

DISEÑO DE UN SISTEMA DE TRANSFERENCIA Y PROCESAMIENTO DE IMÁGENES SOBRE UN FPGA

*REINALDO JAVIER MENÉNDEZ ALONSO¹, BÁRBARO MAYKEL LÓPEZ-PORTILLA VIGIL¹,
MIGUEL ENRIQUE IGLESIAS MARTÍNEZ²*

¹Universidad de Pinar del Río. Departamento de Telecomunicaciones y Electrónica.

e-mail: rey88@tele.upr.edu.cu, barbaro@tele.upr.edu.cu

²Centro de Desarrollo de la Electrónica y la Automática. e-mail: mgi@cdea.co.cu

Recibido: octubre 2012

Recibido en forma final revisado: enero 2013

RESUMEN

En el presente trabajo se describe el diseño de un sistema de transferencia y procesamiento de imágenes entre un ordenador personal y un kit de desarrollo Nexys2 de la compañía Digilent. Para su realización se analizan los referentes teóricos relacionados con el procesamiento de imágenes y con el kit mencionado, de manera que se puedan implementar los diseños necesarios para lograr la aplicación deseada. La investigación realizada propone el uso de un software residente en el ordenador para la transferencia de las imágenes y la toma de decisiones sobre las mismas, mientras que los módulos implementados mediante el lenguaje VHDL en el FPGA (embebida en el kit Nexys2) realizan el procesamiento. El desarrollo de esta investigación permite obtener una plataforma eficiente, flexible y de bajo costo para el procesamiento de imágenes. El resultado obtenido en esta investigación es de gran importancia desde el punto de vista didáctico, pues permite mostrar las ventajas de un dispositivo FPGA y puede motivar a que se desarrollen nuevas arquitecturas de procesamiento de imágenes, sin tener que volver a construir la interfaz de comunicación entre el ordenador y el kit de desarrollo, que puede ser laborioso.

Palabras clave: Transferencia, Procesamiento, Imágenes, Ordenador, FPGA, VHDL.

DESIGN OF A TRANSFER SYSTEM AND IMAGE PROCESSING ON A FPGA

ABSTRACT

This paper describes the design of a transfer and image processing system between a PC and a Digilent Nexys2 development kit. To this aim, we analyze the theoretical framework and the aforementioned kit, so that it is possible to implement the designs needed to achieve the desired application. The research proposes the use of computer resident software for transferring images and making decisions on them, while the modules VHDL implemented in the FPGA (embedded in Nexys2 kit) perform the processing. The development of this research allows for an efficient and flexible platform, along with a low-cost image processing. The result obtained in this research is of great importance from the educational point of view since it can show all of the advantages of an FPGA and can encourage the development of new image processing architectures without having to build the interface communication between the computer and the development kit, a fact that could be laborious.

Keywords: Transferring, Processing, Images, Computer, FPGA, VHDL.

INTRODUCCIÓN

El procesamiento digital de imágenes es el conjunto de técnicas que se aplican sobre una imagen digital con el objetivo de mejorar su calidad. Esta es una rama de la ingeniería muy expandida y dinámica, con aplicaciones en áreas cercanas a nuestra vida diaria como la medicina, la exploración espacial, la vigilancia, la autenticación, la inspección industrial automatizada, entre otras.

Implementar dichas aplicaciones en una computadora de propósito general resulta relativamente fácil, pero no siempre el resultado es eficiente; su rendimiento depende de la potencia de cálculo del procesador empleado, así como de la cantidad de aplicaciones que se encuentran ejecutando al mismo tiempo. Como esta tarea requiere un poder de cómputo intensivo, se necesitan equipos de alto rendimiento para agilizar dicho proceso. En este sentido, las implementaciones hardware para aplicaciones específicas

ofrecen mayor velocidad que las implementaciones software. Con este propósito se han incorporado a los sistemas de procesamiento dispositivos como los CPLD (Complex Programmable Logic Device), los DSP (Digital Signal Processor) y los FPGA (Field Programmable Gate Array) (Daggu *et al.* 2006).

Estos últimos son circuitos integrados de alta densidad que combinan la flexibilidad y el alto nivel de integración de los anteriores con un proceso de diseño simple y asequible. Contienen una matriz de elementos lógicos que pueden ser interconectados para implementar una aplicación dada. Estas interconexiones son controladas por interruptores programables por el usuario. Así, los prototipos pueden ser implementados, probados y modificados rápidamente (Bravo *et al.* 2004).

Por todos estos motivos, la enseñanza de los conceptos relacionados con los dispositivos FPGAs es de vital importancia en carreras de perfil electrónico. Las plataformas de comprobación basadas en FPGA han sido utilizadas con éxito en diversas aplicaciones. De entre todas ellas se ha elegido el procesamiento de imágenes, ya que este tipo de aplicaciones suelen ser las más gratificantes para el estudiante. Este obtiene una recompensa netamente visual del trabajo que va desarrollando, comprobando rápidamente su correcto funcionamiento. Más aún, las operaciones de procesamiento de imágenes por lo general suelen ser fáciles de comprender, sin requerir modelos teóricos complejos, lo cual permite enfocar la docencia al uso de los FPGAs (Vega *et al.* 2000).

