Reproducción de una Imagen en un Monitor VGA Utilizando un FPGA

Michael Alejandro Diaz Illa, Alfredo Granados Ly

Facultad de Ingeniería Electrónica y Eléctrica, Universidad Nacional Mayor de San Marcos, Lima, Perú

RESUMEN: El presente artículo describe la manera cómo a partir de un programa aplicativo en Visual Basic podemos capturar y enviar la información de una imagen hacia un periférico que contiene un FPGA, el cual a su vez procesa la información y la envía a un monitor VGA para su presentación.

Se ha usado como entrada de diseño el lenguaje VHDL y la implementación del sistema se realizó en el FPGA CYCLONE II 2C35C672C6 de la tarjeta de desarrollo DE2 de la empresa ALTERA utilizando la herramienta de síntesis y simulación QUARTUS II.

ABSTRACT: This paper presents the way like from an application program in Visual

BASIC we can capture and send the information of an image towards a FPGA, which processes the information and sends it to a VGA monitor for his presentation.

PALABRAS CLAVES: FPGA, UART, VGA, Memorias SRAM

I. INTRODUCCIÓN

Hoy en día el tratamiento digital de imágenes es muy importante en el campo de procesamiento digital de señales en donde podemos encontrar muchas aplicaciones. Las imágenes digitales se pueden representar en escala de grises o en escala de colores. Una imagen digital en escala de grises es una matriz de MxN elementos numéricos cuyos valores posibles van del 0 (negro) al 255 (blanco), siendo este número la intensidad luminosa en el determinado punto o píxel. Una imagen digital a colores está formada por tres matrices de MxN, siendo este número la intensidad luminosa en cada una de las bandas espectrales del RGB (Rojo, Verde, Azul), de cada punto o píxel, a diferencia de las imágenes en escala de grises, las imágenes a color requieren de la combinación de las 3 bandas de color, para representar el color de un píxel. Por ejemplo, un determinado punto blanco de una imagen en escala de grises se describiría: P(x, y) = 255, sin embargo en una imagen a colores para describir el color del mismo punto se realizaría así: P(x, y) = (255, 55, 255), esto debido a que el (0,0,0) corresponde al negro absoluto y el (255,255,255) al blanco Absoluto. Por otro lado la implementación de un sistema digital con el lenguaje VHDL [1] en un FPGA disminuye el tiempo de diseño y facilita al diseñador poder simular el diseño antes de poder implementarlo reduciendo los errores. El sistema implementado se muestra en la figura 1 y consta de un programa en Visual Basic, la tarjeta de desarrollo DE2 [2] con un reloj de 50 MHz y un monitor VGA.

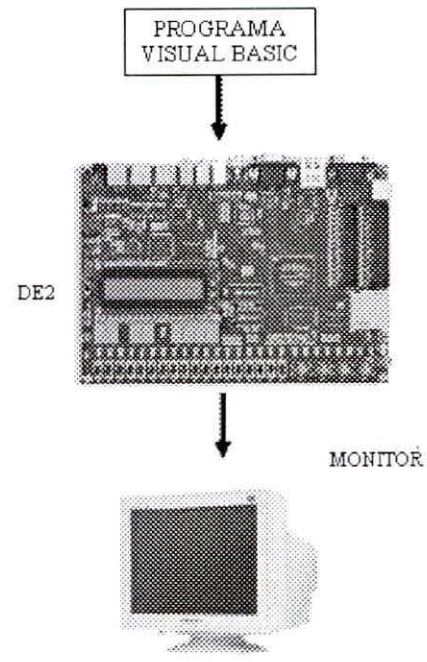

Fig. 1. Diagrama de Bloques del Sistema.

II. DISEÑO DEL SISTEMA

El trabajo se ha dividido en dos partes, una que consiste en el desarrollo de una aplicación en Visual Basic y la otra en el desarrollo de tres módulos descritos en lenguaje VHDL, denominados : *Módulo de Recepción UART, Módulo de Almacenamiento de Datos y el Módulo de Visualizar Pantalla.*

III. APLICACIÓN EN VISUAL BASIC

La figura 2 muestra el diagrama de flujo del programa desarrollado en Visual Basic.

