CIINDET 2011

IX Congreso Internacional sobre Innovación y Desarrollo Tecnológico,
23 al 25 de noviembre de 2011, Cuernavaca Morelos, México.

Metodología para Almacenamiento de Imágenes en Memorias externas de tipo RAM, empleando FPGA.

C. A. Ramos Arreguín, J. C. Moya Morales, J. M. Ramos Arreguín, J. C. Pedraza Ortega, M. A. Aceves Fernández, J. E. Vargas Soto, S. Tovar Arriaga.

Resumen: En el presente trabajo se propone una metodología para almacenar diversas imágenes en memorias externas de tipo RAM (Memorias de Acceso Aleatorio), y procesarlas en FPGA (Arreglo Programable de Compuertas en Campo), para exhibir los resultados en un monitor. Se presentan descripciones generales del proyecto, ya que la aplicación puede variar dependiendo la necesidad del usuario.
Palabras Clave: Metodología, FPGA, RAM.

Abstract: In the present work, a methodology to storage different images in RAM (Random Access Memory) is proposed, also the image processing in the FPGA is shown (Field Programmable Gate Array), later, the results can be observed on a display. General descriptions of the work are presented, because the application could vary depending on the needs of the user.
Keywords: Methodology, FPGA, RAM.

Introducción

En la actualidad, en el área de procesamiento de imágenes, el uso de FPGA, es cada vez más frecuente debido a las ventajas que conlleva, como lo son: portabilidad de código, bibliotecas de código reutilizables, herramientas de programación baratas (algunas gratuitas, ISE de Xilinx, versión Web Pack), bajo costo, reconfigurabilidad, procesamiento paralelo, capacidad de interactuar con interfaces de alta o baja velocidad, protección y reutilización de la propiedad intelectual [1]. Así mismo, al trabajar en el área de procesamiento digital de imágenes, se requiere guardar la imagen adquirida, la cual es procesada por el sistema. Así mismo, los resultados de algunas operaciones realizadas a la imagen, requieren ser almacenados como resultados intermedios de otras operaciones de procesamiento de imágenes, como filtros pasa alta y pasa baja. En el FPGA es posible guardar una imagen, la desventaja es que el almacenamiento llega a ocupar el 100% de los recursos del FPGA, dependiendo del encapsulado y la resolución de la imagen. Esto tiene el inconveniente de que la síntesis de la aplicación requiere demasiado tiempo (aprox. 5 hrs.) para la generación del archivo .bit para la programación del dispositivo [2]. Debido a lo anterior, surge la necesidad de realizar un almacenamiento externo de manera temporal utilizando memorias de tipo RAM.

En la actualidad, existen diversos trabajos de procesamiento de imágenes en FPGA, empleando memorias RAM para el almacenamiento temporal de las imágenes, empleando librerías de fabricantes de tarjetas de desarrollo como es el caso de MemUtil de Digilent [3]. Otra manera, es desarrollar un software para el envío de la información al FPGA y a la memoria RAM, o bien, empleando una interfaz con una cámara para la adquisición de la imagen. Como en algunos trabajos realizados con anterioridad, donde se emplean memorias de tipo RAM para almacenar resultados y analizarlos. Sin embargo no muestran una metodología para el control de la interfaz con la memoria [4-8], además las tarjetas empleadas en algunos casos, tienen un núcleo FPGA de la familia VIRTEX de Xilinx, Cyclone de Altera, los cuáles, son de costo alto y poseen gran capacidad para el procesamiento de imágenes debido a sus características, a su vez[9-13].

Este trabajo se enfoca, en la propuesta de una metodología, para el almacenamiento de imágenes en una memoria RAM y ser leídas y exhibidas en monitor empleando FPGA, únicamente utilizando los estándares del IEEE. Ya que hoy en día no se cuenta con una metodología general para la utilización de memorias RAM, librerías estándar y libres, ya que las existentes tienen un costo adicional, si se desea utilizar módulos ya existentes, las cuáles, son desarrolladas por las empresas fabricantes de tarjetas de desarrollo o FPGA.

Metodología General de Diseño de Sistemas Embebidos

En la figura 1, se muestran las etapas de una metodología general empleada para el diseño de sistemas embebidos [1].

