

Algoritmo de Detección de Bordes en Imágenes con NIOS II

Calle Andrés, Pazmiño Paola, Ponguillo Ronald Ing.
Facultad de Ingeniería Eléctrica y Computación
Escuela Superior Politécnica del Litoral (ESPOL)
Campus Gustavo Galindo, Km 30.5 vía Perimetral
Apartado 09-01-5863. Guayaquil-Ecuador
{arcalles, elpapazm, rponguil}@espol.edu.ec

Resumen

El proyecto pretende dar a conocer la versatilidad de poder trabajar con sistemas embebidos configurables para la elaboración de proyectos que se puedan acoplar a cada una de las necesidades. Nuestro proyecto consiste en la implementación de un algoritmo de procesamiento de imágenes utilizando la tarjeta de desarrollo DE2 ALTERA, basada en un dispositivo ALTERA CYCLONE II FPGA junto con memorias embebidas y un PROCESADOR NIOS II INTEGRADO, que ayudará en el objetivo, el cual es detectar el borde de una imagen.

Para la realización del proyecto se aplican tres etapas. La primera etapa está basada en el almacenamiento de la imagen en la memoria SDRAM, la siguiente etapa se basa principalmente en leer los datos almacenados en la SDRAM y procesarlos correctamente para poder obtener su matriz, la cual contiene la información de la imagen que es de utilidad e irá cambiando en cada uno de los procesos que se realizarán para la obtención de una matriz final y ordenada. Y una última etapa donde por medio de algoritmos implementados en código c, se procede a obtener la imagen resultante, la misma que mostrara solamente los bordes de la imagen originalmente almacenada.

Palabras Claves: FPGA, Nios II, Algoritmo, Procesador, Imagen.

Abstract

The project seeks to highlight the versatility to work with configurable embedded systems for the development of projects that can be coupled to each of the needs. Our project involves the implementation of an image processing algorithm using the ALTERA DE2 development board, a device based on Altera Cyclone II FPGA with embedded memories and a NIOS II PROCESSOR INTEGRATED, which will help in the goal, which is to detect the edge of an image.

For the project applied three stages. The first stage is based on the storage of the image in the SDRAM memory, the next stage is mainly based on read data stored in the SDRAM and processed correctly to obtain the matrix, which contains the picture information that is utility and will change in each one of the processes which are performed to obtain a final matrix and tidy. A final step where by means of algorithms implemented in C code, it proceeds to obtain the resulting image will show only the same as the edges of the image originally stored.

Keywords: FPGA, Nios II, Algorithm, Processor, Image.

1. Introducción

El diseño e implementación de los circuitos digitales ha cobrado mucha importancia en los programas de estudio de ingenierías y es parte fundamental de la formación básica del estudiante. La evolución en este tipo de dispositivos ha venido presentando cambios importantes para el desarrollo de la tecnología.

Actualmente los FPGA's representan dispositivos versátiles, de fácil programación y precio accesible. Una vez que los FPGA's estuvieron al alcance de las universidades, muchas de ellas empezaron a incorporar en sus pensums de estudio cursos basados en el uso y diseño utilizando FPGA's en carreras tecnológicas afines.

En el área de procesamiento de imágenes, la detección de los bordes de una imagen es de suma importancia y utilidad, pues facilita muchas tareas, entre ellas, el reconocimiento de objetos y la segmentación de regiones.

2. Objetivos

2.1. Objetivo Principal

Desarrollar un código en NIOS II capaz de detectar el borde de una imagen que ha sido almacenada en la memoria SDRAM de un FPGA.

2.2. Objetivos Específicos

- Aprender el uso y manejo de la DE2 ALTERA.

- Estudiar la eficiencia de algoritmos para la detección de bordes.
- Implementar un proyecto que demuestre en base a los resultados la capacidad del algoritmo utilizado para detectar bordes.
- Aprender como se codifica la información que posee el formato bmp.

