máscara aplicada sobre los vecinos (vecindad Moore), mediante una operación como la mostrada en la siguiente ecuación:

$$= \sum_{p_1, p_2 \in \{moore\} r=1} c'(\vec{n}_1, \vec{n}_2) h(\vec{p}_1 - \vec{n}_1, \vec{p}_2 - \vec{n}_2)$$

Ecuación 1. Actualización de celda por convolución

Donde h es la máscara que se aplica sobre la vecindad, $c(\vec{n}_1, \vec{n}_2)$ el estado actual de la celda, $c'(\vec{n}_1, \vec{n}_2)$ el nuevo estado y r el radio de la vecindad.

En el caso de que la operación no sea lineal se pueden emplear reglas como:

$$c'(\vec{n}) = f[c(\vec{n} + \vec{n}_1), c(\vec{n} + \vec{n}_2), \dots, c(\vec{n} + \vec{n}_m)]$$

La función f denota la operación sobre la vecindad para determinar el nuevo estado. En muchas de las aplicaciones la función f consiste básicamente en una tabla de búsqueda.

Con base en lo anterior, en las limitaciones propias de las FPGAs y, en las diferencias entre las FPGAs y los autómatas propiamente dichos, hubo que optar por una estrategia tanto para la implementación del autómata sobre la FPGA, así como también para el diseño de la Máquina Genética sobre el PC [3].

Así, la estrategia definida para el desarrollo, implementación y validación de los autómatas sobre la FPGA obedeció a:

- Diseño e implementación de un autómata en dos capas (dos autómatas)
- Implementar un autómata con vecinos Moore de radio $r=1$ y de 256 estados por celda para operaciones lineales sobre cada celda.
- Implementar un autómata vecinos Moore de radio $r=1$, de dos estados para operaciones no lineales sobre datos binarios.
- Emplear una vecindad de Moore de $r=1$ para operaciones lineales, en razón a que con dicha región el número de celdas es 9, mientras que con $r=2$ se requiere de 25 celdas, ello llevaría a un hardware muy complejo y con un consumo muy alto de recursos.
- Implementar los dos autómatas de tal forma que empleen la misma plataforma de comunicaciones para ahorrar recursos de hardware.
- Diseñar una máquina de control que controle el proceso de los dos autómatas.
- Diseñar los autómatas para que se reconfiguren (sin volver a programar la FPGA) cada vez que el usuario lo requiera.

- Implementar máscaras de orden superior (5X5) operando en cascada dos máscaras de 3X3.
- Reconfigurar los autómatas a partir de máscaras o arreglos que se depositan en la memoria de cada autómata y pueden ser actualizadas frecuentemente.
- La máscara del autómata lineal corresponde a filtros espaciales de 3X3 celdas.
- Para el autómata no lineal la regla corresponde a una tabla de búsqueda.

Con el dispositivo construido a partir de las anteriores estrategias se puede validar el funcionamiento del autómata realizando operaciones de filtrado y detección de bordes en imágenes, ya que ello requiere operaciones de alto cómputo entre píxeles tanto a nivel lineal como no lineal.

III. MÁQUINA GENÉTICA

La evolución del autómata se realizó sobre un plataforma basada en PC, ello significa que las máscaras de los autómatas (filtros y tabla de búsqueda) son el resultado del proceso de evolución de la máquina genética.

En el proceso de diseño de la Máquina Genética, hubo que evaluar diferentes opciones para el proceso de evolución y evaluación de aptitud. Se debió analizar las ventajas de trabajar en el dominio del tiempo o en el dominio de la frecuencia para la evolución de los filtros, así también se evaluó los métodos para evolución de filtros y los mecanismos para realizar filtrado no lineal.

Con base en la evaluación de las diferentes posibilidades para la máquina genética, se determinó optar por las siguientes estrategias:

- Emplear el filtrado y el ajuste del histograma para el proceso de evolución.
- Construir los cromosomas de la población a partir de dos máscaras de convolución de 3X3 y un patrón de ecualización de histograma.
- Emplear dos máscaras de 3x3 para poder emular máscaras de 5x5.
- Emplear ecualización de píxeles y filtrado para la emulación de filtrado no lineal.
- Generar la función de ecualización de histograma a partir de polinomios de Lagrange por medio de la matriz de Vandermonde.
- Emplear el histograma de las imágenes como la información para el cálculo de la función de aptitud.
- Construir una función de aptitud que mida contraste de la imagen.
- Realizar el proceso de filtrado en espectro y no por medio de la calculo directo de la

convolución para reducir el número de operaciones.

- Implementar la Transformada en Teoría de Números (NTT) para reducir la complejidad y tiempo ejecución de la transformada rápida de Fourier (FFT) en el proceso de filtrado. (NTT trabaja sobre un campo de Galois, mientras la FFT trabaja sobre un campo de los números complejos).

