UNIVERSIDAD SAN FRANCISCO DE QUITO USFQ

Colegio de Ciencias e Ingenierías

Aplicación de un Algoritmo de Detección de Formas basado en Momentos de Hu en un FPGA para Imágenes en Escala de Grises y Color en Formato 720p HD con una interfaz HDMI en 1080pFull HD

Raúl André Borja Cajiao
William Felipe Toscano Acosta

Ingeniería Electrónica

Trabajo de integración curricular presentado como requisito
para la obtención del título de
Ingeniero Electrónico

Quito, 12 de mayo de 2020

UNIVERSIDAD SAN FRANCISCO DE QUITO USFQ

Colegio de Ciencias e Ingenierías

HOJA DE CALIFICACIÓN
DE TRABAJO DE INTEGRACIÓN CURRICULAR

Aplicación de un Algoritmo de Detección de Formas basado en Momentos de Hu en un FPGA para Imágenes en Escala de Grises y Color en Formato 720p HD con una interfaz HDMI en 1080pFull HD

Raúl André Borja Cajiao
William Felipe Toscano Acosta

Calificación:

Nombre del profesor, Título académico **Luis Miguel Prócel, Ph.D.**

Firma del profesor: _____

Nombre del profesor, Título académico **Ramiro Taco, Ph.D.**

Firma del profesor: _____

Quito, 12 de mayo de 2020

DERECHOS DE AUTOR

Por medio del presente documento certifico que he leído todas las Políticas y Manuales de la Universidad San Francisco de Quito USFQ, incluyendo la Política de Propiedad Intelectual USFQ, y estoy de acuerdo con su contenido, por lo que los derechos de propiedad intelectual del presente trabajo quedan sujetos a lo dispuesto en esas Políticas.

Asimismo, autorizo a la USFQ para que realice la digitalización y publicación de este trabajo en el repositorio virtual, de conformidad a lo dispuesto en el Art. 144 de la Ley Orgánica de Educación Superior.

Firma del estudiante: _____

Nombres y apellidos: Raúl André Borja Cajiao

Código: 00131410

Cédula de identidad: 172264327-5

Lugar y fecha: Quito, mayo de 2020

Firma del estudiante: _____

Nombres y apellidos: William Felipe Toscano Acosta

Código: 00132210

Cédula de identidad: 172593367-3

Lugar y fecha: Quito, mayo de 2020

ACLARACIÓN PARA PUBLICACIÓN

Nota: El presente trabajo, en su totalidad o cualquiera de sus partes, no debe ser considerado como una publicación, incluso a pesar de estar disponible sin restricciones a través de un repositorio institucional. Esta declaración se alinea con las prácticas y recomendaciones presentadas por el Committee on Publication Ethics COPE descritas por Barbour et al. (2017) Discussion document on best practice for issues around theses publishing, disponible en http://bit.ly/COPETheses.

UNPUBLISHED DOCUMENT

Note: The following capstone project is available through Universidad San Francisco de Quito USFQ institutional repository. Nonetheless, this project – in whole or in part – should not be considered a publication. This statement follows the recommendations presented by the Committee on Publication Ethics COPE described by Barbour et al. (2017) Discussion document on best practice for issues around theses publishing available on http://bit.ly/COPETheses.

RESUMEN

En el presente trabajo se muestra la implementación y adaptación de un algoritmo rápido de detección de formas en la plataforma FPGA de Xilinx VC707 VIRTEX-7. Se realizó el diseño de hardware para implementar de forma correcta el algoritmo utilizando lenguaje de descripción de hardware (HDL) y bloques de propiedad intelectual (IP). Se desarrolló las configuraciones necesarias, específicas de la plataforma seleccionada, para obtener una salida de video para imágenes en formato RGB 4:4:4 con una profundidad de color por pixel de 36 bits a través de una interfaz HDMI con la cual mostrar la imagen resultante de la aplicación del algoritmo. Se llevó a cabo de forma correcta la implementación de un sistema capaz de aplicar el algoritmo a imágenes en una resolución máxima de 1080p-60 full-HD. El algoritmo se probó de forma satisfactoria en imágenes con una resolución 720p-HD en formato blanco y negro, escala de grises y color. Se demostró que el sistema funciona de forma óptima en todas las pruebas sin necesidad de modificar el algoritmo original a pesar de las variaciones en resolución o formato de las imágenes.

Palabras clave: FPGA, HDL, Full-HD, IIC, ADV7511, Detección de Formas, HDMI, Escala de Grises, RGB.

ABSTRACT

The present work shows the implementation and adaptation of a fast shape recognition algorithm on the Xilinx VC707 VIRTEX-7 FPGA platform. The hardware design was developed to correctly implement the algorithm using hardware description language (HDL) and intellectual property blocks (IP). The necessary configurations, specific for the selected platform, were developed to obtain a video output for images in RGB 4:4:4 format with a color depth per pixel of 36 bits through an HDMI interface to display the resulting image from the application of the algorithm. It was correctly done the implementation of a system capable of applying the algorithm to images in a maximum resolution of 1080p-60 full-HD. The algorithm was successfully tested on images in 720p-HD resolution in black and white, grayscale, and color formats. The system performance was optimal in all tests without requiring to modify the original algorithm despite variations in resolution or image format.

Key words: FPGA, HDL, Full-HD, IIC, ADV7511, Shape Recognition, HDMI, Gray Scale, RGB.

TABLA DE CONTENIDO

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

INTRODUCCIÓN

El desarrollo acelerado de la tecnología en las últimas décadas ha permitido aplicar e implementar algoritmos inspirados en la teoría de procesamiento de imágenes en sistemas embebidos que automatizan ciertas tareas y facilitan la vida del ser humano. Algunos de estos algoritmos tienen la función de reconocer o detectar de formas. A través estos algoritmos se ha conseguido desarrollar sistemas de reconocimiento cuyas aplicaciones generan beneficios a pequeña y gran escala en ámbitos de seguridad, movilidad, ciencia e investigación. Sin embargo, la aparición de nuevos dispositivos, sistemas embebidos y tecnología en general ha exigido que estos sistemas de detección de forma sean capaces de desarrollarse a la par y ofrecer nuevas o mejores características.

Una de las grandes problemáticas en la implementación de algoritmos de detección de formas es que estos necesitan de sistemas embebidos que demandan muchos recursos electrónicos. Esto hace que dichos sistemas sean muy costosos y que gran parte de sus recursos sean inutilizados ya que son sistemas embebidos diseñados para múltiples propósitos y no solo para correr estos algoritmos. No obstante, se ha visto en los FPGAs la oportunidad de implementar algoritmos de detección de formas en sistemas embebidos novedosos y que se encuentran en constante desarrollo. El mayor reto al implementar estos algoritmos en dichas plataformas es adaptar los sistemas a los estándares más recientes de video y garantizar la mayor calidad posible en los resultados obtenidos al menor costo.