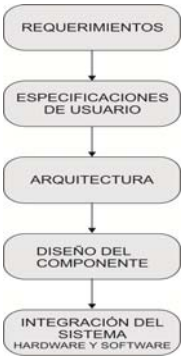


Figura 1. Metodología general empleada para Diseño de Sistemas Embebidos

A continuación se describe cada etapa de la metodología de la figura 1:

Requerimientos: Se refiere a todo requerimiento funcional y no funcional, como lo pueden ser: tamaño, peso, consumo de energía y costo.

Especificaciones de Usuario: Detalles de la interfaz de usuario junto con operaciones necesarias para satisfacer la petición de usuario.

Arquitectura: Hardware (procesador, periféricos, lógica programable y ASSPs) y Software (programas de mando y sus operaciones).

Diseño del Componente: Componentes prediseñados, componentes modificados y nuevos componentes.

Integración del Sistema: Esquema de verificación para encontrar errores rápidamente.

Basándose en la metodología anterior, se propone en este trabajo una metodología para almacenar información de distintas imágenes en memorias de tipo RAM, utilizando el estándar del IEEE para el lenguaje descriptivo de hardware VHDL.

Metodología Propuesta

En la figura 2 se ilustra el diagrama de la metodología propuesta en este trabajo.

Ing. Carlos Alberto Ramos Arreguín. Facultad de Informática U.A.Q., Querétaro Qro, México. (cramos06@alumnos.uaq.mx).
Ing. Juan Carlos Moya Morales. Facultad de Informática U.A.Q., Querétaro Qro, México. (moyajc@gmail.com).
Dr. Juan Manuel Ramos Arreguín. Facultad de Informática U.A.Q., Querétaro Qro, México. (jramos@mecamex.net).
Dr. Jesús Carlos Pedraza Ortega. Facultad de Informática U.A.Q., Querétaro Qro, México. (caryoko@yahoo.com).
Dr. Marco Antonio Aceves Fernández. Facultad de Informática U.A.Q., Querétaro Qro, México. (marco.aceves@uaq.mx).
Dr. José Emilio Vargas Soto. Facultad de Informática U. A. Q., Querétaro Qro., México. (emilio@mecamex.net).
Dr. Saúl Tovar Arriaga. Facultad de Informática U. A. Q., Querétaro Qro, México. (saulotov@yahoo.com.mx).

Figura 2. Metodología propuesta

Adquisición de la Imagen: esta etapa comprende en obtener la imagen requerida para procesar. La adquisición puede ser desde una cámara digital o bien, una imagen almacenada en un PC.

Interfaz de Transferencia: Comprende la interfaz utilizada para el envío de los datos, de la etapa de adquisición de la imagen a la etapa de almacenamiento, a través del FPGA. La interfaz puede ser un sistema digital para el procesamiento de la información de la imagen proveniente de una cámara digital, o bien, vía puerto RS232, cuando la información de la imagen es enviada desde una PC. Para este proyecto se usa el puerto RS232.

FPGA: Interfaz de control para el almacenamiento en memoria RAM y procesamiento de la imagen.

Almacenamiento: Se almacenan los datos de cada pixel de la imagen recibida de las etapas anteriores, o bien, se almacenan los resultados obtenidos en la etapa de arquitectura y cuando se requiera se enviarán los datos guardados a la etapa anterior, para llevar a cabo el procesamiento deseado.

Exhibición: Se muestran los resultados obtenidos en un monitor.

Memorias de tipo RAM

La memoria de acceso aleatorio (RAM), es usada en un sistema digital para almacenamiento masivo de información o datos, una célula RAM es mucho más simple que una célula FF (Flip Flop). El tipo de memoria comúnmente empleado es el SRAM (asynchronous static RAM). Para un sistema síncrono el acceso directo a una memoria SRAM es complicado, para esto es necesario implementar un *controlador de memoria* para la interacción entre el FPGA y la memoria, el cual es activado desde el sistema síncrono principal, el cual genera señales correctamente medidas en tiempo para accesar a la memoria, ya sea para

escritura o lectura. El controlador brinda al sistema principal el tiempo detallado y hace que el acceso a memoria parezca una operación síncrona [8]. Existen diversos tipos de memorias RAM, sin embargo, el funcionamiento básico es el mismo para todos, cambiando algunas características que las hacen más eficientes o con mayor funcionalidad.