3. Alcances y Limitaciones

3.1. Alcances

- Almacenamiento de la imagen en la SDRAM de la Altera DE2.
- Lectura de los datos en un lenguaje de alto nivel.
- Estructuración de la matriz de la imagen a ser procesada.
- Obtener una matriz resultante con datos de la imagen con bordes detectados.
- Habilitar la salida VGA de la tarjeta DE2 Altera, para proceder a mostrar la imagen resultante.

3.2. Limitaciones

- Se cuenta con un solo método para guardar las imágenes en la SDRAM, el mismo que es por medio del programa DE2 CONTROL PANEL.
- Se usa imágenes en formato BMP de 256 colores debido a la factibilidad de trabajar con sus datos, ya que este tipo de formato usan 8 bits por pixel, es decir que necesitan 1 byte para codificar cada pixel, además este formato presenta información del tamaño de imagen, tamaño de archivo, ancho (pixel), alto (pixel), etc. de tal manera que la información necesaria se encuentra en una posición determinada, con lo que se puede formar una matriz de datos de la imagen.

4. Marco Teórico

4.1. Introducción a Quartus II y Lenguaje VHDL

Quartus es una herramienta de software producida por Altera para el análisis y la síntesis de diseños realizados en HDL. Dentro de las funciones que puede realizar Quartus, está el análisis de circuitos lógicos.

VHDL es un lenguaje usado para describir sistemas electrónicos digitales y es el acrónimo que representa la combinación de VHSIC y HDL. Este lenguaje permite la descripción de la estructura de un sistema, es decir, la forma en que se descompone en subsistemas y como estos subsistemas están interconectados. Además, permite la especificación de la función de un sistema mediante un lenguaje de

programación. VHDL permite el diseño de un sistema para ser simulado antes de ser fabricado.

Los diseñadores pueden rápidamente comparar alternativas poniendo a prueba las correcciones sin los gastos asociados a la creación de un prototipo de hardware.

4.2. Introducción a Eclipse

Eclipse es un entorno de desarrollo integrado de código abierto multiplataforma para desarrollar lo que el proyecto llama "Aplicaciones de Cliente Enriquecido", opuesto a las aplicaciones "Cliente-liviano" basadas en navegadores.

Esta plataforma, típicamente ha sido usada para desarrollar entornos de desarrollo integrados (del inglés IDE), como el IDE de Java llamado Java Development Toolkit (JDT) y el compilador (ECJ) que se entrega como parte de Eclipse (y que son usados también para desarrollar el mismo Eclipse). Sin embargo, también se puede usar para otros tipos de aplicaciones cliente, como BitTorrent o Azureus.

4.3. Nios II SBT para Eclipse

Nios II SBT para Eclipse es una pequeña capa de interfaz gráfica de usuario que ayuda a presentar un entorno unificado de desarrollo. El SBT para Eclipse proporciona una plataforma de desarrollo que funciona para todos los procesadores Nios II.

Se puede llevar a cabo todas las tareas de desarrollo de software dentro de Eclipse, incluyendo creación, edición, construcción, funcionamiento, depuración y perfilado de programas. Nios II SBT para Eclipse se basa en la estructura de Eclipse Framework y Eclipse Development Toolkit (CDT) Plugins.

4.4. Tarjeta de Desarrollo Altera DE2

La tarjeta de desarrollo de Altera DE2 fue diseñada con fines educativos, debido a que es una creación de profesores, dirigida para profesores, investigadores y estudiantes.

La razón por la cual tiene fines educativos, es debido a que posee una amplia gama de aplicaciones libres y con demos gratis que se incluyen en el disco de la tarjeta DE2 ALTERA, para la fácil comprensión de lógica digital, organización de computadoras y FPGA's. Además de poseer una amplia lista de ejercicios, donde también se pueden desarrollar tareas simples que ayudan a la fácil comprensión de conceptos fundamentales de cada uno de los distintos segmentos de la tarjeta. También se pueden llegar a realizar proyectos más complejos y avanzados.



