Implementación en FPGA de un conversor de señal de video compuesto a señal VGA

FPGA implementation of a composite video signal to VGA signal converter

Álvaro Anzueto[1], Yesenia González[2], Sergio Garduza[3]

Recibido: Junio 2014 Aceptado: Agosto 2014

Resumen.- En este artículo se presenta el diseño e implementación de un sistema de conversión de señal de video compuesto a señal VGA, el cual realiza la captura y despliegue de video basado en el estándar ITU-R BT.601 para monitores de televisión digital. El sistema cuenta con tres principales componentes: un sensor de video analógico que entrega la señal de video digital compuesta, una tarjeta digitalizadora que realiza la conversión del video analógico al digital en el formato de muestreo 4:2:2 y emplea el espacio de color YCbCr y finalmente un FPGA (tarjeta NEXIS 2, con el circuito integrado 3s1200efg320-5) en donde se ha implementado el diseño del procesamiento de la señal de video para poder ser desplegadas en un monitor. En el FPGA se han desarrollado los módulos de conversión de muestreo 4:2:2 al 4:4:4 y el módulo de conversión entre los espacios de color YCbCr al RGB (formato de color utilizado en los dispositivos de despliegue). La síntesis del diseño en el FPGA se ha realizado utilizando la herramienta XST (Xilinx Synthesis Tool) que es parte de la interfaz de desarrollo ISE Project Navigator 13.3 de la compañía Xilinx®. Las ecuaciones para realizar la transformación entre los espacios de color han sido implementadas y simuladas en MATLAB® para la validación de resultados.

Palabras clave: FPGA; formato 4:2:2; formato 4:4:4; ITU-R BT.601; espacio de color YCbCr; espacio de color RGB; VHDL; señal de video.

Summary.-*This paper describes the design and implementation of a system for converting composite video signal to VGA signal, which performs the capture and display of video based on the ITU-R BT.601 standard for digital television monitors. The system has three main components: an analog sensor that delivers the composite digital video signal, a graphics card that converts analog video to digital in 4:2:2 sampling format and uses the color space YCbCr and finally a FPGA (NEXIS card 2 , with the integrated circuit 3s1200efg320 -5) where the design has been implemented for processing the video signal to be displayed on a monitor. In the FPGA have been developed 4:2:2 to 4:4:4 sampling conversion modules and the conversion module between YCbCr color space to RGB (color format used in display devices). The synthesis of the FPGA design has been made using the XST (Xilinx Synthesis Tool) that is part of the interface development ISE Project Navigator 13.3 of the Xilinx® company. The equations for the transformation between color spaces have been implemented and simulated in MATLAB ® for validation of results.*

Keywords: *FPGA; format 4:2:2; format 4:4:4; ITU-R BT.601; YCbCr space color; RGB space color; VHDL; video signal.*

1 UPIITA - Instituto Politécnico Nacional. México, aanzuetor@ipn.mx (Autor de contacto)
2 UPIITA - Instituto Politécnico Nacional. México, ygonzalezn@ipn.mx
3 UPIITA - Instituto Politécnico Nacional. México, sgarduzag@ipn.mx

1. Introducción.- Los sistemas de visión en tiempo real permiten la extracción de información de una escena a partir de su proyección en un plano de imagen bidimensional, así mismo deben satisfacer las condiciones impuestas en su tiempo de respuesta, es decir, en el tiempo transcurrido entre la presentación de las entradas y la obtención de sus salidas. Debido a los resultados obtenidos, los sistemas de visión en tiempo real han tenido gran aceptación en áreas como el procesamiento de imágenes de video conferencias, telemedicina, monitoreo, robótica, entre otros. Si la aplicación del sistema de visión es para sistemas embebidos, por ejemplo en robótica autónoma móvil, las actividades del robot requerirán de la percepción de su entorno para la toma de decisiones, pero será necesario ser más estricto que en aplicaciones no embebidas al considerar las capacidades, dimensiones y costo del hardware a utilizar para llevar a cabo las tareas requeridas con una optimización de recursos.

El hardware comúnmente utilizado para desarrollar las aplicaciones de análisis de imágenes son los *Procesadores Digitales de Señales (DSPs) o los Circuitos de Aplicaciones Específicas (ASICs)*, pero a medida que aumenta la complejidad en los algoritmos de procesamiento se requieren de sistemas con un mejor desempeño y de menor tiempo de ejecución. *Field-Programmable Gate Arrays* (*FPGA)* son circuitos integrados formados por arreglos de bloques lógicos que permiten reconfigurarse las veces que sea necesario para depurar su funcionalidad, se puede comunicar con ellos (mediante terminales de entrada/salida) por medio de conexiones alámbricas llamadas canales de comunicación. El tamaño y la velocidad de los FPGAs son equiparables a los ASICs, pero los FPGAs son más flexibles, su ciclo de diseño es más corto y su paralelismo da un mayor rendimiento [1-3].

Para la trasmisión de video digital estándar las señales están codificadas bajo las recomendaciones de ITU-R BT.601, que especifica que la señal de video se debe transmitir en formato 4:2:2 y utiliza el espacio de color YCbCr, en este espacio de color la componente Y representa la iluminación o luma presente en el dato, Cb representa la diferencia de la croma en azul a la luminancia y Cr la diferencia de la croma en rojo a la luminancia; en adelante, en este artículo se nombrarán los términos anteriores como croma en azul y croma en rojo respectivamente.

Debido a que el ojo humano es más sensible a los cambios de iluminación que a los de croma, la trasmisión de video digital se envía en el formato de muestreo 4:2:2, lo cual reduce el ancho de banda en la señales; en ese formato una trama de datos estará comprendida de 4 muestras de la luma (Y), dos muestras de la croma en azul (Cb) y dos muestras de la croma en rojo (Cr) [4], de esta forma se ve reducida la información del color pero se mantiene una relación alta de los cambios de iluminación, logrando así una calidad de imagen aceptable [4-6].

Cuando una trama de datos es capturada por un sistema de procesamiento de video, es necesario recodificarla y tener la misma cantidad de datos en luma y croma, por lo tanto, se requiere cambiar al formato de muestreo 4:4:4; con este nuevo formato es posible realizar la trasformación entre espacios de color.

