



Universidad Autónoma de Querétaro  
Facultad de Ingeniería  
Maestría en Mecatrónica

PROCESADOR MORFOLÓGICO DE IMÁGENES EN FPGA  
APLICADO AL ANÁLISIS METALOGRAFICO

TESIS

Que como parte de los requisitos para obtener el grado de

Maestro en Ciencias

Presenta:

Marco Antonio González Aguirre

Dirigido por:

Dr. Luis Morales Velázquez  
Dr. Roque Alfredo Osornio Ríos

SINODALES

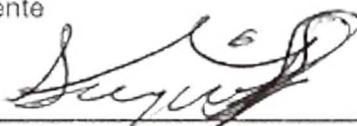
Dr. Luis Morales Velázquez  
Presidente

Dr. Roque Alfredo Osornio Ríos  
Secretario

Dr. René de Jesús Romero Troncoso  
Vocal

Dr. Luis Alberto Morales Hernández  
Suplente

Dr. Jesús Carlos Pedraza Ortega  
Suplente

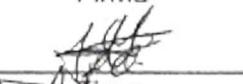
  
Dr. Aurelio Domínguez González  
Director de la Facultad

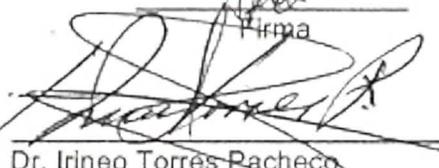
  
Firma

  
Firma

  
Firma

  
Firma

  
Firma

  
Dr. Irineo Torres Pacheco  
Director de Investigación y Posgrado

## RESUMEN

Para realizar el análisis metalográfico de fundiciones nodulares austemperizadas se han desarrollado técnicas para la completa caracterización de su microestructura mediante el procesamiento digital de imágenes basado en morfología matemática. Estas técnicas de caracterización se usan actualmente en el laboratorio de metalografía de la Universidad Autónoma de Querétaro, desarrolladas en software e implementadas en computadoras de propósito general, pero tienen el problema de requerir un tiempo muy alto para la ejecución de los algoritmos. En este trabajo se propone el diseño e implementación de un procesador morfológico de imágenes con la finalidad de reducir el tiempo de procesamiento de los algoritmos de análisis basados en morfología matemática de fundiciones nodulares en comparación con una PC. El procesador desarrollado tiene la característica de ser programable, reconfigurable y de arquitectura abierta. Además del procesador, en este trabajo se desarrollaron núcleos de propiedad intelectual para el despliegado de imágenes en estándar VGA, adquisición de imágenes de 640 por 480 píxeles en escala de grises por medio de un decodificador de video con estándar ITU-R BT.656 en formato NTSC, un gestor de memoria para imágenes de 640 por 480 píxeles y comunicación serial RS-232. Estos núcleos en conjunto con el procesador se describieron en lenguaje VHDL y se sintetizaron en un FPGA, dentro de una plataforma de desarrollo Altera DE2-70. Con esto se logró el desarrollo de un sistema de procesamiento morfológico de imágenes capaz de adquirir las imágenes directamente de una fuente de video compuesto, de desplegar imágenes en un monitor VGA, ya sea la imagen adquirida o procesada, y enviar cualquiera de éstas a una PC vía comunicación serial.

**(Palabras clave:** Procesador morfológico de imágenes, FPGA, sistemas embebidos)

## SUMMARY

To conduct metallographic analysis of austempered ductile iron castings techniques have been developed for the complete characterization of its microstructure by digital image processing based on mathematical morphology. These characterization techniques are currently used in the metallography laboratory of the Autonomous University of Querétaro, developed and implemented in software on general-purpose computers, but have the problem that algorithms require a long execution time. This paper proposes the design and implementation of a morphological image processor in order to reduce the processing time of algorithms based on mathematical morphology analysis of austempered ductile iron castings compared to a PC. The developed processor has the characteristic of being programmable, reconfigurable and with open architecture. Besides the processor, this paper developed IP cores for the display of images in VGA standard, image acquisition of 640 by 480 pixels grayscale images using a standard video decoder with ITU-R BT.656 in NTSC format, a memory manager for images of 640 by 480 pixels and RS-232 serial communication. These IP cores in conjunction with the processor were described in VHDL and synthesized in an FPGA, within a development platform Altera DE2-70. This procure the realization of a morphological processing system capable of acquiring images directly from a composite video source, to display images on a VGA monitor, either the acquired or processed image, and send to any of these to a PC via serial communication.

**(Key words:** Morphological image processor, FPGA, embedded systems)

**A mi familia con cariño y gratitud.**

## **AGRADECIMIENTOS**

A mi familia y seres queridos por su apoyo y comprensión. A mi asesor Dr. Luis Morales Velázquez por su invaluable apoyo en la culminación de este trabajo. A mi coasesor Dr. Roque Alfredo Osornio Ríos por su insistencia y motivación. Al Dr. Luis Alberto Morales Hernández por su tiempo y asesoría. Al Dr. René de Jesús Romero Troncoso y compañeros del grupo HSPdigital por todo su apoyo. Al Consejo Nacional de Ciencia y Tecnología (CONACYT) por la beca otorgada para mis estudios. A mis profesores, personal administrativo y compañeros de la Universidad Autónoma de Querétaro. Al Dr Jesús Carlos Pedraza Ortega por su guía e inspiración en mi formación académica.

# Índice general

Resumen	I
Summary	II
Dedicatoria	III
Agradecimientos	IV
Índice general	V
Índice de figuras	VII
Índice de tablas	IX
<b>I. Introducción</b>	<b>1</b>
1.1. Antecedentes	2
1.2. Justificación	4
1.3. Descripción del problema	5
1.4. Hipótesis y objetivos	6
1.4.1. Hipótesis	6
1.4.2. Objetivo general	6
1.4.3. Objetivos particulares	6
<b>II. Fundamentación teórica</b>	<b>8</b>
2.1. Metalografía	8
2.2. Procesamiento digital de imágenes	10
2.3. Tecnología FPGA	15

<b>III. Metodología</b>	17
3.1. Desplegado en VGA	18
3.2. Adquisición de imágenes	21
3.3. Manejo de memoria	27
3.4. Unidad de procesamiento	31
3.4.1. MCU	35
3.5. Integración del sistema	38
3.6. Implementación de algoritmo	40
<b>IV. Resultados y discusión</b>	42
4.1. Resultados	42
4.2. Conclusiones	47
4.3. Trabajo a futuro	47
Referencias	49
A. Artículo Publicado	52

# Índice de figuras

<b>Figura</b>		<b>Página</b>
2.1	Representación de las coordenadas de una imagen (Gonzalez y Woods, 2002).	11
2.2	Operación algebraica lineal entre imágenes (Myler y Weeks, 1993).	12
2.3	Ejemplo de operaciones lógicas entre imágenes binarias.	13
2.4	Ejemplo de dilatación: (a) Elemento estructural. (b) Imagen X. (c) Resultado de la dilatación.	14
2.5	Ejemplo de erosión: (a) Elemento estructural. (b) Imagen X. (c) Resultado de la erosión.	14
3.1	Diagrama a bloques del sistema en general.	18
3.2	Diagrama del IP core “VGA CONTROLLER”.	19
3.3	DAC ADV7123 de Analog Devices.	20
3.4	Diagrama de tiempo de la sincronía horizontal en VGA.	21
3.5	Diagrama del IP core de decodificación de video “VIDEO-IN CONTROLLER”.	22
3.6	Decodificador de video ADV7180 de Analog Devices.	23
3.7	Composición del tren de datos de la interfaz ITU-R BT.656.	24
3.8	Separación de una imagen en componentes YCbCr.	25
3.9	Codificación YCbCr 4:2:2 y 4:4:4.	25
3.10	Diagrama a bloques del IP core “VIDEO-IN CONTROLLER”.	26

<b>Figura</b>	<b>Página</b>
3.11	Diagrama de tiempo de sincronía horizontal. 27
3.12	Diagrama a bloques del IP core “MMU”. 28
3.13	Diagrama del bloque MMUs. 29
3.14	Diagrama detallado del bloque MMUs. 30
3.15	Diagrama a bloques de operación de binarizado. 32
3.16	Diagrama a bloques de operaciones lógicas. 33
3.17	Diagrama a bloques de operaciones morfológicas. 34
3.18	Codificación de la instrucción. 35
3.19	Diagrama a bloques de la interconexión del sistema de adquisición, almacenaje y desplegado. 38
3.20	Diagrama de interconexión de los IP core “VGA CONTROLLER”, “VIDEO-IN CONTROLLER” Y “MMU”. 39
4.1	Sistema en funcionamiento 43
4.2	Imagen de prueba 44
4.3	Imagen de prueba binarizada 45
4.4	Apertura 30 45
4.5	Apertura 21 46
4.6	Diferencia apertura 30 y 21 46

# Índice de tablas

<b>Tabla</b>		<b>Página</b>
2.1	Operaciones lógicas	12
3.1	Juego de instrucciones del procesador desarrollado.	36

# Capítulo 1

## Introducción

El estudio de las características micro estructurales de un material permite predecir el comportamiento de éste, debido a la relación que existe entre tales características y las propiedades mecánicas del material. En el caso de metales y aleaciones, la metalografía es la disciplina que permite realizar la caracterización de éstos, a través del análisis de una muestra representativa del material mediante el uso de microscopios para comparar y cuantificar los patrones micro estructurales, relaciones espaciales y de forma, con la finalidad de obtener datos numéricos. Aún cuando las características mencionadas anteriormente son fácilmente reconocibles en una imagen, la obtención de datos numéricos confiables de forma manual es muy difícil debido a que dependen del evaluador.

Los desarrollos en hardware y software computacional de los últimos años han hecho posible un cambio significativo en los métodos de caracterización de materiales, permitiendo la extracción cuantitativa de información de forma automática mediante la adquisición, procesamiento y análisis de imágenes en forma digital (ASM Handbook, 2004), ya sea con la digitalización de una fotografía metalográfica o con la adquisición directa de una cámara en el microscopio para su posterior procesamiento y análisis en una PC o en una plataforma

especializada, eliminando de esta manera los errores que pueden ser introducidos por el evaluador. Una de las técnicas de procesamiento de imágenes usada en el análisis metalográfico es la morfología matemática, la cual se basa en el estudio de la forma y se usa para investigar la relación que existe entre una imagen y un elemento estructurante.

### **1.1. Antecedentes**

Dentro de la Universidad Autónoma de Querétaro se han realizado distintos trabajos relacionados con el procesamiento digital de imágenes, como es el caso de Méndez (2008) que usando técnicas de procesamiento digital de imágenes logró la detección automática de unidades formadoras de colonias, Vargas (2000) logró la detección de cisuras en imágenes del cerebro mediante filtros morfológicos, Benítez (2005) desarrolló una herramienta de software para la medición de dimensiones por visión fuera de línea haciendo uso de filtros morfológicos en el pre procesamiento, Morales (2005) propuso la caracterización de huellas digitales a partir de una familia de filtros morfológicos direccionales (particularmente aperturas y cerraduras direccionales). En el caso particular del análisis metalográfico, Morales et al. (2010) lograron la caracterización de nódulos de grafito por medio del estudio de la circularidad de los nódulos y su compacidad, haciendo uso de la técnica de procesamiento de imágenes basada en morfología matemática.

Cabe mencionar que todos los trabajos mencionados anteriormente fueron implementados en software, haciendo uso de una computadora personal (PC, *Personal Computer*) para el procesamiento de los algoritmos y que actualmente en la Universidad Autónoma de Querétaro no se cuenta con una plataforma propia que integre hardware para el procesamiento morfológico de imágenes. El desarrollo de plataformas basadas en arreglos de compuertas programables en campo (FPGA, *Field Programmable Gate Array*) en la universidad se ha enfocado

principalmente al control y monitoreo de maquinaria, como el trabajo de Morales (2010) en el que se desarrolló una plataforma para aplicaciones industriales y en la cual se integraron una gran cantidad de módulos de propiedad intelectual que comprenden desde módulos de comunicación hasta un microprocesador.

A nivel nacional se encuentra trabajos en los que se tienen plataformas de procesamiento de imágenes que integran equipos de hardware, como el caso de Rangel (2005) donde se realizó la descripción en lenguaje de descripción de hardware de muy alta velocidad (VHDL, *Very high speed Hardware Description Language*) y la implementación en FPGA de algoritmos de morfología matemática para el procesamiento de imágenes binarias, así como también la implementación de controladores para la adquisición y binarización de imágenes directamente de una cámara y el despliegado de imágenes en un monitor VGA (sistema gráfico de pantallas para PC); con lo que se creó un sistema de procesamiento morfológico de imágenes rápido y que puede operar sin el uso de una PC, con el inconveniente de que no puede ser programado, por lo que cualquier cambio requiere reconfiguración. Por otra parte Razo (2006) desarrolló un procesador morfológico de imágenes basado en FPGA que permite adquirir imágenes a través de una cámara digital OV7648, el despliegado de imágenes en monitor VGA, seleccionar el tipo de filtro a utilizar y seleccionar el nivel de procesamiento morfológico, aumentando la versatilidad del sistema, pero con el inconveniente de requerir el uso de una PC para el envío de estos comandos, además de estar restringida a un sistema de desarrollo y a una cámara en particular.

En el ámbito internacional, podemos encontrar trabajos que hacen uso del alto paralelismo con el que cuentan los FPGA (Romero, 2007) como es el caso de Krill et al. (2010) que diseñaron un núcleo de propiedad intelectual (IP core, *Intellectual Property core*) para la transformación bidireccional de manera eficiente de los espacios de color YCrCb y RGB o Battlle et al. (2002) que desarrollaron una arquitectura paralela altamente versátil que permite el manejo de rutinas de procesamiento de imagen de alto nivel en tiempo real. Por otra parte, Shao y Liang

(2008) propusieron la arquitectura de un acelerador en hardware para la segmentación de objetos en video, a través del desarrollo de un procesador morfológico de imágenes reconfigurable, donde se propone un set de instrucciones que controlan la operación y conectividad de una serie de elementos de procesamiento que en conjunto conforman este procesador.

De manera comercial existen plataformas de desarrollo que integran elementos de software y equipos hardware para el procesamiento digital de imágenes, como es el caso de Labview, que permite desarrollar aplicaciones de forma sencilla usando bloques, sin embargo, el costo de estos equipos es muy alto, son poco configurables y de arquitectura cerrada.

En base a los antecedentes reportados, se puede destacar la necesidad de desarrollar una plataforma de procesamiento morfológico de imágenes que sea abierta, de bajo costo, que pueda ser reprogramada y reconfigurada, que haga uso del paralelismo de la tecnología FPGA para reducir los tiempos de procesamiento, que sea capaz de adquirir imágenes directamente de cualquier cámara con formato de video compuesto (CVBS, *Composite Video, Blanking and Sync*) y que pueda desplegar imágenes en cualquier monitor VGA.

## **1.2. Justificación**

Conocer las propiedades mecánicas de un material es de suma importancia, ya que estas determinarán ciertas características de los productos elaborados con éste. En el caso de las fundiciones nodulares austemperizadas, el análisis metalográfico por medio del procesamiento digital de imágenes basado en algoritmos de morfología matemática permite conocer sus propiedades mecánicas, ya que estas propiedades están relacionadas con la forma y tamaño de los nódulos de éstas.

En la Universidad Autónoma de Querétaro se realizan este tipo de análisis, pero implementados en PC, la cual al ser una plataforma de propósito general requiere una gran cantidad de tiempo para procesar los algoritmos necesarios. Debido a esto, en este trabajo se propone el desarrollo de un procesador de imágenes basado en tecnología FPGA, ya que de esta manera se pretende desarrollar un procesador programable y reconfigurable, que pueda reducir el tiempo de procesamiento, manteniendo hasta cierto punto la versatilidad que se tiene al usar una PC en la aplicación de algoritmos de morfología matemática aplicados al análisis de imágenes metalográficas.

### **1.3. Descripción del problema**

Actualmente existe una gran variedad de soluciones comerciales para realizar procesamiento digital de imágenes. Estas soluciones las encontramos tanto en software como en hardware o como una mezcla de ambas. Las soluciones basadas en software son las más versátiles, ya que permiten implementar una gran cantidad de algoritmos de procesamiento digital de imágenes de forma rápida y hacer modificaciones fácilmente, pero tienen el inconveniente de que al usar un hardware de propósito general como la PC, ciertos algoritmos como los usados en el procesamiento morfológico de imágenes requieren una gran cantidad de tiempo para ejecutarse. Por otra parte, las soluciones basadas en hardware requieren de un tiempo menor para realizar estos algoritmos, pero tienen el inconveniente de que son desarrollados para realizar algoritmos específicos previamente definidos, por lo que no se pueden modificar. Las soluciones basadas en software y hardware brindan la opción de tener un hardware optimizado para el procesamiento de imágenes, pero con la opción de que este puede ser programado para realizar distintas operaciones, por lo que se obtiene velocidad y versatilidad, pero con el inconveniente que las soluciones comerciales de este tipo tienen un costo elevado y una arquitectura cerrada.

