

Paint con FPGA: Sistema para la Evaluación de las Interfaces VGA y PS/2 de la Tarjeta de Desarrollo Nexys2

Felipe Santiago Espinosa, **Josué Ayala Ramírez, ***Andrés Ramírez Macías, *Roberto Carlos Vázquez Carrasco**

Instituto de Electrónica y Mecatrónica
Universidad Tecnológica de la Mixteca
Carretera a Acatlima Km. 2.5, Huajuapán de León, Oaxaca, 69000, México
Tel. 953-53-202-14, correo electrónico: *fsantiag@mixteco.utm.mx, **joshu_10@hotmail.com,
max_mousetv@hotmail.com, *houc_darko89@hotmail.com

Resumen

En el presente documento se describe la implementación en un FPGA de un sistema que emula el funcionamiento de un programa para dibujar, similar al programa Paint de Windows. La implementación fue realizada en la tarjeta de desarrollo Nexys2, manufacturada por la empresa Digilent. La tarjeta incluye un FPGA Spartan3E-500 de la firma Xilinx, como elemento principal de procesamiento, además de otros dispositivos para el almacenamiento de datos y el manejo de periféricos. El sistema se desarrolló con la finalidad de evaluar dos de las interfaces de la tarjeta: VGA y PS/2. El diseño requirió de un planteamiento modular, empleando una metodología Top-Down, para posteriormente implementar todos los módulos codificando en VHDL. La interfaz PS/2 se utilizó para la entrada de información, por medio de un ratón. La interfaz VGA se manejó bajo un esquema de 640 x 480 píxeles, dedicando 8 bits por píxel. Por lo tanto, también se utilizó a la memoria RAM M45W8MW16, de la firma Micron, para alojar la información necesaria para el refresco de la pantalla. La memoria tiene una capacidad de 8 Mbyte x 16 bits y es parte de los circuitos integrados incluidos en la tarjeta Nexys 2.

Palabras clave: FPGA, Paint, PS/2, SRAM, VGA, VHDL.

I. Introducción

En la actualidad, el uso de FPGAs para la implementación de sistemas basados en hardware y software cada vez es más demandante. Los FPGAs son dispositivos que proporcionan alta velocidad de procesamiento, flexibilidad en su organización y sobre todo, concurrencia en la realización de tareas. Características que los hacen elegibles para algunas aplicaciones, ante microprocesadores o circuitos integrados a medida [1].

En el mercado se ofertan diferentes tarjetas de evaluación y desarrollo, las cuales, además de un FPGA, incluyen una diversidad de periféricos para aumentar su versatilidad. Un ejemplo es la tarjeta Nexys2, manufacturada y distribuida por la

empresa Digilent [2], en la figura 1 se muestra la vista superior de esta tarjeta.

Las características principales de la tarjeta son [3]:

- FPGA Spartan-3E de la firma Xilinx, con 500K compuertas.
- Memoria Micron M45W8MW16, PSDRAM de 16 MByte.
- Memoria Flash de Intel de 16 MByte.
- Oscilador de 50 MHz.
- 75 terminales de E/S directas del FPGA, disponibles en los conectores de expansión (conector Hirose FX2 con 43 señales y 4 conectores PMod, cada uno con 8 señales).

- Periféricos para evaluación que incluyen: 8 LEDs, 4 displays de 7 segmentos, 4 botones, 8 interruptores deslizantes.
- Puertos: PS/2, VGA y RS232 para aplicaciones. USB para programación y alimentación.

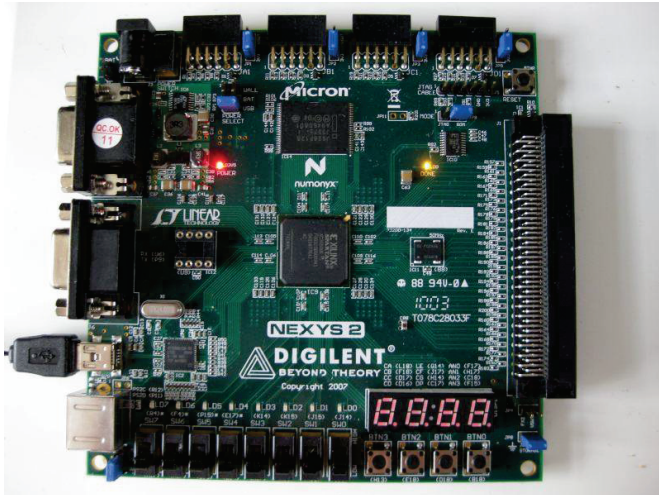


Fig. 1. Vista superior de la tarjeta Nexys2.

