Revista Cubana de Ciencias Informáticas
Vol. 10, No. 3, julio-septiembre, 2016
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 16-26

Tipo de artículo: Artículo original
Temática: Matemática computacional
Recibido: 25/03/2016 | Aceptado: 30/06/2016

Implementación del Algoritmo de Otsu sobre FPGA

Otsu's algorithm implementation on FPGA

Bárbaro M. López-Portilla Vigil [1*], Reinaldo J. Menéndez Alonso [1], Miguel E. Iglesias Martínez [2]

[1] Departamento de Telecomunicaciones y Electrónica, Universidad de Pinar del Río. Calle Martí Final No. 270, Pinar del Río, Cuba. {barbaro, rey88} @upr.edu.cu
[2] Centro de Estudio de Energía y Tecnologías Sostenibles, Universidad de Pinar del Río. Calle Martí Final No. 270, Pinar del Río, Cuba. mgi@upr.edu.cu

[*] Autor para correspondencia: barbaro@upr.edu.cu

Resumen

En este trabajo se describe la implementación de un sistema que se emplea para la detección del umbral óptimo en imágenes en escala de grises con un tamaño de 256x256 píxeles siguiendo el algoritmo propuesto por Otsu. El sistema que se propone se basa en una implementación híbrida hardware-software, donde el módulo principal de la parte hardware es precisamente la arquitectura propuesta que se corresponde con el algoritmo de Otsu. Una de las ventajas de este diseño es que se evita el empleo de divisores y casi la totalidad de multiplicadores. En el diseño se utiliza una placa Nexys2 de Digilent que contiene un FPGA XCS500E de Xilinx. Los resultados obtenidos son satisfactorios en cuanto a exactitud y cantidad de recursos internos del FPGA ocupados.

Palabras clave: Algoritmo de Otsu, FPGA, módulo IP, umbral.

Abstract

In this paper, the implementation of a system used for detecting the optimal threshold in grayscale 256x256 images based on the Otsu's algorithm is described. The proposed system is based on a hybrid hardware-software implementation, where the main module of the hardware part is precisely the proposed architecture that corresponds with the Otsu's algorithm. One advantage of this implementation is that the use of dividers and almost all the multipliers are avoided. For the implementation a Digilent Nexys2 Board that containing a FPGA Xilinx XCS500E is used. The results are satisfactory in terms of accuracy and number of internal FPGA resources occupied.

Keywords: *FPGA, IP module, Otsu's algorithm, threshold.*

Introducción

Revista Cubana de Ciencias Informáticas
Vol. 10, No. 3, julio-septiembre, 2016
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 16-26

La detección de umbral de manera automática es una técnica muy eficaz y sencilla utilizada en los campos de procesamiento de imágenes, reconocimiento de patrones y visión por computadora. Sin embargo, se requiere de un valor de umbral adecuado para extraer objetos de interés en una imagen determinada, ya que los objetos tienen sus propias distribuciones de distintos niveles de gris. Los métodos de detección de umbral son ampliamente utilizados más específicamente en aplicaciones tales como, el reconocimiento óptico de caracteres (MITHE, 2013), la inspección automática de defectos (PAK, 2016), la detección de cambios en video (DEL FARO, 2013), segmentación de objetos en movimiento (OCHS, 2014) y el diagnóstico de imágenes médicas (SENTHILKUMARAN, 2016). Como tarea fundamental en la etapa de pre procesamiento de una imagen, muchos investigadores prestan gran atención a la forma de determinar los umbrales apropiados (SI-QI, 2010).

Estas aplicaciones requieren de ejecución en tiempo real, de ahí la necesidad de su implementación en hardware, especialmente en dispositivos FPGAs (*Field Programmable Gate Array*) (HAHNLE, 2013). Con este tipo de dispositivo se aumenta la eficiencia computacional de los procedimientos de detección de umbrales. Por lo tanto, la elección de un método para la detección de umbral como parte de una determinada aplicación en un dispositivo FPGA es importante. El método de Otsu (OTSU, 1979) es una técnica de umbralización automática con muy buenos resultados y fácil de asimilar, por lo que goza de gran popularidad. Sin embargo, el cálculo del umbral mediante el método de Otsu implica muchas operaciones aritméticas complejas iterativas, tales como multiplicaciones y divisiones, que no contribuyen a una ejecución de alta velocidad y de bajo consumo de recursos.

