

Revista Cubana de Ciencias Informáticas
Vol. 9, No. 4, Octubre-Diciembre, 2015
ISSN: 2227-1899 | RNPS: 2301
<http://rcci.uci.cu>
Pág. 155-170

Tipo de artículo: Artículo original
Temática: Programación paralela y distribuida
Recibido: 30/07/2015 | Aceptado: 15/10/2015

Red de Procesadores Evolutivos para solucionar el Problema de los Tres Colores. Implementación en Hardware

Network of Evolutionary Processors to solve the Problem of the Three Colors. Hardware Implementation

José A. Castaño^{1*}, Valery Moreno², Alejandro Cabrera², Abraham Gutiérrez³ y Víctor Martínez³

¹ Universidad de las Ciencias Informáticas. Carretera a San Antonio de los Baños, km 2 ½, Torrens, Boyeros, La Habana, Cuba. CP.: 19370.

² Instituto Superior Politécnico José Antonio Echeverría. Cuba. {valery, alex}@electronica.cujae.edu.cu

³ Universidad Politécnica de Madrid, España. {abraham, victormh}@euipm.es

* Autor para correspondencia: joseantonio@uci.cu

Resumen

En este trabajo se diseña una Red de Procesadores Evolutivos (NEP) para solucionar el Problema de los Tres Colores. Para obtener el resultado, se utilizó como recurso de Hardware un FPGA. El objetivo principal de este trabajo, es demostrar la factibilidad de la implementación física de algoritmos paralelos para solucionar problemas NP – Completos. Con esta implementación se logra que la ejecución de la solución al problema se haga de forma fiable, rápida y eficaz. El proyecto fue desarrollado usando la herramienta ISE 12.1 de Xilinx, utilizando como lenguaje de descripción de hardware al VHDL. Para la simulación se utilizó la herramienta ISim 12.1, también de Xilinx. Los resultados fueron probados utilizando la plataforma Atlys Board de la compañía Digilent la cual contiene un FPGA Spartan-6 LX45, también de Xilinx. Se muestran imágenes de la herramienta empleada, del Kit de Pruebas, así como de la simulación realizada y una tabla con los datos que demuestran que la implementación de este algoritmo complejo utilizó muy pocos recursos del FPGA empleado.

Palabras clave: NEP, FPGA, VHDL, Algoritmos Paralelos, Problemas NP Completos.

Abstract

In this paper, a Network of Evolutionary Processors (NEP) is implemented, to solve the problem of the three colors. To achieve implementation, it was used as an FPGA hardware resource. The main objective of this work is to demonstrate the feasibility of the physical implementation of parallel algorithms to solve problems NP - complete. With this implementation is achieved that the implementation of the solution to be made reliably, quickly and

efficiently. The project was developed using the Xilinx ISE 12.1 tool, using hardware description language to VHDL. For the simulation tool was used ISim 12.1, also from Xilinx. The results were tested using the platform Atlys Board Company which contains a Digilent Spartan-6 LX45 FPGA, Xilinx also. Images of the tool used, the Test Kit and simulation performed and a table of data showing that the implementation of this complex algorithm used very few FPGA resources used are shown.

Keywords: NEP, FPGA, VHDL, Complex Problem, Parallel Processing.

Introducción

La implementación de algoritmos complejos en hardware, ha sido una de las tareas más desarrolladas por los investigadores del campo de los Sistemas Digitales. (Martínez, 2008) La necesidad de ejecutar algoritmos con rapidez, fiabilidad y bajos costos ha sido un factor determinante en el desarrollo de implementaciones en hardware. La aparición de circuitos integrados reconfigurables, que permiten la ejecución paralela de señales, ha permitido la implementación física de los mencionados algoritmos (Castaño, 2014), como son los FPGA (*Field Programmable Gate Array* – Campo Programable de Arreglos de Compuertas).

El desarrollo de aplicaciones sobre los FPGA se realiza a partir de diversas técnicas, siendo la más utilizada, la descripción del hardware mediante algún lenguaje. En este proyecto se utilizó el lenguaje VHDL (Ashenden, 1990), por su sencillez y por la rapidez con se pueden desarrollar los proyectos, en particular el diseño de las Redes de Procesadores Evolutivos (NEP – *Network Evolutionary Processors*).

La NEP diseñada tiene como fin brindar solución al conocido Problema de los Tres Colores [Díaz, 2008], pues permite dividir el problema complejo en varios de menor complejidad y solucionarlos de forma paralela. Este problema consiste en decidir cómo colorear cada uno de los vértices de un grafo, empleando tres colores de tal manera que después de colorearlos no existan dos vértices conectados por una arista con el mismo color.

El problema a resolver es clasificado como NP – Completo y su complejidad radica en que la solución presenta un tiempo exponencial, debido a que el algoritmo de solución se realiza de forma secuencial. Una forma de solucionarlo es procurar su solución en tiempo polinomial, de ahí la necesidad de implementarlo de forma paralela, para lo cual los FPGA son ideales.

Materiales y métodos

En este apartado, se exponen los elementos característicos de los FPGA, el Lenguaje VHDL y el Problema de los Tres Colores. Además, se documenta toda la teoría asociada con las Redes de Procesadores Evolutivos, con el objetivo de facilitar la comprensión de la solución final.