Este trabajo muestra la implementación de un algoritmo rápido de detección de formas. En este escrito se explican las modificaciones necesarias para lograr que un algoritmo detección de formas funcione dentro de la plataforma de Xilinx VC707 VIRTEX-7 basada en el componente FPGA XCV7VX485T-2FFG1761C y se pueda adaptar a estándares de alta definición para

imágenes. En las siguientes secciones se explicará una breve introducción al algoritmo implementado y la manera en que el FPGA puede resolver las necesidades de este. Se explicará como el algoritmo puede ser implementado a través de máquinas de estado conectadas a módulos combinacionales que realizan las operaciones lógicas y matemáticas propias de los cálculos del algoritmo.

Al seguir avanzando dentro del texto se encontrará la generación de ejemplos y la implementación en hardware que requiere la configuración de componentes complementarios dentro de la tarjeta para garantizar una imagen de salida en formato 1080p-60 full-HD. La última sección de la investigación permitirá observar los resultados de la implementación y aplicación del algoritmo a imágenes de alta definición en escala de grises y color. Se comprobará si el algoritmo es capaz de funcionar en un FPGA y de garantizar la detección de formas en imágenes con las características mencionadas sin necesidad de modificar la naturaleza del algoritmo original.

DESARROLLO DEL TEMA

Algoritmo Rápido de Detección de Formas

Para realizar el reconocimiento de formas se utilizan los momentos de Hu que son 7 momentos espaciales invariantes a la traslación, rotación y escalamiento en una imagen. Con estos valores se logra parametrizar un objeto y así identificarlo. Este método calcula los momentos dentro del logo que se busca detectar y los compara con los momentos obtenidos en la imagen de prueba (Pratt, 2007). Para disminuir el tiempo de respuesta del dispositivo se utilizaron los primeros dos momentos.

Para la detección de objetos se utilizan logos ubicados en vecindarios de acuerdo con el algoritmo implementado y por lo tanto la imagen se ajusta a los parámetros de prueba. La memoria dentro del FPGA tiene los datos de la imagen en un solo vector por lo que no puede determinar la ubicación de un píxel en el espacio. Para esto se utiliza un algoritmo que devuelve la posición del primer píxel de cada vecindario dentro del vector proporcionado.

Figura 1 Vecindarios de 60x60 pixeles

En la Figura 1 se coloca un ejemplo simplificado del funcionamiento del algoritmo, la flecha gris muestra la posición del píxel necesaria para calcular los momentos de Hu dentro de cada vecindario. El barrido se lo realiza por filas cogiendo el primer píxel de cada vecindario, para

esto el algoritmo suma la posición anterior más los 60 del ancho del logo y 20 del espacio que existe entre vecindarios. Al finalizar el barrido de la fila de vecindarios el algoritmo salta al primer vecindario de la siguiente fila, para esto el algoritmo tiene que pasar del píxel 160 al 19200. Este ejemplo muestra el funcionamiento del algoritmo y se lo puede ampliar para imágenes en alta definición, ya que se puede aumentar el número de vecindarios.

Generación de archivos de coeficientes en MATLAB

Los archivos COE se los genera en MATLAB a partir de una imagen png con 24 bits de profundidad, de esta manera, obtenemos una imagen RGB 888, un estándar que devuelve ocho bits por color.

Durante todo el proceso se realizaron varios tipos de archivos dependiendo de las necesidades de las pruebas. Para esto se hicieron varias modificaciones en los parámetros de las memorias explicados en la guía Block Memory Generator v8.2. Los archivos COE tienen dos parámetros que rigen la forma en la que interpretan los datos y, por lo tanto, como se crean las memorias IP: memory_initialization_radix y memory_initialization_vector. El primer parámetro define el sistema de numeración en el que se encuentra el archivo. El segundo parámetro define el contenido de la memoria (Xilinx, 2015).

Para los ensayos se utilizaron los siguientes valores en memory_initialization_radix:

- Pruebas en blanco y negro: 2
- Pruebas en escala de grises: 16 o 10
- Pruebas a color: 16

Cada valor define un sistema de numeración, binario es 2, decimal es 10 y hexadecimal es 16. Para las pruebas en escala de grises y a color se utilizó principalmente el sistema hexadecimal para poder verificar el número de bits debido a la profundidad utilizada.

Después de definir los parámetros se genera el contenido del segundo parámetro. MATLAB genera los valores decimales de la imagen y se realizan diferentes procedimientos dependiendo de las pruebas a realiza.

Pruebas a color: MATLAB a partir de la imagen png genera tres matrices con formato uint8, cada color se representa en una matriz. La imagen se puede ampliar a 36 bits de profundidad, se puede mantener en los 24 bits de la imagen original y también se puede reducir el tamaño de la imagen a otros formatos. Las pruebas se realizaron con 12 bits de profundidad por lo que se pasó de un valor máximo de 255 a 15. Los archivos COE manejan un solo vector por lo que se toman los datos por filas y para cada píxel se coloca la información con los primeros bits para rojo, los siguiente para verde y azul.

Pruebas a escala de grises: Para estas pruebas el archivo COE almacena los datos de una imagen con una profundidad de 36 bits. Esto se lo realiza utilizando la función rgb2gray que nos devuelve la matriz de luminancia de la imagen. Se realiza el mismo proceso para guardar la imagen en el formato COE.

Pruebas en blanco y negro: Estas pruebas utilizan la matriz de luminancia descrita anteriormente. En este caso se divide en dos grupos píxeles con alta luminancia y baja luminancia, este parámetro se selecciona con el criterio de mitad y mitad. Esto sería igual a escoger únicamente el bit más significativo de cada píxel de la matriz de luminancias.

Las pruebas se realizaron empezando por las imágenes en blanco y negro, a medida que se avanzaba hacia la prueba a color se modificaba el código en el FPGA para que este disponga de la imagen en escala de grises utilizando la ecuación de luminancia (1) (Pratt, 2007) y también una imagen binaria.

$$Y = 0.3R + 0.59G + 0.11B \quad (1)$$

La generación de ejemplos utiliza un algoritmo aleatorio para escoger el logo y su rotación. Estos ejemplos ayudan a realizar las pruebas de funcionamiento del algoritmo de detección de logos que se implementa en el FPGA. De esta manera logramos probar la detección del logo en cualquiera de los vecindarios y muestra que la detección del logo es independiente de su rotación. Los ejemplos devuelven una imagen HD con fondos y logos a color para probar la salida HDMI en color.

La ubicación de los logos se realiza dependiendo de la ubicación de los vecindarios en la imagen. Esto está determinado por el algoritmo implementado en el FPGA que determina la ubicación de cada vecindario. Durante este trabajo se utilizó vecindarios de 60x60 píxeles con una separación de 20 píxeles entre cada vecindario. Estas medidas logran que el número de vecindades que puede entrar por cada fila y columna tenga relación con los tamaños estándares para pantallas. Para el caso de 720p tenemos que se puede llegar a un máximo de 16 vecindarios a lo ancho de la imagen y 9 en la parte vertical de la imagen, esto nos da un total de 144 vecindarios en las pruebas de 720p.