A través del desarrollo de un procesador de imágenes propio basado en FPGA, es posible obtener una solución basada en software y hardware abierta, que sea de bajo costo y permita al laboratorio de metalografía de la Universidad Autónoma de Querétaro reducir los tiempos de procesamiento de los análisis metalográficos, teniendo la flexibilidad de ser reconfigurado y adaptado a otras aplicaciones de procesamiento de imágenes requeridas en el laboratorio de metalografía o en la universidad.

## **1.4. Hipótesis y objetivos**

### **1.4.1. Hipótesis**

Es posible reducir el tiempo de procesamiento de imágenes binarias de 640 x 480 píxeles, en el análisis metalográfico basado en morfología matemática mediante el uso de tecnología FPGA.

### **1.4.2. Objetivo general**

Desarrollar un procesador de imágenes basado en FPGA, para ejecutar algoritmos de morfología matemática, aplicado al análisis metalográfico de fundiciones nodulares austemperizadas.

### **1.4.3. Objetivos particulares**

- Desarrollar un IP core para despliegado de imágenes en estándar VGA.
- Desarrollar un IP core para la adquisición de imágenes por medio de un decodificador de video compuesto con estándar ITU-R BT.656.

- Diseñar e implementar un IP core para gestionar la memoria de imágenes de 640 x 480 píxeles.
- Diseñar e implementar la arquitectura del procesador y los operadores de procesamiento digital de imágenes: suma, resta, and, not, or, xor, erosión y dilatación.
- Implementar un algoritmo de análisis metalográfico basado en el procesador desarrollado y comparar el tiempo de procesamiento con respecto a una PC.

## Capítulo 2

### Fundamentación teórica

El desarrollo de un procesador como el que se propone en este trabajo integra elementos de diversas áreas de la ingeniería, entre los que se encuentran la metalografía, el procesamiento digital de imágenes, arquitectura de microprocesadores y tecnología FPGA. En este capítulo se da un breve panorama de los conceptos de cada una de éstas áreas abordados por este trabajo.

#### 2.1 Metalografía

La metalografía es la disciplina científica que examina y determina las características estructurales o constitutivas de los metales y aleaciones. El procedimiento para llevar a cabo el análisis metalográfico es el siguiente (Villela, 2004; ASM Handbook, 2004):

- a) *Selección de muestras*: Consiste en la selección de muestras representativas y de tamaño conveniente de la pieza a ser analizada.

- b) *Corte del material*: En este paso el material debe ser cortado en la orientación que el estudio requiera, ya que dependiendo de esta estructura se observará de distinta forma. El método a usar será aquel que minimice la deformación y el calentamiento en el área de corte, de acuerdo al material usado, ya que en ambos casos se puede afectar a la superficie a ser examinada.
- c) *Esmerilado*: En este paso se remueven las imperfecciones sobre el corte con el fin de tener una superficie plana. Para ello se usa una serie de lijas, pasando de la lija más gruesa a la más delgada. En todo el proceso se usa refrigerante para evitar cambios en la estructura del material debido al calor generado por la fricción.
- d) *Ataque*: Consiste en el uso de químicos sobre la pieza para lograr una corrosión controlada. El químico a usar depende del material de la pieza.
- e) *Microscopio*: El paso final es ver la muestra al microscopio, interpretar lo que se observa, comparar y documentar los resultados.

En este trabajo se enfoca en el análisis metalográfico de una *fundición nodular austemperizada*, comúnmente llamada ADI (*Austempered Ductile Iron*). Esta es una fundición de hierro con grafito libre en morfología esferoidal (también llamada fundición nodular o fundición dúctil). El control de la morfología del grafito y del tamaño de célula, unido a la posibilidad de adecuar la microestructura de la matriz mediante tratamientos térmicos (como el austemperizado) ha permitido a este tipo de fundiciones satisfacer los requerimientos de propiedades mecánicas en la industria, al mejorar la resistencia mecánica y mantener propiedades como la conductividad térmica, autolubricación y amortiguación de las vibraciones. Estas fundiciones se obtienen por medio de un tratamiento térmico de dos fases. Primeramente se aplica austenización a 900 °C seguida por un temple en el rango

de 230 °C a 400 °C. Dependiendo de los tiempos y temperatura en cada fase se modifican las propiedades de dureza y ductilidad (ASM Handbook, 1990).

## 2.2 Procesamiento digital de imágenes

Una imagen monocromática o simplemente *imagen* puede ser definida como una función bidimensional  $f(x, y)$  donde  $x$  y  $y$  son coordenadas espaciales y la amplitud de  $f$  en cualquier par de coordenadas  $(x, y)$  es llamada la intensidad de la imagen en ese punto. Cuando  $x, y$  y el valor de la amplitud de  $f$  son cantidades discretas finitas, llamamos a esa imagen una imagen digital. En cuanto al campo del procesamiento digital de imágenes, éste se refiere al procesamiento por medio de una computadora digital. Cabe notar que una imagen digital está compuesta por una cantidad finita de elementos, con una locación y valor particular. Estos elementos son llamados *elementos de imagen* o *pixeles*, siendo el término *pixel* el más comúnmente usado para denotar estos elementos de las imágenes digitales (Gonzalez y Woods, 2002).

Una imagen digital se puede representar por medio de una matriz numérica, donde el valor de cada pixel representa la intensidad luminosa discreta en el punto  $(x, y)$  (0=negro, 255= blanco para una codificación en 8 bits). En el caso de las imágenes que toman únicamente los valores blanco y negro se tiene 0=negro, 1=blanco y son llamadas imágenes binarias. El tamaño de la imagen puede ser de  $M$  renglones por  $N$  filas, y se puede representar de la siguiente forma (figura 2.1):

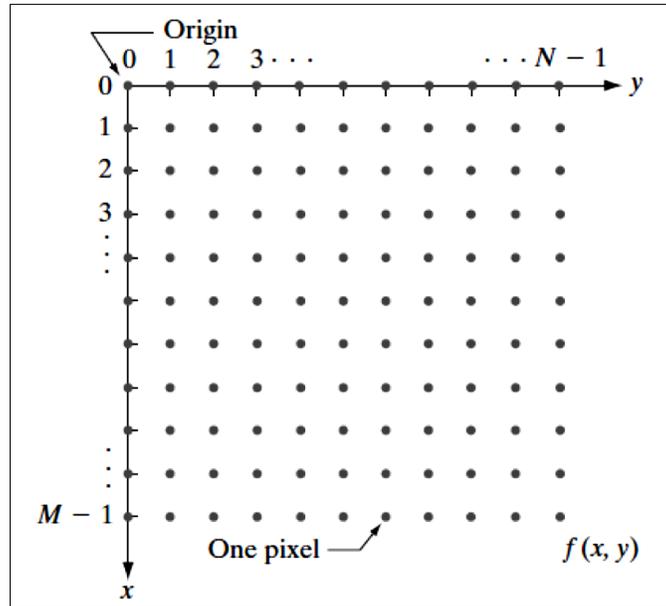


Figura 2.1. Representación de las coordenadas de una imagen (Gonzalez y Woods, 2002).

Este arreglo se puede representar en forma matricial de la siguiente manera:

$$f(x,y) = \begin{bmatrix} f(0,0) & \cdots & f(0,N-1) \\ \vdots & \ddots & \vdots \\ f(M-1,0) & \cdots & f(M-1,N-1) \end{bmatrix}$$

Una vez que una imagen se tiene en forma digital, se obtienen números que pueden ser manipulados de manera aritmética y lógica. Las operaciones aritméticas *suma* y *resta*, así como las operaciones lógicas *and*, *or*, y *xor* se pueden realizar entre dos imágenes del mismo tamaño  $X, Y$  y tener como resultado la imagen  $Z$  ( $Z = X \text{ opn } Y$ ). Esta es una operación algebraica lineal entre imágenes que se realiza entre cada coordenada ( $X(i,j) + Y(i,j) = Z(i,j)$ ) (Myler y Weeks, 1993). Este proceso lineal se muestra en la figura 2.2.

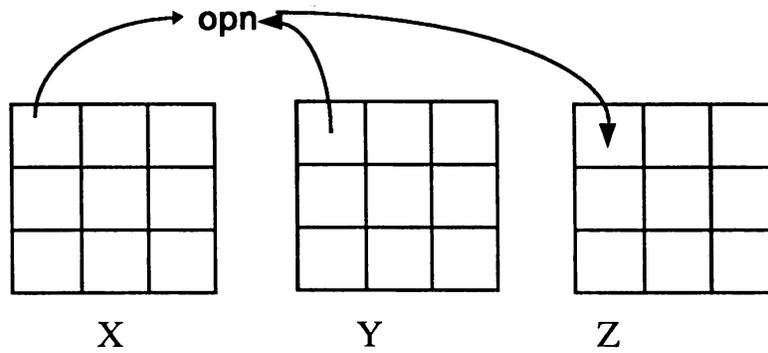


Figura 2.2. Operación algebraica lineal entre imágenes (Myler y Weeks, 1993).

En el caso del operador *not*, este solo opera sobre los píxeles de una imagen. En la tabla 2.1 se muestran las operaciones lógicas entre píxeles correspondientes de dos o más imágenes binarias (con excepción del operador *not*).

Tabla 2.1. Operaciones lógicas.

A	B	A and B	A or B	A xor B	not (A)
0	0	0	0	0	1
0	1	0	1	1	1
1	0	0	1	1	0
1	1	1	1	0	0

Un ejemplo de cada una de las operación lógicas en imágenes binarias descritas anteriormente se muestra en la figura 2.3.

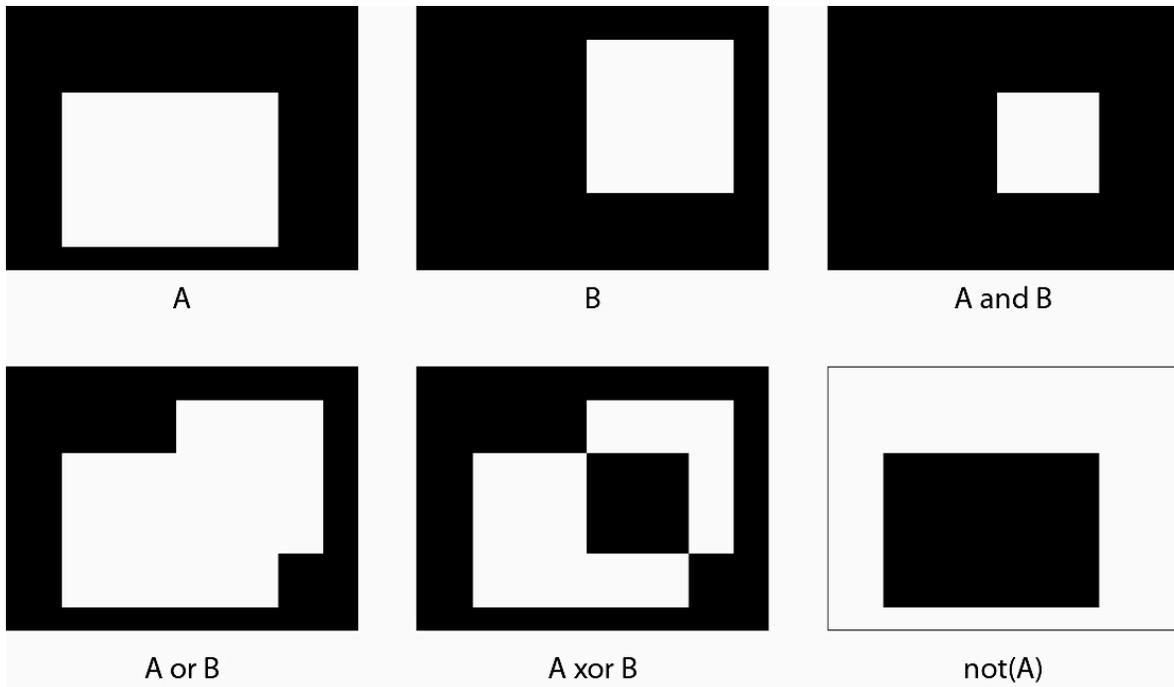


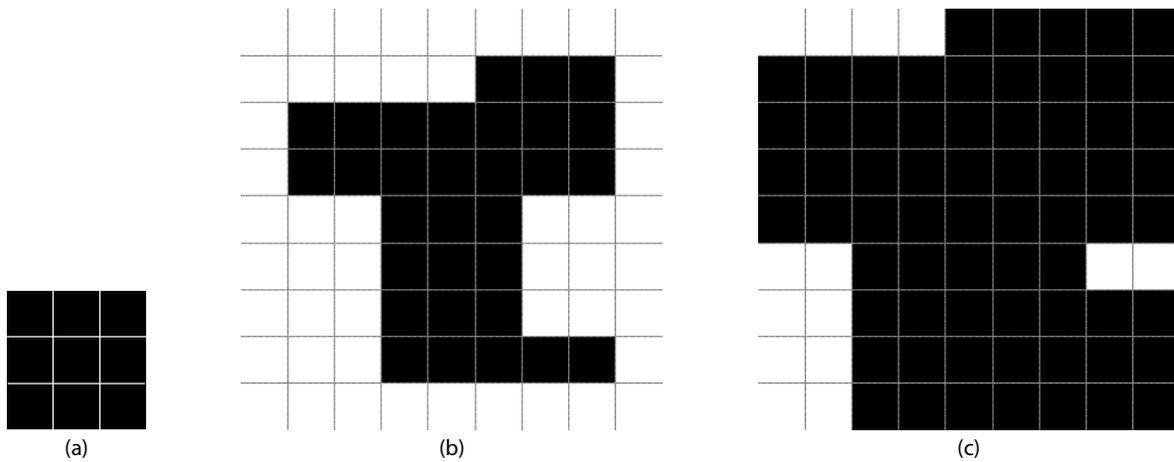
Figura 2.3. Ejemplo de operaciones lógicas entre imágenes binarias.

Otro tipo de operaciones que se pueden realizar son las operaciones morfológicas. En este caso cuando se habla de morfología es en referencia a la herramienta llamada *morfología matemática*, la cual permite extraer los componentes de una imagen que sean útiles en la representación y descripción de la forma de una región de la imagen (Gonzalez y Woods, 2002). La morfología matemática se basa en la teoría de conjuntos y tiene dos operaciones principales, la dilatación y erosión, a través de las cuales se pueden realizar la mayoría de las transformaciones morfológicas, como lo son la apertura y clausura (Serra, 1982). Llamando al objeto a analizar  $X$ , al elemento estructurante  $B$ , y  $B_x$  la traslación de  $B$  de forma que su origen este ubicado en  $x$ , se pueden definir la dilatación y erosión de la siguiente forma (Benítez, 2005):

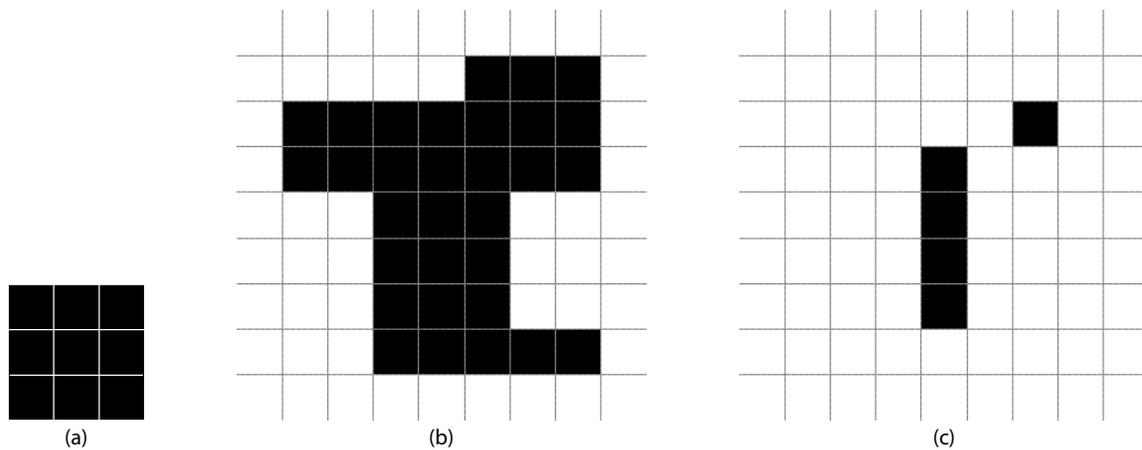
- La *dilatación* de  $X$  por  $B$  se denota  $X \oplus B$  y se define como el conjunto de todos los puntos de  $x$  tales que  $B_x$  intersecciona con  $X$  por lo menos en un punto.  $\delta_B(X) = X \oplus B = \{x: B_x \cap X \neq \emptyset\}$

- La *erosión* de X por B se denota  $X \ominus B$  y se define como el conjunto de todos los puntos de  $x$  tales que  $B_x$  está incluido en X.  
 $\varepsilon_B(X) = X \ominus B = \{x: B_x \subset X\}$

A continuación se muestra un ejemplo de cada una de estas operaciones en las figuras 2.4 y 2.5



*Figura 2.4. Ejemplo de dilatación: (a) Elemento estructural. (b) Imagen X. (c) Resultado de la dilatación.*



*Figura 2.5. Ejemplo de erosión: (a) Elemento estructural. (b) Imagen X. (c) Resultado de la erosión.*

## 2.3 Tecnología FPGA

Un FPGA es un dispositivo lógico programable de propósito general que está compuesto por celdas lógicas cuya interconexión y funcionalidad puede ser programada. Esta característica le brinda la posibilidad de implementar cualquier circuito digital siempre y cuando se dispongan de los recursos necesarios.