Hardware reconfigurable: FPGA

Los FPGA son circuitos integrados que tienen una estructura interna que permite ser configurada por los usuarios, con la intención de implementar funciones digitales de complejidad diversa. Son reconfigurables, lo que permite que sean programados siempre que sea necesario, sin tener que cambiar la arquitectura del sistema en el que se encuentren. [Ramírez, 2011]

Están formados básicamente por Bloques de Entrada/Salida (IOB), Bloques Lógicos Configurables (CLB) y Recursos de Interconexión (IR). Los IOB (Input/Output Block), se encargan de la comunicación de los componentes internos del dispositivo con el exterior, a través de los terminales (pines). En los CLB (Configurable Logic Block) es donde se implementa la lógica del diseño realizado. Los IR (Interconnection Resources) son los elementos que permiten conectar a los IOBs y a los CLBs.

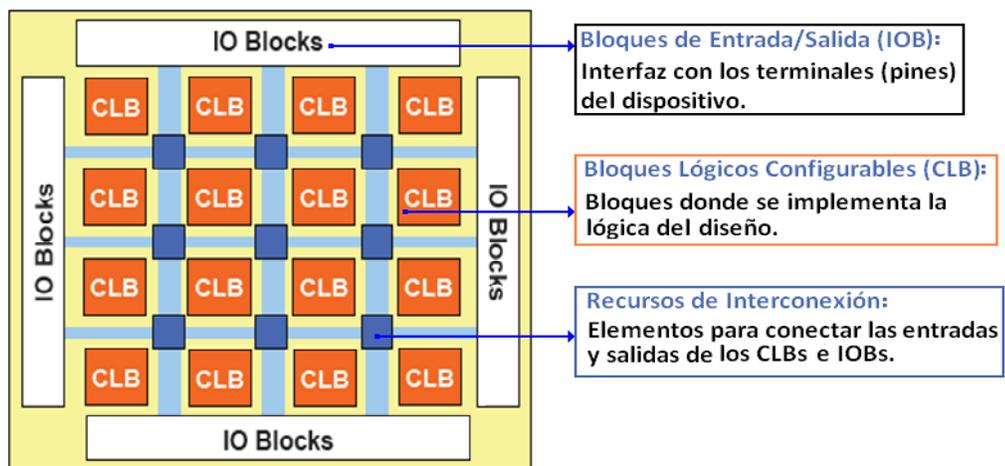


Figura 1. Estructura interna de un FPGA.

La utilización de los FPGA en lugar de otra arquitectura, se basa en cinco factores: la velocidad de trabajo del sistema diseñado, la flexibilidad del hardware reconfigurable, la reducción de costos, el bajo consumo de energía y la fiabilidad del producto final.

Para implementar un sistema digital sobre un FPGA, el mismo se programa a partir de la descripción de sus componentes en un HDL. Se debe utilizar el Entorno de Desarrollo que provea el fabricante de la FPGA a utilizar y este se encargará de la síntesis del diseño, implementación y generación del fichero de programación del dispositivo.

Lenguaje de descripción de hardware VHDL

El Lenguaje de Descripción de Hardware que se utiliza para la implementación de la NEP sobre un FPGA es el VHDL. Proviene de VHSIC – HDL (*Very High Speed Integrated Circuit – Hardware Description Language*). Está diseñado para describir la funcionalidad y la organización de sistemas digitales utilizando un lenguaje formal. (Freire, 2011)

Se redacta como texto y posee una sintaxis amplia y flexible. No es práctico tratar de conocerlo completo antes de usarlo, conociéndolo sólo parcialmente se resuelven muchos problemas. Sus principales características son (Castaño, 2014):

- Descripción textual normalizada: puede usarse para la descripción de sistemas, sin ambigüedades al expresar el comportamiento o representar la estructura de un circuito, minimizando errores y problemas de compatibilidad.
- Es un lenguaje ejecutable, adaptable a distintas metodologías de diseño, lo que permite que la descripción del hardware se materialice en una representación del mismo, utilizable por las diferentes herramientas existentes cubriendo las necesidades de todo tipo de usuarios.
- Amplio rango de capacidad descriptiva: posibilita la descripción, modelado y comprobación de sistemas desde un alto nivel de abstracción, bajando hasta el nivel de definición estructural de compuertas.
- Es independiente de la tecnología, lo cual facilita la actualización y adaptación de los diseños a los avances tecnológicos en cada momento y utilizarlos en las sucesivas fases propias del desarrollo de sistemas.
- Tiene una gran capacidad para el manejo de proyectos de grandes dimensiones. Permite descomponer un diseño y su descripción en unidades más pequeñas (modularidad).

Problema de los tres colores

El Problema de los Tres Colores [Díaz, 2008], tiene como principal aplicación la coloración de países en un mapa. Es un caso típico en el etiquetado de grafos, que consiste en la asignación de etiquetas (colores) a los elementos del grafo. El objetivo de esta asignación, es que los vértices adyacentes de un grafo no compartan el mismo color, principio que se extiende a las aristas y caras del grafo en cuestión.

Este problema, se logra representar matemáticamente empleando enteros no negativos, que codifican la presencia de cada color en un determinado vértice. De forma general se puede emplear un conjunto finito como conjunto de colores. La naturaleza del problema depende del número de colores, y no de cuales sean estos.

