

Procesamiento de Imágenes en Tiempo Real para Mediciones con Sistemas Ópticos

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

Centro de Investigación en Ingeniería y Ciencias Aplicadas, Universidad Autónoma del Estado de Morelos, Av. Universidad 1001, Col. Chamilpa, Cuernavaca, Morelos, 62209 México. Tel.: +52 01 777 329 7084, e-mail: jgpeva@uaem.mx

Resumen. En este trabajo, se describe la implementación de un sistema en un Field Programmable Gate Array (FPGA) para análisis de cambios de fase temporal en un patrón de franjas de interferencia, con la finalidad de medir desplazamientos en tiempo real, introducidos a un sistema óptico interferométrico. Las imágenes de los interferogramas, se introducen al FPGA a través de una Charge Coupled Device (CCD). Se usa una arquitectura paralela para las etapas de análisis de fase y despliegue en pantalla. El sistema completo consume 2030 capas de un chip XC2V3000 y corre a una frecuencia de reloj de 65 MHz.

Palabras Clave: Procesamiento de Imagen, FPGA, Metrología Óptica.

1. Introducción

En la última década, se han estado utilizando arreglos interferométricos para realizar mediciones de diferentes variables tales como vibración [1, 2, 3], deformación [4] y desplazamiento [5]. En los trabajos realizados se ha utilizado el arreglo interferométrico para adquirir patrones de interferencia, estos patrones son evaluados y en base a los resultados obtenidos se proporciona el valor de la variable en cuestión. Los patrones adquiridos a través de una CCD, se analizan con algoritmos implementados en computadora [6], lo que implica un retraso en la obtención de resultados, ya que los algoritmos utilizados para procesamiento, son una tarea muy grande para un procesador. Para mediciones en tiempo real, se requiere de un dispositivo capaz de efectuar el procesamiento bajo este esquema [7]; una alternativa es el uso de FPGA.

El incremento en la capacidad de los chips FPGA y los avances en optimización de algoritmos para procesamiento proporcionan una solución viable para aplicaciones en tiempo real en el área de procesamiento de imagen en tiempo real [8, 9].

En este trabajo se presenta la implementación basada sobre un FPGA de un sistema para medir desplazamiento en tiempo real. Se utiliza un arreglo interferométrico de Michelson para generar los patrones de interferencia, los cuales son introducidos al sistema mediante una CCD. El algoritmo de procesamiento de imagen es programado en el FPGA y en su implementación, se utilizan varias etapas en paralelo para accele-

ran el procesamiento. El resultado de la medición es mostrado en una pantalla junto con el video de entrada correspondiente al patrón de interferencia.

2. Motivación de la Investigación.

“La interferometría es una técnica óptica ampliamente utilizada para medición de desplazamiento” [6]. Diversas herramientas computacionales que usan algoritmos para la evaluación de patrones de franjas de interferencia, son ejecutadas en computadoras convencionales. Para aplicaciones en tiempo real, el uso de dichas herramientas, requiere grandes sistemas de computo, ya que la tarea de realizar el procesamiento de imagen es muy pesada.

Por otro lado, el desarrollo de las herramientas de programación para FPGA y el incremento en sus capacidades, facilitan cada vez mas la implementación de algoritmos para procesamiento de imagen en tiempo real. Por tanto, utilizamos el FPGA para dar solución al retraso en tiempo de respuesta de los sistemas interferométricos convencionales.

3. Interferómetro de Michelson

En el fenómeno de Interferencia, dos ondas son proyectadas sobre un plano, como se ilustra en la Figura 1 [10].

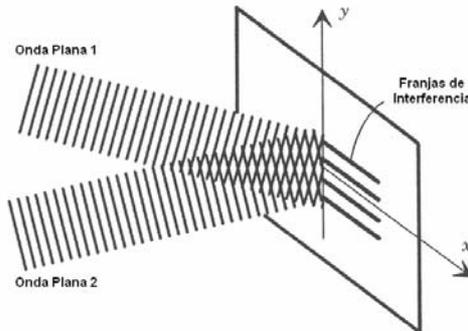


Figura 1. Interferencia de dos ondas planas y patrón generado. En el punto donde las ondas se intersectan se pueden observar un patrón de franjas de interferencia.

