Arquitectura hardware para etiquetado de bordes en una imagen binaria

Bernardo Cante-Michcol, Miguel O. Arias-Estrada
Instituto Nacional de Astrofisica Óptica y Electrónica
Tonantzintla, Puebla. México
bcante@susu.inaoep.mx, ariasm@inaoep.mx

Resumen

La detección y etiquetado de componentes conectados es un paso esencial en muchas técnicas de análisis de imágenes. En este trabajo se ha desarrollado una arquitectura hardware que realiza la detección y umbralización de bordes en una imagen como una primera etapa, posteriormente usando los resultados obtenidos, se usa una arquitectura hardware que realiza el etiquetado de componentes conectados usando conectividad-4. La arquitectura ha sido diseñada y verificada usando VHDL, la síntesis e implementación se ha realizado en un solo FPGA usando Foundation de Xilinx. El funcionamiento de la arquitectura desarrollada fue comprobado mediante simulaciones postsíntesis en Active-VHDL. La arquitectura puede hacer procesamiento en tiempo real que depende del tamaño de la imagen y de la frecuencia final de implementación.

1. Introducción

Muchas tareas de procesamiento de imágenes requieren la identificación y detección de regiones de interés para un posterior procesamiento. La detección de bordes constituye una operación que proporciona información relevante para operaciones posteriores tal como el etiquetado de componentes conectados. En el análisis de imágenes los objetos son usualmente extraídos por medio de una etapa de etiquetado de componentes conectados, la cual consiste en el asignamiento de una etiqueta única a cada región conectada (foreground) que no pertenece al fondo (background) de la imagen, lo cual significa transformar la imagen binaria en una imagen simbólica. La operación de detección de bordes y de etiquetado consumen mucho tiempo de computo y es por ello que en este trabajo se presenta una arquitectura hardware para la realización del etiquetado de componentes conectados (bordes) en imágenes binarias usando conectividad 4, para obtener un procesamiento en tiempo real. Como punto de partida se cuenta con una imagen en escala de grises (256 niveles) a la cual se detectan y umbralizan sus bordes usando una arquitectura hardware del algoritmo SUSAN [6], [7]. Posteriormente, se usa la arquitectura hardware diseñada, del algoritmo de etiquetado de componentes conectados [1] en la imagen de bordes binaria.

2. Algoritmo SUSAN

El algoritmo SUSAN para la detección de bordes y esquinas [6], se fundamenta en un principio básico denominado principio SUSAN "Smallest Univalue Segment Assimilating Nucleus". Para obtener mayor información se recomienda acudir a las referencias [6], [7] y [10].

3. Algoritmo de etiquetado

Viendo la imagen digitalizada como un arreglo rectangular de píxeles, se puede notar que la conectividad en las regiones de una imagen implica que las etiquetas pueden ser propagadas localmente entre los píxeles adyacentes. Sin embargo, la etiqueta asignada a un píxel puede ser la misma que la de un píxel en una localización relativamente distante dentro de la imagen. El problema de etiquetado así posee características locales y globales, que han sido explotadas en el desarrollo de algoritmos secuenciales y paralelos.

El algoritmo secuencial clásico para etiquetado de componentes conectados consiste de dos barridos de arriba a abajo y de izquierda a derecha (raster-scan) de la imagen (I) [17]. Indicando como Lx el píxel a ser etiquetado, el conjunto de los vecinos ya etiquetados, N, es dado por $\{Lp, Lq, Lr, Ls\}$ en caso de conectividad 8 y $\{Lp, Lq\}$ en caso de conectividad 4. Lo anterior se muestra en la Figura 1.

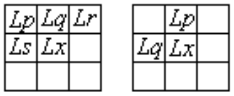


Figura 1. Vecinos usados en el cálculo de los componentes conectados para conectividad 8 y conectividad 4 respectivamente en un segmento de imagen.

El algoritmo de etiquetado usado [1] consiste en dos barridos sobre la imagen. A diferencia de la aproximación clásica, las equivalencias son procesadas durante el

primer barrido para determinar el correcto estado de las equivalencias de clases en cada momento del barrido.

