



**CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS AVANZADOS DEL
INSTITUTO POLITECNICO NACIONAL**

Unidad Zacatenco

Departamento de Control Automático

Adquisición y procesamiento digital de imágenes microscópicas provenientes de pruebas de dureza Vickers

Tesis que presenta

Saúl Manuel Domínguez Nicolás

para obtener el grado de

Doctor en Ciencias

en la especialidad de

Control Automático

Directora de la Tesis: Dra. Petra Wiederhold Grauert

Ciudad de México

Noviembre 2018

Agradecimientos

A MI FAMILIA - MI MADRE, MI PADRE, MI ESPOSA Y MIS HIJOS

por su enorme paciencia, cariño y apoyo incondicional durante los años de este proyecto doctoral y a lo largo de mi vida.

A LA DRA. PETRA WIEDERHOLD

por su gran apoyo académico.

A MIS COMPAÑEROS, AMIGOS Y COLEGAS

por su valioso apoyo a lo largo de estos laboriosos años.

A MIS PROFESORES SINODALES

por el tiempo invertido en la revisión de este trabajo, y por sus valiosos y constructivos comentarios.

AL CINVESTAV Y SU DEPARTAMENTO DE CONTROL AUTOMÁTICO

por permitir mi desarrollo académico en una institución de gran prestigio.

AL CONACYT

por el apoyo económico, el cual me permitió continuar mis estudios.

Resumen

Esta tesis contribuye a la automatización de pruebas de dureza por indentación, en base a adquisición, procesamiento y análisis de imágenes microscópicas de una cámara CCD integrada en el durómetro Mitutoyo HM-125. El trabajo tiene tres componentes: un sistema de adquisición de imagen para filtrar, discretizar y memorizar las imágenes en formatos bmp y png, un algoritmo novedoso para detectar y analizar una indentación Vickers en la imagen digital, y la generación de un Banco de Datos de 230 imágenes de indentación Vickers que incluye anotaciones por un experto humano quien obtuvo las características de indentación de manera manual.

El sistema de adquisición realiza de manera paralela el filtrado de la señal analógica y la adquisición de imagen en tiempo real, mediante módulos programados en Verilog HDL, para un FPGA Altera DE2-115. El sistema incluye una interfaz gráfica para el usuario para la comunicación con el FPGA y la configuración del filtro.

La indentación generada por un indentador Vickers con punta piramidal de diamante, tiene una forma rómbica. Nuestro algoritmo que fue implementado en Matlab, realiza segmentación de imagen por binarización, crecimiento de regiones, filtrado morfológico, identificación y localización de la indentación, para finalmente detectar sus cuatro vértices mediante un detector de esquinas, y así calcular las longitudes de diagonales y el valor de dureza Vickers. El algoritmo fue probado en 230 imágenes de muestras de Acero-316 y de Nitruro de Hafnio, incluyendo muestras con pulido tipo espejo, superficies ásperas o con imperfecciones, así como indentaciones con frontera deformada o dañada. Los errores obtenidos por comparar los resultados de nuestro algoritmo con datos generados por un experto humano quien midió las indentaciones de manera manual, corresponden al estado del arte internacional. Sin embargo, nuestro algoritmo supera a los conocidos en simplicidad y eficiencia.

Abstract

This thesis contributes to the automatization of hardness tests via indentation, by means of acquisition, processing and analysis of the microscopic images generated by a CCD camera integrated in the durometer Mitutoyo HM-125. The work developed consists of three components: an image acquisition system to filter, digitize, and storage the images in bmp and png formats, a novel algorithm to detect and analyze a Vickers indentation footprint in the image, and the generation of a Data Bank of 230 Vickers indentation images, which includes expert annotations generated by a human operator who manually measured the indentation characteristics.

The image acquisition system performs, in a parallel manner, filtering of the analog video signal and real time image acquisition, by means of programs developed in Verilog HDL, for an FPGA Altera DE2-115. The system includes a graphic user interface to communicate with the FPGA, and to configurate the filter.

Indentation footprints generated by a Vickers indenter, which has a pyramidal diamond spike, are of rhombic form. Our algorithm, implemented in Matlab, performs image segmentation by binarization, region growing, morphological filtering, indentation identification and localization, and finally, detection of its four vertices using corner detection, in order to calculate the indentation diagonal lengths and the Vickers hardness number. Our algorithm was tested on 230 images from Steel-316 and Hafnium Nitride specimens, including specular-polished and rough surfaces, specimens with artifacts or imperfections, and indentations with deformed or damaged edges. The errors obtained from comparisons between the results from our algorithm, and expert dates generated by a human operator who measured each indentation in the conventional manual manner, correspond to the international state of the art. Nevertheless, our algorithm is simpler and hence more efficient than algorithms known from the literature.

Índice de contenido

Resumen	v
Abstract	vii
1 Introducción	1
1.1 Motivación y antecedentes	1
1.2 Justificación del trabajo de tesis	6
1.3 Objetivos de la tesis	10
1.4 Resultados y contribuciones de la tesis	12
1.5 Estructura de la tesis	14
2 Adquisición de imagen mediante un sistema de arreglos de compuertas programables (FPGA)	15
2.1 Introducción	15
2.2 Arquitectura del FPGA	17
2.3 Tarjeta de desarrollo FPGA	18
2.4 Video analógico	19
2.4.1 Interfaz de video compuesta NTSC	19
2.4.2 Interfaz de video compuesta PAL	20
2.4.3 Interfaz de video compuesta SECAM	20
2.5 Video digital	20
2.5.1 Formato 2:4:1 a 4:4:4 YCbCr	21
2.5.2 Formato YCbCr 4:2:2	22
2.5.3 Formato YCbCr 4:1:1	22
2.6 Sistema de adquisición de imagen (hardware)	23
2.6.1 Diseño general y componentes del sistema	23

2.6.2	Salida VGA	26
2.6.3	Bloques de arreglos lógicos (LAB)	31
2.6.4	Interconexiones LAB	32
2.6.5	Señales de control de un LAB	33
2.6.6	Unidad de memoria	34
2.6.7	Modo de puerto único	36
2.6.8	Modo de puerto dual simple	37
2.6.9	Modo Buffer FIFO	37
2.6.10	Memoria SDRAM	38
2.6.11	Decodificador de TV a sistema VGA	38
3	Adquisición de imagen mediante módulos programados en Verilog para el FPGA Altera DE2-115	41
3.1	Introducción	41
3.2	Módulo principal programado en Verilog HDL	42
3.3	Módulo ITU-R-656 a 4:2:2	43
3.4	Bloque SDRAM frame buffer	47
3.5	Transformación de YCbCr 4:2:2 a YCbCr 4:4:4 y conversión en formato RGB	49
3.6	Controlador de VGA	51
3.7	Comunicación vía USB JTAG	53
4	Interfaz gráfica de usuario para la comunicación con el FPGA, y para la supresión de ruido de señal	57
4.1	Introducción	57
4.2	Espectro de frecuencias de la señal de video analógica reproducida por una cámara CCD	58
4.3	Filtros lineales	60
4.4	Filtro digital IIR implementado en el FPGA	61
4.5	Interfaz de usuario	64
5	Procesamiento y análisis de imágenes de indentaciones Vickers	69
5.1	Introducción	69
5.2	Estado del arte	72

5.2.1	Métodos propuestos en la literatura para procesar y analizar imágenes de indentación Vickers	72
5.2.2	Métodos basados en un “mapa del área” según Maier, Uhl y Gadermayr	73
5.3	Transformación log-polar	73
5.4	Propuesta de un nuevo algoritmo para analizar imágenes de indentación Vickers	80
5.4.1	Definición de materiales y condiciones para la medición de dureza Vickers mediante el nuevo algoritmo	80
5.4.2	Idea general del nuevo algoritmo	81
5.4.3	Contribución del nuevo algoritmo al estado del arte	83
5.4.4	Detalles del algoritmo propuesto	83
5.5	Banco de imágenes de indentación Vickers	91
6	Experimentos	93
6.1	Ajuste experimental	93
6.2	Análisis de imágenes de indentación	96
	Conclusiones	112
	Bibliografía	115
	Anexos	123
	Anexo A: Códigos de programas elaboradas en Verilog HDL (8 páginas) - inicia en la página 125	
	Anexo B: Tablas con los resultados de experimentos con 230 imágenes de indentación Vickers (23 páginas) - inicia en la página 133	
	Anexo C: Páginas portadas de seis publicaciones (6 páginas) - inicia en la página 159	

Lista de figuras

1.1	Huella producida por indentación con un indentador de forma piramidal.	2
1.2	Componentes básicos de un sistema de indentación.	2
1.3	Representación seccional de una prueba de dureza con un indentador asimétrico.	3
1.4	Desplazamiento de la carga de indentación en un ciclo completo.	3
1.5	Indentaciones en posiciones específicas en un circuito microelectrónico.	5
1.6	Imágenes captadas por un microscopio de un microdurómetro.	7
1.7	Microdurómetro modelo HM-125.	7
1.8	Huella proporcionada por un indentador Vickers en metal.	8
1.9	Huella proporcionada por el indentador a menor fuerza.	8
1.10	Huella de indentación en acrílico.	9
2.1	Tarjeta Altera DE2 Cyclone II.	16
2.2	Diagrama a bloques del dispositivo Cyclone II 2C35.	17
2.3	Conversión del video analógico a video digital.	18
2.4	Relación entre campos y frames.	19
2.5	Proceso del submuestreo croma.	21
2.6	Posición de muestras Y, Cb, Cr en formato 4:4:4.	22
2.7	Posición de muestras Y, Cb, Cr en formato 4:2:2.	22
2.8	Posición de muestras Y, Cb, Cr en formato 4:1:1.	23
2.9	Diagrama a bloque del sistema de adquisición de imagen.	23
2.10	Esquema del decodificador de TV.	24
2.11	Circuito de comunicación del USB JTAG.	26
2.12	Esquema del VGA.	27
2.13	Especificación del tiempo en barrido horizontal del VGA.	27
2.14	Estructura LAB del Cyclone II.	31

2.15	Estructura LAB del FPGA Cyclone II.	32
2.16	Señales de control de un LAB.	33
2.17	Señal de control de bloques de memoria del FPGA.	35
2.18	Habilitación de la entrada de reloj del registro de memoria del FPGA.	36
2.19	Configuración de memoria de puerto único.	36
2.20	Configuración de memoria de puerto dual.	37
2.21	Diagrama a bloques del convertidor de TV a VGA.	39
3.1	Módulos para convertir datos de imagen al formato YCbCr.	43
3.2	Bit de asignación para el cuarto byte de códigos EAV y SAV.	44
3.3	Composición del flujo de datos de la interfaz ITU-R BT656.	46
3.4	Módulos de adquisición de imagen implementados en Verilog HDL.	48
3.5	Lectura de la memoria SDRAM.	48
3.6	Multiplexado de líneas de video entre dos <i>buffers</i>	48
3.7	Operación de desentrelace para cada pixel.	49
3.8	Entradas y salidas del módulo YUV422_to_444.	50
3.9	Interpolación para valores de crominancia.	50
3.10	Bloque de conversión del espacio de color.	51
3.11	Señales de sincronía generadas por el controlador de VGA.	52
3.12	Esquema del módulo USB_JTAG.	53
4.1	Espectro de frecuencia del voltaje de salida del amplificador de una cámara CCD.	59
4.2	Bloques implementados en el FPGA para el diseño del filtro.	62
4.3	Diagrama del Filtro IIR implementado en el FPGA.	62
4.4	Símbolo del bloque multiplicador.	63
4.5	Diagrama a bloque interno de un multiplicador.	64
4.6	Imagen de la interfaz gráfica de usuario, desarrollada en Delphi.	66
5.1	Mapeo del plano cartesiano al plano log-polar.	76
5.2	Efectos de la rotación de una línea y del escalamiento de un círculo en la transformación log-polar.	78
5.3	Imagen de indentación y su transformación log-polar.	79
5.4	Imagen de indentación de una muestra con superficie áspera.	80
5.5	Indentación Vickers con vértices y diagonales marcados.	82

5.6	Indentaciones Vickers con sus correspondientes imágenes binarias. . .	84
5.7	Imágenes de indentación, sus versiones binarias y los resultados del filtro morfológico.	86
5.8	Ejemplo de aplicación del algoritmo propuesto completo.	90
6.1	Sistema de adquisición de imagen para el microdurómetro HM-125. .	93
6.2	Montaje experimental para medir el espectro de frecuencia a la entrada del decodificador de video del FPGA.	94
6.3	Espectro de frecuencia del voltaje de salida del decodificador de TV filtrado.	95
6.4	(a) Imagen de indentación con ruido eléctrico. (b) Imagen captada a través de nuestro sistema de adquisición de imagen que incluye filtrado embebido en el FPGA.	96
6.5	Imágenes del Grupo 1 con error menor a 1% en la longitud promedio de diagonales, con sus vértices de indentación marcados en color rojo.	100
6.6	Grupo 1: imágenes con error de longitud promedio de diagonales entre 1% y 3% (hasta imagen 69), y con error entre 3% y 5% (a partir de imagen 30).	101
6.7	Grupo 2: imágenes con error de longitud promedio de diagonales menor de 1%.	103
6.8	Grupo 2: imágenes con error de longitud promedio de diagonales entre 1% y 3%.	104
6.9	Grupo 2: imágenes con error de longitud promedio de diagonales entre 3% y 5%.	104
6.10	Grupo 2: imágenes con error de longitud promedio de diagonales entre 5% y 9%.	105
6.11	Grupo 2: dos imágenes fuera del alcance del algoritmo. (a) imagen original con los vértices detectados sobrepuestos. (b) resultado de binarización. (c) respuesta del filtro morfológico.	105
6.12	Grupo 3: imágenes con error de longitud promedio de diagonales menor de 1%.	106
6.13	Grupo 3: imágenes con error de longitud promedio de diagonales entre 1% y 3%.	107

6.14	Grupo 3: imágenes con error de longitud promedio de diagonales entre 3% y 5%.	107
6.15	Grupo 3: imágenes con error de longitud promedio de diagonales entre 5% y 9%.	108
6.16	Grupo 3: imagen fuera del alcance del algoritmo. (a) imagen original. (b) resultado de binarización. (c) respuesta del filtro morfológico. . .	108
6.17	Grupo 4: imágenes con error de longitud promedio de diagonales menor a 1%.	109
6.18	Grupo 4: imágenes con error de longitud promedio de diagonales entre 1% y 3%, excepto de la imagen 35 cuyo error está entre 3% y 5%.	110
6.19	Grupo 4: dos imágenes fuera del alcance del algoritmo. (a) imagen original. (b) resultado de binarización. (c) respuesta del filtro morfológico.	111

Lista de tablas

2.1	Descripción de las terminales JTAG.	25
2.2	Especificación de tiempo horizontal en la salida VGA.	28
2.3	Especificación de tiempo vertical en la salida VGA.	29
2.4	Asignación de terminales del ADV7123.	30
2.5	Protocolo de comunicación.	40
3.1	Valor de asignación de cada bit de los códigos EAV y SAV.	45
3.2	Estado de bits de protección en dependencia de F, V y H.	45
3.3	Tiempos de sincronía horizontal y vertical.	52
4.1	Descripción de terminales del multiplicador.	65
5.1	Conversión de píxeles a longitudes, en dependencia de la magnificación.	91
6.1	Recursos usados por el sistema de adquisición de imagen.	95
6.2	Distribución de errores relativos de la longitud promedio de diagonales.	99

Capítulo 1

Introducción

1.1 Motivación y antecedentes

En los procesos de fabricación de dispositivos microelectrónicos, las pruebas de dureza son de vital importancia en la evaluación del material en cuanto a sus propiedades mecánicas y control de calidad del producto a fabricar. Por ejemplo, la tensión residual en las películas delgadas en los substratos de los semiconductores es de especial interés para la industria de la microelectrónica, debido a que dicha tensión inducida durante la fabricación puede dañar el dispositivo y/o reducir el tiempo de vida del mismo. La *prueba de dureza* del material, también llamada *indentación*, consiste en aplicar bajo un tiempo determinado una fuerza estática con una punta de material específico duro sobre la superficie del material a analizar [4, 48, 62, 68]. Materiales analizados de esta manera son por ejemplo cerámica [3], metales [23], polímeros [6] y muestras biológicas [39], dando como resultado la impresión de una huella sobre la superficie de dicho material, tal como se muestra en la Figura 1.1.

La Figura 1.2 presenta un esquema de equipo que realiza pruebas de indentación, consiste de tres componentes básicos:

- Indentador con una geometría específica, usualmente montado sobre una columna rígida, por medio del cual se transmite la fuerza hacia la muestra.
- Actuador que se utiliza para aplicar la fuerza.
- Sensor para medir los desplazamientos del indentador.

Avances en la instrumentación requieren tecnologías que demandan exactitud en

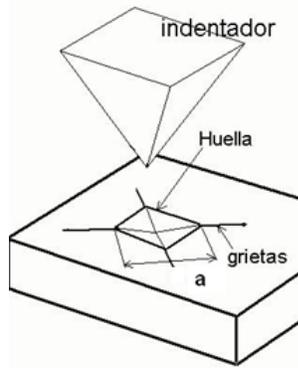


Figura 1.1: Huella producida por indentación con un indentador de forma piramidal.

la medición de propiedades mecánicas, tal es el caso de los procesos de fabricación de circuitos microelectrónicos y dispositivos de medición del campo magnético.

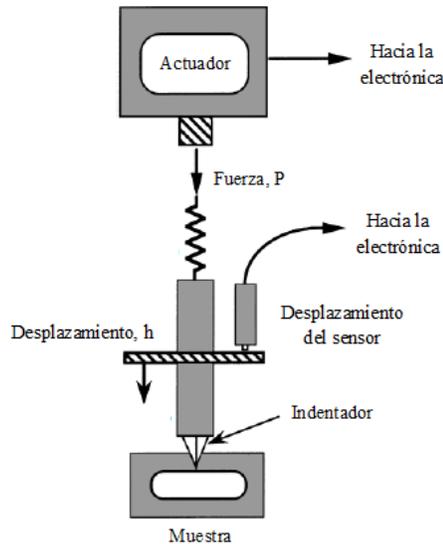


Figura 1.2: Componentes básicos de un sistema de indentación.

La Figura 1.3 simboliza un proceso de prueba de dureza, realizado con un indente asimétrico. A medida que el indente es impulsado en el material, ocurre la deformación elástica, produciendo una huella en la muestra que se ajusta a la forma del indente hasta cierta profundidad h_c de contacto, presentando un círculo de contacto de radio a . A medida que el indente se retira, sólo la parte elástica del desplazamiento tiende a recuperarse, permitiendo observar las propiedades elásticas del material.

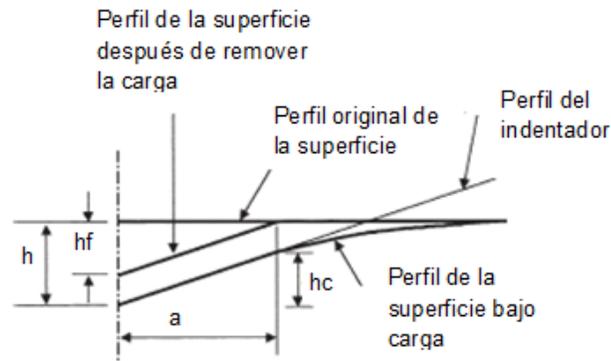


Figura 1.3: Representación seccional de una prueba de dureza con un indentador asimétrico.

El esquema de un ciclo completo de carga y descarga de indentación con una fuerza P contra un desplazamiento h , puede ser observada en Figura 1.4. Las cantidades importantes de esta representación son la carga máxima P_{max} , la máxima profundidad h_{max} , la profundidad residual después de la descarga hf , y la pendiente de la porción superior de la curva de descarga $S = dP/dH$. El parámetro S tiene como tamaño la fuerza por unidad de distancia y es conocido como la rigidez de contacto elástico. La dureza H es determinada por $H = P/A$, donde P es la carga y A es el área de contacto proyectado en la carga.

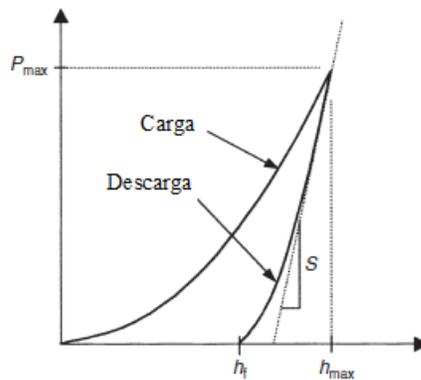


Figura 1.4: Desplazamiento de la carga de indentación en un ciclo completo.

Un microdurómetro permite registrar la fuerza aplicada al material, así como la imagen de la huella durante la aplicación del indentador y después de haberse retirado el mismo, lo que permite obtener otras propiedades mecánicas. Las fuerzas utilizadas oscilan entre $1 \mu\text{N}$ y 300 mN (30 gr). La resolución de estos equipos es

de aprox. $0.2 \mu\text{N}$ con la fuerza aplicada y de 2 nm en los desplazamientos de la muestra al ser sometido por el indentador. En la práctica actual, las mediciones manuales en las pruebas de dureza son interpretativas y subjetivas. Eso se debe a que la estimación de la medida de las diagonales en la huella del material al ser sometido por el indentador, convencionalmente es proporcionada por un operador humano, entonces los resultados dependen mucho del punto de vista de este operador (efecto de paralelismo) así como de su experiencia. Por ello, este proyecto doctoral propone desarrollar un nuevo sistema que coadyuve a la medición de la diagonal de la huella generada por la indentación Vickers, en algunos materiales, aplicando técnicas del procesamiento digital de imágenes. Dichas imágenes son reproducidas por un *dispositivo de carga acoplada* (**CCD**, por su siglas en inglés) de un microdurómetro modelo HM-125. Usando este equipo, nos confrontamos con el primer problema a resolver, que consiste en desarrollar un sistema de adquisición de imágenes que permite al usuario la observación y manipulación de las imágenes en tiempo real, así como también organiza de manera automática mandarlas a ser guardadas en formatos comerciales como bmp y png, en una computadora. Para esta adquisición de imágenes, este proyecto de tesis desarrolló un sistema basado en un **FPGA** (*arreglo de compuertas de campos programables*, de sus siglas en inglés, FPGA significa “field-programmable gate array”).

La indentación o prueba de dureza ha sido usada por mucho tiempo para la caracterización y control de calidad de materiales. Esta técnica fue implementada por Brinell hacia los años 1900, y desarrollada por D. Tabor entre 1952 y 1984 para el estudio de la dureza de metales. En tiempos más recientes, se ha determinado la tensión de películas delgadas mono- y multicapas a partir de pruebas de indentación estándar [34, 21, 67]. Por otra parte, de acuerdo al volumen y compuesto del material como cerámica, se determinó el módulo de Young [5] a través del estudio de pruebas de indentación conocidas como indentación Conica, Berkovich y Vickers [58]. Este módulo define la profundidad de penetración crítica como parte del resultado de la dureza del material, que es una de las propiedades mecánicas importantes del mismo. También se ha reportado la estimación de la tensión residual en películas muy delgadas de cerámica, a través de técnicas de difracción de Rayos-X [3].

Un método nuevo de pruebas mecánicas es la *prueba de indentación instrumentada*, también conocida como *nanoindentación* de ultra baja carga e indentación de almacenamiento continuo, que es considerada un resultado de la evolución de

las tradicionales pruebas de dureza. La nueva técnica emplea instrumentación de alta resolución para el control y monitoreo de cargas, así como del desplazamiento de un indentador dentro y fuera del material [40, 54]. Dependiendo de los detalles específicos del sistema de prueba, cargas de sólo 1 nN son implementadas en el indentador, así como también se pueden medir desplazamientos de 0.1 nm. Por lo regular, estas pruebas son utilizadas para medir la dureza del material, pero a partir de este parámetro, también se puede calcular el módulo de elasticidad del material (*Modulo de Young*) [49, 51]. Esta técnica también se ha usado para estimar la tenacidad a la fractura de materiales quebradizos, en base de medición óptica de las longitudes de las grietas que se forman en las huellas generadas por indentadores especiales [51, 27].

Muchos sistemas de indentación instrumentada están (semi-)automatizados para manipular la muestra y determinar propiedades mecánicas del material, distribuidas a lo largo de su superficie. Por ejemplo, se han determinado pequeñas indentaciones localizadas en puntos específicos en un circuito microelectrónico, como es mostrado en la Figura 1.5.

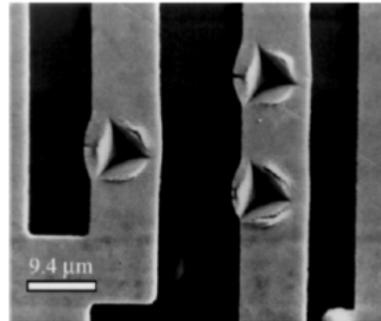


Figura 1.5: Indentaciones en posiciones específicas en un circuito microelectrónico.

El diamante es el material más usado como indentador, ya que por su dureza, minimiza la deformación de sí mismo, y así facilita la medición del desplazamiento del indentador sobre el material. Sin embargo, en la práctica se usan indentadores fabricados de diversos materiales menos rígidos, por ejemplo de zafiro, carburo de tungsteno, o acero inoxidable. Pero entonces, los desplazamientos elásticos del mismo indentador deben ser tomados en cuenta para determinar el desplazamiento de la carga en el material bajo estudio.

Las pruebas de indentación instrumentada tienen un campo amplio de crecimiento, en el cual en un futuro no lejano se esperan nuevos desarrollos tecnológicos.

En particular, se prevé una expansión y evolución de equipos como microdurómetros, con el resultado que las mediciones realizadas con estos equipos ya no dependan en su totalidad del punto de vista o de la experiencia del operador humano.

1.2 Justificación del trabajo de tesis

Algunas investigaciones han propuesto sistemas de adquisición y procesamiento de imagen para computadoras personales, para facilitar el análisis de indentaciones tipo Vickers y Brinell [22, 47]. Diversos microdurómetros reproducen la imagen microscópica de la indentación por medio de una cámara CCD [48]. Sin embargo, muchos de estos equipos no cuentan con frame grabber para la captura de la imagen, o utilizan frame grabbers comerciales para la adquisición de imagen, lo cual hace necesario el uso de software comercial de muy alto costo, como Labview o Matlab, entre otros. El uso de este tipo de software en una computadora personal hace difícil ejecutar el procesamiento en tiempo real [39]. Además, los sensores de carga incorporados en los CCDs presentan gran cantidad de ruido térmico y eléctrico [46], el cual degrada la resolución de energía [29, 37], de aquí la importancia de minimizar el ruido a través del procesamiento digital de señales. En [2], se utilizó un filtro pasa-bajas para disminuir interferencias en las CCDs producidas por ruido eléctrico.

Como se mencionó antes, las pruebas de indentación se aplican para determinar la tensión de películas delgadas, así como para estimar propiedades mecánicas como son el módulo de Young o de Poisson, la fractura y la dureza. Por ejemplo, los indentadores Vickers tienen en su punta un diamante piramidal el cual es aplicado a una superficie plana usando una fuerza conocida. La indentación resultante en la superficie es capturada usando un microscopio, vea Figura 1.6, y entonces las diagonales de la huella de forma rómbica son medidas por un operador humano, para determinar su longitud promedio y poder calcular la dureza del material. Bajo estas condiciones, las mediciones o estimaciones son subjetivas y dependen en gran medida de la experiencia del operador.

Es por eso que han sido publicadas diversas propuestas de la automatización de pruebas de indentación de tipo Vickers y Brinell, usando el procesamiento y el análisis de imágenes [10, 22, 23, 24, 25, 33, 41, 42, 43, 44, 45, 47, 52, 63, 65, 73].

La propuesta para este proyecto doctoral consiste en contribuir al desarrollo y

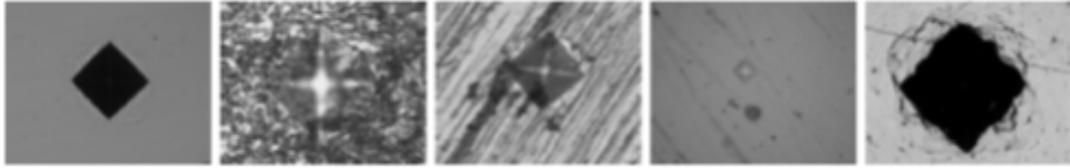


Figura 1.6: Imágenes captadas por un microscopio de un microdurómetro.

a la aplicación de técnicas del procesamiento digital de imágenes, para la medición y evaluación de la huella que es producida por un indentador Vickers en diversos materiales.



Figura 1.7: Microdurómetro modelo HM-125.

Para realizar esta propuesta de investigación, en el Centro de Investigación de Micro- y Nanotecnología (MICRONA) de la Universidad Veracruzana, se tiene un microdurómetro modelo HM-125, vea Figura 1.7. Dicho microdurómetro actualmente es usado para realizar indentación Vickers en metales como acero y en películas delgadas de nitruro de hafnio, un material importante como aislante térmico en la fabricación de circuitos integrados.

Usando el microdurómetro HM-125, se pueden realizar pruebas de indentación de fuerza a partir de $4.903 \cdot 10^{-3}$ N hasta 19.6 N. La duración de la carga puede ser ajustada desde 1 hasta 99 segundos con incrementos de 5 segundos. La mínima lectura de indentación es de $0.01 \mu\text{m}$, lo cual permite muy pequeñas indentaciones para ser medidas con alta resolución. Actualmente, un operador humano realiza la

medición de la huella de indentación a través de observar la imagen de la huella en un monitor de 9". Esta imagen es captada en tiempo real a través de un **CCD** (*dispositivo de carga acoplada*). Figura 1.8 presenta el resultado de una indentación sobre una muestra de acero, visto en el monitor del microdurómetro, en el cual se observan las diagonales de la huella generada por un indentador Vickers que consiste de un diamante piramidal.

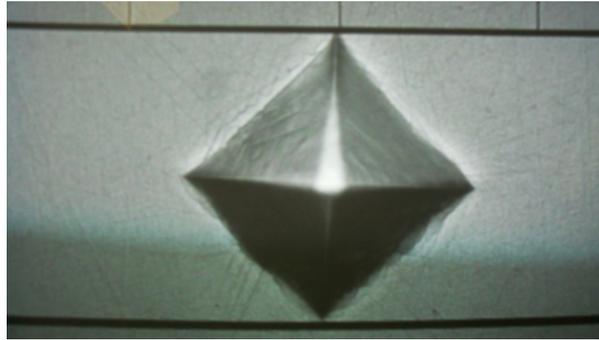


Figura 1.8: Huella proporcionada por un indentador Vickers en metal.

Sin embargo, efectuando una menor fuerza durante la indentación, es difícil para el operador humano determinar las diagonales de la huella proporcionada por el indentador, como se puede observar en la Figura 1.9.

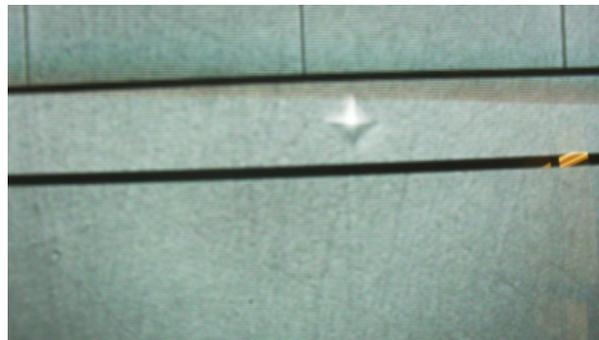


Figura 1.9: Huella proporcionada por el indentador a menor fuerza.

Para materiales frágiles como es el acrílico, se utiliza una carga mínima de 10 gf, durante un segundo; una huella producida por el indentador se aprecia en la Figura 1.10, es evidente que para el operador humano es muy difícil determinar las diagonales de la huella. Sería un reto interesante investigar las propiedades de esta imagen para ver si se pueden detectar y medir las diagonales buscadas de manera (semi-)automática.

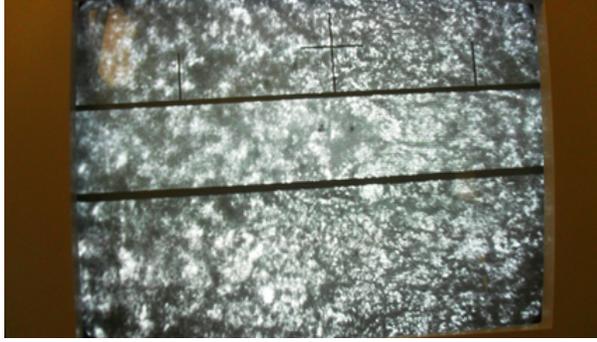


Figura 1.10: Huella de indentación en acrílico.

Se han reportado numerosos trabajos relacionados con la medición de las huellas producidas por indentadores, basados en técnicas de visión por computadora. Sin embargo, el procesamiento de imagen en tiempo real es complicado para ejecutarse en un procesador serial. Grandes cantidades de datos son usados para la representación de una imagen, y regularmente, se necesitan un gran número de operaciones (tanto globalmente como también en cada pixel) para el tratamiento de una imagen. La relación de video en tiempo real de 25 cuadros (*frames*) por segundo de una operación ejecutada en cada pixel de una imagen de color de 768x576 píxeles, es equivalente a 33 millones de operaciones por segundo, esto, sin tomar en cuenta la sobrecarga de memoria por el almacenamiento de imágenes y del procesamiento de las mismas, dando como resultado, muchas operaciones por segundo.

Por ejemplo, aplicaciones del procesamiento de imagen relacionadas con experimentos biomédicos o biológicos, o con la microscopía celular [71], requieren una enorme cantidad de operaciones ejecutadas en cada pixel dando como resultado un número muy grande de operaciones por segundo.

Para resolver el problema del procesamiento en tiempo real, el presente proyecto propone desarrollar un sistema usando un FPGA como plataforma para la adquisición de imagen y eliminación de ruido en la señal del video analógico generado por la CCD del microdurómetro. Un FPGA consiste de una matriz de bloques lógicos conectados por una red de interruptores. Tanto la red, como los bloques lógicos son reprogramables, permitiendo así aplicaciones específicas a nivel hardware, mientras al mismo tiempo mantiene la habilidad de cambiar la funcionalidad del sistema con facilidad. Un FPGA proporciona la flexibilidad de los procesadores de propósito general y la velocidad de procesamiento basada en hardware. Las mejoras

de rendimiento del FPGA comparado con los procesadores convencionales, es evitar la sobrecarga en cuanto a la decodificación de archivos y al inherente paralelismo de hardware digital que poseen.

Paralelismo en algoritmos de procesamiento de imagen existe en dos formas [20]: espacial y temporal. A nivel FPGA se tiene la gran ventaja que esta utiliza ambas formas. Se propone explotar esta característica del FPGA para eliminar ruido eléctrico de la señal de video analógica. La señal de video filtrada se procesará para su conversión a video digital, la cual será capturada en formato bmp y png.

1.3 Objetivos de la tesis

Estamos interesados en el desarrollo de un sistema de visión artificial que opera en tiempo real y de manera automática o semi-automática, para la adquisición, procesamiento y análisis de imágenes para pruebas de dureza Vickers, como objetivo final. Este interés se basa en la idea futura de aplicar las técnicas desarrolladas en este proyecto en aplicaciones industriales reales de manufactura, donde la muestra de material bajo estudio llega al microdurómetro dentro de un proceso automatizado. Entonces la muestra del material tiene que ser analizada automáticamente dentro de cierto límite de tiempo disponible. Por eso, el análisis del material tiene que ser optimizado en tiempo. En particular, eso es importante en el caso de que la muestra sea indentada en muchos puntos de la superficie a la vez, y todas las huellas generadas tengan que ser evaluadas dentro de un tiempo determinado. Sin embargo, estas aplicaciones reales industriales no están contempladas como parte de este proyecto doctoral, sino son parte de trabajo a futuro.

El presente proyecto doctoral plantea los siguientes objetivos:

- Desarrollo de un sistema de adquisición y almacenamiento de imagen basado en un dispositivo CCD del microdurómetro HM-125, un FPGA, y una interfaz gráfica para usuario. Eso involucra un estudio de las condiciones de ruido eléctrico y magnético, con el fin de encontrar las condiciones óptimas de adquisición, o, de incluir la aplicación de técnicas de procesamiento de señal para disminuir o eliminar ruido.
- Desarrollo de técnicas novedosas y adaptación de métodos conocidos del procesamiento de imágenes digitales para el procesamiento y el análisis de imágenes

de indentación Vickers, para diversos materiales. En particular, se deben adaptar o elaborar técnicas para segmentar la imagen con el fin de identificar la región de la (huella de) indentación, para detectar con precisión los cuatro vértices de la indentación de forma rombica, para determinar las longitudes de sus diagonales y así, y así, poder calcular la dureza del material y otras características.

- Implementación del nuevo algoritmo para procesar y analizar las imágenes de indentación Vickers. El proyecto doctoral constituye una fase de aprendizaje sobre el procesamiento y análisis de imágenes de indentación Vickers. Aunque en un futuro se pretende integrar todo un sistema de visión artificial en un mismo ambiente de programación que opera en tiempo real, en el proyecto doctoral, el nuevo algoritmo será implementado para equipo de cómputo y ambientes de programación estándar, es decir, computadora personal y software como Matlab, Delphi, o Python.
- Experimentación del sistema de adquisición y del nuevo método de procesar y analizar las imágenes, con un número considerable de imágenes reales de indentación. Comparación de resultados obtenidos por el sistema con resultados generados por un experto humano mediante mediciones manuales.
- Generación de un banco de imágenes de indentación Vickers, con anotaciones de expertos humanos referente a resultados de medición obtenida de manera tradicional manual.
- Publicación de los resultados obtenidos en este proyecto en revistas internacionales, y presentación de resultados en congresos internacionales.

1.4 Resultados y contribuciones de la tesis

Los objetivos planteados para este trabajo doctoral fueron cumplidos, como se reporta a continuación:

- Como resultado de esta tesis, fue desarrollado un sistema de adquisición y almacenamiento de las imágenes que provienen de la cámara CCD del microdurómetro HM-125, mediante un conjunto de módulos programados en Verilog HDL, para un FPGA Altera DE2-115 [1]. El sistema realiza de manera paralela el filtrado de la señal de video analógica y la adquisición de imagen en tiempo real.

Usando un lenguaje de programación orientada a objetos (Delphi), se elaboró una interfaz gráfica para el usuario para que a través del puerto USB de una computadora convencional se enlace con el FPGA para almacenar las imágenes en formato bmp y png. A través de la interfaz gráfica, el usuario configura los parámetros del filtro que serán procesados por el FPGA, logrando con ello eliminar ruido en la señal analógica de vídeo antes de ser convertido a señal digital, mejorando considerablemente la calidad de la imagen. Diferentes aspectos y etapas de desarrollo del sistema de adquisición de imagen y procesamiento de señal, fueron presentados en [15, 16].

El trabajo sobre el sistema de acondicionamiento de señal para eliminar ruido eléctrico y magnético, continuó una investigación sobre procesamiento de señales relacionado con la sensibilidad de sensores piezorresistivo [13]. La configuración de tres tipos de filtro con gran rango de parámetros, incorporada en el sistema de adquisición de imágenes de indentación, fue aplicada también para contribuir a trabajos de investigación en fisiología [14, 35].