Aprovechando las bondades mencionadas de los FPGA, se implementa un sistema de transferencia y procesamiento de imágenes entre un ordenador personal y un kit de desarrollo Nexys2 de la compañía Digilent, el cual permite aplicar sobre las imágenes diversas transformaciones en el dominio espacial.

TÉCNICAS EXPERIMENTALES

• Etapas básicas del sistema diseñado

Como se muestra en la Figura 1, los sistemas de procesamiento de imágenes, de forma general, constan de tres etapas bien definidas: etapa de adquisición, etapa de procesamiento y etapa de salida. La etapa de adquisición se encarga de transferir hacia la etapa de procesamiento la imagen que llega al sistema, aplicando cualquier operación sobre ella de manera que sea comprensible para un sistema digital. La etapa de procesamiento es la encargada de realizar la mejora de la imagen con el fin de producir

representaciones más útiles en otros mecanismos del sistema. Por último, la etapa de salida visualiza el resultado del procesamiento o bien utiliza la información para la toma de decisión de manera automática (Sosa, 2007).

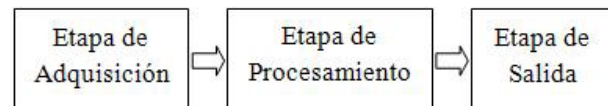


Figura 1. Etapas básicas de un sistema de procesamiento de imágenes

En el caso de esta investigación, la etapa de adquisición y de salida se funden en una sola (Etapa de Adquisición/Visualización), empleándose para ello una PC (Personal Computer); mientras que la etapa de procesamiento está conformada por el kit de desarrollo Nexys2. En la Figura 2 se muestra el esquema general del sistema diseñado.

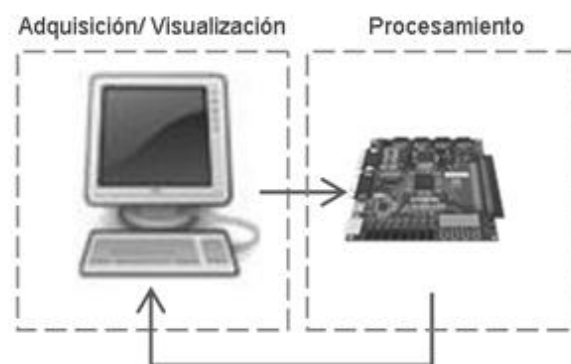


Figura 2. Esquema general del sistema diseñado

• Etapa de Adquisición/Visualización

De forma general el sistema permite importar desde un ordenador una imagen en formato BMP o JPG con resolución de 256x256 píxeles y aplicar sobre ella varias transformaciones en el dominio espacial: negativo, conversión a imagen binaria, ajustes de brillo y contraste, filtrado de mediana, filtrado de suavizado y filtrado de realce. La computadora además se utiliza para visualizar los resultados del procesamiento. Es por ello que se desarrolla una aplicación en alto nivel que responde a estas necesidades.

Visual Studio 2010 Ultimate, de la compañía Microsoft, fue el entorno escogido para el desarrollo del software que sirve como interfaz gráfica del sistema. Este es un ambiente de desarrollo integrado para Sistemas Operativos Windows que soporta varios lenguajes de programación como Visual J#, ASP.NET, Visual Basic .NET, Visual C# y Visual C++.

Usando el lenguaje Visual C++ y el paradigma Programación Orientada a Objetos, se desarrolla una aplicación con una interfaz gráfica muy amigable que presenta una barra de herramientas con el acceso a las siguientes funcionalidades: abrir una imagen, transferirla a la etapa de procesamiento, seleccionar la técnica que será aplicada, leer la imagen procesada, guardarla en la computadora y ver el histograma tanto de la imagen original como de la resultante. Además es dividida en dos paneles: en el izquierdo se muestra la imagen abierta por el usuario, mientras que en el derecho se muestra el resultado de las técnicas aplicadas sobre ella; de modo que se puedan comparar ambas imágenes.

De esta forma los estudiantes observan en un entorno sencillo y totalmente gráfico el resultado de sus diseños sobre FPGA. Además de que una vez creado este entorno, se pueden concentrar más en la etapa de procesamiento, o sea, en la implementación de las técnicas de procesamiento de imágenes sobre FPGA.

En la Figura 3 se muestra la ventana principal de la aplicación una vez que se adquiere una imagen y se le aplica el proceso de conversión a imagen binaria con un umbral seleccionado de forma manual (se han implementado además dos métodos para la selección automática del umbral: el Método de Otsu y el de Gonzalez-Woods).



Figura 3. Visualización del procesamiento

El histograma no contiene información espacial acerca de la imagen; solamente describe la frecuencia relativa de ocurrencia de cada nivel de gris (González & Woods, 2002). Sin embargo esta información es muy valiosa. La inspección visual del histograma permite averiguar dónde están distribuidos fundamentalmente los niveles de gris: en la parte baja, en la parte alta, o por todo el rango dinámico de la imagen. Por eso se incluye esta funcionalidad a la aplicación. De esta forma se puede analizar el histograma tanto de la imagen original como de la procesada.

En la Figura 4 se muestran los histogramas correspondientes a una imagen con bajo contraste (izquierda) y a una imagen a la cual se le aplicó mejora de contraste mediante expansión del histograma usando el sistema diseñado (derecha). De esta forma se puede comprobar la efectividad de las técnicas de procesamiento implementadas.

