

Difusión

Plataformas

Open Source/Free

Software

**Sistema operativo de
licencia libre**

G. Miramontes

Difusión

**Algoritmos de
Procesamiento Digital de**

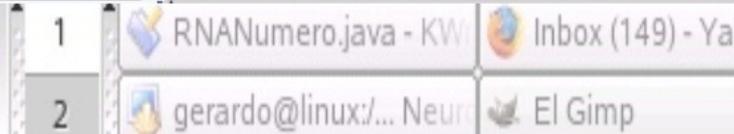
Imágenes en

Arquitecturas

Reconfigurables

FPGA

D. Córdova et al.



PRODUCCIÓN
Universidad Autónoma de Zacatecas

PRODUCCIÓN Y DISEÑO

Gerardo Miramontes de León

D.R. de la Presente Edición

Gerardo Miramontes de León
Universidad Autónoma de Zacatecas
López Velarde 801, Centro
98000 Zacatecas, Zac. México

ISSN 2007-3585

DIFU100ci@ (léase difuciencia) Vol. 2, No.3, enero-abril 2009, es una publicación cuatrimestral editada por la Universidad Autónoma de Zacatecas, "Francisco García Salinas", Jardín Juárez 147, Col Centro Zacatecas, Zac. C.P. 98000. www.uaz.edu.mx/gmiram/Revista.htm. correo-e:gmiram@ieee.org. Reservas de Derechos al Uso Exclusivo del Título expedido por el INDAUTOR, Reserva: 04-2010-110314331900-102. Responsable de la última actualización Gerardo Miramontes de León, López Velarde 801, Zona Centro, Zacatecas, Zac. C.P. 98000. Fecha de última modificación 30 de mayo de 2012.

HECHO EN MÉXICO
MADE IN MEXICO

DIRECTORIO

M. en C. Francisco Javier Domínguez Garay Rector
I. Q. Armando Silva Cháirez Secretario General
M. en C. Jesús Octavio Enriquez Rivera Secretario Académico
M. en A Emilio Morales Vera Secretario Administrativo
Dra. Isabel Terán Elizondo Coord. Investigación y Posgrado
Dr. Luis Alejandro Aguilera Galaviz Coord. de Investigación
Dr. Diego Miramontes de León Coord. de Posgrado

CONSEJO EDITORIAL

Leonardo Acho Zuppa, U Politècnica de Catalunya, España
Miguel Andrés, U. de Valencia, España
Pedro Andrés, U. de Valencia, España
Luis Tupak Aguilar, CITEDIPN, México
David H. Covarrubias Rosales, CICESE, México
Ernesto García Domínguez, U. Autónoma de Zacatecas
Mireya Sara García Vázquez
CITEDIPN, México
Luis García Santander
U. de Concepción, Chile
Geminiano D. Martínez Ponce, CIO, México
Lyle E. McBride, CSU, Chico USA
Oscar Montiel, CITEDIPN, México
Arturo Moreno Báez, U. Autónoma de Zacatecas
Claudia Sifuentes Gallardo, U. Autónoma de Zacatecas

Contenido

Vol. 2, No. 3, enero-abril 2009

EDITORIAL

La confusión de los tecnólogos 55

En esta editorial se presenta una breve reflexión sobre la dependencia y las perspectivas del desarrollo tecnológico en México y probablemente esta reflexión se aplica a toda América Latina. Se admite, de entrada, que existen toda clase de esfuerzos, algunos de los cuales no caben, quizá, en este análisis. Pero se considera que la propuesta tendrá su validez en la mayoría de los casos. La dependencia tecnológica se da desde el uso de herramientas que requieren pagos de licenciamiento excesivos. Si estas herramientas se emplean en la etapa de diseño y luego se propone ese diseño para la planta industrial, eso no aporta beneficios reales de independencia tecnológica. Una alternativa es emplear un Sistema Operativo del tipo GNU/GPL.

Plataforma Open source/Free software 56

por G. Miramontes et al. pp. 56 – 61

El uso de una plataforma de cómputo con licencia GNU/GPL ofrece la posibilidad de ahorrar cantidades significativas en el pago de licencias. No sólo debe verse como un ahorro sino también como una vía hacia la independencia tecnológica. Para favorecer la transición de un sistema operativo que ha dominado los mercados mundiales hacia una plataforma Open Source/Free Software se disponen actualmente de muy variadas versiones de distribución, en las cuales se toma al usuario como usuario final y no necesariamente como desarrollador de software. En este trabajo se presenta una reflexión sobre las ventajas, desde el punto de vista de independencia tecnológica, del uso de una plataforma de código abierto. No es un tutorial sobre cómo instalar un nuevo sistema operativo, más bien se destacan algunos datos que indican cómo esto podría ser una vía hacia la independencia tecnológica.

DIFUSIÓN

Algoritmos de Procesamiento Digital de Imágenes en Arquitecturas Reconfigurables 61

por Diana M. Córdova et al. pp. 61 – 66

En este artículo se presenta la información referente al desarrollo de algoritmos de Procesamiento Digital de Imágenes mediante el uso de arreglos de compuertas programables en campo (FPGA). Los módulos de procesamiento fueron diseñados en el lenguaje de descripción de hardware (VHDL).

La confusión de los tecnólogos el caso Mexicano



VANCE y desarrollo han sido la premisa de muchas civilizaciones, cuyo fin último es el bienestar de sus habitantes. En este mismo contexto, este avance y desarrollo se mide por el avance tecnológico y científico, el cual no sólo está ligado al bienestar de sus habitantes sino al poder que una civilización tendrá sobre otras menos desarrolladas. En Latinoamérica se ha reconocido que fue la diferencia en tecnología el factor clave para que se diera la conquista de nuestros pueblos y una vez hecho este reconocimiento han habido pronunciamientos de algunos líderes oficiales reconociendo que debemos tratar de disminuir esas diferencias.