Desarrollo del Sistema

En la figura 3, se muestra una descripción para un controlador genérico de una memoria RAM, el cual se compone de tres módulos, los cuáles son:

Divisor de frecuencia, consiste en generar un pulso cada el tiempo necesario para cumplir con los tiempos de escritura/lectura de la memoria.

Máquina Secuencial, donde se determina la operación y el tiempo de reposo para que se lleve a cabo cada operación, figura 4.

RAM: a este módulo llegan los datos a ser almacenados y la dirección donde se almacenarán, el bus de salida HAB, agrupa 5 señales de habilitación, que son: *escritura*, *lectura*, *dispositivo* (*chip select*), *byte más significativo* y *byte menos significativo*.

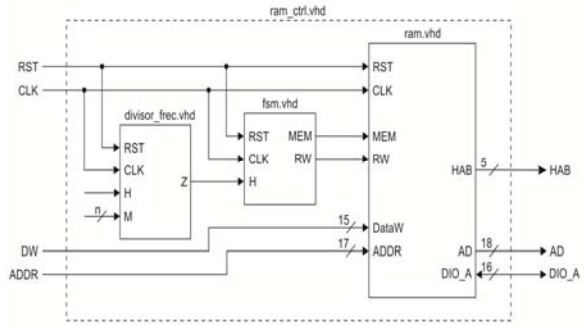


Figura 3. Diagrama de bloques del control para utilizar la memoria RAM.

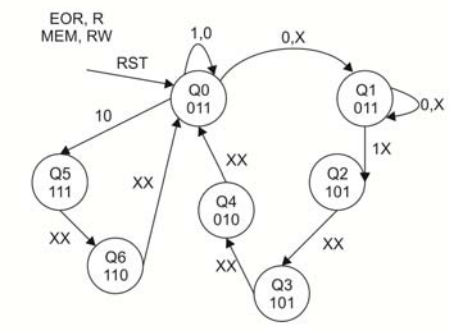


Figura 4. Maquina secuencial del controlador RAM

El módulo controlador, figura 5, funciona en conjunto con el módulo de adquisición de la imagen. El módulo de adquisición puede utilizar cualquier protocolo como RS232, USB, directamente de una cámara digital, entre otros.

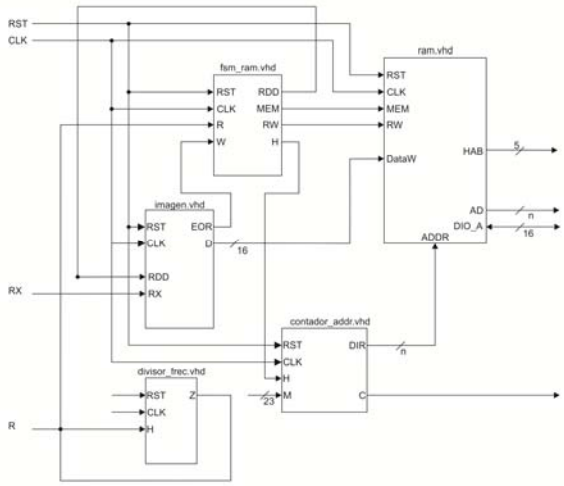


Figura 5. Control SDRAM incorporado con la interfaz de recepción de datos vía RS232

En el módulo *imagen.vhd* de la figura 5, se reciben los datos correspondientes a la imagen, la señal EOR indica la finalización de la recepción de los datos y envía una señal de habilitación para la escritura en el módulo *fsm_ram.vhd*. Así mismo, este módulo habilita al módulo *contador_addr.vhd*, en el cual se incrementa la dirección para almacenar el siguiente dato. La señal C indica cuando la última dirección de la memoria ha sido escrita, es decir, ya no se cuenta con más direcciones para realizar almacenamiento de datos.

En el diagrama de la figura 6, se describe el sistema para almacenamiento de imágenes y su exhibición. Donde se integran los módulos *vgasync.vhd* y *vga_rgb.vhd*, propuestos en trabajos previos [2], [7]. El módulo *procesos.vhd*, consta de las operaciones que requiera aplicar a la imagen, como lo son: escala de grises, binarización, filtros pasa baja y pasa alta, entre otros. También se utiliza el módulo de control RAM, descrito anteriormente en la figura 7, y es modificado para almacenar en distintas direcciones de la memoria.

