

Procesamiento de Vídeo en Tiempo Real Utilizando FPGA

J. G. Velásquez-Aguilar, A. Zamudio-Lara

Centro de Investigación en Ingeniería y Ciencias Aplicadas, Universidad Autónoma del
Estado de Morelos,

Cuernavaca, Morelos, 62209 México

Tel.: 01 (777) 329 7084

e-mail: jgpeva@uaem.mx, azamudio@uaem.mx

Palabras Clave: Vídeo en FPGA, Procesamiento en Tiempo Real

RESUMEN

En este trabajo se describe la implementación de algoritmos de procesamiento en tiempo real, utilizando herramientas de programación para manejo de video un chip FPGA. Los resultados del procesamiento pueden ser observados en tiempo real en una pantalla.

1. INTRODUCCIÓN

En la última década, los algoritmos de procesamiento digital de imagen han alcanzado un grado de optimización bastante alto. En muchas áreas se utilizan algoritmos como la FFT (Fast Fourier Transform), Transformada Coseno (para compresión de imagen JPEG), Transformada Wavelets entre otras. Estos algoritmos cada vez requieren menos cálculos en su implementación, pero para los μP (Microprocesadores) utilizados en las computadoras convencionales, la implementación de estos aun resulta una tarea con mucha demanda de computo. El procesamiento en paralelo es una solución alternativa para este problema, pero esto implica el uso de varias computadoras o en su caso grandes computadoras, lo cual repercute en un elevado costo para realizar dicha tarea.

En aplicaciones en tiempo real, dispositivos tales como DSP (Digital Signal Processor) y FPGA (Field Programmable Gate Array), han incrementado sus capacidades en diferentes parámetros tales como velocidad, memoria, numero de compuertas, etc., a la vez que se ha disminuido su costo y tamaño [1,2]. Debido a la funcionalidad y flexibilidad de estos dispositivos, cada vez es más factible desarrollar sistemas para aplicaciones de procesamiento de imagen, cuyos resultados pueden ser observados en tiempo real. Junto con el incremento de capacidad de los dispositivos, las herramientas de programación de ellos han evolucionado, a grado tal que hoy en día se puede programar un chip utilizando entornos visuales cuyo lenguaje de programación ya no es ensamblador sino lenguaje C. Entornos de programación como DK4 [3] de Celoxica para programar FPGA proporcionan utilerías que permiten desarrollar sistemas de procesamiento en tiempo real, utilizando como lenguaje de programación Handel-C.

En este trabajo se presenta la implementación basada sobre un FPGA de algoritmos para procesamiento de video, utilizando un entorno de programación donde las

instrucciones son escritas en Handel-C. Se desglosan las etapas de desarrollo para la implementación del sistema.

La sección 2 describe se describe el sistema con FPGA en el cual se implementará el procesamiento en tiempo real, así como las etapas de procesamiento de la plataforma de desarrollo. La sección 3 contiene los detalles de implementación de algunos algoritmos y los resultados obtenidos.

2. TARJETA DE DESARROLLO FPGA

La implementación en diferentes áreas con FPGAs, ha dado una gran independencia a los sistemas en aspectos como a bancos de memoria RAM (Random Access Memory) y altas velocidades de reloj, entre otros. La plataforma de desarrollo utilizada es la RC203 de Celoxica, contiene un XC2V3000 Virtex II FPGA de Xilinx, con 3'000,000 de compuertas. La Figura 1 muestra la arquitectura de esta plataforma.

