



Plataforma de desarrollo tecnológico de sistemas digitales basada en FPGA

Derechos de autor:

Gerardo Ornelas Vargas

Todos los derechos reservados.

Prohibida su reproducción total o parcial,
sin consentimiento expreso del autor

México MMXVII



UNIVERSIDAD AUTÓNOMA DE ZACATECAS
"FRANCISCO GARCÍA SALINAS"



UNIDAD ACADÉMICA DE INGENIERÍA ELÉCTRICA
DOCTORADO EN INGENIERÍA Y TECNOLOGÍA APLICADA

Plataforma de desarrollo tecnológico de sistemas digitales basada en FPGA

TESIS

Que como parte de los requisitos
parciales para obtener el grado de

Doctor en Ingeniería y Tecnología Aplicada

presenta

Gerardo Ornelas Vargas

Directores de tesis:
Dr. Rodrigo Castañeda Miranda
Dr. José Manuel Ortiz Rodríguez
Dr. Luis Octavio Solís Sánchez
Dr. Héctor Ramírez Díaz

Zacatecas, Zacatecas, México - Enero de 2018



Plataforma de desarrollo tecnológico de sistemas digitales basada en FPGA



Aprobada por el jurado:

Dr. Rodrigo Castañeda Miranda

Dr. Luis Octavio Solís Sánchez

Dr. José Manuel Ortiz Rodríguez

Dr. Héctor René Vega Carrillo

Dr. Alfredo Lara Herrera

Dr. Héctor Ramírez Díaz

Agradecimientos

A Nathalia, Said y Kathya, a mis padres, a Rodrigo, Héctor, Juanjo, Joshua, a Felipe y Sergio, y al CONACyT por el apoyo a los proyectos 235780, 271878 y 282357 que permitieron adquirir y poner en marcha los equipos con los cuales se pudo realizar la parte práctica de esta tesis.

Resumen

El presente trabajo es un proyecto de investigación, desarrollo tecnológico e innovación donde se muestra el diseño, desarrollo, construcción e implementación de una plataforma de desarrollo tecnológico de sistemas digitales basada en FPGA. Este trabajo se desarrolló en conjunto con la empresa INSTECH S.A.P.I. de C.V. y la plataforma fue sometida ante el Instituto Mexicano de la Propiedad Industrial (IMPI) para su protección industrial bajo la figura de modelo de utilidad. En la presente tesis se explica a detalle la estrategia de diseño de dicha plataforma y sus diferencias con las otras opciones que existen actualmente en el mercado, destacando su enfoque hacia un mercado regional que busca adaptabilidad por sobre especialización. El Laboratorio Nacional CONACYT en Sistemas Embebidos, Diseño Electrónico Avanzado y Microsistemas (SEDEAM), que forma parte del Centro de Investigación e Innovación Tecnológica Industrial (CIITI) en la Unidad Académica de Ingeniería Eléctrica (UAIE) de la Universidad Autónoma de Zacatecas, facilitó el desarrollo de esta tesis con la adquisición, instalación y puesta en marcha de su línea de desarrollo de prototipos de PCBs. Además del diseño e implementación de la plataforma, se presenta una aplicación enfocada en radiografía digital probada inicialmente en la plataforma y portada posteriormente hacia un diseño de aplicación específico. La solución de radiografía fue sometida ante el IMPI para su protección industrial bajo la figura de patente.

Palabras clave: FPGA, Sistemas Digitales, Desarrollo de Prototipos, Reutilización de Código, Diseño Electrónico Avanzado, Radiografía Digital

Abstract

This work is a research, technological development and innovation project where design, development, building and implementation of a FPGA-based digital systems technological development platform are shown. This work was developed along with the INSTECH S.A.P.I. de C.V. company, the platform was submitted for industrial protection at IMPI under the little patent category. This work is focused in the platform design strategy for making it outstand among the many options currently available in the market and it targets a regional market seeking adaptability rather than specialization. CONACYT's National Laboratory in Embedded Systems, Advanced Electronic Design, and Microsystems (SEDEAM), which is part of the Industrial Technological Research and Innovation Center (CIITI) at Electric Engineering Academic Unit (UAIE) from Universidad Autónoma de Zacatecas, enabled this thesis' development with the acquisition, installation and start-up of its PCBs prototype assembly line. Besides the design and implementation of the platform, an application focused in digital radiography tested initially in the platform and then ported towards an application specific design. The radiography solution was submitted for industrial protection to the Mexican Institute for Intellectual Protection under the patent category.

Keywords: FPGA, Digital Systems, Prototyping, Code Reuse, Advanced Electronic Design, Digital Radiography

Índice

Agradecimientos.....	ii
Resumen	iii
Abstract.....	iv
Índice	v
Índice de figuras	vii
Capítulo 1: Introducción.....	1
1.1 Evolución de los sistemas digitales.....	3
1.2 Los principales tipos de implementación de solución digital.	4
Microprocesador	5
Lógica programable	7
PAL	9
PLA	10
GAL	10
CPLD	11
FPGA	12
ASIC.....	12
1.3 El impacto de los FPGA en el mundo.....	16
El caso particular de México.....	17
Antecedentes	17
Objetivo general.....	25
Capítulo 2: Marco Teórico	28
2.1 Diseño de sistemas digitales modernos.....	28
FPGA	28
Lenguajes Descriptivos de Hardware	31
Diseño Electrónico Avanzado.....	33
Radiografía digital.....	35
Capítulo 3: Materiales y Métodos	39
3.1 Diseño e implementación de tarjeta de desarrollo tecnológico de sistemas digitales basada en FPGA para la plataforma.	39
3.2 Prototipo de plataforma para radiografía digital	41
Capítulo 4: Resultados y discusión	45
4.1 Tarjeta de circuito impreso de la plataforma para desarrollo de prototipos.....	45

4.2 Aplicación: Radiografía Digital Dental	52
Capítulo 5: Conclusiones.....	62
5.1 Prospectiva de la investigación	63
Referencias	64
Anexos	73
Nomenclaturas	73
Códigos VHDL	74
Propiedad Intelectual Registrada	85

Índice de figuras

Figura 1. Ley de Moore contrastada con microprocesadores reales. Por W. G. Simon – Trabajo propio, CC BY-SA 3.0. https://commons.wikimedia.org/w/index.php?curid=15193542	2
Figura 2. La arquitectura de procesador von Neumann. De David Strigoï - Trabajo propio, dominio público. https://commons.wikimedia.org/w/index.php?curid=7924651	4
Figura 3. Tarjeta de desarrollo Mojo v3 de Spark Fun.....	19
Figura 4. Tarjeta de desarrollo Zybo, de Digilent.....	19
Figura 5. Kit de desarrollo de 100 Gb Stratix V GX de Altera.....	20
Figura 6. Tarjeta de desarrollo Z-turn Board de MyiR.....	20
Figura 7. Bloque de elemento lógico básico de un FPGA con LUT de 4 entradas.....	29
Figura 8. Estructura de la LUT de 4 entradas para un FPGA basado en SRAM.....	29
Figura 9. Concepto general de la tarjeta propuesta: 1) FPGA; 2) electrónica de alimentación; 3) memoria flash de configuración; 4) elementos de interfaz; y 5) variedad de conectores.....	41
Figura 10. Flujos de operación del sensor S10385.....	42
Figura 11. Diagrama de tiempos general del sensor S10385.....	43
Figura 12. Diagrama de tiempos para lectura de medición de radiación (en 10 bits).....	43
Figura 13. Diagrama de tiempos de transmisión de la imagen obtenida por el sensor.....	43
Figura 14. Presentación de las pistas interiores de la tarjeta.....	46
Figura 15. Presentación 3D de la cara superior de la tarjeta.....	46
Figura 16. Presentación 3D de la cara inferior de la tarjeta.....	47
Figura 17. Esquemático general de las conexiones del FPGA con los conectores, alimentación, periféricos en la tarjeta y la memoria flash.....	47
Figura 18. Esquemático detallado de la conexión del FPGA al display 4x7.....	48
Figura 19. Esquemático detallado de la conexión del FPGA a los LEDs.....	48
Figura 20. Esquemático detallado de la conexión del FPGA al display 4x7.....	49
Figura 21. Esquemático detallado de las conexiones del FPGA.....	49
Figura 22. Esquemático detallado de la conexión del FPGA a los interruptores.....	50
Figura 23. Esquemático detallado de la conexión del FPGA a los botones.....	50
Figura 24. Esquemático detallado de la sección de alimentación.....	51
Figura 25. Costo (eje y, en dólares) del cm de PCB respecto al número de capas (eje x).....	51
Figura 26. Diagrama a bloques del sistema de Radiografía Digital Dental embebido en el FPGA.....	52
Figura 27. Captura de emisión de rayos X por parte del sensor, exposición de 300 ms.....	53
Figura 28. Captura de emisión de rayos X por parte del sensor, exposición de 500 ms.....	53
Figura 29. Imagen sin procesar, 500 ms de exposición.....	54
Figura 30. Imagen postprocesada, 500 ms de exposición.....	54
Figura 31. Imagen sin procesar, 300 ms de exposición.....	55
Figura 32. Imagen postprocesada, 300 ms de exposición.....	55
Figura 33. Cara superior de PCB para radiografía digital.....	57
Figura 34. Cara inferior de PCB de radiografía digital.....	57
Figura 36. Diagrama esquemático general del PCB para radiografía digital.....	58
Figura 37. Esquemático de conexión de banco de memorias para procesamiento en tarjeta.....	59
Figura 38. Esquemático conexión USB.....	59
Figura 39. Esquemático de conexiones seriales a cañón de rayos X y auxiliar/depuración.....	60
Figura 40. Esquemático de conexiones del FPGA.....	61
Figura 41. Tarjeta Prototipo Beta del sistema de radiografía digital.....	61

Capítulo 1: Introducción

La electrónica en general y los sistemas digitales en particular son una rama de investigación y desarrollo relativamente nueva, pero con un alcance y crecimiento nunca antes visto por la humanidad en toda su historia. Este crecimiento exponencial ha llevado a la electrónica a un estado de omnipresencia (Abowd, 2016) en prácticamente todos los nuevos desarrollos de las distintas disciplinas del conocimiento humano (Williams, 2017).

En 1965, Gordon Moore, cofundador de Fairchild Semiconductors e Intel, publicó su artículo “Cramming more components onto integrated circuits”. Este artículo dio paso a lo que se conoce como la Ley de Moore, que predice que la cantidad de transistores que se pueden integrar en un área dada se duplicarán cada dos años, junto con una reducción en el costo y consumo de potencia de cada transistor (Moore, 1965). Dicha ley se ha cumplido de manera aproximada por más de 50 años, aunque en los últimos tiempos las limitaciones físicas de las técnicas de manufactura y de los propios materiales han reducido este aumento a 2^{25} en la cantidad de transistores por chip (Morris, 2012; Morris, 2014a; y Morris, 2015).

Si realizamos una analogía del avance que han experimentado los sistemas digitales con la industria del transporte, sería como haber desarrollado un transporte que se mueva a la velocidad de la luz, con combustible mínimo y a un costo ínfimo (Morris, 2014b). La Ley de Moore habla de transistores en un chip de manera general. Sin embargo, la complejidad del diseño en la parte analógica de los transistores ha generado una división en dos tipos de transistores: para aplicaciones digitales y para aplicaciones analógicas. En las aplicaciones digitales, los transistores trabajan en la región de saturación y de no conducción, para generar señales lógicas de 0 y 1. En las aplicaciones analógicas se trabaja en la región conductiva del transistor para adecuar señales analógicas, es decir, todo lo que está entre el 0 y 1 del circuito digital, y generar circuitos como amplificadores o de RF (Sandige y Sandige, 2012). Debido a esta diferencia, la Ley de Moore ha impactado de manera más notoria en los diseños de circuitos digitales (Kumar et al., 2014).

La figura 1 muestra la cantidad de transistores en un microprocesador en el periodo 1971-2011 contrastado con la predicción de la Ley de Moore.

Microprocessor Transistor Counts 1971-2011 & Moore's Law

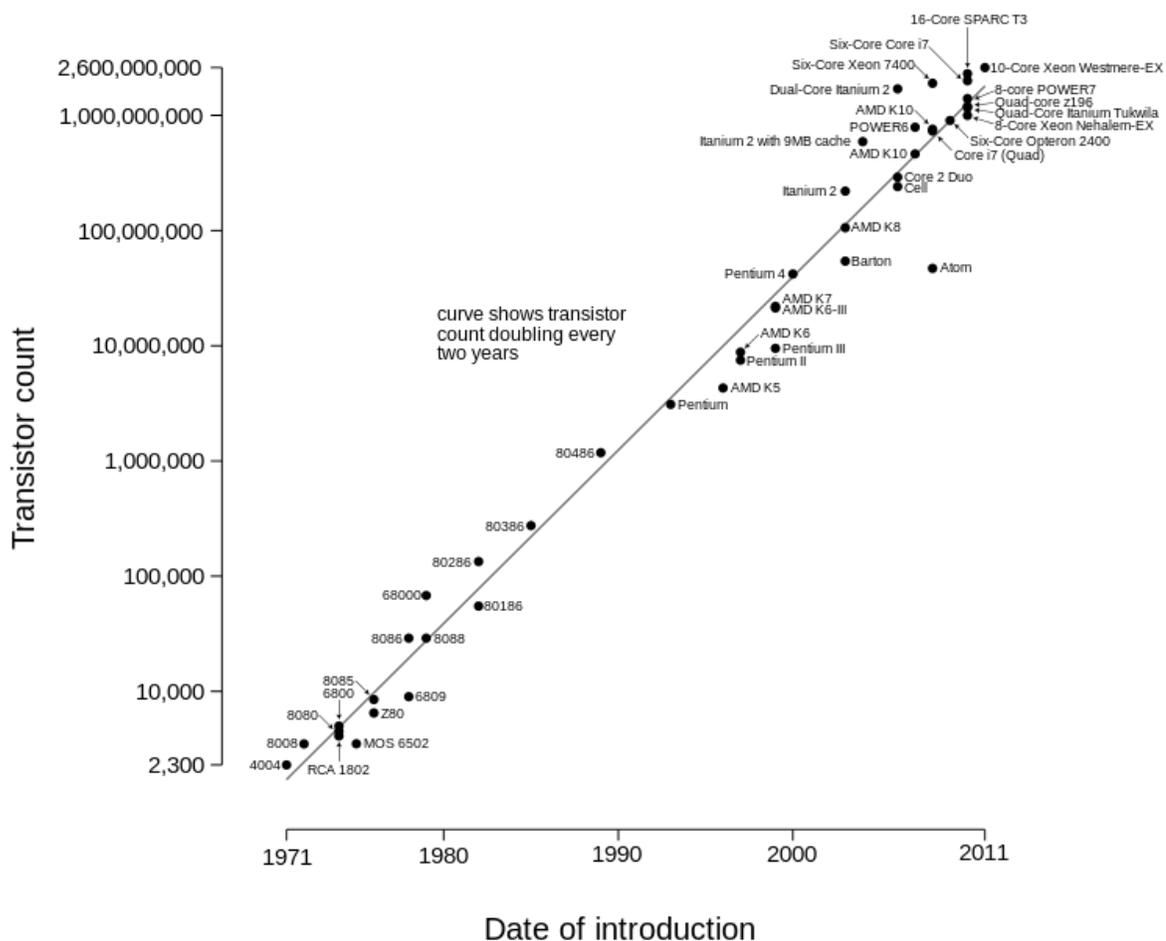


Figura 1. Ley de Moore contrastada con microprocesadores reales. Por W. G. Simon – Trabajo propio, CC BY-SA 3.0. <https://commons.wikimedia.org/w/index.php?curid=15193542>.

Dentro de los avances tangibles gracias al avance de la electrónica y los sistemas digitales tenemos:

- Internet
- Teléfonos inteligentes
- Dispositivos vestibles
- Computadoras de escritorio y portátiles
- Sistemas de navegación
- Sistemas de control digital
- Sistemas embebidos

Estos avances tecnológicos son los que puede aprovechar la industria nacional, tanto para impactar con innovación en sus procesos como para generar nuevas ofertas de valor (CMM, 2010).

1.1 Evolución de los sistemas digitales

El álgebra booleana

Si bien puede decirse que la electrónica inició con el desarrollo del diodo de cristal por Karl Ferdinand Braun en 1874 (Braun, 1874), los sistemas digitales vieron el nacimiento de su base de desarrollo en 1847 con la publicación del libro *The Mathematical Analysis of Logic* y el posterior *An Investigation of the Laws of Thought* en 1854 por George Boole. El álgebra booleana es una rama del álgebra donde los valores son verdadero o falso (1 y 0 respectivamente) y las principales operaciones son conjunción (y), disyuntiva (o) y negación (no). En el siglo XIX fue perfeccionada quedando demostrado que el álgebra booleana es isomórfica a un campo de conjuntos. La naturaleza binaria del álgebra booleana permitió utilizarla como herramienta para el análisis y diseño de los sistemas digitales, que también son binarios. Los primeros circuitos digitales fueron precisamente las operaciones del álgebra booleana y-o-no.

La arquitectura Von Neumann

Las primeras máquinas de cómputo tenían programas fijos: la tarea para la que eran programadas era lo único que podían ejecutar. Para cambiar la programación una vez que la máquina era diseñada y ensamblada era necesario un recableado o hasta un rediseño de la máquina. Este trabajo no resultaba sencillo, ya que al esfuerzo de diseño se sumaba un componente de esfuerzo físico. John von Neumann propuso en 1945 la arquitectura Von Neumann (figura 2) que habilitó la programación de las operaciones de la máquina mediante códigos binarios (Nurmi, 2007).

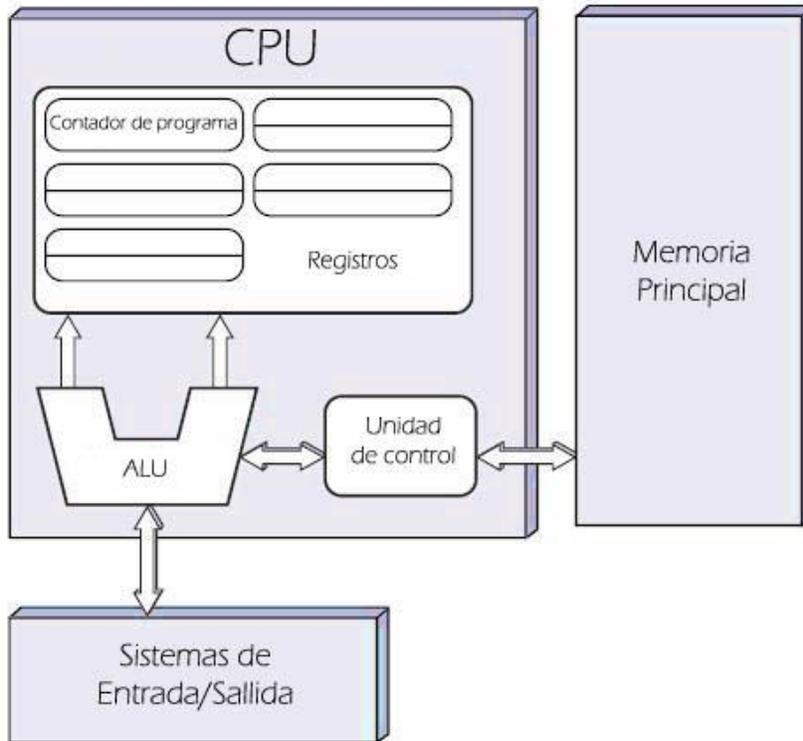


Figura 2. La arquitectura de procesador von Neumann. De David Strigoj - Trabajo propio, dominio público. <https://commons.wikimedia.org/w/index.php?curid=7924651>

1.2 Los principales tipos de implementación de solución digital.

¿Cómo podemos implementar un sistema digital en nuestros días? La respuesta a esta pregunta tiene varias respuestas; posiblemente en un futuro se podrá implementar mediante la computación cuántica y la computación biológica, pero al día de hoy dichas tecnologías se encuentran en una etapa temprana de desarrollo. Si nos limitamos a la tecnología disponible y con madurez en su proceso de desarrollo, existen tres caminos viables: microprocesador, lógica programable y circuitos integrados de aplicación específica (ASIC), cada uno con sus ventajas y sus limitaciones (Maxfield, 2004; Morris y Ciletti, 2007; Sandige y Sandige, 2012; Kumar et al., 2014). A continuación, se verán a detalle estas tres formas de implementación.

Microprocesador

El microprocesador surgió de la evolución de distintas tecnologías predecesoras, básicamente de la computación y de la tecnología de semiconductores. El inicio de esta última data de mitad de la década de 1950; estas tecnologías se fusionaron a principios de los años 1970, produciendo el primer microprocesador. Dichas tecnologías iniciaron su desarrollo a partir de la Segunda Guerra Mundial; en ese tiempo los científicos desarrollaban computadoras específicas para aplicaciones militares. En la posguerra, a mediados de la década de 1940, la computación digital emprendió un fuerte crecimiento también con propósitos científicos y civiles. La tecnología electrónica avanzó y los científicos hicieron grandes progresos en el diseño de componentes de estado sólido (semiconductores). En 1948, los laboratorios Bell crearon el transistor (Maxfield, 2004).

En los primeros años de la década de 1950 aparecieron las primeras computadoras digitales de propósito general. Se fabricaron utilizando tubos al vacío o bulbos como componentes electrónicos activos. Los módulos de tubos al vacío componían circuitos lógicos básicos, tales como compuertas y flip-flops. Mediante el ensamble de diferentes módulos (lógica de control, circuitos de memoria, etc.) se construyó la primera computadora electrónica. Los tubos de vacío también formaron parte de la construcción de máquinas para la comunicación con las computadoras (Maxfield, 2004).

Si para la construcción de un circuito sumador simple se requieren algunas compuertas lógicas, la construcción de una computadora digital precisa de numerosos circuitos o dispositivos electrónicos. Un paso trascendental en el diseño de la computadora fue lograr que los datos fueran almacenados en una memoria. Y la idea de almacenar programas en memoria para luego ejecutarlo también fue de fundamental importancia (Nurmi, 2007).

La tecnología de los circuitos de estado sólido evolucionó en la década de 1950. El empleo del silicio (Si), de bajo costo y con métodos de producción masiva, hicieron del transistor el componente más usado para el diseño de circuitos electrónicos. Gracias a estas ventajas, a finales de la década de 1950 se reemplazó el tubo al vacío por el transistor en el diseño de la computadora digital (Kumar et al., 2014).

A principios de la década de 1960, el estado de arte en la construcción de computadoras de estado sólido avanzó de manera notable; surgieron las tecnologías en circuitos digitales como: RTL (*Resistor*

Transistor Logic, lógica transistor resistor), DTL (*Diode Transistor Logic*, lógica transistor diodo), TTL (*Transistor Transistor Logic*, lógica transistor transistor), ECL (*Emitter Complementary Logic*, lógica complementada emisor) (Sandige y Sandige, 2012; Kumar et al., 2014).

A mediados de los años 1960 se producen las familias de circuitos de lógica digital, dispositivos integrados en escala SSI (*Small Scale Integration*) y MSI (*Medium Scale Integration*), que corresponden a baja y mediana escala de integración de componentes. A finales de los años 1960 y principios de los 70 surgieron los sistemas a alta escala de integración o LSI (*Large Scale Integration*). La tecnología LSI fue haciendo posible incrementar la cantidad de componentes en los circuitos integrados (CI). Sin embargo, pocos circuitos LSI fueron producidos, entre los que destacan los dispositivos de memoria (Maxfield, 2004; Morris y Ciletti, 2007; Sandige y Sandige, 2012; Kumar et al., 2014).

Las primeras calculadoras electrónicas requerían entre 75 y 100 circuitos integrados. Después se dio un paso importante en la reducción de la arquitectura de la computadora a un circuito integrado simple, que fue llamado *microprocesador*, unión de las palabras «*micro*» del griego μικρο-, «pequeño», y procesador (Maxfield, 2004). A continuación, se muestra un breve recorrido por la historia de los microprocesadores.

- El primer microprocesador fue el Intel 4004, producido en 1971. Se desarrolló originalmente para una calculadora y resultó revolucionario para su época. Contenía 2300 transistores, era un microprocesador de arquitectura de 4 bits que podía realizar hasta 60,000 operaciones por segundo trabajando a una frecuencia de reloj de alrededor de 700 kHz (Maxfield, 2004).
- El primer microprocesador de 8 bits fue el Intel 8008, desarrollado a mediados de 1972 para su uso en terminales informáticos. El Intel 8008 integraba 3,300 transistores y podía procesar a frecuencias máximas de 800 kHz (Maxfield, 2004).
- El primer microprocesador realmente diseñado para uso general, desarrollado en 1974, fue el Intel 8080 de 8 bits, que contenía 4500 transistores y podía ejecutar 200,000 instrucciones por segundo trabajando a alrededor de 2 MHz (Maxfield, 2004).
- El primer microprocesador de 16 bits fue el 8086, seguido del 8088. El 8086 fue el primer miembro de la popular arquitectura x86, actualmente usada en la mayoría de los computadores. El chip 8086 fue introducido al mercado en el verano de 1978, pero debido a que no había aplicaciones en el mercado que funcionaran con 16 bits, Intel sacó al mercado el 8088, que fue lanzado en 1979. Estos llegaron a operar a frecuencias mayores de 4 MHz (Maxfield, 2004).

- El Intel 80286 (también conocido simplemente como 286) fue el microprocesador elegido para equipar al IBM Personal Computer/AT, y por lo tanto el más empleado en los PC-AT compatibles entre mediados y finales de los años 1980. Lanzado al mercado en 1982, es un microprocesador de 16 bits, de la familia x86rrrrrrrrr. Contaba con 134,000 transistores. Las versiones finales alcanzaron velocidades de hasta 25 MHz (Maxfield, 2004; Nurmi, 2007).
- Uno de los primeros procesadores de arquitectura de 32 bits fue el 80386 de Intel, fabricado a mediados y fines de la década de 1980; en sus diferentes versiones llegó a trabajar a frecuencias del orden de los 40 MHz (Maxfield, 2004; Nurmi, 2007)
- El microprocesador DEC Alpha se lanzó al mercado en 1992, corriendo a 200 MHz en su primera versión, en tanto que el Intel Pentium surgió en 1993 con una frecuencia de trabajo de 66 MHz. El procesador Alpha, de tecnología RISC y arquitectura de 64 bits, marcó un hito, declarándose como el más rápido del mundo en su época. Llegó a 1 GHz de frecuencia hacia el año 2001 (Nurmi, 2007).
- Los microprocesadores modernos tienen una capacidad y velocidad mucho mayores, trabajan en arquitecturas de 64 bits, integran más de 700 millones de transistores, como en el caso de la serie Core i7 de Intel, y pueden operar a frecuencias normales algo superiores a los 3 GHz (Nurmi, 2007).