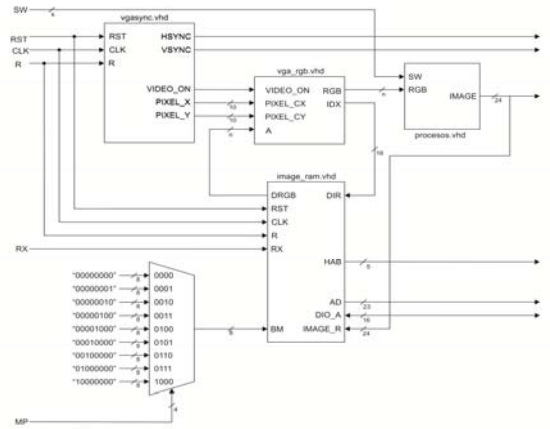


Figura 6. Descripción del sistema para almacenamiento de imágenes y su exhibición.

Tarjeta de Experimentación Nexys 2

En la figura 7 se muestra la tarjeta empleada en el proyecto y sus características son las siguientes [14]:

- FPGA Spartan 3E de Xilinx XC3SE1200 FG320.
- PSDRAM Micron 16 MB (128 Mb).
- Frecuencia de reloj 50 MHz.
- Puerto VGA de 8 bit.
- Comunicación RS232.

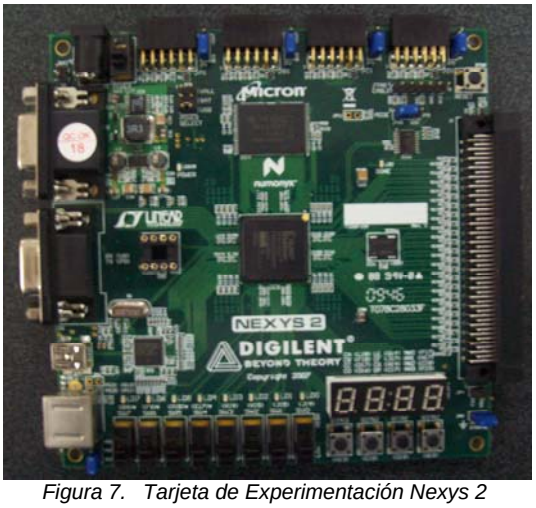


Figura 7. Tarjeta de Experimentación Nexys 2

El dispositivo RAM, empleado en este proyecto es una SDRAM MICRON MT45W8MW16BGX, el cual es un dispositivo CMOS de alta velocidad, memoria de acceso pseudo-estática desarrollada para aplicaciones portables de baja potencia, puede operar como una SRAM asíncrona típica. Este dispositivo contiene un núcleo SDRAM de 128 Mbit, organizado con 8M direcciones cada una de 16 bit para almacenamiento de datos. En la figura 8, se muestra el diagrama de bloques del funcionamiento interno de la memoria [15].

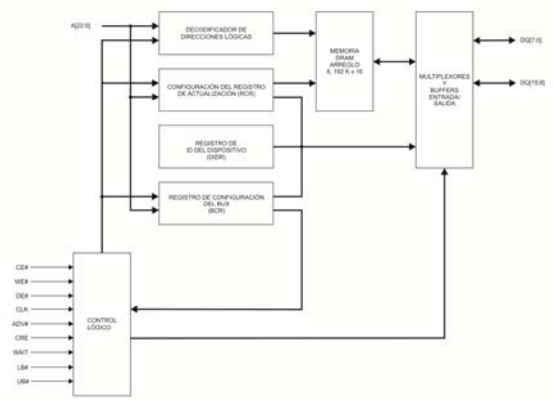


Figura 8. Diagrama simplificado de la operación de la memoria SDRAM de la tarjeta Nexys 2.

La operación de lectura (Read) es iniciada con un estado '0' en las señales CE (chip enable), OE (output enable), LB (Lower Byte), UB (Upper Byte) y WE (Write Enable) en estado '1'. Los datos son enviados después del acceso en el tiempo especificado, el ciclo de trabajo se presenta en la figura 9.