Cuando las ondas son superpuestas, la amplitud resultante bajo ciertas condiciones, es la adición de las amplitudes instantáneas de las dos ondas. En el arreglo interferométrico de Michelson mostrado en la Figura 2, las dos ondas I_1 e I_2 son generadas por la misma fuente, esto es, tienen la misma frecuencia y fase. Al reflejarse las ondas en los espejos, si los caminos ópticos son diferentes, existe una diferencia en la fase con un parámetro constante, entonces la intensidad I resultante registrada por la CCD está dada por:

$$I = I_1 + I_2 + 2\sqrt{I_1 I_2} \cos(\delta) \quad (1)$$

donde:

$$\delta = \frac{2\pi(r_1 - r_2)}{\lambda} \quad (2)$$

siendo r_1 y r_2 los caminos ópticos recorridos por las ondas y λ la longitud de onda. De esta forma, utilizando la diferencia de caminos ópticos l , el número de ciclos N observados en el patrón de interferencia constructiva está dado por:

$$N = \frac{2l}{\lambda} \quad (3)$$

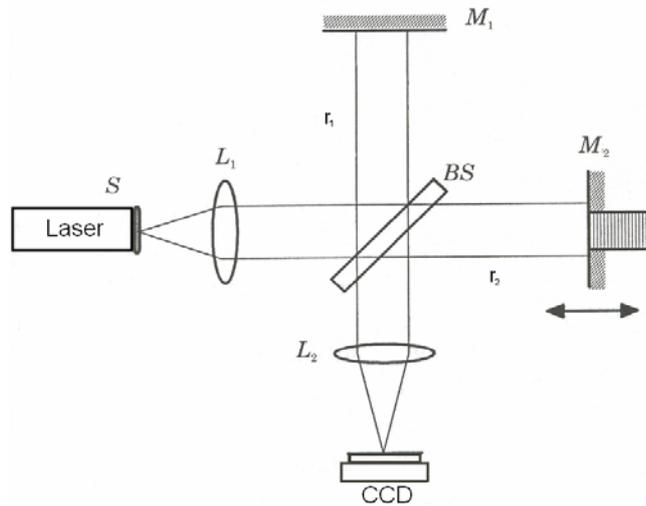


Figura 2. Interferómetro de Michelson. Las ondas son obtenidas usando un divisor de haz BS (Beam Splitter) y son enviadas en diferentes direcciones, las cuales siguen los caminos ópticos r_1 y r_2 , para ser reflejadas por dos objetos planos M_1 y M_2 . La fuente puntual S se coloca a una distancia focal del lente L_1 , la luz de entrada al interferómetro es un haz colimado. El patrón de interferencia es enviado a otro lente L_2 que enfoca en la CCD.

Para el análisis de cambios de fase se obtiene el perfil de una región de franjas y se observa el desplazamiento que presenta al variar uno de los caminos ópticos. El perfil y su correspondiente cambio de fase nos proporcionan el desplazamiento.

4. Hardware y Software para la Implementación

La RC203 de Celoxica [11], contiene un XC2V3000 Virtex II FPGA con 3'000,000 de compuertas. El entorno de programación es DK 4 Suite de Celoxica, utiliza lenguaje Handel-C para definir la arquitectura del FPGA. Handel-C soporta

paralelismo y flexibilidad en el manejo del tamaño de los datos, además de contener librerías predefinidas para implementación tales como RAM read/write, PixelStream, entre otras. El código generado en Handel-C es compilado y trasladado a formato sintetizado de Xilinx [12], para posteriormente ser mapeado en la arquitectura del chip, usando Xilinx Integrated Software Environment (ISE 7.1).

La librería PixelStream permite desarrollar en forma grafica los bloques estructurales del sistema, facilitando la integración de complejas arquitecturas. Al compilarse el proyecto bajo esta librería, se generan los macros en Handel-C, correspondientes a cada bloque funcional, así como las interconexiones entre ellos. Posteriormente, se configuran las variables para cada bloque y se realiza la programación en los macros correspondientes a los algoritmos a implementar en la tarjeta de desarrollo.

5. Implementación, Pruebas y Discusión

El sistema consta de 2 etapas estructuradas como se muestra en la Figura 3.