En el algoritmo se realiza la operación de unión usando una estructura de datos llamada class array (arreglo de clase) C. C es un arreglo unidimensional tan grande como el máximo valor de etiqueta, y conteniendo para cada valor de etiqueta su identificador de clase correspondiente (en otras palabras $C[i]$ es la equivalencia de clase asociada con la etiqueta i). C es inicializado haciendo $C[i] = i, i = 0 \ldots \max label.$ Cuando se encuentra que dos etiquetas, l_i y l_j son equivalentes durante el primer barrido, las clases correspondientes $C[l_i]$ y $C[l_j]$ son unidas. La unión consiste en primero establecer que uno de los dos identificadores de clases sea el sobreviviente y el otro sea borrado. La elección del sobreviviente puede ser arbitraria (por ejemplo retener el valor mínimo de $C[l_i]$ y de. Después de finalizar con el primer barrido el arreglo de clase mantiene el identificador de clase asociado con cada etiqueta temporal, y así puede ser usado como una tabla en el segundo barrido, para cambiar valores de etiquetas temporales por su identificador de clase correspondiente.

4. Arquitectura propuesta

4.1. La arquitectura SUSAN

Esta arquitectura está formada por 7 procesadores que trabajan en forma paralela aprovechando datos que son comunes al realizar el procesamiento de la imagen con ventanas de procesamiento de 7×7 píxeles. La arquitectura desarrollada en este trabajo varía con respecto a la reportada en [7], en la forma en que se realiza la etapa de control y además se agrega una modificación para obtener imágenes binarias.

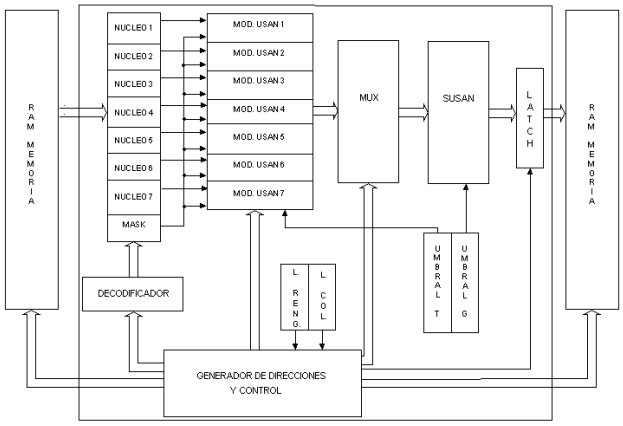


Figura 2. Diagrama de bloques de la arquitectura hardware del algoritmo SUSAN para detección de bordes.

En la Figura 2 puede verse los bloques internos de la arquitectura. Los bloques encerrados por el rectángulo corresponden a la parte de la arquitectura que es sintetizada e implementada en un FPGA a nivel de simulaciones postsíntesis. Las memorias RAM son simuladas sólo como bloques funcionales y sirven para realizar las pruebas a la arquitectura.

4.2. La arquitectura de etiquetado

En la Figura 3 puede verse la forma en que se conectan los módulos de la arquitectura, correspondientes a las etapas del primer barrido (TOP_LEC) y al segundo barrido (TOP_LEC2) controlados con el bloque llamado CONT_PRIN.

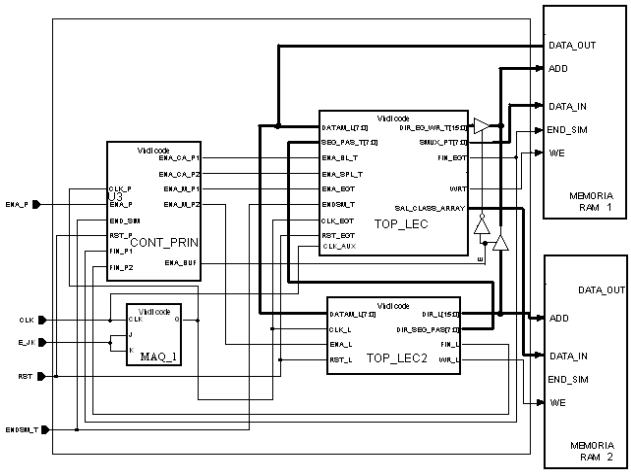


Figura 3. Arquitectura de etiquetado en el nivel de más alta jerarquía.

