ATAQUES DE SEGURIDAD EN LAS FPGAs Y COMO PREVENIRLOS.

Security Attacks in FPGAs and How to Prevent Them.

RESUMEN

Este documento es un estudio de algunos ataques de seguridad a los dispositivos lógicos de propósito general, concretamente a las FPGA (del inglés Field Programmable Gate Array), que se han encontrado. Actualmente es muy común escuchar hablar de este tipo de dispositivos, las FPGA´s y sus múltiples contextos de aplicación, entre los que se encuentran procesos de aceleración, criptografía, entre otros. Sin embargo, se desconocen los ataques que éstas pueden sufrir. Con este documento se pretende mostrar algunos de los ataques conocidos y algunas formas de prevenirlos.

PALABRAS CLAVES: FPGA, Ataques, curva eléctrica, prevenir, hardware.

ANA MARÍA LÓPEZ E.
Ingeniera Electricista
Docente
Universidad Tecnológica de Pereira
Maestría en Ingeniería
Estudiante
Universidad Pontificia Bolivariana de Medellín
anamayi@utp.edu.co

JESSICA ANDREA SANTA V.
Ingeniería de sistemas y computación.
Estudiante.
Universidad Tecnológica de Pereira.
jeansanta@utp.edu.co

ABSTRACT

This paper is a study of some of the physical attacks on hardware, specifically FPGAs, that have been found. Currently it is very common to hear talk of such devices, FPGA and its multiplex contexts of applications, among which you can find speeding up processes and cryptography. However, it is not known the diverse attacks that they can suffer. This document intent to show some of these attacks and some forms of prevention.

KEYWORDS: *FPGA, Attacks, Power Curve, Prevent, Hardware.*

1. INTRODUCCIÓN

Este artículo presenta información relacionada con los dispositivos lógicos programables FPGA´s, dado que su uso creciente como una alternativa de procesamiento específico los ubica como elementos que requieren alta confiabilidad.

Al ser cada día más usadas también se ven las dificultades que éstas tienen y al convertirse en herramientas importantes en desarrollos investigativos tanto educativos como comerciales se vuelven más frecuentes los ataques o imprevistos a los que no están ajenos por ser dispositivos lógicos.

Es por esto que se presentará primero información general sobre las FPGA´s, cómo están compuestas, cuál es su arquitectura, técnicas de programación, entre otros.

Luego se hace una descripción de los diferentes ataques conocidos, a los que son sometidas las FPGA´s, con el propósito de evidenciar las posibles problemáticas que pueden llegar a tener implementaciones de sistemas embebidos que usan dispositivos de lógica programable.

Por último se presenta una serie de recomendaciones sobre cómo superar cada uno de los diferentes tipos de ataques presentados, con el ánimo de que sean tenidas en cuenta en implementaciones que requieren una alta confiabilidad en la operación de los sistemas embebidos.

Se inicia por tanto con una descripción de las FPGA´s.

2. FPGA

[1] "Las FPGAs (del inglés Field Programmable Gate Array), son dispositivos lógicos de propósito general programables por los usuarios. Una FPGA está compuesta de bloques lógicos comunicados por conexiones programables. Este tipo de dispositivos cada vez son más cotidianos por su variedad de utilidades, entre ellas, servir de mecanismo para crear sistemas embebidos".

[2] Sistema embebido es el nombre genérico que reciben los equipos electrónicos que incluyen un procesamiento de datos, pero que están diseñados para satisfacer una función específica. El cerebro de un sistema embebido es típicamente un microcontrolador, aunque los datos también pueden ser procesados por un DSP, una FPGA, un microprocesador o un ASIC, y su diseño está optimizado para reducir su tamaño y su costo, así como también para aumentar su confiabilidad y mejorar su desempeño.

Fecha de Recepción: 13 de Mayo de 2011
Fecha de Aceptación: 15 de Julio de 2011

Actualmente se hace frecuente el uso de las FPGAs en este tipo de sistemas embebidos, porque aunque son más lentas que los ASICs y tienen mayor consumo de potencia, también cuentan con la gran ventaja de ser reprogramables y de tener menores costes de adquisición y desarrollo.