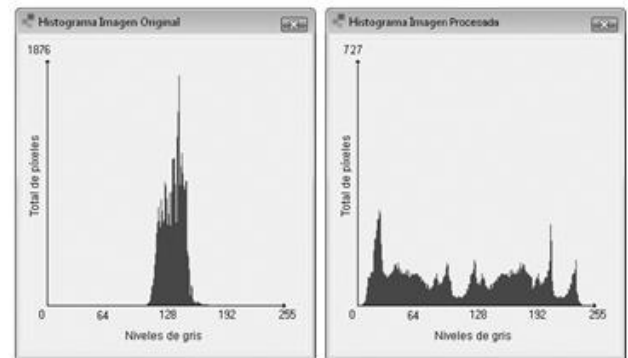


Figura 4. Visualización del histograma

La compañía Digilent ofrece de manera gratuita en su sitio oficial <http://www.digilentinc.com> una DLL (Dynamic Link Library) denominada dpcutil.dll la cual provee una API (Application Programming Interface) que permite a los kits de esta compañía comunicarse con aplicaciones ejecutadas desde una computadora vía Ethernet, USB 2.0 o Puerto Serie (Digilent, 2009). Usando esta API y una correcta lógica en el kit, pueden ser enviados/recibidos datos de ocho bits desde/hacia la computadora.

Para el desarrollo de la aplicación se reutilizaron las funcionalidades implementadas en dpcutil.dll que permiten detectar el kit Nexys2, crear una interfaz de acceso para la transferencia de datos y enviar/recibir registros de ocho bits usando el puerto USB 2.0. De esta forma tiene lugar el intercambio de datos entre la etapa de procesamiento implementada en el FPGA y la aplicación en el ordenador.

- **Etapas de procesamiento**

La función de esta etapa es realizar el procesamiento seleccionado por el usuario desde el ordenador sobre la imagen original. Dicha imagen se almacena en la SRAM (Static Random Access Memory) externa presente en el kit, y los módulos implementados en el FPGA XC3S500E acceden a sus píxeles para transformarlos. Estos módulos se han modelado usando el lenguaje de descripción de hardware VHDL (Very High Speed Integrated Circuit Hardware Description Language), mediante el software ISE v12.1 de la compañía Xilinx. Los píxeles resultantes se guardan en direcciones distintas a los originales para asegurar que ambas imágenes (original y procesada) se encuentren almacenadas en la misma memoria.

Este último software, a pesar de ser utilizado con fines investigativos, desde el punto de vista docente le permite al alumno realizar el ciclo de diseño completo de una operación de procesado de imágenes, llegando a la implementación físicamente en el FPGA. Además, este software es utilizado actualmente por las empresas para sus desarrollos dentro de sus ámbitos de actuación particulares, con lo que se le ofrece al estudiante una formación que pueda ser demandada en el mercado laboral (Vega *et al.* 2000).

En la Figura 5 se muestran los bloques funcionales de la etapa de procesamiento. La misma está formada (de derecha a izquierda) por el controlador de la memoria, la interfaz EPP (Enhanced Parallel Port) para la comunicación USB entre el ordenador y el FPGA, el módulo de multiplexación de acceso para gestionar la lectura y escritura en la memoria y, por último, el módulo que contiene las distintas técnicas de procesamiento sobre la imagen.

Controlador de la memoria

El módulo controlador de la memoria genera la secuencia de señales necesarias para leer o escribir en la SRAM externa. Además en él se implementan siete registros de datos, mediante los cuales se configura la memoria para una lectura o una escritura, se establece la dirección para dichas operaciones y se indica el tipo de transferencia a realizar. En la Tabla 1 se muestran los registros mencionados y una breve descripción de los mismos.

En el sistema acceden a estos registros el módulo de procesamiento y la aplicación en alto nivel mediante los

ciclos de lectura/escritura de datos y direcciones del Protocolo EPP. Por ejemplo, si se deseara guardar desde la aplicación una imagen en la SRAM, primero se accede al Registro de Control para configurar una escritura, luego se conforma la dirección mediante los Registros 1, 2 y 3, y posteriormente se accede al Registro 6 para escribir un bloque de datos de ocho bits correspondientes a los píxeles de la imagen.

Tabla 1. Registros de datos implementados en el controlador de la memoria

Registro	Descripción
0	Registro de control
1	Bits 0-7 de la dirección en la SRAM
2	Bits 8-15 de la dirección en la SRAM
3	Bits 16-23 de la dirección en la SRAM
4	Escritura de un solo dato
5	Lectura de un solo dato
6	Lectura/Escritura de bloques de datos.

Interfaz EPP

Las funciones de transferencia de datos implementadas en dpcutil.dll requieren la presencia en el FPGA de un módulo que sirva como interfaz de puerto paralelo. Para ello Digilent proporciona el módulo EppInterface así como la documentación para su uso.