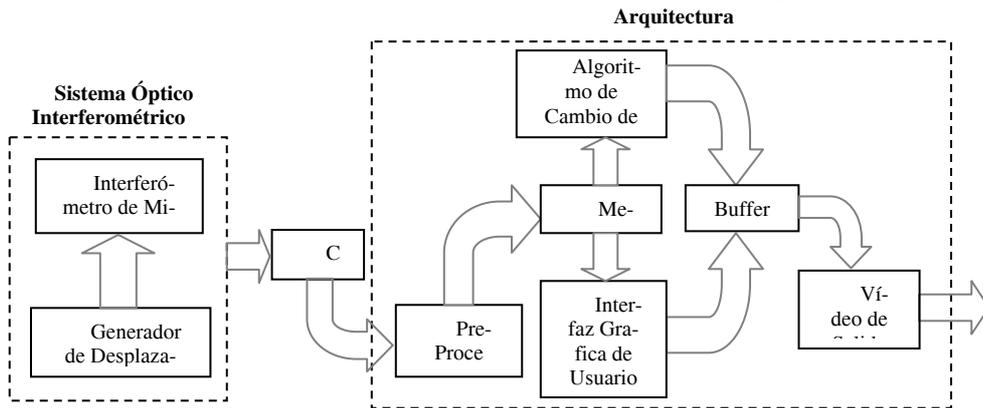


Figura 3. Sistema propuesto.

5.1 Configuración del Interferómetro

Para el sistema óptico interferométrico se utilizó un láser He-Ne con una longitud de onda de 638 nm y una potencia de 10 mW. El desplazamiento se realiza mediante un motor a pasos (23LM-C372, Minebea) el cual desplaza a través de un sistema mecánico uno de los espejos de arreglo. Las imágenes de los interferogramas son enfocadas hacia el sensor CCD NCK41CV que tiene una resolución de 768x494 píxeles como lo muestra la Figura 4.

5.2 Implementación basada en bloques de PixelStream

La Figura 5 muestra el diagrama a bloques del hardware desarrollado con la librería PixelStream, imágenes provenientes de la CCD en formato RGB de tamaño 768x494 son introducidas a la tarjeta de desarrollo. El bloque de procesamiento en el FPGA esta conformado por 13 módulos en arquitectura paralela:

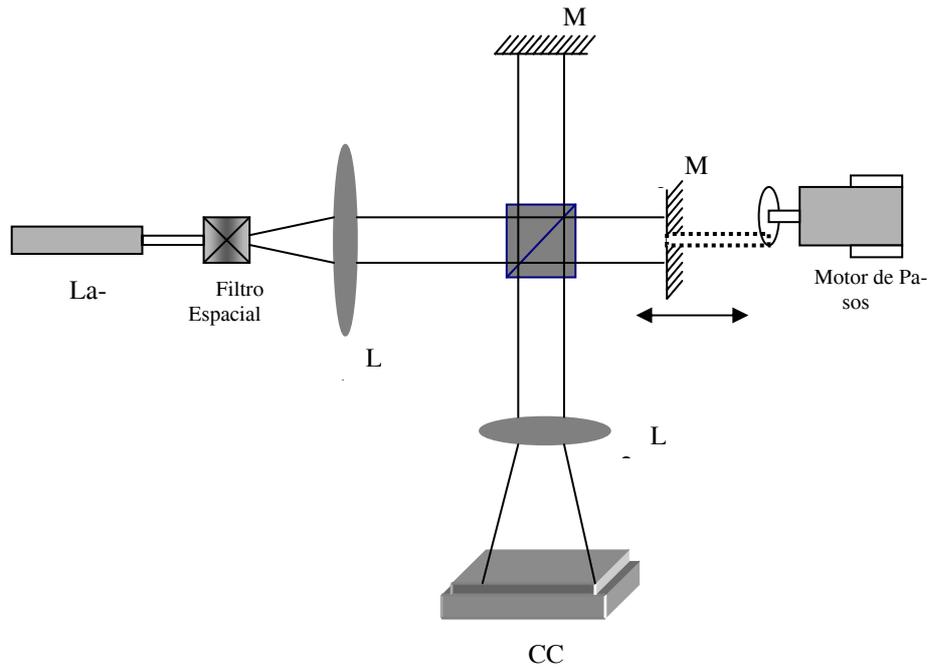


Figura 4. Configuración del sistema óptico para la medición de desplazamiento.

PxsTVIn: El video que proporciona la CCD es introducido a través de la entrada de TV en la tarjeta RC203, el formato a la salida del bloque funcional es RGB 8 bit (24 bits en total).

PxsConvert: El formato RGB es convertido a escala de gris, lo que significa que se tiene una imagen de 768 pixeles de ancho x 494 pixeles de alto x formato de dato de 8 bits para almacenamiento.