Lógica programable

En electrónica y computación, la lógica programable es un tipo de diseño implementado en chips que permite la reconfiguración de los circuitos con el simple cambio de la configuración que incorpora, a diferencia de la lógica cableada que requiere una reconfiguración física (Morris y Ciletti, 2007).

La lógica programada se basa en dispositivos lógicos programables (PLD, *Programmable Logic Device*), los cuales tienen una función no establecida, al contrario de las compuertas lógicas que tienen una función fija en el momento de su fabricación. Antes de poder utilizar el PLD en un circuito, este debe ser configurado (Maxfield, 2004; Xilinx, 2008).

Antes de que se inventaran los PLD, los chips de memoria de solo lectura (ROM, *Read-Only Memory*) se utilizaban para crear funciones de lógica combinacional arbitrarias con un número determinado de

entradas, considerando una ROM con m entradas, a las que se denomina *líneas de dirección*; y con n salidas, a las que se denomina *líneas de datos*. Cuando se utiliza como memoria, la ROM contiene palabras de n bits (Maxfield, 2004).

La ventaja de utilizar una ROM de esta forma es que cualquier función concebible de las m entradas puede ser colocada por las n salidas, haciendo de este el dispositivo lógico combinacional de mayor propósito general disponible. También las PROM (ROM programables), EPROM (PROM de borrado por ultravioleta) y EEPROM (PROM de borrado eléctrico) disponibles pueden ser programadas de esta manera con un programador PROM de hardware o software (Maxfield, 2004). Sin embargo, existen varias desventajas:

- Son bastante más lentas que los circuitos lógicos dedicados.
- No necesariamente proveen de protección ante transiciones lógicas asíncronas.
- Consumen mayor potencia.
- Sólo se utiliza una pequeña fracción de su capacidad en una sola aplicación, lo que representa un uso ineficiente energético y del espacio.
- Por sí solas no pueden ser utilizadas para circuitos de lógica secuencial, puesto que no contienen elementos de memoria como flip flops. Para realizar algunos circuitos secuenciales (como máquinas de estado) se utilizaba un registro TTL externo.

En 1970, Texas Instruments desarrolló un CI de máscara programable basado en la memoria asociativa de sólo lectura (ROAM) de IBM. Este dispositivo, el TMS2000, era programado alterando la capa metálica durante la producción del CI. El TMS2000 tenía hasta 17 entradas y 18 salidas con 8 flip flops JK como memoria. Texas Instruments acuñó el término *Programmable Logic Array* (PLA) para este dispositivo (Maxfield, 2004; Xilinx, 2008).

En 1973 National Semiconductor introdujo un dispositivo PLA de máscara programable (DM7575) con 14 entradas y 8 salidas sin registros de memoria. Este era más popular que el de Texas Instruments, pero el coste de fabricar la máscara metálica limitaba su uso. El dispositivo es significativo por ser la base del FPGA producido por Signetics en 1975, el 82S100 (Maxfield, 2004).

En 1971, General Electric desarrollaba un PLD (*Programmable Logic Device*) basado en la nueva tecnología PROM. Este dispositivo experimental mejoró el ROAM de IBM permitiéndole realizar lógica

multinivel. Intel acababa de introducir la PROM de puerta flotante borrable por UV, tecnología que fue incorporada por los desarrolladores de General Electric. El dispositivo de General Electric fue el primer PLD jamás desarrollado, antecesor del EPLD de Altera de una década después. General Electric obtuvo varias patentes tempranas en PLDs (Maxfield, 2004).

En 1974, General Electric firmó un acuerdo con Monolithic Memories para desarrollar un PLD de máscara programable incorporando las innovaciones de General Electric. El dispositivo se bautizó como *Programmable Associative Logic Array* (PALA, matriz lógica asociativa programable). El MMI 5760 fue terminado en 1976 y podía implementar circuitos multinivel o secuenciales de más de 100 compuertas. El dispositivo estaba soportado por el entorno de desarrollo de General Electric, donde las ecuaciones booleanas podían ser convertidas a patrones de máscara para configurar el dispositivo. El integrado nunca se comercializó (Maxfield, 2004; Morris y Ciletti, 2007; Xilinx, 2008).

PAL

MMI introdujo un dispositivo revolucionario en 1978, la *Programmable Array Logic* (PAL, matriz lógica programable). La arquitectura era más sencilla que la *Field-Programmable Logic Array* (FPLA) de Signetics porque omitía la matriz OR programable. Esto hizo los dispositivos más rápidos, más pequeños y más baratos. Estaban disponibles en encapsulados de 20 pines y DIP de 300 milésimas de pulgada, mientras que las FPLAs venían en encapsulados de 28 pines y DIP de 600 milésimas de pulgada. Ciertas publicaciones sobre PALs desmitificaban el proceso de diseño. El software de diseño PALASM (*PAL Assembler*, ensamblador PAL) convertía las ecuaciones booleanas de los ingenieros en el patrón de fusibles requerido para programar el dispositivo. Los PAL de MMI pronto fueron distribuidos por National Semiconductor, Texas Instruments y AMD (Maxfield, 2004; Morris y Ciletti, 2007; Xilinx, 2008).

Tras el éxito de MMI con los PAL de 20 pines, AMD introdujo los 22V10 de 24 pines con características adicionales. Tras comprar a MMI (1987), AMD desarrolló una operación consolidada como Vantis, adquirida por Lattice Semiconductor en 1999 (Maxfield, 2004; Morris y Ciletti, 2007).

PLA

Definido en inglés como *Programmable Logic Array*, forma parte de los PLD simples (SPLDs). Estos dispositivos fueron los primeros chips desarrollados específicamente para implementar circuitos lógicos. Como los PAL anteriormente vistos, disponen de dos planos diferenciados: AND y OR. En este caso, los PLA tienen ambos planos programables, por lo que su estructura es ideal para implementar funciones lógicas como sumas de productos, aunque presenta la desventaja de que el dispositivo es de mayor tamaño y menor velocidad (Maxfield, 2004; Morris y Ciletti, 2007). Por último, existen variantes de este tipo de estructura:

- Sólo plano AND o plano OR
- Plano AND con reglamentación
- Plano AND y OR con reglamentación
- Con registros o registros programables
- Factores fundamentales de revisión
- Aplicación de álgebra de Boole.

GAL

Una innovación del PAL fue la matriz lógica genérica (*Generic Array Logic*) o GAL. Ambas fueron desarrolladas por Lattice Semiconductor en 1985. Este dispositivo tiene las mismas propiedades lógicas que el PAL, pero puede ser borrado y reprogramado. La GAL es muy útil en la fase de prototipado de un diseño, cuando un fallo en la lógica puede ser corregido por reprogramación. Las GALs se programan y reprograman utilizando un programador OPAL, o utilizando la técnica de programación circuital en chips secundarios (Maxfield, 2004; Lattice Semiconductor Corporation, 2010).

Un dispositivo similar llamado PEEL (*Programmable Electrically Erasable Logic*, lógica programable eléctricamente borrrable) fue introducido por International CMOS.

Una GAL permite implementar cualquier expresión en suma de productos con un número de variables definidas. El proceso de programación consiste en activar o desactivar cada celda E2CMOS con el objetivo de aplicar la combinación adecuada de variables a cada compuerta AND y obtener la suma de productos (Maxfield, 2004; Xilinx, 2008; Lattice Semiconductor Corporation, 2010).

Las celdas E2CMOS activadas conectan las variables deseadas o sus complementos con las entradas adecuadas de las puertas AND. Las celdas E2CMOS están desactivadas cuando una variable o su complemento no se usa en un determinado producto. La salida final de la puerta OR es una suma de productos. Cada fila está conectada a la entrada de una puerta AND, y cada columna a una variable de entrada o a su complemento. Mediante la programación se activa o desactiva cada celda E2CMOS, y se puede aplicar cualquier combinación de variables de entrada, o sus complementos, a una puerta AND para generar cualquier operación producto que se desee. Una celda activada conecta de forma efectiva su correspondiente fila y columna, y una celda desactivada desconecta la fila y la columna. Una celda se puede borrar y reprogramar eléctricamente. Una celda E2CMOS típica puede mantener el estado en que se ha programado durante 20 años o más. Las macroceldas lógicas de salida (OLMCs) están formadas por circuitos lógicos que se pueden programar como lógica combinacional o como lógica secuencial. Las OLMCs proporcionan mucha más flexibilidad que la lógica de salida fija de una PAL (Lattice Semiconductor Corporation, 2010).

CPLD

Las PAL y GAL están disponibles sólo en tamaños pequeños, equivalentes a unos pocos cientos de puertas lógicas. Para circuitos lógicos mayores, se pueden utilizar PLD complejos o CPLD. Estos contienen el equivalente a varias PAL enlazadas por interconexiones programables, todo ello en el mismo circuito integrado. Los CPLD pueden reemplazar miles, o incluso cientos de miles de puertas lógicas (Parnell y Mehta, 2003; Maxfield, 2004; Morris y Ciletti, 2007).

Algunas CPLDs se programan utilizando un programador PAL, pero este método no es manejable para dispositivos con cientos de pines. Un segundo método de programación es soldar el dispositivo en su circuito impreso. Las CPLDs contienen un circuito que descodifica la entrada de datos y configura la CPLD para realizar su función lógica específica (Parnell y Mehta, 2003; Maxfield, 2004; Morris y Ciletti, 2007).

Cada fabricante tiene un nombre propietario para este sistema de programación. Por ejemplo, Lattice Semiconductor la llama *In-System Programming* (programación en el sistema). Sin embargo, estos sistemas propietarios están dejando paso al estándar del *Joint Test Action Group* (JTAG) (Maxfield, 2004).

FPGA

Mientras el desarrollo de las PAL se enfocaba hacia GAL y CPLD, apareció una corriente de desarrollo distinta, que desembocó en un dispositivo basado en la tecnología de matriz de puertas que se denominó *Field-Programmable Gate Array* (FPGA). Algunos ejemplos de los primeros FPGA son la matriz 82S100 y el secuenciador 82S105 de Signetics, presentados a finales de los 70. El 82S100 era una matriz de términos AND, y también tenía funciones de flip flop (Parnell y Mehta, 2003; Maxfield, 2004; Xilinx, 2008).

Los FPGA utilizan una rejilla de puertas lógicas, similar a la de una matriz de puertas ordinarias, pero la programación en este caso la realiza el cliente, no el fabricante. El término *field-programmable* (literalmente “programable en el campo”) se refiere a que la matriz se define fuera de la fábrica, de acuerdo al campo en donde se utiliza (Parnell y Mehta, 2003; Maxfield, 2004; Xilinx, 2008).

Los FPGA se programan normalmente tras ser soldados en la placa, en una forma similar a los CPLD grandes. En los FPGA más grandes, la configuración es volátil y debe ser reescrita cada vez que se enciende o se necesita una funcionalidad diferente. La configuración se guarda normalmente en una PROM o EEPROM. Las versiones EEPROM pueden ser programadas mediante técnicas como el uso de cables JTAG (Parnell y Mehta, 2003; Maxfield, 2004; Xilinx, 2008).

Los FPGA y los CPLD son buenas opciones para una misma tarea. Algunas veces la decisión sobre una u otra es más económica que técnica, o puede depender de la preferencia personal o experiencia del ingeniero (Parnell y Mehta, 2003; Xilinx, 2008).

ASIC

Un Circuito Integrado para Aplicaciones Específicas (*Application Specific Integrated Circuit*, ASIC) es un circuito integrado hecho a la medida para un uso o función específica, en vez de ser concebido para propósitos de uso general. Por ejemplo, un chip diseñado únicamente para ser usado en un teléfono móvil es un ASIC. Por otro lado, los circuitos integrados de la serie 7400 son circuitos lógicos (combinacionales o secuenciales) que se pueden utilizar para una multiplicidad de aplicaciones. En un lugar intermedio entre los ASIC y los productos de propósito general están los Productos Estándar para Aplicaciones Específicas (*Application Specific Standard Product*, ASSP) (Martin, 2000; Maxfield, 2004; Kaeslin, 2008; Salman y Friedman, 2012; Kumar et al., 2014).

Con los avances en la miniaturización y en las herramientas de diseño, la complejidad máxima, y por ende la funcionalidad, de un ASIC ha crecido desde 5,000 puertas lógicas a más de 100 millones. Los ASIC modernos a menudo incluyen procesadores de 32-bit, bloques de memoria RAM, ROM, EEPROM y Flash, así como otros tipos de módulos. Este tipo de ASIC frecuentemente son llamados SoC (*System on-Chip*, sistema en un chip). Los diseñadores de ASIC digitales usan lenguajes descriptivos de hardware (HDL), tales como Verilog o VHDL, para describir la funcionalidad de estos dispositivos (Martin, 2000; Maxfield, 2004; Kaeslin, 2008; Salman y Friedman, 2012; Kumar et al., 2014).

Los primeros ASIC utilizaban tecnología de matriz de puertas. Ferranti fabricó tal vez la primera matriz de puertas, la ULA (*Uncommitted Logic Array*, matriz lógica no fija), alrededor de 1980. El diseño a la medida se realizaba al variar la máscara de interconexión metálica. Las ULA tenían complejidades de hasta algunos miles de puertas. Las versiones posteriores fueron más generalizadas, con moldes base configurados tanto por las capas metálicas como polisilicónicas. Algunos moldes base incluían elementos de RAM (Smith, 1997; Martin, 2000; Kaeslin, 2008; Salman y Friedman, 2012).

A mediados de 1980, un diseñador elegía a un fabricante de ASIC, y luego implementaba el diseño utilizando las herramientas provistas por ese fabricante en particular. A pesar de que existían herramientas de diseño provista por terceros, no había un enlace efectivo entre éstas y los procesos productivos de los fabricantes. Una solución a este problema, que además permitió aumentar la densidad de los ASIC, fue la implementación de Celdas Estándares. Cada fabricante de ASIC creaba bloques funcionales con características eléctricas conocidas, tales como los tiempos de propagación, capacitancias e inductancias, que podían ser representadas en las herramientas desarrolladas por terceros. El diseño basado en Celdas Estándares y el uso de estos bloques funcionales permitió alcanzar densidades de puertas muy altas, y un buen desempeño eléctrico. Este tipo de diseño se ubica entre diseño de Matriz de Puertas, y el diseño hecho totalmente a la medida, en término de los costos fijos y de fabricación de cada unidad (Smith, 1997; Martin, 2000; Salman y Friedman, 2012).

Hacia finales de 1980, estuvieron disponibles las herramientas de síntesis lógica, tales como el *Design Compiler*. Estas herramientas podían compilar descripciones HDL (*Hardware Description Language*, lenguaje de descripción de hardware) en una lista de nodos al nivel de puertas. Esto dio paso a un estilo de diseño llamado Diseño Basado en Celdas Estándares. Este tipo de diseño contempla las siguientes

etapas conceptuales, aunque en la práctica estas etapas se traslapan significativamente (Smith, 1997; Martin, 2000):

1. Un equipo de ingenieros de diseño comienza con la compresión no formal de las funciones requeridas por el ASIC a diseñar, usualmente derivada del análisis de requerimientos.
2. El equipo de diseño construye una descripción del ASIC para alcanzar estos objetivos, utilizando un HDL. Este proceso es similar a escribir un programa computacional en un lenguaje de alto nivel, y usualmente es llamado diseño RTL (*Register Transfer Level*).
3. La validez del diseño es verificada a través de una simulación. Un sistema virtual, implementado a nivel de software puede simular el desempeño de los ASIC a velocidades equivalentes de mil millones de instrucciones por segundo.
4. Una herramienta de síntesis lógica convierte el diseño RTL en un gran conjunto de elementos de bajo nivel, llamados Celdas Estándares. Estos elementos son tomados desde una biblioteca, que consiste en una colección de puertas precaracterizadas (tales como NOR de 2 entradas, NAND de 2 entradas, inversores, etc.). Las celdas estándares usualmente son específicas para el fabricante del ASIC. El conjunto resultante de Celdas Estándares, junto a la interconexión de ellas, es llamado la lista de nodos a nivel de puertas.
5. La lista de nodos es luego procesada por una herramienta de posicionamiento, la cual ubica las Celdas Estándares en una región que representa el ASIC final. Esta ubicación está sujeta a un conjunto de restricciones. En ocasiones se utilizan técnicas avanzadas para optimizar el posicionamiento.
6. La herramienta de ruteo toma la ubicación física de las celdas, y utiliza el listado de nodos para crear las conexiones eléctricas entre ellas. La salida de esta etapa es un conjunto de fotomáscaras, con las que el fabricante producirá los circuitos integrados.
7. Se puede hacer una estimación bastante precisa de los retardos finales, las resistencias y capacitancias parásitas y del consumo de energía. Estas estimaciones son usadas en la ronda final de pruebas. Estas pruebas permiten demostrar si el dispositivo funcionará en los rangos de temperatura y voltaje extremos. Cuando estas pruebas finalizan, la información de las fotomáscaras es entregada para la fabricación del chip.

Estos pasos, llevados a cabo con el nivel de habilidad común en la industria, casi siempre producen un dispositivo final que implementa correctamente el diseño original, a menos que se introduzcan fallas a nivel físico de fabricación (Smith, 1997; Martin, 2000). Estos pasos de diseño son también comunes al

diseño de un producto estándar. La diferencia significativa es que el diseño con Celdas Estándares utiliza la biblioteca de celdas del fabricante, que ha sido utilizada en potencialmente cientos de otros diseños, y por lo tanto constituyen un riesgo mucho menor en comparación con un diseño hecho totalmente a la medida. Las Celdas Estándares producen una densidad de diseño con un costo comparativamente más bajo, y pueden también integrar núcleos IP y SRAM en una forma efectiva, a diferencia de las matrices de puertas (Smith, 1997).

El diseño basado en Matriz de Puertas (*Gate Array*) es un método de manufactura en donde las capas difundidas, es decir, los transistores y otros elementos activos están predefinidos, y las obleas que contienen estos dispositivos se mantienen en *stock* antes de la metalización, es decir, desconectadas. El proceso de diseño físico define la interconexión del dispositivo final. Para la mayoría de los fabricantes de ASIC, este consiste en dos a cinco capas metálicas, cada una perpendicular a la que la precede. Los costos fijos son mucho más bajos, ya que las máscaras litográficas se requieren sólo para las capas metálicas, y los ciclos productivos son mucho más cortos, ya que la metalización es un proceso comparativamente más rápido. Una ventaja adicional para el diseñador es que con este método se pueden conseguir retardos de propagación mínimos, comparado con las soluciones basadas en FPGAs disponibles en el mercado (Smith, 1997; Martin, 2000; Maxfield, 2004).

Los ASIC basados en matriz de puertas requieren siempre de un compromiso, ya que nunca se logra una correspondencia perfecta entre un diseño determinado y las obleas que el fabricante tiene en stock. A menudo las dificultades que aparecen al rutear las interconexiones, requieren migrar a un dispositivo con un arreglo más grande, con el consecuente aumento en el costo del dispositivo. Estos problemas frecuentemente son resultado del software utilizado para desarrollar las interconexiones (Smith, 1997; Martin, 2000).

En la actualidad, los diseños formados solamente por puertas lógicas raramente son implementados con matriz de puertas, y son reemplazados por dispositivos programables, como los FPGA, los cuales pueden ser programadas por el usuario, y mantienen un costo fijo asociado mínimo, un costo por unidad marginalmente superior, y un desempeño comparable. Hoy, las matrices de puertas están evolucionando en ASIC estructurados, que consisten en un gran núcleo IP (*Intellectual Property*), como un procesador, una unidad DSP, periféricos, memorias y bloques lógicos reconfigurables. Este cambio se debe principalmente a que los ASIC son capaces de integrar estos grandes bloques de sistemas funcionales, y

los Sistemas en un Chip (SoC) requieren más que sólo bloques lógicos (Martin, 2000; Kaeslin, 2008; Salman y Friedman, 2012).

El término Matriz de Puertas (*Gate Array*) es casi sinónimo del término “Semi a la medida” (*Semi-Custom*). El término utilizado depende de quién lo utilice; un ingeniero de proceso probablemente utilice el término “Semi a la medida”, mientras que un diseñador a nivel lógico utiliza el término “Matriz de Puertas” (Martin, 2000; Kaeslin, 2008; Salman y Friedman, 2012).

Por otra parte, el diseño hecho totalmente a la medida define la totalidad de las capas litográficas del dispositivo. Este se utiliza tanto para el diseño de ASIC como para el diseño de productos estándares (Martin, 2000; Kaeslin, 2008; Salman y Friedman, 2012).

Los beneficios de este método usualmente incluyen un área reducida (y consecuentemente costos por unidad menores), mejoras en el desempeño y también la habilidad de integrar componentes analógicos y otros componentes pre-diseñados, como son los microprocesadores que forman un SoC (Martin, 2000; Kaeslin, 2008; Salman y Friedman, 2012).

Las desventajas del diseño totalmente a la medida son un costo y tiempo de desarrollo mayores, costos fijos mayores, mayor complejidad del software CAD y la necesidad de habilidades mucho mayores por parte del equipo de diseño (Kaeslin, 2008; Salman y Friedman, 2012).

Sin embargo, para diseños puramente digitales, las librerías de celdas estándares, junto con los sistemas CAD modernos, pueden ofrecer ventajas considerables en términos de costos y desempeño junto a un bajo riesgo. Las herramientas de layout automático son rápidas y fáciles de usar, y ofrecen la posibilidad de optimizar manualmente cualquier aspecto que limite el desempeño del diseño (Kaeslin, 2008; Salman y Friedman, 2012).

1.3 El impacto de los FPGA en el mundo

Los FPGA son herramientas de desarrollo transversales que pueden emplearse en áreas tan diversas como la Internet de las Cosas (IoT) e investigación genética. Las búsquedas en las tres principales editoriales

científicas relacionadas con ingeniería (IEEE, Elsevier, Springer) arrojan más de 60,000 publicaciones cuyos títulos contienen la palabra FPGA. Las implementaciones de FPGA van de máquinas de estados especializadas a sistemas en chip que procesan lenguaje natural mediante redes neuronales, pasando por sistemas de comunicaciones de nueva generación, transportes autónomos, minería de datos, exploración espacial, por mencionar algunos.

Desde una perspectiva económica, el valor estimado del mercado FPGA en 2017 fue de casi 6 mil millones de dólares, tan solo contabilizando los dispositivos FPGA y sus herramientas de desarrollo, sin considerar los productos habilitados por esta tecnología (Markets and Markets, 2017). La firma de investigación y pronósticos Markets and Markets (2017) estima que este mercado seguirá creciendo en un promedio anual de entre 8% y 9% hasta alcanzar cerca de 10 mil millones de dólares para 2023. Entre las razones que mencionan para este crecimiento están las ventajas de los FPGA, como menor riesgo de obsolescencia para los productos basados en esta tecnología, bajo costo, y desarrollo y depuración rápida de prototipos. Actualmente las principales áreas de aplicación de los FPGA a nivel mundial son:

- Automotriz
- Electrónica de consumo
- Procesamiento de datos
- Industrial
- Militar y aeroespacial
- Telecomunicaciones

El caso particular de México

Cuando vemos la distribución del mercado a nivel mundial, América del Norte (excluyendo México) tiene el 51% del mercado FPGA, en tanto que Europa tiene el 32%, Asia-Pacífico 12%, y el resto del mundo solo 5%. Esto implica que la penetración de la tecnología en nuestro país no es la suficiente como para establecer un mercado atractivo (CMM, 2010).

Antecedentes

Los FPGA fueron patentados en 1984 por parte de la empresa Xilinx; desde entonces sus capacidades y alcances han ido creciendo a la par de la ley de Moore. Los FPGA han representado un campo de

investigación y desarrollo tecnológico tanto a nivel hardware (los propios FPGA) como a nivel software (las herramientas para su uso), y al mismo tiempo han habilitado el desarrollo de nuevos productos y tecnologías (Xilinx, 2008). Algunos ejemplos de patentes habilitadas por la aplicación de FPGA son:

- US5684980A – Computadora Virtual FPGA para ejecutar una secuencia de instrucciones de programa al reconfigurar sucesivamente un grupo de FPGA en respuesta a dichas instrucciones.
- US5903744A – Emulador lógico usando una tarjeta interconectada con wire wrap desechable con una tarjeta de emulación FPGA.
- US6173419B1 – Emulador FPGA para depurar software.
- US6073154A – Cómputo de Transformadas Discretas de Fourier en FPGA.
- US20080086274A1 – Método y Aparato para Alineación de Secuencia de Proteínas Utilizando Dispositivos FPGA

Por su parte, el mercado de tarjetas de desarrollo basadas en FPGA se puede dividir de acuerdo a los siguientes tres usos finales:

- Tarjetas de evaluación/arranque. Estas tarjetas tienen como propósito presentar un nuevo FPGA en particular o bien, introducir al diseño FPGA a estudiantes y profesionistas que no están familiarizados con ellos. Los periféricos que acompañan estas tarjetas son de baja complejidad (sensores, puertos seriales, puertos VGA de 8 bits de color, entre otros) y generalmente son tarjetas de bajo costo.
- Tarjetas de desarrollo. Estas tarjetas tienen como propósito el permitir desarrollar casi cualquier sistema digital que pueda implementarse en un producto real, o bien, desarrollar el producto mismo en dicha tarjeta. La colección de periféricos suele ser de alta complejidad (Gigabit Ethernet, USB 2 o 3, puertos HDMI/DVI, PCIe, entre otros). El costo de estas tarjetas puede ir de los 100 dólares a más de 5,000 dólares.
- Tarjetas especializadas. Estas tarjetas están enfocadas a nichos de aplicación, pueden ser procesamiento de imágenes, comunicaciones de gran ancho de banda, emulación y pruebas de prototipos de ASIC. Los periféricos son especializados para el área de aplicación e incluyen una gran cantidad de IP especializado además de herramientas de diseño específicas para el área de aplicación. Estas tarjetas tienen costos que van de los 2,000 dólares a más de 15,000 dólares.

Algunos ejemplos de tarjetas de desarrollo basadas en FPGA se muestran en las siguientes figuras:



Figura 3. Tarjeta de desarrollo Mojo v3 de Spark Fun.