Muy recientemente se dio un nuevo discurso “ALGO HICIMOS MAL” que retoma las mismas conclusiones. Este discurso hecho público en la Cumbre de las Américas, Trinidad y Tobago 18 de abril del 2009, por el Presidente de la República de Costa Rica y Premio Nobel de la Paz, Oscar Arias, admite que “vimos pasar la revolución industrial como un cometa y perdimos la oportunidad de montarnos en ese vagón.” Lamentablemente ese “algo hicimos mal” sigue siendo “algo hacemos mal”, porque no se ve que en el futuro cercano ese estado de cosas vaya a cambiar. El poder de las naciones, se dice, está en el conocimiento, y como apunta el discurso del Sr. Arias, hay países que no lo tenían y ahora lo tienen, como China, Corea del Sur, Singapur y otros más.

Para afirmar que algo hacemos mal, conviene reconocer el estado de confusión que impera no sólo en los líderes políticos sino, lo más grave, en los actores académicos de este país. Es cierto que existen centros de investigación en ciencia básica y centros de investigación en desarrollo tecnológico. Sin embargo, dada la política de un sistema que califica la productividad científica (y su retribución), a partir del número de publicaciones y más actualmente de patentes, los investigadores y tecnólogos han caído en la confusión de que todo aquello que no sea publicable, o ya haya sido hecho, no hay por qué intentarlo siquiera. Con ese fin de tener algo que publicar, se compra siempre equipo importado y licencias de software. Se favorece la dependencia tecnológica del país al formar verdaderos usuarios de tecnología importada, mientras que los ingenieros egresados de universidades, principalmente privadas, se convierten en agentes de ventas de compañías transnacionales, bajo el ostentoso disfraz de “gerente”.

Un hecho que marca nuestras diferencias con países como China es la aceptación, en el área de cómputo, de un Sistema Operativo abierto en lugar del sistema que domina los mercados actualmente (ver IEEE Spectrum, “Microsoft Suffers Chinese Triple Whammy”, pág. 18 Octubre 2003). Una forma de comenzar a romper la dependencia en algunos desarrollos tecnológicos es evitando el uso de herramientas que requieren pagos excesivos de licenciamiento, no sólo mientras se hace el desarrollo, si no en la eventual venta de un producto. Seguramente, este es un tema que a algunos parecerá delicado y que debemos discutir con profundidad.

© abril 2009 G. Miramontes

DIFU100ci@ No. 3, Vol. 2, enero-abril 2009

correo-e:gmiram2002@yahoo.com

DIFU100ci@ (léase difuciencia) es una publicación cuatrimestral editada por la Universidad Autónoma de Zacatecas, Jardín Juárez 147, Col Centro Zacatecas, Zac. Tiene como objetivo difundir conocimientos científicos y tecnológicos del área de la ingeniería a través de artículos de divulgación y artículos que muestren temas de investigación.

EDITOR EN JEFE

Gerardo Miramontes de León
U. Autónoma de Zacatecas

CONSEJO EDITORIAL

[Leonardo Acho Zuppa](#)

U Politècnica de Catalunya
España

[Miguel Andrés](#)

U. de Valencia, España

[Pedro Andrés](#)

U. de Valencia, España

[Luis Tupak Aguilar](#)

CITEDI-IPN, México

[David H. Covarrubias Rosales](#)

CICESE, México

[Ernesto García Domínguez](#)

U. Autónoma de Zacatecas

[Mireya Sara García Vázquez](#)

CITEDI-IPN, México

[Luis García Santander](#)

U. de Concepción, Chile

[Geminiano D. Martínez Ponce](#)

CIO, México

[Lyle E. McBride](#)

CSU, Chico USA

[Oscar Montiel](#)

CITEDI-IPN, México

[Claudia Sifuentes Gallardo](#)

U. Autónoma de Zacatecas

El contenido de los artículos es responsabilidad exclusiva de los autores. Se permite la reproducción total o parcial de los contenidos siempre y cuando se cite la fuente, y en los términos de la Ley Federal de Derechos de Autor y, en su caso, de los tratados internacionales aplicables.



Plataforma Open source/Free software

G. Miramontes

Open source/Free software an option

Recibido: febrero 12, 2009

Aceptado: abril 17, 2009

Palabras clave: código abierto; software libre; licencia GNU/GPL

Abstract:

The use of a computing platform licensed under the GNU / GPL offers the potential to save significant amounts in license fees. Not only should be seen as a saving but also as a way toward technological independence. To facilitate the transition from an operating system that has dominated global markets into a platform Open Source / Free Software is currently available versions of a wide variety of distribution, in which the user is taken as the end user and not necessarily as a software developer. This paper presents a reflection on the advantages from the point of view of technological independence, the use of an open source platform. Not a tutorial on how to install a new operating system, rather it highlights some evidence to suggest how this could be a pathway to technological independence.

Keywords: Open source; free software; GNU/GPL license

EN el espacio editorial de este número, destacamos dos aspectos que consideramos son determinantes para romper la dependencia tecnológica. Uno de ellos es la falta de acciones de ingeniería inversa, y el otro la falta de decisión para imponer el uso de una plataforma de cómputo que no sea la que tradicionalmente ha dominado el mercado mundial, que en adelante referiremos como MS.

Para el primer punto, se requiere de un foro de discusión más amplio que permita el debate, ya que habrán puntos de vista diversos y en diferentes sentidos. Quizá conviene pensar en una sección abierta a las opiniones de los lectores. Para el segundo, aprovechamos este espacio para difundir algunas ideas sobre el uso de lo que se conoce como “Software libre y de código abierto”, en adelante referido como FS (Free Software, de sus siglas en inglés).

INDEPENDENCIA TECNOLÓGICA

En la edición de Octubre de 2003 de IEEE Spectrum, se puede leer que el gobierno de Beijing requeriría a sus ministros comprar software basado en Linux¹ producido domésticamente. Posteriormente, cuando MS demandó que la Comisión de Educación de Shanghai pagara licencia completa por el software utilizado en escuelas, el gobierno simplemente retiró el sistema operativo y aplicaciones de MS de las computadoras de los estudiantes y lo reemplazó por el sistema doméstico. Mientras que, también indica la nota, Scott Kennedy, un profesor de ciencia política en la Universidad de Indiana, escribía un libro donde asegura que “el gobierno está tratando de ganar independencia de multinacionales extranjeras” (se refiere al gobierno de China). Aunque todo esto pueda parecer trivial, sería interesante conocer, en cifras reales, cuánto se paga en licencias a MS desde nuestros centros de investigación y oficinas gubernamentales.