Con todas sus características, la tarjeta Nexys2 es ideal durante el proceso de aprendizaje de FPGAs, permitiendo realizar sistemas completos en un solo dispositivo.

Cualquier sistema electrónico requerirá de periféricos para entrada y salida de datos, la tarjeta Nexys2 cuenta con diferentes puertos para ello. En [1, 4, 5] se documentan algunos módulos para el

manejo de periféricos con un FPGA, empleando VHDL.

Específicamente, en [4] se detallan 2 controladores para VGA, uno del tipo gráfico y otro del tipo texto. El controlador del tipo gráfico es la base para un juego simple de ping-pong, ejemplo orientado para la tarjeta *Spartan-3 Starter*, codificado en VHDL.

Con el sistema Paint se busca la evaluación práctica del puerto PS/2 y VGA, en el primero se conecta un ratón y en el segundo un monitor, creando un sistema interesante y de fácil manejo, el cual sirve de base para el desarrollo de sistemas con requerimientos mayores de procesamiento.

II. Organización del Sistema Paint

El sistema se diseñó empleando la metodología Top-Down, que consiste en la división de un sistema complejo en módulos de menor jerarquía, los cuales a su vez se conformarán por otros módulos, hasta llegar a niveles tan simples, cuya implementación no requerirá de mucho esfuerzo [6, 7].

En la figura 2 se muestra la organización jerárquica del sistema, en donde se observan 4 módulos principales y 6 bloques en un segundo nivel. Todos los módulos se codificaron en VHDL, aunque para algunos sólo fue necesaria una adecuación a partir de las descripciones encontradas en [4]. En el diseño de sistemas basados en VHDL es común y conveniente la reutilización de módulos [8].

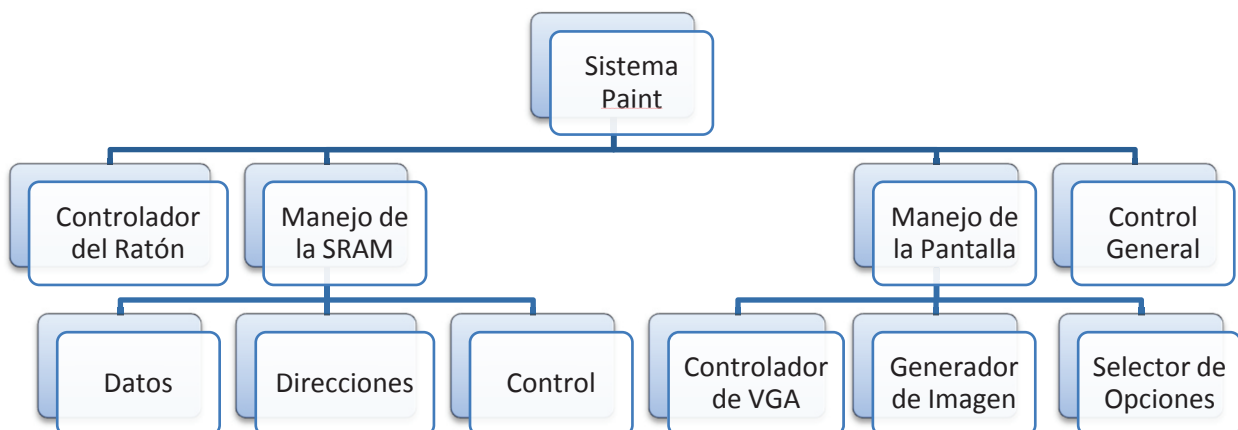


Fig. 2. Organización jerárquica del Sistema Paint.

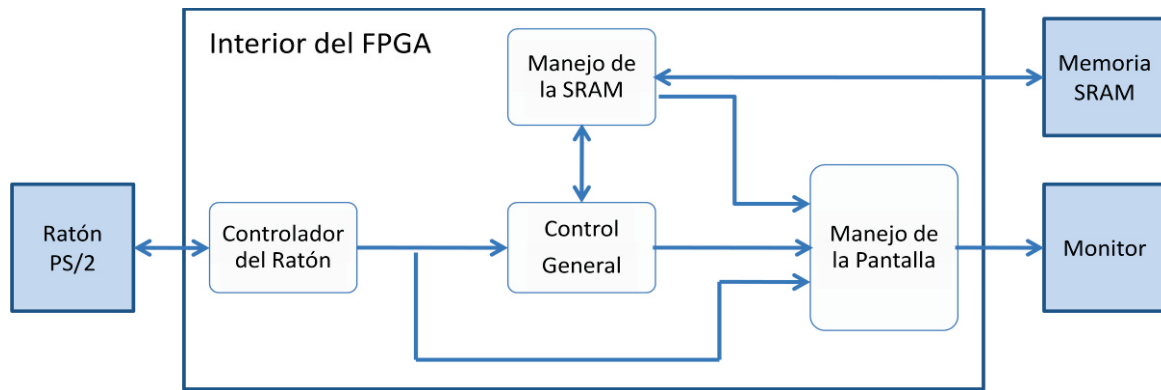


Fig. 3. Diagrama a bloques del Sistema Paint.