- Esta tesis propone un nuevo algoritmo para el procesamiento y el análisis de imágenes para prueba de dureza Vickers, donde cada imagen de entrada contiene la huella de indentación de forma rómbica, cuya medida, posición y orientación exacta en la imagen son desconocidas. El algoritmo es presentado en detalle en el capítulo 5 de esta tesis, y fue dado a conocer en el artículo [17].

- El nuevo algoritmo para procesar y analizar imágenes de indentación Vickers fue implementado en Matlab.

- El sistema de adquisición y el nuevo método de procesar y analizar las imágenes, fueron experimentados con un gran número de imágenes reales de indentación, para las cuales, los resultados de aplicar el algoritmo fueron comparados con resultados obtenidos manualmente por un experto humano. Eso es reportado en el capítulo 6

de la tesis.

- Hasta la fecha de defensa de esta tesis, fueron adquiridas 230 imágenes, para formar un banco de imágenes de indentación Vickers, con anotaciones de un experto humano quien obtuvo los resultados de medición de manera tradicional manual. Gran parte de este banco, 185 imágenes, organizadas en cuatro grupos, con sus anotaciones de experto y resultados detallados de experimentación con el nuevo algoritmo, puede ser accesado en internet desde septiembre 2018 en [12]. El autor de esta tesis ha organizado el banco completo de 230 imágenes, para hacerlo pronto accesible online en un servidor de la Universidad Veracruzana.

Los resultados de la tesis fueron publicados en los siguientes artículos:

[14]: S.M. Domínguez-Nicolás, R. Juárez-Aguirre, A.L. Herrera-May, P.J. García-Ramírez, E. Figueras, E.A. Gutierrez-D., J.A. Tapia, A. Trejo, E. Manjarrez, “Respiratory magnetogram detected with a MEMS device”, *International Journal of Medical Sciences*, Vol.10, Issue 11, pp. 1445-1450, **2013**.

[35]: R. Juárez-Aguirre, S.M. Domínguez-Nicolás, E. Manjarrez, J.A. Tapia, E. Figueras, H. Vázquez-Leal, L.A. Aguilera-Cortéz, A.L. Herrera-May, “Digital signal processing by virtual instrumentation of a MEMS magnetic field sensor for biomedical applications”, *Sensors*, Vol.13, No.11, pp. 15068-15084, **2013**.

[30]: A.L. Herrera-May, J.A. Tapia, S.M. Domínguez-Nicolás, R. Juárez-Aguirre, E.A. Gutierrez-D., A. Flores, E. Figueras, E. Manjarrez, “Improved detection of magnetic signals by a MEMS sensor using stochastic resonance”, *PLOS ONE*, Vol.9, No.10, e109534, 8 pages, Oct. **2014**.

[15]: S.M. Domínguez-Nicolás, P. Wiederhold, “A real-time image acquisition system based on FPGA for hardness tests by indentation”, *Proceedings of Intern. Conf. on Computer Science and Artificial Intelligence (ICCSAI 2014, Wuhan, China, Dec. 2014)*, Paper ID ICCSAI-052, 7 pages, **2014**.

[16]: S.M. Domínguez-Nicolás, P. Argüelles-Lucho, P. Wiederhold, “FPGA based image acquisition and graphic interface for hardness tests by indentation”, *International Journal of Advanced Computer Technology*, Vol.6, No.6, pp. 6-16, **2016**.

[17]: S.M. Domínguez-Nicolás, P. Wiederhold, “Indentation image analysis for Vickers hardness testing”, *15th International Conference on Electrical Engineering, Computing Science, and Automatic Control (IEEE - CCE)*, Mexico City, Sept. 5-7, 2018, 6 pages, in press, **2018**.

1.5 Estructura de la tesis

Después de la Introducción, los capítulos 2 y 3 de la tesis se dedican al desarrollo del hardware y software, respectivamente, del sistema de adquisición de imagen basado en un FPGA. En el capítulo 3, se explica la elaboración de los módulos programados en lenguaje Verilog HDL, sin embargo, los módulos mismos pueden encontrarse en el Anexo A de la tesis.

El capítulo 4 explica la interfaz gráfica de usuario para la comunicación con el FPGA, la cual también es parte del sistema de adquisición de imagen.

El capítulo 5 contiene un análisis del estado del arte de técnicas de procesamiento y análisis de imágenes de indentación Vickers, y presenta el nuevo algoritmo desarrollado en este trabajo de tesis.

El capítulo 6 reporta resultados de experimentación de aplicar el nuevo algoritmo a las imágenes contenidas en el banco de 230 imágenes de indentación. Resultados muy detallados de eso son contenidos en tablas las cuales forman el Anexo B de la tesis.

La tesis termina con sus Conclusiones y algunas ideas para Trabajos Futuros, así como con su Bibliografía. Finalmente, en el Anexo C se reproducen las páginas portada de las publicaciones que reportan resultados de este trabajo doctoral.

Capítulo 2

Adquisición de imagen mediante un sistema de arreglos de compuertas programables (FPGA)

2.1 Introducción

La adquisición de imágenes en tiempo real es importante en una gran variedad de aplicaciones que van desde video vigilancia, gestión de tráfico, hasta aplicaciones en medicina y biología. La primera generación de diseño en sistemas de video implementaba sus algoritmos en software para dispositivos **DSP** (*procesador digital de señales*, DSP por sus siglas en inglés). Sin embargo, los DSP y otros procesadores estándar no pueden ser adaptados a los requerimientos de nuevas generaciones de aplicaciones de video de gran resolución, alta relación de frames y algoritmos complejos.

Los procesadores digitales de señales tienen ventajas para aplicaciones de video, incluyendo el posible uso de modernos lenguajes de programación, tal como C. Sin embargo, el número de instrucciones ejecutables en forma paralela es limitado en los DSP, ya que tienen un número limitado de registros, y las instrucciones en bytes son considerablemente pocas [53].

Por otra parte, los *arreglos de compuertas de campos programables*, abreviados como **FPGA** (por sus siglas en inglés), son capaces de ejecutar cientos o miles de operaciones de manera simultánea en paralelo sobre datos de capacidad de

256x23 bits [56]. Debido a las ventajas proporcionadas por el procesamiento de video de cómputo intensivo, los FPGA, o su combinación con DPS, se han convertido en la mejor opción para el diseño de sistemas de video avanzados [53]. Actualmente las tarjetas de desarrollo FPGA permiten al usuario implementar un rango amplio de diseño de circuitos. La interfaz de usuario de entradas y salidas (interruptores, botones, pantallas de cristal líquido, entrada de video, salida de video, etc.) constituye los componentes básicos de las tarjetas de desarrollo FPGA.

Para el trabajo de esta tesis, la tarjeta de desarrollo **Altera DE2 Cyclone II** fue elegida en base de su alta eficiencia para la implementación del procesamiento de video. Además, en este proyecto se requiere la realización en paralelo tanto del algoritmo de adquisición de imagen como también del procesamiento de señal cuyo objetivo es su aplicación al detector de video para la supresión de ruido eléctrico, lo cual será explicado en el capítulo 4 de esta tesis. La Figura 2.1 muestra la tarjeta Altera DE2 Cyclone II, con el nombre de cada una de sus partes [1].

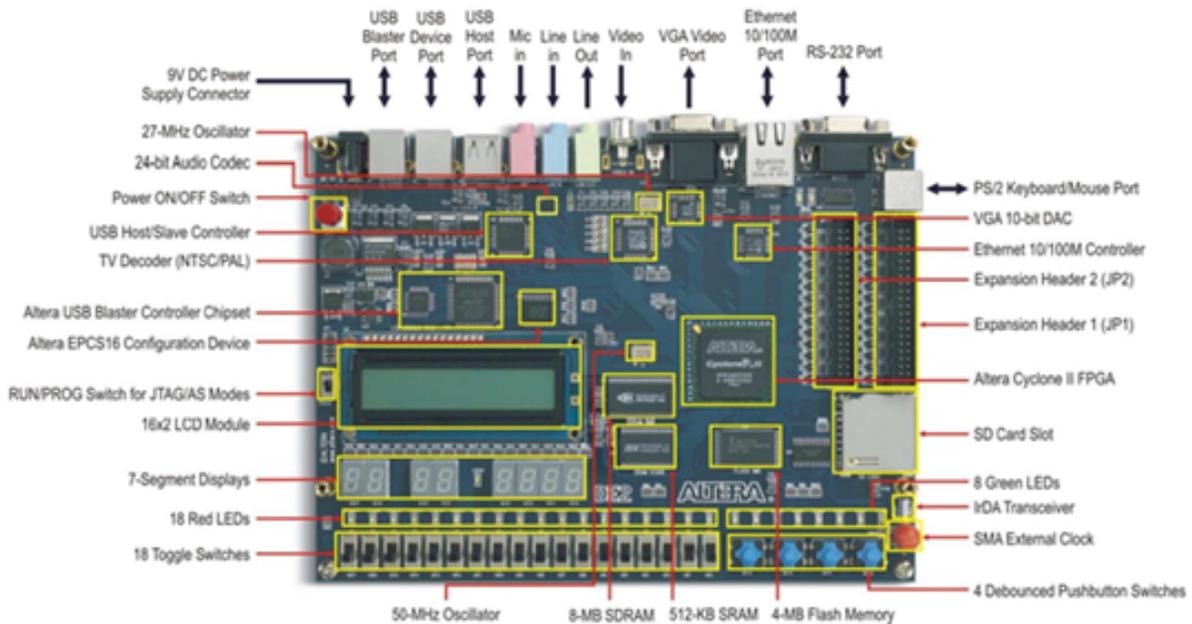


Figura 2.1: Tarjeta Altera DE2 Cyclone II.

Para la configuración de la tarjeta Altera DE2 Cyclone II, se aplicó el software **Quartus II Edición Web**, el cual es libre y escalable a actualizaciones. La tarjeta FPGA contiene una memoria EEPROM que permite almacenar la configuración de los datos. Cuando la tarjeta de desarrollo es energizada, la secuencia de bits de

configuración de la EEPROM es almacenada en el FPGA. Es posible elegir los datos de configuración no volátil de la memoria EEPROM usando el software Quartus II. También, el FPGA puede ser programado directamente en cualquier momento mientras la tarjeta de desarrollo se encuentra energizada.

2.2 Arquitectura del FPGA

En 1985, la empresa Xilinx introdujo los FPGA. Estos dispositivos son estructurados como un *arreglo de compuertas de circuito integrado de aplicación específica* (ASIC, por sus siglas en inglés). Por lo tanto, hoy en día, los FPGA se pueden utilizar en la creación de prototipos ASIC o en el lugar donde se utilizará un ASIC. Los FPGA son llamados campos programables porque pueden ser configurados por el usuario. Xilinx y Altera son las marcas de FPGA más conocidas en el mundo.

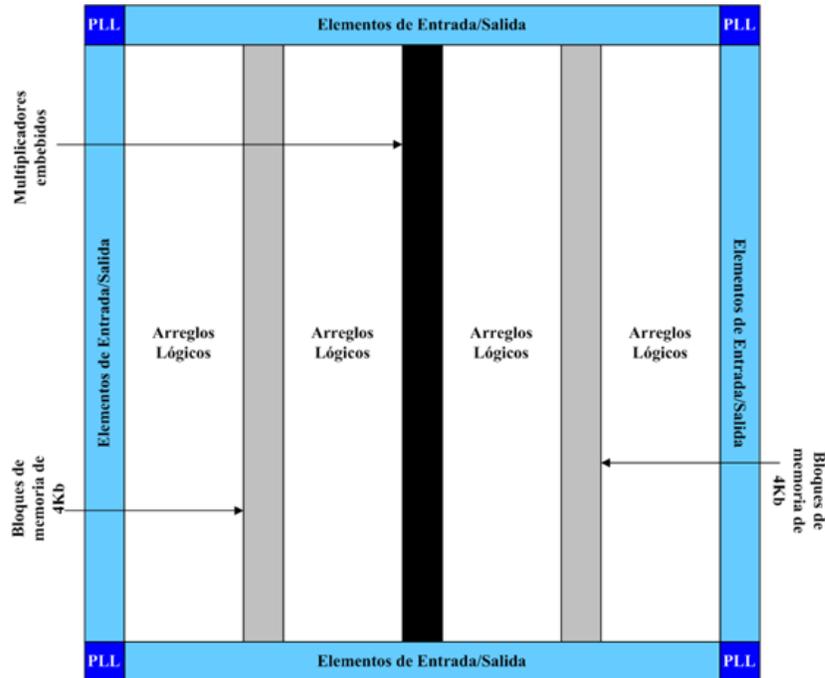


Figura 2.2: Diagrama a bloques del dispositivo Cyclone II 2C35.

La arquitectura del FPGA consiste de bloques lógicos configurables, bloques de entrada/salida configurables, interconexiones programables y circuitos de reloj para controlar la entrada de reloj de cada bloque lógico. También son disponibles fuentes lógicas, como las *unidades aritméticas lógicas* (ALU, por sus siglas en inglés),

bloques de memorias embebidas y decodificadores. La Figura 2.2 [1] muestra el diagrama a bloques del FPGA Cyclone II, el cual consiste de LABs, bloques de memoria de 4K, bloques de entrada/salida, circuitos de amarre de fase (PLL, por sus siglas en inglés) y multiplicadores embebidos.

2.3 Tarjeta de desarrollo FPGA

Uno de los principales objetivos de esta tesis, es desplegar la señal analógica proveniente de la cámara CCD del microdurometro hacia un monitor VGA usando un FPGA, y además, guardar un frame de la señal de video como imagen en formatos bmp y png. Con el fin de procesar las señales de video, el formato analógico debe ser convertido en formato de video digital. Como se observa en la Figura 2.3, una fuente de video analógica pasa a través de los siguientes bloques para obtener la señal de salida del video digital.

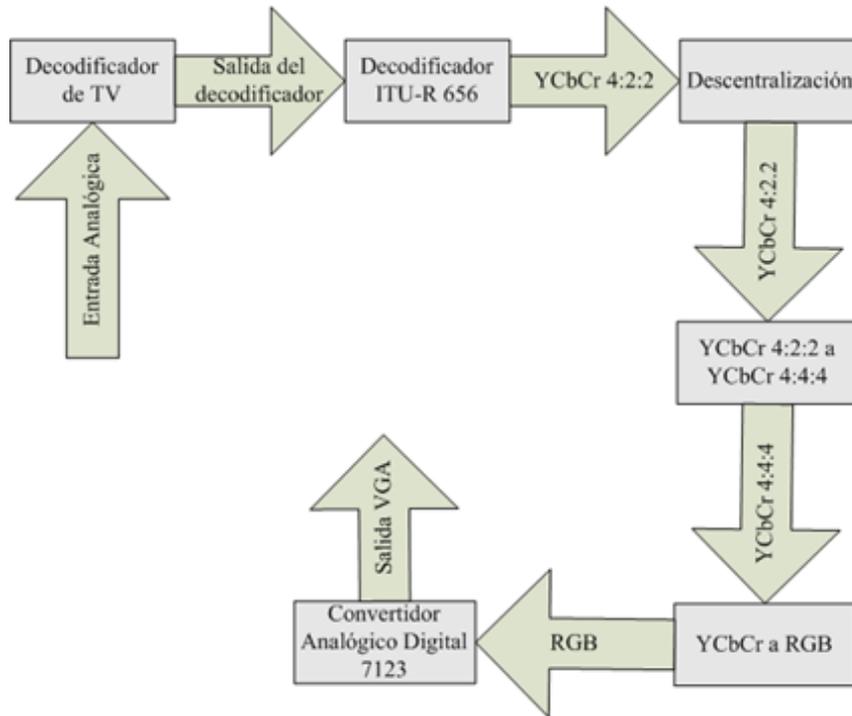


Figura 2.3: Conversión del video analógico a video digital.

2.4 Video analógico

Una señal de video analógico contiene luminancia (brillo) y crominancia (color), por ejemplo, puede ser compuesta de *banda base* (CVBS, por sus siglas en inglés), el *video separado* (video-s) y el *video componente*. En las partes video-s y componente, la luminancia y crominancia son transportadas en señales separadas. Por otra parte, para una señal de video compuesta, las componentes de luminancia (Y) y crominancia (C) son combinadas dentro de una sola señal. NTSC, PAL y SECAM son los formatos de video de difusión estándar, y son detectados por el decodificador de TV del sistema Altera DE2 Cyclone II.

2.4.1 Interfaz de video compuesta NTSC

La señal de video compuesta de banda base NTSC (*comité de sistema de televisión nacional*, NTSC por sus siglas en inglés) tiene una relación de 30 frames completos por segundo. La relación de frecuencia de actualización puede ser de 60 medio frames por segundo porque el NTSC tiene un formato de video entrelazado. En video entrelazado, cada frame completo está compuesto de dos campos; campos pares e impares, tales que son enviados en secuencia. Cada campo tiene 262.5 líneas. Por lo tanto, las líneas de campos pares e impares componen hasta 525 líneas por frame, 486 de estas líneas son visibles. La Figura 2.4 muestra la relación entre campos y frames [36].

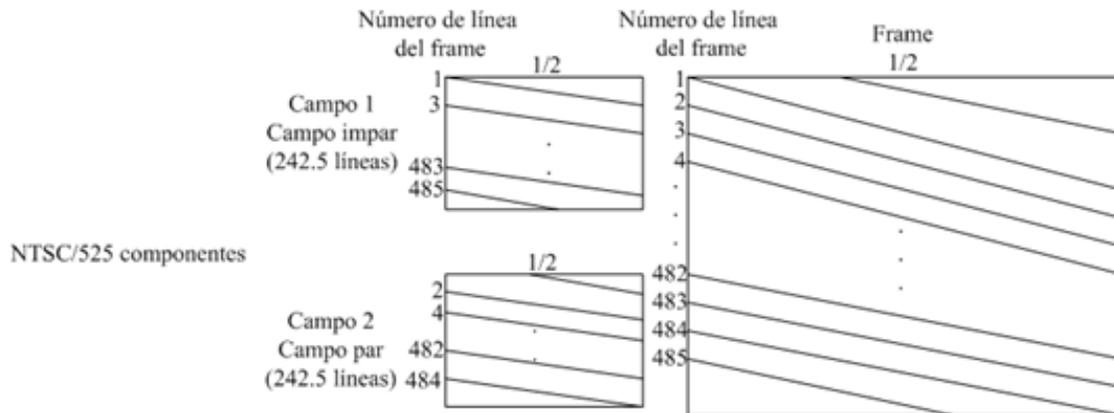


Figura 2.4: Relación entre campos y frames.

2.4.2 Interfaz de video compuesta PAL

PAL (*fase de línea alterna*, PAL por sus siglas en inglés) es una señal de video compuesta analógica. Un video en formato PAL tiene una relación de 50 mitades de frame por segundo. Cada frame completo contiene 625 líneas horizontales, 576 de estas líneas son visibles.

2.4.3 Interfaz de video compuesta SECAM

La interfaz de video compuesta SECAM (*color secuencial con memoria*, SECAM por sus siglas en francés) es utilizada en los países de Francia y Japón. La transmisión televisiva en SECAM, se forma escaneando la pantalla del televisor a 625 líneas y a una frecuencia de 25 frames por segundo. Este sistema es compatible con el sistema PAL, ya que utilizan los mismos formatos de escaneo y velocidades en los frames, la diferencia es la forma de cómo se codifica el color. Es decir, se pueden reproducir filmaciones en aparatos de sistema SECAM o a la inversa.

2.5 Video digital

Las señales digitales RGB y YCbCr son las más comunes en video digital. RGB es la versión digitalizada de la señal de video analógica RGB, YCbCr es la versión digitalizada de la señal analógica YPbPr [36].

Una señal de video analógica compuesta puede ser convertida a video digital a través de un ADC (*convertidor digital analógico*, ADC por sus siglas en inglés). Un decodificador de video embebido en un circuito integrado ejecuta esta conversión, así como operaciones de procesamiento de señal analógica. Las señales de televisión de video analógica de banda base (NTSC, PAL y SECAM) son convertidas a formato digital para ser procesadas y desplegadas. **ITU-R BT.656** describe un protocolo para transmitir video digital para señales de televisión no comprimidas PAL, SECAM, o NTSC. El estándar ITU-R BT.656 define la interfaz paralela y serial para transmitir el video digital YCbCr 4:2:2, y describe los fundamentos del proceso de la digitalización de video.

Hay muchos formatos de muestreo YCbCr tales como 4:4:4, 4:2:2, 4:1:1 y 4:2:0. El formato de video 4:2:2 significa que, la componente luma (Y) es digitalizada en cada posición de pixel (relación de muestreo 13.5 Mhz) y la componente croma (Cb, Cr)

son submuestreadas por un factor 2. Esta operación es llamada como el *submuestreo cromina*. Debido a limitaciones de almacenamiento y ancho de banda, compresión y optimización son necesarias para señales de video. El método de submuestreo cromina reduce el ancho de banda requerido. La cantidad de datos submuestreados de 16 bits para 4:2:2 YCbCr es 2/3 de los 24 bits de datos de 4:4:4 YCbCr. El humano no puede percibir la diferencia visual entre la imagen original y la submuestreada, debido a que el ojo humano es menos sensitivo a variaciones en color que en intensidad [36]. En la Figura 2.5 se puede observar el proceso de submuestreo cromina.

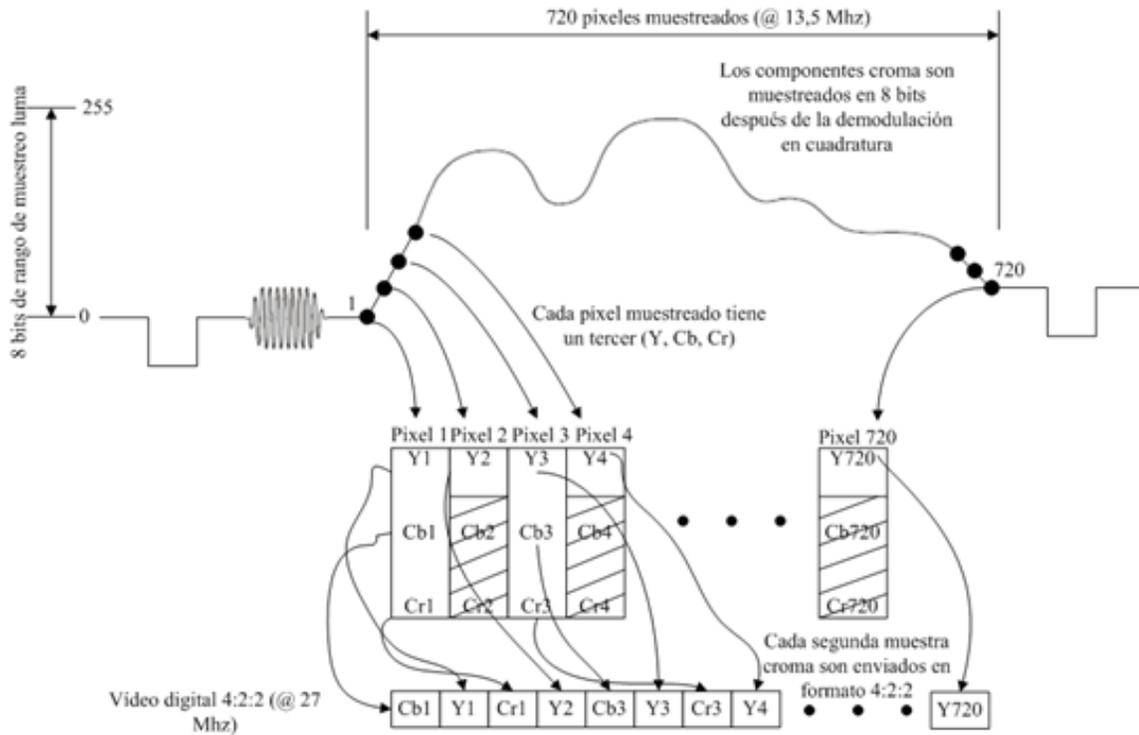


Figura 2.5: Proceso del submuestreo cromina.

2.5.1 Formato 2:4:1 a 4:4:4 YCbCr

Cada muestra tiene valores Y, Cb y Cr, donde los valores típicos son representados por 8 bits. Por lo tanto 24 bits son requeridos por cada muestra. La posición de las muestras Y, Cb, Cr para el formato 4:4:4 es mostrado en la Figura 2.6.

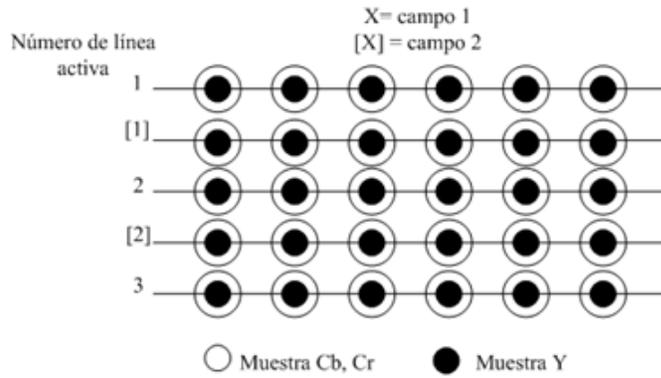


Figura 2.6: Posición de muestras Y, Cb, Cr en formato 4:4:4.

2.5.2 Formato YCbCr 4:2:2

Por cada dos muestras de Y horizontal, hay una muestra de Cb y Cr. Cada valor es de 8 bits. Por lo tanto, se requieren 16 bits por cada muestra. La Figura 2.7 muestra la posición de las muestras de Y, Cb, Cr para el formato 4:2:2.

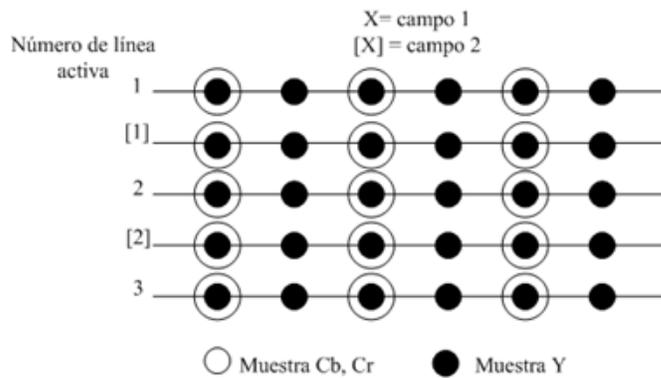


Figura 2.7: Posición de muestras Y, Cb, Cr en formato 4:2:2.

2.5.3 Formato YCbCr 4:1:1

Por cada cuatro muestras de Y horizontales, hay una muestra de Cb y Cr. Cada valor es de 8 bits. Por lo tanto, se requieren 12 bits por cada muestra 4:1:1. El formato YCbCr es usado en video de consumo y aplicaciones de compresión de video. La Figura 2.8 muestra la posición de las muestras de Y, Cb, Cr para el formato 4:1:1.

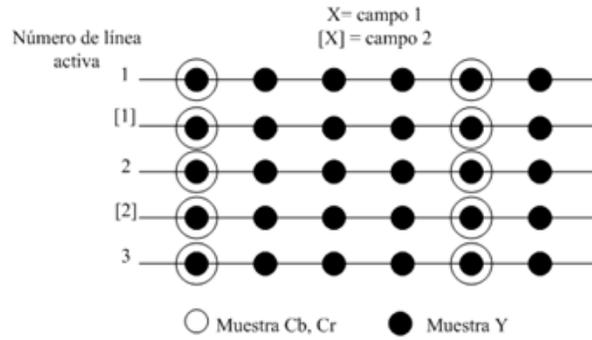


Figura 2.8: Posición de muestras Y, Cb, Cr en formato 4:1:1.

2.6 Sistema de adquisición de imagen (hardware)

2.6.1 Diseño general y componentes del sistema

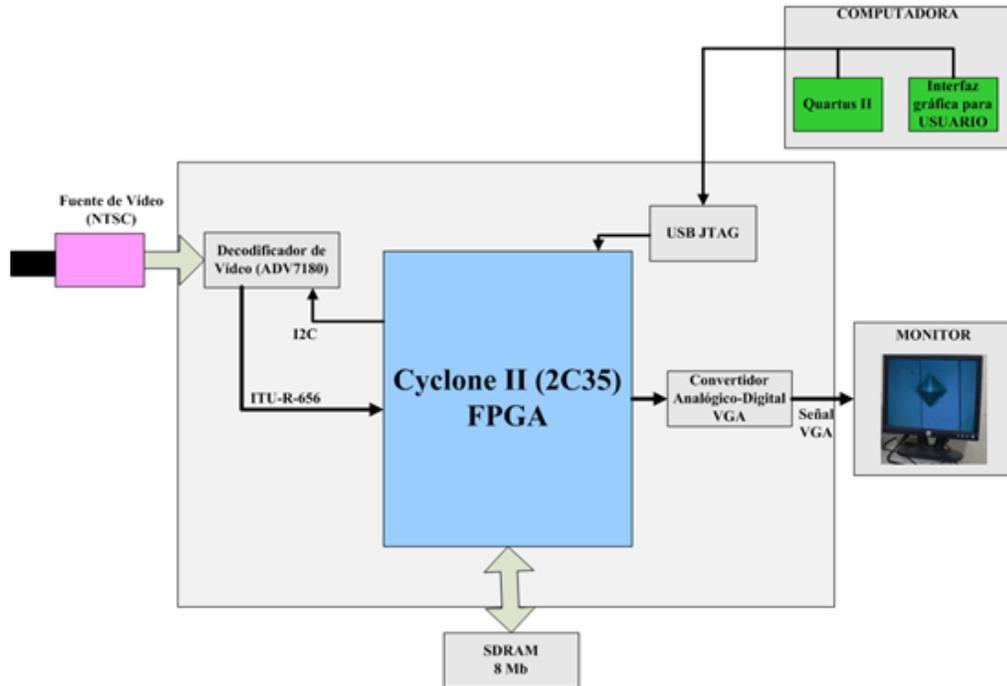


Figura 2.9: Diagrama a bloque del sistema de adquisición de imagen.

La Figura 2.9 muestra el sistema de adquisición de imagen que fue desarrollado en esta tesis y publicado en [15, 16]. En la figura se observa que la señal de video de la cámara es digitalizada usando el dispositivo decodificador ADV7181. Los frames de video capturados son almacenados en la SDRAM. Todas las operaciones

de procesamiento son ejecutadas en el FPGA Cyclone II. Después de la operación de procesamiento, el video es enviado a la salida VGA del sistema.

El decodificador de señal de televisión analógico ADV7181 consiste de un circuito integrado que detecta automáticamente y convierte una señal de televisión en datos de video con un muestreo de 4:2:2. La señal analógica de televisión detectada por el ADV7181, puede estar en los formatos NTSC, PAL o SECAM. El ADV7181 es compatible con un amplio rango de dispositivos de video, incluyendo DVD, cámaras de seguridad y vigilancia. Los registros del decodificador de televisión son programados por un bus serial I2C, el cual es conectado al sistema Altera DE2 Cyclone II, tal como se indica en la Figura 2.10 [1]. La asignación de terminales para su programación está listada en la Tabla 2.1.

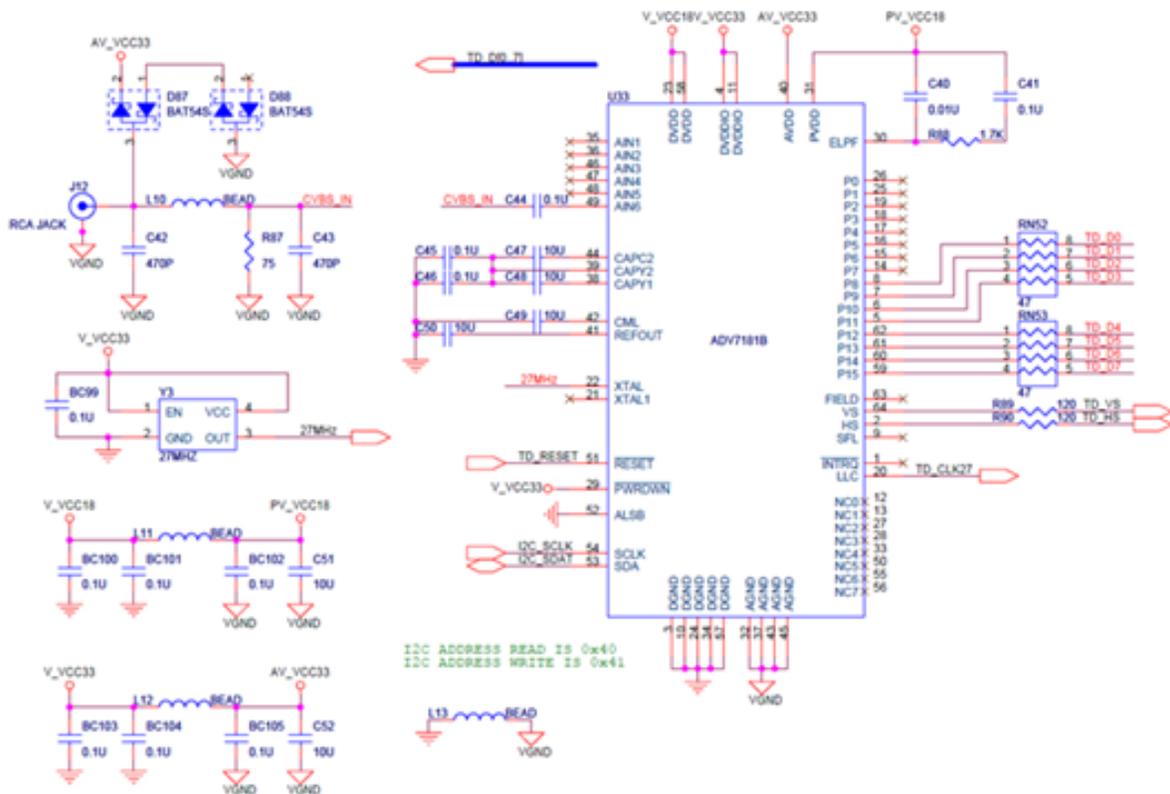


Figura 2.10: Esquema del decodificador de TV.

El FPGA 2C35 establece comunicación con una PC mediante configuración JTAG (estándar IEEE 1149.1). A través de este estándar, se configura los dispositivos y registros en el FPGA. El software Quartus II genera automáticamente

Tabla 2.1: Descripción de las terminales JTAG.

Nombre del Pin	Tipo	Descripción
TDI	Entrada de datos	Pin de entrada serial, así como para prueba y programación de datos. Los datos cambian en el flanco ascendente del pulso de reloj TCK. Si la interfaz JTAG no es requerida en el sistema, el circuito JTAG puede ser deshabilitado, conectando este Pin al suministro de energía (VCC).
TDO	Salida de datos	Pin de salida de datos serial para instrucciones, así como también para prueba y programación. Los datos de salida se modifican en el flanco descendente de la señal TCK. Si la interfaz JTAG no es requerida en el sistema, el circuito JTAG puede ser deshabilitado, dejando este Pin desconectado.
TMS	Selección de modo prueba	Pin de entrada que proporciona la señal de control para determinar las transiciones de estado de la máquina del controlador. Las transiciones ocurren en el flanco ascendente del TCK. Por lo tanto, el TMS será ajustado antes del flanco ascendente del TCK, el cual es evaluado en este mismo flanco.
TCK	Entrada de reloj	Algunas operaciones ocurren en el flanco ascendente, mientras otras ocurren en el flanco descendente. Si la interfaz JTAG no es requerida en el sistema, el circuito JTAG puede ser deshabilitado, conectando este Pin a la tierra del circuito (GND).

archivos SOF que son usados para cargar la configuración hacia el dispositivo FPGA vía cable serial.

El FPGA es diseñado tal que las instrucciones JTAG son precedidas sobre cualquier modo de configuración del dispositivo. Es decir, la configuración JTAG tiene prioridad sobre otros modos de configuración en el dispositivo. Por ejemplo, si se realiza una configuración del FPGA vía JTAG durante una configuración de un teclado vía USB, la configuración del teclado se interrumpe para dar prioridad a la configuración JTAG. El modo de operación de un dispositivo FPGA en comunicación con otros dispositivos vía JTAG, los cuales usan las terminales TDI, TDO, TMS y TCK, son definidas en la Tabla 2.1.

Durante la configuración del JTAG se utiliza un cable USB-Blaster para la descarga de los datos hacia el dispositivo. La Figura 2.11 [1] muestra el circuito para la comunicación del JTAG.

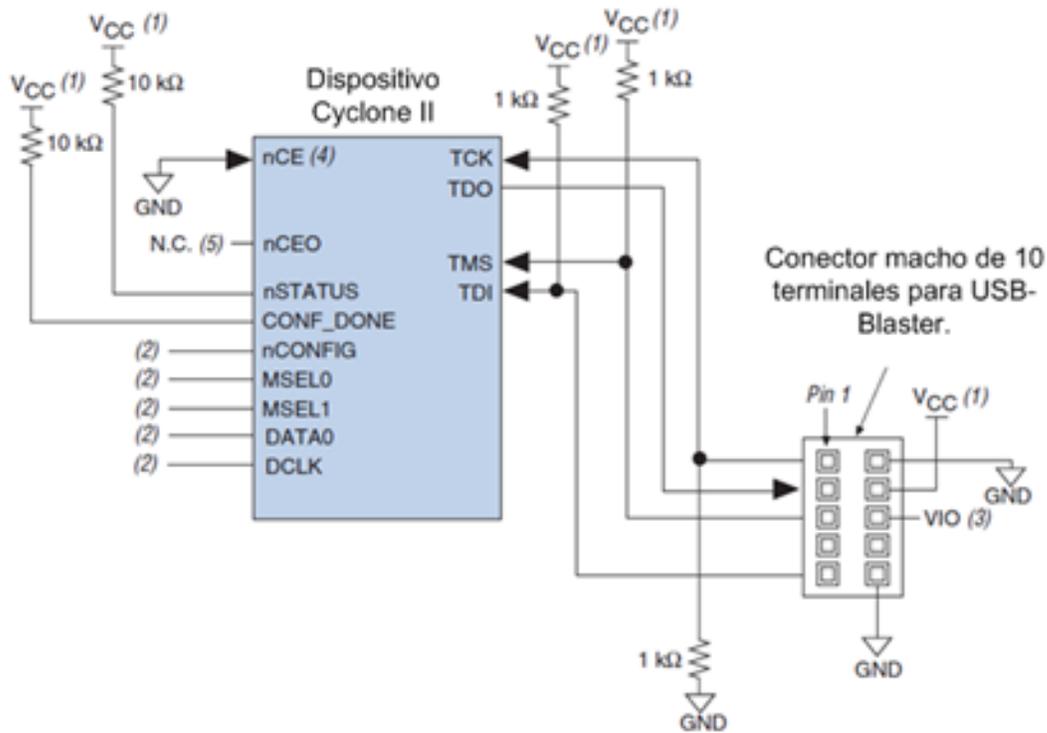


Figura 2.11: Circuito de comunicación del USB JTAG.

Para configurar ún solo FPGA en un JTAG, el software de programación pone a todos los dispositivos en modo BYPASS. En este modo, la programación de los datos del FPGA va desde la terminal TDI a la terminal TDO a través de un registro bypass. Este esquema habilita el software de programación para cargar o verificar la comunicación con el FPGA.