Los bloques encerrados por el rectángulo corresponden a la parte que es sintetizada e implementada en el FPGA a nivel de simulaciones postsíntesis. Los bancos de memoria RAM 1 y RAM 2 son simulados como bloques funcionales y sirven para realizar las pruebas a la arquitectura.

4.2.1. Bloque funcional del primer barrido. En la Figura 4 se muestra el diagrama de bloques.

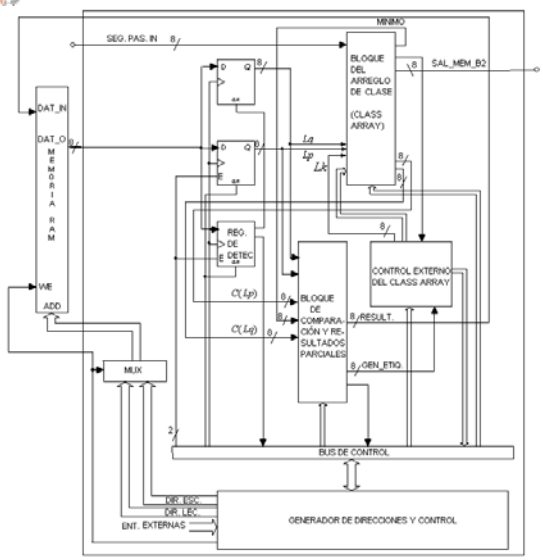


Figura 4. Diagrama de bloques para el primer barrido de la arquitectura de etiquetado.

Durante el primer barrido se realiza la asignación de una etiqueta temporal dependiente del valor que tengan los vecinos empleados en el proceso de etiquetado En este barrido se realiza la actualización de las memorias del bloque del arreglo de clase. Después de terminar de realizar el primer barrido, las memorias del class array ya han sido actualizadas y una de ellas se utiliza como una tabla en la realización del segundo barrido.

En los registros de almacenamiento temporal se almacenan los valores de Lp y Lq. Lp y Lq son los datos leídos de la memoria. Los valores almacenados en las memorias del arreglo de clase y direccionados por Lp y Lq se designan por $C(Lp)$ y $C(Lq)$, dichos valores se encuentran almacenados en registros internos dentro del bloque del arreglo de clase y son usados en el procesamiento del primer barrido de la arquitectura.

El bloque del class array (C) es un arreglo unidimensional tan grande como el máximo valor de etiqueta, y contiene para cada valor de etiqueta su identificador de clase correspondiente, es decir $C[i]$ es la equivalencia de clase asociada con la etiqueta i.

4.2.2. Bloque funcional del segundo barrido. En la Figura 5 se muestra el diagrama de bloques. En esta etapa se sustituye la etiqueta temporal por el valor almacenado en las memorias del class array. Esta operación consiste en direccionar una de las memorias del arreglo de clase con los valores de las etiquetas temporales de la imagen obtenidos del primer barrido (RAM 1), y usar los datos almacenados en dicha memoria para sustituir a las etiquetas temporales por las equivalencias de clases y almacenarlos en la memoria RAM 2.

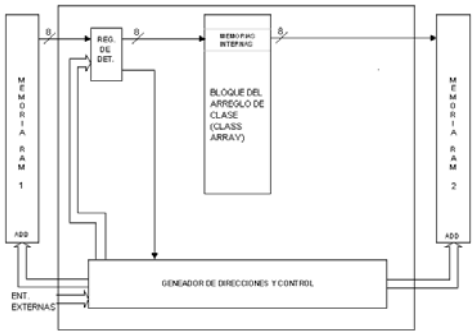
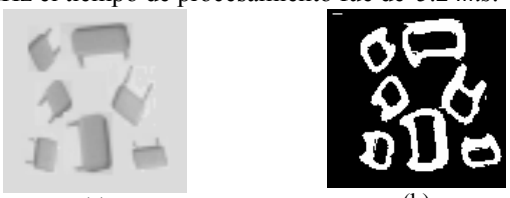


Figura 5. Diagrama de bloques para el segundo barrido de la arquitectura de etiquetado.