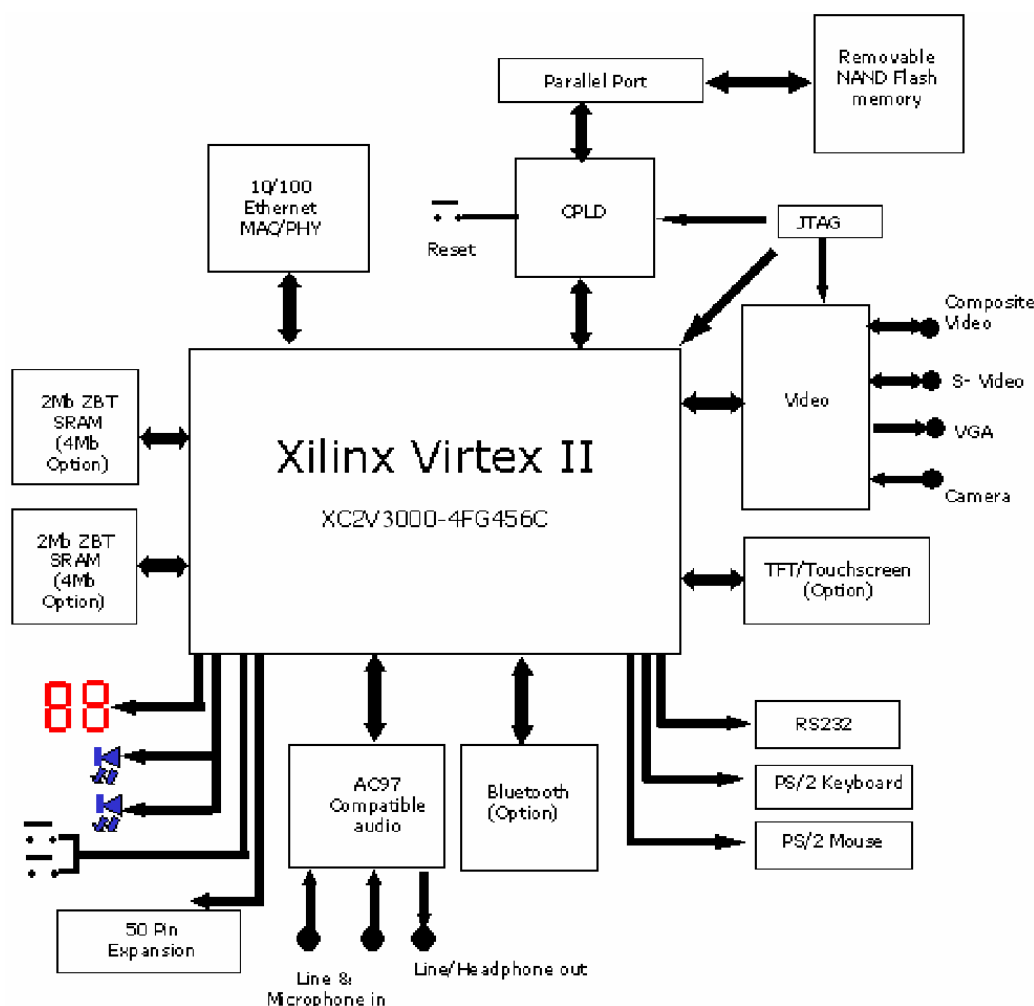


Figura 1. Arquitectura de la tarjeta de desarrollo de la RC203 de Celoxica [3].

El sistema utiliza programación en lenguaje Handel-C para definir la arquitectura del FPGA. Handel-C es un lenguaje que, a diferencia de lenguaje C soporta paralelismo y flexibilidad en el manejo del tamaño de los datos. El DK4 Design Suite es el software de desarrollo el cual provee un ambiente visual para implementar la metodología de diseño del sistema. DK4 contiene rutinas predefinidas para implementación de bloque funcionales a nivel arquitectura tales como RAM read/write, PixelStream, etc., [3]. La Figura 2 muestra el flujo de desarrollo para la implementación del sistema en el chip FPGA [4].

