



# Algoritmos de Procesamiento Digital de Imágenes en Arquitecturas Reconfigurables

Diana Margarita Córdova Esparza, Ernesto García Domínguez, Arturo Moreno Báez

## *Algorithms of Digital Image Processing in Reconfigurable Architectures* FPGA

Recibido: abril 1, 2009

Aceptado: abril 22, 2009

Palabras clave: procesamiento de imágenes; arquitectura reconfigurable; FPGA

### **Abstract:**

*This article presents information concerning the development of algorithms of Digital Image Processing by using arrays of field-programmable gate array (FPGA). Processing modules were designed in the hardware description language (VHDL)*

**Keywords:** image processing; reconfigurable architecture; FPGA



ACE algunos años, la gran mayoría de los sistemas electrónicos estaban basados en la utilización de microprocesadores, debido a factores que contribuyeron a su popularización como: su facilidad de uso y bajo costo. A medida que las aplicaciones se hicieron más variadas, surgieron nuevos dispositivos más eficientes para la realización de algunas tareas específicas, como los Procesadores Digitales de Señales (DSPs), concebidos para implementar muchas aplicaciones tales como: protocolos de voz para internet, comunicaciones inalámbricas, sistemas de radar, procesamiento de imágenes, sistemas multimedia, etc. Sin embargo, aunque los procesadores son programables a través de software, la arquitectura del hardware del DSP no es flexible. Por lo tanto, los DSPs son limitados por la arquitectura fija del hardware y resulta inadecuado su uso para ciertas aplicaciones que podrían requerir implementaciones de funciones de procesamiento digital de señales dedicadas.

Otros dispositivos que también han revolucionado la tecnología son los microcontroladores diseñados especialmente para aplicaciones embebidas. Sin embargo, ninguno de estos dispositivos es capaz de superar la eficiencia en cuanto a procesamiento y consumo de potencia de los Circuitos Integrados de Aplicación Específica (ASIC), que

como su nombre indica sólo pueden realizar la función para la que fueron construidos.

Adicionalmente, se cuenta con otro tipo de herramientas como los Dispositivos Lógicos Programables (PLD), que en términos de procesamiento son tan eficientes como un ASIC y que pueden ser reprogramados como un microprocesador. Actualmente, gracias a los avances en la fabricación de circuitos integrados, surgen los Arreglos de Compuertas Programables en Campo (FPGA) como el resultado de la convergencia de estas dos tecnologías. Estos dispositivos se han convertido en una opción interesante a la hora de diseñar sistemas que requieren una alta capacidad de procesamiento y un bajo consumo de potencia [1].

Los circuitos FPGA suministran una solución reconfigurable y eficiente para implementar tareas de procesamiento digital de señales. Estos circuitos pueden alcanzar una mayor potencia de procesamiento de datos que los DSPs.

Tanto los DSPs, como los FPGAs presentan ventajas y desventajas para una tarea en particular. Por ejemplo, los DSPs presentan ventajas de velocidad de procesamiento, eficiencia en el consumo de energía y buena relación desempeño-costos comparados con los procesadores de propósito general para aplicaciones de procesamiento digital de señales.

Por otra parte los FPGAs son una opción atractiva para el desarrollo de aplicaciones de procesamiento digital, debido a que disponen de una arquitectura reconfigurable que permite una gran capacidad de paralelismo y flexibilidad [2].

Los FPGAs incorporan más elementos con una amplia gama de soluciones de conectividad, haciendo posible el diseño de complejos y potentes sistemas en un solo dispositivo. Una de sus múltiples aplicaciones está dirigida al área de Procesamiento Digital de Imágenes, en donde se han desarrollado plataformas que permiten realizar algoritmos sencillos como el cálculo de histograma y negativo de una imagen, hasta algoritmos más complejos como: filtrado (remoción de ruido) y segmentación, obteniendo resultados óptimos.