[2] "Un FPGA (Field Programmable Gate Array, Arreglo de Compuertas Programables en Campo) ofrece las mismas ventajas de un ASIC, sólo que a un menor costo; es decir, el costo por desarrollar un ASIC (Circuitos Integrados desarrollados para Aplicaciones Específicas) es mucho más alto que el que precisaría un FPGA, además de que tiene la ventaja de ser un circuito reprogramable, en el que es posible modificar o borrar alguna función programada sin alterar el funcionamiento del circuito. Y por el hecho de ser programables en campo significa que la función del dispositivo es definida por el usuario en lugar de definirla el fabricante".

2.1 ARQUITECTURA

[2] Una FPGA es un circuito integrado que contiene tres elementos fundamentales: bloques Lógicos configurables (CLB), bloques de entrada y salida (IOB) y canales de comunicación.

En la figura1 se presenta la arquitectura genérica de la FPGA. Una FPGA está compuesta por bloques lógicos configurables (CLBs) que se comunican a través de redes de interconexión programables (M) que incluyen los vecinos más cercanos como también largos caminos de cables jerarquizados. La periferia de la FPGA contiene bloques de interfaz de entrada/salida (IOBs) para conectar los bloques lógicos internos con los pines de entrada/salida.

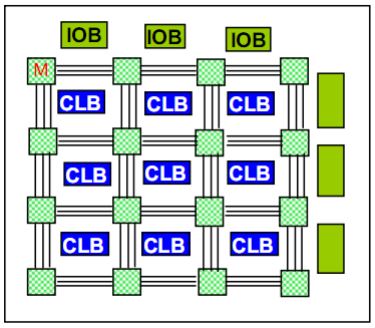

Figura 1. Esquema Interno de un FPGA. Fuente [8]

[3] El tamaño, la estructura, el número de bloques, la cantidad y conectividad de las conexiones varían en las distintas arquitecturas.

[8] Un ejemplo de lo planteado se puede observar en la figura 2, en la que se muestra un resumen de las capacidades de los dispositivos de la familia de dispositivos Spartan 3E. En esta se presenta entre otra información, la cantidad de bloques lógicos configurables (CLBs), número de bloque de entrada-salida (IOBs), número de bloques de RAM (BRAM), número de bloques multiplicadores, número de bloques administradores de reloj digital, entre otros.

Device	System Gates	Equivalent Logic Cells	CLB Array (One CLB = Four Slices)				Distributed RAM bits[1]	Block RAM bits[1]	Dedicated Multipliers	DCMs	Maximum User I/O	Maximum Differential I/O Pairs
			Rows	Columns	Total CLBs	Total Slices						
XC3S100E	100K	2,160	22	16	240	960	15K	72K	4	2	108	40
XC3S250E	250K	5,508	34	26	612	2,448	38K	216K	12	4	172	68
XC3S500E	500K	10,476	46	34	1,164	4,656	73K	360K	20	4	232	92
XC3S1200E	1200K	19,512	60	46	2,168	8,672	136K	504K	28	8	304	124
XC3S1600E	1600K	33,192	76	58	3,688	14,752	231K	648K	36	8	376	156

Figura 2. Características de la Familia de Dispositivos Spartan 3E [7]

[3] "Arreglo bidimensional de bloques lógicos rodeados por conexiones configurables. Una familia contiene idénticos bloques lógicos y conexiones, pero difieren en el tamaño del arreglo".

[3] Los CLBs, son celdas lógicas programables que están constituidas por uno o dos flipflops, multiplexores, una look up table (LUT) y las terminales de entrada, salida y control.

[3] Una LUT es una memoria digital que se encarga de almacenar la tabla de verdad de una función booleana. Los CLBs se pueden configurar para implementar funciones lógicas ya sean secuenciales o combinacionales.

2.2 TECNOLOGÍA DE PROGRAMACIÓN

En descripción de hardware se requiere comunicar al dispositivo programable, qué componentes utilizar y como realizar la conexión entre ellos para conseguir el comportamiento requerido, ésto se consigue a través de un conjunto de herramientas de diseño y descripción, pero especialmente a través de un lenguaje de descripción; para el trabajo con FPGA hay dos estándares de lenguajes utilizados, VHDL y Verilog, el segundo un poco de más alto nivel que el primero.

