

# Implementación en FPGA del proceso de transformación del modelo de color YCbCr al RGB

Álvaro Anzueto-Ríos  
María Gisel Conde-Rangel

Instituto Politécnico Nacional,  
Unidad Profesional Interdisciplinaria en Ingeniería y  
Tecnologías Avanzadas, Laboratorio de Biomecánica.  
Av. Instituto Politécnico Nacional núm. 2580,  
Col. Barrio la laguna Ticomán, Del. Gustavo A. Madero,  
CP 07340, México DF.  
MÉXICO.

Tel. (+52) 5729 6000 ext. 56866  
correo electrónico (email): aanzuetor@ipn.mx  
mconder0700@ipn.mx

Recibido 15-10-2012, aceptado 31-03-2013.

## Resumen

La implementación en FPGA del proceso de transformación del modelo de color YCbCr al RGB basado en el estándar ITU-R BT.601-5, requiere de tres componentes, un sensor de video analógico que se encarga de la adquisición de datos, una tarjeta digitalizadora que realiza la conversión de video analógico a formato digital teniendo como modelo de color a la salida el YCbCr, finalmente un FPGA (Tarjeta Spartan 3E) que se encarga de procesar el algoritmo para la transformación al modelo de color RGB, utilizando el formato de muestreo 4:4:4. Para corroborar los resultados obtenidos de la implementación en FPGA se hace un comparativo con los datos obtenidos de la función "ycbcr2rgb" contenida en el Toolbox de Digital Image Processing de MATLAB.

**Palabras clave:** sistema de visión artificial, tiempo real, modelo de color (YCbCr y RGB), estándar ITU-R BT.601-5, FPGA (Tarjeta Spartan 3E).

**Abstract**  
**(FPGA Implementation Process Model Transformation YCbCr to RGB Color)**

The FPGA implementation of the transformation process model YCbCr to RGB color based on the ITU-R BT.601-5,

requires three components, an analog video sensor that is responsible for data acquisition, digitizing card that converts analog video to digital format, with the color model to the YCbCr output, finally a FPGA (Spartan 3E board) that is responsible for transforming the RGB color model format using 4:4:4. To corroborate the results obtained from the FPGA implementation is a comparison with the data obtained from the "ycbcr2rgb" contained in Digital Image Processing Toolbox for MATLAB.

**Key words:** artificial vision system, real time, YCbCr and RGB color model, standard ITU-R BT.601-5 FPGA (Spartan 3E Board).

## 1. Introducción

Los sistemas de visión en tiempo real permiten la extracción de información de una escena (u objeto) a partir de su proyección en un plano imagen bidimensional, así mismo, deben satisfacer las restricciones impuestas en su tiempo de respuesta, es decir, el tiempo transcurrido entre la presentación de las entradas y la obtención de sus salidas. Debido a esta característica, los sistemas de visión en tiempo real han tenido gran aceptación en áreas como en el procesamiento de imágenes de video conferencias, telemedicina, monitoreo, robótica, entre otras.

El hardware comúnmente utilizado para desarrollar las aplicaciones de análisis de imágenes son los Procesadores Digitales de Señales (DSP) o los Circuitos de Aplicaciones Específicas (ASIC). Pero a medida que la necesidad aumenta se requieren más sistemas con un mejor desempeño y de menor tiempo de ejecución. Los *Field-Programmable Gate Arrays*, de ahora en adelante nombrado FPGA, son circuitos integrados formados por arreglos de bloques lógicos que permiten reconfigurarse las veces que sea necesario para depurar su funcionalidad, se comunican entre ellos con terminales de entrada/salida por medio de alambros llamados canales de comunicación. El tamaño y la velocidad de los FPGA son equiparables a los ASIC, pero los FPGA son más flexibles, su ciclo de diseño es más corto y el paralelismo da un mayor rendimiento [1,2].

Para cumplir con el estándar ITU-R BT.601-5, la tarjeta digitalizadora (se encarga de la conversión analógica a digital)

VDEC-1 de la compañía DIGILENT entrega la señal de video en formato digital del modelo de color YCbCr, estas señales son transformadas al modelo de color RGB para poder ser interpretadas por un monitor [3,4].

Yang y colaboradores presentan un trabajo conversor del modelo de color YCbCr al RGB [5], tomando un rango de salida para las componentes R (Rojo), G (Verde) y B (Azul) de 240, considerando como hardware de implementación un DSP. En este trabajo se extiende el rango hasta 255 y se simula las ecuaciones propuestas en MATLAB.