## FUNDAMENTOS TEÓRICOS

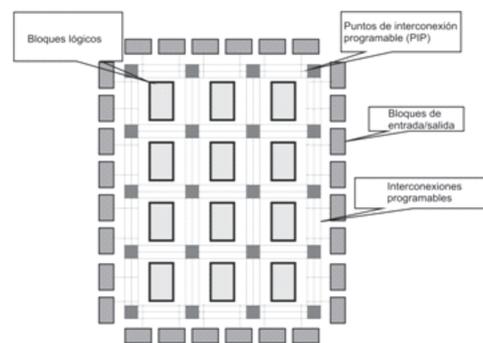
Los FPGAs son circuitos integrados que contienen bloques de lógica configurable (CLB), bloques de entrada y salida (IOB) y canales de comunicación cuya interconexión y funcionalidad se puede programar.

Un FPGA se compone de elementos con recursos no comprometidos que pueden ser seleccionados, configurados e

interconectados por el usuario. Estos dispositivos se componen de cierto número de módulos lógicos, que determinan la capacidad del dispositivo. Los módulos son independientes entre sí y pueden interconectarse para formar un módulo más complejo.

## Arquitectura del FPGA

Existe una gran variedad de FPGAs provistos por varias compañías como Xilinx, Altera, Atmel y Lattice. Cada fabricante provee a su FPGA con una arquitectura única. Un FPGA típico está formado por bloques lógicos configurables, bloques configurables de entrada/salida e interconexiones programables como se muestra en la Figura 1:



*Figura 1. Diagrama a bloques de la arquitectura de un FPGA*

- Bloques lógicos configurables. Los Bloques Lógicos Configurables (CLBs, Configurable Logic Blocks) son recursos lógicos que permiten al usuario realizar diferentes funciones; los CLBs están distribuidos en forma matricial en el dispositivo. En el caso donde estos recursos son de complejidad baja, es decir, las funciones lógicas que se pueden realizar en ellos son sencillas, y existe un número considerable de ellos, se dice que el FPGA es de granularidad fina. Cuando los recursos lógicos están formados por memorias de acceso aleatorio llamadas Tablas de Búsqueda (LUT, Look-up Tables), flip-flops para almacenamiento de elementos que dependen de la señal de reloj, multiplexores que permiten la selección, reset y puesta a uno lógico de elementos, se dice que el FPGA es de granularidad gruesa; en un FPGA de este tipo el número de

CLBs que lo integran es reducido pero tienen la característica de poder implementar funciones de mayor complejidad.

- Bloques configurables de entrada/salida. La matriz de bloques de CLBs está rodeada por un anillo de bloques de interfaz, denominados bloques configurables de entrada/salida. Estos bloques están dedicados a proporcionar la interconectividad entre el FPGA y el exterior, es decir, controlan la entrada y salida de datos entre los pines de entrada/salida y la lógica interna.
- Interconexiones programables. Están formados por: recursos de interconexión, conjunto de líneas y/o interruptores programables que permiten transmitir las señales entre los bloques lógicos internos y entre éstos y los bloques de entrada/salida, y de matriz de interconexión, elementos lógicos que facilitan la comunicación entre los buses de comunicación (recursos de interconexión).
- Circuitería de reloj. Existe un tercer tipo de recursos exclusivos de conexión: las líneas dedicadas a la transmisión de las señales de reloj. Estas señales generalmente están conectadas a un gran número de bloques por lo que son diseñadas para obtener tiempos de propagación pequeños y similares. Además, están distribuidas alrededor del FPGA mediante buffers de reloj especiales, conocidos como drivers de reloj. Estos buffers se encuentran conectados al reloj principal y lo llevan hacia todas las líneas de reloj global con el fin de que puedan ser utilizadas por cada CLB [3].

## ETAPAS DEL DISEÑO EN FPGA

Para llevar a cabo la programación del FPGA es necesario utilizar algún lenguaje de descripción de Hardware como VHDL o Verilog que consiste en una serie de instrucciones y directivas parecidas a los lenguajes de programación de alto nivel. A diferencia de estos lenguajes, los lenguajes de descripción de hardware no se usan para implementar un algoritmo en una computadora. Como su nombre lo indica, se utilizan para describir componentes que forman parte de un circuito y su interconexión. La estructura general de un programa en VHDL está formada por módulos o unidades de diseño, cada uno de ellos compuesto por un conjunto de declaraciones e instrucciones que definen, describen, estruc-

turan, analizan y evalúan el comportamiento de un sistema, como se muestra en la Figura 2.