[3] "Se programa por la carga de celdas de memoria de configuración, que controlan la lógica e interconexiones".

2.3 CARACTERISTICAS DE LA FAMILIA DE FPGA

[14]Una familia contiene idénticos bloques lógicos y conexiones pero presenta diferencias en el tamaño del arreglo. Adicionalmente, cada una de las familias de dispositivos son diseñados con algunas capacidades particulares que las hacen más apropiadas para algunas tareas. Un ejemplo de esto es la familia Virtex II Pro y Virtex II Pro X, que contienen plataformas FPGA para diseños que están basados en núcleos IP y que además poseen módulos configurables. Esta familia permite la

implementación de soluciones completas en el área de las telecomunicaciones, redes inalámbricas, redes de datos, video y procesamiento digital de señales.

2.4 USOS COMUNES DE LAS FPGA´s

Las FPGA´s actualmente están siendo utilizadas en la implementación de sistemas digitales aplicables en muchas áreas, los autores de [8] consideran que esta posibilidad brinda un apoyo importante en procesos de investigación en el área del procesamiento digital de señales en tiempo real. También se ha demostrado en [9] mediante la implementación de un modulador sigma delta para el procesamiento de señales de voz, que esta se comportaba de acuerdo a lo esperado, pero que además puede ser usada en aplicaciones como generadores de ondas, muestreo de señales inalámbricas y mejoramiento de sistemas de medidas acústicas o de vibración.

La características de los sistemas FPGA relacionadas con la flexibilidad en los procesos de diseño están siendo aprovechadas en sistemas híbridos ASIC-FPGA, los cuales permiten tomar ventaja de las mejores capacidades de cada de las dos tecnologías.[10]

Los dispositivos de lógica programable son usados en aplicaciones que requieren gran cantidad de procesamiento como son los algoritmos criptográficos, en [11] se describe la implementación de un modulo de cifrado mediante el algoritmo hash MD5, en la cual se obtuvieron velocidades de procesamiento de 1Gbps, demostrando que esto permite tener procesos de comunicación cifrada sin problemas de latencia.

En [12] se puede observar como los dispositivos de lógica programable son usados para implementar técnicas de inteligencia artificial, específicamente el método para emular el chip ADN con FPGA y su aplicación en sistemas de potencia en detección de fallas. Adicionalmente, en [13] se muestra una implementación de técnicas de lógica de difusa en FPGA´s.

Con base en lo planteado, se ha demostrado la versatilidad de las FPGA´s en términos de los campos de aplicación, que básicamente no tienen fronteras, razón por la cual son cada vez más usadas en un sinnúmero de novedosas aplicaciones en diversos contextos. Lo anterior, genera que a medida que se depende más de los sistemas de lógica programable, se empiezan a presentar ataques a estos dispositivos con el propósito de lograr acceder a la información que estos procesan. Por lo tanto, en la siguiente sección se presentan los diferentes ataques conocidos que se realizan a los dispositivos de lógica programable, con el ánimo de mostrar las dificultades que se pueden presentar pero con la expectativa de generar en los diseñadores conciencia sobre estos asuntos.

3. ATAQUES A FPGA´s

Por muchos motivos como costos o facilidad de acceso a las FPGA, estos dispositivos se vuelven cada vez más comunes y atractivos para ser usados en proyectos con sistemas embebidos o en criptografía. Sin embargo, esto también propicia que se vuelvan más comunes los ataques a estos dispositivos. A la fecha es poco conocido cuál de los posibles ataques puede afectar una implementación. A continuación se presenta una recopilación de cada uno de los ataques conocidos.

3.1 ATAQUE DE RELECTURA

La gran mayoría de familias de FPGA tiene la característica de que se permite leer la configuración de la FPGA para tener la posibilidad de una depuración más simple, la idea del atacante en beneficiarse de esta característica, leyendo la configuración a través del JTGo (Interfaz de programación) con el fin de acceder a la información importante o secreta. [3] La funcionalidad de colación se puede prevenir con elementos de seguridad que ofrece la forma de fabricación del dispositivo. Sin embargo, se puede concebir que un atacante sobrepase estas medidas de seguridad mediante un proceso de inyección de fallas.