La organización de este artículo es la siguiente: en el apartado 2 se presentan 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 utilizando la función "ycbcr2rgb" contenida en el Toolbox de Digital Image Processing de MATLAB [5]. En el apartado 3, se muestran los bloques necesarios para la implementación del sistema de conversión digital entre cada modelo de color (YCbCr y RGB). En el apartado 4, se presentan los resultados y las gráficas obtenidas para la obtención del tiempo de propagación o tiempo de retado (tiempo que tarda el bloque en dar respuesta al presentarse un nuevo dato a la entrada), así mismo, se compraran los resultados numéricos obtenidos por el diseño digital en el FPGA y los mostrados en la sección 2. Finalmente, se presentan las conclusiones obtenidas en el desarrollo de este trabajo.

## 2. Transformación del Modelo de Color YCbCr al RGB

Cumpliendo las recomendaciones del estándar ITU.R BT.601-5 (apartado 3.5.2 [3]), para un sistema normalizado en Cb y Cr, se tiene (1):

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} + (A' \times C') \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (1)$$

de donde,

$$A = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.169 & -0.331 & -0.500 \\ 0.500 & -0.419 & -0.081 \end{bmatrix} \quad (2)$$

$$C = \begin{bmatrix} 220/256 & 0 & 0 \\ 0 & 220/256 & 0 \\ 0 & 0 & 220/256 \end{bmatrix} \quad (3)$$

La ecuación (1), muestra la conversión del modelo de color RGB al modelo de color YCbCr, donde la matriz  $A$  repre-

senta los coeficientes renormalizados de los vectores horizontales,  $Y$  (Luminancia),  $Cb$  (Cromancia en Azul) y  $Cr$  (Cromancia en Rojo) respectivamente, mientras que la matriz  $C$  representa la matriz de recuantificación de los vectores, como se menciona en el estándar ITU-R BT.601-5, (apartado 3.5.3 [3]), dichos vectores sólo pueden ocupar 220 niveles para luminancia y 255 niveles para la cromancia, debido a que el resto de los valores son reservados para señales de sincronización en dispositivos analógicos, con esta recuantificación se amplían todos los vectores a un rango de 256 valores.

Dado que el propósito de este artículo es convertir del modelo de color YCbCr al RGB y en el estándar ITU-R BT.601-5 sólo se menciona la conversión de RGB a YCbCr, es necesario realizar la transformación inversa, como se muestra en (4):

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = D^{-1} \times \begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} \quad (4)$$

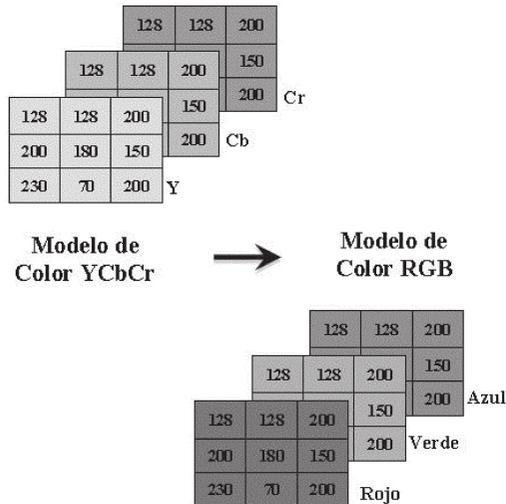
de donde:

$$D = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.169 & -0.331 & -0.500 \\ 0.500 & -0.419 & -0.081 \end{bmatrix} \begin{bmatrix} 220/256 & 0 & 0 \\ 0 & 220/256 & 0 \\ 0 & 0 & 220/256 \end{bmatrix} \quad (5)$$

Se usa la función *ycbcr2rgb* del Toolbox de *Digital Image Processing* de MATLAB [6], se transforman 9 valores de componentes, Y, Cb y Cr a R, G y B, para comparar con los resultados obtenidos de la ecuación (1). La función *ycbcr2rgb* de MATLAB recibe el dato  $Y$  sobre un rango de (16, 235), para  $Cb$  y  $Cr$  en un rango de (16, 240) y entrega el resultado numérico en valores enteros proporcionales de 8 bits, es decir, un rango de (0, 255) para los valores de  $R$ ,  $G$  y  $B$ . Los datos de la conversión se puede observar en la figura 1.