2.6.2 Salida VGA

El sistema Altera DE2 incluye un conector de 16 terminales para la salida VGA (*arreglo gráfico de video*, VGA por sus siglas en inglés) de la imagen. Las señales de sincronización son proporcionadas directamente por el FPGA 2C35 y el DAC (*convertidor analógico-digital*, DAC por sus siglas en inglés) ADV7123 de alta velocidad con resolución de 10 bits, el cual puede soportar una resolución de discretización

de hasta 1600x1200 píxeles a 100 Mhz. Un esquema de la conexión del DAC es mostrada en la Figura 2.12 [1].

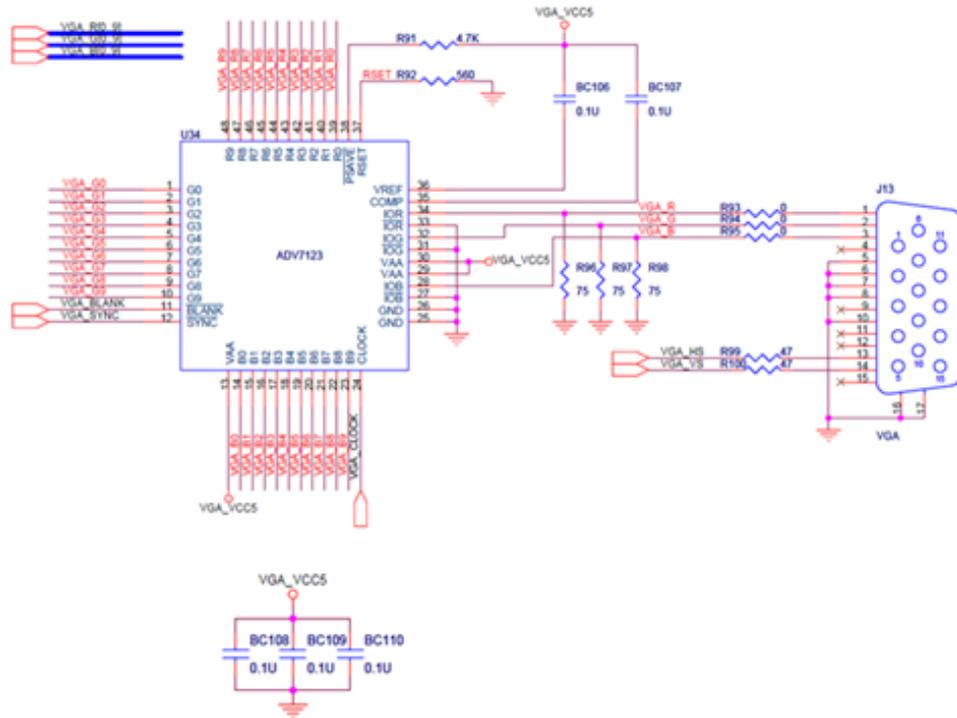


Figura 2.12: Esquema del VGA.

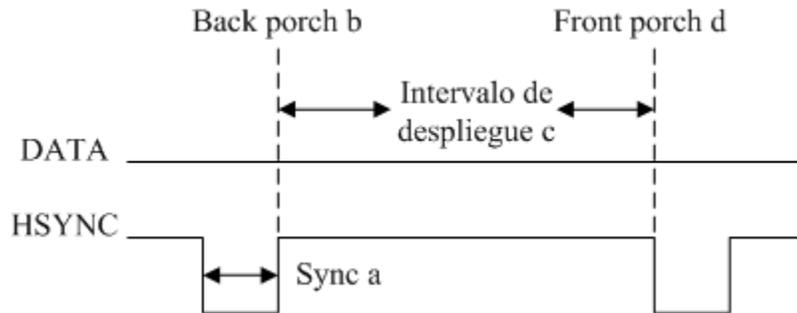


Figura 2.13: Especificación del tiempo en barrido horizontal del VGA.

Tabla 2.2: Especificación de tiempo horizontal en la salida VGA.

Modo VGA Configuración	Modo VGA Resolución (HxV)	a (μs)	b (μs)	c (μs)	d (μs)	Velocidad de envío de píxeles (MHz)
VGA (60Hz)	640x480	3.8	1.9	25.4	0.6	25 (640/c)
VGA (85Hz)	640x480	1.6	2.2	17.8	1.6	36 (640/c)
SVGA (60Hz)	800x600	3.2	2.2	20	1	40 (800/c)
SVGA (75Hz)	800x600	1.6	3.2	16.2	0.3	49 (800/c)
SVGA (85Hz)	800x600	1.1	2.7	14.2	0.6	56 (800/c)
XGA (60Hz)	1024x768	1.8	1.9	13.7	0.3	75 (1024/c)
XGA (85Hz)	1024x768	1.0	2.2	10.8	0.5	95 (1024/c)
1280x1024 (60Hz)	1280x1024	1.0	2.3	11.9	0.4	108 (1280/c)

La Figura 2.13 muestra el tiempo requerido para cada renglón (horizontal) que es desplegado en el monitor VGA. Un pulso bajo de tiempo a es aplicado para la sincronización de la entrada horizontal del monitor, el cual significa el final de un renglón de datos y el inicio del siguiente. La entrada de datos (RGB) en el monitor estarán en estado bajo por un periodo de tiempo llamado *back porch* b después de que ocurra un pulso *hsync*, el cual es seguido por el intervalo de despliegue c . Durante el intervalo de visualización de datos, las señales del RGB controlan cada pixel a través de los renglones que son mostrados en el monitor. Finalmente, hay un periodo de tiempo llamado *front porch* d donde las señales del RGB obtendrán un estado bajo hasta que el siguiente pulso *hsync* vuelva a ocurrir. El tiempo de sincronización vertical (*vsync*) es el mismo que se muestra en la Figura 2.13, excepto que, un pulso *vsync* se presentará al final de un frame y al inicio del siguiente, y los datos se refieren al conjunto de renglones en el frame (tiempo horizontal). Las Tablas 2.2 y 2.3 reportan la duración de los periodos de tiempo a , b , c y d para el tiempo horizontal y vertical, respectivamente. La asignación de terminales entre el FPGA y el ADV7123 es listada en la Tabla 2.4 [1].

Los dispositivos FPGA 2C35 tienen una arquitectura mediante bloques de arreglos lógicos en filas y columnas, para implementar lógica personalizada. Las columnas y renglones son interconectados por señales LAB (*bloques de arreglos lógicos*, LAB por sus siglas en inglés) embebidos en bloques de memoria y multiplicadores.

El arreglo lógico consiste de 16 LE (*elementos lógicos*, LE por su siglas en inglés) en cada LAB. Un LE es una pequeña unidad lógica capaz de proporcionar un efi-

Tabla 2.3: Especificación de tiempo vertical en la salida VGA.

Modo VGA Configuración	Modo VGA Resolución (HxV)	a (líneas)	b (líneas)	c (líneas)	d (líneas)
VGA (60Hz)	640x480	2	33	480	10
VGA (85Hz)	640x480	3	25	480	1
SVGA (60Hz)	800x600	4	23	600	1
SVGA (75Hz)	800x600	3	21	600	1
SVGA (85Hz)	800x600	3	27	600	1
XGA (60Hz)	1024x768	6	29	768	3
XGA (85Hz)	1024x768	3	36	768	1
1280x1024 (60Hz)	1280x1024	3	38	1024	1

ciente arreglo de funciones lógicas implementadas por el usuario. Los LAB son agrupados en renglones y columnas a través del dispositivo. El FPGA 2C35 tiene una densidad de 68416 LE. Los LE son las unidades lógicas más pequeñas en la arquitectura del FPGA. El LE es compacto y proporciona características para la eficiente utilización, como las siguientes:

- LUT (*tabla de consulta*, LUT por sus siglas en inglés) de cuatro entradas, con el cual se puede implementar cualquier función booleana de cuatro variables.
- Registro programable.
- Conexión en cadena para acarreo de bits.
- Una conexión en cadena de registros.
- La habilidad de manejar todos los tipos de interconexiones - renglón, columna, cadena de registros e interconexiones, en enlace directo.
- Soporte para paquete de registros.
- Soporte para retroalimentación de registros.

Cada registro programable de los LE puede ser configurado por operaciones en configuración D, T, JK o SR, correspondientes a datos, tiempo (es decir, reloj), señal para habilitar reloj, y mantener en blanco las entradas. Las señales que usan las terminales de entrada y salida de propósito general o cualquier entrada lógica, pueden controlar la señal de reloj de los registros y limpiar las señales de control. Para funciones de lógica combinacional, la salida LUT controla directamente a las salidas LE.

Tabla 2.4: Asignación de terminales del ADV7123.

Nombre de la señal	Número del Pin del FPGA	Descripción
VGA_R[0]	PIN_C8	VGA Red[0]
VGA_R[1]	PIN_F10	VGA Red[1]
VGA_R[2]	PIN_G10	VGA Red[2]
VGA_R[3]	PIN_D9	VGA Red[3]
VGA_R[4]	PIN_C9	VGA Red[4]
VGA_R[5]	PIN_A8	VGA Red[5]
VGA_R[6]	PIN_H11	VGA Red[6]
VGA_R[7]	PIN_H12	VGA Red[7]
VGA_R[8]	PIN_F11	VGA Red[8]
VGA_R[9]	PIN_E10	VGA Red[9]
VGA_G[0]	PIN_B9	VGA Green[0]
VGA_G[1]	PIN_A9	VGA Green[1]
VGA_G[2]	PIN_C10	VGA Green[2]
VGA_G[3]	PIN_D10	VGA Green[3]
VGA_G[4]	PIN_B10	VGA Green[4]
VGA_G[5]	PIN_A10	VGA Green[5]
VGA_G[6]	PIN_G11	VGA Green[6]
VGA_G[7]	PIN_D11	VGA Green[7]
VGA_G[8]	PIN_E12	VGA Green[8]
VGA_G[9]	PIN_D12	VGA Green[9]
VGA_B[0]	PIN_J13	VGA Blue[0]
VGA_B[1]	PIN_J14	VGA Blue[1]
VGA_B[2]	PIN_F12	VGA Blue[2]
VGA_B[3]	PIN_G12	VGA Blue[3]
VGA_B[4]	PIN_J10	VGA Blue[4]
VGA_B[5]	PIN_J11	VGA Blue[5]
VGA_B[6]	PIN_C11	VGA Blue[6]
VGA_B[7]	PIN_B11	VGA Blue[7]
VGA_B[8]	PIN_C12	VGA Blue[8]
VGA_B[9]	PIN_B12	VGA Blue[9]
VGA_CLK	PIN_B8	VGA Clock
VGA_BLANK	PIN_D6	VGA BLANK
VGA_HS	PIN_A7	VGA H-SYNC
VGA_VS	PIN_D8	VGA V-SYNC
VGA_SYNC	PIN_B7	VGA SYNC

Cada LE tiene tres salidas que controlan la elección de los recursos para los registros de columnas y renglones del FPGA. La LUT o salida de registro puede manejar estas tres salidas de manera independiente. Dos salidas LE controlan columnas o renglones y enlazan directamente la conexión de los recursos y una controla la interconexión de los recursos locales, permitiendo a la LUT manejar una salida mientras los registros controlan otras salidas. Esta característica de control de los registros mejora la utilización del FPGA, debido a que los LE pueden controlar los registros y la LUT controla funciones no relacionadas.

2.6.3 Bloques de arreglos lógicos (LAB)

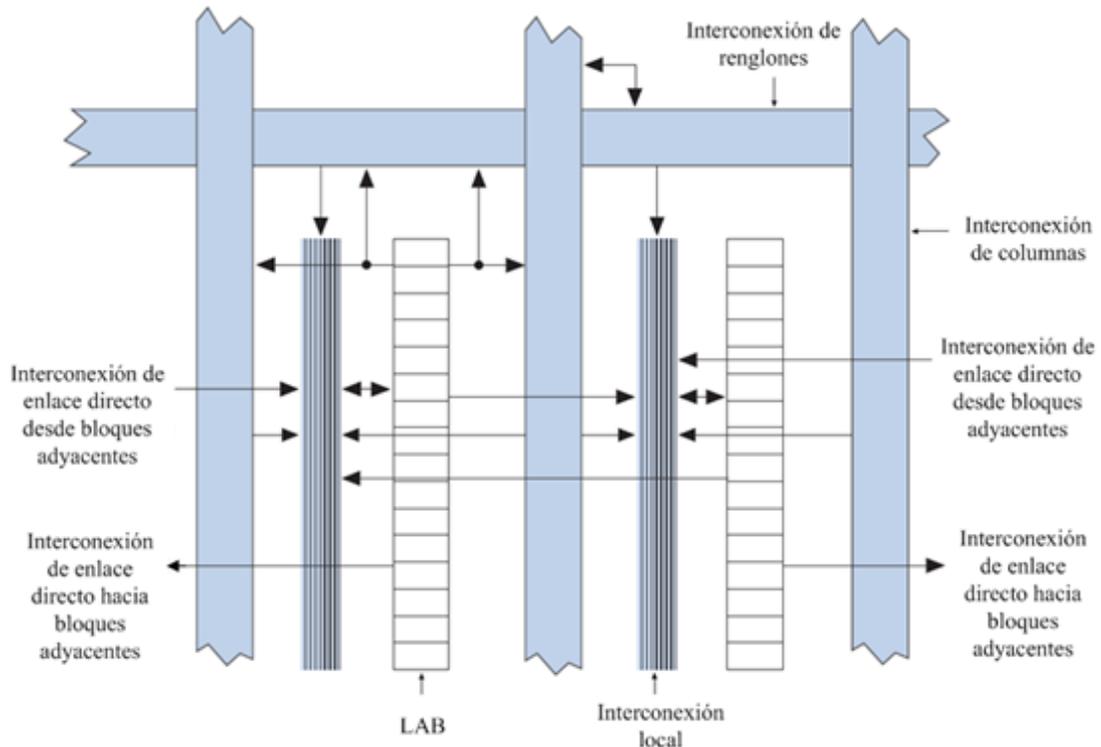


Figura 2.14: Estructura LAB del Cyclone II.

Cada LAB tiene los siguientes componentes:

- 16 LE,
- señales de control LAB,
- cadenas de acarreo de LE,

- cadenas de registro,
- interconexión local.

La interconexión local transfiere señales entre LEs en el mismo LAB. Las conexiones entre cadenas de registro transfieren la salida de uno de los registros LE al registro adyacente LE dentro de una LAB. El compilador Quartus II asocia lógicas con un LAB o LABs adyacentes, permitiendo el uso de registros locales y conexiones de cadenas entre registros para un mejor rendimiento de los recursos a nivel cómputo y hardware, tal como se muestra en la Figura 2.14 [1].

2.6.4 Interconexiones LAB

La interconexión local de una LAB puede controlar los LEs con el mismo LAB, y es determinada por interconexiones de columnas y renglones y salidas de cada LE con el mismo LAB. Los bloques LAB, PLL y memoria RAM de 4 kbytes, así como multiplicadores embebidos, pueden controlar una interconexión local de un LAB a través de una conexión de enlace directo. La conexión de enlace directo minimiza el uso de interconexión de renglones y columnas, proporcionando alta eficiencia y flexibilidad. Cada LE puede controlar 48 LE's a través de una interconexión de enlace directo y local. La Figura 2.15 muestra la conexión de enlace directo [1].

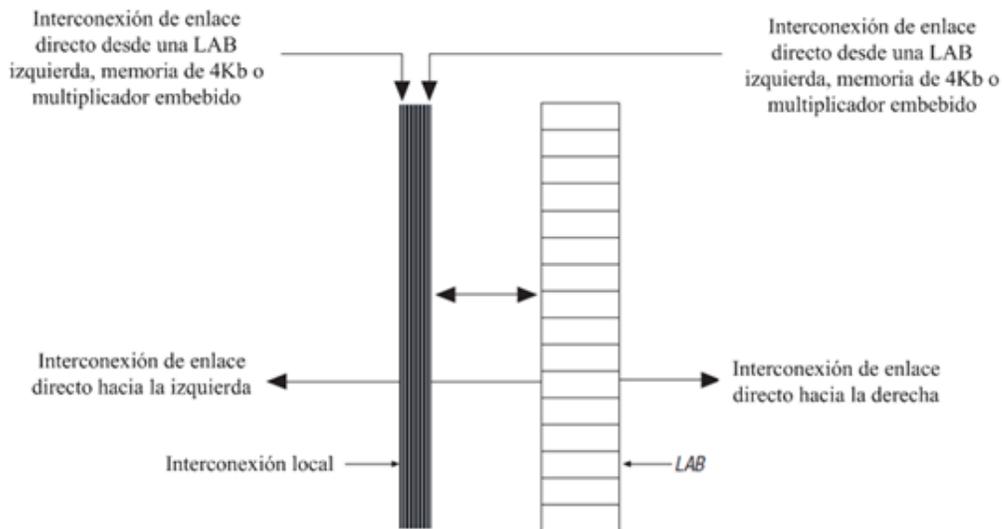


Figura 2.15: Estructura LAB del FPGA Cyclone II.

2.6.5 Señales de control de un LAB

Cada LAB contiene lógica dedicada a los señales de control hacia los LE, consistiendo en lo siguiente:

- dos señales de reloj,
- dos señales para habilitar los pulsos de reloj,
- dos señales para limpiar registros asíncronos,
- una señal para limpiar registros síncronos,
- una señal para cargar datos síncronos.

Esto da un máximo total de siete señales de control. Cuando se utiliza la carga síncrona del LAB, las señales *clkena* del *labclk1* no son disponibles. Además, la carga síncrona y el paquete de registro no se pueden usar de manera simultánea. Cada LAB puede tener hasta cuatro señales de control no globales. Se pueden utilizar señales de control adicionales de un LAB, siempre cuando sean globales. Las señales síncronas de carga y descarga de datos en registros, son de utilidad para implementar contadores y otras funciones. Cada LAB puede usar dos señales de reloj y dos señales de control para habilitar los registros de reloj. Cada reloj de un LAB y su respectiva señal de control son conectados para habilitar una entrada de reloj. Por ejemplo, cualquier LE en un LAB particular usa la señal *labclk1* y también usa *labclkena1*. La Figura 2.16 muestra el circuito que genera la señal de control de un LAB [1].

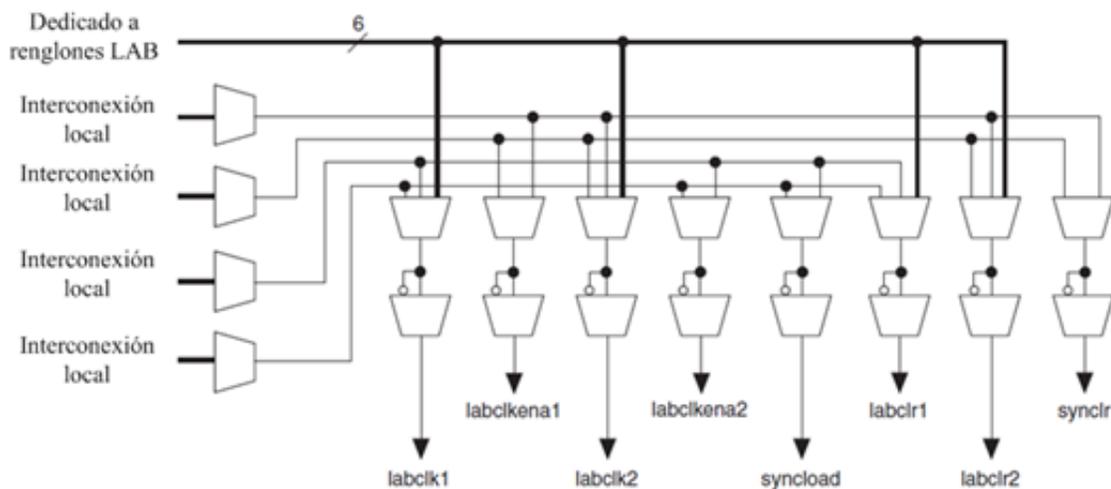


Figura 2.16: Señales de control de un LAB.

2.6.6 Unidad de memoria

En el dispositivo FPGA se encuentran embebidas los bloques de memoria. La memoria consiste en columnas de 4 kb las cuales pueden ser configuradas para proporcionar varios bloques de memoria, tales como: RAM, FIFO (primera entrada - primera salida) y bloques de memoria ROM. Los bloques de memoria de 4 kb proporcionan más de 1 Mbit de RAM a una operación de 250 MHz. Los bloques de memoria de 4 kb proporcionan las siguientes características:

- más de 1 Mbit de RAM disponible sin reducir elementos lógicos,
- 4,096 bits de memoria por bloque (4,608 bits por bloque incluyendo paridad),
- posibilidad de configuración de puertos,
- operación de lectura y escritura (dos de lectura o dos de escritura),
- un byte de enmascaramiento de entrada de datos durante el proceso de escritura,
- inicialización de archivo para precarga del contenido de memoria en modos RAM y ROM,
- velocidad de operación de hasta 250 Mhz.

La Figura 2.17 muestra las señales de control, de reinicio de datos y de reloj, implementadas en el bloque de memoria del FPGA [1]. La señal de control *clock enable* no tan solo controla la entrada y salida de registros, sino también al reloj para la entrada de datos hacia el bloque de memoria. Hay tres formas para reiniciar los registros en el bloque de memoria de 4 kb: 1) quitar la energía hacia el sistema FPGA, 2) usar la señal *aclr* para solo registros de salida, o 3) reiniciar todo el dispositivo usando la opción *DEV_CLRn*.

Para asegurar la integridad de los datos, implementando lógica en los LE, los bloques de memoria de 4 kb soportan la entrada de habilitación de bytes, tal que enmascaran datos de entrada para que bytes específicos sean almacenados en la memoria. Las señales de *wren* (para habilitar escritura), junto con la de *byte enable* (para habilitar bytes), controlan la operación de escritura en el bloque de memoria RAM. El valor predeterminado para la entrada *byteena* es en estado alto, por lo que en este caso, la escritura es controlada por la señal *wren*.

Los bloques de memoria de 4 kb del FPGA almacenan datos en paquete. Se puede implementar dos bloques de memoria de puerto único en un solo bloque, bajo las siguientes condiciones:

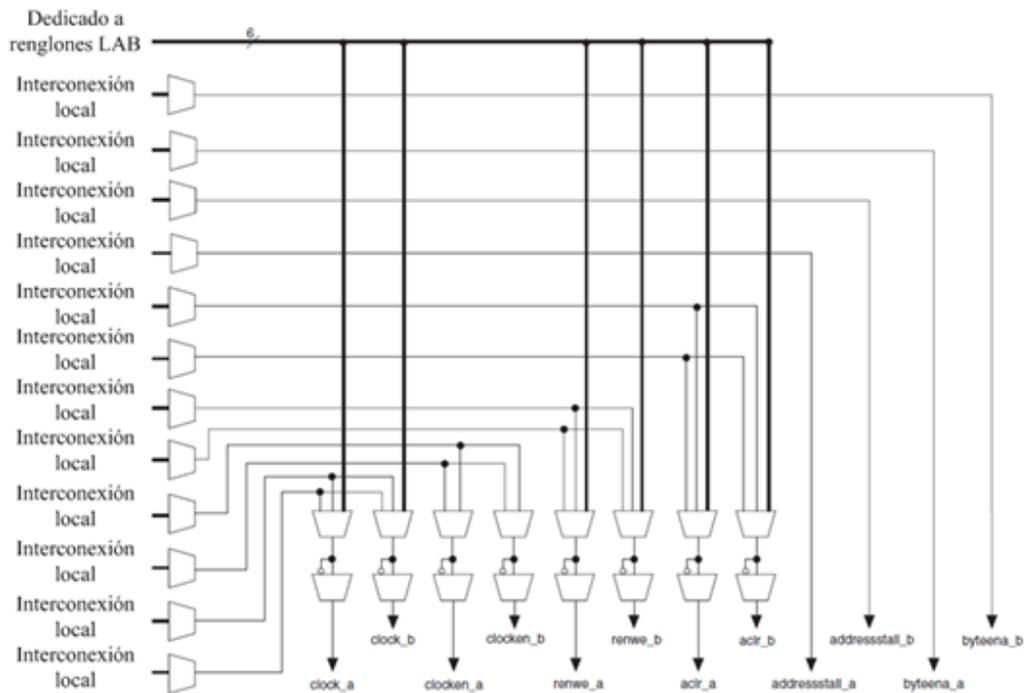


Figura 2.17: Señal de control de bloques de memoria del FPGA.

- La medida de cada uno de los dos bloques independientes es menor o igual que la mitad de los 4 kb. La cantidad de datos máxima para cada bloque independiente es de 18 bits.
- Cada una de las señales de los bloques de memoria es configurado para una entrada de reloj.

Los bloques de memoria mantienen la dirección de memoria de la entrada de reloj que lo habilita. Cuando los bloques de memoria son configurados en modo dual, cada puerto tiene su propia dirección de memoria de la señal de reloj que lo sincroniza.

La Figura 2.18 muestra el diagrama a bloques de una dirección de memoria de la señal que habilita la entrada de reloj. La dirección da la salida del registro es retroalimentada a su entrada a través de un multiplexor. La salida del multiplexor es seleccionado por el registro de memoria del reloj habilitado (*addressstall*). El bloque de direcciones es habilitado cuando la señal *addressstall* está en estado alto. La salida del registro de dirección estará constantemente aplicado a la entrada del registro hasta que la señal *addressstall* se encuentre en estado bajo.

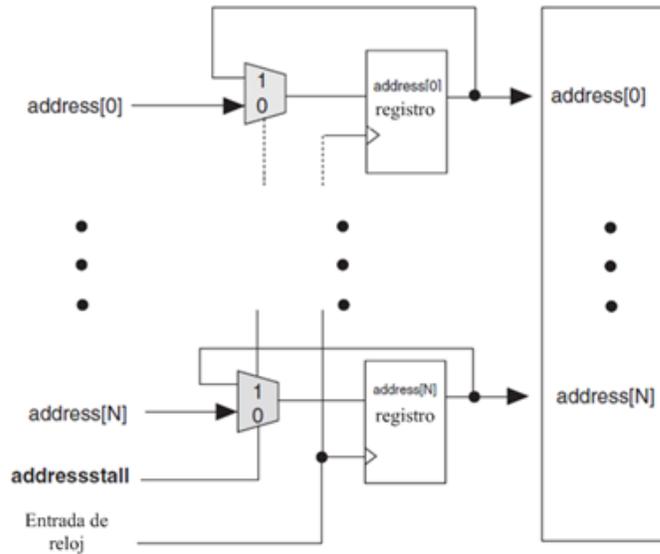


Figura 2.18: Habilitación de la entrada de reloj del registro de memoria del FPGA.

El sistema Altera DE2 Cyclone II utiliza los siguientes bloques de memoria de 4 kb:

- puerto único,
- puerto dual,
- FIFO buffers,
- memoria SDRAM.

2.6.7 Modo de puerto único

El modo de puerto único no soporta operaciones de lectura y escritura de manera simultánea. La Figura 2.19 [1] muestra la configuración de memoria de puerto único para los bloques de memoria del FPGA 2C35.

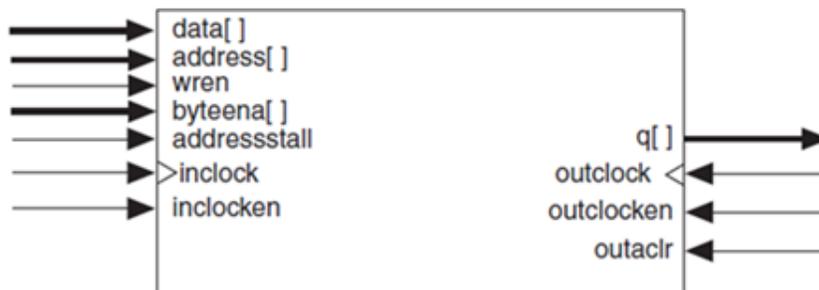


Figura 2.19: Configuración de memoria de puerto único.

En el modo de puerto único, las salidas están en modo lectura durante el proceso de escritura. Esto significa que, durante la operación de escritura, los datos que se escriben en la memoria RAM fluyen a través de la salida del mismo bloque de memoria. Cuando la salida de los registros son bloqueados, el nuevo dato es disponible en el flanco ascendente del ciclo de reloj en el cual fue escrito.

2.6.8 Modo de puerto dual simple

El modo de puerto dual soporta simultáneamente operaciones de lectura y escritura, la Figura 2.20 [1] muestra su configuración de memoria.

En el modo de puerto dual, los bloques de memoria tienen una señal de *write enable* y una de *read enable*. Para ambas acciones, el bloque de memoria no utiliza un puerto para borrar datos en el registro de la memoria. Cuando la señal *read enable* es activada, el dato actual es retenido en el puerto de salida. Si la opción *read enable* es activada durante una operación de escritura con la misma dirección de memoria seleccionada, la salida del puerto dual de la memoria RAM será el dato viejo almacenado en la dirección de memoria.

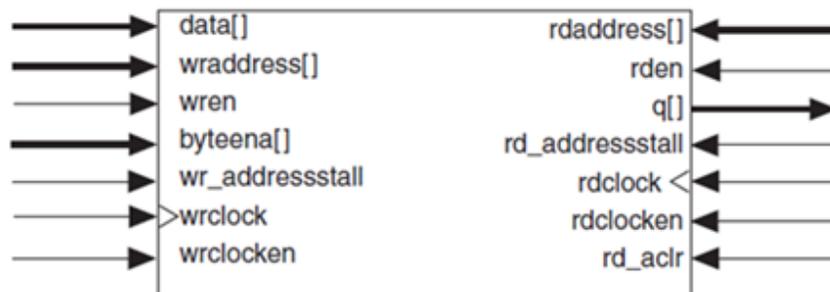


Figura 2.20: Configuración de memoria de puerto dual.

2.6.9 Modo Buffer FIFO

Un buffer FIFO con entrada de reloj único o reloj dual puede ser implementado en los bloques de memoria. El buffer FIFO con reloj dual es de utilidad cuando se transfieren datos de un registro de memoria a otro registro y controlado por pulso de reloj, es decir, de manera síncrona. Es por eso que todas las configuraciones de memoria FIFO tienen entrada síncrona. No obstante, las salidas de buffer FIFO son a través de lógica combinatorial, por lo que no usan registros.

2.6.10 Memoria SDRAM

La SDRAM es una arquitectura de memoria que transmite y recibe datos a la velocidad doble de la de reloj. Estos dispositivos transfieren datos en el flanco ascendente y descendente de un pulso de reloj. La SDRAM es una arquitectura de memoria capaz de transferir datos en una relación de 533 Mega bits por segundo.

Las terminales del dispositivo tales como datos (DQ), datos que cambian su valor en un ciclo de reloj (DQS), comandos y terminales de dirección de memoria, son enviadas a las memorias del FPGA y capturadas al doble de la señal de reloj del sistema. La transferencia de datos es realizada en el flanco positivo y negativo de la señal de reloj.

2.6.11 Decodificador de TV a sistema VGA

El módulo de decodificación de señal de TV con salida VGA es responsable de convertir la señal de TV en un formato que pueda ser mostrado en un monitor o pantalla con entrada VGA. El módulo es responsable también de almacenar un frame local en un bloque de memoria SRAM y comunicarse con el FPGA para enviar y recibir imágenes. La Figura 2.21 muestra un diagrama a bloques del convertidor de TV a VGA que fue implementado para este trabajo de tesis, además, su desarrollo fue publicado en [15, 16].

La conversión es realizada por el decodificador de TV (ADV7181), el cual adquiere una señal de TV de 50 a 60 Hz. La adquisición de imagen en este proyecto fue realizada a 60 Hz usando el formato NTSC. El decodificador de TV es conectado a una configuración I2C la cual ajusta y sincroniza al decodificador. La salida del decodificador es enviada al módulo ITU R656 que fue programado en el FPGA. Este módulo se encarga del muestreo de píxeles y extrae la información de color en formato YCbCr que describe luminancia y crominancia. Sin embargo, para que esta información sea utilizada por la pantalla VGA, antes debe ser convertida al formato RGB, el cual codifica los valores de los píxeles en tres canales de color. El resultado de la conversión de YCbCr a RGB es enviado al convertidor analógico para que esta información sea usada directamente por el VGA. La sincronización adecuada se logra mediante el temporizador del VGA, el cual usa las señales de sincronización de TV y proporciona la sincronía vertical y horizontal para el VGA. Estas señales de sincronía también son utilizadas para mantener el conteo vertical y horizontal que

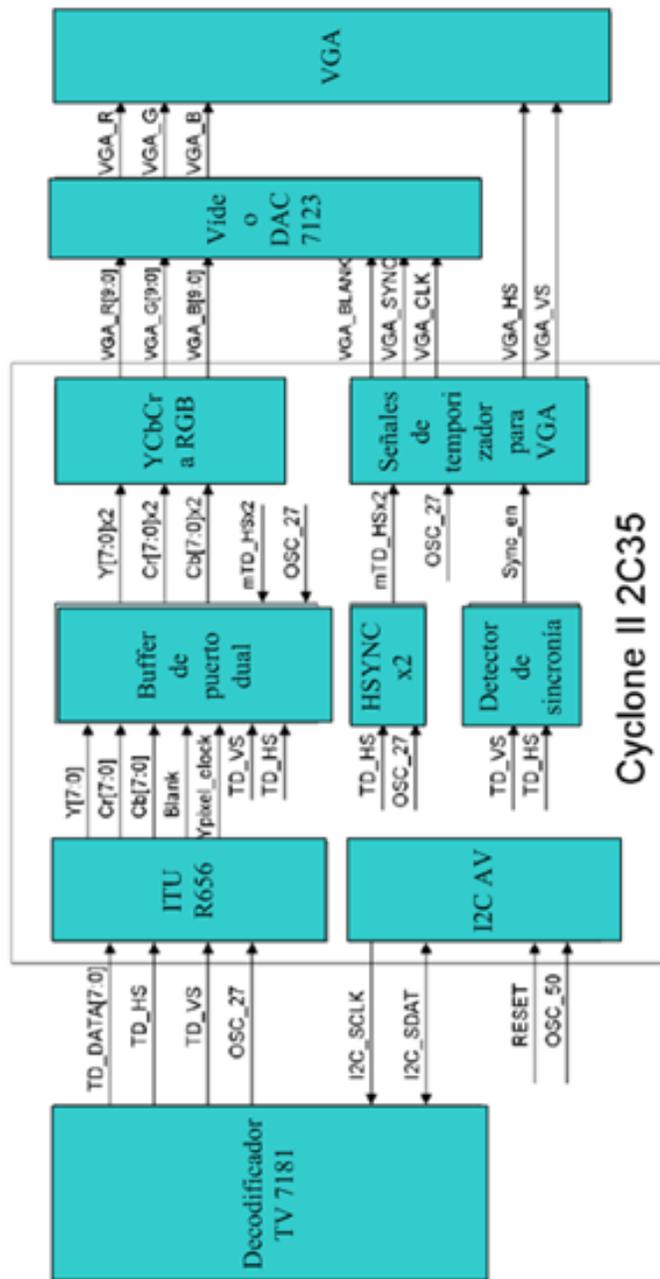


Figura 2.21: Diagrama a bloques del convertidor de TV a VGA.

Tabla 2.5: Protocolo de comunicación.

Valor del protocolo	información del envío
0	ignorado
1	respuesta a la imagen enviada
2	conocimiento de un pixel recibido
3	paquete que contiene un pixel

indica exactamente en qué posición de la pantalla se encuentra un pixel. Cuando el usuario presiona *Enter*, la imagen del monitor VGA es capturada como primera instancia en la memoria SRAM para que posteriormente sea enviada a la memoria SDRAM en donde quedará almacenada la imagen. Como resultado del proceso, se logra enviar imágenes vía remota hacia una computadora.

Cuando el usuario selecciona la opción de “*save actual video image*” en la **GUI** (*interfaz gráfica de usuario*, GUI por sus siglas en inglés), la imagen actual es enviada vía remota a través del puerto serial de la tarjeta Altera DE2. El FPGA comienza enviando una señal de petición indicando la intención de enviar una imagen. Cuando la computadora recibe la señal, se produce una interrupción, y la computadora carga una imagen, enviando posteriormente un mensaje de confirmación. El FPGA entonces envía la imagen, pixel por pixel. Cada paquete se divide en secciones de un byte. La primera sección contiene al valor del protocolo, el cual indica la naturaleza del paquete de datos, ya sea una solicitud, un acuse de recibo o un paquete de datos. Los paquetes contienen información de un pixel, almacenado en los valores iniciales de x y y , seguido por el número de bytes de la información de los píxeles que ha sido enviado. Esto proceso es continuado hasta por 800 bytes de información de píxeles. Dado que la información de cada pixel es representada por 16 bits, cada pixel ocupa dos secciones en el paquete, vea la Tabla 2.5 para observar el protocolo de comunicación.

Una vez que la computadora recibe un paquete, construye la imagen teniendo en cuenta los valores de inicio de x y de y , colocando cada pixel en la ubicación correcta. Una vez realizada la transferencia de la imagen, el usuario puede ver la imagen, junto con otras imágenes adquiridas y procesadas por el FPGA.

Capítulo 3

Adquisición de imagen mediante módulos programados en Verilog para el FPGA Altera DE2-115

3.1 Introducción

Actualmente existe una gran cantidad de algoritmos para la adquisición de imagen, en su mayoría son implementados en Matlab [60], Open CV [69], y usando lenguajes de programación como C o Python. No obstante, hay sistemas de adquisición de imagen implementados en FPGA, por ejemplo, usando el procesador NIOS de Altera [70].

El sistema de adquisición que se desarrolló en este trabajo de tesis, es totalmente diferente: no usamos un algoritmo estándar, tampoco el procesador NIOS, el sistema es implementado en FPGA usando **Verilog HDL** (*lenguaje de descripción de hardware*, HDL por sus siglas en inglés). El FPGA no necesita de un procesador externo, lo cual es una gran ventaja, haciendo posible que el sistema de adquisición de imagen trabaje en tiempo real.

El paquete computacional CAD (*diseño asistido por computadora*, CAD por sus siglas en inglés) facilita la implementación de circuitos lógicos usando dispositivos lógicos programables, tales como los FPGA. La herramienta CAD soporta el ingreso de diseños digitales en base a esquemáticos, los cuales se desarrollan conectando compuertas digitales básicas o de bibliotecas, en un ambiente visual. Sin embargo,

a medida que el sistema aumenta su tamaño, cada vez se vuelve más difícil la descripción por esquemáticos.

En contraste, los lenguajes descriptivos de hardware permiten diseñar en forma abstracta, complejos sistemas digitales para ser implementados en dispositivos programables como un FPGA. Eso es realizable, por ejemplo, describiendo el sistema digital mediante un código Verilog HDL.

El ambiente de programación Verilog existe desde 1984, luego de varios años de uso, se ha estandarizado y su última versión data del año 2001 [64]. Verilog HDL es un lenguaje similar a C, además de las descripciones abstractas, permite representaciones en bajo nivel; es decir, puede describir sistemas digitales en base a compuertas e incluso en base a transistores. Verilog permite la descripción estructural del diseño en base a componentes básicos y descripciones abstractas que se enfocan en el funcionamiento del sistema. El funcionamiento puede describirse mediante expresiones lógicas y también, empleando procedimientos. Las descripciones en niveles estructurales permiten optimizar los circuitos lógicos para maximizar la velocidad y minimizar los recursos a nivel cómputo de los FPGA. A continuación se expondrán las formas de descripción aplicadas en el sistema de adquisición de imagen presentado en esta tesis.