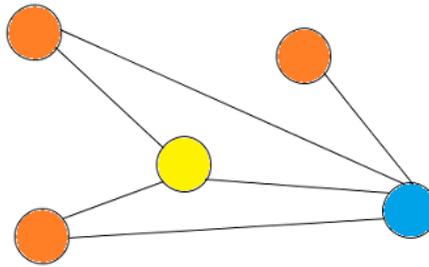


Figura 2. Coloración de vértices en grafos

Redes de procesadores evolutivos

Las NEP, son modelos computacionales de optimización y búsqueda de soluciones inspirados en la biología celular. Estos modelos, son no deterministas y masivamente paralelos y pueden abordar la solución de problemas NP - Completos en tiempo polinomial y se enmarcan dentro de lo que se denomina Computación Natural. (Martínez, 2008) Se considera a la Computación Natural como una rama de la Inteligencia Artificial. En las NEP, las células son descritas por un conjunto de palabras, evolucionando producto de mutaciones, las cuales son representadas como operaciones sobre estas palabras, asemejándose a la manera en que este proceso es llevado a cabo en las cadenas de ADN. (Păun, 1998)

Al terminar el proceso de mutación, solo las células con cadenas correctas sobrevivirán. Estas cadenas serán las soluciones a los problemas. La principal fortaleza de estos modelos radica en la simultaneidad en la ejecución del proceso, para lo cual es necesaria una arquitectura en paralelo y distribuida que consiste de una serie de procesadores simples.

Los procesadores se ubican en los nodos de un grafo completo, los cuales son capaces de manipular datos asociados con el nodo correspondiente. Cada nodo opera sobre los datos locales con la posibilidad de realizar algún tipo de mutación puntual (inserción, borrado o sustitución de un símbolo), similar a la capacidad de las células para dividirse. Estos nodos actúan como un filtro, cuyo funcionamiento está definido por alguna regla, conocidas como Reglas de Evolución. Los datos locales son entonces enviados a través la red siguiendo los protocolos definidos. Solamente los datos que sean capaces de pasar el proceso de filtrado podrán ser comunicados. Adicionalmente, los datos en los nodos son organizados en forma de largos multiconjuntos de palabras donde cada una puede aparecer en un número arbitrariamente largo de copias y donde todas las copias son procesadas en paralelo para que cada posible acción se realice. (Batard, 2012)

Conceptos básicos sobre NEP

Una Red de Procesadores Evolutivos de dimensión n se define como: $\Sigma = \{V, N_0, N_1, \dots, N_n\}$. Donde V es un alfabeto y los procesadores N_i están conectados a grafo. Una cadena χ , de longitud m que pertenece al alfabeto V , es una secuencia de símbolos $a_1 a_2 \dots a_m$ donde el símbolo $a_i \in V$ para todo $1 \leq i \leq m$. El conjunto de todas las cadenas pertenecientes a V se denota por V^* y la cadena vacía por ε . [Díaz, 2008]

Un procesador se define por $N_i = \{M_i, A_i, PI_i, PO_i\}$, donde N_i es el i – ésimo procesador evolutivo de la red. Los parámetros para cada procesador son:

- M_i : es un conjunto finito de reglas de evolución de una de las siguientes maneras (reglas básicas):
 - ◆ $a \rightarrow b, (a, b \in V)$ [Reglas de sustitución]
 - ◆ $a \rightarrow \varepsilon, (a \in V)$ [Reglas de borrado]
 - ◆ $\varepsilon \rightarrow a, (a \in V)$ [Reglas de inserción]
- A_i es un conjunto finito de cadenas sobre V , el cuál es el conjunto de cadenas iniciales sobre el nodo i . Se considera que cada cadena que aparece en un nodo, lo hace es con un número arbitrario grande de copias.
- PI_i y PO_i son subconjuntos de V que representan los filtros de entrada y salida respetivamente. Estos filtros se definen por condiciones de pertenencia, es decir, una cadena puede atravesar los filtros si pertenecen a estos.

La configuración o estado de una NEP se define como una n -tupla $C = (L_1, L_2, \dots, L_n)$, para todo $1 \leq i \leq n$. Una configuración representa el conjunto de cadenas que están presentes en cada nodo en un momento determinado. La configuración inicial de la red es $C_0 = (A_1, A_2, \dots, A_n)$. Una configuración puede cambiar lo mismo por un paso evolutivo que por uno de comunicación. Cuando el cambio es por un paso evolutivo, cada componente de la configuración es modificado en consecuencia a las reglas de evolución asociadas con el nodo i .

Formalmente se plantea que la configuración $C_1 = (L_1, L_2, \dots, L_n)$, cambia directamente a la configuración $C_2 = (L'_1, L'_2, \dots, L'_n)$ por un paso evolutivo, escrito de la forma $C_1 \sqsupset C_2$, si L'_i es el conjunto de cadenas obtenidas al aplicar las reglas de M_i a las cadenas en L_i de la siguiente forma:

- Si la misma regla de sustitución o eliminación puede reemplazar diferentes ocurrencias del mismo símbolo dentro de una cadena, todas estas ocurrencias deben ser reemplazadas dentro de las diferentes copias de esa cadena. El resultado es un multiconjunto en el cual cada cadena que puede ser obtenida aparezca en un número arbitrariamente largo de copias.
- Una regla de inserción se aplica en cualquier posición de una cadena. Nuevamente, el resultado es un multiconjunto similar al descrito en el punto anterior.
- Si más de una regla, es aplicable a una cadena, todas deben ser usadas para las diferentes copias de esa cadena.