[3]Dado lo anterior, un atacante es capaz de desactivar los bits de seguridad y/o las contramedidas. Lo cual resulta en la capacidad de lectura de la configuración de la FPGA.

3.2 ATAQUE DE CANAL LATERAL

[3][4]Como los ataques mencionados anteriormente, el ataque de análisis diferencial de canal lateral (DSCA) se usa para la seguridad de dispositivos criptográficos, cualquiera de estos es vulnerable a que existan fugas de información no deseada por un lado del canal, y las FPGA no están exentos de tener estas fugas, pero concretamente este tipo de ataque es de los más poderosos. Este analiza el consumo de energía o electromagnético, el comportamiento de la sincronización y la radiación, depende de los datos procesados en momentos fijos de tiempo. Para el ataque, se buscan los resultados medios del algoritmo, que dependen de la clave secreta. Entonces, el poder del consumo se mide y se compara con el hipotético consumo de energía de estos valores intermedios.

Sin embargo, parece también cierto que los canales pueden ser explotados en diferente sentido, tal como es el caso de las FPGAs.

3.3. CLONACIÓN DE FPGAS SRAM

[3] Normalmente al ingresar los datos de configuración se almacenan en la memoria no volátil, que es aquella en la que están basadas las memorias ROM (Memoria de solo lectura). Existen de dos tipos, reprogramables y las

no reprogramables. Los datos se trasmiten en el encendido a la FPGA, con el fin de configurarla. Como es claro los atacantes se aprovechan de las debilidades de los sistemas, y en este caso se aprovechan por medio de un proceso de intercepción de la trasmisión, para así conseguir el archivo de configuración. Este ataque es posible tanto para las grandes organizaciones, así como para aquellos con presupuestos bajos y modestos.

3.4 ATAQUE DE CAJA NEGRA

[3] Este tipo de ataque consiste en el método clásico que existe para descompilar un chip, este es llamado ataque de caja negra, consiste en que el atacante ingresa todas las posibles combinaciones de entrada y la guarda en la correspondiente salida. Con la ayuda de algoritmos que simplifiquen las tablas obtenidas o los famosos mapas de Karnaugh, el cual debe su nombre a su inventor Maurice Karnaugh en 1950, y consiste en diagramas utilizados para la simplificación de funciones algebraicas booleanas. En este caso, los atacantes pueden extraer la lógica interna de la FPGA.

[3]Este ataque sólo es factible si una FPGA pequeña con entradas y salidas explícitas es atacada y una gran cantidad de energía del procesador está disponible.

3.5 TÉCNICAS DE INGENIERÍA INVERSA DE LOS FLUJOS DE BITS

Los ataques descritos hasta el momento hablan del flujo de bits de salida de las FPGAs, todo esto con el fin de generar algún ataque para sustraer información como el diseño de los algoritmos o claves de seguridad, para esto se realiza ingeniería inversa al flujo de bits, en este caso el flujo de bit no solo tiene que estar en posesión del atacante si no también que no esté cifrado es decir debe estar desencriptado.

Las FPGA manejaban la seguridad en la cadena de bits en la que llevan los datos de configuración y esta información solo era asequible con un acuerdo de confidencialidad firmado. [4]Pero hace diez años se rompió este tipo de seguridad, cuando la empresa CAD NEOCad, aplicó ingeniería inversa en una FPGA de Xilinx, esta empresa fue capaz de reconstruir la información necesaria sobre tablas de las operaciones, las conexiones y elementos de almacenamiento. Por lo tanto, NEOCad fue capaz de producir software de diseño sin la firma de acuerdos confidencialidad con el fabricante de FPGA. A pesar del gran esfuerzo que se debe hacer para alterar el diseño del flujo de bits, para las grandes organizaciones es muy factible.

3.6 ATAQUE FISICO

[3][4]Al igual que la técnica de ingeniería inversa en los flujos de bits, el ataque físico tiene como objetivo conseguir información sobre los algoritmos de propiedad o sobre las contraseñas secretas, por punto de sondeo en el chip, pero este tiene como objetivo las partes que no están disponibles usualmente, como son las entradas y salidas de la FPGA o jumpers que están visibles.