PxsSplit2: Divide la señal de entrada en 2 salidas, con la finalidad de analizar y desplegar los cuadros de video al mismo tiempo.

PxsClipRectangle: Establece el área para el análisis de la imagen. El área elegida corresponde a lo zona donde el patrón de interferencia presenta una mayor definición.

PxsAnalyse: En este módulo se realiza el algoritmo de conteo de franjas para obtener el perfil, y se lleva a cabo la medición del cambio de fase entre los cuadro de video consecutivos.

PxsFIFO: Reserva el espacio necesario para el almacenamiento de la imagen en memoria con un formato de dato de 8 bits.

PxsVGASyncGen: Genera pulsos de sincronización para la salida del cuadro de video almacenado en memoria.

PxsPalPLIRAMFrameBuffer: Almacena cuadro de imagen en memoria RAM, utilizando el espacio reservado y las coordenadas generadas para los pixeles. Cada que un cuadro de imagen es completado en la entrada y se enlaza asincrónicamente con las coordenadas.

PxsClipRectangle: Establece el área a desplegar del video de entrada.

PXSPerfilDisplay: Genera una grafica en tiempo real del perfil obtenido en el modulo PxsAnalyse.

PxsConsole: Despliega el desplazamiento obtenido en el análisis del patrón de interferencia.

PxsConvert: La imagen obtenida es convertida de escala de gris (8 bits) a un formato RGB (24 bits) para su despliegue en modo VGA en pantalla.

PxsVGAOut: Escribe los pixeles a la salida VGA de la tarjeta para ser desplegado en Pantalla.

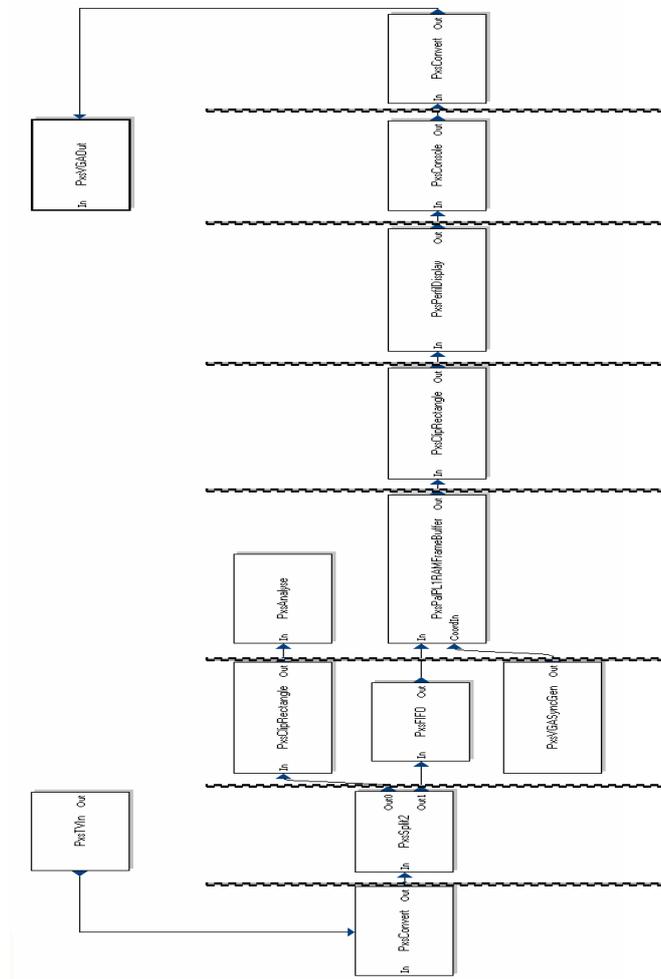


Figura 5. Módulos de la arquitectura hardware del sistema implementado con PixelStream.

Para obtener el perfil del patrón de franjas se usó el algoritmo de “conteo de franjas” [4]. Una vez obtenido el perfil se hace el cálculo del desplazamiento en base al cambio de fase registrado en los cuadros de video consecutivos. La figura 6 muestra el algoritmo utilizado para la obtención del desplazamiento, implementado en el bloque PxsAnalyse.