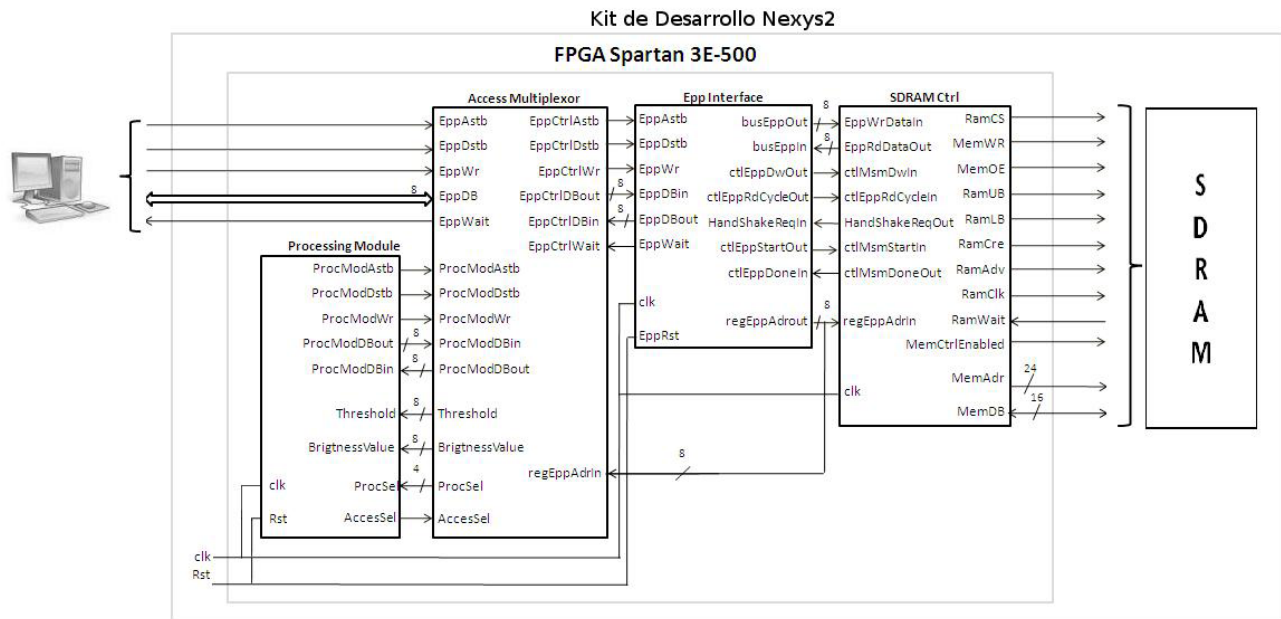


Figura 5. Diagrama en bloques de la etapa de procesamiento

En la Figura 6 se puede apreciar este módulo. El mismo recibe las indicaciones (desde el módulo de procesamiento o desde la aplicación en el ordenador) para controlar los registros mostrados en la Tabla 1 mediante sus entradas EppAstb, EppDstb, EppWr y EppDBin. Por su parte la salida regEppAdrOut indica cuál es el registro de datos del controlador de la memoria al que se está accediendo.

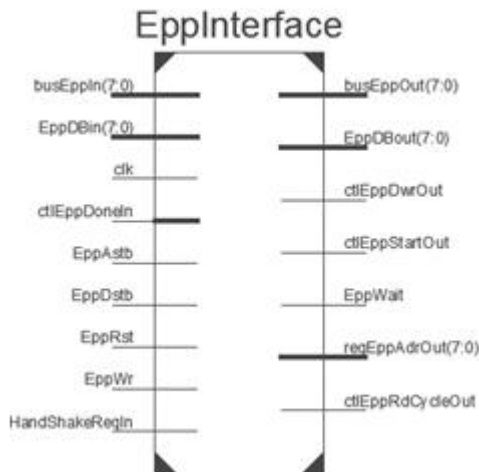


Figura 6. Interfaz EPP para la comunicación USB

El protocolo EPP fue desarrollado como una manera de obtener un puerto paralelo de alta resolución totalmente compatible con el puerto paralelo estándar. El trabajo con los registros mostrados en la Tabla 1 se basa en este protocolo y en sus ciclos de escritura/lectura de datos o direcciones. En la Figura 7 se muestra el comportamiento de las entradas y salidas del módulo EppInterface involucradas en este protocolo para un ciclo de escritura de datos y un ciclo de escritura de direcciones (en los ciclos de lectura el único cambio es que la señal EppWr es '1').

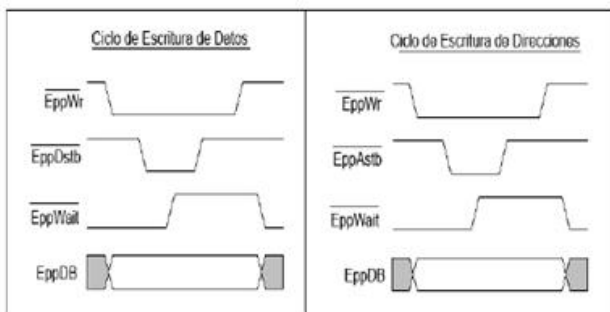


Figura 7. Ciclos de escritura de datos y direcciones del protocolo EPP

Multiplexor de acceso

Los módulos explicados anteriormente se encargan de gestionar la lectura y escritura de datos en la memoria mediante el uso de registros de ocho bits y las señales Astb, Dstb y Wr del protocolo EPP. En el sistema diseñado se

tiene acceso a la memoria desde la aplicación residente en el ordenador y el módulo en el FPGA que realiza el procesamiento, por lo tanto, es necesario gestionar dicho acceso. El módulo encargado de esta función es AccessMultiplexor, el cual se muestra en la Figura 8.