[3] Esto puede potencialmente ser logrado a través de las inspecciones visuales y utilizando herramientas tales como microscopios ópticos y sondas mecánicas. Sin embargo, las FPGAs se están volviendo tan complejas que sólo con métodos avanzados, tales como el proceso de Enfocado Ion Beam (FIB) de sistemas, se podría lanzar un ataque.

4. PREVENCION DE LOS ATAQUES

Se presentan algunas medidas que pueden contrarrestar el efecto de los ataques, cuando es programada la FPGA pero muchos de ellos necesitarían ser realizados por el fabricante, ya que se requieren cambios en el diseño de las tarjetas.

4.1 PREVENCION DEL ATAQUE DE RELECTURA

[3] Este tipo de ataques se puede evitar con los bits de seguridad establecidos, configurándolo correctamente el bit de seguridad, como se indique por los fabricantes. Como se mencionó anteriormente en el literal 3.1, en la mayoría de familias de FPGA se proporciona la característica que permite leer la configuración de la FPGA para facilitar la depuración. Si se quiere estar seguro de que un atacante no es capaz de provocar la inyección de un fallo, la FPGA se debe encontrar en un ambiente seguro, donde se detecte una interferencia en toda la configuración o se destruye la FPGA, es decir se retire del entorna esta FPGA o se cambie, para que no intervenga mas ya que se encuentra infectada, también se podría detener y reconfigurar ya que es una de las características de las FPGA.

4.2 PREVENCION AL ATAQUE DE CANAL LATERAL

[3] En los últimos años, se ha realizado mucho trabajo para prevenir los ataques de canal lateral.

Existe software que hace referencia a los cambios algorítmicos, estas medidas de hardware tratan de suavizar el rastro de energía o los cambios de nivel de la lógica del transistor. Este es otro tipo de ataque que es complejo de prevenir y que requiere el apoyo de los fabricantes, este tipo de ataque puede contribuir a la inclusión de mejoras en los diseños de futuras arquitecturas de las FPGAs.

4.3 PREVENCION DEL ATAQUE DE CLONACIÓN DE FPGAS SRAM

Para este tipo en la investigación que se realizó se encuentran diferentes recomendaciones dadas por los autores, ellas serán nombradas a continuación, ya que son útiles y pertinentes debido a la explicación dada anteriormente.

[3]Una de las sugerencias es vista con la idea de evitar la ingeniería inversa, ésta consiste en comprobar el número de serie antes de ejecutar el diseño o en eliminar circuitos que no han sido diseñados con los parámetros de seguridad.

[3] Otra solución más realista sería la memoria no volátil o un chip, o tal vez la configuración de las dos. La medida más efectiva es el cifrado del archivo de configuración.

[3] Y por último otra solución sería la de suministrar energía a toda SRAM FPGA con una batería, lo que haría la transmisión del archivo de configuración después de una pérdida de energía innecesario. Por lo tanto, una combinación de encriptación y energía de la batería ofrece una posible solución.

4.4 PREVENCION DEL ATAQUE DE CAJA NEGRA

[3] Hoy en día los diseños son cada vez más complejos, seguros y de mayor tamaño, por tal motivo los ataques de caja negra son menos peligrosos o amenazantes debido a que con simples evaluaciones de ingreso y salida de datos no es fácilmente obtenible la lógica interna del sistema. También la naturaleza de los algoritmos de cifrado evita el ataque, puesto que en general tiene una longitud de cifrado de salida menor, al igual que sus longitudes de claves.

4.5 PREVENCION DE TÉCNICAS DE INGENIERÍA INVERSA DE LOS FLUJOS DE BITS

[3][4]Dado que el ataque de ingeniería inversa se basa en el uso del flujo de bits de salida, se recomienda garantizar la privacidad de dichas cadenas de bits. Una manera de lograr lo anterior, es llevar a cabo un proceso de cifrado que permita que la cadena de bits de salida no sea fácilmente legible por un atacante que se encuentre escuchando ó tomando copia de los flujos de bit de salida.