El asunto deja de ser completamente trivial, cuando vemos que en este país sería imposible que un Secretario de Educación tomara una decisión tan importante. Primero porque sería muy fácil para la transnacional llenarle los bolsillos (a él y a cualquier legislador) de modo que jamás se promulgara una ley contra la dependencia tecnológica. Nuestra dependencia tecnológica se da de muchas maneras y lo que genera es un fuga importante de divisas en pagos de licencias. Un caso lamentable es el proyecto de llevar a todas las escuelas una plataforma electrónica, “Enciclomedia”, de cuyos favores se hablaba demasiado. Sin embargo, poco se decía del costo que representaba en pago de licencias (en realidad, da la impresión de que se trata de una compra masiva disfrazada de Encarta), mientras que se tenían y se siguen teniendo escuelas rurales sin las condiciones mínimas para trabajar, con piso de tierra y pizarrones, si los hay, donde difícilmente se puede escribir.

Nuestra dependencia tecnológica se da de muchas maneras.

Algunos casos interesantes son el caso Brasileño [1], y el caso Chileno [2]. Existe un principio denominado de “neu-

¹ Linux en realidad es GNU/Linux, el sistema operativo es GNU y Linux es el núcleo.

tralidad tecnológica”, en el cual el Estado o los gobiernos no deben favorecer la presencia de sólo uno de los modelos de explotación de software. En pocas palabras, el gobierno ofrecería condiciones equitativas para que empresas y en general la sociedad opten por el software de código cerrado o de código abierto. Este principio es la base para que un gobierno evite promover el uso de software de código abierto, y mucho menos hacerlo obligatorio. En pocas palabras, resulta en un excelente pretexto a favor de MS.

El caso es que la equidad no se da, ya que existe una desventaja del FS al no contar con la infraestructura comercial, y sobre todo publicitaria, además de las fuertes “donaciones” que supuestamente hace MS a los gobiernos, digo supuestas porque después se cobran. En el caso de Chile resultó que si se desinstalaba la plataforma de MS, todos los datos quedaban en propiedad de MS por los cuales debían pagar, ver[4]. En fin, la opción de una plataforma FS no sólo debe verse como una estrategia para reducir costos si no más bien como una vía para reducir la dependencia tecnológica, en materia de tecnologías de la información.

EL MOVIMIENTO GNU

El proyecto GNU fue iniciado por Richard Stallman con el objetivo de crear un sistema operativo completamente libre: el sistema GNU. El manifiesto GNU establece que su principal motivación es volver al espíritu de cooperación entre la comunidad de usuarios de computadoras[3]. GNU es un acrónimo recursivo que significa “GNU No es Unix” (GNU is Not Unix), aunque la idea inicial era precisamente crear un sistema operativo compatible con Unix. Hoy en día GNU está asociado al concepto de Free Software, “Software libre”, aunque desafortunadamente haya confusión con “gratis”.

Para aclarar este punto Stallman hace notar que el significado de Free es de libertad (Free as in freedom) y que no se refiere al precio. Un programa es considerado FS si Ud como usuario final:

- Tiene la libertad de ejecutarlo, para cualquier propósito.
- Tiene la libertad de modificarlo para cubrir sus necesidades (para hacer efectiva esta libertad en la práctica, Ud debe tener acceso al código fuente).
- Tiene la libertad de distribuir copias, ya sea gratis o con un costo.

- Tiene la libertad de distribuir versiones modificadas del programa, de modo que la comunidad se pueda beneficiar de sus mejoras.

Debe entenderse que vender copias de FS no va en contra de su filosofía y es además importante como un medio de adquirir fondos para continuar su desarrollo. En 1985, Stallman creó la Free Software Foundation (FSF, Fundación para el Software Libre) para tener así todo el apoyo legal y financiero para el proyecto GNU. A partir de este movimiento se genera una nueva forma de licencia, la licencia GNU GPL (General Public Licence) cuya esencia es contraria al copyright. De lo que se trata es de impedir que a partir de la plataforma de GNU se desarrollen plataformas o programas propietarios, los cuales restringen la libertad de los usuarios.

Free Software vs Open Source

Generalmente se confunde el Free Software con el Open Source. Sin embargo hay diferencias importantes [5], como asegura Stallman. Mientras que Free Software habla de completa libertad en la forma de distribución del sistema operativo de muchas aplicaciones, el Open Source ya presenta ciertas restricciones y en cierto sentido hay compañías que se desempeñan de una manera más cercana a las plataformas y aplicaciones propietarias. Nuevamente aclaramos que la comercialización de productos GNU no es negativa, pero que debe respetar el espíritu de completa libertad para el usuario. Un ejemplo muy exitoso de la distribución comercial de productos GNU es Red Hat.

Linus y su Linux

Un paso importante en el movimiento GNU fue el dado por Linus Torvalds, quien en 1991 comenzó a desarrollar un núcleo, posteriormente conocido como Linux que se distribuyó bajo la licencia GPL. Este núcleo combinado con el sistema GNU es hoy en día un sistema operativo completo y se le conoce como GNU/Linux. A partir de entonces se dan distribuciones las cuales incluyen en su mayoría muchas aplicaciones con licencia libre. En la Figura 1 se muestran algunas distribuciones de este sistema operativo.

Para 1995 ya se contaba con una distribución de Linux por parte de una editorial de amplia circulación. En la Figura 2 se muestra el CD-ROM de la distribución de Linux por PHH. Cabe señalar que para ese entonces, la instalación del



Figura 1. Versiones de distribución de GNU/Linux.

sistema operativo sí requería de un esfuerzo que muchos usuarios no estaban dispuestos a invertir en él.

OPCIONES DE CÓDIGO ABIERTO

Además del concepto de código abierto, debe entenderse como la opción de software libre de licencias. Las opciones actuales son tales que fácilmente se pueden satisfacer las necesidades del usuario. Una instalación común, incluirá muchas herramientas a las cuales están acostumbrados la mayoría de los usuarios, incluyendo las aplicaciones de entretenimiento.

En cuanto a las herramientas de productividad, se puede decir que se encuentran todas las necesarias. Cuando se requieran aplicaciones especiales, como diseño asistido por computador (CAD de sus siglas en inglés) existe disponibilidad de ellas también.