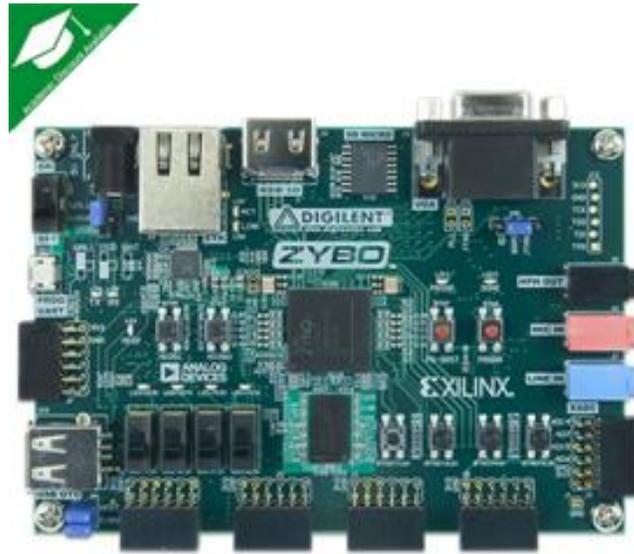


Figura 4. Tarjeta de desarrollo Zybo, de Digilent.

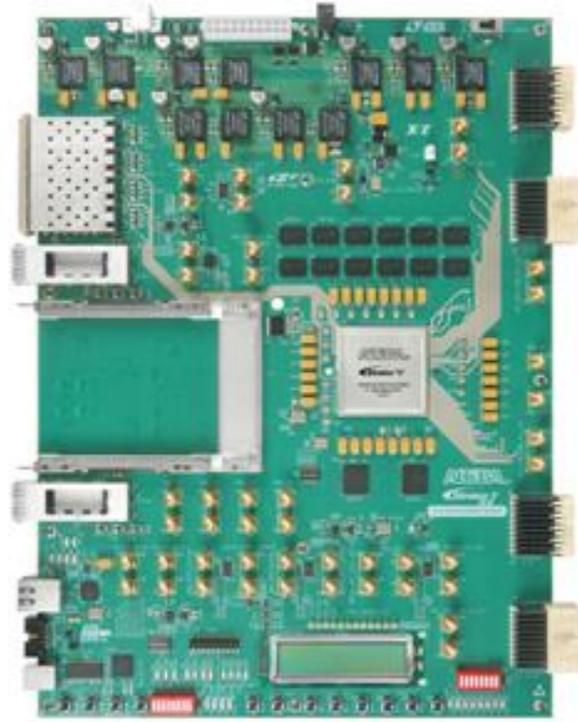


Figura 5. Kit de desarrollo de 100 Gb Stratix V GX de Altera.



Figura 6. Tarjeta de desarrollo Z-turn Board de MyiR.

Si bien estos son productos comerciales recientes, encontramos investigaciones desde los años 90, cuando los FPGA comenzaron a cobrar notoriedad. Los siguientes son algunos ejemplos de investigaciones en este campo y desarrollos de tarjetas con el propósito de facilitar otras soluciones.

Srivastava y Brodersen (1991) desarrollaron un marco de trabajo que acercara el nivel de automatización del diseño de ASIC al desarrollo de sistemas de hardware y software integrados, donde el hardware es un ASIC a la medida y uno o más PCBs para la conexión con otros componentes.

Chan (1994) señala que los FPGA pueden acelerar el desarrollo de prototipos, aunque existen limitantes cuando el prototipo no puede integrarse en un solo FPGA. En su estudio propone usar varios FPGA para rutear conexiones entre los FPGA donde se estén integrando las piezas del diseño a probar, lo cual lleva a cabo en su tarjeta de desarrollo de prototipos XC4000 BORG.

Courtoy (1998) presenta un sistema para validación de prototipos llamado Aptix, que funciona en tiempo real (a diferencia de los emuladores de software) para diseños digitales del área de comunicaciones que se piensan implementar finalmente como ASICs. El desarrollo de prototipos se ha usado desde hace mucho tiempo como método para validar diseños antes de llevarlos a producción. El incremento de la complejidad de los diseños hizo que la inversión en costo y tiempo de un prototipo fuera inviable. Sin embargo, sigue existiendo la necesidad de probar los diseños en escenarios reales que difícilmente se pueden lograr en las herramientas de simulación. Courtoy reconoce que la tecnología FPGA permite regresar al desarrollo de prototipos sin necesidad de una inversión excesiva de tiempo y con resultados aceptables. El sistema integra además un componente llamado FPIC que es un componente de interconexión programable en campo, una especie de tablilla de prototipos a nivel PCB. En cuanto al FPGA, se elige de acuerdo a las preferencias del usuario, por lo que no está ligado a un fabricante en particular.

Ricci y LeHuy (2002) desarrollaron una plataforma de implementación de prototipos de controladores de velocidad variable basada en FPGA combinada con una metodología de diseño utilizando Simulink. Comparan la metodología de desarrollo de prototipos usando una tarjeta con DSP y una tarjeta con FPGA, y mencionan que trabajar con la tarjeta FPGA reduce el tiempo de diseño e implementación, debido al menor tiempo de simulación. Otra ventaja que obtuvieron con el uso de FPGA fue que la plataforma podía adoptar cambios sobre la cantidad de variables medidas, o cambios en el protocolo de comunicación sin tener que modificar la plataforma completa.

Fröning et al. (2006) proponen una plataforma de desarrollo y verificación de prototipos para diseños con tecnología Hypertransport basado en FPGA. Previo a este trabajo no existía una plataforma que permitiera desarrollar un producto con HTX de manera rápida, siendo incluso adoptada por el Hypertransport Consortium como diseño de referencia.

Kang et al. (2007) desarrollaron una plataforma de emulación de circuitos lógicos de bajo costo basada en FPGA. La plataforma sólo contiene un puerto RS232, alimentación, un botón de reset, un oscilador y un generador de patrones y analizador lógico que se integra en el FPGA y se comunica vía puerto RS232. Con esto se pueden verificar diseños en el FPGA sin necesidad de modificar la tarjeta. Si bien el enfoque reduce el costo al reducir al máximo los componentes, la interacción con los diseños sólo es virtual, ya que en este caso no es posible probar los diseños con estímulos externos reales. También denotan la necesidad de ir más allá de sólo simular un diseño y probarlo en una implementación real.

Litz et al. (2009) continúan con el trabajo de Fröning et al. (2006) al desarrollar una plataforma de pruebas de prototipos para HyperTransport 3.x. El estudio muestra que el desarrollo de tarjetas específicas puede verse afectado rápidamente por el cambio de protocolos, no tanto por la capacidad del FPGA, que puede variar, sino por los componentes electrónicos que una vez integrados en un diseño difícilmente pueden cambiarse sin hacer modificaciones mayores.

Henke et al. (2014) proponen una plataforma en línea para el desarrollo de prototipos de sistemas digitales. La plataforma en línea permite interconectar experimentos entre universidades de todo el mundo. Este enfoque es completamente educativo, por lo que las capacidades integradas en la tarjeta para el desarrollo de prototipo son algo limitadas. Sin embargo, el enfoque de hacer la plataforma en línea resulta muy atractivo desde la perspectiva educativa al tener los recursos disponibles de manera permanente para estudiantes y profesores.

Estos son algunos ejemplos de desarrollos e investigaciones con FPGA:

Maldonado et al. (2013) usaron una tarjeta de desarrollo comercial basada en un Spartan 3A (120 dólares) para el control de motor con fuzzy normal y uno optimizado con algoritmos genéticos. Cabe destacar que el lenguaje que seleccionaron para la implementación fue VHDL y utilizaron además la herramienta Toolbox Generator de Xilinx.

Severance y Lemieux (2012) propusieron un procesador vectorial compacto enfocado en la posibilidad de ejecutar implementaciones paralelas para crear un superprocesador SIMD. La implementación está realizada en procesadores Stratix IV, en una tarjeta de desarrollo DE4-530 (3,000 dólares), ambos dispositivos de gama alta. Por la naturaleza del diseño, el procesador propuesto es tratado como IP Core, mientras su funcionalidad es explotada mediante código C.

Chekider et al. (2014) compararon implementaciones de sistema de monitoreo de picos de energía basados en fuzzy, redes neuronales, algoritmos genéticos e híbridos de estos. Usaron Matlab apuntando a código VHDL como ambiente de desarrollo, implementado en un Virtex 5 de una tarjeta ML501-XC5VLX50 (250 dólares).

Tavana y Dinavahi (2015) probaron implementaciones de modelos con matemática de punto fijo y punto flotante para aplicaciones de Hardware en el Lazo usando además técnicas de pipeline. El diseño lo realizaron apoyándose en MATLAB e implementado en VHDL. Utilizan una tarjeta de pruebas VC707 (3500 dólares) con un FPGA Virtex 7 de Xilinx. Cabe señalar que utilizan menos del 10% de los recursos disponibles del FPGA para sus pruebas de implementación.

Herrera et al. (2014) presentaron una plataforma de simulación en tiempo real con un modelo basado en Thevenin de un conmutador para emular las características del estado de encendido considerando los fenómenos electrotérmicos. También incluyen un modelo de hardware en el lazo para un controlador de vehículo para mostrar el desempeño de su modelo. La implementación la realizaron en una tarjeta de aplicación específica para modelado Opal RT, con un FPGA Virtex 6.

Hussain et al. (2014) implementaron un controlador de memoria para aplicaciones de cómputo de alto desempeño en FPGA, que ayuda a mejorar la eficiencia general del procesamiento al administrar mejor el acceso a memoria. La implementación la hicieron en una tarjeta de desarrollo ML505 con un VIRTEX 6 (1,200 dólares), y comparan un algoritmo equivalente al suyo en un Stratix IV.

Zhang et al. (2015) propusieron un método de diseño para redes neuronales por convolución en base a los requerimientos del problema y las restricciones de la implementación. Emplearon una tarjeta de pruebas VC707 (3,500 dólares) con un FPGA Virtex 7 de Xilinx, y utilizan técnicas de HLS para la descripción del diseño.

Van der Wal et al. (2015) realizaron la implementación de un sistema de estereovisión usando una tarjeta de pruebas de diseño propio con un Zynq de Xilinx. Los resultados son comparados contra una implementación en software corriendo en una laptop con procesador de última generación. En los resultados, las ventajas de utilizar la tarjeta en cuanto a procesamiento y reducción de consumo de energía son considerables.

Siddiqui et al. (2014) desarrollaron un procesador para FPGA enfocado en aplicaciones de procesamiento de imágenes. Usaron el SoC Zynq en una Zedboard (500 dólares), aprovechando el espacio reducido del procesador para implementar múltiples procesadores en paralelo. El diseño es codificado a mano utilizando las primitivas de Xilinx, es decir, los bloques de construcción de Xilinx, por lo que el código no es portable a otra familia u otros fabricantes de FPGA.

Toft y Nannarelli (2014) probaron dos diseños de aceleradores de cálculo financiero y lo contrastan con el diseño en software, obteniendo ventajas en capacidad de procesamiento y ahorro en consumo de energía, a expensas del tiempo de diseño y costo de desarrollo. Usaron una tarjeta ADM-XRC-5T2 (cerca de 10,000 dólares) de Alpha Data que porta un Virtex 5 de Xilinx.

Johnson et al. (2015) proponen el aprovechamiento de las características de reconfiguración dinámica parcial de los FPGA sin tener que pagar las licencias de los fabricantes en dispositivos enfocados en la IoT, y en su diseño agregan también el uso de funciones físicamente inclonables en el FPGA para evitar riesgo de intrusiones. Se enfocan en FPGAs basados en SRAM, y el diseño y pruebas los realizan con herramientas de Xilinx.

Podemos apreciar cierta preferencia por las soluciones de Xilinx, quizá debido a que las tarjetas de desarrollo basadas en FPGA de Xilinx son relativamente más económicas que las de Intel/Altera. También, es destacable que solo una investigación de las aquí mencionadas desarrolló un PCB específico, mientras que todas las demás tomaron una tarjeta de desarrollo ya disponible y adaptaron su diseño a la misma. Generalmente en las etapas iniciales de desarrollo de productos, y en primeros niveles de madurez de la tecnología, como se observa en la escala TRL de la NASA, primero se desarrolla una prueba de concepto, es decir, que lo planteado puede funcionar, y sólo después se efectúan los desarrollos de prototipos funcionales (Kedrin, 2011; Héder, 2017).

Con base en la información anterior, el objetivo general de esta investigación es:

Objetivo general

Desarrollar y evaluar el desempeño de una plataforma para desarrollo de prototipos de sistemas digitales basado en FPGA, que pueda abarcar una amplia variedad de aplicaciones de sistemas digitales pertinentes para la diversidad de empresas y centros de investigación de la región.

Con el propósito de alcanzar el objetivo general, se plantearon los siguientes **objetivos particulares**:

- Diseñar y desarrollar una plataforma para desarrollo de prototipos de sistemas digitales, consistente en un PCB que integre un FPGA y una serie de conectores que permitan utilizar una amplia diversidad de periféricos necesarios para la aplicación a desarrollar.
- Integrar una serie de periféricos que faciliten el uso de la plataforma para el desarrollo de prototipos.
- Evaluar el desempeño de la plataforma de manera integral en el desarrollo tecnológico de un sistema de radiografía digital, así como en cada uno de los bloques funcionales digitales que lo componen (FPGA, electrónica de alimentación, memoria flash de configuración, interfaz y conectores) para probar su efectividad como solución genérica para el desarrollo de prototipos.

Justificación

En México es necesario impulsar los sectores con alto potencial de crecimiento, para favorecer la generación de empleos y la integración de todas las regiones del país con los mercados nacionales e internacionales. Una opción viable es impulsar las actividades de alto valor agregado, como el desarrollo electrónico avanzado, al tiempo que se contribuye a la transformación productiva de los sectores tradicionales de la economía.

México actualmente es una potencia manufacturera, y requerimos convertirnos en una economía del conocimiento, para lo cual es necesario fortalecer la innovación y el desarrollo tecnológico nacional

encaminado a incrementar la competitividad del sector industrial, y así generar el conocimiento técnico (know-how) y la independencia tecnológica para fortalecer nuestro país.

Para impulsar estos cambios, es importante desarrollar procesos de planeación y toma de decisiones, basados en el conocimiento de aquellos elementos de innovación que permitirán el desarrollo de ventajas competitivas.

La electrónica se encuentra en todos los aspectos de la vida cotidiana e industrial. De hecho, el sector electrónico es uno de los sectores más dinámicos y con mayor impacto en la actualidad, y es considerado como uno de los sectores con mayor crecimiento en las próximas décadas. Basta mencionar que el porcentaje del costo de un automóvil que representa la electrónica, un 0% en 1960, alcanzará un estimado de 50% para el 2030.

Las nuevas tecnologías electrónicas como FPGA permiten implementar el conocimiento como valor agregado dentro de circuitos integrados, más allá de la simple manufactura de diseños extranjeros (maquila), y de esta manera generar conocimiento que pueda exportarse de manera comercial. Es fundamental desarrollar herramientas que permitan a los estudiantes e ingenieros diseñar e implementar este tipo de tecnología en sus proyectos, para mejorar la competitividad de la región y mantenerse vigentes en capacidad de análisis y solución de problemas.

El desarrollo acelerado de los FPGA y otras tecnologías exige un acercamiento basado en el diseño; es decir, más que preguntarse qué es lo que puede hacer el dispositivo por uno, hay que plantearse qué es lo que uno puede hacer con el dispositivo. De esta manera, si surgen nuevas tecnologías que superen a los FPGA, las capacidades de desarrollo entrenadas seguirán vigentes y se podrán trasladar a nuevas tecnologías, en lugar de sólo tener habilidades en el uso de una plataforma específica que no puede modificarse o adaptarse.

Las técnicas para el diseño con FPGA están relacionadas con las técnicas de diseño de circuitos integrados. En cuanto a la propiedad intelectual, el diseño elaborado pertenece al desarrollador, no a la empresa que manufactura los FPGA. Esto significa que el diseño no es propiedad del fabricante de FPGA, dando al usuario libertad e independencia tecnológica para su uso y comercialización; además, el diseño puede protegerse mediante patente y copyright.

Finalmente, el diseño con FPGA es fácilmente adaptable a las necesidades del diseñador y del usuario. Esto permite aprovechar los nuevos elementos tanto tecnológicos como algorítmicos que puedan surgir una vez que un proyecto ha iniciado, sin necesidad de desechar el trabajo ya realizado ni incurrir en sobrecostos por requerir la adquisición de alguna nueva plataforma de desarrollo.

Capítulo 2: Marco Teórico

2.1 Diseño de sistemas digitales modernos

FPGA

Los FPGA (*Field-Programmable Gate Array*, matriz de puertas programables) son circuitos integrados cuya funcionalidad es definida por el usuario y pueden ser reconfigurables en campo. Los FPGA entraron en el mercado en 1984 de la mano de la empresa Xilinx. Tres años después, otra empresa dedicada a los circuitos lógicos programables, Altera (ahora propiedad de Intel), se unió al mercado. A partir de este momento inició un mercado que se ha visto beneficiado por la rivalidad entre estas dos grandes compañías. El mercado FPGA está acaparado por estas dos empresas que dominan el 88% del mercado FPGA. Otras empresas que también venden FPGA son: Lattice Semiconductors, Microsemi, Quicklogic, Atmel, Achronix y Flexlogic.

Actualmente existen 3 tecnologías principales para la implementación FPGA, estas son: SRAM, FLASH y AntiFuse.

SRAM

Los FPGA basados en SRAM tienen la cualidad de ser fácilmente reconfigurables. La lógica combinacional se separa en LUTs de 2, 3, 4, 5, o hasta 6 entradas (al momento de la escritura de este trabajo) y las entradas de la LUT funcionan como un selector, para cuyas combinaciones hay un 0 o 1 configurado en un registro de SRAM. En las figuras 7 y 8 se aprecia el diseño de un elemento lógico que contiene la tabla de verdad, un circuito de sincronización, un flip flop D y la selección de salida además de la lógica de configuración, y el detalle del funcionamiento de la configuración de la tabla de verdad.

Ventajas:

- Los FPGA SRAM utilizan las técnicas de manufactura de vanguardia. Al estar basados en tecnología CMOS y tener estructuras altamente repetitivas son atractivos de fabricar en los nodos tecnológicos de avanzada.
- No tienen límite en la cantidad de veces que se pueden reconfigurar

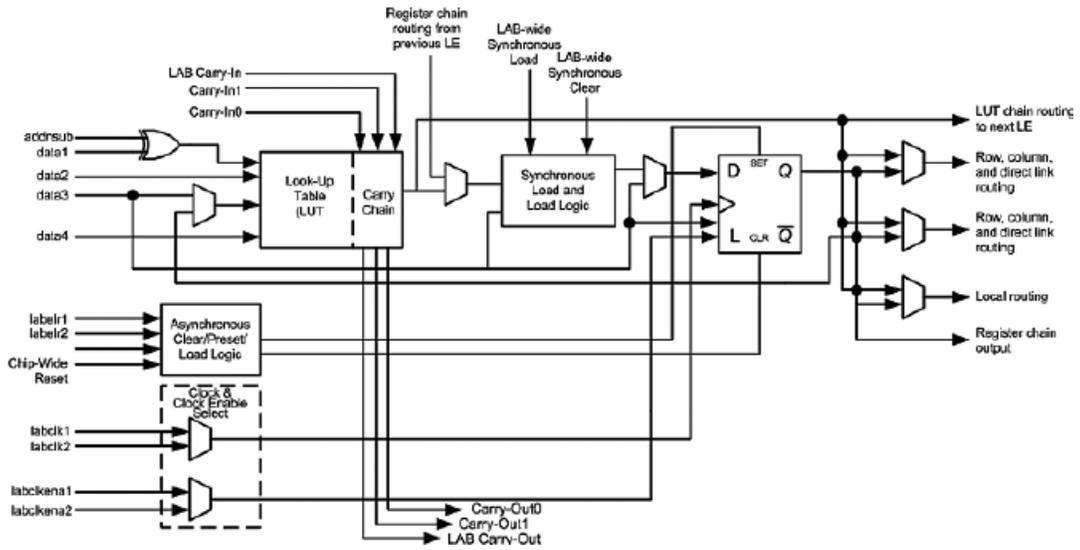


Figura 7. Bloque de elemento lógico básico de un FPGA con LUT de 4 entradas.

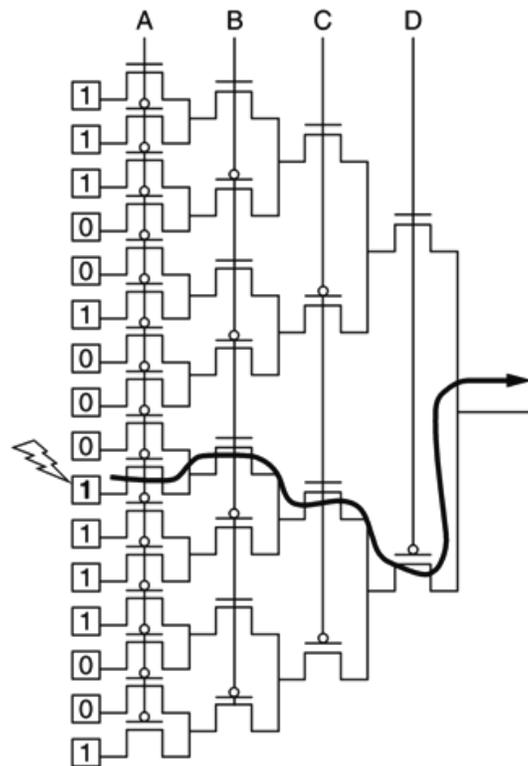


Figura 8. Estructura de la LUT de 4 entradas para un FPGA basado en SRAM.

- El tiempo de configuración es muy bajo (menor a 1 segundo en configuración paralela o desde memoria)
- Mayor capacidad de integración ya que las celdas de SRAM son más pequeñas que las celdas basadas en flash o antifuse.
- Ideal para pruebas de prototipos.

Desventajas:

- Sin energía, no pueden mantener la configuración; es necesario configurar al arranque, generalmente integrando una memoria no volátil que contenga la configuración.
- Susceptible de ataques de ingeniería inversa para obtener la configuración interna (al tener un dispositivo externo que contiene la configuración).
- Susceptible a las alteraciones por eventos únicos. Cuando una partícula cargada impacta el FPGA, puede cambiar el valor de un bit de 0 a 1 o viceversa, pudiendo este bit ser parte de la configuración del FPGA.
- Presentan el consumo de energía más alto entre los FPGA.

FLASH

Los FPGA basados en flash utilizan la misma estrategia de configuración que los FPGA basados en SRAM, con la diferencia de que los elementos de memoria están hechos con memoria FLASH, la cual requiere un proceso de manufactura diferente a los CMOS.

Ventajas:

- Los FPGA están configurados en todo momento, por lo que no requieren configuración externa una vez que han sido programados.
- Pueden configurarse varias veces.
- Tienen protección contra piratería.
- Resistentes a las alteraciones de eventos únicos, por la naturaleza de las celdas FLASH.
- Bajo consumo de energía.

Desventajas:

- La tecnología FLASH está cuando menos 2 o 3 nodos atrás de la tecnología CMOS.
- Los retardos generados por las celdas FLASH son mayores que en las celdas CMOS.

- Las celdas FLASH son considerablemente más grandes que las celdas CMOS, por lo que la integración en el chip es menor.
- El número de veces que se puede reprogramar un FPGA FLASH está limitado a 500.

AntiFuse

Los FPGA con tecnología AntiFuse son más parecidos a los dispositivos PAL de antaño que eran configurados mediante la desactivación de fusibles. El resultado final de estos dispositivos es muy similar a un ASIC de propósito específico.

Ventajas:

- Activos en el arranque.
- Virtualmente imposibles de piratear.
- El menor consumo de energía de todos los FPGA.
- La mayor resistencia a las alteraciones por eventos únicos.

Desventajas:

- Programables sólo una vez.
- La programación no puede ser in-situ (agregando una etapa al proceso de producción).
- No es viable para pruebas de prototipos.
- El menor nivel de integración de todos los FPGA.
- Mayor costo por unidad lógica de todos los FPGA.

Lenguajes Descriptivos de Hardware

VHDL

Es un lenguaje definido por el IEEE (Institute of Electrical and Electronics Engineers) (ANSI/IEEE 1076-1993) usado por ingenieros y científicos para describir circuitos digitales o modelar fenómenos científicos respectivamente. VHDL es el acrónimo que representa la combinación de VHSIC y HDL, donde VHSIC es el acrónimo de *Very High Speed Integrated Circuit* y HDL es a su vez el acrónimo de *Hardware Description Language*. Para el modelado físico existe la derivación del lenguaje VHDL-AMS.

Originalmente, el lenguaje VHDL fue desarrollado por el Departamento de Defensa de los Estados Unidos a inicios de los años 80 basado en ADA, con el fin de realizar simulación de circuitos eléctricos digitales. Sin embargo, posteriormente se desarrollaron las herramientas de síntesis e implementación en hardware a partir de los archivos VHD. Aunque puede ser usado de forma general para describir cualquier circuito digital se usa principalmente para programar PLD (*Programmable Logic Device*, dispositivo lógico programable), FPGA (*Field Programmable Gate Array*), ASIC y similares.

Verilog

Es un lenguaje de descripción de hardware usado para modelar sistemas electrónicos. El lenguaje, algunas veces llamado Verilog HDL, soporta el diseño, prueba e implementación de circuitos analógicos, digitales y de señal mixta a diferentes niveles de abstracción.

Los diseñadores de Verilog querían un lenguaje con una sintaxis similar a la del lenguaje de programación C, de tal manera que le resultara familiar a los ingenieros y así fuera rápidamente aceptada. El lenguaje tiene un preprocesador como C, y la mayoría de palabras reservadas de control como “if”, “while”, etcétera, son similares. El mecanismo de formateo en las rutinas de impresión y en los operadores del lenguaje (y su precedencia) son también similares. En cuanto a las diferencias, Verilog no tiene estructuras, apuntadores o funciones recursivas. Finalmente, el concepto de tiempo, muy importante en un HDL, no se encuentra en C.

Un diseño en Verilog consiste de una jerarquía de módulos. Los módulos son definidos con conjuntos de puertos de entrada, salida y bidireccionales. Internamente un módulo contiene una lista de cables y registros. Las sentencias concurrentes y secuenciales definen el comportamiento del módulo, describiendo las relaciones entre los puertos, cables y registros. Las sentencias secuenciales son colocadas dentro de un bloque begin/end y ejecutadas en orden secuencial, pero todas las sentencias concurrentes y todos los bloques begin/end son ejecutadas en paralelo en el diseño. Un módulo puede contener una o más instancias de otro módulo para definir un sub-comportamiento.

SystemC

Es frecuentemente descrito como un lenguaje de descripción de hardware, pero es más adecuado describirlo como un lenguaje de descripción de sistemas, puesto que es realmente útil cuando se usa para modelar sistemas a nivel de comportamiento.