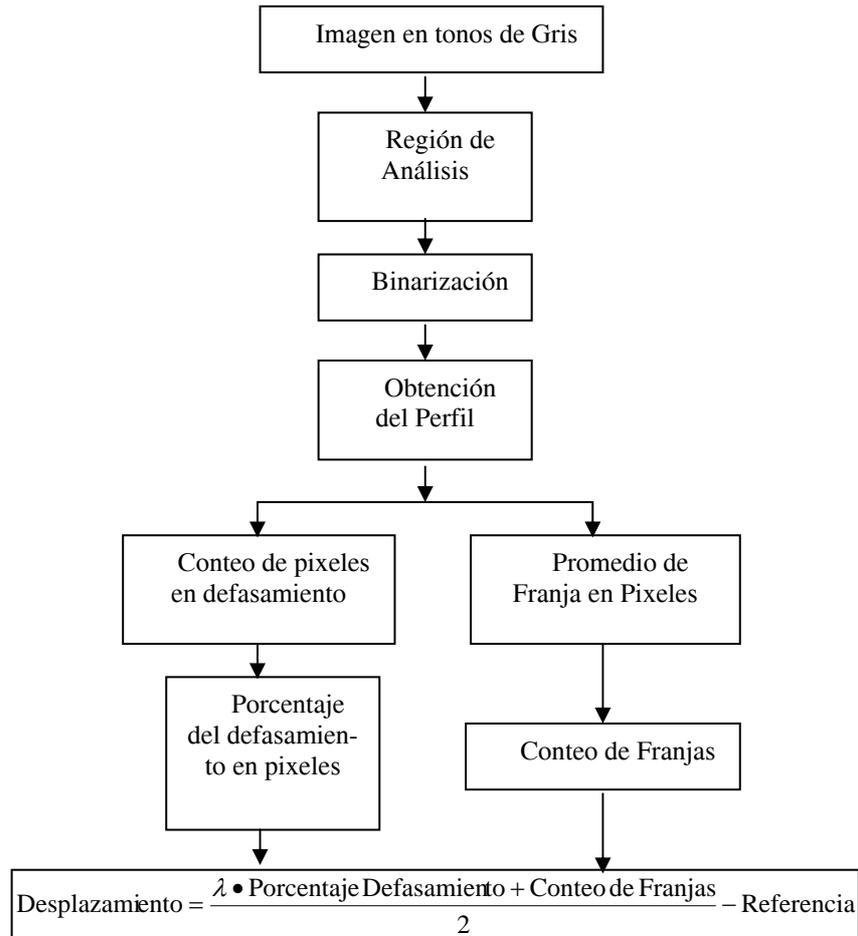


Figura 6. Algoritmo Implementado para obtención del desplazamiento.

Los resultados de la síntesis del sistema obtenidos por Handel-C son:

Numero de capas ocupadas:	2,030 de 14,336	14%
Numero total de LUTs de 4 entradas:	2,512 de 28,672	8%
Equivalente total de compuertas para el diseño:	658,824.	

El sistema completo se muestra en la Figura 7. La interfaz de salida (Figura 8) muestra el cuadro de video analizado, el perfil obtenido del patrón de interferencia y el desplazamiento medido. La sensibilidad del sistema a cambios de desplazamiento con alta frecuencia esta relacionada con resolución y velocidad de adquisición de cuadros de imagen de la CCD utilizada.