Para la ubicación de logos dentro de las imágenes se actualiza los valores dentro de la matriz de la imagen de fondo invirtiendo los pixeles. De esta forma se realiza un barrido por columnas y por filas hasta completar con el total de logos solicitados.

Implementación en Hardware

Se utilizó la plataforma de Xilinx VC707 VIRTEX-7 basada en el componente FPGA XCV7VX485T-2FFG1761C para implementar el algoritmo de detección de formas. La herramienta utilizada para la descripción de hardware fue el software Vivado 2015.1. Se utilizó el ambiente de desarrollo Xilinx SDK 2015.1, compatible con la versión de Vivado, para la programación de software. El algoritmo fue implementado con un reloj sincrónico de 50 MHz. Dicho reloj cumple con las restricciones de tiempo para que el algoritmo se ejecute

correctamente, además es el reloj que se recomienda y utiliza en investigaciones anteriores basadas en el mismo algoritmo de detección de formas (Borja et al., 2019).

La parte algorítmica está compuesta por dos máquinas de estados representadas en las figuras 2 y 3. Se mantuvo la naturaleza de las máquinas de estado tal como se muestran en (Borja et al., 2019). El algoritmo original implementado en hardware estuvo diseñado para trabajar con imágenes con resoluciones específicas y en blanco y negro. Por lo que, para la aplicación del algoritmo con imágenes en alta resolución y resolución variable, en escala de grises y color se aplicaron trasformaciones y algoritmos fuera del sistema original de detección de formas implementado con las dos máquinas de estado.

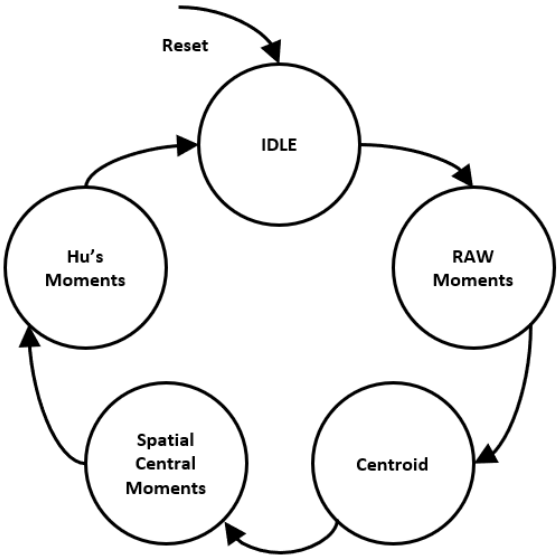


Figura 2 Máquina de estados 1

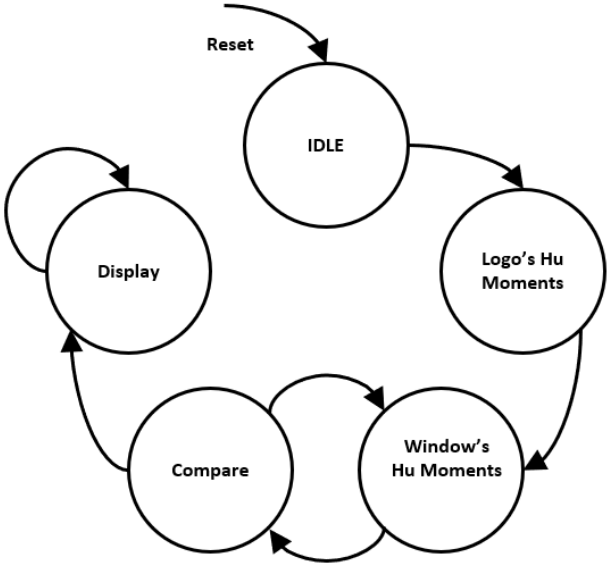


Figura 3 Máquina de estados 2

El algoritmo implementado trabaja de forma que la máquina de estados 1 en el segundo y tercer estado en la máquina de estados 2. De esta forma se obtienen los momentos de Hu del logo y de la imagen de prueba para poder compararlos y finalmente mostrarlos en una pantalla. Estas dos máquinas de estado se encuentran implementadas en hardware en el bloque "Algorithm" que se observa en la Figura 4. Sin embargo, se requiere de hardware adicional para ejecutar el algoritmo y mostrar los resultados.

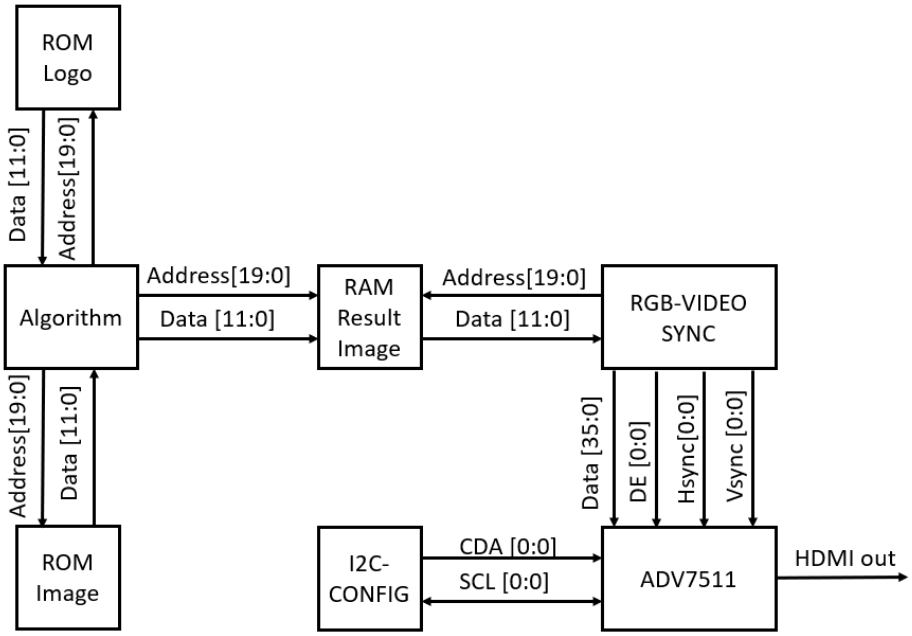


Figura 4. Diagrama de Bloques del Sistema de Detección de Formas

Se modificó el hardware para trabajar con imágenes de prueba de alta resolución, logos de tamaño variable, aplicados a imágenes en escala de grises y color. El algoritmo requiere de tres memorias. Dos memorias son de sólo lectura (memorias ROM) para almacenar el logo y la imagen de prueba. La tercera memoria es tipo RAM y es aquella en la que se escribe la imagen resultante para ser leída y mostrada en una pantalla. El hardware se modificó para soportar imágenes con una profundidad de bits por píxel variable, sin embargo, en la Figura 4 se muestran 12 bits por píxel ya que es la información utilizada en las pruebas en escala de grises y color.