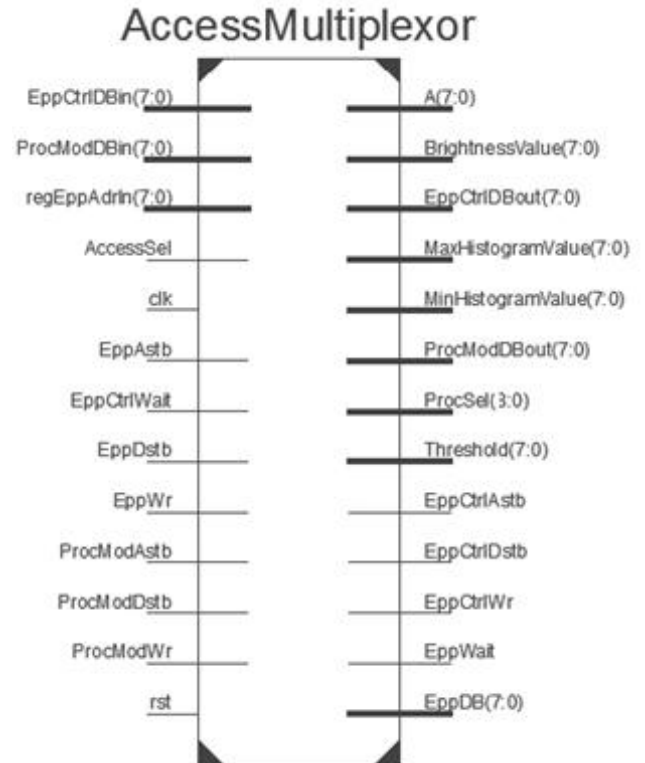


Figura 8. Módulo de multiplexación de acceso

La función principal de este módulo es multiplexar hacia la interfaz EPP las señales provenientes tanto del ordenador como del módulo de procesamiento. Inicialmente este módulo le permite el acceso al primero para que el usuario pueda guardar una imagen en la memoria y seleccionar cualquiera de las técnicas de procesamiento. Luego, AccessMultiplexor permite el acceso del módulo de procesamiento a la memoria para leer y escribir en ella, y así procesar la imagen. Cuando haya finalizado el procesamiento, este módulo devuelve el acceso al usuario para que pueda leer el resultado final.

En este módulo se ha implementado un registro (Registro 8) al que se puede acceder desde la aplicación en el ordenador y escribir en él según el procesamiento que se quiera realizar. En la Tabla 2 se muestran las técnicas de procesamiento aplicadas sobre la imagen según el valor del registro mencionado.

Existen además señales asociadas con cada uno de los valores mostrados en la Tabla 2, las cuales cumplen la función de habilitar las distintas técnicas de procesamiento. La salida

ProcSel es resultado de la combinación de dichas señales y se conecta directamente al módulo de procesamiento. Cuando el módulo AccessMultiplexor detecta el acceso al registro de procesamiento desde el ordenador, configura la salida ProcSel activando en el módulo de procesamiento la técnica seleccionada por el usuario.

Tabla 2. Técnicas aplicadas sobre la imagen según el valor del registro de procesamiento

Valor	Técnica aplicada
0	Negativo
1	Conversión a imagen binaria
2	Función cúbica
3	Expansión del histograma
4	Aumento de brillo
5	Reducción de brillo
6	Filtro de mediana
7	Filtro de suavizado
9	Filtro de realce

Este multiplexor también se encarga de suministrarle al módulo de procesamiento algunos parámetros necesarios provenientes desde el ordenador como el umbral seleccionado para la conversión a imagen binaria, la medida en la que se aumentará o disminuirá el brillo, los límites del histograma para la manipulación del contraste y el elemento central de la máscara difusa en el filtrado de realce.

Módulo de procesamiento

El módulo encargado de realizar el procesamiento es ProcessingModule. Una vez que AccessMultiplexor le permite el acceso a la memoria, éste se encarga de leer uno o varios píxeles, realizar la transformación correspondiente, guardar el resultado en la memoria e indicarle al multiplexor de acceso que puede permitir que el usuario lea desde el ordenador el resultado de la técnica aplicada.

En la Figura 9 se muestra una vista general del mismo. Sus salidas ProcModAstb, ProcModDstb, ProcModWr y ProcModDBout se conectan a las entradas correspondientes en el multiplexor de acceso para desarrollar las operaciones de lectura y escritura de datos y direcciones según corresponda. Su entrada ProcModDBin almacena tanto el valor del píxel leído desde la memoria como los datos de los registros provenientes de EppInterface.

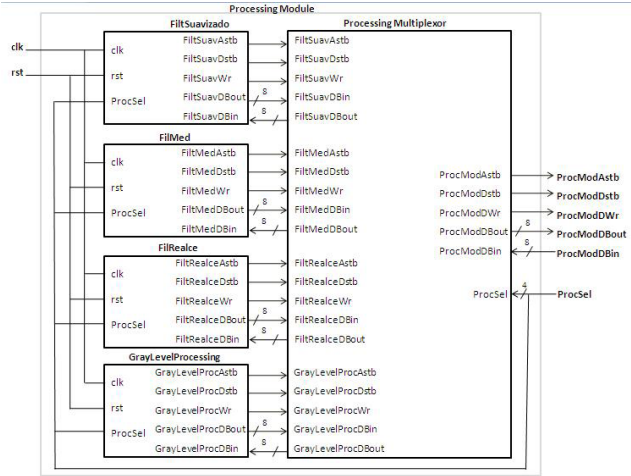


Figura 9. Módulo de procesamiento

En realidad, el módulo de procesamiento está formado por varios sub-módulos que se corresponden con las técnicas implementadas en el sistema (transformación de la escala de grises, filtro de mediana, filtro de suavizado y filtro de realce) y un módulo de multiplexación; con el objetivo de facilitar la comprensión del diseño y la futura adición de nuevas técnicas de procesamiento. Su implementación sigue la filosofía del uso de registros de datos y la ejecución de los ciclos de lectura y escritura del protocolo EPP. Su habilitación depende directamente del valor que tenga el módulo de procesamiento (Top Module) en su entrada ProcSel, que está en correspondencia con el valor del registro de procesamiento.