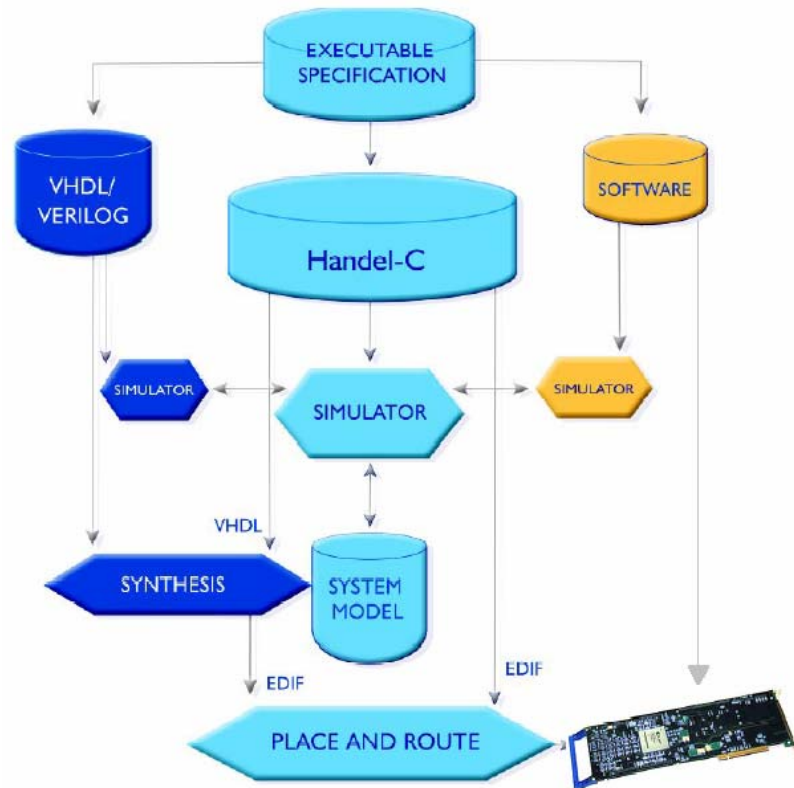


Figura 2. Diagrama de flujo del diseño en sistemas FPGA.

El código generado en *Handel-C* es validado y compilado por el DK, trasladándolo a formato sintetizado cuya extensión es *.edf*, el cual es una lista de conexiones. Una vez sintetizado el diseño se procede a la implementación, para ello se utiliza Xilinx ISE 7.1[1]. La implementación se realiza como se muestra en la Figura 3 [1]. Una vez realizadas estas etapas se obtiene un archivo con la configuración para el FPGA, el cual tiene una extensión *.bit* el cual se descarga al chip. El DK4 cuenta además con una herramienta de simulación la cual permite la evaluación del diseño y proporciona un rápido desarrollo del prototipo.