Las celdas básicas de un FPGA son mucho más simples que las macroceldas de un dispositivo lógico programable, también llamado PLD (*Programmable Logic Device*), además cuentan con una gran interconectividad, lo que se traduce en una mejor utilización de los recursos, alcanzando niveles de utilización cercanos al 100%, cuando un PLD solo alcanza valores cercanos al 50% (Romero, 2007).

Cada proveedor y cada familia de FPGA ofrecen una versión diferente de la celda básica. En un FPGA típico podemos encontrar tres tipos de celdas básicas:

- *Celdas lógicas*, las que existen en mayor cantidad y están formadas por pequeños bloques lógicos de compuertas programables.
- *Celdas de entrada y salida*, estas celdas están dedicadas a proporcionar la interconectividad entre el FPGA y el exterior.
- *Celdas de distribución*, son las celdas encargadas de manejar las señales de reloj hacia dentro del circuito.

Un circuito digital puede ser diseñado en distintos niveles de abstracción. En el nivel más bajo tenemos el diseño a nivel de transistores, le sigue el nivel de compuertas, después el nivel de transferencia de registros hasta llegar al nivel de descripción de comportamiento, en el que se describe la funcionalidad del circuito en un lenguaje de descripción de hardware o HDL (*Hardware Description Language*) (Hwang, 2005). Es precisamente en este último nivel de abstracción en

el que comúnmente se diseñan los circuitos a ser implementados en dispositivos FPGA.

Existe una gran cantidad de lenguajes de descripción de hardware. Los HDL más utilizados son:

- VHDL
- Verilog
- Handel C

El VHDL es el lenguaje descriptivo más popular, debido principalmente a que es un estándar del IEEE, particularmente el estándar 1164 (Romero, 2007).

Cabe mencionar que actualmente el uso de estos dispositivos está creciendo de manera considerable, principalmente en las áreas de procesamiento digital de señales, comunicaciones, procesamiento de imágenes, computación de alto desempeño y cualquier otra área donde se requiera un alto grado de paralelismo.

## Capítulo 3

### Metodología

Para lograr los objetivos propuestos en este trabajo se llevo a cabo la siguiente metodología:

1. Desarrollo del IP core para desplegado en VGA.
2. Desarrollo del IP core para adquisición de imágenes.
3. Desarrollo del IP core para manejo de memoria.
4. Desarrollo del IP core de la unidad de procesamiento.
5. Integración del sistema.
6. Implementación del algoritmo de análisis en el sistema desarrollado.

La implementación del sistema propuesto requiere una tarjeta de desarrollo que cuente con un FPGA, un decodificador de video compuesto que cumpla con la recomendación ITU-R BT.656 en formato de 8-bit YCrCb (“VIDEO DECODER”), un convertidor digital analógico de video de tres canales con un mínimo de 8-bit de resolución y 25 MHz de velocidad (“VGA DAC”), así como una memoria SSRAM de al menos 307,200 kB de capacidad y 200 MHz de velocidad. Debido a los requerimientos mencionados se decidió hacer uso de una tarjeta de desarrollo Altera DE2-70 la cual cumple todos los requerimientos. Todos los IP core

desarrollados se describieron en VHDL. El software de síntesis usado fue Quartus II, debido a la tarjeta de desarrollo usada.

Un diagrama a bloques del sistema en general se muestra en la figura 3.1.

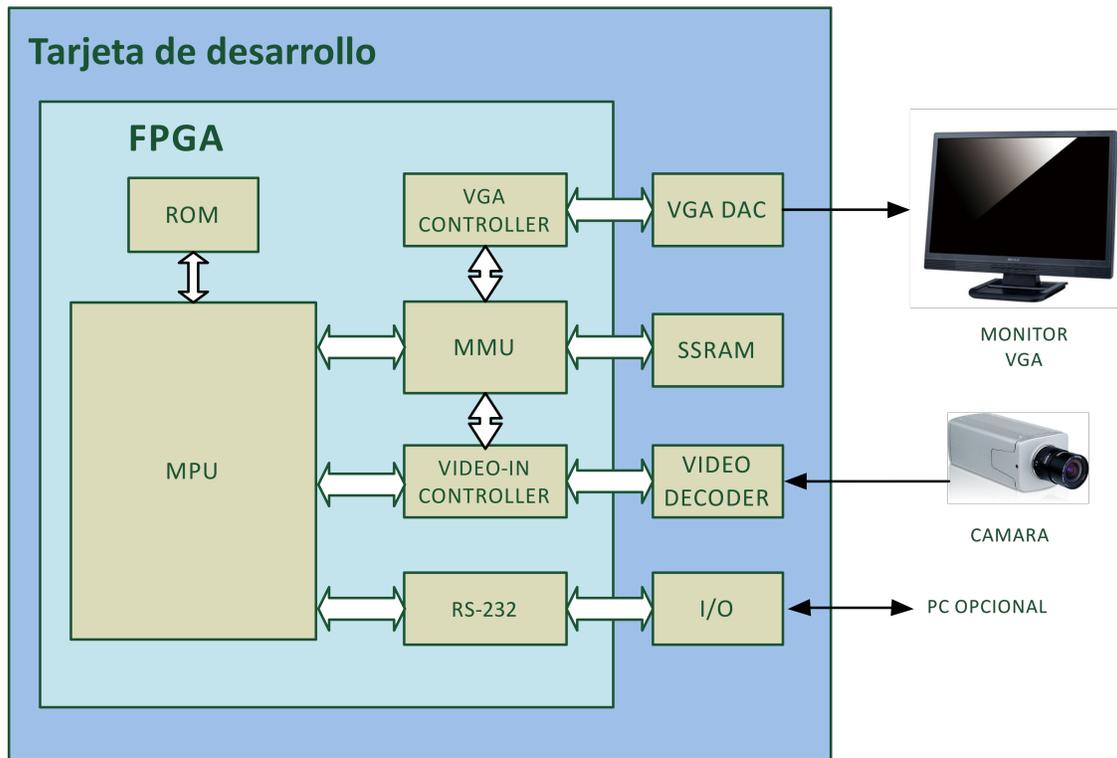


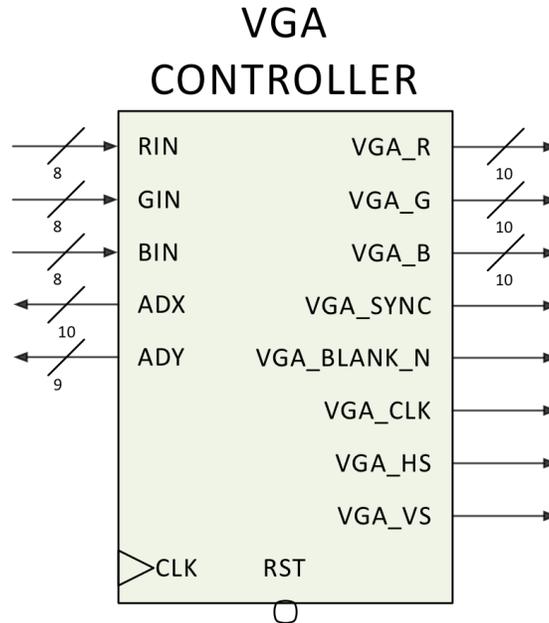
Fig. 3.1. Diagrama a bloques del sistema en general.

A continuación se describen cada uno de los módulos que integran el sistema así como su integración e implementación del algoritmo de análisis.

### 3.1. Despliegado en VGA

El despliegado de la imagen se realiza mediante el IP core "VGA CONTROLLER", que se encarga de pedir los datos de la imagen almacenada al bloque manejador de memoria (MMU) y de proporcionarlos a un convertidor digital analógico de video de alta velocidad (VGA DAC), así como de generar las señales

de sincronía necesarias para el despliegue en un monitor VGA. En la figura 3.2 se muestra un diagrama de este IP core.



*Fig. 3.2. Diagrama del IP core "VGA CONTROLLER".*

A continuación se describen cada uno de los puertos:

- RIN*: Entrada de componente de color rojo.
- GIN*: Entrada de componente de color verde.
- BIN*: Entrada de componente de color azul.
- PIXADDR*: Dirección del pixel siguiente a mostrar.
- ROUT*: Salida al DAC de componente de color rojo.
- GOUT*: Salida al DAC de componente de color verde.
- BOUT*: Salida al DAC de componente de color azul.
- SYNC*: Señal de control para DAC.
- BLANK*: Señal de control para DAC.
- VCLK*: Señal de reloj para DAC.
- VSYNC*: Señal de sincronía vertical para VGA.
- HSYNC*: Señal de sincronía horizontal para VGA.

- CLK: Entrada de reloj.
- RST: Reset asíncrono.

El estándar VGA requiere de los componentes de color rojo, verde y azul. En este trabajo la adquisición de imágenes se realizó en escala de grises, por lo que a cada componente se le asigna el mismo valor de lectura correspondiente a la luminancia del pixel a mostrar.

El convertidor analógico digital usado que se muestra en la figura 3.3 es el ADV7123 de Analog Devices, el cual integra tres convertidores analógico-digitales de alta velocidad con una resolución de 10-bit.

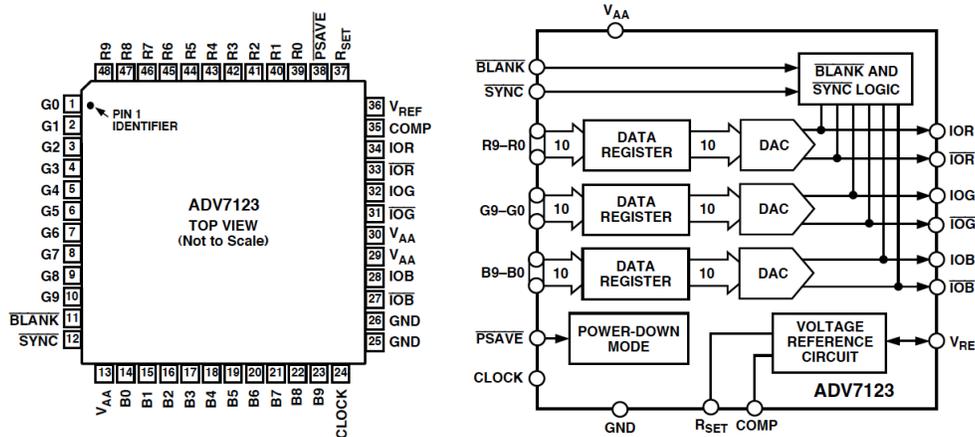


Fig. 3.3. DAC ADV7123 de Analog Devices.

Este IP core ajusta la longitud de los valores de cada componente, cuya entrada es de 8-bit a 10-bit y se los envía al convertidor, junto con las señales de control SYNK, BLANK y la señal de reloj VCLK. Las señales analógicas provenientes del convertidor junto con las señales de sincronía VSYNC y HSYNC son transmitidas al monitor VGA para el despliegado de la imagen. Este bloque necesita una señal de reloj de 50 MHz la cual se toma directamente del reloj incluido en la tarjeta de desarrollo.

En la figura 3.4 se muestra un diagrama de tiempo para la sincronía horizontal del estándar VGA. En Chu (2008) se puede encontrar de manera detallada el funcionamiento del estándar así como la implementación en FPGA de un controlador sencillo (solo contiene el generador de las señales de sincronía y de direcciones).

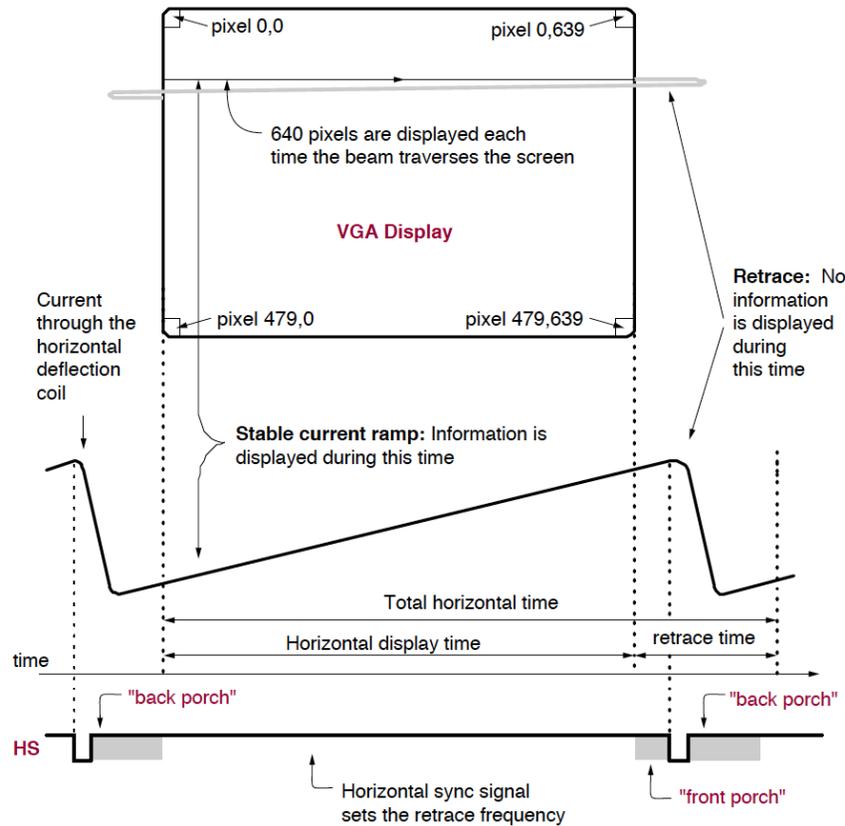


Fig. 3.4. Diagrama de tiempo de la sincronía horizontal en VGA.

### 3.2. Adquisición de imágenes

La adquisición de la imagen se realiza mediante el uso de un decodificador de video compuesto configurado para recibir una señal de video en estándar NTSC, y convertirla en información de video compuesto en formato YCrCb 4:2:2 compatible con el estándar de 8-bit ITU-R BT.656.

El control del decodificador de video se realiza mediante el IP core “VIDEO-IN CONTROLLER” el cual tiene la tarea de configurar el decodificador de video, extraer de éste la información referente al valor de luminancia de cada pixel y generar la dirección y señal de escritura para su almacenamiento en la memoria. En la figura 3.5 se muestra el diagrama de este IP core.

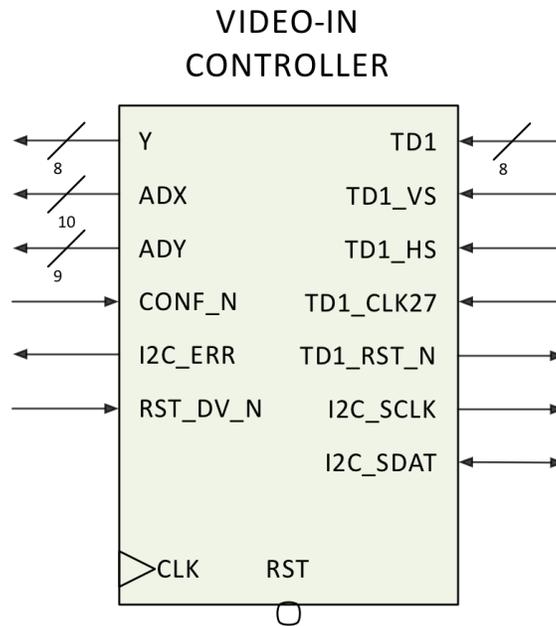


Fig. 3.5. Diagrama del IP core de decodificación de video “VIDEO-IN CONTROLLER”.

A continuación se describen cada uno de los puertos:

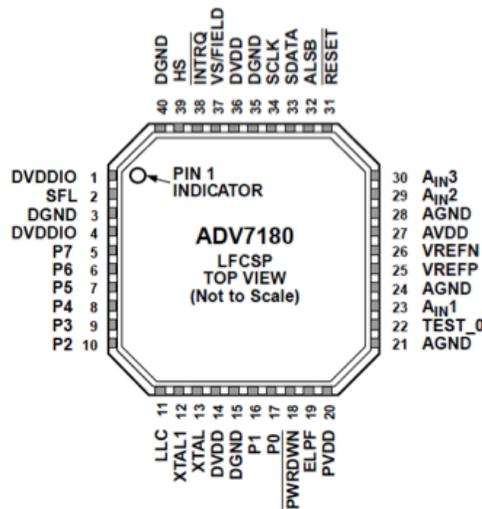
- Y: Dato de luminancia.
- AD: Dirección del pixel a almacenar.
- W: Señal de activación de escritura.
- TD: Dato recibido del decodificador.
- VS: Señal de sincronía vertical.
- HS: Señal de sincronía horizontal.
- LLC: Señal de reloj de 27 MHz generada por el decodificador.
- RESET: Señal de reset para el decodificador.
- I2C\_CLK: Señal de reloj para configuración I2C.