4.6 PREVENCION AL ATAQUE FISICO

[3] Para evitar los ataques físicos, se debe asegurar que los efectos de retención de las celdas sean tan pequeños como sea posible, por lo que un atacante no podría detectar el estado de las celdas. Ya después de almacenar un valor de una celda de memoria SRAM para el 100-500 segundos, el tiempo de acceso y la tensión de funcionamiento va a cambiar. La solución sería invertir los datos almacenados de forma periódica o para mover los datos en torno de la memoria. La neutralización del efecto de retención se puede lograr mediante la aplicación de una corriente opuesta o la inserción de los ciclos simulados en el circuito. En cuanto a las aplicaciones de FPGAs, es muy costoso o muy difícil, incluso para ofrecer soluciones como la que invierte los bits o cambiar la ubicación para el archivo de configuración general. Una posibilidad podría ser que esto se hace sólo para la parte fundamental del diseño, como las claves secretas. Contrarrestar técnicas tales como los ciclos simulados y actual enfoque contrario se puede llevar a hacia adelante a las aplicaciones FPGA.

FPGAs sólo puede ser protegido contra el ataque físico, mediante la construcción de un entorno seguro que le rodee. Si un ataque se detectó todas las celdas deben ser programadas para que no se filtre ninguna información.

En términos de la flash, celda de memoria EEPROM, hay que considerar que el proceso de la primera escritura - borrado ciclos, produce un cambio mayor en el umbral de la celda y que este efecto será menos evidente después de diez ciclos de escritura - borrado. Por lo tanto, se debe programar la FPGA cerca de 100 veces con datos aleatorios para evitar estos efectos.

3. CONCLUSIONES Y RECOMENDACIONES

Las FPGA en general son seguras, todo depende del fabricante que se escoja y de cómo se manejan las configuraciones de seguridad con que éstas son fabricadas, las cuales son usualmente poco usadas, pero igualmente importantes.

El entorno en el que se maneje la FPGA juega un papel importante para la seguridad de los códigos y contraseñas.

Con base en lo planteado en [2] se puede plantear que por costos y eficiencia es mejor el uso de las FPGA´s, además aportan casi las mismas características que otros dispositivos de propósito general.

Conocer este tipo de ataques permite diseñar procesos de prevención en varios entornos, ya sea laboral o académico puesto que muchos de éstos podrían aplicarse a otra clase de hardware o dispositivos.

Es importante conocer la existencia de los diferentes tipos de ataques presentados en el documento, porque esto ayudará a generar una nueva cultura de seguridad en los procesos de implementación de mejoras a la velocidad de procesamiento. Es decir, ahora será necesario, no sólo hacer diseños eficientes, sino además tener en cuenta técnicas de seguridad para contrarrestar las posibles vulnerabilidades existentes.

4. BIBLIOGRAFÍA

[1]Sánchez Suárez Gabriel. Presentación Microelectrónica. Disponible en: http://www.ufps.edu.co/materias/uelectro/htdocs/pdf/fpga.pdf. Fecha de visita Abril 15 de 2011.

[2] Aguilar López/ Jenny Elena. Arquitectura en un Fpga para Encriptación Aes. Tesis Profesional para Obtener el Grado de: Licenciado en Ciencias de la Computación. Benemérita Universidad Autónoma de Puebla Facultad de Ciencias de la Computación. Disponible en: http://perseo.cs.buap.mx/bellatrix/tesis/TES734.pdf Fecha de visita Marzo 19 de 2011.

[3] Thomas Wollinger and Christof Paar. How Secure Are FPGAs in Cryptographic Applications?. In 13th International Conference on Field Programmable Logic and Applications - FPL 2003, Lisbon, Portugal, September 1-3, 2003. Disponible en: https://springerlink3.metapress.com/content/vmg8mv38xrert5ql/resource-secured/?target=fulltext.pdf&sid=3qbj52jjf3mpk455voe5yw45&sh=www.springerlink.com Fecha de visita Abril 8 de 2011

[4 Thomas Wollinger, Jorge Guajardo And Christof Paar. Security On Fpgas: State Of The Art Implementations And Attacks. ACM Transactions on Embedded Computing Systems (TECS), Volume 3 Issue 3, August 2004. Disponible en: http://portal.acm.org/citation.cfm?id=1015052 Fecha de visita 19 Marzo de 2011.

