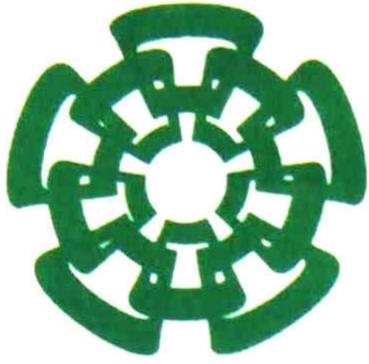


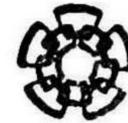


xx(178861.1)



Centro de Investigación y de Estudios Avanzados del I.P.N.  
Unidad Guadalajara

# **Optimización de códigos y dispositivos programables para las tarjetas de validación de Intel**



CENTRO DE INVESTIGACIÓN Y  
DE ESTUDIOS AVANZADOS DEL  
INSTITUTO POLITÉCNICO  
NACIONAL

COORDINACIÓN GENERAL DE  
SERVICIOS BIBLIOGRÁFICOS

Tesis que presenta:  
**Juan Pablo Vázquez Barragán**

para obtener el grado de:  
**Maestro en Ciencias**

en la especialidad de:  
**Ingeniería Eléctrica**

Director de Tesis  
**Dr. José Luis Leyva Montiel**

**CINVESTAV  
IPN  
ADQUISICION  
DE LIBROS**

Guadalajara, Jalisco, Abril de 2009.

CLASIF: TK165.G8 .V39 2009  
ADQUIS: SSI-558  
FECHA: 2-SCP-09'  
PROCES: DON-09'

ID: 160801 -1001

# **Optimización de códigos y dispositivos programables para las tarjetas de validación de Intel**

**Tesis de Maestría en Ciencias  
Ingeniería Eléctrica**

Por:

**Juan Pablo Vázquez Barragán**  
Ingeniero en Electrónica

Instituto Tecnológico y de Estudios Superiores de Occidente  
1998-2002

Becario de CONACYT, expediente no. 180899

Director de Tesis  
**Dr. José Luis Leyva Montiel**

CINVESTAV del IPN Unidad Guadalajara, Abril de 2009.

# Resumen

---

El objetivo de este trabajo de tesis fue el desarrollo de Hardware y Firmware para optimizar los costos y agregar mayores capacidades de observabilidad y controlabilidad en las tarjetas de validación para chipsets de Intel®. Durante la realización de este trabajo se realizaron diferentes estudios para reducir los altos costos que involucran el diseño y la fabricación de una tarjeta de validación, se analizaron los requerimientos recibidos por los grupos de validación y se hicieron optimizaciones de códigos, análisis de diferentes familias de dispositivos programables en el mercado, capacidades, compilaciones y se determinó la familia óptima de dispositivos para reducir los costos y aumentar las capacidades lógicas internas de cada dispositivo programable.

Los resultados que se obtuvieron con la realización de este proyecto trajeron un ahorro de 122 dólares por tarjeta. Generalmente la producción de tarjetas de validación no es tan alta debido a sus altos costos y complejidades, además que solo son usadas por grupos internos de Intel. En este proyecto se fabricaron 270 tarjetas, en las cuales se ahorraron 32,940 dólares en total con los cambios realizados y la nueva arquitectura propuesta, además que se incrementaron las capacidades de observabilidad y controlabilidad, ya que se crearon nuevas rutinas para automatizar pruebas que anteriormente eran hechas manualmente o no eran realizadas debido a que no se tenía el control de ciertos eventos en la tarjeta. Estas nuevas rutinas redujeron en un 20% el tiempo de validación para el grupo de potencia (PV) y cubrieron un mayor número de casos en sus pruebas.

La metodología seguida en este proyecto comienza con un breve estudio de la arquitectura de Intel®, se plantea el problema a resolver, se hacen los análisis, optimizaciones de códigos, compilaciones y simulaciones de los diferentes dispositivos programables usados en las tarjetas de validación. Posteriormente se analizan los requerimientos recibidos por los grupos de validación y se presenta una solución que les permite automatizar varias de sus pruebas a través de rutinas en los dispositivos programables. Se realiza un estudio de factibilidad donde se muestran las reducciones de costos, riesgos, beneficios y el proyecto es aprobado por la gerencia. Finalmente se hacen las pruebas con el hardware diseñado, se presentan los resultados obtenidos y trabajo a futuro.

Friday, February 07 2003

CNDA# 1172512

**CORPORATE NON-DISCLOSURE AGREEMENT**

This Corporate Non-Disclosure Agreement ("Agreement") is entered into and made effective as of the date set forth above, by and between Intel Corporation and its majority owned worldwide subsidiaries ("Intel"), and the Participant, as identified below, and its majority owned worldwide subsidiaries (the "Participant"). Unless the Participant indicates on the Participant's line below that this Agreement will apply only to a specific division or location, this Agreement will apply to the Participant's entire company worldwide.

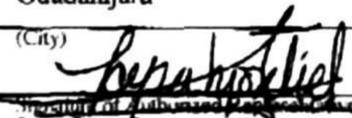
**THE PARTIES AGREE AS FOLLOWS:**

1. **Confidential Information.** The confidential, proprietary and trade secret information of the disclosing party ("Confidential Information") to be disclosed hereunder is (i) information in tangible form that bears a "confidential," "proprietary," "secret," or similar legend, and (ii) discussions relating to that information whether those discussions occur prior to, concurrent with, or following disclosure of the information.
2. **Obligations of Receiving Party.** The receiving party will maintain the confidentiality of the Confidential Information of the disclosing party with at least the same degree of care that it uses to protect its own confidential and proprietary information, but no less than a reasonable degree of care under the circumstances. The receiving party will not disclose any of the disclosing party's Confidential Information to any employees or to any third parties except to the receiving party's employees, parent company and majority-owned subsidiaries who have a need to know and who agree to abide by nondisclosure terms at least as comprehensive as those set forth herein; provided that the receiving party will be liable for breach by any such entity. For the purposes of this Agreement, the term "employees" shall include independent contractors of each party. The receiving party will not make any copies of the Confidential Information received from the disclosing party except as necessary for its employees, parent company and majority-owned subsidiaries with a need to know. Any copies which are made will be identified as belonging to the disclosing party and marked "confidential", "proprietary" or with a similar legend.
3. **Period of Non-Assertion.** The disclosing party will not assert any claims of breach of this Agreement or misappropriation of trade secrets against the receiving party arising from the receiving party's disclosure of the disclosing party's Confidential Information made more than five (5) years from the date of the disclosure. However, unless at least one of the exceptions set forth in Section 4 below has occurred, the receiving party will continue to treat such Confidential Information as the confidential information of the disclosing party and only disclose any such Confidential Information to third parties under the terms of a non-disclosure agreement.
4. **Termination of Obligation of Confidentiality.** The receiving party will not be liable for the disclosure of any Confidential Information which is:
  - (a) rightfully in the public domain other than by a breach of a duty to the disclosing party;
  - (b) rightfully received from a third party without any obligation of confidentiality;
  - (c) rightfully known to the receiving party without any limitation on use or disclosure prior to its receipt from the disclosing party;
  - (d) independently developed by employees of the receiving party; or
  - (e) generally made available to third parties by the disclosing party without restriction on disclosure.
5. **Title.** Title or the right to possess Confidential Information as between the parties will remain in the disclosing party.
6. **No Obligation of Disclosure; Termination.** Neither party has any obligation to disclose Confidential Information to the other. Either party may terminate this Agreement at any time without cause upon written notice to the other party; provided that each party's obligations with respect to Confidential Information disclosed during the term of this Agreement will survive any such termination. Either party may, at any time: (a) cease giving Confidential Information to the other party without any liability, and/or (b) request in writing the return or destruction of all or part of its Confidential Information previously disclosed, and all copies thereof, and the receiving party will promptly comply with such request, and certify in writing its compliance.
7. **General.**
  - (a) This Agreement is neither intended to nor will it be construed as creating a joint venture, partnership or other form of business association between the parties, nor an obligation to buy or sell products using or incorporating the Confidential Information.
  - (b) Both parties understand and acknowledge that no license under any patents, copyrights, trademarks, or maskworks is granted to or conferred upon either party in this Agreement or by the disclosure of any Confidential Information by one party to the other party as contemplated hereunder, either expressly, by implication, inducement, estoppel or otherwise, and that any license under such intellectual property rights must be express and in writing.
  - (c) The failure of either party to enforce any right resulting from breach of any provision of this Agreement by the other party will not be deemed a waiver of any right relating to a subsequent breach of such provision or of any other right hereunder.
  - (d) This Agreement will be governed by the laws of the State of Delaware without reference to conflict of laws principles, if any.
  - (e) This Agreement constitutes the sole and entire agreement between the parties with respect to the Confidential Information and all restrictions thereon; it supersedes any and all prior or contemporaneous oral or written agreements, negotiations, communications, understandings and terms, whether express or implied regarding the Confidential Information, and may not be amended except in a writing signed by a duly authorized representative of the respective parties. Any other agreements between the parties, including non-disclosure agreements, will not be affected by this Agreement.
  - (f) The disclosing party disclaims all warranties regarding all Confidential Information disclosed pursuant to this Agreement, including all warranties as to the accuracy or utility of such Confidential Information.

INTEL CONTACT: Ruben Ruvalcaba M/S: TLP-1 TEL NO: (33) 3540-6042

AGREED:  
INTEL CORPORATION  
2200 Mission College Blvd.  
Santa Clara, CA 95052-8119

  
**Mike Splinter**  
Executive Vice President  
Director, SMG

PARTICIPANT:  
Centro de Tecnologia de Semiconductores  
(Company Name, Division/Sub, if applicable)  
Lopez Mateos Sur 590  
(Street Address)  
Guadalajara Jalisco/Mexico 45090  
(City) (State/Country) (Postal Code)  
  
(Signature of Authorized Representative (e.g. President or V.P.))  
Jose Luis Leyva / Director  
Printed Name & Title

**NOTE: TO ENABLE QUICK EXECUTION, INTEL HAS PRE-SIGNED THIS STANDARD CNDA. ANY MODIFICATIONS TO THIS CNDA WILL RENDER THIS CNDA AND INTEL'S SIGNATURE NULL AND VOID.**

PLEASE SEND THIS COPY TO YOUR LOCAL NDA ADMINISTRATOR AT  
US - INTEL CORPORATION, ATTN: Post Contract Mgmt, EM6-03, 1900 Prunick City Road, Folsom, CA 95630-9598  
EMEA - INTEL CORPORATION, ATTN: EMEA Legal, ISW 74, Piper's Way, Swindon, Wiltshire, United Kingdom, SN2 1FU  
APAC - Asia Legal Team, INTEL TECHNOLOGY SDN BHD (PG9) Bayan Lepas Free Industrial Zone, Phase 3, Halaman Kangkong Jawa, 11900 Penang, Malaysia (rev. 04/02)

## Dedicatoria y agradecimientos

---

*Cuando pienso en los factores que me llevaron a estudiar esta maestría se me vienen a la mente ciertos eventos que me llevaron a tomar la decisión de hacerlo, pero el más importante y estoy seguro que es el que me ha ayudado a seguir adelante en mi carrera ha sido el ejemplo de mi Padre Federico. Siempre que hago memoria de mi niñez recuerdo a mi padre trabajando para que nunca nos faltara nada, lo recuerdo siempre ayudando a los demás y guiándonos con el ejemplo de trabajo, respeto, responsabilidad y voluntad para hacer las cosas. También volteo hacia atrás y veo a mi Madre Isabel desmandándose por nosotros y ayudándonos siempre con las tareas, el estudio y mil cosas más. ¡Dios los llene de bendiciones a ambos!*

*Gracias a todos los profesores del CINVESTAV, principalmente a los doctores Deni Torres, J. Luis Leyva, J. Luis Naredo, Eduardo Bayro y Antonio Treviño, por compartir sus experiencias, por guiarme durante sus clases y porque siempre me dieron un consejo para salir adelante en la vida como estudiante de maestría y fuera de ella. Al personal de CTS, Jorge Gamboa, Mario Muñoz, Jorge de la Torre, Armando Govea y Miguel Ángel Ramírez por su apoyo y aprendizajes compartidos en el periodo de estudios de maestría.*

*Gracias a Intel por permitirme presentar esta información y poder titularme con un proyecto trabajando esta compañía. En especial a Jesús Palomino, Roberto Castañeda y Javier Camino, por ayudarme a definir los criterios a presentar para esta tesis y las revisiones hechas al material presentado.*

*Muchas gracias a mi esposita Amy, por toda la paciencia, apoyo y comprensión durante todos estos meses que he trabajado en la tesis (fines de semana y hasta tardes horas entre semana). ¡Ya pronto tendremos un poco de mas tiempo para compartirlo juntos!*

*Si tratara de incluir a todas las personas que me dieron su apoyo y me ayudaron a lograr esta meta terminaría con una lista enorme, así que resumiré esta lista dando gracias a todos mis familiares, amigos y profesores por todo este tiempo que han estado apoyándome.*

# INDICE

<b>RESUMEN</b> .....	<b>i</b>
<b>CORPORATE NON-DISCLOSURE AGREEMENT</b> .....	<b>ii</b>
<b>DEDICATORIA Y AGRADECIMIENTOS</b> .....	<b>iii</b>
<b>INDICE</b> .....	<b>iv</b>
<b>LISTA DE FIGURAS</b> .....	<b>vi</b>
<b>1. INTRODUCCIÓN</b> .....	<b>2</b>
<b>2. ESQUEMA GENERALIZADO DE UNA PC</b> .....	<b>5</b>
2.1. PROCESADOR.....	6
2.2. CONTROLADOR DE MEMORIA Y GRÁFICOS (MCH).....	6
2.3. CONTROLADOR DE I/OS (ICH).....	6
2.4. CONTROLADOR SÚPER I/O.....	6
2.5. INTERFAZ TPM.....	6
2.6. CONTROLADOR DEL SISTEMA.....	6
2.7. CONTROLADOR DE AUDIO.....	7
2.8. CONTROLADOR DE RED (LAN).....	7
2.9. BUS DEL PROCESADOR (FSB).....	7
2.10. BUS DE MEMORIA.....	7
2.11. BUS PCI.....	7
2.12. BUS PCIE.....	7
2.13. BUS LPC.....	8
2.14. SYSTEM MANAGEMENT BUS.....	8
2.15. BUS SPI.....	8
2.16. INTERFASE DE BUS DMI.....	8
<b>3. CARACTERÍSTICAS DE LAS PLATAFORMAS DE VALIDACIÓN INTEL</b> .....	<b>10</b>
3.1. INTRODUCCIÓN.....	10
3.2. CONCEPTOS USADOS EN LA VALIDACIÓN DE CPUS Y CHIPSETS.....	11
3.2.1. <i>Pre-silicon architecture validation</i> .....	12
3.2.2. <i>Post-silicon system validation</i> .....	12
3.2.3. <i>Post-silicon platform compatibility validation</i> .....	12
3.3. CAPACIDADES DE LAS PLATAFORMAS DE VALIDACIÓN.....	13
3.3.1. <i>Observabilidad</i> .....	13
3.3.2. <i>Controlabilidad</i> .....	14
3.3.3. <i>Configurabilidad</i> .....	15
3.3.4. <i>Medibilidad</i> .....	16

3.4.	ARQUITECTURA GENERAL DE UNA TARJETA DE VALIDACIÓN DESKTOP .....	16
<b>4.</b>	<b>PLANTEAMIENTO DEL PROBLEMA .....</b>	<b>20</b>
4.1.	BREVE RESEÑA DEL GRUPO DE DISEÑO DE PLATAFORMAS EN GDC (VHG).....	20
4.2.	OPORTUNIDAD DE MEJORA DETECTADA EN UNA PLATAFORMA DESKTOP (SB).....	21
4.3.	VENTAJAS EN LA OPTIMIZACIÓN DE LOS DISPOSITIVOS PROGRAMABLES.....	23
4.4.	DESVENTAJAS EN LA OPTIMIZACIÓN DE LOS DISPOSITIVOS PROGRAMABLES.....	23
<b>5.</b>	<b>ESTUDIO DE FACTIBILIDAD .....</b>	<b>26</b>
5.1.	DISPOSITIVOS PROGRAMABLES .....	26
5.2.	DISPOSITIVOS DE ALTERA.....	27
5.2.1.	<i>Dispositivos Altera MAXII</i> .....	27
5.2.2.	<i>Dispositivos Altera CycloneII</i> .....	34
5.3.	DISPOSITIVOS DE XILINX .....	35
5.3.1.	<i>Dispositivos Xilinx Cool Runner-II CPLD</i> .....	36
5.3.2.	<i>Dispositivos Xilinx Spartan3 FPGA</i> .....	38
5.4.	CÓDIGOS COMPILADOS PARA CADA DISPOSITIVO EVALUADO.....	38
<b>6.</b>	<b>SOLUCIÓN IMPLEMENTADA .....</b>	<b>43</b>
6.1.	ARQUITECTURA DE LA TARJETA DE VALIDACIÓN ICH10 .....	43
6.2.	DISPOSITIVO HOOKS PLD.....	43
6.2.1.	<i>Bloques P80 y FSC</i> .....	44
6.2.2.	<i>Bloque SERIRQ</i> .....	45
6.2.3.	<i>Bloque MONITOR</i> .....	46
6.2.4.	<i>Bloque SMBus</i> .....	47
6.3.	DISPOSITIVO RESET PLD .....	47
<b>7.</b>	<b>DATOS Y RESULTADOS .....</b>	<b>51</b>
7.1.	PLAN DE PRUEBAS DE LA TARJETA DE VALIDACIÓN ICH10.....	51
7.2.	PRUEBAS FUNCIONALES A LOS NUEVOS PLDs (MAXII).....	53
7.3.	POWER-ON DE LA TARJETA ICH10.....	54
<b>8.</b>	<b>CONCLUSIONES .....</b>	<b>57</b>
	<b>REFERENCIAS .....</b>	<b>59</b>

# Lista de Figuras

---

## Capítulo 2

- 2.1 Diagrama a bloques de la arquitectura de una computadora (Desktop)

## Capítulo 3

- 3.1 Diagrama que muestra una tarjeta de validación para cuatro CPUs y una tarjeta para FBD
- 3.2 Diferentes tarjetas son usadas para inyectar tráfico en los buses de PCIE y SATA en una plataforma de validación.
- 3.3 Diagrama mecánico de chasis de una plataforma de validación que soporta 2 o 4 procesadores
- 3.4 Tarjeta de medición de voltaje y corriente.
- 3.5 Arquitectura a bloques de una tarjeta NB
- 3.6 Arquitectura a bloques de una tarjeta SB

## Capítulo 5

- 5.1 La figura muestra un bloque de arreglo lógico el cual contiene 10 LEs.
- 5.2 Diagrama a bloques de un dispositivo MAXII
- 5.3 Bloques internos de la familia MAXII correspondiente al dispositivo EPM570
- 5.4 Red de distribución del reloj global usado en los dispositivos MAXII
- 5.5 Estructura interna de un LAB de un dispositivo MAXII
- 5.6 Generación de señales de control dentro del LAB
- 5.7 Estructura interna de un LE usado en un dispositivo MAXII
- 5.8 Estructura de un IOE de la familia MAXII
- 5.9 Configuración de los bloques de IOs por filas y columnas
- 5.10 Diagrama a bloques de un dispositivo CycloneII
- 5.11 Diagrama a bloques de la arquitectura de un CPLD de la familia CR-II de Xilinx
- 5.12 Arquitectura interna de una macrocelda de un dispositivo CR-II
- 5.13 Bloque IO de un dispositivo CR-II
- 5.14 Arquitectura interna de los dispositivos Spartan3

## **Capítulo 6**

- 6.1** Diagrama a bloques de la plataforma de validación para el chipset ICH10
- 6.2** Boques contenidos dentro del dispositivo HOOKSPLD
- 6.3** Diagrama a bloques correspondiente al P80 y FSC
- 6.4** Diagrama a bloques correspondiente al modulo SERIRQ
- 6.5** Diagrama a bloques correspondiente al modulo MONITOR
- 6.6** Módulos contenidos dentro del RESET PLD



# Introducción

---

*Se presentan las inquietudes, expectativas y motivaciones que dieron origen al desarrollo del proyecto, así como la descripción de cada una de las partes que conforman el presente documento.*

---

## **1. Introducción**

Cuando se inicia una disciplina, en este caso una maestría, se tienen muchas expectativas, ilusiones, miedos, todo un panorama nebuloso que paso a paso se va despejando conforme se avanza en los estudios. De forma simultánea cada individuo va creando su propio camino y comienza a adquirir los conocimientos necesarios, siempre limitados, para poder emitir un juicio sobre el área en la que le gustaría desempeñarse profesionalmente.

Un proceso de madurez es necesario para plasmar física y literalmente lo adquirido durante los dos años de maestría, es por eso que se realiza este proyecto de tesis. La filosofía de este trabajo es “Poner en práctica el conocimiento adquirido y obtener un nuevo aprendizaje sobre bases bien fundamentadas, usando como pretexto la realización de una innovación tecnológica”.

El presente documento trata de ubicar al lector dentro de la necesidad que dio origen al desarrollo de este trabajo. Cada una de las secciones es una muestra gráfica del proceso realizado. La información presentada esta organizada en 8 capítulos.

En este primer capítulo se incluye una breve introducción donde se presenta una descripción de cada una de las partes que conforman el presente documento.

En el capítulo 2 se presenta una breve descripción de la arquitectura de Intel®, donde se pretende introducir al lector en el contexto en el que se desarrolla este trabajo. En el capítulo 3 se da una explicación de las características y modos de uso de una tarjeta de validación para chipsets y procesadores.

En el capítulo 4 se presenta el planteamiento del problema que se quiere resolver, la visión general del sistema, arquitectura a bloques, así como las ventajas y desventajas de su uso

En el capítulo 5 se hace un estudio de factibilidad para determinar si el proyecto tiene los elementos necesarios para realizarse. Se presenta una evaluación de diferentes familias de dispositivos programables así como la asignación de recursos económicos y propuesta a la gerencia.

La solución implementada, diagrama a bloques de la arquitectura de la tarjeta de validación, características del diseño y las especificaciones técnicas son presentadas en el capítulo 6.

En el capítulo 7 se reportan los datos y resultados obtenidos, y se realiza una comparación con respecto a los objetivos inicialmente planteados.

En el capítulo 8 se presentan las conclusiones recabadas durante la realización de este proyecto así como el enfoque y los cambios que se fueron dando al diseño en cada una de las etapas realizadas y el trabajo que se puede seguir desarrollando a futuro.