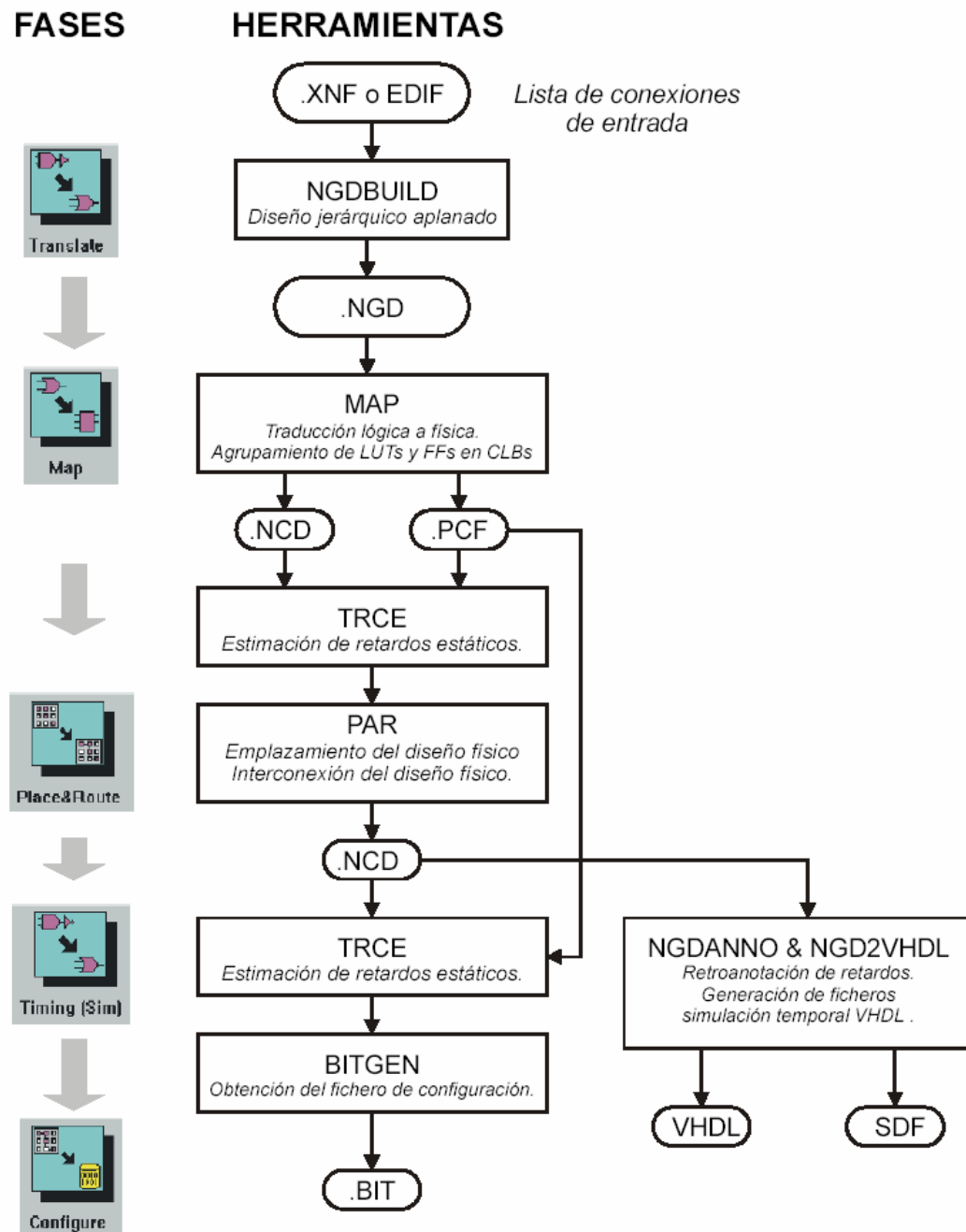


Figura 3. Diagrama de Flujo del proceso de Implementación con Xilinx ISE.

3. IMPLEMENTACIÓN, PRUEBAS Y DISCUSIÓN

Para la realización de los algoritmos se utilizó la herramienta PixelStream del DK4 Design Suite, la cual tiene un entorno de trabajo como lo muestra la Figura 4. El algoritmo utilizado para probar el sistema fue un filtro Sobel.

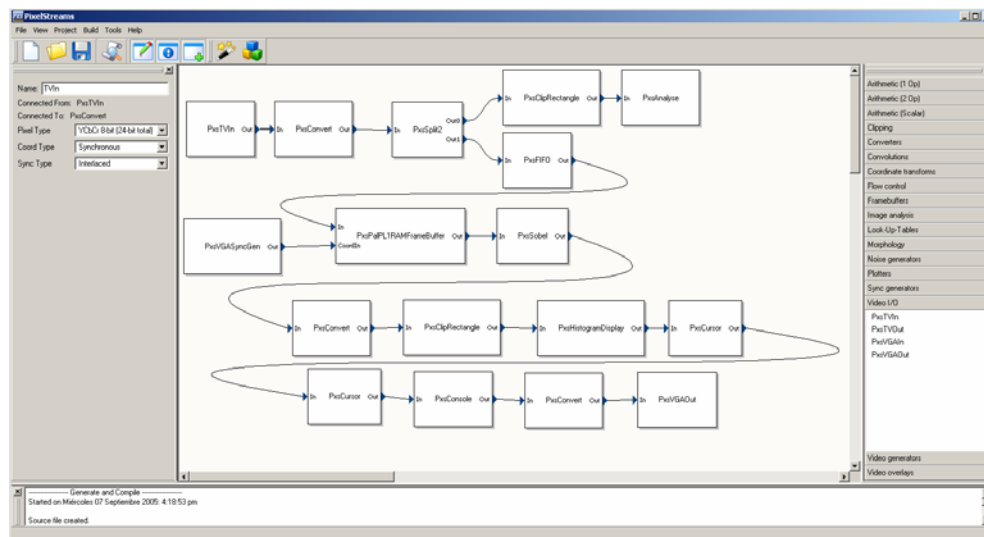


Figura 4. Entorno de Trabajo con la Herramienta PixelStream del DK4 Design Suite.