Figura 1. Kit Tarjeta DE2 de ALTERA

4.5. FPGA´s

Una FPGA es un dispositivo semiconductor que contiene bloques lógicos, cuya interconexión y funcionalidad se puede programar. La lógica programable puede reproducir desde funciones tan sencillas como las de una puerta lógica hasta complejos sistemas en un chip (SoC).

Las FPGAs se utilizan en aplicaciones similares a los ASICs sin embargo son más lentas, tienen un mayor consumo de potencia y no pueden abarcar sistemas tan complejos como ellos. A pesar de esto, las FPGA's tienen las ventajas de ser reprogramables (lo que añade una enorme flexibilidad al flujo de diseño), sus costos de desarrollo y adquisición son mucho menores para pequeñas cantidades de dispositivos.

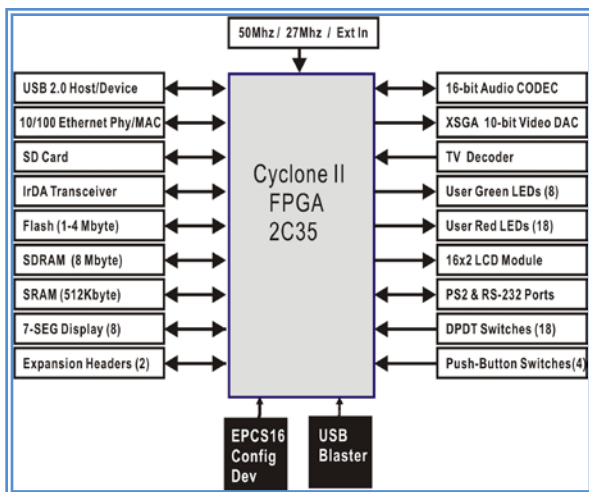


Figura 2. Cyclone FPGA 2C35

4.6. Procesador Embebido NIOS II

El Procesador Embebido Nios II es un sistema flexible capaz de ajustarse a las necesidades del diseñador, esta flexibilidad es gracias a la tecnología del SOPC builder que tiene como ventaja integrar en

un mismo chip una o varias CPUs y sus periféricos asociados, lo cual optimiza el sistema.

El Sopc builder brinda un entorno específico permitiendo la definición y configuración a medida del microprocesador NIOS II y que por medio de la herramienta Quartus II puede ser directamente utilizado sobre el FPGA.

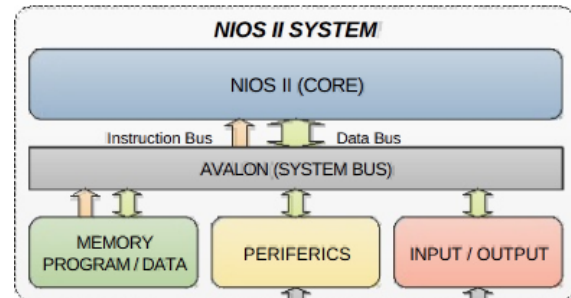


Figura 3 NIOS II System

4.7. Procesamiento Digital de Imágenes

Actualmente el procesamiento Digital de Imágenes es una herramienta muy específica en computación, se puede definir como las técnicas que se aplican a las imágenes digitales. Este procedimiento se utiliza en dos importantes áreas que son:

- mejorar la calidad de la información de una imagen
- procesamiento de los datos de una imagen de forma autónoma.

Para el procesamiento de la imagen es necesario tratar a la imagen como una matriz de valores numéricos, cada elemento de la matriz representa un pixel. Los valores de cada pixel se encuentran entre 0 y 255, en donde el 0 corresponde al negro que es el tono más oscuro y el 255 corresponde al blanco que es el tono más claro.