# 2

## Breve descripción de la arquitectura Intel ®

---

*En este capítulo se presentan los principios generales de la arquitectura de Intel® (Procesador, Chipsets, dispositivos periféricos, reguladores y buses principales).*

---

## 2. Esquema generalizado de una PC

Una tarjeta de computadora esta compuesta por diferentes componentes y buses. Los principales componentes que forman una computadora son el Procesador, el Chipset, el controlador de entradas/salidas (SIO), los controladores de audio, controladores de red, reguladores de voltaje, los generadores y buffers de reloj. Los buses principales que conforman una PC son el Front Side Bus (FSB) actualmente remplazado por el bus QuickPath Interconnect (QPI), bus de memoria, bus de Interconexión de Componentes Periféricos (PCI y PCI Express), DMI, LPC, SPI, SATA, SAS, USB y SMBus. La figura 2.1 muestra un diagrama general de la arquitectura de una computadora.

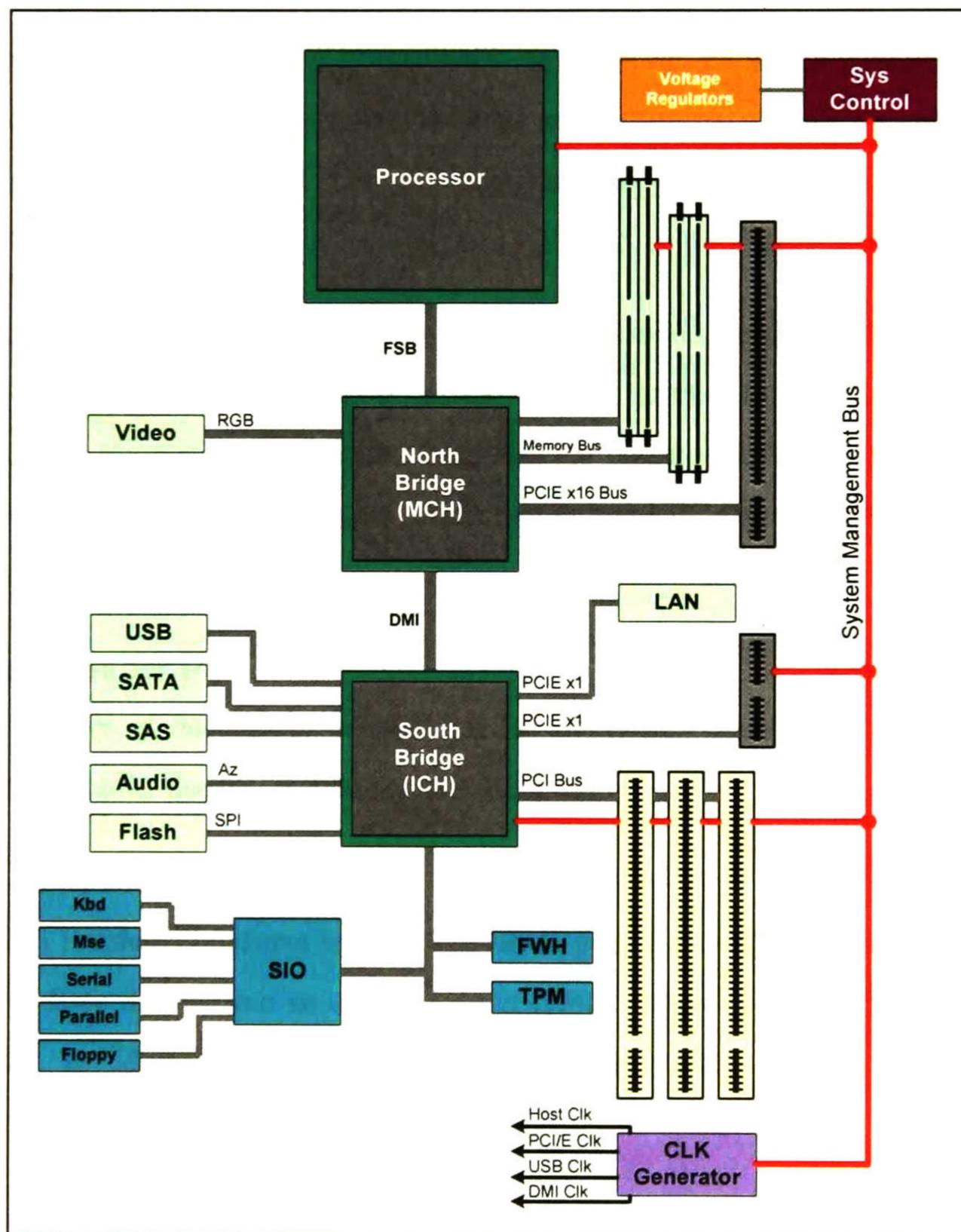


Figura 2.1 Diagrama a bloques de la arquitectura de una computadora (Desktop)

## **2.1. Procesador**

El procesador es el encargado de realizar los cálculos de operaciones aritméticas (incremento/decremento, recursividad, etc) y tomar decisiones (condiciones de falso o verdadero, donde almacenar los datos, corregir rutinas, etc). El desempeño del procesador esta muy ligado a su velocidad de reloj y a la velocidad del Front Side Bus (FSB) que se conecta hacia el controlador de memoria/gráficos (North Bridge o también conocido como Memory Controller Hub MCH)

## **2.2. Controlador de memoria y gráficos (MCH)**

El MCH es el encargado de controlar todas las transacciones que se realizan en las memorias (DIMMs) y también es usado como controlador de gráficos. El MCH o también comúnmente conocido como Puente del Norte (North Bridge) se comunica con el Puente del Sur (South Bridge) a través de la interfase DMI (Direct Media Interface).

## **2.3. Controlador de I/Os (ICH)**

El I/O Controller Hub (ICH) es el encargado de controlar todos los dispositivos de entrada/salida (diferentes del video), los cuales incluyen discos duros, tarjetas I/O PCI y PCIE, dispositivos USB, SAS, SATA y memorias SPI (usadas para leer los códigos de BIOS). Otra de las funciones del ICH es controlar los componentes que residen en el bus LPC (memorias FWH, controlador de I/O y TPM).

## **2.4. Controlador Súper I/O**

El controlador SIO es el encargado de manejar las interfaces comunes (Legacy I/O Interfaces) como son: teclado, ratón, floppy, puerto serial y paralelo.

## **2.5. Interfaz TPM**

La interfaz TPM (Trusted Platform Module) es un dispositivo de seguridad que se incluye en las tarjetas para PC. Este dispositivo se encarga de encriptar datos que son transferidos a través de Internet, provee la identificación de una PC en ambientes de red y cumple con las especificaciones del departamento seguridad de US.

## **2.6. Controlador del sistema**

El controlador del sistema (System Control) es el encargado de monitorear los voltajes y temperaturas del sistema, los cuales incluyen: voltajes y temperaturas del procesador, memorias y

las tarjetas de gráficos. Además se encarga de controlar diferentes funciones de Power Management, secuencias de reset y control de ventiladores del sistema (FSC).

### **2.7. Controlador de audio**

El controlador de audio (High Definition Audio Controller) provee las diferentes funciones de audio como son: Surround, micrófono, CD players y entrada de diferentes fuentes (MIDI, etc.).

### **2.8. Controlador de red (LAN)**

El controlador de Red (LAN) es usado para aplicaciones de red con diferentes velocidades 10/100/1000 Mbps.

### **2.9. Bus del procesador (FSB)**

El Front Side Bus es el bus del procesador que permite transferir información con el North Bridge, las velocidades de transmisión pueden llegar a los 6.4 GigaBytes/s. Actualmente el bus QuickPath Interconnect puede trabajar con una frecuencia de reloj de 6.4GT/s, y la primer generación (procesador Intel Core i7) puede transferir hasta 12.8 GB/s ( $6.4\text{GHz} \times 16 \text{ bits} / 8$ ) en transmisión y en recepción (los buses Tx y Rx se encuentran separados), en cambio el FSC comparte el mismo bus para Tx y Rx.

### **2.10. Bus de memoria**

El bus de memoria es usado por el NB para leer y escribir en los DIMMs. Existen diferentes configuraciones para los buses de memoria y pueden ir soportar desde un DIMM hasta 4 DIMMs por canal. La velocidad de transferencia depende del tipo de tecnología que se use (DDR2, DDR3, etc.).

### **2.11. Bus PCI**

Este es un bus de entrada/salida de propósito general para conectar diferentes periféricos y tarjetas como controladores de LAN, Modems, tarjetas de audio, etc. Su velocidad de transferencia es de 132 MBytes/s y el encargado de controlar este bus es el South Bridge.

### **2.12. Bus PCIE**

El bus de PCI Express esta diseñado para ser una interfase muy versátil y adaptable a diferentes aplicaciones. Es un bus serial punto a punto (full-duplex) y sus conexiones van del Chipset hacia un slot. Cada slot puede tener uno, dos, cuatro, ocho, dieciséis o treinta y dos enlaces de datos, y

cada enlace se escribe con una x de prefijo, por ejemplo x1 para un enlace simple y x32 para una tarjeta con 32 canales. Puede llegar a transferir hasta 1GB/s (250MB/s x 4) para PCIE G3 por enlace.

### **2.13. Bus LPC**

El bus Low Pin Count es usado para transferir datos entre el SB, SIO, FWH y TPM. Es un bus de 4 líneas y su velocidad de transmisión es de 16MB/s.

### **2.14. System Management BUS**

Este es el bus por el cual se accedan a los dispositivos para ser monitoreados o modificar sus configuraciones de registros. Algunos de los componentes que se incluyen en este bus son los generadores de reloj, los medidores de temperatura, el ICH, etc. También es usado como un camino alternativo para configurar dispositivos durante el proceso de arranque de la tarjeta (boot-up process), estos dispositivos son DIMMs y tarjetas PCI y PCIE.

### **2.15. BUS SPI**

El bus Serial Peripheral Interface es usado para transferir datos entre el SB y la memoria Flash ROM. Esta memoria es también usada para leer los códigos de BIOS al encenderse la tarjeta y su acceso de lectura es mucho más rápido (1GB/s) comparada con la memoria FWH que está en el bus LPC

### **2.16. Interfase de BUS DMI**

La interfase DMI provee un canal de transmisión entre el NB y el SB. Es un enlace dedicado similar a PCIE y su velocidad de transmisión llega a 1GB/s.

Con esta breve introducción sobre la arquitectura de Intel se puede proceder a entender la siguiente sección de este documento, donde se describen las principales características, funcionalidades y arquitectura general de las plataformas de validación Intel para procesadores y Chipsets.

# 3

## Características de las plataformas de validación Intel®

---

*En esta sección se presentará una visión general de las características y modos de uso de las tarjetas de validación Intel para Chipsets*

---

### **3. Características de las plataformas de validación Intel**

En el capítulo anterior se presentó de manera general la arquitectura de Intel para una PC, pero después de esto nos encontramos con las siguientes preguntas: ¿Cómo es que un procesador y un chipset se encuentran listos para ser introducidos al mercado? ¿Qué proceso se sigue para validar un procesador o un chipset? ¿Qué características y hardware es necesario para probar estos dispositivos?

En este capítulo se presentará de manera general las características y requerimientos de hardware necesarios para validar estos dispositivos y se enfatizará en la arquitectura de hardware para validación de chipsets.

#### **3.1. Introducción**

El concepto de plataformas [1] define un grupo de ingredientes (Hardware, Software, Tecnologías, Estándares y Servicios) usados para habilitar, validar o introducir diferentes aplicaciones en el mercado. Existen diferentes áreas en el segmento de plataformas de Intel y las más importantes son listadas enseguida:

- Plataformas para Desktop (enterprise & home)
- Plataformas Mobile
- Plataformas para Servidores
- Plataformas para telefonía celular y headsets
- Plataformas Healthcare

Todas estas plataformas contienen procesadores, chipsets y/o memorias fabricadas por Intel y pueden ser plataformas para validación o plataformas de referencia. Las plataformas de validación son usadas por los grupos internos de Intel y su principal objetivo es poder validar la mayor funcionalidad posible de un procesador, un chipset, un bus, un protocolo, etc., antes de ser introducidos al mercado. Las plataformas de referencia son productos y diseños que pueden ser usados por compañías externas a Intel, como son IBM, HP, Toshiba, DELL, etc., estas plataformas pueden ser además introducidas al mercado como un producto comercial de Intel. En este capítulo se presentará de manera general la arquitectura de una plataforma de validación Desktop.

### **3.2. Conceptos usados en la validación de CPUs y Chipsets**

Antes de presentar la arquitectura de una plataforma de validación Desktop se presentaran algunos conceptos relacionados con la validación.

Primeramente se define el concepto de validación como un proceso en el cual se asegura que el diseño de un procesador, un chipset, un bus, una tarjeta, etc. cumple con todas las especificaciones establecidas. Pero recordemos que en un diseño donde se requiere probar millones de transistores y estados siempre existirán errores, o en otras palabras “la perfección es imposible”. Algunas de las fuentes de error que se pueden presentar en el proceso de validación son causadas por errores humanos, ambigüedades o mal interpretación de las especificaciones, el tipo de circuitos usados para implementar funciones lógicas, los miles de combinaciones de estados que se pueden generar, los tiempos de transición, retrasos, variaciones de temperatura, etc.

Lo ideal en una etapa de validación es cubrir todas las combinaciones posibles y comparar los resultados con las especificaciones. Pero en la realidad se pueden encontrar millones de estados posibles, limitaciones en cuanto a la controlabilidad y observabilidad de pines o estados internos en un procesador o chipset, o hay casos en que se pueden encontrar especificaciones incompletas.

Un factor muy importante que se debe de tomar en cuenta en una plataforma de validación es el costo total del diseño. El costo hace la diferencia para que un producto pueda ser competitivo y viable. En Intel se manejan tres diferentes tipos de validación para cumplir con las especificaciones en sus productos [2]:

- Pre-silicon architecture validation
- Post-silicon system validation
- Post-silicon plataform compatibility validation

En estas tres fases se trata de descubrir y corregir cualquier error que no este dentro de las especificaciones antes de que el producto se introduzca en el mercado, y evitar cualquier falla que se pueda presentar con algún cliente externo.

### **3.2.1. Pre-silicon architecture validation**

En la etapa de validación de la arquitectura de pre-silicio se llevan a cabo simulaciones de circuitos y se verifica que la lógica interna de un procesador o chipset funcione correctamente. Este tipo de pruebas permiten a los arquitectos y diseñadores tener controlabilidad y observabilidad de cada transistor en los circuitos, pero la limitante es que estas simulaciones pueden llegar a ser muy tardadas debido a la gran cantidad de vectores de prueba que se puedan generar y además los simuladores no incluyen todas las variables externas que puedan influir en un diseño físico real. Sin embargo muchos de los vectores usados en esta etapa de validación pueden ser usados en la etapa de post-silicio y los resultados pueden ser correlacionados. Errores humanos, ambigüedad en las especificaciones, errores de tiempos y de simulación pueden llevar a generar erratas en esta etapa de validación.

### **3.2.2. Post-silicon system validation**

En esta etapa de validación se lleva a cabo un proceso de debug en un sistema real. Aquí es donde entran las plataformas de validación de sistema, ya que se requiere un hardware especial para poder validar todas las características físicas de un dispositivo. Algunas de las pruebas que se realizan en esta etapa son: pruebas al empaquetado y buffers del dispositivo, pruebas eléctricas de cada señal, pruebas térmicas y de ruido, pruebas para validar las interfaces y buses de cada dispositivo, etc. Se tiene controlabilidad y observabilidad de los procesadores y chipsets utilizando cadenas de JTAG y se pueden generar vectores para probar los bloques funcionales desarrollados. Para tener una mayor cobertura de las pruebas se corren pruebas funcionales utilizando vectores aleatorios de pruebas (Random Instruction Tests) y se comparan los resultados con las simulaciones. Se usan diferentes sistemas operativos basados en Linux/Unix y tarjetas de validación de IOs para correr diferentes tipos de pruebas y validar interfaces como son SATA, DMI, PCIE, USB, etc. En esta etapa de validación se descubren alrededor de 90% de erratas al silicio, combinaciones de lógica errónea, especificaciones que no se cumplen, así como interacciones con firmware (BIOS) y I/Os que solo pueden ser detectadas teniendo el dispositivo en un ambiente real.

### **3.2.3. Post-silicon platform compatibility validation**

En la etapa de validación de compatibilidad se llevan a cabo pruebas con diferentes sistemas operativos comerciales, diferente firmware y software comercial y una variedad de tarjetas y

I/Os. Se realizan pruebas que estresan al máximo los procesos y las aplicaciones que se estén corriendo y se usan diferentes tipos de compiladores para generar tráfico en todos los buses y sobrecargar los procesos de memoria, cache, multiprocesamiento y I/Os. Se utilizan diferentes estándares (OEM) para generar instrucciones y combinaciones aleatorias en los registros y buses de los procesadores y chipsets, con lo cual se asegura una adecuada funcionalidad para el usuario final. Todos los errores encontrados en esta etapa de validación deben de ser reproducidos en la etapa de validación de sistema para descubrir si se trata de un problema del silicio o un problema de hardware.

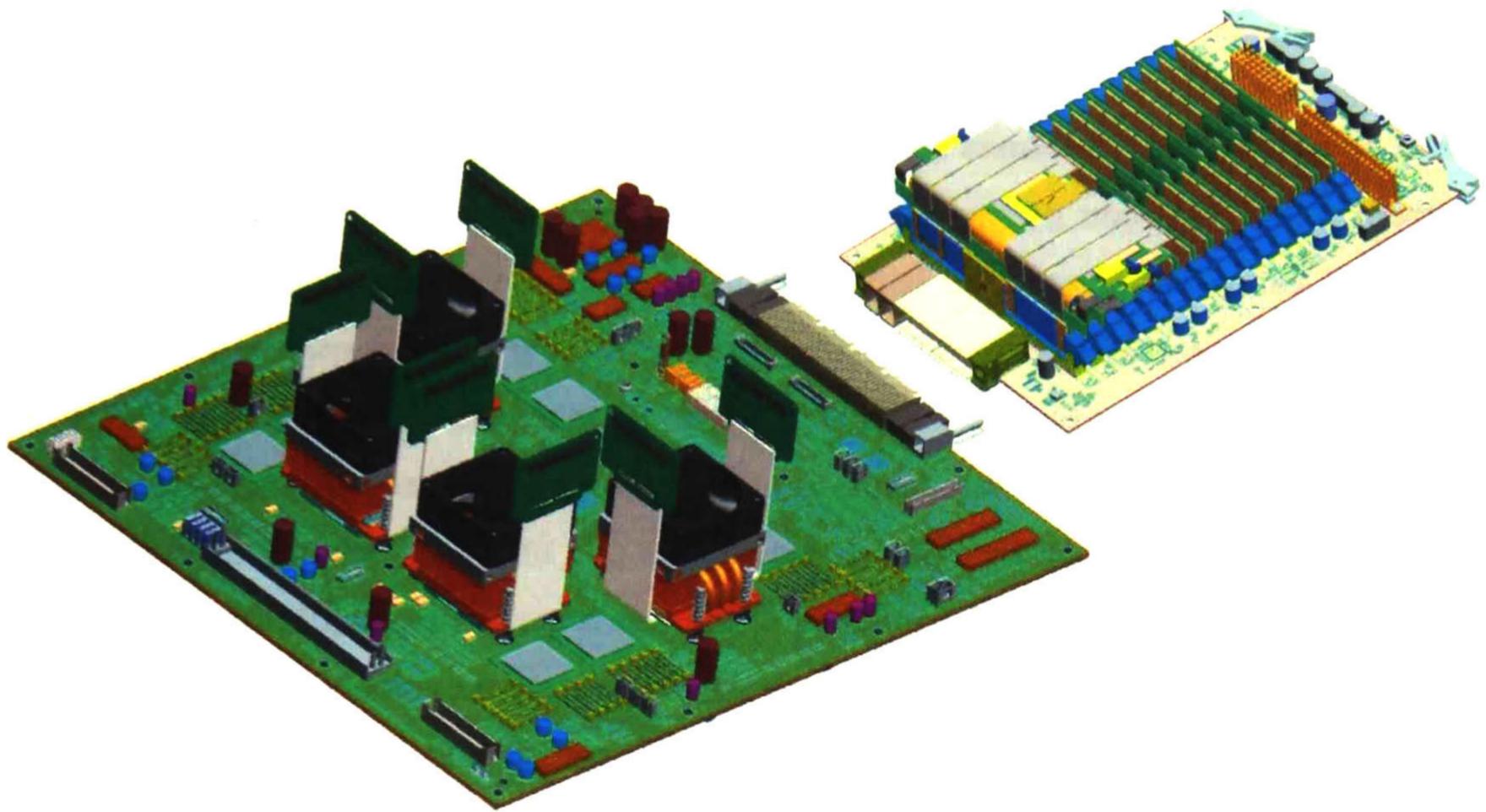
### **3.3. Capacidades de las plataformas de validación**

Existen cuatro características que distinguen a una plataforma de validación comparada con una de referencia, y estas son [3]:

- Observabilidad
- Controlabilidad
- Configurabilidad
- Medibilidad

#### **3.3.1. Observabilidad**

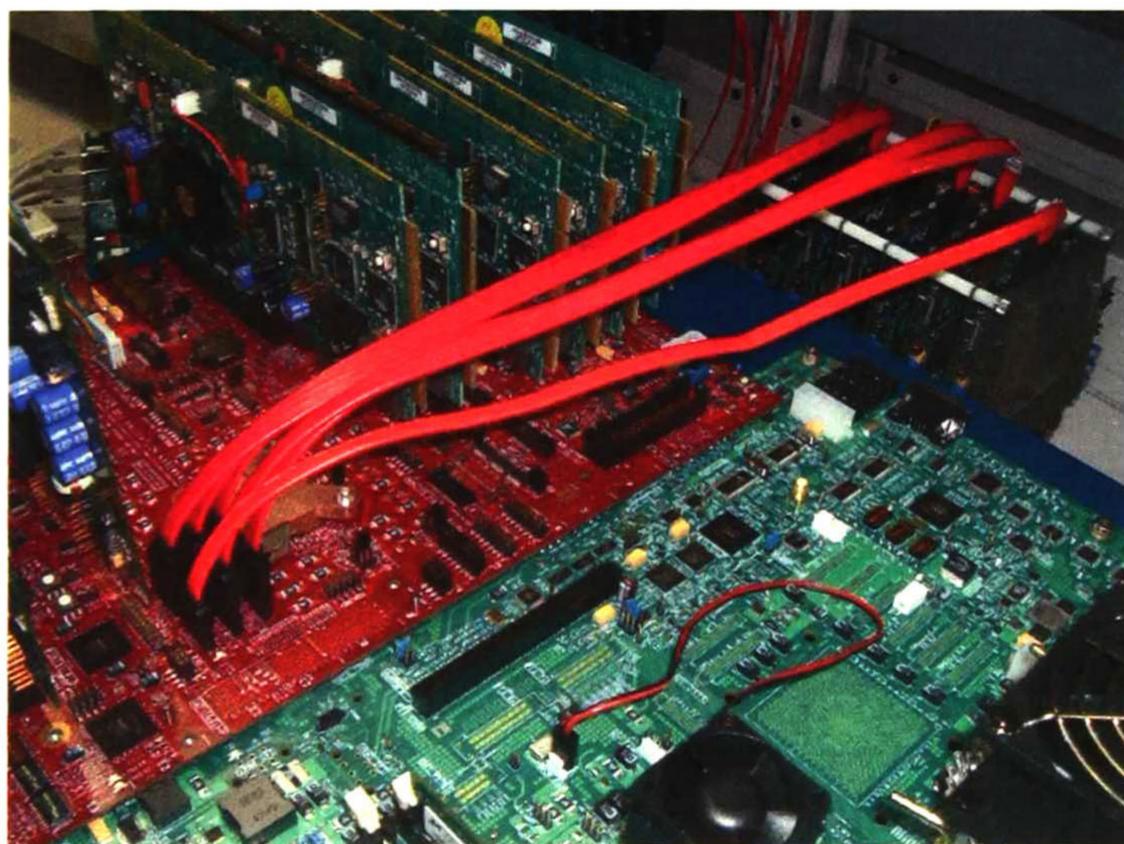
En una plataforma de validación se tiene la capacidad de observar todas o casi todas las señales y el monitoreo del tráfico en cada uno de los buses, por ejemplo se tienen huellas (footprints) para analizadores lógicos (midbus probes) para capturar las activad en los buses de PCIE, FBD, DDR, FSB, etc. Además que se diseñan y fabrican interposers para colocar diferentes CPUs o Chipsets en las tarjetas y existen diferentes tipos de tarjetas de pruebas que se conectan en headers o conectores especiales para monitorear frecuencias de operación, voltajes, corrientes y se cuenta con una gran variedad de LEDs para mostrar errores y/o eventos. La figura 3.1 muestra una tarjeta de validación para cuatro procesadores y una tarjeta de validación para FBD. En la figura se pueden observar los diferentes midbus en cada procesador, conectores mictors, headers y test points.