SystemC es un conjunto de bibliotecas y macros implementadas en C++ que hacen posible una simulación de procesos concurrentes con la sintaxis del lenguaje C++ ordinario. Así los objetos descritos pueden comunicarse durante una simulación de tiempo real usando señales de cualquier tipo ofrecido por C++, además de algunas otras ofrecidas por las librerías de SystemC y también otras definidas por el usuario.

La metodología de diseño es comenzar con un modelo de alto nivel escrito en C++ y aplicar un proceso iterativo consistente en transformar el código para usar sólo los elementos que tengan su equivalente en un lenguaje de descripción de hardware.

OpenCL

OpenCL (*Open Computing Language*, lenguaje de computación abierto) consta de una interfaz de programación de aplicaciones y de un lenguaje de programación. Juntos permiten crear aplicaciones con paralelismo a nivel de datos y de tareas que pueden ejecutarse tanto en unidades centrales de procesamiento (CPU) como unidades de procesamiento gráfico (GPU). El lenguaje está basado en C99, eliminando ciertas funcionalidades y extendiéndolo con operaciones vectoriales.

Este lenguaje fue especificado por Apple y desarrollado en conjunto con AMD, IBM, Intel y Nvidia. The Khronos Group es el encargado de estandarizar el lenguaje.

Si bien las plataformas objetivo del OpenCL eran las GPU, los FPGA pueden aprovechar este lenguaje e implementar una arquitectura de procesadores en paralelo que permita ejecutar el código de manera óptima. El fabricante Altera (ahora propiedad de Intel) tiene amplia experiencia en la implementación de diseños OpenCL en sus FPGA, y el fabricante Xilinx a partir de la versión 2016 introdujo el soporte para OpenCL en sus herramientas de diseño.

Diseño Electrónico Avanzado

La tarjeta de circuito impreso es la plataforma sobre la cual los componentes electrónicos como chips semiconductores, capacitores y/o resistencias son montados, ya que provee las interconexiones eléctricas entre componentes, y se encuentran prácticamente en todos los productos electrónicos. En la década de

1950, la tecnología de interconexión conocida ahora como placas de alambre impresas (PWB, *Printed Wiring Board*) o placas de circuito impreso (PCB, *Printed Circuit Board*) había ganado la aceptación comercial. Ambos términos son sinónimos, pero el primero es más utilizado.

Estas placas de circuito se refieren a menudo como “laminados” porque se construyen de varias capas finas u hojas. En el caso de las placas más simples, una capa de base aislante tiene pistas conductoras formadas en uno o ambos lados. La capa de base puede técnicamente denominarse como el sustrato, pero este término se utiliza raramente en el mundo de la tarjeta de circuitos (LaDou, 2006; Shimohashi et al., 2009; Maxfield, 2009; Tsai, 2012).

La fabricación de las tarjetas de circuito impreso es altamente complicada, se necesita una variedad de tecnologías, una gran inversión en equipos y más de 50 pasos en el proceso. En Asia se producen tres de cada cuatro tarjetas de circuito impreso en el mundo (LaDou, 2006; Digney et al., 2013). El enfoque más tradicional consiste en una tableta de fibra de vidrio impregnado con resina epóxica. Un circuito puede incluir trazos conductores en un lado (tarjeta de circuito impreso de una sola cara), ambos lados (dos caras), y ambos lados más capas internas (multicapa) (Digney et al., 2013; Marques et al., 2013).

De acuerdo a los estándares internacionales que maneja la industria electrónica, regidos por IPC (*Association Connecting Electronics*, antes *Institute of Printed Circuits*), existe cierto grado de aceptabilidad en las imperfecciones que pueden presentar los circuitos electrónicos, ya sea en la fabricación del PCB, fabricación de los componentes, o el soldado de los mismos, dependiendo del grado de sofisticación que se requiera (IPC, 2004; Rowe, 2010).

El estándar internacional IPC-2222 (Ferrari y Madox, 2010) provee información para el diseño de diferentes tipos de tarjetas, ya sea de tecnología SMT (*Surface Mount Technology*, montaje superficial) o bien THT (*Through Hole Technology*, tecnología de orificio). En conjunto con el IPC 2221B (Ferrari y Madox, 2012), estas dos normas establecen los requisitos para el diseño de tarjetas de circuitos impresos orgánicos rígidos y otras estructuras de montajes de componentes e interconexiones. Además, proveen conceptos clave de diseño para la optimización en el ensamble de circuitos impresos.

Un diseño adecuado de las tarjetas de circuito impreso puede hacer la diferencia entre un producto que pase las pruebas de calidad desde el primer ciclo, o no. Se podría decir que el diseño de circuitos electrónicos se divide en 2 áreas: una en la que el circuito diseñado debe cumplir con las funciones

especificadas, algunas veces bajo condiciones de laboratorio; y la segunda en la que el circuito diseñado debe cumplir con las funciones especificadas para todos y cada uno de los circuitos que se produzcan de este modelo, sin presentar otras funciones no deseadas ni especificadas, siempre en su campo de aplicación durante todo su ciclo de vida.

Si bien la confiabilidad de un dispositivo no depende únicamente del diseño y ensamble, ya que existen otros factores como los materiales utilizados para la fabricación del PCB, la manufactura de los componentes y el soldado de los mismos, conocer el proceso de diseño y fabricación nos da una visión del proceso en general y nos permite anticipar algunos contratiempos que podrían suceder.

Una de estas otras razones por las que un diseño podría ser no confiable, es la interferencia electromagnética (EMI, *Electro-Magnetic Interference*). Pero además de saber escoger los componentes y empaquetados con niveles adecuados de blindaje ante la radiación, el propio diseño del PCB es la primera área de oportunidad para optimizar el funcionamiento. Una gran cantidad de revisiones de PCBs, terminan concluyendo que el diseño del PCB podría haber ayudado a ahorrar miles de dólares en pruebas y reparaciones de último minuto, además de reducir costo por unidad, costos de hardware para blindajes, empaquetados, etc. (Williams, 2010).

Radiografía digital

Los rayos X son una forma de onda electromagnética con una longitud de onda entre 10 nm y 0.01 nm, que corresponden a frecuencias entre 30×10^{15} y 30×10^{18} Hz (Patanachai et al., 2010). En 1895, Wilhem Conrad Roentgen tomó la primera radiografía (de la mano de su esposa), generando un gran cambio en el desarrollo de la ciencia médica (Johnston et al., 2011). Ese mismo año se tomó la primera radiografía dental, y desde entonces forma parte de la práctica odontológica (Ilić y Stojanović, 2010). Actualmente, la imagen de proyección de rayos X es una de las principales herramientas de diagnóstico dental (Kolehmainen et al., 2006).

Acompañando al descubrimiento, se suscitaron problemas originados por la sobreexposición a los rayos X. Desde entonces se han establecido reglas para la limitación de la exposición a los rayos X. Esta protección contra la radiación está basada en los principios de justificación, optimización y limitación de

dosis (Fanning, 2008). Las técnicas y tecnologías para la toma de radiografías mejoraron gradualmente, con emisores de rayos X más potentes, mayor control sobre los parámetros de voltaje y corriente de las placas, así como un mejor nivel de sensibilidad. Todos estos avances permitieron disminuir la radiación necesaria para tomar radiografías, entre 1960 y la segunda parte de la década de 1980 (Johnson y Goetz, 1986).

Con la llegada de la era electrónica, se introdujo equipo especializado en diferentes fases del procesamiento de imágenes (Soonil et al., 2005), incluyendo la captura de imágenes y dentro de esta área, la captura de radiografías. El desarrollo tecnológico de la radiografía digital y computacional ocurre al mismo ritmo que las modalidades de imágenes crosccionales como la tomografía computarizada y resonancia magnética. El desarrollo de la tecnología de detectores para la radiografía computarizada ha reducido las dosis requeridas para los diferentes detectores (escaneo lineal, lectura dual, entre otros). La mayoría de los sistemas de radiología digitales modernos reducen de manera sustancial la dosis de radiación que recibe el paciente respecto a los métodos analógicos tradicionales, aunque puede llegar a ocurrir lo contrario (Uffmann y Schaefer-Prokop, 2009).

Dentro de las aplicaciones de la radiografía dental encontramos, además de las habituales como la revisión del canal del nervio de las piezas dentales: generación de mapa 3D de las piezas para valoración de tejidos orales cuyo análisis por tomografía computarizada implicaría una dosis de radiación excesiva para el paciente y una baja resolución local; diagnóstico de caries (Kasunori, 2012); identificación de cuerpos post mortem (Raju y Modi, 2011; Harandi et al., 2011; Po-Whei et al., 2012; Hui et al., 2012); estimación de la edad (Garamendi y Landa, 2010); y detección de osteoporosis mediante el análisis de la radiografía del hueso alveolar (Southard y Southard, 1996).

Las radiografías presentan una diferente calidad de imagen: en general, a mayor nivel de radiación corresponde una mayor definición de imagen, pero también una mayor exposición del paciente. Para lograr una radiografía útil en el diagnóstico médico es necesario estimar la dosis ideal, en base a una evaluación del contraste espacial, contraste de umbral y la relación señal a ruido (González et al., 2012). Reportes recientes muestran que las fuentes de radiación médica superan a la radiación natural y la de fondo como la fuente primaria de exposición para la población en Estados Unidos (Johnston et al., 2011).

Con base en estas y otras características, la Comisión Internacional de Protección Radiológica (ICRP) ha establecido niveles de referencia de diagnóstico (DRL, *Diagnostic Reference Levels*) para estimar la

dosis de radiación mínima necesaria que recibirá el paciente. En general, se busca que este nivel sea tan bajo como sea razonable para obtener una imagen adecuada para el diagnóstico, en detrimento de la calidad de la radiografía. Esto es particularmente crítico en el campo de los exámenes radiográficos pediátricos (Siebert, 2004).

Uno de los principios para la protección contra la radiación es la optimización, en este caso el proceso de mantener la dosis de radiación tan baja como sea razonablemente práctico (Fanning, 2008). En este sentido existen varios esfuerzos en diversas direcciones, como la estandarización de un índice para indicar la sensibilidad de los aparatos de radiografía digital, que se ha visto reflejado en la creación del estándar IEC 62494-1 (Siebert y Morin, 2011). Otras investigaciones establecen un nivel de referencia de diagnóstico (Fanning, 2008) o niveles de referencia de dosis (González et al., 2012). Un detalle importante es estos esfuerzos buscan objetivos diversos y utilizan unidades de medida diferentes las cuales son Gray o Sievert.

Otro camino para la optimización es el cambio a una tecnología más eficiente en la captura de la radiografía como los sensores digitales, que se caracterizan por la flexibilidad para soportar una reducción en el nivel de radiación utilizado sin sacrificar excesivamente la calidad de la imagen (Uffmann y Schaefer-Prokop, 2009). Los sensores digitales permiten reducir de manera significativa la dosis para obtener una buena radiografía respecto a los métodos analógicos tradicionales (Uffmann y Schaefer-Prokop, 2009), aunque a veces estos mismos sensores pueden ocasionar lo contrario.

De acuerdo a las diferentes tecnologías utilizadas, las imágenes de rayos X para diagnóstico pueden ser obtenidas con diversas condiciones de exposición y dosis. La ICRP recomienda establecer niveles de referencia de diagnóstico (DLR). Para radiografía dental, el DLR de dosis absorbida en el aire es de 4 mGy (miliGray) para una proyección de un maxilar, lo cual es muy superior a lo requerido por el sensor digital (González et al., 2012). Un estudio desarrollado en Wicklow, Irlanda, mostró que, con una película rápida, el nivel de referencia de diagnóstico para una radiografía molar maxilar se reduce a 2.54 mGy, en comparación con los 4 mGy establecidos para Europa, y tan solo 0.95 mGy para la radiografía digital (Fanning, 2008).

El procesamiento de imágenes tiene un campo de aplicación importante en las áreas biométrica y biomédica (Oprea et al., 2008). El procesamiento de imágenes abarca desde la mejora de la calidad de la imagen hasta el análisis de la información visual de manera automática, o bien, la preparación de la

imagen para una inspección manual. Oprea et al. (2008) enumeran varias técnicas de procesamiento de imágenes para aplicaciones biométricas y biomédicas. Kolehmainen et al. (2006) desarrollan un método de inversión bayesiana para unir varias proyecciones de rayos X en un mapa 3D. Southard y Southard (1996) proponen el uso de un algoritmo digital para la detección de cambios óseos que puedan reflejar la afectación del hueso por osteoporosis.

Las técnicas de procesamiento de imágenes permiten la segmentación y extracción de características que coadyuvan a identificar dientes de manera individual, características específicas para la identificación post mortem, o bien como ayuda en el análisis manual de la radiografía, como plantean diversos estudios (Patanachai et al., 2010; Harandi et al., 2011; Raju y Modi, 2011; Hui et al., 2012; Po-Whei et al., 2012).

En 1987 Trophy Radiology lanzó el primer sistema de radiografía digital llamado RadioVisioGraphy (RVG). El sistema canaliza la señal con fibra óptica y lo envía a un CCD especializado (Ilić y Stojanović, 2010). En 2005 se reportó un trabajo de desarrollo de un sistema de radiografía digital intraoral basado en un sensor CMOS (Soonil et al., 2005), en el que se destacan los beneficios de un sistema digital respecto a uno analógico, tales como una menor dosis de radiación, disponibilidad de la imagen en tiempo real, evitar el uso de químicos, así como ventajas en el almacenamiento y recuperación de la imagen.

Capítulo 3: Materiales y Métodos

3.1 Diseño e implementación de tarjeta de desarrollo tecnológico de sistemas digitales basada en FPGA para la plataforma.

La tarjeta consistió en un diseño de tarjeta monobloque (una sola tarjeta) que integraba los siguientes componentes:

- FPGA Spartan 6 de 676 pines
- Memoria Flash de configuración
- Electrónica para alimentación (incluye reguladores, inductancias, capacitores y resistencias)
- Elementos de interfaz directa (leds, interruptores, botones y display 4x7)
- Conectores para tarjetas periféricas (6 conectores de 10 pines, 3 conectores de 20 pines, 3 conectores de 40 pines y 2 conectores de 50 pines)
- Conector de configuración

En este diseño se buscó maximizar la cantidad de pines disponibles para el usuario mediante una amplia variedad de conectores para tarjetas periféricas y conectores de distintos tamaños. De esta manera el propio usuario, de acuerdo a los requerimientos y periféricos de su prototipo, podrá seleccionar los conectores adecuados para un correcto funcionamiento.

Esta línea de pensamiento se tomó después de analizar y trabajar con diversas tarjetas de desarrollo, algunas de las cuales tenían una gran cantidad de pines disponibles para el usuario, pero todos se encontraban en un solo conector, lo que obligaba a integrar todos los periféricos en una sola tarjeta. De igual manera, si se requería agregar otra funcionalidad al prototipo, era necesario hacer una nueva tarjeta para conectarla, descartando el desarrollo de la tarjeta periférica anterior. Además, basta con analizar el nivel de aprovechamiento del conector. Si nuestros periféricos requieren sólo 30 pines para comunicarse y se tiene un solo conector de 150 pines, se utiliza sólo el 20% de la capacidad del conector.

También se presentó el caso contrario, tarjetas con varios conectores, pero de menor tamaño (usualmente entre 6 y 10 pines por conector). En este caso, si alguno de nuestros periféricos requiere de 30 pines, para utilizarlo se tendría que diseñar una tarjeta periférica en la que puedan integrarse varios conectores a la vez. Esto requiere que los diversos conectores estén alineados, pudiendo obstaculizar los demás conectores, o bien utilizar cables para realizar la conexión, lo que puede ser una fuente de ruido.

Por tales motivos, en el desarrollo de esta tarjeta se seleccionaron diversos tamaños de conectores de 10, 20, 40 y 50 pines, para abarcar diversas aplicaciones como:

- 10 pines – Memorias seriales, comunicación UART, USB, Ethernet, SPI CAN, sensores de imagen CMOS, entre otras.
- 20 pines – Conversión de señales analógico/digital en paralelo o multicanal, sensores de imagen CCD.
- 40 pines – Conexión a chiplets USB3, Gigabit Ethernet, conexiones paralelas con microcontroladores.
- 50 pines – Conexión a memoria RAM, conexiones de alto ancho de banda.

En el diseño de esta tarjeta se incluyeron como elementos de interfaz directa 8 leds, 8 interruptores, 4 botones y un display 4x7 para pruebas básicas de depuración de los diseños implementados en el FPGA en tiempo real. Un ejemplo de uso es un hipervisor del diseño que permita monitorear los valores en los registros o bien los estados actuales de las máquinas, de acuerdo con la selección de los interruptores. Incluso es posible modificar estados o registros utilizando los interruptores y botones para introducir estados o valores forzados.

El FPGA utilizado es un SPARTAN 6 XC6SLX45-2FGG676C acoplado con una memoria flash de configuración XCF08PVOG48C. Este FPGA presenta una densidad de 43,661 LUT de 6 entradas y el doble de flip flops, 401 Kbits de RAM interna, 58 multiplicadores de 18x18 bits, y 358 pines disponibles para el usuario. En esta versión el FPGA está soldado a la tarjeta; los conectores y los pines del FPGA están conectados de manera directa (figura 9).

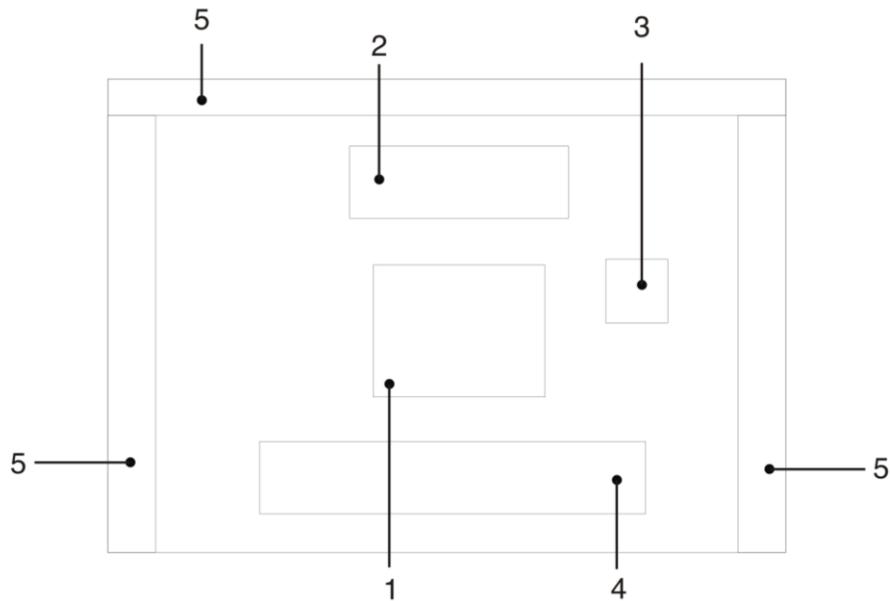


Figura 9. Concepto general de la tarjeta propuesta: 1) FPGA; 2) electrónica de alimentación; 3) memoria flash de configuración; 4) elementos de interfaz; y 5) variedad de conectores.

3.2 Prototipo de plataforma para radiografía digital

Se consiguió un sensor de rayos X CMOS modelo S10835 de la marca HAMAMATSU. El sensor CMOS, a diferencia de los sensores CCD, entrega una salida digital. Este sensor requiere que el sistema maestro genere una señal de reloj (mclk) de entre 1 y 20 MHz, así como una señal de carga (MST) para indicar el momento de la captura de los rayos X para la radiografía.

A diferencia de otros sensores, el S10385 puede emitir un valor durante la etapa de captura de los rayos X que permite conocer la cantidad de radiación capturada por el sensor, lo cual permite a su vez enviar una señal de apagado a la fuente de rayos X en el momento adecuado. Después de finalizar la captura de los rayos X, el sensor envía la imagen capturada mediante dos señales: data y sclk. La señal data lleva la información de los valores de los píxeles capturados y sclk se encarga de sincronizar los datos.

La figura 10 muestra el flujo de operación del sensor S10385. Mientras MST es igual a 0, el sensor CMOS captura los rayos X que detecta; si MST está en cero por un tiempo muy prolongado, la imagen de la radiografía obtenida puede “velarse”, eliminando detalles importantes de la imagen. En caso

contrario, si MST está en cero por un tiempo muy corto, la radiografía puede verse muy “opaca” quedando ocultos detalles importantes para el diagnóstico. El sensor no define un tiempo máximo o mínimo de activación de MST; por tanto, será el radiólogo u odontólogo quien definirá este tiempo o las condiciones de operación, de acuerdo a su experiencia y a las necesidades del paciente.

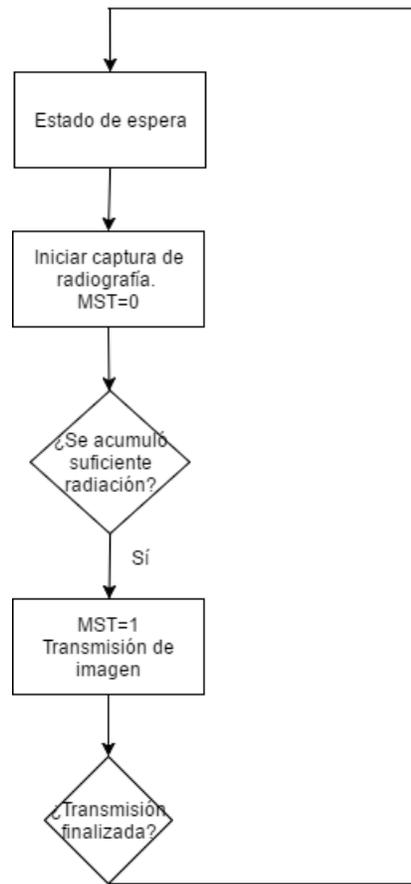


Figura 10. Flujos de operación del sensor S10385.

Durante todas las etapas del ciclo la señal MCLK debe mantener su frecuencia de operación. Durante la etapa de espera y captura, sclk tiene una frecuencia con un valor correspondiente a (frecuencia de MCLK)/56. En la etapa de transmisión de imagen la frecuencia de sclk es igual que la de MCLK. En el caso de data, durante la etapa de espera seguirá un patrón que se describirá más adelante. Durante la etapa de captura, data contiene la información de los sensores de radiación, en una resolución de 10 bits. En la etapa de transmisión, data envía la información de la radiografía píxel por píxel con una resolución de 14 bits.

Las figuras 11, 12 y 13 muestran los diagramas de tiempos de las señales de entrada, salida e internas del sensor en general, durante la etapa de captura y durante la etapa de transmisión respectivamente.

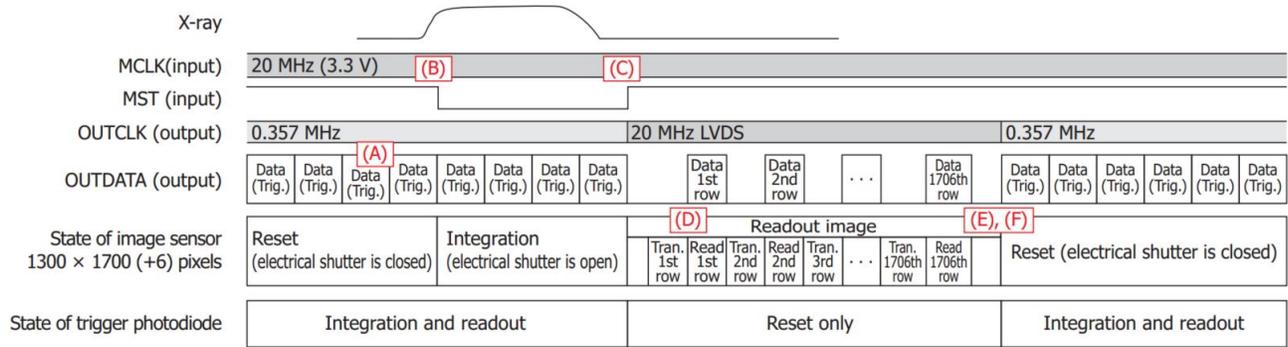


Figura 11. Diagrama de tiempos general del sensor S10385

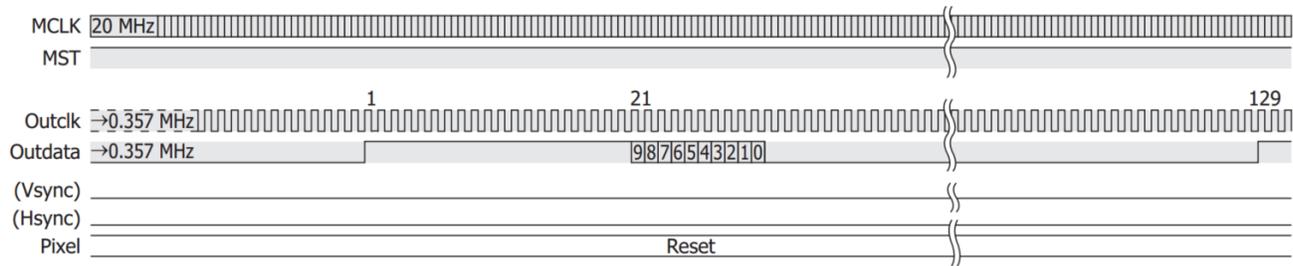


Figura 12. Diagrama de tiempos para lectura de medición de radiación (en 10 bits).

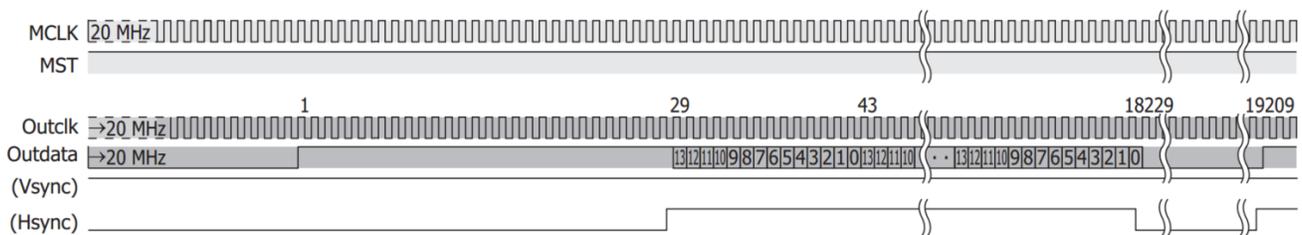


Figura 13. Diagrama de tiempos de transmisión de la imagen obtenida por el sensor.