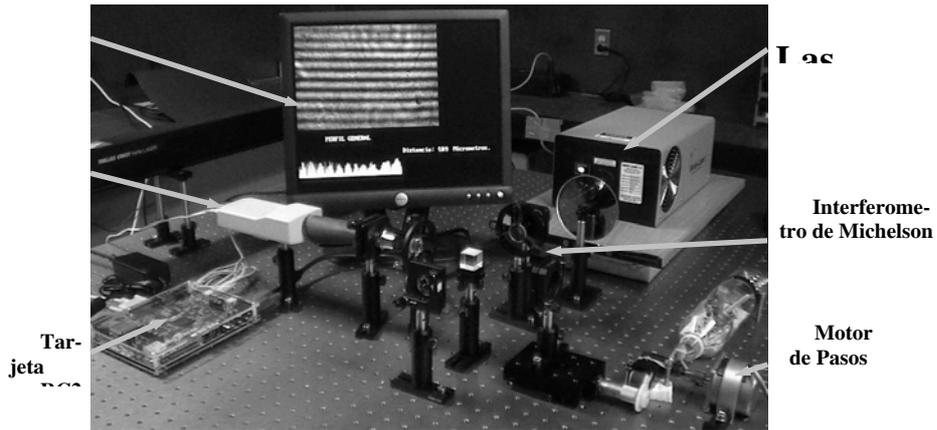
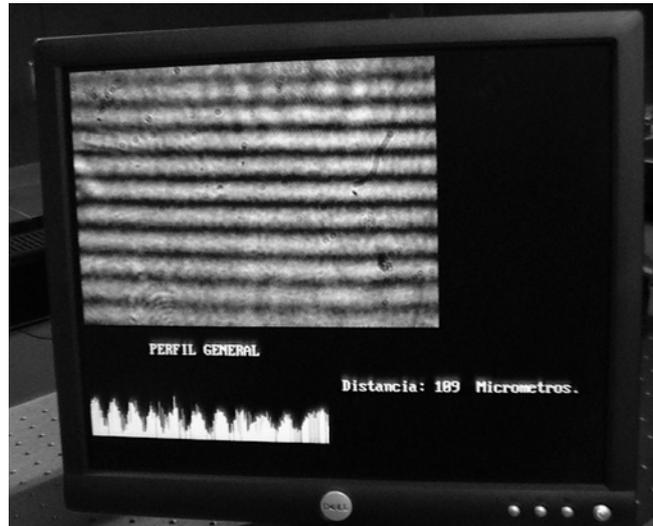


Figura 7. Sistema experimental realizado.



6. Conclusiones

El sistema desarrollado, permite la evaluación de patrones de interferencia en tiempo real. Se analizaron los patrones de interferencia en tonos de gris (8 bits) para disminuir los recursos necesarios para almacenamiento y procesamiento. El sistema utiliza 2030 Slices y el reloj utilizado es a 65 MHz. Una arquitectura paralela es utilizada en el análisis del cuadro de video y en la interfaz gráfica para desplegar el video

analizado, el perfil y el desplazamiento obtenido. La arquitectura propuesta puede ser adaptada para la medición de diferentes variables tales como vibración y deformación. La adaptación solo requiere el cambio del algoritmo utilizado para el procesamiento.

Agradecimientos: Este trabajo ha sido apoyado por el CONACyT, becario No. 130082, proyecto “Reconstrucción Holográfica Digital en Tiempo Real”.

Referencias

1. W. O. Wong, K.T. Chan, T.P. Leung, Contrast and sensitivity of the vibration fringes in time-averaged electronic speckle pattern interferometry: effect of variations or force level, *Optics & Laser Technologies*, 29 (4) (1997) 179-185.
2. N. R. Sivakumar, B.Tan, K. Venkatakrishnan, Measurement of surface profile in vibrating environment with instantaneous phase shifting interferometry, *Optics Communication*, In Press.
3. W.O. Wong, K.T. Chan, Quantitative vibration amplitude measurement with time-averaged digital speckle pattern interferometry, *Optics & Laser Technologies* 30 (1998) 317-324.
4. J. A. Rayas, R. Rodríguez-Vera, A. Martínez, Medición del Módulo de Young en el hule látex usando ESPI, *Revista Mexicana de Física* 49 (6) (2003) 555-564.
5. T. Suzuki, K. Kobayashi, O. Sasaki, Real-Time Displacement Measurement with a Two-Wavelength Sinusoidal Phase-Modulating Laser Diode Interferometer, *Applied Optics* 39 (16) (2000) 2646-2652.
6. M. Hipp, J. Woisetschlager, P. Reiterner, T. Neger, Digital evaluation of Interferograms, *Measurements* 36 (2004) 53-66.
7. Miguel A. Vega-Rodríguez, Juan M. Sanchez-Pérez, Juan A. Gómez-Pulido, Recent advances in computer vision and image processing using reconfigurable hardware, *Microprocessors and Microsystems*, In Press.
8. P. Corsonello, S. Perri, P. Zicari, G. Cocorullo, Microprocessor-based FPGA implementation of SPIHT image compression subsystems, *Microprocessors and Microsystems*, 29 (2005) 299-305.
9. W. Kurdthongmee, Design and implementation of an FPGA-based multiple-colour LED display board, *Microprocessors and Microsystems*, 29 (2005) 327-336.
10. F. T. S. Yu, X. Yang, *Introduction to Optical Engineering*, 1th ed, New York: Cambridge, 1997.
11. Celoxica Inc., <http://www.celoxica.com>
12. Xilinx Inc., <http://www.xilinx.com>