El módulo más importante es el Control General, puesto que se encarga de sincronizar y controlar a los módulos restantes (con excepción del Controlador del Ratón).

En la figura 3 se muestra la relación existente entre los módulos de mayor jerarquía.

Cuando el Controlador del Ratón establece la conexión con el dispositivo, envía señales de posición y activación de alguno de sus botones al Control General, el cuál gestiona la escritura o lectura de determinados pixeles en la SRAM, la dirección en SRAM se relaciona con la posición en la pantalla y el contenido con color del pixel.

El Manejo de la Pantalla implica la generación de las señales de sincronía para el monitor, la lectura de la SRAM para obtener el valor de los pixeles que ha escrito el usuario y la generación de pixeles con valores preestablecidos para crear la interfaz gráfica.

En las siguientes secciones se describen los módulos que integran al Sistema Paint, con sus correspondientes bloques.

III. Controlador del Ratón

Este módulo hace posible la interacción con el usuario. El módulo indica la posición de la pantalla en donde se encuentra el puntero del ratón y si se ha presionado alguno de sus botones, en la figura 4 se pueden ver sus entradas y salidas.

Las señales PS2_CLK y PS2_DATA son líneas bidireccionales propias de la interfaz PS/2, se

mantendrán en un nivel lógico alto mientras no haya actividad en el ratón.

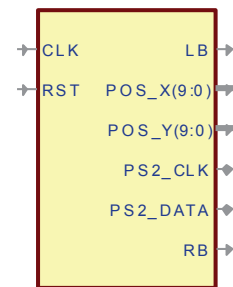


Fig. 4. Controlador del Ratón.

Cada vez que el ratón se mueve o si se presiona uno de sus botones, envía una secuencia de 3 palabras de 11 bits al FPGA. Los 11 bits incluyen: 1 bit de inicio (siempre es '0'), 8 bits de datos (primero el LSB), 1 bit de paridad par y 1 bit de paro (siempre es '1') [3].

El ratón genera la señal de reloj, la cual puede estar en un rango de 20 a 30 KHz. El FPGA va tomando la información en cada flanco de bajada. De los 24 bits de datos, 10 son empleados para indicar la ubicación en cada coordenada, X o Y, 2 indican las actividades en los botones y los otros 2 permanecen sin uso, en un ratón con 2 botones.

El Controlador del Ratón se basa en una máquina de estados por medio de la cual se sondearán las señales PS2_CLK y PS2_DATA, dejando en los vectores POS_X y POS_Y la posición del cursor y poniendo en alto a LB cuando se presione al botón izquierdo o a RB si se presiona el botón derecho.

IV. Manejo de la SRAM

Con este módulo se tendrá acceso a las localidades de la memoria SRAM externa, ya sea para guardar o extraer un dato. Se trata de una memoria Micron M45W8MW16, PSDRAM de 16 MByte, la cual puede trabajar en modo síncrono o asíncrono, en la figura 5 se ilustra el diagrama de tiempos para un acceso en modo asíncrono [9].

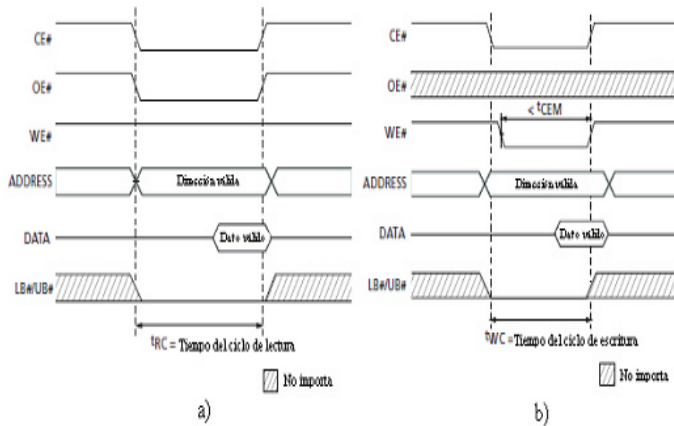


Fig. 5. Tiempos de acceso a la SRAM M45W8MW16 en modo asíncrono (a) Operación de lectura (b) Operación de escritura.