Cuando el cambio se realiza por un paso de comunicación, cada nodo N_i envía todas las copias de las cadenas que posee que sean capaces de pasar su filtro de salida hacia todos los nodos conectados con N_i y recibe todas las copias de cadenas enviadas por cualquier nodo conectado con N_i que sean capaces de pasar su filtro de entrada.

Definición del problema de los tres colores

La solución al Problema de los Tres Colores estará basada en el siguiente grafo:

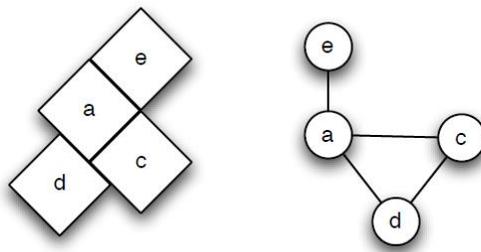


Figura 3. Grafo del problema a resolver

Para tal efecto, se define que el grafo es: $G = (\{1, 2, \dots, n\}, \{e_1, e_2, \dots, e_m\})$ con $e_t = \{k_t, l_t\}$, $1 \leq k_t \leq l_t \leq n$, $1 \leq t \leq m$. Se considera el alfabeto $U = V \cup V' \cup T \cup A$, donde $V = \{b, r, g\}$, $T = \{a_1, a_2, \dots, a_n\}$ y $A = \{A_1, A_2, \dots, A_n\}$.

Se construyen los siguientes procesadores de un NEP:

- Un procesador generador, encargado de generar todas las posibles combinaciones de colores, soluciones o no al problema y enviarlas a los siguientes procesadores. Se define como:

$$N_0 = \{\{a_1, a_2, \dots, a_n\}, \{a_i \rightarrow bA_i, a_i \rightarrow rA_i, a_i \rightarrow gA_i \mid 1 \leq i \leq n\}, \emptyset, \{A_i \mid 1 \leq i \leq n\}\}$$

- Cuatro procesadores de filtrado para cada arista del grafo $e_t = \{k_t, l_t\}$ (donde $i = \{k_t, l_t\}$):

- ◆ $Ne_t^1 = \{0, \{gA_i \rightarrow g'a_i, rA_i \rightarrow r'a_i\}, \{A_i\}, \{g', r'\}\}$
- ◆ $Ne_t^2 = \{0, \{gA_i \rightarrow g'a_i, bA_i \rightarrow b'a_i\}, \{A_i\}, \{g', b\}\}$
- ◆ $Ne_t^3 = \{0, \{bA_i \rightarrow b'a_i, rA_i \rightarrow r'a_i\}, \{A_i\}, \{b', r'\}\}$
- ◆ $Ne_t^4 = \{0, \{r'a_i \rightarrow rA_i, g'a_i \rightarrow gA_i, b'a_i \rightarrow bA_i\}, \{a_i\}, \{A_i\}\}$

En resumen, el procesador N_0 genera todas las cadenas que codifican posibles combinaciones de color y posteriormente los procesadores $Ne_t^1, Ne_t^2, Ne_t^3, Ne_t^4$ filtran dichas cadenas en la arista e_t . La repetición de dicho proceso de filtrado para el resto de las aristas provoca la obtención de la solución al problema. Es decir, en cada paso evolutivo (filtro) se obtienen combinaciones de colores posibles soluciones al problema y al terminar todo el proceso evolutivo (paso por todos los filtros) es que se obtiene la solución final. (Castaño, 2014)

La arquitectura de la NEP que se define para el Problema de los Tres Colores, a partir del grafo G , es la siguiente:

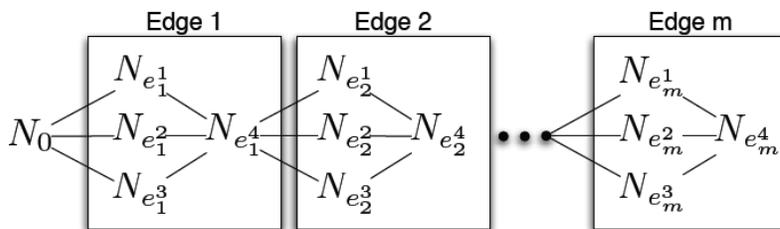


Figura 4. Arquitectura de la NEP utilizada

Implementación

Para la implementación en Hardware (FPGA) de la Red de Procesadores Evolutivos para solucionar el problema antes mencionado, se deben tener en cuenta los siguientes elementos:

- No se diseña el Procesador N_0 , encargado de generar las posibles combinaciones de colores para el grafo utilizado. Se parte de tener las 256 combinaciones de colores posibles para este problema en una memoria, la cuál debe ser implementada.
- Diseñar el bloque de filtrado a partir de la arquitectura de la NEP propuesta.
- Al terminar el proceso de filtrado, almacenar en otra memoria los datos que son soluciones para el problema planteado. Esta memoria también se debe implementar.