3.2 Módulo principal programado en Verilog HDL

Para representar un sistema se requiere definir un módulo principal, en el cual, los argumentos son los nombres de las variables de entrada, salida y señales bidireccionales del sistema. Este módulo implementado en Verilog para el sistema de adquisición de imagen lo definimos como

acquisition_system

el cual tiene como argumentos las terminales de entradas y salidas de los dispositivos que conforma el sistema. En el Anexo A, se reportan los programas, es decir, los códigos escritos en Verilog HDL, del módulo *acquisition_system*, con su respectiva declaración de entradas y salidas, editado en Quartus II y en función a los diagramas esquemáticos presentados en el capítulo 2 de la tesis.

Como se mencionó en el capítulo 2, usamos una cámara CCD implementada en un microdurómetro, con un video de salida estándar NTSC. Los datos de la imagen son leídos desde el conector de entrada de video de la tarjeta DE2 y que

se va procesando a través de un conjunto de módulos de hardware, tal como fue presentado en la Figura 2.9 del capítulo 2.

La Figura 3.1 [16] muestra un diagrama a bloques de los módulos programados en Verilog que son los componentes del módulo principal, para convertir los datos provenientes de la cámara CCD (formato de video ITU656) al formato de video YCbCr, también conocido como formato YUV 4:2:2.

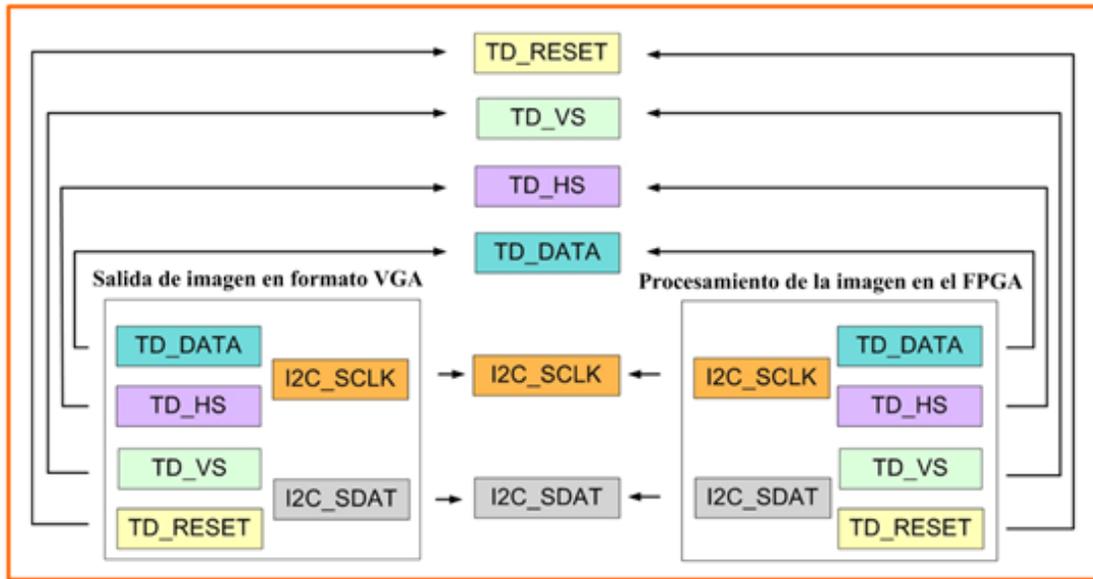


Figura 3.1: Módulos para convertir datos de imagen al formato YCbCr.

3.3 Módulo ITU-R-656 a 4:2:2

El formato de datos de la señal de video ITU-R-656 fue implementado formalmente como una recomendación del sector de la Radiocomunicación de la Unión Internacional de Telecomunicaciones (ITU-R). Este describe un protocolo para la interfaz y envío de datos no comprimidos entre señales de televisión en formato PAL (625 líneas), SECAM (625 líneas) o NTSC (525 líneas) y equipo de televisión digital.

El protocolo define ambas interfaces, el envío de bits serial y paralelo, así como también esquemas de multiplexado, sincronía y generador de pulsos usados en transmisión de señal de video. El módulo programado en Verilog HDL para el controlador BT656 del FPGA utilizado en el sistema de adquisición de imagen, implementa una interfaz paralela de 8 bits, para recibir datos de video en el formato BT656. El

módulo ITU-R-656 define el esquema para obtener de una señal de video analógica, una señal digital adherida al formato ITU-R-601.

Los campos que componen los datos del protocolo BT656 son: EAV (*fin de video activo*, EAV por sus siglas en inglés), *blanking* (sustitución de datos por espacios en blanco) y SAV (*inicio de video activo*, SAV por sus siglas en inglés).

- EAV: marca el fin de los datos de video activo en el flujo de datos presentes y por lo tanto, también el inicio del flujo de datos subsecuente.
- SAV: anuncia el inicio de los datos de videos activos en el flujo actual.

Ambos códigos EAV y SAV constan de una secuencia de 4 bytes. Los primeros tres bytes en la secuencia consisten de un código de advertencia: FFh-00h-00h. El cuarto byte, normalmente etiquetado como *XY*, contiene información sobre los campos que se están transmitiendo (campo 1 o campo 2 en una señal de video), el estado del campo *blanking* (vertical y horizontal). La asignación de bit para este cuarto byte del código se muestra en la Figura 3.2, con la función de cada bit descrito en la Tabla 3.1.

MSB				LSB			
I	F	V	H	P3	P2	P1	P0

Figura 3.2: Bit de asignación para el cuarto byte de códigos EAV y SAV.

Los bits de protección permiten la detección y corrección de uno a dos bits. El estado de los bits P3, P2, P1 y P0, depende del estado de los bits F, V y H, tal como se muestra en la Tabla 3.2. Cuando se recibe una gran cantidad de datos de video, el controlador BT656 ignora los bits de protección (0, 1, 2, 3) de cada cuarto byte de los códigos EAV y SAV.

De acuerdo con el protocolo ITU-R-BT-601, los píxeles individuales son codificados usando los componentes de imagen YCbCr, más precisamente, el muestreo YCbCr 4:2:2. Para un sistema estándar de video de 525 líneas, cada línea de video activa consiste de 720 muestras activas de Y (luminancia o brillo) y 360 muestras de cada Cb (crominancia o color azul) y Cr (crominancia o color rojo). Por lo tanto, el número total de muestras en una región de datos de video activo es de 1440.

Después de que el código SAV ha transmitido el flujo de video, los datos de video activo son enviados en la siguiente secuencia estándar multiplexada, tal como

Tabla 3.1: Valor de asignación de cada bit de los códigos EAV y SAV.

Bit	Símbolo	Valor de función
7	1	siempre en 1
6	F	bit de campo 0-campo: 1, 1-campo: 2
5	V	bit de estado <i>blanking</i> vertical (este bit permanece en estado alto mediante un intervalo de <i>blanking</i> vertical, en otro caso, permanece en estado bajo).
4	H	bit de estado <i>blanking</i> vertical 0: El byte es parte del código SAV (los datos presentes entran en una región de datos de video activos para una línea de flujo actual de datos). 1: El byte es parte del código SAV (los datos presentes inician una nueva línea).
3	P3	bit de protección 3
2	P2	bit de protección 2
1	P1	bit de protección 1
0	P0	bit de protección 0

Tabla 3.2: Estado de bits de protección en dependencia de F, V y H.

F	V	H	P3	P2	P1	P0
0	0	0	0	0	0	0
0	0	1	1	1	0	1
0	1	0	1	0	1	1
0	1	1	0	1	1	0
1	0	0	0	1	1	1
1	0	1	1	0	1	0
1	1	0	1	1	0	0
1	1	1	0	0	0	1

es definido por el protocolo ITU-R BT656:

$$Cb_0 Y_0 Cr_0 Y_1 Cb_1 Y_2 Cr_1 Y_3 \cdots Cb_{359} Y_{718} Cr_{359} Y_{719} .$$

Cada grupo de CbYCr corresponde a un punto único de la imagen transmitida, esto es, a un solo pixel. Las tres muestras en dichos grupos se denominan coubicadas.

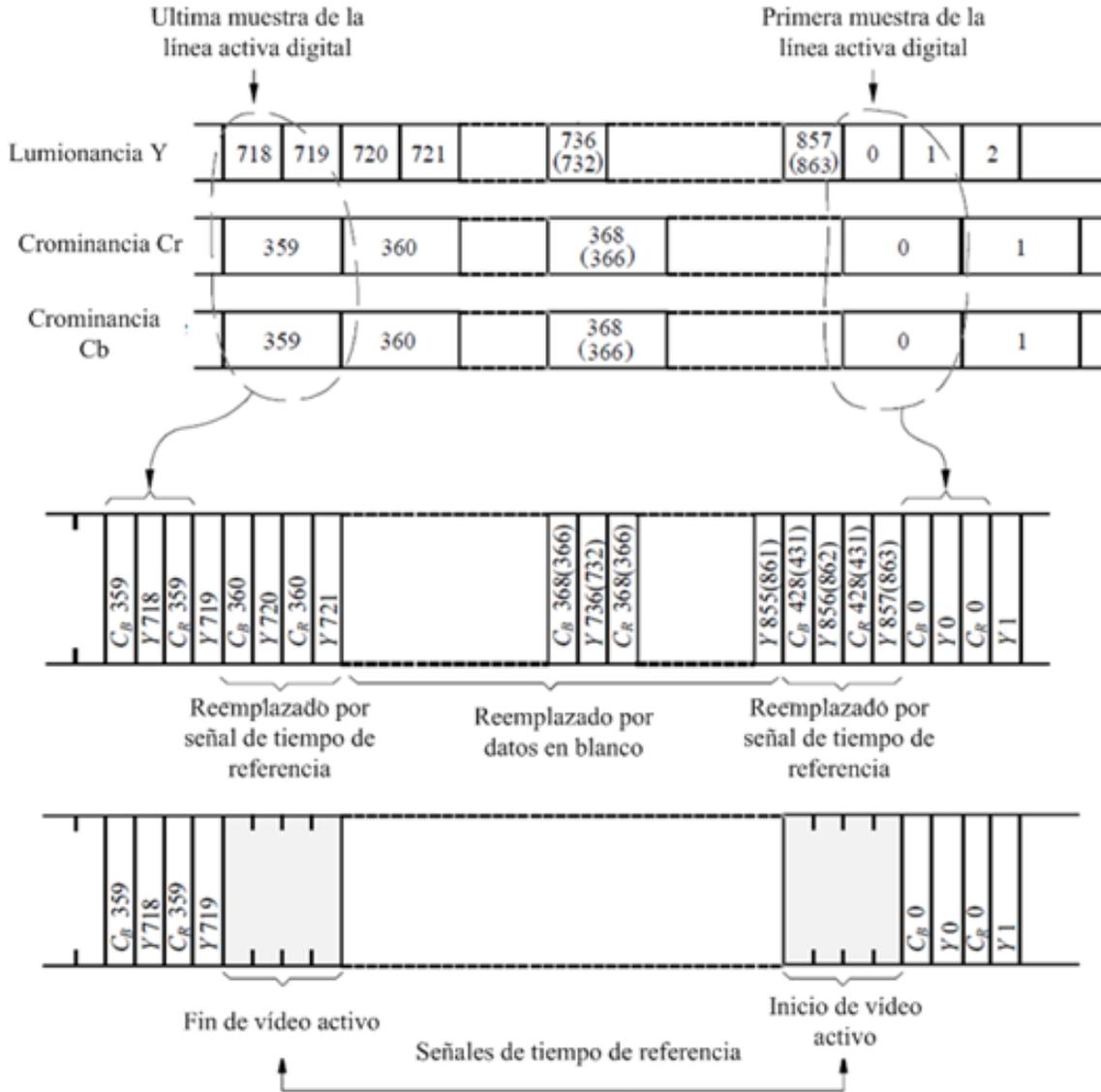


Figura 3.3: Composición del flujo de datos de la interfaz ITU-R BT656.

Durante la transmisión de la señal de video, la porción de la secuencia entre los segmentos de datos activo es conocida como el intervalo de borrado horizontal. En sentido estricto, la totalidad de esta región pertenece al intervalo de borrado, sin embargo, este intervalo incluye los códigos EAV y SAV, mostrado en la Figura 3.3.

El Anexo A presenta el módulo *ITU_656_Decoder* programado en Verilog HDL, donde la señal *iTD_DATA* es la entrada de video del decodificador. La salida de video del decodificador para barrido horizontal y vertical están dadas por las señales *TD_HS* y *TD_VS*, respectivamente, mientras que el argumento *oTV_Cont* es el contador de datos para un campo. La salida en el protocolo 4:2:2 está proporcionada por el argumento *oYCbCr*. El argumento *oDVAL* es la señal que valida si los datos de la señal de video por cada campo son correctos. El argumento *iSwap_CbCr* sirve para controlar el multiplexado del flujo de video en el protocolo ITU-R-BT-601. El argumento *iSkip* es una entrada de control que evalúa si no hay píxeles omitidos en la cadena 4:2:2. Finalmente, los argumentos *iRST_N* y *iCLK_27* son las entradas de control de reinicio y sincronización del decodificador ADV7181.

3.4 Bloque SDRAM frame buffer

El módulo *Sdram_Control_4Port*, programado en Verilog HDL, fue desarrollado para capturar un frame completo del video. El *buffer* para cada frame es necesario para combinar los campos pares e impares de un video ITU-R-BT656. Este módulo permite controlar todas las operaciones de lectura y escritura entre el FPGA y su memoria SDRAM.

A fin de obtener una salida digital VGA, la operación de procesamiento de imagen será ejecutada en el decodificador de video, ya que el protocolo ITU-R-BT656 maneja un video analógico de campos pares e impares.

Los datos de video YCbCr 4:2:2, son escritos a la SDRAM si la señal de validación de datos es activa. El *buffer* de la SDRAM almacena un frame completo. Las líneas de campos pares e impares son almacenadas de manera separada, tal como se observa en la Figura 3.4. Cada línea contiene 640 muestras. La capacidad de almacenamiento en memoria es de 525x640 direcciones, para almacenar en el *buffer* un frame completo. Cada dirección de memoria incluye 16 bits de datos YCbCr 4:2:2.

Los bloques de control del VGA generan señales *VGA_Y* lo cual indica el número de líneas requeridas en el tiempo de ejecución. De acuerdo a la señal *VGA_Y* la dirección leída de la memoria SDRAM, es multiplexada. La señal *VGA_Y* proporciona la información sobre la línea que deberá ser leída desde la memoria. Las líneas pares e impares son leídas secuencialmente, tal como lo muestra la Figura 3.5.

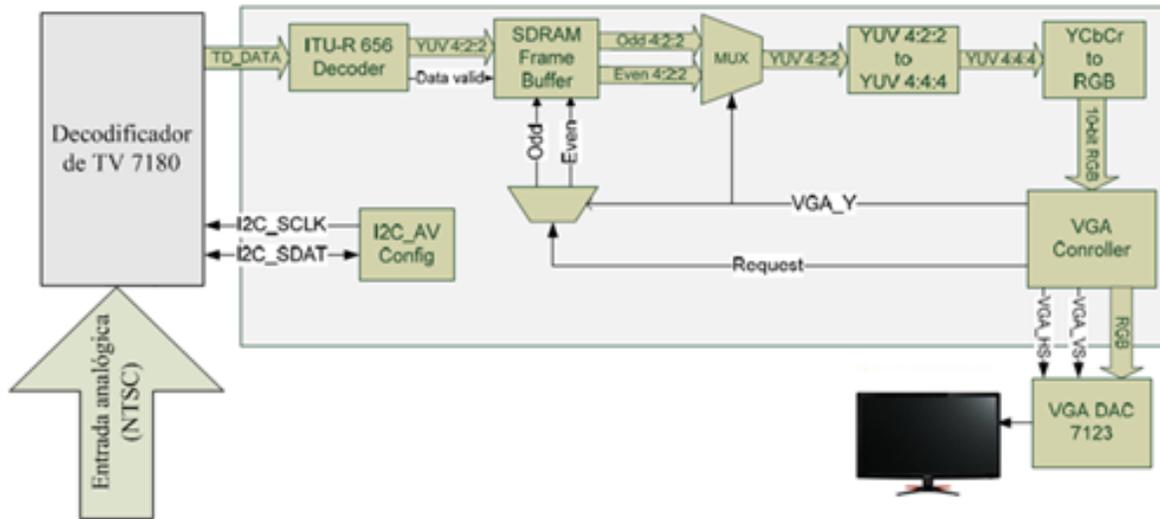


Figura 3.4: Módulos de adquisición de imagen implementados en Verilog HDL.

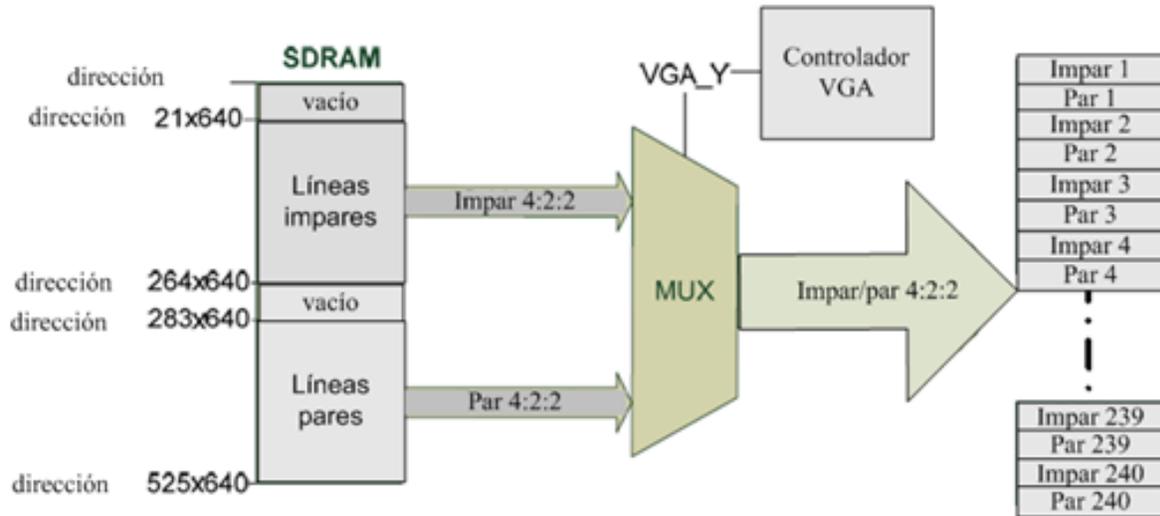


Figura 3.5: Lectura de la memoria SDRAM.

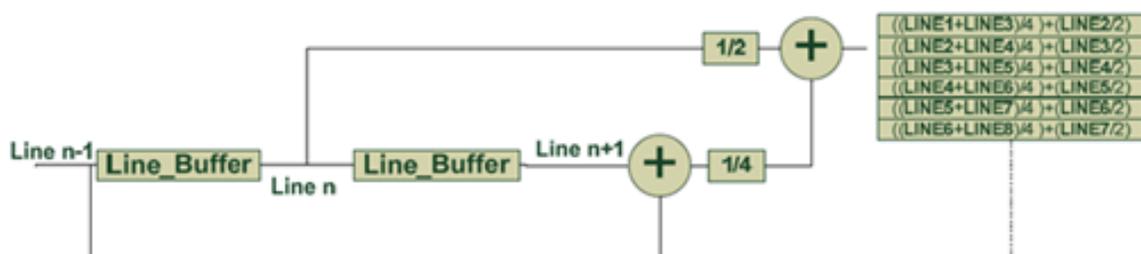


Figura 3.6: Multiplexado de líneas de video entre dos buffers.

A través del multiplexado en la dirección de lectura de la SDRAM, las líneas de video son creadas. En nuestro diseño, hay dos *buffers*, los cuales mantienen los datos de las previas líneas de video. Nuestro algoritmo determina cada línea de video previa y siguiente, durante la operación realizada por el par de *buffers*, tal como lo muestra la Figura 3.6.

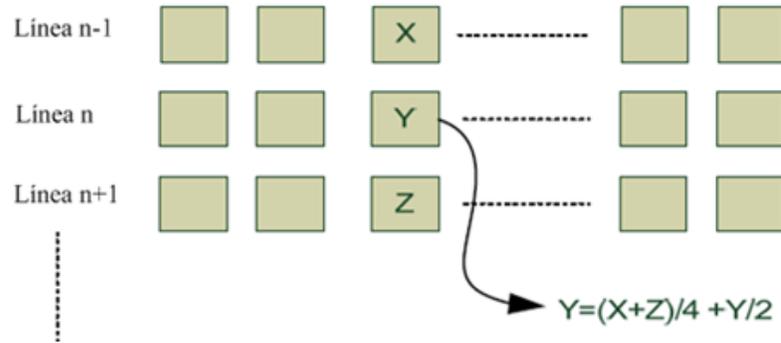


Figura 3.7: Operación de desentrelace para cada píxel.

Mientras se calcula la línea de video en circulación (n), un cuarto de la línea anterior ($n - 1$) y un cuarto de la siguiente línea ($n + 1$) son agregadas a la mitad de la línea en circulación (n). La ecuación

$$línea\ nueva = \frac{1}{4} (línea_{n-1} + línea_{n+1}) + \frac{1}{2} (línea_n)$$

simboliza la operación realizada por nuestro algoritmo. Para cada píxel de cada línea determinada por el algoritmo, se calcula su valor. Así, los píxeles vecinos de una línea vertical se consideran como lo muestra la Figura 3.7.

El Anexo A de la tesis presenta el código escrito en Verilog HDL del módulo *Sdram_Control_4Port*, el cual es utilizado como *buffer* para el almacenamiento de un frame completo de video en formato YUV 4:2:2.

3.5 Transformación de YCbCr 4:2:2 a YCbCr 4:4:4 y conversión en formato RGB

Después de la operación de almacenamiento de un frame completo, los datos de 16 bits de formato YCbCr 4:2:2 son reemplazados por datos de 24 bits de formato YCbCr 4:4:4. La Figura 3.8 muestra las entradas y salidas de los argumentos del

módulo *YUV422_to_444* programado en Verilog HDL. El programa se adjunta en el Anexo A de la tesis.

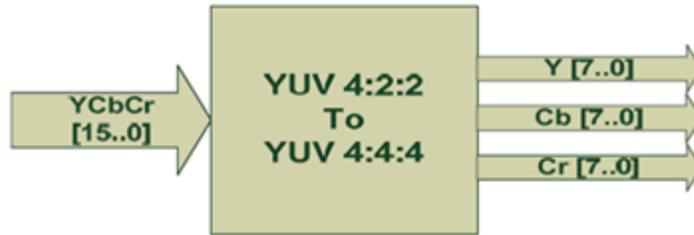


Figura 3.8: Entradas y salidas del módulo YUV422_to_444.

En este módulo, se aplica la técnica de interpolación para los valores de Cb y Cr faltantes en 4:4:4. En el formato YCbCr, para cada muestra horizontal Y, hay una muestra de Cb o Cr, tal como muestra la Figura 3.9. Para valores de Cb y Cr faltantes, los componentes de crominancia son replicados.

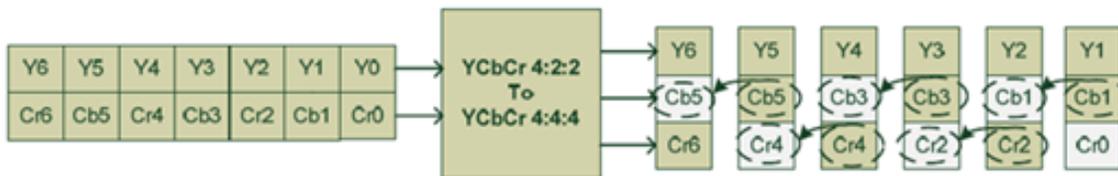


Figura 3.9: Interpolación para valores de crominancia.

El bloque *YCbCr a RGB* transforma los 24 bits del espacio de color YCbCr al espacio de color de 30 bits RGB. El formato YCbCr es una versión compensada del espacio de color YUV, el cual es definido para tener un rango nominal de 16-235. Sin embargo, los datos Cb y Cr son definidos para tener un rango nominal de 16-240. Por eso, los datos RGB deberán estar saturados en el rango de 0-1023 para evitar problemas de desbordamiento y sobreflujo de datos. La Figura 3.10 muestra los argumentos del bloque *YCbCr a RGB* programado en Verilog HDL (para el programa en detalle, vea Anexo A). Nuestro algoritmo consiste de un ciclo indefinido que se sincroniza con la señal del oscilador de 27 Mhz del sistema DE2, y logra evitar la saturación de los datos RGB.

Las señales de sincronización horizontal y vertical son generadas directamente por el FPGA, sin embargo, no pasan a través del convertidor de señal analógica-digital ADV7123 mencionado en el capítulo 2. Esto significa que es necesario pro-



Figura 3.10: Bloque de conversión del espacio de color.

porcionar las señales de sincronización horizontal y vertical del controlador de video como salida, con la finalidad de tener una señal VGA completa.

3.6 Controlador de VGA

Este bloque es usado para generar las señales de sincronía para el circuito integrado ADV7123. Las señales de sincronización del VGA son generadas directamente del bloque del **controlador de VGA**. También las señales *RGB*, *VGA_BLANK* y *VGA_SYNC* que son necesarias para el ADV7123, son generadas a través de este bloque. Así mismo, las señales de entrada *mCursor_Y*, *mCursor_X* y *mOSD_CUR_EN* son utilizadas para ubicar en tiempo real las coordenadas del monitor, así como para interactuar con la interfaz de usuario si se quiere observar las imágenes almacenadas en la SDRAM, o, en tiempo real, el video proporcionado por la salida VGA. La Figura 3.11 muestra los argumentos del controlador de VGA, denominado en Verilog como *VGA_Ctrl*, módulo que se adjunta en el Anexo A de esta tesis. En la figura también pueden observarse las conexiones con el convertidor analógico-digital y el conector VGA en la tarjeta Altera DE2 Cyclone II.

Las señales *RGB* proporcionadas por el bloque *YCbCr_to_RGB*, son transmitidas hacia el *controlador de VGA*, sin cambio alguno. La señal de reloj de 27 Mhz es usada como la señal para obtener la velocidad de transmisión de píxeles transmitidos por segundo. Las señales *VGA_HS* y *VGA_VS* son señales generadas por la señal de sincronía de 27 Mhz. El tiempo de la señal para video de 640x480, usando los 27 MHz de la señal de sincronía, se muestra en la Tabla 3.3 ; estas características son usadas por nuestro algoritmo del controlador de VGA.

La señal *VGA_BLANK* del controlador inicia en estado bajo y permanece así

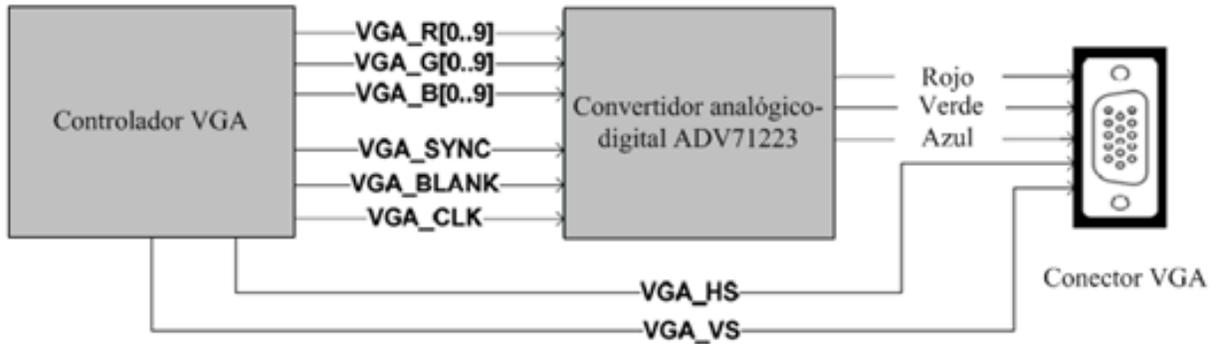


Figura 3.11: Señales de sincronía generadas por el controlador de VGA.

Tabla 3.3: Tiempos de sincronía horizontal y vertical.

Señal horizontal	Velocidad de envío de pixeles (27MHz)	Señal vertical	Número de líneas
H_FRONT	16	V_FRONT	11
H_SYNC	96	V_SYNC	2
H_BACK	48	V_BACK	31
H_ACT	640	V_ACT	480

durante el periodo de sincronía horizontal (H_SYNC) y vertical (V_SYNC). Si la señal VGA_BLANK presenta un cero lógico, las entradas de pixel RGB enviadas desde el controlador de VGA, son ignoradas. La señal VGA_BLANK es baja durante los periodos de tiempo H_BLANK y V_BLANK , las cuales son calculadas como sigue:

$$H_BLANK = H_FRONT + H_SYNC + H_BACK$$

$$V_BLANK = V_FRONT + V_SYNC + V_BACK$$

De acuerdo a nuestro programa del controlador de VGA (código VGA_Ctrl , vea Anexo A), se observa la asignación de los datos de la imagen almacenada en la SDRAM ($mSRAM_VGA_DATA[15:8]$), que son direccionados al registro de dirección de memoria del controlador de VGA. El registro de dirección de memoria del controlador de VGA ($mVGA_ADDR[0]$), es asignado a cada uno de los registros $mVIN_R$, $mVIN_G$ y $mVIN_B$, los cuales se conectarán a las salidas de datos para la VGA.

La información que será enviado a través de la salida VGA hacia un monitor, será el video en tiempo real proporcionado por la cámara CCD del microdurómetro, o la imagen almacenada en la memoria SDRAM. Mediante el módulo principal de todo el sistema, siendo eso el programa $acquisition_system$, a través de operadores

lógicos usados en Verilog, el sistema elige las opciones como sigue:

Si el registro *mOSD_CUR_EN* del módulo *USB_JTAG* recibe un cero lógico a través de la entrada de datos del puerto USB del sistema Altera DE2, un frame completo de la imagen almacenada en la SDRAM del FPGA se enviará a la salida VGA del sistema Altera DE2. Por el contrario, se recibe un 1 lógico, la salida del VGA reproducirá la imagen de video en tiempo real a través de los registros *mRed*, *mGreen* y *mBlue*, declarados en el módulo *YCbCr2RGB*.

3.7 Comunicación vía USB JTAG

Verilog permite descripciones jerárquicas de redes complejas. Por ejemplo, la configuración del FPGA para que establezca comunicación vía el puerto USB JTAG, puede describirse mediante un módulo con asignaciones directas y registros. En nuestro programa, el módulo *USB_JTAG* es un subsistema del programa principal, el cual puede representarse a través del esquema de la Figura 3.12.

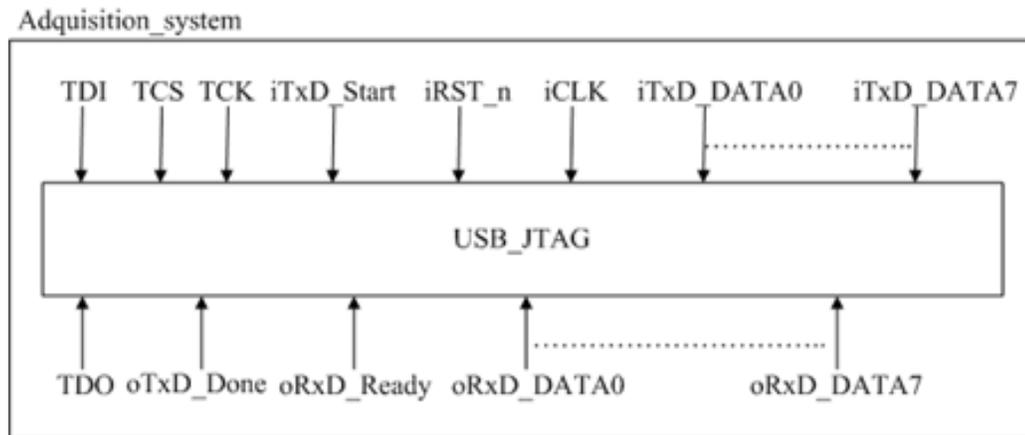


Figura 3.12: Esquema del módulo *USB_JTAG*.

Los argumentos del módulo *USB_JTAG* se definen como entrada, salida, y señales en modo registro, siendo estas últimas representadas como salidas de tipo Flip-Flop, por lo que deben de estar asociados a una entrada de reloj para su sincronización. Por ejemplo, una salida tipo registro se declara en Verilog como sigue:

```
output reg [7:0] oRxD_DATA;
output reg oTxD_Done,oRxD_Ready;
```

Las conexiones o redes entre componentes de un diseño digital generan nodos internos en el sistema. En Verilog, estos nodos internos son declarados como *wire*. Las señales tipificadas con *wire* no tienen capacidad de almacenamiento de información; se consideran declaradas por la salida de un módulo, o a través de una asignación continua. Si las señales de tipo *wire* se dejan desconectadas, toman el valor de *x*, el cual es un valor desconocido para el FPGA. Las declaraciones *wire* son utilizadas en el módulo *USB_JTAG*, el cual puede ser observado de forma completa en el Anexo A. El siguiente ejemplo muestra una declaración tipo *wire* para el módulo *USB_JTAG*.

```
wire [7:0] mRXD_DATA, mTXD_DATA;  
wire mRXD_Ready, mTXD_Done, mTXD_Start;  
wire mTCK;
```

El primer bloque del módulo *USB_JTAG* se encarga de la entrada de datos por el USB JTAG. El proceso de este bloque se ejecuta siempre y cuando se presenten pulsos de reloj en la entrada *iCLK*. Si la entrada *iRST_n* se encuentra en estado bajo, la salida *oRxD_Ready* estará en estado bajo, lo que implica que no hay datos recibidos. Si el *wire mRxD_Ready* tiene valor de un 2 binario (estado activo) y la transmisión está desactivada, la salida del registro *oRxD_Ready* estará en estado alto. Eso genera que la salida del registro *oRxD_DATA* recibirá datos a través del *wire mRxD_DATA*. Por lo contrario, si *mRxD_Ready* no es igual al estado 2 binario, la salida tipo registro *oRxD_Ready* estará en estado bajo, lo cual indicará que no hay datos recibidos por el USB JTAG.

Para el transmisor vía USB JTAG, es exactamente igual que el receptor, el bloque de transmisión, se sincroniza con los mismos pulsos de reloj adquiridos por la entrada *iCLK*. En caso de estar en estado bajo, la entrada *iRST_n* indica que no habrá transmisión de datos, por lo que la salida del registro *oTxD_Done* estará en estado bajo. En caso contrario, si la conexión *wire* presenta un valor 2 binario, el registro *oTxD_Done* estará en estado alto, lo cual indicará que hay transmisión de datos vía USB JTAG; en caso contrario, permanecerá en estado bajo. El código para el transmisor vía USB JTAG, escrito en Verilog, está incluido en el Anexo A.

La siguiente parte del módulo *USB_JTAG*, el submódulo *JTAG_REC*, se refiere a la recepción de datos del registro *oRxD_DATA* hacia la entrada de datos del USB JTAG. El programa *JTAG_REC* se sincroniza con la entrada de reloj *TCK* del FPGA para la comunicación del FPGA vía USB JTAG. Si la entrada

de selección del dispositivo (*TCS*) para el UBS JTAG no está activa, la salida de registro *oRxD_Ready* estará en estado bajo, así como un contador de datos *rcont* permanecerá en estado bajo. Por lo contrario, si la entrada *TCS* está activa, el contador se incrementa y el registro *rDATA* recibe los datos del registro de salida *oRxD_DATA* que serán cargados en tramos de 8 bits a la entrada de datos del USB JTAG.

Finalmente, el subsistema *JTAG_TRANS* realiza la transmisión de datos vía la salida *TDO* del USB JTAG. El programa *JTAG_TRANS* se sincroniza con la entrada *TCK*, o a través de que la entrada *TCS* se encuentre en esta alto. Si esta entrada de selección del dispositivo está en estado bajo, entonces la salida del registro *oTxD_Done*, el contador de envío de datos (*rCont*) y la salida de datos (*TDO*) permanecerán en estado bajo. Por lo contrario, si la entrada de datos *iTxD_Start* está activa, se incrementa el contador de datos *rCont*, y los datos a transmitir son enviados al registro *TDO*, el cual será la terminal de salida del USB JTAG. Si *iTxD_Start* permanece en estado bajo, *rCont* y *TDO* tendrán valor cero. Cuando la cantidad de datos enviados *rCont* alcanza el valor 8, entonces la salida del registro *oTxD_Done* se activará; en caso contrario, permanecerá en estado bajo.

Los argumentos de todos estos subsistemas, son llamados a través del módulo principal *acquisition_system* y por el módulo *USB_JTAG*, tal como puede ser observado en los listados de programa completos en el Anexo A. En particular, las entradas y salidas definidas como argumentos del módulo principal, se conectan con las entradas y salidas correspondientes a los argumentos declarados en el módulo *USB_JTAG*. Los pulsos de reloj para sincronizar la señal de entrada *TCK*, son proporcionados por el oscilador de 50 Mhz del sistema Altera DE2 Cyclone II.

Capítulo 4

Interfaz gráfica de usuario para la comunicación con el FPGA, y para la supresión de ruido de señal

4.1 Introducción

En este capítulo se presenta una interfaz gráfica que permite al usuario comunicarse con el FPGA Altera DE2-115, y un sistema de filtrado embebido en el FPGA, implementado con características originales. Además de utilizar el FPGA como sistema de adquisición de imagen [15, 16], este sistema incluye dicha interfaz y opciones del procesamiento de señales, consistiendo de los siguientes componentes:

- el hardware necesario,
- el firmware de bajo nivel que implementa operaciones de convolución, y
- un GUI (*interfaz gráfica de usuario*, GUI por sus siglas en inglés), el cual configura filtros digitales de acuerdo al requerimiento del usuario, descargándolos al FPGA donde son ejecutados en tiempo real, para eliminar ruido eléctrico de la señal de video analógica proveniente de la cámara CCD del microdurometro.

Componentes de nuestro sistema de procesamiento digital de señales embebido en un FPGA, no solamente fueron usados para la adquisición y el preprocesamiento de imágenes de indentación Vickers, sino también fueron utilizados para aplicaciones biomédicas [35, 14] donde fue necesario eliminar ruido magnético y eléctrico de señales.

En el procesamiento de señales, el objetivo de la aplicación de un filtro es remover partes no deseadas de la señal, tales como ruido aleatorio, o extraer información útil de la señal, por ejemplo, componentes dentro de cierto rango de frecuencias [7]. Los recursos de hardware embebidos en un FPGA son considerablemente superiores a los recursos de un DSP (*procesador digital de señales*). Además, los recursos utilizados en un FPGA, en términos de multiplicadores y registros, pueden ser esencialmente reducidos por el tiempo compartido [38], es decir, la misma estructura implementada es usada muchas veces entre dos muestras de entradas sucesivas. A través de esta técnica, se puede usar los FPGA para la implementación de filtros IIR, únicos o multicanales, implementados en conexión, en cascada de filtros de segundo orden. Por ejemplo, los filtros propuestos en [38, 18], son filtros FIR de alto orden, los cuales son creados a partir de filtros de bajo orden y de tal manera que los rizados de la banda pasante son reducidos y la atenuación en la banda rechazada es incrementada. En este contexto es útil que un filtro FIR de bajo orden puede ser usado como una función simple. No obstante, esta técnica no aplica a filtros que presentan retardos no constantes tales como filtros FIR en general, o filtros IIR.