El tiempo mínimo requerido para una operación de escritura y lectura en el modo asíncrono es de 70nS. En el modo síncrono se puede utilizar una frecuencia hasta de 80Mhz.

En la tabla 1 se muestran las señales necesarias para el manejo de la SRAM.

Manejo de Datos

Aunque la memoria tiene un bus de datos de 16 bits, únicamente se emplearon datos de 8 bits, porque cada dato va a corresponder con el valor de un píxel.

Es necesario un bloque para el manejo de los datos porque el bus es bidireccional, por lo tanto, su objetivo básicamente consiste en la separación del bus bidireccional en 2 buses, uno de entrada y otro de salida.

En la figura 6 se muestran las señales de este bloque, con R y W se determina si se trata de una lectura o escritura. Mientras no exista una escritura el bus será puesto en alta impedancia, pudiendo realizar únicamente lecturas.

Señal	Descripción
ADDR (23:0)	Bus de direcciones de 24 bits.
DATA (15:0)	Bus de datos de 16 bits (bidireccional).
OE	Señal de habilitación de salida.
WE	Señal de habilitación de escritura.
MT-ADV	Dirección válida.
MT-CLK	Señal de reloj para SRAM en modo síncrono.
MT-UB	Señal de activación de byte superior para direccionamiento por byte.
MT-LB	Señal de activación de byte inferior para direccionamiento por byte.
MT-CE	Habilitación del chip.
MT-WAIT	Señal de espera para SRAM en modo síncrono.

Tabla 1. Señales necesarias para la SRAM.

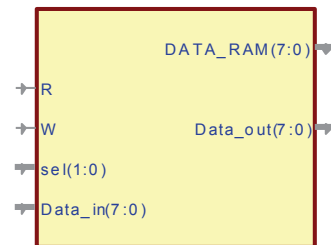


Fig. 6. Bloque para el manejo de datos de la SRAM.

Con el bus de selección se determina si se escribirá el dato o si la localidad será limpiada (escribiendo 0xFF), el bus es de 2 bits para hacerlo compatible con el manejo de direcciones.

Manejo de Direcciones

El manejo de direcciones es fundamental, dado que se podrán realizar diferentes acciones. En realidad este bloque se compone de 4 bloques de menor jerarquía, como se muestra en la figura 7, aunque todos ellos son muy simples.

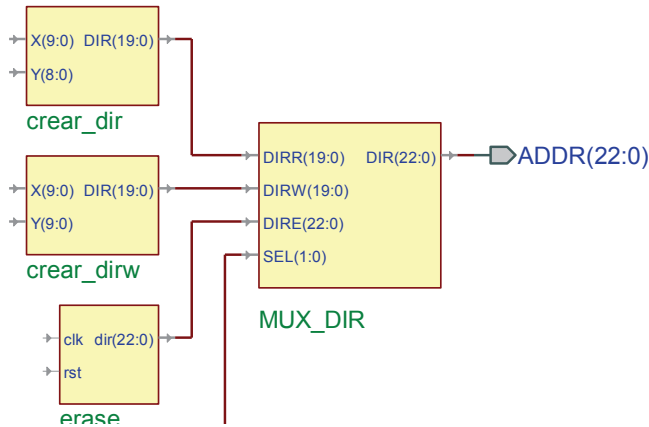


Fig. 7. Bloque para la generación de las direcciones de acceso a la SRAM.

La memoria normalmente será leída, en este caso la dirección se crea a partir de las coordenadas de los pixeles, durante el recorrido para su despliegue, proporcionadas por el Controlador de VGA.

La memoria también será escrita, cuando se mantenga presionado el botón izquierdo del ratón, la dirección se genera con las coordenadas proporcionadas por el Controlador del Ratón.

Finalmente, la memoria será completamente borrada (escribiendo 0xFF en todo su contenido), esta acción se realizará con la inicialización del sistema, con este contenido, en la pantalla se tendrá un color blanco. La selección de la dirección se realiza con el bus SEL, es cual es generado por el Control General, para el manejo del multiplexor de la salida del bloque.

Control de la SRAM

El bloque de control de la SRAM genera las señales necesarias para su manejo. En la figura 8 se muestran sus entradas y salidas, la SRAM normalmente se estará leyendo, a menos que se indique que se realizará una escritura.

Este es un bloque secuencial basado en una máquina de estados, dado que la señal de reloj de la tarjeta tiene una frecuencia de 50 MHz (periodo de 20 nS), con diferentes estados se cumplirá el tiempo mínimo de 70 nS, mientras se realiza la secuencia de habilitaciones correspondiente (ver figura 5).