*Figura 3.1 Diagrama que muestra una tarjeta de validación para cuatro CPUs y una tarjeta para FBD*

### **3.3.2. Controlabilidad**

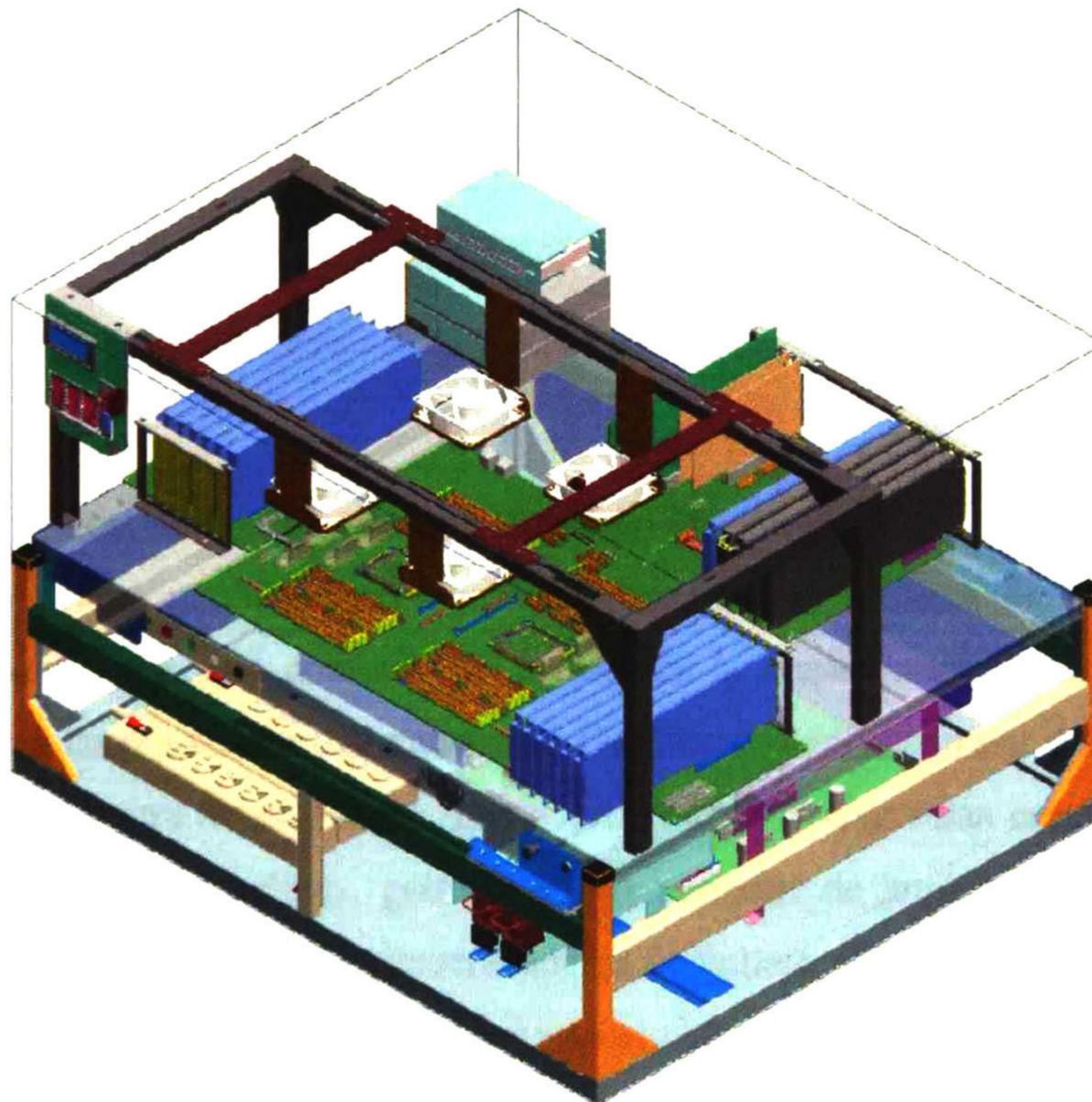
En las plataformas de validación es posible marginar frecuencias de cada uno de los relojes de los dispositivos, así como de los relojes que usa el CPU, también es posible variar los márgenes de voltaje, inyectar tráfico en los buses y controlar la temperatura.



*Figura 3.2 Diferentes tarjetas son usadas para inyectar tráfico en los buses de PCIE y SATA en una plataforma de validación.*

### 3.3.3. Configurabilidad

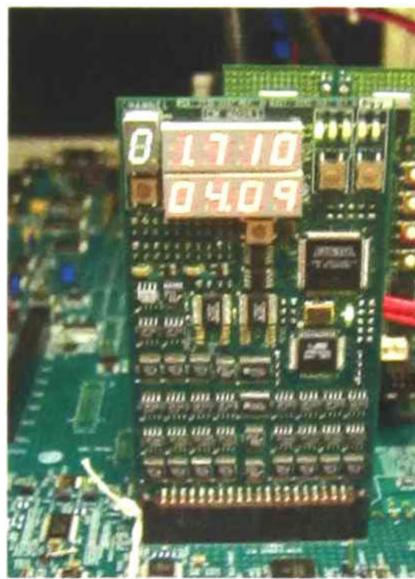
En una plataforma de validación es posible tener diferentes combinaciones (skews) de hardware, por ejemplo, una tarjeta puede ser configurada para trabajar con uno, dos, tres o hasta cuatro procesadores, otro ejemplo de configurabilidad es que se tiene la capacidad de configurar fácilmente los puertos de PCIE (x1, x2, x8, x16, etc) utilizando tarjetas especiales de bifurcación de las líneas. Todas estas variaciones de hardware también requieren variar las fuentes de poder de las plataformas ya que los requerimientos de potencia cambian de una configuración a otra, para esto cada plataforma cuenta con una fuente de alimentación diseñada de acuerdo a las capacidades y configuraciones requeridas por la plataforma, cabe mencionar que las fuentes de poder cuentan con módulos para ser expandibles y pueden generar mayor potencia en caso de ser requerida. La figura 3.3 muestra un chasis de una plataforma de validación que puede ser configurada para soportar 2 o 4 CPUs.



*Figura 3.3 Diagrama mecánico de chasis de una plataforma de validación que soporta 2 o 4 procesadores.*

### 3.3.4. Medibilidad

Las plataformas de validación cuentan con diferentes headers, sensores y registros internos en los dispositivos programables con el status de ciertos eventos que pueden ser accedidos fácilmente, esto tiene la finalidad de medir los diferentes voltajes, frecuencias, corrientes, actividades, estados de un bus y temperatura en diferentes puntos de la plataforma. En la figura 3.4 se muestra una tarjeta para medir el voltaje y corriente en el mismo instante en diferentes planos usados por el procesador o chipset, esta tarjeta de medición esta conectada a un header en la plataforma de validación. En el instante capturado en la figura 3.4 la medición muestra en el canal cero un voltaje de 1.710V @ 4.09A.

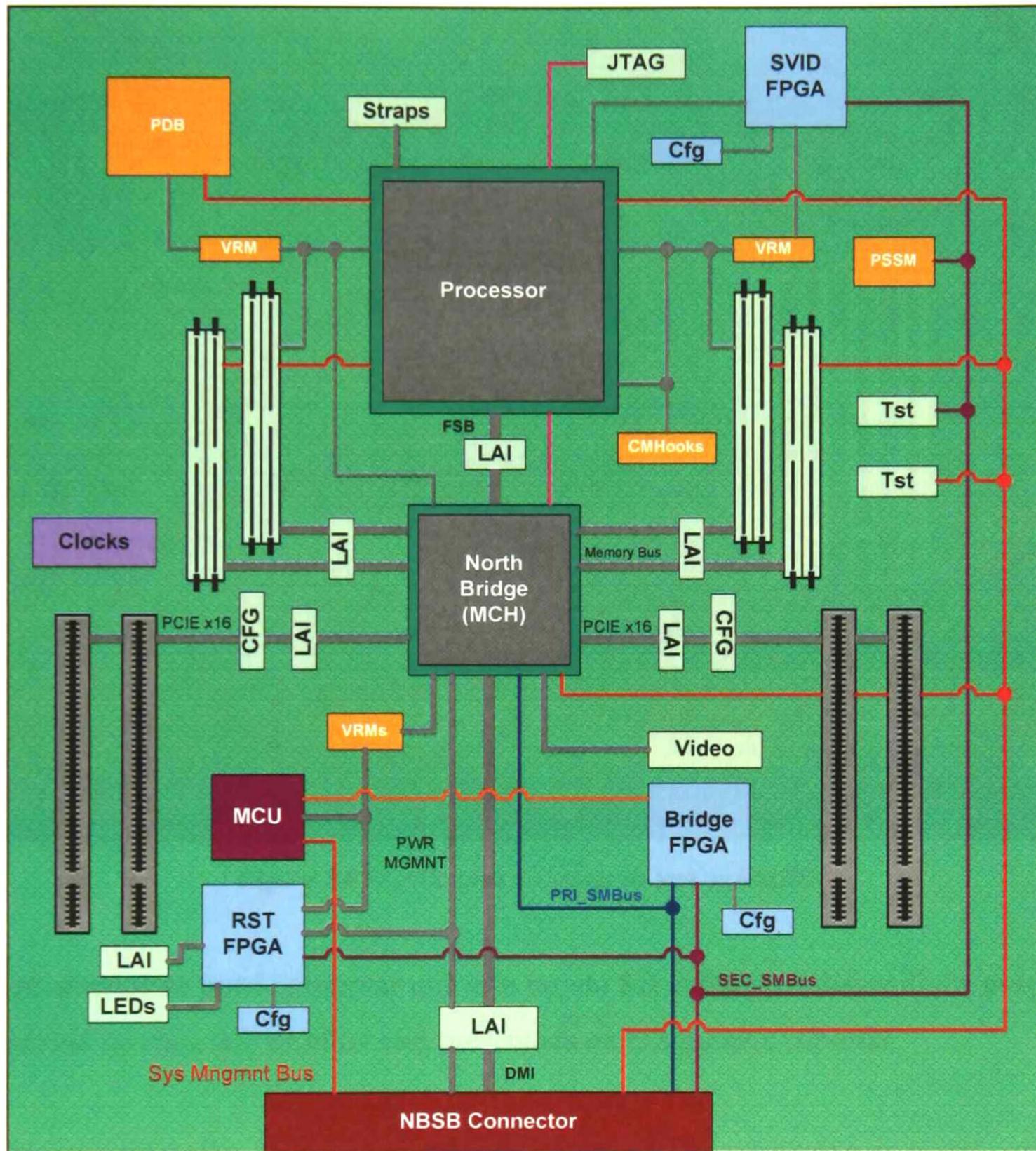


*Figura 3.4 Tarjeta de medición de voltaje y corriente.*

### 3.4. Arquitectura general de una tarjeta de validación Desktop

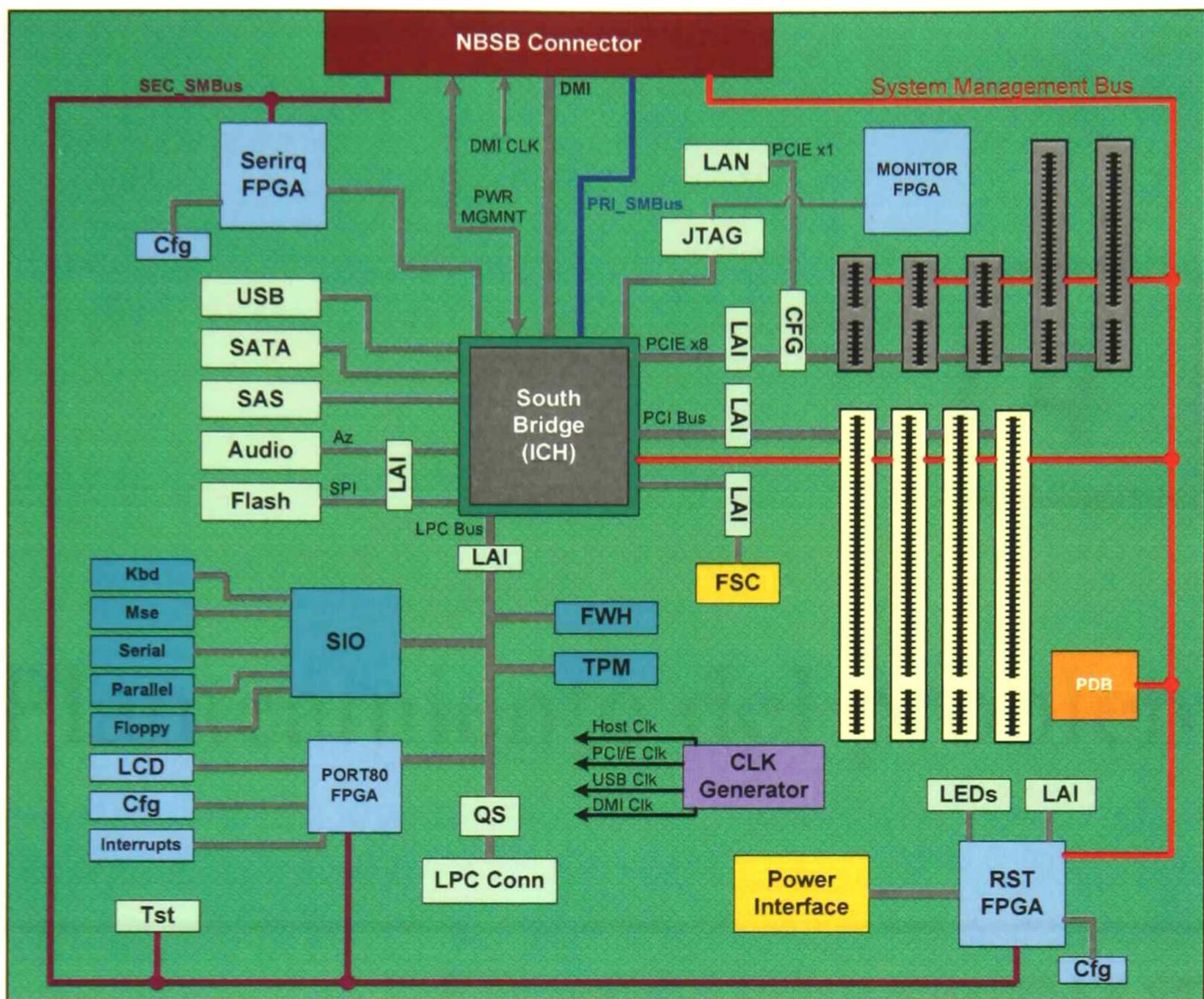
En los apartados anteriores se describieron las características principales de lo que es una plataforma de validación. En este apartado se presentara la arquitectura de una plataforma Desktop. Generalmente las plataformas de validación Desktop se conforman por dos tarjetas, una llamada North Bridge y la otra South Bridge. Las tarjetas North Bridge están compuestas por el procesador, el MCH, memorias, relojes, gráficos y una variedad de hooks (microcontrolador, dispositivos programables, puerto JTAG, secuenciadores de voltaje, etc.) que sirven para validar diferentes interfaces. Las tarjetas South Bidge incluyen el chipset ICH y todo lo referente a I/Os excepto gráficos (USB, SATA, LAN, Audio, Súper IO, PCI, PCIE, etc.) también incluye diferentes hooks como son los dispositivos programables, huellas para diferentes analizadores lógicos, secuenciadores y relojes. En la figura 3.5 se presenta un diagrama a bloques de la

arquitectura de un North Bridge y en la figura 3.6 se presenta el diagrama para una tarjeta South Bridge.



*Figura 3.5 Arquitectura a bloques de una tarjeta NB*

En la figura anterior se pueden ver algunos bloques nuevos que no aparecen en una plataforma comercial o de referencia, los cuales son: la interfase con la fuente de poder (PDB), los reguladores de voltaje (VRM), el modulo CMHooks que se utiliza para medir voltajes y frecuencias, los secuenciadores y supervisores de voltajes (PSSM), los diferentes footprints para LA y los dispositivos programables (FPGAs y MCU).



**Figura 3.6** Arquitectura a bloques de una tarjeta SB

De la figura anterior se puede observar que esta tarjeta SB contiene 4 dispositivos programables, la funcionalidad de cada uno de ellos será presentada en el siguiente capítulo.

Con esta breve explicación de la arquitectura de las tarjetas de validación se podrá entender cual fue la motivación que llevo a realizar este trabajo. En el siguiente capítulo se presentara un estudio de factibilidad que ilustra el análisis seguido para reducir los costos en una plataforma de validación South Bridge.



# Planteamiento del problema

---

*El planteamiento de problema corresponde a la etapa más importante en el desarrollo de un prototipo, incluye la obtención de datos e investigaciones que soportan el proyecto y la forma en la que el problema será resuelto.*

---

#### **4. Planteamiento del problema**

Si se analiza la complejidad, número de capas de un PCB (14 a 20 capas) y todos los componentes que requiere una plataforma de validación se pueden llegar a determinar diferentes áreas de oportunidad para reducir el costo de este tipo de tarjetas; por ejemplo se puede hacer un estudio para determinar que tipo de material es el más conveniente para fabricar la tarjeta de circuitos impresos (PCB) de acuerdo a las especificaciones del sistema, simulaciones y evaluando diferentes procesos de fabricación. Otra área de oportunidad para reducir costos es haciendo optimizaciones al BOM de cada tarjeta y buscando proveedores que ofrezcan precios competitivos de componentes a bajo consumo. Otra de las áreas de oportunidad que pueden ser exploradas y es la que se estudia en este trabajo, es la optimización y automatización de los códigos de los dispositivos programables y selección de los componentes óptimos para cada caso.

##### **4.1. Breve reseña del grupo de diseño de plataformas en GDC (VHG)**

Actualmente el grupo de diseño de plataformas de validación de Guadalajara cuenta con 24 Ingenieros. Dentro del grupo existen diferentes tipos de puestos y responsabilidades, las cuales se conforman de la siguiente forma:

- Manager
- Líder de proyecto (arquitecto)
- Ingeniero de diseño de Hardware
- Ingeniero de Integridad de Señal y simulaciones
- Ingeniero de Layout
- Ingeniero de compras y materiales

El primer proyecto realizado por el grupo de VHG [4] fue una plataforma South Bridge, y fue realizado por 10 personas, cada uno con diferentes responsabilidades (1 líder de proyecto, 5 Ingenieros de hardware, 2 Ingenieros de Simulaciones, 1 de Layout, 1 de materiales), además que este primer proyecto sirvió como rampa inicial para el grupo, ya que se tomaron diferentes entrenamientos con grupos similares de USA para los diferentes puestos. Este primer proyecto fue útil para comprender la metodología de trabajo, arquitectura y diseño que se realiza en Intel para este tipo de proyectos, además que se trabajo directamente con los grupos de validación y se entendieron los modelos de uso de las pruebas que se desarrollan y la forma en que se validan diferentes interfaces. El proyecto se entrego a tiempo y fue exitoso.

El segundo proyecto que se asignó al grupo de VHG fue una plataforma para Desktop que incluía el North Bridge y el South Bridge, pero en este caso solo 3 personas estuvieron trabajando en el South Bridge y otras 4 en el North Bridge. En este segundo proyecto se puede ver que el número de Ingenieros se redujo y el trabajo por realizar se incrementó considerablemente. Fue en este segundo proyecto donde se realizaron diferentes optimizaciones para reducir los costos totales, ya que se tenía la experiencia de un primer proyecto y se conocían los modos de uso y las diferentes pruebas que se requerían cubrir. Además, el reducir costos es uno de los principales objetivos que se tienen para las plataformas de validación (que sean eficientes en cuanto a costo y desempeño).

#### **4.2. Oportunidad de mejora detectada en una plataforma Desktop (SB)**

Durante el primer proyecto realizado por el grupo de VHG en Intel se detectó una oportunidad para optimizar costos en los dispositivos programables y código de los mismos. Los dispositivos programables usados en la tarjeta de validación para el chipset ICH9 fueron 4 PLDs de la familia 7000B de Altera. Los cuales fueron usados de la siguiente forma:

- RST: este dispositivo fue usado para generar todas las secuencias de arranque y reset de la plataforma, así como proveer funcionalidades de power Management y generar los diferentes estados (sleep states) requeridos por el chipset. Este dispositivo además se sincroniza con el RSTPLD que se encuentra en la tarjeta North Bridge, maneja funcionalidades de LAI y controla todos los LEDs de la tarjeta SB.
- PORT80: este dispositivo es el encargado de monitorear el tráfico en el bus LPC y cachar los códigos de BIOS. Estos códigos son desplegados en un LCD controlado por este dispositivo; así mismo, el control de ventiladores (PWM y TACH) es manejado por este dispositivo. Además se tienen registros internos de lectura en este dispositivo donde se puede conocer el status de cada señal que se conecta a este PLD.
- SERIRQ: Este PLD se encarga de configurar las señales (GPIOs y straps) del chipset, además se encarga de generar interrupciones seriales de PCI. Diferentes dipswitches son conectados para configurar el dispositivo, además se cuenta con registros internos de lectura para conocer el estatus de las señales.
- MONITOR: Este dispositivo se usó para monitorear señales que son conectadas en headers de prueba o de configuración, además se utilizaba como interfase de DFX test, que es un protocolo serial para acceder a registros internos del chipset.