Los filtros Sobel utilizados fueron los siguientes:

$$\begin{array}{ccc} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{array}$$

$$\begin{array}{ccc} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{array}$$

Una vez filtrada la imagen se obtiene el histograma de la imagen filtrada y los siguientes parámetros:

Pixeles Activos
 Suma de los tonos
 Valor Mínimo
 Valor Máximo
 Valor Medio
 Valor Moda

El diagrama de flujo del código es el que se muestra en la Figura 5. Cada bloque funcional es utilizado para realizar operaciones en tiempo real por lo que se utiliza una arquitectura paralela.

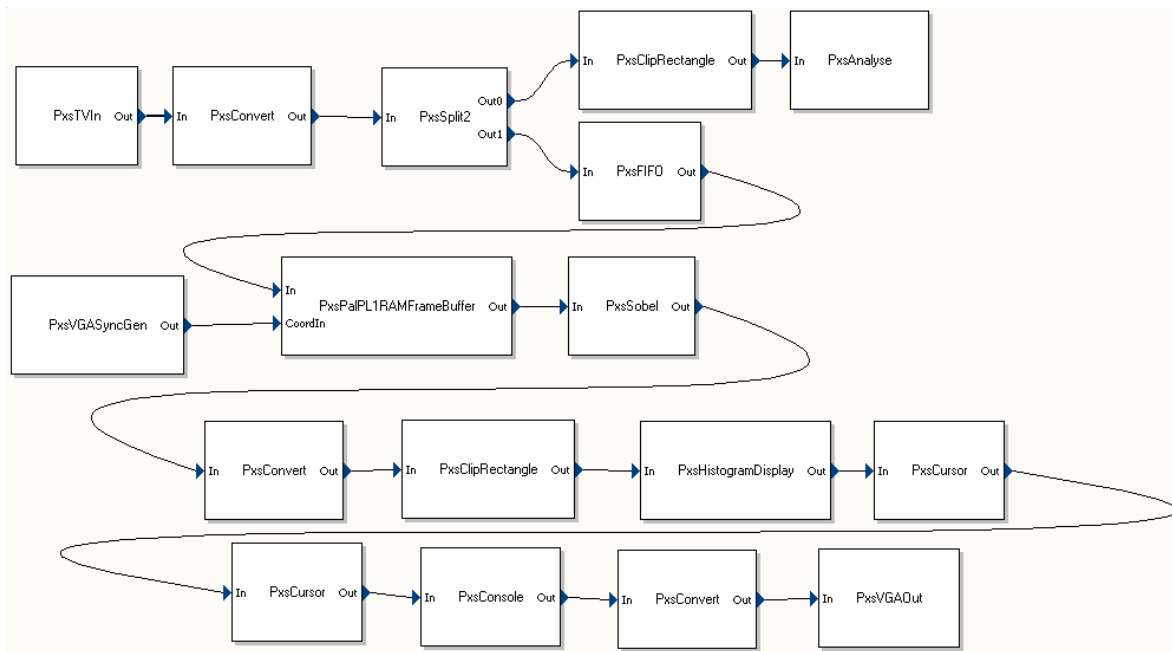


Figura 5. Diagrama de Flujo de los bloques para el procesamiento.

Los resultados obtenidos se muestran en la Figura 6. La Figura 6a nos muestra la imagen normal que capta la CCD, mientras que en la Figura 6b se muestra la imagen procesada según el diagrama anterior.



a)



b)

Figura 6. a) Imagen original, b) Imagen procesada en tiempo real aplicándole un filtro Sobel.

En la Figura 7, se muestra la tarjeta RC300, a la cual se le ha conectado la CCD como imagen de entrada y se muestra la salida ya procesada en tiempo real.