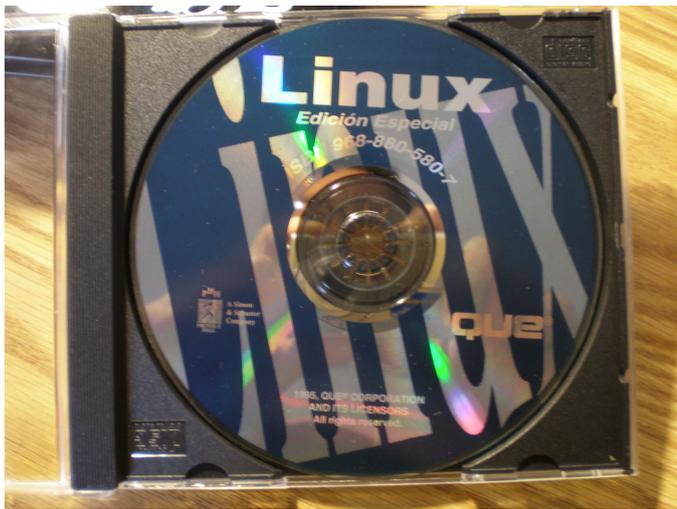


Figura 2. Distribución de Linux en 1995 por PHH.

Listado de algunos programas desarrollados por el proyecto GNU

- Bison: generador de analizadores sintácticos diseñado para substituir a yacc
- Bash: intérprete de comandos
- BFD: archivos de bibliotecas
- Binutils: Ensamblador GNU, Enlazador GNU, y herramientas relacionadas
- Classpath: bibliotecas para Java
- DotGNU: substituto de .NET
- Emacs: editor de texto extensible y autodocumentado
- GCC: compilador optimizado para varios lenguajes, particularmente C
- GDB: depurador de aplicaciones
- GNU Ghostscript: Aplicaciones para PostScript y PDF [3]
- GIMP: programa de edición fotográfica
- Glibc: biblioteca para lenguaje C
- GMP: biblioteca para cálculos con precisión arbitraria
- GNOME: ambiente de escritorio gráfico

Sistema de construcción para GNU

- GUNet: red descentralizada de comunicaciones personales, diseñada para resistir interferencias no autorizadas
- GNUstep: implementación del conjunto de bibliotecas OpenStep, así como herramientas para programar aplicaciones gráficas
- GSL: Biblioteca Científica para GNU
- Gzip: Aplicaciones y bibliotecas para compresión de datos
- Hurd: un micronúcleo y un conjunto de servidores que funcionan del mismo modo que el núcleo UNIX
- Maxima: un sistema para cálculos algebraicos
- Octave: un programa para cómputo numérico similar a MATLAB
- GNU MDK: un conjunto de herramientas para la programación en MIX
- Texinfo: sistema de documentación
- LilyPond: editor de partituras musicales.

El proyecto GNU también ayuda con el desarrollo de otros paquetes

- CVS: sistema de control de versiones para código fuente
- DDD: herramientas gráficas para detección y depuración de errores

Bibliografía

- [1] Proyecto de ley No. 1.269/99 (Pinheiro), de 15 de Diciembre 1999;
- [2] http://www.derechoinformatico.uchile.cl/CDA/der_informatico_complex/0,1491,SCID%253D15831%2526ISID%253D567,00.html
- [3] <http://www.gnu.org/gnu/manifesto.es.html>

-
- [4] <http://liberaciondigital.org/blog/2007/09/implicancias-del-respaldo-de-los-diputad>
- [5] <http://www.gnu.org/philosophy/free-software-for-freedom.html>

Acerca del autor o autores

Profesor-investigador de la Fac. de Ingeniería Eléctrica de la Universidad Autónoma de Zacatecas, y miembro del Cuerpo Académico de Procesamiento Digital de Señales.





Algoritmos de Procesamiento Digital de Imágenes en Arquitecturas Reconfigurables

Diana Margarita Córdova Esparza, Ernesto García Domínguez, Arturo Moreno Báez

Algorithms of Digital Image Processing in Reconfigurable Architectures FPGA

Recibido: abril 1, 2009

Aceptado: abril 22, 2009

Palabras clave: procesamiento de imágenes; arquitectura reconfigurable; FPGA

Abstract:

This article presents information concerning the development of algorithms of Digital Image Processing by using arrays of field-programmable gate array (FPGA). Processing modules were designed in the hardware description language (VHDL)

Keywords: image processing; reconfigurable architecture; FPGA



ACE algunos años, la gran mayoría de los sistemas electrónicos estaban basados en la utilización de microprocesadores, debido a factores que contribuyeron a su popularización como: su facilidad de uso y bajo costo. A medida que las aplicaciones se hicieron más variadas, surgieron nuevos dispositivos más eficientes para la realización de algunas tareas específicas, como los Procesadores Digitales de Señales (DSPs), concebidos para implementar muchas aplicaciones tales como: protocolos de voz para internet, comunicaciones inalámbricas, sistemas de radar, procesamiento de imágenes, sistemas multimedia, etc. Sin embargo, aunque los procesadores son programables a través de software, la arquitectura del hardware del DSP no es flexible. Por lo tanto, los DSPs son limitados por la arquitectura fija del hardware y resulta inadecuado su uso para ciertas aplicaciones que podrían requerir implementaciones de funciones de procesamiento digital de señales dedicadas.

Otros dispositivos que también han revolucionado la tecnología son los microcontroladores diseñados especialmente para aplicaciones embebidas. Sin embargo, ninguno de estos dispositivos es capaz de superar la eficiencia en cuanto a procesamiento y consumo de potencia de los Circuitos Integrados de Aplicación Específica (ASIC), que

como su nombre indica sólo pueden realizar la función para la que fueron construidos.

Adicionalmente, se cuenta con otro tipo de herramientas como los Dispositivos Lógicos Programables (PLD), que en términos de procesamiento son tan eficientes como un ASIC y que pueden ser reprogramados como un microprocesador. Actualmente, gracias a los avances en la fabricación de circuitos integrados, surgen los Arreglos de Compuertas Programables en Campo (FPGA) como el resultado de la convergencia de estas dos tecnologías. Estos dispositivos se han convertido en una opción interesante a la hora de diseñar sistemas que requieren una alta capacidad de procesamiento y un bajo consumo de potencia [1].