5. Resultados

5.1. Arquitectura SUSAN

En la Figura 6(a) se muestra la imagen original, la cual contiene algunos objetos. Esta imagen es de 128×128 píxeles. La Figura 6(b) muestra el resultado de la simulación postsíntesis de la arquitectura SUSAN desarrollada usando los valores de umbral para $t = 12$ y para $g = 36$ [6]. Usando una frecuencia de simulación de 25 MHz el tiempo de procesamiento fue de $5.24ms$.



(a) (b)

Figura 6. Resultados de la detección de bordes y umbralización. (a) Imagen original. (b) Simulación postsíntesis con la arquitectura desarrollada.

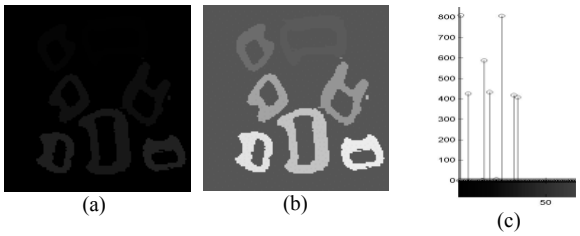
5.2. Arquitectura de etiquetado

En la Figura 7 se muestran las imágenes resultantes al procesar la imagen con la etapa de la arquitectura dedicada al primer barrido. La Figura 7(a) corresponde a la imagen obtenida mediante la arquitectura SUSAN. De aquí en adelante el uso del histograma de la imagen de resultados es con la finalidad de mostrar los niveles de gris presentes en la imagen de resultados (en los resultados se muestra solo parte del histograma). Cada barra del histograma (nivel de gris) corresponde al conteo de píxeles con la misma etiqueta. La Figura 7(b) corresponde al resultado después de realizar el primer barrido (modificando el contraste y brillo de la imagen para mejor visualización). Usando una frecuencia de simulación de 25 MHz el tiempo de procesamiento en el primer barrido fue de $1.86ms$.

(a) (b) (c)

Figura 7. Simulación postsíntesis del primer barrido. (a) Imagen original a ser procesada. (b) Resultado obtenido después de hacer el primer barrido. (d) Parte del histograma de la imagen (b).

En la Figura 8 se muestran las imágenes resultantes al procesar la imagen de la Figura 7(b) con la etapa de la arquitectura dedicada al segundo barrido. La Figura 8(a) es la imagen obtenida al realizar los dos barridos de la imagen original (Figura 7 (a)), la Figura 8(b) es la Figura 8(a) con una mejora en el contraste y brillo para mejor visualización.



(a) (b) (c)

Figura 8. Simulación postsíntesis del segundo barrido. (a) Resultado obtenido después de hacer el segundo barrido. (b) Imagen (a) con una mejora en el nivel de gris para visualización. (c) Parte del histograma de la imagen (a).

En la Figura 8(c) se ven finalmente las regiones etiquetadas correspondientes a cada una de las barras del histograma, en este caso las regiones etiquetadas de interés son 7. Usando una frecuencia de simulación de 25 MHz el tiempo de procesamiento en el segundo barrido fue de 0.81*ms*. El tiempo de procesamiento total fue de aproximadamente 2.67*ms*.

La arquitectura fue diseñada usando VHDL y fue sintetizada e implementada en un FPGA Virtex XCV300-6-pq240 de Xilinx, su prueba se realizó a nivel de simulaciones postsíntesis en Active-VHDL. El resumen de la síntesis e implementación se muestra en la Tabla 1.

Arquitectura Implementada	**Slices**	**Velocidad MHz**	**No. Com-puertas eq.**	**Bloq. Mem. Int.**
Arq. SUSAN	512	27.4	8463	0
Arq. Etiq.	659	31.6	43626	2

Tabla 1. Resumen para la implementación final de la arquitectura.

6. Discusión

La arquitectura SUSAN realizada tiene una frecuencia máxima de operación de 27.4 MHz. Para la arquitectura diseñada procesando ventanas de 7×7 píxeles y usando una frecuencia de 25 MHz, se pueden procesar 12 imágenes por segundo de 512×512 píxeles.