El sensor S10385 captura radiografías con una resolución de 1300 x 1700 píxeles, además de 6 renglones en negro que forman parte del sensor, aunque no son parte del área activa del mismo. Cada píxel es de 14 bits por lo que se requiere un registro de 16 bits por píxel. Esto da un total de 4,420,000 bytes de información, equivalentes a 35,360,000 bits (35.3 Mb) que hay que almacenar. Para la familia Spartan 6, a la cual pertenece el FPGA seleccionado en este trabajo, el máximo de memoria interna es de 34 Mb. Al momento de la redacción, el FPGA con mayor tamaño de memoria RAM es el Virtex 7 de Xilinx, con 68 Mb. Si bien existe un FPGA de Altera que cuenta con 16 Gb de memoria RAM, esta memoria es dinámica, y el dispositivo aún no está disponible de manera comercial.

Capítulo 4: Resultados y discusión

4.1 Tarjeta de circuito impreso de la plataforma para desarrollo de prototipos

El diseño de la tarjeta requirió de 16 capas para su ruteo. El material base de la tarjeta es FR4, por ser el de manejo comercial más estándar. Los conectores son de la marca Samtec y soportan frecuencias de hasta 1 GHz con una atenuación de -3 dB a esa frecuencia. Los conectores están disponibles comercialmente con distribuidores al por menor como Digikey, Mouser y Newark, y sus contrapartes están disponibles para montaje en PCB y para cable, por lo que las tarjetas periféricas pueden diseñarse para conectarse ya sea directamente a la tarjeta o bien mediante cable, y construirse localmente. La selección de los capacitores para el filtrado de la alimentación de los bancos de pines de entrada/salida fue de tantalio en un encapsulado 0201, ya que, al ser una cantidad considerable, otro tamaño habría significado cuadruplicar la superficie del PCB ocupada por estos capacitores.

El tamaño final de la tarjeta fue de 160.02 mm x 152.4 mm, dando una superficie total de 243.87 cm². El tamaño está relacionado con la disposición de componentes y el tipo de conectores, *through hole*, los cuales se colocan en una cara y son soldados por la cara contraria mediante técnica de ola, o bien de manera manual con cautín. Las figuras de la 14 a 24 muestran el diseño de la tarjeta, el renderizado 3D de la tarjeta, y diagramas esquemáticos de las conexiones.

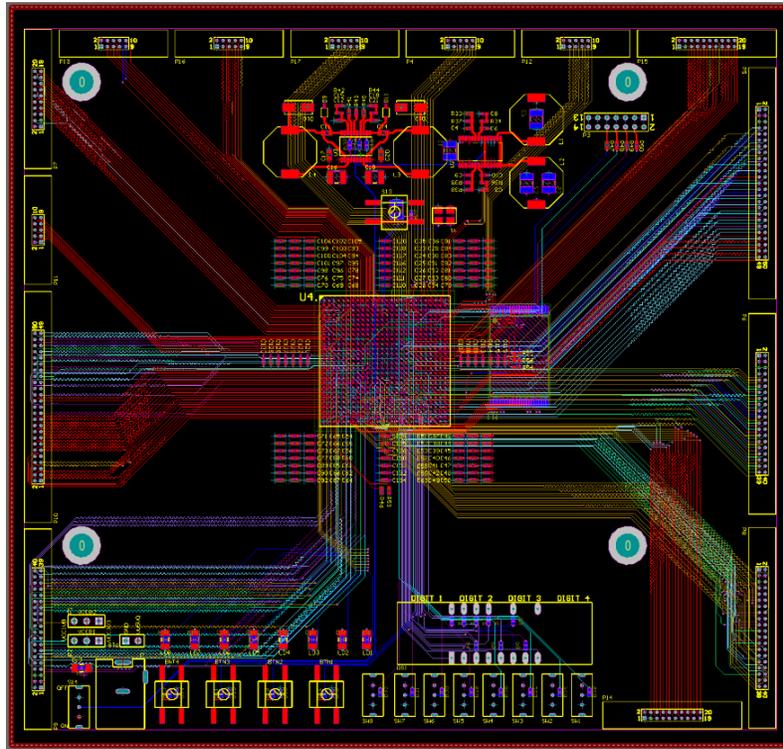


Figura 14. Presentación de las pistas interiores de la tarjeta.

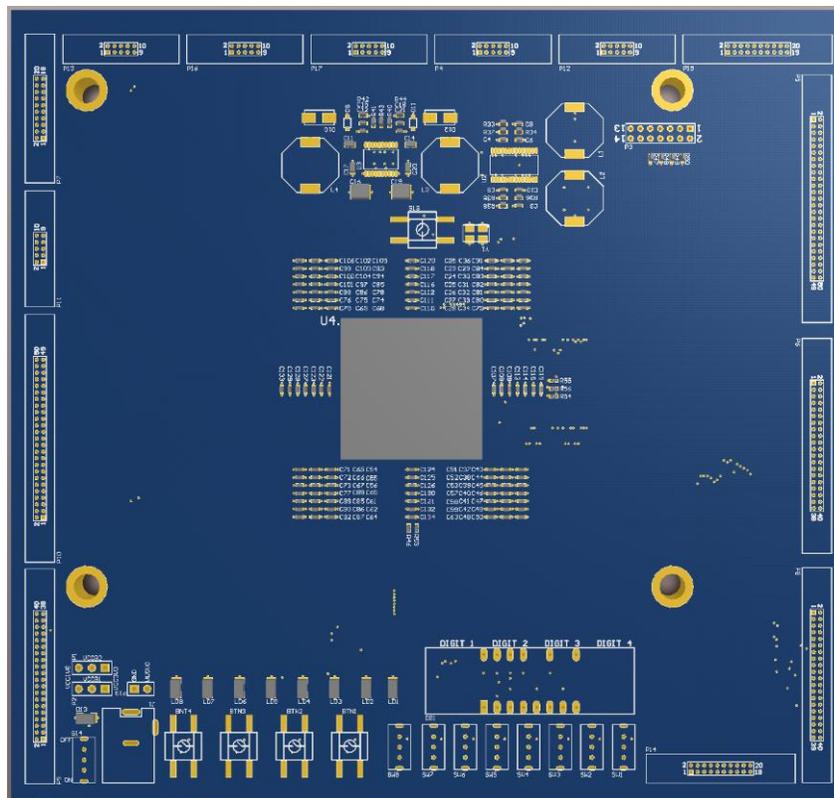


Figura 15. Presentación 3D de la cara superior de la tarjeta.

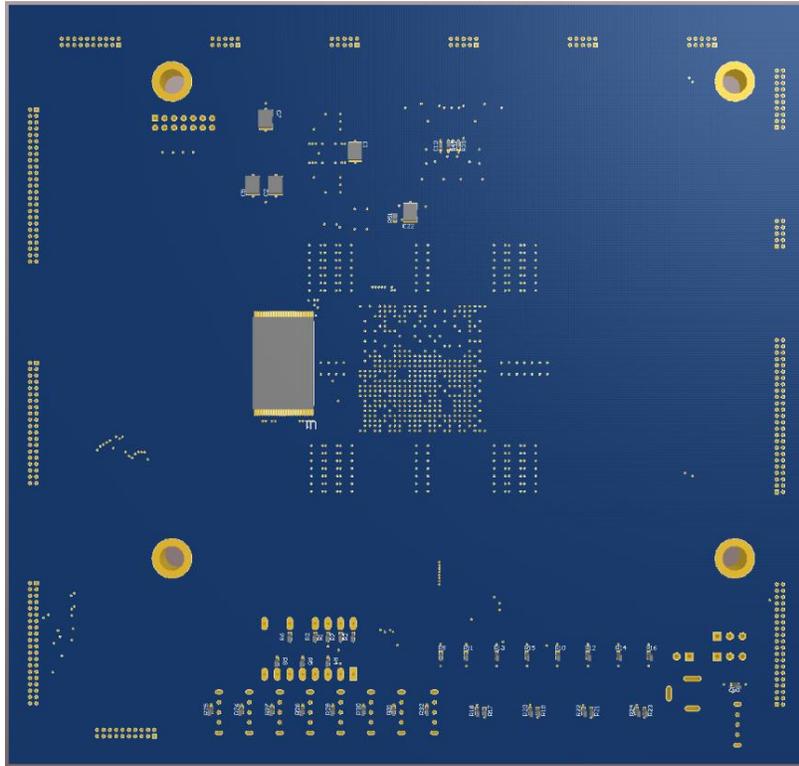


Figura 16. Presentación 3D de la cara inferior de la tarjeta.

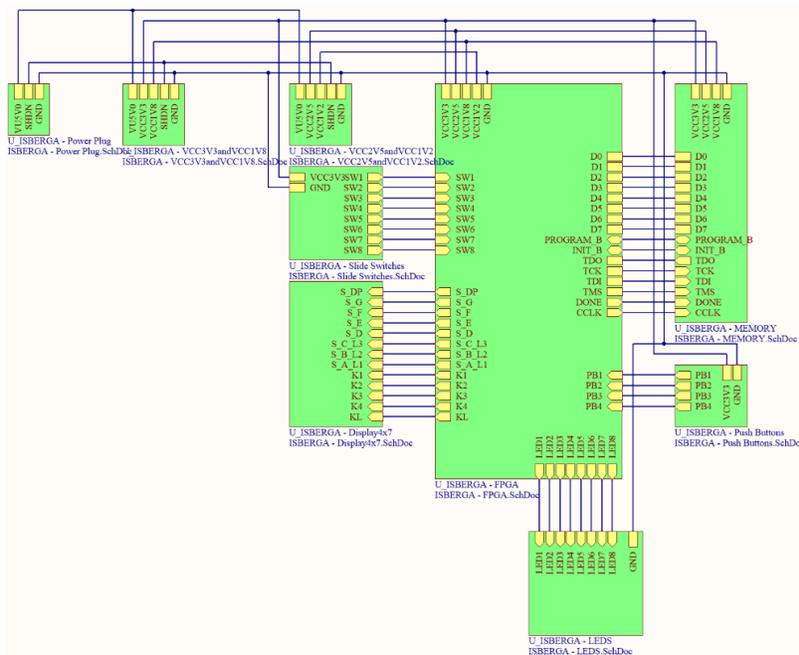


Figura 17. Esquemático general de las conexiones del FPGA con los conectores, alimentación, periféricos en la tarjeta y la memoria flash.

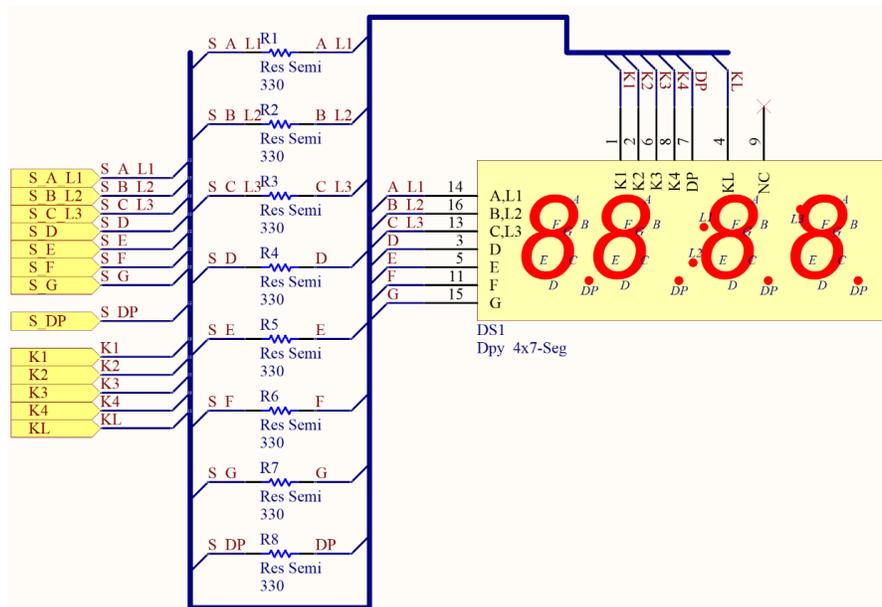


Figura 18. Esquemático detallado de la conexión del FPGA al display 4x7.

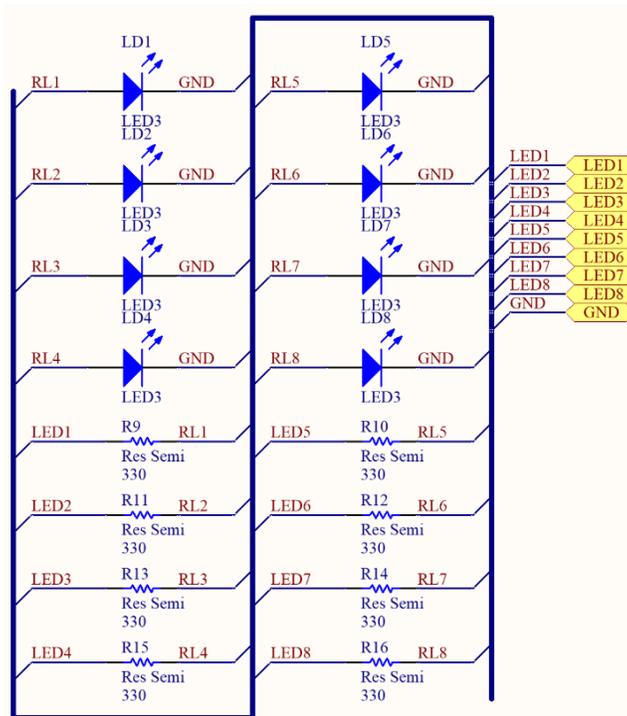


Figura 19. Esquemático detallado de la conexión del FPGA a los LEDs.

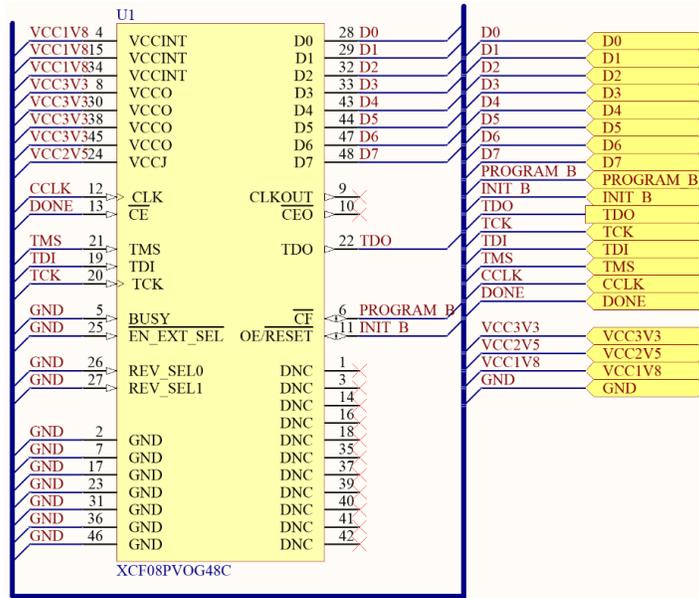


Figura 20. Esquemático detallado de la conexión del FPGA al display 4x7.

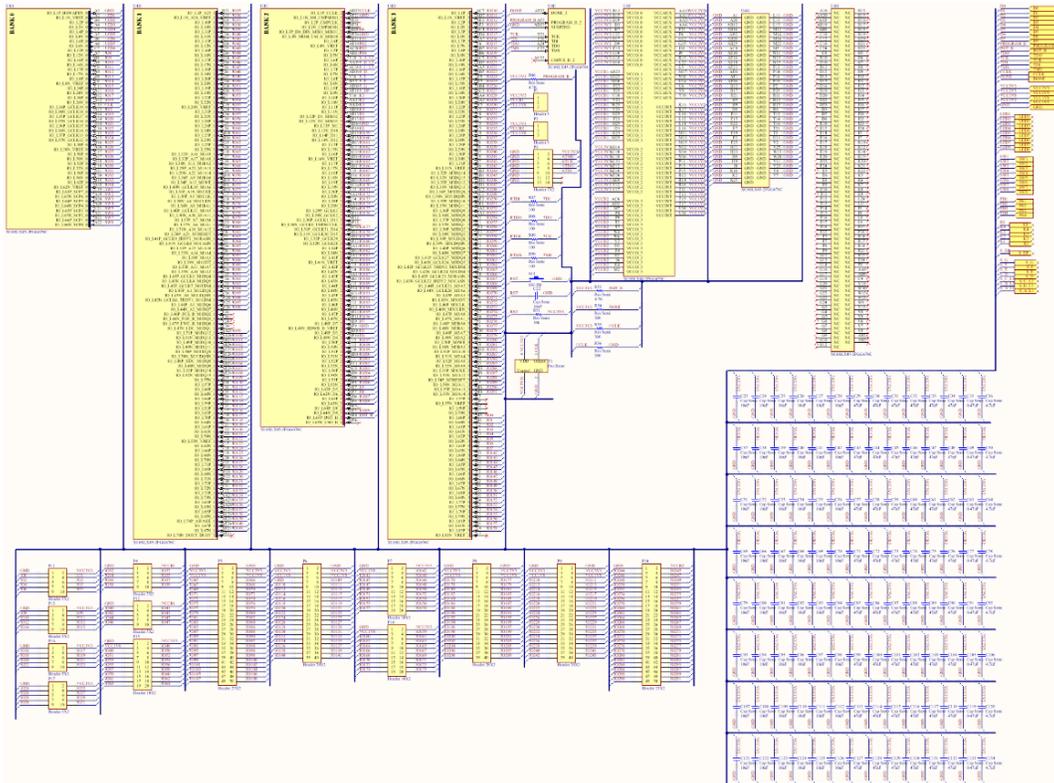


Figura 21. Esquemático detallado de las conexiones del FPGA y los capacitores de regulación de la alimentación de los bancos.

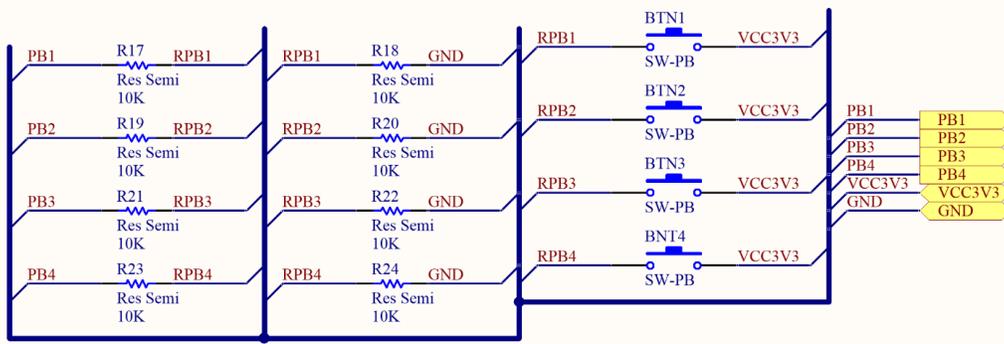


Figura 22. Esquemático detallado de la conexión del FPGA a los botones.

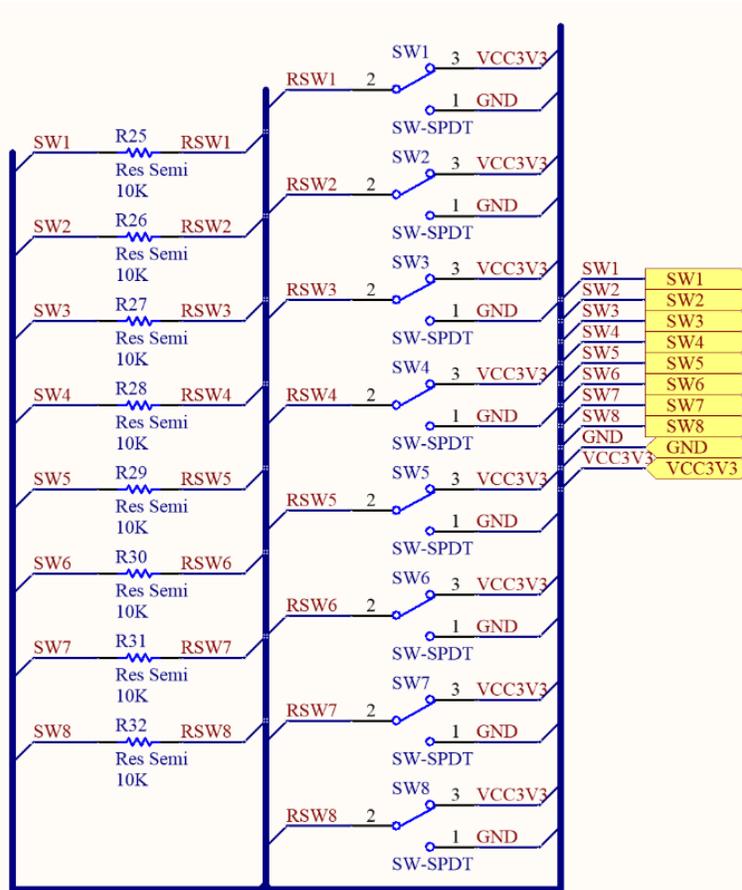


Figura 23. Esquemático detallado de la conexión del FPGA a los interruptores.

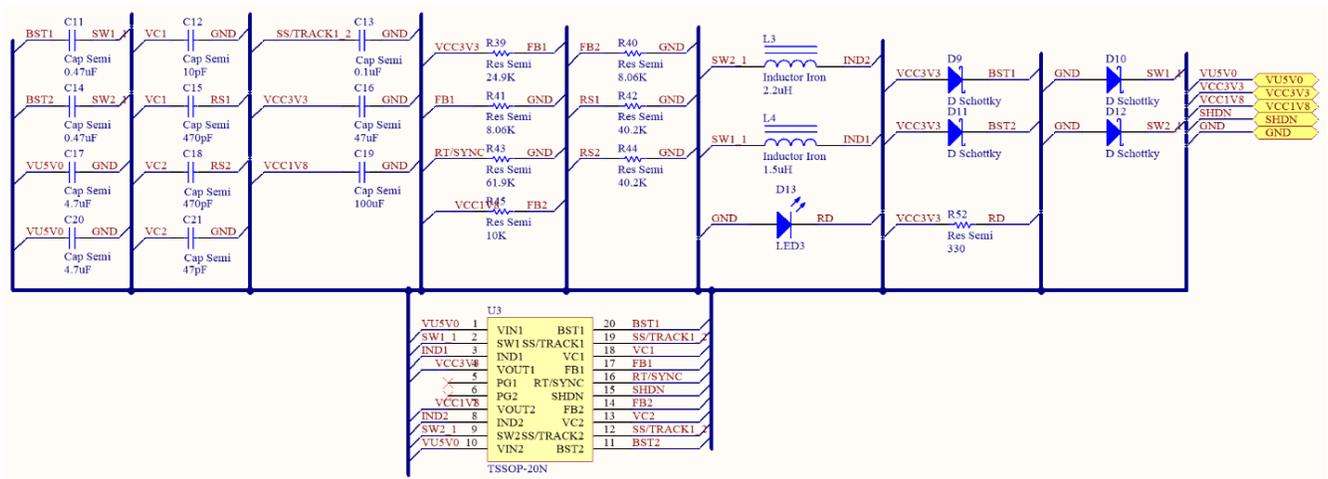


Figura 24. Esquemático detallado de la sección de alimentación.

El costo de este PCB fue de alrededor de 17,000 pesos (aproximadamente 950 dólares). Durante el ensamble nos encontramos con diversos retos interesantes debido a la presencia de un encapsulado BGA y encapsulados 0201, además de presentar componentes en las caras superior e inferior (top y bottom). En consecuencia, requerimos establecer una curva del incremento de precio por cada cm^2 del PCB de acuerdo al número de capas, como se observa en la figura 25:

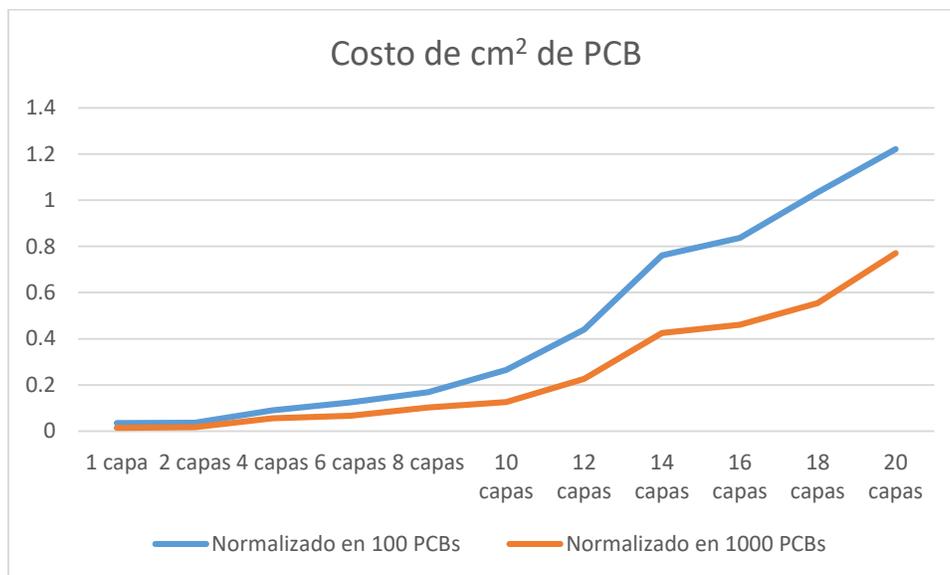


Figura 25. Costo (eje y, en dólares) del cm^2 de PCB respecto al número de capas (eje x).

4.2 Aplicación: Radiografía Digital Dental

El diagrama del sistema de Radiografía Digital Dental se muestra en la figura 26. Esta aplicación en particular es del tipo en el que un microprocesador/microcontrolador no puede utilizarse si dicho dispositivo no cuenta con un periférico integrado que permita leer el sensor, debido al protocolo particular a nivel de hardware.