Los circuitos FPGA suministran una solución reconfigurable y eficiente para implementar tareas de procesamiento digital de señales. Estos circuitos pueden alcanzar una mayor potencia de procesamiento de datos que los DSPs.

Tanto los DSPs, como los FPGAs presentan ventajas y desventajas para una tarea en particular. Por ejemplo, los DSPs presentan ventajas de velocidad de procesamiento, eficiencia en el consumo de energía y buena relación desempeño-costos comparados con los procesadores de propósito general para aplicaciones de procesamiento digital de señales.

Por otra parte los FPGAs son una opción atractiva para el desarrollo de aplicaciones de procesamiento digital, debido a que disponen de una arquitectura reconfigurable que permite una gran capacidad de paralelismo y flexibilidad [2].

Los FPGAs incorporan más elementos con una amplia gama de soluciones de conectividad, haciendo posible el diseño de complejos y potentes sistemas en un solo dispositivo. Una de sus múltiples aplicaciones está dirigida al área de Procesamiento Digital de Imágenes, en donde se han desarrollado plataformas que permiten realizar algoritmos sencillos como el cálculo de histograma y negativo de una imagen, hasta algoritmos más complejos como: filtrado (remoción de ruido) y segmentación, obteniendo resultados óptimos.

FUNDAMENTOS TEÓRICOS

Los FPGAs son circuitos integrados que contienen bloques de lógica configurable (CLB), bloques de entrada y salida (IOB) y canales de comunicación cuya interconexión y funcionalidad se puede programar.

Un FPGA se compone de elementos con recursos no comprometidos que pueden ser seleccionados, configurados e

interconectados por el usuario. Estos dispositivos se componen de cierto número de módulos lógicos, que determinan la capacidad del dispositivo. Los módulos son independientes entre sí y pueden interconectarse para formar un módulo más complejo.

Arquitectura del FPGA

Existe una gran variedad de FPGAs provistos por varias compañías como Xilinx, Altera, Atmel y Lattice. Cada fabricante provee a su FPGA con una arquitectura única. Un FPGA típico está formado por bloques lógicos configurables, bloques configurables de entrada/salida e interconexiones programables como se muestra en la Figura 1:

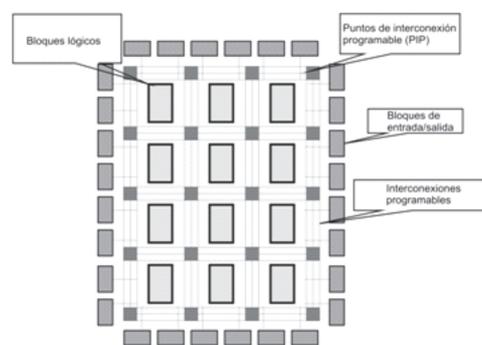


Figura 1. Diagrama a bloques de la arquitectura de un FPGA

- Bloques lógicos configurables. Los Bloques Lógicos Configurables (CLBs, Configurable Logic Blocks) son recursos lógicos que permiten al usuario realizar diferentes funciones; los CLBs están distribuidos en forma matricial en el dispositivo. En el caso donde estos recursos son de complejidad baja, es decir, las funciones lógicas que se pueden realizar en ellos son sencillas, y existe un número considerable de ellos, se dice que el FPGA es de granularidad fina. Cuando los recursos lógicos están formados por memorias de acceso aleatorio llamadas Tablas de Búsqueda (LUT, Look-up Tables), flip-flops para almacenamiento de elementos que dependen de la señal de reloj, multiplexores que permiten la selección, reset y puesta a uno lógico de elementos, se dice que el FPGA es de granularidad gruesa; en un FPGA de este tipo el número de

CLBs que lo integran es reducido pero tienen la característica de poder implementar funciones de mayor complejidad.

- Bloques configurables de entrada/salida. La matriz de bloques de CLBs está rodeada por un anillo de bloques de interfaz, denominados bloques configurables de entrada/salida. Estos bloques están dedicados a proporcionar la interconectividad entre el FPGA y el exterior, es decir, controlan la entrada y salida de datos entre los pines de entrada/salida y la lógica interna.
- Interconexiones programables. Están formados por: recursos de interconexión, conjunto de líneas y/o interruptores programables que permiten transmitir las señales entre los bloques lógicos internos y entre éstos y los bloques de entrada/salida, y de matriz de interconexión, elementos lógicos que facilitan la comunicación entre los buses de comunicación (recursos de interconexión).
- Circuitería de reloj. Existe un tercer tipo de recursos exclusivos de conexión: las líneas dedicadas a la transmisión de las señales de reloj. Estas señales generalmente están conectadas a un gran número de bloques por lo que son diseñadas para obtener tiempos de propagación pequeños y similares. Además, están distribuidas alrededor del FPGA mediante buffers de reloj especiales, conocidos como drivers de reloj. Estos buffers se encuentran conectados al reloj principal y lo llevan hacia todas las líneas de reloj global con el fin de que puedan ser utilizadas por cada CLB [3].

ETAPAS DEL DISEÑO EN FPGA

Para llevar a cabo la programación del FPGA es necesario utilizar algún lenguaje de descripción de Hardware como VHDL o Verilog que consiste en una serie de instrucciones y directivas parecidas a los lenguajes de programación de alto nivel. A diferencia de estos lenguajes, los lenguajes de descripción de hardware no se usan para implementar un algoritmo en una computadora. Como su nombre lo indica, se utilizan para describir componentes que forman parte de un circuito y su interconexión. La estructura general de un programa en VHDL está formada por módulos o unidades de diseño, cada uno de ellos compuesto por un conjunto de declaraciones e instrucciones que definen, describen, estruc-

turan, analizan y evalúan el comportamiento de un sistema, como se muestra en la Figura 2.