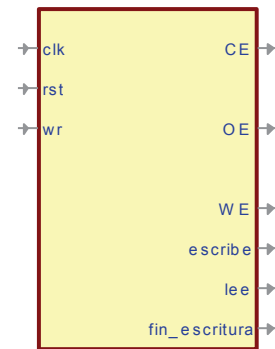


Fig. 8. Bloque para la generación de las señales de control para la SRAM.

V. Manejo de la Pantalla

Las señales necesarias para producir una imagen son [5]:

Señal de luminancia. Prácticamente es la señal de video, pero solo representa un punto de la imagen (pixel). Para desplegar la imagen completa se tiene que mostrar elemento por elemento, por lo cual se debe realizar una exploración, como se muestra en la figura 9.

La exploración se realiza con dos contadores, uno para el barrido horizontal y otro para el vertical. El contador horizontal recorre de izquierda a derecha, y el vertical de arriba a abajo. Una vez que se ha alcanzado el punto inferior derecho se continuará nuevamente con el punto superior izquierdo.

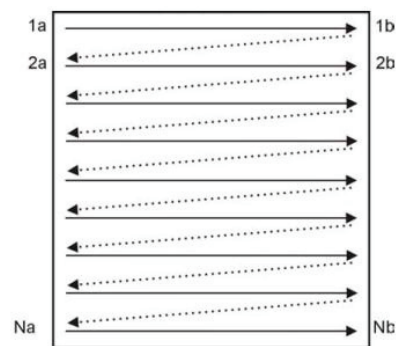


Fig. 9. Exploración de la pantalla.

Pulsos de sincronización. Se utilizan para controlar la trayectoria del haz de electrones. Hay pulsos de sincronización horizontal y vertical, ambos son activos en un nivel lógico bajo. El pulso horizontal marca el inicio y fin de una línea de

exploración, mientras que el pulso vertical indica el inicio y fin de una pantalla.

Pulsos de blanqueo. El objetivo de estos pulsos es poner la pantalla en negro, puesto que durante ese tiempo no se desplegará ningún punto de la imagen. La señal de blanqueo horizontal es independiente a la de blanqueo vertical.

En el sistema Paint, la escritura en la SRAM se realizará únicamente durante los pulsos de blanqueo, ya que cuando éstos no ocurran, se realizarán lecturas.

En la figura 10 se muestra la temporización en las señales empleadas para el manejo de la pantalla.

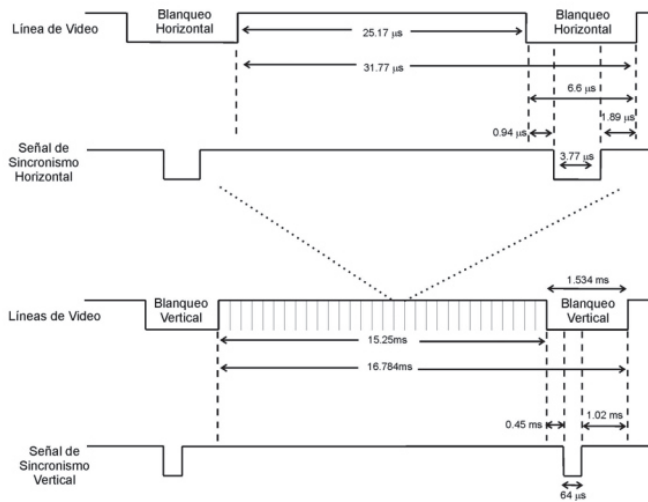


Fig. 10. Temporización en las señales de la pantalla.

El módulo para el manejo de la pantalla incluye 3 bloques: el Controlador de VGA, para generar las señales de sincronía, el Generador de Imagen, para determinar si en la pantalla se colocará un pixel obtenido de SRAM o un pixel con un valor predefinido, y el Selector de Opciones, para definir el color a mostrar en la pantalla, en el momento en que se presione el botón izquierdo del ratón.

Controlador de VGA

En la figura 11 se muestran la interfaz de este bloque, el cual se va a conectar directamente con el puerto VGA.

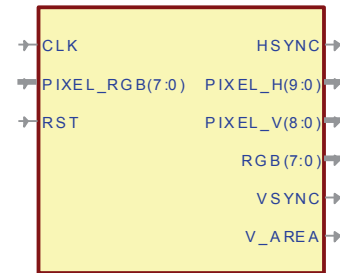


Fig. 11. Bloque Controlador de VGA.