Los registros internos de todos estos dispositivos son accedidos por SMBus.

Durante el desarrollo de la plataforma para el chipset ICH9 se recibieron varios requerimientos por parte de los grupos de validación para automatizar sus pruebas; esto es, agregar la lógica necesaria para que los registros internos de los dispositivos fueran de lectura y de escritura. Otros requerimientos recibidos fueron tener la opción de recibir comandos a través de SMBus para generar retrasos y/o eventos de power Management en las secuencias de inicialización con la finalidad de validar las nuevas capacidades del chipset ICH9.

Los dispositivos programables usados fueron como se menciona anteriormente el MAX7000B de Altera, el dispositivo usado (EPM7256BFC256) contaba con 256 macroceldas. En la siguiente tabla se muestra cada dispositivo con su respectivo porcentaje de utilización y pines consumidos.

*Tabla 4.I Porcentaje de utilización y pines consumidos para la tarjeta de validación ICH9*

Analysis	ALTERA MAX7000B	Total Macrocells	Total Pins
<b>PORT80</b>	EPM7256BFC256-5	230/256 (90%)	66/164 (40%)
<b>MONITOR</b>	EPM7256BFC256-5	122/256 (47%)	143/164 (87%)
<b>SERIRQ</b>	EPM7256BFC256-5	205/256 (80%)	130/164 (79%)
<b>RESET</b>	EPM7256BFC256-5	249/256 (97%)	160/164 (97%)

El costo de cada dispositivo fue de 14 dólares, dando un total de 56 dólares por tarjeta. Durante el proyecto se requirió implementar mas funcionalidades en el RESET agregando mas lógica y pines, lo cual con el dispositivo utilizado no era posible lograrlo, por lo que se investigo si Altera tenia un dispositivo con mayores capacidades de la misma familia y con el mismo footprint para poder utilizarlo en la tarjeta. Sorprendentemente existe un dispositivo con el mismo footprint, su arquitectura cuenta con 512 macroceldas (en vez de las 256 usadas) y con un incremento en el numero de pines de 212. Todo esto se paga a un precio muy alto, ya que el costo de este dispositivo (EPM7512BFC256-5) fue de 90 dólares.

*Tabla 4.II Porcentaje de utilización y pines consumidos con el nuevo dispositivo (512) para el RESET*

Analysis	ALTERA MAX7000B	Total Macrocells	Total Pins
<b>RESET (512)</b>	EPM7512BFC256-5	293/512 (57%)	182/212 (85%)

Con este nuevo dispositivo se lograron implementar varios de los requerimientos que los grupos de validación propusieron, pero el precio total de los dispositivos programables se fue a 132 dólares. Por lo cual en el nuevo proyecto (tarjeta de validación para el chipset ICH10), se decidió hacer un estudio mas detallado para seleccionar dispositivos óptimos en cada caso. Otro de los estudios que también se realizaron para la plataforma ICH10 fue optimizar las rutinas y códigos de los dispositivos programables y migrar el código de VHDL a Verilog. Esto es lo que se presentara en los siguientes apartados.

#### **4.3. Ventajas en la optimización de los dispositivos programables**

Al realizar una optimización de los dispositivos programables en las tarjetas de validación se pueden encontrar las siguientes mejoras:

- Reducción del costo total de cada tarjeta fabricada
- Optimización de la lógica programable
- Capacidad para automatizar pruebas que realizan los grupos de validación
  - Ahorro de tiempo y costos para los grupos de validación
- Conocimiento mas profundo de las diferentes familias (arquitectura, aplicaciones y costos) de los dispositivos lógicos programables
- Este cambio puede dar visibilidad al grupo de VHG ya que se estarían haciendo mejoras (reducción de costos) en las plataformas diseñadas.

#### **4.4. Desventajas en la optimización de los dispositivos programables**

Algunas desventajas que puede traer hacer este estudio son las siguientes:

- El estudio requiere tiempo para revisar cada familia de dispositivos
- Se requiere cambiar el código de VHDL a Verilog, lo cual se traduce en tiempo, simulaciones y corregir posibles errores
- El cambiar de una familia de dispositivos a otra trae riesgos de compatibilidad y posibles retardos en la lógica programada en cada dispositivo.
- Existe el riesgo de que la plataforma no este a tiempo debido a la implementación y trabajo por realizar.

En este apartado se describió de manera general las causas que impulsaron la realización de este proyecto, en el siguiente capítulo se mostrara un estudio de factibilidad donde se revisaron

diferentes familias de dispositivos programables, se analizaron los costos, se hicieron diferentes compilaciones, simulaciones y optimizaciones de códigos en cada caso.



# Estudio de factibilidad

---

*El estudio de factibilidad, permite determinar si el proyecto que se ha planteado tiene los suficientes elementos para garantizar la vida posterior del proyecto, antes de que se tome la decisión de asignar más recursos económicos y humanos para proseguir a la siguiente etapa.*

---

## **5. Estudio de factibilidad**

Establecer una solución única a un problema determinado es, la mayoría de la veces riesgoso. Al comenzar un proyecto se deben establecer criterios que sirvan de punto de partida para la evaluación de posibles soluciones al problema. Para desarrollar este estudio de factibilidad se siguió la siguiente metodología:

- Se revisaron los diferentes tipos de familias de PLDs y FPGAs (de bajo costo) de Xilinx y Altera y se determinaron cuales cumplen los requerimientos de hardware (soporte de diferentes niveles de voltaje en IOs, voltajes de alimentación, retardos y pullups internos programables, etc)
- Se compilaron los códigos actuales de los dispositivos programables de la tarjeta utilizando diferentes dispositivos
- Se cotizaron los costos de cada dispositivo
- Se analizo que compañía provee mayor soporte con sus dispositivos
- Se realizo la optimización de códigos fuente y se agregaron los nuevos fixtures de acuerdo a los requerimientos recibidos para la plataforma
- Los códigos generados fueron simulados para corroborar que fueran correctos.
- En base al estudio de capacidades de cada dispositivo se verifico si el número de PLDs/FPGAs puede ser reducido con la finalidad de optimizar los recursos de hardware.
- En base a estos elementos se realizo una tabla con diferentes ponderaciones y se evaluaron los diferentes dispositivos seleccionados. Aquí se definió que familia fue la seleccionada en la plataforma de validación ICH10.
- Los resultados fueron presentados al líder de proyecto para ser revisados y posteriormente la solución se presento a la gerencia.

En los siguientes apartados se presentan cada uno de los puntos seguidos en la metodología para seleccionar los nuevos dispositivos programables usados en la tarjeta de validación ICH10.

### **5.1. Dispositivos programables**

Los dispositivos programables que se revisaron son fabricados por Altera [5] y Xilinx [6]. Estos dispositivos no requieren tener de grandes capacidades dentro del dispositivo como lo son las familias Stratix de Altera o Virtex de Xilinx, los cuales cuentan con bloques de memoria, procesadores embebidos, multiplicadores y sumadores para aplicaciones de DSPs, entradas para

DDR3, etc. Como se menciona en el capítulo anterior, los dispositivos programables que se usan en este tipo de tarjetas solo requieren de tener los suficientes IOs y que sean compatibles con los estándares utilizados (LVCMOS y GTL), también requieren una cantidad de celdas de mediana o alta capacidad para implementar las funciones lógicas, diferentes interrupciones y registros de lectura y escritura, además de que sean de bajo costo. Por lo tanto la familia de dispositivos que se estudiaron corresponde a los PLDs de alta capacidad y FPGAs de baja capacidad. En los siguientes puntos se presenta un estudio de los dispositivos que manejan las compañías Altera y Xilinx, donde se podrán ver sus principales diferencias en capacidades lógicas, IOs disponibles, estándares usados y comparaciones de precios entre cada uno. Los términos Elementos Lógicos (usado por Altera) y Macrocelda (usado por Xilinx) son puramente comerciales, ya que lo importante es conocer como esta compuesta su arquitectura interna para determinar sus capacidades, desempeño y aplicaciones.

## 5.2. Dispositivos de Altera

Como se menciona en el capítulo anterior, la familia de dispositivos usada en las plataformas de validación de generaciones anteriores usaban el componente MAX7000B de Altera, pero de acuerdo a los nuevos requerimientos de diseño se requería emigrar al dispositivo de mayor capacidad (512 LEs) pero el costo por tarjeta usando estos dispositivos llegaría a 208 dólares. Por lo tanto la familia MAX700 fue descartada para la realización de este estudio.

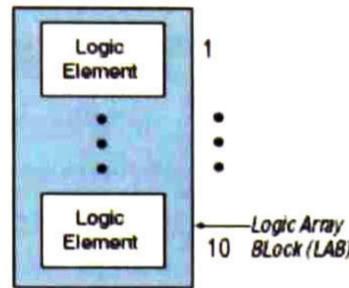
### 5.2.1. Dispositivos Altera MAXII

La familia de PLDs MAXII de acuerdo a las especificaciones de Altera es una familia de bajo costo, bajo consumo de potencia y soporta una gran variedad de estándares para IOs (3.3, 2.5, 1.8 y 1.5V), además de que cuenta con un bloque de memoria no volátil interna para almacenamiento. La siguiente tabla muestra un resumen de las capacidades de cada uno de los dispositivos MAXII.

*Tabla 5.1 Resumen de las características de los dispositivos MAXII*

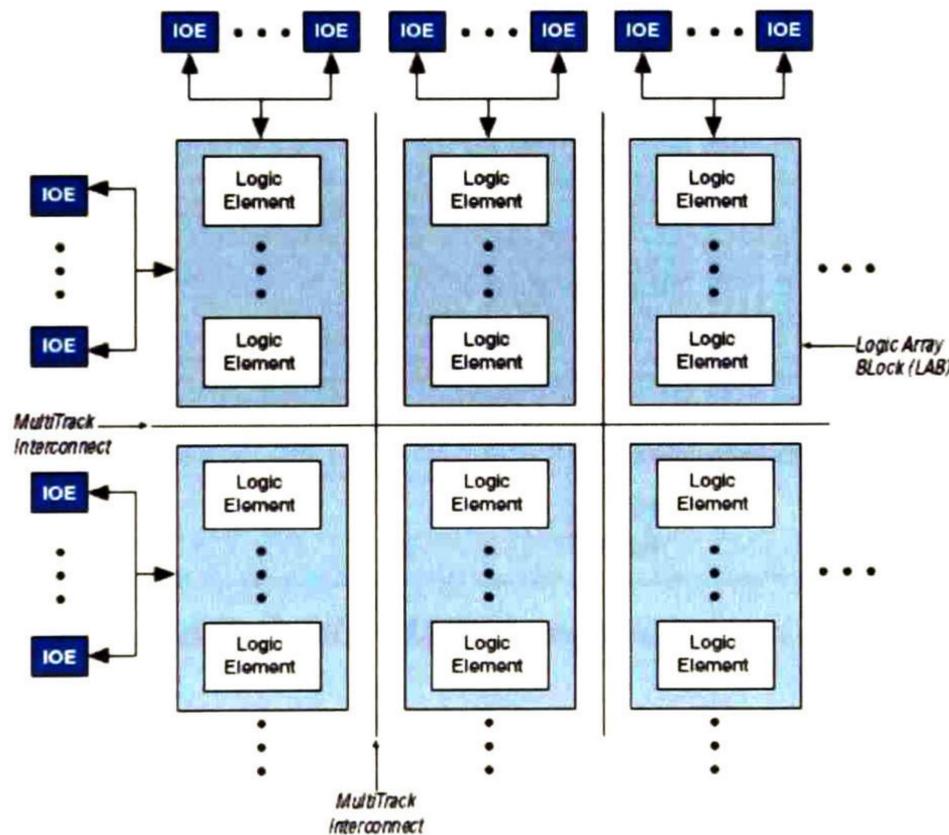
Feature	EPM240 EPM240G	EPM570 EPM570G	EPM1270 EPM1270G	EPM2210 EPM2210G	EPM240Z	EPM570Z
LEs	240	570	1,270	2,210	240	570
Typical Equivalent Macrocells	192	440	980	1,700	192	440
Equivalent Macrocell Range	128 to 240	240 to 570	570 to 1,270	1,270 to 2,210	128 to 240	240 to 570
UFM Size (bits)	8,192	8,192	8,192	8,192	8,192	8,192
Maximum User I/O pins	80	160	212	272	80	160

Esta familia de PLDs tiene en su arquitectura interna filas y columnas que son interconectadas para implementar las funciones lógicas que son determinadas dentro de los bloques de arreglos lógicos (LABs). Cada uno de los LABs contiene 10 elementos lógicos (LEs), donde cada LE es la unidad básica o elemental para implementar las funciones lógicas (ver figura 5.1).



**Figura 5.1** La figura muestra un bloque de arreglo lógico el cual contiene 10 LEs.

Los LABs son agrupados en filas y columnas dentro del dispositivo y pueden ser interconectados para implementar las funciones lógicas programadas. Los pines de entrada y salida IO son conectados a través de elementos de entrada/salida IOEs y se encuentran alrededor de cada LAB. En la figura 5.2 se puede ver el diagrama a bloques de la arquitectura interna de un dispositivo MAXII.



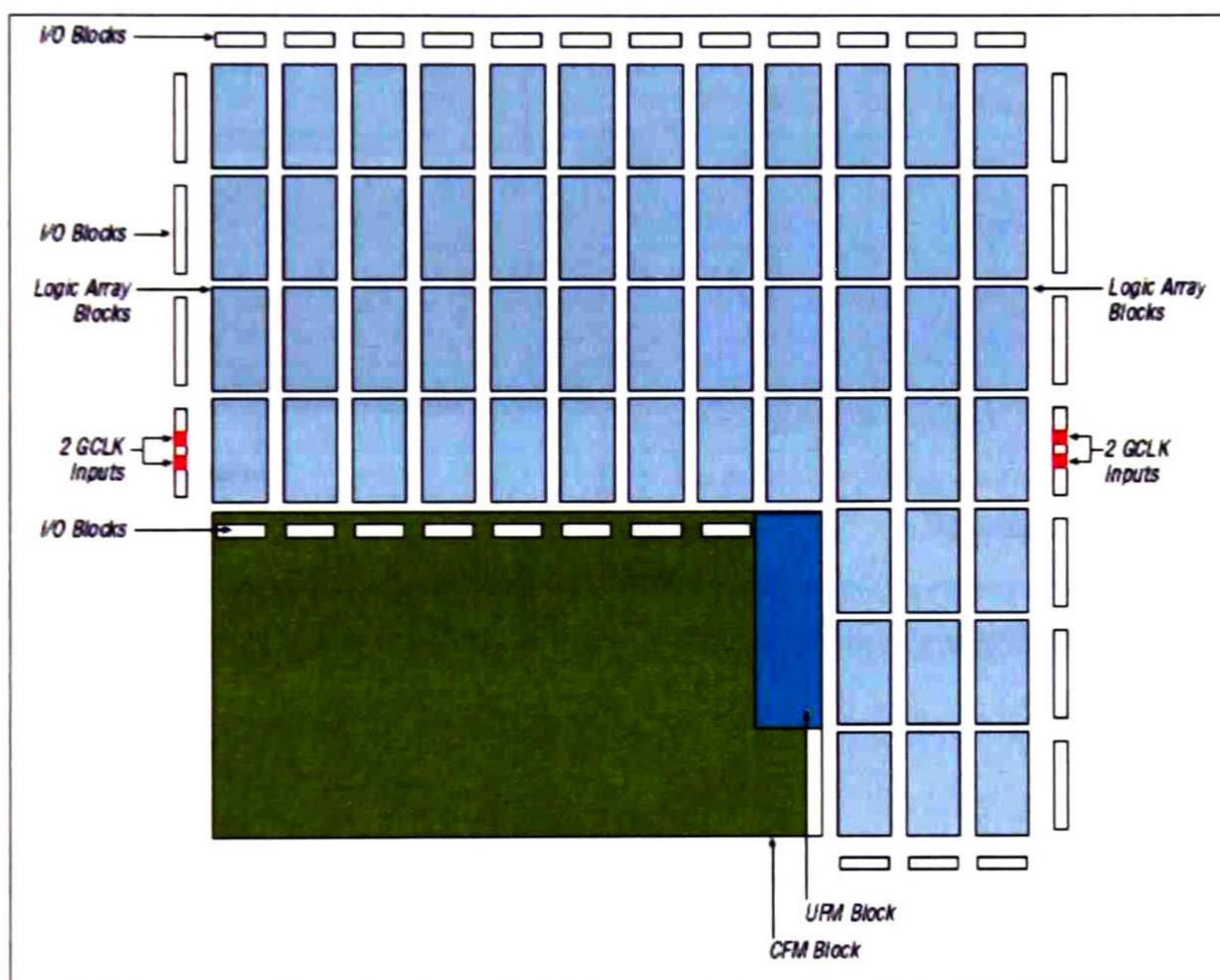
**Figura 5.2** Diagrama a bloques de un dispositivo MAXII

Esta familia de PLDs contiene un bloque de memoria flash interna dedicada (UFM) la cual provee almacenamiento no volátil que puede ser usado para diferentes aplicaciones. La tabla 5.II muestra el número de filas y columnas (LABs) de cada uno de los dispositivos MAXII. Cabe mencionar que existen algunos LABs extra en los dispositivos 570, 1270 y 2210. El acomodo de cada uno de los bloques internos de este dispositivo se puede ver en la figura 5.3 donde el bloque

de memoria CFM corresponde a la memoria de configuración del dispositivo y el UFM es el bloque de memoria flash para uso general.

**Tabla 5.II** Comparación de LABs entre cada dispositivo de la familia MAXII

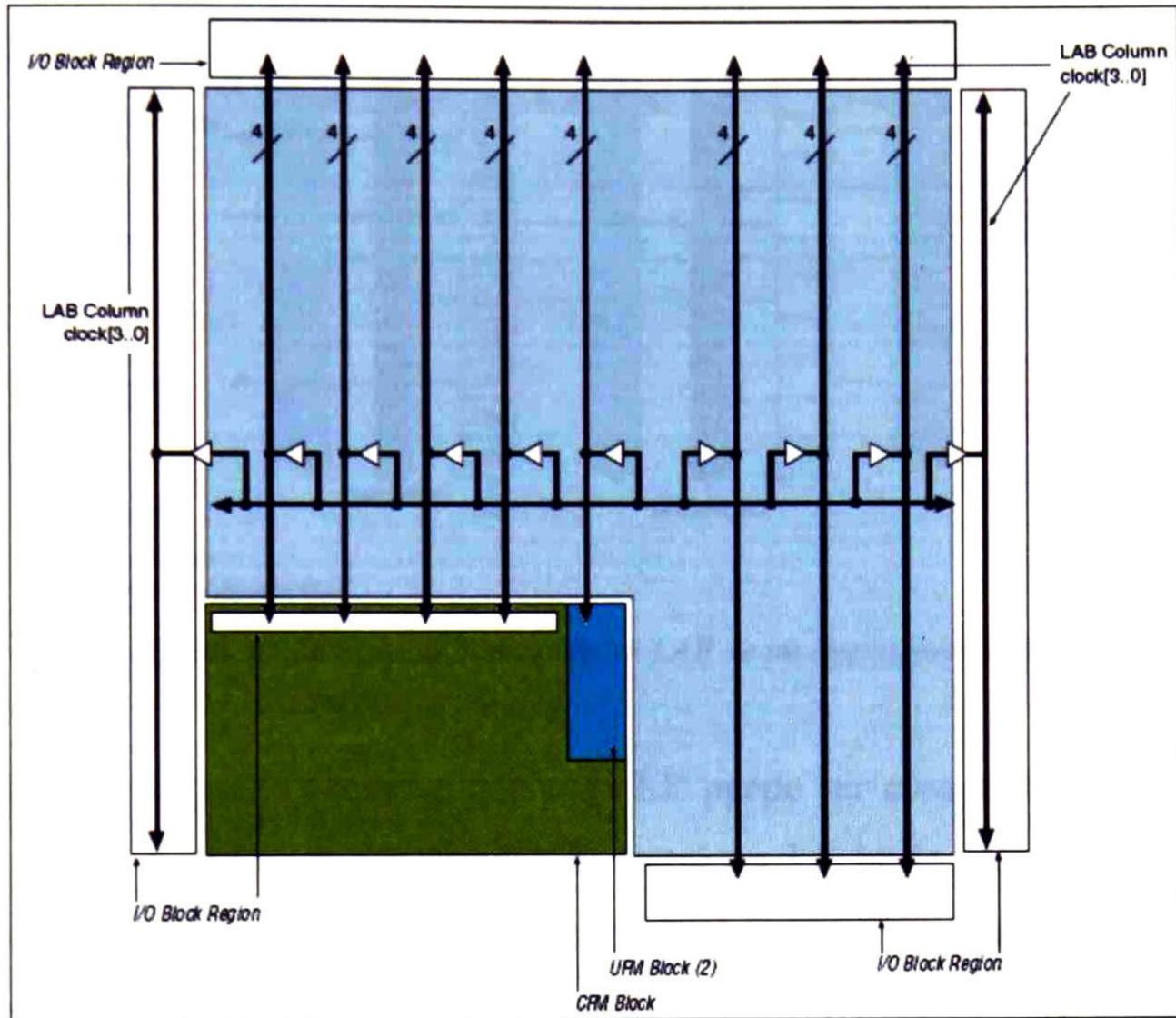
Devices	UFM Blocks	LAB Columns	Long LAB Rows	Total LABs
EPM240	1	6	4	24
EPM570	1	12	4	57
EPM1270	1	16	7	127
EPM2210	1	20	10	221



**Figura 5.3** Bloques internos de la familia MAXII correspondiente al dispositivo EPM570

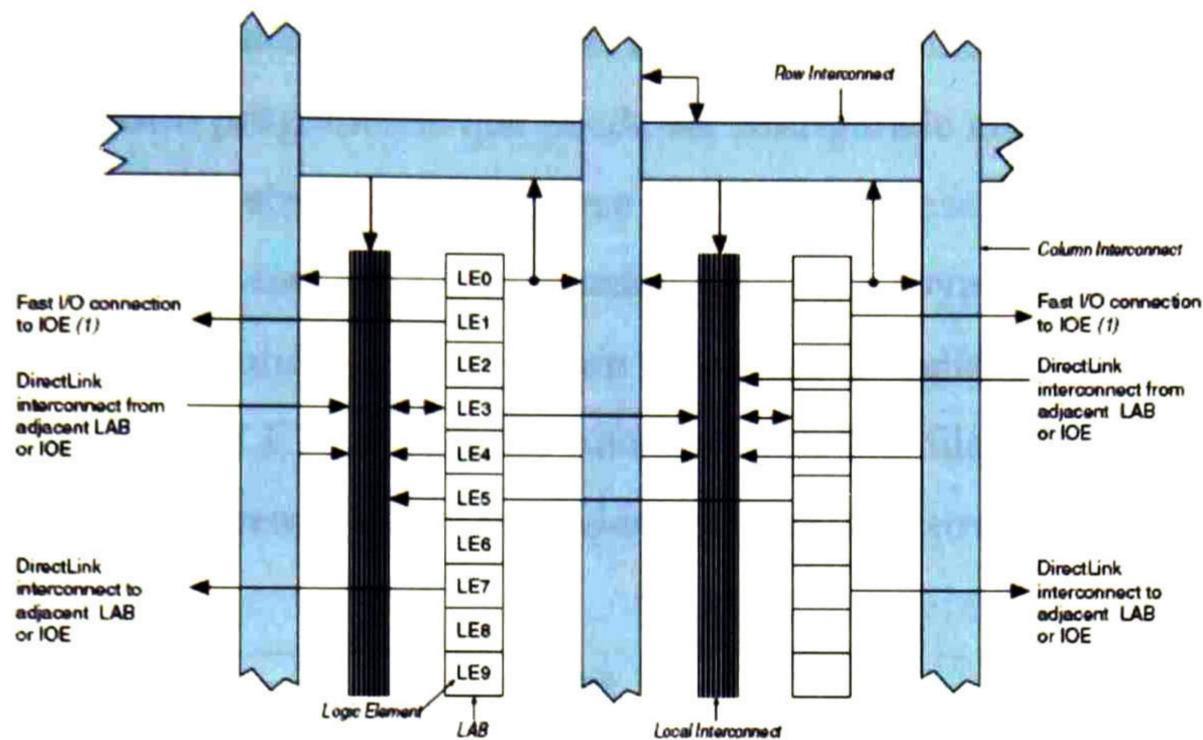
Dentro del dispositivo existe una red de reloj global que consiste en cuatro líneas de reloj que son llevadas a través de todo el dispositivo para proveer los recursos necesarios y la lógica secuencial en el dispositivo. Dos líneas de reloj se encuentran en la parte izquierda del dispositivo y las otras dos en la parte derecha. Cabe mencionar que estos pines pueden ser usados como IOs de propósito general si no son configuradas como relojes. La distribución global de relojes puede ser conectada a los LEs, LABs, IOEs y a la memoria UFM. Además la línea de reloj global es usada para generar señales de control, como clock enables, clear o load síncronos o asíncronos, output

enables o señales de control para algunos protocolos (TRDY o IRDY para PCI). La figura 5.4 muestra la red de distribución del reloj global usado en la familia MAXII.