A partir de lo anterior, se propone el siguiente diagrama en bloques del sistema:

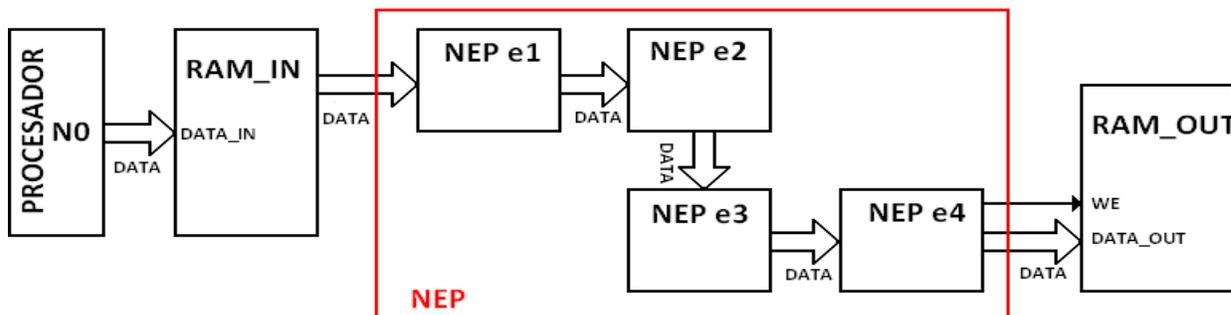


Figura 5. Diagrama en Bloques del Sistema

Bloque RAM_IN

A la memoria donde se guardan las combinaciones de colores generadas por N_0 se le llama RAM_IN. El formato de estas combinaciones de colores, es el siguiente:

<color><NODO A> <color><NODO C> <color><NODO D> <color><NODO E>

Por tanto, una combinación de colores es: bAgCrDbE. En este caso el nodo A está coloreado en azul, el nodo C en verde, el D en rojo y el E en azul.

Para procesar estos datos en el FPGA, se codificaron los colores y los nodos con 2 bits cada uno. Esto provoca que las cadenas que codifican combinaciones de colores tengan 16 bits, 4 bits por cada combinación de <color><NODO>.

Los nodos fueron codificados de la siguiente manera:

00 - Nodo A, 01 - Nodo C, 10 - Nodo D y 11 - Nodo D.

Los colores se codificaron de la siguiente forma:

00 – Rojo, 01 – Verde, 10 - Azul y 11 - Sin color.

La combinación 11 representa el caso en que un nodo no está coloreado. Esto debe ser tenido en cuenta pues en muchas de las combinaciones de colores generadas por el procesador N0 no se le asignan colores a los nodos, sobre todo en las primeras, que es cuando empieza la asignación. (Castaño, 2014)

Estas combinaciones de colores que se ubican en la memoria pueden ser soluciones o no al problema. La NEP que se diseña, se encargará de filtrar cada una de ellas y determinar cuáles son soluciones al problema planteado. La memoria se implementa a partir de la descripción de su comportamiento en VHDL y tiene un tamaño de 8 Kb. Se ubica dentro del FPGA en uno de los bloques predefinidos para memorias de este tipo.

Bloque NEP

Para el proceso de filtrado se diseña el bloque macro “NEP”, que cuenta con 4 bloques internos. Cada uno de estos bloques está en función de una arista (e_i) del grafo del problema. En estos, es donde se implementan las reglas de evolución y se nombran NEP e1 (Ne_i^1), NEP e2 (Ne_i^2), NEP e3 (Ne_i^3) y NEP e4 (Ne_i^4). Internamente cada bloque de filtrado está formado por 4 procesadores evolutivos. Es decir, las combinaciones de colores deben ser analizadas por 16 procesadores evolutivos para determinar si son solución o no al problema.

La arquitectura de cada bloque de procesamiento (filtrado), implementados en VHDL a partir de la descripción de su estructura, se muestra a continuación.

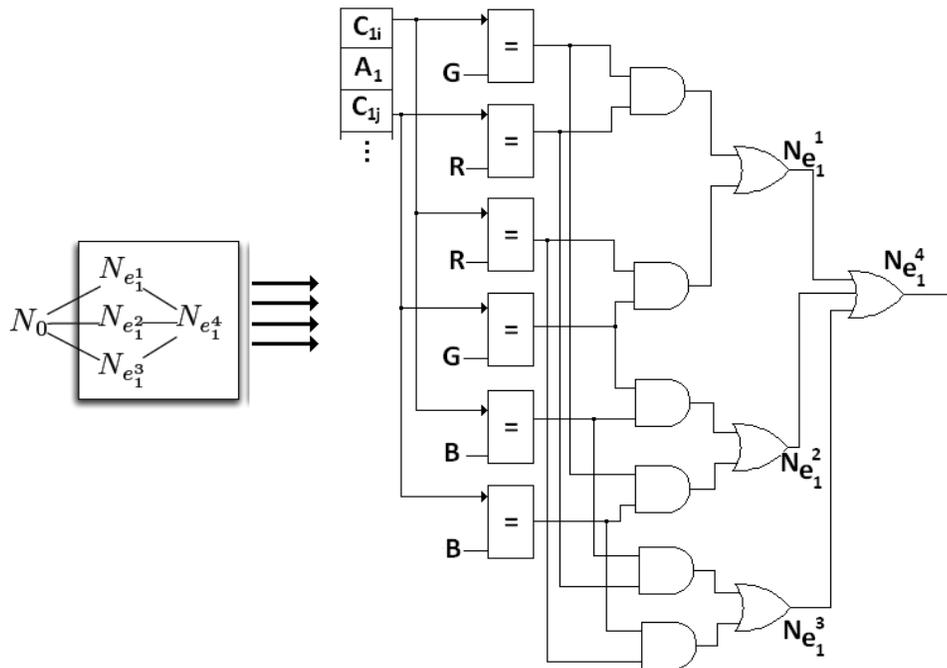


Figura 6. Arquitectura de los bloques de filtrado.