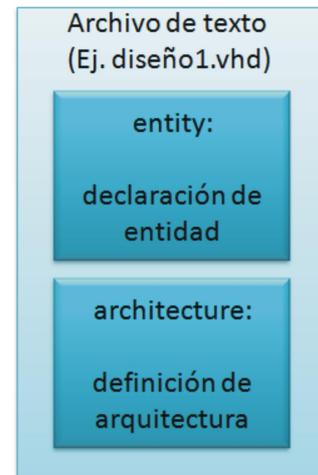


Figura 2. Estructura de un programa en VHDL.

De este modo, una entidad VHDL es simplemente una declaración de las entradas y salidas de un módulo, mientras que una arquitectura VHDL es una descripción detallada del comportamiento o estructura interna del módulo [4]. Otras de las etapas que comprende el diseño de un sistema en un FPGA, son las siguientes:

- Simulación del diseño. Una vez terminado la descripción de VHDL, es importante saber si el diseño así generado realizará la función deseada. Esto se puede lograr con la simulación del diseño, la cual además es útil para verificar la sintaxis del programa. Hay que tomar en cuenta que en este punto hace falta considerar algunos factores, como los retardos de tiempo inherentes de los componentes, por lo que esta simulación es sólo una primera aproximación.
- Síntesis del diseño. En esta parte se crea una descripción a nivel de compuertas del diseño obtenido. El resultado de esta etapa es la obtención de una lista con las compuertas necesarias para implementar un diseño, además de todas las conexiones necesarias entre las mismas.
- Mapeo. El mapeo usa la descripción de compuertas generadas por la herramienta de síntesis, distribuyéndolas entre los diferentes recursos con los que cuenta el FPGA, como son los CLB's, bloques de memoria, multiplicadores y otros.

- Colocación y ruteo. En el mapeo fueron definidos los recursos del FPGA que se deben usar para implementar el diseño. Con la colocación se seleccionan los componentes específicos a ser usados y con el ruteo se definen las conexiones entre los diferentes componentes seleccionados.
- Programación. Después de haber cubierto todas las etapas anteriores, en este momento es posible obtener el archivo necesario para programar el FPGA, conocido como bitstream. La programación consiste en enviar este archivo al FPGA para que establezca las conexiones apropiadas entre los diferentes componentes del mismo [3].

donde  $r$  y  $s$  corresponden a los niveles de gris asociados a los pixeles de entrada y salida respectivamente.

La Figura 3 muestra la función de transformación empleada para la obtención del negativo de la imagen.

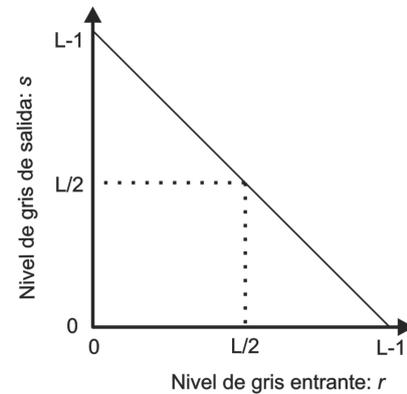


Figura 3. Función de transformación del negativo de una imagen

## TÉCNICAS DE PROCESAMIENTO DIGITAL DE IMÁGENES

Existe una gran variedad de técnicas en el área de Procesamiento Digital de Imágenes, sin embargo algunas de ellas, aunque simples sirven de base para el desarrollo de algoritmos más complejos. El histograma, por ejemplo, aporta información importante acerca de la calidad o contraste que presenta la imagen. El negativo, por otra parte, resulta útil cuando se quieren resaltar detalles blancos o grises que se encuentran en regiones oscuras de la imagen [3].

### Histograma

El histograma de una imagen con niveles de gris en el rango  $[0; L - 1]$ , es una función discreta que representa el número de pixeles que posee cada nivel de gris en la imagen, o dicho en otras palabras, es la frecuencia relativa de ocurrencia de cada nivel de gris en la imagen.

$$h(r_k) = n_k \quad (1)$$

donde  $r_k$  es el  $k$ -ésimo nivel de gris y  $n_k$  es el número de pixeles en la imagen que contienen dicho nivel.