-*I2C\_DAT*: Señal de datos I2C.

-*CLK*: Entrada de reloj.

-*RST*: Reset asíncrono.

El decodificador de video que se usó es el ADV7180 de Analog Devices (fig. 3.6), el cuál es un decodificador de señales de televisión de definición estándar, compatible con señales de televisión analógica en estándar NTSC, PAL y SECAM



*Fig. 3.6. Decodificador de video ADV7180 de Analog Devices.*

El decodificador convierte estas señales en datos de componentes de video en formato YCrCb 4:2:2 compatibles con la interfaz estándar de 8-bit ITU-R BT.656. En la figura 3.7 se muestra la composición del tren de datos de la interfaz. Estos datos están codificados como se especifica en la recomendación ITU-R BT.601. En la figura 3.8 se muestra un ejemplo de una imagen y su correspondiente separación de luminancia y cromancia (YCrCb). En la figura 3.9 se muestra un ejemplo de la codificación YCrCb 4:2:2 y su diferencia con YCrCb 4:4:4, la cual consiste en que en la primera se tiene un dato de luminancia por cada pixel y un dato de cromancia Cr y Cb respectivamente por cada dos pixeles, a diferencia de la segunda codificación que tiene un dato de cada una por pixel.

### Composición del tren de datos en la interfaz

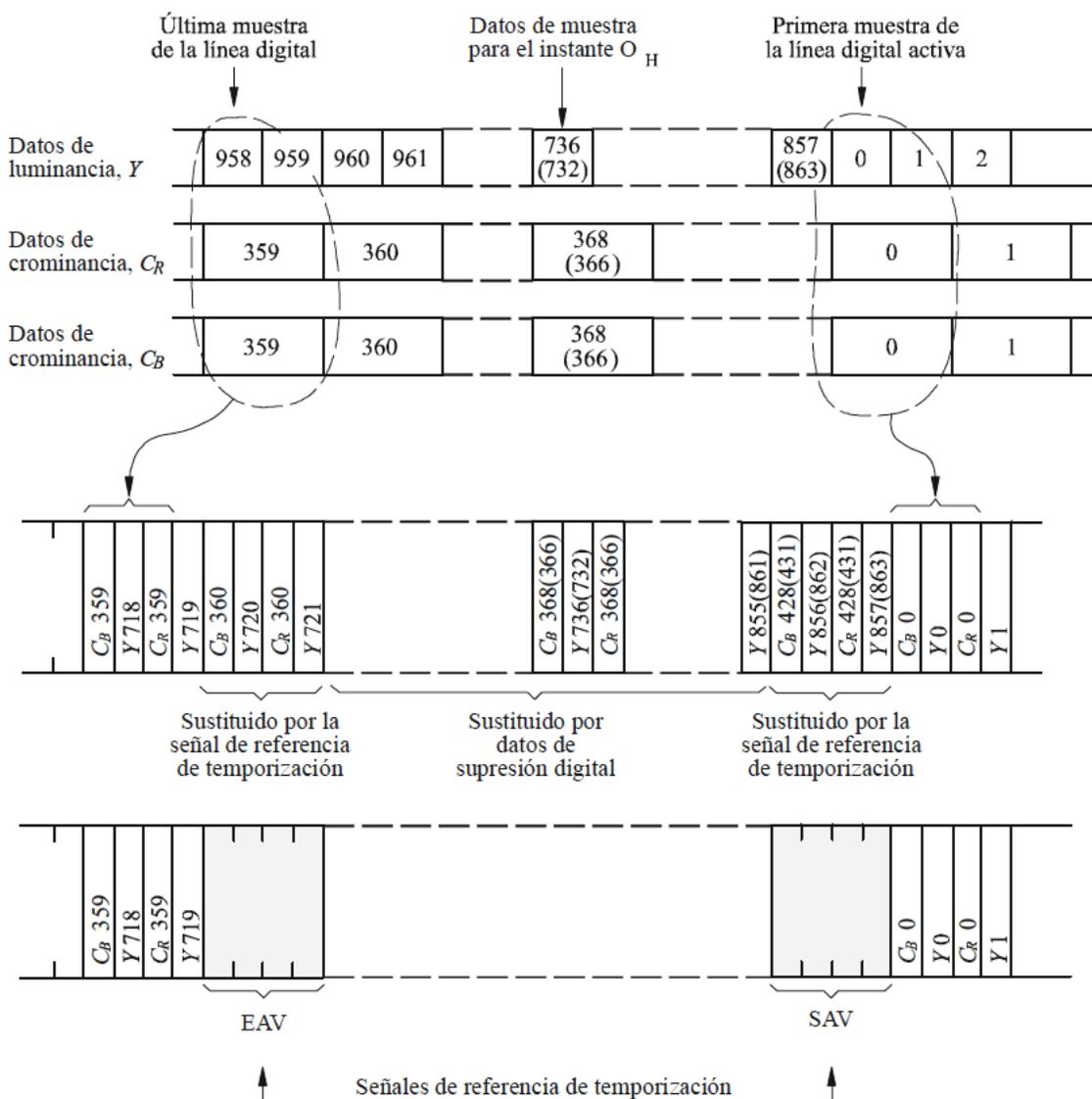


Fig. 3.7. Composición del tren de datos de la interfaz ITU-R BT.656.

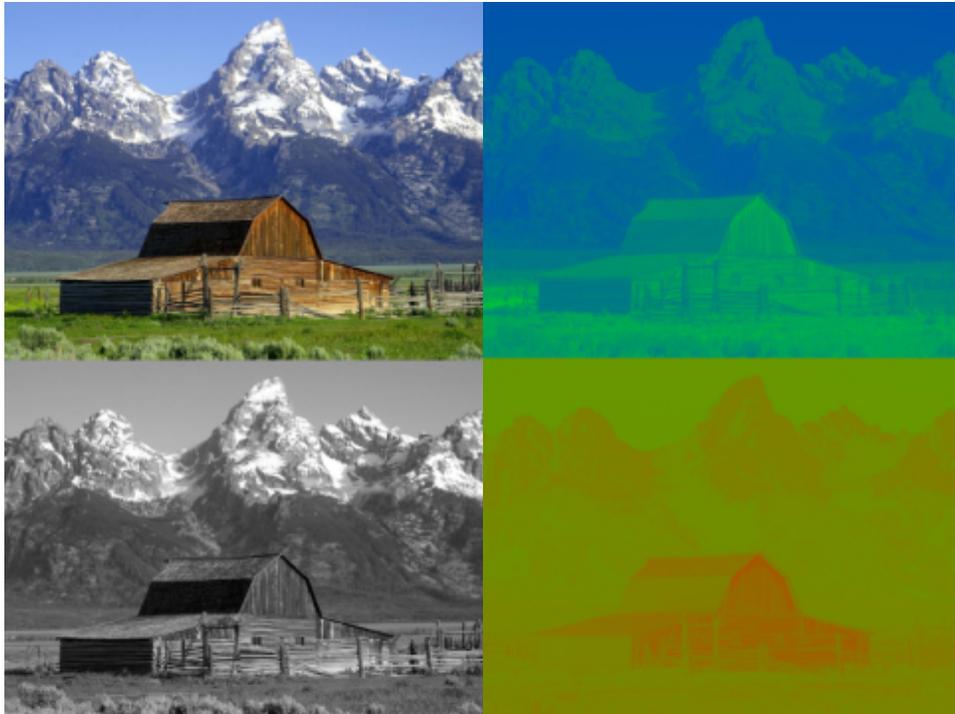


Fig. 3.8. Separación de una imagen en componentes YCbCr.

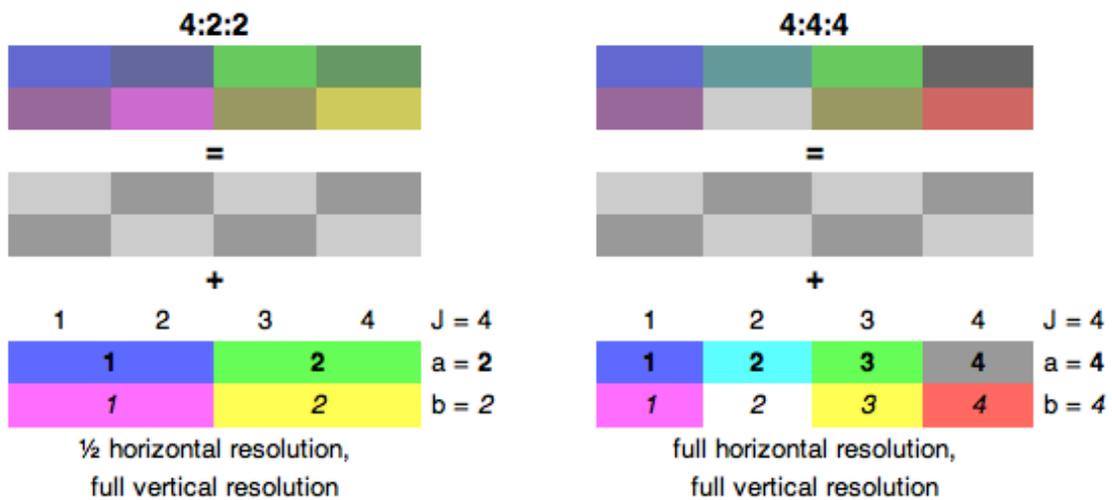


Fig. 3.9. Codificación YCbCr 4:2:2 y 4:4:4.

En la figura 3.10 se muestra un diagrama a bloques del IP core “VIDEO-IN CONTROLLER”. Este IP core está formado por tres bloques principales, un bloque configurador (I2C CONF), un bloque detector de secuencia (LOCK&SYNC) y un

bloque generador de señal de activación y dirección de escritura (AD GEN & WR ACT).

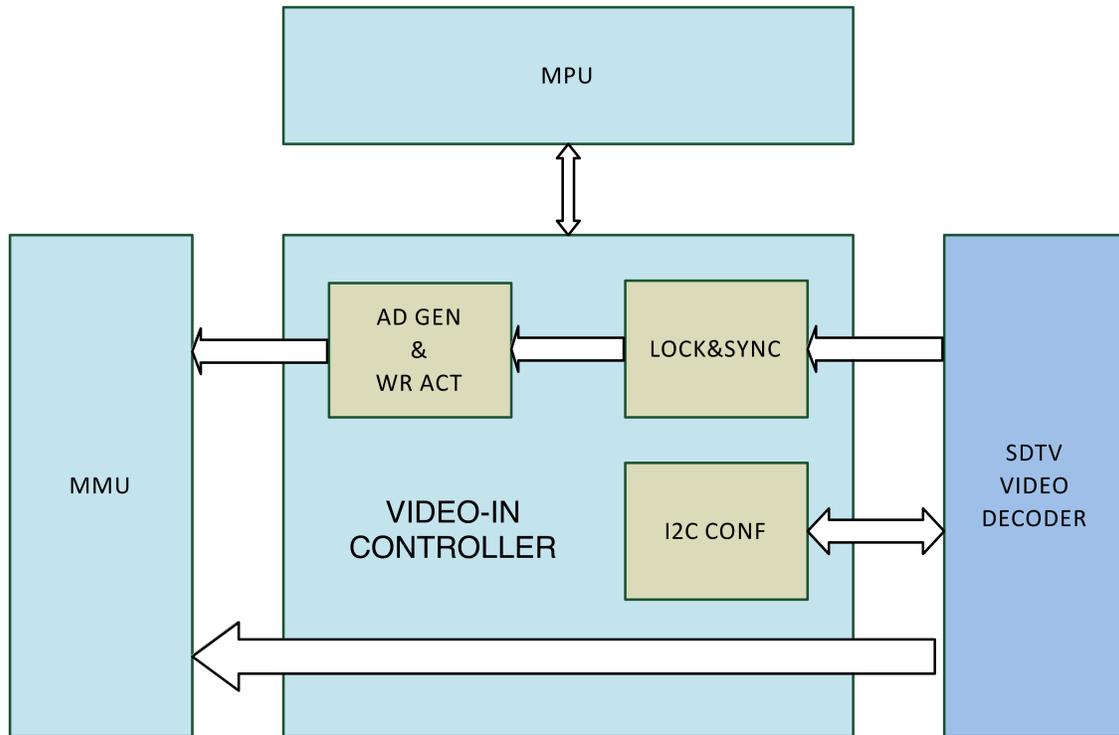


Fig. 3.10. Diagrama a bloques del IP core "VIDEO-IN CONTROLLER".

El bloque configurador se encarga de mandar la configuración al decodificador por medio de comunicación con protocolo I2C, esta configuración consta de registros que le indican al decodificador el modo en el que se va a usar, específicamente en este trabajo se configuro para ser usado con entrada de video en estándar NTSC, en modo de video compuesto en el canal AIN1 del decodificador (lugar donde se encuentra conectado el conector RCA de la tarjeta de desarrollo) y con ajustes de crominancia y luminancia automáticos.

Los datos recibidos del decodificador están entrelazados y en formato 4:2:2, por lo que el segundo bloque se encarga de detectar el inicio y fin de cada línea, denotado por la secuencia de datos en hexadecimal "FF,00,00,XY" así como también sí es línea par o non.

Con las operaciones anteriores, el tercer bloque genera la dirección correspondiente a cada pixel (haciendo el desentrelazado), además de generar la señal de activación de escritura que permite guardar únicamente los datos correspondientes a luminancia. En la figura 3.11 se muestra una parte del diagrama de tiempo de la sincronía horizontal, en donde se puede observar la secuencia de inicio y fin de video activo, así como los datos de luminancia (Y) y crominancia (Cb y Cr).

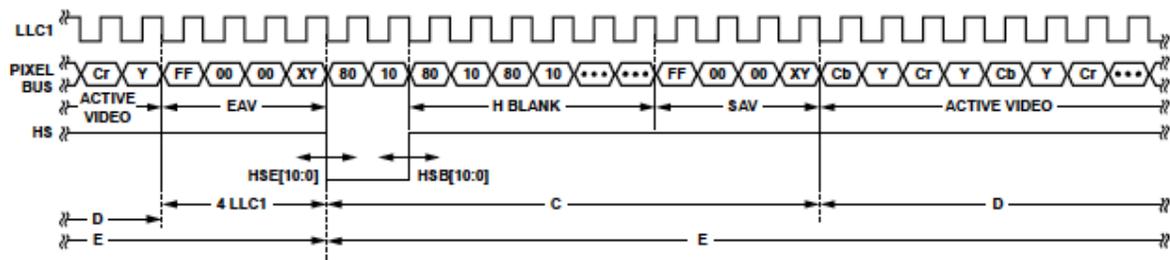
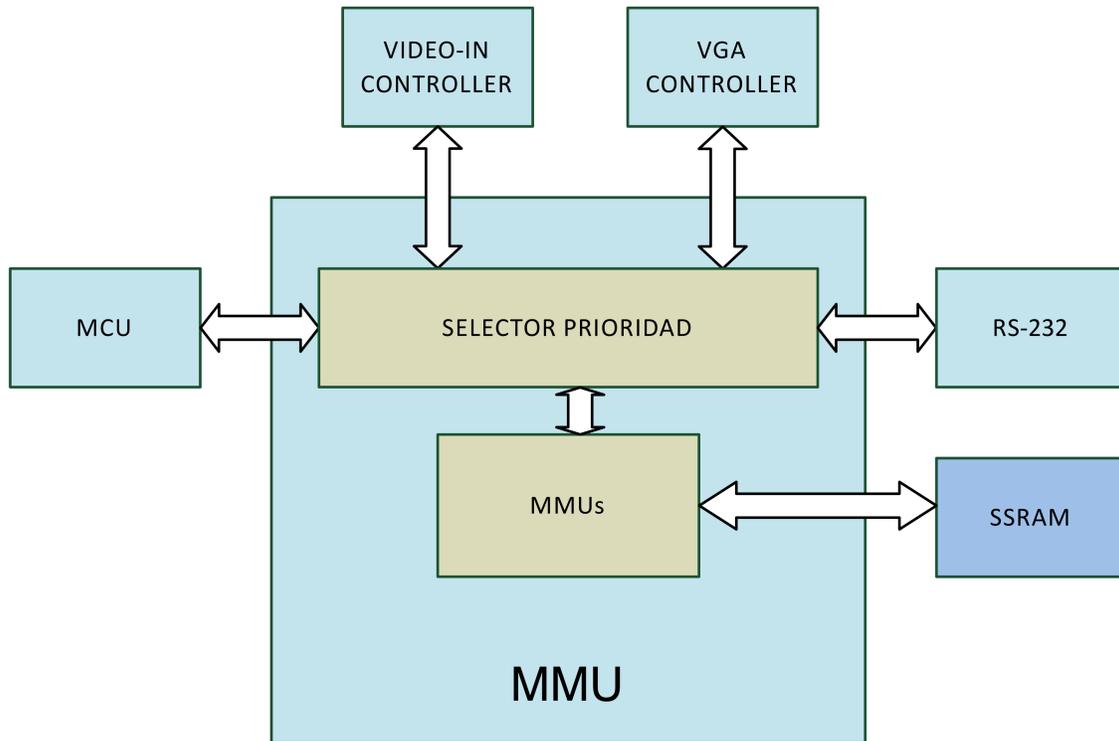


Fig. 3.11. Diagrama de tiempo de sincronía horizontal.