El diseño de filtros por un usuario con falta de experiencia, puede fallar para cumplir con las especificaciones del filtro debido a la cuantificación de sus coeficientes, su rango dinámico, la cuantificación del ruido, y por efectos no lineales como por ejemplo, sobreflujo y efectos de banda muerta. Algunas herramientas para el diseño de filtros son incluidos en sistemas como Matlab o Labview, y en bibliotecas como OpenCV.

Para entender cuáles son las necesidades que nuestro sistema de filtrado debe satisfacer, la siguiente sección revisa las propiedades del espectro de una imagen proveniente de una cámara CCD.

4.2 Espectro de frecuencias de la señal de video analógica reproducida por una cámara CCD

La energía almacenada en la cámara CCD del microdurómetro es amplificada en una cantidad de voltaje medible. El máximo voltaje de una señal de video es de 1 V_{p-p} (voltaje pico a pico), donde 0.5 V es asignado para los pulsos de sincronía y 0.05 V para el ajuste de iluminación, dejando 0.7 V para la información de video

[29]. Un ruido en alta frecuencia se presenta en la diferencia de potencial entre la señal de video con respecto a la tierra del circuito decodificador de video ADV7181 del sistema Altera DE2. En un sistema electrónico, estas señales son generadas porque interfieren capacitivamente con otras señales de bajo ruido, además, debido al ambiente electromagnético exterior, lo cual es conocido como interferencia electromagnética [13].

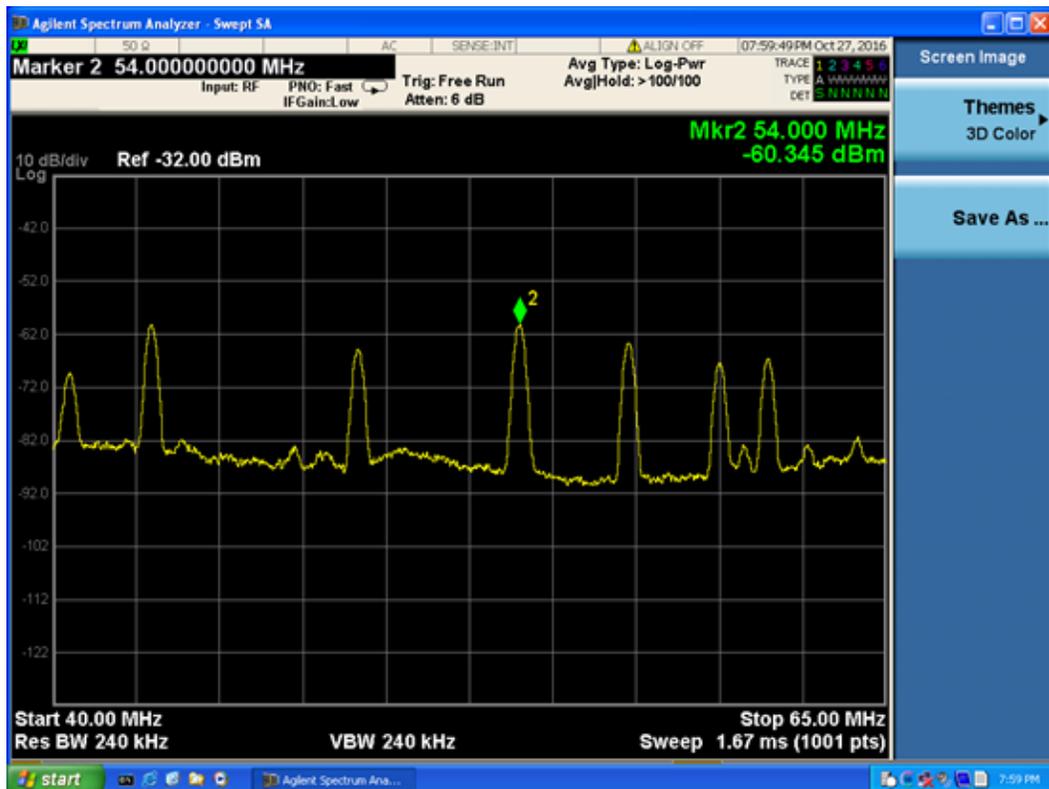


Figura 4.1: Espectro de frecuencia del voltaje de salida del amplificador de una cámara CCD.

Para encontrar las frecuencias en las cuales, el ruido electrónico a la salida del amplificador de la CCD tiene los valores máximos, fue necesario medir el espectro de frecuencia. Para esto, el analizador de espectro N9020A (Agilent Technologies) fue usado para evaluar el comportamiento del voltaje de salida del amplificador de la CCD en el dominio de la frecuencia, considerando los armónicos más grandes, como lo muestra la Figura 4.1.

Una vez representado el espectro de frecuencia, usando la serie de Fourier, a partir de un filtro de primer orden de pasa bajas, se obtuvo el factor de atenuación

de los armónicos más altos del espectro. Este factor de atenuación se configuró en un instrumento virtual programado en Borland Delphi 7.net, el cual tiene comunicación con el sistema FPGA Altera DE2115. Los resultados de este trabajo fueron reportados en [14] y [16].

4.3 Filtros lineales

El proceso lineal del filtrado en el dominio del tiempo es representado por la integral de convolución:

$$y(t) = \int_{-\infty}^{\infty} h(\tau)x(t - \tau)d\tau , \quad (4.1)$$

donde $y(t)$ es la señal filtrada de salida, $x(t)$ es la señal de entrada, τ es el operador de desplazamiento en el tiempo y $h(\tau)$ es la respuesta del filtro al impulso.

Un filtro digital es caracterizado por su función de transferencia, o, lo que es su equivalente a su ecuación de diferencia. El análisis matemático de la función de transferencia puede describir cómo responderá el filtro a cualquier señal de entrada. La función de transferencia de un filtro digital, lineal e invariante en el tiempo, se puede representar en el dominio z ; si es causal, entonces tiene la siguiente forma:

$$H(z) = \frac{a[0] + a[1]z^{-1} + \dots + a[M]z^{-M}}{1 + b[1]z^{-1} + \dots + b[N]z^{-N}} \quad (4.2)$$

donde el máximo de los números M y N es llamado el *orden del filtro*.

En sistemas de tiempo discreto, el filtro digital a menudo se implementa mediante la conversión de la función de transferencia a una ecuación de diferencias con coeficientes constantes lineal, vía transformada z . La ecuación de diferencias lineal resultante es

$$y[n] = \sum_{k=1}^M b_k y[n - k] - \sum_{k=0}^N a_k x[n - k] . \quad (4.3)$$

La ecuación 4.2 muestra que la función de transferencia es representable como cociente entre dos polinomios en potencia negativa ascendente de z . Así, los polinomios de orden superior se asocian con términos muy pequeños en el denominador, de donde surge el riesgo de una división mal condicionada. Por esta razón, los filtros IIR son sensibles a representar sus coeficientes en resolución de 16 bits, con respecto a microcontroladores enfocados al procesamiento digital de señales [31, 55]. En gen-

eral, el alto orden de los filtros y el gran riesgo de inestabilidad hacen que los filtros IIR sean diseñados por muchas secciones de filtros de bajo orden, conectados en cascada.

Con filtros de alto rendimiento para procesar señales de video y audio, la fase lineal es deseable, sin embargo, esta propiedad es esencialmente importante en el procesamiento de señales biomédicas, como fue mostrado en los trabajos [35, 14]. La fase lineal significa que cualquier retardo de tiempo por una componente en frecuencia es experimentado por todas las componentes en igual medida; de aquí que la forma de la señal es preservada en el proceso de filtrado. La fase lineal es garantizada si la respuesta al impulso del filtro es simétrica, es decir, obedece a la siguiente relación:

$$h(n) = h(N - n - 1), \quad n = 0, 1, \dots, \frac{N - 1}{2} \quad (\text{para } N \text{ impar}). \quad (4.4)$$

Filtros IIR pueden ser realizados de diversas formas, dando distintos resultados con respecto a operación, diseño analógico y eficiencia. Entre los muchos métodos para implementar filtros digitales IIR mediante hardware, podemos mencionar las técnicas de “entrada serial - salida serial” y de “entrada paralela - salida serial”.

4.4 Filtro digital IIR implementado en el FPGA

La implementación de un filtro digital en lógica configurable implica la explicación del algoritmo en un lenguaje con descripción de hardware, tal como lo es el lenguaje **VHDL** (*lenguaje de descripción de hardware de muy alta velocidad*, VHDL por sus siglas en inglés). Dicha descripción de hardware tiene muchas ventajas, tales como la modularidad y la interoperabilidad entre sus módulos. Además, eso permite usar el mismo código fuente para implementar diferentes tipos de filtros, como fue realizado en el trabajo [14].

La Figura 4.2 presenta un diagrama a bloques del filtro digital IIR que fue implementado en el FPGA Altera DE2 Cyclone II. En particular, puede observarse la secuencia e interacción de las señales en todos los bloques que integran al filtro.

El diagrama de la Figura 4.3 muestra el diseño del filtro IIR, donde la señal *YK0* presenta la retroalimentación del sistema. Los bloques de retardo están conformados por LRs (*registros de carga*, LR por sus siglas en inglés), los cuales almacenan el

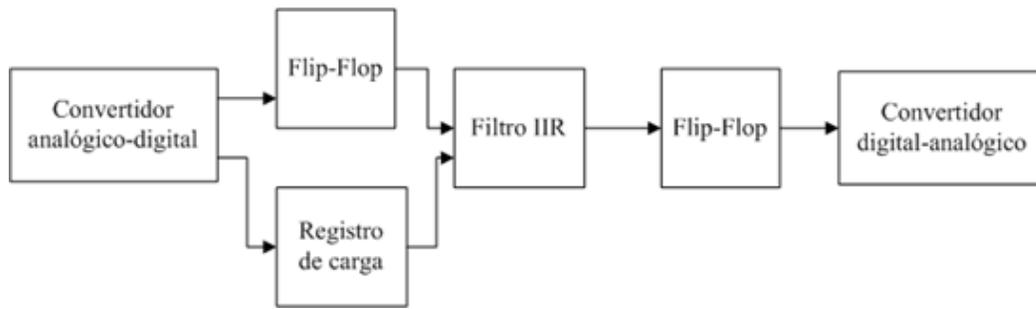


Figura 4.2: Bloques implementados en el FPGA para el diseño del filtro.

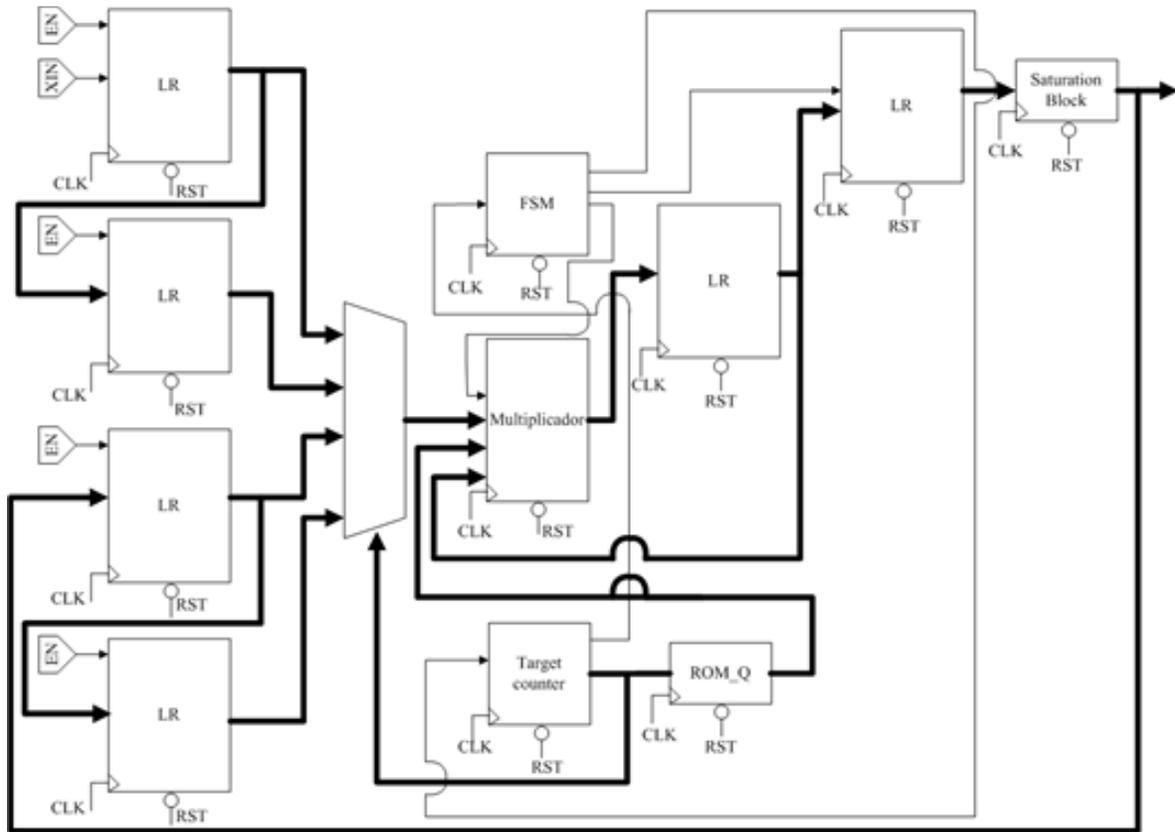


Figura 4.3: Diagrama del Filtro IIR implementado en el FPGA.

valor de la señal de entrada (XIN), cada vez que se active la entrada (EN). De esta manera, esta señal se convierte en el tiempo de muestreo. La cantidad de bloques LRs depende del número de valores dentro de la memoria ROM cuyo bloque es ROM_Q , el cual contiene los coeficientes calculados para el filtro.

El bus del bloque *target counter* cuenta los datos recibidos. Esto implica que la cuenta inicia de cero y el primer valor almacenado es multiplicado con el primer valor almacenado en la memoria ROM hasta llegar al último valor. Todas las señales de entrada y salida son almacenadas.

La **FSM** (*máquina de estado finito*, FSM por sus siglas en inglés) habilita la señal EN obteniendo 10 pulsos de reloj. Por eso, el multiplicador producirá el resultado que será almacenado en un LR, cuya señal de salida será retornada a la entrada C del multiplicador. Entonces la FSM emite un pulso hacia el *target counter*, para incrementar la cuenta de los datos recibidos y, al mismo tiempo, cambiar la señal de salida, para obtener la multiplicación del siguiente par de señales. Finalmente, el *target counter* almacena los coeficientes en otro LR, hasta que un valor previo al último sea calculado. El valor final no puede exceder los 50 bits, ya que pasa a través de un bloque de saturación (*saturation block*) el cual verifica que la señal no alcance los valores límites. En caso de ocurrir, trunca el valor de señal de salida, aplicando técnicas de truncamiento y de selección de bits significativos, como fue propuesto por ejemplo en [26, 57].

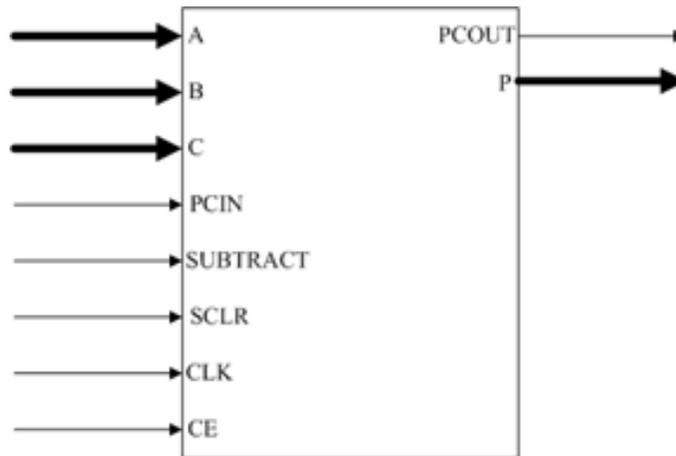


Figura 4.4: Símbolo del bloque multiplicador.

Los nombres de las señales del bloque multiplicador son mostrados en la Figura 4.4. El multiplicador toma dos estados diferentes: uno desde las entradas A y B

hacia la salida P , y el otro desde la entrada $C/PCIN$ hacia la salida $PCOUT$, vea la Figura 4.5.

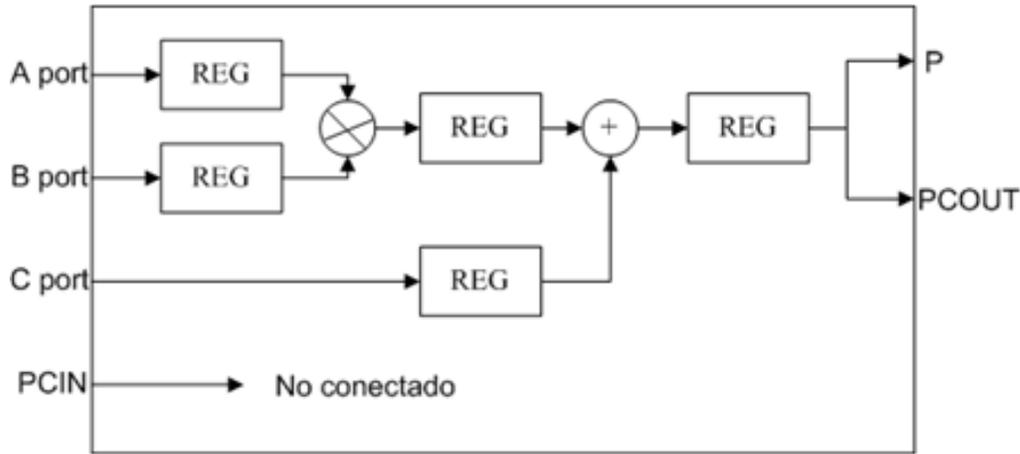


Figura 4.5: Diagrama a bloque interno de un multiplicador.

A la salida del multiplicador no hay cantidades truncadas o redondeadas; por lo que el resultado es bastante preciso. La entrada C es agregada al producto LSB a LSB. Las posiciones MSB y LSB se pueden escoger para extraer el dato de salida deseado. Las terminales del bloque del multiplicador son descritas en la Tabla 4.1.

4.5 Interfaz de usuario

La interfaz gráfica de usuario, la cual funciona en una computadora personal común, a partir del sistema operativo “Windows XP” hasta el sistema “Windows 10 pro”, fue desarrollada en “Borland Delphi 7.net”, el cual es un entorno de programación orientado a objetos. Esta interfaz, además de interactuar con el FPGA para la adquisición de imágenes, es una herramienta para configurar varios tipos de filtros digitales para eliminar ruido eléctrico que se presentan por diversas causas, por ejemplo, una instalación eléctrica no muy adecuada, o por interferencias electromagnéticas. Lo último fue reportado, por ejemplo, en el trabajo [14].

El sistema Delphi permite al usuario escribir el código en un lenguaje de alto nivel basado en Pascal e incorporar instrucciones de nivel ensamblador. Además, Delphi proporciona al programador completo acceso a rutinas API (*interfaz de aplicaciones programables*, API por sus siglas en inglés), para aplicaciones de Windows y de Linux, siendo éstas rutinas eficientes para la operación de los sistemas. Por ejemplo,

Tabla 4.1: Descripción de terminales del multiplicador.

Nombre	Descripción
A[24:0]	señal de entrada A (operando 1 del multiplicador)
B[26:0]	señal de entrada B (operando 2 del multiplicador)
C[55:0]	señal de entrada C (operando 1 de suma y resta)
PCIN	entrada de conexión cascada
SUBTRACT	control de operaciones suma y resta
CE	habilitar pulsos de reloj (en estado alto)
CLK	pulsos de reloj (flanco ascendente)
SCLR	limpia de datos síncronos (en estado alto)
PCOUT[47:0]	salida en cascada
P[54:0]	señal de salida

el software visual de alto nivel usa el llamado a ciertas rutinas API para operar el puerto serial USB de la computadora personal, a través del cual se comunica con el sistema FPGA. Este software tiene características ventajosas como integridad, compatibilidad, portabilidad y escalabilidad.

Estas mismas propiedades, sin embargo, también han permitido filtrar pequeñas interferencias electromagnéticas, como fue reportado por ejemplo en aplicaciones biomédicas en los trabajos [35, 14]. Los filtros digitales para estas aplicaciones nacen del diseño de filtros analógicos para el acondicionamiento de señal de sensores que emiten a la salida muy pequeñas señales del orden de decenas de mili volts. El software desarrollado en Delphi con aplicaciones al procesamiento de señales que fue elaborado y usado en este proyecto doctoral, ejecuta las siguientes funciones principales:

- a) convierte la representación de punto flotante de los coeficientes del filtro a la aritmética fraccional de 24 bits.
- b) descarga el firmware de bajo nivel al hardware.
- c) se comunica con el firmware embebido en el FPGA para cargar los coeficientes del filtro.
- d) permite al usuario alterar la ganancia del sistema y la relación de muestreo durante el tiempo de ejecución.

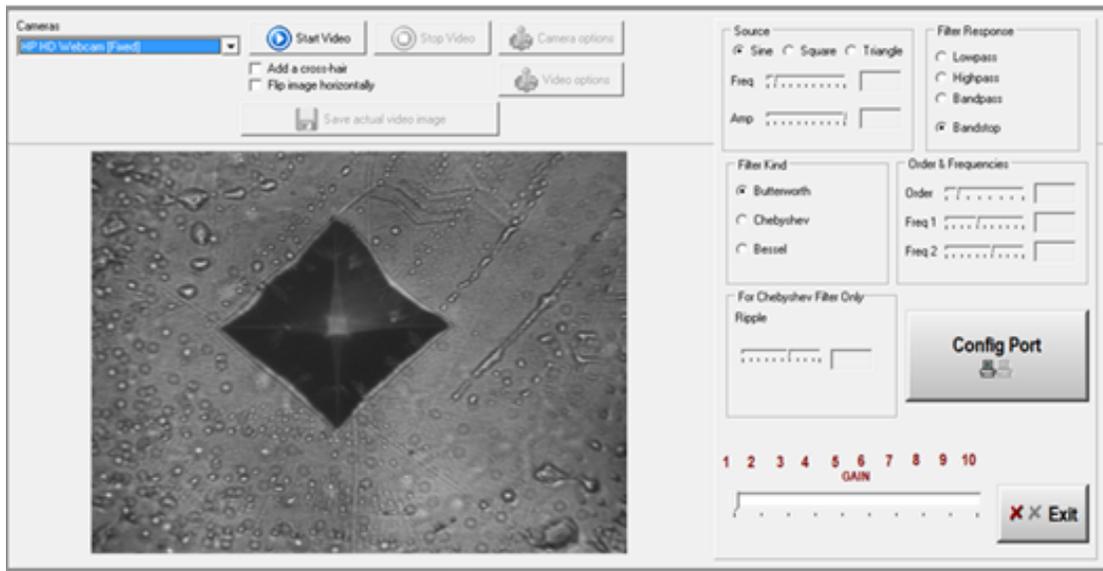


Figura 4.6: Imagen de la interfaz gráfica de usuario, desarrollada en Delphi.

La Figura 4.6 muestra cómo este software aparece al usuario en el monitor de

la computadora. La interfaz incluye una sección para el diseño del filtro, y una sección de control de hardware para cargar un filtro y comunicarse con el hardware del filtro embebido en el FPGA. Para la configuración del filtro de acuerdo a los requerimientos del usuario, se ajusta los parámetros del filtro, considerando los filtros Butterworth, Bessel y Chevyshev. Estas topologías de filtros incluyen las siguientes características: frecuencia de corte mínima de 0 a 40 Mhz y frecuencia de corte máxima de 0 a 70 Mhz, filtros de hasta octavo orden, así como su tipo de respuesta pasa-bajas, pasa-altas, pasa-banda y rechazo-de-banda.

Capítulo 5

Procesamiento y análisis de imágenes de indentaciones Vickers

5.1 Introducción

En la automatización del proceso de medición de dureza a través de indentación Vickers, se han desarrollado y aplicado sistemas de visión por computadora, los cuales asisten al usuario en analizar las imágenes, en particular, en detectar los vértices de la indentación, para que las diagonales de ésta pueden ser medidas.

Como se mencionó en la Introducción de la tesis, pruebas de dureza son usadas para examinación y control de calidad de materiales, con propósitos tanto de investigación como en proceso de fabricación, por ejemplo, para metales, cerámica y polímeros, donde es de interés determinar su resistencia al desgaste y su ductilidad. Las máquinas para pruebas de dureza, llamadas *durómetros*, presionan la superficie del material con una pieza dura, llamada *indentador*, con cierta carga (fuerza) y periodo de tiempo conocidos o medibles. Después de retirar el indentador, la huella de la indentación marcada en la superficie del material es observada a través de un microscopio. Cuando esta escena además es grabada en una fotografía, la huella visible en la imagen es llamada la *indentación*, y es considerada la *región de interés*, para el análisis de la imagen.

La prueba de dureza Vickers reportada por Smith y Sandland [62] consiste en la aplicación de un indentador de diamante que presiona, y parcialmente penetra, la superficie de la muestra del material a prueba. La forma de rombo o cuadrado de

la huella de la indentación es observada, a través del microscopio, por un operador humano quien manualmente mide la longitud de las diagonales con la ayuda de una imagen adquirida por una cámara conectada al microscopio. Usando la longitud promedio d de las dos diagonales, el **número de dureza Vickers HV** es calculado como

$$HV = 1854.4 \cdot F/d^2 \quad (5.1)$$

con la carga F del indentador dada en gf (gramos fuerza) [62, 73].

Mediante la configuración del sistema óptico y el control de la unidad de iluminación, el brillo y el contraste de la imagen de la indentación son determinados. El usuario selecciona el factor de ampliación del microscopio. En la práctica, antes de realizar una prueba de dureza, la superficie del material es preparada para asegurar una buena calidad de la imagen microscópica. Sin embargo, las propiedades propias del material como su color y textura natural, pero también defectos en la superficie causados por el tratamiento mecánico o químico de la muestra a indentar, causan variaciones de diferente tipo, tanto en el fondo de la imagen como en la apariencia de la indentación. También, cambios de las condiciones de iluminación, aún siendo controladas con cuidado en la mayoría de aplicaciones reales, pueden causar variación importante en la apariencia de la imagen. Todo eso puede propiciar una notable incertidumbre en la localización de la indentación en la imagen, y en la determinación precisa de sus vértices. Además, el tamaño y la orientación de la indentación en la imagen, dependen en gran medida del ajuste experimental y de la precisión del equipo de prueba.

El resultado de medición no sólo depende del ajuste experimental, sino también requiere de un experto operador humano para determinar con exactitud los vértices de la indentación, puesto que, de manera tradicional, son encontrados manualmente por una persona quien los señala en la imagen, por ejemplo con ayuda del ratón de la computadora. Se ha mostrado en varios estudios que este proceso manual es fuertemente subjetivo, y puede variar en dependencia de las condiciones de trabajo y de factores personales como el cansancio.

Eso ha motivado el desarrollo de métodos de medición de dureza Vickers basados en el procesamiento y análisis automático o semi-automático, de las imágenes de indentación [10, 22, 23, 24, 33, 41, 44, 45, 47, 63, 73]. Para algunos equipos de medición de dureza, por ejemplo el nuestro, el primer problema no sencillo a resolver consiste en la adquisición de las imágenes, como fue explicado en los capítulos

anteriores y en [15, 16].

La forma de una indentación Vickers es conocida: aproximadamente es un rombo. Para la mayoría de aplicaciones, el ajuste experimental puede asegurar que cada imagen contiene exactamente una indentación. Sin embargo, el tamaño de la indentación, así como su exacta localización en la imagen, y su orientación (es decir, su posible rotación), son desconocidos. Sistemas computacionales tienen que detectar y localizar la indentación en la imagen, y luego estimar sus características, en particular, las longitudes de sus dos diagonales, para lo cual, comunmente, sus vértices son determinados.

Métodos computacionales propuestos en la literatura para el tratamiento de imágenes de indentación Vickers, después de algún preprocesamiento, realizan la segmentación de imagen para detectar e identificar la región de interés de la indentación, y finalmente determinan los cuatro puntos vértices de la indentación. La gran mayoría de estos métodos asumen condiciones demasiado restrictivas, en consecuencia, resultan ser no útiles para ciertos grupos de imágenes [24, 44].

Imágenes reales de indentación presentan los siguientes problemas:

- La forma de la indentación es un rombo, pero solo aproximadamente.
- Los lados de la indentación son segmentos de líneas rectas, pero solo aproximadamente.
- Valores altos del gradiente en puntos muy cercanos a la frontera de la indentación, no pueden ser garantizados. Eso puede motivar el uso de detectores de cantos basados en plantillas pequeñas, los cuales resultan muy sensibles a ruido, causando la detección de cantos que no son parte de la frontera del objeto de interés.
- Las imágenes no siempre presentan alto contraste, o, por lo menos, suficiente contraste para poder distinguir fácilmente la región de interés. Aparecen imágenes de indentación Vickers con muy bajo contraste, o con mucho brillo en el centro de la indentación, o en otras zonas de la imagen.
- La región de la indentación no necesariamente presenta valores de gris o una textura uniformes, dicha uniformidad tampoco es garantizada para el fondo de la imagen.

5.2 Estado del arte

5.2.1 Métodos propuestos en la literatura para procesar y analizar imágenes de indentación Vickers

Entre las técnicas reportadas para segmentar imágenes de indentación Vickers, destacan la binarización por umbralización [22, 41, 47, 63], filtrado morfológico para remover artefactos que no son de interés [41, 47], proyección a los ejes [63, 73], detección de puntos de canto [10, 33], y el amapeamiento con plantillas de cantos [24, 23]. El conjunto de puntos de canto es sometido a la transformada de Hough, para encontrar las líneas rectas que forman la frontera de la indentación [33, 73]. Este último método ha sido refinado por la adaptación de líneas rectas a los lados de la indentación, mediante la aproximación por mínimos cuadrados [33], con el objetivo de modelar y construir los lados de de la indentación, cuyas intersecciones proporcionan los cuatro vértices buscados.

Algoritmos basados en la coincidencia de plantillas [23, 24, 44] son robustos a ruido, ya que reconstruyen valores, suprimiendo ruido en grandes regiones de la imagen. En particular, estos métodos usan plantillas cuadradas de diferentes medidas y rotación. Sin embargo, su utilidad es restringida por el hecho que indentaciones en la práctica real, no son perfectamente cuadradas o rómbicas. Los mismos autores, en varios trabajos, presentaron mejoras a sus algoritmos publicados anteriormente, culminando en un algoritmo basado en la combinación de tres plantillas [24]. Este último algoritmo logra obtener gran estabilidad bajo variaciones de brillo y para bajo contraste de la imagen, robustez contra defectos en la superficie del material, tales como surcos o puntos de porosidad, además de ser confiable en la detección de indentaciones pequeñas, generadas por una fuerza del indentador de aproximadamente 50gf. Sin embargo, este algoritmo resulta no confiable para imágenes donde los defectos de la superficie del material son de aproximadamente del mismo tamaño como el de la indentación. La propuesta más reciente [45] de este grupo de investigadores, combina varios algoritmos desarrollados antes por ellos, dando como resultado un método automático para la segmentación de imagen, localización de la indentación y determinación de sus vértices. No obstante, este último método presenta altos errores tanto para detectar la indentación como para determinar sus vértices, para imágenes donde la frontera de la indentación carece de contraste con respecto al

fondo.

5.2.2 Métodos basados en un “mapa del área” según Maier, Uhl y Gadermayr

El método reportado por A. Maier y A. Uhl en [45], se ha calificado como el método más robusto y exacto, entre los actualmente conocidos, para la detección de los vértices de la indentación Vickers. La localización robusta de la indentación en la imagen bajo este método, es ejecutada usando un operador llamado *AreaMap* [23, 24], mientras que la aproximación del tamaño de la indentación es realizada en el espacio log-polar, y la determinación de la posición de los vértices se basa en un filtro Gabor, siendo este filtro un amapeamiento con un conjunto de plantillas de esquinas. Este método fue llamado ALG (“Area-map, Log-polar, Gabor”) por sus autores [45].

El operador *AreaMap* calcula el valor del área de una región de píxeles que satisfacen cierta propiedad, dependiendo de un pixel inicial. El *mapa del área* es creada por considerar cada pixel de imagen como punto inicial. Los autores usan dos operadores: *DecMap(p)* definido como el número de píxeles que pueden ser alcanzados por 4-caminos que inician en p y sobre los cuales los valores de gris solo disminuyen; *IncMap(p)* siendo el número de píxeles que pueden ser alcanzados por 4-caminos que inician en p y sobre los cuales los valores de gris solo aumentan. Los valores de estos dos operadores son calculados para todos los puntos de la imagen, generando nuevas dos imagenes, las cuales son inspeccionadas, en particular, valores extremos son buscados, para encontrar candidatos del punto centro de la indentación. Aunque este método logra buena robustéz y precisión útil (no muy alta), los cálculos requeridos por los operadores *DecMap* y *IncMap* son computacionalmente muy pesados y solamente aceptables para imágenes muy pequeñas.

5.3 Transformación log-polar

La transformación log-polar es una transformación geométrica que puede modelar la reorganización de información visual de la retina a la corteza visual de los primates. Matemáticamente, esta transformación mapea cada punto $p = (x, y)$ dado por sus coordenadas cartesianas en el plano euclidiano, entonces llamado *plano imagen*, al

plano log-polar, en el cual, p es representado por sus *coordenadas log-polares* (ξ, γ) . Debido a que el mapeo se basa en la función logarítmica para una variable compleja z , esta transformación también es llamada “el modelo $\log(z)$ ” en la literatura.

La idea intuitiva y motivación para transformar una imagen definida sobre una porción de \mathbb{R}^2 hacia el plano log-polar, sigue el objetivo de representar con mucho detalle, una región circular de la imagen original, alrededor de un punto centro especialmente preseleccionado. El “costo” para obtener la gran nitidez de detalle en la cercanía de este punto centro, es que regiones de la imagen original que se encuentran lejos de este punto, son representadas en el plano log-polar con poca resolución, es decir, con poco detalle.

Este tipo de representación es usado en sistemas visuales, tanto naturales como artificiales, que manejan grandes cantidades de información proveniente del entorno que lo rodea. Cuando es requerida una operación en tiempo real, como sucede con animales o robots en entornos dinámicos y no estructurados, la adquisición y el procesamiento de imagen deben realizarse en muy corto tiempo (unos pocos milisegundos), con el fin de proporcionar una respuesta suficientemente rápida a los estímulos externos. Los mamíferos, cuyos ojos son capaces de mover la retina, presentan un área de alta resolución en el centro del campo visual, llamado fovea. La distribución del campo receptivo dentro de la retina es fija, mientras que la fovea puede ser redirigida a otros objetivos por movimiento ocular.

Hallazgos biológicos en la corteza visual de los monos [11] demostraron que la dislocación de un estímulo ligero en la retina produce desplazamientos en la corteza que son inversamente proporcional a la distancia de la fovea. También, se encontró que la respuesta a los estímulos lineales originados en la fovea se encuentra aproximadamente a lo largo de las líneas de la corteza, y los estímulos circulares centrados en la fovea producen respuestas lineales en la corteza a orientaciones aproximadamente ortogonales [66]. De eso, Schwartz dedujo que la información transmitida entre la retina y la corteza visual es organizado en una aproximación logarítmica-polar [59]. La aportación de Schwartz no sólo fue la descripción del sistema de visión de los primates, sino también el modelo matemático con el cuál evidenció sus teorías. Este modelo aproxima la distribución de los fotorreceptores en la retina mediante una regla matemática sencilla. Primero, el modelo supone a los fotorreceptores como una malla de puntos divididos en sectores y anillos. En la intersección de cada sector con cada anillo, se encuentra posicionada el área central de un fotorreceptor. Otra

manera de verlo es como un conjunto de anillos de diferentes radios que contienen un número igual de divisiones. En cada división hay un fotoreceptor y esto implica que en cada anillo existe un mismo número de fotoreceptores. La distancia que separa a un anillo de otro es establecida por una proporción logarítmica que inicia a partir de la zona central. Esta proporción produce el efecto de una alta concentración de fotoreceptores alrededor del centro y dicha concentración decrece conforme los anillos estén más alejados del centro. El modelo matemático caracteriza cada fotoreceptor mediante dos datos: su distancia euclideana al centro de la retina, la cual está determinada en proporciones logarítmicas, y el ángulo en que se encuentra orientado con respecto al centro. De acuerdo a este modelo, las coordenadas cartesianas x, y de puntos del plano \mathbb{R}^2 , pueden ser transformadas en coordenadas log-polares ξ, γ , mediante las siguientes ecuaciones:

$$\xi = \ln \left(\sqrt{(x - x_0)^2 + (y - y_0)^2} \right) , \quad \gamma = \tan^{-1} \left(\frac{y - y_0}{x - x_0} \right) , \quad (5.2)$$

donde un punto centro $p_0 = (x_0, y_0)$ tiene que haber sido fijado. Nótese que ξ simplemente es el resultado de aplicar el logaritmo a la distancia euclidiana entre p y p_0 . Ambas coordenadas ξ y γ tienen una singularidad cuando $x = x_0$ y $y = y_0$. Como consecuencia, la transformada log-polar sólo representa puntos o píxeles del plano original con distancia estrictamente positiva del centro p_0 ; se genera una falta de información en el centro. La Figura 5.1 muestra cómo se mapea el plano de imagen con sus coordenadas x, y , al plano log-polar con sus ejes de coordenadas ξ, γ , donde ξ representa la distancia logarítmica-radial de $p = (x, y)$ al punto centro p_0 , y γ es la distancia angular entre p y la parte positiva del eje x del sistema de coordenadas trasladado para hacer coincidir su origen con p_0 . Para modelar el sistema biológico de visión mencionado arriba, la transformación proporciona una forma sencilla de agrupar la información de los receptores en una estructura matricial.

Para aplicar la transformación log-polar a imágenes digitales, definidas sobre el plano discreto donde los píxeles son representados por sus coordenadas cartesianas, primero debe ser seleccionado un pixel centro $p_0 = (x_0, y_0)$. Luego, el plano imagen es dividido en anillos y sectores alrededor de p_0 , vea Figura 5.1. También el plano log-polar es discretizado, es decir, las coordenadas log-polar calculadas para cada pixel del plano de imagen, son muestreadas o redondeadas. Así, es común que varios puntos p_1, p_2, \dots, p_k generen el mismo resultado de punto (ξ, γ) en el plano log-polar.

Eso hace necesario el diseño de un procedimiento de promediación, para calcular el valor de (ξ, γ) en la imagen log-polar, a partir de los valores $f(p_1), f(p_2), \dots, f(p_k)$ de la imagen digital original. La parte derecha de la Figura 5.1 muestra cómo cada circunferencia del plano imagen, más bien dicho, cada anillo de la imagen digital original, es mapeado a una línea vertical en el plano log-polar, más precisamente, a una columna en la matriz de imagen log-polar. Claramente, una columna con coordenada ξ pequeña corresponde a un anillo cercano a p_0 , mientras que una columna con coordenada ξ muy grande representa un anillo lejos de p_0 , supuestamente en la periferia de la región de interés, o ya fuera de ella, en la imagen original.