Los dispositivos FPGAs modernos permiten empotrar muchos módulos IPs predefinidos, tales como elementos de procesamiento digital de señales, memorias, microprocesadores, controladores de periféricos, buses, entre otros (HERNÁNDEZ, 2012). En este trabajo, se propone una arquitectura basada en FPGA para la implementación del algoritmo de Otsu.

Metodología computacional

El método de Otsu se basa en maximizar una medida estadística que se denomina varianza entre clases. Un umbral que produzca la mejor separación entre clases, en términos de los valores de intensidad de estas, será el umbral óptimo (OTSU, 1979; SEZGIN, 2004; ZHIYONG, 2013). Si se tiene una imagen en escala de grises con L niveles de intensidad de gris (0, 1, 2, …, L-1), el número de píxeles con nivel i se denota por n_i y el total del número de píxeles es $N = n_0 + n_1 + n_2 + \cdots + n_{L-1}$; el histograma normalizado de la imagen se obtiene de la siguiente manera:

$$p_i = \frac{n_i}{N}, p_i \geq 0, \sum_{i=0}^{L-1} p_i = 1 \qquad (1)$$

Si se supone que los píxeles de la imagen se agrupan en dos clases C_0 y C_1 (objeto y fondo de la imagen, o viceversa) mediante un umbral de nivel t, se tiene que la clase C_0 se corresponde con los píxeles con niveles [0…t] y la clase C_1 se corresponde con los píxeles con niveles [t+1,…,L-1]. Por tanto, la probabilidad de que un píxel sea asignado a cada una de estas clases viene dada por (OTSU, 1979; KALATHIYA, 2014):

$$w_0 = P_r(C_0) = \sum_{i=0}^{t} p_i = w(t) \qquad (2)$$

$$w_1 = P_r(C_1) = \sum_{i=t}^{L-1} p_i = 1 - w(t) \qquad (3)$$

Los valores medios de las intensidades correspondientes a las clases C_0 y C_1 respectivamente son:

$$\mu_0 = \sum_{i=0}^{t} iP_r(i|C_0) = \sum_{i=0}^{t} \frac{ip_i}{w_0} = \frac{\mu(t)}{w(t)} \qquad (4)$$

$$\mu_1 = \sum_{i=t+1}^{L-1} iP_r(i|C_1) = \sum_{i=t+1}^{L-1} \frac{ip_i}{w_1} = \frac{\mu_T - \mu(t)}{1-w(t)} \qquad (5)$$

Donde $w(t) = \sum_{i=0}^{t} p_i$ y $\mu(t) = \sum_{i=0}^{t} ip_i$ son la suma de las intensidades e intensidad promedio hasta el nivel t respectivamente para la imagen. Mientras que $\mu(t) = \mu(L-1) = \sum_{i=0}^{L-1} ip_i$ es la intensidad media global para toda la imagen. Con las definiciones antes realizadas se cumplen las siguientes relaciones:

$$w_0\mu_0 + w_1\mu_1 = \mu_T, w_0 + w_1 = 1 \qquad (6)$$

La varianza de las clases C_0 y C_1 viene dada por:

$$\sigma_0^2 = \sum_{i=0}^{t} (i-\mu_0)^2 P_r(i|C_0) = \sum_{i=0}^{t} \frac{(i-\mu_0)^2 P_i}{w_0} \qquad (7)$$

$$\sigma_1^2 = \sum_{i=t+1}^{L-1} (i-\mu_1)^2 P_r(i|C_1) = \sum_{i=t+1}^{L-1} \frac{(i-\mu_1)^2 P_i}{w_1} \qquad (8)$$

Con el fin de evaluar la factibilidad del umbral (en el nivel t), se toman las siguientes medidas de criterio discriminante (o medidas de la separabilidad entre clases) utilizadas en el análisis discriminante como se muestra en la ecuación (9).

$$\sigma_B^2 = \omega_0(\mu_0 - \mu_T)^2 + \omega_1(\mu_1 - \mu_T)^2 \qquad (9)$$

Entonces el problema se reduce a un problema de optimización, el cual consiste en buscar un umbral t que maximice las funciones de objeto (las medidas de criterio) en la ecuación (9).