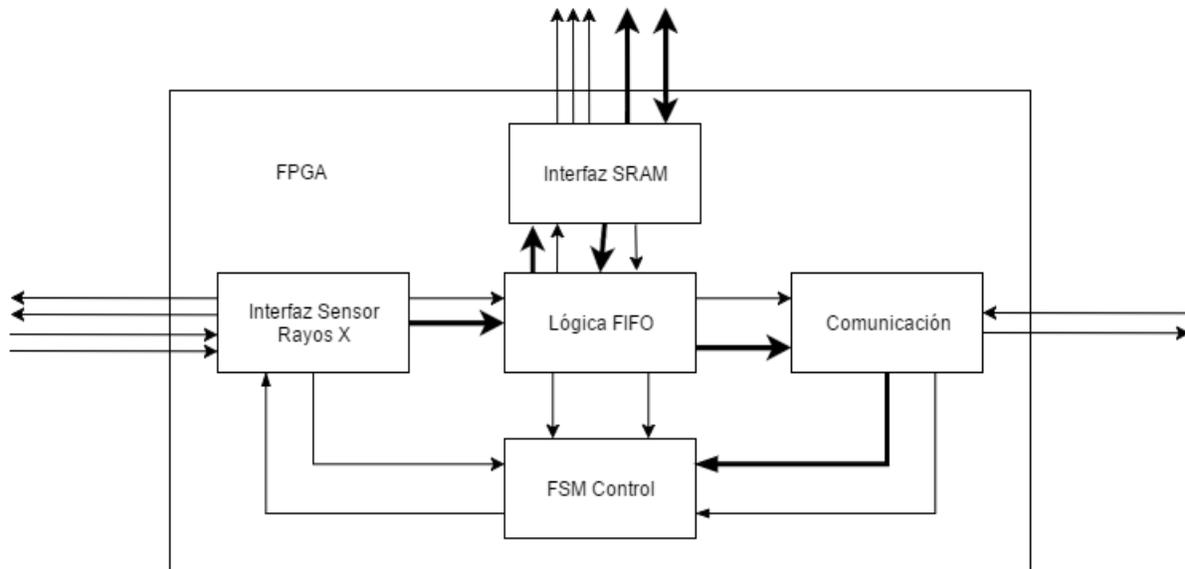


Figura 26. Diagrama a bloques del sistema de Radiografía Digital Dental embebido en el FPGA

La frecuencia máxima de operación de la interfaz del sensor es de 269.687 MHz, en tanto que la frecuencia de operación ideal del diseño es 80 MHz. El bloque de interfaz sólo requiere 110 LUT, 77 flip flops, 23 pines de conexión (4 ligadas a pines del FPGA). El diseño con todos los elementos integrados utiliza 833 LUT, 482 flip flops y 72 pines de conexión y puede operar a 180.016 MHz. Esto significa que podría ser posible utilizar un FPGA o algún otro dispositivo lógico programable muy pequeño que funcionara como lógica de pegamento entre el sensor y un procesador.

En las figuras 27 y 28, se muestran dos secuencias de datos de detección de rayos X. Se puede observar que en ninguno de los casos el valor detectado cae por debajo de 400. Esta radiación detectada es la llamada radiación de fondo. Los picos que sobresalen de la radiación de fondo son los rayos X emitidos por el cañón de rayos X. Es posible apreciar que el cañón de rayos X no entrega una salida máxima de manera inmediata. Una propuesta de uso para esta información es aplicar una integración de los picos de

radiación detectada para estimar la cantidad recibida por el sensor y generar una salida que corte el suministro de rayos X una vez que la radiación acumulada llegue a un límite preestablecido. De este modo se podría ajustar de manera automática la dosis de rayos X que recibe el paciente, al mismo tiempo que se obtiene radiografías con buena repetibilidad independientemente de la complejión del paciente.

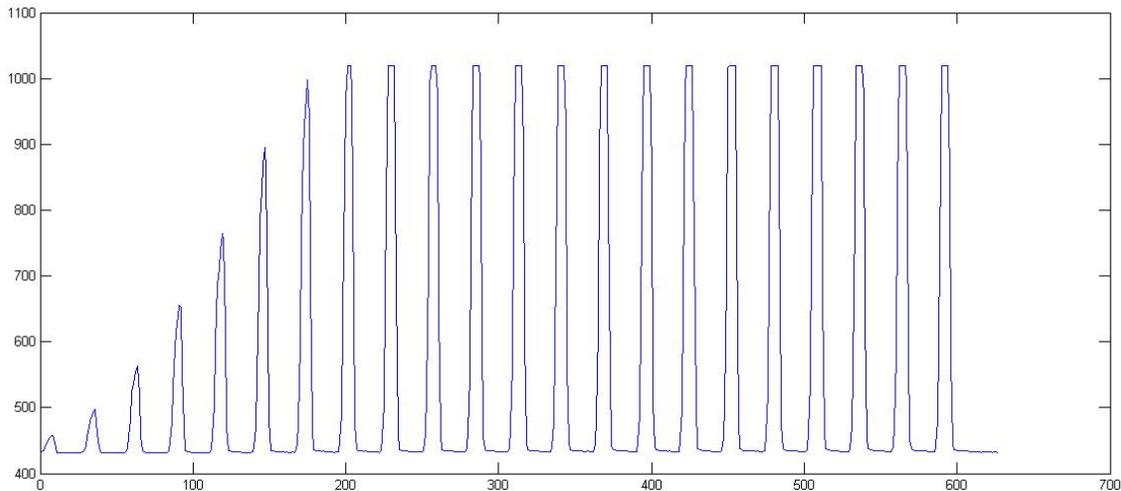


Figura 28. Captura de emisión de rayos X por parte del sensor, exposición de 300 ms.

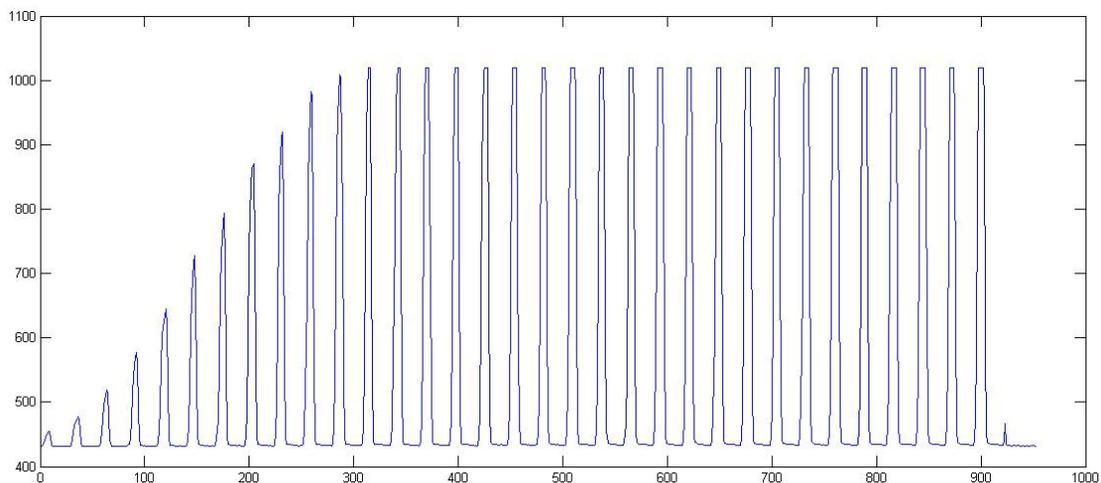


Figura 27. Captura de emisión de rayos X por parte del sensor, exposición de 500 ms.

Las figuras 29 a 32 muestran las imágenes obtenidas directamente del sensor comparadas con las imágenes aplicando los siguientes ajustes: conversión de 14 a 16 bits, ajuste de rango de datos, negativo. Estas operaciones pueden integrarse dentro del FPGA, sin un costo excesivo en términos de recursos. Sin embargo, ya que estas operaciones pueden afectar la información original contenida en la imagen, se prefirió enviar la imagen tal cual la entrega el sensor y aplicar el procesamiento de imagen posteriormente



Figura 29. Imagen sin procesar, 500 ms de exposición.

en la PC, conservando además una copia del original para pruebas con otros algoritmos de procesamiento.



Figura 30. Imagen postprocesada, 500 ms de exposición.



Figura 31. Imagen sin procesar, 300 ms de exposición.



Figura 30. Imagen postprocesada, 300 ms de exposición.

Dado el buen resultado alcanzado con el diseño en la tarjeta de prototipado, se exploró el desarrollo de una tarjeta específica para radiografía digital derivada del diseño previamente validado. Se tomaron en cuenta aspectos como la necesidad de que el sistema estuviera vivo al momento de encender, que fuera inherentemente robusto frente a la radiación y finalmente, que integrara todos los periféricos específicos en una sola tarjeta.

Para este diseño se seleccionó un FPGA basado en Flash, por las características específicas del diseño. El código VHDL fue portado sin cambios a las herramientas del fabricante del FPGA (Microsemi). El diseño resultante se muestra en las figuras 33 a 41. Los esquemáticos y los diseños de PCB se hicieron considerando lo probado previamente en la tarjeta de prototipado, lo que facilitó definir las conexiones.

Se comprobó además que el FPGA destino mantuviera la frecuencia de trabajo dentro del rango aceptable para que el sistema de radiografía permaneciera con el mismo comportamiento sin necesidad de aplicar cambios que pudieran afectar la portabilidad del código. Los resultados obtenidos mostraron que el sistema de radiografía en general podía trabajar a 91.5 MHz, frecuencia superior a los 80 MHz de frecuencia de operación ideal del sistema. En cuanto al porcentaje de utilización de recursos lógicos, el diseño requirió 69% del dispositivo (2,128 de 3,072 elementos de lógica).

Esta tarjeta fue desarrollada en dos capas, por lo que su costo fue considerablemente menor (alrededor de 120 pesos, poco más de 6 dólares) y contiene únicamente lo requerido para la aplicación final de radiografía digital y para el sistema digital diseñado para ese propósito.

El desarrollo de la Plataforma de Desarrollo Tecnológico de Sistemas Digitales Basada en FPGA derivó en la aplicación de una solicitud de protección de propiedad industrial ante el IMPI bajo la figura de Modelo de Utilidad, en tanto que la Plataforma de Radiografía Digital derivó en una solicitud de protección de propiedad industrial ante el IMPI bajo la figura de Patente.

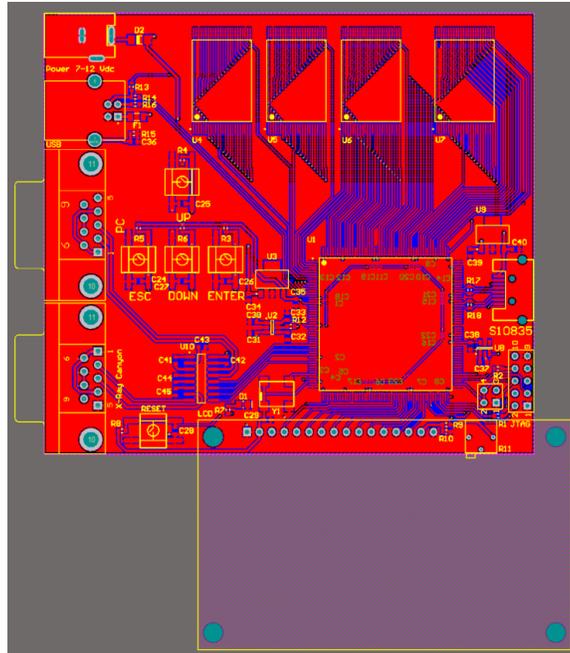


Figura 31. Cara superior de PCB para radiografía digital

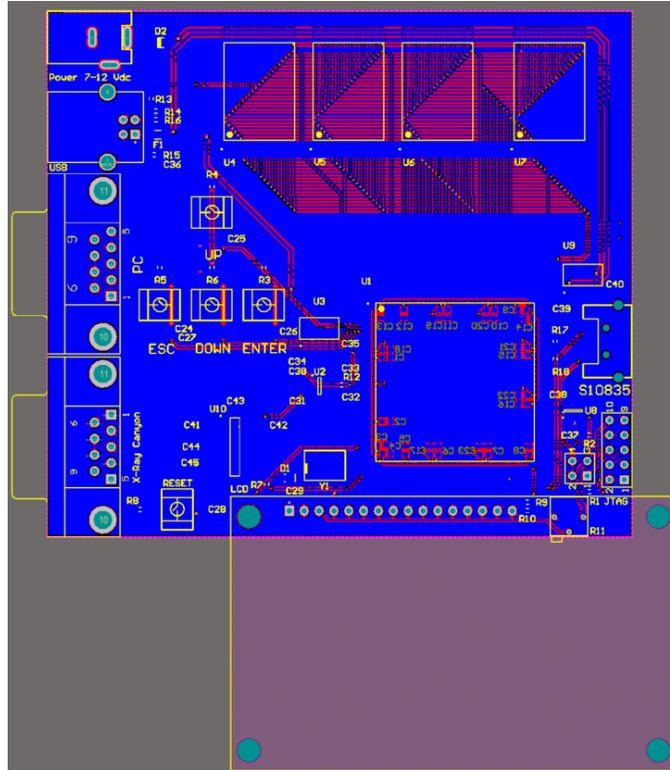


Figura 32. Cara inferior de PCB de radiografía digital

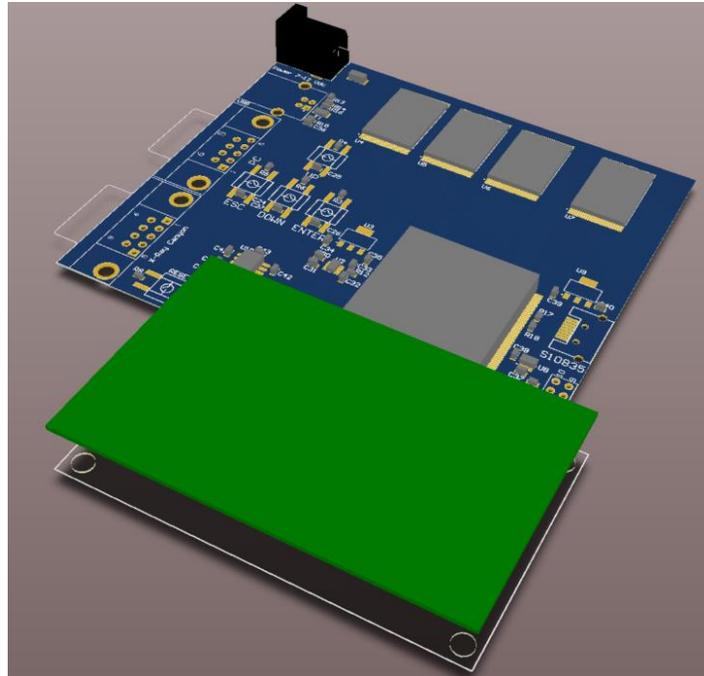


Figura 35. Vista 3D isométrica del PCB para radiografía digital

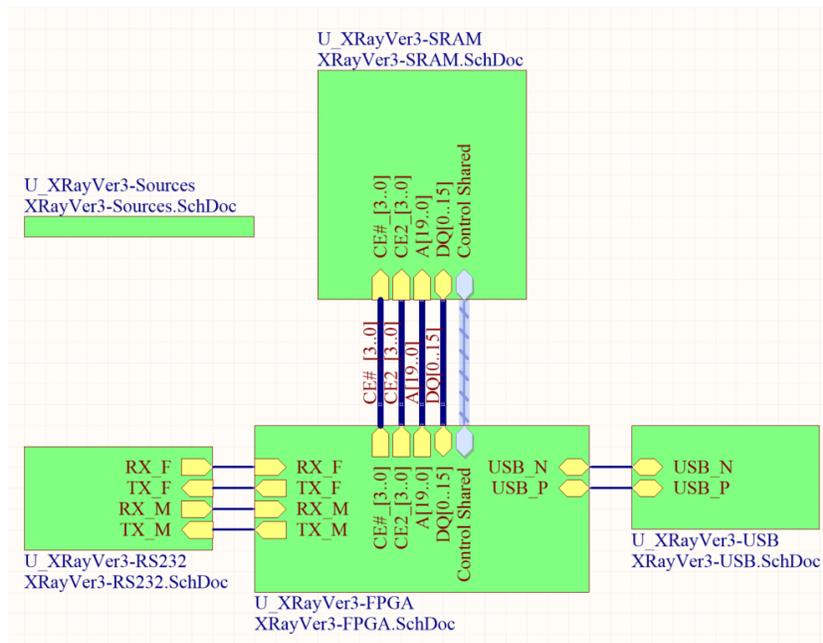


Figura 33. Diagrama esquemático general del PCB para radiografía digital.

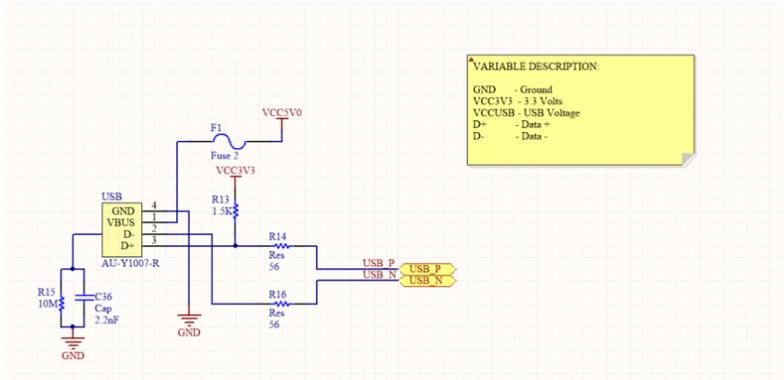


Figura 34. Esquemático conexión USB.

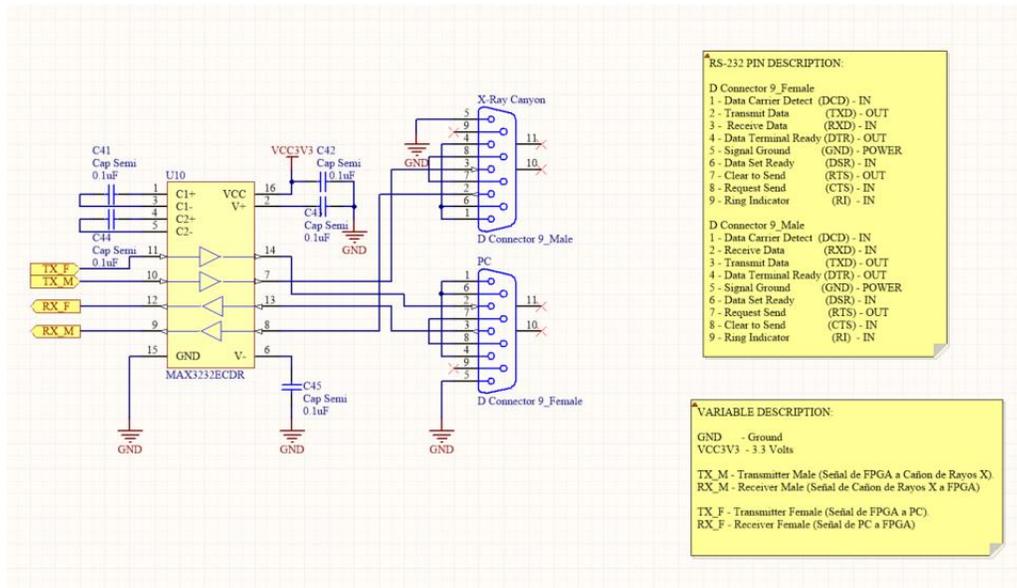


Figura 35. Esquemático de conexiones seriales a cañón de rayos X y auxiliar/depuración.

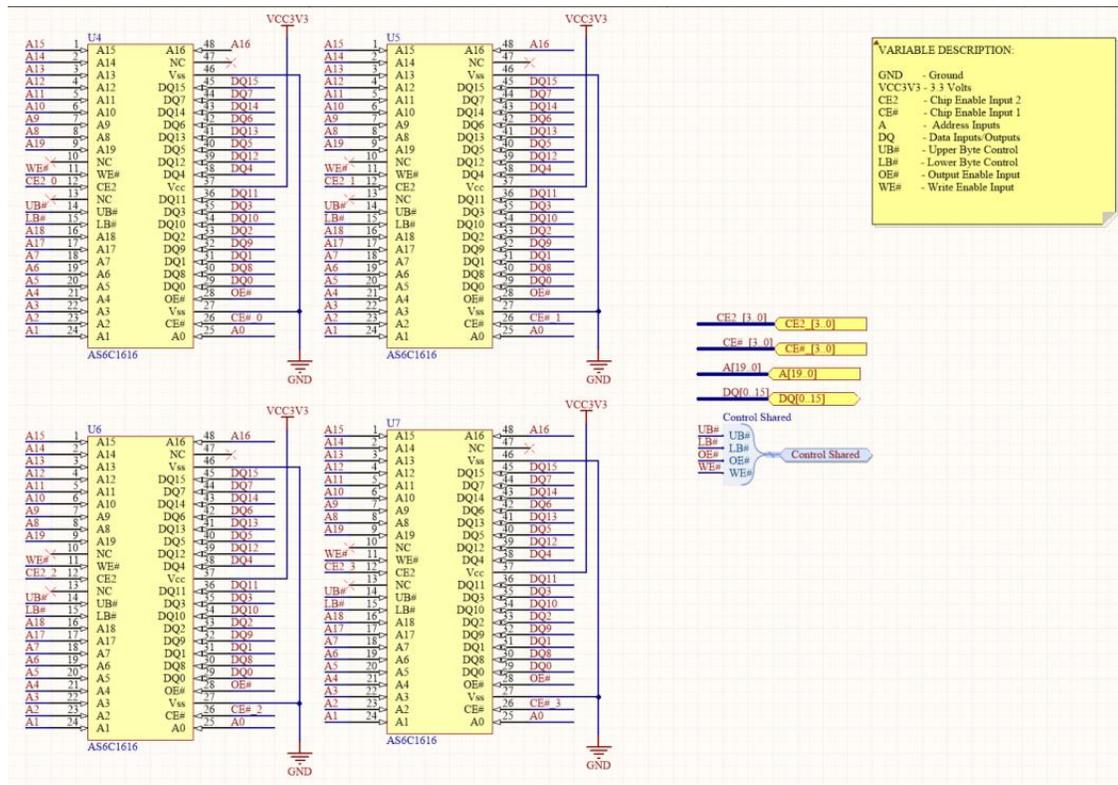


Figura 39. Esquemático de conexión de banco de memorias para procesamiento en tarjeta

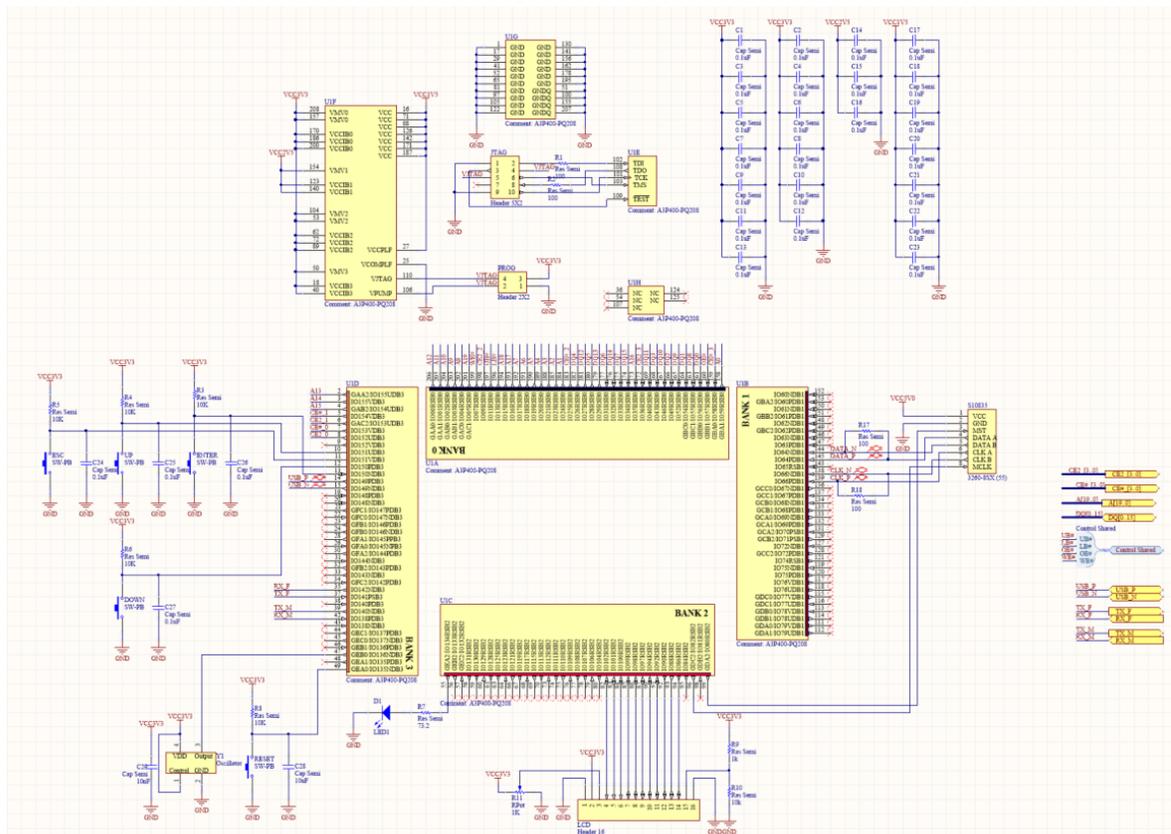


Figura 40. Esquemático de conexiones del FPGA.

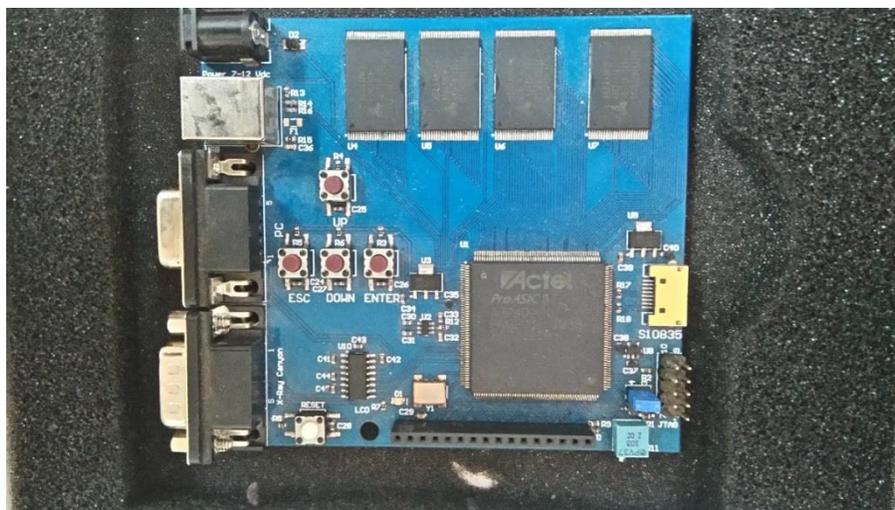


Figura 41. Tarjeta Prototipo Beta del sistema de radiografía digital

Capítulo 5: Conclusiones

Se desarrolló y evaluó el desempeño de una plataforma de prototipos de sistemas digitales con base en FPGA versátil, que incluye una amplia variedad de aplicaciones de sistemas digitales pertinentes para las empresas e instituciones educativas del país. Esta nueva Plataforma de Desarrollo Tecnológico de Sistemas Digitales Basada en FPGA nos permite establecer un punto de partida flexible para el desarrollo de soluciones específicas basadas en sistemas digitales, mejorando significativamente el tiempo y costo en el desarrollo de prototipos y pruebas de concepto, ya que sólo se requiere agregar los dispositivos periféricos que la aplicación necesite. Otro de los beneficios de esta plataforma es que permite trabajar con un prototipo funcional previo a la producción, ya que este sistema digital permite comprobar las condiciones de ruteo de pistas con las cuales el sistema funciona de manera óptima, y de esta manera programar un código fuente que no requiera cambios significativos al portarlo a un FPGA más adecuado o específico de la aplicación destino.