En HSYNC y VSYNC se tendrán los pulsos de sincronía, de acuerdo con los tiempos mostrados en la figura 10, y en RGB se colocará el valor instantáneo del pixel.

El bloque se basa en contadores, los cuales se reflejarán en los vectores PIXEL_H y PIXEL_V, para que internamente se identifique el pixel al que se está haciendo referencia. La señal V_AREA permitirá detectar si un pulso de blanqueo está activo, para considerar una posible escritura en SRAM.

Generador de Imagen

Con este bloque se determina si en la pantalla se colocará un pixel obtenido de SRAM o un pixel con un valor predefinido. El bloque genera la información que corresponde con el puntero del ratón y con las opciones que se muestran en la pantalla, como la paleta de colores, la selección del lápiz o del borrador.

En la figura 12 se muestra la interfaz de este bloque, se observa que recibe las coordenadas del Controlador de VGA y del Controlador del Ratón, comparando ambas coordenadas se determina si en RGB_sal se colocará la información del cursor del ratón o se dejará la información recibida en RGB_ENT.

La entrada SEL define si se mostrará el lápiz o el borrador. La entrada Ver_color representa el color que actualmente está habilitado (el color con el que se pintará en la pantalla).

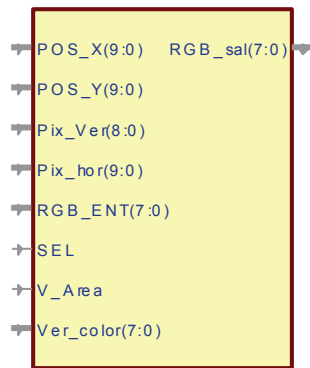


Fig. 12. Bloque Generador de Imagen.

Selector de Opciones

Como su nombre lo indica, este bloque tiene como objetivo almacenar el color con el que se pintará y limitar el área de las barras de herramientas y paletas (Figura 13). Las opciones mostradas en pantalla son: Color, Lápiz y Borrador.

El valor del color activo se almacenará en la SRAM cuando el usuario dé un clic izquierdo a una parte del área de dibujo.

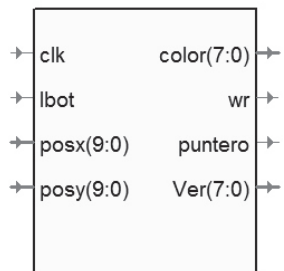


Fig.13 Bloque Selector de Opciones.

El bloque verifica la posición del ratón para activar la opción que corresponda con el área en donde el usuario haya dado un clic izquierdo.

VI. Control General

El Control General gestiona las acciones que deben tomar los distintos módulos, a partir de las decisiones tomadas por el usuario. También sincroniza el tiempo de escritura, ya que sólo se puede escribir en la SRAM cuando se presenta la etapa de blanqueo. En la figura 14 se muestra su interfaz.

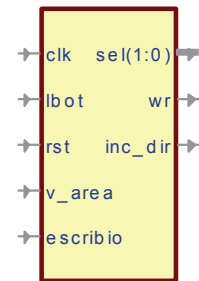


Fig. 14. Módulo del Control General.

Al alimentar el sistema, el Control General hace que las localidades de la memoria sean llenadas con el valor 0xFF, que corresponde al color blanco, con el fin de no contener “basura” en la memoria y mostrar la pantalla en blanco.

Posteriormente activa al control de la SRAM en modo de lectura y permanece en ese estado hasta que el usuario, al presionar el botón izquierdo del ratón, active la escritura en la SRAM.

Desde el Control General se determina si en la SRAM se realizará una lectura o escritura, y por lo tanto, también se define de donde proviene la dirección de acceso.

El comportamiento del Control General se esquematiza con la máquina de estados mostrada en la figura 15.

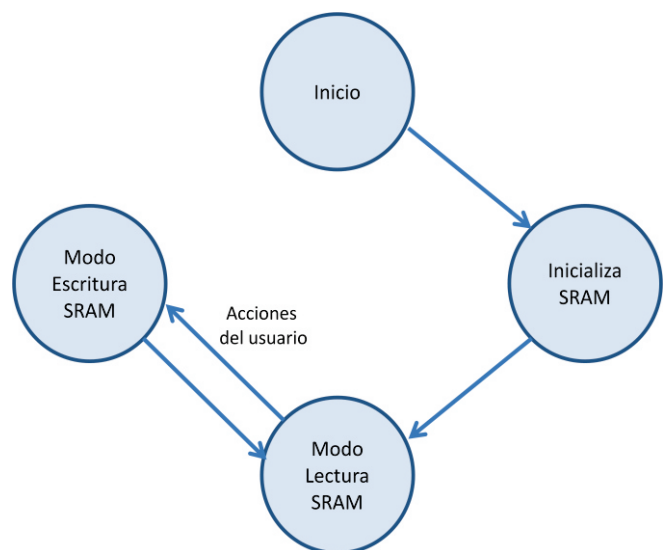


Fig.15. Máquina de estados del Control General.