Este punto de vista está motivado por el hecho de que la varianza entre clases es una medida de la diferencia entre dos partes. Cuanto mayor sea la varianza entre clases, mayor es la diferencia entre las dos partes. Si la varianza entre clases disminuye, esto implica cometer un error a la hora de extraer un objeto del fondo de la imagen.

Como resultado, se hace que la varianza entre clases sea máxima, lo que implica que la probabilidad de error sea mínima. El umbral que contribuya a la mejor separación de las clases en niveles de gris, sería el mejor umbral. Esta es precisamente la esencia del algoritmo de Otsu.

El umbral óptimo t^* que maximiza σ_B^2 , se selecciona en la siguiente búsqueda secuencial para los valores de t de 0 a L-1 mediante el uso de las cantidades acumuladas simples $\omega(t) = \sum_{i=0}^{t} p_i$ y $\mu(t) = \sum_{i=0}^{t} ip_i$ o explícitamente usando las ecuaciones (2; 3; 4) y (5):

$$\sigma_B^2(t) = \frac{[\mu_T \omega(t) - \mu(t)]^2}{\omega(t)[1-\omega(t)]} \qquad (10)$$

Por tanto, el umbral óptimo t* es:

$$\sigma_B^2(t^*) = max_{0 \le t \le L-1} \left| \sigma_B^2(t) \right| \qquad (11)$$

En la ecuación (10), se observa que el cálculo de la varianza entre clases requiere un gran número de operaciones matemáticas de suma, multiplicación y división. En el caso especial de la división, su implementación en hardware es difícil y ocupa gran cantidad de recursos. Una posible solución es el empleo del logaritmo, de esta forma las operaciones de multiplicación y división se convierten en sumas y restas respectivamente. Por lo tanto, la ecuación (10) puede ser transformada de la siguiente manera:

$$\log_2 \sigma_B^2(t) = 2\log_2[\mu_T \omega(t) - \mu(t)] - \log_2 \omega(t) - \log_2[1 - \omega(t)] \qquad (12)$$

La función antilogaritmo no es requerida en este caso, ya que $\sigma_B^2(t)$ y $\log_2 \sigma_B^2(t)$ alcanzan su valor máximo para el mismo umbral.

En este caso, el umbral óptimo puede ser obtenido como:

$$\sigma_B^2(t*) = max_{0 \le t \le L-1} \left| \log_2 \sigma_B^2(t) \right| \qquad (13)$$

En el algoritmo de Otsu, el cálculo de la varianza entre clases tiene que ser iterado para los niveles de L grises, y por lo tanto, el punto crítico a la hora de implementar una arquitectura hardware correspondiente a dicho algoritmo se encuentra en el cálculo de la varianza entre clases. De ahí la importancia de que se desarrolle una buena arquitectura para mejorar la eficiencia de este algoritmo.

Revista Cubana de Ciencias Informáticas
Vol. 10, No. 3, julio-septiembre, 2016
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 16-26

Arquitectura propuesta

La arquitectura propuesta correspondiente al algoritmo de Otsu se basa en la ecuación (12) y se muestra en la Figura 1. El hardware se implementa usando la programación VHDL (Very High Speed Integrated Circuit Hardware Description Language) mediante la herramienta ISE 12.1 de la compañía Xilinx y como placa de desarrollo se emplea la Nexys2 (DIGILENT INC, 2008) de la compañía Digilent, la cual posee como dispositivo central un FPGA XCS500E.

La arquitectura consiste en algunos módulos. Inicialmente las memorias RAM 1 y RAM 2 reciben simultáneamente por ADDR de manera sincronizada con el flanco de subida de la señal de reloj CLK, los valores de todos los píxeles de la imagen que se le aplica el algoritmo de Otsu. Estas memorias son de 256x16 bits y 256x24 bits respectivamente, pues las imágenes que se emplean son en escala de grises y tienen un tamaño de 256x256 píxeles.