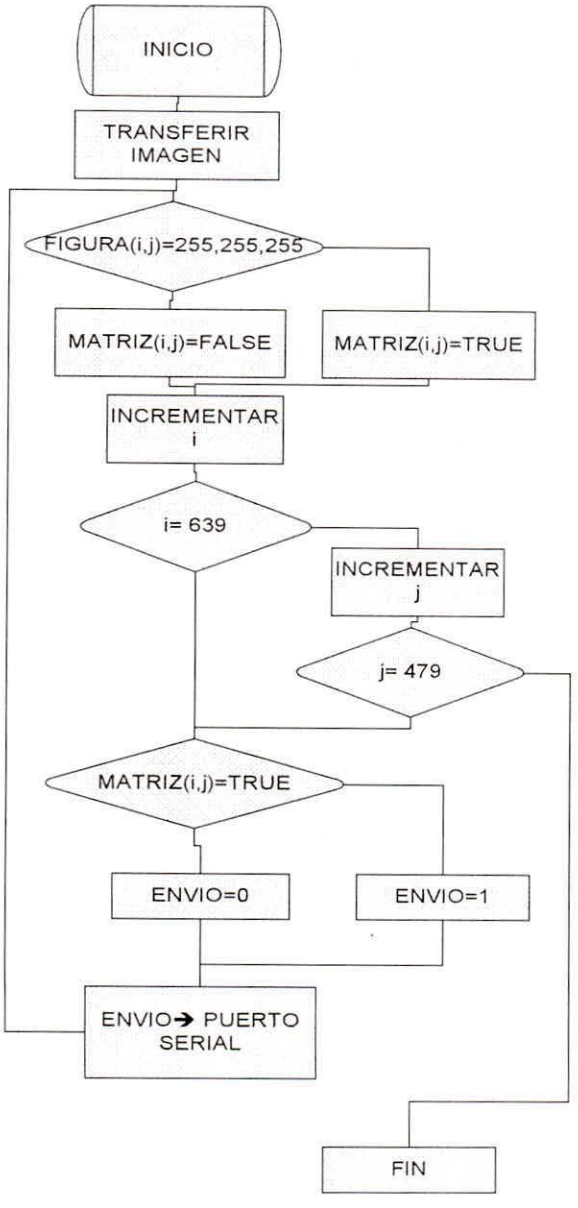

Fig. 2. Diagrama de Flujos del programa en Visual Basic.

El Programa carga una imagen en blanco y negro a escala 640x480 y va escaneando la imagen desde la columna 0 hasta la columna 639 luego incrementa la fila desde 0 hasta la fila 479 donde termina el programa. De esta manera la imagen es llevada a una $matriz_{(i,j)}$, en donde encuentra un valor (255,255,255) pondrá el valor de la $matriz_{(i,j)}$ a TRUE, y cuando encuentre el valor (0,0,0) pondrá un FALSE. Luego de cada escaneada de una columna el programa envía un 1(TRUE) para píxel blanco o un 0(FALSE) para un píxel negro al puerto serial de la computadora a través del objeto MSCOMM [3]. La velocidad de transmisión esta dada en baudios y es configurable en el programa. En la figura 3, se muestra el programa en ejecución.

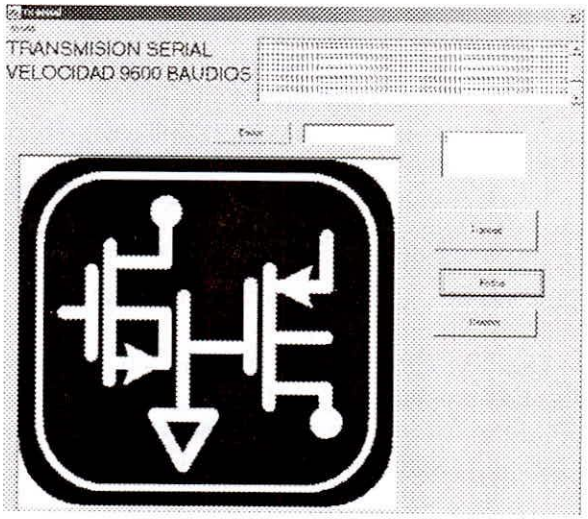

Fig. 3. Programa en Visual Basic.

IV. MÓDULO DE RECEPCIÓN UART

El módulo de recepción UART consta de dos bloques diseñados en VHDL que fueron unidos con el estilo estructural, ver figura 4, uno de los bloques es para fijar la velocidad en baudios de la recepción y el segundo módulo es una maquina de estados que se encarga de almacenar los datos que llegan. El módulo UART toma el dato del circuito integrado MAX232 de la tarjeta DE2 que está conectado a la computadora por un cable serial DB9.