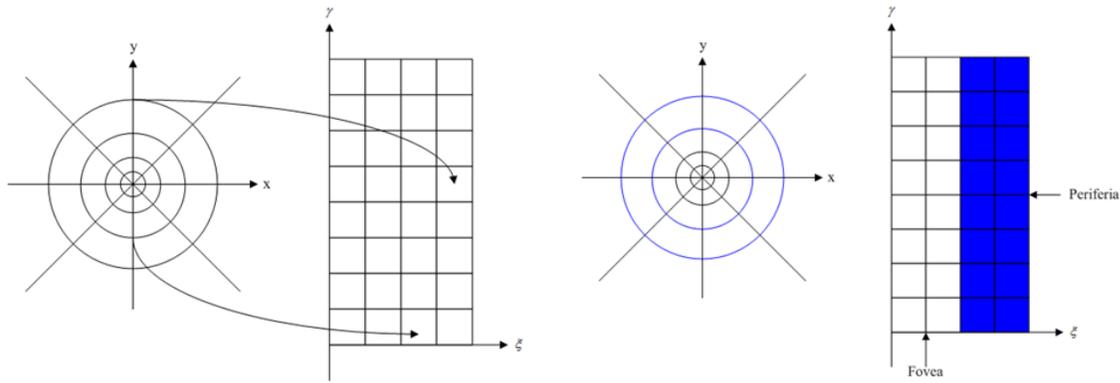


Figura 5.1: Mapeo del plano cartesiano al plano log-polar.

A partir de las ecuaciones 5.2, lo comentado arriba y la Figura 5.1, suponiendo que $p = (x, y)$ es un punto del plano imagen con su correspondiente representación (ξ, γ) en el plano log-polar, para el punto centro $p_0 = (x_0, y_0)$, se obtiene lo siguiente:

- Una rotación en el plano imagen corresponde a una traslación vertical en el plano log-polar: si p es rotado en \mathbb{R}^2 alrededor de p_0 por un ángulo θ , dando como resultado el punto p' , entonces, en el plano log-polar, $p' = (\xi, \gamma + \theta)$, presentando un cambio con respecto a p solo en su segunda coordenada.
- Un escalamiento en el plano imagen corresponde a una traslación horizontal en el plano log-polar: si p es escalado en \mathbb{R}^2 con un factor α , dando como resultado el punto $p' = \alpha p$, entonces la distancia angular entre p' y p_0 coincide con la entre p y p_0 . Sin embargo, la distancia radial entre p' y el centro p_0 , siendo eso la distancia euclidiana $d_{euclid}(p', p_0)$, es el producto de α con la distancia entre p' y p_0 . En consecuencia, en el plano log-polar, p' y p se distinguen

solamente en la primera coordenada, $p' = (\xi', \gamma)$, con $\xi' = \ln(d_{euclid}(p', p_0))$
 $= \ln(\alpha \cdot d_{euclid}(p, p_0)) = \ln(\alpha) + \ln(d_{euclid}(p, p_0)) = \ln(\alpha) + \xi$.

Como aplicación, consideremos la transformación log-polar de una línea recta del plano imagen que pasa por el centro p_0 , y su efecto a la rotación. Esta línea es la unión de dos rayos que inician en p_0 . Todos los puntos del primer rayo tienen la misma segunda coordenada polar γ , sin embargo, sus distancias euclidianas al centro p_0 , y entonces sus coordenadas ξ , son números crecientes positivos. Eso implica que este rayo corresponde a una línea horizontal en el plano log-polar, o a un renglón en la matriz de imagen digital log-polar. El segundo rayo inicia igualmente en p_0 pero tiene la dirección contraria, todos sus puntos tienen la segunda coordenada polar $\gamma + \pi$, pero sus primeras coordenadas ξ son números crecientes positivos. Es decir, este rayo es transformado a otra línea horizontal en el plano log-polar, o renglón en la matriz de imagen digital log-polar, la Figura 5.2 muestra esta situación. Si la línea recta completa es rotada en el plano de imagen, la rotación de cada punto corresponde a una traslación vertical en el plano log-polar, implicando que cada una de las dos líneas horizontales en el plano log-polar, correspondientes a un rayo en el plano imagen, es trasladada completamente en dirección vertical. En resumen, la línea recta completa original aparece en el plano log-polar como dos líneas horizontales, y la transformación de la línea recta completa rotada es un par de dos líneas horizontales, en otra posición de altura (coordenada γ) pero con la misma distancia vertical $\pi = 180^\circ$ entre ambas líneas.

Como otro ejemplo, consideremos un círculo en el plano de imagen \mathbb{R}^2 , o un anillo de la partición de los píxeles del plano discreto. Su transformación log-polar es una línea vertical en el plano log-polar, o una columna en la matriz de imagen digital log-polar, vea Figura 5.2, consistiendo por muchos puntos con la misma coordenada ξ (logaritmo de la distancia al centro) pero cubriendo todos los posibles ángulos (γ) desde 0 hasta 360 grados. Si el círculo original es escalado, por ejemplo, agrandado como en la Figura 5.2, el nuevo círculo corresponde a una nueva línea vertical o columna en el plano log-polar, que se encuentra en una posición ξ distinta a la columna generada por el círculo original como se observa en Figura 5.2.

Para imágenes digitales en ambos planos, el de imagen y el log-polar, los efectos arriba descritos pueden verse notablemente distorcionados, debido a errores inevitables de discretización o de interpolación. La Figura 5.3 muestra un ejemplo de imagen de indentación con su transformación log-polar. La región de indentación

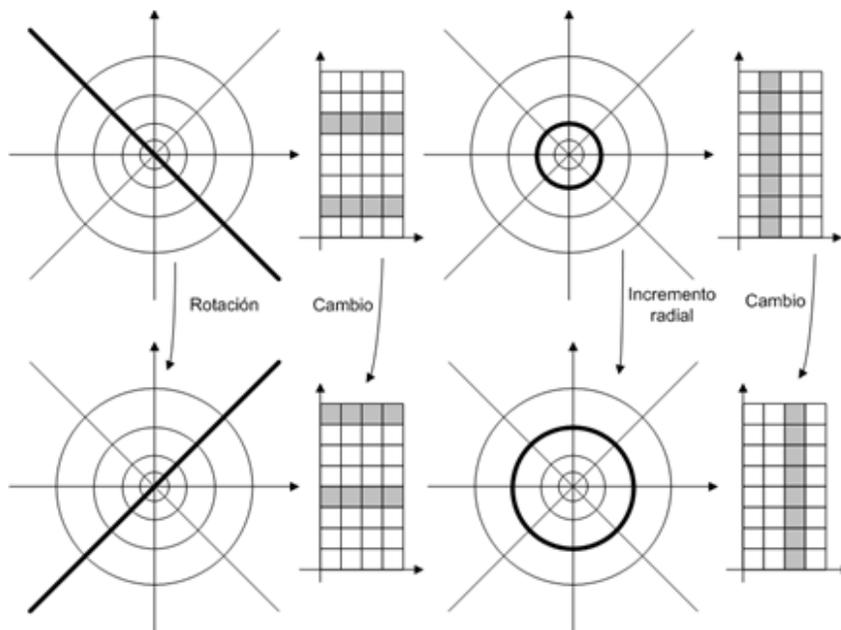


Figura 5.2: Efectos de la rotación de una línea y del escalamiento de un círculo en la transformación log-polar.

es transformada en la parte izquierda grande oscura de la imagen log-polar, los lados del rombo que aproxima la indentación, aparecen en el plano log-polar como segmentos de arcos alineados verticalmente.

La Figura 5.3 ilustra las dificultades de precisión al manejar la transformación log-polar y su transformación inversa, en imágenes digitales. En este ejemplo, los bordes de la indentación se encuentran entre el quinto y el séptimo de los círculos concéntricos usados para la partición del plano imagen. La figura en su imagen izquierda, presenta puntos específicos, uno para cada canto del rombo, con sus coordenadas cartesianas. Los cuatro puntos están situados sobre o muy cerca del círculo con radio 5, es decir, todos estos puntos deberían tener la coordenada polar $\xi = 5$. Sin embargo, la transformación log-polar experimentada en Matlab proporciona valores entre 4.49 y 4.62, en lugar de 5. Algunos de estos puntos en el plano log-polar, quedan “lejos” de los arcos que representan los bordes de la indentación. En la parte derecha de la Figura 5.3, las coordenadas llamadas “Aproximación cartesiana” fueron calculadas por transformación log-polar inversa, aplicada a los puntos que habían sido transformados antes. Eso debería recuperar los puntos originales del plano imagen, sin embargo, generan mucho error estos cálculos, en parte por efectos de discretización, en parte por el punto centro usado en la transformación, el cual

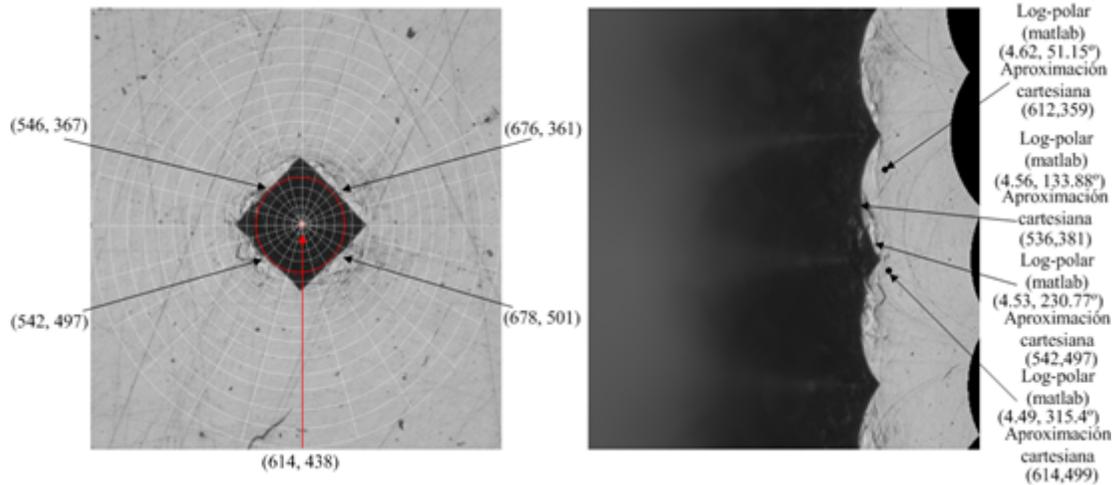


Figura 5.3: Imagen de indentación y su transformación log-polar.

pudo haber sido no el punto más apropiado. Por ejemplo, el punto original sobre el canto superior izquierdo de la indentación tiene las coordenadas $x = 546, y = 367$. La ecuación 5.2 proporciona el resultado $\xi = 4.58$, sin embargo, el resultado de Matlab es $\xi = 4.56$, lo cual equivale, bajo la transformación inversa, a una coordenada $x = 536$.

En el trabajo [45], la imagen log-polar es usada para estimar tamaño y orientación de la indentación. Eso se basa en las propiedades descritas arriba por las cuales, cualquier aumento o disminución de tamaño de la indentación causa un desplazamiento horizontal de la toda la imagen log-polar, mientras que cualquier rotación de la indentación, causa un desplazamiento vertical. Sin embargo, la detección de dichos desplazamientos en el plano log-polar enfrente varias dificultades. A parte, este método tiene los problemas de exactitud de cálculos mencionados arriba, y la necesidad de haber encontrado una estimación muy precisa y confiable del centro de indentación, la cual puede ser usada como centro para la transformación log-polar.

5.4 Propuesta de un nuevo algoritmo para analizar imágenes de indentación Vickers

5.4.1 Definición de materiales y condiciones para la medición de dureza Vickers mediante el nuevo algoritmo

Por las distintas propiedades de los materiales, causando una gran variedad de formas exactas y tamaños de indentaciones, probablemente no será posible encontrar un algoritmo de procesamiento de imagen que trabaje de manera eficiente y exacta para obtener de manera automática la dureza Vickers para una gran gama de materiales. En este trabajo consideramos dos tipos de materiales:

- Acero-316,
- Nitruro de Hafnio (HfN).

Para el material de acero se realizó un pulido en acabado espejo, obteniendo indentaciones con poco ruido en el fondo de la imagen, logrando no alterar de manera considerable los cantos de la indentación. También se realizaron indentaciones en Acero-316 con superficie áspera, presentando imperfecciones tales como: ralladuras, porosidad y surcos en la superficie del material. Además, se generaron indentaciones en muestras de acero con oxido, presentando defectos en la indentación, tales como dispersión en los pixeles que conforma los cantos de la indentación, y perdiendo uniformidad en el borde de la indentación. Para tales muestras, la superficie del material también aparece áspera, por lo cual puede presentar ralladuras, porosidad y surcos, como se puede observar en la Figura 5.4.

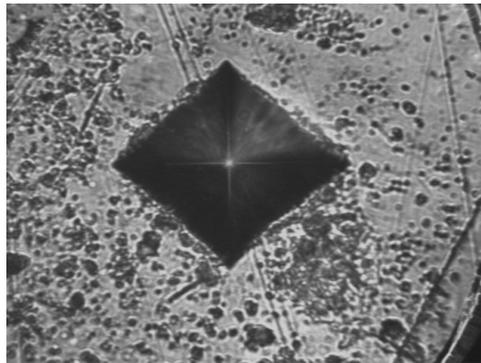


Figura 5.4: Imagen de indentación de una muestra con superficie áspera.

Otro material de prueba usado en este trabajo es el de películas delgadas de Nitruro de Hafnio (HfN), las cuales son utilizadas como recubrimiento en la construcción de circuitos integrados [50], ya que satisfacen los requerimientos de buena conductividad eléctrica y térmica, y promueven una barrera anticorrosiva que disminuye la velocidad de corrosión en componentes mecánicos [6, 61, 72]. No obstante, debido a la composición elástica de este material, la indentación generada en estas muestras, por lo regular presentan deformaciones, afectando a los cantos de la indentación.

Para todas las muestras de acero analizadas en esta tesis, se aplicó una carga por el indentador de 500 gf, mientras que una carga de 300 gf fue usada para las muestras de HfN.

El algoritmo propuesto fue implementado en Matlab para detectar la indentación Vickers, ubicándola a través de su centro, y después de estimar con exactitud los cuatro vértices y calcular la longitud promedio de los diagonales, obtener el número de dureza Vickers.

5.4.2 Idea general del nuevo algoritmo

Cada imagen de entrada es tratada como imagen 2D monocromática, con valores de gris en el intervalo $[0, 1]$, donde valores altos corresponden a píxeles con mucha iluminación ($1 = \text{blanco}$), y píxeles oscuros tiene valores bajos ($0 = \text{negro}$). Cada imagen contiene exactamente una indentación que aproximadamente es de forma de un rombo cuyas diagonales son casi alineadas con el sistema estándar de coordenadas cartesianas. Sin embargo, el tamaño exacto, posición y orientación exacta en la imagen (la indentación puede aparecer como ligeramente rotada), son características desconocidas. La indentación es la región a interés a ser detectada a analizada, en todas nuestras imágenes de entrada al algoritmo, esta región aparece como oscura sobre un fondo de imagen más claro.

En la mayoría de nuestras imágenes, los lados de la indentación son muy similares a segmentos de línea recta. Sin embargo, nuestro algoritmo no usa suposición alguna sobre esta propiedad, en contraste a muchos métodos de análisis de indentaciones, conocidos de la literatura [73].

Recordemos de arriba la fórmula 5.1 para calcular la dureza Vickers HV , usando la longitud promedio d de las dos diagonales de la indentación, $HV = 1854.4 \cdot F/d^2$, siendo F la carga del indentador dada en gf (gramos fuerza) [62].

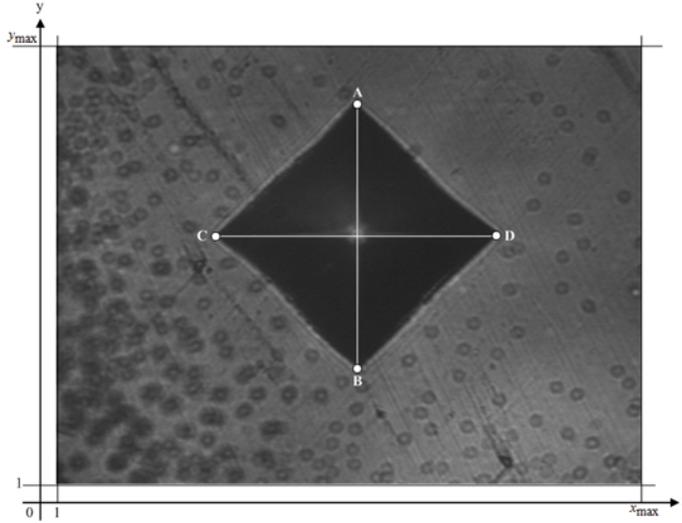


Figura 5.5: Indentación Vickers con vértices y diagonales marcados.

La Figura 5.5 muestra una imagen de indentación donde los vértices A, B, C, D de la indentación y sus diagonales son marcados en blanco. Denotando el vértice norte por $A = (A_x, A_y)$, el vértice sur por $B = (B_x, B_y)$, el de occidente por $C = (C_x, C_y)$, y el de oriente por $D = (D_x, D_y)$, las diagonales son los segmentos de línea \overline{AB} y \overline{CD} , y la longitud promedio de ellas, necesaria para calcular la dureza HV , es dada por

$$d = (d_1 + d_2)/2 \quad (5.3)$$

donde $d_1 = \sqrt{(A_x - B_x)^2 + (A_y - B_y)^2}$ y $d_2 = \sqrt{(C_x - D_x)^2 + (C_y - D_y)^2}$.

Nuestro algoritmo para el análisis de indentaciones Vickers, consiste en cinco pasos:

1. Segmentación de imagen por binarización.
2. Filtrado morfológico y crecimiento de regiones.
3. Detección y localización de la indentación.
4. Determinación de los vértices A, B, C, D , usando una técnica de detección de esquinas, y una evaluación y selección de los puntos de esquina detectados.
5. Cálculo de las longitudes de diagonales d_1, d_2 , y del número de dureza Vickers HV , por las ecuaciones 5.1 y 5.3.

5.4.3 Contribución del nuevo algoritmo al estado del arte

Aunque nuestro algoritmo sigue un esquema similar al de algunos algoritmos para el análisis de indentaciones Vickers, conocidos de la literatura, se distingue de los anteriores en varios aspectos:

- Nuestro algoritmo determina el umbral para la binarización, de cada imagen de entrada, a base de valores de gris extremos y el valor promedio, lo cual es diferente de las técnicas de binarización aplicadas en [10, 22, 23, 41, 47, 63].
- Nuestro filtro morfológico usa un elemento estructurante de tipo y tamaño distintos de los elementos estructurantes utilizados en todos los filtros morfológicos aplicados en métodos conocidos para el tratamiento de imágenes de indentación Vickers.
- Nuestro método de detección de los vértices de la indentación, no se basa en detectores de cantos como ha sido común en métodos anteriores [10, 33, 24, 23], ni en plantillas de esquinas como en [23, 24, 45]. Nuestro algoritmo aplica el detector de esquinas según Harris y Stephen [28], luego realiza un proceso de evaluación y selección de los puntos de esquina encontrados, para así obtener los cuatro vértices de la indentación.

5.4.4 Detalles del algoritmo propuesto

Paso 1: Segmentación de imagen

El proceso de segmentación de imagen inicia con una binarización, usando el valor de gris promedio de la imagen de entrada como umbral, y la diferencia entre éste y el valor de gris máximo, como criterio discriminante. Estos valores son características globales de la imagen de entrada. Denotando la imagen de entrada como $F = (f(x, y))$, con $x = x_1, \dots, x_{max}$, $y = y_1, \dots, y_{max}$, vea Figura 5.5, su valor de gris promedio es dado por

$$f_{mean} = \frac{1}{x_{max}y_{max}} \sum \{f(x, y) : x = x_1, \dots, x_{max}; y = y_1, \dots, y_{max}\},$$

y valor de gris máximo por

$$f_{max} = \max\{f(x, y) : x = x_1, \dots, x_{max}; y = y_1, \dots, y_{max}\}.$$

Basado en estos valores, se propone el siguiente *criterio de binarización*: para cada pixel de entrada $p = (x, y)$: p es considerado un *pixel de interés* si

$$|f(x, y) - f_{max}| > f_{mean} .$$

El resultado es una imagen binaria $G = (g(x, y))$, $x = x_1, \dots, x_{max}$, $y = y_1, \dots, y_{max}$, donde cada pixel de interés es presentado como negro, y cada otro pixel es blanco, vea Figura 5.6. La región de interés, supuestamente la indentación, será encontrada como subconjunto de los píxeles negros. Esta representación, contraria al estándar en Matlab, respeta el hecho que la indentación es una región oscura en la imagen original.

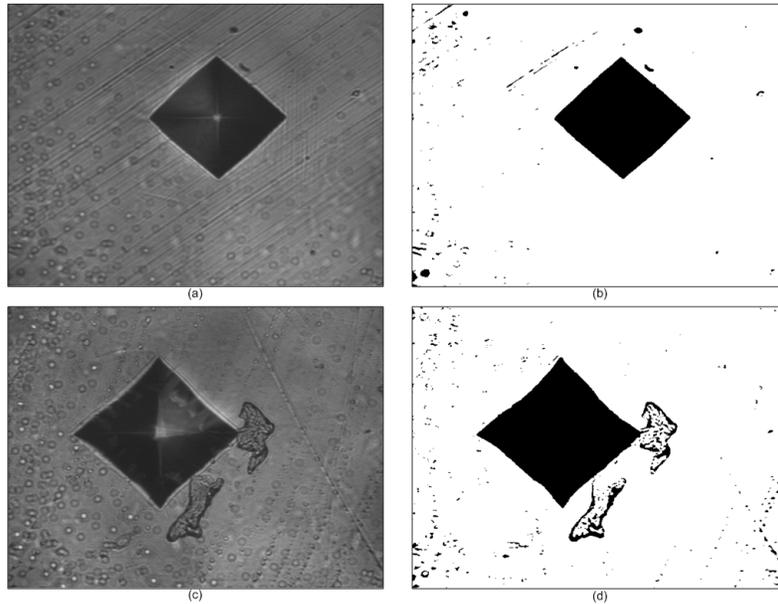


Figura 5.6: Indentaciones Vickers con sus correspondientes imágenes binarias.

Paso 2: Filtrado morfológico y crecimiento de regiones

Como muestra la Figura 5.6, muchos píxeles son encontrados como falsos positivos de la región de interés. Para eliminar estos puntos negros que no serán de la indentación, filtros morfológicos han sido aplicados en algoritmos propuesto antes [41]. Sin embargo, nuestro filtro morfológico aplica un tipo y medida de elemento estructurante no usado antes en el análisis de imágenes de indentación Vickers.

Para la imagen binaria G , cualquier conjunto A de píxeles negros, y un (pequeño) conjunto de píxeles S , llamado en la literatura *elemento estructurante*, la *dilatación* de A con S es definida como el siguiente conjunto de píxeles [19] :

$$A \oplus S = \{p = (p_x, p_y) : x_1 \leq p_x \leq x_{max}, y_1 \leq p_y \leq y_{max}, \\ p = a + s = (a_x + s_x, a_y + s_y) \text{ para } a = (a_x, a_y) \in A, s = (s_x, s_y) \in S\}$$

Usar la dilatación como método de filtrado, significa extender el conjunto de píxeles negros en G , dándoles el valor de negro a todos los píxeles de $A \oplus S$. La *erosión* de A por S es definida como el conjunto de píxeles

$$A \ominus S = \{p \in A : p + s \in A \text{ para todo } s \in S\}.$$

Erosión, vista como método de filtrado, reduce al conjunto de píxeles negros en G , convirtiendolos en blanco todos los píxeles (negros) de A que no pertenecen a $A \ominus S$. Un pixel de A es preservado como negro solamente si en su posición específica en la imagen, el conjunto de píxeles entero S cabe dentro de A . La transformación de una erosión seguida por una dilatación es llamada *apertura morfológica*, mientras que, *cerradura morfológica* es definida como erosión de un conjunto dilatado.

Nuestro filtro propuesto consiste de una apertura morfológica del conjunto de píxeles negros de G , por un elemento estructurante distinto a los usados en trabajos previamente publicados. Basado en el conocimiento a priori que la región de indentación tiene forma rómbica, usamos la forma de rombo (en Matlab llamado “diamante”) como elemento estructurante S , esto es, un disco según la d_4 -métrica. Escogemos S de una medida suficientemente grande para suprimir ruido estructural en la imagen, pero más pequeño que la medida mínima de pequeñas indentaciones que esperamos encontrar en la imagen. En nuestro algoritmo, S es un rombo (o diamante) con radio de 10 píxeles. Esta apertura morfológica substancialmente reduce ruido estructural de la imagen, haciendo posible en el siguiente paso, encontrar la indentación como la región 8-conectada más grande de píxeles negros.

Por otro lado, este filtro preserva bien tamaño y forma de la indentación. En particular, los cuatro vértices de la indentación son bien preservados bajo este procedimiento, lo cual no se lograría usando otros elementos estructurales como discos euclidianos, octágonos, o cuadrados.

La Figura 5.7 muestra imágenes de indentación (columna izquierda), cada una

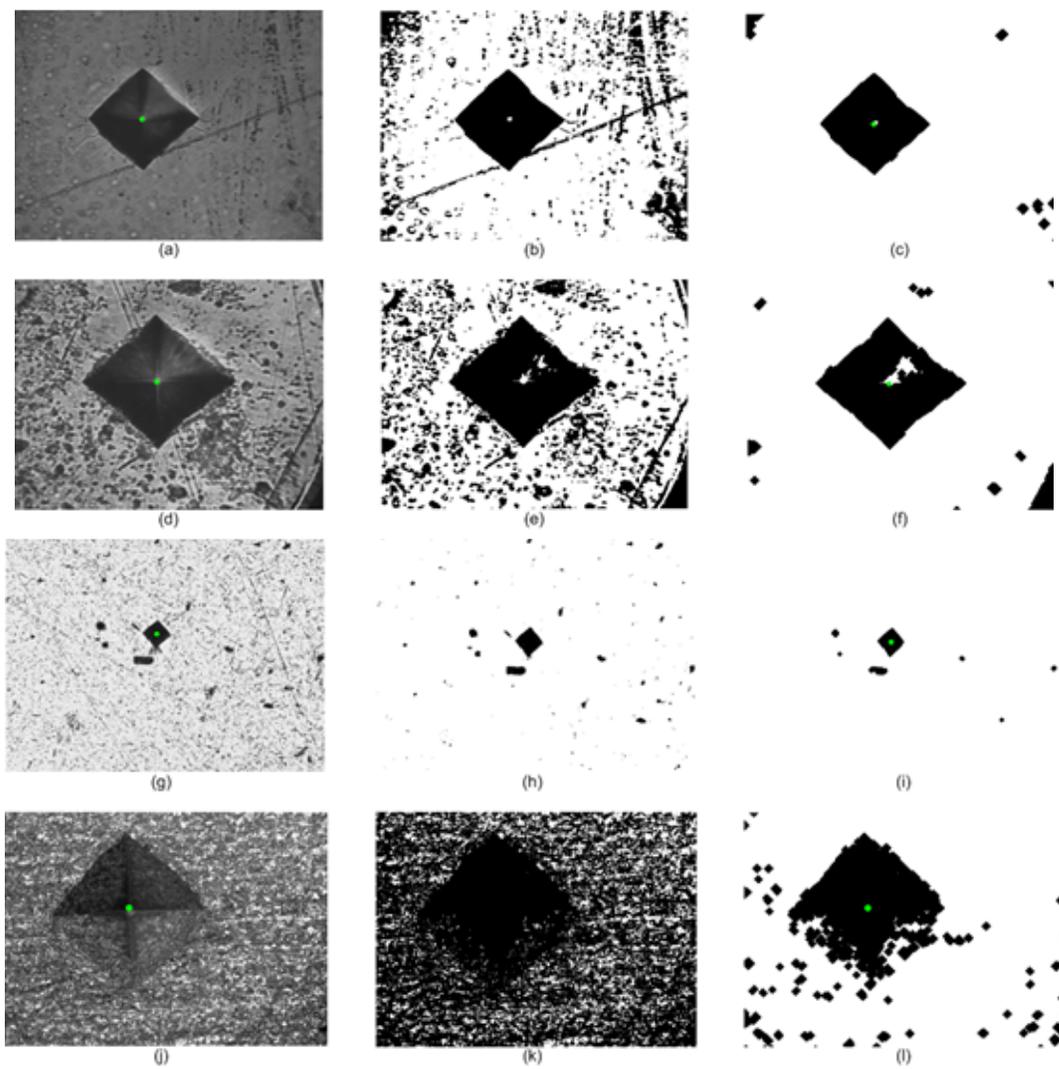


Figura 5.7: Imágenes de indentación, sus versiones binarias y los resultados del filtro morfológico.

con su versión binaria (columna en medio), así como el resultado de la aplicación del filtro morfológico (columna derecha). Note que cada ejemplo presenta algún problema específico: el primero contiene una indentación ligeramente deforme, el segundo presenta imperfecciones en la superficie, la tercera indentación es muy pequeña, y el cuarto caso presenta dificultad para ser procesado debido al bajo contraste de imagen. En la primer y tercer columna de esta figura, los centros de la indentación han sido marcados en color verde, su detección será explicada más adelante.

La segmentación de imagen es completada mediante crecimiento de regiones, lo cual es un procedimiento estándar en el procesamiento digital de imagen y puede ser ejecutado de manera eficiente. Para nuestra imagen binaria, ya filtrada, el crecimiento de regiones consiste en determinar todas las componentes 8-conectados de píxeles negros. Durante este proceso, diversas características de cada región pueden ser fácilmente calculadas, por ejemplo,

- su número de píxeles que estima el área de la región, o
- las coordenadas mínimas y máximas x y y de todos sus puntos, eso proporciona los datos del rectángulo que circumscribe la región, llamado *caja delimitadora*.

Paso 3: Detección y localización de la indentación

Nuestro algoritmo determina la indentación como la componente negra 8-conectada de máxima área, de la imagen ya binarizada y filtrada morfológicamente. Esto completa la detección e identificación de la región de interés, o equivalentemente, de la indentación. Posteriormente, el punto central $q = (q_x, q_y)$ de esta región que sea denotada por M , es calculado como su centro gravitacional: si $p^i = (p_x^i, p_y^i)$, $i = 1, \dots, m$, denotan los píxeles de M , entonces

$$q_x = \frac{1}{m} \sum_{i=1}^m p_x^i, \quad q_y = \frac{1}{m} \sum_{i=1}^m p_y^i. \quad (5.4)$$

Eso completa la localización de la indentación, o región de interés. Ejemplos de puntos centros determinados bajo este procedimiento son marcados con color verde en la Figura 5.7.

Las coordenadas del centro de la indentación son usadas en el siguiente paso como un dato auxiliar para detectar los vértices. También usaremos la caja delimitadora de la indentación, dada como $[x_{Rect1}, x_{Rect2}] \times [y_{Rect1}, y_{Rect2}]$, la cual fue determinada

durante el crecimiento de regiones en el Paso 2.

Paso 4: Determinación de los vértices

Para determinar los vértices A, B, C, D de la indentación, vea Figura 5.5, aplicamos una técnica conocida de detección de esquinas, según Harris y Stephen [28], seguido por una evaluación y selección automática de los puntos de esquinas encontrados.

El detector Harris-Stephen es un operador de imagen local, el cual para cada pixel p decide si se trata de un punto esquina, basado en los valores de los gradientes digitales G_x (en dirección x) y G_y (in dirección y) de píxeles dentro de una pequeña vecindad N_p de p , y los eigenvalores λ_1, λ_2 de la matriz

$$H = \begin{pmatrix} H_x & H_{xy} \\ H_{xy} & H_y \end{pmatrix}, \text{ donde } H_x = \sum_{q \in N_p} G_x(q)^2, \quad (5.5)$$

$$H_y = \sum_{q \in N_p} G_y(q)^2, \quad H_{xy} = \sum_{q \in N_p} G_x(q)G_y(q).$$

Usamos la 3x3-vecindad de p como N_p . Como fue desarrollado en [28], λ_1, λ_2 ambos son números reales no negativos que tienen valores altos cuando el punto p es un *punto de esquina*, definido como la intersección de dos cantos sobresalientes de orientaciones distintas en la imagen. No es necesario calcular λ_1 y λ_2 explícitamente, como fue mostrado en [28], en cambio, se determina el valor de la *función de respuesta a esquinas* (*corner response function*) definida por

$$R(p) = \det H - k \cdot (\text{traza}(H))^2, \quad (5.6)$$

donde la constante k tiene que ser determinada empíricamente, la recomendación es $k \in [0.04, 0.06]$ [28]. Nótese que $\det H = \lambda_1\lambda_2$, $\text{traza}(H) = \lambda_1 + \lambda_2$, pero la determinante y la traza de la matriz H pueden ser calculadas también directamente, lo cual además es sencillo para esta matriz de tamaño 2x2. En consecuencia,

$$R(p) = (H_x H_y - H_{xy}^2) - k \cdot (H_x + H_y)^2. \quad (5.7)$$

Un pixel p es considerado **punto de esquina** (según Harris-Stephen) si $R(p)$ sobrepasa cierto umbral [28].

Para hacer más eficiente esta parte del algoritmo, es suficiente buscar puntos

de esquina en una región que cubre la indentación y que, además, contiene las vecindades N_p para todos los píxeles p de la indentación, para poder calcular la función $R(p)$. Una solución simple es tomar una extensión de la caja delimitadora $[x_{Rect1}, x_{Rect2}] \times [y_{Rect1}, y_{Rect2}]$: para el caso que las vecindades $N(p)$ son los cuadrados de 3×3 píxeles, centrados en p , el conjunto de los píxeles $a = (a_x, a_y)$ tal que

$$x_{Rect1} - 1 \leq a_x \leq x_{Rect2} + 1, \quad y_{Rect1} - 1 \leq a_y \leq y_{Rect2} + 1,$$

es una región de búsqueda útil para puntos esquina. Incluso es necesario calcular los gradientes G_x y G_y solamente para esta región de búsqueda. Además, se busca puntos esquina sólo entre los píxeles negros.

Los vértices A, B, C, D de la indentación son determinados a partir del conjunto E de píxeles (negros) que pertenecen a la indentación y fueron detectados como puntos de esquina según Harris-Stephen, por el siguiente procedimiento de evaluación y selección.

Determinación de vértices para indentaciones no rotadas, o rotadas por un ángulo pequeño:

Cuando conocimiento a-priori permite suponer que la indentación está alineada con el sistema de coordenadas, o puede presentar una rotación por un ángulo dentro de un intervalo muy pequeño, los vértices A, B, C, D pueden ser encontrados fácilmente como los puntos esquina situados al extremo norte, sur, oeste y este, respectivamente, de la región de indentación.

A partir del conjunto E , determinamos A como el píxel norte extremo de E , es decir, con máxima coordenada y , en el sistema de coordenadas como en Figura 5.5. En caso de que existan muchos puntos de E con esta propiedad, seleccionamos el punto cuya coordenada x es más cercana a la coordenada x del centro gravitacional q de la indentación que fue calculado en el Paso 3. Similarmente, los píxeles del extremo sur de E son aquellos con coordenada y mínima, los píxeles del extremo oeste de E son aquellos con coordenada x mínima, y los píxeles del extremo este de E son aquellos con coordenada x máxima. Los vértices B, C, D son seleccionados lo más cerca posible a las líneas rectas vertical y horizontal que pasan a través del centro q de la indentación.

Determinación de vértices para indentaciones con orientación arbitraria:

Cuando no se pueden asumir restricciones fuertes en los ángulos de rotación para las

indentaciones, el promedio a , de las coordenadas x de todos los puntos de extremo norte de E , es calculado, y entonces A es seleccionado como el punto extremo norte de E cuya coordenada x está más cercana al valor a . De manera análoga, los promedios

- b , de las coordenadas x de todos los puntos de extremo sur de E ,
 - c , de las coordenadas y de todos los puntos de extremo oeste de E ,
 - d , de las coordenadas y de todos los puntos de extremo este de E ,
- son utilizados para determinar los vértices B, C, D .

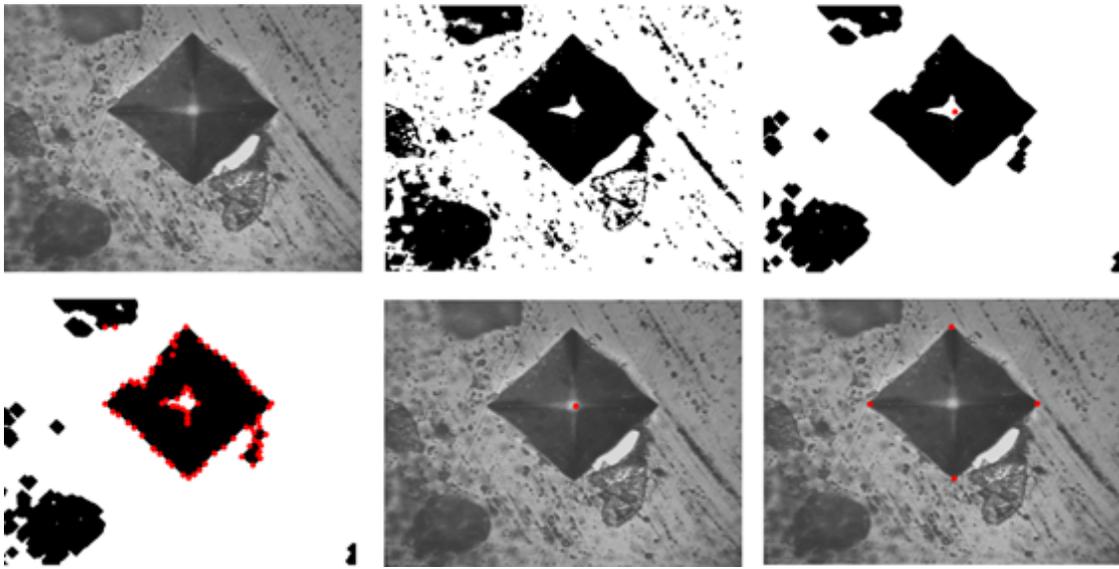


Figura 5.8: Ejemplo de aplicación del algoritmo propuesto completo.

La Figura 5.8 muestra un ejemplo de aplicación de nuestro algoritmo completo. El primer renglón presenta la imagen de indentación original, su versión binaria y la imagen resultado del filtro morfológico, donde además, el centro gravitacional q es marcado con color rojo. En el segundo renglón, la primera imagen muestra los puntos esquina según Harris-Stephen que pertenecen a la región de indentación, marcados en color rojo. La siguiente imagen presenta la imagen de entrada con el centro de indentación sobrepuesto en rojo. En la última imagen, los cuatro vértices de la indentación, encontrados por nuestro algoritmo, son presentados en color rojo sobre la imagen original de entrada.

Tabla 5.1: Conversión de píxeles a longitudes, en dependencia de la magnificación.

Magnificación (Lente)	Ancho de la imagen (640 píxeles) corresponde a:	Altura de la imagen (480 píxeles) corresponde a:
10x	800 μm	600 μm
50x	160 μm	120 μm
100x	80 μm	60 μm

Paso 5: Cálculo del número de dureza Vickers

A partir de los cuatro vértices de la indentación, son calculadas las longitudes de las diagonales d_1 , d_2 , y la longitud d promedio, según las ecuaciones 5.3. Antes del cálculo del número de dureza Vickers HV por la ecuación 5.1, se debe hacer la conversión de longitudes dadas en píxeles hacia la medida μm del mundo real. Eso se realiza con la ayuda de la Tabla 5.1, donde la magnificación se refiere al lente del microscopio del microdurómetro Mitutoyo HM-125.