*Figura 5.4 Red de distribución del reloj global usado en los dispositivos MAXII*

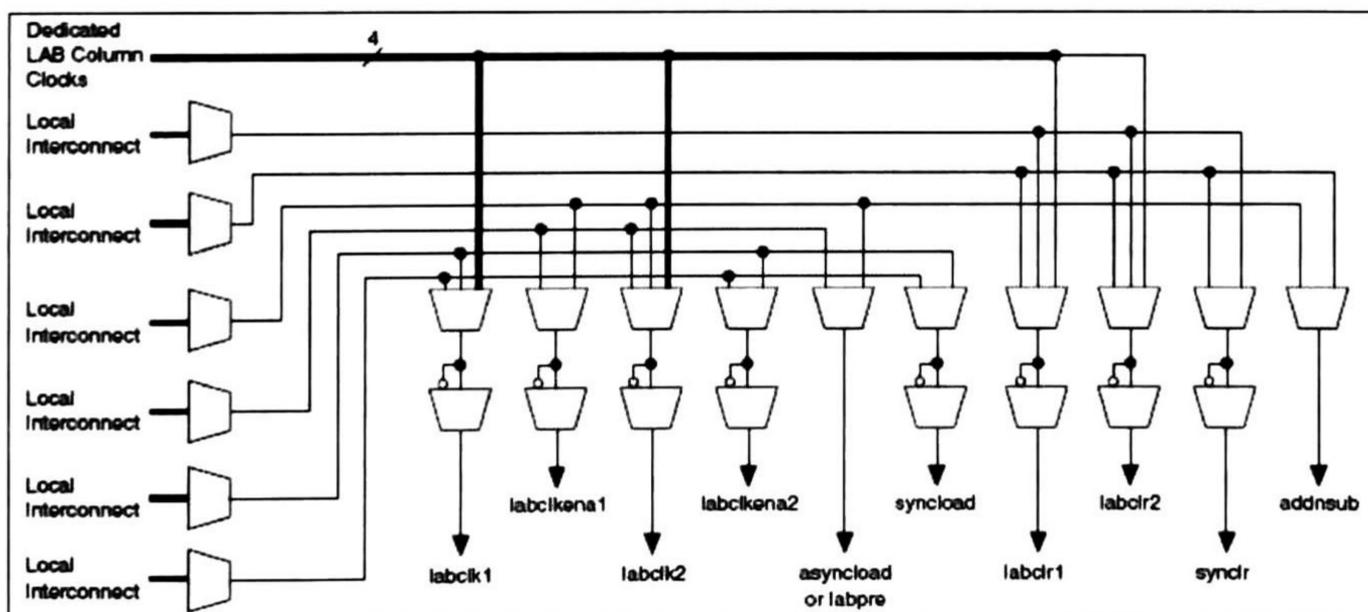
Como se mencionó anteriormente los LABs están formados por 10 LEs, además de señales de control, interconexiones locales entre LEs e interconexiones externas entre LABs. La figura 5.5 muestra la estructura interna de un LAB.



**Note to Figure 2-3:**  
 (1) Only from LABs adjacent to IOEs.

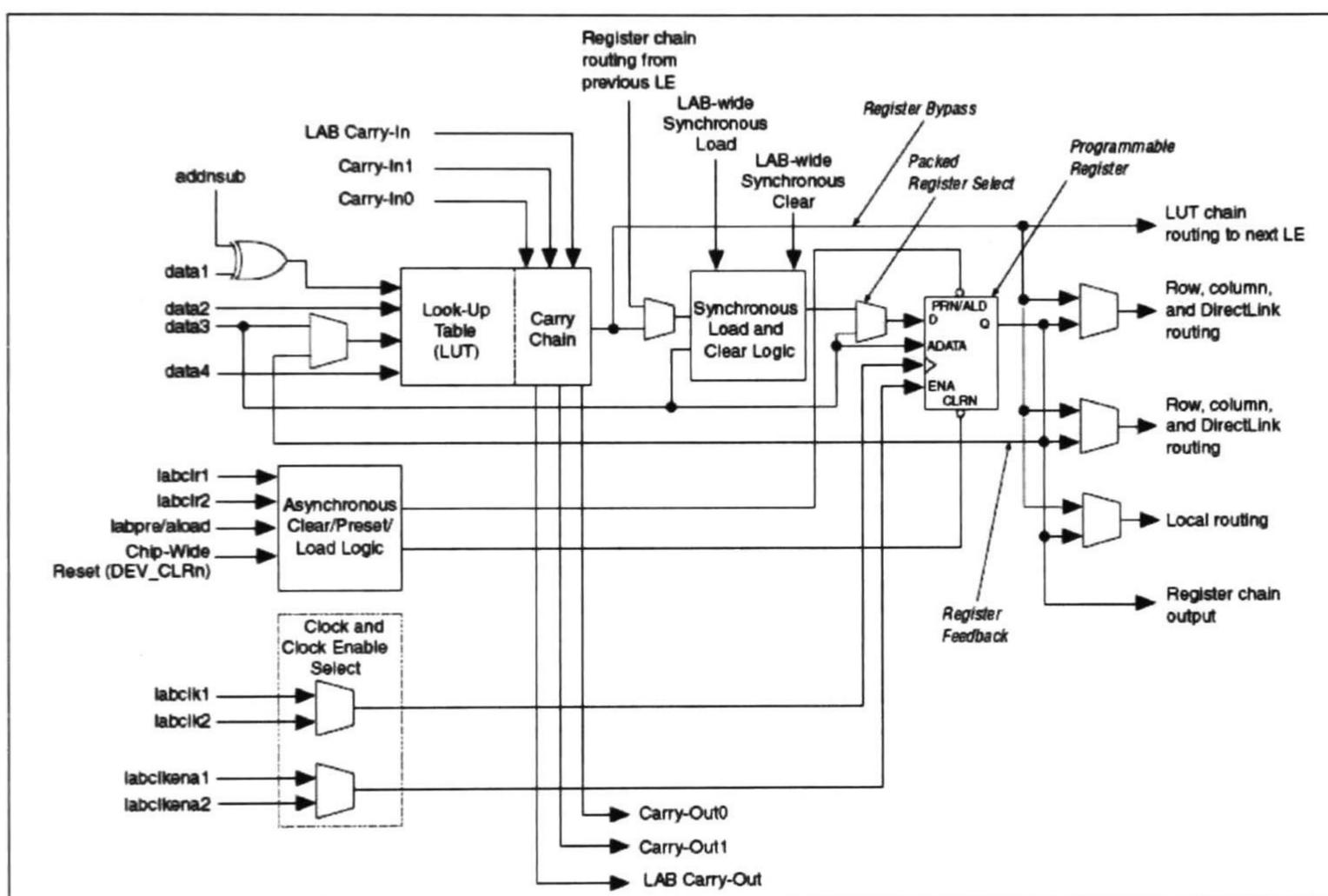
**Figura 5.5 Estructura interna de un LAB de un dispositivo MAXII**

En la figura anterior se puede observar que cada LE puede ser conectado a otro LE dentro del mismo LAB, además hay líneas que pueden conectar un LE hacia otro LAB o hacia otro LE conectado en un LAB diferente. Todo esto minimiza el retraso que pueda existir al ir de un LE hacia otro. Cada LAB contiene lógica dedicada para controlar las señales dentro de cada LE. Estas señales de control incluyen 2 relojes, 2 habilitadores de reloj, 2 señales de borrado (clear) asíncronas, una señal clear sincronía, señales asíncronas de preset y load, una señal sincronía de load y señales de control para suma o resta. Se pueden proveer un máximo de 10 señales de control a la vez hacia un LE. La figura 5.6 muestra la forma en que se generan las señales de control en el LAB.



**Figura 5.6 Generación de señales de control dentro del LAB.**

Finalmente se presenta la arquitectura de los LEs, los cuales contienen 4 entradas hacia los LUTs (look up tables), un registro programable que puede ser configurado como slip flor D, T, JK o SR y puede funcionar de forma sincronía o cargarse datos (load/preset) de forma asíncrona. Para programar lógica combinacional las LUTs pueden brincarse (bypass) el registro y llevar la función directamente a la salida del LE. Cada LE tiene 3 salidas que pueden ser usadas localmente (retroalimentar el LE) o pueden ser ruteadas hacia una fila o una columna y las salidas son controladas independientemente. La siguiente figura muestra el diagrama a bloques y conexiones de un LE.

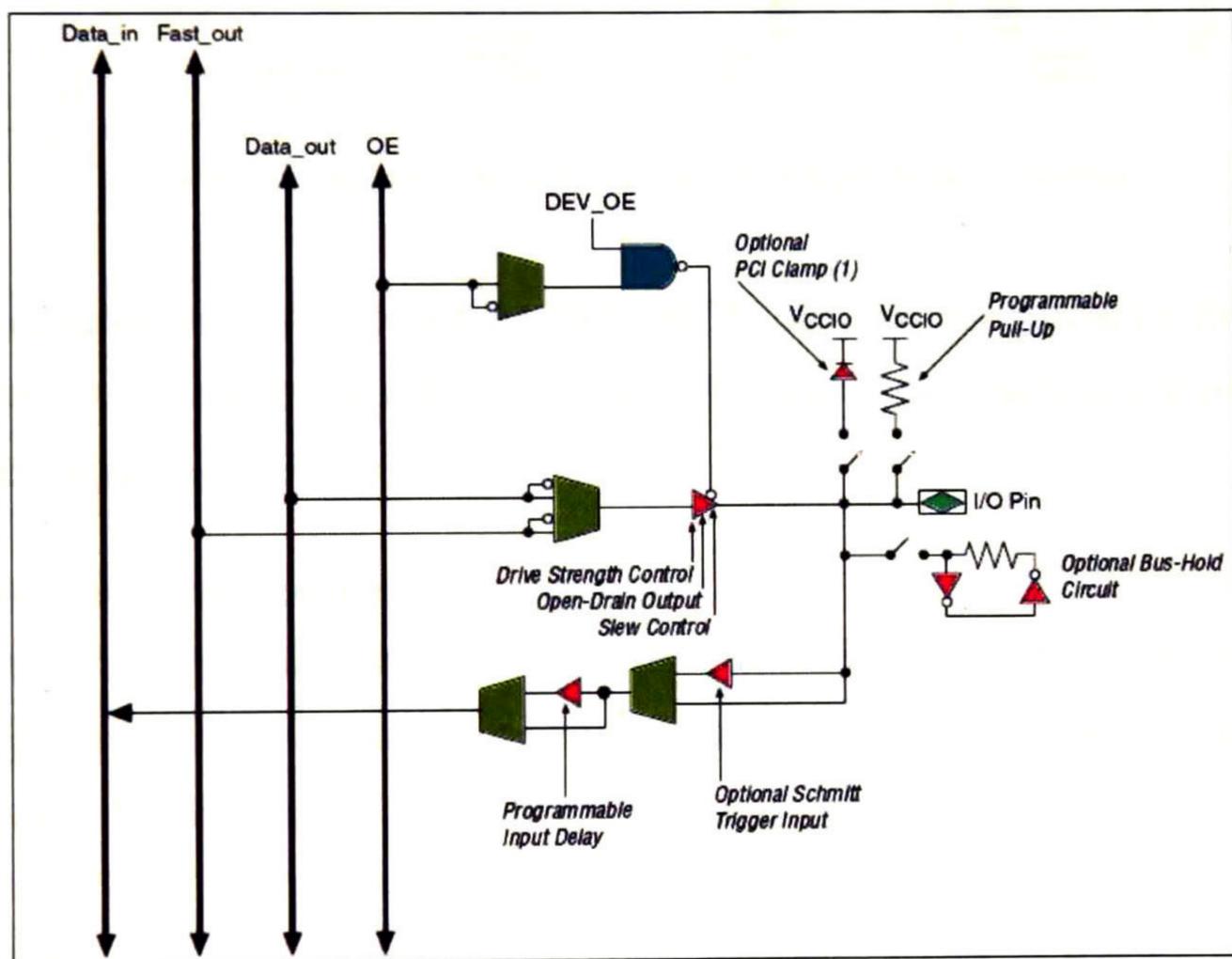


**Figura 5.7** Estructura interna de un LE usado en un dispositivo MAXII

En la arquitectura de los MAXII las interconexiones se llevan a cabo a través de una estructura llamada MultiTrack, la cual consiste de líneas continuas y caminos optimizados para conectar cada uno de los bloques del PLD (LEs, LABs, UFM y pines de IO). La herramienta de compilación Quatus II determina estas interconexiones de forma óptima y el usuario final no debe preocuparse por hacer este tipo de conexión. Si se requiere mayor información de cómo esta estructurada la arquitectura de interconexión se puede consultar en la documentación del dispositivo MAXII de Altera [#]. En cuanto a la memoria flash disponible (UFM) la familia MAXII cuenta con una EEPROM serial para almacenar información hasta 8kbits. Este bloque de

memoria es conectado por medio del MultiTrack hacia los arreglos lógicos permitiendo a los LEs acceder al bloque de UFM. Para mayor información sobre programación y borrado de la UFM referirse a la documentación de Altera [5].

Finalmente se presenta la estructura de IOs soportados para la familia MAXII. Los dispositivos MAXII soportan los estándares LVTTTL, LVCMOS y GTL. Además son compatibles con estándares para PCI, JTAG, resistores de pull-ups y pull-down programables, cuentan con buffers de tercer estado, salidas de colector abierto (open drain), entradas schmitt trigger y retrasos programables. La figura 5.8 muestra la estructura de un IOE para la arquitectura de MAXII.



*Figura 5.8 Estructura de un IOE de la familia MAXII*

Los IOEs se encuentran en la periferia del dispositivo y son agrupados en bloques de IOs. Puede haber hasta 7 IOEs por cada fila y hasta 4 IOEs por columna en cada bloque IO. Cada columna o fila de los bloques IO puede ser conectado a los LABs a través de las conexiones internas de la estructura MultiTrack como se muestra en la siguiente figura.

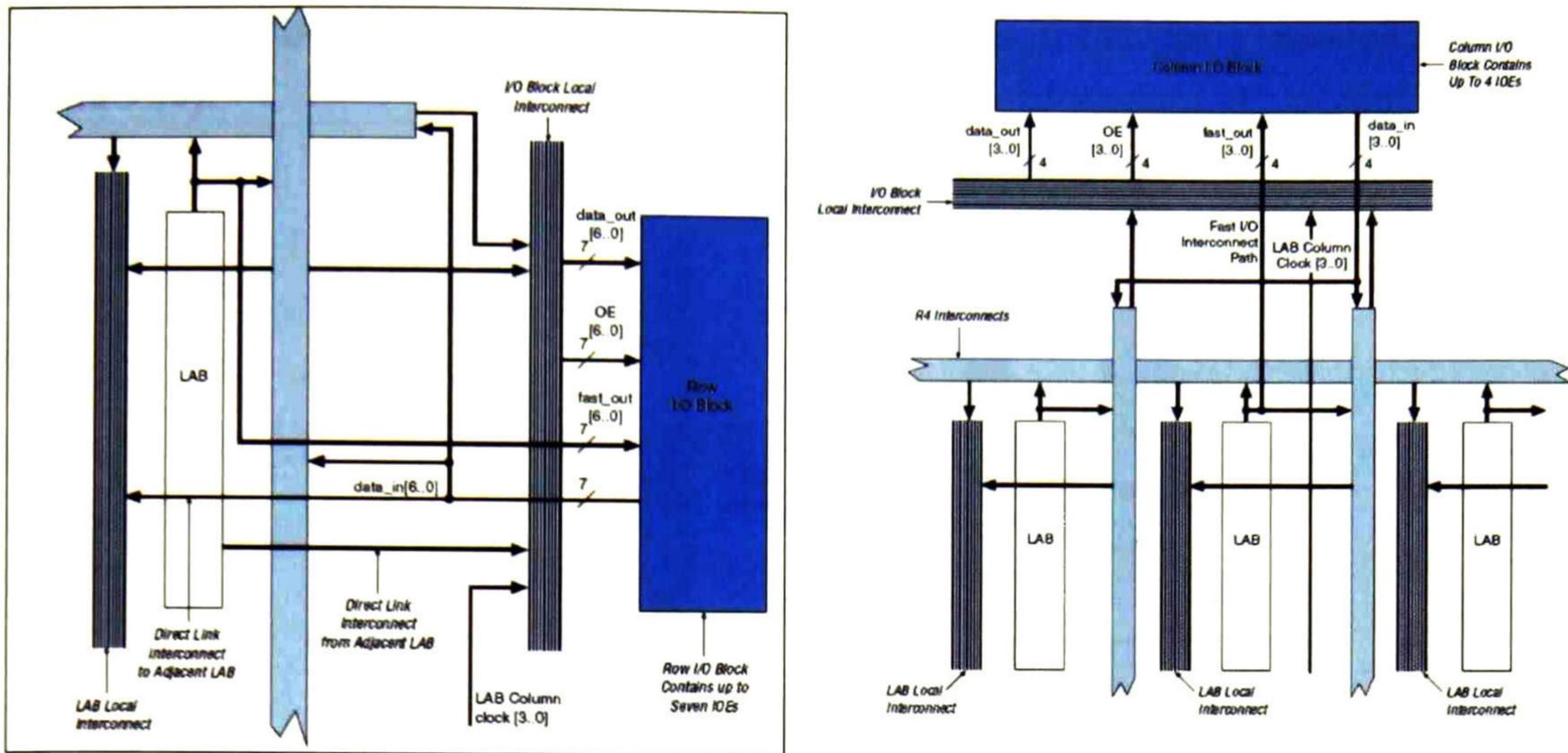


Figura 5.9 Configuración de los bloques de IOs por filas y columnas

El costo de cada dispositivo MAXII (EPM1270F256C5) por parte del proveedor fue de 5 dólares, lo cual hace que este dispositivo pueda ser considerado como opción para usarse en las siguientes plataformas de validación.

### 5.2.2. Dispositivos Altera CycloneII

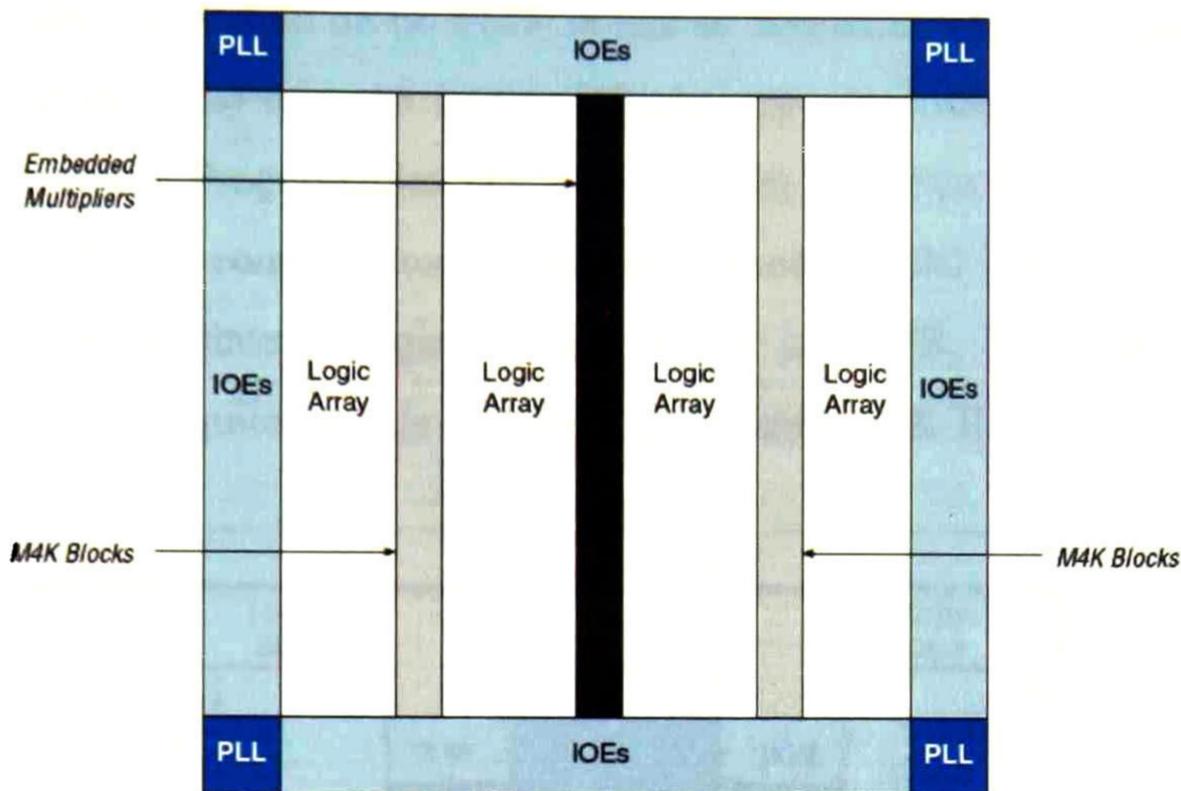
La siguiente tabla muestra un resumen de las características de los dispositivos CycloneII de Altera. El dispositivo que se evaluó corresponde a la parte EP2C15, el cual cuenta con 14,448 LEs y 315 pines de los disponibles. El costo que se cotizó para este dispositivo fue de 28 dólares.