Los dispositivos de despliegue de imágenes o video trabajan bajo el espacio de color RGB, por lo cual, es necesario trasformar los datos de luma y croma a las componentes de rojo (R), verde (G) y azul (B) [4].

La tarjeta digitalizadora VDEC-1 [7], distribuida por la compañía DIGILENT® y que es la utilizada en este trabajo, cumple los estándares de transmisión y codificación de señales de video digital ITU-R BT.601, por lo que entrega la señal de video muestreada en formato 4:2:2 y en el espacio de color YCbCr. Estas señales son recodificadas y transformadas al espacio de color RGB para poder ser interpretadas por un monitor [3,4]. Y. Yang y colaboradores [8], presentaron un

conversor de espacio de color YCbCr a RGB considerando como hardware de implementación un DSP, ellos consideran 240 niveles para las componentes R, G y B. En este trabajo se consideran 256 niveles para cada componente y es similar a lo expuesto en el comando *"ycbcr2rgb"* contenido en el Toolbox de Procesamiento de Imágenes de Matlab®.

El sistema para obtener los datos, que sustenta lo planteado en este artículo, cuenta con un sensor de video analógico de tipo CMOS con una resolución VGA (640 x 480) y velocidad de captura de 30 cuadros por segundo. El formato VGA es comúnmente utilizado en robótica móvil debido a que la cantidad de datos a procesar es baja (comparado al formato HD) y ofrece una calidad de imagen suficiente para la realización de las tareas requeridas en los sistemas actuales [9, 10]. Para digitalizar la señal de video se empleó la tarjeta VDEC-1, la cual entrega la señal digital de video en formato 4:2:2 en el espacio de color YCrCb con 8 bits de resolución para cada canal. La trama de datos es procesada por la tarjeta FPGA NEXIS 2 de la compañía Xilinx® [3], en ella se implementan los bloques de transformación del formato 4:2:2 al 4:4:4 y la trasformación entre los espacios de color YCbCr al RGB. El diagrama a bloques de las partes físicas empleadas en el desarrollo de este trabajo se muestra en la Figura I.

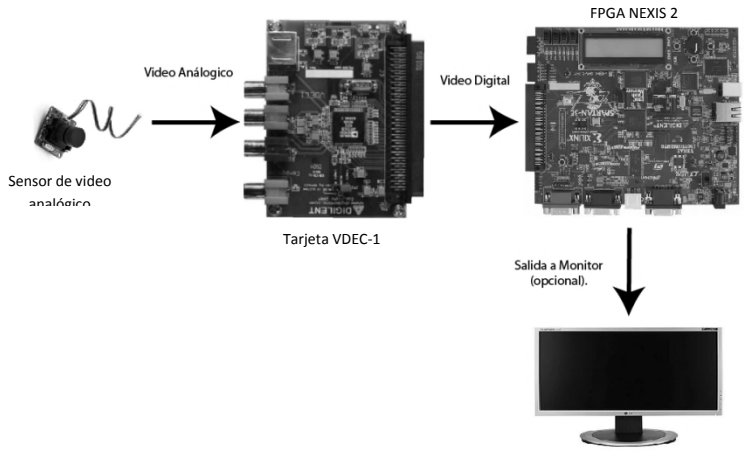

Figura I.- Diagrama de componentes físicos del sistema.

Para la trasformación del formato 4:2:2 al 4:4:4, es necesario realizar una expansión de los datos de croma; para realizar esto se tienen diferentes alternativas [11]; i.e. se toman dos valores de croma, se promedian y el resultado es utilizado como uno de los datos de expansión. En este trabajo se considera el duplicar los valores de croma para obtener los datos de expansión, esto ayuda tanto en la reducción del hardware utilizado como en el tiempo de ejecución.

Lo presentado en este artículo es una etapa requerida en el procesamiento de imágenes en sistemas robóticos autónomos, ya que la mayoría de los sensores de video de bajo costo operan con señales analógicas en el formato de video compuesto.

El presente artículo se encuentra distribuido de la siguiente manera: en la sección 2 se presentan las estructuras de los formatos 4:2:2 y 4:4:4 y las ecuaciones para obtener los coeficientes de posición en la trasformación entre ambos formatos. En la sección 3 son presentadas las ecuaciones de transferencia del modelo de color YCbCr al RGB, sugeridas por el estándar ITU-R BT.601-5, los resultados numéricos de estas ecuaciones son comprobados con los datos obtenidos al utilizar la función *"ycbcr2rgb"* contenida en el Toolbox de Procesamiento de Imágenes de MATLAB® [13]. En la sección 4, se presenta la implementación en FPGA, utilizando lenguaje de descripción de

hardware de los bloques desarrollados. Los datos obtenidos y las gráficas de tiempo se discuten en la sección 5. En sección 6 se dan las conclusiones obtenidas de este trabajo.

2. Proceso de transformación entre los formatos 4:2:2 al 4:4:4.- Existen diversos esquemas de muestreo para las componentes del espacio de color YCrCb, los más utilizados son: 4:4:4, 4:2:2, 4:2:0 y 4:1:1. En estos formatos el primer número representa la muestra de iluminación (luma) o componente Y, y los restantes las componentes de croma del color Cb y Cr. Para el formato 4:2:2, basado en el estándar ITU-R BT.601, los valores de croma son muestreados a la mitad de la frecuencia de luminancia; esto se ve representado en Figura II, en donde los cuadros oscuros representan una muestra completa de Y, Cb y Cr y los círculos únicamente la muestra de Y. Las líneas horizontales representan las tramas de datos y las líneas verticales los pulsos de reloj que determinan la velocidad de muestreo, por lo tanto, después de cuatro líneas verticales se obtendrá cuatro datos de Y, dos datos de Cb y dos de Cr, lo que da nombre al formato de muestreo 4:2:2. En la Figura III se muestra el formato 4:4:4, en este formato después de 4 pulsos de reloj se tiene la misma cantidad de valores para las tres componentes Y, Cb, y Cr.