Los bloques de configuración por IIC (I2C-CONFIG) y de sincronización de video (RGB-VIDEO SYNC) son nuevos en comparación con el sistema original (Borja et al., 2019). El bloque de configuración por IIC permite modificar los registros del chip ADV7511 para poder mostrar el resultado por una interfaz HDMI y en un formato 1080p-60 full-HD. El bloque de

sincronización de video genera todas las señales de sincronización en función del formato utilizado y transmite la imagen resultante en formato RGB 4:4:4 con una profundidad de 36 bits por píxel, 12 por cada color.

Configuración IIC para la interfaz HDMI

La plataforma FPGA utilizada en este proyecto permite la salida de imágenes a través de una interfaz de multimedia de alta definición (HDMI). La tarjeta VC707 utiliza un transmisor de HDMI de alta velocidad y bajo consumo de potencia integrado en el chip ADV7511 (Xilinx, 2019). Este transmisor genera las señales de la interfaz HDMI a partir de señales proporcionadas por el usuario. El ADV7511 es capaz de soportar múltiples formatos de video, alcanzando una definición máxima de 1080p full-HD a 60 hercios (Analog Devices, 2011). El objetivo de este proyecto fue alcanzar dicha definición para tener la capacidad de probar el algoritmo en imágenes de alta resolución.

Gracias al ADV7511, la tarjeta VC707 es capaz de soportar un formato de imagen RGB 4:4:4 con una profundidad de color por pixel desde 24 hasta 36 bits (Analog Devices 2011). En este proyecto se utilizó la profundidad máxima de 36 bits. En esta investigación se proveyó al transmisor de los 36 bits de información por pixel y de un reloj de 148.5 megahercios el cual indica al chip cuando coger cada dato de cada pixel. Dicho reloj tiene un desface de 180 grados respecto al reloj de 148.5 megahercios que genera los 36 datos, así se cumplen las restricciones de tiempo. Para producir las señales de la interfaz HDMI es necesario proveer al ADV7511 de señales de sincronización (Vsync, Hsync y DE). Dichas señales pueden ser generadas en el chip o producidas por el usuario. En el proyecto se probó ambas configuraciones obteniendo resultados satisfactorios en los dos casos.

El transmisor ADV7511 debe ser configurado a través del protocolo síncrono IIC en función del formato de entrada y salida de video, además, existen configuraciones básicas que son

obligatorias. La configuración de registros del ADV7511 en la plataforma FPGA VC707 no puede realizarse de forma sencilla mediante Lenguaje de Descripción de Hardware (HDL), por lo tanto, en este proyecto se utilizó adicionalmente el Kit de Desarrollo de Software (SDK) de Xilinx compatible con Vivado 15.1. La configuración IIC del ADV7511 para la interfaz HDMI se hizo combinando diseño en Hardware a través de bloques de IP en Vivado y diseño en Software con la herramienta SDK.

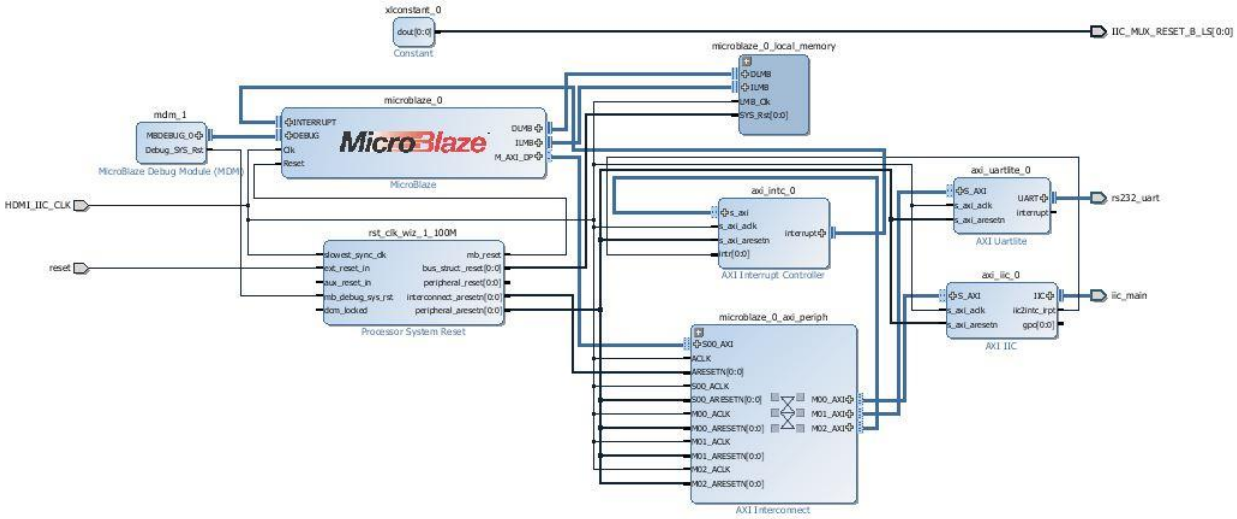


Figura 5. Diseño de hardware para la configuración IIC del ADV7511

La primera parte de la configuración del ADV7511 se la hace mediante bloques de IP en Vivado tal como lo muestra la Figura *5*. Las partes principales del hardware están compuestas por la unidad microprocesadora para FPGAs: Microblaze; y los módulos IP para interfaz UART y IIC: AXI IIC y AXI Uarlite respectivamente. El módulo AXI IIC fue diseñado para trabajar con interrupciones por lo que necesita de un microprocesador para atender dichas señales (Xilinx, 2016a). A través de AXI IIC se produce el hardware necesario para generar las señales del protocolo IIC. El microprocesador Microblaze es el encargado de producir y controlar las señales de todos los módulos, además requiere una unidad de memoria para almacenar la configuración de los registros para el ADV7511 (Xilinx, 2016b).

El segundo módulo de comunicación serial es el AXI Uartlite que, a pesar de no ser imprescindible para la configuración IIC, permite tener una retroalimentación de la configuración del ADV7511 y verificar que los registros hayan sido configurados correctamente. A través de este bloque se puede enviar por UART a una computadora los valores escritos en los registros del ADV7511 luego de ser leídos por el AXI IIC (Xilinx, 2017). De esta forma se tiene garantiza que todas las configuraciones se realizaron exitosamente. Para leer la información transmitida por UART del VIRTEX-7 es necesario el uso de un emulador de terminal de computadora. En este caso, se utilizó la terminal de Tera Term con una tasa de baudios de 9400 bits por segundo, ya que es la tasa con la que se configuró el bloque AXI Uartlite.