[5] Chang Kyun, Martin Schlaffer, y Sang Jae Moon. Differential Side Channel Analysis Attacks on FPGA Implementations of ARIA. ETRI Journal, Volume 30, Number 2, April 2008. Disponible en: http://etrij.etri.re.kr/Cyber/Download/PublishedPaper/3002/30-02-16.pdf. Fecha de visita Abril 15 de 2011.

[6] YongBin Zhou, DengGuo Feng. Side-Channel Attacks: Ten Years After Its Publication and the Impacts on Cryptographic Module Security Testing. Academia de Ciencias de China, Beijing, 100080, China. 2005. Disponible en: http://eprint.iacr.org/2005/388.pdf. Fecha de visita 19 de Abril de 2011

[7] Xilinx. Spartan 3E FPGA Family Data Sheet. Agosto de 2009. Disponible en: http://www.xilinx.com/support/documentation/data_sheets/ds312.pdf. Fecha de visita Abril 25 de 2011

[8] Alba Blanco Emiliano F. Implementación de filtros digitales en FPGA. VII Congreso de la Sociedad Cubana de Bioingeniería, Habana 2007. Disponible en: http://cencomed.sld.cu/socbio2007/trabajos/pdf/t093.pdf Visitado 13 de Mayo de 2011

[9] Delgadillo Anthony, Mendez Luis, Melgarejo Miguel. Modulador Sigma Delta en FPGA para el Procesamiento de Señales de Voz. Grupo De Investigación En Lógica Programable Y Técnicas Digitales. Universidad Distrital Francisco Jose De Caldas. Bogotá, Colombia. Disponible en:http://148.202.12.20/materias/ET201/modulo_07/aplicaciones/sigma_delta.pdf Fecha de visita 12 de Mayo de 2011

[10] Zuchowski/ Paul S., Reynolds/ Chistopher B., Grupp Richard J., Davis/ Shelly G., Cremen/ Brendan, Troxel/ Bill. A Hybrid ASIC and FPGA Architecture. ICCAD 02 Proceedings of the 2002 IEEE/ACM international conference on Computer-aided design. Disponible en: http://portal.acm.org/citation.cfm?id=774600 Fecha de visita Mayo 13 de 2011.

[11] Ignacio Algredo Badillo, René Armando Cumplido Parra, Claudia Feregrino Uribe. Desarrollo de un Módulo MD5 para un Sistema Criptográfico Reconfigurable en un FPGA. Coordinación de Ciencias Computacionales, Instituto Nacional de Astrofísica Óptica y Electrónica, INAOE. Disponible en: http://ccc.inaoep.mx/~rcumplido/papers/ReCOnFig04%20-%20MD5.pdf Fecha de visita Mayo 12 de 2011.

[12] D. Perlaza, A. Delgado. Detección de Fallas en Sistemas de Potencia con Chip ADN en FPGA. Departamento de Ingeniería Eléctrica y Electrónica Universidad Nacional de Colombia, Bogotá. Disponible en:http://alumnos.elo.utfsm.cl/~rlopeza/automatizacion/Deteccion%20de%20fallas%20de%20instrumentacion/instrumentacion%20cadenas%20ADN.pdf Fecha de visita Mayo 13 de 2011.

[13] Gerardo Aranguren Aramendia, Miguel Rodriguez Gomez, Mariano Barron Ruiz. Controlador Secuencial Segmentado para la Logica Borrosa Implementado en FPGA. Disponible en: http://www.softcomputing.es/estylf08/es/1996VI%20Congreso/II%20Jornada%20sobre%20Transferencia%20de%20Tecnolog%C3%ADa%20Fuzzy/II%20Jornada%20sobre%20Transferencia%20de%20Tecnolog%C3%ADa%20Fuzzy.pdf Fecha de visita Mayo 13 de 2011.

[14] Xilins. “Virtex II Pro y Virtex II Pro X Platform FPGA: Complete Data Sheet”. Noviembre de 2007. Disponible en: http://www.xilinx.com/support/documentation/datasheets/ds083.pdff. Fecha de visita Abril 25 de 2011