Como se ha mencionado, la tarjeta empleada para la digitalización de la señal de video analógica es la nombrada V-DEC1, esta tarjeta cuenta con el circuito integrado ADV7183 (de la compañía Analog Devices®), que es el encargado de proveer la señal digital en el formato 4:2:2 y cumplir con el estándar ITU-R BT.601.

La transformación entre formatos 4:2:2 a 4:4:4 se realiza con base en la trama de salida de la tarjeta V-DEC1; los desarrolladores de la compañía Xilinx® en [11] proponen tres métodos para completar los datos croma que faltan en el formato 4:4:4., pero presenta el inconveniente de consumir muchos recursos de hardware al momento de su implementación.

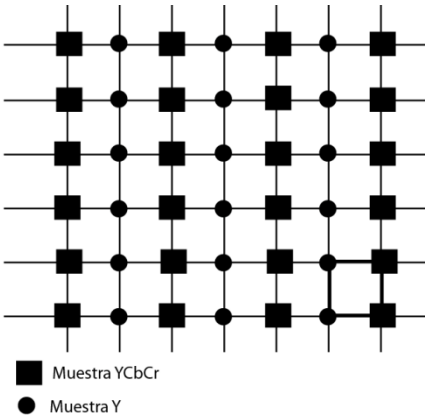

Figura II.- Representación del formato 4:2:2.

En este trabajo para completar los datos en el formato 4:4:4, se duplicaron los valores de croma, debido a que después de implementar los métodos antes mencionados, se observó que en los datos de croma resultantes, comparados con un valor precedente, mantenían un valor numérico similar; esto reduce drásticamente la complejidad de la trasformación entre los formatos; tan solo se toma el valor de croma correspondiente y se repite el dato un ciclo de reloj posterior. Un ejemplo gráfico de este proceso es presentado en la Figura IV.

La trama de datos que proviene de la tarjeta V-DEC1 se subdivide en paquetes, un paquete de datos contendrá la información correspondiente a un píxel en el formato 4:2:2. El paquete de datos está ordenado de la siguiente manera: en la primera localidad se tiene información de luma Y_{impar},

ISSN 2301-1092 • ISSN (en línea) 2301-1106

seguido de un dato de croma en rojo Cr, un nuevo dato de luma Y_{par}, y finalmente el dato que corresponde a croma en azul Cb; esta distribución se mantiene en toda la trama de datos. De un paquete de datos, que corresponde a un píxel en el formato 4:2:2, se obtienen la información para formar dos píxeles en el formato 4:4:4.

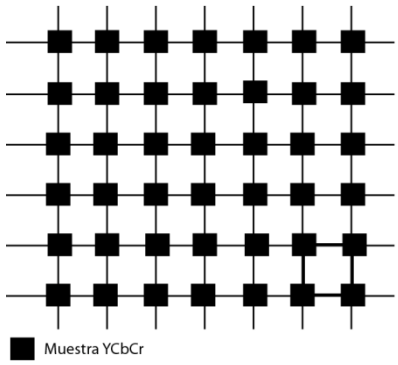

Figura III.- Representación del formato 4:4:4.

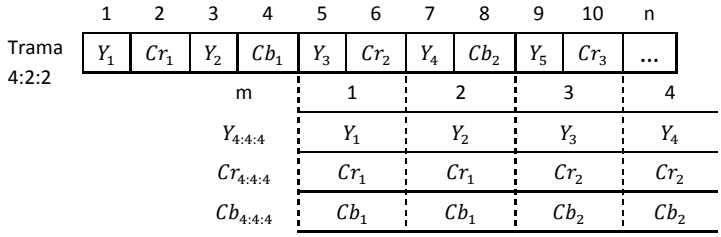

Figura IV.- Sistema de recodificación del muestreo Y4:2:2, Cr4:2:2 y Cb4:2:2 al Y4:4:4, Cr4:4:4 y Cb4:4:4, se muestra la organización de datos en el FPGA.

Las Ecuaciones (1), (2), y (3) indican las posiciones, en la nueva trama de datos de las componentes del espacio de color. Con apoyo de la Figura IV se puede observar que la conversión se realiza después de que se hayan presentado al menos uno de los paquetes de información del formato de entrada, para dar una mejor idea del proceso considere a *n = 5* y *m=1*, el resultado sería para luma: $Y_{4:4:4\,(1)} = Y_{4:2:2\,(1)}$, para croma en rojo $Cr_{4:4:4\,(1)} = Cr_{4:2:2(1)}$, y para croma en azul $Cb_{4:4:4\,(1)} = Cb_{4:2:2(1)}$. Ahora, si consideramos para *m=2* y *n=7* y las sustituimos en las ecuaciones tendremos que para la segunda posición de luma en la trama de salida se tendrá el segundo valor de luma de la trama de entrada, esto es $Y_{4:4:4\,(2)} = Y_{4:2:2\,(2)}$, pero las cromas, en la segunda posición de la trama de salida se repiten las cromas de la primera posición de la trama de entrada.

$$Y_{4:4:4\,m} = Y_{4:2:2\left(\frac{n-3}{2}\right)}, \tag{1}$$

$$Cr_{4:4:4\,m} = Cr_{4:2:2\left(\frac{n-1}{4}\right)}, \tag{2}$$

$$Cb_{4:4:4\,m} = Cb_{4:2:2\left(\frac{n-1}{4}\right)}. \tag{3}$$

3. Transformación del espacio de color YCbCr al RGB.- El espacio de color YCbCr fue desarrollado como una parte de la Recomendación ITU-R BT.601 por la "Worldwide Digital

Component Video Standard" y es usada en la trasmisión de señal de televisión. Los dispositivos de despliegue de imágenes (monitores, pantallas de tv, etc.) usan el espacio de color RGB, el cual trabaja con los tres colores primarios rojo (Red), verde (Green) y azul (Blue) y los demás colores pueden ser formados dependiendo de la contribución de cada uno de estos 3.

Cumpliendo las recomendaciones del estándar ITU-R BT.601 en su apartado 3.5.2 [4], se tiene la Ecuación (4) para un sistema normalizado en sus componentes, luma (Y) y croma (Cb y Cr).