En la figura anterior se puede apreciar la implementación de las reglas de evolución, donde A1 es la arista en análisis (et) y C1i y C1j son los colores de los nodos en los extremos de la arista.

El bloque de filtrado NEP e1 (Net1), es el encargado de recibir el dato procedente de la memoria RAM_IN. En este se valida la pertinencia de los datos asociados a los colores y los nodos de toda la combinación de colores. Si los datos anteriores son válidos y se cumplen las reglas de evolución para la arista et1, el dato pasa al bloque siguiente en su formato original. De lo contrario, lo que se transfiere es el valor “0000” hexadecimal, lo cual indica que la cadena recibida no es solución al problema. Lo anterior significa, que la célula no sobrevivió al proceso de mutación.

Los bloques NEP e2 (Net2) y NEP e3 (Net3) trabajan de igual forma, actuando sobre sus aristas et2 y et3 respectivamente. En ambos casos se recibe la combinación de colores procedente del bloque anterior, si se cumplen las reglas de evolución, el dato se transmite al siguiente. De lo contrario, también se transmite el dato “0000” hexadecimal.

Al final, el bloque NEP e4 (Net4) recibe la cadena procedente de NEP e3 (Net3) para filtrar la arista et4. Si se cumplen las reglas de evolución, entonces la combinación de colores recibida es solución al problema, por ser este el

último paso evolutivo. Se procede entonces a enviar el dato para su almacenamiento en la memoria RAM_OUT y se habilita la escritura (WE) en dicha memoria. Si el dato no cumpliera con las reglas, no se guarda nada en RAM_OUT.

En este diseño, la secuencia del procesamiento de las combinaciones de colores entre los bloques de filtrado se realiza de forma serie. Esto se debe a que cada bloque solo puede comprobar el cumplimiento de las reglas de evolución de su arista (et) cuando le llega el dato desde el bloque anterior. Sin embargo, el procesamiento interno de cada bloque de filtrado es en paralelo, pues la salida toma valores según la concurrencia con que cambie la entrada. Esto último es producto de las ventajas que ofrecen los FPGA.

Bloque RAM_OUT

Este bloque corresponde a la memoria donde se guardan las combinaciones de colores generadas por el procesador N_0 que cumplen con todas las reglas de evolución, es decir, las soluciones al problema. Esta memoria es igual a RAM_IN, la cual se implementa a partir de la descripción de su comportamiento en VHDL y tiene un tamaño de 8 Kb. También se ubica dentro del FPGA en uno de los bloques predefinidos para memorias de este tipo.

En la tabla siguiente se muestran las 12 posibles soluciones al Problema de los Tres Colores, a partir del grafo G . Esto es lo que debe contener la memoria RAM_OUT al terminar el proceso de filtrado.

Tabla 1. Soluciones al Problema de los Tres Colores

| No | Combinación | Binario | Hexadecimal |
|----|-------------|------------------|-------------|
| 1 | rAbCgDbE | 0000010110100111 | 05A7 |
| 2 | gAbCrDbE | 0100100100101011 | 492B |
| 3 | gAbCrDrE | 1000010100100011 | 8523 |
| 4 | rAgCbDgE | 1000000101100011 | 8163 |
| 5 | bAgCrDgE | 0100000110100011 | 41A3 |
| 6 | rAgCbDbE | 1000000101100111 | 8167 |
| 7 | rAbCgDgE | 0100000110101011 | 41AB |
| 8 | bAgCrDrE | 0000100101100111 | 0967 |
| 9 | bArCgDrE | 0000100101101011 | 096B |
| 10 | gArCbDrE | 0100100100100011 | 4923 |
| 11 | bArCgDgE | 1000010100100111 | 8527 |
| 12 | gArCbDbE | 0000010110101011 | 05AB |

Resultados y discusión

Síntesis, Implementación y Simulación

De la síntesis del diseño, se obtiene el esquema eléctrico de la Implementación de la NEP sobre el FPGA, la cual se muestra a continuación. Se identifican claramente los componentes NEP e1, NEP e2, NEP e3, NEP e4, RAM_IN y RAM_OUT. Nótese la similitud de este esquema eléctrico con el diagrama en bloques presentado anteriormente.