A. Módulo Baudios

Este módulo fue diseñado en VHDL con el estilo algorítmico, ver figura 5, se encarga de fijar la velocidad de recepción. Las velocidades en baudios que se pueden configurar son una combinación de 3 bits y

son 1200, 2400, 4800, 9600, 19200, 38400, 57600 y 115200 baudios.

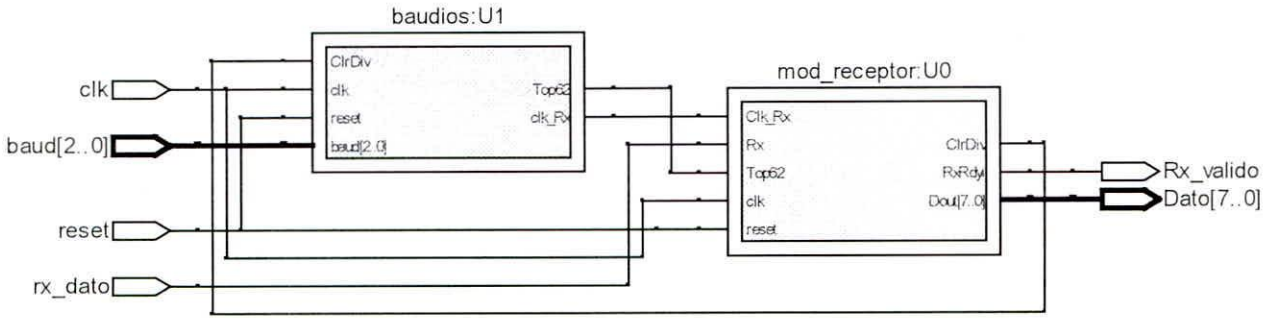

Fig. 4. Módulo de Recepción UART

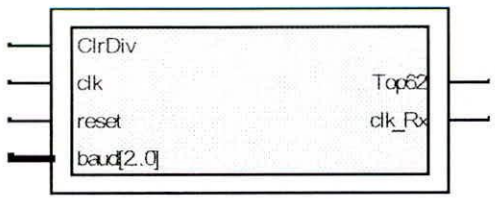

Fig. 5. Módulo Baudios.

V. MÓDULO DE ALMACENAMIENTO DE DATOS

Este módulo consta básicamente de una máquina de estados implementado en VHDL y se muestra en la figura 6. Se encarga de almacenar en un registro los ocho bits que son enviados por la computadora.

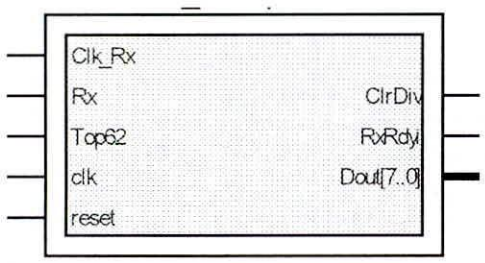

Fig. 6. Módulo de Almacenamiento de Datos.

Cuando se completa la llegada de los ochos bits en el registro, el módulo envía una señal de salida por un instante de tiempo. De esta manera el Módulo de Recepción UART advierte al Módulo de Escritura de Datos el fin de recepción de datos.

A. Módulo de Escritura de Datos en la Memoria SRAM

Este módulo está compuesto por una máquina de estado y se muestra en la figura 7. Recibe los datos del Módulo de Recepción UART y los almacena en la memoria SRAM de la tarjeta de desarrollo DE2. El diagrama de estados que gobierna esta máquina se muestra en la figura 8. El funcionamiento de esta máquina es recibir los 640x480 datos que provienen del módulo de Baudios y almacenarlos en las posiciones de la memoria SRAM, para lo cual utiliza dos contadores. Los datos que le llegan son 1 (píxel blanco) y 0 (píxel negro) y se ponen en formato RGB. Por ejemplo para el píxel blanco sería 111 y para el píxel negro sería 000. Se guardan 4 datos (formato RGB) en una posición de memoria, por lo que sólo se usan los 12 bits menos significativos de una posición de memoria lográndose cubrir 76800 posiciones de memoria para todo el monitor VGA.