5.5 Banco de imágenes de indentación Vickers

Otra contribución importante de la tesis consiste en la generación de un banco de imágenes de indentación Vickers, con anotaciones de experto. Hasta la fecha de la defensa de esta tesis, fueron adquiridas e incluidas en este banco 230 imágenes. Los datos reportados en este banco incluyen anotaciones de un experto humano quien obtuvo los resultados de medición de las coordenadas de los vértices de indentación y de las longitudes de diagonales, de manera convencional manual.

Gran parte de este banco, 185 imágenes, organizadas en cuatro grupos, con sus anotaciones de experto y resultados detallados de experimentación con nuestro algoritmo, puede ser accesado en internet desde septiembre 2018 [12].

Se ha organizado el banco completo de actualmente 230 imágenes, para hacerlo accesible en internet por un servidor de la Universidad Veracruzana.

Capítulo 6

Experimentos

6.1 Ajuste experimental

Para probar el algoritmo propuesto de procesamiento y análisis de imágenes de indentación Vickers, fue generado un banco de datos de estas imágenes, adquiridas por una cámara CCD implementada en el microdurómetro Mitutoyo modelo HM-125 [48]. Este microdurómetro contiene un microscopio óptico, el cual proporciona magnificaciones de 10x, 50x y 100x.

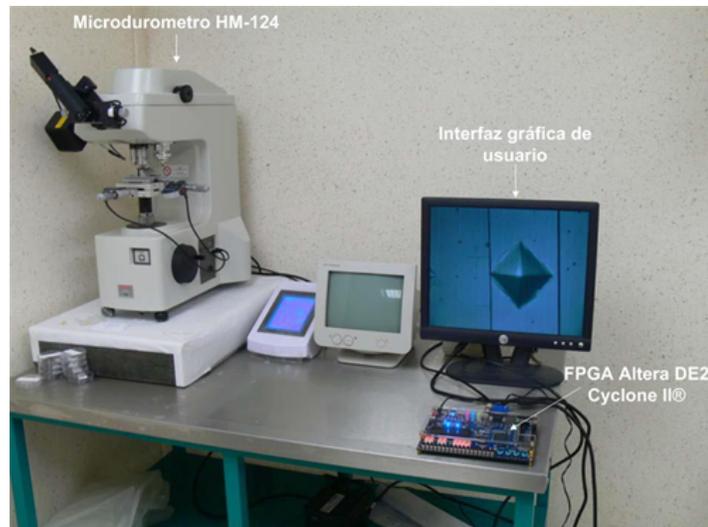


Figura 6.1: Sistema de adquisición de imagen para el microdurómetro HM-125.

Un FPGA Altera DE2-115 Cyclone II y la interfaz gráfica de usuario, que fue desarrollada en este trabajo de tesis, son usados para la adquisición de imagen en

tiempo real, como fue explicado en el capítulo 4 de la tesis. Nuestro sistema de adquisición de imagen almacena las imágenes de indentación en formatos bmp y png, como imágenes en escala de gris. Todas las imágenes de prueba tienen una medida de 640x480 píxeles, muchas de ellas fueron adquiridas por un lente de magnificación de 50x. Cada imagen contiene exactamente una huella de la indentación de forma rómbica con orientación casi alineada. La Figura 6.1 muestra el montaje experimental con el sistema de adquisición de imagen.

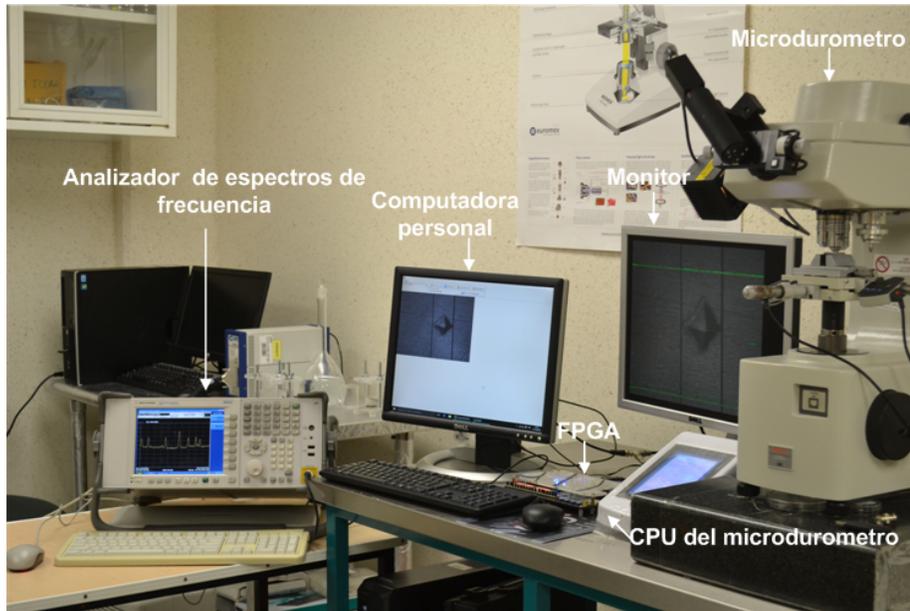


Figura 6.2: Montaje experimental para medir el espectro de frecuencia a la entrada del decodificador de video del FPGA.

La interfaz gráfica interactúa con el FPGA para aplicar filtros digitales que pueden ser configurados por el usuario, con el fin de eliminar ruido eléctrico y magnético que pudieran interferir en la captura de imagen. Como se mostró en el capítulo 4, para medir el ruido que interfiere en la adquisición de imagen a través de la etapa de entrada del FPGA, se utilizó un analizador de espectro de frecuencia N9020A (Agilent Technologies), vea Figura 6.2.

La Figura 6.3 presenta la respuesta en frecuencia al aplicar un filtro configurado mediante la interfaz gráfica de usuario y ejecutado por el sistema FPGA, el cual atenúa considerablemente la cantidad de armónicos en el espectro de frecuencia. Para este trabajo de investigación, el filtro que dió mejor resultado, fue un filtro tipo Bessel de cuarto orden, rechaza-banda, con frecuencia de corte mínima de 10

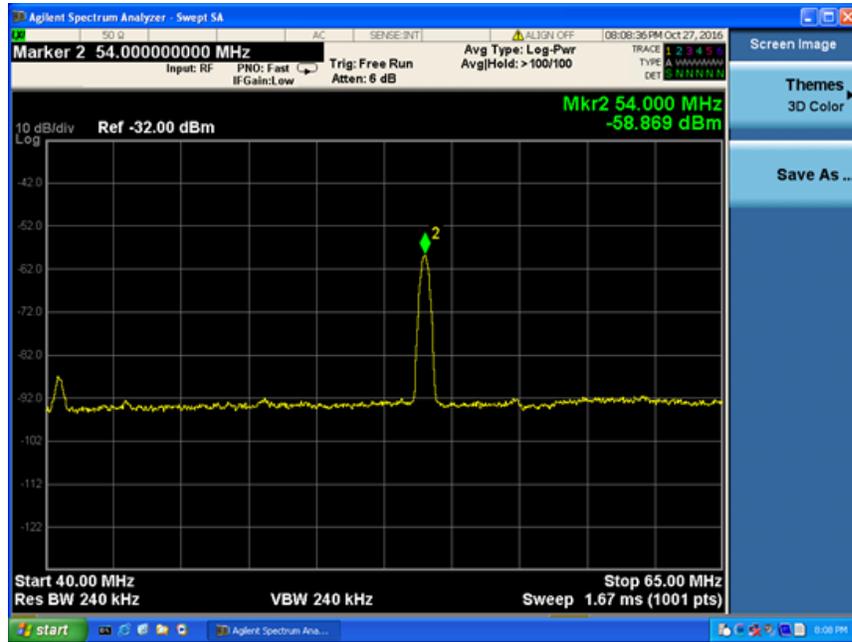


Figura 6.3: Espectro de frecuencia del voltaje de salida del decodificador de TV filtrado.

Tabla 6.1: Recursos usados por el sistema de adquisición de imagen.

Total de elementos lógicos	2731/33216
Registro total	1613/33216
Número total de bits usados de memoria	426/475
Multiplicadores embebidos	4/70
Número total de PLLs	2/4
Velocidad de escritura promedio	25 Mb/s

MHz y frecuencia de corte máxima de 60 MHz, como fue reportado en [16].

En la Figura 6.4 se puede apreciar la gran diferencia entre una imagen de indentación captada sin filtrado alguno (lado izquierdo), y una imagen de indentación captada por nuestro sistema de adquisición embebido al FPGA, que incluye un procesamiento apropiado, es decir filtrado, de la señal de video (lado derecho). El filtro implementado en el FPGA fue configurado a través de la interfaz gráfica de usuario, su efecto es una supresión de ruido eléctrico [35, 14, 16]. En este experimento, se aplicó una fuerza del indentador de 0.5 kgf a una muestra de acero-316 de 3 mm de espesor.

En nuestro sistema de adquisición de imagen usamos una relación de muestreo

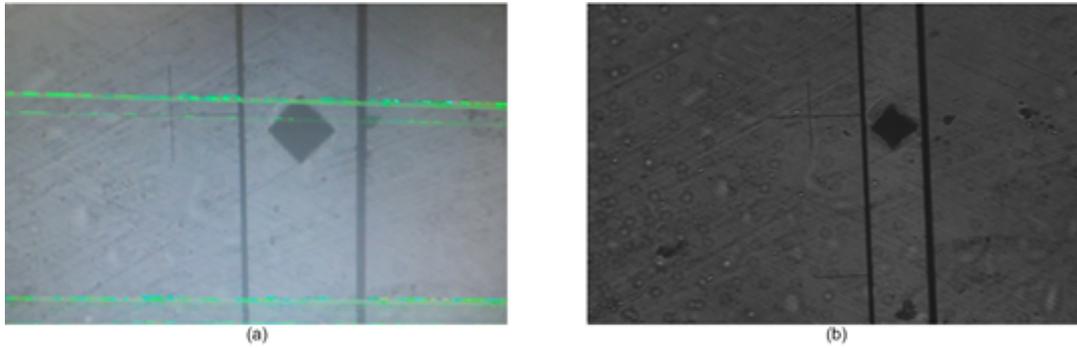


Figura 6.4: (a) Imagen de indentación con ruido eléctrico. (b) Imagen captada a través de nuestro sistema de adquisición de imagen que incluye filtrado embebido en el FPGA.

de 4:1:1, alcanzando un tiempo para comprimir una imagen en el decodificador de TV alrededor de 22 ms. La eficiencia total del sistema asegura un procesamiento de imagen en tiempo real. Los recursos computacionales utilizados por el sistema son reportados en la Tabla 6.1.

Además, el consumo de potencia de nuestro sistema de adquisición de imagen embebido en el FPGA es de 81.36 mW. Todos los parámetros fueron calculados por el software Quartus II. Una característica especial de nuestro sistema de adquisición de imagen consiste en el control de la calibración de las lentes del microdurómetro. El sistema es capaz de realizar ajuste en tiempo real de la longitud focal de los objetivos de la máquina de prueba de dureza. A través de ejecutar pruebas por largo tiempo, observamos que el sistema captura imágenes de buena calidad, dado que el sistema no produce fenómenos anormales. Como resultado, encontramos que nuestro sistema es estable, confiable y satisface los requerimientos de un procesamiento para captura de imagen en tiempo real.

6.2 Análisis de imágenes de indentación

El algoritmo propuesto de procesamiento y análisis de imágenes de indentación, que fue presentado en detalle en el capítulo 5 de esta tesis, fue aplicado a una variedad de imágenes con indentaciones con diversas características, como las siguientes:

- Indentaciones con uniformidad de valores de gris en sus cantos, y sin exceso de ruido en la superficie.

- Indentaciones con superficies ásperas, las cuales presentan ralladuras y ruido, generado por porosidad y/o exceso de grasa en la superficie del material.
- Indentaciones con deformidad en sus bordes, presentado dispersión de los píxeles de sus cantos.
- Indentaciones con deformaciones dentro del área de la indentación, originando discontinuidades en el borde de la indentación.
- Imágenes con bajo contraste en los bordes de la indentación, con respecto al fondo de imagen.

Las imágenes de indentación analizadas en este trabajo de tesis, tienen características muy similares a las que fueron presentadas en [73]. El algoritmo propuesto en esta tesis, también fue aplicado a algunas imágenes de indentación reportadas como parte de la base de datos mencionada en [23, 24, 44, 45], donde nuestro algoritmo obtiene resultados similares a los de estos autores.

Concretamente, nuestro algoritmo fue probado con 230 imágenes adquiridas por nuestro sistema de adquisición embebido al FPGA Altera DE2-115, el cual recibe las imágenes microscópicas provenientes de la cámara CCD del microdurómetro. Estas 230 imágenes fueron agrupadas como sigue:

- **Grupo 1 (Acero pulido):** 70 imágenes de muestras de acero-316 con superficie pulida especular.
- **Grupo 2 (Acero no pulido):** 70 imágenes de muestras de acero-316 con superficie pulido áspero.
- **Grupo 3 (Deformadas):** 50 imágenes de muestras de nitruro de hafnio (HfN), donde muchas indentaciones resultan deformadas.
- **Grupo 4 (Imperfecciones):** 40 imágenes de muestras de acero-316 que presentan fuertes imperfecciones como manchas de grasa, porosidad, rasguños, y presencia de otros pequeños artefactos.

En todos los experimentos con muestras de acero, se aplicó una carga por el indentador de 500 gf, mientras que una carga de 300 gf fue usada para las muestras de HfN.

Como resultado de nuestro ajuste experimental, cada imagen contiene exactamente una huella de indentación, la cual constituye la región de interés que debe

ser detectada y analizada. Esta región de interés, siendo la huella, es llamada *indentación*. Recordemos que cada indentación, aproximadamente, tiene la forma de un rombo, o cuadrado rotado por 45 grados. En todas nuestras imágenes de prueba, las indentaciones satisfacen bien dicha forma de rombo, con sus diagonales casi alineadas con el sistema de coordenadas cartesianas estándar. Sin embargo, el tamaño exacto, la localización en la imagen, y la orientación exacta, son características de indentación desconocidas de antemano.

Para todas las imágenes, las coordenadas de los vértices de la indentación fueron obtenidas por un experto humano de manera convencional manual, a partir de las cuales, las longitudes de las dos diagonales, y, basado en la longitud promedio, el número de dureza Vickers fueron calculados. Estos datos fueron comparados con los correspondientes resultados determinados por nuestro método computacional. Los resultados detallados de los experimentos con las 230 imágenes son reportados en tablas, una para cada grupo de imágenes, incluidas en el Anexo B de esta tesis.

La Tabla 6.2 presenta un análisis del error relativo del valor promedio de las longitudes de diagonales de la indentación, por un lado, determinado a partir de los vértices encontrados por nuestro algoritmo, por otro lado, calculado de los vértices obtenidos manualmente por un experto humano. Para este error relativo, los datos obtenidos por el método manual son tomados como datos de referencia (“datos de verdad”, *ground truth*).

La Figura 6.5 muestra ejemplos de las imágenes del Grupo 1, donde los vértices de indentación determinados por nuestro método computacional están marcados en rojo, y donde el error relativo de la longitud promedio de diagonales resultó en menos de 1%. El número puesto bajo cada imagen en la figura, corresponde el número de la imagen usado en la tabla del Grupo 1, incluida en el Anexo B. En esta tabla, todos los datos y resultados para las 70 imágenes del Grupo 1 - coordenadas de los vértices determinados, longitudes de ambas diagonales y su promedio, número de dureza calculado, son reportados en detalle. El Anexo B contiene una tabla para cada uno de los cuatro grupos de las 230 imágenes en total.

La Figura 6.6 muestra imágenes del Grupo 1, con error de longitud promedio de diagonales entre 1% y 3% (primera fila hasta imagen 69 de la sexta fila), y con error entre 3% y 5% (a partir de imagen 30 sexta y séptima fila). En estas

Tabla 6.2: Distribución de errores relativos de la longitud promedia de diagonales.

	Grupo 1 <i>Acero pulido</i> (70)	Grupo 2 <i>Acero no pulido</i> (70)	Grupo 3 <i>Deformadas</i> (50)	Grupo 4 <i>Imperfecciones</i> (40)
Número de imágenes con error relativo menor a 1%	42	53	29	23
Número de imágenes con error relativo entre 1% y 3%	21	9	14	14
Número de imágenes con error relativo entre 3% y 5%	7	3	3	1
Número de imágenes con error relativo entre 5% y 9%	0	3	3	0
Número de imágenes con error relativo mayor a 9%	0	2	0	0
Número de imágenes con resultado erróneo (<i>outlier</i>)	0	0	1	2

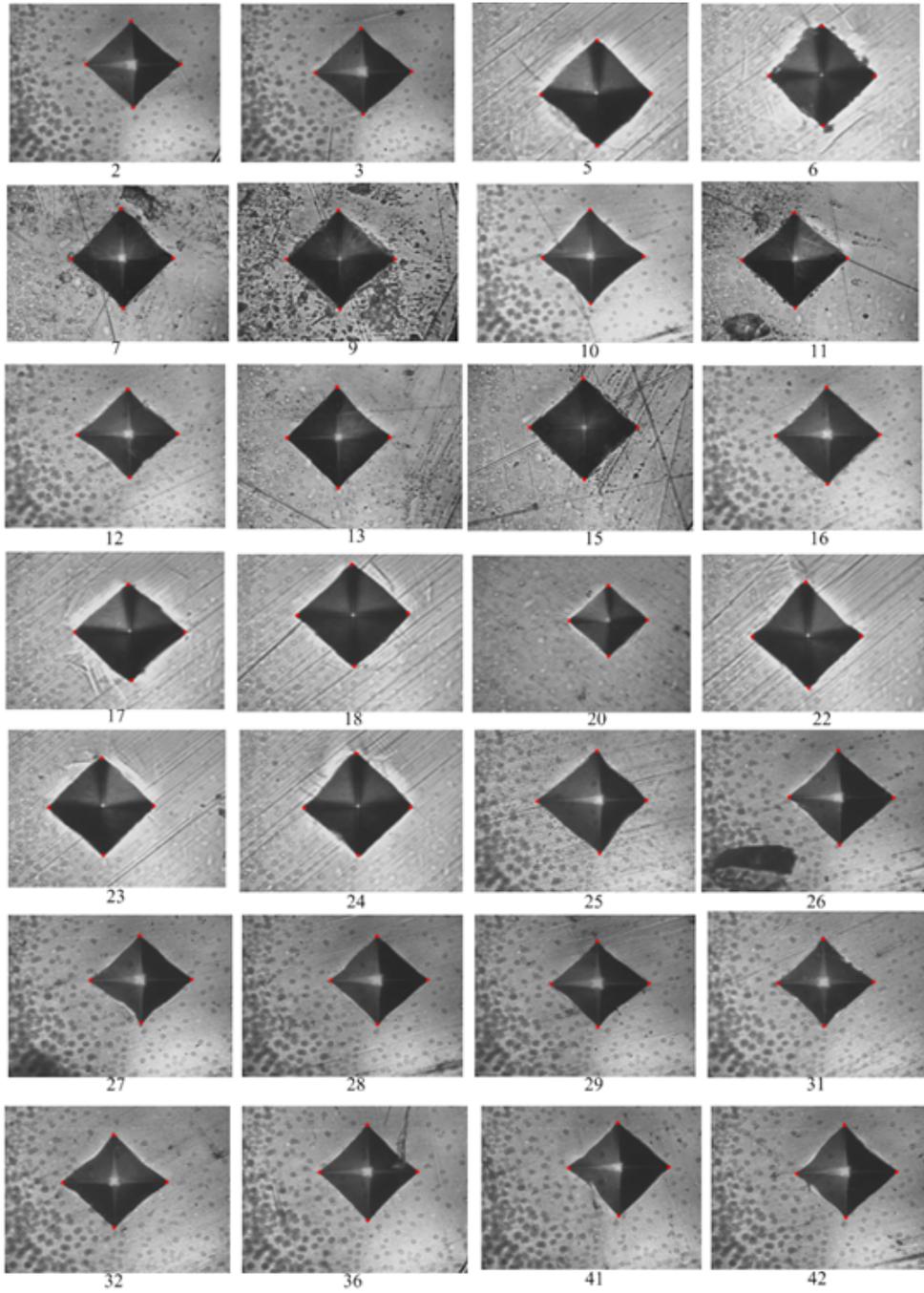


Figura 6.5: Imágenes del Grupo 1 con error menor a 1% en la longitud promedio de diagonales, con sus vértices de indentación marcados en color rojo.

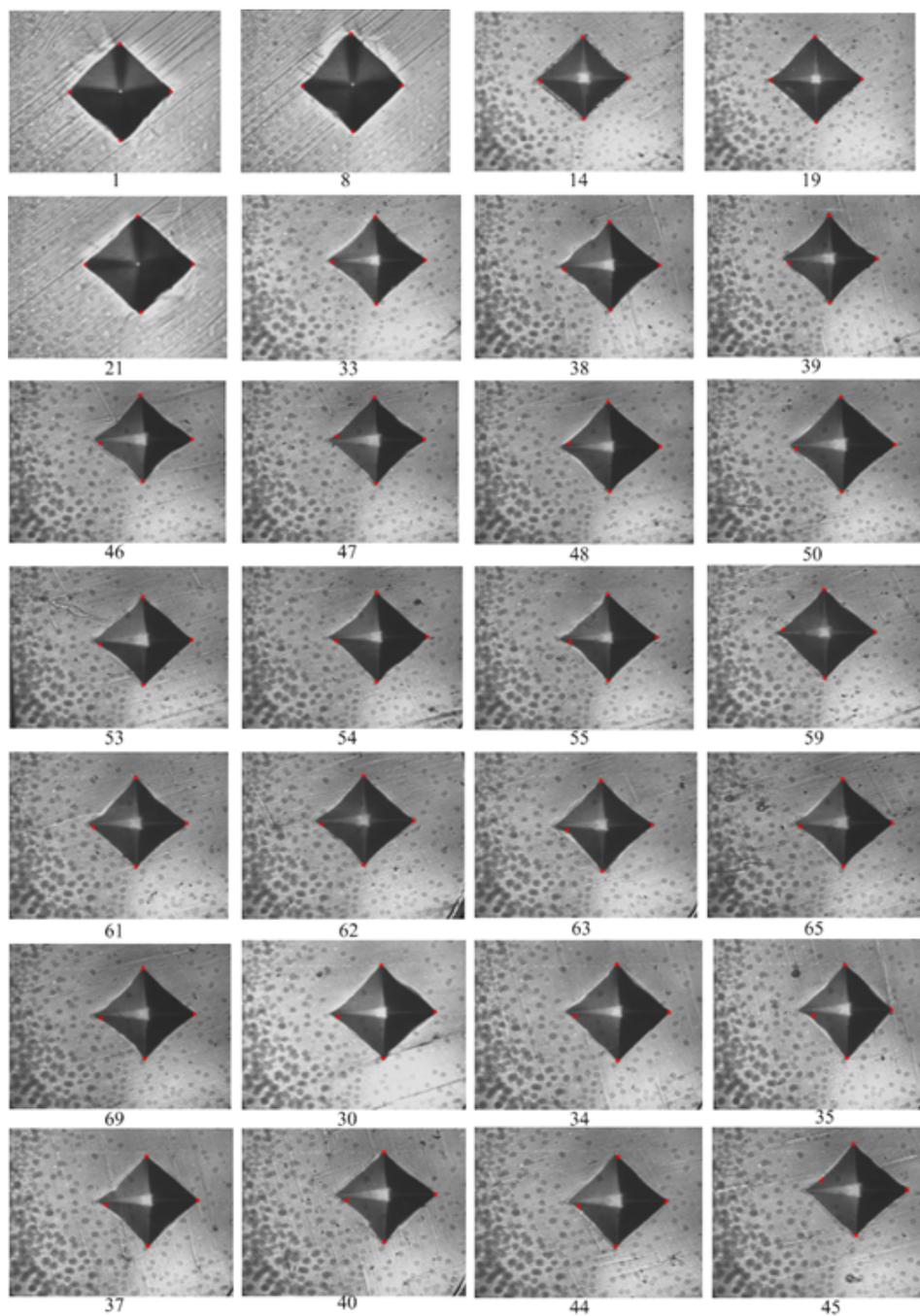


Figura 6.6: Grupo 1: imágenes con error de longitud promedio de diagonales entre 1% y 3% (hasta imagen 69), y con error entre 3% y 5% (a partir de imagen 30).

imágenes el error se ve incrementando, debido a que en algunas muestras, la luz de la cámara CCD se ve reflejada y capturada con mucho brillo sobre el contorno de la indentación.

La Figura 6.7 muestra imágenes del Grupo 2 con error de longitud promedio de diagonales menor de 1%. En las imágenes de la Figura 6.8, igualmente del Grupo 2, se observa mayor el reflejo de la cámara CCD sobre la superficie de la muestra del material, por lo cual, los errores son ligeramente más altos.

Las imágenes del Grupo 2 de la Figura 6.9 presentan mayor efecto de iluminación en la superficie del material, causando errores relativos de la longitud promedio de diagonales entre 3% y 5%. Conforme aumenta la intensidad luminosa reflejada en la superficie del material, la binarización va generando mayor error, con el resultado de mayores errores en la detección de los vértices de la indentación, como se puede observar en las imágenes de la Figura 6.10 del Grupo 2.

La imagen 65 de la Figura 6.10 se puede considerar como fuera del alcance de la detección de los vértices de la indentación por parte de nuestro algoritmo, causado principalmente por limitaciones de la segmentación por binarización con un umbral que se basa en propiedades globales de la imagen.

La Figura 6.11 muestra dos imágenes del Grupo 2 cuyo error de longitud promedio de diagonales resultó mayor de 9%, por lo cual, estas imágenes realmente están fuera del alcance del algoritmo propuesto. Estos errores se presentan por el alto brillo generado por la luz del mismo microdurómetro. Es posible solucionar este problema para ambas imágenes, ajustando el valor del umbral para mejorar el resultado de la binarización. Uno de los problemas pendientes de este trabajo de tesis, para investigaciones futuras, es desarrollar un método de preprocesamiento para tratar estos efectos de brillo causado por inhomogeneidades de la iluminación.

Las Figuras 6.12 y 6.13 muestran imágenes del Grupo 3 con error relativo de longitud promedio de diagonales menor a 1%, y entre 1% y 3%, respectivamente. Como se puede observar en la imagen 40, la indentación está deformada por expansión. Aún así, nuestro algoritmo logra detectar los vértices de la indentación con un error bajo. Cabe notar que debido a las deformaciones de la indentación, la medición visual manual realizada por un experto humano también tiene resultados con cierta incertidumbre.

Las Figuras 6.14 y 6.15 contienen imágenes del Grupo 3 con error de longitud promedio de diagonales entre 1% y 3%, entre 3% y 5%, respectivamente.

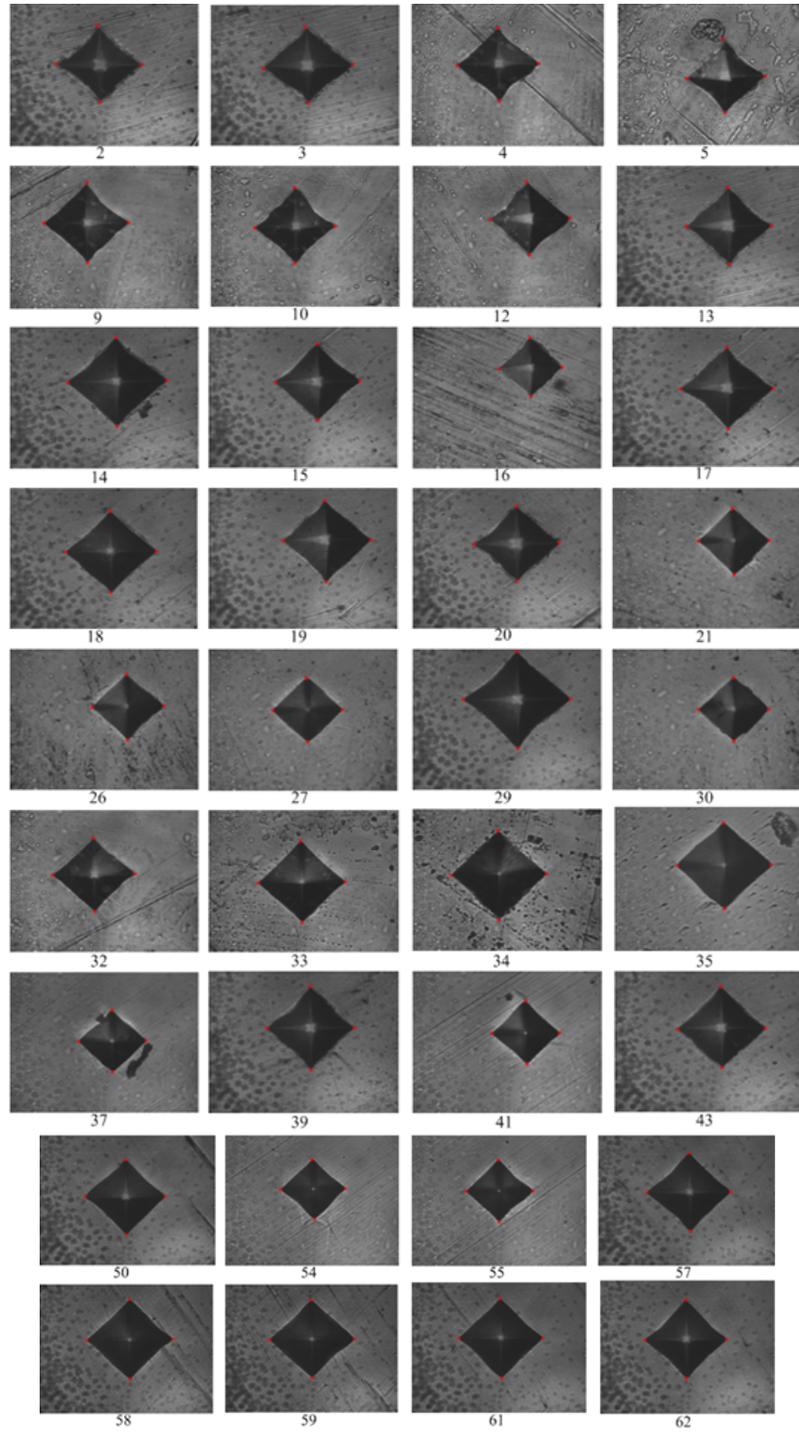


Figura 6.7: Grupo 2: imágenes con error de longitud promedio de diagonales menor de 1%.

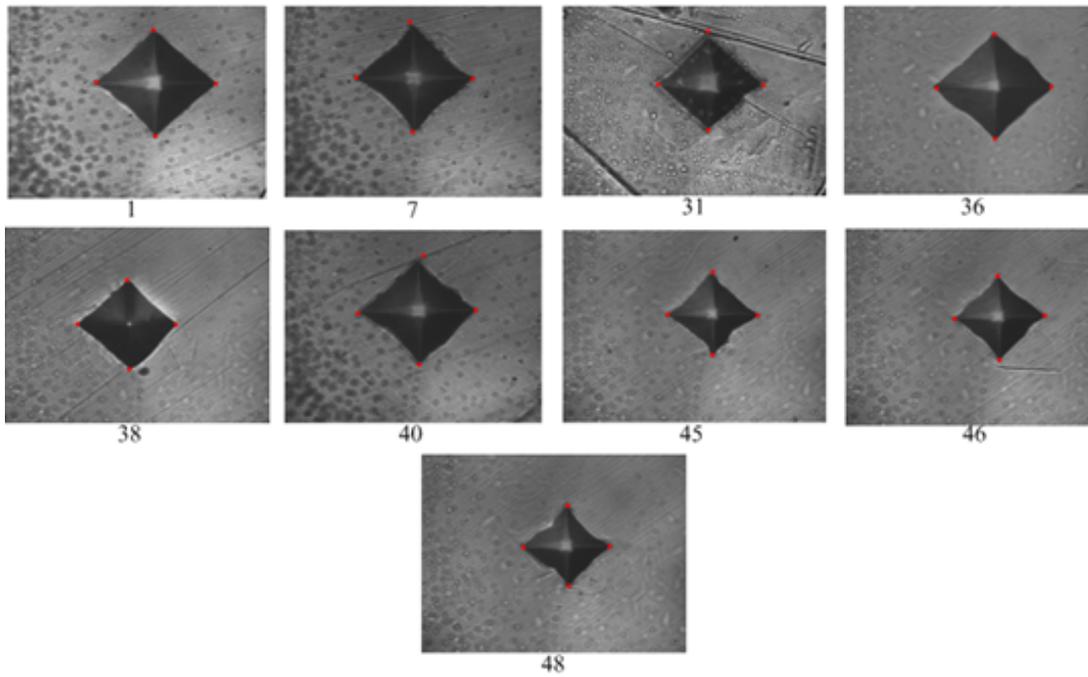


Figura 6.8: Grupo 2: imágenes con error de longitud promedio de diagonales entre 1% y 3%.

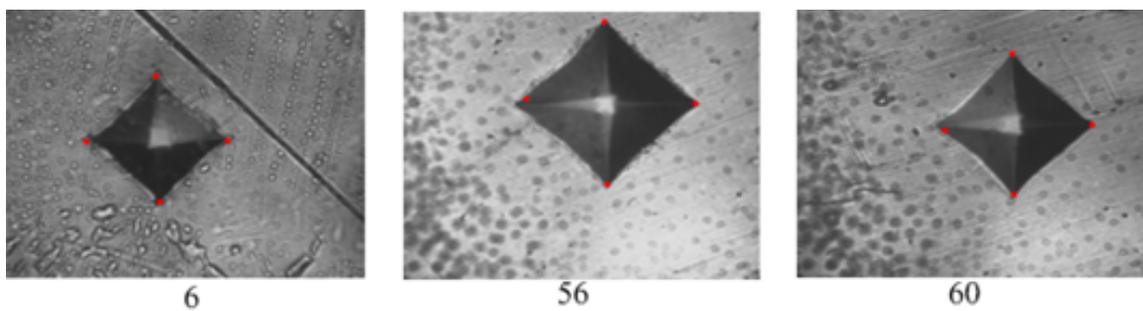


Figura 6.9: Grupo 2: imágenes con error de longitud promedio de diagonales entre 3% y 5%.

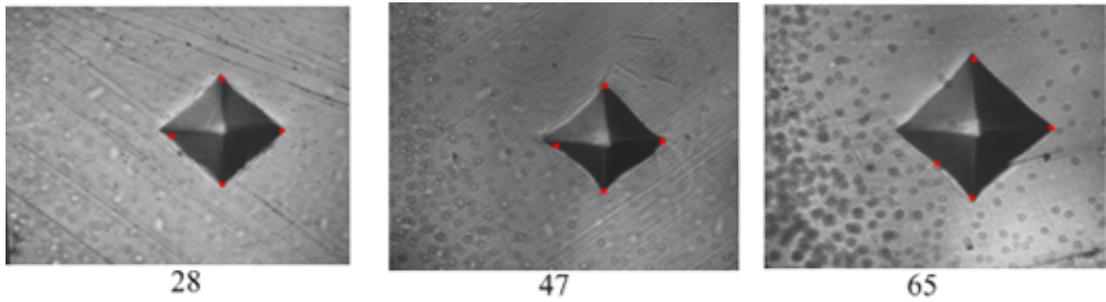


Figura 6.10: Grupo 2: imágenes con error de longitud promedia de diagonales entre 5% y 9%.

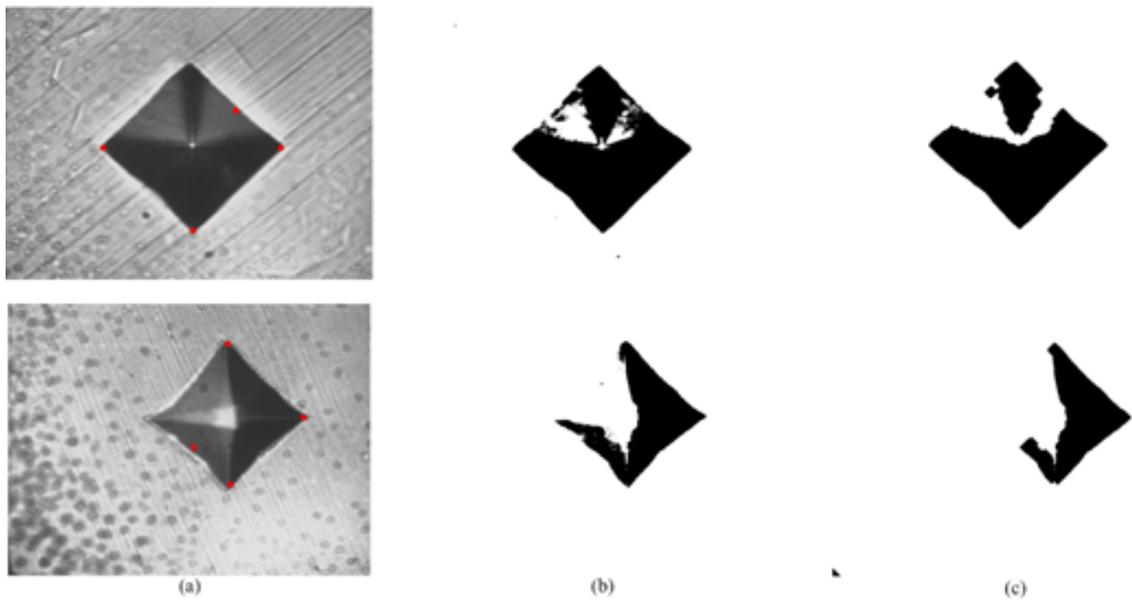


Figura 6.11: Grupo 2: dos imágenes fuera del alcance del algoritmo. (a) imagen original con los vértices detectados sobrepuestos. (b) resultado de binarización. (c) respuesta del filtro morfológico.