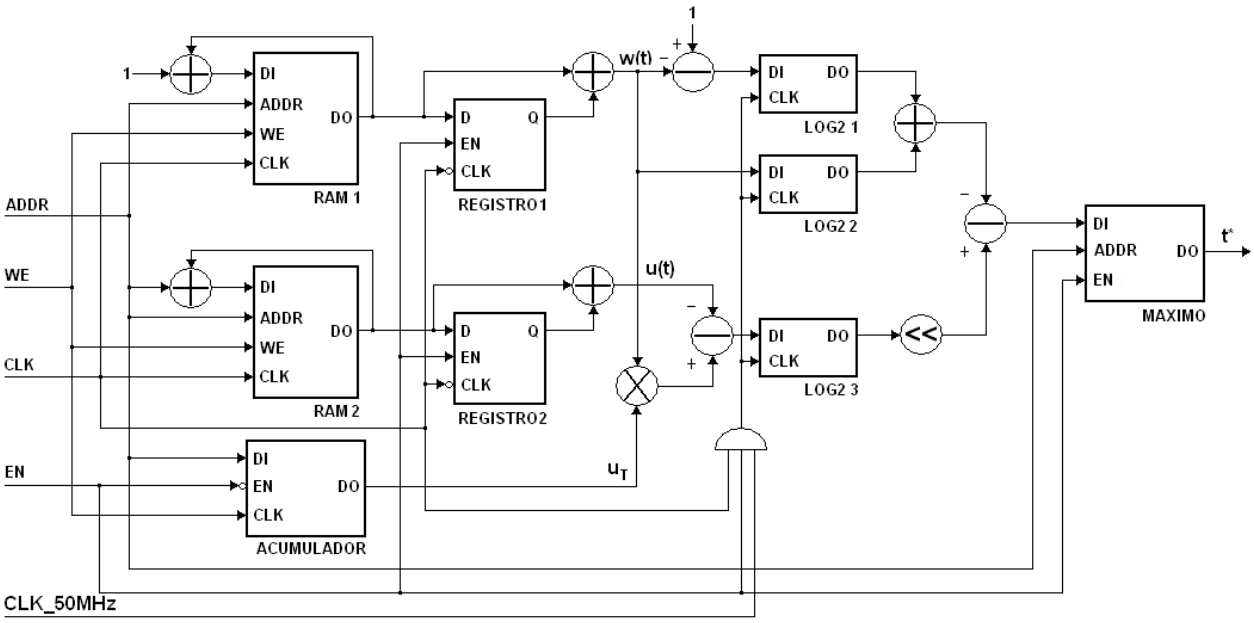

Figura 1. Arquitectura propuesta

En la memoria RAM 1 se obtiene el histograma normalizado de la imagen. Por ejemplo, si la imagen tiene 150 píxeles con un valor de 200, esto implica que en la posición 200 (C8H) de dicha memoria se encuentra el dato 150/(256*256) = 0,002288818359375, o sea, el número 150 desplazado 16 veces a la derecha (0096H). Es necesario resaltar que solamente se almacena la parte fraccionaria. Por otro lado, en la memoria RAM 2 se obtiene el histograma normalizado de las intensidades medias. Si se toman los mismos datos del ejemplo antes expuesto, se tiene que en la posición 200 (C8H) de esta memoria se encuentra el dato 150*200/(256*256) = 0,457763671875, o

Revista Cubana de Ciencias Informáticas
Vol. 10, No. 3, julio-septiembre, 2016
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 16-26

sea, el número 30000 desplazado 16 veces a la derecha (007530H). Es necesario resaltar que los ocho bits más significativos del dato almacenado corresponden a la parte entera y los restantes 16 bits a la parte fraccionaria.

El acumulador también recibe los mismos datos que las memorias. La función de este módulo es acumular la suma de todos los datos que recibe, estos datos previamente se dividen entre 256*256 de manera similar a como se explicó en el caso de las memorias. La suma total obtenida a partir de este módulo equivale a sumar todos los valores almacenados en la memoria RAM 2, o sea, la intensidad media global de toda la imagen.

En la Figura 2 se muestra un diagrama temporal de las señales para el acceso a los módulos antes mencionados. Se debe resaltar que cada uno de los 65536 (256*256) valores de los píxeles permanecen constantes durante dos períodos de la señal de reloj CLK, pues primero se realiza el proceso de lectura y después el de escritura sobre las memorias. Durante todo este tiempo la señal EN permanece a nivel bajo.