Las conclusiones más importantes derivadas de este trabajo son las siguientes:

- a) Se creó una plataforma de desarrollo tecnológico mexicana de alto nivel de integración y flexibilidad para el desarrollo de soluciones de ingeniería basada en sistemas digitales.
- b) La plataforma base tiene una gran versatilidad, ya que puede adaptarse a una gran variedad de aplicaciones sin impactar en su coste de desarrollo, dado que sólo se requiere agregar el costo de los periféricos requeridos.
- c) Las dimensiones de la plataforma permiten realizar conexiones con una diversidad de periféricos optimizando hasta el 100% de los recursos de entradas y salidas del FPGA.
- d) El código desarrollado permite que la plataforma integre una gran cantidad de periféricos y de bloques funcionales de código reutilizables para distintos desarrollos de prototipos de aplicaciones tecnológicas, que pueden ser distribuidos de manera libre o como bloques funcionales con propiedad intelectual. Esto permite al usuario de la plataforma un desarrollo acelerado de soluciones tecnológicas, ya que sólo se requiere integrar los bloques funcionales y agregar su procesamiento específico o diferenciador en la solución.

- e) De acuerdo a las pruebas de la plataforma con la aplicación de radiografía digital en particular, logramos desarrollar un prototipo alfa (prueba de concepto) y beta (funcional) con una calidad de imagen aceptable para ser procesada mediante un software específico de tratamiento de imágenes, para que el profesional de la odontología pueda realizar su diagnóstico.
- f) El desempeño tanto de la plataforma como de la aplicación final de radiografía fue completamente satisfactorio, tanto en uso de recursos como en frecuencia de trabajo y portabilidad del código utilizado.
- g) La versatilidad y las especificaciones del hardware utilizado en la plataforma le otorgan una gran flexibilidad en el manejo de información y desarrollo del software, en tanto que otras soluciones comerciales presentan diversos problemas de implementación ya que no cuentan con esta flexibilidad.
- h) El código desarrollado para la plataforma es completamente portable para el producto final, así como hacia cualquier otra plataforma comercial de FPGA.

5.1 Prospectiva de la investigación

La experiencia y el conocimiento adquirido en el diseño de esta Plataforma de Desarrollo Tecnológico de Sistemas Digitales Basada en FPGA, así como los buenos resultados obtenidos, me motivan a seguir trabajando en una segunda iteración con un diseño optimizado, de manera que resulte más viable en términos de costos y recursos invertidos. Con este proyecto, la meta es contribuir al desarrollo de herramientas tecnológicas nacionales que permitan generar valor agregado para las PyMEs de México y del mundo.

En cuanto al desarrollo de software, consideramos que la generación de una biblioteca de código depurado y probado facilitará la creación de nuevos proyectos e incrementará su alcance, al reducir el tiempo que se requiere para su implementación. A su vez, estos proyectos permitirán acelerar el desarrollo tecnológico de nuestro país y sentar las bases de proyectos subsecuentes.

Referencias

(Abowd, 2016) Gregory D. Abowd, “Beyond Weiser: From Ubiquitous to Collective Computing”, *Computer*, vol. 49, pp. 17-23, Enero 2016.

(Braun, 1874) Ferdinand Braun, “Ueber die Stromleitung durch Schwefelmetalle” (De la conducción de corriente de los sulfuros metálicos), *Annalen der Physik und Chemie*, vol. 153, pp. 556-563, 1874.

(Chan, 1994) Pak K. Chan, *A Field-Programmable Prototyping Board: XC4000 BORG User's Guide*, Abril 1994.

(Chekider et al., 2014) F. Chekired, A. Mellit, S. A. Kalogiroud, C. Larbese, “Intelligent maximum power point trackers for photovoltaic applications using FPGA chip: A comparative study”, *Solar Energy*, vol. 101, pp. 83-99, Marzo 2014.

(CMM, 2010) *Estudio prospectivo y plan estratégico de impulso a la electrónica digital para el desarrollo de aplicaciones innovadoras en empresas mexicanas*, Fumec - Secretaría de Economía, 2010.
http://www.cmm.org.mx/htdocs/estudio_prospectivo_electronica_digital.pdf

(Courtoy, 1998) Michel Courtoy, “Rapid System Prototyping for Real-Time Design Validation”, *Ninth International Workshop on Rapid System Prototyping*, Junio 1998.

(Digney et al., 2013) P. Digney, S. Griffiths, S. Miri, “Precise positioning in the mining sector: An estimate of the economic and social benefits of the use of augmented GNSS I in the mining sector”, *ACIL ALLEN Consulting*, 2013.

(Fanning, 2008) Brendan Fanning, “The establishment of a diagnostic reference level for intra-oral dental radiography in Co. Wicklow”, *Journal of the Irish Dental Association*, vol. 54, pp. 181-183, Agosto-Septiembre 2008.

(Ferrari y Maddox, 2010) Gary Ferrari y Cliff Maddox, “Sectional Design Standard for Rigid Organic Printed Boards”, *IPC-2222A Report*, IPC-Association Connecting Electronics Industries, 2010.

(Ferrari y Maddox, 2012) Gary Ferrari y Cliff Maddox, “Generic Standard on Printed Board Design”, *IPC-2221B Report*, IPC-Association Connecting Electronics Industries, 2012.

(Fröning et al., 2006) Holger Fröning, Mondrian Nüssle, David Slognat, Heiner Litz, Ulrich Brüning, “The HTX-Board: A Rapid Prototyping Station”, *Proceedings of the 3rd annual FPGA World Conference*, 2006.

(Garamendi y Landa, 2010) Pedro Manuel Garamendi González y María Irene Landa Tabuyo, “Determinación de la edad mediante la radiología”, *Revista Española de Medicina Legal*, pp. 3-13, 2010.

(González et al., 2012) Luciano Gonzalez, Eliseo Vano, Ramon Fernández, Viviana Ziraldo, Jaime Delgado, Victor Delgado, Javier Moro, Carlos Ubeda, “Evaluating Phantom Image Quality Parameters to Optimize Patient Radiation Dose in Dental Digital Radiology”, *Radiation Protection Dosimetry*, vol. 151, pp. 95-101, Oxford University Press, 2012.

(Harandi et al., 2011) Azam Amini Harandi, Hossein Pourghassem, Hamid Mahmoodian, “Upper and Lower Jaw Segmentation in Dental X-ray Image Using Modified Active Contour”, *International Conference on Intelligent Computation and Bio-Medical Instrumentation*, pp. 124-127, IEEE, 2011.

(Héder, 2017) Mihály Héder, “From NASA to EU: the evolution of the TRL scale in Public Sector Innovation”, *The Innovation Journal: The Public Sector*, vol. 22, 2017.

(Henke et al., 2014) Karsten Henke, Galina Tabunshchyk, Heinz-Dietrich Wuttke, Tobias Vietzke, Steffen Ostendorff, “Using Interactive Hybrid Online Labs for Rapid Prototyping of Digital Systems”, *11th International Conference on Remote Engineering and Virtual Instrumentation*, pp. 61-66, Febrero 2014.

(Herrera et al., 2014) Luis Herrera, Cong Li, Xiu Yao, Jin Wang, “FPGA-Based Detailed Real-Time Simulation of Power Converters and Electric Machines for EV HIL Applications”, *IEEE Transactions on Industry Applications*, vol. 51, pp. 1702-1712, Marzo-Abril 2015.

(Hui et al., 2012) Hui Li, Guoxia Sun, Huiqiang Sun, Wei Liu, “Watershed algorithm based on morphology for dental X-ray images segmentation”, *8th International Conference on Signal Processing (ICSP)*, pp. 877-880, IEEE, 2012.

(Hussain et al., 2014) Tassadaq Hussain, Oscar Palomar, Osman Unsal, Adrian Cristal, Eduard Ayguadé, Mateo Valero, "Advanced Pattern based Memory Controller for FPGA based HPC applications", *International Conference on High Performance Computing & Simulation (HPCS)*, Julio 2014.

(Ilić y Stojanović, 2010) Dragan V. Ilić y Ljiljana S. Stojanović, "Application of radiovisiography (digital radiology) in dental clinical practice", *Vojnosanitetski Pregled*, vol. 69, pp 81-84, 2010.

(IPC, 2004) IPC, "Acceptability of printed board", *IPC-A-600G Report*, IPC-Association Connecting Electronics Industries, 2004.

(Johnson et al., 2015) Anju P. Johnson, Rajat Subhra Chakraborty, Debdeep Mukhopadhyay, "A PUF-Enabled Secure Architecture for FPGA-Based IoT Applications", *IEEE Transactions on Multi-Scale Computing Systems*, vol. 1, pp. 110-122, Abril-Junio 2015.

(Johnson y Goetz, 1986) D.W. Johnson y W.A. Goetz, "Patient exposure trends in medical and dental radiology", *Health Physics: The Radiation Safety Journal*, vol. 50, Lippincott Williams & Wilkins, Enero 1986.

(Johnston et al., 2011) James Johnston, Jerrey B. Killion, Beth Vealé, Robert Comello, "U.S. Technologists' Radiation Exposure Perceptions and Practices", *Journal of the American Society of Radiologic Technologists*, pp. 311-320, Marzo-Abril 2011.

(Kaeslin, 2008) Hubert Kaeslin, *Digital Integrated Circuit Design. From VLSI Architectures to CMOS Fabrication*, Cambridge University Press, 2008.

(Kang et al., 2007) Dong-Soo Kang, Soo Yun Hwang, Kyoung-Son Jhang, Kang Yi, "A Low Cost and Interactive Rapid Prototyping Platform for Digital System Design Education", *IEEE International Conference on Microelectronic Systems Education*, Junio 2007.

(Kasunori, 2012) Kazunori Yoshiura, "Image quality assessment of digital intraoral radiography - perception to caries diagnosis", *Japanese Dental Science Review*, vol. 48, pp. 42-47, Elsevier, 2012.

(Kedrin, 2011) Ivan Kedrin, “New Product Development Process. Building & Launching Products the Right Way”, 2011. <https://nyccto.wordpress.com/2011/06/10/new-product-development-process-building-launching-products-the-right-way/>

(Kolehmainen et al., 2006) V. Kolehmainen, A. Vanne, S. Siltanen, S. Jarvenpaa, J.P. Kaipio, M. Lassas, Martti Kalke, “Parallelized bayesian inversion for three-dimensional dental X-ray imaging”, *Transactions on Medical Imaging*, pp. 218-228, IEEE, 2006.

(Kumar et al., 2014) Subir Kumar Sarkar, Asish Kumar De, Souvik Sarkar, *Foundation of Digital Electronics and Logic Design*, Pan Stanford Publishing, 2014.

(LaDou, 2006) Joseph LaDou, “Printed circuit board industry”, *International Journal of Hygiene and Environmental Health*, vol. 209, pp. 211-219, 2006.

(Lattice Semiconductor Corporation, 2010) Lattice Semiconductor Corporation, “GAL®22V10 Device Datasheet”, Septiembre 2010.

(Litz et al., 2009) Heiner Litz, Holger Froning, Maximilian Thilmer, Ulrich Bruning, “An FPGA based Verification Platform for HyperTransport 3.x”, *International Conference on Field Programmable Logic and Applications*, pp. 631-634, Septiembre 2009.

(Maldonado et al., 2013) Yazmin Maldonado, Oscar Castillo, Patricia Melin, “Particle swarm optimization of interval type-2 fuzzy systems for FPGA applications”, *Applied Soft Computing*, vol. 13, pp. 496-508, Enero 2013.

(Markets and Markets, 2017) Markets and Markets, “FPGA Market by Technology (SRAM, Antifuse, Flash), Node Size (Less than 28 nm, 28-90 nm, More than 90 nm), Configuration (High-End FPGA, Mid-Range FPGA, Low-End FPGA), Vertical (Telecommunications, Automotive), and Geography - Global Forecast to 2023”, Octubre 2017.

(Marques et al., 2013) André Canal Marques, José-María Cabrera, Célia de Fraga Malfatti, “Printed circuits boards: A review on the perspective of sustainability”, *Journal of Environmental Management*,

vol. 131, pp. 298-306, 2013.

(Martin, 2000) Kenneth William Martin, *Digital Integrated Circuit Design*, Oxford University Press, 2000.

(Maxfield, 2004) Clive Maxfield, *The Design Warrior's Guide to FPGAs*, Elsevier-Newnes, 2004.

(Maxfield, 2009) Clive Maxfield, "Printed Circuits Board (PCBs)", *Bebop to the Boolean Boogie*, vol. 3, pp. 251-276, 2009.

(Moore, 1965) Moore, Gordon E., "Cramming More Components onto Integrated Circuits", *Electronics*, pp. 114-117, April 1965.

(Morris y Ciletti, 2007) M. Morris Mano y Michael D. Ciletti, *Digital Design*, 4^a ed., Pearson Education, 2007.

(Morris, 2012) Kevin Morris, "(Moore's) Law of diminishing return. When do we have enough?", *Electronic Engineering Journal*, 2012. <http://www.eejournal.com/article/20120327-diminishing/>

(Morris, 2014a) Kevin Morris, "The Sun sets on Moore's Law. Are FPGAs harbingers of a new era?", *Electronic Engineering Journal*, 2014a. <http://www.eejournal.com/article/20140805-mooreslaw/>

(Morris, 2014b) Kevin Morris, "The Next Moore's Law. What if it happened again?", *Electronic Engineering Journal*, 2014b. <http://www.eejournal.com/article/20140812-nextmoore/>

(Morris, 2015) Kevin Morris, "The Back Side of Moore's Law. Surfing the new innovation wave". *Electronic Engineering Journal*. 2015, <http://www.eejournal.com/article/20151202-backside/>

(Nurmi, 2007) Jari Nurmi, *Processor Design. System-On-Chip Computing for ASICs and FPGAs*, Springer, 2007.

(Oprea et al., 2008) Stefan Oprea; Costin Marinescu; Ioan Lita; Mariana Jurianu; Daniel Alexandru

Visan; Ion Bodgan Cioc; “Image Processing Techniques used for Dental X-Ray Image Analysis. Non-destructive testing, picture acquisition methods”, *Electronics Technology*, pp. 125-129, IEEE, 2008.

(Parnell y Mehta, 2003) Karen Parnell y Nick Mehta, *Programmable Logic Design Quick Start Hand Book*, 4a ed., Xilinx, Junio 2003.

(Patanachai et al., 2010) Nakintorn Patanachai; Nongluk Covavisaruch; Chanjira Sinthanayothin; “Wavelet Transformation for Dental X-ray Radiographs Segmentation Technique”, *Eighth International Conference on ICT and Knowledge Engineering*, vol. 41, pp. 103-106, IEEE, 2010.

(Po-Whei et al., 2012) Po-Whei Huang; Phen-Lan Lin; Chun-Hung Kuo; Y. S. Cho, “An Eective Tooth Isolation Method for Bitewing Dental X-ray Images”, *Proceedings of the 2012 International Conference on Machine Learning and Cybernetics*, pp. 1814-1820, IEEE, Julio 2012.

(Raju y Modi, 2011) Jincy Raju y Chintan K. Modi, “A Proposed Feature Extraction Technique for Dental X-Ray Images Based on Multiple Features”. *International Conference on Communication Systems and Network Technologies (CSNT)*, pp. 545-549, IEEE, 2011.

(Ricci y Le-Huy, 2002) Francesco Ricci, Hoang Le-Huy, “An FPGA-Based Rapid Prototyping Platform for Variable-Speed Drives”, *28th Annual Conference of the Industrial Electronics Society*, pp. 1156-1161, IEEE, Noviembre 2002.

(Rowe, 2010) Teresa Rowe, “Acceptability of Electronic Assemblies (IPC-A-610)”, IPC-Association Connecting Electronics Industries, 2010.

(Salman y Friedman, 2012) Emre Salman y Eby G. Friedman, *High Performance Integrated Circuit Design*", McGraw Hill Profesional, 2012.

(Sandige y Sandige, 2012) Richard S. Sandige y Michael L. Sandige, *Fundamentals of Digital and Computer Design with VHDL*, McGraw Hill, 2012.

(Severance y Lemieux, 2012) A. Severance y G. Lemieux, “VENICE: A compact vector processor for

FPGA applications”, *International Conference on Field-Programmable Technology (FPT)*, IEEE, Diciembre 2012.

(Shimohashi et al., 2009) Koki Shimohashi, Xiaoying Zhou, Julie M. Schoenung, “A test-rework process yield performance model for estimation of printed wiring board assembly cost”, *International Journal of Production Economics*, vol. 119, pp. 161-173, 2009.

(Siddiqui et al., 2014) Fahad Manzoor Siddiqui, Matthew Russell, Burak Bardak, Roger Woods, Karen Rafferty, “IPPro: FPGA based image processing processor”, *Workshop on Signal Processing Systems (SiPS)*, IEEE, Octubre 2014.

(Siebert y Morin, 2011) J. Anthony Siebert y Richard L. Morin, “The standardized exposure index for digital radiography: an opportunity for optimization of radiation dose to the pediatric population”, *Pediatric Radiology*, vol. 41, pp. 573-581, Springer, 2011.

(Siebert, 2004) J. Anthony Siebert, “Tradeoffs between image quality and dose”, *Pediatric Radiology*, vol. 34, pp. 183-185, Springer-Verlag, 2004.

(Smith, 1997) Michael John Sebastian Smith, *Application-Specific Integrated Circuits*, Addison-Wesley Professional, 1997.

(Soonil et al., 2005) Soonil Hong; Haijo Jung; Kee-Deog Kim; Su-Gil So; Jong-Ho Kim; Sun Kook Yoo; Hyung Sik Yoo; Hee-Joung Kim, “Development and Evaluation of a CMOS Sensor-Based Digital Intra-Oral Radiographic System”, *Transactions on Nuclear Science*, pp. 256-261, IEEE, 2005.

(Southard y Southard, 1996) T.E. Southard y K.A. Southard, “Detection of simulated osteoporosis in maxillae using radiographic texture analysis”. *Transactions on Biomedical Engineering*, pp. 123-132, IEEE, 1996.

(Srivastava y Brodersen, 1991) Mani B. Srivastava, Robert W. Brodersen, “Rapid-Prototyping of Hardware and Software in a Unified Framework”, *IEEE International Conference on Computer-Aided Design*, pp. 152-155, IEEE, Noviembre 1991.

(Tavana y Dinavahi, 2015) Nariman Roshandel Tavana y Venkata Dinavahi, “A General Framework for FPGA-Based Real-Time Emulation of Electrical Machines for HIL Applications”, *Transactions on Industrial Electronics*, vol. 62, pp. 2041-2053, IEEE, Abril 2015.

(Toft y Nannarelli, 2014) Jakob Kenn Toft y Alberto Nannarelli, “Energy efficient FPGA based hardware accelerators for financial applications”, *NORCHIP*, Octubre 2014.

(Tsai, 2012) Tsung-Nan Tsai, “Thermal parameters optimization of a reflow soldering profile in printed circuit board assembly: A comparative study”, *Applied Soft Computing*, vol. 12, pp. 2601-2613, 2012.

(Uffmann y Schaefer-Prokop, 2009) Martin Uffmann y Cornelia Schaefer-Prokop, “Digital radiography: The balance between image quality and required radiation dose”, *European Journal of Radiology*, vol. 72, pp. 202-208, Elsevier, 2009.

(Van der Wal et al., 2015) Gooitzen Van der Wal, David Zhang, Indu Kandaswamy, Jim Marakowitz, Kevin Kaighn, Joe Zhang, Sek Chai, “FPGA Acceleration for Feature Based Processing Applications”, *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pp. 43-48, IEEE, Junio 2015.

(Williams, 2010) Tim Williams, “Passive components”, *The circuit designer’s companion*, vol. 2, pp. 70-108, Elsevier, 2010.

(Williams, 2017) Lawrence Williams, “Enabling the Future for Electronics”, *Ansys Advantage*, vol. 11, Ansys, 2017.

(Xilinx, 2008) *Programmable Logic Design Quick Start Guide*, Xilinx, Mayo 2008.

(Zhang et al., 2015) Chen Zhang, Peng Li, Guangyu Sun, Yijin Guan, Bingjun Xiao, Jason Cong, “Optimizing FPGA-based Accelerator Design for Deep Convolutional Neural Networks”, *Proceedings of the 2015 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, pp. 161-170, Febrero 2015.

Anexos

Nomenclaturas

ASIC – Application Specific Integrated Circuit / Circuito Integrado de Aplicación Específica.

CCD – Charge-Coupled Device / Dispositivos de Carga Acoplada.

CMOS – Complementary Metal-Oxide Semiconductor / Semiconductor de Óxido-Metal Complementario.

ENIAC – Electronic Numerical Integrator And Computer / Computador e Integrador Numérico Electrónico.

FPGA – Field Programmable Gate Array / Arreglo de Compuertas Programable en Campo.

FSM – Finite State Machine / Máquina de Estado Finito.

IPC – Institute for Printed Circuits, Institute for Interconnecting and Packaging Electronic Circuits. Actualmente es IPC – Association Connecting Electronics Industries. Es una asociación de industrias electrónicas que estandariza los requerimientos de producción y ensamble de electrónicos.

PCB – Printed Circuit Board / Tarjeta de Circuito Impreso.

SEDEAM – Sistemas Embebidos, Diseño Electrónico Avanzado, y Microsistemas.

VHDL – Very High Speed Integrated Circuit Hardware Description Language.

Códigos VHDL

--control del sensor de rayos x

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.NUMERIC_STD.ALL;

entity control_cmos_beta is
  Port (clk : in STD_LOGIC;
        rst : in STD_LOGIC;
        ini_ext : in std_logic;
        ini_cap : in std_logic;
        ini_trans : in std_logic;
        ini_ack : in STD_LOGIC;
        t_val : in std_logic;
        captura_ack : in std_logic;
        ft : in std_logic;
        dv_pc : in STD_LOGIC;
        dato_pc : in STD_LOGIC_VECTOR (7 downto 0);
        dv_cmos : in STD_LOGIC;
        env_dat,captura : out STD_LOGIC;
        t_ack : in STD_LOGIC;
        dae : out STD_LOGIC_VECTOR (7 downto 0);
        iniciar : out STD_LOGIC;
        fin_ack : out std_logic;
        fin_cmos : in STD_LOGIC;
        l_ack,e_ack,lf : in std_logic;
        e,f : in std_logic;
        leer,escribir : out std_logic;
        rad_val : in std_logic_vector(9 downto 0);
        estado : out Std_logic_vector(1 downto 0);
        dato_de_ram : in std_logic_vector(15 downto 0)--);
end control_cmos_beta;
```

architecture Behavioral of control_cmos_beta is

```
type edos_adq is
(espera,esperar_ack,iniciar_cmos,guardar,leer_m,leer_m1,leer_m2,leer_m3,leer_m4,leer_m5,fin,
debug_ser,debug_ser1,leer_m6,leer_m7,pre_guardar,pre_guardar2,integ,integ2,integ3,integ4,integ5);
signal edo_beta: edos_adq;
signal ini_trans2,it,it2:std_logic;
signal iv: std_logic_vector(9 downto 0);
signal cta_mem: std_logic_vector(14 downto 0);
signal r: std_logic_vector(15 downto 0);
begin
```

```

process(clk,rst)
begin
    if rst= '1' then
        it <= '1';
        it2 <= '1';
        ini_trans2 <= '0';
    elsif rising_edge(clk) then
        it <= ini_trans;
        it2 <= it;
        ini_trans2 <= '0';
        if it2='0' and it='1' then
            ini_trans2 <= '1';
        end if;
    end if;
end process;

```

```

ctrl:process(clk,rst)
begin
    if rst='1' then
        edo_beta <= espera;
        r <= (others => '0');
        iv <= (others => '0');
        iniciar <= '0';
        dae <= (others => '0');
        env_dat <= '0';
        fin_ack <= '0';
        leer <= '0';
        escribir <= '0';
        captura <= '0';
        estado <= "00";
        cta_mem <= (others=> '0');
    elsif rising_edge(clk) then
        case edo_beta is
            when espera=>
                iniciar <= '0';
                fin_ack <= '0';
                captura <= '0';
                dae <= (others => '0');
                r <= (others => '0');
                iv <= (others => '0');
                cta_mem <= (others => '0');
                env_dat <= '0';
                leer <= '0';
                escribir <= '0';
                estado <= "00";
                if ini_ext='0' then
                    edo_beta <= iniciar_cmos;
                    estado <= "10";
                end if;
        end case;
    end if;
end process;

```

```

if dv_pc='1' then
    edo_beta <= debug_ser;
end if; -- debug
if ini_trans2 = '1' then
    edo_beta <= integ; estado <= "10";
end if;
when iniciar_cmos =>
iniciar <= '1';
estado <= "01";
if ini_ack='1' then
    edo_beta <= pre_guardar;
end if;
when pre_guardar =>
if ini_cap='0' then
    edo_beta <= pre_guardar2; captura <= '1';
end if;
when pre_guardar2 => estado <= "01";
if captura_ack = '1' then
    edo_beta <= guardar; captura <= '0';
end if;
when guardar =>
estado <= "10";
if dv_cmos = '1' then
    escribir <= '1';
end if;
if e_ack = '1' then
    escribir <= '0';
end if;
if f = '1' or fin_cmos = '1' then
    edo_beta <= leer_m; fin_ack <= '1'; escribir <= '0';
end if;
when leer_m => leer <= '1'; fin_ack <= '0';
estado <= "11";
if l_ack='1' then
    edo_beta <= leer_m1; leer <= '0';
end if;
if e='1' then
    edo_beta <= fin;
end if;
when leer_m1 =>
if lf='1' then
    r <= dato_de_ram; edo_beta <= leer_m2;
end if;
when leer_m2 => env_dat <= '1'; dae <= r(15 downto 8);
if t_ack = '1' then
    edo_beta <= leer_m3; env_dat <= '0';
end if;
when leer_m3 =>
if ft='1' then

```