La arquitectura de etiquetado realizada tiene una frecuencia máxima de operación de 31.6 MHz. En la Tabla 2 se muestra una comparación de rendimiento para la arquitectura de etiquetado diseñada contra otras arquitecturas de etiquetado. Usando una frecuencia de 25 MHz para imágenes de 128×128 se obtienen al menos 323 imágenes por segundo. Para imágenes de 512×512 se obtienen al menos 27 imágenes por segundo.

Arquitectura	**Dispositivo**	**Imagen**	**Tiempo De proc. *ms*.**
Ran, Meh, Sub [4] (40MHz)	Chip VLSI	128×128	0.85
Manohar [3] (10 MHz)	MPP (16384 PEs)	128×128	94.6
Rasqui., Ran. [5] (66 MHz)	4 FPGAs (128 PEs)	512×512	15.76
		128×128	0.992
Yang [8]	Secuencial	512×512	300
Nicol [9]	Chip sistólico	128×128	1.25
Arq. Desarrollada (25 MHz) Secuencial, 2 barridos	**FPGA (Virtex) XCV300-6-pq240**	512×512	**37**
		128×128	**3.1**

Tabla 2. Comparación de la arquitectura de etiquetado diseñada contra otras arquitecturas.

7. Conclusiones

Se ha presentado una arquitectura hardware que realiza el etiquetado de componentes conectados usando conectividad 4.

La arquitectura puede realizar procesamiento en tiempo real.

Las simulaciones postsíntesis garantizan al menos en un 90 % el funcionamiento de la arquitectura en una implementación real (hardware necesario y arquitectura diseñada).

Como parte del trabajo futuro se realizarán modificaciones en el bloque dedicado a realizar el barrido y actualización de la tabla de equivalencias para hacer más eficiente la etapa de actualización de la misma. Además se harán modificaciones para disponer de una tabla de equivalencias con más de 256 identificadores de clases (etiquetas temporales) disponibles.

8. Referencias

[1] Luigi Di Stefano, Andrea Bulgarelli, A simple and efficient connected components labeling Algorithm, *Proceedings of the 10th International Conference on Image Analysis and Processing,* IEEE 1999 (ICIAP'99), September 27 - 29, Venice, Italy, 322-327.

[2] Hussein M. Alnuweiri, Victor K. Prasanna, Parallel architectures and algorithms for image component

labeling, *IEEE PAMI vol. 14*(10), 1014-1034, October 1992.

[3] M. Manohar and H. K. Ramapriyan, Connected component labeling of binary images on a mesh connected massively parallel processor, *Computer vision, graphics, and image processing, vol. 45*(2), 133-149, 1989.

[4] N. Ranganathan, R. Mehrotra y S. Subramaniam, A high speed systolic architecture for labeling connected components in an image, *IEEE Trans. On systems, man and cybernetics, vol 25*(3), 415-423, March 1995.

[5] Rasquinha Ashley and Ranganathan N., C^3L : A chip for connected component labeling, *Proceedings of the tenth international conference on VLSI design: VLSI in Multimedia Applications*, 1997 IEEE, 446-450.

[6] Smith S. M., Brady J. M., SUSAN A New Approach to Low level image processing, Defense Research Agency Technical report TR95SMS1, 1995.

[7] C. Torres H., M. Arias E., Arquitectura FPGA para la extracción en tiempo real de bordes y esquinas de una imagen, *Encuentro nacional de computación (ENC'99)*, Pachuca Hgo., 12-15 Sept. 1999.

[8] X. D. Yang. Design of fast connected components hardware, *Proc. Computer Vision and Pattern Recognition Conference*, 1988, 937-944.

[9] C. J. Nicol, A Systolic Approach for Real Time Connected Component Labeling, *CVGIP: Image Understanding, vol 61*(1), 17-31, January 1995.

[10] C. Torres H., M. Arias E., An FPGA Architecture for high speed edge and corner detection, *Proc. on Computer architecture for Machine Perception, (CAMP 2000)*, Padova, Italy, 11-13 Sept.