Este IP core requiere una señal de reloj de 50 MHz, mientras que el decodificador requiere de un reloj de 28.63 MHz, los cuales vienen integrados en la tarjeta de desarrollo.

### 3.3. Manejo de memoria

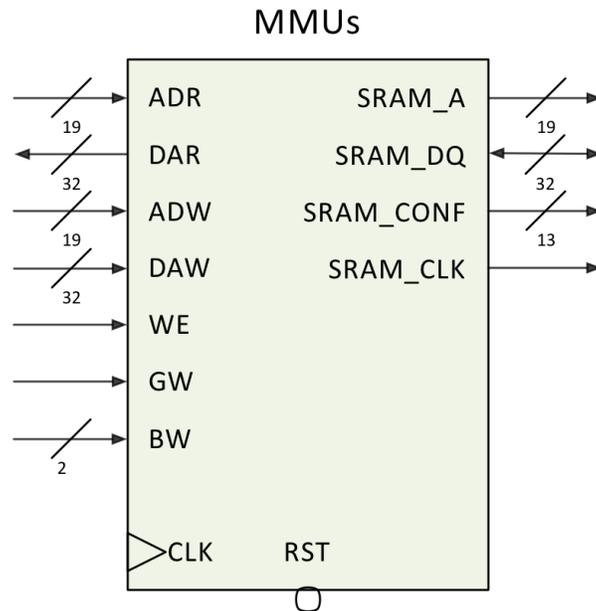
El MMU (Memory Management Unit, unidad de manejo de memoria) es el encargado de controlar el acceso a la unidad de memoria. En la figura 3.12 se muestra un diagrama a bloques del IP core "MMU".



*Fig. 3.12. Diagrama a bloques del IP core "MMU".*

Este IP core está conformado por dos bloques principales: MMUs y SELECTOR PRIORIDAD. En conjunto logran el control del tipo de memoria usados y el fácil acceso a escritura y lectura por parte de los demás bloques que integran el sistema.

El bloque MMUs es el encargado de manejar la memoria SSRAM, en este caso una ISSI IS61LPS51236A de 2 MB y 200 MHz de velocidad. En la figura 3.13 se muestra un diagrama de este bloque.



*Fig. 3.13. Diagrama del bloque MMUs.*

A continuación se describe cada uno de los puertos:

- ADR*: Dirección de lectura.
- ADW*: Dirección de escritura.
- DAW*: Dato a ser guardado.
- DAR*: Dato leído.
- WE*: Activación de escritura en flanco positivo.
- SRAM\_A*: Selección de dirección en la SSRAM.
- SRAM\_DQ*: Puerto bidireccional para lectura y escritura en la SSRAM.
- SRAM\_CONF*: Bus que integra la salida para los pines de configuración de la SSRAM (CE, BWE, OE, etc.).
- *SRAM\_CLK*: Salida de reloj a la SSRAM.
- CLK*: Entrada de reloj.
- RST*: Reset asíncrono.

La aplicación requiere que los datos puedan ser leídos y grabados a una velocidad mínima de 28 MHz, además de poder escribir y leer al mismo tiempo. La SSRAM empleada solo tiene un puerto de dirección y un puerto bidireccional para

los datos, por lo que se desarrollo una estructura digital que integra flip-flops tipo D, un multiplexor y una FSM, con lo que se logra emular una RAM con puerto de lectura y escritura independiente. En la figura 3.14 se muestra un diagrama mas detallado de este bloque.

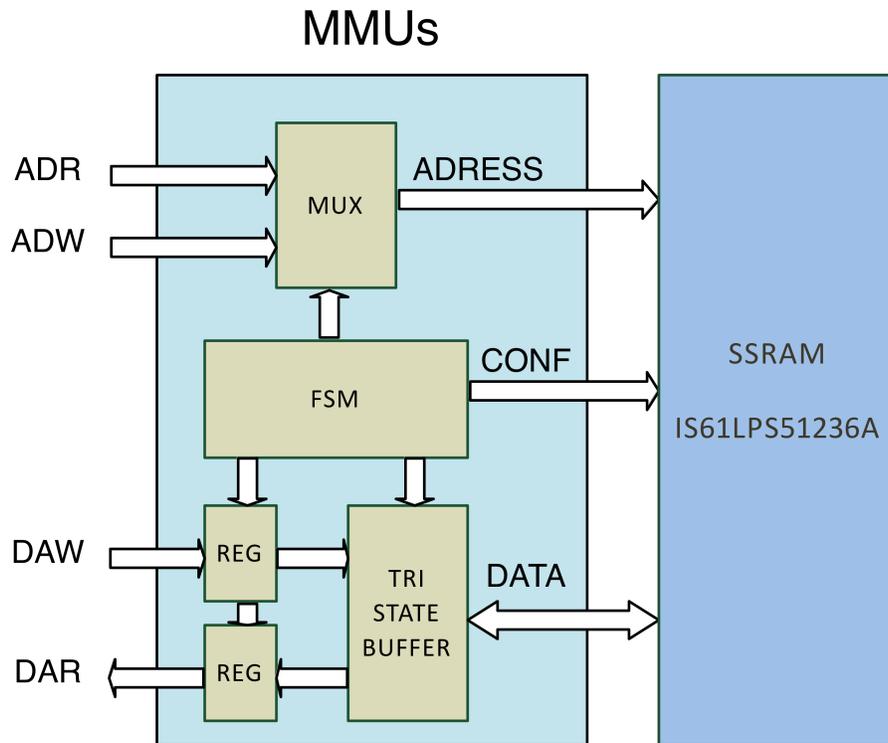


Fig. 3.14. Diagrama detallado del bloque MMUs.

Para que los datos puedan ser grabados y leídos al mismo tiempo (visto desde fuera del bloque) se requiere una velocidad de reloj interna del bloque 4 veces mayor a la velocidad a la que se va a usar, ya que se usan dos ciclos para la escritura y dos para la lectura. Considerando la futura aplicación del proyecto y la velocidad máxima de operación de la SSRAM, se definió la velocidad de operación a 50 MHz, por lo que el bloque requiere una señal de reloj de 200 MHz, la cual es generada a partir de un PLL integrado en el FPGA, ya que el reloj incluido en la tarjeta de desarrollo es de 50 MHz.

El bloque SELECTOR PRIORIDAD es un multiplexor controlado por el MPU el cual permite el acceso a la escritura a solamente un bloque a la vez. Otra característica es la de poder detener el proceso de escritura mientras se realiza la lectura si así se desea, ya que se detecto que a la máxima velocidad permitida por la memoria SSRAM integrada en la tarjeta de desarrollo, se presenta un error en la integridad de los datos de lectura al usar el bloque MMUs descrito previamente.

### **3.4. Unidad de procesamiento**

En este IP core es donde se lleva a cabo la decodificación de las instrucciones y el procesamiento de las imágenes. Esta conformado por los tres bloques principales: MCU (unidad de control), ALU (unidad lógico aritmética) y AGN (generador de direcciones).

La descripción del funcionamiento de esta unidad esta organizada en los tres distintos tipos de operaciones en imágenes que se implementaron: operaciones morfológicas, lógicas y de binarización. Las operaciones implementadas con excepción de la función de binarizado están diseñadas para realizarse en imágenes binarias.

La primer funcionalidad implementada es la del binarizado de imágenes. La imagen de entrada obtenida del decodificador de video se encuentra en escala de grises, por lo que este bloque es requerido para hacer la conversión a imagen binaria.

La operación se realiza por medio de la funcionalidad de “Binarizado” dentro del MPU. La ALU recibe como parámetro el nivel de gris a partir del cual se hará el binarizado. Se hace lectura de la imagen original que se encuentra almacenada en la memoria SSRAM y se guarda directamente el resultado en la misma memoria pero en una locación diferente. En este bloque no es necesaria una FIFO ya que

la MMU se diseño de forma en que virtualmente se puede escribir y leer al mismo tiempo. Una vez que se realiza esta operación, la imagen guardada puede ser procesada por las demás operaciones. En la figura 3.15 se muestra un diagrama a bloques de la operación de binarizado.

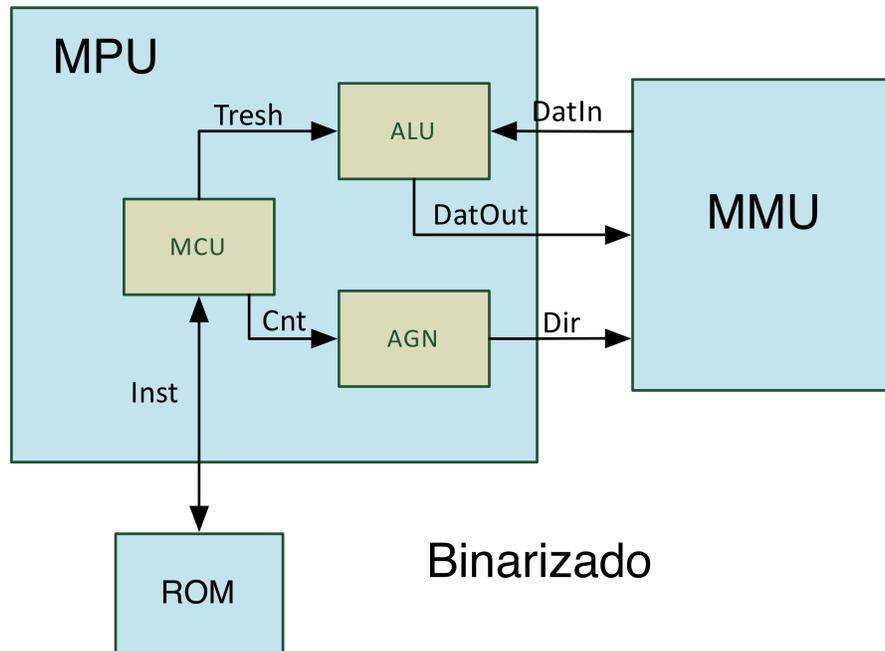
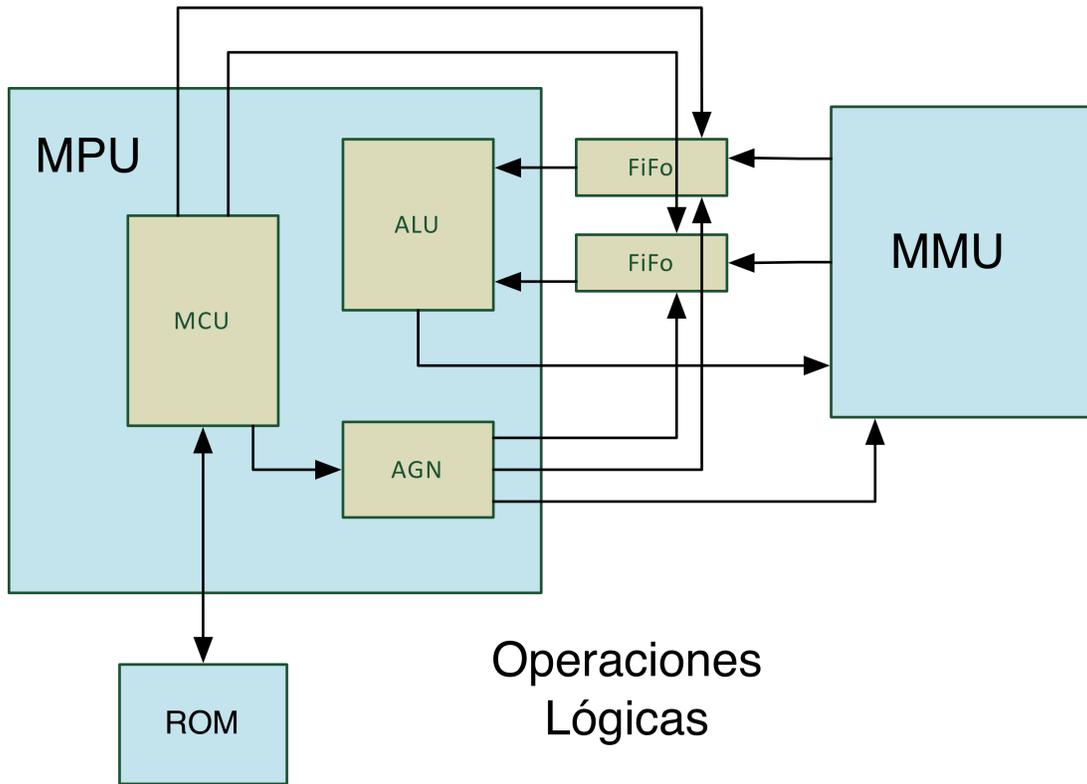


Fig. 3.15. Diagrama a bloques de operación de binarizado.

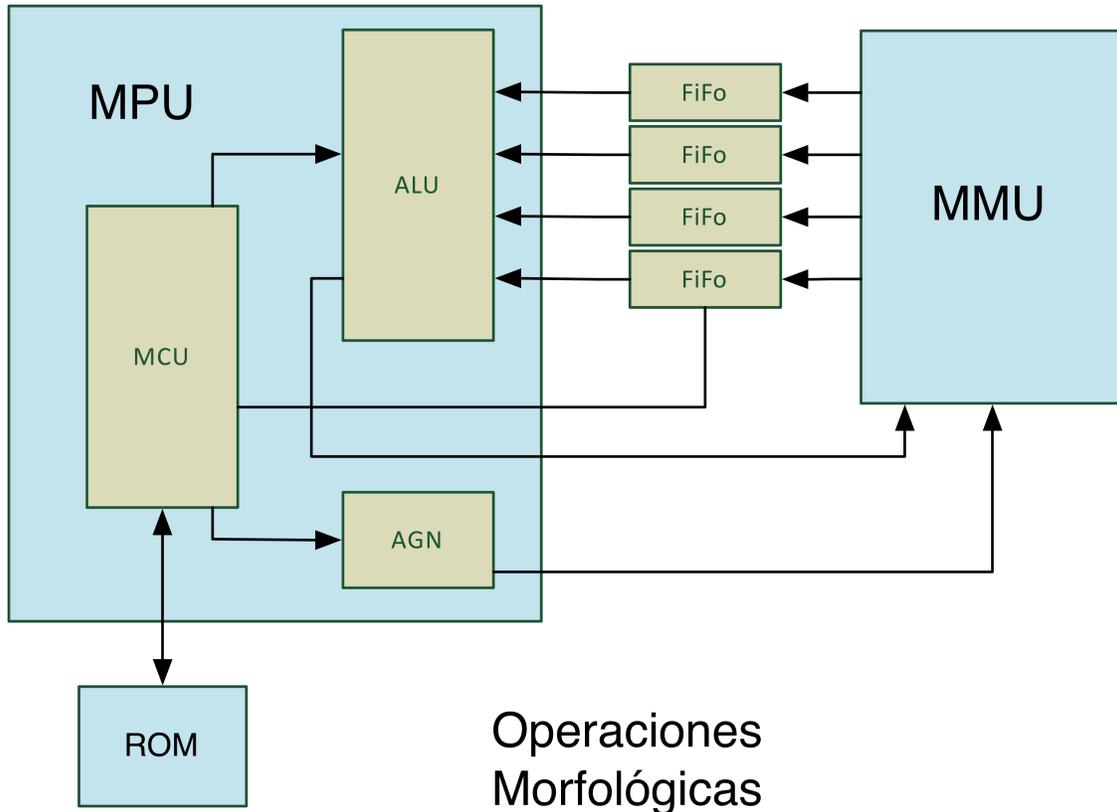
La segunda funcionalidad es la de procesamiento lógico (debido a que las operaciones se realizan sobre imágenes binarias, las operaciones aritméticas se realizan mediante las operaciones lógicas). En este bloque es donde están implementadas las operaciones: and, or, xor y not.

Esta funcionalidad hace uso de dos arreglos de memorias FIFO como buffer de entrada. El sistema carga un renglón de cada imagen en cada FIFO a la vez, realiza la operación, guarda el renglón resultante en la memoria y vuelve a repetir el proceso hasta que todos los renglones de las imágenes de entrada han sido procesados. En la figura 3.16 se muestra un diagrama a bloques de las operaciones lógicas.



*Fig. 3.16. Diagrama a bloques de operaciones lógicas.*

La tercer funcionalidad es la de procesamiento morfológico. En ésta se encuentran implementadas las operaciones erosión y dilatación. Esta funcionalidad hace uso de cuatro arreglos de memorias FIFO como buffer de entrada. En la figura 3.17 se muestra el diagrama a bloques de operaciones morfológicas.