VII. Resultados

Integrando todos los módulos se construye al sistema Paint, cuya interfaz es mostrada en la figura 16. Las señales en la parte superior corresponden a la memoria SRAM, aunque se agregan algunas para inhabilitar a la memoria Flash, dado que comparten algunos buses en la tarjeta Nexys2.

Las siguientes señales corresponden con el puerto VGA y por último se tienen las señales para el manejo del ratón, las salidas “lbot” y “rbot” hacen referencia a los botones, mostrando su estado en 2 LEDs de la tarjeta.

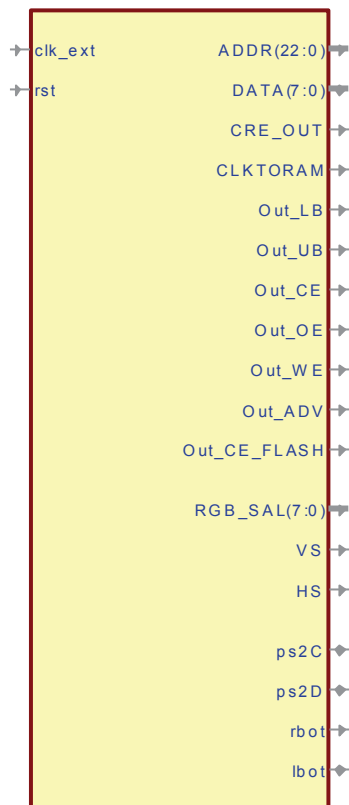


Fig. 16 Sistema Paint.

Se probó al sistema completo, obteniendo un buen desempeño al momento de mostrar los pixeles pintados en la pantalla. Las opciones colocadas en la barra de herramientas cumplen con los requerimientos mínimos de un sistema para realizar dibujos.

En la figura 17 se muestra la interfaz gráfica que genera el sistema. Se puede observar la barra de herramientas en la parte superior de la pantalla, la cual contiene una área de opciones de puntero (parte superior izquierda) y una paleta de selección de colores. El área de trabajo se ha limitado a la parte inferior de la barra de herramientas impidiendo que el usuario pinte sobre el área de la paleta y demás opciones.

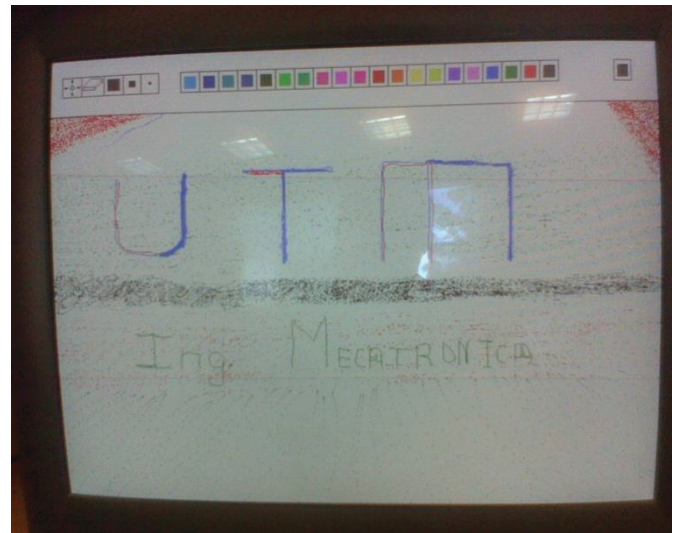


Fig. 17. Sistema Paint, puesto en marcha.

En la figura 18 se puede ver la conexión del monitor en el puerto VGA, y la conexión del ratón en el puerto PS/2. Ambos puertos en la tarjeta Nexys2.

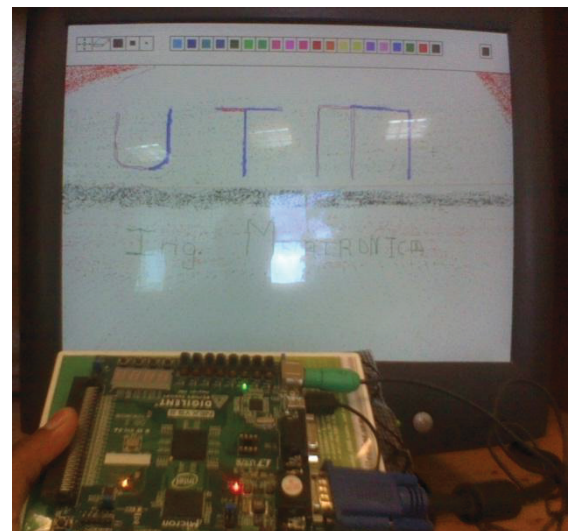


Fig. 18. Sistema implementado.