### Negativo de una imagen

El negativo de una imagen con niveles de gris en el rango  $[0; L - 1]$ , es obtenido al restar cada uno de sus pixeles de entrada del valor mínimo de nivel de gris  $(L - 1)$  permitido en la imagen, esto es:

$$s = (L - 1) - r \quad (2)$$

## SÍNTESIS DE MÓDULOS DE PROCESAMIENTO EN FPGA MEDIANTE VHDL

En esta sección se realiza la descripción tanto del módulo para el negativo como para el histograma de una imagen.

### Módulo: Negativo de una imagen

En la Figura 4 se muestra el diagrama a bloques de la función de transformación utilizada para obtener el negativo de la imagen.

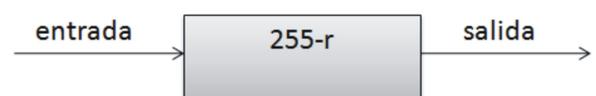


Figura 4. Diagrama a bloques para la obtención del negativo.

A partir del modelo anterior, se define el diseño funcional de este bloque, en el que se describe la relación entrada-salida.

A continuación se muestra el listado correspondiente a esta descripción, utilizando el lenguaje VHDL:

```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity negativo is
  Port ( clk : in  STD_LOGIC;
        rst: in  STD_LOGIC;
        Dato: in  STD_LOGIC_VECTOR(7 downto 0);
        Neg: out STD_LOGIC_VECTOR(7 downto 0));
end negativo;

architecture Behavioral of negativo is
begin

negativ: process (clk,rst,Dato)
begin
  if clk'event and clk='1' then
    if rst='1' then
      Neg <= (others => '0');
    else
      --Neg <= 255 - Dato;
      Neg <= not Dato;
    end if;
  end if;
end process;

end Behavioral;
%end{verbatim}

```

En la entidad se definen las terminales de entrada y salida del módulo, así como el tipo de línea usada (simple o bus). En la parte correspondiente a la descripción funcional (process), se realiza el cálculo del negativo, de acuerdo a la función de transformación dada en (2). Este proceso se realiza en forma síncrona, lo cual es controlado por el reloj (clk) del sistema.

Finalmente, como resultado de la síntesis del diseño anterior se obtiene el circuito mostrado en la Figura 5.

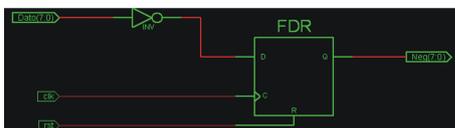


Figura 5. Síntesis para el módulo del negativo.

En esta figura se observa que la síntesis de esta función de transformación genera básicamente una compuerta inversora, lo cual representa una estructura bastante simple para la implementación del negativo.

## Módulo: Histograma de una imagen

Enseguida se muestra la estructura propuesta para calcular el histograma de una imagen dentro del FPGA (ver, Figura 6). Básicamente consiste en una memoria RAM que permite registrar la cantidad de pixeles de cada nivel de gris. Para esto, utiliza un sumador que incrementa en uno la localidad de memoria asociada a cada nivel de gris. En este caso el valor del pixel sirve para apuntar a la dirección de memoria a incrementar [5]. Cuando se han terminado de procesar todos los puntos de la imagen, la memoria contiene el total de pixeles en cada nivel de gris.

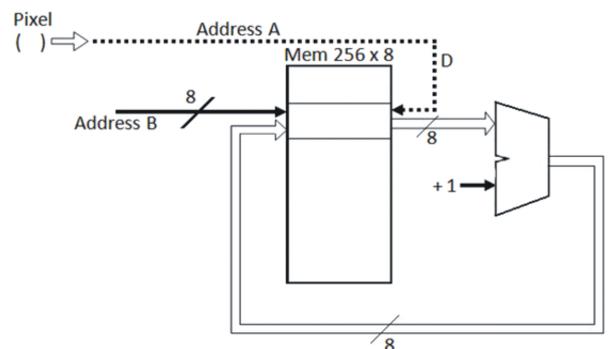


Figura 6. Diagrama a bloques para el cálculo del histograma.