```

        edo_beta <= leer_m4;
    end if;
    when leer_m4 => env_dat <= '1'; dae <= r(7 downto 0);
    if t_ack = '1' then
        edo_beta <= leer_m5; env_dat <= '0';
    end if;
    when leer_m5 =>
    if ft='1' then
        edo_beta <= leer_m;
    end if;
    when debug_ser => env_dat <= '1'; dae <= dato_pc;
    if t_ack = '1' then
        edo_beta <= debug_ser1; env_dat <= '0';
    end if;
    when debug_ser1 =>
    if ft='1' then
        edo_beta <= espera;
    end if;
    when fin => edo_beta <= espera;
    when integ =>
    if t_val = '1' then
        escribir<='1';cta_mem<=std_logic_vector(unsigned(cta_mem)+1);
    end if;
    if e_ack = '1' then
        escribir <= '0';
        if cta_mem="0000000000000000" then
            edo_beta <= leer_m;
        end if;
    end if;
    if f = '1' then
        edo_beta <= leer_m; escribir <= '0';
    end if;
    when integ2 => env_dat <= '1'; dae <= "000000" & iv(9 downto 8);
    if t_ack = '1' then
        edo_beta <= integ3; env_dat <= '0';
    end if;
    when integ3 =>
    if ft='1' then
        edo_beta <= integ4;
    end if;
    when integ4 => env_dat <= '1'; dae <= iv(7 downto 0);
    if t_ack = '1' then
        edo_beta <= integ5; env_dat <= '0';
    end if;
    when integ5 =>
    if ft='1' then
        edo_beta <= integ;
    end if;
    when others => edo_beta <= espera;

```

```

                end case;
            end if;
        end process;

```

```

end Behavioral;

```

--Bloque conector de los elementos que componen el Sistema de radiografía digital

```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.NUMERIC_STD.ALL;

```

```

entity sensor_rayos_x_cmos_beta is
    Port ( clk : in STD_LOGIC;
          rst : in STD_LOGIC;
          dato_p,dato_n : in STD_LOGIC;
          clk_cmos_p,clk_cmos_n : in STD_LOGIC;
          clk_master : out STD_LOGIC;
          start_cnv : out STD_LOGIC;
          rx : in STD_LOGIC;
          tx : out STD_LOGIC;
          ce0,ce20,ce1,ce21,ce2,ce22,ce3,ce23: out std_logic;
          we: out std_logic;
          oe: out std_logic;
          lb: out std_logic;
          ub: out std_logic;
          dir: out std_logic_vector(19 downto 0);
          datos_ram: inout std_logic_vector(15 downto 0);
          sel: in std_logic;
          en: out std_logic;
          rs: out std_logic;
          w: out std_logic;
          datos: inout std_logic_vector(7 downto 0);
          str_ext: in std_logic; --up
          str_int: in std_logic; --esc
          str_cap: in std_logic; -- down
          alive: out std_logic);
end sensor_rayos_x_cmos_beta;

```

```

architecture Behavioral of sensor_rayos_x_cmos_beta is

```

```

    component modulo_cmos is
        Port ( clk : in STD_LOGIC;
              rst : in STD_LOGIC;
              ini,str_env : in STD_LOGIC;
              fin_ack: in std_logic;
              dato_p,dato_n : in STD_LOGIC;
              clk_cmos_p,clk_cmos_n : in STD_LOGIC;

```

```

clk_master : out STD_LOGIC;
start_cnv,t_val,env_ack: out STD_LOGIC;
dato_val : out std_logic;
fin : out std_logic;
dt: out std_logic_vector(9 downto 0);
dato_sen : out STD_LOGIC_VECTOR (13 downto 0);
d_ac : out std_logic_vector(15 downto 0);
nc: out std_logic_vector(10 downto 0);
nr: out std_logic_vector(10 downto 0);
rad_prom: out std_logic_vector(13 downto 0)
);
end component;

```

```

component control_cmos_beta is
  Port ( clk : in STD_LOGIC;
        rst : in STD_LOGIC;
        ini_ext : in std_logic;
        ini_cap : in std_logic;
        ini_trans: in std_logic;
        ini_ack : in STD_LOGIC;
        t_val : in std_logic;
        captura_ack : in std_logic;
        ft: in std_logic;
        dv_pc : in STD_LOGIC;
        dato_pc : in STD_LOGIC_VECTOR (7 downto 0);
        dv_cmos : in STD_LOGIC;
        env_dat,captura : out STD_LOGIC;
        t_ack : in STD_LOGIC;
        dae : out STD_LOGIC_VECTOR (7 downto 0);
        iniciar : out STD_LOGIC;
        fin_ack : out std_logic;
        fin_cmos : in STD_LOGIC;
        l_ack,e_ack,lf: in std_logic;
        e,f: in std_logic;
        leer,escribir: out std_logic;
        rad_val: in std_logic_vector(9 downto 0);
        estado: out Std_logic_vector(1 downto 0);
        dato_de_ram: in std_logic_vector(15 downto 0)
  );
end component;

```

```

component bloque_tran_re is
  port(
    clk : in STD_LOGIC;
    rst : in STD_LOGIC;
    ini_t: in STD_LOGIC;
    rx : in STD_LOGIC;
    d_a_e: in std_logic_vector(7 downto 0);
    tx: out std_logic;
  );
end component;

```

```

    fin_t: out std_logic;
    bsy_t: out std_logic;
    dato_valido : out STD_LOGIC;
    dato_r : out STD_LOGIC_VECTOR(7 downto 0)
);
end component;

component FIFO_Memoria_RAM is
port (
    clk: in std_logic;
    rst: in std_logic;
    escribir: in std_logic;
    leer: in std_logic;
    dato_cmos : in std_logic_vector(13 downto 0);
    empty, full: out std_logic;
    escribir_ack, escribir_fin: out std_logic;
    leer_ack, leer_fin: out std_logic;
    ce0,ce20,ce1,ce21,ce2,ce22,ce3,ce23: out std_logic;
    we: out std_logic;
    oe: out std_logic;
    lb: out std_logic;
    ub: out std_logic;
    datos_de_ram: out std_logic_vector(15 downto 0);
    dir: out std_logic_vector(19 downto 0);
    datos_ram: inout std_logic_vector(15 downto 0)
);
end component;

component LCD is
port(
    clk: in std_logic;
    rst: in std_logic;
    inst: in std_logic_vector(1 downto 0);
    param: in std_logic_vector(7 downto 0);
    ini_rdy: out std_logic;
    inst_ack: out std_logic;
    inst_done: out std_logic;
    en: out std_logic;
    rs: out std_logic;
    w: out std_logic;
    datos: inout std_logic_vector(7 downto 0)
);
end component;

component fsm_ctrl_lcd is
port(
    clk: in std_logic;
    rst: in std_logic;
    sel: in std_logic;

```

```

ini_rdy: in std_logic;
inst_ack: in std_logic;
inst_done: in std_logic;
nc: in std_logic_vector(10 downto 0);
nr: in std_logic_vector(10 downto 0);
estado: in std_logic_vector(1 downto 0);
d_sen: in std_logic_vector(13 downto 0);
inst: out std_logic_vector(1 downto 0);
param: out std_logic_vector(7 downto 0)
);
end component;

signal iniciar,f_ack,str,dv_cmos,fin_cmos,ft,dv_pc,env,tack,e,l,ef,lf,ea: std_logic;
signal la,emp,full,ini_r,inst_ack,inst_done,nrst: std_logic;
signal str_env,str_env_ack,tv: std_logic;
signal inst,edo: std_logic_vector(1 downto 0);
signal dato_pc,dae,param: std_logic_vector(7 downto 0);
signal dt: std_logic_vector(9 downto 0);
signal nc,nr: std_logic_vector(10 downto 0);
signal dato_cmos: std_logic_vector(13 downto 0);
signal dato_de_ram: std_logic_vector(15 downto 0);
signal rad_prom,d_ac,calive: std_logic_vector(15 downto 0);

begin

nrst <= not rst;
start_cnv <= str;

cmos: modulo_cmos Port map(
clk => clk,
rst => nrst,
str_env => str_env,
env_ack => str_env_ack,
ini => iniciar,
fin_ack => f_ack,
dato_p => dato_p,
dato_n => dato_n,
clk_cmos_p => clk_cmos_p,
clk_cmos_n => clk_cmos_n,
clk_master => clk_master,
start_cnv => str,
dato_val => dv_cmos,
fin => fin_cmos,
dato_sen => dato_cmos,
t_val => tv,
dt => dt,
d_ac => d_ac,
nc => nc,
nr => nr,

```

```
rad_prom => rad_prom(13 downto 0)
);
```

```
ctrl: control_cmos_beta Port map(
  clk => clk,
  rst => nrst,
  ini_ext => str_ext,
  ini_cap => str_cap,
  ini_trans => str_int,
  ini_ack => str,
  captura => str_env,
  t_val => tv,
  captura_ack => str_env_ack,
  ft => ft,
  dv_pc => dv_pc,
  dato_pc => dato_pc,
  dv_cmos => dv_cmos,
  env_dat => env,
  t_ack => tack,
  dae => dae,
  iniciar => iniciar,
  fin_cmos => fin_cmos,
  fin_ack => f_ack,
  l_ack => la,
  e_ack => ea,
  lf => lf,
  e => emp,
  f => full,
  leer => l,
  escribir => e,
  rad_val => dt,
  dato_de_ram => dato_de_ram,
  estado => edo
);
```

```
comm: bloque_tran_re port map(
  clk => clk,
  rst => nrst,
  ini_t => env,
  rx => rx,
  d_a_e => dae,
  tx => tx,
  fin_t => ft,
  bsy_t => tack,
  dato_valido => dv_pc,
  dato_r => dato_pc
);
```

```
mem: FIFO_Memoria_RAM port map(
```

```

clk => clk,
rst => nrst,
escribir => e,
leer => l,
dato_cmos => dato_cmos,
empty => emp,
full => full,
escribir_ack => ea,
escribir_fin => ef,
leer_ack => la,
leer_fin => lf,
ce0 => ce0,
ce20 => ce20,
ce1 => ce1,
ce21 => ce21,
ce2 => ce2,
ce22 => ce22,
ce3 => ce3,
ce23 => ce23,
we => we,
oe => oe,
lb => lb,
ub => ub,
datos_de_ram => dato_de_ram,
dir => dir,
datos_ram => datos_ram);

```

```

ctrl_lcd: fsm_ctrl_lcd port map(
  clk => clk,
  rst => nrst,
  sel => sel,
  ini_rdy => ini_r,
  inst_ack => inst_ack,
  inst_done => inst_done,
  nc => nc,
  nr => nr,
  estado => edo,
  d_sen => rad_prom(13 downto 0),
  inst => inst,
  param => param);

```

```

inter_LCD: LCD port map(
  clk => clk,
  rst => nrst,
  inst => inst,
  param => param,
  ini_rdy => ini_r,
  inst_ack => inst_ack,
  inst_done => inst_done,

```

```

en => en,
rs => rs,
w => w,
datos => datos);

proc_alive: process(clk,nrst)
begin
    if nrst='1' then
        calive <= (others => '0');
    elsif rising_edge(clk) then
        calive <= std_logic_vector(unsigned(calive)+1);
    end if;
end process;

sal_alive: process (calive,edo)
begin
    alive <= '0';
    case edo is
        when "00" =>
            if calive < "0001000000000000" then
                alive <= '1';
            end if;
        when "01" =>
            if calive < "1111000000000000" then
                alive <= '1';
            end if;
        when "10" =>
            if calive > "1111111111111111" then
                alive <= '1';
            end if;
        when others =>
            if calive > "1000000000000000" then
                alive <= '1';
            end if;
    end case;
end process;
end Behavioral;

```

Propiedad Intelectual Registrada

MX/u/2018/000001

gob mx
Instituto Mexicano de la Propiedad Industrial

REPRESENTACIÓN DE LA SOLICITUD DE REGISTRO DE MODELO DE UTILIDAD ENVIADA A TRAVÉS DEL PORTAL DE PAGOS Y SERVICIOS ELECTRÓNICOS.

Homoclave del formato	Folio
IMPI-00-009	Folio: MX/E/2018/000265

Fecha de publicación en el DOF	Fecha de solicitud del trámite
18 03 2010	08 01 2018

Datos generales de la solicitud

<ul style="list-style-type: none"> <input checked="" type="radio"/> Solicitud de Patente <input checked="" type="radio"/> Solicitud de Registro de Modelo de Utilidad Normal <input checked="" type="radio"/> Solicitud de Registro de Diseño Industrial <p style="margin-left: 20px;">Especifique cual:</p> <p style="margin-left: 40px;"> <input checked="" type="radio"/> Modelo Industrial <input checked="" type="radio"/> Dibujo Industrial </p>	<p>Expediente: MX/u/2018/000001</p> <p>ID Solicitud: 2730</p> <p>Fecha: 08/01/2018 12:39:15</p>
---	---

Datos del (de los) solicitante(s)

Personas físicas	Personas morales
CURP:	RFC: INS15110486A
Nombre(s):	Denominación o razón social:
Primer apellido:	INSTECH S.A.P.I. DE C.V.
Segundo apellido:	Nacionalidad: MEXICO
Nacionalidad:	Teléfono (Lada, Número, Extensión):
Teléfono (Lada, Número, Extensión):	4921624703
Correo electrónico:	Correo electrónico: rcastm@instech.com.mx
Continúa en anexo	Continúa en anexo

gob mx	
Instituto Mexicano de la Propiedad Industrial	
Domicilio del (de los) solicitante(s)	
Código postal: 90610	
Calle: BLVD LOPEZ PORTILLO	
Número exterior: 100	Número interior: A
Colonia: DEPENDENCIAS FEDERALES	
Municipio o delegación: GUADALUPE	Localidad:
Estado o entidad federativa: ZACATECAS	Entre calles:
País: MEXICO	

Datos del (de los) inventor(es)	
CURP: OEVG820503HTSRRR04	
Nombre(s): Gerardo	
Primer apellido: ORNELAS	
Segundo apellido: VARGAS	
Nacionalidad: MEXICO	
Teléfono (Lada, Número, Extensión):	
Correo electrónico: ornelashls@gmail.com	 Continúa en anexo

Domicilio del (de los) inventor(es)	
Código postal: 90610	
Calle: HACIENDA DEL CUIDADO	
Número exterior: 18	Número interior:
Colonia: NUEVO BERNARDEZ	
Municipio o delegación: GUADALUPE	Localidad: querétaro
Estado o entidad federativa: ZACATECAS	Entre calles:
País: MEXICO	

gob mx

Instituto Mexicano de la Propiedad Industrial

Datos del (de los) apoderado(s)

CURP:	RGP:
Nombre(s): Rodrigo	Personas autorizadas para oír/recibir notificaciones: Gerardo ORNELAS VARGAS
Primer apellido: CASTAÑEDA	
Segundo apellido: MIRANDA	
Nacionalidad: MEXICO	
Teléfono (Lada, Número, Extensión): 4921624703	
Correo electrónico: rcastm@gmail.com	 Continúa en anexo

Domicilio del (de los) apoderado(s)

Código postal: 98610	
Calle: INFANTE	
Número exterior: 137	Número interior:
Colonia: PRIVADA CONDE SANTIAGO DE LA LAGUNA	
Municipio o delegación: GUADALUPE	Localidad:
Estado o entidad federativa: ZACATECAS	Entre calles:
País: MEXICO	

Datos de la Solicitud

Denominación o título de la invención: PLATAFORMA DE DESARROLLO DE SISTEMAS DIGITALES BASADA EN FPGA
Fecha divulgación previa:

Lista Inventores

Inventor 2

- Nombre: Celina Lizeth CASTAÑEDA MIRANDA

- CURP:

- Nacionalidad: MEXICO

- Domicilio: Calle CIRCUITO DE PONTEVEDRA, Ext. 84, Col. LOMAS DE GALICIA, C. P. 98610

- Población, Estado y País: GUADALUPE, ZACATECAS, MEXICO

Inventor 3

- Nombre: José Manuel ORTIZ RODRÍGUEZ

- CURP:

- Nacionalidad: MEXICO

- Domicilio: Calle ANDADOR SANTA MARÍA, Ext. 4, Col. LOMAS DEL CONVENTO, C. P. 98609

- Población, Estado y País: GUADALUPE, ZACATECAS, MEXICO

Inventor 4

- Nombre: Rodrigo CASTAÑEDA MIRANDA

- CURP: CAMR760523

- Nacionalidad: MEXICO

- Domicilio: Calle INFANTE, Ext. 137, Col. CONDE DE SANTIAGO DE LA LAGUNA, C. P. 98610

- Población, Estado y País: GUADALUPE, ZACATECAS, MEXICO

Inventor 5

- Nombre: Luis Octavio SOLIS SÁNCHEZ

- CURP:

- Nacionalidad: MEXICO

- Domicilio: Calle CIRCUITO DE PONTEVEDRA, Ext. 84, Col. LOMAS DE GALICIA, C. P. 98610

- Población, Estado y País: GUADALUPE, ZACATECAS, MEXICO

Inventor 6
- Nombre: Gerardo Marx CHÁVEZ CAMPOS
- CURP:
- Nacionalidad: MEXICO
- Domicilio: Calle SIERRA DE INGUARAN, Ext. 140, Col. SANTIAGUITO, C. P. 58120
- Población, Estado y País: MORELIA, MICHOACÁN, MEXICO

Inventor 7
- Nombre: Héctor Alonso GUERRERO OSUNA
- CURP:
- Nacionalidad: MEXICO
- Domicilio: Calle GIRASOLES, Ext. 60, Col. GARDENIAS, C. P. 98604
- Población, Estado y País: GUADALUPE, ZACATECAS, MEXICO

Inventor 8
- Nombre: Ma. Del Rosario MARTÍNEZ BLANCO
- CURP:
- Nacionalidad: MEXICO
- Domicilio: Calle ANDADOR SANTA MARÍA, Ext. 4, Col. LOMAS DEL CONVENTO, C. P. 98609
- Población, Estado y País: GUADALUPE, ZACATECAS, MEXICO

Inventor 9
- Nombre: Jesús Antonio NAVA PINTOR
- CURP:
- Nacionalidad: MEXICO
- Domicilio: Calle SAN MATEO, Ext. 84, Col. CAMPO BRAVO, C. P. 98609
- Población, Estado y País: GUADALUPE, ZACATECAS, MEXICO

Inventor 10
- Nombre: Juan José GARCÍA ESCALANTE
- CURP:
- Nacionalidad: MEXICO
- Domicilio: Calle BLVD. MISIÓN DE SAN FRANCISCO, Ext. 385, Int. 50, Col. PUNTA JURQUILLA, Localidad condominio naranjos, C. P. 76230
- Población, Estado y País: QUERETARO, QUERETARO, MEXICO

gob mx

Instituto Mexicano de la Propiedad Industrial

REPRESENTACIÓN DE LA SOLICITUD DE REGISTRO DE PATENTE ENVIADA A TRAVÉS DEL PORTAL DE PAGOS Y SERVICIOS ELECTRÓNICOS.

Homoclave del formato	Folio
IMPI-00-009	Folio: MX/E/2018/000259

Fecha de publicación en el DOF	Fecha de solicitud del trámite
18 03 2010	08 01 2018

Datos generales de la solicitud

<input checked="" type="radio"/> Solicitud de Patente Normal <input type="radio"/> Solicitud de Registro de Modelo de Utilidad <input type="radio"/> Solicitud de Registro de Diseño Industrial Especifique cual: <input type="radio"/> Modelo Industrial <input type="radio"/> Dibujo Industrial	Expediente: MX/a/2018/000044 ID Solicitud: 2885 Fecha: 08/01/2018 12:36:23
--	--

Datos del (de los) solicitante(s)

Personas físicas	Personas morales
CURP:	RFC: INS15110486A
Nombre(s):	Denominación o razón social:
Primer apellido:	INSTECH S.A.P.I. DE C.V.
Segundo apellido:	Nacionalidad: MEXICO
Nacionalidad:	Teléfono (Lada, Número, Extensión):
Teléfono (Lada, Número, Extensión):	Correo electrónico:
Correo electrónico:	
 Continúa en anexo	 Continúa en anexo

gob mx	
Instituto Mexicano de la Propiedad Industrial	
Domicilio del (de los) solicitante(s)	
Código postal: 98610	
Calle: BLVD LOPEZ PORTILLO 100	
Número exterior: 100	Número interior: A
Colonia: DEPENDENCIAS FEDERALES	
Municipio o delegación: GUADALUPE	Localidad:
Estado o entidad federativa: ZACATECAS	Entre calles:
País: MEXICO	

Datos del (de los) inventor(es)	
CURP:	
Nombre(s): Gerardo	
Primer apellido: ORNELAS	
Segundo apellido: VARGAS	
Nacionalidad: MEXICO	
Teléfono (Lada, Número, Extensión):	
Correo electrónico:	 Continúa en anexo

Domicilio del (de los) inventor(es)	
Código postal: 98610	
Calle: HACIENDA DEL CUIDADO	
Número exterior: 18	Número interior:
Colonia: NUEVO BERNARDEZ	
Municipio o delegación: GUADALUPE	Localidad:
Estado o entidad federativa: ZACATECAS	Entre calles:
País: MEXICO	

gob mx

Instituto Mexicano de la Propiedad Industrial

Datos del (de los) apoderado(s)

CURP:	RGP:
Nombre(s): Rodrigo	Personas autorizadas para oír/recibir notificaciones: Gerardo ORNELAS VARGAS
Primer apellido: CASTAÑEDA	
Segundo apellido: MIRANDA	
Nacionalidad: MEXICO	
Teléfono (Lada, Número, Extensión): 4921624703	
Correo electrónico: rcastm@gmail.com	 Continúa en anexo

Domicilio del (de los) apoderado(s)

Código postal: 98610	
Calle: INFANTE	
Número exterior: 137	Número interior:
Colonia: PRIVADA CONDE SANTIAGO DE LA LAGUNA	
Municipio o delegación: GUADALUPE	Localidad:
Estado o entidad federativa: ZACATECAS	Entre calles:
País: MEXICO	

Datos de la Solicitud

Denominación o título de la invención: SISTEMA PARA DESARROLLAR DISPOSITIVOS DE RADIOGRAFÍA DIGITAL
Fecha divulgación previa:

Lista Inventores

Inventor 2

- Nombre: Juan José GARCÍA ESCALANTE

- CURP:

- Nacionalidad: MEXICO

- Domicilio: Calle BLVD MISIÓN DE SAN FRANCISCO, Ext. 385, Int. 50, Col. PUNTA JURQUILLA, Localidad Condominio Naranjos III, C. P. 76230

- Población, Estado y País: QUERÉTARO, QUERÉTARO, MEXICO

Inventor 3

- Nombre: José Manuel ORTIZ RODRÍGUEZ

- CURP:

- Nacionalidad: MEXICO

- Domicilio: Calle ANDADOR SANTA MARÍA, Ext. 4, Col. LOMAS DEL CONVENTO, C. P. 98609

- Población, Estado y País: GUADALUPE, ZACATECAS, MEXICO

Inventor 4

- Nombre: Ma. Del Rosario MARTÍNEZ BLANCO

- CURP:

- Nacionalidad: MEXICO

- Domicilio: Calle ANADADOR SANTA MARIA, Ext. 4, Col. LOMAS DEL CONVENTO, C. P. 98609

- Población, Estado y País: GUADALUPE, ZACATECAS, MEXICO

Inventor 5

- Nombre: Nahum Mauricio MENDOZA JASSO

- CURP:

- Nacionalidad: MEXICO

- Domicilio: Calle SAN MATEO, Ext. 53, Col. SAN MIGUEL DEL CORTIJO, C. P. 98615

- Población, Estado y País: GUADALUPE, ZACATECAS, MEXICO

Inventor 6

- Nombre: Luis Octavio SOLÍS SÁNCHEZ

- CURP:

- Nacionalidad: MEXICO

- Domicilio: Calle CIRCUITO DE PONTEVEDRA, Ext. 84, Col. LOMAS DE GALICIA, C. P. 98610

- Población, Estado y País: GUADALUPE, ZACATECAS, MEXICO

Inventor 7
- Nombre: Rodrigo CASTAÑEDA MIRANDA
- CURP:
- Nacionalidad: MEXICO
- Domicilio: Calle INFANTE, Ext. 137, Col. CONDE DE SANTIAGO DE LA LAGUNA, C. P. 98810
- Población, Estado y País: GUADALUPE, ZACATECAS, MEXICO
Inventor 8
- Nombre: Iván DELGADO ENCISO
- CURP:
- Nacionalidad: MEXICO
- Domicilio: Calle ZENZONTLE, Ext. 53, Col. RDCIAL SANTA BÁRBARA, C. P. 28017
- Población, Estado y País: COLIMA, COLIMA, MEXICO
Inventor 9
- Nombre: Margarita De La Luz MARTÍNEZ FIERRO
- CURP:
- Nacionalidad: MEXICO
- Domicilio: Calle LÓPEZ MATEOS, Ext. 11, Col. BARRIO LA EMBARCACIÓN, C. P. 98640
- Población, Estado y País: TRANCOSO, ZACATECAS, MEXICO
Inventor 10
- Nombre: Celina Lizeth CASTAÑEDA MIRANDA
- CURP:
- Nacionalidad: MEXICO
- Domicilio: Calle CIRCUITO DE PONTEVEDRA, Ext. 84, Col. LOMAS DE GALICIA, C. P. 98810
- Población, Estado y País: GUADALUPE, ZACATECAS, MEXICO
Inventor 11
- Nombre: Irám Pablo RODRÍGUEZ SÁNCHEZ
- CURP:
- Nacionalidad: MEXICO
- Domicilio: Calle AVENIDA FRANCISCO I MADERO, Ext. 3107, Int. 2, Col. MITRAS CENTRO, C. P. 64460
- Población, Estado y País: MONTERREY, NUEVO LEÓN, MEXICO

Inventor 12

- Nombre: Joel Bacilio CARLOS PARRA

- CURP:

- Nacionalidad: MEXICO

- Domicilio: Calle MANUEL DE LA PEÑA, Ext. 967, Col. NUEVO MÉXICO, C. P. 80297

- Población, Estado y País: CULIACÁN, SINALOA, MEXICO

Inventor 13

- Nombre: Idalia GARZA VELOZ

- CURP:

- Nacionalidad: MEXICO

- Domicilio: Calle AV. DEL DERECHO, Ext. 42, Col. S.P.A.U.A.Z., C. P. 98613

- Población, Estado y País: GUADALUPE, ZACATECAS, MEXICO

Inventor 14

- Nombre: Héctor Alonso GUERRERO OSUNA

- CURP:

- Nacionalidad: MEXICO

- Domicilio: Calle GIRASOLES, Ext. 60, Col. GARDENIAS, C. P. 98604

- Población, Estado y País: GUADALUPE, ZACATECAS, MEXICO