Muchos bloques del hardware implementado requieren de un reloj sincrónico. El reloj más restrictivo es el del bloque AXI IIC el cual puede oscilar entre los 25 megahercios hasta los 300 megahercios (Xilinx, 2016a). En este proyecto se trabajó en base a un reloj de 148.5 megahercios el cual es el mismo utilizado en el hardware implementado para generar las señales de sincronización de video. Adicionalmente se implementó una constante con valor de 1 a través de un bloque IP. La función de esta constante es producir una señal de reset necesaria en la plataforma VC707 para acceder a la configuración IIC del ADV7511 (Xilinx, 2019). Una vez que el hardware fue diseñado, implementado y sintetizado, fue exportado para poder ser usado en el SDK de Xilinx.

A través de SDK se programó el software para el microprocesador Microblaze. Mediante funciones basadas en Lenguaje de C se realizó un programa para configurar los registros del ADV7511 con los valores necesarios para transmitir una imagen en 1080p-60 full-HD. Dichas configuraciones se encuentran en la Tabla *1*. Las configuraciones básicas fueron programadas para realizarse primero y garantizar la correcta escritura de las configuraciones de video tal

como recomienda el fabricante del transmisor (Analog Devices, 2012). El programa diseñado permite realizar una escritura de prueba a un registro del ADV7511 y luego leer dicho registro para verificar que la escritura fue exitosa realizando la configuración básica y de video. Finalmente, se leen todos los registros escritos y se los imprime en la terminal para su verificación. Acabado este punto, el VIRTEX-7 está en capacidad de transmitir imágenes mediante su interfaz HDMI.

Tabla 1. Configuración básica y de video para el ADV7511

ADV7511 Register Configuration			
Settings	**Address**	**Value**	**Function**
Basic	0x41	0x10	Turn-on the chip
	0x98	0x03	Proper Operation
	0x9A	0xE0	Proper Operation
	0x9C	0x30	Proper Operation
	0x9D	0x61	Proper Operation
	0xA2	0xA4	Proper Operation
	0xA3	0xA4	Proper Operation
	0xE0	0xD0	Proper Operation
	0xF9	0x00	Fixed IIC Address
Video Input	0x15	0x00	Video Format: RGB 4:4:4 (36-bits)
	0x16	0x20	Color Depth: 12bits/color
	0x17	0x02	Aspect Ratio: 16:9
Video Output	0x16	0x20	Output Format: 4:4:4
	0x18	0x46	CSC: Disabled
	0xAF	0x06	Output Mode: HDMI
	0x40	0x00	Deep Color Convertion: Disabled
	0x4C	0x06	Output Color: 36bits/pixel

Implementación para Imágenes con Resolución HD

En esta investigación se trabajó en base al algoritmo rápido de detección de formas mostrado en una investigación previa (Borja et al., 2019). En dicha publicación se menciona un algoritmo capaz de trabajar en plataformas FPGAs con una resolución de imagen desde 155x155 hasta

310x310 pixeles con una imagen de logo de 60x60 pixeles. El algoritmo fue adaptado para poder ser aplicado en imágenes de alta resolución con una definición máxima de 1920x1080 pixeles. Esto fue posible gracias a los recursos del VC707 que lo convierten en un FPGA de alto rendimiento capaz de trabajar con imágenes de alta definición. Esta adaptación permite aplicar el algoritmo a imágenes con largos y anchos independientes generando una infinidad de combinaciones en la resolución de la imagen. A través de HDL también se modificó el hardware para que el algoritmo pueda funcionar con cualquier tamaño de logo sin afectar su comportamiento.

El algoritmo de detección fue implementado de forma que las vecindades de la imagen de prueba puedan ser tomadas de forma automática. En (Borja et al., 2019) las vecindades de pixeles a las cuales se aplica la detección de formas son tomadas de forma manual limitando su cantidad a 4 o 16 vecindades únicamente. En esta investigación se desarrolló una "función" parametrizada en HDL capaz de tomar cualquier número de vecindades limitado únicamente por la resolución de la imagen y del logo. De esta forma, el usuario puede, a través de simples parámetros establecer la cantidad de vecindades que requiera tomar y en que ubicación de la imagen de prueba hacerlo. Esta mejoría permite que el algoritmo no tenga que ser siempre calculado en las mismas posiciones dentro de una imagen, permitiendo su futuro uso en aplicaciones de tiempo real con cámaras de video.

En esta investigación se trabajó con la máxima resolución disponible en la plataforma VC707 (1080p-60 full-HD). Se utilizó un formato de imagen RGB 4:4:4 (36-bits), este formato es el máximo permitido por el transmisor ADV7511 y que garantiza la mayor profundidad de bits por píxel (12 bits por píxel) (Analog Devices 2012). Los datos por pixel se transmiten con un reloj de 148.5 megahercios que corresponde con la resolución 1080p para tener una taza de

refresco de 60 hercios. Para poder transmitir imágenes con la resolución y formato requeridos es necesario proveer al ADV7511 de señales de sincronización a través de la plataforma VC707. Las señales de sincronización de video necesarias son Vertical Synchronization, Horizontal Synchronization y Data Enable (Jack, 2008) . En la Tabla 2 se muestran los parámetros que rigen el tiempo y comportamiento de las señales de sincronización según el estándar 1080p-60 full-HD (Ogier & Schmitz, 2004). Estos parámetros se compararon con los valores recomendados por el fabricante del transmisor para el estándar mencionado (Analog Devices, 2012).

Tabla 2. Sincronización para una Resolución 1080p-60 full-HD

	1080p-60 full-HD Synchronization Signals				
	Vertical Synchronization		**Horizontal Synchronization**		**Data Enable**
	Pixels Amount	**VSync Value**	**Pixels Amount**	**HSync Value**	**DE Value**
Active Region	1080	0	1920	0	1
Front Porch	4	0	88	0	0
Sync Width	5	1	44	1	0
Back Porch	36	0	148	0	0
Total Pixels	1125	-	2200	-	-

El ADV7511 tiene la capacidad de recibir las señales de sincronización de forma externa o puede recibir solo algunas de ellas y generar el resto de forma automática (Analog Devices, 2012). En este proyecto se probó ambas configuraciones para verificar mejorías en la calidad de la imagen de una respecto a la otra. Para realizar una sincronización externa, las tres señales de sincronización fueron generadas a través de HDL, considerando las restricciones temporales

que se pueden entender a través de la Figura 2, después se implementaron en el FPGA para que sean transmitidas al ADV7511.