## PRUEBAS Y RESULTADOS

Para evaluar el desempeño de los algoritmos realizados, se inyecta una imagen de prueba al módulo FPGA. Dicha imagen es transferida desde la computadora hacia el FPGA, a través del puerto serie. La información que resulta del procesamiento es reenviada a la computadora para su despliegue.

A continuación se presentan los resultados experimentales obtenidos al implementar en el FPGA los algoritmos descritos anteriormente.

### Prueba 1: Negativo de la imagen

La Figura 7 corresponde al resultado que se obtiene al aplicar el algoritmo de negativo a la imagen de prueba.



Figura 7. Imagen original y negativo.

## Prueba 2: Histograma de la imagen

La Figura 8 presenta la imagen original y el histograma que genera el FPGA. Para este ejemplo, el nivel de gris mínimo corresponde al valor de 81, lo cual equivale aproximadamente a una tercera parte del intervalo de tonos de gris posibles (generalmente  $L=255$  niveles).

Con esta información que arroja el histograma es posible cuantificar el nivel de contraste que presenta la imagen de prueba, ya que existe una relación directa con el intervalo dinámico de tonos de gris que contiene la imagen.

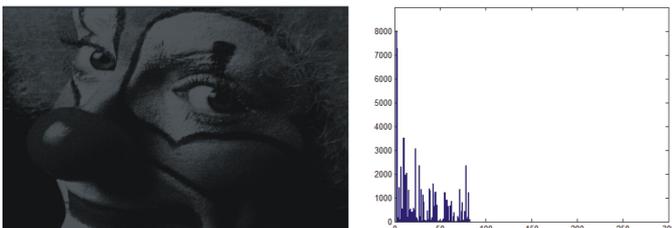


Figura 8. Imagen original e histograma.

## CONCLUSIONES

Se pueden realizar técnicas de Procesamiento Digital de Imágenes por medio de arquitecturas reconfigurables (FPGA) obteniendo resultados satisfactorios. Una de las principales ventajas de este tipo de tecnología es que se trata de dispositivos reconfigurables, lo cual permite un diseño acorde a las necesidades específicas de cada aplicación.

El diseño y síntesis de algoritmos de Procesamiento Digital de Imágenes en un FPGA, es una tarea que requiere no sólo del completo conocimiento de las técnicas, sino también de una vasta experiencia en el diseño de sistemas digitales, ya que esto permitir plantear estructuras simples, que realicen

su función de manera óptima y con el mínimo tiempo de procesamiento posible, lo cual permitirá su eventual aplicación en tareas de procesamiento en tiempo real.

## Bibliografía

- [1] Alexander Quintero M. y Eric Vallejo R., “Algoritmos de procesamiento de imágenes en FPGA”, Revista Colombiana de Tecnologías de Avanzada, pp.11-12, 2006.
- [2] Ferney O. Amaya., “Aplicaciones para telecomunicaciones empleando FPGAs: Una aplicación a radio software”, Revista Colombiana de Tecnologías de Avanzada, pp. 39, 2006.
- [3] Manuel A. Mendoza., “Procesamiento y análisis digital de imágenes mediante dispositivos lógicos programables”, pp.16-40, 2009.
- [4] David G. Maxinez y Jessica Alcal , VHDL, El arte de programar sistemas digitales, Patria, Pub. Co.,2008, México.
- [5] Edgard Garcia., “Implementing a Histogram for Image Processing Applications”, Revista Xcelljournal, pp. 46, 2000.

## Acerca del autor o autores

Diana M. Córdova es estudiante de la Maestría en Ingeniería con orientación en Procesamiento Digital de Señales, Ernesto García es Profesor-investigador de la Fac. de Ingeniería Eléctrica de la Universidad Autónoma de Zacatecas, y miembro del Cuerpo Académico de Procesamiento Digital de Señales. Arturo Moreno es estudiante de doctorado en la Facultad de Ingeniería Eléctrica, todos ellos en la Universidad Autónoma de Zacatecas.