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} + (A^T \times C^T) \begin{pmatrix} R \\ G \\ B \end{pmatrix}. \tag{4}$$

De donde

$$A = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.169 & -0.331 & 0.500 \\ 0.500 & -0.419 & -0.081 \end{bmatrix}, \tag{5}$$

$$C = \begin{bmatrix} {}^{220}/_{256} & 0 & 0 \\ 0 & {}^{225}/_{256} & 0 \\ 0 & 0 & {}^{225}/_{256} \end{bmatrix}. \tag{6}$$

Las filas en la matriz A representan los coeficientes que normalizan los valores de Y, Cb y Cr respectivamente; la matriz C representa la re-cuantificación de los vectores, ya que como lo menciona el estándar ITU-R BT.601 en su apartado 3.5.3 [4], el vector de luma ocupa 220 niveles y las cromas 225, de 256 niveles permitidos al trabajar con 8 bits para su representación; el resto de valores son reservados para señales de sincronización en dispositivos analógicos. Debido a que el sistema en este trabajo no ocupa esa información, se realiza una re-cuantificación y se amplían todos los vectores a un rango de 256 valores.

Debido a que el propósito de este artículo es convertir del espacio de color YCbCr al espacio de color RGB y en el apartado de ITU-R BT.601 sólo se menciona la conversión de RGB a YCbCr, es necesario realizar la transformación inversa para obtener la conversión de YCbCr a RGB, esto se logra al obtener la matriz inversa de $A^T \times C^T$ (ver Ecuación (7)).

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = B \times \left(\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} - \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} \right). \tag{7}$$

De donde

$$B = \left(\left[\begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.169 & -0.331 & 0.500 \\ 0.500 & -0.419 & -0.081 \end{bmatrix}^T \times \begin{bmatrix} 220/256 & 0 & 0 \\ 0 & 225/256 & 0 \\ 0 & 0 & 225/256 \end{bmatrix} \right]^T \right)^{-1}. \tag{8}$$

La ecuación (7) fue implementada en el lenguaje de programación Matlab® y los datos se compararon con los resultados obtenidos de la función *ycbcr2rgb* del Toolbox de Procesamiento de Imágenes de Matlab® [12], obteniéndose iguales resultados. En la Figura V se presentan 9 datos que han sido transformados entre los espacios de color. La función *ycbcr2rgb* de Matlab® recibe el dato de Y sobre un rango de (16, 235), y para Cb y Cr en un rango de (16,240) y entrega el

ISSN 2301-1092 • ISSN (en línea) 2301-1106

resultado numérico para R, G, y B en un rango de (0,255), con valores enteros proporcionales a 8 bits. Se considera el valor 0 como ausencia de color y el valor 255 como saturación de color.

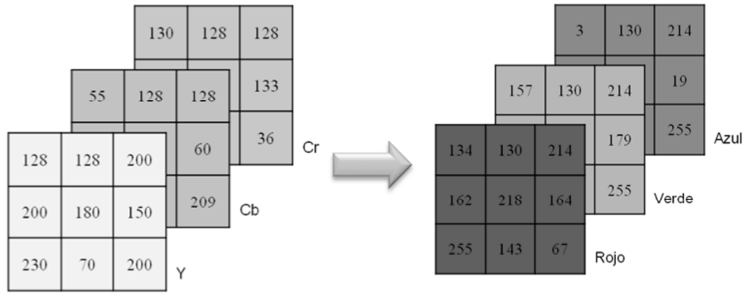

Figura V.- Transformación de datos del espacio de color YCbCr a RGB.

4. Bloques de la implementación.- En la Figura VI se presenta el diagrama a bloques del sistema. El bloque con la etiqueta "Sensor" representa la adquisición de la señal de video de forma analógica, esto se realiza por medio de un sensor de video analógico de tipo CMOS, con señal VGA (resolución de 640x480). El bloque encargado de realizar el proceso de digitalización de la señal de video está constituido por la tarjeta VDEC-1 de la compañía Digilent®, quien provee la información en formato 4:2:2 de las componentes Y, Cb y Cr, y son los datos de entrada de la tarjeta FPGA NEXIS 2 de la compañía Xilinx®, los datos capturados por el FPGA primeramente son transformados al formato 4:4:4 pero aun en el espacio de color YCrCb, el siguiente proceso es representado por el bloque con la etiqueta "Ecuación 5" que realiza el proceso de transformación entre los espacios de color, como paso final es el despliegue del video, para ello la tarjeta NEXIS cuenta con un conector de tipo DB-15 que permite realizar la comunicación con un monitor para presentar los resultados de forma visual.

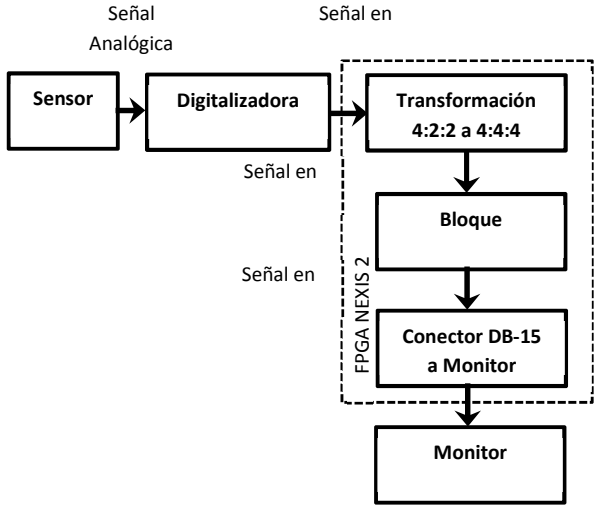

Figura VI.- Diagrama a bloques del Sistema.

4.1 Implementación del bloque de transformación del formato 4:2:2 al 4:4:4.- El pseudocódigo mostrado en la Figura VII describe la asignación de los valores en los buffers de salida de Y, Cb, y Cr, las variables Y_1, Cr_1 y Y_2 son buffers internos que ayudan en el proceso de obtención de los valores de salida; la variable *conpix* es un contador de eventos de 4

posibilidades, de las cuales las 2 primeras definen un píxel y su complemento a otro.