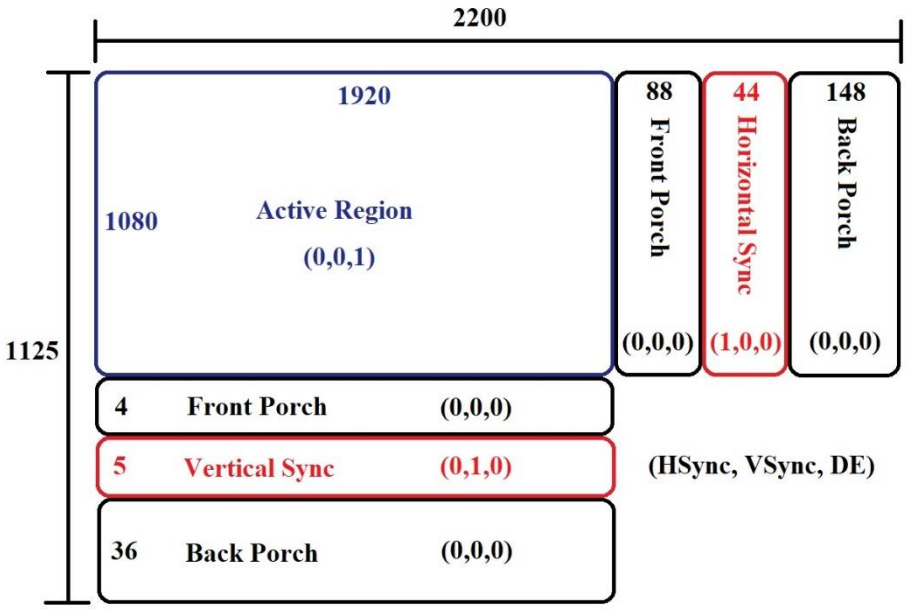


Figura 6. Señales de Sincronización según la región de video

El modo automático de sincronización de imagen requiere de una combinación de hardware con programación en software. Si bien este método implica un paso adicional, el hardware implementado es más sencillo de describir y no se requiere de una comprensión total del comportamiento temporal de la sincronización de imagen. Para esta sincronización se generó únicamente la señal de sincronización vertical y horizontal (Vsync, Hsync). La señal que habilita la entrada de datos (DE) fue generada por el mismo transmisor luego de la modificación de registros adicionales, dicha configuración se hizo en SDK. La configuración adicional se muestra la Tabla 3, la función de este cambio es la de proveer de todas las características temporales que rigen la sincronización para que el transmisor decida de forma automática el mejor comportamiento que debe tener la señal DE. Una vez que la señal DE fue generada, las

señales Vsync, Hsync son ajustadas o generadas nuevamente según sugiere el fabricante del transmisor(Analog Devices, 2012). Este último paso no es imprescindible, pero es una buena práctica que garantiza una imagen con resultados óptimos.

Tabla 3. Configuración adicional para una Sincronización Automática

ADV7511 Register Configuration			
Settings	**Address**	**Value**	**Function**
DE Generation	0x17	0x03	DE Generation: Enabled
	0x35	0x2F	Hsync Back Porch: 148
	0x36	0xE9	Vsync Back Porch: 36
	0x37	0x0F	Active Width: 1920
	0x38	0x00	
	0x39	0x43	Active Height: 1080
	0x3A	0x80	
Sync Adjustment	0x17	0x03	Hsync, Vsync polarity: High for both
	0xD0	0x30	Sync pulse: DE
	0xD7	0x16	Hsync Front Porch: 88
	0xD8	0x02	Hsync Width: 44
	0xD9	0xC0	Vsync Front Porch: 4
	0xDA	0x10	Vsync Width: 5
	0xDB	0x05	

La imagen de salida del ADV7511 se encuentra en resolución 1080p-60 full-HD, sin embargo, en el proyecto no se utiliza todo el espacio de esta imagen. Se decidió trabajar con una imagen de prueba de 1280x720 pixeles equivalente a una resolución 720p-HD (Ogier & Schmitz, 2004). Los pixeles restantes, debido a que la salida es de mayor resolución, se configuraron con color negro en todos los casos. Esto se hizo debido a las limitaciones de memora BRAM disponible en el VC707, dado que las imágenes en alta resolución demandan más espacio de memoria. Además de esta forma, se reduce el tiempo de síntesis e implementación del proyecto y facilita las pruebas.

Aplicación en Imágenes en Escala de Grises

Luego de probar el algoritmo en imágenes en blanco y negro, se modificó el hardware para soportar el análisis de imágenes en escala de grises manteniendo el desempeño y confiabilidad del algoritmo. Este cambio implicó el aumento de la profundidad de bits por píxel de 1 (imágenes en blanco y negro) a 12 bits. Gracias a que el número de bits por pixel en las imágenes en escala de grises coincide con el número de bits por color (12 bits por color), se asignó estos bits a cada color para mantener los 36 bits de profundidad por pixel. De esta manera, se garantiza que no exista pérdida de definición.

El algoritmo original trabaja con un solo bit por píxel para ejecutar los cálculos. Con el objetivo de mantener el diseño del algoritmo, este no fue modificado para trabajar directamente con los 12 bits por píxel en los cálculos. Antes de la aplicación del algoritmo se realizó una transformación a través de HDL que permite producir 1 bit a partir de los 12 bits por pixel. Este cambio se basa en transformar la imagen de escala de grises a blanco y negro, por este motivo se obtiene un solo bit por píxel. Para conseguir esto, los pixeles con luminancia mayor a la mitad del valor máximo se aproximaban a 1 (color blanco), las restantes se aproximaban a 0 (color negro). De esta forma se puede determinar si el algoritmo es capaz de funcionar a pesar de esta transformación.

Aplicación en Imágenes a Color

El siguiente paso luego de aplicar el algoritmo en imágenes a blanco y negro, fue hacerlo en imágenes a color. Para reducir la complejidad del hardware y los tiempos entre pruebas, así como para cumplir con las limitaciones de memoria del dispositivo, se utilizó una profundidad de 12 bits por píxel, donde cada color tenía una definición de cuatro bits. Debido a que la

profundidad por píxel en la imagen de salida del HDMI era de 36 bits, debía hacerse una transformación. La primera alternativa era llenar los 8 bits menos significativos de cada color con ceros, sin embargo, esta aproximación, a pesar de ser aceptable, altera los colores de la imagen de salida. Por lo tanto, se realizó una transformación de los 4 bits por color a su equivalente en 12 bits. De esta forma, las imágenes tienen una menor definición, pero sin alterar los colores.

Como en las pruebas anteriores se procuró mantener el algoritmo de detección intacto por lo que las operaciones se siguieron realizando con 1 bit por pixel tanto para el logo como para la imagen de prueba, considerando que ambas imágenes se encontraban a color. Se utilizó la ecuación (1) para obtener la luminancia por píxel en función de la intensidad de colores RGB. La luminancia es equivalente a la imagen en escala de grises por lo que después de este paso, se transformó el resultado a blanco y negro tal como se hizo en la imagen en escala de grises. El objetivo de este procedimiento es determinar si el algoritmo sigue siendo robusto y produciendo resultados positivos.

RESULTADOS

Verificación de la Configuración IIC de la Interfaz HDMI

A través de la retroalimentación del VC707 por UART hacia la terminal de computadora fue posible verificar el contenido de los registros escritos en el ADV7511. En la Figura 6 se evidencia que todas las pruebas de comunicación IIC fueron exitosas, dando paso a la escritura de los registros del transmisor. Se verificó que todos los valores correspondan con aquellos de la Tabla 2 y 3 obteniendo resultados correctos en todos los casos. A través del registro 0x41 con un valor de 0x10 se determinó que el chip se encendió correctamente y estuvo listo para transmitir video por HDMI.