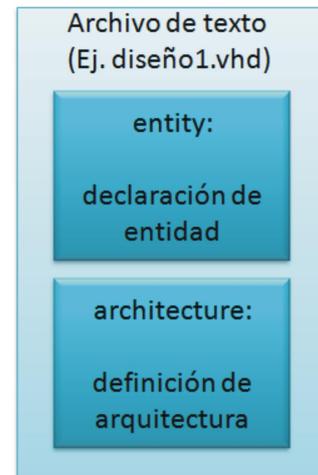


Figura 2. Estructura de un programa en VHDL.

De este modo, una entidad VHDL es simplemente una declaración de las entradas y salidas de un módulo, mientras que una arquitectura VHDL es una descripción detallada del comportamiento o estructura interna del módulo [4]. Otras de las etapas que comprende el diseño de un sistema en un FPGA, son las siguientes:

- Simulación del diseño. Una vez terminado la descripción de VHDL, es importante saber si el diseño así generado realizará la función deseada. Esto se puede lograr con la simulación del diseño, la cual además es útil para verificar la sintaxis del programa. Hay que tomar en cuenta que en este punto hace falta considerar algunos factores, como los retardos de tiempo inherentes de los componentes, por lo que esta simulación es sólo una primera aproximación.
- Síntesis del diseño. En esta parte se crea una descripción a nivel de compuertas del diseño obtenido. El resultado de esta etapa es la obtención de una lista con las compuertas necesarias para implementar un diseño, además de todas las conexiones necesarias entre las mismas.
- Mapeo. El mapeo usa la descripción de compuertas generadas por la herramienta de síntesis, distribuyéndolas entre los diferentes recursos con los que cuenta el FPGA, como son los CLB's, bloques de memoria, multiplicadores y otros.

- Colocación y ruteo. En el mapeo fueron definidos los recursos del FPGA que se deben usar para implementar el diseño. Con la colocación se seleccionan los componentes específicos a ser usados y con el ruteo se definen las conexiones entre los diferentes componentes seleccionados.
- Programación. Después de haber cubierto todas las etapas anteriores, en este momento es posible obtener el archivo necesario para programar el FPGA, conocido como bitstream. La programación consiste en enviar este archivo al FPGA para que establezca las conexiones apropiadas entre los diferentes componentes del mismo [3].

TÉCNICAS DE PROCESAMIENTO DIGITAL DE IMÁGENES

Existe una gran variedad de técnicas en el área de Procesamiento Digital de Imágenes, sin embargo algunas de ellas, aunque simples sirven de base para el desarrollo de algoritmos más complejos. El histograma, por ejemplo, aporta información importante acerca de la calidad o contraste que presenta la imagen. El negativo, por otra parte, resulta útil cuando se quieren resaltar detalles blancos o grises que se encuentran en regiones oscuras de la imagen [3].

Histograma

El histograma de una imagen con niveles de gris en el rango $[0; L - 1]$, es una función discreta que representa el número de pixeles que posee cada nivel de gris en la imagen, o dicho en otras palabras, es la frecuencia relativa de ocurrencia de cada nivel de gris en la imagen.

$$h(r_k) = n_k \quad (1)$$

donde r_k es el k -ésimo nivel de gris y n_k es el número de pixeles en la imagen que contienen dicho nivel.

Negativo de una imagen

El negativo de una imagen con niveles de gris en el rango $[0; L - 1]$, es obtenido al restar cada uno de sus pixeles de entrada del valor mínimo de nivel de gris $(L - 1)$ permitido en la imagen, esto es:

$$s = (L - 1) - r \quad (2)$$

donde r y s corresponden a los niveles de gris asociados a los pixeles de entrada y salida respectivamente.

La Figura 3 muestra la función de transformación empleada para la obtención del negativo de la imagen.

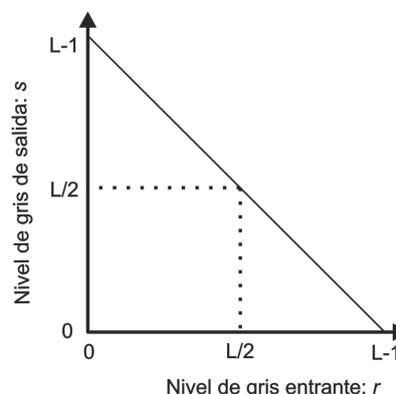


Figura 3. Función de transformación del negativo de una imagen

SÍNTESIS DE MÓDULOS DE PROCESAMIENTO EN FPGA MEDIANTE VHDL

En esta sección se realiza la descripción tanto del módulo para el negativo como para el histograma de una imagen.

Módulo: Negativo de una imagen

En la Figura 4 se muestra el diagrama a bloques de la función de transformación utilizada para obtener el negativo de la imagen.

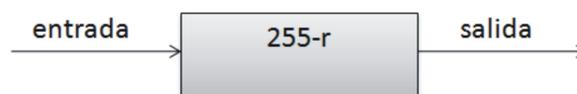


Figura 4. Diagrama a bloques para la obtención del negativo.

A partir del modelo anterior, se define el diseño funcional de este bloque, en el que se describe la relación entrada-salida.

A continuación se muestra el listado correspondiente a esta descripción, utilizando el lenguaje VHDL:

```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity negativo is
  Port ( clk : in STD_LOGIC;
        rst: in STD_LOGIC;
        Dato: in STD_LOGIC_VECTOR(7 downto 0);
        Neg: out STD_LOGIC_VECTOR(7 downto 0));
end negativo;

architecture Behavioral of negativo is
begin

negativ: process (clk,rst,Dato)
begin
  if clk'event and clk='1' then
    if rst='1' then
      Neg <= (others => '0');
    else
      --Neg <= 255 - Dato;
      Neg <= not Dato;
    end if;
  end if;
end process;

end Behavioral;
%end{verbatim}

```

En la entidad se definen las terminales de entrada y salida del módulo, así como el tipo de línea usada (simple o bus). En la parte correspondiente a la descripción funcional (process), se realiza el cálculo del negativo, de acuerdo a la función de transformación dada en (2). Este proceso se realiza en forma síncrona, lo cual es controlado por el reloj (clk) del sistema.

Finalmente, como resultado de la síntesis del diseño anterior se obtiene el circuito mostrado en la Figura 5.

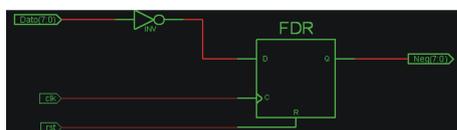


Figura 5. Síntesis para el módulo del negativo.