Los diagramas de tiempo se realizaron con la herramienta ISIM que forma parte del IDE de Xilinx®. Las gráficas son presentadas en Figura VIII. Es importante notar que después de cuatro pulsos de reloj se obtienen los valores en los buffers de salida.

```
if (conpix(1 downto 0)=0) then
Y_1 <= entrada;
end if;
if (conpix(1 downto 0)=1) then
Cr_1 <= entrada;
Y      <= Y_2;
end if;
if (conpix(1 downto 0)=2) then
Y_2 <= entrada;
end if;
if (conpix(1 downto 0)=3) then
Cb <= entrada;
Cr <= Cr_1;
Y  <= Y_1;
end if;
```

Figura VII.- Pseudocódigo para la asignación de datos en los buffers de salida.

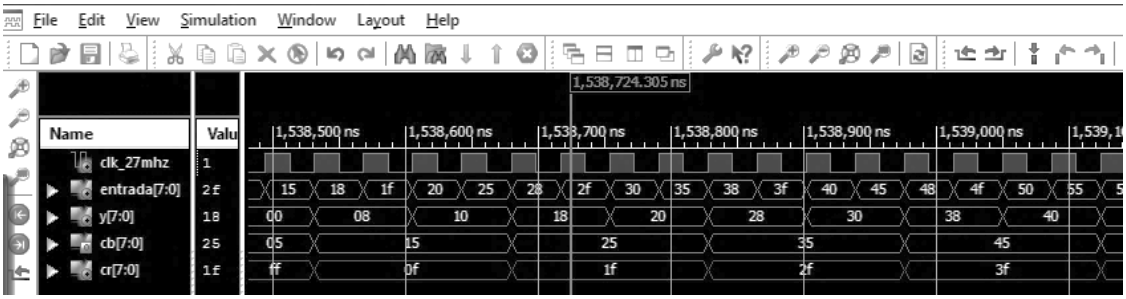

Figura VIII.- Grafica en simulador ISIM de los pulsos de luma "Y", croma en azul "Cb" y croma en rojo "Cr" en el FPGA.

Utilizando la herramienta Schematic viewer de Xilinx® en su opción RTL se generó el bloque esquemático del proceso de transformación (ver Figura IX), en donde se observa en la parte izquierda de la figura las señales de entrada y a la derecha las señales de salida. La señal de entrada etiquetada con el nombre "entrada (7,0)", representa la trama de datos de forma serial proveniente de la tarjeta digitalizadora V-DEC1. La etiqueta CLK_27, es asignada a la señal de reloj de sincronización de video proveniente de la tarjeta V-DEC1, y determinada por el estándar ITU-R BT.601. Considerando que la aplicación es desarrollada para procesamiento de señal de video, a la entrada, se presentan 3 señales: senF, senH, senV, que indican el número de cuadro (también llamado *frame*), la señal de sincronía horizontal y la señal de sincronía vertical respectivamente. A la salida, el bloque contiene únicamente 3 buses de 8 bits que corresponden a las señales de $Y_{4:4:4}$, $Cb_{4:4:4}$, $Cr_{4:4:4}$. En un proceso posterior, estas señales son procesadas para la conversión al espacio de color RGB.

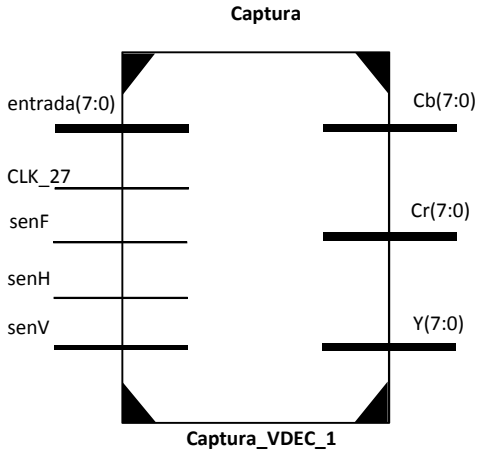

Figura IX.- Bloque esquemático del proceso de transformación entre los formatos 4:2:2 al 4:4:4.

4.2 Implementación del bloque de transformación de YCrCb a RGB.- La ecuación (7) ha sido implementada en el FPGA utilizando el lenguaje de descripción de hardware VHDL. En la Figura X se presenta un fragmento del código que corresponde a la obtención de la componente rojo (R) del espacio de color RGB. La componente R está defina por: $R = [(Y - 16) * 19077] + [(c_r - 128) * 26149]$.

```
std_logic_vector ( 9 downto 0):= (others=>'0');
constant menos_16 : std_logic_vector (17 downto 0):= "11111111111110000"; -- (-16)
constant menos_128 : std_logic_vector (17 downto 0):= "11111111110000000";
                                                        -- (-  128)
constant c1          : std_logic_vector (17 downto 0):= "000100101010000101"; -- ( +19077)
constant c2          : std_logic_vector (17 downto 0):= "000110011000100101"; -- ( +26149)

R_temp  <= T1+T2;

multi_1: MULT18X18SIO  -- T1 = (Y-16) * c1
        generic map ( AREG => 0, BREG => 0, B_INPUT => "DIRECT", PREG => 0)
        port map (
                P =>T1, -- Salida de 36 bits del multiplicador
                A =>Y_temp, -- Entrada de 18 bits del multiplicador
                B =>c1,          -- Entrada de 18 bits del multiplicador
                BCIN => (others=>'0'), CEA => '0', CEB => '0', CEP => '0',
                CLK => clk_mult , RSTA =>'0', RSTB =>'0', RSTP =>'0' );

multi_2: MULT18X18SIO  -- T2 = (Cr-128) * c2
        generic map ( AREG => 0, BREG => 0, B_INPUT => "DIRECT", PREG => 0)
        port map (
                P =>T2, -- Salida de 36 bits del multiplicador
                A =>Cr_temp, -- Entrada de 18 bits del multiplicador
                B =>c2,          -- Entrada de 18 bits del multiplicador
                BCIN => (others=>'0'), CEA => '0', CEB => '0', CEP => '0',
                CLK => clk_mult, RSTA =>'0', RSTB =>'0', RSTP =>'0' );
```

Figura X.- Código para la obtención del canal rojo (R).