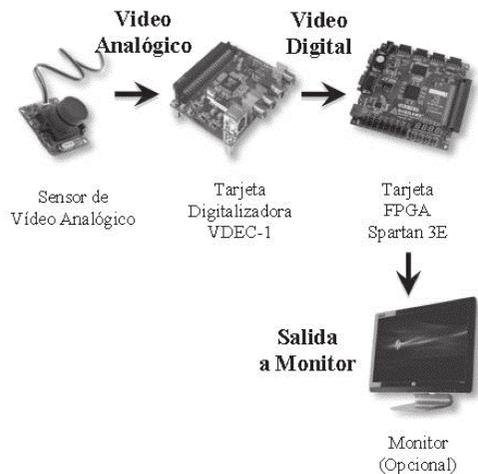
## 3. Bloques de transformación

La figura 2 muestra el diagrama de bloques de las componentes físicas utilizadas en este trabajo, para un mejor entendimiento del proceso de transformación entre modelos de color en la figura 3, se presenta un diagrama de bloques de la secuencia para la obtención de los datos. El bloque *Sensor* representa la adquisición de información en forma analógica, para esta tarea se emplea un sensor de video analógico de tipo CMOS, comercializado por la compañía SparkFun, con una

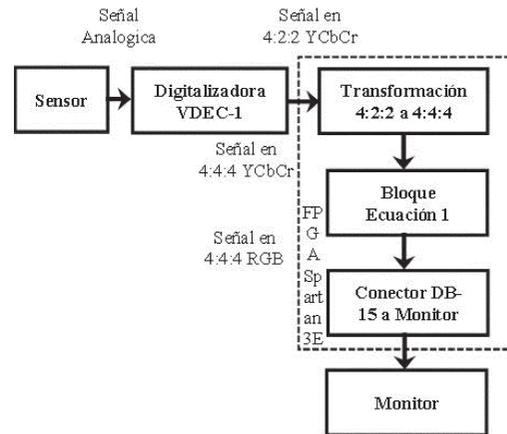


**Fig. 1.** Transformación de datos del modelo de color YCbCr al modelo de color RGB.

resolución de imagen de salida de 640 x 480. El bloque *Digitalizadora* está constituido por la tarjeta digitalizadora VDEC-1 (de la compañía Digilent), que provee la información en formato 4:2:2 de las componentes Y, Cb y Cr, los datos digitales son ingresados en el FPGA XC3S500E Spartan



**Fig. 2.** Diagrama de los componentes físicos de la transformación del modelo de color YCbCr a RGB.



**Fig. 3.** Diagrama de los componentes físicos de la transformación del modelo de color YCbCr a RGB.

3E de la compañía Xilinx para realizar la transformación entre los modelos de color, es necesario modificar el muestreo de los datos de entrada a un tipo de muestreo 4:4:4 [1][6].

La señal YCbCr, con el nuevo tipo de muestreo (4:4:4), ingresa al bloque que realiza la transformación al modelo de color RGB, este bloque es de principal interés en este artículo, ya que en el se implementan las ecuaciones de transformación. La tarjeta Spartan 3E cuenta con un conector tipo DB15 que permite realizar la comunicación con el monitor para presentar los resultados de forma visual.

### 3.1. Implementación del bloque de transformación del modelo de color YCbCr a RGB

La IDE, *ISE Project Navigator* 11.2 provista por la compañía Xilinx y el lenguaje de descripción de hardware VHDL, son herramientas para la realización de esta aplicación. Cabe mencionar que todas las variables y constantes utilizadas se implementan para representar números enteros con signo, usando el sistema binario o base 2.

En la figura 4, se muestran las líneas de código en VHDL para la obtención de la componente *R* (Rojo) del modelo de color RGB.

De la figura 4, la variable *R\_temp* (valor digital de rojo) se determina por la sumatoria de las variables *T1* y *T2*; la variable *T1* se integra por la suma de las variables (*Y-16*) (dado en (2)) y por la multiplicación con el coeficiente *cl*. La va-

```

constant cero      : 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):= "111111111110000000"; --(-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');
    
```

Fig. 4. Adquisición de la componente R (rojo) del modelo de color RGB.

riable nombrada (-16) es la resta del valor 16 a la componente de luminancia (Y), mientras que la variable nombrada c1, es el coeficiente (1, 1) de la matriz de transformación.