4.8. Detección de Bordes

Los bordes de una imagen son las secciones en donde ocurre un cambio brusco de intensidad, es decir en donde existe una transición entre dos partes que tienen diferentes niveles de gris. El resultado de este proceso sirve como entrada para tareas adicionales como reconocer objetos, compresión de imágenes, registro y alineación.

Hasta el momento se cuenta con varios algoritmos de detección de bordes, entre los más simples están:

- Algoritmo de Sobel
- Algoritmo de Prewitt
- Algoritmo Laplaciano

5. Diseño e Implementación

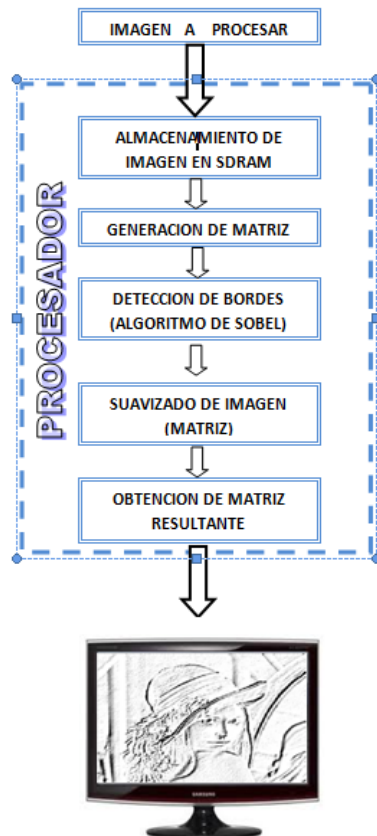


Figura 4. Diseño de Algoritmo Detector de Bordes

5.1. Almacenamiento de Imagen

En esta sección se hizo uso del disco que viene incluido dentro del Kit de la Tarjeta Altera DE2, se usó el programa DE2_Control_Panel que facilitó el almacenamiento de la imagen dentro de la SDRAM, además que permite especificar en qué sección de memoria dentro de la SDRAM, se quiere comenzar a almacenar la imagen y al finalizar el almacenamiento nos indica el tamaño de la misma.

Para poder almacenar la imagen por medio del programa DE2_Control_Panel, se debe habilitar los periféricos I/O por medio del DE2_Control_panel.sof, cuyo archivo lo encontramos en el disco que viene incluido con la tarjeta DE2 ALTERA, debido a que

contiene los periféricos necesarios para el almacenamiento de la imagen en la SDRAM

En la sección de Sequential Write, procedemos a seleccionar "file length", y en el campo Address escribimos la dirección donde se va a comenzar a guardar la imagen, en nuestro caso utilizamos la dirección 0x1000, una vez llenado estos campos damos clic sobre WRITE A FILE TO SDRAM, y seleccionamos la imagen que se va a procesar, de esta manera la imagen seleccionada se irá almacenando secuencialmente en la SDRAM desde la dirección indicada.

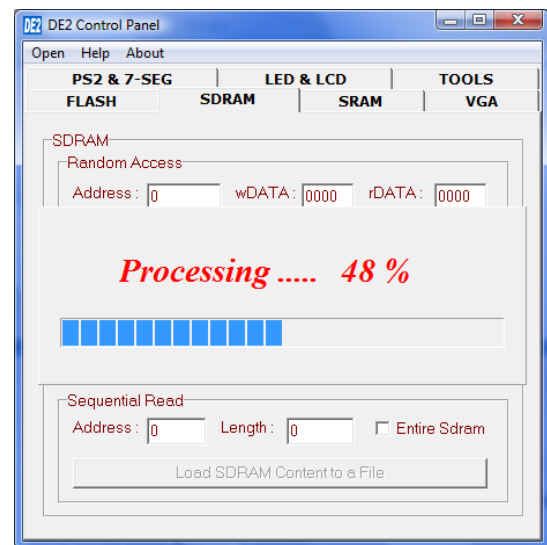


Figura 5 Proceso de Almacenamiento en SDRAM

5.2. Procesamiento de Imagen Matriz

En la sdram la imagen es almacenada en memoria de una forma especial, es decir que cuando se guarda en la posición 0x1000 como se observa en la Figura 3.4, en realidad el archivo comienza a ser almacenado en memoria desde la posición 0x4000 cuyo valor decimal es 16384. El número de filas se encontrará almacenado desde la posición 16406 hasta la 16407, y el número de columnas desde la posición 16402 hasta la 16403.