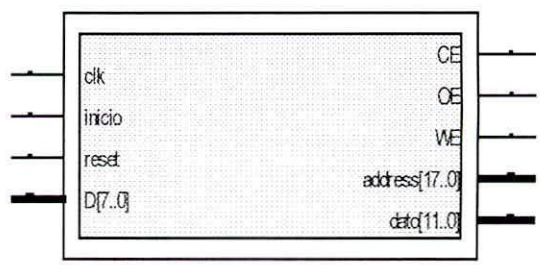

Fig. 7. Diagrama de Bloques de la Memoria SRAM.

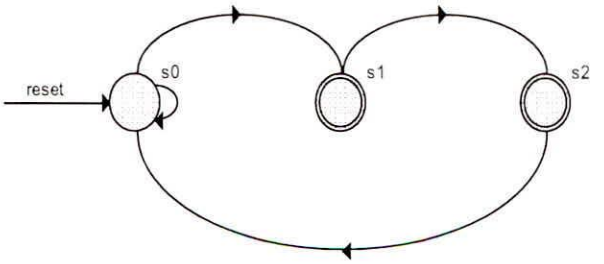

Fig. 8. Diagrama de estado de la unidad de control que gobierna el Módulo de escritura de datos en la SRAM.

La memoria SRAM es un tipo de memoria volátil que no necesita un refresco en sus datos. Los datos se pierden cuando se les quita la alimentación o cuando se graban nuevos datos. Las memorias utilizadas son de la empresa ISSI y tiene una capacidad de 256Kx16 bits de datos con tiempos de accesos muy cortos entre 10 a 15 nanosegundos. El diagrama de bloques se muestra en la figura 9.

VI. MÓDULO DE VISUALIZAR PANTALLA

El módulo de visualizar pantalla se muestra en la figura 10 y consiste de un controlador de memoria que envía los datos de la SRAM de la tarjeta de desarrollo DE2 a un generador de sincronismo con lo cual visualizamos los datos de la memoria en el monitor VGA. La figura 11, muestra el generador de sincronismo, básicamente consiste de un generador de sincronismo horizontal y otro vertical. Estos dos circuitos deben generar señales de acuerdo a la frecuencia y resolución del monitor.

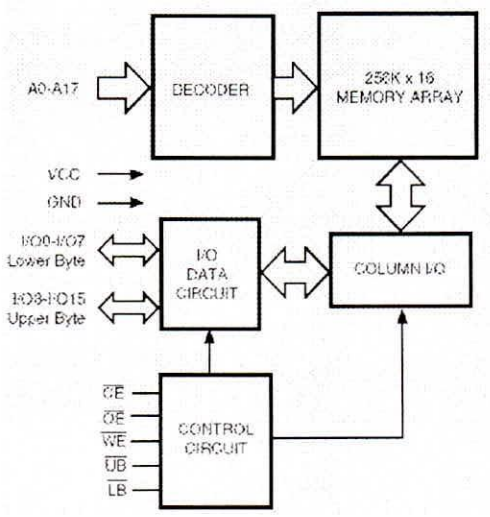

Fig. 9. Diagrama de Bloques de la Memoria SRAM de 256Kx16 de la empresa ISSI.

VI. MÓDULO DE VISUALIZAR PANTALLA

El módulo de visualizar pantalla se muestra en la figura 10 y consiste de un controlador de memoria que envía los datos de la SRAM de la tarjeta de desarrollo DE2 a un generador de sincronismo con lo cual visualizamos los datos de la memoria en el monitor VGA. La figura 11, muestra el generador de sincronismo, básicamente consiste de un generador de sincronismo horizontal y otro vertical. Estos dos circuitos deben generar señales de acuerdo a la frecuencia y resolución del monitor. Los sincronismos vertical y horizontal que son contadores sirven para dar la dirección al controlador de memoria y luego este toma los datos de la memoria y los lleva a las entradas RGB del módulo de sincronismo. De esta manera el módulo lee las porciones de memoria y envía los datos (píxeles) al monitor.