*Fig. 3.17. Diagrama a bloques de operaciones morfológicas.*

La operación morfológica sigue los pasos siguientes:

1. Se cargan cuatro renglones de la imagen a operar, uno en cada FIFO.
2. Se recorre una mascara de 3 x 3 pixeles en las tres primeras FIFO y el resultado del primer renglón se almacena directamente en la memoria.
3. Se vuelve a recorrer la mascara pero ahora desde la segunda FIFO hasta la cuarta y se almacena el resultado (equivalente al segundo renglón) en la memoria. Al mismo tiempo la primer FIFO carga los datos del siguiente renglón (renglón cinco) de la imagen original.
4. Se recorre nuevamente la mascara pero ahora con la primera, tercera y cuarta FIFO y se almacena el resultado (equivalente al tercer renglón) en la memoria. Al mismo tiempo la segunda FIFO carga los datos del siguiente renglón (renglón seis) de la imagen original.

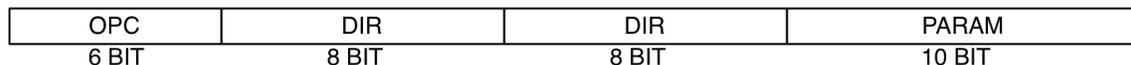
5. Se sigue la secuencia hasta que se termina de recorrer la máscara por toda la imagen.

Este IP core requiere de una señal de reloj de 200 MHz, la cuál se toma de un PLL dentro del mismo FPGA que multiplica la frecuencia original de la tarjeta de desarrollo que es de 50 MHz.

### 3.4.1. MCU

El “MCU” o unidad controladora es el bloque principal dentro del MPU. Este bloque tiene la tarea de leer las instrucciones programadas por el usuario y ejecutar las acciones indicadas.

El procesador desarrollado tiene un juego de 14 instrucciones, con un tamaño de instrucción de 32 bit. Se definió de este tamaño para que en un solo conjunto se tuviera toda la información requerida para todas las instrucciones desarrolladas. En la figura 3.18 se muestra como esta constituida la instrucción.



*Fig. 3.18. Codificación de la instrucción.*

A continuación se describen cada uno de estos campos:

-*OPC*: En este campo se especifica la instrucción que se debe ejecutar. Tiene una longitud de 6 bit por lo que se podría definir hasta un máximo de 64 instrucciones.

-*DIR*: La instrucción contempla dos campos de dirección de 8 bit cada uno.

Corresponden a las direcciones de origen y destino, y en algunas instrucciones a las dos direcciones de origen.

-PARAM: Este campo define los requerimientos extras de algunas instrucciones, como dirección destino para las instrucciones que ocupan dos direcciones de origen o la definición del elemento estructurante para las instrucciones de morfología.

En la tabla 3.1 se indican las instrucciones y una descripción breve de las instrucciones implementadas en este trabajo.

*Tabla 3.1. Juego de instrucciones del procesador desarrollado.*

<b>Copiar</b>	
MOVB or,de	Copiar origen a destino binario.
MOVG or,de	Copiar origen a destino en escala de grises.
MOVR de	Copiar origen a memoria de desplegado en grises.
MOVO or	Copiar a destino la imagen en la memoria de captura en grises.
<b>Umbral</b>	
THL or,de,val	Leer origen en grises y guardar en destino el resultado de la binarización ( menor a val ).

THG or,de,val	Leer origen en grises y guardar en destino el resultado de la binarización ( mayor a val ).
THE or,de,val	Leer origen en grises y guardar en destino el resultado de la binarización ( igual a val ).

### Operaciones Lógicas

AND or1,or2,de	Operación AND entre imagen origen 1 y origen 2 y almacenamiento de resultado en destino.
OR or1,or2,de	Operación OR entre imagen origen 1 y origen 2 y almacenamiento de resultado en destino.
XOR or1,or2,de	Operación XOR entre imagen origen 1 y origen 2 y almacenamiento de resultado en destino.
NOT or1,or2,de	Operación NOT entre imagen origen 1 y origen 2 y almacenamiento de resultado en destino.

### Operaciones Morfológicas

DIL or,de,mask	Operación de dilatación de imagen origen con mascara "mask" y guardado en destino.
ERO or,de,mask	Operación de erosión de imagen origen con mascara "mask" y guardado en destino.

## Otros

DESB or,de

Operación de conversión imagen origen en binario a grises en destino.

### 3.5. Integración del sistema

Una vez terminados todos los IP core descritos anteriormente se procedió a integrarlos en un solo sistema. En primera instancia se realizó una prueba de interconexión de los módulos de adquisición, almacenaje y despliegado de imágenes. En la figura 3.19 se muestra un diagrama a bloques de la interconexión del sistema de adquisición, almacenaje y despliegado.

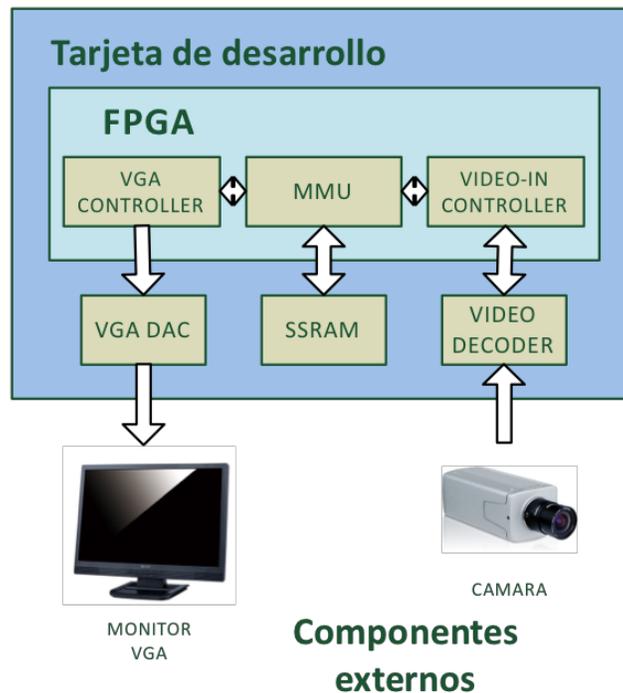


Fig. 3.19. Diagrama a bloques de la interconexión del sistema de adquisición, almacenaje y despliegado.

En la figura 3.20 se muestra el diagrama de la interconexión de los IP core correspondientes a cada sistema así como su conexión a los circuitos externos.

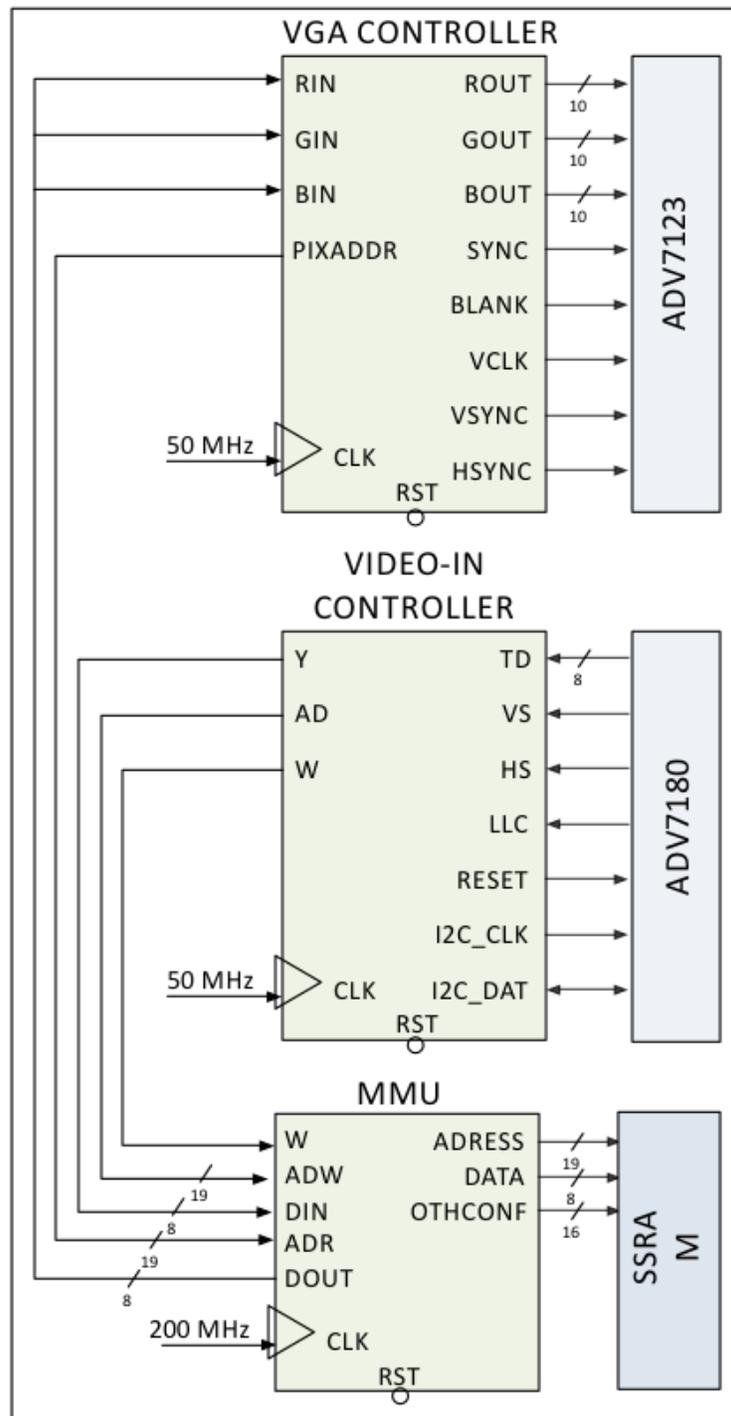


Fig. 3.20. Diagrama de interconexión de los IP core "VGA CONTROLLER", "VIDEO-IN CONTROLLER" Y "MMU".

### 3.6. Implementación de algoritmo

El algoritmo que se implementó en el procesador desarrollado consiste en la adquisición de la imagen directamente en el sistema y la aplicación de filtros morfológicos, los cuáles pudieran ayudar a reducir el tiempo de procesamiento en la aplicación de la técnica usada en granulometría descrita en Morales et al. (2010), la cuál requiere la determinación de los residuos entre aperturas de tamaño diferentes.

A continuación se describen los pasos necesarios para determinar el residuo de una apertura tamaño nueve con una tamaño tres empleando el procesador desarrollado.

1. Se usa la función umbral para binarizar la imagen de entrada la cuál continuamente se esta almacenando en la locación numero uno de la memoria y se almacena el resultado en la locación de memoria dos.
2. Se realiza la erosión de la memoria dos y se almacena en tres. Después se realiza la dilatación de tres y se almacena en cuatro. Al termino de la operación se tiene la apertura tamaño tres en la memoria cuatro (tamaño tres debido a que la mascara implementada en este trabajo es de tres por tres, por lo que solamente se pueden realizar operaciones en múltiplos de tres).
3. Para la apertura tamaño nueve se realiza la erosión de memoria dos y se almacena en tres, después nuevamente la erosión pero ahora de memoria tres a dos y por último de memoria dos a tres. Al terminar seguimos con la dilatación de memoria tres a dos, de dos a tres y de tres a dos, con lo que al termino tendremos la apertura tamaño nueve en la memoria dos.
4. El residuo se calcula usando la operación xor entre la memoria dos y cuatro, almacenando el resultado en la memoria tres.

Los pasos en ensamblador se muestran a continuación:

1.

*THG 1,2,128*

2.

*ERO 2,3*

*DIL 3,4*

3.

*ERO 2,3*

*ERO 3,2*

*ERO 2,3*

*DIL 3,2*

*DIL 2,3*

*DIL 3,2*

4.

*XOR 2,4,3*

Para mostrar los resultados en el monitor hay que activar los switch 0 y 1 del sistema de desarrollo para multiplexar la salida a la memoria tres. Para enviar la imagen a la PC hay que presionar el botón 1 (la imagen que se manda es la misma que esta seleccionada en la salida del monitor).

## **Capítulo 4**

### **Resultados y discusión**

En este capítulo se presentan las diferencias de tiempo entre el sistema desarrollado y la PC, en el procesamiento de el algoritmo de análisis implementado. Además se presentan los logros obtenidos con este trabajo así como las conclusiones y el trabajo a futuro.

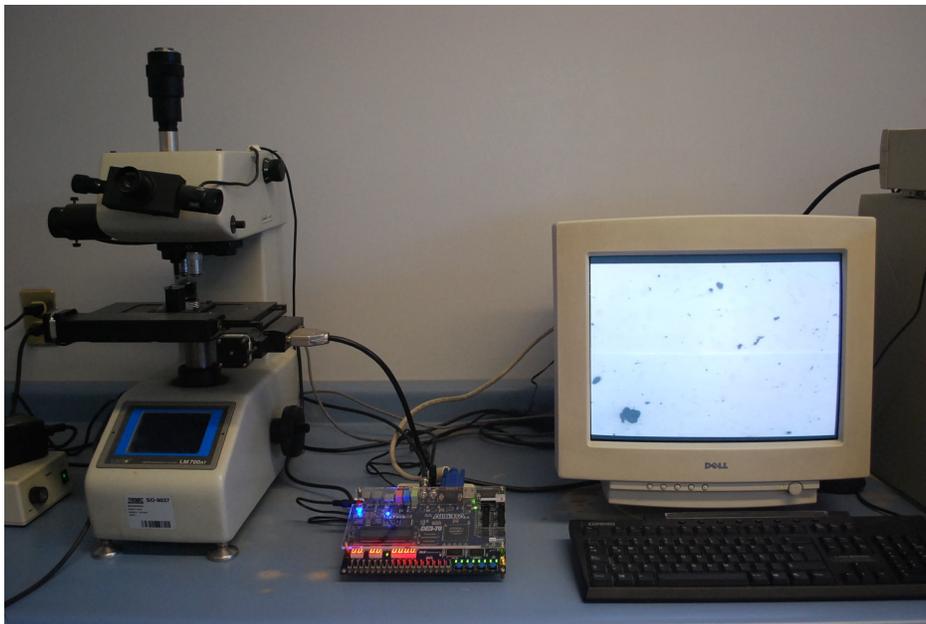
#### **4.1. Resultados**

El resultado de este trabajo es un procesador morfológico de imágenes embebido en FPGA para el análisis metalográfico de fundiciones nodulares austemperizadas capaz de reducir los tiempos de procesamiento con respecto a una PC.

El procesador desarrollado se implemento en una tarjeta de desarrollo que en conjunto con distintos módulos hacen un sistema capaz de adquirir la imagen de tamaño 640 x 480 pixeles en escala de grises directamente de cualquier fuente de video analógica con estándar NTSC, procesar cualquier algoritmo basado en

morfología matemática y las operaciones suma, resta, and, not, or, xor, y mostrar los resultados en un monitor VGA.

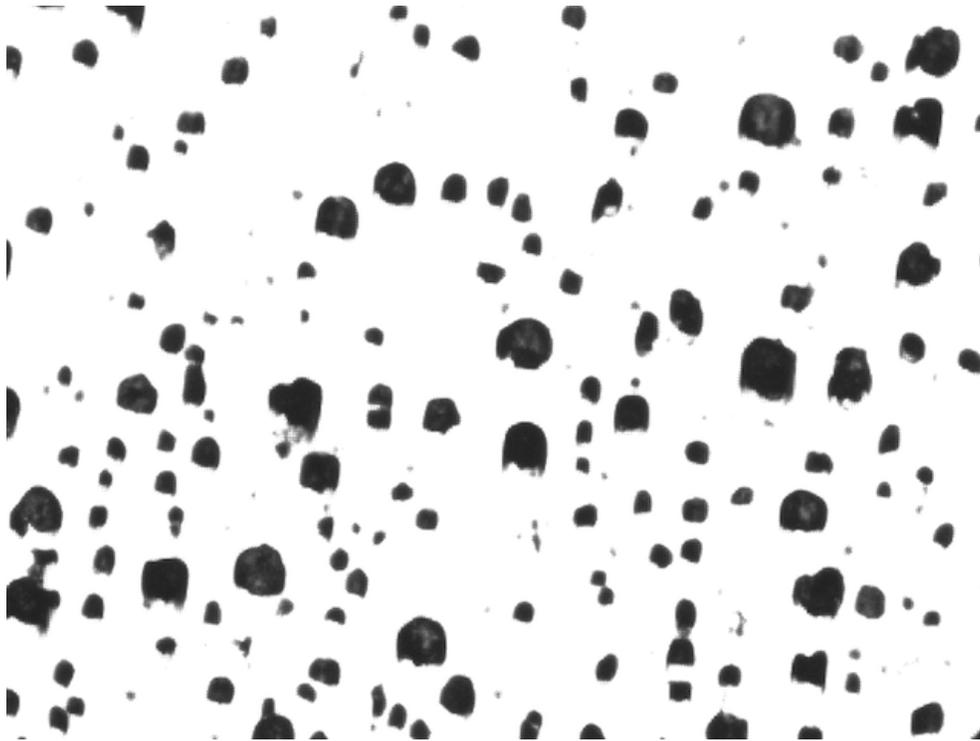
El sistema se probó en el laboratorio de metalografía de la Universidad Autónoma de Querétaro. Se usó una muestra de una fundición nodular austemperizada. La muestra se situó en el microscopio de metalografía, el cual cuenta con una cámara de video analógica, que se conectó al sistema. El despliegado de la imagen obtenida se hizo mediante un monitor VGA. En la figura 4.1 se muestra una foto del sistema en funcionamiento.