Posteriormente se empieza a leer los datos de la imagen que son necesarios para detectar el borde de la imagen, los mismos que se encuentran a partir de la posición 18486. Además la imagen se encuentra distribuida internamente en memoria en saltos de 512, es decir desde donde empieza a almacenarse la imagen (16384) hasta la posición 16895 se van a encontrar los primeros 512 datos(pixeles), luego en las siguientes 512 posiciones vamos a encontrar datos que no pertenecen a la información de la imagen (16896 – 17407), de igual manera desde la posición 17408 hasta la posición 17920 nuevamente se encuentran almacenados datos(pixeles) correspondientes a la

imagen, y así sucesivamente hasta terminar con la lectura de todos los datos de la imagen dependiendo del tamaño de la imagen.

Debido a que los datos que contienen la información de la imagen se encuentran almacenadas a partir de la posición 18486, los primeros datos que obtendremos no serán 512 sino 458, es decir que se empezará a leer desde la posición 18486 hasta la posición 18944.

5.3. Suavizado de Imagen

El objetivo del suavizado es aumentar o disminuir ciertas características que se encuentran presentes en la imagen, el filtro que hemos usado está basado en la convolución de una máscara y la media aritmética que tiene como fin reducir presencia de ruido, es decir eliminar componente de altas frecuencias y mantener las de bajas frecuencias.

5.4. Procesamiento de Detección de Borde

El procesamiento de detección de bordes se podría decir que es nuestro último paso, antes de mostrar los resultados finales en pantalla, Para realizar este paso de detección nos basaremos en el principio del operador de Sobel, debido a que entre los operadores en procesamiento Digital de Imágenes para la detección de bordes en Imágenes es uno de los más usados.

Una vez realizado este paso, procedemos a ejecutar nuestro proyecto en el procesador NIOS II de la FPGA.

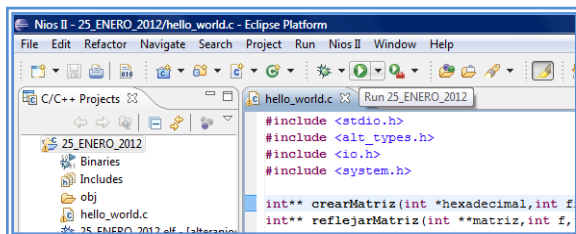


Figura 6 Compilación de Proyecto

6. Resultados

6.1. Matriz Resultante y Visualización de la Imagen

Una vez obtenida la matriz resultante con la información de la imagen con los bordes detectados gracias al Operador de Sobel, procedemos a mostrar la imagen por medio de la salida VGA de la tarjeta DE2 ALTERA.