Tabla 5.II Resumen de las características de los dispositivos CycloneII

Feature	EP2C5	EP2C8	EP2C15	EP2C20	EP2C35	EP2C50	EP2C70
LEs	4,608	8,256	14,448	18,752	33,216	50,528	68,416
M4K RAM blocks (4 Kbits plus 512 parity bits)	26	36	52	52	105	129	250
Total RAM bits	119,808	165,888	239,616	239,616	483,840	594,432	1,152,000
Embedded multipliers	13	18	26	26	35	86	150
PLLs	2	2	4	4	4	4	4
Maximum user I/O pins	158	182	315	315	475	450	622

La arquitectura interna de los dispositivos CycloneII esta compuesta por 2 bloques de filas y columnas para implementar funciones lógicas. Las filas y columnas pueden se conectadas a los

bloques de arreglos lógicos (LABs), bloques de memoria embebidos y multiplicadores (ver figura 5.10).



*Figura 5.10 Diagrama a bloques de un dispositivo CycloneII*

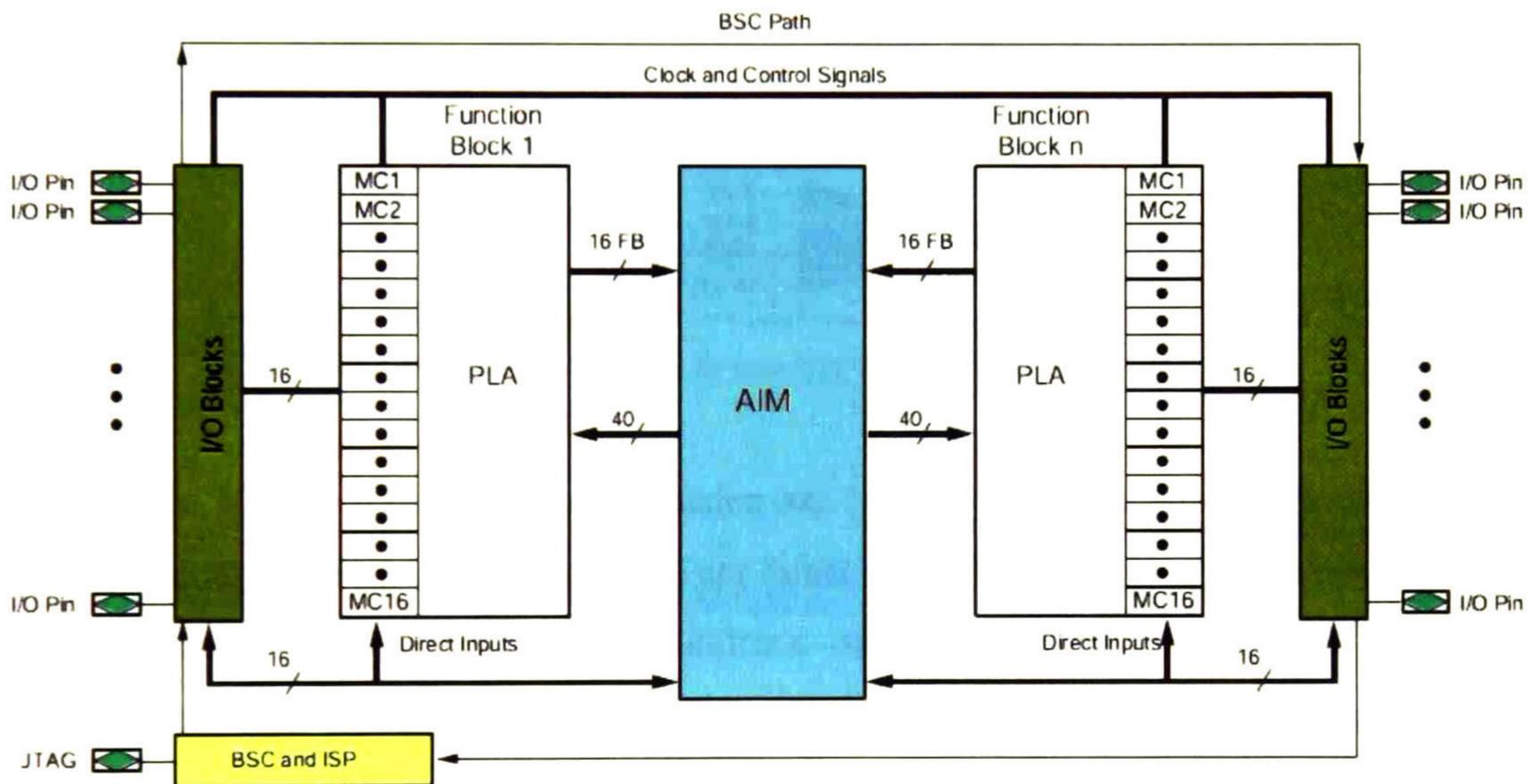
Dentro de cada LAB se encuentran 16 Les, donde cada LE representa la unidad básica para implementar las diferentes funciones lógicas; la arquitectura de cada LE corresponde a la revisada para el caso del PLD MAXII. Se puede apreciar en la figura 5.10 que el dispositivo cuenta con cuatro PLL, los cuales son usados para alinear los relojes de entrada, hacer corrimientos de fase, realizar multiplicaciones o divisiones de reloj, etc; también se observa un bloque M4K de memoria, el cual consiste en memoria interna disponible. Para mayor información sobre los bloques PLL, Multiplicadores, IOEs y memoria interna se puede ir directamente a la especificación del dispositivo CycloneII [5].

### **5.3. Dispositivos de Xilinx**

Los dispositivos de Xilinx que fueron evaluados son el CPLD Cool Runner-II y el FPGA Spartan-3. El CPLD analizado de la familia Cool Runner-II corresponde al XC2C512 el cual cuenta con 12,000 compuertas, 512 macroceldas y 270 IOs de propósito general. El FPGA Spartan 3 evaluado es el XC3S200A el cual cuenta con 200K compuertas y soporta hasta 248 IOs para propósito general.

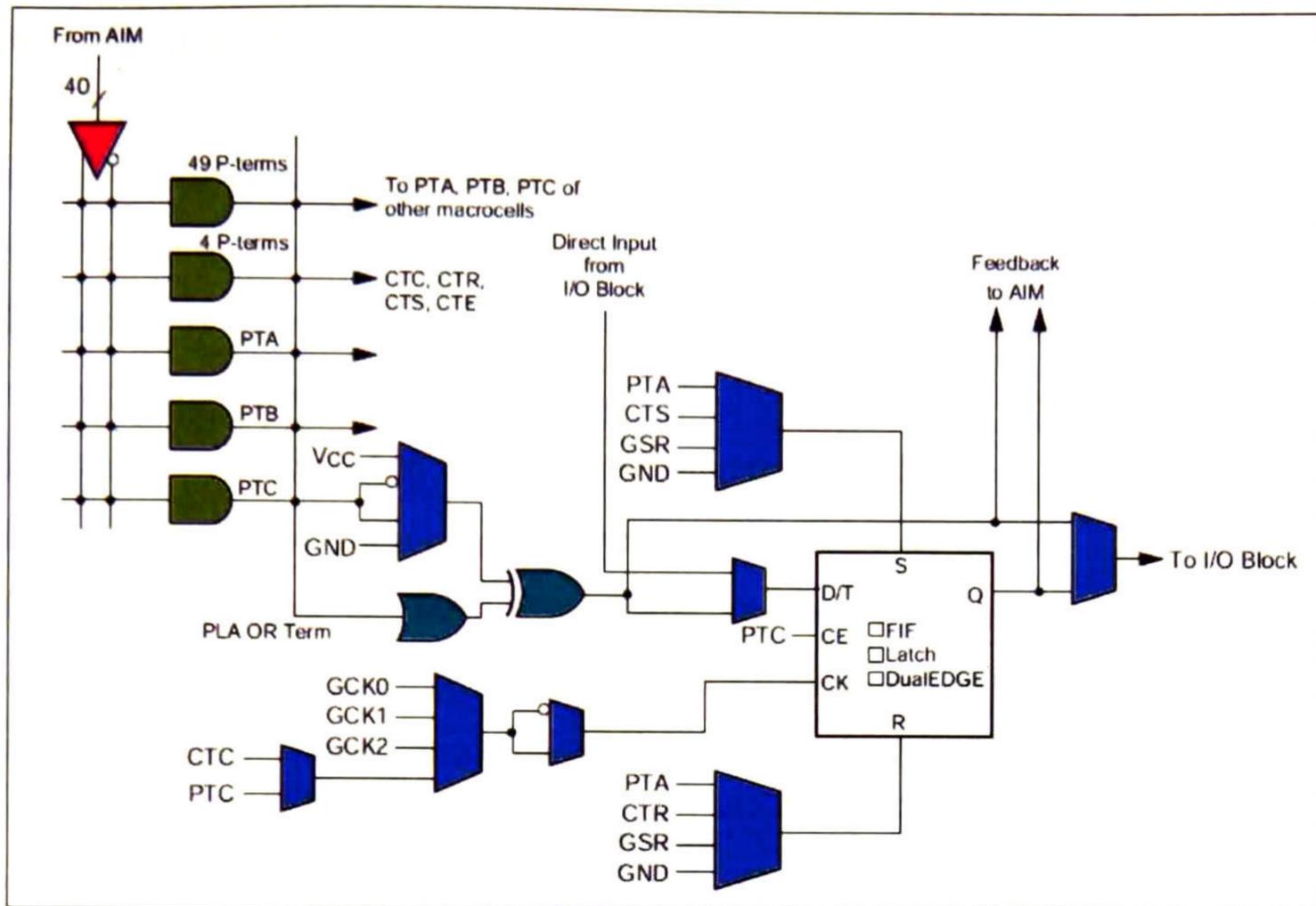
### 5.3.1. Dispositivos Xilinx Cool Runner-II CPLD

La familia de dispositivos Cool Runner-II (CR-II) es conocida por su bajo consume de potencia, y su arquitectura sigue el modelo de un PLD, donde se combinan bloques (macroceldas) dentro de bloques funcionales (FBs) que son interconectados con una matriz global (AIM). Los FBs utilizan Arreglos Lógicos Programables (PLA) los cuales permiten que todos los términos de productos sean ruteados y compartidos en alguna macrocelda del FB. El software de Xilinx optimiza los recursos de síntesis lógica para conectar cada FB. La figura 5.11 muestra el diagrama a bloques de la arquitectura de un CPLD de la familia CR-II.



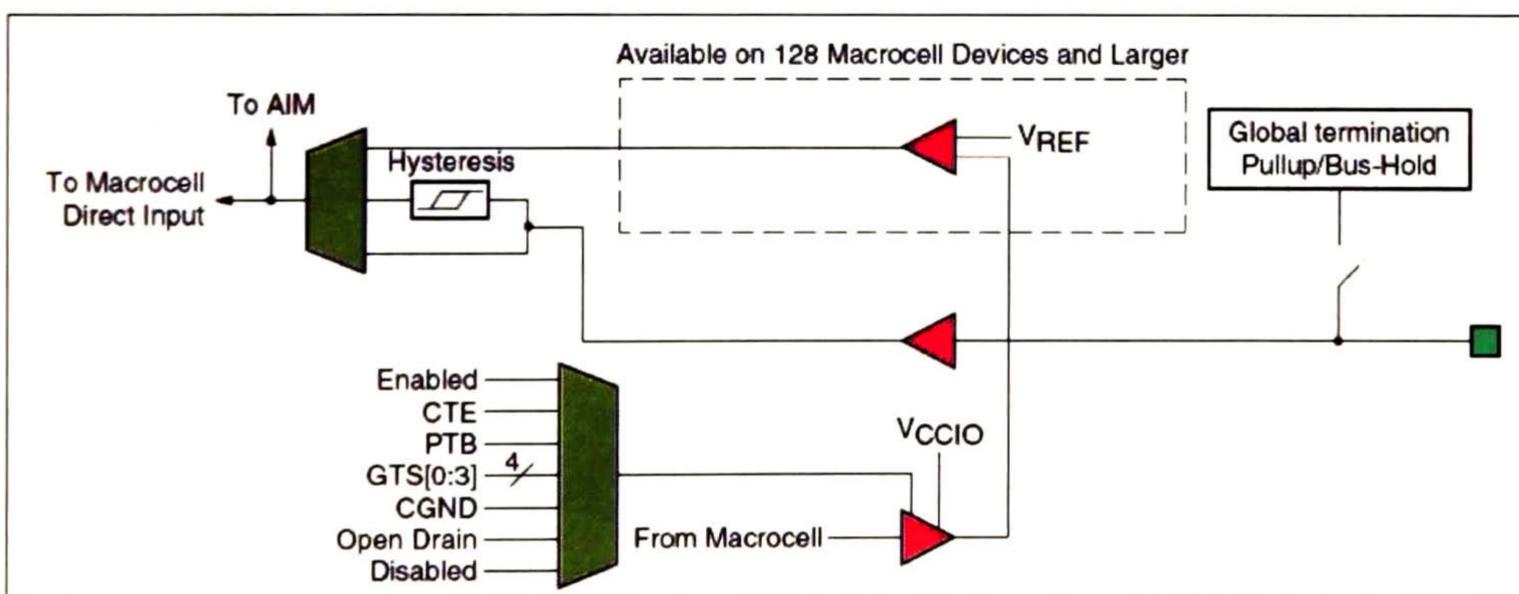
**Figura 5.11** Diagrama a bloques de la arquitectura de un CPLD de la familia CR-II de Xilinx

Cada FB contiene 16 macroceldas con 40 entradas que llegan de conexiones u otra lógica creada en el dispositivo. Dentro de cada FB se tienen términos de productos que residen en un arreglo lógico programable (PLA). El usuario puede desarrollar sumas de productos (SOP) de expresiones lógicas hasta de 40 entradas, y expandir 56 productos de términos dentro de un bloque. Dentro de las macroceldas se pueden combinar expresiones SOP y compuertas XOR, así mismo las funciones lógicas pueden ser puramente combinacionales o con registro, donde se pueden configurar elementos de almacenamiento slip flor tipo D, T o simples Latches. En la figura 5.12 se puede ver la arquitectura interna de una macrocelda de un dispositivo CR-II.



**Figura 5.12** Arquitectura interna de una macrocelda de un dispositivo CR-II

Los bloques de IO son transceivers que pueden ser programados con diferentes estándares de voltajes, adicionalmente cada entrada puede ser conectada a un Schmitt-trigger para agregar un pequeño retardo y filtrar el ruido en la entrada del pin. Los datos de salida pueden ser configurados en tercer estado, colector abierto, pull-up, etc. La siguiente figura muestra el diagrama de un bloque IO.



**Figura 5.13** Bloque IO de un dispositivo CR-II

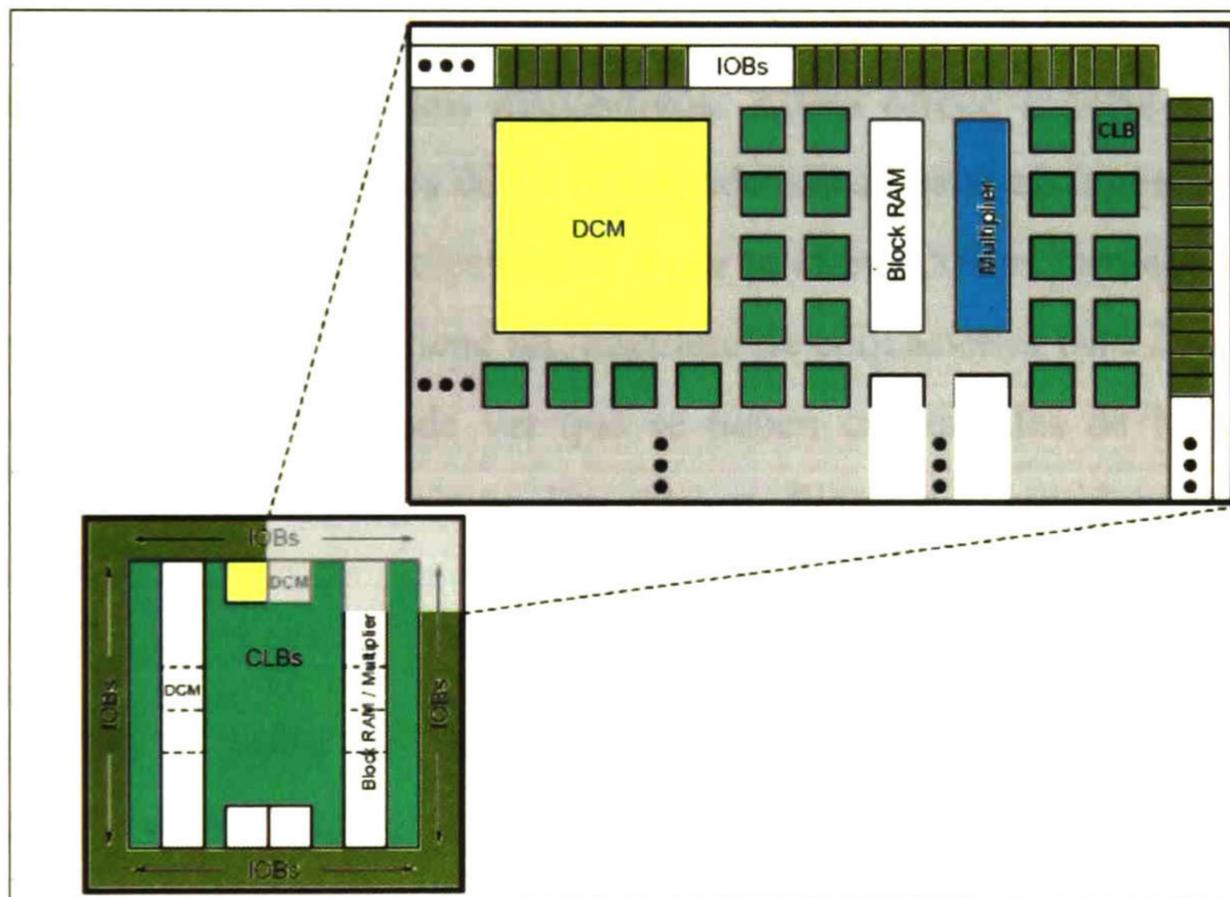
El costo recibido en la cotización para este dispositivo fue de 18 dólares.

### 5.3.2. Dispositivos Xilinx Spartan3 FPGA

La arquitectura de dispositivos Spartan3 esta compuesta por 5 elementos principales:

1. Bloques configurables de lógica (CLBs), los cuales contienen diversos LUTs para implementar elementos lógicos, slip-flops y latches.
2. Bloques de I/O, los cuales controlan el flujo de datos entre los pines de I/O y la lógica interna del dispositivo.
3. Bloques de memoria RAM para proveer almacenamiento
4. Bloques multiplicadores que aceptan números binarios de 18 bits
5. Bloques de manejo de reloj, los cuales proveen funciones para calibración propia, distribución de reloj, retrasos, multiplicadores y divisores de reloj y corrimientos de fase.

En la figura 5.14 se puede observar el diagrama a bloques interno de un dispositivo Spartan3



5.14 Arquitectura interna de los dispositivos Spartan3

Más información sobre los bloques internos DCM, Multiplicadores, CLBs, IOBs y memoria interna se puede consultar en la especificación del dispositivo Spartan3 [6]. El precio de acuerdo a la cotización recibida para este dispositivo fue de 26 dólares.

### 5.4. Códigos compilados para cada dispositivo evaluado

El siguiente paso de acuerdo a la metodología seguida fue compilar los códigos en cada dispositivo y hacer una comparación de recursos usados en cada uno. La tabla 5.III muestra los

resultados de las compilaciones para los dispositivos evaluados, además de mostrarse el total de lógica requerida y pines usados, se muestra el costo total usando cada dispositivo.

*Tabla 5.III Comparación de cada uno de los dispositivos evaluados después de realizar la síntesis de cada código*

Code	MAXII		CycloneII		CoolRunnerII		Spartan3	
	Logic	Pins	Logic	Pins	Logic	Pins	Logic	Pins
<b>MONITOR</b>	93/1270 (7%)	76/212 (36%)	93/18752 (1%)	76/315 (24%)	51/512 (10%)	76/270 (28%)	93/3840 (2%)	76/248 (30%)
<b>PORT80</b>	355/1270 (28%)	62/212 (29%)	359/18752 (2%)	62/315 (20%)	227/512 (45%)	62/270 (23%)	319/3840 (8%)	62/248 (25%)
<b>SERIRQ</b>	167/1270 (13%)	35/212 (17%)	161/18752 (1%)	35/315 (11%)	121/512 (24%)	35/270 (13%)	161/3840 (4%)	35/248 (14%)
<b>RESET</b>	209/1270 (16%)	156/212 (74%)	209/18752 (1%)	156/315 (50%)	181/512 (36%)	209/270 (77%)	213/3840 (5%)	209/248 (84%)
<b>Total Price:</b>	20 USD		112 USD		72 USD		104 USD	

De la tabla anterior se puede ver que el dispositivo MAXII puede cubrir perfectamente los requerimientos del diseño a un precio muy bueno comparado con los otros dispositivos. En cuanto al soporte se puede decir que tanto Altera como Xilinx ofrecen buen servicio para resolver cualquier pregunta relacionada con sus dispositivos. Altera ofrece soporte en línea, donde las respuestas pueden generarse en menos de 24 horas, además se tiene un ingeniero de aplicaciones que puede ser consultado vía correo electrónico o por teléfono. Xilinx también ofrece soporte en línea a través de su página Web y se tiene un Ingeniero de aplicaciones para Latinoamérica.

De la tabla anterior también se puede ver que se tienen capacidades de lógica extra en cada dispositivo, lo cual permitirá programar los nuevos fixtures requeridos por los grupos de validación.

Para reducir aun más el costo del diseño se estudio la opción de integrar los códigos de MONITOR, PORT80 y SERIRQ en un solo dispositivo llamado HOOKS, además se añadieron los fixtures para el grupo de power validation en el código del RESET. La codificación de estos dispositivos se realizo utilizando el lenguaje Verilog, ya que se requería estar de acuerdo con los guidelines de los grupos de diseño de plataformas. Los resultados de síntesis obtenidos integrando estos tres códigos y los requerimientos de power validation se muestran en la siguiente tabla.