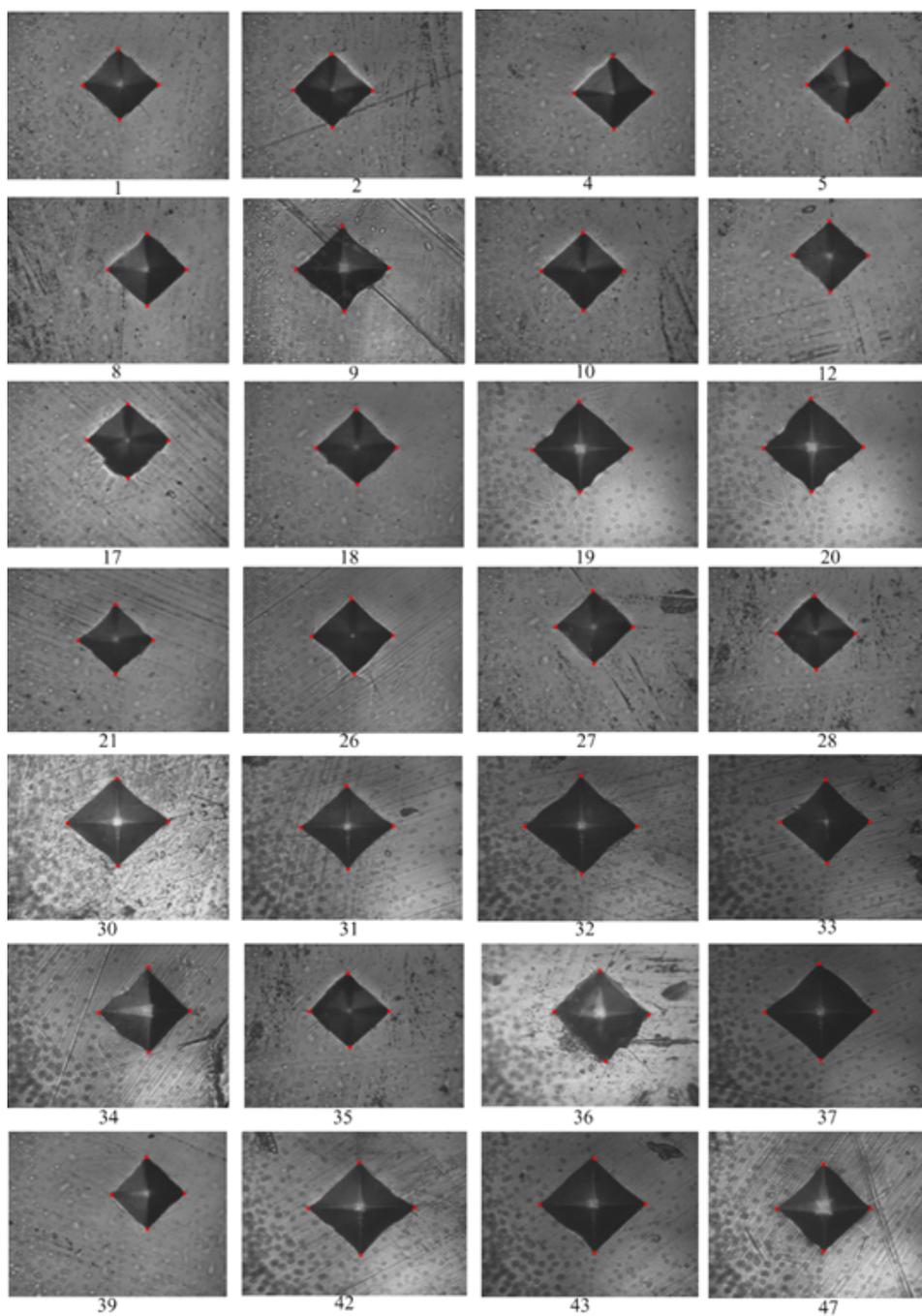


Figura 6.12: Grupo 3: imágenes con error de longitud promedio de diagonales menor de 1%.

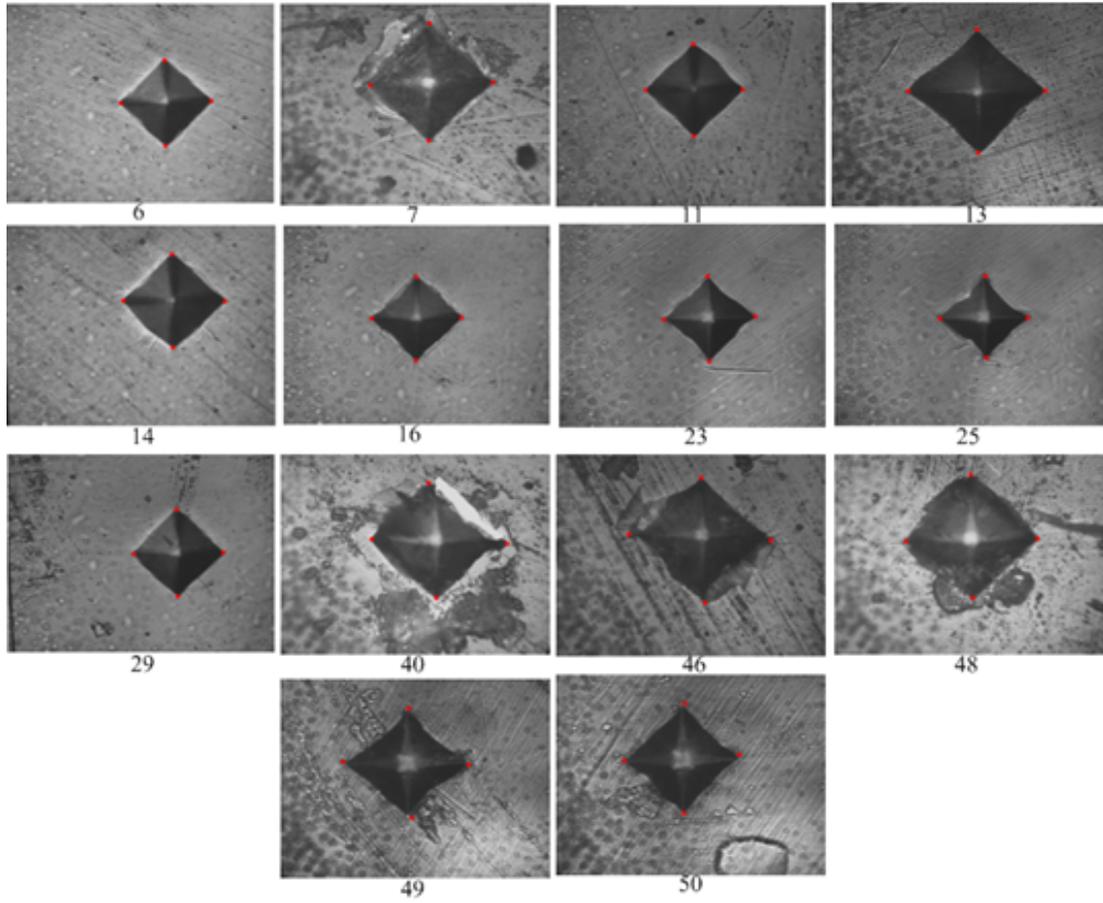


Figura 6.13: Grupo 3: imágenes con error de longitud promedia de diagonales entre 1% y 3%.

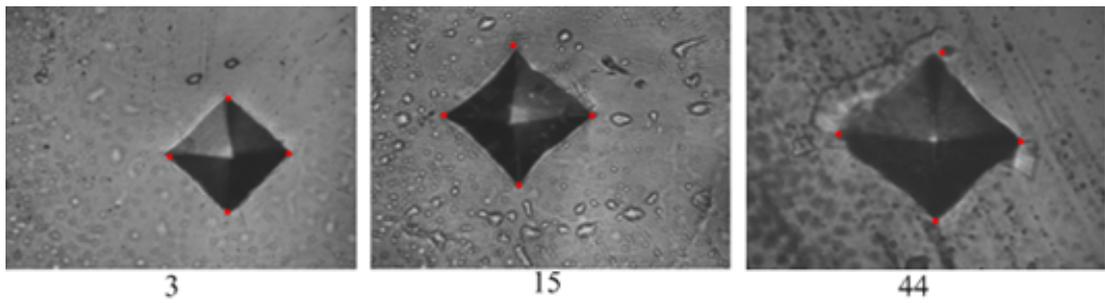


Figura 6.14: Grupo 3: imágenes con error de longitud promedia de diagonales entre 3% y 5%.

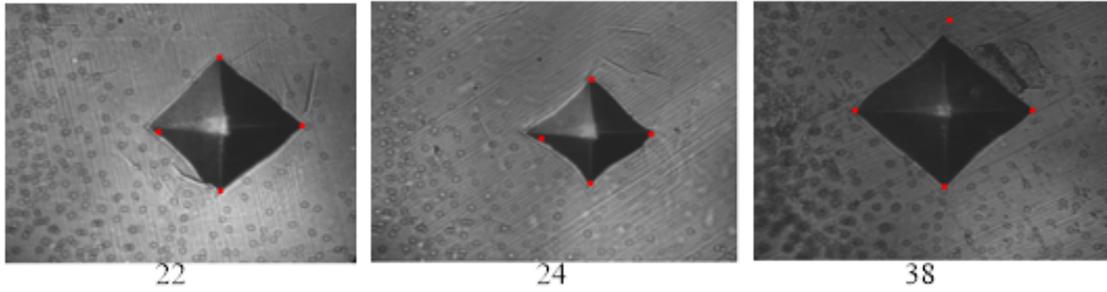


Figura 6.15: Grupo 3: imágenes con error de longitud promedio de diagonales entre 5% y 9%.

Para una imagen del Grupo 3, el algoritmo propuesto no logra detectar uno de los cuatro vértices de la indentación, como muestra la Figura 6.16. Esto se debe al bajo contraste entre el fondo de la imagen y ciertas regiones de la indentación. El vértice norte de la indentación no es encontrado como consecuencia de un resultado erróneo de la segmentación por binarización.

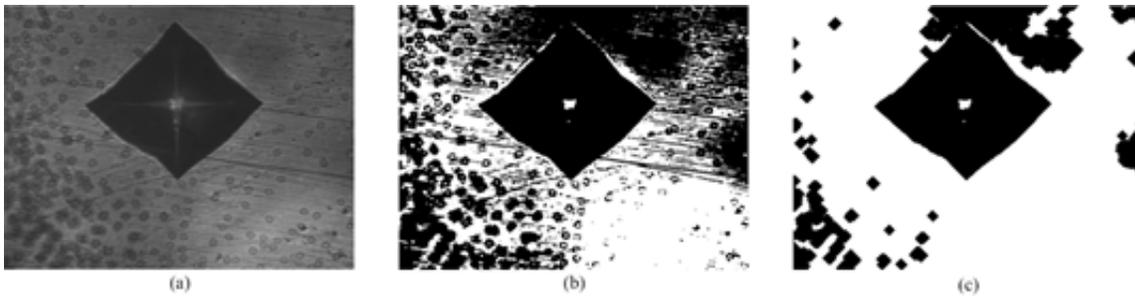


Figura 6.16: Grupo 3: imagen fuera del alcance del algoritmo. (a) imagen original. (b) resultado de binarización. (c) respuesta del filtro morfológico.

Las Figuras 6.17 y 6.18 presentan imágenes del Grupo 4, reportando sus intervalos de errores relativos de longitud promedio de diagonales. El error alto, en el intervalo de 3% a 5%, de la imagen 35 de la Figura 6.18 es causado por el bajo contraste entre la indentación y el fondo de la imagen.

El Grupo 4 presenta dos imágenes que resultaron fuera del alcance del algoritmo, donde en cada una, el vértice norte de la indentación, no se logró obtener. Estas dos imágenes presentan bajo contraste entre el fondo y cierta región en la parte norte de la indentación, como lo muestra la Figura 6.19.

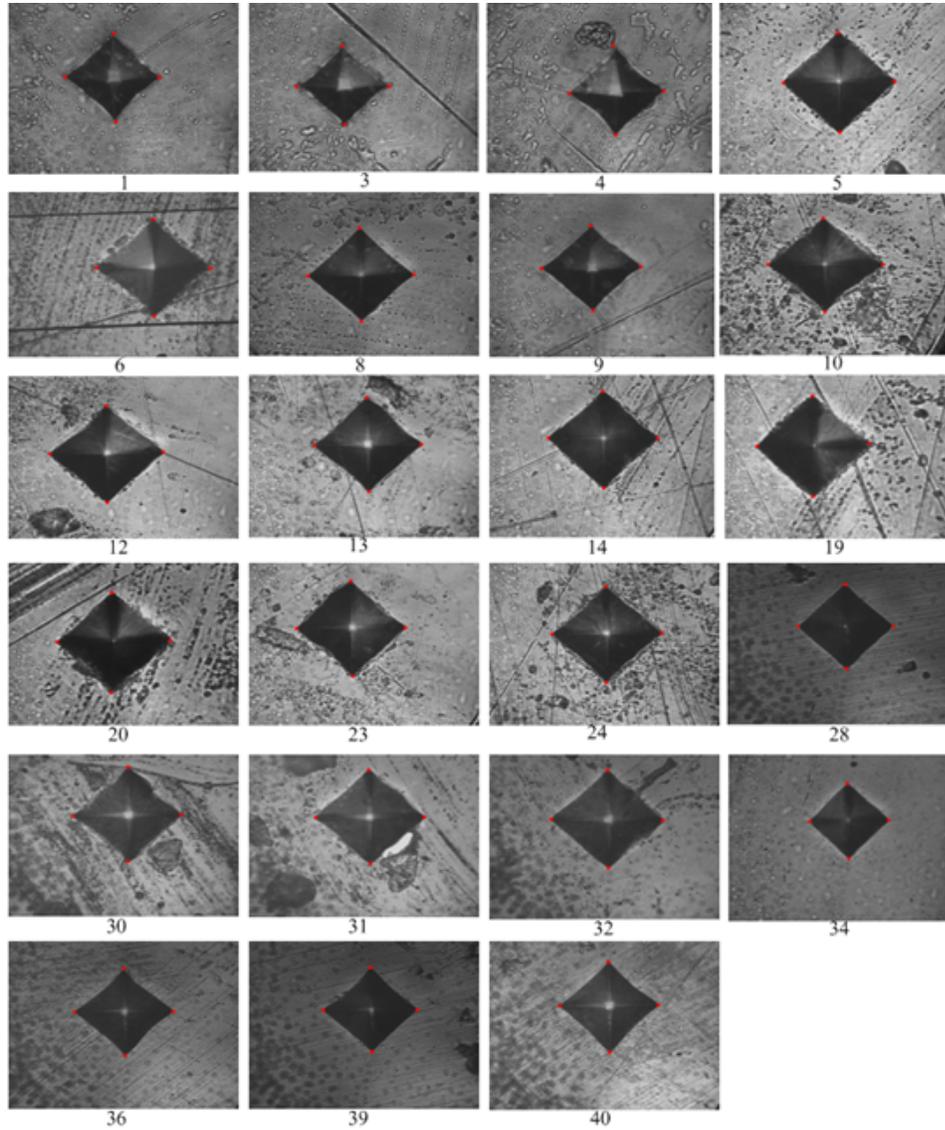


Figura 6.17: Grupo 4: imágenes con error de longitud promedio de diagonales menor a 1%.

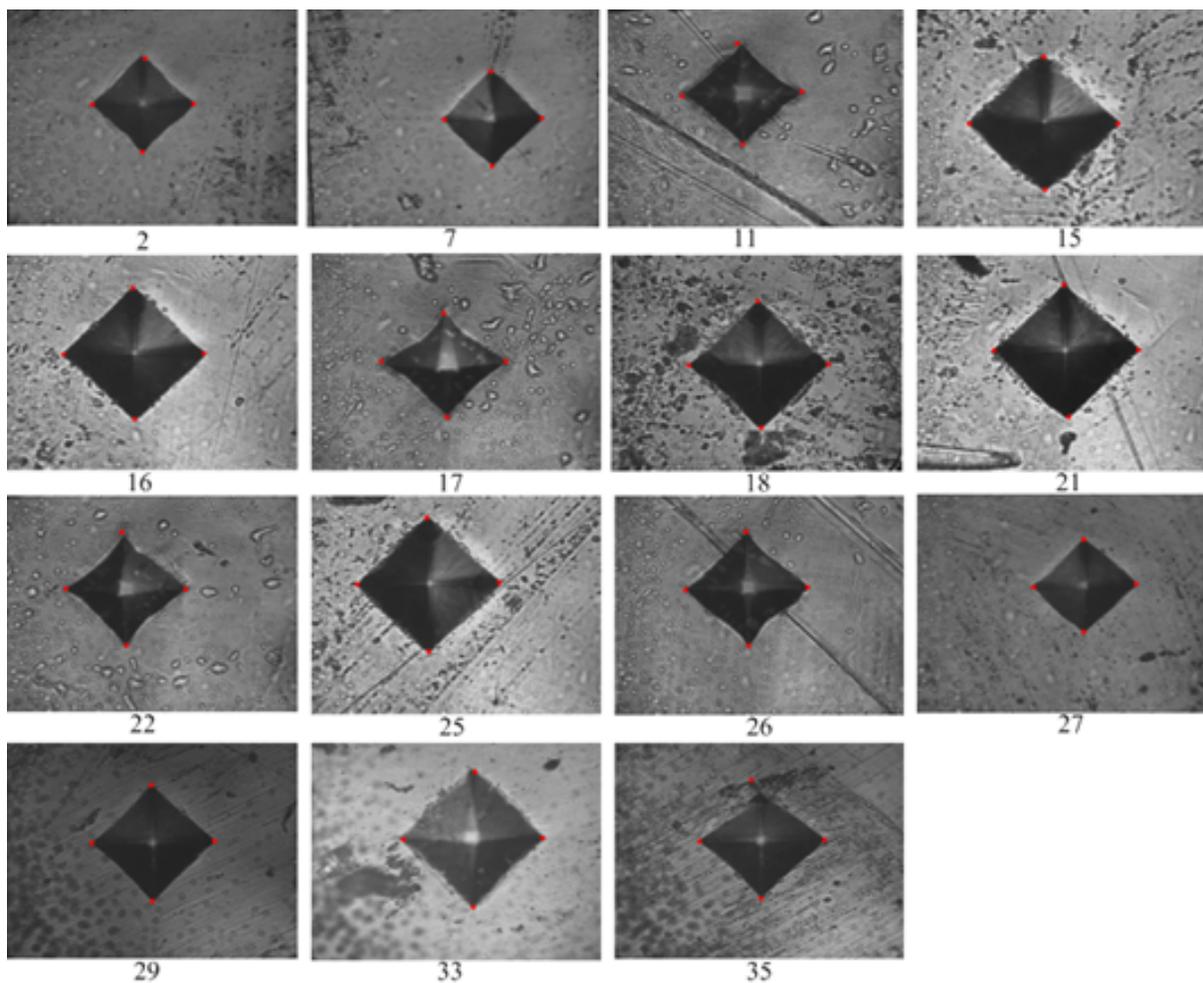


Figura 6.18: Grupo 4: imágenes con error de longitud promedio de diagonales entre 1% y 3%, excepto de la imagen 35 cuyo error está entre 3% y 5%.

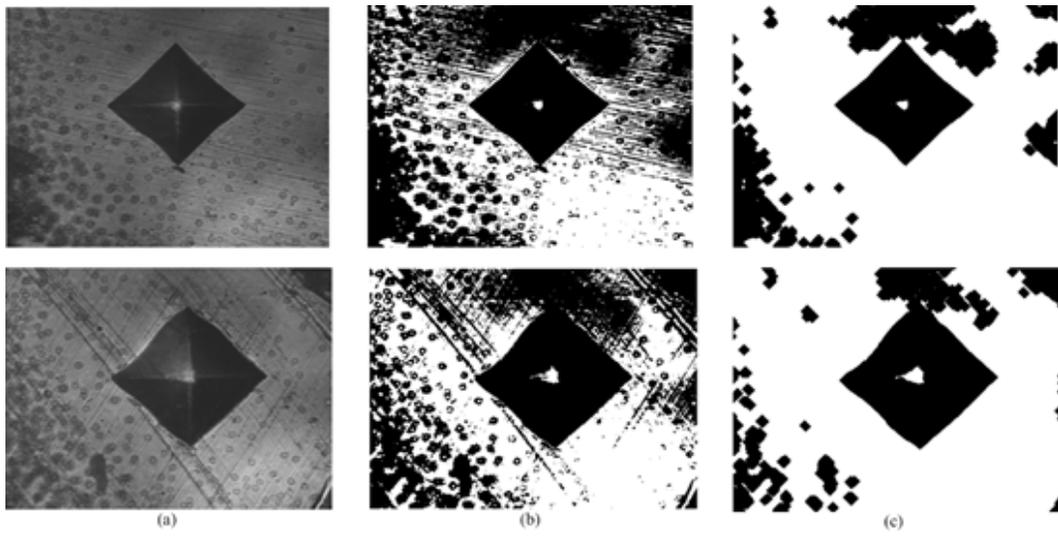


Figura 6.19: Grupo 4: dos imágenes fuera del alcance del algoritmo. (a) imagen original. (b) resultado de binarización. (c) respuesta del filtro morfológico.

Conclusiones y trabajo futuro

En esta tesis, se presentó un sistema de adquisición de imagen que trabaja en tiempo real con alta velocidad de captura y procesamiento, el cual fue desarrollado en un FPGA, adquiriendo las imágenes provenientes de una cámara CCD implementada en el microdurómetro Mitutoyo modelo HM-125.

La alta velocidad de los dispositivos FPGA ofrece una solución flexible para su implementación en un amplio rango de aplicaciones. La capacidad de reconfiguración parcial de estos dispositivos permite su utilización de manera eficiente. Una importante característica en cómputo reconfigurable como en el caso de los FPGA, es que permite el intercambio rápido de módulos embebidos o no en el FPGA, sin la necesidad de reiniciar el dispositivo por completo para una total reconfiguración. Comparado con una PC general usado como sistema de adquisición de imagen, nuestro sistema tiene distintas características como: dimensión pequeña, bajo costo, bajo consumo de potencia de orden de varios miliWatts, alta integración y alta velocidad para alcanzar el rango de ser un sistema de adquisición de imagen en tiempo real. Nuestro sistema realiza adquisición de imagen, despliega en pantalla las imágenes adquiridas y almacena las imágenes en formato comercial (bmp y png).

En este trabajo, se desarrolló una interfaz gráfica, el cual ajusta parámetros de filtros de respuesta al impulso embebidos en el FPGA, tal que permitirá eliminar ruido eléctrico y magnético que perturba a la etapa de salida del circuito electrónico de la CCD implementada en el microdurómetro. Este sistema de adquisición de imagen basado en un FPGA, es simple, fácil de usar, y es apropiado para interactuar en el control de otros parámetros de la CCD.

Esta tesis propone un nuevo algoritmo para el procesamiento y análisis de imágenes de indentación Vickers, el cual obtiene resultados competitivos de exactitud y robustez contra ruido en la imagen e imperfección en la superficie. Las novedades de nuestro algoritmo consisten en una simple binarización automática, filtro mor-

fológico por un elemento estructurante el cual nunca se había usado con anterioridad en el análisis de imágenes con indentaciones, y la aplicación de un detector de esquinas [28] para encontrar los vértices de la indentación, el cual tampoco se había utilizado antes para este tipo de imágenes.

Trabajo futuro:

Como trabajo futuro se pretende lograr un método completamente automático que incluya la decisión y aplicación de técnicas de preprocesamiento, por ejemplo, resolviendo problemas de bajo contraste entre el fondo de la imagen e indentación. Otro desafío es integrar el método en el ámbito de fabricación industrial donde cada imagen con indentación sea adquirida en tiempo real y procesada de manera inmediata.

Bibliografía

- [1] “Altera DE2-115 Development and Education Board”, Terasic Inc., Altera Corporation, WebPage (accessed October 2018): <https://www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English& No=502>, 2007.
- [2] T. Andersson, U. Stigh, “The stress-elongation relation for an adhesive layer loaded in peel using equilibrium of energetic forces”, *Int. Journal of Solids and Structures*, 41(2): 413-434, 2004.
- [3] E. Atar, C. Sarioglu, U. Demirler, E.S. Kayali, H. Cimenoglu, “Residual stress estimation of ceramic thin films by X-ray diffraction and indentation techniques”, *Scripta Materialia*, 48(9): 1331-1336, 2003.
- [4] P.J. Blau, B.R. Lawn, “Microindentation Techniques in Materials Science and Engineering: A Symposium Sponsored by ASTM Committee E-4 on Metallography and by the Int. Metallographic Society”, Philadelphia, PA, USA, July 1984.
- [5] S.I. Bulychev, V.P. Alekhin, M. Kh. Shorshorov, A.P. Ternorskii, “Determining Young’s modulus from the indenter penetration diagram”, *Zavod. Lab.* 419, p. 11137-11140, 1975.
- [6] L. Bureš, M. Hruží, L. Müller, D. Mañas, M. Mañas, “Automatic Optical Detection of Indentation into Polymers for Inner Stress Analysis”, *Proc. of Int. Conf. on Mining, Material and Metallurgical Engineering*, (Prague, Czech Republic, Aug. 2014), Paper No. 74, 6 pages, 2014.
- [7] C. Burrus, “Digital filter structures described by distributed arithmetic”, *IEEE Transactions on Circuits and Systems*, 24(12): 674-680, 1977.
- [8] Chao Yang y otros, “Bulk TiB₂-based ceramic composites with improved mechanical property using Fe-Ni-Ti-Al as a sintering aid”, *Materials*, 7(10): 7105-7117, 2014.

- [9] Chi-Tsong Chen, “Analog and Digital Control System Design: Transfer-Function, State-Space, and Algebraic Methods”, Oxford University Press, 2006.
- [10] B.N. Coelho, A. Guarda, G.L. Faria, D. Menotti, “Automatic Vickers microhardness measurement based on image analysis”, Proc. 19th Int. Conf. on Image Processing, Computer Vision and Pattern Recognition (IPCV15, Las Vegas, Nevada, USA, July 2015), 249-255, WorldComp, 2015.
- [11] P.M. Daniel, D. Whitteridge, “The representation of the visual field on the cerebral cortex in monkeys”, Journal of Physiology, 159: 203-221, 1961.
- [12] Data Bank of Vickers indentation images with expert annotations, images acquired by the microdurometer hardness testing machine Mitutoyo HM-124, <http://www.ctrl.cinvestav.mx/~biene/Indentations.htm> (accessed October 2018).
- [13] S.M. Domínguez-Nicolás, R. Juárez-Aguirre, P.J. García-Ramírez, A.L. Herrera-May, “Signal conditioning system with a 420 mA output for a resonant magnetic field sensor based on MEMS technology”, IEEE Sensors Journal, 12(5): 935-942, May 2012.
- [14] S.M. Domínguez-Nicolás, R. Juárez-Aguirre, A.L. Herrera-May, P.J. García-Ramírez, E. Figueras, E.A. Gutierrez-D., J.A. Tapia, A. Trejo, E. Manjarrez, “Respiratory magnetogram detected with a MEMS device”, Int. Journal of Medical Sciences 10(11): 1445-1450, 2013.
- [15] S.M. Domínguez-Nicolás, P. Wiederhold, “A real-time image acquisition system based on FPGA for hardness tests by indentation”, Proc. Int. Conf. on Computer Science and Artificial Intelligence (ICCSAI 2014, Wuhan, China, Dec. 2014), Paper ID ICCSAI-052, 7 pages, 2014.
- [16] S.M. Domínguez-Nicolás, P. Argüelles-Lucho, P. Wiederhold, “FPGA based image acquisition and graphic interface for hardness tests by indentation”, Int. Journal of Advanced Computer Technology, 6(6):6-16, 2016.
- [17] S.M. Domínguez-Nicolás, P. Wiederhold, “Indentation image analysis for Vickers hardness testing”, 15th Int. Conf. on Electrical Engineering, Computing Science, and Automatic Control (IEEE - CCE), Mexico City. Sept. 5-7, 2018, 6 pages, in press, 2018.
- [18] M. Donadio, “Lost knowledge refund: sharpened FIR filters. IEEE Signal Processing Magazine, 20(5): 61-63, 2003.

- [19] E.R. Dougherty, “An Introduction to Morphological Image Processing”, SPIE Press, USA, 1992.
- [20] A. Downton, D. Crookes, “Parallel architectures for image processing”, *Electronics and Communication Engineering Journal*, 10(3): 139 - 151, 1989.
- [21] P.A. Engel, A.R. Chitzas, E.Y. Hsue, “Interpretation of superficial hardness for multilayer platings”, *Thin Solid Films*, 207(1-2): 144-152, 1992.
- [22] P.P.R. Filho, T.S. Cavalcante, V.H.C. Albuquerque, J.M.R.S. Tavares, “Brinell and Vickers hardness measurement using image processing and analysis techniques”, *Journal of Testing and Evaluation*, 38(1):1-19, JTE102220, 2010.
- [23] M. Gadermayr, A. Maier, A. Uhl, “Algorithms for microindentation measurement in automated Vickers hardness testing”, *SPIE Proc. of 10th Int. Conf. on Quality Control for Artificial Vision (QCAV’11, St. Etienne, France)*, 8000: 8000M-1–8000M-10, 2011.
- [24] M. Gadermayr, A. Maier, A. Uhl, “Robust algorithm for automated microindentation measurement in Vickers hardness testing”, *SPIE and IS-T Journal of Electronic Imaging*, 21(2): 021109-1–021109-12, 2012.
- [25] M. Gadermayr, A. Uhl, “Dual-resolution active contours segmentation of Vickers indentation images with shape prior initialisation”, *Proc. 5th Int. Conf. on Image and Signal Processing (ICISP12, Agadir, Morocco, June 2012)*, *Lecture Notes on Computer Science* vol. 7340, 362-369, Springer, 2012.
- [26] J. Hao, G. Li, “Optimal ρ -realisation for digital controller implementation with stability consideration”, *Electronics Letters*, 40(4): 226-227, 2004.
- [27] D.S. Harding, W.C. Oliver, G.M. Pharr, “Cracking during nanoindentation and its use in the measurement of fracture toughness”, *Symposium B2 - Thin Films: Stresses and Mechanical Properties V*, 356(2): 663-669, 1994.
- [28] C. Harris, M. Stephens, “A combined corner and edge detector”, *Proc. 4th Alvey Vision Conference (August 1988)*, p. 147-151, 1988.
- [29] G.E. Healey, R. Kondepudy, “Radiometric CCD camera calibration and noise estimation”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(3): 267-276, 1994.

- [30] A.L. Herrera-May, J.A. Tapia, S.M. Domínguez-Nicolás, R. Juárez-Aguirre, E.A. Gutierrez-D., A. Flores, E. Figueras, E. Manjarrez, “Improved detection of magnetic signals by a MEMS sensor using stochastic resonance”, PLOS ONE, 9(10), e109534, 8 pages, Oct. 2014.
- [31] E.C. Ifeachor, B.W. Jervis, “Digital Signal Processing - a Practical Approach”, (1st Ed. 1993 by Addison Wesley), Second Edition, Pearson Education Limited, Prentice Hall, USA, Oct. 2001.
- [32] A.K. Jain, “Fundamentals of Digital Image Processing”, Prentice Hall, Englewood Cliffs, NJ, USA, 1989.
- [33] Y. Ji, A. Xu, “A new method for automatically measurement of Vickers hardness using thick line Hough transform and least square method”, Proc. 2nd Int. IEEE Conf. on Image and Signal Processing (CISP’09, New York, USA), p. 1-4, 2009.
- [34] B. Jönsson, S. Hogmark, “Hardness measurements of thin films”, Thin Solid Films, 114(3): 257-259, 1984.
- [35] R. Juárez-Aguirre, S.M. Domínguez-Nicolás, E. Manjarrez, J.A. Tapia, E. Figueras, H. Vázquez-Leal, L.A. Aguilera-Cortéz, A.L. Herrera-May, “Digital signal processing by virtual instrumentation of a MEMS magnetic field sensor for biomedical applications”, Sensors 13(11): 15068-15084, 2013.
- [36] J. Keith, “Video Demystified - A Handbook for the Digital Engineer”, 5th Edition, Newnes, Oxford, UK, 2007.
- [37] B. Kern, “Charge-injection noise in CCDs”, IEEE Transactions on Electron Devices, 53(10): 2478-2484, 2006.
- [38] N. Lall, B. Taylor, B., “Designing control circuits for FPGA-based DSP systems”, EDN Network, Aug. 2006.
- [39] J. Li, H. Liu, K. Althoefer, L.D. Seneviratne, “A stiffness probe based on force and vision sensing for soft tissue diagnosis”, Annual Int. Conf. of the IEEE Engineering in Medicine and Biology Society (San Diego, CA, USA, Sept. 2012), 2012.
- [40] J.L. Loubet, J.M. Georges, O. Marchesini, G. Meille, “Vickers indentation curves of magnesium oxide (MgO). Journal of Tribology, 106(1): 43-48, 1984.
- [41] M. Macedo, V.B. Mendes, A. Conci, F.R. Leta, “Using Hough transform as an auxiliary technique for Vickers hardness measurement”, Proc. 13th Int. Conf. on Systems, Signals and Image Processing (IWSSIP 2006, Kosice, Slovakia), p. 287-290, 2006.

- [42] A. Maier, G. Niederbrucker, S. Stenger, A. Uhl, “Efficient focus assessment for a computer vision-based Vickers hardness measurement system”, SPIE and IS-T Journal of Electronic Imaging, vol. 21(2), p. 021114-1–021114-14, 2012.
- [43] A. Maier, A. Uhl, “Robust automatic indentation localisation and size approximation for Vickers microindentation hardness indentations”, Proc. 7th Int. Symposium on Image and Signal Processing (ISPA 2011, Dubrovnik, Croatia), p. 295-300, IEEE Press, New York, USA, 2011.
- [44] A. Maier, A. Uhl, “The areamap operator and its application to Vickers hardness testing images”, Int. Journal of Future Generation Communication and Networking, Special Issue on Image and Signal Processing (papers of ICISP 2012), 5(4): 1-16, 2012.
- [45] A. Maier, A. Uhl, “Areamap and Gabor filter based Vickers hardness indentation measurement”, Proc. Signal Processing Conference EUSIPCO, p. 1-5, 2013.
- [46] Y. Matsunaga, S. Oosawa y otros, “A high sensitivity output amplifier for CCD image sensor”, Int. Electron Devices Meeting (Washington DC, USA, Dec. 1987), 1987.
- [47] V.B. Mendes, F.R. Leta, “Automatic measurement of Brinell and Vickers hardness using computer vision techniques”, Proc. XVII IMEKO World Congress Metrology in the 3rd Millennium (Dubrovnik, Croatia, June 2003), p. 992-995, 2003.
- [48] Mitutoyo (American Corporation), “Micro hardness testing machines, Hardness testing equipment, Product Information for Model HM-112/114/122/124”, Illinois, USA, <http://www.mitutoyo.co.jp/global.html>, 2013.
- [49] W.D. Nix, “Mechanical properties of thin films”, Metallurgical Transactions A, 20A(1): 2217-2245, 1989.
- [50] R. Nowak, C.L. Li, S. Maruno, “Low-Load indentation behavior of HfN thin films deposited by reactive rf sputtering”, Journal of Materials Research, 12(1): 64-69, 1997.
- [51] W.C. Oliver, G.M. Pharr, “An improved technique for determining hardness and elastic modulus using load and displacement sensing indentation experiments”, Journal of Materials Research, 7(6): 1564-1583, 1992.
- [52] Yong Pan y otros, “A new method for automatically measuring Vickers hardness based on region-point detection algorithm”, SPIE Proc. 4th Int. Symposium on Pre-

- cision Mechanical Measurements (Anhui, China, Dec. 2008), vol. 7130, p. 71304C-1–71304C-6, 2008.
- [53] D. Pellerin, “Send in the clown fish: implementing video analysis in Xilinx FPGAs”, Xcellence In Embedded Applications, http://www.impulseaccelerated.com/emb5_fastvideo_01.pdf (accessed October 2018), April 2009.
- [54] J.B. Pethica, R. Hutchings, W.C. Oliver, “Hardness measurement at penetration depths as small as 20nm”, *Philosophical Magazine A*, 48(4): 593-606, 2006.
- [55] A.G. Place, G.H. Allen, “Generalized pole sensitivity analysis due to parameter perturbation”, *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, 44(10): 869-873, 1997.
- [56] T. Raikovich, B. Fehér, “Application of partial reconfiguration of FPGAs in image processing”, *Proc. of IEEE 6th Conf. on Ph.D. Research in Microelectronics and Electronics* (Berlin, Germany, July 2010), Sept. 2010.
- [57] M.A. Rotea, D. Williamson, “Optimal realizations of finite wordlength digital filters and controllers”, *IEEE Transactions on Circuits and Systems I*, 42(2): 61-72, 1995.
- [58] N.A. Sakharova, J.V. Fernandes, J.M. Antunes, M.C. Oliveira, “Comparison between Berkovich, Vickers and conical indentation tests: A three-dimensional numerical simulation study”, *Int. Journal of Solids and Structures*, 46(5): 1095-1104, 2009.
- [59] E.L. Schwartz, “Spatial mapping in the primate sensory projection: Analytic structure and relevance to perception”, *Biological Cybernetics*, 25(4): 181-194, 1977.
- [60] F. Scotti, “Automatic morphological analysis for acute leukemia identification in peripheral blood microscope images”, *IEEE Int. Conf. on Computational Intelligence for Measurement Systems and Applications* (CIMSAs, Giardini Naxos, Italy, July 2005), p. 96-101, 2005.
- [61] Jiann-Shing Jeng, Chin-Hung Liu, J.S. Chen, “Effects of substrate bias and nitrogen flow ratio on the resistivity, composition, crystal structure, and reflectance of reactively sputtered hafnium-nitride film”, 486(1-2): 649-652, 2009.
- [62] R.E. Smith, G.E. Sandland, “An accurate method of determining the hardness testing of the materials with particular reference to those of high degree of hardness”, *Proc. of Inst. Mech. Eng.*, pt. 1, p. 623-641, 1922.

- [63] T. Sugimoto, T. Kawaguchi, "Development of an automatic Vickers hardness testing system using image processing technology", *IEEE Trans. Industrial Electronics*, 44(5): 696-702, 1997.
- [64] Sutherland HDL, www.sutherland-hdl.com (accessed October 2018), Inc. Home Page, 2001.
- [65] H.W. Thoo, M.M. Ratnam, "An improved method of projected area determination in nanoindentation using image processing with sub-pixel edge location", *Experimental Techniques*, Society for Experimental Mechanics (SEM), p. 1-15, 2014.
- [66] R.B. Tootell, M.S. Silverman, E. Swikes, R.L. DeValois, "Deoxyglucose analysis of retinotopic organization in primate striate cortex", *Science*, 218(4575): 902-904, 1982.
- [67] J.R. Tuck, A.M. Korsunsky, R.I. Davidson, S.J. Bull, D.M. Elliott, "Modelling of the hardness of electroplated nickel coatings on copper substrates", *Surface and Coatings Technology*, 127(1): 1-8, 2000.
- [68] G.F. Vander Voort, "Results of an ASTM E04 round robin on the precision and bias of measurements of microindentation hardness", in: *Factors that affect the precision of mechanical tests*, p. 3-39, ASTM, Philadelphia, PA, USA, 1989.
- [69] Y.M. Wang, Y. Li, J.B. Zheng, "A camera calibration technique based on OpenCV", 3rd Int. Conf. on Information Sciences and Interaction Sciences (Chengdu, China, June 2010), *IEEE Proc.*, 403-406, 2010.
- [70] Z. Wenge, H. Huiming, "FPGA-based video image processing system research", 3rd Int. Conf. on Information Sciences and Interaction Sciences (Chengdu, China, June 2010), *IEEE Proc.*, 680-682, 2010.
- [71] Q. Wu, F. Merchant, K. Castleman, "Microscope Image Processing", Elsevier and Academic Press, 2008.
- [72] C. Yang y otros, "Bulk TiB₂-based ceramic composites with improved mechanical property using Fe-Ni-Ti-Al as a sintering aid", *Materials (Basel)*, 7(10): 7105-7117, 2014.
- [73] L. Yao, C.H. Fang, "A hardness measuring method based on Hough fuzzy vertex detection algorithm", *IEEE Trans. Industrial Electronics*, 53(3): 950-962, 2006.

Anexos

Anexo A

Códigos de programas elaboradas en Verilog HDL.

Este Anexo inicia en la página 125 de la tesis, y tiene 8 páginas.

Anexo B

Tablas con los resultados de experimentos con 230 imágenes de indentación Vickers.

Este Anexo inicia en la página 133 de la tesis, tiene una hoja inicial y 23 páginas de tablas.

Anexo C

Página portada de cada una de las seis publicaciones reportadas como resultados de la tesis.

Este Anexo inicia en la página 159 de la tesis, y tiene 6 páginas.

Los anexos terminan con la página 164 de la tesis.