RESULTADOS Y DISCUSIÓN

El desarrollo de esta investigación arrojó como resultado la implementación de un sistema de transferencia y procesamiento de imágenes utilizando una PC y el kit de desarrollo Nexys2. Este sistema permite la aplicación de técnicas de procesamiento en el dominio espacial a imágenes con resolución de 256x256 píxeles en formato BMP o JPG; utilizando como medio de comunicación el puerto USB 2.0.

El negativo resulta útil cuando se quiere resaltar detalles blancos o grises que se encuentran en regiones oscuras de una imagen, especialmente cuando las áreas negras son dominantes en tamaño. En la Figura 10 se muestra el resultado de esta técnica aplicada sobre una mamografía que presenta una lesión. Nótese que es más fácil estudiar el tejido del seno en la imagen procesada.

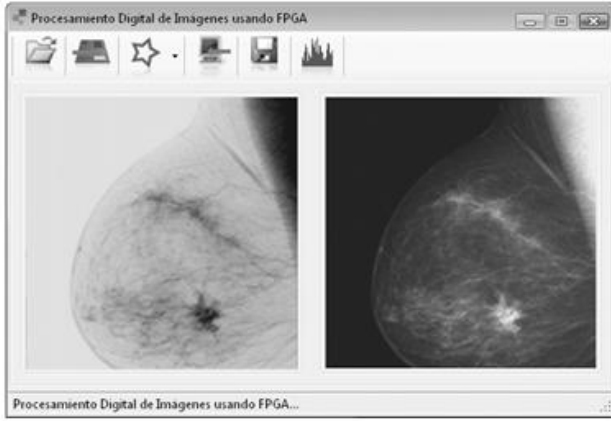


Figura 10. Negativo de una imagen

El brillo de una imagen está dado por el valor de intensidad luminosa que presenten los píxeles. Cuando este nivel no está dentro de los parámetros correctos, la visualización de la imagen se dificulta. Los ajustes de brillo implementados en el sistema permiten aumentarlo o reducirlo de forma manual. En la Figura 11 se muestra el controlador que integra ambas funciones (aumentar hacia la derecha, reducir hacia la izquierda) y el efecto de aumentar el brillo en un factor de 70.

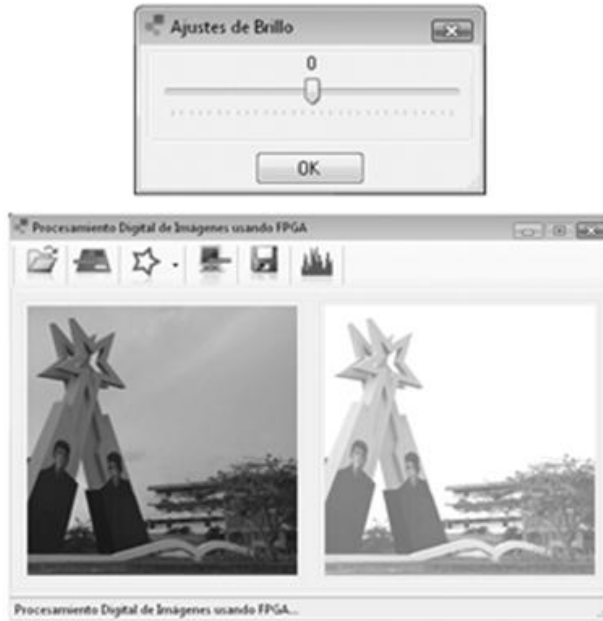


Figura 11. Ajustes de brillo

La facilidad con que el ojo humano puede diferenciar varios objetos o áreas en una imagen se entiende como contraste (Mendoza, 2009). En esta investigación se han implementado dos técnicas para mejorarlo: la función cúbica y la expansión del histograma. En la Figura 12 se muestran los efectos de esta última.



Figura 12. Mejoras de contraste mediante la expansión del histograma

Hasta aquí las técnicas de procesamiento empeladas son de tipo puntuales, ya que el resultado del píxel de salida depende únicamente del píxel de entrada. Este tipo de operaciones tienen expresiones analíticas simples y, por tanto, un algoritmo sencillo, de aquí que sea fácil su programación. Sin embargo, cuando se emplean técnicas de tipo locales (o de ventana) como las que veremos a continuación, el análisis matemático es más complejo, pues el valor del píxel de salida depende tanto del valor del píxel de entrada como de los valores de los píxeles que lo rodean. Para la implementación de los filtros (mediana, suavizado y realce) se emplea un algoritmo de convolución, el cual al ser implementado en FPGA permite aumentar la velocidad de ejecución con respecto a las soluciones software. La estructura de arreglos de compuertas y registros en paralelo de estos dispositivos, los hacen una opción viable para explotar el paralelismo de datos de las imágenes (Johnston *et al.* 2006).