*Figura 4.1. Sistema en funcionamiento.*

Una vez adquirida la imagen se envió a la PC para poder realizar la comparación de tiempo con ésta. Después de ser enviada, se procedió a realizar el algoritmo de procesamiento tanto en la PC como en el sistema.

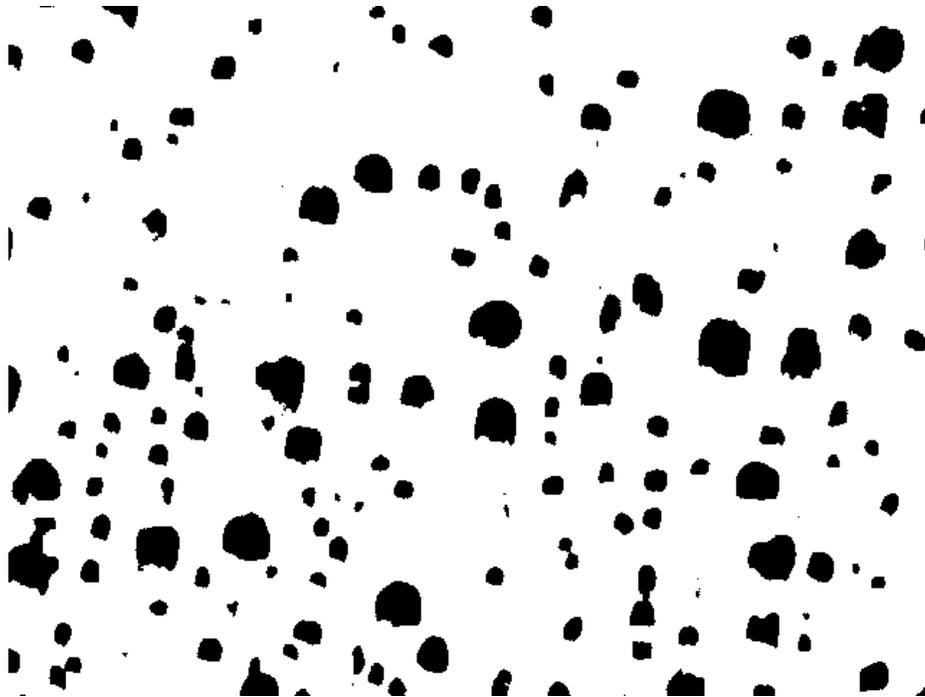
En la figura 4.2 se muestra la imagen de prueba que se usó tanto en el sistema como en la PC. La imagen tiene un tamaño de 640 x 480 píxeles y está en escala de grises.



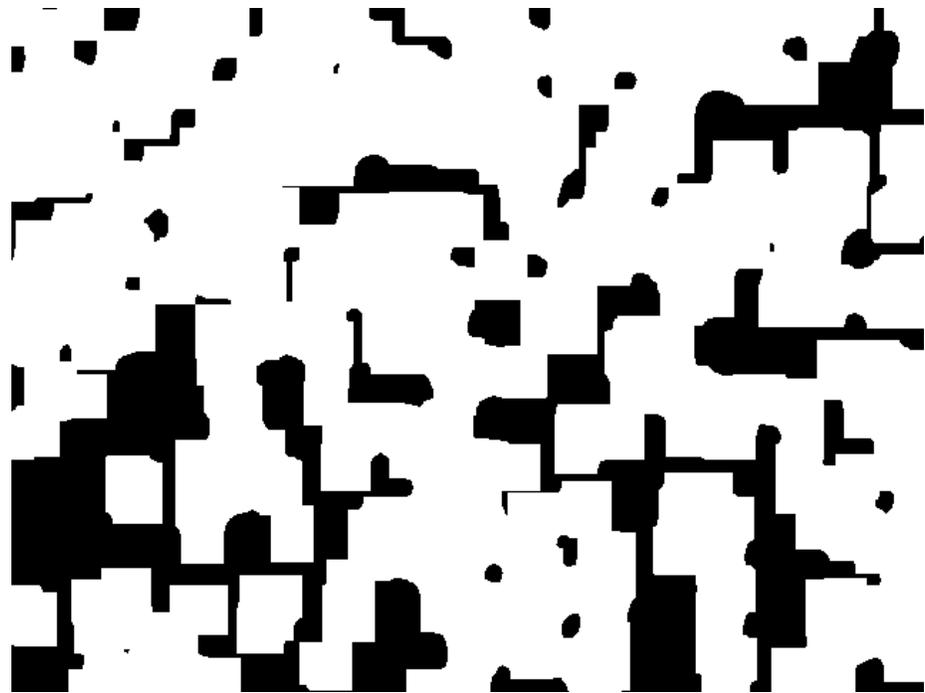
*Figura 4.2. Imagen de prueba.*

La prueba que se realizó fue la diferencia entre la apertura 30 de la imagen y la 21. La binarización se hizo con un umbral de 128 (escala de 0 a 255). Solo se contabilizó el tiempo de procesamiento de las dos aperturas y la diferencia, sin contar el tiempo de adquisición, lectura y binarización de la imagen.

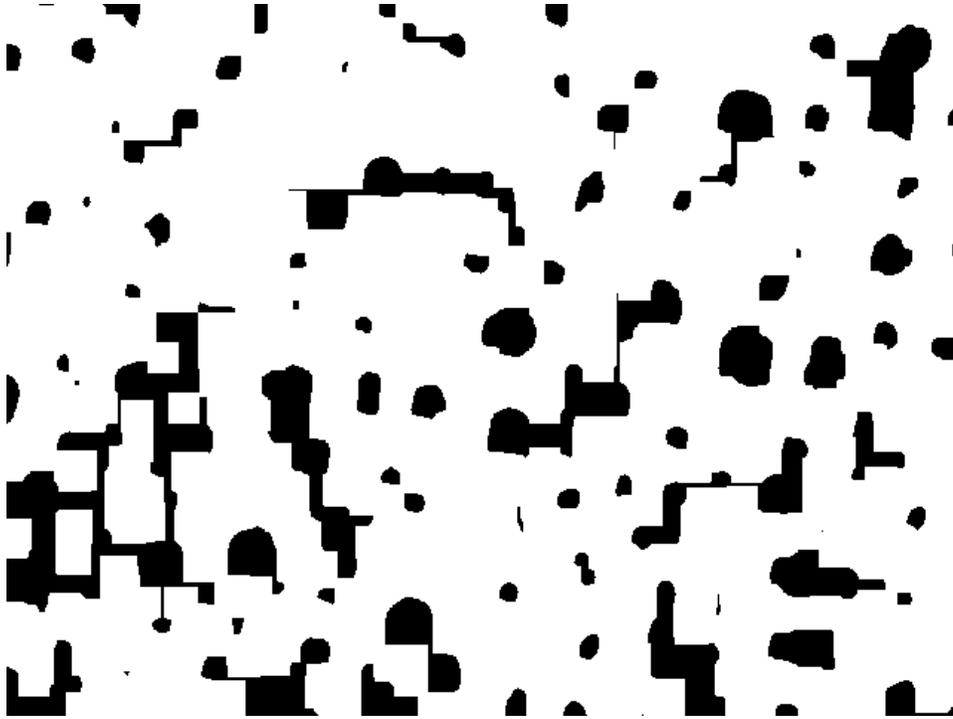
A continuación se muestran de las figuras 4.3 a 4.6 las imágenes resultantes de la binarización, apertura 30, apertura 21, diferencia y en la tabla 4.1 la comparación de los tiempos de procesamiento.



*Figura 4.3. Imagen de prueba binarizada.*



*Figura 4.4. Apertura 30.*



*Figura 4.5. Apertura 21.*



*Figura 4.6. Diferencia apertura 30 y 21.*

## **4.2. Conclusiones**

El presente trabajo de tesis destaca como el uso de un sistema de procesamiento de imágenes basado en tecnología FPGA permite una opción que gracias al paralelismo de esta tecnología y la posibilidad de integrar una gran cantidad de tareas en el mismo circuito integrado, reduce los tiempos de ejecución de los algoritmos de análisis metalográfico basados en morfología matemática con respecto al uso de una PC. Además, gracias a la capacidad de reconfigurabilidad del FPGA, es posible la posterior integración de más algoritmos de procesamiento de imagen en el mismo integrado.

Gracias a los módulos desarrollados en este trabajo, además de lograr reducir los tiempos de procesamiento, se obtuvo un sistema capaz de adquirir, procesar y desplegar imágenes directamente.

Cabe destacar que el sistema desarrollado no está limitado al análisis de imágenes metalográficas, ya que al contar con los operadores morfológicos básicos, se podrá realizar cualquier otro algoritmo de procesamiento digital de imágenes programado por el usuario, siempre y cuando este se base únicamente en los operadores de morfología matemática y aritmético lógicos implementados.

## **4.3. Trabajo a futuro**

Como trabajo a futuro se tiene el desarrollo potencial en tres áreas del trabajo presentado. En primer lugar el módulo de adquisición de imagen se puede modificar para obtener imágenes a color y no solo en escala de grises como se realizó en este trabajo. En segundo lugar tenemos el aumento de las instrucciones del procesador, como conteo de objetos y generación de gráficas, lo que permitiría tener un sistema completo para análisis metalográfico que no necesitaría el uso de una PC para obtener los resultados finales. Por último se puede mejorar el

sistema de manejo de memoria, lo que permitiría reducir aún más los tiempos de ejecución de los algoritmos.

## Referencias

ASM Handbook. 1990. ASM Handbook Volume 1: Properties and Selections: Irons, Steels, and High-Performance Alloys. ASM International. Materials Park, Ohio. ISBN: 978-0-87170-377-4

ASM Handbook. 2004. ASM Handbook Volume 9: Metallography and microstructures. ASM International. Materials Park, Ohio. ISBN: 978-0-87170-706-2

Batlle, J., et al. 2002. A New FPGA/DSP-Based Parallel Architecture for Real-Time Image Processing. Real-Time Imaging, 8:345-356.

Benítez, J. P. 2005. Medición de dimensiones por visión fuera de línea. Universidad Autónoma de Querétaro - Facultad de Ingeniería, Santiago de Querétaro, Qro.

Chu, P. 2008. FPGA Prototyping by VHDL examples (1st Ed.). John Wiley and Sons, Inc., USA. ISBN: 978-0-470-18531-5.

Gonzalez, R. C., R. E. Woods. 2002. Digital Image Processing (2nd Ed.). Prentice-Hall, Inc., New Jersey. ISBN: 0-201-18075-8.

Hwang, E. O. 2005. Digital Logic and Microprocessor Design with VHDL (1st Ed.). Thomson. ISBN: 9780534465933.

Krill, B., et al. 2010. An efficient FPGA-based dynamic partial reconfiguration design flow and environment for image and signal processing IP cores. Signal Processing: Image Communications, 25:377-387.

Méndez, A. 2008. Sistema automático para análisis de unidades formadoras de colonias mediante técnicas de procesamiento digital de imágenes. Tesis de maestría. Universidad Autónoma de Querétaro - Facultad de Ingeniería, Santiago de Querétaro, Qro.

Morales, L. 2010. Diseño de plataforma hardware-software para el desarrollo de aplicaciones industriales basadas en FPGA. Tesis de doctorado. Universidad Autónoma de Querétaro - Facultad de Ingeniería, Santiago de Querétaro, Qro.

Morales, L. A., et al. 2010. Spatial distribution and spheroidicity characterization of graphite nodules based on morphological tools. Journal of Materials Processing Technology, 210:2:335-342.

Morales, L. A. 2005. Identificación de huellas digitales. Tesis de maestría. Universidad Autónoma de Querétaro - Facultad de Ingeniería, Santiago de Querétaro, Qro.

Myler, H. R., A. R. Weeks. 1993. Computer Image Recipes in C. Prentice-Hall, Inc., New Jersey. ISBN: 0-13-189879-5.

Rangel, P. 2005. Procesamiento morfológico de imágenes bajo VHDL y FPGA. Tesis de maestría. Universidad de Guanajuato - Facultad de Ingeniería Mecánica Eléctrica y Electrónica, Salamanca, Gto.

Razo, D. A. 2006. Procesador morfológico de imágenes. Tesis de licenciatura. Universidad de Guanajuato - Facultad de Ingeniería Mecánica Eléctrica y Electrónica, Salamanca, Gto.

Recommendation ITU-R BT.656-5. 2007. Interface for digital component video signals in 525-line and 625-line television systems operating at the 4:2:2 level of Recommendation ITU-R BT.601\*. International Telecommunication Union.

Romero, R. J. 2007. Electrónica Digital y Lógica Programable. Universidad de Guanajuato. ISBN: 968-864-449-8.

Serra, J. 1982. Image Analysis and Mathematical Morphology (1st Ed.). Academic Press, London.

Shao, C., C. Liang. 2008. Reconfigurable Morphological Image Processing Accelerator for Video Object Segmentation. Journal of Signal Processing Systems, 62:77-96.

Vargas, D. 2000. Filtros Morfológicos Conexos: Detección de cisuras en imágenes obtenidas por resonancia magnética del cerebro. Tesis de maestría. Universidad Autónoma de Querétaro - Facultad de Ingeniería, Santiago de Querétaro, Qro.

Villela, I. J. 2004. Evaluación a la microestructura y a las propiedades mecánicas del acero inoxidable 316LS y del titanio Ti-6Al-4V como biomateriales. Tesis de licenciatura. Universidad de las Américas Puebla. Cholula, Pue.

## **Apéndice A**

### **Artículo publicado**

# IP core Genérico para Adquisición y Despliegado de Imágenes en Plataforma Basada en FPGA

González Aguirre Marco Antonio<sup>1</sup>, Morales Velázquez Luis<sup>1</sup>, Osornio Ríos Roque Alfredo<sup>1</sup>  
y Morales Hernández Luis Alberto<sup>2</sup>

<sup>1</sup> HSPdigital - CA Mecatrónica, Facultad de Ingeniería, Universidad Autónoma de Querétaro, Campus San Juan del Río, Río Moctezuma 249, 76807 San Juan del Río, Qro., México.

<sup>2</sup> Facultad de Ingeniería, Universidad Autónoma de Querétaro, Campus San Juan del Río, Río Moctezuma 249, 76807 San Juan del Río, Qro., México.  
(E-MAIL de contacto: magonzalez@hspdigital.org)

## Resumen

*El presente trabajo consiste en la adquisición y despliegado de imágenes de 640 x 480 pixeles en escala de grises de 8-bit, dentro de una plataforma basada en FPGA. La adquisición de la imagen se realiza mediante el uso de un decodificador de video compuesto configurado para recibir una señal de video en estándar NTSC, y convertirla en información de video compuesto en formato YCrCb 4:2:2 compatible con el estándar de 8-bit ITU-R BT.656. Esta información se transmite a un bloque desarrollado en el FPGA que se encarga de extraer los datos de luminancia de la información proporcionada y de generar las direcciones de almacenamiento. Los datos y las direcciones generadas se transmiten a otro bloque encargado de manejar una memoria SSRAM, en donde se almacenan los datos de luminancia correspondientes a la dirección de cada uno de los pixeles de la imagen. El despliegado de la imagen se realiza mediante un bloque controlador VGA, que se encarga de pedir los datos de la imagen almacenada al bloque manejador de memoria y de proporcionarlos a un convertidor digital analógico de video de alta velocidad, así como de generar las señales de sincronía necesarias para el despliegue en un monitor VGA.*

**Palabras clave:** FPGA, adquisición de imagen, despliegado VGA.

## 1. Introducción

Actualmente en el desarrollo de plataformas especializadas para procesamiento de imágenes se tiene la tendencia al uso de tecnología FPGA, debido

a la reconfigurabilidad y al alto grado de paralelismo de esta tecnología [1], lo que permite la implementación de algoritmos que reducen sustancialmente el tiempo de procesamiento en comparación con los sistemas basados en PC, además de que la reconfigurabilidad permite hacer modificaciones posteriores, lo que le da una ventaja con respecto a los sistemas basados en ASICs.

Estas plataformas requieren adquirir la imagen o imágenes a ser procesadas y retornar de alguna forma los resultados por lo que se decidió desarrollar en primera instancia un sistema de adquisición y despliegado de imágenes.

Para evitar restringir el sistema a una sola fuente de imágenes (una cámara en particular), se usó de un decodificador de video compuesto para señales de video analógicas en estándar NTSC, con lo que se brinda la posibilidad de cambiar fácilmente la fuente, ya que una gran cantidad de cámaras comerciales y productos de entretenimiento cuentan con este estándar. De la misma manera, en lo que se refiere al despliegado de imágenes, se optó por la salida en estándar VGA, ya que hace posible la conexión a casi cualquier monitor de PC.