*Tabla 5.IV Tabla que muestra la síntesis y comparación de precios para cada dispositivo con los códigos optimizados*

	MAXII		CycloneII		CoolRunnerII		Spartan3	
Code	Logic	Pins	Logic	Pins	Logic	Pins	Logic	Pins
<b>HOOKS</b>	619/1270 (49%)	167/212 (79%)	593/18752 (3%)	167/315 (53%)	398/512 (78%)	167/270 (62%)	577/3840 (15%)	167/248 (67%)
<b>RESET</b>	610/1270 (48%)	194/212 (92%)	874/18752 (5%)	194/315 (62%)	Cant fit		569/3840 (14%)	194/248 (78%)
<b>Total Price:</b>	10 USD		56 USD		36 USD		52 USD	

Los códigos del nuevo dispositivo HOOKS mas el RESET fueron simulados con la herramienta de ModelSim para verificar que fueran correctos. El siguiente paso fue hacer una tabla con ponderaciones para determinar cual dispositivo cumple con los requerimientos y es el más óptimo en cuanto a capacidades y costo. Se realizo una tabla donde se tomo en cuenta el numero de compuertas disponibles, pines, costo, soporte y herramientas de programación y verificación; el rango tomado fue de 1 a 5 donde 1 corresponde a la ponderación mas baja y 5 a la mas alta. La tabla 5.V muestra los resultados registrados para cada una de las familias evaluadas.

*Tabla 5.V Resultados obtenidos para cada uno de los dispositivos evaluados*

Dispositivo	Compuertas lógicas disponibles	Pines disponibles	Costo	Soporte	Herramientas de programación y verificación	Totales
MAXII	4	3	5	5	5	22
CycloneII	5	3	1	5	5	19
CoolRunnerII	1	2	3	5	5	16
Spartan3	5	3	1	5	5	19

El dispositivo que se adecua mas a las necesidades y requerimientos fue el CPLD MAXII, un punto importante de este dispositivo fue su costo (5 dólares cada dispositivo). Cada tarjeta de validación requerirá el uso de dos dispositivos MAXII, dando un total de 10 dólares, que comparados con la familia de dispositivos que se usaron en las plataformas previas (MAX7000) resulto una opción muy viable y atractiva, ya que el costo total por tarjeta utilizando el dispositivo MAX700 alcanzaba los 132 dólares. La demanda de tarjetas de validación para el chipset ICH10 fue de 270, por lo tanto utilizando la familia de CPLDs MAXII el costo total de los dispositivos programables fue de 2,700 dólares, en cambio se hubiera continuado usando la familia MAX7000 el costo total de los dispositivos programables hubiera sido de 35,640 dólares

(132\*270). El ahorro por tarjeta fue de 122 dólares y el ahorro total considerando las 270 tarjetas fabricadas fue de 32,940 dólares.

El estudio de factibilidad fue presentado al líder de proyecto (Javier Camino) y después fue presentado a la gerencia, donde se analizó y se le dio el visto bueno para llevarse a cabo.



# Solución implementada

---

*Se presenta la solución implementada, diagrama a bloques de la arquitectura de la tarjeta de validación, las características de diseño así como los diagramas a bloques de los dispositivos programables implementados.*

---

## 6. Solución implementada

En esta parte del documento se incluye la arquitectura a bloques de la tarjeta de validación ICH10 donde se usaran los dos dispositivos MAXII evaluados, también se incluirán a manera de bloques los diagramas de las implementaciones en los dos dispositivos HOOKSPLD y RSTPLD.

### 6.1. Arquitectura de la tarjeta de validación ICH10

El diagrama a bloques de la tarjeta de validación para el chipset ICH10 se muestra en la figura 6.1, se puede observar que en vez de usar los 4 dispositivos programables que se usaban en las plataformas previas ahora se están usando solo dos (HOOKS y RST).

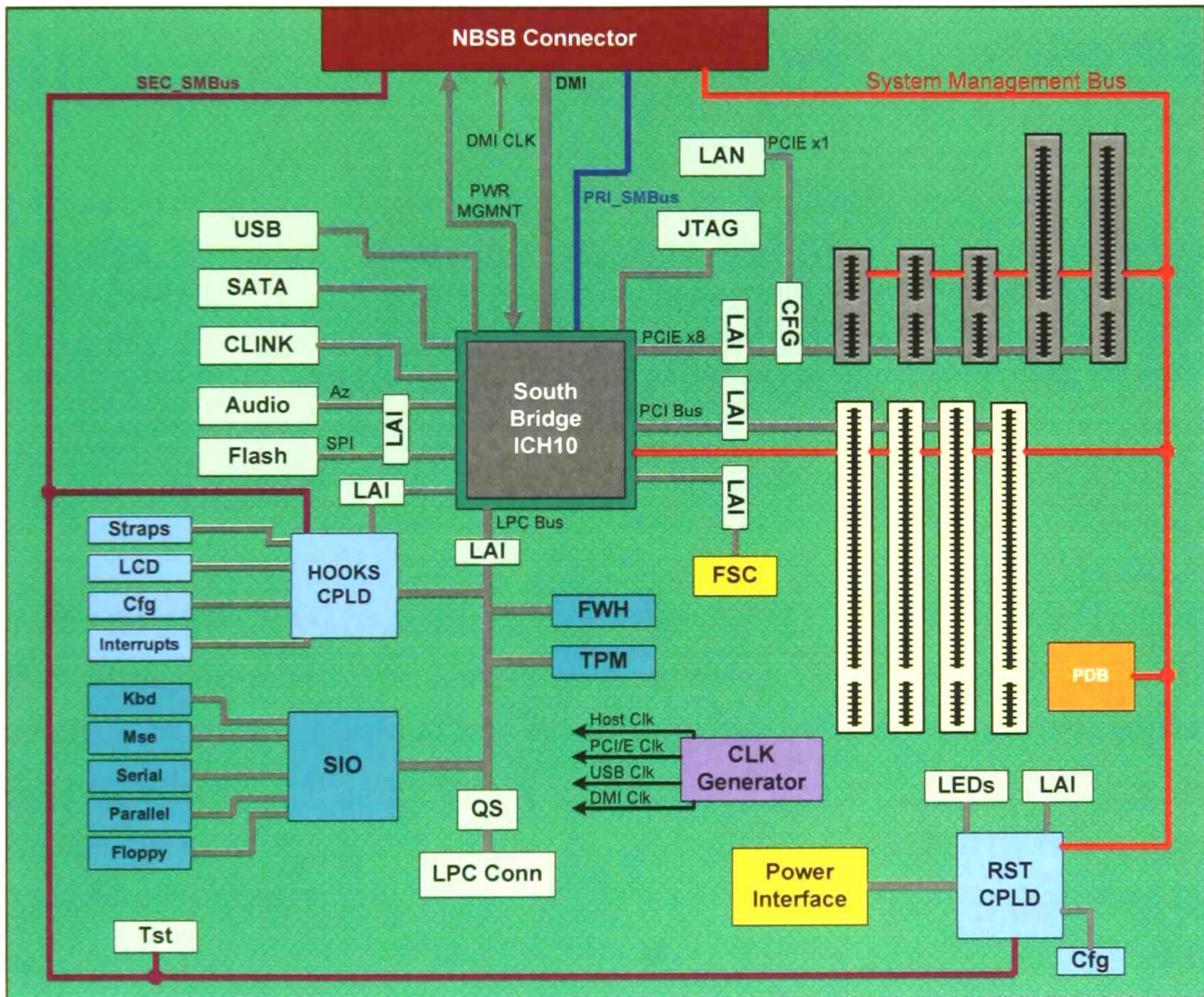


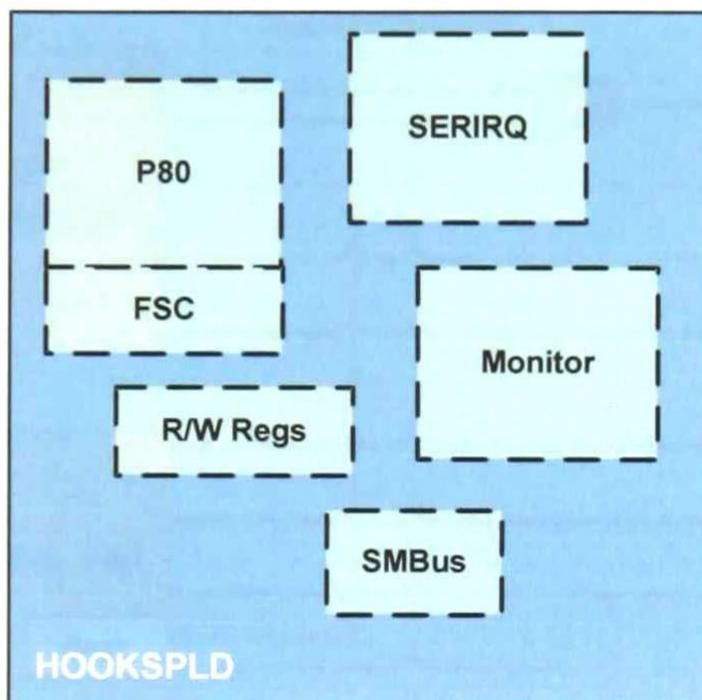
Figura 6.1 Diagrama a bloques de la plataforma de validación para el chipset ICH10

### 6.2. Dispositivo HOOKS PLD

El dispositivo programable llamado HOOKS [7] está compuesto de diferentes funciones lógicas para validar diversas interfaces en la tarjeta de validación ICH10. Como se ha mencionado en el documento, en las plataformas previas a la ICH10 se usaban 4 PLDs para cubrir las necesidades

de validación de la plataforma, pero en esta nueva tarjeta se presenta una arquitectura con solo 2 PLDs (HOOKS y RST).

El dispositivo HOOKS tiene tres conexiones principales, una es hacia el bus secundario serial (SEC SMBus), la otra es hacia el bus LPC y la última se conecta a diferentes pines de configuración del ICH10. Los bloques principales del HOOKSPLD son el Puerto 80, el control de ventiladores, interrupciones seriales para PCI, SMBus, puerto monitor y registros de lectura y escritura (ver figura 6.2).



*Figura 6.2* Bloques contenidos dentro del dispositivo HOOKSPLD

### 6.2.1. Bloques P80 y FSC

El bloque interno de P80 y FSC corresponde a la captura de códigos desde el bus LPC y el control de ventiladores del sistema. Dentro del bloque de P80 existe un módulo que se encarga de decodificar el Bus LPC, este módulo contiene una máquina de estados la cual recibe las entradas del bus LPC y las decodifica todo el tiempo, la información decodificada es almacenada en registros, los cuales pueden ser leídos a través de SMBus o pueden ser desplegados en una pantalla de 8 segmentos. El bloque FSC se encarga de recibir las señales de PWM y TACH del chipset y las distribuye de acuerdo a la configuración que se programe, es decir, estas señales pueden ser distribuidas a los diferentes ventiladores contenidos en la tarjeta y pueden estar configurados como ventiladores de 3 o de 4 pines. El diagrama a bloques de los módulos P80 y FSC se muestra en la figura 6.3.

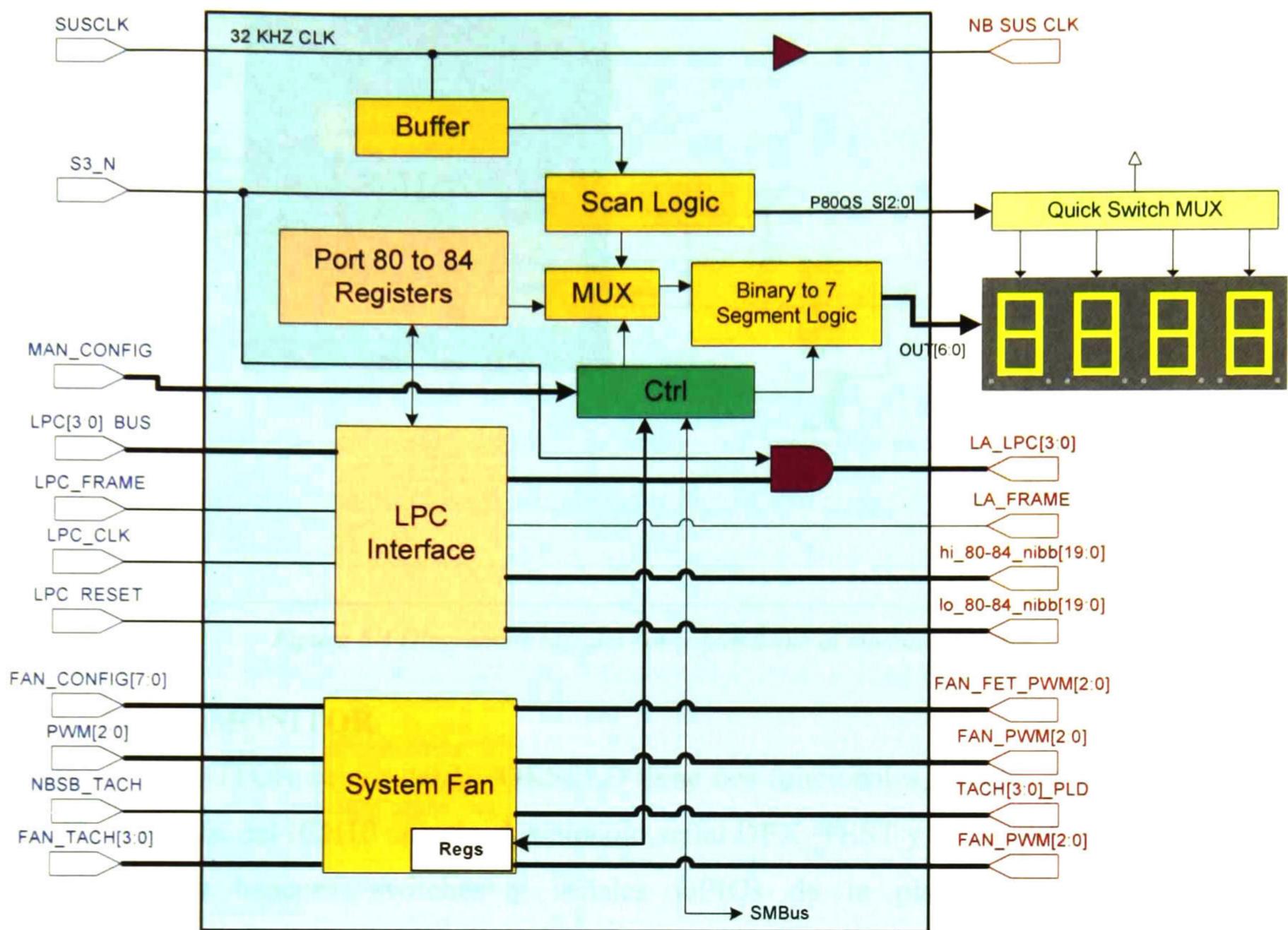
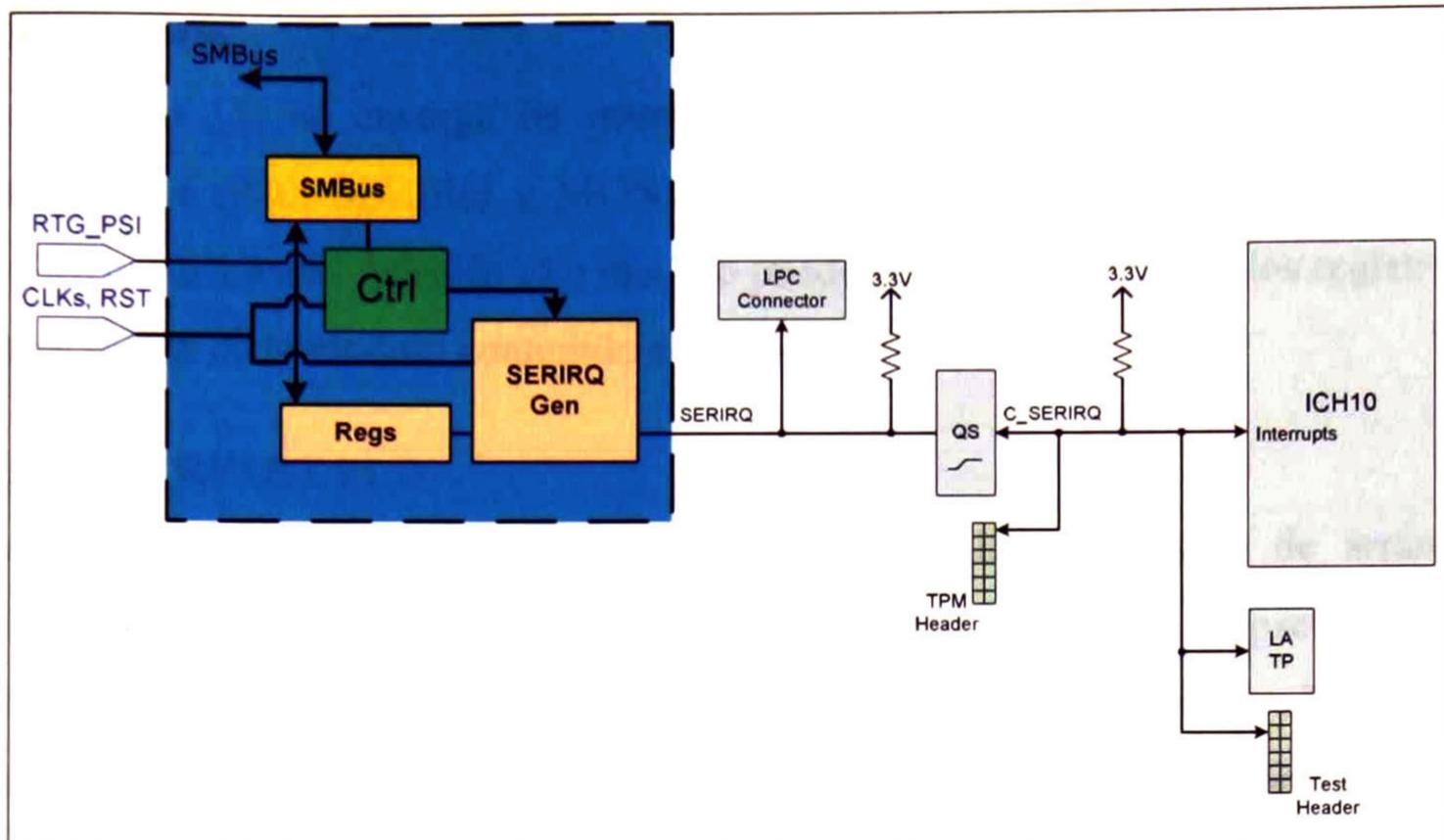


Figura 6.3 Diagrama a bloques correspondiente al P80 y FSC

### 6.2.2. Bloque SERIRQ

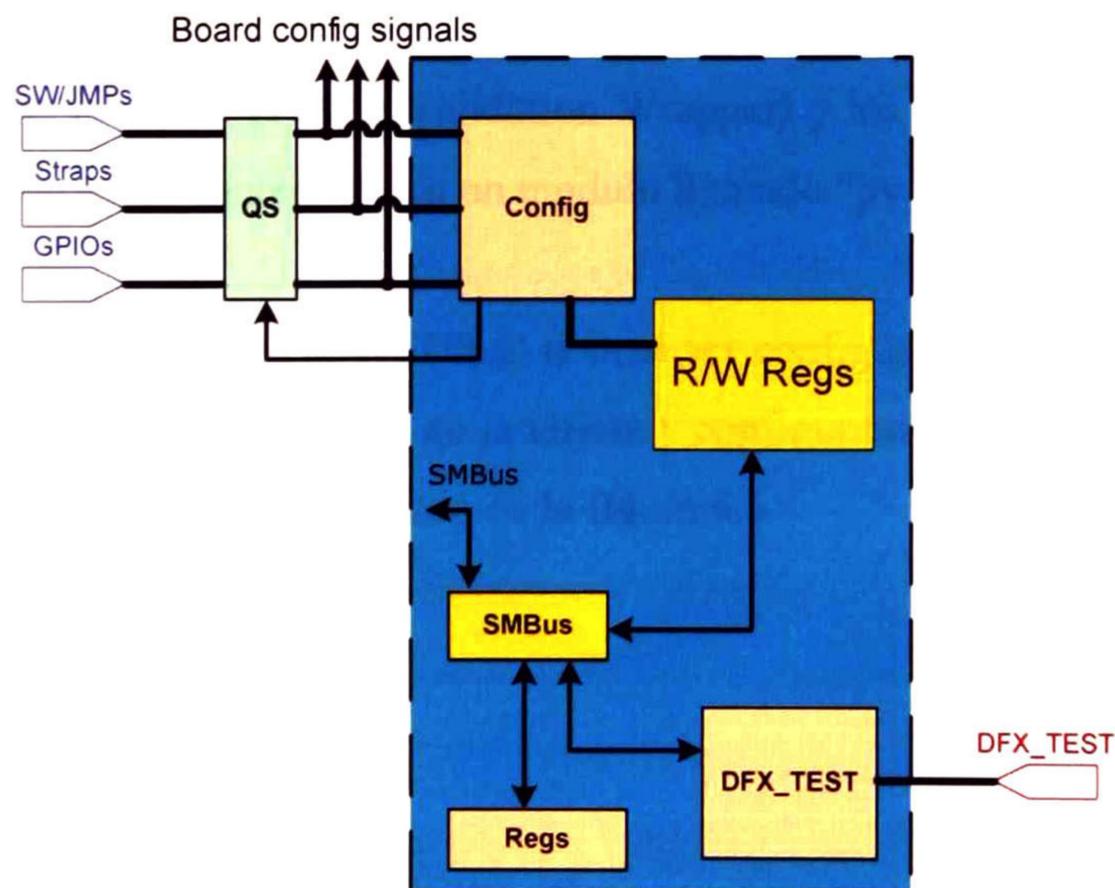
El bloque SERIRQ [9] se encarga de generar interrupciones seriales de PCI para validar y debugear el modulo de interrupciones del Chipset. Dentro del bloque SERIRQ hay registros en los cuales se programan cuales interrupciones serán generadas así como los tiempos en que serán activadas. Los registros internos son configurados a través de SMBus. El diagrama a bloques del modulo SERIRQ es mostrado en la figura 6.4.



**Figura 6.4** Diagrama a bloques correspondiente al modulo SERIRQ

### 6.2.3. Bloque MONITOR

El bloque MONITOR dentro del HOOKSPLD tiene dos funcionales, la primera es acceder a los registros internos del ICH10 usando el protocolo serial DFX\_TEST y la segunda funcionalidad es monitorear los headers, switches y señales GPIOs de la plataforma y guardar estas configuraciones en registros internos R/W. La figura 6.5 muestra un diagrama a bloques del modulo MONITOR.