La obtención del término T2, se desarrolla un procedimiento similar al utilizarse T1, con la diferencia de emplear las variables (-128) y c2. Para facilitar el manejo de (2), se desarrolla en forma de componente el bloque conversor de color, como el que se muestra en la figura 5.

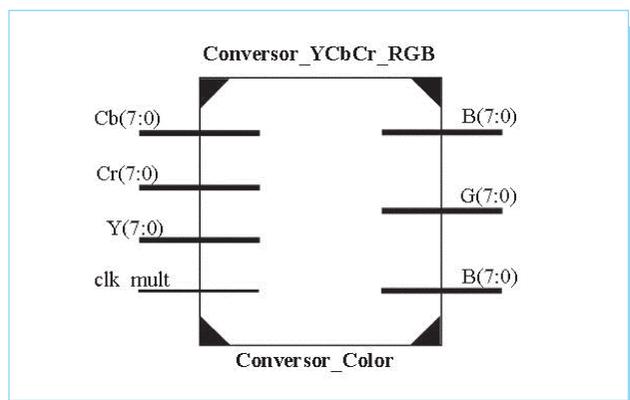


Fig. 5. Diagrama de los componentes físicos de la transformación del modelo de color YCbCr a RGB.

Los valores de entrada del bloque de la figura 5, son las señales Y, Cb y Cr con un bus de 8 bits en formato 4:4:4, de forma similar se entrega a la salida las componentes R, G y B. Por lo que este bloque de forma interna está constituido por 5 multiplicadores que desarrollan los términos T1, T2, T3, T4 y T5, que conforman a (2).

En la figura 6, se muestra el bloque que desarrolla el término  $T1 = c1 * (Y_{temp} + (-16))$ , que se utiliza para la obtención de la componente R, véase figura 6, por lo que los términos T2, T3, T4 y T5 se obtienen de forma similar a T1 (líneas de código de la figura 4), cada uno con sus respectivas variables y coeficientes.

#### 4. Resultados

Se presentan diferentes valores de Y, Cb y Cr para convertir y corroborar la eficiencia del bloque de transformación entre los modelos de color. En la tabla 1, se listan los valores de salida del modelo de color RGB, son obtenidos empleando la función *ycbcr2rgb* del Toolbox de procesamiento de imágenes de MATLAB.

En la tabla 2 se presentan los resultados numéricos de la implementación en VHDL entre los modelos de color.

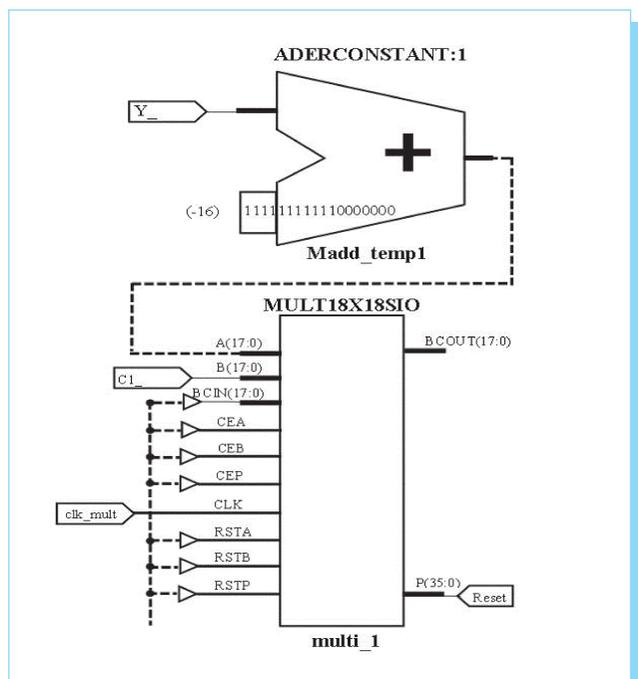


Fig. 6. Bloque gráfico del desarrollo del término  $T1 = c1 * (Y_{tem} + (-16))$ .

**Tabla 1.** Datos de prueba. Función del Toolbox de procesamiento de imágenes de MATLAB.

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 2.** Datos de prueba para la función del Toolbox de procesamiento de imágenes de MATLAB.