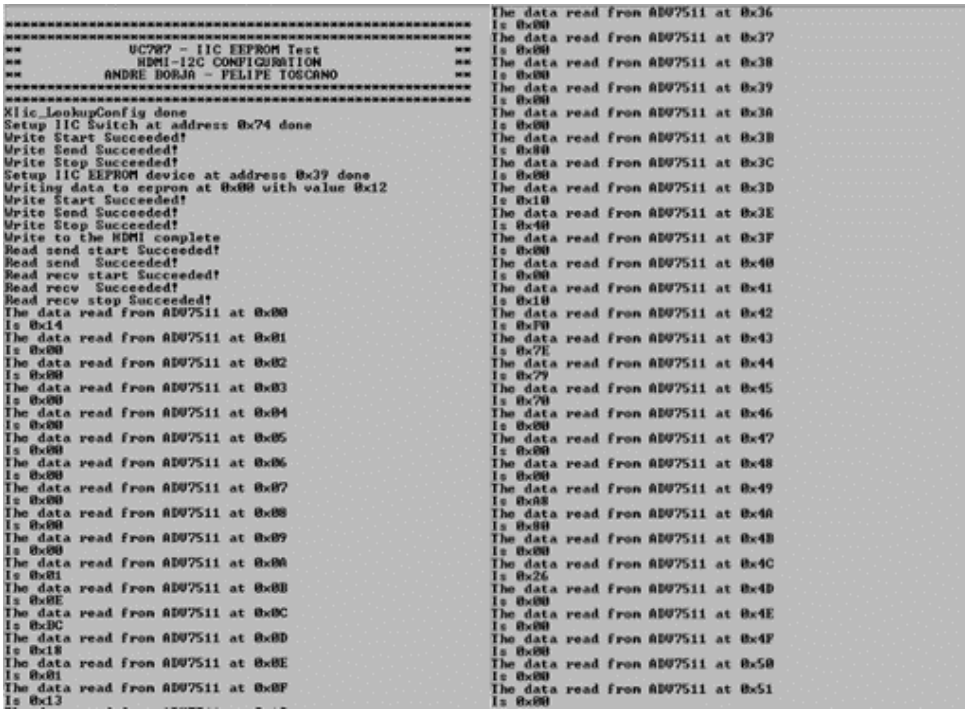
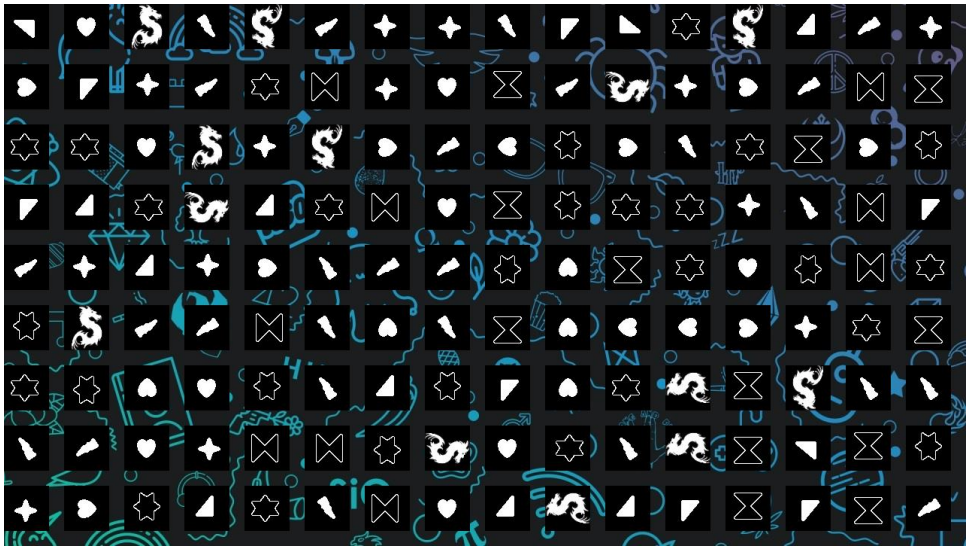


Figura 7. Verificación de Registros del ADV7511 en Tera Term

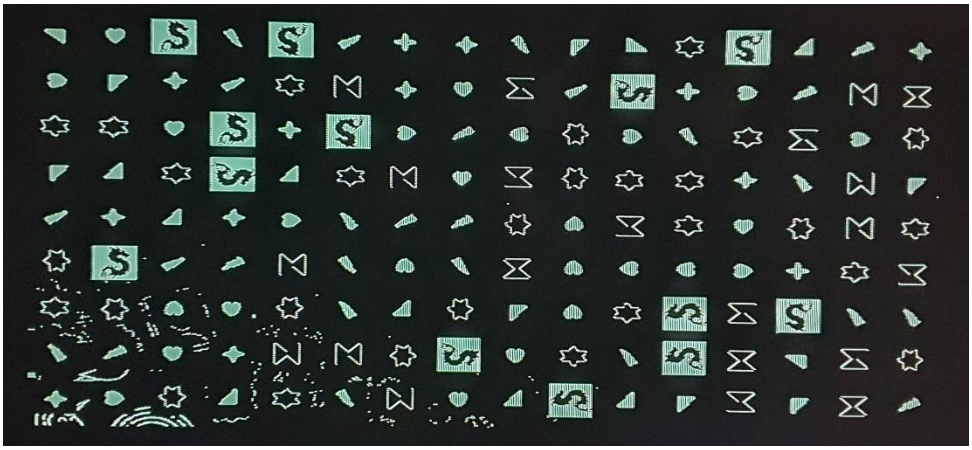
Detección de formas en Imágenes en Blanco y Negro

El algoritmo de detección de formas fue probado inicialmente con imágenes en blanco y negro. Se utilizó un logo de 60x60 pixeles y una imagen de prueba de 1280x720 pixeles con un total de 144 figuras como se muestra en la Figura 7. Dichas figuras fueron diferentes en tamaño, forma y relleno para aumentar la dificultad de detección para el algoritmo. Se observó que el algoritmo funciona correctamente y es capaz de detectar el logo en todos los casos incluso cuando este se encuentra rotado. Esto demuestra que la propiedad de invariabilidad frente a rotaciones de los momentos de Hu se está aplicando en el algoritmo.