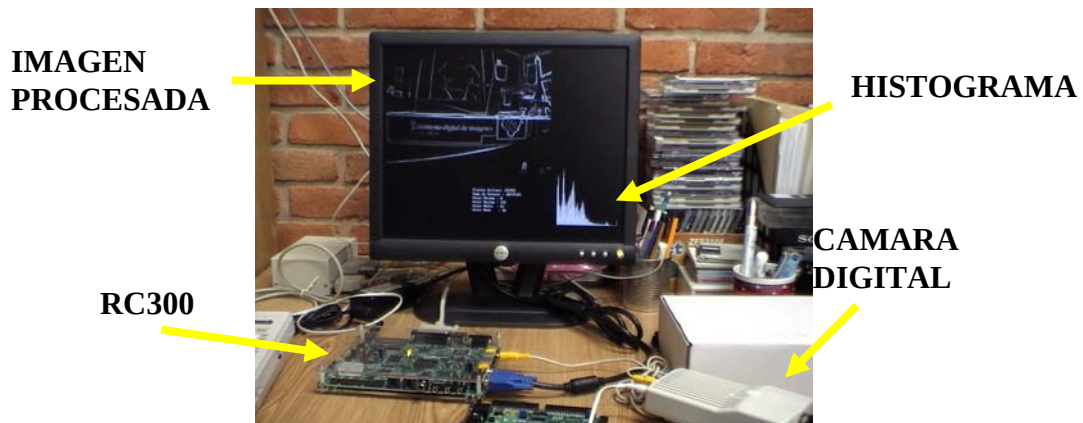
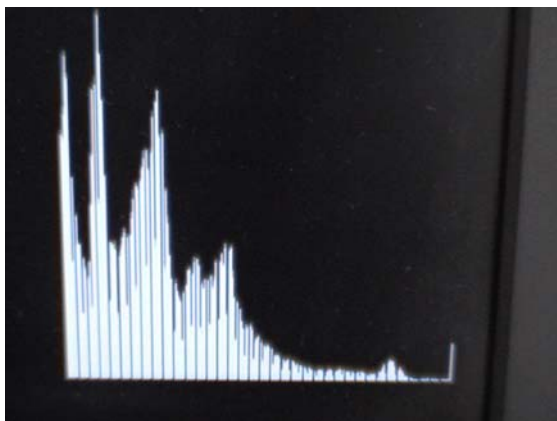
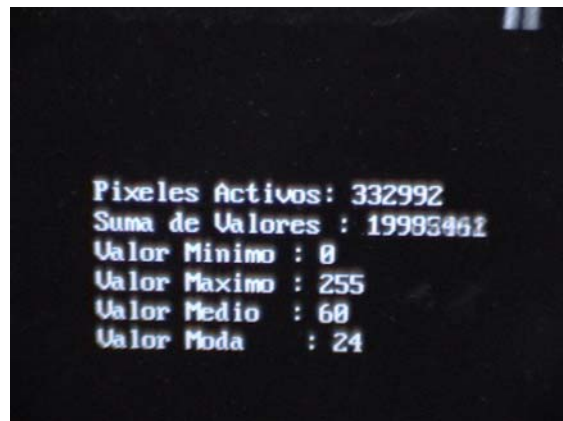


Figura 7. Sistema Implementado para procesamiento en Tiempo Real.

En la Figura 8a, se aprecia con mayor detalle el Histograma de la imagen adquirida; al ir moviendo la cámara y tomando diferentes figuras se aprecia una variación en las barras de dicho histograma. Además, en la Figura 8b se muestran los parámetros analizados en la imagen, mencionados anteriormente.



a)



b)

Figura 8. a) Muestra el Histograma obtenido de la imagen procesada, b) parámetros analizados de la imagen procesada en Tiempo Real.

4. CONCLUSIONES

Se ha implementado un sistema de procesamiento en Hardware mediante el uso de un FPGA. Lo cual representa un procesamiento en Tiempo Real sin la necesidad de un sistema de cómputo sofisticado y caro.

AGRADECIMIENTOS

Este trabajo ha sido apoyado por el CIICAp, el cual ha prestado sus instalaciones y equipo para el desarrollo de este proyecto. Además, se agradece al CONACYT por la beca otorgada.

Referencias

- [1] Xilinx Inc., <http://www.xilinx.com>
- [2] Texas Instruments Inc., <http://www.ti.com>
- [3] Celoxica Inc., <http://www.celoxica.com>
- [4] S. Chappell, C. Sullivan, Handel-C for co-processing & co-design of Field Programmable System on Chip, Celoxica Ltd. Oxford UK