En esta figura se observa que la síntesis de esta función de transformación genera básicamente una compuerta inversora, lo cual representa una estructura bastante simple para la implementación del negativo.

Módulo: Histograma de una imagen

Enseguida se muestra la estructura propuesta para calcular el histograma de una imagen dentro del FPGA (ver, Figura 6). Básicamente consiste en una memoria RAM que permite registrar la cantidad de pixeles de cada nivel de gris. Para esto, utiliza un sumador que incrementa en uno la localidad de memoria asociada a cada nivel de gris. En este caso el valor del pixel sirve para apuntar a la dirección de memoria a incrementar [5]. Cuando se han terminado de procesar todos los puntos de la imagen, la memoria contiene el total de pixeles en cada nivel de gris.

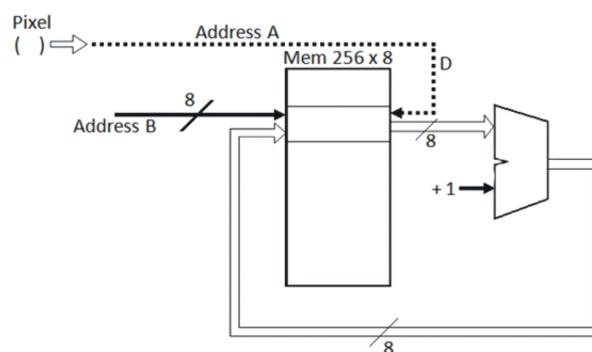


Figura 6. Diagrama a bloques para el cálculo del histograma.

PRUEBAS Y RESULTADOS

Para evaluar el desempeño de los algoritmos realizados, se inyecta una imagen de prueba al módulo FPGA. Dicha imagen es transferida desde la computadora hacia el FPGA, a través del puerto serie. La información que resulta del procesamiento es reenviada a la computadora para su despliegue.

A continuación se presentan los resultados experimentales obtenidos al implementar en el FPGA los algoritmos descritos anteriormente.

Prueba 1: Negativo de la imagen

La Figura 7 corresponde al resultado que se obtiene al aplicar el algoritmo de negativo a la imagen de prueba.



Figura 7. Imagen original y negativo.

Prueba 2: Histograma de la imagen

La Figura 8 presenta la imagen original y el histograma que genera el FPGA. Para este ejemplo, el nivel de gris mínimo corresponde al valor de 81, lo cual equivale aproximadamente a una tercera parte del intervalo de tonos de gris posibles (generalmente $L=255$ niveles).

Con esta información que arroja el histograma es posible cuantificar el nivel de contraste que presenta la imagen de prueba, ya que existe una relación directa con el intervalo dinámico de tonos de gris que contiene la imagen.

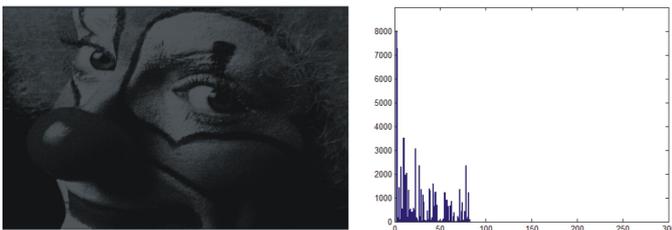


Figura 8. Imagen original e histograma.

CONCLUSIONES

Se pueden realizar técnicas de Procesamiento Digital de Imágenes por medio de arquitecturas reconfigurables (FPGA) obteniendo resultados satisfactorios. Una de las principales ventajas de este tipo de tecnología es que se trata de dispositivos reconfigurables, lo cual permite un diseño acorde a las necesidades específicas de cada aplicación.

El diseño y síntesis de algoritmos de Procesamiento Digital de Imágenes en un FPGA, es una tarea que requiere no sólo del completo conocimiento de las técnicas, sino también de una vasta experiencia en el diseño de sistemas digitales, ya que esto permitir plantear estructuras simples, que realicen

su función de manera óptima y con el mínimo tiempo de procesamiento posible, lo cual permitirá su eventual aplicación en tareas de procesamiento en tiempo real.

Bibliografía

- [1] Alexander Quintero M. y Eric Vallejo R., “Algoritmos de procesamiento de imágenes en FPGA”, Revista Colombiana de Tecnologías de Avanzada, pp.11-12, 2006.
- [2] Ferney O. Amaya., “Aplicaciones para telecomunicaciones empleando FPGAs: Una aplicación a radio software”, Revista Colombiana de Tecnologías de Avanzada, pp. 39, 2006.
- [3] Manuel A. Mendoza., “Procesamiento y análisis digital de imágenes mediante dispositivos lógicos programables”, pp.16-40, 2009.
- [4] David G. Maxinez y Jessica Alcal , VHDL, El arte de programar sistemas digitales, Patria, Pub. Co.,2008, México.
- [5] Edgard Garcia., “Implementing a Histogram for Image Processing Applications”, Revista Xcelljournal, pp. 46, 2000.

Acerca del autor o autores

Diana M. Córdova es estudiante de la Maestría en Ingeniería con orientación en Procesamiento Digital de Señales, Ernesto García es Profesor-investigador de la Fac. de Ingeniería Eléctrica de la Universidad Autónoma de Zacatecas, y miembro del Cuerpo Académico de Procesamiento Digital de Señales. Arturo Moreno es estudiante de doctorado en la Facultad de Ingeniería Eléctrica, todos ellos en la Universidad Autónoma de Zacatecas.

INSTRUCCIONES PARA AUTORES

DIFU100ci@ (léase difuciencia) es una publicación cuatrimestral del Cuerpo Académico de Procesamiento e Instrumentación óptica, de la Facultad de Ingeniería Eléctrica, Universidad Autónoma de Zacatecas. Tiene como objetivo difundir conocimientos científicos y tecnológicos del área de la ingeniería, a través de artículos de divulgación y artículos que muestren temas de investigación. La revista cuenta con el Certificado de Reserva de Derecho al Uso Exclusivo del Título expedido por el INDAUTOR, Reserva: 04-2010-110314331900-102.