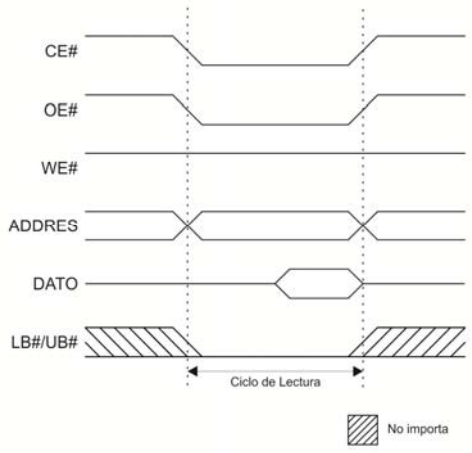


Figura 9. Ciclo de lectura

La operación de escritura (Write), ocurre cuando CE, WE, LB y UB, se encuentran en estado '0' y OE en estado '1'. ADV puede estar en estado '0' para cualquiera de las dos operaciones. Su ciclo de trabajo se ilustra en la figura 10.

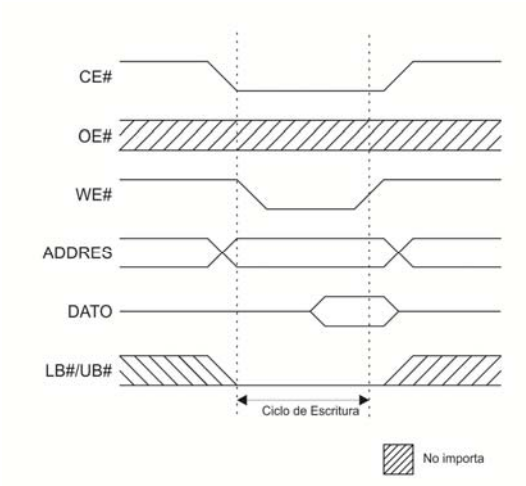
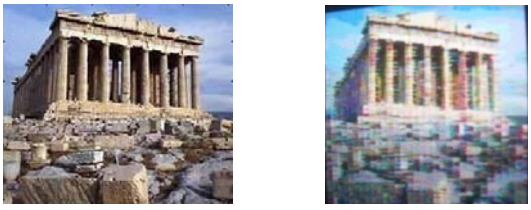


Figura 10. Operación de Escritura

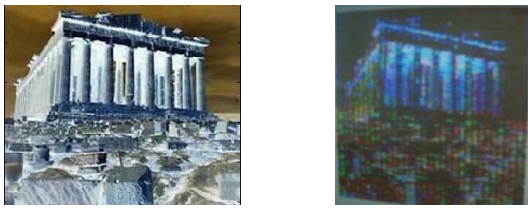
Pruebas y Resultados

Para implementar el sistema en la tarjeta de experimentación es necesario adaptar el módulo de mapeo de las direcciones dependiendo de la cantidad de imágenes que requieran ser almacenadas. En este trabajo se almacenan: imagen original, escala de grises, negativo de color y binarización. Por lo que el multiplexor debe ser ajustado para 4 imágenes. Así mismo, la imagen es normalizada de 24 bit de color a 8 bit [2, 7]. En las figuras 11 a 14, se muestran los resultados de las imágenes, reproducidas en FPGA y exhibidas en un monitor de tipo CRT.



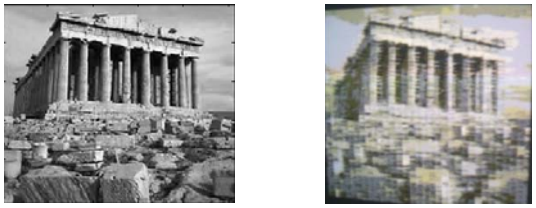
a) b)

Figura 11. a) Imagen original a 24 bit, b) Imagen normalizada almacenada en la memoria SDRAM



a) b)

Figura 12. a) Negativo de color, obtenido en software, b) Resultado en Hardware



a) b)

Figura 13. Escala de Grises, obtenida en software, b) Resultado en hardware

a) b)

Figura 14. Imagen Binarizada, obtenida en software, b) Binarización en FPGA.

Cabe mencionar que la diferencia o pérdida de calidad en las imágenes reproducidas en el hardware, es debido a la normalización efectuada antes de almacenar la primera imagen en la memoria SDRAM.

Conclusiones y trabajo a futuro

Es factible utilizar memorias externas para almacenamiento de datos, como imágenes, por la gran cantidad de datos que se puede almacenar temporalmente, ya que realizar el almacenamiento directamente en el FPGA, resulta costoso por el alto número de recursos consumidos, y tiempo de síntesis alto. Además, los recursos empleados del FPGA, son mínimos (1%).