En este sentido, el diseño de algoritmos de convolución basados en la ecuación 1 permite explotar el paralelismo del algoritmo, aunque puede ocupar un porcentaje elevado de los bloques de multiplicación dedicados del FPGA. La operación de convolución muestra una total independencia de los píxeles de la ventana en el algoritmo (Garcés *et al.* 2011).

$$A_c = (i-r, j-r) = \sum_{k=i-r}^{i+r} \sum_{l=j-r}^{j+r} A(k,l)C'(k+\alpha, l+\beta) \quad (1)$$

donde: A es una imagen de m×n píxeles, C es la matriz de convolución (máscara) de h×h elementos y radio $r = (h-1)/2$, y A_c la imagen resultante. Además tenemos que: $\alpha = r-i+1$ y $\beta = r-j+1$.

En la Figura 13 se muestra la arquitectura (en su versión genérica) empleada para la realización del algoritmo descrito.

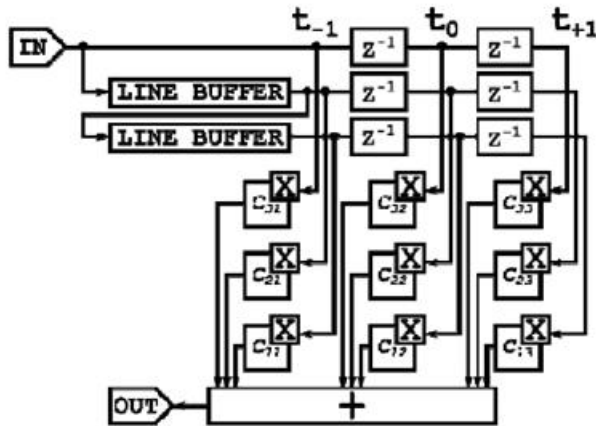


Figura 13. Arquitectura genérica para la implementación de filtros mediante la convolución

Después del almacenamiento de los píxeles en la ventana, los mismos son multiplicados por el respectivo valor de la máscara de convolución rotando, para luego sumar todos los resultados parciales. Si bien es verdad que esta arquitectura emplea un número elevado de multiplicadores, estos se reducen cuando se emplean máscaras que contengan valores como 0, 1 y -1; o cuando éstas sean simétricas.

El filtrado de mediana es un método robusto para eliminar el ruido impulsivo (también llamado ruido de Sal y Pimienta), el cual consiste en la aparición aleatoria de píxeles blancos y píxeles negros en la imagen (Vega *et al.* 2002). En la Figura 14 se puede apreciar cómo se ha filtrado la imagen original para eliminar este ruido. Además se puede comprobar el efecto secundario del mismo que consiste en suavizar los contornos de la imagen.



Figura 14. Reducción de ruido Sal y Pimienta mediante el filtrado de mediana

Los filtros de suavizado tienen el efecto de difuminar los contornos en las imágenes y esto es muy usado para la reducción de ruido. En la Figura 15 se pueden apreciar los coeficientes del filtro usado en esta investigación y como, aunque no se elimina completamente, se reduce bastante el ruido presente en la imagen original.

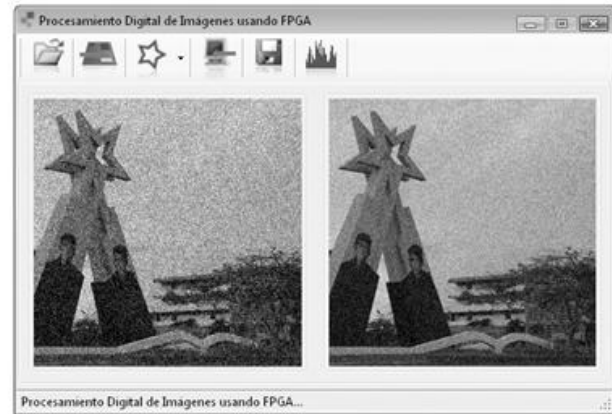
$$\frac{1}{16} \times \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$


Figura 15. Filtro de suavizado usado y su efecto en la reducción de ruido

Por otra parte, los filtros de realce enfatizan los contornos de las imágenes. Estos implementan varios operadores que realizan el realce mediante la diferenciación digital (Bovick, 2000). En esta investigación se ha implementado una variante de estos filtros conocida como máscara difusa y realce en altas frecuencias. En la Figura 16 se muestra la máscara aplicada. Como se puede apreciar esta depende de un factor A, el cual define además la iluminación de la imagen resultante.

-1	-1	-1
-1	A + 8	-1
-1	-1	-1

Figura 16. Máscara difusa.

En la Figura 17 se muestra el efecto que tiene aplicar la máscara difusa con un valor de $A = 2$ sobre una imagen con abundante relieve. Se puede observar que los contornos se han enfatizado y además ha mejorado el contraste debido al valor asignado al factor A .