ISSN 2301-1092 • ISSN (en línea) 2301-1106

En el fragmento de código de la Figura X la variable nombrada *"R_temp"* (valor digital de rojo) esta determinada por la sumatoria de las *variables "T1"* y *"T2"*. La variable *"T1"*, está compuesta por la suma de las variables *"Y"* y *"menos_16"* y por la multiplicación con el coeficiente *"c1"*. La variable nombrada *"menos_16"* representa la resta del valor 16 a la componente de luma (Y) y la variable nombrada *"c1"*, representa el coeficiente de la matriz de transformación inversa en la ecuación (7).

La operación de multiplicación se realiza utilizando la siguiente primitiva en la línea de código: *multi_1: MULT18X18SIO -- T1 = (Y-16) * c1*.
Para la obtención del término *"T2"*, se desarrolla un procedimiento similar al utilizado en *"T1"*. Con la diferencia de emplear las variables *"menos_128"* y *"c2"*.
En la Figura XI se presenta el bloque esquemático del conversor de espacio de color YCrCb al RGB.

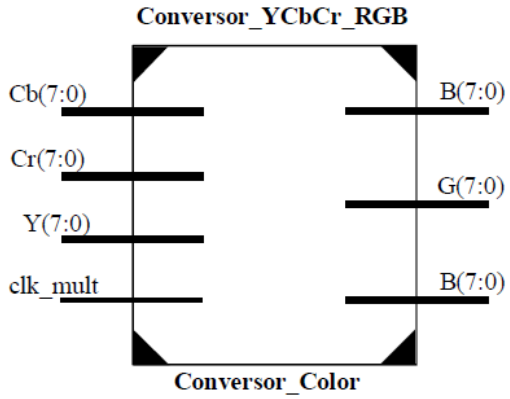

Figura XI.- Bloque del proceso de transformación YCrCb a RGB.

Los valores de entrada en el bloque de la Figura XI son las señales de Y, Cb y Cr, obtenidas del bloque de conversion del formato 4:2:2 al 4:4:4 (todas ellas en bus de 8 bits), a la salida se tienen tres buses que corresponden a las componetes de color RGB. De forma interna el bloque está constituido por 5 multiplicadores que desarrollan los terminos *T1*, *T2*, *T3*, *T4* y *T5*, que son necesarios para desarrollar las operaciones de la ecuación (7). Como ejemplo gráfico de estas multiplicaciones se presenta en la Figura XII el bloque que desarrolla el termino *T1= c1*(Y_tem+menos16)*, este término es utilizado en la obtencion de la componente de color rojo R.

ISSN 2301-1092 • ISSN (en línea) 2301-1106

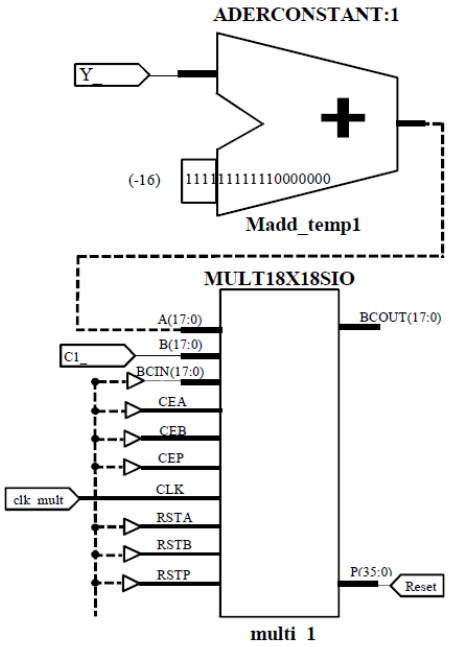

Figura XII.- Bloque gráfico del desarrollo del término T1=cl(Y_tem+(-16)).*

5. Resultados.- Las ecuaciones que describen el proceso de transformación entre formatos de muestreo y las transformaciones entre espacios de color fueron implementadas empleando lenguaje de descripción de hardware (VHDL) en la tarjeta NEXIS 2. Las mediciones de las gráficas de tiempo presentadas se realizaron empleando el osciloscopio de fosforo digital serie Tektronix 4000.

5.1. Resultados del cambio de formato 4:2:2 al 4:4:4.- Para el proceso de trasformación entre los formatos de muestreo ha sido necesario implementar las ecuaciones (1), (2) y (3) en el FPGA utilizando el lenguaje de descripción de hardware. En las Figuras XIII, XIV y XV se muestran 3 series de pulsos, la serie superior, en las tres imágenes, representa la información de salida de la tarjeta V-DEC1 muestreada en formato 4:2:2, la segunda serie representa el dato de salida en formato 4:4:4 y la tercera, la señal de reloj de 27 Mhz que sirve como señal de sincronía entre los formatos. Los flancos de subida son considerados como el momento de ejecución de alguna acción. En la Figura XIII, la segunda trama de pulsos representa el valor de luma (Y) muestreado en formato 4:4:4. El primer dato en la trama de entrada tiene el valor decimal de 18, este dato es colocado en la trama de salida 4 pulsos después (reloj de 27 Mhz) y corresponde al primer dato válido en la trama de salida (Y_1), el segundo dato en luma (Y_2) tiene el valor decimal 20 y corresponde al tercer dato presente de la trama de entrada.

En la Figura XIV, la segunda trama contiene los valores de croma en azul (Cb) en formato 4:4:4. El primer dato válido de esta señal se obtiene 4 pulsos reloj después de que se ha presentado el primer dato en la trama de entrada. Caso similar sucede para la componente de croma en rojo (Cr), ver Figura XV.