La dimensión de prueba de cada imagen es de 256x256 pixeles, es decir, se ocupan 65536 direcciones y en cada dirección se almacenan 8 bit de datos, en la memoria SDRAM para almacenar una imagen.
El ciclo de trabajo de la memoria RAM es especificada por el fabricante, la metodología permite adaptar diferentes ciclos de operación, para diversos dispositivos RAM de distintos fabricantes. Así mismo, puede ser implementada a FPGAs de distintos fabricantes, debido a que únicamente se utilizan librerías estándar del IEEE.
La aportación de este trabajo de investigación es una metodología estándar para el manejo de memorias externas RAM, donde se describe un controlador básico para llevarlo a cabo, el cual, puede ser extendido fácilmente en funcionamiento dependiendo la necesidad del usuario.
Como trabajo a futuro, se propone trabajar con imágenes de 24 bit de color, para obtener una imagen de la misma calidad a la obtenida en software (PC). Implementar algoritmos de procesamiento de imágenes de mayor complejidad, detección de bordes, filtros pasa baja y pasa alta, transformada de Fourier, entre otros.
Desarrollar e implementar un sistema digital, para la adquisición de imágenes provenientes de una cámara digital.

Referencias

[1] Dubey Rahul, Introduction to Embedded System Design Using Field Programmable Gate Arrays, Springer-Verlag London Limited, ISBN: 978-1-84882-015-9, Londres, 2009.

[2] Ramos Arreguín Carlos Alberto, Cora Gallardo Orlando Marcos, Ramos Arreguín Juan Manuel, Pedraza Ortega Jesús Carlos, Canchola Magdaleno Sandra Luz, Vargas Soto José Emilio, Metodología para Manejo de Imágenes en FPGA, 6º Congreso Internacional de Ingeniería, pp. 219-226, ISBN: 978-607-7740-39-1, Querétaro, Qro., Abril 2010.

[3] Software MemUtil de Digilent Incorporated para almacenamiento de datos en memorias de tipo RAM incorporadas en algunas tarjetas de desarrollo, www.digilentinc.com, última consulta 29 de abril 2010.

[4] Quintero M. Alexander, Vallejo R. Eric, Image Processing Algorithms using FPGA, Revista Colombiana de Tecnologías de Avanzada, Vol. 1; No. 7; pp. 11-16, ISSN: 1692-7257, 2006.

[5] Bravo Muñoz Ignacio, Arquitectura basada en FPGA para la Detección de Objetos en Movimiento, utilizando Visión Computacional y Técnicas PCA, Tesis Doctoral, Departamento de Electrónica de la Escuela Politécnica de la Universidad de Alcalá, España 2007.

[6] A. Castillo, J. Vázquez, J. Ortegón, C. Rodríguez, Prácticas de Laboratorio para Estudiantes de Ingeniería con FPGA, IEEE Latin America Transactions, pp. 130-136, Junio 2008.

[7] Ramos Arreguín Carlos Alberto, Moya Morales Juan Carlos, Ramos Arreguín Juan Manuel, Pedraza Ortega Jesús Carlos, Metodología de una Etapa Básica de un Sistema de Procesamiento de Imágenes basado en FPGA, 9º Congreso Nacional de Mecatrónica, pp. 235-240, ISBN: 978-607-95347-2-1, Octubre 2010.

[8] Pong P. Chu, FPGA Prototyping by VHDL Examples Xilinx Spartan 3 Version, Wiley Interscience, ISBN: 978-0-470-18531-5, USA 2008.

[9] Kalomiros J. A., Lygouras J., Design and evaluation of a hardware/software FPGA-based system for fast image processing, Microprocessors and Microsystems, Elsevier, pp. 95 – pp. 106, doi:10.1016/j.micpro.2007.09.001.

[10] Chaikalis D., Sgouros N. P., Maroulis D., A Real Time FPGA Architecture for 3D reconstruction from integral images, Journal of Visual Communication & Image Representation, Elsevier, pp. 9 - pp. 16, doi:10.1016/j.jvcir.2009.09.004.

[11] Siéler L., Tanougast C., Bouridane A., A scalable and embedded FPGA architecture for efficient computation of gray level co-ocurrence matrices and Haralick textures features, Microprocessors and Microsystems, Elsevier, pp. 14 - pp. 24, doi:10.1016/j.micpro.2009.11.001.