En la tabla 2 se muestra un resumen de los recursos ocupados por el sistema Paint en el FPGA Spartan3E, demostrando que son muy pocos, con respecto al total de recursos disponibles.

Recurso	Usado	Disponible	Uso
Slices	858	4, 656	18 %
Bloques I/O	56	232	24 %
Bloques RAM	0	20	0 %
GCLKs	4	24	16 %

Tabla 2. Recursos empleados por el sistema Paint.

VIII. Conclusiones

El sistema Paint permitió evaluar favorablemente los puertos VGA y PS/2 incluidos en la tarjeta Nexys2, se diseñó e implementó un sistema completo que hace uso de ellos.

Además, también fue necesario el uso de la memoria M45W8MW16, incluida en la tarjeta. En principio no se había considerado, sin embargo, con una resolución en el monitor de 640 x 480 y colores de 8 bits, implica un total de 307, 200 bytes, que agotan los recursos del FPGA.

El sistema funcionó de manera correcta, no obstante, cuando se encuentra presionado el botón izquierdo del ratón y éste se mueve rápidamente, no se alcanzan a pintar todos los puntos en la pantalla, esto puede deberse a que la duración del pulso de blanqueo no es suficiente para escribir en todas las direcciones relacionadas. El acceso a la memoria es asíncrono, por lo que una posible mejora consistiría en un acceso síncrono, el cual puede realizarse a una velocidad más alta, con ello, se podrán escribir más píxeles durante los pulsos de blanqueo.

En la tabla 2 se puede ver que aún quedan demasiados recursos para el desarrollo de aplicaciones, basadas en la tarjeta Nexys2, que requieran utilizar un ratón y un monitor, como elementos de entrada y salida. Por ejemplo, podría acondicionarse un brazo robótico, haciendo que el

brazo dibuje en alguna superficie lo que el usuario dibujó en la pantalla.

IX. Referencias

- [1] P. Wilson, "Design Recipes for FPGAs", 2007, Newnes of Elsevier.
- [2] Digilent Inc. - Digital Design Engineer's Source, <http://www.digilentinc.com/>, última visita: Junio de 2011.
- [3] Digilent Nexys2 Board Reference Manual, Doc: 502-107, Copyright Digilent, Inc., June 21, 2008.
- [4] P. Chu, "FPGA Prototyping by VHDL Examples, Xilinx Spartan-3Version", A John Wiley & Sons inc., Publication 2008
- [5] J. C. Tepozán Ríos, "Diseño de controladores para la tarjeta de desarrollo XSA-100". Tesis de Ingeniería, Universidad Tecnológica de la Mixteca, Mayo de 2004.
- [6] S. Brown, Z. Vranesic, "Fundamentos de Lógica Digital con Diseño VHDL". 2ª Ed., 2006, Editorial McGraw-Hill.
- [7] R. Romero Troncoso, "Sistemas Digitales con VHDL"; Legaria Ediciones.
- [8] F. Pardo, J. A. Boluda, "VHDL, Lenguaje para síntesis y modelado de circuitos", Ed. Alfaomega, 2000
- [9] M. A. Mendoza Manzano, "Procesamiento y Análisis Digital de Imágenes Mediante Dispositivos Lógicos Programables", Tesis de Ingeniería, Universidad Tecnológica de la Mixteca, Febrero de 2009.

X. Autores

M. C. Felipe Santiago Espinosa es Maestro en Ciencias con especialidad en Electrónica por parte del INAOE, Licenciado en Electrónica por parte de la BUAP, desde 1998 labora como Profesor – Investigador en la Universidad Tecnológica de la Mixteca, adscrito al instituto de Electrónica y Mecatrónica.

Josué Ayala Ramírez es estudiante de la Ingeniería en Mecatrónica, en la Universidad Tecnológica de la Mixteca.

Andrés Ramírez Macías es estudiante de la Ingeniería en Mecatrónica, en la Universidad Tecnológica de la Mixteca.

Roberto Carlos Vázquez Carrasco es estudiante de la Ingeniería en Mecatrónica, en la Universidad Tecnológica de la Mixteca.