ISSN 2301-1092 • ISSN (en línea) 2301-1106

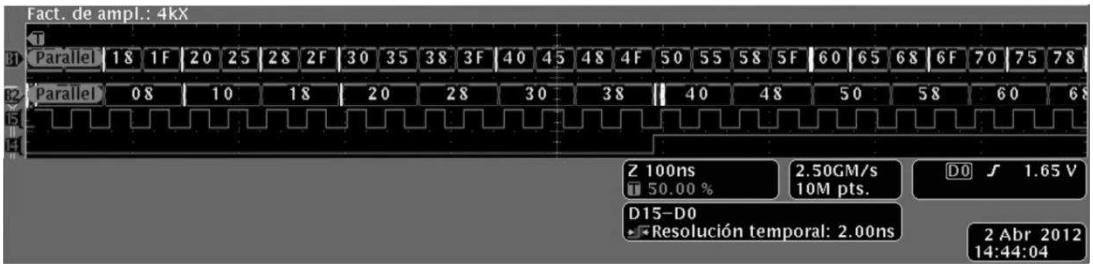

Figura XIII.- Gráfica en osciloscopio de los pulsos de luma Y obtenidas del FPGA basado en la ecuación (1).

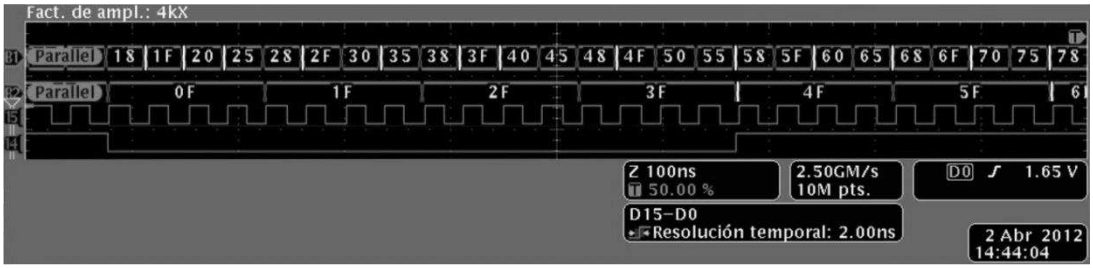

Figura XIV.- Grafica en osciloscopio de los pulsos de croma en azul Cb en el FPGA basado en la ecuación (2).

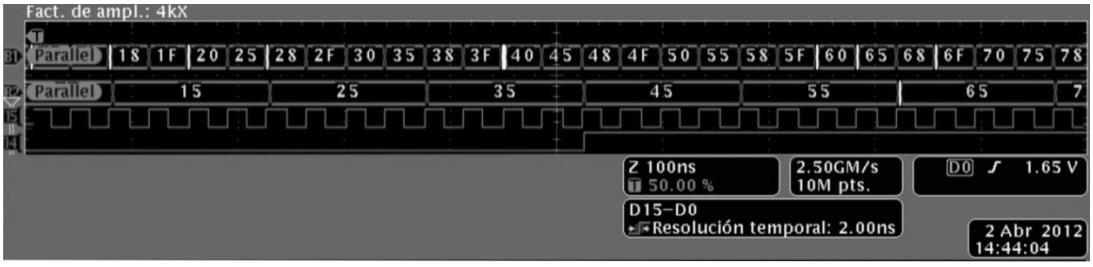

Figure XV.- Gráfica en osciloscopio de los pulsos de croma en rojo Cr en el FPGA basado en la ecuación (3).

5.2. Resultado de la transformación del espacio de color YCrCb al RGB.- Para comprobar el resultado numérico del bloque de transformación de espacios de color implementado en hardware, se procedió a comparar los datos obtenidos en hardware y software. Para el caso del software se utiliza la función *ycbcr2rgb* del Toolbox de Procesamiento de Imágenes de Matlab®. A manera de ejemplo se presentan 9 datos en el espacio de color YCbCr y el resultado de la función, todos ellos se enlistan en la Tabla I.

Entrada				Salida			
Y	128	128	200	R	134	130	214
Cb	55	128	128	G	157	130	214
Cr	130	128	128	B	3	130	214

Tabla I.- DATOS DE PRUEBA. Función "ycb2rgb" del Toolbox de Procesamiento de Imágenes de Matlab®.

ISSN 2301-1092 • ISSN (en línea) 2301-1106

La Tabla II presenta los datos obtenidos del bloque diseñado en VHDL del conversor de espacios de color, éste fue alimentado con los mismos datos que en el proceso por software.

Entrada				Salida			
Y	128	128	200	R	133	130	214
Cb	55	128	128	G	157	130	214
Cr	130	128	128	B	0	130	214

Tabla II.- DATOS DE PRUEBA. Implementación en VHDL de la transformación del espacio de color "YCbCr" al espacio de color "RGB".

La Figura XVI muestra las gráficas de tiempo del bloque diseñado al presentarle los datos en la Tabla II.

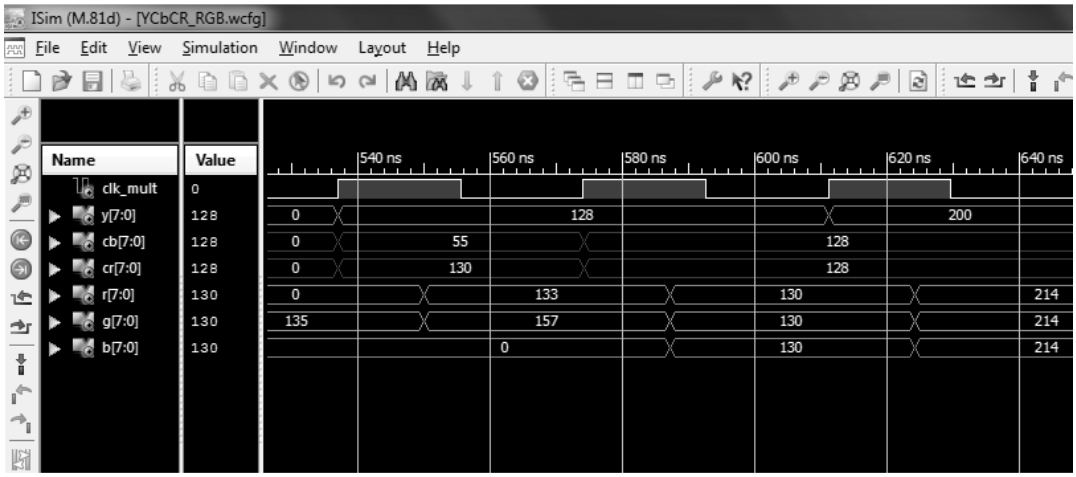

Figura XVI.- Gráficas de propagación de tiempos del bloque de conversión YCrCb a RGB en el FPGA.