[12] Krill B., Ahmad A., Amira A., Rabah H., An efficient FPGA-based dynamic partial reconfiguration design flow and environment for image and signal processing IP cores, Signal Processing: Image Communication, Elsevier, pp. 377 - pp. 387, doi:10.1016/j.image.2010.04.005.

[13] Satake Shin-ichi, Sorimachi Gaku, Masuda Nobuyuki, Ito Tomoyoshi, Special-purpose computer for particle image velocimetry, Computer Physics Communications, Elsevier, pp. 1178 – pp. 1182, doi:10.1016/j.cpc.2011.01.022.

[14] Datasheet Nexys 2, www.digilentinc.com, última consulta 29 de Abril 2010.

[15] Micron Technology, Inc. Datasheet MT45W8MW16BGX, última consulta 29 de Abril de 2010.

Currículo corto de los autores

Carlos Alberto Ramos Arreguín
Ingeniero en Computación egresado de la Facultad de Informática, Universidad Autónoma de Querétaro en 2011. Actualmente es estudiante de Maestría en Ciencias de la Computación en la Facultad de Informática, Universidad Autónoma de Querétaro y su línea de investigación es procesamiento digital de imágenes en hardware.

Juan Carlos Moya Morales
Ingeniero en Computación egresado de la Facultad de Informática, Universidad Autónoma de Querétaro en 2011. Actualmente es estudiante de Maestría en

Ciencias de la Computación en la Facultad de Informática, Universidad Autónoma de Querétaro y su línea de investigación es en procesamiento digital de imágenes.

Juan Manuel Ramos Arreguín

Ingeniero en Electrónica egresado de la Facultad de Ingeniería Mecánica, Eléctrica y Electrónica en 1994, Universidad de Guanajuato. Maestría en Ingeniería Eléctrica en 1996, en la Facultad de Ingeniería Mecánica, Eléctrica y Electrónica, Universidad de Guanajuato. Sus estudios de Doctorado fueron en Ciencia y Tecnología con especialidad en Mecatrónica en el Centro de Investigación y Desarrollo Industrial CIDESI, Querétaro en 2008. Actualmente es profesor-investigador en la Facultad de Informática, Universidad Autónoma de Querétaro y su línea de investigación es sistemas embebidos.

Jesús Carlos Pedraza Ortega

Ingeniero en Electrónica egresado del Instituto Tecnológico de Celaya en 1993. Maestría en Ingeniería Eléctrica en la Facultad de Ingeniería Mecánica, Eléctrica y Electrónica, Universidad de Guanajuato. Sus estudios de Doctorado fueron en Ingeniería Mecánica en la Universidad de Tsukuba, Japón en 2002. Actualmente es profesor-investigador en la Facultad de Informática, Universidad Autónoma de Querétaro y su línea de investigación es en procesamiento digital de imágenes.

Marco Antonio Aceves Fernández

Ingeniero en Telemática egresado de la Universidad de Colima en 2000. Cursó la Maestría y Doctorado en la Universidad de Liverpool, Inglaterra en el 2006. Actualmente es Profesor-Investigador en la Facultad de Informática de la Universidad Autónoma de Querétaro y su línea de investigación es en sistemas embebidos.

José Emilio Vargas Soto

Ingeniero Mecánico con mención honorífica por la Universidad Nacional Autónoma de México. Maestría en Ingeniería en el área de Control por la Universidad Politécnica de Madrid, España. Doctor en Ciencias Físicas, en el área de Informática y Automática por la Universidad Compútense de Madrid. Post-Doctorado por sus investigaciones sobre la locomoción libre de robots caminantes, por The Electrocommunications University of Tokio, Japón. Desde el 2010 a la fecha, es profesor-investigador en la Facultad de Informática, Universidad Autónoma de Querétaro y su línea de investigación es en inteligencia artificial y procesamiento digital de imágenes.

Saúl Tovar Arriaga

Ingeniero en Electrónica egresado del Instituto Tecnológico de Querétaro. Maestría en Mecatrónica por la Universidad de Siegen, Alemania 2006. Sus estudios de Doctorado fueron Biología Humana (equivalente a Ingeniería Biomédica) por el Instituto de Física Médica por la Universidad Friederich-Alexander en Nuremberg, Alemania en 2009. Desde el 2010 es profesor-investigador en la Facultad de Informática, Universidad Autónoma de Querétaro y su línea de investigación es en robótica aplicada a la medicina y procesamiento digital de imágenes.