a) Logo



b) Imagen de Prueba



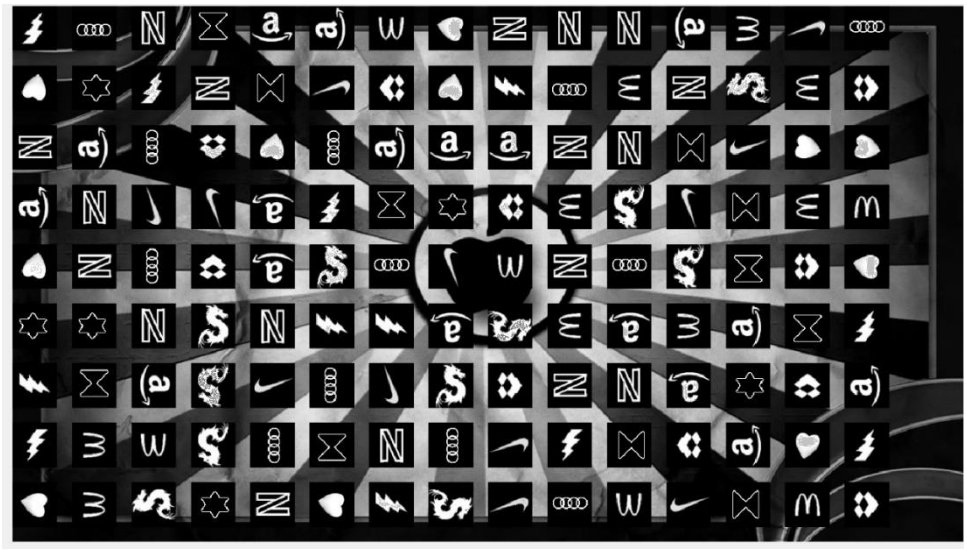
c) Imagen Resultante

Figura 8. Reconocimiento de formas en imágenes en blanco y negro

Detección de formas en Imágenes en Escala de Grises

El segundo reto fue probar el algoritmo con imágenes en escala de grises. Esta clase de imágenes representan un reto mayor para el algoritmo debido a que se utilizó la transformación de 12 a 1 bit por pixel, en otras palabras, de escala de grises a blanco y negro. Tal como se muestra en la Figura 8, se utilizó el mismo logo que en la prueba anterior, la resolución del logo y de la imagen de prueba también fueron las mismas. Se mantuvo un logo en blanco y negro, pero con una profundidad de 36 bits por pixel. Sin embargo, la imagen de prueba se encontraba en escala de grises, por lo que sus figuras tienen intensidades de luz variables. A pesar de la pérdida de información producto de la transformación, el logo es detectado en todos los casos dentro de la imagen sin importar las variaciones producidas por las figuras en escala de grises.

a) Logo



b) Imagen de Prueba

c) Imagen Resultante

Figura 9. Reconocimiento de formas en imágenes en escala de grises

Detección de formas en Imágenes a Color

La prueba más desafiante de esta investigación es la implementación del algoritmo en imágenes a color en alta definición. La Figura 9 muestra el logo e imagen de prueba utilizados. Para esta prueba, se utiliza la misma resolución en las imágenes. El logo se definió en blanco y negro, pero con 12 bits en formato RGB de 4 bits por color. La imagen de prueba se encuentra completamente a color. Para aumentar la dificultad de la prueba y comprobar el funcionamiento de esta implementación, los logos de la imagen de prueba tuvieron un fondo de color distinto

al logo original, de igual forma, se produjo variaciones de color en varias figuras. De esta manera, es posible averiguar si la transformación para poder aplicar el algoritmo es válida.

La transformación realizada implicó un cambio de color a luminancia y finalmente a blanco y negro. Este proceso produce aproximaciones o pérdida de información, sin embargo, simplifica la implementación del algoritmo para imágenes a color. Se aprecia en la Figura 9 que el algoritmo funciona perfectamente y ha detectado todos los logos en la imagen de prueba. También se observa que la definición de la imagen resultante es muy buena en comparación con la imagen original, por lo que la transformación de 12 a 36 bits por pixel es válida.

a) Logo

b) Imagen de Prueba

c) Imagen Resultante

Figura 10. Reconocimiento de formas en imágenes a color

CONCLUSIONES

Se logró implementar un algoritmo rápido de detección de formas para imágenes en alta definición de forma satisfactoria. El algoritmo fue implementado en una plataforma FPGA VIRTEX-7 con un puerto HDMI para la salida de video. Se desarrolló la configuración del transmisor de video ADV7511 incorporado en la tarjeta combinando descripción de hardware y programación de software. Se consiguió que el transmisor sea capaz de transmitir imágenes en resolución 1080p-60 full-HD.

Se probó el algoritmo con imágenes en blanco y negro aumentando la dificultad hasta imágenes en escala de grises e imágenes a color. Se realizaron varias transformaciones para mantener el algoritmo original e idéntico en todas las pruebas sacrificando información, pero garantizando un algoritmo rápido y sencillo de implementar. Se comprobó que estos cambios no disminuyeron la fidelidad y robustez del algoritmo. Por lo tanto, la implementación para detección de formas en imágenes a color en alta definición funciona perfectamente. Esta investigación puede dar paso a la aplicación e implementación del algoritmo utilizando cámaras a color y en alta definición para obtener un sistema de detección de formas en tiempo real.

REFERENCIAS BIBLIOGRÁFICAS

Analog Devices. (2012). ADV7511 Low/Power HDMI 1.4 Compatible Transmitter with Audio Return Channel PROGRAMMING GUIDE. Obtenido el 30 de abril de 2020 de https://www.analog.com.

Analog Devices. (2011). ADV7511 Low/Power HDMI 1.4 Compatible Transmitter with Audio Return Channel HARDWARE USER'S GUIDE. Obtenido el 30 de abril de 2020 de https://www.analog.com.

Borja, A., Varengues, G., Procel, L. M., Trojman, L., Arévalo, G., & Cardenas, D. (2019, November). Implementation and Comparison of a Fast Shape Recognition Algorithm using Different FPGA Platforms. In *2019 IEEE Fourth Ecuador Technical Chapters Meeting (ETCM)* (pp. 1-5). IEEE.

Jack, K. (2008). *Digital video and DSP: Instant access*. Burlington, MA 01803, USA: Newnes.

Ogier, M., & Schmitz, T. (2004). *Basics of Video: From Simple Analog to HDTV APPLICATION NOTE AN 1695.*

Pratt, W. (2007). *Digital Image Processing* (4th edition). Los Altos, California: John Wiley & Sons, Inc.

Xilinx. (2019). VC707 Evaluation Board for the Virtex-7 FPGA User Guide. Obtenido el 30 de abril de 2020 de https://www.xilinx.com.

Xilinx. (2015). Block Memory Generator v8.2. Obtenido el 30 de abril de 2020 de https://www.xilinx.com.

Xilinx. (2016a). AXI IIC Bus Interface v2.0. LogiCORE IP Product Guide. Obtenido el 30 de abril de 2020 de https://www.xilinx.com.

Xilinx. (2016b). MicroBlaze Processor Reference Guide. Obtenido el 30 de abril de 2020 de https://www.xilinx.com.

Xilinx. (2017). AXI UART Lite v2.0. LogiCORE IP Product Guide. Obtenido el 30 de abril de 2020 de https://www.xilinx.com.