En consecuencia la Máquina Genética evoluciona filtros y ecualizadores hasta obtener un par de filtros y un ecualizador que aumente el contraste de la imagen en un número dado de iteraciones. La evolución se basa en el cruce y mutación de los cromosomas y se evalúa su aptitud midiendo el contraste de la imagen ecualizada y filtrada por medio de la NTT. Los filtros y el ecualizador de histograma se transmiten al autómata en la FPGA, en el cual se realiza el proceso de filtrado ecualización y además se realiza la detección de bordes empleando la medición de información direccional y el autómata no lineal [11].

Las máscaras de los filtros pasa-bajos y direccionales se pueden apreciar en la siguiente figura:

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

$$\begin{bmatrix} -1 & 0 & -1 \\ 0 & 4 & 0 \\ -1 & 0 & -1 \end{bmatrix} \begin{bmatrix} -1 & -1 & -1 \\ -1 & -8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Fig.3 Filtros básicos

En cuanto al cromosoma se dispuso de la siguiente estructura:

$\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & k \end{bmatrix}$	$\begin{bmatrix} l & m & n \\ o & p & q \\ r & s & u \end{bmatrix}$	$[y_1 \ y_2 \ y_3]$
---	---	---------------------

Fig.4 Estructura del cromosoma

Las dos matrices 3X3 corresponden a la respuesta impulso de los filtros; el vector (1x3) corresponde a los puntos de ecualización obtenidos por el proceso de evolución.

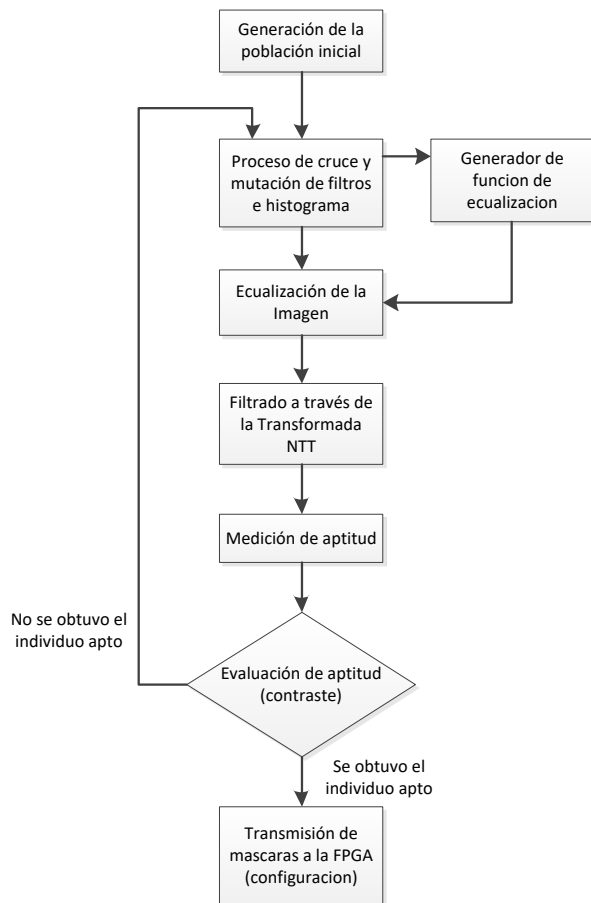


Fig.5 Diagrama en bloque de la máquina genética.

IV. IMPLEMENTACIÓN SOBRE LA FPGA

La implementación del autómata se dividió en las siguientes componentes:

- Mux_vecinos
- Mod_memoria
- Multiplicador
- Acumulador
- Reg_estado
- Autómata no lineal

En la figura 6 se presenta el multiplexor de interconexión de celdas vecinas (Mux-vecinos: C1) cuya función corresponde a la alimentar los estados de cada uno de los 9 vecinos para la actualización de la celda. En la memoria (Mod_memoria: C2) se guardan las mascarar fruto de la evolución para el proceso de filtrado lineal, además se almacenan las mascarar necesarias para el proceso de medición de orientación de la información.

El proceso de filtrado lineal se realiza mediante las mascarar almacenadas en la memoria en conjunto con

el multiplicador (multiplicador: C3) y acumulador (acumulador:C4).

El resultado del proceso es decir el estado de la celda se almacena en el un registro de 8 bits (Reg_estado: C5), cada celda del autómata alimenta a los demás celdas a partir de los datos dispuestos en su registro de estado.

El proceso no lineal se lleva a cabo en la capa no lineal del autómata (Capa_no_lineal:C6), se alimenta de los datos almacenados en los registros de estado de las celdas vecinas. Las operaciones realizadas por este modulo corresponden a:

- Ecualización de la imagen mediante tablas de “look-up”.
- Medición de la orientación de la información, con base en la aplicación de mascarar lineales (realizadas por la capa lineal) y un banco de comparadores para encontrar los máximos.
- Binarización de la imagen.

El resultado del proceso no lineal se almacena finalmente en el registro de estado de la celda, y en el caso de datos binarios su valor se almacena en el bit mas significativos del registro.

La estructura del autómata consiste en una matriz de 16X16 celdas como las mostradas en la figura 6, mas una capa de celdas para las condiciones de frontera. El proceso concurrente del autómata es gobernada a partir de la máquina de control.