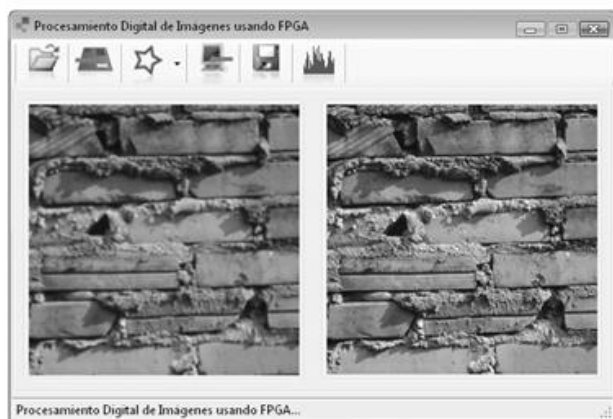


Figura 17. Efecto de la máscara difusa con $A = 2$

En las Tablas 3 y 4 se muestran los recursos ocupados en el FPGA por todos los módulos embebidos.

Tabla 3. Recursos (Slices y Flip-Flops) ocupados en el FPGA

Módulos	Slices	Flip-Flops
SDRAMCtrl	127(2%)	73(0%)
EppInterface	12(0%)	12(0%)
AccessMultiplexor	18(0%)	-
ProcessingModule	1165(30%)	1222(11%)
General	1322(32%)	1307(11%)

Tabla 4. Recursos (LUTs y Multiplicadores) ocupados en el FPGA

Módulos	LUTs	Multiplicad.
SDRAMCtrl	236(2%)	-
EppInterface	24(0%)	-
AccessMultiplexor	31(0%)	-
ProcessingModule	2834(28%)	12(60%)
General	3125(30%)	12(60%)

CONCLUSIONES

Se ha presentado el diseño de una aplicación en alto nivel para la transferencia de las imágenes, la selección del procesamiento y la visualización de los resultados del mismo y, por otra parte, el diseño de módulos VHDL para el procesamiento en el dominio espacial de las imágenes. Al

estar el sistema de procesamiento basado en FPGA, permite una opción de bajo costo que, gracias al paralelismo de esta tecnología, puede integrar una gran cantidad de tareas en el mismo circuito integrado.

Una vez que se posea la interfaz de transferencia entre el ordenador y el kit, el estudiante puede concentrarse más en el diseño arquitectural de las aplicaciones que desea realizar, y de esta manera ellos pueden comprobar experimentalmente, y ampliar, los distintos aspectos teóricos de la electrónica digital, y en particular sobre los dispositivos FPGAs.

El uso de los FPGAs por parte de los estudiantes para el procesamiento de imágenes es una innovación educativa que produce una mejora en la calidad de la enseñanza de la microelectrónica. Además, es un claro ejemplo de cómo los resultados de la investigación desarrollada pueden revertir en la docencia.

REFERENCIAS

- BOVIK, A. (2000). Handbook of Image and Video Processing. Academic Press. 71-81.
- BRAVO, I., RIVERA, R., HERNÁNDEZ, A., MATEOS, R., GARDEL, A., MECA, F. J. (2004). Implementación de filtros FIR en FPGAs. VI Congreso de Tecnologías Aplicadas a la Enseñanza de la Electrónica (TAE), Universidad Politécnica de Valencia, España.
- DAGGU, R., SHURTI, P., NAVEEN, A., MUTHUKUMAR, V. (2006). Implementation and evaluation of image processing algorithms on reconfigurable architecture using C-based Hardware Descriptive Languages. International Journal of Theoretical and Applied Computer Sciences, Vol. 1, 2006, pp. 9-34.
- DIGILENT. (2009). Digilent Port Communications Programmers Reference Manual. Recuperado el 6 de enero de 2012, de <http://digilentinc.com/Data/Products/ADEPT/DPCUTILProgrammersReferenceManual.pdf>.
- GARCÉS, L. M., CABRERA, A. J., SANCHEZ, S., BROX, P. (2011). Diseño de bloques de convolución para procesado de imágenes con FPGA. Revista de Ingeniería Electrónica, Automática y Comunicaciones, Vol. XXXII, pp. 56-69.
- GONZÁLEZ, R. C. & WOODS, R. E. (2002). Digital Image Processing. Prentice Hall, pp. 76-137.

- JOHNSTON, C. T., BAILEY, D. G., LYONS, P. (2006). A Visual Environment for Real-Time Image Processing in Hardware. EURASIP Journal on Embedded Systems, pp. 1-8.
- MENDOZA, M. A. (2009). Procesamiento y análisis digital de imágenes mediante dispositivos lógicos programables. Tesis de Grado no publicada. Universidad Tecnológica de la Mixteca, Oaxaca, México.
- SOSA, J. C. (2007). Sistema de visión basado en procesado guiado por cambios y lógica reconfigurable para el análisis de movimiento de alta velocidad. Tesis de Doctorado no publicada. Universidad de Valencia, Valencia, España.
- VEGA, M. A., SÁNCHEZ, J. M., GÓMEZ, J. A. (2000). Experiencia piloto: Procesamiento de imágenes mediante FPGA con fines educativos. IV Congreso de Tecnologías Aplicadas a la Enseñanza de la Electrónica (TAEE), Universidad Autónoma de Barcelona, Barcelona, España, pp. 599-602.
- VEGA, M. A., SÁNCHEZ, J. M., GÓMEZ, J. A. (2002). An FPGA-based implementation for median filter meeting the real-time requirements of automated visual inspection systems. 10th Mediterranean Conference on Control and Automation, Lisbon, Portugal, pp. 131-136.