Cabe mencionar que la mayoría de los fabricantes de FPGA ofrecen núcleos de propiedad intelectual (IP cores, *Intellectual Property cores*) que realizan las operaciones de control de los dispositivos de adquisición, despliegado y almacenamiento desarrolladas en este trabajo, pero su uso requiere el pago de derechos de autor, además de que su uso está restringido a sus productos. El desarrollo de estos IP cores propios, permite reducir costos y ser independientes del fabricante de los dispositivos.

## 2. Metodología

A continuación se describen los elementos desarrollados para realizar el objetivo del trabajo. Un diagrama del sistema en general se muestra en la figura 1.



Fig. 1 Diagrama a bloques del sistema.

La implementación del sistema propuesto requiere una tarjeta de desarrollo que cuente con un FPGA, un decodificador de video compuesto que cumpla con la recomendación ITU-R BT.656 en formato de 8-bit YCrCb (“VIDEO DECODER”), un convertidor digital analógico de video de tres canales con un mínimo de 8-bit de resolución y 25 MHz de velocidad (“VGA DAC”), así como una memoria SSRAM de al menos 307,200 kB de capacidad y 200 MHz de velocidad. Debido a los requerimientos mencionados se decidió hacer uso de una tarjeta de desarrollo Altera DE2-70 (figura 2) la cual cumple todos los requerimientos.

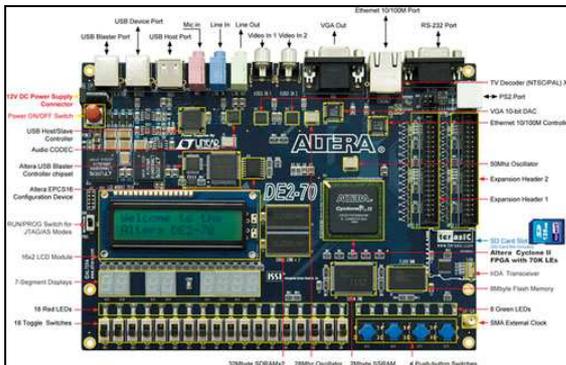


Fig. 2 Tarjeta de desarrollo Altera DE2-70.

Mediante el lenguaje de descripción de hardware VHDL y la herramienta de síntesis Quartus de Altera, se desarrollaron e implementaron en el

FPGA los IP cores “VIDEO-IN CONTROLLER” encargado del control del decodificador de video, “MMU” cuyo objetivo es administrar el manejo de la SSRAM y “VGA CONTROLLER” encargado del manejo del convertidor digital analógico para la salida al monitor VGA. La interconexión de los IP cores se muestra en la figura 3.

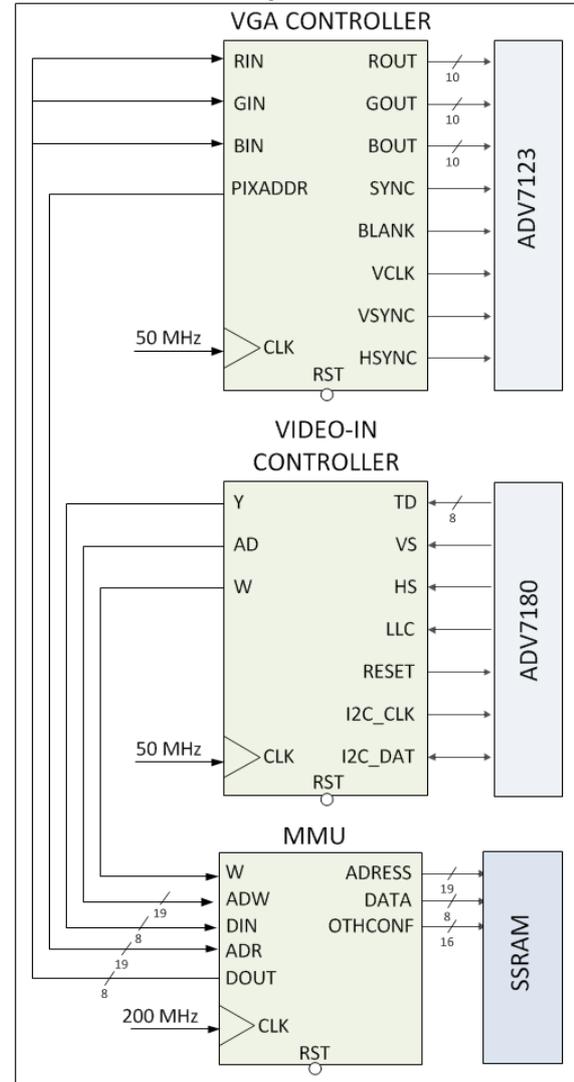


Fig. 3 Diagrama de interconexión de los IP cores.

A continuación se describe cada uno de los IP cores.

### 2.1 MMU

Este IP core es el encargado de manejar la memoria SSRAM, en este caso una ISSI IS61LPS51236A de 2 MB y 200 MHz de velocidad. En la figura 4 se muestra un diagrama de este IP core.

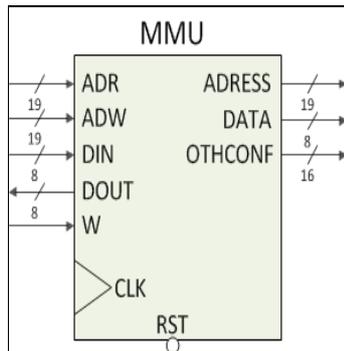


Fig. 4 Diagrama del controlador de memoria.

A continuación se describe cada uno de los puertos:

- ADR*: Dirección de lectura.
- ADW*: Dirección de escritura.
- DIN*: Dato a ser guardado.
- DOUT*: Dato leído.
- W*: Activación de escritura en flanco positivo.
- ADRESS*: Selección de dirección en la SSRAM.
- DATA*: Puerto bidireccional para lectura y escritura en la SSRAM.
- OTHCONF*: Bus que integra la salida para los pines de configuración y reloj de la SSRAM (CE, BWE, OE, etc.).
- CLK*: Entrada de reloj.
- RST*: Reset asíncrono.

La aplicación requiere que los datos puedan ser leídos y grabados a una velocidad mínima de 28 MHz, además de poder escribir y leer al mismo tiempo. La SSRAM empleada solo tiene un puerto de dirección y un puerto bidireccional para los datos, por lo que se desarrolló una estructura digital que integra flip-flops tipo D, un multiplexor y una FSM, con lo que se logra emular una RAM con puerto de lectura y escritura independiente. Para que los datos puedan ser grabados y leídos al mismo tiempo (visto desde fuera del bloque) se requiere una velocidad de reloj interna del bloque 4 veces mayor a la velocidad a la que se va a usar, ya que se usan dos ciclos para la escritura y dos para la lectura. Considerando la futura aplicación del proyecto y la velocidad máxima de operación de la SSRAM, se definió la velocidad de operación a 50 MHz, por lo que el bloque requiere una señal de reloj de 200 MHz, la cual es generada a partir de un PLL integrado en el FPGA, ya que el reloj incluido en la tarjeta de desarrollo es de 50 MHz.

## 2.2 VIDEO-IN CONTROLLER

Este IP core tiene la tarea de configurar el decodificador de video, extraer de éste la información referente al valor de luminancia de cada pixel y

generar la dirección y señal de escritura para su almacenamiento en la memoria. En la figura 5 se muestra un diagrama de este IP core.

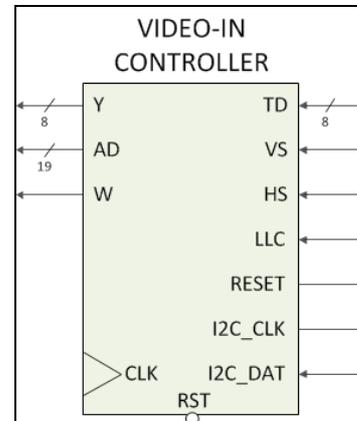


Fig. 5 Diagrama del controlador del decodificador de video.

A continuación se describen cada uno de los puertos:

- Y*: Dato de luminancia.
- AD*: Dirección del pixel a almacenar.
- W*: Señal de activación de escritura.
- TD*: Dato recibido del decodificador.
- VS*: Señal de sincronía vertical.
- HS*: Señal de sincronía horizontal.
- LLC*: Señal de reloj de 27 MHz generada por el decodificador.
- RESET*: Señal de reset para el decodificador.
- I2C\_CLK*: Señal de reloj para configuración I2C.
- I2C\_DAT*: Señal de datos I2C.
- CLK*: Entrada de reloj.
- RST*: Reset asíncrono.

El decodificador de video que se usó es el ADV7180 de Analog Devices, el cuál es un decodificador de señales de televisión de definición estándar, compatible con señales de televisión analógica en estándar NTSC, PAL y SECAM. El decodificador convierte estas señales en datos de componentes de video en formato YCrCb 4:2:2 compatibles con la interfaz estándar de 8-bit ITU-R BT.656 [2]. Estos datos están codificados como se especifica en la recomendación ITU-R BT.601 [3]. El decodificador requiere de un reloj de 28.63 MHz el cual viene incluido en la tarjeta de desarrollo.

Este IP core está formado por dos bloques principales, un bloque configurador y un bloque detector de secuencia y generador de señal de activación y dirección de escritura.

El bloque configurador se encarga de mandar la configuración al decodificador por medio de comunicación con protocolo I2C, esta configuración consta de registros que le indican al decodificador el modo en el que se va a usar, específicamente en este trabajo se configuro para ser usado con entrada de video en estándar NTSC, en modo de video compuesto en el canal AIN1 del decodificador (lugar donde se encuentra conectado el conector RCA de la tarjeta de desarrollo) y con ajustes de crominancia y luminancia automáticos.

Los datos recibidos del decodificador están entrelazados y en formato 4:2:2, por lo que el segundo bloque se encarga de detectar el inicio y fin de cada línea, denotado por la secuencia de datos en hexadecimal "FF,00,00,XY" así como también si es línea par o non. Con las operaciones anteriores se genera la dirección correspondiente a cada pixel (haciendo el desentrelazado), además de generar la señal de activación de escritura que permite guardar únicamente los datos correspondientes a luminancia. En la figura 6 se muestra una parte del diagrama de tiempo de la sincronía horizontal, en donde se puede observar la secuencia de inicio y fin de video activo, así como los datos de luminancia (Y) y crominancia (Cb y Cr).

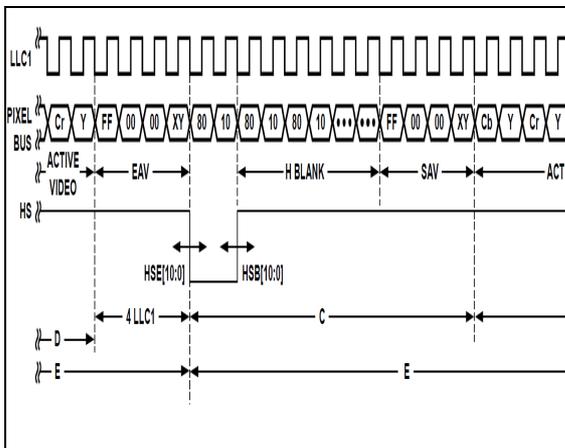


Fig. 6 Diagrama de tiempo de sincronía horizontal.

Este bloque requiere una señal de reloj de 50 MHz la cual se toma directamente del reloj incluido en la tarjeta de desarrollo.

### 2.3 VGA CONTROLLER

Este IP core tiene la tarea de generar las direcciones de lectura, leer los datos, ajustarlos y enviarlos al convertidor digital analógico de video junto con las señales de sincronía necesarias para el

despliegado en monitor VGA. En la figura 7 se muestra un diagrama de este IP core.

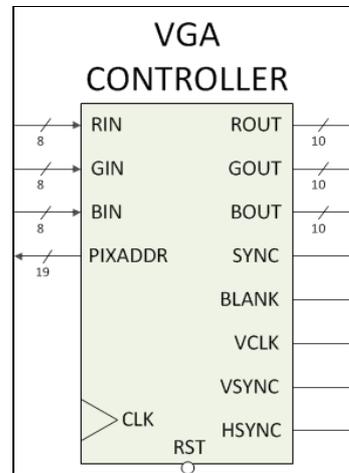


Fig. 7 Diagrama del controlador VGA.

A continuación se describen cada uno de los puertos:

- RIN: Entrada de componente de color rojo.
- GIN: Entrada de componente de color verde.
- BIN: Entrada de componente de color azul.
- PIXADDR: Dirección del pixel siguiente a mostrar.
- ROUT: Salida al DAC de componente de color rojo.
- GOUT: Salida al DAC de componente de color verde.
- BOUT: Salida al DAC de componente de color azul.
- SYNC: Señal de control para DAC.
- BLANK: Señal de control para DAC.
- VCLK: Señal de reloj para DAC.
- VSYNC: Señal de sincronía vertical para VGA.
- HSYNC: Señal de sincronía horizontal para VGA.
- CLK: Entrada de reloj.
- RST: Reset asíncrono.

El estándar VGA requiere de los componentes de color rojo, verde y azul. En esta aplicación la adquisición se realiza en escala de grises, por lo que a cada componente se le asigna el mismo valor de lectura correspondiente al pixel a mostrar.

El convertidor analógico digital usado es el ADV7123 de Analog Devices, el cual integra tres convertidores analógico-digitales de alta velocidad con una resolución de 10-bit.

Este IP core ajusta la longitud de los valores de cada componente, cuya entrada es de 8-bit a 10-bit y se los envía al convertidor, junto con las señales de control SYNC, BLANK y la señal de reloj VCLK. Las señales analógicas provenientes del convertidor

junto con las señales de sincronía VSYNC y HSYNC son transmitidas al monitor VGA para el despliegado de la imagen. Este bloque necesita una señal de reloj de 50 MHz la cual se toma directamente del reloj incluido en la tarjeta de desarrollo.

En [4] se puede encontrar de manera detallada el funcionamiento del estándar VGA así como la implementación en FPGA de un controlador sencillo (solo contiene el generador de las señales de sincronía y de direcciones).

### 3. Resultados

Como resultado se obtuvo la implementación en una tarjeta de desarrollo basada en FPGA de un sistema capaz de adquirir imágenes de 640 x 480 pixeles en escala de grises de 8-bit provenientes de una fuente de video compuesto en estándar NTSC, así como su despliegado en una pantalla con estándar VGA. La prueba del sistema se realizó en el laboratorio de metalografía de la Universidad Autónoma de Querétaro campus San Juan del Río, en donde se conecto a la cámara de un microscopio que contenía una muestra metalográfica, así como a un monitor VGA donde se desplego la imagen. En la figura 8 se muestra una foto del sistema en funcionamiento.

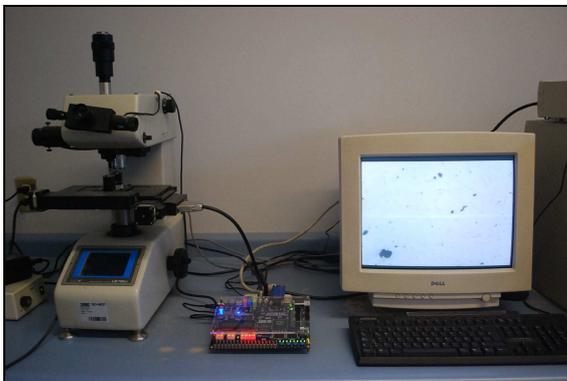


Fig. 8 Prueba del sistema.

### 4. Conclusiones y trabajo futuro

El uso de un sistema de adquisición y despliegado de imágenes basado en tecnología FPGA permite una opción de bajo costo, que gracias al paralelismo de esta tecnología, puede integrar una gran cantidad de tareas en el mismo circuito integrado. Además, gracias a la capacidad de reconfigurabilidad del FPGA, es posible la posterior integración de algoritmos de procesamiento de imagen en el mismo integrado.

Como trabajo futuro se tiene el desarrollo de un sistema de análisis metalográfico completo, que obtenga la imagen directamente de la cámara del microscopio, la procese y despliegue los resultados del análisis en un monitor VGA, reduciendo los costos de adquirir un sistema comercial y mejorando la velocidad con respecto a un análisis basado en PC.

### Referencias

- [1] Romero R. "Electrónica digital y lógica programable", Universidad de Guanajuato, 2ed, 2007.
- [2] Recommendation ITU-R BT.656-5. "Interface for digital component video signals in 525-line and 625-line television systems operating at the 4:2:2 level of Recommendation ITU-R BT.601", International Telecommunication Union, 2007.
- [3] Recommendation ITU-R BT.601-7. "Studio encoding parameters of digital television for standard 4:3 and wide-screen 16:9 aspect ratios", International Telecommunication Union, 2011.
- [4] Chu P. "FPGA PROTOTYPING BY VHDL EXAMPLES", John Wiley and Sons, USA, 1ed, 2008.