**Figura 6.5** Diagrama a bloques correspondiente al modulo MONITOR

#### **6.2.4. Bloque SMBus**

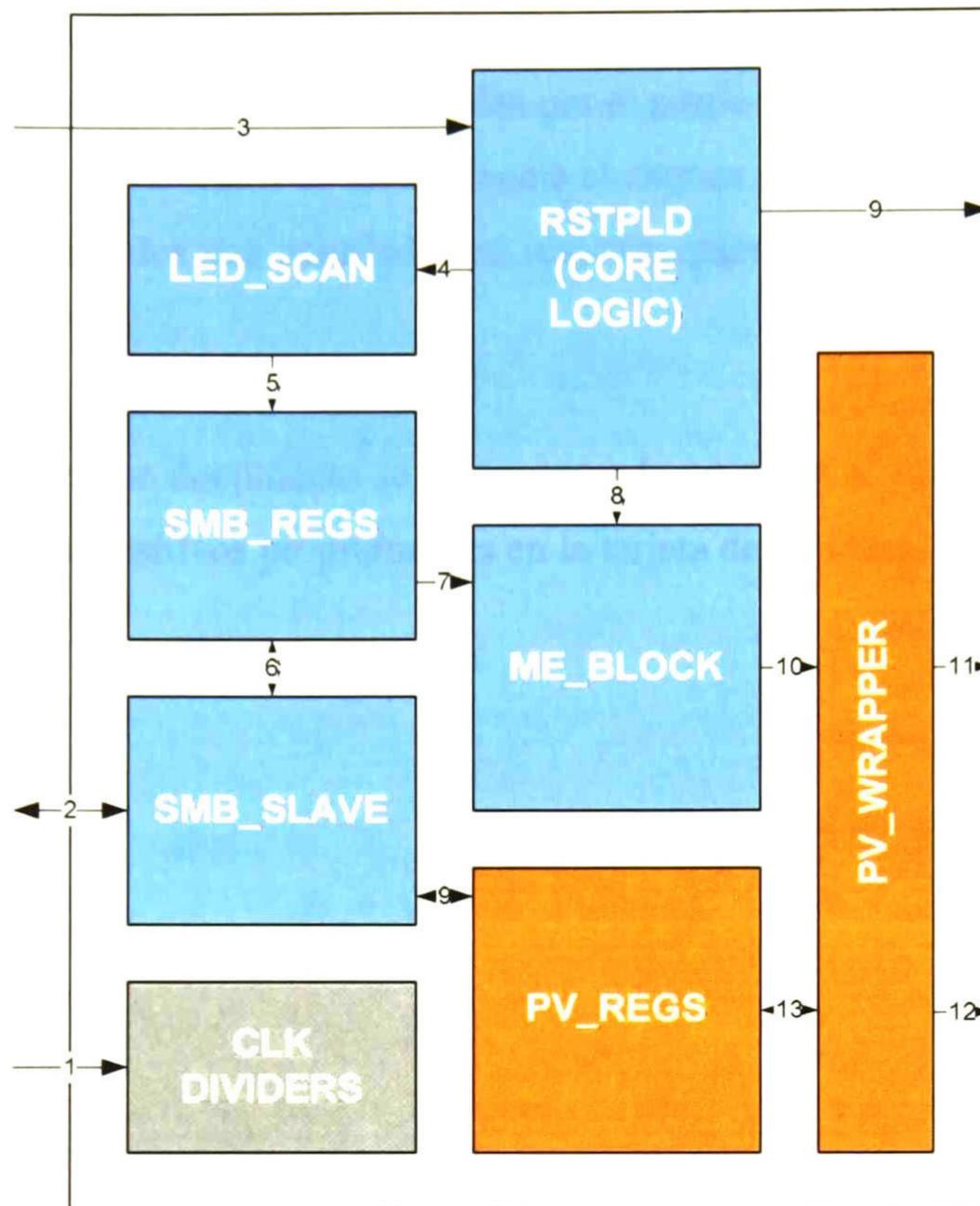
El bloque SMBus [8] se encarga de interpretar y direccionar los comandos SMBus a los diferentes módulos (P80, SERIRQ y MONITOR) ya que cada uno de los módulos tiene una dirección SMBus diferente. Además este modulo puede acceder y modificar los registros internos de configuración de cada modulo contenido en el dispositivo.

#### **6.3. Dispositivo RESET PLD**

El dispositivo RESETPLD [10] se encarga de controlar las secuencias de arranque de la plataforma y configura diferentes estados en los que deben trabajar el chipset y el CPU. Este dispositivo es en gran medida el responsable de poder encender la plataforma adecuadamente, así mismo genera mensajes de error que se puedan presentar en la inicialización de los diferentes planos de poder y se tiene la capacidad de desplegarlos externamente a través de LEDs o ser monitoreados por SMBus usando registros internos de lectura dentro del RSTPLD.

Este dispositivo a parte de hacer las funcionalidades mencionadas anteriormente provee nuevas rutinas que fueron requeridas por los grupos de power validation, las cuales incluyen el retraso de eventos de diferentes señales y programación de tiempos de activación usando SMBus. Además se pueden inducir eventos que generan errores dentro del chipset como sobrecalentamiento del procesador, error de sistema, error catastrófico, etc. Todos estos eventos pueden ser generados a través de comandos SMBus. Estas nuevas implementaciones están contenidas dentro de un modulo llamado “pv\_wrapper” (Power Validation Wrapper) y los registros de SMBus asociados a esta implementación están contenidos en un modulo llamado “pv\_regs”.

Para activar estos nuevos fixtures el RSTPLD deberá ser configurado en PV\_MODE, lo cual se logra quitando una resistencia contenida en la tarjeta y configurando un dipswitch. El diagrama a bloques del dispositivo RESET es mostrado en la figura 6.6.



**Figura 6.6** Módulos contenidos dentro del RESET PLD

Las interconexiones que se muestran en la figura anterior son las siguientes:

1. Entradas de reloj (33MHz y 32KHz)
2. Interfaz SMBus
3. Señales de entrada de straps, señales de CPU, señales de monitoreo de reguladores y señales de estados internos del chipset.
4. Señales que se monitorean a través de LEDs externos al dispositivo.
5. Señales W/R contenidas en registros internos.
6. Bus de acceso hacia los registros internos del dispositivo
7. Señales que se usan para configurar los registros internos de ME usando SMBus.
8. Configuración manual para ME usando straps externos
9. Salidas que controlan los diferentes FETs, reguladores y secuencias de activación
10. Señales que pueden ser manipuladas (generar retrasos o activaciones) a través del PV\_WRAPPER.

11. Cuando no esta activo el modulo PV\_WRAPPER, las señales pasan sin sufrir algún cambio (estas señales fueron determinadas por el grupo de PV)
12. Señales que generan eventos de errores hacia el chipset o CPU.
13. Los eventos generados son guardados en registros para ser leídos o modificados a través de SMBus.

En la siguiente parte de este documento se presentaran los datos y resultados obtenidos con esta nueva arquitectura de dispositivos programables en la tarjeta de validación para el chipset ICH10.



# Datos y resultados

---

*En este apartado son reportadas las pruebas realizadas al producto, con la finalidad de verificar el correcto funcionamiento del dispositivo. Se realiza una comparación entre las especificaciones inicialmente planteadas y los resultados obtenidos.*

---

## **7. Datos y resultados**

En esta parte del documento se presentarán las pruebas iniciales una vez que se fabrico la tarjeta de circuitos impresos, así como las pruebas funcionales y su análisis de desempeño de los dispositivos programables.

### **7.1. Plan de pruebas de la tarjeta de validación ICH10**

La tarjeta fabricada requiere de pasar ciertas pruebas antes de ser usada para el power-on del chipset, para esto se hace un plan de pruebas [11] donde se establecen los criterios y se definen los métodos para validar las diferentes interfaces de la tarjeta en cada una de sus fases. Esta parte del documento mencionara algunas de las pruebas que se hicieron a la tarjeta antes de ser validadas las implementaciones de los dispositivos programables (HOOKS y RESET) solo para referencia del lector.

La primera prueba que se realizo fue un chequeo mecánico, donde se imprimieron las capas superior e inferior de la tarjeta y se prosiguió a verificar que los componentes físicos no tuvieran errores mecánicos (esto antes de mandar a fabricar el PCB). Con respecto a las huellas de los nuevos dispositivos programables (MAXII) no se tuvo problema, ya que su dibujo mecánico coincidía con el presentado en las hojas de especificación y con el dispositivo físico. Después de hacer este chequeo mecánico se prosiguió a fabricar el PCB, donde se especifican las impedancias que deberá soportar el fabricante.

Para verificar que la tarjeta fabricada cumple con las especificaciones dadas se ponen en el PCB puntos de prueba diferenciales en las diferentes capas para medir las impedancias y corroborar con un equipo TDR que sean correctas. En el caso de los PLDs no se requirió utilizar señales de impedancias controladas, solo se utilizaron señales con impedancias de 50 ohms. Una vez realizadas las pruebas de TDR se procede a ensamblar la tarjeta, y cuando llega la tarjeta ensamblada se hace una inspección visual, donde el objetivo es verificar que los componentes estén puestos correctamente, que los capacitores tengan correcta su polaridad, que no existan cortos o puentes de soldadura y que el silkscreen sea visible. Para los PLDs no se detectaron cortos o puentes de soldadura en sus pads, y como son encapsulados BGA no se puede determinar visualmente si hay un mal contacto en sus bolas de soldadura. Para este tipo de componentes (BGAs) el ensamblador realiza una prueba de rayos X con la cual pueden verificar

que el componente quedo bien soldado en la tarjeta. Para el caso de los PLDs no se reportaron fallas en el ensamble.

Enseguida se procede a verificar que la tarjeta no tenga cortos entre sus planos, para esto se mide la resistencia entre cada plano y se verifica que no tengan continuidad entre ellos. Los PLDs manejan dos tipos de planos de poder, uno es para alimentar su core (1.2V) y otro para los IOs (3.3V); en esta etapa se verifico que estos planos estuvieran aislados. Enseguida se enciende la tarjeta (aun sin el chipset) y se verifica que todo este correcto (generalmente se le conoce como la primera prueba de fuego, ya que se energiza la tarjeta y si no sale algo volando o comienza a salir humo es un buen principio). Una vez energizada la tarjeta se miden los diferentes voltajes de la tarjeta (alrededor de 80 diferentes planos fueron verificados).

La siguiente prueba que involucra a los PLDs es verificar que los dispositivos puedan ser reconocidos a través de su cadena JTAG. Para esto se conecto el cable USB-Blaster en el header de JTAG y se verifico que los dos dispositivos fueran leídos. Nota: Generalmente se deben de instalar los driver de los cables de JTAG en la computadora, el fabricante (en este caso Altera) pone en su Web los drivers necesarios y el procedimiento de instalación para poder usar sus cables. Una vez que se pueden detectar los PLDs se programan con códigos especiales que sirven para habilitar algunos dispositivos como generadores de reloj, buffers, LEDs, etc. En el caso de la tarjeta de validación ICH10 se programo el RESET para que encendiera los diferentes planos de la tarjeta, activas algunas secuencias de arranque, habilitar los FETs y verificar que los LEDs funcionaran correctamente.

Una vez cubiertas estas pruebas se procede a encender la tarjeta con el chipset anterior a la plataforma diseñada (en este caso con un ICH9) y con un interponer que fue diseñado por el grupo de Guadalajara. Al encender la plataforma con un chipset anterior permite tener mayor seguridad de que el hardware no tiene errores y la tarjeta esta funcionando correctamente. Al encendido de la tarjeta con un chipset anterior y con un interponer se le conoce como pre-power on.

Para realizar las pruebas de pre-power on se monta el interposer y el chipset, se programan las memorias con el BIOS correspondiente, se programan los PLDs y supervisores de voltaje y se monta un sistema operativo DOS para arrancar la plataforma. Para el caso de la tarjeta de validación ICH10 se tuvieron que hacer algunas modificaciones a los códigos de los PLDs, ya que el chipset ICH9 requería de habilitar algunas señales de GPIO y algunos straps para poder ser inicializado correctamente. Después de hacer los ajustes correspondientes el sistema pudo ser levantado correctamente utilizando DOS, Windows y Linux.

## **7.2. Pruebas funcionales a los nuevos PLDs (MAXII)**

Una vez que la tarjeta pudo ser levantada con un sistema operativo se comienzan a correr diferentes pruebas para probar cada una de las interfaces, pero en este caso la que se requiere verificar y documentar es la relacionada a los nuevos PLDs. Las pruebas que se realizaron a estos nuevos dispositivos fueron las siguientes:

- Se verifico que los registros internos fueran accesados por SMBus, para esto se utilizo la herramienta iPort, la cual es una interfaz I2C que conecta la PC con la plataforma y en la cual se pueden enviar comandos de lectura y escritura. Los registros del RESET y del HOOKS PLD fueron accesados correctamente.
- Se verifico que los arreglos de LEDs encendieran las secuencias correctamente con el RESET PLD.
- Se modificaron manualmente los straps para el RESET y el HOOKS PLD y se verifico que la activación de módulos fuera correcta.
- Para el RESET PLD se habilito el modulo PV-WRAPPER y se pudieron retrasar y activar las secuelas de señales correctamente. Así mismo se valido que los errores internos generados hacia el ICH9 funcionaran correctamente.
- Se verifico que los registros fueran W/R dentro del HOOKS PLD.
- A través del display de 7 segmentos se validaron los diferentes códigos de BIOS al encender la plataforma.
- La interfaz de FSC dentro del HOOKS PLD no estaba funcionando correctamente debido a que se requería habilitar pull-ups internos dentro del PLD para estas señales. Los Pull-

ups son habilitados con la herramienta de Quartus al generar el archivo de configuración del PLD.

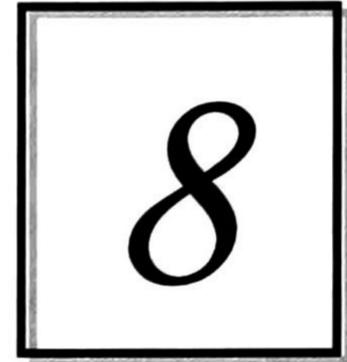
- Se verifico que las interrupciones seriales fueran generadas correctamente. En esta parte se tuvieron algunos problemas ya que la maquina de estados programada al momento de ser implementada en compuertas generaba un glitch al ir de un estado a otro. Se encontraron dos soluciones a este problema, una consistió en indicarle a la herramienta de Quartus que optimizara los recursos para rutear esta señal libre de retrasos y por el camino mas corto. La segunda opción fue utilizando la herramienta Synplicity, la cual esta diseñada para optimizar los recursos de los dispositivos de acuerdo al código implementado. Después de usar alguna de estas dos alternativas se pudo validar este modulo.
- Se leyeron algunos de los registros internos permitidos dentro del ICH9 utilizando el protocolo DFX\_TEST con el modulo MONITOR.

Una vez hecha esta verificación se tuvo mayor confianza en el hardware diseñado para poder ser usado en el power-on de la tarjeta utilizando el nuevo chipset ICH10.

### **7.3. Power-on de la tarjeta ICH10**

La siguiente etapa importante en el proyecto es el power-on del chipset ICH10. Pero con el pre-power on realizado previamente se tiene una mayor confianza en que el diseño tendrá menos problemas de los que se pudieran esperar. Durante el power-on se trabaja bastante con el BIOS del sistema y con las secuencias de arranque e inicialización del chipset. BIOS realiza los ajustes correspondientes para activar registros y el grupo de plataformas realiza lo cambios necesarios en las secuencias de reset para poder levantar la plataforma correctamente. Los procesos de power-on pueden llevar desde unas cuantas horas, un par de días o varias semanas, dependiendo que tan sano este el silicio y que el hardware diseñado este correcto. Para el caso de la tarjeta de validación ICH10 se requirieron un par de días para levantar todo el sistema debido a las inicializaciones de BIOS.

Una vez concluido el power-on del chipset se procede a enviar las plataformas a los diferentes grupos de validación, los cuales comenzaran a realizar pruebas especializadas y especificas a las interfaces y al nuevo chipset.



# Conclusiones

---

*En este capítulo se presentan todas las conclusiones recabadas durante la realización del presente proyecto, así como el enfoque que se le fue dando al producto en cada una de las etapas realizadas.*

---

## **8. Conclusiones**

El problema presentado durante el desarrollo de esta tesis es un ejemplo claro y palpable que se tiene en diferentes compañías, donde muchas veces no se toman riesgos porque pueden ocasionar retrasos o pérdidas en los procesos que se han desarrollado en anteriores proyectos. En varias ocasiones no se toman riesgos porque no existen los argumentos necesarios para llevarlos a cabo, es decir, no hay una investigación de por medio donde se analicen las ventajas y desventajas de desarrollar un nuevo proceso o de cambiar a un nuevo esquema. En este proyecto se realizó todo un análisis de datos y se documentaron los procesos realizados con el fin de reducir los costos en una plataforma de validación Intel®. La metodología seguida durante todo el desarrollo del proyecto va de acuerdo a la estructura de solución de problemas aplicada en Intel, por lo tanto se puede decir que funciona bien sabiéndola aplicar. Muchas veces no es necesario “reinventar la rueda” para poder crear algo benéfico para una organización, mas bien se pudiera decir que “en base a la rueda que ya fue creada, debemos de analizar pequeños detalles para hacerla mas optima y eficiente donde se vaya a usar”. La motivación que dio fuerza a este proyecto fue encontrar los detalles necesarios para optimizar un proceso en el cual se redujeron los costos de fabricación y se optimizaron los códigos de los dispositivos programables en las tarjetas de validación Intel®, dando como resultado una mejora en calidad y costo.

La retroalimentación recibida por los clientes que usaron las tarjetas de validación fue que encontraron las tarjetas muy estables, confiables y fáciles de configurar. La documentación presentada (External Design Specification) resulto bastante clara y concisa. Esta plataforma resulto mas estable que una generación anterior gracias al esfuerzo de todo el equipo en hacer los estudios de planos de poder, placement y a la expansibilidad de capacidades de los dispositivos programables implementada.

El cambio de la familia de PLDs trajo beneficios de reducción de costos e incremento de capacidades para implementar nuevos requerimientos recibidos por parte de los grupos de validación, además que los códigos fueron optimizados y escritos en Verilog.

La planeación del proyecto fue muy buena y los entregables siempre estuvieron a tiempo, lo cual habla muy bien del grupo de diseño.

El prepower-on de la tarjeta se desarrollo simultáneamente en Folsom CA y Guadalajara, donde en ambos sites de Intel se montaron las estaciones de trabajo para hacer este trabajo. Se verifico que los códigos de los dispositivos programables estuviera funcionando correctamente, se realizaron algunos ajustes (tunning) de retardos y se habilitaron algunos pull-ups internos en los PLDs de señales requeridas. Se probaron los nuevos fixtures de validación y se recibió buena retroalimentación por parte de los clientes que solicitaron estos nuevos requerimientos. Gracias a estas nuevas implementaciones los grupos de PV y SV lograron reducir el tiempo de validación del Chipset en un 20%.

Algunos problemas detectados durante el power-on de la tarjeta fueron que el encendido de planos de poder y las secuencias de reset requirieron ser ajustadas, esto fue debido a algunas discrepancias en los requerimientos recibidos comparados con la especificación del Chipset; pero gracias a esto se detectaron y se corrigieron algunas descripciones erróneas en la documentación. Para futuros proyectos se recomienda revisar y entender a más detalle la especificación del Chipset y verificar la información con los arquitectos del Silicio.

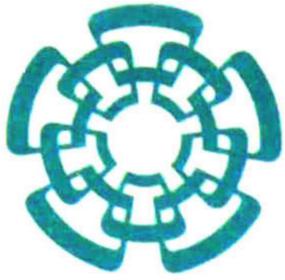
Para seguir reduciendo costos en las tarjetas de validación y como trabajo a futuro se propone hacer un estudio detallado sobre procesos de fabricación del PCB, evaluar e investigar otro tipo de materiales, realizar optimizaciones en el BOM, investigar mas a detalle sobre los auto ruteadores (algoritmos y confiabilidad) para el desarrollo del Layout, etc.

Como comentario final de este trabajo se invita a todos los ingenieros trabajando en la industria privada y en la investigación a desarrollar e innovar la tecnología. En estos tiempos de crisis financiera, no nos queda más que desarrollar productos con valor agregado y que los proyectos se queden aquí en México. Las empresas privadas están viendo en México una opción para desarrollar tecnología, así que desarrollémosla. Seamos capaces de traer beneficios extra a la empresa con la finalidad de traer más proyectos y podamos generar más oportunidades de empleo para la sociedad.

## Referencias

---

- [1] **Intel® Platform Technologies – Changing the FACE of Business.** Disponible en Internet: <<http://software.intel.com/sites/oss/platformtechnologies/>>
- [2] **Intel® Validation training portal, “Internal Intranet training portal”** Disponible solo para uso interno.
- [3] **Intel® Pre-Silicon EV and Validation Hardware, “Platform Design Training”** Disponible solo para uso interno.
- [4] **Intel® PEVH, Validation Hardware Guadalajara, “Internal VHG Intranet portal”** Disponible solo para uso interno.
- [5] **ALTERA homepage.** Disponible en Internet: <<http://www.altera.com/>>
- [6] **XILINX homepage.** Disponible en Internet: <<http://www.xilinx.com/>>
- [7] **J. Vazquez, “Hooks FUB External Design Specification (EDS),”** Disponible solo para uso interno, Rev1.0 publicada en Julio, 2007.
- [8] **System Management Bus (SMBus) Specification, version 2.** Disponible en Internet: <<http://smbus.org/specs/>>
- [9] **Serialized IRQ Support for PCI Systems revision 6.0.** Disponible en Internet, 1 Septiembre, 1995: <<http://www.smsc.com/main/tools/papers/serirq60.doc>>
- [10] **J. Vazquez, L. Urzua, “Reset FUB External Design Specification (EDS),”** Disponible solo para uso interno, Rev1.0 publicada en Septiembre de 2007
- [11] **J. Vazquez, “Tulum Platform ICH10 Power-on Test Plan,”** Disponible solo para uso interno, Rev1.0 publicada en Octubre de 2007



**CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS AVANZADOS DEL I.P.N.  
UNIDAD GUADALAJARA**

El Jurado designado por la Unidad Guadalajara del Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional aprobó la tesis

Optimización de códigos y dispositivos programables para las tarjetas de validación de Intel

del (la) C.

Juan Pablo VÁZQUEZ BARRAGÁN

el día 01 de Abril de 2009.

Dr. José Luis Naredo Villagrán  
Investigador CINVESTAV 3C  
CINVESTAV Unidad Guadalajara

Dr. Ramón Parra Michel  
Investigador CINVESTAV 2B  
CINVESTAV Unidad Guadalajara

Dr. Mario Angel Siller González  
Pico  
Investigador CINVESTAV 2A  
CINVESTAV Unidad Guadalajara



CINVESTAV  
BIBLIOTECA CENTRAL



SSIT000009056