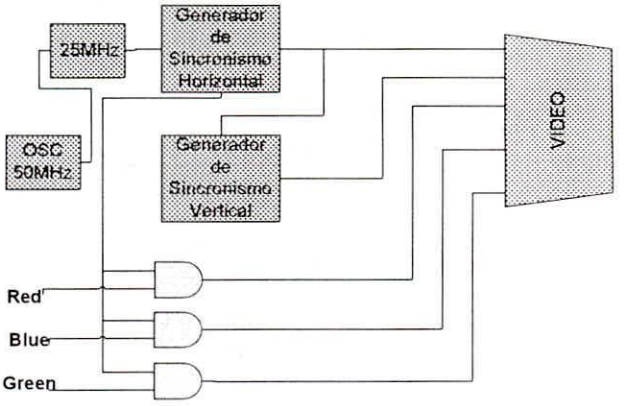

Fig. 11. Generador de Sincronismo.

El controlador de VGA que posee la tarjeta DE2 tiene tres DAC´s para el R, G y B de 10 bits cada uno, ver figura 12. De esta manera, se puede tener muchas más combinaciones de colores.

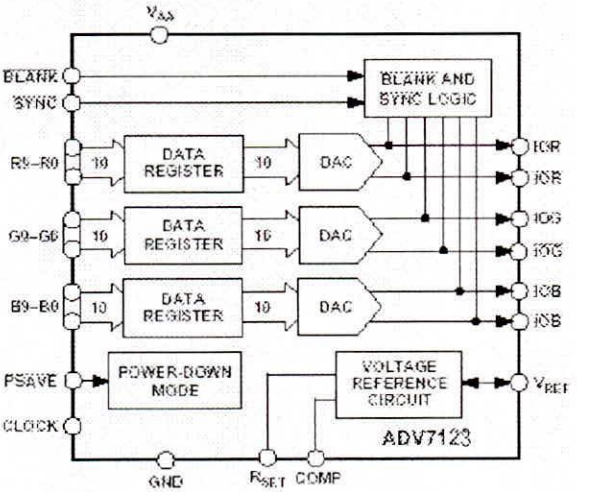

Fig. 12. Generador de Sincronismo.

RESULTADOS

La figura 13, muestra el instante que se envía una foto al monitor VGA conectado al sistema, cabe señalar que en ese instante el sistema se está ejecutando pudiendo visualizarse casi la mitad de la figura 3.

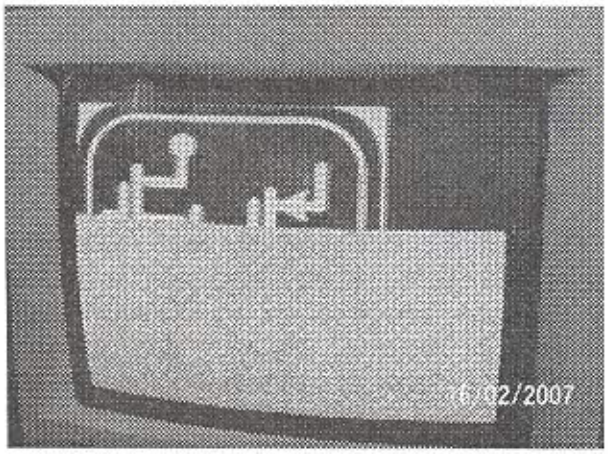

Fig. 13. Foto tomada en el instante en que se pinta la imagen en un Monitor VGA conectado al Sistema.

VI. CONCLUSIONES

Se llegó a implementar el sistema digital para reproducir una imagen en un monitor VGA. El tiempo de diseño del sistema fue de tres días incluyendo la aplicación en Visual Basic.

Se estudió el Standard RS232 para el envío de datos desde una computadora al FGPA, además se estudio los tiempos de sincronismo que necesita el monitor VGA para poder ver una imagen. Con este sistema sólo se envió una imagen en blanco y negro y se deberá cambiar el diseño para enviar una imagen a colores.

Se han usado un total del 10 % de elementos lógicos del FPGA CYCLONE II 2C35C672C6 de la tarjeta de desarrollo DE2. Se realizaron pruebas con una velocidad de 9600 y 57600 y se vio que la imagen demora en pasar desde el programa en Visual Basic al monitor en más de 1 hora. Con esto podemos sugerir que para trabajos posteriores se deberá usar el protocolo USB de mayor velocidad.

VII. REFERENCIAS

[1] Brown, Stephen. "Fundamentals of digital logia with VHDL design", 2005.

[2] "DE2 ALTERA Board User Guide." http://www.altera.com/education/univ/materials/boards/DE2_UserManual.pdf , May 2006.

[3]Aprenda Visual Basic 6.0 Ya, Halvorson, Michael, McGraw-Hill, 1999.