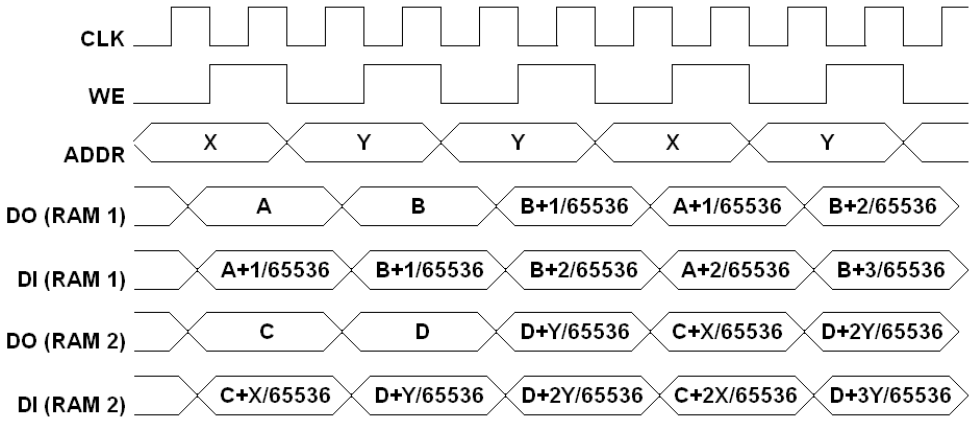

Figura 2. Diagrama temporal durante la recepción de los datos de la imagen

Una vez recibidos todos los píxeles de la imagen, se procede a calcular los umbrales. Inicialmente se leen de las memorias los datos almacenados desde la primera posición hasta la última (256 posiciones), como se muestra en la Figura 3. Con el empleo de los registros y el sumador correspondiente se logra que la lectura sea acumulativa. Por ejemplo, cuando se lea el valor del dato almacenado en la segunda posición, este valor se le suma al dato almacenado en la primera posición, el cual fue previamente leído y registrado. Durante todo este tiempo la señales EN y WE permanecen a nivel alto y bajo respectivamente.

Revista Cubana de Ciencias Informáticas
Vol. 10, No. 3, julio-septiembre, 2016
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 16-26

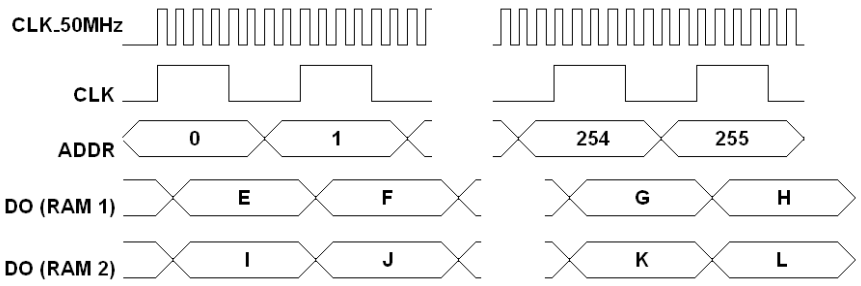

Figura 3. Diagrama temporal durante el cálculo de los umbrales

Posteriormente, se realizan algunas operaciones de suma, resta, multiplicación y logaritmo según la ecuación (12). El cálculo de esta última operación se realiza mediante un módulo descargado gratuitamente de (OPENCORES, 2011), el cual es reutilizado tres veces. Estos módulos trabajan en punto fijo y tienen una latencia de tres pulsos de reloj, por lo que esta señal es tomada directamente de la placa que se emplea, la cual tiene una elevada frecuencia de 50 MHz. Se debe resaltar que debido a la compuerta AND de tres entradas empleada, los módulos LOG 1, LOG 2 y LOG 3, sólo van a trabajar después que se reciban todos los datos de la imagen y cuando la señal de reloj CLK se encuentre a nivel alto.

El último módulo (MAXIMO) recibe los 256 datos resultantes de todo el proceso antes descrito, determina cuál de ellos es el mayor, y, por último, muestra la posición de la memoria correspondiente al mismo, o sea, el umbral óptimo.

Debido a que el algoritmo de Otsu en muchas aplicaciones constituye solamente una etapa de estas, se decide empaquetar la arquitectura implementada como un módulo IP para su empleo en un sistema híbrido hardware-software basado en un módulo IP del procesador Microblaze (XILINX INC, 2010) conectado a distintos módulos IP que actúan como periféricos. Esta interconexión se realiza a través de un bus de conexión de propósito general denominado PLB (*Processor Local Bus*). Todo este sistema se implementa mediante el entorno Xilinx Platform Studio de la compañía Xilinx.

En la Figura 4 se muestra el diagrama de bloques de la plataforma de desarrollo.