Los trabajos pueden ser clasificados, al menos, en tres categorías: Divulgación, Investigación, y Tutoriales.

Divulgación: Artículos que no necesariamente contienen resultados de proyectos de investigación propiamente. En esta sección también se podrán incluir trabajos que presenten nuevos enfoques a temas de investigación de modo que permitan ampliar su difusión.

Investigación: Se pueden incluir resultados de investigación aunque conservando el enfoque a la divulgación, es decir, no necesariamente con el rigor de un “paper”. Con ese enfoque se pretende motivar aun mayor número de lectores al hacerles llegar los nuevos tópicos que se estudian actualmente. Los trabajos sobre desarrollo tecnológico pueden ser considerados en esta categoría y se recomienda que un mayor número de trabajos correspondan a esa importante tarea.

Tutoriales: Se muestran temas novedosos, pero poco conocidos. O bien, nuevos enfoques a temas básicos, con un objetivo didáctico, de modo que permitan ampliar el conocimiento y motiven su aplicación en proyectos de ingeniería (desarrollo tecnológico). Página provisional de la Revista:

<http://www.uaz.edu.mx/gmiram/Revista.htm> **Sobre el formato del texto:** Los trabajos se pueden enviar en formato txt y las gráficas o figuras en formato jpg con buena resolución. De ser posible debe enviarse un solo archivo rar o zip que contenga tanto el texto como las figuras. En el caso de utilizar Word, se deberá enviar en formato .doc, sin utilizar macros (no se aceptará formato docx). También se aceptarán trabajos en formato OpenOffice. No se solicita ningún formato de página en especial, ya que el texto será llevado al formato de la revista. El trabajo debe incluir

1. Título y Lista de autores.
2. Resumen en un máximo de 200 palabras. El Resumen se utiliza como descripción del trabajo en el índice ampliado, y no aparece en el cuerpo del documento (ver algún número anterior como muestra).
3. Cuerpo del documento: El título de las Secciones será en MAYÚSCULAS, debidamente acentuadas. Las Subsecciones en minúsculas con la primera letra en mayúscula.
4. Las ecuaciones deberán indicarse lo más claramente posible, aun en formato txt. Por ejemplo: $H(\omega_1) = z^2 / (z - 0.5)$ where $z = e^{(j)\omega_1}$
5. Biografía de los autores. El lugar de adscripción se incluirá en una sección “acerca del autor” donde se podrá incluir una breve descripción del puesto que desempeña o ha desempeñado cada autor.

El autor principal deberá enviar debidamente llenado y firmado el formato de “cesión de derechos”, manifestando además que el trabajo no ha sido publicado previamente.

POLÍTICA EDITORIAL:

Los originales serán sometidos a un proceso editorial en varias fases. En primer lugar, los artículos recibidos serán objeto de una evaluación preliminar por parte del Comité Editorial, quien determinará la pertinencia de su publicación, con base a los requisitos temáticos. En la segunda fase, los artículos son enviados a dos pares académicos externos, quienes determinarán en forma anónima uno de los siguientes dictámenes: a) publicar sin cambios, b) publicar después de cumplir correcciones menores, c) publicar una vez que se haya revisado a fondo, d) rechazar. En caso de discrepancia entre los dos árbitros, el texto será enviado a un tercer árbitro, cuya decisión definirá si es aceptado o rechazado. Los resultados del proceso del dictamen son inapelables en todos los casos.

INSTRUCTIONS FOR AUTHORS

DIFU100ci@ (read difuciencia) is a quarterly publication of the Processing and Optical Instrumentation Academic Group, Faculty of Electrical Engineering, Universidad Autonoma de Zacatecas. It aims to disseminate scientific and technological knowledge in the field of engineering, through application oriented articles and articles showing research topics. The magazine has a certified copyright number for exclusive use of the title issued by INDAUTOR, Reserve: 04-2010-110314331900-102.

Aims and Scope: The articles can be classified in at least three categories: Divuligation, Research, and Tutorials.

Divuligation: The articles do not necessarily contain results of research projects themselves. This section will also include works that showcase new approaches to research subjects so as to broaden its distribution.

Research: The articles include research results while maintaining the focus on disclosure (divuligation), ie not necessarily with the rigor of a “paper”. This approach is intended to motivate even more to bring readers to the new topics that are studied today. Works on technological development can be considered in this category and it is recommended to have more articles related to this important task.

Tutorials: These include new issues, but little known. Alternatively, they may include new approaches to basic knowledge, with a didactic purpose, so that will expand knowledge and encourage its application in engineering/technological projects. Provisional Web site of the Magazine: <http://www.uaz.edu.mx/gmiram/Revista.htm>

Format for submmision: The work can be sent in .txt formats and graphics or pictures in .jpg format with good resolution. It will be acceptable to send a single .zip or .rar file containing both the text and figures. In the case of using word, it must be sent in .doc format without using macros (.docx format is not accepted). It will be also accepted as an OpenOffice file. It is not requested any page format, especially since the text will be brought to the format of the magazine.

The article should include:

1. Title.
2. List of authors.
3. Summary in a maximum of 200 words. The summary is used as the article description in the expanded index, and does not appear in the document body (see a previous issue as a sample).
4. Main body of document. The title of the section will be in UPPERCASE, properly spelled. Subsections must be written in lowercase with the first letter capitalized.
5. The equations should be indicated as clearly as possible, even in txt format. For example:
$$H(\omega_1) = z^2 / (z - 0.5) \text{ where } z = e^{(j)\omega_1}$$
6. Biography of the authors. The actual job position will be included in a section “about the author” which may include a brief description of the position played or had played each author.

The lead author must submit the duly completed and signed form of “transfer of rights”, saying that the work has not been published previously.

EDITORIAL POLICY:

Manuscripts will undergo an editorial process in several phases. First of all the items received will be subject to a preliminary assessment by the Editorial Committee, who will determine the relevance of its publication, based on thematic requirements. In the second phase, items are sent to two external academic peers, who determine anonymously one of the following opinions: a) accept unchanged, b) accept after serving minor corrections, c) accept once it has been fully reviewed, d) reject. In case of discrepancy between the two arbitrators, the text will be sent to a third arbitrator, whose decision will define if it is accepted or rejected. The results of the opinion process are final in all cases.