De la gráfica en la Figura XVI, puede obtenerse el tiempo de retardo de propagación del bloque conversor de color, es decir, el tiempo que transcurre a partir de que las señales en las entradas están presentes hasta que el bloque conversor presenta los datos resultantes, este tiempo se mide a partir de que se presenta un cambio dentro de cualquiera de las señales Y, Cb o Cr y hasta que presenta la conmutación en las señales de salida R, G y/o B; el tiempo medido en la gráfica nos indica un retardo de 12 ns. En la gráfica también se puede observar que las señales Y, Cb y Cr están sincronizadas con el flanco de subida de la señal de reloj "clk_mult", pero la respuesta a estas señales no está relacionada con dicha señal, debido a que la implementación en vhdl es mediante primitivas. La señal de reloj "clk_mult" es requerida por las primitivas de multiplicación, pero en el presente trabajo no controla ningún proceso; dicha señal se puede utilizar para posteriores aplicaciones.

De la misma gráfica se puede hacer la siguiente observación: la discrepancia entre los valores obtenidos usando la función de Matlab® y los valores de la implementación digital se debe a la resolución tomada por el sistema respecto al punto flotante, ya que para la conversión se realiza un corrimiento equivalente a una división de 2^{13} para obtener valores numéricos enteros, el factor 2^{13} puede ser aumentado para mejorar la aproximación, pero esto conlleva a utilizar una mayor cantidad de recursos en el dispositivo (NEXIS 2).

En las Figuras XVII y XVIII se presentan las imágenes que se obtuvieron después de la trasformación entre los formatos de muestreo 4:2:2 a 4:4:4 y el cambio entre los espacios de color YCrCb al RGB. Las dimensiones de las imágenes son de 640×480 (VGA), y 8 bits por cada canal R, G y B. La imagen en la Figura XVII se muestra en perspectiva para apreciar el sistema completo y en la Figura XVIII se muestra la imagen de despliegue en monitor de la señal de video resultante del proceso de trasformación.

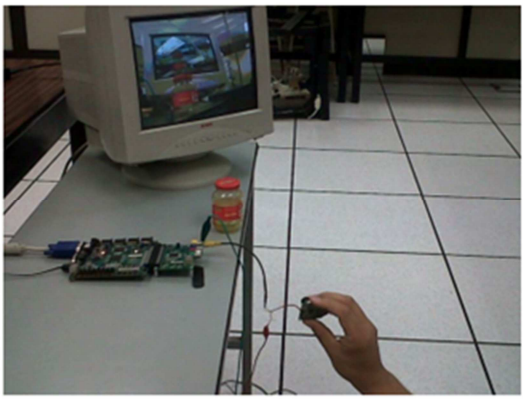

Figura XVII.- Imagen del sistema.

Figura XVIII.- Señal de video desplegada en monitor.

6. Conclusiones.- En este trabajo se ha abordado el proceso de conversión de señal de video compuesto a señal VGA. Para llevar a cabo esta tarea se diseñaron e implementaron los bloques de trasformación entre los formatos de muestreo 4:2:2 al 4:4:4 y la transformación entre los espacios de color YCrCb al RGB.

Se presentaron las ecuaciones para obtener los coeficientes de posición al transformar los formatos de muestreo 4:2:2 al 4:4:4. Considerando estas ecuaciones se diseñó el hardware que realiza la transformación, implementado en un FPGA utilizando el lenguaje de descripción de hardware VHDL.

Para poder desplegar el video capturado fue desarrollado el bloque de transformación entre espacios de color, este bloque recibe la señal muestreada en 4:4:4 en el espacio de color YCrCb, las

ISSN 2301-1092 • ISSN (en línea) 2301-1106

ecuaciones presentadas en la sección 3 fueron implementadas para obtener las señales de salida en formato de color RGB.

La señal de video capturada con el sensor ha sido desplegada en un monitor en formato VGA y se ha comprobado la eficiencia del sistema propuesto.

Las velocidades de respuesta obtenidas en este trabajo y basados en los tiempos presentados por la norma ITU-R BT.601, sugieren que el sistema presentado puede trabajar con sensores de adquisición de video que operen a mayores velocidades de captura, es decir, por arriba de los 30 cuadros por segundo.

7. Referencias

[1] D. G. Bailey, H. Poor. Design for embedded image processing on FPGAs. Singapore: John Wiley & Sons, 2011.
[2] N. Zainalabedin. VHDL: analysis and modeling of digital systems. 2nd ed. New York: McGraw-Hill, 1998.
[3] Xilinx, NEXIS 2 Family: Data Sheet product specification, Agosto 2009.
[4] ITU-R Recommendation BT.601-5: Studio encoding parameters of digital television for standard 4:3 and wide-screen 16:9 aspect ratios.
[5] R. C. González and R. E. Woods. Digital Image Processing. 3rd. ed. New Jersey: Pearson Education, 2008.
[6] Russ J. C., The Image Processing Handbook, 6th edition, 2011, CRC Press by Taylor and Francis Group.
[7] Digilent, Digilent Video Decoder Board (VDEC1) Reference Manual, Abril 2005.
[8] Y. Yang, P. Yuhua and L. Zhaoguang, "A Fast Algorithm for YCbCr to RGB Conversion," IEEE Transactions on Consumer Electronics, vol. 53, no. 4, pp. 1490-1493, Nov., 2007.
[9] A. Ghorbel, M. Jallouli, et. al. "An FPGA Based Platform for Real Time Robot Localization," Int. Conf. on Individual and Collective Behaviors in Robotics, pp. 56-61, Dic. 2013.
[10] Hagiwara H., Asami K., Komori M. Real-Time Image Processing System by Using FPGA for Service Robots. 1st Global Conference on Consumer Electronics, p. 720 – 723, Octubre 2012.
[11] G. Hawkes, "Digital Component Video Conversion 4:2:2 to 4:4:4," Xilinx reference manual, Dec. 19, 2001.
[12] C. M. Thompson and L. Shure "Image processing toolbox for use with Matlab," The Math works inc., Ago., 1993.