El periférico UART se encarga de la recepción de la imagen que se desea procesar y una vez terminado el procesamiento realiza la transmisión del resultado, ambos procesos se llevan a cabo por vía RS-232. El resultado también es mostrado en los diodos LEDs que posee la placa mediante el periférico Puestos E/S. El Controlador de Interrupciones se encarga de manejar las interrupciones de los periféricos que así lo requieran.

Revista Cubana de Ciencias Informáticas
Vol. 10, No. 3, julio-septiembre, 2016
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 16-26

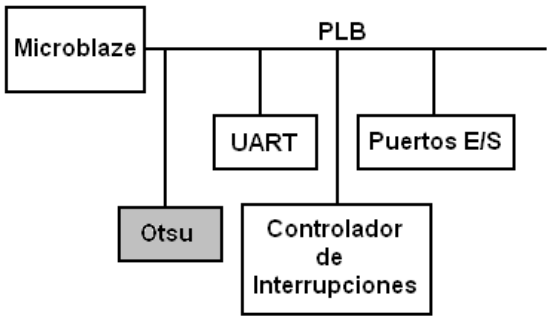

Figura. 4: Diagrama de bloques de la plataforma de desarrollo

Resultados y discusión

Los resultados del diseño implementado se analizan en términos de exactitud y cantidad de recursos internos del FPGA consumidos. Para el primer caso, se utilizan seis imágenes diferentes de entrada, estas imágenes son de las más empleadas en la literatura especializada, las cuales se muestran en la Figura 5.

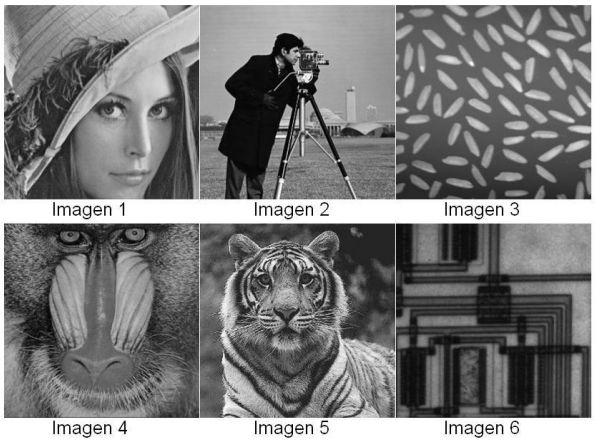

Figura. 5: Imágenes empleadas para realizar las pruebas

Con el propósito de comparar los valores umbrales óptimos de cada una de las seis imágenes anteriores que se obtienen a partir del sistema diseñado, se emplea la función graythresh() del software Matlab. Esta función también se basa en el algoritmo de Otsu. En la Tabla 1 se muestran los resultados obtenidos empleando ambas alternativas.

En la Tabla 1 se muestra que con ambas alternativas se obtienen valores de umbrales muy parecidos, lo cual refleja una buena exactitud de la arquitectura propuesta. Sin embargo, si se aumenta el número de bits correspondientes a la parte fraccionaria de los módulos LOG 1, LOG 2 y LOG 3; entonces los resultados obtenidos por la arquitectura propuesta pueden ser aún más exactos.

Revista Cubana de Ciencias Informáticas
Vol. 10, No. 3, julio-septiembre, 2016
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 16-26

Tabla 1. Resultados de la comparación

Imagen	Matlab	Arquitectura propuesta
1	113	110
2	88	86
3	125	124
4	101	101
5	120	118
6	86	88

Un resumen de los recursos internos ocupados en el FPGA por la plataforma de desarrollo (Figura 4) se puede apreciar en la Tabla 2. Los valores que se encuentran entre paréntesis en la segunda y cuarta columna se refieren a la arquitectura propuesta correspondiente al algoritmo de Otsu únicamente (Figura 1).

Tabla 2. Recursos ocupados en el FPGA

Recursos	Ocupados	Disponibles	Utilización
Slices	2089 (1352)	4656	44 % (7 %)
BRAMs	6 (3)	20	30 % (15 %)
MULT18X18SIOs	7 (4)	20	35 % (20 %)

Según la Tabla 2, la cantidad de recursos internos del FPGA que ocupa la arquitectura propuesta en la Figura 1 es mínima, aun cuando se considere la plataforma de desarrollo de la Figura 4, la cantidad de recursos ocupados no llegan a la mitad de los disponibles. De esta forma quedan suficientes recursos libres que pueden ser utilizados para incorporar nuevos módulos IP con el propósito de implementar una aplicación más compleja. En tal caso, este proceso es muy sencillo debido a la plataforma de desarrollo escogida.