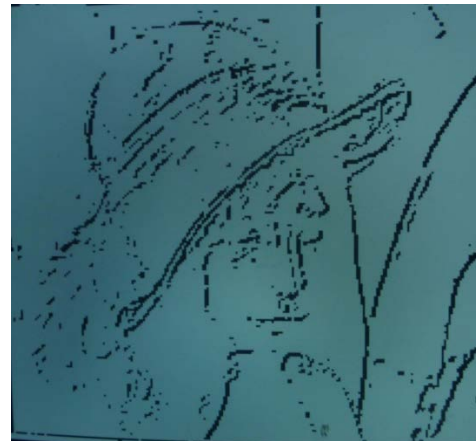


Figura 7 Imagen Resultante

7. Conclusiones

1. Se pudo desarrollar un programa para NIOS II que facilitó la respectiva visualización y comparación de resultados en las distintas etapas del proyecto.
2. El formar nuevas matrices para poder trabajar con la detección de bordes fue más que necesario, debido a que el manejo de datos desde la sdram donde se encontraban inicialmente daba dificultades por la forma como se encontraba almacenada.
3. La calidad de la imagen resultante no es tan precisa debido a que se está usando imágenes en escala de grises formato BMP de 256 colores, lo que indica que se están usando 8 bites por pixel .
4. La unidad de almacenamiento SDRAM fue una gran limitación en el desarrollo de nuestro proyecto debido a su capacidad de almacenamiento para guardar datos, siendo esta una de las razones principales por la cual se usaron imágenes de formato BMP de 8 bits (256 colores).
5. El uso del suavizado de imagen en conjunto con el umbral, nos fue de suma ayuda para poder eliminar ligeramente el ruido que se presentaba en la imagen, logrando así de obtener de una manera un poco más precisas los bordes de imágenes.

8. Agradecimientos

Agradecemos a cada uno de nuestros maestros que han contribuido de una u otra manera para la realización de este trabajo, y de igual manera agradecemos a nuestros familiares, amigos y compañeros de clase.

9. Referencias

- [1] Altera, DE2 Development and Education Board Altera, http://www.altera.com/education/univ/materials/boards/de2/unv-de2-board.html?GSA_pos=1&WT.oss_r=1&WT.oss=e2, Mayo 2011.
- [2] Altera, DE2 Development and Education Board Altera, ftp://ftp.altera.com/up/pub/Webdocs/DE2_introduction.pdf, Mayo 2011.
- [3] Nios II C2H Compiler User Guide, www.altera.com/.../ug_nios2_c2h_compiler.pdf, Abril 2011.
- [4] Vargas Javier, Mohammed Gharbi y Lucas Alejandro, Un nuevo algoritmo de detección de bordes en imágenes con ruido Gaussiano, <https://portal.us.es/pid/public/default/grupo/ver/id/30>, Junio 2011.
- [5] Universidad de Sevilla, Algoritmos genéticos: Detección de bordes de imágenes, www.sav.us.es/formaciononline/textosz/genbordes03/presenta.ppt, Junio 2011.
- [6] Asociación Española de Teledetección, Detección de bordes en imágenes SST, www.aet.org.es/congresos/ix/Lleida126.pdf, Junio 2011
- [7] Bravo Ignacio, Arquitectura basada en FPGA's para la detección de objetos en movimiento, utilizando visión computacional y técnicas PCA, http://dSPACE.uah.es/dSPACE/bitstream/handle/10017/1381/tesis_ignacio_bravo.pdf?sequence=1, Julio 2011.
- [8] Mosquera Oscar Leonardo, Convolución con máscaras para detectar bordes, <http://es.scribd.com/doc/37184885/Detectar-Bordes-de-una-Imagen>, Julio 2011
- [9] Valverde Rebaza Jorge, Detector de Bordes de Sobel en Matlab, <http://jc-info.blogspot.com/2011/06/detector-bordes-sobel-codigo-matlab.html>, Julio 2011.
- [10] Interactive and Cooperative Technologies Lab, Formatos de Imagen, <http://ict.udlap.mx/people/raulms/avances/formatos.html>, Agosto 2011
- [11] Aula Clic SL, Formatos de Imagen, http://www.aulaclik.es/html/a_5_1_1.htm, Agosto 2011
- [12] Wikipedia, Eclipse (software) http://es.wikipedia.org/wiki/Eclipse_%28software%29, Agosto 2011
- [13] Altera, Nios II processor performance benchmarks data sheet, http://www.altera.com/literature/ds/ds_nios2_perf.pdf, Agosto 2011
- [14] Ramblingbytes World, Eclipse Ganymade, http://ramblingbyte.wordpress.com/2008/07/09/pdt-on-ganymade/ganymade_splash/, Diciembre 2011