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

La gráfica de la figura 7, se obtiene al presentar los diferentes valores de Y, Cb y Cr, así mismo, se obtiene el tiempo de retardo de propagación del bloque conversor de color (véase figura 5), es decir, el tiempo que transcurre a partir de que las señales en las entradas están presentes, hasta que el bloque conversor obtenga los datos resultantes.

Este tiempo se mide a partir de que se presenta un cambio dentro de cualquier señal, ya sea de Y, Cb o Cr hasta que el bloque conversor obtiene la conmutación en las señales R, G o B, teniendo un retardo aproximado de 12 ns.

La discrepancia entre los valores obtenidos, utilizando la función de MATLAB y los valores de la implementación digital, se debe a la resolución tomada por el sistema respecto al punto flotante, por lo tanto, la conversión realiza un corrimiento equivalente a una división de 213 para obtener valores numéricos enteros, este valor puede ser aumentado para mejorar la aproximación, lo que conlleva a utilizar una mayor cantidad de recursos en el dispositivo.

Actualmente el bloque de transformación entre modelos de color, está siendo utilizado como parte de un sistema de visión artificial encargado del guiado y evasión de robots móviles.

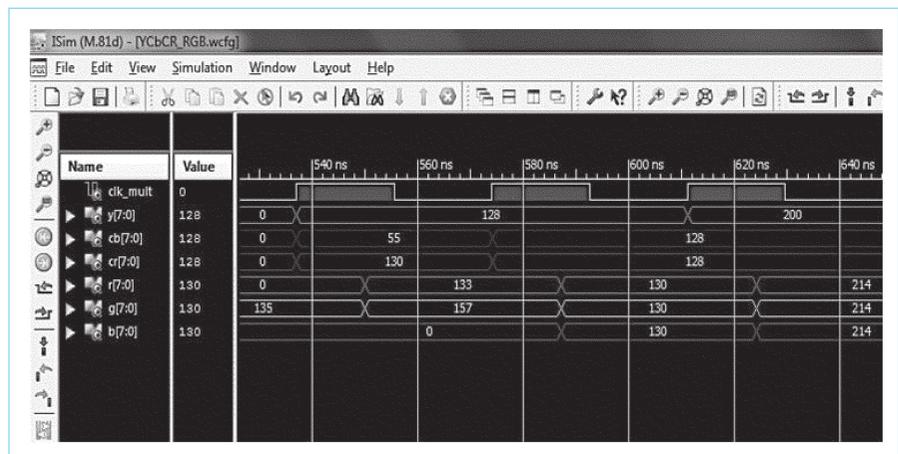
#### 4. Conclusiones

El presente trabajo muestra el desarrollo de un bloque digital para la transformación del modelo de color YCbCr al RGB, empleando el lenguaje descriptivo de Hardware VHDL para su implementación en un entorno de desarrollo programable ISE Project Navigator, utilizando la tarjeta Spartan 3E de la compañía Xilinx. El sistema propuesto se ha implementado sin utilizar herramientas o programas de diseño que contengan bloques prediseñados, logrando con esto trabajar a niveles de abstracción menor, optimizando tiempos de ejecución y recursos utilizados.

El tiempo de propagación de las señales es menor al requerido por la norma ITU-601, por lo tanto, el desarrollo presentado puede ser utilizado en sistemas de procesamiento de video regidos bajo esta norma.

#### Referencias

- [1] J. A. Kalomiros y J. Lygouras, "A host/co-processor fpga-based architecture for fast image processing", in *IEEE International Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications*, pp. 373-378, septiembre, 2007.
- [2] Xilinx, "Spartan-3E FPGA Family: Datasheet product specification", Xilinx, agosto, 2009.
- [3] ITU-R, "ITU-R Recommendation BT.601-5: Studio encoding parameters of digital television for standard 4:3 and wide, screen 16:9 aspect ratios", ITU-R.
- [4] P. Yuhua Y. Yang and L. Zhoanguang, "A fast algorithm for ycbcr to rgb conversion", in *IEEE Transactions on Consumer Electronics*, 53(4): 1490 - 1493, noviembre, 2007.
- [5] C. M. Thompson y L. Shure, *Image Processing Toolbox for Use with MATLAB*, 1993.
- [6] John C. Russ, *The Image Processing Handbook*, 6ª ed., 2011.



**Fig. 5.** Gráficas de propagación de tiempos del bloque conversor del modelo de color YCbCr al RGB.