Conclusiones

En este trabajo se ha diseñado una arquitectura hardware usando el lenguaje VHDL para el desarrollo del algoritmo de Otsu. El empleo de módulos que realizan la operación logaritmo es una variante para evitar divisores y multiplicadores, aunque estos últimos no se pueden eliminar en su totalidad. Esta arquitectura se convierte en un módulo IP para facilitar su empleo en un sistema híbrido hardware-software, el cual también es propuesto. Los valores de umbral generados por el sistema propuesto para distintas imágenes se comparan con los obtenidos mediante el software Matlab y los resultados son satisfactorios. El total de recursos internos del FPGA consumidos por la arquitectura propuesta es bajo, aun cuando se tenga en cuenta el sistema general no excede a la mitad de los mismos, de esta manera quedan recursos libres suficientes para implementar nuevas aplicaciones correspondientes al procesamiento de imágenes.

Referencias

MITHE, R., INDALKAR, S., DIVEKAR, N. Optical Character Recognition. International Journal of Recent Technology and Engineering, 2013, 2(1): p. 72-75.

PAK, M., KIM, S. A Study on Defect Ispection System using Efficient Thresholding Method. INFORMATION, 2016, 19(2): p. 623-630.

DEL FARO, M., BOSZORMENYI, L. State-of-art and future challenge in video scene detection: a survey. Multimedia Systems, 2013, 19(5): p. 427-454.

OCHS, P., MALIK, J., BROX, T. Segmentation of moving objects by long term video analysis. IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, 2014, 36(6): p. 1187-1200.

SENTHILKUMARAN, N., VAITHEGI, S. Image Segmentation By Using Thresholding Techniques For Medical Images. Computer Science & Engineering: An International Journal, 2016, 6(1).

SI-QI, H., LEI, W. A Survey of Thresholding Methods for Image Segmentation. Systems Engineering and Electronics, 2010, 24(6): p. 91-94.

HAHNLE, A., SAXEN, F., HISUNG, M., BRUNSMANN, K. FPGA-Based Real-Time Pedestrian Detection on High-Resolution Images. IEEE Conference on Computer Vision and Pattern Recognition, 2013: p. 629-635.

OTSU, N. A threshold selection method from gray-level histogram. IEEE Transactions on Systems, Man and Cybernetycs, 1979, 9(1): p. 62-66.

HERNÁNDEZ, E., CABRERA, A., SÁNCHEZ, S. IMPLEMENTACIÓN HÍBRIDA HARDWARE SOFTWARE DEL ALGORITMO DE DETECCIÓN DE ROSTROS DE VIOLA-JONES SOBRE FPGA. UNIVERSIDAD, CIENCIA Y TECNOLOGÍA, 2012, 16(63): p. 114-124.

SEZGIN, M., SANKUR, B. Survey over image thresholding techniques and quantitative performance evaluation. Journal of Electronic Imaging, 2004, 13(1): p. 146-165.

ZHIYONG, H., LINING, S., LIGUO, C. Fast Computation of Threshold Based on Otsu Criterion. Acta Electronica Sinica, 2013, 41(2): p. 267-272.

KALATHIYA, S., PATEL, V. Implementation of Otsu Method With Two Different Approaches. International Journal of Software & Hardware Research in Engineering, 2014, 2(2): p. 24-28.

DIGILENT INC. Digilent Nexys2 Board Reference Manual. [En línea]. 2008. [Consultado el: 16 de marzo de 2015]. Disponible en: http://digilent.org/Data/Products/NEXYS2/Nexys2_rm.pdf

OPENCORES. Logarithm function, base-2, single-cycle :: Overview. [En línea]. 2011. [Consultado el: 20 de marzo de 2015]. Disponible en: Disponible en: http://opencores.org/project,fast_log

XILINX INC. MicroBlaze Processor Reference Guide. [En línea]. 2010. [Consultado el: 16 de marzo de 2015]. Disponible en: http://www.xilinx.com/support/documentation/sw_manuals/xilinx13_4/mb_ref_guide.pdf