La máquina de control de los dos autómatas se puede apreciar en la figura 7. La máquina de control se implementó sobre el procesador NIOS y consiste básicamente de tres módulos:

- Contador de direcciones para la lectura de datos
- Modulo de control de la máquina de estado
- Contador de 4 bits para los vecinos de las celdas

Básicamente, la función de la máquina de control correspondió al siguiente conjunto de operaciones:

- Alimentar el estado inicial al autómata.
- Almacenar en la celda las mascarar del proceso lineal y no lineal
- Almacenar las tablas de “look-up” para la capa no lineal
- Controlar el proceso de evolución de las celdas.

Fig.6 Diagrama en bloques del autómata completo.

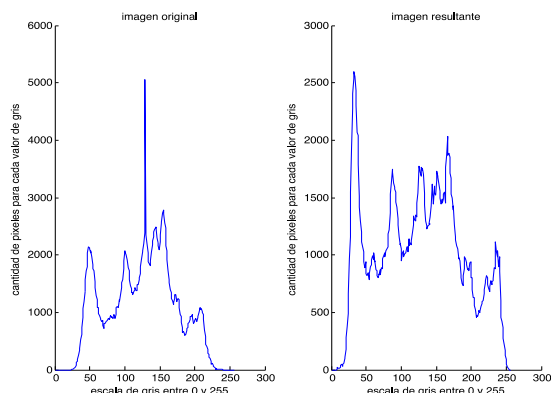


Fig.8 Histogramas



Fig.9 Resultado del filtrado- maquina genética

En la figura 9 a la izquierda se observa la imagen original y a la derecha la imagen con realce de contraste. En la siguiente figura 10 (derecha) se tiene la imagen con la información de orientación, y en la izquierda se tienen los bordes detectados por el autómatas.

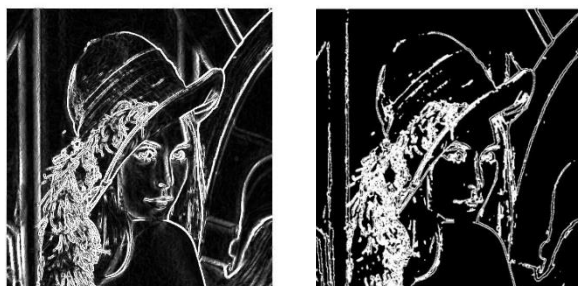


Fig.10 Resultado del filtrado- maquina genética

VI. CONCLUSIONES

- La selección de la función de aptitud determina que la población converja y por tanto que la máquina llegue a obtener un

individuo óptimo que cumpla con los requerimientos funcionales del autómatas.

- En el caso de operaciones lineales que involucren multiplicaciones y un número considerable de estados, la implementación consume bastantes recursos sobre la FPGA, en cuyo caso se debe trabajar con FPGAs orientadas al procesamiento de señales.
- Cuando se trata de autómatas binarios o de dos estados, las implementaciones pueden ser de un tamaño mucho mayor, en el caso de la STRATIX II puede significar un total de 498000 celdas.
- Es posible implementar hardware reconfigurable sobre la FPGA sin la necesidad de reprogramar el dispositivo, sin embargo seleccionar la configuración de las tablas de búsqueda puede resultar en una tarea muy compleja.
- Se concluye que los autómatas celulares se constituyen en una herramienta poderosa para operaciones de gran complejidad como las involucradas en el procesamiento de señales.

V REFERENCIAS

- [1] Sekanina L. "Evolvable Components: from theory to hardware implementations". Ed. New York: Springer, 2004.
- [2] Koncal O. Sekanina L. "Cartesian Genetic programming as an optimizer of programs evolved with geometric semantic genetic programming". Genetic Programming: 22nd European Conference, EuroGP 2019, Held as Part of EvoStar 2019, Leipzig, Germany, April 24–26, 2019, Proceedings (Lecture Notes in Computer Science (11451)).
- [3] Greenwood G., Tyrrel A. "Introduction to evolvable hardware". Ed. New Jersey: Wiley Interscience, 2007.
- [4] El-Haik B., Yang K. "The components of complexity in engineering design". *IIE Trans.*, vol. 31, No 10, pp. 925-934, Aug. 1998.
- [5] Joel L. Schiff, "Cellular Automata: A Discrete View of the World". New York Wiley 2008.
- [6] Haykin S., Kosko B. "Intelligent signal Processing". Ed. New Jersey: Wiley- IEEE, 2001.
- [7] Martin O., Odizko A., Wolfram S. "Algebraic properties of cellular automata". *Communications in Mathematical Physics.* 93, pp. 219-258. 1984.
- [8] G.A. Jullien. Number Theoretic Transforms - Extract from Number Theoretic Techniques in Digital Signal Processing Book Chapter Advances in Electronics and Electron Physics, Academic Press. Inc., (Invited), vol. 80, Chapter 2, pp. 69-163, 1991.
- [9] Sáenz, B. Hidalgo J. "Implementación de autómatas celulares sobre FPGA". Proyecto de Grado Fundación Universitaria Los Libertadores. Facultad de Ingeniería. Programa de Ingeniería Electrónica, 2011.