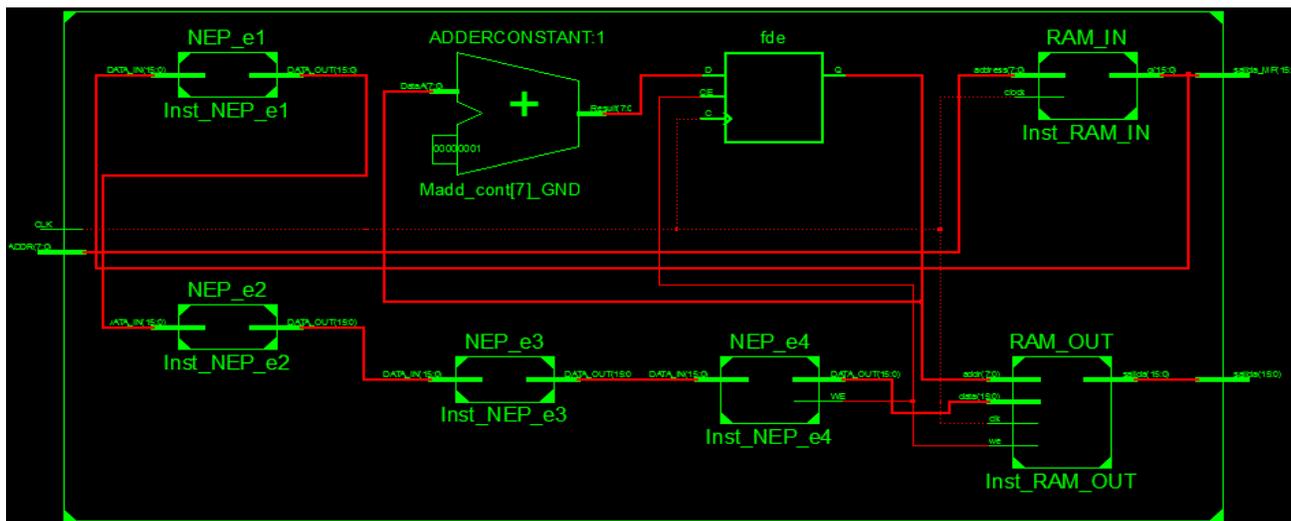


Figura 7. Esquema Eléctrico del Sistema

Luego de la implementación del sistema, se obtuvo el siguiente resumen de la utilización de los componentes internos del FPGA. Los datos demuestran que este algoritmo complejo fue implementado utilizando muy pocos recursos de la FPGA, lo cual demuestra la factibilidad de la solución.

Tabla 2. Consumo de Recursos del FPGA

| No. | Utilización de Recursos | Usados | Disponibles | Utilización |
|-----|-----------------------------|--------|-------------|-------------|
| 1 | Registros de Desplazamiento | 8 | 4,800 | 1% |
| 2 | Bloques de Entrada y Salida | 41 | 132 | 31% |
| 3 | Bloques RAM 8 Kb | 2 | 24 | 8% |

La simulación del sistema arrojó el resultado que se muestra a continuación. Se visualizan varias señales:

- Clk: reloj del sistema.
- ADDR: valores de las direcciones para acceder a la memoria RAM_IN.
- Salida_mp: datos que son extraídos de RAM_IN para ser procesados.

– Salida: soluciones al problema de los tres colores guardados en la memoria RAM_OUT.

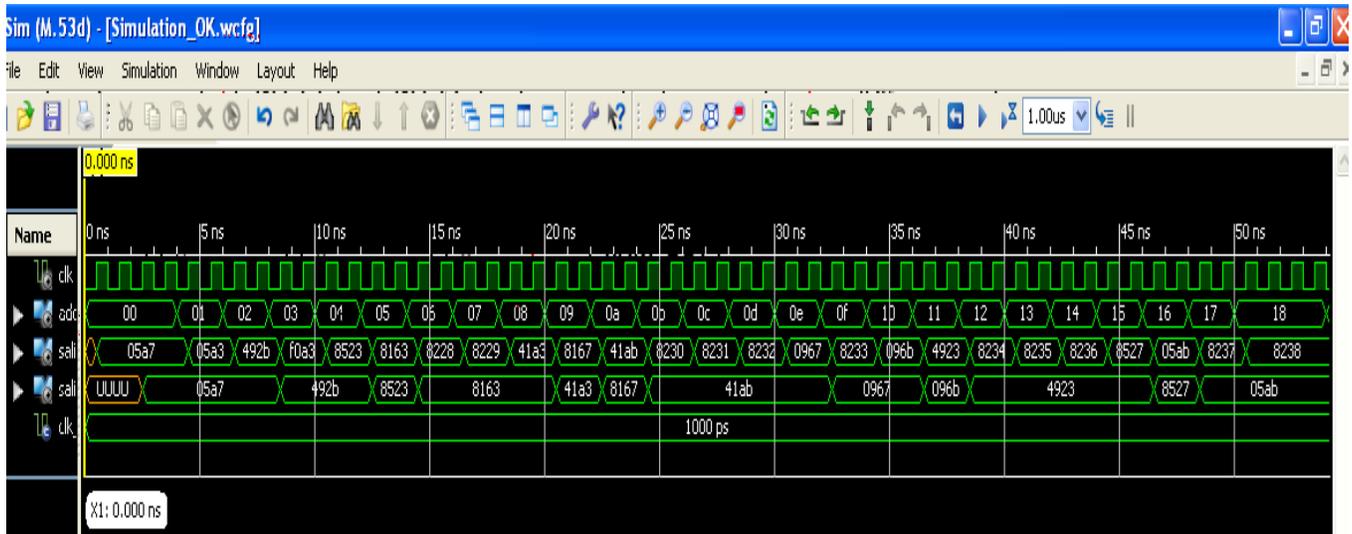


Figura 8. Simulación del sistema

Los valores de las señales “salida_mp “y “salida” tienen al inicio de la simulación estados indeterminados (color naranja). Esto se debe a que el acceso a las memorias se hace secuencial, y mientras que por sus entradas de reloj no llegue un flanco de reloj, no habrá nada que mostrar.

Los valores de las 12 soluciones al problema, en la simulación, se encuentran en hexadecimal para simplificar el tamaño de la imagen y facilitar su entendimiento. Las combinaciones asociadas a estos datos hexadecimales pueden ser consultadas en la tabla 1.

Herramientas y Dispositivos

La herramienta de software que fue utilizada para la programación, síntesis, implementación y simulación del sistema es el entorno de diseño ISE 12.1 (Entorno de Software Integrado). Este proporciona un conjunto integrado de herramientas para la implementación de sistemas sobre FPGA, producidos por la compañía Xilinx. (Xilinx, 2015)

El flujo de diseño con ISE es el siguiente: (Castaño, 2014)

1. Introducir el diseño: descripción mediante VHDL.
2. Acceso a herramientas de síntesis y simulación.
3. Implementación de los diseños.
4. Programación de los dispositivos (FPGA).

El sistema fue probado en el Kit de Desarrollo Atlys Board de Digilent. Contiene un FPGA Spartan-6 LX45 de Xilinx y otras múltiples funcionalidades. (Digilent, 2014) La programación del FPGA de dicho Kit, se realizó vía USB.

Los kits de desarrollo son plataformas de hardware que permiten la implementación práctica de sistemas digitales. Con ellos se puede comprobar el funcionamiento real de los diseños, previamente simulados. Ofrecen múltiples ventajas, como que no se necesita fabricar placas de circuito impreso, la misma placa se puede reutilizar en múltiples aplicaciones y permite el desarrollo rápido de prototipos. (Ramírez, 2011)

El FPGA del Kit está desarrollada sobre tecnología de 90nm, con una capacidad de hasta 376 pines de E/S simples ó 156 pares de E/S diferenciales. La estructura interna de esta familia está conformada por Bloques de E/S (IOB, *Input Output Block*), Bloques Lógicos Configurables (CLB, *Configurable Logic Blocks*), Manejador de Reloj Digital (DCM, *Digital Clock Manager*) y Ram de Bloque (BRAM, *Block Ram*). (Xilinx, 2015)

Conclusiones

Las Redes de Procesadores Evolutivos son propicias para la solución de problemas NP – Completos, como el de los Tres Colores. Luego de la realización este trabajo, se obtuvieron las siguientes conclusiones:

- Se logró implementar en Hardware Reconfigurable (FPGA) una Red de Procesadores Evolutivos (NEP) para solucionar el Problema de los Tres Colores.
- Se alcanza un tiempo lineal de cómputo, ya que el procesamiento de la información no se realiza en una máquina secuencial.
- Se logra realizar el procesamiento de los datos de forma paralela, debido al empleo de la FPGA.

Todo lo anterior demuestra la viabilidad de la implementación de las NEP en dispositivos de hardware.

Referencias

Ashenden, Peter J. The VHDL Cookbook. Departamento de Ciencias de la Computación, Universidad de Adelaide, Australia. Julio, 1990.

Batard Lorenzo, David & Martínez Hernando, Víctor J. “Simulación de NEP en Java”, Evento UCIENCIA 2012, La Habana, Cuba, 2012.

Implementación sobre FPGA de una Red de Procesadores Evolutivos (NEP) para solucionar el Problema de los Tres Colores. I Convención Internacional de Ciencias Técnicas y VII Conferencia Internacional de Ingeniería Eléctrica, Universidad de Oriente, Santiago de Cuba, Cuba, junio 2014. ISBN: 978-959-207-529-0

Implementación en Hardware Reconfigurable de una Red de Procesadores Evolutivos para la Solución a un Problema NP-Completo. I Conferencia Internacional UCIENCIA, Universidad de las Ciencias Informáticas, La Habana, Cuba, abril 2014. ISBN 978-959-286-026-1

Díaz Martínez, Miguel Ángel. “Redes de Procesadores Evolutivos con Filtros en las Conexiones”. Tesis Doctoral, Universidad Politécnica de Madrid, España, 2008.

Digilent. Información obtenida en fecha 26 de junio de 2014. Información en <http://www.digilentinc.com/Products/ATLYS>.

Freire Rubio, Miguel Ángel. “Manual de Introducción al lenguaje VHDL”. Universidad Politécnica de Madrid, España. Año 2011.

Martínez Hernando, Víctor. “Desarrollo de Sistemas Físicos para Implantar Modelos de Computación con Membranas”. Tesis Doctoral, Universidad Politécnica de Madrid, España (2008).

Păun, Gh., Rozenberg, G., & Salomaa, A., “DNA Computing. New Computing Paradigms”, Berlin, Springer, 1998.

Ramírez Despaine, Maikel. “Controlador Lógico Programable basado en Hardware Reconfigurable”. Tesis de Maestría, CUJAE, La Habana, Cuba, 2011.

Xilinx. Información obtenida en <http://www.Xilinx.com>. Información Publicada el 20 de febrero de 2013, consultada el 15 de mayo de 2015.